એકમ–3 ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ DATABASE MANAGEMENT SYSTEMS

: માળખું :

3.0 ઉદ્દેશ
3.1 પ્રસ્તાવના
3.2 વ્યાખ્યા અને મૂળભૂત અભિગમ
3.2.1 ડેટા અને માહિતી
3.2.2 ડેટાબેઝ અને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ DBMS
3.2.3 ડેટા હાયરાર્કિ
3.2.4 ડેટા અખંડિતતા
3.2.5 ડેટા સ્વતંત્રતા
3.3 ડેટા મેનેજમેન્ટ સિસ્ટમનો હેતુ DBMS
3.3.1 DBMSની જરૂરિયાત
3.3.2 DBMSનો ધ્યેય
3.4 DBMS નો વિકાસ
3.4.1 કાલક્રમાનુસાર
3.4.2 DBMS ના કાર્ય અને ઘટકો
3.5 DBMS નું માળખું (સ્ટ્રકચર)
3.6 ડેટા મોડેલીંગ
3.6.1 એન્ટીટી - રીલેશનશીપ મોડેલ (અસ્તિત્વ સંબંધિત મોડેલ)
3.6.2 ટાઈપ ઓફ રીલેશનશીપ (સંબંધના પ્રકાર)
3.7 ડેટા મોડેલ્સ
3.7.1 હાયરાર્કિ મોડેલ
3.7.2 નેટવર્ક મોડેલ
3.7.3 રીલેશનલ મોડેલ (સંબંધિત મોડેલ)
3.7.4 ઓબ્જેક્ટ ઓરિએન્ટેડ મોડેલ (વસ્તુલક્ષી મોડેલ)
3.8 રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ RDBMS
3.8.1 સંબંધની લાક્ષણિકતાઓ
3.8.2 કી અને તેનું કાર્ય
3.8.3 ક્રાઈટેરીયા ફોર એ DBMS ટુ બી રીલેશનલ - સંબંધિત થવા માટે DBMS નો માનદંડ
3.9 નોર્મલાઈઝેશન ઓફ રીલેશન્સ - (સંબંધને સામાન્ય બનાવવા)
3.9.1 ડિપેન્ડન્સીસ (આધારિત)
3.9.2 નોર્મલ ફોર્મસ (સહજ આકાર)
3.10 ડિઝાઈનીંગ ડેટાબેઝીસ
3.11 ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ સિસ્ટમ : વિભાજિત ડેટાબેઝ સિસ્ટમ
3.11.1 આર્કિટેક્ચર ઓફ ડીસ્ટ્રીબ્યુટેડ ડેટાબેઝીસ - વિભાજિત ડેટાબેઝનું માળખુ

3.11.2 જસ્ટીફિકેશન એન્ડ ઓપશન્સ ફોર ડિસ્ટ્રીબ્યુટેડ ડેટાને વિભાજિત કરવા માટેનું સમર્થન અને વિકલ્પ

3.12 ડેટાબેઝ સિસ્ટમ ફોર મેનેજમેન્ટ સપોર્ટ - મેનેજમેન્ટ સપોર્ટ માટે ડેટાબેઝ સિસ્ટમ

3.13 આર્ટીફિશયલ ઈન્ટેલીજન્સ એન્ડ એક્સપર્ટ સિસ્ટમ બનાવટી સમજશક્તિ અને નિષ્ણાંત સિસ્ટમ

3.14 સારાંશ

3.15 તમારી પ્રગતિ ચકાસોના ઉત્તરો

3.16 ચાવીરૂપ શબ્દો

3.17 સંદર્ભો અને વિશેષ વાચન

3.0 ઉદ્દેશ (OBJECTIVES)

◆ **આ પ્રકરણને વાંચ્યા પછી તમે કરી શકશો -**

- ડેટાબેઝના અભિગમના વિકાસના ઉદ્‌ભવની પ્રક્રિયા સમજી શકશો.
- ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (DBMS)ની જરૂરિયાતને તેના પ્રાથમિક હેતુ, ડેટાબેઝ આર્કીટેક્ચર અને જટિલ ડેટાબેઝ મેનેજમેન્ટના પર્યાવરણને સમજી શકશો.
- વિવિધ ડેટા મોડેલનો તફાવત અને દરેકની વિશિષ્ટતા જાણી શકશો.
- રેશનલ ડેટાબેઝ ટેકનોલોજી સાથે માહિતગાર થઈ શકશો.
- ડેટા મેનેજમેન્ટના ઉભરતા માપનું વલણ સમજી શકશો.

3.1 પ્રસ્તાવના (INTRODUCTION)

ડેટાબેઝ સિસ્ટમે જીવનના બધા જ કાર્યક્ષેત્રોને વ્યાપકરૂપ આપ્યું છે. આ પ્રકરણ ડેટાબેઝ મેનેજમેન્ટને સમજવા માટે નક્કર માહિતી પુરી પાડે છે. ડેટાબેઝ અભિગમની જરૂરિયાતથી માંડીને, ત્રણ-લેવલ ડેટાબેઝ આર્કીટેકચરને સમજાવ્યું છે. મોડેલીંગ અભિગમ અને E. R મોડેલીંગમાં ભજવેલી ભૂમિકા ડેટાબેઝ ડિઝાઈનની વિચારધારાની ચર્ચા કરી છે. ક્લાસીકલ ડેટા મોડેલ દા.ત. હાયરાર્કિ મોડેલ, નેટવર્ક મોડેલ અને રીલેશનલ મોડેલને ઉદાહરણ સહિત સમજાવ્યા છે. રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ વ્યાપક પ્રમાણમાં ઉપયોગમાં છે. તેને ધ્યાનમાં રાખીને સમગ્રતયા આવરી લેવાયા છે, આધારિત અને સાનૂકુળ અભિગમ ઉદાહરણ સાથે બતાવવામાં આવ્યા છે. ડેટાબેઝ સિસ્ટમને આ પ્રકરણ યથાયોગ્ય વિસ્તારમાં ચોક્કસ વણી લે છે.

3.2 વ્યાખ્યા અને મૂળભૂત અભિગમ (DEFINITIONS AND BASIC CONCEPTS)

ડેટાબેઝમાં સામાન્યપણે વપરાતા શબ્દપ્રયોગ ડેટા- આર્કીટેકચરના અભિગમને સમજવા માટે, આ પછીના પેરેગ્રાફમાં સ્પષ્ટ કરવામાં આવ્યા છે.

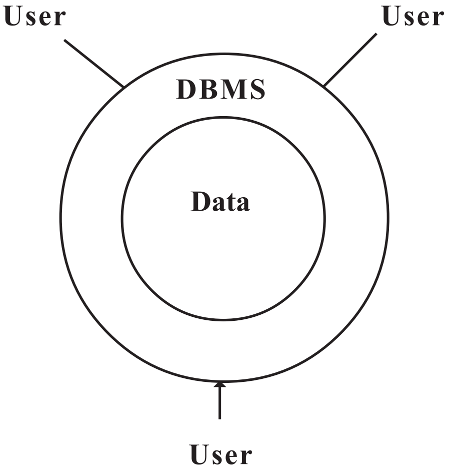
3.2.1 ડેટા અને માહિતી (Data and Information)

કોઈપણ પદાર્થના છેલ્લા તબક્કા સુધી માહિતી મેળવવા માટેનું આ પુરોગામી સાધન છે. ડેટા એ અંકોનો સેટ છે કે જેને પોતાના સ્વતંત્ર અર્થ નથી પણ તેનું પ્રતિનિધિત્વ કરે છે. એટલે કે દા.ત. તે ફક્ત ચિન્હ્ ધરાવે છે, પ્રક્રિયા દ્વારા ડેટા સાથે સંકળાયેલો અર્થ કે જે માહિતીમાં રૂપાંતરિત થાય છે.

આ બન્ને શબ્દપ્રયોગના તફાવતને ઉદાહરણ આપી સ્પષ્ટ કરીએ. અંક 050643 નો કોઈ અર્થ નથી. પણ આપણને એમ કહેવામાં આવે કે પહેલાં બે અંક એક મહિનાનું પ્રતિનિધિત્વ કરે છે, બીજા બે અંક મહિનાને અને છેલ્લા બે અંક વર્ષનો સંકેત આપે છે, તો અંકનો સમુચ્ચય 050643 કોઈ એક વ્યક્તિની જન્મતારીખ બતાવે છે. બીજી રીતે કહીએ તો એ જ આંક 643050 તરીકે લખાય ત્યારે તે કોઈ એક વ્યક્તિનો ટેલિફોન નંબર સૂચવે છે.

3.2.2 ડેટાબેઝ અને ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ : (DBMS) (Database and Database Management System)

- **ડેટાબેઝ :** ડેટાબેઝ એ તાર્કિક સંબંધિત જે ડિઝાઈનમાંથી સ્ટ્રક્ચર ફોર્મમાં મલ્ટીપલ ઉપભોક્તા માટે આવશ્યક માહિતી મેળવવા ગોઠવાયેલા હોય, તેને આવશ્યક ઓપરેશનલ ડેટા (નોન રીડન્ડન્ટ - ઓપરેશનલ ડેટા) કે જે જુદા જુદા વિચારાધિન સિસ્ટમોની ભાગીદારી કરે છે તેનું સંગઠન કહી શકાય.
- **ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ :** એ સોફ્ટવેરનો સંગ્રહ છે કે જેનો ઉપયોગ સ્ટોર, સુધારણા અને ડેટાની ડેટાબેઝમાં સ્ટોર કરવામાં આવેલા ડેટાને પુનઃપ્રાપ્તી માટે ઉપયોગ કરવામાં આવે છે. (DBMS) એ ઉપભોક્તા અને ડેટા વચ્ચે સાંકળનું કામ કરે છે. (આકૃતિ 3.1)



આકૃતિ 3.1 ડેટા (DBMS) અને ઉપભોક્તા

3.2.3 ડેટા હાયરર્કિ :

ડેટાના સંગઠનમાં ઉતરતા ક્રમમાં જટિલ હાયરર્કિ આકૃતિ ૩.૨માં દર્શાવી છે.

Database
↓
File
↓
Record
↓
Field
↓
character
↓
Byte
↓
Bit

આકૃતિ 3.2 : હાયરર્કિ

આ ચડતા ઉતરતા ક્રમથી સ્પષ્ટ થાય છે કે ડેટાબેઝ એ ફાઈલનું બનેલુ છે. ફાઈલ એ રેકોર્ડનો સંગ્રહ છે અને દરેક રેકોર્ડ ફિલ્ડ અથવા ડેટા આઈટમ ધરાવે છે. દરેક ફિલ્ડ અંકોનું બનેલુ છે જે બાઈટ દ્વારા રચાયેલ છે. અને છેલ્લે કહીએ તો બાઈટ બીટસમાં વિભાજિત થયેલ છે.

3.2.4 ડેટાની પ્રમાણિકતા (Data Integrity)

ડેટાબેઝમાં સ્ટોર કરવામાં આવેલા ડેટાની સત્યતાની માત્રા, સમયપાલન અને ડેટાનું નિરૂપણ એ ડેટાબેઝની પ્રામાણિકતા દર્શાવે છે. ડેટા-પ્રામાણિકતા કેટલાક પરિક્ષણ અને સદ્ધરતા માટેની રીતો અમલમાં મૂકીને અપડેટ સંચાલનનો પ્રયાસ કરવામાં આવે છે અને તે ઉપરાંત અનાવશ્યક ડેટાને નાબૂદ કરીને ડેટાની પ્રમાણિકતાને નિશ્ચિત કરવામાં આવે છે, કેટલીક ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ પાસે વિગતો હોય છે, કે જે ડેટાની સત્યતાને પૃષ્ટિ આપે છે. દા.ત. (ORACLE) માં (અ રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ) આ કાર્ય માટે બંદૂકના નાળચાનો ઉપયોગ કરવામાં આવે છે.

3.2.5 ડેટાની સ્વતંત્રતા (Data Independence)

ડેટાની સ્વતંત્રતાએ - ડેટાફાઈલને એપ્લીકેશન પ્રોગ્રામથી તે ફાઈલ વાપરનારથી અલગ રાખે છે. ડેટા અને એકસેસ પ્રોગ્રામ વચ્ચેના ઘનિષ્ટ સંબંધ કમજોર બને છે. ડેટાબેઝ ઉપભોક્તાની જરૂરિયાત માટે વધારે લવચીક બને છે. ડેટાની સ્વતંત્રતા સાથે FORTRAN, PASCAL માં પ્રોગ્રામર, પ્રોગ્રામ લખી શકે છે. મૂળ કઈ ભાષામાં પ્રોગ્રામ રચવામાં આવ્યો છે, તેને ગણત્રીમાં લીધા સિવાય તેને મેળવવા માટે જરૂર હોય તે ફાઈલ લખી શકે છે ડેટાની સ્વતંત્રતા, લોજિકલ, ફિઝીકલ કે ભૌગોલિક હોઇ શકે છે.

લોજીકલ અથવા ફિઝીકલ ડેટા - સ્વતંત્રતા એટલે કે ડેટાના લોજીકલ અથવા ફિઝીકલ માળખાને તેના પ્રોગ્રામીંગ કે જે ડેટાનો દોરી સંચાર કરે છે. તેમાં ફેરફાર કર્યા સિવાય બદલવાની ક્ષમતા ધરાવે છે. ભૌગોલિક ડેટા સ્વતંત્રતા એ વિભાજીત ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમની લાક્ષણિકતા, ઉપભોક્તાને ડેટાનું સ્થળ સ્પષ્ટ કરે છે. ડેટા સ્વતંત્રતા એ અગત્યનો અભિગમ છે, કે જે ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમના આર્કીટિકચરની ચર્ચા કરતી વખતે વધુ વિગતો માટે ધ્યાનમાં લેવામાં આવશે.

◆ **તમારી પ્રગતિ ચકાસો**

1. ડેટા સ્વતંત્રતા એટલે શું ? ડેટાબેઝ સિસ્ટમમાં તેનું પ્રદાન શું ?

નોંધ (i) તમારા જવાબો નીચે આપેલી જગ્યામાં લખો.

(ii) તમારા જવાબો પ્રકરણને અંતે આપેલા ઉત્તરો સાથે સરખાવો.

....
....
....
....
....
....

3.3 ડેટાબેઝ મેનેજમેન્ટ પદ્ધતિનો હેતુ (DBMS) (OBJECTIVES OF DATABASE MANAGEMENT SYSTEMS)

3.3.1 (DBMS) માટે જરૂરિયાત (The Need For DBMS)

ડેટાબેઝ મેનેજમેન્ટની રચના અગાઉ ડેટા પ્રોસેસીંગ માટેની પ્રક્રિયા કેવી હતી તે પરિસ્થિતિનું અવલોકન કરીએ. તે વખતે વપરાશમાં આવતી ફાઈલ, ઓરીએન્ટેડ સિસ્ટમ ફાઈલમાં (ફાઈલ જેમાં વિષયવાર અપટુડેટ ડેટા સચવાયેલા હોય.) પ્રોગ્રામીંગ ભાષાનો ઉપયોગ કરીને માસ્ટર ફાઈલ બનાવવામાં આવે છે. તે ડેટા ઉપર જરૂરી પ્રશ્નોત્તરના આધારે તેની રચનાના સમયે એક્સેસ-ટેક્નિકનું આરોપણ કરવામાં આવે છે.માસ્ટર ફાઈલમાંનો કોઇપણ ફેરફાર દા.ત. ફિલ્ડનો વધારો અથવા ચાલુ ફિલ્ડના સ્ટ્રકચરમાં ફેરફારને સુધારા- વધારા સાથે એક્સેસ ટેક્નિકનો અમલ કરવો પડે છે.

માસ્ટર ફાઈલના ઉપયોગ માટેની ડેટા મેનેજમેન્ટ માટેની મર્યાદાનું ઉદાહરણ આપીએ તો, વ્યાવસાયિક અભ્યાસક્રમો ચલાવતી શૈક્ષણિક સંસ્થાના ડેટાબેઝ, આપણે ધારી લઈએ કે ડેટાબેઝ ત્રણ માસ્ટર ફાઈલ રાખે છે. વિદ્યાર્થી, ફેકલ્ટી અને કોર્સીસનો રેકોર્ડ. વિદ્યાર્થીની માસ્ટર ફાઈલ FORTRAN નો ઉપયોગ કરીને બનાવવામાં આવી છે, અને તેમાં ફિલ્ડ જેવાં કે વિદ્યાર્થીનો આઈડેન્ટીફીકેશન નંબર, નામ, સરનામુ, જાતિ, અભ્યાસક્રમ પાસ કરેલી પરીક્ષાનો હાઈસ્કૂલ ગ્રેડ.

ફેકલ્ટી માસ્ટર ફાઈલ COBO2નો ઉપયોગ કરે છે. અને ફિલ્ડ જેવા કે ફેકલ્ટી આઈડેન્ટીફીકેશન નંબર, નામ, જાતિ, ડિપાર્ટમેન્ટ, પગાર, અભ્યાસ અને શિક્ષણના કલાકો. અભ્યાસક્રમની માસ્ટર ફાઈલ PASCAL આધારિત હોય છે અને ડેટાને આ પ્રમાણે આવરી લે છે. દા.ત. કોર્સ આઈડેન્ટીફિકેશન નંબર, કોર્સનું ટાઈટલ, ક્લાસ નંબર, વિભાગ નંબર અને કોર્સમાં દાખલ વિદ્યાર્થીઓ.

હવે ધારો કે મહિલા ફેકલ્ટી દ્વારા અભ્યાસ કરાવતા મહિલા વિદ્યાર્થિનીઓના નામ માટે કોઈએ માહિતી માંગી, તો ડેટાબેઝમાં આ બધી માહિતી હોવા છતાં, માસ્ટર ફાઈલમાંથી તેનો જવાબ મળી શકશે નહીં. કારણ કે માંગવામાં આવેલી માહિતી - ડેટા - બે માસ્ટર ફાઈલમાં જુદી પ્રોગ્રામીંગ ભાષામાં તેમની પોતાની એક્સેસ ટેક્નિકમાં પડેલી છે. આનો જવાબ આપવા માટે વિદ્યાર્થી અને ફેકલ્ટીની માસ્ટર ફાઈલના ડેટાની મદદ લઈને એક નવી માસ્ટર ફાઈલ નવા પ્રોગ્રામ સાથે બનાવવી પડે. આ પ્રક્રિયા ડેટાનું આરોપણ કંટાળાજનક અને સમયનો વ્યય કરે છે, બીજી પરિસ્થિતિને લઈએ - આ ઉદાહરણમાં જ્યારે સંસ્થાના એકાઉન્ટ ડિપાર્ટમેન્ટને આ ડેટાબેઝનો ઉપયોગ કરવો હોય અને વધારાના ફિલ્ડ દા.ત. સ્ટાઈપંડ પેઈડ, બાકી ફી પેન્લટી ચાર્જીસ વગેરે વધારાની માહિતી સાથેની માસ્ટર ફાઈલની જરૂર પડે છે. આ જરૂરિયાતને મેળવવા માટે નવા ફિલ્ડ બનાવેલી એક વધારાની નવી સ્ટુડન્ટ માસ્ટર ફાઈલની નકલ કરવી પડે. એ જ પ્રમાણે ચોક્કસ પ્રકારની જરૂરિયાત માટે ફેકલ્ટી અને કોર્સની માસ્ટર ફાઈલની નકલ કરવી પડે. આ પ્રક્રિયામાં ડેટાનું બેવડાપણું થાય છે એટલે કે ડેટાની અનાવશ્યકતા આવા સંજોગોમાં અપડેટેડ માસ્ટર ફાઈલો બનાવવાનું અઘરું પડે છે.

આ મર્યાદા અને અસુવિધા ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમના વિકાસનું હાર્દ છે.

3.3.2 (DBMS) નું લક્ષ્ય (The Goals of DBMS)

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનું લક્ષ્ય નીચે પ્રમાણે છે.

(i) લવચીક આરોપણ પૂરું પાડવું, વિવિધ ફાઈલમાંથી ડેટાને પ્રમાણમાં સહેલાઈથી ડેટાલીંક કરી શકાય.

(ii) માસ્ટર ફાઈલમાંથી અનેકાનેક નકલ કરવામાંથી છૂટકારો. ડેટાનું ડુપ્લીકેશન અને માસ્ટર ફાઈલમાંથી બિન જરૂરી ડેટા ઉપરનો અંકુશ અપડેટીંગ પ્રશ્નોને સુધારીને, ડેટા પ્રામાણિકતાનું પ્રયોજન કરવામાં મદદ કરે છે.

