

ગુજરાત રાજ્યના શિક્ષણવિભાગના પત્ર-ક્રમાંક
મશબ/1213/110/છ, તા. 18-3-2013-થી મંજૂર

કમ્પ્યુટર-અધ્યયન

ધોરણ 11



પ્રતિજ્ઞાપત્ર

ભારત મારો દેશ છે.

બધાં ભારતીયો મારાં ભાઈબહેન છે.

હું મારા દેશને ચાહું છું અને તેના સમૃદ્ધ અને

વૈવિધ્યપૂર્ણ વારસાનો મને ગર્વ છે.

હું સદાય તેને લાયક બનવા પ્રયત્ન કરીશ.

હું મારાં માતાપિતા, શિક્ષકો અને વડીલો પ્રત્યે આદર રાખીશ

અને દરેક જણ સાથે સભ્યતાથી વર્તીશ.

હું મારા દેશ અને દેશબાંધવોને મારી નિષ્ઠા અર્પું છું.

તેમનાં કલ્યાણ અને સમૃદ્ધિમાં જ મારું સુખ રહ્યું છે.

રાજ્ય સરકારની વિનામૂલ્યે યોજના હેઠળનું પુસ્તક



ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ
'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર-382010

© ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, ગાંધીનગર

આ પાઠ્યપુસ્તકના સર્વ હક ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળને હસ્તક છે.
આ પાઠ્યપુસ્તકનો કોઈ પણ ભાગ કોઈ પણ રૂપમાં ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળના નિયામકની લેખિત પરવાનગી વગર પ્રકાશિત કરી શકાશે નહિ.

વિષય-સલાહકાર

પ્રો. આર. પી. સોની

લેખન-સંપાદન

ડૉ. હર્ષલ આરોલકર (કન્વીનર)

ડૉ. સોનલ જૈન

ડૉ. નિલેષ મોદી

શ્રી તૃપ્તિબહેન ડોડીયા

અનુવાદ

શ્રી રોહિતભાઈ દોશી

શ્રી ગિરીશ બ્રહ્મભટ્ટ

શ્રી સાકેત દવે

સમીક્ષા

શ્રી બીમલભાઈ રાવલ

શ્રી પંકજકુમાર શુક્લ

શ્રી નિશીતાબહેન ગાંધી

શ્રી રાજેશ્રીબહેન પરિયા

ભાષાશુદ્ધિ

શ્રી ઓ. બી. દવે

સંયોજન

શ્રી આશિષ એચ. બોરીસાગર

(વિષય-સંયોજક : ગણિત)

નિર્માણ-આયોજન

શ્રી હરેન પી. શાહ

(નાયબ નિયામક : શૈક્ષણિક)

મુદ્રણ-આયોજન

શ્રી હરેશ એસ. લીખાચીયા

(નાયબ નિયામક : ઉત્પાદન)

પ્રસ્તાવના

માધ્યમિક અને ઉચ્ચતર માધ્યમિક શિક્ષણ બોર્ડે ઓપન સોર્સ ઓપરેટિંગ સિસ્ટમ અને તેને સુસંગત વિવિધ મુદ્દાઓ માટેના કમ્પ્યુટર-અધ્યયનને લગતા ઓપન સોર્સ સોફ્ટવેર આધારિત નવો અભ્યાસક્રમ તૈયાર કર્યો છે. આ અભ્યાસક્રમ ગુજરાત સરકાર દ્વારા મંજૂર કરવામાં આવ્યો છે.

ગુજરાત સરકાર દ્વારા મંજૂર થયેલા **ધોરણ 11**ના **કમ્પ્યુટર-અધ્યયન** વિષયના નવા અભ્યાસક્રમ અનુસાર તૈયાર કરવામાં આવેલું આ પાઠ્યપુસ્તક વિદ્યાર્થીઓ સમક્ષ મૂકતાં મંડળ આનંદ અનુભવે છે.

આ વિષયનું અંગ્રેજી માધ્યમનું પાઠ્યપુસ્તક પ્રસિદ્ધ કરતાં પહેલાં એની હસ્તપ્રતની આ સ્તરે શિક્ષણકાર્ય કરતા શિક્ષકો અને તજજ્ઞો દ્વારા સર્વાંગી સમીક્ષા કરાવવામાં આવી છે અને તેમનાં સૂચનો અનુસાર હસ્તપ્રતમાં યોગ્ય સુધારા-વધારા કર્યા પછી આ પાઠ્યપુસ્તક પ્રસિદ્ધ કરવામાં આવ્યું છે. અંગ્રેજી માધ્યમના પાઠ્યપુસ્તકનો આ ગુજરાતી અનુવાદ છે.

પ્રસ્તુત પાઠ્યપુસ્તકને વિષયવસ્તુલક્ષી, રસપ્રદ અને ક્ષતિરહિત બનાવવા માટે મંડળે પૂરતી કાળજી લીધી છે, તેમ છતાં શિક્ષણમાં રસ ધરાવનાર વ્યક્તિઓ પાસેથી પુસ્તકની ગુણવત્તા વધારે તેવાં સૂચનો આવકાર્ય છે.

પી. ભારતી (IAS)

નિયામક

તા. 04-11-2019

કાર્યવાહક પ્રમુખ

ગાંધીનગર

પ્રથમ આવૃત્તિ : 2013 પુન:મુદ્રણ : 2016, 2017, 2018, 2019, 2020

પ્રકાશક : ગુજરાત રાજ્ય શાળા પાઠ્યપુસ્તક મંડળ, 'વિદ્યાયન', સેક્ટર 10-એ, ગાંધીનગર વતી
પી. ભારતી (IAS), નિયામક

મુદ્રક :

મૂળભૂત ફરજો

ભારતના દરેક નાગરિકની ફરજ નીચે પ્રમાણે રહેશે :*

- (ક) સંવિધાનને વફાદાર રહેવાની અને તેના આદર્શો અને સંસ્થાઓનો, રાષ્ટ્રધ્વજનો અને રાષ્ટ્રગીતનો આદર કરવાની;
- (ખ) આઝાદી માટેની આપણી રાષ્ટ્રીય લડતને પ્રેરણા આપનારા ઉમદા આદર્શોને હૃદયમાં પ્રતિષ્ઠિત કરવાની અને અનુસરવાની;
- (ગ) ભારતનાં સાર્વભૌમત્વ, એકતા અને અખંડિતતાનું સમર્થન કરવાની અને તેમનું રક્ષણ કરવાની;
- (ઘ) દેશનું રક્ષણ કરવાની અને રાષ્ટ્રીય સેવા બજાવવાની હાકલ થતાં, તેમ કરવાની;
- (ચ) ધાર્મિક, ભાષાકીય, પ્રાદેશિક અથવા સાંપ્રદાયિક ભેદોથી પર રહીને, ભારતના તમામ લોકોમાં સુમેળ અને સમાન બંધુત્વની ભાવનાની વૃદ્ધિ કરવાની, સ્ત્રીઓના ગૌરવને અપમાનિત કરે તેવા વ્યવહારો ત્યજી દેવાની;
- (છ) આપણી સમન્વિત સંસ્કૃતિના સમૃદ્ધ વારસાનું મૂલ્ય સમજી તે જાળવી રાખવાની;
- (જ) જંગલો, તળાવો, નદીઓ અને વન્ય પશુપક્ષીઓ સહિત કુદરતી પર્યાવરણનું જતન કરવાની અને તેની સુધારણા કરવાની અને જીવો પ્રત્યે અનુકંપા રાખવાની;
- (ઝ) વૈજ્ઞાનિક માનસ, માનવતાવાદ અને જિજ્ઞાસા તથા સુધારણાની ભાવના કેળવવાની;
- (ટ) જાહેર મિલકતનું રક્ષણ કરવાની અને હિંસાનો ત્યાગ કરવાની;
- (ઠ) રાષ્ટ્ર પુરુષાર્થ અને સિદ્ધિનાં વધુ ને વધુ ઉન્નત સોપાનો ભણી સતત પ્રગતિ કરતું રહે એ માટે, વૈયક્તિક અને સામૂહિક પ્રવૃત્તિનાં તમામ ક્ષેત્રે શ્રેષ્ઠતા હાંસલ કરવાનો પ્રયત્ન કરવાની;
- (ડ) માતા-પિતાએ અથવા વાલીએ 6 વર્ષથી 14 વર્ષ સુધીની વયના પોતાના બાળક અથવા પાલ્યને શિક્ષણની તકો પૂરી પાડવાની.

અનુક્રમણિકા

1. મલ્ટિમીડિયાનો પરિચય	1
2. એનિમેશન ટૂલ : સીન્ડિંગ	20
3. સીન્ડિંગ વડે એનિમેશન	38
4. સ્તર (Layer) નો પરિચય	57
5. સીન્ડિંગમાં ચિત્રનો ઉપયોગ	73
6. ઉબન્ટુ લિનક્સના મૂળભૂત કમાન્ડ	99
7. વિમ એડિટર અને મૂળભૂત સ્ક્રિપ્ટિંગ	137
8. એડવાન્સ સ્ક્રિપ્ટિંગ	158
9. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનો પરિચય	184
10. ટેબલ્સ સાથે કાર્ય	208
11. ક્વેરીનો ઉપયોગ કરી માહિતી મેળવવી	233
12. ફોર્મ અને અહેવાલ	263
13. તાજેતરના પ્રવાહો અને ટેકનોલોજી	286
● પરિશિષ્ટ	313



આ પાઠ્યપુસ્તક વિશે...

પ્રિય શિક્ષકો,

એક વેગીલા રહે કમ્પ્યુટર-સાક્ષરતા ફેલાવવાના ધ્યેય સાથે, ગુજરાત સરકારે ICT@School કાર્યક્રમ હેઠળ સહાય મેળવતી 6000 કરતાં વધુ શાળાઓને અદ્યતન કમ્પ્યુટર સાધનસામગ્રી પૂરી પાડી છે. નવી નીતિની પહેલ તરીકે તમામ શાળાઓને ઉબન્ટુ (લિનક્સનું ભિન્ન સ્વરૂપ) ઓપરેટિંગ સિસ્ટમ અને અન્ય ઓપનસોર્સ સોફ્ટવેર પેકેજિસ આપવામાં આવ્યાં છે, જેથી શાળાઓ લાઇસન્સ જેવી સમસ્યાઓની ચિંતા કર્યા વિના કોઈ પણ જાતના દબાણ કે રોકટોક વિના સોફ્ટવેર વાપરી તેમજ બદલી પણ શકે. અગાઉનાં પાઠ્યપુસ્તકો મોટા ભાગે પ્રોપ્રાઇટરી સોફ્ટવેર પર આધારિત હોવાથી નવા અભ્યાસક્રમ આધારિત પાઠ્યપુસ્તકો લખવાની ફરી આવશ્યકતા જણાઈ. આ ઉપરાંત 8મું ધોરણ પ્રાથમિક વિભાગમાં સ્થાનાંતરિત કરવામાં આવ્યું હોવાથી પણ આ અનિવાર્ય હતું. આના પરિણામ રૂપે, કમ્પ્યુટર-અભ્યાસના વિવિધ વિષયો માટે ઓપનસોર્સ ઓપરેટિંગ સિસ્ટમ અને સુસંગત ઓપનસોર્સ સોફ્ટવેર ટૂલ્સ આધારિત નવું વિષયવસ્તુ 9થી 12 ધોરણ માટે તબક્કાવાર પૂરું પાડવામાં આવ્યું છે.

આ પાઠ્યપુસ્તક ધોરણ 11 માટે ‘કમ્પ્યુટર-અધ્યયન’ના વિષય માટેની શ્રેણીમાંનું ત્રીજું પાઠ્યપુસ્તક છે. આ પાઠ્યપુસ્તકનો ઉદ્દેશ નીચેના વિષયોથી પરિચિત થવાનો છે : મલ્ટિમીડિયાનું પ્રાથમિક જ્ઞાન, લિનક્સ ઉપરનું ઓપનસોર્સ મલ્ટિમીડિયા ટૂલ સીન્ડ્રિંગનો ઉપયોગ કરીને એનિમેશનનો પરિચય, વિવિધ કાર્યો માટે અનિવાર્ય ઉબન્ટુ લિનક્સના આદેશોની ચર્ચા, શેલ સ્ક્રિપ્ટિંગનો પરિચય, ઓપન ઓફિસ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનું ટૂલ બેઝ વડે ડેટાબેઝની પ્રક્રિયાઓ અને કેટલીક ઊભરી રહેલી ટેકનોલોજીનું વિહંગાવલોકન કે જે ભવિષ્યમાં ઘણી પ્રગતિ કરી શકે.

અમે આશા રાખીએ છીએ કે આવરી લીધેલો અભ્યાસક્રમ, વિદ્યાર્થીઓને કમ્પ્યુટર એપ્લિકેશન્સ અંગે સ્પષ્ટ સમજ મેળવવામાં ઉપયોગી બને અને તમે ઓપનસોર્સ સોફ્ટવેર ટૂલ્સનો ઉપયોગ કરીને શિક્ષણ આપવામાં અને પ્રાયોગિક કાર્યમાં આનંદ અનુભવો.

પ્રિય વિદ્યાર્થીઓ,

અમે એમ માની લઈએ છીએ કે તમે ઓપનસોર્સ ઓપરેટિંગ સિસ્ટમ ઉબન્ટુ લિનક્સ અને ઓપન ઓફિસના ઘટકોથી પરિચિત હશો. આ પાઠ્યપુસ્તકમાં તમે નીચેની બાબતોથી પરિચિત બનશો : મલ્ટિમીડિયાની મૂળભૂત બાબતો, લિનક્સમાં ફાઇલ માટેના પાયારૂપ આદેશો અને સામાન્ય પ્રક્રિયાઓ, એડિટરમાં કામ કરવાની પદ્ધતિ અને પ્રાથમિક સંચાલન માટે શેલસ્ક્રિપ્ટ. તમે ડેટાબેઝ કેવી રીતે બનાવવા, માહિતી મેળવવા માટે ક્વેરીઝ (Queries) કેવી રીતે લખવી તેમજ ફોર્મ્સ અને અહેવાલો તૈયાર કરવા બાબત પણ ભણશો. અંતમાં, કેટલીક ઊભરી રહેલી ટેકનોલોજી અને વલણોની ચર્ચા પણ કરવામાં આવશે, જેથી ઇન્ફર્મેશન ટેકનોલોજી નજીકના ભવિષ્યમાં કઈ દિશામાં લઈ જાય તેવી શક્યતા છે, તે બાબત પણ તમે પરિચિત બનો.

પ્રકરણ 1થી 5 નીચેની બાબતોને આવરી લે છે : મલ્ટિમીડિયાનો પરિચય, મલ્ટિમીડિયા ટૂલ સીન્ડ્રિંગની લાક્ષણિકતાઓ, સીન્ડ્રિંગ વાપરીને એનિમેશન બનાવવું, લેયર્સનો ઉપયોગ કરીને એનિમેશનનાં અદ્યતન ખ્યાલો સમજવા અને લેયર્સને એકબીજા ઉપર મૂકી શું હાંસલ કરી શકાય, સીન્ડ્રિંગ ટૂલમાં ચિત્રો અને ઇમેજિસનું કેવી રીતે સંચાલન કરી શકાય વગેરે. પ્રકરણ 6 સામાન્ય કાર્યો કરવા માટે મૂળભૂત લિનક્સ આદેશો; ફાઇલ અને ડિરેક્ટરીનો કાર્યક્રમ ઉપયોગ; ઇનપુટ/આઉટપુટ

રિડાઈરેક્શન અને પાઈપ; તેમજ રેકોર્ડ્સને સોર્ટ કરવા બાબતની માહિતીને આવરી લે છે. પ્રકરણ 7માં વિમ-એડિટરનો ઉપયોગ કરીને ફાઈલ બનાવવી તેમજ તેમાં સુધારા-વધારા કરવા અને શેલસ્ક્રિપ્ટિંગનો પરિચય આવરી લીધો છે. પ્રકરણ 8માં અનેક શેલસ્ક્રિપ્ટિંગનાં ઉદાહરણ આપ્યાં છે, જેથી પ્રાથમિક સંચાલનનાં કાર્ય કરવા માટે સ્ક્રિપ્ટ લખવાનો તમારો આત્મવિશ્વાસ વધે.

પ્રકરણ 9 અને 10 ઓપન ઓફિસ ટૂલબેઝનો ઉપયોગ કરીને ડેટાબેઝ (ટેબલ્સ) બનાવવા અને બનાવેલાં ટેબલ્સમાં ખરેખરી (વાસ્તવિક) માહિતી ભરવા બાબત પરિચય આપે છે. પ્રકરણ 11 અને 12 માહિતી પાછી મેળવવા માટે queries લખવી, ફોર્મની રચના કરવી તેમજ આઉટપુટ માટે રિપોર્ટ્સ બનાવવા બાબત ચર્ચા કરી છે. છેલ્લું પ્રકરણ 13 કેટલીક ઊભરી રહેલી ટેકનોલોજી અને વલણો જેમ કે મોબાઈલ કમ્પ્યૂટિંગ, ક્લાઉડ કમ્પ્યૂટિંગ, RFID, બાયોમેટ્રિક્સ અને બાકીના અન્યમાં ગ્રીન કમ્પ્યૂટિંગ વિશે છે. અમને ખાતરી છે કે કમ્પ્યૂટિંગનું વર્તમાન અને તેનાં ભાવિ વલણોની માહિતી તમારા વાચન અને શીખવાને રસપ્રદ બનાવશે.

અહીં એ અપેક્ષિત છે કે જો તમે પાઠ્યપુસ્તકનો અભ્યાસ અને પ્રાયોગિક કાર્યની કાળજીપૂર્વક પ્રેક્ટિસ કરશો, તો સીન્ડ્રિંગ એનિમેશન ટૂલ સાથે કામ કરવામાં, શેલસ્ક્રિપ્ટ લખવામાં, ડેટાબેઝની રચના કરવામાં ને ફોર્મ્સ બનાવવામાં, અહેવાલો તૈયાર કરવામાં અને માહિતી મેળવવા માટે queries લખવામાં તમારા આત્મવિશ્વાસનો વિકાસ જરૂર થશે.

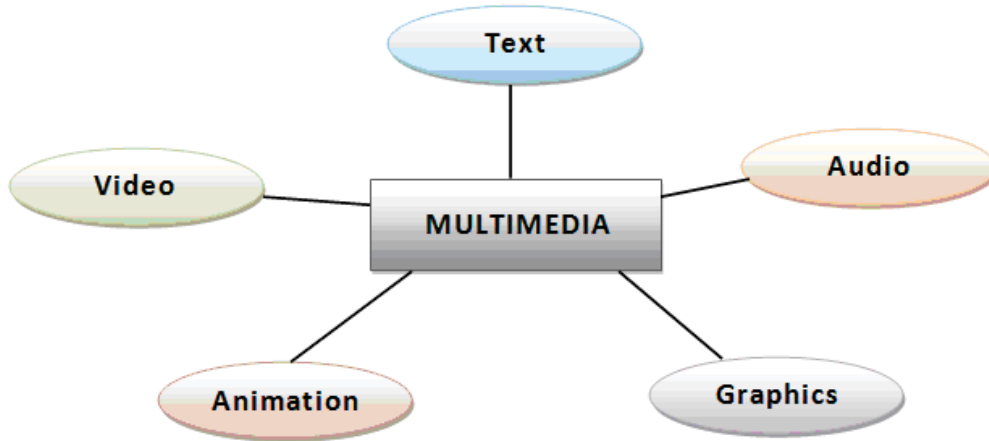




સંચારણ (Communication) આપણા જીવનનો અભિન્ન ભાગ છે. સંચારણ માટે આપણે રેડિયો, વર્તમાનપત્ર, ટેલીવિઝન, થિયેટર, મૂવી ફિલ્મો, ઇન્ટરનેટ અને અન્ય વિવિધ માધ્યમોનો ઉપયોગ કરીએ છીએ. આ તમામ માધ્યમો આપણને જ્ઞાનપ્રાપ્તિ કે શિક્ષણ મેળવવામાં મદદરૂપ બને છે. વિવિધ પ્રકારના દર્શકો સમક્ષ અર્થપૂર્ણ સંદેશ રજૂ કરવા માટે સંચારણનું એક જ માધ્યમ પૂરતું હોતું નથી. જેમકે, કોઈ મિત્ર સાથે ટેલીફોન પર વાત કરતી વખતે તેનો અવાજ સાંભળી શકાય છે, પરંતુ તેના ચહેરાના હાવભાવ જોઈ શકાતા નથી. જ્યારે તમે મિત્રને તમારી કાશ્મીરની સહેલગાહ વિશે માહિતી આપતો પત્ર લખો છો ત્યારે, માત્ર લખાણ મોકલી શકો છો, પરંતુ મિત્ર તમારો અવાજ સાંભળી શકતો નથી. જો તમે પત્ર સાથે ચિત્રો પણ મોકલશો તો તે સહેલગાહ દરમિયાન તમે અનુભવેલા આનંદની કલ્પના કરી શકશો. આમ છતાં, તમે વીડિયોક્લિપ મોકલશો, તો તે તમે અનુભવેલા આનંદને વધુ સારી રીતે કલ્પી શકશો. આમ, માહિતીનાં વધુ માધ્યમોનો ઉપયોગ કરવાથી સંચારણની અસરકારકતા વધારી શકાય છે. સંચારણને વધુ પ્રભાવશાળી બનાવવા માટે મલ્ટિમીડિયાનો ઉપયોગ કરવામાં આવે છે. આ પ્રકરણમાં આપણે મલ્ટિમીડિયા, તેના મૂળભૂત ઘટકો અને મલ્ટિમીડિયા વિનિયોગો વિશે અભ્યાસ કરીશું.

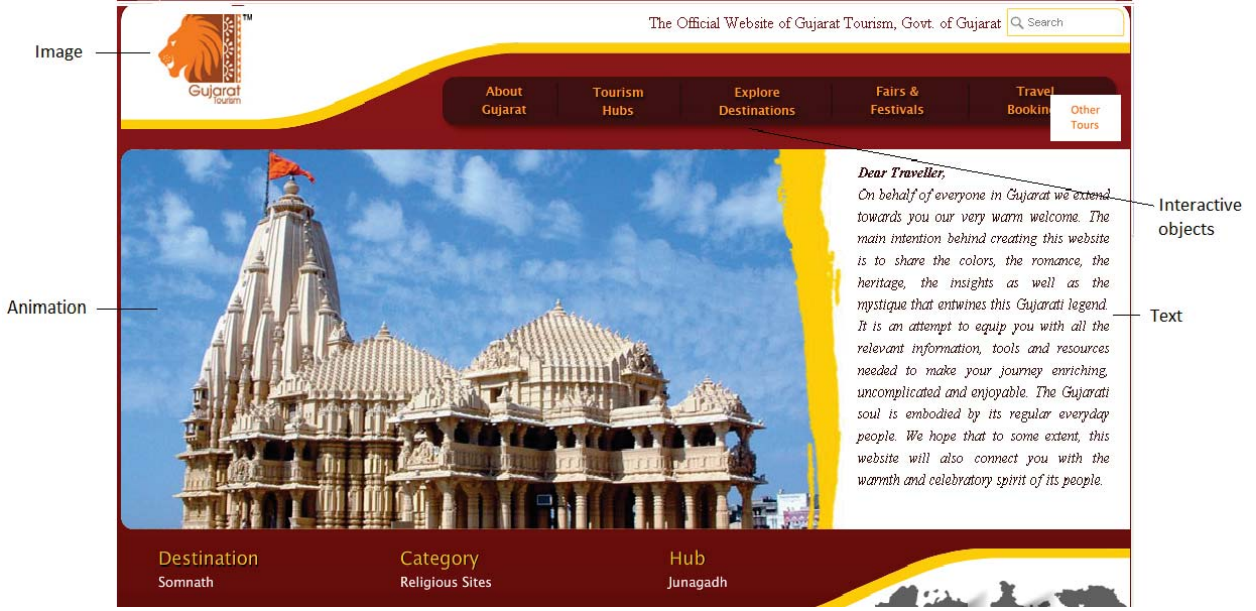
મલ્ટિમીડિયા (Multimedia)

મલ્ટિમીડિયા શબ્દ બે શબ્દો મળીને બન્યો છે : મલ્ટિપલ (multiple) અને મીડિયા (media). માટે જ, સંચારણ માટે એકથી વધુ (મલ્ટિપલ) માધ્યમોનો ઉપયોગ એટલે મલ્ટિમીડિયા. બીજા શબ્દોમાં, મલ્ટિમીડિયા એટલે લખાણ (text), ચિત્રો (graphics), વીડિયો અને એનિમેશન જેવાં વિવિધ માધ્યમોનું એકત્રીકરણ. આ તમામ માધ્યમોને એકત્રિત કરવાથી ઉપયોગકર્તા સમક્ષ રજૂ કરવામાં આવનાર સંચારણને વધુ સુગ્રથિત (structured) અને સમજણપૂર્વકનું (understandable) બનાવી શકાય છે. હાલમાં શિક્ષણ, રંગમંચ, જાહેરાત, શૈલી (ફેશન), રમતો વગેરેનાં ક્ષેત્રોમાં મલ્ટિમીડિયાનો ઉપયોગ થઈ રહ્યો છે. આકૃતિ 1.1 મલ્ટિમીડિયાનાં જુદાં-જુદાં ઘટકો દર્શાવે છે.



આકૃતિ 1.1 : મલ્ટિમીડિયાનાં ઘટકો

આ માધ્યમોનાં ઘટકોનો વિકાસ કરવા માટે કમ્પ્યુટર અગત્યનો ભાગ ભજવે છે. લખાણ, ચિત્રો, ધ્વનિ, વીડિયો અને એનિમેશનને ચલાવી શકતાં કમ્પ્યુટરો કેટલીક વાર મલ્ટિમીડિયા કમ્પ્યુટરના નામથી પણ ઓળખાય છે. મલ્ટિમીડિયાનાં જુદાં-જુદાં ઘટકોનો ઉપયોગ આકૃતિ 1.2માં દર્શાવ્યો છે.



આકૃતિ 1.2 : મલ્ટિમીડિયાનાં ઘટકોનો ઉપયોગ

મલ્ટિમીડિયાનાં ઘટકો

મલ્ટિમીડિયાને લખાણ, ધ્વનિ, ચિત્રો, વીડિયો અને એનિમેશન જેવા વિવિધ ઘટકોના સમૂહ તરીકે વ્યાખ્યાયિત કરી શકાય, તે આપણે જોયું. હવે, આ દરેક ઘટક વિશે માહિતી મેળવીએ.

લખાણ (Text)

કોઈ પણ વિષયની રજૂઆત માટે તેમાં લખાણ ઉમેરવાની ક્રિયા એ મલ્ટિમીડિયાનું એક પાયારૂપ પગલું છે. ઉપયોગકર્તાને માહિતી પૂરી પાડવા માટે લખાણનો ઉપયોગ કરવામાં આવે છે. મલ્ટિમીડિયા રજૂઆતમાં લખાણ અને શબ્દોનો યોગ્ય ઉપયોગ ઉપયોગકર્તા સુધી યોજનાઓ અને સંદેશા પહોંચાડવા માટે ઘણો મદદરૂપ બને છે. શબ્દ, લીટી કે ફકરા એમ કોઈ પણ સ્વરૂપે લખાણ હોઈ શકે છે. મલ્ટિમીડિયા માટે લખાણ સ્વરૂપની વિગતો દર્શાવવા માટે કોઈ પણ ટેક્સ્ટ એડિટરનો ઉપયોગ કરી શકાય છે. જોકે, તેમાં વિશિષ્ટ અસર મેળવવા ગ્રાફિક સોફ્ટવેર જરૂરી છે. વ્યાવસાયિક જરૂરિયાત માટે મલ્ટિમીડિયા સોફ્ટવેર દ્વારા લખાણના જુદા-જુદા પ્રકાર, કદ, રંગ અને શૈલી મેળવી શકાય છે.

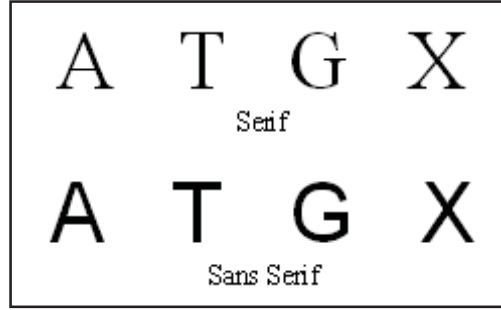
આકૃતિ 1.3માં દર્શાવેલા નમૂના દેખાવમાં સમાન નથી તેની નોંધ કરો કારણકે તેમને જુદા-જુદા પ્રકારનાં ટાઈપફેસ (typeface), ફોન્ટ (font) અને કદ (size) આપવામાં આવ્યાં છે.



આકૃતિ 1.3 : નમૂનારૂપ લખાણ

એકસમાન દેખાવ ધરાવતા અક્ષરોનો સમૂહ ટાઈપફેસ તરીકે ઓળખાય છે. ઉદાહરણ તરીકે Times, Arial, Courier અને અન્ય ટાઈપફેસ 'ફોન્ટ' (Font) નામે ઓળખાતી જુદી-જુદી શૈલીઓ અને કદ ધરાવે છે. શૈલીમાં ત્રાંસા અક્ષરો (italic), ઘાટા અક્ષરો (bold), નીચે લીટી ધરાવતા અક્ષરો (underline) વગેરેનો સમાવેશ થાય છે. કદ એટલે અક્ષરના ઉપરના ભાગથી નીચેના ભાગ જેટલું અંતર. તેને પોઈન્ટ (point) એકમ દ્વારા રજૂ કરવામાં આવે છે. જેમકે, 12 પોઈન્ટ, 20 પોઈન્ટ વગેરે. ઉદાહરણ તરીકે, જો Times new roman 14 point અક્ષરોનો ઉપયોગ કરવામાં આવે, તો તેમાં Times new roman શૈલી છે અને 14 point તેનું કદ છે.


ટાઈપફેસને બે વર્ગમાં વહેંચી શકાય : સેરિફ (Serif) અને સાન્સ સેરિફ (Sans Serif). સેરિફ પ્રકારના ફોન્ટમાં અક્ષરના છેડા પર થોડું સુશોભન કરવામાં આવેલું હોય છે. Times, Century, Bookman વગેરે સેરિફ ફોન્ટનાં ઉદાહરણ છે. લીટીમાં આવેલ લખાણને વધુ સરળતાપૂર્વક વાંચી શકાતા હોવાને કારણે સેરિફ ફોન્ટનો ઉપયોગ છાપેલાં પાનાં માટે વધુ કરવામાં આવે છે. સામાન્ય રીતે મોટા પ્રમાણમાં લખાણનું વાંચન કરવાનું હોય ત્યારે તેનો ઉપયોગ કરવામાં આવે છે. બીજી તરફ, નામ સૂચવે છે તે મુજબ સાન્સ સેરિફ (ફિન્ય ભાષામાં sansનો અર્થ ‘વગર’ થાય છે) સુશોભન વગરના ફોન્ટ છે. Arial, Verdana અને Helvetica સાન્સ સેરિફ પ્રકારના ફોન્ટનાં ઉદાહરણ છે. તેનો ઉપયોગ સામાન્ય રીતે મથાળાં કે શીર્ષક માટે કરવામાં આવે છે. પરંતુ કમ્પ્યુટરના સ્ક્રીન પર ઓછા રેઝોલ્યુશન (resolution)ને કારણે સેરિફ ફોન્ટને વાંચવા સાન્સ સેરિફ કરતાં મુશ્કેલ હોય છે. આકૃતિ 1.4માં ટાઈપફેસનાં ઉદાહરણ દર્શાવેલાં છે.

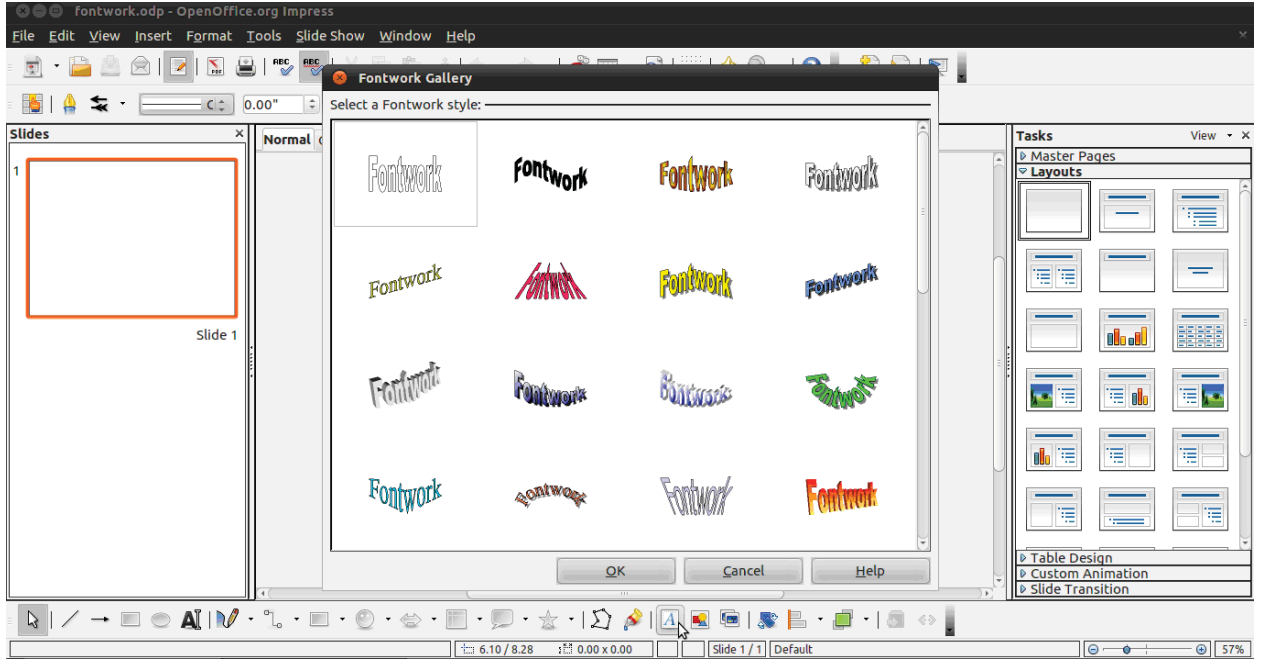


આકૃતિ 1.4 : ટાઈપફેસનાં ઉદાહરણ

કાગળ અને પેનની પરંપરાગત પદ્ધતિ કરતાં મલ્ટિમીડિયામાં ઉપયોગમાં લેવામાં આવતું લખાણ અલગ છે. પરંપરાગત રીતમાં લખાણનો ઉપયોગ કરીને માહિતી અભિવ્યક્ત કરવામાં આવે છે, જ્યારે મલ્ટિમીડિયામાં લખાણ સાથે અન્ય ઘટકોનો સમાવેશ પણ થઈ શકે છે. આમ, મલ્ટિમીડિયામાં માહિતીની રજૂઆત ઉપયોગકર્તા સમક્ષ એટલી સરળ હોવી જોઈએ કે જેથી તે ગૂંચવાઈ ન જાય. ઉપયોગકર્તાનું ધ્યાન દોરવા માટે મલ્ટિમીડિયામાં આવેલ લખાણ સંક્ષિપ્ત, મુદ્દાસર અને આકર્ષક હોવું જોઈએ.

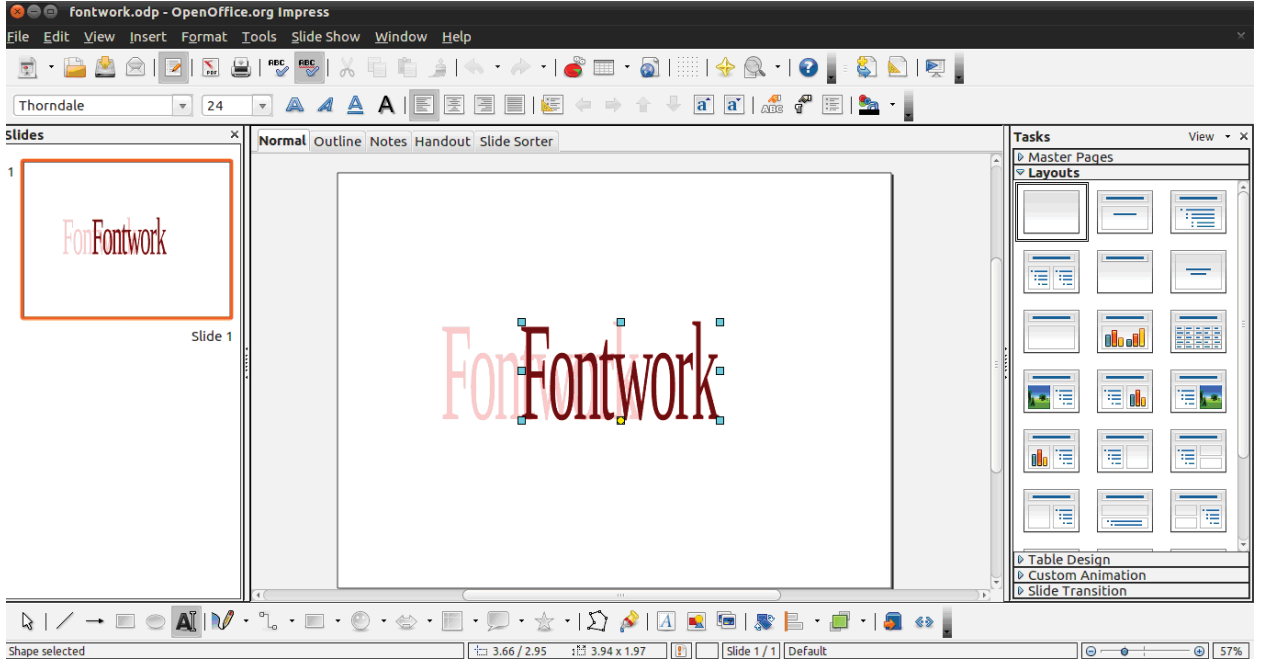
ધારોકે, આપણે ઓપન ઓફિસ ઈમ્પ્રેસનો ઉપયોગ કરી “Gujarat Tourism” માટે એક રજૂઆત તૈયાર કરવી છે. સૌપ્રથમ આ માટે Gujarat Tourismના શીર્ષક સાથે મુખ્ય પાનું બનાવીશું. આ માટે આપણે સામાન્ય લખાણ ટાઈપ કરી શકીએ, પરંતુ લખાણને આકર્ષક સ્વરૂપે રજૂ કરવા ફોન્ટવર્ક (Fontwork)નો ઉપયોગ કરી શકાય. લખાણની રચના કરવા માટેનાં પગલાં નીચે જણાવ્યાં છે :

- ઓપન ઓફિસ ઈમ્પ્રેસમાં નવી (કોરી) રજૂઆત ખોલો.
- View → Toolbars → Drawing પર ક્લિક કરો. આમ કરવાથી ડ્રોઈંગ ટૂલબાર રજૂ કરવામાં આવશે.
- ડ્રોઈંગ ટૂલબાર પર Fontwork Gallery ના icon  પર ક્લિક કરો. આમ કરવાથી આકૃતિ 1.5માં દર્શાવ્યા મુજબ Fontwork Gallery ડાયલોગબોક્સ ખૂલશે.



આકૃતિ 1.5 : Fontwork Gallery ડાયલોગબોક્સ

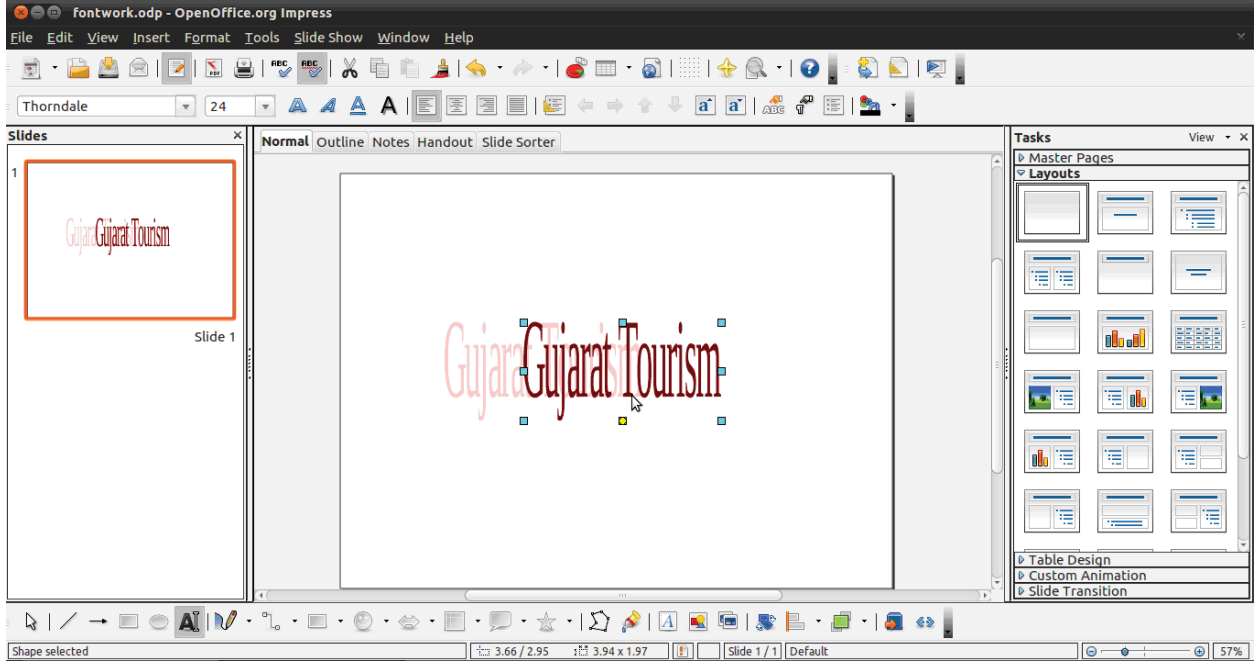
- Fontwork Gallery ડાયલોગબોક્સમાં આવેલી કોઈ પણ શૈલી પસંદ કરી OK બટન પર ક્લિક કરો. આકૃતિ 1.6માં દર્શાવ્યા મુજબ ફોન્ટવર્ક ઓબ્જેક્ટ દસ્તાવેજમાં ઉમેરવામાં આવશે.
- લખાણ સુધારવા માટે ઓબ્જેક્ટ પર ડબલ ક્લિક કરો.



આકૃતિ 1.6 : ફોન્ટવર્ક લખાણમાં સુધારો

- પૂર્વનિર્ધારિત લખાણને “Gujarat Tourism” લખાણ સાથે બદલો.
- લખાણના સુધારા પૂરા કરવા માટે ESC કી દબાવો.

- હવે આકૃતિ 1.7માં દર્શાવ્યા મુજબ “Gujarat Tourism” લખાણ સ્લાઇડમાં જોવા મળશે.
- હવે પછીના ઉપયોગ માટે ફાઇલનો સંગ્રહ કરો.



આકૃતિ 1.7 લખાણ સાથેની તૈયાર સ્લાઇડ

ધ્વનિ (Audio)

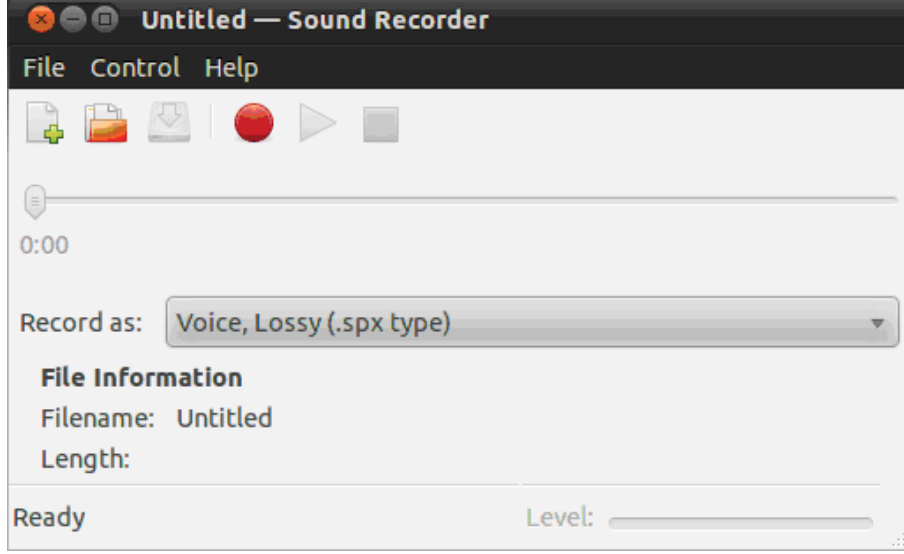
અવાજ (sound) એ મલ્ટિમીડિયાનો સૌથી મહત્વનો ઘટક છે. ધ્વનિ (audio) એટલે કોઈ પણ ભાષાની અર્થપૂર્ણ બોલી (speech). અવાજના એનેલોગ સ્વરૂપને સાઉન્ડ (sound) તરીકે તથા તેના ડિજિટલ સ્વરૂપને ઓડિયો (audio) તરીકે ઓળખવામાં આવે છે.

પ્રેક્ષકો સમક્ષ હાજર રહીને રજૂઆત કરવાની હોય ત્યારે ધ્વનિ વગરનાં લખાણ અને ચિત્રો મદદરૂપ બને છે. પરંતુ પ્રસ્તુતકર્તાની ગેરહાજરીમાં ધ્વનિ મહત્વનો ભાગ ભજવે છે. કોઈ પણ શૈક્ષણિક સીડી આ માટેનું ઉત્તમ ઉદાહરણ છે. ઉપયોગકર્તા ધ્વનિને ચાલુ કે બંધ રાખી શકે છે, પરંતુ શિક્ષણ-પ્રક્રિયા ધ્વનિની હાજરીમાં વધુ ઉપભોગ્ય બને છે. આ પ્રકારની મલ્ટિમીડિયા રજૂઆત બનાવવા માટે સૌપ્રથમ ધ્વનિનું મુદ્રણ કરવું પડે છે અને તેને રજૂઆત સમયે એનિમેશન મુજબ વગાડવું પડે છે. આમ, એનિમેશનને ધ્વનિ સાથે જોડવાથી રજૂઆતને વધુ સરળતાથી સમજી શકાય છે.


હાલના કમ્પ્યુટર મલ્ટિમીડિયા વિનિયોગને સમર્થન આપે છે. તે મલ્ટિમીડિયા માટે જરૂરી હાર્ડવેર અને સોફ્ટવેર સાથે ઉપલબ્ધ બને છે. મલ્ટિમીડિયા માટે ઉપયોગી એવા સાઉન્ડકાર્ડ, સ્પીકર, માઈક્રોફોન અને વેબકેમેરા તેમાં પૂર્વસ્થાપિત હોય છે. ઓપરેટિંગ સિસ્ટમ દ્વારા પૂરા પાડવામાં આવેલા વિનિયોગ દ્વારા ઓડિયો કે વીડિયો કમ્પ્યુટરમાં સીધા જ ઉમેરી શકાય છે.

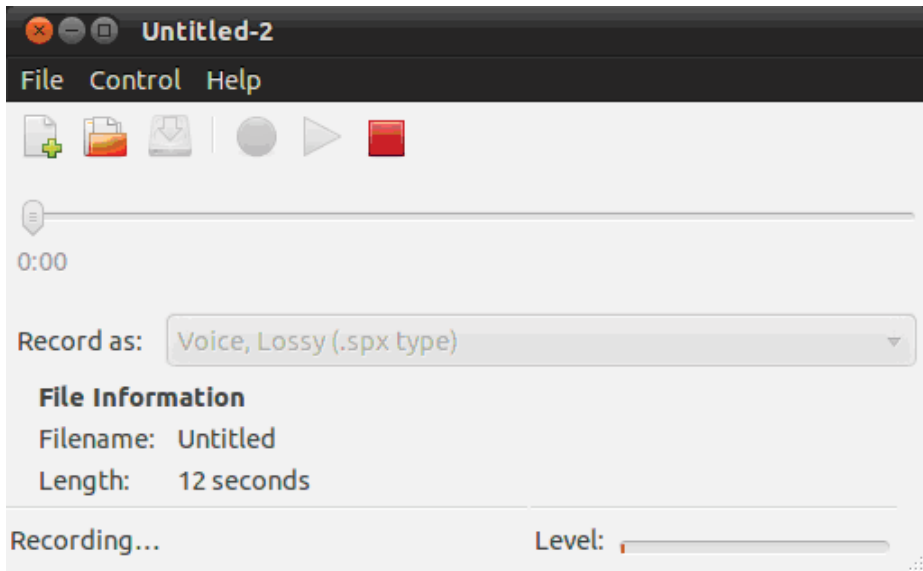
હવે આપણે ઉબન્ટુ લિનક્સ દ્વારા ઉપલબ્ધ એવા એક ધ્વનિમુદ્રણ માટેના વિનિયોગ (audio recording application)નો ઉપયોગ કરીએ. કમ્પ્યુટર સાથે માઈક્રોફોન અને સ્પીકર જોડાયેલાં હોય, તેની ખાતરી કરી લો. નીચે દર્શાવેલ પગલાં દ્વારા વિનિયોગનો ઉપયોગ કરી ધ્વનિને મુદ્રિત કરીએ :

- Applications → Sound and Video → Sound Recorder પસંદ કરો. આમ કરવાથી આકૃતિ 1.8માં દર્શાવ્યા મુજબ *Sound Recorder* ખૂલશે.
- File → New પસંદ કરો.
- Record as ડ્રોપડાઉન યાદીમાંથી નીચે દર્શાવેલ વિકલ્પોમાંથી કોઈ એકની પસંદગી કરો.
 - CD quality, Lossless
 - CD quality, Lossy
 - Voice



આકૃતિ 1.8 : સાઉન્ડ રેકોર્ડર

- રેકોર્ડિંગ શરૂ કરવા માટે આકૃતિ 1.8માં દર્શાવ્યા મુજબ Control → Record પસંદ કરો અથવા Record બટન  પર ક્લિક કરો.
- હવે, ધ્વનિ રેકોર્ડ કરવાનું શરૂ કરો. ઉદાહરણ તરીકે, “Gujarat Tourism”ની રજૂઆત માટે ગુજરાતના કોઈ પ્રવાસનસ્થળ અંગે માહિતી રેકોર્ડ કરો. આકૃતિ 1.9માં ધ્વનિમુદ્રણ માટેની પ્રક્રિયા દર્શાવવામાં આવી છે.



આકૃતિ 1.9 : ધ્વનિમુદ્રણ (Audio Recording)

- ધ્વનિનું રેકોર્ડિંગ પૂરું કરવા વિકલ્પ Control → Stop પસંદ કરો અથવા Stop (■) બટન પર ક્લિક કરો.
- રેકોર્ડિંગ સાંભળવા માટે Control → Play પસંદ કરો અથવા Play (▶) બટન પર ક્લિક કરો.
- ધ્વનિ-મિશ્રણ (audio mixer) માટે File → Open Volume Control પસંદ કરો.
- રેકોર્ડિંગનો સંગ્રહ કરવા માટે File → Save As પસંદ કરી ધ્વનિફાઇલનું નામ ટાઇપ કરો.

આ ધ્વનિફાઇલમાં ફેરફાર કરવા પણ શક્ય છે. જેમકે, કોઈ ભાગ દૂર કરવો, ધ્વનિ વાગવાની ગતિ બદલવી, તેની તીવ્રતા બદલવી, પડઘા (echo) ઉમેરવા વગેરે. ધ્વનિ અને વીડિયો જેવા મલ્ટિમીડિયા ઘટકોનો સંગ્રહ મીડિયાફાઇલમાં કરવામાં આવે છે. મીડિયાફાઇલનો પ્રકાર જાણવા માટે તેના અનુલંબન (એક્સટેન્શન) નામનો ઉપયોગ કરવામાં આવે છે. સામાન્ય રીતે ઉપયોગમાં લેવાતાં ધ્વનિસ્વરૂપો વિશે કોષ્ટક 1.1માં માહિતી આપવામાં આવી છે.

ફાઇલ એક્સટેન્શન	ફાઇલનો પ્રકાર	સમજૂતી
.mid, .midi	MIDI ફાઇલ	MIDI (Musical Instrument Digital Interface) ફાઇલ સંગીત પ્રકારની વિગતોનો સંગ્રહ કરે છે.
.rm, .ram	Real Audio ફાઇલ	.ram (Real Audio Metadata) એ ધ્વનિ અને વીડિયોફાઇલનું સંયોજન છે.
.wav	Wave ફાઇલ	વેવસ્વરૂપમાં ધ્વનિફાઇલ
.wma	Windows media audio ફાઇલ	વિન્ડોઝ મીડિયા કોમ્પ્રેશનથી સંકુચિત કરવામાં આવેલી ધ્વનિફાઇલ.
.mp3, .mpga	MP3 Audio ફાઇલ	સંકુચિત ધ્વનિસ્વરૂપ

કોષ્ટક 1.1 : સામાન્ય રીતે ઉપયોગમાં લેવામાં આવતાં ધ્વનિફાઇલ સ્વરૂપો

ઇન્ટરનેટ પર સૌથી પ્રચલિત સંકોચનરહિત (uncompressed) ફાઇલસ્વરૂપ .wav છે અને તેને તમામ પ્રચલિત બ્રાઉઝર દ્વારા સમર્થન આપવામાં આવે છે. મુદ્રિત સંગીત માટે .mp3 એ નવું સંકુચિત સ્વરૂપ છે. જો તમે મુદ્રિત સંગીતનો ઉપયોગ કરવા માંગતા હો, તો mp3 એ શ્રેષ્ઠ પસંદગી છે.

ચિત્ર (Image)

સામાન્ય રીતે મલ્ટિમીડિયા રજૂઆતો ચિત્ર (graphics/image) આધારિત હોય છે. ચિત્ર દ્વારા રજૂ કરવામાં આવેલી માહિતી યાદ રાખવી અને સમજવી વધુ સરળ હોય છે. મલ્ટિમીડિયા વિનિયોગોમાં ઉપયોગમાં લેવામાં આવતાં ચિત્રો સ્કેનરની મદદથી ડિજિટલ સ્વરૂપમાં ફેરવવામાં આવેલી તસવીરો હોઈ શકે છે અથવા તો તેને કમ્પ્યુટર દ્વારા જ બનાવવામાં આવે છે. ચિત્રનો સંગ્રહ કરવા માટે કમ્પ્યુટર બીટમેપ (bitmap) કે રાસ્ટર (raster) ચિત્ર નામે ઓળખાતા પિક્સેલ-નકશા (pixel map)નો ઉપયોગ કરે છે. Pixel (picture element)નું ટૂંકું સ્વરૂપ એ સ્ક્રીન પર આવેલું એક નાનું ટપકું છે. નકશો (map) એ આ ટપકાંઓનો દ્વિપરિમાણીય શ્રેણિક છે. આમ, બીટમેપ એ નાનાં-નાનાં ટપકાંઓનો એક સરળ શ્રેણિક છે, જે કમ્પ્યુટરના સ્ક્રીન પર ચિત્રની રચના કરે છે. દરેક પિક્સેલ તેમાં આવેલા રંગની કિંમત દર્શાવે છે. જ્યારે ચિત્રને સ્ક્રીન પર દર્શાવવા મોકલવામાં આવે છે, ત્યારે આ ચિત્ર-વિગતોનું પિક્સેલમાં રૂપાંતર કરવામાં આવે છે.

કમ્પ્યુટર દ્વારા બનાવવામાં આવતા ચિત્રના અન્ય એક સ્વરૂપને સદિશ ચિત્ર (vector graphic)ના નામથી ઓળખવામાં આવે છે. આ પદ્ધતિ ચિત્રનો ગાણિતિક સમીકરણોના સ્વરૂપમાં સંગ્રહ કરે છે. સ્ક્રીન પર બીટમેપ અને સદિશ બંને પ્રકારનાં

ચિત્રો એકસમાન દેખાય છે. પરંતુ જ્યારે બીટમેપ ચિત્રને વિસ્તૃત કરવામાં આવે, ત્યારે તે આકૃતિ 1.10માં દર્શાવ્યા મુજબ ઝાંખું (blur) થઈ જાય છે અને તેમાં આવેલાં પિક્સેલ વિસ્તૃત બને છે. બીજી તરફ, સદિશ ચિત્રો વિસ્તૃત કર્યા બાદ પણ પહેલાનાં જેવાં જ દેખાય છે. કારણ કે તેમાં વિગતોનો ગાણિતિક સમીકરણોના સ્વરૂપમાં સંગ્રહ કરવામાં આવ્યો હોય છે. જ્યારે-જ્યારે ચિત્રને સ્ક્રીન પર દર્શાવવામાં આવે, ત્યારે સમીકરણોની કિંમતની પુનઃગણતરી કરવામાં આવે છે.



બીટમેપ ચિત્ર



પિક્સેલ દર્શાવતું વિસ્તૃત બીટમેપ ચિત્ર

આકૃતિ 1.10 : બીટમેપ ચિત્ર

બન્ને પ્રકારનાં ચિત્રો કોષ્ટક 1.2માં જણાવ્યા મુજબનાં વિવિધ ફાઇલસ્વરૂપો દ્વારા સાચવી શકાય છે. પરંપરાગત રીતે, મેમરીની જગ્યાનો બચાવ કરવા માટે આ પ્રકારની ફાઇલોને સંકુચિત કરવામાં આવે છે. gif, jpeg અને png જેવાં બીટમેપ ફાઇલસ્વરૂપો સ્વયં-સંકોચનની પદ્ધતિનો ઉપયોગ કરે છે. કોષ્ટક 1.2માં ચિત્ર માટે ઉપયોગમાં લેવાતાં સામાન્ય સ્વરૂપો દર્શાવ્યાં છે.

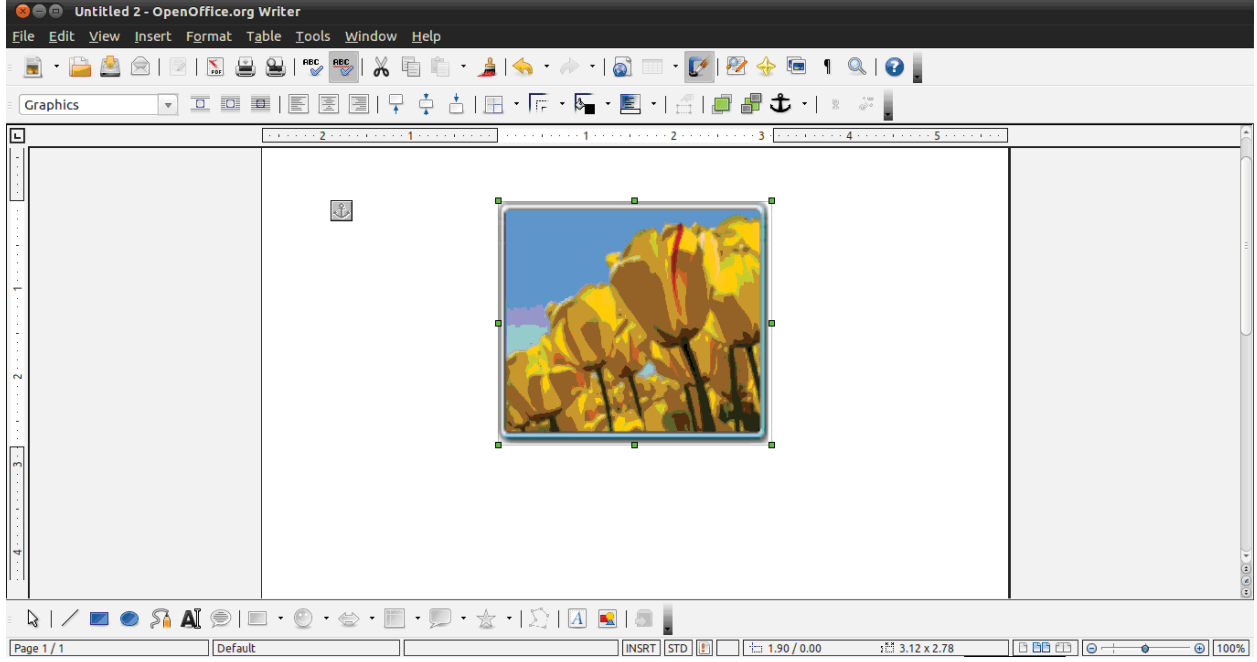
ફાઇલ એક્સ્ટેન્શન	ફાઇલનો પ્રકાર	સમજૂતી
.bmp	Bitmap ફાઇલ	બીટમેપ ડિજિટલ ચિત્રનો સંગ્રહ કરવા માટેની સંકોચનરહિત ચિત્રફાઇલ.
.gif	Graphical Interchange Format ફાઇલ	લખાણ સાથે નાનાં ચિત્રો તથા વેબચિત્રો માટે સામાન્ય મર્યાદિત રંગોનો ઉપયોગ.
.png	Portable Network Graphic ફાઇલ	વેબચિત્રોનો સંગ્રહ કરવા માટેનું સ્વરૂપ અને 32 બીટ રંગોનું સમર્થન. તે gifનું સુધારેલું સ્વરૂપ છે.
.jpeg / .jpg	Joint Photographic Expert Group ચિત્ર ફાઇલ	ડિજિટલ કેમેરા દ્વારા ઉપયોગમાં લેવામાં આવતું સામાન્ય ચિત્રફાઇલ સ્વરૂપ. 24-bit રંગો સુધીનું સમર્થન.
.psd	Photoshop document	એડોબ ફોટોશોપ દ્વારા બનાવવામાં આવેલી ચિત્રફાઇલ
.tif	Tagged Image File સ્વરૂપ	અત્યંત લવચીક (flexible), પ્લેટફોર્મથી સ્વતંત્ર (Platform independent) અને હાલમાં બહોળા પ્રમાણમાં ઉપયોગમાં લેવાતું સ્વરૂપ

કોષ્ટક 1.2 સામાન્ય રીતે ઉપયોગમાં લેવામાં આવતાં ચિત્ર ફાઇલસ્વરૂપો

સામાન્ય રીતે રજૂઆત બનાવવા માટે પૂર્વસ્થાપિત ચિત્રોનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે ઓપન ઓફિસ રાઈટરમાં Insert → Picture વિકલ્પ દ્વારા ચિત્ર ઉમેરી શકાય છે. ઉપયોગમાં લેવામાં આવતું ચિત્ર પહેલેથી કમ્પ્યૂટરમાં

સંગૃહીત થયેલું હોઈ શકે છે અથવા તો સ્કેનરનો ઉપયોગ કરી, ચિત્રને સ્કેન કરી ડિજિટલ સ્વરૂપમાં પરિવર્તિત કરવામાં આવે છે, જેને કમ્પ્યુટર દ્વારા ઉપયોગમાં લેવામાં આવે છે. આકૃતિ 1.11માં ઓપન ઓફિસ રાઈટર દ્વારા ઉમેરવામાં આવેલ ચિત્ર ધરાવતો દસ્તાવેજ દર્શાવ્યો છે.

આકૃતિ 1.11માં ઉપયોગમાં લેવામાં આવેલું ચિત્ર પહેલેથી આપણી પાસે ઉપલબ્ધ હતું. આપણી પસંદગી મુજબનું કોઈ પણ ચિત્ર આ રીતે ઉપયોગમાં લઈ શકાય છે. ચિત્રમાં સુધારા કરવા એડોબ ફોટોશોપ, GIMP, માઈક્રોસોફ્ટ પેઈન્ટ અને અન્ય પ્રોગ્રામનો ઉપયોગ પણ કરી શકાય છે.



આકૃતિ 1.11 : ચિત્ર સાથેનો દસ્તાવેજ

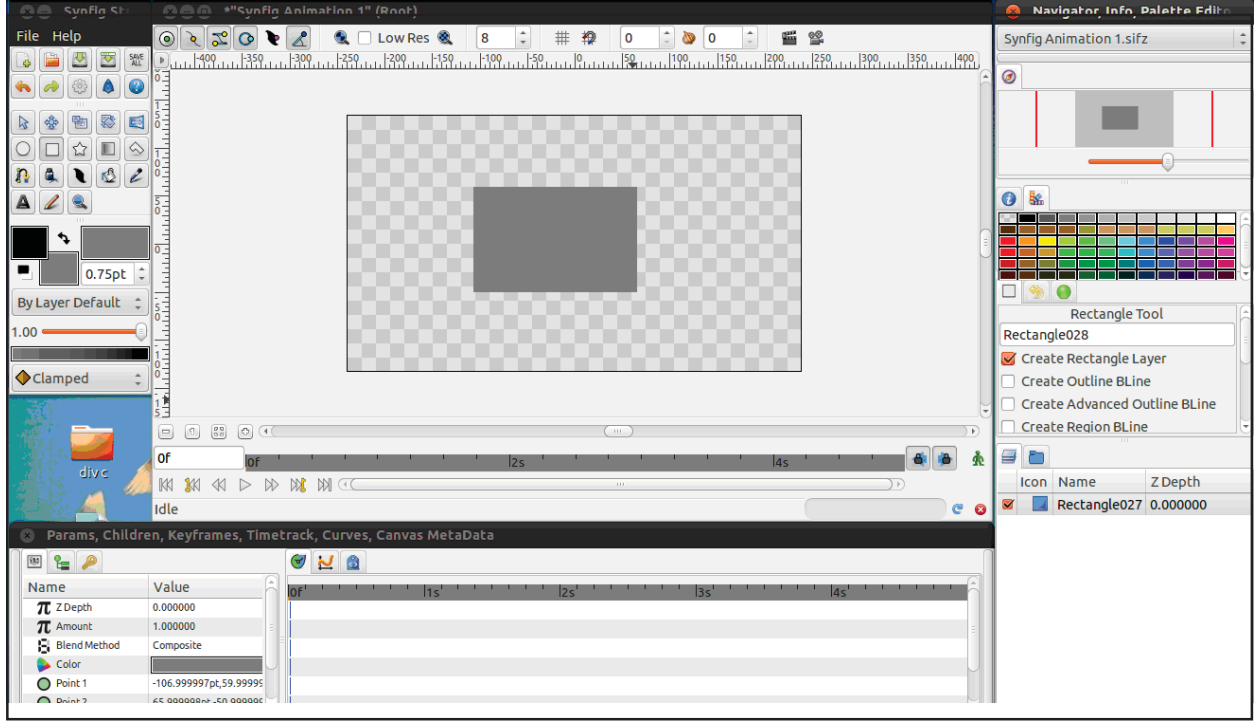
વીડિયો અને એનિમેશન (Video and Animation)

એનિમેશન દ્વારા કોઈ પણ મલ્ટિમીડિયા વિનિયોગને અત્યંત સક્ષમ બનાવી શકાય છે. વીડિયો કે એનિમેશન ઉમેરવાથી કોઈ પણ સ્થિર રજૂઆતને ગતિશીલ બનાવી શકાય છે. સૌપ્રથમ એનિમેશન અને વીડિયો વચ્ચેનો તફાવત સમજાવે. એનેલોગ કે ડિજિટલ પ્રકારના વીડિયો લઈ શકાય તેવા સાધન (video capturing device) દ્વારા મેળવવામાં આવેલાં કુદરતી દૃશ્યોની શ્રેણીને 'વીડિયો' કહે છે. આ સાધન વેબકેમેરા, ડિજિટલ કેમેરા કે પછી મોબાઈલ ફોન પણ હોઈ શકે છે.

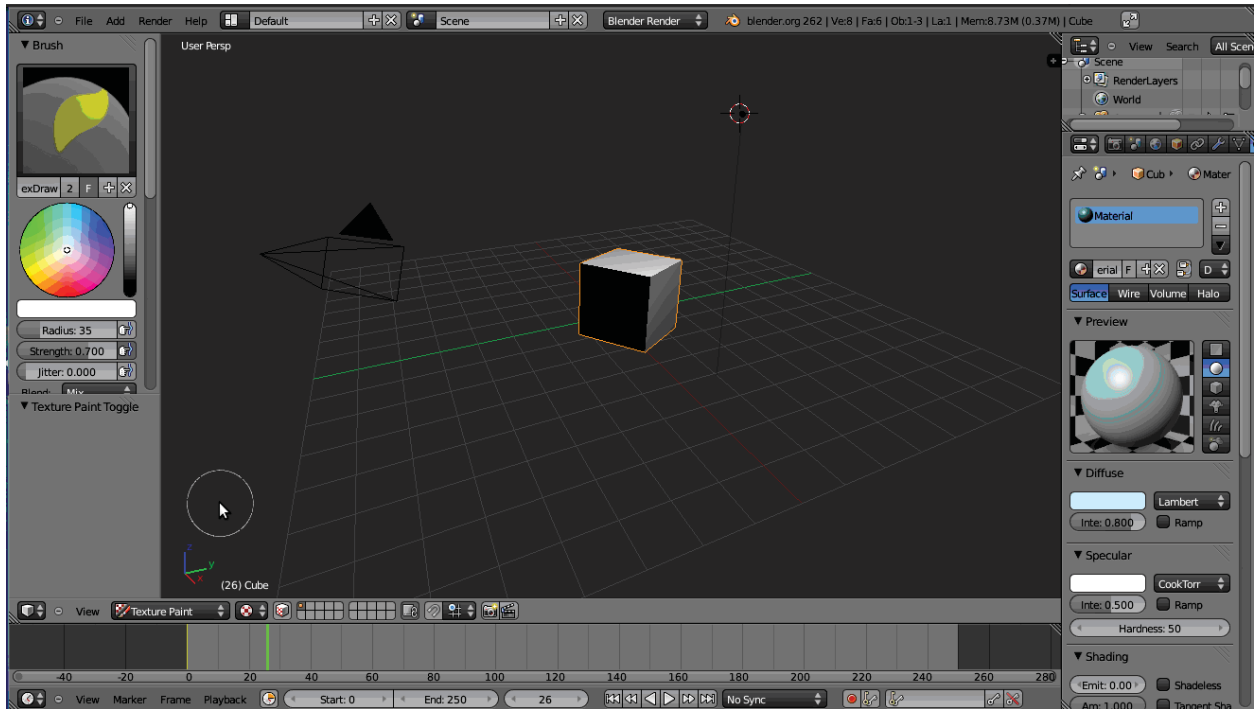
સમયની ગતિ પ્રમાણે બદલાતાં ચિત્રોને એનિમેશન કહે છે. આ માટે ગતિમાન અસરો દર્શાવવા ડિજિટલ ચિત્રોને એક પછી એક દર્શાવવામાં આવે છે. એમ કહી શકાય કે કેટલીક ચિત્રિત આકૃતિઓ દ્વારા એનિમેશનની રચના કરવામાં આવે છે. જ્યારે વીડિયોની રચના માટે પ્રત્યક્ષ સમયનાં દૃશ્યો (real time visuals)નો ઉપયોગ કરવામાં આવે છે. સુનિયોજિત અને સારી રીતે અમલ કરવામાં આવ્યો હોય, તેવી વીડિયો-ક્લિપ કે એનિમેશન, મલ્ટિમીડિયા વિનિયોગની અસરકારકતામાં નોંધપાત્ર વધારો કરે છે. Kung Fu Panda અને Smurfs એનિમેશન ફિલ્મનાં જાણીતાં ઉદાહરણ છે.

એનિમેશનને દ્વિ-પરિમાણીય (two-dimensional-2D) કે ત્રિ-પરિમાણીય (three dimensional-3D) એમ બે વિભાગમાં વિભાજિત કરી શકાય. દ્વિપરિમાણીય એનિમેશનમાં સ્ક્રીન પરના ચિત્રમાં X અને Y અક્ષને સંબંધિત દાર્શનિક ફેરફારો (visual change) કરવામાં આવે છે. આ પ્રકારનાં એનિમેશન સરળ છે. Macromedia Flash, Synfig Studio અને Pencil એ દ્વિપરિમાણીય પ્રકારનાં એનિમેશન સોફ્ટવેરનાં ઉદાહરણ છે. ત્રિ-પરિમાણીય એનિમેશનમાં દર્શનીય ફેરફારોને X, Y અને Z અક્ષને અનુલક્ષીને દર્શાવવામાં આવે છે. આ પ્રકારનાં એનિમેશન માનવ-ચક્ષુ સમક્ષ ચિત્રનું પ્રાયઃ વાસ્તવિક

દૃશ્ય રજૂ કરે છે. Maya, Blender અને 3D Max એ ત્રિ-પરિમાણીય એનિમેશનનાં કેટલાંક ઉદાહરણ છે. આકૃતિ 1.12 દ્વિ-પરિમાણીય એનિમેશન સૉફ્ટવેરમાં વસ્તુનો દેખાવ દર્શાવે છે અને ત્રિ-પરિમાણીય એનિમેશનમાં વસ્તુનો દેખાવ આકૃતિ 1.13માં દર્શાવ્યો છે. એનિમેશન તૈયાર કરવાની રીતનો આપણે હવે પછીના પ્રકરણમાં અભ્યાસ કરીશું.



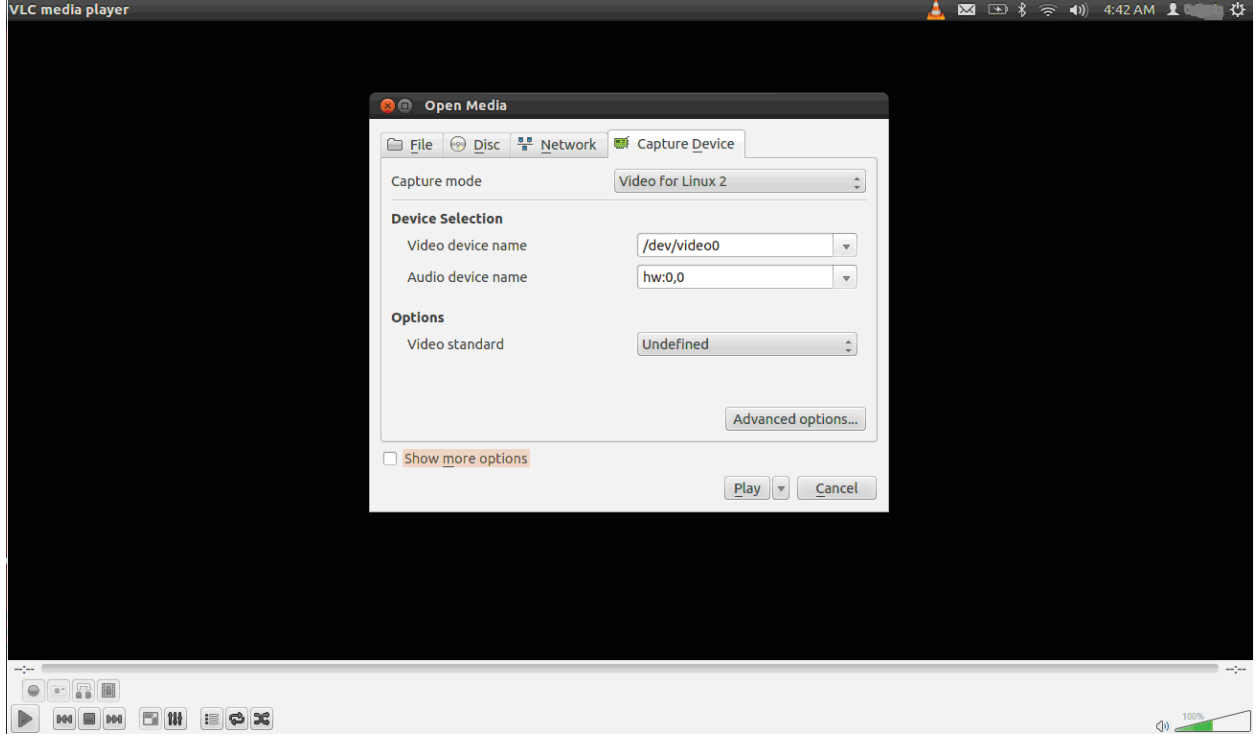
આકૃતિ 1.12 : વસ્તુનો દ્વિ-પરિમાણીય દેખાવ



આકૃતિ 1.13 : વસ્તુનો ત્રિ-પરિમાણીય દેખાવ

હવે આપણે VLC મીડિયાપ્લેયરનો ઉપયોગ કરી વીડિયો કેવી મુદ્રિત (રેકોર્ડ) કરી શકાય તે જોઈએ. આ માટે કમ્પ્યુટરમાં VLC મીડિયા પ્લેયર સ્થાપિત થયેલું હોય તે જરૂરી છે. જો ન હોય તો તેને ઉબન્ટુ સોફ્ટવેર સેન્ટર પરથી ડાઉનલોડ કરી શકાય છે. તે નિ:શુલ્ક અને ઓપનસોર્સ મલ્ટિમીડિયા પ્લેયર છે. હવે કમ્પ્યુટર સાથે વેબકેમેરા જોડાયેલ હોય તેની સૌપ્રથમ ખાતરી કરી લો. નીચેનાં પગલાંને અનુસરી વીડિયો મુદ્રિત કરવાનું કાર્ય કરી શકાય.

- VLC મીડિયાપ્લેયર ખોલો.
- Media → Open Capturing Device પસંદ કરો. આમ કરવાથી આકૃતિ 1.14માં દર્શાવ્યા મુજબ Open Media ડાયલોગબોક્સ રજૂ થશે. વીડિયો અને ઓડિયો માટેનાં સાધનનાં નામ પસંદ કરો. આકૃતિ 1.14 પસંદ કરેલ વિકલ્પો દર્શાવે છે.

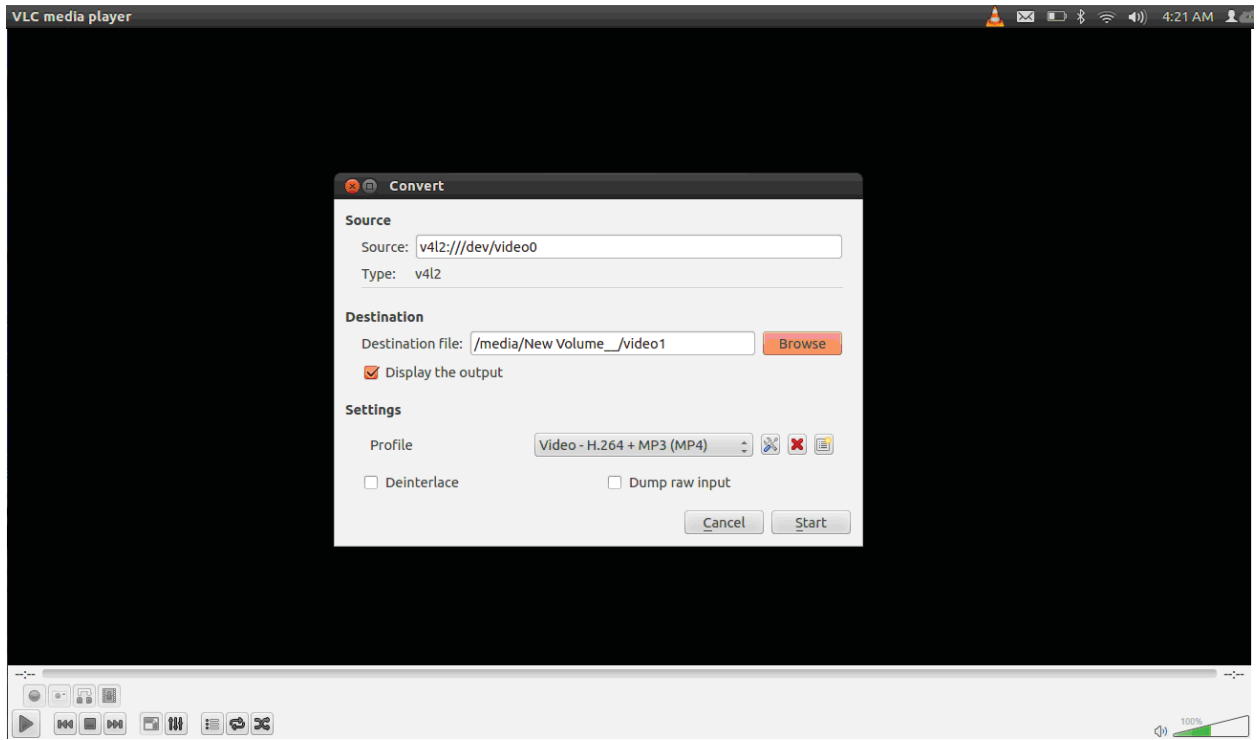


આકૃતિ 1.14 : Open Media ડાયલોગબોક્સ


નોંધ :

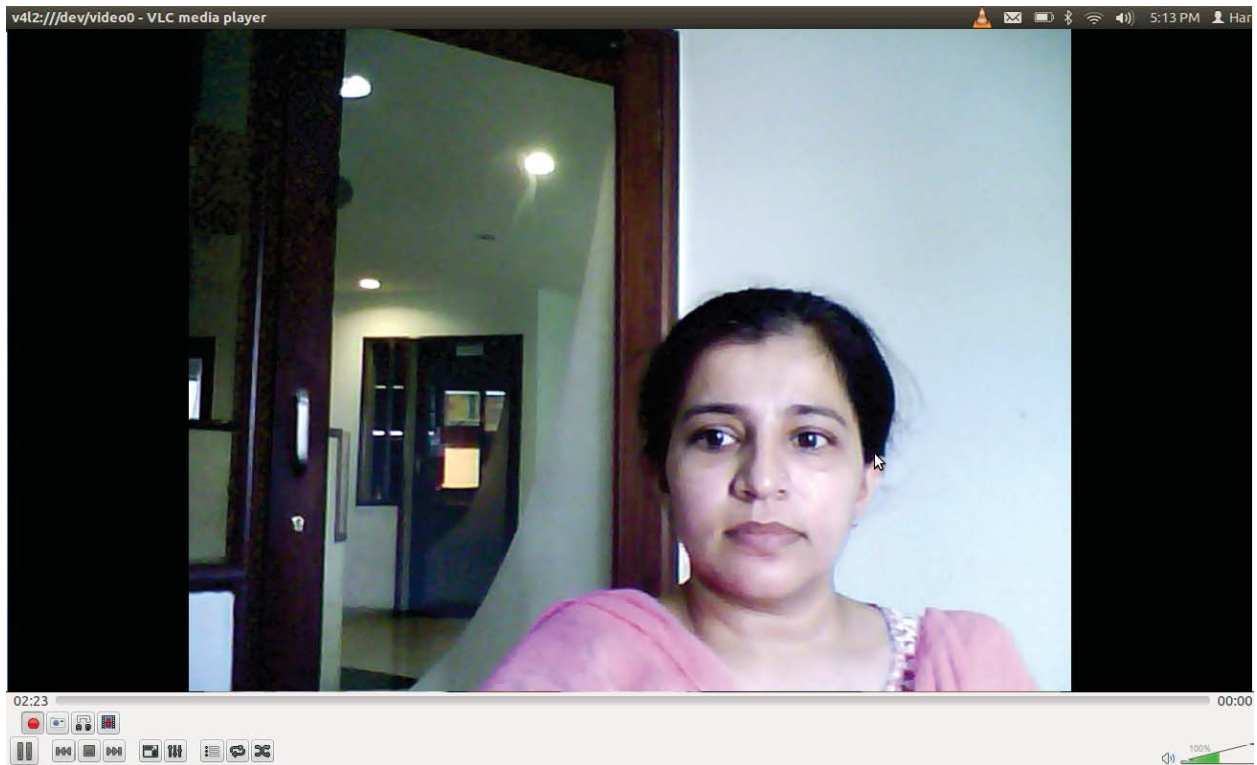
અહીં ઓડિયો અને વીડિયો સાધનોનાં નામ વાચકના મશીનની ગોઠવણ મુજબ જુદાં હોઈ શકે છે. એકથી વધુ સાધનોની હાજરીમાં ઉપયોગ માટે તેમાંથી કોઈ એકની પસંદગી કરી શકાય.

- આકૃતિ 1.14માં આવેલા *Open Media* ડાયલોગબોક્સમાં દર્શાવવામાં આવેલા Play ડ્રોપડાઉન મેનુમાંથી Convert વિકલ્પ પસંદ કરો. આમ કરવાથી આકૃતિ 1.15માં દર્શાવ્યા મુજબ *Convert* ડાયલોગબોક્સ ખૂલશે. *Destination file* લેબલની બાજુમાં આવેલા ટેક્સ્ટબોક્સમાં નિર્દિષ્ટ સ્થાન અને ફાઇલનું નામ પસંદ કરો. જે મુદ્રિત થઈ રહ્યું છે, તેને જોવા માટે 'Display the output' ચેકબોક્સ પસંદ કરો.
- Start બટન પર ક્લિક કરો.



આકૃતિ 1.15 : વીડિયોફાઇલનો સંગ્રહ કરવો

આકૃતિ 1.16માં વીડિયો લેવાની પ્રક્રિયા દર્શાવી છે. મુદ્રણ અટકાવવા Stop બટન () પર ક્લિક કરો.



આકૃતિ 1.16 : વીડિયો લેવાની પ્રક્રિયા

હવે પછી આ મુદ્દિત થયેલા વીડિયોને કોઈ મલ્ટિમીડિયા વિનિયોગ સાથે પણ સાંકળી શકશે. વીડિયો લેવા માટે ઉબન્ટુ સોફ્ટવેર સેન્ટરમાં ઉપલબ્ધ અન્ય એક સોફ્ટવેર Cheese Webcam Boothનો ઉપયોગ પણ કરી શકાય છે. કોષ્ટક 1.3માં વીડિયો માટે સામાન્ય રીતે ઉપયોગમાં લેવામાં આવતાં સ્વરૂપો દર્શાવ્યાં છે:

ફાઈલ એક્સટેન્શન	ફાઈલનો પ્રકાર	સમજૂતી
.avi	Audio Video Interleave ફાઈલ	વિન્ડોઝ ઓપરેટિંગ સિસ્ટમમાં વીડિયો ચલાવવા માટે માઈક્રોસોફ્ટ દ્વારા બનાવેલ ફાઈલ સ્વરૂપ
.wmv	Windows Media Format ફાઈલ	ઈન્ટરનેટ પર સ્ટ્રિમિંગ વિનિયોગો માટે માઈક્રોસોફ્ટ દ્વારા બનાવેલું એક સંકુચિત વીડિયો ફાઈલ સ્વરૂપ
.mpg, .mpeg	Moving Pictures Expert Group ફાઈલ	ઈન્ટરનેટ પર વીડિયો વહેંચવા માટેનું પ્રચલિત સ્વરૂપ
.mov	Quick Time Format ફાઈલ	એપલ કંપની દ્વારા રચિત, ઈન્ટરનેટ પરથી સરળતાથી ડાઉનલોડ થઈ શકે તે માટેનું સંકુચિત ફાઈલસ્વરૂપ
.rm, .ram	Real Video Format ફાઈલ	.ram (Real Audio Metadata) એ ધ્વનિ અને વીડિયોનું સંકલન છે.
.swf, .flv	Adobe Flash Shockwave ફાઈલ	એડોબ ફ્લેશ સોફ્ટવેર દ્વારા બનાવવામાં આવેલી લખાણ અને ચિત્રો ધરાવતી એનિમેશન ફાઈલ.
.mp4	MPEG4 Video ફાઈલ	ઈન્ટરનેટ પર વીડિયો વહેંચવા માટેનું સામાન્ય ફાઈલસ્વરૂપ

કોષ્ટક 1.3 વીડિયોના સામાન્ય રીતે ઉપયોગમાં લેવામાં આવતાં ફાઈલસ્વરૂપો

હાલમાં, ઈન્ટરનેટ પર સૌથી વધુ ઉપયોગમાં લેવામાં આવતું વીડિયો ફાઈલસ્વરૂપ mp4 છે. તેને ફ્લેશ પ્લેયર્સ, યુ-ટ્યૂબ તથા HTML5 દ્વારા પણ સમર્થન આપવામાં આવ્યું છે.

સંવાદિત ઘટકો (Interactive Objects)

મહત્તમ સંવાદિતતા ધરાવતી મલ્ટિમીડિયા રજૂઆત સૌથી વધુ અસરકારક નીવડે છે. કેટલીક વાર ઉપયોગકર્તા રજૂઆતના કોઈ અંશને છોડી દેવા માંગતો હોય અથવા તો વારંવાર નિહાળવા ઇચ્છતો હોય તેમ પણ બનવાજોગ છે. આ પ્રકારના કિસ્સામાં ઉપયોગકર્તાને નેવિગેશન માટે બટન અથવા હાઈપરલિંક આપવામાં આવે છે. જ્યારે આ સંવાદિત ઘટકો ઉપર ક્લિક કરવામાં આવે છે, ત્યારે રજૂઆતનો પ્રવાહ ઉપયોગકર્તાની પસંદગી મુજબ બદલાઈ જાય છે.

શરૂઆતના સમયમાં ઈન્ટરનેટ માટેનાં વેબબ્રાઉઝર માત્ર એક જ માધ્યમ-લખાણને સમર્થન આપતાં હતાં. લખાણ પણ એક જ ફોન્ટ અને રંગ પૂરતું મર્યાદિત રહેતું. માધ્યમ માટેના ઉપર સમજાવેલા તમામ ઘટકોને હવેના અદ્યતન ઈન્ટરનેટ બ્રાઉઝર દ્વારા જુદા-જુદા પ્રકારે સમર્થન આપવામાં આવી રહ્યું છે. કેટલાંક માધ્યમોના ઘટકો પૂર્વસ્થાપિત હોય છે, જ્યારે કેટલાંક માટે પ્લગ-ઇન (Plug-in) અનિવાર્ય બને છે. પૂર્વસ્થાપિત મીડિયા ઘટકોનો પ્રોગ્રામની અંદર જ <inlinemediaobject> ટેગ દ્વારા ઉપયોગ કરવામાં આવે છે. પ્લગ-ઇન એવા અતિરિક્ત અને સહાયક પ્રોગ્રામ છે, જેને સરળતાથી પ્રસ્થાપિત કરી વેબબ્રાઉઝરના એક ભાગ તરીકે ઉપયોગમાં લઈ શકાય છે. યોજનાના વિષય સંબંધિત તેમાં ઉપયોગમાં લેવામાં આવનાર

મીડિયા-ઘટકોની પસંદગી કરવામાં આવે છે. ઉદાહરણ તરીકે, “Gujarat Tourism” પરની મલ્ટિમીડિયા રજૂઆતમાં ગુજરાતનાં વિવિધ સ્થળોને લગતાં ચિત્રો, વીડિયો-ક્લિપ અને લખાણનો ઉપયોગ કરવામાં આવે. આ ઘટકો વિશેની વધુ માહિતી આપણે હવે પછીના પ્રકરણમાં મેળવીશું.

મલ્ટિમીડિયાનું વર્ગીકરણ (Classification of Multimedia)

સામાન્ય રીતે, મલ્ટિમીડિયાને સંવાદિત (Interactive) અને અસંવાદિત (Non-Interactive) એમ બે ભાગમાં વર્ગીકૃત કરી શકાય.

સંવાદિત મલ્ટિમીડિયામાં માધ્યમ-ઘટકોનો ક્રમ અને સમય ઉપયોગકર્તા દ્વારા નિયંત્રિત કરી શકાય છે. ઉદાહરણ તરીકે, “સંગીતનાં વાદ્યોનો અભ્યાસ” પરથી મલ્ટિમીડિયા સીડીમાં ઉપયોગકર્તા તેની મરજી મુજબનાં જુદાં-જુદાં વાદ્યો વગાડી શકે છે. વળી, ઉપયોગકર્તા જે-તે વાદ્ય સાથે સાંભળી શકાય તે માટે તેનું કોઈ મનપસંદ ગીત પણ પસંદ કરી શકે છે. બજારમાં ઉપલબ્ધ મોટા ભાગનાં CD-ROM આલ્બમ અને રમતો કાર્યમાં સંવાદિત હોય છે. સામાન્ય રીતે, સંવાદિતતાને કારણે ઉપયોગકર્તાની સહભાગિતા વધે છે તથા તે તેને વધુ માણી શકે છે. સંવાદિતતાનું પ્રમાણ મલ્ટિમીડિયા યોજનાના ક્ષેત્ર ઉપર આધારિત છે.

ટેલીવિઝન, રેડિયો અથવા વર્તમાનપત્ર જેવાં પરંપરાગત સામૂહિક માધ્યમો દ્વારા કરવામાં આવતું સંચારણ એકમાર્ગીય (one-way) હોય છે. બીજા શબ્દોમાં એમ કહી શકાય કે આ પ્રક્રિયા સ્રોત પરથી શરૂ થાય છે અને પ્રેક્ષકોના મોટા સમૂહને પ્રેષિત કરવામાં આવે છે. ઉદાહરણ તરીકે ટેલીવિઝન પર દર્શાવાતા મૂવીને ઉપયોગકર્તા નિહાળે છે કે નહીં તેનાથી અનપેક્ષ રીતે નિશ્ચિત પ્રારંભ અને અંત હોય છે. આમ, આ તકનીકો કોઈ સંવાદિતતા વગર વીડિયો, ધ્વનિ, લખાણ અને ચિત્રોનો ઉપયોગ કરે છે આથી વિપરીત મલ્ટિમીડિયામાં ઉપયોગકર્તા પોતે નિષ્ક્રિય રહેતો નથી, પરંતુ ઘટકોને નિયંત્રિત કરી શકે છે. આમ, સામૂહિક માધ્યમ (mass media) અને મલ્ટિમીડિયા વચ્ચેનો તફાવત એ છે કે મલ્ટિમીડિયામાં પ્રેક્ષક ઉપયોગકર્તા બને છે અને સંચારણ એકમાર્ગીય (one-way) ને બદલે દ્વિમાર્ગીય (two-way) હોય છે.

અસંવાદિત મલ્ટિમીડિયામાં ઉપયોગકર્તા માધ્યમને પ્રારંભથી અંત સુધી સીધેસીધું નિહાળે છે. પ્રવાહ પર તેનું કોઈ નિયંત્રણ હોતું નથી. ઉદાહરણ તરીકે કોઈ નિગમ (corporate)ની રજૂઆત અથવા મલ્ટિમીડિયાનું પ્રદર્શન.

કાર્યક્ષેત્રને અનુલક્ષીને મલ્ટિમીડિયાનું અન્ય પ્રકારે પણ વર્ગીકરણ કરી શકાય છે : મનોરંજનક્ષેત્ર અને શિક્ષણક્ષેત્ર.

તમામ પ્રકારની રમતો, મૂવી વગેરે મનોરંજનક્ષેત્રનાં મલ્ટિમીડિયા છે, જેનો ઉદ્દેશ ઉપયોગકર્તાને મનોરંજન પૂરો પાડવાનો હોય છે, જ્યારે ઉપયોગકર્તાને શિક્ષણ પૂરું પાડતાં શૈક્ષણિક સીડી-ટાઇટલ્સ શિક્ષણક્ષેત્રમાં ઉપયોગમાં લેવાતાં મલ્ટિમીડિયાનાં ઉદાહરણ છે.

મલ્ટિમીડિયાનો ઉપયોગ (Usage of Multimedia)

મલ્ટિમીડિયા આપણા જીવનનું અભિન્ન અંગ બની ચૂક્યું છે. મલ્ટિમીડિયા વિનિયોગોનો ઉપયોગ જાહેરાત, કલા, શિક્ષણ, મનોરંજન, ઈજનેરી, દવાઓ, ગણિત, વ્યવસાય અને વૈજ્ઞાનિક શોધ જેવાં અનેક ક્ષેત્રોમાં કરવામાં આવે છે. આપણે જુદાં-જુદાં કેટલાંક ક્ષેત્રોમાં મલ્ટિમીડિયાના ઉપયોગનો અભ્યાસ કરીએ.

શિક્ષણ અને તાલીમ (Education and Training)

શિક્ષણના ક્ષેત્રમાં મલ્ટિમીડિયાનું મહત્ત્વ ઘણું છે. આપણે ઘણા સમયથી શિક્ષણ આપવા માટે આલેખો, કોષ્ટકો અને અન્ય ઘટકો ધરાવતી રજૂઆતોનો ઉપયોગ કરતા આવ્યા છીએ. પરંતુ હાલના વર્ગોમાં આપવામાં આવતું શિક્ષણ પહેલાની પરંપરાગત પદ્ધતિઓ સુધી મર્યાદિત રહ્યું નથી, પરંતુ હવે તેમાં દૃશ્ય-શ્રાવ્ય માધ્યમોનો ઉપયોગ પણ કરવામાં આવે છે. મલ્ટિમીડિયાનો ઉપયોગ કમ્પ્યુટર આધારિત તાલીમ (Computer Based Training) માટે કરવામાં આવે છે, જેને સામાન્ય રીતે CBTથી ઓળખવામાં આવે છે. CBTમાં ઉપયોગકર્તા સમક્ષ રજૂઆતો, એનિમેશન, કોઈ મુદ્દા પરનું લખાણ વગેરેની શ્રેણીને ઉદાહરણ સહિત રજૂ કરવામાં આવે છે. હાલમાં CBT નો ઉપયોગ બાળવાડીથી લઈ અનુસ્નાતક શિક્ષણ સુધીનાં

તમામ વયજૂથ ધરાવતાં વિદ્યાર્થીઓ માટે કરવામાં આવે છે. ઉદાહરણ તરીકે 'બાળગીતો' અથવા 'કમ્પ્યુટર શીખો' જેવી સીડી. શિક્ષણ (Education) અને મનોરંજન (Entertainment)ને મલ્ટિમીડિયાના ઉપયોગથી સંયુક્ત રીતે રજૂ કરવા માટે પદનો ઉપયોગ કરવામાં આવે છે.

મલ્ટિમીડિયાનો ઘણો મોટો ઉપયોગ તાલીમ-કાર્યક્રમોમાં કરવામાં આવે છે. તબીબી તાલીમ આપનાર અને તબીબો પ્રત્યક્ષ શસ્ત્રક્રિયા કરતાં પહેલાં શસ્ત્રક્રિયાની પદ્ધતિનો અભ્યાસ કરી શકે છે; મિકેનિક એન્જિનનું સમારકામ શીખી શકે છે અથવા તો વેચાણ-અધિકારી વસ્તુની વિગતો વિશેનો અભ્યાસ કરી શકે છે.

શિક્ષણ અને તાલીમ માટે મલ્ટિમીડિયાના કેટલાક ઉપયોગો નીચે દર્શાવેલ છે :

- વ્યાખ્યાન તૈયાર કરી પ્રસારિત કરી શકાય છે.
- વીડિયો-કોન્ફરન્સિંગ (video-conferencing)ની મદદથી અન્યત્ર રહેલ નિષ્ણાતને જોઈ કે સાંભળી શકાય છે.
- સામાન્ય રીતે શસ્ત્રક્રિયા કે તેના જેવી અન્ય તકનિકો જોવાની તક મળતી નથી. વિદ્યાર્થીઓ સમક્ષ મલ્ટિમીડિયા દ્વારા તેનું પ્રદર્શન કરી શકાય છે, જેથી પછીથી તેને પ્રત્યક્ષ અમલમાં મૂકી શકાય.
- ફીડબેક (feedback)નો ઉપયોગ કરી વિદ્યાર્થીની પ્રગતિની નોંધ રાખી શકાય છે.

વિજ્ઞાપન (Advertisements)

વિજ્ઞાપનક્ષેત્રમાં મલ્ટિમીડિયા આવશ્યક ભાગ ભજવે છે. આજે, બજારમાં ઉત્પાદક સંસ્થાઓ દ્વારા કોઈ ઉત્પાદનને રજૂ કરવા તથા આગળ વધારવા માટે જાહેરાત એ એક મુખ્ય સ્ત્રોત છે. રોમાંચક એનિમેશન અને ટૂંકા અસરકારક શબ્દપ્રયોગો (slogan) જાહેરાતને આકર્ષક બનાવી શકે છે, તેનાથી વેચાણ વધે છે. આપણે આજકાલ ઘણાં વિજ્ઞાપનો જોઈએ છીએ અને તેમાંનાં કેટલાંક આપણું વિશેષ ધ્યાન આકર્ષિત કરે છે અને છેવટે આપણે તે ઉત્પાદન ખરીદી લઈએ છીએ. ઉદાહરણ તરીકે જુજુઝ (Jujus) અને pug શ્વાનનો ઉપયોગ કરી વોડાફોન સંસ્થાએ તેના અનેકવિધ પ્લાન અને પેકેજ આગળ વધાર્યા છે.

મનોરંજન (Entertainment)

એનિમેશન અને મૂવીમાં વિશિષ્ટ અસરો (special effects) ઉમેરવા માટે મનોરંજનક્ષેત્રમાં મલ્ટિમીડિયાનો ઉપયોગ મોટા પાયે થઈ રહ્યો છે. આઈસ એજ, જ્યુરાસિક પાર્ક, અવતાર વગેરે જેવી ફિલ્મોને તેમની વિશિષ્ટ અસરો અને એનિમેશનને કારણે હંમેશાં યાદ રાખવામાં આવશે.

બાળકોમાં મલ્ટિમીડિયા રમતો ઘણી પ્રચલિત છે અને આ પ્રકારની વિવિધ રમતો CD-ROM કે ઓનલાઈન ઉપલબ્ધ છે. વૈયક્તિક કે જૂથમાં રમી શકાય તે માટેની રમતોનાં અનેક સોફ્ટવેર ઉપલબ્ધ થવાને કારણે આભાસી રમતો આજે વાસ્તવિકતા બની ગઈ છે. આ પ્રકારના સોફ્ટવેરના ઉપયોગ દ્વારા જુદાં-જુદાં સ્થળે રહેલ બે વ્યક્તિઓ પોતાના કમ્પ્યુટર પર બેસી એક જ રમતમાં ભાગ લઈ શકે છે.

પત્રકારત્વ (Journalism)

પત્રકારત્વક્ષેત્રમાં મલ્ટિમીડિયા ઘણું ઉપયોગી છે. નિયત સમયાંતરે ઘણાં સામયિકો અને વર્તમાનપત્રો પ્રકાશિત થતાં હોય છે. આજે આપણે વર્તમાનપત્રમાં માત્ર લખાણ નહીં, પરંતુ તસવીરો પણ જોઈ શકીએ છીએ. ઈ-વર્તમાનપત્રો (E-newspaper) અને ઈ-સામયિકો (E-magazines) પણ ઓનલાઈન ઉપલબ્ધ હોય છે; જેમાં સમાચારને સંબંધિત વીડિયો પણ જોઈ શકાય છે.

મલ્ટિમીડિયાની રચનાનાં વિવિધ સોપાન (Stages in Multimedia Production)

કોઈ પણ મલ્ટિમીડિયા રજૂઆત સંભાળપૂર્વક તૈયાર કરેલી આયોજનબદ્ધ અને રૂપરેખા સહિત હોવી જોઈએ. આપણે જોયું કે મલ્ટિમીડિયા લગભગ તમામ ક્ષેત્રોમાં લાગુ પડે છે માટે તેનું આયોજન પસંદ કરેલ ક્ષેત્રને અનુલક્ષીને થવું જોઈએ.

ઉદાહરણ તરીકે, મનોરંજન માટે વિકસાવવામાં આવેલ વિનિયોગ, શિક્ષણ માટે યોગ્ય કાર્ય ન કરે તેમ બની શકે અને તેથી વિરુદ્ધ પણ કહી શકાય. મલ્ટિમીડિયાની રચનાનાં સોપાન કોઈ પણ ક્ષેત્રના વિનિયોગ માટે સમાન છે. મલ્ટિમીડિયાની રચનાનાં વિવિધ સોપાનોની ચર્ચા કરીએ.

સંશોધન અને વિશ્લેષણ (Research and Analysis)

આ સોપાન દરમિયાન પ્રેક્ષકો, તેમની કક્ષા, તેમની આવશ્યકતાઓ અને યોગ્યતા વિશે સંશોધન કરવામાં આવે છે. પ્રસ્તુતિ માટેની શક્ય એવી તમામ વિગતો પણ અહીં એકત્રિત કરવામાં આવે છે.

સ્ક્રિપ્ટિંગ કે ફ્લોચાર્ટિંગ (Scripting or Flowcharting)

મલ્ટિમીડિયાની યોજનાના પ્રવાહને નક્કી કરવાની ક્રિયા સ્ક્રિપ્ટિંગ કે ફ્લોચાર્ટિંગ તરીકે ઓળખાય છે. આ માટે ઉપયોગકર્તા પસંદ કરી શકે તે માટે મુખ્ય મેનુ અને તેમાં આવેલી શાખાઓ માટે ફ્લોચાર્ટની રચના કરવામાં આવે છે. ઉદાહરણ તરીકે જો ધોરણ-11ના કોઈપણ વિષય માટે આપણે શૈક્ષણિક મલ્ટિમીડિયા યોજના બનાવી રહ્યા હોઈએ, તો તેના હોમપેજમાં તમામ પ્રકરણોનાં નામ દર્શાવીશું. ચોક્કસ પ્રકરણ પસંદ કરવાથી તેની સમજૂતી, સ્વાધ્યાય, પ્રશ્નપત્રો અને તે પ્રકરણની અન્ય માહિતી દર્શાવવામાં આવશે. આમ, ફ્લોચાર્ટ બનાવવાની શરૂઆત મુખ્ય મેનુથી કરવામાં આવે છે અને ત્યાર બાદ તેના સબમેનુની રચના કરવામાં આવે છે, જે પરસ્પર સંકલિત હોય છે.

નોંધ : અહીં 'સ્ક્રિપ્ટિંગ (Scripting) પદ એક્શન-સ્ક્રિપ્ટિંગ (Action-scripting) કે જાવા સ્ક્રિપ્ટિંગ (Java scripting) રજૂ કરતું નથી.

સ્ટોરીબોર્ડિંગ (Storyboarding)

આ સોપાન અન્વયે યોજના વાસ્તવિક રીતે દર્શનીય બને છે. દરેક સ્ક્રીન કેવો દેખાશે, કયા માધ્યમ-ઘટકોનો ઉપયોગ કરવાનો છે અને તેમને સ્ક્રીન પર કયા સ્થાને મૂકવાના છે તે અહીં નક્કી કરવામાં આવે છે. અગાઉનું ઉદાહરણ લઈએ તો સ્ટોરીબોર્ડમાં હોમપેજની ગોઠવણ, બટન માટેનાં સ્થાન, તેનો દેખાવ, બટન પર ક્લિક કરવાથી અમલમાં મૂકવાની ક્રિયા અને અન્ય જરૂરી ઘટકોનો સમાવેશ કરી શકાય.

માધ્યમ-ઘટકોનું એકત્રીકરણ અને સંરચના (Collection of media elements and construction)

સ્ટોરીબોર્ડ તૈયાર થઈ ગયા બાદ, ડિઝાઇનર પાસે યોજનાનો નમૂનો (prototype) તૈયાર હશે અને હવે યોજના માટે ઉપયોગમાં લેવાનાં હોય તેવાં ચિત્રો અને અન્ય માધ્યમ-ઘટકોની રચના તે શરૂ કરશે. ઉદાહરણ તરીકે, પાત્રોની રચના કરવામાં આવશે, ધ્વનિ અને વીડિયોનું મુદ્રણ કરવામાં આવશે તથા એનિમેશન બનાવવામાં આવશે.

પ્રોગ્રામિંગ (Programming)

માધ્યમ-ઘટકોના એકત્રીકરણ અને સંરચના પછી મેક્રોમીડિયા ફ્લેશ, સીન્કિંગ, હાઈપર કાર્ડ વગેરે જેવાં સોફ્ટવેર પેકેજની મદદથી તેમને અંતિમ ઉત્પાદન સ્વરૂપે યોજના સાથે સાંકળવામાં આવે છે.

ચકાસણી (Testing)

મલ્ટિમીડિયા યોજનાનું આ અંતિમ સોપાન છે. આમાં તમામ માધ્યમ-ઘટકો આવશ્યકતાનુસાર કાર્ય કરી રહ્યા છે કે નહીં તેની ચકાસણી કરવામાં આવે છે. તદુપરાંત પ્રેક્ષકોને પ્રસ્તુતિની શૈલી અને વિગતો આકર્ષક લાગશે કે નહીં તેનો વિચાર પણ કરવામાં આવે છે.

અહીં આપવામાં આવેલ સોપાનો મુજબ જો મલ્ટિમીડિયા વિનિયોગનું આયોજન કરવામાં આવે, તો તે મોટા ભાગે શ્રેષ્ઠ વિનિયોગ બની શકે છે.

સારાંશ

આ પ્રકરણમાં આપણે મલ્ટિમીડિયા અને તેના ઘટકો જેવા કે, લખાણ, ધ્વનિ, ચિત્રો, વીડિયો અને એનિમેશન વિશે ચર્ચા કરી. મલ્ટિમીડિયા યોજનાના વિકાસ માટે આ દરેક મહત્વનો ભાગ ભજવે છે માટે આપણે દરેક ઘટક વિશે વિસ્તૃત ચર્ચા કરી. મલ્ટિમીડિયાને સંવાદિત, અસંવાદિત, મનોરંજનક્ષેત્ર માટે અને શિક્ષણ માટે એમ વિભાજિત કર્યાં. શિક્ષણ અને તાલીમ, વિજ્ઞાપન, મનોરંજન, પત્રકારત્વ જેવાં અનેક ક્ષેત્રોમાં મલ્ટિમીડિયાનો ઉપયોગ વ્યાપક પ્રમાણમાં થઈ રહ્યો છે. મલ્ટિમીડિયા વિનિયોગને પદ્ધતિસર અને લાભદાયી બનાવવા માટે તેના વિકાસ માટેનાં સોપાન અગત્યનો ભાગ ભજવે છે.

સ્વાધ્યાય

1. મલ્ટિમીડિયા એટલે શું? મલ્ટિમીડિયાનાં ઘટકોની યાદી બનાવો.
2. ટાઈપફેસ અને ફોન્ટ એટલે શું?
3. સંવાદિત ઘટકો વિશે માહિતી આપો.
4. મલ્ટિમીડિયાનું વર્ગીકરણ યોગ્ય ઉદાહરણ સાથે સમજાવો.
5. જુદાં-જુદાં ક્ષેત્રોમાં થતાં મલ્ટિમીડિયાના ઉપયોગ વર્ણવો.
6. મલ્ટિમીડિયાની રચના માટેનાં સોપાનની યાદી બનાવો. દરેક સોપાન સંક્ષેપમાં સમજાવો.
7. નીચેના ઘટકો માટેનાં જુદાં-જુદાં ફાઈલસ્વરૂપોની યાદી બનાવો.
(a) ધ્વનિ (b) વીડિયો (c) ચિત્ર

8. હેતુલક્ષી પ્રશ્નો

- (1) મલ્ટિમીડિયા શબ્દ કયા શબ્દો પરથી બનાવવામાં આવ્યો છે ?
(a) mix, media (b) mid, mediate
(c) multi, mediator (d) multiple, media
- (2) મલ્ટિમીડિયામાં મહત્તમ કેટલાં માધ્યમોને સાંકળી શકાય ?
(a) 3 (b) 4
(c) 5 (d) 8
- (3) કોઈ પણ ભાષામાં અર્થપૂર્ણ ધ્વનિને શું કહે છે ?
(a) Video (b) Sound
(c) Text (d) Animation
- (4) Times, Century, Bookmark વગેરે નીચેનાંમાંથી કયા પ્રકારના ફોન્ટનાં ઉદાહરણ છે :
(a) Arial (b) Typeface
(c) Serif (d) Sans serif
- (5) વિન્ડોઝમીડિયા કોમ્પ્રેશનનો ઉપયોગ કરી બનાવેલ ધ્વનિ ફાઈલસ્વરૂપ નીચેનાંમાંથી કયું છે ?
(a) .mid (b) .wav
(c) .mp3 (d) .wma

- (6) અક્ષરની ટોચથી સૌથી નીચેના બિંદુ વચ્ચેના અંતરને કયા પદ દ્વારા રજૂ કરવામાં આવે છે ?
- (a) કદ (b) શૈલી
(c) લંબાઈ (d) પહોળાઈ
- (7) જુદી-જુદી શૈલીઓ અને કદ ધરાવતા ટાઈપફેસને રજૂ કરવા માટે કયા પદનો ઉપયોગ કરવામાં આવે છે ?
- (a) ફોન્ટ (b) ટેક્સ્ટ
(c) લેટર (d) કેરેક્ટર
- (8) મેકોમીડિયા ફ્લેશ, સીન્કિંગ સ્ટુડિયો અને પેન્સિલ કયા પ્રકારનાં એનિમેશન સોફ્ટવેરનાં ઉદાહરણ છે ?
- (a) 3D (b) 2D
(c) 4D (d) 5D
- (9) કમ્પ્યુટરની સ્ક્રીન પર આવેલા નાના ટપકાને કયા નામથી ઓળખવામાં આવે છે ?
- (a) Pixel (b) Point
(c) Cursor (d) Cell
- (10) ચિત્રોને કયા વર્ગોમાં વર્ગીકૃત કરી શકાય ?
- (a) સદિશ, રાસ્ટર (b) બીટમેપ, રાસ્ટર
(c) ચિત્ર, ક્લિપઆર્ટ (d) ગ્રાફિક, ચિત્ર
- (11) સમય પ્રમાણે થતા દર્શનીય ફેરફારોને રજૂ કરવા શેનો ઉપયોગ કરવામાં આવે છે ?
- (a) લખાણ (b) ધ્વનિ
(c) એનિમેશન (d) ચિત્ર
- (12) ટપકાંઓના દ્વિ-પરિમાણીય સમૂહને કયા નામથી ઓળખવામાં આવે છે ?
- (a) Screen (b) Map
(c) Cell (d) Array
- (13) MIDIનું સંપૂર્ણ સ્વરૂપ કયું છે ?
- (a) Musical Instrument Data Interface (b) Musical Instrument Digital Interface
(c) Musical Image Digital Interface (d) Musical Instrument Digital Image
- (14) ડિજિટલ કેમેરા દ્વારા ઉપયોગમાં લેવામાં આવતું સૌથી સામાન્ય ચિત્ર ફાઈલસ્વરૂપ કયું છે ?
- (a) .gif (b) .jpeg
(c) .tif (d) .bmp
- (15) નીચેનામાંથી કઈ પદ્ધતિમાં ગાણિતિક સમીકરણોની મદદથી ચિત્રનો સંગ્રહ કરવામાં આવે છે ?
- (a) સદિશ (b) રાસ્ટર
(c) બીટમેપ (d) પિક્ચર
- (16) તસવીરોને ડિજિટલ સ્વરૂપમાં રૂપાંતરિત કરવા કયા સાધનનો ઉપયોગ કરવામાં આવે છે ?
- (a) સ્કેનર (b) મોડેમ
(c) પ્રિન્ટર (d) મોનિટર

(17) PNG એટલે શું ?

- (a) Programmable Network Graphic (b) Photo Network Graphic
(c) Portable New Graph (d) Portable Network Graphic

(18) વેબબ્રાઉઝરના ભાગ તરીકે સરળતાથી પ્રસ્થાપિત કરી ઉપયોગમાં લઈ શકાય તેવા સહાયક પ્રોગ્રામને શું કહે છે ?

- (a) plug-in (b) text
(c) video (d) Picture

(19) મલ્ટિમીડિયાને નીચેનામાંથી કયા પદમાં વહેંચી શકાય ?

- (a) બુદ્ધિશાળી અને બુદ્ધિહીન (intelligent and non-intelligent)
(b) સંવાદિત અને અસંવાદિત (interactive and non-interactive multimedia)
(c) અંતજ્ઞાની અને અનંતજ્ઞાની (intuitive and non-intuitive)
(d) માહિતીપ્રદ અને માહિતીરહિત (informative and non-informative)

(20) નીચેનામાંથી એનિમેશનના કયા પ્રકારમાં x અને y અક્ષ પર દર્શનીય ફેરફારો ઉદ્ભવે છે ?

- (a) 2D (b) 3D
(c) 4D (d) 5D

પ્રાયોગિક સ્વાધ્યાય

1. “Gujarat” લખાણ માટે ઓપન ઓફિસ ઈમ્પ્રેસ કે રાઈટરનો ઉપયોગ કરી શીર્ષક લખાણ તૈયાર કરો.
2. ‘અડાલજની વાવ’ વિશે સાઉન્ડ રેકોર્ડરનો ઉપયોગ કરી તમારો અવાજ મુદ્રિત કરો.
3. ‘મલ્ટિમીડિયા’ શું છે?’ વિષયને સમજાવતો વીડિયો મુદ્રિત કરો.





પ્રથમ પ્રકરણમાં આપણે મલ્ટિમીડિયા અને તેમાં ઉપયોગમાં લેવામાં આવતા મૂળભૂત ઘટકો વિશે ચર્ચા કરી. આ મૂળભૂત ઘટકોનો ઉપયોગ કરીને મલ્ટિમીડિયા રજૂઆતની રચના કરવા માટે મલ્ટિમીડિયા ઓથરિંગ ટૂલ્સ (Multimedia Authoring Tools)ના નામે ઓળખાતા વિનિયોગ પ્રોગ્રામની જરૂર પડે છે. આ ઓથરિંગ વિનિયોગો ચિત્રો, ધ્વનિ, વીડિયો અને એનિમેશન જેવા મલ્ટિમીડિયા ઘટકોની ગોઠવણ કરવા માટે મહત્વનું માળખું પૂરું પાડે છે.

સામાન્ય રીતે ઓથરિંગ ટૂલ્સને નીચે મુજબ વર્ગીકૃત કરી શકાય :

- કાર્ડ અથવા પૃષ્ઠ આધારિત ટૂલ્સ
- આઈકન અને ઘટના (Event) આધારિત ટૂલ્સ
- સમય આધારિત ટૂલ્સ

કાર્ડ કે પૃષ્ઠ આધારિત ટૂલ્સમાં ઘટકોને પુસ્તકનાં પાનાં કે કાર્ડના ઢગલા સ્વરૂપે ગોઠવવામાં આવે છે. ઓથરિંગ ટૂલ આ પાનાંને શ્રેણીમાં ગોઠવે છે. જુદા-જુદા ઘટકોને પુસ્તકનાં પાનાં સ્વરૂપે સ્વતંત્ર રીતે દર્શાવવાના હોય ત્યારે આ પ્રકારનાં ટૂલ્સ ઉપયોગી છે. Hyper card અને Multimedia Toolbox એ કાર્ડ આધારિત ઓથરિંગ ટૂલનાં ઉદાહરણ છે.

આઈકન અને ઘટના આધારિત ઓથરિંગ ટૂલમાં ઘટકોને માળખાકીય ગોઠવણ અથવા પ્રક્રિયાના સ્વરૂપે ગોઠવવામાં આવે છે. અહીં ઘટના (event) અથવા કાર્યો (task) માટે ફ્લોચાર્ટ બનાવવામાં આવે છે અને ત્યાર બાદ માળખા પ્રમાણે તેમાં ઘટકો ઉમેરવામાં આવે છે. આઈકન અને ઘટના આધારિત ઓથરિંગ ટૂલ્સમાં Authorware અને IconAuthor જેવાં સોફ્ટવેરનો સમાવેશ કરવામાં આવે છે.

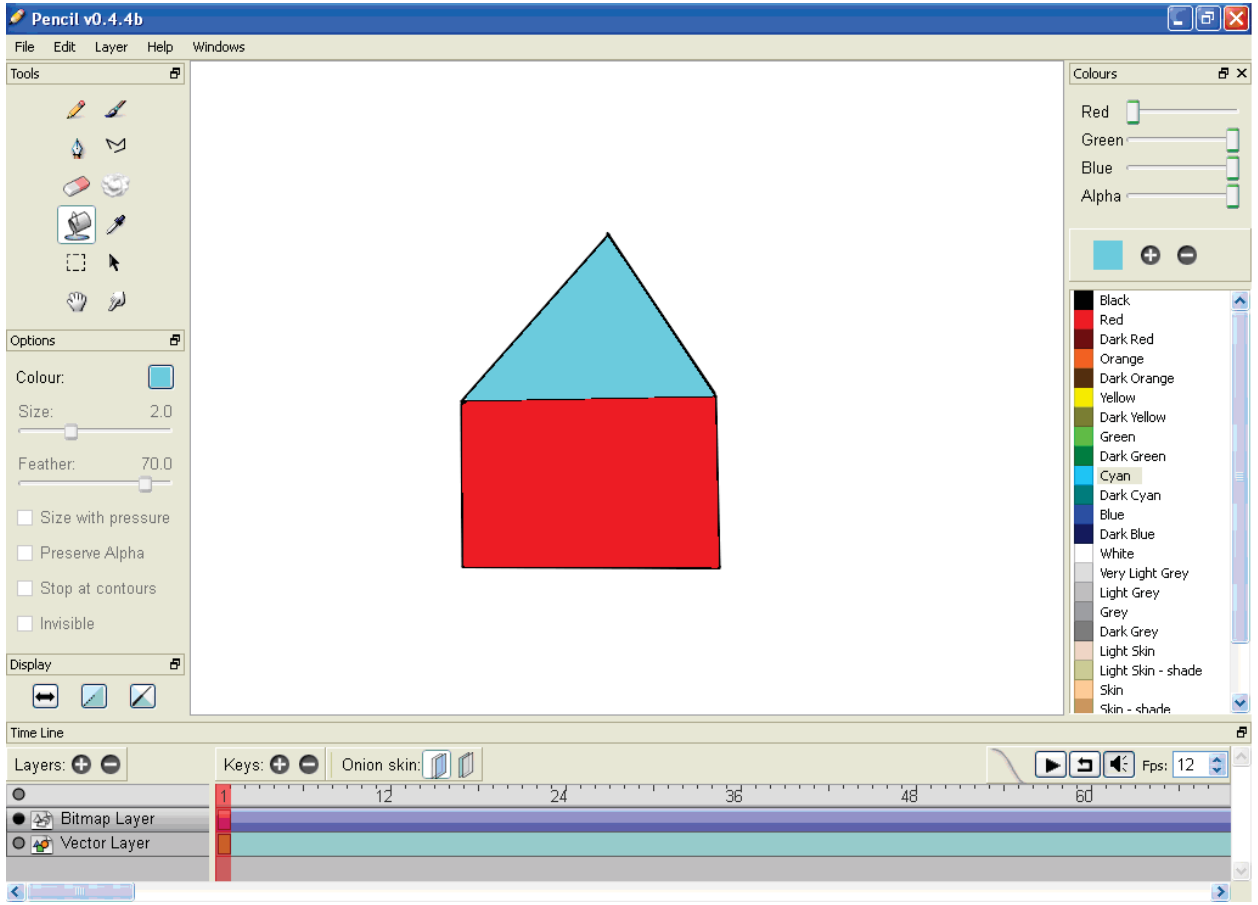
સમય આધારિત ઓથરિંગ ટૂલ્સમાં ઘટકોને સમયરેખા (ટાઈમલાઈન) દ્વારા ગોઠવવામાં આવે છે, જે સંદેશને પ્રારંભ અને અંત હોય તેવા કિસ્સામાં આ ઉપયોગી છે. ઘટનાના સમય અને સ્થાન મુજબ અહીં ઘટકોને દર્શાવવામાં આવે છે. સમય આધારિત ઓથરિંગ ટૂલ્સ સૌથી વધુ પ્રચલિત છે. Synfig, Pencil, Flash અને Director એ સમય આધારિત ઓથરિંગ ટૂલ્સનાં કેટલાંક ઉદાહરણ છે.

બજારમાં ઘણાં મલ્ટિમીડિયા ટૂલ્સ ઉપલબ્ધ છે. તેમાંનાં કેટલાંક માલિકીહક ધરાવતાં તો કેટલાંક મુક્ત-વિતરિત (open source) છે. ઇન્ટરનેટ પર સરળતાથી ઉપલબ્ધ એવાં કેટલાંક ઓપનસોર્સ એનિમેશન ટૂલ્સમાં 'પેન્સિલ' અને 'સીન્ડિંગ'નો સમાવેશ કરી શકાય.

પેન્સિલ (Pencil)

બીટમેપ અને સદિશ એમ બન્ને પ્રકારનાં ચિત્રો માટે પરંપરાગત હસ્તચિત્રિત એનિમેશન બનાવવા માટે પેન્સિલ નામના દ્વિ-પરિમાણીય એનિમેશન સોફ્ટવેરનો ઉપયોગ કરવામાં આવે છે. તેને <http://www.pencil-animation.org>. પરથી નિ:શુલ્ક ડાઉનલોડ કરી શકાય છે. તે વિન્ડોઝ, લિનક્સ અને મેક જેવી ઘણી ઓપરેટીંગ સિસ્ટમ માટે ઉપલબ્ધ છે.

આકૃતિ 2.1માં પેન્સિલ સોફ્ટવેરનો કાર્યવિસ્તાર દર્શાવ્યો છે. આ કાર્યક્ષેત્ર અન્ય એનિમેશન કે ચિત્ર બનાવવા માટેના સોફ્ટવેરને મળતું આવે છે. પદાર્થચિત્ર (object drawing)ને બદલે મુક્તહસ્ત ચિત્ર (freehand drawing)માં પારંગત હોય તેવી વ્યક્તિ પેન્સિલ એનિમેશન સોફ્ટવેરમાં સારાં એનિમેશન તૈયાર કરી શકે છે.



આકૃતિ 2.1 : પેન્સિલ એનિમેશન સોફ્ટવેરનો કાર્યવિસ્તાર

પદાર્થચિત્ર અને એનિમેશન માટે સીન્ફિગ સ્ટુડિયો (Synfig Studio) નામના એક ઓપનસોર્સ એનિમેશન ટૂલનો અભ્યાસ કરીએ.

સીન્ફિગ સ્ટુડિયો (Synfig Studio)

સીન્ફિગ સ્ટુડિયો એ સમય આધારિત મલ્ટિમીડિયા ઓથરિંગ ટૂલ છે, જેનો સમાવેશ ઓપનસોર્સ તરીકે કરવામાં આવ્યો છે. તે ઇન્ટરનેટ પર નિ:શુલ્ક ઉપલબ્ધ છે અને <http://www.synfig.org> પરથી ડાઉનલોડ કરી શકાય છે. સીન્ફિગ સ્ટુડિયો (અથવા માત્ર સીન્ફિગ) એ GNU GPL (General Public License) હેઠળ ઈ.સ. 2005માં રજૂ કરવામાં આવેલું દ્વિ-પરિમાણીય સદિશ એનિમેશન સોફ્ટવેર છે. અલ્પતમ વ્યક્તિઓ અને સ્રોત સાથે ફિલ્મ કક્ષાનું એનિમેશન બનાવી શકાય તે માટે સીન્ફિગની રચના કરવામાં આવી છે. તે લિનક્સ, વિન્ડોઝ અને MacOS X જેવી વિવિધ ઓપરેટિંગ સિસ્ટમ માટે ઉપલબ્ધ છે. સીન્ફિગની સૌપ્રથમ આવૃત્તિ 0.61.05 હતી. તેમાં વધારાની સુવિધાઓ સાથેની નવી આવૃત્તિઓ નિશ્ચિત સમયાંતરે આજ સુધી બહાર પડતી રહી છે. સીન્ફિગની અદ્યતન આવૃત્તિ 0.63.05 છે. ઉબન્ટુ લિનક્સ સાથે પૂર્વનિર્ધારિત રીતે ઉપલબ્ધ ન હોવાને કારણે તેને પ્રસ્થાપિત કરવું જરૂરી છે. પ્રસ્થાપના બાદ તે Applications મેનુના Graphics વિકલ્પમાં ઉપલબ્ધ બને છે.

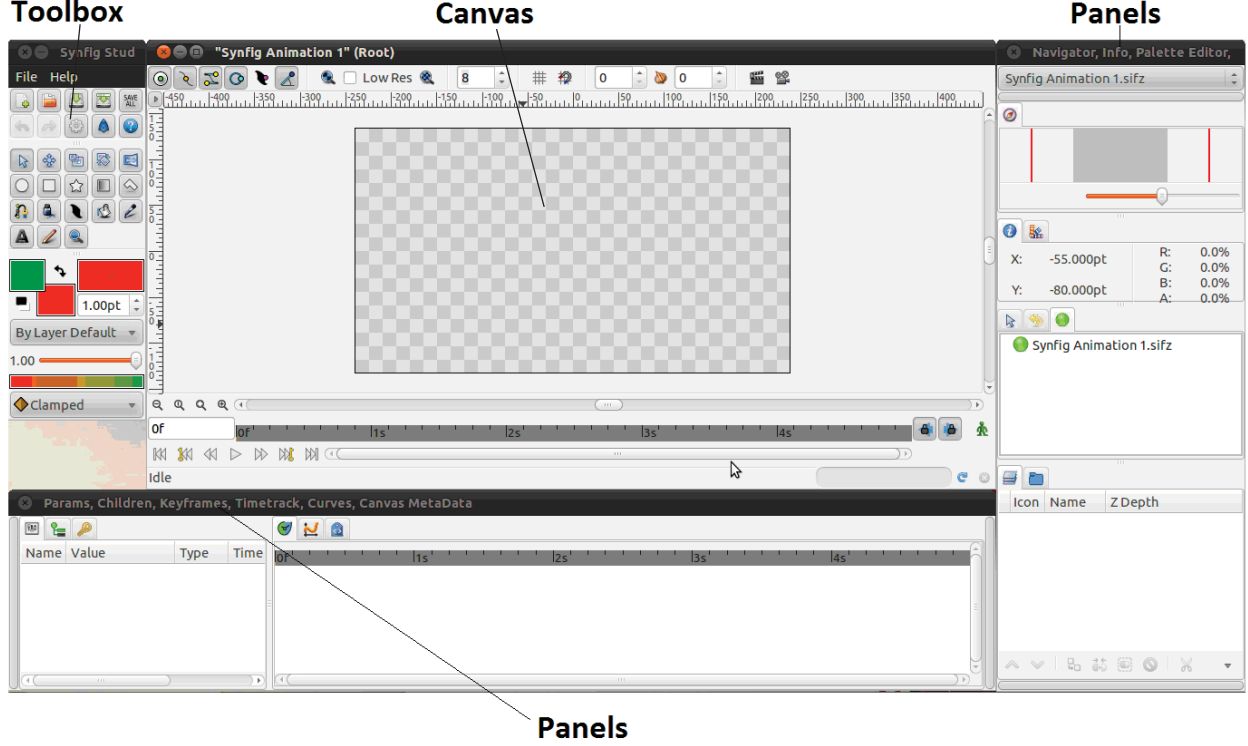
સીન્ફિગ એનિમેશનને XML (Extensible Markup Language) ફાઇલ સ્વરૂપમાં સાચવે છે અને કેટલીક વાર gzip નામના સંકોચન સોફ્ટવેર દ્વારા સંકુચિત બનાવે છે. gzip એ GNU zipનું ટૂંકું રૂપ છે. સીન્ફિગ ફાઇલોનું એક્સટેન્શન તેના સંકોચનરહિત (uncompressed) સ્વરૂપમાં .sif અને સંકુચિત (compressed) સ્વરૂપમાં .sifz આપવામાં આવે છે.

પ્રારંભ (Getting Started)

સીન્ફિગ શરૂ કરવા માટે Applications → Graphics → Synfig મેનુ-વિકલ્પ પસંદ કરવામાં આવે છે. જ્યારે સીન્ફિગ સ્ટુડિયો શરૂ કરવામાં આવે ત્યારે જોઈ શકાય છે કે તે એક જ વિન્ડો દ્વારા ખોલવામાં આવતું નથી, પરંતુ ડેસ્કટોપ

પર અનેક સ્વતંત્ર વિન્ડો રજૂ કરવામાં આવે છે. આ વિન્ડોની સીમારેખાને ડ્રેગ કરી તેનું કદ બદલી શકાય છે. આકૃતિ 2.2માં દર્શાવ્યા મુજબ જો સીન્કિંગ સ્ટુડિયોની વિન્ડોને તેની પૂર્વનિર્ધારિત રચના દ્વારા ગોઠવવી હોય તો Toolbox → File menu → Panels → Reset Windows to Original layout વિકલ્પ પસંદ કરી શકાય છે.

સૌપ્રથમ સીન્કિંગના કાર્યક્ષેત્રનો પરિચય મેળવીએ. આકૃતિ 2.2માં સીન્કિંગ સ્ટુડિયો વિન્ડોની ગોઠવણ દર્શાવેલી છે.



આકૃતિ 2.2 : સીન્કિંગનું કાર્યક્ષેત્ર

આકૃતિ 2.2માં દર્શાવ્યા પ્રમાણે કાર્યક્ષેત્રમાં ટૂલબોક્સ (Toolbox), કેનવાસ (Canvas) અને પેનલ (Panel)નો સમાવેશ કરવામાં આવ્યો છે. આ દરેક ઘટક વિશે ચર્ચા કરીએ.

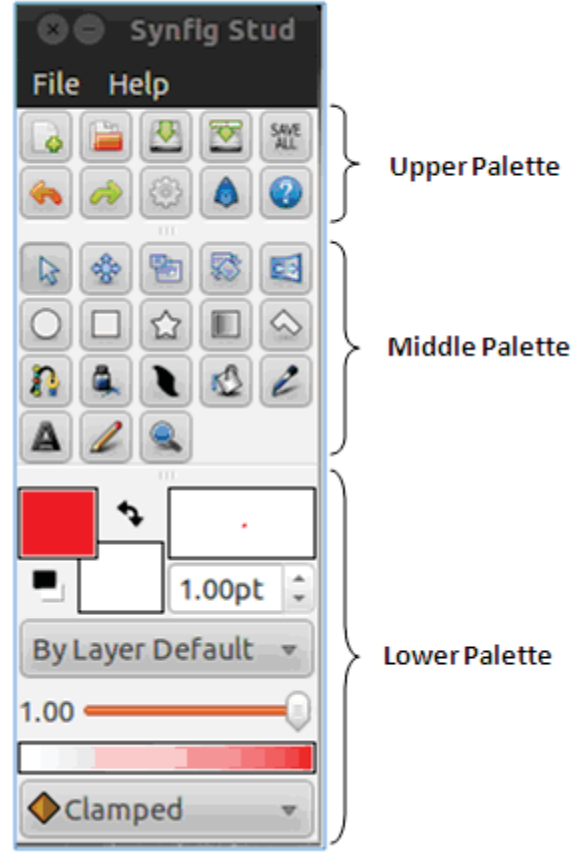
ટૂલબોક્સ (Toolbox)

આકૃતિ 2.3માં દર્શાવ્યા મુજબ ટૂલબોક્સ એ મુખ્ય વિન્ડો છે; જેમાં સિસ્ટમ-મેનુ અને ચિત્રો (artwork) બનાવવા કે સુધારવા માટેનાં સાધનોનો સમાવેશ કરવામાં આવે છે. ટૂલબોક્સને બંધ કરવાથી સીન્કિંગ વિનિયોગ પણ બંધ થઈ જાય છે. સીન્કિંગમાં અનેક યોજનાઓ ખોલી શકાય છે, પરંતુ તે દરેક માટે એક જ ટૂલબોક્સ ઉપલબ્ધ હોય છે. ટૂલબોક્સના ઉપયોગથી નવી ફાઈલ બનાવી શકાય છે, સંગ્રહ કરેલી ફાઈલ ખોલી શકાય છે, ફાઈલનો સંગ્રહ કરી શકાય છે તથા લાક્ષણિકતાઓ (properties) ગોઠવી શકાય છે. ટૂલબોક્સમાં Circle, Rectangle, Transform અને અન્ય ઓબ્જેક્ટ બનાવવા તથા સુધારવા માટેના સામાન્ય ટૂલ્સનો પણ સમાવેશ થયેલો છે ટૂલબોક્સ વિન્ડો ત્રણ ક્ષેત્ર (area) અથવા પેલેટ (palette)માં વહેંચાયેલી છે :

- ઉપરની પેલેટમાં ફાઈલ બનાવવા માટે, સંગૃહીત ફાઈલ ખોલવા માટે, વર્તમાન કે તમામ ફાઈલોનો સંગ્રહ કરવા માટે, હાલમાં કરેલ કાર્ય રદ કરવા માટે (undo) કે ફરી કરવા માટે (redo) તથા ગોઠવણી માટેના ડાયલોગ-બોક્સ ખોલવા અને મદદ મેળવવા માટેનાં બટન આપેલાં છે.
- વચ્ચેની પેલેટમાં Circle, Rectangle, Transform વગેરે જેવા ઓબ્જેક્ટની રચના કરવા તથા સુધારવા માટેનાં ટૂલ્સ આપવામાં આવ્યાં છે. આપણે હવે પછી આ પ્રકરણમાં આ દરેક ટૂલ વિશેનો અભ્યાસ કરીશું.

- સૌથી નીચેની પેલેટમાં નવા લેયર માટેની પૂર્વનિર્ધારિત ગોઠવણો સમાયેલી હોય છે, જેમકે,

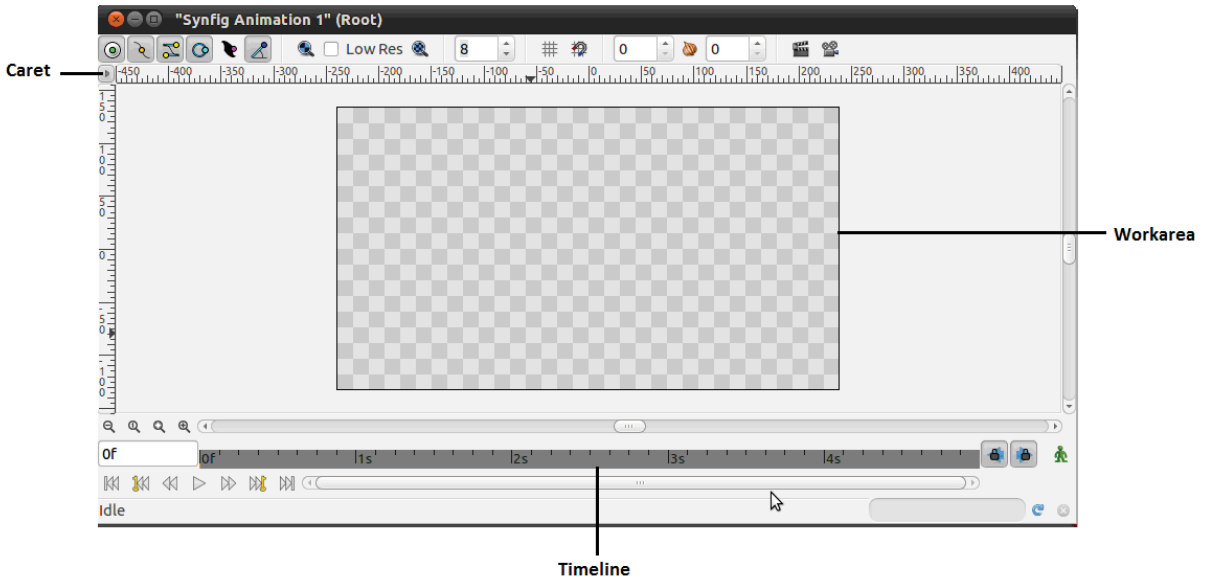
- **Foreground and Background Color :** તે ઓબ્જેક્ટની અંદરના (fill) રંગ અને સીમારેખા (outline)ના રંગની રચના કરે છે. જો સીમારેખા વગરના લેયરની રચના કરવામાં આવે, તો તેને માત્ર આંતરિક (fill) રંગ હશે. આંતરિક રંગ કે સીમારેખા પર ક્લિક કરી, તેનો રંગ બદલી શકાય છે.
- **Brush Size :** સીમારેખાના નવા લેયર માટે લીટીનું કદ નક્કી કરે છે.
- **Blend Method :** બ્લેન્ડ પદ્ધતિની ગોઠવણ કરે છે, જેના વિશે હવે પછીના પ્રકરણમાં ચર્ચા કરવામાં આવી છે. તેને “By Layer Default” તરીકે ગોઠવવામાં આવી છે.
- **Opacity :** તે લેયરની દૃશ્યતાનું નિયંત્રણ કરે છે. 0નો અર્થ થાય છે કે લેયર અદૃશ્ય છે, જ્યારે 1 કિંમત લેયરની દૃશ્યતા દર્શાવે છે.
- **Gradient :** લેયરના ‘ફિલ’ તરીકે ગ્રેડિયન્ટનો ઉપયોગ કરે છે. તે ફોરગ્રાઉન્ડ અને બેકગ્રાઉન્ડ રંગો વચ્ચે ક્રમશઃ વધતા જતા રંગની અસર તરીકે ગોઠવવામાં આવે છે.
- **Interpolation :** દરેક માર્ગબિંદુ (waypoint) પાસે વધારાની ઉમેરેલી (interpolation) ગોઠવણ હોય છે, જે પ્રાયલ (parameter)ના બદલાવની પદ્ધતિ નક્કી કરે છે.



આકૃતિ 2.3 : ટૂલબોક્સ


કેનવાસ (Canvas)

જ્યારે નવી યોજના માટે નવી ફાઇલ ખોલવામાં આવે, ત્યારે સ્ક્રીન પર કેનવાસ વિન્ડો દેખાય છે. આકૃતિ 2.4માં દર્શાવ્યા મુજબ તે કાર્યક્ષેત્રના મધ્ય ભાગમાં ગોઠવવામાં આવે છે. કેનવાસમાં એનિમેશન બનાવવા માટેનાં ચિત્રો (artwork) ઉમેરવામાં આવે છે. જ્યારે સીન્કિંગ સ્ટુડિયો શરૂ કરવામાં આવે છે, ત્યારે નવી કેનવાસ વિન્ડો રજૂ કરવામાં આવે છે.



આકૃતિ 2.4 : કેનવાસ વિન્ડો

અહીં રાખોડી ચોખંડાની ભાત ધરાવતો વિસ્તાર એ કાર્યવિસ્તાર છે, જેમાં વિવિધ ઘટક/લેયર બનાવવામાં આવે છે તથા તેમાં ફેરફારો કરી શકાય છે.

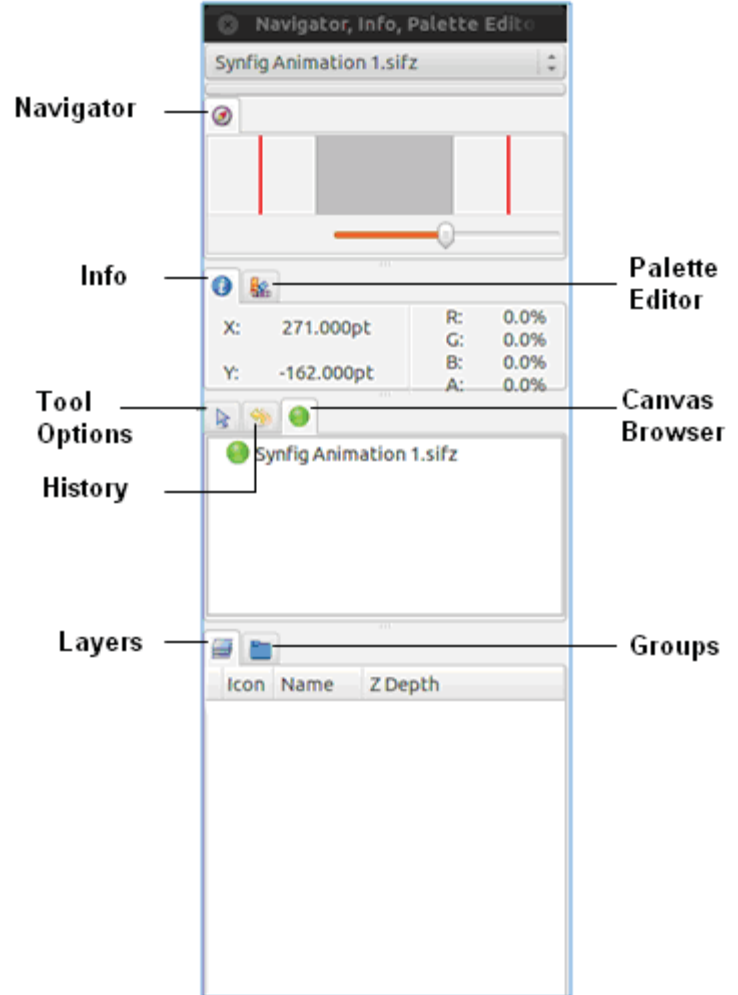
કેનવાસની ઉપરના ડાબી બાજુના ખૂણા પર એક કાળું અને નાનું ત્રિકોણાકાર બટન જોવા મળે છે, જેને કેરેટ (caret)  તરીકે ઓળખવામાં આવે છે. આ બટન પર ક્લિક કરવાથી કેનવાસ મેનુ ખૂલે છે, જેના દ્વારા સીન્ફિગની અધિકાંશ સુવિધાઓનો ઉપયોગ કરી શકાય છે. સામાન્ય રીતે, દરેક વિનિયોગમાં મેનુની ગોઠવણ સ્કીન અથવા કાર્યવિસ્તારના ઉપરના ભાગમાં જોવા મળે છે.

જો યોજનાની ગોઠવણ શૂન્યેતર સમય માટે રાખેલી હોય, તો સ્કીન ઉપર સમયરેખા (timeline) પણ જોઈ શકાય છે. ડાબી બાજુના વિભાગમાં વર્તમાન ફેમનો ક્રમ જાણી શકાય છે અને જમણી બાજુ એનિમેશનની સ્થિતિને બદલવા માટે તથા કી-ફેમને લોક/અનલોક કરવા માટેનાં બે બટન આપવામાં આવેલા છે.

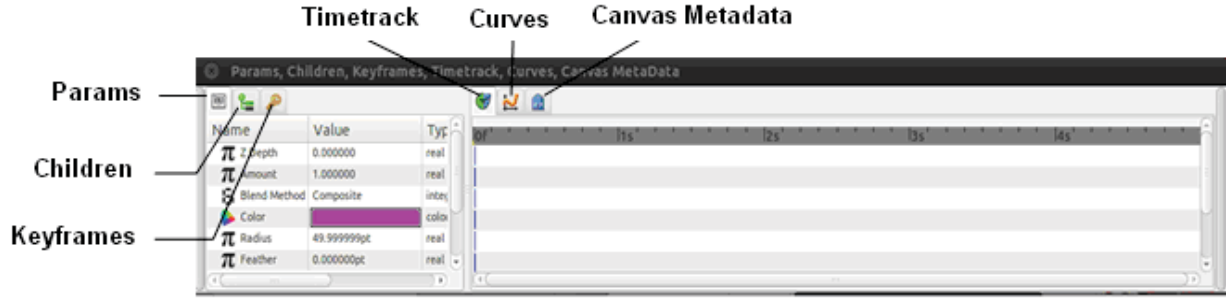
પેનલ (Panels)

સીન્ફિગમાં લેયર, પેરામીટર, હિસ્ટ્રી વગેરે જેવી જુદા-જુદા પ્રકારની પેનલ આવેલી છે. યોજનાના ચોક્કસ ઘટકો માટેની માહિતી અને ટૂલ્સ પેનલમાં આપવામાં આવે છે. કેટલીક પેનલ દ્વારા આ ઘટકોને બદલી શકાય છે.

નીચેની તરફ જમણી બાજુ આવેલી બે વિન્ડો અનુકૂળ ગોઠવણી કરી શકાય તે મુજબની (customizable) ડોક ડાયલોગ (dock dialogs) છે. ડોક ડાયલોગમાં પેનલનું જૂથ સમક્ષિતિજ (horizontal) કે શિરોલંબ (vertical) ગોઠવાયેલું હોય છે. કેટલીક પેનલ ડોક ડાયલોગમાં એક જ સ્થાનને વહેંચે છે તથા તેના ટેબ પર ક્લિક કરી કઈ પેનલ દર્શાવવી છે તે પસંદ કરી શકાય છે. ડોક ડાયલોગમાં આપેલી પેનલને ડ્રેગ કરી ઈચ્છિત સ્થાન પર તેની ફેર-ગોઠવણી કરી શકાય છે. જો અકસ્માતે કોઈ પેનલ બંધ થઈ જાય તો ટૂલબોક્સમાંથી File → Panels વિકલ્પ દ્વારા જરૂરી પેનલ ફરી ખોલી શકાય છે. ત્યાર બાદ પેનલને ડ્રેગ કરી ડોક ડાયલોગમાં ઈચ્છિત સ્થાને ગોઠવી શકાય છે. આકૃતિ 2.5 અને આકૃતિ 2.6માં સમક્ષિતિજ અને શિરોલંબ સ્વરૂપે ગોઠવાયેલી પેનલ દર્શાવી છે.



આકૃતિ 2.5 : શિરોલંબ (vertically) ગોઠવાયેલ પેનલ



આકૃતિ 2.6 : સમક્ષિતિજ (horizontally) ગોઠવાયેલ પેનલ

હવે કેટલીક મહત્વની પેનલનાં કાર્યો વિશે ચર્ચા કરીએ.

લેયર્સ પેનલ (Layers Panel) :

તે હાલના કાર્યરત કેનવાસનું લેયર દર્શાવે છે. આ લેયરમાં ફેરફાર પણ કરી શકાય છે.

પેરામ્સ પેનલ (Params Panel) :

તે વર્તમાન લેયરના પ્રાયલ (parameter) દર્શાવે છે. જ્યારે એકથી વધુ લેયરની પસંદગી કરવામાં આવી હોય ત્યારે તમામ લેયરના સામાન્ય પ્રાયલ દર્શાવવામાં આવે છે.

ટૂલ-ઓપ્શન્સ પેનલ (Tool Options Panel) :

તે હાલમાં પસંદ કરવામાં આવેલા ટૂલના વિકલ્પો દર્શાવે છે.

નેવિગેટર (Navigator) :

તે હાલમાં પસંદ કરેલા કેનવાસનો દેખાવ લઘુ સ્વરૂપે (thumbnail) દર્શાવે છે. તેનો દેખાવ વિસ્તારી (zoom) ઈચ્છિત ભાગ પર ધ્યાન કેન્દ્રિત કરી શકાય છે.

હિસ્ટ્રી પેનલ (History Panel) :

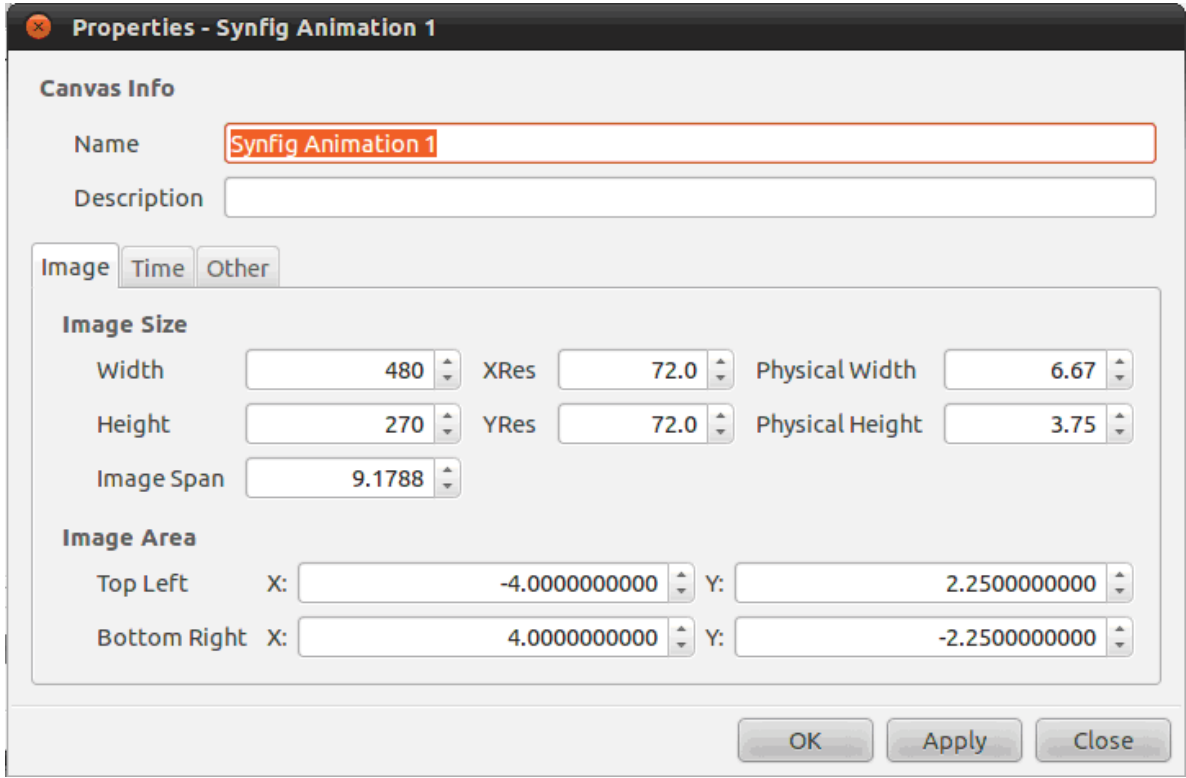
ફાઈલમાં સુધારા કરતી વખતે ઉપયોગમાં લેવામાં આવેલી તમામ ક્રિયાઓની નોંધ અહીં રાખવામાં આવે છે. તમામ ક્રિયાઓની યાદીમાં ગયા વગર ચેકબોક્સ કોલમની મદદથી પસંદગીની ક્રિયા રદ કરી શકાય છે. કોઈ ક્રિયા કે ક્રિયાના જૂથને તેના ચેકબોક્સ પર માત્ર ક્લિક કરીને અન-ડુ (undo) કે રી-ડુ (redo) કરી શકાય છે.

તદુપરાંત અન્ય અનેક પેનલ આપવામાં આવેલી છે, જેના વિશે માહિતી મેળવવા માટે દરેક આઈકન પર માત્ર માઉસ-પોઈન્ટર ગોઠવવાથી તેના કાર્યને દર્શાવતી ટૂલટીપ જોવા મળે છે.

નવી ફાઈલ બનાવવી (Creating New File)

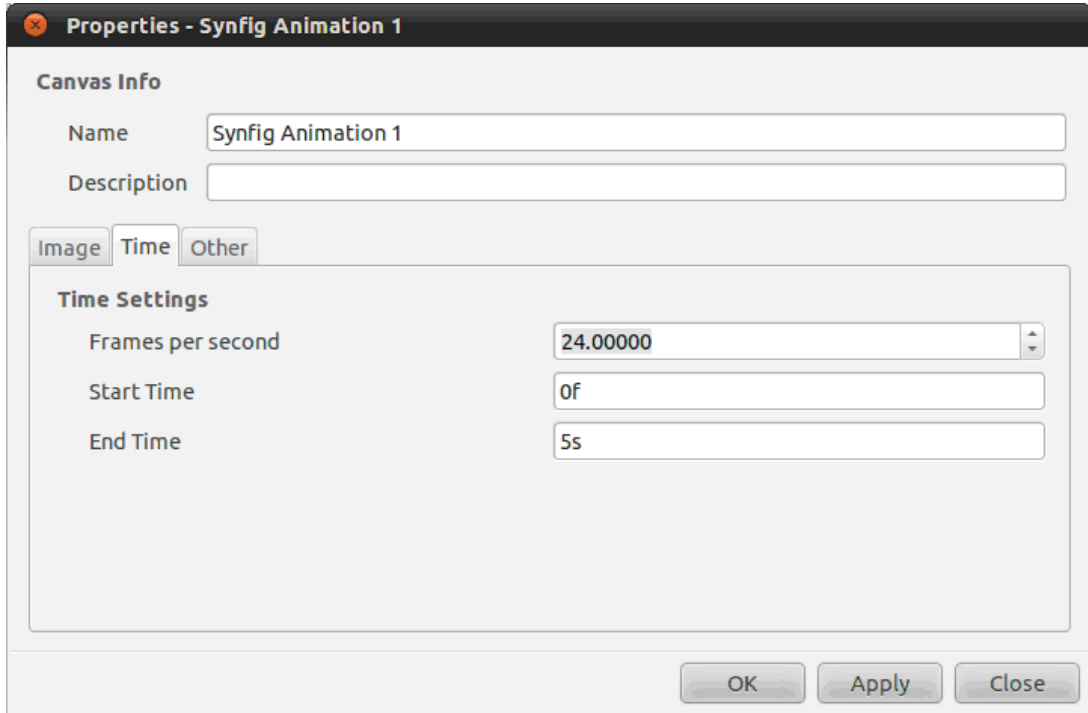
હવે આપણે નવી ફાઈલ બનાવી તેની લાક્ષણિકતાઓ ગોઠવીએ. નવી ફાઈલની રચના કરવા માટે નીચે જણાવેલાં પગલાંને અનુસરો :

- Toolbox → New file વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે, જ્યારે સીન્ડ્રિંગ શરૂ કરવામાં આવે છે, ત્યારે આપોઆપ નવી ફાઈલની રચના કરવામાં આવે છે.
- Caret → Edit → Properties પસંદ કરો. આકૃતિ 2.7માં દર્શાવ્યા પ્રમાણેનું ડાયલોગબોક્સ ખૂલશે.



આકૃતિ 2.7 : File properties ડાયલોગબોક્સ

- અહીં ફાઇલનું નામ આપી એનિમેશનની સમજૂતી ઉમેરી શકાય છે. વિન્ડોમાં ત્રણ ટેબ જોવા મળે છે : *Image*, *Time* અને *Other*. આકૃતિ 2.7માં દર્શાવ્યા મુજબ *Image* ટેબનો ઉપયોગ કરી ચિત્રનું કદ અને વિસ્તાર ગોઠવી શકાય છે. *Time* ટેબના ઉપયોગ દ્વારા આરંભસમય, સમાપ્તિસમય અને દર સેકન્ડે પસાર થતી ફ્રેમની સંખ્યા (frame per second) નિયંત્રિત કરી શકાય છે. આકૃતિ 2.8માં પૂર્વનિર્ધારિત ગોઠવણો દર્શાવી છે. આપણે આ લાક્ષણિકતાઓ વિશે આગળ ઉપર વધુ અભ્યાસ કરીશું.



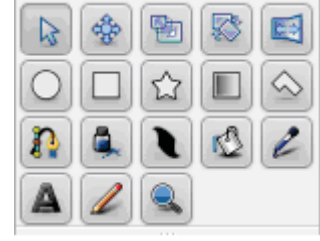
આકૃતિ 2.8 : Time ટેબ પસંદ કરેલ હોય ત્યારે File properties ડાયલોગબોક્સ

- OK બટન પર ક્લિક કરવાથી ફેરફારોનો સંગ્રહ કરવામાં આવશે.
- ફાઇલની રચના કરી તેની લાક્ષણિકતાઓ ગોઠવી લીધા બાદ ચિત્ર (artwork) અને એનિમેશન બનાવવાનું શરૂ કરી શકાય છે.

હવે, ઓબ્જેક્ટની રચના કરી તેમાં ફેરફાર કરવા માટે ઉપયોગી એવાં સાધનો (ટૂલ્સ) વિશે માહિતી મેળવીએ.

ટૂલ્સ (Tools)

મલ્ટિમીડિયા યોજના અંતર્ગત ચિત્રો (artwork) અને એનિમેશનની રચના કરવા માટે ટૂલ્સ મદદરૂપ બને છે. ઓબ્જેક્ટ બનાવવા અને તેમાં ફેરફાર કરવા માટે સીન્કિંગમાં ઘણાં ટૂલ્સ ઉપલબ્ધ છે. આકૃતિ 2.9 ટૂલ્સ દર્શાવે છે. દરેક ટૂલને પોતાનાં વિકલ્પો અને પ્રાયલોનું જૂથ હોય છે. ટૂલ્સના વિકલ્પોને Tool options પેનલમાં દર્શાવવામાં આવે છે અને પ્રાયલોને (Params-Parameter) પેનલમાં દર્શાવાય છે. જ્યારે ચોક્કસ ટૂલ પસંદ કરવામાં આવે ત્યારે, તે ટૂલ માટે ઉપલબ્ધ વિકલ્પોને Tool options પેનલમાં દર્શાવવામાં આવે છે.



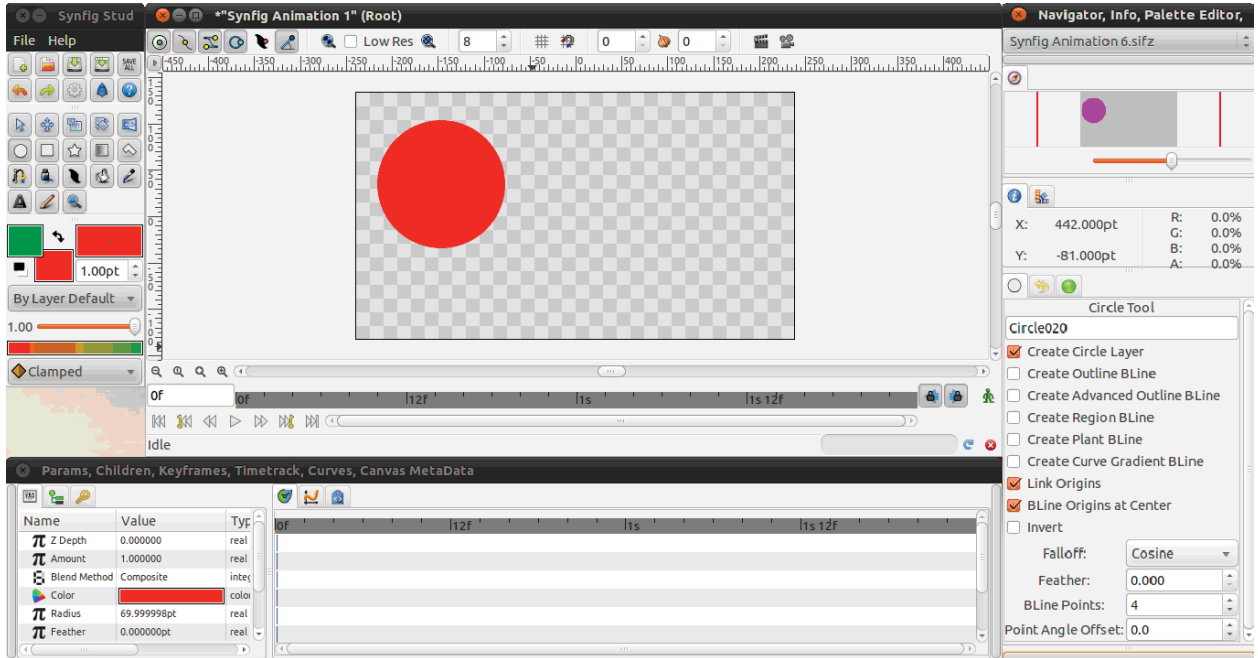
આકૃતિ 2.9 : ચિત્રની રચના અને ફેરફાર કરવા માટેનાં ટૂલ્સ

સૌપ્રથમ આપણે ઓબ્જેક્ટની રચના કરવા માટે ઉપયોગી ટૂલ્સની ચર્ચા કરીએ.

સર્કલ ટૂલ (Circle Tool)

વર્તુળ માટેના નવા લેયરની રચના કરવા માટે સર્કલ ટૂલનો ઉપયોગ કરવામાં આવે છે. સર્કલ લેયર વર્તુળને રજૂ કરે છે. વર્તુળને બે અગત્યનાં પ્રાયલો છે : કેન્દ્રબિન્દુ અને ત્રિજ્યા. પેરામીટર પેનલમાં આ પ્રાયલો જોઈ શકાય છે. ચિત્ર દોર્યા બાદ કોઈ પણ પ્રાયલને બદલી પણ શકાય છે.

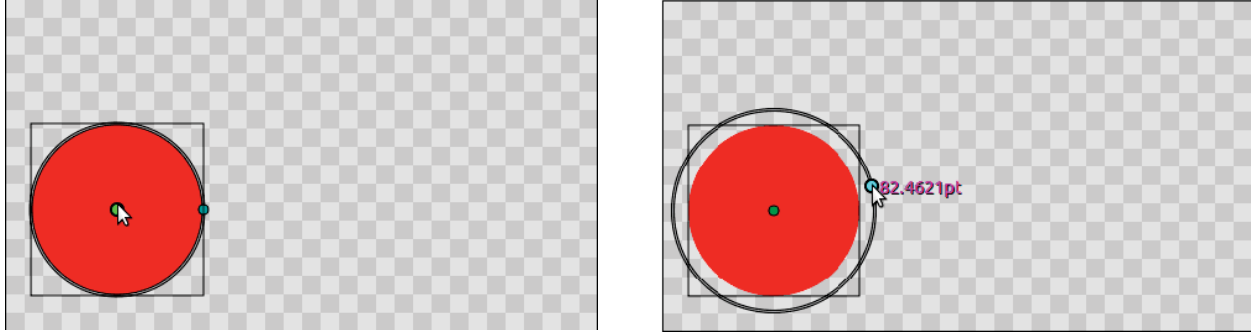
વર્તુળ દોરવા માટે સર્કલ ટૂલ પસંદ કરો. આકૃતિ 2.10માં દર્શાવ્યા પ્રમાણે કાર્યક્ષેત્રમાં વર્તુળના કેન્દ્ર માટે ક્લિક કરી ત્રિજ્યા પસંદ કરવા માટે માઉસને ડ્રેગ કરો.



આકૃતિ 2.10 : સર્કલ ટૂલનો ઉપયોગ

Tool options પેનલનો ઉપયોગ કરી સર્કલ લેયરનું નામ બદલી શકાય છે તથા ચેકબોક્સનો ઉપયોગ કરી સીમારેખા (outline) માટેનું લેયર ઉમેરી શકાય છે, જે વર્તુળ ફરતે સીમારેખા દર્શાવશે. આકૃતિ 2.10માં માત્ર *Create Circle Layer* પસંદ કરવામાં આવ્યું છે. જો *Create Outline Bline* વિકલ્પ પસંદ કરવામાં આવે તો, વર્તુળની રચના સીમારેખા સહિત કરવામાં આવે છે.

વર્તુળની ત્રિજ્યા વધારવી કે વર્તુળને ખસેડવું - જેવા ફેરફારો કરવા માટે આકૃતિ 2.11માં દર્શાવ્યા મુજબ 'ડક' (duck) પર ક્લિક કરવામાં આવે છે, જો ડક દર્શ્યમાન ન હોય, તો Transform ટૂલ પર ક્લિક કરવાથી ડક્સ જોઈ શકાય છે. તમને આશ્ચર્ય થતું હશે કે આ 'ડક' શું છે ?



આકૃતિ 2.11 : વર્તુળ માટેનાં ડક્સ

'ડક' એ લેયરનાં કેટલાંક પ્રાયલોનું નિયંત્રણ કરવા માટેના હાથા (હેન્ડલ) છે. ઉદાહરણ તરીકે, વર્તુળને બે ડક્સ હોય છે : એક, વર્તુળના કેન્દ્રનું સ્થાન નિયંત્રિત કરવા માટેનું ડક. અને બીજું, તેની ત્રિજ્યાનું નિયંત્રણ કરવા માટેનું ડક. આકૃતિ 2.11માં લીલા અને ભૂરા રંગનાં ડક જોઈ શકાય છે. ડક જુદા-જુદા રંગોમાં દર્શાવાય છે અને દરેક રંગનો નિશ્ચિત અર્થ હોય છે. પસંદ કરેલ ઓબ્જેક્ટમાં ડકનો રંગ નીચે જણાવેલ ફેરફાર સૂચવે છે :

- લીલો રંગ - ઓબ્જેક્ટનું સ્થાન
- ભૂરો રંગ - વર્તુળની ત્રિજ્યા
- કેસરી રંગ - શિરોબિંદુ
- પીળો રંગ - વક્ર
- ઘેરો ભૂરો રંગ - સ્ટારમાં ખૂણો બદલવા માટે

આકૃતિ 2.11માં લીલા ડકનો ઉપયોગ વર્તુળનું સ્થાન બદલવા માટે તથા ભૂરા ડકનો ઉપયોગ વર્તુળની ત્રિજ્યા બદલવા માટે કરવામાં આવ્યો છે. કાર્યક્ષેત્રમાં પસંદ કરેલા તમામ લેયર માટે ડક દર્શાવવામાં આવે છે. પ્રાયલોને બદલવા માટે ડકને ડ્રેગ કરી શકાય છે. વધુ ઝીણવટપૂર્વકના નિયંત્રણ માટે કર્સરનો ઉપયોગ પણ કરી શકાય છે. શિફ્ટ-કી દબાવીને ડ્રેગ કરવાથી હલનચલનને માત્ર સમક્ષિતિજ કે શિરોલંબ દિશામાં મર્યાદિત કરી શકાય છે.

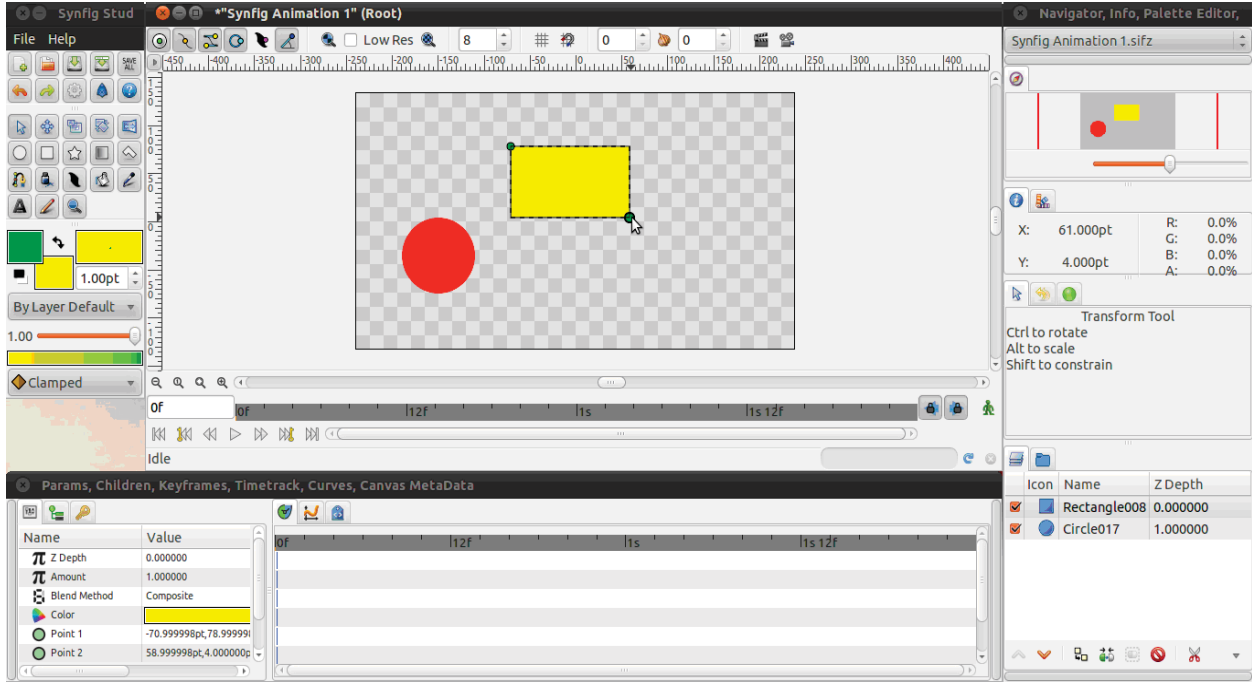
કેટલીક વાર એક પ્રકારનું ડક અન્ય પ્રકારનાં ડકના કાર્યમાં અવરોધરૂપ બને છે. આકૃતિ 2.12માં દર્શાવ્યા મુજબ કેનવાસ વિન્ડોની ઉપરના ભાગમાં આવેલા વિકલ્પ દ્વારા ડકનો દેખાવ દર્શ્યમાન કે અદર્શ્ય બનાવી શકાય છે.



આકૃતિ 2.12 : ડકનો દેખાવ નિયંત્રિત કરવો

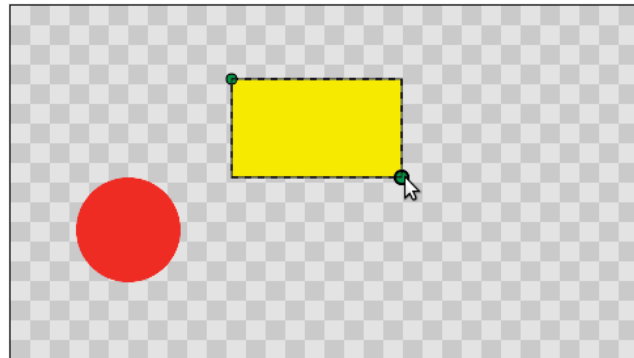
રેક્ટેન્ગલ ટૂલ (Rectangle Tool)

રેક્ટેન્ગલ ટૂલનો ઉપયોગ પણ સર્કલ ટૂલની જેમ જ કરી શકાય છે. તેના દ્વારા લંબચોરસ માટેના લેયરની રચના કરી શકાય છે. લંબચોરસ આકાર દોરવા માટે રેક્ટેન્ગલ ટૂલ પસંદ કરી આકૃતિ 2.13માં દર્શાવ્યા મુજબ લંબચોરસના એક ખૂણાની પસંદગી કરવા માટે ક્લિક કરવામાં આવે છે અને તેની સામેના ખૂણા સુધી માઉસ-પોઇન્ટરને ડ્રેગ કરવામાં આવે છે.



આકૃતિ 2.13 : રેક્ટેન્ગલ ટૂલનો ઉપયોગ

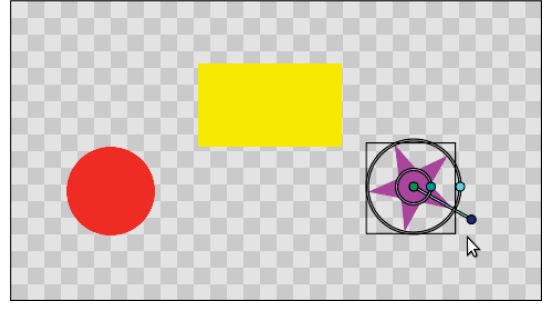
Tool options પેનલમાં રેક્ટેન્ગલ ટૂલને સંબંધિત વિકલ્પો જોઈ શકાશે. સાથે-સાથે પેરામીટર પેનલમાં પ્રાયલો દર્શાવવામાં આવશે. કાર્યક્ષેત્રમાં હવે બે ઓબ્જેક્ટ હોવાથી પેરામીટર પેનલમાં વર્તુળ અને લંબચોરસ બંનેનાં સામાન્ય પ્રાયલોને દર્શાવવામાં આવશે. આ રીતે એક જ પ્રાયલને બદલીને તમામ પસંદ કરેલા ઓબ્જેક્ટમાં પણ ફેરફાર કરી શકાય છે. માત્ર લંબચોરસનાં પ્રાયલ જોવા માટે લેયર પેનલમાં લંબચોરસના લેયરને પસંદ કરવું જોઈએ. આમ કરવાથી હવે માત્ર લંબચોરસનાં પ્રાયલ દર્શાવવામાં આવશે. લંબચોરસમાં ફેરફાર કરવા માટે તેના ડકને પસંદ કરી ડ્રેગ કરી શકાય છે. આકૃતિ 2.14માં રેક્ટેન્ગલ ટૂલ માટેનાં ડકસ દર્શાવ્યા છે.



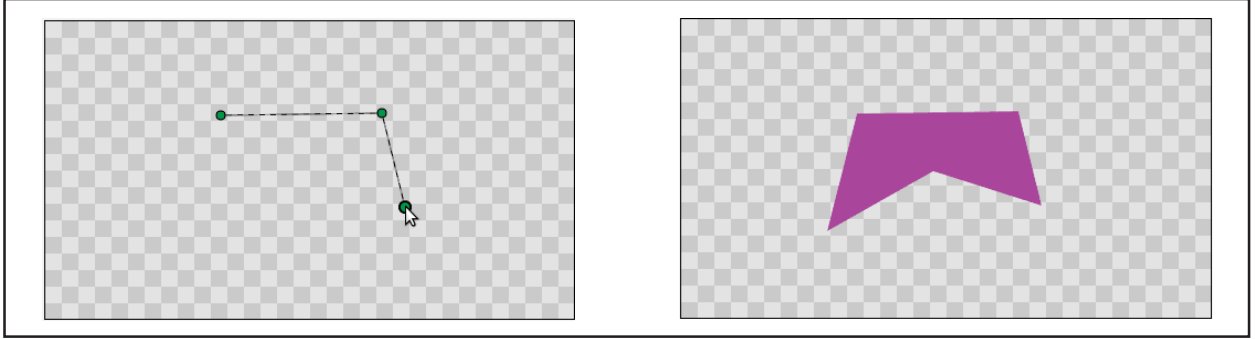
આકૃતિ 2.14 : રેક્ટેન્ગલ ટૂલનાં ડકસ

સ્ટાર અને પોલિગોન ટૂલ (Star and Polygon Tool)

આ જ રીતે, સ્ટાર અને બહુકોણ માટેનાં લેયરની રચના કરવા માટે સ્ટાર અને પોલિગોન ટૂલનો ઉપયોગ કરી શકાય છે. માત્ર તેમના સુધારા કરતી વખતે તફાવત જણાય છે. દરેક આકાર માટે જુદાં-જુદાં ડક આપવામાં આવ્યાં છે. આકૃતિ 2.15 (a) સ્ટારનાં ટૂલનો ઉપયોગ અને તેને સુધારવા માટેનાં ડકને દર્શાવે છે. તફાવત જોવા માટે તેમાં ફેરફાર કરો. આકૃતિ 2.15 (b) પોલિગોન ટૂલ વડે કેવી રીતે બહુકોણની રચના કરવી તે દર્શાવે છે. બહુકોણને ગમે તેટલાં બિંદુઓનો ઉપયોગ કરી રચી શકાય છે અને છેલ્લે પ્રથમ બિંદુ પર ક્લિક કરી તેને પૂર્ણ કરી શકાય છે.



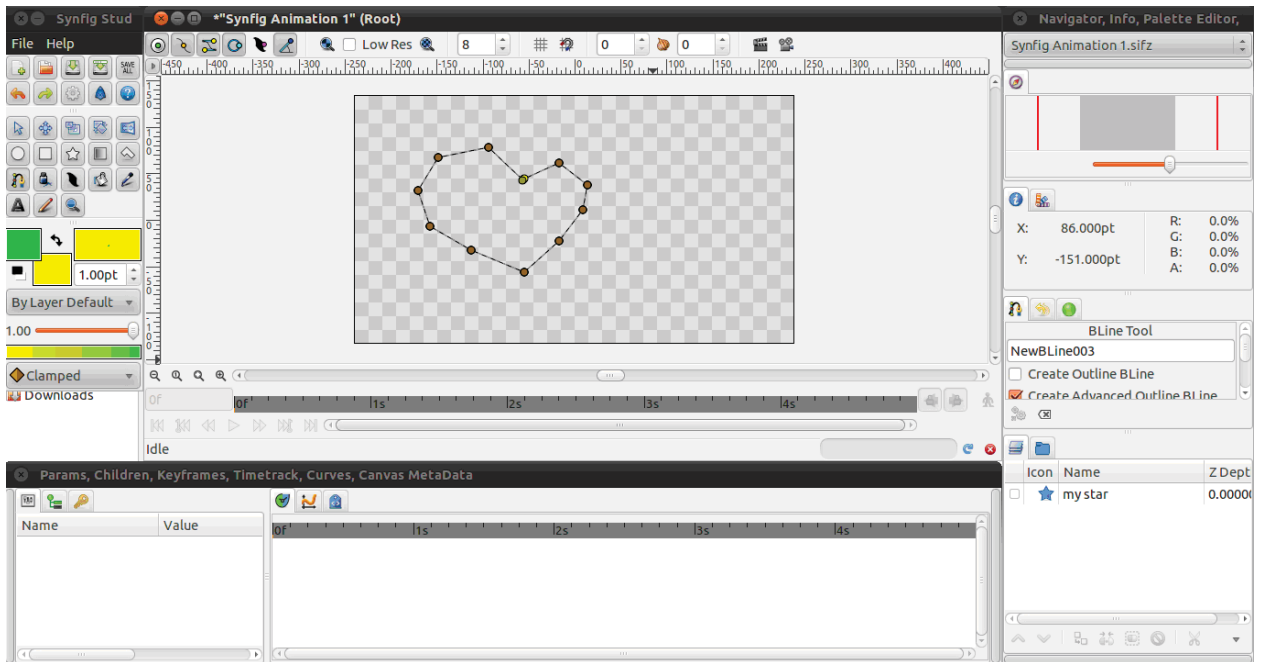
આકૃતિ 2.15(a) : સ્ટાર ટૂલનો ઉપયોગ



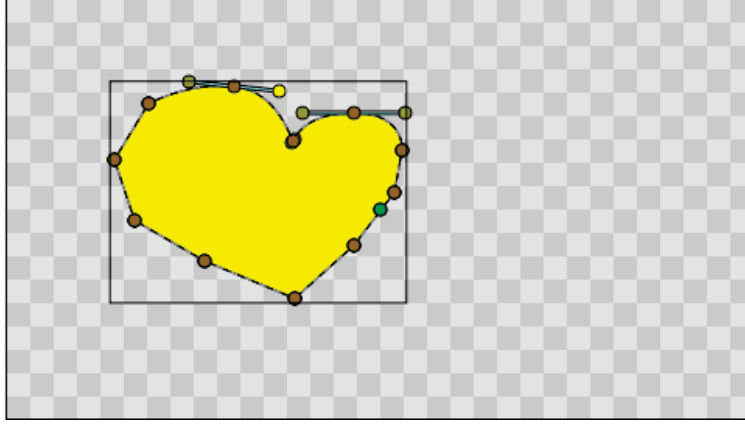
આકૃતિ 2.15(b) : પોલિગોન ટૂલનો ઉપયોગ

બી લાઇન ટૂલ (B Line Tool)

ગમે તેટલી સંખ્યાનાં બિંદુઓ અને વક્રો સાથેનો ઓબ્જેક્ટ બનાવવા માટે બી લાઇન ટૂલનો ઉપયોગ કરવામાં આવે છે. આ માટે બી લાઇન ટૂલ પર ક્લિક કરી બિંદુઓ ઉમેરવાનું શરૂ કરો. આકૃતિ 2.16 (a)માં બી લાઇન ટૂલ વડે ઉમેરેલું ચિત્ર જોઈ શકાય છે. ઓબ્જેક્ટને પૂર્ણ કરવા માટે છેલ્લા બિંદુ પર રાઇટ ક્લિક કરવામાં આવે છે. ચિત્રને ત્યાર બાદ ડકની મદદથી સુધારી શકાય છે. બી લાઇન ટૂલ દ્વારા દોરવામાં આવેલા આકાર માટેનાં ડક્સ આકૃતિ 2.16 (b)માં દર્શાવ્યાં છે.



આકૃતિ 2.16(a) : બી લાઇન ટૂલ

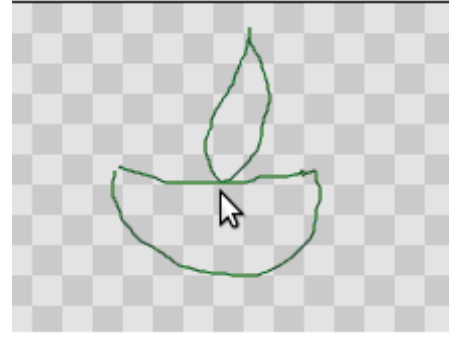


આકૃતિ 2.16 (b) : દોરવામાં આવેલો આકાર-ડક્સ સાથે

ડ્રો ટૂલ (Draw Tool)

ચિત્ર દોરવા માટેના સૌથી વધુ સામાન્ય એવા ડ્રો ટૂલ વિશે જાણકારી હોવી ખૂબ જરૂરી છે. આ ટૂલનો ઉપયોગ કરી માઉસની મદદથી ચિત્ર દોરી શકાય છે. ડ્રો ટૂલનો ઉપયોગ આકૃતિ 2.17માં દર્શાવ્યો છે.

ચિત્ર દોરવા માટેનાં કેટલાંક ટૂલ વિશે ચર્ચા કર્યા બાદ અન્ય ટૂલ વિશે માહિતી મેળવીએ.



આકૃતિ 2.17 : ડ્રો ટૂલ

ફિલ ટૂલ (Fill Tool)

ઓબ્જેક્ટનો રંગ બદલવા માટે ફિલ ટૂલનો ઉપયોગ કરવામાં આવે છે. ઓબ્જેક્ટમાં રંગ ભરવા માટે કલર પેલેટમાંથી રંગ પસંદ કરો. ફિલ ટૂલને પસંદ કરી ઓબ્જેક્ટ પર ક્લિક કરો. આ ટૂલ નીચે જણાવેલ લેયરને અસરકર્તા છે :

- વર્તુળ (Circle)નું લેયર
- લંબચોરસ (Rectangle)નું લેયર
- બહુકોણ (Polygon)નું લેયર
- સ્ટાર (Star)નું લેયર
- વિસ્તાર (Region)નું લેયર
- સીમારેખા (Outline)નું લેયર
- ચેકરબોર્ડ(Checkerboard)નું લેયર

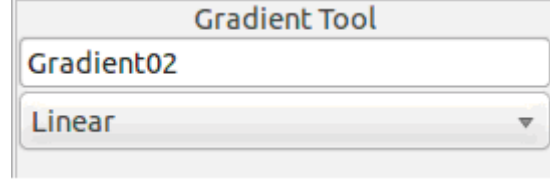
અન્ય લેયર આ ટૂલ દ્વારા અસર પામતાં નથી, કારણકે દૃશ્યમાન વિસ્તારને આવરી લેવા તે ગ્રેડિયન્ટ (gradient)નો ઉપયોગ કરે છે.

આઈડ્રોપ ટૂલ (Eyedrop Tool)

કાર્યક્ષેત્રમાંથી રંગ પસંદ કરવા માટેની સુવિધા આઈડ્રોપ ટૂલ પૂરી પાડે છે. આઈડ્રોપ ટૂલ પસંદ કરી કાર્યક્ષેત્રના કોઈ પણ ભાગ પર ક્લિક કરવાથી ત્યાં આવેલ રંગને વર્તમાન ફોરગ્રાઉન્ડ (foreground) રંગ તરીકે પસંદ કરવામાં આવશે.

ગ્રેડિયન્ટ ટૂલ (Gradient Tool)

ઑબ્જેક્ટમાં બે કે વધુ રંગોને એક્સરખી સંક્રમિત (transition) અસર સાથે પૂરવા માટે ગ્રેડિયન્ટ ટૂલ ઉપયોગી છે. જ્યારે ગ્રેડિયન્ટ ટૂલ પસંદ કરવામાં આવે છે, ત્યારે આકૃતિ 2.18માં દર્શાવ્યા મુજબ તેના માટેના વિકલ્પો દર્શાવવામાં આવે છે.

