એકમ–2

ડેટા સ્ટ્રક્ચર, ફાઇલ ઓર્ગેનાઇઝેશન અને ફિઝીકલ ડેટાબેઝ ડિઝાઇન

DATA STRUCTURES, FILE ORGANISATION AND PHYSICAL DATABASE DESIGN

: માળખું :

2.0 ઉદ્દેશ

2.1 પ્રસ્તાવના

2.2 વ્યાખ્યા અને મૂળભૂત અભિગમ

- 2.2.1 ડેટા સ્ટ્રક્ચર શા માટે ?
- 2.2.2 મેકારી હાયરાર્કિ
- 2.2.3 રેઈડ ટેકનોલોજી
- 2.2.4 ઈન્ડેક્ષીસ
- 2.2.5 બાયનરી સર્ચ

2.3 ડેટા સ્ટ્રક્ચર

- 2.3.1 લીંક્ડ લિસ્ટ
- 2.3.2 ઈન્વરટેડ લિસ્ટ
- 2.3.3 B ટ્રીઝ

2.4 ફાઈલ અને તેનું ઓર્ગેનાઈઝેશન

- 2.4.1 ફાઈલ સ્ટોરેજ અભિગમ
- 2.4.2 સીકવન્શીયલ એસેસ મેથડ (SAM)
- 2.4.3 ઈન્ડેક્ષ સીકવન્શીયલ એસેસ મેથડ
- 2.4.4 ડાયરેક્ટ એક્સેસ મેથડ (DAM)

2.5 ફિઝીકલ ડેટાબેઝ ડિઝાઈન

2.6 સારાંશ

2.7 તમારી પ્રગતિ ચકાસોના ઉત્તરો

2.8 ચાવીરૂપ શબ્દો

2.9 સંદર્ભ અને વિશેષ વાચન

2.0 ઉદ્દેશ (OBJECTIVES)

◆ **આ યુનિટના વાચન પછી તમે...**

- ડેટા સ્ટ્રક્ચર અને ફાઇલ સંઘટના કઈ રીતે કરવી તેનો મૂળભૂત અભિગમ જાણી શકશો.
- ડેટાના ફિઝીકલી સ્ટોરેજના સ્ટ્રક્ચર અને ફાઇલસંઘટના કેવી રીતે કરવી તેનુ આકલન કરી શકશો, અને
- ડેટા સ્ટ્રક્ચર અને ફાઇલ સંઘટનમાં (ઓર્ગેનાઇઝેશનમાં) ડેટાબેઝ કુશળતાપૂર્વક કેવી રીતે મેળવી શકાય તેના લાભ અને તે અંગેની સૂઝ મેળવી શકશો.

2.1 પ્રસ્તાવના (INTRODUCTION)

ડેટા સ્ટ્રક્ચર અને ફાઇલ ઓર્ગેનાઇઝેશન ડેટાને ડેટાબેઝમાં કેવી રીતે ઓર્ગેનાઇઝ કરવું તેની રીત બનાવે છે. સામાન્ય રીતે તે ડેટાને ફિઝીકલ સ્ટોરેજ સાથે નિસ્બત ધરાવ છે જે ડેટાબેઝને ડેટામાં સ્ટોરીંગ અને સંગઠન માટે પુનઃ પ્રાપ્ત કરવા પ્રાથમિક રીતે સંકળાયેલા છે. ડેટા સ્ટ્રકચરમાં લીંક્ડ લિસ્ટ, ઈન્વરટેડ લિસ્ટ, B- ટ્રીઝ મિકસ્ડ ટેબલનો અન્ય સાથે સમાવેશ કરવામાં આવે છે. ડેટાબેઝનો

ઉપયોગ ડટા ફાઇલને મજબૂત બનાવવા (Build) કરવા માટે થઈ શકે. (ડેટા ફાઇલ અથવા તો ફાઇલ એ સમાન રેકોર્ડનું સંગઠન છે) અને ફાઇલ ઓર્ગેનાઇઝેશન ફાઇલને સરળતાથી પ્રાપ્ત કરવા માટેની પ્રક્રિયા નક્કી કરે છે.

ફાઇલનું સંગઠન (અથવા ફાઇલ સ્ટ્રક્ચર) એ ફાઇલમાં ડેટા માટેનું અને ડેટા મેળવવા માટેની પ્રક્રિયાનું સંયુક્તપણે પ્રતિનિધિત્વ કરે છે. ફાઇલ સ્ટ્રક્ચર વિનિયોગને વાંચવા, લખવા અને ડેટામાં ફેરફાર કરવા માટે દાખલ કરે છે. માનદંડને અનુરૂપ કરવા માટે ડેટાને અમુક ઓર્ડરમાં શોધવા માટે મદદરૂપ ફાઇલ સંગઠન ડેટાબેઝની ફિઝીકલ ડિઝાઇનની વ્યાખ્યા સ્પષ્ટ કરે છે અને તેના કામકાજ ઉપર બારીક નજર રાખે છે.

2.2 વ્યાખ્યા અને મૂળભૂત અભિગમ (DEFINITION AND BASIC CONCEPTS)

ડેટા સ્ટ્રક્ચર સાથે સંકળાયેલા કેટલાક મૂળભૂત અભિગમ ફાઇલ સંગઠન કે જે તેની સમજૂતી માટે જરૂરી પાર્શ્વભૂમિ પૂરી પાડવા આવશ્યક છે તેને નીચેના પેરેગ્રાફમાં આપવામાં આવ્યા છે.

2.2.1 ડેટા સ્ટ્રક્ચર શા માટે ? (Why Data Structure ?)

ડેટાબેઝની ડિઝાઇનમાં અને ફાઇલ સંગઠનમાં અગત્યનું પાસું - પ્રમાણમાં હાર્ડડિસ્કની ધીમી ગતિ અને ડિસ્કમાંથી માહિતી મેળવવા માટે જરૂરી છે. બધાં જ ડેટા સ્ટ્રક્ચર અને ફાઇલ સંગઠન ડિઝાઇન ડિસ્કની પ્રાપ્તિ અને તેની સંભાવના કે ઉપભોક્તા જે માહિતી માગે છે તે મેમરીમાં પહેલેથી જ સચવાયેલી છે. ડિસ્ક મેળવવાની જટિલ પ્રક્રિયા સામાન્ય રીતે 1.0 બોટલનેક તરીકે વર્ણવાય છે ડિસ્કના મલ્ટીપલ ટ્રીપ્સના ઉપયોગ દ્વારા મેળવવાની માહિતી માટે પ્રાપ્ત કરવાનો સમય ઘણું ખરું તો ઓછો કરે છે. આપણને જોઈતી માહિતી ડિસ્કના એક જ પ્રયાસ સાથે અથવા તો શક્યતઃ વધારે પ્રયાસ દ્વારા મળવો જોઈએ.