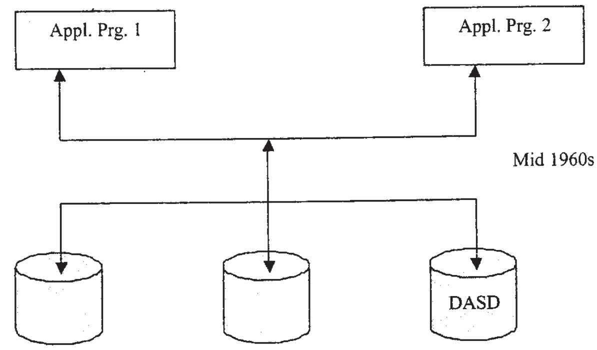
(iii) ડેટાની સ્વતંત્રતાની ઉચ્ચ કક્ષાઓ ચોક્કસ કરવી. ડેટા, પ્રોગ્રામીંગ ભાષા, ઓપરેટીંગ સિસ્ટમ અને પર્યાવરણની પ્રક્રિયામાંથી છૂંપુ હોય છે. સ્ટોર કરેલા ડેટાને પ્રોગ્રામર જે કોઈ ભાષામાં ઉપયોગ કરવા માંગતા હોય, તેને બદલવા માટેનો આધાર (DBMS) પર રહે છે.

3.4 ડેટાબેઝનો વિકાસ (EVOLUTION OF DBMS)

વર્ષો દરમિયાન ડેટાબેઝ મેનેજમેન્ટના અભિગમમાં બહુ જ મોટો ફેરફાર થયો છે. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનો વિકાસ ક્રમાનુસાર નીચે પ્રમાણે વર્ણવવામાં આવ્યો છે.

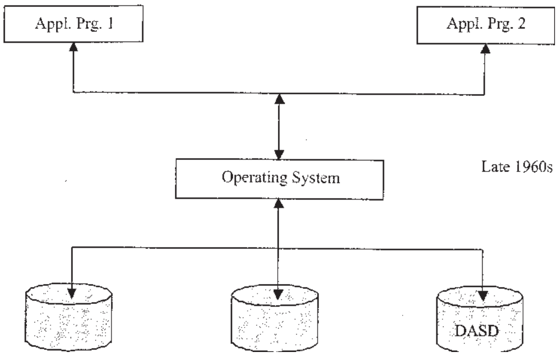
3.4.1 ક્રમાનુસાર (Chronology)

(i) **પ્રાથમિક (પ્રથમ જનરેશન) સિસ્ટમ :** આ કક્ષાને આકૃતિ 3.3.a માં દર્શાવી છે. અહીં સમગ્રતયા પ્રોગ્રામીંગ દર ઉંચો છે, કારણ કે ઉપભોક્તાની એ જવાબદારી છે કે બધાંને I/O (ઈનપુટ / આઉટપુટ) પ્રોગ્રામીગ તથા ફાઈલ મેનેજમેન્ટ એપ્લીકેશન પ્રોગ્રામ (સોફ્ટવેર કે જે ઉપભોક્તાની કાર્યવાહીના સપોર્ટને બંધ કરે છે.) સેકન્ડરી સ્ટોરેજ ડિવાઈસીસમાં (ડાયરેક્ટ એક્સેસ સ્ટોરેજ ડિવાઈસીસ) સ્ટોર કરવામાં આવેલા ડેટાને લખવાની જરૂર પડે છે. આ તબક્કો 1960ના મધ્ય સમયનો કહી શકાય.



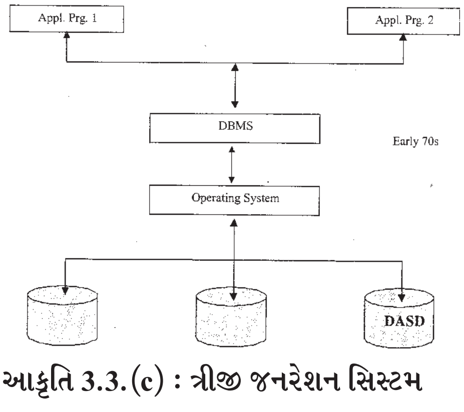
(આકૃતિ) 3.3.a પ્રથમ જનરેશન સિસ્ટમ

(ii) **બીજી જનરેશન સિસ્ટમ** : 1960ના ઉત્તરાર્ધમાં આવેલ આ જનરેશન આકૃતિ 3.3. bમાં બતાવવામાં આવી છે. આ કેસમાં ઓપરેટીંગ સિસ્ટમ ફાઈલ મેનેજમેન્ટ અને અન્ય નિયમિત કાર્યનું ધ્યાન રાખે છે અને પ્રોગ્રામરને આવા કામોમાંથી મુક્તિ અપાવે છે, જો કે ફિઝીકલ ડેટા ઇન્ડિપેન્ડેસ મેળવવામાં આવે છે, તેમ છતાં લોજીકલ ડેટા ઈન્ડિપેન્ડન્સ સિસ્ટમમાં બંધાયેલા રહે છે.



આકૃતિ 3.3.(b) બીજી જનરેશન સિસ્ટમ

(iii) **ત્રીજી જનરેશન સિસ્ટમ** : આકૃતિ 3.3.(c)નો ઉદ્‌ભવ 1970ના પૂર્વાર્ધમાં થયો છે. સિસ્ટમ એ ફિઝીકલ અને લોજીકલ ડેટા ઈન્ડિપેન્ડન્સને પ્રયોજે છે. આનો લાભ એ છે કે પ્રશ્નોત્તરની પ્રક્રિયાને ડેટાના લોજીકલ વ્યૂ ઉચ્ચ કક્ષાના ઓપરેશન લેવલને (DBMS) માંથી પ્રાપ્ય ડેટાને એપ્લીકેશન પ્રોગ્રામમાં પ્રદાન કરે છે.



આકૃતિ 3.3.(c) : ત્રીજી જનરેશન સિસ્ટમ

3.4.2 ડેટાબેઝના કાર્યો અને ઘટકો (Functions and Components of a DBMS)

મૂળભૂત રીતે ડેટાને સંચાલન માટે ફક્ત બે ઓપરેશનની જરૂર પડે છે. દા.ત. આરોપણ અને મેઈન્ટેનન્સ. આરોપણ એ ફાઈલમાંથી ઉપભોક્તાને જોઈતી માહિતી અને ફોર્મને ઉપલબ્ધ કરવા માટેનું અગત્યનું પરિબળ છે. મેઈન્ટેનન્સ સ્ટોર કરવામાં આવેલી ફાઈલના ડેટાને બદલવાનું કાર્ય કરે છે.

ડેટા મેઈન્ટેનન્સમાં ત્રણ ઓપરેશનન્સ સમાયેલા છે - (એડિશન, ડિલીશન અને મોડીફિકેશન) ઉમેરવું, રદ કરવું અને સુધારા - વધારા કરવા, કે જે નવા રેકોર્ડને ઉમેરવા, ચાલુ રેકોર્ડને કેદ કરવા અને ચાલુ રેકોર્ડમાં સુધારા- વધારા કરવા તેમજ વેલ્યુ અપડેટ કરવાના કામને સાદશ્ય કરે છે.

ડેટા મેનેજમેન્ટ સિસ્ટમને બે અગત્યના ઘટકો હોય છે. ડેટા ડેફિનેશન પાર્ટ અને ડેટા મેનીપ્યુલેશન પાર્ટ. ડેટા ડેફિનેશન પાર્ટ ડેટાબેઝ પદાર્થની વ્યાખ્યા અથવા તો વર્ણન કરે છે, અને ડેટા ડેફીનેશન લેંન્ગવેઝમાં લખાય છે. (DDL). જ્યારે ડેટાબેઝને સેટઅપ કરવામાં આવે છે ત્યારે આ ભાગ લોજીકલ સ્ટ્રકચરનું અસ્તિત્વ ઊભું કરે છે.

ડેટાને ઉપજાવવાનો ભાગ મેનિપ્યુલેટીંગ ડેટાને સંયોજે છે અને મેનિપ્યુલેશન લેંગવેજનો (DML) ઉપયોગ કરીને અમલમાં મૂકે છે. ડેટા મેનીપ્યુલેશનની મુખ્ય ચાર રીતો છે. - પ્રોગ્રામીંગ લેંગવેજ ઈન્ટરફેસ, ક્વેરી લેંન્ગવેઝ, રીપોર્ટ રાઈટર્સ અને સિસ્ટમ યુટીલીટીસ.

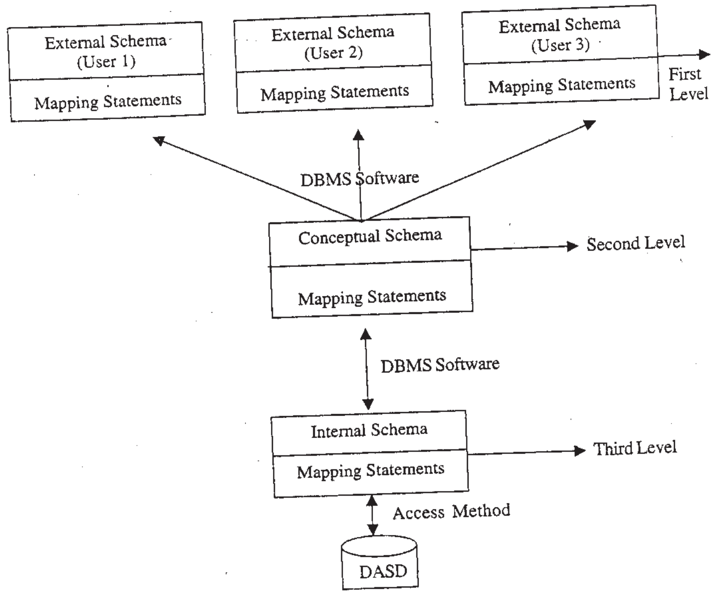
પ્રોગ્રામીંગ લેંગ્વેજ ઈન્ટરફેસ (PLI) અથવા હોસ્ટ લેંગ્વેજ ઈન્ટરફેસ કેટલીક ટાઈપ પ્રોગ્રામીંગ લેંગવેજ મારફત ડેટાબેઝને પ્રાપ્ત કરે છે. PASCAL, C, COBALT વગેરે ક્વેરી લેંગવેજ ડેટાને ઝડપથી પુનઃપ્રાપ્તી માટે મદદ કરે છે. અને તેમાંના કેટલાંક ચોથી જનરેશન - લેંગવેજ (4GL) તરીકે ઓળખાય છે. 4GLs એ નોન પ્રોસીજરલી ભાષા છે, કે જેને ઉપભોક્તાએ સ્પષ્ટ કરવી પડે છે. કે તેને કયા ડેટાની જરૂર છે અને નહીં કે તેની પુનઃ પ્રાપ્તી કેવી રીતે થઈ શકે.

ક્વેરી (query) લેંગવેજને બે વિભાગમાં વહેંચી શકાય - કમાંડ ડ્રાઈવને ક્વેરી લેન્ગવેઝીસ અને સ્ક્રીન ઓરિએન્ટેડ ક્વેરી લેંગવેજીસ. પહેલા કેસમાં ક્રમાંક અંગ્રેજી જેવી ટેક્સ્ટમાં આપવામાં આવે છે, જ્યારે બીજા કેસમાં ઉપભોક્તા કમાંડનો ઉપયોગ (ફીલ - ઈન- ધ- બ્લેંક) 'ખાલી જગ્યા પૂરો' યુક્તિ દ્વારા કરે છે. SQL (સ્ટ્રકચર્ડ ક્વેરી લેંગવેજી) 4GL અને રીલેશનલ સ્ટેન્ડર્ડ લેંગ્વેજ રીલેશનલ DBMSને ઈન્ટરફેસ માટે પહેલી કેટેગરીમાં આવે છે, જ્યારે SQL ફોર્મસ મારફત કવેરીંગ ડેટા એ બીજા વિભાગના ઉદાહરણ છે.

રીપોર્ટ લખનારા પ્રોગ્રામને અનુસરે છે, કે જે ડેટાબેઝમાંથી માહિતી મેળવવા માટે ઉપયોગ કરવામાં આવે છે અને આ માહિતી ઈચ્છીત ફેશનમાં રજૂ કરવા માટે ઉપજાવે છે, અને અંતે સિસ્ટમની ઉપયોગિતાએ પ્રોગ્રામ કે જે સિસ્ટમ મેનેજરને ડેટાબેઝને પુનઃ મેળવી શકાય. ડેટાબેઝમાં, ડેટાને લોડ કરી શકાય, અને ડેટાબેઝ વ્યવસ્થાપન સાથે સંકળાયેલા કાર્યોને કેશ એન્ડ કેરી આઉટ બીજા કાર્યો રીસ્ટોર કરી શકાય.

3.5 ADBMS નું માળખુ (ARCHITECTURE OF DBMS)

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ ત્રણ પ્રકારનું માળખુ ધરાવે છે. (આકૃતિ 3.4) સ્કેમા લેવલ અને વ્યૂનો ઉપયોગ DBMSના સ્ટ્રકચરનું આબેહૂબ વર્ણન કરવા માટે ઉપયોગમાં આવે છે. સ્ટ્રકચરમાં ઉચ્ચ કક્ષાનું લેવલ એ બાહ્ય લેવલ છે, કે જે સામાન્ય રીતે ઉપભોક્તા ડેટાને નિહાળે છે. બાહ્ય લેવલ પણ ક્યારેક સબ-સ્કેમા તરીકે ઓળખાય છે. સામાન્ય રીતે ઉપભોક્તા ટોટલ ડેટાબેઝના અમુક ભાગ સાથે રસ ધરાવે છે, કે જે તેના બાહ્ય દશ્યનો ભાગ બને છે. એ જ ડેટાબેઝના ઉપભોક્તાની જરૂરિયાતના આધારે ડેટાની સુરક્ષા ઘણાં બાહ્ય દશ્ય હોઈ શકે. બાહ્ય સ્કેમાનો ઉપયોગ ડેટાબેઝને મર્યાદામાં રાખીને મેળવવા માટે ઉપયોગ કરી શકે. કન્સેપ્યુઅલ સ્કેમાને, કન્સેપ્યુઅલ DDL (ડેટા ડેફિનેશન લેંગવેજ)નો ઉપયોગ કરીને લખી શકાય.



આકૃતિ 3.4 : આર્કીટેકચર ઓફ અ DBMS

મેપીંગ સ્ટેટમેટ (સોફટવેર પ્રોગ્રામ) પ્રથમ લેવલે જે બાહ્ય અને કાલ્પનિક રૂપરેખા વચ્ચે પત્રવ્યવહાર ધરાવે છે, તે ડેટા ઈન્ડિપેન્ડન્સ પૂરું પાડે છે. તેના અર્થ એવો થયો કે તે વધારાના ડેટાબેઝ પદાર્થનો વધારો કરે છે. અથવા વર્તમાન ડેટાબેઝ સ્ટ્રકચરમાં વર્તમાન ઉપભોક્તને તેનાં એપ્લીકેશન પ્રોગ્રામમાં ફેરફાર કર્યા સિવાય ડેટા ઈન્ડિપેન્ડન્સને પૂરું પાડે છે. લોજીકલ ડેટા ઈન્ડિપેન્ડન્સ સાથે બાહ્ય રૂપરેખાને સ્પર્શ્યા વિના કાલ્પનિક રૂપરેખા વધી શકે છે અને વિકાસ પામે છે.

આર્કિટેકચરનું ત્રીજુ લેવલ એ ઈન્ટરનલ સ્કેમા છે. કોમ્પ્યુટરના બીજા સ્ટોરેજમાં ડેટા ખરેખર કેવી રીતે સ્ટોર થાય છે. અને સંચાલન થાય છે. તેનું વર્ણન કરે છે. ઈન્ડેક્ષ શું છે, તે સ્ટોર કરેલા ફિલ્ડ કેવું પ્રતિનિધિત્વ કરે છે, ફિઝીકલ સીકવન્સ ઓફ સ્ટોર્ડ રેકોર્ડ વગેરે સ્પષ્ટ કરે છે. કન્સેપ્યુઅલ / ઈન્ટરનલ મેપીંગ સ્ટેટમેંટ ફિઝીકલ ડેટા ઈન્ડિપેન્ડન્સને ચોક્કસ કરે છે - એટલે કે સ્ટોર કરવામાં આવેલા ડેટામાં - કન્સેપ્યુઅલ સ્કેમામાં ફેરફાર કર્યા સિવાય ડેટા ખરેખર સ્ટોર કરવામાં આવે છે.

મેપીંગ કોમ્પોનંટ (ઘટક) ઈન્ટરનલ સ્કેમા અને સેકન્ડરી સ્ટોરેજ ડિવાસીસ વચ્ચે મેપીંગ કોમ્પોનંટ (ઘટક) એક્સેસ મેથડ કહેવાય છે. એક્સેસ મેથડ એ સેટ ઓફ રૂટીન છે કે જેનું કામ બધી જ ડીવાઈસીસ આધારિત વિગતોને (DBMS) માંથી રદ કરવાનું છે અને સ્ટોર કરેલા રેકોર્ડ વ્યૂની સાથે (DBMS) રજૂ કરવાનું છે - I.e. ઈન્ટરનલ વ્યૂ. ઘણાં કેસોમાં નીચે દર્શાવેલી ઓપરેટીંગ સિસ્ટમ આ ફંકશનને મહત્ત્વ આપે છે.

◆ **તમારી પ્રગતિ ચકાસો**

2. ડેટાબેઝ સિસ્ટમમાં સ્કેમા અને સબ-સ્કેમા શું છે ?

ડેટાબેઝ આર્કિટેકચરમાં વ્યૂઝનો હેતુ સંક્ષેપમાં સમજાવો.

નોંધ (i) તમારા જવાબો નીચે આપેલી જગ્યામાં લખો.

(ii) તમારા જવાબો પ્રકરણને અંતે આપેલા ઉત્તરો સાથે સરખાવો.

....

....

....

....

....

....

3.6 ડેટા મોડેલિંગ (DATA MODELING)

ડેટા મોડેલીંગ એ એક પ્રક્રિયા છે જે સંસ્થાને અથવા એપ્લીકેશન એરીઆને જરૂરી ડેટા આપવાનું કામ કરે છે. કોન્સપ્યુઅલ મોડેલીંગ એ ડેટાબેઝ એપ્લીકેશનની ડિઝાઈનીંગમાં એક અગત્યનું માધ્યમ છે. પરંપરાગત અભિગમ એટલે આ માટે એન્ટીઝ રીલેશનશીપ (ER) મોડેલનો ઉપયોગ કરવો. સુધારેલ ER અથવા EERમોડેલનો ઉપયોગ, ઓબ્જેક્ટ ઓરિએન્ટેડ ડેટાબેઝના મોડેલીંગ માટે કરવામાં આવે છે.

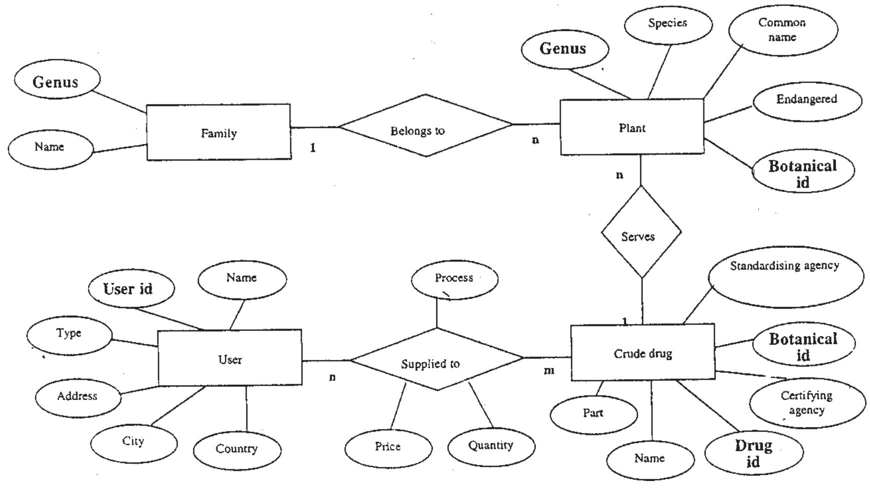
3.6.1 એન્ટીટી - રીલેશનશીપ મોડેલ (Entity Relationship Model)

1976 - 77માં ચેન (CHEN) દ્વારા એન્ટીટી રીલેશનશીપ મોડેલ વિકસાવવામાં આવ્યું હતું. ડેટાબેઝ ડિઝાઈન પ્રોસેસ માટે તે એક શ્રેષ્ઠ હથિયાર છે. તે એન્ટીટી એટ્રીબ્યુટ અને રીલેશનશીપ ગ્રાફિકનું પ્રતિનિધિત્વ કરે છે.

