



ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનો પરિચય

પરિચય

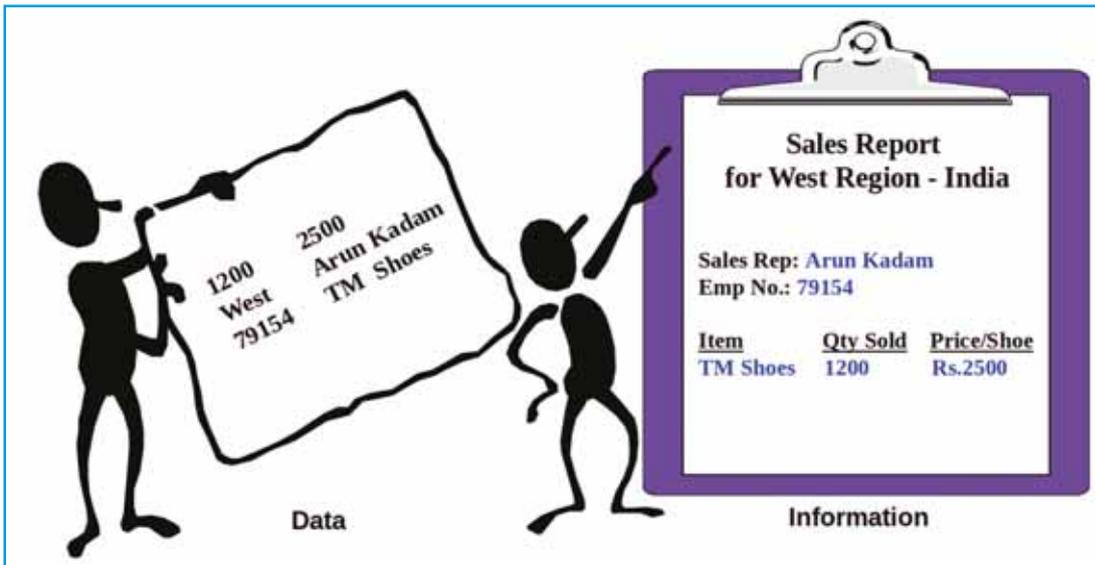
આજના ડિજિટલ યુગમાં, સંસ્થાઓ દર સેકન્ડે મોટા પ્રમાણમાં ડેટા ઉત્પન્ન કરે છે. આ ડેટાને યોગ્ય રીતે સંગ્રહ, પ્રોસેસ અને સુવ્યવસ્થિત (Organize) કરવાની જરૂર પડે છે જેથી તેનો ઉપયોગ ભવિષ્યમાં નિર્ણયો લેવા માટે થઈ શકે. આથી, અસરકારક ડેટા મેનેજમેન્ટ માટે ડેટા અને ઈન્ફોર્મેશન વચ્ચેનો તફાવત સમજવો મહત્વપૂર્ણ બને છે. ડેટાબેઝ એક સંરચિત ભંડાર (Structured Repository) તરીકે કાર્ય કરે છે જે ડેટાને કાર્યક્ષમ રીતે મેળવવા અને ડેટાને મેનીપ્યુલેટ કરવાની સુવિધા આપે છે. આ પ્રકરણમાં, આપણે ડેટા અને ઈન્ફોર્મેશન વચ્ચેનો તફાવત સમજશું. આપણે ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (DBMS) અને તેના ઘટકો વિશે જાણીશું. આપણે એક નમૂનાનો ડેટાબેઝ પણ બનાવીશું અને ડેટાબેઝમાં જે મહત્વનો ભાગ ભજવે છે તે ડેટા ટાઈપ વિશે અભ્યાસ કરીશું.

ડેટા વિરુદ્ધ (vs) ઈન્ફોર્મેશન

ઘણીવાર યુઝર ડેટા અને ઈન્ફોર્મેશન શબ્દોનો સમાનાર્થી તરીકે ઉપયોગ કરે છે. તેથી, આ બંને શબ્દોનો ચોક્કસ અર્થ જાણવો જરૂરી છે. ડેટા અને ઈન્ફોર્મેશનને અનેક રીતે વ્યાખ્યાયિત કરી શકાય છે, તો ચાલો આપણે તે શું છે તે સમજવાથી શરૂઆત કરીએ.

ડેટા એ લોકો, સ્થળો, વસ્તુઓ અથવા ઘટનાઓ સંબંધિત અસંગઠિત હકીકતો, આંકડાઓ અને વિગતો છે. તે મોટાભાગે કાચો (raw) ગણવામાં આવે છે. ડેટા કોઈપણ ફોર્મેટમાં હોઈ શકે છે, જેમ કે સંખ્યાઓ, ટેક્સ્ટ, ઈમેજ અથવા વિડિયો. તે લખેલો કે બોલેલો પણ હોઈ શકે છે. કાચા સ્વરૂપમાં ડેટા કદાચ બહુ ઉપયોગી હોતો નથી. નિર્ણય લેવાની પ્રક્રિયામાં ડેટાના મહત્વને જોતાં, ઘણી કંપનીઓ તેને એક મુખ્ય સંપત્તિ માને છે. ડેટાનો મૂળભૂત ગુણધર્મ એ છે કે તે ઘણીવાર અપ્રસ્તુત અને અસંરચિત (unstructured) હોય છે.

ઈન્ફોર્મેશન એ પ્રોસેસ થયેલો ડેટા છે. જ્યારે કાચા ડેટાને કોઈ પ્રકારની પરિવર્તન પ્રક્રિયામાંથી પસાર કરવામાં આવે છે ત્યારે તે ઈન્ફોર્મેશનમાં રૂપાંતરિત થાય છે. ઈન્ફોર્મેશનનો મૂળભૂત ગુણધર્મ એ છે કે તે સંરચિત (structured), સુસંગત અને અર્થપૂર્ણ હોય છે.



આકૃતિ 1.1 : ડેટા અને ઈન્ફોર્મેશન વચ્ચેનો તફાવત

ચાલો આપણે એક ઉદાહરણ લઈને ડેટા અને ઈન્ફોર્મેશન વચ્ચેનો તફાવત સમજવાનો પ્રયાસ કરીએ. 1200, 2500, 79154, “પશ્ચિમ”, “અરુણ કદમ” અને “TM શૂઝ” જેવી સંખ્યાઓ અને ટેક્સ્ટ માત્ર ડેટા છે. સ્વતંત્ર રીતે, તેમાંથી કોઈ પણ સુસંગત લાગતું નથી. જોકે, તમે તેને યોગ્ય સંદર્ભમાં મૂકો: “શ્રી અરુણ કદમ, કર્મચારી નંબર: 79154, એ પશ્ચિમ ભારતના ક્ષેત્રમાં 2500 રૂપિયાના ભાવે 1200 TM શૂઝનું વેચાણ કર્યું છે”, તો તે ઈન્ફોર્મેશન બની જાય છે, કારણ કે તે અર્થ અને સંદર્ભ પ્રદાન કરે છે. આકૃતિ 1.1 ડેટા અને ઈન્ફોર્મેશન વચ્ચેનો તફાવત દર્શાવે છે.

ડેટાબેઝ (Database) શું છે?

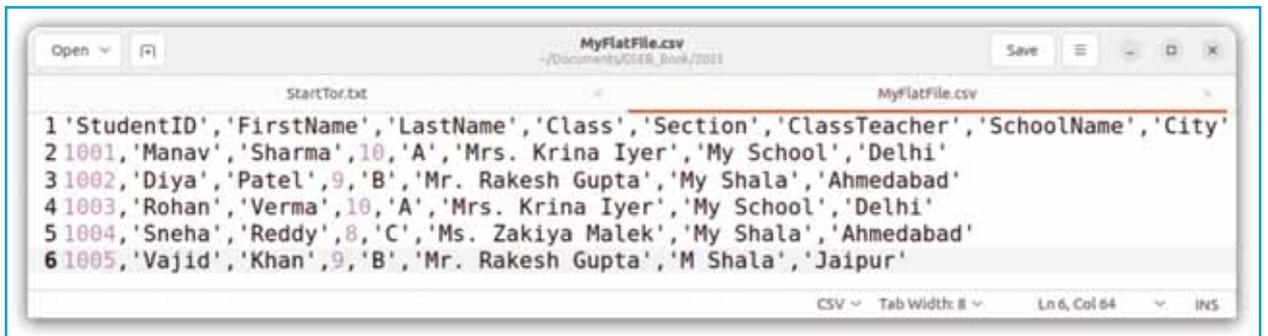
આજે આપણે બધા આપણા રોજિંદા જીવનમાં એક અથવા વધુ પ્રકારના ડેટાબેઝનો ઉપયોગ કરીએ છીએ. ડેટાબેઝનું સૌથી સરળ ઉદાહરણ આપણા મોબાઇલ ફોનમાં રહેલી કોન્ટેક્ટ બુક છે, જેમાં આપણા પરિચિતના નામ, કોન્ટેક્ટ નંબર અને ઈમેલ એડ્રેસ હોય છે. કોઈ વ્યક્તિગત પરિચિતને લગતી વિગતોને રેકોર્ડ કહેવામાં આવે છે.

ડેટાબેઝ એ યોગ્ય રીતે વ્યવસ્થિત સ્ટોર કરેલ સંબંધિત ડેટા આઈટમનો સમૂહ છે. આ વ્યાખ્યા મુજબ, જ્યારે આપણે ડેટાબેઝનો ઉલ્લેખ કરીએ ત્યારે બે મુખ્ય મુદ્દાઓ ધ્યાનમાં લેવા જરૂરી છે. પ્રથમ; તેમાં એકબીજા સાથે સંબંધિત ડેટા આઈટમ હોવી જોઈએ અને બીજું, તે એક વ્યવસ્થિત કલેક્શન હોવું જોઈએ.