આટલા વર્ષોના પ્રયાસ બાદ ડેટા સ્ટ્રક્ચરના વિકાસ અને ફાઇલ સંગઠનનું સંશોધન કરતા એક વાત જાણવામાં આવે છે કે અગાઉ ફાઇલ ઉપરના સંશોધનની ટેપ ઉપર હતા અને એકબીજા સાથે સંકળાયેલા હતા. એકબીજાને સાંકળરૂપ જોડાયેલાની કિંમત ફાઇલમાં પ્રમાણમાં વધી હતી. ફાઇલ જ્યારે મદદ વગર અસાધારણ રીતે ક્રમશઃ મળવા માટે અતિશય વિશાળતા ધારણ કરે ત્યારે ઈન્ડેક્ષ ફાઇલમાં ઉમેરવામાં આવે છે. ઝડપથી સર્ચ કરી શકાય તે માટે નાની ફાઇલમાં ઈન્ડેક્ષ્ટ કીનું લિસ્ટ અને પોઇન્ટર્સ તૈયાર કરવામાં આવે છે. કી અને પોઇન્ટરની મદદથી ઉપભોક્તા મોટી, પ્રાથમિક ફાઇલને ડાયરેક્ટ મેળવી શકે છે. આમ છતાં ઈન્ડેક્ષના વધવા સાથે તેનું સંચાલન કરવું અઘરું પડે છે. ખાસ કરીને ગતિશીલ ફાઇલ કે જેમાં સેટ ઑફ કી બદલાતી રહે છે તે પછી 1960ના પૂર્વાર્ધમાં ટ્રી સ્ટ્રક્ચરનો વિચાર ઉદ્ભવ્યો અને તેને આજના B ટ્રીના ફોર્મમાં મેળવવા માટે વર્ષો વીતી ગયા આગળ ઉપર ફાઇલની ઝડપી પ્રક્રિયા માટે હેર્શીંગ ટેક્નિકને વિકસાવવામાં આવી.

2.2.2 મેમરી હાયરાર્કિ (Memory Hierarchy)

કોમ્પ્યુટર સ્ટોરેજ મીડિયા મેમરી હાયરાર્કિ ધરાવે છે કે જેમાં બે મુખ્ય કેટેગરી સ્ટોરેજ કરવામાં આવે છે.

પ્રાયમરી સ્ટોરેજ : સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (સી.પી.યુ.)માં ઉપયોગમાં આવતા સ્ટોરેજ મીડિયાને સ્પર્શે છે. દાત. મુખ્ય મેમરી અને વળી કેચ (ગુપ્ત મેમરી) પ્રાયમરી સ્ટોરેજ મેમરી RAM (રેન્ડમ એક્સેસ મેમરી) તરીકે પણ ઓળખાય છે તે ડેટાને ઝડપથી પૂરા પાડે છે, અને તે અસ્થિર હોય છે જેમ કે જો પાવર ન મળે તો તેની વિગતો નાશ પામે છે.

સેકન્ડરી સ્ટોરેજ : તેમાં મેગ્નેટિક ડિસ્ક ઓપ્ટીકલ ડિસ્ક અને ટેપનો સમાવેશ થાય છે. સેકન્ડરી સ્ટોરેજ એ (RAM) કરતા ધીમી ગતિએ ડેટા મેળવે છે. મેમરી હાયરાર્કિ આકૃતિ 2.1માં દર્શાવેલ છે જેમ જેમ કેચ મેમરીની નીચે ઉતરીએ તેમ ગતિમાં વધારો થાય છે, અને કિંમતો ઘટે છે.

(Cache Memory
(SRAM)
↓
Main Merory
(DRAM)
↓
Flash Memory
(EEPROM)
↓
Magnatic Disks
(Hard Disks & Floppies)
↓
Optical Disks
(CD ROM, WORM, CD-R, DVD)
↓
Magntic Tapes)

આકૃતિ : 2.1 મેમરી હાયરાર્કિ

સ્ટેટિક RAM (SRAM) કે જે ગુપ્ત (કેચ) મેમરી છે તેનો ઉપયોગ (CPU) પ્રોગ્રામના સંચાલન માટે ગતિ વધારવા માટે કરવામાં આવે છે. જ્યારે ડાયનેમિક RAM (IRAM)) (CPU)માટે મુખ્ય વર્ક એરિયા પ્રદાન કરે છે. ફ્લેશ મેમરી કે જે સ્થિર છે EEPROM (ઇલેક્ટ્રીકલી ઈરેઝેબલ પ્રોગ્રામેબેલ રીડ ઓન્લી મેમરી) (DRAM) અને મેગ્નેટિક ડિસ્ક વચ્ચે વધારે ગતિ અને કાર્યદક્ષતા ધરાવે છે. CD-ROM (કોમ્પેક્ટ ડિસ્ક રીડ ઓન્લી મેમરી) ડિસ્કનો ઉપયોગ ડેટાને કાયમી સાચવવા માટે કરવામાં આવે છે. અને ડેટાને એક વખત લખવા માટે વધારે વખત વાંચવા માટે ઉપયોગ કરવામાં આવે છે (DVD) (ડીજીટલ વિડિયો ડિસ્ક) ઓપ્ટીકલી ડિસ્કનો પ્રકાર એક ડિસ્કમાં ચારથી 15 ગીગાબાઈટ્સના સ્ટોરેજ ધરાવે છે. મેગ્નેટીક ટેપનો ઉપયોગ કાયમી જાળવણી અને ગુપ્ત (બૅકઅપ) સ્ટોરેજ માટે કરવામાં આવે છે. અને ટેરાબાઈટ્સના સ્ટોરેજ માટે તે વધારે જાણીતું છે. જ્યુક બોક્સ (ઓપ્ટીકલ અને ટેપ) (CD-ROM) અને ટેપને શણગારવા માટે દાખલ કરવામાં આવ્યા છે.

2.2.3 રેઈડ ટેકનોલોજી (RAID Technology) : સેકન્ડરી સ્ટોરેજ ટેકનોલોજીના ક્ષેત્રે મુખ્ય પ્રગતિ RAID (રીડન્ટન્ટ ઑફ ઇન એક્સપેન્સીવ ઈન્ડીપેન્ડટન્સ ડિસ્ક) ટેકનોલોજીની છે. રેઈડનો વિચાર રેઈડ આર્કિટેક્ચરના મોટા ભાગે વિકલ્પનો વિકાસ કરવાનો છે.

રેઈડનો મુખ્ય લક્ષ્ય મેમરી અને માઈક્રોપ્રોસેસરની સરખામણીમાં ડિસ્કનો બહોળા પ્રમાણમાં કાર્યક્રમ સુધારા માટેનો રેટ (દર) છે. જ્યારે (RAM)ની કેપેસીટી બે-ત્રણ વર્ષમાં ચારગણી થાય છે, ત્યારે ડિસ્કની સુધારણાનું પ્રમાણ વર્ષે 1 ટકા કરતા પણ ઓછું છે અને ડિસ્ક ટ્રાન્સફરનો સુધારણા દર લગભગ વર્ષે 20 ટકા જેટલો થાય છે. આમ છતાં ડિસ્કની કેપેસીટી વધારે ઝડપથી સુધરે છે ગતિ અને એક્સેસ ટાઈમની સુધારણા બહુ જ જૂજ મહત્ત્વ ધરાવે છે. ગતિ અને એક્સેસ ટાઈમનો પ્રશ્ન મોટા ભાગે નાની સ્વતંત્ર ડિસ્કનો ઉપયોગ કે જે સિંગલ હાઈપરફોર્મન્સ લોજીકલ ડિસ્ક તરીકે ઓળખાય છે. ડિસ્કની કામગીરી સુધારણા માટે ડેટા સ્ટ્રીપીંગ અભિગમનો ઉપયોગ કરે છે કે જે સમાંતરના ઉપયોગ કરે છે. ડિસ્ક સ્ટ્રીપીંગ અગત્યનો અભિગમ ઉદાહરણીત કરે છે, જેને આપણે વધારે ને વધારે સિસ્ટમ કન્ફીગ્યુરેશન પેરૂલલીઝમમાં જોઈ શકીએ છીએ . જ્યારે સિસ્ટમમાં કોઈ જગ્યાએ

બોટલ નેકની પરિસ્થિતિ ઊભી થાય ત્યારે બોટલનેકના સ્રોતને વધારવાનું વિચારો અને સિસ્ટમની રૂપરેખા દોરી તેને સમાંતર ઓપરેટ કરો, ડેટા સ્ટ્રીપીંગ, ડેટાને પારદર્શક રીતે મલ્ટીપલ ડિસ્કને સીંગલ લાર્જ, ફાસ્ટ ડિસ્ક બનાવવા માટે વિભાજીત કરે છે. આચ્છાદન (સ્ટ્રીપીંગ) 1/10ના સમગ્ર કાર્યદક્ષતાને મલ્ટીપલ 1/10ને દાખલ કરીને ડેટા સ્ટ્રાઈપીંગ ડિસ્કમાં સમાંતર ક્રમમાં ટ્રાન્સફર રેટને સુધારે છે. ડેટા સ્ટ્રાઈપીંગ ડિસ્ક વચ્ચે લોડબેલેન્સીંગની નિપુણતા મેળવે છે.