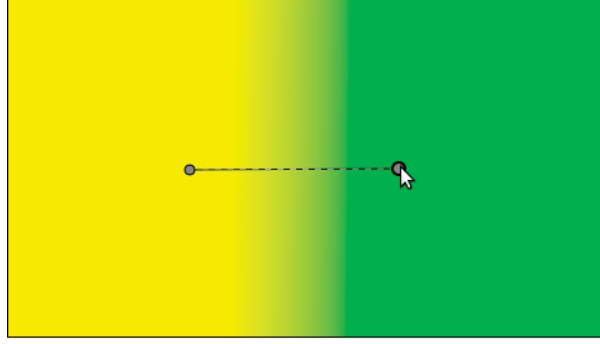


આકૃતિ 2.18 : ગ્રેડિયન્ટ ટૂલના વિકલ્પો

તે લેયરનું નામ આપવાની તથા ગ્રેડિયન્ટનો પ્રકાર પસંદ કરવાની સુવિધા આપે છે.

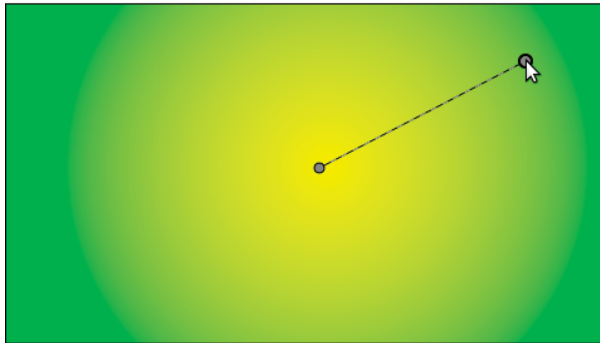
ગ્રેડિયન્ટ ચાર પ્રકારનાં છે : લિનિયર (Linear), રેડિયલ (Radial), કોનિકલ (Conical) અને સ્પાઈરલ (Spiral)

લિનિયર (Linear) : આ પ્રકારનું ગ્રેડિયન્ટ સીધી લીટીમાં ટ્રાન્ઝિશન દર્શાવે છે. ફોરગ્રાઉન્ડ અને બેકગ્રાઉન્ડ રંગ પસંદ કરો. ગ્રેડિયન્ટ શરૂ કરવાનું હોય તે સ્થાન પર ક્લિક કરી ગ્રેડિયન્ટ પૂરું કરવાનું હોય તે સ્થાન સુધી માઉસ ડ્રેગ કરો. માઉસની દિશાને લંબરૂપ (perpendicular) ગ્રેડિયન્ટની રચના કરવામાં આવશે. સામાન્ય ટૂલના ઉપયોગથી અંતિમ બિંદુને ખસેડી ગ્રેડિયન્ટમાં સુધારો કરી શકાય છે. આકૃતિ 2.19માં લિનિયર ગ્રેડિયન્ટની અસર દર્શાવી છે.



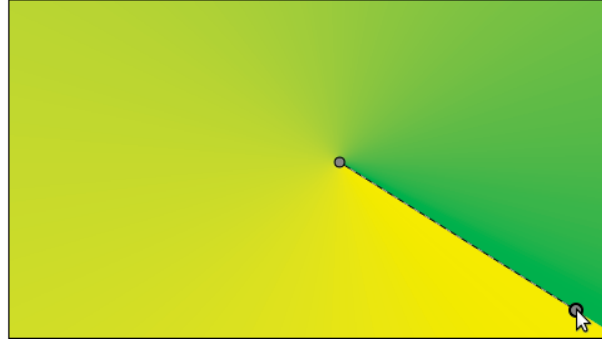
આકૃતિ 2.19 : લિનિયર ગ્રેડિયન્ટ

રેડિયલ (Radial) : આ પ્રકારનું ગ્રેડિયન્ટ વર્તુળાકાર રંગો દર્શાવે છે, જેમાં આ વર્તુળોનાં કેન્દ્રબિંદુએ ટ્રાન્ઝિશનની અસર ગોઠવવામાં આવે છે. વર્તુળનું કેન્દ્રબિંદુ રાખવું હોય તે સ્થાને ક્લિક કરી ગ્રેડિયન્ટની ત્રિજ્યા પૂરી પાડવા માટે માઉસ ડ્રેગ કરવામાં આવે છે. આકૃતિ 2.20 રેડિયલ ગ્રેડિયન્ટની અસર દર્શાવે છે.



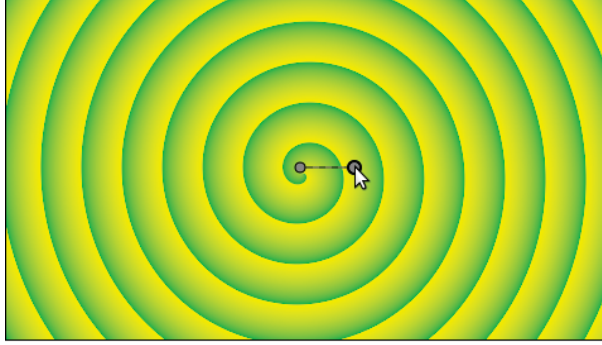
આકૃતિ 2.20 : રેડિયલ ગ્રેડિયન્ટ

કોનિકલ (Conical) : આ પ્રકારના ગ્રેડિયન્ટને શંકુ (cone) આકારની ટોચ પરથી નીચેની તરફ દર્શાવવામાં આવે છે. ગ્રેડિયન્ટની અસર વર્તુળના વક્રભાગથી શરૂ થઈ તમામ દિશાઓમાં વિસ્તરે છે કેન્દ્રબિંદુ નક્કી કરવા માટે ક્લિક કર્યા બાદ ફોરગ્રાઉન્ડ અને બેકગ્રાઉન્ડ રંગ કઈ તરફ પ્રસારવા જોઈએ, તે નક્કી કરવા માટે માઉસને ડ્રેગ કરવામાં આવે છે. કેન્દ્રના અંત્યબિંદુ (centre endpoint) ગ્રેડિયન્ટના કેન્દ્રને ગોઠવે છે અને અન્ય અંત્યબિંદુઓ ગ્રેડિયન્ટની દિશા નક્કી કરે છે. આકૃતિ 2.21માં કોનિકલ ગ્રેડિયન્ટની અસર દર્શાવી છે.



આકૃતિ 2.21 : કોનિકલ ગ્રેડિયન્ટ

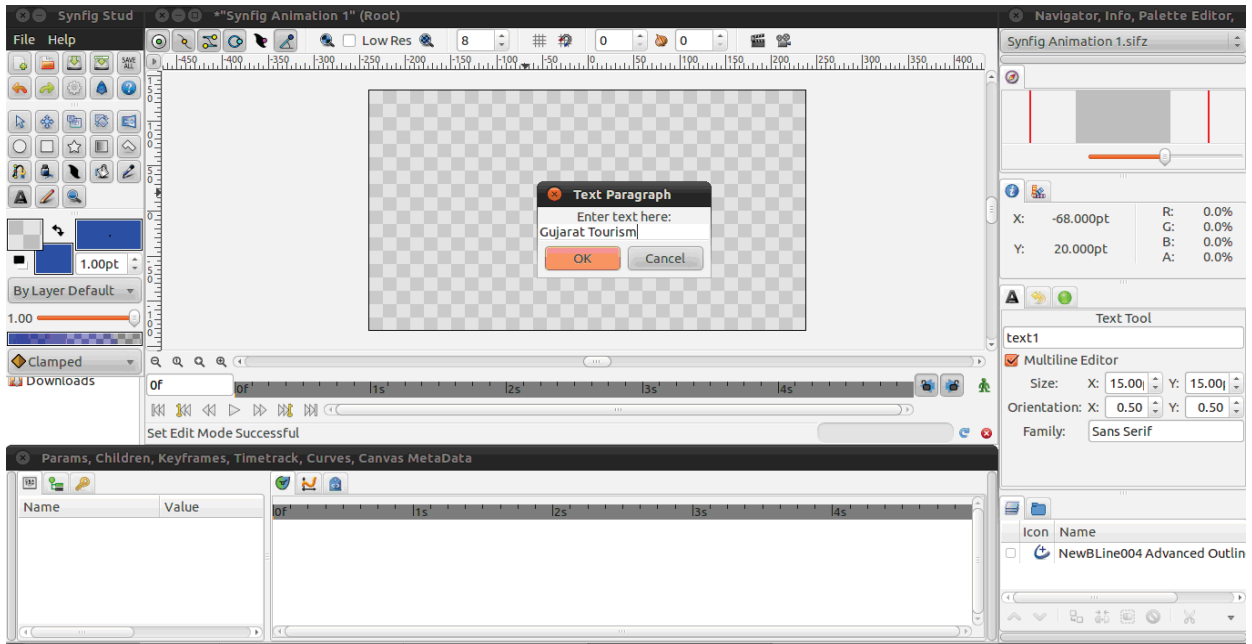
સ્પાઈરલ (Spiral) : આ પ્રકાર ગૂંચળાકાર (સ્પાઈરલ) ગ્રેડિયન્ટ પ્રસ્તુત કરે છે. સ્પાઈરલના કેન્દ્રબિંદુને ગોઠવવા ક્લિક કરી તેની સુગઠિતતા (tightness) નક્કી કરવા માટે માઉસ ટ્રેગ કરવામાં આવે છે આકૃતિ 2.22 સ્પાઈરલ ગ્રેડિયન્ટની અસર દર્શાવે છે.



આકૃતિ 2.22 સ્પાઈરલ ગ્રેડિયન્ટ

ટેક્સ્ટ ટૂલ (Text Tool)

ટેક્સ્ટ લેયરની રચના કરી લખાણ ઉમેરવા માટે ટેક્સ્ટ ટૂલનો ઉપયોગ કરવામાં આવે છે. ટૂલબોક્સમાંથી ટેક્સ્ટ ટૂલના આઈકોનને પસંદ કરી કેનવાસ પર જ્યાં ટેક્સ્ટ લેયરની રચના કરવી હોય તે સ્થાને ક્લિક કરો. આકૃતિ 2.23(a)માં દર્શાવ્યા મુજબ એક ડાયલોગબોક્સ રજૂ કરવામાં આવશે, જેમાં તે લેયરમાં ઉમેરવા માટેનું લખાણ ટાઈપ કરી શકાશે.



આકૃતિ 2.23(a) : ટેક્સ્ટ લેયર ડાયલોગબોક્સ

આકૃતિ 2.23 (b) નવું બનાવેલું ટેક્સ્ટ લેયર તેના ડક્સ સાથે દર્શાવે છે. પેરામીટર પેનલનો ઉપયોગ કરી ઉમેરેલ લખાણને આવશ્યકતાનુસાર બદલી શકાય છે.



આકૃતિ 2.23 (b) : ડક્સ સાથે લખાણ

ટૂલ-ઓપ્શન (Tool option) પેનલ દ્વારા નીચેનાં કાર્યો કરી શકાય છે :

- નવા લેયર માટેનું નામ પસંદ કરવું. જેમ જેમ નવાં લેયર ઉમેરવામાં આવશે તેમ આ ક્રમ વધતો રહેશે.
- લખાણ ઉમેરવા એક (single) અથવા અનેક (multi) લીટીઓ માટેનું એડિટર પસંદ કરવું. જો એક જ લીટીનું એડિટર પસંદ કરવામાં આવેલ હોય, તો લખાણ પૂરું કરવા એન્ટર-કીનો ઉપયોગ કરવામાં આવે છે. જ્યારે અનેક લીટીના એડિટરમાં OK બટન પર ક્લિક કરી લખાણ ઉમેરવાનું કાર્ય પૂરું કરી શકાય છે.
- લખાણનું સમક્ષિતિજ અને શિરોલંબ કદ નક્કી કરવું.
- લખાણનું અનુમાપન (Orientation) નક્કી કરવું. પૂર્વનિર્ધારિત અનુમાપન (0.5, 0.5) હોય છે, માટે ક્લિક કર્યું હોય તેની આજુબાજુમાં લખાણ ગોઠવાશે.
- ફોન્ટ પસંદ કરવા.

ઝૂમ ટૂલ (Zoom Tool)

કાર્યક્ષેત્રમાં કોઈ ઓબ્જેક્ટને નજીક લાવીને અથવા તો દૂર લઈ જઈને જોવા માટે ઝૂમ ટૂલનો ઉપયોગ કરવામાં આવે છે. ઝૂમિંગ પરિણામ પર કોઈ અસર કરતું નથી. ઓબ્જેક્ટને દૂર અથવા નજીક લાવીને માત્ર જોવા માટે તેનો ઉપયોગ કરવામાં આવે છે.

વિડ્થ ટૂલ (Width Tool)

લીટીની પહોળાઈ વધારવા કે ઘટાડવા માટે વિડ્થ ટૂલનો ઉપયોગ કરવામાં આવે છે. પેન્સિલની મદદથી લીટીને પહોળી કરવી કે રબરની મદદથી લીટીને પાતળી કરવા જેવું આ કાર્ય છે. ચોક્કસ લીટીની પહોળાઈ બદલવા માટે સીમારેખાનું લેયર પસંદ કરી માઉસ-પોઇન્ટરને તે લીટી પર રાખી, માઉસનું બટન દબાવી આગળ અને પાછળ ફેરવો. કર્સરને જે સ્થાને ફેરવવામાં આવશે, ત્યાં સીમારેખાની પહોળાઈ વધારવામાં આવશે. કર્સર ફેરવતી વખતે “CTRL” કી દબાવી રાખવાથી પહોળાઈ ઘટાડી શકાશે. લીટીની પહોળાઈને બારીકાઈથી બદલવા આ ટૂલ ઉપયોગી નીવડે છે.

ટ્રાન્સફોર્મ ટૂલ (Transform Tool)

પસંદ કરેલા ઓબ્જેક્ટને ખસેડવાની (move), વર્તુળાકારે ફેરવવાની (rotate) કે કદ બદલવા માટે (scale)ની સુવિધા ટ્રાન્સફોર્મ ટૂલ દ્વારા પૂરી પાડવામાં આવે છે. કેનવાસ વિન્ડો પર આવેલ ઓબ્જેક્ટ પસંદ કરી ટ્રાન્સફોર્મ ટૂલ પર ક્લિક

કરો. જો ઓબ્જેક્ટ અન્ય કોઈ લેયરની પાછળ હોય અને પસંદ થઈ શકે તેમ ન હોય તો લેયર પેનલમાંથી તેના લેયરની પસંદગી કરી શકાશે. આમ કરવાથી ઓબ્જેક્ટ પર તેનાં ડક્સ દર્શાવવામાં આવશે. ડક્સ પસંદ કરી આવશ્યકતા અનુસાર ફેરફાર કરી શકાશે.

સ્મૂથ મૂવ ટૂલ (Smooth Move Tool)

સ્મૂથ મૂવ ટૂલ એ ટ્રાન્સફોર્મ ટૂલ જેવું જ કાર્ય કરતું એક ટૂલ છે, સિવાય કે એકથી વધુ ડક્સ પસંદ કરવામાં આવ્યાં હોય. એકથી વધુ ડક્સ પસંદ કરવા માટે CTRL+A કી દબાવી શકાય અથવા લંબચોરસ આકારે માઉસ-પોઈન્ટરને ડ્રેગ કરી તેના દ્વારા બનાવવામાં આવતા આભાસી લંબચોરસની વચ્ચે આવેલાં તમામ ડક્સ પસંદ કરી શકાય. હવે જો કોઈ પણ ડક્સ બંધાવવામાં આવશે, તો પસંદ કરેલાં તમામ ડક્સ પણ તેની સાથે સ્થાનાંતર પામશે.

સ્કેલ ટૂલ (Scale Tool)

સ્મૂથ મૂવ ટૂલની જેમ જ, અહીં પણ જો એકથી વધુ ડક્સ પસંદ કર્યાં બાદ, કોઈ એક ડક્સને ડ્રેગ કરવામાં આવે, તો અન્ય તમામ ડક્સના સ્કેલ પસંદ કરેલા ડક્સના કેન્દ્રને અનુલક્ષીને બદલાય છે.

રોટેટ ટૂલ (Rotate Tool)

સ્મૂથ મૂવ ટૂલ અને સ્કેલ ટૂલની જેમ, જ્યારે અનેક ડક્સ પસંદ કરી તેમાંથી કોઈ એક ડક્સને ડ્રેગ કરવામાં આવશે, ત્યારે પસંદ કરેલા તમામ ડક્સ જૂથના કેન્દ્રબિંદુને અનુલક્ષીને વર્તુળાકારે ફરશે.

સારાંશ

આ પ્રકરણમાં આપણે ઓથરિંગ ટૂલ્સ અને તેના પ્રકારો વિશે અભ્યાસ કર્યો. સીન્ક્રિંગના કાર્યક્ષેત્રનો પરિચય મેળવ્યો અને નવી ફાઈલ બનાવી તેની લાક્ષણિકતાઓ ગોઠવવા વિશે માહિતી મેળવી. આપણે દરેક પેનલના હેતુની ચર્ચા કરી, દરેક ટૂલની તેના વિકલ્પો સાથે ચર્ચા કરી. આ ટૂલના ઉપયોગ દ્વારા ઓબ્જેક્ટ બનાવી કે સુધારી શકાય, જેનો આગળ ઉપર એનિમેશનની રચનામાં ઉપયોગ થઈ શકશે. હવે આપણે આ ઓબ્જેક્ટને એનિમેશન આપતા શીખીશું. સીન્ક્રિંગમાં ઉપલબ્ધ જુદાં-જુદાં ટૂલ્સનાં કાર્યો વિશે અભ્યાસ કર્યા બાદ આપણે આ ટૂલ્સનો ઉપયોગ કરી એનિમેશનની રચના તરફ આગળ વધીશું.

સ્વાધ્યાય

1. ઓથરિંગ ટૂલ્સ એટલે શું ? જુદાં-જુદાં ઓથરિંગ ટૂલની યાદી બનાવો.
2. ગ્રેડિયન્ટ ટૂલનો હેતુ જણાવો. આ પ્રકરણમાં આપેલા ગ્રેડિયન્ટના ચાર પ્રકાર જણાવો.
3. ટૂલબોક્સ વિન્ડોની ત્રણ પેલેટ ટૂંકમાં સમજાવો.
4. મહત્વની પેનલ અને તેની કાર્યપદ્ધતિની યાદી બનાવો.
5. હેતુલક્ષી પ્રશ્નો
 - (1) નીચેનામાંથી કયા પ્રકારના ઓથરિંગ ટૂલમાં ઘટકોને પુસ્તકનાં પાનાં અથવા કાર્ડના ઢગલા સ્વરૂપે ગોઠવવામાં આવે છે ?
 - (a) કાર્ડ અથવા પૃષ્ઠ આધારિત ટૂલ્સ
 - (b) આઈકન અને ઇવેન્ટ આધારિત ટૂલ્સ
 - (c) સમય આધારિત ટૂલ્સ
 - (d) એનિમેશન ટૂલ્સ

- (2) ઓબ્જેક્ટનો રંગ બદલવા માટે નીચેનામાંથી કયું ટૂલ ઉપયોગી છે ?
- (a) draw (b) fill
(c) circle (d) rectangle
- (3) સીન્કિંગ સ્ટુડિયો નીચેનામાંથી કયા પ્રકારનું ઓથરિંગ ટૂલ છે ?
- (a) સમય આધારિત ઓથરિંગ ટૂલ (b) ઘટના આધારિત ઓથરિંગ ટૂલ
(c) પૃષ્ઠ આધારિત ઓથરિંગ ટૂલ (d) આઈકન આધારિત ઓથરિંગ ટૂલ
- (4) ટૂલબોક્સ વિન્ડો કેટલી પેલેટમાં વહેંચાયેલી છે ?
- (a) બે (b) ત્રણ
(c) ચાર (d) પાંચ
- (5) સીન્કિંગ સ્ટુડિયોમાં બનેલી ફાઈલને કયું એક્સ્ટેન્શન આપવામાં આવે છે ?
- (a) .fiz (b) .sifz
(c) .zif (d) .fis
- (6) કઈ પેનલ પસંદ કરેલ લેયરનાં પ્રાયલ દર્શાવે છે ?
- (a) Layer (b) Params
(c) Timetrack (d) History
- (7) વર્તુળની ત્રિજ્યા બદલવા માટે કયા રંગનું ડક રજૂ કરવામાં આવે છે ?
- (a) લીલા રંગનું (b) પીળા રંગનું
(c) કેસરી રંગનું (d) ભૂરા રંગનું
- (8) ગમે તેટલાં બિંદુઓ અને વક્રોની મદદથી ઓબ્જેક્ટની રચના કરવા માટે કયા ટૂલનો ઉપયોગ કરવામાં આવે છે ?
- (a) bline (b) circle
(c) rectangle (d) star
- (9) ઓબ્જેક્ટમાં બે રંગો વચ્ચે એક્સરખી અસરથી સંક્રમિત (transition) પૂરવા માટે કયા ટૂલનો ઉપયોગ કરવામાં આવે છે ?
- (a) width (b) fill
(c) eyedrop (d) gradient
- (10) નીચેનામાંથી કયું ગ્રેડિયન્ટ સીધી લીટીમાં સરળ ટ્રાન્ઝિશનની રચના કરે છે ?
- (a) Conical (b) Radial
(c) Linear (d) Spiral
- (11) ઓથરવેર અને આઈકનઓથર કયા પ્રકારનાં ટૂલ્સ છે ?
- (a) કાર્ડ અને પૃષ્ઠ આધારિત ટૂલ્સ (b) આઈકન કે ઈવેન્ટ આધારિત ટૂલ્સ
(c) સમય આધારિત ટૂલ્સ (d) એનિમેશન ટૂલ્સ
- (12) પરંપરાગત હસ્તનિર્મિત એનિમેશન બનાવવા માટે નીચેનામાંથી કયું એનિમેશન સોફ્ટવેર ઉપયોગી છે ?
- (a) પેન્સિલ (b) સીન્કિંગ
(c) ફ્લેશ (d) ડાઈરેક્ટર

- (13) વર્તુળના બે અગત્યનાં પ્રાયલ કયાં છે ?
- (a) કેન્દ્ર, વ્યાસ (b) ત્રિજ્યા, વ્યાસ
(c) વ્યાસ, બિંદુ (d) કેન્દ્ર, ત્રિજ્યા
- (14) લેયરનાં કેટલાંક પ્રાયલનું નિયંત્રણ કરવા માટેના હાથાને કયા નામથી ઓળખવામાં આવે છે ?
- (a) point (b) tool
(c) duck (d) panel
- (15) ઓબ્જેક્ટનું સ્થાન બદલવા માટે કયા રંગના ડકનો ઉપયોગ કરવામાં આવે છે ?
- (a) લીલા રંગના (b) ભૂરા રંગના
(c) લાલ રંગના (d) પીળા રંગના
- (16) નીચેનામાંથી કઈ ક્રિયા દ્વારા લેયરની દૃશ્યતા નિયંત્રિત કરી શકાય છે ?
- (a) Opacity (b) Gradient
(c) Interpolation (d) Blend
- (17) ટેક્સ્ટ લેયરની રચના કરી લખાણ ઉમેરવા માટે કયું ટૂલ ઉપયોગી છે ?
- (a) Circle (b) Text
(c) Draw (d) Fill

પ્રાયોગિક સ્વાધ્યાય

1. કેનવાસ પર વર્તુળ, લંબચોરસ અને સ્ટારની રચના કરો. દરેક ઓબ્જેક્ટ માટે જુદા રંગની પસંદગી કરો. ઓબ્જેક્ટને અનુરૂપ લખાણ પણ ઉમેરો.
2. બી લાઈન ટૂલની મદદથી મીણબત્તી દોરો.
3. યોગ્ય ટૂલનો ઉપયોગ કરી પુષ્પ અને મીણબત્તીનું ચિત્ર બનાવો.





આગળના પ્રકરણમાં આપણે સીન્ડિંગના ઈન્ટરફેસ વિશે વાત કરી. મૂળભૂત કલાકૃતિઓ તૈયાર કરીને તેમાં ફેરફાર કરવા માટેનાં જુદાં-જુદાં સાધનોનો ઉપયોગ કેવી રીતે કરવો તે પણ આપણે જોયું. તો ચાલો હવે, ચિત્રને જીવંત બનાવવા ‘એનિમેશન’ (Animation) કેવી રીતે કરવું તે શીખીએ. આપણે અગાઉ ચર્ચા કરી તે પ્રમાણે એનિમેશન એટલે સમયાંતરે ચિત્રના દૃશ્યમાં પરિવર્તન. બીજા શબ્દોમાં કહીએ તો, એનિમેશન નિયત સમયમાં દર્શાવાતી એક પછી એક છબીઓની હારમાળા છે. તો ચાલો, હવે એનિમેશનની કેટલીક મૂળભૂત બાબતોની ચર્ચા કરીએ.

એનિમેશન

અગાઉ ચર્ચા કર્યા મુજબ એનિમેશન એ નિયત સમયમાં દર્શાવાતી એક પછી એક છબીઓની હારમાળા છે, જેમાં ચિત્ર કે આકૃતિને જીવંત લાગે તેવી બનાવવા સાંખ્યિક છબીઓ (ડિજિટલ ઈમેજ)ને એક પછી એક રજૂ કરવામાં આવે છે. એનિમેશનમાં ઉપયોગમાં લેવાની છબીઓને પહેલાં તૈયાર કરવામાં આવે છે અને ત્યાર બાદ તેને ક્રમબદ્ધ રીતે દર્શાવવામાં આવે છે. આ પ્રકારના એનિમેશનને ‘ફ્રેમ બાય ફ્રેમ એનિમેશન’ તરીકે ઓળખવામાં આવે છે. ફ્રેમને દસ્તાવેજમાં સમયની અવધિ રૂપે દર્શાવાય છે. દરેક ફ્રેમમાં સમયાંતરે દર્શાવવા માટેના કેટલાક ઓબ્જેક્ટ હોય છે. આમ, ‘ફ્રેમ બાય ફ્રેમ એનિમેશન’માં દરેક છબીમાં નાનો ફેરફાર સર્જવામાં આવે છે અને પછી ક્રમબદ્ધ રીતે દર્શાવવામાં આવે છે. આ પદ્ધતિ ધીમી અને કંટાળાજનક છે. તેમાં ઘણો સમય અને સંશાધનોની જરૂર પડે છે.

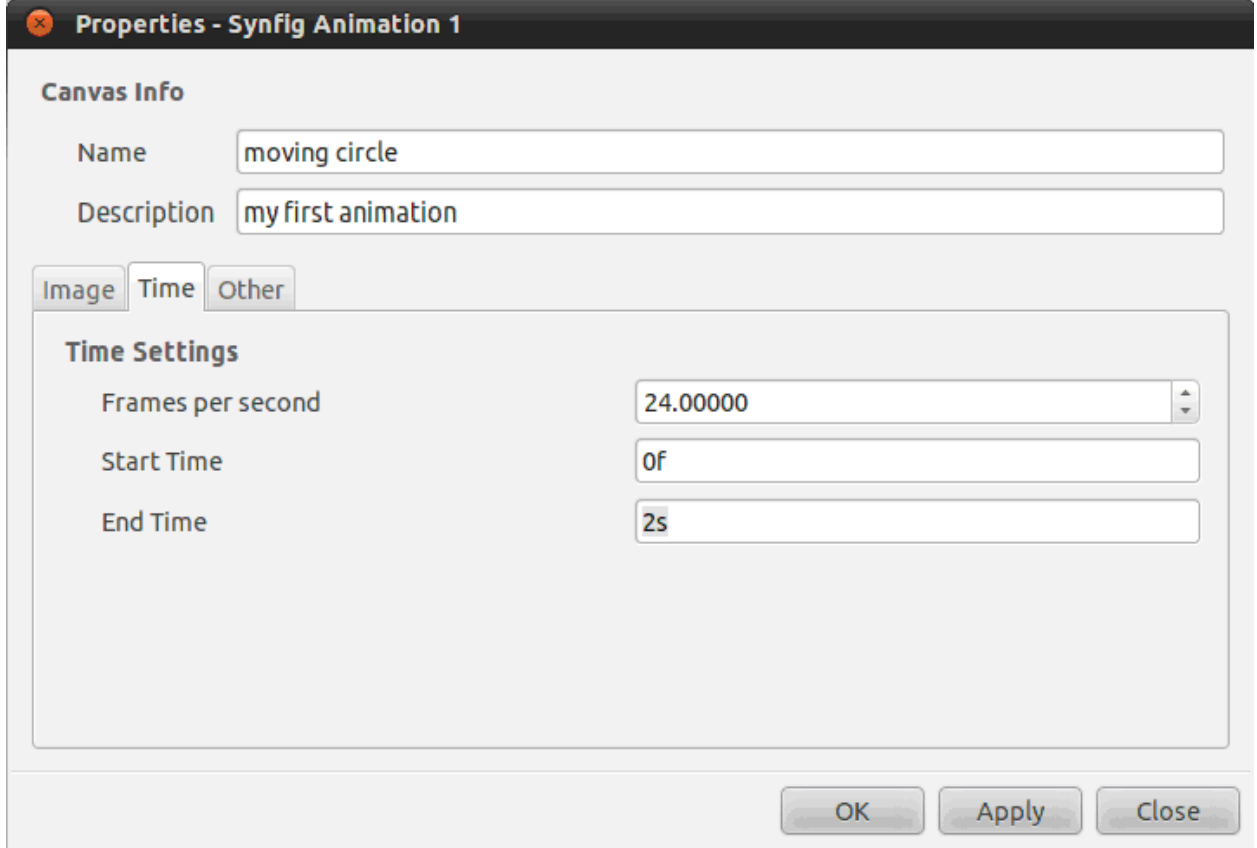
‘કી-ફ્રેમ’ મારફત એનિમેશન તૈયાર કરવાની પદ્ધતિ એક સરળ પદ્ધતિ છે. ‘કી-ફ્રેમ’ એટલે એવી ફ્રેમ કે જેમાં આપણે કોઈ ઓબ્જેક્ટના ગુણધર્મોમાં થયેલ ફેરફારને રાખીએ છીએ. કી-ફ્રેમનો ઉપયોગ કરવા તમારે ઓબ્જેક્ટની શરૂઆતની અને આખરની સ્થિતિ દર્શાવવી પડે. તેની વચ્ચે આવતી ફ્રેમ આપમેળે જ સીન્ડિંગને સમજાઈ જાય છે અને તેથી તે દર્શાવવાની જરૂર રહેતી નથી. ઉદાહરણ તરીકે, ધારોકે આપણે ડાબી બાજુથી જમણી બાજુ તરફ સરકતું જતું એક વર્તુળ બતાવવું છે, તો અહીં તમારે ડાબી બાજુ પહેલી કી-ફ્રેમ બનાવવી પડશે અને જમણી બાજુ છેલ્લી કી-ફ્રેમ મૂકવાની રહે. આ બન્નેની વચ્ચેની જરૂરી તમામ ફ્રેમ આપમેળે દોરાઈ જશે. આ રીતે અપાતી આવી અસરને ‘ટૂવીનિંગ’ (Tweening) કહે છે. ટૂવીનિંગ એ એક એવી ક્રિયા છે, જે બે કી-ફ્રેમની વચ્ચે આવનારી ફ્રેમની સંખ્યા ગણી કાઢે છે. આ પ્રકારના એનિમેશનને ‘સેલ એનિમેશન’ (cel animation) તરીકે પણ ઓળખવામાં આવે છે અને સામાન્ય રીતે, કમ્પ્યુટરમાં આ પ્રકારના એનિમેશનને રાખવામાં આવે છે. ઉપયોગકર્તા દ્વારા કેટલું દોરવું પડે અને કમ્પ્યુટર દ્વારા કેટલું આપમેળે દોરાઈ જાય એ તફાવત જ જુદાં-જુદાં એનિમેશન સોફ્ટવેરને એકબીજાંથી જુદાં પાડે છે.

કમ્પ્યુટર એનિમેશનનો અન્ય એક બીજો પ્રકાર ‘કાઈનેમેટિક્સ’ (Kinematics) તરીકે ઓળખાય છે. આ પદ્ધતિ કોઈ ઓબ્જેક્ટની ગતિ અને હલનચલનનો અભ્યાસ છે, જેવાં કે, ચાલતો માણસ કે દોડતો ચિત્તો. કોઈક એનિમેશનમાં આપણે ‘મોર્ફિંગ’ (Morphing) તરીકે ઓળખાતી અસરનો ઉપયોગ કરી શકીએ, જેમાં એક છબી બીજી છબીમાં રૂપાંતર પામે છે. ઉદાહરણ તરીકે તમે એવી ટીવી જાહેરાત જોઈ હશે કે જેમાં એક વ્યક્તિનો ચહેરો દર્શાવાય અને થોડીક જ પળોમાં તે વ્યક્તિનો ચહેરો બદલાઈને બીજી વ્યક્તિનો ચહેરો થઈ જાય. આ પ્રકારની અસર થઈ શકતી હોય તેવાં ઘણાં બધાં એનિમેશન સોફ્ટવેર બજારમાં ઉપલબ્ધ છે.

એનિમેશન બનાવવું

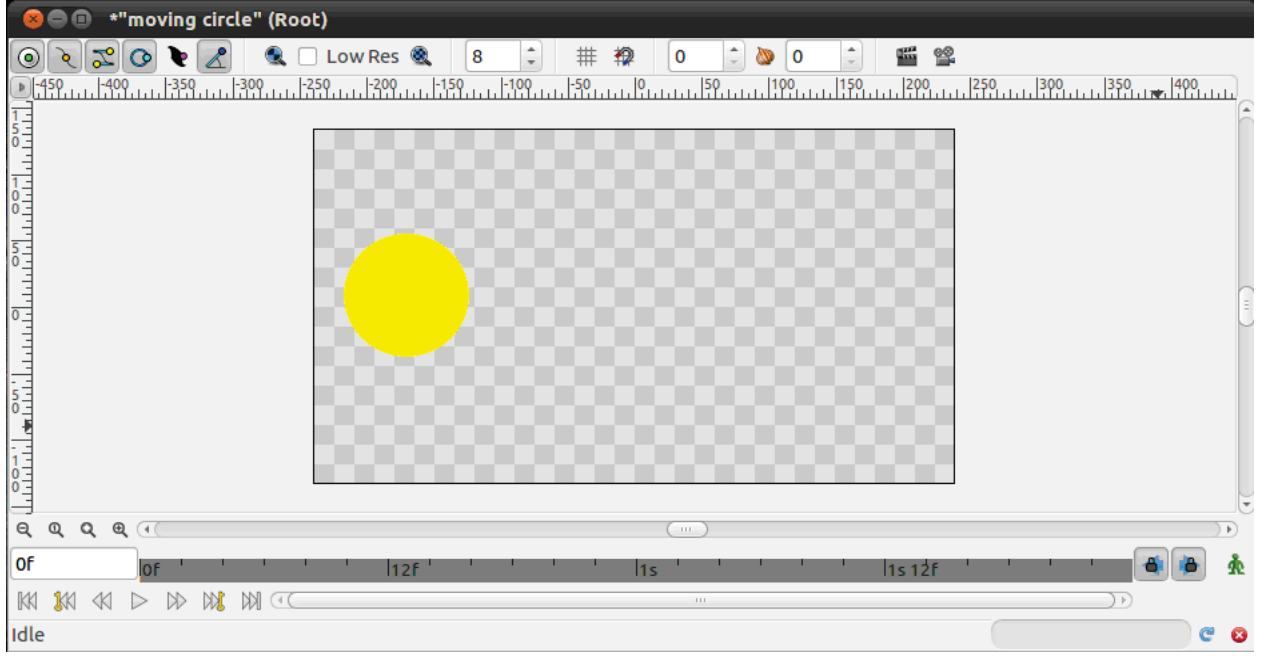
તો ચાલો, હવે આપણે આપણું પહેલું એનિમેશન બનાવીએ, જેમાં એક વર્તુળ ડાબી બાજુથી સરકતું-સરકતું જમણી બાજુ જતું હોય તેવું દેખાય. આવું એનિમેશન બનાવવા નીચે મુજબનાં પગલાં અનુસરો.

- Synfig શરૂ કરો (અથવા Application → Graphics → Synfig પસંદ કરો) આમ કરવાથી આપમેળે તમારા માટે એક નવી ફાઈલ ખૂલી જશે.
- Caret → Edit → Properties પર ક્લિક કરો. આમ કરવાથી આકૃતિ 3.1માં દર્શાવ્યા મુજબ Canvas Properties ડાયલોગબોક્સ દર્શાવાશે.



આકૃતિ 3.1 : Properties ડાયલોગબોક્સ

- તમારા પ્રથમ એનિમેશનનું નામ અને વર્ણન દર્શાવો. આકૃતિ 3.1માં દર્શાવ્યા મુજબ આપણે આપણા એનિમેશનને moving circle એવું નામ અને my first animation, એવું વર્ણન આપ્યું છે. હવે Time ટેબ પર ક્લિક કરી End time નામના ખાનામાં 2s ટાઈપ કરો. એનો અર્થ એ થાય કે, તમારું એનિમેશન 2 સેકન્ડનું થશે. સામાન્ય રીતે, આ સમય 5 સેકન્ડનો ગોઠવાયેલો જ હોય છે.
- OK બટન ક્લિક કરો. આમ કરવાથી Properties ડાયલોગબોક્સ બંધ થશે. અને આપણે કરેલ ફેરફાર જોઈ શકાય છે.
- હવે આપણે વર્તુળ દોરવું પડશે. ફોરગ્રાઉન્ડ રંગ પર ક્લિક કરીને અથવા પેલેટ (Palette)નો ઉપયોગ કરીને વર્તુળ માટેનો રંગ પસંદ કરો. પાછળથી પેરામીટર પેનલ (Parameter Panel)નો ઉપયોગ કરીને પણ આપણે રંગ બદલી શકીએ. હાલ આપણે વર્તુળમાં રંગ પૂરવા (fill color) માટે પીળો રંગ પસંદ કરીએ. એ પછી circle ટૂલ પસંદ કરીને આકૃતિ 3.2માં દર્શાવ્યા મુજબ વર્તુળ દોરો. દોરાયેલ વર્તુળમાં ફેરફાર કરવા માટે ટ્રાન્સફોર્મ (Transform) ટૂલ પસંદ કરો.

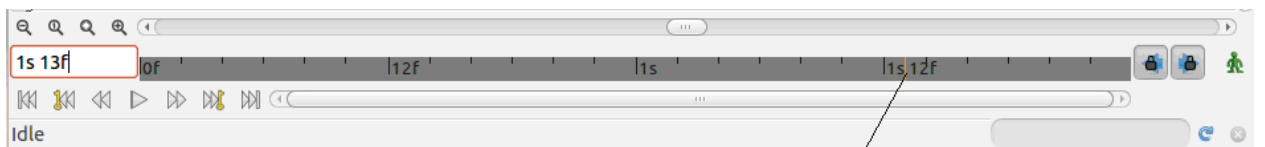
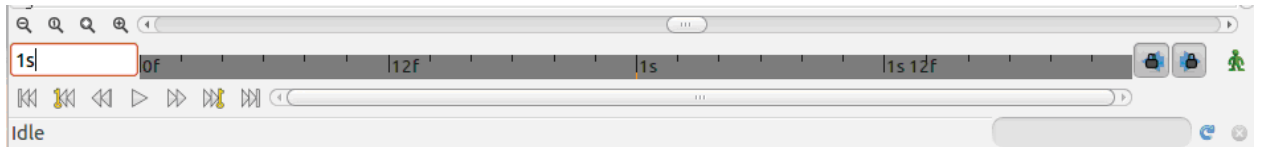
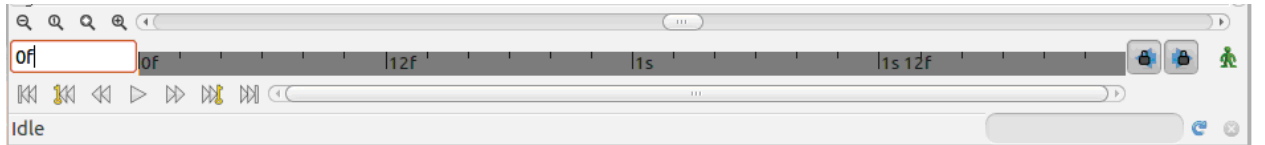


આકૃતિ 3.2 : Circle ટૂલનો ઉપયોગ

જેનું એનિમેશન કરવાનું છે, તે ઓબ્જેક્ટ સાથે હવે આપણે તૈયાર છીએ. એનિમેશન શરૂ કરતાં પહેલાં, ચાલો આપણે બે અગત્યની વસ્તુ સમજી લઈએ : ‘ટાઈમબાર’ (Timebar) અને ‘એનિમેટ એડિટિંગ મોડ’ (Animate Editing Mode).

ટાઈમબાર



આકૃતિ 3.3માં દર્શાવ્યા પ્રમાણે ટાઈમબાર પર ક્લિક કરો. આમ કર્યા પછી, સમયની સ્થિતિ દર્શાવતું નારંગી રંગનું બટન જોઈ શકો છો. જુદાં-જુદાં સ્થાન પર ક્લિક કરવાથી તમે આગળના ખાનામાં “0f, 1s, 1s 10f” વગેરે કિંમતો જોઈ શકો છો. આ આગળના ખાનામાં આપણે સીધીસીધી કિંમત ટાઈપ કરીને પણ Enter કી દબાવી શકીએ. આમ કરવાથી બટનનું સ્થાન બદલાશે. આ કિંમતો સેકન્ડ (s) અને ફ્રેમ (f)ના આધારે ટાઈમલાઈન પરનું ચોક્કસ સ્થાન દર્શાવે છે. સામાન્ય રીતે પ્રત્યેક સેકન્ડને 24 ફ્રેમમાં વિભાજિત કરવામાં આવે છે. તમે આકૃતિ 3.1માં જોઈ શકો છો કે, પ્રત્યેક સેકન્ડમાં ફ્રેમની સંખ્યા 24 રાખવામાં આવી છે. ફ્રેમ 0થી શરૂ થઈને 24 સુધી જાય છે. ઉદાહરણ તરીકે, જ્યારે એક સેકન્ડ અને તેર ફ્રેમ પસાર થઈ જાય તે વખતે આ ખાનામાં 1s 13f દર્શાવાશે.



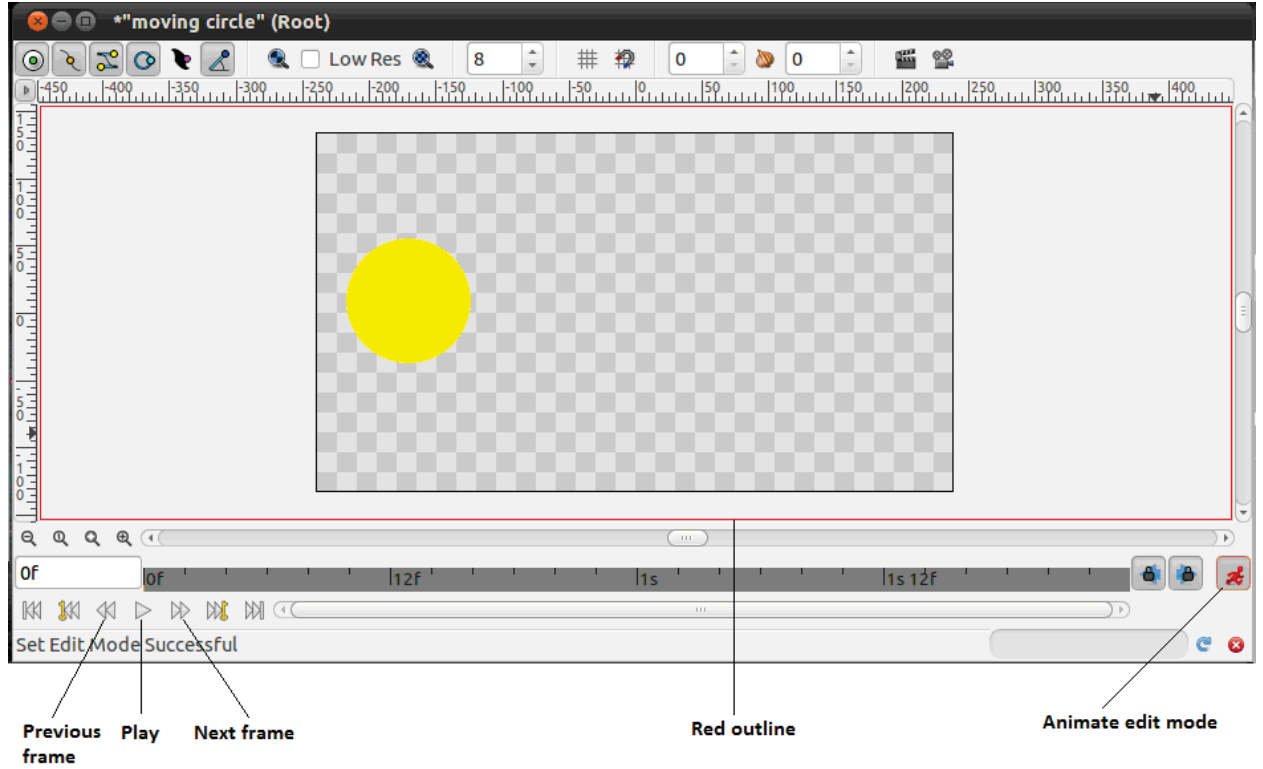
Orange indicator

આકૃતિ 3.3 : 0f, 1s અને 1s 13f કિંમત સાથે ટાઈમબાર

એનિમેટ એડિટિંગ મોડ

આકૃતિ 3.3માં દર્શાવેલ ટાઈમબારની જમણી બાજુ એક લીલા રંગનું બટન () જોઈ શકાય છે. જો અંત સમય શૂન્ય નહીં હોય તો તે દેખાતું હશે. આ બટન પર ક્લિક કરવાથી, બટન લાલ રંગનું () થઈ જશે અને કેનવાસની ફરતે લાલ રંગની લીટી દર્શાવાશે. આનો અર્થ એ થયો કે, હવે ‘એનિમેટ એડિટિંગ મોડ’ સક્રિય થઈ ગયો અને હવે આકૃતિ 3.4માં દર્શાવ્યા મુજબ કરવામાં આવતા કોઈ પણ ફેરફારની અસર ટાઈમ-સ્લાઈડરમાં થશે.

‘એનિમેટ એડિટિંગ મોડ’ માં જ્યારે પણ કોઈ ઓબ્જેક્ટનાં કોઈ પરિબળ બદલાશે, ત્યારે સ્થાનમાં થયેલ ફેરફાર અને ટાઈમ-સ્લાઈડર પર જે સમયે ફેરફાર થયો તે સમયને યાદ રાખવા એક નોંધ તૈયાર કરવામાં આવે છે.

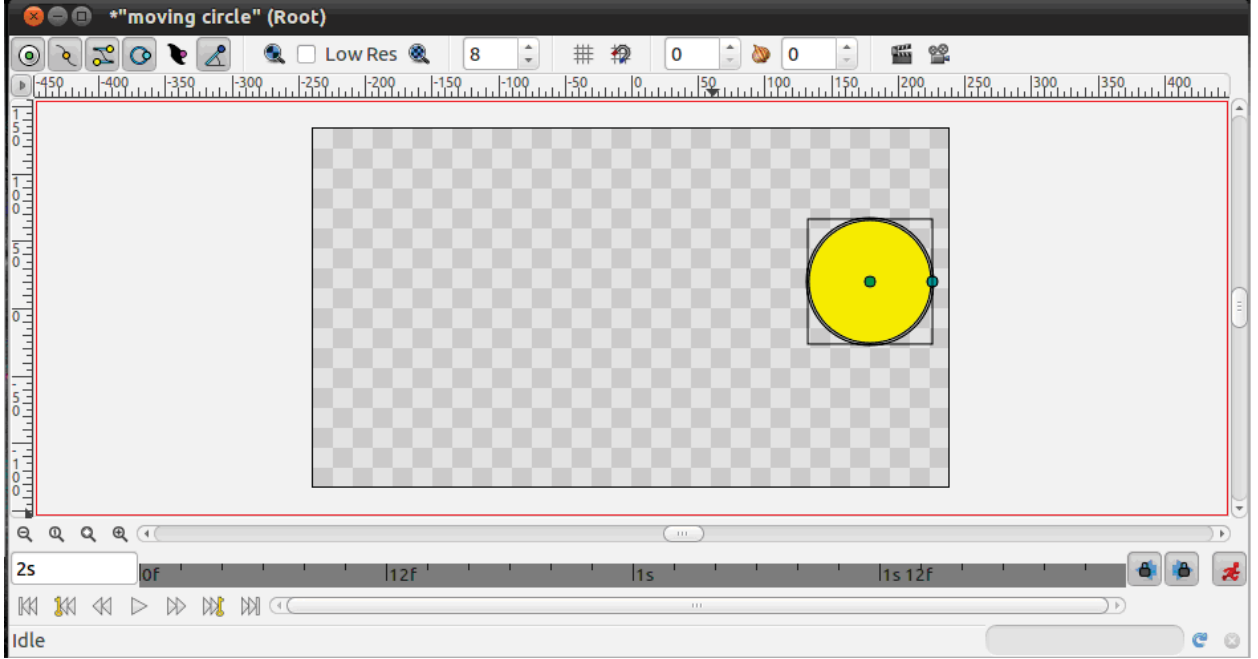


આકૃતિ 3.4 : એનિમેટ એડિટિંગ મોડ


જ્યારે ‘એનિમેટ એડિટિંગ મોડ’ સક્રિય ન હોય ત્યારે કોઈ પણ પરિબળમાં કરાતા ફેરફાર એનિમેશનની આખી ટાઈમલાઈનને લાગુ પાડવામાં આવે છે.

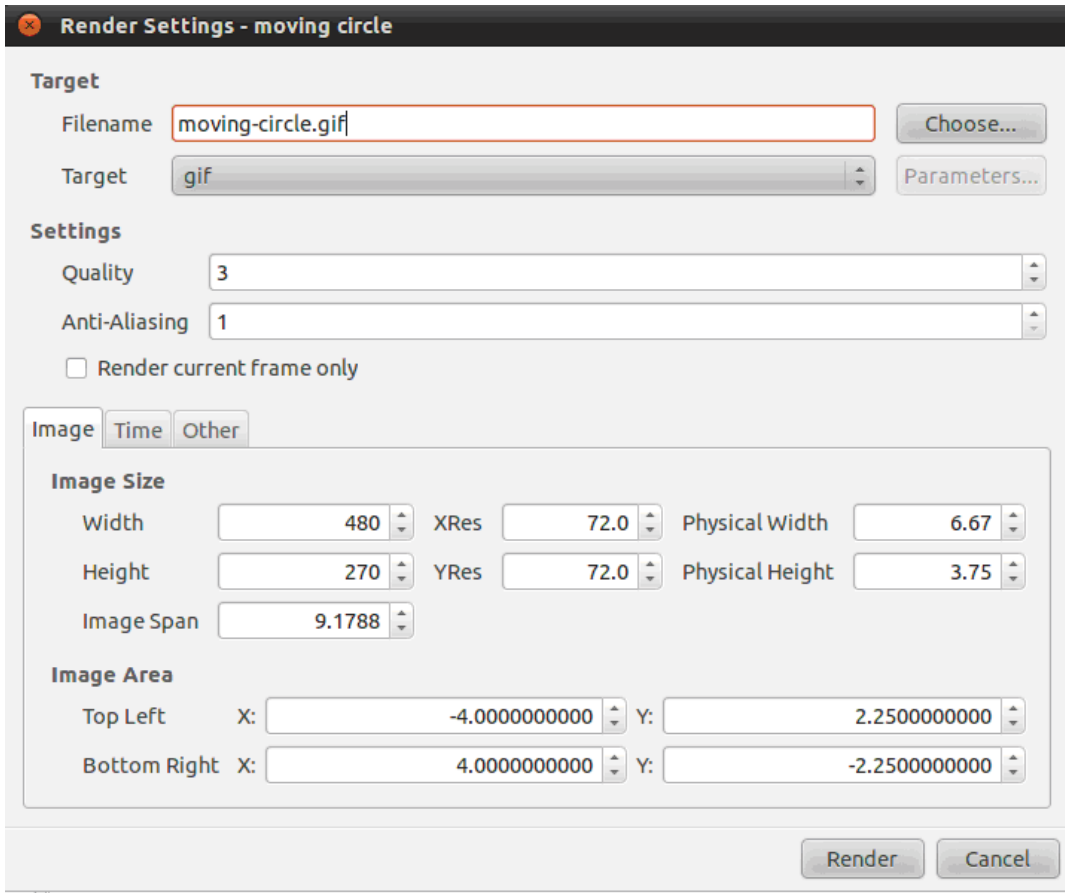
- તો ચાલો, હવે આપણું એનિમેશન બનાવવાનું શરૂ કરીએ. ‘એનિમેટ એડિટિંગ મોડ’ પર ક્લિક કરો.
- ‘ટાઈમલાઈન એન્ટ્રી ફિલ્ડ’ નામના ખાનામાં તમને 0f જોવા મળશે. વર્તુળને થોડુંક ફેરવો, જેથી કરીને તે પ્રથમ ફ્રેમનું સ્થાન નોંધી શકે. પ્રારંભિક સ્થાન તરીકે તમે કાર્યવિસ્તાર પર નવા સ્થાન ઉપર વર્તુળને ખસેડી શકો.
- હવે ટાઈમલાઈનના અંત ભાગ પર ક્લિક કરો. આમ કરવાથી ‘ટાઈમલાઈન એન્ટ્રી ફિલ્ડ’ નામના ખાનામાં ‘2s’ દર્શાવાશે. ટાઈમલાઈનના અંત પર પહોંચવા તમે જાતે પણ આ ખાનામાં ‘2s’ ટાઈપ કરી શકો.

આકૃતિ 3.5માં દર્શાવ્યા પ્રમાણે વર્તુળને કાર્યવિસ્તારની જમણી બાજુ સીધી લીટીમાં ખસેડવા, તેને પસંદ કરી shift કી દબાવી રાખો.



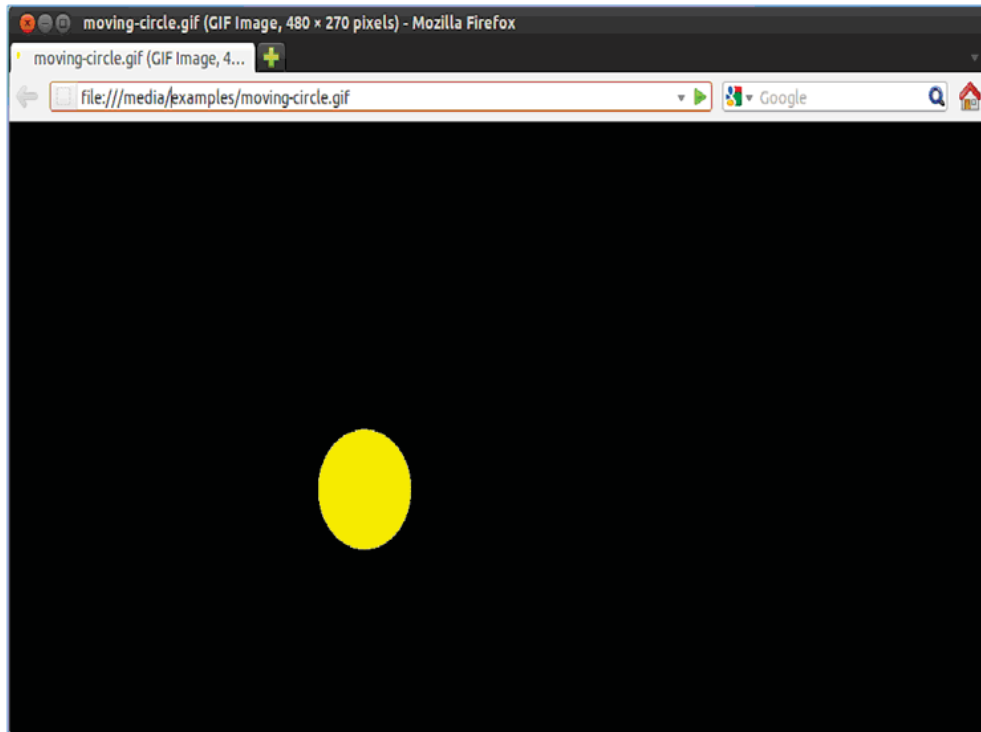
આકૃતિ 3.5 : વર્તુળને જમણી બાજુ ખસેડવું

- ટાઇમ-સ્લાઇડર પર કોઈ પણ સ્થાન પર ક્લિક કરશો, એટલે તરત જ વર્તુળ નવા સ્થાન પર જોવા મળશે. આપણે તો માત્ર વર્તુળનું શરૂનું અને અંતનું સ્થાન જ દર્શાવ્યું હતું, પરંતુ Synfig આપમેળે વચ્ચેની વર્તુળની ફેમની છબીઓ તૈયાર કરી દે છે. અગાઉ આપણે ચર્ચા કરી તે મુજબ આ પ્રક્રિયાને ‘ટૂવીનિંગ’ (tweening) કહે છે. આ બધી ફેમો જ્યારે એકસાથે દર્શાવવામાં આવે છે, ત્યારે તે વર્તુળ ખસતું હોય તેવી અસર ઊભી કરે છે.
- આપણું એનિમેશન કેવું દેખાશે તે જોવા માટે, આકૃતિ 3.4માં દર્શાવ્યા પ્રમાણે Play બટન પર ક્લિક કરો. એ જ રીતે એનિમેશન એક પછી એક ફેમ જોવા તમે Previous Frame અથવા Next Frames બટન પર પણ ક્લિક કરી શકો.
- છેલ્લે, આપણે આપણા કામનું ‘રેન્ડરિંગ’ (Rendering) કરવું પડે. રેન્ડરિંગ એટલે, તમે બનાવેલ ઓબ્જેક્ટ પર દર્શાવેલ અસરો આપવા કમ્પ્યુટર દ્વારા ઉપયોગમાં લેવાતા વિવિધ તર્ક (અલ્ગોરિધમ).
- રેન્ડરિંગ કરતાં પહેલાં ‘એનિમેટ એડિટિંગ મોડ’માંથી બહાર નીકળવા  બટન પર ક્લિક કરો.
- Caret → File → Save કમાન્ડ પસંદ કરી ફાઇલને moving-circle.sifz નામ આપી સાચવી દો.
- Caret → File → Render કમાન્ડ પસંદ કરો, અને ફાઇલનું નામ બદલીને moving-circle.gif કરી, તેને જ્યાં moving-circle.sifz સાચવી હતી, ત્યાં સાચવી દો. ફાઇલસ્વરૂપ તરીકે Autoને બદલે gif. પસંદ કરો, ત્યાર બાદ આકૃતિ 3.6માં દર્શાવ્યા મુજબ Render બટન પર ક્લિક કરો.



આકૃતિ 3.6(a) : ફાઇલ-રેન્ડરિંગ કરવું

- તમારું એનિમેશન નિહાળવા વેબબ્રાઉઝરમાં અથવા ઇમેજ વ્યુઅર moving-circle.gif ફાઇલ ખોલો. આકૃતિ 3.6 (b) વેબબ્રાઉઝરમાં દેખાય છે, તેવું પરિણામ દર્શાવે છે.



આકૃતિ 3.6(b) : વેબબ્રાઉઝર પર દેખાતું પરિણામ

સરસ. આમ, હવે તમે તમારું પ્રથમ એનિમેશન તૈયાર કર્યું.

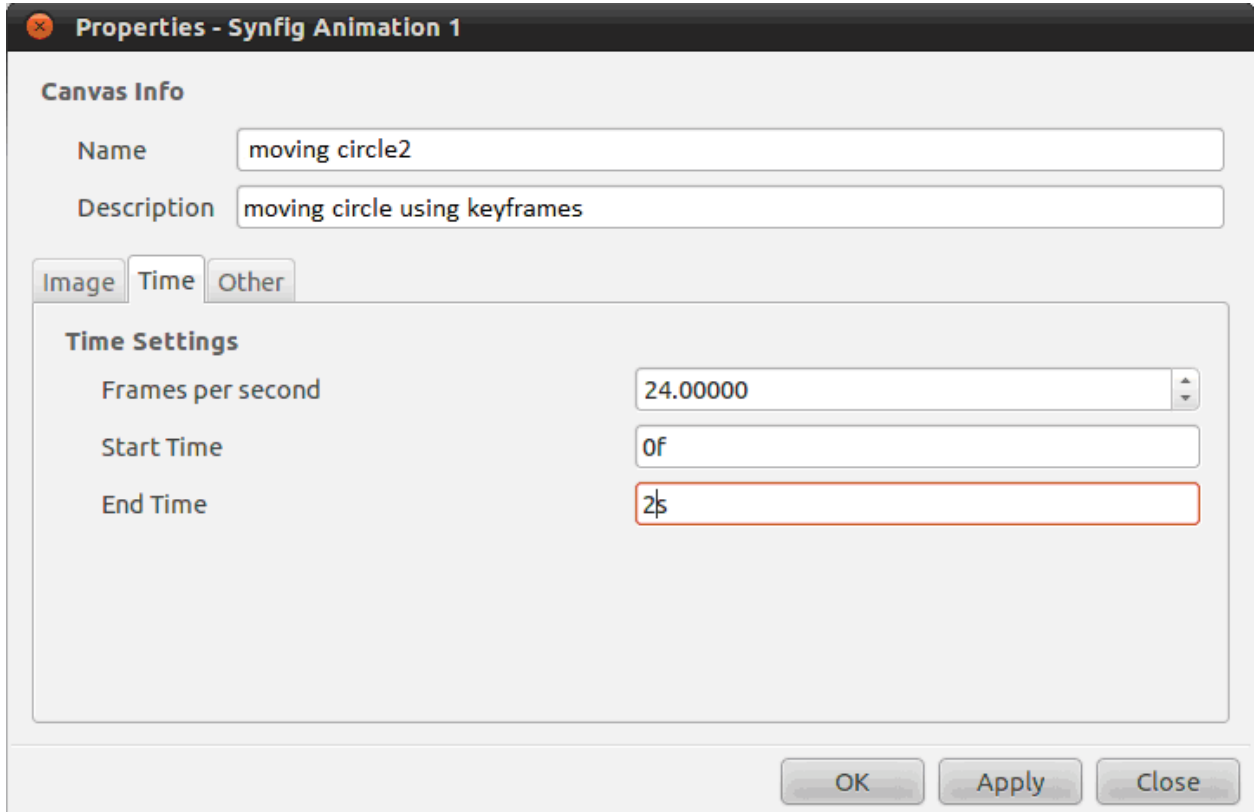
કી-ફ્રેમના ઉપયોગથી એનિમેશન

ચાલો હવે, કી-ફ્રેમના ઉપયોગ દ્વારા ડાબેથી જમણે અને પાછું ડાબી બાજુ સરકતા વર્તુળનું એનિમેશન તૈયાર કરીએ. આ માટે આપણે નીચે મુજબનાં ત્રણ પગલાં અનુસરવાં પડે :

- પ્રથમ વર્તુળ ડાબી બાજુ
- બીજું વર્તુળ જમણી બાજુ
- અને ત્રીજું વર્તુળ પાછું ડાબી બાજુ

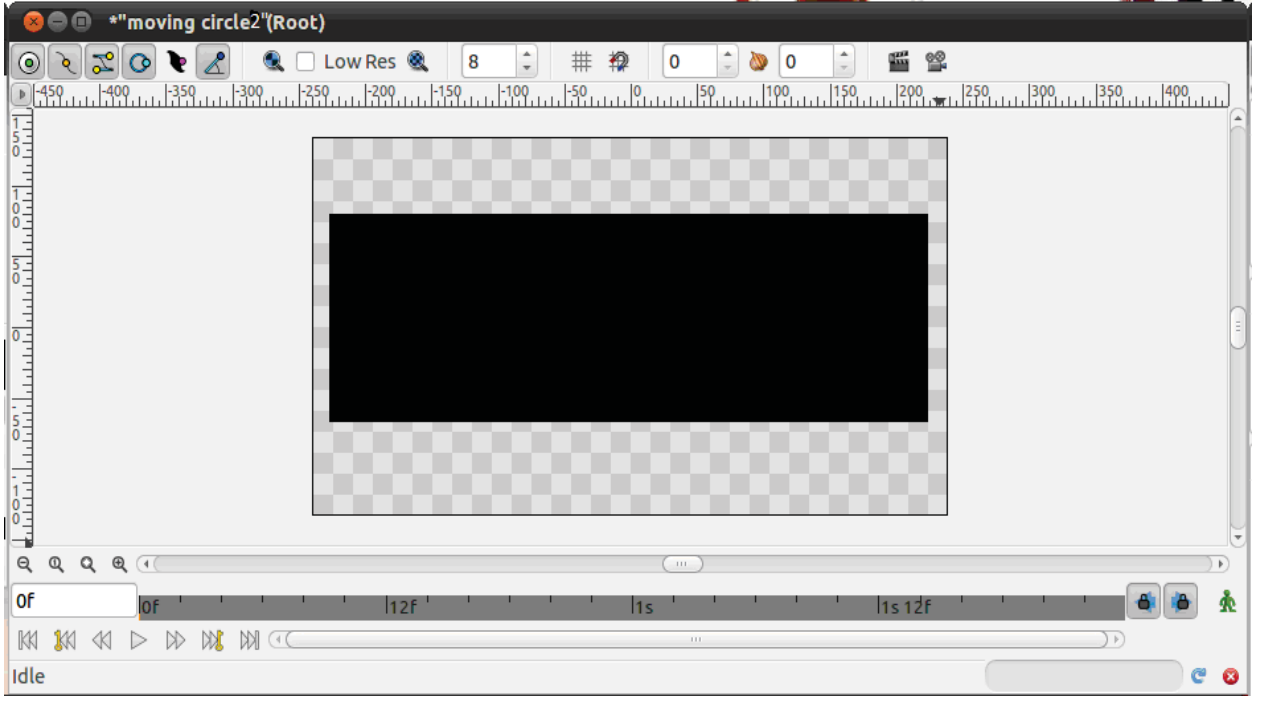
આ એનિમેશન તૈયાર કરવા નીચે મુજબનાં પગલાં અનુસરો :

- Synfig શરૂ કરીને નવી ફાઈલ ખોલો.
- Caret → Edit → Properties મેનુ-વિકલ્પ પસંદ કરો. તમારા એનિમેશનને નામ આપી વર્ણન દર્શાવો. આકૃતિ 3.7માં દર્શાવ્યા મુજબ આપણે આપણા એનિમેશનને moving circle2 નામ આપ્યું અને વર્ણન તરીકે moving circle using keyframes આપ્યું. હવે, Time ટેબ પર ક્લિક કરીને અંતસમય તરીકે 2S કરો. ફેરફાર કાયમ કરવા OK પર ક્લિક કરો.



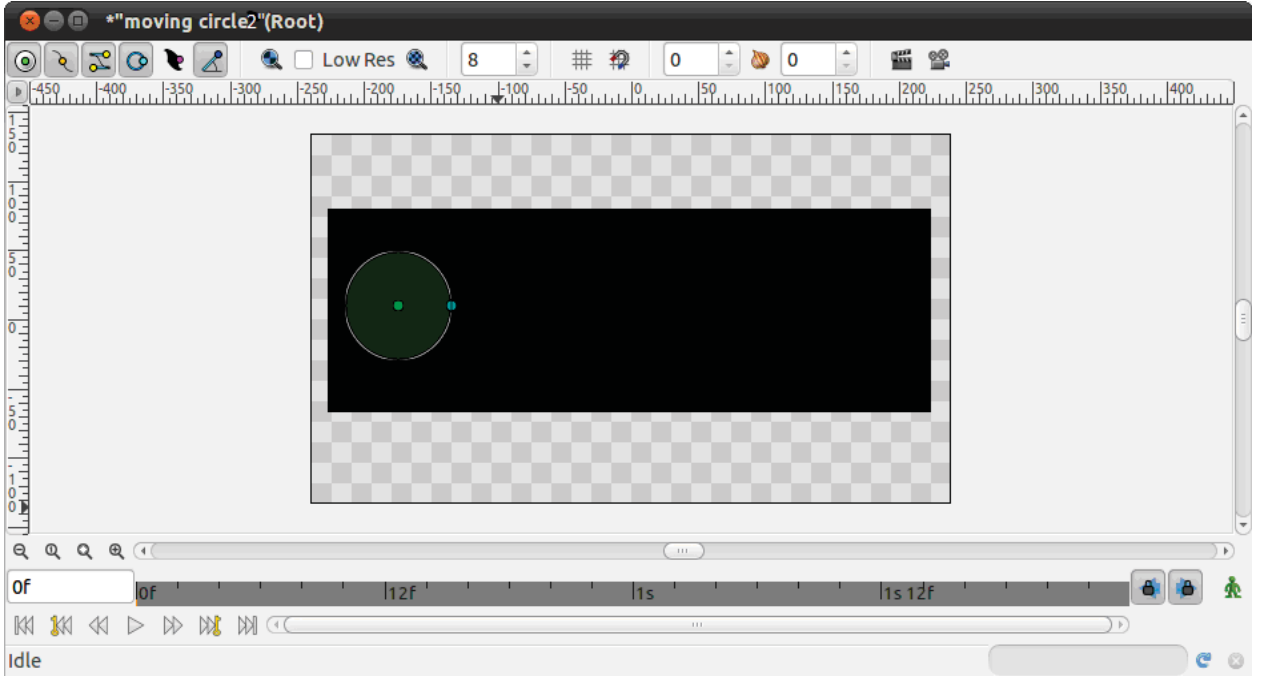
આકૃતિ 3.7 : Canvas properties ડાયલોગબોક્સ

- તો ચાલો, હવે આપણે આપણા એનિમેશન માટે બેકગ્રાઉન્ડ બનાવીએ. Rectangle ટૂલને પસંદ કરો અને આકૃતિ 3.8માં દર્શાવ્યા મુજબ એક સાદો ડાળો લંબચોરસ બનાવો.





આકૃતિ 3.8 : Rectangle ટૂલ વડે લંબચોરસ

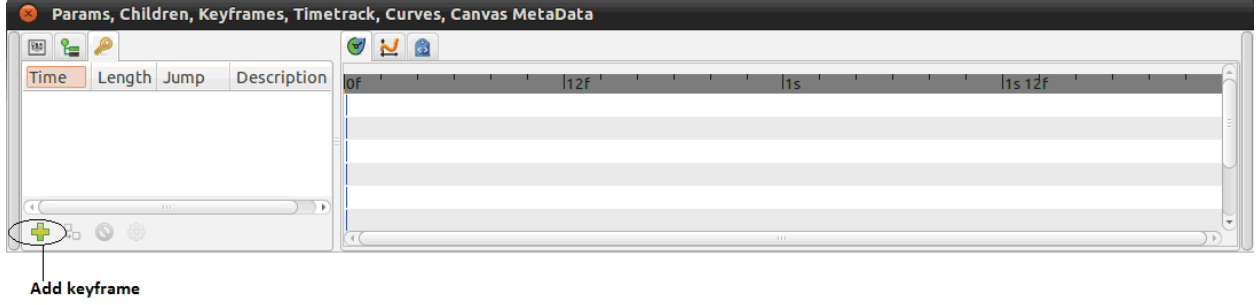
- હવે આપણે વર્તુળ દોરવાની જરૂર છે. સૌપ્રથમ Fill Colorને બદલીને 'લીલો' રંગ પસંદ કરો. એ પછી Circle ટૂલ પસંદ કરી આકૃતિ 3.9માં દર્શાવ્યા મુજબ વર્તુળ બનાવો. દોરેલા વર્તુળમાં ફેરફાર કરવા તમે Transform ટૂલનો ઉપયોગ કરી શકો.




આકૃતિ 3.9 : Circle ટૂલ વડે વર્તુળ

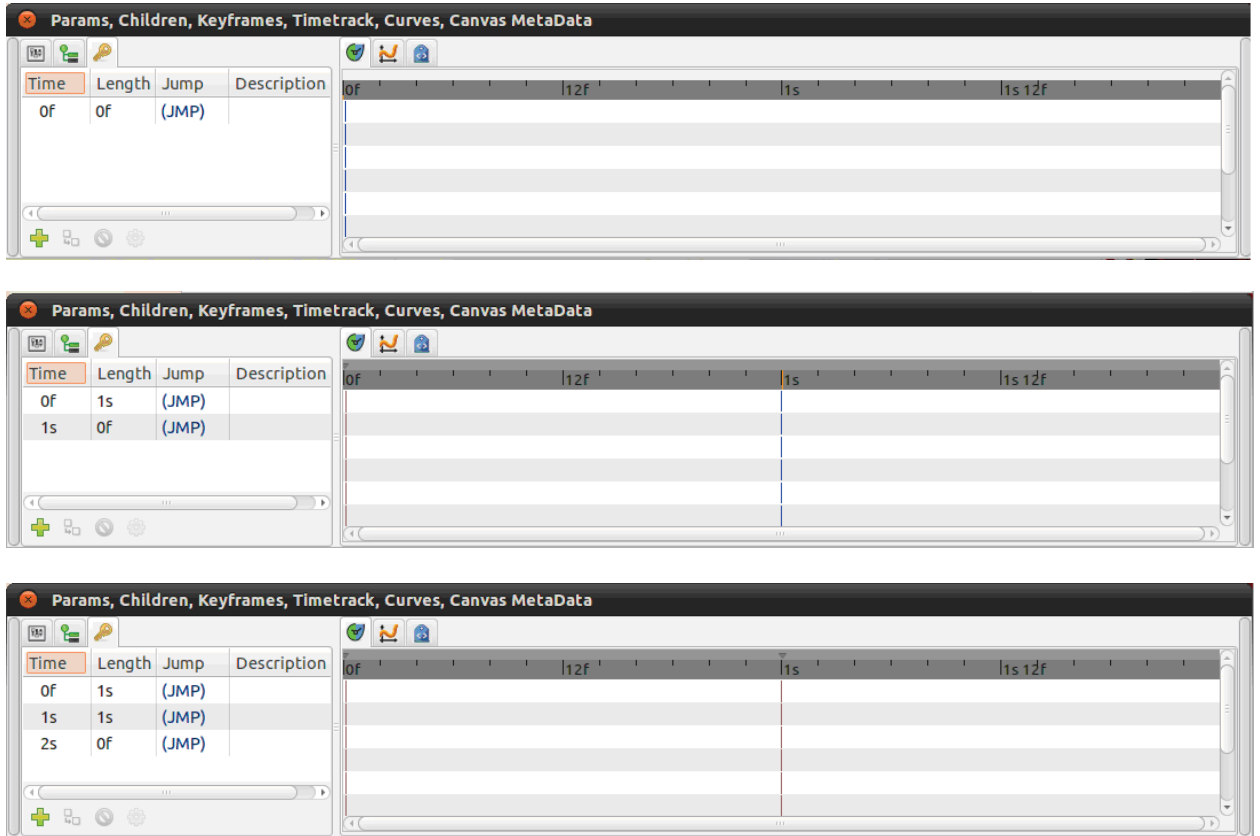
- હવે જ્યારે ઓબ્જેક્ટ તૈયાર છે, ત્યારે ચાલો હવે, Animate Editing Mode પર ક્લિક કરીને આપણું એનિમેશન તૈયાર કરવાનું શરૂ કરીએ.

- ‘પેરામીટર પેનલ’ પર જાવ અને કી-ફ્રેમ ઉમેરવા કી-ફ્રેમ  પેનલ ખોલો. આગળ આપણે ત્રણ પગલાંની ચર્ચા કરી હતી : તે હવે આપણી કી-ફ્રેમ બનશે. આકૃતિ 3.10માં દર્શાવ્યા મુજબ નવી કી-ફ્રેમ ઉમેરવા  બટન પર ક્લિક કરો. આમ, કરવાથી યાદીમાં નવી લીટી ઉમેરાશે, જે 0f, 0f (JMP) દર્શાવશે.



આકૃતિ 3.10 : કી-ફ્રેમ પેનલ

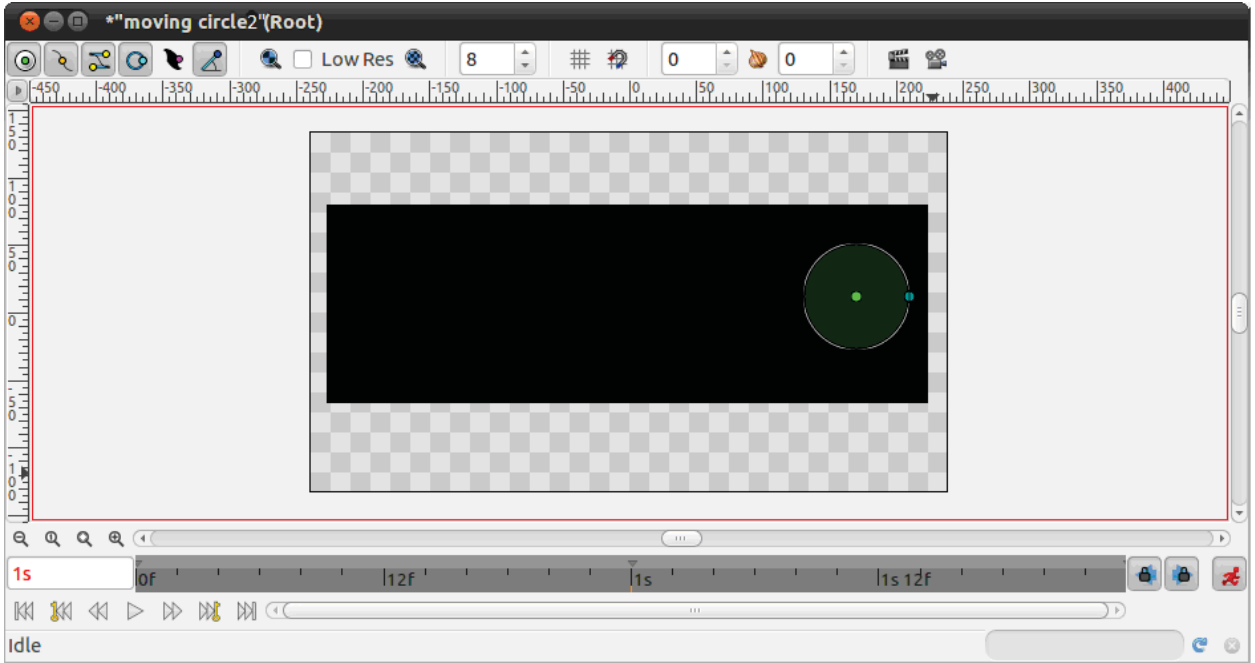
- ટાઈમ-સ્લાઈડરમાં 1s નિશાની પર જાવ. અન્ય કી-ફ્રેમ ઉમેરવા  બટન પર ક્લિક કરો. ફરીથી ટાઈમ-સ્લાઈડરમાં 2s પર જાવ અને અન્ય કી-ફ્રેમ ઉમેરો. આમ હવે, આકૃતિ 3.11માં દર્શાવ્યા મુજબ યાદીમાં ત્રણ કી-ફ્રેમ છે.



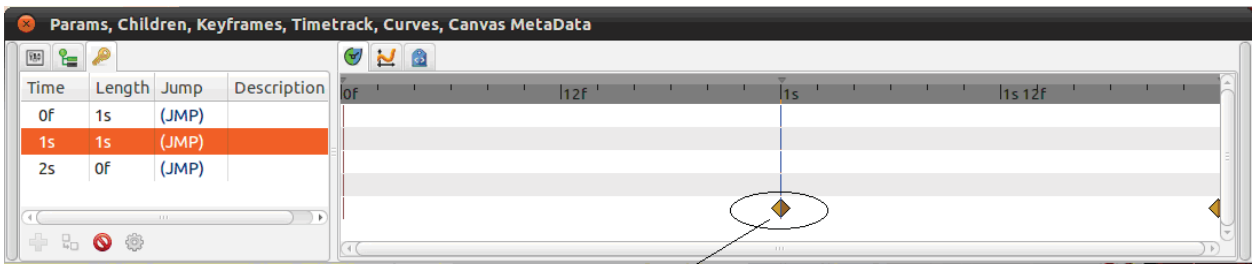
આકૃતિ 3.11 : 0f, 1s અને 2s પર ત્રણ કી-ફ્રેમ

તો ચાલો, હવે 'કી-ફ્રેમ પેનલ' વિશે સમજીએ. તે Time દર્શાવે છે, જે શરૂઆતના સમયનો નિર્દેશ કરે છે. Length જે સમય-અવધિ દર્શાવે છે. Jump, જે સચોટ સમય અને Description પર કૂદવા માટે ઉપયોગમાં લેવાતી કડીઓ છે.

- અહીં આપણે ડાબી બાજુ પર તમામ ત્રણેય કી-ફ્રેમમાં વર્તુળો દોર્યા છે. પરંતુ આપણાં પગલાંઓમાં ચર્ચા કર્યા પ્રમાણે વર્તુળનું બીજું સ્થાન જમણી બાજુ છે. તો ચાલો, હવે બીજી કી-ફ્રેમના વર્તુળમાં સુધારા કરીએ, જેમકે 1s સ્થાન પરના (JMP) પર ક્લિક કરો અને આકૃતિ 3.12 (a)માં દર્શાવ્યા મુજબ વર્તુળને ડાબી બાજુથી જમણી બાજુ ખસેડો. આમ, આખરે આપણું વર્તુળ ડાબી બાજુ 0f પર, એ પછી જમણી બાજુ 1s પર અને ફરી ડાબી બાજુ 2s પર હશે. આકૃતિ 3.12 (b)માં દર્શાવ્યા મુજબ ટાઈમ-ટ્રેક પેનલ પર તમે નારંગી રંગના ત્રણ ચતુષ્કોણ (ડાયમંડ) જોઈ શકો છો, જેને વે-પોઈન્ટ (Waypoint) કહે છે. એ ઓબ્જેક્ટનાં વિવિધ પરિબલો (જેવાં કે, રંગ કે સ્થાન) બદલાયાં ત્યારનો સમય રજૂ કરે છે.
- ટાઈમ-સ્લાઈડરમાં કોઈ પણ સ્થાન પર ક્લિક કરશો, તો વર્તુળ તમને નવા સ્થાન પર જોવા મળશે. આપણે આપણા વર્તુળનાં માત્ર ત્રણ સ્થાન જ દર્શાવ્યાં, પરંતુ Synfigએ આપમેળે એની વચ્ચેની છબીઓ કે ફ્રેમ બનાવી દીધી. આ ફ્રેમ જ્યારે બધી એકસાથે દર્શાવાશે, ત્યારે આપણને સરકતા વર્તુળનો આભાસ કરાવશે.




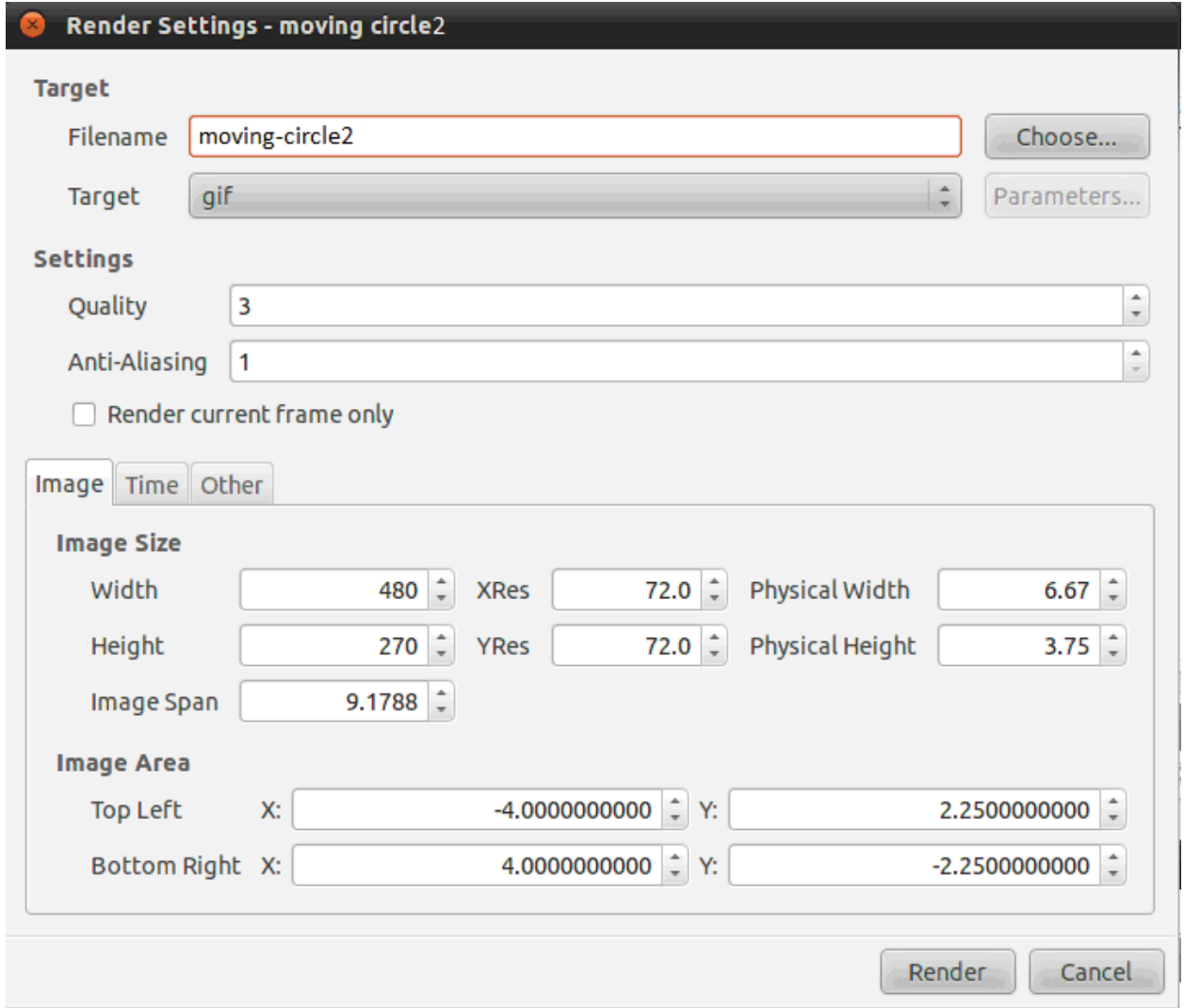
આકૃતિ 3.12 (a) : જમણી બાજુ 1s પર વર્તુળને ખસેડવા



Waypoints

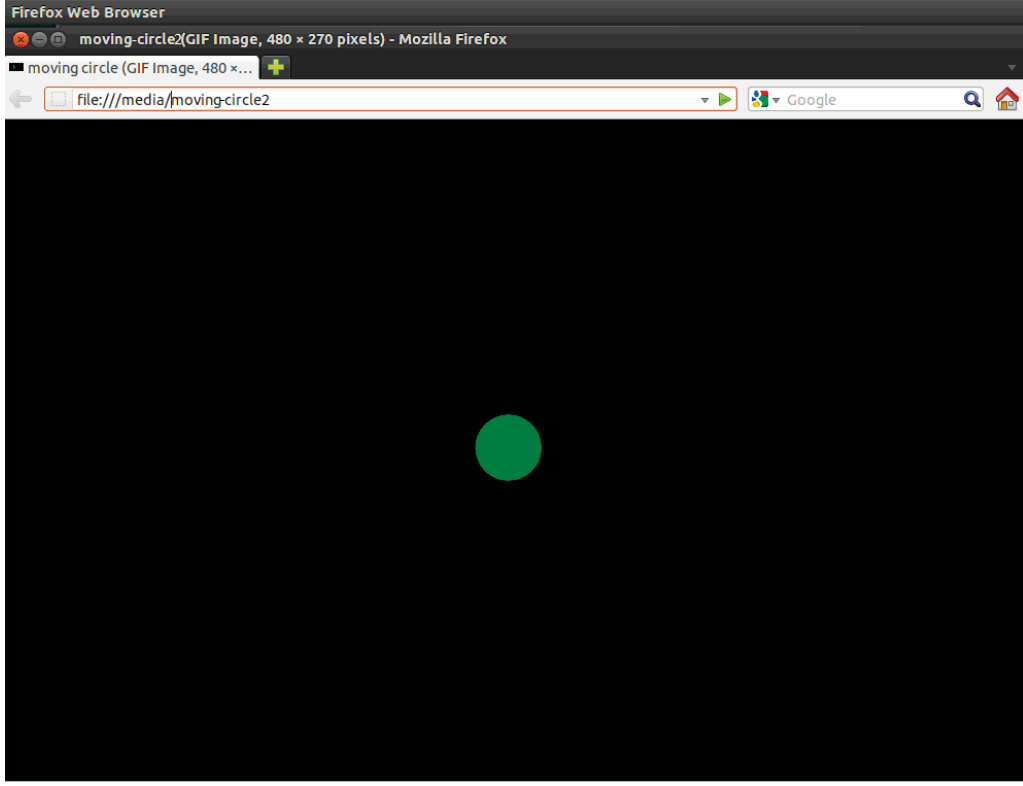
આકૃતિ 3.12(b) : વે-પોઈન્ટ

- હવે એનિમેશનને નિહાળવા માટે આપણે આપણા કામનું 'રેન્ડરિંગ' કરવું જરૂરી છે.
- રેન્ડરિંગ કરતાં પહેલાં, 'એનિમેટ એડિટિંગ મોડ'માંથી બહાર નીકળવા  બટન પર ક્લિક કરો.
- Caret → File → Save પર ક્લિક કરીને moving-circle2.sifz નામ વડે ફાઇલને સાચવી દો.
- Caret → File → Render પર ક્લિક કરીને moving-circle2.sifz ફાઇલ જ્યાં સાચવી હતી, ત્યાં જ નામ બદલીને moving-circle.gif તરીકે સાચવો. ફાઇલના સ્વરૂપ તરીકે Autoને બદલે .gif પસંદ કરીને આકૃતિ 3.13(a)માં દર્શાવ્યા મુજબ Render પર ક્લિક કરો. વિન્ડોના નીચેના ભાગે આવેલ સ્ટેટસ બાર પર તમને File rendered successfully સંદેશો દેખાશે.



આકૃતિ 3.13(a) : ફાઇલ રેન્ડરિંગ કરવી

- તમારા એનિમેશનને નિહાળવા વેબબ્રાઉઝરમાં moving-circle.gif ફાઇલ ખોલો. આકૃતિ 3.13 (b) પરિણામ દર્શાવે છે.

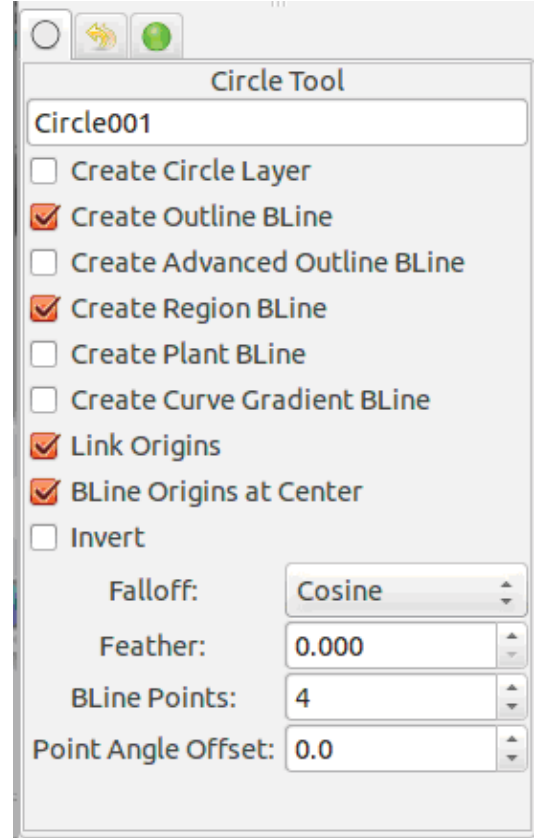


આકૃતિ 3.13(b) : એનિમેશનનું પરિણામ

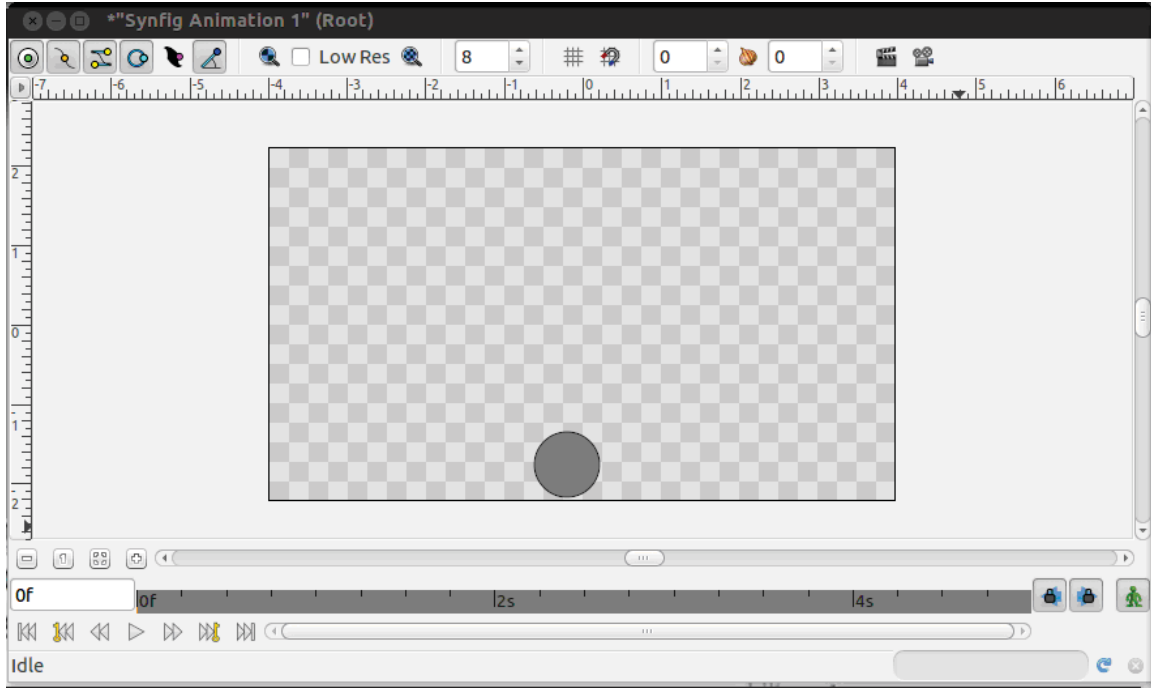
ટાઈમ લૂપનો ઉપયોગ કરવો

જો એનિમેશનનું ફરીથી પુનરાવર્તન થાય તેવું તમે ઈચ્છતા હોય, તો ટાઈમ લૂપ (Time loop)નો ઉપયોગ કરો. સાદા શબ્દોમાં લૂપ (loop) એટલે ‘ચક્કર’ કે ‘ફેરા’ અને ‘ટાઈમ લૂપ’ એટલે અમુક સમય સુધી ચાલતું ચક્કર. તો ચાલો, ટાઈમ લૂપનો ઉપયોગ સમજવા આપણે ઊછળતા દડા (bouncing ball)નું ઉદાહરણ લઈએ. કી-ફ્રેમની કેવી રીતે નક્કલ કરવી તે પણ આપણે જોઈશું, જેથી એનિમેશનમાં ફરીથી ઓબ્જેક્ટને ગોઠવવાનો સમય બચે. આ ઉદાહરણમાં આપણે દડાને દીવાલ બાજુ ઉછાળીશું. ડુપ્લિકેટ કી-ફ્રેમ સિદ્ધાંતના ઉપયોગ દ્વારા એનિમેશનનું બે વાર સર્જન થશે અને એનિમેશનના અંત સમય સુધી દડાનું ઊછળવાનું ચાલુ રહે તેવું દર્શાવવા ટાઈમ લૂપનો ઉપયોગ કરો.


- નવી ફાઈલ બનાવો.
- Circle ટૂલ પસંદ કરો અને Tool વિકલ્પમાં આકૃતિ 3.14માં દર્શાવ્યા પ્રમાણે Region અને Outline પસંદ કરો. આકૃતિ 3.15માં દર્શાવ્યા પ્રમાણે કેનવાસ પર વર્તુળ દોરો.

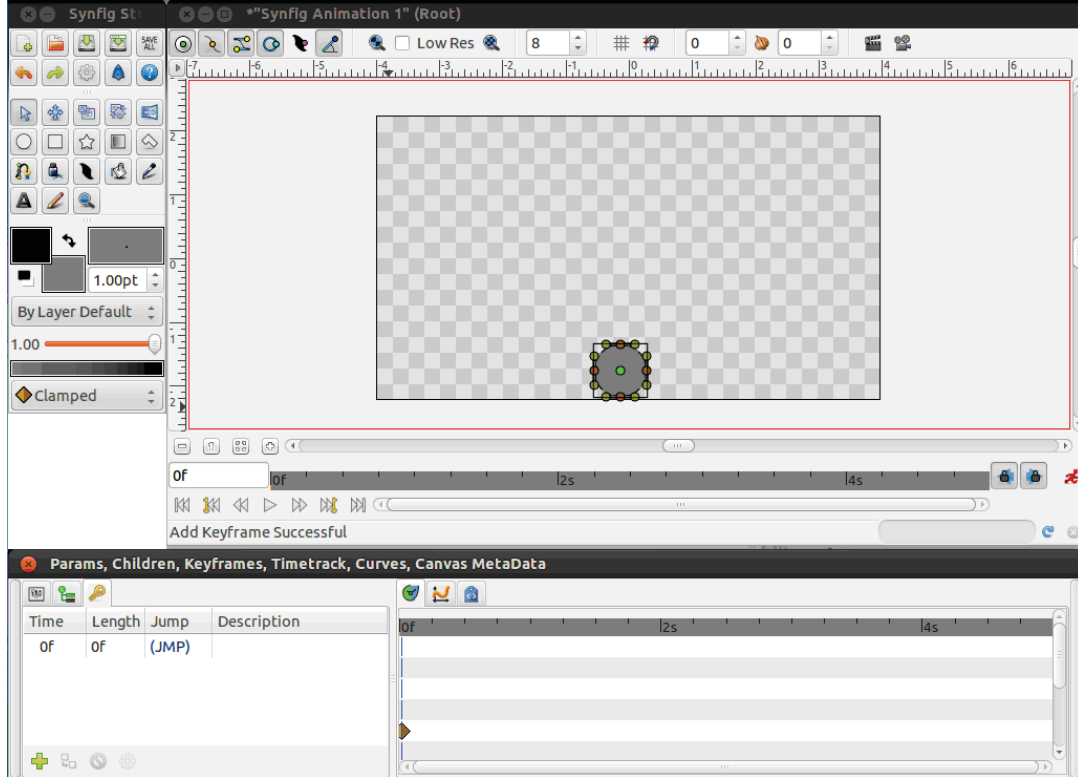


આકૃતિ 3.14 : Circle ટૂલના વિકલ્પો



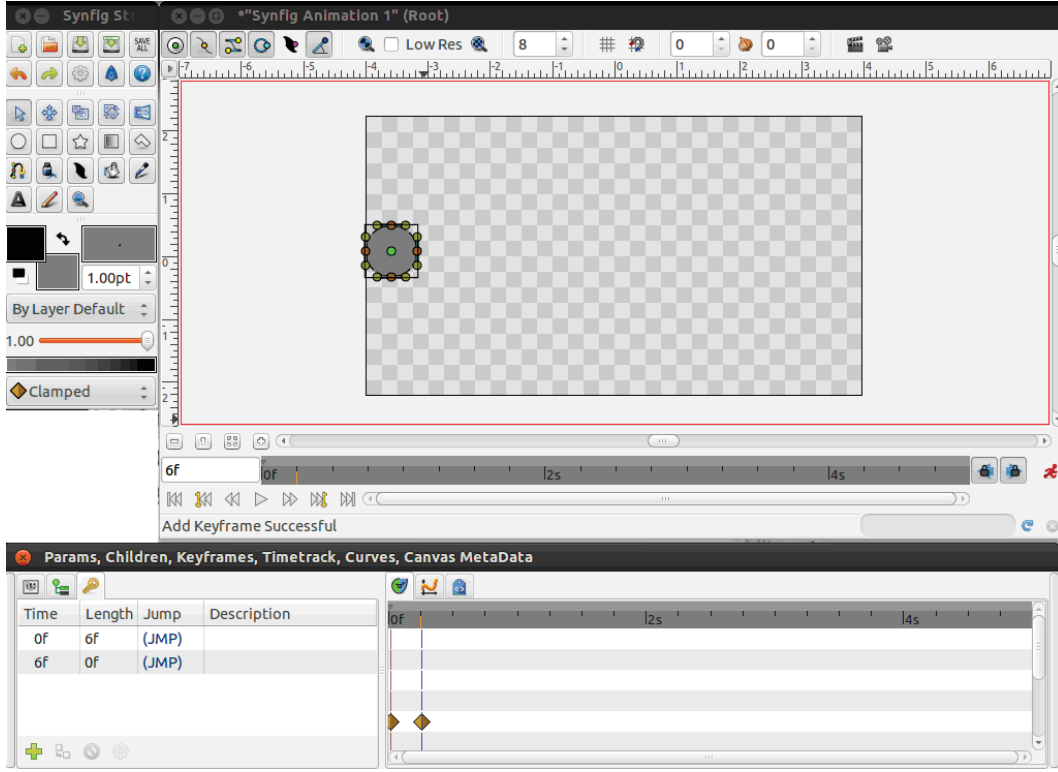
આકૃતિ 3.15 : Circle ટૂલ વડે દોરાયેલ દડો

- ‘એનિમેટ મોડ’ ચાલુ કરો. ટાઈમ-લાઈન ખાનામાં દડાને તેના સ્થાનથી થોડો ખસેડીને 0f સ્થાન પર લઈ આવો. હવે કી-ફ્રેમ પેનલ પસંદ કરો અને કી-ફ્રેમ પેનલના નીચેના ભાગે  બટન પર ક્લિક કરીને કી-ફ્રેમ ઉમેરો.



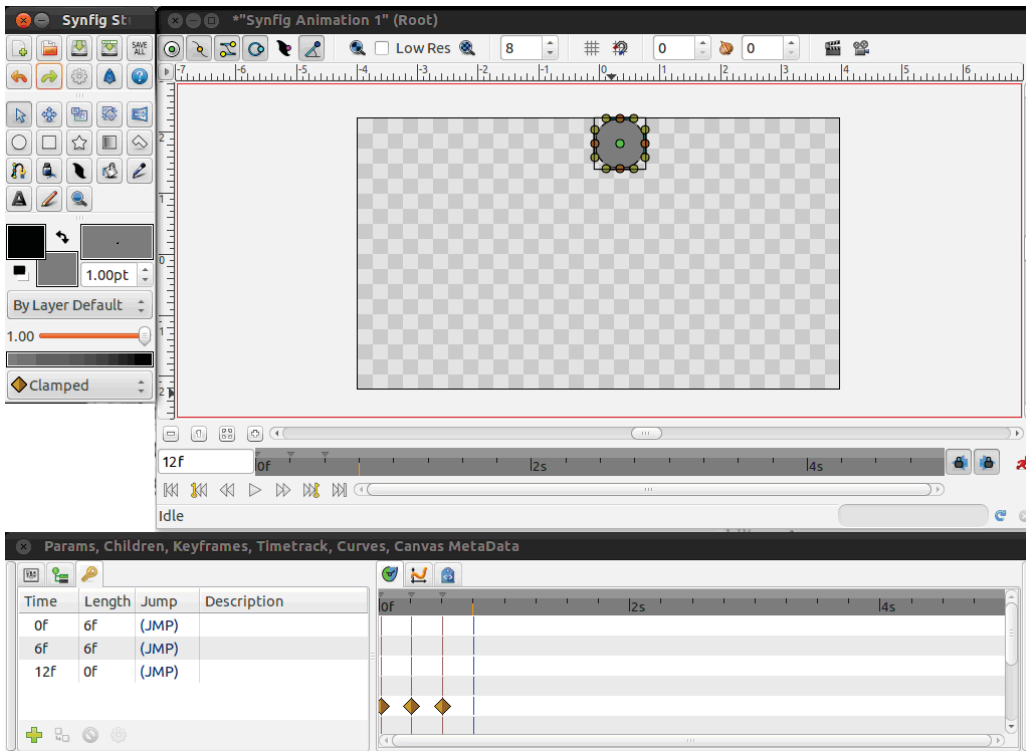
આકૃતિ 3.16 : 0f પર કી-ફ્રેમ

- હવે ટાઈમલાઈનમાં 6f પર આવો અને દડાને નવા સ્થાન તરફ ઘસડી જાવ. આકૃતિ 3.17માં દર્શાવ્યા પ્રમાણે 6f પર નવી કી-ફ્રેમ ઉમેરો.

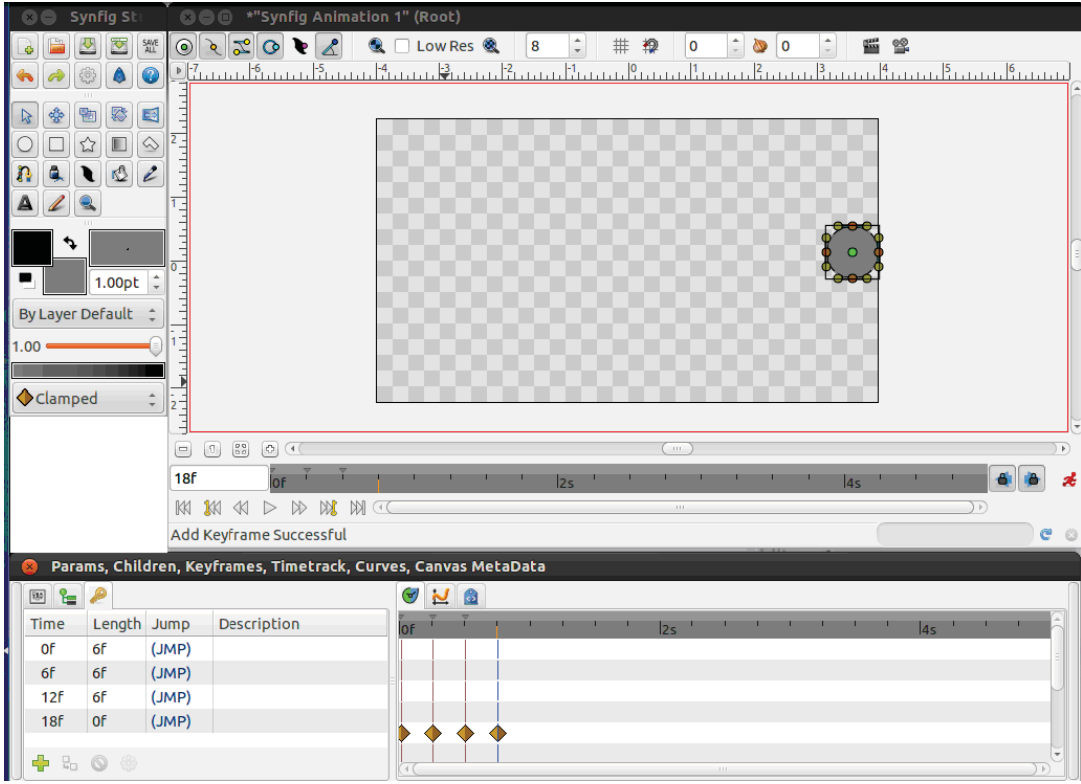


આકૃતિ 3.17 : 6f પર ઉમેરેલ કી-ફ્રેમ

- એ જ રીતે, “12f”, “18f” પર જઈને દડાને આકૃતિ 3.18 અને 3.19માં દર્શાવ્યા પ્રમાણેના સ્થાન પર ઘસેડો.

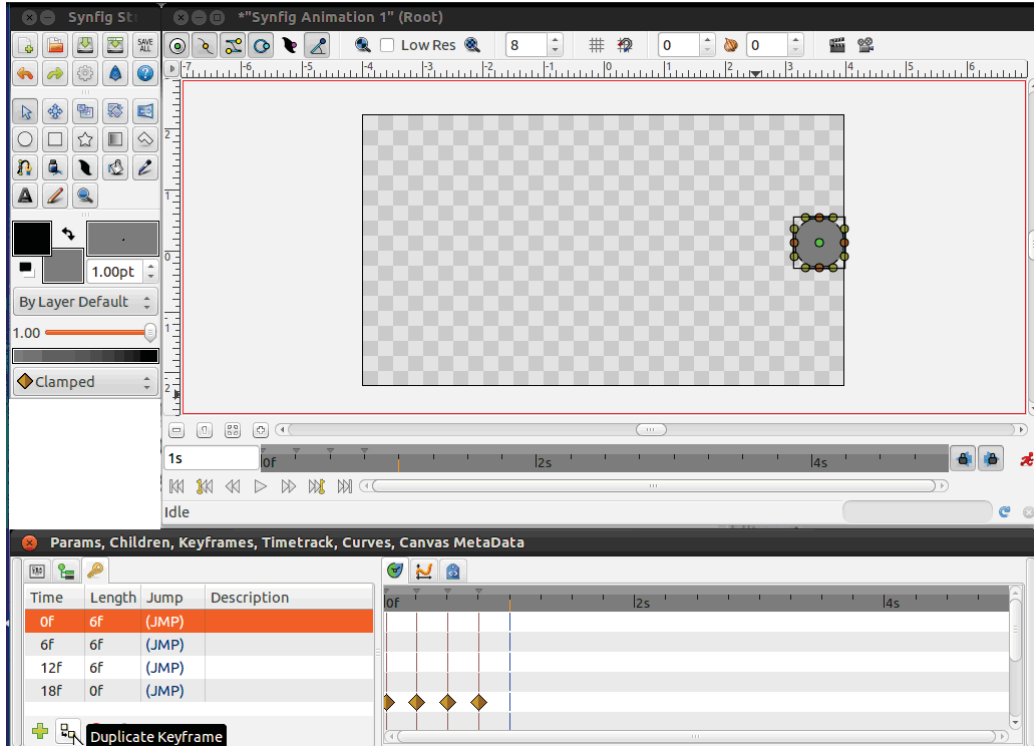


આકૃતિ 3.18 : 12f પર ઉમેરેલ કી-ફ્રેમ

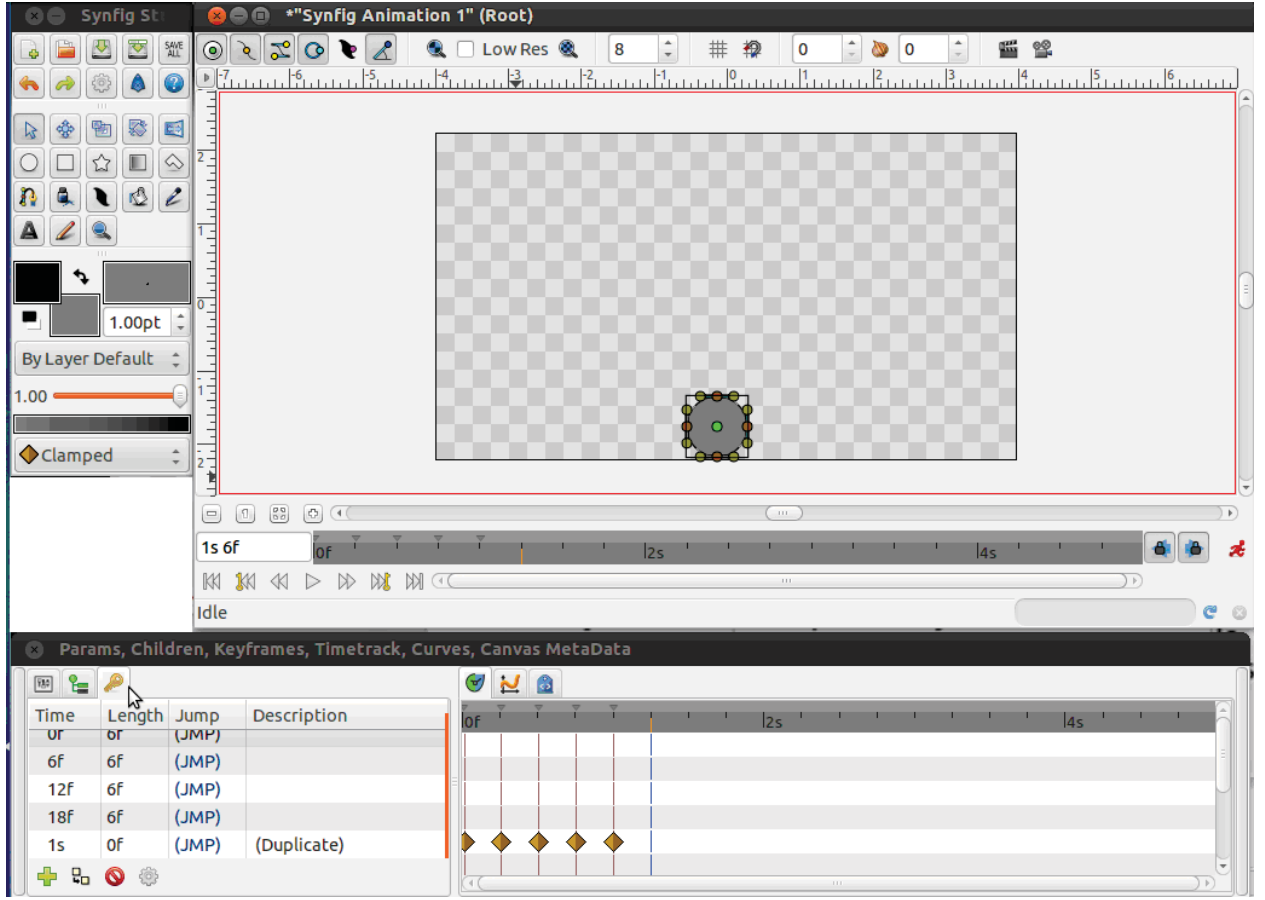


આકૃતિ 3.19 : 18f પર ઉમેરેલ કી-ફ્રેમ

- હવે ટાઈમલાઈનમાં “1s” પર પાછા આવો. અહીં દરો તેની મૂળ સ્થિતિમાં આવે કે જે “0f” પર હતી તે જરૂરી છે. માટે “0f” પરની કી-ફ્રેમની આપણે નકલ બનાવવી જરૂરી છે. કી-ફ્રેમ પેનલમાંથી “0f” કી-ફ્રેમ પસંદ કરીને આકૃતિ 3.20માં દર્શાવ્યા મુજબ Duplicate Keyframe પર ક્લિક કરો. આમ કરવાથી આકૃતિ 3.21માં દર્શાવ્યા મુજબ “1s” પર નવી ડુપ્લિકેટ કી-ફ્રેમ ઉમેરશે.

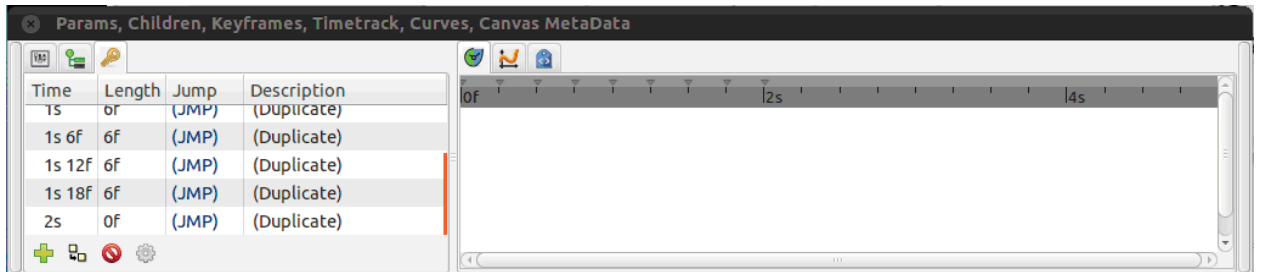


આકૃતિ 3.20 : 1s પર એક ‘ડુપ્લિકેટ કી-ફ્રેમ’ ઉમેરવી



આકૃતિ 3.21 : 'ડુપ્લિકેટ કી-ફ્રેમ' ઉમેર્યા પછી દર્શાવાતી પેનલ

- આ જ રીતે, દડાનું બે વાર એનિમેશન કરવા, આપણે 1s6f, 1s12f, 1s18f અને 2s પર ડુપ્લિકેટ કી-ફ્રેમ બનાવવી પડશે. આકૃતિ 3.22માં ડુપ્લિકેટ કી-ફ્રેમ ઉમેરાયેલી જોઈ શકાય છે.



આકૃતિ 3.22 : ડુપ્લિકેટ કી-ફ્રેમ ઉમેરવી

- હવે File → Preview કમાન્ડનો ઉપયોગ કરી એનિમેશનનું પૂર્વાવલોકન કરો. તમે દડાને બે સેકન્ડ સુધી બે વાર ઊછળતો જોઈ શકશો અને બે સેકન્ડ પછી તે રોકાઈ જશે. એનિમેશન છોક અંત સમય સુધી ચાલુ રહે તેવું આપણે ઈચ્છીએ છીએ. આપણા કિસ્સામાં અંતસમય 5 સેકન્ડ છે. આપણે Circle Layerની ઉપર એક નવું Timeloop Layer ઉમેરીશું.
- સૌથી ઉપરના Circle Outline સ્તર પસંદ કરો. એ પછી રાઈટ ક્લિક કરી Add new layer → Other → Timeloop પસંદ કરો. આમ કરવાથી આકૃતિ 3.23માં દર્શાવ્યા મુજબ એક નવું સ્તર (લેયર) ઉમેરશે.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Time Loop	0.000000
<input checked="" type="checkbox"/>	Circle001 Outline	1.000000
<input checked="" type="checkbox"/>	Circle001 Region	2.000000

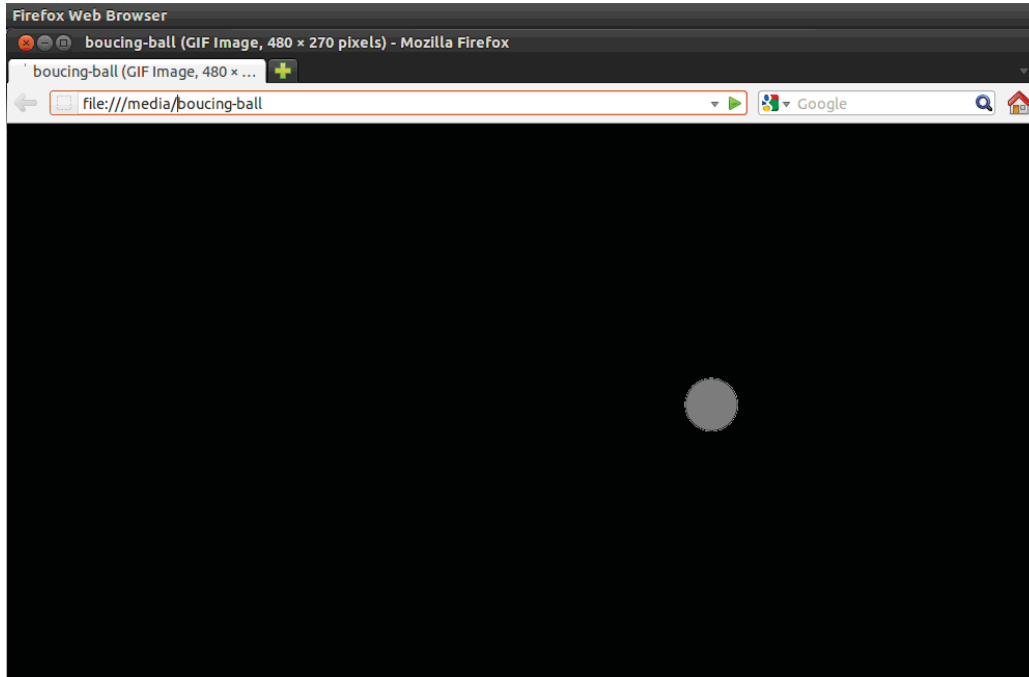
આકૃતિ 3.23 : Time loop લેયર ઉમેરવું

- આકૃતિ 3.24માં (Timeloop) ના ઘટકો સમય અવધિ દર્શાવે છે, જે કેટલી ફ્રેમ કે સેકન્ડ સુધી લૂપ ફરશે, તે દર્શાવે છે. આપણું એનિમેશન 2s સુધી બે વાર ચલાવાતું હોઈ આપણે તેને બદલીને 2s કરી શકીએ.

Name	Value	Type
π Z Depth	0.000000	real
<input checked="" type="checkbox"/> Link Time	0f	time
<input checked="" type="checkbox"/> Local Time	0f	time
<input checked="" type="checkbox"/> Duration	1s	time
<input type="checkbox"/> Only For Positive Duration		bool
<input checked="" type="checkbox"/> Symmetrical		bool

આકૃતિ 3.24 : Timeloopમાં સમયઅવધિ માટેના ઘટકો

- એનિમેશનનું પૂર્વાવલોકન કરો. તમે જોશો તો એનિમેશન છેક અંત સમય સુધી ચાલશે. જો તમે File properties વિન્ડોમાં અંત સમય બદલીને 10s કરશો, તો એનિમેશન 10 સેકન્ડ સુધી ચાલ્યા કરશે.
- ફાઈલને રેન્ડરિંગ કરો અને સાચવી લો. આકૃતિ 3.25 વેબબ્રાઉઝર ઉપરનું પરિણામ દર્શાવે છે.



આકૃતિ 3.25 : વેબબ્રાઉઝર ઉપર પરિણામ

સારાંશ

આ પ્રકરણમાં આપણે એનિમેશન કેવી રીતે બનાવાય તે શીખ્યા. Timebarના ઉપયોગ દ્વારા જુદાં-જુદાં સમયે ઓબ્જેક્ટના સ્થાનમાં કરાતા ફેરફારને નોંધવા એનિમેટ એડિટ મોડ મદદરૂપ થાય છે. જ્યારે “એનિમેટ એડિટ મોડ”માં નહીં હોય ત્યારે ઓબ્જેક્ટમાં કરવામાં આવતાં કોઈ પણ ફેરફાર અથવા તેનાં પરિબળોની એનિમેશન પર કોઈ અસર થશે નહીં. ‘કી-ફ્રેમ’ અને ‘વે-પોઈન્ટ’ના ખ્યાલની ચર્ચા કરવામાં આવી. TimeLoop સ્તરના ઉપયોગ દ્વારા અંત સમય સુધી એનિમેશનને કેવી રીતે પુનરાવર્તિત કરવું, તે પણ આપણે શીખ્યા. સ્તર (લેયર)ના ખ્યાલ વિશે અને ઓબ્જેક્ટ પર અસર આપવા તેનો ઉપયોગ કરવા અંગે આપણે આગળના પ્રકરણમાં શીખીશું.

સ્વાધ્યાય

1. એનિમેશનની વ્યાખ્યા આપી તેના પ્રકારો જણાવો.
2. ‘ફ્રેમ-બાય-ફ્રેમ’ અને ‘સેલ (cel) એનિમેશન’ વર્ણવો.
3. ‘કી-ફ્રેમ’ એટલે શું ?
4. ‘ટૂવીનિંગ’ એટલે શું ?
5. ‘એનિમેટ એડિટિંગ મોડ’ વર્ણવો.
6. ‘ટાઈમબાર’નો હેતુ જણાવો.
7. ‘મોર્ફિંગ’ એટલે શું ?
8. ‘કાઈનેમેટિક્સ’ એટલે શું ? ત્રણ ઉદાહરણ આપો.
9. હેતુલક્ષી પ્રશ્નો :
 - (1) નીચેના પૈકી કયો શબ્દ ઓબ્જેક્ટને અમુક સમયાંતરે દર્શાવવાની બાબતને રજૂ કરે છે ?
 - (a) ફ્રેમ
 - (b) ઓબ્જેક્ટ
 - (c) કી-ફ્રેમ
 - (d) પેનલ
 - (2) નીચેના પૈકી કઈ ટેકનિકમાં, છબીમાં પહેલાં દરેક નાના ફેરફાર કરવામાં આવે છે અને પછી ક્રમબદ્ધ રીતે દર્શાવવામાં આવે છે ?
 - (a) ફ્રેમ-બાય-ફ્રેમ એનિમેશન
 - (b) સેલ-એનિમેશન
 - (c) ટૂવીનિંગ
 - (d) મોર્ફિંગ
 - (3) અંતસમય માટે સામાન્ય રીતે નીચે પૈકી કયો સમય ગોઠવાયેલ હોય છે ?
 - (a) 2s
 - (b) 5s
 - (c) 3s
 - (d) 6s
 - (4) નીચેના પૈકી કઈ બાબત એક ફ્રેમ રજૂ કરે છે, જે ઓબ્જેક્ટના ગુણધર્મોમાં ફેરફાર નક્કી કરે છે ?
 - (a) ટાઈમલાઈન
 - (b) સ્તર (લેયર)
 - (c) કી-ફ્રેમ
 - (d) પેનલ
 - (5) નીચેના પૈકી કયું ‘વે-પોઈન્ટ’ રજૂ કરતો નિર્દેશક દર્શાવે છે ?
 - (a) વર્તુળ
 - (b) ચોરસ
 - (c) લંબચોરસ
 - (d) ચતુષ્કોણ (ડાયમન્ડ)

- (6) એનિમેટ એડિટ મોડમાં કેનવાસની ફરતે કિનારી (outline)નો રંગ કયો થઈ જાય છે?
- (a) નારંગી (b) પીળો
(c) લાલ (d) લીલો
- (7) નીચેનાં પૈકી કયું પદ દસ્તાવેજમાં સમયની લંબાઈ નક્કી કરે છે?
- (a) ફેમ (b) કી-ફેમ
(c) લેયર (d) ટૂલીનિંગ
- (8) નીચેનાં પૈકી કઈ સ્લાઈડબારની સાચી કિંમત છે ?
- (a) 3f9f (b) 3s9f
(c) 3f9s (d) 3s9s
- (9) નીચેનાં પૈકી કયું પદ અસરને રજૂ કરે છે, જેમાં એક છબી બીજી છબીમાં પરિવર્તિત થાય છે?
- (a) મોર્ફિંગ (b) ટૂલીનિંગ
(c) કાઈનેમેટિક્સ (d) કી-ફેમ
- (10) ફેમ-પ્રતિ-સેકન્ડ માટે સામાન્ય રીતે નીચેના પૈકી કઈ ગોઠવણ હોય છે ?
- (a) 15 (b) 30
(c) 24 (d) 28
- (11) નીચેનાં પૈકી કયું પદ સાંધા હોય તેવા ઓબ્જેક્ટની ગતિ અને હલનચલનનો અભ્યાસ રજૂ કરે છે ?
- (a) એનિમેશન (b) કાઈનેમેટિક્સ
(c) ટૂલીનિંગ (d) મોર્ફિંગ

પ્રાયોગિક સ્વાધ્યાય

1. ઉપરથી નીચે સરકતા જતા તારા (star)નું એનિમેશન તૈયાર કરો.
2. એક એવું એનિમેશન તૈયાર કરો કે જેમાં એક દડો ભોંયતળિયે ઊછળતો હોય. અંતસમય સુધી એનિમેશન ચાલુ રાખવા ટાઈમ લૂપનો ઉપયોગ કરો.
3. વર્તુળ ડાબેથી જમણી તરફ જતું હોય અને એ જ વખતે તારો (star) જમણી બાજુથી ડાબી બાજુ જતો હોય, તેવું એનિમેશન તૈયાર કરો.
4. અલગ-અલગ ઝડપ સાથે ઊછળતા ત્રણ દડાનું એનિમેશન તૈયાર કરો.





ઊગતા સૂર્યનું સુંદર દૃશ્ય હંમેશાં નયનરમ્ય લાગે છે. એમાંય વળી જો આ સૂર્યોદય પર્વતોની વચ્ચે જોવા મળે, તો દૃશ્ય વધુ આનંદદાયક બને છે. આ દૃશ્ય ઘણી બધી બાબતો કે વસ્તુઓ (ઓબ્જેક્ટ)નું બનેલું છે. જેમકે, સૂરજ, પર્વતની ટેકરીઓ, પક્ષીઓ, વૃક્ષો વગેરે. ધારોકે, આપણે આવા સૂર્યોદયના દૃશ્યનું એનિમેશન બનાવવું છે. હવે એ તો તમે સમજી જ શકો કે, કેનવાસ પર જ્યારે આપણે આ દૃશ્ય દોરીએ ત્યારે, આ દરેક ઓબ્જેક્ટ કાં તો કોઈ ઓબ્જેક્ટની આગળ હોય કે પાછળ હોય. ઉદાહરણ તરીકે, સૂર્ય (sun) પર્વતો (mountains)ની પાછળથી ઉદય પામે છે. આમ ‘sun’ નામનો ઓબ્જેક્ટ ‘mountain’ નામના ઓબ્જેક્ટની પાછળ જ મૂકવો પડે. એ જ રીતે જો પર્વતની ટેકરીઓ ઝૂંપડી (hut)ની પાછળ હોય તો ‘hut’ નામનો ઓબ્જેક્ટ ‘mountain’ ઓબ્જેક્ટની ઉપર મૂકવો જોઈએ. આ પ્રમાણે આ ઉદાહરણમાં આપણે ઝાડ, પક્ષીઓ વગેરે જેવા અન્ય ઓબ્જેક્ટને કોઈ ઓબ્જેક્ટની આગળ કે પાછળ મૂકી શકીએ. આ પ્રકરણમાં આપણે ‘લેયર’ (સ્તર) નામના એક વિશિષ્ટ ખ્યાલ વિશે શીખીશું, જે આપણને ઉપર મુજબનું દૃશ્ય સર્જવામાં મદદરૂપ બનશે.

સ્તર (Layers)

સીન્ફિગમાં જુદાં-જુદાં સ્તર કે જેના ઉપર આપણે આપણાં ચિત્રો કે ઓબ્જેક્ટ મૂકી શકીએ, તેને સ્તર (લેયર) તરીકે ઓળખવામાં આવે છે. આવાં સ્તર એકબીજાની ઉપર કે નીચે ઉપરા-છાપરી રાખવામાં આવે છે. સ્વાભાવિક રીતે જ ઉપરનું સ્તર નીચેના સ્તરને ઢાંકી દે છે. સ્તરને આપણે જુદા-જુદા કાગળ સ્વરૂપે કલ્પી શકીએ કે જે જુદા-જુદા આકારમાં કપાયેલા હોય. એક સ્તર પર આપણી પાસે તારા (star)ના આકારમાં કાપેલ કાગળ છે. બીજા સ્તર પર ચોરસ આકારમાં કપાયેલ કાગળ છે. શક્ય છે કે, ચોરસ કાગળ તારાના કોઈક ભાગને ઢાંકી પણ દેતો હોય.

સ્તરનો ઉપયોગ આપણને દરેક ઓબ્જેક્ટ સાથે અલગ રીતે કાર્ય કરવાની સ્વતંત્રતા આપે છે. જ્યારે કોઈ ઓબ્જેક્ટને સુધારવામાં આવે ત્યારે તે તેની ઉપરના કે નીચેના અન્ય ઓબ્જેક્ટને જરાય અસર પહોંચાડતું નથી. સ્તરોની પુનઃગોઠવણી પણ કરી શકાય છે, અને એ રીતે બધા ઓબ્જેક્ટ કયા ક્રમમાં દર્શાવાય તે ક્રમ બદલવાની આપણને છૂટ આપે છે. આમ, આપણે આપણા દૃશ્યના દેખાવને બદલી શકીએ છીએ.

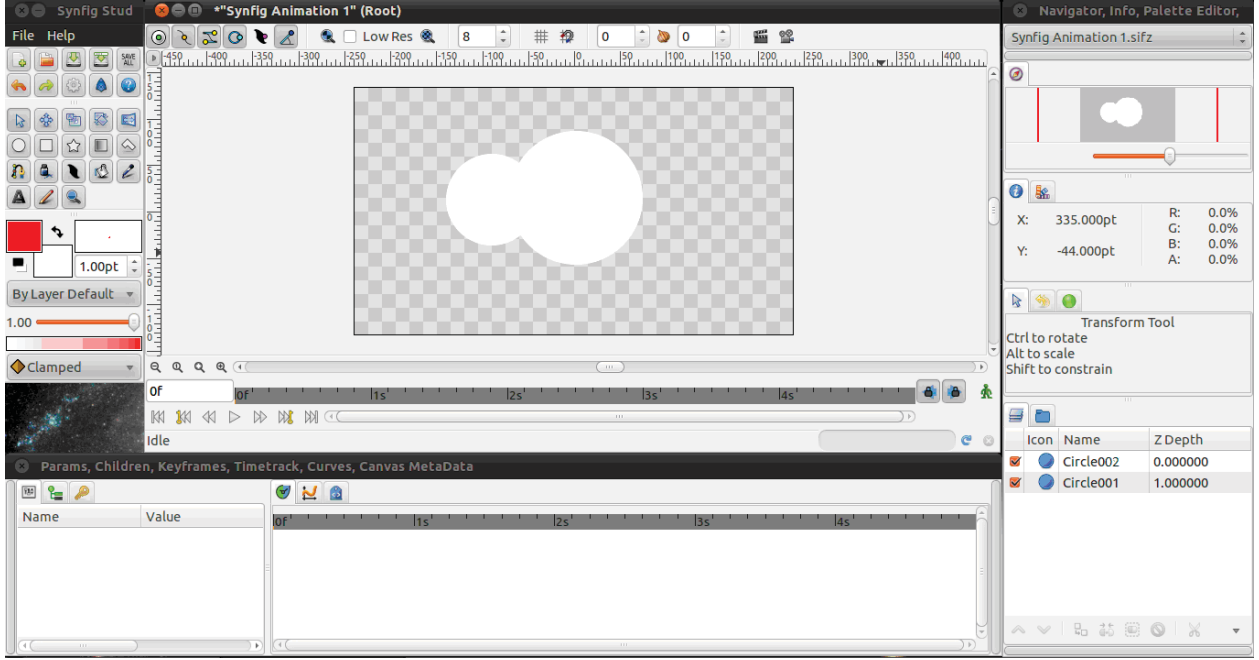
સીન્ફિગમાં કેનવાસના દરેક અલગ-અલગ તત્ત્વ અલગ-અલગ સ્તરોમાં વિભાજિત કરવામાં આવે છે. ઉદાહરણ તરીકે, જ્યારે આપણે વર્તુળ દોર્યું, ત્યારે તે એક અલગ સ્તર પર દર્શાવાયું. સીન્ફિગના સ્તર અન્ય એનિમેશન સોફ્ટવેરના સ્તર કરતાં નીચેની બે બાબતે જુદું પડે છે :

1. દરેક ઓબ્જેક્ટ, તત્ત્વ અને અસરોને તેમનું અલાયદું સ્તર હોય છે, અનેકવિધ અસર સાથેનું એક સ્તર હોતું નથી. બીજા શબ્દોમાં કહીએ તો સ્તર કોઈ એક બાબત જેમકે, ઓબ્જેક્ટ, આઉટલાઈન કે છબી જ રજૂ કરે છે.
2. ઉપરનું સ્તર તેની નીચેના સ્તરના વર્તનમાં ફેરફાર કરી શકે. તે નીચેના સ્તરને સુધારી શકે કે તેને વિકૃત કરી શકે. અસરનો કે ‘ફિલ્ટર-લેયર’નો ઉપયોગ કરીને આવું કરી શકાય.

જુદા-જુદા પ્રકારનાં સ્તર હોય છે જેવાં કે ‘ગ્રેડિયન્ટ’, ‘બ્લર’, ‘ડિસ્ટોર્સન’, ‘ફિલ્ટર’ અને અન્ય. દરેક સ્તરને તેનાં પોતાનાં પરિબળો હોય છે. કોઈ પણ સ્તરને Layers Panelમાંથી પસંદ કરીને પછી Params Panelમાં જઈને આપણે જે-તે સ્તરનાં પરિબળો જોઈ

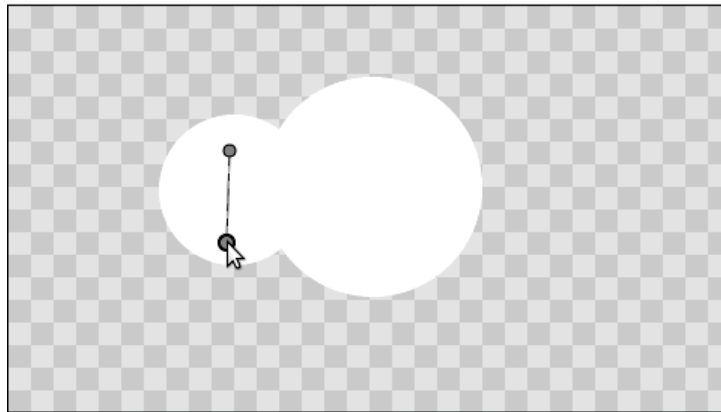
શકીએ છીએ. તો ચાલો, ઉદાહરણની મદદથી સ્તર (લેયર)ના ખ્યાલને સમજીએ. ધારોકે, આપણે વર્તુળ ધરાવતા સ્તરને ધીરે ધીરે આછા થતા રંગ ગ્રેડિયન્ટ જેવી (gradient) અસર આપવી છે. ગ્રેડિયન્ટ જેવી અસર મેળવવા માટે નીચે મુજબનાં પગલાં અનુસરો :

- નવી ફાઇલ બનાવો.
- Circle ટૂલનો ઉપયોગ કરીને બે વર્તુળ દોરો. વર્તુળનો રંગ સફેદ જ રાખો કે જેથી કરીને ગ્રેડિયન્ટની અસર જોઈ શકાય. આકૃતિ 4.1 આ બે વર્તુળ દર્શાવે છે.



આકૃતિ 4.1 : જુદાં-જુદાં કદનાં બે વર્તુળ

- ટૂલબોક્સમાંથી Gradient ટૂલ પસંદ કરો. outline color માટે તમારી પસંદગી મુજબનો રંગ પસંદ કરો. અહીં આપણે લાલ રંગ પસંદ કર્યો છે. આકૃતિ 4.2માં દર્શાવ્યા મુજબ Gradient બિંદુને ઘસડો.



આકૃતિ 4.2 : ગ્રેડિયન્ટ ઉમેરવો

- આકૃતિ 4.3માં ગ્રેડિયન્ટ રંગ દ્વારા લગભગ આખું કેનવાસ રંગાઈ ગયેલું આપણે જોઈ શકીએ છીએ. જો તમે બેકગ્રાઉન્ડ સ્તર તરીકે ગ્રેડિયન્ટ રાખવા માંગતા હોય, તો આ ખૂબ જ ઉપયોગી બની શકે. પરંતુ આપણે તો ગ્રેડિયન્ટની અસર માત્ર ઓબ્જેક્ટ પર જ થાય તેવું ઇચ્છીએ છીએ.

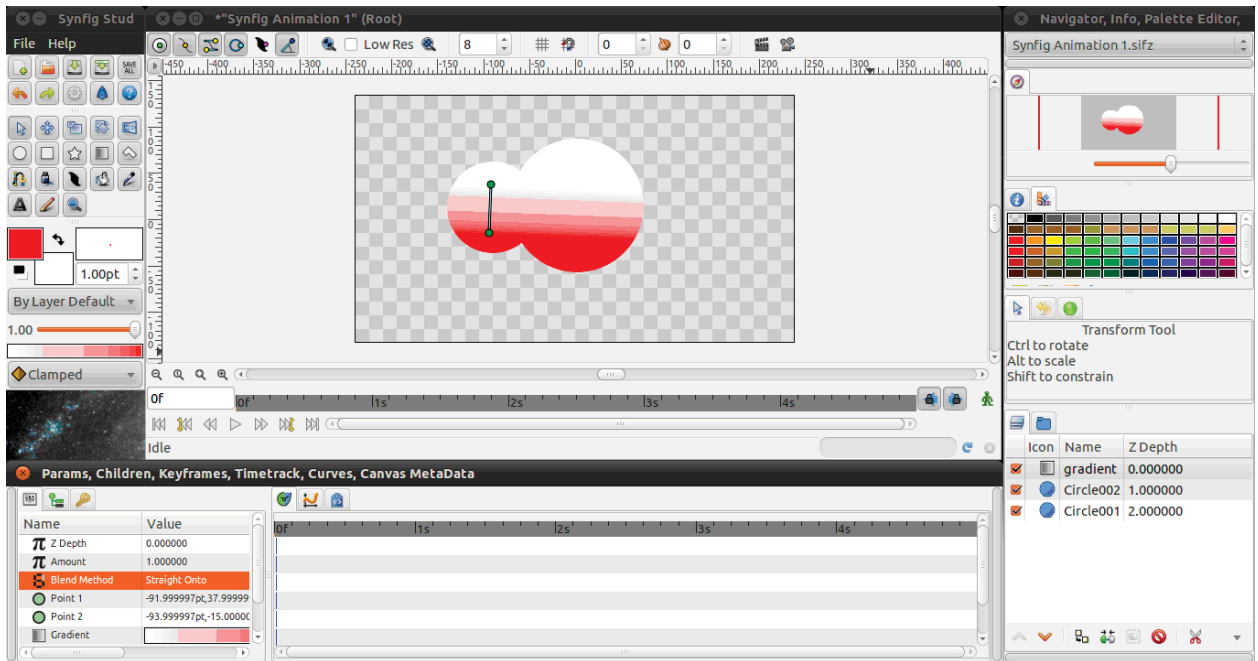


આકૃતિ 4.3: કેનવાસ પર ગ્રેડિયન્ટની અસર

- ગ્રેડિયન્ટ સ્તરને પસંદ કરો અને Params ભાગ પર જાવ ત્યાં તમને blend method નામનું પ્રાચલ (પેરામીટર) જોવા મળશે. 'બ્લેન્ડ (Blend) મોડ' એ નક્કી કરે છે કે કેવી રીતે છબીને તેની નીચે આવેલા ઘટકો સાથે ગોઠવવામાં આવે. સામાન્ય રીતે 'Composite' ગોઠવણ હોય છે, જે સ્તર પર રહેલી વિગતો દર્શાવે છે. Blend પદ્ધતિમાં આકૃતિ 4.4માં દર્શાવ્યા પ્રમાણે પરિબળો ગોઠવણીને બદલીને 'straight onto' કરી દે છે. તમે એ જોઈ શકો છો કે, ગ્રેડિયન્ટનો ઉપયોગ હવે ઓબ્જેક્ટ લેયરની પાછળ કરવામાં આવ્યો, પરંતુ આપણે એવું ઈચ્છીએ છીએ કે, ગ્રેડિયન્ટ માત્ર એક જ ઓબ્જેક્ટ સાથે ઉપયોગમાં આવે તેમજ ગ્રેડિયન્ટ સ્તરની નીચેના બધા ઓબ્જેક્ટ પર નહીં. આ માટે આપણે 'એનકેપ્સ્યુલેશન' (encapsulation)નો ઉપયોગ કરીશું.

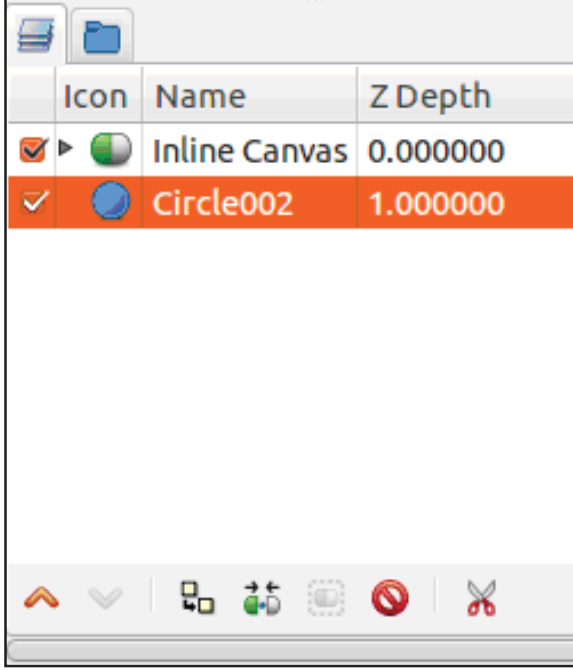
એનકેપ્સ્યુલેશન (પ્રાવૃત્ત) એટલે જુદી-જુદી વસ્તુઓને જૂથમાં ભેગી (ગ્રુપિંગ) કરવી. જો આપણે એવું ઈચ્છતા હોઈએ કે, કોઈ ચોક્કસ ઓબ્જેક્ટ અથવા ઓબ્જેક્ટના અમુક સમૂહ પર ગ્રેડિયન્ટનો અમલ થાય તો આપણે તેવા ઓબ્જેક્ટને એક સ્તરમાં ભેગા (એનકેપ્સ્યુલેટ) કરવા પડે. આમ, એનકેપ્સ્યુલેશનના ઉપયોગ દ્વારા આપણે ગ્રેડિયન્ટને ફક્ત એક સ્તર માટે, અમલી બનાવી શકીએ. (અને એ પણ આ સ્તરની નીચેનાં સ્તરોને અસર આપ્યા વગર).

આમ, આપણા ઉદાહરણમાં આપણે ગ્રેડિયન્ટ અને જેની ઉપર ગ્રેડિયન્ટની અસર આપવી છે, તે ઓબ્જેક્ટ પસંદ કરવા પડે અને તે બધાને ભેગા (એનકેપ્સ્યુલેટ) કરવા પડે.

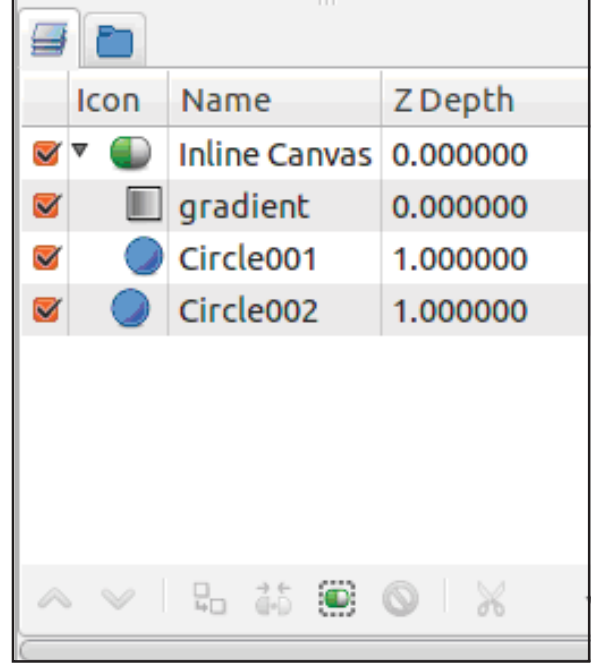


આકૃતિ 4.4 'Straight onto' બ્લેન્ડ પદ્ધતિનો ઉપયોગ

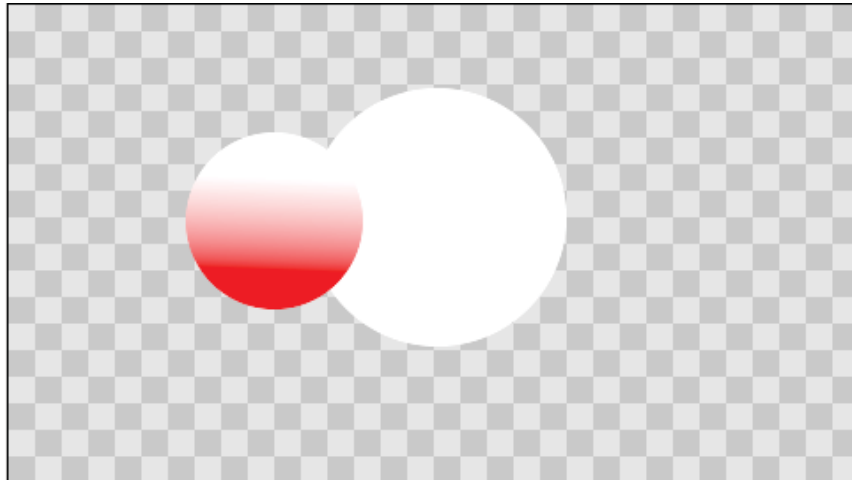
- જેના પર આપણે ગ્રેડિયન્ટની અસર આપવી છે, તે ગ્રેડિયન્ટ સ્તર અને સર્કલ લેયરને પસંદ કરો (બે સ્તરને પસંદ કરવા CTRL કી દબાવીને પકડી રાખો.) જેના પર આપણે ગ્રેડિયન્ટની અસર આપવી છે. માઉસનું જમણું બટન દબાવીને Encapsulate પસંદ કરો. આમ કરવાથી આકૃતિ 4.5માં દર્શાવ્યા પ્રમાણે ‘inline canvas’ નામનું નવું સ્તર બનશે. આ Inline canvasની નજીક તમે એક નાનો ત્રિકોણ જોઈ શકશો. આ ત્રિકોણ પર ક્લિક કરશો એટલે આકૃતિ 4.6માં દર્શાવ્યા મુજબ ‘ઇનલાઇન કેનવાસ’, ગ્રેડિયન્ટ અને સર્કલ લેયર એમ બન્ને સ્તરને દર્શાવવા વિસ્તરીને મોટું થશે. આકૃતિ 4.7માં માત્ર નાના વર્તુળમાં ગ્રેડિયન્ટની અસર સાથેનું કેનવાસ દર્શાવાય છે.



આકૃતિ 4.5 : ઇનલાઇન કેનવાસ સ્તર

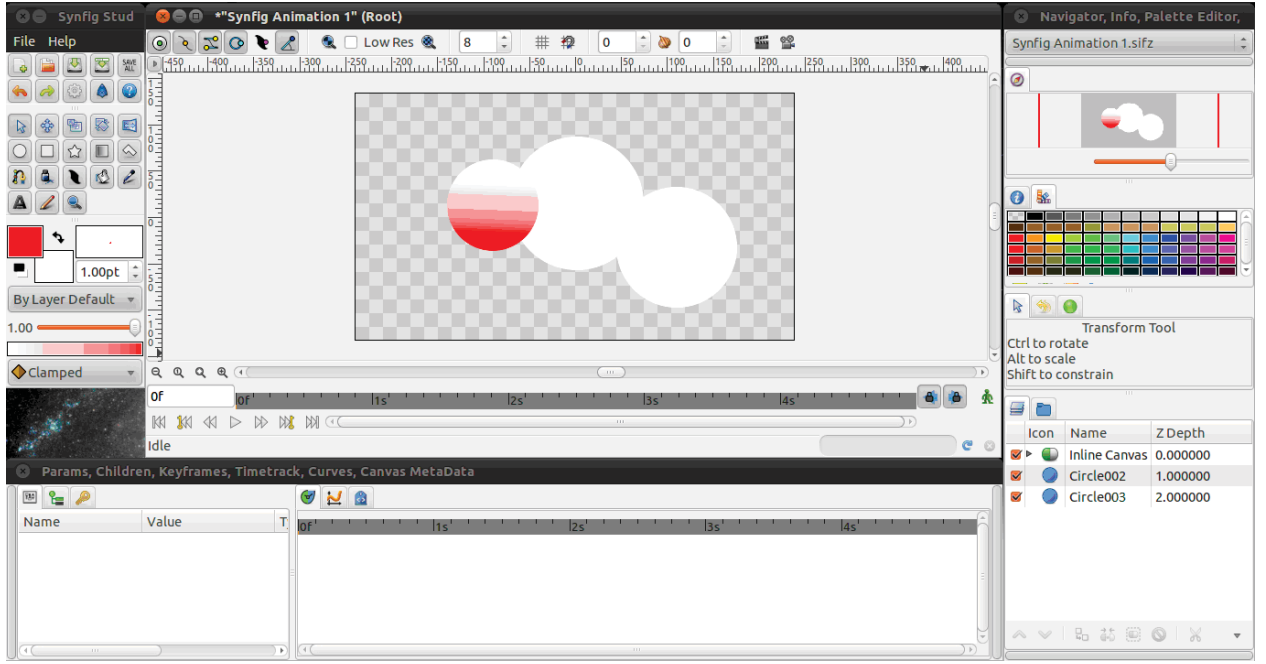


આકૃતિ 4.6 : વિસ્તરેલું ઇનલાઇન કેનવાસ લેયર



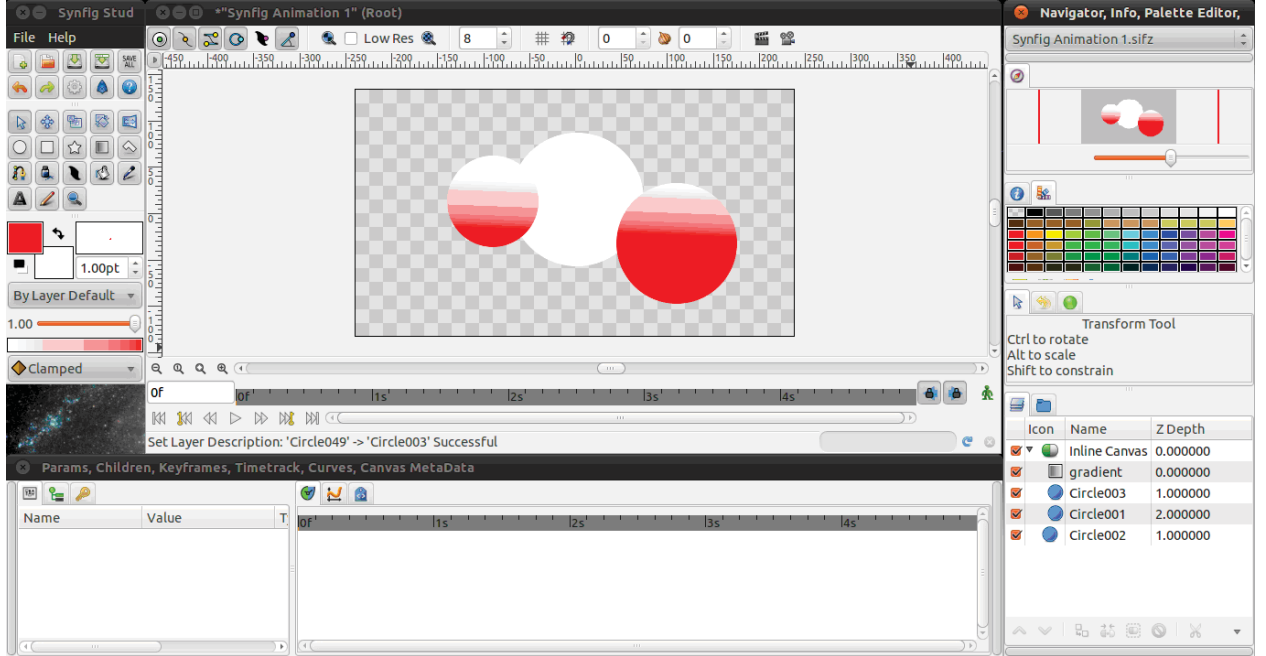
આકૃતિ 4.7 : એનકેપ્સ્યુલેશન પછી ગ્રેડિયન્ટ અસર દર્શાવતું કેનવાસ

આમ, આપણે એ જાણ્યું કે, એનકેપ્સ્યુલેશનનો ઉપયોગ કરીને કેવી રીતે ઓબ્જેક્ટ પર ગ્રેડિયન્ટની અસર આપી શકાય છે. આને વધુ સ્પષ્ટ કરવા આકૃતિ 4.8માં દર્શાવ્યા પ્રમાણે ચાલો circle ટૂલનો ઉપયોગ કરી એક વધુ વર્તુળ ઉમેરીએ.



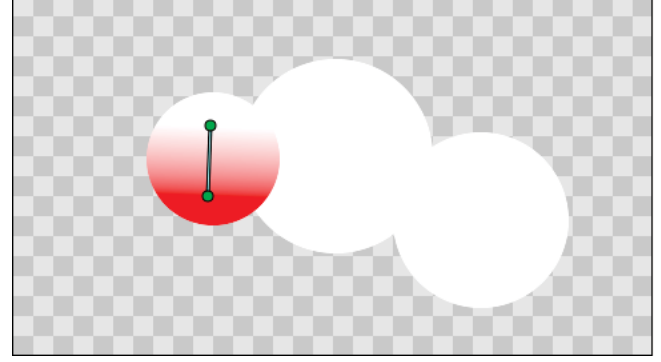
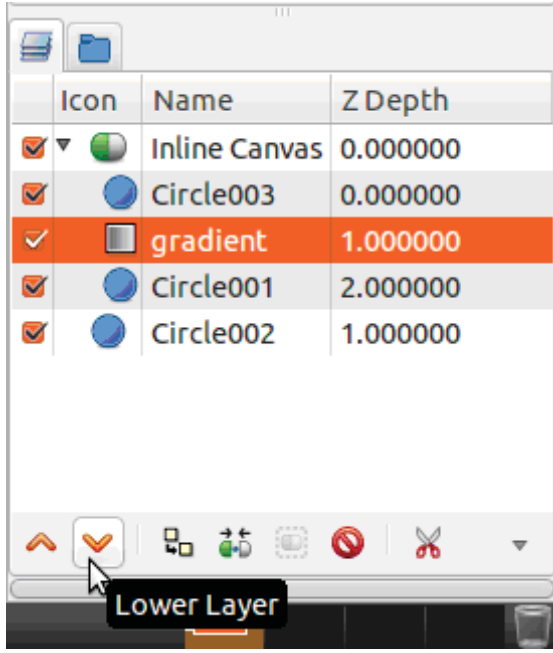
આકૃતિ 4.8 : નવા સ્તર સાથે નવું વર્તુળ

હવે, આ નવા સર્કલ લેયરને ઈનલાઈન કેનવાસ સ્તરમાં ઘસડીને લઈ જાવ. આપણા સર્કલ લેયરને બિલકુલ ગ્રેડિયન્ટ સ્તરની નીચે મૂકીશું, અને એની અસર આકૃતિ 4.9 માં દર્શાવ્યા મુજબ દેખાશે. અહીં તમે એ જોઈ શકશો કે, ગ્રેડિયન્ટની અસર એનકેમ્પ્યુલેટેડ વિસ્તારની અંદરના ઓબ્જેક્ટને આપવામાં આવી છે.



આકૃતિ 4.9 : ગ્રેડિયન્ટ સ્તરની નીચે મૂકવામાં આવેલ નવું સર્કલ લેયર

જો આપણે આકૃતિ 4.10 (a)માં દર્શાવ્યા પ્રમાણે Lower Layer બટનનો ઉપયોગ કરીને જો આપણે ગ્રેડિયન્ટ સ્તરની ફેરગોઠવણી કરીએ તો, એ પછી ગ્રેડિયન્ટ સ્તરની માત્ર એકાદ સર્કલ લેયરને અસર અપાશે. બીજા શબ્દોમાં આપણે કહી શકીએ કે, ગ્રેડિયન્ટ સ્તરની અસર માત્ર એ જ સ્તર પર થશે કે જે એનકેમ્પ્યુલેટેડ વિસ્તારમાં તેની નીચે આવેલા હોય. આકૃતિ 4.10 (b) આ ફેરગોઠવણીની અસર દર્શાવે છે.



આકૃતિ 4.10 (a) :

ગ્રેડિયન્ટ સ્તરની ફેરગોઠવણી

આકૃતિ 4.10 (b) :

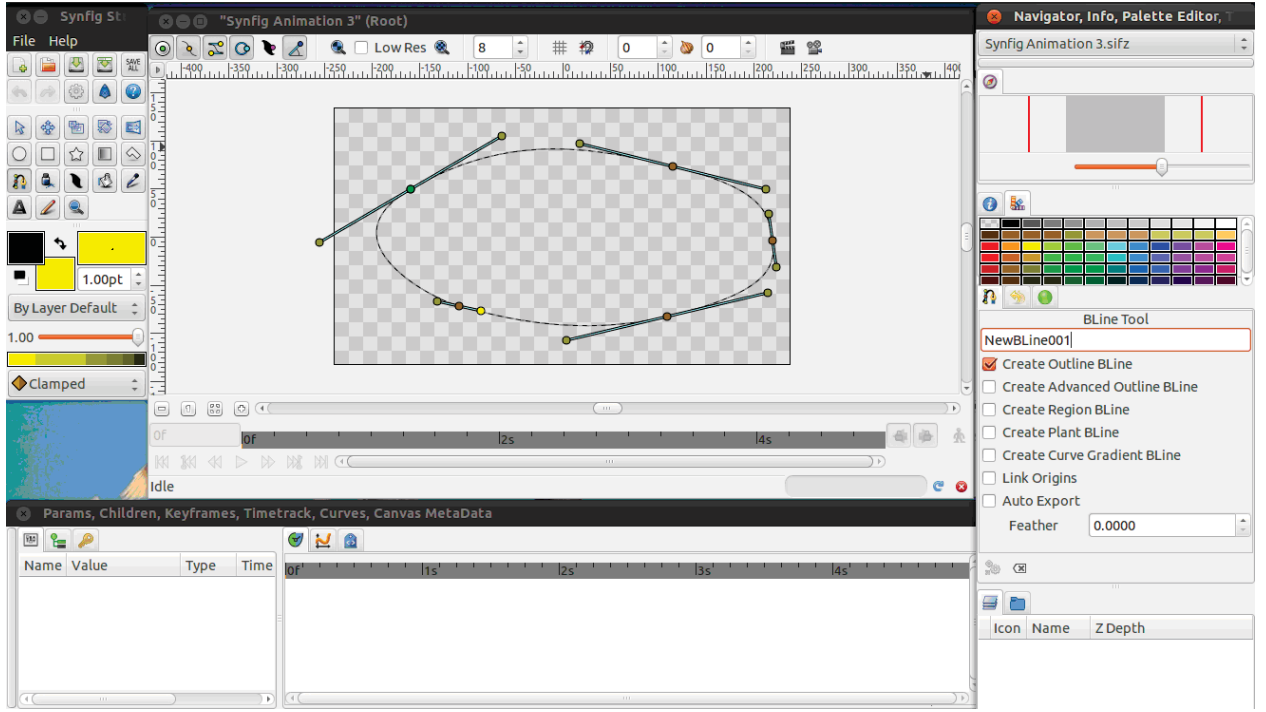
ગ્રેડિયન્ટ સ્તરની ફેરગોઠવણીની અસર

અહીં યાદ રાખવા જેવો અગત્યનો મુદ્દો એ છે કે, સ્તર માત્ર એ માહિતીમાં સુધારો કરી શકે છે કે, જે તેની તરત નીચેના સ્તર પાસેથી મળે. ઉદાહરણ તરીકે, જો આપણે ‘ઇનલાઇન કેનવાસ’ની અંદર, સ્તરની ઉપર Blur Layer (બ્લર લેયર)નો ઉપયોગ કર્યો હોય, તો તે માત્ર ‘ઇનલાઇન કેનવાસ’ સ્તરને જ થોડું ઝાંખું (બ્લર) કરશે, નહીં કે, ઇનલાઇન કેનવાસ હેઠળ આવેલા તમામ સ્તરને.

નિર્ધારિત પથ પર ચકાકાર ભ્રમણ

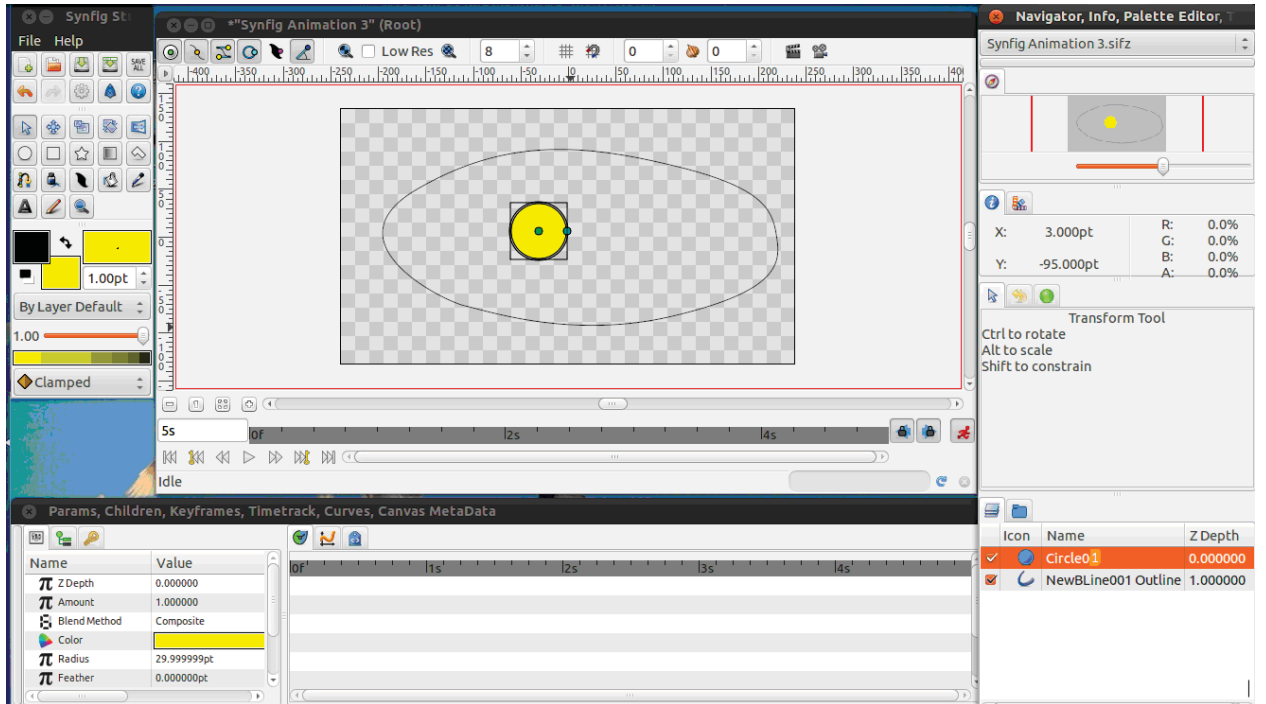
એનકેપ્સ્યુલેશનના ખ્યાલનો ઉપયોગ કરીને, ચાલો ચોક્કસ પથની આસપાસ ફરતા દડાનું અન્ય ઉદાહરણ ચર્ચા કરીએ. અગાઉના પ્રકરણમાં વર્ણવેલ ડાબી બાજુથી જમણી બાજુ તરફ સરકતા વર્તુળના આપણા એનિમેશનના ઉદાહરણમાં આપણે શરૂઆતનું સ્થાન અને અંતનું સ્થાન દર્શાવેલ. સીન્કિંગ પથ સીધો હોવાનું આપમેળે ધારી લે છે. પરંતુ અહીં આપણા ઉદાહરણમાં તો આપણે આપણા ઓબ્જેક્ટને આપણા નિર્દિષ્ટ પથ પર ફેરવવું છે. ઉપયોગકર્તા દ્વારા નિર્ધારિત પથ બનાવવા નીચે મુજબનાં પગલાં અનુસરો :

- નવી ફાઇલ બનાવો.
- તો ચાલો, Bline ટૂલનો ઉપયોગ કરીને સૌપ્રથમ આપણે ચકાકાર ભ્રમણ માટેનો આપણો પથ તૈયાર કરીએ. Bline ટૂલ પસંદ કરીને આકૃતિ 4.11માં દર્શાવ્યા પ્રમાણેનો લંબગોળ આકારનો પથ દોરો. Tool વિકલ્પમાં માત્ર *Create Outline Bline* વિકલ્પ પસંદ કરો.



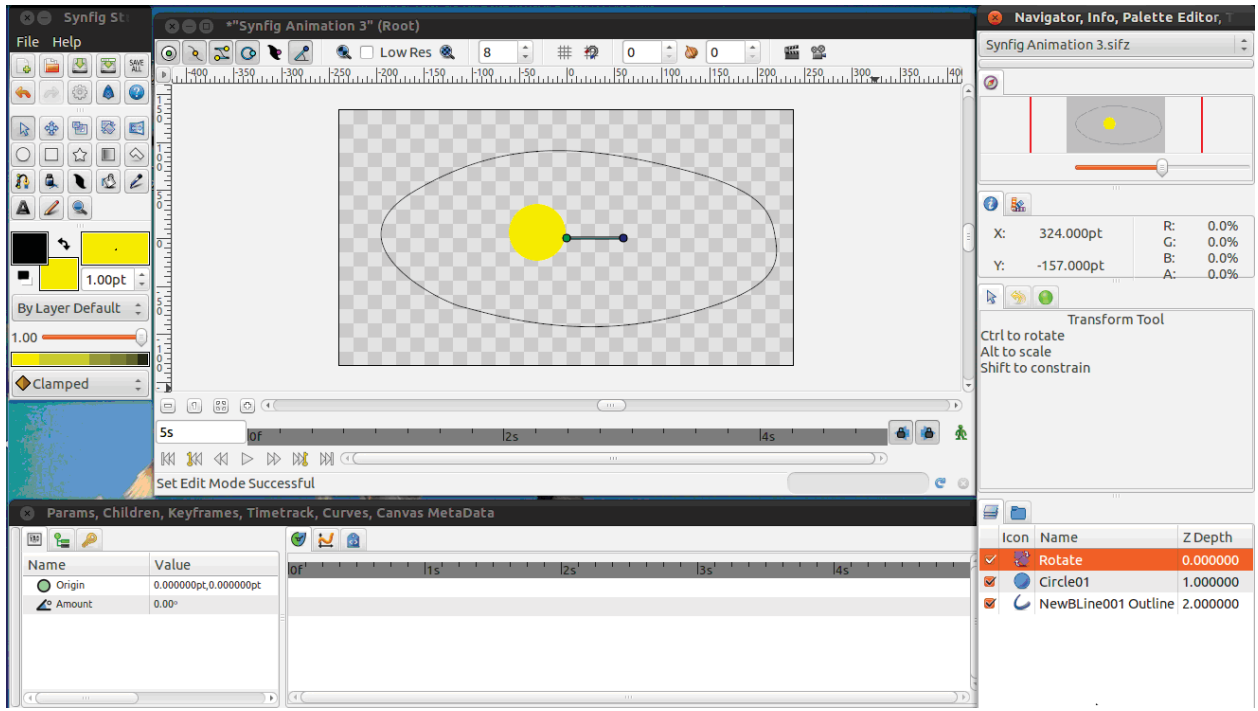
આકૃતિ 4.11 : Bline ટૂલ વડે પથ તૈયાર કરવો

- Blineને બંધ કરવા Transform ટૂલ પસંદ કરો. (Esc કી દબાવવાથી Blineને દૂર કરાશે.)
- હવે Circle ટૂલને પસંદ કરો. Palette Editorનો ઉપયોગ કરીને તમને ગમતો રંગ પસંદ કરો અને આકૃતિ 4.12માં દર્શાવ્યા મુજબ કેનવાસ પર વર્તુળ દોરો.



આકૃતિ 4.12 : વર્તુળ બનાવવું

- ચાલો, Circle Layerની ઉપર એક નવું Rotate Layer ઉમેરીએ. layers પેનલમાં Circle Layer → Right Click → New layer → Transform → Rotate પસંદ કરો. આમ કરવાથી આકૃતિ 4.13 (a) અને 4.13 (b) માં દર્શાવ્યા પ્રમાણે Circle Layerની ઉપર rotate નામનું એક નવું સ્તર ઉમેરાશે.



આકૃતિ 4.13 (a) : Circle Layerની ઉપર નવું Rotate Layer

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Rotate	0.000000
<input checked="" type="checkbox"/>	Circle01	1.000000
<input checked="" type="checkbox"/>	NewBLine001 Outline	2.000000

આકૃતિ 4.13 (b) : Circle Layerની ઉપર Rotate Layer દર્શાવતી Layers Panel

આપણે વર્તુળને ફેરવવાનું હોવાથી આપણે Circle Layerની ઉપર Rotate Layer ઉમેર્યું છે.

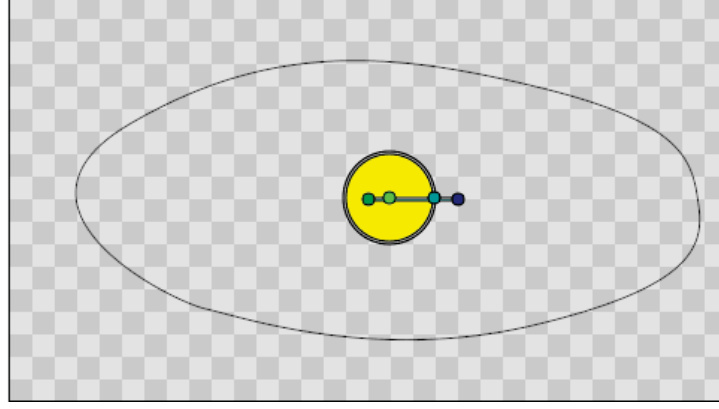
- હવે આપણે Rotate Layer અને Circle Layer બન્નેને ભેગા (એનકેપ્સ્યુલેટ) કરીશું. આ બન્ને સ્તરોને પસંદ કરવા Layers panelમાં Rotate Layer પર ક્લિક કરો, shift કી દબાવી રાખીને Circle Layer પસંદ કરો.
- માઉસનું જમણું બટન દબાવી Encapsulate પસંદ કરો. આમ, Rotate અને Circle Layer બન્ને inline canvas નામના સ્તરમાં

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	Rotate	0.000000
<input checked="" type="checkbox"/>	Circle01	1.000000
<input checked="" type="checkbox"/>	NewBLine001 Outline	1.000000

આકૃતિ 4.14 : ભેગાં થયેલ Rotate અને Circle સ્તર

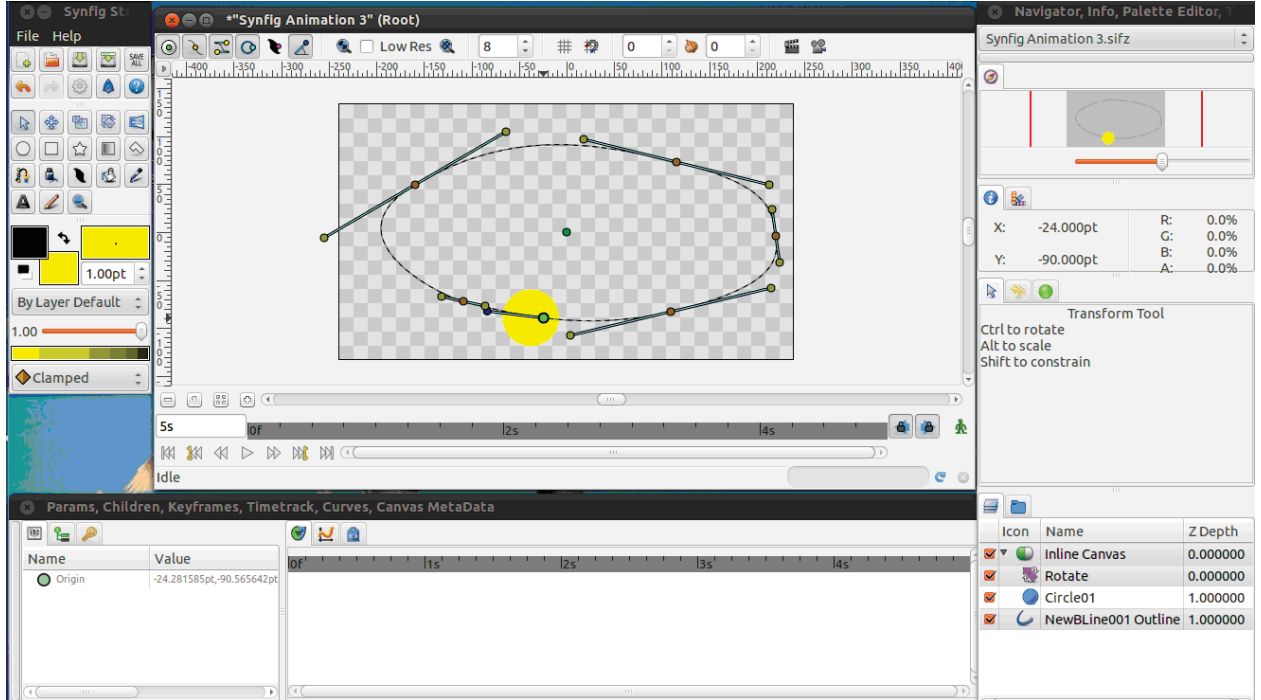
ભેગાં (એનકેપ્સ્યુલેટ) થઈ ગયાં. આ રીતે બનેલા એનકેપ્સ્યુલેટેડ સ્ટરનું નામ તમે બદલી શકો. આકૃતિ 4.14માં આ એનકેપ્સ્યુલેટેડ સ્ટર Inline Canvas દર્શાવાય છે.

- Rotate અને Circle સ્ટરને પસંદ કરીને તેના Green Duckને આકૃતિ 4.15માં દર્શાવ્યા પ્રમાણે એકબીજાંની નજીક લઈ આવો.



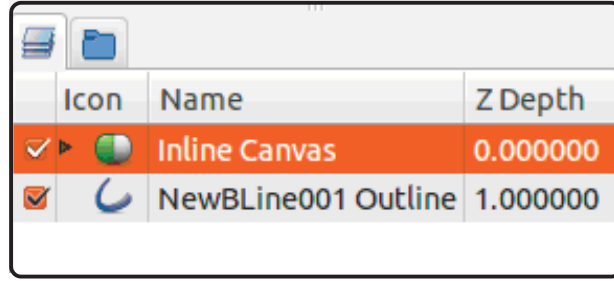
આકૃતિ 4.15 : Green Ducksને એકબીજાંની નજીક ઘસડવા

- Inline Canvas પસંદ કરો અને તેનાં 'ગ્રીન ડક'ને પસંદ કરો. CTRL કી દબાવી રાખીને Rotation Layerને પસંદ કરો. CTRL કી દબાવેલી રાખીને Rotation Layer પરનું Blue Duck પસંદ કરો. CTRL કી દબાવી રાખીને Bline સ્ટર પસંદ કરો. હવે કેનવાસમાં જ્યાંથી વર્તુળ શરૂ થાય તેમ ઈચ્છો છો, ત્યાં Bline પરના માઉસનું જમણું બટન દબાવો, 'Link to Bline' વિકલ્પ પસંદ કરો.
- Link to Bline લાક્ષણિકતા ઓબ્જેક્ટના શિરોબિંદુ (vertex)ને Blineના શિરોબિંદુ સાથે જોડવા માટે ઉપયોગમાં લેવાય છે. શિરોબિંદુ Blineના શિરોબિંદુ સાથે જોડાઈ જાય છે અને તેને તેની સાથે જ ખસેડી શકાય છે. Bline વિકલ્પની કડીના ઉપયોગની અસર આકૃતિ 4.16માં દર્શાવાય છે.



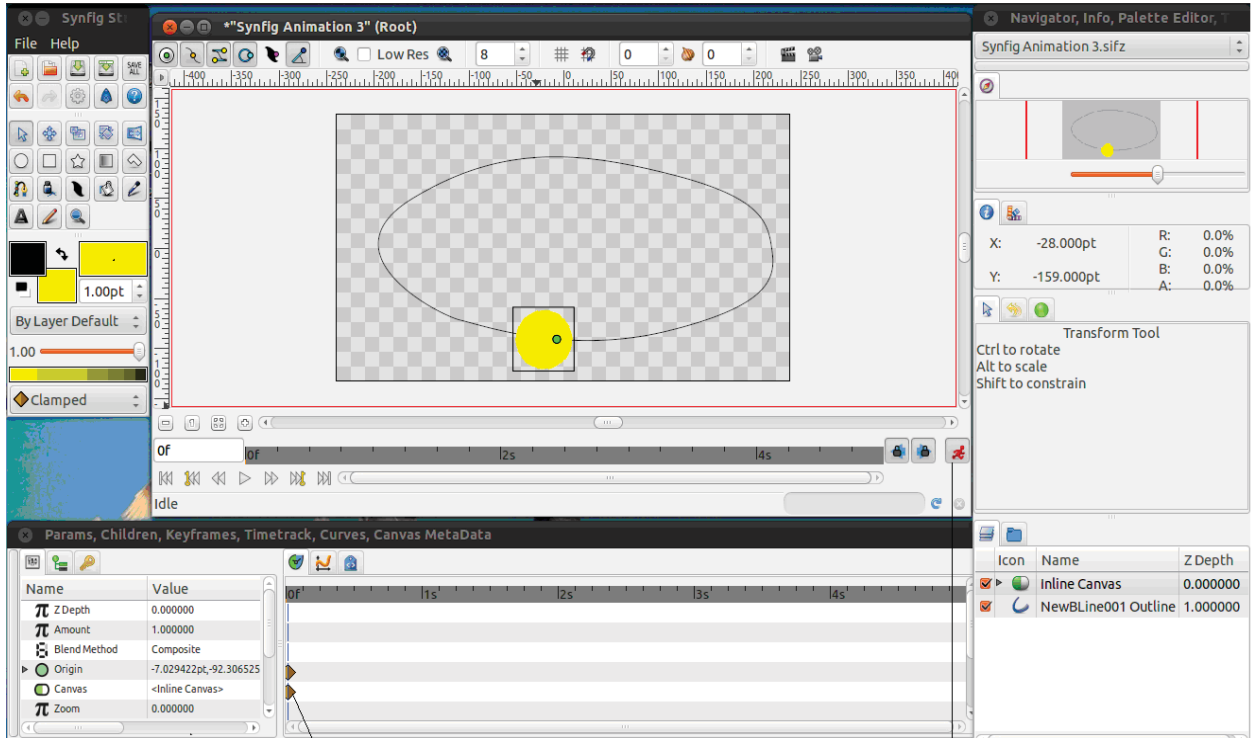
આકૃતિ 4.16 : Link to Bline વિકલ્પ પસંદ કર્યા બાદ

- Inline Canvas સ્તરને બંધ કરવા તેની નજીક દેખાતા નાના ત્રિકોણ પર ક્લિક કરો. આકૃતિ 4.17માં દર્શાવ્યા પ્રમાણે Inline Canvas પસંદ કરો.



આકૃતિ 4.17 Inline canvas સ્તરની પસંદગી

- 'એનિમેશન એડિટિંગ મોડ' ચાલુ કરો.
- આકૃતિ 4.18માં તમે કેનવાસની ફરતે લાલ રંગની લીટી જોઈ શકો છો, જે એમ દર્શાવે છે કે, 'એનિમેશન એડિટિંગ મોડ' ચાલુ થઈ ગયો છે. આકૃતિ 4.18માં દર્શાવ્યા મુજબ શરૂઆતનું સ્થાન નોંધવા ટાઈમલાઈનમાં 0f ઉપર 'ગ્રીન ડક'નો ઉપયોગ કરી વર્તુળને થોડુંક ઘસડો. 0f પર એક 'વે-પોઈન્ટ' ઉમેરાયો હશે, જે તમે જોઈ શકશો.



Waypoint

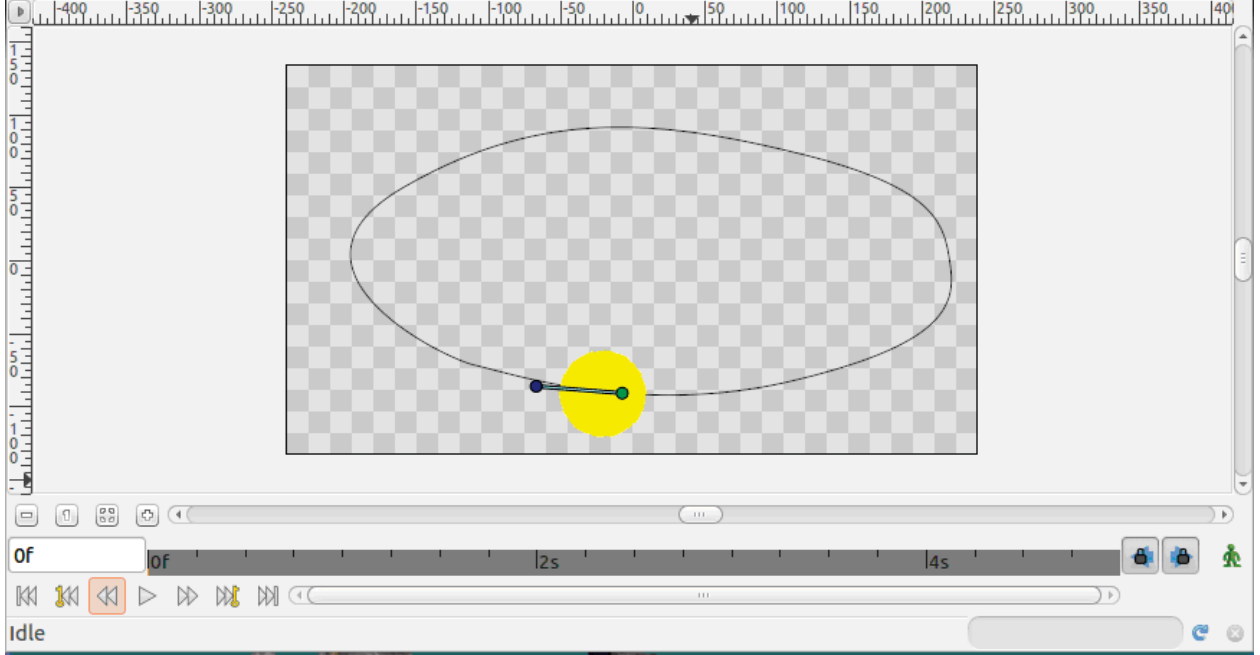
animate edit mode on

આકૃતિ 4.18 : ટાઈમલાઈનમાં 0s સ્થાન પર નોંધણી

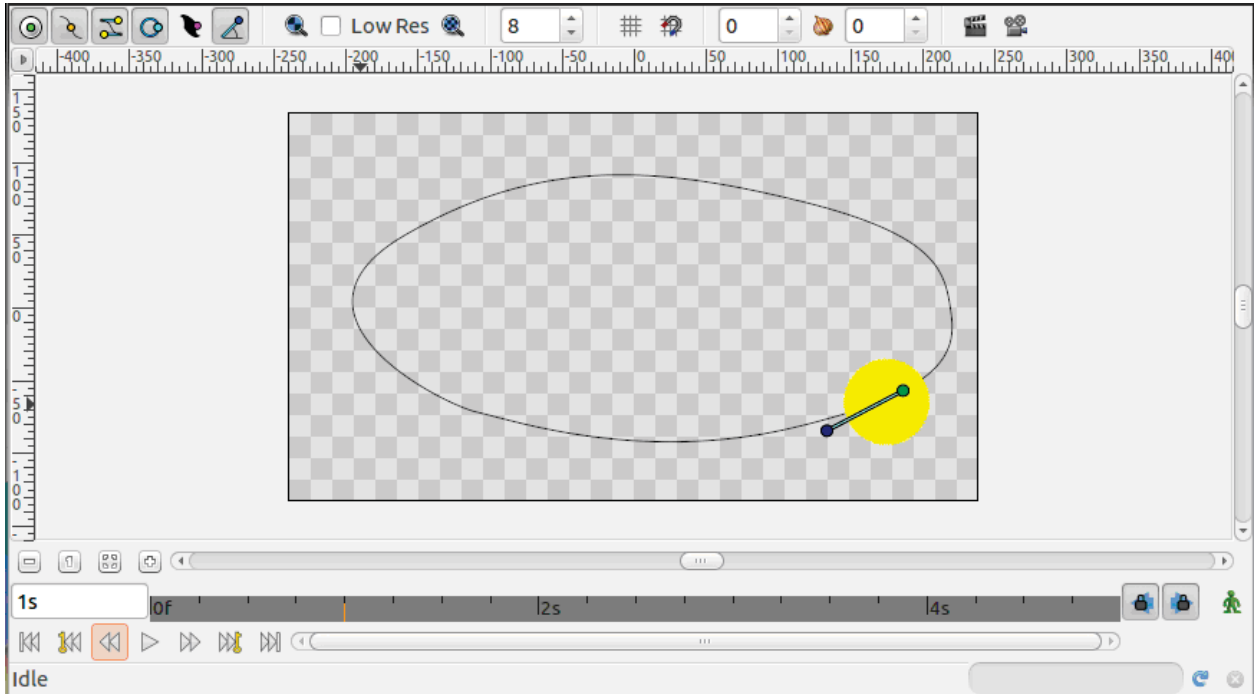
- હવે ટાઈમલાઈનમાં 5s સ્થાન પર ક્લિક કરો, અને ગ્રીન ડકનો ઉપયોગ કરીને તમે blineના અંતસ્થાન પર પહોંચો, ત્યાં સુધી વર્તુળને ઘડિયાળની ઊંધી દિશામાં ઘસડો. આકૃતિ 4.19 4.19 (a), (b) (c), (d),

(e) અને (f) વર્તુળ શરૂઆતના સ્થાન પરથી અંત સ્થાન સુધી ઘસડાયાનું દર્શાવે છે. દરેક છબીમાંની ટાઇમલાઇન તે ચોક્કસ સમયે વર્તુળનું સ્થાન દર્શાવે છે.