ડિઝાઈન ડેટાબેઝ પૃથક્કરણની જરૂરિયાત, જરૂરી માહિતી મેળવવા માટે ડેટાબેઝમાં સમાવાતા એન્ટીટી એટ્રીબ્યુટનો સમાવેશ કરવામાં આવે છે. એન્ટરપ્રાઈઝના નિયમ પ્રમાણે એન્ટીટી વચ્ચે સંબંધોની ઓળખ આપવામાં આવે છે અને ડેટાબેઝના પ્રકારને સુનિશ્ચિત કરવામાં આવે છે. ER મોડેલ કન્સેપ્ચુઅલ સ્કેમાનું વર્ણન કરે છે અને તેને ડિઝાઈનમાં બ્લૂ પ્રિન્ટ ગણવામાં આવે છે. આખરી રૂપ આપ્યા પછી ER આકૃતિને (જે સામાન્ય રીતે એન્ટી રીલેશનશીપ મોડેલ તરીકે ઓળખાય છે.) કોઈ એક ડેટાબેઝમોડેલમાં પસંદ કરવામાં આવે છે. (આગળ લખાણમાં ચર્ચા કરવામાં આવી છે.) અને ડેટાબેઝ બનાવવાની સિસ્ટમ આધારિત પ્રક્રિયા શરૂ થાય છે.

E R આકૃતિનું ઉદાહરણ આકૃતિ 3.5માં દર્શાવવામાં આવ્યું છે. મેડીસીનલ અને એરોમેટીક પ્લાન્ટમાંથી ડ્રગના માર્કેટીંગને આવરી લે છે. પ્લાન્ટ અથવા તો તેના ભાગ કુદરતી ડ્રગ તરીકે ભાગ ભજવે છે. કે જેનું બજારમાં વેચાણ થાય છે. જ્યારે કોઈ ડ્રગને નિકાસ કરવાના હોય ત્યારે માન્ય એજન્સી ડ્રગની ગુણવત્તાનું સર્ટીફિકેટ આપે છે, જ્યારે સર્ટીફિકેટ આપનારી એજન્સી ડ્રગના નિકાસ માટે માન્યતા પ્રદાન કરે છે. ઉપભોક્તાને મોકલતા પહેલાં ક્રૂડ ડ્રગ્સને કેટલીકવાર પ્રોસેસ કરવામાં આવે છે.

ER આકૃતિમાં લંબચોરસ એન્ટીટીનું, લંબગોળ ડાયમંડ રીલેશનશીપનું, અન્ડર સ્કોર પ્રાયમરી કીનું, ડબલ અંડરસ્કોર ફોરેન કીનું અને I, N, M રીલેશનશીપ ટાઈપ દર્શાવે છે.



3.5 E R આકૃતિનું ઉદાહરણ :

3.6.2 ટાઈપ્સ ઓફ રીલેશનશીપ્સ (Types of Relationship)

બે કે તેથી વધારે એન્ટીટીઝ વચ્ચેનો સહયોગ એટલે રીલેશનશીપ. ડેટાબેઝમાં એન્ટીટીસ રેકોર્ડ ટાઈપને કે જે સેટ્સ છે તેને આવકારે છે. આમ રીલેશનશીપ n સેટના એલીમેન્ટ વચ્ચેના વ્યવહારનું પ્રતિનિધિત્વ કરે છે. બે સેટ ઉપરનું રીલેશનશીપ બાયનરી રીલેશનશીપ કહેવાય છે. ત્રણ સેટ ઉપરનું રીલેશનશીપ - ટર્મરી અને n સેટ ઉપરના સેટને n-ary રીલેશનશીપ કહેવામાં આવે છે.

રીલેશનશીપને પોતાને એન્ટીટી તરીકે ગણવામાં આવે છે અને આરોપણ કરાયેલા રીલેશનશીપને નીચેના વિભાગોમાં ગૃપ કરવામાં આવે છે.

(i) I : I (એક થકી એક)

(ii) I : n (એકથી અનેક)

(iii) n : I (અનેક થી એક)

(iv) n : m (અનેક થી અનેક)

તબક્કાવાર આ સંબંધ વિષે ઉદાહરણ સાથે જોઈએ. એન્ટીટીનો એક દાખલો જોઈએ તો આપેલા ટાઈપ ફડન એક જ સભ્યનો સહયોગ કરે છે. દેશના નામ અને શહેરના નામનો સેટ તેમાં હોઈ શકે. આગળ ઉપર જોઈએ તો આપણે ધારીએ કે સેટમાં આપેલુ એક શહેર એ કેપીટલ છે. આ બન્ને વચ્ચેના સંબંધો જેને કેપીટલ કહી શકાય એ I : I કારણ કે દરેક દેશના નામ, એ ફક્ત એક શહેરનું નામ અને પત્ર વહેવાર માટે, દરેક શહેરનું નામ ફક્ત એક દેશનું નામ ધરાવે છે.

I : n રીલેશનશીપમાં આપેલી એન્ટીટી ટાઈપનું એક દૃષ્ટાંત એ અન્ય ટાઈપના ઘણા દૃષ્ટાંતને સંબંધિત હોય છે. સેટ ઓફ ડિપાર્ટમેન્ટમાં નિયુક્ત સેટ ઓફ ફેકલ્ટી મેમ્બર હોઈ શકે છે. ડિપાર્ટમેન્ટ ફેકલ્ટી રીલેશનશીપ કે જેને કામદાર કહેવામાં આવે છે તે I : n પ્રકારમાં આવે છે કારણ કે દરેક વિભાગ ઘણી ફેકલ્ટી મેમ્બરને નોકરીમાં રાખે છે અને દરેક ફેકલ્ટી મેમ્બર એક વિભાગમાં જ કામ કરે છે.

અનેક થી એક (n : I) રીલેશનશીપને પણ સરખી પરિસ્થિતિ છે કારણ કે તે I : n ઉપરનાં ઉદાહરણમાં જો આપણે રીલેશનશીપને ફેકલ્ટી ડિપાર્ટમેન્ટમાં બદલીએ તો (ડિપાર્ટમેન્ટ ફેકલ્ટીને બદલે) n : I રીલેશન ટાઈપ મળી શકે.

છેવટે n : m રીલેશનશીપમાં એન્ટીટીનાં ઘણાં દૃષ્ટાંત અન્ય એન્ટીટી ટાઈપ સાથે સંકળાયેલા હોય છે. વિદ્યાર્થીઓના સમૂહને અભ્યાસ કરાવતા સેટ ઓફ ફેકલ્ટી મેમ્બરની વાત કરીએ. (ફેકલ્ટી - સ્ટુડન્ટ રીલેશનશીપ (ટીચીંગ) એ n : m) ટાઈપનું દૃષ્ટાંત છે, કારણ કે ફેકલ્ટી મેમ્બર વિદ્યાર્થીઓને ભણાવે છે અને વિદ્યાર્થીઓ 'n'ફેકલ્ટી દ્વારા અભ્યાસ કરે છે.

- **તમારી પ્રગતિ ચકાસો**

3. એન્ટીટી રીલેશનશીપ (ER) આકૃતિનો ભાવાર્થ શું છે ?

નોંધ (i) તમારા જવાબો નીચે આપેલી જગ્યામાં લખો.

(ii) તમારા જવાબો પ્રકરણને અંતે આપેલા ઉત્તરો સાથે સરખાવો.

....

....

....

....

....

....

3.7 ડેટા મોડેલ્સ (DATA MODELS)

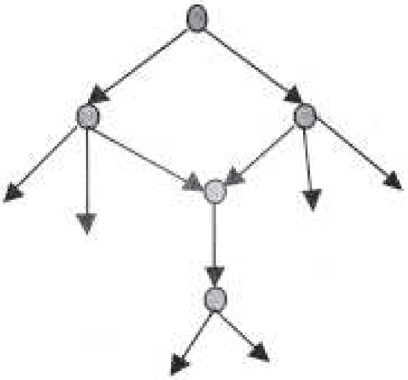
ડેટા મોડેલ્સ એ એવી રીત છે કે જે દ્વારા ડેટાનું માળખુ ખરેખર વિશ્વનું પ્રતિનિધિત્વ કરે છે, અને ડેટાને જે રીતે મેળવવામાં આવે છે. આ મોડેલ રીલેશનશીપના પીકચરીંગનું વારાફરતી પ્રદાન કરે છે ડેટાબેઝના કન્સેપ્ચુઅલ સ્કેમાના મેચીંગ માટે ફ્રેમવર્ક તરીકે કામ કરે છે. આજના સમયમાં ડેટા મોડેલનો ઉપયોગ કરવામાં આવે છે, પણ નીચેના ત્રણનો ઉપયોગ વધારેમાં વધારે છે.

- હાયરાર્કિકલ
- નેટવર્ક
- રીલેશનલ

આ ત્રણ મોડેલ ઉપરાંત કે જેને ક્યારેક ક્લાસીકલ મોડેલ તરીકે સંબોધવામાં આવે છે તે પોસ્ટ રીલેશનશીપ રીસર્ચ એ ન્યૂ ડેટા મોડેલમાં પરિણામે છે, જેને ઓબ્જેક્ટ ઓરીએન્ટેડ મોડેલ તરીકે ઓળખવામાં આવે છે.

3.7.1 હાયરાર્કિકલ મોડેલ (Hierarchical Model)

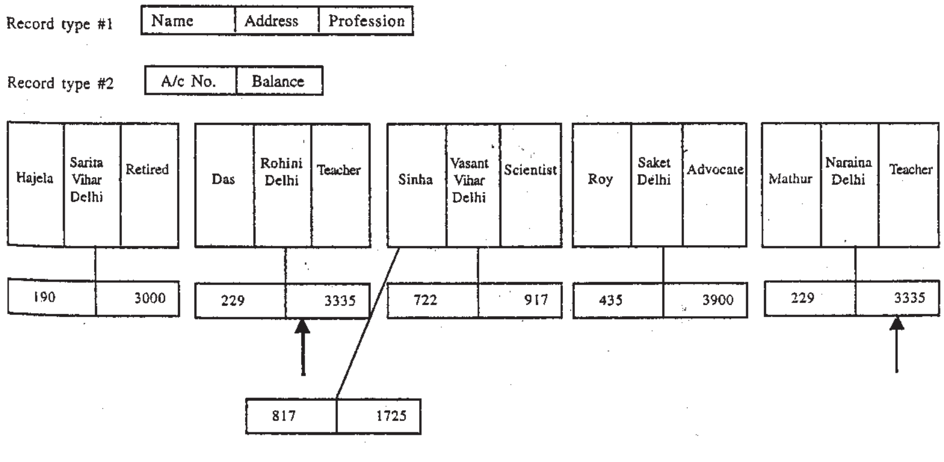
હાયરાર્કિકલ મોડેલ એ ત્રણ પૈકી ઘણું પ્રાચીન છે. આ મોડેલ ડેટા એ રીતે સ્ટ્રક્ચર કરે છે કે જેથી નિશ્ચિત હારયાર્કિકલ પરિસ્થિતિમાં દરેક એલીમેન્ટ એકબીજા પર આધારિત રહે છે. હાયરાર્કિકલ મોડેલ I : n રીલેશનશીપનું રેકોર્ડ ટાઈપ વચ્ચે (આકૃતિ 3.6) પ્રતિનિધિત્વ કરે છે. એક રેકોર્ડ ટાઈપ (I : n) રીલેશનશીપમાંથી "પેરન્ટ" તરીકે નિયોજીત કરવામાં આવે છે, આ પેરન્ટ આઈલ્ડ રીલેશનશીપમાં, આઈલ્ડ રેકોર્ડ ટાઈમને ફક્ત એક જ પેરન્ટ રેકોર્ડ ટાઈપ હોઈ શકે, પણ પેરન્ટ રેકોર્ડને ઘણાં આઈલ્ડ રેકોર્ડ ટાઈપ હોઈ શકે.



આકૃતિ 3.6 : હાયરાર્કિ (I : n રીલેશનશીપ)

ફિઝીકલ સાનિધ્યમાં ડેટાને સ્ટોર કરીને અને ડીપેન્ડન્ટ ચાઈલ્ડ રેકોર્ડને પોઈન્ટર્સનો ઉપયોગ કરીને, હાયરાર્કિકલ મોડેલ અમલમાં મૂકવામાં આવ્યું છે. બિન જરૂરી અનાવશ્યક ડેટાને લીધે મોડેલને તકલીફ પડે છે.

હાયરાર્કિકલ મોડેલનું ઉદાહરણ આકૃતિ 3.7 માં દર્શાવ્યું છે. અહીં હાયરાર્કિને બે રેકોર્ડ ટાઈપ - એક જેમાં ફિલ્ડ ધરાવતા નામ, સરનામુ, વ્યવસાય એકાઉન્ટ નંબર અને અન્ય બીજો એકાઉન્ટ નંબર અને બેલેન્સ, અનાવશ્યક ડેટાને તીર દ્વારા દર્શાવવામાં આવ્યા છે.

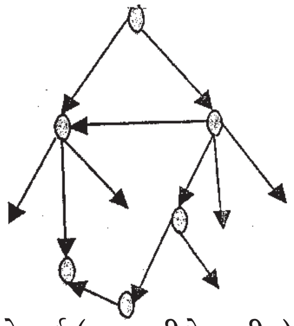


આકૃતિ 3.7 હાયરાર્કિકલ મોડેલનું ઉદાહરણ.

હાયરાર્કિકલ મોડેલના વ્યવહારિક અમલીકરણનું ઉદાહરણ એ IBM's IMS ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ અને CDS / ISIS બીલ્લીઓ ગ્રાફીક એપ્લીકેશન માટે લોકપ્રિય ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ.

3.7.2 નેટવર્ક મોડેલ (Network Model)

નેટવર્ક મોડેલમાં બાળકનો ફક્ત એક પેરન્ટ રેકોર્ડ રદ્દ કરવામાં આવે છે. નેટવર્ક n : m સંબંધને સહયોગ આપે છે. (આકૃતિ 3.8) નેટવર્ક હાયાર્કિ બની શકે છે. (I : n, n : mનો વિશેષ કેસ) પણ હાયરાર્કિ નેટવર્ક બની શકે નહીં.

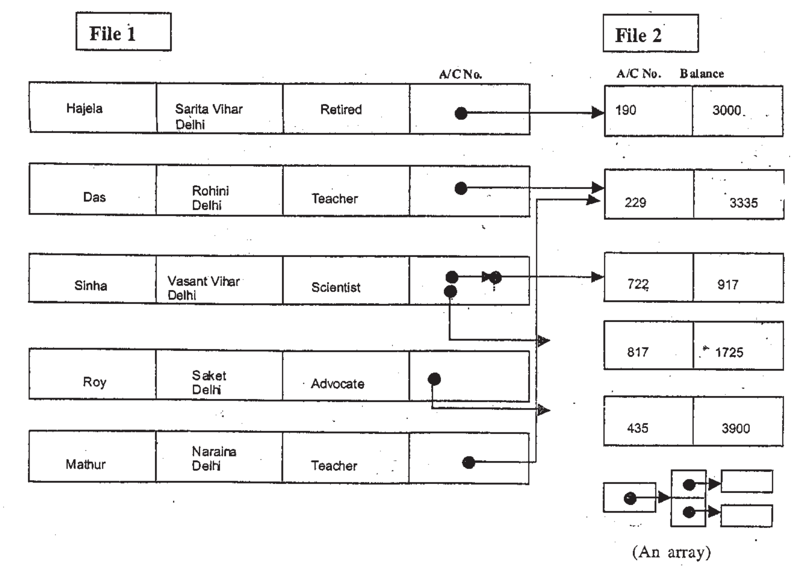


આકૃતિ 3.8 : નેટવર્ક (n : m રીલેશનશીપ)

નેટવર્કનું સંચાલન વિવિધ પોઈન્ટર યોજનાઓ દ્વારા થાય છે. નેટવર્ક એ હાયરાર્કિનું ઉમેરણ હોઈને નેટવર્ક મોડેલની એક સમાન મિલ્કત એ હાયરાર્કિકલ મોડેલના સમાન છે.

નેટવર્ક મોડેલ આકૃતિ 3.9માં દર્શાવી છે. આપેલું ઉદાહરણ હાયરાર્કિ મોડેલની જેમ એ જ રેકોર્ડ ટાઈપનો ઉપયોગ કરે છે. ફેર માત્ર એટલો જ છે કે પ્રથમ રેકોર્ડ ટાઈપ પોઈન્ટર્સ દ્વારા બીજા રેકોર્ડ ટાઈપનો એકાઉન્ટ નંબરનો ઉપયોગ કરે છે. પોઈન્ટર સ્વયં ઘણા બધા પોઈન્ટર તરફ નિર્દેશ કરે છે. (ARRANS) કહેવામાં આવે છે.

કુલીનેટનું (Cullinet's) (IDMS) એ નેટવર્ક મોડેલ આધારિત કોમર્શીયલ (વ્યાપારિક) ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમનું ઉદાહરણ છે.



આકૃતિ 3.9 નેટવર્ક મોડેલનું ઉદાહરણ :

3.7.3 રીલેશનલ મોડેલ (Relational Model)

રીલેશનલ મોડેલ ડેટાને ટેબ્યુલર ફોર્મમાં સાચવે છે, ઉપભોક્તાને તે સહજ છે અને તેનાંથી તે માહિતગાર છે. મોડેલ સારી રીતે વિકસીત મેથેમેટીકલ થીયરી પર આધારિત છે, કે જેમાંથી તે પોતાનું નામ મેળવી શકે છે. સ્ટોર કરવામાં આવેલા ડેટા, સંબંધિત છે તે હકીકત સાથે નામને કોઈ સંબંધ નથી. (ઘણું ખરું હોય છે.)

રીલેશનલ મોડેલ મદદ કરે છે - (I : I, I : n, n : I) અને (n : m) રીલેશનશીપને, રીલેશનલ મોડેલનો દેખીતો અભિગમ એ ડેટા આઈટમ વચ્ચેના સંબંધો પોઈન્ટર્સ દ્વારા સ્પષ્ટપણે કહેવામાં આવે છે, તેને બદલે (DBMS) માટે વિવિધ ટેબલમાંના હયાત મેચીંગ ડેટાને ઓછો કરવાનો છે. ફિઝીકલ લીંકની ગેરહાજરી લવચીક ડેટામેનીપ્યુલેશન પર્યાવરણ પૂરું પાડે છે.

રીલેશનલ મોડેલનું ઉદાહરણ આકૃતિ 3.9માં દર્શાવ્યું છે. અહીં પહેલા ટેબલમાં, બે ટેબલ વચ્ચેના સંબંધ, કોલમને (A/C No.) રીપીટ કરીને સર કરવામાં આવ્યા છે.

Table Name Consumer — Table Name Account

Name	Address	Profession	A/C No.
Hajela	Sarita Vihar	Retired	190
Das	Rohini Delhi	Teacher	229
Sinha	Vasant Vihar Delhi	Scientist	817
Sinha	Vasant Vihar	Scientist	722
Roy	Saket Delhi	Advocate	435
Mathur	Naaina	Teacher	229

A/C No	Balance
190	3000
229	335
817	1725
722	917
435	3900

Collumn repeated to establish relationship

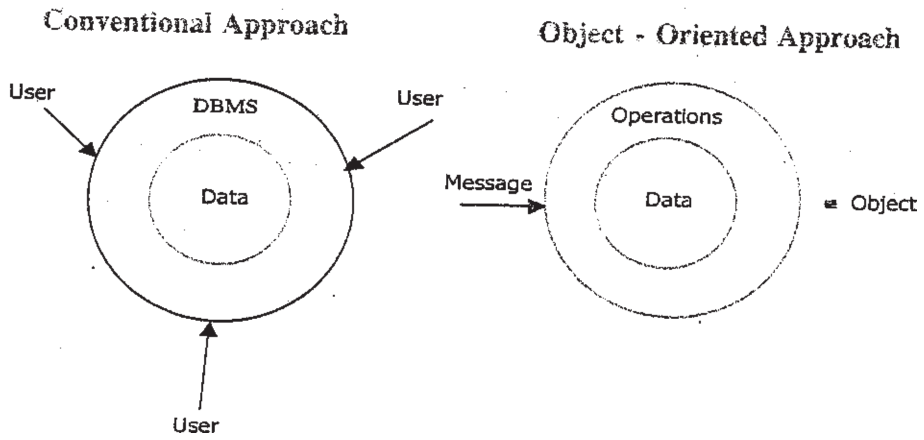
આકૃતિ 3.10 : રીલેશનલ મોડેલનું ઉદાહરણ.