એ વાતની નોંધ લેવી જોઈએ કે ડેટા ફક્ત એક જ વખત એક સાથે વાંચી કે લખી શકાય છે. તેથી નમૂનારૂપ બદલાવ 512 બાઈટ્સ ધરાવે છે (બ્લોક સાઈઝ = 512 બાઈટ્સ) ડેટાનું સ્ટ્રાપીંગ ફાઈનર ગ્રેજ્યુઆલારીટી બાઈટ ઑફ ડેટાને બાઈટ્સ સીસ્ટમમાં વિભાજીત કરીને મલ્ટીપલ ડિસ્કમાં ફેરવી શકાય છે. 8 બીટ બાઈટ્સનો બીટ લેવલ સ્ટ્રાઈપીંગમાં ઉપયોગ કરીને ફિઝીકલ ડિસ્કની એક લોજીકલ ડિસ્ક તરીકે ઉપયોગ કરીને 8 ફિઝીકલ ડિસ્કની એક લોજીકલ ડિસ્ક તરીકે ગણના કરીને ડેટા ટ્રાન્સફર રેટમાં 8 ગણો વધારો કરી શકાય દરેક ડિસ્ક 1/10માં ભાગ લે છે. ટોટલ ડેટા વિનંતી દ્વારા વહેંચવામાં આવે છે તે 8 ગણું હોય છે ડેટા સ્ટ્રાઈપીંગ બ્લોકલેવલે કરી શકાય કે જે ફાઈલના બ્લોક ડિસ્ક સાથે વહેંચે છે.

કામગીરીની સુધારણા ઉપરાંત Raid, ડિસ્ક પર (રીડન્ડન્ટ) અનાવશ્યક માહિતી ખાત્રીપૂર્વક સુધારણા માટે ઉપયોગમાં આવે છે. અનાવશ્યક માહિતીને સામેલ કરવા માટેની એક ટેક્નિક મીરરીંગ કહેવામાં આવલે છે. ડેટાને રીડન્ડન્ટલી બે સમાન ફિઝીકલ ડિસ્ક પર લખવામાં આવે છે, જે લોજીકલ ડિસ્ક તરીકે ઓળખાય છે જો ડિસ્ક ફેઈલ થાય તો તે રીપેર ન થાય ત્યાં સુધી બીજી ડિસ્કનો ઉપયોગ કરવામાં આવે છે.

આમ, (Raid) ટેકનોલોજી ડિસ્ક પર ડેટા સ્ટોરની કામગીરી અને વિશ્વાસપાત્રતાની દેખીતી સુધારણા માટે (Raid) ટેકનોલોજીએ મહત્ત્વનું યોગદાન આપ્યું છે.

2.2.4 ઈન્ડેક્ષીસ (INDEXES) :

ઈન્ડેક્ષ એ એક ફાઈલ છે, કે જેમાં દરેક એન્ટ્રી (રેકોર્ડ) એક કે વધારે પોઈન્ટર્સ સાથે ડેટાવેલ્યુનો સમાવેશ થાય છે. (ફિઝીકલી સ્ટોરેજ સરનામા) ડેટાવેલ્યુ એ કેટલીક ઈન્ડેક્ષ ફાઈલના ફિલ્ડ વેલ્યું છે (ઈન્ડેક્ષ ફિલ્ડ) અને પોઈન્ટર્સને તે ફિલ્ડમાં સમાવિષ્ટ ઈન્ડેક્ષ ફાઈલમાં રેકોર્ડને એકરૂપતા આપે છે ઈન્ડેક્ષીંગનો અભિગમ સર્ચિંગની પ્રક્રિયા સાથે ઘનિષ્ઠ રીતે સંકળાયેલો છે. ઈન્ડેક્ષ (ક્યારેક લિસ્ટ તરીકે ઓળખાય છે.) જે બે રીતે ઉપયોગી છે.

તેના ઉપયોગ ઈન્ડેક્ષને ક્રમશઃ મેળવવામાં આવે છે એટલે કે ઈન્ડેક્ષ ફાઈલને ઓર્ડર લાદીને ઈન્ડેક્ષ ફિલ્ડની વેલ્યુ હાંસલ કરવાનો (2) તે ફિલ્ડની આપેલી વેલ્યુને આધારે તેના ડાયરેક્ટ ઉપયોગ કરી શકાય. સામાન્ય રીતે ઈન્ડેક્ષીંગ પુનઃપ્રાપ્તિને ગતિ આપે છે પણ અપડેટને ધીમું પાડે છે.

એડ્રેસ / પોઈન્ટર

Last Name	Pointer
Chauhan	A-67
Dhote	A-58
Farukh	A-63
Ganguly	A-22
Hopkin	A-42
Janadhan	A-47
Ramchandran	A-32
Ray	A-11

Index File

	ID	Last Name	First Name	Date Joined
A-11	1	Ray	Suchanda	1-29-96
A-22	2	Ganguly	Akhilesh	3-31-96
A-32	2	Ramchandran	Muthu	2-17-96
A-42	4	Hopkin	Bill	2-8-96
A-47	5	Janardan	Jana	1-6-96
A-58	6	Dhote	Anil	8-23-96
A-63	7	Farukh	Shiekh	3-28-96
A-67	8	Chauhan	Rakesh	12-29-96

Indexed File

આકૃતિ 2.2. ઈન્ડેક્ષીસનું ઉદાહરણ

2.2.5 બાયનરી સર્ચ (Binary Search) : બાયનરી સર્ચિંગ ટેક્નિક લાંબા ઇન્વેર્ટેડ લીસ્ટ (2.3.2.)ના ઇન્ડેક્ષના સર્ચ માટેનો સારો એવો સમય ઓછો કરવા માટે ઉપયોગમાં આવે છે. એ ટેક્નિકમાં જે વેલ્યુ શોધવાની હોય છે તેમાં સૌ પ્રથમ લીસ્ટના મધ્યમની વેલ્યુ સાથે સરખાવવામાં આવે છે. જે વેલ્યુ મેળવવાની છે તે લિસ્ટના મધ્યમાની વેલ્યુ સાથે સરખાવવામાં આવે છે. જે વેલ્યુમેળવવાની છે તે લીસ્ટના ઉપરના ભાગે અથવા નીચેના અડધા ભાગમાં છે, તે પછી મેળવવા માંગતી વેલ્યુ બરોબર અધવચ્ચે મધ્યમાની એન્ટ્રી સાથે સરખાવવામાં આવે છે. વેલ્યુ ચોથા ભાગના મધ્ય ભાગ સાથે સરખાવવામાં આવે છે અને જ્યાં સુધી ઇચ્છિત વેલ્યુ મળે નહી ત્યાં સુધી આ પ્રક્રિયા ચાલુ રાખવામાં આવે છે આ રીતે જ્યાં સુધી ઇચ્છિત વેલ્યુ મળે નહીં ત્યાં સુધી ડેટાને વિભાજીત કરતા રહેવું.