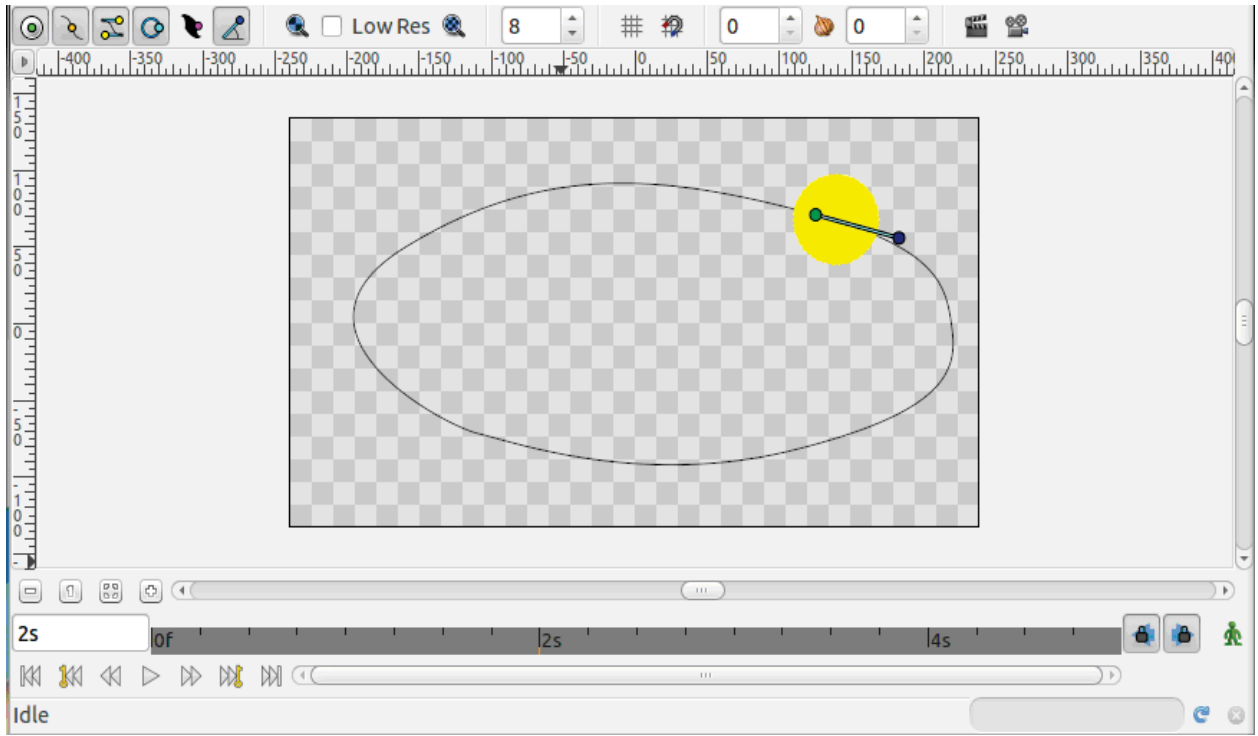
નોંધ : વર્તુળને ઘડિયાળની દિશામાં ફેરવીને સીધું જ અંત સ્થાન પર ન ઘસડો. આમ કરવાથી જુદાં-જુદાં સમય-સ્થાન પર વર્તુળનું સ્થાન યોગ્ય રીતે નોંધી શકાશે નહીં.



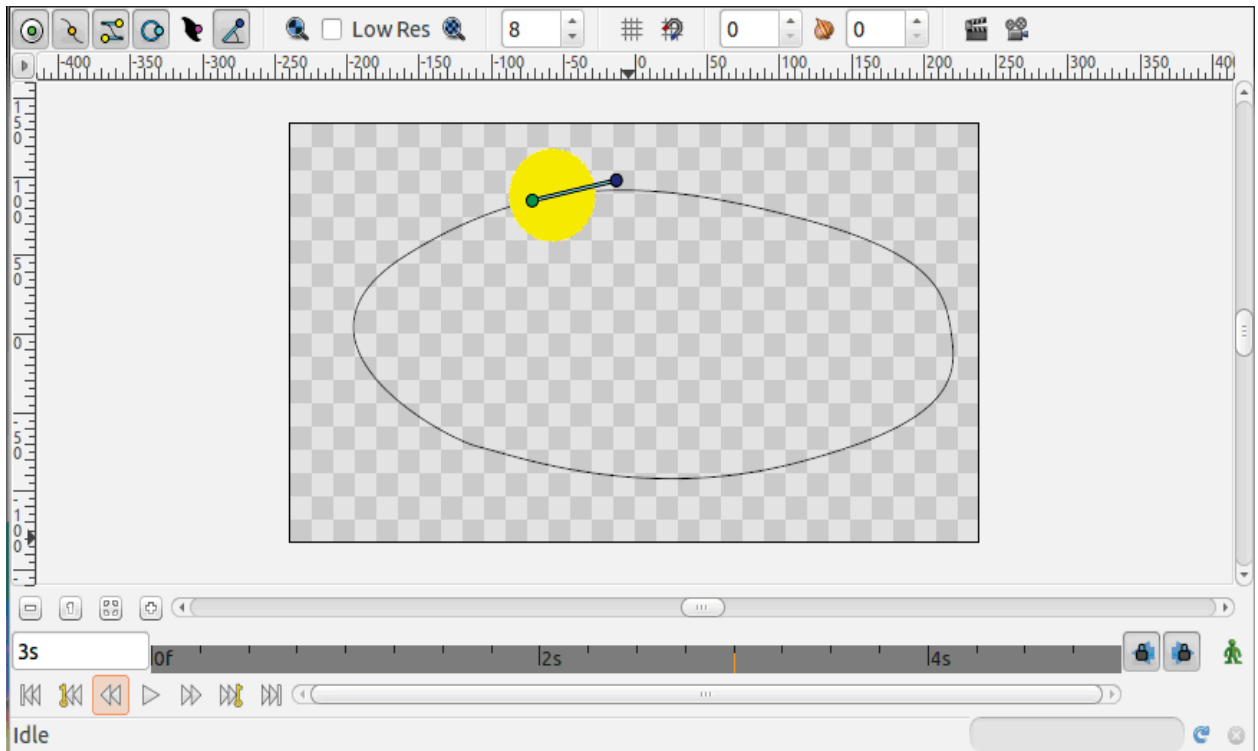
આકૃતિ 4.19(a) : 0f પર વર્તુળનું સ્થાન



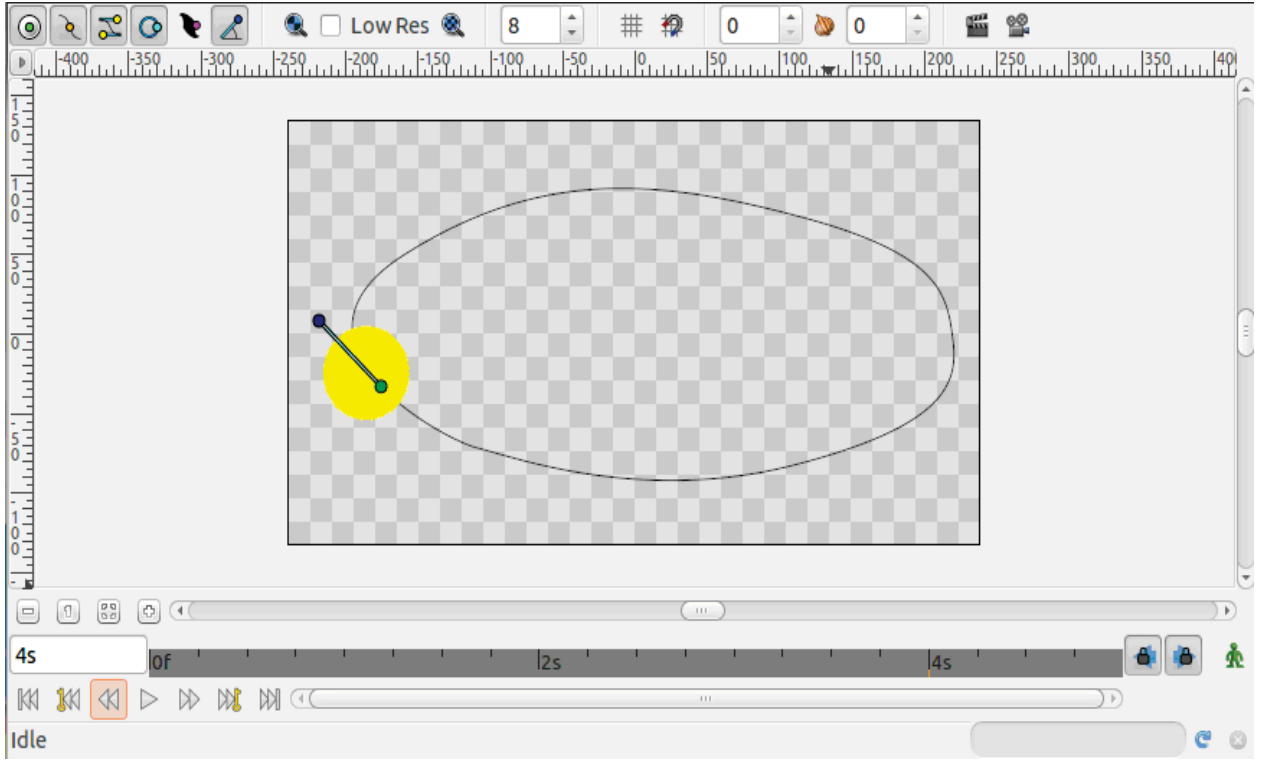
આકૃતિ 4.19 (b) : 1s પર વર્તુળનું સ્થાન



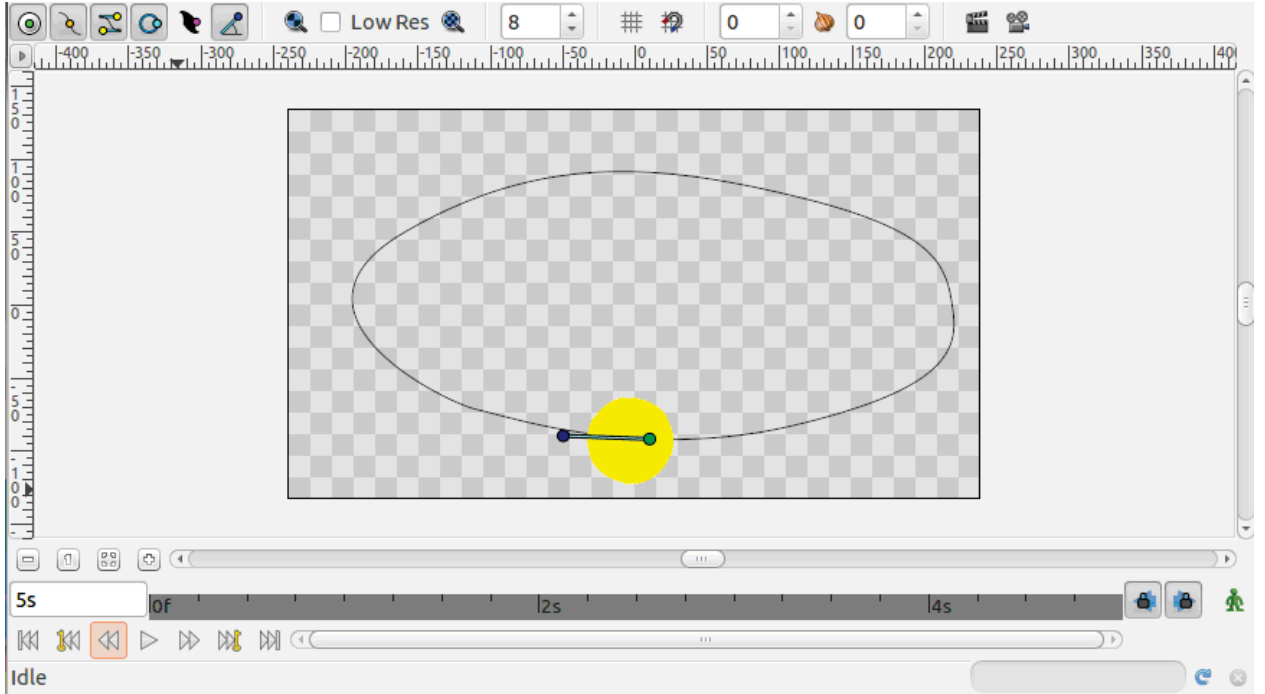
આકૃતિ 4.19(c) : 2s પર વર્તુળનું સ્થાન



આકૃતિ 4.19 (d) : 3s પર વર્તુળનું સ્થાન

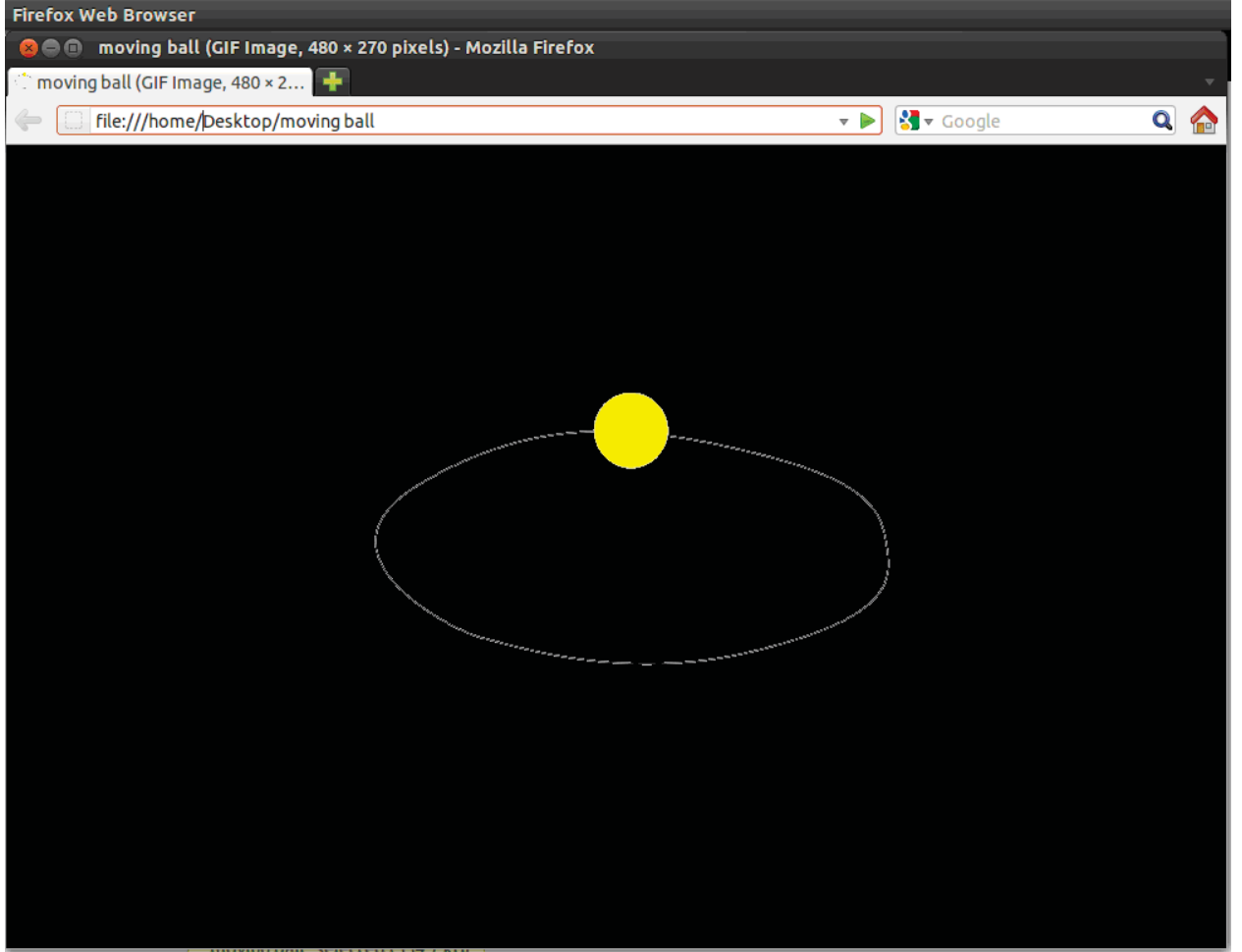


આકૃતિ 4.19(e) : 4s પર વર્તુળનું સ્થાન



આકૃતિ 4.19(f) : 5s પર વર્તુળનું સ્થાન

- એનિમેશનનું પૂર્વાવલોકન કરવા Play પર ક્લિક કરો. તમે વર્તુળને ઘડિયાળની ઊંધી દિશામાં ફરતું જોઈ શકશો.
- ફાઇલને સાચવીને પછી રેન્ડરિંગ (save and render) કરો. આકૃતિ 4.20 વેબબ્રાઉઝરનો ઉપયોગ કરીને જોઈ શકાતું પરિણામ દર્શાવે છે.



આકૃતિ 4.20 : વેબ બ્રાઉઝરમાં પરિણામ

સારાંશ

આ પ્રકરણમાં આપણે સ્તરના ખ્યાલનો અને એક ઓબ્જેક્ટની ઉપર કે નીચે આપણે બીજો ઓબ્જેક્ટ કેવી રીતે મૂકી શકીએ, તેનો પરિચય મેળવ્યો. સ્તર આપણને દરેક ઓબ્જેક્ટ સાથે સ્વતંત્ર રીતે કામ કરવાની છૂટ આપે છે. એનકેપ્સ્યુલેશનના ખ્યાલનો પરિચય પણ મેળવ્યો. એનકેપ્સ્યુલેશનનો ઉપયોગ કરીને એનિમેશનમાં બાકીના ઓબ્જેક્ટમાં ફેરફાર કર્યા સિવાય અમુક ચોક્કસ ઓબ્જેક્ટને અસર આપી શકાય છે. આપણે એ પણ ચર્ચા કરી કે, ઉપયોગકર્તા દ્વારા નિર્ધારિત પથ પર ઓબ્જેક્ટને કેવી રીતે એનિમેટ કરી શકાય.

સ્વાધ્યાય

1. સ્તર (Layers) વર્ણવો. ઉદાહરણ આપો.
2. એનકેપ્સ્યુલેશન એટલે શું ? આપણને શા માટે એની જરૂર પડે છે ?
3. એક શેરીના દૃશ્યની કલ્પના કરો, જેમાં મકાન, કાર, સૂર્ય, ટ્રાફિકલાઈટ અને વ્યક્તિઓ હોય. કયું સ્તર ઉપર હશે અને કયું સ્તર નીચે હશે, તે જણાવો.
4. Link to Bline વિકલ્પનો ઉદ્દેશ શું છે? તે જણાવો.

5. હેતુલક્ષી પ્રશ્નો

- (1) નીચેનાં પૈકી કયો સ્તરનો પ્રકાર છે ?
- (a) ટાઈમબાર (Timebar) (b) ગ્રેડિયન્ટ (Gradient)
(c) પેલેટ (Palette) (d) એનિમેટ એડિટીંગ મોડ (Animate editing mode)
- (2) નીચેના પૈકી કોને તેમનું પોતાનું લેયર હોય છે ?
- (a) ઓબ્જેક્ટ, એલિમેન્ટ (તત્ત્વ), ઈફેક્ટ (અસર) (b) પેરામીટર, ઓબ્જેક્ટ, પેનલ
(c) ટાઈમ, પેરામીટર, લેયર (d) અસર, ઓબ્જેક્ટ, પેનલ
- (3) નીચેના પૈકી કઈ પેનલ સ્તરનાં પરિબળો દર્શાવે છે ?
- (a) સ્તર (Layer) (b) કી-ફ્રેમ (Key Frame)
(c) પેરામ્સ (Params) (d) ટાઈમટ્રેક (Timetrack)
- (4) નીચેના પૈકી કઈ બ્લેન્ડ (blend) પદ્ધતિની સામાન્ય ગોઠવણી છે ?
- (a) કમ્પોઝીટ (Composite) (b) આલ્ફા ઓવર (Alpha over)
(c) સ્ટ્રેઈટ (Straight) (d) સ્ટ્રેઈટ ઓનટુ (Straight onto)
- (5) નીચેના પૈકી કઈ લાક્ષણિકતાનો ઉપયોગ કરીને આપણે કોઈ એક સ્તરની નીચેના સ્તરને અસર પહોંચાડવા વગર માત્ર એક જ સ્તરને અસર આપી શકીએ ?
- (a) એનકેપ્સ્યુલેશન (Encapsulation) (b) ગ્રુપ્સ (Groups)
(c) તત્ત્વો (Elements) (d) પ્રાયલ (Parameters)
- (6) કેનવાસનાં અલગ-અલગ દરેક તત્ત્વને નીચેના પૈકી કયા વિભાગમાં વિભાજિત કરવામાં આવે છે ?
- (a) ફ્રેમ (Frames) (b) સ્તર (Layers)
(c) કી-ફ્રેમ (Keyframes) (d) પેનલ (Panels)
- (7) ઓબ્જેક્ટના શિરોબિંદુ (vertex)ને Blineના શિરોબિંદુ સાથે જોડવા માટે નીચે પૈકી કઈ લાક્ષણિકતાનો ઉપયોગ કરાય છે ?
- (a) લિન્ક ટુ ઓબ્જેક્ટ (Link to object) (b) લિન્ક ટુ લાઈન (Link to line)
(c) લિન્ક ટુ bline (Link to bline) (d) લિન્ક ટુ વર્ટેક્સ (Link to vertex)
- (8) નીચે પૈકી કયું એનકેપ્સ્યુલેશન ક્રિયા પછી બનાવવામાં આવતા નવા સ્તરને રજૂ કરે છે ?
- (a) ઈનલાઈન લેયર (Inline layer) (b) ઈન લાઈન ફ્રેમ (Inline frame)
(c) ઈનલાઈન કી-ફ્રેમ (Inline keyframe) (d) ઈનલાઈન કેનવાસ (Inline canvas)
- (9) દરેક સ્તરના નીચેના પૈકી કયા પોતાના ભાગ હોય છે ?
- (a) ફ્રેમ (Frame) (b) કી-ફ્રેમ (Keyframe)
(c) પેરામીટર (Parameter) (d) પેનલ (Panel)
- (10) નીચેનાં પૈકી કયું પદ જુદી-જુદી વસ્તુઓને એક જૂથમાં ભેગી કરવા માટે છે ?
- (a) ક્લસ્ટર (Cluster) (b) ઈનલાઈન (Inline)
(c) પેરામીટર (Parameter) (d) એનકેપ્સ્યુલેશન (Encapsulation)

(11) નીચેનાં પૈકી કયું પદ જુદાં-જુદાં સ્તર રજૂ કરે છે, જેના પર આપણે એકબીજાની ઉપર ઓબ્જેક્ટ મૂકીએ છીએ ?

- (a) સ્તર (Layer) (b) ફ્રેમ (Frame)
(c) પેરામીટર (Parameter) (d) પેનલ (Panel)

(12) ગ્રેડિયન્ટ (Gradient), બ્લર (blur), ડિસ્ટોર્સન (distortion) અને ફિલ્ટર (filter) કયા ભાગના પ્રકારો છે ?

- (a) ફ્રેમ (Frame) (b) લેયર (Layer)
(c) કી-ફ્રેમ (Keyframe) (d) પેનલ (Panel)

પ્રાયોગિક સ્વાધ્યાય

1. એકબીજાની ઉપર આવતાં (overlapping) હોય તેવાં લંબચોરસ-વર્તુળ અને ચોરસ દોરો. કેનવાસને પાછળનો (બેકગ્રાઉન્ડ) રંગ પણ કરો.
2. તારો દર્શાવતું Star Layer અને વર્તુળ દર્શાવતું Circle Layer બનાવો. Star Layer પર ગ્રેડિયન્ટ અસર આપો. ચોરસ દર્શાવતું Square Layer બનાવીને તેના પર આછી દેખાય તેવી (બ્લર) અસર આપો. હવે લંબચોરસ માટે Rectangle બનાવીને તેને પણ બ્લર (blur) અસર આપો. (નોંધ : ઓબ્જેક્ટ એકબીજાની ઉપર (overlap) થવા જોઈએ.)
3. કેનવાસ પર ગ્રેડિયન્ટ (Gradient) અસરનો ઉપયોગ કરીને આકાશ અને મેદાન જેવું ચિત્ર દેખાય તેવું બનાવો. બન્ને માટે જુદા રંગનાં ગ્રેડિયન્ટનો ઉપયોગ કરો. ગ્રેડિયન્ટ તરીકે આકાશ માટે નીલા રંગનો અને મેદાન માટે લીલા રંગનો ઉપયોગ કરો.
4. એકબીજાની ઉપર આવતાં હોય તેવાં બે વર્તુળ દોરીને તેમાંથી માત્ર એક જ વર્તુળ પર બ્લર (blur) અસર આપો.
5. ઉપયોગકર્તા દ્વારા નિર્ધારિત પથ પર ચાલતો હોય તેવા તારા (star)નું એનિમેશન તૈયાર કરો.
6. એક તારો (star) ડાબી બાજુથી જમણી બાજુ સરકતો હોય, ત્યારે તે ફરતો (rotating) હોય તેવું એનિમેશન તૈયાર કરો.
7. આકાશમાં પતંગ ઊડતો હોય તેવું એનિમેશન તૈયાર કરો.



5

સીન્ડિંગમાં ઇમેજનો ઉપયોગ

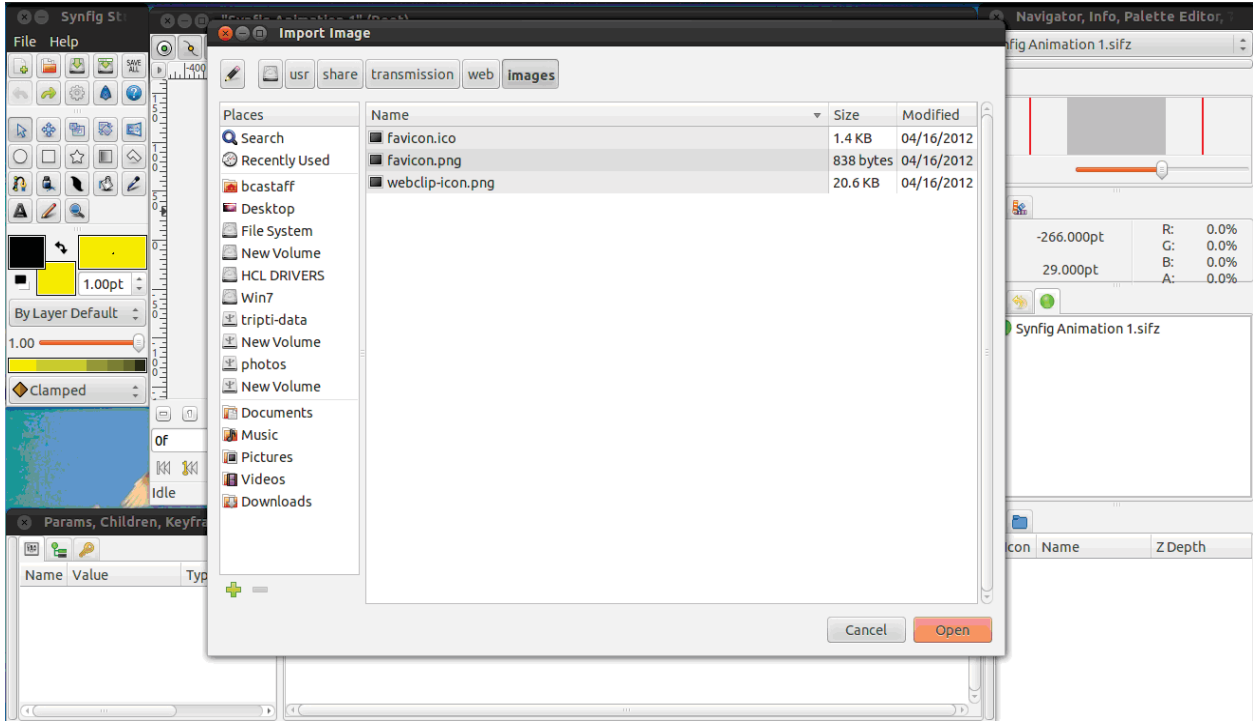


જ્યારે કોઈ રજૂઆત (પ્રેઝન્ટેશન) કે એનિમેશનમાં ચિત્રોનો ઉપયોગ કરવામાં આવે છે, ત્યારે તે રજૂઆતને ગ્રહણ કરવાની ગુણવત્તામાં ઘણો સુધારો થાય છે. આપણા એનિમેશનને વિશિષ્ટ દેખાવ આપવા માટે આપણે સીન્ડિંગમાં ચિત્રો (ઇમેજિસ) ઉમેરી શકીએ છીએ. ટૂલ્સનો ઉપયોગ કરીને આપણે ઓબ્જેક્ટ્સ સાથે જે રીતે કામ કરીએ છીએ, એ જ રીતે આ ચિત્રો/ઇમેજિસ સાથે કામ કરી શકીએ છીએ અને આપણી જરૂરિયાત પ્રમાણે તેને ગોઠવી શકીએ છીએ. આપણે આ પ્રકરણમાં ઇમેજ સાથે કેવી રીતે કામ કરવું તે બાબત જોઈશું.

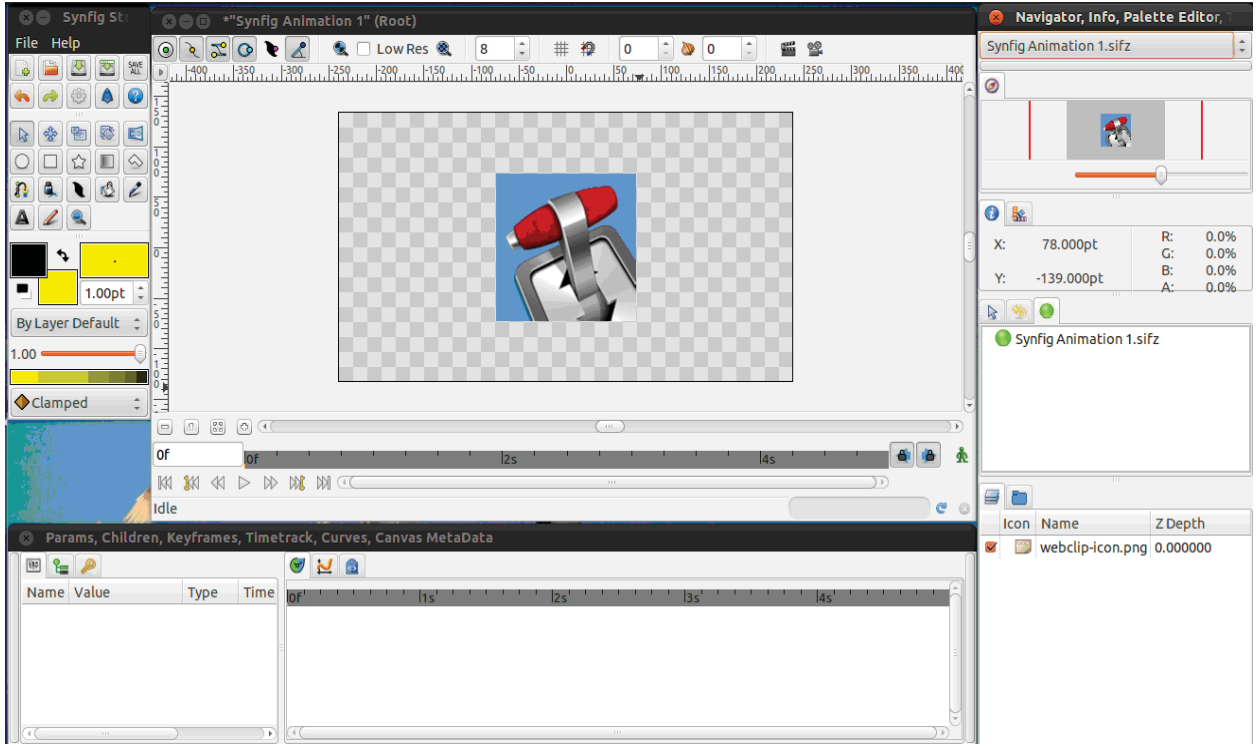
ઇમેજ ઉમેરવી (Inserting Image)

ચાલો, આપણે એક લેયર (Layer) બનાવવાનો પ્રયત્ન કરીએ કે જે આ ઇમેજને ધરાવશે. ઇમેજને કેનવાસ ઉપર આયાત કરવા માટે નીચે જણાવેલાં પગલાંઓ પ્રમાણે કાર્ય કરો :

- એક નવી ફાઇલ બનાવો.
- પસંદ કરો File → Import. આના વિકલ્પ રૂપે, તમે CTRL + i પણ દબાવી શકો. આ એકીકરણ CTRL + i ઇમેજને આયાત કરવા માટે શોર્ટકટ કી છે. આથી આકૃતિ 5.1માં દર્શાવ્યા પ્રમાણે તમે જેમાંથી ઇમેજ આયાત કરવા ઇચ્છો છો તે ફોલ્ડર્સ બતાવતું એક ડાયલોગબોક્સ ખૂલશે. હવે તમે જે ઇમેજ આયાત કરવા માંગો છો તે પસંદ કરો અને Open કી દબાવો. આથી કેનવાસ ઉપર ઇમેજ આયાત થશે. આકૃતિ 5.2માં આપણે કેનવાસ ઉપર આયાત કરેલી ઇમેજને દર્શાવેલ છે. અહીં એ નોંધ કરશો કે આપણે જે ઇમેજ આયાત કરવા પસંદ કરીએ છીએ, તેના ઉપર આપણા સ્કીનના દેખાવનો આધાર છે.

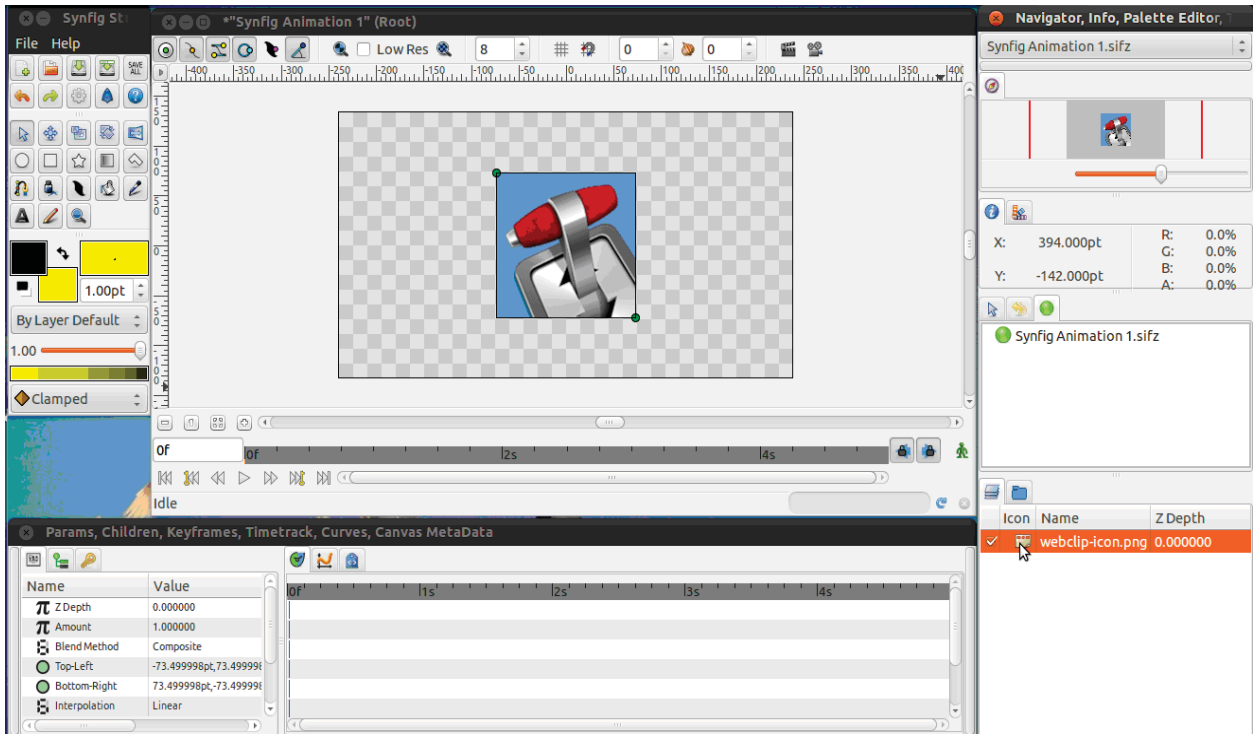


આકૃતિ 5.1 : 'ઇમ્પોર્ટ ઇમેજ' ડાયલોગબોક્સ



આકૃતિ 5.2 : કેનવાસ ઉપર રાખવામાં આવેલી ઈમેજ

- હવે આયાત કરેલી ઈમેજનું કદ બદલવા માટે લેયર્સ પેનલમાંથી ઈમેજ લેયર પસંદ કરો. આકૃતિ 5.3માં દર્શાવ્યા પ્રમાણે ઈમેજ ઉપર તમે બે લીલાં બિંદુઓ જોઈ શકશો. ઈમેજનું કદ બદલવા માટે તમે હવે આ બિંદુઓનો ઉપયોગ કરો છો. આકૃતિ 5.4માં દર્શાવ્યા પ્રમાણે જ્યારે લીલાં બિંદુઓને ખસેડવામાં આવે છે, ત્યારે ઈમેજનું બદલાયેલું કદ જોવા મળે છે.



આકૃતિ 5.3 : ઈમેજનું કદ બદલવા માટેનાં લીલાં બિંદુઓ



આકૃતિ 5.4 : લીલા બિંદુઓનો ઉપયોગ કરીને ઇમેજનું બદલાયેલું કદ

- અહીં તમે એ અવલોકન કરશો કે જ્યારે ઇમેજનું કદ બદલવામાં આવે છે, ત્યારે તે ઇમેજને બેરોળ બનાવે છે. આ બનવાનું કારણ એ છે કે આપણે સાપેક્ષ ગુણોત્તર (aspect ratio) જાળવી રાખ્યો નથી. જો આપણે ઇમેજનું કદ બદલવા ઇચ્છતા હોય અને તેનો સાપેક્ષ ગુણોત્તર જાળવી રાખતા માગતા હોઈએ, તો લેયરને પ્રાવૃત્ત (encapsulate) કરવું આવશ્યક બની જાય છે.
- હવે ઇમેજ લેયર ઉપર રાઈટ ક્લિક કરો અને તે પછી encapsulate પસંદ કરો. આથી તે ઇનલાઈન કેનવાસ લેયર ઉમેરશે. નાના ત્રિકોણનો ઉપયોગ કરીને પ્રાવૃત્ત લેયરને ખોલો. આથી આકૃતિ 5.5માં દર્શાવ્યા પ્રમાણે ઇનલાઈન કેનવાસમાં તમે ઇમેજ લેયર જોઈ શકશો.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	webclip-icon.png	0.000000

આકૃતિ 5.5 : પ્રાવૃત્ત ઇમેજ લેયર

- હવે આપણે ઇમેજ લેયર ઉપર નવું સ્કેલ લેયર ઉમેરવું પડશે. આ માટે નીચે પ્રમાણે કમાન્ડ આપો : Image layer → Right click → New Layer → Transform → Scale. આથી આકૃતિ 5.6માં દર્શાવ્યા પ્રમાણે તમે ઇમેજ લેયર ઉપર સ્કેલ લેયરને ઉમેરાયેલું જોઈ શકશો.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	Scale	0.000000
<input checked="" type="checkbox"/>	webclip-icon.png	1.000000

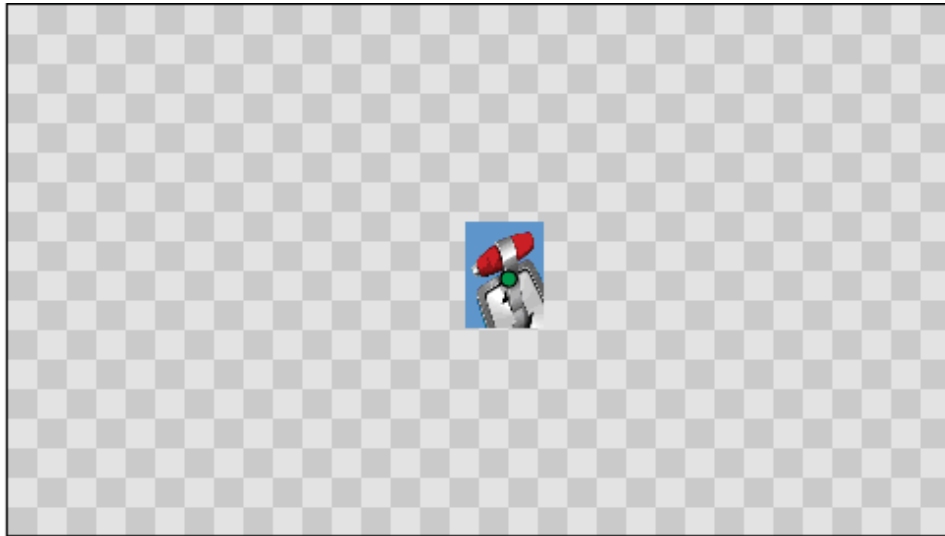
આકૃતિ 5.6 : ઈમેજ લેયર ઉપર ઉમેરાયેલું સ્કેલ લેયર

- હવે સ્કેલ લેયર પસંદ કરો અને આકૃતિ 5.7માં દર્શાવ્યા પ્રમાણે પ્રાયલોની પેનલમાં amountને 0 માંથી -1માં બદલો, આથી આકૃતિ 5.8માં દર્શાવ્યા પ્રમાણે તમે કેનવાસમાં ઈમેજનું કદ બદલાતું જોઈ શકશો.

Name	Value
π Amount	0.000000
Origin	15.083985pt,-4.058743pt

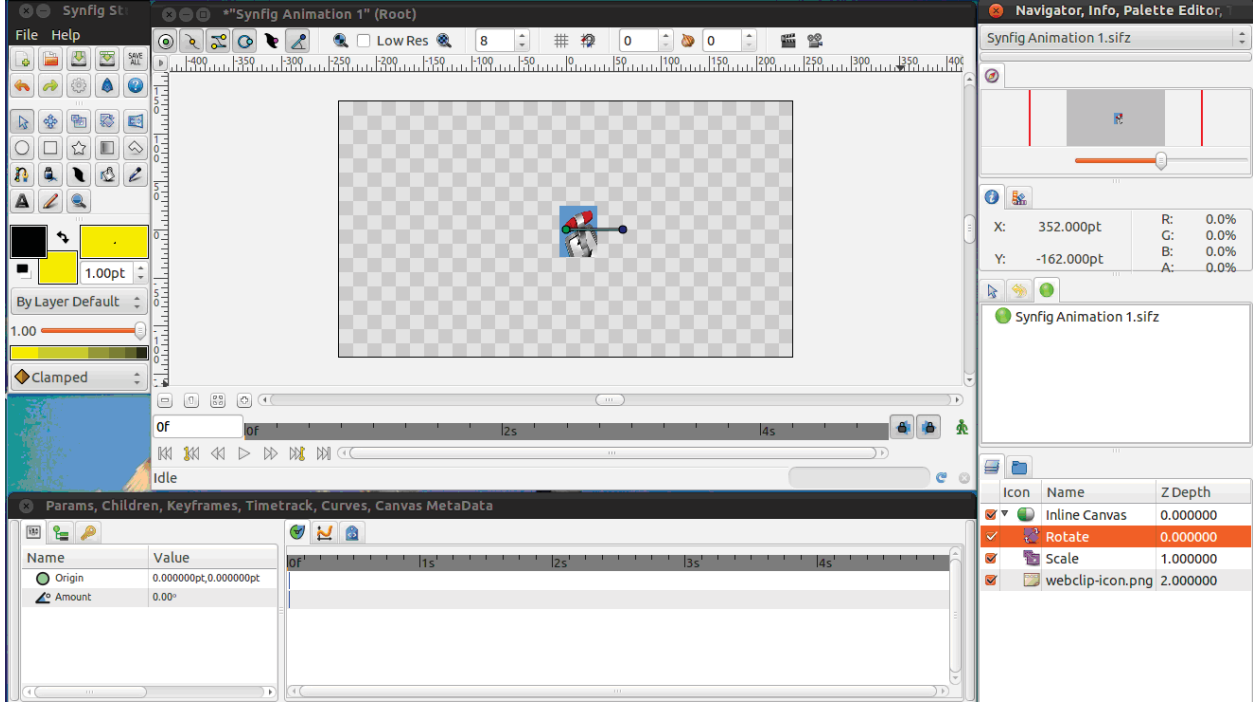
Name	Value
π Amount	-1.0000000000000000
Origin	15.083985pt,-4.058743pt

આકૃતિ 5.7 : સ્કેલ લેયરમાં Amountને 0માંથી -1 સેટ કરો



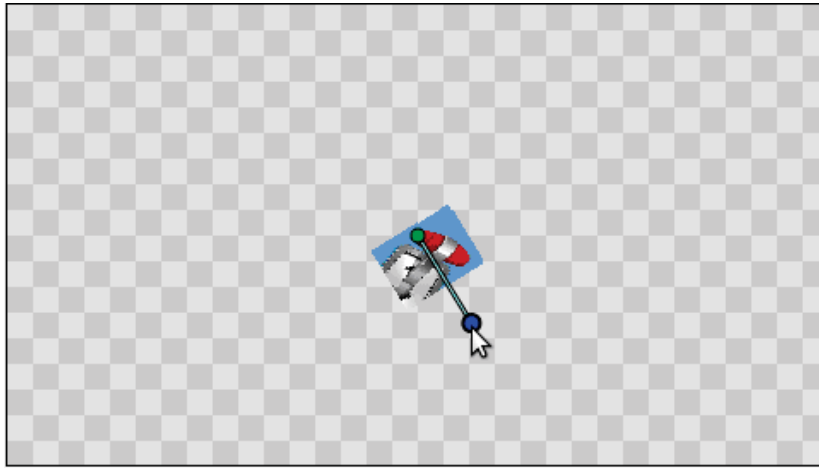
આકૃતિ 5.8 : સ્કેલ થયેલી ઈમેજ

- હવે જો તમે ઇમેજને ફેરવવા ઇચ્છતા હો, તો સ્કેલ લેયર ઉપર એક નવું રોટેટ લેયર ઉમેરો. આ માટે નીચે પ્રમાણે કમાન્ડ આપો : Scale Layer → Right Click → New Layer → Transform → Rotate. આથી આકૃતિ 5.9માં દર્શાવ્યા પ્રમાણે તમે સ્કેલ લેયર અને ઇમેજ લેયરની ઉપર રોટેટ લેયર ઉમેરાયેલું જોઈ શકશો.



આકૃતિ 5.9 : સ્કેલ લેયર ઉપર ઉમેરાયેલું રોટેટ લેયર

- રોટેશન ડક (rotation duck) (કે જે આકૃતિમાં વાદળી રંગનું ડક છે)નો ઉપયોગ કરીને તમે ઇમેજને ઇચ્છો તે પ્રમાણે ફેરવી શકો છો. આકૃતિ 5.10માં ફેરવેલી ઇમેજ દર્શાવેલી છે.



આકૃતિ 5.10 : ડકનો ઉપયોગ કરીને ફેરવેલી ઇમેજ

અહીં અવલોકન કરો કે આપણે એક વાર ઇમેજ સામેલ કરીએ તે પછી તેને આપણે પ્રમાણસર કરી શકીએ, ફેરવી શકીએ અને તેના ઉપર વિવિધ પ્રકારની કામગીરી કરી શકીએ છીએ. એનિમેશનમાં ઇમેજનો ઉપયોગ તેની ઉપયોગિતામાં હંમેશાં વધારો કરે છે.

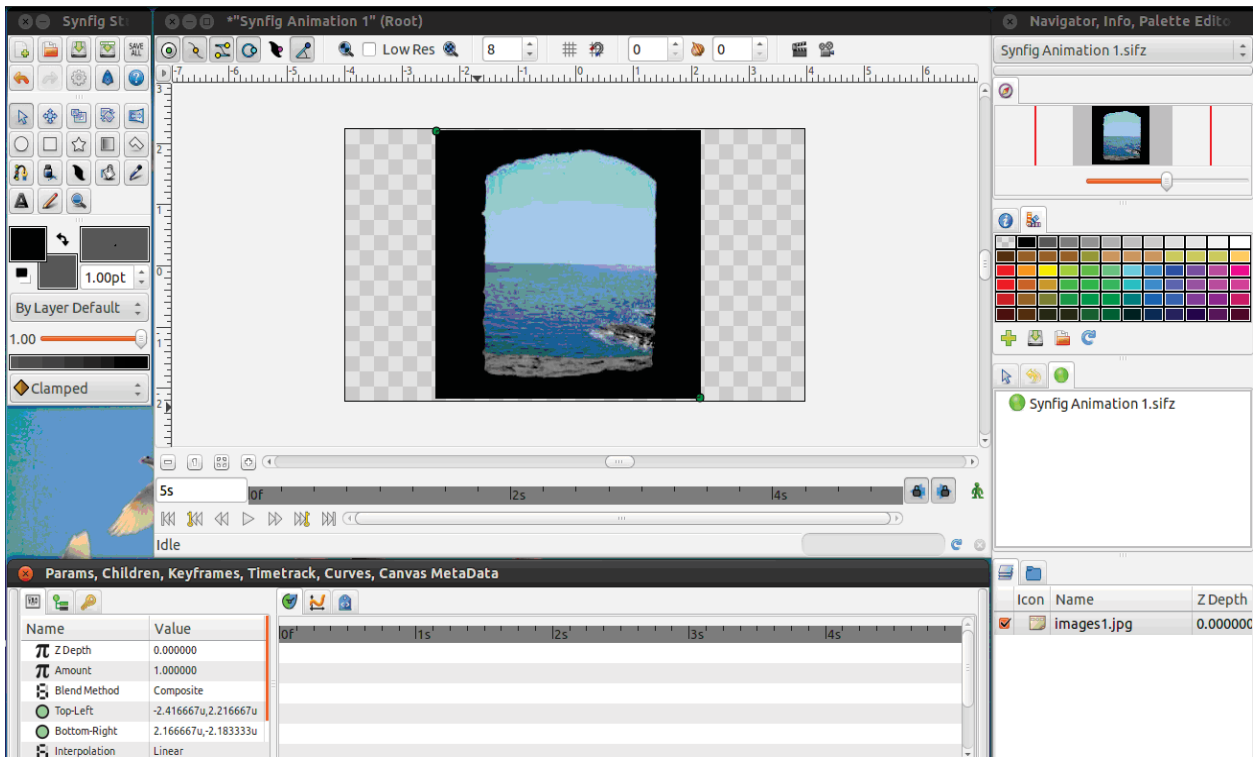
માસ્કિંગ (Masking)

ધારો કે તમે એનિમેશનમાં કોઈ વસ્તુ/ઓબ્જેક્ટનું હલનચલન બતાવવા ઈચ્છો છો, જેમકે બારીમાંથી દડો બહાર આવતો હોય. જો તમે સીન્કિંગમાં બારીનું ચિત્ર બનાવ્યું હોય, તો ઓબ્જેક્ટ (અહીં દડો)ને બારીની પાછળ રાખી શકો અને તે ઓબ્જેક્ટને ખસેડી પણ શકો. પણ બારી જો કોઈ ચિત્ર હોય, તો દડાને એનિમેટ કરવાનું કાર્ય મુશ્કેલ છે.

માસ્કિંગ એ એક એવી લાક્ષણિકતા છે જેના દ્વારા લેયરના કોઈ ભાગને છુપાવી શકાય તેમજ પુનઃ દર્શાવી પણ શકાય છે. મૂળભૂત રીતે જે આકાર માસ્ક તરીકે વાપરીએ છીએ, તે એક બારીનું કામ આપે છે, જેના વડે તેની નીચેના ઓબ્જેક્ટને તમે જોઈ શકો છો. વૈકલ્પિક રીતે, તેની નીચેના ઓબ્જેક્ટને છુપાવવા માટે થીગડા તરીકે ઉપયોગ કરી શકાય છે.

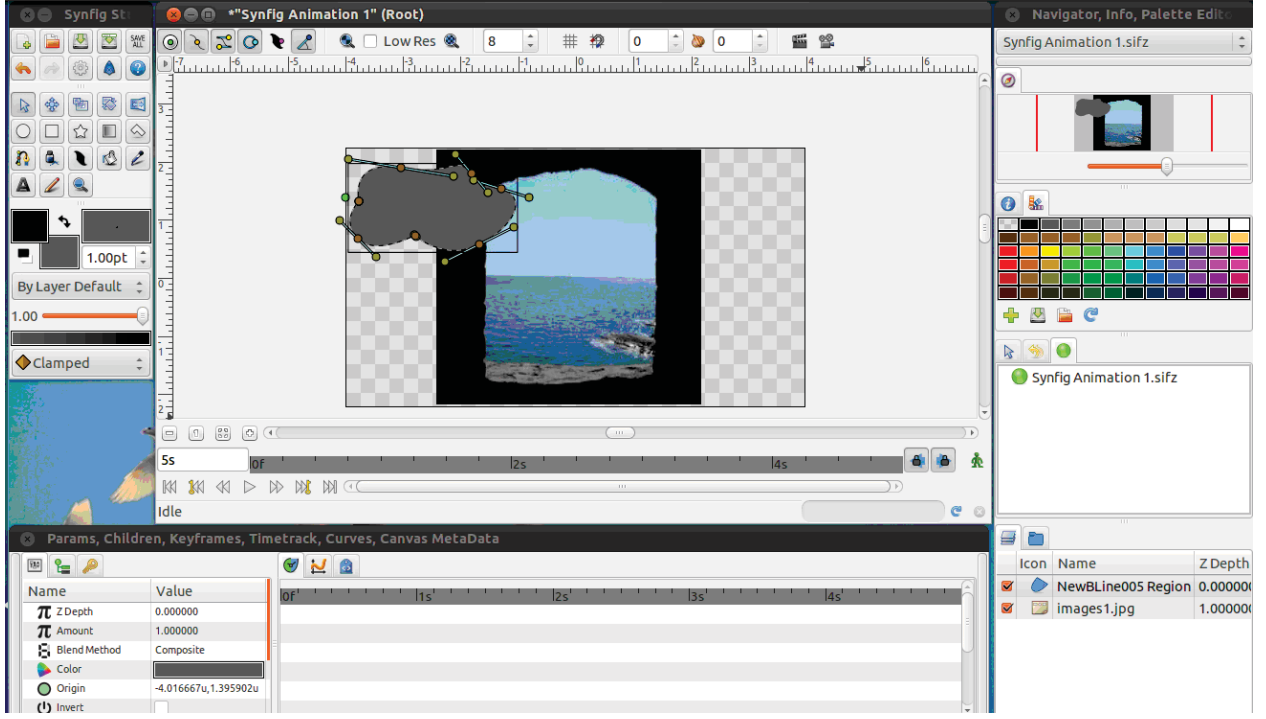
ચાલો, હવે આપણે તે ઉદાહરણ દ્વારા સમજાવે અને માસ્કિંગના ખ્યાલનો ઉપયોગ કરીને બારીમાંથી દડાનું હલનચલન જોવાનો પ્રયત્ન કરીએ.

- એક નવી ફાઇલ બનાવો.
- હવે CTRL + i દ્વારા કેનવાસ ઉપર ઈમેજ આયાત કરો. તેમાંથી ઈમેજ પસંદ કરો. આપણી જરૂરિયાત પ્રમાણે તેનું કદ બદલો. આકૃતિ 5.11માં આયાત કરેલી ઈમેજ દર્શાવેલી છે. અહીં એ યાદ રાખો કે તમારો સ્ક્રીન તમે પસંદ કરેલી ઈમેજ પ્રમાણે જુદો દેખાઈ શકે છે. તમે ઈમેજ લેયરને લેયર્સ પેનલમાં ઉમેરાયેલી જોઈ શકો છો.



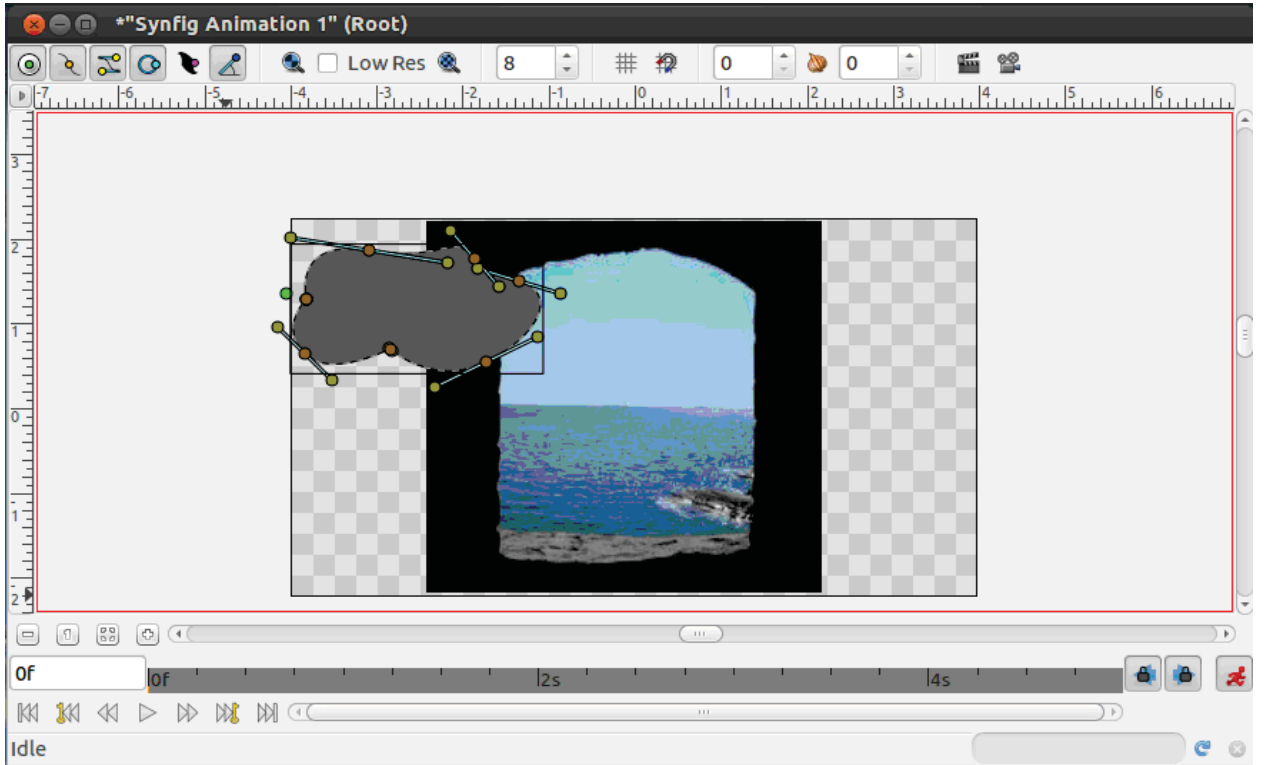
આકૃતિ 5.11 : કેનવાસ ઉપર આયાત કરેલી ઈમેજ

- હવે આકૃતિ 5.12માં દર્શાવ્યા પ્રમાણે બીલાઈન ટૂલ (Bline tool) ઓબ્જેક્ટનો ઉપયોગ કરીને કેનવાસની ડાબી બાજુએ એક વાદળ (cloud) દોરો. તમે ઈમેજ લેયરની ઉપર બીલાઈન લેયર જોઈ શકશો.

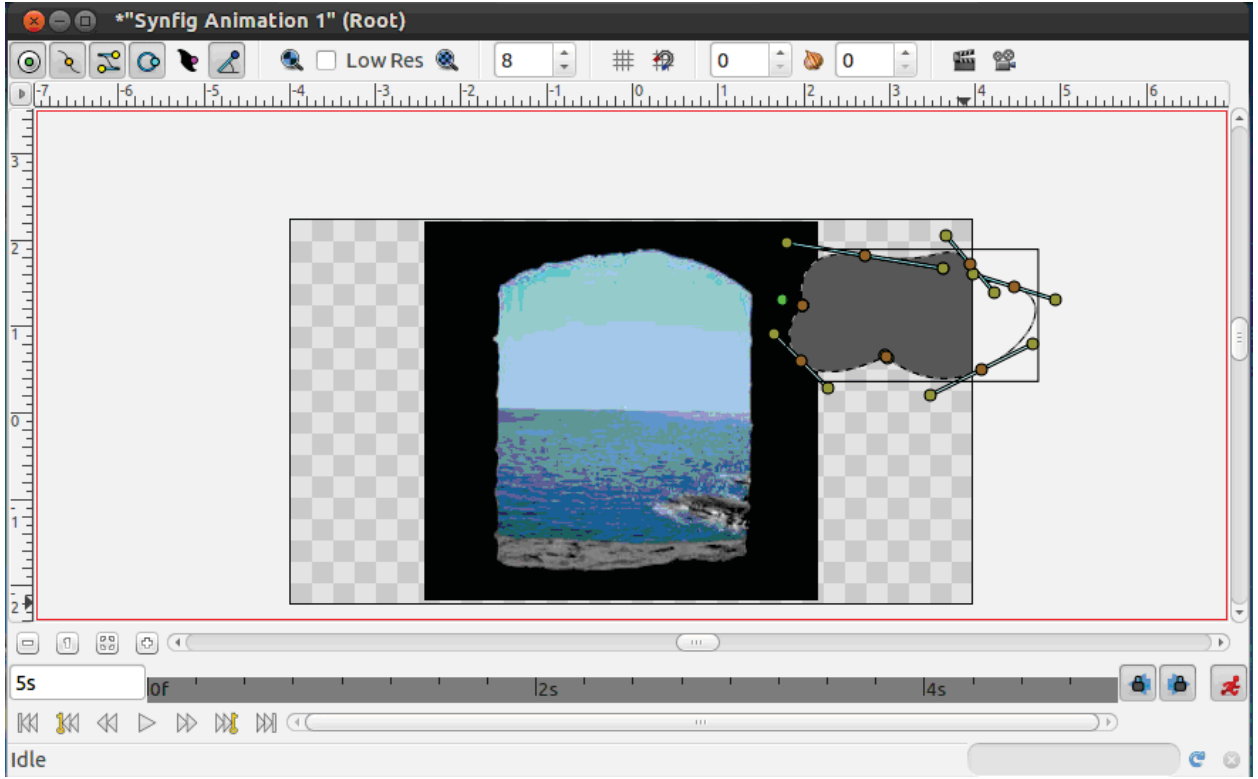


આકૃતિ 5.12 : બીલાઈન ટૂલ વડે દોરેલું વાદળ

- હવે animate edit વિકલ્પ ચાલુ કરો.
- હવે આપણે કેનવાસ ઉપર વાદળને ડાબી બાજુથી જમણી બાજુ તરફ જતું એક નાનું એનિમેશન બનાવીશું. ટાઈમલાઈનમાં 0f જગ્યા ઉપર રેકોર્ડ કરવા માટે વાદળને થોડું ખસેડો. હવે ટાઈમલાઈન ઉપર 5s માર્ક ઉપર ક્લિક કરો અને વાદળને જમણી બાજુ ડ્રેગ કરો (ખેંચીને ખસેડો). આકૃતિ 5.13 (a) અને (b)માં '0f' અને '5s' જગ્યા પરની એનિમેશનની સ્થિતિ અનુક્રમે દર્શાવેલી છે.



આકૃતિ 5.13(a) : 0f સ્થિતિ ઉપર વાદળનું એનિમેશન



આકૃતિ 5.13(b) : 5s સ્થિતિ ઉપર વાદળનું એનિમેશન

- હવે એનિમેટ મોડ બંધ કરો અને એનિમેશન જોવા માટે પ્લે બટન દબાવો.
- તમે આકૃતિ 5.14 (a)માં ઈમેજની ઉપર વાદળનું હલનચલન જોઈ શકશો, કારણકે ઈમેજ લેયરની ઉપર ક્લાઉડ લેયર (અથવા NewBLine005 લેયર) છે.
- લેયર પેનલમાં એનિમેશન બદલવા માટે આકૃતિ 5.14(b) માં દર્શાવ્યા પ્રમાણે ક્લાઉડ લેયરને ઈમેજ લેયરની નીચે ડ્રેગ કરો. એનિમેશન જોવા માટે પ્લે બટન ક્લિક કરો. તમે એનિમેશનમાં તફાવત જોઈ શકશો. હવે ઈમેજની પાછળ વાદળનું હલનચલન થઈ રહ્યું છે.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	NewBLine005 Region	0.000000
<input checked="" type="checkbox"/>	images1.jpg	1.000000

આકૃતિ 5.14(a) : લેયર્સની વાસ્તવિક સ્થિતિ

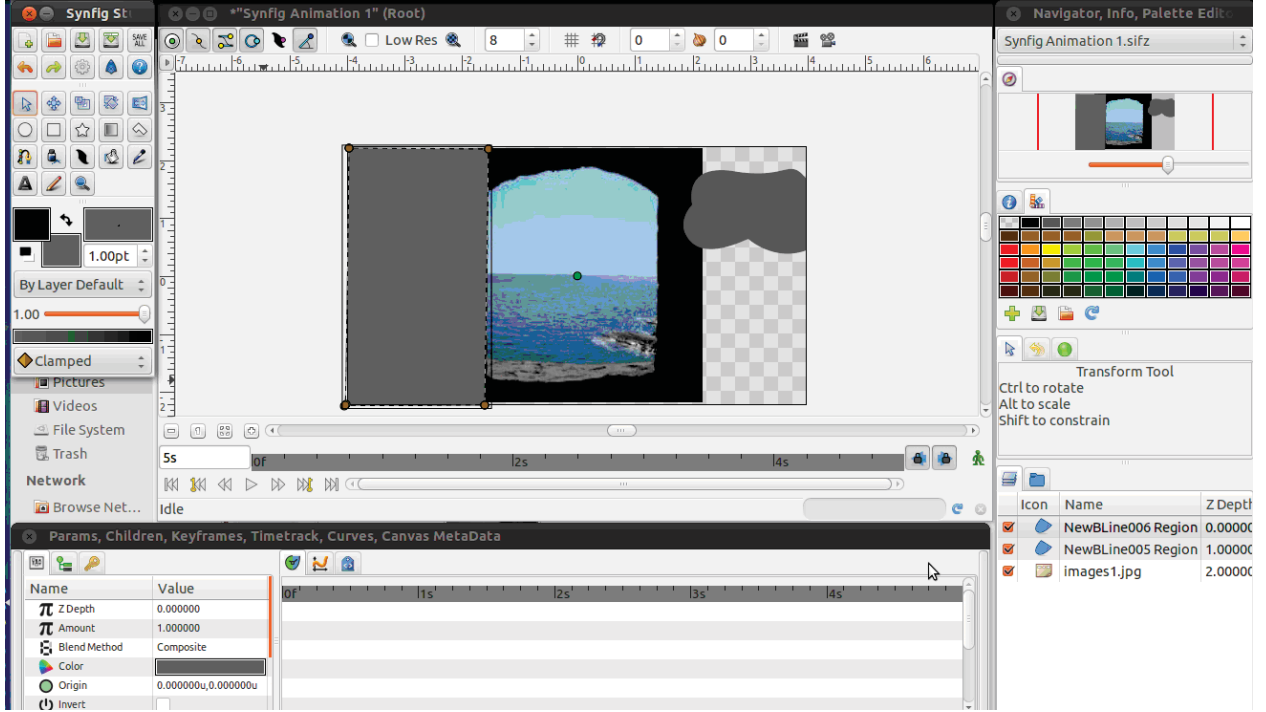
Icon	Name	Z Depth
<input checked="" type="checkbox"/>	images1.jpg	0.000000
<input checked="" type="checkbox"/>	NewBLine005 Region	1.000000

આકૃતિ 5.14 (b) : લેયર્સમાં ફેરફાર પછીની સ્થિતિ

પણ આપણે ફક્ત વિન્ડો ચિત્રમાંથી વાદળનું હલનચલન જોવા ઈચ્છીએ છીએ. આ કામ આપણે માસ્કિંગ કરીને કરીશું. માસ્કિંગનો ઉપયોગ ઓબ્જેક્ટને છુપાવવા અથવા ઉઘાડા પાડવા માટે (બતાવવા માટે) થાય છે.

ચાલો, આપણે પહેલાં ઓબ્જેક્ટને છુપાવવા (સંતાડવા) માટે માસ્કિંગનો ઉપયોગ કરીએ. અહીં આપણે ઈમેજના બન્ને છેડા ઉપર વાદળ ઓબ્જેક્ટને છુપાવવા ઈચ્છીએ છીએ.

- સૌપ્રથમ આપણે કેનવાસની બન્ને બાજુએ વાદળને છુપાવવા માટે માસ્ક આકાર બનાવવો પડશે. આપણે માસ્ક આકાર બનાવવા માટે બીલાઈન ટૂલનો ઉપયોગ કરીશું. બીલાઈન ટૂલ પસંદ કરી તેમાંથી Tool વિકલ્પમાંથી ‘Create Region Bline’ પસંદ કરો. આકૃતિ 5.15માં દર્શાવ્યા પ્રમાણે લંબચોરસ આકાર દોરો. બીલાઈન પૂરું કરવા માટે છેલ્લા બિંદુ ઉપર રાઈટ ક્લિક કરો અને ‘Loop Bline’ પસંદ કરો.



આકૃતિ 5.15 : બીલાઈન ટૂલ વડે માસ્ક આકારની રચના

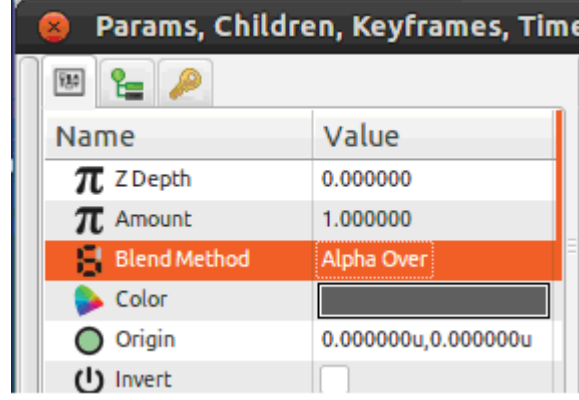
- જે ભાગને છુપાવવો હોય તેના ઉપર માસ્ક આકાર ગોઠવો. આપણા ઉદાહરણમાં ક્લાઉડ લેયર ઉપર આપણે માસ્ક આકાર ગોઠવીશું. આ ઉપરાંત આપણે ફક્ત ક્લાઉડ લેયર ઉપર જ છુપાવવા ઈચ્છતા હોવાથી આપણે ક્લાઉડ લેયર અને માસ્ક આકારને પ્રાવૃત્ત (encapsulate) કરીશું. આકૃતિ 5.16માં પ્રાવૃત્ત લેયર ‘inline canvas’ બનાવેલું છે અને ઈનલાઈન કેનવાસની માહિતી પણ દર્શાવેલ છે. આપણે NewBline005નું નામ બદલીને ક્લાઉડ લેયર અને NewBline006નું નામ બદલીને માસ્ક લેયર કરેલું છે. લેયરના નામ ઉપર ક્લિક કરવાથી નામ બદલવાની તે પરવાનગી આપે છે.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.00000
<input checked="" type="checkbox"/>	images1.jpg	1.00000

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	mask	0.000000
<input checked="" type="checkbox"/>	cloud	1.000000
<input checked="" type="checkbox"/>	images1.jpg	1.000000

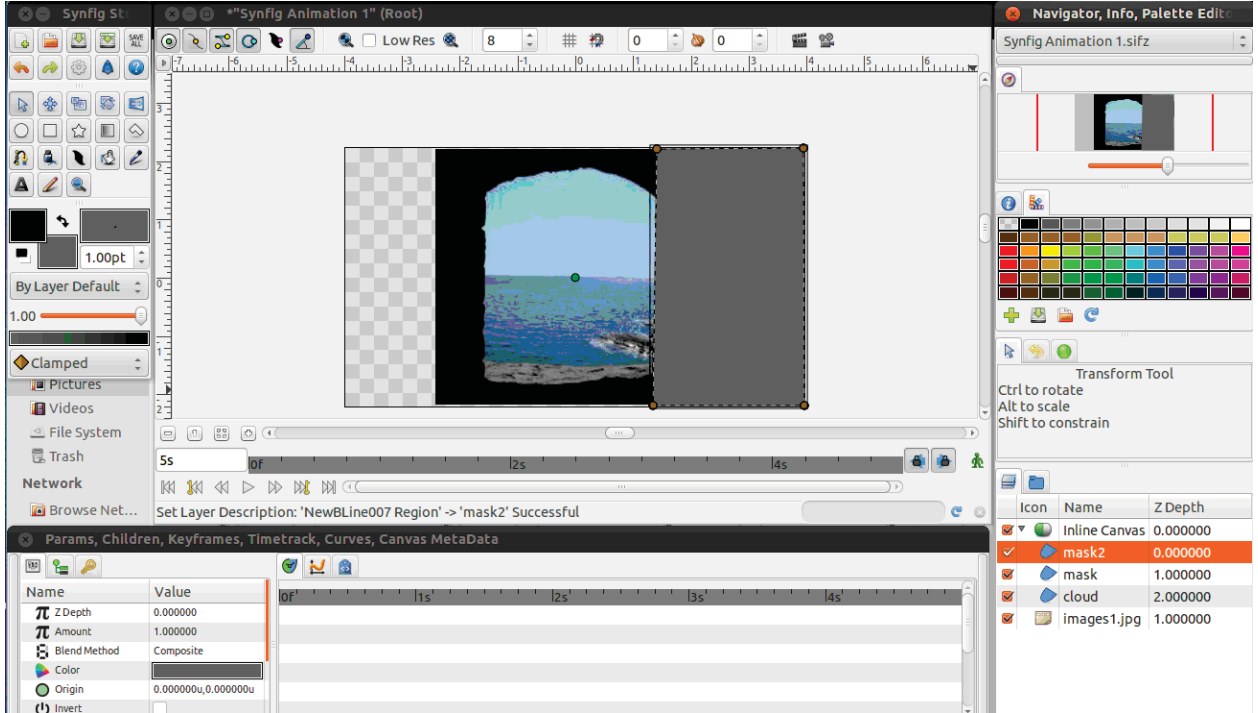
આકૃતિ 5.16 : પ્રાવૃત્ત માસ્ક આકાર અને ક્લાઉડ લેયર

- હવે લેયર્સ પેનલમાંથી માસ્ક લેયર પસંદ કરો અને પ્રાયલ પેનલમાં (પેરામીટર પેનલમાં) 'Blend method' બદલીને 'Alpha over' સેટ કરો. આકૃતિ 5.17માં 'blend method'ને બદલી 'Alpha over' દર્શાવેલ છે.



આકૃતિ 5.17 : 'Blend method'માંથી 'Alpha over'માં બદલો

- હવે એનિમેશન પ્લે કરો અને તમે જોઈ શકશો કે હવે કેનવાસની ડાબી બાજુએ વાદળ દૃષ્ટિગોચર થતું નથી.
- કેનવાસની જમણી બાજુ પણ આ જ કામ કરવાની જરૂર છે. અગાઉ આપણે ડાબી બાજુએ કર્યું હતું તે પ્રમાણે જ કેનવાસની જમણી બાજુ બીલાઈન ટૂલના ઉપયોગ વડે એક લંબચોરસ માસ્ક આકાર દોરો. આકૃતિ 5.18માં કેનવાસની જમણી બાજુએ માસ્ક આકાર દર્શાવેલ છે.



આકૃતિ 5.18 : કેનવાસની જમણી બાજુ માસ્ક આકારની રચના

- આ લેયરનું નામ બદલીને mask2 રાખો અને આકૃતિ 5.19માં દર્શાવ્યા પ્રમાણે માસ્ક લેયરની ઉપરના પ્રાવૃત્ત લેયરમાં mask2 લેયરને ડ્રેગ કરીને રાખો.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	mask2	0.000000
<input checked="" type="checkbox"/>	mask	1.000000
<input checked="" type="checkbox"/>	cloud	2.000000
<input checked="" type="checkbox"/>	images1.jpg	1.000000

આકૃતિ 5.19 : mask2 લેયરની ફેરગોઠવણી

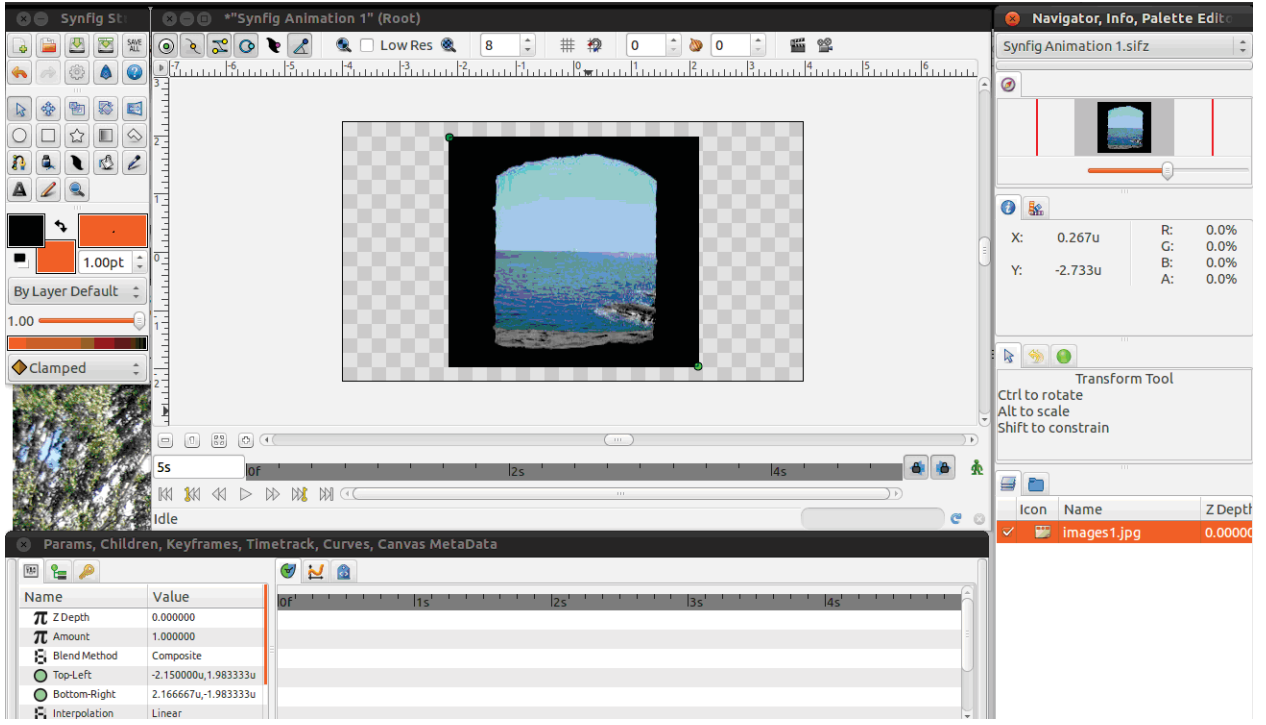
- હવે લેયર પેનલમાંથી mask2 લેયર પસંદ કરો અને પ્રાયલોની પેનલમાં 'blend method' બદલીને 'Alpha over' સેટ કરો.
- હવે એનિમેશન પ્લે કરો અને હવે તમે વાદળને ચિત્રમાંથી પસાર થતું જોશો.
- ફાઇલને સેવ અને રેન્ડર કરો. આકૃતિ 5.20માં વેબબ્રાઉઝર પરનો આઉટપુટ દર્શાવ્યો છે.



આકૃતિ 5.20 : વેબબ્રાઉઝર ઉપર પ્રદર્શિત આઉટપુટ

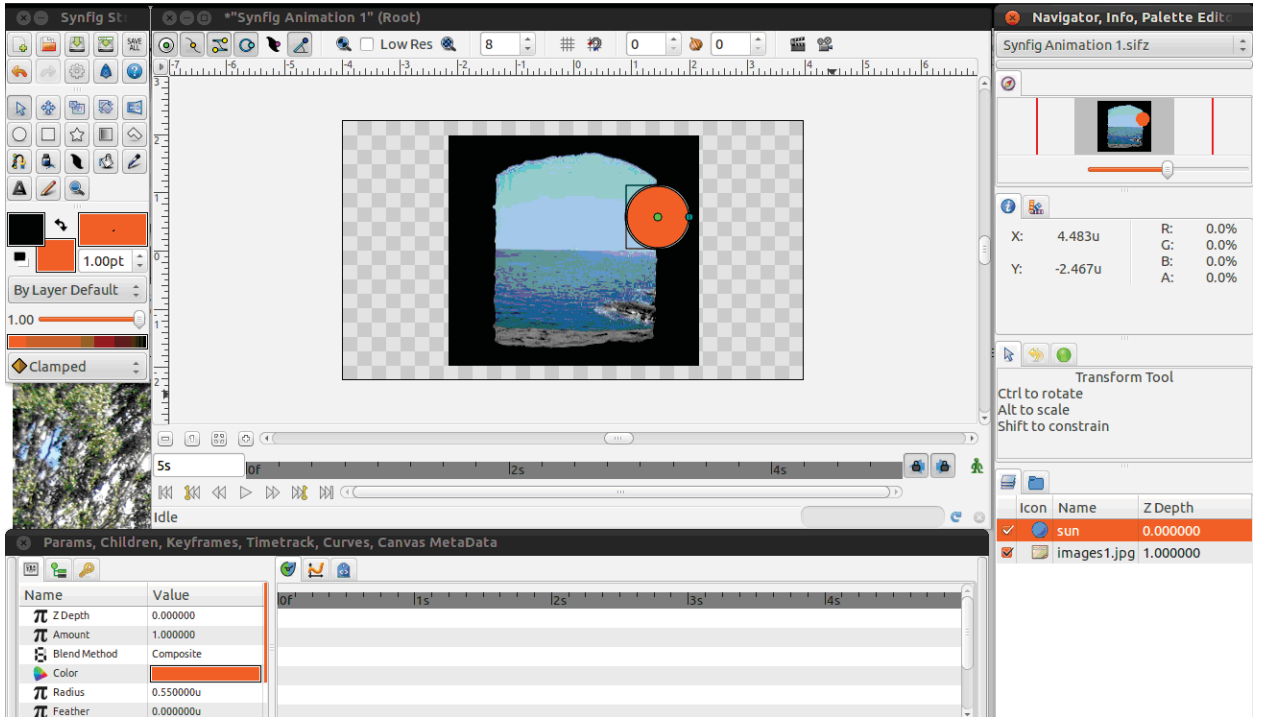
માસ્કિંગનો ઉપયોગ કરીને આ રીતે ઓબ્જેક્ટને છુપાવી શકાય છે. ઓબ્જેક્ટને પુનઃ દર્શાવવા માટે પણ માસ્કિંગનો ઉપયોગ કરી શકાય છે. ચાલો, ધારો કે આપણે ચિત્રમાં સૂર્ય બતાવવા ઈચ્છીએ છીએ. ચિત્રમાં સૂર્યનો ફક્ત કેટલોક ભાગ જ દેખાય છે, અહીં આપણે ઓબ્જેક્ટને પુનઃ દર્શાવવા માટે માસ્કિંગનો ઉપયોગ કરીએ.

- સૌપ્રથમ એક નવી ફાઇલ બનાવો.
- CTRL + i કમાન્ડ આપી કેનવાસ ઉપર ઈમેજ આયાત કરો. ચિત્ર પસંદ કરો. આકૃતિ 5.21માં આયાત કરેલી ઈમેજ દર્શાવેલ છે અને લેયર્સ પેનલમાં ઈમેજ પેનલ ઉમેરાયેલી છે.



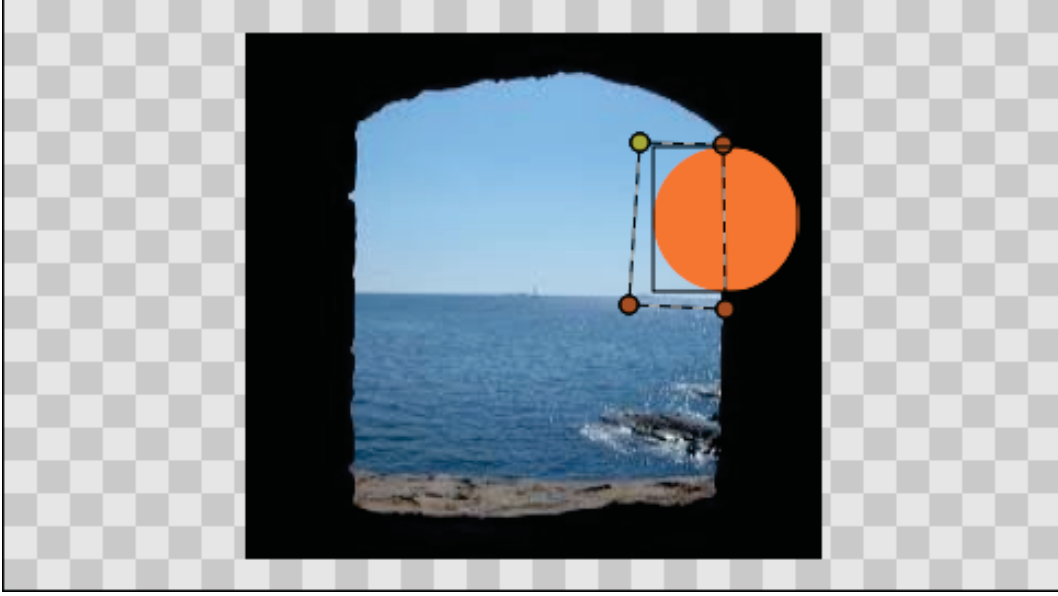
આકૃતિ 5.21 : ઇમેજને આયાત કરો

- વર્તુળ દોરવા માટે સર્કલ ટૂલ પસંદ કરો અને પેલેટમાંથી રંગ ભરો. આકૃતિ 5.22માં દર્શાવ્યા પ્રમાણે એક નારંગી રંગનું વર્તુળ દોરો. તે લેયરનું નામ બદલીને 'sun' નામ કરો.



આકૃતિ 5.22 : એક વર્તુળ દોરો

- BLine ટૂલ વડે એક માસ્ક આકાર દોરો. અહીં આપણે માસ્ક દોરવાની જરૂર છે કે જે સૂર્યના કેટલાક ભાગને ખુલ્લો રાખે. આકૃતિ 5.23માં માસ્ક દર્શાવેલ છે. આકૃતિ 5.24માં દર્શાવ્યા પ્રમાણે લેયરને 'mask' નામ આપો.



આકૃતિ 5.23 : માસ્ક આકાર દોરો

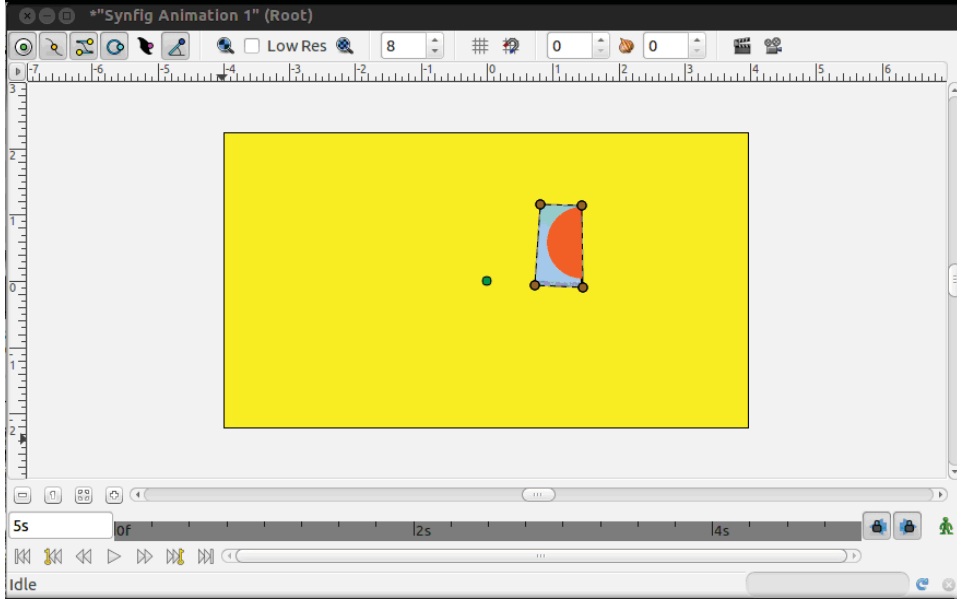
Icon	Name	Z Depth
<input checked="" type="checkbox"/>	mask	0.000000
<input checked="" type="checkbox"/>	sun	1.000000
<input checked="" type="checkbox"/>	images1.jpg	2.000000

આકૃતિ 5.24 : નવા બીલાઈન લેયરનું નામ બદલો

- હવે માસ્ક લેયર પસંદ કરો અને આકૃતિ 5.25માં દર્શાવ્યા પ્રમાણે પ્રાયલ પેનલમાંથી ‘invert’ વિકલ્પ પસંદ કરો. આકૃતિ 5.26માં ‘invert’ વિકલ્પ પસંદ કર્યા પછી કેનવાસ ઉપરની અસર દર્શાવેલી છે.

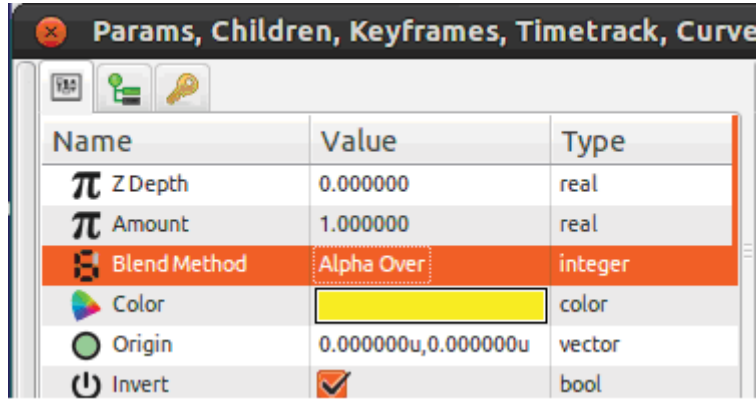
Name	Value	Type
Blend Method	Composite	integer
Color		color
Origin	0.000000u,0.000000u	vector
Invert	<input checked="" type="checkbox"/>	bool
Antialiasing	<input checked="" type="checkbox"/>	bool
Feather	0.000000u	real

આકૃતિ 5.25 : પ્રાયલ પેનલમાંથી ‘Invert’ વિકલ્પની પસંદગી

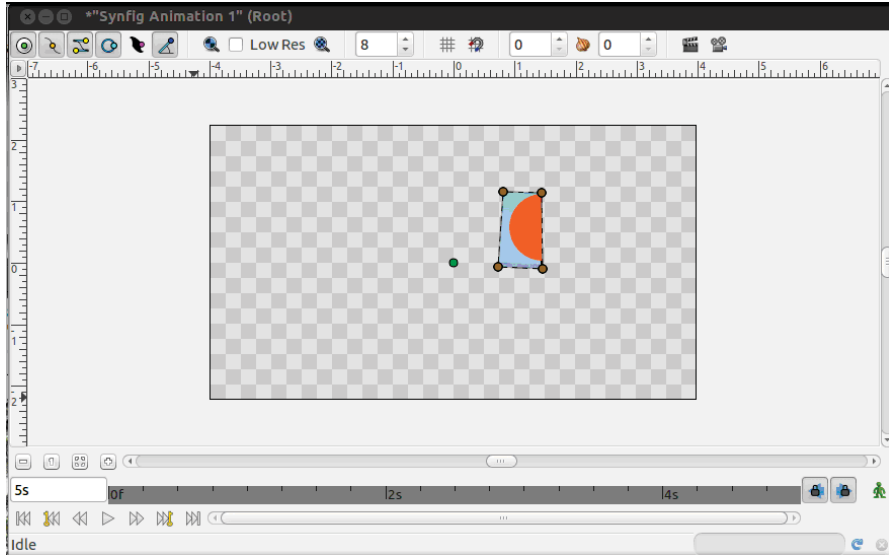


આકૃતિ 5.26 : 'Invert' વિકલ્પ પસંદ કર્યા પછી કેનવાસ ઉપરની અસર

- આકૃતિ 5.27માં દર્શાવ્યા પ્રમાણે 'blend method' બદલીને 'Alpha over' કરો. આકૃતિ 5.28માં દર્શાવ્યા પ્રમાણે હવે કેનવાસ દેખાશે.



આકૃતિ 5.27 : 'blend method' બદલીને 'Alpha over' કરવી



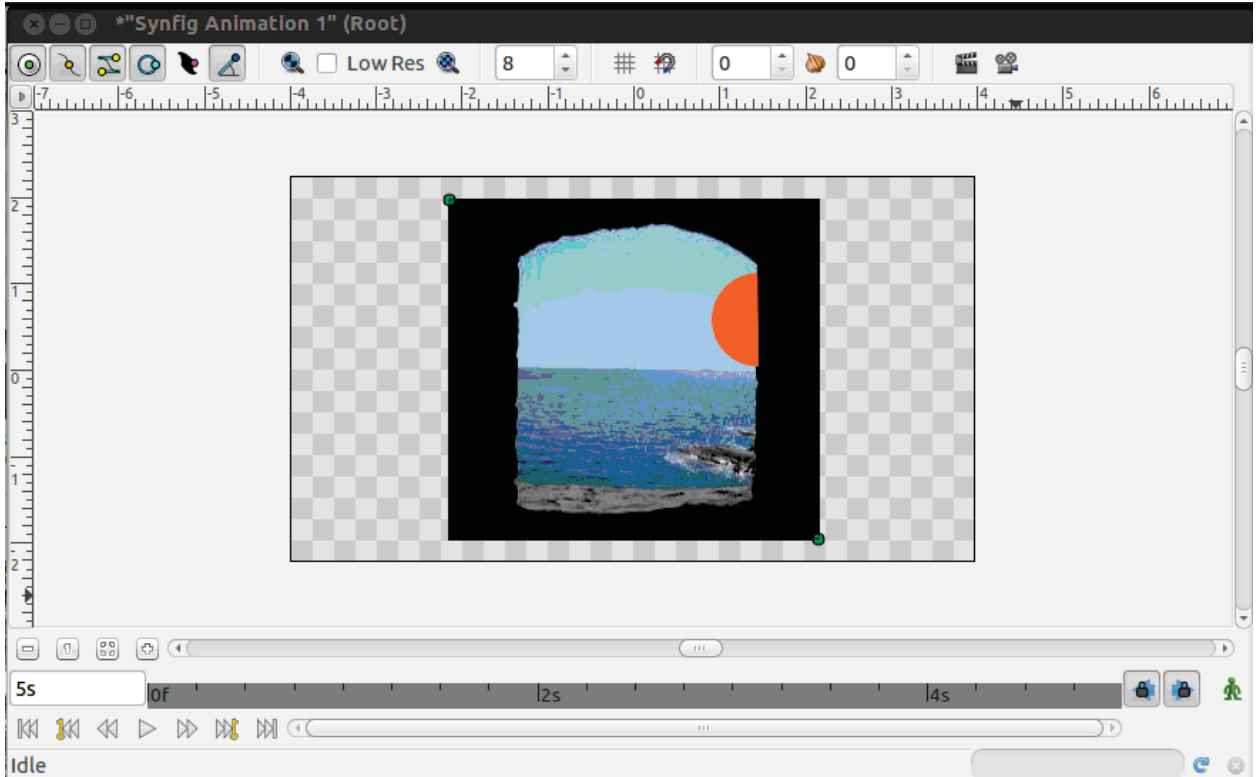
આકૃતિ 5.28 : બ્લેન્ડ મેથડ બદલ્યા પછી કેનવાસ ઉપર અસર

- માસ્ક લેયરની નીચેના ઓબ્જેક્ટ ફક્ત જોઈ શકાય છે, કારણકે ઈમેજ લેયર માસ્ક લેયરની નીચે રાખેલ છે. માસ્કની અસર આપણને ઈમેજ ઉપર નહીં પણ ફક્ત Sun લેયર ઉપર જોઈએ છે, આથી આકૃતિ 5.29માં દર્શાવ્યા પ્રમાણે બંનેને પ્રાવૃત્ત કરવાની જરૂર છે.

Icon	Name	Z Depth
<input checked="" type="checkbox"/>	Inline Canvas	0.000000
<input checked="" type="checkbox"/>	mask	0.000000
<input checked="" type="checkbox"/>	sun	1.000000
<input checked="" type="checkbox"/>	images1.jpg	1.000000

આકૃતિ 5.29 : માસ્ક અને સન લેયરને પ્રાવૃત્ત કરો

- આપણે માસ્કિંગ અસર આકૃતિ 5.30માં જોઈ શકીએ છીએ. માસ્ક લેયરની નીચે આપણે ગમે એટલા ઓબ્જેક્ટ રાખી શકીએ છીએ જે આપણે બતાવવાના હોય. જે ઓબ્જેક્ટ ઉપર માસ્કિંગની અસર ન ઈચ્છતા હોય તેને આપણે પ્રાવૃત્ત લેયરની બહાર રાખવા જોઈએ.



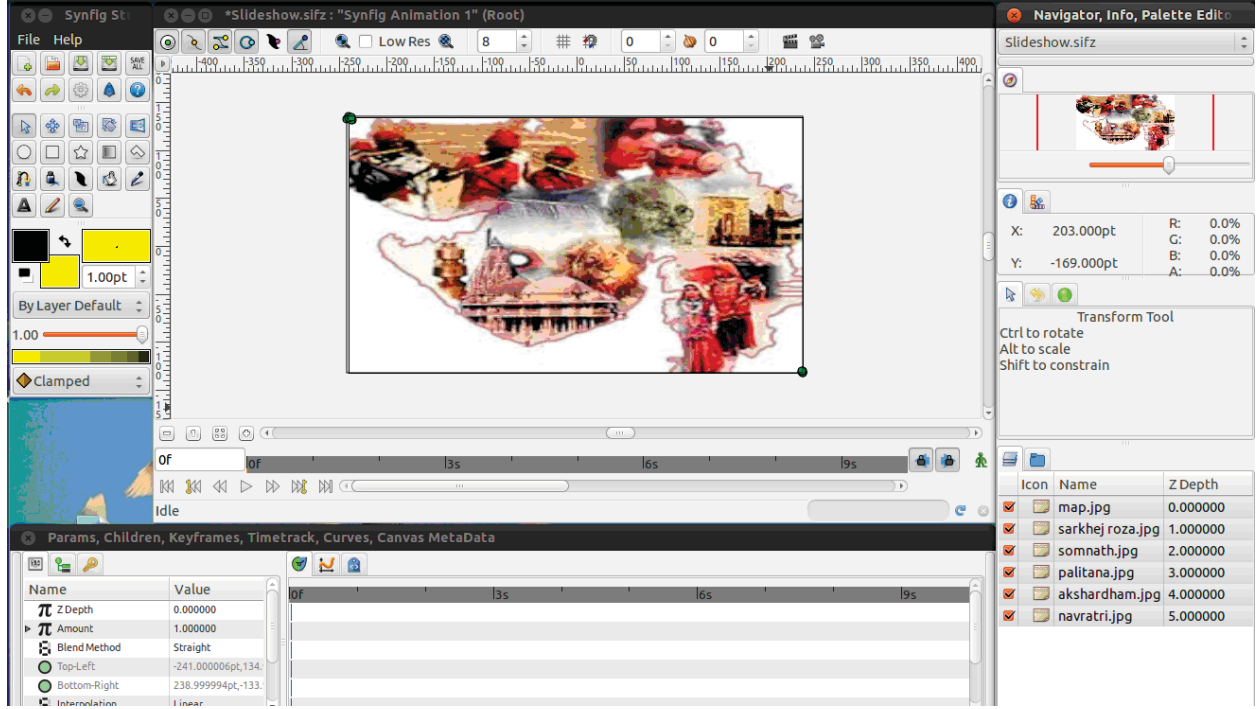
આકૃતિ 5.30 : ઓબ્જેક્ટ બતાવવામાં માસ્કિંગની અસર

સીન્કિંગ દ્વારા સ્લાઈડ શોની રચના (Create slideshow using Synfig)

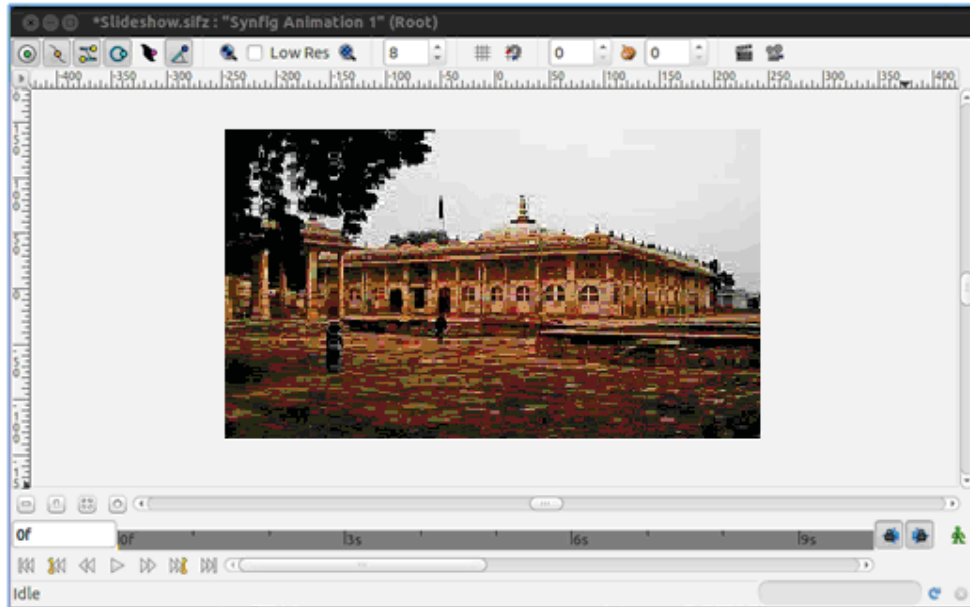
ધારો કે તમે ‘ગુજરાત ટૂરિઝમ’ ઉપર એક સ્લાઈડ શોની રચના કરવા ઈચ્છો છો જેમાં તમે ગુજરાતની ઝલક આપવા માગો છો. આ ઝલક આપવા માટે તમારે એક પછી એક ચિત્ર પ્રદર્શિત કરવાં પડશે. એક સમયે ફક્ત એક જ ચિત્ર દેખાડવા ઈચ્છો છો. આપણે દરેક ચિત્ર ઉપર અલગ-અલગ કામ કરવા ઈચ્છતા હોવાથી તે દરેક ચિત્ર જુદાં-જુદાં લેયર ઉપર મૂકવાં જોઈએ. ચિત્ર બતાવવાનું સંચાલન કરવા માટે આપણે દરેક લેયર માટે ‘blend method’ અને ‘amount’ પ્રાયલોનો ઉપયોગ કરીશું. આપણે અગાઉ બ્લેન્ડ મેથડ ગુણધર્મ વાપર્યો, જે તેની નીચેના ઈમેજ લેયરની માહિતી આપે છે. ‘alpha’ કિંમતની જેમ જ ઈમેજને

બતાવવા માટે 'amount' ગુણધર્મ વાપર્યો છે. Amount ની કિંમત 1 લેયરને પૂરેપૂરું દૃશ્યમાન સૂચવે છે, જ્યારે તેની કિંમત 0 લેયરને પૂરેપૂરું પારદર્શક સૂચવે છે. સ્લાઇડ શોની રચના કરવા માટે નીચે જણાવેલાં પગલાંઓ મુજબ કાર્ય કરો :

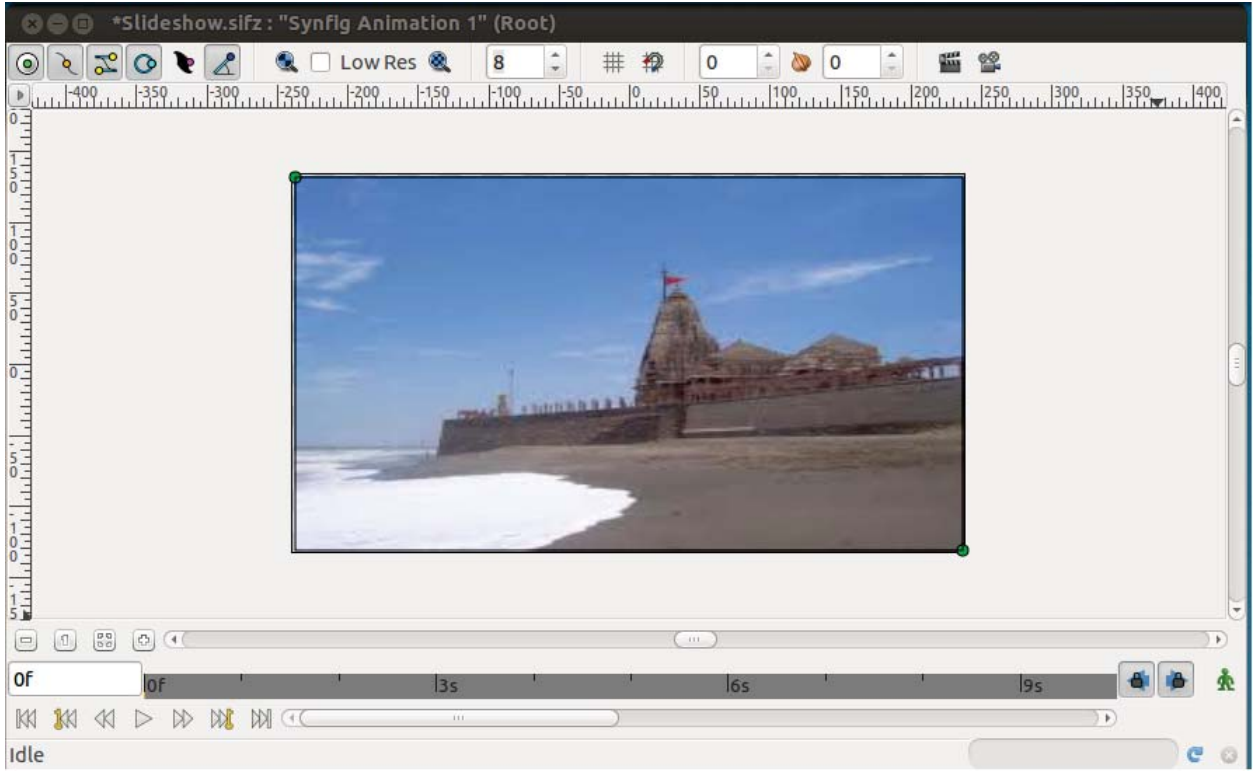
- એક નવી ફાઇલ બનાવો.
- હવે ચિત્ર આયાત કરવા માટે CTRL + i દબાવો. જો તમારી પાસે ચિત્ર ન હોય, તો ઇન્ટરનેટ ઉપરથી કેટલીક ઇમેજ ડાઉનલોડ કરો. તે પછી કેનવાસ ઉપર એક પછી એક ઇમેજ આયાત કરો. દરેક ઇમેજને અલગ અલગ લેયર ઉપર રાખો. આકૃતિ 5.31 (a), (b), (c), (d), (e), (f)માં લેયર્સ પેનલમાં આયાત કરેલા છ ચિત્રોને દર્શાવેલ છે. ઇમેજનું કદ કેનવાસનાં કદ પ્રમાણે બદલો.



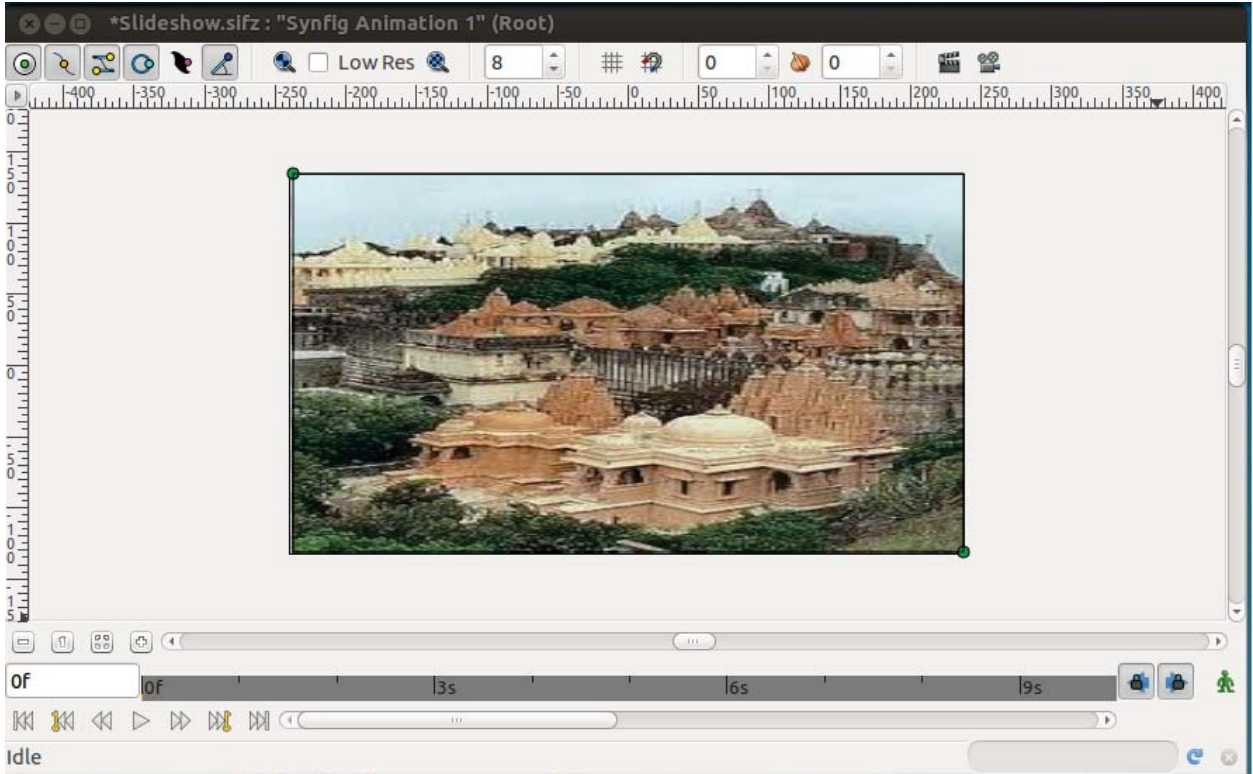
આકૃતિ 5.31 (a) : ગુજરાતના નકશાનું આયાત કરેલું ચિત્ર



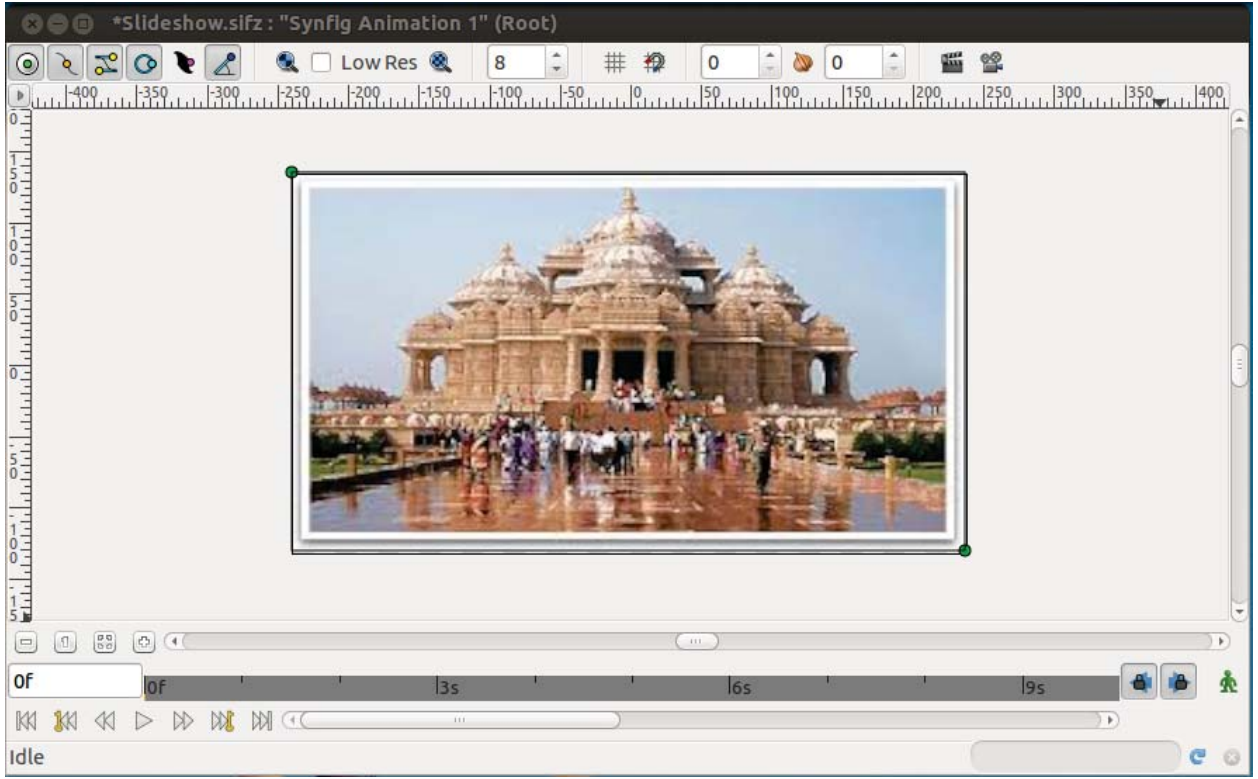
આકૃતિ 5.31 (b) : સરખેજ રોજાનું આયાત કરેલું ચિત્ર



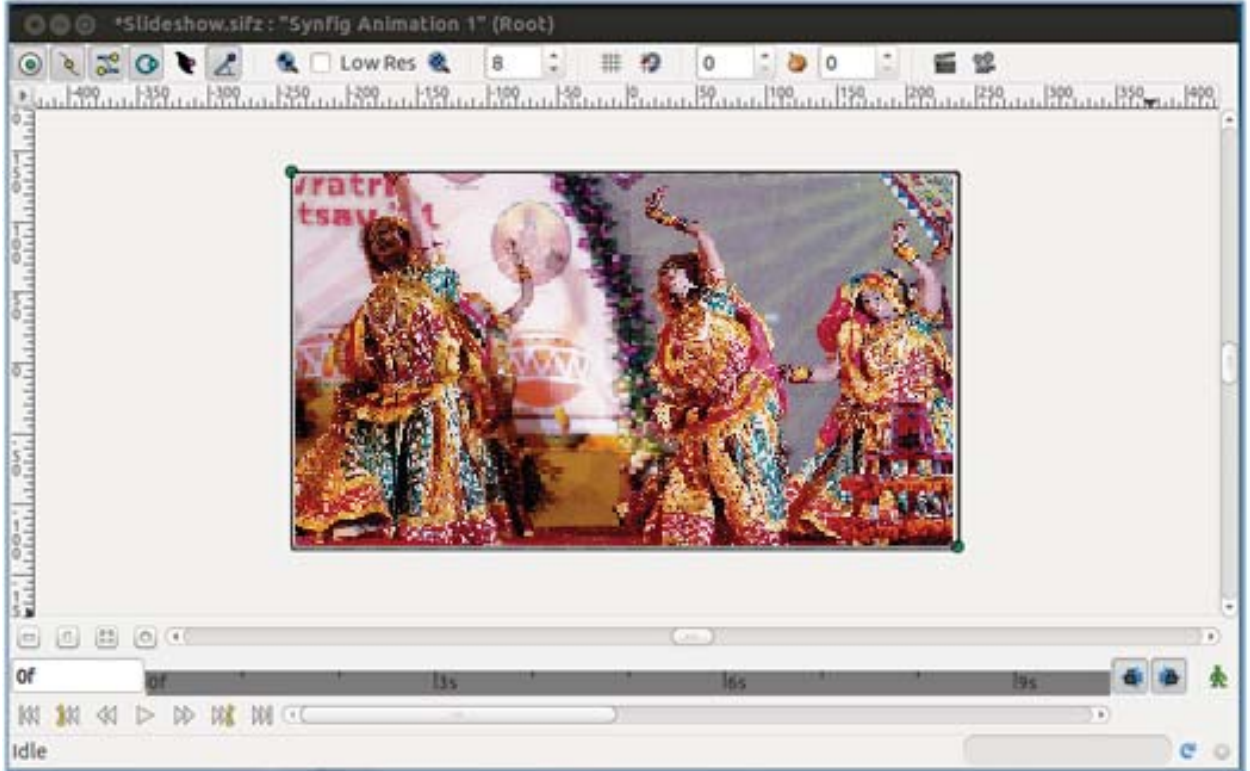
आकृति 5.31(c) : सोमनाथनुं आयात करेलुं चित्र



आकृति 5.31 (d) : पाळीताणानुं आयात करेलुं चित्र

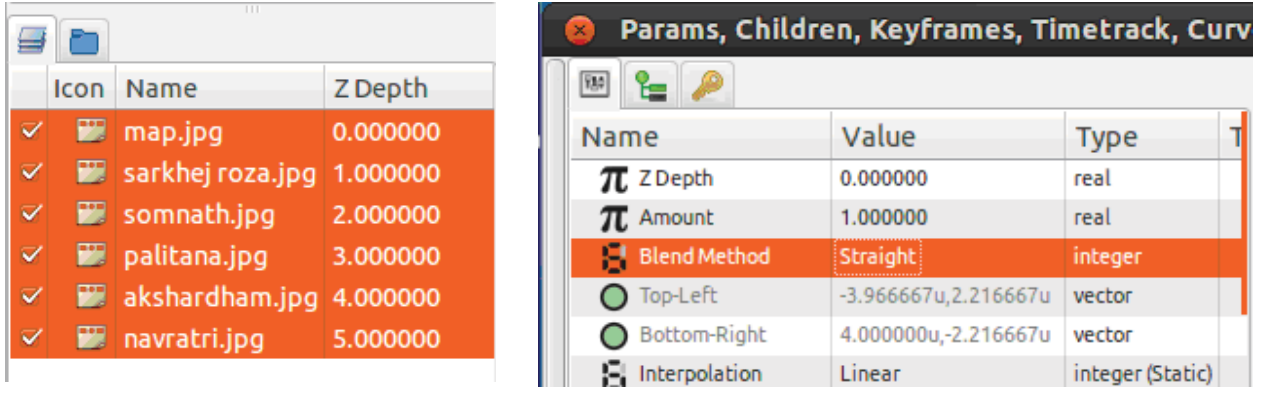


આકૃતિ 5.31 (e) : અક્ષરધામનું આયાત કરેલું ચિત્ર



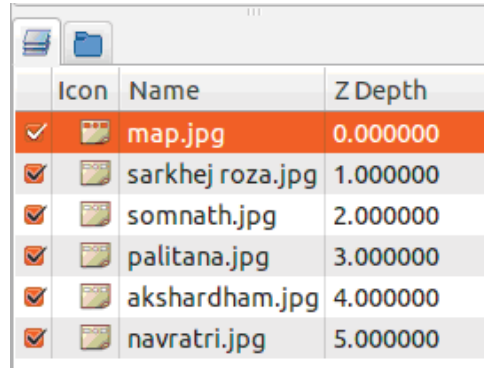
આકૃતિ 5.31 (f) : નવરાત્રીનું આયાત કરેલું ચિત્ર

- શિક્ષક કી દબાવીને લેયર્સ પેનલની બધી ઇમેજિસને પસંદ કરો. આકૃતિ 5.31માં દર્શાવ્યા પ્રમાણે પ્રાયલની પેનલમાં જઈને બધી ઇમેજિસ માટે બ્લેન્ડ મેથડ બદલીને 'straight' કરો.

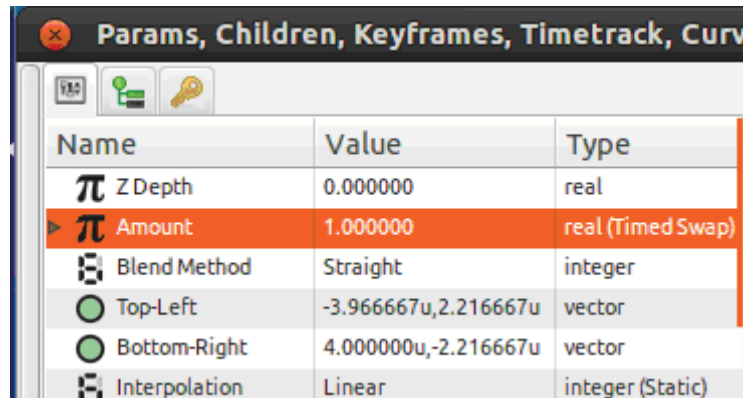


આકૃતિ 5.32 : ઈમેજિસને પસંદ કરી બ્લેન્ડ મેથડ બદલી **straight** કરો

- બ્લેન્ડ મેથડ પ્રોપર્ટી સેટ કર્યા પછી આપણે 'Amount' પ્રાયલ ઉપર કામ કરવું પડશે. બ્લેન્ડ મેથડની જેમ બધાં જ લેયર્સ માટે એકસાથે 'Amount' પ્રાયલ સેટ કરી શકાતું નથી. દરેક લેયર માટે 'Amount' પ્રોપર્ટી માટે અલગ-અલગ કાર્ય કરવું પડશે. અહીં Amount પ્રોપર્ટીને 'Timed swap'માં બદલવું પડશે.
- આકૃતિ 5.33માં દર્શાવ્યા પ્રમાણે સૌથી ઉપર મેપ લેયર પસંદ કરો અને પ્રાયલ પેનલમાં Amount પ્રાયલ પસંદ કરી રાઈટ ક્લિક કરો. રાઈટ ક્લિક કરવાથી કન્ટેક્સ્ટ મેનૂ (context menu) ખૂલશે. પસંદ કરો : Convert → Timed swap. 'convert' સૂચવે છે કે પ્રાયલ જુદી-જુદી રીતે આપોઆપ નિયંત્રિત રહે છે. દરેક પ્રાયલને કન્વર્ટ કરી શકાય છે અને તેમાં વિવિધ વિકલ્પો રહેલા છે. આકૃતિ 5.34માં Amount પ્રોપર્ટીમાં Timed swap દર્શાવેલ છે. આકૃતિ 5.35માં દર્શાવ્યા પ્રમાણે Amount પ્રોપર્ટીમાં નવાં પેટાં પ્રાયલો ઉમેરાશે : 'Before', 'After', 'Time' અને 'Length'. આ વિકલ્પો ચિત્રની દૃશ્યતા સૂચવે છે. 'length'નો ઉપયોગ કરીને 'before'માંથી 'after' પ્રાયલની અદલાબદલી કરે છે અને 'time' દર્શાવીને અદલાબદલી પૂરી કરી શકાય છે.



આકૃતિ 5.33 : સૌથી ઉપરનું ચિત્ર લેયર પસંદ કરો



આકૃતિ 5.34 : એમાઉન્ટ પ્રાયલને **Timed swap**માં બદલો

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	1.000000	real
Time	2s	time
Length	1s	time

આકૃતિ 5.35 : એમાઉન્ટ પ્રાયલમાં ઉમરાયેલ નવું પ્રાયલ

- હવે પ્રાયલ 'before' ને 1.0 અને 'after'ને 0.0થી સેટ કરો. પ્રાયલ 'before'ને 1.0 સેટ કર્યું તે સૂચવે છે કે પ્રાયલ 'time'માં સૂચવેલા સમય પહેલાં લેયર દૃશ્યમાન હશે અને પ્રાયલ 'after'ને 0.0 સેટ કર્યું તે સૂચવે છે કે 'time'માં સૂચવેલા સમય પછી લેયર અદૃશ્ય હશે. જો તમે દરેક વચ્ચે 1 સેકન્ડ સંક્રમિત સમય (transition time) સાથે દરેક ચિત્ર 5 સેકન્ડ માટે પ્રદર્શિત કરવા માંગો છો, તો પ્રાયલ 'length'ને '1s' થી અને time ને '5s'થી બદલો. આથી 5 સેકન્ડ પછી ઇમેજ દેખાતી બંધ થઈ જશે. આકૃતિ 5.36માં પ્રાયલ amountમાં સેટ કરેલી કિંમત દર્શાવી છે.

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
Time	5s	time
Length	1s	time

આકૃતિ 5.36 : મેપ લેયરનાં પેટા પ્રાયલોનાં સેટિંગ

- હવે આગળનું ઇમેજ લેયર પસંદ કરો. આપણે અગાઉના લેયરમાં કર્યું હતું, તે જ પ્રમાણે ટાઈમની અદલાબદલી કરો. પ્રાયલ 'before'ને 1.0થી, 'after'ને 0.0થી અને 'length'ને 1sથી સેટ કરો. 'time' ને 10sથી સેટ કરો. આથી 10s પછી અદલાબદલી થશે અને ઇમેજ 10s પછી દેખાતી બંધ થશે. આ જ રીતે પછીનાં

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
Time	10s	time
Length	1s	time

આકૃતિ 5.37 (a) :

સરખેજ રોજા લેયરનાં ટાઈમ સેટિંગ

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
Time	15s	time
Length	1s	time

આકૃતિ 5.37 (b) :

સોમનાથ લેયરનાં ટાઈમ સેટિંગ

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
π Time	20s	time
π Length	1s	time

આકૃતિ 5.37 (c) :

પાલીતાણા લેયરનાં ટાઈમ સેટિંગ

ઈમેજ લેયર્સ 'time' 15s, 20s, 25s પછી બદલાશે. આકૃતિ 5.37 (a), (b), (c), (d) અને (e) માં બધા ઈમેજ લેયર્સનાં સેટિંગ્સ બતાવ્યાં છે.

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
π Time	25s	time
π Length	1s	time

આકૃતિ 5.37 (d) :

અક્ષરધામ લેયરનાં ટાઈમ સેટિંગ

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real (Timed Swa
π Before	1.000000	real
π After	-0.000000	real
π Time	30s	time
π Length	1s	time

આકૃતિ 5.37 (e) : નવરાત્રિ લેયરનાં ટાઈમ સેટિંગ

- હવે આ પ્રમાણે કમાન્ડ આપો : Caret → Edit → Properties. ટાઈમ ટેબને પસંદ કરીને અંતસમયને છેલ્લી ઈમેજ લેયરની Time કિંમત બરાબર સેટ કરો. આપણા દાખલામાં છેલ્લા ઈમેજ લેયરનો Time 30s છે. જો અંતમાં તમારે બ્લેન્ક સ્ક્રીન જોઈતો હોય, તો અંતસમય 30s કરતાં થોડો વધારે સેટ કરો. આકૃતિ 5.38માં અંતસમયનાં સેટિંગ દર્શાવ્યાં છે.

Properties - Synfig Animation 1

Canvas Info

Name: Synfig Animation 1

Description:

Image Time Other

Time Settings

Frames per second: 24.00000

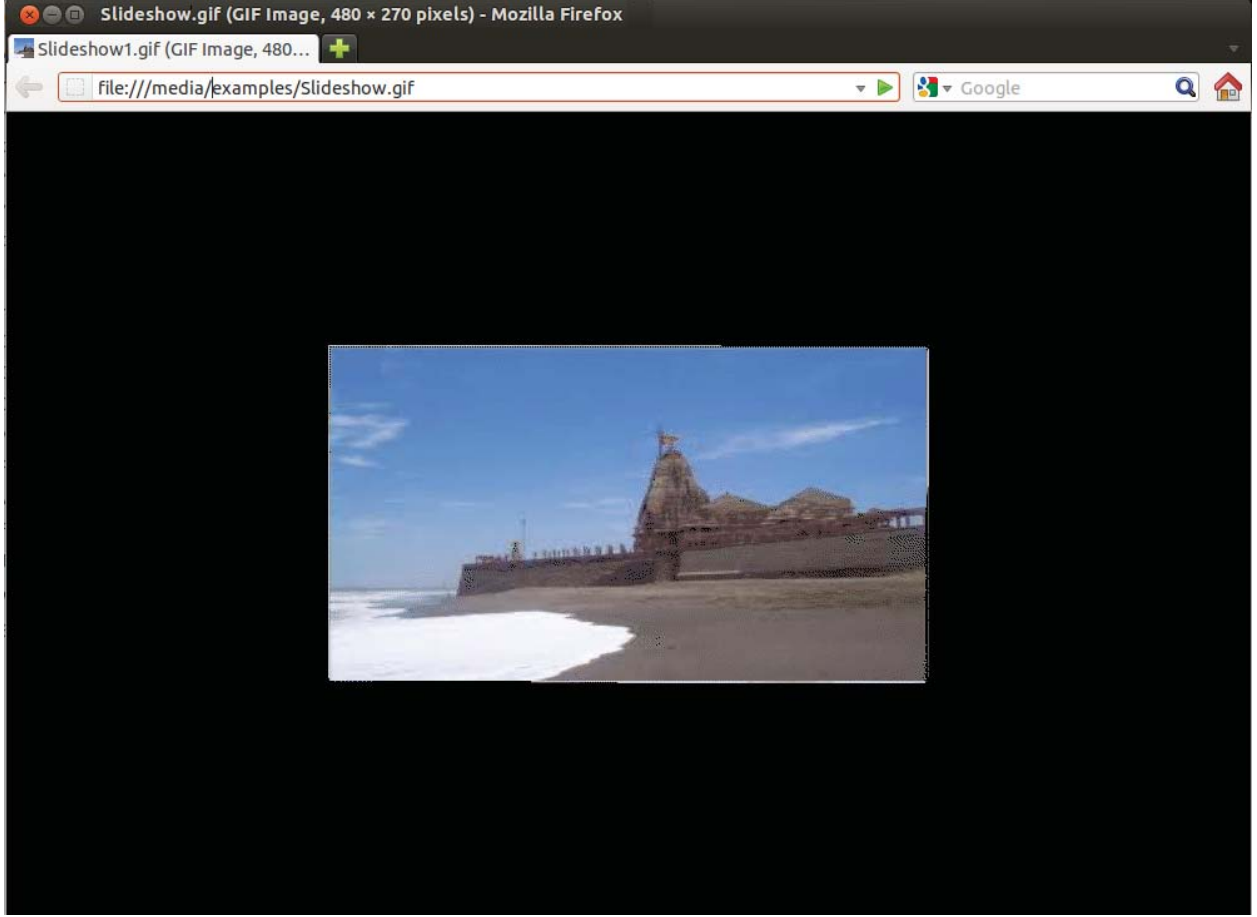
Start Time: 0f

End Time: 30s

OK Apply Close

આકૃતિ 5.38 : અંતસમય 30s સેટ કરવો

- રજૂઆતનો પ્રિવ્યૂ જોવા માટે પ્લે બટન દબાવો.
- ફાઇલને સેવ અને રેન્ડર કરો. આકૃતિ 5.39માં વેબબ્રાઉઝર પરનો આઉટપુટ દર્શાવ્યો છે.



આકૃતિ 5.39 : બ્રાઉઝર ઉપર આઉટપુટ

જો તમારે સ્લાઇડ શોમાં પશ્ચાદ્ભૂમિ બતાવવી હોય, તો બધાં ઇમેજ લેયર્સને પ્રાવૃત્ત કરો. પશ્ચાદ્ભૂમિને સૌથી નીચેના લેયર તરીકે રાખો. blend અને amount પ્રાયલ ફક્ત પ્રાવૃત્ત લેયરને લાગુ પાડવામાં આવશે. આપણે ચિત્રમાં કોઈ શાબ્દિક માહિતી પણ ઉમેરી શકીએ અને સ્લાઇડ શો દરમિયાન તે શાબ્દિક માહિતી પ્રદર્શિત થાય.

ક્રિયા-પ્રતિક્રિયા (Interactivity)

એનિમેશનમાં અમુક સમયે વપરાશકર્તાની પસંદ પ્રમાણે તમારે તેમાં આગળ વધવાનું હોય છે. કેટલાક સમયે વપરાશકર્તા પણ એનિમેશનનો અમુક ચોક્કસ ભાગ કુદાવવા (સ્કિપ) ઇચ્છે અને આગળ જવા ઇચ્છે. ક્રિયા-પ્રતિક્રિયા માટે એનિમેશન સોફ્ટવેરમાં વપરાતી આ પ્રકારના પગલાંને એક્શન સ્ક્રિપ્ટ (Action script) કહેવામાં આવે છે. એનિમેશન પ્રોગ્રામ્સમાં લચીલાપણું/સાનુકૂળતા (flexibility) ઉમેરવા માટે એક્શન સ્ક્રિપ્ટ ઉમેરવામાં આવે છે. એક્શન સ્ક્રિપ્ટ શબ્દ એનિમેશન દ્વારા કરવામાં આવતી ક્રિયામાં સાનુકૂળતા ઉમેરવા માટે સ્ક્રિપ્ટિંગ ભાષાનો ઉપયોગ સૂચવે છે. તે વપરાશકર્તાને ચોક્કસ ક્રિયાના એનિમેશન માટે રમવા અથવા એનિમેશન દરમિયાન પોતાની પસંદ પ્રમાણે આગળ પગલાં લેવામાં મદદ કરે છે.

હાલની તારીખમાં, એનિમેશન સોફ્ટવેરમાં સીન્ક્રિંગ સ્ટુડિયો એક્શન સ્ક્રિપ્ટિંગને સીધો ટેકો આપતા નથી પણ આપણા એનિમેશનમાં આપણે જાવા સ્ક્રિપ્ટ અથવા જાવાનો ઉપયોગ કરી શકીએ છીએ. આપણા એનિમેશનમાં જોકે જાવા સ્ક્રિપ્ટ કે જાવા બાબત ઊંડાણપૂર્વક ચર્ચા કરવી અહીં શક્ય નથી. પણ ચાલો, આપણે એક ઉદાહરણની ચર્ચા કરીએ કે જેમાં આપણે એનિમેશનમાં ક્રિયા-પ્રતિક્રિયા ઉમેરવા માટે જાવા સ્ક્રિપ્ટનો ઉપયોગ કર્યો હોય.

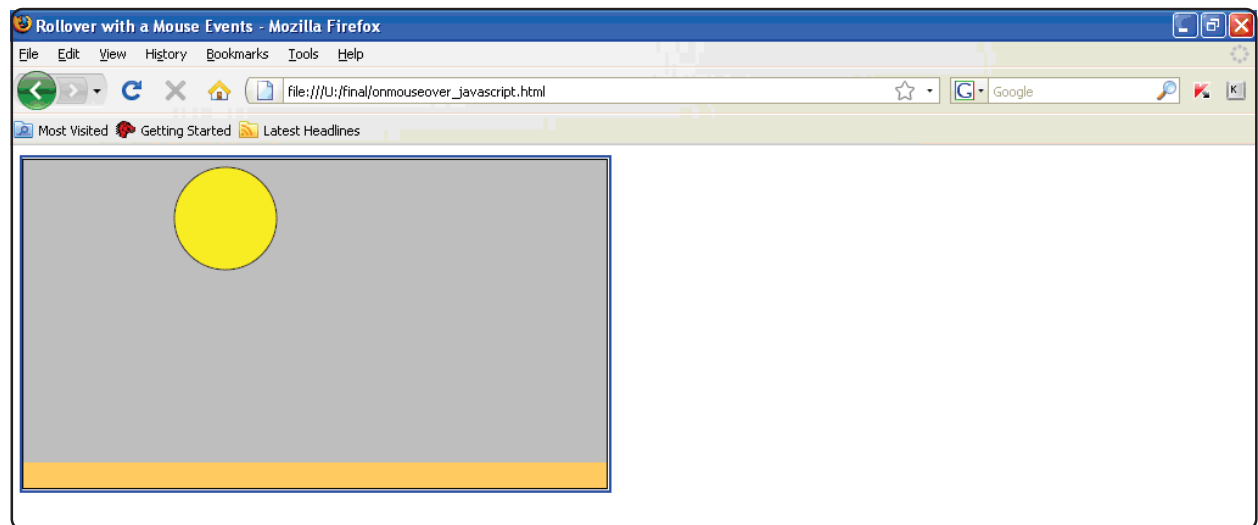
ધારોકે જ્યારે વપરાશકર્તા ચિત્ર ઉપર માઉસ લઈ જાય ત્યારે જ તમે ફક્ત એનિમેશન પ્લે કરવા ઈચ્છો છો. આ કાર્ય હાંસલ કરવા માટે સૌપ્રથમ આપણે બે ઇમેજિસ લઈશું. આમાંની એક ઇમેજ સ્થિર છે. આપણા ઉદાહરણમાં સ્થિર ચિત્ર static-ball.gif છે. બીજી ઇમેજ એનિમેટેડ ચિત્ર છે. આપણા ઉદાહરણમાં એનિમેટેડ ચિત્ર animated-ball.gif છે. એનિમેટેડ ચિત્ર સીન્કિંગમાં gif, jpeg કે bmp જેવાં ફોર્મેટમાં રેન્ડર ફાઈલ છે. આપણે અગાઉના પ્રકરણમાં ચર્ચા કરી તે પ્રમાણે જ્યારે આપણને સીન્કિંગમાં ફાઈલ રેન્ડર કરીએ છીએ, ત્યારે વિકલ્પોની યાદી ઉપલબ્ધ થાય છે. જ્યારે વપરાશકર્તા વેબ બ્રાઉઝર ખોલે છે ત્યારે સ્થિર ચિત્ર પ્રદર્શિત થાય છે. જ્યારે તે/તેણી ચિત્ર ઉપર માઉસ લઈ જાય, ત્યારે જાવા સ્ક્રિપ્ટ કોડના ઉપયોગ વડે એનિમેટેડ ચિત્ર પ્રદર્શિત થશે. જ્યારે વપરાશકર્તા ચિત્રથી માઉસ દૂર ખસેડે ત્યારે એનિમેશન બંધ થશે. આ માટે નીચે જણાવેલાં પગલાંને અનુસરો :

- Geditમાં નવી ફાઈલ બનાવો અને તેમાં નીચે જણાવેલ કોડ લખો :

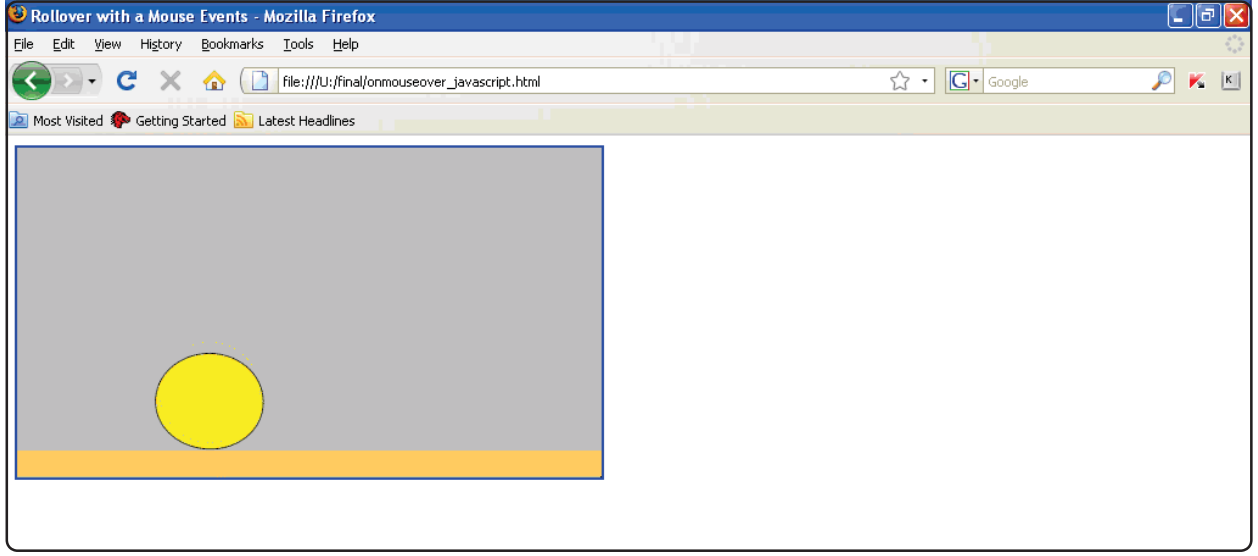
```
<html>
<head>
<title>Rollover with a Mouse Events</title>
<script type="text/javascript">
if( document.images ){
  var image1 = new Image();      // Preload an image
  image1.src = "static-ball.gif";
  var image2 = new Image();      // Preload second image
  image2.src = "animated-ball.gif";
}
</script>
</head>
<body>
<a href="#" onmouseover="document.myImage.src=image2.src" onmouseout="document.myImage.src=image1.src">

</a>
</body>
</html>
```

- ફાઈલનો અનુલંબન .html સાથે સંગ્રહ કરો.
- આઉટપુટ જોવા માટે ફાઈલને વેબ બ્રાઉઝરમાં ખોલો. આકૃતિ 5.40માં પાનું લોડ થતાં સમયનો આઉટપુટ બતાવે છે. આકૃતિ 5.41માં એનિમેટેડ ઇમેજ દર્શાવેલી છે.



આકૃતિ 5.40 : પાનું લોડ થતાં સમયની પ્રદર્શિત ઇમેજ



આકૃતિ 5.41 : જ્યારે માઉસ ઈમેજ ઉપર હોય ત્યારે પ્રદર્શિત એનિમેટેડ ઈમેજ

આ રીતે, જાવા સ્ક્રિપ્ટનો ઉપયોગ કરીને એનિમેશનમાં ક્રિયા-પ્રતિક્રિયાનો સમાવેશ કરી શકાય છે.

ફાઇલ્સની સુવાહ્યતા (Portability of Files)

કોઈ ચોક્કસ એનિમેશન સોફ્ટવેરના ઉપયોગ બનાવેલ એનિમેશન ફાઇલને આગળ કાર્યક્ષમ ઉપયોગ માટે એડોબ ફ્લેશ ફાઇલ્સ (SWF ફાઇલ્સ)માં રૂપાંતરિત કરી શકાય છે. SWFTOOLS એક ઓપનસોર્સ સોફ્ટવેર ટૂલ છે, જે અનેક વિવિધ પ્રકારનાં ફાઇલ ફોર્મેટનું SWF ફાઇલ્સમાં રૂપાંતર કરે છે. SWF એડોબ ફ્લેશ ફાઇલ ફોર્મેટ છે, જે વેક્ટર ગ્રાફિક્સ એનિમેશનમાં વપરાય છે. SWF ફોર્મેટમાં ફાઇલ્સ રૂપાંતર કર્યા પછી તેનો આગળ કાર્યક્ષમ ઉપયોગ કરી શકાય છે અથવા એડોબ ફ્લેશ એનિમેશન સોફ્ટવેરનો ઉપયોગ કરીને બનાવવામાં આવે છે. આ ટૂલ SWF ફાઇલ્સ વાંચવા માટે; તેમને જોડવા માટે; અને ઈમેજિસ, સાઉન્ડ અને વીડિયો ફાઇલ્સ જેવી અન્ય માહિતીમાંથી ફાઇલ બનાવવા માટેના પ્રોગ્રામ્સનો સમાવેશ કરે છે. કોષ્ટક 5.1માં કેટલાક પ્રોગ્રામ્સની યાદી છે, જે SWF ફોર્મેટમાં ફાઇલ્સ રૂપાંતર કરે છે.

Programmes	Description
GIF2SWF	Converts gif files into swf files
PNG2SWF	Converts png files into swf files
JPEG2SWF	Converts jpeg files into swf files
PDF2SWF	Converts pdf files into swf files
WAV2SWF	Converts wav audio files into swf files
AVI2SWF	Converts avi animation files into swf files
FONT2SWF	Converts Font files to swf files
SWFExtract	Extracts images, sounds and movie clips from swf file

કોષ્ટક 5.1 : SWFTOOLSમાં ઉપલબ્ધ પ્રોગ્રામ્સની યાદી

કોષ્ટક 5.1ના પ્રોગ્રામ્સની યાદીનો ઉપયોગ કરીને આપણે સીન્કિંગ વડે gif અને jpeg ફાઇલ બનાવીને એડોબ ફ્લેશ ફાઇલ (swf)માં રૂપાંતર કરી શકીએ છીએ.

સારાંશ

આપણે આ પ્રકરણમાં ચિત્રો કઈ રીતે ઉમેરી શકાય અને તેની સાથે કામ કરી શકીએ તે બાબત શીખ્યાં. અન્ય ઓબ્જેક્ટની જેમ સીન્કિંગમાં બનાવેલાં ચિત્રો ફેરવી શકાય, બદલી શકાય તેમજ તેનું કદ બદલી શકાય છે. આપણે માસ્કિંગના ખ્યાલ વડે ઓબ્જેક્ટને છુપાવી શકીએ તેમજ ઉઘાડા પાડી શકીએ છીએ, તેની ચર્યા પણ કરી. આપણે બ્લેન્ડ મેથડ અને ટાઈમ્ડ સ્વેપ પ્રાયલ દ્વારા ઈમેજિસનો સ્લાઈડ શો બનાવી શકીએ છીએ. જાવા સ્ક્રિપ્ટ અને જાવાનો ઉપયોગ કરીને એનિમેશનમાં ક્રિયા-પ્રતિક્રિયાને ઉમેરી શકીએ છીએ.

સ્વાધ્યાય

1. માસ્કિંગ એટલે શું ?
2. ‘Timed Swap’ વિકલ્પનો ઉપયોગ સમજાવો.
3. એક્શન સ્ક્રિપ્ટ શું છે ? આપણને તે શા માટે જરૂરી છે ?
4. હેતુલક્ષી પ્રશ્નો
 - (1) ઈમેજ આયાત કરવા માટે નીચેનામાંથી કયો કમાન્ડ વપરાય છે ?
 - (a) CTRL + l
 - (b) CTRL + m
 - (c) CTRL + i
 - (d) CTRL + n
 - (2) નીચેનામાંથી કયો રંગ ઈમેજનું કદ બદલવા માટે પોઈન્ટર્સ બતાવે છે ?
 - (a) લીલો
 - (b) પીળો
 - (c) વાદળી
 - (d) લાલ
 - (3) amountની કઈ કિંમત ઈમેજને સંપૂર્ણપણે દૃશ્યમાન સૂચવે છે ?
 - (a) 1
 - (b) 0
 - (c) 2
 - (d) -1
 - (4) એનિમેશન સોફ્ટવેરમાં ક્રિયા-પ્રતિક્રિયાના ભાગ રૂપે ક્રિયાઓ ઉમેરવા નીચેનામાંથી શું વપરાય છે ?
 - (a) એક્શન સ્ક્રિપ્ટ
 - (b) જાવા સ્ક્રિપ્ટ
 - (c) એક્શન પ્રોગ્રામ્સ
 - (d) જાવા એક્શન્સ
 - (5) લેયરના વિસ્તારને છુપાવવા અથવા ઉઘાડા પાડવા માટે નીચેનામાંથી કઈ લાક્ષણિકતા વપરાય છે ?
 - (a) ટૂવીનિંગ (tweening)
 - (b) માસ્કિંગ (masking)
 - (c) મોર્ફિંગ (morphing)
 - (d) એનકેપ્સ્યુલેશન (encapsulation)
 - (6) નીચેનામાંથી કયા પ્રાયલ મારફત ઈમેજની દૃશ્યતાનું સંચાલન કરી શકાય છે ?
 - (a) બ્લેન્ડ, એમાઉન્ટ
 - (b) માસ્ક, એમાઉન્ટ
 - (c) બ્લેન્ડ, ઈમ્પોર્ટ
 - (d) માસ્ક, બ્લેન્ડ

(7) નીચેનામાંથી કઈ બ્લેન્ડ મેથડ ઓબ્જેક્ટને છુપાવવા માટે અને ઉઘાડા પાડવા વપરાય છે ?

- (a) કમ્પોઝિટ (composite) (b) આલ્ફા ઓવર (alpha over)
(c) સ્ટ્રેઈટ ઓન ટુ (straight onto) (d) સ્ટ્રેઈટ (straight)

(8) Timed swap ગુણધર્મ નીચેનામાંથી કયા પેટા પ્રાયલનો ઉમેરો કરે છે ?

- (a) up, down, time, length (b) before, after, size, shape
(c) before, after, time, length (d) top, bottoms, time, length

પ્રાયોગિક પ્રવૃત્તિ

1. 'ખુશ્બૂ ગુજરાત કી' વિષય ઉપર સ્લાઈડ શો બનાવો.
2. 'ભારતનાં ઉત્સવો' વિષય ઉપર સ્લાઈડ શો બનાવો. દરેક ઉત્સવનું નામ પ્રદર્શિત કરવા શાબ્દિક માહિતી વાપરો.
3. આકાશ અને પર્વતો સાથેનું કુદરતી દૃશ્ય દર્શાવતી ઇમેજ ઉમેરો. માસ્કિંગનો ઉપયોગ કરીને ઇમેજમાં સૂર્યોદય બતાવો.





ધોરણ 9માં આપણે ઉબન્ટુ નામની ઓપરેટિંગ સિસ્ટમની કામગીરી વિશે ઊંડાણપૂર્વક અભ્યાસ કર્યો છે. આ અભ્યાસ દરમિયાન આપણે એ શીખ્યા કે ઉબન્ટુ લિનક્સ ધરાવતી કમ્પ્યુટર સિસ્ટમમાં ઉપયોગકર્તા એક વાર લોગ-ઇન થાય એ પછી તે/તેણી કમાન્ડ લાઇન ઇન્ટરફેસ અથવા ગ્રાફિક્સ યુઝર ઇન્ટરફેસનો ઉપયોગ કરીને કમ્પ્યુટર સાથે સંવાદ કરી શકે. આ બંને ઇન્ટરફેસ અગત્યના છે અને તે બંનેના પોતાના ઉપયોગો છે. આ પ્રકરણમાં, કમાન્ડ લાઇન ઇન્ટરફેસનો ઉપયોગ કેવી રીતે કરવો તે વિશે આપણે વિસ્તૃત જાણકારી મેળવીશું. કમાન્ડ લાઇન ઇન્ટરફેસ આપણને ઊંચી કાર્યક્ષમતા સાથે લિનક્સની સાચી શક્તિનો ઉપયોગ કરવા દે છે. તે ઉબન્ટુ સિસ્ટમનો સૌથી વધુ પ્રભાવશાળી ભાગ છે.

ટર્મિનલ ચાલુ કરવું (Starting Up the Terminal)

ગ્રાફિકલ ઇન્ટરફેસમાં કમાન્ડ લાઇન કોન્સોલ (command line console) ખોલવા, લિનક્સમાં ટર્મિનલ (Terminal) નામની વિન્ડો આપવામાં આવી છે. ટર્મિનલ વિન્ડો ખોલવા માટે Applications → Accessories → Terminal ક્લિક કરો અથવા એના બદલે તમે એકસાથે CTRL+ ALT + t કી દબાવી શકો. આમ કરવાથી સ્ક્રીન ઉપર આકૃતિ 6.1માં દર્શાવ્યા મુજબ ટર્મિનલ વિન્ડો દેખાશે. અહીં એ ખાસ નોંધજો કે તમારા સ્ક્રીન પર દેખાતી વિન્ડોના શબ્દો આકૃતિ 6.1માં દર્શાવેલી વિન્ડોના શબ્દો જેવા ના પણ હોય, કારણકે ઉપયોગકર્તા જુદા હોય ને.....!



આકૃતિ 6.1 : ટર્મિનલ વિન્ડો

એક વાર વિન્ડો સ્પષ્ટ રીતે દેખાતી થાય એ પછી આગળ કેટલાક અક્ષર અને કદાચ આંકડા તેમજ ખાસ ચિહ્નો સાથે છેલ્લે \$ અને પછી સતત ઝબૂકતી એક નાની લીટી (જેને કર્સર કહે છે) દર્શાવાય છે. આ બધામાં પ્રથમ શબ્દ તરીકે ઉપયોગકર્તાનું નામ (યુઝરનેમ) હોય છે, અને એના પછી @ ચિહ્ન હોય છે. @ ચિહ્નની પાછળ આપણે જે કમ્પ્યુટરનો ઉપયોગ કરતાં હોઈએ તે કમ્પ્યુટર સિસ્ટમનું નામ દર્શાવાય છે. અંતમાં તમને : ચિહ્ન જોવા મળશે અને એના પછી તમે જે ડિરેક્ટરીમાં કામ કરતા હોય તે ડિરેક્ટરીનું નામ દર્શાવાય છે. (સામાન્ય રીતે તમે તમારી મૂળ (home) ડિરેક્ટરીમાંથી જ કાર્ય શરૂ કરતાં હોવ તેથી તેને ~ ચિહ્ન દ્વારા દર્શાવાય છે.)

કમાન્ડ પ્રોમ્પ્ટ એવું સૂચવે છે કે, ઇન્ટરફેસ હવે ઉપયોગકર્તા સાથે કમાન્ડ સ્વરૂપે સંવાદ કરવા તૈયાર છે. કમાન્ડ એ મૂળભૂત રીતે એક પ્રોગ્રામ હોય છે, આપણું જે કોઈ ચોક્કસ કાર્ય પાર પાડે છે. એક વાર પ્રોમ્પ્ટ દર્શાવાય એટલે આપણે આ પ્રકરણમાં વર્ણવવામાં આવેલ કમાન્ડ ટાઇપ કરી શકીએ. વિવિધ કમાન્ડની ચર્ચા કરીએ એ પહેલાં, ચાલો આપણે ‘શેલ’ (Shell) શબ્દ વિશે આપણું જ્ઞાન ચકાસી લઈએ. શેલ એ કમાન્ડ લાઇન ઇન્ટરફેસ છે. શેલ એ ઉપયોગકર્તા સાથે પરસ્પર સંવાદ કરી શકાય તે માટેનો ઉપયોગકર્તાનો પ્રોગ્રામ છે. તે કમાન્ડ લેંગ્વેજ ઇન્ટરપ્રિટર (Command Language Interpreter) છે, જે કમાન્ડ આપે છે અને સ્વીકારે છે, તેનું અર્થઘટન કરે છે અને આપેલ સૂચના મુજબ અમલ કરવા તેમજ પરિણામ દર્શાવવા તે કર્નલ સાથે પરામર્શ કરે છે. ઉબન્ટુ લિનક્સ સિસ્ટમ ઉપર કામ કરવા અનેક શેલ ઉપલબ્ધ છે, પરંતુ કોઈ એક ચોક્કસ સિસ્ટમ પર ઉપલબ્ધ શેલની સંખ્યા અલગ-અલગ હોઈ શકે.

લિનક્સ સાથે ઉપલબ્ધ કેટલાક પ્રચલિત શેલ છે, બોર્ન (Bourne) શેલ (sh), C શેલ (csh અને tcsh), કોર્ન (Korn) શેલ (ksh) અને બાશ (sh) શેલ. shના ટૂંકા નામ સાથેનો બોર્ન શેલ એ કમાન્ડ લાઈન ઈન્ટરફેસ તરીકે ઉપયોગમાં લેવાતાં યુનિક્સનો શરૂઆતનો શેલ છે. બોર્ન શેલ, શેલસ્ક્રિપ્ટ પ્રોગ્રામિંગ માટેની મૂળ તંત્રવ્યવસ્થા (mechanism) પૂરી પાડે છે, જે આપણને સંપૂર્ણ કમાન્ડ પર આધારિત પ્રોગ્રામ લખવાની છૂટ આપે છે. csh તરીકે ઓળખાતો C શેલ એ લિનક્સ સિસ્ટમ પર સામાન્ય રીતે ઉપલબ્ધ અન્ય શેલ છે. આ શેલમાં પ્રોગ્રામિંગ કરી શકાય છે. cshની નવી આવૃત્તિને tcsh તરીકે ઓળખવામાં આવે છે. cshની મર્યાદાઓને પૂરી કરવા તે વધારાની શેલસ્ક્રિપ્ટ પ્રોગ્રામિંગની સવલતો આપે છે. કોર્ન શેલ અથવા ksh શેલ, sh અને csh બંનેની લાક્ષણિકતાઓને ભેગી કરવા માટે બનાવવામાં આવેલ. Bash શેલ એ બોર્ન શેલની નવી આવૃત્તિ છે. આમ, તે sh જેવાં જ કાર્યો અને વાક્યરચના ધરાવે છે. આજકાલ, bashને લિનક્સ સિસ્ટમ માટે પ્રમાણભૂત (સ્ટાન્ડર્ડ) શેલ ગણવામાં આવે છે અને એટલા માટે તે તમામ લિનક્સ ઓપરેટિંગ સિસ્ટમ ઉપર ઉપલબ્ધ છે અને વપરાય છે.

સિસ્ટમ પર ઉપલબ્ધ શેલની યાદી મેળવવી

તમારી સિસ્ટમમાં ઉપલબ્ધ તમામ શેલ શોધી કાઢવા તમે cat કમાન્ડનો ઉપયોગ કરી શકો. કમાન્ડ પ્રોમ્પ્ટની સામે આકૃતિ 6.2માં દર્શાવ્યા મુજબ કમાન્ડ ટાઈપ કરો. આમ કરવાથી તમને તમારી કમ્પ્યુટર સિસ્ટમમાં ઉપલબ્ધ શેલની યાદી મળશે. તમારી સિસ્ટમની ગોઠવણ અને રૂપરેખા અનુસાર જે શેલની યાદી આઉટપુટ તરીકે તમને દર્શાવાશે તે કદાચ આકૃતિ 6.2માં દર્શાવાયા કરતા જુદી હોઈ શકે છે. cat કમાન્ડની ચર્ચા આ પ્રકરણમાં આગળ કરવામાં આવી છે.

```

File Edit View Terminal Help
administrator@ubuntu:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
administrator@ubuntu:~$

```

આકૃતિ 6.2 : વિવિધ ઉપલબ્ધ શેલ

પૂર્વનિર્ધારિત શેલ નક્કી કરવો

દરેક ‘ઉબન્ટુ લિનક્સ એકાઉન્ટ’ સાથે સામાન્ય રીતે તેના કમાન્ડ લાઈન ઈન્ટરફેસ તરીકે કોઈ એક ચોક્કસ શેલ ગોઠવાયેલ હોય છે. દર વખતે જ્યારે તમે તેમાં લોગ-ઓન થાવ, ત્યારે સિસ્ટમ સાથે કામ કરવા આ પૂર્વનિર્ધારિત શેલને ઉપયોગમાં લેવામાં આવે છે. લિનક્સ ઓપરેટિંગ સિસ્ટમ સાથે કેટલાક એવા ચોક્કસ ચલ હોય છે, જેની કિંમત તરીકે હાલના પર્યાવરણ (environment)ની ગોઠવણ હોય છે અને તેથી જ આવા ચલને પર્યાવરણીય ચલ (environment variables) તરીકે ઓળખવામાં આવે છે.

પૂર્વનિર્ધારિત શેલની કિંમત SHELL નામના આવા એક પર્યાવરણીય ચલમાં સંગ્રહવામાં આવે છે. આમ, SHELL ચલની કિંમત દર્શાવીને આપણે એ જાણી શકીએ છીએ કે આપણો પૂર્વનિર્ધારિત શેલ કયો છે. કોઈ પણ ચલની કિંમત દર્શાવવા *echo* કમાન્ડ ઉપયોગમાં લઈ શકાય છે. કમાન્ડ પ્રોમ્ટની સામે `echo $SHELL` કમાન્ડ ટાઈપ કરીને Enter કી દબાવો. આકૃતિ 6.3માં દર્શાવ્યા મુજબ સ્ક્રીન પર પૂર્વનિર્ધારિત શેલનું નામ દર્શાવાશે. અહીં એ ખાસ નોંધવું જરૂરી છે કે લિનક્સ કમાન્ડ ‘કેસ-સેન્સિટીવ’ છે. (અંગ્રેજીના મોટા અક્ષર (upper case) તેમજ નાના અક્ષર (lower case) પ્રત્યે સંવેદનશીલ હોય, એટલે કે A અને a બંને જુદા ગણે, તો તેને case sensitive કહેવાય) અને તેથી લિનક્સ માટે SHELL, Shell અને shell એ ત્રણેય એકસરખા નથી.

```
File Edit View Terminal Help
administrator@ubuntu:~$ echo $SHELL
/bin/bash
administrator@ubuntu:~$
```

આકૃતિ 6.3 : પૂર્વનિર્ધારિત શેલ

હાલના શેલને બદલવો

અગાઉ ચર્ચા કર્યા મુજબ લિનક્સમાં આપણી પાસે જુદા-જુદા શેલ ઉપલબ્ધ છે. તમારા પૂર્વનિર્ધારિત (default) શેલને બદલવા, તમારે જે શેલનો ઉપયોગ કરવો હોય તેનું નામ કમાન્ડ લાઈન પર ટાઈપ કરો. ઉદાહરણ તરીકે, જો તમારે C શેલનો ઉપયોગ કરવો હોય (તે સિસ્ટમમાં ઉપલબ્ધ હોય તે શરતે), તો કમાન્ડ પ્રોમ્ટની સામે `cs` ટાઈપ કરો. આમ કર્યા પછી કમાન્ડ પ્રોમ્ટ `cs`નો ઈન્ટરફેસ આપશે.

નોંધ :

શેલનો ફેરફાર કામચલાઉ છે અને જ્યાં સુધી તમે તે કમાન્ડ લાઈન પર લોગ-ઓન હોવ ત્યાં સુધી જ તે ચાલુ રહેશે.

પૂર્વનિર્ધારિત શેલમાં પરત આવવા માટે નવા શેલના કમાન્ડ પ્રોમ્ટની સામે `exit` ટાઈપ કરો અથવા `CTRL + d` કી દબાવો.

કમાન્ડની વાક્યરચના (Command Syntax)

લિનક્સ કમાન્ડની વાક્યરચના એકસરખી છે. તેમાં ત્રણ ભાગ હોય છે, જે નીચે દર્શાવેલા ક્રમ અનુસાર હોય છે :

- **નામ (Name) :** આ કમાન્ડનું નામ છે. દા.ત., `echo` વગેરે.
- **વિકલ્પ (Option) :** કમાન્ડની સાથે ઈચ્છિત વધારાના વિકલ્પો લગાડવાથી કમાન્ડની વર્તણૂકમાં બદલાવ લાવવો શક્ય છે. કમાન્ડની સાથે શૂન્ય કે વધુ વિકલ્પો હોઈ શકે છે. વિકલ્પો જ્યારે દર્શાવાય ત્યારે હંમેશા - થી શરૂ થાય અને સામાન્ય રીતે એક અક્ષર અથવા એક આંકડાનો હોય છે. કમાન્ડ કયો છે, તેના આધારે વિકલ્પની સંખ્યા અને અર્થ જુદાં-જુદાં હોઈ શકે.
- **સંલગ્ન મૂલ્ય (આર્ગ્યુમેન્ટ - Arguments) :** વિકલ્પોની સાથે ઉપયોગકર્તા કમાન્ડનું ઈચ્છિત સ્વરૂપે પરિણામ મેળવવા તેની સાથે સંલગ્ન મૂલ્ય (arguments) પણ આપી શકે. કોઈ પણ કમાન્ડ કાર્ય પાર પાડવા માટે શૂન્ય કે વધુ સંલગ્ન મૂલ્યો લઈ શકે. સંલગ્ન મૂલ્યોની સંખ્યા અને તેનો ઈચ્છિત અર્થ દરેક કમાન્ડ માટે જુદો હોય છે. સંલગ્ન મૂલ્યની બાબતમાં કોઈ કમાન્ડ એક પણ સંલગ્ન મૂલ્ય ન લેતા હોય, કેટલાક કમાન્ડ અનુક નિશ્ચિત સંખ્યામાં લેતા હોય જ્યારે કેટલાક કમાન્ડ ગમે તેટલી સંખ્યામાં સંલગ્ન મૂલ્ય લેતા હોઈ શકે.

લિનક્સ કમાન્ડની દ્વિઅંકી (બાઈનરી) ફાઈલ હયાત છે કે કેમ? તેના આધારે લિનક્સના કમાન્ડને આંતરિક (ઈન્ટરનલ) કમાન્ડ અથવા બાહ્ય (એક્સ્ટર્નલ) કમાન્ડ તરીકે વર્ગીકૃત કરી શકાય. એવા કમાન્ડ કે જેની દ્વિઅંકી ફાઈલ અલગ રીતે કાં તો `/sbin`, `/usr/sbin`, `/usr/bin`, `/bin` અથવા `/usr/local/bin` ડિરેક્ટરીમાં સંગ્રહવામાં આવેલી હોય, તો તેને બાહ્ય કમાન્ડ કહે છે. તેનો અમલ સામાન્ય રીતે કર્નેલ (kernel) દ્વારા અમલમાં મુકાય છે અને અમલ દરમિયાન તે એક પ્રક્રિયા-ઓળખ

(process Id) બનાવે છે. લિનક્સમાં મોટા ભાગના કમાન્ડ જે આપણે વાપરીએ છીએ, તે બાહ્ય કમાન્ડ જ છે. બીજી બાજુ એવા કમાન્ડ કે જે સીધા શેલ (shell) દ્વારા અમલમાં મૂકી શકાય છે, તેને આંતરિક કમાન્ડ કહે છે. આંતરિક કમાન્ડ નવી પ્રક્રિયા ઊભી કરતા નથી.

કોઈ પણ કમાન્ડ આંતરિક કમાન્ડ છે કે બાહ્ય કમાન્ડ તે જાણવા માટે આપણે Type કમાન્ડનો ઉપયોગ કરી શકીએ. Type કમાન્ડની વાક્યરચના નીચે મુજબ છે :

Stype command

ઉદાહરણ તરીકે આપણે નીચે મુજબ કમાન્ડનો અમલ કરીએ :

Stype info

આપણને આ કમાન્ડનું પરિણામ નીચે મુજબ મળશે :

info is /usr/bin/info

આપણે જોઈ શકીએ છીએ કે, આ કમાન્ડ દ્વિઅંકી ફાઇલ *info* સ્વરૂપે છે જે /usr/bin/માં સંગ્રહિત છે. એટલે એનો મતલબ એ કે આ કમાન્ડ બાહ્ય કમાન્ડ છે.

સામાન્ય હેતુ માટેના કમાન્ડ આપવા

હવે જ્યારે તમે કમાન્ડની વાક્યરચનાથી પરિચિત થઈ ગયા છો, ત્યારે ચાલો આપણે કમાન્ડ લાઈન ઈન્ટરફેસ મારફત કેવી રીતે કમાન્ડ આપવા તે જોઈએ. લિનક્સ કમાન્ડને શીખવાની શરૂઆત કરવા સામાન્ય હેતુ માટેના કેટલાક કમાન્ડ આપવાનો રસ્તો શ્રેષ્ઠ છે. કમાન્ડ આપવા તમે કમાન્ડનું નામ ટાઈપ કરી તેની પાછળ જરૂરી વિકલ્પો અને સંલગ્ન મૂલ્ય ટાઈપ કરી શકો. બીજો રસ્તો એ છે કે, તમે કમાન્ડના શરૂઆતના અમુક અક્ષરો ટાઈપ કરી Tab કી દબાવો. આમ કરવાથી shell આપમેળે બાકીની માહિતી દર્શાવશે. ઉદાહરણ તરીકે, કેલેન્ડર દર્શાવવા માટે કમાન્ડ પ્રોમ્ટની સામે ca ટાઈપ કરીને Tab કી દબાવો. લિનક્સ આપમેળે ca શબ્દ સાથે શરૂ થતા બધા કમાન્ડની યાદી દર્શાવશે, જેમાં કેલેન્ડર માટેનો કમાન્ડ cal પણ દર્શાવશે. જો યાદીમાં એક કરતાં વધુ કમાન્ડ દર્શાવાય, તો કમાન્ડ પ્રોમ્ટ પર ઈચ્છિત કમાન્ડ ટાઈપ કરી અમલ કરાવવા Enter કી દબાવો.

કેલેન્ડર (cal)

કોઈ ચોક્કસ મહિનાનું અથવા આખા વર્ષનું તારીખિયું (કેલેન્ડર) દર્શાવવા માટે cal કમાન્ડનો ઉપયોગ કરવામાં આવે છે. cal કમાન્ડના પરિણામ રૂપે સામાન્ય રીતે ચાલુ માસનું કેલેન્ડર જોવા મળે છે (જુઓ આકૃતિ 6.4).

```
File Edit View Terminal Help
administrator@ubuntu:~$ cal
  February 2013
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28

administrator@ubuntu:~$
```

આકૃતિ 6.4 : cal કમાન્ડનું પરિણામ

આપણે આપણી જરૂરિયાત મુજબ કેલેન્ડરને બદલી શકીએ છીએ. ઉદાહરણ તરીકે, 2013ના જાન્યુઆરી મહિનાનું કેલેન્ડર દર્શાવવા માટે કમાન્ડ પ્રોમ્પ્ટની સામે નીચે મુજબ કમાન્ડ ટાઇપ કરી Enter કી દબાવો.

Scal 01 2013

આ કમાન્ડનું પરિણામ પણ આકૃતિ 6.4માં દર્શાવેલ પરિણામ જેવું જ હશે, ફરક માત્ર એટલો જ કે મહિનો જાન્યુઆરી અને વર્ષ 2013 હશે. એ જ રીતે જો આપણે 2013ના આખા વર્ષનું કેલેન્ડર દર્શાવવા ઇચ્છતા હોઈએ તો, આપણે નીચે મુજબ cal ટાઇપ કરી તેની પાછળ ફક્ત વર્ષ દર્શાવવું પડે.

Scal 2013

અહીં એ બાબત નોંધો કે, આખા વર્ષનું કેલેન્ડર મોનિટરના સ્ક્રીન પર દર્શાવવું શક્ય નથી, તેથી હવે આપણે નીચે દર્શાવ્યા મુજબ પાઇપ (|) ઓપરેટરનો ઉપયોગ કરવો પડશે.

Scal 2013 | more

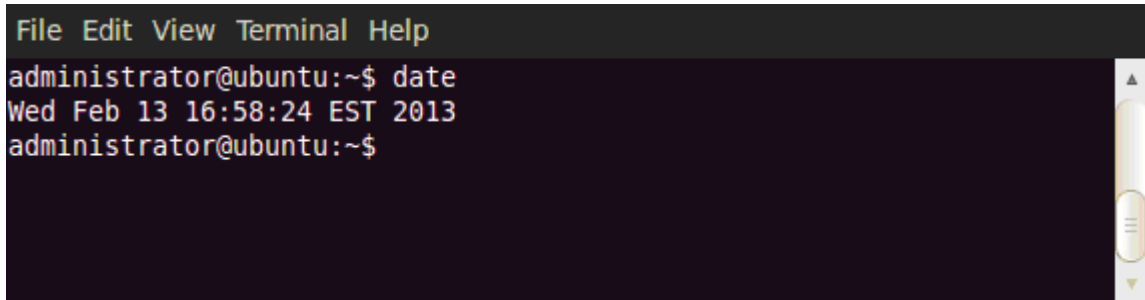
ઉપર દર્શાવેલ કમાન્ડમાં આપણે બે કમાન્ડને ભેગા કરવા પ્રયત્ન કર્યો છે. અહીં more પણ એક કમાન્ડ છે, જે cal નામના કમાન્ડ પાસેથી પરિણામ ઈનપુટ તરીકે મેળવે છે. આ બે કમાન્ડની વચ્ચે દર્શાવેલ પાઇપ ચિહ્ન (|) વિશે વિસ્તૃત ચર્ચા આ પ્રકરણમાં પાછળ કરવામાં આવશે.

તારીખ (date)

અન્ય ઉપયોગી (utility) કમાન્ડ date છે, તેનો ઉપયોગ કમ્પ્યુટર સિસ્ટમની તારીખ દર્શાવવા માટે કરવામાં આવે છે.

\$date

આ કમાન્ડનું પરિણામ આકૃતિ 6.5માં દર્શાવેલ છે.



```
File Edit View Terminal Help
administrator@ubuntu:~$ date
Wed Feb 13 16:58:24 EST 2013
administrator@ubuntu:~$
```

આકૃતિ 6.5 : date કમાન્ડનું પરિણામ

અહીં એ અવલોકન કરો કે દર્શાવાતા પરિણામમાં તારીખ ઉપરાંત સમય એમ બન્ને દર્શાવાય છે. date કમાન્ડની પાછળ આર્ગ્યુમેન્ટ તરીકે ચોક્કસ સ્વરૂપ નિર્દેશકો (format specification) સાથે પણ ઉપયોગમાં લઈ શકાય. દરેક સ્વરૂપની શરૂઆતમાં + ચિહ્ન, એ પછી % ચિહ્ન અને છેલ્લે જે-તે સ્વરૂપ વર્ણવતો એક ચોક્કસ અક્ષર ટાઇપ કરવામાં આવે છે. ઉદાહરણ તરીકે, માત્ર હાલની તારીખને mm/dd/yy સ્વરૂપે દર્શાવવા માટે નીચે દર્શાવ્યા મુજબ કમાન્ડ ટાઇપ કરો.

\$date +%D

આકૃતિ 6.6 કમાન્ડનો આઉટપુટ દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ date +%D
02/13/13
administrator@ubuntu:~$
```

આકૃતિ 6.6 : ઈચ્છિત સ્વરૂપે date કમાન્ડનું પરિણામ

કમાન્ડ લાઈન કેલ્ક્યુલેટર (bc)

લિનક્સમાં bc કમાન્ડ એ કમાન્ડ લાઈન કેલ્ક્યુલેટર છે. સાદી ગાણિતિક ક્રિયાઓ કરવા ઉપરાંત તે કોઈ પણ કિંમતનું જુદી- જુદી સંખ્યાલેખન પદ્ધતિઓ વચ્ચે રૂપાંતર પણ કરી શકે છે, તદ્ઉપરાંત તે આપણને કેટલીક વૈજ્ઞાનિક ગણતરીઓ પણ કરવા દે છે. આ કમાન્ડનો ઉપયોગ કરવા માટે નીચે આપ્યા મુજબની વાક્યરચનાનો ઉપયોગ કરો :

\$bc -l

આમ કરવાથી આકૃતિ 6.7માં દર્શાવ્યા મુજબનો સ્ક્રીન દર્શાવવામાં આવશે. અહીં એ નોંધ લો કે \$ પ્રોમ્પ્ટ સ્ક્રીન પર અદૃશ્ય થઈ ગયો છે; તે એવું દર્શાવે છે કે, bc કમાન્ડ હવે તમારી પાસેથી ઈનપુટ મેળવવા તૈયાર છે. -l સ્વિચ (switch)નો ઉપયોગ પ્રમાણભૂત ગણિત લાઈબ્રેરી (standard math library) સામેલ કરવા માટે થાય છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ bc -l
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Found
ation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
```

આકૃતિ 6.7 : bc કમાન્ડની શરૂઆત

હવે, તમારે જેની ગણતરી કરવી હોય તે સૂત્ર (formula) ને ઝબૂકતી લીટી (કર્સર) સામે ટાઈપ કરો અને પછી Enter કી દબાવો. તમે $5 * 5$, જેવી સીધી પદાવલિ ટાઈપ કરી શકો અથવા ઘણા બધા પ્રક્રિયકો સાથેની જટીલ પદાવલિ પણ ટાઈપ કરી શકો. તો ચાલો $((99.1 / 5.15) * 99.9)$, પદાવલિ ટાઈપ કરીને, Enter કી દબાવીએ. આમ કરવાથી આ કમાન્ડ આકૃતિ 6.8માં દર્શાવ્યા મુજબ પછીની હરોળ પર પરિણામ દર્શાવશે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ bc -l
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Found
ation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
((99.1/5/15)*99.9)
132.0011999999999999999996
```

આકૃતિ 6.8 : bc કમાન્ડનું પરિણામ

અહીં એ નોંધ લો કે, હજુ પણ આપણે સ્ક્રીન પર પ્રોમ્પ્ટ જોઈ શકતા નથી. એનો મતલબ એ જ કે, હજુ પણ આપણે કમાન્ડ લાઈન કેલ્ક્યુલેટર સાથે કામ કરી શકીએ છીએ.

સામાન્ય ગાણિતિક ક્રિયાઓ જેવી કે, સરવાળા, બાદબાકી, ગુણાકાર, ભાગાકાર, વગેરે ઉપરાંત આપણે ત્રિકોણમિતી અથવા લઘુગુણક વિધેયો જેવાં કે, સાઈન (sine), કોસાઈન (cosine), આર્કટેન્જન્ટ (arctangent) અને લોગ (log)નો પણ ઉપયોગ કરી શકીએ. ઉદાહરણ તરીકે, જો તમારે કિંમત 2013નો કુદરતી લઘુગુણક (Natural Logarithm) શોધવાની જરૂર પડે, તો કમાન્ડ તરીકે **l(2013)** ટાઈપ કરો અને તમને પરિણામ તરીકે **7.60738142563979148420** મળશે.

ibase વિધેય આપણને ઈનપુટ માટે જે સંખ્યાલેખન પદ્ધતિ ઉપયોગમાં લેવા ઈચ્છતા હોઈએ, તેને ગોઠવવા દે છે. એ જ રીતે કઈ સંખ્યાલેખન પદ્ધતિમાં પરિણામ દર્શાવાય તે ગોઠવવા *obase* વિધેય ઉપયોગમાં લઈ શકાય છે. તો ચાલો, દશાંશપદ્ધતિમાં ઈનપુટ કરેલી કિંમતને સોળ-અંકી પદ્ધતિ (Hexadecimal System)માં પરિવર્તિત કરીને દર્શાવવા પ્રયત્ન કરીએ. સૌપ્રથમ આપણે નીચે દર્શાવ્યા મુજબ *obase*ની ગોઠવણી કરવી પડશે.

obase = 16

હવે, ઉદાહરણમાં દર્શાવ્યા પ્રમાણે તમારે જે દશાંશપદ્ધતિની સંખ્યાને સોળ-અંકી પદ્ધતિમાં ફેરવવી હોય તે સંખ્યા ટાઈપ કરી Enter કી દબાવો.

256

100

અહીં 100 એ દશાંશપદ્ધતિની સંખ્યા 256 જેટલું જ મૂલ્ય ધરાવતી સોળ-અંકી પદ્ધતિ મુજબની સંખ્યા છે.

એ જ રીતે, જો તમે આ જ દશાંશકિંમતને દ્વિઅંકી પદ્ધતિની સંખ્યામાં રૂપાંતર કરવા ઈચ્છતા હોવ, તો ફરીથી નીચે દર્શાવ્યા પ્રમાણે માત્ર *obase* બદલી દો.

obase=2

હવે **100** ટાઈપ કરો. આમ કરવાથી પરિણામ 1100100, આવશે. અહીં એ અવલોકન કરો કે આ દ્વિઅંકી રકમ દશાંશ-કિંમત 100 માટેની છે, નહીં કે 256 માટેની. આ માટેનું કારણ એકદમ સરળ છે, આપણે *ibase*ની ગોઠવણ બદલી નથી અને માટે તમામ ઈનપુટ કરાતી કિંમત દશાંશકિંમત જ ગણવામાં આવે છે.

સોળ-અંકી પદ્ધતિની 100 સંખ્યાને તેની સમમૂલ્ય દ્વિઅંકી પદ્ધતિની સંખ્યામાં રૂપાંતરિત કરવા નીચે મુજબ ibase અને obaseની ગોઠવણ બદલો.

ibase=16

obase=2

હવે તમે એવી સોળ-અંકી સંખ્યાની કિંમત ટાઈપ કરો, જેનું તમે દ્વિઅંકી પદ્ધતિની સંખ્યામાં રૂપાંતરિત કરવા ઇચ્છતા હોવ. ઉદાહરણ તરીકે, **100** ટાઈપ કરો અને તમને પરિણામ રૂપે **100000000** મળશે.

દશાંશપદ્ધતિમાં પુનઃ પાછા ફરવા ibase તરીકે 10 ગોઠવી દો. math લાઈબ્રેરીમાં ઉપલબ્ધ sqrt વિધેયનો ઉપયોગ કરી આપેલ સંખ્યાનું વર્ગમૂળ શોધવા નીચે મુજબના કમાન્ડનો અમલ કરો.

sqrt(256)

16.000000000000000000000000

ગ્રાફિક્સ કેલ્ક્યુલેટરની સરખામણીએ ‘કમાન્ડ લાઈન bc કેલ્ક્યુલેટર’ વધુ ઝડપી અને લવચિક છે. લિનક્સના કમાન્ડ પ્રોમ્પ્ટ પર પરત આવવા CTRL+ d કી દબાવો.

સંદેશો દર્શાવવો (echo) :

કમાન્ડ પ્રોમ્પ્ટનો ઉપયોગ કરતી વખતે આપણને વારંવાર કોઈ ને કોઈ સંદેશો દર્શાવવાની જરૂર પડે છે. ટર્મિનલ ઉપર સંદેશો દર્શાવવા માટે echo કમાન્ડ ઉપયોગમાં લેવાય છે. ઉદાહરણ તરીકે, નીચે મુજબ કમાન્ડ ટાઈપ કરી Enter કી દબાવો. echo કમાન્ડ પછી લખવામાં આવેલ શબ્દોનો સમૂહ તમારા મોનિટરના સ્ક્રીન પર દર્શાવાશે.

Secho Hi, I am learning Ubuntu Linux

Hi, I am learning Ubuntu Linux

આ શબ્દોના સમૂહને અવતરણચિહ્નોમાં આવરી લેવો પણ શક્ય છે. પરિણામમાં આવાં અવતરણચિહ્ન સમાવાશે નહીં. કોઈ ચલની કિંમત દર્શાવવા માટે પણ echo કમાન્ડનો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે, cost નામનો એક ચલ તૈયાર કરો અને નીચે દર્શાવ્યા મુજબ તેની કિંમત તરીકે 10 આપો.

\$cost=10

એક વાર તમે Enter કી દબાવશો કે તરત જ કર્સર પાછું કમાન્ડ પ્રોમ્પ્ટ પર આવી જશે. હવે નીચે દર્શાવ્યા મુજબ કમાન્ડ ટાઈપ કરો :

Secho The cost of product is Rs. \$cost

The cost of product is Rs. 10

costની કિંમત સ્ક્રીન પર દર્શાવવા માટે તેને echo કમાન્ડને પહોંચાડવામાં આવી. અહીં એ નોંધો કે કમાન્ડમાં આપણે બે વાર cost શબ્દ ટાઈપ કર્યો છે. એમાં પહેલી વાર ટાઈપ કરેલ cost શબ્દ સીધીસાદી રીતે લખાયો છે, જે એક સામાન્ય શબ્દ જ છે. જ્યારે એ પછી આગળ બીજી વખત લખાયેલ cost શબ્દની આગળ \$ ચિહ્ન મૂકવામાં આવેલ

છે. \$ શબ્દથી શરૂ થતા શબ્દને ચલનું નામ સમજવામાં આવે છે. આમ, echo કમાન્ડને જ્યાં \$થી શરૂ થતો કોઈ શબ્દ મળશે તો તે તેને ચલનું નામ સમજશે અને તેથી તે શબ્દને બદલે તે ચલની કિંમત જાણીને તે છાપશે.

અર્થપૂર્ણ પરિણામ દર્શાવવા માટે echo કમાન્ડની સાથે અન્ય કમાન્ડનો ઉપયોગ પણ કરી શકાય છે. ઉદાહરણ તરીકે આપણે નીચે દર્શાવ્યા મુજબ echo કમાન્ડની સાથે date કમાન્ડનો પણ ઉપયોગ કરી શકીએ.

\$echo Current time is 'date +%T'

Current time is 14 :55 :04

અહીં એ જોઈ શકાય છે કે, જે કમાન્ડને echo કમાન્ડના ભાગ તરીકે ઉપયોગમાં લેવો છે તે કમાન્ડને અવળા અવતરણ-ચિહ્ન (back quote ' ') (~ ચિહ્નવાળી કી પર ઉપલબ્ધ અવતરણચિહ્ન)માં આવરીને દર્શાવવામાં આવે છે. આમ, પહેલાં date કમાન્ડનો અમલ થશે અને એ પછી તેના પરિણામની સાથે echo કમાન્ડનો અમલ કરવામાં આવશે.

પાસવર્ડ બદલવો (passwd)

વિવિધ કારણોસર ઉપયોગકર્તાને અવારનવાર પાસવર્ડ બદલવાની જરૂર પડે છે. આ કાર્ય પાર પાડવા માટે passwd કમાન્ડ આપણને મદદરૂપ નીવડે છે. તેના દ્વારા સામાન્ય રીતે તાજેતરના લોગ-ઈન ખાતા (ઉપયોગકર્તા)નો જ પાસવર્ડ બદલી શકાય છે. નીચે મુજબ કમાન્ડ ટાઈપ કરવાથી તે આપણને આપણો પાસવર્ડ બદલવામાં મદદરૂપ બનશે.

\$passwd

એક વાર આપણે જેવી Enter કી દબાવીએ કે તરત જ સ્ક્રીન ઉપર કર્સર સાથે નીચે દર્શાવ્યા મુજબનો એક સંદેશો દર્શાવાશે.

Changing password for administrator

(current) UNIX password :

તમારો અત્યારનો પાસવર્ડ ટાઈપ કરી Enter કી દબાવો. લિનક્સ એ ચકાસણી કરશે કે તમે ટાઈપ કરેલો પાસવર્ડ સાચો છે કે કેમ? અને જો સાચો હોય તો તે નવો પાસવર્ડ ટાઈપ કરવા જણાવશે. નીચે દર્શાવ્યા મુજબ તમને નવો પાસવર્ડ ટાઈપ કરવા અને ફરીથી એનો એ જ પાસવર્ડ ટાઈપ (retype) કરવા જણાવવામાં આવશે :

Enter new UNIX password :

Retype new UNIX password :

passwd : password updated successfully

જો તમે નવો પાસવર્ડ સચોટ રીતે ટાઈપ કર્યો હશે અને જો તે પાસવર્ડ માટે નિયત કરેલ માર્ગદર્શિકાઓ મુજબ અનુકૂળ હોય, તો સિસ્ટમ દ્વારા તમારો નવો પાસવર્ડ નોંધી લેવામાં આવશે. જો આ બાબતે કોઈ પણ પ્રતિકૂળતા હશે, તો તરત જ તમને તે બાબતનો ભૂલસંદેશ દર્શાવવામાં આવશે. એડમિનિસ્ટ્રેટર દ્વારા સિસ્ટમના કોઈ પણ અન્ય ઉપયોગકર્તાનો પાસવર્ડ બદલવો પણ શક્ય છે, એ માટે passwd કમાન્ડની પાછળ તે ઉપયોગકર્તાનું નામ (username) ટાઈપ કરવું પડે. ઉદાહરણ તરીકે, જો આપણી પાસે username તરીકે harshal, હોય તો તેમનો પાસવર્ડ બદલવા આપણે નીચે મુજબ કમાન્ડ આપવો પડે.

\$passwd harshal

વાસ્તવિક ઉપયોગકર્તાના કિસ્સામાં તમને પાસવર્ડ બદલવા દેવામાં આવશે.

સ્ક્રીન સાફ કરવો (clear)

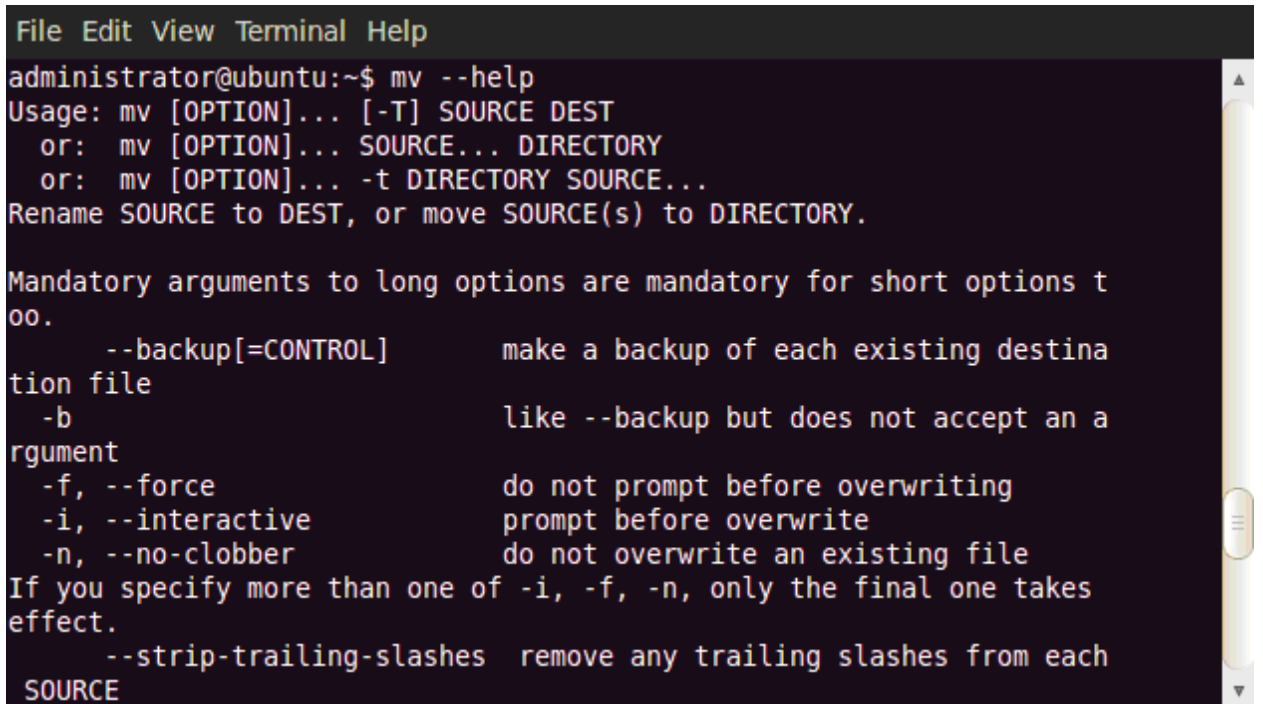
કમાન્ડ પ્રોમ્પ્ટ પર કામ કરતી વખતે તમે એ અવલોકન કર્યું હશે કે, વારંવાર સ્ક્રીન અક્ષરો અને આંકડાઓથી ભરચક થઈ જાય છે. આવા કિસ્સામાં પરિણામને સ્પષ્ટ રીતે નિહાળવામાં તકલીફ પડે છે. આ સમસ્યા માટે આપણી પાસે એક સરળ ઉપાય છે. આ માટે સ્ક્રીન પર દર્શાવેલ લખાણને ભૂંસવા માટે *clear* કમાન્ડનો ઉપયોગ કરી શકાય.

Sclear

લિનક્સ કમાન્ડ વિશે મદદ મેળવવી

બીજા કમાન્ડ વિશે જાણીએ એ પહેલાં, ચાલો સૌપ્રથમ લિનક્સ પ્લેટફોર્મ પર કમાન્ડનો ઉપયોગ કરતી વખતે કેવી રીતે મદદ મેળવવી તે શીખીએ. ઉપયોગકર્તા જ્યારે કમાન્ડ લાઈન ઈન્ટરફેસ દ્વારા કાર્ય કરતો હોય, ત્યારે તેને મદદરૂપ થવા માટે લિનક્સમાં *help* અને *man* નામના બે આંતરપ્રસ્થાપિત કમાન્ડ આપવામાં આવ્યા છે.

લિનક્સમાં આપણે જેનો ઉપયોગ કરીએ છીએ, તે બધા *-h* (*-help*) વિકલ્પને સમર્થન આપે છે. આ વિકલ્પ, કમાન્ડ કઈ રીતે વાપરવો તે બાબતે ટૂંકું વર્ણન દર્શાવે છે. આકૃતિ 6.9 *help* કમાન્ડનો ઉપયોગ દર્શાવે છે.