(Oracle INGress)અને (Sybase) અને રીલેશનલ મોડેલ આધારિત, થોડીક બહુ જ જાણીતી કોમર્શિયલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ છે.

3.7.4 ઓબ્જેક્ટ ઓરિએન્ટેડ મોડેલ (Object Oriented Model)

ઓબ્જેક્ટ ઓરિએન્ટેડ મોડેલ રેકોર્ડને બદલે વિષયને રચવા માટે સગવડ કરી આપે છે. ઓબ્જેક્ટ ઓરિએન્ટેડ મોડેલમાં એન્ટીટી એ એક ક્લાસનું દૃષ્ટાંત છે. (ઓબ્જેક્ટ) કે જેને ઓબ્જેક્ટને સ્પર્શતા સેટ ઓફ પ્રોપર્ટીઝ ઓપરેશનનું (મેથડ) પ્રતિનિધિત્વ કરે છે. ક્લાસ એ એબ્ટ્રક્ટ ડેટા ટાઈપનું પ્રતિનિધિત્વ કરે છે. અને એવું કોચલુ છે કે જેમાંથી વ્યક્તિને જોઈતી, જેટલી નકલ જોઈએ (દૃષ્ટાંત) તે ઉપજાવી શકે. ઓબ્જેક્ટ ઓરિએન્ટેડ અભિગમમાં ઓબ્જેક્ટની વર્તણૂંકએ તેની વ્યાખ્યાનો એક ભાગ છે. સેટ ઓફ મેથડસ્ દ્વારા વર્તણૂંકનું વર્ણન કરવામાં આવે છે. તેને ઓબ્જેક્ટ ઈન્ટરફેસ તરીકે ઓળખવામાં આવે છે, ક્લાસ અને પછી ઓબ્જેક્ટ એ રીલેટેડ ક્લાસીસમાંથી પ્રોપર્ટી અને મેથડને વારસો આપી શકે. ઓબ્જેક્ટ અને ક્લાસીસ એ ક્રિયાશીલ છે અને ગમે તે સમયે યોજી શકાય છે. ડેટાને રેકોર્ડ તરીકે જોવાને બદલે ઓબ્જેક્ટ તરીકે જોવાથી તે વધારે લવચીકતા અર્પિત કરે છે અને સામાન્ય ડેટાની જરૂરિયાતને રદ કરે છે.

આકૃતિ 3.11 પરંપરાગત ડેટાબેઝ અભિગમ અને ઓબ્જેક્ટ ઓરિએન્ટેડ ડેટાબેઝ અભિગમની સરખામણી આપે છે.



આકૃતિ 3.11 પરંપરાગત અને ઓબ્જેક્ટ ઓરિએન્ટેડ ડેટાબેઝ અભિગમની સરખામણી.

કેટલાંક ઓબ્જેક્ટ બિલ્ડીંગની ઓળખ આ પ્રમાણે આપી શકાય.

- **ઓબ્જેક્ટ** : ઓબ્જેક્ટ એ એક અસ્તિત્વ ધરાવનાર, વાસ્તવિક અથવા ભાવવાચક કે જેને લાગણી, વર્તણૂંક અને ઓળખ હોય.
- **મેસેજીસ (સંદેશા)** : ઓબ્જેક્ટ સંદેશા દ્વારા એક બીજા સાથે વાતચીત કરે છે. ઓબ્જેક્ટ દ્વારા કેવું સંચાલન કરવું તે સંદેશા નક્કી કરે છે. સંદેશ એ ઓપરેશનનું નામ અને દલીલોનું લીસ્ટ સ્પષ્ટ કરે છે.
- **ક્લાસીસ** : ક્લાસ એ ઓબ્જેક્ટસનો સેટ છે કે જે સામાન્ય આરોપણ અને વર્તણૂંક સાથે સહયોગ કરે છે. દરેક ઓબ્જેક્ટ એ કોઈ એક ક્લાસનું ઉદાહરણ છે.

  ઓબ્જેક્ટ ઓરિએન્ટેડ અભિગમ ઇન્ક્રીમેન્ટલ સોફ્ટવેર ડેવલપમેન્ટ પર ભાર મૂકે છે, આ અભિગમના સિદ્ધાંતો નીચે પ્રમાણે છે.
- સોફ્ટવેરને ઉછેરો, તેને બાંધો નહીં
- સમગ્ર સિસ્ટમને બદલે તેના ઘટકોને બાંધો અને
- મૂળભૂત સિસ્ટમને એકત્ર કરો અને પછી તેમાં વધારો કરો

  સ્મોલટોક C++, જાવા અને ઓબ્જેક્ટ પાસ્કલ/ ડેલ્ફીએ આ અભિગમમાં ઉપયોગમાં લેવાતી ઓબ્જેક્ટ ઓરિએન્ટેડ પ્રોગ્રામીંગ ભાષા છે.

- **તમારી પ્રગતિ ચકાસો**

4 બીજા ડેટા મોડેલ કરતાં RDBMS વિશાળ એપ્લીકેશનને ધરાવતું શા માટે જણાય છે ? RDBMSના થોડાક ઉદાહરણો આપો

નોંધ (i) તમારા જવાબો નીચે આપેલ જગ્યામાં લખો.

(ii) આ પ્રકરણના અંતે આપેલા જવાબો સાથે તમારા ઉત્તરો સરખાવો

....

....

....

....

....

....

3.8 રીલેશનલ ડેટાબેઝ સિસ્ટમ (RDBMS) (RELATIONAL DATABASE MANAGEMENT SYSTEMS (RDBMS))

રીલેશનલ ડેટા મોડેલ 1970માં Edgar F.cod દ્વારા સૌ પ્રથમ સૂચવવામાં આવ્યું ત્યારથી રીલેશનલ ડેટાબેઝ સિસ્ટમનો ઉદ્‌ભવ થયો છે. તે ડેટાબેઝ મેનેજમેન્ટમાં ડી-ફેક્ટો આંતરરાષ્ટ્રીય સ્ટેન્ડર્ડ બન્યા છે. ઓબ્જેક્ટ ઓરિએન્ટેડ ડેટા મેનેજમેન્ટની આગેકૂચ થવા છતાં, રીલેશનલ સિસ્ટમમાં ઘણા સમય સુધી તે લોકપ્રિય રહેશે.

રીલેશનલ ટેકનોલોજીની કેટલીક ટર્મને જોઈએ : રીલેશનલ ડેટાબેઝ એ ડેટાબેઝનો સંગ્રહ છે કે જે સિસ્ટમ ઘણા રીલેશનને આવરી લે છે.

અગાઉ નોંધ્યું તે પ્રમાણે રીલેશન એ ગણિતિક અભિગમ છે. ગાણિતિક રીતે રીલેશન એ ક્ષેત્રના લીસ્ટનો કાર્ટીઝન પ્રોડક્ટનો સબસેટ છે ક્ષેત્ર (Domain)એ સેટ ઓફ વેલ્યુઝનું પ્રતિનિધિત્વ કરે છે કે જે ક્ષેત્રનું આરોપણ કરે છે (દા.ત. સંખ્યા, અક્ષરો વગેરે) રીલેશનનું માપ, રીલેશનમાં આરોપણ કરવામાં આવેલા અંકર પર આધારિત છે. રીલેશનમાં આવેલ ટપલના નંબરને મૂળભૂત પાયાના નંબર કહેવામાં આવે છે. દા.ત. ઉપભોક્તા ટેબલ આકૃતિ 3.9ને ચાર એટ્રીબ્યુટ્સ છે અને છ ટપલ્સ છે, અને તેથી આ રીલેશનની માત્રા ચારની છે અને (Cardinality) મૂળભૂત છની છે.

રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ (RDBMS) એ રીલેશનલ ડેટાબેઝમાં ડેટાનું સંચાલન અને ડેટાને ઉપજાવી કાઢવામાં (મેનીપ્યુલેટ) મદદ કરનાર પ્રોગ્રામનો સંગ્રહ છે.

(RDBMS) નિરઝોનમાં અદલ- બદલ નીચેની ટર્મનો ઉપયોગ કરવામાં આવે છે.

- રીલેશન, ટેબલ, ડેટાબેઝ, ફાઇલ
- (એટ્રીબ્યુટ) ક્ષેત્ર, કોલમ, ફિલ્ડ
- ટપલ, રો, રેકોર્ડ
- ડોમેઈન, રેન્જ, ટાઈપ

3.8.1 રીલેશનની લાક્ષણિકતા (Characteristics of a Relation)

(i) રીલેશનને હંમેશા નામ હોય છે.

(ii) રીલેશનના દરેક કોલમને ચોક્કસ નામ હોય છે (જો કે વિવિધ રીલેશન્સના કોલમનું એ જ નામ હોઈ શકે.)

(iii) રીલેશનમાં કોલમના ક્રમાંકનું કોઈ મહત્ત્વ નથી હોતું.

(iv) રીલેશનમાં બે એકસમાન ટપલ્સ હોઈ શકે નહીં

(v) રીલેશનમાં ટપલ્સના ક્રમાંકનું કોઈ મહત્ત્વ હોતું નથી.

(vi) રીલેશનના દરેક કોલમનું નિશ્ચિત ક્ષેત્ર હોય છે.

(vii) એ જ ક્ષેત્ર ધરાવતા પ્રકારમાં એક કરતા વધારે કોલમ હોઈ શકે.

(viii) રીલેશનમાં કોલમને એકથી વધારે ક્ષેત્રનું માપ ન હોઈ શકે

(ix) દરેક કોલમ એ જ વેલ્યુના આરોપણ અને દરેક ટેબલ સેલ વેલ્યુ (રો અને કોલમના ઇન્ટરસેક્શન) એ સીંગલ વેલ્યુડ હોવા જોઈએ

રીલેશનનું સ્ટ્રક્ચર દા.ત. કોઈપણ જાતની વેલ્યુનું તેમના ઉપર, સોંપણી કર્યા વગર આરોપણ કરવું તેનું રીલેશન ઇન્સટન્સ કહેવામાં આવે છે.

ટેબલ સામાન્ય, રીતે તેની સંબંધિત યોજનાનું પ્રતિનિધિત્વ કરે છે, કે જેને ટેબલ નામની વ્યાપ્તિ છે અને કૌંસમાં આપેલા આરોપણ કરવામાં આવેલ નામને અનુસરે છે. ટેબલની સંબંધિત યોજનાને આકૃતિ 3.10માં દર્શાવી છે.

ગ્રાહક (Customer) (નામ, સરનામું, વ્યવસાય (A/C) નંબર)

એકાઉન્ટ (A/C નંબર, બેલેન્સ)

પ્રાયમરી કીનું આરોપણ એ રીલેશન યોજનામાં (Underlined) ભાર મૂકવામાં આવે છે.

3.8.2 કી અને તેનું કાર્ય (Keys and their Functions)

કી-એ રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમમાં અગત્યનો ભાગ ભજવે છે.

(i) ટપલ્સનું આઈડેન્ટીફિકેશન

(ii) ટેબલ વચ્ચેના સંબંધ નિશ્ચિત કરવા.

કી (ડેટા આઈટમ)ને રચવા માટે સીંગલ કીનું આરોપણ કરવામાં આવે છે તેને સીંગલ કી એટ્રીબ્યુટ અથવા મલ્ટીપલ કી એટ્રીબ્યુટ કહેવામાં આવે છે. મલ્ટીપલ એટ્રીબ્યુટ દ્વારા કરવામાં આવેલી કી એ સંકલિત અથવા સંકળાયેલી કી તરીકે ઓળખાય છે.

સુપર કી એ સેટ ઓફ એટ્રીબ્યુટ છે કે જેને એકત્રિત રીતે એન્ટીટી સેટમાં યુનિકલી આઈડેન્ટીફાય કરવા આપણને મંજૂરી આપે છે જો કોઈ કી સુપર કી હોય તો સુપર કીનો સુપર સેટ પણ સુપર કી છે. (જો (X1) એ (X)નો સેટનો સબ સેટ છે, તો (X) એ (X1)નો સુપર સેટ કહેવાય.)

સૌથી નાની સુપર કી જે મીનીમલ કી તરીકે પણ ઓળખાય છે, કારણ કે તેના જેવા સબ સેટમાં તે યોગ્ય નથી, તેથી તે સુપર કી મીનીમલ કીમાંની એક પ્રાયમરી કી તરીકે પસંદ કરવામાં આવે છે. મીનીમલ કીના સેટમાં કી કેન્ડીડેટ અથવા ઓલ્ટરનેટ કી તરીકે ઓળખાય છે. તે ડેટાબેઝ ડિઝાઇનરને આધિન છે કે, તે કેન્ડીડેટ કી માંથી પ્રાયમરી કી ની પસંદગી કરે.

પ્રાયમરી કી એ આરોપણ છે અથવા આરોપણનું જોડાણ છે કે જે રેકોર્ડને અસામાન્ય રીતે ઓળખાય છે. જ્યારે સેકન્ડરી કી એ રેકોર્ડને અસામાન્ય રીતે ઓળખતું નથી, સેકન્ડરી કી સાદશ્ય બધા જ રેકોર્ડની ઓળખ આપે છે.

ફોરેન કી એ આરોપણ (Attribute) અથવા આરોપણનું જોડાણ છે, કે જે ટેબલ્સને લીંક કરવા માટે ઉપયોગ કરવામાં આવે છે. રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમમાં ફોરેન કીનો ઉપયોગ ટેબલ વચ્ચે લીંકીંગ પીન તરીકે કરવામાં આવે છે. ફોરેન કી, પ્રાયમરી કીને લીંક કરવાનું કામ કરે છે.

પ્રાયમરી અને ફોરેન કી રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમમાં પ્રમાણિકતાના નિયમો પ્રદાન કરવા માટે ગંભીર છે. રીલેશનલ સિસ્ટમમાં સર્વશ્રેષ્ઠ માર્ગદર્શન છે કે રીલેશનની પ્રાયમરી કી અથવા અન્ય આરોપણ સંકલિત પ્રાયમરી કીમાં ભાગ લે છે તેની વેલ્યુ કંઈ પણ નહી (Null) ન હોઈ શકે. આ નિયમને એન્ટીટી ઇન્ટીગ્રીટી કહેવામાં આવે છે.

ઈન્ટીગ્રીટીનો બીજો પણ એક નિયમ છે, જે ફોરેન કીને લાગુ પડે છે. આ નિયમ પ્રમાણે એટ્રીબ્યુટ કે જે એક ટેબલમાં ફોરેન કી એ બીજા ટેબલમાં પ્રાયમરી કી હોવી જોઈએ. આ નિયમ રેફરન્શીયલ ઇન્ટીગ્રીટી તરીકે ઓળખાય છે.

3.8.3 DBMSને રીલેશનલ થવા માટેના માનદંડ (Criteria for a DBMS to be Relational)

ઘણાં નવા (DBMS) પેકેજીસ 'રીલેશનલ' તરીકે ગણવામાં (Labelled) આવે છે તેથી ખરા રીલેશનલ DBMSને રીલેશન લાઇક સિસ્ટમનો તફાવત સ્પષ્ટ કરવો તે અગત્યનું છે. સિસ્ટમને રિલેશનલ ગણવા માટેની ઓછામાં ઓછી શરતો

(i) ડેટાને ટેબલના ફોર્મમાં પ્રતિનિધિત્વ હોવું જોઈએ.

(ii) DBMS ઉપભોક્તાને કોઈ પણ પોઇન્ટર મિકેનિઝમ (ટ્રાન્સપરન્ટ) પારદર્શક હોવું જોઈએ.

(iii) Select, Project અને (Join)ના રીલેશન એલ્જીબ્રા ઓપરેટર્સને સિસ્ટમનો આધાર હોવો જોઈએ.

કોઈ પણ સિસ્ટમ આ ત્રણ માનદંડને પૂર્ણ કરે તેને મીનીમલી રીલેશન કહેવાય.

પ્રથમ બે શરતોને સંતોષે, તો તે રીલેશનલ સિસ્ટમ નથી અને ટેબ્યુલર DBMS કહેવાય.

(DBMS) માટે સંપૂર્ણ રીલેશન થવા માટે તેને એન્ટીટી અને રેફરન્સીઅલ ઈન્ટીગ્રીટી નિયમ એમ બન્નેનો વધારાનો સપોર્ટ હોવો જોઈએ. અને બધાં રીલેશનલ એલ્જીબ્રા ઓપરેશન્સનો અમલ કરવો જોઈએ.

રીલેશનલ ડેટાબેઝ થીયરીના શોધક (E.F. Cold) એ સંપૂર્ણ રીલેશનલ ડેટાબેઝ સિસ્ટમને સ્પષ્ટ કરવા 12 નિયમોને તૈયાર કર્યા છે. આ નિયમો રિલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ એ ડેટાબેઝને રીલેશનલ કેપેસીટીઝના આધારે સંચાલન કરવું જોઈએ, તે બાબત પર આધારિત છે.

3.9 રીલેશનની સામ્યતા (NORMALISATION OF RELATIONS)

ડેટાબેઝ ડીઝાઈનમાં રીલેશનની સામન્યતા એ એક અગત્યનું પાસું છે, કે જે ડેટાની ભાષાશાસ્ત્રને સ્પર્શે છે. એ એવી રીતે છે કે જે ડેટાને એવી રીતે રચે છે કે જેથી ડેટાબેઝમાં કોઈ દેખીતી ભૂલ ન રહી જાય. દેખીતી ભૂલ એ અનિચ્છનીય સાઈડ ઈફેક્ટ છે કે જે રીલેશનમાં નબળા ડેટાબેઝ ડિઝાઈનમાં પરિણમે છે. સામાન્યતાની પ્રક્રિયા ઘણું ખરું થોડા આરોપણ દ્વારા બે અથવા વધારે રીલેશન્સનું પૃથક્કરણ કરે છે. દા.ત. પેરન્ટ રીલેશન્સના વર્ટીકલ (શિરોબિંદુ) સબસેટને લઈએ. રીલેશન્સને વિભિન્ન કરવા માટે ઉપયોગમાં લેવાતા માનદંડ સામાન્યતાનું લેવલ નક્કી કરે છે, તેને નોર્મલ ફોર્મ કહેવામાં આવે છે.

તેની નોંધ લેવી જ જોઈએ કે સામાન્યતઃ પ્રાથમિક રીતે પુનઃ પ્રાપ્તિ કરવાની નિપૂણતા સુધારવાના બદલે ડેટાના મેઈન્ટેનન્સના પ્રશ્નોને અટકાવવા માટે અથવા ઓછું કરવા માટે લક્ષિત કરે છે. રીલેશનને સામાન્ય ફોર્મમાં બદલવા માટે વિચ્છિન્ન થયેલા રીલેશન્સને જોડવા, ડેટા પુનઃ પ્રાપ્ત કરવા મુખ્ય એક અસલ ટેબલ સ્પીડ પુનઃ પ્રાપ્ત કરવા માટે સારી ભવિષ્યવાણી કરી શકે નહીં. આમ, રીલેશન્સને સામાન્ય કરવા માટે ઈન્ટીગ્રીટી એકરૂપતા અને ડેટાબેઝમાં સ્ટોર કરવામાં આવેલા ડેટાનું સમગ્રતયા મેઈન્ટેનન્સને સુધારવા માટે આપણે રીટાઈવલ સ્પીડનું બલિદાન આપીએ છીએ.

ઉપર જણાવ્યા પ્રમાણે રીલેશનની સામાન્યતા ડેટાબેઝમાં આવેલી દેખીતી ભૂલને નિવારે છે. દેખીતી ભૂલને ત્રણ વિભાગમાં વહેંચી શકાય.

- ઈન્સર્સન એનામલીસ
- ડિલીશન એનામલીસ
- અપડેટ એનામલીસ

જ્યારે ટપલને ટેબલમાં ઈન્સર્ટ ન કરી શકીએ ત્યારે ઈન્સર્સન એનામલી બને છે. જ્યારે પ્રાયમરી કી ની વેલ્યુ જાણવામાં નથી હોતી, ત્યારે આ પરિસ્થિતિ ઉદ્દભવે છે. એન્ટીટી ઈન્ટીગ્રેશન નિયમ પ્રમાણે પ્રાયમરી કીની વેલ્યુ કંઈ પણ નહીં (Null) ન હોઈ શકે. તેથી ટપલની પ્રાયમરી કીના આરોપણની સમકક્ષ ટપલને દાખલ કરતા પહેલાં ટપલની સોંપણી કરવી જોઈએ. જો આ વેલ્યુ અજાણી હોય તો ટપલ ટેબલમાં દાખલ કરી શકાય નહીં.