બાયનરી સર્ચનું ઉદાહરણ આકૃતિ 2.3માં દર્શાવવામાં આવ્યું છે. જર્નાર્દનનો પ્રવેશ શોધવા માટે લીસ્ટનો મધ્યભાગ શોધો (ગૌતમ) જનાર્દન ગૌતમ પછી આવે છે તેથી બીજા અડધા અડધેથી ભાગો (કમલા) જનાર્દન ન મળે ત્યાં સુધી શેષને વિભાજીત કરતા રહો.

Alexandr
Bhtnagar
Chand
Dhani
Ejaj
Feroz
1. Gautam
Hegde
3 Ipshita
4 Janardan
2 Kamla
Lata
Minakshi
Nirmal

આકૃતિ 2.3 બાયનરી સર્ચનું ઉદાહરણ

◆ **તમારી પ્રગતિ ચકાસો**

(1) ડેટાબેઝ સર્ચમાં ઇન્ડેક્ષની ભૂમિકા કેવી હોય છે ?

નોંધ (i) નીચે આપેલી જગ્યામાં તમારો જવાબ લખો

(ii) પ્રકરણના અંતે આપેલા જવાબો સાથે તમારા ઉત્તરો ચકાસો

....

....

....

....

....

....

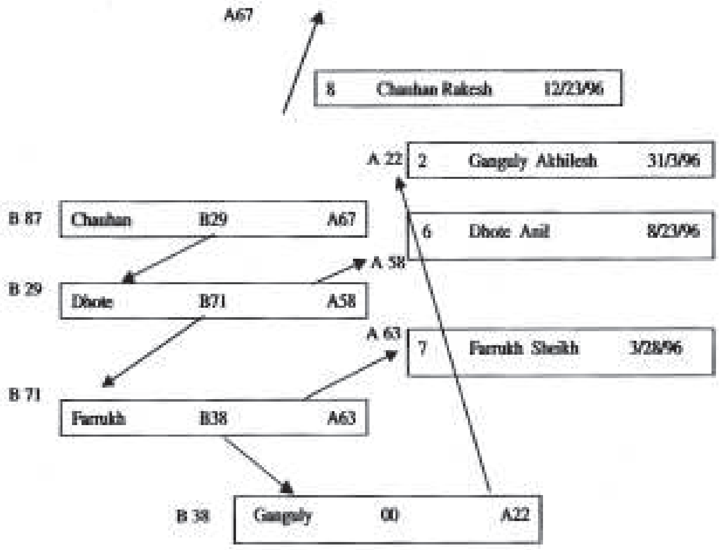
2.3 ડેટા સ્ટ્રક્ચર (DATA STRUCTURES)

ટર્મ ડેટા સ્ટ્રક્ચર ડેટા એલિમેન્ટ કોમ્પ્યુટર સિસ્ટમમાં કેવા સંબંધો ધરાવે છે તેની માહિતી આપે છે. ઇન્ડેક્ષીસનું સંચાલન સ્ટોર કરવામાં આવેલા ફિલ્ડ, સ્ટોર રેકોર્ડના ફિઝીકલ સીકવન્સ વગેરે ડેટા

સ્ટ્રક્ચરના ક્ષેત્રમાં આવે છે. ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમને સમજવા માટે ડેટા સ્ટ્રક્ચરને સમજવું અગત્યનું છે. ક્ષેત્રમાં આવ છે. ડેટા સ્ટ્રક્ચરના મુખ્ય પ્રકારોમાં લીંક્ડ લીસ્ટ (ઇન્ડેક્ષ) ઇનવરટેડ લીસ્ટ (ઇન્ડેક્ષીસ અને B ટ્રીઝ આને ડેટા સ્ટ્રક્ચર પછીના પેરેગ્રાફમાં આપવામાં આવ્યા છે.

2.3.1 લીંક્ડ લિસ્ટ (Linked Lists) : સાદુ લિંક્ડ લિસ્ટ એ રેકોર્ડમાં નિબદ્ધ કરવામાં આવેલા પોઇન્ટર્સની સાંકળ છે તે પ્રાયમરી કી અથવા સામાન્ય પ્રોપર્ટી સહિત બધા જ રેકોર્ડ સિવાય રેકોર્ડ સિકવન્સનું આરોપણ કરે છે. લીંક્ડ લિસ્ટ સાથે કોઈ પણ ડેટા એલીમેન્ટ અલગ સ્ટોર કરી શકાય છે. પોઇન્ટર તે પછી અન્ય ડેટા આઇટમનો ઉપયોગ કરે છે.

આકૃતિ 2.4 મૂળભૂત લિંક્ડ લિસ્ટના અભિગમને સમજાવે છે. આ ઉદાહરણમાં ડેટાની દરેક લાઇન અલગ રીતે સ્ટોર કરવામાં આવી છે. તે પછી ફિલ્ડ પર (કી) છેવટનું નામ (Last Name) રચવામાં આવે છે. જો કે ઇન્ડેક્ષના દરેક એલીમેન્ટ અલગ સ્ટોર કરવામાં આવે છે. ઇન્ડેક્ષ એલીમેન્ટ ત્રણ ભાગમાં આવરી લેવાયા છે. કી વેલ્યુ તે લાઇનના બાકીના ડેટાનું પોઇન્ટર અને બીજા ઇન્ડેક્ષનું એલીમેન્ટનું પોઇન્ટર ક્રમશઃ ડેટા પુનઃપ્રાપ્ત કરવા માટે પ્રથમ એલીમેન્ટ (ચૌહાણ)થી શરુ કરો, અને અન્ય એલીમેન્ટ (ધોટે) લીંકનને (પોઇન્ટર) અનુસરો, ઇન્ડેક્ષના દરેક એલીમેન્ટ અન્ય એલીમેંટના લીંકનને અનુસરતા જોવામાં આવે છે. સમગ્ર ડેટાલાઇનને કી વેલ્યુ માટે દરેક ઇન્ડેક્ષ પૂરું પાડે છે. લીંક્ડ લિસ્ટની શક્તિ ડેટાને કેટલી સરળતાથી અને ઝડપથી ઇન્સર્ટ કરવા માટે અને ડિલીટ કરવા માટેની કાર્યશક્તિ પર આધાર રાખે છે.



(આકૃતિ 2.4 લીંક્ડ લીસ્ટનું ઉદાહરણ)

2.3.2 ઇન્વર્ટેડ લિસ્ટ (Inverted Lists) :

ઇન્વર્ટેડ લિસ્ટને સરળતાથી જોવામાં આવે છે. કારણ કે પોઇન્ટર્સના ટેબલને ડેટા રેકોર્ડથી સ્ટોર કરવામાં આવેલા રેકોર્ડને બદલે પોઇન્ટર ફિલ્ડમાં આવરી લેવામાં આવે છે.

છૂટછવાયા (નોનડેન્સ) અને ઘનિષ્ઠ (ડેન્સ) લીસ્ટ વચ્ચેની સરખામણી કરવી જોઈએ. છૂટાછવાયા લીસ્ટના કેસમાં ફાઇલમાં થોડાંક જ રેકોર્ડ લીસ્ટના ભાગરૂપ હોય છે. જ્યારે ઘનિષ્ટ લિસ્ટમાં ફાઇલમાં મોટા ભાગનાં રેકોર્ડ પોઇન્ટર સાથે હોય છે.