ઘટકોની તાર્કિક ગોઠવણ જ્યારે જરૂર પડે ત્યારે ઘટકો શોધવાનું હંમેશા સરળ બનાવે છે. ડેટાબેઝ ઘણીવાર પ્રિ-ડિફાઇન્ડ નિયમોના સમૂહ અને ડેટા મોડેલ અનુસાર ડિઝાઇન કરવામાં આવે છે. ડેટા મોડેલ એ ડેટાને સ્ટોર કરવા અને ફરીથી મેળવવાની રીતનું વર્ણન કરે છે. DBMSમાં વિવિધ પ્રકારના ડેટા મોડેલ છે જેમ કે હાયરાર્કિકલ, નેટવર્ક, રિલેશનલ, ઓબ્જેક્ટ-ઓરિએન્ટેડ, ફ્લેટ, સેમી-સ્ટ્રક્ચર્ડ, એસોસિયેટિવ અને કન્ટેક્ટ. આમાંના બે મોડેલ, ફ્લેટ અને રિલેશનલ, મહત્વનો ભાગ ભજવે છે. ફ્લેટ મોડેલ ડેટા સ્ટોર કરવાનું પ્રારંભિક મોડેલ હતું, જ્યારે રિલેશનલ મોડેલ આજે સૌથી વધુ વ્યાપકપણે ઉપયોગમાં લેવાતું મોડેલ છે.

ફ્લેટ ડેટા મોડેલ (Flat Data Model)

ફ્લેટ ડેટા મોડેલમાં, ડેટા એક જ, દ્વિ-પરિમાણીય (two-dimensional) ટેબલ તરીકે રજૂ થાય છે જેમાં કોઈ વ્યાખ્યાયિત જોડાણો હોતા નથી. બધા રેકોર્ડમાં ફીલ્ડની સંખ્યા અને પ્રકાર સુસંગત હોય છે. તેની સરળતા અને સમજવામાં સરળતાને કારણે, આ ફોર્મેટ નાના અથવા એક વખતના ડેટાસેટ માટે આદર્શ છે. આ ફોર્મેટનો ઉપયોગ વારંવાર કોમા સેપરેટેડ વેલ્યૂઝ (CSV), લિબ્રેઓફિસ કેલ્સી (LibreOffice Calc) અને એમએસ એક્સેલ (MS Excel) ફાઇલોમાં થાય છે. એક CSV ફાઇલ અને લિબ્રેઓફિસ કેલ્સીનો સેમ્પલ ડેટાસેટ આકૃતિ 1.2 અને 1.3માં દર્શાવેલ છે.



આકૃતિ 1.2 : ફ્લેટ ડેટા મોડેલનું ઉદાહરણ (CSV ફાઇલ)

StudentID	FirstName	LastName	Class	Section	ClassTeacher	SchoolName	City
1001	Manav	Sharma	10A		Mrs. Krina Iyer	My School	Delhi
1002	Diya	Patel	9B		Mr. Rakesh Gupta	My Shala	Ahmedabad
1003	Rohan	Verma	10A		Mrs. Krina Iyer	My School	Delhi
1004	Sneha	Reddy	8C		Ms. Zakiya Malek	My Shala	Ahmedabad
1005	Vajid	Khan	9B		Mr. Rakesh Gupta	M Shala	Jaipur

આકૃતિ 1.3 : ફ્લેટ ડેટા મોડેલનું ઉદાહરણ

ફ્લેટ ડેટા મોડેલ ખૂબ જ સરળ અને સમજવામાં સહેલું છે. જોકે, કેટલીક મર્યાદાઓને કારણે તેનો ઉપયોગ જટિલ અથવા મોટા પાયાના ડેટાસેટ માટે થઈ શકતો નથી. ફ્લેટ ડેટા મોડેલની કેટલીક મર્યાદાઓ નીચે મુજબ છે:

ડેટા રિડન્ડન્સી (Data Redundancy) : ફ્લેટ ડેટાબેઝમાં એક જ ડેટા ઘણી વખત રિપીટ થવાની શક્યતાઓ ઘણી વધારે છે. આકૃતિ 1.3માં ક્લાસ ટીચરનું નામ ઘણી વખત દેખાય છે.

અપડેટ એનોમલીઝ (Update Anomalies) : ડેટા ઘણી વખત રિપીટ થતો હોવાથી, જો ઉદાહરણ તરીકે 'My School' નામ અપડેટ કરવાનું હોય તો આપણે આ ડેટા ધરાવતી દરેક રો (હરોળ)માં ફેરફાર કરવો પડશે. જો રોની સંખ્યા ખૂબ વધારે હોય તો ભૂલો થવાનું જોખમ વધી જાય છે.

મર્યાદિત મોડ્યુલારિટી (Limited Modularity) : આકૃતિ 1.3માં જોઈ શકાય છે કે બધો ડેટા એક જ ટેબલમાં સ્ટોર કરેલ છે. આ કારણે ફેરફારોને અલગ કરવા અથવા ડેટા ઘટકોનો જુદા જુદા સંદર્ભમાં ફરીથી ઉપયોગ કરવો ખૂબ મુશ્કેલ બને છે.

નબળી સ્કેલેબિલિટી (Poor Scalability) : 100 રેકોર્ડ્સ જેવી નાની માત્રાના ડેટા માટે, ફ્લેટ ડેટા મોડેલ બરાબર કામ કરે છે. જેમ જેમ ડેટા વધે છે, તેમ તેનું મેનેજમેન્ટ, અપડેટિંગ અને ક્વેરીંગ બિનકાર્યક્ષમ બને છે. સ્ટ્રક્ચરના અભાવને કારણે પર્ફોર્મન્સ સંબંધિત સમસ્યાઓ પણ ઊભી થાય છે.

ડેટા રિલેશનશિપનો અભાવ (No Data Relationships) : ફ્લેટ ડેટા મોડેલમાં વાસ્તવિક દુનિયાના સ્ટ્રક્ચર્સનું પ્રતિનિધિત્વ કરવું મુશ્કેલ છે, કારણ કે તે વિવિધ એન્ટિટીઝ વચ્ચે રિલેશનશિપ ડિફાઈન કરી શકતું નથી. ઉદાહરણ તરીકે, આકૃતિ 1.3 માં વિદ્યાર્થીઓ-શિક્ષકો અથવા ક્લાસ-વિષયો વચ્ચેનો રિલેશન ઓળખવો ખૂબ મુશ્કેલ છે.

ડેટા ઇન્ટિગ્રિટીની સમસ્યાઓ (Data Integrity Issues) : ઓપરેશન દરમિયાન સુસંગતતા જાળવવી મુશ્કેલ બની શકે છે, જેનાથી ડેટા ઇન્ટિગ્રિટીની સમસ્યાઓ ઊભી થાય છે. ઉદાહરણ તરીકે, શાળાનું નામ અપડેટ કરતી વખતે એક ટાઈપોગ્રાફિકલ ભૂલ ધ્યાન બહાર રહી શકે છે.

જટિલ ડેટાને ક્વેરી કરવું મુશ્કેલ (Querying Complex Data is Difficult) : સંસ્થામાં કેટલાક નિર્ણયો લેવા માટે ડેટાબેઝની જરૂર હોય છે. ફ્લેટ ડેટા મોડેલમાં ડેટા વચ્ચે સ્ટ્રક્ચર્ડ લિંક્સના અભાવને કારણે વિશ્લેષણ કરવું અથવા અર્થપૂર્ણ રિલેશનશિપ શોધવી ખૂબ મુશ્કેલ બની જાય છે.

રિલેશનલ ડેટા મોડેલ (Relational Data Model)

રિલેશનલ ડેટા મોડેલ ફ્લેટ ડેટા મોડેલમાં આવતી તમામ સમસ્યાઓનો ઉકેલ લાવે છે. રિલેશનલ ડેટા મોડેલમાં ડેટાને ટેબલના સ્વરૂપમાં ગોઠવવામાં આવે છે. ટેબલ રો અને કોલમથી બનેલા હોય છે. એક ટેબલનો ઉપયોગ ચોક્કસ એન્ટિટીનું પ્રતિનિધિત્વ કરવા માટે થાય છે, અને વિવિધ ટેબલ વચ્ચેના સંબંધોને કી (ટેબલના ફીલ્ડ) નો ઉપયોગ કરીને નિયંત્રિત કરવામાં આવે છે. આથી, આકૃતિ 1.3માં દર્શાવેલ ડેટાબેઝને વિદ્યાર્થી, શિક્ષક, શાળા અને અન્ય એન્ટિટીનું પ્રતિનિધિત્વ કરતા ટેબલમાં વિભાજિત કરી શકાય છે.

રિલેશનલ ડેટા મોડેલ જટિલ ડેટાબેઝને ક્વેરી (પ્રશ્ન) કરવા માટે સ્ટ્રક્ચર્ડ ક્વેરી લેંગ્વેજ (SQL) (જેની વાક્યરચના અંગ્રેજી જેવી છે) નો ઉપયોગ કરે છે. તેની સરળતા, અનુકૂળન ક્ષમતા અને મજબૂત સૈદ્ધાંતિક આધારને કારણે, આ મોડેલનો વ્યાપકપણે ઉપયોગ થાય છે. આપણે આ પ્રકરણના આગામી વિભાગમાં ટેબલ કેવી રીતે બનાવવું તે શીખીશું.

ડેટાબેઝની જરૂરિયાત

આજના સમયમાં વ્યવસાય ડેટા પર આધારિત છે. ડેટા વિના તેમના માટે નવા નવા ઉકેલ અને પ્રોડક્ટ્સ બનાવવાનું અશક્ય છે. તો, ચાલો હવે આપણે જોઈએ કે ડેટાબેઝને આટલો મહત્ત્વપૂર્ણ કોણ બનાવે છે અને તેની શું જરૂરિયાત છે. સંસ્થાની નીચે મુજબની જરૂરિયાતો ડેટાબેઝની આવશ્યકતા પર પ્રકાશ પાડે છે.