```
File Edit View Terminal Help
administrator@ubuntu:~$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
  or: mv [OPTION]... SOURCE... DIRECTORY
  or: mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  --backup[=CONTROL]  make a backup of each existing destination file
  -b                  like --backup but does not accept an argument
  -f, --force         do not prompt before overwriting
  -i, --interactive   prompt before overwrite
  -n, --no-clobber    do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes effect.
  --strip-trailing-slashes  remove any trailing slashes from each SOURCE
```

આકૃતિ 6.9 : *help* કમાન્ડનો ઉપયોગ

અહીં એ નોંધો કે, આકૃતિ 6.9માં મદદ દર્શાવવા માટે ઉપયોગમાં લેવાયેલ કમાન્ડ *mv --help* છે. કમાન્ડ વિશે મદદ મેળવવા માટેનો અન્ય વૈકલ્પિક વ્યવસ્થા એ લિનક્સ ઓનલાઈન મેન્યુઅલનો ઉપયોગ કરવાની છે. આપણે જે કમાન્ડ વિશે જાણવું હોય તે સંબંધિત ચોક્કસ કમાન્ડ માટે મેન્યુઅલ સક્રિય કરવાનું કામ *man* કમાન્ડ કરે છે. ઉદાહરણ તરીકે *man mv* કમાન્ડ આપણને *mv* કમાન્ડની મદદ માટે મેન્યુઅલ દર્શાવશે. આકૃતિ 6.10 *man*નું પરિણામ દર્શાવે છે.

અહીં એ નોંધ લેવી જોઈએ કે, *man* કમાન્ડ આપણને કોઈ પણ કમાન્ડ વિશેની સંપૂર્ણ માહિતી દર્શાવે છે અને તેથી કદાચ એક કરતાં વધુ સ્ક્રીન જેટલી થઈ શકે છે. તે સામાન્ય રીતે એક સમયે માત્ર એક જ સ્ક્રીન દર્શાવે છે. એ પછી જ્યારે આપણે Enter કી દબાવીએ એ પછી સ્ક્રીન બદલાય છે. *manual* કમાન્ડને પૂર્ણ કરી સ્ક્રીનમાંથી બહાર નીકળવા 'q', કી દબાવો. આમ કરવાથી તમને કમાન્ડ પ્રોમ્પ્ટ પર પાછા લઈ આવશે.


```
File Edit View Terminal Help
MV(1) User Commands MV(1)
NAME
  mv - move (rename) files
SYNOPSIS
  mv [OPTION]... [-T] SOURCE DEST
  mv [OPTION]... SOURCE... DIRECTORY
  mv [OPTION]... -t DIRECTORY SOURCE...
DESCRIPTION
  Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

  Mandatory arguments to long options are mandatory for short options
  too.

  --backup[=CONTROL]
                make a backup of each existing destination file
Manual page mv(1) line 1
```

આકૃતિ 6.10 : man કમાન્ડનો ઉપયોગ

જો આપણે કમાન્ડનું માત્ર ટૂંકું વર્ણન જોઈતું હોય તો તેવા કિસ્સામાં આપણે *whatis* કમાન્ડનો ઉપયોગ કરી શકીએ. તે આપણને કમાન્ડનું એક લાઇનનું વર્ણન આપે છે. પરંતુ વિકલ્પ બાબતે કોઈ પણ વધારાની માહિતી બાકાત રાખે છે. આકૃતિ 6.11 *whatis* કમાન્ડનું નમૂનારૂપ પરિણામ દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ whatis mv
mv (1) - move (rename) files
administrator@ubuntu:~$
```

આકૃતિ 6.11 : *whatis* કમાન્ડનું નમૂનારૂપ પરિણામ

ઘણીવાર કદાચ એવું પણ બને કે, આપણે ચોક્કસ કયા કમાન્ડ વિશે જોવું તે જ ન જાણતા હોઈએ, આવી પરિસ્થિતિમાં આપણે *apropos* કમાન્ડનો ઉપયોગ કરી શકીએ. કમાન્ડની વાક્યરચના નીચે દર્શાવ્યા મુજબ છે :

Sapropos string

આપણે જ્યારે આ કમાન્ડનો અમલ કરીશું, ત્યારે આપણને એવા બધા કમાન્ડની યાદી મળશે, જેના કમાન્ડ કે કમાન્ડના વર્ણનમાં *string* શબ્દ આવતો હોય છે. ઉદાહરણ તરીકે, જો આપણે કમાન્ડ પ્રોમ્પ્ટ પર *apropos copy* ટાઇપ કરીને અમલ કરવા કોશિશ કરીશું, તો જે કમાન્ડમાં કે તેના વર્ણનમાં *copy* શબ્દ આવતો હશે તેવા કમાન્ડની યાદીથી આપણો સ્ક્રીન ભરાઈ જશે. આ કમાન્ડ વાપરતી વખતે ઉપયોગકર્તાએ કાળજી લેવી જોઈએ. *apropos* કમાન્ડનું નમૂનારૂપ પરિણામ આકૃતિ 6.12માં દર્શાવેલ છે.

```

File Edit View Terminal Help
administrator@ubuntu:~$ apropos copy
bcopy (3)          - copy byte sequence
bf_copy (1)       - shell script to copy a bogofilter working...
bf_copy-bdb (1)   - shell script to copy a bogofilter working...
copysign (3)      - copy sign of a number
copysignf (3)     - copy sign of a number
copysignl (3)     - copy sign of a number
cp (1)            - copy files and directories
cpgr (8)          - copy with locking the given file to the p...
cpio (1)          - copy files to and from archives
cppw (8)          - copy with locking the given file to the p...
dd (1)            - convert and copy a file

```

આકૃતિ 6.12 : apropos કમાન્ડનું નમૂનારૂપ પરિણામ

ડિરેક્ટરી સાથે કાર્ય કરવું

લિનક્સમાં ડિરેક્ટરી એ એક એવી વિશિષ્ટ ફાઇલ છે, જે વિવિધ ફાઇલો અને અન્ય ડિરેક્ટરીઓને સંગ્રહવા માટે ઉપયોગમાં લેવાય છે. અહીં ‘ / ’ ચિહ્ન મૂળ (root) ડિરેક્ટરીનો નિર્દેશ કરે છે. બાકીની બધી ડિરેક્ટરી મૂળ ડિરેક્ટરીમાં આવેલી હોય છે, તો ચાલો, ટર્મિનલ વિન્ડોનો ઉપયોગ કરીને ડિરેક્ટરી સાથે કેવી રીતે કામ કરી શકાય તે શીખીએ.

હોમ ડિરેક્ટરી (Home Directory)

ઉપયોગકર્તા જ્યારે સિસ્ટમમાં લોગ-ઓન થાય છે, ત્યારે લિનક્સ આપમેળે ઉપયોગકર્તાને હોમ ડિરેક્ટરીના નામે ઓળખાતી ડિરેક્ટરીમાં મૂકે છે. તે જ્યારે ઉપયોગકર્તાનું ખાતું બનાવવામાં આવે છે તે સમયે તૈયાર કરવામાં આવે છે અને સામાન્ય રીતે તેનો પથ (path) /home/username હોય છે. અહીંયાં username એટલે લોગ-ઇન નામ. ઉદાહરણ તરીકે, ધારો કે યુઝરનેમ harshal હોય, તો /home/harshal એ હોમ ડિરેક્ટરી હશે. જરૂર પડે તો આ પથ બદલી શકાય છે. સામાન્ય રીતે વર્કિંગ ડિરેક્ટરી પથ HOME નામના સિસ્ટમ ચલમાં સંગ્રહાયેલ હોય છે. આપણે echo કમાન્ડનો ઉપયોગ કરીને ડિરેક્ટરીની ઊલટતપાસ કરી શકીએ.

Secho \$HOME

/home/harshal

અહીં એ નોંધ લો કે, આ કમાન્ડના ઉપયોગ દ્વારા દર્શાવાતો પથ એ સંપૂર્ણ પથનામ છે. સંપૂર્ણ પથનામ એ / (સ્લેશ) ચિહ્ન વડે છૂટી પાડીને દર્શાવાતી ડિરેક્ટરીઓનાં નામની હારમાળા છે. સંપૂર્ણ પથનામ એ મૂળ (રૂટ) ડિરેક્ટરીના સંદર્ભમાં સ્થાન દર્શાવે છે. પહેલાં સ્લેશ એ મૂળ ડિરેક્ટરીના નામનો નિર્દેશ કરે છે જ્યારે બાકીના સ્લેશ એ બે ડિરેક્ટરીનાં નામ વચ્ચે ચિહ્ન તરીકે કાર્ય કરે છે. આમ, harshal નામની ડિરેક્ટરી home ડિરેક્ટરીમાં આવેલી છે અને home ડિરેક્ટરી મૂળ (રૂટ) ડિરેક્ટરીમાં આવેલી છે. એ જ રીતે /home/administrator એ administrator નામના યુઝરનેમ માટેની હોમ ડિરેક્ટરી છે.

હોમ ડિરેક્ટરીનો પથ જો બદલવો હોય, તો તે બદલી શકીએ છીએ. ધારોકે, તમે નીચે દર્શાવ્યા મુજબનું પરિણામ જોઈ શકો છો :

/home/its/ug1/svics

અહીં svics (ઉપયોગકર્તાની હોમડિરેક્ટરી) ug1 (એક ડિરેક્ટરી કે જે પેટાજૂથ રજૂ કરે છે.) નામની પેટાડિરેક્ટરીમાં છે, જે વળી પાછું its (એક ડિરેક્ટરી કે જે જૂથ રજૂ કરે છે) નામની ડિરેક્ટરીમાં છે. home ડિરેક્ટરી અને ‘/’ ને તેનો પોતાનો પૂર્વનિર્ધારિત અર્થ છે.

હાલના કામની ડિરેક્ટરી (*pwd*)

સિસ્ટમમાં આપણે લોગ થઈએ એ પછી આપણે જરૂરિયાત અનુસાર એક ડિરેક્ટરીમાંથી બીજી ડિરેક્ટરીમાં આવન-જાવન કરી શકીએ છીએ પણ કોઈ એક સમયે આપણે માત્ર જે-તે એક જ ડિરેક્ટરીમાં હોઈ શકીએ. કોઈ ચોક્કસ ક્ષણે આપણે જે ડિરેક્ટરીમાં હોઈએ તે ડિરેક્ટરીને કરન્ટ ડિરેક્ટરી (હાલની ડિરેક્ટરી) અથવા પ્રેઝન્ટ વર્કિંગ ડિરેક્ટરી (હાલના કામની ડિરેક્ટરી) તરીકે ઓળખવામાં આવે છે. અત્યારે હાલ આપણે કઈ ડિરેક્ટરીમાં કાર્ય કરી રહ્યા છીએ તે કરન્ટ ડિરેક્ટરી જાણવા આપણે *pwd* કમાન્ડનો ઉપયોગ કરી શકીએ.

\$pwd

નવી ડિરેક્ટરી બનાવવી (*mkdir*)

લિનક્સમાં *mkdir* કમાન્ડનો ઉપયોગ કરીને આપણે નવી ડિરેક્ટરી બનાવી શકીએ છીએ. જે નામની ડિરેક્ટરી બનાવવી હોય, તે નામ આ કમાન્ડની પાછળ આર્ગ્યુમેન્ટ તરીકે દર્શાવવું પડે. તો ચાલો, *subject* નામની નવી ડિરેક્ટરી બનાવીએ.

\$mkdir subject

કમાન્ડ લાઈન ઈન્ટરફેસની સામે GUIની શક્તિ એ તેની લવચિકતા (flexibility)માં રહેલી છે. જો આપણે GUIનો ઉપયોગ કરીને ડિરેક્ટરી બનાવીશું, તો એક સમયે આપણે માત્ર એક જ ડિરેક્ટરી બનાવી શકીએ, જ્યારે કમાન્ડ પ્રોમ્પ્ટ પર એક જ *mkdir* કમાન્ડ આપીને એકસાથે અનેક ડિરેક્ટરી બનાવવી શક્ય છે. કમાન્ડની નીચેની વાક્યરચના એ જ બાબત દર્શાવે છે.

\$mkdir animals birds vehicles plants

આ કમાન્ડનો જ્યારે અમલ કરવામાં આવશે, ત્યારે તે કરન્ટ ડિરેક્ટરીમાં *animals*, *birds*, *vehicles* અને *plants* નામની ચાર ડિરેક્ટરી બનાવશે.

ડિરેક્ટરી બદલવી (*cd*) :

જ્યારે આપણે કોઈ પણ ડેટાને કોઈ ચોક્કસ ડિરેક્ટરીની અંદર સંગ્રહવાની જરૂર પડે, ત્યારે તેવા કિસ્સામાં સૌથી પહેલાં આપણે આવી ચોક્કસ ડિરેક્ટરીને કરન્ટ ડિરેક્ટરી બનાવવી પડે. *cd* કમાન્ડનો ઉપયોગ કરીને આપણે કોઈ ચોક્કસ ડિરેક્ટરીમાં જઈ શકીએ (બદલી શકીએ). તો ચાલો, *subject* નામની ડિરેક્ટરીમાં *math*, *science* અને *economics* નામની ડિરેક્ટરી બનાવવાનો પ્રયત્ન કરીએ. આ બધી ડિરેક્ટરીઓ બનાવવા માટે સૌપ્રથમ આપણું *subject* ડિરેક્ટરીમાં હોવું જરૂરી છે. નીચે આપેલી કમાન્ડની હારમાળાથી આ ક્રિયા થઈ જશે.

\$cd subject

\$pwd

/home/administrator/subject

\$mkdir math science economics

એ નોંધ લો કે, અહીં યુઝરનેમ *administrator* છે. ફરીથી *administrator* ડિરેક્ટરીમાં પાછા જવા માટે માત્ર નીચે મુજબનો કમાન્ડ ટાઈપ કરો :

\$cd ..

ઉપર્યુક્ત કમાન્ડમાં બે ટપકાં (..) પેરન્ટ ડિરેક્ટરી (parent directory)નો નિર્દેશ કરે છે. અહીં એ ખાસ નોંધ કરો કે cd કમાન્ડ અને .. ની વચ્ચે એક અક્ષર જેટલી જગ્યા રાખવી જરૂરી છે.

ધારો કે તમે કોઈ આંતરિક ડિરેક્ટરીમાં હોવ કે જેનો પથ /home/administrator/subject/economics હોય અને તમારે ઉપયોગકર્તાની હોમ ડિરેક્ટરીમાં આવવાની જરૂર હોય, તો એ માટે ફરી cd કમાન્ડ ઉપયોગી બને છે. હોમ ડિરેક્ટરીમાં પાછા આવવા માટે આપણે નીચે દર્શાવ્યા મુજબ બે વાર કમાન્ડ આપી શકીએ :

Scd ..

Scd ..

અહીં આ કમાન્ડ દર વખતે આપણને એક સ્તર નીચે લઈ આવશે. જો આપણે એમ કહીએ કે આપણે M સ્તર સુધી ડિરેક્ટરીમાં અંદર ગયા હોઈશું, તો આપણે M વખત cd કમાન્ડનો ઉપયોગ કરવો પડશે. એના બદલે cd કમાન્ડનો ઉપયોગ એક જ વાર કરવા માટેનો વૈકલ્પિક ઉપાય નીચે મુજબ છે :

Scd ../..

cd કમાન્ડના ઉપયોગના વર્ણન સાથેનાં કેટલાંક ઉદાહરણ કોષ્ટક 6.1માં આપવામાં આવેલ છે :

આપવામાં આવતો કમાન્ડ	કરવામાં આવતું કાર્ય
cd ~/Desktop	કોઈ પણ કરન્ટ પથમાંથી /home/username/Desktop, ડિરેક્ટરી બદલે છે. અહીં ~ ચિહ્ન ઉપયોગકર્તાની હોમ ડિરેક્ટરી દર્શાવે છે.
cd /	હાલના પથ પરથી ડિરેક્ટરી બદલીને રૂટ-ડિરેક્ટરીમાં જશે.
cd	હાલના પથ પરથી હોમ ડિરેક્ટરીમાં ડિરેક્ટરી બદલે છે.
cd -	અગાઉ બદલેલી ડિરેક્ટરીમાં ડિરેક્ટરી બદલે છે.
cd /var/www	ડિરેક્ટરી બદલીને સીધા var ડિરેક્ટરીની અંદર આવેલી www ડિરેક્ટરીમાં પહોંચે છે. જ્યારે આપણને આખો પથ ખબર હોય ત્યારે આ ખૂબ જ ઉપયોગી છે.

કોષ્ટક 6.1 : cd કમાન્ડનાં કેટલાંક ઉદાહરણ

ડિરેક્ટરી કાઢી નાંખવી (rmdir)

કોઈ પણ ખાલી ડિરેક્ટરીને rmdir કમાન્ડનો ઉપયોગ કરીને કાઢી નાંખી શકાય છે.

Srmdir science

અહીં science એ ડિરેક્ટરીનું નામ છે અને જો તે ખાલી હશે તો આ કમાન્ડથી તે નીકળી જશે. જો તે ખાલી નહીં હોય, તો આપણને ‘rmdir : failed to remove ‘science’ : Directory not empty’. એવો ભૂલસંદેશ દર્શાવશે. આ કિસ્સામાં આપણે ડિરેક્ટરીમાં પડેલું બધું જ સાહિત્ય કાઢી નાંખવું પડશે અને પછી ફરીથી કમાન્ડ આપવો પડશે. અહીં એ ખાસ નોંધો, કે જે રીતે આપણે બનાવી હતી, એ જ રીતે એકસાથે અનેક ખાલી ડિરેક્ટરી કાઢી નાંખવી શક્ય છે.

ખાલી ન હોય તેવી ડિરેક્ટરી તેના સંપૂર્ણ સાહિત્ય સાથે કાઢી નાંખવા માટે આપણે નીચે દર્શાવ્યા મુજબ *rm* કમાન્ડનો ઉપયોગ કરી શકીએ છીએ.

Srm -r science

આ પ્રકરણના પાછળના ભાગે *rm* કમાન્ડની વિસ્તૃત ચર્ચા કરવામાં આવી છે.

લિનક્સમાં નામ આપવાની પ્રણાલિકા

mkdir કમાન્ડનો ઉપયોગ કરીને આપણે જુદાં-જુદાં નામની ડિરેક્ટરીઓ બનાવી. લિનક્સમાં કોઈ પણ ફાઈલ કે ડિરેક્ટરી બનાવતી વખતે આપણે કેટલાક ચોક્કસ નિયમો અનુસરવા પડે. આજની મોટા ભાગની લિનક્સ સિસ્ટમમાં ફાઈલ કે ડિરેક્ટરીનું નામ વધુમાં વધુ 255 અક્ષરોનું હોઈ શકે. Windows OS કરતાં જુદું, લિનક્સમાં નામ વ્યવહારું રીતે માત્ર સ્લેષ (/) અને ખાલી જગ્યા (NULL character)ને બાદ કરતાં બધા ASCII અક્ષરો વાપરી શકાય છે. અન્ય કોઈ પણ ‘કંટ્રોલ ક્રેક્ટર્સ’ અથવા છાપી ન શકાય તેવા (non-printable) અક્ષરો વાપરવાની પણ છૂટ છે. `name`, `^myname^`, `++`, `-{ }()`, `test$#`, `xy.ab.ef` વગેરે કેટલાંક માન્ય ફાઈલ કે ડિરેક્ટરીનાં નામોનાં ઉદાહરણ છે.

તેમ છતાં, એ ખાસ સલાહભર્યું છે કે, ફાઈલ કે ડિરેક્ટરીનાં નામ અર્થપૂર્ણ તેમજ તર્કસંગત હોય અને તેમાં બને ત્યાં સુધી અંગ્રેજી મૂળાક્ષરો, અંકો, પૂર્ણવિરામનું ચિહ્ન (.), હાઈફન (-), અને નીચે લીટી (અન્ડરસ્કોર ‘_’)નો જ ઉપયોગ કરવો. લિનક્સ ચુસ્તપણે ‘કેસ-સેન્સિટીવિટી’ને અનુસરે છે અને તેથી આપણા માટે ભલે `math`, `Math` અને `MATH` વગેરે એકસરખા (મેચ) જ હોય, પરંતુ લિનક્સ માટે આ ત્રણેય નામ જુદાં-જુદાં છે. એક જ સ્તર પર આવી રીતનાં નામ ધરાવતી ડિરેક્ટરીઓ એકસાથે હયાત હોય તે શક્ય છે. જો આ નામ ફાઈલનાં હોય, તો વળી પાછા તે એ જ ડિરેક્ટરીમાં એકસાથે હયાત હોઈ શકે છે.

ફાઈલો સાથે કામ કરવું

ડિરેક્ટરી સામાન્ય રીતે એક પાત્ર તરીકે કાર્ય કરે છે. આપણો ડેટા સામાન્ય રીતે ફાઈલમાં સંગ્રહવામાં આવે છે અને આવી ઘણી બધી સંબંધિત ફાઈલોને સરળતાથી શોધી શકાય તેમજ યોગ્ય વ્યવસ્થાપન કરી શકાય તે માટે તેને કોઈ ચોક્કસ નામ ધરાવતી ડિરેક્ટરીમાં રાખી શકાય છે. ટેક્સ્ટફાઈલ તૈયાર કરવા માટે સામાન્ય રીતે `nano`, `pico`, `vi` અથવા `vim`, `ed` અને એવા બીજા ટેક્સ્ટએડિટર એકદમ યોગ્ય છે. તેમ છતાં, ઘણીવાર ઉપયોગકર્તાને કોઈ ફાઈલ વધુ ઝડપથી બનાવવી હોય છે. આવી પરિસ્થિતિમાં *cat* કમાન્ડ હાથવગું હથિયાર સાબિત થઈ શકે છે. આમ તો, મોટે ભાગે આ કમાન્ડનો ઉપયોગ નાની ફાઈલને ટર્મિનલ પર દર્શાવવા માટે કરવામાં આવે છે. પરંતુ નવી ફાઈલ તૈયાર કરવા અને બે ફાઈલોને જોડવા તેમજ કોઈ ફાઈલના છેડે માહિતી ઉમેરવા માટે પણ તેનો ઉપયોગ કરી શકાય છે.

cat કમાન્ડનો ઉપયોગ કરી ફાઈલ બનાવવી

તો ચાલો *economics* નામની ડિરેક્ટરીમાં *introduction* નામની ફાઈલ બનાવીએ. આ માટે સૌથી પહેલાં તમારે *economics* નામની ડિરેક્ટરીમાં જવું પડે, એટલે કે તેને કરન્ટ ડિરેક્ટરી બનાવવી પડે. અહીં તમે `cd` કમાન્ડનો ઉપયોગ કરી શકો. હવે તમે નીચે દર્શાવ્યા મુજબ *cat* કમાન્ડ ટાઈપ કરી, તેની પાછળ (>) ‘ના કરતાં મોટી’ (greater than) માટેનું ચિહ્ન (>) ટાઈપ કરી તેની પાછળ ફાઈલનું નામ ટાઈપ કરો.

Scat > introduction

જ્યારે આપણે આ કમાન્ડનો અમલ કરીશું ત્યારે ફાઈલમાં સંગ્રહવા માટેનું લખાણ ટાઈપ કરવા આપણી રાહ જોતું કર્સર કમાન્ડ લાઈનની નીચેની લીટી પર આવી જશે. તમારે ફાઈલમાં જે લખાણ સંગ્રહવું હોય તે ટાઈપ કરો અને CTRL + d કી દબાવો. આમ કરવાથી કર્સર પાછું કમાન્ડ પ્રોમ્પ્ટ પર આવી જશે. લિનક્સમાં CTRL + dનું સંયોજન ફાઈલના અંત (end of file)નો નિર્દેશ કરે છે. અહીં ઉપરના કમાન્ડમાં ઉપયોગમાં લેવાયેલું ના કરતાં વધુ (greater than)નું ચિહ્ન (>)

દિશાફેર (redirection) ચિહ્ન તરીકે ઓળખાય છે. શેલ shellને એવી સૂચના આપવા માટે કે દિશાફેર કરવાની જરૂર છે, તે માટે તેનો ઉપયોગ કરવામાં આવે છે, ઉદાહરણ તરીકે ઈનપુટ કોઈ ચોક્કસ ફાઈલમાં જવું જોઈએ. cat કમાન્ડનો જ્યારે 'ના કરતાં વધુ' (greater than) ચિહ્ન (>) વગરનો ઉપયોગ કરવામાં આવે છે, ત્યારે આર્ગ્યુમેન્ટ તરીકે દર્શાવેલ ફાઈલના લખાણને દર્શાવે છે. આકૃતિ 6.13 ફાઈલ બનાવવાની અને પછી દર્શાવવાની પ્રક્રિયા દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~/subject/economics$ cat > introduction
A field of social science that is used to study and analyze
the process of production, distribution and consumption of a
product or service.
administrator@ubuntu:~/subject/economics$ cat introduction
A field of social science that is used to study and analyze
the process of production, distribution and consumption of a
product or service.
administrator@ubuntu:~/subject/economics$
```

આકૃતિ 6.13 : cat કમાન્ડનો ઉપયોગ કરી ફાઈલ બનાવવી અને દર્શાવવી

cat કમાન્ડ દ્વારા વિગતો ઉમેરવી

એવું અનુમાન કરો કે, તમારી પાસે એક ફાઈલ હયાત છે અને તમે એમાં થોડી વધુ માહિતી ઉમેરવા માંગો છો, આ માટે અહીં ફરી થોડાક ફેરફાર સાથે cat કમાન્ડનો ઉપયોગ કરવામાં આવશે. અગાઉના ઉદાહરણમાં વપરાયેલ દિશાફેર (redirection) ચિહ્નને બદલે એપેન્ડ આઉટપુટ (>>) દિશાફેર ચિહ્નનો ઉપયોગ કરવો પડશે. *introduction* નામની ફાઈલના છેડે ડેટા ઉમેરવા માટેનો કમાન્ડ નીચે દર્શાવેલ છે.

\$cat >> introduction

An alternate definition states that Economics is a science which studies human behaviour as a relationship between ends and scarce means which have alternative uses.

[CTRL+d]

અહીં એ ખાસ નોંધ લો કે, જો કોઈ ફાઈલ હયાત જ હોય અને આપણે *cat > filename* કમાન્ડનો ઉપયોગ કરીએ તો, ફાઈલમાં પહેલા અગાઉના લખાણની ઉપર નવું લખાણ લખાઈ જશે. માટે, જે ફાઈલમાં કંઈક ડેટા પડ્યો હોય, તેવી ફાઈલને પસંદ કરતાં પહેલાં કાળજી રાખવી જરૂરી છે.

cat કમાન્ડના ઉપયોગથી ફાઈલોને જોડવી

એક કરતાં વધુ ફાઈલોનાં લખાણને જોડીને ભેગું કરી નવી ફાઈલમાં સંગ્રહવા માટે પણ *cat* કમાન્ડનો ઉપયોગ કરી શકાય છે. આવા કાર્ય માટે કમાન્ડની વાક્યરચના નીચે દર્શાવેલ છે :

\$cat file1 file2 > file3

ઉપર્યુક્ત કમાન્ડ દ્વારા *file3* તરીકે દર્શાવેલ નામની નવી ફાઈલ બનશે, જેમાં *file1* અને *file2* તરીકે દર્શાવેલ બન્ને ફાઈલનાં લખાણને સંગ્રહવામાં આવશે. આ રીતે બનતી નવી ફાઈલમાં વિગતોની ગોઠવણ દર્શાવેલ ફાઈલોના ક્રમ પર આધારિત છે. એટલે કે નવી ફાઈલમાં શરૂઆતમાં *file1* તરીકે દર્શાવેલ ફાઈલની માહિતી અને તેની પાછળ *file2* તરીકે દર્શાવેલ ફાઈલની માહિતી મૂકવામાં આવશે.

ફાઇલને કાઢી નાંખવી (rm)

એક કે તેથી વધુ ફાઇલોને કાઢી નાંખવા/દૂર કરવા (delete/remove કરવા) *rm* કમાન્ડનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે *introduction* નામની ફાઇલને કાઢી નાંખવા નીચે મુજબના કમાન્ડનો અમલ કરો :

\$rm introduction

એક જ કમાન્ડ આપીને એકસાથે એક કરતાં વધુ ફાઇલોને કાઢી નાંખવા માટે આપણે નીચે મુજબ *rm* કમાન્ડ આપી શકીએ.

\$rm file1 file2 file3

ઉપર મુજબ કમાન્ડ આપવાથી કમાન્ડના આર્ગ્યુમેન્ટ તરીકે આપેલ ત્રણેય ફાઇલો દૂર (delete) થઈ જશે. *rm* કમાન્ડની સાથે ઉપયોગમાં લઈ શકાય, તેવા વિવિધ વિકલ્પો કોષ્ટક 6.2માં દર્શાવવામાં આવેલા છે.

વિકલ્પ	ઉપયોગ
<i>rm -i</i> Filename	પરસ્પર સંવાદની સ્થિતિ (interactive mode)માં ફાઇલોને કાઢે છે. ફાઇલ કાઢી નાંખતાં પહેલાં ઉપયોગકર્તાને ચકાસણી માટે પૂછવામાં આવશે.
<i>rm -r</i> Directory name	ડિરેક્ટરીમાં સમાવિષ્ટ તમામ વિગતો સાથે સંપૂર્ણ ડિરેક્ટરીને કાઢી નાંખવામાં આવે છે.
<i>rm -r *</i>	હાલની ડિરેક્ટરીમાં પડેલી તમામ વિગતો (ફાઇલ અને/અથવા ડિરેક્ટરી)ને દૂર કરે છે. અહીં * ચિહ્ન વાઈલ્ડકાર્ડ અક્ષર તરીકે ઓળખાય છે. આ વિકલ્પ ખરેખર ભયજનક છે, કારણકે તે હાલની ડિરેક્ટરીમાં પડેલી તમામ ડિરેક્ટરી અને ફાઇલોને એકસાથે કાઢી નાંખે છે. આમ, આ કમાન્ડ પૂરતી જાણકારી સાથે સભાનપણે આપવો જરૂરી છે. એ કમાન્ડના પરિણામની ગંભીરતા જાણવી જરૂરી છે.
<i>rm -rf *</i>	ઉપરના <i>rm -r</i> જેવો જ છે, પરંતુ ફરક માત્ર એટલો છે કે, તે 'રાઈટ પ્રોટેક્ટ' (ફાઇલમાં લખવા કે ફાઇલ કાઢવા બાબતે સુરક્ષિત) કરેલી ફાઇલો પણ કાઢી નાંખે છે.

કોષ્ટક 6.2 : *rm* કમાન્ડના કેટલાક વિકલ્પો

Moving around the File system

અત્યાર સુધી આપણે ફાઇલ કે ડિરેક્ટરી કેવી રીતે બનાવવી અને કઈ રીતે કાઢી નાંખવી તે જ શીખ્યા. તો ચાલો, હવે આપણે એ જોઈએ કે, આપણી ફાઇલ-સિસ્ટમનો ભાગ સમાન વિગતો કેવી રીતે નિહાળવી. *ls* કમાન્ડ આપણને હાલની ડિરેક્ટરી કે દર્શાવેલ ડિરેક્ટરીમાં ઉપલબ્ધ ફાઇલ/ડિરેક્ટરીની યાદી આપે છે. *ls* કમાન્ડ દ્વારા મળતાં પરિણામ બદલવા માટે *ls* કમાન્ડ વિવિધ વિકલ્પો સાથે ઉપયોગમાં લઈ શકાય. તો ચાલો, સૌપ્રથમ કોઈ પણ વિકલ્પ વગર ખાલી *ls* કમાન્ડથી શરૂઆત કરીએ. આકૃતિ 6.14 *ls* કમાન્ડનું પરિણામ દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ ls
address.txt Desktop Downloads Music Public Templates
CSI2012 Documents examples.desktop Pictures subject Videos
administrator@ubuntu:~$
```

આકૃતિ 6.14 : *ls* કમાન્ડનું પરિણામ

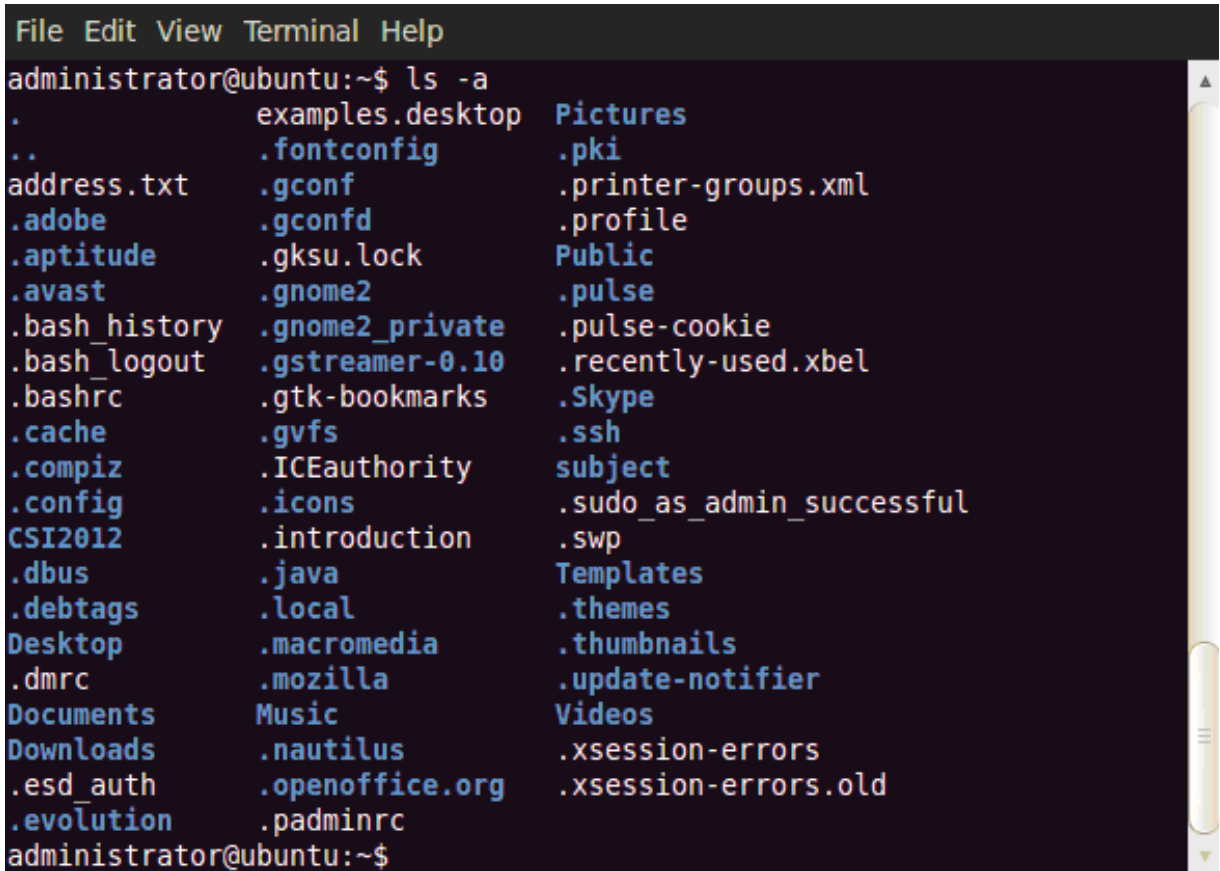
તો ચાલો, હવે આપણે *.introduction* નામની ફાઈલને બનાવીએ અને ચકાસીએ કે આપણે તેની યાદી મેળવી શકીએ છીએ કે નહીં. ફાઈલ બનાવવા માટે નીચે દર્શાવેલ કમાન્ડ ટાઈપ કરો :

Scat > .introduction

Learning Ubuntu Linux is fun....

[CTRL + d]

હવે ફાઈલોની વિગતની યાદી મેળવવા માટે ફરી ls કમાન્ડનો ઉપયોગ કરો. તમે એ અવલોકન કર્યું હશે કે તાજેતરમાં બનાવેલ ફાઈલ સ્ક્રીન પર દેખાતી નથી. અહીં એ નોંધ કરો કે, લિનક્સમાં ‘.’ થી શરૂ થતી દરેક ફાઈલને છૂપી ફાઈલ ગણવામાં આવે છે અને માટે સામાન્ય સંજોગોમાં આવી ફાઈલ દર્શાવવામાં આવતી નથી. હાલની ડિરેક્ટરીમાં પડેલી આવી છૂપી ફાઈલોની યાદી મેળવવા આપણે ls કમાન્ડની સાથે -a વિકલ્પ ઉપયોગમાં લેવો પડે. છૂપી ફાઈલોની યાદી કેવી રીતે મેળવવી તે આકૃતિ 6.15માં દર્શાવાયું છે.



```
File Edit View Terminal Help
administrator@ubuntu:~$ ls -a
.          examples.desktop  Pictures
..         .fontconfig       .pki
address.txt .gconf            .printer-groups.xml
.adobe     .gconfd           .profile
.aptitude  .gksu.lock        Public
.avast     .gnome2           .pulse
.bash_history .gnome2_private  .pulse-cookie
.bash_logout .gstreamer-0.10  .recently-used.xbel
.bashrc    .gtk-bookmarks   .Skype
.cache     .gvfs             .ssh
.compiz    .ICEauthority    subject
.config    .icons            .sudo_as_admin_successful
CSI2012    .introduction    .swp
.dbus      .java             Templates
.debtags   .local            .themes
Desktop    .macromedia       .thumbnails
.dmrc      .mozilla          .update-notifier
Documents  Music              Videos
Downloads  .nautilus         .xsession-errors
.esd_auth  .openoffice.org   .xsession-errors.old
.evolution .padminrc
administrator@ubuntu:~$
```

આકૃતિ 6.15 : છૂપી ફાઈલોની યાદી રજૂ કરવી

આકૃતિ 6.14 અને 6.15નાં પરિણામ વચ્ચેના તફાવતનું અવલોકન કરો. આકૃતિ 6.15 આગળની આકૃતિ 6.14માં દર્શાવેલ ફાઈલો કરતાં થોડી વધુ ફાઈલો દર્શાવાય છે. આકૃતિ 6.15માં દેખાતી ‘.’ અને ‘..’ નામની શરૂઆતની બે વિગતો અગત્યની છે. જ્યારે નવી ડિરેક્ટરી બનાવવામાં આવે, ત્યારે આપમેળે આવી બે વિગતો બની જ જાય છે. ls કમાન્ડની સાથે ઉપયોગમાં લઈ શકાતા કેટલાક વિકલ્પો કોષ્ટક 6.3માં આપવામાં આવેલ છે.

વિકલ્પ	ઉપયોગ
ls ~	ઉપયોગકર્તાની હોમડિરેક્ટરીમાં પડેલી ફાઇલોની યાદી રજૂ કરે છે.
ls [svics]*	એ તમામ ફાઇલોની યાદી દર્શાવો, જેના ફાઇલના નામનો પ્રથમ અક્ષર ચોરસ કૌંસમાં આપેલ કોઈ પણ અક્ષર સાથે મળતો આવતો હોય. ફાઇલ નામના બાકીના અક્ષર કોઈ પણ માન્ય ASCII અક્ષર હોઈ શકે છે.
ls [n-s][5-7]??	ચાર અક્ષરવાળી તમામ ફાઇલોની યાદી રજૂ કરે છે. એ શરત સાથે કે ફાઇલના નામનો પ્રથમ અક્ષર nથી sની વચ્ચેનો હોવો જોઈએ. બીજો અક્ષર 5થી 7ની વચ્ચેનો હોવો જોઈએ. જ્યારે ત્રીજો અને ચોથો અક્ષર કોઈ પણ માન્ય ASCII અક્ષર હોવો જોઈએ.
ls -r	ઊંધા ક્રમમાં ગોઠવીને ફાઇલની યાદી દર્શાવવા.
ls -t	સુધારો કર્યાના સમયને આધારે ગોઠવીને ફાઇલોની યાદી રજૂ કરવી.
ls -F	ફાઇલોની યાદી દર્શાવો અને તમામ એક્ઝિક્યુટેબલ ફાઇલોને * વડે અને ડિરેક્ટરીને '/' ચિહ્ન વડે નિશાની કરો.
ls -l	એક લીટી પર એક ફાઇલ એ રીતે યાદી દર્શાવો.

કોષ્ટક 6.3 : ls કમાન્ડના કેટલાક કમાન્ડ

ચોક્કસ ઢબની ફાઇલો દર્શાવવા - વાઈલ્ડકાર્ડ

ઉપરોક્ત ચર્ચામાં તમે * (asterisk) અને ? (પ્રશ્નાર્થચિહ્ન) અક્ષરનો ઉપયોગ તો જોયો જ છે. આ અક્ષરોને વાઈલ્ડકાર્ડ અક્ષર (wildcard characters) તરીકે ઓળખવામાં આવે છે, જે ઉપયોગકર્તાને તેની ઈચ્છા અનુસાર ચોક્કસ ઢબ (pattern)વાળા નામ ધરાવતી ફાઇલો શોધીને ઉપયોગમાં લેવા મદદરૂપ થાય છે. શેલ દ્વારા વપરાતા વાઈલ્ડકાર્ડની કામગીરીનો સારાંશ કોષ્ટક 6.4માં આપવામાં આવેલ છે.

વાઈલ્ડકાર્ડ	સરખાવવામાં આવનાર ચોક્કસ ઢબ (Pattern to be matching)
*	ગમે તેટલી સંખ્યામાં ગમે તે અક્ષર
?	કોઈ પણ માત્ર એક અક્ષર
[abc]	એક અક્ષર - a, અથવા b કે c ઉપયોગકર્તા આ રીતે abcને બદલે તેની જરૂરિયાત મુજબના કોઈ પણ અક્ષર મૂકી શકે.
[!abc]	a, b કે c સિવાયનો ગમે તે માત્ર એક અક્ષર (ઉપયોગકર્તા) આ રીતે abc ને બદલે તેની જરૂરિયાત મુજબના કોઈ પણ અક્ષર મૂકી શકે.
[p-s]	pથી s સુધી વચ્ચે આવતો કોઈ પણ એક અક્ષર (ઉપયોગકર્તા પોતાની જરૂરિયાત અનુસાર આ સિવાય બીજા કોઈ પણ અક્ષરનો ઉપયોગ કરી શકે.)
[!p-s]	pથી s સિવાયનો કોઈ પણ એક અક્ષર (ઉપયોગકર્તા પોતાની જરૂરિયાત અનુસાર આ સિવાય બીજા કોઈ પણ અક્ષર આ રીતે દર્શાવી શકે.)

કોષ્ટક 6.4 : wildcard અક્ષરો

ફાઇલો અને ડિરેક્ટરીઓનો કાર્યક્ષમ ઉપયોગ

આગળના વિભાગમાં આપણે એ શીખ્યા કે ફાઇલ તેમજ ડિરેક્ટરીને કેવી રીતે બનાવવી કે કાઢી નાંખવી. તો ચાલો, હવે આપણે એ જોઈએ કે copy, move અને ઉપયોગના હક્કો આપવાનું કાર્ય કેવી રીતે કરી શકાય.

ફાઇલની નકલ કરવી (cp)

ઘણીવાર આપણને આપણે તૈયાર કરેલ વિગતોની નકલ કરી રાખવાની જરૂર પડે છે. cp કમાન્ડ તેના આર્ગ્યુમેન્ટ તરીકે દર્શાવેલ એક કે તેથી વધુ ફાઇલની નકલ તૈયાર કરે છે. તે ડિસ્ક પર ઉપયોગકર્તા દ્વારા દર્શાવાયેલ સ્થાન પર ફાઇલની પ્રતિકૃતિ તૈયાર કરે છે. cp કમાન્ડને ઓછામાં ઓછા બે આર્ગ્યુમેન્ટની જરૂર પડે છે. પ્રથમ આર્ગ્યુમેન્ટ તરીકે જેની નકલ બનાવવાની હોય તે (સોર્સ) ફાઇલનું નામ આપવું પડે, જ્યારે બીજા આર્ગ્યુમેન્ટ તરીકે નકલને જ્યાં અને જે નામે સંગ્રહવાની હોય તે (ટાર્ગેટ) ફાઇલનું નામ દર્શાવવું પડે. તો ચાલો, આપણે નીચેના કમાન્ડનો ઉપયોગ કરીને introduction નામની ફાઇલની નકલ તૈયાર કરીએ :

Scp introduction new_introduction

ઉપર્યુક્ત કમાન્ડના અમલ પછી, new_introduction ફાઇલ નામ સાથે introduction નામની ફાઇલની વિગતની નકલ તૈયાર કરાશે. જો new_introduction ફાઇલ હયાત જ હશે તો, કોઈ પણ જાતની ચેતવણી વગર જૂની વિગત ઉપર નવી વિગત (overwrite) લખાઈ જશે. જો આવી કોઈ ફાઇલ હયાત ન હોય તેવા કિસ્સામાં પહેલાં એક નવી ફાઇલ બનાવવામાં આવશે અને પછી introduction નામની ફાઇલની વિગતની તેમાં નકલ કરી દેવામાં આવશે.

એક કરતાં વધુ ફાઇલોની કોઈ ચોક્કસ ડિરેક્ટરીમાં નકલ કરવા માટે પણ cp કમાન્ડનો ઉપયોગ થઈ શકે છે. ઉદાહરણ તરીકે નીચે મુજબનો કમાન્ડ :

Scp file1 file2 my_dir

file1 અને file2 નામની બે ફાઇલોની my_dir નામની ડિરેક્ટરીમાં નકલ કરાશે. my_dir નામની ડિરેક્ટરી હયાત હોવી અત્યંત આવશ્યક છે., નહીંતર ભૂલ સંદેશ દર્શાવાશે. કોષ્ટક 6.5માં cp કમાન્ડના ઉપયોગનાં કેટલાંક ઉદાહરણ આપેલ છે.

કમાન્ડ	વર્ણન
cp /vol/examples/tutorial/science.txt .	science.txt ફાઇલની હાલની ડિરેક્ટરીમાં નકલ કરે છે. છેલ્લે દર્શાવેલ ટપકું (dot .) હાલની ડિરેક્ટરીનો નિર્દેશ કરે છે.
cp chap01 progs/unit1	chap01 નામની ફાઇલની progs નામની ડિરેક્ટરીમાં unit1 નામની ફાઇલ તરીકે નકલ કરશે. (Progs ડિરેક્ટરીમાં unit1 નામની ડિરેક્ટરી હયાત ન હોવી જોઈએ.)
cp chap01 progs	chap01 નામની ફાઇલની progs ડિરેક્ટરીમાં એ જ નામે નકલ કરવામાં આવશે. (કારણકે progs ડિરેક્ટરી છે.)
cp -r progs newprogs	progs ડિરેક્ટરીમાં પડેલ તમામ વિગતો સાથે આખી ડિરેક્ટરીની નકલ કરવામાં આવશે અને newprogs નામની ડિરેક્ટરી તરીકે સંગ્રહે છે.

કોષ્ટક 6.5 : cp કમાન્ડના કેટલાંક નમૂના

ફાઇલનું નામ બદલવું અને/અથવા અન્યત્ર લઈ જવી (mv)

ફાઇલ કે ડિરેક્ટરીનું નામ બદલવું એ એવી અન્ય ક્રિયા છે કે જે ઉપયોગકર્તા નિયમિત રીતે કરતા હોય છે. mv કમાન્ડનો ઉપયોગ ફાઇલ કે ડિરેક્ટરીનું નામ બદલવા ઉપયોગમાં લેવાય છે. ઉદાહરણ તરીકે, introduction નામની ફાઇલને introduction.txt તરીકે નામ આપવા નીચે મુજબના કમાન્ડનો અમલ કરો.

\$mv introduction introduction.txt

આ કમાન્ડ ફાઇલને નવા નામ સાથે તે જ જગ્યાએ સંગ્રહશે. આમ, ફાઇલનું નામ બદલતી વખતે કોઈ જગ્યા વપરાશે નહીં.

કોઈ એક ફાઇલ કે ફાઇલોના સમૂહને અન્ય જુદી ડિરેક્ટરીમાં સ્થાનાંતર કરવા માટે પણ *mv* કમાન્ડનો ઉપયોગ કરી શકાય. ઉદાહરણ તરીકે,

\$mv file1 file2 my_dir

આ કમાન્ડ *file1* અને *file2* નામની ફાઇલને *my_dir* નામની ડિરેક્ટરીમાં લઈ જશે.

ડિરેક્ટરીના નામને બદલવા માટે પણ આ કમાન્ડનો ઉપયોગ થાય છે. ઉદાહરણ તરીકે નીચેનો કમાન્ડ જુઓ :

\$mv math mathematics

આ કમાન્ડ *math* નામની ડિરેક્ટરીનું નામ બદલીને *mathematics* કરી આપશે.

પરિણામનું પાઈપિંગ (piping) કરવું (more)

લખાણના પાનાને સ્ક્રીન પર દર્શાવતી વખતે એક સમયે એક પાનું દર્શાવાય તેવું કરવા *more* કમાન્ડ ઉપયોગમાં આવે છે. ઉદાહરણ તરીકે, જો *introduction.txt* ફાઇલનું લખાણ એક જ સ્ક્રીનમાં સમાઈ જાય તેટલું ન હોય, તો સ્ક્રીન પર ફાઇલનું લખાણ વાંચવું મુશ્કેલીભર્યું બને છે. જો *more* કમાન્ડનો ઉપયોગ કરવામાં આવે, તો તે એક સમયે માત્ર એક જ પાનું દર્શાવે છે. પછીનું પાનું દર્શાવાય તે માટે આપણે કી-બોર્ડની કોઈ પણ કી દબાવવાની હોય છે. સામાન્ય રીતે અગાઉનું (previous) પાનું જોવા માટે ‘b’ કી અને સૌથી પહેલું પાનું જોવા માટે ‘f’ કી દબાવી શકીએ. આ કમાન્ડનો એક ઉદાહરણરૂપ ઉપયોગ નીચે દર્શાવેલ છે:

\$more introduction.txt

બે ફાઇલોને સરખાવવી (cmp)

cmp કમાન્ડ કોઈ પણ પ્રકારની બે ફાઇલોની સરખામણી કરે છે અને તેના પરિણામને સ્ટાન્ડર્ડ આઉટપુટ સાધન (standard output device) પર રજૂ કરે છે. જો સરખાવવામાં આવતી બે ફાઇલો તેના લખાણની બાબતમાં જુદી પડતી હશે, તો સૌથી પહેલાં જે લીટી અને લાઈન પર પહેલો તફાવત માલૂમ પડશે તે દર્શાવવામાં આવશે. જો બે ફાઇલના લખાણમાં કોઈ તફાવત નહીં મળે તો તેવા કિસ્સામાં આપણને સીધો જ કમાન્ડ પ્રોમ્પ્ટ દેખાશે. કમાન્ડનો એક ઉદાહરણરૂપ ઉપયોગ નીચે દર્શાવેલ છે.

\$cmp introduction introduction.txt

તફાવત (Diff)

cmp કમાન્ડનું વિસ્તરણ *diff* કમાન્ડ છે. *diff* કમાન્ડ તેને આર્ગ્યુમેન્ટ તરીકે મળેલ બે ફાઇલોનાં લખાણને સરખાવે છે અને બંને ફાઇલોનાં લખાણને દર્શાવે છે અને જણાવે છે કે તફાવત ક્યાં છે. *diff* કમાન્ડની કાર્યશૈલી સમજવા માટે આપણે *introduction.txt* ફાઇલની નકલ બનાવીને તેનું નામ *new_intro.txt* આપ્યું છે. આપણે નવી ફાઇલમાંથી કેટલીક લીટીઓ પણ કાઢી નાંખી છે. નીચેના કમાન્ડનું પરિણામ આકૃતિ 6.16માં દર્શાવાયું છે :

\$diff introduction.txt new_intro.txt

```
File Edit View Terminal Help
administrator@ubuntu:~/subject/economics$ diff introduction.txt new_intro.txt
3d2
< product or service.
administrator@ubuntu:~/subject/economics$
```

આકૃતિ 6.16 : diff કમાન્ડનું પરિણામ

આકૃતિ 6.16 માં < ચિહ્ન સાથે શરૂ થતી લીટી એવો નિર્દેશ કરે છે કે introduction.txt ફાઇલમાં દર્શાવેલ લખાણ છે, પરંતુ new_intro.txt ફાઇલમાં એ લખાણ નથી. new_intro.txt ફાઇલમાં કોઈ પણ ફેરફાર હશે, તો તેવા કિસ્સામાં લીટીની શરૂઆતમાં > ચિહ્ન દર્શાવશે.

ફાઇલમાં સંગ્રહેલા અક્ષરો, શબ્દો અને લીટીઓ ગણવા (wc)

કોઈ ચોક્કસ ફાઇલ કે ફાઇલોમાં પડેલી કુલ લીટીઓ, શબ્દો અને અક્ષરોની સંખ્યા ગણવા માટે wc કમાન્ડ ઉપયોગી છે. ફાઇલમાં પડેલી લીટીઓ, શબ્દો અને અક્ષરોની સંખ્યા ગણવા માટે wc કમાન્ડની સામે ત્રણ વિકલ્પો અનુક્રમે -l, -w અને -c નો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે, introduction.txt ફાઇલમાં લીટીઓની સંખ્યા ગણવા માટે નીચે મુજબના કમાન્ડનો અમલ કરો.

```
$wc -l introduction.txt
```

```
4 introduction.txt
```

એ જ રીતે, **wc -w introduction.txt** અને **wc -c introduction.txt** કમાન્ડ આપણને introduction.txt ફાઇલમાં ઉપલબ્ધ કુલ શબ્દોની સંખ્યા તેમજ કુલ અક્ષરોની સંખ્યા આપશે. બધી જ માહિતી એકસાથે મેળવવા માટે આપણે નીચે મુજબ કમાન્ડ આપવો પડે :

```
$wc -l -w -c introduction.txt
```

```
4 49 307 introduction.txt
```

ફાઇલ-પરવાનગી (File permissions)

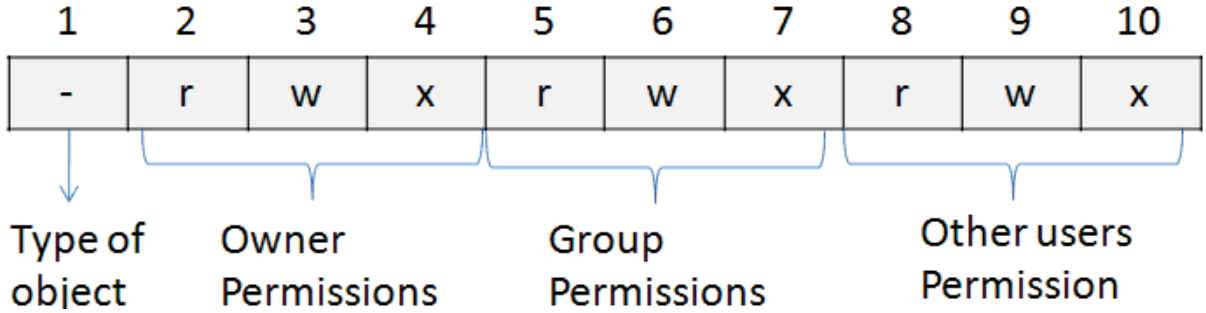
અગાઉના વિભાગમાં આપણે ls કમાન્ડની સાથે ઉપયોગમાં લઈ શકાતા કેટલાક વિકલ્પો જોયા હતા. ls કમાન્ડના બીજા ઘણા વિકલ્પો પણ છે. ઉદાહરણ તરીકે નીચે મુજબના કમાન્ડનો જ્યારે અમલ કરાય, ત્યારે તેની નીચે દર્શાવ્યા મુજબનું પરિણામ જોવા મળે.

```
$ls -l
```

```
total 6
```

```
-rw-r--r-- 1 administrator administrator 313 2013-02-15 18 :04 about_Gandhiji.txt
-rw-r--r-- 1 administrator administrator 444 2013-02-15 18 :19 introduction.txt
-rw-r--r-- 1 administrator administrator 401 2013-02-20 16 :43 address.txt
drwxr-xr-x 1 administrator administrator 4096 2013-02-21 18 :15 backup
-rw-r--r-- 1 administrator administrator 144 2013-02-13 18 :49 city.txt
-rw-r--r-- 1 administrator administrator 226 2013-02-20 14 :11 script10.sh
```

પરિણામનું અવલોકન કરો. તે આપણને આપણી ફાઇલ પદ્ધતિમાં એક ઓબ્જેક્ટ વિશેની સંપૂર્ણ યોજના આપી દે છે. ઓબ્જેક્ટ એટલે કોઈ એક ફાઇલ, એક ડિરેક્ટરી અથવા એક પ્રક્રિયા. તે આપણને ઓબ્જેક્ટના માલિક, ઓબ્જેક્ટનું કદ, ઓબ્જેક્ટના ઉદ્ભવની તારીખ અને સમયની સાથે-સાથે ઓબ્જેક્ટનું નામ દર્શાવે છે. તો ચાલો, આપણે ફાઇલ-પરવાનગીને વિસ્તારપૂર્વક સમજીએ. આકૃતિ 6.17 માલિક, જૂથ અને અન્ય ઉપયોગકર્તાના પરિપ્રેક્ષ્યમાં વિવિધ પરવાનગીઓનો સંબંધ દર્શાવે છે.



આકૃતિ 6.17 : ફાઇલ-પરવાનગીઓ

આકૃતિમાં જોઈ શકાય છે કે પહેલું ખાનું ઓબ્જેક્ટનો પ્રકાર દર્શાવે છે. પહેલા ખાનામાં ‘-’ અક્ષર ફાઇલનો નિર્દેશ કરે છે, ‘d’ અક્ષર ડિરેક્ટરીનો નિર્દેશ કરે છે અને ‘p’ અક્ષર પ્રક્રિયા (process)નો નિર્દેશ કરે છે. પછીના નવ અક્ષર ફાઇલ સિસ્ટમને જણાવે છે કે, આ ઓબ્જેક્ટ માટે કયા-કયા ઉપયોગોની પરવાનગી અપાઈ છે અને માટે જ તેને પરવાનગીઓ (permissions) તરીકે ઓળખવામાં આવે છે. લિનક્સમાં કોઈ પણ ઓબ્જેક્ટ માટે ત્રણ પરવાનગી મળી શકે છે, જેને વાંચવા માટે (read permission - r), લખવા માટે (write permission - w) અને અમલ કરવા (execute permission - x) તરીકે ઓળખવામાં આવે છે.

આકૃતિ 6.17માં ફાઇલના પ્રકાર બાદ 2થી 4 ખાનાં સુધી દર્શાવાયેલ ત્રણ અક્ષરોનાં સમૂહ એ રજૂ કરે છે કે ફાઇલના માલિકને કઈ-કઈ પરવાનગી છે. ફાઇલના માલિક (owner) એટલે એવો ઉપયોગકર્તા કે જેણે આ ફાઇલ બનાવી હોય. (આપણા કિસ્સામાં administrator) પ્રથમ સ્થાન પર **r** અક્ષરનો અર્થ છે તમને ફાઇલમાં પહેલું લખાણ વાંચવાની પરવાનગી છે. બીજા સ્થાન પર **w** અક્ષરનો અર્થ છે, તમને ફાઇલમાં લખવાની પરવાનગી છે, આમાં ફાઇલ કાઢી નાંખવાની (ડિલીટ કરવાની) પરવાનગીનો પણ સમાવેશ થઈ જાય છે. ત્રીજા સ્થાન પર **x** અક્ષરનો અર્થ છે, તમને ફાઇલનો અમલ (execute) કરવાની પરવાનગી છે. કોઈ પણ સ્થાન પર ‘-’નો અર્થ છે, તમને તે ચોક્કસ ક્રિયાની પરવાનગી આપવામાં આવી નથી. જે રીતે ls -l કમાન્ડના પરિણામમાં આપણે જોઈ શકીએ છીએ કે, administrator એટલે કે ઉપયોગકર્તા કે જે ફાઇલની માલિકી ધરાવે છે, તે about_Gandhiji.txt, introduction.txt અને address.txt ફાઇલને લખી અને વાંચી શકે છે. આકૃતિ 6.17માં સ્થાન 5થી 7 એ ઉપયોગકર્તા જૂથ (group)ને મળેલી પરવાનગીઓ દર્શાવે છે. એ જ રીતે આકૃતિ 6.17માં દર્શાવ્યા મુજબ 8થી 10 સ્થાન અન્ય કોઈ ઉપયોગકર્તા કે જૂથને આ ફાઇલના ઉપયોગ માટે આપવામાં આવેલ પરવાનગી સ્પષ્ટ કરે છે. અહીં અન્ય ઉપયોગકર્તા કે જૂથ એટલે સિસ્ટમનો એવો ઉપયોગકર્તા કે જે, ફાઇલનો માલિક જે જૂથમાં સંકળાયેલો છે, તે જૂથનો સભ્ય નથી. ઉપયોગકર્તા જો ઈચ્છે તો, તે જેની માલિકી ધરાવતો હોય, તે ઓબ્જેક્ટની પરવાનગી બદલી શકે છે.

ફાઇલોની પરવાનગી અંક સ્વરૂપે પણ આપી શકાય. અંક સ્વરૂપે પરવાનગી દર્શાવવા આપણે અષ્ટ-અંકી (8નો આધાર ધરાવતી) સંખ્યાલેખન પદ્ધતિનો ઉપયોગ કરીએ છીએ. દરેક અષ્ટ-અંકી સંખ્યા એકસાથે વાંચવાની, લખવાની અને અમલ કરવાની પરવાનગી ધરાવે છે. ઉદાહરણ તરીકે, પરવાનગી-અંક 644માં અંક “6” ઉપયોગકર્તાના અધિકાર દર્શાવે છે. “4” જૂથના અધિકારો દર્શાવે છે અને અન્ય “4” અન્યના અધિકાર રજૂ કરે છે. જ્યારે પરવાનગી તરીકે 0644 આપવામાં આવે ત્યારે, માલિકને લખવાની અને વાંચવાની પરવાનગી, જૂથ અને અન્યને વાંચવાની પરવાનગી એ રીતે અર્થઘટન કરવામાં આવે છે, જ્યારે અષ્ટ-અંકી સંખ્યાલેખન પદ્ધતિના અંકોને પરવાનગી તરીકે ઉપયોગમાં લેવાય, ત્યારે કરાતું અર્થઘટન કોષ્ટક 6.6માં દર્શાવાયેલ છે.

શાબ્દિક પરવાનગી અક્ષર	અષ્ટ-અંકી પરવાનગી-અંક	અર્થ
---	0	કોઈ પરવાનગી અપાઈ નથી.
--X	1	માત્ર અમલ કરવાની પરવાનગી અપાઈ છે.
-w-	2	માત્ર લખવાની પરવાનગી અપાઈ છે.
-wx	3	લખવાની અને અમલ કરવાની પરવાનગી અપાઈ છે.
r--	4	માત્ર વાંચવાની પરવાનગી અપાઈ છે.
r-x	5	વાંચવાની અને અમલ કરવાની પરવાનગી અપાઈ છે.
rw-	6	વાંચવાની અને લખવાની પરવાનગી અપાઈ છે.
rwX	7	બધી જ પરવાનગી અપાઈ છે.

કોષ્ટક 6.6 : અષ્ટ-અંકી સંખ્યા અને પરવાનગી

પરવાનગી બદલવી (*chmod*)

પરવાનગી બદલવા માટે આપણે *chmod* કમાન્ડનો ઉપયોગ કરીએ છીએ. પરવાનગી બદલવાના કાર્યને ‘ચેન્જમોડ ઓપરેશન’ (change mode operation) તરીકે પણ ઓળખવામાં આવે છે. ઉદાહરણ તરીકે, ઉપરના ઉદાહરણમાં જોયું કે, ઉપયોગકર્તા (માલિક)ને ફાઇલ પર વાંચવાની અને લખવાની પરવાનગી હોય છે. ફાઇલને માત્ર વાંચી શકાય તેવી ફાઇલ બનાવવા માટે નીચે મુજબનો કમાન્ડ આપી શકાય છે.

Schmod ugo-w introduction.txt

ઉપર્યુક્ત *chmod* કમાન્ડમાં ‘u’ અક્ષર ઉપયોગકર્તા (user) માટે ‘g’ જૂથ (group) માટે અને ‘o’ અન્ય (other) માટે છે. કમાન્ડનો અમલ કર્યા પછી જો આપણે ફરીથી ફાઇલની યાદી મેળવીશું, તો નીચે દર્શાવ્યા મુજબનું પરિણામ દર્શાવાશે :

Sls -l introduction.txt

-r--r--r-- 1 administrator administrator 307 2013-02-11 14 :19 introduction.txt

મહેરબાની કરીને એ નોંધ લો કે, આ ફાઇલ પર લખવાની પરવાનગી મળશે નહીં. વધુમાં, તે ઈરાદાપૂર્વક કે અજાણતા પણ ઉપયોગકર્તાને ફાઇલ દૂર (ડિલીટ) કરવા દેશે નહીં.

ફાઇલના માલિક (owner) ને લખવાની (write) અને અમલ (execute) કરવાની પરવાનગી આપવા માટે નીચે મુજબના કમાન્ડનો અમલ કરો.

Schmod u+wx script10.sh

Sls -l script10.sh

-rwxr--r-- 1 administrator administrator 226 2013-02-20 16 :05 script10.sh

અહીં *script10.sh* ફાઇલ સ્ક્રિપ્ટ-ફાઇલ તરીકે ઓળખાય છે. શેલસ્ક્રિપ્ટિંગ વિશે આપણે હવે પછીના પ્રકરણમાં શીખવાના છીએ.

જ્યારે *chmod* સાથે ઉપયોગમાં લેવાય ત્યારે ઉપયોગમાં લેવાતા કેટલાક ટૂંકાક્ષરો (abbreviations) અને તેના અર્થ કોષ્ટક 6.7માં દર્શાવવામાં આવ્યા છે.

Category	Operation	Permission
u-user	+ assign permission	r- read permission
g-group	- remove permission	w - write permission
o-other	= assign absolute permission	x - execute permission
a-all		

ફોટક 6.7 : **chmod** દ્વારા ઉપયોગમાં લેવાતા ટૂંકાક્ષરો

I/O દિશાફેર (I/O Redirection)

ઉપયોગકર્તા સ્ટાન્ડર્ડ ઇનપુટ ડિવાઇસ (કી-બોર્ડ)નો ઉપયોગ કરીને ઓપરેટિંગ સિસ્ટમ સાથે સંવાદ સાધે છે. ઓપરેટિંગ સિસ્ટમ સ્ટાન્ડર્ડ આઉટપુટ ડિવાઇસ (મોનિટર) પર પરિણામ દર્શાવે છે. આમ, જો કોઈ પણ કમાન્ડનો અમલ કરવામાં આવશે, તો તે માટેનું ઇનપુટ કી-બોર્ડ દ્વારા લેવાશે અને આઉટપુટ મોનિટર પર દર્શાવવામાં આવશે. કેટલીક વાર ઇનપુટ કે આઉટપુટની દિશાફેર કરી ફાઇલ કે પ્રિન્ટર તરફ લઈ જવી ઉપયોગી બને છે. સ્ટાન્ડર્ડ ઇનપુટના પ્રવાહને બદલવા લિનક્સમાં દિશાફેર ચિહ્ન (redirection symbol) આપવામાં આવ્યા છે. ‘ના કરતાં મોટું’, દર્શાવવા માટેનું ચિહ્ન (greater than symbol) ‘ > ’ આઉટપુટનો દિશાફેર દર્શાવે છે. આવું ચિહ્ન ઓપરેટિંગ સિસ્ટમને સૂચના આપે છે કે, આઉટપુટને મોનિટર પર દર્શાવવાને બદલે ઉપયોગકર્તા દ્વારા સૂચવવામાં આવેલ સ્થાન (ફાઇલ) પર મોકલવામાં આવે. એ જ રીતે, ‘ના કરતાં ઓછું’ દર્શાવવા માટેનું ચિહ્ન ઓપરેટિંગ સિસ્ટમને સૂચના આપે છે કે, ઇનપુટને કી-બોર્ડ પાસેથી મેળવવાને બદલે ઉપયોગકર્તા દ્વારા સૂચવવામાં આવેલ સ્થાન (file) પરથી ઇનપુટ સ્વીકારે.

ધારોકે, આપણે **wc -l < introduction.txt**, કમાન્ડ આપીએ છીએ. અહીં આપણે ઓપરેટિંગ સિસ્ટમને કી-બોર્ડને બદલે introduction.txt પરથી ઇનપુટ મેળવવા સૂચના આપીએ છીએ. એ જ રીતે **ls > list.txt** કમાન્ડનો જ્યારે અમલ કરાશે, ત્યારે ls કમાન્ડના આઉટપુટને મોનિટર પર દર્શાવવાને બદલે list.txt નામની ફાઇલમાં મોકલશે. આઉટપુટની દિશાફેર કરવામાં આવશે ત્યારે પરિણામ મોનિટર પર દર્શાવવામાં આવશે નહીં. જોકે, આઉટપુટ જોવા માટે આપણે **cat list.txt**. કમાન્ડ આપવો પડે.

પાઈપિંગ (Piping)

ઉપર ચર્ચા કરેલ દિશાફેર સવલત લિનક્સ કમાન્ડને ફાઇલ સાથે જોડવા માટે મદદરૂપ થાય છે. ઘણીવાર કોઈ એક કાર્ય પાર પાડવા માટે આપણે એક કરતાં વધુ કમાન્ડનો ઉપયોગ કરવાની જરૂર પડે છે. આવા કિસ્સામાં લિનક્સની પાઈપિંગ સવલત મદદરૂપ નીવડે છે. એક કમાન્ડમાં આઉટપુટને બીજા કમાન્ડના ઇનપુટ તરીકે પહોંચાડવા માટેની સવલતને પાઈપિંગ (piping) સવલત તરીકે ઓળખવામાં આવે છે. આ માટે બે કમાન્ડની વચ્ચે પાઈપિંગ દર્શાવવા (|) પાઈપિંગ ચિહ્ન તરીકે ટાઈપ કરવામાં આવે છે. તો ચાલો પાઈપિંગનું એક ઉદાહરણ જોઈએ.

ls | wc -l

ઉપરના કમાન્ડનો જ્યારે અમલ કરવામાં આવશે, ત્યારે ls કમાન્ડનું આઉટપુટ wc કમાન્ડને ઇનપુટ તરીકે અપવામાં આવશે. આમ, આપણને આખરી પરિણામ સ્વરૂપે હાલની ડિરેક્ટરીમાં ફાઇલોની કુલ સંખ્યા વિશેની માહિતી મળશે. જ્યારે આપણે ફિલ્ટરની સાથે ઉપયોગમાં લઈએ, ત્યારે પાઈપિંગ સવલતની સાચી ક્ષમતાનો ખ્યાલ મેળવી શકીએ છીએ. ફિલ્ટરની ચર્ચા હવે પછીના વિભાગમાં કરવામાં આવી છે.

ફિલ્ટર (Filters)

‘ફિલ્ટર’ એ એવા કમાન્ડ છે, જે સ્ટાન્ડર્ડ ઇનપુટ તરફથી ડેટા સ્વીકારે છે, તેના પર પ્રક્રિયા કરે છે અને છેલ્લે તેના પરિણામને સ્ટાન્ડર્ડ આઉટપુટ પર દર્શાવે છે. ઉબન્ટુ લિનક્સમાં વિવિધ ફિલ્ટર્સ જેવાં કે, head, tail, cut, paste, sort અને uniq ઉપલબ્ધ છે. તો ચાલો, આ ફિલ્ટરની કામગીરી જોઈએ.

ફાઈલની ટોચ પરથી લીટીઓ દર્શાવવી (head)

head કમાન્ડનો ઉપયોગ ફાઈલની શરૂઆતમાં આવેલી લીટીઓમાંથી જોઈતી સંખ્યામાં લીટીઓ દર્શાવવા કરાય છે. જો કોઈ પણ વિકલ્પ વગર ઉપયોગમાં લેવાય, તો ફાઈલની શરૂઆતની 10 લીટીઓ દર્શાવાય છે. ઉપયોગકર્તાની જરૂરિયાત અનુસાર લીટીઓ દર્શાવવા માટે આપણે head કમાન્ડની સાથે આર્ગ્યુમેન્ટ તરીકે એક સંખ્યા આપવી પડે છે. ઉદાહરણ તરીકે, *introduction.txt*, ફાઈલની પ્રથમ 2 લાઈનો દર્શાવવા નીચેના કમાન્ડનો અમલ કરો :

Shead -2 introduction.txt

ફાઈલમાં નીચેના ભાગે રહેલી લીટીઓ દર્શાવવી (tail)

tail કમાન્ડ head કમાન્ડથી બિલકુલ ઊલટું કાર્ય કરે છે. તે ફાઈલના નીચેના ભાગે રહેલી અમુક ચોક્કસ સંખ્યા જેટલી લીટીઓ દર્શાવે છે. *introduction.txt* ફાઈલની છેલ્લી 2 લીટી દર્શાવવા નીચેના કમાન્ડનો અમલ કરો :

Stail -2 introduction.txt

ફાઈલની અંદર nth લીટીથી પછીની લીટીઓ દર્શાવવા આપણે tail કમાન્ડનો ઉપયોગ કરી શકીએ. ઉદાહરણ તરીકે, જો આપણે નીચે મુજબના કમાન્ડનો અમલ કરીએ :

\$tail +5-n introduction.txt, તો *introduction.txt* ફાઈલમાંથી 5મી લીટી પછીની લીટીઓ દર્શાવાશે.

ફાઈલની ઊભી ચીરી કાપવી (cut)

ઉપરના વિભાગમાં ચર્ચેલ head અને tail કમાન્ડનો ઉપયોગ ફાઈલની માહિતીની આડી રીતે કાપવા માટે થતો હતો. cut કમાન્ડનો ઉપયોગ કરીને આપણે ફાઈલની ઊભી રીતે કાપી શકીએ. જો ફાઈલમાં ચોક્કસ ડેલિમીટર (Delimiters)નો ઉપયોગ કરવામાં આવ્યો હોય, તો cut કમાન્ડ એકદમ સરોટ અને ચોક્કસ પરિણામ આપી શકે. તો ચાલો, આ cut કમાન્ડની કાર્યશૈલી સમજવા માટે આવી ડેલિમીટેડ ફાઈલ બનાવીએ. cut કમાન્ડનો ઉપયોગ કરી *address.txt* નામની ફાઈલ બનાવો, જે નીચે દર્શાવ્યા મુજબ ડેટા સંગ્રહે :

Scat address.txt

20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023

20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120

20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058

20016, Abdul, Shamsheer, Khan, Khan Villa, M G Road, Nadiad, 388011

20017, Nirav, Jose, Mackwan, Jose House, M G Road, Nadiad, 388011

20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058

તો ચાલો, વિવિધ વિકલ્પો સાથે cut કમાન્ડ કેવી રીતે વાપરવો તે જોઈએ.

અક્ષરો કાપવા (-c)

ફાઈલની દરેક લીટીમાંથી ચોક્કસ અક્ષરોને મેળવવા -c વિકલ્પ સાથે cut કમાન્ડનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે, *address.txt* નામની ફાઈલમાંથી રોલનંબર અને નામની માહિતી બહાર કાઢવા નીચે મુજબના કમાન્ડનો અમલ કરો :

Scut -c 1-15 address.txt

20013, Vaidehi,

20014, Dhrumil,

20015, Harshit,

20016, Abdul, S

20017, Nirav, J

20018, Vidita,

આઉટપુટ ભલે અમુક અંશે સારું દેખાતું હોવા છતાં, તે બિલકુલ તેવું નથી, જેવી આપણે અપેક્ષા રાખી હતી. Abdul અને Niravના ડેટા જુઓ. તેમાં પાછળ વધારાના અક્ષરો દેખાય છે. -c વિકલ્પ ચોક્કસ લંબાઈના ફિલ્ડવાળા ડેટા માટે ઉપયોગી છે, આપણા ડેટામાં નામનાં ફિલ્ડ ચોક્કસ લંબાઈ (fixed length)નો ઉપયોગ કરીને સંગ્રહવામાં નથી આવ્યાં, માટે આપણને આઉટપુટમાં સમસ્યા થાય છે.

ફિલ્ડ કાપવાં (-f)

-c વિકલ્પમાં દર્શાવેલ સમસ્યાનો ઉકેલ લાવવા આપણે ડેલિમીટરનો ઉપયોગ કરી શકીએ. ડેલિમીટર દ્વારા છૂટી પાડેલ કિંમતોને cut કમાન્ડ અલગ ફિલ્ડ તરીકે ગણી શકે છે. અહીં એ અવલોકન કરો કે, address.txt file ફાઇલમાં આપણે ડેલિમીટર તરીકે ‘,’ (અલ્પવિરામ) ચિહ્નનો ઉપયોગ કર્યો છે. આમ, માત્ર રોલનંબર અને નામ અલગ તારવવા આપણે નીચેના કમાન્ડનો અમલ કરી શકીએ :

```
Scut -d "," -f 1,2 address.txt
```

20013, Vaidehi

20014, Dhrumil

20015, Harshit

20016, Abdul

20017, Nirav

20018, Vidita

અહીં એ અવલોકન કરો કે, હવે આપણને ઈચ્છિત પરિણામ મળ્યું છે. આ કમાન્ડમાં -d વિકલ્પનો ઉપયોગ ફાઇલમાં આવતા ડેલિમીટર દર્શાવવા માટે થાય છે. (આપણા કિસ્સામાં ‘,’) અને જે ફિલ્ડને સામેલ કરવા હોય, તે ફિલ્ડનો ક્રમ દર્શાવવા -f વિકલ્પનો ઉપયોગ કરવામાં આવે છે. (આપણા કિસ્સામાં roll number (1) અને first name (2))

ફાઇલની વચ્ચેથી ફિલ્ડને તારવીને પણ ફાઇલની માહિતીને ઊભી ચીરી શકાય છે. ઉદાહરણ તરીકે, ધારો કે, આપણે first name, city અને pincode ની માહિતી દર્શાવવા માંગીએ છીએ, તો આપણે ફિલ્ડક્રમ 2થી 7થી આગળનાં ફિલ્ડ કાપવા પડે. આ કાર્ય નીચેના કમાન્ડનો અમલ કરવાથી પાર પાડી શકાય છે :

```
Scut -d "," -f 2,7- address.txt
```

Vaidehi, Gandhinagar, 382023

Dhrumil, Vidyanagar, 388120

Harshit, Ahmedabad, 380058

Abdul, Nadiad, 388011

Nirav, Nadiad, 388011

Vidita, Ahmedabad, 380058

અહીં કમાન્ડમાં 7- નો અર્થ એ થાય કે, આપણે address.txt ફાઇલમાંથી ફિલ્ડ ક્રમ 7 પછીનાં તમામ ફિલ્ડ દર્શાવવા

ઈચ્છીએ છીએ. આઉટપુટને ફાઇલ તરફ દિશાકર કરવાનું પણ શક્ય છે. ઉદાહરણ તરીકે, જો આપણે નીચેના કમાન્ડનો અમલ કરીશું તો :

```
Scut -d "," -f 2,7- address.txt > output_cut.txt
```

કમાન્ડનું આઉટપુટ મોનિટર પર દર્શાવવાને બદલે output_cut.txt નામની ફાઇલમાં જશે.

લખાણને જોડવું (*paste*)

બે ફાઇલોને paste કમાન્ડ દ્વારા ભેગી કરી શકાય છે. *paste* કમાન્ડ સારી રીતે કામ કરી શકે તે માટે આપણે એ ખાતરી કરી લેવી પડે કે, બંને ફાઇલમાં એક જ સરખી સંખ્યામાં લીટીઓ હોય. જો બંને ફાઇલમાં એક જ સરખી સંખ્યામાં લીટીઓ નહીં હોય, તો કદાચ paste કમાન્ડનું પરિણામ આપણી અપેક્ષા મુજબ નહીં મળે, કારણકે આ કમાન્ડ paste કાર્ય ફાઇલની શરૂઆતથી કરે છે.

સૌપ્રથમ આપણે cut કમાન્ડનો ઉપયોગ કરી part1.txt અને part2.txt નામની બે જુદી-જુદી ફાઇલ બનાવીશું અને પછી paste કમાન્ડ દ્વારા એને જોડી દઈશું. આકૃતિ 6.18 આ કાર્ય કરવા માટેની પ્રક્રિયા દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ cut -d "," -f 1-4 address.txt > part1.txt
administrator@ubuntu:~$ cut -d "," -f 5-8 address.txt > part2.txt
administrator@ubuntu:~$ cat part1.txt
20013, Vaidehi, Sanjay, Shah
20014, Dhruvil, Ajay, Patel
20015, Harshit, Amit, Jain
20016, Abdul, Shamsher, Khan
20017, Nirav, Jose, Mackwan
20018, Vidita, Harshal, Arolkar

administrator@ubuntu:~$ cat part2.txt
Sector-23, GH-6, Gandhinagar, 382023
Yesh Enclave, Mota Bazar, Vidyanagar, 388120
58, Jaldeep I, Ahmedabad, 380058
Khan Villa, M G Road, Nadiad, 388011
Jose House, M G Road, Nadiad, 388011
17, Jaldeep I, Ahmedabad, 380058

administrator@ubuntu:~$ paste part1.txt part2.txt
20013, Vaidehi, Sanjay, Shah      Sector-23, GH-6, Gandhinagar, 382023
20014, Dhruvil, Ajay, Patel      Yesh Enclave, Mota Bazar, Vidyanagar, 388120
20015, Harshit, Amit, Jain       58, Jaldeep I, Ahmedabad, 380058
20016, Abdul, Shamsher, Khan     Khan Villa, M G Road, Nadiad, 388011
20017, Nirav, Jose, Mackwan     Jose House, M G Road, Nadiad, 388011
20018, Vidita, Harshal, Arolkar  17, Jaldeep I, Ahmedabad, 380058

administrator@ubuntu:~$
```

આકૃતિ 6.18 : paste કમાન્ડનું ઉદાહરણ

આઉટપુટને ક્રમબદ્ધ ગોઠવવું (sort)

ફાઈલમાં સંગ્રહેલી વિગતોને દર્શાવતી વખતે ચઢતા કે ઊતરતા ક્રમમાં ગોઠવીને રજૂ કરવા sort કમાન્ડનો ઉપયોગ થાય છે. cut કમાન્ડની જેમ તે ફિલ્ડને ઓળખી કાઢે છે અને તે ચોક્કસ ફિલ્ડ પર ડેટાને ગોઠવે છે. જ્યારે કોઈ પણ વિકલ્પ વગર sort કમાન્ડનો ઉપયોગ કરવામાં આવે તો તે આખી લીટીને આધારે ફાઈલને ગોઠવે છે. તે ASCII ક્રમના આધારે લીટીઓને પુનઃ ગોઠવે છે. ગોઠવણીની ક્રિયા, સૌપ્રથમ ખાલી જગ્યાઓ એના પછી અંકો, ત્યાર બાદ મોટા અક્ષરો (અપરકેસ) અને છેલ્લે નાના (લોઅરકેસ) અક્ષરો પર કરવામાં આવે છે.

તો ચાલો, આપણે address.txt ફાઈલને રોલનંબરના ઊતરતા ક્રમમાં ગોઠવવા પ્રયત્ન કરીએ. (તમે એ અવલોકન કર્યું જ હશે કે આ ફાઈલ ચઢતા ક્રમમાં તો ગોઠવાયેલી છે જ.) ઊલટા (ઊતરતા) ક્રમમાં ગોઠવાયેલી ફાઈલની માહિતી દર્શાવવા નીચે મુજબ કમાન્ડનો અમલ કરીએ :

```
$sort -r address.txt
```

20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058

20017, Nirav, Jose, Mackwan, Jose House, M G Road, Nadiad, 388011

20016, Abdul, Shamsheer, Khan, Khan Villa, M G Road, Nadiad, 388011

20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058

20014, Dhrumil, Ajay, Patel, Yesh Enclave, Mota Bazar, Vidyanagar, 388120

20013, Vaidehi, Sanjay, Shah, Sector-23, GH-6, Gandhinagar, 382023

નોંધ :

sort કમાન્ડનો અમલ, મૂળ ફાઈલમાં કોઈ જાતનો સુધારો કરશે નહીં. મૂળ ફાઈલમાં તો રેકોર્ડ તેની મૂળ સ્થિતિમાં જ રહેશે. sort કમાન્ડ મુજબ રેકોર્ડની ગોઠવણી તો માત્ર આઉટપુટ દર્શાવતી વખતે જ અમલમાં આવશે.

મોટે ભાગે sort કમાન્ડ હંમેશાં કોઈ અન્ય કમાન્ડની સાથે જ પ્રયોજાય છે. ઉદાહરણ તરીકે, આપણે તેને cut કમાન્ડના આઉટપુટને ગોઠવવા માટે આપણે cut કમાન્ડ સાથે વાપરી શકીએ. sort કમાન્ડ અને cut કમાન્ડ બંનેને જ્યારે એક સાથે વાપરવામાં આવે, તો પરિણામ કેવું આવે, તે જોવા નીચેના કમાન્ડનો ઉપયોગ કરો:

```
$cut -d "," -f 2-4 address.txt | sort
```

Abdul, Shamsheer, Khan

Dhrumil, Ajay, Patel

Harshit, Amit, Jain

Nirav, Jose, Mackwan

Vaidehi, Sanjay, Shah

Vidita, Harshal, Arolkar

ઉપરના ઉદાહરણમાં cut કમાન્ડ address.txt ફાઈલમાંથી બીજું, ત્રીજું અને ચોથું ખાનું (કોલમ) તારવે છે. આ રીતે અલગ તારવેલું આઉટપુટ sort કમાન્ડને ઈનપુટ તરીકે આપવામાં આવે છે. એ પછી sort કમાન્ડ મળેલ ઈનપુટને ગોઠવીને સ્ક્રીન પર દર્શાવે છે.

અક્ષર-રૂપાંતરણ (tr)

ફિલ્ટર તરીકે વપરાતો કમાન્ડ ડેટાની આડી કે ઊભી હરોળ સાથે કાર્ય કરે છે. જ્યારે *tr* (translate) કમાન્ડ કોઈ એક આડી હારના કોઈ એક અક્ષર સાથે કાર્ય કરવા અનુમતિ આપે છે. એક પ્રકારના અક્ષરોમાંથી અન્ય પ્રકારના અક્ષરોમાં લખાયેલી લાઈન કે ઢબ (patterns)ને અનુવાદિત (રૂપાંતરિત) કરવા ઉપયોગમાં લેવાય છે.

address.txt નામની ફાઈલ સાથે કામ કરતી વખતે તમે એ નોંધ્યું હશે કે તેમાં ડેલિમીટર તરીકે “;”નો ઉપયોગ કરવામાં આવેલ છે. ધારો કે, આઉટપુટ દર્શાવતી વખતે આપણે ડેલિમીટર દર્શાવવા નથી. તેના બદલે, આપણે ખાલી જગ્યા (blank space) દર્શાવવી છે. *tr* કમાન્ડ આપણને આવું કામ કરી આપશે. આ માટે નીચે આપેલ કમાન્ડનો અમલ કરો :

```
Scat address.txt | tr -s '[,]' '[' ]'
```

20013 Vaidehi Sanjay Shah Sector-23 GH-6 Gandhinagar 382023

20014 Dhrumil Ajay Patel Yesh Enclave Mota Bazar Vidyanagar 388120

20015 Harshit Amit Jain 58 Jaldeep I Ahmedabad 380058

20016 Abdul Shamsheer Khan Khan Villa M G Road Nadiad 388011

20017 Nirav Jose Mackwan Jose House M G Road Nadiad 388011

20018 Vidita Harshal Arolkar 17 Jaldeep I Ahmedabad 380058

અનુવાદની પ્રક્રિયા માત્ર આઉટપુટ દર્શાવતી વખતે જ થશે. તે મૂળ ડેટામાં ડેલિમીટરને કાયમી બદલી નહીં નાંખે. -s વિકલ્પ ફાઈલમાં ક્યાંય પણ દેખાતી વધારાની ખાલી જગ્યાને કાઢી નાંખે છે. જો આપણે દેખાતું અનુવાદિત આઉટપુટ ફાઈલમાં સાચવી લેવું હોય, તો તેવા કિસ્સામાં આપણે આઉટપુટને કોઈ નવી ફાઈલ તરફ redirect કરી શકીએ.

પેટર્ન મેચિંગ (grep)

ક્રિયાને શોધવી એ GUI વિનિયોગોમાં સૌથી વધુ વપરાતી ક્રિયાઓમાંની એક છે. ફાઈલની અંદર કોઈ ચાવીરૂપ શબ્દ (keyword) શોધવા માટે આપણે CTRL + f કીનો ઉપયોગ કર્યો જ હશે. કમાન્ડ લાઈન ઈન્ટરફેસ પરથી એવું જ કાર્ય *grep* કમાન્ડ કરે છે. આ કમાન્ડ search globally for a regular expression (g/re/p) ના મૂળભૂત વિચાર ઉપર કાર્ય કરે છે અને કોઈ નિયમિત પદાવલિ (regular expression)ને આખા દસ્તાવેજમાંથી (globally) શોધે છે અને જ્યાં આવી પદાવલિ મળે તે લીટીને દર્શાવે છે.

તો ચાલો, address.txt ફાઈલમાંથી નામ શોધીને તેનો રેકોર્ડ દર્શાવવા *grep* કમાન્ડનો ઉપયોગ કરીએ. આકૃતિ 6.19 કમાન્ડની કાર્યશૈલી દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ grep "Vidita" address.txt
20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058
administrator@ubuntu:~$
```

આકૃતિ 6.19 : *grep* કમાન્ડની કાર્યશૈલી

અહીં તમે એ અવલોકન કરો કે, જે શબ્દોને આપણે શોધીએ છીએ તે શબ્દો લાલ રંગે દર્શાવાયા છે. વળી, આપણે શોધવા માટેના ચાવીરૂપ શબ્દને બે અવતરણચિહ્નમાં આવરીને દર્શાવેલ છે. આ રીતે શોધવા માટેના ચાવીરૂપ શબ્દને બે અવતરણચિહ્નમાં આવરીને દર્શાવવા ફરજિયાત નથી. આમ, **grep Vidita address.txt** કમાન્ડ પણ આપણને એકસરખું જ આઉટપુટ આપશે. grep કમાન્ડની સાથે આપવામાં આવતા શોધવા માટેના શબ્દ ‘કેસ સેન્સિટીવ’ હોય છે અને તેથી “Vidita” અને “vidita” એ બંને અલગ ગણવામાં આવશે. grep કમાન્ડની સાથે આપણે જુદા-જુદા વિકલ્પોનો ઉપયોગ કરી શકીએ, જે આપણી શોધને વધુ સારી રીતે વિશુદ્ધ કરવામાં મદદરૂપ નીવડશે. કોષ્ટક 6.8માં આવા વિકલ્પો અને તેમના ઉપયોગોની યાદી આપવામાં આવી છે.

વિકલ્પ	ઉપયોગ
-c	લખાણ દર્શાવ્યા વગર માત્ર મળેલ શોધની સંખ્યા દર્શાવે છે.
-i	શોધતી વખતે અક્ષરો નાના છે કે મોટા (અપરકેસ/લોઅરકેસ) તેને અવગણે છે.
-l	લખાણને દર્શાવ્યા વગર માત્ર શબ્દો જેમાં મળ્યા હોય, તે ફાઇલનું નામ આપે છે.
-n	શોધેલ શબ્દ મળે તો તે શબ્દવાળું આખું લખાણ દર્શાવવાની સાથે લીટીનો ક્રમ સંખ્યા પણ દર્શાવાય છે.
-v	જેમાં શોધેલ શબ્દ મળતો ન હોય, તેવી લીટીઓ દર્શાવે છે.
-w	માત્ર તેવી જ લીટીઓ દર્શાવે છે, જેમાં શોધેલ આખો શબ્દ મળે.
-o	માત્ર શોધેલ શબ્દો જ દર્શાવે છે.

કોષ્ટક 6.8 : grep કમાન્ડના વિકલ્પો

grep કમાન્ડનું એક મજબૂત પાસું એ છે કે, તેમાં ચાવીરૂપ શબ્દો તરીકે નિયમિત પદાવલિનો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે, આપણે એવી વ્યક્તિઓની વિગત શોધવી છે કે જેમાં શબ્દની શરૂઆત ‘Jal’ થી થતી હોય અને અંતમાં ‘I’ આવતો હોય, તો આવા કિસ્સામાં નિયમિત પદાવલિનો ઉપયોગ કરી શકાય છે. તો ચાલો, અહીં દર્શાવેલ કમાન્ડનો અમલ કરીએ :

\$grep "Jal.*I" address.txt

આ કમાન્ડનું પરિણામ આકૃતિ 6.20માં દર્શાવ્યું છે. અહીં “Jal.*I” એ એક નિયમિત પદાવલિ છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ grep "Jal.*I" address.txt
20015, Harshit, Amit, Jain, 58, Jaldeep I, Ahmedabad, 380058
20018, Vidita, Harshal, Arolkar, 17, Jaldeep I, Ahmedabad, 380058
administrator@ubuntu:~$
```

આકૃતિ 6.20 : grep કમાન્ડમાં નિયમિત પદાવલિનો ઉપયોગ

નિયમિત પદાવલિમાં સામાન્ય રીતે કોષ્ટક 6.9માં દર્શાવેલ ઘણાં પુનરાવર્તનચિહ્નો (repetition operators) જોડાય છે.

પુનરાવર્તન	અર્થ
?	આગળની બાબત મરજિયાત છે અને એક વાર સરખાવાય છે.
*	આગળની બાબત શૂન્ય કે વધુ વખત સરખાવાશે.
+	આગળની બાબત એક કે વધુ વખત સરખાવવામાં આવશે.
{n}	આગળની બાબત પૂરેપૂરી n વખત સરખાવાશે.
{n,}	આગળની બાબત n કે વધુ વખત સરખાવવામાં આવશે.
{,m}	આગળ બાબત m વખત તો સરખાવાશે.
{n, m}	આગળની બાબત ઓછામાં ઓછી n વખત સરખાવાશે, પરંતુ m કરતાં વધુ વખત નહીં.

કોષ્ટક 6.9 : પુનરાવર્તનચિહ્ન

ફાઇલ કે ડિરેક્ટરી શોધવી (*find*)

ઘણીવાર આપણે બનાવેલી ફાઇલ કે ડિરેક્ટરી ક્યાં બનાવી હતી, તે સ્થાન ભૂલી જઈએ છીએ. આવી ભુલાઈ જવાયેલી ફાઇલ કે ડિરેક્ટરી શોધવામાં *find* કમાન્ડ મદદરૂપ નીવડે છે. *find* કમાન્ડ આપણે દર્શાવેલા, “શોધવા માટેના માપદંડ” (criteria) અનુસાર આપણે દર્શાવેલી ડિરેક્ટરી અને એ પછી તેની અંદરની તમામ પેટા ડિરેક્ટરીઓમાં શોધ કરે છે. આપણે ફાઇલ કે ડિરેક્ટરીને તેનાં નામ, માલિક, જૂથ, પ્રકાર, પરવાનગી, તારીખ અને અન્ય માપદંડને આધારે પણ શોધી શકીએ છીએ. અહીં એ ખાસ નોંધો કે જ્યારે અન્ય કોઈ પણ આર્ગ્યુમેન્ટ વગર *find* કમાન્ડનો ઉપયોગ કર્યો હોય, તો તે હાલની ડિરેક્ટરી તેમજ તેની બધી પેટા ડિરેક્ટરીઓમાં પડેલી ફાઇલો અને પેટા ડિરેક્ટરીઓનાં પાથનેમ દર્શાવે છે.

ધારોકે, આપણે અગાઉ બનાવેલ *introduction.txt* ફાઇલનું સ્થાન શોધવા ઈચ્છતા હોઈએ, તો તે માટેનો કમાન્ડ નીચે મુજબ લખાશે :

\$find -name introduction.txt

જો આ નામની ફાઇલ ઉપલબ્ધ હશે તો, પરિણામ તરીકે તેનો પથ દર્શાવાશે. અન્યથા આપણને કાંતો ભૂલસંદેશ મળે અથવા જો આવી કોઈ ફાઇલ નહીં હોય, તો કમાન્ડ પ્રોમ્પ્ટ દર્શ્યમાન થશે. અહીં એ વાત ખાસ નોંધો કે આ કિસ્સામાં તો આપણને ફાઇલનું નામ ખબર હતી. પણ ધારોકે આપણને ફાઇલના આખા નામને બદલે માત્ર શરૂઆતના અમુક જ અક્ષર યાદ હોય તો શું ? આવા કિસ્સામાં વાઈલ્ડકાર્ડ અક્ષરોનો ઉપયોગ કરી શકાય. નીચે દર્શાવેલ ઉદાહરણ “*intro*” શબ્દથી શરૂ થતી તમામ ફાઇલો શોધવામાં મદદરૂપ નીવડશે.

\$find -name intro*

```
./subject/economics/my1_dir/intro1
./subject/economics/my1_dir/intro
./subject/economics/introduction.txt
./subject/economics/introduction
./subject/economics/my_dir/intro1
./subject/economics/my_dir/intro
./subject/economics/intro1
```

તમારા કમ્પ્યુટર પર આવી “intro” શબ્દથી શરૂ થતી કેટલી ફાઇલો હશે, તેના આધારે તમારા સ્ક્રીન ઉપર આ કમાન્ડનું પરિણામ જુદું દેખાઈ શકે. વળી લિનક્સમાં આપણે ફાઇલનું અનુલંબન (એક્સ્ટેન્શન) દર્શાવતા નથી. અને તેથી ઉપરના ઉદાહરણમાં intro1, intro અને introduction એ ફાઇલ છે કે ડિરેક્ટરી તે આપણે જાણી શકતા નથી. નીચે દર્શાવ્યા મુજબ *type* વિકલ્પનો ઉપયોગ કરીને આપણે જરૂર પડે, તો શોધકાર્યને વધુ ચોક્કસ બનાવી શકીએ :

\$find -name intro* -type f

કોષ્ટક 6.10માં find કમાન્ડનાં કેટલાંક ઉદાહરણ અને તેનાં અપેક્ષિત પરિણામનું વર્ણન દર્શાવવામાં આવેલ છે :

કમાન્ડ	વર્ણન
find / -type d	રૂટ-ડિરેક્ટરીમાં ઉપલબ્ધ તમામ ડિરેક્ટરી અને સબડિરેક્ટરી શોધે છે.
find . -mtime -1	છેલ્લા 24 કલાકની અંદર જેમાં સુધારા કરેલ હોય તેવી ફાઇલ અને ડિરેક્ટરી શોધે છે.
find . -mtime +1	48 કલાક કરતાં વધુ સમય અગાઉ સુધારા કરેલ ઓબ્જેક્ટ શોધે છે.
find ./dir1 ./dir2 -name script.sh	script.sh ફાઇલ માટે “./dir1” અને “./dir2” ડિરેક્ટરીમાં શોધે છે.
find -size 0 -delete	શૂન્ય બાઈટવાળી ફાઇલ શોધે છે અને તેને ડિસ્ક પરથી કાઢી નાંખે છે.
find -executable	હાલની ડિરેક્ટરીમાં એક્ઝિક્યુટેબલ ફાઇલ શોધે છે.
find /home -user jagat	હોમ ડિરેક્ટરી અને તેની સબડિરેક્ટરીમાં જેનો માલિક (owner) 'jagat' હોય તેવી ફાઇલ કે ડિરેક્ટરી શોધે છે.
find . -perm 664	એવા ઓબ્જેક્ટને શોધે છે, જેના માલિકને અને જૂથને વાંચવા અને લખવાની પરવાનગી હોય, પરંતુ અન્ય ઉપયોગકર્તાને માત્ર વાંચવાની પરવાનગી હોય.

કોષ્ટક 6.10 : find કમાન્ડના ઉદાહરણ

સુપર યુઝર તરીકે કમાન્ડ ચલાવવા

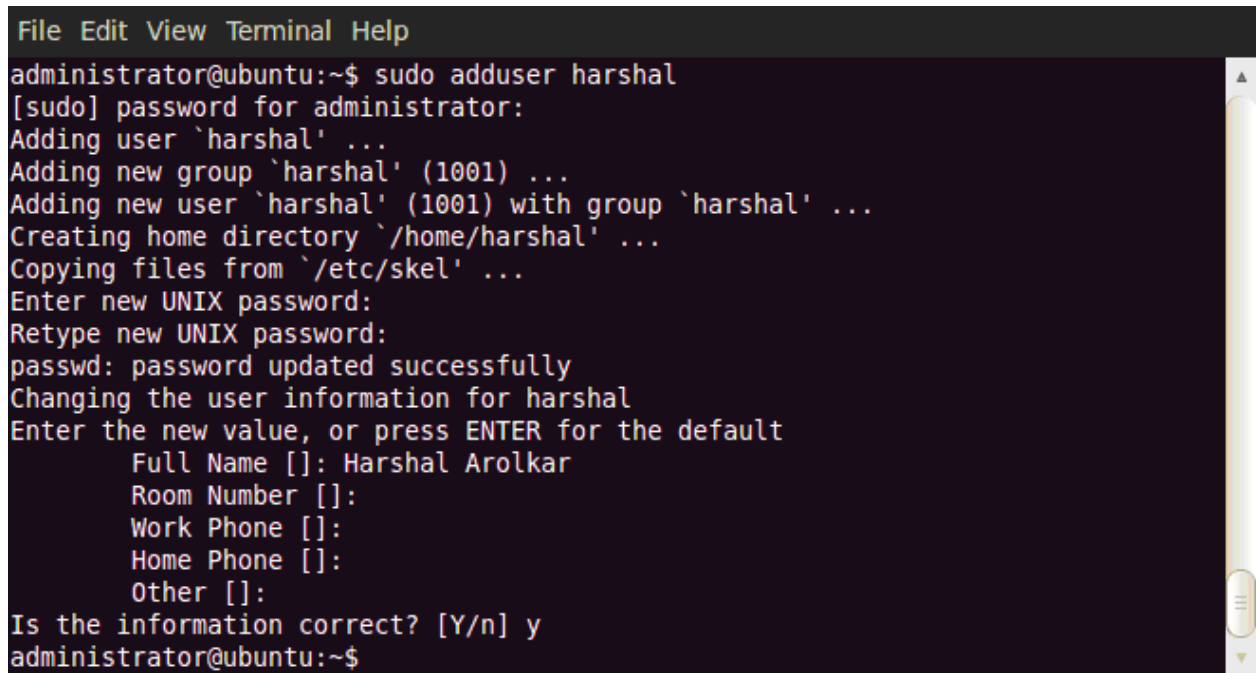
જ્યારે તમે તમારા કમ્પ્યુટરમાં લોગ-ઈન થાવ છો. ત્યારે જે ખાતું તમે વાપરો છો, તે નિયમિત ઉપયોગકર્તાનું ખાતું છે. આ ખાતાને મર્યાદિત હક્કો છે. ઉબન્ટુનું સલામતી મોડેલ (security model) સામાન્ય રીતે તમને સામાન્ય ઉપયોગકર્તા તરીકે કામ કરવા દે છે. ‘એડમિનિસ્ટ્રેટિવ રાઈટ્સ’ નહીં આપવાને કારણે અકસ્માતે થતા કોઈ પણ ફેરફાર અથવા જે સિસ્ટમની કાર્યપ્રણાલિને ખોરવી નાંખે એવા બદઈરાદાવાળા પ્રોગ્રામને પ્રસ્થાપિત થતા અટકાવે છે. પરંતુ ઘણી વખત ઉપયોગકર્તાને એડમિનિસ્ટ્રેટિવ હક્કોની જરૂર પડે છે. એડમિનિસ્ટ્રેટિવ હક્કો માત્ર ‘સુપરયુઝર’ (superuser) તરીકે ઓળખાતા ઉપયોગકર્તાને જ મળે છે. ટર્મિનલનો ઉપયોગ કરતી વખતે સુપરયુઝરના ખાતાનો ઉપયોગ કરવા, આપણે જે કમાન્ડનો અમલ કરવો હોય તે કમાન્ડની આગળ *sudo* લખવું પડે. ઉદાહરણ તરીકે, કમાન્ડ લાઈન પરથી skype નામના પ્રોગ્રામને પ્રસ્થાપિત કરવા નીચે મુજબ કમાન્ડનો અમલ કરો :

\$sudo apt-get install skype

તમે જ્યારે કમાન્ડનો અમલ કરશો, ત્યારે તે પાસવર્ડ પૂછશે, અહીં સુપરયુઝરનો પાસવર્ડ આપો. (સામાન્ય રીતે તે સામાન્ય ઉપયોગકર્તા કરતાં જુદો હોય છે.) આ પાસવર્ડ એ પ્રથમ ઉપયોગકર્તાનો પાસવર્ડ છે, જેને કમ્પ્યુટર પર લિનક્સ પ્રસ્થાપિત કરતી વખતે તમે ઉમેર્યો હતો. ટર્મિનલના માધ્યમથી એક વાર ઉપયોગકર્તા `sudo` તરીકે આધારભૂત થઈ જાય એ પછી સોફ્ટવેર પ્રસ્થાપિત થવાનું શરૂ થઈ જશે. એક વાર ઇન્સ્ટોલેશન પૂર્ણ થઈ જાય એ પછી આપણે સોફ્ટવેર વાપરવાનું શરૂ કરી શકીએ.

સુપરયુઝર પણ સિસ્ટમમાં ઉપયોગકર્તા, જૂથ અથવા ઓબ્જેક્ટને ઉમેરી શકે, કાઢી નાંખી શકે અથવા સુધારી શકે. નીચે યાદીમાં દર્શાવેલ કેટલાક કમાન્ડ આવા હેતુ માટે ઉપયોગમાં લઈ શકાય.

adduser : *adduser* કમાન્ડ સિસ્ટમ પર એક નવો ઉપયોગકર્તા બનાવે છે. આકૃતિ 6.20 ઉપયોગકર્તા બનાવવાની પ્રક્રિયા દર્શાવે છે. કમાન્ડનો જ્યારે અમલ કરવામાં આવશે ત્યારે તે પાસવર્ડ અને આકૃતિ 6.21માં દર્શાવ્યા મુજબની કેટલીક વધારાની માહિતી પૂછશે. એક વાર બધી જરૂરી વિગતો આપી દીધા બાદ સિસ્ટમમાં હોમડિરેક્ટરી સાથે એક નવો ઉપયોગકર્તા બનાવવામાં આવશે.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sudo adduser harshal
[sudo] password for administrator:
Adding user `harshal' ...
Adding new group `harshal' (1001) ...
Adding new user `harshal' (1001) with group `harshal' ...
Creating home directory `/home/harshal' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for harshal
Enter the new value, or press ENTER for the default
  Full Name []: Harshal Arolkar
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
administrator@ubuntu:~$
```

આકૃતિ 6.21 : ઉપયોગકર્તા ઉમેરવો

passwd : જ્યારે સુપરયુઝર તરીકે *passwd* કમાન્ડનો ઉપયોગ કરવામાં આવશે, ત્યારે તે આપણને સિસ્ટમના કોઈ પણ માન્ય ઉપયોગકર્તાના પાસવર્ડ બદલવા દે છે.

who : જ્યારે *who* કમાન્ડનો અમલ કરવામાં આવશે, ત્યારે સિસ્ટમમાં જેટલા ઉપયોગકર્તા લોગ-ઈન થયેલા હશે, તે બધાની યાદી દર્શાવશે.

addgroup : *addgroup* કમાન્ડ એક નવું જૂથ ઉમેરે છે. ઉપયોગકર્તાઓને સામાન્ય રીતે જૂથમાં વહેંચવામાં આવે છે, જેથી કરીને તેમના પર વધુ સારો અંકુશ રાખી શકાય.

deluser : *deluser* કમાન્ડ સિસ્ટમમાંથી એક ઉપયોગકર્તાને કાઢી નાંખે છે. અહીં એ નોંધ કરો કે, `-remove -home` વિકલ્પનો ઉપયોગ કરી ઉપયોગકર્તાની ફાઇલો અને હોમડિરેક્ટરી આપણે દૂર કરવી પડે.

delgroup : *delgroup* કમાન્ડ-સિસ્ટમમાંથી એક આખા જૂથ (group)ને કાઢી નાંખે છે. આ કાર્ય પાર પાડવા માટે આપણે સૌ પ્રથમ એ વાતની ખાતરી કરી લેવી પડે કે, આપણે જેને કાઢી નાંખવા માંગીએ છીએ, એ જૂથ સાથે કોઈ ઉપયોગકર્તા સંકળાયેલ નથી.

સારાંશ

આ પ્રકરણમાં આપણે ઉબન્ટુ કમાન્ડ લાઈન ઈન્ટરફેસનો કેવી રીતે ઉપયોગ કરવો તે શીખ્યા. જ્યારે CLI (Command Line Interface)નો ઉપયોગ કરવામાં આવે છે, ત્યારે તે આપણને GUI દ્વારા કરી શકાતાં તમામ કાર્યો કાર્યક્ષમ અને ઝડપી રીતે પાર પાડવાની સવલત આપે છે. લિનક્સ ટર્મિનલનો ઉપયોગ કરીને કેવી રીતે CLI શરૂ કરવું તે આપણે જોયું. આપણે એ પણ શીખ્યા કે કોઈ ફાઈલ કે ડિરેક્ટરી કેવી રીતે બનાવવી, તેનું નામ બદલવું, અને તેને કાઢી નાંખવી, આપણે જેમાં કામ કરી રહ્યા છીએ, તે ડિરેક્ટરી શોધી કાઢવી, જરૂર પડે તો ડિરેક્ટરી બદલવી. પાછળથી આપણે એ જોયું કે, ફાઈલ કે ડિરેક્ટરીની નકલ કેવી રીતે બનાવવી. આપણા ડેટાનો દૂરુપયોગ નથી થતો, તેની ખાતરી કરવા માટે ઉપયોગના હક્ક (access right) એ એક રસ્તો છે. આપણે એ જોયું કે ઉપયોગના હક્ક કેવી રીતે આપી શકાય કે બદલી શકાય છે. આપણે એ પણ જોયું કે પાઈપિંગનો ઉપયોગ કરી એક કરતાં વધુ કમાન્ડ જોડી કમાન્ડની અસરકારકતા કેવી રીતે વધારી શકાય. વળી, આપણે એવી લાક્ષણિકતાઓ પણ જોઈ જેવી કે, ફાઈલમાં રહેલી કુલ લીટીઓ કે શબ્દોની સંખ્યા ગણવી, ફાઈલની માહિતીને આડી કે ઊભી કાપીને જોવી, ફાઈલોને જોડવી, ફાઈલોને કે ડિરેક્ટરીને શોધવી, ફાઈલમાં અમુક ચોક્કસ શબ્દોને શોધવા માહિતી ચઢતા કે ઊતરતા ક્રમમાં ગોઠવીને દર્શાવવી. અંતમાં આપણે એ પણ જોયું કે કોઈ નવા ઉપયોગકર્તાએ નવા સોફ્ટવેરને કેવી રીતે પ્રસ્થાપિત કરવું, નવા ઉપયોગકર્તા કે જૂથ ઉમેરવાં કે કાઢી નાંખવાં, અત્યારે કોણ સિસ્ટમ વાપરી રહ્યું છે તે શોધવું અથવા કોઈ ઉપયોગકર્તાનો પાસવર્ડ બદલવો વગેરે જેવાં સામાન્ય વહીવટી કાર્યો એક સામાન્ય ઉપયોગકર્તા કેવી રીતે પાર પાડી શકે.

સ્વાધ્યાય

1. કમાન્ડ પ્રોમ્પ્ટ એટલે શું?
2. શેલનું વર્ણન કરો. કોઈ પણ ત્રણ લિનક્સ શેલનાં નામ આપો.
3. શેલ કમાન્ડનું અર્થઘટન કેવી રીતે કરે છે ? સમજાવો.
4. ઉબન્ટુ લિનક્સમાં ટર્મિનલ શરૂ કરવા માટે ઉપયોગમાં લેવાતાં પગલાં લખો.
5. નીચેના કમાન્ડની વિગતે ચર્ચા કરો :
ls, cat, wc, chmod
6. ઉબન્ટુ લિનક્સમાં 'ઈન્ટરનલ કમાન્ડ'નો અર્થ શું છે?
7. ઉબન્ટુ લિનક્સમાં કોઈ પણ કમાન્ડ વિશેની મદદ કેવી રીતે શોધી શકો?
8. યોગ્ય ઉદાહરણ આપીને વિવિધ વાઈલ્ડકાર્ડ અક્ષરો વિશે સમજાવો.
9. યોગ્ય ઉદાહરણ આપીને પાઈપ (pipe)નો ઉપયોગ સમજાવો.
10. દિશાફેર (redirection) એટલે શું ? યોગ્ય ઉદાહરણ આપી સમજાવો.
11. લિનક્સમાં વપરાતા ફિલ્ટર કમાન્ડની યાદી આપો.

12. હેતુલક્ષી પ્રશ્નો

- (1) ફાઇલમાં સમાવિષ્ટ માહિતીમાં કુલ લીટીઓ, શબ્દો અને અક્ષરોની કુલ સંખ્યા ગણવા માટેના કમાન્ડનું નામ શું છે ?
- (a) countw (b) wcount
(c) wc (d) wordcount
- (2) ફાઇલોને દૂર (remove) કરવા નીચે પૈકી કયો કમાન્ડ ઉપયોગમાં લેવાય છે ?
- (a) dm (b) rm
(c) delete (d) erase
- (3) નીચેના પૈકી કયો કમાન્ડ ડિરેક્ટરી દૂર કરવા માટે છે ?
- (a) rdir (b) remove
(c) rd (d) rmdir
- (4) કોઈ ફાઇલમાંથી માત્ર કુલ લીટીઓની સંખ્યા જાણવી હોય, તો નીચેના પૈકી કયો કમાન્ડ ઉપયોગમાં લઈ શકાશે ?
- (a) wc -r (b) wc -w
(c) wc -c (d) wc -l
- (5) chmod 761 કમાન્ડ નીચે પૈકી કયા ઉપયોગના હક્ક (access rights)ની સમકક્ષ છે ?
- (a) chmod u=7, g = 6, o = 1
(b) chmod a = 761
(c) schmod u = rws, g = rw, o = x
(d) chmod 167
- (6) લિનક્સમાં નીચેનાં પૈકી કયું ફાઇલના નામની મહત્તમ લંબાઈ રજૂ કરે છે ?
- (a) 8 (b) 10
(c) 200 (d) 255
- (7) લિનક્સમાં ફાઇલસિસ્ટમમાં ડિરેક્ટરી માળખામાં સૌથી પ્રથમ ડિરેક્ટરી કઈ હોય છે ?
- (a) \home (b) \root
(c) /home (d) /root
- (8) ફાઇલની નકલ કરવા માટે નીચેના પૈકી કયો કમાન્ડ ઉપયોગમાં લઈ શકાય ?
- (a) tar (b) cpio
(c) cp (d) copy
- (9) તમારી હાલની કામની ડિરેક્ટરી (current directory) દર્શાવવા નીચેના પૈકી કયો કમાન્ડ ઉપયોગમાં લેવાય છે ?
- (a) path (b) pwd
(c) prompt \$p\$g (d) dir

- (10) ફાઇલમાં ચોક્કસ ઢબ (pattern) શોધવા માટે નીચેના પૈકી કયો કમાન્ડ ઉપયોગમાં લેવાય છે ?
- (a) grep (b) find
(c) lookup (d) All of the above
- (11) નીચેનાં પૈકી કયું દિશાફેર (redirection) ચિહ્ન નથી ?
- (a) > (b) <
(c) * (d) >>
- (12) ઉપયોગકર્તાની ફાઇલને વાંચવાની પરવાનગી આપવા નીચેના પૈકી કઈ વાક્યરચના સાચી છે ?
- (a) chmod r filename (b) chmod u+r filename
(c) chmod filename r (d) chmod filename u+r
- (13) cp કમાન્ડની ઓછામાં ઓછી આર્ગ્યુમેન્ટ કેટલી હોય છે ?
- (a) One (b) Two
(c) Three (d) એક પણ નહીં
- (14) લિનક્સમાં mv કમાન્ડ નીચેના પૈકી કયા હેતુ માટે ઉપયોગમાં લેવાય છે ?
- (a) ડિરેક્ટરીનું નામ બદલવા (b) ફાઇલને અન્યત્ર ખસેડવા
(c) ફાઇલની નકલ કરવી (d) ઉપરના બધા
- (15) નીચેના પૈકી કયા કમાન્ડનો ઉપયોગ એક સમયે માત્ર એક પાનું ભરીને માહિતી જોવા થઈ શકે ?
- (a) More (b) more
(c) PAGE (d) page

પ્રાયોગિક સ્વાધ્યાય

1. લિનક્સ કમાન્ડનો ઉપયોગ કરીને નીચેનાં કાર્યો કરો :

- (a) ડિસેમ્બર-2012નું કેલેન્ડર બનાવો.
- (b) એવા કમાન્ડનો અમલ કરો કે જે લોગ-ઇન નેઇમ, તમારા ટર્મિનલનું નામ અને ઉપયોગકર્તાના લોગ-ઇનનો સમય અને તારીખ દર્શાવે.
- (c) 'n' અથવા 'N'થી શરૂ થતા નામવાળી ફાઇલોની યાદી મેળવો.
- (d) હાલની કામની (કરન્ટ) ડિરેક્ટરી દર્શાવો.
- (e) 11મા ધોરણના વિદ્યાર્થીઓની માહિતી ધરાવતી class11_A.txt અને class11_B.txt નામની બે ફાઇલ બનાવો. ફાઇલમાં વિદ્યાર્થીઓનાં નામ હોવાં જોઈએ. હવે આ બે ફાઇલને ભેગી કરી class11_.txt નામની એક ફાઇલ બનાવો (cat કમાન્ડનો ઉપયોગ કરો.)
- (f) ઉપર્યુક્ત 'e'માં તૈયાર કરેલી ત્રણેય ફાઇલોને છુપાવી દો.
- (g) માત્ર ડિરેક્ટરીઓની યાદી મેળવો.
- (h) cat કમાન્ડના ઉપયોગ બાબતે મદદ મેળવો.
- (i) એવી તમામ ફાઇલો કે જેનો ચોથો અક્ષર 'g' હોય અને છઠ્ઠો અક્ષર કોઈ અંક હોય તેવી ફાઇલોની યાદી મેળવો.

- (j) ટર્મિનલ કેલ્ક્યુલેટરનો ઉપયોગ કરી નીચેની ક્રિયાઓ કરો :
- (1) 2500/7 ગણો.
 - (2) દશાંશસંખ્યા 50ને તેની સમકક્ષ દ્વિઅંકી સંખ્યામાં રૂપાંતરિત કરો.
 - (3) દશાંશસંખ્યા 25ને તેની સમકક્ષ સોળ-અંકી સંખ્યામાં રૂપાંતર કરો.
 - (4) 36નું વર્ગમૂળ શોધો.
 - (5) સોળ-અંકી સંખ્યા 25ને તેની સમકક્ષ દશાંશસંખ્યામાં રૂપાંતરિત કરો.

2. નીચેના લિનક્સ કમાન્ડનું પરિણામ દર્શાવો :

- (a) `cat f1 >> f2`
- (b) `echo $SHELL`
- (c) `mkdir d1 d2 d3`
- (d) `ls s*t`
- (e) `ls [a-f]*`
- (f) `ls -R`
- (g) `ls -xR`
- (h) `cp f1 f2`
- (i) `ls | sort`
- (j) `ls | tr -s " " | cut -d " " -f 5 | sort`
- (k) `ls -l | grep -c "address.txt"`
- (l) `grep "Harshit Jain" address.txt`
- (m) `chmod u-w address.txt`
- (n) `wc -l address.txt > totalstudents`





અગાઉના પ્રકરણમાં આપણે ઉબન્ટુ લિનક્સમાં કામ કરવા માટે વાપરી શકાય તેવા આદેશો (કમાન્ડ્સ)ની ચર્ચા કરી. જેમાં એક સમયે ફક્ત એક આદેશનો અમલ કરવામાં આવતો હતો. જોકે પાઇપ્સ (pipes)નો ઉપયોગ કરીને ઘણા બધા આદેશોનો અમલ કરી શકાય, પણ જેમ આદેશોની સંખ્યા વધે તેમ પ્રક્રિયા જટિલ બને છે. આ માટે કોઈ એક ટેક્સ્ટફાઇલ (textfile)માં કમ પ્રમાણે આદેશો ટાઇપ કરવા એ એકસાથે વધારે આદેશોનો અમલ કરવાનો એક વધારે સારો રસ્તો છે. તે પછી આ ફાઇલનો અમલ કરવા માટે તે લિનક્સ શેલ (Linux shell)ને આપો. આથી, લિનક્સ શેલ ટેક્સ્ટફાઇલમાં જણાવેલા બધા આદેશોનો કમ પ્રમાણે અમલ કરશે. આ ટેક્સ્ટફાઇલ શેલસ્ક્રિપ્ટ (shell scripts) તરીકે ઓળખાય છે. શેલસ્ક્રિપ્ટ એ રીતે વ્યાખ્યાયિત કરી શકાય કે તે આદેશોની શ્રેણી લખેલી એક સાદી ટેક્સ્ટફાઇલ છે. રોજિંદા અલગ-અલગ કાર્યો કરવા અને સિસ્ટમના વહીવટ માટે વપરાશકર્તાઓ સામાન્ય રીતે શેલસ્ક્રિપ્ટ્સનો ઉપયોગ કરે છે. આ પ્રકરણમાં આપણે કેટલીક નમૂનાની શેલસ્ક્રિપ્ટ્સ સાથે એડિટર (editor) કે જે શેલસ્ક્રિપ્ટ લખવામાં મદદ કરે છે, તેના વિશે શીખીશું.

વિમ એડિટર દ્વારા કાર્ય (Working with Vim Editor) :

cat આદેશનો ઉપયોગ કરીને ફાઇલ કેવી રીતે બનાવી શકાય તે આપણે શીખી લીધું છે. જોકે cat આદેશ આપણને શેલસ્ક્રિપ્ટ બનાવતાં સમયે ફાઇલ બનાવવાની સગવડ આપે છે, પણ તે એક સારો વિકલ્પ નથી. પણ આવી કામગીરી કરવા માટે આપણને એક સારા ટેક્સ્ટએડિટરની જરૂર છે. Nano, pico, vi અથવા Vim વગેરે જેવા ટેક્સ્ટએડિટર સામાન્ય રીતે ટેક્સ્ટફાઇલ બનાવવા માટે શ્રેષ્ઠ છે. GNOME ડેસ્કટોપ વાતાવરણમાં ગ્રાફિકલ એડિટર Gedit ઉપલબ્ધ છે. જ્યારે KDE ડેસ્કટોપ વાતાવરણમાં ગ્રાફિકલ એડિટર Kwrite ઉપલબ્ધ છે. આપણે શેલસ્ક્રિપ્ટ લખવા માટે Vim એડિટરનો ઉપયોગ કરીશું, જે એક વિઝ્યુઅલ ડિસ્પ્લે એડિટર છે. આ એડિટર યુનિક્સ અને લિનક્સની લગભગ તમામ આવૃત્તિઓ સાથે ઉપલબ્ધ છે.

આ Vim ટેક્સ્ટએડિટર (Vi એ સુધારેલું એડિટર) Bram Moolenaar વડે લખાયેલું છે અને તેને 1991માં જાહેરમાં મૂકવામાં આવ્યું. તે મોટા ભાગની UNIX સિસ્ટમ સાથે વિતરણ કરવામાં આવતાં Vi એડિટરની ઉન્નત આવૃત્તિ (enhanced version) છે. ટેક્સ્ટએડિટિંગનું કાર્ય અતિ કાર્યક્ષમતાથી કરવા માટે બનાવેલું Vim એ એક અત્યંત વિન્યાસી (જરૂરિયાત પ્રમાણે ગોઠવી શકાય તેવું configurable) એડિટર છે. કમાન્ડલાઇન ઈન્ટરફેઇસમાં અને એક અલગ વિનિયોગ તરીકે ગ્રાફિકલ યુઝર ઈન્ટરફેઇસમાં બંનેમાં Vim એડિટરનો ઉપયોગ કરી શકાય છે.

વિમ એડિટરમાં કામ કરવા માટે આપણે પ્રથમ શરૂઆત કરવી પડશે. સૌપ્રથમ એક નવી ટર્મિનલ વિન્ડો (Terminal Window) ખોલો. આપણે બે રીતે વિમ એડિટર ઓપન કરી શકીએ (ખોલી શકીએ) : પ્રોમ્પ્ટ (prompt) ઉપર **vi** ટાઇપ કરી એન્ટર કી દબાવો અથવા **vim** લખ્યા પછી ફાઇલનું નામ લખ્યા પછી એન્ટર કી દબાવો. આકૃતિ 7.1માં વિમ એડિટર ઈન્ટરફેઇસ દર્શાવેલ છે કે જ્યારે આપણે ફાઇલનું નામ જણાવેલું હોતું નથી.

વિમ પદ્ધતિઓ (Vim Modes)

વિમ એડિટર ત્રણ જુદી-જુદી પદ્ધતિઓમાં કામ કરે છે, એટલે કે (1) કમાન્ડ મોડ (command mode), (2) ઇન્સર્ટ મોડ (insert mode) અને (3) લાસ્ટ લાઈન મોડ (last line mode).

કમાન્ડ મોડ (The command mode)

આપણે જ્યારે વિમ એડિટર વડે કોઈ ફાઈલને એડિટ કરવાનું શરૂ કરીશું ત્યારે વિમ એડિટર કમાન્ડ મોડમાં ખૂલશે. આપણે અનેક વિવિધ કમાન્ડ આપી શકીએ કે જે આપણને ફાઈલમાં વચ્ચે લખાણ ઉમેરવા (ઇન્સર્ટ-insert), અંતમાં લખાણ ઉમેરવા (અપેન્ડ-append), લખાણ રદ કરવા (ડિલીટ-delete) અથવા શોધવા (સર્ચ - search) અને ફાઈલમાં ફરવા (નેવિગેટ - navigate)ની સગવડ આપે છે. અહીં આપણે નોંધ કરીશું કે કમાન્ડ મોડ વાપરવાનું ચાલુ કરીએ કે તરત જ આપણે લખાણ ઉમેરી શકતા નથી. ફાઈલમાં કોઈ લખાણ ઉમેરવા માટે આપણે ઇન્સર્ટ (i), અપેન્ડ (a) અથવા ઓપન (o) કમાન્ડ સૌપ્રથમ આપવો પડે છે.

કમાન્ડ મોડનું વિસ્તરણ એ વિઝ્યુઅલ મોડ (Visual mode) છે. ફાઈલનાં લખાણનો કોઈ ભાગ પસંદ કરવા વિઝ્યુઅલ મોડ એ એક લવચીક (flexible) અને સરળ રસ્તો છે. માહિતીનો કોઈ બ્લોક કે જેમાં ફેરફાર કરવાનો છે, તેને પસંદ કરવા માટે આ એક માત્ર રસ્તો છે. કોષ્ટક 7.1માં અક્ષરો (કેરેક્ટર્સ) આપેલાં છે, જે આપણને વિઝ્યુઅલ મોડમાં મદદ કરે છે.

કમાન્ડ	ઉપયોગ
v	વિઝ્યુઅલ મોડ સક્રિય બનાવવા માટે (અક્ષરોમાં સુધારની સુવિધા આપે છે)
V	વિઝ્યુઅલ મોડ સક્રિય બનાવવા માટે (લીટીઓમાં સુધારની સુવિધા આપે છે.)
CTRL + v	બ્લોક-વિઝ્યુઅલ મોડ સક્રિય બનાવવા માટે (લખાણના લંબચોરસ વિસ્તારમાં સુધારની સુવિધા આપે છે.)

કોષ્ટક 7.1 : વિઝ્યુઅલ મોડમાં વપરાતા અક્ષરો (કેરેક્ટર્સ)

ઇન્સર્ટ મોડ (The insert mode)

જ્યારે આપણે insert, append અથવા open આદેશ આપીએ છીએ ત્યારે આપણે ઇન્સર્ટ મોડમાં રહીશું. વર્તમાન ટેક્સ્ટ એડિટર આપણને કાર્યની ચાલુ સ્થિતિ બતાવે છે. એક વખત આપણે ઇન્સર્ટ મોડમાં આવીએ, પછી આપણી ફાઈલમાં આપણે લખાણ ટાઈપ કરી શકીએ તેમજ ફાઈલની અંદર આપણે નેવિગેટ કરી શકીએ.

ESC કી દબાવીને આપણે કમાન્ડ મોડ અને ઇન્સર્ટ મોડમાં ટોગલ (toggle - એક સ્થિતિમાંથી બીજી સ્થિતિમાં બદલવું) કરી શકીએ. આપણે જ્યારે વિમ એડિટરનો ઉપયોગ કરીએ છીએ, ત્યારે આ ક્રિયા વારંવાર કરીએ છીએ. કોષ્ટક 7.2માં અક્ષરો આપેલાં છે, જે આપણને ઇન્સર્ટ મોડમાં મદદ કરે છે.

કમાન્ડ	ઉપયોગ
a	કર્સરના વર્તમાન સ્થાન પછી લખાણ ઉમેરવા માટે
i	કર્સરના વર્તમાન સ્થાન પહેલા લખાણ ઉમેરવા માટે
A	વર્તમાન લીટીના અંતમાં લખાણ ઉમેરવા માટે
I	વર્તમાન લીટીની શરૂઆતમાં લખાણ ઉમેરવા માટે
O	કર્સરના વર્તમાન સ્થાનની ઉપર એક નવી લીટી ઉમેરવા માટે
o	કર્સરના વર્તમાન સ્થાનની નીચે એક નવી લીટી ઉમેરવા માટે

કોષ્ટક 7.2 : ઇન્સર્ટ મોડમાં વપરાતા અક્ષરો (કેરેક્ટર્સ)

ફાઇલનો સંગ્રહ કરવા માટે હાલની સ્થિતિ અને વપરાશ પ્રમાણે અન્ય અનેક આદેશો ઉપલબ્ધ છે. કોષ્ટક 7.3માં એડિટરમાં વપરાતાં આદેશો (commands) અને તેનો વપરાશ (usage) દર્શાવેલ છે.

કમાન્ડ	ઉપયોગ
.w	ફાઇલનો સંગ્રહ કરી એડિટિંગ મોડ ચાલુ રાખવા માટે
:wq	ફાઇલનો સંગ્રહ કરી એડિટિંગ મોડમાંથી બહાર નીકળવા માટે
x	ફાઇલનો સંગ્રહ કરી એડિટિંગ મોડમાંથી બહાર નીકળવા માટે (ઉપરની જેમ જ)
:q	સુધારા કર્યા ન હોય ત્યારે એડિટિંગ મોડમાંથી બહાર નીકળવા માટે
:q!	ફાઇલમાં કરેલા સુધારાનો સંગ્રહ કર્યા વગર એડિટિંગ મોડમાંથી બહાર નીકળવા માટે
:saveas FILENAME	હયાત ફાઇલને નવું નામ આપી સંગ્રહ કરવા માટે અને નવી બનેલી ફાઇલમાં એડિટિંગ ચાલુ રાખવા માટે

કોષ્ટક 7.3 : ફાઇલનો સંગ્રહ કરવા માટેના લાસ્ટ લાઇન મોડના આદેશો

અહીં નોંધ કરશો કે ફાઇલનું નામ આપ્યા વગર આપણે વિમ એડિટર ખોલીશું, તો શરૂઆતમાં લખાણનો સંગ્રહ સિસ્ટમ બફર (મેઇન મેમરી)માં થશે. બફરમાંથી માહિતીને હાર્ડડિસ્કમાં સ્થળાંતર કરવા માટે *wq* આદેશ સાથે ફાઇલનું નામ આપણે ટાઇપ કરવું પડે.

ડોક્યુમેન્ટમાં અલગ-અલગ જગ્યાએ જવું (Moving around in the document)

તમે કદાચ નિરીક્ષણ કર્યું જ હશે કે *about_Gandhiji* ફાઇલ બનાવતાં સમયે એરો કી (આકસ્મિક રીતે જો કદાચ તમે વાપરી હોય તો) આપણી અપેક્ષા પ્રમાણે કામ કરતી નથી. સામાન્ય રીતે, એરો કીનો ઉપયોગ કર્સરને ઉપર, નીચે, ડાબી અને જમણી દિશામાં ખસેડવા માટે થાય છે. ઈન્સર્ટ મોડમાં આપણે શાબ્દિક માહિતી (ટેક્સ્ટ) ટાઇપ કરવા સિવાય બીજું કંઈ પણ કરી શકતા નથી. જ્યારે આપણે વિમ એડિટર માં કમાન્ડ મોડમાં કામ કરતાં હોઈએ, ત્યારે ડોક્યુમેન્ટમાં અન્ય જગ્યાએ જવા માટે ખાસ કી-સ્ટ્રોકની આપણને જરૂર પડે છે. એક વખત કમાન્ડ મોડમાં આવ્યા પછી ડોક્યુમેન્ટની અંદર એક જગ્યાએથી બીજી જગ્યાએ જવા માટે આપણે એરો કીનો ઉપયોગ કરી શકીએ છીએ. કોષ્ટક 7.4માં ડોક્યુમેન્ટમાં નેવિગેટ કરવા માટે વપરાતાં કી-સ્ટ્રોકની યાદી આપેલી છે.

કમાન્ડ	ઉપયોગ
h	કર્સર ડાબી તરફ ખસેડવા માટે
l	કર્સર જમણી તરફ ખસેડવા માટે
j	કર્સર નીચેની લીટીમાં લઈ જવા માટે
k	કર્સર ઉપરની લીટીમાં લઈ જવા માટે
Spacebar	કર્સરને એક જગ્યા જેટલું જમણી તરફ લઈ જવા માટે

-/+ Keys	કર્સરને નીચે કે ઉપરના પ્રથમ સ્તંભમાં ખસેડવા માટે
CTRL + d	અડધું પાનું નીચેની તરફ ખસેડવા (scroll) માટે
CTRL + u	અડધું પાનું ઉપરની તરફ ખસેડવા (scroll) માટે
CTRL + f	એક પાનાં જેટલું આગળ વધવા માટે
CTRL + b	એક પાનાં જેટલું પાછળ જવા માટે
M	કર્સરને પાનાંની મધ્યમાં લઈ જવા માટે
H	કર્સરને પાનાંની ઉપરના ભાગમાં લઈ જવા માટે
L	કર્સરને પાનાંની નીચેના ભાગમાં લઈ જવા માટે
\$	કર્સરને લીટીના અંતમાં લઈ જવા માટે
)	કર્સરને પછીના વાક્યની શરૂઆતમાં લઈ જવા માટે
(કર્સરને વર્તમાન વાક્યની શરૂઆતમાં લઈ જવા માટે
G	કર્સરને ફાઈલના અંતમાં લઈ જવા માટે
W	કર્સરને પછીના શબ્દ પર લઈ જવા માટે
Nw	કર્સરને શબ્દો N જેટલું આગળ લઈ જવા માટે
b	કર્સરને પહેલાના શબ્દ પર લઈ જવા માટે
Nb	કર્સરને શબ્દો N જેટલું પાછળ લઈ જવા માટે
e	કર્સરને શબ્દના અંતમાં લઈ જવા માટે
gg	કર્સરને ફાઈલની પ્રથમ લાઈનમાં લઈ જવા માટે
0	કર્સરને લીટીની શરૂઆતમાં લઈ જવા માટે

કોષ્ટક 7.4 : ફાઈલમાં નેવિગેટ કરવા માટેની વિવિધ કી

ઉપર જણાવેલા વિવિધ કી-સ્ટ્રોકથી પરિચિત બનવા પ્રયત્ન કરો.

ઑક્યુમેન્ટમાં સુધારા કરવા (Editing the Document)

એક વખત ઑક્યુમેન્ટ તૈયાર થઈ જાય પછી ઑક્યુમેન્ટમાં સુધારા કરવા (એડિટ-Edit કરવું) એ વપરાશકર્તા દ્વારા કરવામાં આવતી એક સૌથી સામાન્ય પ્રક્રિયા છે. આપણી જરૂરિયાત પ્રમાણે ઑક્યુમેન્ટમાં કોઈ ચોક્કસ જગ્યાએ આપણે કોઈ માહિતી

સામેલ કરવી કે કાઢી નાખવી શક્ય છે. જો જરૂરી હોય તો કોઈ માહિતી (contents) બદલી પણ શકીએ છીએ તેમ અલગ-અલગ અક્ષરોને અપરકેસ કે લોઅરકેસમાં ફેરફાર પણ કરી શકીએ છીએ. જ્યારે આપણે ડોક્યુમેન્ટમાં સુધારા-વધારા કરતાં હોઈએ ત્યારે કમાન્ડ મોડ અને ઈન્સર્ટ મોડને ફેરવવાની જરૂરિયાત ઊભી થાય છે. કોષ્ટક 7.2માં દર્શાવેલાં વિવિધ કૅરેક્ટર્સ આપણને જુદા-જુદા ઈન્સર્ટના વિકલ્પો કેવી રીતે શરૂ કરવા તે જણાવે છે. ચાલો, હવે આપણે about_Gandhiji ફાઈલમાં નીચે ખાનામાં આપેલ માહિતી સામેલ કરવા પ્રયત્ન કરીએ :

His mother, Putlibai, was a very religious lady and used to tell him stories from the scriptures and mythology.

Little Gandhi grew up to be an honest and a decent student. At the age of 13 he was married to Kasturba.

ડોક્યુમેન્ટમાં સુધારા કરવા માટે આપણે vi આદેશ આપીને ડોક્યુમેન્ટ ફરી ઓપન કરવું પડશે, આથી **vi about_Gandhiji** આદેશનો ફરી વાર અમલ કરો. તમે નિરીક્ષણ કરશો કે ડાબી બાજુએ સૌથી ઉપર પહેલા અક્ષર પર બ્લીન્કિંગ કર્સર દેખાય છે. સામાન્ય સંજોગોમાં ઈન્સર્ટ મોડમાં જવા માટે આપણે ‘I’ વિકલ્પનો ઉપયોગ કર્યો હોત પણ આપણે ફાઈલના અંતમાં માહિતી ઉમેરવાની જરૂરિયાત છે.

આથી, G ટાઈપ કરો અને તમે જોઈ શકશો કે કર્સર છેલ્લી લીટી ઉપર ગોઠવાયેલું છે. કર્સરની જગ્યાનો આધાર અગાઉ ફાઈલ કેવી રીતે સેવ કરી હતી, તેના ઉપર છે. જો ફાઈલ સેવ કરતાં પહેલાં Enter કી દબાવેલી હશે, તો G ટાઈપ કરવાથી કર્સર છેલ્લી લાઈન પછીની નવી લાઈનમાં ગોઠવાશે અને જો ફાઈલ સેવ કરતાં પહેલાં Enter કી ન દબાવેલી હોય, તો G ટાઈપ કરવાથી છેલ્લી લાઈનમાં પ્રથમ અક્ષર ઉપર કર્સર મૂકવામાં આવશે.

જો કર્સર નવી લાઈનમાં મૂકેલું હોય તો ESC કી દબાવીને આપણે ‘I’ દબાવીશું અને પછી માહિતી ટાઈપ કરવાનું શરૂ કરીશું. જો કર્સર છેલ્લી લાઈનનાં પ્રથમ અક્ષર ઉપર હોય તો ‘o’ દબાવો. આ પગલાથી કર્સરને નવી લાઈનમાં લઈ જશે. હવે માહિતી ટાઈપ કરવાનું શરૂ કરો અને ટાઈપ કરતાં સમયે કોઈ ભૂલ થાય, તો ભૂલ સુધારવા માટે બેકસ્પેસ કી નો ઉપયોગ કરી એકસાથે કર્સરની એક-એક જગ્યા પાછળ જઈ સુધારો. એક વખત સુધારાનું કાર્ય પૂર્ણ થાય પછી ESC કી દબાવો અને લાસ્ટ લાઈન મોડમાં જવા :wq ટાઈપ કરો, આથી ફાઈલ સેવ થશે અને વિમ સેશન પૂરું થશે. આકૃતિ 7.4માં નવી માહિતી ઉમેર્યા પછીનો ઓરિટરનો દેખાવ દર્શાવ્યો છે.

```
File Edit View Terminal Help
Mahatma Gandhi was born on 2nd October 1869, Gujarat.
His name was Mohandas Karamchand Gandhi.
His mother, Putlibai was a very religious lady and used to tell him stories
from the scriptures and mythology.
Little Gandhi grew up to be an honest and a decent student. At the age of
13 he was married to Kasturba.
~
~
:wq
```

આકૃતિ 7.4 : હયાત ફાઈલમાં ડેટા ઉમેરવા

કોઈ પણ ડોક્યુમેન્ટ ઉપર વિવિધ પ્રકારના સુધારાના કાર્યો માટેના આદેશો કોષ્ટક 7.5માં આપેલા છે, તે આપણે વાપરી શકીએ.

કમાન્ડ	ઉપયોગ
u	અંતિમ સુધારો રદ કરવા માટે (undo)
U	લીટીમાં કરવામાં આવેલા તમામ સુધારા રદ કરવા માટે
dd	એક લીટી દૂર કરવા માટે
Ndd	N સંખ્યામાં લીટીઓ દૂર કરવા માટે
D	કર્સર પછી આવેલું લીટીનું તમામ લખાણ દૂર કરવા માટે
C	કર્સર પછી આવેલું લીટીનું તમામ લખાણ દૂર કરી નવું લખાણ ઉમેરવા માટે. લખાણ ઉમેરવાનું પૂરું કરવા ESC-કી દબાવો.
dw	એક શબ્દ દૂર કરવા માટે
Ndw	N સંખ્યામાં શબ્દો દૂર કરવા માટે
cw	શબ્દ બદલવા માટે
x	કર્સરની નીચે આવેલો એક અક્ષર દૂર કરવા માટે
X	કર્સરની પહેલાં આવેલો એક અક્ષર દૂર કરવા માટે (બેક સ્પેસ)
r	એક અક્ષર બદલવા માટે
R	કર્સર પછી આવેલા અક્ષરો દૂર કરી નવાં લખાણ ઉમેરવા માટે (ઓવર રાઈટ)
s	કર્સરની નીચે રહેલો એક અક્ષર બદલી લખાણ ઉમેરવાનું ચાલુ રાખવા માટે
S	આખી લીટીને બદલી લીટીની શરૂઆતમાં લખાણ ઉમેરવા માટે
~	એક અક્ષરનો 'કેસ' (case) બદલવા માટે
.	છેલ્લે આપવામાં આવેલો કમાન્ડ પુનરાવર્તિત કરવા માટે

કોષ્ટક 7.5 : સુધારા-વધારા કરવા માટેના આદેશો (એડિટિંગ કરવા માટેના કમાન્ડ્સ)

કોષ્ટક 7.5માં આપેલાં આદેશો ઉપરાંત વિમ એડિટર આપણી ફાઈલમાંથી માહિતીની નકલ હંગામી બફરમાં અને હંગામી બફરમાંથી ફાઈલમાં કરવાની પરવાનગી આપે છે. દરેક બફર એક હંગામી મેમરી તરીકે કામ કરે છે, જે સામાન્ય રીતે 'ક્લિપબોર્ડ' તરીકે ઓળખાય છે. કોષ્ટક 7.6માં ડેટા મેળવવા અને ડેટા જોડવા (Data capturing and pasting) માટેના કેટલાક આદેશોની યાદી આપેલી છે.

કમાન્ડ	ઉપયોગ
yy	કર્સર ધરાવતી વર્તમાન લીટીની નકલ બફર (Buffer)માં કરવા માટે
Nyy	કર્સરના વર્તમાન સ્થાનેથી N સંખ્યામાં રહેલી લીટીઓની બફરમાં નકલ કરવા માટે
p	બફરમાં આવેલી લીટીઓને કર્સરના વર્તમાન સ્થાનની નીચે ઉમેરવા (paste) માટે

કોષ્ટક 7.6 : ડેટા મેળવવા અને જોડવા માટેના આદેશો (કમાન્ડ્સ ટુ કૅપ્ચર એન્ડ પેસ્ટ)

માહિતી શોધવી અને બદલવી (Searching and replacing text)

ડોક્યુમેન્ટમાં માહિતી શોધવી અને તેને બદલવી એ વપરાશકર્તા દ્વારા કરવામાં આવતી એક અન્ય સામાન્ય ક્રિયા છે. ફાઇલમાં કોઈ શબ્દિક માહિતી કે પદાવલિ શોધવા માટે વિમ એડિટર વિશિષ્ટ આદેશો વાપરવાની પરવાનગી આપે છે. આપણે આદેશ વાપરીને એક શબ્દની જગ્યાએ બીજો શબ્દ પણ રાખી શકીએ છીએ. કોષ્ટક 7.7માં ફાઇલમાં માહિતી શોધવા અથવા શોધીને બદલવાની ક્રિયા કરવામાં ઉપયોગી વિવિધ આદેશોની યાદી આપેલી છે :

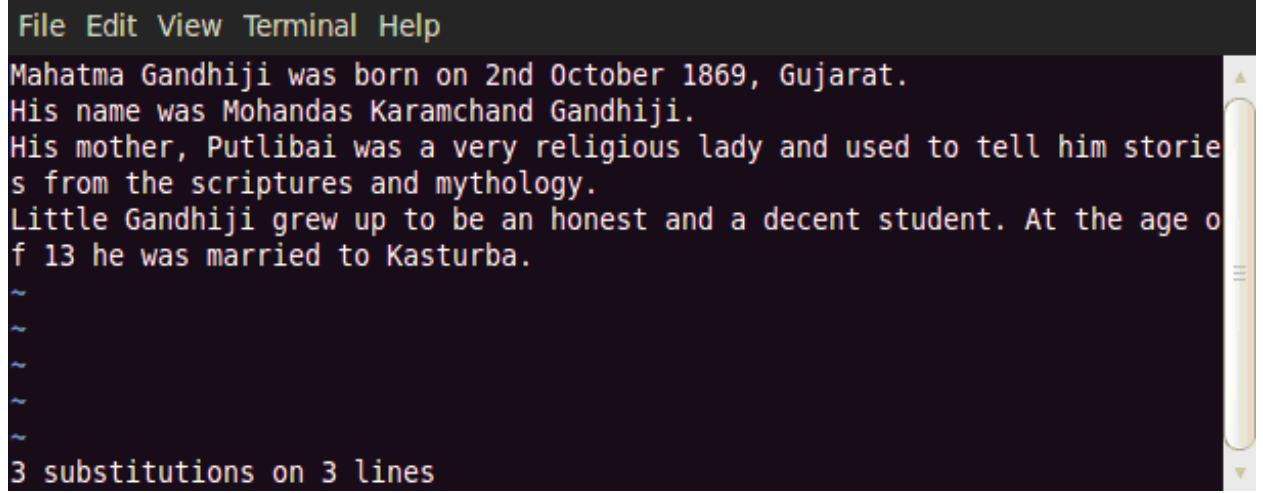
કમાન્ડ	ઉપયોગ
/	આગળની દિશામાં (forward) લખાણની શોધ કરવા માટે
?	પાછળની દિશામાં (backward) લખાણની શોધ કરવા માટે
n	ફરી તે જ દિશામાં શોધ કરવા માટે
SHIFT + n	ફરી વિરુદ્ધ દિશામાં શોધ કરવા માટે
f	f-કી દબાવી શોધ માટેનો અક્ષર ટાઈપ કરો. વર્તમાન લીટીમાં આવેલા તે અક્ષર ઉપર કર્સર ગોઠવાશે.
SHIFT + f	f જેવું જ કાર્ય કરે છે પરંતુ વિરુદ્ધ દિશામાં
t	f જેવું જ કાર્ય કરે છે પરંતુ કર્સરને શોધ માટેના અક્ષરની પહેલા ગોઠવશે.
SHIFT + t	t જેવું જ કાર્ય કરે છે, પરંતુ વિરુદ્ધ દિશામાં
:s/old_string/new_string	વર્તમાન લીટીમાં સૌપ્રથમ આવેલ old_string ને new_string સાથે બદલશે.
:s/old_string/new_string/g	વર્તમાન લીટીમાં તમામ old_string ને new_string સાથે બદલશે.
:%s/old_string/new_string/g	આખી ફાઇલમાં આવેલ તમામ old_string ને new_string સાથે બદલશે.
:%s/old_string/new_string/gc	આખી ફાઇલમાં આવેલ તમામ old_string ને new_string સાથે બદલશે. પરંતુ, દરેક શબ્દ બદલતા પહેલાં પુષ્ટિ (confirmation) માંગશે.

કોષ્ટક 7.7 : શોધવું અને બદલવું ક્રિયાઓ કરવા માટેના આદેશો

ચાલો, આપણે આમાંના કેટલાક આદેશો about_Gandhiji ફાઇલમાં વાપરવાનો પ્રયત્ન કરીએ. ધારોકે આપણે દરેક જગ્યાએ આવતો શબ્દ “Gandhi” ને “Gandhiji” શબ્દ વડે બદલવા ઇચ્છીએ છીએ. આ કાર્ય કરવા માટે આપણે સૌપ્રથમ vi about_Gandhiji આદેશ આપીને ફાઇલ ખોલવી પડશે, તે પછી ESC કી દબાવીને લાસ્ટ લાઈન મોડમાં ગયા પછી નીચે આપેલા આદેશનો અમલ કરો :

:%s/Gandhi/Gandhiji/g

અહીં, %s જણાવે છે કે આપણે કોઈ સ્ટ્રિંગ (string - અક્ષરોના સમૂહની હાર)ને બદલાવવાનો પ્રયત્ન કરી રહ્યા છીએ. “Gandhi” પદ જૂની સ્ટ્રિંગનો નિર્દેશ કરે છે કે જેને આપણે બદલવાની છે અને “Gandhiji” પદ નવી સ્ટ્રિંગનો નિર્દેશ કરે છે. અહીં આપેલો વિકલ્પ “g” જણાવે છે કે આખી ફાઈલમાં દરેક જગ્યાએ આવતાં પદ “Gandhi” ને “Gandhiji” પદ વડે બદલવાના છે. આ આદેશનો આઉટપુટ આકૃતિ 7.5માં દર્શાવ્યો છે. અહીં અવલોકન કરશો કે કેટલી જગ્યાએ બદલ્યાં છે તે સંખ્યા પણ અહીં દર્શાવે છે. જો આપણે આ પરિવર્તનફાઈલમાં જોઈતા હોય તો ફાઈલનો સંગ્રહ કરવો પડશે. જો આપણે સંગ્રહ કરવા માટે સેવ કર્યા વગર છોડી જઈશું, તો ફાઈલમાં આ પરિવર્તન દેખાશે નહીં.



```
File Edit View Terminal Help
Mahatma Gandhiji was born on 2nd October 1869, Gujarat.
His name was Mohandas Karamchand Gandhiji.
His mother, Putlibai was a very religious lady and used to tell him stories
from the scriptures and mythology.
Little Gandhiji grew up to be an honest and a decent student. At the age of
13 he was married to Kasturba.
~
~
~
~
~
3 substitutions on 3 lines
```

આકૃતિ 7.5 : શોધવું અને બદલવાની ક્રિયા

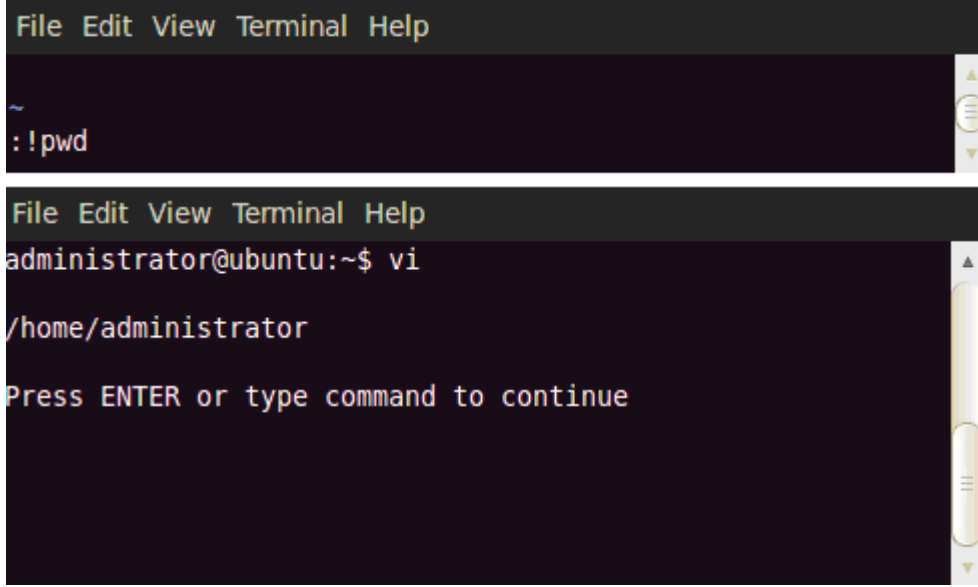
વિમ વડે લિનક્સ આદેશોનો અમલ (Executing Linux commands through Vim) :

વિમ એડિટર માંથી જ લિનક્સના આદેશોનો અમલ કરવાનું પણ શક્ય છે. કોઈ પણ લિનક્સ આદેશનો અમલ કરવા માટે આપણે આદેશ પહેલાં ઉદ્ગારચિહ્ન (!) ટાઈપ કરવાની જરૂર રહે છે.

ઉદાહરણ તરીકે, જે ડિરેક્ટરીમાં આપણે કામ કરી રહ્યાં છીએ તે વર્તમાન ડિરેક્ટરી આપણે જોવાં ઇચ્છીએ છીએ, તો નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરો :

- કમાન્ડ પ્રોમ્પ્ટ ઉપર vi ટાઈપ કરીને વિમ એડિટર ખોલો.
- ‘ESC:’ દબાવીને લાસ્ટ લાઈન મોડમાં જાઓ.
- હવે r!pwd ટાઈપ કરો.
- એન્ટર કી દબાવો.

આથી, આપણે વર્તમાન ડિરેક્ટરી જોઈ શકીશું. આકૃતિ 7.6માં કરેલું કાર્ય અને તેનો આઉટપુટ દર્શાવ્યો છે.



```
File Edit View Terminal Help
~
:!pwd

File Edit View Terminal Help
administrator@ubuntu:~$ vi
/home/administrator
Press ENTER or type command to continue
```

આકૃતિ 7.6 : વિમ એડિટરમાં લિનક્સ આદેશ

આ જ પ્રમાણે જો આપણે ફાઇલમાં કર્સર જે જગ્યાએ છે, તેની પછીની લીટીમાં આજની તારીખ ઉમેરવી હોય, તો નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરો :

- વિમ એડિટરમાં ફાઇલ ખોલો.
- ‘ESC:’ દબાવીને લાસ્ટ લાઇન મોડમાં જાઓ.
- હવે r!date ટાઇપ કરો.
- એન્ટર કી દબાવો.

અહીં r વિકલ્પ આપણને ફાઇલ કે બફરમાં ડેટા ઉમેરવાની પરવાનગી આપે છે.

શેલસ્ક્રિપ્ટ (Shell Script)

આપણે કમાન્ડ પ્રોમ્પ્ટ ઉપરથી તેમજ વિમ એડિટર માંથી આદેશનો અમલ કઈ રીતે કરી શકાય તે જોયું. આ બન્ને રીતમાં આપણને એક સમયે ફક્ત એક જ આદેશનો અમલ કરવાની પરવાનગી મળે છે. જ્યારે શેલસ્ક્રિપ્ટમાં ઘણી સારી રીતે એક કરતાં વધારે આદેશોનો અમલ કરવાની આપણને સગવડતા મળે છે. આથી, જ્યારે પણ આ કાર્ય કરવાનું હોય, ત્યારે દરેક સમયે કમાન્ડ પ્રોમ્પ્ટ ઉપર આદેશ ટાઇપ કરીને સમય બગાડવા કરતાં શેલસ્ક્રિપ્ટ બનાવીને આપણે એક લાઇનનો આદેશ ટાઇપ કરીને આદેશોની શ્રેણીનો અમલ કરી શકીએ.

શેલસ્ક્રિપ્ટને એ રીતે વ્યાખ્યાયિત કરી શકાય કે “આદેશોના સમૂહ લખેલી એક ટેક્સ્ટ ફાઇલ કે જે નીમેલું કાર્ય (designated task) નિયંત્રિત રીતે કરે” શેલસ્ક્રિપ્ટની રચના પારસ્પરિક ક્રિયા-પ્રતિક્રિયા કરી શકે તેવી (ઈન્ટરએક્ટિવ) પણ બનાવી શકાય; આવી સ્ક્રિપ્ટ વપરાશકર્તા પાસેથી ઈનપુટ લે અને આપેલા ઈનપુટ પ્રમાણે અલગ-અલગ કાર્યો કરે.

શેલસ્ક્રિપ્ટ બનાવવી અને તેનો અમલ કરવો (Creating and Executing a Shell Script)

આપણે કોઈ પણ ટેક્સ્ટએડિટર વડે શેલસ્ક્રિપ્ટ બનાવી શકીએ. આપણે વિમ એડિટર માં કામ કરતાં શીખ્યા હોવાથી આપણે તેમાં સ્ક્રિપ્ટ બનાવીશું. ચાલો, આપણે એક નાની શેલસ્ક્રિપ્ટ બનાવીએ, જે વપરાશકર્તાનું સ્વાગત કરે. નીચે ખાનામાં જણાવ્યા પ્રમાણે વિમ એડિટર માં ટાઇપ કરો અને તેનો *script1.sh*. નામથી સંગ્રહ કરો. ફાઇલના અનુલંબનમાં “sh”નો

ઉપયોગ એ મૂળભૂત રીતે ફાઇલ એક શેલસ્ક્રિપ્ટ છે, તેનો નિર્દેશ કરવા માટે કર્યો છે. પણ અહીં એ નોંધ કરશો કે ફાઇલનું અનુલંબન “sh” હોવું ફરજિયાત નથી અને અનુલંબન સિવાયની ફાઇલ પણ શેલસ્ક્રિપ્ટ તરીકે વાપરી શકાય. પણ ફાઇલને અનુલંબન આપવું એ એક હંમેશની સારી આદત છે કારણકે તે આપણને એક સામાન્ય ફાઇલ અને શેલસ્ક્રિપ્ટ ફાઇલને ઓળખવામાં મદદ કરે છે.

#Script 1: Script to welcome the user who has logged into the system

```
clear
```

```
echo Hello
```

```
who am i
```

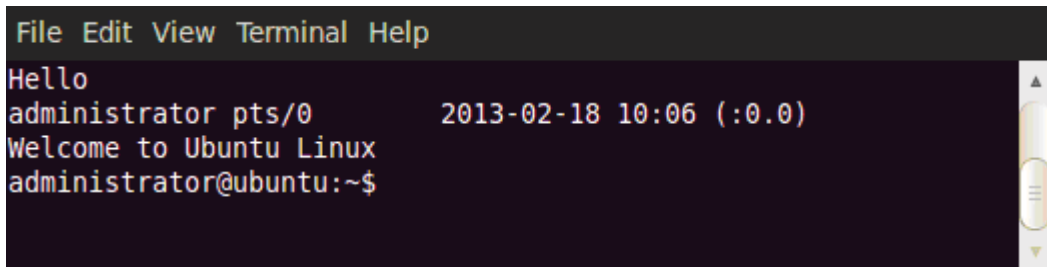
```
echo Welcome to Ubuntu Linux
```

અહીં સ્ક્રિપ્ટની પ્રથમ લાઇનનું અવલોકન કરો; કે “#” ચિહ્ન સાથે શરૂ થાય છે. કોઈ પણ લાઇન “#” ચિહ્ન સાથે શરૂ થાય તે એક કોમેન્ટ (comment) તરીકે ગણવામાં આવે છે. સ્ક્રિપ્ટના ભાગ તરીકેની કોમેન્ટનો અમલ થતો નથી; તે ફક્ત એક સંદેશ છે જે વપરાશકર્તાને સ્ક્રિપ્ટનો ઉપયોગ અને તેનો અર્થ સમજવામાં મદદ કરે છે. બીજી લાઇનમાં એક આદેશ (કમાન્ડ) છે જે સ્ક્રિપ્ટનો આઉટપુટ આપતાં પહેલાં સ્ક્રીન ઉપરની માહિતી ભૂંસી નાખે છે. ત્રીજી લાઇન એક સંદેશ “Hello” પ્રદર્શિત કરે છે, જ્યારે ચોથી લાઇન “who am i” આદેશનો અમલ કરે છે અને તે સમયે સિસ્ટમનો ઉપયોગ કરતાં વપરાશકર્તાનું નામ અને થોડી વધારાની વિગત આપશે. છેલ્લી લાઇન ફરી એક સંદેશ “Welcome to Ubuntu Linux” પ્રદર્શિત કરશે.

સ્ક્રિપ્ટનો અમલ કરવા માટે આપણે *sh* અથવા *bash* આદેશનો ઉપયોગ કરવો પડે. જો સ્ક્રિપ્ટ વર્તમાન ડીરેક્ટરીમાં હોય, તો નીચે જણાવ્યા પ્રમાણે આદેશ ટાઇપ કરો:

\$bash script1.sh અથવા \$sh script1.sh

ઘણી વખત આપણી સામે ફાઇલ પ્રવેશના હક્કો (file privileges) સંબંધિત મુદ્દાઓ આવે છે. સ્ક્રિપ્ટનો અમલ કરવા માટે સ્પષ્ટ રીતે અમલ કરવાની પરવાનગી સેટ કરેલી જરૂરી છે. કદાચ જો આવી કોઈ સમસ્યા થાય, તો આપણે *chmod* આદેશ વાપરીને જરૂરી વિશેષાધિકારો સેટ કરવા પડે. ઉદાહરણ તરીકે, *chmod +x script1.sh* આદેશ આપવાથી ફાઇલનો અમલ કરી શકાય તેવી (એક્ઝિક્યુટેબલ) બનાવી શકાય છે. જો બધું બરાબર ચાલે, તો આકૃતિ 7.7માં દર્શાવ્યા મુજબનો આઉટપુટ આપણે મેળવીશું.



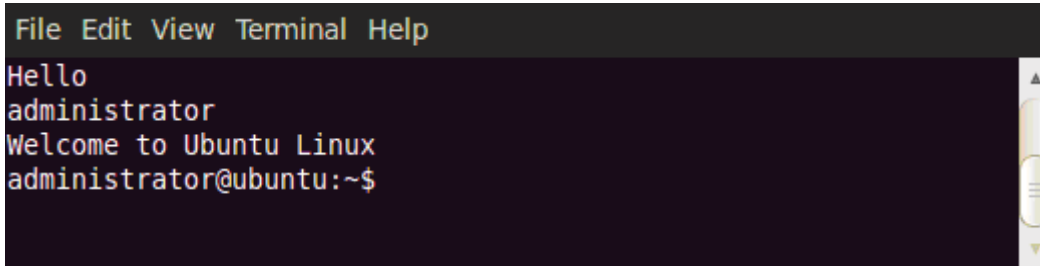
```
File Edit View Terminal Help
Hello
administrator pts/0      2013-02-18 10:06 (:0.0)
Welcome to Ubuntu Linux
administrator@ubuntu:~$
```

આકૃતિ 7.7 : Script 1નો આઉટપુટ

આકૃતિ 7.7માં અવલોકન કરશો કે વપરાશકર્તાના નામ સાથે કેટલીક વધારાની માહિતી પણ આપણે મેળવીએ છીએ. ચાલો, આપણે આપણા ફિલ્ટર (filter) અંગેના જ્ઞાનનો ઉપયોગ કરીને વધારાની માહિતી દૂર કરવાનો પ્રયત્ન કરીએ. ફેરફાર કરેલી સ્ક્રિપ્ટ નીચે મુજબ છે :

```
#Script 2: Modified script to welcome the user who has logged into the system
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
```

અહીં તમે જોશો કે *who am i* આદેશ સાથે આપણે *cut* ફિલ્ટરનો ઉપયોગ કર્યો છે. આ ઉપરાંત આપણે પાઇપ (pipe) નો ઉપયોગ કરીને બે આદેશને જોડી દીધા છે. *echo* આદેશ પછી ડબલ અવતરણચિહ્નોમાં મૂકેલી માહિતીને એક સંદેશ તરીકે ન ગણવામાં આવે તેની ખાતરી કરવા માટે આપણે ઊલટાં અવતરણચિહ્નો (' ') વચ્ચે તેમને સમાવી લીધી છે. ઊલટા અવતરણચિહ્ન કી-બોર્ડ ઉપર ~ ચિહ્ન સાથેની કી ઉપર પ્રિન્ટ કરેલ છે. વિમ એડિટર વડે સુધારેલી સ્ક્રિપ્ટ ટાઇપ કરો અને તેને *script2.sh* ના નામથી સંગ્રહ (સેવ) કરો. હવે આ સ્ક્રિપ્ટનો અમલ કરો અને તમે જુઓ કે જે માહિતી આપણે ઇચ્છતા હતા તે જ માહિતી ફક્ત આપણે જોઈ શકીએ છીએ. આકૃતિ 7.8માં સુધારેલી સ્ક્રિપ્ટનો આઉટપુટ દર્શાવ્યો છે.



```
File Edit View Terminal Help
Hello
administrator
Welcome to Ubuntu Linux
administrator@ubuntu:~$
```

આકૃતિ 7.8 : Script 2નો આઉટપુટ

ચાલો, હવે આજની તારીખ અને સમય પ્રદર્શિત કરવા માટે સ્ક્રિપ્ટને વધારે સુધારીએ. સ્ક્રિપ્ટ નીચેના ખાનામાં આપેલી છે.

```
#Script 3: Script to welcome the user and display login date and time
clear
echo Hello
echo "`who am i | cut -d " " -f 1`"
echo Welcome to Ubuntu Linux
echo The current date and time is
date
```

ચાલો, હવે આજની તારીખ અને સમય પ્રદર્શિત કરવા માટે સ્ક્રિપ્ટને વધારે સુધારીએ. સ્ક્રિપ્ટ નીચેના ખાનામાં આપેલી *script3.sh* નામની ફાઇલ બનાવો અને તેમાં *script3*ની માહિતી ટાઇપ કરો. તે પછી તે સ્ક્રિપ્ટનો અમલ કરો અને આઉટપુટ જુઓ. એક વાર આપણને શેલસ્ક્રિપ્ટ વાપરવી અનુકૂળ આવી જાય પછી પુનરાવર્તિત કાર્યો કરવામાં આપણને તે ખૂબ મદદરૂપ જણાય છે.

શેલસ્ક્રિપ્ટના ચલ (Shell Script Variables) :

શેલસ્ક્રિપ્ટ લખવાની ક્રિયા લગભગ ઉચ્ચ સ્તરની ભાષામાં પ્રોગ્રામ લખવાની ક્રિયા જેવી જ છે. ઉચ્ચસ્તરના પ્રોગ્રામિંગની એક સૌથી સામાન્ય લાક્ષણિકતા ચલ (variables)ની જોગવાઈ છે. જે રીતે નામ સૂચવે છે, તે પ્રમાણે ચલ એક એન્ટિટી (entity) છે કે જેમાં આપણે કોઈ કિંમતનો સંગ્રહ કરી શકીએ અથવા તેને સુધારી શકીએ છીએ. વપરાશકર્તાની જરૂરિયાત

પ્રમાણે ચલમાં સંગ્રહ કરેલી કિંમતનો ફરી વાર ઉપયોગ પણ કરી શકાય અને કિંમત બદલી પણ શકાય. અન્ય પ્રોગ્રામિંગ ભાષાના ચલની જેમ શેલસ્ક્રિપ્ટના ચલ એ શેલસ્ક્રિપ્ટિંગનું અભિન્ન અંગ છે. શેલસ્ક્રિપ્ટમાં જ્યારે ચલ વાપરવામાં આવે છે, ત્યારે આપણે તેને કોઈ કિંમત આપી શકીએ છીએ અથવા વપરાશકર્તા પાસેથી તેની કિંમત મેળવે છે. echo આદેશ વાપરીને આપણે તેની કિંમત પ્રદર્શિત પણ કરી શકીએ છીએ. ચાલો, આપણે ચલનો ઉપયોગ બતાવતી એક નાની સ્ક્રિપ્ટ લખીએ. નીચેના ખાનામાં સ્ક્રિપ્ટનો કોડ (code) લખેલો છે.

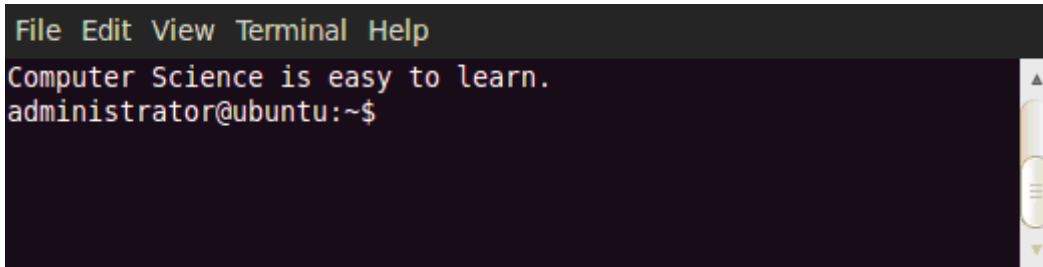
```
# Script 4: Shell script to show use of variables
```

```
clear
```

```
subject="Computer Science"
```

```
echo $subject is easy to learn.
```

script4.sh નામની ફાઈલમાં script 4ની માહિતી ટાઈપ કરો. આપણે સંગ્રહ કરેલી સ્ક્રિપ્ટને સમજવા પ્રયત્ન કરીએ. અન્ય બધી સ્ક્રિપ્ટની જેમ પહેલી લાઈન એ કોમેન્ટ છે. બીજી લાઈનના આદેશ વડે સ્ક્રીન ઉપરની અગાઉની માહિતી ભૂંસાઈ જાય છે. ત્રીજી લાઈનમાં આપણે એક ચલ વ્યાખ્યાયિત કરેલો છે, જેનું નામ subject છે અને એક સાદા એસાઈમેન્ટ ઓપરેટર (assignment operator) વડે તે ચલની કિંમત “Computer Science” બનાવી છે. સ્ટ્રિંગના બે શબ્દો વચ્ચે ખાલી જગ્યા (white space) હોવાથી આપણે તેને ડબલ અવતરણચિહ્ન વચ્ચે મૂકેલી છે. ચોથી લીટી સ્ક્રીન ઉપર એક સંદેશ પ્રદર્શિત કરે છે. subject ચલ પહેલાનું ચિહ્ન ‘\$’ શેલને સૂચના આપે છે કે ચલમાં સંગ્રહ કરેલી કિંમત અથવા ચલને સોંપેલી કિંમત મેળવવાની (extract) છે. આ સ્ક્રિપ્ટનો અમલ કરવાથી મળતો આઉટપુટ આકૃતિ 7.9માં દર્શાવ્યો છે.



```
File Edit View Terminal Help
Computer Science is easy to learn.
administrator@ubuntu:~$
```

આકૃતિ 7.9 : Script 4નો આઉટપુટ

વપરાશકર્તાએ ખાસ ધ્યાન રાખવું જોઈએ કે કોઈ કિંમત આપતાં સમયે (assigning) બરાબરના ચિહ્ન (=) ની બન્ને બાજુએ કોઈ ખાલી જગ્યા (સ્પેસ) રહેવી ન જોઈએ. કોઈ પણ કારણસર બન્નેમાંથી કોઈ બાજુએ જો સ્પેસ રહી જાય તો શેલ તે સ્ટ્રિંગની સ્પેસ પછી આદેશ છે, તેમ અર્થઘટન કરશે. તે અપેક્ષિત પરિણામ ન પણ આપે. ત્રીજી લાઈનનું વિધાન subject="Computer Science" એ પહેલાં *subject* નામનું ચલ બનાવશે અને પછી તેને “Computer Science” કિંમત આપશે (assign કરશે). જો આ ચલને આપણે ફરી વાપરશું, તો તેમાં રહેલી કિંમત ભૂંસાઈ જશે અને તેના ઉપર નવી કિંમત લખાઈ જશે (overwritten). ચાલો, આપણે છેલ્લું વિધાન એક સાદી સ્ક્રિપ્ટ લખીને સમજવાનો પ્રયત્ન કરીએ, જે નીચે ખાનામાં લખેલી છે :

```
# Script 5: Shell script to show use of variables
```

```
clear
```

```
subject="Computer Science"
```

```
echo $subject is easy to learn.
```

```
subject="Economics"
```

```
echo $subject is easy to learn.
```

આ સ્ક્રિપ્ટને script5.sh નામથી સંગ્રહ કરો અને તેનો અમલ કરી તેના આઉટપુટનું નિરીક્ષણ કરો.

આપણે શેલસ્ક્રિપ્ટમાં જ્યારે કોઈ ચલ વ્યાખ્યાયિત કરીએ છીએ, ત્યારે આપણે નીચે જણાવેલા નિયમોને અનુસરવા પડે.

- કોઈ પણ ચલનું નામ મૂળાક્ષરો, અંક અથવા અન્ડરસ્કોર (_)ના સંયોજનથી બનેલું હોવું જોઈએ.
- ચલના નામમાં અન્ડરસ્કોર સિવાય અન્ય કોઈ વિશિષ્ટ અક્ષર (special character) ન હોવો જોઈએ.
- ચલના નામનો પ્રથમ અક્ષર કોઈ મૂળાક્ષર કે અન્ડરસ્કોર હોવો જોઈએ.

નોંધ

જો શેલ કોઈ શબ્દને ચલ તરીકે ઓળખી ન શકે, તો તેને તે લિનક્સનો આદેશ માની લે છે.

વપરાશકર્તા સાથેની ક્રિયા-પ્રતિક્રિયા અને શેલસ્ક્રિપ્ટ (User Interaction and Shell Script)

અગાઉના ઉદાહરણમાં તમે કદાચ નિરીક્ષણ કર્યું હશે કે સ્ક્રિપ્ટની અંદર જ આપણે ચલની કિંમત એસાઈન કરેલી છે. જ્યારે ચલનો આ રીતે ઉપયોગ કરીએ છીએ, ત્યારે તેની ખાસ અગત્યતા હોતી નથી. સામાન્ય રીતે ચલ ત્યારે વાપરવામાં આવે છે કે જ્યારે તે વધારાની કેટલીક ક્રિયાઓમાં વાપરવામાં આવે અને જરૂર પડે તો તેની કિંમતમાં ફેરફાર પણ થાય. આ લાક્ષણિકતા ફક્ત ત્યારે જ મેળવી શકાય કે જ્યારે આપણે વપરાશકર્તા પાસેથી ચલની કિંમત મેળવવા સમર્થ હોઈએ. *read* આદેશ વપરાશકર્તા પાસેથી સ્ટાન્ડર્ડ ઈનપુટ એકમ ઉપરથી ડેટાટાઈપ કરવાની અપેક્ષા રાખે છે, ટાઈપ કરેલી બધી માહિતી ચલને આર્ગ્યુમેન્ટ (argument) તરીકે સંગ્રહ કરવા આપે છે. ચાલો, હવે વપરાશકર્તા પાસેથી વિષયનાં નામ લેવા માટે આપણે script 5ને ફરી લખીએ. નીચે ખાનામાં આપેલા કોડનો *script6.sh*ના નામથી સંગ્રહ કરો.

```
# Script 6: Shell script to accept value of variable from user
```

```
clear
```

```
echo -n "Enter your name: "
```

```
read name
```

```
echo -n "Enter name of a subject: "
```

```
read subject
```

```
echo Hello I am $name and I am studying $subject.
```

આપણે જ્યારે આ સ્ક્રિપ્ટનો અમલ કરીશું, ત્યારે તેનું પહેલું અમલમાં મૂકી શકાતું વિધાન (executable statement) સ્ક્રીન ઉપરની માહિતી ભૂંસી નાખશે. તે પછી સ્ક્રીન ઉપર “Enter your name:” સંદેશ પ્રદર્શિત કરશે, તે પછીનું વિધાન વપરાશકર્તા પાસેથી તેનું નામ લેવા માટે રાહ જોશે. એન્ટર કી દબાવવી એ ઈનપુટ માહિતીનો અંત સૂચવે છે, આથી સાવધ રહેજો કે નામ ટાઈપ થઈ જાય પછી જ એન્ટર કી દબાવજો. કંઈ પણ ટાઈપ કાર્ય વિના જો આપણે એન્ટર કી દબાવીશું તો સ્ક્રિપ્ટ ચલને નલ કિંમત (NULL value) સોંપશે અને પછીના આદેશ ઉપર જશે. એ પછીનાં બે

વિધાન પણ એ જ કાર્ય, સંદેશ પ્રદર્શિત કરવાનું અને વપરાશકર્તા પાસેથી વિષયનું નામ લેવા માટે રાહ જોવાનું કામ કરે છે. છેલ્લું વિધાન બંને ચલની કિંમત સાથે યોગ્ય સંદેશ પ્રદર્શિત કરે છે.