યુનિક સેકન્ડરી કી (1:1 પ્રાયમરી કી સંકળાયેલી હોય) 1: (M)ના પ્રાયમરી કીના જોડાણ કરતા અલગ હોય છે. પ્રથમ કેસમાં ઘનિષ્ઠ ઇન્ડેક્ષ નિર્માણ કરવામાં આવે છે જ્યારે પછીના કેસમાં નોન ડેન્સ (છૂટછવાયા) ઇન્ડેક્ષ આપે છે.

ઇન્વર્ટેડ લીસ્ટનું (ઉદાહરણ) નીચે આપવામાં આવ્યું છે.

લીસ્ટ-1

Company	Area	Primary Key
Digital	Comuter	1245
Ford	Auto	1175
GM	Auto	1323
Intel	Computer	1232
Lockneed	Aerospace	1152

ઈન્ડેક્ષ ફાઈલ

લીસ્ટ-2

Company Symbol	Primary Key
Dec	1245
F	1175
GM	1323
INTL	1231
L	1152

ઈન્ડેક્સ ફાઈલ

આકૃતિ 2.5 ઘનિષ્ઠ ઈન્વર્ટેટ લિસ્ટ

ઉપર જણાવેલું લિસ્ટ એ ઘનિષ્ટ છે. કારણ કે તેમાં એક ઉપર એક એમ બંને કંપનીઓના નામ અને પ્રાયમરી કી અને કંપની સિમ્બોલ અને પ્રાયમરી કી ના સંબંધ બંધાયેલા છે.

આકૃતિ 2.6 એરિયા માટે અઘનિષ્ઠ ઈન્વર્ટેડ લિસ્ટનું ઉદાહરણ છે.

(એરિયા અને પ્રાયમરી કીના સંબંધ એકથી અધિક)

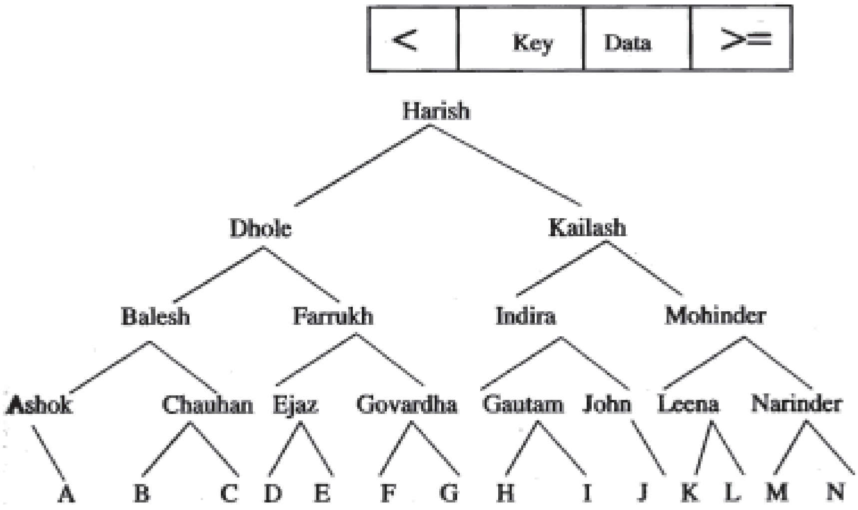
Aria	Primary Key
Aerospace	1152
Auto	1175, 1323
Computer	1231, 1245)

આકૃતિ 2.6 અઘનિષ્ઠ ઈર્ન્વટેડ લિસ્ટ

આ લીસ્ટને ઈન્વરટેડ કહેવામાં આવે છે કે કારણ કે કંપનીના નામને (અથવા વિસ્તારનું નામ) વર્ણમાળા પ્રમાણે ગોઠવવામાં આવ્યા છે અને સમકક્ષ પ્રાયમરી કીને 'ઇન્વરટેડ' કરવામાં આવી છે.

2.3.3 B-ટ્રીઝ (B-Trees) : કેટલાક લેખકો કહે છે કે અક્ષર "B" એ મૂળ બાયર માટે વપરાયો છે જ્યારે અન્યનું માનવું છે કે તે અક્ષર બેલેન્સડ માટે વપરાયો છે. B ટ્રીઝને સમતોલ એટલા માટે કહીશું કે બધા જ ટર્મિનલ (નીચાણ)નું મધ્યબિંદુની મૂળ સુધીની (ઉંચાઈ) લંબાઈ એકસરખી રહે છે. B ટ્રીઝ ઈન્ડેક્ષની માવજત અને નિપૂણતાપૂર્વક શોધ (સર્ચિંગ) માટે અલ્ગોરીધમને વિકસાવવામાં આવ્યું છે. કે જે પ્રાથમિક અને સેકન્ડરી ઈન્ડેક્ષનું પ્રતિનિધિત્વ કરવા ખાસ જાણીતું છે. B ટ્રીઝ ક્રમશઃ અને ઈન્ડેક્ષ બન્નેને મેળવવા માટે પૂરા પાડવામાં આવે છે અને તે સારા એવા લવચીક છે.

B ટ્રીની ઉંચાઈ એ હાયરાર્કિમાં લેવલનો ક્રમ છે. ઝાડ ઉપરના દરેક કેન્દ્રબિંદુ ઈન્ડેક્ષ એલીમેન્ટ ધરાવે છે. જેને કી વેલ્યુ અન્ય વધારાના ડેટા માટે પોઈન્ટર અને બે લીંક્ડ પોઈન્ટર ધરાવે છે. (આકૃતિ 2.7) એક લીંક (ડાબી બાજુ) પોઈન્ટ એલીમેન્ટને (કેન્દ્રબિંદુ) કે જેની વેલ્યુ ઓછી હોય જ્યારે બીજી લીક પોઈન્ટ (જમણી બાજુ) એલીમેન્ટ કરતા વધારે અથવા તો એકસરખી વેલ્યુ ધરાવે છે. કેન્દ્રબિંદુના નીચેના ભાગને પાંદડા કહેવામાં આવે છે કારણ કે ઝાડની ડાળીઓના છેડાના ભાગમાં આવેલા હોય છે,



આકૃતિ 2.7 B ટ્રીનું ઉદાહરણ

જો છેડાના પાંદડાઓ એક સરખા ન હોય તો (B) ટ્રી અસમતોલ કહેવાય છે એટલે કે જો જુદા જુદા ટર્મિનલ નોડ્સ ટોપ નોડ્સની જુદી જુદી ઊંડાઈએ હોય છે.

(B) ટ્રી વિશેષ કાર્યદક્ષતા ધરાવે છે પણ ફાઈલને કુશળતાપૂર્વક ક્રમાનુસાર મેળવી શકતી નથી. (B)ની ટ્રીની ઉપરની લેવલે લીંક્ડ લિસ્ટ સ્ટ્રક્ચરને ઉમેરીને પ્રશ્નને ઉકેલી શકાય છે. (B) ટ્રી અને ક્રમશઃ લીંક્ડ લિસ્ટના એકત્રીકરણને (B^+) ટ્રી કહેવામાં આવે છે.

◆ **તમારી પ્રગતિ ચકાસો**

2. વેપારી દષ્ટિએ બી ટ્રી બી (+) ટ્રીનો ઉપયોગ કરતા પ્રાપ્ય ડેટાબેઝ મેનેજમેન્ટ / સિસ્ટમના ઉદાહરણો આપો.

નોંધ 1 તમારા જવાબો નીચેની જગ્યામાં લખો.

2 તમારા જવાબો પ્રકરણના અંતે આપેલા ઉત્તરો સાથે સરખાવો.

....

....

....

....

....

....