ડેટાની ગોઠવણી અને માળખું (Data Organization and Structure)

પહેલા જણાવ્યા મુજબ, ડેટાબેઝ સંસ્થા દ્વારા જનરેટ થતા મોટા પ્રમાણમાં ડેટાને સ્ટ્રક્ચર્ડ ફોર્મેટમાં ગોઠવવા માટે એક વ્યવસ્થિત માર્ગ અથવા અભિગમ પ્રદાન કરે છે. તે સંબંધિત ઈન્ફોર્મેશનના તાર્કિક જૂથ બનાવે છે, જેનાથી કાર્યક્ષમ ડેટા સંગ્રહ થાય છે.

વિદ્યાર્થી, શિક્ષક અને શાળા જેવા અનેક ટેબલ બનાવવાથી કોઈપણ શાળા સંચાલન પ્રવૃત્તિઓ માટે જરૂરી ડેટાને ગોઠવવામાં મદદ મળે છે.

કાર્યક્ષમ ડેટા રીટ્રાઈવલ (Efficient Data Retrieval)

વ્યવસાયોએ તેમની પાસેના ડેટાના આધારે ઝડપી નિર્ણયો લેવાની જરૂર હોય છે. ડેટાને તાર્કિક જૂથમાં સંગ્રહિત કરવામાં આવતો હોવાથી, તે જરૂરી ઈન્ફોર્મેશનને ઝડપથી શોધવા અને રીટ્રાઈવ (પરત મેળવવા) કરવા સક્ષમ બનાવે છે.

ધારો કે આપણે કોઈ વિદ્યાર્થી વિશેની વિગતો શોધવી છે, તો તે વિદ્યાર્થીના ટેબલમાં સ્ટોર કરેલ ડેટામાંથી સરળતાથી મેળવી શકાય છે. જો વધારાની માહિતીની જરૂર હોય, ઉદાહરણ તરીકે વિદ્યાર્થીઓના માર્ક્સ, તો તે માર્ક્સ સ્ટોર કરેલા બીજા સંબંધિત ટેબલના ડેટાનો ઉપયોગ કરીને મેળવી શકાય છે.

ડેટા ઈન્ટેગ્રિટી અને કન્સિસ્ટન્સી (Data Integrity and Consistency)

કોઈપણ વ્યવસાય ડેટા ખોટો હોય તે પસંદ કરશે નહીં. ડેટા પર વેલિડેશન નિયમો લાગુ કરી શકે તેવો ડેટાબેઝ બનાવવાથી વ્યવસાયોને તેમની તમામ કામગીરીમાં સચોટ અને સુસંગત ડેટા જાળવી રાખવામાં મદદ મળે છે.

ડેટાને તાર્કિક અને સ્ટ્રક્ચર્ડ રીતે ગોઠવવામાં આવે છે, જેથી કોઈપણ જરૂરી અપડેટ ફક્ત એક જ જગ્યાએ થશે. આ ફેરફારની અસર પછી સમગ્ર ડેટાસેટમાં પ્રતિબિંબિત થશે. ઉદાહરણ તરીકે, જો આપણે શાળાનું નામ બદલીએ, તો આપણે તેને ફક્ત શાળાના ટેબલમાં જ બદલીશું. અપડેટ થયેલ મૂલ્ય પછી બધા ટેબલમાં પ્રતિબિંબિત થશે.

એકસાથે એક્સેસ (Concurrent Access)

ઘણીવાર ડેટાના એક જ ભાગને અનેક હિસ્સેદારો દ્વારા એક્સેસ (ઉપયોગ) કરવાની જરૂર પડે છે. ડેટાબેઝ ઘણા યુઝર્સને એકસાથે ડેટાને એક્સેસ અને ડેટામાં ફેરફાર કરવાની સવલત પૂરી પાડે છે. આકૃતિ 1.3 ના કિસ્સામાં, ક્લાસ ટીચરના નામને વિદ્યાર્થી તેમજ શાળાના પ્રિન્સિપાલ દ્વારા એક્સેસ કરી શકાય છે.

એક્સેસ નિયંત્રણ (Access Control)

ઘણીવાર ડેટાને વારંવાર એક્સેસ સાથે વહેંચવાની કરવાની જરૂર હોય છે, પરંતુ તે સુરક્ષિત અને પ્રતિબંધિત રીતે થવું જોઈએ. ડેટાબેઝ સંવેદનશીલ ઈન્ફોર્મેશનને સુરક્ષિત રાખવા માટે રોલ-આધારિત પરમિશન અને યુઝર-લેવલ સિક્યુરિટી પ્રદાન કરે છે. આકૃતિ 1.3માં જો આપણે દરેક રોમાં કુલ માર્ક્સનું ફીલ્ડ ઉમેરીએ તો, આપણને વિવિધ પ્રકારના એક્સેસ કંટ્રોલની જરૂર પડશે.

ઉદાહરણ તરીકે, એક શિક્ષક ચોક્કસ ક્લાસના બધા વિદ્યાર્થીઓના માર્ક્સ જાણવા માંગશે, તેથી તેને તે ચોક્કસ ક્લાસના તમામ માર્ક્સનો એક્સેસ આપવો જરૂરી છે. બીજા બાજુ, એક વિદ્યાર્થી ફક્ત તેના અથવા તેણીના માર્ક્સ જ જોઈ શકવો જોઈએ.

ડેટા બેકઅપ અને રિકવરી (Data Backup and Recovery)

કોઈપણ વ્યવસાય માટે વૈકલ્પિક વ્યવસ્થા હોવી અત્યંત આવશ્યક છે. ડેટાબેઝ હોવાથી, ડેટાના નુકસાનને રોકવા માટે ઓટોમેટેડ બેકઅપ માટેની સગવડ ઊભી કરી શકાય છે. આમ, બેકઅપ સમયસર રિકવરી ક્ષમતાને સક્ષમ બનાવે છે.

ધારો કે આપણે છેલ્લા દસ વર્ષના વિદ્યાર્થીઓના પરિણામની હાર્ડકોપી સ્ટોર કરી હતી, પરંતુ કોઈ કારણોસર તે રેકોર્ડ્સ નાશ પામ્યા. આ ડેટા પાછો મેળવવો લગભગ અશક્ય હશે, પરંતુ જો તે જ રેકોર્ડ્સ ડેટાબેઝમાં ઉપલબ્ધ હોય, તો માર્ક્સીટ ફરીથી બનાવવી ખૂબ જ સરળ હશે.

અહેવાલ અને વિશ્લેષણ (Reporting and Analytics)

ડેટાબેઝ હોવાથી, કોઈ વ્યવસાય નિર્ણયો લેવા માટે અહેવાલ અને ડેશબોર્ડ બનાવી શકે છે. તે જટિલ ડેટા વિશ્લેષણ અને બિઝનેસ ઈન્ટેલિજન્સ કાર્યોને સુવિધાજનક બનાવે છે, જેનો ઉપયોગ વ્યવસાયની ક્ષમતાઓને વધારવા માટે થઈ શકે છે.

જો ડેટા ડેટાબેઝનો ભાગ હોય, તો સેંકડો વિદ્યાર્થીઓની માર્ક્સીટ બનાવવી અને પ્રિન્ટ કરવી ખૂબ જ સરળ હશે. ઉપરાંત, વિષયમાં વિદ્યાર્થીઓની નિષ્ફળતા, શ્રેષ્ઠ પ્રદર્શન કરનાર શિક્ષક અને અન્ય જેવા વિશ્લેષણો કરવાનું કાર્ય પણ ખૂબ સરળ બની જશે.

ડેટાબેઝની શા માટે જરૂર છે તેના ઘણા વધુ કારણો વિશે વિચારી શકાય છે, પરંતુ અત્યારે ઉપરોક્ત કારણો પૂરતા છે.

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ શું છે? (What is Database Management System?)

સરળ શબ્દોમાં કહીએ તો, ડેટાબેઝ એ એક સંગ્રહસ્થાન છે. આ સંગ્રહસ્થાન જાતે અથવા કમ્પ્યુટર દ્વારા બનાવી શકાય છે. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (DBMS) એક એવું સૉફ્ટવેર છે જે યુઝર અને ડેટાબેઝ વચ્ચે સેતુ તરીકે કામ કરે છે. તે કમ્પ્યુટરનો ઉપયોગ કરીને ડેટાને અસરકારક રીતે સંગ્રહ કરવા, મેનેજ કરવા અને તેમાં ફેરફાર કરવાની સુવિધા પૂરી પાડે છે.

DBMS એ ડેટાને વ્યવસ્થિત રીતે ગોઠવવા માટે એક કેન્દ્રીય પ્રણાલી તરીકે કાર્ય કરે છે. તે ડેટાની ઈન્ટેગ્રિટી અને કન્સિસ્ટન્સી જાળવવા માટે ટેબલ, રિલેશનશીપ, અને સ્કીમાનો ઉપયોગ કરે છે. DBMS ડેટાને સ્ટોર કરવા, રીટ્રાઈવ કરવા, ફેરફાર કરવા અને કાઢી નાખવા જેવી આવશ્યક સુવિધા SQL જેવી પ્રમાણભૂત ક્વેરી લેંગ્વેજ દ્વારા પ્રદાન કરે છે. તે એક્સેસ અને ક યુઝરને એક્સેસ કરવાની મંજૂરી આપે છે અને સુરક્ષાના પાસાઓ પણ જાળવે છે. તે ટ્રાન્ઝેક્શન મેનેજમેન્ટ, બેકઅપ અને ડેટા રિકવરી જેવા જટિલ કાર્યોનું પણ સંચાલન કરે છે. આ ઉપરાંત, તે ડેટાબેઝના કદને વિસ્તૃત કરીને અને વધુ યુઝરનું સંચાલન કરીને વ્યાપકતા પણ પૂરી પાડે છે.