તમે નિરીક્ષણ કરો કે echo આદેશ સાથે આપણે -n વિકલ્પ વાપરેલો છે. આ વિકલ્પ echo આદેશને સૂચના આપે છે કે સંદેશ પ્રદર્શિત કર્યા પછી નવી લાઇનમાં જવાનું નથી. echo આદેશ પૂર્વનિર્ધારિત રીતે આર્ગ્યુમેન્ટ તરીકે સંદેશ પ્રદર્શિત કર્યા પછી નવી લાઇન ઉમેરી દે છે. આકૃતિ 7.10માં સ્ક્રિપ્ટનો આઉટપુટ દર્શાવ્યો છે.

```
File Edit View Terminal Help
Enter your name: Vidita
Enter name of a subject: Computer Science
Hello I am Vidita and I am studying Computer Science.
administrator@ubuntu:~$
```

આકૃતિ 7.10 : Script 6નો આઉટપુટ

આપણે બન્ને ચલની કિંમત વપરાશકર્તા પાસેથી મેળવતાં હોવાથી દરેક સમયે આપણે સ્ક્રિપ્ટનો અમલ કરીશું, ત્યારે વપરાશકર્તા જે કંઈ એન્ટર કરશે તે પ્રમાણે આઉટપુટ બદલાશે. ચાલો, હજી એક વધારે સ્ક્રિપ્ટ લખીએ કે જે વપરાશકર્તા પાસેથી ફાઇલનું નામ ઈનપુટ લે અને તે ફાઇલમાં કુલ કેટલી સંખ્યામાં લાઇન છે, તે પ્રદર્શિત કરે. આ માટેનો કોડ નીચે આપેલો છે, જેનો તમે *script7.sh* નામથી સંગ્રહ કરો.

```
#Script 7: Shell script to display total number of lines in a file

clear

echo -n "Enter a file name: "

read fname

echo "There are `cat $fname | wc -l` lines in a file $fname"
```

આ સ્ક્રિપ્ટનો અમલ કર્યા પછીનો આઉટપુટ આકૃતિ 7.11માં દર્શાવ્યો છે.

```
File Edit View Terminal Help
Enter a file name: script6.sh
There are 7 lines in a file script6.sh
administrator@ubuntu:~$
```

આકૃતિ 7.11 : Script 7નો આઉટપુટ

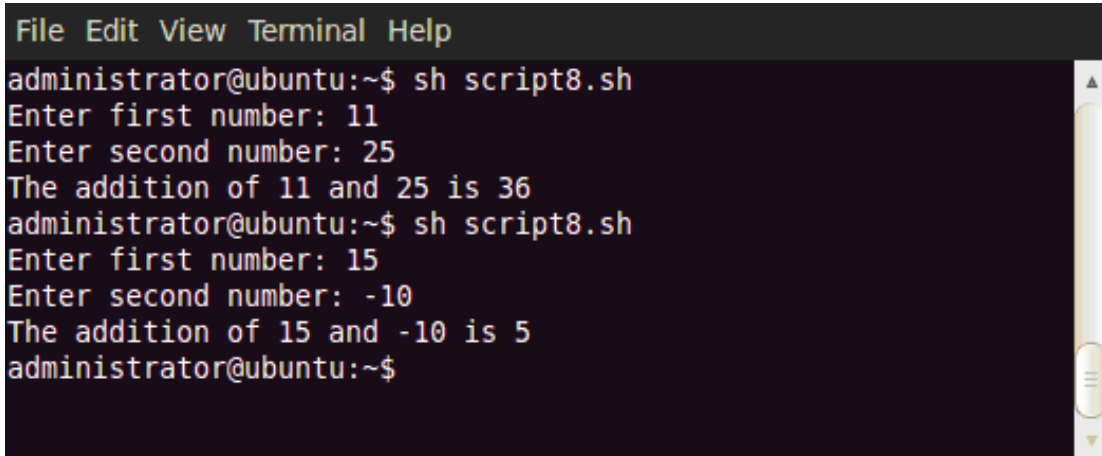
શેલ અંકગણિત (Shell Arithmetic)

આપણે અગાઉના વિભાગમાં ચલને કઈ રીતે વ્યાખ્યાયિત કરવા, તેને કિંમત એસાઈન કરવી અને તેમાં સંગ્રહ કરેલી કિંમત પાછી મેળવવી તે બાબત જોયું. જે કંઈ કિંમત એસાઈન કરી તે હજી સુધી તમામ સ્ટ્રિંગ (મૂળાક્ષરો, અંકો અથવા વિશિષ્ટ અક્ષરો) હતી. આપણે ચલને ફક્ત આંકડાકીય કિંમત પણ એસાઈન કરી શકીએ અને તેના ઉપર ગણતરી કરી શકીએ. ચાલો, આપણે એક સ્ક્રિપ્ટ લખીએ, જેમાં તે બે સંખ્યાનું ઇનપુટ લે અને આ બંને સંખ્યાનો સરવાળો કરે. આ માટેનો કોડ નીચેના ખાનામાં આપેલો છે.

```
# Script 8: Script to add two numbers
echo -n "Enter first number: "
read num1
echo -n "Enter second number: "
read num2
sum=`expr $num1 + $num2`
echo "The addition of $num1 and $num2 is $sum"
```

સ્ક્રિપ્ટ ટાઇપ કરી તેનો *script8.sh*. નામથી સંગ્રહ કરો. આ સ્ક્રિપ્ટમાં લખેલ પદ *expr*નો અર્થ expression થાય છે. અહીં નોંધ કરો કે સંકારક-ઓપરેટર (operator) (અહીં +) અને સંકાર્ય - ઓપરેન્ડ (operands) (અહીં \$num1, અને \$num2) વચ્ચે એક જગ્યા (સ્પેસ) હોવી જોઈએ. આ ઉપરાંત, એસાઈનમેન્ટ ઓપરેટર (=) ની પહેલાં અને પછી એક પણ જગ્યા (સ્પેસ) ન હોવી જોઈએ.

આકૃતિ 7.12માં સ્ક્રિપ્ટનો બે વખત અમલ કરીને બે જુદા-જુદા આઉટપુટ બતાવેલા છે. આપણે સ્ક્રિપ્ટમાં clear આદેશ વાપરેલો ન હોવાથી એક જ સ્ક્રીનમાં જુદા-જુદા આઉટપુટ જોઈ શકીએ છીએ.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script8.sh
Enter first number: 11
Enter second number: 25
The addition of 11 and 25 is 36
administrator@ubuntu:~$ sh script8.sh
Enter first number: 15
Enter second number: -10
The addition of 15 and -10 is 5
administrator@ubuntu:~$
```

આકૃતિ 7.12 : Script 8નો આઉટપુટ

આ ઉપરાંત આપણે બાદબાકી, ગુણાકાર, ભાગાકાર અને મોડ્યુલર ડિવિઝન અનુક્રમે -, *, / અને % પ્રક્રિયક વાપરીને કરી શકીએ છીએ. ગણિતના સામાન્ય નિયમો પ્રમાણે પદાવલિઓનું મૂલ્ય શોધવામાં આવે છે. જ્યારે એકસમાન અગ્રતાવાળા

પ્રક્રિયક (operator) વચ્ચે ટાઈ (tie - સરખાપણું) પડે, ત્યારે જે પ્રક્રિયક પહેલાં આવે તેને પસંદગી આપવામાં આવે છે. જ્યારે કોઈ ક્રિયા અન્ય કરતાં પહેલાં કરવાની ફરજ પાડવી હોય, ત્યારે આપણે તે ક્રિયાને કૌંસમાં લખીએ છીએ.

ઉદાહરણ તરીકે, $\$num1 \ * \ (\$num2 + \$num3 \) / \$num4$ પદાવલિમાં $\$num2 + \$num3$ નું મૂલ્ય સૌથી પ્રથમ કાઢવામાં આવે છે, કારણકે તે કૌંસમાં છે. નિરીક્ષણ કરશો કે આપણે "*" ચિહ્ન તેમજ ડાબો કૌંસ અને જમણો કૌંસ પહેલાં બેકસ્લેશ કેરેક્ટર (\) મૂકેલું છે.

નોંધ :

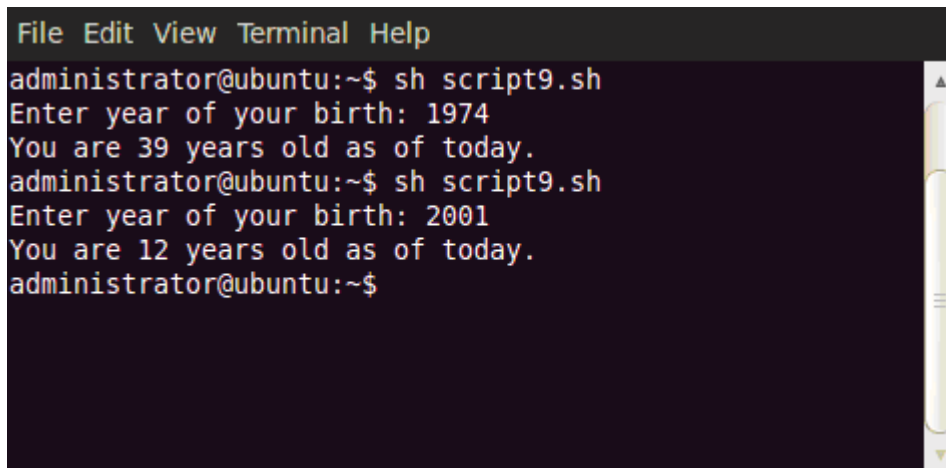
જ્યારે બે સંખ્યાઓનો ગુણાકાર શોધવો હોય ત્યારે આપણે ગુણાકારના ચિહ્ન (*) ના પૂર્વગ તરીકે બેકસ્લેશ (\) કેરેક્ટર મૂકવું પડે. નહીંતર શેલ (*) ચિહ્નને એક વાઈલ્ડકાર્ડ (wildcard) કેરેક્ટર તરીકે ગણતરી કરશે.

ચાલો, આપણે હજી એક વધારે સ્ક્રિપ્ટ બનાવીએ, જેમાં વપરાશકર્તા પાસેથી જન્મવર્ષનું ઈનપુટ લેવામાં આવે અને વપરાશકર્તાનાં હાલનાં વર્ષ પ્રદર્શિત કરવામાં આવે. આ માટેનો કોડ નીચેના ખાનામાં બતાવ્યો છે.

Script 9: Script to calculate age of user in years

```
echo -n "Enter year of your birth: "  
read byear  
cyear=`date | tr -s ' ' | cut -d " " -f 6`  
age=`expr $cyear - $byear`  
echo "You are $age years old as of today."
```

script 9માં તમે નિરીક્ષણ કરશો કે date આદેશના આઉટપુટમાં તમામ બહુવિધ (multiple) ખાલી જગ્યા (સ્પેશ)ને દબાવીને (squeeze) ફક્ત એક જ ખાલી જગ્યામાં મેળવવાની ખાતરી કરવા માટે આપણે tr આદેશ સાથે -s વિકલ્પ વાપરેલો છે. તારીખમાંથી આપણે ફક્ત વર્ષની સંખ્યા જ જોઈતી હોવાથી કે જે date આદેશનો અમલ કરવાથી આઉટપુટની છઠ્ઠી જગ્યાએ આવે છે, આપણે cut ફિલ્ટરનો ઉપયોગ કર્યો છે. આકૃતિ 7.13માં સ્ક્રિપ્ટનો અમલ કરતાં મળેલા વિવિધ આઉટપુટ જણાવેલા છે.



```
File Edit View Terminal Help  
administrator@ubuntu:~$ sh script9.sh  
Enter year of your birth: 1974  
You are 39 years old as of today.  
administrator@ubuntu:~$ sh script9.sh  
Enter year of your birth: 2001  
You are 12 years old as of today.  
administrator@ubuntu:~$
```

આકૃતિ 7.13 : Script 9નો આઉટપુટ

શેલસ્ક્રિપ્ટનો ઉપયોગ (Use of Shell Scripts)

શેલસ્ક્રિપ્ટ લિનક્સનું એક અતિ સક્ષમ સાધન છે. તેમાં ઉચ્ચસ્તરીય પ્રોગ્રામિંગ ભાષાની લગભગ તમામ શક્તિ છે. તેનાથી એક વાર પરિચિત થયા પછી આપણે અનેક કાર્યો (ટાસ્ક)ને કરી શકીએ અને સ્વયંસંચાલિત પણ કરી શકીએ છીએ. સામાન્ય રીતે પુનરાવર્તિત થતાં કાર્યો શેલસ્ક્રિપ્ટ વાપરીને કરવાં જોઈએ. શેલસ્ક્રિપ્ટનાં કેટલાંક કાર્યો નીચે જણાવેલાં છે :

- અનેક આદેશોના સમૂહમાંથી એક નવો આદેશ બનાવો.
- કમ્પ્યુટર જાળવણીનાં અનેક પાસાંઓને સ્વયંસંચાલિત કરવા, ઉદાહરણ તરીકે, 1000 વપરાશકર્તાનાં ખાતાંઓ બનાવવાં, જે ફાઈલનું કદ 0 બાઈટ હોય તે તમામ રદ કરવી (ડિલીટ કરવી), નવાં સોફ્ટવેરનું પ્રસ્થાપન કરવું વગેરે.
- ડેટા બેક-અપ (Data backup - ડેટાની નકલ કરીને રાખવી.)

જોકે શેલસ્ક્રિપ્ટ વપરાશની કોઈ મર્યાદા નથી; વપરાશકર્તા પોતાની ઈચ્છા પ્રમાણે ગમે તે કાર્ય માટે વાપરી શકે છે.

સારાંશ

આપણે આ પ્રકરણમાં ઉબન્ટુ લિનક્સમાં આપવામાં આવેલું વિમ એડિટર કેવી રીતે વાપરવું તે બાબત શીખ્યાં. આ ટેક્સ્ટ એડિટર ઘણું સમર્થ છે; તે ફાઈલ બનાવવાની, તેમાં સુધારા-વધારા કરવાની તેમજ માહિતી રદ કરવાની સગવડ પૂરી પાડે છે. આ ઉપરાંત ફાઈલમાં આપણે કોઈ જરૂરી માહિતી શોધી શકીએ છીએ. આ એડિટર વડે કોઈ સાદી ટેક્સ્ટફાઈલ તેમ જ શેલસ્ક્રિપ્ટ કેવી રીતે બનાવવી તે પણ શીખ્યાં. શેલસ્ક્રિપ્ટ એ એક ટેક્સ્ટફાઈલ છે, જે આદેશોની શ્રેણીનો સમાવેશ કરે છે અને ફક્ત શેલસ્ક્રિપ્ટની ફાઈલનું નામ જણાવીને તેનો અમલ કરી શકાય છે. અંતમાં આપણે ચલનો ઉપયોગ અને પછી તેનો પદાવલિમાં ઉપયોગ કરીને શેલસ્ક્રિપ્ટને એક ઉચ્ચસ્તરીય પ્રોગ્રામ બરાબર કેવી રીતે બનાવાય તે પણ જોયું.

સ્વાધ્યાય

1. વિમ એડિટરમાં ઉપલબ્ધ વિવિધ મોડ સમજાવો.
2. વિમ એડિટરમાં ઉપલબ્ધ saveના વિવિધ વિકલ્પોની યાદી બનાવો અને સમજાવો.
3. ddt અને 2dd આદેશનો તફાવત સમજાવો.
4. શેલસ્ક્રિપ્ટ એટલે શું ?
6. શેલસ્ક્રિપ્ટના ઓછામાં ઓછા ત્રણ ઉપયોગો જણાવો.
7. હેતુલક્ષી પ્રશ્નો
 - (1) વિમ એડિટર કેટલા મોડમાં કામ કરે છે ?
 - (a) એક
 - (b) બે
 - (c) ત્રણ
 - (d) ચાર
 - (2) નીચેનાંમાંથી કયું વિધાન Gedit માટે ખરું છે ?
 - (a) તે એક કમાન્ડલાઈન એડિટર છે.
 - (b) તે ગ્રાફિકલ એડિટર છે.
 - (c) તે એડિટર નથી.
 - (d) તે KDE ડેસ્કટોપ વાતાવરણમાં ઉપલબ્ધ છે.

- (3) વિમ એડિટરમાં :wq એ નીચેનામાંથી કયા કાર્ય માટે વપરાય છે?
- (a) ફાઇલ સેવ કરવી અને એડિટ મોડમાં જ રહેવું.
 (b) ફાઇલ સેવ કરવી અને એડિટ મોડમાંથી બહાર નીકળી જવું.
 (c) ફાઇલમાં કરેલાં સુધારા સેવ કર્યા વિના જ એડિટ મોડમાંથી બહાર નીકળી જવું.
 (d) ઉપરના તમામ વિકલ્પ.
- (4) નીચેનામાંથી કઈ કી વિમ એડિટરના ઈન્સર્ટ મોડમાં જવા વપરાતી નથી ?
- (a) o (b) i
 (c) a (d) cw
- (5) નીચેનામાંથી કઈ કી લાઇન રદ કરવા માટે વપરાય છે ?
- (a) ce (b) ge
 (c) dd (d) d\$
- (6) નીચેનામાંથી કયા વિધાન ફાઇલમાંથી શબ્દસમૂહ શોધવા માટે વપરાય છે ?
- (a) :set is (b) :help cmd
 (c) :!cmd (d) /phrase<ENTER>
- (7) નીચેનામાંથી કઈ સિન્ટેક્સ (syntax) વપરાશકર્તાની મંજૂરી વિના આખી ફાઇલના તમામ phase 1ને phase 2માં બદલી નાખવા ઉપયોગમાં લઈ શકાય?
- (a) :%/phrase1/phrase2/g (b) :%/phrase1/phrase2/gc
 (c) :s/phrase1/phrase2/g (d) :s/phrase1/phrase2/gc
- (8) શેલસ્ક્રિપ્ટમાં નીચેનામાંથી કયું કેરેક્ટર કોમેન્ટ લાઇનનો નિર્દેશ કરવા વપરાય છે?
- (a) * (b) %
 (c) \$ (d) #
- (9) નીચેનામાંથી કયું ચિહ્ન શેલસ્ક્રિપ્ટને ચલમાંથી કિંમત મેળવવાની સૂચના આપે છે ?
- (a) * (b) %
 (c) \$ (d) #

પ્રાયોગિક સ્વાધ્યાય

નીચેનાં કાર્યો કરવા માટે શેલસ્ક્રિપ્ટ લખો :

- (a) નીચેનાં સ્વરૂપમાં તારીખ અને સમય દર્શાવવા માટે :

“Today is February 15, 2013 and current time is 12:10:23”

(b) વર્તમાન ઉપયોગકર્તાની લોગ-ઇન (login) માહિતી નીચેના સ્વરૂપમાં પ્રદર્શિત કરવા માટે;

Name of the user :

Login date :

Login time :

(c) તારીખ, સમય, ઉપયોગકર્તાનું નામ અને વર્તમાન ડિરેક્ટરી પ્રદર્શિત કરવાં.

(d) ઉપયોગકર્તા પાસેથી કોઈ સ્ટ્રિંગ અને ફાઇલનાં નામનું ઇનપુટ લો. તે ફાઇલમાં આપેલી સ્ટ્રિંગ કઈ કઈ જગ્યાએ છે તે શોધો.

(e) ઉપયોગકર્તા પાસેથી ફાઇલ નામનું ઇનપુટ લો અને તેમાં કેટલી લાઇન છે, તે ગણતરી કરો.

(f) ઉપયોગકર્તા પાસેથી બે ફાઇલ નામ ઇનપુટ લો અને એક નવી ફાઇલ બનાવો, જેમાં બન્ને ફાઇલની માહિતીનો સમાવેશ થાય.

(g) ઉપયોગકર્તા પાસેથી બે ફાઇલ નામ ઇનપુટ લો અને તેની સરખામણી કરો.





પ્રકરણ-7માં આપણે Vim એડિટરનો ઉપયોગ શીખ્યા અને તેમાં મૂળભૂત રીતે શેલસ્ક્રિપ્ટ કેવી રીતે લખવી તેનો અભ્યાસ કર્યો. શેલસ્ક્રિપ્ટ એક ઉચ્ચસ્તરીય ભાષાને સમકક્ષ છે તેનો ઉલ્લેખ પણ આપણે કર્યો હતો. શું આપણે નવી ભાષા શીખી રહ્યા છીએ? ના, આપણે કોઈ નવી ભાષા નથી જ શીખી રહ્યા પરંતુ ઓપન સોર્સ ઓપરેટિંગ સિસ્ટમ દ્વારા પૂરી પાડવામાં આવતી એક શ્રેષ્ઠ સુવિધાનો અભ્યાસ કરી રહ્યા છીએ. સિસ્ટમના રોજિંદા વહીવટ માટે શેલસ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે છે. દૂરના સ્થાને હોવા છતાં સિસ્ટમને સરળતાથી જોઈ અને નિયંત્રિત કરી શકાય તે માટેનું આ શ્રેષ્ઠ સાધન છે. અત્યાર સુધી બનાવવામાં આવેલી શેલસ્ક્રિપ્ટ ક્રમાંકુસાર (sequential) કાર્ય કરતી હતી; સ્ક્રિપ્ટમાં ઉમેર્યા હોય તે જ ક્રમમાં તેના આદેશોનો અમલ કરવામાં આવતો હતો. વહીવટી કાર્યો કરતી વખતે કેટલાંક વિધાનોનો અમલ પુનરાવર્તિત રીતે કરવાની જરૂર પડે છે. પૂર્વવ્યાખ્યાયિત શરત આધારિત કેટલાંક વિધાનોના અમલીકરણને છોડી દેવાની જરૂર પણ કદાચ પડે. આ પ્રકરણમાં આપણે સિસ્ટમના વહીવટને લગતી કેટલીક સ્ક્રિપ્ટ જોઈશું અને શેલસ્ક્રિપ્ટમાં નિર્ણાયક વિધાનો અને લૂપિંગ બંધારણો વિશે ચર્ચા કરીશું.

પ્રોસેસ આઈડી શોધવો (Finding Process Id)

લિનક્સમાં તમામ પ્રોગ્રામો (હાર્ડડિસ્ક પર સંગ્રહેલી અમલીકૃત ફાઇલો)નો ‘પ્રોસેસ’ (મેમરીમાં પ્રોગ્રામને મૂકી રન કરવો) સ્વરૂપે અમલ કરવામાં આવે છે. દરેક પ્રોસેસને ચાલુ કરવામાં આવે, ત્યારે તેની સાથે ‘પ્રોસેસ આઈ ડી’ (PID) નામે ઓળખાતો એક અનન્ય અંક સાંકળવામાં આવે છે. પ્રોસેસને જોઈ કે અટકાવી શકાય છે. *ps* કમાન્ડનો ઉપયોગ કોઈ પેરામીટર વગર કરવામાં આવે, તો વર્તમાન શેલ સાથે જોડાયેલી તમામ પ્રોસેસ નિહાળી શકાય છે. *ps -ef* કમાન્ડનો ઉપયોગ કરવાથી તમામ ઉપયોગકર્તાઓ દ્વારા શરૂ કરવામાં આવેલી પ્રોસેસની યાદી જોઈ શકાય છે. સિસ્ટમ પર ચાલતી પ્રોસેસની યાદી આકૃતિ 8.1માં દર્શાવી છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ ps -ef
UID          PID    PPID  C  STIME TTY          TIME CMD
root          1      0   0  15:14 ?           00:00:00 /sbin/init
root          2      0   0  15:14 ?           00:00:00 [kthreadd]
root          3      2   0  15:14 ?           00:00:00 [migration/0]
root          4      2   0  15:14 ?           00:00:00 [ksoftirqd/0]
root          5      2   0  15:14 ?           00:00:00 [watchdog/0]
root          6      2   0  15:14 ?           00:00:00 [migration/1]
root          7      2   0  15:14 ?           00:00:00 [ksoftirqd/1]
root          8      2   0  15:14 ?           00:00:00 [watchdog/1]
root          9      2   0  15:14 ?           00:00:00 [events/0]
root         10      2   0  15:14 ?           00:00:00 [events/1]
```

આકૃતિ 8.1 : પ્રોસેસની યાદી

આકૃતિ 8.1માં દર્શાવવામાં આવેલા કેટલાક સ્તંભની સમજૂતી કોષ્ટક 8.4માં આપવામાં આવી છે.

સ્તંભનું નામ	વર્ણન
UID	પ્રોસેસ ધરાવતા ઉપયોગકર્તાનું નામ અથવા ક્રમ
PID	દરેક પ્રોસેસ સાથે જોડાયેલ એક અનન્ય પ્રોસેસક્રમાંક
PPID	વર્તમાન પ્રોસેસ જેના દ્વારા શરૂ કરવામાં આવી હોય, તે પિતૃ-પ્રોસેસનો પ્રોસેસ-આઈડી
STIME	વર્તમાન પ્રોસેસ શરૂ કર્યાનો સમય
TTY	વર્તમાન પ્રોસેસને નિયંત્રિત કરતા ટર્મિનલનું નામ
TIME	વર્તમાન પ્રોસેસ દ્વારા ઉપયોગમાં લેવામાં આવેલા CPUના કુલ સમયની કિંમત
CMD	પ્રોસેસ શરૂ કરવા માટે ઉપયોગમાં લેવામાં આવેલ કમાન્ડ

કોષ્ટક 8.1 : ps -ef કમાન્ડ દ્વારા દર્શાવવામાં આવેલા સ્તંભની સમજૂતી

ઘણીવાર પ્રબંધક (એડમિનિસ્ટ્રેટર)ને કોઈ ચોક્કસ ઉપયોગકર્તા કુલ કેટલી પ્રોસેસનો ઉપયોગ કરી રહ્યો છે, તે જાણવાની જરૂર પડે છે, હવે આપણે કોઈ ચોક્કસ ઉપયોગકર્તા દ્વારા ચલાવવામાં આવેલી કુલ પ્રોસેસની સંખ્યા શોધવામાં એડમિનિસ્ટ્રેટરને મદદરૂપ બને તેવી એક સ્ક્રિપ્ટ લખીએ.

#Script 10: Script to find out how many processes a user is running.

```
clear
echo -n "Enter username: "
read username
cnt=`ps -ef | cut -d " " -f 1 | grep -o $username | wc -w`
echo "User $username is running $cnt processes."
```

સ્ક્રિપ્ટને *script10.sh* નામ આપી સેવ કરો. આ સ્ક્રિપ્ટના કાર્યને સમજવાનો પ્રયત્ન કરીએ. પ્રથમ કમાન્ડ સ્ક્રીન પરથી તમામ લખાણ દૂર કરશે. ત્યાર પછી ઉપયોગકર્તા સમક્ષ તેનું યુઝરનેમ ઉમેરવા માટેનો સંદેશ રજૂ કરવામાં આવશે. ત્યાર પછી કી-બોર્ડ પરથી આપવામાં આવેલી સ્ટ્રિંગ read કમાન્ડ દ્વારા *username* નામના ચલમાં મૂકવામાં આવશે. ત્યાર બાદ pipe વડે ps, cut, grep અને wc એમ ચાર કમાન્ડનું સંયોજન કરવામાં આવ્યું છે. સિસ્ટમના તમામ ઉપયોગકર્તાઓ દ્વારા ચલાવવામાં આવેલી પ્રોસેસની યાદી ps -ef કમાન્ડ દ્વારા દર્શાવે છે. તેનું પરિણામ cut કમાન્ડને આપવામાં આવે છે. આ પરિણામમાંથી cut કમાન્ડ પ્રથમ ફિલ્ડ (ઉપયોગકર્તાનું નામ) અલગ પાડે છે. ત્યાર બાદ છૂટા પાડવામાં આવેલા આ પ્રથમ ફિલ્ડની યાદી grep કમાન્ડને આપવામાં આવે છે. grep કમાન્ડ *username* ચલમાં આપવામાં આવેલી કિંમત તમામ ઉપયોગકર્તાઓની યાદી સાથે સરખાવે છે. આ સરખાવેલી યાદી ત્યાર પછી wc કમાન્ડને આપવામાં આવે છે, જે આપેલ શબ્દ (ઉપયોગકર્તાનાં નામ)ની સંખ્યા ગણે છે. અંતમાં મળેલ આ સંખ્યાનો cnt નામના ચલમાં સંગ્રહ કરવામાં આવે છે. અંતિમ કમાન્ડ ત્યાર પછી જરૂરી પરિણામને સ્ક્રીન પર દર્શાવે છે. આકૃતિ 8.2માં આ સ્ક્રિપ્ટનું નમૂનારૂપ પરિણામ દર્શાવ્યું છે.

```
File Edit View Terminal Help
Enter username: root
User root is running 103 processes.
administrator@ubuntu:~$
```

આકૃતિ 8.2 : સ્ક્રિપ્ટ-10નું પરિણામ

આગળ જણાવ્યું તે મુજબ જરૂરી હોય તો આપણે કોઈ પ્રોસેસને દૂર કરી કેટલીક મેમરી જગ્યાને મુક્ત કરી શકીએ છીએ. મેમરીમાંથી પ્રોસેસને દૂર કરવા માટે *kill* કમાન્ડનો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે જો આપણે

Skill -9 101

આદેશ આપીશું તો PID=101 ધરાવતી પ્રોસેસને મેમરીમાંથી પ્રયત્નપૂર્વક (forcibly) દૂર કરવામાં આવે છે. *script10.sh* જેવી જ એક બીજી સ્ક્રિપ્ટ જોઈએ જે કમાન્ડલાઈન આર્ગ્યુમેન્ટ તરીકે ઉપયોગકર્તાનું નામ સ્વીકારશે અને તે ઉપયોગકર્તા દ્વારા ઉપયોગમાં લેવામાં આવેલા ટર્મિનલની સંખ્યા ગણી આપશે. આ સ્ક્રિપ્ટ માટેનો કોડ નીચેના બોક્સમાં આપવામાં આવ્યો છે.

```
#Script 11: Script to find out how many terminals a user has opened.
cnt=`who | cut -d " " -f 1 | grep -o $1 | wc -w`
echo "User $1 has opened $cnt terminals"
```

સ્ક્રિપ્ટને *script11.sh* નામ આપી સંગ્રહ કરો. આ પહેલાંના ઉદાહરણમાં બનાવવામાં આવેલ સ્ક્રિપ્ટમાં આપણે ચલનો ઉપયોગ કરેલો તે યાદ કરો. આ સ્ક્રિપ્ટમાં આપણે કમાન્ડલાઈન આર્ગ્યુમેન્ટનો ઉપયોગ કર્યો છે. અહીં \$1 કમાન્ડલાઈન આર્ગ્યુમેન્ટ રજૂ કરે છે. આ સ્ક્રિપ્ટનો અમલ કરવા માટે નીચે આપેલ કમાન્ડનો ઉપયોગ કરી શકાય.

Ssh script11.sh administrator

તમે જોઈ શકો છો કે આ સ્ક્રિપ્ટનો પણ આપણે અગાઉની સ્ક્રિપ્ટની જેમ જ અમલ કર્યો, પરંતુ આપણે અહીં એક વધારાની કિંમત ‘administrator’નો ઉપયોગ કર્યો છે. (વાચક પોતાની પસંદગીનું કોઈ પણ નામ અહીં આપી શકે છે.) કમાન્ડલાઈન દ્વારા મેળવવામાં આવેલા ચલનો લિનક્સ \$1, \$2, \$3 એમ ‘ડોલર ચલ’ (dollar variables)માં સંગ્રહ કરે છે. પ્રથમ આર્ગ્યુમેન્ટ \$1માં, બીજી આર્ગ્યુમેન્ટ \$2માં, ત્રીજી આર્ગ્યુમેન્ટ \$3માં એમ \$9 સુધી સંગ્રહ કરવામાં આવે છે. આર્ગ્યુમેન્ટને કમાન્ડલાઈન આર્ગ્યુમેન્ટ તરીકે ઓળખવામાં આવે છે. આકૃતિ 8.3માં આ સ્ક્રિપ્ટનું પરિણામ દર્શાવ્યું છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script11.sh administrator
User administrator has opened 2 terminals
administrator@ubuntu:~$
```

આકૃતિ 8.3 : સ્ક્રિપ્ટ-11નું પરિણામ

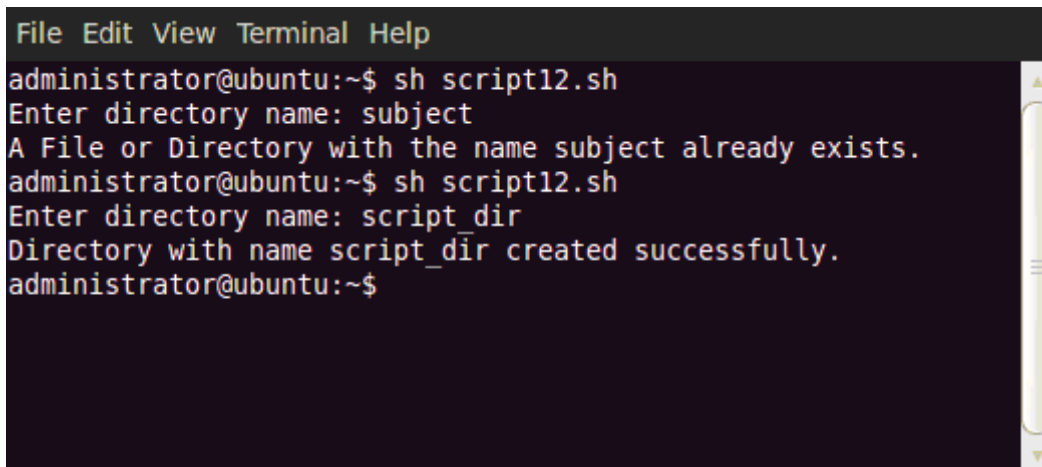
આ સ્ક્રિપ્ટનું કાર્ય સમજવાનો પ્રયત્ન કરીએ. પ્રથમ વિધાનમાં કોમેન્ટનો ઉપયોગ કર્યા બાદ આપણે ચાર કમાન્ડ who, cut, grep અને wcનો સંયુક્ત ઉપયોગ પાઠપત્રની મદદથી કરેલો છે. સિસ્ટમમાં લોગ-ઈન થયેલા તમામ ઉપયોગકર્તાની યાદી who કમાન્ડ દ્વારા દર્શાવવામાં આવે છે. આ પરિણામમાંથી પ્રથમ ફિલ્ડને અલગ પાડે છે. ત્યાર બાદ છૂટા પાડવામાં આવેલ પ્રથમ ફિલ્ડની યાદી grep કમાન્ડને આપવામાં આવે છે. કમાન્ડલાઈન આર્ગ્યુમેન્ટ તરીકે ઉમેરવામાં આવેલી કિંમતને grep કમાન્ડ દ્વારા તમામ ઉપયોગકર્તાની યાદીમાંથી શોધવામાં આવે છે. (\$1 = administrator). સરખામણી થયેલી યાદી wc કમાન્ડને આપવામાં આવે છે, જે આપેલ શબ્દ (ઉપયોગકર્તાનું નામ)ની ગણતરી કરે છે. અંતે, ગણવામાં આવેલા શબ્દોની સંખ્યાનો cnt ચલમાં સંગ્રહ કરવામાં આવે છે. ત્યાર બાદ અંતિમ કમાન્ડ કમાન્ડ જરૂરી પરિણામને સ્ક્રીન પર દર્શાવે છે.

નિર્ણય લેવાનાં કાર્યો (Decision Making Tasks)

ધારો કે, એડમિનિસ્ટ્રેટરને એક ડિરેક્ટરી બનાવવી છે, તો તે માટે તે mkdir કમાન્ડનો ઉપયોગ કરી શકે છે, પરંતુ આ માટે જો સ્ક્રિપ્ટનો ઉપયોગ કરવામાં આવે, તો તેમાં યોગ્ય સંદેશા પણ દર્શાવી શકાય છે. હવે આપણે એક એવી સ્ક્રિપ્ટ બનાવીએ, જે એડમિનિસ્ટ્રેટરને એક ડિરેક્ટરી બનાવવામાં મદદરૂપ બને.

```
#Script 12: Script to create a directory with appropriate message.
echo -n "Enter directory name: "
read mydir
if [ -d $mydir -o -f $mydir ]
then
    echo "A File or Directory with the name $mydir already exists".
    exit 0
fi
mkdir $mydir
echo "Directory with name $mydir created successfully."
```

સ્ક્રિપ્ટને *script12.sh* નામ આપી સેવ કરો. આ સ્ક્રિપ્ટમાં આપણે if-then-*fi* બંધારણનો ઉપયોગ કર્યો છે તે જુઓ. આ બંધારણ દ્વારા શેલસ્ક્રિપ્ટમાં નિર્ણય સંબંધી કાર્યો કરી શકાય છે. લિનક્સમાં *if* વિધાન આપેલ શરતની નિર્ગમ સ્થિતિ (Exit status) સાથે સંબંધિત છે કમાન્ડનો અમલ સફળતાપૂર્વક અમલ કરવામાં આવ્યો હોય, તો Exit statusની કિંમત 0 હશે, નહીં તો તેની કિંમત 1 થશે. આકૃતિ 8.4માં સ્ક્રિપ્ટનું પરિણામ દર્શાવ્યું છે.



```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script12.sh
Enter directory name: subject
A File or Directory with the name subject already exists.
administrator@ubuntu:~$ sh script12.sh
Enter directory name: script_dir
Directory with name script_dir created successfully.
administrator@ubuntu:~$
```

આકૃતિ 8.4 : સ્ક્રિપ્ટ-12નું પરિણામ

આકૃતિ 8.4ના પરિણામને ધ્યાનપૂર્વક જુઓ. પહેલા કિસ્સામાં ડિરેક્ટરી પહેલેથી હયાત હોવાનો આપણને સંદેશ મળે છે. જ્યારે બીજા કિસ્સામાં આપવામાં આવેલ નામની ડિરેક્ટરી બનાવવામાં આવે છે. ઉપરની સ્ક્રિપ્ટમાં જોઈ શકાય છે કે શરતને ચોરસ કૌંસમાં આવરીને લખવામાં આવી છે. અહીં ઊઘડતા ચોરસ કૌંસ પછી અને પૂરા થતા ચોરસ કૌંસ પહેલાં એક જગ્યા રાખવી જરૂરી છે. જો આપવામાં આવેલી શરતનું પરિણામ હકારાત્મક (true) આવશે તો જ then પછી આવેલાં વિધાનોનો અમલ કરવામાં આવશે. *if* વિધાનના અંતને *fi* વિધાન દ્વારા દર્શાવવામાં આવે છે. અહીં એ નોંધ લેવી પણ જરૂરી છે કે *then* કી-વર્ડને *if* વિધાનની નીચેની લીટીમાં ટાઈપ કરવું જરૂરી છે, અન્યથા ભૂલનો સંદેશ દર્શાવવામાં આવશે. સ્ક્રિપ્ટમાં ઉપયોગમાં લેવામાં આવેલા -d, -f અને -o વિકલ્પ વિશે આ પ્રકરણમાં આગળ ચર્ચા કરવામાં આવશે.

લિનક્સમાં શેલસ્ક્રિપ્ટની રચના કરતી વખતે નિર્ણય માટે નીચેનાં ચાર વિધાનોનો ઉપયોગ કરી શકાય છે :

if-then-fi

if-then-else-fi

if-then-elif-then-else-fi

case-esac

ફાઈલની નકલ કરી (બેક-અપ માટે) તેનો અન્ય ડિરેક્ટરીમાં સંગ્રહ કરવો એ ઘણી સામાન્ય ક્રિયા છે. આ સમયે બન્ને ફાઈલો એક સમાન છે કે નહીં તે બાબતે ઉપયોગકર્તાના મનમાં ઘણીવાર દ્વિધા પ્રવર્તે છે. આપણે આવી ફાઈલોને સરખાવવા માટેની સ્ક્રિપ્ટ લખીએ. આ સ્ક્રિપ્ટનો અમલ કરવામાં આવે, ત્યારે *cmp* કમાન્ડનો ઉપયોગ કરી બન્ને ફાઈલોને સરખાવવામાં આવશે. *cmp* કમાન્ડ પરિણામને આધારે યોગ્ય સંદેશ દર્શાવવામાં આવશે.

#Script 13: Script to compare files.

echo -n "Enter a file name: "

read fname

if cmp ./\$fname ./backup/\$fname

then

echo "\$fname is same at both places."

else

echo "Both \$fname are different."

fi

સ્ક્રિપ્ટને *script13.sh* નામ આપી સેવ કરો. અહીં સૌપ્રથમ આપણે ઉપયોગકર્તા પાસેથી ફાઈલનું નામ સ્વીકારીએ છીએ. સ્ક્રિપ્ટની સરળતા ખાતર ડિરેક્ટરીના સંપૂર્ણ (absolute) પાથનો ઉપયોગ કરેલો છે. તમે તેને સંબંધિત (relative) પાથમાં રૂપાંતરિત કરી શકો છો. આપણે અહીં ધારણા કરી છે કે બંને સ્થાને રહેલ ફાઈલનાં નામ સમાન છે. આકૃતિ 8.5 સ્ક્રિપ્ટનું પરિણામ દર્શાવે છે.

```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script13.sh
Enter a file name: address.txt
address.txt is same at both places.
administrator@ubuntu:~$ sh script13.sh
Enter a file name: address.txt
./address.txt ./backup/address.txt differ: byte 398, line 6
Both address.txt are different.
administrator@ubuntu:~$
```

આકૃતિ 8.5 : સ્ક્રિપ્ટ-13નું પરિણામ

અહીં સ્ક્રિપ્ટનો અમલ બે વખત કરવામાં આવ્યો છે. પ્રથમ અમલ વખતે બન્ને ફાઇલની વિગતો સમાન હોવાને કારણે ફાઇલો સરખી હોવાનો સંદેશ જોઈ શકાય છે. બીજી વખતમાં અમલના પહેલાં વર્તમાન ડિરેક્ટરીની ફાઇલનું લખાણ બદલવામાં આવ્યું છે, જેથી આપણને ફાઇલો સમાન ન હોવાનો સંદેશ દર્શાવવામાં આવે છે.

આ પહેલાના પ્રકરણમાં આપણે લોગ-ઇન થયેલા ઉપયોગકર્તાનું સ્વાગત કરતી એક નાની સ્ક્રિપ્ટ લખી હતી. હવે તેને સુધારીએ, જેથી તે સ્ક્રિપ્ટ ઉપયોગકર્તાના લોગ-ઇન સમય અનુસાર Good morning, Good Afternoon કે Good Evening સંદેશ દર્શાવશે.

```
#Script 14: Script to display welcome message to the user.
clear
hour=`date +%H`
usrname=`who am i | cut -d " " -f 1`
if [ $hour -ge 0 -a $hour -lt 12 ]
then
    echo "Good Morning $usrname, Welcome to Ubuntu Linux Session."
elif [ $hour -ge 12 -a $hour -lt 18 ]
then
    echo "Good Afternoon $usrname, Welcome to Ubuntu Linux Session."
else
    echo "Good Evening $usrname, Welcome to Ubuntu Linux Session."
fi
```

આકૃતિ 8.6માં આ સ્ક્રિપ્ટનું પરિણામ દર્શાવ્યું છે. સ્ક્રિપ્ટનું પરિણામ ઉપયોગકર્તાના લોગ-ઇન સમય પ્રમાણે બદલાતું રહેશે.

```
File Edit View Terminal Help
Good Afternoon administrator, Welcome to Ubuntu Linux Session.
administrator@ubuntu:~$
```

આકૃતિ 8.6 : સ્ક્રિપ્ટ-14નું પરિણામ

test કમાન્ડ (The test command)

આ if વિધાનનો વિવિધ સ્વરૂપોમાં ઉપયોગ શક્ય છે. આ પહેલાંની સ્ક્રિપ્ટમાં આવેલા ચોરસ કૌંસના સ્થાને ઉપયોગમાં લઈ શકાય તે માટે લિનક્સ test કમાન્ડની સુવિધા આપે છે. હવે આપણે ઉપયોગકર્તાએ આપેલ માહિનામાં નિશ્ચિત સંખ્યામાં ફાઈલ બનાવી છે કે નહીં તે તપાસી આપતી સ્ક્રિપ્ટ બનાવી તેમાં આ કમાન્ડનો ઉપયોગ કરીએ.

```
#Script 15: Script to see whether user has created more than specified files in a month.
clear
cnt=`ls -l | grep -c [-]"$1"`
echo -n "Enter number of files: "
read nfile
if test $cnt -gt $nfile
then
    echo "You have created more than $nfile files in the month of $1"
else
    echo "You have not created more than $nfile files in the month of $1"
fi
```

આ સ્ક્રિપ્ટને સમજવાનો પ્રયત્ન કરીએ. અહીં આપણે *cnt* નામનો એક ચલ વ્યાખ્યાયિત કર્યો છે. આપેલ માહિનામાં બનાવવામાં આવેલી તમામ ફાઈલોની સંખ્યા આ ચલમાં સંગ્રહવામાં આવે છે. ફાઈલની સંખ્યા ગણવા માટે *ls* અને *grep* કમાન્ડનો ઉપયોગ કરવામાં આવ્યો છે. *ls -l* કમાન્ડની મદદથી ફાઈલ અને ડિરેક્ટરીની વિગતો દર્શાવી શકાય છે. તેનું પરિણામ ત્યાર પછી *grep* કમાન્ડને આપી *[-]"\$1"* શબ્દસમૂહ સાથે સરખાવવામાં આવે છે. માહિનાની કિંમત બે અંકોની હોય છે અને તે કમાન્ડલાઈન આર્ગ્યુમેન્ટ દ્વારા મેળવી *\$1*માં સંગ્રહવામાં આવે છે. સરખામણી થયેલ ફાઈલોની સંખ્યાને *nfile* નામના એક ચલમાં સંગ્રહવામાં આવી છે. શરતમાં આપવામાં આવેલ *-gt* વિકલ્પ ‘થી વધુ’ (greater than) સરખામણી દર્શાવે છે. અહીં આપણે *cnt*ની કિંમત *nfile*થી વધુ છે કે નહીં તે ચકાસીએ છીએ. *cnt*ની કિંમત *nfile*ની કિંમત કરતાં વધુ હોય તો “You have created more than \$nfile files in the month of \$1”, સંદેશ દર્શાવવામાં આવે છે. જો *\$nfile* અને *\$1* યોગ્ય કિંમતો સાથે બદલવામાં ન આવે, તો “You have not created more than \$nfile files in the month of \$1” સંદેશ દર્શાવવામાં આવશે.

કમાન્ડ પ્રોમ્પ્ટ પર નીચેનો કમાન્ડ આપવામાં આવે ત્યારે મળતું પરિણામ આકૃતિ 8.7માં દર્શાવ્યું છે.

\$ssh script15.sh 02

```
File Edit View Terminal Help
Enter number of files: 10
You have created more than 10 files in the month of 02
administrator@ubuntu:~$
```

આકૃતિ 8.7 : સ્ક્રિપ્ટ-15નું પરિણામ

if વિધાન આંકડાકીય વિગતો, અક્ષરોની હારમાળા (strings) અને ફાઈલ સાથે કાર્ય કરી શકે છે. આ ચકાસણીઓને અનુક્રમે આંકડાકીય ચકાસણી, સ્ટ્રિંગ-ચકાસણી અને ફાઈલ-ચકાસણીના નામથી ઓળખવામાં આવે છે. આપણે અત્યાર સુધી બનાવેલ સ્ક્રિપ્ટમાં -d, -f, -o, -a, -ge, -lt અને -gt જેવા વિકલ્પનો ઉપયોગ કર્યો. આ તમામ વિકલ્પો જુદા-જુદા પ્રકારની શરતો સરખાવવા માટેની સુવિધા આપે છે.

સંબંધિત પ્રક્રિયકો (Relational Operators)

આંકડાકીય ચકાસણીઓ કરવા માટે રિલેશનલ ઓપરેટરનો ઉપયોગ કરવામાં આવે છે. -ge, -lt અને -gt ઓપરેટરો રિલેશનલ ઓપરેટર છે. આ ઓપરેટરનો ઉપયોગ બે આંકડાકીય ઓપરેન્ડની સરખામણી માટે કરવામાં આવે છે. શેલસ્ક્રિપ્ટમાં ઉપયોગમાં લઈ શકાય તે માટેના તમામ રિલેશનલ ઓપરેટરની યાદી તેના ઉપયોગ સાથે કોષ્ટક 8.2માં આપેલી છે.

ઓપરેટર	ઉપયોગ
-gt	-થી વધુ (greater than)
-lt	-થી ઓછું (less than)
-ge	-થી વધુ અથવા બરાબર (greater than or equal to)
-le	-થી ઓછું અથવા બરાબર (less than or equal to)
-ne	-સમાન નહીં (not equal to)
-eq	-સમાન (equal to)

કોષ્ટક 8.2 : રિલેશનલ ઓપરેટર

તાર્કિક પ્રક્રિયકો (Logical Operators)

ચોક્કસ અને યોગ્ય નિર્ણયો લેવા માટે ઉપયોગકર્તાને ઘણીવાર એકથી વધુ શરતોનું સંયોજન કરવું પડે છે. શરતોને સંયોજિત કરવા લોજિકલ ઓપરેટરનો ઉપયોગ કરવામાં આવે છે. કોષ્ટક 8.3માં લોજિકલ ઓપરેટરની યાદી તેના ઉપયોગ સાથે દર્શાવી છે.

ઓપરેટર	ઉપયોગ	સંયોજિત કરવામાં આવનાર ન્યૂનતમ શરતો	પરિણામ
-a	AND	બે	બન્ને શરત સાચી હોય તો જ True, નહીં તો False
-o	OR	બે	કોઈ પણ એક શરત સાચી હોય તો True, બંને શરતો ખોટી હોય તો જ False
!	NOT	એક	True પરિણામને False અને False શરતને Trueમાં રૂપાંતરિત કરવા

કોષ્ટક 8.3 : લૉજિકલ ઓપરેટર

ફાઇલ પ્રક્રિયકો (File Operators)

ફાઇલ કે ડિરેક્ટરીની સ્થિતિ (status) તપાસવા માટે પણ *if* વિધાનનો ઉપયોગ શક્ય છે. રીલેશનલ ઓપરેટરની જેમ ફાઇલની સ્થિતિ તપાસવા ફાઇલ-ઓપરેટર ઉપલબ્ધ છે. આ ઓપરેટરનો ઉપયોગ *if* વિધાન સાથે શરતના સ્વરૂપમાં કરવામાં આવે છે. ફાઇલ-ઓપરેટરનો ઉપયોગ કરી જાણી શકાય છે કે આપેલ નામ માત્ર સામાન્ય ફાઇલ છે કે ડિરેક્ટરી છે ફાઇલને આપવામાં આવેલ મંજૂરી (permissions) અંગેની માહિતી મેળવવા માટે પણ તેનો ઉપયોગ કરી શકાય છે કોષ્ટક 8.4માં આ વિકલ્પોની યાદી દર્શાવી છે.

તપાસેલી શરત	પરિણામ
-s name	આપેલ નામ ધરાવતી ફાઇલ અસ્તિત્વમાં હોય અને તેનું કદ શૂન્યથી વધુ હોય.
-f name	આપેલ નામ ધરાવતી ફાઇલ અસ્તિત્વમાં હોય અને તે ડિરેક્ટરી ન હોય.
-d name	આપેલ નામ ધરાવતી ડિરેક્ટરી અસ્તિત્વમાં હોય.
-r name	આપેલ નામ ધરાવતી ફાઇલ અસ્તિત્વમાં હોય અને ઉપયોગકર્તા read મંજૂરી ધરાવતો હોય.
-w name	આપેલ નામ ધરાવતી ફાઇલ અસ્તિત્વમાં હોય અને ઉપયોગકર્તા write મંજૂરી ધરાવતો હોય.
-x name	આપેલ નામ ધરાવતી ફાઇલ અસ્તિત્વમાં હોય અને ઉપયોગકર્તા execute મંજૂરી ધરાવતો હોય.

કોષ્ટક 8.4 : ફાઇલ-ચકાસણીની શરતો

એડમિનિસ્ટ્રેટરને ઘણીવાર આપેલ ફાઇલનું કદ શૂન્ય છે કે નહીં તે જાણવાની જરૂર પડે છે. ત્યાર પછી જો ફાઇલનું કદ શૂન્ય હોય તો તે ફાઇલને દૂર કરવા જેવાં જાળવણી (maintenance)નાં કાર્યો કરી શકે છે. આ ઉપરાંત તે જાણવા માગે છે કે ફાઇલને write મંજૂરી આપવામાં આવી છે કે નહીં. એક સ્ક્રિપ્ટ બનાવીએ જે એડમિનિસ્ટ્રેટરને ફાઇલનું કદ તપાસી તેને આપવામાં આવેલ મંજૂરી વિશે માહિતી દર્શાવે.

#Script 16: Script to check file size.

```
echo -n "Enter a file name: "
```

```
read fname
```

```
if [ -s $fname -a -w $fname ]
```

```
then
```

```
    echo $fname has size greater than 0 and user has write permission on it.
```

```
else
```

```
    echo $fname has size 0 or user does not have write permission on it.
```

```
fi
```

ફાઇલને *script16.sh* નામ આપી સેવ કરો. અહીં **if [-s \$fname -a -w \$fname]** એકથી વધુ શરત દર્શાવે છે. જો બન્ને શરતોનું પરિણામ હકારાત્મક મળશે તો જ *if* વિધાનનો અમલ કરવામાં આવશે. *if* વિધાનના અમલ સમયે મળી શકે તે શક્ય પરિણામોની યાદી કોષ્ટક 8.5માં દર્શાવી છે અને આકૃતિ 8.8 સ્ક્રિપ્ટનાં જુદાં-જુદાં પરિણામ દર્શાવે છે.

-s \$fname	કારણ	-w \$fname	કારણ	if [-s \$fname -a -w \$fname]
False	ફાઇલનું કદ = 0 અથવા ફાઇલ અસ્તિત્વમાં નથી	False	Write મંજૂરીનો અભાવ	False
False	ફાઇલનું કદ = 0 અથવા ફાઇલ અસ્તિત્વમાં નથી	True	Write મંજૂરી આપેલી છે.	False
True	ફાઇલનું કદ > 0	False	Write મંજૂરીનો અભાવ	False
True	ફાઇલનું કદ > 0	True	Write મંજૂરી આપેલી છે.	True

કોષ્ટક 8.5 : if [-s \$fname -a -w \$fname] નું પરિણામ

```
File Edit View Terminal Help
administrator@ubuntu:~$ sh script16.sh
Enter a file name: script5.sh
script5.sh has size 0 or user does not have write permission on it.
administrator@ubuntu:~$ sh script16.sh
Enter a file name: script10.sh
script10.sh has size greater than 0 and user has write permission on it.
administrator@ubuntu:~$
```

આકૃતિ 8.8 : સ્ક્રિપ્ટ-16નું પરિણામ

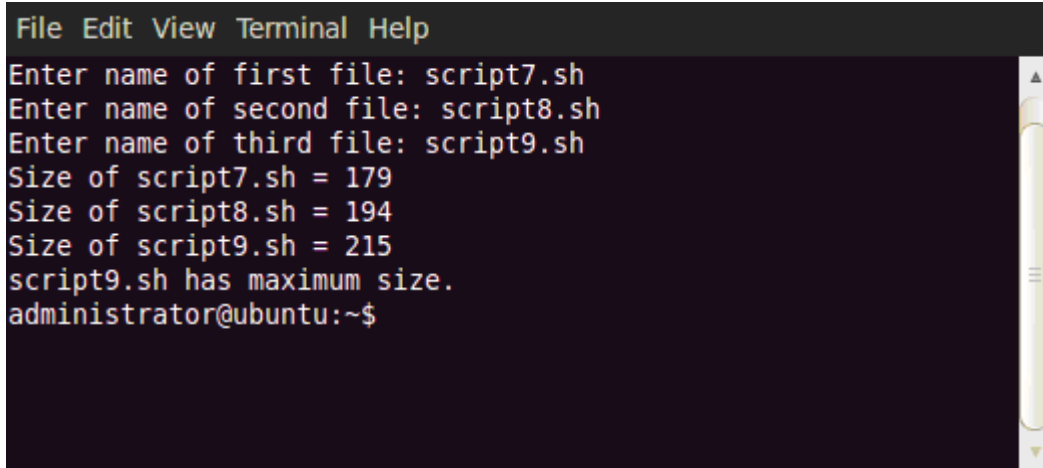
અત્યાર સુધીમાં ઉપયોગમાં લેવામાં આવેલ if-then-fi અને if-then-else-fi વિધાનો આપેલ શરતોને મર્યાદિત સંખ્યામાં ચકાસી શકે છે. જો ઉપયોગકર્તા એકથી વધુ શરતોને ચકાસવા ઇચ્છતો હોય, તો આ વિધાનો વધુ મદદરૂપ થઈ શકતાં નથી, આવા કિસ્સામાં if-then-elif-then-else-fi અથવા case વિધાનનો ઉપયોગ કરી શકાય છે.

હવે આપણે એક એવી સ્ક્રિપ્ટ બનાવીએ જે ત્રણ ફાઈલો ઉપયોગકર્તા તરફથી અને તેમાંથી મહત્તમ કદ ધરાવતી ફાઈલ દર્શાવશે.

```
#Script 17: Script to find the file with the maximum size.
```

```
clear  
echo -n "Enter name of first file: "  
read fname1  
echo -n "Enter name of second file: "  
read fname2  
echo -n "Enter name of third file: "  
read fname3  
fsize1=`wc -c $fname1 | cut -d " " -f 1`  
fsize2=`wc -c $fname2 | cut -d " " -f 1`  
fsize3=`wc -c $fname3 | cut -d " " -f 1`  
echo Size of $fname1 = $fsize1  
echo Size of $fname2 = $fsize2  
echo Size of $fname3 = $fsize3  
if [ $fsize1 -eq $fsize2 -a $fsize1 -eq $fsize3 ]  
then  
    echo "All files have same size"  
elif [ $fsize1 -gt $fsize2 -a $fsize1 -gt $fsize3 ]  
then  
    echo "$fname1 has maximum size."  
elif [ $fsize2 -gt $fsize1 -a $fsize2 -gt $fsize3 ]  
then  
    echo "$fname2 has maximum size."  
else  
    echo "$fname3 has maximum size."  
fi
```

સ્ક્રિપ્ટને *script17.sh* નામ આપી સેવ કરો. `clear` કમાન્ડ પછીનાં છ વિધાનો દ્વારા ઉપયોગકર્તા પાસેથી ફાઇલનાં નામ મેળવવામાં આવે છે. ત્યારપછીનાં ત્રણ વિધાન દ્વારા ફાઇલનાં કદ શોધવામાં આવે છે, જે પછીથી ઉપયોગકર્તા સમક્ષ દર્શાવવામાં આવે છે. અંતમાં `if` વિધાનની મદદથી મહત્તમ કદ ધરાવતી ફાઇલ શોધવામાં આવે છે. આકૃતિ 8.9 સ્ક્રિપ્ટનું પરિણામ દર્શાવે છે.



```
File Edit View Terminal Help
Enter name of first file: script7.sh
Enter name of second file: script8.sh
Enter name of third file: script9.sh
Size of script7.sh = 179
Size of script8.sh = 194
Size of script9.sh = 215
script9.sh has maximum size.
administrator@ubuntu:~$
```

આકૃતિ 8.9 : સ્ક્રિપ્ટ-17નું પરિણામ

case વિધાન (The case statement)

સરખામણીઓની સંખ્યા વધે ત્યારે `if-then-elif-then-else-fi` વિધાન કઢંગું બની જાય છે. એકાધિક શરતો તપાસવા માટેનો અન્ય વિકલ્પ `case` વિધાન છે. હવે આપણે એક એવી સ્ક્રિપ્ટ બનાવીએ, જે ઉપયોગકર્તા પાસેથી પસંદગી મેળવી તે મુજબ વિવિધ પ્રકારની ફાઇલ-પ્રક્રિયાઓ અમલમાં મૂકે.

```
# Script 18: Script to perform various file and directory operations.
echo "1 - Display Current Dir "
echo "2 - Make Dir "
echo "3 - Copy a file "
echo "4 - Rename a file "
echo "5 - Delete a file "
echo "0 - Exit "
echo -n "Enter your choice [0-5] : "
read choice
case $choice in
1)
    echo $PWD
;;
```

2)

```
echo -n "Enter name of the directory to be created: "  
read dname  
if [ -d $dname ]  
then  
    echo "Directory with the name $dname already exists."  
    exit 0  
else  
    mkdir $dname  
    echo "Directory $dname created successfully."  
fi  
;;
```

3)

```
echo -n "Enter source file name : "  
read sfile  
echo -n "Enter destination file name : "  
read dfile  
cp -u $sfile $dfile  
;;
```

4)

```
echo -n "Enter old file name : "  
read oldf  
echo -n "Enter new file name : "  
read newf  
mv $oldf $newf  
;;
```

5)

```
echo -n "Enter file name to delete : "  
read fname  
rm $fname  
;;
```

```

0)
    exit 0
;;
*)
    echo "Incorrect choice exiting script."
esac

```

સ્ક્રિપ્ટને *script18.sh* નામ આપી સેવ કરો. અહીં દરેક પ્રક્રિયા માટે જુદા-જુદા વિભાગ લખવામાં આવ્યા છે. જ્યારે ઉપયોગકર્તા 0 થી 5 વચ્ચેની કોઈ આંકડાકીય કિંમત ઉમેરે છે, ત્યારે તેનો *choice* નામના ચલમાં સંગ્રહ કરવામાં આવે છે. case વિધાન choice ચલમાં આવેલ કિંમતને વાંચે છે અને પૂરા થતા કૌંસની પહેલાં આવેલ જે કિંમત સાથે સરખામણી મળે તે વિભાગમાં નિયંત્રણ લઈ જાય છે. બે અર્ધવિરામની નિશાની (;;) ન આવે ત્યાં સુધી આ વિભાગનાં તમામ વિધાનોનો અમલ કરવામાં આવે છે. બે અર્ધવિરામની નિશાની આવ્યા બાદ નિયંત્રણને case વિધાનના અંત પછીની લીટી પર લઈ જવામાં આવે છે. case વિધાનનો અંત દર્શાવવા esac કી-વર્ડનો ઉપયોગ કરવામાં આવે છે. શેલ case વિધાનના અંત પછી લખવામાં આવેલાં વિધાનોનું અમલીકરણ ત્યાર પછી શરૂ કરે છે. જો ઉપયોગકર્તા એવી કિંમત ઉમેરે કે જે કોઈ પણ કેસની કિંમત સાથે સરખાવી ન શકાય, ત્યારે ફૂટી (*) સ્વરૂપે કિંમત આપવામાં આવેલી હોય તે વિભાગમાં નિયંત્રણને ખસેડવામાં આવે છે. જો આ વિભાગનો ઉપયોગ કરવામાં આવ્યો હોય, તો તે ઉપયોગકર્તાને સ્ક્રિપ્ટમાંથી બહાર નીકળવાની કે યોગ્ય સંદેશ દર્શાવી વધારાની કોઈ ક્રિયા કરવાની સુવિધા આપે છે. સ્ક્રિપ્ટ-18નાં જુદાં-જુદાં પરિણામ આકૃતિ 8.10માં દર્શાવ્યાં છે.

```

File Edit View Terminal Help
administrator@ubuntu:~$ sh script18.sh
1 - Display Current Dir
2 - Make Dir
3 - Copy a file
4 - Rename a file
5 - Delete a file
0 - Exit
Enter your choice [0-5] : 1
/home/administrator
administrator@ubuntu:~$ sh script18.sh
1 - Display Current Dir
2 - Make Dir
3 - Copy a file
4 - Rename a file
5 - Delete a file
0 - Exit
Enter your choice [0-5] : 2
Enter name of the directory to be created: LinuxScript
Directory with the name LinuxScript already exists.
administrator@ubuntu:~$

```

આકૃતિ 8.10 : સ્ક્રિપ્ટ-18નું પરિણામ

case વિધાનની વાક્યરચના (syntax) નીચે આપેલી છે :

```
case variable_name in
value1)
    Command1
    Command 2
    ....
    ;;
value 2)
    Command 1
    Command 2
    ....
    ;;
*)
Command 1
    Command 2
    ....
    ;;
esac
```

case વિધાનની વાક્યરચના

નોંધ :

case વિધાનમાં પસંદગી માટે કિંમત તરીકે આંકડાકીય, અક્ષર પ્રકારની કે સ્ટ્રિંગ સ્વરૂપે આપી શકાય છે. સ્ટ્રિંગ કિંમતોને case વિધાનમાં એક અવતરણચિહ્ન (' ') માં આવરીને લખવી જરૂરી છે. ઉદાહરણ તરીકે, જો આપણે case વિધાનમાં કિંમત તરીકે *abc* સ્ટ્રિંગ સ્વીકારવા માંગતા હોઈએ, તો તેને 'abc' સ્વરૂપે લખવી જોઈએ.

પુનરાવર્તન કરવું (Handling Repetition)

એડમિનિસ્ટ્રેટરના સામાન્ય કાર્યોમાં ડિસ્કની જગ્યાની સાચવણીનો પણ સમાવેશ થાય છે. હવે આપણે એક સ્ક્રિપ્ટ બનાવીએ, જે શૂન્ય કદ ધરાવતી ફાઈલો શોધીને તેને દૂર કરવા માટે એડમિનિસ્ટ્રેટરને મદદરૂપ બને. આ માટેની સ્ક્રિપ્ટ નીચે આપેલી છે :

```
#Script 19: Script to delete zero sized files.
echo -n "Enter directory name : "
read dname
if [ ! -d $dname ]
then
    echo Directory $dname does not exist.
```

```

else
ctr=0
for i in `find "$dname/" -type f -size 0c`
do
rm $i
echo "File $i" : deleted"
ctr=`expr $ctr + 1`
done
if [ $ctr -gt 0 ]
then
echo "$ctr zero sized files have been deleted."
else
echo "No zero sized files present in directory."
fi
fi

```

આ સ્ક્રિપ્ટમાં *for i in 'find "\$dname/" -type f -size 0c'* વિધાનનો ઉપયોગ કર્યો છે, તે ધ્યાનમાં લો. આ વિધાનની મદદથી કેટલીક ક્રિયાઓ વારંવાર કરી શકાય છે. આકૃતિ 8.11 સ્ક્રિપ્ટ-19નું પરિણામ દર્શાવે છે.

```

File Edit View Terminal Help
Enter directory name : LinuxScript
File LinuxScript/test : deleted
File LinuxScript/test1 : deleted
File LinuxScript/test2 : deleted
3 zero sized files have been deleted.
administrator@ubuntu:~$

```

આકૃતિ 8.11 : સ્ક્રિપ્ટ-19નું પરિણામ

આ પ્રકારનાં કાર્યો કરવા માટેની સ્ક્રિપ્ટ લખતી વખતે કેટલીક ક્રિયાઓ અનેક વાર કરવાની જરૂર પડે છે. એકસરખા કમાન્ડ અનેક વખત પુનરાવર્તિત કરવાની પ્રક્રિયાને ‘લૂપિંગ’ (looping) કહે છે. પુનરાવર્તિત કાર્યો કરવા માટે લિનક્સમાં *for*, *while* અને *until* એમ ત્રણ કી-વર્ડ પૂરા પાડવામાં આવ્યા છે.

સ્ક્રિપ્ટ-19માં *for* વિધાનનો ઉપયોગ કરવામાં આવ્યો છે. *for* લૂપ તેના વિધાનમાં કિંમતોની યાદી ઉમેરવાની છૂટ આપે છે. યાદીમાં આવેલ દરેક કિંમત માટે લૂપનો અમલ કરવામાં આવે છે. *for* વિધાનની સામાન્ય વાક્યરચના નીચે દર્શાવી છે :

```
for control-variable in value1, value2, value3.....
do
    command 1
    command 2
    command 3
done
```

Forની વાક્યરચના

ફાઇલનો બેક-અપ લેવાની પ્રક્રિયા પણ એડમિનિસ્ટ્રેટર દ્વારા પાર પાડવામાં આવતી અન્ય એક સામાન્ય પ્રવૃત્તિ છે. ધારો કે એડમિનિસ્ટ્રેટરને નિશ્ચિત પ્રકારની ફાઇલનો બેક-અપ લેવાની જરૂર પડે છે. આ કિસ્સામાં એક-એક ફાઇલનો બેકઅપ લેવાનું કાર્ય યોગ્ય નહીં ગણાય. અન્ય સ્થળ પર જે-તે ફાઇલની નકલ કરવાથી પણ સંગ્રાહક જગ્યાનો વ્યય થશે. આ કિસ્સામાં એડમિનિસ્ટ્રેટર એવી સ્ક્રિપ્ટનો ઉપયોગ કરી શકે જે સૌ પ્રથમ ફાઇલ જ્યાં આવેલી હોય તે સ્થળે એક ફોલ્ડરની રચના કરશે. ત્યાર પછી જે ફાઇલનો બેક-અપ લેવાનો હોય તેની તે ફોલ્ડરમાં નકલ કરવામાં આવશે. ડિરેક્ટરીને ત્યાર પછી સંકુચિત (compressed) કરવામાં આવશે અને અંતમાં નવા સ્થાને ખસેડવામાં આવશે. નીચે આપવામાં આવેલી સ્ક્રિપ્ટ આ પ્રક્રિયા પાર પાડશે.

#Script 20: Script to backup and compress desired files from current location.

```
clear
dat=`date +"%d_%m_%Y"`
bdir=backup_$(dat)
if [ ! -d $bdir ]
then
    mkdir $bdir
else
    echo "Directory with name $bdir already exist."
    exit 0
fi
echo -n "Enter the extension of the files to backup: "
read fextn
ctr=0
for i in `ls -l *.$fextn`
do
    cp $i ./$(bdir)
    ctr=`expr $ctr + 1`
done
```



```

if [ $ctr -gt 0 ]
then
tar -czf $bdir.tar $bdir
cd $bdir
rm -r *.*
cd ..
rmdir $bdir
echo "All files with extension .$fextn stored in $bdir.tar"
else
rmdir $bdir
echo "No files with the extension found."
fi

```

સ્ક્રિપ્ટને *script20.sh* નામથી સેવ કરો. હવે સ્ક્રિપ્ટનું કાર્ય સમજાવે. શરૂઆતમાં *dat* અને *bdir* નામના બે ચલ વ્યાખ્યાયિત કરવામાં આવ્યા છે. આપેલ સ્વરૂપમાં વર્તમાન તારીખનો સંગ્રહ કરવા માટે *dat* ચલનો ઉપયોગ કરવામાં આવ્યો છે. ઉદાહરણ તરીકે, જો વર્તમાન તારીખ 21 ફેબ્રુઆરી, 2013 હોય, તો *dat* ચલમાં 21_02_2013 વિગતનો સંગ્રહ કરવામાં આવશે. ત્યાર પછી *bdir* ચલમાં backup_21_02_2013 કિંમત ઉમેરવામાં આવશે. ત્યાર પછી આ ડિરેક્ટરી અસ્તિત્વમાં છે કે નહીં તે ચકાસવામાં આવશે. જો ડિરેક્ટરી અસ્તિત્વમાં નહીં હોય, તો તેની રચના કરવામાં આવશે, નહીં તો ‘ડિરેક્ટરી પહેલેથી અસ્તિત્વમાં છે’ તે પ્રકારનો સંદેશ દર્શાવી સ્ક્રિપ્ટ પૂર્ણ કરવામાં આવશે. જો ડિરેક્ટરી બનાવવામાં આવે તો ઉપયોગકર્તાને ફાઇલનું અનુલંબન (extension) ઉમેરવાનું કહેવામાં આવશે. સ્ક્રિપ્ટ ત્યાર બાદ આ પ્રકારની ફાઇલોની વર્તમાન ડિરેક્ટરીમાં શોધ ચલાવશે અને જો ફાઇલ મળશે તો તેની નકલ બેક-અપ ડિરેક્ટરીમાં કરવામાં આવશે. તમામ ફાઇલની નકલ થઈ ગયા પછી tar કમાન્ડની મદદથી બેક-અપ ડિરેક્ટરીનું સંકોચન (compression) કરવામાં આવશે. આ પ્રક્રિયા માટે tar **-czf \$bdir.tar \$bdir** કમાન્ડનો ઉપયોગ કરવામાં આવ્યો છે. અહીં આપણે backup_currentdate.tar નામની ફાઇલ બનાવી છે. ત્યારપછી બેક-અપ ડિરેક્ટરી ખાલી કરી દૂર કરવામાં આવશે. જો આપેલ અનુલંબન ધરાવતી એકપણ ફાઇલ શોધી ન શકાય, તો યોગ્ય સંદેશ દર્શાવવામાં આવશે. હવે પછી એડમિનિસ્ટ્રેટર ઈચ્છશે, તો આ સંકુચિત ફાઇલને અન્ય અપેક્ષિત જગ્યાએ ખસેડી શકશે. સંકુચિત ફાઇલનું વિસ્તરણ (uncompression) કરવા માટે **tar -xvf filename** કમાન્ડનો ઉપયોગ કરી શકાય.

પુનરાવર્તન માટે while વિધાન (Repetition : while statement)

લૂપિંગ માટે *while* વિધાનનો ઉપયોગ પણ થઈ શકે છે. જ્યાં સુધી ‘એક્સ્પ્રેશન’ સ્વરૂપે આપવામાં આવેલી શરત સાચી ઠરે, ત્યાં સુધી *do* અને *done* કી-વર્ડ વચ્ચે આવેલાં તમામ વિધાનોનું પુનરાવર્તન કરવાનું કાર્ય *while* વિધાન કરે છે. હવે આપણે એક એવી સ્ક્રિપ્ટ બનાવીએ જે એડમિનિસ્ટ્રેટરને ડિરેક્ટરીમાંથી ચોક્કસ સંખ્યામાં ફાઇલોને દૂર કરવામાં મદદરૂપ બને.

```

#Script 21: Script to delete specified number of files from a directory.
clear
echo -n "Enter the name of directory from where you want to delete: "

```

```

read dname
if [ -d $dname ]
then
cd $dname
echo -n "Enter the number of files you want to delete: "
read fdel
ctr=1
while [ $ctr -le $fdel ]
do
echo -n "Enter the name of the file to be deleted: "
read fname
if [ -f $fname ]
then
rm $fname
echo "$fname deleted successfully."
else
echo "File with name $fname not found."
fi
ctr=`expr $ctr + 1`
done
else
echo "Directory $dname does not exist."
fi
cd ..

```

સ્ક્રિપ્ટને *script21.sh* નામ આપી સેવ કરો. આ સ્ક્રિપ્ટનું કાર્ય સમજાવે. શરૂઆતમાં ઉપયોગકર્તાને એક ડિરેક્ટરીનું નામ પસંદ કરવાનું કહેવામાં આવશે. આ કિંમત *dname* નામના ચલમાં ઉમેરાશે. ત્યાર પછી તે ડિરેક્ટરી અસ્તિત્વમાં છે કે નહીં, તેની ચકાસણી કરવામાં આવશે. જો ડિરેક્ટરીનું અસ્તિત્વ હશે, તો તેને વર્તમાન ડિરેક્ટરી બનાવી ઉપયોગકર્તા પાસેથી દૂર કરવામાં આવનાર કુલ ફાઇલોની સંખ્યા માંગવામાં આવશે. ત્યાર પછી ફાઇલને શોધવા માટે while લૂપ શરૂ કરવામાં આવશે. જો ફાઇલ મળશે તો તેને દૂર કરવામાં આવશે, નહીં તો 'ફાઇલ મળેલ નથી' તેવો સંદેશ દર્શાવવામાં આવશે. જ્યાં સુધી ctr ચલની કિંમત ઉપયોગકર્તાએ આપેલ કિંમત કરતાં ઓછી અથવા બરાબર હોય, ત્યાં સુધી આ લૂપનો અમલ થતો રહેશે. એક વાર પ્રક્રિયા પૂર્ણ થયે પિતૃડિરેક્ટરીમાં પાછા ફરવામાં આવશે. while લૂપની વાક્યરચના નીચે દર્શાવેલ છે :

```
while [ test_condition ]
do
    command or set of commands
done
```

while લૂપની વાક્યરચના

પુનરાવર્તન માટે until વિધાન (Repetition: until statement)

કાર્યોના પુનરાવર્તિત અમલ માટે એક અન્ય વિધાન - *until*નો ઉપયોગ પણ કરી શકાય છે. *until* લૂપ એ *while* લૂપની જેમ જ કાર્ય કરતું બંધારણ છે, પરંતુ તે વિધાનોના અમલનું કાર્ય ત્યાં સુધી પુનરાવર્તિત કરે છે, જ્યાં સુધી આપેલ શરત ખોટી (false) હોય. જ્યારે શરત સાચી (true) હોય, ત્યાં સુધી વિધાનોનો અમલ કરવાનું કાર્ય *while* લૂપનું છે.

નિર્ણાયક વિધાનો અને લૂપિંગ બંધારણનો ઉપયોગ કરી શેલસ્ક્રિપ્ટ કેવી રીતે લખી શકાય છે, તે આપણે જોયું. ઉપર ચર્ચા કરી તેવાં કેટલાંક બંધારણોનો ઉપયોગ કરી સ્ક્રિપ્ટ 22 લખવામાં આવી છે. તેમાં વર્તમાન ડિરેક્ટરીમાં આવેલ ફાઇલની યાદી દર્શાવવા માટે, પાસવર્ડ બદલવા માટે, વર્તમાન તારીખ અને સમય દર્શાવવા માટે અને ફાઇલમાંથી શબ્દ શોધવા માટે *until* લૂપનો ઉપયોગ કરી એક મેનુ-આધારિત સ્ક્રિપ્ટ લખવામાં આવી છે.

```
#Script 22: Script to perform operations till user decides to exit.
```

```
choice=y
```

```
until [ $choice = n ]
```

```
do
```

```
clear
```

```
echo "....."
```

```
echo "    Choose an option from menu given below    "
```

```
echo "....."
```

```
echo "a: List of files and directories in a current directory."
```

```
echo "b: Display current working directory"
```

```
echo "c: Display current date and time"
```

```
echo "d: Searching a word from file"
```

```
echo "e: Exit"
```

```
echo " "
```

```
echo "....."
```

```
echo -n "Enter your choice [a-e]: "
```

```
read ch
```

```
case $ch in
```

```
a)
```

```

ls -l
;;
b)
echo "You are working in `pwd`"
;;
c)
echo "Current date and time is is `date`"
;;
d)
echo -n "Enter the word to be searched: "
read word
echo -n "Enter the file name from which the word is to be searched: "
read file
if [ -f $file ]
then
grep $word $file
else
echo -n "File with name $file does not exist."
fi
;;
e)
exit
;;
*)
echo "Incorrect choice, try again.."
;;
esac
echo -n "Do you want to continue? : "
read choice
done

```

સ્ક્રિપ્ટને *script22.sh* નામ આપી સેવ કરો. જ્યારે આ સ્ક્રિપ્ટનો અમલ કરવામાં આવશે, ત્યારે ઉપયોગકર્તા સમક્ષ એક મેનુ પ્રદર્શિત કરી પસંદગીની માંગણી કરવામાં આવશે. પસંદગી અનુસાર case વિધાનની પ્રક્રિયાનો અમલ કરવામાં આવશે. દર વખતે જુદી-જુદી પસંદગીઓ દ્વારા મળતું પરિણામ જુઓ. સ્ક્રિપ્ટનો વારંવાર અમલ કરવામાં આવે છે. પરંતુ જો ઉપયોગકર્તા *e* વિકલ્પ પસંદ કરે, તો caseના exit વિધાનનો અમલ કરવામાં આવે છે અથવા “Do you want to continue?” પ્રશ્નના ઉત્તર સ્વરૂપે *n* આપવામાં આવે, તો સ્ક્રિપ્ટનો અમલ અટકાવવામાં આવે છે. સ્ક્રિપ્ટ-22નું પરિણામ આકૃતિ 8.12માં દર્શાવ્યું છે.

```
File Edit View Terminal Help
.....
          Choose an option from menu given below
.....
a: List of files and directories in a current directory.
b: Display current working directory
c: Display current date and time
d: Searching a word from file
e: Exit
.....
Enter your choice: [a-e]   c
Current date and time is is Fri Feb 22 14:36:21 EST 2013
Do you want to continue? : n
administrator@ubuntu:~$
```

આકૃતિ 8.12 : સ્ક્રિપ્ટ-22નું પરિણામ

સ્ક્રિપ્ટમાં વિધેય (Functions in script)

લિનક્સ શેલસ્ક્રિપ્ટમાં ‘વિધેયો’ (functions)ની રચના કરવા માટેની પણ સુવિધા ઉપલબ્ધ છે. શેલસ્ક્રિપ્ટમાં આવેલ નાના વિભાગ (subscripts) ને ફંક્શન કહે છે. સ્ક્રિપ્ટને વધુ વિભાગીય (modular) બનાવવા માટે તેનો ઉપયોગ કરવામાં આવે છે. ફંક્શનની મદદથી સ્ક્રિપ્ટની વાચનક્ષમતા સુધારી શકાય છે. શેલસ્ક્રિપ્ટમાં આવેલ ફંક્શન કોઈ કિંમત પરત કરતા નથી, પરંતુ ‘સ્ટેટસ કોડ’ (status code) પરત કરે છે. હવે એક એવી સ્ક્રિપ્ટ બનાવીએ, જે ઉપયોગકર્તાને વર્તમાન તારીખમાં બનાવવામાં આવેલ કુલ ફાઈલોની સંખ્યા ગણવામાં કે કોઈ ફાઈલ છેલ્લે ક્યારે સુધારવામાં આવી છે, તે જાણવામાં મદદરૂપ બને.

#Script 23: Script to show use of function.

```
file_today(){
    cur_date=`date +%Y-%m-%d`
    cnt=`ls -l tr | grep "$cur_date" | wc -l`
    echo "Current date is : "$cur_date
    echo "No. of files created today : "$cnt
}
```

```

modified_today(){
if [ -f "$1" ]
then
stat -c %y "$1"
else
echo ""$1" does not exist"
fi
}
choice=y
until [ $choice = n ]
do
clear
echo "....."
echo "    Choose an option from menu given below    "
echo "....."
echo "a: List of files created today."
echo "b: Display last file modification date."
echo "c: Exit"
echo " "
echo "....."
echo -n "Enter your choice [a-c]: "
read ch
case $ch in
a)
file_today
;;
b)
echo -n "Enter a file name: "
read fname
modified_today $fname
;;

```

```

c)
exit
;;
*)
echo "Incorrect choice, try again.."
;;
esac
echo -n "Do you want to continue? : "
read choice
done

```

સ્ક્રિપ્ટને *script23.sh* નામ આપી સેવ કરો. સ્ક્રિપ્ટ 23માં `file_today()` અને `modified_today()` નામનાં બે ફંક્શનનો ઉપયોગ કર્યો છે તેની નોંધ લો. ચલના નામની પાછળ આવતા કૌંસ () તે ફંક્શન હોવાનો નિર્દેશ કરે છે. જ્યારે ઉપયોગકર્તા *a* પસંદ કરે છે, ત્યારે `file_today()` ફંક્શનને બોલાવી (call કરી) અમલમાં મૂકવામાં આવે છે. જેમાં વર્તમાન તારીખમાં બનાવવામાં આવેલ ફાઇલની સંખ્યા શોધવા માટેનો કોડ લખવામાં આવ્યો છે. આ જ રીતે ઉપયોગકર્તા જો *b* પસંદ કરે તો તેને ફાઇલનું નામ પૂછવામાં આવે છે. આ નામ પછી `modified_today()` ફંક્શનને આપવામાં આવશે, જેના દ્વારા ફાઇલનું અસ્તિત્વ તપાસવામાં આવશે. જો ફાઇલ અસ્તિત્વમાં હશે, તો તેને છેલ્લે સુધારવામાં આવી હોય તે તારીખ દર્શાવવામાં આવશે, અન્યથા યોગ્ય સંદેશ દર્શાવવામાં આવશે. આકૃતિ 8.13માં સ્ક્રિપ્ટ 23નું પરિણામ દર્શાવ્યું છે.