2.4 ફાઈલ અને તેનું સંગઠન (FILES AND THEIR ORGANISATION)

2.4.1 ફાઈલ સ્ટોરેજ અભિગમ (File Storage Concepts) :

ફાઈલ એ રેકોર્ડનો ક્રમ છે ફાઈલનું સંચાલન ફિઝીકલ લે આઉટ અથવા ફાઈલમાં વારંવાર આવતા રેકોર્ડનું માળખુ છે. ફાઈલ સંચાલન રેકોર્ડ કેવી રીતે સ્ટોર કરવામાં આવે છે અને પ્રાપ્ત કરવામાંઆવે છે તે નક્કી કરે છે.

ઘણી બાબતોમાં ફાઈલમાં જ બધાં જ રેકોર્ડ એક સરખી ટાઈપના હોય છે. જો ફાઈલના દરેક રેકોર્ડ એટલી જ સાઈઝ બરાબર હોય તો (બાઈટ્સમાં) ફાઈલને ફીક્સલેંગ્થ રેકોર્ડમાં મૂકવામાં આવેલી હોય છે. જો ફાઈલમાં જુદા જુદા રેકોર્ડની

સાઈજ જુદી જુદી હોય તો ફાઈલને વેરીએબલ લેંગ્થ રેકોર્ડ હોઈ શકે.

(i) ફાઈલ રેકોર્ડ એક સમાન રેકોર્ડ જેવા હોય, પણ એક કે વધારે ફિલ્ડ જુદી જુદી સાઈઝના હોય છે. (વેરીએબલ લેંગ્થ ફિલ્ડસ્)

(ii) ફાઈલ રેકોર્ડ સમાન રેકોર્ડ ટાઈપના હોય છે, પણ એક કે વધારે ફિલ્ડ અમુક રેકોર્ડ માટે મલ્ટીપલ વેલ્યુ ધરાવતા હોઈ શકે.આ પ્રકારના ફિલ્ડને રીપીટીંગ ફિલ્ડ કહેવામાં આવે છે અને વેલ્યુના ગ્રુપને ફિલ્ડ માટે ઘણીવાર રીપીટીંગ ફિલ્ડ કહેવામાં આવે છે.

(iii) ફાઈલ રેકોર્ડ સમાન રેકોર્ડ જેવા હોય છે, પણ એક કે વધારે ફિલ્ડ વૈકલ્પિક હોય છે.

(iv) ફાઈલના રેકોર્ડ ભિન્ન રેકોર્ડ ટાઈપ હોય છે અને તેથી જુદી જુદી સાઈઝના હોય છે. (મિશ્ર ફાઈલ) જો જુદા જુદા સંબંધતિ રેકોર્ડ ડિસ્ક બ્લોકમાં એક જૂથ કરવામાં આવે તો આ પ્રમાણે થઈ શકે.

ફાઈલના રેકોર્ડ ડિસ્ક બ્લોકને આપવામાં આવે છે કારણ કે બ્લોક એ યુનિટ ડેટા છે, જે ડિસ્ક અને મેમરી વચ્ચે બદલાય છે. ડિસ્ક બ્લોકની સરખી સાઈઝમાં ટ્રેકનો વિભાગ (સ્ટોરેજ મીડિયા પર) ડિસ્ક ફોર્મેટીંગ દરમ્યાન ઓપરેટીંગ સિસ્ટમ દ્વારા સેટ કરવામાં આવે છે. હાર્ડવેર એ સર્ફેસ નંબર, ટ્રેક નંબર અને બ્લોક નંબર ધરાવતા બ્લોકને સંબોધે છે. બફર – મુખ્ય સ્ટોરેજમાં અનામત રખાયેલા નજીકના વિસ્તારને આવરી લેતા એક બ્લોકને પણ સરનામુ હોય છે. રીડ કમાંડ માટે ડિસ્કમાના બ્લોક બફરમાંથી નકલ કરવામાં આવે છે. જ્યારે રાઈટ (લેખન) કમાંક માટે બફરની વિગતો ડિસ્ક બ્લોકમાં કોપી કરવામાં આવે છે. કેટલીકવાર ઘણાં નજીકના બ્લોક જેને લુમ (ક્લસ્ટર) કહેવામાં આવે છે, તેને યુનિટ તરીકે ટ્રાન્સફર કરવામાં આવે છે. આવા કિસ્સામાં બફરસાઈઝને ઝૂમખાના રૂપમાં ગોઠવવામાં આવે છે.

જ્યારે બ્લોક સાઈઝ રેકોર્ડ સાઈઝ કરતાં મોટી હોય, દરેક બ્લોક સાઈઝ ઘણાં રેકોર્ડ ધરાવે છે. જ્યારે ઘણાં રેકોર્ડ ધરાવતી ફાઈલ એક બ્લોકમાં સમાવિષ્ટ ન થઈ શકે તેથી રેકોર્ડને એક કરતાં વધારે બ્લોકમાં આવરી લેવામાં આવે છે.

અહીં ટર્મ ફાઈલ ઓર્ગેનાઈઝેશન અને પ્રાપ્તિ માટેની રીત વિશેનો તફાવત નોંધવો ઉપયોગી થઈ પડશે. ફાઈલનું સંગઠન રેકોર્ડમાં ડેટાફાઈલનું સંગઠન, રેકોર્ડ, બ્લોક અને સ્ટ્રકચરની પ્રાપ્તિને સૂચવે છે. રેકોર્ડ અને બ્લોક કેવી રીતે મીડિયામાં જગ્યા ધરાવે છે અને આંતરિક રીતે સંકળાયેલા છે. તેને આવરી લે છે. બીજી બાજુ રેકોર્ડની પ્રાપ્તિ માટેની રીત – વિવિધ માત્રામાં સંચાલન ધરાવે છે. દા.ત. શોધો, વાંચો અને સુધારો વગેરે કે જે ફાઈલને લાગુ પડે છે. સામાન્ય રીતે એ શક્ય છે કે કેટલીક રીતો ફાઈલ સંચાલનને આવરી લે છે. કેટલીક રીતો, જો કે ફાઈલ સંચાલનને અમુક જ રીતે લાગુ પડે છે. દા.ત. આપણે ફાઈલમાં ઈન્ડેક્ષ એકસેસ રીતને ઈન્ડેક્ષ સિવાય લાગુ પાડી શકાય નહી.

2.4.2 સીક્વન્શીઅલ એક્સેસ મેથડ (SAM) (Sequential Access Method)

સીકવન્શીઅલ એક્સેસ મેથડ (S A M) ક્રમશઃ ફાઈલમાં પ્રીડીફાઈન ઓર્ડરમાં સ્ટોર કરવામાં આવે છે. સીક્વન્શીયલ ફાઈલમાં તેનું આગમન સામાન્ય રીતે પ્રાયમરી કી ઉપર વિભાજીત કરવામાંઆવે છે. અને પ્રાયમરી કી દ્વારા સ્ટોરેજ

મીડિયામાં ફિઝીકલ ગોઠવવામાં આવે છે. જો ફક્ત સીક્વન્શીયલ એક્સેસની જરૂર હોય (તેવું કવચિત જ બને છે) સીક્વન્શીયલ મીડિયા (મેગ્નેટીકટેપ)આવી ફાઈલને પ્રોસેસ કરવા માટે બહુ જ યોગ્ય અને કરકસરયુક્ત રીત છે. ડાયરેક્ટ એક્સેસ ડીવાઈસીસ - ડિસ્ક હોઈ શકે, પરંતુ નિર્દેશ ક્રમશઃ હોવું જરૂરી નથી. ડાયરેક્ટ એક્સેસ ડીવાઈસીસનો ઉપયોગ હોવા છતાં, કેટલાંક પ્રોસેસીંગ સંરચના સીકવન્શીયલ એક્સેસ દ્વારા ઉત્તમ રીતે કરવામાં આવે છે.