ઓરેકલ (Oracle), આઈબીએમ ડીબી2 (IBM DB2), માઈક્રોસોફ્ટ એક્સેસ (Microsoft Access) અને માઈક્રોસોફ્ટ એસક્વ્યુએલ સર્વર (Microsoft SQL Server) જાણીતા ખાનગી માલિકીનાં (Proprietary) DBMSના ઉદાહરણો છે. આ ઉપરાંત, ઘણાં ઉત્તમ ઓપન સોર્સ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ પણ ઉપલબ્ધ છે, જેમ કે કાઉચડીબી (CouchDB), મારિયાડીબી (MariaDB), પોસ્ટગ્રેએસક્વ્યુએલ (PostgreSQL), લિબ્રેઓફિસ બેઝ (LibreOffice Base) અને માયએસક્વ્યુએલ (MySQL). આ દરેકની પોતાની વિશિષ્ટતાઓ અને ક્ષમતાઓ છે જે તેમને વિવિધ ઉપયોગો માટે આદર્શ બનાવે છે.

સામાન્ય રીતે, DBMS કેટલાક સોફ્ટવેર એપ્લિકેશનનો એક મહત્વપૂર્ણ ભાગ છે, કારણ કે તે વ્યવસાયોને તેમના રોજિંદા કામકાજ અને નિર્ણય લેવાની પ્રક્રિયાઓ માટે જરૂરી વિશાળ માત્રામાં ડેટાનું સંચાલન કરવા માટે એક મજબૂત, સલામત અને અસરકારક માળખું પૂરું પાડે છે.

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમના ઘટકો (Components of Database Management System)

મોટા ભાગની DBMS સિસ્ટમમાં ડેટાબેઝ સાથે કામ કરવા માટે ચાર મૂળભૂત ઓબ્જેક્ટ્સ અથવા ઘટકો હોય છે, જેમને ટેબલ (Table), ક્વેરી (Query), ફોર્મ (Form) અને રિપોર્ટ (Report) તરીકે ઓળખવામાં આવે છે. આ ઓબ્જેક્ટ કોઈપણ ડેટાબેઝ બનાવવા, ઉપયોગ કરવા અને મેનેજ કરવા માટેના બિલ્ડીંગ બ્લોક છે. ચાલો આ દરેક ઓબ્જેક્ટ વિશે સંક્ષિપ્તમાં જોઈએ.

ટેબલ (Table)

DBMSમાં ડેટા સંગ્રહ માટેનો મૂળભૂત એકમ ટેબલ તરીકે ઓળખાય છે. ટેબલ એ એક દ્વિ-પરિમાણીય માળખું છે, જેમાં ઘણીવાર અનેક રો અને કોલમ હોય છે. કોલમને 'ફીલ્ડ' અને રો ને 'રેકોર્ડ' કહેવામાં આવે છે. ટેબલ સામાન્ય રીતે કોઈ ચોક્કસ એન્ટિટી સાથે સંબંધિત હોય છે, જેમ કે વ્યક્તિ, સ્થળ, એકાઉન્ટ, પરીક્ષા અથવા અન્ય.

	First Name	Last Name	Birth Date	Birth City	Gender
Records	Vidi	Arolkar	12-12-2000	Ahmedabad	Female
	Sunny	Jain	09-11-1999	Jaipur	Male
	Sazid	Khan	10-01-2001	Lucknow	Male
	Tony	Gomes	13-09-2000	Goa	Male

આકૃતિ 1.4 : વ્યક્તિઓના ડેટાને સ્ટોર કરતા ટેબલનો નમૂનો

ચાલો, આકૃતિ 1.4ના પરિપ્રેક્ષ્યમાં એન્ટિટી (entity), ફીલ્ડ (field) અને રેકોર્ડ (record) શબ્દોને સમજાવે.

એન્ટિટી (Entity) : આપણે ટેબલમાં વ્યક્તિ સંબંધિત ડેટાને સ્ટોર કર્યો છે. અહીં, એક વ્યક્તિ એ એક એન્ટિટી બને છે.

ફીલ્ડ (Field) : તે એવા ડેટાના નાના અને અલગ ભાગનું પ્રતિનિધિત્વ કરે છે, જેને એટ્રીબ્યુટ તરીકે ઓળખવામાં આવે છે. સ્વતંત્ર રીતે તેનું કદાચ બહુ મહત્ત્વ ન પણ હોય. અહીં, First Name , Last Name , Birth Date , Birth City અને Gender ને ટેબલના ફીલ્ડ તરીકે ઓળખવામાં આવે છે.

રેકોર્ડ (Record) : એન્ટિટીના એક જ ઉદાહરણ વિશેની વિગતો દર્શાવતા ફીલ્ડના સમૂહને રેકોર્ડ કહેવાય છે

First Name , Last Name , Birth Date , Birth City અને Gender ને સંયુક્ત રીતે એક રેકોર્ડ કહેવાય છે. જે એક વ્યક્તિની વિગતો રજૂ કરે છે.

આકૃતિ 1.4માં આપેલા ટેબલને જોતાં, આપણે કહી શકીએ કે આપણે ચાર અલગ-અલગ વ્યક્તિઓના રેકોર્ડને સ્ટોર કર્યા છે અને દરેક વ્યક્તિના પાંચ એટ્રીબ્યુટ છે.

ફોર્મ (Form)

જરૂરિયાત મુજબ આપણે ટેબલમાં ડેટા ઇન્સર્ટ કરી શકીએ છીએ. એકવાર ડેટા ઇન્સર્ટ થઈ જાય પછી, હાલના રેકોર્ડને એડિટ કરવા, ડીલીટ કરવા અથવા ટેબલમાં ઉપલબ્ધ રેકોર્ડ જોવાની જરૂર પડી શકે છે. ફોર્મ આપણને એક ગ્રાફિકલ યુઝર ઇન્ટરફેસ આપે છે જેને વ્યક્તિગત જરૂરિયાત અથવા પસંદગી મુજબનું બનાવી શકાય છે. ફોર્મમાં ચોક્કસ ડેટા એન્ટ્રી ફોર્મેટ, ડિઝાઇન અથવા લેઆઉટ હોઈ શકે છે જે યુઝરનું કાર્ય સરળ બનાવે છે.

ક્વેરી (Queries)

ટેબલમાં રહેલા ડેટાનો ઉપયોગ યુઝર દ્વારા પૂછાયેલા પ્રશ્નોના જવાબ આપવા માટે થાય છે. ડેટાબેઝમાં પૂછવામાં આવેલા પ્રશ્નને ક્વેરી કહેવામાં આવે છે. “વ્યક્તિઓના ટેબલમાં કેટલા પુરુષો છે?” અથવા “સની જૈનની જન્મ તારીખ શું છે?” જેવા પ્રશ્નોના જવાબ આપવા માટે ક્વેરી બનાવી શકાય છે. ક્વેરીનો ઉપયોગ ડેટાબેઝમાં જુદા જુદા ટેબલમાં સમાવિષ્ટ પસંદગીના ડેટા સેટને દર્શાવવા માટે કરી શકાય છે.

રિપોર્ટ (Reports)

ડેટાબેઝમાં ટેબલ અથવા ક્વેરીનું આઉટપુટ રો અને કોલમના સ્વરૂપમાં દર્શાવવામાં આવે છે. જોકે, યુઝર પૂછેલા પ્રશ્નો માટે વધુ સમજાવટવાળું અને ફોર્મેટ કરેલું આઉટપુટ ઇચ્છે છે જેથી વસ્તુઓને યોગ્ય રીતે સમજી શકાય. રિપોર્ટ એ જરૂરી માહિતીને યોગ્ય રીતે ફોર્મેટ કરેલા, વ્યવસ્થિત અને વાંચી શકાય તેવા સ્વરૂપમાં રજૂ કરે છે.

મેક્રો (Macro)

બેઝનો એક વધારાનો ઘટક મેક્રો (Macro) એ યુઝર માટે ખૂબજ મદદરૂપ છે. મેક્રો એ કમબાંધ કમાન્ડનો સમૂહ છે. જે ડેટાબેઝમાં પુનરાવર્તિત કાર્યોને સ્વયંચાલિત કરે છે. તે એક જ ક્લિકથી ક્રિયાઓની શ્રેણી બનાવી શકે છે, જેમ કે ક્વેરી ચલાવવી, રિપોર્ટ ખોલવા અથવા રેકોર્ડ અપડેટ કરવા. આપણે યુઝર ઇન્ટરફેસ મેક્રો બનાવીને ફોર્મને કન્ટ્રોલ કરવા અથવા ટેબલ ઇવેન્ટ દ્વારા ટ્રિગર થતી ક્રિયાઓને સ્વયંચાલિત કરી શકીએ છીએ. વધુમાં મેક્રોનો ઉપયોગ ચકાસણી કરવા અને નિયમો લાગુ કરવા માટે થઈ શકે છે, જે ડેટા એન્ટ્રીને વધુ વિશ્વસનીય બનાવે છે.

નમૂનારૂપ ડેટાબેઝ બનાવવો (Sample Database Creation)

કોઈપણ એપ્લિકેશન માટે ડેટાબેઝ બનાવવા માટે એક સુવ્યવસ્થિત અભિગમ જરૂરી છે. શરૂઆતમાં, આપણે એપ્લિકેશનની જરૂરિયાતો, તેની પ્રક્રિયાઓ અને વિવિધ પ્રવૃત્તિઓને જોડતા પ્રવાહોને સમજવાની જરૂર છે. આના આધારે, આપણે જરૂરી મુખ્ય એન્ટિટી અને તેમના એટ્રીબ્યુટને ઓળખવા પડશે.

ચાલો, એક રિલેશનલ ડેટા મોડેલનું માળખું બનાવવાનો પ્રયાસ કરીએ જેનો ઉપયોગ શાળાના સંચાલન સંબંધિત પ્રવૃત્તિઓનો ડેટા સ્ટોર કરવા માટે થઈ શકે. શાળાના સંચાલનમાં વિદ્યાર્થીઓ અને શિક્ષકોનું સંચાલન, પરીક્ષાનું સંચાલન, પગારપત્રક અને બીજી ઘણી બધી પ્રવૃત્તિઓનો સમાવેશ થાય છે, પણ આપણે ફક્ત તેના એક નાના ભાગને જોઈશું, જેમાં આપણે વિદ્યાર્થી, શિક્ષક, વિષય, વર્ગ અને ગ્રેડની વિગતો જોઈશું.