ડીલીશન એનામલીના કેસમાં ટપલનું રદ કરવું તે ડેટાબેઝમાં પ્રશ્નો ઉભા કરે છે. જ્યારે ટપલને રદ કરવામાં આવે છે ત્યારે આવું બને છેકે માહિતી ધરાવતા ટેબલમાં માહિતી ધરાવતું છેલ્લું ટપલ હોઈને અગત્યની માહિતી ધરાવે છે. ટપલના રદ કરવાથી અગત્યની માહિતી ડેટાબેઝમાંથી રદ થઈ જાય છે.

અપડેટ અનામલી ત્યારે બને છે જ્યારે આપણી પાસે ડેટામાં ઘણું અનાવશ્યક ડેટા હોય. વારંવાર અપાતા એટ્રીબ્યુટ વેલ્યુને અપડેટ કરવાની છે. વારંવાર આવતી તે વેલ્યુનું આપણે સર્ચ કરવું પડે છે અને પછી તેને બદલવું પડે છે. જ્યારે સામાન્ય ફોર્મની ચર્ચા કરીશું ત્યારે એનોમલીને વધારે વિસ્તૃત કરવામાં આવશે. આપણે સામાન્યતઃ વિશે વધારે ચર્ચા કરીએ તે પહેલાં ડિપેન્ડન્સીનો અભિગમ સમજવો વધારે ઉપયોગી થશે.

3.9.1 પરતંત્રતા (Dependencies) : ડિપેન્ડન્સી એ આરોપણ મધ્યે રીલેશનશીપને અનુસરે છે. આ આરોપણ એ જ રીલેશનને લાગુ પડે અથવા જુદાજુદા રીલેશનને લાગુ પડે. ડિપેન્ડન્સીસ ઘણા પ્રકારના હોઈ શકે દા.ત. ફંક્શનલ કી ડિપેન્ડન્સીસ, કન્ઝીટીવ ડિપેન્ડન્સીસ, મલ્ટી વેલ્યુડ ડિપેન્ડસીસ, જોઈન-ડિપેન્ડન્સીસ વગેરે આપણે આમાંના થોડાક ડિપેન્ડન્સીસને સંક્ષેપમાં તપાસીશું.

◆ **ફંક્શનલ ડિપેન્ડન્સી (FD)** ફંક્શનલ ડિપેન્ડન્સી એ આરોપણ વચ્ચેના ભાષાશાસ્ત્રના સહયોગનું પ્રતિનિધિત્વ કરે છે. જો એટ્રીબ્યુટ Aની વેલ્યુ બીજા એટ્રીબ્યુટ- Bની વેલ્યુ નક્કી કરે તો આપણે કહી શકીએ કે B, A ઉપર ફંક્શનલી ડીપેન્ડન્ટ છે. આને આ પ્રમાણે દર્શાવી શકાય.

A -> B અને વંચાય "A determing B" અને A ડીટરમીટંટ કહેવાય છે.

એની નોંધ લેવી જોઈએ કે જ્યારે ડેટા આઇટમ (અથવા ડેટા આઇટમનો સંગ્રહ) અન્ય ડેટા આઇટમ નક્કી કરે છે ત્યારે એ જરૂરી નથી કે સેકન્ડરી ડેટા આઇટમ પ્રથમને ડિટરમાઇન કરે. એમ કહેવાય છે કે જો તે સંયુક્ત આરોપણમાં ડિપેન્ડન્ટ હોય અને કોઈ યોગ્ય જોડાણના સબસેટ ઉપર આધારિત ન હોય તો તે સંપૂર્ણ ફંક્શનલી ડીપેન્ડન્ટ છે. ફંક્શનલ ડિપેન્ડન્સીનું ઉદાહરણ આપીએ તો, નીચે આપેલી રીલેશન યોજનાઓના સંબંધી વિશે જોઈએ. આપણે તેને આ પ્રમાણે બતાવી શકીએ.

બુક (બુક id, સબ્જેક્ટ, યર ઓફ પબ્લિકેશન, પ્રાઇસ) બુક id (બુક આઇડેન્ટીફાયર અથવા આઇડેન્ટીફિકેશન નંબર) એ રીલેશનની પ્રાયમરી કી છે અને તેથી સબ્જેક્ટ, યર ઓફ પબ્લિકેશન અને પ્રાઇસ નક્કી કર છે.

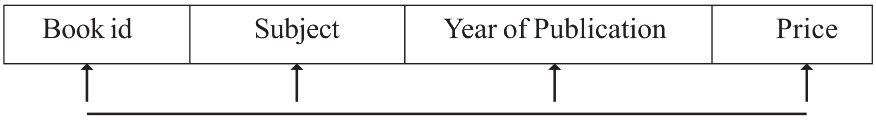
આપણે તેને આ પ્રમાણે બતાવી શકીએ.

Book id → Subject

Book id → Year of Publication

Book id → Price

આને આપણે ડિપેન્ડન્સી ડાયાગ્રામના માધ્યમ દ્વારા પણ ગ્રાફિકલી બતાવી શકીએ.



◆ **ટ્રાન્સીઝીટવ ડિપેન્ડન્સી** : ટ્રાન્ઝીટીવ ડિપેન્ડન્સી એ ઇન્ટરમીડિએટ ડિપેન્ડન્સીનો પ્રકાર છે. દા.ત. જો આપણી પાસે એટ્રીબ્યુટ અથવા એટ્રીબ્યુટનો સમૂહ (A, B,) અને (C) હોય, જેમ કે (A B)ને નક્કી કરે છે અને B C ને નક્કી કરે છે. દા.ત.

A → B

B → C

તો પછી આપણે કહીએ કે ટ્રાંઝીટીવ ડિપેન્ડન્સી (A B C) દ્વારા પ્રતિનિધિત્વ કરે છે. તે (A) અને (C) વચ્ચે અસ્તિત્વ ધરાવે છે.

આપણે રીલેશન ફેકલ્ટીને ધ્યાનમાં લઈએ

ફેકલ્ટી (ફેકલ્ટી (id), નેમ, ડિપાર્ટમેન્ટ, ઑફિસ સેલરી) આ સંબંધમાં આપણે માની લઈએ કે દરેક વિભાગને તેમનું પોતાનુ ઑફિસ બિલ્ડીંગ, અન્ય ઉપરાંત આપણી પાસે નીચે પ્રમાણે ફંક્શનલ ડિપેન્ડન્સી છે.

ફેકલ્ટી id → ડિપાર્ટમેન્ટ, (પ્રાયમરી કી હોવાને લીધે ફેકલ્ટી id)

ડિપાર્ટમેન્ટ → ઑફિસ (એન્ટરપ્રાઇઝના નિયમોના આધારે અથવા રીલેશન ઉપર લાદવામાં આવેલા નિયમ પ્રમાણે.)

આ પ્રમાણે આપણી પાસે રીલેશનમાં નીચે પ્રમાણે ટ્રાન્ઝીટીવ ડિપેન્ડન્સી છે.

ફેકલ્ટી id → ડિપાર્ટમેન્ટ ઑફિસ

◆ **મલ્ટી વેલ્યુડ ડિપેન્ડન્સી** : મલ્ટીવેલ્યુડ ડિપેન્ડન્સી M : N (અનેકથી અનેક) રીલેશનશીપને રીફર કરે છે. આપણે કહીએ છીએ કે મલ્ટી વેલ્યુડ ડિપેન્ડન્સી બે ડેટા આઇટમ વચ્ચે હોય છે કે જ્યારે પ્રથમ ડેટા આઇટમની એક વેલ્યુ બીજા ડેટા આઇટમની વેલ્યુના સંગ્રહને આપે છે. દા.ત. બીજી ડેટા આઇટમને મલ્ટી ડીટરમાઇન કરે છે.

◆ **જોઈન ડિપેન્ડન્સી :** જો આપણે રીલેશનને નાના રીલેશનમાં વિભાજીત કરીએ અને નાના રીલેશનના જોડાણ (Join) આપણને ટપલ્સ આપે નહીં તો, આપણે કહીએ કે રીલેશનને જોઇન ડિપેન્ડન્સી છે.

રીલેશન સેમ્પલને ફંક્શનલ ડિપેન્ડન્સીસ સાથે દર્શાવ્યા પ્રમાણે વિચારીએ

Sample

A	B	C	FD	A-B
a1	b1	c1		
a2	b2	c3		CB
a3	b1	c2		
a4	b2	c4		

હવે આપણે આ રીલેશનને બે રીલેશનમાં વિભાજિત કરીએ (Split-(A, B) અને Split 2 (B,. C) રીલેશન સેમ્પલમાં ફંક્શનલ ડિપેન્ડન્સી એ જ રહેશે

Split1		Split2	
A	B	B	C
a1	b1	b1	c1
	F.D.-A		F.D.C- B
a2	b2	b2	c3
a3	b1	b1	c2
a4	b2	b2	c4

હવે જો આપણે Split-1 અને Split-2ને સમાન એટ્રીબ્યુટ સાથે જોડીએ, તો આપણને રીલેશન Sample-1 મળે

Sample-1

A	B	C
a1	b1	c1
a1	b1	c2*
a2	b2	c3
a2	b2	c4*
a3	b1	c2
a3	b1	c1*
a4	b2	c3*
a4	b2	c4

રીલેશન Sample-1 દ્વારા આપણે જોઈ શકીએ છીએ કે તેને ચાર વધારાના (નકલી) ટપલ્સ છે (* દ્વારા દર્શાવ્યા છે) કે જે મૂળ રીલેશન્સ Sample માં હાજર નહોતા. આ પ્રકારનું જોડાણ વિખૂટું જોડાણ કહેવાય છે કારણ કે મૂળ ટેબલમાં સચવાયેલી માહિતી ખોવાઈ જાય છે. આમ થવા પાછળનું કારણ એ છે કે મૂળ રીલેશનમાં જોઈન એટ્રીબ્યુટ એ ડીટરમીટંટ નહોતું અને તેથી તેને જે રીતે વિભાજીત કરવામાં આવ્યું તે પ્રમાણે કરવું જોઈતું નહોતું. આપણે કહીએ કે જોઇન ડિપેન્ડન્સી આ કેસમાં ઉપસ્થિત છે. જો આપણે રીલેશનને વિભાજીત કરીએ અને કોન્સ્ટીટ્યુઅંટ રીલેશનને જોડાણ કરીએ તો આપણને નુકસાનરહિત (Lossless) જોઈન મળી શકે. દાખલા તરીકે (Sample New)નું ઉદાહરણ લઈએ તેને Sample New અને Sample New2 માં નીચે પ્રમાણે વિભાજીત કરીએ.

Sample New

X	Y	Z	
x1	y1	z1	
x2	y2	z2	F.D. X-Y Xz
X3	y2	z1	
x4	y1	z2	

Samle New1 Sampel New 2

x	y	FD X-y	Xz	FD X-z
x1	y1		x1	z1
x2	y2		x2	z2
x3	y2		x3	z1
x4	y1		x4	z2)

Sample New1 અને Sample New 2નું જોડાણ કોઈ પણ બનાવટી રો વગર અસલ ટેબલ Sample New આપે છે, કારણ કે, એટ્રીબ્યુટ (X) કે જેના ઉપર ટેબલ વિભાજીત કરેલું છે, તે ડીટરમીટંટ એટ્રીબ્યુટ છે.

3.9.2 નોર્મલ ફોર્મસ (Normal Forms)

સામાન્યતાની ટેક્નિકનો આધાર એટ્રીબ્યુટ વચ્ચે ફંક્શનલ ડિપેન્ડન્સીના પૃથક્કરણ પર રહેલો છે. સમગ્ર રૂપરેખાને ધ્યાનમાં લેતા એન્ટરપ્રાઇઝ નિયમો અને ભાષાશાસ્ત્રનું મથામણ, રીલેશનમાં આ બધા જ ફંકશનલ ડિપેન્ડીસને કબજામાં લેવામાં આવ્યા છે અને રીલેશન્સને યોગ્ય સામાન્ય ફોર્મમાં બદલવામાં આવે છે. સામાન્ય ફોર્મ રીલસેશનના સામાન્યતા લેવલનું પ્રતિનિધિત્વ કરે છે. રીલેશન્સને લોઅર નોર્મલ ફોર્મમાંથી હાયર નોર્મલ ફોર્મમાં બદલવા માટે સામાન્યતા આગળ વધે છે.

નોર્મલ ફોર્મ્સ : પ્રથમ નોર્મલ ફોર્મ (INF), બીજું (2NF) અને ત્રીજું નોર્મલ ફોર્મ (3 NF) બોયસ કોડ નોર્મલફોર્મ (BCNF) ચોથું નોર્મલ ફોર્મ (4NF), પાંચમું નોર્મલ ફોર્મ (5NF) અને સૌથી ઉંચુ નોર્મલ ફોર્મ કે જે ડોમેઇન / કી નોર્મલ ફોર્મ (DK/NF) કહેવાય છે.

(EF)કોડ એ પહેલા ત્રણ નોર્મલ ફોર્મની રચના કરી જેની આપણે વિગતવાર ચર્ચા કરીશું. મોટા ભાગના મળી આવતા (DBMS) અને (3 NF) અથવા (BCNF) સુધી સામાન્યતા બની રહે છે એટલે કે જો રીલેશન (3 NF)માં હોય ત તે (1NF) અને (2 NF)માં પણ હોય છે.

- **પ્રથમ નોર્મલ ફોર્મ (1NF)** : જો તે ફ્લેટ ફાઈલનું પ્રતિનિધિત્વ કરે તો, રીલેશન એ પ્રથમ નોર્મલ ફોર્મમાં હોય મોટા છે એટલે કે રો અને કોલમના છેદબિંદુઓ વચ્ચે રીલેશન સિંગલ વેલ્યુ ધરાવે છે.

  બીજી રીતે કહીએ તો ટપલમાં દરેક એટ્રીબ્યુટ વેલ્યુ ઓટોમિક છે (i.e) અભિન્ન (non-decomposable) રીલેશન (Person) જે (inf)માં નથી તેને સમજીએ Person (નામ, ઉંમર, જાતિ (Name, Age,)

  આ રીલેશનમાં (Name, Age) મલ્ટીપલ વેલ્યુ, (નામ અને ઉંમર) એટ્રીબ્યુટ ડીપેન્ડન્ટ ચાઇલ્ડ સાથે સંબંધ ધરાવે છે. આ રીલેશનને (1NF)માં બદલવા માટે તેને નીચે પ્રમાણે બે રીલેશનમાં વિભાજીત કરવા પડે.

  (Person) (નામ, ઉંમર, જાતિ)

  (Dependent) (નામ) (Name, Age)

- **સેકન્ડ નોર્મલ ફોમ (2NF)** : સેકન્ડ નોર્મલ ફોર્મ (2NF) રીલેશન અને બીજા સામાન્ય ફોર્મમાં હોય છે, જો તે (1NF)માં અને દરેક નોન-કી એટ્રીબ્યુટ એ સંપૂર્ણ પ્રાયમરી કી પર ફંક્શનલી ડીપેન્ડન્ટ હોય સેકન્ડ નોર્મલ ફોર્મ ફક્ત સંકલિત પ્રાયમરી કી સાથે સંબંધ ધરાવે છે. જો રીલેશન (1NF)માં હોય અને તેને સીંગલ એટ્રીબ્યુટ પ્રાયમરી કી ધરાવતું હોય, તો તે સ્વયં (2NF)માં હોય છે.

સેકન્ડ નોર્મલ ફોર્મને સમજવા માટે નીચેના રીલેશનનું ઉદાહરણ લઇએ.

આ રીલેશનમાં કોર્સ Id, Courseh કોર્સ- id કોર્સ નામ, ક્લાસ નંબર, વિદ્યાર્થી id ફેકલ્ટી id.

કોર્સ નામ અને ફેકલ્ટી Id, સંકલિત પ્રાયમરી કી રચે છે. આ રીલેશનમાં નીચે બતાવેલા ફંક્શનન ડિપેન્ડન્સીમાં જાણવામાં આવે છે.

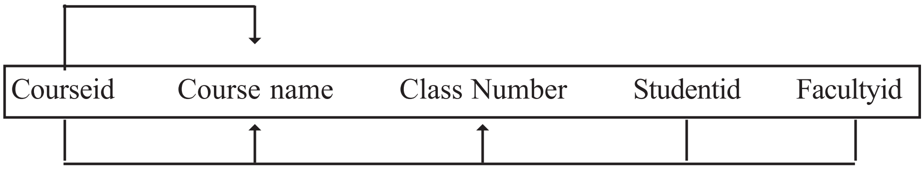
Course-id, Student-id, Faculty-id → Course Name

Course-id, Student-id, Faculty-id → Class Number

રીલેશનને સીમેન્ટીકલી તપાસતા એવું એવલોકન કરવામાં આવ્યું છે કે, કોર્સ id અનન્ય રીતે કોર્સનું નામ નક્કી કરે છે. i.e

Course-id → Course Name

આનો અર્થ એવો થયો કે કોર્સ નામ, પ્રાયમરી કી પર સંપૂર્ણ રીતે ફંક્શનલી ડિપેન્ડન્ટ નથી, તેથી રીલેશન 2NF માં નથી. રીલેશનનો ડિપેન્ડન્સી ડાયગ્રામ નીચે પ્રમાણે દર્શાવી શકાય.



આ રીલેશનને 2NF માં બદલવા માટે તેને બે રીલેશનમાં નીચે પ્રમાણે વિભાજીત કરવામાં આવે છે.

Course (Course-id, Class Number, Student-id, Faculty-id)

Course Title (Course-id Course Name)

રીલેશનમાંથી ઉદ્‌ભવતા પ્રશ્નો અને આ એટ્રીબ્યુટ દ્વારા નવા રીલેશન રચવા માટે, એટ્રીબ્યુટને નીચોવીને પૃથક્કરણ કરવામાં આવે છે.

આ રીલેશનને (સામાન્યતા અને અસામાન્યતા) આપણે અગાઉ વર્ણન કરેલી અનોમલીને આધારે તપાસીએ

Course-id	Course Name	Class Number	Student-id	Faculty-id
C701	Computer Science	7	S009	BO3
C701	Computer Science	8	S008	GO4
C702	Library Automation	7	S009	AO1
C702	Library Automation	8	S006	GO4
E500	Micro Economics	7	S009	PO2
E501	Macro Economics	7	S001	PO2
M200	Management Studies	7	S001	VO1

ઉપરના રીલેશનમાંથી પ્રાપ્ત સામાન્યતા રીલેશન

Course-id	Class Number	Student-id	Faculty-id
C701	7	S009	BO3
C701	8	S008	GO4
C702	7	S009	AO1
E500	8	S006	GO4
E501	7	S007	PO2
M200	7	S001	VO1

Course Title	
Course id	**Course Name**
C701	Computer Science
C702	Library Automation
E500	Microeconomics
E501	Microeconomics
M200	Management Studies

- **ઈન્સર્સન :** અસામાન્યતાના કેસમાં ધારો કે નવો કોર્સ નામ "કોમ્પ્યુટર નેટવર્ક"ને શરું કરવાનું છે આપણે કોર્સનું નામ રીલેશનમાં દાખલ કરી શકીએ નહીં કારણ કે આ કોર્સ ચાલુ કર્યો ત્યારે આપણી પાસે Studentid અને ફેકલ્ટી id નિશ્ચિત કરવાનું કોઈ સાધન નહોતું અને આનું આરોપણ પ્રાયમરી કીમાંથી કોર્સ આઇ.ડી. સાથે કંઈ પણ નહીં (Null) વેલ્યુ આવી શકે નહીં. આમ છતાં સામાન્યતા રીલેશનમાં Course Title આ પ્રશ્ન ઉદ્ભવતો નથી.
- **ડિલિશન :** ધારો કે અસામાન્યતા રીલેશનમાં સ્ટુડન્ટ સ્ટુડન્ટ id Soot સાથે આવો અભ્યાસક્રમ છોડી દે છે. આ વિદ્યાર્થી ચોક્કસ અભ્યાસક્રમમાં એકમાત્ર વિદ્યાર્થી હોવાથી ટપલનું ડિલીશન માહિતીને ગુમાવશે, અને આપણી પાસે એવો કોઈ રસ્તો નહીં રહે કે જેથી આપણે 'મેનેજમેન્ટ સ્ટડીઝ' કોર્સ ચાલુ છે એ જાણી શકીએ. સામાન્યતા રીલેશનમાં ટપલનું આ રદ્દ કરવું, માહિતી ગુમાવવામાં પરિણમતું નથી, કારણ કે કોર્સ નામની માહિતી સેકન્ડ રીલેશનમાં મોજુદ હોય છે.
- **અપડેટીંગ :** આપણે ધારી લઈએ કે કોર્સનું નામ 'લાયબ્રેરી ઓટોમેશન' બદલીને 'લાઈબ્રેરી મેનેજમેન્ટ' થાય છે. અસામાન્યતા રીલેશનમાં આપણે માહિતીને બે ટપલમાં બદલવી પડે. જ્યારે સામાન્યતા રીલેશનમાં ફક્ત એક જ ટપલને અપડેટ કરવાની જરૂર પડશે. આનો ફાયદો સૂચક રીતે ન દેખાય, પણ જો ડેટાબેઝની સાઇઝ મોટી હોય તો, આ પ્રશ્નોના ઘેરા પ્રત્યાઘાત પડે છે.
- **ત્રીજું સામાન્ય ફોર્મ (3NF) :** જો તે બીજા સામાન્ય ફોર્મમાં હોય અને ટ્રાન્ઝીસ્ટીવ ડિપેન્ડન્સીસ ધરાવતા ન હોય તો, તે રીલેશન ત્રીજા સામાન્ય ફોર્મમાંહોય છે. આ સામાન્ય ફોર્મ એ બહુ જ અગત્યનું નોર્મલ ફોર્મ ગણવામાં આવે છે.