સીક્વન્શીયલ એક્સેસ અવાર-નવાર પ્રોસેસ કરવા માટે જરૂરી મોટા પ્રમાણમાં ડેટા એ ઝડપી અને કાર્યક્ષમ છે. - આમ છતાં સીક્વન્શીયલ એક્સેસ પ્રોસેસીંગમાં બધા જ નવા ટ્રાન્ઝેકશન યોગ્ય સીકવન્સમાં વિભાજિત કરવામાં આવે તે જરૂરી છે. આ ઉપરાંત મોટા ભાગના ડેટાબેઝ અથવા ફાઈલને શોધવા, સ્ટોર કરવા અથવા સુધારા - વધારા કરવા માટે, ભલે ડેટા રેકોર્ડની સંખ્યા નાની હોય, તો પણ સર્ચ કરવા પડે છે. આમ, ઝડપથી અપડેટીંગ અથવા પ્રતિભાવ માટે આ રીત ખૂબ ધીમી છે.

સીક્વન્શીયલ ફાઈલ સામાન્ય રીતે બેક-અપ અથવા જુદી જુદી સિસ્ટમમાં ડેટાને અલગ સિસ્ટમમાં વહન કરવા માટે ઉપયોગમાં કરવામાં આવે છે. સીક્વન્શીયલ (A S C I I) ફાઈલ એ નિકાસ / આયાત માટે જાણીતું ફોર્મેટ છે કે જેને ઘણી ડેટાબેઝ સિસ્ટમ ટેકો આપે છે.

2.4.3 ઈન્ડેક્ષ સીકવન્શીયલ એક્સેસ મેથડ (ISAM) (Indexed Sequential Access Method)

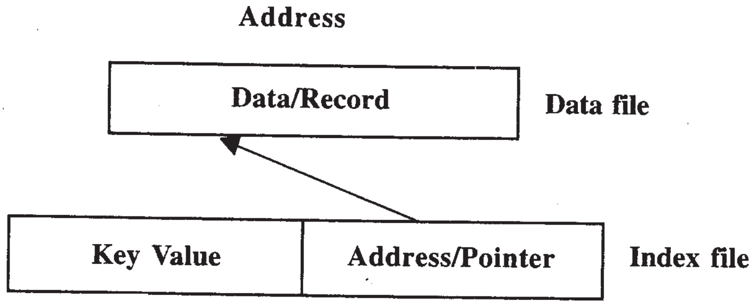
ઈન્ડેક્ષ સીકવન્શીયલ ફાઈલમાં વારંવાર આવતા રેકોર્ડને ડાયરેક્ટ એક્સેસ સ્ટોરેજ ડિવાઈસ ઉપર પ્રાયમરી કી વડે વિભાજિત કરી વ્યવસ્થિત સ્ટોર કરવામાં આવે છે. આ ઉપરાંત એક જુદું ટેબલ (અથવા ફાઈલ) જેને ઈન્ડેક્ષ કહેવામાં આવે છે. તેને દરેક રેકોર્ડની રચના સાથે ફિઝીકલ એડ્રેસ આપવા માટે સંચાલન કરવામાં આવે છે. આ અભિગમ રેકોર્ડની ઘટનાને ડાયરેક્ટ એક્સેસ ઈન્ડેક્ષ ટેબલ અને સીકવન્શીયલ એક્સેસ દ્વારા સ્ટોરેડ મીડિયામાં પડેલા રેકોર્ડને આપે છે.

ઈન્ડેક્ષ ફાઈલ દ્વારા રેકોર્ડના ફિઝીકલ સરનામાને પોઈન્ટર કહેવામાં આવે છે. પોઈન્ટર અથવા તો સરનામુ ઓપરેટીંગ સિસ્ટમ અને તે થકી ઉપયોગમાં આવતા ડેટાબેઝ કેટલીક પદ્ધતિઓ અપનાવી શકે છે.

હાલમાં સિસ્ટમ ફિઝીકલ સરનામાને બદલે વાસ્તવિક સરનામાનો ઉપયોગ કરે છે. વાસ્તવિક સરનામા એ કલ્પિત ડિસ્ક ડ્રાઈવ લે આઉટ પર આધારીત હોઈ શકે. ડેટાબેઝ, બેઝ સેટ ઓફ ટ્રેક્સ અને સિલિન્ડરને અનુસરે છે. તે પછી કોમ્પ્યુટર આ વેલ્યુને ખરેખર સ્ટોરેજ સ્થળમાં નકશો તૈયાર કરે છે. આ વ્યવસ્થા એ અભિગમના મૂળભૂત તરીકે વરચ્યુઅલ સીકવન્શીયલ એક્સેસ મેથડ (V S A M) તરીકે ઓળખાય છે. અન્ય સામાન્ય અભિગમ એ ફાઈલની શરૂઆતથી તેના અંતરના સ્થળને દર્શાવવાનો છે. વાસ્તવિક અથવા સંબંધિત સરનામા એ તેની શરૂઆતને લીધે સતત વધારે સારા હોય છે.

થોડાંક રેકોર્ડને જો ઝડપથી પ્રક્રિયા કરવાની હોય તો જરૂરી રેકોર્ડને પ્રત્યક્ષ મેળવવા ઈન્ડેક્ષનો ઉપયોગ કરવામાં આવે છે. આમ છતાં રેકોર્ડના ઘણાં નંબરની નિયમિત પણે પ્રક્રિયા કરવાની હોય તો આ રીત માટે આપવામાં આવેલા ક્રમશઃ સંગઠનનો ઉપયોગ કરવામાં આવે છે.

ઈન્ડેક્ષ ફાઈલનો ઉપયોગ કરતી પ્રક્રિયાનું ઉદાહરણ - આકૃતિ - 2.8માં આપવામાં આવ્યું છે.



આકૃતિ 2.8 ઈન્ડેક્ષ ફાઈલનો ઉપયોગ કરતા ડેટા એક્સેસ

2.4.4 ડાયરેક્ટ એક્સેસ મેથડ (D A M) (Direct Access Method)

જ્યારે ડાયરેક્ટ એક્સેસ રીતનો ઉપયોગ કરવામાં આવે છે ત્યારે સ્ટોરેજ મીડિયા પર કોઈ ચોક્કસ ફાઈલમાં ઘટિત રેકોર્ડને કોઈ ચોક્કસ ક્રમમાં ગોઠવવાની જરૂર નથી. આમ છતાં ડાયરેક્ટ સંગઠન વિવિધ રીતનો ઉપયોગ કરતા દરેક રેકોર્ડના વિસ્તારના સ્ટોરેજનું કોમ્પ્યુટરે ધ્યાન રાખવું જોઈએ, કે જેથી જરૂર પડે ત્યારે ડેટાને પુનઃ પ્રાપ્ત કરી શકાય. નવા ગોઠવાયેલા ડેટાને જુદા તારવવાની જરૂર નથી, અને તાત્કાલીક ઉપયોગ અથવા અપડેટીંગ પ્રક્રિયા સહેલાઈથી કરી શકાય.

ડાયરેક્ટ એક્સેસ રીતમાં રેકોર્ડના સરનામાને કોમ્પ્યુટર કરવામાં અલ્ગારીધમનો ઉપયોગ કરવામાં આવે છે. પ્રાયમરી કી વેલ્યુ એ અલ્ગારીધમનું ઈનપુટ છે અને રેકોર્ડના બ્લોક સરનામા એ આઉટપુટ છે.