હવે આપણે નક્કી કર્યું કે કોનો ડેટા સ્ટોર કરવામાં આવશે. આ બધા શબ્દો, કે જે તમામનો ડેટા સ્ટોર કરવામાં આવશે, તેને એન્ટિટી તરીકે ઓળખવામાં આવે છે. આપણે દરેક એન્ટિટી માટે એક અલગ ટેબલ બનાવીશું. આમ, આ કિસ્સામાં આપણે પાંચ ટેબલ ડિઝાઇન કરીશું:

Student (વિદ્યાર્થી), Teacher (શિક્ષક), Subject (વિષય), Class (વર્ગ) અને Grade (ગ્રેડ).

પછીનું પગલું એ નક્કી કરવાનું છે કે આ દરેક એન્ટિટીમાં કયા એટ્રીબ્યુટ હશે અને તેમાં કઈ પ્રકારની કિંમતો સ્ટોર થશે. આ એટ્રીબ્યુટને ટેબલની અંદર ફીલ્ડ તરીકે દર્શાવવામાં આવશે. આ તમામ એન્ટિટી અને તેમના એટ્રીબ્યુટનું નમૂનારૂપ સ્કીમા ટેબલ 1.1માં દર્શાવેલ છે

એન્ટિટી (Entity)	મુખ્ય એટ્રીબ્યુટ (Key Attributes)
Student	StudentID, FirstName, MiddleName, LastName, BirthDate, Gender, Address ClassID
Teacher	TeacherID, FirstName, MiddleName, LastName, BirthDate, Gender, Address, SubjectID
Subject	SubjectID, SubjectName, ClassID
Class	ClassID, ClassName, TeacherID
Grade	GradeID, StudentID, SubjectID, GradeDate, Marks

ટેબલ 1.1 : શાળા ડેટાબેઝ માટેનો નમૂનારૂપ સ્કીમા

આ સ્કીમામાં શાળાની જરૂરિયાત મુજબ કોઈપણ વધારાની સુવિધાઓ ઉમેરીને સ્કીમાને વિસ્તૃત કરી શકાય છે. ઉદાહરણ તરીકે, આપણે પગારપત્રક (payroll), તાલીમ (training), અથવા ભરતી (recruitment) જેવી વિગતો ઉમેરી શકીએ છીએ.

ચાલો, ટેબલ 1.2, 1.3, 1.4, 1.5 અને 1.6 માં દર્શાવ્યા મુજબ દરેક ટેબલમાં સ્ટોર થનારા એટ્રીબ્યુટની વિગતો સમજવાનો પ્રયાસ કરીએ.

સ્ટુડન્ટ (STUDENT)		
એટ્રીબ્યુટ	વિગત	સ્ટોર થનાર વેલ્યુની ટાઈપ
StudentID	વિદ્યાર્થીનો આઈડી સંગ્રહ કરવા માટે વપરાય છે	સંસ્થાની નીતિ પ્રમાણે ટેક્સ્ટ (text) અથવા સંખ્યા (numeric) હોઈ શકે છે
FirstName	વિદ્યાર્થીનું પહેલું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
MiddleName	વિદ્યાર્થીનું મધ્ય નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
LastName	વિદ્યાર્થીનું છેલ્લું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
BirthDate	વિદ્યાર્થીની જન્મ તારીખ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
Gender	વિદ્યાર્થીની જાતિ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
Address	વિદ્યાર્થીનું સરનામું સ્ટોર કરવા માટે વપરાય છે	આલ્ફા-ન્યુમેરિક વેલ્યુ (જેમાં લખાણ, સંખ્યા તેમજ ખાસ અક્ષરોનો સમાવેશ થઈ શકે છે)
ClassID	વિદ્યાર્થી જે વર્ગમાં અભ્યાસ કરે છે તેનો આઈડી સ્ટોર કરવા માટે વપરાય છે	ક્લાસ ટેબલ (Class table)માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે

ટેબલ 1.2 : સ્ટુડન્ટ ટેબલની વિગતો

ટીચર (TEACHER)		
એટ્રીબ્યુટ	વિગત	સ્ટોર થનાર વેલ્યુની ટાઈપ
TeacherID	શિક્ષકનો આઈડી સ્ટોર કરવા માટે વપરાય છે	સંસ્થાની નીતિ પ્રમાણે ટેક્સ્ટ (text) અથવા સંખ્યા (numeric) હોઈ શકે છે
FirstName	શિક્ષકનું પહેલું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
MiddleName	શિક્ષકનું મધ્ય નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
LastName	શિક્ષકનું છેલ્લું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
BirthDate	શિક્ષકની જન્મ તારીખ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
Gender	શિક્ષકની જાતિ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યુ
Address	શિક્ષકનું સરનામું સ્ટોર કરવા માટે વપરાય છે	આલ્ફા-ન્યુમેરિક વેલ્યુ (જેમાં લખાણ, સંખ્યા તેમજ ખાસ અક્ષરોનો સમાવેશ થઈ શકે છે)
SubjectID	શિક્ષક જે વિષય ભણાવે છે તેનો આઈડી સ્ટોર કરવા માટે વપરાય છે	સબ્જેક્ટ ટેબલ (Subject table)માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે

ટેબલ 1.3 : ટીચર ટેબલની વિગતો

સબ્જેક્ટ (SUBJECT)		
એટ્રીબ્યુટ	વિગત	સ્ટોર થનાર વેલ્યૂની ટાઈપ
SubjectID	વિષયનો આઈડી સ્ટોર કરવા માટે વપરાય છે	સંસ્થાની નીતિ પ્રમાણે ટેક્સ્ટ (text) અથવા સંખ્યા (numeric) હોઈ શકે છે
SubjectName	વિષયનું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યૂ
ClassID	જે ક્લાસમાં વિષય ભણાવાય છે, તેનો આઈડી સ્ટોર કરવા માટે વપરાય છે	ક્લાસ ટેબલ (class table)માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે

ટેબલ 1.4 : સબ્જેક્ટ ટેબલની વિગતો

ક્લાસ (CLASS)		
એટ્રીબ્યુટ	વિગત	સ્ટોર થનાર વેલ્યૂની ટાઈપ
ClassID	ક્લાસ આઈડી સ્ટોર કરવા માટે વપરાય છે	સંસ્થાની નીતિ પ્રમાણે ટેક્સ્ટ (text) અથવા સંખ્યા (numeric) હોઈ શકે છે
ClassName	ક્લાસનું નામ સ્ટોર કરવા માટે વપરાય છે	ટેક્સ્ટ વેલ્યૂ
TeacherID	શિક્ષકનો આઈડી સ્ટોર કરવા માટે વપરાય છે જે ક્લાસનું સંચાલન કરે છે	ટીચર ટેબલ (teacher table)માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે

ટેબલ 1.5 : ક્લાસ ટેબલની વિગતો

ગ્રેડ (GRADE)		
એટ્રીબ્યુટ	વિગત	સ્ટોર થનાર વેલ્યૂની ટાઈપ
GradeID	ગ્રેડ આઈડી સ્ટોર કરવા માટે વપરાય છે	ઈન્ટીજર નંબર
StudentID	જે વિદ્યાર્થીને ગ્રેડ આપવામાં આવ્યો છે, તેનો આઈડી સ્ટોર કરવા માટે વપરાય છે	સ્ટુડન્ટ ટેબલ (student table)માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે
SubjectID	જે વિષય માટે વિદ્યાર્થીને ગ્રેડ આપવામાં આવ્યો છે, તેનો આઈડી સ્ટોર કરવા માટે વપરાય છે	સબ્જેક્ટ ટેબલ (subject table) માં સ્ટોર થનારા ડેટાની ટાઈપ પર આધાર રાખશે
GradeDate	વિદ્યાર્થીને ગ્રેડ કઈ તારીખે આપવામાં આવ્યો તે સ્ટોર કરવા માટે વપરાય છે	ડેટ વેલ્યૂ
Marks	માર્ક્સ સ્ટોર કરવા માટે વપરાય છે	ન્યુમેરિક વેલ્યૂ

ટેબલ 1.6 : ગ્રેડ ટેબલની વિગતો

આગળના પ્રકરણમાં જ્યારે આપણે લિબ્રેઓફિસ બેઝ (LibreOffice Base)નો ઉપયોગ કરીને ડેટાબેઝ બનાવીશું, ત્યારે આપણે આ ટેબલનો ઉપયોગ કરીશું. આ તબક્કે, ચાલો આપણે રિલેશનલ ડેટા મોડેલમાં કી (Keys)ના ખ્યાલ વિશે વાત કરીએ. અગાઉ જણાવ્યા મુજબ, રિલેશનલ ડેટા મોડેલમાં ડેટાને રો અને કોલમથી બનેલા ટેબલ (રિલેશનશિપ)માં ગોઠવવામાં આવે છે. કી આપણને આ રિલેશનશિપને યોગ્ય રીતે મેન્ટેઈન

કરવામાં મદદ કરે છે. ચાર કી; પ્રાથમિક (primary), ફોરેન (foreign), કેન્ડિડેટ (candidate) અને અલ્ટરનેટ (alternate) કી મહત્વપૂર્ણ ભૂમિકા ભજવે છે.

પ્રાથમિક કી (Primary Key)

એવું ફીલ્ડ જે ટેબલમાં આવેલી રોને અનન્ય (યુનિક) રીતે ઓળખે છે, તેને પ્રાથમિક કી કહેવામાં આવે છે. પ્રાથમિક કી એક અથવા વધુ કોલમની બનેલી હોઈ શકે છે જેમાં વિશિષ્ટ વેલ્યુ હોય. ટેબલમાં એક કરતાં વધારે રો માટે પ્રાથમિક કીની વેલ્યુ સમાન હોઈ શકે નહીં, અને તે ફીલ્ડની વેલ્યુ ખાલી (empty) પણ છોડી શકાય નહીં. ઉદાહરણ તરીકે, સ્ટુડન્ટ ટેબલમાં દરેક વિદ્યાર્થી પાસે એક વિશિષ્ટ StudentID હોય છે, જેને પ્રાથમિક કી તરીકે ગણી શકાય. જો, કોઈ ટેબલમાં આપણે રેકોર્ડને ઓળખવા માટે એક કરતાં વધુ ફીલ્ડનો ઉપયોગ કરીએ, તો ફીલ્ડ્સના આ સંયુક્ત સમૂહને કોમ્પોઝિટ કી (Composite Key) તરીકે ઓળખવામાં આવે છે.