```

File Edit View Terminal Help
.....
Choose an option from menu given below
.....
a: List of files created today.
b: Display last file modification date.
c: Exit
.....
Enter your choice [a-c]: a
Current date is : 2013-02-22
No. of files created today : 12
Do you want to continue? : n
administrator@ubuntu:~$

```

આકૃતિ 8.13 : સ્ક્રિપ્ટ-23નું પરિણામ

સારાંશ

આ પ્રકરણમાં આપણે સિસ્ટમના વહીવટ (એડમિનિસ્ટ્રેશન) માટે શેલસ્ક્રિપ્ટનો ઉપયોગ કેવી રીતે કરી શકાય તે શીખ્યા. શેલસ્ક્રિપ્ટમાં નિર્ણાયક વિધાનો અને લૂપિંગ બંધારણોનાં ઉપયોગ વિશે માહિતી મેળવી. ફંક્શનની મદદથી શેલ સ્ક્રિપ્ટ કેવી રીતે લખી શકાય એ પણ આપણે જોયું. આમ, શેલસ્ક્રિપ્ટ દ્વારા અનેક આંતરપ્રસ્થાપિત (inbuilt) કમાન્ડનું સંયોજન કરી શકાય છે અને શેલસ્ક્રિપ્ટ કોઈ ઉચ્ચસ્તરીય ભાષાની સમકક્ષતા પૂરી પાડે છે.

સ્વાધ્યાય

1. યોગ્ય ઉદાહરણનો ઉપયોગ કરી શરતી અમલીકરણને શેલ સ્ક્રિપ્ટ સંબંધે સમજાવો.
2. શેલસ્ક્રિપ્ટના case વિધાનને તેમાં ઉપયોગમાં લઈ શકાતી ભાત (pattern)ની યાદી સાથે સમજાવો.
3. While લૂપ વિશે સમજાવો.
4. Until લૂપ વિશે સમજાવો.

5. હેતુલક્ષી પ્રશ્નો

- (1) ફાઈલને અમલ કરવા માટેની (execute) મંજૂરી આપવા માટે નીચેનામાંથી કયા કમાન્ડનો ઉપયોગ કરવામાં આવે છે ?
(a) grep (b) chmod
(c) ls (d) x
- (2) case વિધાનમાં નિયંત્રણના પ્રવાહને અવરોધવા નીચેનામાંથી કઈ નિશાનીનો ઉપયોગ કરવામાં આવે છે ?
(a) ** (b) ;;
(c) ++ (d) >>
- (3) case વિધાનનો અંત દર્શાવતો કી-વર્ડ કયો છે ?
(a) end-case (b) end case
(c) esac (d) stop-case
- (4) શરત સાચી (true) હોય, ત્યાં સુધી વિધાનોનો અમલ કરતું લૂપ કયું છે ?
(a) while (b) until
(c) for (d) case
- (5) વિધાનમાં કિંમતોની યાદી ઉમેરવાની સુવિધા નીચેનામાંથી કયા લૂપ દ્વારા પૂરી પાડવામાં આવે છે ?
(a) while (b) until
(c) for (d) if
- (6) case વિધાનમાં કયો અક્ષર પૂર્વનિર્ધારિત caseનો નિર્દેશ કરે છે ?
(a) * (b) +
(c) d (d) આપેલમાંથી એક પણ નહીં.
- (7) આપેલ ફાઈલ 'રીડ-ઓન્લી' (read only) છે કે નહીં તે તપાસવા માટે નીચેનામાંથી કયું વિધાન સાચું છે ?
(a) test -read filename (b) check -read filename
(c) test -r filename (d) check -r filename
- (8) if વિધાનનો અંત દર્શાવવા શેલસ્ક્રિપ્ટમાં નીચેનામાંથી શેનો ઉપયોગ થાય છે ?
(a) end-if (b) fi
(c) } (બંધ થતો છગડિયો કૌંસ) (d) તેને કોઈ અંતવિધાન નથી.
- (9) લિનક્સમાં નીચેનામાંથી કયો ઓપરેટર 'થી ઓછું' (less than) દર્શાવે છે ?
(a) < (b) lessthan
(c) lt (d) -lt
- (10) if વિધાનમાં શરત માટે ઉપયોગમાં લેવામાં આવતા ચોરસ કૌંસના સ્થાને નીચેનામાંથી શેનો ઉપયોગ કરી શકાય છે ?
(a) છગડિયા કૌંસ (b) test કમાન્ડ
(c) check કમાન્ડ (d) એક પણ નહીં

નીચેનાં કાર્યો કરવા માટેની શેલસ્ક્રિપ્ટ બનાવો.

- (a) ઉપયોગકર્તા પાસેથી બે ફાઈલનાં નામ મેળવો. સ્ક્રિપ્ટ દ્વારા તપાસો કે બન્ને ફાઈલની વિગતો સમાન છે કે નહીં. જો બન્ને ફાઈલની વિગતો સરખી હોય તો બીજી ફાઈલને દૂર કરો.
- (b) કમાન્ડલાઈન આર્ગ્યુમેન્ટ દ્વારા આપવામાં આવેલા પાથમાં આપેલ દરેક સબડિરેક્ટરીમાં આવેલ તમામ ઘટકોની સંખ્યા ગણી દર્શાવો.
- (c) ડિરેક્ટરીમાં આવેલ 1000 બાઈટથી વધુ કદ ધરાવતી તમામ ફાઈલના નામની યાદી તેના કદ સાથે દર્શાવો. (ડિરેક્ટરનું નામ શેલસ્ક્રિપ્ટને આર્ગ્યુમેન્ટ સ્વરૂપે આપવામાં આવશે.)
- (d) ફાઈલનું નામ બદલવા માટેની સ્ક્રિપ્ટ લખો.
- (e) ઉપયોગકર્તાની જરૂરિયાત મુજબ ફાઈલના તમામ લખાણને કેપિટલ કે સ્મોલ અક્ષરોમાં ફેરવો.
- (f) સિસ્ટમમાં ઉપલબ્ધ શેલનાં નામ શોધો અને વર્તમાન શેલનું નામ દર્શાવો.
- (g) વર્તમાન ડિરેક્ટરીમાં આવેલ ન્યૂનતમ કદ ધરાવતી ફાઈલ શોધો.





રિજિટલ ટેકનોલોજીના આગમન અને ફેલાવા સાથે વર્તમાન સ્થિતિમાં લગભગ તમામ માહિતીનો સંગ્રહ અને સંચાલન ઇલેક્ટ્રોનિક સ્વરૂપે થાય છે. નાની અને મધ્યમ સંસ્થાઓનાં રોજિંદા વ્યવહારોને ઇન્ફર્મેશન સિસ્ટમ તરીકે ઓળખાતી સિસ્ટમમાં સંગ્રહ કરવામાં આવે છે. ઇન્ફર્મેશન સિસ્ટમમાં વપરાશકર્તાઓની વૃદ્ધિ ઘણી ઝડપે વધી રહી છે. ઉપયોગકર્તા પોતે ગમે તે જગ્યાએ હોય અને ડેટાનો સંગ્રહ પણ કોઈ જગ્યાએ કર્યો હોય છતાં ડેટા પાછો મેળવવામાં ઘણી સાનુકૂળતાની અપેક્ષા ઉપયોગકર્તા રાખે છે. સંસ્થાઓ વૈશ્વિક બનવા માટે ઇન્ફર્મેશન સિસ્ટમ હાલમાં સંસ્થાના વ્યવહારોનો સંગ્રહ અને જાળવણી પૂરતી મર્યાદિત રહેતી નથી કે જે સંસ્થાના રજિસ્ટર્સ/ફાઇલ્સની જાળવણીને બદલે સમજવામાં આવતી હતી. સર્ચએન્જિન, આરક્ષણ પદ્ધતિ, ઓનલાઇન બેન્કિંગ, ઓનલાઇન ટેક્સની ચુકવણી અને સ્ટોક-ટ્રેડિંગ સિસ્ટમ એવી સિસ્ટમ્સનાં ઉદાહરણો છે, જેમાં એક જ સમયે એકસાથે કરોડો ઉપયોગકર્તા તેનો ઉપયોગ કરે છે. આજે ઇન્ફર્મેશન સિસ્ટમ્સનો ઉપયોગ અનિવાર્ય બની ગયો છે. આપણે દરેક વ્યક્તિ સીધી કે આડકતરી રીતે એક અથવા બીજી ઇન્ફર્મેશન સિસ્ટમનો ઉપયોગ કરીને તેના વપરાશકર્તાની ભૂમિકા ભજવીએ છીએ. ડોક્ટર્સ, એન્જિનિયર્સ, શોપિંગ સ્ટોરના માલિકો, બ્રોકિંગ એજન્ટો, કેળવણીકારો અને વિદ્યાર્થીઓ વગેરેમાંથી કોઈ પણ તેમાંથી બાકાત રહ્યા નથી. આજે નિર્ણયો લેવામાં ઇન્ફર્મેશન સિસ્ટમ એક મજબૂત મદદરૂપ સાધન બની રહ્યું છે. કોઈ પણ શક્ય શ્રેષ્ઠ નિર્ણય લેવા માટે દરેક વ્યક્તિને સમયસર સાચી, નિશ્ચિત અને સંબંધિત માહિતીની જરૂરત રહે છે. અહીં એ સ્પષ્ટ છે કે કોઈ પણ સચોટ નિર્ણયનો આધાર આપણી પાસે ઉપલબ્ધ માહિતીની ચોકસાઈ ઉપર છે. ઇન્ફર્મેશન એ આઉટપુટ છે, જે કાચા (રૉ) ડેટા ઉપર પ્રક્રિયા કરીને પેદા કરવામાં આવે છે. આથી, ચાલો આપણે પહેલાં બે અગત્યના શબ્દો ડેટા અને ઇન્ફર્મેશનને સમજીએ.

ડેટા અને ઇન્ફર્મેશન (Data and Information)

ઘણા સમયે ડેટા અને ઇન્ફર્મેશન શબ્દોનો ઉપયોગ અદલાબદલી કરીને કરવામાં આવે છે. આથી બન્ને શબ્દોનો સ્પષ્ટ અર્થ સમજવો જરૂરી બને છે. ચાલો, આપણે પહેલાં તેની શક્ય વ્યાખ્યા દ્વારા ડેટા અને ઇન્ફર્મેશન શબ્દોને સમજીએ.

“દરેક પ્રકારની હકીકત, આંકડાઓ અને વ્યક્તિઓ વિષેની વિગત, જગ્યાઓ, વસ્તુઓ અથવા પ્રસંગોને ડેટા કહેવામાં આવે છે.” ડેટા જુદા-જુદા કોઈ પણ સ્વરૂપમાં હોઈ શકે. તે મૌખિક, લેખિત, કમ્પ્યૂટીકરણ કરેલા અથવા કમ્પ્યૂટરીકરણ ન કરેલા પણ હોઈ શકે. ડેટા તેના મૂળ સ્વરૂપમાં ખાસ ઉપયોગી હોતા નથી. ડેટા ઉપર કોઈ યોગ્ય પ્રક્રિયા કરી તેને ઉપયોગી અને અર્થપૂર્ણ ઇન્ફર્મેશનના સ્વરૂપમાં પેદા કરી શકાય. નિર્ણય લેવાની પ્રક્રિયામાં ડેટાની અગત્યતા સમજીને અનેક સંસ્થાઓ તેને ધંધાની એક અગત્યની સંપત્તિ માને છે.

બીજી બાજુ ઇન્ફર્મેશનને વ્યાખ્યાયિત એ રીતે કરી શકાય કે “ડેટા ઉપર પ્રક્રિયા કરીને મેળવેલું જરૂરી પરિણામ”. આનો અર્થ એ થાય કે ઇન્ફર્મેશન બીજું કંઈ નથી પણ પ્રક્રિયા કરેલાં ડેટા છે. ઇન્ફર્મેશન આપણને ઉપયોગી છે, કારણકે તે નિર્ણય લેવા માટેનો પાયો બનાવે છે. સાચી અને સંબંધિત ઇન્ફર્મેશનના આધારે સમયસર લીધેલા નિર્ણય આપણને ખૂબ ફાયદાકારક હોઈ શકે. બીજી બાજુ અધૂરી કે ખોટી ઇન્ફર્મેશનના આધારે લીધેલા નિર્ણય ધંધાને ખરાબ અસર પણ કરી શકે.

ચાલો, આપણે ઉદાહરણ દ્વારા ડેટા અને ઇન્ફર્મેશનના ખ્યાલને સ્પષ્ટ રીતે સમજીએ. ધારોકે તમે અને તમારી સ્કૂલની ટીમ સ્કૂલના દરેક વિદ્યાર્થીને જન્મદિવસે શુભેચ્છા મોકલવાનો નિર્ણય લે છે. આ વિચારના અમલ માટે તમારે દરેક વિદ્યાર્થીની જન્મતારીખ ભેગી કરવાની જરૂર પડે છે, જેમકે 10મી સપ્ટેમ્બર, 1998. આ ઉદાહરણમાં ભેગી કરેલી જન્મતારીખો ડેટાનું કામ આપે છે. જો આજે 10મી સપ્ટેમ્બર હોય, તો આ તારીખે જે છોકરા કે છોકરીની જન્મતારીખ હોય તેનાં નામ તમે શોધશો. તમારી યાદીમાં નામનો સેટ ઇન્ફર્મેશન તરીકે કામ આપશે.

ડેટાબેઝ અને ડેટામોડેલ (Database and Data Model)

આપણે ડેટા અને ઇન્ફર્મેશન બાબત અહીં શીખ્યા પછી ચાલો આપણે ડેટાબેઝ તરફ નજર કરીએ. આપણે બધા આપણી રોજિંદી જિંદગીમાં એક અથવા વધારે પ્રકારના ડેટાબેઝનો ઉપયોગ કરીએ છીએ. આપણે આપણા મિત્રો અને સંબંધીઓનાં

ફોનનંબર અને સરનામાં ધરાવતી ડાયરી ઘરે રાખીએ છીએ. એ જ પ્રકારે સ્કૂલમાં દાખલ કરેલા બધા વિદ્યાર્થીઓની માહિતી તમારી સ્કૂલમાં રાખતા હશો. દરેક વિદ્યાર્થીનો સ્કૂલમાં હાજરી બાબત રેકૉર્ડ સ્કૂલ વડે જાળવવામાં આવતો હશે. ભારત સરકાર આધાર પ્રોજેક્ટમાં દરેક નાગરિકને એક અજોડ ઓળખનંબર આપી રહી છે. આ રીતે દરેક વ્યક્તિની માહિતી સાચવવામાં આવે છે. પાસપોર્ટ ઓફિસ ઈસ્યુ કરેલા પાસપોર્ટની માહિતી જાળવે છે. ડૉક્ટર્સ દરેક દર્દીઓનો મેડિકલ ઇતિહાસ જાળવે છે. રેલવેખાતું દરેક મુસાફરની વિગત જાળવે છે. આ બધાં રેકૉર્ડનાં ઉદાહરણો છે.

ડેટાબેઝની એક સામાન્ય વ્યાખ્યા પ્રમાણે “ડેટાબેઝ એ એકબીજા સાથે સંબંધિત સુવ્યવસ્થિત રીતે સંગૃહીત ડેટા-આઈટમ્સનો જથ્થો છે.” આપણે ઉપર ચર્ચા કરી તે પ્રમાણે આપણે ડેટાનો ઉપયોગ કરીએ છીએ, તેથી તેનો સંગ્રહ કરવામાં આવે છે. તેથી આપણી ડેટાબેઝની વ્યાખ્યામાં ‘સંબંધિત’ અને ‘સુવ્યવસ્થિત’ શબ્દો અત્યંત મહત્ત્વપૂર્ણ છે. તમે ડેટાના સંગ્રહને તમારા રૂમમાં વસ્તુઓને વ્યવસ્થિત ગોઠવવા સાથે સરખાવી શકો છો. તમે તમારાં માતા-પિતા અને શિક્ષકો તરફથી તમારો સામાન એક યોગ્ય અને પૂર્વનિર્ધારિત જગ્યાએ રાખવા બાબત સૂચનો મેળવતાં હશો. તમે સ્કૂલબેંગ, સ્ટેશનરી અને બુક્સ વગેરે માટે પણ જગ્યા નક્કી કરતાં હશો. તમે બુકશેલ્ફના વિવિધ ખંડ/ભાગને નામ પણ આપતા હશો. ઉદાહરણ તરીકે વિજ્ઞાન, સામાન્ય જ્ઞાન વગેરે. જો તમારા પિતા માનવશરીર ઉપર કોઈ એક બુક ભેટ આપે, તો તમે તેને વિજ્ઞાન વિભાગમાં રાખવાનું નક્કી કરો છો. આથી આગામી સમયમાં જ્યારે તમે તે બુક વાંચવા ઈચ્છો છો ત્યારે એ બુક શોધવી તમારા માટે ખૂબ જ સરળ બની જાય છે. વસ્તુઓની તર્કસંગત (લોજિકલ) ગોઠવણી આપણે જ્યારે કોઈ વસ્તુની જરૂર હોય ત્યારે તે શોધવાના કાર્યને હંમેશાં સરળ બનાવે છે.

તમારી બુક્સની જેમ કબાટમાં કપડાંની તર્કસંગત ગોઠવણી પણ જરૂરી છે. જોકે તમે અહીં નોંધ કરી હશે કે બુકશેલ્ફ અને ગોઠવણી સામાન્ય રીતે અલગ-અલગ રીતે હોઈ શકે છે. ઘરેણાં અને રસોડાનાં વાસણોનો સંગ્રહ કરવાનું નિયત સ્થળ અલગ માળખું ધરાવે છે. ઉપરનાં ઉદાહરણો દર્શાવે છે કે જે વસ્તુઓનો સંગ્રહ કરવાનો હોય, તેનો આધાર રાખીને અભેરાઈઓના માળખાની રચના કરવી જોઈએ.

ઉપર ઉદાહરણો આપ્યાં છે, એ રીતે ડેટાબેઝ ચોક્કસ નિયમો અનુસાર તૈયાર કરવામાં આવે છે. ડેટાબેઝનું અંતર્ગત માળખું ડેટામોડેલ તરીકે ઓળખાય છે. ડેટામોડેલ એ ડેટાનો સંગ્રહ અને પુનઃપ્રાપ્તિની રીત દર્શાવે છે. હાયરારકિલ ડેટામોડેલ, નેટવર્ક ડેટામોડેલ અને રીલેશનલ ડેટામોડેલ જેવા વિવિધ પ્રકારનાં ડેટામોડેલ ઉપલબ્ધ છે. અહીં આપણે આ અભ્યાસક્રમમાં રિલેશનલ મોડેલની ચર્ચા કરીશું. બજારમાં અનેક વિવિધ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ ઉપલબ્ધ છે, જે ઉપર જણાવેલા ડેટામોડેલમાંથી એક ઉપર આધારિત હોય છે. ચાલો, હવે આપણે ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ શું છે તે સમજીએ.

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (Database Management Systems)

કમ્પ્યુટીકરણ કરેલા ડેટાબેઝ સાથે કામ કરવા માટે આપણે સૌપ્રથમ ડેટાબેઝને વ્યાખ્યાયિત કરવો પડે છે. ત્યાર પછી ડેટાબેઝની રચના કરવી પડે છે. જોકે ડેટાબેઝ તૈયાર કરવો એ જ માત્ર કાર્ય નથી. ડેટાબેઝમાં આજ સુધીની તાજી માહિતી તૈયાર રાખવા માટે નીચે જણાવેલી મૂળભૂત પ્રક્રિયાઓ શક્ય એટલી સરળતાથી નિયમિતપણે કરવામાં આવે તેવી અપેક્ષા રાખવામાં આવે છે :

- જ્યારે કોઈ વ્યવહાર (ટ્રાન્ઝેક્શન) થાય, ત્યારે ડેટાબેઝમાં તે ઉમેરવામાં આવે.
- જ્યારે કોઈ પણ સુધારા ધ્યાનમાં આવે, ત્યારે ડેટાબેઝમાં તે એડિટ કરવામાં આવે.
- જો કંઈ પણ બિનજરૂરી માહિતી ડેટાબેઝમાં હોય, તો તે દૂર કરવામાં આવે.
- યોગ્ય રીતે ડેટાબેઝનું આયોજન કરવામાં આવે.
- વપરાશકર્તાની જરૂરિયાત પ્રમાણે ડેટાની પુનઃપ્રાપ્તિ કરવામાં આવે.
- ડેટાબેઝમાં રહેલા ડેટાની સુરક્ષા સચવાઈ રહે.
- ડેટાબેઝના બેક-અપની જાળવણી કરવામાં આવે.
- એક જ સમયે અનેક વપરાશકર્તાને ડેટાબેઝ વાપરવાની મંજૂરી આપે.
- બહુવિધ ભાષાઓમાં ડેટાનો સંગ્રહ કરવાની મંજૂરી આપે.
- બહુવિધ સ્થળોએ ડેટાનો સંગ્રહ કરવાની મંજૂરી આપે.

ડેટાબેઝના નિયંત્રણ સંબંધિત બધી ક્રિયાઓ ડેટાબેઝ મેનેજમેન્ટ તરીકે ઓળખાય છે. સ્વાભાવિક છે કે આ પ્રકારના ખાસ કામ માટે વિશિષ્ટ પ્રકારના સોફ્ટવેરની જરૂર પડે. આવાં સોફ્ટવેરને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ્સ (DBMS) તરીકે ઓળખવામાં

આવે છે. DBMSને ડેટાનો કાર્યક્ષમ સંગ્રહ, જાળવણી અને પુનઃપ્રાપ્તિ માટે બનાવેલી સિસ્ટમ તરીકે વ્યાખ્યાયિત કરી શકાય. ટૂંકમાં, DBMS એવું સોફ્ટવેર છે, જે ડેટાબેઝને વ્યવહારુ અને વધારે ઉપયોગી બનાવે છે. આજની દુનિયામાં ડેટાબેઝ પોતે નિરર્થક છે જો તેમાંની માહિતી પાછી મેળવવા માટે તેની સાથે કોઈ DBMS સંકળાયેલ ન હોય. ઓરેકલ (Oracle), ઇન્ગ્રેસ (Ingress), ડીબી2 (DB2), એસક્યુએલ સર્વર (SQL Server), માયએસક્યુએલ (MYSQL), એમએસ એક્સેસ (MS Access), બેઝ (Base) વગેરે બજારમાં ઉપલબ્ધ કેટલાંક લોકપ્રિય DBMS સોફ્ટવેરનાં નામો છે. આ પુસ્તકમાં આપણે બેઝ (Base) ડેટાબેઝ કે જે ઓપન ઓફિસ બેઝ તરીકે ઓળખાય છે, તેની ચર્ચા કરીશું. હવે પછી આપણે ફક્ત બેઝ શબ્દથી તેને ઓળખીશું. આપણે ડેટા, ઇન્ફર્મેશન, ડેટાબેઝ અને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ શબ્દાવલિથી પરિચિત છીએ, આથી ચાલો, આપણે હવે બેઝ (Base)થી પરિચિત થવા આગળ વધીએ કે જે ઓપન ઓફિસ સ્યૂટનો એક ભાગ ઓપનસોર્સ ડેટાબેઝ છે. બેઝ રિલેશનલ ડેટામોડેલ આધારિત છે.

બેઝનો પરિચય (Introduction to Base)

બેઝ રિલેશનલ મોડેલ આધારિત હોવાથી રિલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (RDBMS) છે. આપણે અગાઉ ચર્ચા કરી ગયા તે પ્રમાણે ઇન્ફર્મેશનને સરળતાથી અને કાર્યક્ષમ રીતે મેળવવા માટે અને તેનાં સંચાલનમાં તેની ગોઠવણી કયા પ્રકારે કરી છે, તેની ઊંડી અસર થઈ શકે છે. કદાચ સરળ પરંતુ વૈવિધ્યતાસભર ઇન્ફર્મેશનની ગોઠવણી એ ટેબલ (table)ના સ્વરૂપમાં તેનો સંગ્રહ છે. રિલેશનલ મોડેલ આ વિચાર ઉપર કેન્દ્રિત છે : દ્વિપરિમાણીય કોષ્ટકોમાં ડેટાની ગોઠવણી કે જેને “રિલેશન્સ” (relations) કહેવામાં આવે છે. ઈ.એફ. કોડ (E.F. Codd) દ્વારા ડિઝાઇન કરેલું રિલેશનલ મોડેલ ગણિતની સેટ થિયરી ઉપર આધારિત છે. જે ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (DBMS) રિલેશનલ મોડેલ ઉપર આધારિત હોય છે તે RDBMS તરીકે ઓળખાય છે. રિલેશનલ મોડેલ સૌથી સફળ ડેટામોડેલ છે. રિલેશનલ મોડેલની સફળતા પાછળનું એક કારણ તેની સરળતા છે. તેમાં ડેટાનો સંબંધ સમજવો અને તેના ઉપર કાર્ય કરવું ઘણું સરળ છે.

બેઝ એ સંબંધિત ડેટા-ઓબ્જેક્ટ્સનો સંગ્રહ છે, જે ટેબલ (Tables), ક્વેરીઝ (Queries), રિપોર્ટ્સ (Reports) અને એપ્લિકેશન મોડ્યુલ્સ (Application Modules)થી ઓળખાય છે. ચાલો, આપણે ઓબ્જેક્ટ કઈ રીતે બેઝમાં બનાવી શકાય તે બાબત ટૂંકમાં જાણીએ. આ બધા જ ઓબ્જેક્ટ્સની ઊંડાણપૂર્વક ચર્ચા આગળનાં પ્રકરણોમાં કરી છે.

ટેબલ (Table) : બેઝ એક RDBMS હોવાથી ડેટાબેઝમાં ડેટાનો સંગ્રહ કરવા માટે ટેબલ એક મૂળભૂત એકમ છે. ટેબલની રચના સ્તંભ અને હાર (કોલમ અને રો)ના સ્વરૂપે કરવામાં આવે છે. ટેબલમાં કઈ-કઈ ઇન્ફર્મેશનનો સંગ્રહ કરવો અને સ્તંભમાં શીર્ષકો (કોલમ ટાઇટલ્સ) શું રાખવા તેનો નિર્ણય કરવા માટે ડેટાબેઝ ડિઝાઇનરે સૌપ્રથમ ડેટાબેઝની એન્ટિટીઝ (entities) નક્કી કરવી પડશે (એટલે કે ડેટાબેઝમાં કઈ-કઈ માહિતીનો સમાવેશ કરવો છે). એન્ટિટી (entity) એ વાસ્તવિક દુનિયાનાં ઓબ્જેક્ટ (object) છે, જેના વિશેની માહિતી ડેટાબેઝમાં સંગ્રહ કરવાની છે. ઉદાહરણ તરીકે, કર્મચારીની હાજરી માટેની સિસ્ટમના ડેટાબેઝની ડિઝાઇન તૈયાર કરતાં સમયે આપણા મગજમાં સૌપ્રથમ એન્ટિટી આવે તે છે કર્મચારી. ડેટાબેઝમાં દરેક કર્મચારીની વિગતનો સંગ્રહ કરવો પડે. આ વિગત હારનાં સ્વરૂપમાં સંગ્રહ થાય છે. દરેક એન્ટિટી સામાન્ય રીતે કેટલાક એટ્રીબ્યુટ હોઈ શકે. એન્ટિટીનાં એટ્રીબ્યુટને સ્તંભના સ્વરૂપે રાખવામાં આવે છે. રિલેશનલ મોડેલમાં ડેટા જેવી રીતે રજૂ કરવામાં આવે છે અને ટેબલના વિવિધ ભાગોને કયાં પદોથી ઓળખવામાં આવે છે તે આકૃતિ 9.1માં દર્શાવ્યું છે.

		Fields							
		First Name	Last Name	Address	City	Birthdate	Joining Date	Contact Number	Designation
Record	-Anya	Gangele	15-Shub bungalows, SG Road	Ahmedabad	12-10-1978	15-1-2012	99988870	Relationship Manager	
	-Robina	Arora	A-4, Sukh Shanti Appartment	Surendranagar	15-6-1970	10-10-2001	98707651	Receptionist	
	-Akhil	Jain	106, Nirvan nagar	Surat	1-9-1980	1-1-2001	99678901	Sales Person	
	-Majid	Khan	Gala Street	Ahmedabad	4-6-1983	4-5-2007	81167453	Sales Person	

આકૃતિ 9.1 : કર્મચારી માટેનું ટેબલ

આકૃતિ 9.1માં જોઈ શકાય છે કે એન્ટિટીનાં એટ્રીબ્યુટને ફિલ્ડ (fields) તરીકે ઓળખવામાં આવે છે. આમ, કર્મચારી (Employee) ટેબલમાં First Name, Last Name, Address, City, Birthdate, Joining Date, Contact Number અને Designation ફિલ્ડ છે. આ બધાં ફિલ્ડનો સમૂહ એક રેકોર્ડ બનાવે છે. આ રીતે << ‘Any’, ‘Gangele’, ‘15-Shub bungalows, SG Road’, ‘Ahmedabad’, ‘12-10-1978’, ‘99988870’, ‘15-1-2012’, ‘Relationship Manager’>> એક રેકોર્ડ (record) છે.

ફોર્મ્સ (Forms) : આપણને ટેબલમાં નવા રેકોર્ડ ઉમેરવાની, પ્રવર્તમાન રેકોર્ડમાં ફેરફાર કરવાની અને ટેબલમાં ઉપલબ્ધ રેકોર્ડ જોવાની જરૂર પડે. ફોર્મ એ એક ઓબ્જેક્ટ છે, જે ટેબલમાં ડેટા દાખલ (એન્ટર) કરવાની અને ટેબલમાં રહેલા ડેટામાં સુધારા-વધારા (એડિટિંગ) અથવા કાઢી નાખવા (ડિલીટ કરવા)ની સગવડ આપે છે. તેમાં ફોર્મેટ, સ્ટાઈલ અને રેડિયો બટન, લિસ્ટ બોક્સિસ જેવાં વિજેટ્સ (widgets)નો સમાવેશ થાય છે, જે ડેટા સાથે ટેબલની સરળતાથી અને વ્યવસ્થિત રીતે રચના કરવાની સગવડ પૂરી પાડે છે.

ક્વેરીઝ (Queries) : કોઈ ડેટાબેઝ પર્યાવરણમાં પૂછવામાં આવેલો પ્રશ્ન ક્વેરી (Query) તરીકે ઓળખાય છે. ઉદાહરણ તરીકે, વર્તમાન વર્ષના જુલાઈ માસમાં કેટલા એન્ડ્રોઈડ આધારિત મોબાઈલ ફોન વેચાયા ? આ ક્વેરી ડેટાબેઝનાં વિવિધ ટેબલમાંથી કેટલાક ડેટા પ્રદર્શિત કરશે.

રિપોર્ટ્સ (Reports) : બેઝમાં ક્વેરીનું આઉટપુટ હાર અને સ્તંભના સ્વરૂપમાં પ્રદર્શિત થાય છે. આપણે અગાઉ ચર્ચા કરી તે પ્રમાણે જરૂરી ઈન્ફર્મેશન મેળવવા માટે ઉપલબ્ધ ડેટાને પ્રોસેસ કરવામાં આવે છે અને ઈન્ફર્મેશન સ્વરૂપે આઉટપુટ પેદા કરવામાં આવે છે. દેખીતી રીતે, ઉપયોગકર્તા વધુ ઔપચારિક અને આકર્ષક આઉટપુટની અપેક્ષા રાખે છે. બેઝ અસરકારક વ્યાવસાયિક રિપોર્ટ (અહેવાલ) તૈયાર કરવાની સવતલ ધરાવે છે. વપરાશકર્તાની જરૂરિયાત મુજબ એક સંગૃહિત અને વાચનીય સ્વરૂપમાં કરેલી ઈન્ફર્મેશનની રજૂઆત રિપોર્ટ તરીકે ઓળખાય છે. બેઝમાં અનેક વિવિધ જટિલ રિપોર્ટ્સ તૈયાર કરી શકાય છે, જેનો ઉપયોગ સંસ્થાઓનાં મેનેજમેન્ટને નિર્ણય લેવામાં મદદ કરી શકે.

વિનિયોગ માટે નમૂનાનો ડેટાબેઝ તૈયાર કરવો (Creating a Sample Database for an Application)

આપણે બેઝ શીખવાની સમગ્ર સફર દરમિયાન એક કાલ્પનિક ‘મોડર્ન ઇલેક્ટ્રોનિક સ્ટોર’ માટે એક નાના વિનિયોગની રચના કરવા તરફ આગળ વધીશું. આ સ્ટોર મોબાઈલ્સ, કેમેરા, હેન્ડીકેમ્સ, લેપટોપ્સ, નોટબુક્સ, આઈપેડ્સ, સ્માર્ટફોન્સ અને તેને સંબંધિત એસેસરીઝ જેવાં ઉપકરણો વેચે છે. આ ઉત્પાદનો વિવિધ સપ્લાયર્સ પાસેથી મેળવવામાં આવે છે. સ્ટોરને ભૌતિક તેમજ વેબશોપ છે. આથી, ગ્રાહકો જાતે સ્ટોરમાં આવીને વસ્તુઓ ખરીદી શકે છે અથવા ઓનલાઈન ઓર્ડર નોંધાવી શકે છે. સંભવતિ ગ્રાહકમાંથી ગ્રાહકમાં ફેરવવા માટે વેચાણ કરનાર વ્યક્તિઓને ખાસ પ્રોત્સાહનો ચૂકવવામાં આવે છે. આપણે જે વિનિયોગ (એપ્લિકેશન) બનાવવાનો ધ્યેય રાખીએ છીએ, તેને ચાલો, સ્ટોર મેનેજમેન્ટ સિસ્ટમ નામ આપીએ. સ્ટોર મેનેજમેન્ટ સિસ્ટમ માટે ચાલો, એક નમૂનાનો ડેટાબેઝ તૈયાર કરીએ. આ પછીનાં પ્રકરણોમાં આપણે આ સિસ્ટમ માટે જરૂરી ફોર્મ્સ અને રિપોર્ટ્સની ડિઝાઈન બનાવીશું. નીચે જણાવેલાં પગલાંઓ આપણને જરૂરી ડેટાબેઝની ડિઝાઈનમાં મદદ કરશે.

પગલું 1 :

સૌપ્રથમ એ નક્કી કરો કે કયો ડેટા સંગૃહિત કરવો છે? આપણા ઉદાહરણમાં આપણે પ્રોડક્ટ (માલસમાન), સપ્લાયર્સ (માલ મોકલનાર), એમ્પ્લોયીઝ (કર્મચારીઓ) અને કસ્ટમર્સ (ગ્રાહકો) સંબંધિત માહિતીનો સંગ્રહ કરીશું. આ તમામ એન્ટિટીઝ (Entities) છે. આપણે અગાઉ ચર્ચા કર્યા પ્રમાણે એન્ટિટી કોઈ વ્યક્તિ, વસ્તુ, જગ્યા અથવા ઓબ્જેક્ટ હોઈ શકે, જેના વિશેનો ડેટા આપણે ડેટાબેઝમાં સંગ્રહ કરીએ છીએ. દરેક એન્ટિટી માટે અલગ-અલગ ટેબલની ડિઝાઈન બનાવવી જોઈએ. આથી આપણે ચાર ટેબલ બનાવીશું, જેનાં નામ Product, Supplier, Employee અને Customer રાખીશું.

પગલું 2 :

હવે દરેક એન્ટિટી માટે કયા એટ્રીબ્યુટ (attributes)નો સંગ્રહ કરીશું તે નક્કી કરો. દા.ત., Product, Supplier, Employee, Customer માટે કયા એટ્રીબ્યુટનો સંગ્રહ કરીશું ? આ બધાં એટ્રીબ્યુટ એ ટેબલનાં ફિલ્ડ બની જશે. એટ્રીબ્યુટ (ફિલ્ડ)ને એન્ટિટીની લાક્ષણિકતા તરીકે વ્યાખ્યાયિત કરી શકાય. આપણે કોષ્ટક 9.1, કોષ્ટક 9.2, કોષ્ટક 9.3 અને કોષ્ટક 9.4માં દર્શાવ્યા પ્રમાણે માલ મોકલનાર, કર્મચારીઓ, ગ્રાહકો અને માલસામાનની માહિતીનો સંગ્રહ કરીશું.

Supplier	
Attributes	Description
Supplier Name	Name of supplier, it could be name of company like Samsung India
Address Line 1	Address from where products are dispatched
Address Line 2	Street and/or area
City	City from where products are dispatched
Pincode	Pincode of area within a city
Contact Person Name	Person to be contacted
Email Id	E-mail address of contact person
Contact Number	Contact number of office

કોષ્ટક 9.1 : Supplier એન્ટિટીનાં એટ્રીબ્યુટ

Employee	
Attributes	Description
Firstname	Name of the Employee
Lastname	Surname of the Employee
Address Line 1	Address of the Employee
Address Line 2	Street and/or area
City	City
Pincode	Pincode of area of above city
Email Id	E-mail address of Employee
Mobile Number	Mobile number of Employee
Birthdate	Birthdate of Employee
Joiningdate	Joining date of Employee
Designation	Current designation of an employee

કોષ્ટક 9.2 : Employee એન્ટિટીનાં એટ્રીબ્યુટ

Customer	
Attributes	Description
Firstname	Name of the Employee
Customer First Name	Name of customer
Customer Last Name	Surname of customer
Address Line 1	Address of customer
Address Line 2	Street or area
City	City
Pincode	Pincode of area of above city
Email Id	E-mail address of customer
Mobile number	Mobile number of customer
CardHolder	Whether customer is member of Modern Electronic Store

કોષ્ટક 9.3 : Customer એન્ટિટીનાં એટ્રીબ્યુટ

Product	
Attributes	Description
Product Category	Category of Product like ipad, mobile, smartphone etc.
Model Name	Name of the Model like Samsung Galaxy III
Supplier Name	Name of the Supplier
Reorder Level	Stock level that triggers purchase order
Selling Price	Current selling price of the product
Quantity on Hand	Stock available
Product Image	Image of Product
Camera Pixel Size	Mega pixels of Camera
Processor	Processor name
OS Support	Name of Operating Systems supported by model
Touch Screen	Touch Screen Feature available or not
Wi-fi Enabled	Whether the phone is wi-fi enabled
Memory Slot	Capacity of Initial slot of memory
FM Support	Whether FM support is available or not
Display Size	Display size in inches
HDD Capacity	Hard disk capacity
Weight	Weight of the item

કોષ્ટક 9.4 : Product એન્ટિટીનાં એટ્રીબ્યુટ

એટ્રીબ્યુટનાં લૉજિકલ અને ફિઝિકલ નામો (Logical Names and Physical Names of Attributes)

જ્યારે આપણે એટ્રીબ્યુટનાં નામો વિશે વિચાર કરતાં હોઈએ ત્યારે એટ્રીબ્યુટના લૉજિકલ નામ અને ડેટાબેઝ એપ્લિકેશનમાં વાપરેલાં ફિઝિકલ નામ વચ્ચેનો તફાવત સમજવો જરૂરી બને છે. ઉદાહરણ તરીકે “First Name” એ એક ફિલ્ડનું નામ છે, જે વ્યક્તિના નામનો સંગ્રહ કરે છે, જેમકે “Sonal” અથવા “Minal”. અહીં “First Name” એ એક ડેટાએન્ટિટી છે, જેના સાથે આપણે કામ કરીએ છીએ. જોકે “First Name” વૈકલ્પિક રીતે “FIRST NAME” અથવા “firstName” અથવા “first_name” કે પછી “FName” લખાય તોપણ કોઈ વાંધો નથી કે જ્યાં સુધી આપણે વિચારીને એ સમજી શકીએ કે તે વ્યક્તિના નામનો નિર્દેશ કરે છે.

જ્યારે આપણે કાલ્પનિક કે તાર્કિક સંબંધો વિશે વિચારીએ છીએ, ત્યારે સ્પષ્ટ અને વર્ણનાત્મક નામની આપણને જરૂર પડે છે. ડેટામોડેલના ડિઝાઇન સમયે લૉજિકલ નામ વપરાય છે. પરંતુ કમનસીબે ડેટાબેઝ સોફ્ટવેર જે નામ વાપરવા પરવાનગી આપે છે, તેમાં લૉજિકલ નામ સરળતાથી રૂપાંતર થતાં નથી. ઉદાહરણ તરીકે, બેઝમાં ફિલ્ડને “First Name” આપી શકાય પણ અન્ય ડેટાબેઝ એપ્લિકેશન્સ તેનો અસ્વીકાર કરશે અને તેને બદલે “first_name” વાપરવા સૂચવશે. આ નિયંત્રણ કોઈ ચોક્કસ ડેટાબેઝ સોફ્ટવેરની ડિઝાઇન એ રીતે કરવામાં આવી છે, તેને કારણે છે. ટેબલના આંતરિક માળખામાં ખરેખર જે નામ વાપરવામાં આવેલું છે, તેને “ફિઝિકલ” (physical) નામ કહેવામાં આવે છે, જે લૉજિકલ નામની ચર્ચા ઉપર કરી તેનાથી તદ્દન ઊલટું છે. આ ફિઝિકલ નામ હંમેશાં આપણે જે સોફ્ટવેર વાપરીએ છીએ, તેની પ્રણાલિકા પ્રમાણે જ હોવાં જરૂરી છે.

નીચે જણાવેલી પ્રણાલિકાને અનુસરીને એટ્રીબ્યુટનાં નામ રાખવાથી ડેટાબેઝની ડિઝાઇનને અન્ય ડેટાબેઝ સોફ્ટવેરમાં વાપરવામાં સવલત રહેશે (આપણા કિસ્સામાં બેઝ સોફ્ટવેર).

1. બધાં એટ્રીબ્યુટનાં નામ અક્ષરથી શરૂ કરો.
2. અનુગામી અક્ષરો માટે અક્ષરો, અંકો અથવા અન્ડરસ્કોર ચિહ્ન વાપરો.
3. બે શબ્દો વચ્ચે જગ્યા (સ્પેસ)નો ઉપયોગ ન કરો, પણ તેના બદલે અન્ડરસ્કોર ચિહ્નથી તે શબ્દોને છૂટા કરો.
4. અન્ડરસ્કોર સિવાય અન્ય કોઈ વિશિષ્ટ કેરેક્ટરનો ઉપયોગ ન કરો.
5. એટ્રીબ્યુટનાં નામ ટૂંકાં રાખવા માટે જો જરૂર જણાય તો શબ્દનું સંક્ષિપ્ત રૂપ વાપરો.

આપણે નક્કી કરેલી સેમ્પલ એપ્લિકેશન માટે ડેટાબેઝ ડિઝાઇનમાં કેમલ બેક (camel back) સંકેતોનો ઉપયોગ કરીશું. આનો અર્થ એ થાય કે ફિલ્ડનેઈમનો દરેક શબ્દ કેપિટલ અક્ષરથી શરૂ થશે. ઉદાહરણ તરીકે, FirstName, LastName, AddressLine1 વગેરે. આ પ્રકરણમાં હવે પછી આપેલી આકૃતિ 9.6 અને આકૃતિ 9.7માં મોડર્ન ઇલેક્ટ્રોનિક સ્ટોર માટે આપણા નમૂનાના ડેટાબેઝમાં વપરાયેલાં ફિલ્ડનાં નામો દર્શાવ્યા છે.

પગલું 3 :

હવે, તમે એ નક્કી કરો કે દરેક એટ્રીબ્યુટમાં કયા પ્રકારના ડેટા તમે દાખલ કરવા માંગો છો. તેના આધારે ટેબલની ડિઝાઇન કરતાં સમયે દરેક ફિલ્ડના ડેટાટાઈપ (Data type) નક્કી કરવા પડે છે. દરેક ફિલ્ડની ડેટાટાઈપ વ્યાખ્યાયીત કરવી ફરજિયાત છે. ડેટાટાઈપ એ ફિલ્ડમાં ડેટા સંગ્રહ કરવાનું માળખું દર્શાવવાનો એક માર્ગ છે. કમ્પ્યૂટર્સમાં ડેટાનો સંગ્રહ કરવાની વિવિધ રીતો છે. સામાન્ય રીતે, ચોક્કસાઈ માટે મેમરી કે ઝડપમાં ફેરબદલ કરવામાં આવે છે. જે ગણતરીઓમાં વધારે ચોક્કસાઈની જરૂર હોય, તેની પ્રક્રિયા ધીમી હોઈ શકે અને વધારે મેમરીનો ઉપયોગ કરે છે. જ્યારે બેઝમાં ટેબલ બનાવવામાં આવે છે, ત્યારે આપણી સમક્ષ એક ડ્રોપડાઉન મેનુમાં વિવિધ વિકલ્પો રજૂ કરી આપણે કયા પ્રકારના એટ્રીબ્યુટનો સંગ્રહ કરવો છે, તેની પસંદગી આપે છે. અહીં આપણે જે પસંદગી કરીએ, તેની ડેટાબેઝની કામગીરી ઉપર અસર થશે. આપણે ટેબલ વિશે જે ચર્ચા કરી તેમાં આકૃતિ 9.2, 9.3, 9.4 અને 9.5માં દર્શાવ્યા પ્રમાણે નમૂનાના રેકોર્ડ ઉમેરો. બેઝમાં ઉપલબ્ધ ડેટાટાઈપની ચર્ચા આપણે પછી કરીશું. આપણા નમૂનાના ટેબલનાં એટ્રીબ્યુટ માટે કયા ડેટાટાઈપ નક્કી કરવા તે નિર્ણય તે પછી કરીશું.

Supplier Name	Address Line 1	Address Line 2	City	Pincode	Contact Person Name	Email Id	Contact Number
Samsung Corporation	27, Maker Chambers	Nariman point	Mumbai	400021	Mr.Mahesh Wadkar	purchase@samsung.com	66006600
Hewlett-Packard India Sales Pvt. Ltd	24, Salarpuria Areana	Adugod, Hosur Road	Bangalore	560030	Mr.Venkatesh Rao	placeorder@hp.com	26543123
Sony India	A-31, Mohan Co-operative Indisurial Estate	Mahura Road	Delhi	110001	Mr.Habib Ansari	purchase@sony.com	33344441
Nokia India	Salarpuria Hallmark	Outer Ring Road	Bangalore	560089	Ms.Rita Kulkarni	po@nokia.com	89674210

આકૃતિ 9.2 : Supplierના નમૂનાના રેકોર્ડ

Product Category	Model Name	Supplier Name	Reorder Level	Selling Price	Quantity on Hand	Product Image	Camera Pixel Size	Processor	OS Support	Touch Screen	Wi-fi Enabled	Memory Slot (in GB)	FM Support	Display Size (in inches)	HDD Capacity (in GB)	Weight (in gms)
Smartphone	Samsung Galaxy SIII	Samsung Corporation	20	42500	43	/images/s_gIII.jpg	8	1.4Ghz Quad Core Processor	Android 4.0	yes	yes	16	yes	4.8		133
Laptop	hp ENVY Sleekbook 6t-1000	Hewlett-Packard India Sales Pvt. Ltd	15	35000	56	/images/hp6t.jpg		Intel Core i5	Windows		yes	16		15.6	500	4.75
Laptop	hp ENVY Sleekbook 6t-1000-L	Hewlett-Packard India Sales Pvt. Ltd	15	30000	56	/images/hp6t.jpg		Intel Core i5	Linux		yes	16		15.6	500	4.75
Mobile	Nokia Lumia 710	Nokia India	30	15499	32	/images/Lumina710.jpg	5	Qualcomm MSM8255 (WCDMA)	Windows Phone 7.5	yes	yes	8	yes	3.7		125.5
Camera	DSC-S3000/S	Sony India	35	4490	38	/images/dscs3000.tiff	10.1							6.14		11.9

આકૃતિ 9.3 : Productના નમૂનાના રેકોર્ડ

First Name	Last Name	Address Line 1	Address Line 2	City	Pincode	Email Id	Mobile Number	Birthdate	Joiningdate	Designation
Anya	Gangele	15-Shub Bungalows	SG Road	Ahmedabad	380067	anya.gangele@gmail.com	99988870	12/12/1981	1/6/2011	Relationship Manager
Robina	Arora	A-4, Sukh Shanti Apartment	Thaltej	Surendranagar	363002	robina.arora@yahoo.com	98707651	6/7/1984	2/7/2006	Receptionist
Akhil	Jain	106, Nirvan Nagar	Army Road	Surat	395001	akhil.jain@yahoo.in	99678901	8/10/1983	1/1/2000	Sales Person
Majid	Khan	B-12, Nishant Appts.	Satellite	Ahmedabad	380058	majid@rediffmail.com	81167453	7/11/1985	1/5/2001	Sales Person

આકૃતિ 9.4 : Employeeના નમૂનાના રેકોર્ડ

Customer Fname	Customer Lname	Address Line1	Address Line 2	City	Pincode	Email Id	Mobile Number	Card Holder
Puja	Sethia	17,Akash	Jodhpur Char Rasta	Ahmedabad	380056	puja.sethia@yahoo.com	9998886662	Yes
Sakina	Nagingar	T-37, Abhimanyu Nagar	Curry Road	Mehsana	384001	sakina82@gmail.com	8111203450	No
Joseph	Thomas	BG-1, Sunstar Complex	Sola Road	Patan	362268	joseph@yahoo.com	9980054321	Yes
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmedabad	380089	prerna@gmail.com	9004532189	No
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmedabad	380058	harshit@yahoo.com	9012354310	Yes

આકૃતિ 9.5 : Customerના નમૂનાના રેકોર્ડ

બેઝમાં ઉપલબ્ધ ડેટાટાઈપ (Data Types Available in Base)

બેઝમાં ઉપલબ્ધ ડેટાટાઈપને ત્રણ પ્રકારમાં વિભાજિત કરી શકાય છે, જેમકે આલ્ફાન્યુમરિક ટાઈપ, કેલેન્ડર ટાઈપ અને બાયનરી ટાઈપ.

આલ્ફાન્યુમરિક ટાઈપ (Alphanumeric Type)

જ્યારે આપણે અક્ષર, અંક અને વિશિષ્ટ અક્ષરો કે ચિહ્નો ધરાવતો ડેટા એન્ટર કરવા ઇચ્છતા હોઈએ ત્યારે કોષ્ટક 9.5માં દર્શાવ્યા પ્રમાણેના ડેટાટાઈપમાંથી કોઈ એક ટાઈપ પસંદ કરી શકીએ. ઉદાહરણ તરીકે, સરનામામાં મૂળાક્ષરો, વિશિષ્ટ

અક્ષરો અને અંક હોઈ શકે, આથી આલ્ફાન્યુમરિક ટાઈપમાંથી કોઈ એક ટાઈપ નક્કી કરવી ડેટાટાઈપ માટે યોગ્ય રહેશે. એડ્રેસ ફિલ્ડ માટે આપણે Text (Varchar) ડેટાટાઈપ પસંદ કરી શકીએ. ઉપયોગકર્તા એ કેટલાં કેરેક્ટર એન્ટર કર્યા તેના પ્રમાણે VarChar ડેટાટાઈપમાં બાઈટ્સની સંખ્યા ફાળવવામાં આવશે. ઉદાહરણ તરીકે, હર્ષિતના સરનામામાં જો 100 કેરેક્ટર હશે તો, 100 બાઈટ અનામત રાખવામાં આવશે અને તેમાં જો 50 કેરેક્ટર હશે તો ફક્ત 50 બાઈટ અનામત રાખશે.

Name	Data type	Max Length	Description
Memo	Long VarChar	2 GB	ઉપયોગકર્તા દ્વારા નિર્દેશ કરવામાં આવેલ હોય તે મહત્તમ લંબાઈ કે સંખ્યામાં માહિતીનો સંગ્રહ કરે છે. તે કોઈપણ UTF 8 અક્ષરો સ્વીકારે છે.
Text (fix)	Char	2GB	ઉપયોગકર્તા દ્વારા નિર્દેશ કરવામાં આવેલ હોય તેટલી લંબાઈના લખાણનો સંગ્રહ કરે છે. નાની અક્ષરમાળા સાથે ખાલી જગા (space) ઉમેરે છે. કોઈપણ UTF 8 અક્ષરો સ્વીકારે છે.
Text	Var Char	2GB	આપેલ લંબાઈ સુધીની વિગતોનો સંગ્રહ કરે છે. ખાલી જગ્યા ઉમેરવામાં આવતી નથી. (long var charની જેમ)
Text	Var Char Ignore Case	2GB	આપેલ લંબાઈ સુધીની વિગતોનો સંગ્રહ કરે છે. સરખામણી કેસ સેન્સીટીવ નથી. તમે ટાઈપ કરો તે મુજબ કેપીટલનો સંગ્રહ કરે છે.

કોષ્ટક 9.5 : આલ્ફાન્યુમરિક ટાઈપનાં એટ્રીબ્યુટ્સ : આલ્ફાન્યુમરિક કેરેક્ટરનો સંગ્રહ કરવા માટે વપરાય

‘CHARACTER’ ડેટાટાઈપનાં લક્ષણો અલગ હોય છે. આપણે જ્યારે ફિલ્ડમાં ચોક્કસ સંખ્યાના કેરેક્ટર દાખલ કરવા બાબત જાણતા હોઈએ ત્યારે તેનો ઉપયોગ થાય છે. ઉદાહરણ તરીકે, લાઈસન્સનંબર અને પાસપોર્ટનંબર માટે આપણે તેનો ઉપયોગ કરી શકીએ છીએ, કારણકે તેમાં ચોક્કસ સંખ્યામાં કેરેક્ટર હોય છે. જોકે character ડેટાટાઈપમાં કેરેક્ટરની સંખ્યા વાપરી શકાય તેની ઉપર મર્યાદા છે.

જ્યારે આપણે કોઈ વર્ણનાત્મક ડેટાનો સંગ્રહ કરવાનો હોય કે જેમાં 255 કરતાં વધારે અક્ષરો હોય, તો મેમો (Memo) ડેટાટાઈપ આપણને વધારે ઉપયોગી બને છે. આ ડેટાટાઈપમાં 64000 અક્ષરો સુધી કોઈ પણ શાબ્દિક માહિતીનો સંગ્રહ કરી શકાય છે. જ્યારે કોઈ ફિલ્ડ સર્યમાં વપરાય અને એ અનિશ્ચિત હોય કે ઉપયોગકર્તા જે માહિતી શોધવાની હોય તે લોઅરકેસમાં કે અપરકેસમાં દાખલ કરશે, તો Text (Var Char) Ignore casenનો ઉપયોગ વધારે કાર્યક્ષમ રહે છે. ઉદાહરણ તરીકે, વ્યક્તિનાં પ્રથમ નામ કે છેલ્લા નામનો સંગ્રહ Text (Var Char) Ignore caseમાં કરી શકાય.

અહીં નોંધ કરો કે આલ્ફાન્યુમરિક કેરેક્ટરનો સંગ્રહ કરવા માટે કમ્પ્યૂટર્સમાં અલગ-અલગ રીતો હોય છે. ઉદાહરણ તરીકે, આસ્કી (ASCII) કોડમાં એક કેરેક્ટરનો સંગ્રહ કરવા માટે ફક્ત એક બાયટની જરૂર પડે છે. પણ કમનસીબે, તમે વાપરી શકો તેવાં શક્ય કેરેક્ટરની સંખ્યા તે ફક્ત 256 સુધી જ મર્યાદિત કરે છે. જોકે તે ઘણા ખરા વિનિયોગ માટે આ સંખ્યા પૂરતા પ્રમાણમાં હોવા છતાં જો તમે ગુજરાતી અથવા હિન્દી ભાષાનાં કેરેક્ટર જેવા એક્સ્ટેન્ડેડ કેરેક્ટર સેટનો વપરાશ કરવા માંગો, તો તે ઓછાં પડે છે. જોકે વધારે સંખ્યામાં કેરેક્ટર વાપરી શકાય તેવાં ધારાધોરણો બનાવેલાં છે, જેમકે યુનિકોડ (Unicode) એક કેરેક્ટરના સંગ્રહ માટે વધારે બાઈટ વાપરે છે.

UTF-8 નો ઉપયોગ કરીને બેઝ આલ્ફાન્યુમરિક કેરેક્ટરનો સંગ્રહ કરે છે, જે ASCII અને Unicode બંને સાથે સુસંગત છે. આંતરિક ગણતરી પ્રમાણે બેઝ દરેક કેરેક્ટર માટે એક અથવા વધારે બાઈટ વાપરે છે. જ્યારે બેઝ તમને કોઈ ચોક્કસ ફિલ્ડની લેન્થ (length) પૂછે છે, ઉદાહરણ તરીકે કર્મચારીની અટક, તો તે કેટલા બાઈટની ફાળવણી કરવી તે પૂછતું નથી પણ તમે કેટલાં કેરેક્ટરનો સંગ્રહ કરવા ઇચ્છો છો તે પૂછે છે. ખરેખર કેટલા બાઈટનો ઉપયોગ થશે તે સોફ્ટવેર નક્કી કરે છે.

ન્યુમરિક ટાઈપ (Numeric Type)

આંકડાકીય માહિતીનો સંગ્રહ કરવા માટે આ ડેટાટાઈપનો ઉપયોગ થાય છે, જેમકે વિદ્યાર્થીના ગુણ, કર્મચારીનો પગાર, હાલમાં ઉપલબ્ધ માલનો જથ્થો કે કોઈ વ્યક્તિનાં બેંકના ખાતામાં સિલક.

આપણે અગાઉ ચર્ચા કરી ગયા છીએ કે આલ્ફાન્યુમરિક ડેટાટાઈપમાં કેટલા બાઈટનો ઉપયોગ થયો તે હકીકતમાં સોફ્ટવેર નક્કી કરે છે. પણ તમે જ્યારે સંખ્યાની કિંમતનો સંગ્રહ કરો છો, ત્યારે તે પ્રમાણે બનતું નથી. સંખ્યાનો સંગ્રહ અલગ-અલગ રીતે કરવાથી ઓછા અથવા વધારે બાઈટની જરૂર પડે છે. બાઈટની સંખ્યાનો ઉપયોગ અને સંખ્યાચિહ્ન સાથે કે ચિહ્ન વગર છે, તેના આધારે ન્યુમરિક ડેટાટાઈપના એટ્રીબ્યુટ નક્કી કરવામાં આવે છે. આ બંને મુદ્દાઓને ધ્યાનમાં રાખીને તેના આધારે તેમાં સમાવી શકાય તેવી શક્ય કિંમતની સીમા (રેન્જ) નક્કી કરી શકાય છે. બેઝ અનેક પ્રકારના ન્યુમરિક ડેટાટાઈપની સગવડ પૂરી પાડે છે, જેમાં ચિહ્ન સાથે અને ચિહ્ન વગરની સંખ્યા અલગ-અલગ સંખ્યામાં બાઈટ વાપરે છે.

સંખ્યાના સંગ્રહ માટે સૌથી ઓછી મેમરીનો વપરાશ કરવાની રીતે આપણી પાસે બુલિયનનંબર (Boolean number) છે. બુલિયનનંબર હકીકતમાં ફક્ત એક જ બીટ વાપરે છે અને આપણે તેનો ઉપયોગ હા/ના પ્રકારનાં ડેટાના સંગ્રહ માટે વાપરીએ છીએ, જેમકે ‘મોબાઈલ વાઈ-ફાઈ શક્તિમાન (enable) છે કે નહીં?’ પ્રકારના પ્રશ્નના જવાબ માટે. બીજી બાજુએ આપણા પાસે ફ્લોટિંગ પોઈન્ટ નંબર્સ (floating point numbers) જેવાં એટ્રીબ્યુટ પણ છે, જે આપણને દશાંશચિહ્ન સાથેની સંખ્યા સંગ્રહ કરવાની પરવાનગી આપે છે, જેમકે 1.715249 સંખ્યા. આ પ્રકાર સૌથી વધારે મેમરી વાપરે છે પણ વધારે ચોક્કસાઈ સાથે ભાગાકાર કરવા ફક્ત તે જ સમર્થ છે. કોષ્ટક 9.6માં વિવિધ ન્યુમરિક ડેટાટાઈપ સાથે તેમાં વપરાતી બિટ્સ/બાઈટની સંખ્યા અને તેની સીમા (રેન્જ) દર્શાવેલી છે.

Name	Data type	No. of Bits/Bytes	Signed	Range
Boolean	Yes/No	1 Bit	–	0 – 1
Tinyint	Tiny Integer	1 Byte	No	0 – 255
Smallint	Small Integer	2 Bytes	Yes	-32768 to 32768
Integer	Integer	4 Bytes	Yes	-2.14×10^9 to 2.14×10^9
Bigint	Big integer	8 Bytes	Yes	-2.3×10^{18} to 2.3×10^{18}
Numeric	Number	No limit	Yes	Unlimited
Decimal	Decimal	No limit	Yes	Unlimited
Real	Real	4 Bytes	Yes	$5 \times 10^{(-324)}$ to $1.79 \times 10^{(308)}$

કોષ્ટક 9.6 : ન્યુમરિક ટાઈપ એટ્રીબ્યુટ : આંકડાકીય કિંમતોનો સંગ્રહ કરવા માટે વપરાય છે.

કેલેન્ડર ટાઈપ (Calendar Type)

અન્ય એક મહત્વપૂર્ણ ડેટાટાઈપ એ કેલેન્ડર ટાઈપ છે. તે વર્ષ, મહિનો, દિવસ, કલાક, મિનિટ, સેકન્ડનો નાનો અંશ (fraction) જેવી કેલેન્ડરની માહિતીનો સંગ્રહ કરવા માટે વપરાય છે. Date આપણને સિસ્ટમમાં સંગ્રહ હોય છે તે રીતે વર્ષ, મહિનો અને દિવસની માહિતી એન્ટર કરવાની પરવાનગી આપે છે. એ જ રીતે Time એટ્રીબ્યુટ દિવસના સમયનો સંગ્રહ કરે છે : કલાક, મિનિટ અને સેકન્ડ. અંતમાં, કોઈ ઘટનાનો દિવસ અને સમય બંનેની માહિતીનું રેકોર્ડિંગ કેટલીક પ્રક્રિયાની માંગ હોઈ શકે છે. Timestamp એટ્રીબ્યુટની ડિઝાઇન એકસાથે બંધી માહિતીનો સમાવેશ કરવાની છે. આપણા ડેટાબેઝના ઉદાહરણમાં Joining date, Birthdate માટે Date ડેટાટાઈપ વાપરવી જોઈએ. કોષ્ટક 9.7માં કેલેન્ડર ડેટાટાઈપનું નામ અને વર્ણન આપેલાં છે.

Name	Description
Date	Stores month, day and year information
Time	Stores hour, minute and second information
Timestamp	Stores date and time information

કોષ્ટક 9.7 : કેલેન્ડર ટાઈપ એટ્રીબ્યુટ : તારીખ અને સમયનો સંગ્રહ કરવા માટે

બાયનરી ટાઈપ (Binary Type)

એવી કોઈ પણ માહિતી જે શૂન્ય અને એકની શબ્દમાળા (string of zeros and ones)માં હોય તેનો સંગ્રહ કરવા માટે બાયનરી ટાઈપ વપરાય છે. ડિજિટલ ચિત્રો અને ધ્વનિ આ ફોર્મેટ વાપરે છે. તે કમ્પ્યુટર વડે ઓળખાય છે, કારણકે માહિતીની શરૂઆતના શૂન્ય અને એક વડે તે કયા પ્રકારની ફાઈલ છે, તેનો નિર્દેશ કરે છે (તે JPEG ઈમેજ કે MP3 ફાઈલ છે વગેરે). જોકે તમે કયા પ્રકારની ફાઈલનો સંગ્રહ કર્યો છે, તેને ઓળખવાનો કોઈ પ્રયત્ન બેઝ નહીં કરે. અહીં એ કહેવું છે કે તે ફાઈલ MP3 છે કે TIFF ફાઈલ તેની દરકાર કર્યા વગર બેઝ તેનો ઉમળકાભેર સંગ્રહ કરશે. આના પરિણામરૂપ અર્થ એ થાય કે માહિતીનો સંગ્રહ કરવા માટે તમે બેઝ ડેટાબેઝનો ઉપયોગ કરી શકો છો. ઉદાહરણ તરીકે, વસ્તુ કે કર્મચારીના ફોટા કે ધ્વનિનો નાનો અંશ (snippets) અથવા અવાજસ્વરૂપમાં સંદેશા. કોષ્ટક 9.8માં બાયનરી ડેટાટાઈપની વિગત આપેલી છે.

Name	Data type	Max Length	Description
Image	Long Var Binary	2GB	Stores any array of bytes (images, sounds, etc).
Binary	Var Binary	2GB	Stores any array of bytes.
Binary (fix)	Binary	2GB	Stores any array of bytes.

કોષ્ટક 9.8 : બાયનરી ટાઈપ એટ્રીબ્યુટ : ઈમેજ અને ધ્વનિનો સંગ્રહ કરવા માટેનો ઉપયોગ

ચાલો, હવે આપણી જરૂરિયાત પ્રમાણે ઈનપુટ ડેટા માટે ડેટાટાઈપ નક્કી કરીએ. આકૃતિ 9.6 અને 9.7માં Supplier, Product, Employee અને Customer ટેબલનાં દરેક ફિલ્ડના ડેટાટાઈપ નક્કી કરેલા છે.

Supplier	
Field Name	Data type
SupplierName	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]
ContactPersonName	Text [VARCHAR_IGNORECASE]
EmailId	Text [VARCHAR]
ContactNumber	Integer

Employee	
Field Name	Data type
FirstName	Text [VARCHAR_IGNORECASE]
LastName	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]
EmailId	Text [VARCHAR]
MobileNumber	Text [VARCHAR]
BirthDate	Date
JoiningDate	Date
Designation	Text [VARCHAR_IGNORECASE]

આકૃતિ 9.6 : Supplier અને Employee ટેબલનાં દરેક ફિલ્ડના ડેટાટાઈપ નક્કી કર્યા

Product	
Field Name	Data type
ProductCategory	Text [VARCHAR_IGNORECASE]
ModelName	Text [VARCHAR_IGNORECASE]
SupplierName	Text [VARCHAR_IGNORECASE]
ReorderLevel	SmallInt
SellingPrice	Integer
QuantityonHand	SmallInt
ProductImage	Text [VARCHAR]
CameraPixelSize	Integer
Processor	Text [VARCHAR_IGNORECASE]
OSSupport	Text [VARCHAR_IGNORECASE]
TouchScreen	Boolean
Wi-fiEnabled	Boolean
MemorySlotInGB	SmallInt
FMSupport	Boolean
DisplaySizeInInches	SmallInt
HDDCapacity	SmallInt
WeightInGrams	SmallInt

Customer	
Field Name	Data type
CustomerFname	Text [VARCHAR_IGNORECASE]
CustomerLname	Text [VARCHAR_IGNORECASE]
AddressLine1	Text [VARCHAR_IGNORECASE]
AddressLine2	Text [VARCHAR_IGNORECASE]
City	Text [VARCHAR_IGNORECASE]
Pincode	Text [VARCHAR]
EmailId	Text [VARCHAR_IGNORECASE]
MobileNumber	Text [VARCHAR]
Card Holder	Boolean

આકૃતિ 9.7 : Product અને Customer ટેબલનાં દરેક ફિલ્ડના ડેટાટાઈપ નક્કી કર્યા

નોંધ : જો આપણે જાણતા હોઈએ કે ફિલ્ડમાં માત્ર સંખ્યા જ છે પણ તે ફિલ્ડની કિંમત ઉપર આપણને કોઈ પણ ગણતરી કે કમ ગોઠવણીની અપેક્ષા ન હોય, તો તેના માટે Text ડેટાટાઈપ વાપરવો વધારે સારો છે. આથી, અહીં Pincode અને MobileNumber ફિલ્ડ માટે Text ડેટાટાઈપ વાપરેલો છે.

અહીં તમે એવું પણ વિચારો કે આપણા નમૂનાના ડેટાબેઝમાં Product ટેબલનાં DisplaySizeInInches ફિલ્ડ માટે આપણે Text ડેટાટાઈપ પણ વાપરી શકીએ. બેઝ 1, 12, 457 વગેરે જેવા ડેટા ઉમેરવાની પરવાનગી આપશે, પણ જ્યારે આપણે DisplaySize ફિલ્ડ ઉપર રેકોર્ડનું સોર્ટિંગ કરીશું, જેની ચર્ચા હવે પછીના પ્રકરણમાં કરેલી છે, તો આપણે સંખ્યાના ઊલટા ક્રમમાં : 457, 12, 1 પરિણામ મેળવીશું. આનું કારણ એ છે કે બેઝ જ્યારે Text ફિલ્ડની સંખ્યાને સોર્ટ કરે છે, ત્યારે સંખ્યાને જમણી બાજુથી ડાબી બાજુ વાંચવાને બદલે તે ડાબી બાજુથી જમણી બાજુ વાંચે છે. આથી બેઝમાં સોર્ટ કરતાં સમયે તે સૌથી જમણી બાજુનો અંક (457ના કિસ્સામાં 7) વાંચે છે અને તે પ્રમાણે સોર્ટ કરે છે.

પગલું 4

હવે, તમે ડિઝાઇન કરેલાં ટેબલનું કી-ફિલ્ડ (Key Field) નક્કી કરો. ટેબલનું એ ફિલ્ડ જેની અજોડ અગત્યતા હોય તે કી-ફિલ્ડ તરીકે ઓળખાય છે.

કોઈ પણ કંપનીની ઉત્પાદિત દરેક વસ્તુને મોડેલ નામ આપવાનો હેતુ શું હોઈ શકે? આપણને તે એકબીજા ઉત્પાદનથી અલગ કરવામાં મદદ કરે છે. તમે મોડર્ન ઈલેક્ટ્રોનિક સ્ટોરની મુલાકાત લઈ સીધા સેમસંગ ગેલેક્સી III બતાવવાનું તમે કહી શકો. તમને દુકાનદારના ચહેરા ઉપર કોઈ પણ જાતની મૂંઝવણ જોવા નહીં મળે. જ્યારે બીજા બાજુએ Employee ટેબલમાં પ્રથમ નામ એકસરખાં હોય તેવા એક કરતાં વધુ કર્મચારી હોઈ શકે છે. તેઓ કેવી રીતે ઓળખી શકાય છે? આપણે ડેટાબેઝમાંથી “આન્યા”ની વિગત પરત કરવા માટેની ક્વેરી ડેટાબેઝને કરી શકીએ પણ ડેટાબેઝ “આન્યા” નામનાં એક કરતાં વધારે રેકોર્ડ પાછા આપી શકે છે. પરિણામ જોઈને આપણે ગૂંચવણમાં પડી જઈએ કે આમાંની કઈ આન્યાની વિગત આપણને જોઈતી હતી? આથી, દરેક ટેબલમાં એક ફિલ્ડ એવું હોવું જોઈએ કે જે ટેબલના દરેક રેકોર્ડને અનન્ય (અજોડ/યુનિક) રીતે ઓળખી શકે. ઉદાહરણ તરીકે, દરેક કર્મચારીને એક અજોડ કર્મચારીનંબર આપી શકાય. આથી, આને Employee ટેબલમાં એક નવું ફિલ્ડ ઉમેરવાની આપણને જરૂર પડે છે.

ડેટાબેઝની દૃષ્ટિએ, એક અથવા વધારે ફિલ્ડના જોડાણ વડે દરેક રેકોર્ડને અજોડ રીતે ઓળખ માટે સક્ષમ બનાવે તેને પ્રાઈમરી કી (Primary key) કહેવામાં આવે છે. તમે એમ પણ વિચારો કે ડ્રાઈવિંગ લાઈસન્સ કે પાસપોર્ટનંબર પ્રાઈમરી કી તરીકે કામ આપી શકે, પરંતુ નલવેલ્યુ (null value) સાથેના અનેક રેકોર્ડ્સ હોઈ શકે. હંમેશાં એ જરૂરી નથી કે કર્મચારીની જ્યારે ભરતી થાય, ત્યારે તેની પાસે ડ્રાઈવિંગ લાઈસન્સ અથવા પાસપોર્ટ હોય. આકૃતિ 9.3ના Product ટેબલના નમૂનાના રેકોર્ડ્સ ઉપર હવે ફરી નજર કરો. તમે ચોક્કસ નોંધ્યું હશે કે સ્માર્ટફોનની HDDની ક્ષમતામાં કોઈ કિંમત આપણે દાખલ કરી નથી. આ માટે બે કારણો હોઈ શકે : સ્માર્ટફોનની HDD ક્ષમતા આપણે જાણતા ન હોઈએ અથવા તે ચોક્કસ એટ્રીબ્યુટ કોઈ કિંમત ધરાવતું હોય કે ન ધરાવતું હોય. આ પ્રકારની કિંમતને નલવેલ્યુ કહેવામાં આવે છે. નલવેલ્યુ એટલે અજ્ઞાત (unknown) કે ગેરહાજર (ખૂટતી) કિંમત. પ્રાઈમરી કીની કિંમતો ડુબ્લિકેટ વેલ્યુ તેમજ નલવેલ્યુ ન હોઈ શકે.

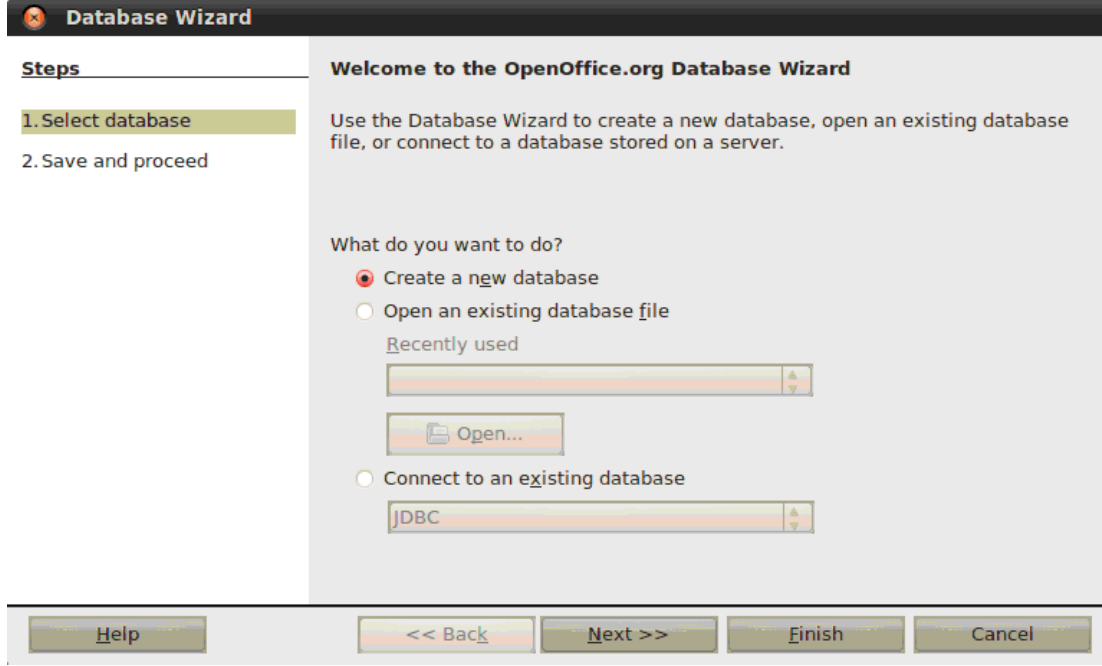
આકૃતિ 9.6 અને 9.7માં Supplier, Product, Employee અને Customer નામનાં ચાર ટેબલ આપેલાં છે. Supplier ટેબલમાં જોઈ શકાય છે કે તેમાં એવું કોઈ ફિલ્ડ નથી કે જે અજોડ કિંમત ધરાવતું હોય. આથી Supplier ટેબલમાં આપણે એક વધારાનું ફિલ્ડ Scode ઉમેરવું પડશે, જેથી Supplierનો દરેક રેકોર્ડ અજોડ રીતે ઓળખી શકાય. આપણે Scodeનું માળખું નક્કી કરવાની પણ જરૂર છે. ચાલો, ધારો કે તે એક ફિલ્ડ છે જેમાં ચાર કેરેક્ટર છે. પ્રથમ અક્ષર મૂળાક્ષર ‘S’ હોવો જોઈએ અને બાકીના ત્રણ અક્ષર સંખ્યાનો નિર્દેશ કરે છે. આથી મોડર્ન ઈલેક્ટ્રોનિક સ્ટોરના સપ્લાયર્સને S001, S002, S003 વગેરેથી ઓળખી શકાય. આ જ પ્રમાણે Product ટેબલમાં Pcode, Employee ટેબલમાં Ecode અને Customer ટેબલમાં Ccode એટ્રીબ્યુટ ઉમેરવું જોઈએ.

તમે Product ટેબલમાં જોઈ શકો છો કે “મોડેલ નેઈમ” જે છે તે Product ટેબલ માટે પ્રાઈમરી કી તરીકે કામ કરી શકે છે. જોકે એ પણ લક્ષમાં આવ્યું હશે કે મોડેલ નેઈમમાં અક્ષરો ચોક્કસ સંખ્યામાં નથી અને દરેક પ્રોડક્ટ માટે આશરે 15 કરતાં વધારે અક્ષરોનાં નામ છે. જો આપણે દરેક પ્રોડક્ટ માટે 8 કેરેક્ટર ફાળવીએ તો આપણે 9999999 પ્રોડક્ટને અજોડ રીતે દર્શાવી શકીએ (પહેલો અક્ષર P હોઈ શકે). ટેબલની પ્રાઈમરી કી ડેટાબેઝનાં બહુવિધ ટેબલ સાથે સંબંધ સ્થાપિત કરવા માટે પણ વપરાય છે. બે ટેબલ વચ્ચે સંબંધ (Relation) કઈ રીતે સ્થાપિત કરી શકાય, તેની ચર્ચા આપણે પછીના પ્રકરણમાં કરીશું.

દરેક ટેબલ માટે પ્રાઈમરી કી વ્યાખ્યાયિત કરવી જરૂરી છે. કોઈ પણ ટેબલની એક કરતાં વધારે પ્રાઈમરી કી ન હોઈ શકે. જોકે એક કરતાં વધારે ફિલ્ડનો સમન્વય (જોડાણ) પ્રાઈમરી કી તરીકે કામ આપી શકે છે. આ રીતે બનાવેલી પ્રાઈમરી કી કમ્પોઝિટ પ્રાઈમરી કી (Composite Primary Key) તરીકે ઓળખાય છે. આપણી એપ્લિકેશનમાં વપરાયેલાં અન્ય ટેબલની ડિઝાઇન બાબત ચર્ચા કરતાં પહેલાં ચાલો, આપણે બેઝનો ઉપયોગ કરીને અગાઉ ચર્ચા કરેલાં ટેબલની ડિઝાઇન બનાવીએ.

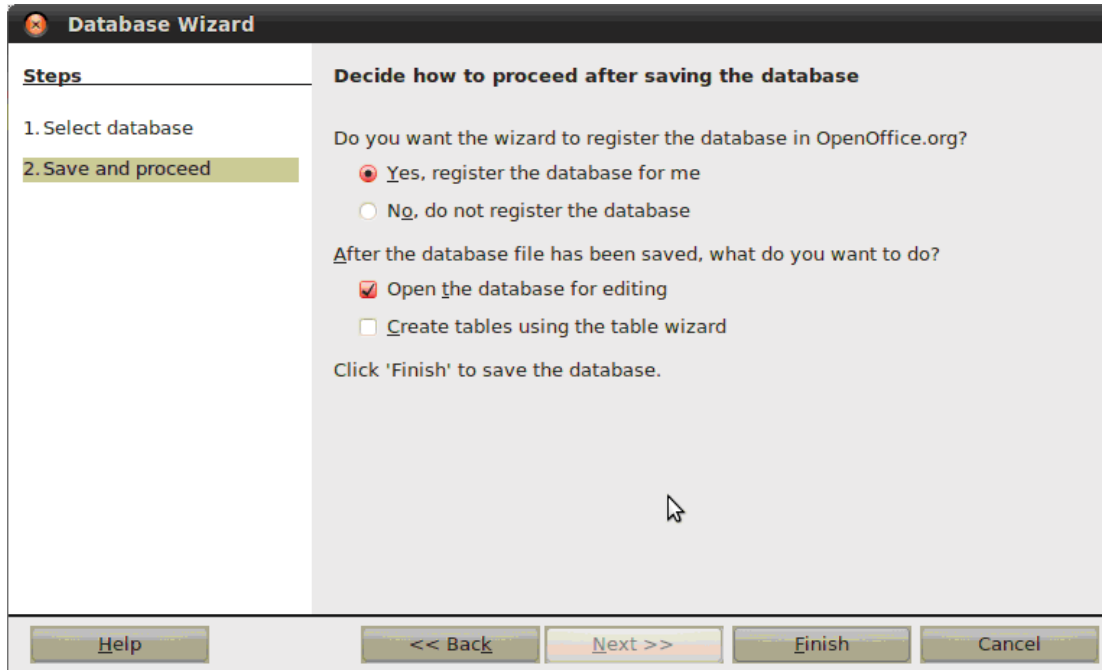
બેઝ ચાલુ કરવું (Opening Base)

ચાલો, આપણે સૌપ્રથમ બેઝ ચાલુ કરીએ. આ માટે કમાન્ડ પસંદ કરો : Applications → Office → OpenOffice.org 3.2 Base. જ્યારે તમે બેઝ સોફ્ટવેર ચાલુ કરો છો, ત્યારે આકૃતિ 9.8માં દર્શાવ્યા પ્રમાણે બેઝના ઓપનિંગ સ્ક્રીનમાં તમે કેટલાક વિકલ્પો જોઈ શકશો.



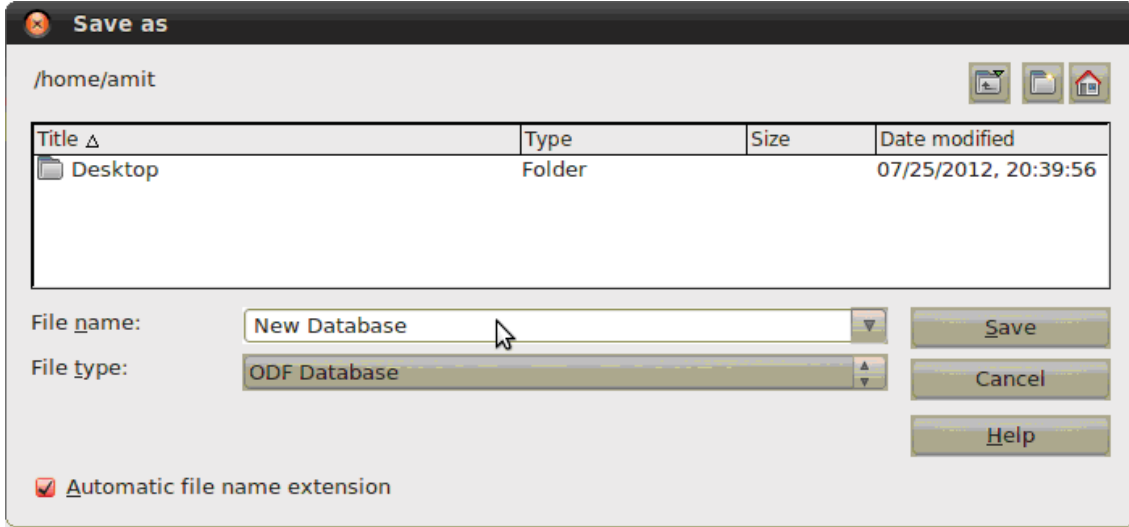
આકૃતિ 9.8 : બેઝનો ઓપનિંગ સ્ક્રીન

આપણે અગાઉ બનાવેલા ડેટાબેઝ ઉપર જો કામ કરવા ઈચ્છતા હોઈએ તો આપણે 'Open an existing database file' પસંદ કરી આવા ડેટાબેઝ ખોલી શકીએ છીએ. અહીં, આપણે હજી સુધી કોઈ પણ ડેટાબેઝ બનાવેલો નથી. આથી, આપણે 'Create a new database' વિકલ્પ પસંદ કરીશું. આ કમાન્ડ આપવાથી બેઝ આકૃતિ 9.9માં દર્શાવ્યા પ્રમાણેનો સ્ક્રીન પ્રદર્શિત કરશે.



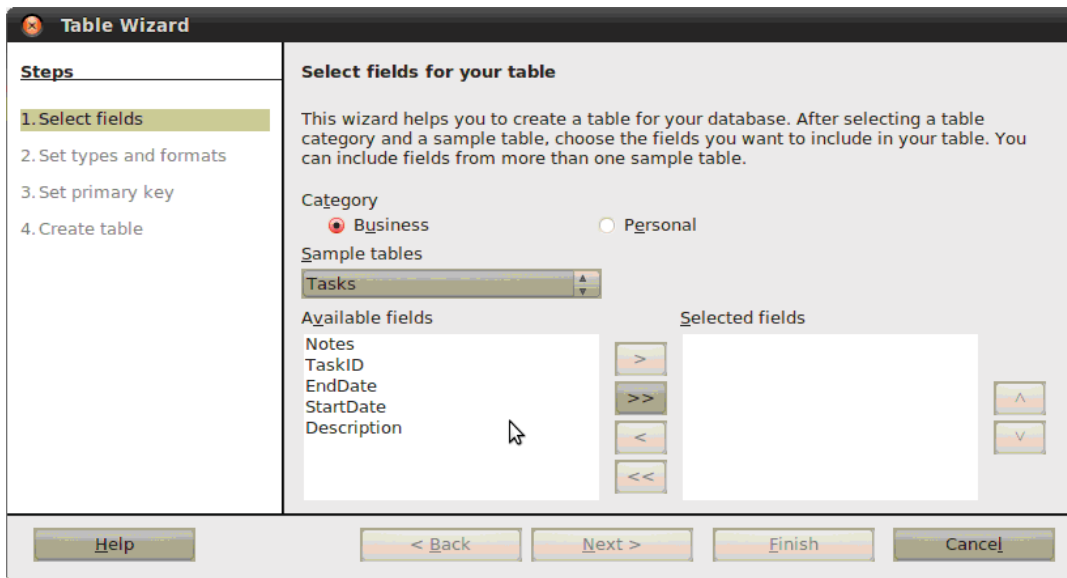
આકૃતિ 9.9 : ડેટાબેઝ ઓપન કરવો

આકૃતિ 9.9માં તમે નિહાળી શકશો કે OpenOffice.org વેબસાઈટ ઉપર તમે તમારો ડેટાબેઝ રજિસ્ટર કરાવવાનો વિકલ્પ મેળવો છો. જો આપણે ડેટાબેઝને સાર્વજનિક (public) બનાવવો હોય, તો આપણે આ વિકલ્પ પસંદ કરી શકીએ છીએ. સામાન્ય રીતે, આપણે ડેટાબેઝને પબ્લિક બનાવતા નથી, આથી બીજો વિકલ્પ “No, do not register the database” પસંદ કરો. બેઝ આપણને આગળ કામ કરવા માટે બે વિકલ્પો આપે છે. આપણે ડિઝાઈન વ્યૂ (Design view)નો ઉપયોગ કરીને અથવા વિઝાર્ડ (Wizard) વાપરીને ટેબલ બનાવી શકીએ છીએ. પૂર્વનિર્ધારિત (ડિફોલ્ટ) રીતે એડિટિંગ વિકલ્પ તપાસવા માટે ડેટાબેઝ ઓપન કરો. જો આપણે વિઝાર્ડનો ઉપયોગ કરવા ઈચ્છતા હોઈએ તો *Create tables using the table wizard* વિકલ્પના ચેકબોક્સને પસંદ કરો. ચાલો, આપણે વિઝાર્ડનો ઉપયોગ કરીને ટેબલ બનાવીએ. *Create tables using the table wizard* વિકલ્પ પસંદ કરો અને Finish બટન ઉપર ક્લિક કરો. આથી આકૃતિ 9.10માં દર્શાવ્યા પ્રમાણે Save As ડાયલોગબોક્સ ખૂલશે.



આકૃતિ 9.10 : ડેટાબેઝ ફાઈલને નામ આપવું

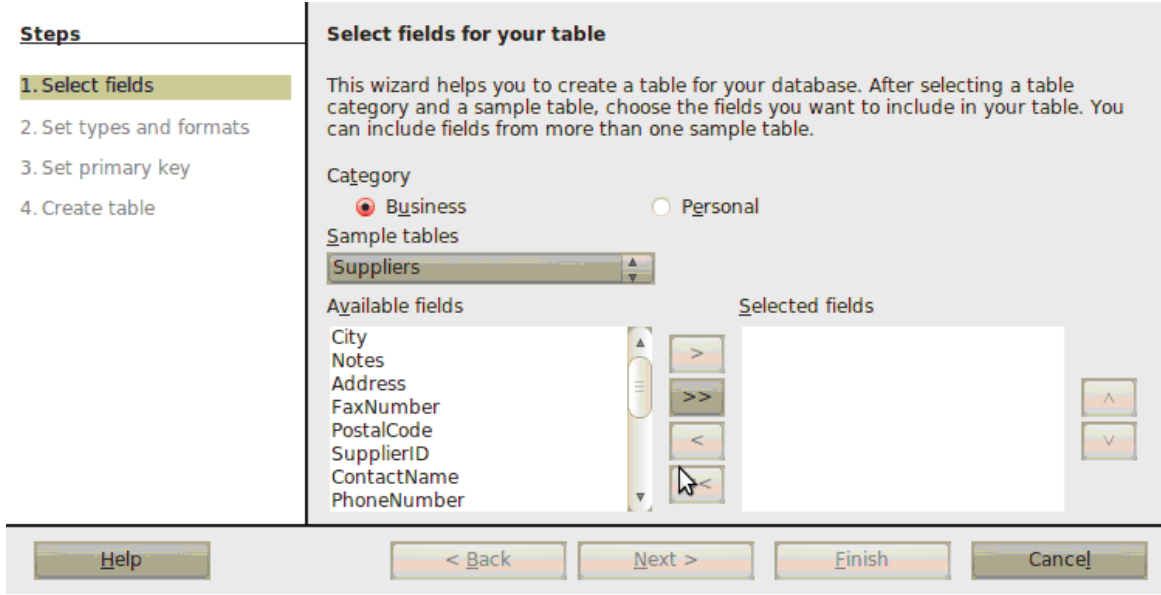
File name : લેબલ સાથેના ટેક્સ્ટબોક્સ (textbox) કે જ્યાં *New Database* લખેલું દેખાય છે, ત્યાં ModernElectronicStore ટાઈપ કરો અને પછી Save બટન ક્લિક કરો. ફાઈલને સેવ કરવા માટે ડ્રોપડાઉન એરો ઉપર ક્લિક કરીને યોગ્ય જગ્યા પસંદ કરો. ડેટાબેઝ ફાઈલને બેઝ જાતે .odb અનુલંબન આપશે. Save બટન ક્લિક કરો અને આકૃતિ 9.11માં દર્શાવ્યા પ્રમાણે ટેબલવિઝાર્ડ પ્રદર્શિત થશે.



આકૃતિ 9.11 : વિઝાર્ડ વાપરીને ટેબલની રચના

સૌપ્રથમ આપણે બેઝ સોફ્ટવેરમાં ઉપલબ્ધ ટેબલવિઝાર્ડનો ઉપયોગ કરીશું કે જેમાં અગાઉથી તૈયાર કરેલાં ટેબલ છે, જે બિઝનેસ (ધંધાકીય) અને પર્સનલ (અંગત) બે પ્રકારમાં વિભાજિત કરી શકાય.

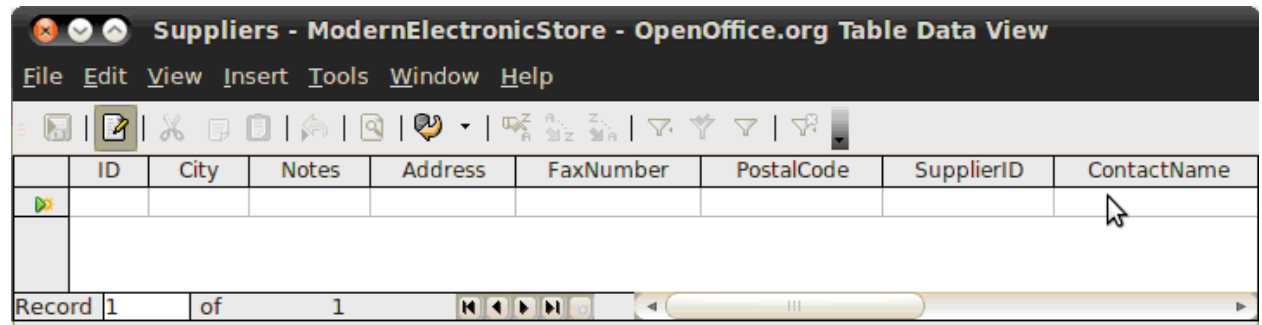
બિઝનેસ કેટેગરીમાં Tasks, Assets, Contacts, Expenses, Invoices વગેરે જેવાં નમૂનાનાં ટેબલનો સમાવેશ થાય છે. પર્સનલ કેટેગરીમાં Photographs, DVDCollection, Recipes, Plants જેવાં ટેબલનો સમાવેશ થાય છે. બિઝનેસ કેટેગરી પસંદ કરો અને *Sample tables*ની યાદીમાં ઉપલબ્ધ ડ્રોપડાઉન લિસ્ટ ઉપર ક્લિક કરો. તમે આખી યાદી તપાસો અને તમે જોઈ શકશો કે તમે જે ટેબલ બનાવવા માંગો છો તે તમામ ચાર ટેબલ તેમાં ઉપલબ્ધ છે. Supplier પસંદ કરો. આકૃતિ 9.12માં દર્શાવ્યા પ્રમાણે ફિલ્ડની યાદી તમે જોઈ શકશો.



આકૃતિ 9.12 : Suppliers ટેબલ બનાવવું

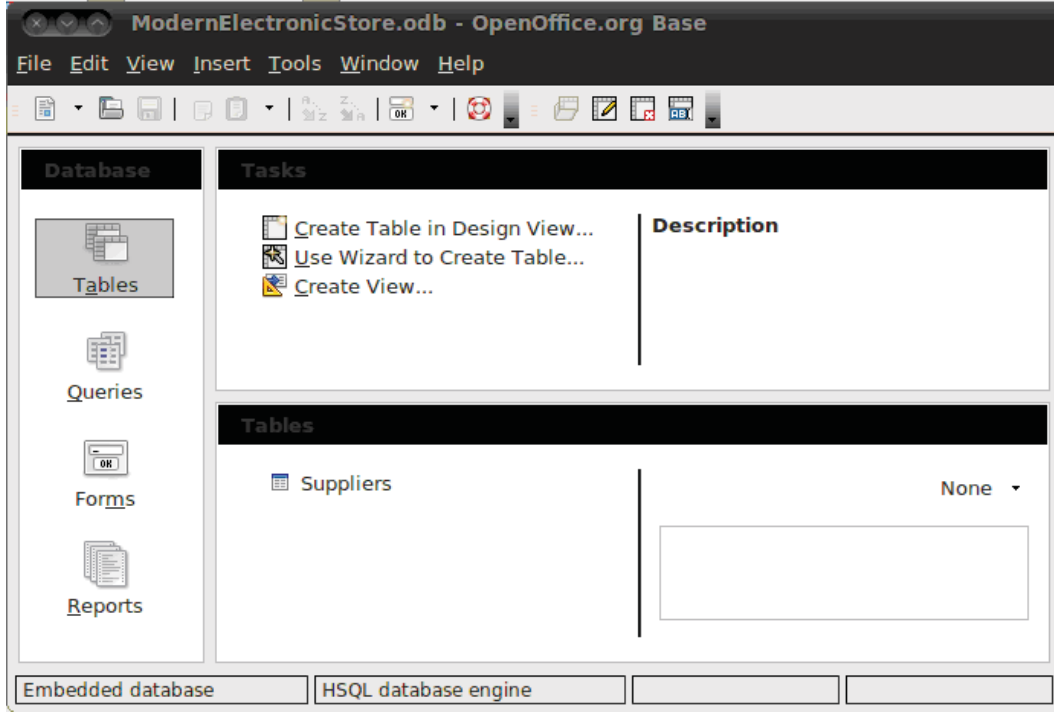
હવે *Available fields* ની યાદીમાં દર્શ્યમાન બધાં ફિલ્ડને >> બટન ઉપર ક્લિક કરી પસંદ કરો. આના વિકલ્પ રૂપે કન્ટ્રોલ કી દબાવેલી રાખીને તમે ઈચ્છતા હોવ તે ફિલ્ડ એક પછી એક પસંદ કરો. આ કાર્ય તમે પૂરું કરો પછી > બટન ઉપર ક્લિક કરો. હાલપૂરતું આપણે >> બટન ઉપર ક્લિક કરો, જેથી બધાં ફિલ્ડ આપણા ટેબલના ભાગ બની રહે.

Finish બટન ઉપર ક્લિક કરો, ડેટાવ્યૂમાં Suppliers ટેબલ ખૂલશે. ડેટાવ્યૂમાં ટેબલ ખોલવાથી ઉપયોગકર્તાને ટેબલમાં રેકોર્ડ ઉમેરવા દેશે. આકૃતિ 9.13માં Suppliers ટેબલનો ડેટાવ્યૂ દર્શાવેલ છે.



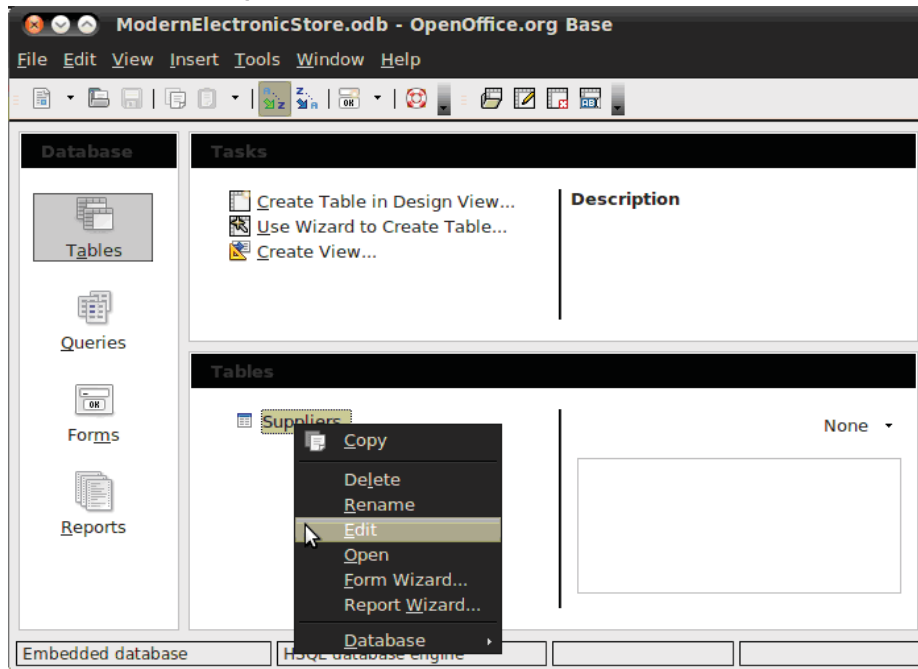
આકૃતિ 9.13 : Suppliers ટેબલનો ડેટાવ્યૂ

બેઝ દરેક ટેબલને અલગ-અલગ વિન્ડોમાં ખોલશે. Supplier ડેટાવ્યૂ વિન્ડો બંધ કરો. હવે ModernElectronicStore.odb વિન્ડો ખોલો અને આકૃતિ 9.14માં દર્શાવ્યા પ્રમાણે ડેટાબેઝની મુખ્ય વિન્ડો જોવા મળશે.



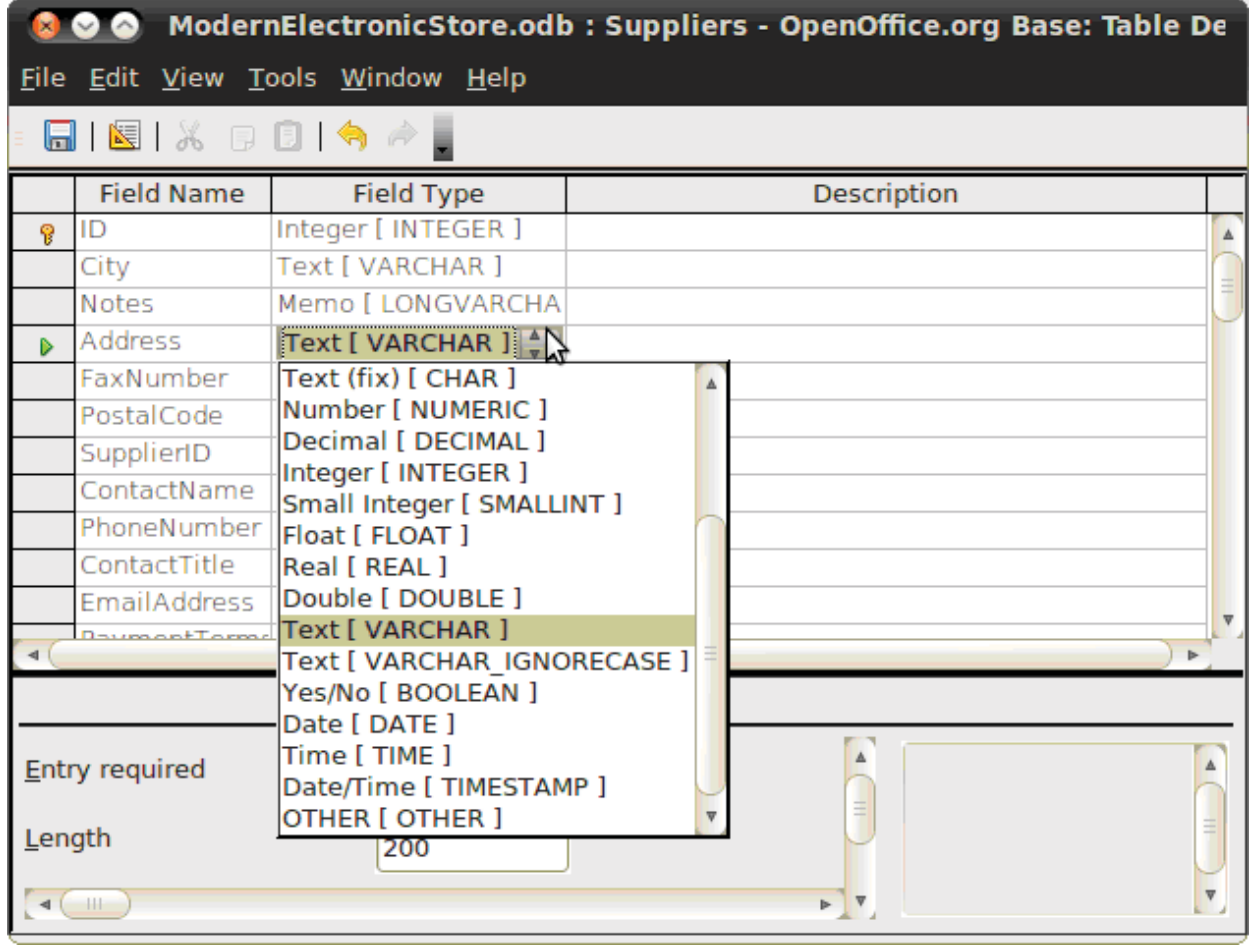
આકૃતિ 9.14 : ડેટાબેઝ વિન્ડો

ડાબી બાજુના વિભાગમાં (pane) તમે ડેટાબેઝ ઓબ્જેક્ટ્સ ટેબલ, ક્વેરીઝ, ફોર્મ્સ અને રિપોર્ટ્સ જોઈ શકશો. ટેબલ ઓબ્જેક્ટ એ પૂર્વનિર્ધારિત (ડિફોલ્ટ) પસંદગી છે અને તમે Tables લેબલ નીચે Suppliers ટેબલ જોઈ શકશો. ચાલો, હવે Suppliers ટેબલનાં ફિલ્ડની વિગત જોઈએ. Suppliers ટેબલ આઈકોન ઉપર રાઈટ ક્લિક કરો. ત્યાર પછી ઉપલબ્ધ વિકલ્પોમાંથી Edit વિકલ્પ પસંદ કરો (જુઓ આકૃતિ 9.15).



આકૃતિ 9.15 : Suppliers ટેબલમાં સુધારા-વધારા કરવા

આકૃતિ 9.16માં દર્શાવ્યા પ્રમાણે Suppliers ટેબલ હવે ડિઝાઇનવ્યૂમાં ખૂલેલું છે. હવે આપણે ટેબલના ફિલ્ડને એડિટ, અપડેટ અથવા ડિલીટ કરી શકીએ છીએ.



આકૃતિ 9.16 : કોઈ ફિલ્ડ માટે ડેટાટાઈપ પસંદ કરવો

આ વ્યૂમાં દરેક ફિલ્ડને વ્યાખ્યાયિત કરવું એ નીચે જણાવેલી ચાર પદ્ધતિઓ સાથે સંકળાયેલું છે :

1. ફિલ્ડનેઈમ
2. ફિલ્ડટાઈપ : તે આપણને દરેક ફિલ્ડની ડેટાટાઈપ નક્કી કરવા દે છે.
3. ડિસ્ક્રિપ્શન : જોકે ટેબલનાં દરેક ફિલ્ડ માટે વર્ણન લખવું ફરજિયાત નથી પણ દરેક ફિલ્ડ માટે વર્ણન લખવું એ એક સારી આદત છે. દરેક ફિલ્ડનો શું હેતુ છે, એ સમજવામાં વપરાશકર્તાને વર્ણન મદદરૂપ થાય છે.
4. ફિલ્ડ પ્રોપર્ટિઝ : ડેટા જે ઉમેરવાનો છે, તેના ઉપર નિયંત્રણ માટે અને યથાર્થતા (Validation) ચકાસવા માટે આનો ઉપયોગ થાય છે. જે ડેટાટાઈપ પસંદ કરેલ હોય છે, તે પ્રમાણે તેને સંબંધિત ફિલ્ડ પ્રોપર્ટિઝ પ્રદર્શિત થાય છે. તેમાં કેટલીક પૂર્વનિર્ધારિત પસંદગી પણ હોય છે. આપણી જરૂરિયાત પ્રમાણે બધી જ ફિલ્ડ પ્રોપર્ટિઝ આપણે બદલી શકીએ છીએ. હવે પછીના પ્રકરણમાં વિવિધ ફિલ્ડ પ્રોપર્ટિઝની ઊંડાણપૂર્વક ચર્ચા કરવામાં આવશે.

આપણે આ પ્રકરણમાં અગાઉ ચર્ચા કરી ગયા કે એડ્રેસ ફિલ્ડ માટે Text[VARCHAR_IGNORECASE] ડેટાટાઈપ આપણે વાપરીશું. આકૃતિ 9.16માં દર્શાવ્યા પ્રમાણે એડ્રેસ ફિલ્ડ સામેની ફિલ્ડ ટાઈપ કોલમમાં ડેટાટાઈપના ડ્રોપડાઉન ઓરો

ઉપર ક્લિક કરો. હવે Text[VARCHAR_IGNORECASE] ડેટાટાઈપ પસંદ કરો. તમે એડ્રેસફિલ્ડનું નામ બદલીને AddressLine1 કરી શકો અને તેનો ડેટાટાઈપ પણ અપડેટ કરી શકો. આ જ પ્રમાણે તમારી જરૂરિયાત પ્રમાણે Products, Employees અને Customers ટેબલ બનાવી શકાય છે, તેમજ તેમાં સુધારા-વધારા કરી શકાય છે.

ઓબ્જેક્ટ્સ દૂર કરવા (Deleting Objects)

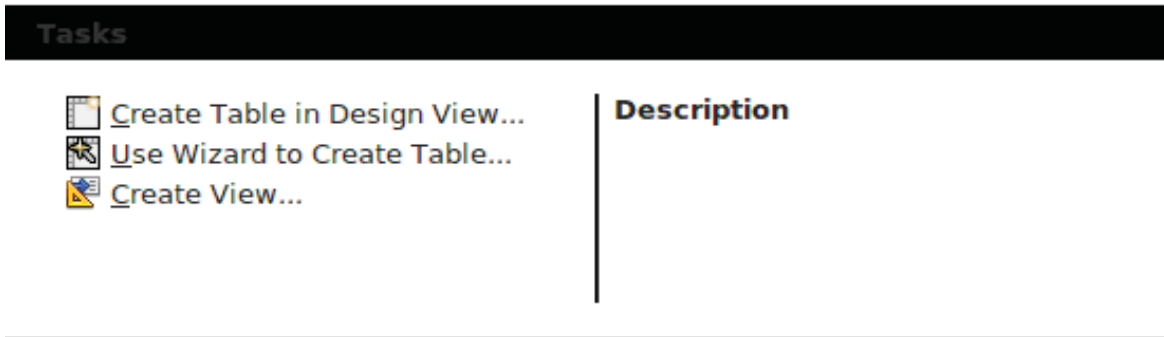
બેઝમાં બનાવેલો કોઈ પણ ઓબ્જેક્ટ ડિલીટ કરવા માટે, ઓબ્જેક્ટના આઈકોન ઉપર રાઈટ ક્લિક કરો અને Delete વિકલ્પ ઉપર ક્લિક કરો. વિઝાર્ડ વડે બનાવેલાં Suppliers ટેબલ સાથે બધાં ટેબલ ડિલીટ કરો. આકૃતિ 9.6 અને 9.7માં દર્શાવ્યા પ્રમાણે સ્પેશીફિકેશન (વિગતવાર વર્ણન) મુજબ આપણે ડિઝાઈનવ્યૂમાં જરૂરી ટેબલ ફરી બનાવીશું.

નોંધ : ડિઝાઈનવ્યૂમાં Supplier, Product, Employee અને Customer ટેબલ બનાવતા સમયે અનુક્રમે Scode, Pcode, Ecode અને Ccode ફિલ્ડ ઉમેરીશું.

ડિઝાઈન વ્યૂમાં ટેબલ બનાવવું (Creating Table using Design View)

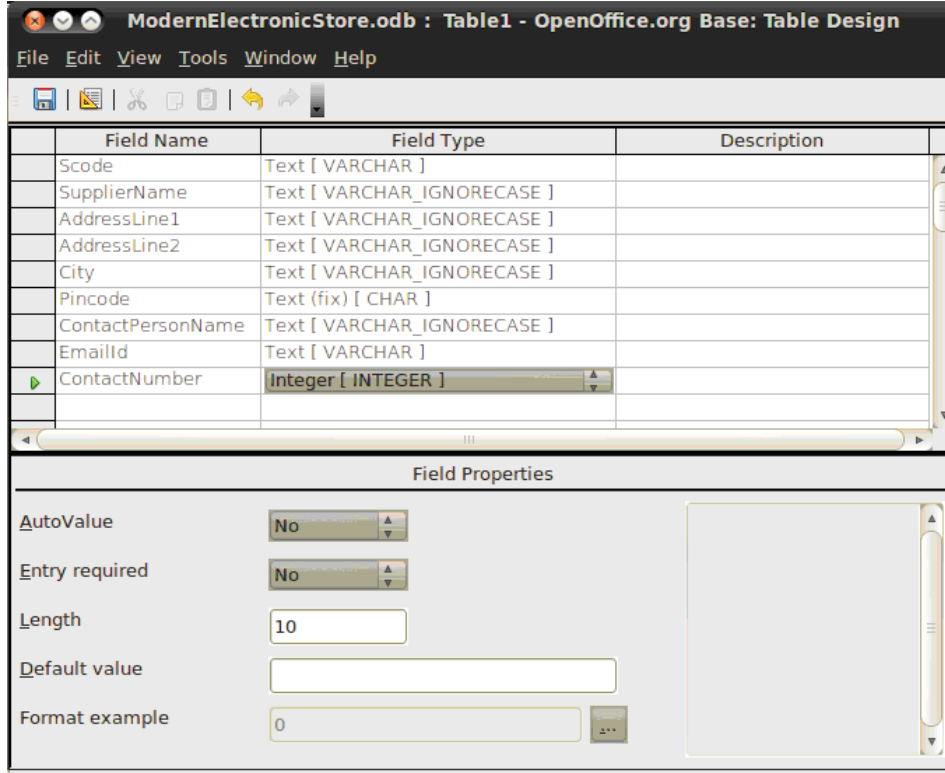
તમે કદાચ નિરીક્ષણ કર્યું હશે કે ટેબલ બનાવવા માટે આપણે જ્યારે વિઝાર્ડનો ઉપયોગ કરીએ છીએ, ત્યારે અગાઉથી ડિઝાઈન કરેલાં (pre-designed) ટેબલમાં કેટલાક ફેરફાર કરવા પડે છે, જેથી તે સહેલાઈથી વાપરી શકાય. વાસ્તવિક સ્થિતિમાં બેઝ દ્વારા પૂરાં પાડવામાં આવેલાં ટેબલ પર્યાપ્ત નથી. તેથી ડિઝાઈન વ્યૂમાં બનાવેલાં ટેબલ વધારે ફાયદાકારક છે.

ચાલો, આપણે ડિઝાઈનવ્યૂમાં ટેબલ બનાવીએ. આકૃતિ 9.17માં દર્શાવ્યા પ્રમાણે ડેટાબેઝની મેઈન વિન્ડોમાં ટાસ્કપેનમાં ટેબલ બનાવવા માટેનાં બે વિકલ્પો (1) *Create Table in Design View...* અને (2) *Use Wizard to Create Table...* ઉપલબ્ધ હોય છે. હવે *Create Table in Design View* વિકલ્પ ઉપર ક્લિક કરો. આથી Table Design વિન્ડો ખૂલશે.



આકૃતિ 9.17 : ડેટાબેઝ વિન્ડોમાં ટાસ્કપેન

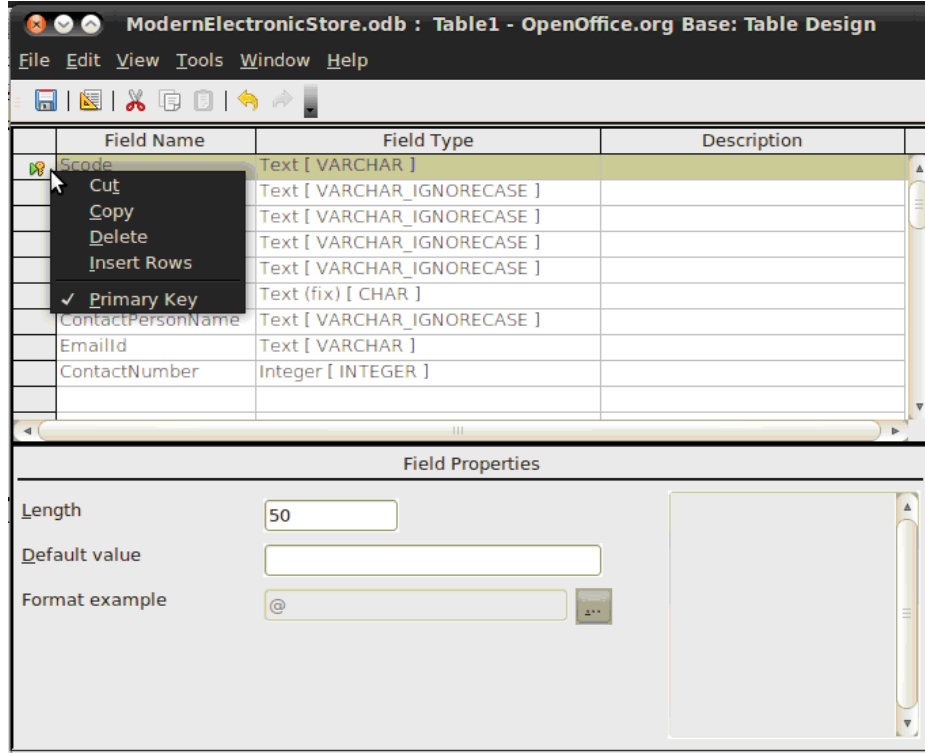
આકૃતિ 9.18માં દર્શાવ્યા પ્રમાણે ટેબલ ડિઝાઈન વ્યૂ એ એક જાળી જેવું માળખું (ગ્રીડ સ્ટ્રક્ચર) છે, જેમાં ત્રણ કોલમ Field Name, Field Type અને Description હોય છે અને વિન્ડોની મધ્યમાં નીચે Field Properties વિભાગ હોય છે. આકૃતિ 9.6માં ચર્ચા કરી હતી, તે પ્રમાણે Supplier ટેબલ બનાવો. ફિલ્ડનું નામ ટાઈપ કરો અને દરેક ફિલ્ડ માટે ફિલ્ડટાઈપ પસંદ કરો અને પહેલા ફિલ્ડ તરીકે Scode ઉમેરવાનું ન ભૂલો.



આકૃતિ 9.18 : ટેબલ ડિઝાઇનવ્યૂ વિન્ડો

પ્રાઇમરી કી સેટ કરો (Set Primary Key)

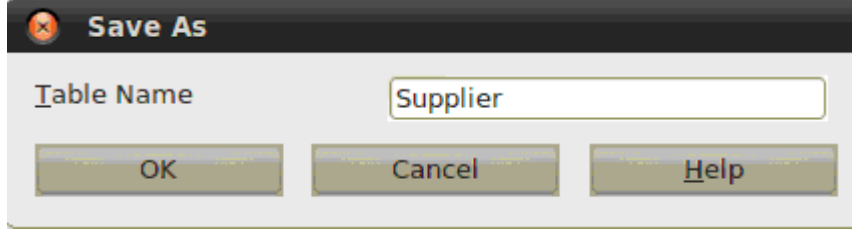
કોઈ ફિલ્ડને પ્રાઇમરી કી તરીકે ઓળખાવવા માટે Scode ફિલ્ડ ઉપર રાઇટ ક્લિક કરો, જેથી આકૃતિ 9.19માં દર્શાવ્યા પ્રમાણે સબ-મેનુ ખૂલશે. *Primary Key* વિકલ્પ પસંદ કરો અને પસંદ કરેલા ફિલ્ડની પહેલી કોલમમાં કી જેવું દેખાતું ચિહ્ન (🔑) પ્રદર્શિત થશે.



આકૃતિ 9.19 : ટેબલની પ્રાઇમરી કી નક્કી કરવી

ટેબલ સેવ કરો (Save Table)

ટેબલનું માળખું બનાવ્યા પછી હવે ટેબલનો સંગ્રહ કરવાનો સમય છે. આકૃતિ 9.19માં દર્શાવ્યા પ્રમાણે Save બટન () ઉપર ક્લિક કરો. આથી આકૃતિ 9.20માં દર્શાવ્યા પ્રમાણે Save as ડાયલોગબોક્સ પ્રદર્શિત થશે. હવે ટેબલનું નામ ટાઈપ કરો અને OK બટન ઉપર ક્લિક કરો.



આકૃતિ 9.20 : ટેબલ સેવ કરવું

આ જ પ્રમાણે Product, Employee અને Customer ટેબલ બનાવો. અત્યાર સુધીમાં બનાવેલાં માસ્ટર ટેબલ (Master Tables) તરીકે ઓળખાય છે. આ ટેબલ “મોડર્ન ઇલેક્ટ્રોનિક્સ સ્ટોરમાંથી મોબાઈલ ફોનની ખરીદી કરતો ગ્રાહક” અથવા “મોડર્ન ઇલેક્ટ્રોનિક્સ સ્ટોરની તેના સપ્લાયર સોની ઇન્ડિયા પાસેથી ખરીદીની વિગત” જેવા વ્યવહારોની વિગતનો સંગ્રહ કરે છે. વ્યવહારો (ટ્રાન્ઝેક્શન્સ)ના સંગ્રહ માટે અલગ-અલગ ટેબલની ડિઝાઈન કરવી અને પછી તેને માસ્ટર ટેબલ સાથે સાંકળી લેવાં વધારે યોગ્ય છે. આ પછીના પ્રકરણમાં આપણે ટ્રાન્ઝેક્શન ટેબલ વિશે ચર્ચા કરીશું.

સારાંશ

આ પ્રકરણમાં આપણે ડેટા, ઇન્ફર્મેશન, ડેટાબેઝ અને ડેટાબેઝ ઇન્ફર્મેશન સિસ્ટમ (DBMS) ની અગત્યતા વિશે ચર્ચા કરી. સામાન્ય રીતે કોઈ સીધી જ હકીકત કે માહિતી ડેટાનો નિર્દેશ કરે છે, જ્યારે ડેટા ઉપર પ્રક્રિયા કરવામાં આવે છે, ત્યારે તે ઇન્ફર્મેશન બને છે અને યોગ્ય રીતે ગોઠવાયેલ માહિતીનો ડેટાબેઝમાં સંગ્રહ કરવામાં આવે છે. DBMS ટેબલ, ક્વેરી, રિપોર્ટ અને ફોર્મ જેવાં વિવિધ ઓબ્જેક્ટ્સનો સમાવેશ કરે છે. આપણે ડેટાબેઝની ડિઝાઈન માટે જરૂરી પગલાંઓની પણ ચર્ચા કરી. આ ઉપરાંત આપણે વિઝાર્ડ વડે અને ડિઝાઈન વ્યૂમાં ટેબલ બનાવવા માટે ઓપન ઓફિસ બેઝ સોફ્ટવેરના ઉપયોગ વિશે શીખ્યાં. આ પછીના પ્રકરણમાં આપણે બેઝમાં ટેબલને કઈ રીતે સાંકળી શકાય અને ટેબલ ઉપર પ્રક્રિયાઓ કરી શકાય તે બાબત શીખીશું.

સ્વાધ્યાય

1. ડેટા અને ઇન્ફર્મેશન પદાવલિઓ ઉદાહરણ સાથે સમજાવો.
2. ઇન્ફર્મેશનની જરૂરિયાત સમજાવો.
3. ડેટાબેઝ, ડેટામોડેલ અને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ પદાવલિઓ વ્યાખ્યાયિત કરો.
4. ટેબલ, રેકોર્ડ અને ફિલ્ડને ઉદાહરણ તેમજ આકૃતિ સાથે વ્યાખ્યાયિત કરો.
5. લોજિકલ ફિલ્ડ નામ અને ફિઝિકલ ફિલ્ડ નામ શું છે ?
6. ફિલ્ડનું નામ નક્કી કરતાં સમયે અનુસરવામાં આવતી સૂચનાઓ લખો.
7. ડેટાબેઝની ડિઝાઈન કરવાનાં પગલાંઓ લખો.
8. Text (fix)[CHAR] અને Text[VARCHAR] વચ્ચેનો તફાવત શું છે ? યોગ્ય ઉદાહરણ આપી સમજાવો.

9. કી-ફિલ્ડની વ્યાખ્યા આપો.
10. પ્રાઈમરી કી શું છે ? યોગ્ય ઉદાહરણ આપી સમજાવો.
11. પ્રાઈમરી કીની પ્રોપર્ટીઝ શું-શું છે ?
12. ડેટાટાઈપ એટલે શું ? યોગ્ય ઉદાહરણ આપી સમજાવો.
13. મેમો ડેટાટાઈપનો ઉપયોગ ક્યારે કરવો જોઈએ?
14. બેઝમાં ઉપલબ્ધ ઓબ્જેક્ટ્સની યાદી બનાવો.
15. નલવેલ્યુ એટલે શું? યોગ્ય ઉદાહરણ આપી સમજાવો.
16. **હેતુલક્ષી પ્રશ્નો**

(1) DBMS શું છે ?

- | | |
|-------------------------------|------------------------------------|
| (a) ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ | (b) ડેટાબેઝ માઈગ્રેશન સિસ્ટમ |
| (c) ડેટા મેનેજમેન્ટ સિસ્ટમ | (d) ડાઈરેક્ટ બેઝ મેનેજમેન્ટ સિસ્ટમ |

(2) “રો (કાચી) હકીકત અથવા આંકડાઓ” કઈ પદાવલિ માટે વપરાય છે ?

- | | |
|----------------|-------------------|
| (a) ઈન્ફર્મેશન | (b) વેરીએબલ (અચલ) |
| (c) ડેટા | (d) ફિલ્ડ |

(3) ડેટાબેઝમાં ગ્રાહક માટે નીચેનામાંથી શું સૌથી સારી રીતે નિર્દેશ કરે છે ?

- | | |
|---------------|----------------|
| (a) રિલેશનશિપ | (b) એટ્રીબ્યુટ |
| (c) એન્ટિટી | (d) ડેટા |

(4) પ્રક્રિયા કરેલા ડેટા શું છે ?

- | | |
|----------------|----------------------|
| (a) હકીકત | (b) તૈયાર કરેલા ડેટા |
| (c) ઈન્ફર્મેશન | (d) નિર્ણય |

(5) નીચેનામાંથી કયો ડેટાબેઝ પ્રોગ્રામ નથી ?

- | | |
|--------------------|----------------|
| (a) માય એસક્યુએલ | (b) બેઝ |
| (c) એસક્યુએલ સર્વર | (d) સ્પ્રેડશીટ |

(6) નીચેનામાંથી કયું ઓપનસોર્સ DBMS નથી ?

- | | |
|------------------|--------------|
| (a) માય એસક્યુએલ | (b) ઈન્ગ્રેસ |
| (c) બેઝ | (d) ઓરેકલ |

(7) RDBMSમાં R નીચેનામાંથી શેના માટે વપરાય છે?

- | | |
|-------------|-------------|
| (a) રોટેશનલ | (b) રિલેશનલ |
| (c) રેન્ડમ | (d) રિંગ |

(8) ઓપન ઓફિસ સ્યૂટના ભાગ તરીકે ઉપલબ્ધ ડેટાબેઝનું નામ શું છે ?

- | | |
|--------------|------------|
| (a) OfficeDB | (b) Base |
| (c) OpenDB | (d) Access |

(9) નીચેનામાંથી શું બેઝ ડેટાબેઝ ફાઈલનું ડિફોલ્ટ અનુલંબન હોય છે ?

- | | |
|----------|----------|
| (a) .bdf | (b) .odf |
| (c) .odb | (d) .dbf |

(10) નીચેનામાંથી શું બેઝમાં ઓબ્જેક્ટ તરીકે જાળવવામાં આવતું નથી ?

- (a) ટેબલ (b) ક્વેરીઝ
(c) ચાર્ટ્સ (d) ફોર્મ્સ

(11) સંખ્યાનો સંગ્રહ કરવા માટે કયો ડેટાટાઇપ વાપરી ન શકાય ?

- (a) Decimal (b) Integer
(c) Text (d) Date

(12) બેઝ ડેટાબેઝમાં ઈમેજનો સંગ્રહ કરવા માટે કયો ડેટાટાઇપ વપરાય છે ?

- (a) બાયનરી (b) ફોટો
(c) લોંગ (d) લુજ

(13) નીચેનામાંથી કયું વાક્ય ખોટું છે ?

- (a) પ્રાઈમરી કીમાં નલવેલ્યુ ન હોઈ શકે.
(b) પ્રાઈમરી કીમાં ડુપ્લિકેટ વેલ્યુ ન હોઈ શકે.
(c) પ્રાઈમરી કી એક કરતાં વધારે ફિલ્ડનું જોડાણ હોઈ શકે.
(d) પ્રાઈમરી કી હંમેશાં ન્યુમરિક ફિલ્ડ હોવું જોઈએ.

પ્રાયોગિક સ્વાધ્યાય

1. આકૃતિ 9.21માં Student, Teacher અને Subject ટેબલનાં ફિલ્ડ અને તેનું વર્ણન આપેલાં છે. દરેક ફિલ્ડ માટે ડેટાટાઇપ નક્કી કરો અને બેઝમાં ટેબલ બનાવો. તેમાં ઓછામાં ઓછા 5 રેકોર્ડ એન્ટર કરો.

Student	
Field Name	Description
Gno	General Register Number
Firstname	Name of the Student
Surname	Surname of the Student
Address	Address of the Student
City	City
Pincode	Pincode
Birthdate	Date of Birth
Gender	Male or Female
Standard	Studying in which standard
Join_date	Date of Joining School
Leaving_date	Date of Leaving School

Teacher	
Field Name	Description
Firstname	Name of the Teacher
Surname	Surname of the Teacher
Address	Address of the Teacher
City	City
Pincode	Pincode
Phone_no	Phone number of teacher
Email_id	E-mail id of teacher
Mobile_no	Mobile number of teacher

Subject	
Field Name	Description
Sub_Name	Name of the Subject
Details	Description of the subject

આકૃતિ 9.21 : સ્કૂલમેનેજમેન્ટ સિસ્ટમ માટેનાં ટેબલ

2. આકૃતિ 9.22માં દર્શાવ્યા પ્રમાણે બેઝમાં ટ્રાન્ઝેક્શન ટેબલ બનાવો. તેમાં ઓછામાં ઓછા 5 રેકોર્ડ એન્ટર કરો.

Standard_Subj : To store details about subjects taught in each standard		
Field Name	Data Type	Description
Standard	Number	Which Standard
Scode	Text	Which Subject

Attendance : To store daily attendance of the Students		
Field Name	Data Type	Description
Gmno	Number	General Register of a Student
Date	Date/Time	On which Date
Pr_ab	Yes/No	Present or Absent

Fees : To store details of Fee amount paid by students		
Field Name	Data Type	Description
Gmno	Number	General Register Number
Date	Date/Time	On which date
Tuition_fee	Currency	Tuition fee paid by the student
Term_fee	Currency	Term fee paid by the student

Cultural : To keep record of Interest of students in Cultural Activities		
Field Name	Data Type	Description
Gmno	Number	General Register Number
Interest	Text	Interested in which activity
Achievement	Memo	Any achievements?

Subj_Teacher : To store details about which teacher is teaching the subject in each standard		
Field Name	Data Type	Description
Year	Number	Current Year
Standard	Number	In which Standard
Scode	Text	Subject Code
Tcode	Text	Teacher Code

Result : To store result details of each student		
Field Name	Data Type	Description
Gmno	Number	General Register No.
Scode	Text	Of which Subject
Month	Text	In which Month
Year	Number	Which Year
Term	Number	Name of the Term
Marks	Number	Marks scored

Sports : To keep record of Interest of students in Sports		
Field Name	Data Type	Description
Gmno	Number	General Register No.
Interest	Text	Interested in which game
Achievement	Memo	Any achievements?

આકૃતિ 9.22 : સ્કૂલ મેનેજમેન્ટ સિસ્ટમ માટેનાં ટેબલ





આગળના પ્રકરણમાં ચર્ચા કર્યા પ્રમાણે આપણે દરરોજ અનેક નિર્ણયો લેવા પડે છે અને તે માટે એક અથવા બીજા પ્રકારની માહિતીની આપણને જરૂર પડે છે. યોગ્ય સમયે સાચી માહિતી મેળવવા માટે આપણે ડેટાબેઝ બનાવવો પડે અને તેમાં ડેટા ઉમેરવા, અગાઉ દાખલ કરેલા ડેટામાં સુધારા કરવા, નકામો અને બિનજરૂરી ડેટા રદ કરવા, ડેટાને જરૂરિયાત પ્રમાણે ગોઠવવા જેવી વિવિધ પ્રક્રિયાઓ કરવી પડે છે. આપણે આગળના પ્રકરણમાં, બેઝમાં ડેટાબેઝ અને વિવિધ ટેબલ કેવી રીતે બનાવી શકાય તેની ચર્ચા કરી ચૂક્યા છીએ. હવે, આપણે બનાવેલાં ટેબલ ઉપર વિવિધ પ્રક્રિયાઓ કેવી રીતે કરી શકાય તે શીખીએ.

ટેબલમાં ડેટા ઉમેરવો (Inserting Data In the Table)

ટેબલમાં રેકોર્ડ ઉમેરવા માટે સૌપ્રથમ આપણે તે ટેબલ ઓપન કરવું પડે. ટેબલ ઓપન કરવા માટે ડેટાબેઝ વિન્ડોના ટેબલ પેનમાંથી જરૂરી ટેબલનાં આઈકોન ઉપર ડબલ ક્લિક કરો. આના વિકલ્પ રૂપે જરૂરી ટેબલ ઉપર ડબલ ક્લિક કરી, પ્રદર્શિત થયેલાં સબ-મેનુમાંથી ઓપન વિકલ્પ પસંદ કરો.


ચાલો, આપણે ડિઝાઈન વ્યૂમાં બનાવેલું Supplier ટેબલ ખોલીએ. જ્યારે આપણે Supplier ટેબલ ખોલીએ તો આકૃતિ 10.1માં દર્શાવ્યા પ્રમાણે ડેટાશીટવ્યૂ (datasheet view)માં તેનું સ્ટ્રક્ચર પ્રદર્શિત થશે.


Score	SupplierName	AddressLine1	AddressLine2	City	Pincode	ContactPersonName	EmailId	ContactNumber

Record 1 of 1

આકૃતિ 10.1 : શરૂઆતનો ડેટાએન્ટ્રી સ્ક્રીન

અહીં, આપણે જોઈ શકીએ છીએ કે ફિલ્ડનાં નામ આડી લીટીમાં પ્રદર્શિત થાય છે. ફિલ્ડનાં નામવાળી લીટી ટાઈટલ લાઈન તરીકે ઓળખાય છે. આપણે જોઈ શકીએ છીએ કે તે ફક્ત કેટલાક મર્યાદિત ફિલ્ડને દર્શાવી શકે છે, જે સ્ક્રીનમાં સમાઈ શકે. બાકી રહેલાં ફિલ્ડને જોવા માટે સ્ક્રીનના નીચેના ભાગમાં રહેલા સ્ક્રોલબારનો ઉપયોગ કરવો પડે છે. ટાઈટલ-લાઈનની નીચે ખાલી બોક્સથી બનેલી એક હરોળ (Row) છે. આ ખાલી બોક્સમાં ઉચિત ડેટા ભરવાનાં કામને ડેટાએન્ટ્રી (Data Entry) કહેવામાં આવે છે. પહેલાં ફિલ્ડની પાસે એક નાના બોક્સમાં એક તીરનું ચિહ્ન (Pointing arrow sign) છે. આ બોક્સ 'રેકોર્ડ સિલેક્ટર આઈકોન'નો નિર્દેશ કરે છે. કેટલાક રેકોર્ડ સિલેક્ટર આઈકોન્સ અને તેનાં કાર્યો નીચે જણાવેલાં છે.

કાળો પોઈન્ટિંગ એરો આઈકોન  રેકોર્ડ પોઈન્ટર તરીકે ઓળખાય છે અને તે કોઈ ચોક્કસ સમયે ટેબલનો ચાલુ રેકોર્ડ (ચાલુ હરોળ) દર્શાવે છે. આનો અર્થ એ છે કે જો આપણે ટાઈપિંગ શરૂ કરીશું, તો ચાલુ રેકોર્ડમાં ફિલ્ડની કિંમતમાં ફેરફાર જોઈ શકાશે.

ફ્લેશ સાથેનો લીલો આઈકોન  એ ટેબલના અંતનું નિશાન છે. તે ટેબલના છેલ્લા રેકોર્ડ પછી પ્રદર્શિત થાય છે.

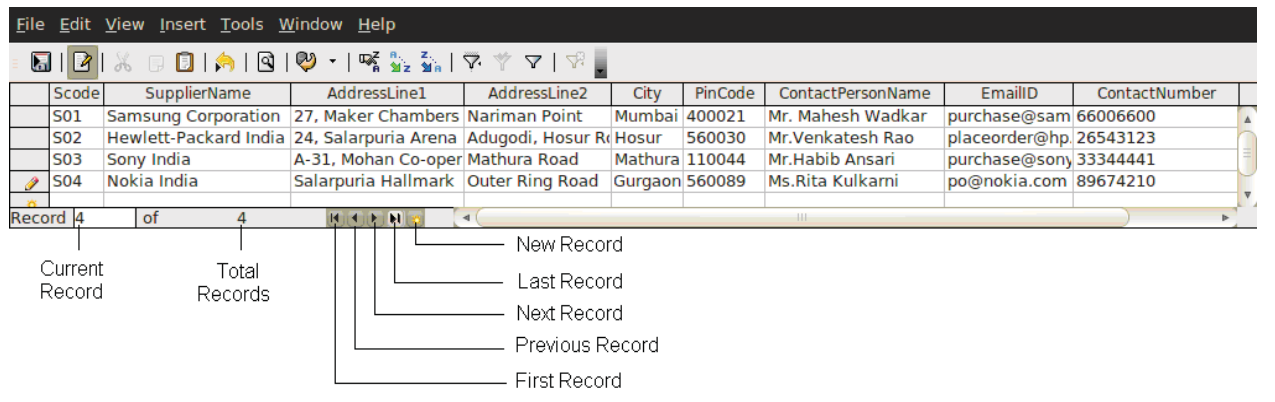
ડેટાશીટ વ્યૂમાં નવો રેકોર્ડ ઉમેરવા માટે છેલ્લી હરોળને સ્કોલ કરો અને પછી ક્લિક કરો. આમ કરવાથી આપણું કર્સર ફિલ્ડમાં ગોઠવાઈ જશે અને આઈકોન કાળો પોઈન્ટિંગ ઍરો (▶) બની જશે.

પેન્સિલ આઈકોન (✎) એ ઍડિટ આઈકોન (Edit icon) છે. જ્યારે ડેટા સુધારવા માટે કોઈ ફિલ્ડ ઉપર આપણે ક્લિક કરીએ, તો કાળો પોઈન્ટિંગ ઍરો આઈકોન બદલાઈને પેન્સિલ જેવો આઈકોન બની જશે. આ આઈકોન પ્રદર્શિત થશે જ્યારે રેકોર્ડમાં આપણે ફેરફાર કર્યા છે કે જે હજી સુધી સેવ કરેલ નથી. જો આપણે કર્સર બીજા રેકોર્ડ ઉપર ફેરવીશું, તો સુધારાનો સંગ્રહ થશે અને જો ESC કી દબાવીશું તો આપણે કરેલાં સુધારા રદ થશે અને મૂળ માહિતી પાછી મળશે.

આકૃતિ 10.2માં ડાબી બાજુએ નીચેના વિભાગમાં બારનું નિરીક્ષણ કરતાં, આપણે Record શબ્દ, તે પછી બોક્સમાં આંકડામાં કિંમત '4' અને અંતમાં શબ્દ 'of 4' જોઈ શકીશું. આ બાર 'નેવિગેશન બાર' (Navigation Bar) તરીકે ઓળખાય છે. તે રેકોર્ડને ઊભા (vertically) સ્કોલ કરવા માટે કેટલાક નેવિગેશન બટન્સ ધરાવે છે. તે રેકોર્ડ પોઈન્ટરની વર્તમાન સ્થિતિ પણ જણાવે છે. 'Record 4 of 4'નો અર્થ એ થાય કે આપણા ટેબલમાં કુલ 4 રેકોર્ડ છે અને હાલમાં આપણું કર્સર ચોથા રેકોર્ડ ઉપર છે.

ચાલો, હવે આપણે આગલા પ્રકરણમાં ચર્ચા કર્યા પ્રમાણે જુદાં-જુદાં ફિલ્ડમાં સંબંધિત ડેટા ભરીએ. આપણે દરેક ફિલ્ડ ડેટાટાઈપ અને ફિલ્ડ પ્રોપર્ટીથી વ્યાખ્યાયિત કરેલાં હોય તે પ્રમાણે બેઝમાં ડેટાએન્ટ્રી નિયંત્રિત અને પ્રતિબંધિત રહે છે. અહીં એ નિરીક્ષણ કરો કે વર્તમાન હરોળમાં છેલ્લા ફિલ્ડમાં ડેટા એન્ટર કરવાથી કર્સર આપોઆપ પછીની હરોળનાં પ્રથમ ફિલ્ડ Scodમાં આવી જાય છે.

બુલિયન પ્રકારના કિસ્સામાં ફિલ્ડમાં એક ચોરસ બોક્સ જોવા મળે છે. જો આપણે તેના ઉપર ક્લિક કરીએ, તો તેમાં ખરાનું ચિહ્ન જોવા મળશે. જો ખરાના ચિહ્ન સાથેનું બોક્સ જોવા મળે, તો એનો અર્થ એ થાય કે ફિલ્ડની કિંમત સાચી છે અને નહીંતર ખોટી. કેટલાક રેકોર્ડ ઉમેર્યા પછી આપણો સ્ક્રીન કંઈક અંશે આકૃતિ 10.2માં દર્શાવ્યા પ્રમાણે જોવા મળશે.



આકૃતિ 10.2 : કેટલાક રેકોર્ડ ઉમેર્યા પછી ડેટાશીટવ્યૂ

ટેબલના રેકોર્ડને ઍડિટ કરવા (Editing Records in the Table)

એક વખત ડેટા ઉમેર્યા પછી નીચે જણાવેલાં બેમાંથી કોઈ એક કારણથી તેમાં સુધારા કરવા પડે :

1. ડેટાએન્ટ્રી સમયે ખોટો ડેટા ઉમેર્યો હોય અથવા
2. સાચી ડેટાએન્ટ્રી કર્યા પછી ડેટાની કિંમતમાં બદલાવ આવ્યો હોય, તો આ કિસ્સામાં આપણે રેકોર્ડને સુધારવો પડશે. ઉદાહરણ તરીકે, ધારોકે કોઈ ચોક્કસ સપ્લાયરનું સાચું સરનામું આપણે ઉમેરેલું છે. કેટલાક સમય

પછી માલ મોકલનાર કંપની સ્થળ બદલે છે. આ કિસ્સામાં સપ્લાયરનું સાચું સરનામું ઉમેર્યું હોવા છતાં સ્થળ બદલાવા માટે આપણે તે એડિટ કરવું પડશે.

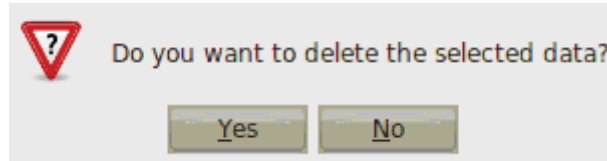
અગાઉ ઉમેરેલા ડેટાને સુધારવાની પ્રક્રિયા એડિટિંગ (Editing) તરીકે ઓળખાય છે. સુધારવાનું કાર્ય કરવા માટે, આપણે ટેબલ ખોલવું પડશે અને આપણે જેને સુધારવા ઇચ્છતા હોય તે ફિલ્ડની કિંમત ઉપર માત્ર કર્સર મૂકી અને જરૂરી સુધારા કરી.

ટેબલમાંથી રેકોર્ડ ડિલીટ કરવા (Deleting Records from the Table)

આપણા ડેટાબેઝને ક્ષતિ વિનાનો અને અર્થપૂર્ણ રાખવા માટે આપણા ટેબલમાંથી બિનજરૂરી અથવા ખોટા રેકોર્ડ્સને ડિલીટ કરવા (રદ કરવા) અતિજરૂરી છે. આમ કરવાથી સ્પષ્ટ ચિત્ર મળશે અને સાથે-સાથે ડિસ્કમાં કેટલીક જગ્યા મુક્ત થશે. ટેબલમાંથી કોઈ પણ રેકોર્ડ રદ કરવા માટે સૌપ્રથમ ટેબલ ઓપન કરો અને અનિચ્છનીય રેકોર્ડ પસંદ કરો. રેકોર્ડ પસંદ કર્યા પછી આપણે નીચે જણાવેલી બે રીતે તેને દૂર કરી શકીએ :

1. કી-બોર્ડ ઉપરની DEL કી દબાવીને અથવા એડિટ મેનુમાંથી ડિલીટ વિકલ્પ પસંદ કરીને.
2. પસંદ કરેલાં રેકોર્ડ ઉપર રાઈટ ક્લિક કરી, સબમેનુમાંથી ડિલીટ રેકોર્ડ વિકલ્પ પસંદ કરીને.

જ્યારે તમે રેકોર્ડ દૂર કરો છો, ત્યારે આકૃતિ 10.3માં દર્શાવ્યા પ્રમાણે બેઝ એક ચેતવણી સાથે ડાયલોગબોક્સ પ્રદર્શિત કરશે.






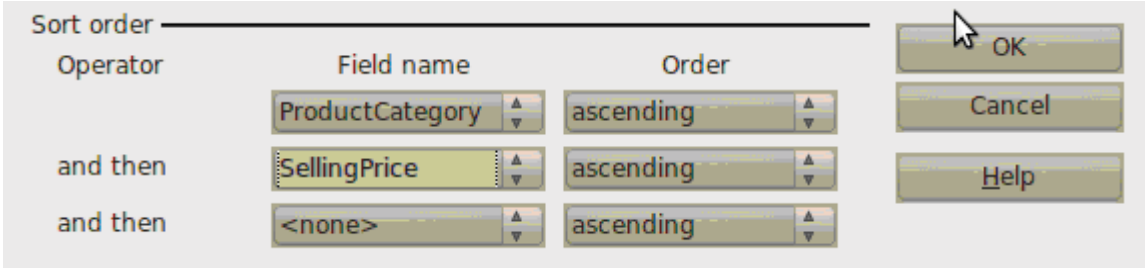
આકૃતિ 10.3 : રેકોર્ડ કાઢી નાખવાનું ચેતવણીબોક્સ

આ ડાયલોગબોક્સમાંથી Yes બટન ઉપર જો ક્લિક કરવામાં આવશે, તો પસંદ કરેલો રેકોર્ડ ટેબલમાંથી કાયમ માટે દૂર થશે. પણ જો આપણે No બટન ઉપર ક્લિક કરીશું, તો ટેબલમાંથી રેકોર્ડ દૂર નહીં થાય અને દૃષ્ટિમાં તે ફરી પ્રદર્શિત થશે.

ટેબલમાં ડેટા સોર્ટ કરવા (Sorting Data in the Table)

આપણે ડેટાબેઝ બનાવીએ અને તેની જાળવણી કરીએ છીએ, કારણકે આપણું અંતિમ ધ્યેય જ્યારે જરૂર હોય ત્યારે માહિતી મેળવવાનું હોય છે. ધીમે-ધીમે ટેબલમાં રેકોર્ડ વધશે, આથી સરળતાથી અને ખૂબ ઝડપે માહિતી મેળવવા માટે ટેબલનો ડેટા કોઈ ચોક્કસ ક્રમમાં ગોઠવાયેલો હોય એ ઇચ્છનીય છે. Product-ટેબલમાં વિવિધ ઉત્પાદનોમાં હજારો રેકોર્ડ હોઈ શકે. ધારો કે તમે કોઈ ચોક્કસ ઉત્પાદનની કિંમત સાથે વિગતો શોધવા માંગો છો. હવે આ કિસ્સામાં, જો પ્રોડક્ટટેબલ Pcode ફિલ્ડ એટલે કે પ્રોડક્ટ કોડ (product code)ના ક્રમમાં ગોઠવવામાં આવે, તો આપણા માટે તે શોધવાનું કામ ખૂબ જ સરળ બની જાય છે. પણ, ધારોકે આપણે પ્રોડક્ટ કોડ જાણતાં નથી, તો.....!! ચોક્કસપણે, આપણે તે ઉત્પાદનનો રેકોર્ડ સરળતાથી અને ખૂબ ઝડપી શોધી શકતા નથી. પણ જો આપણે Product ટેબલને Pcode અને Selling Price ફિલ્ડના ક્રમમાં ગોઠવેલું હોય તો જરૂરી રેકોર્ડ મેળવવો સરળ છે. આથી, ચાલો આપણે ટેબલને કઈ રીતે ક્રમમાં ગોઠવવું તે જોઈએ. ટેબલને નીચે જણાવેલી બેમાંથી એક રીતે ક્રમમાં ગોઠવી શકાય :

1. આપણે જે ફિલ્ડ ઉપર ક્રમમાં ગોઠવવા માંગીએ છીએ તે પસંદ કરો અને પછી આપણી જરૂરિયાત પ્રમાણે ટૂલ- બાર પરથી Sort Ascending બટન  અથવા Sort Decending બટન  ઉપર ક્લિક કરો.
2. આના વિકલ્પ રૂપે Sort બટન  ઉપર ક્લિક કરો. આથી આકૃતિ 10.4માં દર્શાવ્યા પ્રમાણે ડાયલોગબોક્સ પ્રદર્શિત થશે.



Operator	Field name	Order
	ProductCategory	ascending
and then	SellingPrice	ascending
and then	<none>	ascending

આકૃતિ 10.4 : સોર્ટના ક્રમની પસંદ

Field name ડ્રોપડાઉન બોક્સની નીચે યોગ્ય ફિલ્ડનું નામ અને Order ડ્રોપડાઉન બોક્સની નીચે જરૂરી સોર્ટિંગનો ક્રમ પસંદ કરો. આપણા કિસ્સામાં, Product ટેબલ product category ના ચઢતા ક્રમમાં (ascending order) પહેલાં ક્રમમાં ગોઠવવામાં આવશે અને પછી ડેટા selling price ફિલ્ડ ઉપર ફરી ક્રમમાં ગોઠવવામાં આવશે. ઉદાહરણ તરીકે બધા લેપટોપ તમામ મોબાઈલ કરતાં પહેલાં આવશે. લેપટોપ અને મોબાઈલ ગ્રુપની અંદર તેની વેચાણકિંમત પ્રમાણે ઓછી કિંમતની ઉત્પાદિત વસ્તુ યાદીમાં પ્રથમ અનુસાર ક્રમમાં રેકોર્ડ ગોઠવાશે.

ડેટા-નિરર્થકતા નિયંત્રણ (Controlling Data Redundancy)

હવે, અત્યારે તમે બેઝથી એકદમ માહિતગાર થઈ ગયા હશો તેથી આપણી નમૂનાની એપ્લિકેશન માટેના ડેટાબેઝની ડિઝાઇન બાબત ચર્ચા ચાલુ રાખીએ. આપણે પહેલેથી જ Supplier, Product, Employer અને Customer નામનાં ચાર ટેબલ્સની ડિઝાઇન કરેલી છે. ચાલો, હવે ટેબલની ડિઝાઇન ડેટા-નિરર્થકતાના દૃષ્ટિકોણથી ફરી તપાસીએ. ડેટા-નિરર્થકતા શબ્દ ડેટાની બિનજરૂરી દ્વિ-આવૃત્તિ (duplicate) અથવા પુનરાવર્તન માટે વપરાય છે.

આપણે જોયું છે કે Customer ટેબલમાં City નામનું ફિલ્ડ છે. મોડર્ન ઇલેક્ટ્રોનિક સ્ટોરમાં ઘણાં ગ્રાહકો હશે, જે એક શહેરના હોય. આથી, જો 'Ahmedabad' નામના શહેરના 100 ગ્રાહકો હોય, તો આ દરેક ગ્રાહક માટે 'Ahmedabad' શબ્દમાળાનું પુનરાવર્તન થશે. આમ, ડેટા-નિરર્થકતા સંગ્રહ માટેની જગ્યાના બગાડ (દુર્વ્યય) તરફ દોરી જાય છે અને અનેક કારણોસર બિનકાર્યક્ષમ છે. ઉદાહરણ તરીકે, નિરર્થક ડેટામાં કોઈ પણ ફેરફાર ડેટાબેઝમાં બહુવિધ સ્થળોએ ફેરફાર જરૂરી બનાવે છે.

ડેટાબેઝમાંથી નિરર્થક ડેટા નાબૂદ કરવા માટે ટેબલમાં સંગ્રહિત ડેટાની ગોઠવણી માટે ખાસ કાળજી આપણે રાખવી જોઈએ. ડેટાબેઝ ડિઝાઇનર એક ટેકનિકની મદદથી તેને શક્ય એટલી નાબૂદ કરવાનો પ્રયત્ન કરે છે તેને ડેટા-નોર્મલાઇઝેશન (Data-Normalization) કહેવામાં આવે છે. નોર્મલાઇઝેશન એક પ્રક્રિયા છે જે એક ટેબલમાંથી બહુવિધ ટેબલ્સમાં વિઘટન સૂચવે છે અને આ રીતે પેરેન્ટ-ચાઇલ્ડ રિલેશનશિપ (Parent-child relationship) બનાવે છે. આથી, આપણા નમૂનાના ડેટાબેઝમાં Customer ટેબલના કિસ્સામાં શહેરની માહિતીનો સંગ્રહ કરવા માટે નવું ટેબલ બનાવવું જોઈએ. ચાલો, આપણે

આ ટેબલને City નામ આપીએ. આપણે જાણીએ છીએ કે શહેરની અંદર દરેક વિસ્તાર એક અલગ જ પીનકોડ સાથે સંકળાયેલ હોય છે. શહેરની અંદર શહેર અને વિસ્તારને અજોડ રીતે ઓળખવા માટે પિનકોડ વાપરી શકાય. આથી City ટેબલના દરેક રેકોર્ડને ઓળખવા માટે પિનકોડને પ્રાઈમરી કી તરીકે વાપરી શકાય. કોષ્ટક 10.1માં City ટેબલનું માળખું અને તેના નમૂનાના રેકોર્ડ બતાવ્યા છે.

City		
Pincode	Area	CityName
380058	Bopal	Ahmedabad
380006	Law Garden	Ahmedabad
363421	Raska	Limbdi

કોષ્ટક 10.1 : City ટેબલ

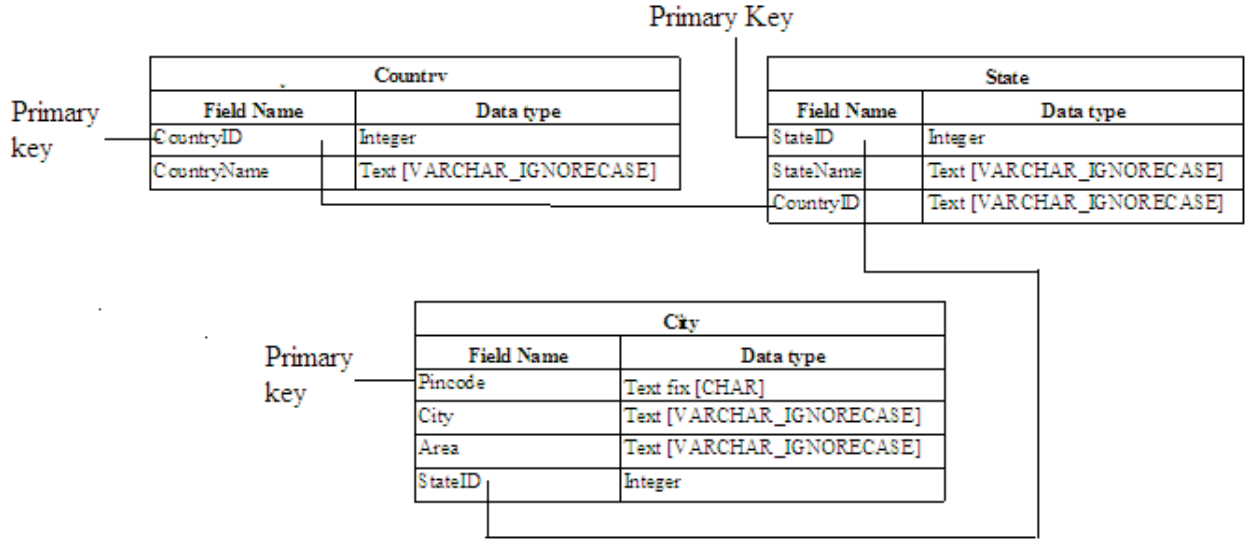
હવે આપણે Customer ટેબલમાં ફેરફાર કરવા પડશે. Customer ટેબલમાંથી City ફિલ્ડ આપણે દૂર કરી શકીએ. જ્યારે Customer ટેબલમાં નવા ગ્રાહકનો ડેટા ઉમેરવામાં આવે, ત્યારે ગ્રાહકના શહેર અને વિસ્તારની માહિતી મેળવવા માટે આપણે Pincode ફિલ્ડનો ઉપયોગ કરીશું.

એ જ રીતે, Supplier ટેબલ અને Employee ટેબલમાંથી City ફિલ્ડ દૂર કરવું જોઈએ. સંપૂર્ણ ટપાલના સરનામાની માહિતી જાળવી રાખવા માટે આપણે શહેર જે રાજ્ય અને દેશનું હોય તે State અને Country ફિલ્ડ પણ ઉમેરી શકાય. આ માહિતી જાળવવાની એક રીત કોષ્ટક 10.2માં દર્શાવી છે.

Pincode	Area	CityName	StateName	CountryName
380058	Bopal	Ahmedabad	Gujarat	India
380006	Law Garden	Ahmedabad	Gujarat	India
363421	Raska	Limbdi	Gujarat	India

કોષ્ટક 10.2 : રાજ્ય અને દેશ સાથેનું City ટેબલ

હવે જો પિનકોડ આપણને આપવામાં આવે, તો ગ્રાહકના સ્થળને લગતી માહિતી City ટેબલમાંથી શોધી શકાય છે. પરંતુ તમે અવલોકન કરી શકો છો કે ઉપર ડિઝાઇન કરેલું City ટેબલ એ જ ક્ષતિ ધરાવે છે, જેને ડેટા-નિરર્થકતા કહેવાય. એક જ શહેર અને રાજ્યનું ઘણી વખત પુનરાવર્તન થાય છે. ડેટા-નિરર્થકતાથી બચવા માટે, City ટેબલને ફરી અનેક ટેબલમાં વિઘટિત કરવાં પડે. StateName ફિલ્ડના સ્થાને StateID ફિલ્ડનો સંગ્રહ કરવો જોઈએ અને CountryName ફિલ્ડના સ્થાને CountryID ફિલ્ડનો સંગ્રહ કરવો જોઈએ. એક નવું ટેબલ કે જે રાજ્ય અને દેશની માહિતીનો સંગ્રહ કરે તે બનાવવું જોઈએ. આકૃતિ 10.5 આ ટેબલનું માળખું અને તેનો સંબંધ (Relation) દર્શાવે છે.



આકૃતિ 10.5 : Country, State અને City ટેબલ

અહીં આપણે Country, State અને City ટેબલમાં અનુક્રમે CountryID, StateID અને Pincode fields ને પ્રાથમરી કી તરીકે નક્કી કરેલ છે. આકૃતિ 10.5માં દર્શાવ્યા પ્રમાણે હવે City, State અને Country ટેબલ એકબીજાં સાથે સંકળાયેલાં છે. Customer, Employee અને Supplier ટેબલમાં પણ પીનકોડ એક ફિલ્ડ તરીકે હશે. આથી આ ટેબલ City ટેબલ સાથે સંકળાયેલાં છે. એ ટેબલ કે જેમાં પ્રાથમરી કી હોય તેને માસ્ટર ટેબલ કહેવામાં આવે છે. આપણે અગાઉ ચર્ચા કરી તે પ્રમાણે એક કોમન ફિલ્ડથી બે ટેબલ એકબીજાંથી સાંકળી શકાય છે. ઉપરના ઉદાહરણમાં CountryID ફિલ્ડનો ઉપયોગ કરીને State અને Country ટેબલને સાંકળેલાં છે. Country ટેબલમાં CountryID ફિલ્ડ પ્રાથમરી કીનું કાર્ય કરે છે. State ટેબલનું CountryID ફિલ્ડ Country ટેબલના CountryID ફિલ્ડ પાસેથી આ રીતે કિંમતો મેળવે છે. State ટેબલમાંનું CountryID ફિલ્ડ ફોરેન કી (Foreign key) તરીકે ઓળખાય છે.

ફોરેન કી ને એ રીતે વ્યાખ્યાયિત કરી શકાય કે તે એક ફિલ્ડ અથવા ફિલ્ડનું જોડાણ કે જેની કિંમત બીજા ટેબલમાંથી મેળવવામાં આવે છે. જ્યારે તે એક કરતાં વધારે ફિલ્ડનું જોડાણ હોય છે, ત્યારે તે કમ્પોઝિટ ફોરેન કી (Composite Foreign Key) તરીકે ઓળખાય છે. જ્યારે માસ્ટર ટેબલમાં કમ્પોઝિટ પ્રાથમરી કી હોય છે, ત્યારે તેની સાથે સંકળાયેલાં ટેબલમાં કમ્પોઝિટ ફોરેન કી હોય છે.

નોંધ : ફોરેન કી ફિલ્ડ કોઈ કિંમત ધરાવતું નથી જો તેની સાથે સંકળાયેલા પ્રાથમરી કી ફિલ્ડમાં તે અસ્તિત્વમાં ન હોય.

હવે જો તમે Product ટેબલ ઉપર નજર કરશો, તો ઉત્પાદન-શ્રેણી ‘મોબાઈલ’ અથવા ‘કેમેરા’ ઘણી વખત સંગૃહીત કરેલાં જોવા મળશે. આથી, એક અલગ ટેબલ ProductCategory નામનું બનાવવું પડશે. દરેક ઉત્પાદન-શ્રેણી (પ્રોડક્ટ કેટેગરી)ને હવે એક અજોડ નંબર આપવો પડશે. ઉત્પાદનની શ્રેણીનો નિર્દેશ કરવામાં આ અજોડ નંબરનો ઉપયોગ કરી શકાય. ઉદાહરણ તરીકે, મોબાઈલને કેટેગરી કોડ 1 આપી શકાય. આથી, Product ટેબલમાં જ્યારે નવા મોબાઈલ માટે રેકોર્ડ ઉમેરવામાં આવે, ત્યારે ProductCategory ફિલ્ડમાં કિંમત ‘1’નો સંગ્રહ કરવો જોઈએ. અહીં એ નોંધ કરશો કે Product ટેબલનાં ProductCategory ફિલ્ડની અને પ્રોડક્ટ કેટેગરી ટેબલમાં CategoryCode ફિલ્ડનો ડેટાટાઈપ એક-સમાન હોવો જોઈએ.

એ જ પ્રમાણે Product ટેબલના SupplierName ફિલ્ડને Scode ફિલ્ડથી બદલવું જોઈએ કે જે Supplayer કોડનો ઉલ્લેખ કરે છે. કોષ્ટક 10.3માં અત્યાર સુધી સૂચવેલા સુધારા પછીની પ્રાથમરી કી અને ફોરેન કીની યાદી આપેલ છે.

Table name	Primary Key	Foreign Key
Country	CountryID	-
State	StateID	CountryID (referencing Country table)
City	CityID	StateID (referencing State table)
Supplier	Scode	Pincode (referencing City table)
Customer	Ccode	Pincode (referencing City table)
Employer	Ecode	Pincode (referencing City table)
ProductCategory	CategoryCode	-
Product	Pcode	CategoryCode (Referencing ProductCategory table) Scode (Referencing Supplier table)

કોષ્ટક 10.3 : નમૂનાના ડેટાબેઝ માટે ચર્ચા કરેલા દરેક ટેબલની પ્રાઈમરી કી અને ફોરેન કી

ચાલો, હવે આપણે કોઈ ગ્રાહક દ્વારા ખરીદવામાં આવેલા કોઈ ઉત્પાદનના વ્યવહાર જેવા ડેટાનો સંગ્રહ કરવા માટે અન્ય ટેબલની ડિઝાઈન બનાવીએ. આપણે તેને ગ્રાહક વડે આપેલો પરચેઝ ઓર્ડર કહી શકીએ. ગ્રાહક ઓનલાઈન ઓર્ડર આપી શકે અથવા મોડર્ન ઇલેક્ટ્રોનિક સ્ટોરમાં આવીને પ્રોડક્ટ (ઉત્પાદન)ની ખરીદી કરી શકે. ગ્રાહક દ્વારા આપેલો ઓર્ડર સૂચવે છે કે ગ્રાહક, કર્મચારી અને વ્યવહાર થયેલું ઉત્પાદન સંબંધિત માહિતીની જાળવણી રાખવી પડે. આપણે ઓર્ડર આપ્યા તારીખ ચૂકી ન શકીએ. આ ઉપરાંત દરેક પ્રોડક્ટનો જથ્થો પણ સમાન રીતે મહત્વપૂર્ણ છે. ગ્રાહક એક જ મોડેલનાં અનેક નંગ ખરીદી શકે. કોષ્ટક 10.4માં Order ટેબલનાં ફિલ્ડ દર્શાવ્યાં છે.

Order	
FieldName	Data type
OrderID	Integer
OrderDate	Date
Ccode	Text fix [CHAR]
Ecode	Text fix [CHAR]
Pcode	Text fix [CHAR]
Quantity	Integer
SalePrice	Decimal

કોષ્ટક 10.4 : Order ટેબલ

અહીં OrderIDને પ્રાઈમરી કી તરીકે નક્કી કરેલી છે. આ કિસ્સામાં વધારાનું ફિલ્ડ જરૂરી છે, કારણકે અન્ય કોઈ ફિલ્ડ કે ફિલ્ડના સમૂહનું જોડાણ અજોડ રીતે રેકોર્ડ નક્કી કરી શકતું નથી. Order ટેબલનાં દરેક ફિલ્ડનું પુનરાવર્તન હોઈ શકે. નીચે જણાવેલો એક રેકોર્ડ દાખલ કરવા પ્રયત્ન કરો :

<1, 23-June-2012, C01, E01, P00000001, 2> હવે ધારો કે ગ્રાહકે બે જુદી-જુદી વસ્તુઓ ખરીદી છે, P00000001 અને P00000002. હવે નવો રેકોર્ડ ઉમેરવો પડશે, તે હશે : <1,23-June-2012,C01,E01, P00000002,1>.

તમે અવલોકન કરી શકો છો કે order date, employee code અને customer code જેવી વિગતનું પુનરાવર્તન કરવામાં આવેલું છે. જો ગ્રાહક એકસાથે 10 વસ્તુઓની ખરીદી કરે, તો 10 અલગ-અલગ રેકોર્ડ ઘણા બધા ડેટાના પુનરાવર્તન સાથે ઉમેરવામાં આવે છે. આ કિસ્સામાં OrderID પ્રાઈમરી કી તરીકે ન ચાલી શકે. OrderID અને Pcodeના જોડાણને પ્રાઈમરી કી બનાવી શકાય છે. આ ચર્ચા કરેલ સમસ્યાનો ઉકેલ Order ટેબલને બે ટેબલમાં વિઘટિત કરવાનો છે, એક Order ટેબલ અને બીજું OrderDetail ટેબલ. આથી OrderDate, Ccode અને Ecode ફિલ્ડનો સંગ્રહ OrderID સાથે ફક્ત એક જ વાર થાય છે અને તે Order ટેબલમાં અજોડ રીતે ઓળખાય છે. હવે OrderDetail ટેબલમાં OrderID, Pcode, Quantity અને SalePriceનો સમાવેશ થશે. OrderIDની કિંમતો Order ટેબલમાંથી મેળવવામાં આવશે અને આથી તે એક ફોરેન કી ફિલ્ડ હશે. આ રીતે OrderDetailમાં એવી વિગત છે જે રીપીટ થાય છે. કોષ્ટક 10.5માં OrderDetail ટેબલના નમૂનાના રેકોર્ડ બતાવ્યા છે.

OrderDetail			
OrderID	Pcode	Quantity	SalePrice
O00001	P00000001	2	35000
O00001	P00000002	1	20000
O00002	P00000009	1	43000

કોષ્ટક 10.5 : OrderDetail ટેબલના નમૂનાના રેકોર્ડ

હવે, OrderDetail ટેબલની પ્રાઈમરી કી શું હોવી જોઈએ ? આપણે અગાઉ ચર્ચા કર્યા પ્રમાણે, જે ફિલ્ડને પ્રાઈમરી કી બનાવેલ હોય તેની ડુપ્લિકેટ કિંમત ન હોઈ શકે. આવા સંજોગોમાં આપણે જોડાણ કરેલાં ફિલ્ડ શોધવા પડે, જેની કિંમત અજોડ હોય. OrderDetail ટેબલમાં OrderID અને Pcodeનું જોડાણ ચોક્કસપણે અજોડ હશે. એક જ ઓર્ડરમાં પ્રોડક્ટ કોડનું પુનરાવર્તન નહીં થાય. માટે બે ફિલ્ડનાં જોડાણથી પ્રાઈમરી કી બનાવવામાં આવે છે. જ્યારે એક કરતાં વધારે ફિલ્ડનાં જોડાણને પ્રાઈમરી કી તરીકે નક્કી કરવામાં આવે ત્યારે તેને *કમ્પોઝિટ પ્રાઈમરી કી (Composite Primary Key)* તરીકે ઓળખવામાં આવે છે. આકૃતિ 10.6માં Order અને OrderDetail ટેબલનાં માળખાં દર્શાવ્યાં છે.

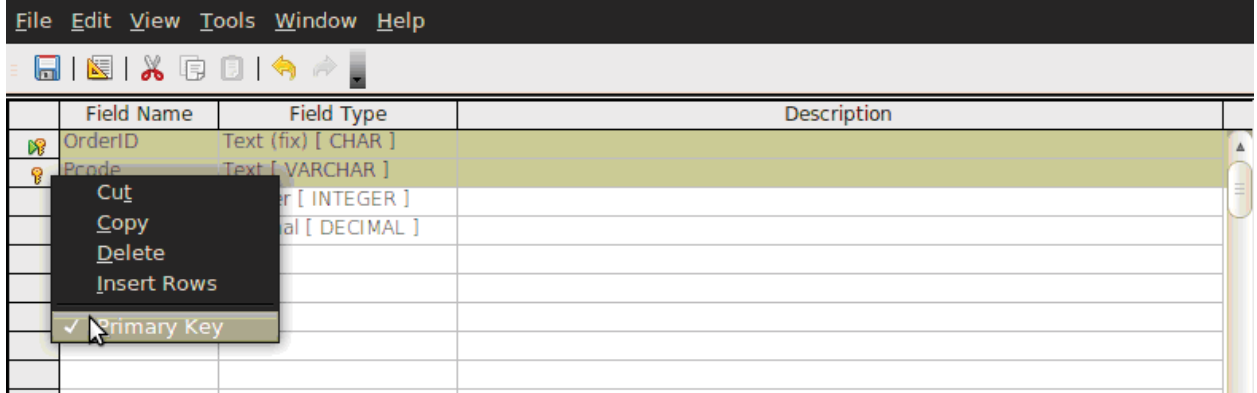
Order		OrderDetail	
To store details of orders		To store details of products purchased by customer	
Field Name	Data type	Field Name	Data type
OrderID	Integer	OrderID	Text fix [CHAR]
OrderDate	Date	Pcode	Text fix [CHAR]
Ccode	Text fix [CHAR]	Quantity	Integer
Ecode	Text fix [CHAR]	SalePrice	Decimal

આકૃતિ 10.6 : Order અને OrderDetail ટેબલનાં માળખાં (સ્ટ્રક્ચર)

કમ્પોઝિટ પ્રાઈમરી કી (Composite Primary Key)

OrderDetail ટેબલમાં કમ્પોઝિટ પ્રાઈમરી કી બનાવવા માટે

- OrderID ફિલ્ડ પસંદ કરો.
- CTRLકી દબાવો અને Pcode હરોળ પસંદ કરો. બન્ને હરોળો એકસાથે પસંદ થશે. જુઓ આકૃતિ 10.7.
- પસંદ કરેલાં એરિયા (વિસ્તાર) ઉપર રાઈટ ક્લિક કરો. આથી પ્રાઈમરી કી સાથેનું એક પોપઅપ મેનુ પ્રદર્શિત થશે.
- Primary key વિકલ્પ ઉપર ક્લિક કરો, આથી પસંદ કરેલી બે હરોળોની ડાબી બાજુએ કી-સિમ્બોલ (ચાવી જેવું ચિહ્ન) પ્રદર્શિત થશે.



આકૃતિ 10.7 : કમ્પોઝિટ પ્રાઈમરી કી વ્યાખ્યાયિત કરવી

OrderPayment અને OrderShipment જેવાં કેટલાંક વધુ ટેબલની ડિઝાઈન પણ બનાવવી પડશે. આકૃતિ 10.8માં આ બન્ને ટેબલનું માળખું દર્શાવેલું છે.

OrderShipment	
To record shipping address of a customer who has placed order online	
Field Name	Data type
OrderID	Integer
ShippingAddress1	Text [VARCHAR_IGNORECASE]
ShippingAddress2	Text [VARCHAR_IGNORECASE]
Pincode	Text fix [CHAR]
ShippingDate	Date
ShippingStatus	Boolean

OrderPayment	
To store details of payment made by customer	
Field Name	Data type
OrderID	Integer
PaymentID	Integer
PaymentDate	Date
PaymentAmount	Decimal
PaymentMethod	Text [VARCHAR_IGNORECASE]
ChequeDDNumber	Text [VARCHAR_IGNORECASE]
BankName	Text [VARCHAR_IGNORECASE]
BankBranch	Text [VARCHAR_IGNORECASE]

આકૃતિ 10.8 : OrderPayment અને OrderShipment ટેબલ

Ordershipment ટેબલમાં સરનામાનો સમાવેશ કરેલ છે, જ્યાં માલનું વિતરણ કરવામાં આવશે. ShippingStatus ફિલ્ડનો ઉપયોગ માલ પહોંચાડવાની સ્થિતિ જાળવવા માટે થાય છે. શરૂઆતમાં તેની કિંમત 'N' સેટ કરવામાં આવશે. પાછળથી જ્યારે ગ્રાહક માલ મેળવે છે, ત્યારે તેને 'Y' વડે સેટ કરવું જરૂરી બને છે.

OrderPayment ટેબલ ગ્રાહક પાસેથી મેળવેલ ચુકવણીની વિગતો ધરાવશે. ગ્રાહકને હપતાથી ચુકવણીનો વિકલ્પ પણ આપી શકાય અને આથી OrderPayment ટેબલમાં OrderID અને PaymentIDને કમ્પોઝિટ પ્રાઈમરી કી તરીકે નક્કી કરવામાં આવે છે. ઉદાહરણ તરીકે, OrderID '1' માટે PaymentID 1, 2, 3, 4 વગેરે હોઈ શકે. OrderID '2' માટે PaymentID ફરી 1થી શરૂ થશે. ચુકવણીની પદ્ધતિ રોકડ, ચેક, ક્રેડિટ અથવા ડેબિટ કાર્ડ હોઈ શકે. PaymentMethod ફિલ્ડ આ ચારમાંથી એક વિકલ્પ ધરાવશે. જો ચુકવણી ચેક વડે કરવામાં આવે તો OrderPayment ટેબલમાં ચેકનંબર અને ચેક અદા કરનાર બેંકના નામનો પણ સંગ્રહ કરવામાં આવશે.


હવે તમે ખાતરી કરો કે જરૂરી ફેરફારો સાથેનાં Supplier, Employee, Customer, Product, Country, State, City, ProductCategory, Order, OrderDetail, OrderShipment ટેબલ તમારા ડેટાબેઝમાં છે. જો આ બધાં ટેબલ ન હોય, તો વિઝાર્ડ વડે અથવા ડિઝાઈન વ્યૂમાં તે બનાવો.

બેઝમાં ટેબલ વચ્ચે સંબંધ બનાવવો (Creating Relationships between Tables in Base)

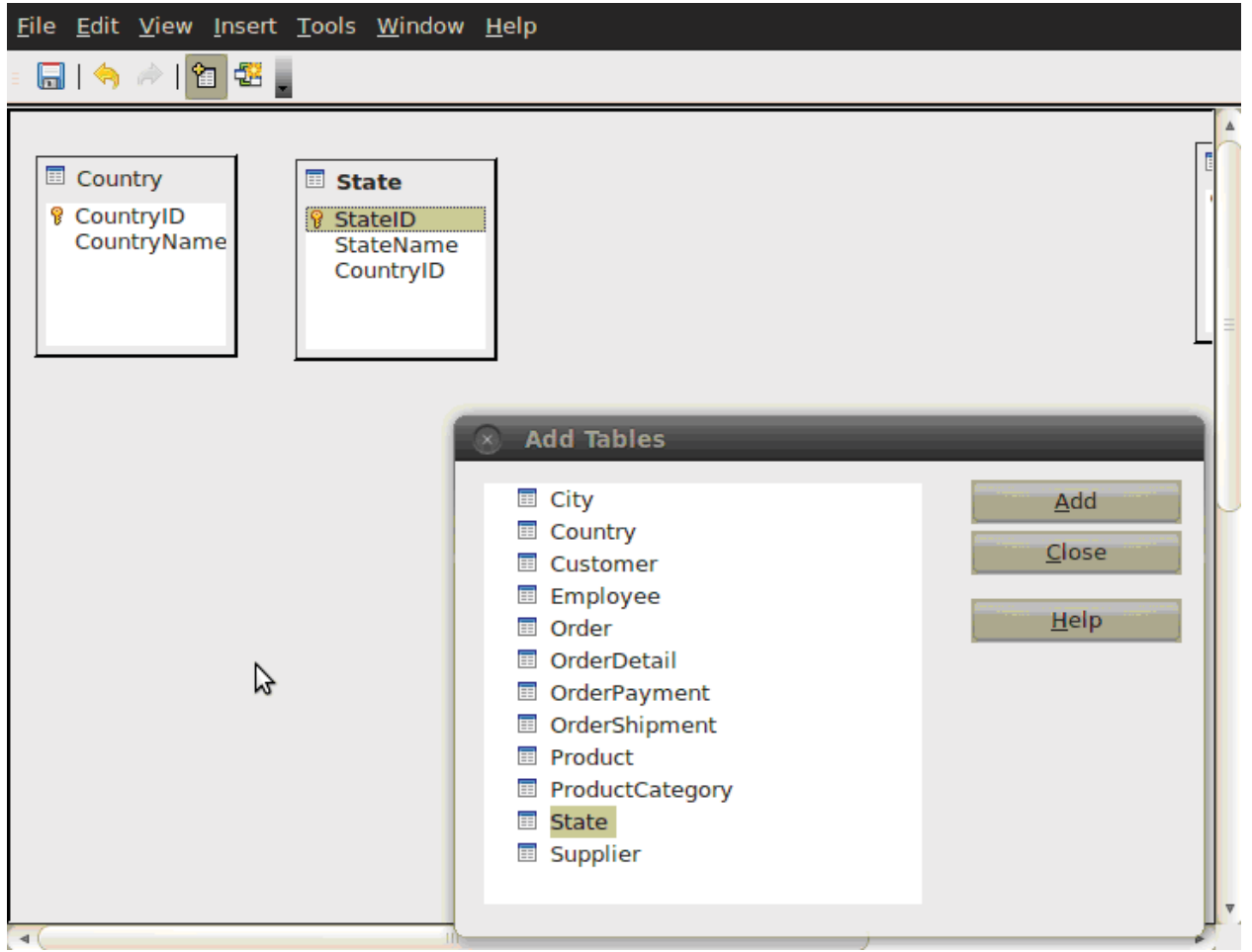
અત્યાર સુધીમાં આપણે ચર્ચા કરેલ ટેબલ તમે ડિઝાઈન કરી લીધાં હશે. ધારો કે ગોરેગાંવ, મુંબઈ, મહારાષ્ટ્રના નવા ગ્રાહકનો રેકોર્ડ Customer ટેબલમાં ઉમેરવો છે. હજી સુધી State ટેબલમાં Maharashtra ઉમેરેલ નથી અને City ટેબલમાં Mumbai સિટીનો Goregaon વિસ્તાર ઉમેરેલો નથી. પીનકોડ નંબર 452001 સાથે ગ્રાહકનો રેકોર્ડ ઉમેરો. અહીં મહારાષ્ટ્ર રાજ્યના મુંબઈ સિટીના ગોરેગાંવ વિસ્તારના પીનકોડની કિંમત 452001 છે. તમે ઉમેરેલા રેકોર્ડનો બેઝ સ્વીકાર કરશે. પણ તમે લોજિકલી (તાર્કિક રીતે) વિચારો. આ રેકોર્ડ ઉમેરવાની પરવાનગી આપવી જોઈએ ? ના. આપણે Customer ટેબલમાં આ પ્રકારના રેકોર્ડ ઉમેરવા માટે પરવાનગી ન આપવી જોઈએ. આનું કારણ ખૂબ જ સહેલું છે; આ એન્ટ્રીને સંબંધિત City ટેબલમાં કોઈ અનુરૂપ રેકોર્ડ નથી. આવા પ્રતિબંધ લાગુ પાડવા માટે આપણે બેઝમાં સંબંધ લાક્ષણિકતાનો ઉપયોગ કરવો જોઈએ. ટેબલ વચ્ચે સંબંધ સ્થાપિત કરવાથી સંદર્ભિત (Referenced) ફિલ્ડમાં વપરાશકર્તાને નકામો ડેટા એન્ટર કરતાં રોકશે.

ચાલો, આપણે બેઝનો ઉપયોગ કરીને બનાવેલાં ડેટાબેઝનાં વિવિધ ટેબલ વચ્ચે યોગ્ય સંબંધ સ્થાપવાનું શીખીએ.

- ટૂલ્સ મેનુમાંથી *Relationship...* વિકલ્પ ઉપર ક્લિક કરો.

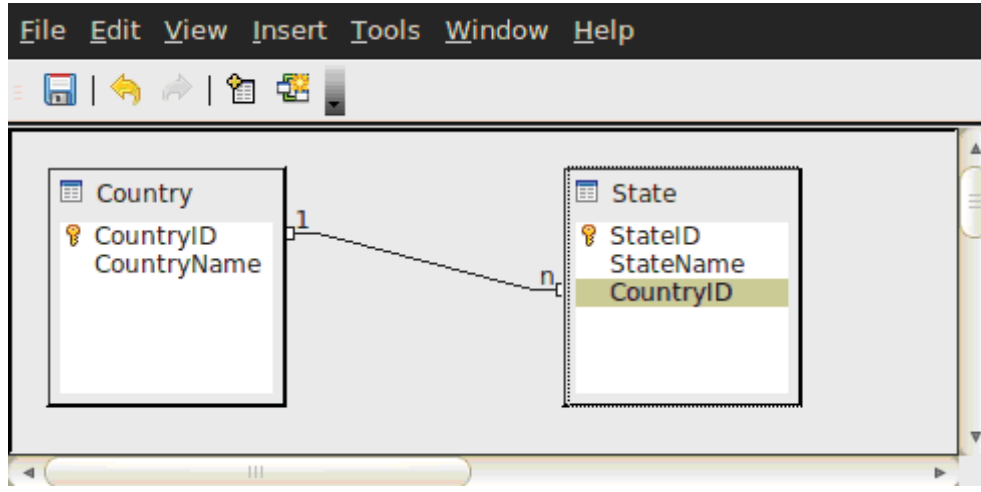
પછી *Insert* → *Add Tables*. પસંદ કરો. આના વિકલ્પ રૂપે, આકૃતિ 10.9માં દર્શાવ્યા પ્રમાણે તમે Add Tables બટન () ક્લિક કરી શકો. આથી Add Tables ડાયલોગ બોક્સ એક્ટિવ ડેટાબેઝનાં ટેબલની યાદી સાથે પ્રદર્શિત થશે.

- Country ટેબલ આઈકોન પસંદ કરો અને Add બટન ઉપર ક્લિક કરો. તમે જોઈ શકશો કે રિલેશનશિપ વિન્ડોની પશ્ચાદ્ભૂમિમાં પસંદ કરેલાં ટેબલનાં બધાં ફિલ્ડ પ્રદર્શિત થયાં છે.
- એ જ રીતે State ટેબલ પસંદ કરો. આકૃતિ 10.9માં દર્શાવ્યા પ્રમાણે રિલેશનશિપ વિન્ડો જોવા મળશે.



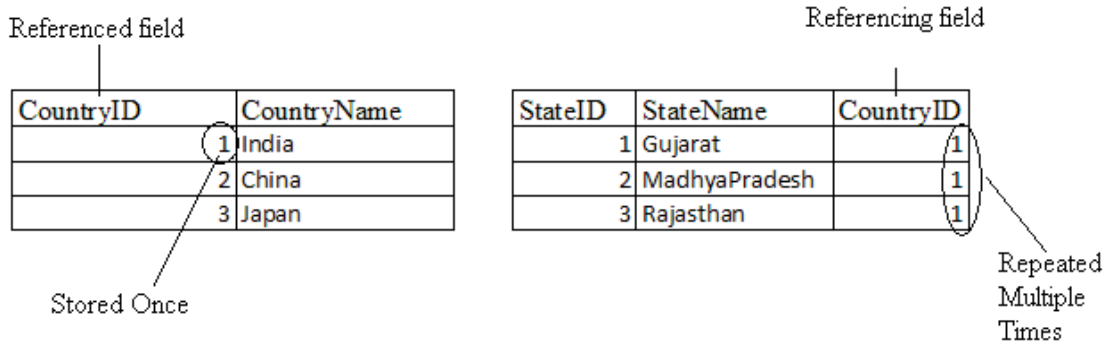
આકૃતિ 10.9 : રિલેશનશિપ ડાયલોગબોક્સ

- સંબંધ બનાવવા માટે માત્ર 'ડ્રેગ અને ડ્રોપ' કાર્ય કરો. country ટેબલના CountryID ફિલ્ડ (પ્રાઈમરી કી) ઉપર ક્લિક કરો.
- Country ટેબલના CountryID ફિલ્ડને ડ્રેગ કરો અને State ટેબલના CountryID ફિલ્ડ (ફોરેન કી) ઉપર છોડી દો.
- આકૃતિ 10.10માં દર્શાવ્યા પ્રમાણે 1 અને n લેબલ સાથેનાં બન્ને ફિલ્ડને જોડતી લાઈન પ્રદર્શિત થશે. અહીં નોંધ કરશો કે લેબલ ટેક્સ્ટ 1 પ્રાઈમરી કી બાજુએ અને લેબલ ટેક્સ્ટ n ફોરેન કી બાજુએ પ્રદર્શિત થયેલ છે. તે જણાવે છે કે CountryID ફિલ્ડ એક પ્રાઈમરી કી છે અને તેની કિંમતો અજોડ હોય છે અને પ્રાઈમરી કીમાં સંગ્રહિત દરેક અજોડ કિંમત ફોરેન કીમાં n વખત રીપીટ થઈ હોઈ શકે. આ જાતના સંબંધ કે જ્યાં રિફર્ડ ટેબલ (referred table)ની એક કિંમત રેફરન્સિંગ ટેબલની ઘણી કિંમતો સાથે સંકળાયેલી હોય તેને **એકથી અનેક સંબંધ (One-to-Many relationship)** કહેવામાં આવે છે. આકૃતિ 10.11માં એકથી અનેક સંબંધમાં સંગ્રહિત કેટલાક નમૂનાના રેકોર્ડ દર્શાવ્યા છે. સંબંધના પ્રકારો વિશે પણ હવે પછી આ પ્રકરણમાં ચર્ચા કરી છે.



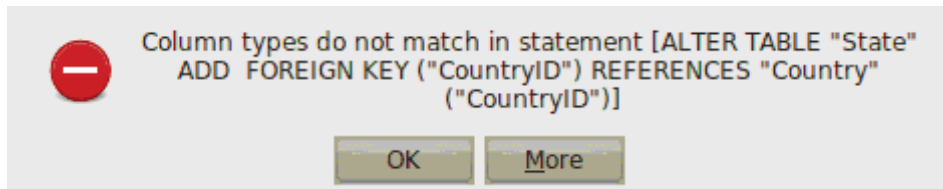
આકૃતિ 10.10 : Country અને State ટેબલ વચ્ચેનો એકથી એક સંબંધ

આકૃતિ 10.10માં Country અને State ટેબલ વચ્ચેનો સંબંધ દર્શાવ્યો છે. આકૃતિ 10.10માં બતાવેલાં બંને ટેબલ્સને તેનાં ટાઇટલબાર ઉપર ક્લિક કરી અને પછી ડ્રેગ કરીને કોઈ પણ ઈચ્છિત સ્થાન ઉપર લઈ જઈ શકાય. અહીં Country ટેબલનું CountryID ફિલ્ડ રેફરન્સ ફિલ્ડ (referenced field) તરીકે ઓળખાય છે, જ્યારે State ટેબલનું CountryID ફિલ્ડ રેફરન્સિંગ ફિલ્ડ (referencing field) તરીકે ઓળખાય છે. (જુઓ આકૃતિ 10.11)



આકૃતિ 10.11 : એકથી અનેક સંબંધ દર્શાવતા રેકોર્ડ

નોંધ : બે ટેબલ વચ્ચે સંબંધ બનાવવા માટે રેફરન્સિંગ ફિલ્ડ અને રેફરન્સ ફિલ્ડની ડેટાટાઇપ એકસમાન હોવી જોઈએ. બંને ફિલ્ડની ડેટાટાઇપ એકસમાન ન હોય એ કિસ્સામાં આકૃતિ 10.12માં દર્શાવ્યા પ્રમાણેની ભૂલ (error) સંબંધ બનાવતાં સમયે દર્શાવે છે.



આકૃતિ 10.12 : સંબંધ બનાવતાં સમયે કૉલમ ડેટાટાઇપ મિસમેચ એરર

સંદર્ભ સંપૂર્ણતા (રેફરેન્શલ ઇન્ટિગ્રિટી - Referential Integrity)

એક વખત બે ટેબલ વચ્ચે સંબંધ બની જાય પછી સંબંધ દર્શાવતી લાઇન ઉપર ડબલ ક્લિક કરો. આથી આકૃતિ 10.13માં દર્શાવ્યા પ્રમાણે Relations ડાયલોગબોક્સ પ્રદર્શિત થશે. અત્યાર સુધીમાં, આપણે એ વિધાન સાથે અત્યંત સ્પષ્ટ છીએ કે Country ટેબલ અને State ટેબલ એકબીજાં સાથે જોડાયેલાં છે. વપરાશકર્તા ફક્ત તે જ દેશની માહિતીના ડેટા

State ટેબલમાં દાખલ કરે કે જે Country ટેબલમાં પહેલાથી દાખલ કરેલ હોય. આકૃતિ 10.11માં જોઈ શકાય છે કે Country ટેબલમાં પહેલા India ઉમેરવામાં આવ્યું. તે પછી State ટેબલમાં Indianાં ત્રણ રાજ્યો ઉમેરવામાં આવ્યાં. Country ટેબલના Indianે સંબંધિત રેકોર્ડને માસ્ટર અથવા પેરન્ટ રેકોર્ડ (Master or parent record) ગણવામાં આવે છે. જ્યારે State ટેબલમાં India સંબંધિત રેકોર્ડને ટ્રાન્ઝેક્શન અથવા ચાઈલ્ડ રેકોર્ડ (transaction or child record) ગણવામાં આવે છે.

પણ હવે જો વપરાશકર્તા Country ટેબલમાંથી પેરન્ટ રેકોર્ડ કાઢી નાખે તો શું થાય ? અન્ય ટેબલમાં તેને સંબંધિત રેકોર્ડનું શું થાય ? ઉદાહરણ તરીકે, ધારો કે આપણે State ટેબલમાં 10 રેકોર્ડ ઉમેરેલાં છે જેની CountryIDની કિંમત 1 છે. State ટેબલના CountryIDની કિંમત 1 હોય એવા Country ટેબલના રેકોર્ડનું શું કરવું જોઈએ, જો વપરાશકર્તા માસ્ટર ટેબલનાં એ રેકોર્ડને ડિલીટ અથવા અપડેટ કરવાનો નિર્ણય લે તો ? State ટેબલમાં એક પણ એન્ટ્રી એવી ન હોવી જોઈએ કે જેને સંબંધિત રેકોર્ડ માસ્ટર ટેબલમાં ન હોય. આ ખ્યાલને રેફરેન્શલ ઇન્ટિગ્રિટી (Referential Integrity) કહેવામાં આવે છે. રેફરેન્શલ ઇન્ટિગ્રિટી સિદ્ધાન્તને એ રીતે જણાવી શકાય કે કોઈ પણ મેળ ન ખાતી (unmatched) ફોરેન કીનું અસ્તિત્વ ડેટાબેઝમાં ન હોવું જોઈએ.

ડેટાબેઝ ડિઝાઈનર કંપનીના ટ્રાન્ઝેક્શન (વ્યવહાર)ની જરૂરિયાત ઉપર આધાર રાખીને આકૃતિ 10.13માં દર્શાવ્યા પ્રમાણે ચારમાંથી એક વિકલ્પ પસંદ કરીને સેટ કરે છે. આ વિકલ્પો અપડેટ (સુધારા-વધારા) અથવા ડિલીટ (રદ કરવા) ના કાર્ય સમયે ડેટાબેઝમાં રેફરેન્શલ ઇન્ટિગ્રિટી જાળવવાની આપણે પરવાનગી આપે છે.

State	Country
CountryID	CountryID

Update options

- No action
- Update cascade
- Set null
- Set default

Delete options

- No action
- Delete cascade
- Set null
- Set default

OK Cancel Help

આકૃતિ 10.13 : રેફરેન્શલ ઇન્ટિગ્રિટીનું પાલન કરાવવું (Enforcing referential integrity)

No action : આ વિકલ્પ દર્શાવે છે કે ઉપયોગકર્તાને કોઈ રેકોર્ડ ભૂંસવાની કે અદ્યતન બનાવવાની મંજૂરી ન આપવી જોઈએ જો અન્ય કોઈ ટેબલમાં તેના સંબંધિત રેકોર્ડ અસ્તિત્વમાં હોય. આ No Action વિકલ્પ પસંદ કરી Indianી માહિતીવાળો રેકોર્ડ દૂર કરવાનો પ્રયત્ન કરો. બેઝ પહેલાં ખાતરી કરશે કે તમે આ રેકોર્ડ દૂર કરવા માંગો છો કે નહીં અને આકૃતિ 10.14માં બતાવ્યા પ્રમાણે એક સંદેશો પ્રદર્શિત કરશે.

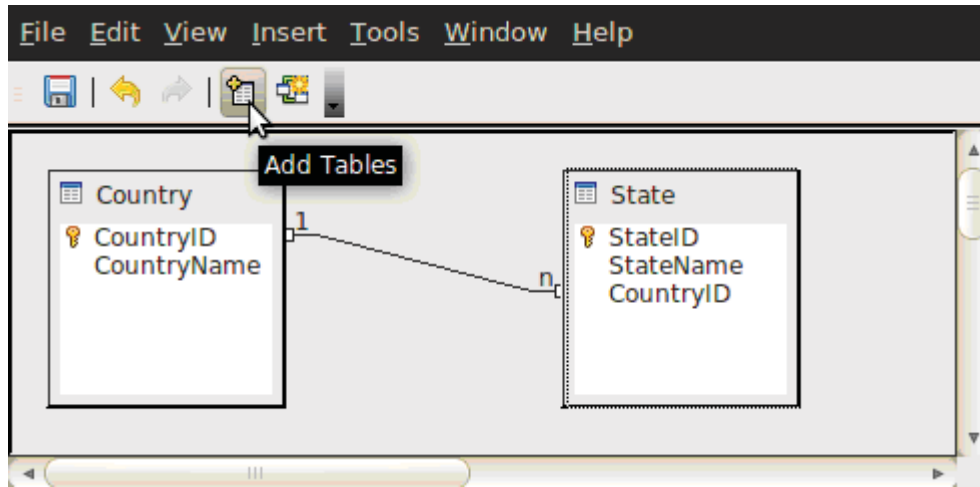


આકૃતિ 10.14 : No Action વિકલ્પ વાપરીને રેફરેન્સલ ઇન્ટિગ્રિટીનું પાલન કરાવવું

Update cascade : આ વિકલ્પ દર્શાવે છે કે જો ઉપયોગકર્તા રેફરન્સ કિંમતો દૂર કરવાની કે અદ્યતન કરવાની મંજૂરી આપે, તો તેને સંબંધિત તમામ રેકોર્ડ આપમેળે દૂર કે અદ્યતન થઈ જવા જોઈએ.

Set null : આ વિકલ્પ દર્શાવે છે કે જો ઉપયોગકર્તા રેફરન્સ ફિલ્ડ દૂર કે અદ્યતન કરે, તો તેને સંબંધિત તમામ રેકોર્ડનાં સંબંધિત ફિલ્ડની નલવેલ્યુ કરવામાં આવે.

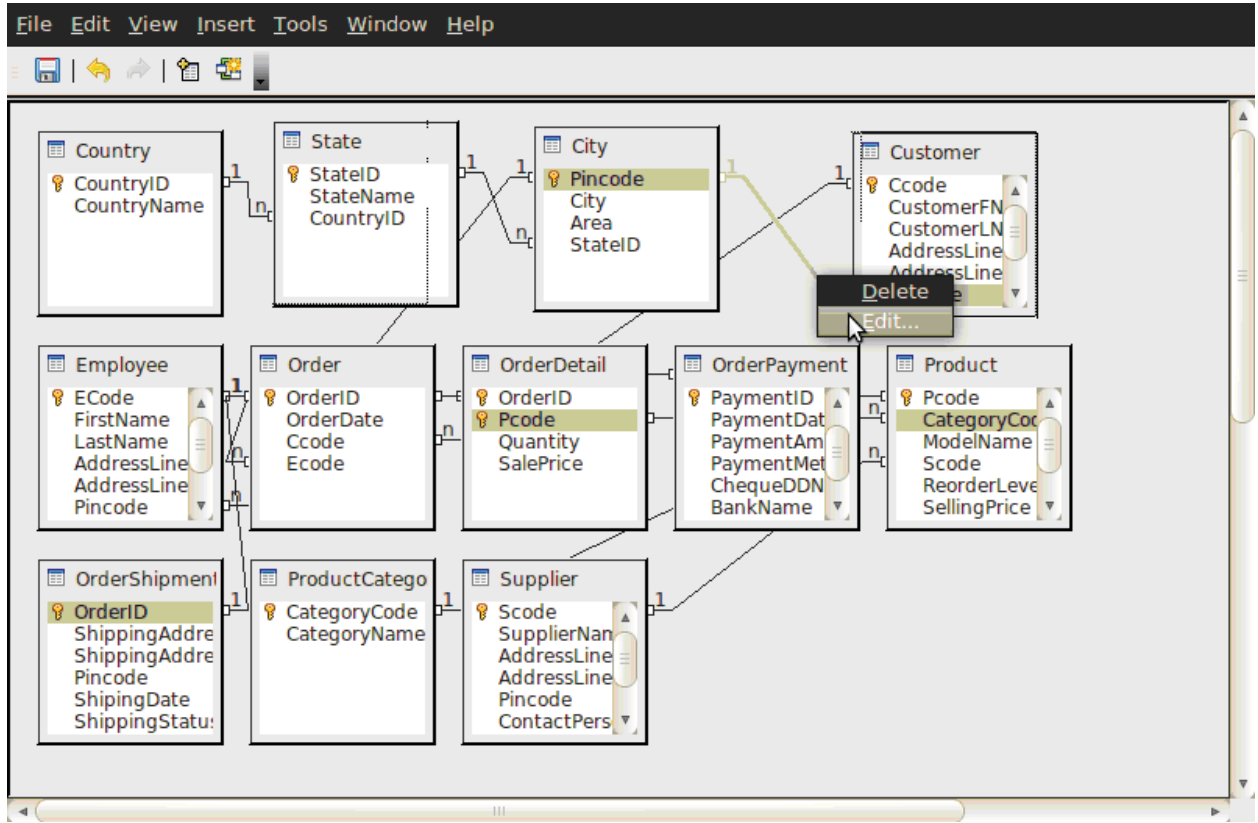
આ જ રીતે તમે મોડર્ન ઇલેક્ટ્રોનિક સ્ટોરમાં અન્ય ટેબલ વચ્ચે રિલેશનશિપ બનાવી શકો. આકૃતિ 10.15માં દર્શાવ્યા પ્રમાણે તમે ટૂલબાર પરના *Add Tables* આઈકોન ઉપર ક્લિક કરીને અન્ય ટેબલ ઉમેરી શકો છો. આથી અગાઉ દર્શાવેલી આકૃતિ 10.9 પ્રમાણે એડ ટેબલ ડાયલોગબોક્સ પ્રદર્શિત થશે.



આકૃતિ 10.15 : રિલેશનશિપ વિન્ડોમાં ટેબલ ઉમેરવાં

આપણા નમૂનાના ડેટાબેઝમાં એક વખત બધાં ટેબલ વચ્ચે સંબંધ બનાવ્યા પછી રિલેશનશિપ વિન્ડો આકૃતિ 10.16માં બતાવ્યા જેવી લાગશે.

હવે આપણે હાલમાં બનાવેલાં સંબંધનો સંગ્રહ કરવો જરૂરી છે, આથી આપણે ટૂલબાર ઉપરનાં Save બટન ઉપર ક્લિક કરી સંગ્રહ કરો. જો આપણે સંગ્રહ કરવાનું ભૂલી જઈએ, તો આપણને યાદ કરાવવા માટે સ્ક્રીન ઉપર એક ચેતવણી-સંદેશ બતાવવામાં આવશે. અમુક સમયે, સંગ્રહ કર્યા પછી પણ સંબંધમાં ફેરફાર ડેટાબેઝમાં પ્રતિબિંબિત થતા નથી. જો આવું થાય તો ડેટાબેઝને બંધ કરો અને ફરી તેને ખોલો. તમે જોશો કે સંબંધમાં ફેરફારો અમલમાં મુકાયેલા છે અને તે જોઈ શકાય છે.



આકૃતિ 10.16 : ડેટાબેઝમાં તમામ ટેબલનો સંબંધ

રિલેશનશિપમાં સુધારા કરવા (એડિટિંગ રિલેશનશિપ્સ - Editing Relationships)

આપણે કોઈ પણ બે ટેબલ વચ્ચે સ્થાપેલાં સંબંધને દૂર કે સુધારી શકીએ છીએ. આ માટે આપણે ફક્ત રિલેશનશિપ સ્કીન ખોલીને બે ટેબલ વચ્ચેની ઈચ્છિત સંબંધ રેખા ઉપર ક્લિક કરવાનું છે. આમ કરવાથી, તે લાઈન જાડી થઈ જશે. તે પછી જો આપણે રાઈટ ક્લિક કરીશું, તો આકૃતિ 10.16માં બતાવ્યા પ્રમાણે એક પોપઅપ મેનુ જોવા મળશે.

અહીં, આપણી જરૂરિયાત પ્રમાણે આપણે જરૂરી વિકલ્પ પસંદ કરીશું. જો આપણે પોપઅપ મેનુમાંથી Edit વિકલ્પ પસંદ કરીશું, તો રિલેશનશિપ ડાયલોગબોક્સમાં બતાવ્યા પ્રમાણે આપણી જરૂરિયાત પ્રમાણે આપણે સંબંધના એટ્રીબ્યુટ્સને એડિટ કરી શકીશું. આના સિવાય, આપણે ડિલીટ વિકલ્પ પસંદ કરીને સંબંધ કાઢી નાખીએ. હવે આપણે ડેટાબેઝનાં બે ટેબલ વચ્ચેનાં વિવિધ પ્રકારનાં સંબંધ બનાવવા, સુધારવા કે દૂર કરવા તે જાણીએ છીએ. તમામ ટેબલ વચ્ચે સંબંધ બનાવો, જો હજી સુધી ન બનાવ્યો હોય તો,

ચાલો, હવે આપણે વિવિધ પ્રકારના સંબંધ બાબત ઊંડાણપૂર્વક ચર્ચા કરીએ. સૈદ્ધાંતિક રીતે ડેટાબેઝનાં ટેબલ વચ્ચે ત્રણ મુખ્ય પ્રકારની રિલેશનશિપ હોઈ શકે :

- એકથી એક (one to one)
- એકથી અનેક (one to many)
- અનેકથી અનેક (Many to many)

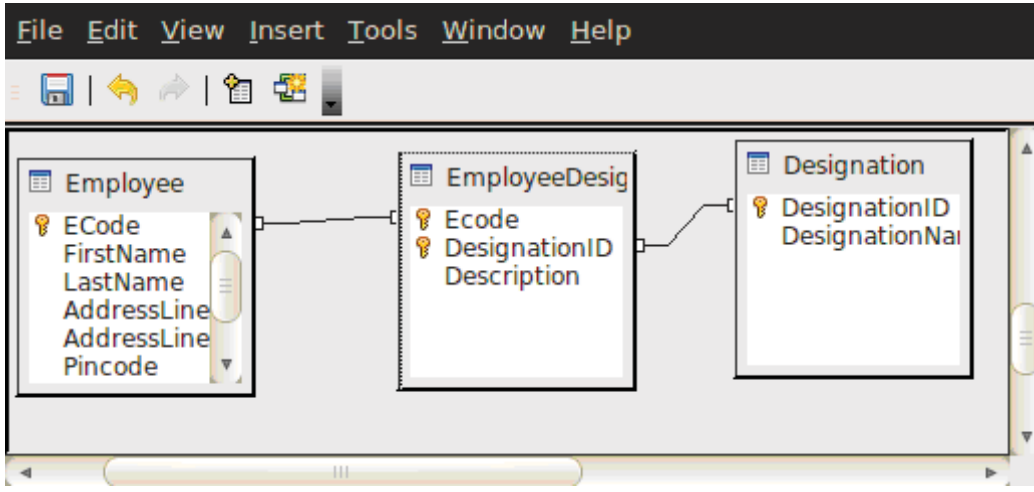
બે ટેબલના ડેટા વચ્ચે કયા પ્રકારના સંબંધ વ્યાખ્યાયિત કરવાં તેનો આધાર પહેલા ટેબલને અનુરૂપ બીજા ટેબલમાં લાગતાવળગતા (તેને અનુલક્ષીને) કેટલા રેકોર્ડ છે. ચાલો, હવે તે દરેકને પદ્ધતિસર સમજીએ.

એકથી એક સંબંધ : તેનું નામ સૂચવે છે તે પ્રમાણે એકથી એક સંબંધ દર્શાવે છે કે એક ચોક્કસ ટેબલના કોઈ ચોક્કસ રેકોર્ડને ડેટાબેઝના અન્ય ટેબલમાં એક અને ફક્ત એક જ અનુલક્ષીને રેકોર્ડ ધરાવતો હશે. આશ્ચર્ય પામશો નહીં. કારણકે આપણા માટે તે નવું નથી અને આપણા રોજિંદા જીવનમાં ઉદાહરણો ઘણાં સામાન્ય છે. કોઈ થિયેટર વિશે વિચારો. એક ખુરશીમાં કેટલા લોકો બેસી શકે ? સ્વાભાવિક રીતે, એક. હવે એક વ્યક્તિ કેટલી ખુરશીનો કબજો મેળવી શકે ? ચોક્કસપણે, એક. આથી, આપણે કહી શકીએ કે વ્યક્તિ અને ખુરશી વચ્ચેનો સંબંધ એ એકથી એક સંબંધ છે. તમારા વર્ગ અને વર્ગશિક્ષક અથવા ગ્રંથાલય અને વિદ્યાર્થીને આપવામાં આવેલા ગ્રંથાલય કાર્ડ વચ્ચેની રિલેશનશિપ વિશે તમે વિચારી શકો છો. તમે ચોક્કસપણે સંમત થશો કે તમારી સ્કૂલના કોઈ પણ વર્ગ માટે એક અને ફક્ત એક જ વર્ગશિક્ષક હશે. અને આ જ સમયે કોઈ પણ શિક્ષકને માત્ર એક જ વર્ગનું ધ્યાન રાખવા માટે સોંપાયેલ હોય છે. આથી, આપણે કહી શકીએ કે વર્ગ અને વર્ગશિક્ષક એ એકથી એક સંબંધનું એક ઉદાહરણ છે.

આપણા નમૂનાના ડેટાબેઝમાં બનાવેલ સંબંધોનું નિરીક્ષણ કરો. Order ટેબલનું OrderID ફિલ્ડ એ OrderShipment ટેબલના OrderID ફિલ્ડ સાથે સંબંધિત છે. હવે દરેક ઓર્ડર, ઉદાહરણ તરીકે 1 માટે ફક્ત એક રેકોર્ડ OrderShipment ટેબલમાં મળશે. તે દર્શાવે છે કે Order અને OrderShipment ટેબલ વચ્ચે એકથી એક સંબંધ છે. તમે એમ પણ વિચારો કે તો પછી OrderShipment નાં ફિલ્ડ Order ટેબલમાં શા માટે ઉમેર્યાં નથી ? ModernElectronicDatabaseના કિસ્સામાં દરેક ઓર્ડર માટે OrderShipmentની માહિતી બનાવવાની નથી. OrderShipmentની માહિતી માત્ર ત્યારે જ નોંધવાની છે જ્યારે ગ્રાહક ઓનલાઇન ઓર્ડર નોંધાવે અથવા જ્યારે તે જાતે સ્ટોરમાં આવીને ખરીદી કરે અને પોતાને ઘરે અથવા કોઈ અન્ય સરનામા ઉપર માલના વિતરણની અપેક્ષા રાખે. આથી, જો Order ટેબલનું વિઘટન ન કરવામાં આવે, તો તમામ OrderShipment ફિલ્ડમાં નલ કિંમત આવી જશે. આથી, આવા કિસ્સામાં ટેબલનું વિઘટન કરીને એકથીએક સંબંધ જાળવવો સારો છે.

એકથી અનેક સંબંધ : ટેબલ વચ્ચે સૌથી સામાન્ય સંબંધ એ એકથી અનેક સંબંધ છે. આ પ્રકારના સંબંધમાં કોઈ અમુક ટેબલનો કોઈ એક ચોક્કસ રેકોર્ડ ડેટાબેઝના સંબંધિત ટેબલના અનેક રેકોર્ડ સાથે સંકળાયેલાં હોઈ શકે. એક ગ્રાહક અનેક વસ્તુઓ ખરીદી શકે. એ જ પ્રમાણે, એક શહેર એક રાજ્યમાં હોય છે, પરંતુ એક રાજ્યમાં અનેક શહેરો હોય છે. તમારી સ્કૂલના વર્ગશિક્ષકનું દૃશ્ય વિચારો. કોઈ પણ વર્ગ માટે માત્ર એક જ વર્ગશિક્ષક હોઈ શકે. પણ કોઈ વર્ગશિક્ષક સાથે અનેક વિદ્યાર્થીઓ નોંધાયેલા હોઈ શકે. આથી, વર્ગશિક્ષક અને વિદ્યાર્થી વચ્ચેનો સંબંધ એકથી અનેકનો છે.

અનેકથી અનેક સંબંધ : કેટલાંક ખાસ રેકોર્ડના કિસ્સામાં કે જેમાં બંને ટેબલમાં એક કરતાં વધુ વખત તેનો સંગ્રહ કરવો પડે છે, ત્યારે અનેકથી અનેક સંબંધ બને છે. આપણા ડેટાબેઝના ઉદાહરણમાં, આપણે કર્મચારીઓની માહિતી રાખીએ છીએ. હજી સુધી આપણે ધારીએ છીએ કે કર્મચારી કોઈ એક હોદ્દો સંભાળે. અમુક સમયે સંસ્થામાં કોઈ કર્મચારી કોઈ એક કરતાં વધારે હોદ્દાઓ પણ સંભાળે. આ રીતે તે કિસ્સામાં આપણે કહી શકીએ કે એક કર્મચારીના અનેક હોદ્દાઓ હોઈ શકે અને એક હોદ્દા ઉપર અનેક કર્મચારીઓ હોઈ શકે. આ કિસ્સામાં કર્મચારીઓના હોદ્દાઓનું એક અલગ ટેબલ રાખવું પડે. આ રીતે બે એન્ટિટી વચ્ચે અનેકથી અનેક સંબંધ હોય છે. આકૃતિ 10.17માં નમૂનાની અનેકથી અનેક સંબંધની પરિસ્થિતિ બતાવેલ છે.



આકૃતિ 10.17 : Employee અને Designation વચ્ચે અનેકથી અનેકનો સંબંધ

તમે આકૃતિ 10.17માં જોઈ શકશો કે આપણી પાસે Employee અને Designation ટેબલ વચ્ચે કોઈ સીધો સંબંધ નથી. આનું કારણ એ છે કે ડેટાબેઝમાં અનેકથી અનેકનો સંબંધનું એક તૃતીય ટેબલ બનાવીને બે એકથી અનેક સંબંધમાં વિભાજન કરવામાં આવે છે. આપણા ઉદાહરણમાં EmployeeDesignation નામનું નવું ટેબલ બનાવ્યું કે જેની Employee ટેબલ અને Designation ટેબલ બન્ને સાથે એકથી અનેક સંબંધ છે. આમ, Employee ટેબલ અને Designation ટેબલ વચ્ચે જંકશન-ટેબલ તરીકે કામ આપે છે.

અત્યાર સુધીમાં આપણે ટેબલની રચના કરી, પ્રાઈમરી કી ફિલ્ડ અને ફોરેન કી ફિલ્ડ નક્કી કર્યા અને ટેબલનો એકબીજા સાથે સંબંધ નક્કી કર્યો. હવે, ચાલો આપણે ફિલ્ડની પ્રોપર્ટી બાબત ચર્ચા કરીએ.

ફિલ્ડ પ્રોપર્ટી (Field Properties)

અગાઉના પ્રકરણમાં ચર્ચા કરી તે પ્રમાણે ડેટા સાથે કામ કરતાં પહેલાં આપણે ટેબલને સારી રીતે વ્યાખ્યાયિત કરવાં પડે (ટેબલનું સ્ટ્રક્ચર બનાવવું પડે). કોઈ ટેબલના ફિલ્ડને વ્યાખ્યાયિત કરવા માટે આપણે નીચે જણાવેલાં પગલાં પ્રમાણે કાર્ય કરો :

- ફિલ્ડનું નામ (Field Name) સ્પષ્ટ કરો.
- ફિલ્ડ માટે યોગ્ય ડેટાટાઈપ (Datatype) પસંદ કરો.
- ફિલ્ડનું વર્ણન (Description) લખો. (વૈકલ્પિક)

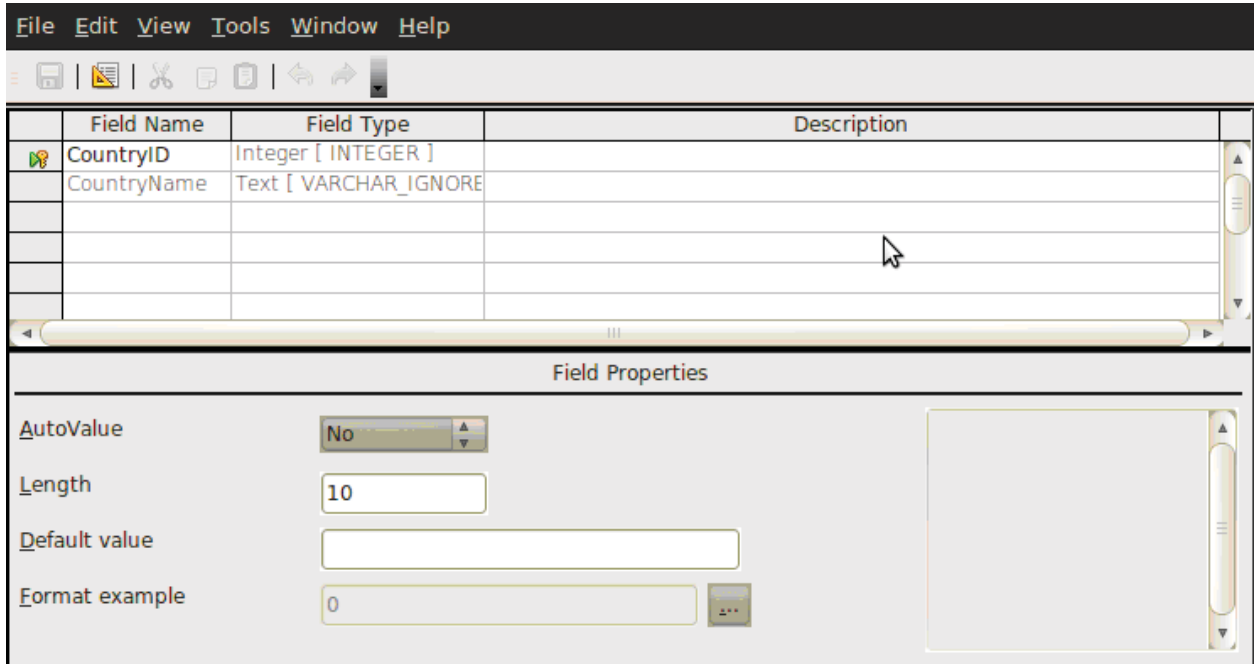
ઈન્ફોર્મેશન ટેકનોલોજીના ક્ષેત્રમાં, ડેટા એ કાર્યો માલ (Row material) છે, જે તૈયાર માલ (finished goods)નું ઉત્પાદન કરવામાં વપરાય છે. આ તૈયાર માલ ઈન્ફોર્મેશન તરીકે ઓળખાય છે. એ એક વિશ્વવ્યાપી સત્ય છે કે ‘સારું આપ્યા વિના આપણે શ્રેષ્ઠ મેળવવાની અપેક્ષા રાખી ના શકીએ.’ અન્ય શબ્દોમાં, આપણે કહી શકીએ કે આઉટપુટની ગુણવત્તા દેખીતી રીતે ઈનપુટની ગુણવત્તા ઉપર આધાર રાખે છે. જો આપણે ડેટાએન્ટ્રી સમયે સંગ્રહ કરવામાં આવતી કિંમતો ઉપર નિયંત્રણ અને પ્રતિબંધ કરી શકીએ, તો ઈનપુટ ડેટાને નિયંત્રિત કરવાની સારી પ્રથા (પ્રેક્ટિસ) કહેવાય કે જે સાચી ઈન્ફોર્મેશન બનાવી શકે.

જો વપરાશકર્તા Customer ટેબલનું Name ફિલ્ડ ખાલી છોડી દે તો ? ઘણી વખત સારો અનુભવી ડેટાએન્ટ્રી ઓપરેટર પણ ભૂલથી ખોટો ડેટા દાખલ કરે છે. આ ભૂલોને રોકવા માટે, બેઝ ‘ફિલ્ડલેવલ વેલિડેશન’ (Filed level valiadation) નામની વિશિષ્ટ સગવડતા પૂરી પાડે છે. તે બીજું કંઈ નહીં પણ દરેક ફિલ્ડમાં દાખલ કરવાના ડેટા માટેના અમુક શક્ય નિયમો અને ચકાસણી-સ્થળ છે. તે ચોક્કસ ફિલ્ડમાં અમુક કિંમતો દાખલ કરવાને પ્રતિબંધિત કરે છે. ડેટા-વેલિડેશન એ

ડેટા-પ્રોસેસિંગ (માહિતી-પ્રક્રિયા)નું એક અગત્યનું પાસું છે. તમારે નોંધ કરવી જોઈએ કે વિવિધ ટેબલમાં દાખલ કરેલો ડેટા સાચો અને ચોક્કસ હશે તો જ ફક્ત તે ડેટાબેઝ તમારી સંસ્થા માટે મૂલ્યવાન સંપત્તિ બની રહેશે.

બેઝ ખૂબ જ મહત્વપૂર્ણ અને રસપ્રદ લક્ષણ ધરાવે છે, જેને ફિલ્ડ પ્રોપર્ટી (field properties) કહેવામાં આવે છે. ફિલ્ડ પ્રોપર્ટી નક્કી કરે છે કે ફિલ્ડમાં કિંમતોનો સંગ્રહ અને પ્રદર્શિત કેવી રીતે થાય. દરેક પ્રકારના ફિલ્ડની અમુક ચોક્કસ પ્રોપર્ટીનો સેટ હોય છે. ઉદાહરણ તરીકે, OrderDate ફિલ્ડમાં વપરાશકર્તા ક્યા ફોર્મેટમાં ડેટા (dd-mm-yy અથવા mm-dd-yy) દાખલ કરશે, ડેટા પ્રિન્ટ કરતાં સમયે કે વ્યૂ (જોવામાં) કરતાં સમયે કઈ રીતે પ્રદર્શિત થશે અને તારીખના dd-mm-yy ફોર્મેટમાં વપરાશકર્તા 12/14/12 જેવી અસામાન્ય તારીખ દાખલ કરે.

ટેબલનું માળખું (સ્ટ્રક્ચર) ડિઝાઇન કરતાં સમયે ફિલ્ડની પ્રોપર્ટી નક્કી કરવામાં આવે છે. ટેબલ ડિઝાઇન વિન્ડોમાં ફિલ્ડ પસંદ કરો, આથી આકૃતિ 10.18માં દર્શાવ્યા પ્રમાણે બેઝ પસંદ કરેલી ડેટાટાઇપને અનુરૂપ યોગ્ય ડિકોલ્ટ (પૂર્વ- નિર્ધારિત) કિંમતો સાથે ફિલ્ડ પ્રોપર્ટી બોક્સ પૂરું પાડશે.



આકૃતિ 10.18 : ફિલ્ડ પ્રોપર્ટી ડાયલોગબોક્સ

કેટલીક ફિલ્ડ પ્રોપર્ટી ટેબલ માટે ફિલ્ડ ક્ષાએ વેલિડેશન પૂરું પાડે છે. સ્ક્રીન ઉપર પ્રદર્શિત ફિલ્ડ પ્રોપર્ટી આપણે પસંદ કરેલા ફિલ્ડના ડેટાટાઇપ પ્રમાણે બદલાય છે. ચાલો, આપણે કેટલીક સામાન્ય ફિલ્ડ પ્રોપર્ટીની ઊંડાણપૂર્વક ચર્ચા કરીએ:

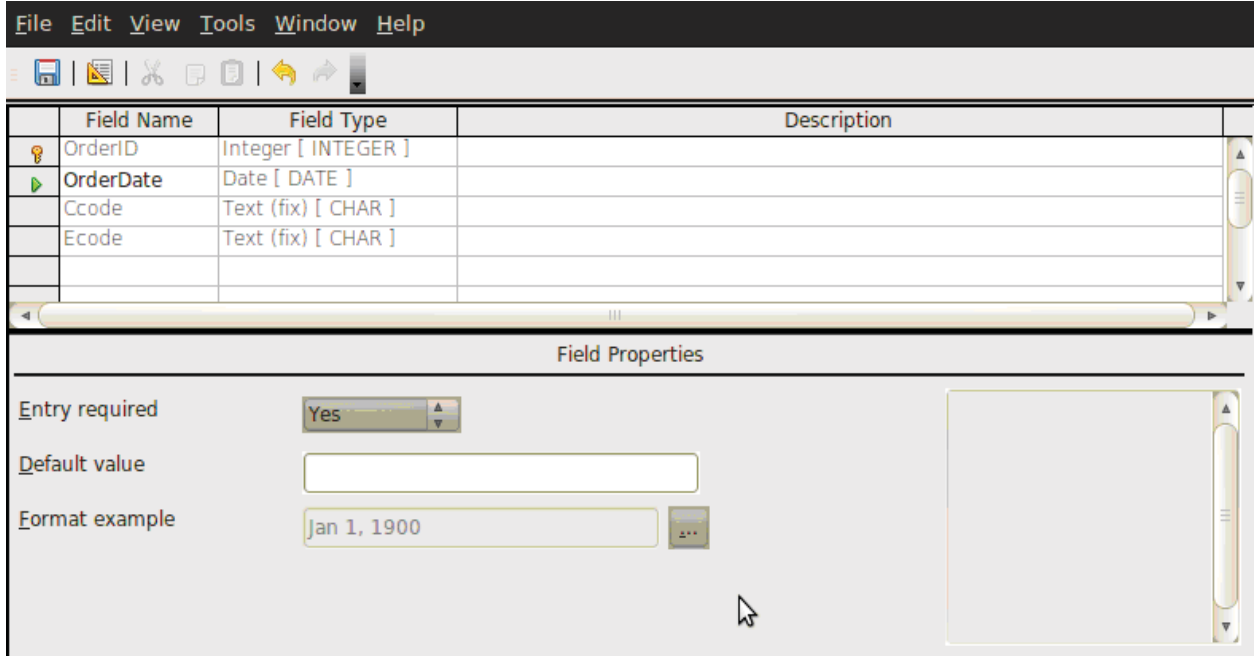
ઑટોવેલ્યુ (AutoValue) : આકૃતિ 10.18માં જોઈ શકાય છે કે, *AutoValue* પ્રોપર્ટી પાછળ એક ડ્રોપડાઉન બોક્સ (કોમ્બોબોક્સ) છે. આ ડ્રોપડાઉન બોક્સ ઉપર ક્લિક કરવાથી, તમે તેમાં 'Yes' અને 'No' બે કિંમતો તેમાં જોઈ શકશો. જો ન્યુમરિક ફિલ્ડ માટે ઑટોવેલ્યુની કિંમત 'Yes' સેટ કરવામાં આવશે, તો દરેક નવા રેકોર્ડ માટે બેઝ આપોઆપ કિંમત વધારશે. આ ફિલ્ડ પ્રોપર્ટી ખાસ કરીને ન્યુમરિક ફિલ્ડ માટે વપરાય છે કે જેને પ્રાર્થમરી કી બનાવી છે. ઉદાહરણ તરીકે, Country ટેબલમાં CountryID ફિલ્ડ છે, જે પ્રાર્થમરી કી છે. CountryID ફિલ્ડની ડેટાટાઇપ *integer* છે. આ ફિલ્ડની અપેક્ષિત કિંમત 1, 2, 3, વગેરે છે. આ કિસ્સામાં ઑટોવેલ્યુને 'Yes'થી સેટ કરી શકાય.

એન્ટ્રી રિક્વાયર્ડ (Entry required) : ડેટાબેઝનાં તમામ ફિલ્ડમાંથી કેટલાક અગત્યનાં ફિલ્ડ હોઈ શકે કે જે ખાલી

છોડી ના શકાય. હવે એવું બને કે ડેટાએન્ટ્રી ઓપરેટર અગત્યનાં ફિલ્ડમાં જરૂરી કિંમત દાખલ કરવાનું ભૂલી જાય. પણ ડેટાએન્ટ્રી સમયે જ જો આવા અગત્યનાં ફિલ્ડને છોડી દેવાની પરવાનગી વપરાશકર્તાને ન આપીએ તો ? આપણે આ જરૂરિયાત ફિલ્ડપ્રોપર્ટીમાં *Entry Required* ને સેટ કરી પ્રાપ્ત કરી શકીએ. આ પ્રોપર્ટીને ‘Yes’ અથવા ‘No’માં સેટ કરી શકાય.

આ પ્રોપર્ટી નક્કી કરે છે કે કોઈ રેકોર્ડને સમાપ્ત કરવા કોઈ ચોક્કસ ફિલ્ડમાં વપરાશકર્તાએ કિંમત દાખલ કરવી પડશે કે નહીં ? જો આપણા ટેબલના કોઈ અગત્યના ફિલ્ડની એન્ટ્રી સ્કીપ કરવા માટે પરવાનગી આપવા ન માંગતા હોઈએ તો આ પ્રોપર્ટીને ‘Yes’ વડે સેટ કરવી જોઈએ. ઉદાહરણ તરીકે, આપણે એવાં ફિલ્ડ કે જેમાં વ્યક્તિનાં નામ સંબંધિત કિંમતોનો સંગ્રહ કરતાં હોઈએ, તેને આ પ્રોપર્ટીથી સેટ કરવાં જોઈએ.

નોંધ : આપણે જે ફિલ્ડને પ્રાઈમરી કી બનાવેલ છે, તેના માટે *Entry Required* પ્રોપર્ટીને ‘Yes’ થી સેટ કરવાની જરૂર નથી.

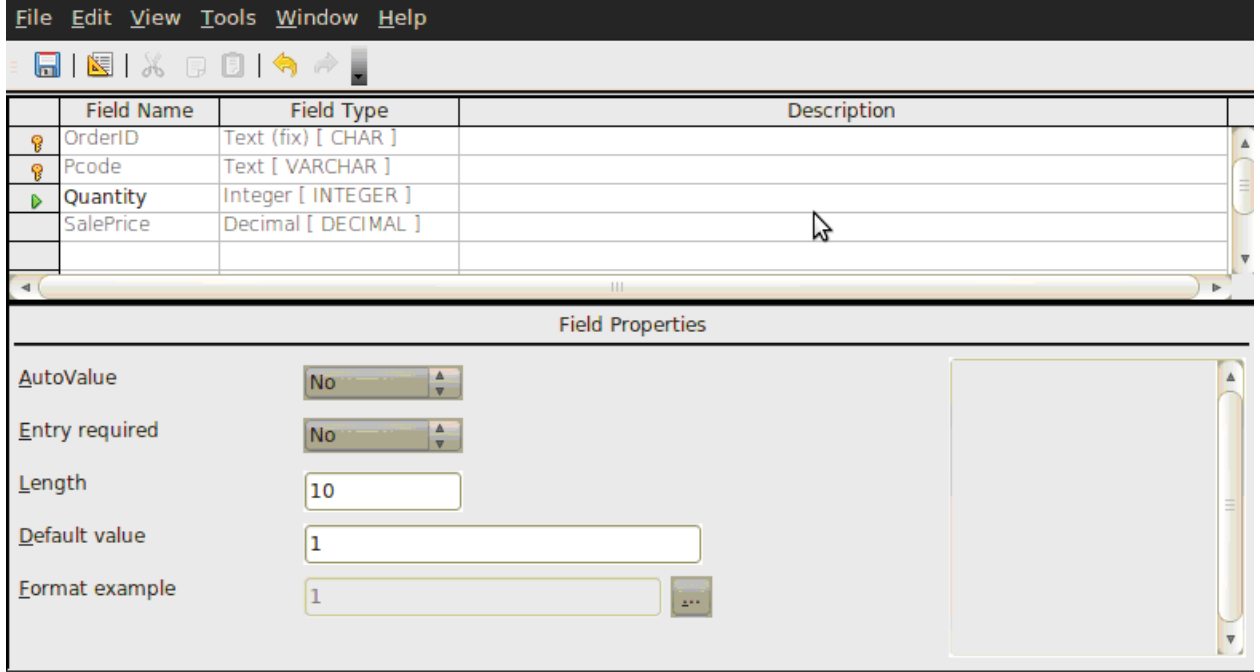


આકૃતિ 10.19 : એન્ટ્રી રિક્વાયર્ડ પ્રોપર્ટીને ‘Yes’ થી સેટ કરી

લેન્થ (Length) : જ્યારે કોઈ ચોક્કસ ફિલ્ડ માટે આપણે *Text* ડેટાટાઈપ પસંદ કરીએ, ત્યારે આ પ્રોપર્ટી આપોઆપ કેટલીક પૂર્વનિર્ધારિત કિંમત લઈ લે છે, દા.ત., *Text* ડેટાટાઈપ માટે 50. આપણી જરૂરિયાત પ્રમાણે આપણે ફિલ્ડનું કદ જણાવી શકીએ. બેઝ આપોઆપ રીતે વિવિધ ન્યુમરિક, ડેઈટ/ટાઈમ, યસ/નો, અને મેમો ડેટાટાઈપ્સને અગાઉથી નિર્ધારિત કિંમતો આપે છે. આ કારણે આવા કિસ્સાઓમાં સ્કીન ઉપર આ પ્રોપર્ટી નિષ્ક્રિય કરી દેવામાં આવે છે.

ડિફોલ્ટ વેલ્યુ (Default value) : ઘણી વખત એવું બને છે કે ટેબલમાં નવો રેકોર્ડ એન્ટર કરતાં સમયે આપણે કોઈ ચોક્કસ ફિલ્ડમાં કોઈ પૂર્વનિર્ધારિત કિંમત આપોઆપ સંગ્રહ કરવા ઇચ્છતા હોઈએ. ધારોકે OrderDetail ટેબલમાં Quantity

નામથી કોઈ ફિલ્ડ છે. હવે, આ ફિલ્ડ માટે આપણે અગાઉથી નક્કી કરેલી કિંમત '1' નો સંગ્રહ કરવા ઈચ્છીએ છીએ. આ માટે Quantity ફિલ્ડ પસંદ કરો. આકૃતિ 10.20માં દર્શાવ્યા પ્રમાણે *Default value* પાસેના ટેક્સ્ટબોક્સમાં '1' ટાઈપ કરો. એક વખત કોઈ ફિલ્ડ માટે આ પ્રોપર્ટી સેટ કરીએ પછી જ્યારે ટેબલમાં આપણે નવો રેકોર્ડ ઉમેરીએ છીએ ત્યારે ઉલ્લેખિત પૂર્વનિર્ધારિત કિંમત આપોઆપ પ્રદર્શિત થાય છે. ડેટાએન્ટ્રી સમયે વપરાશકર્તા ઈચ્છે, તો કિંમતમાં ફેરફાર કરી શકે છે.



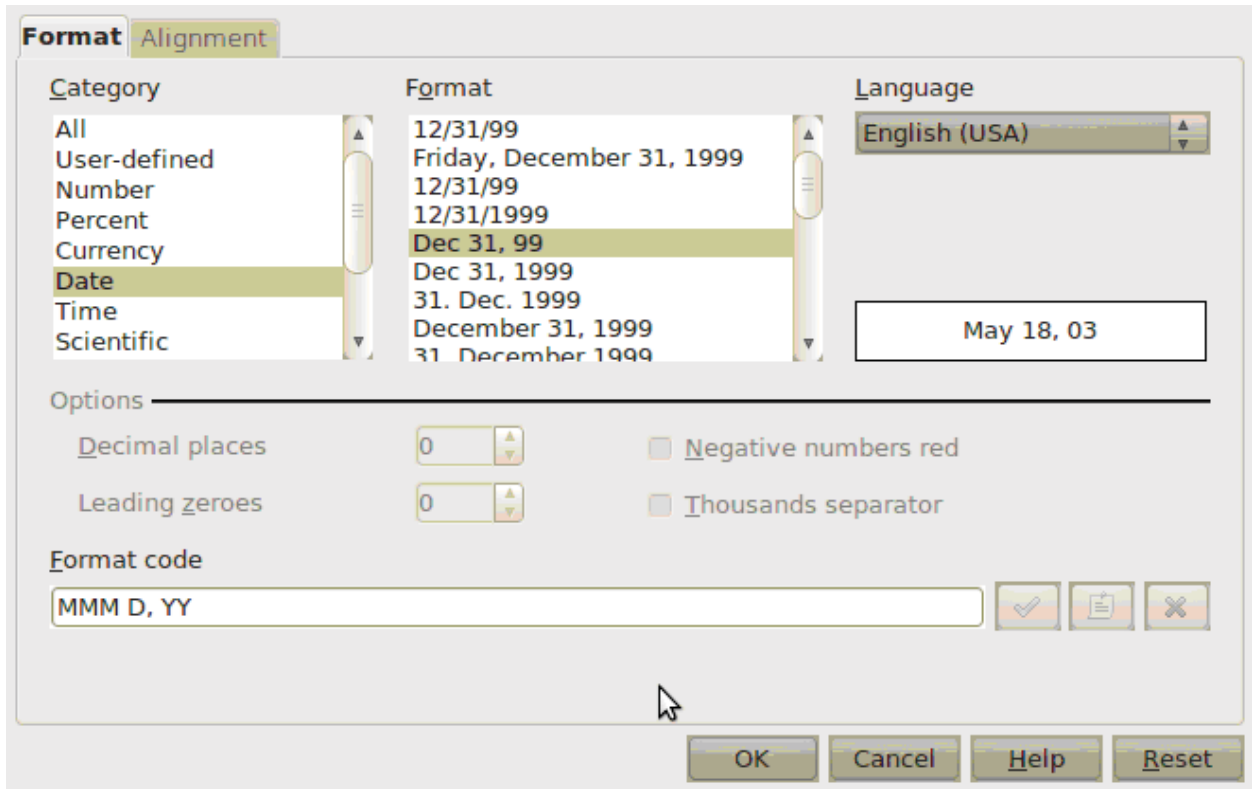
આકૃતિ 10.20 : ફિલ્ડ માટે ડિફોલ્ટ વેલ્યુ જણાવવી

ફોર્મેટ (Format) : આપણે ફિલ્ડમાં જે ડેટા એન્ટર કરીએ છીએ, તેને પ્રદર્શિત કરવા અને છાપવા માટેનું બંધારણ આ પ્રોપર્ટી સ્પષ્ટ કરે છે. ડેટાનો સંગ્રહ કઈ રીતે થયો છે, તેની કોઈ અસર તેના ઉપર નથી અને તે અમાન્ય કિંમતો (ડેટા) પણ ચકાસશે નહીં. વિવિધ પ્રકારની ડેટાટાઈપ્સ માટે ફોર્મેટ-પ્રોપર્ટી અલગ-અલગ સેટિંગ્સ વાપરે છે. બેઝનંબર, ડેઈટ/ટાઈમ અને યસ/નો ડેટા ટાઈપ્સ માટે કેટલાક પૂર્વનિર્ધારિત ફોર્મેટ્સ પૂરાં પાડે છે.

ઉદાહરણ તરીકે, Order tableમાં OrderDate ફિલ્ડનું ફોર્મેટ બદલવા માટે :

- ડિઝાઈન વ્યૂમાં Order ટેબલ ઓપન કરો.
- OrderDate ફિલ્ડ પસંદ કરો અને Field Properties પેનમાં રહેલાં 'Format example' લેબલના અંતમાં રહેલા બટન ઉપર ક્લિક કરો. આકૃતિ 10.21માં દર્શાવ્યા પ્રમાણે ફિલ્ડ ફોર્મેટ ડાયલોગબોક્સ ખૂલશે.
- Category શીર્ષક હેઠળની યાદીમાંથી Date વિકલ્પ પસંદ કરો.
- Format શીર્ષક હેઠળની યાદીમાંથી Dec 31, 1999 વિકલ્પ પસંદ કરો.
- હવે OK બટન ઉપર ક્લિક કરો.

હવે ટેબલ ઓપન કરો અને કોઈ માન્ય તારીખ કોઈ પણ સ્વરૂપમાં એન્ટર કરો. તમે અહીં જોઈ શકશો કે આ ફિલ્ડમાં એન્ટર કરેલી તમામ કિંમત આપોઆપ બદલીને મહિનાના પહેલા ત્રણ અક્ષર, પછી એક સ્પેશ ત્યાર પછી બે આંકડામાં તારીખ અને વર્ષ કે જેને અલ્પવિરામથી જુદા કરેલા છે.



આકૃતિ 10.21 : પૂર્વનિર્ધારિત વિવિધ ફોર્મેટ

સારાંશ

આ પ્રકરણમાં ટેબલ ઉપર કરવામાં આવતાં કાર્યો ઉપર આપણે ચર્ચા કરી. ટેબલ વચ્ચે કઈ રીતે સંબંધ બનાવી શકાય અને ટેબલ વચ્ચે કયા-કયા પ્રકારના સંબંધ હોઈ શકે તે બાબત આપણે શીખ્યાં. અંતમાં આપણે બેઝની લાક્ષણિકતા ફિલ્ડ પ્રોપર્ટીનો ઉપયોગ કરીને ડેટાએન્ટ્રી ઉપર નિયંત્રણ કરી શકાય તેમજ ટેબલમાં ખોટા (અમાન્ય) ડેટા દાખલ કરતાં અટકાવી શકાય તે બાબત શીખ્યાં. આપણે મોડર્ન ઇલેક્ટ્રોનિક્સ સ્ટોર માટે ડિઝાઇન કરેલા ડેટાબેઝમાં અર્થપૂર્ણ રેકોર્ડ એન્ટર કરવાની જરૂર છે અને આગળના પ્રકરણમાં ક્વેરીઝનો ઉપયોગ કરીને ઉમેરેલાં ડેટા આપણને ઈન્ફર્મેશન તૈયાર કરવામાં કઈ રીતે મદદરૂપ થાય છે તે જોઈશું.

સ્વાધ્યાય

1. ટેબલમાં ઈન્સર્શનનું કાર્ય આપણે કઈ રીતે કરી શકીએ તે સમજાવો.
2. બેઝમાં તમે સંબંધનો શું અર્થ સમજો છો ?
3. સંબંધના પ્રકારો યોગ્ય ઉદાહરણ આપી સમજાવો.
4. ફોરેન કીનો ખ્યાલ યોગ્ય ઉદાહરણ આપી સમજાવો.
5. તમે કમ્પોઝિટ પ્રાઇમરી કીનો શું અર્થ સમજો છો ?
6. ડેટા-નિરર્થકતા યોગ્ય ઉદાહરણ આપી સમજાવો.
7. રેફરેન્શલ ઈન્ટિગ્રિટી યોગ્ય ઉદાહરણ આપી સમજાવો.
8. ફિલ્ડપ્રોપર્ટીની જરૂરિયાત શું સૂચવે છે ?

9. ડેટાબેઝમાં 'ડિકોલ્ટ વેલ્યુ'ની અગત્યતા સમજાવો.

10. 'ઓટોવેલ્યુ' પ્રોપર્ટી સમજાવો.

11. 'એન્ટ્રી રિક્વાયર્ડ' પ્રોપર્ટીનો શું ઉપયોગ છે ?

12. 'ફોર્મેટ' પ્રોપર્ટીનો શું ઉપયોગ છે ?

13. હેતુલક્ષી પ્રશ્નો

(1) નીચેનામાંથી કયું કાર્ય ડેટાબેઝના રેકોર્ડ ઉપર કરી શકાતું નથી?

- (a) ઈન્સર્ટ (b) ડિલીટ
(c) હાઈડ (d) અપડેટ

(2) નીચેનામાંથી શું બેઝવિઝાર્ડ વડે સેટિંગનાં મહત્તમ લેવલ રજૂ કરે છે ?

- (a) 3 (b) 4
(c) 5 (d) 6

(3) નીચેનામાંથી શું ડેટા-નિરર્થકતાનો નિર્દેશ કરે છે ?

- (a) ડેટા ડિલીટ કરવા (b) ડેટાનું પુનરાવર્તન
(c) ટેબલનું વિઘટન (d) ટેબલની રિલેશનશિપ્સ

(4) TeacherSubject (Tcode, Scode, Standard), રિલેશન આપેલું હોય, તો નીચેનામાંથી કયું ફિલ્ડ ફોરેન કી માટે યોગ્ય રહેશે ?

- (a) Tcode (b) Scode
(c) Tcode,Standard (d) Tcode,Scode

(5) નીચેનામાંથી કયું વિધાન ખોટું છે ?

- (a) દરેક ટેબલને પ્રાઈમરી કી હોવી જોઈએ.
(b) પ્રાઈમરી કી એક કરતાં વધારે ફિલ્ડનું કોમ્બિનેશન (જોડાણ) હોઈ શકે.
(c) પ્રાઈમરી કી અને ફોરેન કીની ડેટાટાઈપ એકસમાન હોવી જોઈએ.
(d) પ્રાઈમરી કી અને ફોરેન કીનું નામ એક જ હોવું જોઈએ.

(6) નીચેનામાંથી કયા પ્રકારનો સંબંધ વિદ્યાર્થી અને શિક્ષક વચ્ચે હોય ?

- (a) એકથી એક (b) એકથી અનેક
(c) અનેક થી અનેક (d) કોઈ સંબંધ નહીં.

(7) નીચેનામાંથી કયા પ્રકારની રિલેશનશિપ વિદ્યાર્થી અને વર્ગ શિક્ષક વચ્ચે હોય ?

- (a) વન ટુ વન (b) વન ટુ મેની
(c) મેની ટુ મેની (d) કોઈ રિલેશનશિપ નહીં.

(8) રેફરેન્શલ ઈન્ટિગ્રિટીનો અમલ કરવા માટે નીચેનામાંથી કયો વિકલ્પ શક્ય છે ?

- (a) નો એક્શન (b) સેટ નલ
(c) ડિલીટ ટેબલ (d) અપડેટ કેસ્કેડ

(9) નીચેનામાંથી કયા ડેટાટાઈપ માટે ફિલ્ડની 'ઓટોવેલ્યુ' પ્રોપર્ટી સેટ કરી શકાય ?

- (a) ટેક્સ્ટ (b) ઈમેજ
(c) ઈન્ટિજર (d) બુલિયન

(10) નીચેનામાંથી કયું વિધાન બેઝમાં 'Default field' પ્રોપર્ટી માટે ખરું છે ?

- (a) ડિફોલ્ટવેલ્યુ ફક્ત ન્યુમરીક વેલ્યુ (આંકડાકીય કિંમત) જ હોઈ શકે.
- (b) એક વખત સેટ કરેલી ડિફોલ્ટવેલ્યુ પછી બદલી શકાય.
- (c) ડિફોલ્ટવેલ્યુ 500 કરતાં વધારે ન હોઈ શકે.
- (d) ડિફોલ્ટવેલ્યુ લેન્થમાં સેટ કરેલી કિંમત કરતાં વધારે હોવી જોઈએ.

(11) નીચેનામાંથી કઈ પ્રોપર્ટી NOT NULL બરાબર છે ?

- (a) લેન્થ
- (b) ડિફોલ્ટ
- (c) રિક્વાયર્ડ
- (d) ફોર્મેટ

પ્રાયોગિક સ્વાધ્યાય

1. ટેબલનું સ્ટ્રક્ચર Tablename (Attribute1, Attribute2,)ના સ્વરૂપમાં આપેલું છે. બેઝમાં દરેક ડેટાબેઝ માટે નીચેનાં પાંચ કાર્યો કરો :

- (a) આપેલાં નામો અને એટ્રીબ્યુટ્સ સાથેનાં ટેબલ બનાવો.
- (b) દરેક એટ્રીબ્યુટ માટે યોગ્ય ડેટાટાઇપ નક્કી કરો.
- (c) દરેક ડેટાબેઝની પ્રાઇમરી કી અને ફોરન કી દર્શાવો.
- (d) ડેટાબેઝની અંદર બનાવેલાં ટેબલ વચ્ચે રિલેશનશિપ્સ બનાવો.
- (e) દરેક ટેબલમાં ઓછામાં ઓછા પાંચ યોગ્ય રેકોર્ડ ઉમેરો.

A.	Movie(MovieId, MovieName, DateOfRelease) Screen(ScreenId, Location, MaxCapacity) Current(MovieId, ScreenId, DateOfArrival, DateOfClosure)
B.	Customer(CustomerID, CustomerName, Address, City, BirthDate, ContactNo) Account (AccountNo, CustomerId, AccountType, AccountBalance) Transaction(TransId, AccNo, TransDate, TransType, TransAmount)
C.	Book(BookId, BookTitle, Description, BookAuthor, Status) Student(StudId, StudName, Address,City, BirthDate,ContactNo) Book_Issue(TransId, BookId, StudId, IssueDate, ReturnDate)
D.	Employee(EmpId, EmpName, Address, BirthDate, ContactNo, ManagerId) Department(DeptId, DeptName) Employee_Department(EmpId, DeptId, Salary)
E.	Flight(FlightId, CompanyName, FlightFrom, FlightTo, FlightFare, Capacity) Passenger(PassengerId, Name, Address, City, BirthDate, Gender, ContactNo) Flight_Scheduled(Transid, FlightId, DepartureDate) Flight_Passenger(Transid, PassengerId)

F.	Train(TrainId, TrainName, TrainFrom, TrainTo, DepartureTime, ArrivalTime) Train_fare(TrainId, Class, Fare) Passenger(PassengerId, Name, Address, City, BirthDate, ContactNo, Email_Id) Train_Passenger(TrainId, PassengerId)
G.	Vehicle(VehicleId, Name, Type, Price, Description) Customer(CustomerId, CustomerName, Address, BirthDate, ContactNo) Vehicle_Customer(VehicleId, CustomerId, PurchaseDate, DeliveryDate)
H.	Product(ProductId, ProductName, Quantity, ProductPrice) Salesman(SCode, SName, SAddress, BirthDate, ContactNo) SalesOrder(SCode, ProductId, QtySold)
I.	Customer(CustomerId, CustName, Gender, CustAddr, CustCity, EmailID, ContactNo) Magazine(MagazineId, MagazineName, UnitRate, Publication) Subscription(CustomerId, MagazineId, StartDate, EndDate)
J.	Employee(EmpCode, EmpName, EmpAddress, EmpCity, EmpSalary, EmpJobName) Project(ProjCode, ProjName, StartDate, ProjPrice) WorksFor(ProjCode, EmpCode, HoursWorked)

2. આકૃતિ 10.22 અને 10.23માં દર્શાવેલાં નમૂનાનાં ટેબલ્સમાં પ્રાર્થમરી કી અને ફોરન કી ઓળખો.

Student	
Field Name	Description
Gmo	General Register Number
Firstname	Name of the Student
Surname	Surname of the Student
Address	Address of the Student
City	City
Pincode	Pincode
Birthdate	Date of Birth
Gender	Male or Female
Standard	Studying in which standard
Join_date	Date of Joining School
Leaving_date	Date of Leaving School

Teacher	
Field Name	Description
Firstname	Name of the Teacher
Surname	Surname of the Teacher
Address	Address of the Teacher
City	City
Pincode	Pincode
Phone_no	Phone number of teacher
Email_id	E-mail id of teacher
Mobile_no	Mobile number of teacher

Subject	
Field Name	Description
Sub_Name	Name of the Subject
Details	Description of the subject

આકૃતિ 10.22 : સ્ટુડન્ટ મેનેજમેન્ટ સિસ્ટમનાં માસ્ટર ટેબલ

Standard_Subj : To store details about subjects taught in each standard		
Field Name	Data Type	Description
Standard	Number	Which Standard
Scode	Text	Which Subject

Attendance : To store daily attendance of the Students		
Field Name	Data Type	Description
Grno	Number	General Register of a Student
Date	Date/Time	On which Date
Pr_ab	Yes/No	Present or Absent

Fees : To store details of Fee amount paid by students		
Field Name	Data Type	Description
Grno	Number	General Register Number
Date	Date/Time	On which date
Tuition_fee	Currency	Tuition fee paid by the student
Term_fee	Currency	Term fee paid by the student

Cultural : To keep record of Interest of students in Cultural Activities		
Field Name	Data Type	Description
Grno	Number	General Register Number
Interest	Text	Interested in which activity
Achievement	Memo	Any achievements?

Subj_Teacher : To store details about which teacher is teaching the subject in each standard		
Field Name	Data Type	Description
Year	Number	Current Year
Standard	Number	In which Standard
Scode	Text	Subject Code
Tcode	Text	Teacher Code

Result : To store result details of each student		
Field Name	Data Type	Description
Grno	Number	General Register No.
Scode	Text	Of which Subject
Month	Text	In which Month
Year	Number	Which Year
Term	Number	Name of the Term
Marks	Number	Marks scored

Sports : To keep record of Interest of students in Sports		
Field Name	Data Type	Description
Grno	Number	General Register No.
Interest	Text	Interested in which game
Achievement	Memo	Any achievements?

આકૃતિ 10.23 : સ્ટુડન્ટ મેનેજમેન્ટ સિસ્ટમનાં ટ્રાન્ઝેક્શન ટેબલ



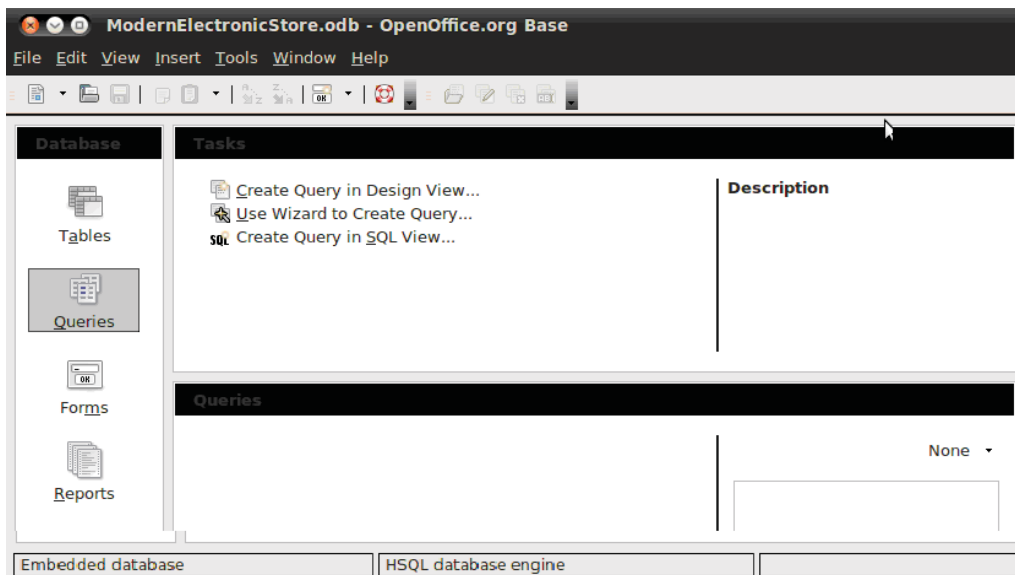


આ પહેલાના પ્રકરણમાં આપણે જોયું કે વિગતો (data)ના વિશાળ સમૂહને ડેટાબેઝ તરીકે ઓળખવામાં આવે છે. હવે એવી સ્થિતિ વિશે વિચારીએ કે જેમાં આપણને આ ડેટાબેઝમાંથી માહિતી મેળવવાની જરૂર પડે. ધારોકે, આપણા ગ્રાહકોને સ્વાતંત્ર્યદિવસ અંગેની માહિતી મોકલવા માંગતા હોઈએ, તો તે માટે આપણને ગ્રાહકોનાં નામ અને સરનામાંની યાદીની જરૂર પડશે. જો customer નામના ટેબલમાં આ વિગતોનો સંગ્રહ કરેલો હોય તો તે ટેબલ ખોલવાથી તેને મેળવી શકાશે. ડેટાબેઝમાંથી નિશ્ચિત માહિતીના સમૂહને મેળવવા માટે બેઝ સોફ્ટવેરમાં ક્વેરી (Query) નામની સુવિધાનો ઉપયોગ કરવામાં આવે છે. ડેટાબેઝમાં વિગતોનો સંગ્રહ કેવી રીતે કરવામાં આવ્યો છે. તેનાથી અનપેક્ષ ઉપયોગકર્તા પોતાની પસંદગી, માપદંડ કે ગોઠવણી મુજબની માહિતી મેળવી શકે છે. “વર્તમાન વર્ષમાં એક લાખ રૂપિયાથી વધુ ઉત્પાદનો ખરીદ્યાં હોય તેવા ગ્રાહકની વિગતો દર્શાવો” અથવા “એક પણ વખત વેચાણ થયું ન હોય તેવી વસ્તુઓની વિગતો દર્શાવો” જેવા પ્રશ્નો આપણે ક્વેરી દ્વારા પૂછી શકીએ છીએ. ક્વેરીની રચના કરવાની, સંગ્રહ કરવાની અને જરૂર પડે ઉપયોગમાં લેવાની ક્ષમતા એ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનો ઉપયોગ કરવાનું એક મુખ્ય કારણ છે.

ક્વેરી વ્યાખ્યાયિત કરવી (Defining Query)

ક્વેરીનો મૂળભૂત અર્થ થાય છે : પ્રશ્નો પૂછવા, તપાસ કરવી અથવા પૃથક્કરણ કરવું. બેઝમાં ક્વેરીનો અર્થ છે ડેટાબેઝમાં આવેલી કોઈ માહિતી વિશે પ્રશ્નોત્તરી કરવી. ક્વેરી દ્વારા બેઝને સૂચવવામાં આવે છે કે ઉપયોગકર્તા સમક્ષ માત્ર નિશ્ચિત ફિલ્ડ અને રેકોર્ડ દર્શાવવાનાં છે. કોઈ એક કે વધુ ટેબલના જૂથમાંથી એક જ સમયે માહિતી મેળવવા માટેના નિયમોના ગણને ક્વેરી કહે છે. ક્વેરીનું પરિણામ ટેબલ સ્વરૂપે દર્શાવવામાં આવે છે. ક્વેરીમાં રેકોર્ડનો સમૂહ આડી હરોળ (દરેક રેકોર્ડ પ્રમાણે) અને ઊભી હરોળ (દરેક ફિલ્ડ પ્રમાણે) સ્વરૂપે ગોઠવાયેલ હોય છે.

ક્વેરીની રચના કરવા માટે ડેટાબેઝ ખોલી, ડાબી બાજુના વિભાગમાં આવેલા Queries લખેલા આઈકન પર ક્લિક કરો. ક્વેરી વિન્ડોની ગોઠવણ બેઝમાં આવેલ અન્ય વિન્ડો જેવી જ છે. ટાસ્કપેનમાં ઉપરની બાજુએ ક્વેરી બનાવવા માટેની જુદી-જુદી પદ્ધતિઓ દર્શાવેલી હોય છે. પહેલા બનાવેલી ક્વેરીની યાદી Queries શીર્ષક હેઠળના વિભાગમાં આપવામાં આવે છે. Queries આઈકન પસંદ કરવામાં આવ્યું હોય, ત્યારે વિન્ડોનો દેખાવ આકૃતિ 11.1માં દર્શાવ્યો છે.



આકૃતિ 11.1 : ક્વેરી વિન્ડો

આકૃતિ 11.1માં દર્શાવ્યા મુજબ બેઝમાં ટાસ્કપેનમાં ક્વેરી બનાવવા માટેની ત્રણ જુદી-જુદી રીત આપવામાં આવી છે:

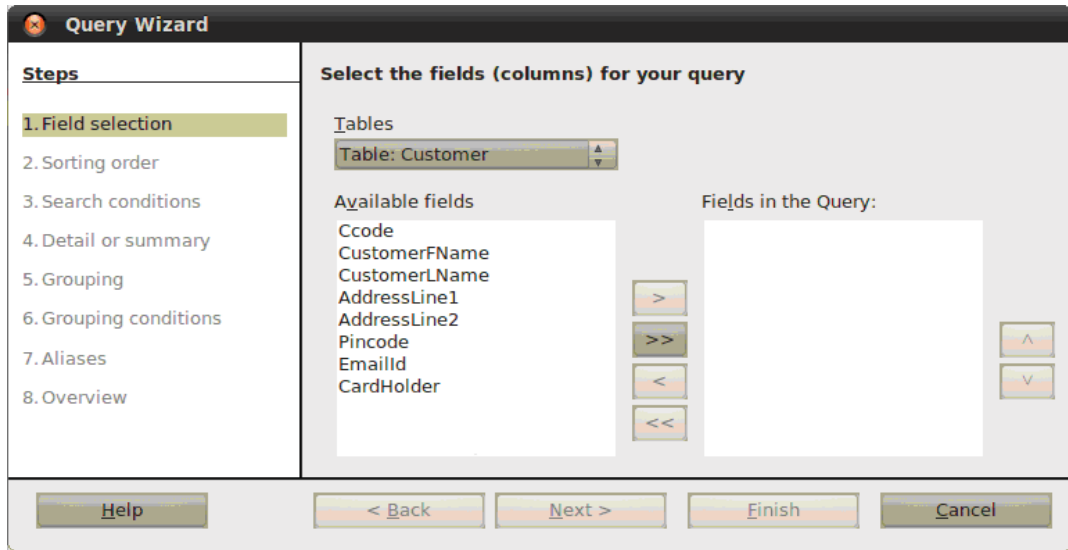
- Create Query in Design View...: આ રીતમાં ન્યૂનતમ માર્ગદર્શન આપવામાં આવે છે.
- Use Wizard to Create Query... : ક્વેરીની રચના કરવાની પદ્ધતિ વિશે વિઝાર્ડ પૂરો પાડવામાં આવશે.
- Create Query in SQL View... : આ રીતમાં કોઈ જ પ્રકારનું માર્ગદર્શન પૂરું પાડવામાં આવશે નહીં. ઉપયોગકર્તા પાસે યોથી પેઢીની કમ્પ્યુટર લેંગ્વેજ Structured Query Languageનું જ્ઞાન હોવું જરૂરી બનશે.

જો તમે ડેટાબેઝ નિષ્ણાત હો, તો Design View કે SQL View પસંદ કરી શકો છો. જોકે, આપણા વિચારોને ક્વેરી દ્વારા રજૂ કરવા માટે બેઝના ક્વેરી વિઝાર્ડનો ઉપયોગ કરવામાં આવે, તો તે માટે કોઈ વિશેષ જ્ઞાનની આવશ્યકતા રહેતી નથી.

હવે, આ દરેક વિકલ્પની કમશ: ચર્ચા કરીશું. ગ્રાહકોનાં નામ અને સરનામાં દર્શાવે તે માટે આપણે એક ક્વેરીની રચના કરીશું.

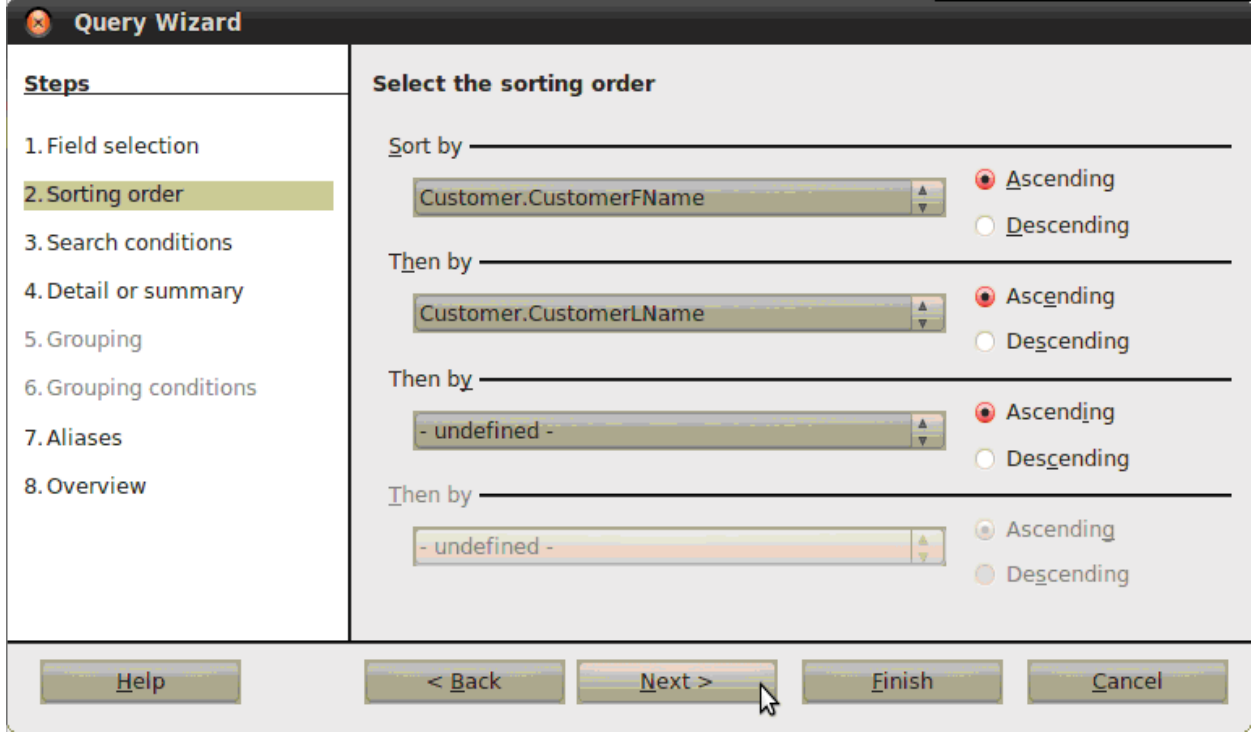
વિઝાર્ડની મદદથી ક્વેરી બનાવવી (Query Creation Using Wizard)

- Use Wizard to Create Query.... વિકલ્પ ઉપર ડબલ ક્લિક કરો. આમ કરવાથી આકૃતિ 11.2માં દર્શાવ્યા મુજબ Query Wizard ડાયલોગબોક્સ રજૂ કરવામાં આવશે. ડાયલોગબોક્સની ડાબી બાજુએ જોતાં જણાશે કે ક્વેરીની રચના કરવા માટેનાં આઠ પગલાં અહીં આપવામાં આવેલાં છે. તેમાં પ્રથમ પગલું ફિલ્ડની પસંદગી માટેનું છે તે ફરજિયાત છે. પરિણામમાં દર્શાવવાનાં હોય તેવા ફિલ્ડની પસંદગી આ પગલામાં કરવામાં આવે છે. પરિણામની સંરચના (format) નિશ્ચિત કરવા માટે અન્ય પગલાંનો ઉપયોગ કરવામાં આવે છે તથા જરૂરી ન હોય, તો તેને છોડી પણ શકાય છે.
- ક્વેરી બનાવવા માટેનાં સૌપ્રથમ પગલામાં ટેબલ પસંદ કરી તેમાંથી માહિતી મેળવવાની હોય તેવા ફિલ્ડની પસંદગી કરવામાં આવે છે. Available fields લિસ્ટબોક્સમાં આવેલા ફિલ્ડમાંથી પસંદગીનાં ફિલ્ડ અથવા તો તમામ ઉપલબ્ધ ફિલ્ડની પસંદગી કરી શકાય છે. Available fields અને Field in the Query: લિસ્ટબોક્સમાં ફિલ્ડનું સ્થાનાંતરણ કરવા માટે ડાબી અને જમણી બાજુની તીરની નિશાની ધરાવતા બટનનો ઉપયોગ કરી શકાય છે. અંતમાં, Field in the Query: લિસ્ટબોક્સમાં પસંદ કરવામાં આવેલ ફિલ્ડને ક્વેરીના પરિણામમાં સમાવવામાં આવશે. પસંદ કરવામાં આવેલા ફિલ્ડને ત્યાર બાદ up અને down બટનનો ઉપયોગ કરી ક્રમબદ્ધ કરી શકાય છે. આકૃતિ 11.2માં આપણે Customer ટેબલ પસંદ કરેલ છે અને તેમાં આવેલાં તમામ ફિલ્ડ Available fields : લિસ્ટબોક્સમાં જોવા મળે છે.



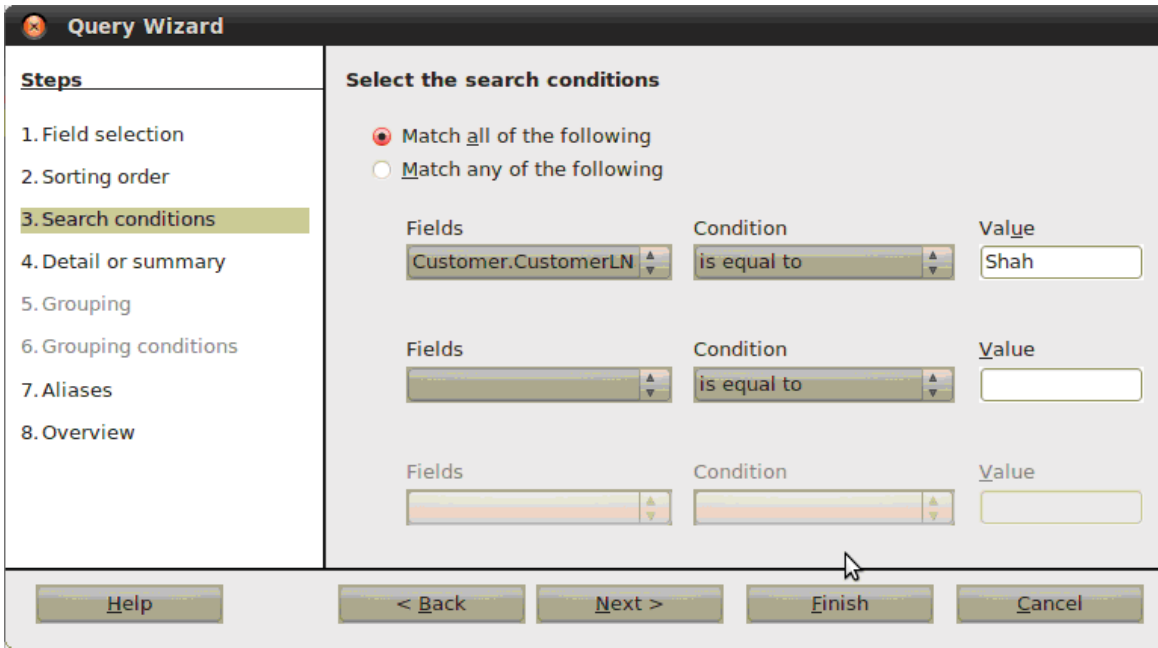
આકૃતિ 11.2 : ક્વેરી માટે ટેબલ અને ફિલ્ડની પસંદગી

- બીજા પગલામાં કોઈ નિશ્ચિત ફિલ્ડની વિગતોને ક્વેરીના પરિણામમાં ક્રમબદ્ધ ગોઠવવા માટેનો વિકલ્પ પૂરો પાડવામાં આવે છે. પરિણામ રૂપે દર્શાવાતી માહિતીને ક્રમબદ્ધ ગોઠવવા માટે અહીં વધુમાં વધુ ચાર ફિલ્ડ સુધીની સુવિધા ઉપલબ્ધ છે. જેમ કે, આપણે ક્વેરીના પરિણામને પ્રથમ ગ્રાહકનાં નામ પ્રમાણે અને પછીથી તેની અટક પ્રમાણે ગોઠવીને દર્શાવવું હોય. ફિલ્ડને ક્રમબદ્ધ ગોઠવ્યા બાદ Next બટન પર ક્લિક કરો. આકૃતિ 11.3માં ફિલ્ડને ક્રમબદ્ધ કરવાની રીત દર્શાવી છે.



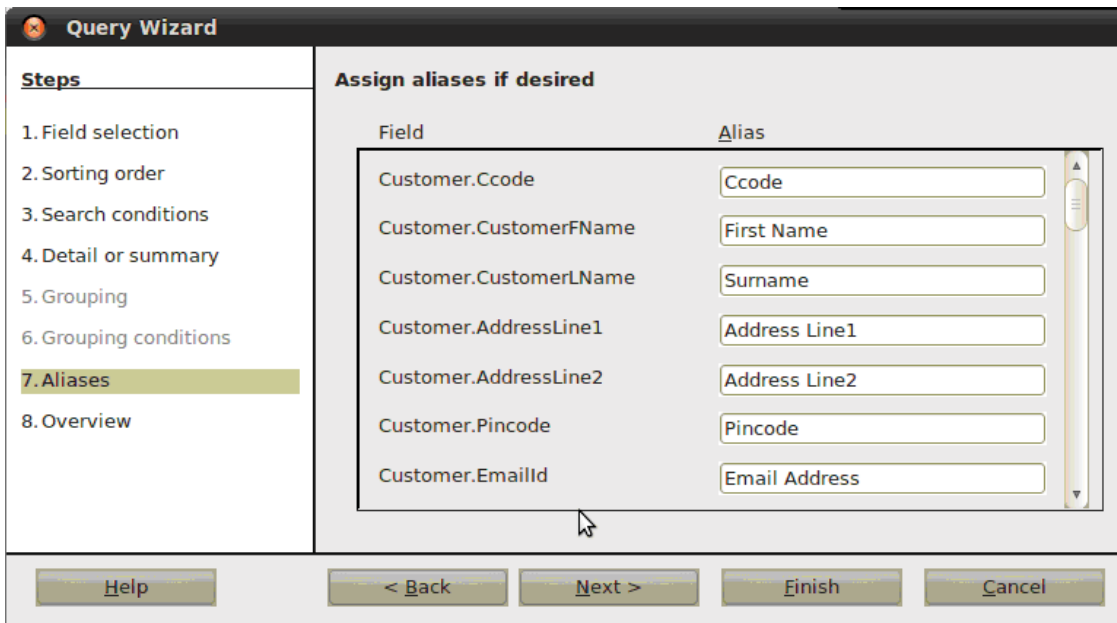
આકૃતિ 11.3 : ફિલ્ડને ક્રમબદ્ધ ગોઠવવાં

- વિઝાર્ડના ત્રીજા પગલામાં આપણે બરેબર ક્વેરીની ગોઠવણ કરીશું. અહીં ફિલ્ડ માટેની યોગ્ય કિંમત, શરત અને પ્રાયલકિંમતો (*Value parameters*) પસંદ કરવામાં આવે છે. એક ક્વેરીમાં મહત્તમ ત્રણ માપદંડોનો ઉપયોગ કરી શકાય છે. આપણા ઉદાહરણમાં “Shah” અટક ધરાવતા ગ્રાહકોની યાદી દર્શાવવી હોય, તો *Fields* ડ્રોપડાઉન મેનુમાંથી *CustomerLName* પસંદ કરી, ત્યાર બાદ *Condition* નામના ડ્રોપડાઉન મેનુમાંથી *equal to* પસંદ કરો અને અંતમાં *Value* નામના ટેક્સ્ટબોક્સમાં “Shah” ટાઈપ કરો. અહીં આપણી પાસે બે વિકલ્પ ઉપલબ્ધ છે : *Match all of the following* અને *Match any of the following*. આપણે એક જ માપદંડ ઉમેર્યો હોવાથી અત્યારે પૂર્વનિર્ધારિત ગોઠવણમાં ફેરફાર કરવાની કોઈ જરૂર નથી. જો એકથી વધુ માપદંડ ઉમેરવાના હોય, જેમકે, Shah અથવા Patel અટક ધરાવતા ગ્રાહકોની શોધ ચલાવવાની હોય, તો *Match any of the following* વિકલ્પ પસંદ કરવો જરૂરી બને. શોધ માટેની શરતો ઉમેર્યા બાદ Next બટન પર ક્લિક કરો. આકૃતિ 11.4માં Shah અટક ધરાવતા ગ્રાહક શોધવા માટેના માપદંડની ગોઠવણ દર્શાવવામાં આવી છે.



આકૃતિ 11.4 ફિલ્ડ પર શોધ માટેની શરત ઉમેરવી

- હવે આ વિઝાર્ડનાં ત્રણ પગલાં છોડીને બેઝ સાતમા પગલાં પર પહોંચી જશે. આપણા ઉદાહરણમાં પસંદ કરવામાં આવેલ Customer ટેબલમાં કોઈ આંકડાકીય ફિલ્ડ નથી અને તેથી આંકડાકીય ગણતરીઓ અને સારાંશ મેળવવા માટેના વિકલ્પોને છોડી દેવામાં આવશે.
- બેઝ સાતમા પગલામાં પસંદ કરેલા ફિલ્ડને ઉપનામ (alias) આપવાની અપેક્ષા રાખે છે. આ પગલાંનો હેતુ ફિલ્ડના નામને વિઝાર્ડ દ્વારા વધુ વાંચનક્ષમ સ્વરૂપે દર્શાવવાનો છે. તેના દ્વારા ફિલ્ડના નામમાં સૂક્ષ્મ સુધારા કરી શકાય છે. જેમ કે, શબ્દોની વચ્ચે જગ્યા ઉમેરી શકાય તથા CustomerFName જેવા ટૂંકા નામ ધરાવતા ફિલ્ડને ક્વેરીમાં First Name જેવા સંપૂર્ણ નામ સાથે દર્શાવી શકાય છે આ પગલું પણ વૈકલ્પિક છે. જરૂર જણાય તો ઉપનામ ઉમેરી શકાય છે. ઉપનામ ઉમેરવામાં ન આવે, તો ટેબલના ફિલ્ડ નામને તે જ સ્વરૂપે ક્વેરીમાં પણ દર્શાવવામાં આવે છે. તમામ ઉપનામ અપાઈ ગયા બાદ Next બટન પર ક્લિક કરો. આકૃતિ 11.5માં ઉપનામ ઉમેરવાની રીત દર્શાવી છે.



આકૃતિ 11.5 : ઉપનામ ઉમેરવાં

- અંતિમ આઠમા પગલામાં, અત્યાર સુધી પસાર થયેલાં તમામ પગલાંઓનું સર્વ સામાન્ય નિરીક્ષણ (overview) રજૂ કરવામાં આવશે. આકૃતિ 11.6માં હાલમાં તૈયાર કરવામાં આવેલી ક્વેરીનું આ પ્રકારનું નિરીક્ષણ આપેલું છે. *Name of the query* નામ ધરાવતા ખાનામાં ક્વેરીનું નામ ટાઈપ કરો. આપણે અહીં ક્વેરીને CustomerList નામ આપેલું છે. એક વાર આપણે પસંદ કરેલા વિકલ્પ તરફ નજર નાંખી લો. જરૂરી જણાય તો Back બટનનો ઉપયોગ કરી ફેરફારો કરી શકાય છે. એક વાર ક્વેરી તૈયાર થઈ ગયા બાદ માત્ર Design Viewની મદદથી તેમાં ફેરફારો કરી શકાશે. હવે, *How do you want to proceed after creating a query?* પ્રશ્નની નીચે આપેલા બે વિકલ્પ જુઓ. *Display Query* વિકલ્પ પસંદ કરવાથી તત્કાલ ક્વેરીનું પરિણામ જોઈ શકાશે અથવા તો *Modify Query* વિકલ્પ પસંદ કરી ક્વેરીને તેના ડિઝાઈન વ્યૂમાં ખોલી શકાય. વિઝાર્ડની મર્યાદા બહાર હોય તેવી કેટલીક અતિરિક્ત સુવિધાઓ ક્વેરીમાં ઉમેરવા માટે ડિઝાઈન વ્યૂનો ઉપયોગ કરી શકાય છે.

આકૃતિ 11.6 : તૈયાર કરેલી ક્વેરીનું નિરીક્ષણ

ઉપરોક્ત ઉદાહરણમાં માનો કે આપણે ગ્રાહકનાં નામ અને સરનામાંની સાથે તેના શહેરનું નામ, રાજ્યનું નામ અને દેશનું નામ પણ જોવું હોય. આપણે પસંદ કરેલા ટેબલમાં માત્ર Pincode ફિલ્ડ છે. Pincode ફિલ્ડને સંબંધિત એવી શહેર, રાજ્ય અને દેશના નામની વિગતો અન્ય ટેબલમાં આપેલી છે. ક્વેરી બનાવવા માટેનું વિઝાર્ડ એક જ ટેબલ પસંદ કરવાની અનુમતિ આપે છે. આમ, જટિલ ક્વેરી બનાવવા માટે ડિઝાઈન વ્યૂનો ઉપયોગ કરવામાં આવે છે.

- Finish બટન પર ક્લિક કરવાથી આકૃતિ 11.7માં દર્શાવ્યા મુજબ ક્વેરીનું પરિણામ ડેટાશીટ વ્યૂમાં જોવા મળે છે.

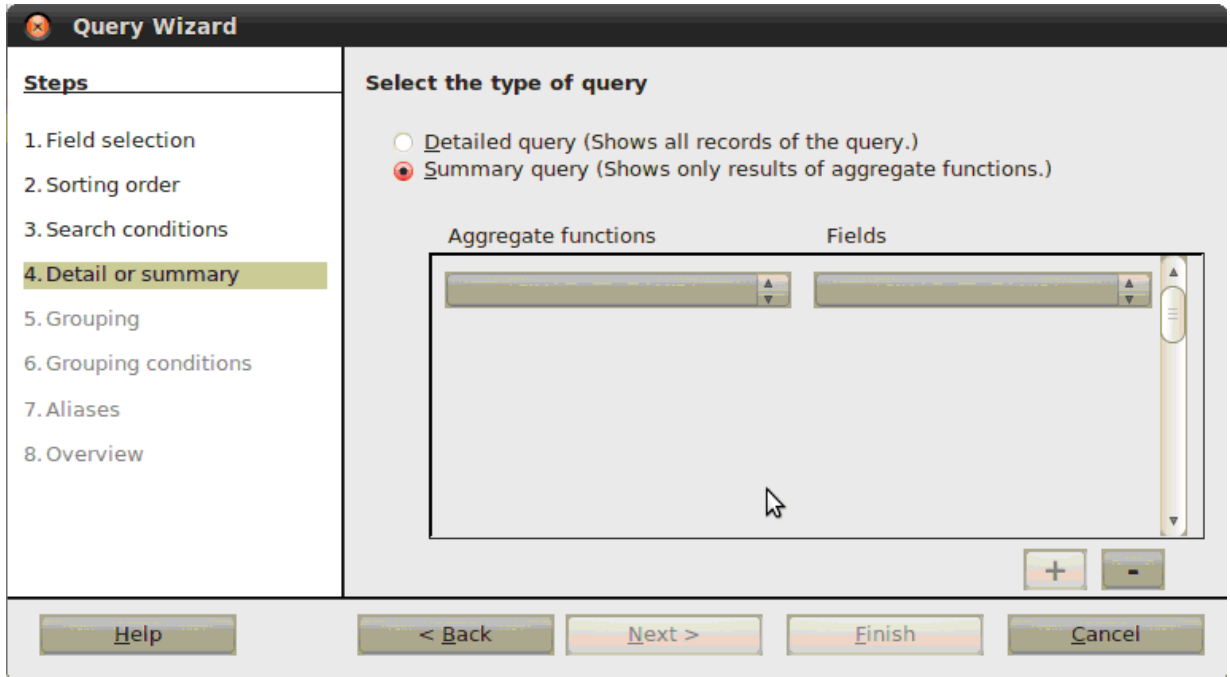
Ccode	First Name	Surname	Address Line1	Address Line2	Pincode	Email Address	CardHolder
C05	Harshit	Shah	17, Jaldeep-1	Bopal	380058	harshit@yahoo.cor	<input checked="" type="checkbox"/>

Record 1 of 1

આકૃતિ 11.7 ક્વેરીનું પરિણામ

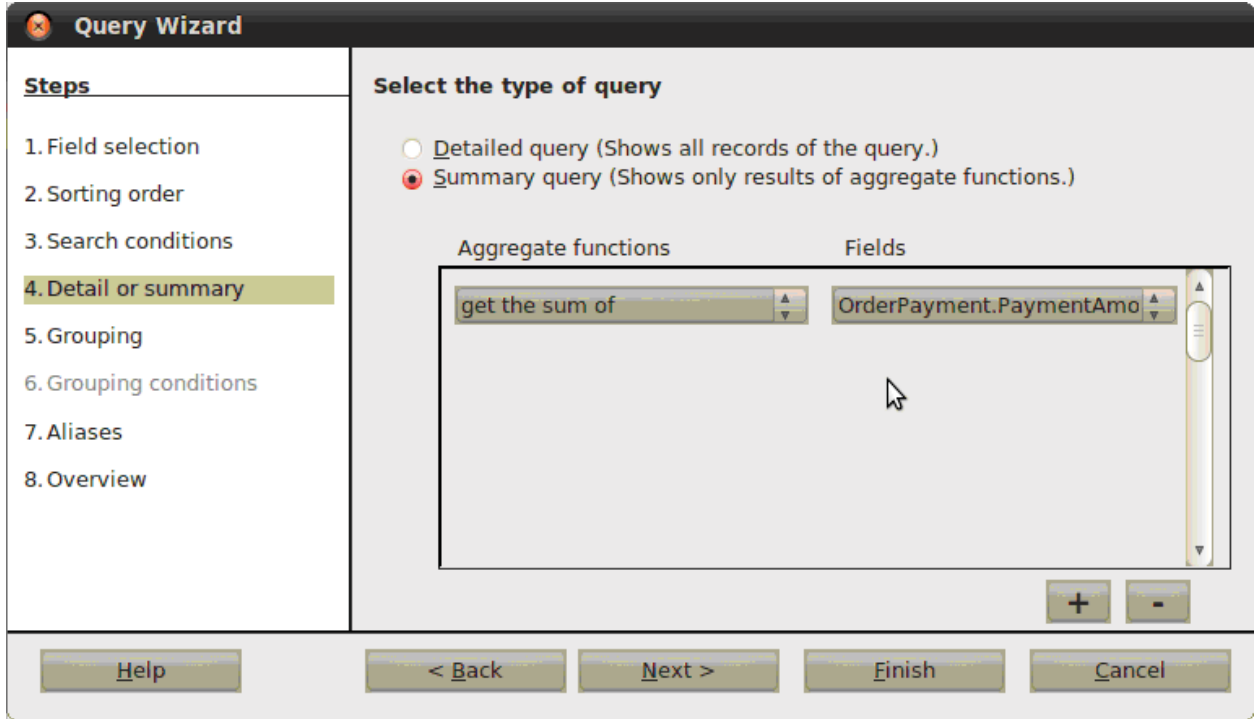
હવે, દરેક ઓર્ડર માટે મેળવેલી રકમની ગણતરી કરીએ. આંકડાકીય ફિલ્ડ સાથેના કાર્ય માટે ફરી એક વાર ક્વેરી વિઝાર્ડનો ઉપયોગ કરીએ. Queries વિભાગ પસંદ થયેલો છે, તેની ખાત્રી કરી લો.

- Use Wizard to Create Query... વિકલ્પ પર ડબલ ક્લિક કરો.
- ક્વેરી વિઝાર્ડના પ્રથમ પગલાંમાં OrderPayment ટેબલ પસંદ કરો. અહીં ડ્રોપડાઉન યાદીમાં અન્ય તમામ ટેબલનાં નામની સાથે CustomerList ક્વેરી પણ સમાવાયેલી જોઈ શકાય છે. બેઝ અન્ય ક્વેરીની રચના માટે પહેલેથી બનાવવામાં આવેલી ક્વેરીનો ઉપયોગ કરવાની અનુમતિ પણ આપે છે.
- OrderID અને PaymentAmount ફિલ્ડને પસંદ કરી 'ગ્રેટર થેન' (>) બટન પર ક્લિક કરી, *Fields in the Query*: લિસ્ટબોક્સમાં ઉમેરો.
- Next બટન પર ક્લિક કરો.
- બીજા પગલામાં *Sort by* લખેલા ડ્રોપડાઉન બોક્સમાં OrderID ફિલ્ડ પસંદ કરો.
- Next બટન પર ક્લિક કરો.
- અહીં આપણે રેકોર્ડને તારવવાના ન હોવાથી કોઈ માપદંડનો ઉલ્લેખ કરવાની જરૂર નથી. માટે Next બટન પર ક્લિક કરી ત્રીજા પગલાને છોડી દો. હવે આકૃતિ 11.8માં દર્શાવ્યા પ્રમાણે ચોથું પગલું રજૂ કરતું ડાયલોગબોક્સ દેખાશે.



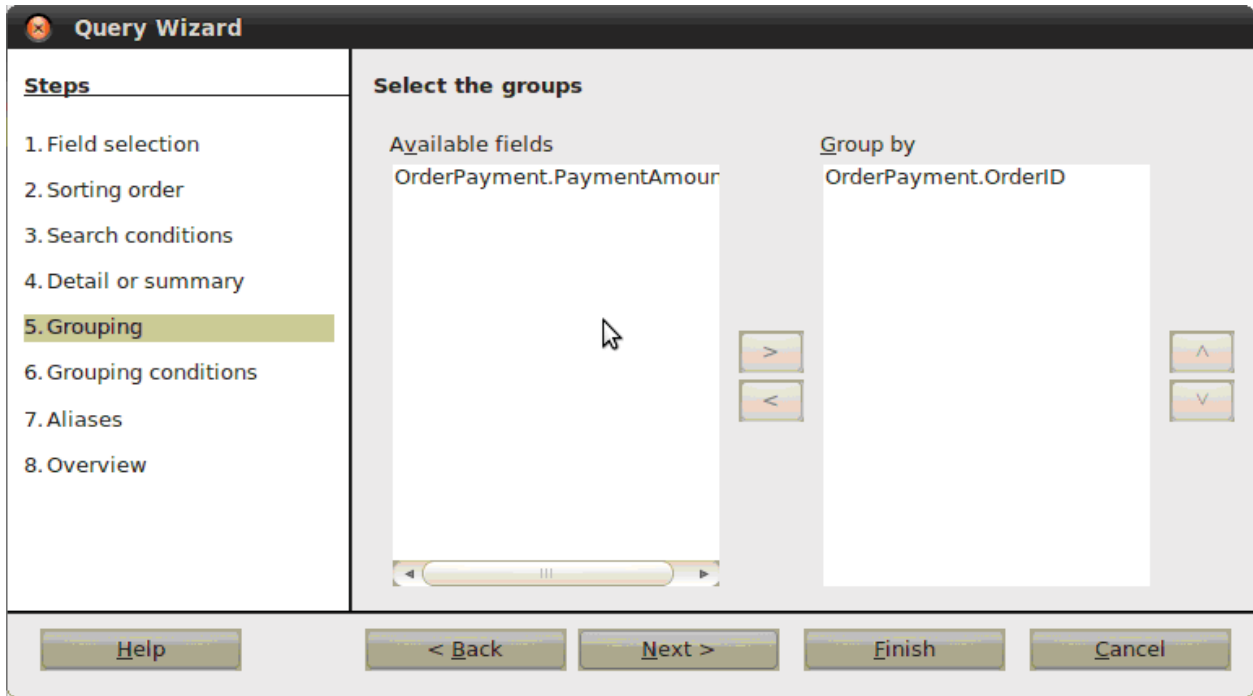
આકૃતિ 11.8 : પરિણામનો સારાંશ મેળવવો

- આકૃતિ 11.8માં દર્શાવ્યા મુજબ *Summary query (Show only results of aggregate functions)* વિકલ્પ પસંદ કરો.
- *Aggregate functions* નામના ડ્રોપડાઉન બોક્સ પર ક્લિક કરો. તેમાં *get the sum of* વિકલ્પ પસંદ કરો. આ જ રીતે, field ડ્રોપડાઉન બોક્સમાંથી *OrderPayment.PaymentAmount* ફિલ્ડ પસંદ કરો, જેથી *PaymentAmount* ફિલ્ડની કિંમતોનો સરવાળો કરી શકાય. આ રીત આકૃતિ 11.9માં દર્શાવી છે.



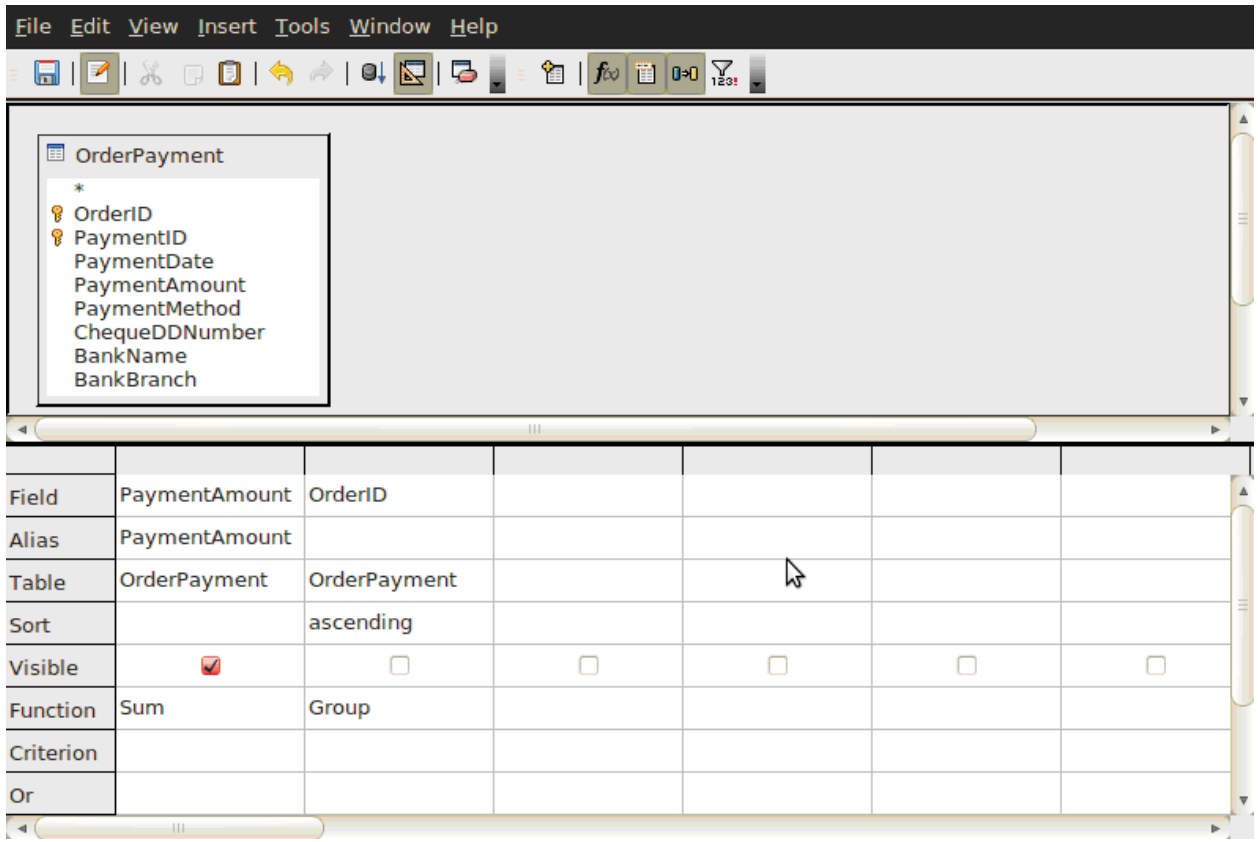
આકૃતિ 11.9 : એગ્રિગેટ વિધેયનો ઉપયોગ

- Next બટન પર ક્લિક કરો. તેનાથી પાંચમું પગલું સક્રિય થશે. અહીં જૂથ બનાવવા માટેનાં ફિલ્ડ પસંદ કરવામાં આવે છે. અહીં તમામ રેકોર્ડ માટેની ચુકવણીની રકમનો સરવાળો કરવાનું પસંદ કરી શકાય. પરંતુ તેના દ્વારા મોડર્ન ઇલેક્ટ્રોનિક સ્ટોર દ્વારા મેળવવામાં આવેલી તમામ ગ્રાહકોની રકમનો સરવાળો ગણવામાં આવશે. આપણે દરેક ઓર્ડર પ્રમાણે કુલ ચુકવણીની રકમ મેળવવા માંગીએ છીએ. માટે, જો આપણે જાતે (manually) આ ક્રિયા કરવાની હોય, તો સૌપ્રથમ OrderID ફિલ્ડ પ્રમાણે તમામ રેકોર્ડનાં જૂથ બનાવવાની જરૂર પડે. આનો અર્થ એ થયો કે OrderIDની કિંમત '1' હોય તેવા રેકોર્ડને એક જૂથમાં અને એ રીતે તમામ રેકોર્ડને ગોઠવવાની જરૂર પડે. ત્યાર બાદ દરેક જૂથ માટે ચુકવણીની રકમ ઉમેરવાની રહે. પરિણામ સ્વરૂપે ક્વેરીમાં દરેક OrderID માટે એક-એક રેકોર્ડ દર્શાવવામાં આવે. આમ, આકૃતિ 11.10માં દર્શાવ્યા પ્રમાણે આ પગલામાં OrderID ફિલ્ડને *Groupby* લિસ્ટબોક્સમાં પસંદ કરવામાં આવશે.



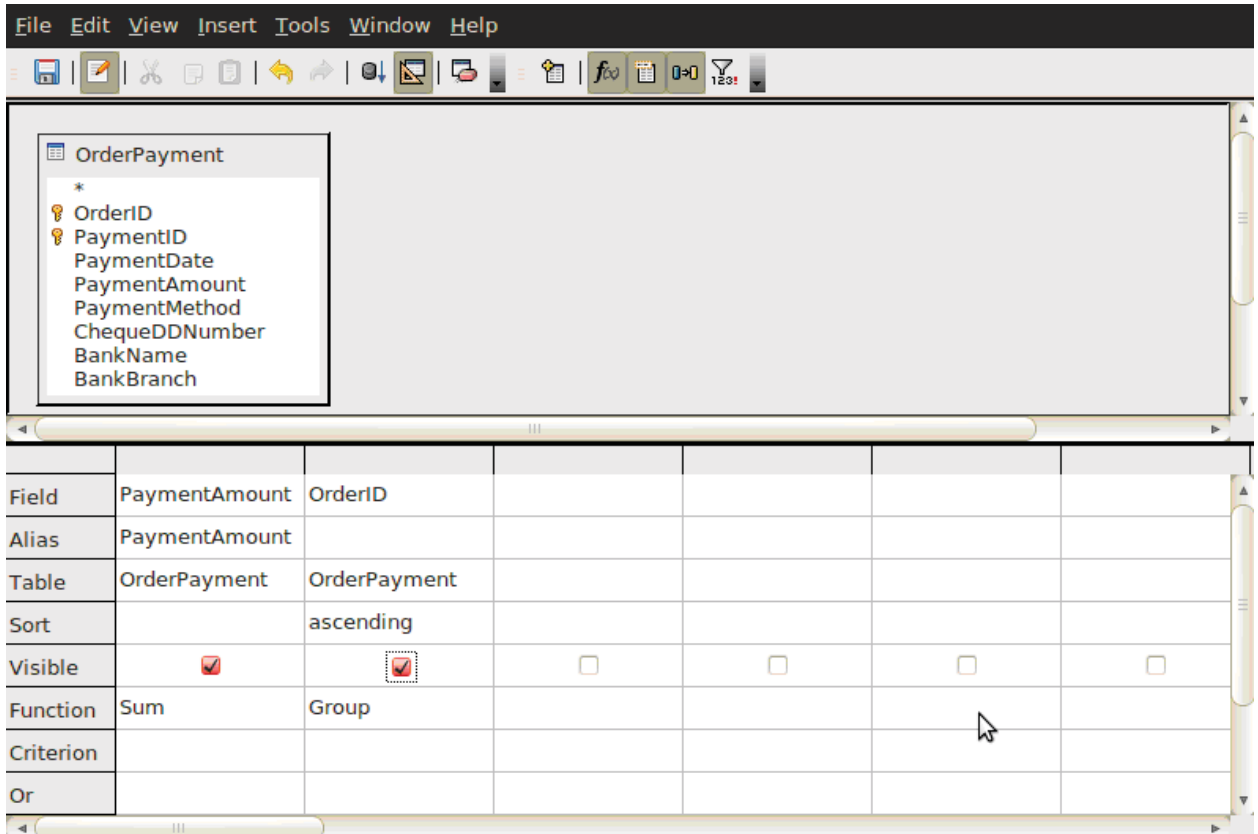
આકૃતિ 11.10 : ફિલ્ડ પર Groupby વિકલ્પનો ઉપયોગ

- Next બટન પર ક્લિક કરો.
- છઠ્ઠા પગલામાં વધુ તારવણી માટેની કેટલીક જૂથ આધારિત શરતો જરૂર હોય, તો ઉમેરી શકાય છે. જેમકે, એવા રેકોર્ડ પ્રદર્શિત કરવા છે, જેમાં ચુકવણીની રકમ ₹. 10,000થી વધુ હોય. અહીં આપણે વધારાની કોઈ તારવણીની ન કરવાની હોવાથી Next બટન પર ક્લિક કરો.
- સાતમા પગલામાં જો જરૂરી હોય, તો ઉપનામ (Aliases) ઉમેરી Next બટન પર ક્લિક કરો.
- આઠમા પગલામાં Query_OrderPayment નામ ઉમેરી *Modify Query* વિકલ્પ પસંદ કરો. અહીં આ વિકલ્પ પસંદ કરવો હિતાવહ છે, કારણકે જો ક્વેરીનો સીધો જ અમલ કરવામાં આવે, તો બેઝ દરેક ઓર્ડર માટેનું માત્ર એક જ PaymentAmount ફિલ્ડ દર્શાવશે. પરંતુ આપણને PaymentAmount સાથે OrderID ફિલ્ડની પણ અપેક્ષા છે.
- Finish બટન પર ક્લિક કરો. આકૃતિ 11.11માં દર્શાવ્યા મુજબ ક્વેરીને Design View દ્વારા રજૂ કરવામાં આવશે.




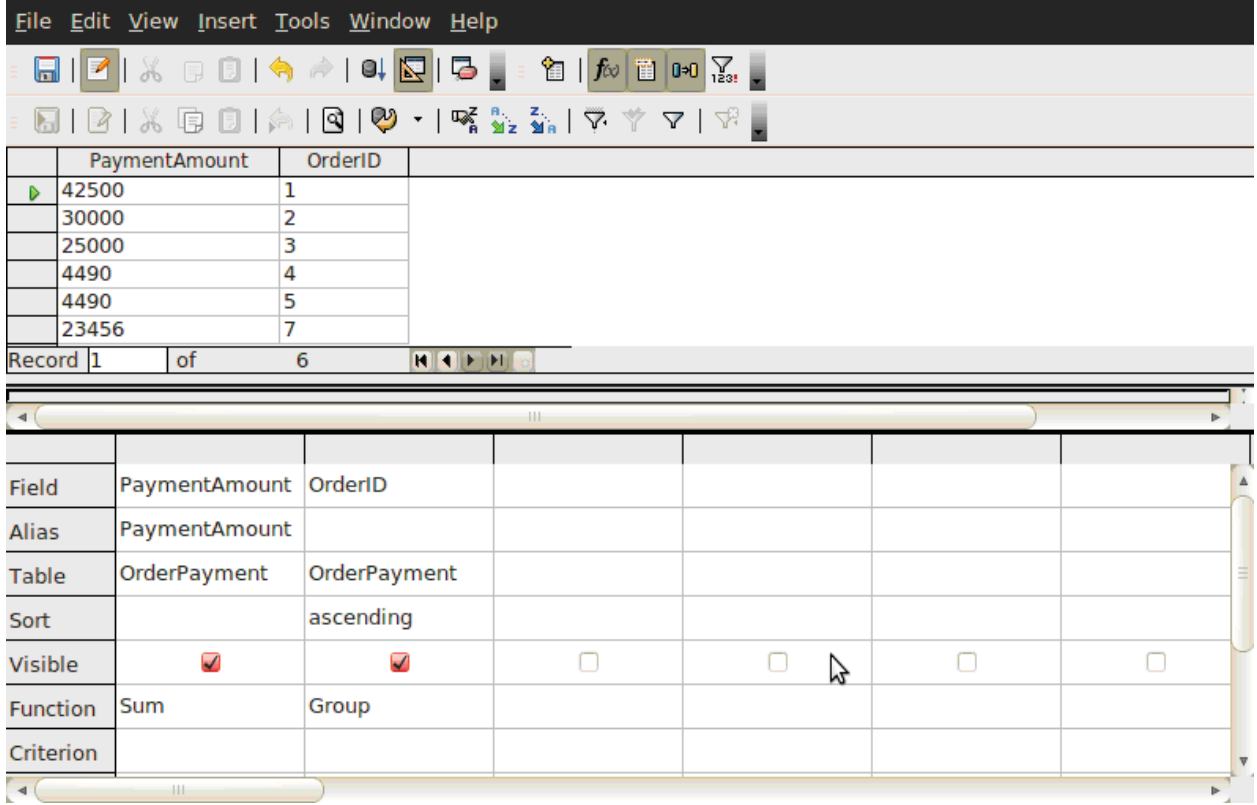
આકૃતિ 11.11 : ક્વેરીનો ડિઝાઇન વ્યૂ

- આકૃતિ 11.12માં દર્શાવ્યા મુજબ OrderID ફિલ્ડની નીચે આવેલું ચેકબોક્સ પસંદ કરો.



આકૃતિ 11.12 : દર્શાવવામાં આવનાર ફિલ્ડની પસંદગી

- *Run Query* બટન () બટન પર ક્લિક કરી ક્વેરીનો અમલ કરો. આકૃતિ 11.13 દર્શાવ્યા મુજબનું પરિણામ રજૂ કરવામાં આવશે.



The screenshot shows the Microsoft Access interface. The top part displays a query result table with the following data:

	PaymentAmount	OrderID
▶	42500	1
	30000	2
	25000	3
	4490	4
	4490	5
	23456	7

Record 1 of 6

The bottom part shows the design view table with the following structure:

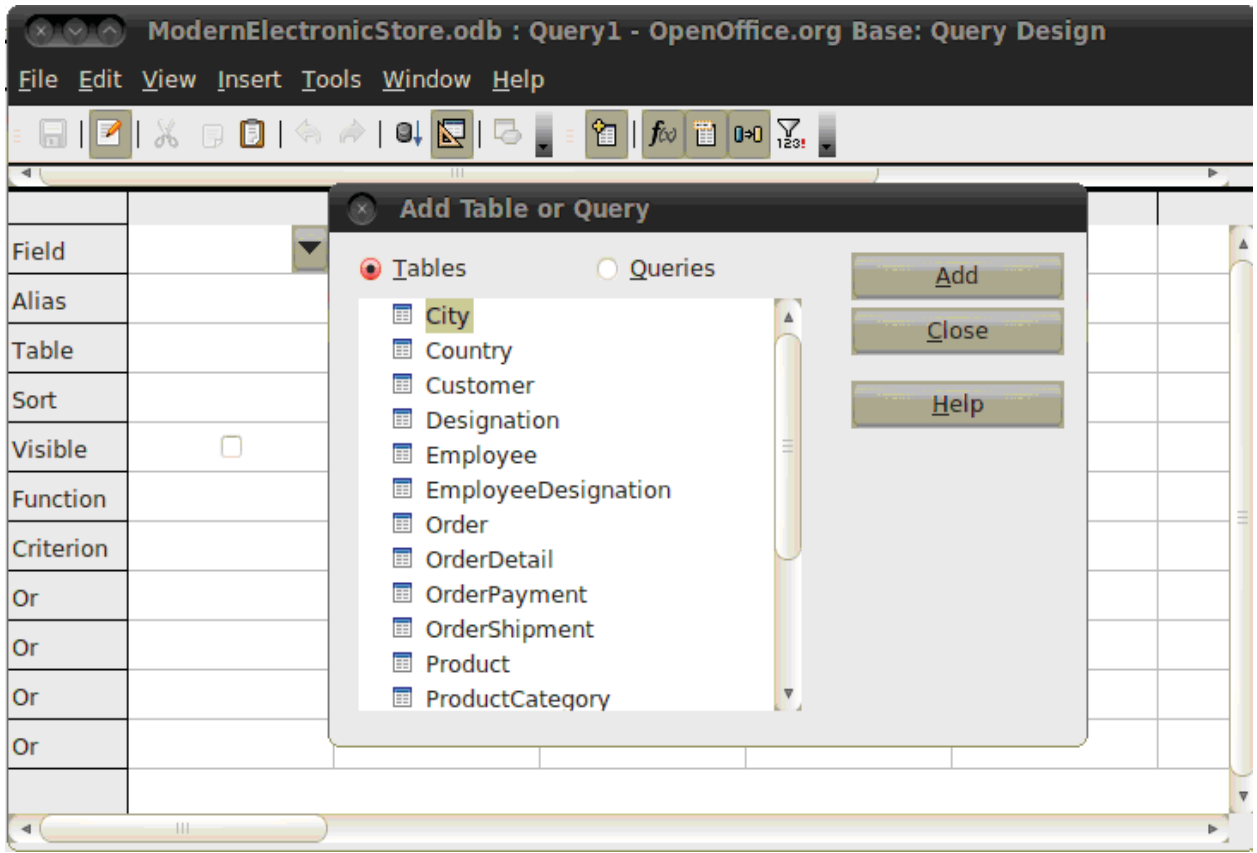
Field	PaymentAmount	OrderID				
Alias	PaymentAmount					
Table	OrderPayment	OrderPayment				
Sort		ascending				
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Function	Sum	Group				
Criterion						

આકૃતિ 11.13 : ક્વેરીનું પરિણામ


ડિઝાઇન વ્યૂનો ઉપયોગ કરી ક્વેરીની રચના (Creation of Query Using Design View)

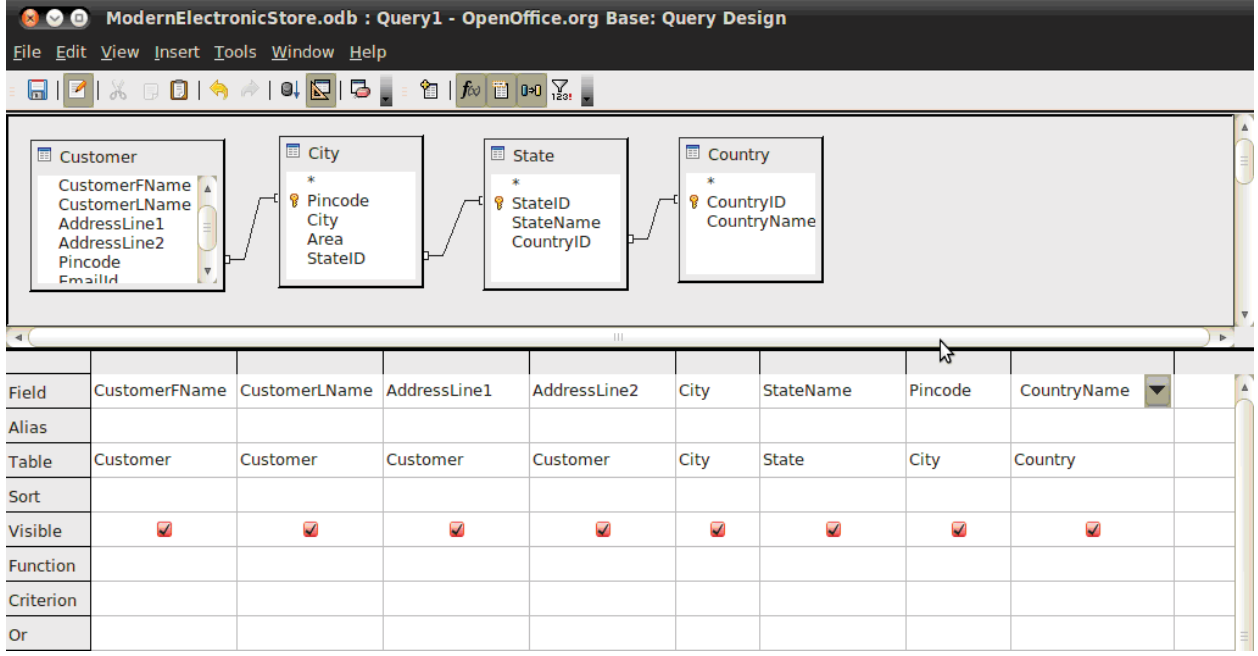
હવે આપણે ડિઝાઇન વ્યૂની મદદથી ક્વેરીની રચના કરવા માટેની રીતે વિશે ચર્ચા કરીએ. ગ્રાહકોનાં સરનામાંની યાદી મેળવવા માટેની ક્વેરીની રચના કરતી વખતે આપણે એ ઉલ્લેખ કર્યો હતો કે City, State અને Country ટેબલની વિગતો મેળવવી શક્ય નથી. આમ થવાનું કારણ એ કે વિઝાર્ડ દ્વારા ક્વેરી તૈયાર કરતી વખતે માત્ર એક જ ટેબલનો ઉપયોગ કરી શકવાની મર્યાદા રહે છે, માટે હવે આપણે તે કાર્ય ક્વેરી ડિઝાઇન વ્યૂનો ઉપયોગ કરી પૂરું કરીએ.

- ડેટાબેઝ વિન્ડોમાં *Queries* આઈકન પર ક્લિક કરો.
- ટાસ્કપેન પર આવેલા *Create Query in Design View...* વિકલ્પ પર ડબલ ક્લિક કરો.
- આકૃતિ 11.14માં દર્શાવ્યા પ્રમાણે *Add Table or Query* ડાયલોગબોક્સ જોઈ શકાશે.




આકૃતિ 11.14 : Add Table or Query ડાયલોગબોક્સ

- Customer ટેબલ પસંદ કરી Add બટન પર ક્લિક કરો.
- આ જ રીતે, City, State અને Country ટેબલ પસંદ કરો. આકૃતિ 11.15માં દર્શાવ્યા મુજબ ટેબલ પેન (Table pane)માં ચાર ટેબલ જોવા મળશે. બેઝ અગાઉ વ્યાખ્યાયિત કરવામાં આવેલ સંબંધ (relationship) પણ અહીં દર્શાવશે.
- Close બટન પર ક્લિક કરો. ફરી ટેબલ ઉમેરવાની જરૂર હોય, ત્યારે Query design ટૂલબાર પર આવેલા Add Table or Query બટન ()નો ઉપયોગ કરી Add Table or Query ડાયલોગબોક્સ ફરી ખોલી શકાય છે.
- Customer ટેબલમાં આવેલા CustomerFName, CustomerLName, AddressLine1, AddressLine2 ફિલ્ડ પર ડબલ ક્લિક કરો. આ જ રીતે City ટેબલમાંથી City અને Pincode ફિલ્ડ, State ટેબલમાંથી StateName ફિલ્ડ પસંદ કરો. આકૃતિ 11.15માં દર્શાવ્યા મુજબ ક્વેરીની ગ્રિડમાં આ તમામ ફિલ્ડનાં નામ તેને સંબંધિત ટેબલના નામ સાથે જોવા મળશે.



આકૃતિ 11.15 : ફિલ્ડની પસંદગી

- આ ઉપરાંત ક્વેરી ડિઝાઇન વિન્ડોમાં Alias, Sort, Visible, Function, Criterion અને Or જેવા શીર્ષક ધરાવતી અન્ય કેટલીક રો (રેકોર્ડ) પણ જોવા મળશે. અહીં એ જોઈ શકાય છે કે પૂર્વનિર્ધારિત રીતે તમામ ફિલ્ડના visible વિકલ્પને પસંદ કરવામાં આવ્યા છે. તે સૂચવે છે કે પસંદ કરેલા તમામ ફિલ્ડને પરિણામમાં દર્શાવવામાં આવશે.
- ફિલ્ડ માટેનાં અર્થપૂર્ણ નામ પસંદ કરવા માટે Aliasનો ઉપયોગ કરી શકાય છે. ઉદાહરણ તરીકે, ક્વેરીના પરિણામમાં CustomerFname સ્થાને *Name of Customer* શીર્ષક વધુ સ્વીકૃત હશે. CustomerFname કોલમની નીચે આપેલા Alias રો શીર્ષકના ટેક્સ્ટબોક્સમાં *Name of Customer* ટાઇપ કરો.
- ગ્રાહકોના રેકોર્ડને તેમનાં નામના ક્રમમાં દર્શાવવા માટે CustomerFname કોલમની નીચે આવેલા Sort શીર્ષક ધરાવતી રોના ડ્રોપડાઉન બોક્સમાં *ascending* વિકલ્પ પસંદ કરો. આ જ રીતે CustomerLName કોલમની નીચે આપેલા Sort વિકલ્પમાં *ascending* પસંદ કરો.
- ક્વેરી ડિઝાઇન ટૂલબારમાં આવેલા Run Query બટન () પર ક્લિક કરો. આકૃતિ 11.16માં દર્શાવ્યા મુજબનું ક્વેરીનું પરિણામ રજૂ કરવામાં આવશે.
- ભવિષ્યના ઉપયોગ માટે ક્વેરીનો સંગ્રહ કરવા માટે File મેનુમાંથી Save વિકલ્પ પસંદ કરો. વૈકલ્પિક રીતે Close બટન પર ક્લિક કરવાથી બેઝ Save ડાયલોગબોક્સ રજૂ કરશે.
- *CustomerAddresses* નામ આપી OK બટન પર ક્લિક કરો.

ModernElectronicStore.odb : Query1 - OpenOffice.org Base: Query Design

File Edit View Insert Tools Window Help

	Name of Customer	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
▶	Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	380058	India
	Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	362268	India
	Prema	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	380089	India
	Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	380056	India
	Sakina	Nagingar	T-37, Abhimanyu	Curry Road	Mehsan	Gujarat	384001	India

Record 1 of 5

Field	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Alias	Name of Customer							
Table	Customer	Customer	Customer	Customer	City	State	City	Country
Sort	ascending	ascending						
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function								
Criterion								
Or								
Or								

આકૃતિ 11.16 : ક્વેરીનું પરિણામ

પરિણામ જોઈ લીધા બાદ ક્વેરી-વિન્ડો બંધ કરો.

ક્વેરી સુધારવી (Editing a Query)

એક વાર ક્વેરી બનાવી લીધા પછી તેને સુધારવાની જરૂર હોઈ શકે છે. ઉદાહરણ તરીકે, ઉપર બનાવેલી ક્વેરીમાં CustomerLName ફિલ્ડના ઉપનામ તરીકે *Surname* ઉમેરવાની જરૂર લાગે.

આ ફેરફાર કરવા માટે નીચે જણાવેલ પગલાંને અનુસરો :

- *Queries* આઈકન પર ક્લિક કરો. CustomerAddresses ક્વેરી પર રાઈટ ક્લિક કરી *Edit* વિકલ્પ પસંદ કરો. આમ કરવાથી ક્વેરી માટે Design View રજૂ કરવામાં આવશે.
- CustomerLNameની નીચે આવેલા *Alias* : શીર્ષક ધરાવતી રોના ટેક્સ્ટબોક્સમાં *Surname* ટાઈપ કરો.
- ક્વેરી રન કરો.

માપદંડ ઉમેરવા (Applying Criteria)

ટેબલના પસંદ કરેલ ફિલ્ડની વિગતો દર્શાવતી ક્વેરી કેવી રીતે બનાવી શકાય તે આપણે શીખ્યા. હવે ધારો કે આપણે તમામ રેકોર્ડને બદલે માત્ર અમદાવાદ શહેરમાં વસતા ગ્રાહકોના રેકોર્ડ જોવા ઇચ્છીએ છીએ. આનો અર્થ એ થયો કે આપણે પસંદ કરવામાં આવેલ રેકોર્ડનો પેટા સમૂહ બેજ દ્વારા દર્શાવવા માંગીએ છીએ. આમ કરવા માટે આપણે એવો માપદંડ ઉમેરવાની જરૂર પડે કે જેથી City ફિલ્ડમાં “Ahmedabad” કિંમત હોય તેવા જ રેકોર્ડ ક્વેરીના પરિણામમાં પસંદ કરવામાં આવે.

એક જ ફિલ્ડનો ઉપયોગ (Using Single Field)

- CustomerAddresses ફિલ્ડ પર રાઈટ ક્લિક કરો.
- ક્વેરી ડિઝાઈન વ્યૂમાં ખોલવા માટે Edit વિકલ્પ પર ક્લિક કરો.
- આકૃતિ 11.17માં દર્શાવ્યા પ્રમાણે City ફિલ્ડના Criterion સેલમાં 'Ahmedabad' ટાઈપ કરો. અહીં એ યાદ રાખવું જરૂરી છે કે લખાણને ક્વોટેશન (' ') નિશાની દ્વારા, તારીખને # (હેશ) નિશાની દ્વારા આવરીને લખવું જરૂરી છે, જ્યારે આંકડાકીય વિગતોને કોઈ નિશાનીની જરૂર નથી. અલબત્ત, આ નિશાનીઓ ઉમેરવાનું રહી જશે, તો બેઝ ભૂલનો સંદેશ દર્શાવશે નહીં, પરંતુ જાતે જ જે-તે નિશાની ઉમેરી લેશે.
- ક્વેરીનો સંગ્રહ કરી રન કરવાથી ઈચ્છિત પરિણામ જોઈ શકાશે.

Field	CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	Pincode	CountryName
Alias	Name of Customer							
Table	Customer	Customer	Customer	Customer	City	State	City	Country
Sort	ascending	ascending						
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function								
Criterion					'Ahmedabad'			
Or								
Or								

આકૃતિ 11.17 : માપદંડ ગોઠવવા

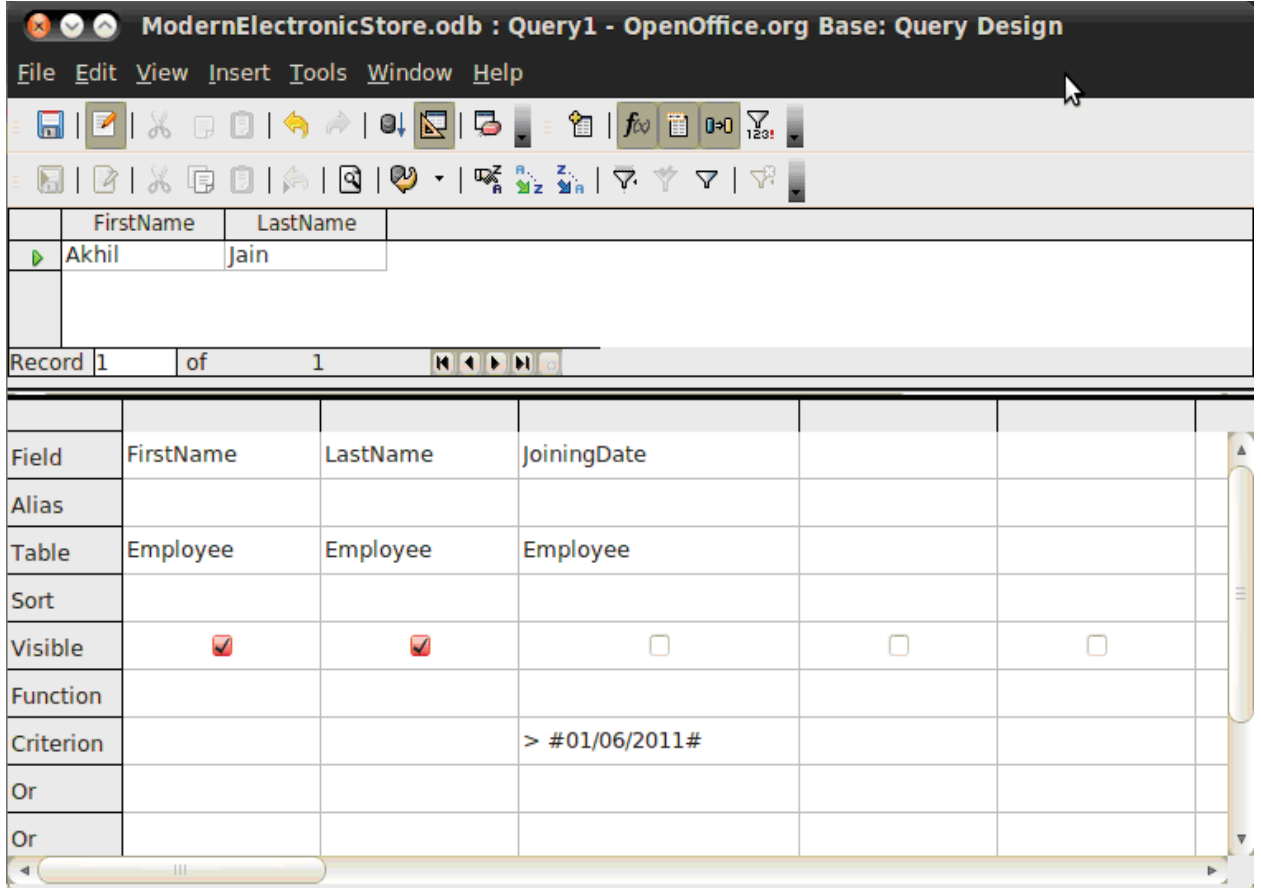
આકૃતિ 11.17માં દર્શાવ્યા મુજબ અચળ કિંમતોનો ઉપયોગ કરવા ઉપરાંત જુદા-જુદા ઓપરેટરો દ્વારા બનાવવામાં આવેલાં સમીકરણોનો ઉપયોગ કરવાની સુવિધા પણ બેઝ પૂરી પાડે છે. કોષ્ટક 11.1માં ઓપરેટરોની યાદી દર્શાવી છે.

Operator	Symbols
Comparison	=, >, <, >=, <=, <>
Logical	And, Or, Not
Special	Like, Is, Between, In

કોષ્ટક 11.1 : બેઝમાં ઉપયોગમાં લઈ શકાય તેવા ઓપરેટરો

ધારો કે, આપણે પહેલી જૂન, 2011 પછી જોડાયેલા કર્મચારીઓની યાદી જોવા માંગીએ છીએ. આ માટે ડિઝાઇન વ્યૂમાં નવી ક્વેરી બનાવીએ. Employee ટેબલને ક્વેરીમાં ઉમેરો. ત્યાર બાદ FirstName, LastName અને JoiningDate ફિલ્ડ પસંદ કરો. JoiningDate ફિલ્ડના *Criterion* સેલમાં “> #01/06/2011#” ટાઇપ કરો. JoiningDate ફિલ્ડના *Visible* સેલમાં આવેલ ચેકબોક્સને નાપસંદ કરો. હવે ક્વેરીનો સંગ્રહ કરી તેને રન કરો. આકૃતિ 11.18માં દર્શાવ્યા મુજબનું પરિણામ જોવા મળશે. પરિણામમાં JoiningDate ફિલ્ડ દર્શાવવામાં આવ્યું નથી, તેની નોંધ કરો.

નોંધ : જો ક્વેરીના પરિણામમાં તેમાં ઉપયોગમાં લેવામાં આવેલા ફિલ્ડની કિંમત દર્શાવવી ન હોય, તો તે ફિલ્ડની *Visible* લાક્ષણિકતા સાથે રહેલ ચેકબોક્સ નાપસંદ કરવામાં આવેલ છે.



આકૃતિ 11.18 : તારીખના ફિલ્ડમાં માપદંડ ઉમેરવો

આ જ રીતે પહેલી જૂન, 2005 અને પહેલી નવેમ્બર, 2012 વચ્ચે જોડાયા હોય તેવા કર્મચારીઓની વિગતો દર્શાવવા માટે JoiningDate ફિલ્ડના *Criterion* સેલમાં “>= # 1/6/2005 # And <= # 11/1/2012 #” માપદંડ ઉમેરી શકાય. આ જ માપદંડ ઉમેરવા માટે બેઝમાં *Between* ઓપરેટર પણ ઉપલબ્ધ છે, જે આકૃતિ 11.19માં દર્શાવ્યો છે.

ModernElectronicStore.odt : Query1 - OpenOffice.org Base: Query Design

File Edit View Insert Tools Window Help

FirstName	LastName	JoiningDate
Anya	Gangele	01/05/11
Robina	Arora	02/06/06
Akhil	Jain	12/31/11

Record 1 of 3

Field	FirstName	LastName	JoiningDate
Alias			
Table	Employee	Employee	Employee
Sort			
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function			
Criterion			Between #1/6/2005 # And # 11/1/2012 #
Or			

આકૃતિ 11.19 : Between ઓપરેટરનો ઉપયોગ

હવે, ધારો કે, આપણે અમદાવાદ અને પાટણ શહેરમાં રહેતા ગ્રાહકોને વળતરની કૂપનો મોકલવા ઈચ્છીએ છીએ. આ માટે આપણને અમદાવાદ અથવા પાટણમાં રહેતા ગ્રાહકોની યાદીની જરૂર પડશે. ડિઝાઇન વ્યૂની મદદથી એક નવી ક્વેરી બનાવો. તેમાં Customer, City અને State ટેબલ ઉમેરો. ત્યાર પછી CustomerFName અને CustomerLName, AddressLine1, AddressLine2, City, StateName અને CardHolder ફિલ્ડ આકૃતિ 11.20માં દર્શાવ્યા મુજબ ઉમેરો. City ફિલ્ડમાં દર્શાવ્યા મુજબ માપદંડ ઉમેરો. હવે ક્વેરીને સેવ કરી રન કરો. આકૃતિ 11.20માં દર્શાવ્યા પ્રમાણેનું પરિણામ જોવા મળશે.

File Edit View Insert Tools Window Help

CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	<input type="checkbox"/>
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	<input type="checkbox"/>
Purna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	<input type="checkbox"/>
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	<input checked="" type="checkbox"/>

Record 1 of 4

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Alias						
Table	Customer	Customer	Customer	City	State	Customer
Sort						
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function						
Criterion				'Ahmedabad'		
Or				'Patan'		
Or						

આકૃતિ 11.20 : એક જ ફિલ્ડ પર એકથી વધુ માપદંડનો ઉપયોગ

ઉપર દર્શાવેલ માપદંડને *IN* ઓપરેટરની મદદથી પણ ઉમેરી શકાય. *City* ફિલ્ડની *Criterion* હરોળમાં *IN* ('Ahmedabad'; 'Patan') ટાઈપ કરવાથી પણ આ જ પરિણામ મેળવી શકાય. અમદાવાદ અને પાટણ સિવાયનાં શહેરોમાં રહેતા ગ્રાહકોના રેકોર્ડ મેળવવા માટે *Criterion* હરોળમાં *NOT IN* ('Ahmedabad'; 'Patan') માપદંડ ઉમેરી શકાય.

એકથી વધુ ફિલ્ડનો ઉપયોગ

હવે ધારો કે આપણે અમદાવાદ અને પાટણ શહેરમાં વસતા અને મોડર્ન ઇલેક્ટ્રોનિક્સનાં કાર્ડ ધરાવતા ગ્રાહકોને કૂપનો મોકલવા ઇચ્છીએ છીએ. આના ઉકેલ માટે *AND* ઓપરેટરનો ઉપયોગ કરવો પડશે. આ માટે *City* ફિલ્ડની ક્રિમત અમદાવાદ કે પાટણ રાખવાની છે તથા *CardHolder* ફિલ્ડ ખાલી ન હોવું જોઈએ. આકૃતિ 11.21માં દર્શાવ્યા પ્રમાણેનો માપદંડ ટાઈપ કરો. અહીં *NULL* (*CardHolder* ફિલ્ડમાં ચેકમાર્ક નહીં) ક્રિમત ફિલ્ડ ખાલી છે, તેમ દર્શાવે છે.

The screenshot shows a query design window for 'ModernElectronicStore.odb'. The table below the grid contains the following data:

CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	<input type="checkbox"/>
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	<input type="checkbox"/>
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	<input type="checkbox"/>
Harshit	Shah	17, jaldeep-1	Bopal	Ahmeda	Gujarat	<input checked="" type="checkbox"/>

The query design grid below the table is as follows:

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Alias						
Table	Customer	Customer	Customer	City	State	Customer
Sort						
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function						
Criterion				IN ('Ahmedabad'; 'Patan')		Is Not Empty
Or						
Or						

આકૃતિ 11.21 : એકથી વધુ ફિલ્ડ પર માપદંડનો ઉપયોગ

હવે, જો આપણે એવા ગ્રાહકોના રેકોર્ડ દર્શાવવા માંગતા હોઈએ કે જેઓ અમદાવાદ કે પાટણના નિવાસી હોય અથવા સભ્યપદનું કાર્ડ ધરાવતા હોય, તો આ માટે *City* ફિલ્ડની *Criterion* હરોળમાં *IN* ('Ahmedabad'; 'Patan') ટાઈપ કરવું જોઈએ. સાથે-સાથે *CardHolder* કોલમની *Or* હરોળમાં *Is Not Empty* ઉમેરવું જોઈએ (જુઓ આકૃતિ 11.22).

આ પહેલાંની ક્વેરી સાથે હાલની ક્વેરીને સરખાવો. પહેલાંની ક્વેરીમાં બે ફિલ્ડ વચ્ચે *AND* ઓપરેટરનો ઉપયોગ કરવાનો હોવાથી બન્ને માપદંડોને એક જ આડી હરોળમાં લખવામાં આવ્યા છે. જ્યારે હાલની ક્વેરીમાં *OR* ઓપરેટરના ઉપયોગ

માટે બે જુદી હરોળોમાં ફિલ્ડના માપદંડ ઉમેરવામાં આવ્યા છે. આકૃતિ 11.22માં દર્શાવેલા પરિણામમાં મહેસાણા શહેર ધરાવતા એક નવા રેકોર્ડની નોંધ કરો, જે વધારાનો રેકોર્ડ છે, કારણકે તે ગ્રાહક પાસે પણ સભ્યપદનું કાર્ડ છે.

CustomerFName	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Puja	Sethia	17, Akash	Jodhpur Char Rast	Ahmeda	Gujarat	<input type="checkbox"/>
Sakina	Nagingar	T-37, Abhimanyu	Curry Road	Mehsan	Gujarat	<input checked="" type="checkbox"/>
Joseph	Thomas	BG-1, Sunstar	Sola Road	Patan	Gujarat	<input type="checkbox"/>
Prerna	Agrawal	Smita Bunglows	PNT Road	Ahmeda	Gujarat	<input type="checkbox"/>
Harshit	Shah	17, Jaldeep-1	Bopal	Ahmeda	Gujarat	<input checked="" type="checkbox"/>

Field	CustomerLName	AddressLine1	AddressLine2	City	StateName	CardHolder
Alias						
Table	Customer	Customer	Customer	City	State	Customer
Sort						
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Function						
Criterion				IN ('Ahmedabad'; 'Patan')		
Or						Is Not Empty
Or						

આકૃતિ 11.22 : એકથી વધુ ફિલ્ડ માટે OR માપદંડનો ઉપયોગ

વાઈલ્ડકાર્ડનો ઉપયોગ (Using Wild Cards)

ધારો કે આપણે એવાં ઉત્પાદનોની યાદી દર્શાવવી છે, જેમનાં મોડેલ નામ *hp* અક્ષરોથી શરૂ કરવામાં આવ્યા હોય. આ માટે Product ટેબલનો ઉપયોગ કરી નવી ક્વેરી બનાવો. આ માટે PCode, ModelName, SellingPrice અને OSSupport ફિલ્ડ પસંદ કરો. આકૃતિ 11.23માં દર્શાવ્યા મુજબ માપદંડ ઉમેરો.

Pcode	ModelName	SellingPrice	OSSupport
P00000002	hp ENVY Sleekbook 6t 1000	35000	Windows
P00000003	hp ENVY Sleekbook 6t 1000	30000	Linux

Field	Pcode	ModelName	SellingPrice	OSSupport
Alias				
Table	Product	Product	Product	Product
Sort				
Visible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Function				
Criterion		LIKE 'hp*'		
Or				

આકૃતિ 11.23 : વાઈલ્ડકાર્ડ

આકૃતિ 11.23માં ModelName ફિલ્ડના *Criterion* સેલમાં ઉપયોગમાં લેવામાં આવેલ ફૂદડી (*) ની નિશાનીને વાઈલ્ડ-કાર્ડ અક્ષર તરીકે ઓળખવામાં આવે છે. અક્ષર કે અક્ષરોના સમૂહને રજૂ કરતી નિશાનીને વાઈલ્ડકાર્ડ તરીકે ઓળખવામાં આવે છે. આમ hp* શબ્દો પ્રથમ બે અક્ષરો hp અને તેની પાછળ અક્ષરોના કોઈ પણ સમૂહને રજૂ કરે છે. આ જ રીતે Like *hp માપદંડ એવાં ઉત્પાદનોનાં નામ દર્શાવશે, જેનાં નામ hp અક્ષરોથી પૂરાં થતાં હોય અને Like hp*hp એવા ઉત્પાદનોનાં નામ દર્શાવશે, જેમનાં નામ hp અક્ષરોથી શરૂ અને પૂરાં થતાં હોય. અહીં એ નોંધ કરવી જરૂરી છે કે વાઈલ્ડકાર્ડ અક્ષરો સાથે Like ઓપરેટરનો ઉપયોગ કરવો ફરજિયાત છે.

ગણતરીઓ કરવી (Performing Calculation)

બેઝમાં ગણતરીઓ કરવા માટે નીચે જણાવેલ પૈકી કોઈ પણ એક વિકલ્પનો ઉપયોગ કરી શકાય :

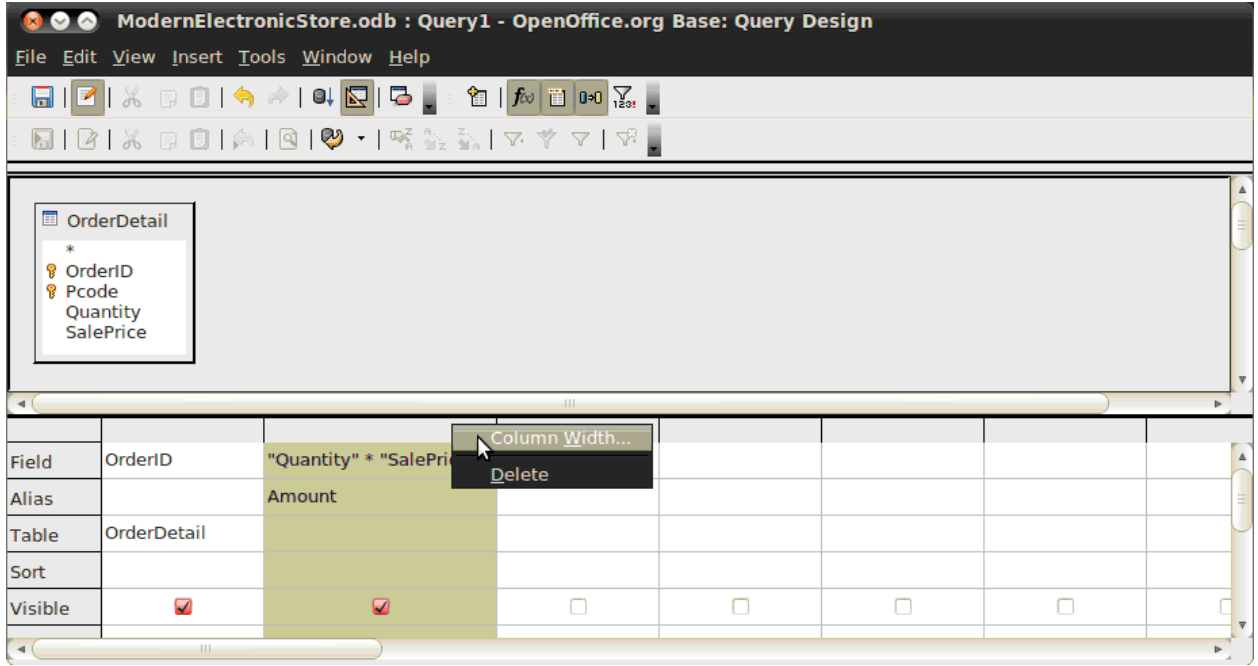
- કસ્ટમ-ગણતરીઓ (Custom calculation)
- પૂર્વવ્યાખ્યાયિત ગણતરીઓ (Predefined calculation)

કસ્ટમ-ગણતરીઓ (Custom calculation) : એક કે વધુ ફિલ્ડમાં આવેલ વિગતોનો ઉપયોગ કરી દરેક રેકોર્ડ પર આંકડાકીય, તારીખ સંબંધિત કે લખાણ માટેની ગણતરીઓ કરવા માટે કસ્ટમ-ગણતરીનો ઉપયોગ કરવામાં આવે છે. આ ગણતરીઓમાં બે કે વધુ ફિલ્ડમાં આવેલ કિંમતોનાં સરવાળા, બાદબાકી, ગુણાકાર કે ભાગાકારનો સમાવેશ પણ કરી શકાય છે. કસ્ટમ-ગણતરીઓ માટે એક વધારાના ફિલ્ડ - Custom field ને ઉમેરવાની જરૂર પડે છે.

આ બાબત એક ઉદાહરણ દ્વારા સમજાવે. OrderDetail ટેબલ જુઓ. તેમાં ગ્રાહકે ખરીદેલી દરેક ઉત્પાદનની કુલ રકમ શોધવા માટેની ગણતરી કરીએ. Quantityની વિગતનો SalePriceની વિગત સાથે ગુણાકાર કરવાથી Amount મળશે. Amountની કિંમતનો સંગ્રહ કરતા આ નવા ફિલ્ડને ગણતરીનું ફિલ્ડ (calculated field) કહે છે.

દરેક ગ્રાહકે દરેક મહિનામાં ચૂકવેલી કુલ રકમની ગણતરી કરવા માટે નીચેનાં પગલાં અનુસરો :

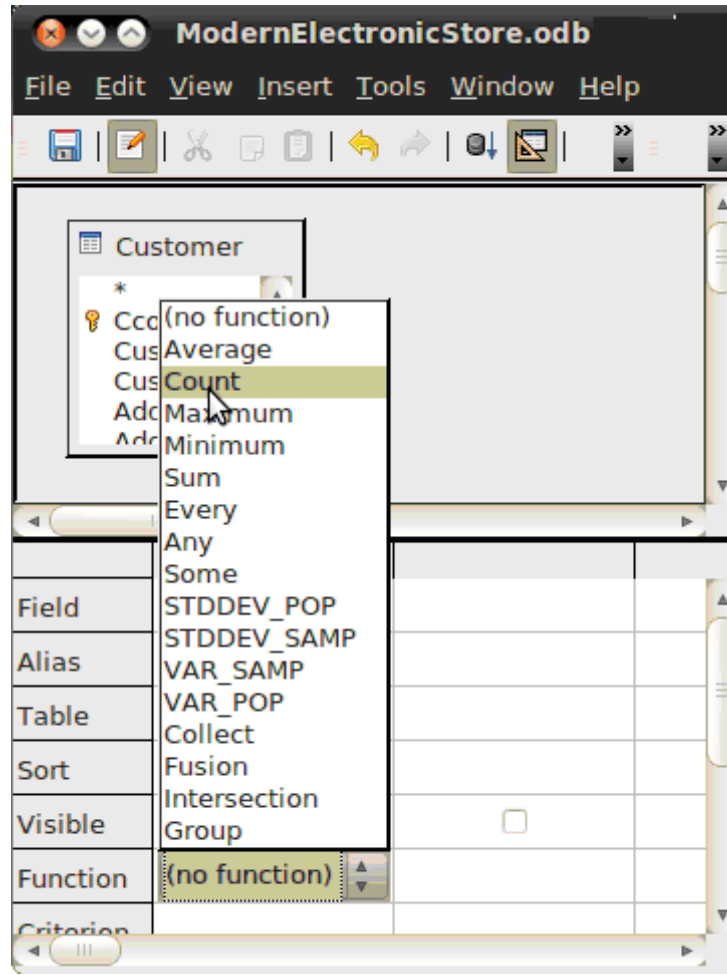
- ડિઝાઇન વ્યૂનો ઉપયોગ કરી નવી ક્વેરી બનાવો.
- *Add Table or Query* ડાયલોગબોક્સમાંથી OrderDetail ટેબલ પસંદ કરો.
- OrderID ફિલ્ડ પર ડબલ ક્લિક કરી, તેને ક્વેરી ગ્રિડમાં ઉમેરો.
- ક્વેરી-ડિઝાઇન ગ્રિડની બીજી કોલમની *Field* હરોળમાં Quantity *SalePrice સમીકરણ ટાઈપ કરો. *Field* હરોળની પહોળાઈ હાલમાં પૂરતી ન હોવાના કારણે આખું સમીકરણ જોઈ ન શકાય એમ પણ બને.
- બીજી કોલમની *Field* હરોળમાં જમણી બાજુની ક્લિક કરો. આમ કરવાથી આકૃતિ 11.24માં દર્શાવ્યા મુજબ પોપઅપ મેનુ ખૂલશે. તેમાંથી *Column Width* પસંદ કરો. *Field* હરોળના લખાણને દૃશ્યમાન બનાવવા માટે તેમાં 4.20 સેમી કિંમત ઉમેરો.
- બીજી કોલમની *Alias* હરોળમાં *Amount* ટાઈપ કરો (જુઓ આકૃતિ 11.24).



આકૃતિ 11.24 : ગણતરી માટેના ફિલ્ડનો ઉપયોગ

પૂર્વવ્યાખ્યાયિત ગણતરીઓ - વિગતોનું તારણ (Predefined Calculations - Summarizing the data) : ફિલ્ડ ઉપર કેવી રીતે ગણતરીઓ કરવી તે આપણે જોયું. ફિલ્ડને બદલે આપણને ઘણીવાર રેકોર્ડનાં જૂથ પર ગણતરીઓ કરવાની જરૂર પડે. ઉદાહરણ તરીકે, ઉત્પાદનોની કુલ સંખ્યા ગણવા કે દરેક ગ્રાહકે ખર્ચેલી સરેરાશ રકમની ગણતરી કરવા માટે સારાંશની ગણતરી (Summary Calculation) કરવાની જરૂર પડે. રેકોર્ડના જૂથ પર સરવાળો (Sum), સરેરાશ (Average), ગણન (Count), ન્યૂનતમ (minimum), મહત્તમ (maximum), પ્રમાણિત વિચલન (standard deviation) કે ચલનાંક (variance) ગણવા માટેની પૂર્વવ્યાખ્યાયિત ગણતરીઓ બેઝ દ્વારા પૂરી પાડવામાં આવે છે. આ ગણતરીઓ કસ્ટમ-ગણતરીથી જુદી પડે છે. કારણકે તેને ટેબલના એકથી વધુ રેકોર્ડ પર લાગુ કરવામાં આવે છે અને પરિણામસ્વરૂપે તે એક કિંમત દર્શાવે છે. મોર્ડન ઇલેક્ટ્રોનિક સ્ટોરના કુલ ગ્રાહકોની સંખ્યા શોધવા માટેની ક્વેરી બનાવવા નીચેનાં પગલાં અનુસરો :

- *Queries* આઈકન પર ક્લિક કરી *Create Query in Design View* પસંદ કરો.
- *Add Table or Query* ડાયલોગબોક્સમાંથી *Customer* ટેબલ પસંદ કરો.
- *Ccode* ફિલ્ડ પર ડબલ ક્લિક કરો.
- *Alias* હરોળમાં 'Total Customers' ટાઈપ કરો.
- ક્વેરી ડિઝાઈન ગ્રિડની *Function* નામની હરોળમાં આવેલ ડ્રોપડાઉન મેનુ ખોલો.
- આકૃતિ 11.25માં દર્શાવ્યા મુજબ એગ્રિગેટ વિધેયોની યાદીમાંથી *Count* વિધેય પસંદ કરો.
- ક્વેરી રન કરવાથી મોર્ડન ઇલેક્ટ્રોનિક સ્ટોરના કુલ ગ્રાહકોની સંખ્યા મેળવી શકાશે.



આકૃતિ 11.25 : એગ્રિગેટ વિધેયનો ઉપયોગ

અહીં એ યાદ રાખવું જરૂરી છે કે ફિલ્ડમાં આવેલું ગણતરીનું પરિણામ માત્ર દર્શાવવા માટેનું છે. આ પરિણામનો સંગ્રહ સંબંધિત ટેબલમાં કરવામાં આવતો નથી. તેને બદલે જ્યારે-જ્યારે ક્વેરી રન કરવામાં આવે છે, ત્યારે બેઝ ફેરગણતરી કરે છે. આમ, આ પ્રકારની ક્વેરીએ આપેલું પરિણામ હંમેશા ડેટાબેઝમાં આવેલી અદ્યતન વિગતો પર આધારિત હોય છે.

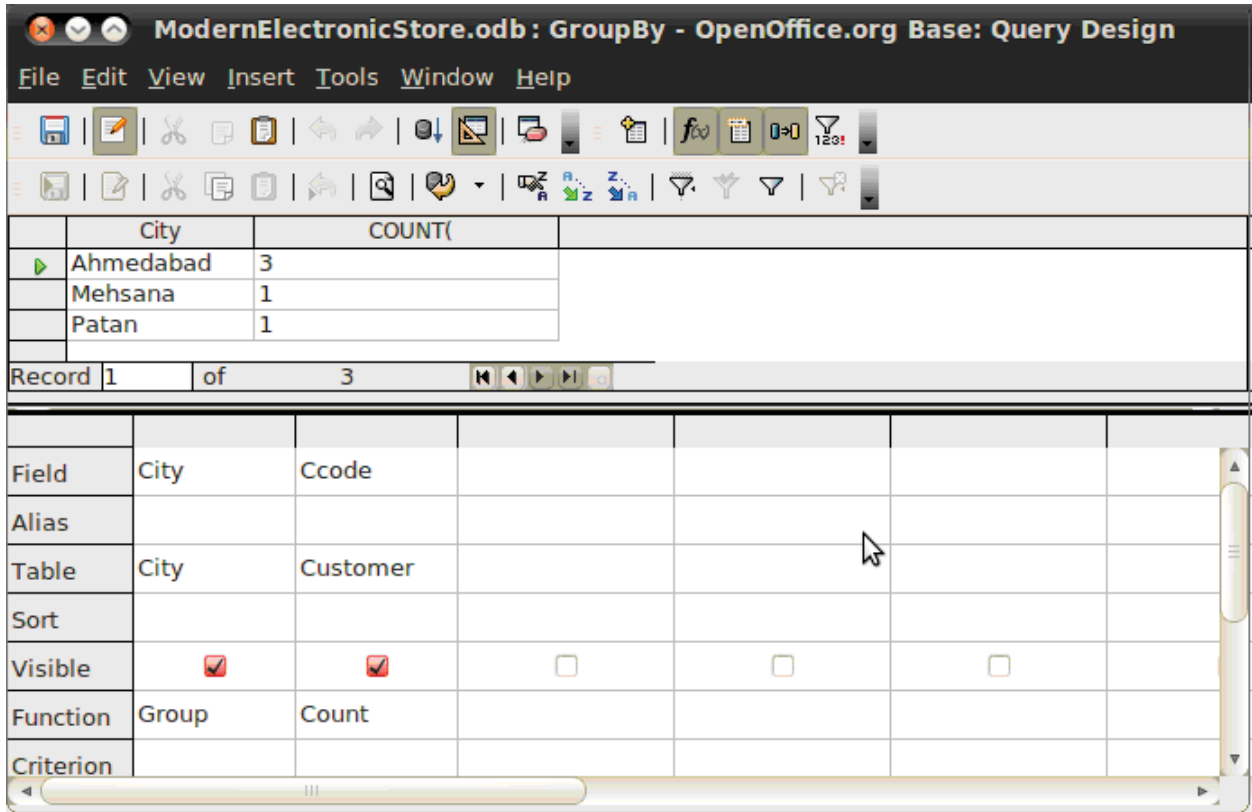
વિગતોનાં જૂથ બનાવવાં (Grouping the Data)

અમદાવાદ કે પાટણ શહેરમાં વસતા ગ્રાહકોની યાદી મેળવવા માટેની રીત આપણે અગાઉ જોઈ ગયા. અહીં આપણે વળતરની કૂપનો મોકલવા માટે ગ્રાહકોની વિગતો પૂરી પાડી હતી. હવે વ્યવસ્થાપક સમક્ષ એક અન્ય પ્રશ્ન છે, “શું દરેક શહેરમાં વસતા ગ્રાહકોની કુલ સંખ્યા મેળવી શકાય ?” આ સરવાળો Customer ટેબલમાં આવેલ દરેક શહેર માટે હોવો જોઈએ. તમે વિચારી શકો કે, હા, CustomerFName ફિલ્ડ પર સરવાળાને લાગુ પાડવાથી તે મેળવી શકાય. આ માટેનો પ્રયત્ન કરો અને પરિણામ જુઓ. તમને લાગશે કે તમે ખોટા છો. પરિણામસ્વરૂપે ગણવામાં આવેલો સરવાળો તે અમદાવાદ અને પાટણમાં વસતા ગ્રાહકોની સંખ્યાનો છે. સારું, તો તમે કહેશો કે Count વિધેયનો ઉપયોગ City ફિલ્ડ પર કરવો જોઈએ. ફરી પ્રયત્ન કરો અને ફરી તમને ખોટું પરિણામ જ મળશે.

કાગળ અને પેન લો. હવે વિચારવા લાગો કે આ શક્ય બનાવવા શું કરવું જોઈએ. ગ્રાહકોનાં શહેરને દર્શાવો. તે અમદાવાદ, મહેસાણા કે પાટણ હોઈ શકે છે. અમદાવાદના ગ્રાહકોનું જૂથ બનાવો, પાટણના ગ્રાહકોનું જૂથ બનાવો અને એ રીતે દરેક શહેરના ગ્રાહકોનું જૂથ બનાવો. હવે દરેક જૂથ પર સરવાળો લાગુ પાડો અને વ્યવસ્થાપકને જરૂર છે તેવું પરિણામ

મેળવી શકશો. તમને લાગશે કે આ ખરેખર જટિલ છે. પરંતુ બેઝ સાથે કાર્ય કરતી વખતે આ અત્યંત સરળ છે. નીચેનાં પગલાંને અનુસરો :

- ડિઝાઇન વ્યૂમાં નવી ક્વેરીની રચના કરો.
- *Add Table or Query* ડાયલોગબોક્સમાંથી Customer ટેબલ અને City ટેબલ ઉમેરો.
- City ટેબલના City ફિલ્ડ પર ડબલ ક્લિક કરો. આ જ રીતે Customer ટેબલમાંથી Ccode ફિલ્ડ પસંદ કરો.
- આકૃતિ 11.26માં દર્શાવ્યા મુજબ City ફિલ્ડની Function હરોળમાંથી Group પસંદ કરો.
- Ccode ફિલ્ડની Function હરોળમાં Count પસંદ કરો.
- ક્વેરી રન કરવાથી અપેક્ષિત પરિણામ દર્શાવી શકાશે.



આકૃતિ 11.26 : પરિણામના તારણ માટે રેકૉર્ડનાં જૂથ બનાવવાં

પેરામીટર ક્વેરી (Parameter Query) : ક્વેરી રન કરતી વખતે ઉપયોગકર્તા પાસેથી કિંમતો મેળવવા માટે પેરામીટર ક્વેરીની રચના કરવામાં આવે છે. અત્યાર સુધી બનાવવામાં આવેલી ક્વેરીમાં ચોક્કસ માપદંડોનો ઉપયોગ કરવામાં આવ્યો હતો. વ્યાખ્યાયિત કરવામાં આવેલા માપદંડને દરેક અમલ વખતે બદલવામાં આવતા નથી, પરંતુ ક્વેરીનું પરિણામ ટેબલમાં આવેલી વર્તમાન વિગતો અનુસાર બદલાતું રહેશે. સામાન્ય રીતે જ્યારે પેરામીટર ક્વેરી રન કરવામાં આવે છે, ત્યારે તે પેરામીટરની કિંમતોની માંગણી કરતું એક ડાયલોગબોક્સ રજૂ કરે છે. ત્યાર બાદ આ કિંમતોનો ઉપયોગ વિગતો મેળવવા માટેના માપદંડ તરીકે કરવામાં આવે છે.

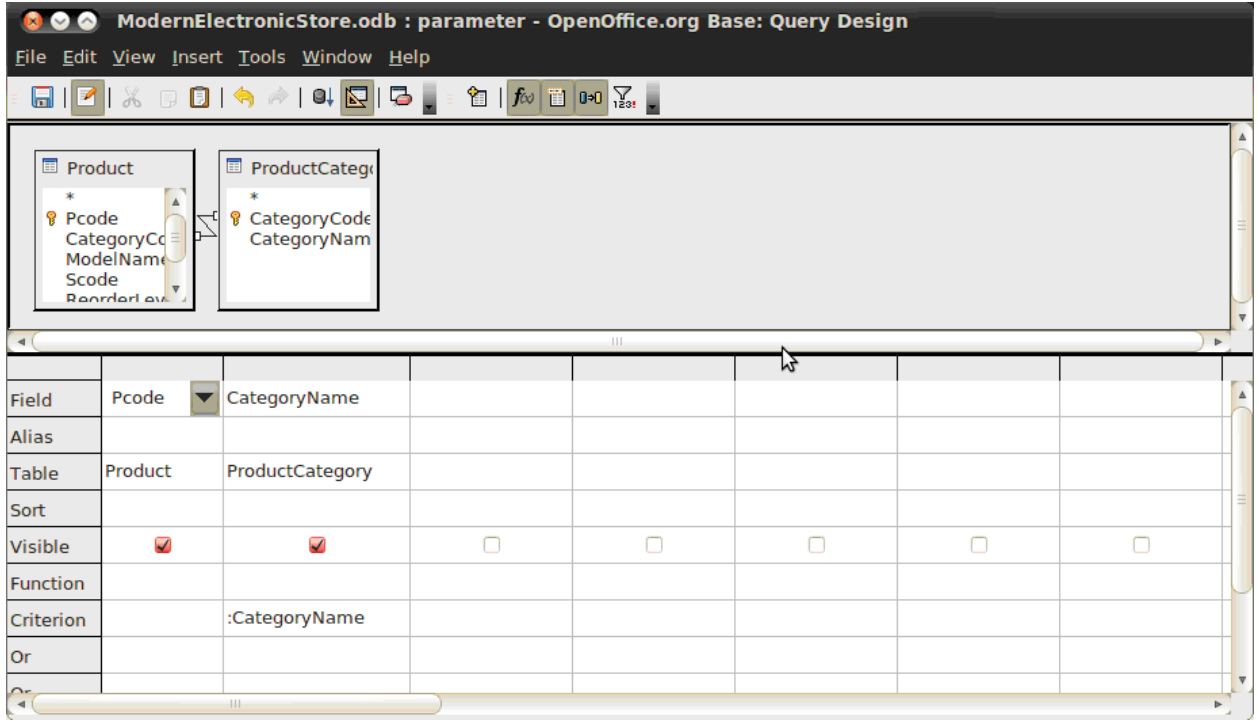
હવે આપણે મોડર્ન ઈલેક્ટ્રોનિક્સ સ્ટોરમાં ઉપલબ્ધ લેપટોપની વિગતો દર્શાવવા માટેની ક્વેરી બનાવીએ. આ માટેનું અપેક્ષિત પરિણામ મેળવવા માટે નીચેનાં પગલાં અનુસરો :

- ડિઝાઇન વ્યૂમાં નવી ક્વેરીની રચના કરો.
- Product અને ProductCategory ટેબલ ઉમેરો.
- Product tableમાં દેખાતા * પર ડબલ ક્લિક કરો. આમ કરવાથી Product ટેબલનાં તમામ ફિલ્ડ ક્વેરીમાં ઉમેરવામાં આવશે.
- Category ટેબલમાંથી CategoryName ફિલ્ડ પર ડબલ ક્લિક કરો.
- CategoryName ફિલ્ડના Criteria સેલમાં Laptop ટાઇપ કરો.
- ક્વેરીને DetailsOfLaptops નામ આપી સેવ કરો.

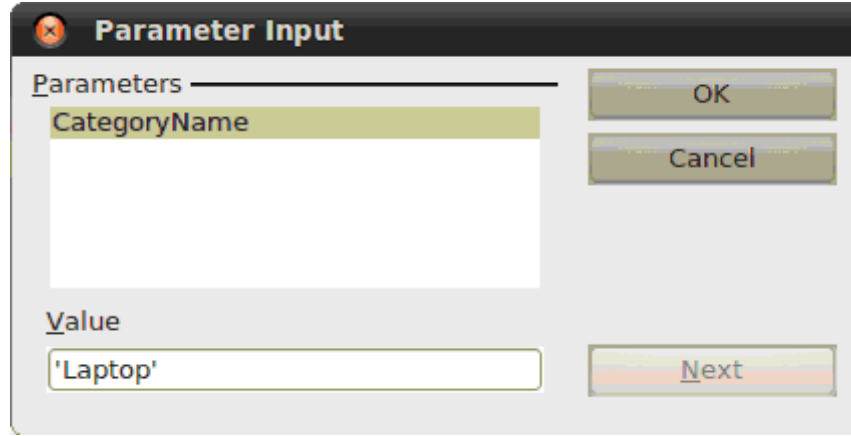
હવે ધારો કે તમને સ્માર્ટફોન વિશેની માહિતી મેળવવા કહેવામાં આવ્યું છે. આ માટે તમારે CategoryName ફિલ્ડના માપદંડ તરીકે Smartphone ઉમેરી નવી ક્વેરી બનાવવી પડશે અને તેને DetailsOfSmartphone નામ આપી સેવ કરવી પડશે. જો આ જ પ્રશ્ન દરેક ઉત્પાદન માટે પૂછવામાં આવે તો ? તમારે અનેક ક્વેરી બનાવવી પડશે !!

આ પ્રકારની સ્થિતિને પહોંચી વળવા બેઝ પાસે એક ઘણી જ ઉપયોગી અને રસપ્રદ સુવિધા છે. આ માટે આપણે પેરામીટર ક્વેરી બનાવી શકીએ. પેરામીટર ક્વેરી બનાવવા માટે ક્વેરીની ગોઠવણ પહેલાં જેવી જ રહેશે, પરંતુ Criterion ફિલ્ડમાં ખરેખર કિંમત લખવાને બદલે પેરામીટર ઉમેરવાની જરૂર પડશે. જ્યારે ક્વેરી રન કરવામાં આવશે, ત્યારે બેઝ જરૂરી પેરામીટરની માંગણી કરતું ડાયલોગબોક્સ રજૂ કરશે. જુદાં-જુદાં પ્રકારનાં ઉત્પાદનોની યાદી દર્શાવવા માટે પેરામીટર ક્વેરી બનાવવા નીચે જણાવેલ પગલાંને અનુસરો :

- ડિઝાઇન વ્યૂમાં નવી ક્વેરી બનાવો.
- તેમાં Product અને ProductCategory ટેબલ ઉમેરો.
- Product ટેબલના Pcode ફિલ્ડ પર ડબલ ક્લિક કરો.
- Product ટેબલના CategoryName ફિલ્ડ પર ડબલ ક્લિક કરો.
- CategoryName ફિલ્ડના Criterion સેલમાં :CategoryName ટાઇપ કરો. ક્વેરી આકૃતિ 11.27માં દર્શાવ્યા મુજબ દેખાશે.
અહીં એ નોંધ લેવી જરૂરી છે કે માપદંડ માટેના પેરામીટરની આગળ હંમેશા વિસર્ગ (Colon :) નિશાની ઉમેરવી જરૂરી છે.
- Run બટન પર ક્લિક કરી ક્વેરીનું પરિણામ જુઓ.
બેઝ આકૃતિ 11.28માં દર્શાવ્યા મુજબનો ડાયલોગબોક્સ રજૂ કરશે.
- Value નામના બોક્સમાં Laptop ટાઇપ કરી OK બટન પર ક્લિક કરો. લેપટોપની યાદી દર્શાવવામાં આવશે.



આકૃતિ 11.27 : પેરામીટર ક્વેરી



આકૃતિ 11.28 : પેરામીટર કિંમત

ઉત્પાદનનાં જુદાં-જુદાં નામ સાથે ક્વેરીનો ફરી અમલ કરવાનો પ્રયત્ન કરો અને પરિણામ જુઓ.

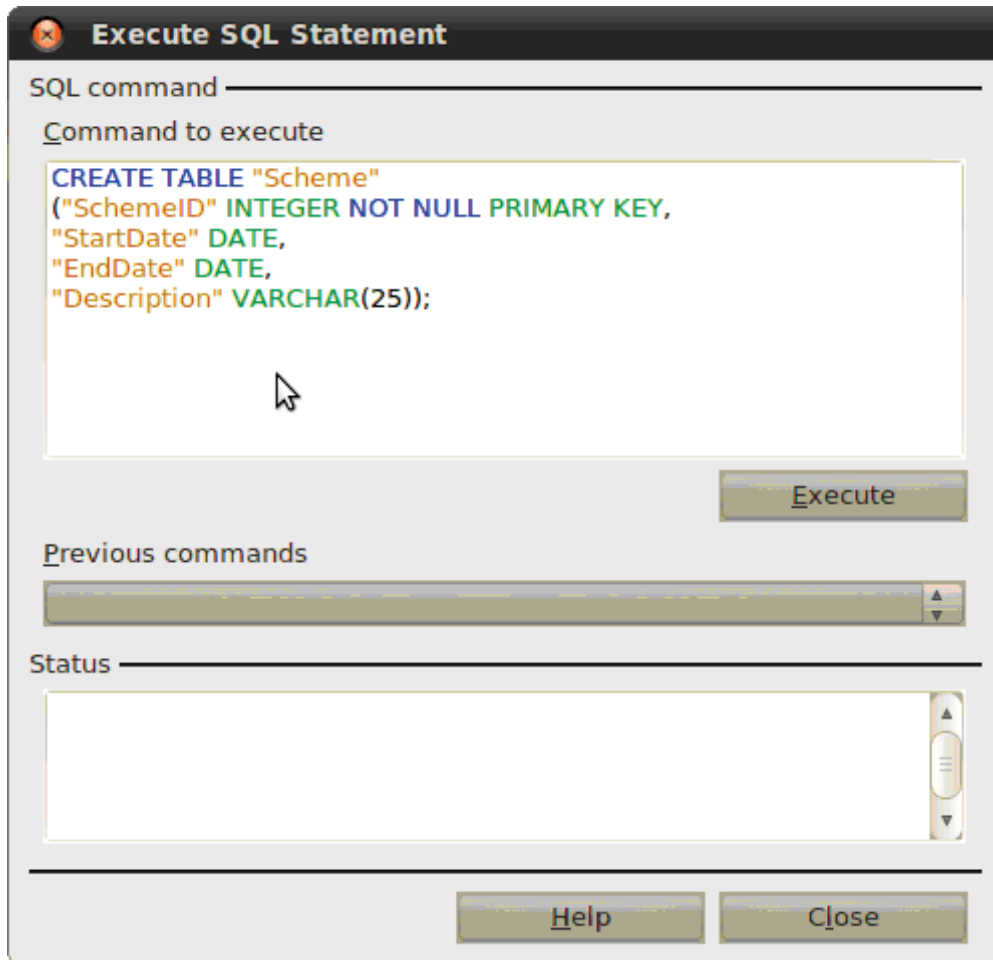
સ્ટ્રક્ચર્ડ ક્વેરી લેન્ગ્વેજ (Structured Query Language)

આપણે અત્યાર સુધીમાં જોયું કે બેઝમાં કોઈ પણ કાર્ય કરવા માટેના હંમેશા એકથી વધુ વિકલ્પ ઉપલબ્ધ હોય છે. ઉદાહરણ તરીકે, ડેટાબેઝમાં ટેબલ ઉમેરવા માટે *Create Table in Design View...* અને *Use Wizard to Create Table...* એ બન્ને વિકલ્પ ઉપલબ્ધ છે. તફાવત એ છે કે વિઝાર્ડ કાર્યને સરળ બનાવે છે, જ્યારે ડિઝાઇન વ્યૂ વધુ વિકલ્પો પૂરા પાડે છે. ટેબલ બનાવવા માટે હજી એક વધુ વિકલ્પ પણ ઉપલબ્ધ છે : SQL આદેશો, જે સૌથી વધુ લવચીકતા (Flexibility) અને નિયંત્રણ (Control) પૂરાં પાડે છે.

SQLનું પૂર્ણ નામ Structured Query Language છે. રિલેશનલ ડેટાબેઝમાં ક્વેરી પૂરી પાડવા માટેની આ પ્રમાણભૂત ભાષા છે. SQL ક્વેરી વિધાનોના સ્વરૂપમાં લખાય છે. આગળના પ્રકરણમાં આપણે ટેબલની રચના કરવા, તેમાં વિગતો ઉમેરવા, ટેબલને સુધારવા કે દૂર કરવાની રીતો જોઈ. આ તમામ કાર્ય SQL વિધાનોની મદદથી પણ કરી શકાય છે. હવે આપણે SQL વિધાનોની મદદથી ટેબલની રચના કરીએ.

મેનુબારમાં Tools મેનુ પર ક્લિક કરી SQL વિકલ્પ પસંદ કરો. તેનાથી *Execute SQL Statement* ડાયલોગ- બોક્સ ખૂલશે. તેમાં *Command to Execute* નામના બોક્સમાં કર્સર જોવા મળશે. અહીં આપણે ટેબલ કે ક્વેરીને સંબંધિત કાર્યો માટેની સૂચના ઉમેરી શકીએ છીએ. આપણે Scheme નામનું ટેબલ બનાવવાનો પ્રયત્ન કરીએ, જેમાં SchemeID, StartDate, EndDate અને Description નામનાં ચાર ફિલ્ડ હશે. ક્વેરીના અમલમાં થતી ક્ષતિ નિવારવા માટે અવતરણચિહ્નો, કેપિટલ કે સ્મોલ અક્ષરો અને વાક્યરચના પર ખાસ ધ્યાન આપો. *Command to Execute* નામના ટેક્સ્ટબોક્સમાં નીચે દર્શાવેલ વિધાન ટાઈપ કરો (જુઓ આકૃતિ 11.29).