3NFના ઉદાહરણ માટે નીચે પ્રમાણેના રીલેશન તપાસીઓ

Faculty (Faculty – id, Faculty – Name, Department, Gender, Salary, Office)

આપણે એ પણ માની લઈએ કે દરેક ડિપાર્ટમેન્ટને તેની ઑફિસ એક જ બિલ્ડીંગમાં છે.

આ જ પ્રકારના રીલેશનને ફેકલ્ટી id કે પ્રાયમરી કી હોવાથી ફંક્શનલ ડિપેન્ડન્સ નીચે પ્રમાણે હોય છે.

Faculty – id → Faculty – Name

Faculty – id → Department

Faculty – id → Gender

Faculty – id → Office

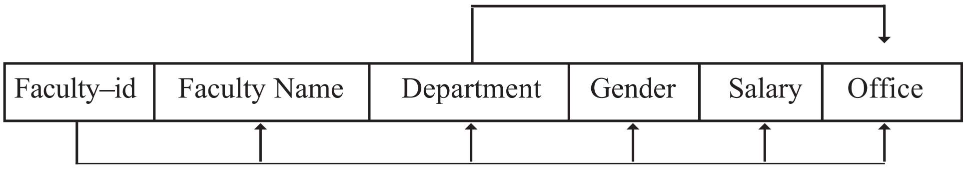
આપણે આ પ્રકારના રીલેશનને સીમેન્ટીક્સ દ્વારા પણ જાણી શકીએ

Department → Office

તેથી આપણી પાસે નીચે પ્રમાણે ટ્રાન્ઝીટીવ ડિપેન્ડન્સી છે.

Faculty – id → Department → Office

આમ, રીલેશનના ડિપેન્ડન્સીસ ડાયાગ્રામ આ પ્રમાણે બતાવી શકાય.



આ રીલેશનમાં 3NFમાં બદલવા માટે આપણે તેનું બે વિભાગમાં વિભાજન કરવું પડે, નવા રીલેશન Office – Nameને એટ્રીબ્યુટ ઑફિસ કે જે ટ્રાન્ઝીટીવ ડિપેન્ડન્સી અને તેનું ડીટરમીટંટ ઉપજાવે છે. પૃથક્કરણ રીલેશન આ પ્રમાણે છે.

Faculty (Faculty–id, Faculty–Name, Department, Gender, Salary)

Office–Name (Department, Office)

આ રીલેશનમાં થોડી ટપલ વેલ્યુ સાથે આપણે તેની અનોમલી કે જે રીલેશન ટેસ્ટ કરવા માટે ઉપયોગમાં આવે છે તેને તપાસીએ

Faculty (Unnormalised)

Faculty–id	**Faculty–Name**	**Department**	**Gender**	**Salary**	**Office**
BO3	Bindra	Computer Science	M	4000	Birla Block
GO4	Ganguly	Computer Science	M	4500	Birla Block
AO1	Arora	Computer Science	F	3450	Birla Block
PO2	Pandey	Economics	F	3500	Kanishka Block
VO1	Vohra	Mathametics	M	4500	Ashoka Block
B02	Bansal	Physics	M	3000	Ashoka Block

To Normalised Relations are

Faculty

Faculty–id	**Faculty–name**	**Department**	**Gender**	**Salary**
BO3	Bindra	Computer Science	M	4000
Go4	Ganguly	Computer Science	M	4500
AO1	Arora	Computer Science	F	3450
PO2	Pandey	Economics	F	3500
Vo1	Vohra	Mathematics	M	4500
Bo1	Bansal	Physics	M	3000

Office-Name

Department	**Office**
Computer Sciece	Birla Block
Economics	Kanishka Block
Mathematics	Ashoka Block
Physics	Asoka Block

- **ઈન્સર્સન** : આપણે માની લઈએ કે નવું ડિપાર્ટમેન્ટ, કે જેની ફેકલ્ટી ફાઈનલ થઈ નથી તેની રચના કરવામાં આવે છે આપણે આ માહિતી અસામાન્યતા રીલેશનમાં દાખલ કરી શખીએ નહીં, કારણ કે આપણે પ્રાયમરી કીની ફેકલ્ટી (id) વેલ્યુ જાણતા નથી. નવા ડીપાર્ટમેન્ટને અનુલક્ષીને ફેકલ્ટી (id)ને વેલ્યુ આપવા માટે આપણે બીજા આરોપણની વેલ્યુ જાણવી જોઈએ. (ફેકલ્ટી-નેમ, જેન્ડર, સેલરી, ઑફિસ)

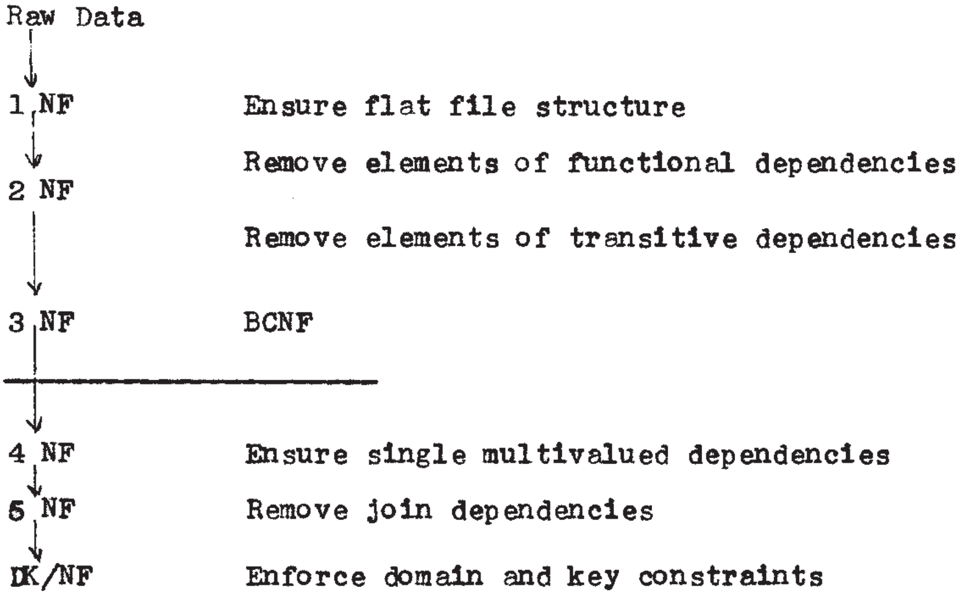
  આ પ્રશ્ન સામાન્યતા રીલેશનમાં ઉપસ્થિત નથી થતો.
- **ડિલીશન** : ધારો કે મેથેમેટીક્સ ડિપાર્ટમેન્ટના ફેકલ્ટી મેમ્બર વ્હોરા ફેકલ્ટી છોડે

છે. જો આપણે અસામાન્યતા રીલેશનમાંથી આ ફેકલ્ટી નામ સંબંધિત ટપલને ડીલીટ કરીએ, તો મેથેમેટીક્સ ડિપાર્ટમેન્ટની ચાલુ માહિતી પણ આદશ્ય થશે. આમ છતાં સામાન્યતા રીલેશનમાં આવુ બનતું નથી.

- **અપડેટ** : કોમ્પ્યુટર સાયન્સ ડિપાર્ટમેન્ટની ઑફિસ, બિરલા બ્લોકમાંથી નહેરુ બ્લોકમાં ખસેડવામાં આવે છે, તો અસામાન્યતા રીલેશનમાં ત્રણ ટપલને અપડેટ કરવાની જરૂર પડશે. જ્યારે સામાન્યતા રીલેશનમાં ફક્ત એટ ટપલને સુધારવાની જરૂર પડશે.
- **Boyce - Codd normal form (BCNF)** મૂળભૂત રીતે સામાન્ય ફોર્મ 3FN એ બંધ થાય છે. આમ છતાં ડિપેન્ડન્સીસમાંનુ સંશોધન નોર્મલ ફોર્મ પ્રતિ દોરે છે. BCNF ત્રીજા સામાન્ય ફોર્મનું વિસ્તૃતીકરણ છે. તે જણાવે છે કે જો રીલેશન 3NFમાં હોય અને બધાં જ ડીટરમિન્ટટએ કેન્ડીડેટ કી હોય તો, તે Boyce - Codd નોર્મલ ફોર્મ કહી શકાય.
- **The fourth normal form (4NF)** :

  ચતુર્થ સામાન્ય ફોર્મ : ચતુર્થ સામાન્ય ફોર્મ મલ્ટીવેલ્યુડ ડિપેન્ડન્સીસને અનુસરે છે, રીલેશન જો તેને ફક્ત એક જ મલ્ટી વેલ્યુડ ડિપેન્ડન્સીસ. હોય તો એને ચતુર્થ સામાન્ય ફોર્મ કહેવામાં આવે છે.
- (**The fifth normal form (5NF)** પંચમ સામાન્ય ફોર્મ એ પ્રોજેક્ટ જોઈન નોર્મલ ફોર્મ (PJNF) તરીકે પણ ઓળખાય છે. જો રીલેશન 5NF માં હોય તો, કોઈપણ જાતની માહિતી ગુમાવ્યા વગર, આપણે રીલેશનના પ્રોજેકશનને જોડી શકીએ અને અસલ રીલેશનની નવરચના કરી શકીએ.
- (**The Domain Key normal form (DK/ NF)** : ધ ડોમેઈન કી નોર્મલ ફોર્મ સૌથી ઉચ્ચ નોર્મલ ફોર્મ કહી શકાય. જો બધાં જ સંકોચોને સેટ ઓફ એટ્રીબ્યુટ - નામ અને તેનાં અંડર-લાઈંગ ડોમેઈનને તેની સેટ ઓફ કી સાથે માત્ર જાણીએ અને નિષ્કર્ષ કરવામાં આવે તો રીલેશન - DK/ NFમાં હોય છે. આમ DK/ NFમાં ફક્ત બે પ્રકારના નિગ્રહની છૂટ હોય છે - ડોમેઈન કોન્સ્ટ્રેઈન્ટ અને કી કોન્સ્ટ્રેઈંટ. જો આ પ્રકારના નિગ્રહનો સંપૂર્ણ અમલ કરવામાંઆવે તો, અન્ય નિગ્રહને (ડિપેન્ડન્સીસ) દૂર કરવામાં આવે છે અને રીલેશનમાં કોઈ દેખીતી ભૂલ સર્જાતી નથી.

નીચેથી ઉચ્ચ નોર્મલ ફોર્મમાં સામાન્યતાની પ્રક્રિયા આકૃતિ 3.12માં દર્શાવી છે.



આકૃતિ 3.12 : નોર્મલાઈઝેશન પ્રોસેસ

◆ **તમારી પ્રગતિ ચકાસો**

5. નોર્મલાઈઝેશન ઓફ રીલેશન્સના ફાયદા અને ગેરફાયદા કયા કયા છે ?

નોંધ (i) તમારા જવાબો નીચે આપેલ જગ્યામાં લખો.

(ii) આ પ્રકરણના અંતે આપેલા જવાબો સાથે તમારા ઉત્તરો સરખાવો

....

....

....

....

....

....

3.10 ડિઝાઈનિંગ ડેટાબેઝીસ (DESIGNING DATABASE)

ડિઝાઈનીંગ ડેટાબેઝએ એક વધુ અઘરું ઓપરેશન છે. જો કે ડેટાબેઝના નવમા ડિઝાઈનને પ્રમાણમાં નિર્દિષ્ટ કરવું સહેલું છે. છતાં શ્રેષ્ઠ ડિઝાઈન પ્રતિદોરી જતા સર્વ સામાન્ય અભિગમ અસ્તિત્વમાં નથી.

ડેટાબેઝ ડિઝાઈન પૂરતા લવચીક (Flexible) હોવી જોઈએ કે જેથી મહત્તમ વપરાશકર્તાની જરૂરિયાતને તે સમગ્રપણે પૂરી કરે. ડિઝાઈન કરવા ઉપરાંત, તેને અમુદ હદે અગાઉથી વિચારી ભવિષ્યની જરૂરિયાત અને તેને પૂરી કરવા જોગવાઈ કરવી પડે. ડેટાબેઝ ડિઝાઈનરના ભાગે આને માટેની અંતઃસ્ફૂર્ણા જરૂરી છે.

ડેટાબેઝને ડિઝાઈનીંગ કરવાથી એક અરસ -પરસ અસર કરતી પ્રક્રિયા છે. એનો અર્થ એવો થયો કે અસલ ડેટાબેઝ સ્ટ્રક્ચર તેનાં વપરાશની સાથે બદલાતું રહે છે. આમ છતાં સમયના વહેણ સાથે, ડિઝાઈન સ્થિરતા બતાવે છે. ઘણુંખરું, ડેટાબેઝ એડમિનીસ્ટ્રેટર (DBA) તરીકે નિયુક્ત વ્યક્તિ ડેટાબેઝ ડિઝાઈન અને વ્યવસ્થાપનને કાબૂમાં રાખે છે.

ડેટાબેઝ ડિઝાઈનીંગની પ્રક્રિયાનું સ્ટેપ-બાય-સ્ટેપ નીચે પ્રમાણે વિવરણ કરી શકાય.

1. સૌ પ્રથમ ડેટાબેઝમાં સચવાયેલા ડેટાને નક્કી કરવામાં આવે છે. આ માટે વપરાશકર્તા માટે ઉપયોગી માહિતીનો વિગતવાર અભ્યાસ કરવો પડે. માહિતીની જરૂરિયાતના આધારે પૃથક્કરણ અને જરૂરિયાતનું અસ્તિત્વ ચોક્કસ કરવામાં આવે અને તેનું આરોપણ તપાસવું જોઈએ.
2. કન્સેપ્ચુઅલ સ્કેમાનુ પ્રતિનિધિત્વ કરતા ડેટાબેઝનું ER મોડેલ દોરવું જોઈએ. ડેટાબેઝ ડિઝાઈનની પ્રક્રિયામાં આ સૌથી અગત્યનો તબક્કો છે. ER ડાયાગ્રામ કે જે અસ્તિત્વ અને તેના સંબંધને લાગે વળગે છે, તે જેટલું બને તેટલું સંક્ષેપમાં હોવું જોઈએ.
3. ER મોડેલને પસંદગીના ડેટાબેઝ સ્ટ્રક્ચરમાં (હાયરાર્કિકલ નેટવર્ક, રીલેશનલ અથવા અન્ય કોઈ મોડેલ) નકશીત (Mapped) કરવા જોઈએ. જો રીલેશનલ મોડેલને પસંદ કરવામાં આવે તો, ટેબલને એન્ટીટી અને તેનાં રીલેશન્સને આખરી રૂપ આપવું જોઈએ. સામાન્યતાની પ્રક્રિયા, ટેબલને ચકાસવા અને તેને જરૂર હોય તો પુનઃ આકાર આપવા માટે આવાહન કરવું જોઈએ.
4. ખાલી ડેટાબેઝ DBMS ક્રમાંકનો ઉપયોગ કરીને રચવામાં આવે છે. (Create TABLE INDEX etc.) ડેટા ડિક્ષનરી કે જે ડેટા આટમ નામ અને તેમનાં ઈન્ટરનલ સ્ટોર્ડ ફોર્મેટને પણ DBMS દ્વારા રચવામાં આવે છે. ડેટાબેઝ ડિઝાઈન સ્ટેપ 3 સ્ટેપ ઈન્ડિપેન્ડન્ટને પ્રોસેસ કરે છે.
5. ડેટાબેઝ ઘનિષ્ઠ છે. આ ખાલી ડેટાબેઝ ડેટા કે જે ઈન્સર્ટ કરવાના છે તે મશીન રીડેબલ ફોર્મમાં પ્રાપ્ય છે. DBMSના ડેટા લોડીંગ યુટીલીટી નો ઉપોગ થઈ શકે.
6. ડેટાબેઝની કામગીરીને ટ્યુનીંગની જરૂર છે કે કેમ તે નક્કી કરવા કાળજીપૂર્વક મોનિટરીંગ કરવામાં આવે છે. ડેટાબેઝની લવચીકતા અને ગતિનું બારીકાઈથી મૂલ્યાંકન કરવામાં આવે છે. ડેટા મેનેન્ટેનન્સના પ્રશ્નો માટે ડેટાબેઝને પણ તપાસવામાં આવે છે.

7. ડેટાબેઝની કામગીરી ઉપર વપરાશકર્તાના પ્રત્યાઘાતોનું પણ અવલોકન કરવામાં આવે છે. અને મહત્તમ વપરાશ માટે સ્ટ્રક્ચરને બદલવામાં આવે છે.

3.11 ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ સિસ્ટમ (DISTRIBUTED DATABASE SYSTEMS)

3.11.1 ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝની રચના (Architecture of Distributed Database)

વિભાજિત ડેટાબેઝ એ સીંગલ લોજીકલ ડેટાબેઝ છે અને તે સુગંધીત છે. તેની સુગંધ કોમ્પ્યુટરના વિવિધ સ્થળોએ ફેલાયેલી છે, કે જે ડેટાને ઘનિષ્ઠ રીતે મેળવવા માટે કોમ્યુનિકેશન નેટવર્ક દ્વારા ઈન્ટરલીંક થયેલ છે. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ વાતાવરણ ભાગીદારી કરવાનું જરૂરી સમજે છે, ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ, ભૌગોલિક ડેટા સ્વતંત્ર રીતે આપે છે. એટલે કે ડેટાની વિનંતી કરતા વપરાશકર્તાને ડેટા કયા સ્થળે આવેલા છે તે જાણવાની જરૂર નથી. પ્રક્રિયા મોટેભાગે લોકેશન ટ્રાન્સપરન્સી તરીકે ઓળખાય છે અને દરેક લેવલ સાઈટને નોડ (node) કહેવામાં આવે છે. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝનું આર્કિટેકચર અને સ્કીમેટીકનું પ્રતિનિધિત્વ આકૃતિ 3.13માં દર્શાવ્યું છે.

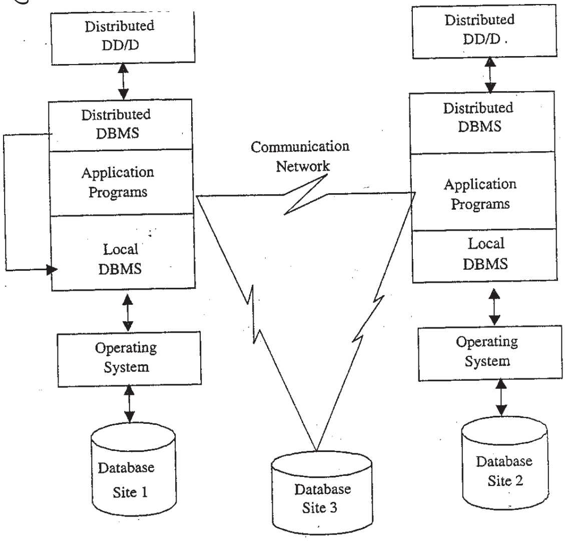


Fig. 3.13: Architecture and Schematic Representation of a Distributed Database

આકૃતિ 3.13 પરથી સ્પષ્ટ થાય છે કે સાઈટને લોકલ DBMS અને ડિસ્ટ્રીબ્યુટેડ DBMS ની કોપી પણ હોય છે. ડિસ્ટ્રીબ્યુટેડ ડેટા ડિક્ષનરી / ડિરેકટરી (DD /D) નેટવર્કમાં ડેટાના લોકેશનમાં અને સાથે ડેટા ડેફિનેશનમાં, માહિતી સ્ટોર કરે છે. જરૂરી ડેટાને શોધવા માટે ડિસ્ટ્રીબ્યુટેડ ડેટા ડિક્ષનરી / ડિરેકટરીમાંથી ડેટા માટેની વિનંતી પ્રથમ ચેક કરવામાં આવે છે જો ડેટા લોકલ સાઈટમાં પ્રાપ્ય હોય તો ડિસ્ટ્રીબ્યુટ DBMS વિનંતીને લોકલ DBMS તરફ મોકલે છે. જો વિનંતી ડેટાને બીજી સાઈટમાંથી મેળવવાની હોય તો ડિસ્ટ્રીબ્યુટેડ DBMS આ વિનંતીને આ સાઈટ તરફ લઈ જાય છે.