ફોરેન કી (Foreign Key)

ફોરેન કી એ ટેબલમાં આવેલું એવું ફીલ્ડ છે જેનો ઉપયોગ બીજા ટેબલમાંના રેકોર્ડને યુનિક રીતે ઓળખવા માટે કરી શકાય છે, પછી ભલે તે એકલું હોય કે અન્ય ફીલ્ડ સાથે સંયોજનમાં હોય. આ ફોરેન કી બે ટેબલ વચ્ચે રિલેશનશિપ સ્થાપિત કરવામાં મદદ કરે છે. ઉદાહરણ તરીકે, સ્ટુડન્ટ ટેબલમાં આવેલું ClassID ફીલ્ડ ફોરેન કી તરીકે કાર્ય કરે છે; તે આપણને જણાવે છે કે વિદ્યાર્થી કયા ક્લાસમાં અભ્યાસ કરે છે. તે ક્લાસ ટેબલમાં યુનિક રેકોર્ડને પણ ઓળખે છે. આમ, આપણે કહી શકીએ કે ClassID કી Student અને Class ટેબલને જોડે છે.

કેન્ડિડેટ કી (Candidate Key)

ટેબલ માટેની કેન્ડિડેટ કી એ તમામ ફીલ્ડ માટેની વેલ્યુ છે જે પ્રાથમિક કી બનવા માટે યોગ્ય છે. આવા ફીલ્ડ્સમાં કોઈ ડુપ્લિકેટ વેલ્યુ હોવી જોઈએ નહીં, અને તેમને ખાલી છોડી શકાય નહીં. તેથી, ક્લાસ ટેબલમાં ClassID અને ClassName બંને સંભવિત કેન્ડિડેટ કી છે.

અલ્ટરનેટ કી (Alternate Key)

કેન્ડિડેટ કીમાંથી એક અથવા બે કીને ટેબલ માટે પ્રાથમિક કી તરીકે પસંદ કરવામાં આવે છે. જે કેન્ડિડેટ કી બાકી રહી જાય છે તેને અલ્ટરનેટ કી તરીકે ઓળખવામાં આવે છે. તેથી, જો ClassIDને પ્રાથમિક કી તરીકે ઉપયોગમાં લેવામાં આવે, તો ક્લાસ ટેબલમાં ClassName એ અલ્ટરનેટ કી છે.

ડેટા અને ડેટા ટાઈપ (Data and Data type)

આપણે ટેબલ 1.2માં પહેલાથી જ જોયું છે કે જુદા જુદા ફીલ્ડમાં સ્ટોર થતો ડેટા અલગ અલગ હોઈ શકે છે. ઉદાહરણ તરીકે, આપણા ડેટા સેટમાં FirstName, LastName અને Gender ફીલ્ડમાં ટેક્સ્ટ ડેટા છે, જ્યારે BirthDate ફીલ્ડમાં ડેટ છે. આપણે Address નામના ફીલ્ડના કિસ્સામાં આલ્ફાન્યુમેરિક ડેટા અને Age નામના ફીલ્ડમાં ન્યુમેરિક ડેટા અને Percentage નામના ફીલ્ડમાં દશાંશ સંખ્યા (decimal number)નો પણ ઉપયોગ કરી શકીએ છીએ.

આમ, ડેટા ટાઈપ એ ચોક્કસ ફીલ્ડમાં સ્ટોર થનારા ડેટાના પ્રકારનો ઉલ્લેખ કરે છે. ડેટા ટાઈપ એ પણ નક્કી કરે છે કે ચોક્કસ ફીલ્ડને કેટલી મેમરી ફાળવવામાં આવશે. એક જ ડેટા ટાઈપની અંદર પણ મેમરીની સાર્થક અલગ અલગ હોઈ શકે છે. મોટા ભાગના DBMS અમુક સામાન્ય ડેટા ટાઈપને સપોર્ટ કરે છે જેમ કે ટેક્સ્ટ (text), ન્યુમેરિક (numeric), કરન્સી (currency), ડેટ/ટાઈમ (date/time), બૂલીયન (boolean) અને બાઈનરી (binary). આ વિભાગમાં આપણે લિબ્રેઓફીસ બેઝની ડેટા ટાઈપની ચર્ચા કરીશું.

ટેક્સ્ટ ડેટા ટાઈપ (Text Data Type)

ટેક્સ્ટ ડેટા ટાઈપ આપણને ફીલ્ડમાં ડેટા વેલ્યુ તરીકે આલ્ફાબેટ્સ (મૂળાક્ષરો), સંખ્યાઓ અથવા ખાસ અક્ષરો (spe-

cial characters) ના સંયોજનને સ્ટોર કરવાની મંજૂરી આપે છે. આવા ડેટા પર આપણે કોઈ ગાણિતિક ગણતરીઓ કરી શકતા નથી. જે ફીલ્ડ ટેક્સ્ટ ડેટા ટાઈપનો ઉપયોગ કરી શકે છે તેના કેટલાક ઉદાહરણો નીચે પ્રમાણે છે.

AADHAAR Card Number, FirstName, Delivery Address અને Email. ટેક્સ્ટ ડેટા ટાઈપના ચાર અલગ અલગ પ્રકારો છે, અને તે મુખ્યત્વે સ્પેસનો વપરાશ કેવી રીતે કરે છે તે પ્રમાણે અલગ પડે છે. ટેબલ 1.7 માં લિબ્રેઓફીસ બેઝમાં ઉપયોગમાં લઈ શકાય તેવી વિવિધ ટેક્સ્ટ ડેટા ટાઈપની યાદી દર્શાવવામાં આવી છે.

ટાઈપ	બેઝમાં ફીલ્ડ ડેટા ટાઈપ	મેમરીમાં સ્ટોરેજ
ટેક્સ્ટ	VARCHAR	વેરીએબલ
ટેક્સ્ટ	VARCHAR_IGNORECASE	વેરીએબલ
ટેક્સ્ટ (ફિક્સ)	CHAR	ફિક્સ
મેમો	LONGVARCHAR	વેરીએબલ

ટેબલ 1.7 : ટેક્સ્ટ ડેટા ટાઈપ

VARCHAR : આ શબ્દ વેરીએબલ કેરેક્ટર્સનું પ્રતિનિધિત્વ કરે છે. જ્યારે VARCHARનો ઉપયોગ કરવામાં આવે છે, ત્યારે ફીલ્ડમાં એન્ટર કરી શકાય તેવા કેરેક્ટર્સની મહત્તમ સંખ્યાને વ્યાખ્યાયિત કરવી શક્ય છે. ઉદાહરણ તરીકે, જો કોઈ ફીલ્ડની ડેટા ટાઈપ VARCHAR(30) હોય તો આપણે તેમાં મહત્તમ 30 કેરેક્ટર્સ ડેટા તરીકે એન્ટર કરી શકીએ છીએ. જો યુઝર 30 કેરેક્ટર્સ કરતાં ઓછો ડેટા એન્ટર કરે તો જરૂરી સ્પેસ DBMS દ્વારા આપોઆપ ફાળવવામાં આવશે.

VARCHAR_IGNORECASE : આ VARCHAR જેવું જ છે, પરંતુ તે VARCHARનું કેસ ઈન્-સેન્સિટિવ વર્ઝન છે. અહીં ઈગ્નોર-કેસ સર્ચ કરતી વખતે મહત્વપૂર્ણ ભૂમિકા ભજવે છે. ધારો કે યુઝર city નામના ફીલ્ડમાં "AHMEDABAD" ડેટા સ્ટોર કર્યો છે, હવે સર્ચ કરતી વખતે જો યુઝર "Ahmedabad" શોધે તો પણ તેને આઉટપુટ મળશે.

CHAR : આ ડેટા ટાઈપ એક નિશ્ચિત કદના ટેક્સ્ટ ફીલ્ડનો ઉલ્લેખ કરે છે. VARCHAR ડેટા ટાઈપની જેમ જ, ફીલ્ડની સાઈઝ ફીલ્ડ બનાવતી વખતે સેટ કરવામાં આવે છે. જો ફીલ્ડમાં એન્ટર કરાયેલ ટેક્સ્ટ તમામ જગ્યાનો ઉપયોગ ન કરે, તો બાકીના કેરેક્ટરને સ્પેસ વડે ભરી દેવામાં આવે છે. આ ડેટા ટાઈપનો ઉપયોગ ત્યારે શ્રેષ્ઠ છે જ્યારે આપણે PAN અથવા ક્રેડિટ કાર્ડ નંબરની જેમ પહેલેથી નક્કી હોય તેટલી સંખ્યામાં કેરેક્ટરની જરૂર હોય.

LONGVARCHAR : આ ડેટા ટાઈપ ખૂબ મોટા ટેક્સ્ટ બ્લોકને સ્ટોર કરવા માટે રચાયેલ છે. તે યુઝર દ્વારા નિર્દેશિત કેરેક્ટરની મહત્તમ લંબાઈ સુધીનો ડેટા સ્ટોર કરે છે. તેનો ઉપયોગ સામાન્ય રીતે 255 કેરેક્ટરથી વધુ ડેટા સ્ટોર કરવા માટે થાય છે. જેનું સારું ઉદાહરણ એક લેખ છે, જે 64000 કેરેક્ટર સુધીનો હોઈ શકે છે.