```
CREATE TABLE "Scheme"  
("SchemeID" INTEGER NOT NULL PRIMARY KEY,  
"StartDate" DATE,  
"EndDate" DATE,  
"Description" VARCHAR(25));
```



આકૃતિ 11.29 : SQL આદેશ માટેની વિન્ડો

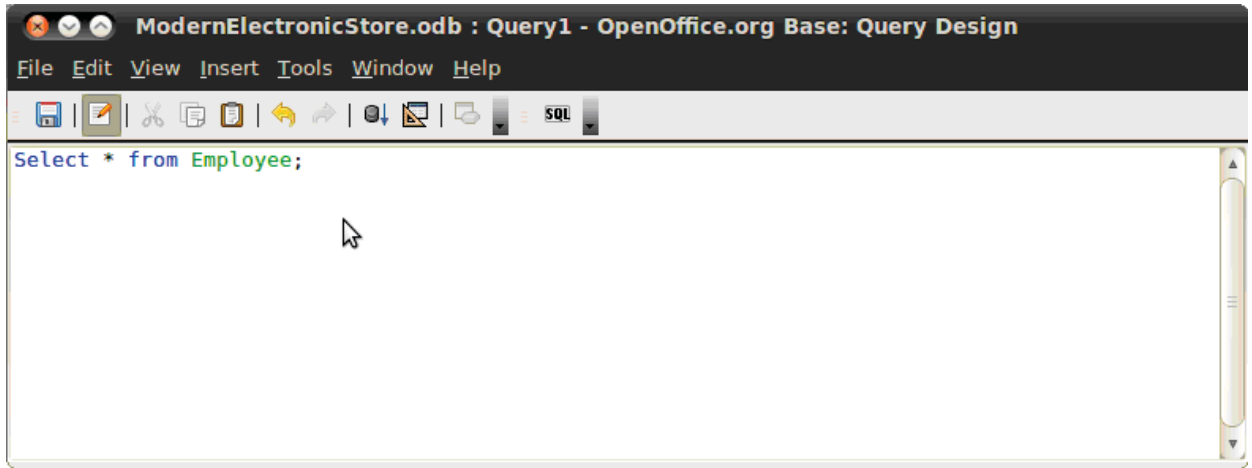
જ્યારે સૂચના ઉમેરવામાં આવે છે, ત્યારે *Execute* બટન સક્રિય (highlighted) બને છે. ક્વેરી ઉમેરાઈ ગયા પછી તેના પર ક્લિક કરો. આમ, કરવાથી લખવામાં આવેલા SQL આદેશનો અમલ કરી ટેબલની રચના કરવામાં આવશે. થોડી જ સેકન્ડમાં સૂચનાઓના સફળ અમલનો નિર્દેશ કરતી વિન્ડો રજૂ કરવામાં આવશે. આ સિવાય કોઈ અન્ય દર્શનીય પરિણામ સ્ક્રીન પર જોવા મળશે નહીં. *View* મેનુમાં *Refresh Tables* પસંદ કરવાથી સ્ક્રીનના નીચેના ભાગમાં અત્યાર સુધીમાં બનાવવામાં આવ્યાં હોય તે તમામ ટેબલની યાદી દર્શાવવામાં આવે છે, જેમાં હાલમાં બનાવેલ ટેબલ પણ જોવા મળશે.

કોઈ વાર ડેટાબેઝમાંથી ટેબલને દૂર કરવાની જરૂર પણ પડે છે. ટેબલને દૂર કરવા માટે **DROP TABLE** વિધાનનો ઉપયોગ કરવામાં આવે છે. ઉદાહરણ તરીકે , હમણાં જ બનાવેલા **Scheme** ટેબલને દૂર કરવા માટે *Command to Execute* લખેલા બોક્સમાં નીચે જણાવેલ વિધાન ઉમેરવું જોઈએ.

DROP TABLE Scheme IF EXISTS;

હવે, *Execute* બટન પર ક્લિક કરો. **Scheme** ટેબલ ડેટાબેઝમાંથી દૂર થયેલું જોઈ શકાશે.

ટેબલમાંથી માહિતી મેળવવા માટે પણ SQL વિધાનોનો ઉપયોગ કરી શકાય. આ માટે આપણે જુદી વિન્ડોનો ઉપયોગ કરવો પડશે. આ વિન્ડો ખોલવા માટે *Query* આઈકન પર ક્લિક કરો. ઉપરની પેનલમાં *Tasks* લખાણ હેઠળ ત્રણ વિકલ્પ દર્શાવવામાં આવ્યા છે. તેમાંથી *Create Query in SQL View...* પસંદ કરો. આમ કરવાથી આકૃતિ 11.30માં દર્શાવ્યા મુજબ એક ઝબકતા કર્સર સાથેના ડાયલોગબોક્સ રજૂ કરવામાં આવશે. માહિતી મેળવવા માટેનું SQL વિધાન અહીં લખવામાં આવે છે. આ વિધાન **SELECT** કી-વર્ડથી શરૂ કરવામાં આવે છે અને તેને SQL ક્વેરીના નામથી પણ ઓળખવામાં આવે છે. આકૃતિ 11.30 SQL ક્વેરીનું ઉદાહરણ દર્શાવે છે.



આકૃતિ 11.30 : SQL Select Query વિન્ડો

આકૃતિ 11.30માં દર્શાવેલ **SELECT * from Employee** ક્વેરીનું પરિણામ મેળવવાનો પ્રયત્ન કરીએ.

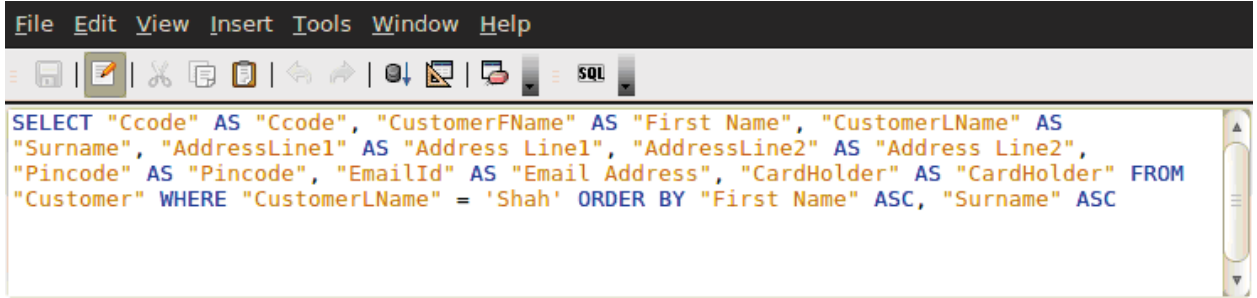
પરિણામ જોવા માટે ઉપરની બાજુએ રહેલા ફંક્શનબારમાં **Run Query** બટન પર ક્લિક કરો. **Employee table**માં સંગ્રહ કરવામાં આવેલી તમામ વિગતો દર્શાવવામાં આવશે. ક્વેરીનો સંગ્રહ કરવા માટે **save** બટન પર ક્લિક કરો. ક્વેરીને *EmployeeDetails* નામ આપી વિન્ડો બંધ કરો.

હવે આપણે કેટલીક વધુ SQL ક્વેરી બનાવીએ. ડિઝાઇન વ્યૂમાં નવી ક્વેરીની રચના કરો. *Create Query in SQL View...* વિકલ્પ પસંદ કરો. તેમાં નીચેનું વિધાન ટાઈપ કરો.

SELECT FirstName, LastName from Employee;

પરિણામ જોવા માટે સ્ક્રીનના ઉપરના ભાગમાં આવેલા ફંક્શનબારમાંથી Run Query બટન પર ક્લિક કરો. હવે તમામ કર્મચારીઓનાં FirstName અને LastName નામના ફિલ્ડની વિગતો જ દર્શાવવામાં આવી છે, તેની નોંધ કરો.

SQL વિધાનની મદદથી બનાવવામાં આવેલી ક્વેરી પણ ડિઝાઇનવ્યૂથી બનાવવામાં આવેલી ક્વેરીની જેમ જ Queries વિભાગની નીચેના ભાગમાં ઉપલબ્ધ બને છે તે જુઓ. કોઈ પણ ક્વેરી પર રાઈટ ક્લિક કરો. તેમાં Edit in SQL View વિકલ્પ જોવા મળશે. SQL વિધાનોનો ઉપયોગ કરી ક્વેરીની સુધારણા કરવા માટે તેને પસંદ કરો. ખરેખર તો વિઝાર્ડ કે ડિઝાઇનવ્યૂ દ્વારા બનાવવામાં આવેલી તમામ ક્વેરી માટેનાં SQL વિધાનો અહીં જોઈ શકાશે. ઉદાહરણ તરીકે, CustomerList ક્વેરી પર રાઈટ ક્લિક કરી Edit in SQL View વિકલ્પ પસંદ કરવાથી આકૃતિ 11.31માં દર્શાવ્યા મુજબનું વિધાન રજૂ કરવામાં આવશે.