જયારે ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝમાં વિવિધ નોડ NODSને મિક્ષ DBMSહોય. દા.ત. નોડ - 1ને રીલેશનલ DBMS હોઈ શકે, અને નોડ - 2ને નેટવર્ક DBMS, તો પછી આ વાતાવરણને હલ કરવા શક્તિમાન ડિસ્ટ્રીબ્યુટેડ DBMS, એ હેટરજીનિયસ ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ કહી શકાય.

ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ મેનેજમેન્ટ, કેન્દ્રિત અને વિકેન્દ્રિત પ્રોસેસીંગના બધાં જ લાભોનો ઉપયોગ કરે છે. વિકેન્દ્રિત ડેટાબેઝ જેવા કે ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ પણ અન્ય કોમ્પ્યુટરમાં મલ્ટીપલ લોકેશનમાં સ્ટોર કરે છે. પણ આ કેસમાં, કોમ્પ્યુટર ઈન્ટરકનેકટેડ નથી અને તેથી ડેટાની ભાગીદારી કરી શકાય નહીં.

3.11.2 જસ્ટીફિકેશન એન્ડ ઓપરેશન ફોર ડિસ્ટ્રીબ્યુટીંગ ડેટા (Justifications and Options for Distributed Database)

- **ડિસ્ટ્રીબ્યુટીંગ ડેટા માટેનું સમર્થન અને વિકલ્પ :** ડિસ્ટ્રીબ્યુશન ડેટા માટેના સમર્થનને આ પ્રમાણે સંક્ષેપમાં કહી શકાય. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ વધારાનું સમર્થન અને પ્રાપ્યતા આપે છે. કેન્દ્રિકરણ સિસ્ટમની સરખામણીએ, કે જે ફેઈલ જવાથી સર્વ વપરાશકર્તાઓને અપ્રાપ્ય થાય છે. ડિસ્ટ્રીબ્યુટેડ સિસ્ટમની કામગીરી નોડ ફેઈલ જાય તો પણ, ચાલુ રાખસે, અલબત્ત ઓછા લેવલે.

- વિવિધ સાઈટમાં ડેટાને બદલે લોકલ કંટ્રોલમાં ઉત્તેજન આપીને ડેટાની પ્રામાણિકતામાં સુધારો થાય છે અને ડેટાનો વહીવટ સરળ બને છે.
- લોકલ ડેટાને, લોકલી સ્ટોર કરવામાં આવે તો, ડેટાનું ડિસ્ટ્રીબ્યુશન વધારાના સમયમાં સુધારો કરી શકે. ડેટાને તેના ઉપયોગના પોઈન્ટની નજીક લોકેટ કરવામાં આવે, તો સંદેશાવ્યવહારનો ખર્ચ ઘટાડી શકાય અને પ્રશ્નોત્તર (ક્વેરી)નો સમય સુધરી શકે.
- ડિસ્ટ્રીબ્યુટેડ સિસ્ટમ મોડ્યુલર ગ્રોથને સગવડ આપે છે. વધારાના ડેટાબેઝનું આયોજન કરનાર નવા નોડ, સિસ્ટમમાં અવશેષોને ઉમેરી શકે.

ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝમાં ડેટાને ડિસ્ટ્રીબ્યુટેડ કરવા માટે ઘણાં વિકલ્પો પ્રાપ્ત છે. આ વિકલ્પોમાં સમાવેશ થાય છે -

1. ડેટા એપ્લીકેશન
2. હોરીજેન્ટલ પાર્ટીશનિંગ
3. વર્ટીકલ પાર્ટીશનિંગ અને
4. ઉપરોક્તનું જોડાણ.

ડેટા પ્રતિકૃતિના (રેપ્લીકેશન) કેસમાં ડેટાબેઝની કૉપી કેટલીક સાઈટ અથવા તો બધી જ સાઈટમાં સ્ટોર કરવામાં આવે છે. વિશ્વાસ પાત્રતા, ટેલી કૉમ્યુનિકેશન ચાર્જીસમાં બચત અને ઝડપી પ્રતિભાવ, એ આ વિકલ્પનો ફાયદો છે. પણ વધારાના સ્ટોરેજની જરૂરિયાત અને પ્રસારને અપડેટ કરવાની મુશ્કેલી એ તેની મૂળભૂત મર્યાદા છે. આ વિકલ્પ અપડેટ કરવાની જરૂરિયાત ક્વચિત જ હોય અને ડેટાબેઝનું આદાન- પ્રદાન રીડ ઓન્લી પૂરતું મર્યાદિત હોય. (CD- ROM Compact disk read only memury) પ્રતિકૃતિ ડેટાબેઝ માટે શ્રેષ્ઠ માધ્યમ પૂરું પાડે છે.

- **હોરિઝન્ટલ પાર્ટીશનિંગ** : ડેટાબેઝનું હોરિઝેન્ટલ પાર્ટીશનિંગ રીલેશનની રો ને મલ્ટીપલ સાઈટમાં વિભાજિત કરવા માટે ડેટાબેઝનું હોરિઝેન્ટલ પાર્ટીશનીંગ ભાગ ભજવે છે. હોરિઝેન્ટલ પાર્ટીશનીંગ ડેટા બેઝના અવશેષોને (FRAGMENT) જ્યાં તેનો મહત્તમ વપરાશ થાય છે તે સાઈટમાં, કાર્યદક્ષતા વધારી શકે.

- **વર્ટીકલ પાર્ટીશનીંગ** : ડેટાબેઝનું વર્ટીકલ પાર્ટીશનીંગ રીલેશનના પસંદ કરેલા કોલમને જે જુદી જુદી સાઈટમાં સ્ટોર કરેલા છે, તે નવી રીલેશનમાં યોજવામાં આવે છે. વર્ટીકલ પાર્ટીશનીંગનો મુખ્ય માનદંડ એ વ્યક્તિગત સાઈટ પર ચોક્કસ ડેટા આઈટમની જરૂરિયાત છે.

ડેટા ડિસ્ટ્રીબ્યુશનના ઉપરોક્ત સંયુક્ત વિકલ્પો, ડિસ્ટ્રીબ્યુશન સિસ્ટમની જરૂરિયાત આધારિત ઉપયોગ કરી શકાય. મૂળભૂત સિદ્ધાંત કે જેને સૌએ યાદ કરવા જેવો છે, તે ડેટા સાઈટમાં સ્ટોર કરેલો હોવો જોઈએ કે જ્યાં તેનો ઉપયોગ વારંવાર થતો હોય.

3.12 મેનેજમેન્ટ આધારિત ડેટાબેઝ સિસ્ટમ (DATABASE SYSTEMS FOR MANAGEMENT SUPPORT)

મેનેજમેન્ટ આધારિત ડેટાબેઝ સિસ્ટમ વ્યાપક રીતે મેનેજમેન્ટ ઈન્ફોર્મેશન સિસ્ટમ (MIS) તરીકે વર્ણવવામાં આવે છે. આમ છતાં મેનેજમેન્ટની વિવિધ લેવલ માટે મેનેજમેન્ટ સંચાલન અને અપેક્ષિત આઉટપુટ માટે ડેટાબેઝ સિસ્ટમનું વિભાજન કરવામાં આવ્યું છે. આકૃતિ 3.14 મેનેજમેન્ટની ત્રણ લેવલને સાદશ્ય કરતા ઈન્ફોર્મેશન સિસ્ટમના હાયરાર્કિનું ઉદાહરણ છે.

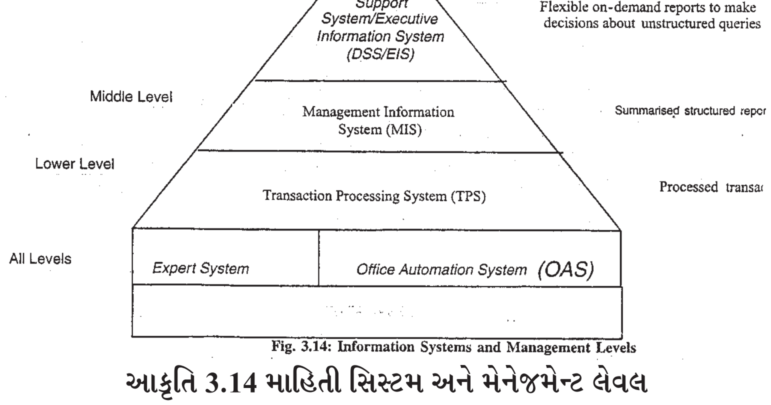


Fig. 3.14: Information Systems and Management Levels

આકૃતિ 3.14 માહિતી સિસ્ટમ અને મેનેજમેન્ટ લેવલ

આકૃતિમાં દર્શાવ્યા પ્રમાણે લોઅર લેવલમાં મેનેજર માટે ટ્રાન્ઝેકશન પ્રોસેસીંગ સિસ્ટમ (TPS) પ્રોસેસ્ડ ટ્રાન્ઝેકશન (બીલ, ઓર્ડર વિગેરે) માટે પર્યાપ્ત છે. મિડલ ઓર્ડર મેનેજરને સંક્ષેપમાં સ્ટ્રકચર રિપોર્ટ મેનેજમેન્ટ સિસ્ટમની જરૂર પડે છે. ઉચ્ચ કક્ષાએ ડિસિઝન સપોર્ટ સિસ્ટમ (DSS) અથવા એક્ઝીક્યુટીવ ઈન્ફોર્મેશન સિસ્ટમ (EIS) કે જે અનેક સ્ટ્રકચર ક્વેરી માટે બ્રીફ ઓન ડિમાન્ડ રિપોર્ટ આપવા શક્તિમાન હોય. ઑફિસ ઓટોમેશન સિસ્ટમ અને એક્સપર્ટ સિસ્ટમનો ઉપયોગ નોન - મેનેજમેન્ટ સહિત બધી જ કક્ષા ઉપયોગ કરે છે.

◆ **તમારી પ્રગતિ ચકાસો**

6. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝના ફાયદા કયા કયા?

નોંધ (i) તમારા જવાબો નીચે આપેલ જગ્યામાં લખો.

(ii) આ પ્રકરણના અંતે આપેલા જવાબો સાથે તમારા ઉત્તરો સરખાવો

....

....

....

....

....

....

3.13 કૃત્રિમ સમજશક્તિ અને નિષ્ણાત સિસ્ટમ (ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS)

આર્ટીફિશિયલ ઈન્ટેલીજન્સ (AI) એ ટેકનોલોજીને સ્પર્શતુ ગૃપ છે, જે એવા યંત્રને વિકસાવે છે કે જે માનવ જેવી લાયકાતને - જેવી કે અભ્યાસ કરવો, વિચારશક્તિ, અરસપરસ વાતચીત કરવી, જોવું, સાંભળવું વગેરેનું અનુકરણ કરે.

તેના અગાઉના વિકાસમાં કોમ્પ્યુટરને "ઈલેક્ટ્રોનિક બ્રેઈન" કહેવામાં આવતુ અને ત્યારથી માનવ-બ્રેઈન સાથે તેના જેવું સશક્ત અને તેની બરોબરીને પહોંચી વળવા મટે પ્રયત્નો થઈ રહ્યા છે.

આમ છતાં કોમ્પ્યુટરના અને માનવ-બ્રેઈન વચ્ચે મૂળ તફાવત તેના સંચાલનમાં છે, કોમ્પ્યુટર અંકની પ્રક્રિયા કરે છે, જ્યારે માનવ-બ્રેઈન સંજ્ઞા પર કામ કરે છે. પરંપરાગત કોમ્પ્યુટર પ્રોગ્રામ અલગોરીધમનો ઉપયોગ કરે છે. એટલે કે પ્રશ્નને ઉકેલવા માટે સ્પષ્ટ રીતે સ્ટેપ - બાય -સ્ટેપ પ્રથાને અનુસરે છે, જ્યારે માનવની વિચાર પ્રક્રિયા એ અલગોરીધમ નથી અને તે સંજ્ઞાની પ્રક્રિયા આધારિત છે. માનવ-બુદ્ધિશક્તિ અથવા વિચારોની પ્રક્રિયા માનવ સંવેદનાની રીત ઉપર આધારિત છે.

માનવ વિચારની પ્રક્રિયાને નીચે દર્શાવેલી (હ્યુરીસ્ટીકસ ઓફ હ્યુમનકોર્ટ પ્રોસેસ) આકૃતિ 3.15માં દર્શાવી છે.

Human decisions
↓
Based on
Judgements
↓
Based on
Reasoning
↓ ↓ ↓
Learned facts | Experience | Human

3.15 : Heuristics of Human Thought Procese

◆ **The main areas of AI are** :

- Robotics
- Perception Systems

- Fuzzy logic
- Neural networks
- Genetic algorithm
- Natural language processing
- Expert systems

રીબોટિક્સ એ ફિલ્ડ છે, જે એવું મશીન વિકાસ કરવાનો પ્રયત્ન કરે છે કે જે લોકો દ્વારા સામાન્ય રીતે કરાતા કાર્યો કરી શકે. તે મશીન એ જ માનવ-યંત્ર કહેવાય. ઈન્દ્રિયજન્ય સિસ્ટમ એવા સાધનોની રચના કરવા માગે છે, કે જે માનવશક્તિની - જોવાની, શ્રવણની, સ્પર્શ અને સુંઘવાની ક્રિયાની બરોબરી કરી શકે. દેખીતી રીતે ઈન્દ્રિયજન્ય સિસ્ટમ રોબોટિક્સ સાથે જોડાયેલી છે, કારણ કે રોબોટને સમજશક્તિની જરૂર પડે છે. ફઝી (ધૂંધળુ) લોજિક એ અવહેવારુ ડેટા અને અનિશ્ચિતતાને નોતરે છે. કે જેને એક પ્રશ્નના એકને બદલે ઘણાં ઉત્તરો હોય છે. ફઝી લોજિક એ ક્લાસિકલ લોજીકની જેમ નહીં પણ માનવ વિચારશક્તિની જેવું છે. તે શક્યતાઓ અને વિશ્વાસપાત્રતા સાથે વ્યવહાર કરે છે એટલે કે ખરું કે ખોટું કહેવાને બદલે અનુમાન કરવામાં આવે કે મોટેભાગે ખરું અથવા મોટેભાગે ખોટું અથવા વધારે ખરું અથવા વધારે ખોટું.

ફઝી લોજીક સિદ્ધાંતોને ન્યૂરલ નેટવર્કમાં ઉપયોગ કરવામાં આવે છે. ન્યૂરલ નેટવર્ક ફિઝીકલ ઈલક્ટ્રોનિક ઉપકરણો અથવા સોફ્ટવેરનો હ્યુમન બ્રેઈનના ન્યૂરોલોજીકલ સ્ટ્રકચરની નકલ કરવા ઉપયોગ કરે છે. જેનેટિક અલગોરીધમ પ્રોગ્રામને અનુસરે છે કે જે રેન્ડમ પરિવર્તનના દ્વાવીડીઅન સિદ્ધાંતોને પોતાની જાતને સુધારવા માટે ઉપયોગ કરે છે.

નેચરલ લેંગવેજ પ્રોસેસીંગમાં વપરાશકર્તા સ્ટ્રકચરલ કમાંડ લેંગેવજને બદલે રોજબરોજ ઉપયોગમાં આવતી ભાષાનો ઉપયોગ કરે છે. નેચરલ લેંગવેજ સિસ્ટમ એ ઈન્ટરફેસ છે જે નેચરલ લેંગેવજ ઈન્ટરફેસ તરીકે ઓળખાય છે, અને જે નેટરલ લેંગવેજને એપ્લીકેશન લેંગવેજ માં (i.e. SQL) અનુવાદ કરે છે. લેંગવેજ સરળ રીતે સમજી શકાય તે માટે બે પર્યાયો છે -

દા.ત. SYNTAX અને SEMANTICS કે જે બહુ જ અગત્યનો ભાગ ભજવે છે. સીન્ટેક્ષ એ શબ્દને જોડવાનું કામ કરે છે, કે જેથી વાક્યના અંશ, ઉપવાક્ય અને વાક્યની રચના થઈ શકે. જ્યારે સીમેન્ટીક્સ શબ્દાર્થને અને જે રીતે વિચારને વ્યક્ત કરવામાં આવે છે તેનું કામ કરે છે. નેચરલ લેંગવેજ ઈન્ટરફેસ એ સીન્ટેક્ષ અને સીમેન્ટીક્સનું પૃથક્કરણ કરવા માટે શક્તિમાન છે. આમ છતાં નેચરલ લેંગવેજ સિસ્ટમ સાથે ઘણી મર્યાદાઓ છે, કે જે તેનાં વ્યાપક ઉપયોગ માટે અડચણરૂપ છે. સમય જતાં સિસ્ટમ એ વધારે સ્વચ્છ અને લોકપ્રિય બનશે.

ઍક્સપર્ટ સિસ્ટમને A. I પ્રોગ્રામમાં કે જેને ચોક્કસ વિસ્તારમાં માનવ નિપૂણતા (હ્યુમન એક્સપરટાઈઝ) ટીપીકલ એક્સપર્ટ સિસ્ટમ એપ્લીકેશન એરીઆમાં નિદાન, પ્લાનીંગ, સૂચના અને સંચાલન, મોનીટરીંગ અને ડિઝાઈનનો સમાવેશ થાય છે.

અક્સપર્ટ સિસ્ટમના હાર્દમાં છે - નોલેજ બેઝ કે જે વસ્તુસ્થિતિ (બેઝીક ડેટા), નિયમો અને કોઈ એક ચોક્કસ વિસ્તારની અન્ય માહિતી, વસ્તુ સ્થિતિ અમુક ફિલ્ડમાં પ્રાપ્ત કરેલ જ્ઞાન (સામાન્ય રીતે નિષ્ણાંતના તાબામાં હોય છે) અને નિયમો - I F સંગ્રહનું પ્રતિનિધિત્વ કરે છે. THEN સ્ટેટમેન્ટ નિષ્કર્ષ મેળવવા માટે “રૂલ્સ ઓફ થંભ” પણ કહેવામાં આવે છે. (Heuristics) જો આ-કે - તે સાચું હોય તો THEN માને છે કે અમુક - તમુક સાચું છે.

ડેટાબેઝની જેમ, જે નોલેજ આધારિત સ્પષ્ટ માહિતી તેના IF - THEN નિયમ સાથે બેઝીક ડેટામાં વધારાની માહિતી ડાયરેકટલી સ્ટોર કરવામાં આવતી નથી. LISP (લીપ્સ પ્રક્રિયા) અને PROLOG (પ્રોગ્રામીગ ઈન લોજીક) એ નોલેજ બેઝીસ દ્વારા સામાન્ય રીતે ઉપયોગમાં આવતી લેંગવેજ છે.

સમગ્ર પણે એક્સપર્ટ સિસ્ટમનો વિકાસ કરવાની પ્રક્રિયાને નોલેજ એન્જિનિયરીંગ કહેવામાં આવે છે.

હ્યુમન નોલેજને (માનવજ્ઞાન) કોમ્પ્યુટર નોલેજમાં સમાવવા માટે, નોલેજના નેચરને સમજવો જોઈએ. નોલેજનું ગૃપ આ પ્રમાણે કશરી શકાય. પ્રોસીજરલ નોલેજ, ડિકલેરેટીવ નોલેજ, સીમેન્ટીક નોલેજ અને એપીસોડીક નોલેજ. પ્રોસીજરલ નોલેજમાં કુશળતાનો સમાવેશ થાય છે. પ્રોસીજરલ નોલેજ વસ્તુ કેવી રીતે કરવી તેને સંબંધિત છે. ડિકલેરેટીવ નોલેજ માહિતી સંબંધિત છે કે જે શાબ્દિક અથવા વાચા દ્વારા વિશ્વ વિષેની ઘટનાનું બયાન કરે છે. સીમેન્ટીક નોલેજ એ ડીપ - લેવલ નોલેજ છે, કે જે કોગ્નીટીવ સ્ટ્રકચરને, ઓર્ગેનાઈઝેશનને અને રેપ્રીઝેન્ટેશનને પરાવર્તીત કરે છે. એપિસોડિક નોલેજ, સંકુલીત કરવામાં આવેલી માહિતીને આવરી લે છે. તે નિષ્ણાંતના અનુભવ દ્વારા માહિતીનું પ્રદાન કરે છે. એક્સપર્ટ સિસ્ટમ ત્રણ મૂળભૂત ઘટકો દ્વારા રચવામાં આવે છે. નોલેજબેઝ, એન્જિનનું અનુમાન અને વપરાશકર્તા (INTERFACE)નું નિષ્ણાંત કે જેને નોલેજ એન્જિનિયર કહેવામાં આવે છે. તે સાંકેતિક લિપિમાં નોલેજબેઝમાં મોકલે છે.નિષ્ણાંત એન્જિનમાં નોલેજબેઝમાં નિયમો કેવી રીતે કાબુમાં લેવામાં આવે છે તે પ્રોગ્રામનો ઉપયોગ અથવા પ્રક્રિયા કરવામાં આવે છે. વપરાશકર્તા માધ્યમ સગવડો, વાતચીતનો વહેવાર અથવા એક્સપર્ટ સિસ્ટમ અને છેવટના વપરાશકર્તા વચ્ચેનું સંકલન કરે છે.