ન્યુમેરિક ડેટા ટાઈપ (Numeric Data Type)

આ ડેટા ટાઈપનો ઉપયોગ સંખ્યાત્મક સ્વરૂપના ડેટાને સ્ટોર કરવા માટે થાય છે. ન્યુમેરિક ડેટા પૂર્ણાંક સંખ્યાઓ અથવા દશાંશ સંખ્યાઓના સ્વરૂપમાં હોઈ શકે છે. પૂર્ણાંક અને દશાંશ સંખ્યાઓ બંને સાઈન અથવા અનસાઈન હોઈ શકે છે. આવા ડેટા પર ગાણિતિક ક્રિયાઓ કરવી શક્ય છે. જે ફીલ્ડ્સ ન્યુમેરિક ડેટા ટાઈપનો ઉપયોગ કરી શકે છે તેના કેટલાક ઉદાહરણો Marks, Salary, Interest અને TotalPayment છે. લિબ્રેઓફીસ બેઝમાં ચાર ઈન્ટિજર ડેટા ટાઈપ છે જે સામાન્ય રીતે સાઈઝમાં અલગ હોય છે, અને ચાર ફ્લોટિંગ-પોઈન્ટ ડેટા ટાઈપ છે જે

ચોક્કસાઈના સ્તરમાં અલગ હોય છે. ટેબલ 1.8 વિવિધ ન્યુમેરિક ડેટા ટાઈપની યાદી દર્શાવે છે, તેની સાથે તે ઉપયોગમાં લેતા બિટ્સ/બાઈટ્સ ની સંખ્યા અને તેની રેન્જ પણ દર્શાવે છે.

ટાઈપ	બેઝમાં ફીલ્ડ ડેટા ટાઈપ	વેલ્યૂની રેન્જ	મેમરીમાં જરૂરી બાઈટ્સ
ટાઈની ઇન્ટિજર	TINYINT	0 to 255 જો અનસાઈન હોય તો -128 to +127 જો સાઈન હોય તો	1 બાઈટ
સ્મોલ ઇન્ટિજર	SMALLINT	0 to 65535 જો અનસાઈન હોય તો -32768 to +32767 જો સાઈન હોય તો	2 બાઈટ
ઇન્ટિજર	INTEGER	0 to 4294967295 -2147483648 to +2147483647	4 બાઈટ
બીગઇન્ટ	BIGINT	0 to 18446744073709551615	8 બાઈટ
નંબર	NUMERIC	અમર્યાદિત (Unlimited)	વેરીએબલ
ડેસીમલ	DECIMAL	અમર્યાદિત (Unlimited)	વેરીએબલ
ફ્લોટ	FLOAT	$5 \times 10^{(-234)}$ to $1.79 \times 10^{(308)}$	4 બાઈટ
રીઅલ	REAL	ચોક્કસ નહીં (Not exact)	4 બાઈટ
ડબલ	DOUBLE	મહત્તમ 15 દશાંશ સુધી કરી શકાય	8 બાઈટ

ટેબલ 1.8 : ન્યુમેરિક ડેટા ટાઈપ

TINYINT : તે ઇન્ટિજર ડેટા ટાઈપમાં સૌથી નાનું છે. તેનું કદ એક બાઈટ છે, તેથી તે મેમરીમાં 8 બિટ જેટલી જગ્યા રોકે છે.

SMALLINT : તે TINYINTના કદ કરતાં બમણું છે. તેનું કદ 2 બાઈટ્સ છે, તેથી તે મેમરીમાં 16 બિટ જેટલી જગ્યા રોકે છે.

INTEGER અથવા INT : સૌથી વધુ ઉપયોગમાં લેવાતું ન્યુમેરિક ડેટા ટાઈપ INTEGER છે. તેનું કદ 4 બાઈટ્સ છે, તેથી તે મેમરીમાં 32 બિટ જેટલી જગ્યા રોકે છે.

BIGINT : મોટા ભાગના સરળ પ્રોગ્રામમાં તેની જરૂર પડતી નથી, તેથી તેનો ભાગ્યે જ ઉપયોગ થાય છે. તેનું કદ 8 બાઈટ્સ છે, તેથી તે મેમરીમાં 64 બિટ જેટલી જગ્યા રોકે છે.

ફ્લોટિંગ-પોઇન્ટ નંબર્સ દશાંશવાળી સંખ્યાઓ અથવા વાસ્તવિક સંખ્યાઓ છે. તે દશાંશ બિંદુ દ્વારા વિભાજિત સંપૂર્ણ ભાગ અને આંશિક ભાગથી બનેલા હોય છે.

DECIMAL અને NUMERIC : આ ડેટા ટાઈપ અમર્યાદિત રેન્જ ધરાવે છે. તેમને ડિફાઈન કરતી વખતે, આપણે દશાંશ બિંદુ પછી આવનારા ડિજિટ્સની સંખ્યા સાથે, કુલ ડિજિટ્સની સંખ્યાનો ઉલ્લેખ કરવાની જરૂર રહે છે. ઉદાહરણ તરીકે, DECIMAL (10, 2)નો અર્થ એ થશે કે ફીલ્ડમાં મહત્તમ 10 સ્થાન અને દશાંશ બિંદુ પછી બે ડિજિટ્સ હશે. DECIMAL અને NUMERICની ચોક્કસાઈ લગભગ પૂર્ણ હોય છે.

DOUBLE અથવા REAL : આ ડેટા ટાઈપ એક મર્યાદિત રેન્જ ધરાવે છે અને તેમાં મહત્તમ 15 દશાંશ સ્થાનો હોઈ શકે છે. આ ડેટા ટાઈપની ચોક્કસાઈ એટલી સારી હોતી નથી.

કરન્સી ડેટા ટાઈપ (Currency Data Type)

કરન્સી ડેટા ટાઈપ આપણને નાણાકીય મૂલ્યો દર્શાવતી ન્યુમેરિક વેલ્યુ સ્ટોર કરવાની સગવડ આપે છે. આપણે વિવિધ દેશોના ચલણનો ઉપયોગ કરીને ડેટા સ્ટોર કરી શકીએ છીએ. ઉદાહરણ તરીકે, ₹ 1250.50, \$ 1500 અથવા £ 700. આ ડેટા ટાઈપ ખાતરી કરે છે કે કરન્સીને લગતી ગણતરીઓ સચોટ છે અને નાણાકીય વેલ્યુનું ડિસ્પ્લે યોગ્ય રીતે ફોર્મેટ થયેલું છે.

ડેટ/ટાઈમ ડેટા ટાઈપ (Date/Time Data Type)

ડેટ/ટાઈમ ડેટા ટાઈપનો ઉપયોગ વર્ષ, મહિનો, દિવસ, કલાક, મિનિટ, સેકન્ડ અને સેકન્ડના અપૂર્ણાંક જેવી માહિતીને સ્ટોર કરવા માટે થાય છે. જે ફીલ્ડ ડેટ/ટાઈમ ડેટા ટાઈપનો ઉપયોગ કરી શકે છે તેના કેટલાક ઉદાહરણો Date of Joining, Birth Date, In Time અને Out Time છે. લિબ્રેઓફીસ બેઝ માં ડેટ/ટાઈમ ડેટા ટાઈપમાં ત્રણ પ્રકારો છે, જે સ્ટોર કરેલા કન્ટેન્ટ પ્રમાણે અલગ પડે છે. ટેબલ 1.9 વિવિધ ડેટ/ટાઈમ ડેટા ટાઈપ દર્શાવે છે.

ટાઈપ	બેઝમાં ફીલ્ડ ડેટા ટાઈપ	વેલ્યુની રેન્જ	મેમરીમાં જરૂરી બાઈટ્સ
ડેટ	DATE	-	4 બાઈટ
ટાઈમ	TIME	-	4 બાઈટ
ડેટ/ટાઈમ	TIMESTAMP	એડજસ્ટેબલ (0.5 – 5 with milliseconds)	8 બાઈટ

ટેબલ 1.9 : ડેટ/ટાઈમ ડેટા ટાઈપ

DATE : તે સિસ્ટમની તારીખ સ્ટોર કરે છે. તારીખ એન્ટર કરવા માટેનું ફોર્મેટ YYYY-MM-DD છે. ઉદાહરણ તરીકે, 2025-05-30

TIME : તે ઘડિયાળનો સમય સ્ટોર કરે છે. સમય માટેનું ડિફોલ્ટ ફોર્મેટ 24 કલાકનું હોય છે, જેમ કે HH:MM:SS (કલાક:મિનિટ:સેકન્ડ), એટલે કે 3:30:36 PM (બપોરે 3 વાગીને 30 મિનિટ અને 36 સેકન્ડ) માટે 15:30:36.

TIMESTAMP અથવા DATETIME : તારીખ અને સમય બંનેનું સંયોજન છે. અહીં ડેટાનું ડિફોલ્ટ ફોર્મેટ YYYY-MM-DD HH:MM:SS છે, એટલે કે 2025-05-30 15:30:36. આ ડેટા ટાઈપનો ઉપયોગ Time of Transaction (ટાઈમ ઓફ ટ્રાન્ઝેક્શન) જેવા ઓડિટ ફીલ્ડમાં થાય છે.

બૂલીયન ડેટા ટાઈપ (Boolean Data Type)

બૂલીયન ડેટા ટાઈપ આપણને ફક્ત બે વેલ્યુ True અથવા False સ્ટોર કરવાની સવલત આપે છે. આ બે વેલ્યુ Yes/No અને On/Off જેવા અનેક ફોર્મેટમાં પણ રજૂ કરી શકાય છે.

બાઈનરી ડેટા ટાઈપ (Binary Data Type) :

બાઈનરી ડેટા આપણને ડિજિટાઈઝ્ડ ઇમેજ, વીડિયો, અવાજ અથવા ફાઈલ જેવી માહિતી સ્ટોર કરવાની મંજૂરી આપે છે. આ તમામ માહિતી શૂન્ય (0) અને એક (1)ની લાંબી સ્ટ્રિંગ તરીકે સ્ટોર થાય છે. તેના ઉદાહરણો Profile Picture અને Voice Sample છે. ટેબલ 1.10 વિવિધ બાઈનરી ડેટા ટાઈપની સૂચિ દર્શાવે છે.