```
File Edit View Insert Tools Window Help
SELECT "Ccode" AS "Ccode", "CustomerFName" AS "First Name", "CustomerLName" AS
"Surname", "AddressLine1" AS "Address Line1", "AddressLine2" AS "Address Line2",
"Pincode" AS "Pincode", "EmailId" AS "Email Address", "CardHolder" AS "CardHolder" FROM
"Customer" WHERE "CustomerLName" = 'Shah' ORDER BY "First Name" ASC, "Surname" ASC
```

આકૃતિ 11.31 : ડિઝાઇન વ્યૂથી બનાવવામાં આવેલ ક્વેરી માટેનું SQL વિધાન

As કી-વર્ડ પછી દર્શાવવામાં આવેલા First Name, Surname, AddressLine1, AddressLine2, Pincode, Email Address અને CardHolder શબ્દો ઉપનામ (Aliase) રજૂ કરે છે. જ્યારે FROM પછીનો Customer શબ્દ ટેબલનું નામ સૂચવે છે. માપદંડ અનુસાર માહિતી મેળવવા માટે WHERE કી-વર્ડને પણ આકૃતિ 11.31માં આવેલ SQL વિધાનમાં સમાવવામાં આવ્યો છે. ORDER BY કી-વર્ડ નિર્દેશ કરે છે કે પરિણામને CustomerFName ફિલ્ડ (જેને FirstName ઉપનામ સાથે રજૂ કરવામાં આવ્યું છે.) ઉપર ક્રમબદ્ધ ગોઠવવામાં આવ્યું છે. ASC શબ્દ દર્શાવે છે કે પરિણામને નામના ચડતા ક્રમમાં ગોઠવવામાં આવ્યું છે.

સારાંશ

DBMSનો એક ઉદ્દેશ વ્યાવસાયિક વિગતોનો સંગ્રહ કરવાનો છે. વધુ અગત્યનો ઉદ્દેશ આ જ વિગતોને જરૂર પડે ત્યારે અને તે સ્વરૂપે પાછી મેળવવાનો છે. ક્વેરી દ્વારા માહિતી કેવી રીતે મેળવી શકાય, તે આપણે આ પ્રકરણમાં જોયું. વિઝાર્ડ, ડિઝાઇન વ્યૂ અને SQL આદેશોની મદદથી ક્વેરીની રચના કરતા આપણે શીખ્યા. આપણે સરળ તેમજ પેરામીટર ક્વેરી બનાવી. સરળ ક્વેરીના અમલ વખતે કોઈ કિંમતની માંગણી કરવામાં આવતી નથી, જ્યારે પેરામીટર ક્વેરી રન થતી વખતે કિંમતોની માંગણી કરે છે. દરેક વખતે ઉમેરવામાં આવેલા પેરામીટર અલગ હોવાની શક્યતાને કારણે દર વખતે પેરામીટર ક્વેરીનું પરિણામ અલગ હોઈ શકે છે. હવે પછીના પ્રકરણમાં આ ક્વેરીનો ઉપયોગ કરી આપણે અહેવાલ (રિપોર્ટ)ની રચના કરીશું.

સ્વાધ્યાય

1. બેઝમાં ક્વેરી એટલે શું ? ક્વેરી શા માટે બનાવવી જોઈએ ?
2. ક્વેરીમાં માપદંડનો ઉપયોગ જણાવો.

3. વાઈલ્ડકાર્ડ અક્ષરો એટલે શું ? દરેકની વિસ્તૃત સમજૂતી આપો.
4. એગ્રિગેટ વિધેયો એટલે શું ? તેનો ઉપયોગ શેના માટે કરવામાં આવે છે ?
5. મુખ્યત્વે ઉપયોગમાં લેવાતાં એગ્રિગેટ વિધેયનાં નામ લખી સમજૂતી આપો.
6. ગણતરી માટેનાં ફિલ્ડ (Calculated field) ની વ્યાખ્યા આપી યોગ્ય ઉદાહરણ સાથે સમજાવો.
7. જૂથની રચના યોગ્ય ઉદાહરણ સાથે સમજાવો.
8. પેરામીટર ક્વેરી એટલે શું ?
9. Empty કિંમત એટલે શું ? યોગ્ય ઉદાહરણ સાથે સમજાવો.
10. હેતુલક્ષી પ્રશ્નો

- (1) સિલેક્ટ ક્વેરીના પરિણામને સામાન્ય રીતે કયા સ્વરૂપે દર્શાવવામાં આવે છે ?

(a) Table	(b) Query
(c) Constraint	(d) Relation
- (2) ડેટાબેઝમાંથી નિશ્ચિત માહિતી મેળવવા માટે બેઝની કઈ સુવિધાનો ઉપયોગ કરવામાં આવે છે ?

(a) Table	(b) Query
(c) Form	(d) Report
- (3) નીચેનામાંથી કયું એગ્રિગેટ વિધાન નથી ?

(a) Square root	(b) Sum
(c) Min	(d) Max
- (4) ક્વેરીના પરિણામને જૂથ સ્વરૂપે દર્શાવવા શું ઉપયોગી છે ?

(a) Orderby	(b) Groupby
(c) Arrangeby	(d) Setof
- (5) નીચેનામાંથી કયું વિધાન સાચું છે ?
 - (a) ક્વેરીનો એક ઓબ્જેક્ટ તરીકે સંગ્રહ કરી પુનઃ ઉપયોગ શક્ય છે.
 - (b) ક્વેરીને એક જ વાર લખી શકાય છે.
 - (c) ક્વેરીની રચના વિઝાર્ડ દ્વારા કરી શકાય નહીં.
 - (d) ક્વેરીની રચનામાં અન્ય ક્વેરીનો ઉપયોગ કરી શકાય નહીં.
- (6) ઉપનામ (Alias) એટલે શું ?
 - (a) ક્વેરીની નકલ કરવી.
 - (b) ક્વેરીમાં પસંદ કરેલ ફિલ્ડને અર્થપૂર્ણ નામ આપવું.
 - (c) ક્વેરીને પ્રિન્ટ કરવી.
 - (d) ક્વેરીને સંબંધિત નથી.
- (7) ક્વેરીના માપદંડમાં લખાણને શેમાં આવરીને લખવામાં આવે છે ?

(a) “ ” (ડબલ ક્વોટ)	(b) ‘ ’ (સિંગલ ક્વોટ)
(c) \$ (ડોલર)	(d) # (હેશ)

- (8) વાઈલ્ડકાર્ડ સાથે નીચેનામાંથી કયા ઓપરેટરનો ઉપયોગ કરી શકાય છે ?
- (a) Like (b) Is
(c) Equal (d) =
- (9) બેઝમાં નીચેનામાંથી કઈ નિશાનીનો વાઈલ્ડકાર્ડ અક્ષર તરીકે ઉપયોગ કરી શકાય ?
- (a) & (b) +
(c) - (d) *
- (10) માહિતી મેળવવા માટેની ક્રિમતની માંગણી કરતું ડાયલોગબોક્સ કઈ ક્વેરી રજૂ કરે છે ?
- (a) Select Query (b) Insert Query
(c) Parameter Query (d) Update query
- (11) પેરામીટર ક્વેરીમાં પેરામીટર આગળ કઈ નિશાની ઉમેરવામાં આવે છે ?
- (a) અલ્પવિરામ (b) વિસર્ગ
(c) પ્રશ્નચ્છેદ (d) આશ્ચર્ય ચિહ્ન
- (12) SQL એટલે શું ?
- (a) Simple Query Language
(b) Structured Query Language
(c) Simple Question for large Databases
(d) Structured queries for large Databases

પ્રાયોગિક સ્વાધ્યાય

1. પ્રકરણ 9 અને પ્રકરણ 10ના સ્વાધ્યાયમાં બનાવેલા **student** ડેટાબેઝનો ઉપયોગ કરી નીચે જણાવેલ ક્વેરી બનાવો.
- (a) દસમા ધોરણમાં અભ્યાસ કરતા વિદ્યાર્થીઓની યાદી બનાવો.
(b) શાળા છોડી ગયેલા વિદ્યાર્થીઓનાં નામ અને સરનામાં દર્શાવો.
(c) અમદાવાદ અને સુરતના શિક્ષકોની યાદી બનાવો.
(d) શાળામાં ભણાવવામાં આવતા કુલ વિષયોની સંખ્યા ગણો.
(e) જાન્યુઆરી, 2012માં Gr No-10 ધરાવતા વિદ્યાર્થીની કુલ હાજરીની ગણતરી કરો.
(f) શાળામાં શ્રી અબિલ મહેતા કુલ કેટલા વિષયો ભણાવે છે ? તેઓ ભણાવતા હોય તેવા વિષયોને તેના સંલગ્ન ધોરણ સાથે દર્શાવો.
(g) ઓક્ટોબર, 2011ની પ્રથમ પરીક્ષા માટે Gr No-1 ધરાવતા વિદ્યાર્થીના ટકા શોધો.
(h) દરેક વિદ્યાર્થીનું ઓક્ટોબર, 2011માં લેવાયેલ ગણિત વિષયનું પ્રથમ પરીક્ષાનું પરિણામ દર્શાવો.
(i) પેરામીટર તરીકે શહેરના નામની માંગણી કરે, તેવી પેરામીટર ક્વેરી બનાવો અને તે શહેરના વિદ્યાર્થીઓની યાદી દર્શાવો.
2. નીચેનાં ટેબલ બનાવો. દરેક ટેબલમાં યોગ્ય દસ રેકોર્ડ ઉમેરો :
- Student (StudentId , Name , Branch , Institute)
Exam (CourseNo , CourseName , DateOfExam)
Appeared (StudentId , CourseNo)

નીચેની ક્વેરી બનાવો :

- (a) અભ્યાસક્રમક્રમાંક 8 અને 12 માટે લેવાયેલ પરીક્ષાની વિગતો દર્શાવો.
- (b) પરીક્ષામાં હાજર વિદ્યાર્થીનાં ક્રમ, નામ અને અભ્યાસક્રમનું નામ દર્શાવો.
- (c) “Satyam” સંસ્થામાં અભ્યાસ કરતા તમામ વિદ્યાર્થીઓની યાદી બનાવો.
- (d) અભ્યાસક્રમ-ક્રમાંક 4માં નોંધાયેલા કુલ વિદ્યાર્થીઓની સંખ્યા ગણો.
- (e) પરીક્ષામાં હાજર હોય તેવા તમામ વિદ્યાર્થીઓના અભ્યાસક્રમનું નામ, પરીક્ષાની તારીખ અને નામની યાદી દર્શાવો.
- (f) 12/2/2012ના રોજ લેવાયેલ પરીક્ષા માટેના અભ્યાસક્રમનાં ક્રમાંક અને નામની યાદી દર્શાવો.
- (g) અભ્યાસક્રમ-ક્રમાંક 8 કે 10 હોય તથા પરીક્ષાની તારીખ 02/02/2012 હોય તેવી પરીક્ષાઓની વિગતો દર્શાવો.
- (h) A અક્ષરથી શરૂ થતી શાખાની યાદી દર્શાવો.
- (i) ABC સંસ્થાના તમામ રેકૉર્ડ કાઢી નાંખો.

3. નીચેનાં ટેબલ બનાવી દરેકમાં દસ યોગ્ય રેકૉર્ડ ઉમેરો :

EMPLOYEE (EmpId, EmpName, Salary, Gender , Department, JoiningDate)

નીચે જણાવેલ ક્વેરી બનાવો :

- (a) D અક્ષરથી શરૂ થતા કર્મચારીઓનાં નામની યાદી દર્શાવો.
- (b) ₹.1000થી ₹.3000ની વચ્ચે પગાર મેળવતા કર્મચારીઓની યાદી દર્શાવો.
- (c) તમામ પુરુષ કર્મચારીઓની યાદી દર્શાવો.
- (d) Marketing ડિપાર્ટમેન્ટમાં હોય તેવા કર્મચારીની યાદી દર્શાવો.
- (e) સંસ્થા દ્વારા આપવામાં આવતો સરેરાશ પગાર ગણો.
- (f) ₹.5000થી વધુ પગાર ધરાવતા કર્મચારીઓની વિગતો દર્શાવો.
- (g) 01/01/2012 પહેલા જોડાયેલા કર્મચારીઓની વિગતો દર્શાવો.
- (h) Marketing અથવા Finance ડિપાર્ટમેન્ટમાં કાર્ય કરતા કર્મચારીઓની વિગતો દર્શાવો.
- (i) Purchase ડિપાર્ટમેન્ટમાં કાર્ય કરતા ન હોય તેવા કર્મચારીઓની વિગતો દર્શાવો.
- (j) Finance ડિપાર્ટમેન્ટમાં કાર્ય કરતા 10/9/2011 તારીખ પહેલા જોડાયેલા કર્મચારીઓની વિગતો દર્શાવો.





આગળના પ્રકરણમાં, આપણે ડેટાબેઝ કેવી રીતે બનાવવો, Table Datasheet Viewનો ઉપયોગ કરીને ડેટાબેઝના ટેબલ (કોષ્ટક) પર વિવિધ ક્રિયાઓ કરવી અને Query Datasheet Viewનો ઉપયોગ કરીને કેવી રીતે માહિતીને નિહાળવી તેની ચર્ચા કરી. Datasheet View ઊભી અને આડી હરોળના સ્વરૂપમાં હોય છે. જ્યારે આપણે Product નામના ટેબલમાં (આડી અને ઊભી હરોળ સ્વરૂપે રજૂ કરાતી માહિતીને અંગ્રેજીમાં અને ડેટાબેઝની ભાષામાં 'ટેબલ' કહે છે.) ડેટા દાખલ કર્યો ત્યારે આપણે તેને વિસ્તારપત્રકની ઢબે દાખલ કર્યો હતો. આ રીતે 'બ્લેક એન્ડ વ્હાઈટ' સ્વરૂપે ડેટા દાખલ કરવાની અને નિહાળવાની પદ્ધતિ ક્યારેક ઉપયોગકર્તા માટે અણગમો પેદા કરનારી કે કંટાળાજનક બને છે. Customer ટેબલ તૈયાર કરતી વખતે આપણે ફિલ્ડનું નામ આપતી વખતે ટૂંકાં નામોનો ઉપયોગ કરેલો, જેમકે, Ccode જે customer code (કસ્ટમર કોડ) માટે ટૂંકું નામ છે. ઘણીવાર ફિલ્ડનાં આવાં ટૂંકાં નામ સ્વયંસ્પષ્ટ બનતાં નથી. આ પ્રકરણમાં આપણે ડેટાને દાખલ કરવા અને દર્શાવવા ફોર્મ (Forms) અને રિપોર્ટ (Report)ના નામે ઓળખાતી એક વૈકલ્પિક વ્યવસ્થાની ચર્ચા કરીશું. Report સવલતનો ઉપયોગ કરીને કેવી રીતે નિર્ધારિત સ્વરૂપે પરિણામ દર્શાવવું તે પણ આપણે ચર્ચાશું. આગળના પ્રકરણમાં ચર્ચા કર્યા મુજબ Table અને Queryની જેમ જ Form અને Report પણ Database Windowના ડાબા ભાગમાં દર્શાવવામાં આવતા ઓબ્જેક્ટ છે. તો ચાલો આ બંનેની એક પછી એક ચર્ચા કરીએ.

ફોર્મ (Forms)

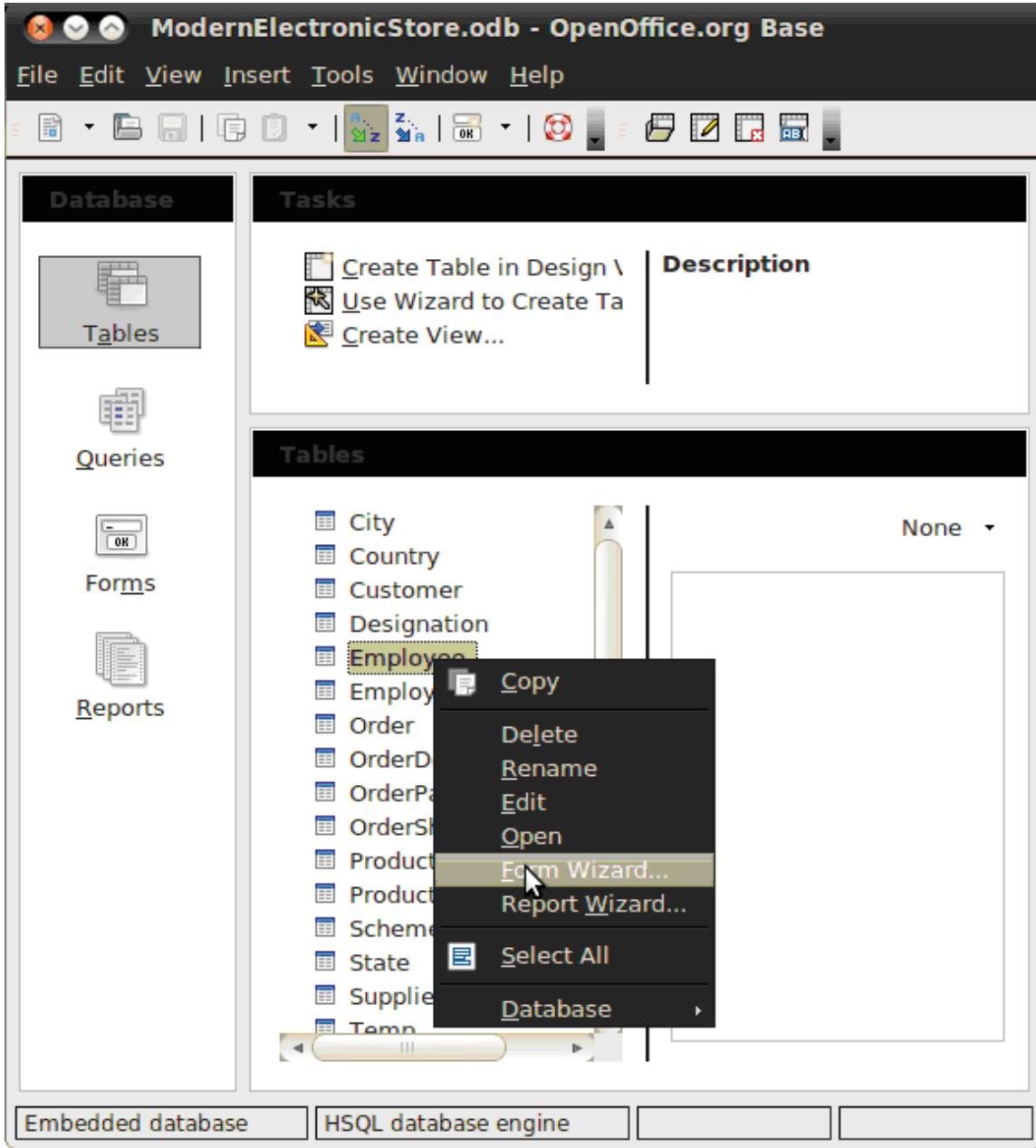
એ યાદ કરો કે જ્યારે તમે શાળામાં પ્રવેશ માટે ગયા હતા, જ્યાં કદાચ તમે કે પછી તમારા વાલીએ તમારી વિગતો દર્શાવવા પ્રવેશ ફોર્મ ભર્યું હશે. આ પ્રવેશફોર્મ 'ખાલી જગ્યા પૂરી' એ પદ્ધતિથી બનેલું હોય છે, જેમાં ફોર્મમાં માંગ્યા મુજબ જે-તે ખાનામાં તમારે તમારી વિગતો લખવાની હોય છે. Base પણ વિગતો નિહાળવા અને ટેબલમાંના ડેટા સાથે વિવિધ કામ કરવા આવી જ વ્યવસ્થા પૂરી પાડે છે, જેને ફોર્મ (Form) તરીકે ઓળખવામાં આવે છે. Baseનું આ Form આપણા કાગળ અને પેન દ્વારા ઉપયોગી બનતા પરંપરાગત ફોર્મની કામગીરીને મળતું આવે છે. ડેટાબેઝની પરિભાષામાં ફોર્મ એટલે ડેટા દાખલ (ડેટાએન્ટ્રી) કરવા અને ડેટામાં સુધારા-વધારા કરવા માટેનું ફ્રન્ટ-એન્ડ (front end). વિવિધ ઢબ, રંગો સહિત નામ અને લોગોનો ઉપયોગ કરીને ફોર્મ બનાવી શકાય છે. ફોર્મ તૈયાર કરવું એ ખરેખર રસપ્રદ કાર્ય છે. ફોર્મ તૈયાર કરવા માટેના બે રસ્તા છે.

1. વિઝાર્ડનો ઉપયોગ કરીને,
2. DesignViewનો ઉપયોગ કરીને

સરળમાં સરળ રસ્તો વિઝાર્ડનો ઉપયોગ કરવાનો છે. તો ચાલો, Baseમાં વિઝાર્ડનો ઉપયોગ કરીને ફોર્મ બનાવવાનું શીખવાની શરૂઆત કરીએ.

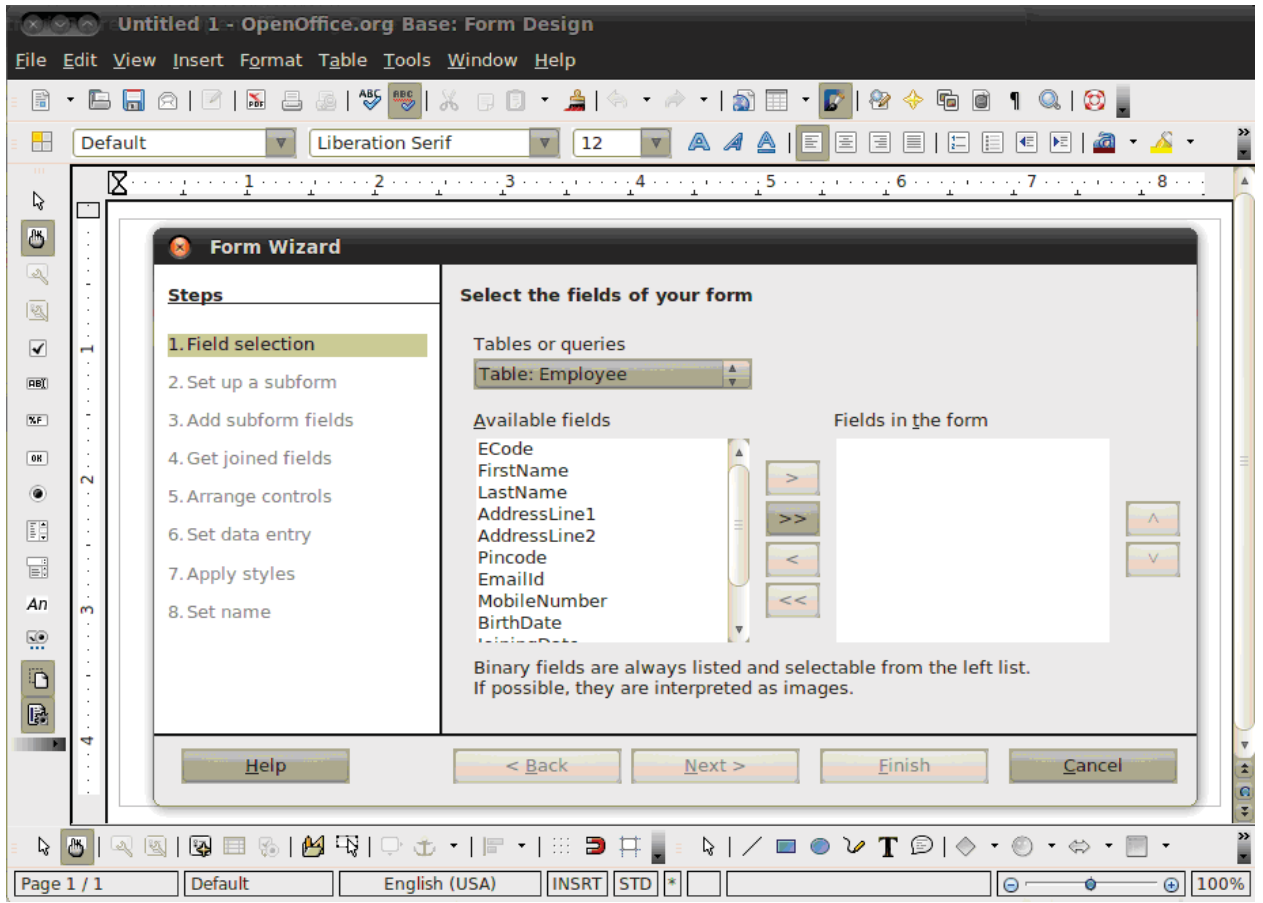
વિઝાર્ડનો ઉપયોગ કરીને ફોર્મ બનાવવું

ફોર્મવિઝાર્ડ શરૂ કરવાના બે રસ્તા છે. પહેલો રસ્તો છે : માઉસનું જમણું બટન દબાવવાનો. જે મુજબ જે ટેબલ માટે ફોર્મ બનાવવું હોય તે ટેબલ પર માઉસ લઈ જઈ જમણું બટન દબાવવું પડે. બીજા રસ્તા મુજબ Forms આઈકોન પર ક્લિક કરી Tasks ભાગમાંથી Use Wizard to create a form વિકલ્પ પસંદ કરો. પ્રથમ વિકલ્પનો ઉપયોગ કેવી રીતે કરવો તે આકૃતિ 12.1માં દર્શાવાયો છે. વિઝાર્ડનો ઉપયોગ કરીને એક સાદું ફોર્મ બનાવવા આપેલાં પગલાં અનુસરો.



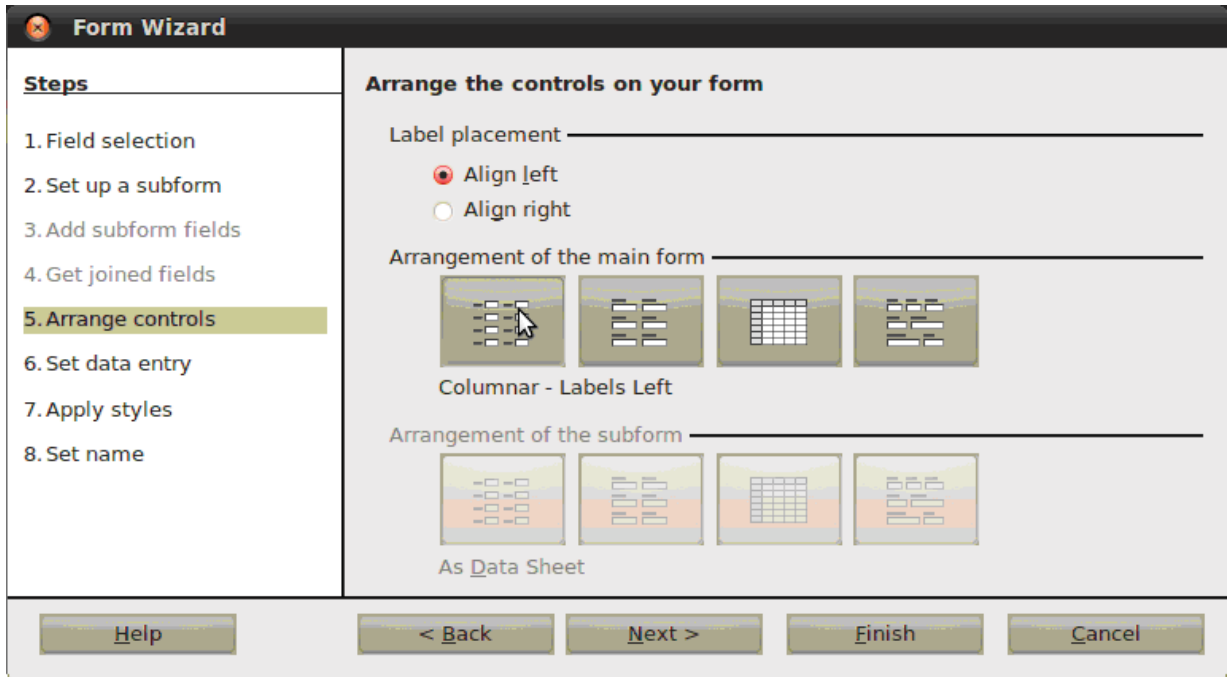
આકૃતિ 12.1 : વિઝાર્ડનો ઉપયોગ કરીને ફોર્મ બનાવવું

- Employee નામના ટેબલ પર જમણું બટન દબાવો.
- પોપ-અપ મેનુમાંથી *FormWizard* વિકલ્પ પસંદ કરો. (જુઓ આકૃતિ 12.1)
- વિઝાર્ડ આકૃતિ 12.2માં દર્શાવ્યા મુજબ *FormWizard* અને *FormDesign* નામનાં બે ડાયલોગ બોક્સ ખોલશે. હાલ પૂરતું આપણે *Form Wizard* ડાયલોગ બોક્સ તરફ જ ધ્યાન આપીશું. એ પણ અવલોકન કરો કે, *Form wizard* ડાયલોગબોક્સના *Available Fields* લિસ્ટ બોક્સમાં *Employee* ટેબલનાં બધાં ફિલ્ડની યાદી દર્શાવાય છે.
- *Fields in the form* ખાનામાં આ બધાં ફિલ્ડને મોકલવા જમણી બાજુનાં બે તીર (>>) ની નિશાની પર ક્લિક કરો.
- *Next* બટન પર ક્લિક કરો.



આકૃતિ 12.2 : Form બનાવવા માટે ફિલ્ડની પસંદગી

- આ પગલામાં સબ-ફોર્મની ગોઠવણી કરવાનો પણ સમાવેશ થાય છે. સબ-ફોર્મ વિશે પાછળથી આ પ્રકરણનાં અંતે ચર્ચા કરવામાં આવશે. એક સાદું ફોર્મ બનાવવા માટે Next બટન પર ક્લિક કરો અને આકૃતિ 12.3માં દર્શાવ્યા મુજબ ફોર્મ વિઝાર્ડનું 5મું પગલું દર્શાવાશે.



આકૃતિ 12.3 : નિયંત્રકોની ગોઠવણ

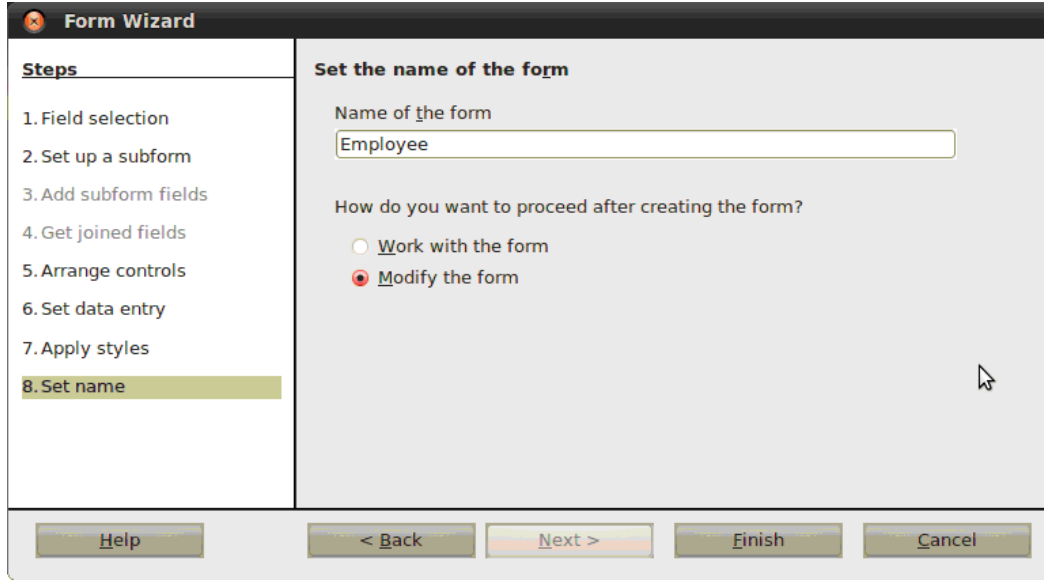
- આ પગલું ફોર્મ પર દેખાતા વિવિધ નિયંત્રકોને ગોઠવવાનું આવરી લે છે. નિયંત્રકમાં *label* અને *field* એમ બે ભાગોનો સમાવેશ થાય છે. આ પગલું જ્યારે અમલમાં મૂકવામાં આવશે, ત્યારે તે નક્કી કરે છે કે સ્ક્રીન પર Label અને Fields કેવી રીતે દેખાશે. લેબલ ડાબી બાજુ અથવા જમણી બાજુની ધારી તરફ ગોઠવાય છે. આપણા કિસ્સામાં *AlignLeft* વિકલ્પ પસંદ કરાયેલ છે. લેબલને આનુષંગિક ફિલ્ડ ચાર રીતે ગોઠવી શકાય : *Columnar – LabelsLeft*, *Columnar - Labels on Top*, *As DataSheet*, અને *In Blocks - LabelsAbove*. *Arrangement of the main form*, શીર્ષક હેઠળ *Columnar - LabelsLeft* આઈકોન પર ક્લિક કરો.
- Next બટન પર ક્લિક કરો. આમ કરવાથી આકૃતિ 12.4માં દર્શાવ્યા મુજબ Formwizard નું છઠ્ઠું પગલું દર્શાવાશે.

આકૃતિ 12.4 : ડેટાએન્ટ્રી નિયંત્રક ગોઠવણી

- જો આમાંની કોઈ પણ સવલતનો ઉપયોગ ન કરવો હોય તો, સામાન્ય રીતે જે ગોઠવણ છે (By Default) તે જ ગોઠવણને સ્વીકારી લો. એ પછી Next બટન પર ક્લિક કરો. આમ કરવાથી આકૃતિ 12.5માં દર્શાવ્યા મુજબ Form Wizardનું 7મું પગલું દર્શાવાશે.

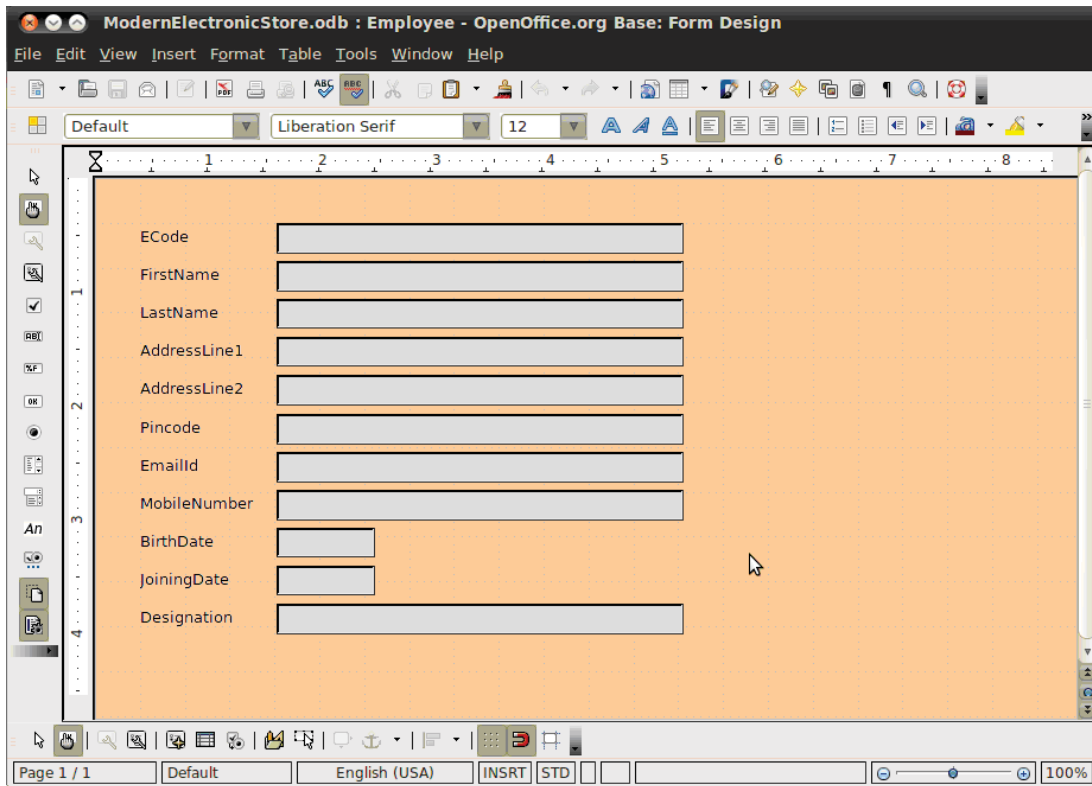
આકૃતિ 12.5 : વિવિધ ઢબનો અમલ

- જે રંગ તમે ઈચ્છતા હોવ તે *ApplyStyles* યાદીમાંથી પસંદ કરો. આકૃતિ 12.5માં એવું દર્શાવાયું છે કે, રંગ તરીકે *ApplyStyles* યાદીમાંથી Beige રંગ પસંદ કરાયો છે જ્યારે *FieldBorder* ના ત્રણ વિકલ્પોમાંથી 3D look પસંદ કરાયો છે. વિવિધ શક્ય ગોઠવણીઓનો તમે અભ્યાસ અને અવલોકન કરી શકો. Form wizardનું આઠમું પગલું દર્શાવાય તે માટે Next બટન પર ક્લિક કરો. આઠમા પગલાના સ્ક્રીનનો દેખાવ આકૃતિ 12.6માં દર્શાવ્યા મુજબનો દેખાશે.



આકૃતિ 12.6 : નામ આપવા માટેનું ફોર્મ

- ફોર્મનું નામ ટાઈપ કરો. અહીં આ કિસ્સામાં તે Employee છે. *Modify the form* રેડિયો બટન પર ક્લિક કરો. એ પછી Finish બટન પર ક્લિક કરો. હવે આકૃતિ 12.7માં દર્શાવ્યા મુજબ એડિટ મોડ (Edit Mode)માં ફોર્મ ખૂલશે.



આકૃતિ 12.7 : Edit મોડમાં ફોર્મ

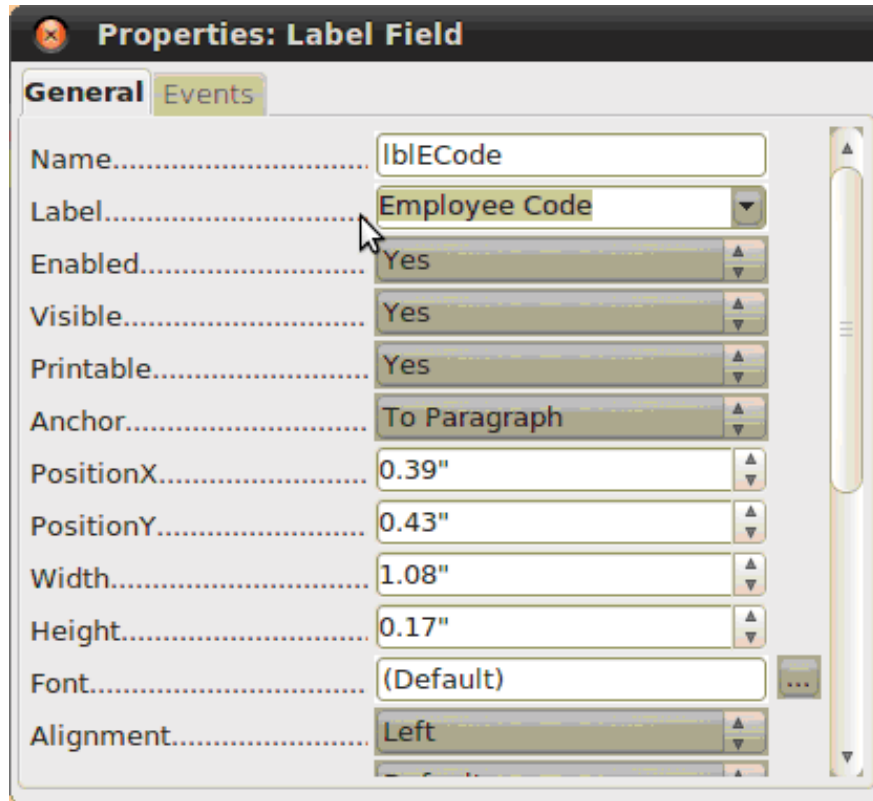
ફોર્મમાં સુધારા-વધારા કરવા

એક વાર ફોર્મ તૈયાર કર્યા પછી આપણી ઈચ્છા અનુસાર ફોર્મ પરના નિયંત્રકો (controls)ને ખસેડીને સરળતાથી ફેરફાર કરી શકાય. ફોર્મમાં નિયંત્રકોને આપણે જુદાં-જુદાં સ્થાન પર ફેરવી શકીએ. ફોર્મમાં નિયંત્રકોની પાછળ દેખાતું દૃશ્ય (બેકગ્રાઉન્ડ) પણ બદલી શકીએ. ફોર્મમાં કોઈ પણ ફિલ્ડના લેબલને પણ આપણે બદલી શકીએ જેમકે, Ecode લેબલને બદલીને EmployeeCode કરી શકીએ. ટેક્સ્ટ બોક્સને બદલીને લિસ્ટ બોક્સ પણ બનાવી શકાય. તો ચાલો, વિઝાર્ડ દ્વારા બનાવેલ ફોર્મને સુધારીએ.

લેબલ ટેક્સ્ટ બદલવી :

તો ચાલો, ECode નામના લેબલને બદલીને Employee Code કરવા નીચે મુજબનાં પગલાં અનુસરીએ :

- Ecode ધરાવતા લેબલને પસંદ કરવા CTRL + Click કરો.
- હવે લેબલ ઉપર માઉસનું જમણું બટન દબાવો. આમ કર્યા પછી પોપ-અપ મેનુમાંથી Control... વિકલ્પ પસંદ કરો. આકૃતિ 12.8માં દર્શાવ્યા મુજબ Properties : LabelField નામનું ડાયલોગબોક્સ ખૂલશે.
- Labelના ખાનામાં Employee Code ટાઈપ કરો. (જુઓ આકૃતિ 12.8)
- Properties: LabelField નામનું ડાયલોગબોક્સ બંધ કરો. આમ કર્યા પછી તમે જોઈ શકશો કે Form-Designમાં દેખાતું Ecode બદલાઈને Employee Code થઈ ગયું હશે.



આકૃતિ 12.8 : Properties ડાયલોગબોક્સ

આકૃતિ 12.8માં જોઈ શકાય છે કે, Properties: Label Field ડાયલોગબોક્સમાં આ નિયંત્રક કંટ્રોલ સંબંધી ઘણી લાક્ષણિકતાઓ આપવામાં આવે છે. આ નિયંત્રકની અન્ય લાક્ષણિકતાઓ જેવી કે ઊંચાઈ (Height), પહોળાઈ (Width), લખાણને કઈ બાજુ ગોઠવવું (Alignment) અને Visibility of a control વગેરેને પણ આ ડાયલોગબોક્સ મારફતે ગોઠવી શકાય છે.

નિયંત્રકોને ખસેડવા

મુખ્ય ફોર્મમાં નિયંત્રકોને એક લેબલ અને તેનું ફિલ્ડ હોય છે. આ લેબલ કે ફિલ્ડ બેમાંથી ગમે તે એક નિયંત્રક પર ક્લિક કરવાથી; બન્ને એકસાથે પસંદ થાય છે. આકૃતિ 12.9માં દર્શાવ્યા મુજબ આ નિયંત્રકની ફરતે આઠ લીલા રંગનાં (ગ્રીન) હેન્ડલ સાથેની લીટી દોરાઈ જાય છે. આમ થયા પછી તમે આ નિયંત્રકને ફોર્મમાં માઉસ દ્વારા ઘસડીને ગમે ત્યાં મૂકી શકો છો.

ઘણીવાર એવી જરૂર પડે કે જેમાં આપણે માત્ર નિયંત્રકના લેબલને જ બદલવું હોય. એ માટે અહીં માત્ર એક જ ભાગ એટલે કે લેબલને જ પસંદ કરવું પડે. માત્ર લેબલ કે પછી ટેક્સ્ટફિલ્ડ બેમાંથી કોઈ એકને જ પસંદ કરવા માટે CTRL કી દબાવી રાખી લેબલ અથવા ટેક્સ્ટફિલ્ડ પર ક્લિક કરો. આકૃતિ 12.10 આવી રીતે માત્ર ટેક્સ્ટફિલ્ડ પસંદ થયેલી દર્શાવે છે.

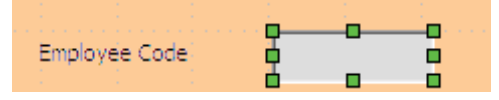
નિયંત્રકના કદમાં ફેરફાર કરવો

ચાલો, હવે EmployeeCode ફિલ્ડની પહોળાઈ વધારીએ.

- EmployeeCode ફિલ્ડ ટેક્સ્ટ બોક્સને પસંદ કરવા CTRL કી દબાવી રાખી ફિલ્ડ ટેક્સ્ટ બોક્સ પર ક્લિક કરો.
- ફિલ્ડ ટેક્સ્ટ બોક્સની જમણી બાજુના ગ્રીન હેન્ડલમાંથી વચ્ચેનું ગ્રીન હેન્ડલ ઉપર કર્સરને લઈ જાવ. આમ કર્યા પછી કર્સર આકૃતિ 12.11માં દર્શાવ્યા મુજબ બે બાજુ તીરના નિશાનવાળું બની જવું જોઈએ.
- જ્યાં સુધી ફિલ્ડ ટેક્સ્ટ બોક્સનું કદ તમારી ઈચ્છાનુસાર મોટું ન થાય ત્યાં સુધી માઉસનું ડાબું બટન દબાવી રાખીને માઉસને જમણી બાજુ ઘસડો. આમ કરવાથી તમે આકૃતિ 12.12માં દર્શાવ્યા મુજબ સ્ક્રીનના નીચેના ભાગે ફિલ્ડ ટેક્સ્ટ બોક્સનું બદલાયેલું કદ જોઈ શકશો.



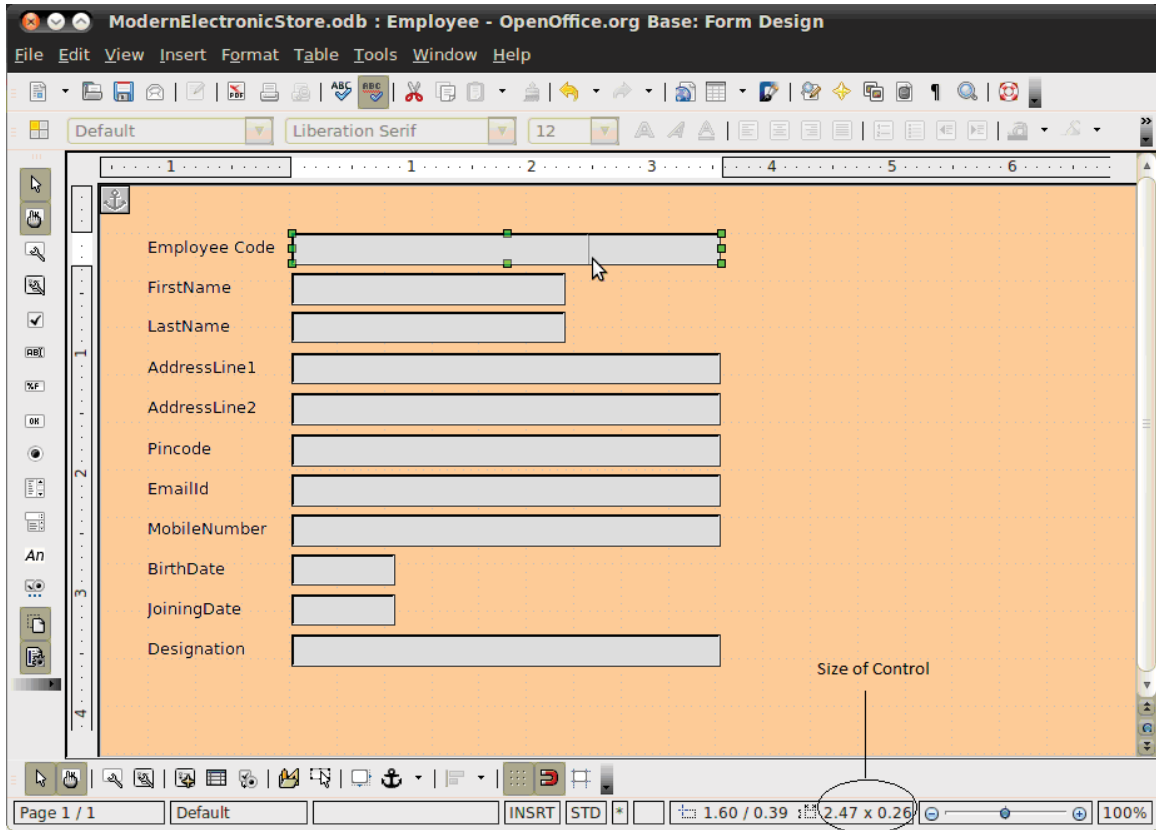
આકૃતિ 12.9 : A Selected control



આકૃતિ 12.10 :
નિયંત્રકનું માત્ર ફિલ્ડ પસંદ કરવું

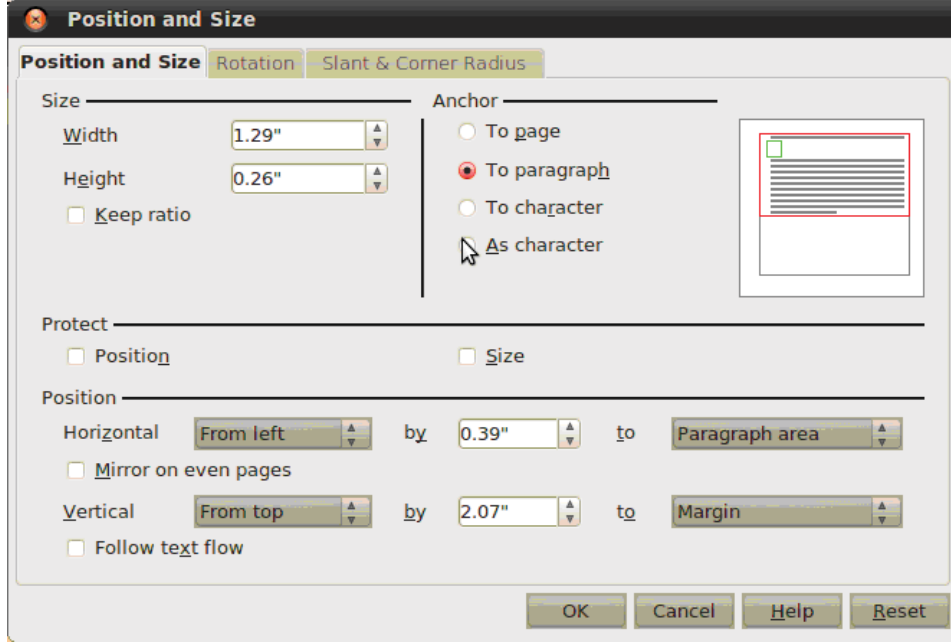


આકૃતિ 12.11 :
નિયંત્રકનું કદ બદલવું



આકૃતિ 12.12 : નિયંત્રકોનું કદ બદલવા અંગે

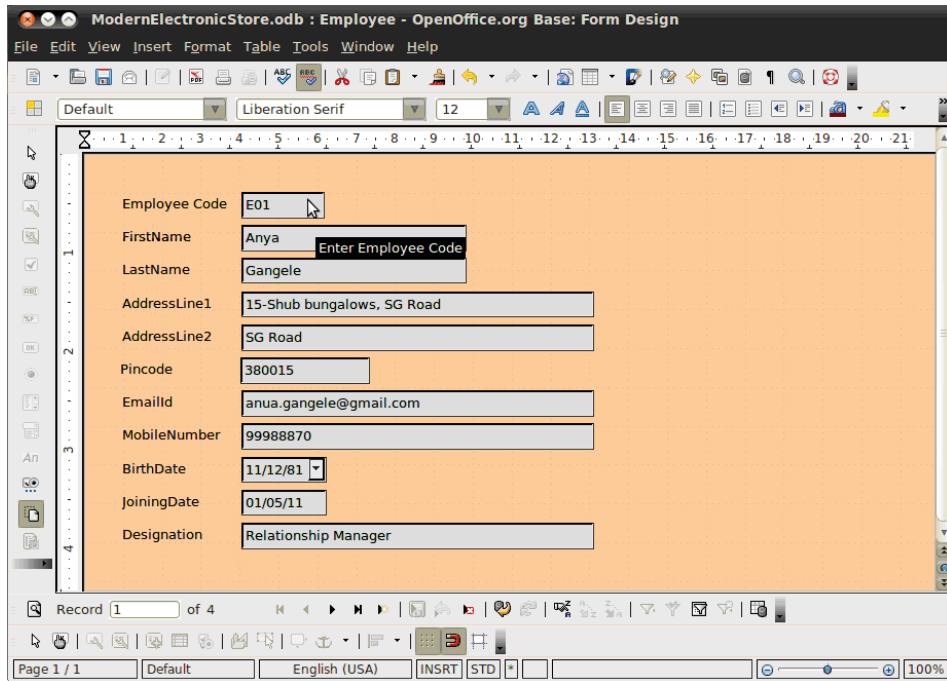
- આકૃતિ 12.13માં દર્શાવ્યા મુજબ Position અને Size નામનું ડાયલોગબોક્સ ખોલવા તમે *Position* અને *Size* વિસ્તાર પર ડબલ ક્લિક કરી શકો.



આકૃતિ 12.13 : Position અને size ડાયલોગબોક્સ

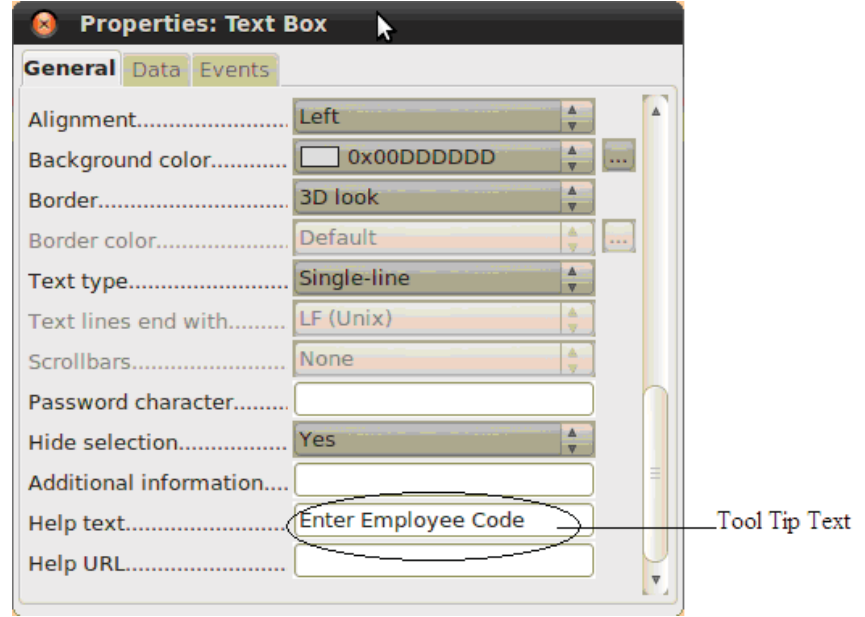
સહાયરૂપ શબ્દો બનાવવા :

શું તમે ક્યારેય માઉસપોઇન્ટરને ટૂલબારના કોઈ ટુલ પર લઈ જઈને થોડીક વાર રાખી મૂકવાની કોશિશ કરી છે ખરી ? જરા કરી જુઓ! આમ કરવાથી એ ટૂલની નીચે એક પીળા ખાનામાં આ ટૂલના ઉદ્દેશ્ય અંગે એક-બે શબ્દ દર્શાવાશે. આ પીળા રંગના ખાનામાં દર્શાવાતા શબ્દને 'ટૂલટીપ' (ToolTip) તરીકે ઓળખવામાં આવે છે. આ જ રીતે, આકૃતિ 12.14માં દર્શાવ્યા પ્રમાણે Baseમાં ફોર્મની અંદર ડેટાએન્ટ્રી માટે મૂકેલ નિયંત્રકોની સાથે આવા સહાયરૂપ શબ્દો સાથેની ટૂલટીપ જોડી શકાય છે.



આકૃતિ 12.14 : સહાયરૂપ શબ્દો સાથે ટૂલટીપ

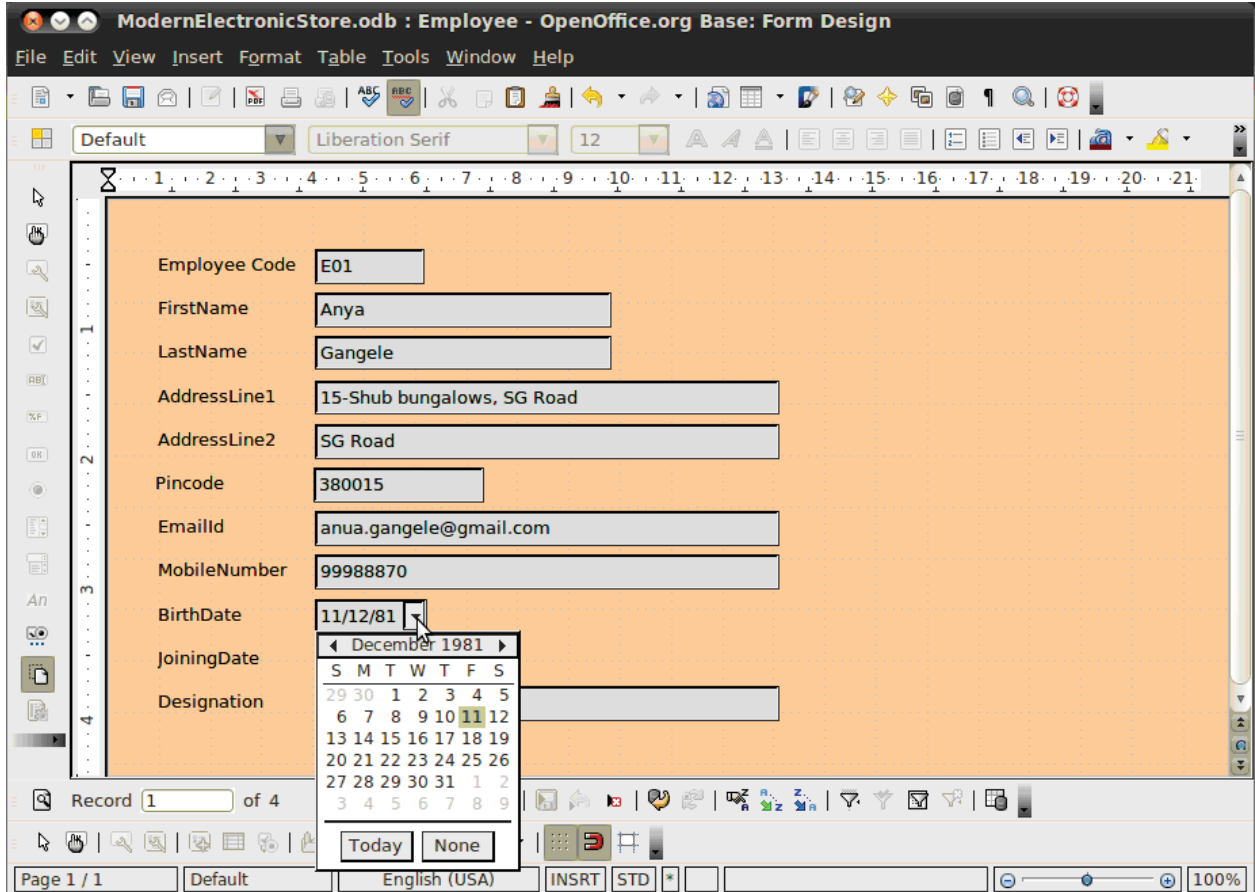
તો ચાલો, *EmployeeCode* લેબલની સાથે સંકળાયેલ ટેક્સ્ટ બોક્સ પર સહાયરૂપ શબ્દો (ટૂલટીપ) તરીકે Enter EmployeeCode જોડીએ. નિયંત્રકને પસંદ કરવા CTRL કી દબાવી રાખીને ટેક્સ્ટ બોક્સ પર ક્લિક કરો. નિયંત્રક પર માઉસનું જમણું બટન દબાવી Control વિકલ્પ પસંદ કરો. આકૃતિ 12.15માં દર્શાવ્યા મુજબ *Properties: TextBox* દર્શાવશે. *Helptext*..... નામનું લેબલ ન દેખાય ત્યાં સુધી સ્કોલ ડાઉન કરો. હવે *Helptext* નામના ખાનામાં આકૃતિ 12.15માં દર્શાવ્યા મુજબ “Enter Employee Code” ટાઈપ કરો.



Date Fieldને બદલવું.

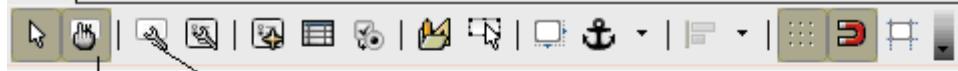
આકૃતિ 12.15 : Help text ગોઠવવી

આકૃતિ 12.16માં દર્શાવ્યા મુજબ Date ફિલ્ડને બદલીને કેલેન્ડર દ્વારા નિયંત્રણ માટેનું ફિલ્ડ બનાવી દેવા માટે નીચેનાં પગલાં અનુસરો.



આકૃતિ 12.16 : DateFieldને બદલવું

- CTRL કી દબાવી રાખીને BirthDate ફિલ્ડના ટેક્સ્ટ બોક્સ પર ક્લિક કરીને ફિલ્ડ પસંદ કરો.
- કર્સરને ખસેડીને ફિલ્ડ ટેક્સ્ટ બોક્સની જમણી બાજુ દર્શાવાયેલ ગ્રીન હેન્ડલની વચ્ચેના ગ્રીન હેન્ડલ પર કર્સરને લઈ જાવ. આમ કર્યા પછી બે બાજુ તીરના નિશાન જેવું કર્સર થઈ જાય તે જુઓ.
- માઉસનું ડાબું બટન દબાવી રાખીને માઉસને જમણી બાજુ ઘસડો અને જ્યાં સુધી જરૂરિયાત મુજબની લંબાઈ ન થાય ત્યાં સુધી માઉસને ઘસડતા રહો.
- માઉસના બટનને છોડી દો.
- ટેક્સ્ટ બોક્સ પર માઉસનું જમણું બટન દબાવીને *Control* વિકલ્પ પસંદ કરો. એના વિકલ્પ રૂપે આકૃતિ 12.17માં દર્શાવ્યા મુજબ *FormControl* ટૂલબાર પર *Control* આઈકોન પર ક્લિક કરી શકાય. આ ટૂલબાર ફોર્મની ડાબી બાજુ દેખાશે. (જુઓ આકૃતિ 12.2) *Properties: DateField* ડાયલોગબોક્સ ખૂલશે.



Design mode on/off Control icon

આકૃતિ 12.17 : ફોર્મકંટ્રોલ ટૂલબાર

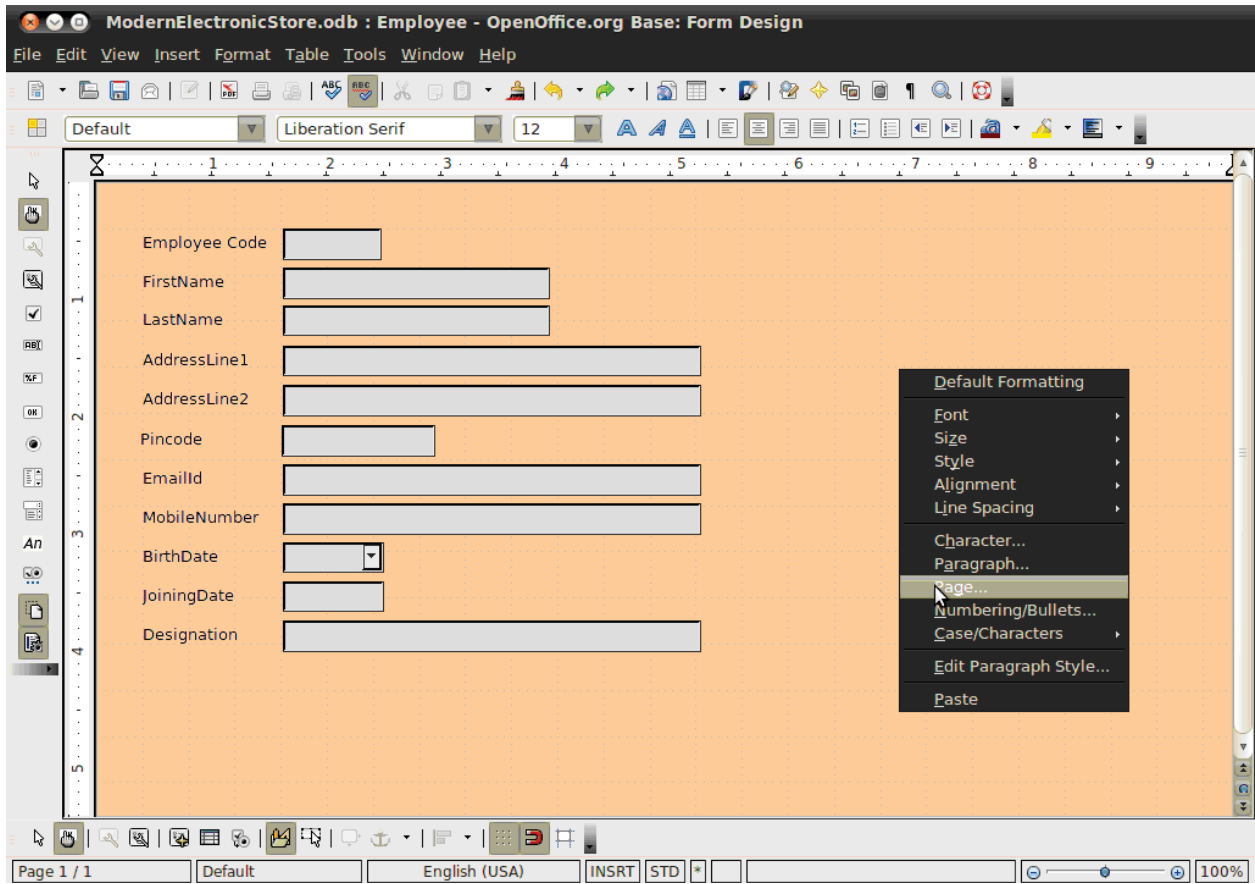
- નીચે તરફ સ્કોલ કરી *DateFormat* લાક્ષણિકતા પર જાવ. આ એક ડ્રોપડાઉન લિસ્ટ બોક્સ છે (એરો બટન પર ક્લિક કરતાં જ તેમાં ઉપલબ્ધ વિકલ્પોની યાદી નીચે ઉતરી પડે છે-તેથી તેને ડ્રોપ-ડાઉન લિસ્ટ બોક્સ કહે છે.) જેમાં સામાન્ય રીતે *Standard (short)* ગોઠવણી હોય જ છે.
- યાદી દર્શાવાય તે માટે આ *Standard (short)* પર ક્લિક કરો. આમ કર્યા પછી દર્શાવાતી યાદીમાંથી *Standard (long)* વિકલ્પ પસંદ કરો.
- નીચે તરફ સ્કોલ કરીને *Dropdown* લાક્ષણિકતા પર કર્સર લઈ જાવ. સામાન્ય રીતે તેની ગોઠવણ *No* હોય છે, તેને બદલીને *Yes* કરી દો.
- ડાયલોગબોક્સને બંધ કરી દો. તમે એ જોઈ શકશો કે, 'ટેક્સ્ટબોક્સ' હવે બદલાઈને 'ડ્રોપડાઉન' બની ગયું હશે.

નોંધ : આપણે *DesignMode On/Off* આઈકોન પર પણ ક્લિક કરી શકીએ. (આકૃતિ 12.17માં ડાબી બાજુથી બીજો આઈકોન)

બેકગ્રાઉન્ડ બદલવું

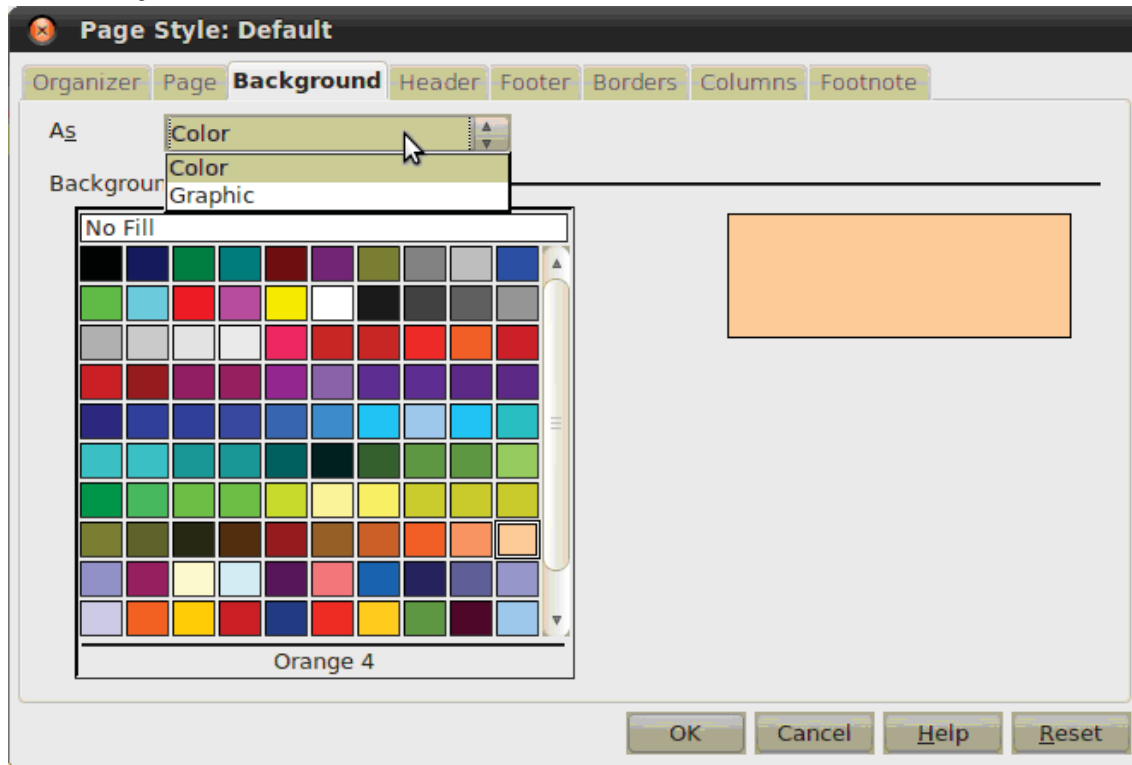
ફોર્મની પાછળના ભાગે (બેકગ્રાઉન્ડ) રંગ અથવા આકૃતિ કે ચિત્ર હોઈ શકે છે. તમે *Tools-Options-OpenOffice.org-Colors*માં ઉપલબ્ધ રંગોના કોષ્ટકમાંથી કોઈ પણ રંગનો ઉપયોગ કરી શકો. તમે બેકગ્રાઉન્ડ તરીકે કોઈ ગ્રાફિક્સ ફાઈલનો પણ ઉપયોગ કરી શકો. ફોર્મના બેકગ્રાઉન્ડને બદલવા માટે નીચે મુજબનાં પગલાં અનુસરો.

- ફોર્મમાં કોઈ ખાલી ભાગ પર માઉસનું જમણું બટન દબાવો. આમ કરવાથી આકૃતિ 12.18માં દર્શાવ્યા મુજબનું એક મેનુ દર્શાવાશે.



આકૃતિ 12.18 : Formનું સબ મેનુ

- *Page* વિકલ્પ પસંદ કરો. આમ કર્યા પછી આકૃતિ 12.19માં દર્શાવ્યા મુજબ *Page Style : Default* નામનું ડાયલોગબોક્સ દર્શાવાશે.



આકૃતિ 12.19 : PageStyle ડાયલોગબોક્સ

- આ ડાયલોગબોક્સમાંથી પસંદગી મુજબનો રંગ પસંદ કરીને OK બટન પર ક્લિક કરો. આમ કર્યા પછી ફોર્મનું બેકગ્રાઉન્ડ પસંદ કરેલ રંગ મુજબ બદલાઈ જશે.

ફોર્મના ઉપયોગ વડે રેકોર્ડ શોધવા :

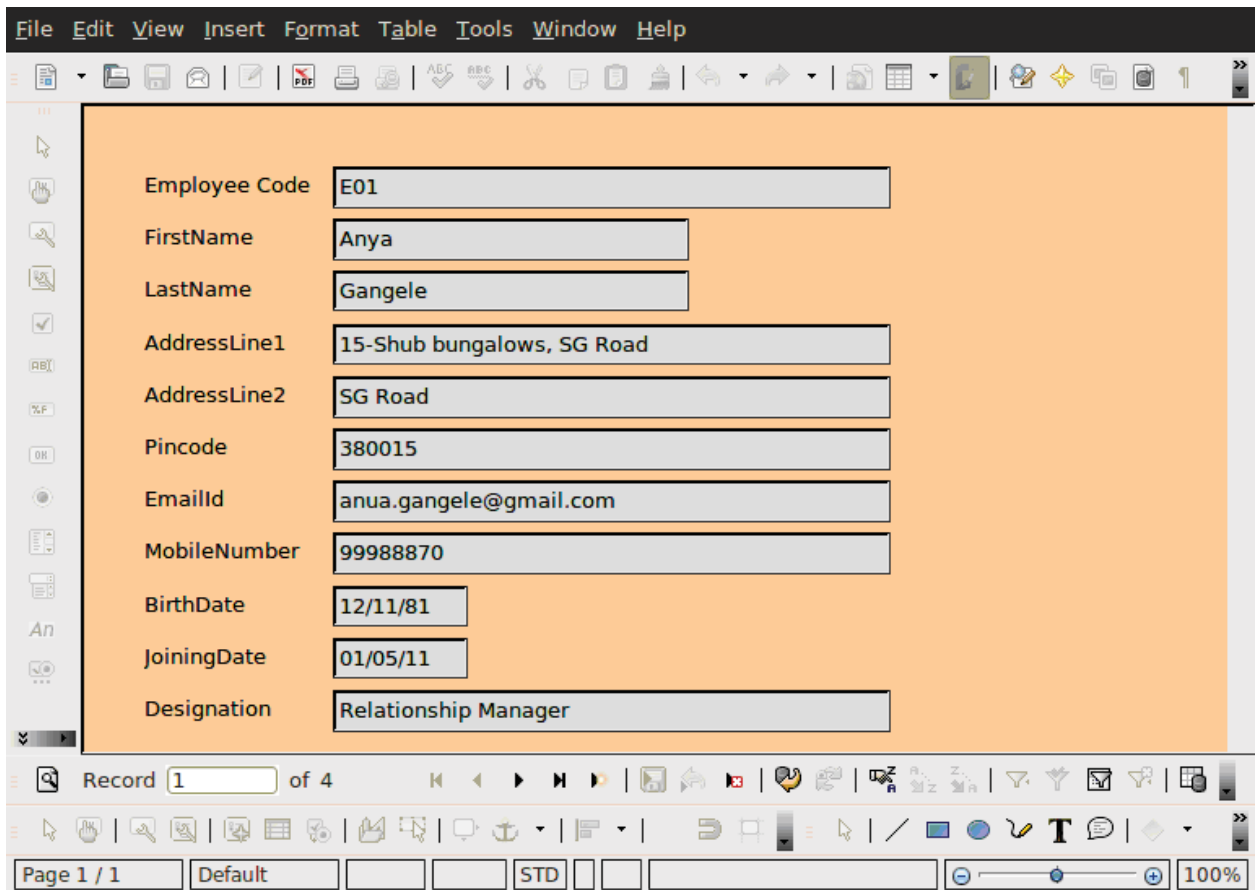
Forms ટેબ હેડબ દર્શાવાતા Employee આઈકોન પર ડબલ ક્લિક કરીને, તૈયાર કરેલું Employee ફોર્મ ખોલો. આમ કરવાથી આકૃતિ 12.20માં દર્શાવ્યા મુજબનો સ્ક્રીન દેખાશે.

આકૃતિ 12.20 : ડેટાએન્ટ્રી મોડમાં Employee ફોર્મ

અહીં એ નોંધો કે, આકૃતિ 12.20માં જ્યારે ફોર્મ ખૂલે છે, ત્યારે સ્ક્રીન પર આપમેળે પ્રથમ રેકોર્ડ દર્શાવાય છે. ફોર્મ દર્શ્યમાં એક સમયે માત્ર એક જ રેકોર્ડ જોવા મળે છે. એક પછી એક બાકીના રેકોર્ડ જોવા માટે ફોર્મના નીચેના ભાગે દેખાતાં આગળ-પાછળ જવામાં મદદરૂપ એવા (નેવિગેશન) બટનનો ઉપયોગ કરો. (જુઓ આકૃતિ 12.20)

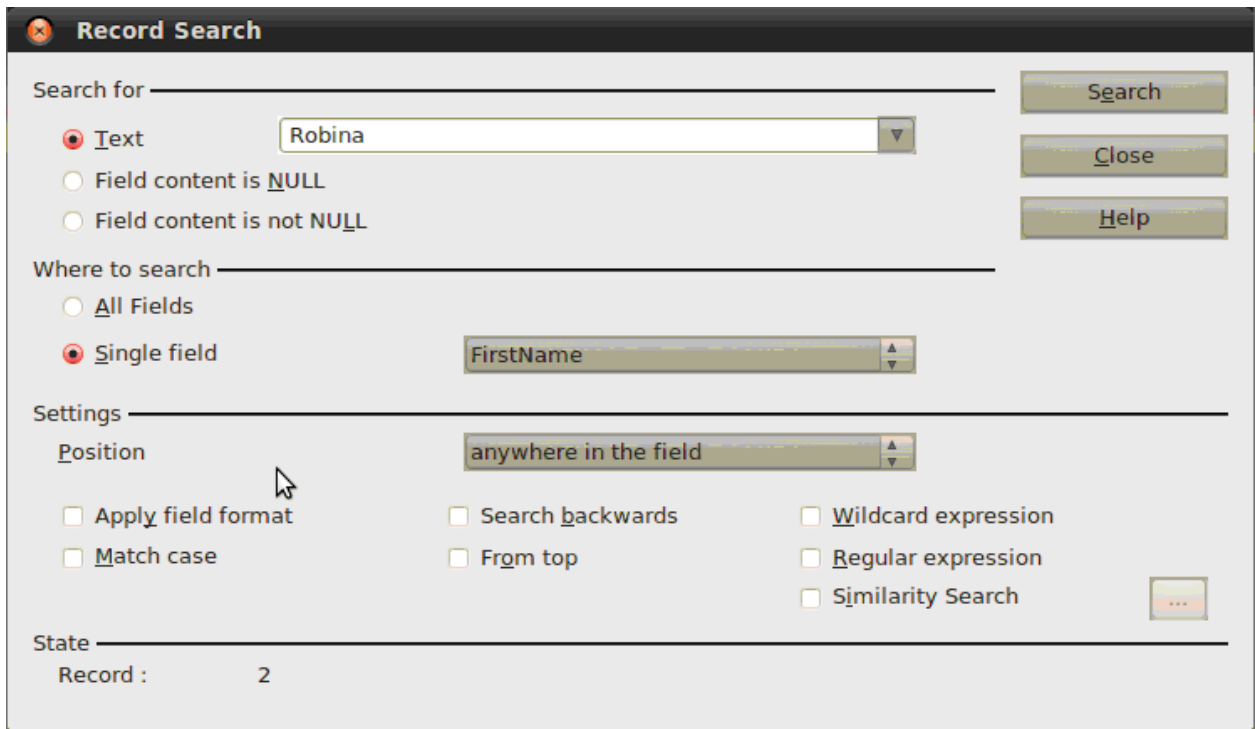
અમુક વખતે આપણે કોઈ ચોક્કસ રેકોર્ડ જોવો હોય એવું બને. આ માટે ઈચ્છિત રેકોર્ડને શોધવા માટે નીચે મુજબનાં પગલાં અનુસરો.

આકૃતિ 12.21માં દર્શાવ્યા મુજબ 'ફોર્મ નેવિગેશન બાર' પર આવેલ Find Record વિકલ્પ પસંદ કરો.



આકૃતિ 12.21 : Find Record વિકલ્પ

તમારા સ્ક્રીન પર આકૃતિ 12.22માં દર્શાવ્યા મુજબ *Record Search* નામનું એક ડાયલોગબોક્સ દર્શાવાશે.





આકૃતિ 12.22 : Record Search ડાયલોગબોક્સ

- *Where to search* શીર્ષક હેઠળ *Single field* વિકલ્પ પસંદ કરો.
- *Single field* વિકલ્પની સામેના ડ્રોપડાઉન બોક્સમાંથી *FirstName* નામનું ફિલ્ડ પસંદ કરો. (જુઓ આકૃતિ 12.22)
- હવે *Text* વિકલ્પની સામેના ડ્રોપડાઉન બોક્સમાંથી *Robina* પસંદ કરો. તમે જુઓ કે *Text* વિકલ્પની આગળ રહેલ રેડિયો બટન પસંદ થયેલું છે.
- *Search* બટન પર ક્લિક કરો.
- *Close* બટન પર ક્લિક કરો. હવે ફોર્મની તમામ જુદી-જુદી વિગત *Robina*ના રેકોર્ડમાં પડેલી વિગતો દ્વારા ભરાઈ ગઈ હશે.

ફોર્મ દર્શના ઉપયોગ દ્વારા રેકોર્ડને ઉમેરવા કે કાઢવા :

ફોર્મ દર્શના ઉપયોગ વડે ટેબલમાં નવો રેકોર્ડ ઉમેરવા નીચે મુજબનાં પગલાં અનુસરો :

- ફોર્મ દર્શ્ય મારફત જે ટેબલમાં તમારે નવા રેકોર્ડ ઉમેરવા હોય તે ટેબલને સંબંધિત ફોર્મ ખોલો.
- નેવિગેશન બાર પર *New Record Button*  પર ક્લિક કરો.
- હવે પછી ફોર્મમાં દેખાતાં ખાલી ખાનાંઓમાં સંબંધિત વિગતો ટાઈપ કરો.
- ફોર્મને બંધ કરો.
- ડેટાબેઝ વિન્ડોમાં રહેલાં *Tables* આઈકોન પર ક્લિક કરો.
- હવે ટેબલને ખોલો અને તમે જોઈ શકશો કે, તમારા દ્વારા ઉમેરવામાં આવેલ વિગત ટેબલમાં રેકોર્ડ રૂપે સચવાઈ ગઈ હશે.
- ‘ફોર્મ વ્યૂ’ ટૂલબાર પર ઉપલબ્ધ *Delete Record*  નામના બટનનો ઉપયોગ કરીને તમે ફોર્મ મારફત અનિચ્છિત રેકોર્ડ કાઢી નાંખી શકો છો. આ બટન પર ક્લિક કરવામાં આવશે, તો ફોર્મમાં હાલ દેખાતો રેકોર્ડ જતો રહેશે અને આપમેળે એની પછીના રેકોર્ડની વિગત ફોર્મમાં દર્શાવાશે. આ વિકલ્પ દ્વારા ફોર્મની સાથે-સાથે સંબંધિત ટેબલમાંથી પણ રેકોર્ડ કાઢી નાંખશે.

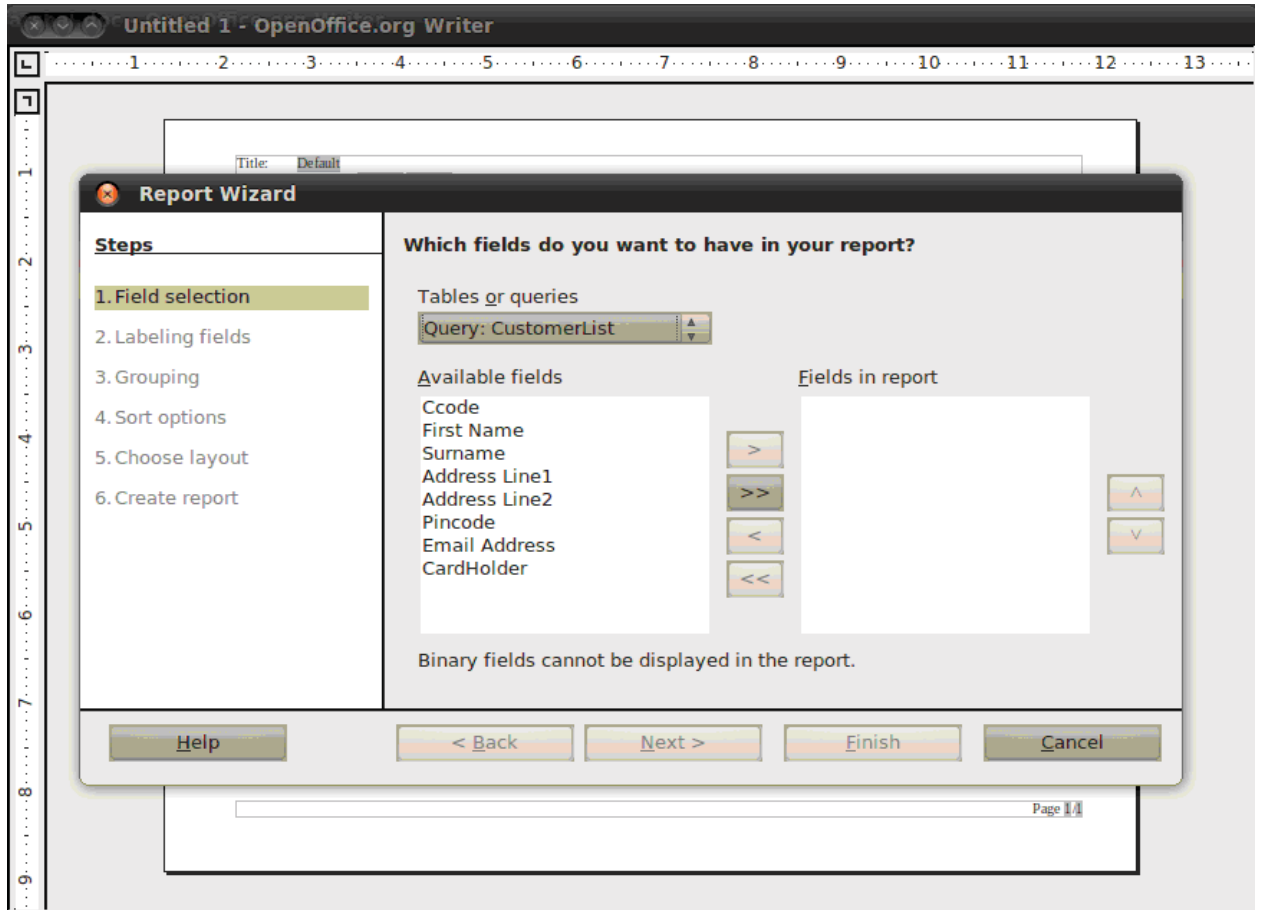
અહેવાલ (Reports)

ફોર્મ કેવી રીતે તૈયાર કરવું તે શીખ્યા બાદ, તમને હવે એ વાતની તો ખાતરી થઈ જ ગઈ હશે કે *DataSheet View* કરતાં ફોર્મ મારફત ડેટાએન્ટ્રી કરવાનું કામ વધુ સરળ છે. આપણે એ જોયું કે, ‘ક્વેરી’ (query)નો ઉપયોગ કરીને *base* માં કોઈ પણ માહિતી શોધી કાઢવી શક્ય છે. ક્વેરી દ્વારા મેળવાતું પરિણામ ભલે કોષ્ટક રૂપે મળતું હોવા છતાં તે વધુ સુઘડ તેમજ વ્યવસ્થિત ગોઠવાયેલ હોતું નથી. અહેવાલ (Report) ટેબલમાંથી મેળવાયેલ વિગતને આકર્ષક, સુઘડ અને નિર્ણાયક સ્વરૂપે રજૂ કરવાનો વિકલ્પ આપે છે. અહેવાલ તૈયાર કરવાનો એક ઉદ્દેશ પરિણામની છાપેલી નકલ (હાર્ડ કોપી) મેળવવાનો છે. જો કે, અહેવાલની રચના કરતી વખતે છાપેલી નકલ કેવી જોઈએ છે, તેના દેખાવને ધ્યાનમાં રાખવામાં આવે છે. આપણે કોઈ ટેબલ, ક્વેરી કે પછી બંનેનો ઉપયોગ કરીને અહેવાલ તૈયાર કરી શકીએ.

આ પ્રકરણમાં તૈયાર કરાયેલ અહેવાલ માત્ર કાં તો ટેબલ અથવા ક્વેરી પર આધારિત છે. જો તમે કોઈ એક કરતાં વધુ ટેબલના ફિલ્ડનો ઉપયોગ કરીને અહેવાલ બનાવવા ઈચ્છતા હોવ તો એ સલાહભર્યું છે કે તમે આ જુદાં-જુદાં ટેબલોનાં ફિલ્ડોનો સમાવેશ કરતી ક્વેરી તૈયાર કરો અને એ પછી આ ક્વેરીનો ઉપયોગ કરીને ઈચ્છાનુસારનો અહેવાલ તૈયાર કરો.

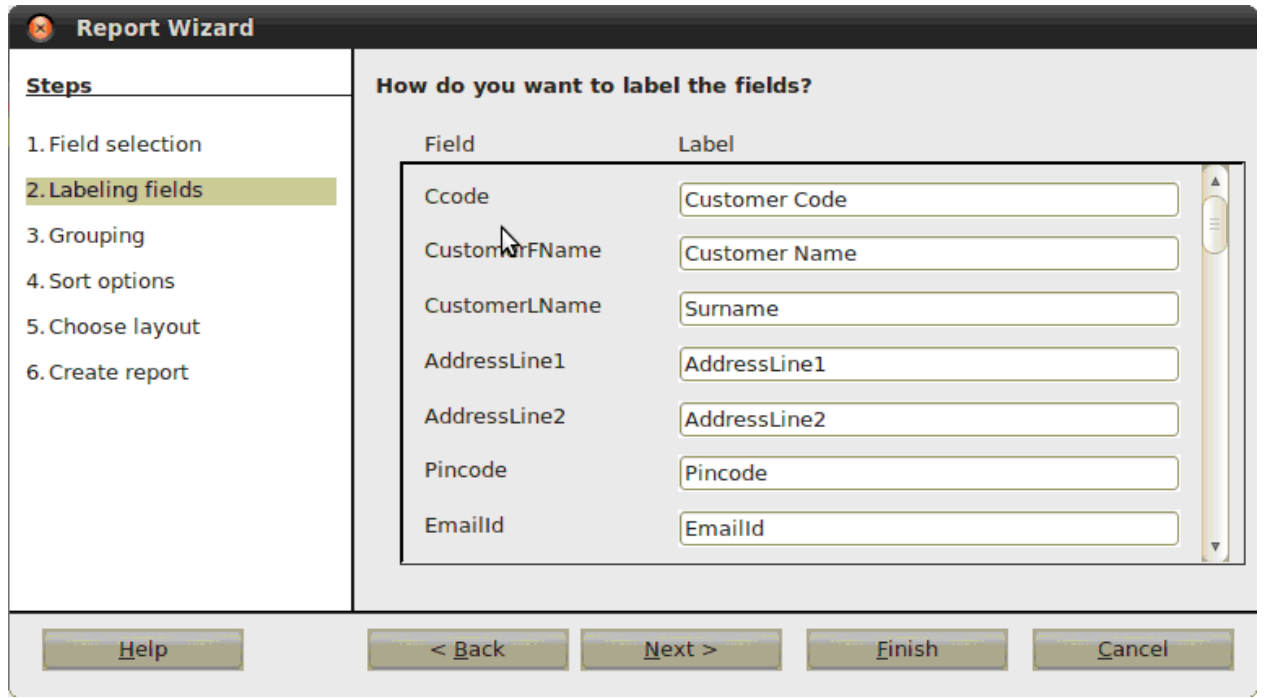
ઉદાહરણ તરીકે, અગાઉનાં પ્રકરણોમાં, આપણે ગ્રાહકનાં નામ અને સરનામાં સહિતની યાદી દર્શાવવા Customer, City, State and Country નામનાં ચાર ટેબલને સાંકળતી એક ક્વેરી તૈયાર કરી હતી. આ ક્વેરીનું નામ આપણે Customer-list આપ્યું હતું. તો ચાલો, આ જ ક્વેરીના આધારે આપણે અહેવાલ તૈયાર કરીએ. નીચે દર્શાવેલ બે રીતે Report-Wizard શરૂ કરી શકાય.

- જે ટેબલ કે ક્વેરી માટે અહેવાલ તૈયાર કરવો હોય તેના પર માઉસનું જમણું બટન દબાવ્યા બાદ દર્શાવાતા પોપ-અપ મેનુમાંથી *Report Wizard* વિકલ્પ પસંદ કરીને.
- અન્ય વિકલ્પ તરીકે ડેટાબેઝ વિન્ડોમાં *Reports* આઈકોન પર ક્લિક કરીને *Use Wizard to Create Report* વિકલ્પ પસંદ કરો.
- ઉપર દર્શાવેલ પૈકી કોઈ પણ વિકલ્પનો ઉપયોગ કરીશું તો આકૃતિ 12.23માં દર્શાવ્યા મુજબ બે ડાયલોગબોક્સ ખૂલશે. અત્યારે હાલ પૂરતું આપણે Report Wizard ડાયલોગબોક્સ પર ધ્યાન કેન્દ્રીત કરીશું.
- સૌપ્રથમ *Report Wizard* ડાયલોગબોક્સમાં *Table or Queries* નામના ડ્રોપડાઉન લિસ્ટમાંથી CustomerList નામની આપણી ક્વેરી પસંદ કરવી પડે. એ પછી *Available fields* તરીકે દર્શાવાતાં બધાં ફિલ્ડને *Fields in report* ખાનામાં લઈ આવવા >> બટનનો ઉપયોગ કરો.
- Next બટન પર ક્લિક કરો.



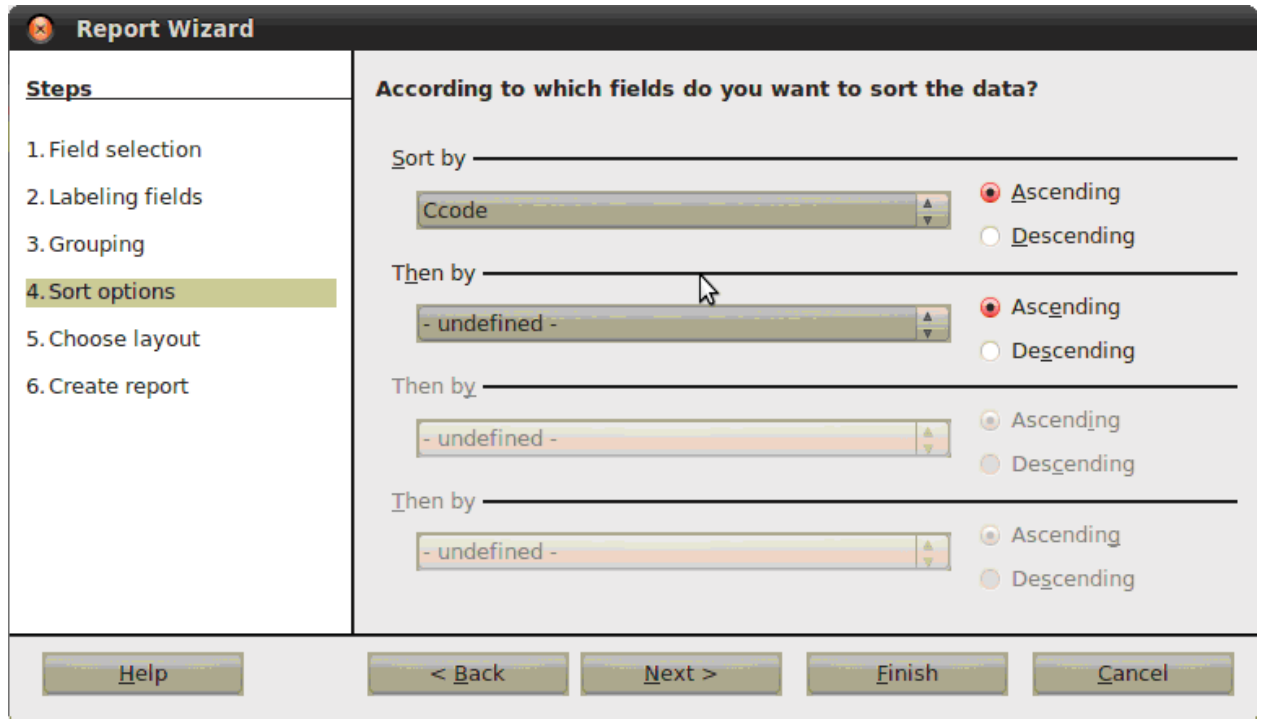
આકૃતિ 12.23 : Report wizard નું પ્રથમ પગલું

- બીજા પગલામાં, તમારી જરૂરિયાત મુજબ દરેક ફિલ્ડને યોગ્ય નામ (લેબલ) આપો. નમૂનારૂપ લેબલ આકૃતિ 12.24માં દર્શાવેલ છે.



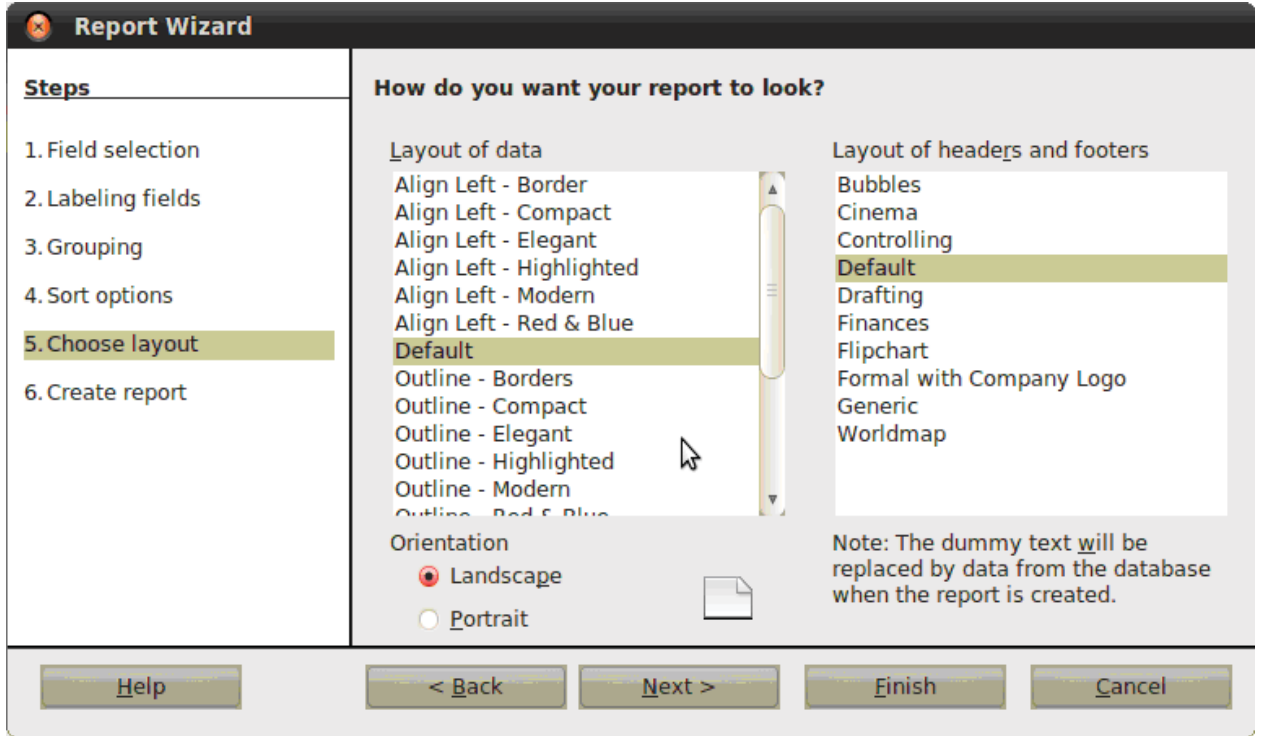
આકૃતિ 12.24 : Report Wizard નું બીજું પગલું : ફિલ્ડને લેબલ આપવાં

- Next બટન પર ક્લિક કરો.
- વિઝાર્ડનું ત્રીજું પગથિયું જુદાં-જુદાં ફિલ્ડનું જૂથ (Grouping) બનાવવાનું છે. જોકે, આપણે આ અહેવાલમાં કોઈ પણ ફિલ્ડનાં જૂથ બનાવવાનાં હોવાથી Next બટન પર ક્લિક કરો. આકૃતિ 12.25માં દર્શાવ્યા મુજબ સ્ક્રીન પર Report Wizardનું ચોથું પગલું દર્શાવાશે.



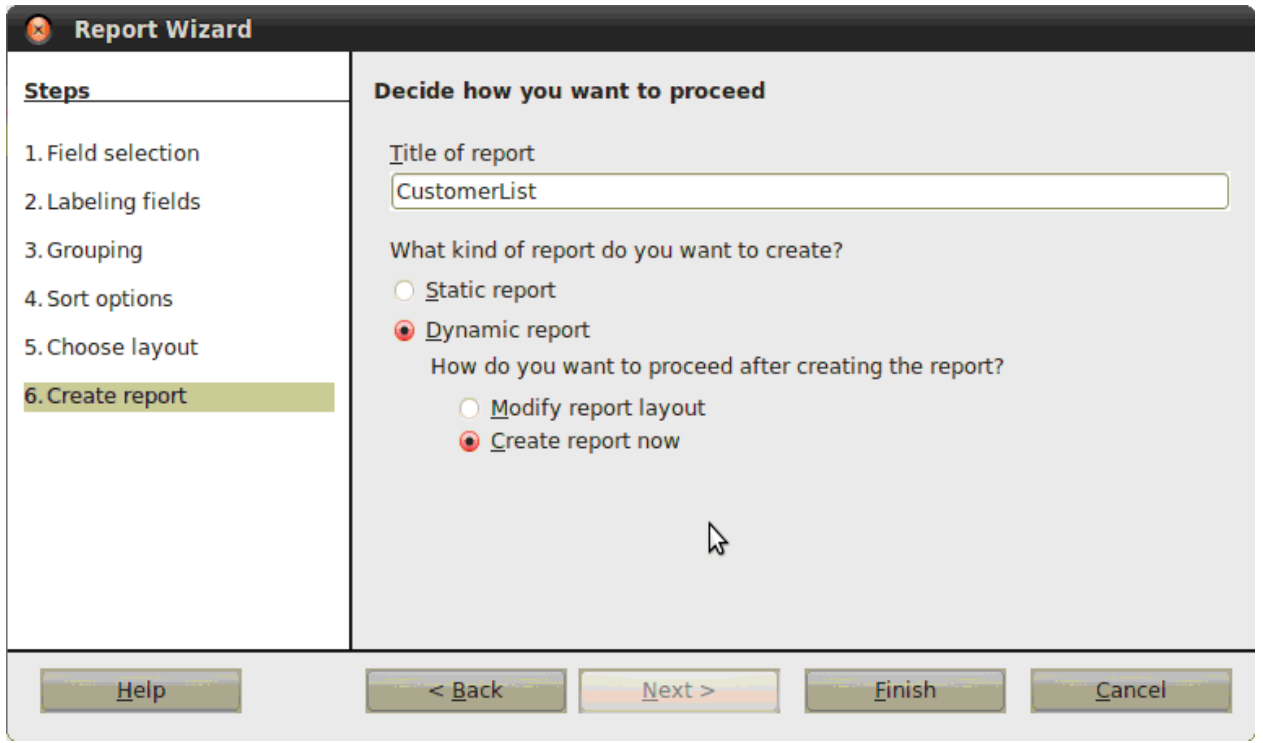
આકૃતિ 12.25 : Report Wizard ચોથું પગલું : Sort વિકલ્પ દર્શાવવા

- ચોથા પગલામાં આપણે માહિતી ગોઠવવા (sort કરવા) માટેના વિકલ્પો દર્શાવી શકીએ. ચાલો, આપણે માહિતીને Ccode ફિલ્ડના ચઢતા ક્રમમાં ગોઠવીએ. આકૃતિ 12.26માં દર્શાવ્યા મુજબ પછીનું પગલું દર્શાવાય તે માટે Next બટન દબાવો.
- Report Wizardનું પાંચમું પગલું આપણને વિવિધ રચનાઓમાંથી ઈચ્છિત રચનાની પસંદગી કરવાની તક આપે છે. રચના (Layout)માં રંગસંયોજન, ઉપરાંત ફિલ્ડના સ્થાનની ગોઠવણી અને લખાણની ગોઠવણીનો સમાવેશ થાય છે. તમે એક પછી એક આ દરેકનો ઉપયોગ કરી શકો. આપણે રચનાની સામાન્ય ગોઠવણી (Default Layout)નો ઉપયોગ કરીશું (જુઓ આકૃતિ 12.26).
- Next બટન પર ક્લિક કરો.



આકૃતિ 12.26 : Report Wizardનું પાંચમું પગલું : Choose Layout

- છઠ્ઠા પગલામાં તમને અહેવાલ માટે એક શીર્ષક આપવાનું કહેવામાં આવશે. આકૃતિ 12.27માં દર્શાવ્યા મુજબ Title of reportની નીચેના ખાના (ટેક્સ્ટ બોક્સ)માં CustomerList ટાઇપ કરો. તમે એ અવલોકન કરો, કે What kind of report do you want to create?ની નીચે બે વિકલ્પ આપવામાં આવ્યા છે. આપણા માટે તૈયાર થનાર અહેવાલ પર અહીં પસંદ કરવામાં આવતા વિકલ્પની અસર થશે. જો આપણે Static report વિકલ્પ પસંદ કરીશું, તો દર વખતે જ્યારે આપણે અહેવાલને ખોલવાનો પ્રયત્ન કરીશું, ત્યારે અહેવાલના પરિણામમાં ડેટા એનો એ જ રહેશે. એના બદલે Dynamic report વિકલ્પ પસંદ કરવાથી, દર વખતે જ્યારે-જ્યારે આપણે અહેવાલને ખોલવાનો પ્રયત્ન કરીશું ત્યારે, ટેબલમાં કરેલા કોઈ પણ ફેરફારની અસર જોવા મળશે.



આકૃતિ 12.27 : ReportWizardનું છટું પગલું : અહેવાલનું શીર્ષક આપવું.

નોંધ :

Baseમાં તૈયાર કરવામાં આવતા ઓબ્જેક્ટ એકબીજા પર આધારિત હોય છે. એનો અર્થ એ થાય કે, જો T નામના ટેબલ પર આધારિત Q નામની ક્વેરી તૈયાર કરી હોય, તો T નામના ટેબલ પર કરવામાં આવતો કોઈ પણ ફેરફાર ક્વેરી Qમાં પ્રતિબિંબિત થશે અને એ જ ઊલટા કિસ્સામાં પણ. એ જ રીતે જો આપણે Q નામની ક્વેરીને આધારે R નામનો અહેવાલ બનાવીએ, તો Qમાં કરાતો કોઈ પણ ફેરફાર R માં પણ દેખાવો જોઈએ. અલબત્ત, અહેવાલમાં ફેરફાર શક્ય નથી. *Static report* વિકલ્પ પસંદ કરવાથી ફેરફારની અસર અહેવાલ પર પ્રતિબિંબિત થશે.

- તમારી જરૂરિયાત શું છે તેના આધારે છટ્ટા પગલામાં દર્શાવેલ વિકલ્પોમાંથી કોઈ એક વિકલ્પ પસંદ કરો. આપણા કિસ્સામાં આપણે *Dynamic report* પસંદ કર્યો છે. *Create report now* વિકલ્પ પસંદ કરો.
- Finish બટન પર ક્લિક કરો અને તમને આકૃતિ 12.28માં દર્શાવ્યા મુજબનો અહેવાલ જોવા મળશે.

ModernElectronicStore.odb : CustomerList (read-only) - OpenOffice.org Base: Report Design

File Edit View Insert Format Table Tools Window Help

le:
Author: Amit Jain
Date: 8/17/12

Customer Code	Customer Name	Surname	Address-Line1	Address-Line2	Pincode	EmailId	Card-Holder	City	State-Name	CountryName
C01	Puja	Sethia	17, Akash	Jodhpur Char Rasta	380056	puja.sethia@yahoo.com	<input type="checkbox"/>	Ahmedabad	Gujarat	India
C02	Sakina	Nagingar	T-37, Abhimanu	Curry Road	384001	sakina82@gmail.com	<input checked="" type="checkbox"/>	Mehsana	Gujarat	India
C03	Joseph	Thomas	BG-1, Sunstar	Sola Road	362268	joseph@yahoo.com	<input type="checkbox"/>	Patan	Gujarat	India
C04	Prema	Agrawal	Smita Bunglows	PNT Road	380089	prema@gmail.com	<input type="checkbox"/>	Ahmedabad	Gujarat	India
C05	Harshit	Shah	17, Jaldeep-1	Bopal	380058	harshit@yahoo.com	<input checked="" type="checkbox"/>	Ahmedabad	Gujarat	India

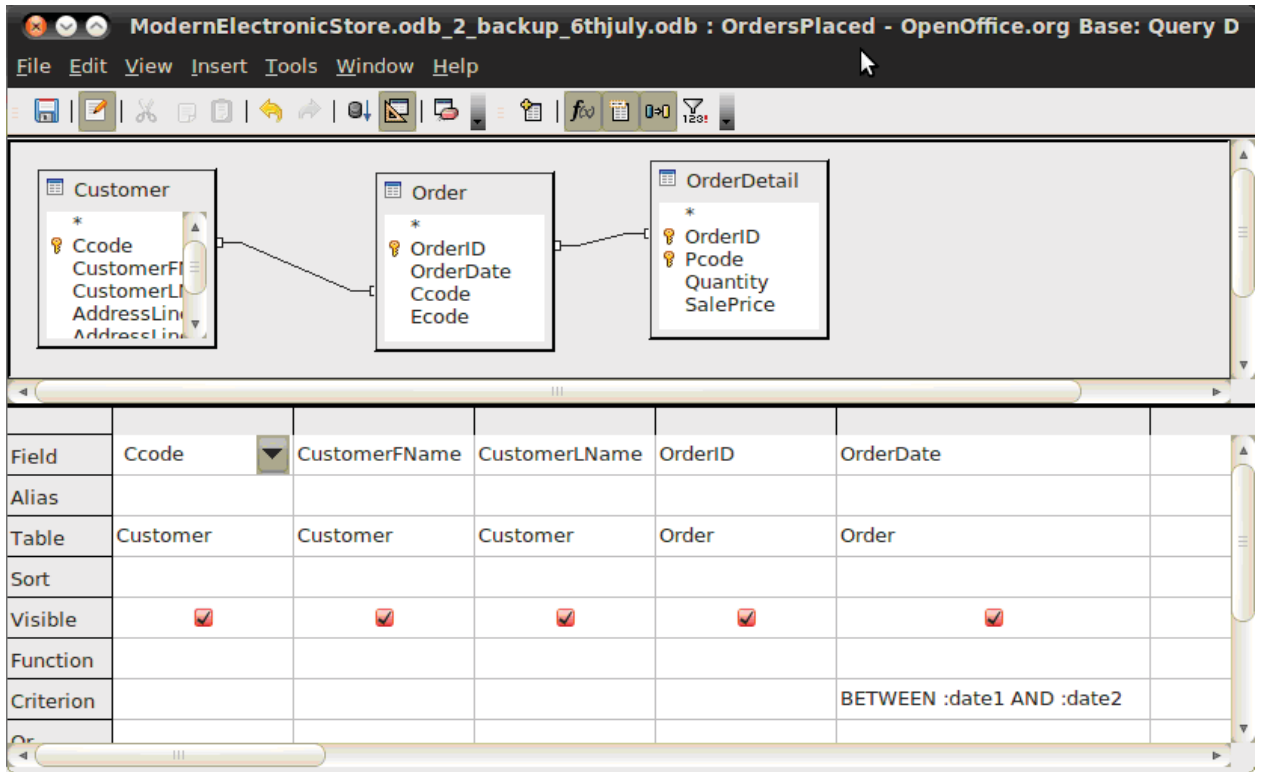
Page 1 / 1 First Page STD 100%

આકૃતિ 12.28 : CustomerAddresses ધરાવતો અહેવાલ

નોંધ : અહેવાલ-લેખકનું નામ (Author Name) અને તારીખ એ બંને મશીનનું નામ અને કઈ તારીખે તમે અહેવાલ ખોલ્યો / બનાવ્યો છે, તેના પર આધારિત છે. અહેવાલમાં સમાવિષ્ટ વિગતો તમે ટેબલમાં કરેલી ડેટાએન્ટ્રીને અનુરૂપ હશે.

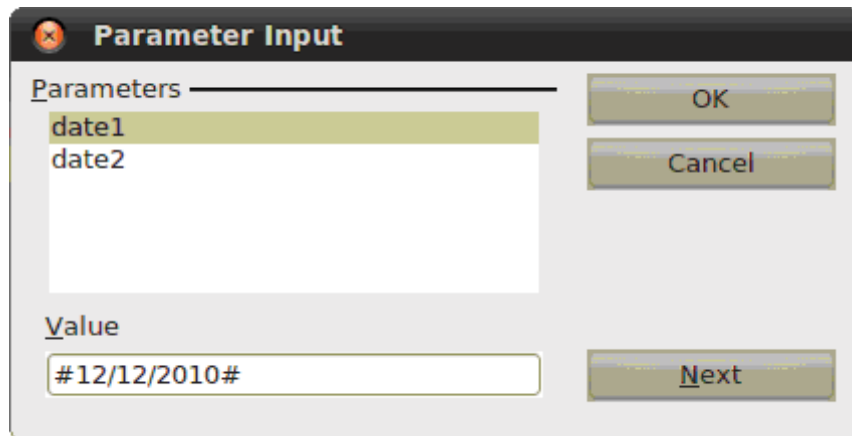
ચાલો, હવે કોઈ બે તારીખોની વચ્ચે દરેક ગ્રાહક દ્વારા મૂકવામાં આવેલ ખરીદ-ઓર્ડરોની વિગત દર્શાવતો અહેવાલ તૈયાર કરીએ. સૌપ્રથમ આપણે પેરામીટર આધારિત ક્વેરી બનાવીએ કે જેથી આપણે આપેલી કોઈ પણ બે તારીખો વચ્ચેના રેકોર્ડ ધરાવતો અહેવાલ બનાવી શકીએ. પેરામીટર આધારિત ક્વેરી આપણા અહેવાલને ગતિશીલ અહેવાલ (ડાયનેમિક રિપોર્ટ) બનાવવામાં મદદરૂપ નીવડશે. જ્યારે-જ્યારે આપણે અહેવાલ તૈયાર કરવા પ્રયત્ન કરીશું, ત્યારે આપણને બે તારીખ પૂછવામાં આવશે અને પછી દર્શાવેલી આ બે તારીખોની વચ્ચેના ડેટાને આધારે અહેવાલ તૈયાર કરવામાં આવશે.

- DesignViewમાં ક્વેરી ખોલો. Customer, Order અને OrderDetail ટેબલ ઉમેરો.
- સંબંધિત ટેબલમાંથી Ccode, CustomerFname, CustomerLname, OrderID અને OrderDate ફિલ્ડ પર ડબલ ક્લિક કરો.
- OrderDate ફિલ્ડના Criterion ખાનામાં આકૃતિ 12.29માં દર્શાવ્યા પ્રમાણે BETWEEN :date1 AND :date2 ટાઈપ કરો.



આકૃતિ 12.29 : પેરામીટર આધારિત ક્વેરી

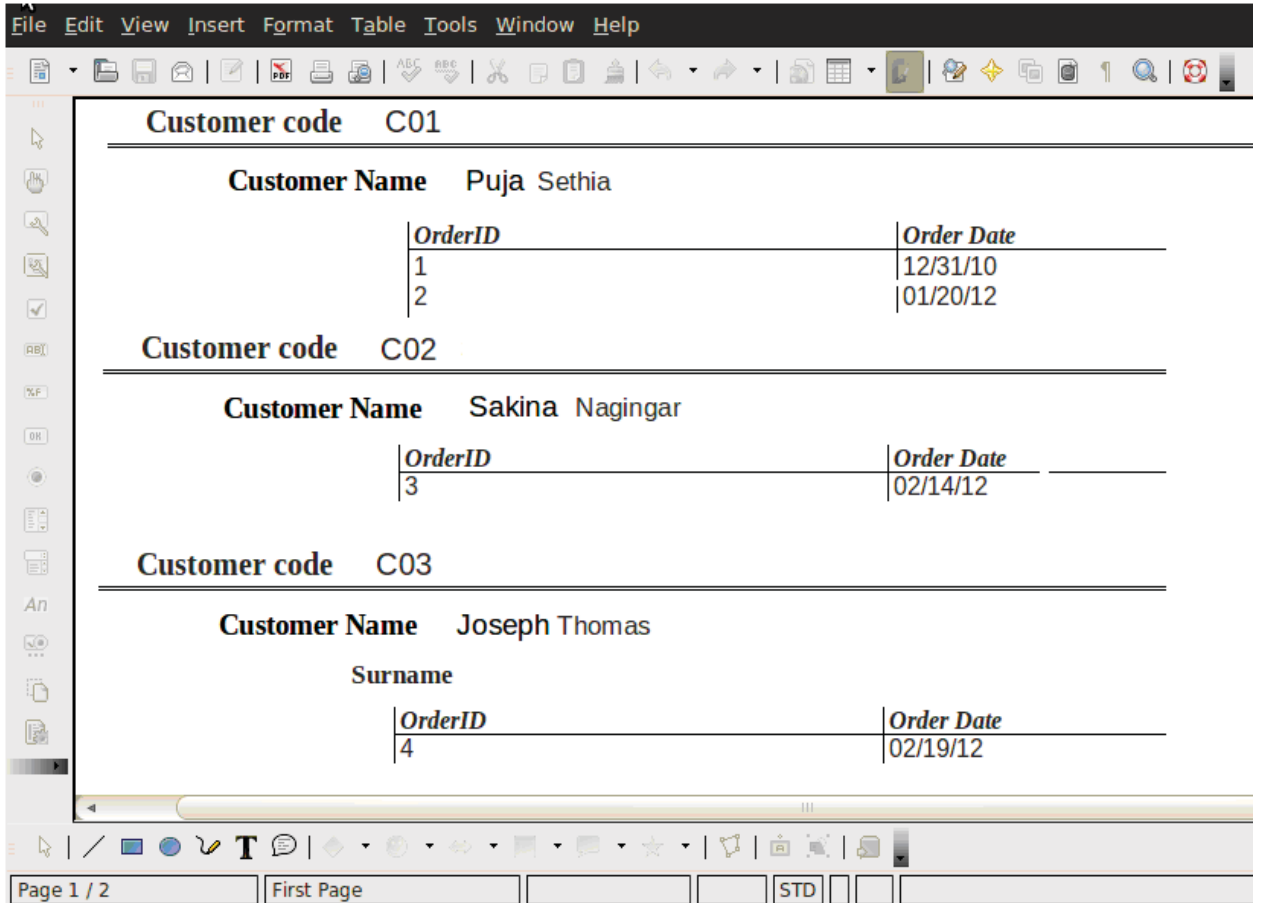
- ક્વેરીને *OrdersPlaced* નામ આપી સાચવી લો.
- એ ચાલે છે કે નહીં તે ચકાસવા પરિણામ જોવા ક્વેરી ચલાવો. ઈનપુટ તરીકે બે પેરામીટર અપેક્ષિત છે (જુઓ આકૃતિ 12.30).



આકૃતિ 12.30 : ક્વેરી માટે પેરામીટર દાખલ કરવા

- ક્વેરીને બંધ કરો.
- ક્વેરી પર માઉસનું જમણું બટન દબાવો. આમ કર્યા પછી દર્શાવાતા પોપ-અપ મેનુમાંથી *Report Wizard* વિકલ્પ પસંદ કરો.
- *Available Fields* તરીકે દર્શાવાતા બંને ફિલ્ડને *Fields in report* ખાનામાં લઈ આવવા >> બટનનો ઉપયોગ કરો. એ પછી Next પર ક્લિક કરો.
- ફિલ્ડના નામને યોગ્ય ઉપનામ aliases આપો. એ પછી Next પર ક્લિક કરો.

- ReportWizardના grouping દર્શાવવાના પગલામાં Ccode ફિલ્ડ પર ક્લિક કરો, જેથી તે હાઈલાઈટ થાય. તે પછી > બટન દબાવી ફિલ્ડને Groupings listમાં સામેલ કરો. એ જ રીતે CustomerFname અને CustomerLname ફિલ્ડને આધારે રેકોર્ડ ગોઠવાઈ જશે. Next બટન ક્લિક કરો.
- તમે એ જોઈ શકશો કે Ccode, CustomerFName અને CustomerLName ફિલ્ડને આધારે રેકોર્ડ આપમેળે ગોઠવાઈ જશે. Next બટન પર ક્લિક કરો.
- Layout પસંદ કરી Next બટન પર ક્લિક કરો.
- અહેવાલની ફાઈલ માટે સૂચવવામાં આવેલ ફાઈલનું નામ ઉપયોગમાં લો. અહીં ક્વેરીના નામ મુજબનું જ નામ હશે.
- *Dynamic report* વિકલ્પ પસંદ કરો.
- Finish બટન ક્લિક કરો અને તમે જોશો કે Input Parameter ડાયલોગબોક્સ ખૂલી ગયું હશે.
- date1 and date2 માટે બે પેરામીટર આપો. આમ કર્યા પછી આકૃતિ 12.31માં દર્શાવ્યા મુજબ આપેલી બે તારીખો વચ્ચે મૂકવામાં આવેલ ઓર્ડરનો અહેવાલ દેખાશે.



Customer code C01	
Customer Name Puja Sethia	
OrderID	Order Date
1	12/31/10
2	01/20/12
Customer code C02	
Customer Name Sakina Nagingar	
OrderID	Order Date
3	02/14/12
Customer code C03	
Customer Name Joseph Thomas	
Surname	
OrderID	Order Date
4	02/19/12

આકૃતિ 12.31 : પેરામીટર આધારિત અહેવાલ

એક વાર અહેવાલ તૈયાર થઈ જાય એ પછી, એની જરૂર મુજબ છાપેલી નકલ (હાર્ડ કોપી) અથવા કમ્પ્યૂટર વાંચી શકે તેવી (સોફ્ટ કોપી) બનાવી શકીએ. હાર્ડ કોપી બનાવવા માટે સ્ટાન્ડર્ડ ટૂલબાર પર આપવામાં આવેલ print (🖨) બટનનો ઉપયોગ કરીને મેળવી શકીએ. એના બદલે જો સોફ્ટકોપી બનાવવી હોય તો સ્ટાન્ડર્ડ ટૂલબાર પરથી PDF (📄) બટન પર ક્લિક કરો. આમ કરવાથી અહેવાલ PDF ફાઈલ સ્વરૂપે સાચવવામાં આવે છે.

સારાંશ

કોઈ પણ વિનિયોગ (એપ્લીકેશન) માટે Forms અને Reports એ ઓબ્જેક્ટ છે, જેની સાથે ઉપયોગકર્તા કામ કરે છે. આ પ્રકરણમાં આપણે વ્યાવસાયિક ઢબે ફોર્મ મારફત ટેબલમાં ડેટાએન્ટ્રી કરવા અંગે ચર્ચા કરી. આપણે ટેબલ તેમજ ક્વેરીનો ઉપયોગ કરી અહેવાલ તૈયાર કરવાની કામગીરીની પણ ચર્ચા કરી.

સ્વાધ્યાય

1. ફોર્મ એટલે શું ? ફોર્મ શા માટે બનાવવાં પડે ?
2. ફોર્મ પરના નેવિગેશન બટન શેના માટે ઉપયોગમાં લેવાય છે ?
3. ફોર્મ બનાવતી વખતે Base દ્વારા આપવામાં આવતી સામાન્ય (Default) રચના (લે-આઉટ) અને ઢબ (સ્ટાઇલ)ની યાદી આપો.
4. અહેવાલ એટલે શું ? અહેવાલની ઉપયોગિતા લખો.
5. ફોર્મ અને અહેવાલ વચ્ચેનો તફાવત શું છે ?
6. અહેવાલ તૈયાર કરતી વખતે Base દ્વારા પૂરી પાડવામાં આવતી ડિફોલ્ટ રચનાઓ (લે-આઉટ) અને ઢબ (સ્ટાઇલ)ની યાદી આપો.
7. હેતુલક્ષી પ્રશ્નો
 - (1) કયા મુખ્ય ઉદ્દેશ સાથે ફોર્મ તૈયાર કરવામાં આવે છે ?
 - (a) વિગતો દર્શાવવા માટે
 - (b) વિગતો દાખલ કરવા માટે
 - (c) ટેબલ તૈયાર કરવા માટે
 - (d) ક્વેરીનું પરિણામ નિહાળવા માટે
 - (2) કયા મુખ્ય ઉદ્દેશ સાથે અહેવાલ તૈયાર કરવામાં આવે છે ?
 - (a) વ્યાવસાયિક ઢબે માહિતી રજૂ કરવા.
 - (b) વિગતો દાખલ કરવા માટે.
 - (c) ટેબલ તૈયાર કરવા માટે.
 - (d) રેકોર્ડના ઉપયોગની આંકડાકીય માહિતી માટે.
 - (3) નીચેના પૈકી શેના માટે FormWizard ડેટાએન્ટ્રી માટેનું ફોર્મ તૈયાર કરવા દે છે ?
 - (a) અહેવાલ
 - (b) ટેબલ
 - (c) ક્વેરી
 - (d) b અને c બન્ને
 - (4) ફિલ્ડને પસંદ કરવા માટે કયો વિકલ્પ ઉપયોગમાં લેવાય છે ?
 - (a) CTRL + Click
 - (b) ક્લિક
 - (c) ડબલ ક્લિક
 - (d) રાઈટ ક્લિક
 - (5) ઉપયોગકર્તા દ્વારા તૈયાર કરવામાં આવતી સહાયરૂપ શબ્દો ધરાવતી 'ટૂલટીપ' સામાન્ય રીતે નીચે પૈકી કયા ઓબ્જેક્ટ સાથે સંકળાયેલી છે ?
 - (a) ટેબલ
 - (b) ફોર્મ
 - (c) ફોર્મ પરનું નિયંત્રક ફિલ્ડ
 - (d) અહેવાલ

(6) એકસાથે કેટલા ટેબલમાંથી અહેવાલ તૈયાર કરવા Report Wizard ઉપયોગમાં લેવાય છે ?

(a) એક

(b) બે

(c) ત્રણ

(d) ત્રણથી વધુ

પ્રાયોગિક સ્વાધ્યાય

1. પ્રકરણ 9 અને 10માં School Management System માટે તૈયાર કરવામાં આવેલ બધાં ટેબલ માટે ફોર્મ તૈયાર કરો.
2. નીચે મુજબના અહેવાલો તૈયાર કરો. :
 - કોઈ ચોક્કસ સત્ર અને વર્ષ માટે વિદ્યાર્થીનું ગુણપત્રક
 - દર માસે શાળા દ્વારા ઉઘરાવાયેલ ફી
 - દરેક ધોરણમાં અભ્યાસ કરતાં કુલ વિદ્યાર્થીઓની સંખ્યા
 - દરેક ધોરણમાં ભણાવવામાં આવતા વિષયો અને તેને ભણાવતા શિક્ષકોની યાદી
 - દર માસે પ્રત્યેક વિદ્યાર્થીની કુલ હાજરીનો અહેવાલ





પ્રસ્તાવના

ટેકનોલોજી આજે માનવજીવનનો એક અંતર્ગત ભાગ બની ગઈ છે. માનવજીવનનું કોઈ પણ પાસું તેની અસરથી પ્રભાવિત થયા વિનાનું નથી. જે ગતિએ ટેકનોલોજી બદલાઈ રહી છે, તે અભૂતપૂર્વ છે. દર થોડાક માસમાં બજારમાં કોઈ નવી ટેકનોલોજી આવી ગઈ હોવાનું આપણે સાંભળીએ છીએ. આ પ્રકરણમાં આપણે કમ્પ્યૂટિંગ, વસ્તુની ઓળખ, ડિજિટલ ફોટોગ્રાફી અને ડેટા-સંગ્રહના ક્ષેત્રે કેટલીક ઊભરતી ટેકનોલોજી વિશે જોઈશું.

કમ્પ્યૂટિંગ ક્ષેત્રે ઊભરતો પ્રવાહ

કમ્પ્યૂટરના ઉપયોગ દ્વારા આપણે જે ક્રિયાઓ કરીએ છીએ, તેને સામાન્ય રીતે ‘કમ્પ્યૂટિંગ’ (computing) તરીકે ઓળખવામાં આવે છે. જુદા જુદા ઉપયોગકર્તાએ ભલે કાર્યો અને ક્રિયાઓ જુદી-જુદી હોય તેમ છતાં પ્રક્રિયા કરી શકતા, માહિતી સંગ્રહી શકતા અને અન્ય સાધનો સાથે માહિતીસંચાર કરી શકતાં ઇલેક્ટ્રોનિક સાધનોનો ઉપયોગ એ ક્રિયા કે કાર્ય પાર પાડવા માટે અગત્યનો ભાગ ભજવે છે.

આપણે બધા કોઈક ને કોઈક સમયે કમ્પ્યૂટરના પ્રકારોના પરિચયમાં આવ્યા હોઈશું. આ પ્રકારોમાં સુપર કમ્પ્યૂટર, મેઈનફ્રેમ કમ્પ્યૂટર, મીની કમ્પ્યૂટર, માઈક્રો કમ્પ્યૂટર અને મોબાઈલ કમ્પ્યૂટરનો સમાવેશ થાય છે. પ્રથમ ચાર પ્રકારનાં કમ્પ્યૂટરને કાર્યરત રાખવા માટે ખાસી એવી જગ્યાની જરૂર પડે છે. આજકાલ મોબાઈલ કમ્પ્યૂટરોએ કમ્પ્યૂટરના ઉપયોગો બાબતે ક્રાંતિ આણી છે અને તેને ખૂબ જ ઓછી જગ્યાની જરૂર પડે છે.

લેપટોપ કમ્પ્યૂટર અને ઇલેક્ટ્રિક વાયરના ઉપયોગ વગરની સંચાર વ્યવસ્થા (વાયરલેસ કમ્યુનિકેશન સિસ્ટમ)ને લીધે ચલાયમાનતા (મોબિલિટી) પર મોટી અસર પડી છે. પર્સનલ ડિજિટલ આસિસ્ટન્ટ (ટૂંકમાં PDA તરીકે ઓળખાતા) નાના મોબાઈલ કમ્પ્યૂટિંગ ડિવાઈસ અસ્તિત્વમાં આવ્યાં હતાં, છતાં તેની ઊંચી કિંમત અને મર્યાદિત સવલતને લીધે તેની ભારે અસર વર્તાઈ ન હતી. બ્લ્યુ-ટૂથ, વાઈ-ફાઈ (Wi-Fi), GPS વગેરેની ઉપલબ્ધતા અને GSMના ટૂંકા નામે ઓળખાતી ગ્લોબલ સિસ્ટમ ફોર મોબાઈલ કમ્યુનિકેશન, કોડ ડિવિઝન મલ્ટિપલ એક્સેસ (CDMA), જનરલ પેકેટ રેડિયો સર્વિસ (GPRS) અને હવે થર્ડ જનરેશન (3G) જેવી સેલ્યુલર ડેટાની સેવાઓને કારણે હવે ચિત્ર સંપૂર્ણપણે બદલાઈ ગયું છે.

આ ટેકનોલોજીએ નેનો-ટેકનોલોજીની સાથે સૂક્ષ્મ સાધનોનો વિકાસ કર્યો છે. સ્માર્ટફોન અને ટેબ્લેટ્સ (tablets) એ આ યુગની મહાન શોધ છે. આ વિભાગમાં આપણે અલગ ચીલો પાડનાર આવાં બે કમ્પ્યૂટિંગ સાધનો, ‘મોબાઈલ કમ્પ્યૂટિંગ’ અને ‘ક્લાઉડ કમ્પ્યૂટિંગ’ (Cloud computing) વિશે ચર્ચા કરીશું. આ બંને કમ્પ્યૂટિંગ પ્રવાહોએ કમ્પ્યૂટરના ઉપયોગોનું આખું દૃશ્ય બદલી નાખ્યું છે.

મોબાઈલ કમ્પ્યૂટિંગ (Mobile Computing)

સ્માર્ટ મોબાઈલ કમ્પ્યૂટિંગ સાધનો એ ખરેખર સમાજના સમયમાં બાજી પલટી નાંખનારાં સાધનોમાંનાં એક છે. કોઈ એક લાક્ષણિક મોબાઈલ કમ્પ્યૂટિંગ સાધનમાં ત્રણ પ્રાથમિક ભાગ હોય છે : હાર્ડવેર, સોફ્ટવેર અને પ્રત્યાયન (કમ્યુનિકેશન) આજના મોબાઈલ સાધનો આંતરપ્રસ્થાપિત (ઈનબિલ્ટ) ઓપરેટિંગ સિસ્ટમ અને કેટલાક ઉપયોગી સોફ્ટવેર (એપ્લિકેશન) સાથે જ મળે છે. મોબાઈલ ઓપરેટિંગ સિસ્ટમનાં કેટલાંક ઉદાહરણ તરીકે એન્ડ્રોઈડ (Android), iOS સિમ્બિયન (Symbian) વગેરે ગણાવી શકાય છે. મોબાઈલમાં ઉપલબ્ધ એપ્લિકેશન આપણને રમતો, સંવાદી (ઈન્ટરેક્ટિવ) પ્રોગ્રામ, અખબારોનાં

લખાણ, પુસ્તકો, સામયિકો વગેરેને ડાઉનલોડ (વેબ પરથી આપણા કમ્પ્યુર કે મોબાઇલ ઉતારવાની ક્રિયા) કરવામાં મદદરૂપ નીવડે છે. આગળની વાતમાં ચર્ચેલ શબ્દ પ્રત્યાયન (કમ્પ્યુનિકેશન) એ મોબાઇલ સાધન દ્વારા ઈન્ટરનેટ સાથે જોડાણ અને ફોન દ્વારા સંદેશાની આપ-લે જેવી સક્ષમતાના સંદર્ભે છે.

મોબાઇલ સાધનો હવે સર્વવ્યાપી બન્યાં છે અને વિશ્વના છેવાડાના વિસ્તાર સુધી પહોંચી રહ્યાં છે. આ મોબાઇલ સાધનો વસતિના વિશાળ સમુદાયને અમર્યાદિત સક્ષમતાઓ પ્રદાન કરવા સાથે હવે તે આપણા સમયનું અનિવાર્ય સાધન બની ગયાં છે. કોઈ એક ચોક્કસ જગ્યાએ બંધાઈ જવાને બદલે ગમે ત્યાંથી આપણે આપણા કામને આગળ ધપાવી શકીએ છીએ, તે મોબાઇલ સાધનોએ આપેલો મોટો ફાયદો છે. આ ટેકનોલોજી ઉપલબ્ધ થઈ તે પહેલાં, વ્યક્તિએ કોઈ ચોક્કસ સ્થાન પર ઉપસ્થિત રહેવું પડતું હતું. જોકે, આજે વિવિધ પ્રકારનાં મોબાઇલ કમ્પ્યૂટિંગ સાધનોની ઉપલબ્ધતાને લીધે કોઈ પણ વ્યક્તિ વિશ્વના કોઈ પણ ખૂણે રહીને પોતાને સોંપેલું કામ પાર પાડી શકે છે.

વાઈ-ફાઈ અને 3G જેવી વિસ્તૃત દૂરસંચાર નેટવર્ક (Wide-area Telecommunications network) ટેકનોલોજીના ટેકાને કારણે મોબાઇલ કમ્પ્યૂટિંગ પ્રવાહને અદ્ભુત સફળતા મળી છે. આ ટેકનોલોજી આપણને વિશ્વના કોઈ પણ ખૂણેથી બાકીના સમગ્ર વિશ્વ સાથે જોડાવાની તક આપે છે. તે આપણને WWWના ટૂંકા નામે ઓળખાતા વર્લ્ડ-વાઈડ વેબ (World Wide Web) સાથે અસ્ખલિત સંપર્ક બનાવી આપી અદ્યતન માહિતીનો ઉપયોગ શક્ય બનાવી સક્ષમ બનાવે છે. આજના મોટા ભાગના મોબાઇલ નેટવર્કમાં જે કેટલીક ચોક્કસ સવલતો મળે છે. તેમાં શોર્ટ મેસેજ સર્વિસ (SMS), જનરલ પેકેટ રેડિયો-સર્વિસ (GPRS), મલ્ટિમીડિયા મેસેજિંગ સર્વિસ (MMS), બ્લ્યુ-ટુથ (Bluetooth) અને તારલેસ એપ્લિકેશન પ્રોટોકોલ (WAP)નો સમાવેશ કરી શકાય. ચાલો, આપણે આમાંની કેટલીક ટેકનોલોજી વિશે થોડી માહિતી મેળવીએ.

વાઈ-ફાઈ (Wi-Fi)

વાઈ-ફાઈ એ તાર વિનાનો (વાયરલેસ) નેટવર્કિંગ પ્રોટોકોલ છે, જે વાઈ-ફાઈથી સજ્જ સાધનોને ભૌતિક રીતે કેબલથી જોડ્યા વગર પર માહિતીનું આદાન-પ્રદાન (પ્રત્યાયન) કરવા દે છે. જે 802.11 IEEE નેટવર્ક સ્ટાન્ડર્ડ પર આધારિત સ્થાનિક (લોકલ એરિયા) તારલેસ નેટવર્ક પ્રોટોકોલ માટે વપરાતો આ એક ઉદ્યોગનો શબ્દ (ઈન્ડસ્ટ્રી ટર્મ) છે. આજે, મર્યાદિત વિસ્તારમાં તાર વિના સાધનો વચ્ચે માહિતીની આપ-લે કરવા ઉપયોગમાં લેવાતું સૌથી વધુ પ્રચલિત પ્રત્યાયનનું માધ્યમ છે. આ ટેકનોલોજીએ વ્યક્તિગત તેમજ કામચલાઉ નેટવર્ક સ્થાપવા માટેનો પાયો રચવાનું કામ કર્યું છે. સમયાંતરે, વાઈ-ફાઈનાં જુદાં-જુદાં ધોરણો જેવાં કે 802.11a, 802.11b, 802.11g અને 802.11n અસ્તિત્વમાં આવ્યાં. આ ચારેય વિભિન્ન ધોરણો માન્ય ડેટા-રેટ (Data Rate) અને સંકેતો કેટલા અંતર સુધી પહોંચી શકે તે બાબતોની ભિન્નતાને બાદ કરતાં લગભગ એકસરખાં જ છે. 802.11a સૌથી ઓછો ડેટા-રેટ આપે છે. (6 Mbits/sથી 54 Mbits/s) અને તેના સંકેતો પહોંચવાની મર્યાદા 120 મીટર જ છે, જ્યારે બીજી બાજુ 802.11n સૌથી ઊંચો ડેટા-રેટ આપે છે. (15 Mbits/sથી 150 Mbits/s) અને તેના સંકેતો પહોંચવાની મર્યાદા 250 મીટર છે. આજે તમે જો મોબાઇલ ખરીદો, તો તેમાંથી મોટા ભાગના મોબાઇલ વાઈ-ફાઈ સવલતવાળા જ હોય છે.

જનરલ પેકેટ રેડિયો સેવાઓ (GPRS)

મોબાઇલ સાધનોના ઉપયોગમાં વધારો થયો છે, એટલે સ્વાભાવિક રીતે જ ઉપયોગ કરનારાઓની અપેક્ષા પણ વધી છે. આ અપેક્ષાની ભૂખમાં GPRS એ જમાનાને અનુરૂપ ક્રાંતિકારી ટેકનોલોજી પુરવાર થઈ છે. ‘પેકેટ સ્વિચ ટેકનોલોજી’ (Packet Switched Technology)ને કારણે ડેટા-પ્રત્યાયન (ડેટા-કમ્પ્યુનિકેશન) શક્ય બનવાને લીધે મોબાઇલની ઉપયોગિતા એક સ્તર ઊંચી થઈ. મોબાઇલ ફોન દ્વારા ઈન્ટરનેટ જોડાણ સ્થાપવા માટે આ ટેકનોલોજી ઉપયોગી છે. મૂળભૂત રીતે, કોઈ પણ નેટવર્ક જોડાણ જે અવાજ અથવા શાબ્દિક સંદેશા નથી તે GPRS જેવા ડેટા જોડાણનો ઉપયોગ કરતા હોય છે. તે સરકીટ-સ્વિચ ટેકનોલોજી કરતાં દસ ગણી વધુ ઝડપ (ડેટાસ્પીડ) આપે છે. સૈદ્ધાંતિક રીતે નિર્ધારિત ઝડપ 115kbit/s છે, જો કે ખરેખર વ્યવહારું ઝડપ 30-40 Kbpsની આસપાસ મળે છે.

આ ટેકનોલોજીનો ઉપયોગ કરીને ઉપયોગકર્તા હંમેશા માટે ઈન્ટરનેટ સાથે જોડાયેલ હોઈ શકે.

3G અને 4G

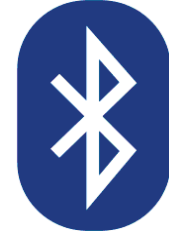
3G અને 4G પદ મોબાઇલનાં પ્રત્યાયન-ધોરણો (મોબાઇલ કમ્યુનિકેશન સ્ટાન્ડર્ડ)ની ત્રીજી અને ચોથી પેઢીના સંબંધે પ્રયોજાય છે. 1G સેવા એનાલોગ સંકેતો (Analog signals) પર આધારિત હતી અને માંડ 19.2 Kbpsની ડેટાઝડપ સુધી પહોંચી શકી હતી. મોબાઇલ સાધનોની લોકપ્રિયતા વધવાને પરિણામે મોબાઇલ ધારકોની સંખ્યા પણ વધી. આ વિકાસ 'ડિજિટલ નેટવર્કિંગ' (Digital Networking)ને જન્મ આપ્યો અને એના ઉપયોગ દ્વારા આવેલી સેવાઓ 2G તરીકે ઓળખાઈ. તેની નેટવર્કક્ષમતા વિસ્તૃત હતી. અને અંદાજે 1.2 Mbps જેટલી ડેટાઝડપ હતી.

હાલમાં, આપણે 3G અને 4G નેટવર્કિંગનો ઉપયોગ કરીએ છીએ. 3G સેવાઓ ઝડપી ડેટા-તબદીલી (ડેટા-ટ્રાન્સફર) પ્રદાન કરે છે, લગભગ 3 Mbps સુધીની ઝડપ. તે શક્તિશાળી મલ્ટિમીડિયા સેવાઓ પ્રદાન કરે છે, જે કોઈ પણ સંસ્થાને 'ઓનલાઇન બિલિંગ સિસ્ટમ' અને વીડિયો-કોન્ફરન્સિંગ જેવા ઉપયોગી પ્રોગ્રામ તૈયાર કરવા સમર્થ બનાવે છે. ભરપૂર ગ્રાફિક્સ અને એનિમેશન ધરાવતી ઘણી બધી લોકપ્રિય મોબાઇલ રમતો માટે પણ મંચ પૂરા પાડે છે. આ ટેકનોલોજીનો ઉપયોગ કરીને મોબાઇલ ટીવી, તત્કાલ સંદેશા (Instant Messaging) અને વીડિયોચર્ચા (Video Chatting) માટેના પ્રોગ્રામ (એપ્લિકેશન) તૈયાર કરવાનું પણ શક્ય બને છે. એની ઉપયોગિતાને અસર કરતું એક માત્ર પરિબળ એ તેનો નોંધણી ખર્ચ છે.

4G નેટવર્ક, વીડિયો અને ચલચિત્રને 3Gની સરખામણીએ ઘણી વધુ ઝડપે દર્શાવવા જેવી આધુનિક મોબાઇલ સેવાઓ પ્રદાન કરે છે. સૈદ્ધાંતિક રીતે, 4G સેવાઓ 100Mbps સુધીના ડેટા ડાઉનલોડ કરવાની અને 50Mbps જેટલો ડેટા અપલોડ કરવાની (મોબાઇલ કે કમ્પ્યુટર પરથી માહિતીને વેબ પર મુકવાની ક્રિયા) ઝડપ આપી શકે છે. તેનો સમાવેશ-વિસ્તાર વિસ્તૃત છે અને તેથી તે વત્તાઓછા પ્રમાણમાં સતત જોડાણ (Constant Connectivity) પ્રદાન કરે છે. 4G નેટવર્ક હજુ બાલ્યાવસ્થામાં છે, પરંતુ તેનો વિસ્તાર કદાચ ઝડપથી વિસ્તરે પણ ખરો. 3G અને 4G બંને નેટવર્કિંગ, ઝડપ અને માહિતીની આપ-લેની ગુણવત્તાના સંદર્ભમાં ઘણું આપી શકે તેમ છે.

બ્લ્યુ-ટુથ (Bluetooth)

મોબાઇલ ફોનના વધતા જતા ઉપયોગે, એક મોબાઇલથી અન્ય મોબાઇલ વચ્ચે ડેટાની આપ-લેની જરૂરિયાતને જન્મ આપ્યો. ડેટાની આપ-લે માટે કોઈ વ્યક્તિ SMSનો ઉપયોગ પણ કરી શકે, જોકે તેનો ખર્ચ થતો હોય છે. આ મર્યાદાનો અંત લાવવા બે મોબાઇલ વચ્ચે ટૂંકા અંતરેથી ડેટાની આપ-લે કરવા માટેની તારના જોડાણ વિનાની (વાયરલેસ) એવી બ્લ્યુ-ટુથ નામની આશીર્વાદરૂપ ટેકનોલોજી ઉપલબ્ધ બની. તે પ્રત્યાયન માટેના ISM (Industrial, Scientific and Medical) બેન્ડમાં શોર્ટ વેવલેન્થ રેડિઓ ટ્રાન્સમિશનનો ઉપયોગ કરે છે. આ ટેકનોલોજી આપણને ઉચ્ચ સ્તરની સુરક્ષાવાળી પર્સનલ એરિયા નેટવર્ક (Personal Area Networks PANs) રચવામાં મદદરૂપ થાય છે.



આકૃતિ 13.1 બ્લ્યુ-ટુથ લોગો

બ્લ્યુ-ટુથનો વિચાર શરૂઆતમાં એરિક્સન (Ericsson) દ્વારા કરવામાં આવ્યો અને એ પછી મોટે ભાગે બાકીની બધી કંપનીઓએ સ્વીકાર્યો. આજે, તે એક નાની સસ્તી રેડિયોચીપ બની ગઈ છે, જે કમ્પ્યુટર, પ્રિન્ટર, મોબાઇલ ફોન અને અન્ય ઈલેક્ટ્રોનિક સાધનોમાં ખોસીને લગાવી શકાય છે. બ્લ્યુ-ટુથ ચીપ, એક ખાસ ફ્રિક્વન્સી (frequency) પર ડેટાને મેળવનાર બ્લ્યુ-ટુથ તરફ પ્રસારિત કરે છે. એ પછી, મેળવનાર ચીપ, કમ્પ્યુટરને, મોબાઇલ ફોનને અથવા અન્ય કોઈ પણ સંલગ્ન સાધનને આ માહિતી આપે છે. તમે બધાએ બ્લ્યુ-ટુથ સાથેનો મોબાઇલ જોયો હશે, જેને મોબાઇલ ફોનના હેડસેટ સાથે જોડ બનાવી શકાય. આકૃતિ 13.1 બ્લ્યુ-ટુથનો લોગો દર્શાવે છે. આવો લોગો ધરાવતું કોઈ પણ સાધન એમ દર્શાવે છે કે તે બ્લ્યુ-ટુથ સક્ષમ છે.

સ્માર્ટફોન અને ટેબ્લેટ્સ (Smartphones and Tablets)

સેલ્યુલર મોબાઇલ ફોને તો જોડાણક્ષમતાની પરિભાષા જ બદલી નાંખી છે. સાદા ફોન આપણને વાતચીત કરવાની અને શાબ્દિક સંદેશાઓ મોકલવાની સવલત પૂરી પાડે છે. બીજો એક શબ્દ કે જે આજે બહુ જ સામાન્ય બની ગયો છે, એ છે ‘સ્માર્ટફોન’. પરંતુ આ સ્માર્ટફોન એ ખરેખર છે શું ? તમને ક્યારેય એવું આશ્ચર્ય થયું છે કે, આ સ્માર્ટફોન સાદા મોબાઇલ સેલફોન કરતાં કઈ રીતે અલગ છે? તે અચાનક કઈ રીતે ‘સ્માર્ટફોન’ બની ગયો છે ? સરસ, તો આ ‘સ્માર્ટફોન’ એ એક સાધન છે કે જે તમને ફોન કોલ કરવાની સુવિધા આપે છે અને તદ્દુપરાંત કેટલીક વધારાની સુવિધાઓ જેવી કે, પર્સનલ ડિજિટલ આસિસ્ટન્ટ (PDA), ઇ-મેઇલ મોકલવા અને મેળવવા, Officeના દસ્તાવેજને સુધારવા, મોબાઇલ એપ્સ (એપ્લિકેશન)નો ઉપયોગ કરવો, રમતો રમવી, રેડિયો ચાલુ કરી સાંભળવા, ચલચિત્રો જોવાં, ફોટા સુધારવા, GPS દ્વારા વાહન ચલાવવાની દિશા મેળવવી, તત્કાલ સંદેશા મોકલવા, Wi-Fi ઉપરાંત બ્લ્યુ-ટૂથની સવલત આપે છે અને પસંદગીના ડિજિટલ ગીતોને વગાડવા માટેની કમયાદી (play-list) બનાવવા વગેરે સવલત પણ પૂરી પાડે છે. મહેરબાની કરીને એ વાત ખાસ નોંધી લેજો કે અહીં રજૂ કરેલી સવલતોની યાદી અનંત છે અને તે બધી જ અહીં દર્શાવી શકાય તેમ નથી. વધુમાં, ‘સ્માર્ટફોન’માં ઓપરેટિંગ સિસ્ટમ ઉપલબ્ધ બની છે તેમજ મોબાઇલના પડદા (સ્ક્રીન) પર આંગળીથી સ્પર્શ (Touch) કરીને જરૂરી કાર્ય કરી શકાય તેવી સવલત (ટચસ્ક્રીન મિકેનિઝમ) પણ મળે છે. જો તમે તેની વિવિધ સવલતો અને લાક્ષણિકતાઓ જોશો, તો તેમાં તમને લગભગ કમ્પ્યૂટર જેવી જ લાક્ષણિકતાઓ અને સવલતો જોવા મળશે. એટલે જ એમ કહી શકાય કે, જ્યારે તમારી પાસે આ ‘સ્માર્ટફોન’ હોય, એટલે એમ સમજો કે તમારી પાસે એક પોકેટ કમ્પ્યૂટર જ છે. તેમાં માત્ર એક જ ખામી જણાય છે, અને તે છે પ્રમાણમાં નાનો સ્ક્રીન. આકૃતિ 13.2 માં આજકાલ બજારમાં ઉપલબ્ધ બનેલા સ્માર્ટફોન અને ટેબ્લેટ દર્શાવાયા છે.



આકૃતિ 13.2 : સ્માર્ટફોન અને ટેબ્લેટ

ટેબ્લેટ (લખવા-વાંચવાની પાટી) એ સુવાહ્ય (Portable) અને મોબાઇલ કમ્પ્યૂટિંગ સાધન છે. તેમાં મોબાઇલના પ્રમાણમાં મોટો સ્ક્રીન હોય છે અને લગભગ તેમાં આંતરપ્રસ્થાપિત ફોન સુવિધા પણ હોય છે. તેમાં ટાઇપિંગ કરવા માટે સામાન્ય રીતે સ્ક્રીન પર દૃશ્યમાન થતાં આભાસી કી-બોર્ડનો ઉપયોગ કરવામાં આવે છે. ઘણીવાર કી-બોર્ડને બદલે સ્ટાઇલસ પેન (stylus pen) અથવા ડિજિટલ પેન (digital pen) પણ ઉપયોગમાં લેવાતી હોય છે. તેમાં એ બધાં જ કાર્યો કરી શકાય છે, જે મોબાઇલ ફોન તેમ જ કમ્પ્યૂટર દ્વારા કરી શકાતાં હોય.

સ્માર્ટફોન અને ટેબ્લેટ સંબંધી ટેકનોલોજી સતત બદલાતી રહે છે. સ્ક્રીનનું કદ ઉપરાંત સંગ્રહક્ષમતા ખૂબ જ ઝડપથી બદલાય છે. આમ, આજે જેને આપણે સ્માર્ટફોન કહીએ છીએ, તે આવતા મહિને અથવા આવતા વર્ષે કે એ પછી કદાચ બદલાઈ પણ જાય અથવા પછી કદાચ સુપર સ્માર્ટફોન પણ બની જાય!

સેટેલાઈટ ફોન (Satellite Phones)

આપણે અગાઉ પણ એ વાતનો ઉલ્લેખ કરેલ છે કે, સેલ્યુલર મોબાઈલ ટેકનોલોજીએ સંપર્કની બાબતમાં ક્રાંતિ લાવી દીધી છે. પરંતુ બધી જ ટેકનોલોજી આવી જવા છતાં, હજુ પણ કોઈ વાર આપણને એક સમસ્યા નડે છે, એ છે, નેટવર્ક ઉપલબ્ધ ન હોવાની (*Network not available*) સમસ્યા. સેલ્યુલર મોબાઈલ ફોનની એક મર્યાદા એ પણ છે કે, તે સેલ્યુલર ટાવરની હદમાં હોવા જરૂરી છે. ઘણી જગ્યાએ આવાં ટાવર ઊભાં કરવાં ખૂબ જ મુશ્કેલ હોય છે. ઊંડે દરિયામાં તેલ શોધીને બહાર કાઢવાનું કામ કરતી સંસ્થાઓ (Oil Exploration Companies)ની કલ્પના કરો. આપણી પાસે દરિયાની વચ્ચે મોબાઈલનાં ટાવર ઊભાં કરવા માટેની કોઈ વ્યવસ્થા નથી. આવા કિસ્સાઓમાં શું થાય? આવા કિસ્સામાં કેવી રીતે આપણે સંપર્ક કરી શકીએ કે ડેટા મોકલી શકીએ? ઉપગ્રહ પ્રત્યાયન ટેકનોલોજી (સેટેલાઈટ કમ્યુનિકેશન ટેકનોલોજી) અને સેટેલાઈટ ફોન એ આવી સમસ્યાનું સમાધાન છે.

મોટે ભાગે સેટફોન (*Satphone*)ના નામે ઓળખાતો સેટેલાઈટ ફોન એ આમ તો એક જાતનો મોબાઈલ ફોન જ છે. તે ધરતી પર આવેલા સેલ્યુલર ટાવર સાથે સંપર્ક કરવાને બદલે પૃથ્વીની ભ્રમણકક્ષામાં ફરતા પ્રત્યાયન ઉપગ્રહ (કમ્યુનિકેશન સેટેલાઈટ) સાથે સંપર્ક કરે છે. ધરતી પરના સેલ્યુલર નેટવર્કની જેમ જ સેટફોન ધ્વનિસંદેશા, શાબ્દિક સંદેશા મોકલવાની સેવા અને નીચા બેન્ડવિડ્થ (Bandwidth) સાથેની ઈન્ટરનેટ ઉપયોગની સુવિધા પ્રદાન કરે છે.

સેટફોનની રચના અનુસાર તેના દ્વારા આવરી લેવાતો વિસ્તાર કદાચ આખી પૃથ્વી અથવા માત્ર અમુક ચોક્કસ વિસ્તાર પણ હોઈ શકે. સામાન્ય રીતે સેટફોનનો ઉપયોગ સરકાર દ્વારા નિયંત્રિત છે. ભારતમાં જો આપણે સેટેલાઈટ ફોનનો ઉપયોગ કરવો હોય, તો તે માટે ભારત સરકાર પાસેથી ખાસ પરવાનગી મેળવવી પડે. Inmarsat એ જૂનામાં જૂનો સેટેલાઈટ ફોન ઓપરેટર છે. તેણે તેની સેવાઓ વહાણો માટે પ્રદાન કરી હતી. Thuraya એ બીજો સેટેલાઈટ ફોન સેવા પ્રદાનકર્તા છે. તેના શક્તિશાળી ઉપગ્રહ દુનિયાની કુલ વસ્તીના બે તૃતીયાંશ લોકોને સેટેલાઈટ ફોનના ઉપયોગ દ્વારા પ્રત્યાયન કરવા સક્ષમ બનાવે છે. તે યુરોપ, આફ્રિકા, એશિયાના મધ્યપૂર્વ પ્રદેશો અને ઓસ્ટ્રેલિયામાં ઉપગ્રહ પ્રત્યાયનના ઉકેલ પૂરા પાડે છે. આકૃતિ 13.3 સેટેલાઈટ ફોનની એક ઝલક રજૂ કરે છે.

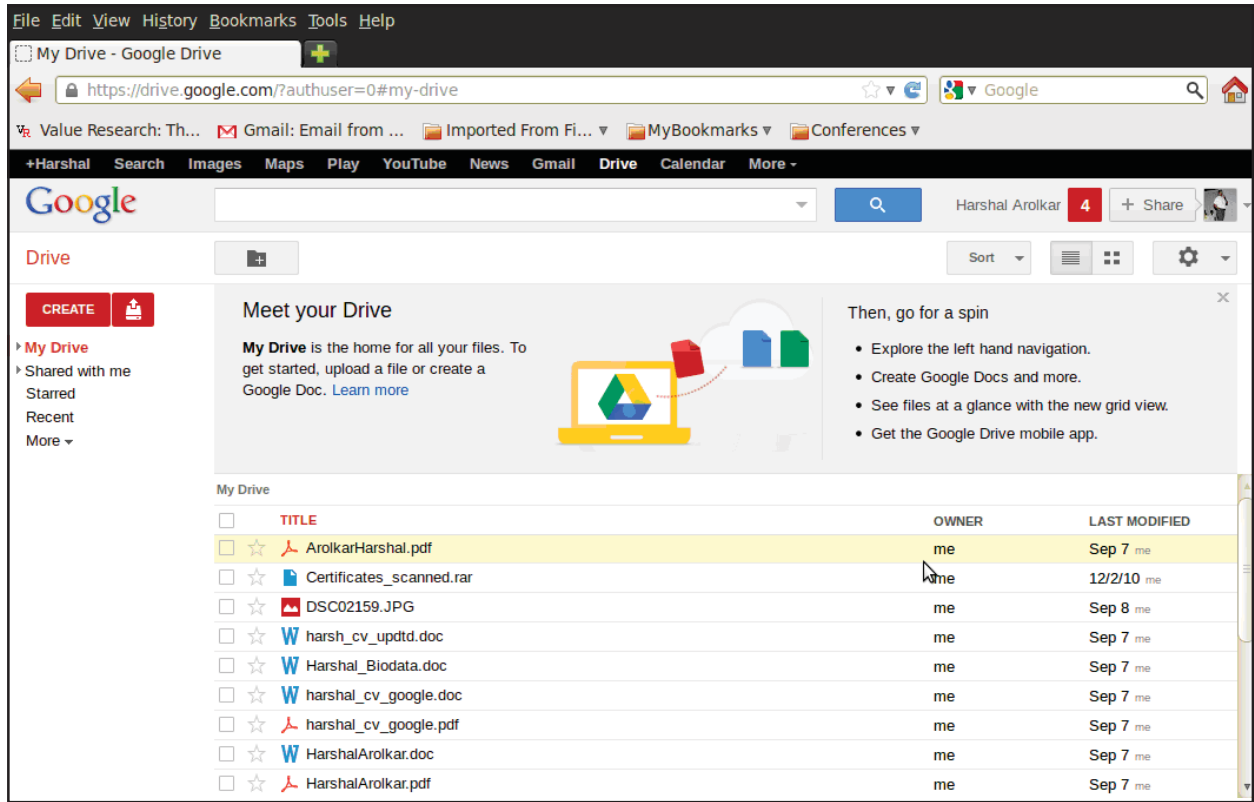


આકૃતિ 13.3 : સેટેલાઈટ ફોન

આકૃતિ 13.3માં જોઈ શકાય છે તે મુજબ, બજારમાં ઉપલબ્ધ સેટેલાઈટ ફોન એ મોટે ભાગે મોબાઈલ ફોન જેવા જ છે. જ્યાં ધરતી પરના સેલ્યુલર નેટવર્કની સેવાઓ ઉપલબ્ધ ન હોય તેવાં સ્થળો પર જ્યારે પ્રત્યાયનની જરૂર પડે, ત્યારે સેટફોન વધુ લોકપ્રિય છે. સેટફોનના ઉપયોગની કિંમત ઊંચી છે, પરંતુ તેમાં આપણે અવરોધરહિત જોડાણ મેળવી શકીએ છીએ.

ક્લાઉડ કમ્પ્યૂટિંગ (Cloud Computing)

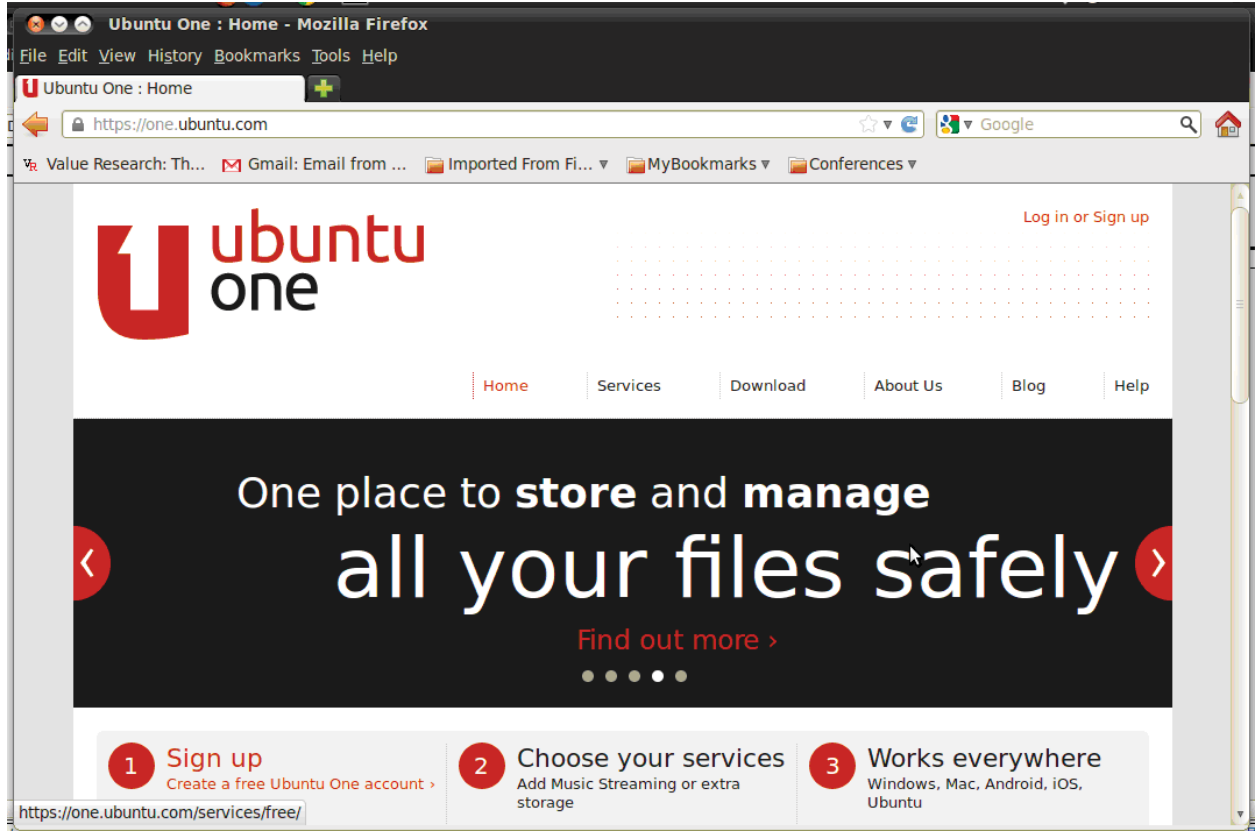
ઉપયોગકર્તાઓને આજકાલ ટેકનોલોજીની બીજી એક મોટી સવલત ઉપલબ્ધ બની છે તે છે, ક્લાઉડ કમ્પ્યૂટિંગ (Cloud computing) ક્લાઉડ કમ્પ્યૂટિંગ ઉપયોગકર્તાઓને તેમના ડેટા અને સંસાધનોને દુનિયાના કોઈ પણ ખૂણેથી ઉપયોગ કરી શકવાનું સામર્થ્ય બક્ષે છે. ક્લાઉડ કમ્પ્યૂટિંગ સવલત અંગત તેમજ કચેરી એમ બંને ઉપયોગ માટે ઉપલબ્ધ છે. પ્રથમ પ્રકારના ઉપયોગ અંગત ક્લાઉડ કમ્પ્યૂટિંગ (Personal Cloud Computing) તરીકે ઓળખાય છે. અંગત ક્લાઉડ કમ્પ્યૂટિંગનું એક લાક્ષણિક ઉદાહરણ છે, ગૂગલડ્રાઈવ (Google Drive). આપણને Google Drive અને Google + Photos વચ્ચે વહેંચાતી 5 GBની જગ્યા (drive space) બિલકુલ મફત આપે છે, જેથી કરીને તમે આ જગ્યામાં તમારી ફાઈલો, ઈ-મેઈલ અને ફોટા રાખીને ગમે તે સાધન વડે, ગમે ત્યાંથી તેનો ઉપયોગ કરી શકો. આ વિકલ્પ પેનડ્રાઈવ લઈ જવા કરતાં વધુ સારો છે. આકૃતિ 13.4 (Google Drive)નું એક લાક્ષણિક ગ્રાફિકલ યુઝર ઈન્ટરફેસ (GUI) દર્શાવે છે.



આકૃતિ 13.4 : Google Driveનું GUI

પર્સનલ ક્લાઉડ કમ્પ્યૂટિંગના અનેક ફાયદાઓ પૈકીનો એક મહત્વનો ફાયદો એ આપમેળે સુમેળ (Automatic Synchronization) છે. આ સુવિધા ત્યારે હાથવગી બને છે, જ્યારે ઉપયોગકર્તા અનેક જગ્યા પર નકલો રાખી મૂકવા ઈચ્છતો હોય. ઉદાહરણ તરીકે, ધારો કે ઉપયોગકર્તા વેબ ઉપર કોઈ વીડિયો શોધીને આ વીડિયોને મોબાઇલ સાધન દ્વારા ખરીદી લે, તો આ વીડિયો ફાઇલ માત્ર તેમના મોબાઇલ સાધન પર જ ડાઉનલોડ કરવાને બદલે આપમેળે ક્લાઉડ કમ્પ્યૂટિંગ સાથે જોડાયેલાં બધાં જ સાધનો પર ડાઉનલોડ કરવામાં આવશે. ક્લાઉડ ખાતાને અનેક સાધનો જેવાં કે, ઉપયોગકર્તાનું ઘરનું ડેસ્કટોપ કમ્પ્યૂટર, ઓફિસનું લેપટોપ કમ્પ્યૂટર, મોબાઇલ ફોન વગેરે સાથે સુમેળમાં રાખવું (sync રાખવું) શક્ય છે.

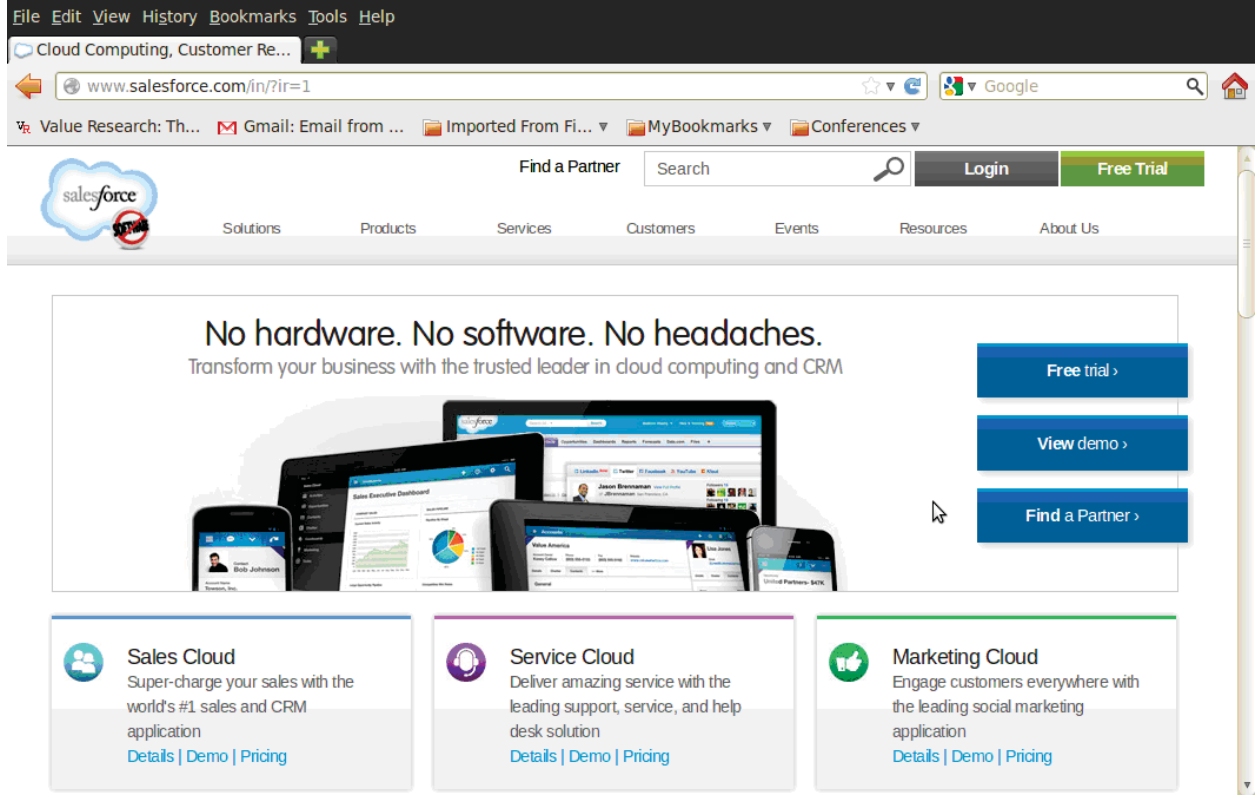
Ubuntu One એ એક અંગત ક્લાઉડમંચ (Personal Cloud Platform) છે, જે આપણા તમામ ડિજિટલ ડેટાને ભેગા રાખવાની સવલત પૂરી પાડે છે. આપણા આ સમાવિષ્ટ સાહિત્યને આપણે જ્યાં પણ હોઈએ, ત્યાંથી આપણી રીતે ઉપયોગમાં લઈ શકીએ છીએ. સંગ્રહ સૂમેળ (Sync), ડેટા-વહેંચણી (Sharing) અને ડેટાને સ્ટ્રિમિંગ (Streaming) કરવાની સવલત ધરાવતી ક્લાઉડ સેવાઓ માટેના Ubuntu One સ્યુટ દ્વારા આપણે આપણા સંગ્રહેલ સંગીત, ગમતા ફોટો, વીડિયો, અગત્યના દસ્તાવેજો અને બીજી ઘણી બધી માહિતીનો ગમે તે સાધન મારફત અને ગમે તે સમયે તરત જ ઉપયોગ કરી શકીએ છીએ. વિનામૂલ્યે ઉપલબ્ધ ઓપરેટિંગ સિસ્ટમ, Ubuntu માટે સ્થિર આવક (Sustainable Revenue) પૂરી પાડવા મે-2009માં આ સુવિધા શરૂ કરવામાં આવી હતી, જેને કેનોનિકલ (Canonical) દ્વારા આર્થિક પીઠબળ પૂરું પાડવામાં આવ્યું હતું. Ubuntu One પર તમારું ખાતું ખોલવા માટે, <https://one.ubuntu.com/> ની મુલાકાત લો. આકૃતિ 13.5 Ubuntu Oneનું હોમપેજ દર્શાવે છે.



આકૃતિ 13.5 : Ubuntu Oneનું હોમપેજ

ક્લાઉડ કમ્પ્યુટિંગની રચનાને લીધે ધંધા-ઉદ્યોગોને પણ ઘણો મોટો લાભ થયો છે. ધંધા-ઉદ્યોગો માટે ક્લાઉડ કમ્પ્યુટિંગની રૂપરેખા (scenario), 'સોફ્ટવેર એઝ એ સર્વિસ' (Software as a Service-SaaS), ઈન્ફ્રાસ્ટ્રક્ચર એઝ એ સર્વિસ' (Infrastructure as a Service-IaaS) અને 'પ્લેટફોર્મ એઝ એ સર્વિસ' (PaaS) નામનાં જુદાં-જુદાં મોડેલમાં ઉપલબ્ધ છે. કોઈ પણ વ્યક્તિ પોતાના ધંધાકીય એકમની જરૂરિયાતને આધારે આમાંના કોઈ પણ મોડેલ પસંદ કરી શકે છે. તો ચાલો, આ દરેક મોડેલના મૂળભૂત ખ્યાલ વિશે ટૂંકમાં જોઈએ.

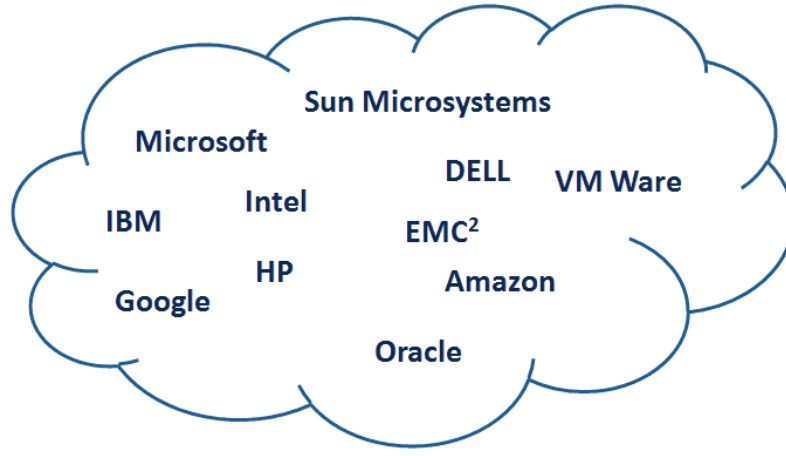
સોફ્ટવેર એઝ એ સર્વિસ (Software as a Service - SaaS) : ધંધાકીય એકમો પોતાના ઉપયોગ માટેનાં સોફ્ટવેર તૈયાર કરવાને બદલે સામાન્ય રીતે ક્લાઉડ સેવાઓ પ્રદાનકર્તાને કિંમત ચૂકવીને આવાં સોફ્ટવેર વાપરતાં હોય છે. સેવા-પ્રદાનકર્તા, આવા સોફ્ટવેરને પોતાના વેબ-સર્વર પર મૂકે છે, જેથી કરીને સેવા લેનાર કંપનીએ તેને પોતાના કમ્પ્યુટર પર પ્રસ્થાપિત (Install) કરવાની જરૂર રહેતી નથી. સેવા પ્રદાનકર્તા, સેવા લેનાર કંપની વતી સોફ્ટવેરનું વ્યવસ્થાપન પણ કરશે. આ મોડેલ ધંધાકીય એકમોને સોફ્ટવેર ખરીદવાની, પ્રસ્થાપિત કરવાની અને સોફ્ટવેર કે હાર્ડવેરની જાળવણી કરવાની જંઝટથી મુક્ત કરે છે. આ બધા માટે તેમની પાસે માત્ર આ સોફ્ટવેર સાથે જોડાણ સાધી તેનો ઉપયોગ કરી શકે તે માટે ઈન્ટરનેટ જોડાણની જ જરૂર પડે છે. SaaS (સાસ)ના ઉદાહરણ તરીકે 'કસ્ટમર રિલેશનશિપ મેનેજમેન્ટ'ની સેવાઓનો સમાવેશ થાય છે. 'કસ્ટમર રિલેશનશિપ મેનેજમેન્ટ' (CRM) સોફ્ટવેર માટે Salesforce.com એ સૌથી વધુ ઉપયોગમાં લેવાતી ક્લાઉડ (Cloud) સેવા છે. આકૃતિ 13.6 <http://www.salesforce.com/in/>નું હોમપેજ દર્શાવે છે.



આકૃતિ 13.6 : <http://www.salesforce.com/in/?ir=1> હોમપેજ

ઇન્ફ્રાસ્ટ્રક્ચર એઝ એ સર્વિસ (Infrastructure as a Service IaaS) : પોતાનાં ભૌતિક સંસાધનો વસાવવાને બદલે કોઈ ધંધાકીય એકમ ક્લાઉડ સેવા પ્રદાનકર્તા પાસેથી હાર્ડવેર-ઇન્ફ્રાસ્ટ્રક્ચરનો ભાડાની સેવા રૂપે ઉપયોગ કરી શકે છે. ક્લાઉડ સેવા રૂપે હાર્ડવેરને જુદી-જુદી ક્રિયાઓ જેવી કે, ગણતરીઓ, સંગ્રહ, છાપકામ, બેક-અપ અને નેટવર્કના ભાગ માટે ઉપયોગમાં લઈ શકાય છે. સેવા પ્રદાનકર્તા તમામ સાધનોની માલિકી ધરાવે છે અને તેને રાખવા, ચલાવવા અને જાળવણી માટે જવાબદાર છે. IaaS પ્રદાનકર્તાઓમાંનાં કેટલાંક નામો તરીકે Windows Azure Virtual Machines, Google Compute Engine અને HP Cloud ગણાવી શકાય.

પ્લેટફોર્મ એઝ એ સર્વિસ (Platform as a Service - PaaS) : અહીં ધંધાકીય એકમ તરીકે સામાન્ય રીતે, કોઈ સોફ્ટવેર કંપની હોય છે, જે ક્લાઉડ સેવા પ્રદાનકર્તા પાસેથી સોફ્ટવેર ટૂલ્સ અને/અથવા લાઇબ્રેરીનો ઉપયોગ કરી સોફ્ટવેર તૈયાર કરે છે. તે સોફ્ટવેરની સ્થાપના/વિસ્થાપના (deployment) અને કન્ફિગ્યુરેશન સેટિંગને પણ નિયંત્રિત કરે છે. સેવા પ્રદાનકર્તા સોફ્ટવેર તૈયાર કરવા માટે જરૂરી એવી નેટવર્ક, સર્વર, સંગ્રહ (સ્ટોરેજ) અને અન્ય તમામ સેવાઓ પૂરી પાડે છે. આ મોડેલ, સોફ્ટવેર તૈયાર કરવા ઉપયોગી એવા હાર્ડવેર અને સોફ્ટવેરને ખરીદવાના અને જાળવણીના ખર્ચ અને ઝંઝટ વગર સોફ્ટવેરની સ્થાપના/વિસ્થાપના કરવાની (deployment of software) સવલત પૂરી પાડે છે. ઘણી બધી કંપનીઓએ હવે ક્લાઉડ કમ્પ્યુટિંગના ક્ષેત્રે ઝંપલાવ્યું છે. આકૃતિ 13.7 આવી કેટલીક કંપનીઓની એક ઝલક પૂરી પાડે છે.



આકૃતિ 13.7 : ક્લાઉડ સેવાઓ પ્રદાન કરતી કંપનીઓ

સામાન્ય કમ્પ્યુટરીંગ સુવિધા કરતાં ક્લાઉડ કમ્પ્યૂટિંગ (scenario)ને તેની કેટલીક ચાવીરૂપ લાક્ષણિકતાઓને આધારે જુદી પાડી શકાય. જેવી કે, તે મોટે ભાગે ઓફ-સાઇટ (off site) હોય છે, ઇન્ટરનેટ દ્વારા તેનો ઉપયોગ કરી શકાય છે, સેવાઓનો ઉપયોગ કરવા ખૂબ જ મર્યાદિત IT જ્ઞાનની જરૂર પડે છે, ડેસ્કટોપથી મોબાઇલ સુધીના તમામ ઉપકરણ મારફત તે ઉપલબ્ધ બની શકે છે, વગેરે. સ્થાપના અંગેનાં ખર્ચની બચત થતી હોવાને લીધે આજે વપરાશકર્તા માટે તેમજ ધંધાકીય એકમો માટે ક્લાઉડ કમ્પ્યૂટિંગની સવલત અત્યંત ઉપયોગી છે.

માનવ/વસ્તુની પરખ માટે ઊભરતા પ્રવાહો (Emerging trends in Object/Human Recognition)

આજે આપણે, સર્વવ્યાપી (Ubiquitous) કમ્પ્યૂટિંગની દુનિયામાં જીવી રહ્યા છીએ, જેમાં સમગ્ર ભૌતિક વાતાવરણમાં આપણે કમ્પ્યુટર અને પરખ કરતાં (sensing) સાધનોથી ઘેરાયેલા છીએ. અસ્તિત્વમાં હોવા છતાં આ કમ્પ્યુટરો સર્વવ્યાપી ઉપયોગકર્તા માટે અદૃશ્ય છે. ક્લાઉડ કમ્પ્યૂટિંગ પ્લેટફોર્મની ઉપલબ્ધતાને કારણે આવી સર્વવ્યાપકતા (ubiquity), શક્ય બનાવી શકાઈ છે. (ubiquitous) કમ્પ્યૂટિંગ ક્ષેત્રની સમસ્યાઓ પૈકીની એક મોટી સમસ્યા, કોઈ ચોક્કસ કમ્પ્યૂટિંગ અથવા સેન્સર (sensor) જેવી વસ્તુઓની પરખ અંગેની છે. આ વિભાગમાં આપણે કોઈ પણ વસ્તુ (object)ને ઓળખવા માટે ઉપયોગમાં લેવાતી કેટલીક ટેક્નિક જેવી કે, રેડિયો-ફ્રિક્વન્સી પરખ (Radiofrequency Identification - RFID) બાયોમેટ્રિક્સ (Biometrics), વૈશ્વિક સ્થાનનિર્ધારણ પદ્ધતિ (Global Positioning System - GPS) અને અવાજ-પરખ (Speech Recognition) વિશે શીખીશું.

રેડિયો ફ્રિક્વન્સી પરખ (Radio Frequency Identification - RFID)

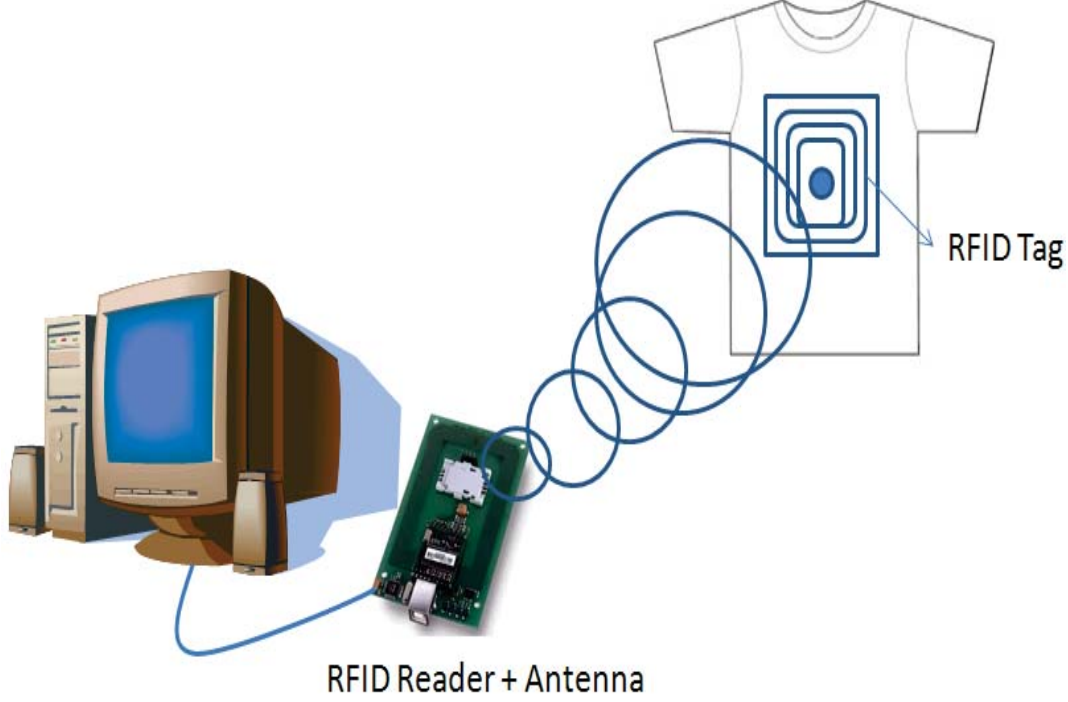
કોઈક વસ્તુની ભાળ મેળવવા (tracking) અને પરખ કરવા બજારમાં સૌથી વધુ ઉપયોગમાં લેવાતી ટેક્નોલોજી છે. રેડિયો ફ્રિક્વન્સી આઇડેન્ટિફિકેશન સિસ્ટમ તે એન્ટેના (antenna), ટ્રાન્સિવર (transceiver) અને ટ્રાન્સપોન્ડર (transponder) એમ ત્રણ ભાગોની બનેલી છે.

એન્ટેના પ્રમાણમાં ટૂંકા અંતર (range)માં અંદર રેડિયો-ફ્રિક્વન્સી સંકેતો ઉત્પન્ન કરવાના ઉપયોગમાં આવે છે. આ અંતર 10 સેન્ટિમીટરથી 200 મીટરની વચ્ચે બદલાઈ શકે છે. એન્ટેના, હાથમાં રાખવાનું અથવા લગાડવા માટેનું હોઈ શકે છે, અને કમ્પ્યુટરથી દીવાલ સુધી કોઈ પણ સપાટી પર તે લગાવી શકાય છે.

ટ્રાન્સિવર રેડિયો-ફ્રિક્વન્સી સંકેતો વાંચે છે અને પ્રક્રિયા કરતા સાધન, સામાન્ય રીતે 'રેડિયો-ફ્રિક્વન્સી રીડર' સુધી તેની માહિતી પહોંચાડે છે. તે ટ્રાન્સપોન્ડરમાં સંગ્રહાયેલ ડેટાનું અર્થઘટન કરવા 'ડીકોડર' (Decoder) તરીકે કાર્ય કરે છે.

ટ્રાન્સપોન્ડર એ RFID ટેગ (Tag) છે, જે ઓબ્જેક્ટ સાથે જોડાયેલ હોય છે. ટેગ (Tag) એક તૈયાર પ્રોગ્રામ સંગ્રહેલી માઈક્રોચીપ (Programmed microchip) છે, અને તે આશરે 2000 બાઈટ્સ કે તેથી ઓછા બાઈટ્સ જેટલી માહિતી સંગ્રહી શકવા સક્ષમ છે. વધુમાં, આ ટેગ સક્રિય (Active) કે નિષ્ક્રિય (Passive) હોઈ શકે છે. અક્રિયાશીલ ટેગને કાર્ય કરવા માટે બેટરીની જરૂર પડતી નથી, જ્યારે ક્રિયાશીલ ટેગ તેમની પોતાની બેટરી સાથે જ આવે છે.

RFID ટેગમાંથી ડેટાનું પ્રસારણ ત્યારે જ થાય છે, જ્યારે ટેગ એન્ટેનાના ક્ષેત્રમાંથી પસાર થાય છે. ટેગ એ પછી એન્ટેનામાંથી સક્રિય થયાના સંકેત (એક્ટિવેશન સિગ્નલ) ચકાસે છે, આમ થવાથી RFID ટેગ સક્રિય થાય છે, અને સ્કેનિંગ એન્ટેના દ્વારા ઝીલવા માટે તેની માઈક્રોચિપ પર રહેલી માહિતી પ્રસારિત કરે છે, RFID સિસ્ટમ ઘણા વિનિયોગમાં ઉપયોગમાં લેવાય છે, જેવાં કે just-in-time ઉત્પાદનના ટ્રેકિંગ ભાગોમાં, શોપિંગમોલમાં ગ્રાહકો દ્વારા ખરીદી પર ધ્યાન રાખવા માટે વગેરે. આકૃતિ 13.8 RFID પદ્ધતિનો અમલ કરવાના ખ્યાલને દર્શાવે છે.



આકૃતિ 13.8 : RFID પદ્ધતિની કાર્યશૈલી

ઘણા બધા ઉદ્યોગો દ્વારા જુદા-જુદા હેતુઓ માટે RFID ટેકનોલોજીનો ઉપયોગ કરવામાં આવી રહ્યો છે. 2008ની ‘બેઈજિંગ સમર ઓલિમ્પિક ગેમ્સ’માં ટિકિટની વ્યવસ્થા (Hightech Ticketing) અને ખાણી-પીણીની ચીજવસ્તુઓ શોધી કાઢવા (Foodtracking) માટે RFID ટેકનોલોજીનો ઉપયોગ કરવામાં આવ્યો હતો. બેઈજિંગ ગેમ્સ દરમિયાન RFID સંલગ્ન ટિકિટો રાખવાનો મુખ્ય ઉદ્દેશ ટિકિટોના ધસારાને પહોંચી વળવા માટેનો હતો, જેથી મુલાકાતીઓને કાર્યક્રમના સ્થળે દરવાજામાંથી ઝડપથી પ્રવેશ આપી શકાય અને ઓલિમ્પિકના મેદાનમાં ચોક્કસ વિસ્તારોમાં બિનઅધિકૃત પ્રવેશ સામે સુરક્ષા આપી શકાય.

તેણે સુરક્ષા કર્મચારીઓને હોટેલો, સ્થળો, ઉત્પાદકો, વિતરણ-કેન્દ્રો અને હોસ્પિટલો પર નજર રાખવા માટે અને ઓલિમ્પિકમાં ‘ફૂડ સેફ્ટી ટ્રેકિંગ સિસ્ટમ’ (Food safety tracking system)ને પીકબળ પુરું પાડ્યું હતું. ‘ફૂડ-સેફ્ટી ટ્રેકિંગ સિસ્ટમ’માં ઉપયોગમાં લેવાયેલ RFID સિસ્ટમ દ્વારા ખાણી-પીણીની ચીજવસ્તુઓનું ઉત્પાદન, પ્રક્રિયા અને વિવિધ સ્થળે ખોરાકનાં પડીકાં (ફૂડપેકેટ)ના પરિવહનનું નિયમન કરવામાં આવ્યું હતું.

પ્રાણીઓની ગતિવિધિ પર દેખરેખ (Animal-Tracking) એ RFIDના અનેક વિનિયોગો પૈકીનો શરૂઆતનો વિનિયોગ છે. પશુધન એ તેના માલિકો માટે મિલકત સમાન છે. ચરવા માટે તે સતત ફરતાં રહેતાં હોવાથી મોટા સમૂહ (large herds)ને સંભાળવામાં ઘણી મુશ્કેલી પડે છે. આવા સતત રખડતાં રહેતાં પશુઓનાં ટોળાંની ભાળ મેળવવા અને ધ્યાન રાખવા RFID ટેકનોલોજી ઉપયોગમાં લેવાય છે. પહેલાનાં વખતમાં જેમ પશુની અલગ ઓળખ માટે ગળામાં ઘંટડી લટકાવવામાં આવતી હતી, તેવી જ રીતે RFID ટેગ ઢોરના ગળામાં લટકાવવામાં આવે, કાનમાં પહેરાવવામાં આવે અથવા કેટલીક વાર તો આવા પશુના શરીરમાં મૂકવામાં આવે છે, કારણકે રેડિયો-સંકેતો શરીરને અસર પહોંચાડતા નથી. એક વાર લગાવ્યા પછી, RFID ટેગ દરેકેદરેક પ્રાણીની ભાળ રાખવાનું અને નોંધવાનું કાર્ય કરે છે.

આ વિશિષ્ટ ટેગનો ઉપયોગ કરીને આપણે પશુઓનો ડેટાબેઝ પણ તૈયાર કરી શકીએ છીએ, જે દરેક પશુનો જન્મ, ઉછેર અને પ્રજાતિની નોંધ રાખે. RFID નો ઉપયોગ આપણને પ્રાણીઓની તંદુરસ્તીનું ધ્યાન રાખવામાં પણ મદદરૂપ નીવડે છે, વળી, દરેક પશુની દાક્તરી સારવાર અને રસી મૂક્યાની નોંધ રાખવાની પણ સુવિધા આપે છે અને દરેક પશુનું ધ્યાન રાખીને ચેપી ઢોરને રોગની શરૂઆતમાં જ અલગ તારવીને પશુઓમાં રોગચાળાનો ફેલાવો થતો અટકાવી શકીએ છીએ. RFID ટેકનોલોજીના ઉપયોગથી મળતા અનેક લાભને કારણે હવે આધુનિક કૃષિવ્યવસ્થાપનનો તે એક નિયમિત હિસ્સો બની ગયેલ છે.

RFID ટેકનોલોજી આજકાલ હવે ગ્રંથાલયોમાં પણ ઉપયોગમાં લેવાઈ રહી છે. પુસ્તકોની હેરફેરની ભાળ મેળવવા તેમજ પુસ્તકોની ચોરીને પકડી પાડવા તેનો વિશેષ ઉપયોગ થાય છે. RFID આધારિત ગ્રંથાલય-વ્યવસ્થાપન પદ્ધતિમાં ચીલાચાલુ બારકોડને બદલે ‘સ્માર્ટ RFID લેબલ’, લેબલની માહિતી વાંચવા માટેનું RFID રીડર’ (નિયત સ્થાને ગોઠવેલું કે છૂટું હાથમાં રાખી શકાય તેવું (Handheld) અને ગ્રંથાલય-વ્યવસ્થાપન સોફ્ટવેરનો સમાવેશ થાય છે. આવી પદ્ધતિનો ઉપયોગ ગ્રંથાલયની રોજબરોજની કામગીરીનું સંચાલન વધુ અસરકારક રીતે કરવામાં મદદરૂપ નીવડે છે.

આ ટેકનોલોજીનો ઉપયોગ કરવા માટે કાગળ જેવા પાતળા ‘સ્માર્ટ લેબલ’ (Smart label)ની જરૂર પડે, તેનું કદ જુદી-જુદી લાઈબ્રેરીની જરૂરિયાત અનુસાર જુદું-જુદું હોઈ શકે. પરંતુ મોટા ભાગના કિસ્સામાં 2” X 2” નું લેબલ પુરતું ગણાય. આ લેબલ દરેક પુસ્તકના મુખ્ય પાના (પૂંકા)ની અંદરની બાજુ એ રીતે લગાડવામાં આવે છે કે જેથી પુસ્તક ઉઘાડતી કે બંધ કરતી વખતે તે અવરોધરૂપ ન બને. લેબલને એક નાનું એન્ટેના અને એક નાનકડી ચીપ હોય છે. જેમાં પુસ્તક અંગેની માહિતી સંગ્રહવામાં આવે છે, દરેક પુસ્તકને ઓળખવા માટે આ માહિતીમાં એક અજોડ સ્થિતિ સ્થાનપ્રાપ્તિ ક્રમ પણ હોય છે. લેબલમાં માઈક્રોચીપ ઉપર સમાવિષ્ટ માહિતી, RFID સ્કેનર/રીડરની મદદથી લાઈબ્રેરીમાં કોઈ પણ જગ્યાએથી વાંચી શકાય છે. જો આપણે આ લેબલમાં પડેલી માહિતીને ગ્રંથાલય માહિતી પદ્ધતિમાં તબદીલ કરવી હોય તો, વધારામાં એક તાર વિનાની સંચાર પદ્ધતિ (Wireless communication system)ની જરૂર પડે.

ટેકનોલોજીનો ઉપયોગ કરવાથી વાંચકોને પુસ્તકો આપવાં અને પાછાં જમા લેવાની કામગીરી ઝડપી અને સરળ બને છે. બારકોડ કરતાં RFID ટેગમાંથી ઝડપથી માહિતી વાંચી શકાતી હોવાથી, આ ટેકનોલોજીના ઉપયોગ દ્વારા ‘સર્ક્યુલેશન ઓપરેશન’ (Circulation Operations) પાર પાડવા માટે લાગતો સમય ઘટાડી શકાય છે. એક પણ પુસ્તકને કાઢ્યા વિના એકસાથે એક જ રેકમાં પડેલાં અનેક પુસ્તકોની વિગતો પણ વાંચી શકાય છે. આમ, તમામ અજોડ ઓળખને વાંચવા માટે હાથમાં પકડવાના (હેન્ડલ હેલ્ડ) સ્કેનરને પુસ્તકોના કબાટ પરથી ઝડપથી ફેરવી શકાય. આ વળી, યોગ્ય રીતે જરૂરી જથ્થાની જાળવણી કરવાની અને કોઈ વસ્તુ પૂરી નથી થઈ, તે ચકાસવાની સવલત આપે છે.

આપણામાંથી મોટા ભાગના કોઈક ને કોઈક સમયે કોઈ સુપરમાર્કેટ (ખરીદી માટેનાં મોટા ભવનો)માં તો ગયા જ હોઈશું! ઘણા લોકો આવા સ્થળની મુલાકાત લેતા હોય છે અથવા ચીજવસ્તુઓની ખરીદી કરવા માટે જતા હોય છે. કદાચ તમે એ અવલોકન કર્યું હશે કે, આવા સુપરમાર્કેટના બહાર નીકળવાના દરવાજે. તમે જોરદાર ‘બીપ...’ એવો અવાજ સાંભળ્યો હશે. ચીજવસ્તુની ચોરી અટકાવવા માટે ઉપયોગમાં લેવાતી આ એક સલામતી વ્યવસ્થા છે. દરેક ચીજવસ્તુ પર RFID ટેગ લગાવાયેલી હોય છે, જે બિલ બનાવનાર વ્યક્તિ દ્વારા કાઢી નાંખવામાં આવે છે. કદાચ કોઈ વ્યક્તિ જાણીજોઈને બિલ બનાવ્યા વગર જ ચીજવસ્તુ ઉઠાવી જવાનો પ્રયત્ન કરતી હોય, તો તેવા સંજોગોમાં આ એલાર્મ વાગે છે.

ઘણીવાર સુપરમાર્કેટમાં ચીજવસ્તુ ખરીદવાની પ્રક્રિયા બહુ સમય બગાડનારી હોય છે, જેમકે, સુપરમાર્કેટમાં પ્રવેશતાં જ આવતી પરસાળમાં લાગતો સમય, જોઈતી વસ્તુ ક્યાં ઉપલબ્ધ છે, તે શોધવા માટે લાગતો સમય અને છેલ્લે ચુકવણી કરી તેમજ બહાર નીકળવા તપાસ માટે લાગતો સમય. ગ્રાહકને વસ્તુ ખરીદવામાં લાગતો સમય ઘટાડવા માટે પણ હવે RFID ટેકનોલોજીનો ઉપયોગ થાય છે. મોટા સુપરમાર્કેટમાં ફરીને ખરીદી માટે ચીજવસ્તુ ભેગી કરવા એવી હાથલારી (શોપિંગકાર્ટ કે ટ્રોલી) હોય છે, જેની સાથે એક આંગળીના સ્પર્શ દ્વારા કાર્યરત થતું ‘ટચસ્ક્રીન’ મોનિટર, ગ્રાહકનું કાર્ડ વાંચવા માટેનું ‘કાર્ડરીડર’, સ્થાન-નિર્દેશ માટે ‘ટ્રાન્સમીટર’ અને ‘પ્રોડક્ટ રીડર’ હોય છે.

ગ્રાહક જ્યારે આવા સુપરમાર્કેટમાં આવે છે ત્યારે, તે પહેલાં આવું એક શોપિંગકાર્ટ લઈને તેના 'ક્લાયન્ટકાર્ડ' દ્વારા સુપરમાર્કેટની સિસ્ટમમાં લોગ-ઇન થઈ જાય છે. ગ્રાહક માટે સ્વાગત કર્યા બાદ સ્ક્રીન પર ગ્રાહકે છેલ્લે ખરીદી કરેલ ચીજવસ્તુઓની યાદી દર્શાવાય છે. ગ્રાહક આ જૂની યાદીમાં જરૂરી નવી ચીજવસ્તુઓ ઉમેરી શકે તેમજ ન જોઈતી હોય તે ચીજવસ્તુ યાદીમાંથી કાઢી નાંખી શકે. એ પછી ખરીદી માટે દિશાનિર્દેશ કરતી શોપિંગકાર્ટની નેવિગેશન સિસ્ટમની મદદથી યાદીમાં સામેલ વસ્તુઓ કઈ બાજુ, કઈ જગ્યાએ, કઈ છાજલી પર મળી શકશે, તે અંગે ગ્રાહકને માર્ગદર્શન મળશે. ગ્રાહક શોપિંગકાર્ટમાં જેવી કોઈ ચીજવસ્તુ મૂકશે કે તરત જ સિસ્ટમ ગ્રાહકે કેટલા રૂપિયા ચૂકવવા પડશે, તે ગણીને સ્ક્રીન પર દર્શાવી દેશે. બધી જ ખરીદી થઈ ગયા પછી ગ્રાહક બહાર નીકળવા માટે દરવાજા તરફ જશે. બિલની ગણતરી કરવાની બધી વિધિ પૂરી થઈ ગઈ હોવાથી હવે તેણે માત્ર ચુકવણી જ કરવાની રહે અને ચુકવણી કરીને સુપરમાર્કેટમાંથી બહાર નીકળી જશે. આજકાલ RFIDના ઘણા બધા ઉપયોગ શરૂ થઈ ગયા છે, પરંતુ તે બધાની ચર્ચા આ પુસ્તકની મર્યાદા બહાર હોવાથી તેની ચર્ચા અહીં કરેલ નથી.

બાયોમેટ્રિક્સ (Biometrics)

ટેક્નોલોજીના વધતા જતા ઉપયોગને કારણે માહિતીના ઉપયોગ પર નિયંત્રણો મૂકવાં જરૂરી બન્યાં છે. આ માટે ઘણી બધી પદ્ધતિઓ જેવી કે, પાસવર્ડ દ્વારા સુરક્ષા (Password Protection) માહિતીના ઉપયોગ માટેના અધિકાર (Access rights) સંકેતીકરણ (એનક્રિપ્શન) વગેરે, ઉપયોગમાં લેવાઈ રહી છે. જોકે તેમ છતાં, માહિતીના ઉપયોગ પર વધુ સારી રીતે નિયંત્રણ મૂકવા માટે હવે ઘણા વિનિયોગમાં બાયોમેટ્રિક્સ તરીકે ઓળખાતી એક નવી ટેક્નિકનો સઘન રીતે ઉપયોગ કરવામાં આવી રહ્યો છે. બાયોમેટ્રિક્સ (અથવા બાયોમેટ્રિક પ્રમાણિતતા) સામાન્ય રીતે માનવીઓની તેમની લાક્ષણિકતાઓ અને વર્તન પર આધારિત ઓળખ (identification of humans) છે. માનવીની આવી શારીરિક લાક્ષણિકતાઓમાં આંગળીઓની રેખાની છાપ, આંખની કીકી અને નેત્રપટલ, અવાજનો પ્રકાર, ચહેરાની રચના, હસ્તાક્ષર અને હાથનું માપ વગેરેનો સમાવેશ થાય છે. બાયોમેટ્રિક્સ પદ્ધતિ સામાન્ય રીતે પાંચ ભાગની બનેલી હોય છે.

1. એક સેન્સર, જેનો ઉપયોગ માહિતી એકઠી કરવા માટે થાય છે.
2. સંકેતો પર પ્રક્રિયા કરવા માટે અલ્ગોરિધમ (Signal Processing Algorithms), જે ગુણવત્તા નિયંત્રણનું કાર્ય કરે છે અને 'બાયોમેટ્રિક્સ પેટર્ન' (Biometrics Pattern) તૈયાર કરે છે.
3. ડેટા-સંગ્રાહક (Data storage) જે નવી 'બાયોમેટ્રિક્સ પેટર્ન' સાથે સરખાવવા જરૂરી માહિતી સંગ્રહે છે.
4. સરખામણી કરવા માટેનું અલ્ગોરિધમ (Matching Algorithm) કે જે ખરેખર સરખામણી કરે છે,
5. નિર્ણય-પ્રક્રિયા (સ્વયંચાલિત હોય અથવા માનવીય મદદથી થતી હોય) કે જે નિર્ણય કરવા માટે 'મેચિંગ અલ્ગોરિધમ' (Matching Algorithm) પાસેથી પરિણામ મેળવે છે.

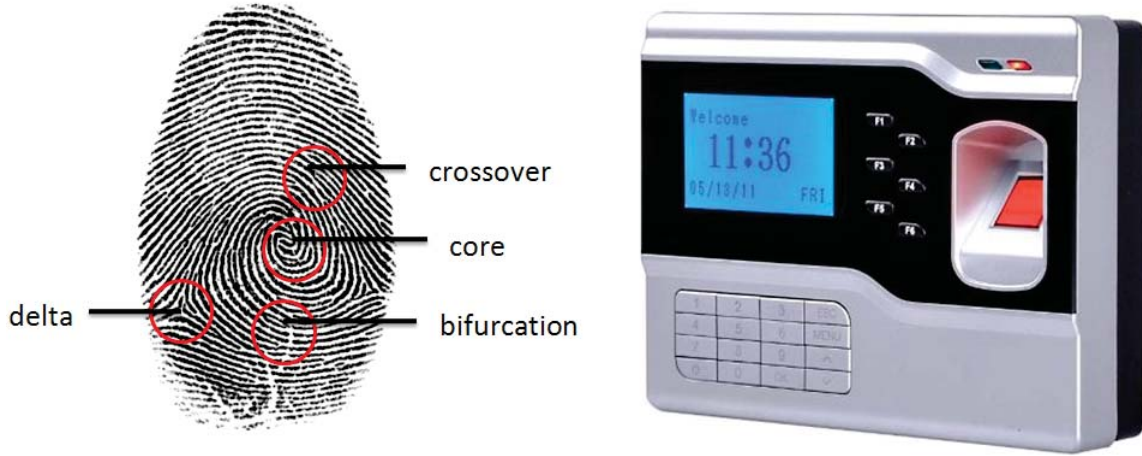
આજકાલ, આપણાં રોજિંદા જીવનનાં ઘણા બધાં પાસાંઓમાં બાયોમેટ્રિક્સ ટેક્નોલોજીનો ઉપયોગ જોવા મળે છે. ફોનના બિલની ચુકવણી માટે અંગત કમ્પ્યૂટરનો ઉપયોગ કરવા અંગેના વિનિયોગમાં આપણને તે જોવા મળશે. આ વિભાગમાં આપણે ઉપયોગ પર નિયંત્રણ અને ઉપયોગકર્તાની ઓળખ તેમજ અધિકૃતતા માટે ઉપયોગમાં લેવાતી બાયોમેટ્રિક્સની આંગળાંઓની છાપ (ફિંગરપ્રિન્ટ), આંખની કીકી અને અવાજ ઓળખ ટેક્નિક વિષે ચર્ચા કરીશું.

આંગળાંઓની છાપની ઓળખ

સદીઓથી સહીને બદલે આંગળાંઓની કે અંગૂઠાની છાપનો ઉપયોગ થતો આવ્યો છે. સામાન્ય રીતે એવી કોઈ વ્યક્તિ કે જેને લખતાં-વાંચતાં આવડતું ન હોય તેને જો કોઈ કાનૂની દસ્તાવેજ પર સહી કરવાની જરૂર પડે, તો તેવા સંજોગોમાં તે શાહીવાળો અંગૂઠો કરી દસ્તાવેજ પર નિયત જગ્યા પર અંગૂઠો દબાવી અંગૂઠાની છાપ પાડે છે. વ્યક્તિને ઓળખવા માટેની આ એક અધિકૃત રીત છે.

આધુનિક ફિંગરપ્રિન્ટ રેકગ્નિશન વ્યવસ્થા આંગળાંનાં ટેરવા પર આવેલી અજોડ, બારીક રેખાઓની રચના કે ભાત દ્વારા બનતી છાપનો ઉપયોગ કરે છે. સંશોધકોએ અવલોકન દ્વારા એ સાબિત કર્યું છે કે, કોઈ બે આંગળાંઓની છાપ ભલે સરખી દેખાતી હોય પરંતુ, તેની રેખાઓની સંખ્યા અને રચનાની ભાત એકસરખી હોતી નથી. વળી, આંગળાંઓ પરની રેખાઓની આ જન્મજાત કુદરતી રચના ક્યારેય બદલાતી કે બગડતી પણ નથી, સિવાય કે અકસ્માત જેવા અસામાન્ય સંજોગો.

આકૃતિ 13.9માં આંગળીની છાપ (ફિંગરપ્રિન્ટ)ની લાક્ષણિકતાનું ઉદાહરણ અને હાજરી નોંધવા માટે ઉપયોગમાં લેવાતું 'ફિંગરપ્રિન્ટ રીડર' દર્શાવાયું છે. આજકાલ બજારમાં મળતા મોટા ભાગનાં લેપટોપ અધિકૃતતા તપાસવા માટે આંગળાંઓની છાપ દ્વારા ઓળખ માટેની (Fingerprint Recognition) પદ્ધતિ ધરાવે છે.



આકૃતિ 13.9 : ફિંગરપ્રિન્ટ રીડર

કોષ્ટક 13.1 આંગળાંઓની છાપની ઓળખ કેટલીક સરકારી, ફોરેન્સિક અને વાણિજ્યિક વિનિયોગની યાદી રજૂ કરે છે :

સરકારી	ફોરેન્સિક	વાણિજ્યિક
<ul style="list-style-type: none"> ● AADHAR કાર્ડ, રેશનકાર્ડ, BPL કાર્ડ વગેરે જેવાં ઓળખપત્રો ● વાહન ચલાવવાના પરવાના (ડ્રાઇવિંગ લાયસન્સ) ● કલ્યાણ યોજનાઓના નાણાંની ભરપાઈ ● પાસપોર્ટ-નિયંત્રણ વગેરે 	<ul style="list-style-type: none"> ● ગુનાઈત વ્યક્તિની તપાસ માટે ● સેના કે રક્ષકદળની ઓળખ માટે ● ગુના અંગેની માહિતી રાખવા ● ત્રાસવાદીઓની ઓળખ માટે 	<ul style="list-style-type: none"> ● કમ્પ્યૂટરને નેટવર્ક લૉગ ઓન ● ઇલેક્ટ્રોનિક ડેટાની સલામતી માટે ● ATM કાર્ડ, ડેબિટ કાર્ડ જેવાં 'સ્માર્ટકાર્ડ' માટે ● સવલતના ઉપયોગના નિયમન માટે (Facility Access Control) ● પર્સનલ ડિજિટલ આસિસ્ટન્ટ (P. D. A.) ● દૂરવર્તી શિક્ષણ (Distance Learning)

કોષ્ટક 13.1 : આંગળાંઓની છાપની ઓળખ માટેના કેટલાક ઉપયોગો

આંખની કીકીના સ્નાયુની ઓળખ (Iris Recognition)

આ પુસ્તક વાંચી રહેલા મોટા ભાગના વાચકો AADHAR (આધાર)કાર્ડ કઠાવવાની પ્રક્રિયામાંથી પસાર થયા જ હશે. જો તમને યાદ હોય તો આ પ્રક્રિયામાં કોઈ એક સમયે તમને કોઈ એક સાધનની સામે એકીટસે જોઈ રહેવાનું કહેવામાં આવ્યું હશે, જેમાં તમારી આંખની કીકીની આસપાસના સ્નાયુઓની રચનાની તસ્વીર ખેંચવામાં આવે છે. આંખની અંદર રહેલા

સ્નાયુઓને ‘આઈરિસ’ (Iris) તરીકે ઓળખવામાં આવે છે, જે કીકીના કદનું નિયમન કરે છે અને આંખમાં કેટલો પ્રકાશ પ્રવેશે તેનું નિયંત્રણ કરે છે. આ આંખનો રંગીન ભાગ છે, જે સ્નાયુઓમાં રહેલ ‘મેલાટોનીન’ (Melatonin) દ્રવ્યના પ્રમાણ અનુસાર રંગાયેલો હોય છે.

કોઈ પણ વ્યક્તિની કીકીની આસપાસના સ્નાયુની રચના જીવનભર બદલાતી નથી. આંખો પર શક્તિક્રિયા કરવામાં આવે તો પણ તેની લાક્ષણિકતાઓ બદલાતી નથી, વધુમાં તેની 266 અજોડ લાક્ષણિકતાઓ છે. (આ દરેકની વધુ ચર્ચા કરવી આ પુસ્તકના વિષયની મર્યાદા બહાર છે.) બજારમાં આજકાલ ઉપલબ્ધ મોટા ભાગની આઈરિસ ‘રેકગ્નિશન સિસ્ટમ’માં માત્ર 173 લાક્ષણિકતાઓનો જ ઉપયોગ કરે છે. આઈરિસની દૃશ્યમાન પ્રાથમિક લાક્ષણિકતાઓમાં ટ્રાબેક્યુલર મેશવર્ક (Trabecular Meshwork) સૂક્ષ્મ માંસપેશીઓ (tissue) જે આઈરિસને જુદી પાડતી હોવાનો દેખાવ આપે છે.

‘આઈરિસ રેકગ્નિશન સિસ્ટમ’માં આઈરિસની તસવીર ખેંચવા માટે એક નાના ‘હાઈ-રિઝોલ્યુશન’ (High Resolution) કેમેરાની જરૂર પડે છે. તે સામાન્ય રીતે શ્વેત-શ્યામ (બ્લક એન્ડ વ્હાઈટ) તસવીર ખેંચે છે. આવું ચિત્ર લેવા માટે તે ‘ઈન્ફ્રારેડ ઇમેજિંગ’ (Infrared Imaging) પર વધુ મદાર રાખે છે. ઉપયોગકર્તાના ખરેખર ઉપયોગ માટેના વિનિયોગમાં કેમેરાથી 3થી 10 ઈંચ સુધીમાં કોઈ વ્યક્તિ ઊભી હોય તે જરૂરી છે. તે પછી સીધા કેમેરાની સામે 1 ચોરસ ઈંચ વિસ્તારમાં અરીસામાં આઈરિસને કેન્દ્રિત કરે છે. આજકાલ આપણા લેપટોપમાં પણ કમ્પ્યુટરના તાર્કિક ઉપયોગને રોકવા આ પદ્ધતિ મળે છે. તે ડેસ્કટોપ કેમેરાનો ઉપયોગ કરે છે અને આઈરિસની છબી ખેંચવા આશરે 18 ઈંચના અંતર સુધી સારી રીતે કામ કરે છે. આજકાલ જોવા મળતાં આઈરિસ રેકગ્નિશનના કેટલાક મુખ્ય વિનિયોગ નીચે દર્શાવ્યા મુજબ છે:

- રોકડ લેણ-દેણનાં મશીનો પર બેન્કખાતાનો ઉપયોગ કરવા.
- ચોરી રોકવા માટેનાં સાધનો
- મકાનો અને ઘરોમાં અધિકૃત પ્રવેશ માટે
- સ્વયંસંચાલિત આંતરરાષ્ટ્રીય સરહદ ઓળંગવા (બિનઅધિકૃત વ્યક્તિ દ્વારા ગેરકાયદેસર પ્રવેશ અટકાવવા પાસપોર્ટની અવેજીમાં)
- વાહનોને ખોલવાં માટે અને ચાલુ કરવાં
- વિમાન ઉડ્ડયન સલામતી
- બાયોમેટ્રિક-કી ક્રિપ્ટોગ્રાફી
- સેલફોન અને અન્ય ‘તારલેસ’ આધારિત અધિકૃતતા
- પ્રતિબંધિત વિસ્તારોના ઉપયોગ પર નિયંત્રણ માટે
- ત્રાસવાદીઓ પર નજર રાખવા સરકાર દ્વારા ઉપયોગમાં લેવાતા ‘Watch list’ ડેટાબેઝનું સર્જન કરવા.
- ડેટાબેઝનો ઉપયોગ અને કમ્પ્યુટર લોગ-ઈન

અવાજ/બોલીની ઓળખ (Voice/Speech Recognition)

અવાજ અથવા બોલીની ઓળખ (વોઈસ/સ્પીચ રેકગ્નિશન) એ એક બાયોમેટ્રિક ટેકનિક છે, જે કોઈ વ્યક્તિના અવાજનો ઈનપુટ તરીકે ઉપયોગ કરી તેના દ્વારા કરી શકાતી ક્રિયાઓ પર અંકુશ મૂકે છે. તે કોઈ પણ વ્યક્તિના અવાજના તરંગોનું જૈવિક માળખું અને વ્યક્તિના વર્તનની લાક્ષણિકતાઓ એ બંને પર અસર કરતાં લક્ષણો પર મદાર રાખે છે. જોકે આમ તો આપણે તેને ‘વોઈસ/સ્પીચ રેકગ્નિશન’ તરીકે ઓળખીએ છીએ, તેમ છતાં સાચો શબ્દ ‘વોઈસ/સ્પીચ વેરિફિકેશન’ છે.

બેન્કો દ્વારા ટેલીફોન મારફત વ્યવહારો કરતી વખતે અને ખાતાની જાળવણી માટે ફોન કરનારની ઓળખની ચકાસણી કરવા માટે આ ટેક્નોલોજીનો ઉપયોગ થાય છે. મોબાઇલ દ્વારા જે વ્યક્તિ સાથે વાત કરવી હોય તે વ્યક્તિનું માત્ર નામ બોલીને ફોન જોડવા, મોબાઇલ વાપરનારાઓ દ્વારા આ ટેક્નિકનો સઘન ઉપયોગ થાય છે. ફોરેન્સિક વિજ્ઞાનમાં તપાસના ભાગ રૂપે સાક્ષી દ્વારા આપવામાં આવતી પ્રતિક્રિયાઓની સત્યપૂર્ણતા નક્કી કરવા માટે પણ તેનો ઉપયોગ થાય છે.

અન્ય બાયોમેટ્રિક પદ્ધતિઓની જેમજ ‘વોઇસ રેકોર્ડિંગ સિસ્ટમ’ સાથે કામ કરવા માટે, સૌપ્રથમ તમારે સિસ્ટમને તાલીમ આપવી પડે. તાલીમ દરમિયાન અવાજના નમૂનાનું સૌ પ્રથમ એનાલોગ (analog) સ્વરૂપમાંથી સાંખ્યિક (ડિજિટલ) સ્વરૂપમાં રૂપાંતર કરવામાં આવે છે. દરેક વ્યક્તિના વ્યક્તિગત અવાજની લાક્ષણિકતાઓ અલગ તારવીને ભવિષ્યના ઉપયોગ માટે સંગ્રહવામાં આવે છે. આ ટેક્નિકના અસરકારક ઉપયોગની આડે પર્યાવરણીય પરિબળો જેવાં કે, આજુબાજુનો અવાજ, રેકોર્ડિંગ સાધનની ખામી વગેરે પણ ભાગ ભજવે છે. વોઇસ/સ્પીચ રેકોર્ડિંગના કેટલાક ઉપયોગ નીચે મુજબ છે :

સાદી માહિતીને ફરી વગાડવી : વ્યક્તિના અવાજને ઓળખવા માટે ફોરેન્સિક તપાસ અર્થે

કોલ-સ્ટીયરિંગ : બહારથી આવતા નિયમિત ફોન કરનાર વ્યક્તિના અવાજને ઓળખીને તેને આધારે યોગ્ય વિભાગ કે વ્યક્તિ તરફ તબદીલ કરવામાં આવે છે. તેનો ઉપયોગ ગ્રાહક સેવાકાર્યો માટે થાય છે.

ફોન કરનારની આપમેળે ઓળખ : ગ્રાહકની ઓળખ માટે ગ્રાહક સેવાકાર્યો માટે તેનો ઉપયોગ થાય છે.

ઈન્ટરેક્ટિવ વોઇસ-રિસ્પોન્સ (Interactive Voice Response - IVR) મેનુને કાઢી નાંખવું : IVR પૂર્વતૈયારી વિનાની હોય છે અને આપમેળે કોલ સંભાળવાની પદ્ધતિ હવે ગ્રાહકોમાં બિનલોકપ્રિય થતી જાય છે. આ ટેક્નિક પુશબટન મેનુને કાઢી નાંખે છે અને તેના બદલે સીધા કોલ સંભાળવાનું અમલમાં મૂકે છે.

અવાજથી કાર્યરત ક્રિયાઓ (Voice Enabled Operations) : ઉપયોગકર્તાને નંબર જોડવા સક્ષમ બનાવે છે અથવા અવાજ દ્વારા આદેશ આપીને મોબાઇલ ફોન પર ક્રિયા પાર પાડવા સક્ષમ બનાવે છે. આવા વિનિયોગો અંધ અથવા અંશતઃ અંધ ઉપયોગકર્તા માટે ખૂબ જ ઉપયોગી રહે છે. અવાજ વડે આદેશ આપીને આપણે ઍપ્લિકેશન વિન્ડો ખોલી શકીએ, પત્ર ટાઇપ કરી શકીએ અથવા કમ્પ્યુટર બંધ કરી શકીએ. આજની મોટા ભાગની ઍપ્લિકેશન સિસ્ટમ આવી સવલતને સમર્થન આપે છે.

ગ્લોબલ પોઝિશનિંગ સિસ્ટમ (GPS)

વૈશ્વિક સ્થિતિ સ્થાનદર્શક પદ્ધતિ (ગ્લોબલ પોઝિશનિંગ સિસ્ટમ - GPS) એ સ્થિતિ-સ્થાન શોધતા વિનિયોગ (નેવિગેશન ઍપ્લિકેશન) માટે સૌથી વધુ વપરાતી પદ્ધતિ છે. આ પદ્ધતિ સંયુક્ત રાષ્ટ્રના સંરક્ષણ વિભાગ (U.S. Department of Defense) દ્વારા તૈયાર કરવામાં આવી હતી. આ પદ્ધતિમાં પૃથ્વીથી ઉપર ભ્રમણકક્ષામાં મૂકવામાં આવેલા 24 ઉપગ્રહોના નેટવર્કનો સમાવેશ થાય છે. GPS શરૂઆતમાં લશ્કરી ઉપયોગ માટે તૈયાર કરવામાં આવ્યું હતું. પરંતુ 1980માં સરકાર દ્વારા આ પદ્ધતિ નાગરિકો માટે પણ ઉપલબ્ધ બનાવાઈ. આ એક એવી સક્ષમ પદ્ધતિ છે, જે ગમે તેવા હવામાનમાં, દુનિયામાં ગમે ત્યાં, દિવસના ચોવીસેય કલાક કામ કર્યા જ કરે છે. GPSનો ઉપયોગ કરવા માટે અત્યારે કોઈ નોંધણી- ખર્ચ કે ગોઠવણીનો ખર્ચ નથી. GPSના સંદેશા ઝીલી શકે તેવું GPS રિસીવર ધરાવતું કોઈ પણ સાધન આ પદ્ધતિનો ઉપયોગ કરી શકે છે. આજકાલ મળતા મોટા ભાગના બધા સ્માર્ટફોન આંતરપ્રસ્થાપિત GPS રિસીવર વાળા જ હોય છે. કોઈ પણ વ્યક્તિ GPSનો ઉપયોગ કરતું ઍપ્સ (apps) ‘ડાઉનલોડ’ કરીને સ્થિતિ-સ્થાનનિર્દેશ માટે મદદ મેળવી શકે.

GPS ટેકનોલોજી ઘણા વિનિયોગો માટે ઉપયોગી સાબિત થઈ છે. જેવા કે, વહાણો/વિમાનોની ભાળ મેળવવી, સેલ્યુલર ટેલીફોનવ્યવસ્થા, આપત્તિ-રાહત અને આપાતકાલીન સેવાઓ, રોબોટિક્સ. કોઈ પણ વસ્તુના સ્થાનને એકદમ બરાબર શોધી કાઢવા GPS ઉપગ્રહ તેની નિશ્ચિત ભ્રમણકક્ષામાં દિવસમાં બે વાર પૃથ્વીની પ્રદક્ષિણા કરે છે અને પૃથ્વી પર તેના સંકેતો મોકલી આપે છે. GPS રિસીવર (GPSના સંકેતો ઝીલનારું સાધન) આ માહિતી મેળવે છે અને 'ટ્રાયન્ગ્યુલેશન' (Triangulation) નામની પ્રક્રિયા દ્વારા વસ્તુના એકદમ સચોટ સ્થાનની ગણતરી કરે છે. GPS રિસીવર ઉપગ્રહ દ્વારા સંકેત મોકલાયાના સમય સાથે, સંકેત મેળવાયાના સમયની સરખામણી કરે છે. સમયનો તફાવત GPS રિસીવરને જણાવે છે કે, ઉપગ્રહ કેટલો દૂર છે. આ પ્રક્રિયાને વધુ બે કે ત્રણ ઉપગ્રહ માટે ફરીથી કરવામાં આવે છે અને એ પછી રિસીવર ઉપયોગકર્તાનું એકદમ સચોટ સ્થાન નક્કી કરે છે અને એ પછી ઉપયોગકર્તા પાસે ઉપલબ્ધ કોઈ પણ ઇલેક્ટ્રોનિક નકશા પર તે દર્શાવે છે. માત્ર અક્ષાંશ અને રેખાંશની ગણતરી માટે GPS રિસીવર ઓછામાં ઓછા ત્રણ ઉપગ્રહનો ઉપયોગ કરે છે. વધારાની માહિતી જેવી કે, ઝડપ, વર્તણૂક, ધરી, પ્રવાસફેરાનું અંતર, સ્થળનું અંતર, સૂર્યોદય અને સૂર્યાસ્તનો સમય અને એવી વધુ માહિતીના કિસ્સામાં GPS રિસીવર ચાર કે વધુ ઉપગ્રહ સાથે જોડાય તે જરૂરી છે. GPS ટેકનોલોજીના કેટલાક રસપ્રદ વિનિયોગ નીચે દર્શાવેલ છે :

ખેતી : તે વિસ્તારની માપણી, ડિજિટલ (સાંખ્યિક) ખેતીવિષયક નકશાઓ અને ડિજિટલ (સાંખ્યિક) જમીનમાલિકી અંગેના નકશાઓ તૈયાર કરવા ઉપયોગમાં લેવાય છે.

આપત્તિ-વ્યવસ્થાપન : શોધ અને બચાવ-કામગીરી માટે તેનો ઉપયોગ થાય છે. ઉદાહરણ તરીકે ભૂલા પડેલા પગપાળા યાત્રીઓ અને ખોવાયેલા સાહસિક પ્રવાસીઓ, કાટમાળમાં દટાયેલા લોકો અથવા પાણીમાં ગરકાવ થયેલા લોકો પાસે જો સેલફોન હોય અથવા GPS પદ્ધતિ ચાલુ હોય, તો તેમને શોધી શકાય છે.

લશ્કરને લગતું : તે લક્ષ્યસ્થાન તેમજ વિમાનની સચોટ ઓળખ માટે વિશિષ્ટ વિમાનો જેવાં કે, લડાયક વિમાન (ફાઈટર પ્લેન), હવાઈ ટેન્કરો, બોમ્બ ફેંકનાર વિમાન (bombers) વગેરેમાં ઉપયોગમાં લેવાય છે. નૌકાદળનાં વહાણો પણ આવા જ કાર્ય માટે તેનો ઉપયોગ કરે છે. ઉપગ્રહ પર ભ્રમણકક્ષાના સચોટ ડેટા મેળવવા અને અવકાશયાનની દિશા નક્કી કરવા માટે પણ તેનો ઉપયોગ કરાય છે.

વહાણ દ્વારા કે અન્ય રીતે મોકલાવાયેલ માલની ભાળ મેળવવી : ગ્રાહકને મોકલાવાયેલ માલ કે ટપાલ પહોંચેલ છે કે નહીં તેની દેખરેખ માટે મોટા ભાગની કુરિયર સેવાઓ આપતી સંસ્થાઓ 'GPS ટ્રેકિંગ સિસ્ટમ'નો ઉપયોગ કરે છે. એ જ રીતે પરિવહન કાર્યમાં રોકાયેલ કંપનીઓ તેમની પરિવહનસેવા પર દેખરેખ રાખવા તેનો ઉપયોગ કરે છે. આ ટેકનોલોજીનો ઉપયોગ કરતી કંપની, જો તેનું કોઈ વાહન મોડું પડ્યું હોય, તો તે ચકાસી શકે છે અને ઝડપથી પહોંચવા માટેનો કોઈ રસ્તો બતાવી શકે છે. એ જ રીતે આ પદ્ધતિથી તે વાહન હાલ ક્યાં છે, તે સ્થાન જાણી શકે છે.

ખોવાયેલ મિલકતો શોધી કાઢવા : સલામતી સેવાઓ આપતી કંપનીઓ GPS સુવિધા ધરાવતા ખોવાયેલ કે ચોરાયેલ લેપટોપ કે મોબાઇલ ફોન જેવા સાધનો અથવા વાહનોનું સચોટ સ્થાન જાણી શકે છે અને એવાં સાધનો કે વાહનોને ખૂબ જ સરળતાથી શોધી શકે છે.

સંગ્રહ-સાધનો અંગે નવતર પ્રવાહ

તમે ક્લાયન્ટ-સર્વર વિનિયોગના પરિચયમાં આવ્યા જ હશો, જેમાં એક કમ્પ્યુટર મશીન સાધન/સુવિધા પ્રદાન કરનાર તરીકે કાર્ય કરે છે. જ્યારે અન્ય કમ્પ્યુટર મશીન સુવિધાનો ઉપયોગ કરનારા હોય છે. ફાઇલસર્વર એક આવો જ વિનિયોગ છે, તે તેની સાથે જોડાયેલાં તમામ ક્લાયન્ટ મશીનને ફાઇલ સંબંધી સુવિધાઓ પ્રદાન કરે છે. સામાન્ય રીતે ફાઇલસર્વર એ એક મશીન હોય છે, જે ઉપયોગકર્તાને લોગ-ઇન થવા માટે તેની અધિકૃતતા ચકાસે છે, અને સિસ્ટમ વહીવટકર્તા (એડમિનિસ્ટ્રેટર) એ નિયત કરેલ ફાઇલના ઉપયોગ અંગેના નિયંત્રણની નીતિના આધારે ફાઇલોનો નિયંત્રિત ઉપયોગ કરવા દે છે. અનેકવિધ

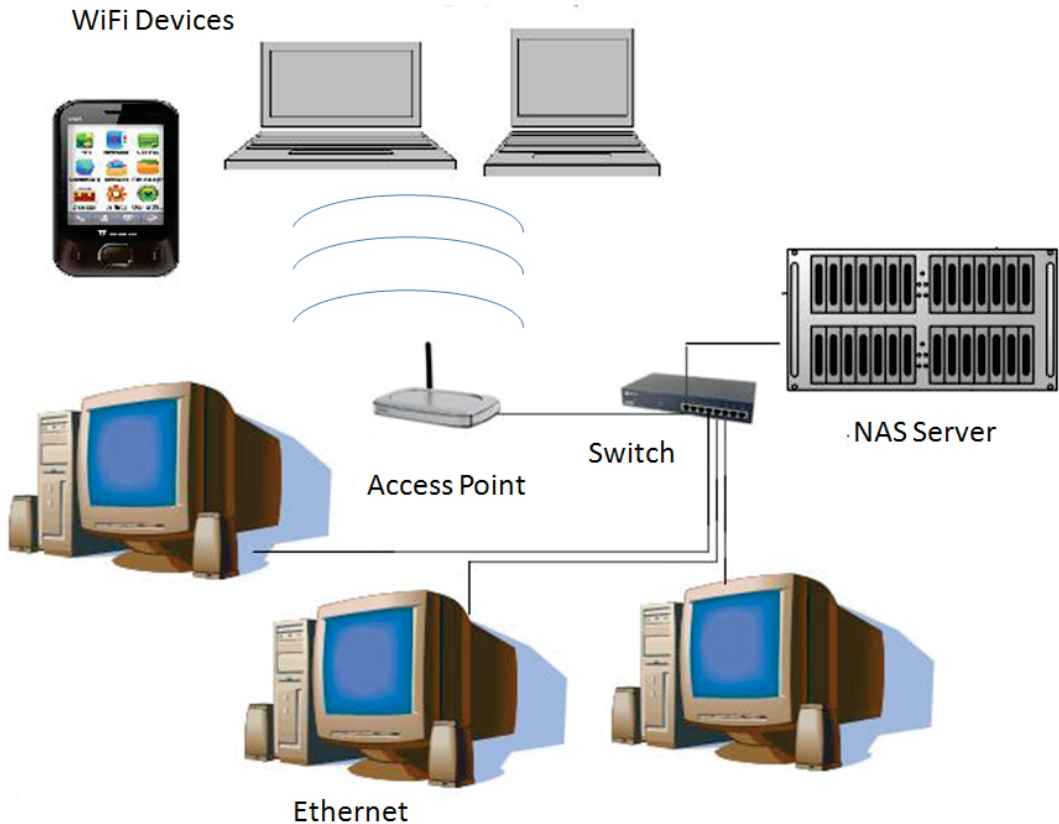
કાર્યોથી વ્યસ્ત રહેવાને કારણે આવા સંશોધનનો ઉપયોગ ધીમો પડે છે. ઝડપી પ્રક્રિયા ક્ષમતા ધરાવતાં અદ્યતન મશીનોની ઉપલબ્ધતાને કારણે માહિતી-સંગ્રહ અંગેનું ચિત્ર પણ બદલાઈ ગયું છે. આજકાલ આપણે ડેટાને સંગ્રહવા કે પછી માત્ર ડેટાનો બેક-અપ લઈ રાખવા માટે વધારાના સાધન તરીકે USB સમર્થિત બાહ્ય હાર્ડડ્રાઈવનો ઉપયોગ કરીએ છીએ. સર્વર પણ હવે ટેરાબાઈટના માપની ક્ષમતા ધરાવતા મળવાં લાગ્યાં છે. પરંતુ આવાં બાહ્ય સાધનોનું સંચાલન કરવું હંમેશાં અડચણરૂપ હોય છે. આ વિભાગમાં આપણે ડેટા-સંગ્રહ સાધનો અંગેના કેટલાક તાજેતરના પ્રવાહો વિશે ચર્ચા કરીશું.

નેટવર્ક એટેચ્ડ સ્ટોરેજ (Network Attached Storage - NAS)

કોઈ પણ સંસ્થામાં કમ્પ્યુટર કામગીરીની ક્ષમતા વિસ્તારવા દ્વાયકાઓથી નેટવર્કનો ઉપયોગ કરવામાં આવે છે. આપણે એ જાણીએ છીએ કે, નેટવર્ક-પ્રિન્ટર (કોઈ કમ્પ્યુટર સાથે સીધું જોડવાને બદલે નેટવર્કના સાધન તરીકે જોડાયેલ પ્રિન્ટર) મારફત આપણે છાપવાનું કાર્ય કરી શકીએ છીએ. આવા નેટવર્ક પ્રિન્ટરને પોતાની એક અલાયદી ઓળખ (અજોડ IP એડ્રેસ) હોય છે, જે તેને કમ્પ્યુટર વચ્ચે વહેંચેલ (shared) પ્રિન્ટર કરતાં વધુ ઝડપે કાર્ય કરી આપવાનું શક્ય બનાવે છે. નેટવર્ક એટેચ્ડ સ્ટોરેજ (NAS) એટલે કે નેટવર્ક સાથે જોડાયેલ સંગ્રહ-વ્યવસ્થા પણ કંઈક આવી જ ટેકનિક છે. તે હાર્ડડિસ્કની સંગ્રહ અંગેની તંત્રવ્યવસ્થા (હાર્ડડિસ્ક સ્ટોરેજ મિકેનિઝમ) છે, એટલે કે કોઈ વ્યક્તિના કમ્પ્યુટર સાથે સીધું લગાડવાને બદલે તેને તેનું પોતાનું એક નેટવર્ક એડ્રેસ આપવામાં આવે છે.

NAS ડિવાઈસમાં ટેરાબાઈટ ક્ષમતાવાળી એક અથવા આવી એક કરતાં વધુ હાર્ડડિસ્ક હોઈ શકે. તે ‘લોકલ એરિયા નેટવર્ક’ સાથે જોડેલી હોય છે અને તેને એક અજોડ IP એડ્રેસ આપવામાં આવે છે. ઉપયોગકર્તા જ્યારે મુખ્ય સર્વર પાસે કોઈ એક ચોક્કસ ફાઈલ માંગશે ત્યારે મુખ્ય સર્વર દ્વારા ફાઈલ માટેની ઉપયોગકર્તાની વિનંતી ચકાસવામાં આવશે, જેથી ઉપયોગકર્તા તે ફાઈલનો ઉપયોગ કરવા અધિકૃત છે કે કેમ તે જાણી શકાય.

ફાઈલસંગ્રહનો ઉપયોગ તેમજ ‘ઓથેન્ટિકેશન સર્વર’માંથી તેનું સંચાલન દૂર કરવાને લીધે, હવે આ માટે એક જ પ્રોસેસર સંસાધન પર ભારણ ઘટી જવાને કારણે એપ્લિકેશન પ્રોગ્રામિંગ અને ફાઈલોને મોકલવાની અને મેળવવાની બન્ને પ્રક્રિયા ઝડપી બનશે.

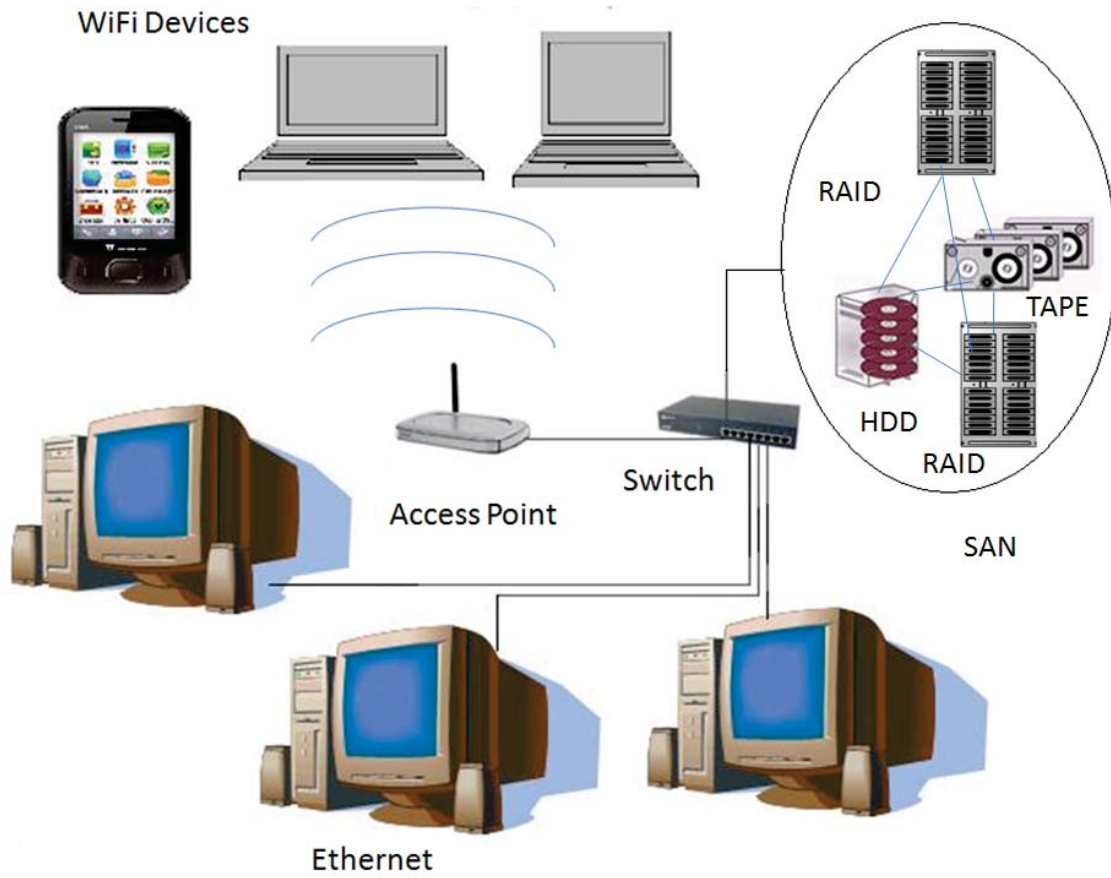


આકૃતિ 13.10 : NAS આર્કિટેક્ચર

NAS પદ્ધતિમાં સંગીત, ચલચિત્રો, છબીઓ અને અન્ય ફાઇલોને લગતો આપણો ડેટા આપણે કોઈ એક અનુકૂળ સ્થળેથી ઝડપથી અને સરળતાથી સંગ્રહી શકીએ છીએ અને અન્ય લોકોને વહેંચી શકીએ છીએ. આ સંસાધનોનો ઉપયોગ લેપટોપ, ડિજિટલ કેમેરા અથવા સ્માર્ટફોન જેવા વિવિધ ક્લાયન્ટ દ્વારા પણ કરી શકાય છે. આકૃતિ 13.10 કોઈ એક સંસ્થામાં NAS ના અમલનું લાક્ષણિક દૃશ્ય રજૂ કરે છે.

સ્ટોરેજ એરિયા નેટવર્ક (Storage Area Network - SAN) :

સ્ટોરેજ એરિયા નેટવર્ક, એનું નામ જ સૂચવે છે તે પ્રમાણે તે સંગ્રહ માટેનું સમર્પિત નેટવર્ક છે. તે એકત્રિત કરેલ બ્લોક સ્તરના ડેટાસંગ્રહનો ઉપયોગ પ્રદાન કરે છે. SANનો મૂળ ઉદ્દેશ્ય, એપ્લિકેશન સર્વરને જુદાં-જુદાં સંગ્રહ-સાધનો જેવાં કે હાર્ડડિસ્કનો સમૂહ, ટેપનો સમૂહ અને ઓપ્ટિકલ સંગ્રહ-સાધનો વગેરે જેવાં વિવિધ સંગ્રહ-સાધનોનો એકસરખો ઉપયોગ પૂરો પાડવાનો છે. આવા એકરૂપ ઉપયોગની લાક્ષણિકતાને કારણે આવા વિવિધ જાતનાં સાધનો જાણે કે ઓપરેટિંગ સિસ્ટમની સાથે જ જોડાયેલ હોય તેવો આભાસ ઊભો થાય છે. આકૃતિ 13.11 કોઈ સંસ્થામાં અમલમાં મૂકેલ SANનું લાક્ષણિક દૃશ્ય રજૂ કરે છે.



આકૃતિ 13.11 : SAN આર્કિટેક્ચર

એ નોંધો કે SAN એ લોકલ એરિયા નેટવર્કનો જ એક પ્રકાર છે, કે જે મોટા પાયે ડેટાની હેરફેરનું સંચાલન કરવા તૈયાર કરવામાં આવી છે. તે ડેટાસંગ્રહ, ડેટાની પુનઃપ્રાપ્તિ અને પ્રતિકૃતિઓ તૈયાર કરવા જેવાં કાર્યો કરવા દે છે. આ ટેકનોલોજીનો લાભ લેવા માટે આપણે ઉચ્ચ કાર્યક્ષમતાવાળાં સર્વર, એકથી વધુ ડિસ્કનો સમૂહ અને આ બધા વચ્ચે પ્રત્યાયન માટે ફાઇબર યેનલના આંતરજોડાણ વ્યવસ્થાની જરૂર પડે.

ઑનલાઇન સ્ટોરેજ વિકલ્પો (Online Storage options)

ક્લાઉડ કમ્પ્યૂટિંગના વિભાગમાં આપણે Google ડ્રાઇવ વિશે શીખ્યા. ક્લાઉડ કમ્પ્યૂટિંગ સવલતની શોધની સાથે જ આપણને ઑનલાઇન સંગ્રહ માટેની સુવિધા પણ મળી. ઑનલાઇન સંગ્રહની સુવિધાનો ઉપયોગ અંગત તેમજ ધંધાના હેતુ માટે પણ કરી શકીએ. ઑનલાઇન સંગ્રહ-સુવિધા એ ખાસ કરીને આપણે જ્યારે બેક-અપ રાખવાનું અને આપણા ડેટાનો ગમે ત્યાંથી સાર્વત્રિક ઉપયોગ કરવાની જરૂરિયાત હોય, ત્યારે મદદરૂપ બને છે. આ સુવિધાની મહત્તા એ છે કે આપણે દુનિયાના કોઈ પણ ભાગે રહી, કોઈ પણ સાધન વડે આપણા આ ડેટાનો ઉપયોગ કરી શકીએ છીએ. બસ ફક્ત ઈન્ટરનેટની સારી જોડાણવ્યવસ્થા હોવી જોઈએ. કેટલાક ઉપલબ્ધ નિઃશુલ્ક ઑનલાઇન સંગ્રહ-વિકલ્પોમાં A Drive, Microsoft Live Sky-Drive, 4Shared, Mozy, Dropbox, Box, MediaFire, DivShare વગેરે ગણાવી શકાય. તેઓ ઑનલાઇન સંગ્રહ માટે જુદી-જુદી સેવાઓ પૂરી પાડે છે અને આપણને સંગ્રહની જુદી-જુદી ક્ષમતા પ્રદાન કરે છે. નિઃશુલ્ક વિકલ્પોની જેમ સંગ્રહ માટેના વિકલ્પો ભાડેથી પણ ઉપલબ્ધ છે. આવી સેવા પ્રદાન કરનારા, નિઃશુલ્ક સેવા આપનારા કરતાં વધુ સારી સેવા અને સલામતીની ખાતરી આપે છે. IBackup અને Amazon S3 એ આવા ભાડેથી સંગ્રહ-સુવિધા પ્રદાન કરનારાના ઉદાહરણ છે. એક રીતે જોઈએ તો આ ઑનલાઇન સ્ટોરેજ વિકલ્પોને ક્લાઉડ સેવા ગણી શકાય.

કમ્પ્યૂટર નિયંત્રિત સાધનો (Computer Controlled Devices) :

દિવસે ને દિવસે કમ્પ્યૂટર વધુ ચપલ અને વધુ બુદ્ધિમત્તાવાળાં બનતાં જાય છે. કમ્પ્યૂટરની આ નવી ક્ષમતા દ્વારા આપણે સૌ પ્રભાવિત થયા છીએ. એ કહેવામાં જરાય નવાઈ નથી કે ઘણીવાર આપણા માટે એ જાણવું ખરેખર મુશ્કેલ બની જાય છે કે, ફોન પર સામેના છેડે આપણી સાથે વાત કરનાર એ ખરેખર માણસ છે કે પછી મશીનમાં સંગૃહીત અવાજ.

આપણે સંસ્થાઓના સ્વયંસંચાલન વિશે સાંભળીએ છીએ, જેમાં પહેલાં જે કાર્યો માણસોના સમૂહ દ્વારા થતાં હતાં તેને બદલે તેમનું સ્થાન મશીનોએ લીધું. ઉદાહરણ તરીકે, ઠંડા પીણાંની બોટલો માણસો દ્વારા ભરવાને બદલે હવે બોટલિંગ પ્લાન્ટ દ્વારા ભરાય છે. જુદા જુદા આકાર અને કદની બાટલીઓ ભરવા માટે પ્રોગ્રામ દ્વારા નિયંત્રિત મશીનનો ઉપયોગ કરવામાં આવે છે. કમ્પ્યૂટર હાર્ડવેર અને સૉફ્ટવેર ટેકનોલોજીના વિકાસની સાથે રોબોટિક્સ (Robotics)નું ક્ષેત્ર પણ વિકાસ પામ્યું છે. રોબોટિક્સનો અભ્યાસ મુખ્યત્વે ગતિશીલ પરિસ્થિતિમાં સેન્સર (Sensors) અને એક્ચ્યુએટર (Actuators)ના ઉપયોગ દ્વારા રોબોટ કેવો અસરકારક રીતે કામ કરી શકે છે તે જાણવાનો છે. આજકાલના રોબોટ કમ્પ્યૂટર દ્વારા નિયંત્રિત સંપૂર્ણ સ્વયંસંચાલિત મિકેનિકલ અને ઈલેક્ટ્રોનિક મેનિપ્યુલેટર (Manipulators) હોય છે.

રોબોટિક્સના વિનિયોગોનાં કેટલાંક ક્ષેત્રોમાં બુદ્ધિશાળી ઘર (Intelligent Home), લશ્કરી વિનિયોગ, ખેતી અને પશુપાલન, વાહનવ્યવહાર; ઉદ્યોગો, ઈસ્પિતાલો, આપત્તિ-વ્યવસ્થાપન અને મનોરંજનને ગણાવી શકાય. તો ચાલો, આ બધા વિનિયોગોમાં રોબોટનો ઉપયોગ કેવી રીતે થાય છે તે ટૂંકમાં જોઈએ.

બુદ્ધિશાળી ઘર (Intelligent Home) : આજકાલ ઘરની સલામતી અને અન્ય આંતરિક કામગીરી સ્વયંસંચાલિત પદ્ધતિ દ્વારા કરવામાં આવે છે. ઘરનાં બારી-બારણાં કોઈ ચોક્કસ ક્રિયા કે આદેશના આધારે આપમેળે ખોલી શકાય છે. ઈલેક્ટ્રોનિક ઉપકરણો જેવાં કે, લાઈટ, પંખા, હિટર, ઍરકંડિશનર વગેરે ચાલુ કરવાં, બંધ કરવાં અથવા આ ક્રિયાની વચ્ચે તેની ગોઠવણી બદલવા પહેલેથી તેનો પ્રોગ્રામ ગોઠવી શકાય છે. આ ટેકનોલોજી તેના રહેવાસીને, તે ક્યાં છે, તેની પરવા કર્યા વિના તેને મદદરૂપ થાય છે. ઘરમાં આરામથી હરી-ફરી શક્તી વ્યક્તિ હોય કે પછી પથારીવશ હોય તેવી વ્યક્તિ દ્વારા પણ આ ટેકનોલોજીનો એકસરખી સરળતાથી ઉપયોગ કરી શકાય છે.

લશ્કરી વિનિયોગ (Military Applications) : તમારી આસપાસના વિસ્તારોની અંદર કે આજુબાજુ થતી ક્રિયાઓથી સલામતી માટેની ચાવી છે, ચાંપતો બંદોબસ્ત. જો તમને 3 ઈડિયટ્સ ફિલ્મનું એ દૃશ્ય યાદ હોય કે જેમાં આમિરખાન જે ઓબ્જેક્ટને સુધારા-વધારા કરી કાર્યરત કરે છે, તે ખરેખર ચાંપતા બંદોબસ્ત માટે બનાવાયું હોય છે. આજના આધુનિક લશ્કરમાં હવામાં ઊડતો રોબોટ (Airborne robot drones) ચાંપતા બંદોબસ્ત માટે ઉપયોગમાં લેવાય છે. આ હવામાં ઊડતો રોબોટ (એર-બોર્ન ડ્રોન) એ પક્ષીના આકારનું અથવા અન્ય વસ્તુના આકારનું નાનું ઊડતું યંત્ર હોય છે, જેમાં વધુ ઉચ્ચ ગુણવત્તાવાળા કેમેરા, સેન્સર (sensors) અને માહિતીના આદાનપ્રદાન માટે તેમજ નેટવર્કિંગ માટેના સાધનો લગાડેલાં હોય છે. તે કોઈ પણ વ્યક્તિ પાસેથી દેશ માટે રસના વિષય બાબતે કોઈ પણ વસ્તુ (ઓબ્જેક્ટ)ના મોટા પ્રમાણમાં ડેટા ભેગો કરવા માટે ઉપયોગમાં લેવાય છે.

ખેતી અને પશુપાલન (Farming and Animal Husbandry) : ખેતી એ એક એવું ક્ષેત્ર છે, જેમાં રોબોટિક્સની ચકાસણી માટે અનેક પ્રયાસો કરવામાં આવ્યા છે. આવા કેટલાક ઉપયોગોમાં જમીનની સ્થિતિ અનુસાર આપમેળે પાણી છોડવું, આપમેળે પાકની લણણી કરવી, ગાયોને દોહવી વગેરે છે.

વાહન-ઉદ્યોગ (Vehicle Industry) : વાહન-ઉદ્યોગ આજકાલ બજારમાં ઉપલબ્ધ સ્વયંચાલિત સાધનોનો બહોળા પ્રમાણમાં સાધનોનો ઉપયોગ કરે છે. તે કારઉત્પાદન પ્રક્રિયા દરમિયાન યાંત્રિક હાથ (રોબોટિક આર્મ્સ)નો ઉપયોગ કરે છે. જે અનેકવિધ કામો કરી શકવા સક્ષમ છે. રોબોટ જે કામ કરી શકે તેમાં વેલ્ડિંગ, કટિંગ, ઊંચકવાનો (lifting), જુદું તારવવાનો (sorting) અને વાળવા (bending)નો સમાવેશ થાય છે. આજનાં વાહનો હવે વધારાના પ્રોસેસર સાથે ઉપલબ્ધ બનવા લાગ્યાં છે, જે તેને અગાઉનાં વાહનોની સરખામણીએ વધુ ચપળ (સ્માર્ટ) બનાવે છે. GPS આધારિત સ્થાન-સ્થિતિનિર્દેશ પદ્ધતિ (Navigation System) સાથેના વાહનો આજકાલ કંઈ નવાઈની વાત નથી. તેનો ઉપયોગ કારને કઈ તરફ લઈ જવી તે દિશા-નિર્દેશ માટે, જગ્યા શોધવા માટે, જે રસ્તે જઈ રહ્યા છીએ તે રસ્તો સાચો છે કે કેમ તેની ખાતરી કરવા માટે અને બીજાં ઘણાં કામ માટે થાય છે.

ચાલક (ડ્રાઈવર) વિનાની કાર તૈયાર કરવામાં ગૂગલે (Google) સફળતા હાંસલ કરી છે. ચાલક વિનાની કાર બનાવવાની ટેક્નોલોજી બનાવવાની યોજનાને ગૂગલ દ્વારા પીઠબળ મળ્યું છે. નેવાડા (U.S. state of Nevada) દ્વારા 29મી જૂન, 2011ના રોજ નેવાડા અને કેલિફોર્નિયામાં ચાલક વિનાની કારની ગતિવિધિઓ કરવા મંજૂરી આપતો કાયદો સંયુક્ત રાષ્ટ્રસંઘ, પસાર કરી દેવામાં આવ્યો. નેવાડાનો કાયદો 1લી માર્ચ, 2012ના રોજ અસ્તિત્વમાં આવ્યો અને 2012ના મે મહિનામાં તો નેવાડાના મોટર વાહન વિભાગે પ્રથમ સ્વયંચાલિત કાર (self-driven car) માટે પહેલો પરવાનો (લાઈસન્સ) આપી પણ દીધો. ચાલક વિનાની આ પ્રથમ કાર કે જેને પ્રથમ પરવાનો અપાયો તે કાર ગૂગલની ચાલક રહિત ટેક્નોલોજીના પ્રયોગો સાથે સુધારેલી 'ટોયોટા પ્રાયસ' (Toyota Prius) છે.

Rotating Sensors to generate
3D map of surroundings

Video camera to detect moving
traffic and traffic lights



Radar sensors at front and back
to determine position

આકૃતિ 13.12 : ગૂગલની ચાલકરહિત કાર

ચાલકરહિત કારની યોજના આગળ ધપાવવા; ટુકડી દસ વાહનોથી સજ્જ હતી, જેમાં જરૂરી ટેકનોલોજી સાથે છ 'ટોયોટા પ્રાયસ', એક ઓડી ટીટી (Audi TT) અને ત્રણ લેક્સસ RX450h (Lexus RX4504) નો સમાવેશ કરવામાં આવ્યો હતો. આ આખી પદ્ધતિ, કારમાં સંગ્રહેલ સ્થાન-સ્થિતિનિર્દેશ કરતો નકશો (Navigation Map) અને સેન્સર પર આધારિત છે. કાર તેની અંદર સંગ્રહેલ નકશામાં દર્શાવેલી ગતિ મુજબ જ દોડે છે. તેમાં લગાવેલા સેન્સરના આધારે તે બીજાં વાહનોથી પોતાને દૂર રાખશે. વળી, તેમાં એવી પદ્ધતિ પણ છે કે જેમાં જરૂર પડ્યે તે કોઈ વ્યક્તિ (ચાલક)ને પણ કાર ચલાવવાનો કબજો લેવા દે. આ કારની ચકાસણી કરતી વખતે તેમાં બે વ્યક્તિઓને બેસાડવામાં આવી હતી, એક અનુભવી ચાલક અને બીજો એન્જિનિયર. જો કોઈ દુર્ઘટના ઘટે તેવું જણાય, તો કારમાં બેઠેલા અનુભવી ચાલકે કાર પર કાબૂ મેળવી લેવાનો. જ્યારે જો કારના સ્વયંચાલનમાં કાર્યરત સોફ્ટવેરમાં કોઈ ખામી જણાય તો તે કારમાં બેઠેલા એન્જિનિયરે સુધારી લેવાની હતી.

ઈસ્પિતાલો : સ્વાસ્થ્યસંભાળ પણ એક ક્ષેત્ર છે, જેમાં યંત્રમાનવવિજ્ઞાન (રોબોટિક્સ) પગપેસારો કરીને લોકોનાં જીવન બચાવવામાં મદદરૂપ થવા સાથે લોકોને તેમના જીવનમાં સુધારો લાવવા માટે સહાયરૂપ બની રહ્યું છે. સંશોધકોએ એવું રોબોટિક સ્યુટ (robotic suit) તૈયાર કર્યું છે, જે દર્દીની પીઠને નુકસાન પહોંચાડ્યા વગર દર્દીને ઊંચકવામાં પરિચારિકા (નર્સ)ને સક્ષમ બનાવે છે. મગજમાં નાનાં-નાનાં સેન્સરો સાથે જોડવામાં આવેલા રોબોટિક અવયવોને બ્રેઇનગેટ (BrainGate) તરીકે ઓળખાય છે. જે કોઈ ભૌતિક ફિલ્મને બદલે વ્યક્તિના વિચારો પર કાર્ય કરે છે. તદ્ઉપરાંત રોબોટિક હાથ (Robotic Arm) ચોક્કસાઈપૂર્વક શસ્ત્રક્રિયા પણ કરી શકે છે.

આપત્તિ-વ્યવસ્થાપન અને બાહ્ય અંતરિક્ષ અભિયાન (Outer space expedition and disaster management) : રોબોટ, જવાળામુખી, ઊંડા દરિયા અને ચંદ્ર જેવા માનવજાત માટે નુકસાનકારક વાતાવરણની મુલાકાત પણ લઈ શકે છે. ધરતીકંપ અને એવી અન્ય કુદરતી આપત્તિઓના સમયે બચાવકામગીરીના સંદર્ભે આમની પસંદગી એક સારી પસંદગી સાબિત થઈ શકે છે.

ભારતના 'ચંદ્રયાન-1' નામના ચંદ્ર અભિયાન વિશે તો તમે સાંભળ્યું જ હશે. ભારતીય અંતરિક્ષ સંશોધન સંસ્થા (ISRO) દ્વારા શરૂ કરવામાં આવેલ માનવ રહિત ચંદ્ર અભિયાનોમાંનું તે એક હતું, જે ઓક્ટોબર-2008માં શરૂ કરવામાં આવ્યું હતું અને છેક ઓગસ્ટ-2009 સુધી ચલાવવામાં આવ્યું હતું. ચંદ્રયાન-1 પર રહેલ NASAના સાધનોએ અનુમાન કરતાં વધુ પાણીના અણુ શોધી બતાવ્યા હતા. ચંદ્ર ઉપર પાણી એ આ અભિયાનનું એક સૌથી વધુ કટોકટીભર્યું તારણ હતું.

મનોરંજન : મનોરંજન ઉદ્યોગ પણ રોબોટિક્સ પ્રવાહોની અસરમાં આવ્યા વિના રહ્યો નથી. પરસ્પર સંવાદ સાધી શકતો, શીખવાની ક્ષમતા અને વર્તણૂંક પ્રદર્શિત કરતા રોબોટ Sony અને Honda કંપની દ્વારા બનાવવામાં આવ્યા હતા. Sony કંપની દ્વારા બનાવવામાં આવેલ રોબોટ QRIO (Quest for cuRIOsity) જેનું મૂળ નામ SONY Dream Robot એ એક હ્યુમનોઇડ (Humanoid) રોબોટ હતું, જે મુક્ત રીતે હરી-ફરી શકતો હતો, ગોલ્ફબોલ રમી શકતો હતો અને મૌખિક સૂચનાઓને પણ અનુસરી શકતો હતો. જ્યારે Honda કંપની દ્વારા બનાવવામાં આવેલો HEARBO નામનો રોબોટ એકસાથે આવતા અનેક અવાજોને મેળવી તેને કોઈ પણ જાતની તકલીફ વગર જુદા તારવીને તેનું પૃથક્કરણ પણ કરી શકે. તે એક બાજુ રમી રહેલાં બાળકોના અવાજને સરળતાથી ભેદ પારખી અલગ તારવી શકે. આકૃતિ 13.13માં પ્રેક્ષકોને સલામ કરતો રોબોટ QRIO દર્શાવાયો છે.



આકૃતિ 13.13 : પ્રેક્ષકોને સલામ કરતો QRIO

એક ક્ષેત્ર કે જે કમ્પ્યુટર દ્વારા નિયંત્રિત સાધનોના ઉપયોગને નોંધપાત્ર રીતે વિકસાવવામાં મદદરૂપ નીવડ્યું છે તે છે, કૃત્રિમ બુદ્ધિમત્તા. (આર્ટિફિસિયલ ઇન્ટેલિજન્સ), જેને ટૂંકમાં AI તરીકે ઓળખવામાં આવે છે. માનવીને બુદ્ધિમત્તાની ભેટ મળેલી છે. જે આપણને જ્ઞાન અને આવડત પ્રાપ્ત કરવાની ક્ષમતા બક્ષે છે. વળી, તે આપણે મેળવેલા જ્ઞાન અને આવડતનો ઉપયોગ કરી કોઈ સમસ્યાનો ઉકેલ મેળવવા દે છે, પછી ભલે તે સામાન્ય હોય કે જટિલ. કૃત્રિમ બુદ્ધિમત્તાનો હેતુ કમ્પ્યુટરમાં એવી ક્ષમતા ઊભી કરવાનો છે કે જેથી તે જ્ઞાન મેળવી શકે અને જરૂરિયાતના સમયે તેનો ઉપયોગ કરી શકે.

કમ્પ્યુટરમાં કૃત્રિમ બુદ્ધિમત્તા ઉમેરવાનું એક સાદું ઉદાહરણ ચેસની રમતનો પ્રોગ્રામ છે. જે બધા હરીફોની કોઈ ચોક્કસ ચાલની સામે એણે લીધેલાં પગલાં અને તેનાં પરિણામોને સંગ્રહી (યાદ) રાખે છે. એ પછી કોઈક વાર જ્યારે કોઈ હરીફ એવી જ ચાલ રમે તો તેવા સમયે કમ્પ્યુટર તેની અગાઉની બધી ચાલનું વિશ્લેષણ કરે છે અને તેને જિતાડી શકે તેવી ચાલ રમવા તે નિર્ણય લે છે.

કમ્પ્યુટરમાં કૃત્રિમ બુદ્ધિમત્તા (AI)નો સમાવેશ કરવાથી તે માણસની જેમ સમજદારીપૂર્વક વિચારતું થઈ જાય છે. તે કમ્પ્યુટરને માણસની જેમ જ વિચારપૂર્વક વર્તવા માટે તૈયાર કરે છે. વૈજ્ઞાનિકો અને ઈજનેરો હજુ પણ, માણસની સાથે હરીફાઈ કરી શકે તેવાં પૂરતા બુદ્ધિશાળી કમ્પ્યુટર બનાવવા માટેના પ્રયત્નો કરી રહ્યા છે.

ડિજિટલ ફોટોગ્રાફીમાં ઊભરતા પ્રવાહો (Emerging Trends in Digital Photography)

તસવીર ખેંચવાની કળા અથવા 'તસવીરકલા' ઘણાં વર્ષોથી પ્રચલિત થઈ છે. તેની શરૂઆત 1840માં થઈ હતી. જ્યારે પ્રથમ ફોટોગ્રાફિક છાપ મેળવવા વિલિયમ હેનરી ફોક્સ ટાલબોટે પ્રકાશ, કાગળ, થોડાં રસાયણો અને એક લાકડાના ખોખાને એકત્ર કર્યાં. તસવીર બનાવવાની આ પ્રક્રિયા દર વખતે સુધારવામાં આવી. પરંતુ મુખ્ય બદલાવ, ટેકનોલોજિકલ વિકાસ અને નવી શોધખોળોને કારણે આવ્યો. જેવી કે, 'ડિજિટલ કેમેરા'. તેણે તસવીરકલાને રાસાયણિક પ્રક્રિયામાંથી બહાર કાઢી સાંખ્યિક પ્રક્રિયા (digital)માં પ્રવેશ કરાવ્યો.

ડિજિટલ ફોટોગ્રાફી એ પહેલાંની પરંપરાગત ફિલ્મ ફોટોગ્રાફી જેવી જ છે. ડિજિટલ કેમેરા પણ છબી તૈયાર કરવા પ્રકાશનો ઉપયોગ કરે છે. આ બંને વચ્ચેનો એકમાત્ર ફરક એ છે કે, ડિજિટલ કેમેરામાં છબીને સંગ્રહવા માટે પ્લાસ્ટિકની પટ્ટી (ફિલ્મ)ને બદલે પિક્સેલની મેટ્રિક્સ (Matrix of Pixels) સ્વરૂપે સાચવવામાં આવે છે. પિક્સેલની વધુ સંખ્યા વડે છબી વધુ સ્પષ્ટ બને છે અને એટલા માટે જ, ડિજિટલ કેમેરાની વિશેષતાઓમાં આપણને 10 મેગાપિક્સેલ, 12 મેગાપિક્સેલ વગેરે દર્શાવવામાં આવે છે.

તાજેતરમાં કેમેરા સાથેના મોબાઇલ ફોનની શોધ થતાં જ ડિજિટલ કેમેરાના ઉપયોગ દ્વારા ડિજિટલ ફોટોગ્રાફીનું ચિત્ર, બદલાઈ ગયું છે. મોબાઇલ ફોનના શરૂઆતના મોડેલમાં આપણને 2 મેગાપિક્સેલના કેમેરા મળતા હતા. જ્યારે 2012માં Nokia કંપનીએ 41 મેગાપિક્સેલવાળો કેમેરા ધરાવતો મોબાઇલ બહાર પાડ્યો છે. તેના દ્વારા 7728 × 5354 રિઝોલ્યુશનની તસવીર મેળવી શકાય છે. કોઈ પણ યાદગાર અથવા આનંદની પળોની તસવીર ખેંચી તેને થોડી જ ક્ષણોમાં સોશિયલ નેટવર્ક મારફતે લોકો વચ્ચે વહેતી કરવાનું કાર્ય ખરેખર કાબીલેદાદ છે. સ્માર્ટફોનની તસવીર ખેંચવાની સુગમતા ઉપરાંત ઝડપથી તેને લોકો વચ્ચે વહેતી કરવાની ક્ષમતાને કારણે શરૂઆતના ડિજિટલ કેમેરાઓને સ્માર્ટફોન સાથે તીવ્ર હરીફાઈનો સામનો કરવો પડ્યો. આ પ્રશ્નનો સામનો કરવા સેમસંગ કંપનીએ ગેલેક્સી (GALAXY) નામનો એન્ડ્રોઇડ (Android) આધારિત કેમેરા બનાવ્યો. આકૃતિ 13.14માં સેમસંગ ગેલેક્સી કેમેરાના આગળનું અને પાછળનું દૃશ્ય બતાવાયું છે.



આકૃતિ 13.14 : સેમસંગ ગેલેક્સી કેમેરાનું આગળ અને પાછળનું દૃશ્ય

કેમેરામાં ઓપરેટિંગ સિસ્ટમ હોય તેની કલ્પના તો કરી જુઓ. સેમસંગનો એવા દાવો છે કે, આ કેમેરા 3G અને Wi-Fiને સમર્થન આપતો હોવાને કારણે એ સાચા અર્થમાં દુનિયા સાથે જોડાયેલ કેમેરા છે. આપણે ઇન્ટરનેટના માધ્યમથી સોશિયલ નેટવર્ક સાથે જોડાઈ શકીએ, તસવીર અપલોડ કરી શકીએ. વળી, તેમાં ‘ઓટોકલાઉડ બેકઅપ’ (Autocloud Backup) સવલત પણ ઉપલબ્ધ છે, જેના લીધે આપણે જેવી તસવીર ખેંચીએ કે તરત જ તે આપણા કિંમતી ફોટાને આપમેળે ક્લાઉડ (cloud)માં સાચવી દે છે. તેને ‘સ્માર્ટકેમેરો’ પણ કહી શકાય, કારણ કે તે આપણા અવાજને પણ સાંભળે છે. તે આપણા મૌખિક આદેશો સાંભળીને તેના આધારે કાર્ય કરવા સક્ષમ છે.

ઇન્ટરનેટ સમર્થિત ટેલીવિઝન (Internet Enabled Television) :

ગઈકાલના સાદા ફોન આજે ‘સ્માર્ટફોન’ બની ગયા, તેના માટે આપણે ઇન્ટરનેટના આભારી છીએ. ટેલીવિઝનમાં પણ CRTથી LED એમ ધરખમ ટેકનોલોજિકલ બદલાવ આવ્યો છે. આજકાલ હવે આપણને સપાટ સ્માર્ટ (Flat smart) ટેલીવિઝન મળવા માંડ્યાં છે, જે દીવાલ પર ખીંટીએ લટકાવી શકાય છે અને Wi-Fi સમર્થિત હોય છે. આવા ટેલીવિઝન વર્લ્ડવાઇડ વેબ સાથે સીધાં જોડાઈ શકે છે અને YouTubeના વીડિયો, હવામાનના અહેવાલ કે ટેલીવિઝન શો દર્શાવી શકે છે. આ માટે આપણને કમ્પ્યુટર કે અન્ય કોઈ બહારનાં સાધનો કે ઉપકરણોની જરૂર પડતી નથી.

આ ટેકનોલોજી હજુ તેની બાલ્યાવસ્થામાં છે, તેથી ઉપયોગકર્તા દ્વારા જોઈ શકાતી ઇન્ટરનેટ પરની માહિતી જુદા-જુદા ઉત્પાદકોનાં ટીવીમાં જુદી-જુદી હોય છે. ઉદાહરણ તરીકે, Panasonicના ઇન્ટરનેટ સમર્થિત ટેલીવિઝન YouTube, Amazon Video-On-Demand, Picasa Web Albums, Bloomberg News અને હવામાનની ચેનલનો ડેટા દર્શાવી શકે છે. સેમસંગના ઇન્ટરનેટ સમર્થિત ટેલીવિઝન વધારામાં eBay અને Twitterમાંથી માહિતી દર્શાવી શકે છે. LG કંપનીના ઇન્ટરનેટ સમર્થિત ટેલીવિઝનમાં Netflix’sનો ફિલ્મનો વિશાળ સંગ્રહ અને TV shows પણ દર્શાવી શકે છે. કેટલાક ઇન્ટરનેટ સમર્થિત ટેલીવિઝનમાં ઓપરેટિંગ સિસ્ટમ લગાવેલી જ હોય છે અને તમે તારલેસ માધ્યમ અથવા ઇથરનેટ જોડાણ મારફત ટેલીવિઝન સાથે ઇન્ટરનેટ જોડી શકો છો.

ગ્રીન કમ્પ્યૂટિંગ (Green Computing) :

જ્યારે પણ આપણે કમ્પ્યુટર-સાધનોનો ઉપયોગ કરીએ છીએ, ત્યારે આપણે એવાં સંસાધનોનો ઉપયોગ કરવો પડે છે, જે કદાચ પર્યાવરણની સમસ્યા માટે કારણભૂત બને છે. ટેકનોલોજીનો ઉપયોગ હંમેશાં એક સિક્કાની બે બાજુ જેવો હોય છે. ટેકનોલોજીના ઉપયોગથી એક તરફ માનવીના જીવનની ગુણવત્તામાં સુધારો થાય છે, જ્યારે બીજી બાજુ પર્યાવરણને નુકસાન થાય છે. ગ્લોબલ વોર્મિંગ (વૈશ્વિક તાપમાન) એ ટેકનોલોજીના વધુ પડતા ઉપયોગની જ આડઅસર છે. કમ્પ્યુટર સંસાધનોના કાર્યક્ષમ ઉપયોગને રજૂ કરતો એક શબ્દ પ્રચલિત બન્યો છે તે છે, 'ગ્રીન કમ્પ્યૂટિંગ'.

ગ્રીન કમ્પ્યૂટિંગની પહેલનો મુખ્ય ઉદ્દેશ્ય કમ્પ્યૂટિંગ સંસાધનોનો ઉપયોગ કરવાને લીધે ઉદભવતી પર્યાવરણીય અસરોને શક્ય તેટલી ઓછી કરવી અને કમ્પ્યૂટિંગ કામગીરીને આર્થિક રીતે વધુ પોષણક્ષમ બનાવવાનો છે. 1992માં શરૂ કરવામાં આવેલ ENERGY STAR કાર્યક્રમ એ ગ્રીન કમ્પ્યૂટિંગની દિશામાં પહેલું કદમ હતું. આ કાર્યક્રમ અંતર્ગત જે-તે કંપની દ્વારા તૈયાર કરાયેલા એવાં કમ્પ્યૂટિંગ સાધનો કે જેણે ઊર્જાનો વપરાશ ન્યૂનતમ કર્યો અને કાર્યક્ષમતા ઉચ્ચતમ કરી હોય તેને લેબલ એનાયત કર્યાં. આવાં લેબલ કમ્પ્યુટરનાં મોનિટર, ટેલીવિઝન, સેટ અને વાતાવરણને નિયંત્રિત કરતાં સાધનો જેવાં કે, રેફ્રિજરેટર, એરકંડિશનર તેમજ એના જેવી જ અન્ય વસ્તુઓ માટે આ લેબલ મેળવી શકાય.

કમ્પ્યુટરના મોનિટરને અમુક ચોક્કસ પરિસ્થિતિ વખતે સુષુપ્ત અવસ્થા (sleep mode)માં લઈ જવાનું કાર્ય, એ ગ્રીન કમ્પ્યૂટિંગ પહેલનું પ્રથમ પરિણામ હતું. આવી અવસ્થા હવે ઘણાં બધાં સાધનોમાં ઉપલબ્ધ થવા લાગી છે. જ્યારે અમુક પૂર્વનિશ્ચિત સમયગાળા માટે ઉપયોગકર્તા તરફથી કોઈ ક્રિયા થતી માલૂમ ન પડે, તો તેવા સંજોગોમાં આવી સવલત ધરાવતાં ઈલેક્ટ્રોનિક સાધન આપમેળે સુષુપ્ત અવસ્થામાં મુકાઈ જાય છે. સર્વરનું આભાસીપણું એ પણ ગ્રીન કમ્પ્યૂટિંગનો વર્તમાન પ્રવાહ છે. સર્વર તરીકે અનેક કમ્પ્યુટર મશીનનો ઉપયોગ કરવાને બદલે કંપનીઓ આજકાલ વર્ચ્યુઅલાઇઝેશનવાળા ઉચ્ચ ક્ષમતા ધરાવતા એક સર્વરનો ઉપયોગ કરે છે અને તેમાંથી અનેક સર્વર બનાવે છે.

કોઈપણ ઉપકરણમાં ભલે આવી સવલત ના પણ આપી હોય તેમ છતાં આપણે ગ્રીન કમ્પ્યૂટિંગ માટે કેટલાંક પગલાં લઈ શકીએ. જ્યારે થોડા સમય માટે ઉપયોગ ન કરવો હોય, ત્યારે આપણે જાતે જ મોનિટરને બંધ કરી દઈએ અથવા પરંપરાગત CRT પ્રકારના મોનિટરને બદલે LCD કે LED મોનિટરનો ઉપયોગ કરીએ.

સારાંશ

આ પ્રકરણમાં આપણે કમ્પ્યુટર ટેકનોલોજીમાં ઊભરતા પ્રવાહો વિશે શીખ્યા. મોબાઇલ કમ્પ્યૂટિંગ અને ક્લાઉડ કમ્પ્યૂટિંગના નામે ઓળખાતાં બે કમ્પ્યૂટિંગ ક્ષેત્રો વિશે પણ જાણ્યું. ક્લાઉડ કમ્પ્યૂટિંગના ત્રણ આર્કિટેક્ચર SaaS, PaaS અને IaaS વિશે પણ જાણ્યું. આપણે માનવ કે ઓબ્જેક્ટને ઓળખી કાઢવા ઉપયોગમાં લેવાતી RFID, GPS અને બાયોમેટ્રિક ટેકનોલોજી પણ જોઈ. બાયોમેટ્રિકમાંય વળી આપણે આંગળાંની છાપની ઓળખ, આંખની કીકીની ઓળખ અને અવાજની ઓળખ વિશે જાણ્યું. એ પછી આપણે ડેટાસંગ્રહમાં ઊભરતી ટેકનોલોજી વિશે જાણ્યું. જેમાં વિવિધ ઓનલાઇન સંગ્રહ વિકલ્પોની સાથે-સાથે NAS અને SAN નામના બે સંગ્રહ આર્કિટેક્ચરની પણ ચર્ચા કરી. રોબોટિક્સ અને કમ્પ્યુટર અંકુશિત સાધનો આજકાલ સ્વીકૃત છે, આપણે આવી કેટલીક ટેકનોલોજીની ચર્ચા કરી. છેલ્લે, આપણે ડિજિટલ કેમેરા અને ઇન્ટરનેટ સમર્થિત ટેલીવિઝન સ્વરૂપે ઉપલબ્ધ કેટલીક ટેકનોલોજી જોઈ. ગ્રીન કમ્પ્યૂટિંગના છેલ્લા મુદ્દા દ્વારા આપણે ગ્લોબલ વોર્મિંગની સમસ્યા અને નાનાં-નાનાં પગલાં લઈને તેને કેવી રીતે કાબૂમાં લઈ શકાય તે અંગે ચર્ચા કરી.

સ્વાધ્યાય

1. મોબાઈલ કમ્પ્યૂટિંગ માટે સૌથી સામાન્ય ગણાતી ટેકનોલોજીની યાદી આપો.
2. અંગત ક્લાઉડ વિનિયોગ અને ધંધાકીય ક્લાઉડ વિનિયોગો વચ્ચેનો તફાવત સ્પષ્ટ કરો.
3. RFIDના કાર્યને ટૂંકમાં વર્ણવો.
4. ઓબ્જેક્ટની ઓળખમાં બાયોમેટ્રિક ટેકનિકનું મહત્ત્વ શું છે ?
5. આંગળાંની છાપની ઓળખ અને આંખની કીકીની ઓળખ વચ્ચે ઓછામાં ઓછા ત્રણ તફાવત ઓળખી બતાવો.
6. SaaS પદને ટૂંકમાં વર્ણવો.
7. PaaS પદને ટૂંકમાં વર્ણવો.
8. NAS અને SAN વચ્ચેનો તફાવત સ્પષ્ટ કરો.
9. વ્યક્તિ માટે ઓનલાઈન ડેટાસંગ્રહ કેવી રીતે ફાયદાકારક છે ?
10. કમ્પ્યૂટર અંકુશિત સાધનોનો ઉપયોગ થતો હોય, તેવાં કેટલાંક વિનિયોગ ક્ષેત્રોની યાદી આપો.
11. હેતુલક્ષી પ્રશ્નો
 - (1) 'ક્લાઉડ કમ્પ્યૂટિંગ' પદમાં ક્લાઉડ શબ્દ નીચેના પૈકી શેના માટે વપરાયો છે ?
 - (a) તારલેસ
 - (b) હાર્ડડ્રાઈવ
 - (c) લોકો (People)
 - (d) ઈન્ટરનેટ
 - (2) Google Drive (ગૂગલ ડ્રાઈવ) એ નીચેનાં પૈકી કયા પ્રકારના ક્લાઉડનું ઉદાહરણ છે ?
 - (a) ખાનગી ક્લાઉડ
 - (b) જાહેર ક્લાઉડ
 - (c) અંગત ક્લાઉડ
 - (d) ધંધાકીય ક્લાઉડ
 - (3) નીચેનામાંથી કયો ક્લાઉડનો ફાયદો ન ગણાય ?
 - (a) જગ્યા ભરાઈ જવાની ચિંતા રહેતી નથી.
 - (b) ક્લાઉડ નેટવર્કની જાળવણી આસાન છે.
 - (c) કમ્પ્યૂટર સંસાધનોના ત્વરિત ઉપયોગ માટે
 - (d) તમે જે વાપરો તેના માટે જ ચૂકવો છો.
 - (4) નીચેનાં પૈકી કયું ક્લાઉડ કમ્પ્યૂટિંગનું મોડેલ નથી ?
 - (a) PaaS
 - (b) IaaS
 - (c) BaaS
 - (d) SaaS
 - (5) નીચેનાં પૈકી કયું SaaSનું સાચું પૂર્ણ સ્વરૂપ છે ?
 - (a) Software alternative accounting Standard
 - (b) Short alert activation Supplement
 - (c) Software as a Service
 - (d) Service as a Software

- (6) નીચેનાં પૈકી કયું RFIDમાં ઉપયોગમાં લેવાતી માઈક્રોચીપની મહત્તમ સંગ્રહક્ષમતા રજૂ કરે છે?
- (a) 10 બાઈટ્સ (b) 2000 બાઈટ્સ
(c) 1000 બાઈટ્સ (d) 500 બાઈટ્સ
- (7) માત્ર અક્ષાંશ અને રેખાંશની ગણતરી કરવા GPS રિસીવરને કેટલા સેટેલાઈટની જરૂર પડે છે ?
- (a) એક (b) બે
(c) ત્રણ (d) ચાર
- (8) નીચેનાં પૈકી કયો વિકલ્પ નેટવર્ક સાથે સીધા જોડાયેલ સંગ્રહ-સાધનનો નિર્દેશ કરે છે?
- (a) Network Attached Storage
(b) Storage Area Network
(c) Direct Attached Storage
(d) RAID
- (9) નીચેનાં પૈકી કયું સંગ્રહ-સાધનોના નેટવર્કનો નિર્દેશ કરે છે, જેનો નેટવર્ક દ્વારા ઉપયોગ કરી શકાય છે?
- (a) Direct Attached Storage
(b) RAID
(c) Network Attached Storage
(d) Storage Area Network
- (10) નીચેનાં પૈકી કયું પદ, ડાયનેમિક વાતાવરણમાં સેન્સર અને એક્ઝ્યુએટરના ઉપયોગ દ્વારા મશીન કેટલી કાર્યક્ષમતાથી કાર્ય કરી શકે તે શોધી કાઢતા અભ્યાસને રજૂ કરે છે ?
- (a) કાયનેટિક્સ (Kinetics)
(b) જીનેટિક્સ (Genetics)
(c) રોબોટિક્સ (Robotics)
(d) બાયોમેટ્રિક (Biometric)
- (11) નીચેના પૈકી ગ્રીન કમ્પ્યૂટિંગ માટેનાં પગલાં તરીકે 1992માં શરૂ કરવામાં આવેલ કાર્યક્રમ કયો છે ?
- (a) STAR ENERGY
(b) ENERGY STAR
(c) ENERGY SAVE
(d) SAVE ENERGY

પ્રાયોગિક સ્વાધ્યાય

1. કેટલીક ફાઇલો અપલોડ કરવા Google Driveનો ઉપયોગ કરો અને તમે કરી શકો તેવાં વિવિધ કાર્યો કરી જુઓ.
2. ADriveમાં એક ખાતું ખોલો અને Google Driveની સવલતો સાથે તેની સરખામણી કરો.
3. Ubuntu One પર ખાતું બનાવીને તેની સવલત તપાસો.
4. Salesforce.com ની નિઃશુલ્ક મર્યાદિત આવૃત્તિને તપાસો.
5. તમારું એન્ટારોનમેન્ટ જોઈ જાવ અને કેવી રીતે વિવિધ સ્થળે RFIDનો ઉપયોગ કરવામાં આવે છે તે અંગે અહેવાલ લખો.
6. VEDICS (Voice Enabled Desktop Interaction and Control System) ને તમારા કમ્પ્યુટર પર પ્રસ્થાપિત કરીને જુઓ કે મૌખિક આદેશ સાંભળીને કમ્પ્યુટર કેવું કાર્ય કરે છે.





पाठ्यपुस्तकमां समाविष्ट ढीनकसना मूणलूत आदेशो

Command	Usage
type	To check whether a command is internal or external
cal	To display calendar as per users requirement
date	To display current date and system time
bc	To initiate command line calculator
echo	To display message on terminal
passwd	To change the password of the current user
clear	To clear the content on the screen
man	To display the manual of a specified command
whatis	To display small description of a specified command
apropos	To display list of all commands with a specified string
pwd	To display the path of current working directory
mkdir	To create a directory
cd	To change a directory
rmdir	To remove a directory
cat	To create a file or to display contents of file
rm	To delete/remove a file
ls	To see the list of contents in a given directory
cp	To copy a file
mv	To rename a file or directory or to change location of file

more	To show one page of output on screen at a time
cmp	To compare two files
diff	To compare two files and display the contents where the difference lies.
wc	To count characters, words or number of lines
chmod	To change the permissions of file or directory
head	To display lines from the beginning of the file
tail	To display lines from the end of the file
cut	To cut the file vertically
paste	To paste contents of two files together
sort	To arrange the files in ascending or descending order
tr	To translate (convert) strings or patterns from one set of characters to another
grep	To search for strings or patterns within a file
find	To search for file/directory
sudo	To run command as a super user
adduser	To add new user to system
who	To display information of all users currently logged in the system
addgroup	To add user group to the system
deluser	To delete a user from the system
delgroup	To delete a user group form the system
sh	To execute specified shell script
bash	To execute specified shell script
who am i	To display information of the user currently logged in the system
ps	To see the processes associated with the current shell
kill	To kill (remove from main memory) the process with specified id.
tar	To compress (pack) a file/directory