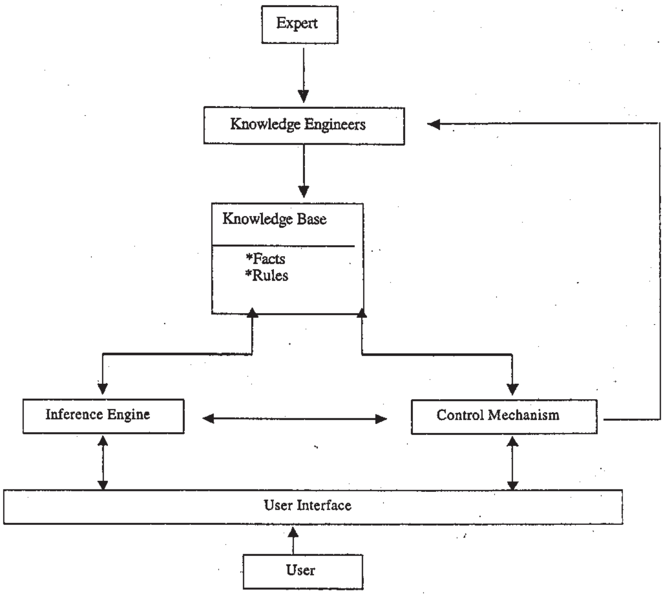


Fig. 3.16: Components of an Expert System

એક્સપર્ટ સિસ્ટમના ઘટકોની રૂપરેખા આકૃતિ 3.16માં દર્શાવી છે.

વપરાશકર્તા દ્વારા ચોક્કસ માહિતી મેળવવા કરવામાં આવેલી વિનંતી માટે સિસ્ટમ, ડેટા અને નિયમોની નોલેજબેઝમાં શોધ કરે છે. ઈન્ફરન્સ એન્જિન નક્કી કરે છે કે કયા નિયમોનો અને કેવી રીતે અમલ કરવો, તે ઉપરાંત પ્રશ્નોત્તર પ્રક્રિયાને ક્યારે પૂર્ણ કરવી અને પ્રત્યુત્તર આપવો તે નક્કી કરે છે. પ્રશ્નોત્તરને (ક્વેરી) ઝડપથી સિસ્ટમ રીસ્પોન્ડને અસરકારક રીતે મદદ કરવા માટે યુઝર ઈન્ટરફેસ વધારે ઈન્પુટ આપવાની ઝડપ કરે છે.

એક્સપર્ટ સિસ્ટમનો ઉપયોગ ઝડપથી વધી રહ્યો છે. આમ છતાં માનવ-નિપુણતા, નોકરીની દૃષ્ટિએ એક્સપર્ટ સિસ્ટમ મર્યાદામાં રહેવાની શક્યતા છે.

◆ **તમારી પ્રગતિ ચકાસો**

7. નોલેજબેઝ અને ડેટાબેઝ વચ્ચે શું તફાવત છે ?

8. આર્ટીફિશિયલ ઈન્ટેલિજન્સ (A I) શું છે ? તેના પ્રયોગના (એપ્લીકેશન) કેટલાક ક્ષેત્રોનું (એરિયા) લીસ્ટ આપો.

નોંધ (i) તમારા જવાબો નીચે આપેલ જગ્યામાં લખો.

(ii) આ પ્રકરણના અંતે આપેલા જવાબો સાથે તમારા ઉત્તરો સરખાવો.

....

....

....

....

....

....

3.14 સારાંશ (SUMMARY)

આ પ્રકરણ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમની કેટલીક મૂળભૂત વિગતોની ચર્ચા કરે છે અને આ સિસ્ટમની કામગીરીને સમજી શકવા માટે જરૂરી પાર્શ્વભૂમિકા પૂરી પાડે છે.

ડેટાબેઝ આર્કિટેકચર (ત્રણ પડ) એ ડેટાબેઝ સિસ્ટમમાં સ્કેમા લેવલને તાદશ કરવા માટે હાથવગુ સાધન છે. આ આર્કિટેકચર ડેટા ઈન્ડિપેન્ડન્સ - ફિઝીકલ અને લોજિકલ બન્ને મેળવવા માટે સરળતા કરે છે.

(E-R) મોડેલકન્સેપ્યુઅલ ડેટાબેઝ ડિઝાઈનમાં અગત્યનો ભાગ ભજવે છે. હાયરાર્કિકલ મોડેલમાંથી રીલેશનશીપ અને તેમાંથી ઓબ્જેક્ટ ઓરિઓન્ટેડ મોડેલ તરફનો ડેટાબેઝનો વિકાસ પંથના ડેટાબેઝ ડિઝાઈન ટેક્નિકમાં અસાધારણ ફેરફાર થયો છે. રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ અત્યાર સુધી ખૂબ લોકપ્રિય છે. ડેટાબેઝમાંથી અનમોલી (દેખીતીભૂલ)સુધારવાની દૃષ્ટિએ રીલેશનની સામાન્યતા એ ડેટાબેઝ ડિઝાઈનનો અગત્યનો અભિગમ છે. તે ડેટાની પ્રામાણિકતા અને એકરૂપતાને સુધારે છે, જો કે તે પુનઃપ્રાપ્તિ માટેની ગતિને ધીમી કરે છે. ડિઝાઈનીંગ ડેટાબેઝ એ બહુજ જટિલ પ્રક્રિયા છે. વપરાશકર્તાઓના ફીડબેકને આધારે સામાન્ય રીતે ડેટાબેઝ સમયાતંરે ડિઝાઈનમાં સ્થિર છે.

ડેટાબેઝ ટેકનોલોજી ખૂબ ઝડપથી પ્રગતિ કરે છે અને ડેટાબેઝ સિસ્ટમની વિવિધતા વર્તમાન વર્ષોમાં છવાઈ ગઈ છે. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝ સિસ્ટમ જે ભૌગોલિક ડેટા ઈન્ડિપેન્ડન્સને પ્રોવાઈડ કરે છે તે વપરાશકર્તાને પારદર્શક હોય તેવા ઘણાં લોકેશન્સ ડિસ્ટ્રીબ્યુટ કરવા માટે વિસ્તીર્ણ ડેટાબેઝીસને પરવાનગી આપે છે. એક્સપર્ટ સિસ્ટમ અને નોલેજ બેઝીસ સાથે નોલેજ એન્જિનીયરીંગ એ સોસીઓ - ઈકોનોમીક વિકાસને કેટલાઈઝીંગ માટે નવું સાધન ઉપલબ્ધ કરે છે. પ્રકાશમાં આવતી માહિતી સોસાયટીની જરૂરિયાતોને પહોંચી વળવા માટે ડેટાબેઝ ટેકનોલોજીની પ્રગતિ પર ગંભીર રીતે આધાર રાખે છે.

3.15 તમારી પ્રગતિ ચકાસોના ઉત્તરો (ANSWERS TO SELF CHECK EXERCISE)

1. ડેટાબેઝ સિસ્ટમમાં ડેટા ઈન્ડિપેન્ડન્સ એ અગત્યનો વિચાર છે. તે ડેટાને જે પ્રોગ્રામનો ઉપયોગ કરે છે, તેમાંથી છૂટા પાડવાનું કામ કરે છે. ડેટા ઈન્ડિપેન્ડન્સ, પ્રોગ્રામમાં ફેરફાર કર્યા સિવાય ડેટા ડેફીનેશનને બદલવાનું કામ કરે છે.

   ડેટા ઈન્ડિપેન્ડન્સ ફિઝીકલ, લોજીકલ અથવા ભૌગોલિક (ડિસ્ટ્રીબ્યુટીવ) હોઈ શકે. ફિઝીકલ ઈન્ડિપેન્ડન્સ એટલે ડેટા એ જ રીતે સ્ટોર કરવામાં આવ્યા હોય અથવા સિસ્ટમમાં મેળવવાના

હોય તેને એપ્લીકેશન પ્રોગ્રામમાં ફેરફારની જરૂરિયાત વગર બદલવાની રીત. લોજીકલ ઈન્ડિપેન્ડન્સ એટલે ડેટાબેઝને પુનર્ગઠન કરવું. (i .e) કન્સેપ્યુઅલ લેવલમાં ફેરફાર કરી શકાય, પરંતુ એ જ એપ્લીકેશન પ્રોગ્રામનો ઉપયોગ કરી શકાય. ભૌગોલિક ઈન્ડિપેન્ડન્સ સૂચવે છે કે એપ્લીકેશન પ્રોગ્રામ ડેટાના લોકેશનને લીધે અસર થતી નથી. એટલે કે ડેટા લોકલ ડિસ્કમાં અથવા રીમોટ ફાઈલ સરવરમાં સચવાયેલા છે. ડેટાબેઝ સિસ્ટમમાં ડેટાને યોજવા માટે ડેટા ઈન્ડિપેન્ડન્સ ઘણી લવચીકતા બક્ષે છે.

2. સ્કીમા એ સમગ્રતયા ડેટાબેઝમાં ડેટાનો કન્સેપ્યુએલ અથવા લોજીકલ વ્યૂ છે. ડેટાબેઝમાં બધાં જ ટેબલનો સંગ્રહ ધરાવે છે. સ્કીમા બધાં જ ફિલ્ડનું સિસ્ટ રાખે છે. - ફિલ્ડ ટાઈપ અને દરેક ફિલ્ડ માટે મહત્તમ અને લઘુત્તમ સ્વીકાર્ય વેલ્યુ ડેટાબેઝના દરેક રો ના સ્ટ્રકચરની માહિતી સહિત.

3. ER ડાયાગ્રામ એ એક સાધન છે, કે જે ડેટાબેઝમાં આવેલા અસ્તિત્વ રીલેશનશીપને મોડેલ કરે છે. ER ડાયાગ્રામ કન્સેપ્યુઅલ સ્કીમાને યોજે છે અને ડેટાબેઝ ડિઝાઈનિંગ માટે બ્લ્યૂ પ્રિન્ટ તરીકે કામ કરે છે.

4. રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ RDBMS ડેટાબેઝ મેનેજમેન્ટમાં ડિફેક્ટો ઈન્ટરનેશનલ સ્ટેન્ડર્ડ બન્યું છે. ઓબ્જેક્ટ ઓરિએન્ટેડ સિસ્ટમ અને બીજી સિસ્ટમમાં પ્રગતિ થઈ હોવા છતાં, રીલેશનલ સિસ્ટમે તેની બહોળી સ્વીકૃતિ જાળવી રાખી છે.

   RDBMSને અન્ય ડેટામોડેલના પ્રમાણમાં વધારે લાભ છે, કારણ કે રીલેશનલ મોડેલ એ રીલેશન્સની અતિ વિકસિત મેથેમેટીકલ થિયરી પર આધારિત છે કે જયાંથી આ નામ તેણે મેળવ્યું છે - મેથેમેટિક્સનું એપ્લીકેશન રીલેશનલ મોડેલને ઘણું બળ પૂરું પાડે છે. રીલેશનલ સિસ્ટમમાં ડેટા વપરાશકર્તા સરળતાથી ઉપયોગ કરી શકે તેવા ટેબલના ફોર્મનું પ્રતિનિધિત્વ કરે છે. RDBMS ના ઉદાહરણ છે - ORACLE, SYBASE અને INGRESS.

5. રીલેશનની સામાન્યતાનો ફાયદો એ છે કે તે ડેટા ઈન્ટીગ્રીટીનો અમલ કરાવે છે અને ડેટાબેઝમાંથી વિસંગતતાને દૂર કરે છે. તે ડેટા અતિરેકતા (રીડન્સીસ)ને પણ ઓછી કરે છે અને ડેટાને ડેટાબેઝમાંથી ચોકસાઈ અને એકરૂપતાને વેગ આપે છે.

   સામાન્ય રીતે પ્રક્રિયા ટેબલને બે અથવા વધારે ટેબલમાં વિભાજીત કરવાનું કામ કરે છે, કે જેનું ડેટાનું પુનઃપ્રાપ્ત કરવા માટે જોડાણ કરવામાં આવ્યું છે. તે પુનઃ પ્રાપ્તિ માટેની ગતિ ઉપર વિપરીત અસર કરે છે.

6. ડિસ્ટ્રીબ્યુટેડ ડેટાબેઝના ફાયદા નીચે પ્રમાણે છે -

- Increased reliablity and availabilaty. (વધારેલી વિશ્વાસપાત્રતા અને ઉપલબ્ધતા)
- Improved data intrgrity and data administration સુધારેલી ડેટા ઈન્ટીગ્રીટી અને ડેટાનો વહીવટ
- Improved access time સુધારેલા સમયની પ્રાપ્યતા
- Modular growth (મોડ્યુલર ગ્રોથ)

7. ડેટાબેઝ પસંદગીની માહિતી સ્ટોર કરે છે. જ્યારે નોલેજ-બેઝ IF - THEN નિયમ વધારાની માહિતી બેઝીક ડેટામાં ડાયરેકટલી સ્ટોર કરતા નથી. નોલેજબેઝ નિષ્કર્ષમાં નિયમો (rules) રીલેશનશીપને ડેટામાંથી મેળવવા માટે છૂટ આપે છે.

8. આર્ટીફિશીયલ ઈન્ટેલીજન્સ કોમ્પ્યુટર પ્રોગ્રામને કે જે માનવ-બુદ્ધિ શક્તિને ઉત્તેજન આપે છે તેનું અનુસરણ કરે છે. એટલે કે પ્રોગ્રામ અમુક કામ કરવા માટે કોમ્પ્યુટરને સગવડ કરી

આપે છે કે જેને માટે માનવ દ્વારા કામ કરવામાં આવે તો બુદ્ધિશક્તિની જરૂર પડે. આવા કામના ઉદાહરણ છે - સાદશ્ય જ્ઞાન, સમજશક્તિ, નેચરલ લેંગવેજ, રમત-ગમત, થિયરમ-સિદ્ધ કરવો, મેડીકલ નિદાન અને એન્જિનિયરીંગ ડિઝાઈન.

(A I) રીત જે ક્ષેત્રોમાં વધારે સફળતાપૂર્વક લાગુ કરવામાં આવે છે તે જ્ઞાન - આધારિત સિસ્ટમ, (એક્સપર્ટ સિસ્ટમ નોલેજ બેઝીસ વગેરે) રોબોટિક્સ, કોમ્પ્યુટર વિઝન, મશીન ટ્રાન્સલેશન, ન્યૂરલ નેટવર્ક અને અન્ય.

3.16 ચાવીરૂપ શબ્દો (KEY WORDS)

Dependency : આધારિતતા જે સંબંધોમાં જુદાં જુદાં સંબંધોની વચ્ચે અસ્તિત્ત્વ ધરાવતી હોય.

E-R Diagram : એવી આકૃતિ જે અલગ અલગ વસ્તુઓ વચ્ચેનો સંબંધ દર્શાવે.

Expert System : એવી પદ્ધતિ જે જ્ઞાન આધારિત માહિતીનો જથ્થો ધરાવે છે અને એવા નિયમો જે માહિતીનાં નિષ્ણાતોને અસરકારક નિર્ણયો લેવા શક્તિમાન બનાવે છે.

Freign key : એક ટેબલમાંનો સ્તંભ કે જે બીજા ટેબલની મુખ્ય ચાવીરૂપ હોય. તે પહેલા ટેબલ માટે ચાવીરૂપ હોય તેવું જરૂરી નથી.

Knowledge Base : 'નોલેજ બેઝ' એ નિષ્ણાત માહિતી પદ્ધતિનાં ડેટાબેઝ છે જે કોઇ ચોક્કસ વિષય માટે હોય છે, જેમાં સુસંગત વાસ્તવિકતા, નિયમો અને પદ્ધતિઓ પ્રશ્નોનાં નિરાકરણ માટે હોય છે. જ્ઞાનનું મુખ્ય એકમ IF-THEN-ELSE નિયમ પ્રમાણે પ્રદર્શિત કરવામાં આવે છે.

Norvealisation : સંગ્રહિત માહિતીની ચોક્કસાઇ અને માહિતીની સુસંગતતા માટે જુદાં જુદાં ટેબલ બનાવવા.

Primary Key : એક સ્તંભ કે સ્તંભોનો સમૂહ જે ટેબલમાંની ચોક્કસ હરોળને ઓળખે છે.

Relation : સંબંધ એ ટેબલ (સારણી) છે.

Relationship : બે અથવા વધારે વસ્તુઓ વચ્ચેનાં સંબંધો.

Schemea : માહિતી અને માહિતીનાં સમૂહ વચ્ચેનો તાર્કિક દેખાવ અને સંબંધ.

Sub Schemea : કોઈપણ માહિતી જે ચોક્કસ માહિતી પ્રોગ્રામ દ્વારા જરૂરી હોય તેનો તાર્કિક દેખાવ.

Trasparent : કૉમ્પ્યુટીંગમાં પ્રક્રિયા કે પદ્ધતિ જેમાં ઉપભોક્તા શામેલ હોય અને તેના અસ્તિત્ત્વથી અજાણ હોય.

3.17 સંદર્ભ અને વિશેષ વાચન (REFERENCE AND FURTHER READING)

CISMOD 93. *Proceedings of the International Conference on Information faste, and Management of Data* (1993). Indian National Scientific Documentatioi. ,en (INSDOC): New Delhi.

Courtney, James F. and Paradice, David B. (1988). *Database Systems for Managenu* Toronto: Times Mirror/Mosby College Publishing.

Codd, E.F. (1990). *The Relational Model for Database Management.* New York: Wes Publishing Company. Inc.

Curtin, Dennis P. [et al.] (1999). *Information Technology: The Breaking Wave.* I*Delhi: Tata McGraw-Hill Publishing Company Limited.

Date, C.J. (1989). *Introduction to Database Systems.* New Delhi: Narosa Publisl House.

Delobel, Claude and Adiba, Michel (1985). *Relational Database Systems.* Affisterc Elsevier Science Publishers B.V.

Elmasri, Ramaz and Navathe, Shaukan B. (2000). *Fundamentals of Database Syst* Asia:Pearson Education Asia.

MeFadden, Fred. R. and Hoffer, Jeffrey A. (1988). *Database Management.* California : The Benjamin/Cummings Publishing Company. Inc.

Martin, James (1988). *Principles of Database Management.* New Delhi: Prentice-Hall of India Private Limited.

McNurlin, Barbara C. and Spragu, Ralph H. (1989). *Information Systems Management in Practice.* New Jersey: Prentice-Hall.Inc.

McGraw, Karn, L. and Karan, Harbison-Briggs (1989). *Knowledge Acquisition: Principles and Guidelines.* New Jersey: Prentice-Hall. Inc.

Murray, Linda. A. and John Richardon, T.E. (1989). *Intelligent Systems in a Human Context.* Oxford : Oxford University Press.

O'Brien, James A (1997). *Introduction to Information Systems.* Singapore: Post, Gerld V. (2000). Database Management System. Delhi: Tata McGraw-Hill Publishing Company Ltd. *Irwin/McGraw Hill.*

Post , Gerald V.(2000). *Database Management Systems.* New Delhi.: Tata McGraw-Hill Publishing Company Limited.

Shepherd, Robert D. (2001). *Introduction to Computers and Technology.* New Delhi: Crest Publishing House.

Su Stanley, Y.W (1988). *Database Computers: Principles, Architectures and Techniques.* Singapore: McGraw-Hill Book Company.

Ullman, Jeffrey, D. (1991). *Principles of Database Systems.* New Delhi: Galgotia.

Williams Brain K. [et al.] (1999). *Using Information Technology.* Singapore: Irwin/ McGraw-Hill.