ટાઈપ	બેઝમાં ફીલ્ડ ડેટા ટાઈપ	વેલ્યૂની રેન્જ	મેમરીમાં જરૂરી બાઈટ્સ
બાઈનરી ફીલ્ડ (ફિક્સ)	BINARY	ઈન્ટિજર જેટલી	ફિક્સ
બાઈનરી ફીલ્ડ	VARBINARY	ઈન્ટિજર જેટલી	વેરીએબલ
ઈમેજ	LONGVARBINARY	ઈન્ટિજર જેટલી	વેરીએબલ - ખૂબ મોટી ઈમેજ માટે ઉપયોગી.

ટેબલ 1.10 : બાઈનરી ડેટા ટાઈપ

સારાંશ

આ પ્રકરણમાં આપણે ડેટા અને માહિતી વચ્ચેનો તફાવત સમજાવ્યો. ડેટામાં કાચી હકીકતો હોય છે, જ્યારે માહિતી એ ડેટાનું વ્યવસ્થિત સ્વરૂપ છે. આગળ, આપણે ડેટાબેઝમાં ડેટા મોડેલ વિશે જોયું, જે દર્શાવે છે કે ડેટા કેવી રીતે સ્ટોર થાય છે અને રીટ્રાઈવ કરવામાં આવે છે. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (DBMS) વિવિધ ડેટા મોડેલનો ઉપયોગ કરે છે, જેમાં ફ્લેટ, હાયરાર્કિકલ અને રિલેશનલ મોડેલનો સમાવેશ થાય છે. રિલેશનલ મોડેલ ડેટાબેઝની અંદરના ટેબલને જોડે છે. આપણે ડેટાબેઝ અને DBMSના ખ્યાલો પણ શીખ્યા. ડેટાબેઝ એ ડેટાનો એક વ્યવસ્થિત સંગ્રહ છે અને DBMS આ ડેટાને એડ કરવા, અપડેટ કરવા અને એક્સેસ કરવામાં મદદ કરે છે. આપણે જાણ્યું કે વાસ્તવિક દુનિયાની વસ્તુઓને એન્ટિટી કહેવામાં આવે છે, અને તેમની ચોક્કસ વિગતો એટ્રીબ્યુટ તરીકે ઓળખાય છે. ટેબલ એ એન્ટિટીનું પ્રતિનિધિત્વ છે અને તેમાં રેકોર્ડ હોય છે જે રો અને કોલમના સ્વરૂપમાં ગોઠવાયેલા હોય છે, જેમાં સૌથી નાનો એકમ ફીલ્ડ છે જે એટ્રીબ્યુટનું પ્રતિનિધિત્વ કરે છે. ફીલ્ડમાં સ્ટોર કરેલ ડેટા સંખ્યાઓ, અક્ષરો અથવા ખાસ અક્ષરો હોઈ શકે છે. આ રીતે, એક રેકોર્ડ એ ચોક્કસ એન્ટિટી માટે ડેટા વેલ્યૂનો સંપૂર્ણ સમૂહ છે. પ્રાઈમરી કી એક રોને યુનિક રીતે ઓળખે છે, અને ફોરેન કી વિવિધ ટેબલને જોડે છે. કેન્ડિડેટ કી એ ટેબલ માટે સંભવિત મુખ્ય કી હોય છે. આ પ્રકરણે આવનારા પ્રકરણોમાં ડેટાબેઝના વધુ ખ્યાલો શીખવા માટે એક મજબૂત પાયો નાખ્યો છે.

સ્વાધ્યાય

1. ડેટા અને માહિતીને યોગ્ય ઉદાહરણો સાથે સમજાવો.
2. ડેટાબેઝ અને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ વચ્ચેનો તફાવત સ્પષ્ટ કરો.
3. ટેબલ, રેકોર્ડ અને ફીલ્ડની વ્યાખ્યા આપો.
4. ડેટાબેઝની જરૂરિયાત શા માટે છે તે સમજાવો.
5. ડેટાબેઝના સામાન્ય ઘટકો પર નોંધ લખો.
6. ડેટા ટાઈપ શું છે? બેઝમાં ઉપલબ્ધ ડેટા ટાઈપની યાદી બનાવો.
7. CHAR અને VARCHAR વચ્ચે શું તફાવત છે?
8. કરન્સી ડેટા ટાઈપનો ઉપયોગ શું છે?
9. ક્યારે બૂલીયન ડેટા ટાઈપનો ઉપયોગ કરવો જોઈએ?
10. STUDENT એન્ટિટીના કોઈપણ પાંચ એટ્રીબ્યુટ ઓળખો.
11. સાચું કે ખોટું જણાવો.

1) ડેટાબેઝ એ એક સોફ્ટવેર છે જે આપણને ડેટામાં ફેરફાર કરવાની મંજૂરી આપે છે.



- 2) જો આપણે બસ વિશે ડેટા સ્ટોર કરીએ તો તેને એક એન્ટિટી ગણી શકાય.
- 3) ફોર્મ આપણને ટેબલમાંથી ડેટા ડીલીટ કરવાની મંજૂરી આપે છે.
- 4) ટેક્સ્ટ ડેટા ટાઇપમાં ન્યુમેરિક મૂલ્ય સ્ટોર કરવું શક્ય છે.
- 5) ઓડિયો ક્લિપને ટાઇમ ડેટા ટાઇપમાં સ્ટોર કરી શકાય છે.

12. ખાલી જગ્યા પૂરો.

- (1) ડેટાબેઝ એ સંબંધિત ડેટા આઈટમનો _____ છે.
- (2) એન્ટિટીના એટ્રિબ્યુટસને ટેબલમાં _____ તરીકે ઓળખવામાં આવે છે.
- (3) ટેબલમાં એન્ટિટીની વિગતો આપતા _____ નો સમૂહ હોય છે.
- (4) BIGINT મેમરી સ્ટોરેજ માટે _____ બાઈટ્સનો ઉપયોગ કરે છે.
- (5) ડિજિટાઇઝ્ડ ઇમેજ્સ _____ ડેટા ટાઇપ માં સ્ટોર કરી શકાય છે.

13. બહુવિકલ્પી પ્રશ્નો. સૌથી યોગ્ય જવાબ પસંદ કરો.

- (1) નીચેનામાંથી કયું DBMS નું પૂરું નામ છે?
 - (a) ડેટા બ્લોક મેનેજમેન્ટ સિસ્ટમ (Data Block Management System)
 - (b) ડેટા બેઝ મેનેજમેન્ટ સિસ્ટમ (Data Base Management System)
 - (c) ડેટા બાઈટ મેનેજમેન્ટ સિસ્ટમ (Data Byte Management System)
 - (d) ડેટા બિટ મેનેજમેન્ટ સિસ્ટમ (Data Bit Management System)
- (2) નીચેનામાંથી કયો શબ્દ પ્રોસેસ કરેલા ડેટાનો ઉલ્લેખ કરે છે?
 - (a) ઇન્ફોર્મેશન
 - (b) રો
 - (c) ફેક્ટ્સ
 - (d) ફીલ્ડ
- (3) નીચેનામાંથી કયું ડેટા મોડેલ નથી?
 - (a) ફ્લેટ (Flat)
 - (b) રીલેશનલ (Relational)
 - (c) રોટેશનલ (Rotational)
 - (d) A અને B બંન્ને
- (4) નીચેનામાંથી કયું ડેટાબેઝમાં StudentNameનું શ્રેષ્ઠ વર્ણન કરે છે?
 - (a) રિલેશનશિપ
 - (b) એટ્રિબ્યુટ
 - (c) એન્ટિટી
 - (d) ડેટા
- (5) નીચેનામાંથી કયું ડેટાબેઝ નથી?
 - (a) MySQL
 - (b) લિબ્રેઓફ્સિસ બેઝ
 - (c) લિબ્રેઓફ્સિસ ઇમ્પ્રેસ
 - (d) SQL સર્વર
- (6) નીચેનામાંથી કયું ડેટાબેઝનું સામાન્ય ઘટક નથી?
 - (a) ચાર્ટ
 - (b) ટેબલ
 - (c) ક્વેરીઝ
 - (d) ફોર્મ
- (7) બેઝ ડેટાબેઝમાં ઇમેજ સ્ટોર કરવા માટે નીચેનામાંથી કઈ ડેટા ટાઇપ નો ઉપયોગ થાય છે?
 - (a) ટેક્સ્ટ (Text)
 - (b) બાઈનરી (Binary)
 - (c) બૂલીયન (Boolean)
 - (d) બાઈટ (Byte)

- (8) નીચેનામાંથી કયા સ્વરૂપમાં ડેટા વેલ્યૂ રજૂ કરી શકાતી નથી?
- (a) ટેક્સ્ટ (Text) (b) ન્યુમેરિક (Numeric)
(c) આલ્ફાન્યુમેરિક (Alphanumeric) (d) પિક્ચર (Picture)
- (9) નીચેનામાંથી કયું આપણને સરળ અને યુઝર ફ્રેન્ડલી રીતે ટેબલમાં ડેટા એન્ટર કરવાની મંજૂરી આપે છે?
- (a) રિપોર્ટ (b) ક્વેરી (c) ફોર્મ (d) ફીલ્ડ
- (10) નીચેનામાંથી કોનો ઉપયોગ સંખ્યાઓ સ્ટોર કરવા માટે કરી શકાતો નથી?
- (a) TINYINT (b) DOUBLE (c) DATE (d) CHAR

પ્રાયોગિક સ્વાધ્યાય

1. નીચેની એન્ટિટીઝ માટે તેના એટ્રિબ્યુટ્સ ઓળખીને ડેટા સ્ટોર કરવા માટે યોગ્ય ડેટા ટાઈપ નક્કી કરો.
- બુક (Book)
 - કસ્ટમર (Customer)
 - ડીપાર્ટમેન્ટ (Department)
 - એમ્પ્લોયી (Employee)
 - ફ્લાઈટ (Flight)
 - મેગેઝીન (Magazine)
 - મુવી (Movie)
 - પ્રોડક્ટ (Product)
 - સેલ્સમેન (Salesman)
 - ટ્રેન (Train)
 - વીહિકલ (Vehicle)