અભિગમનો અમલ કરવા માટે સ્ટોરેજ જગ્યાનો થોડોક ભાગ ફાઈલ માટે અલાયદો રાખવામાં આવે છે. આ જગ્યા ફાઈલને સાચવવા માટે તથા તેના વિકાસને માટે પૂરતી મોટી હોવી જોઈએ. તે પછી આપવામાં આવેલી પ્રાયમરી કી ને યોગ્ય રીતે સરનામા મેળવવા માટે અલ્ગારીધમનો ઉપયોગ કરવામાં આવે છે. અલ્ગારીધમ એ સામાન્ય રીતે હશીંગ અલ્ગારીધમ તરીકે ઓળખાય છે. પ્રાયમરી કી વેલ્યુને સરનામામાં ફેરવવામાં આવતી પ્રક્રિયાને કી -ટુ - એડ્રેસ ટ્રાન્સફોર્મેશન તરીકે ઓળખવામાં આવે છે.

એક કરતાં વધારે લોજીકલ રેકોર્ડ ઘણું ખરું બ્લોકમાં બંધ બેસે છે. તેથી આપણે સ્ટોરેજના વિસ્તારને રીઝર્વ રાખવાનો વિચાર કરીએ છીએ. રેકોર્ડ સ્લોચને ક્રમશઃ L થી N સુધી નંબર આપેલા છે. આ ક્રમશઃ ક્રમાંક રીલેટીવ પોઈન્ટર્સ અથવા રીલેટીવ એડ્રેસીસ કહેવાય છે, કારણ કે તે ફાઈલને શરૂઆતના અનુસંધાનને રેકોર્ડના સ્થળનો નિર્દેશ કરે છે.

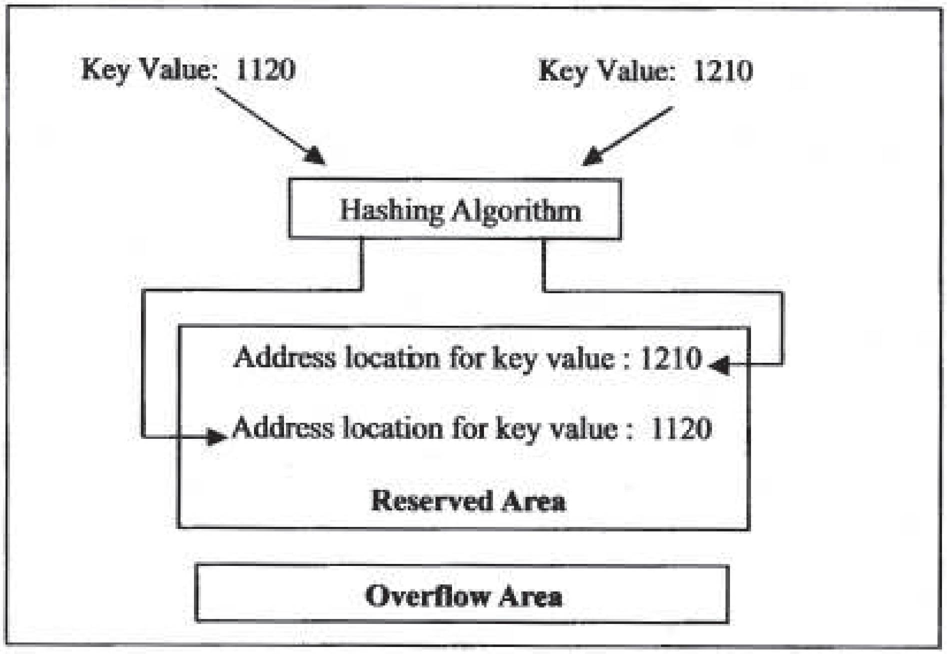
હેશીંગ અલ્ગોરીધમનો હેતુ સંબંધિત સરનામા કે જે રીઝર્વ સ્ટોરેજ જગ્યાએ અવ્યવસ્થિતપણે પણ એક સમાન રીતે રેકોર્ડને છૂટા પાડે છે. રેકોર્ડને બહુ ઝડપથી પુનઃ પ્રાપ્ત કરી શકાય છે. કારણ કે ઈન્ડેક્ષ દ્વારા ડિસ્કફાઈલમાં સ્ટોર કરવામાં આવેલા ટેબલ-લુક-અપ દ્વારા સરનામાને કોમ્પ્યુટ કરવામાં આવે છે.

જો એ જ બ્લોકમાં એક કરતાં વધારે રેકોર્ડ સચવાયેલા હોય તો અથડામણ થવાની સંભાવના છે. કારણ કે એક બ્લોક ઘણાં રેકોર્ડ ધરાવે છે. અથડામણ ત્યારે જ સંભવ છે જ્યારે બ્લોક કેપેસીટી કરતાં રેકોર્ડના નંબર વધારે હોય. આવી પરિસ્થિતિને

નિવારવા ડાયરેક્ટ એક્સેસ મેથડ અથડામણ નિવારણ માટે વ્યાપક વિસ્તારને મદદ કરે છે, કે જે ક્રમશઃ શોધ કરે છે.

હેશ કી અભિગમ એ ખૂબ જ ઝડપી છે કારણ કે વેલ્યુ સ્ટોરેજ લોકેશનમાં ઝડપથી બદલાય છે, અને ડિસ્કના એક પ્રયત્ને પુનઃ પ્રાપ્ત થઈ શકે છે.

ડાયરેક્ટ એક્સેસ મેથડ કે જે હેશ કી નો ઉપયોગ કરે છે. તેને આકૃતિ 2.9માં દર્શાવી છે.



આકૃતિ 2.9 હેશ કીના અભિગમનો ઉપયોગ કરતી ડાયરેક્ટ એક્સેસ

2.5 ફિઝીકલ ડેટાબેઝ ડિઝાઈન (PHYSICAL DATABASE DESIGN)

ફિઝીકલ ડેટાબેઝ ડિઝાઈનમાં ડેટા સ્ટ્રકચર અને ફાઈલ સંગઠન અગત્યનો ભાગ ભજવે છે. (યુનિટ 3 ડેટા માટે ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમને ડેટાબેઝ આર્કીટિકચર અને ડેટાબેઝ ડિઝાઈન સ્વરૂપને તપાસો.) ફિઝીકલ ડિઝાઈનનું ધ્યેય સ્ટોરેજમાં આવેલા યથાયોગ્ય ડેટા સાથે તાલ-મેલ કરવાનું છે, કે જેથી ડેટાબેઝની કામગીરી સારી રીતે થઈ શકે. જ્યાં સુધી આપણે પ્રશ્ન ફેર-બદલાવ અને ડેટાબેઝને સરળ રીતે ચલાવવા માટે આવશ્યક વિનિયોગને જાણી શકીએ ત્યાં સુધી ફિઝીકલ ડિઝાઈનના નિર્ણયને અર્થપૂર્ણ રીતે પૃથક્કરણ કરી શકીએ નહી.

ફિઝીકલ ડેટાબેઝ ડિઝાઈન પર નીચેના અભિગમ પ્રભાવ પાડે છે.

(1) ડેટાબેઝ પ્રશ્ન અને ફેરબદલનું પૃથક્કરણ.

દરેક પ્રશ્ન માટે આપણે સ્પષ્ટતા કરવી જોઈએ. -

(a) પ્રશ્નના સમાધાન માટે ફાઈલ

(b) પ્રશ્ના સ્પષ્ટીકરણ માટે ઉપયોગમાં લેવાયેલી પસંદગીની શરતો.

(c) કોઈ સંયુક્ત શરતો અથવા લીક્ડ મલ્ટીપલ ટેબલ્સ કે જેને માટે પ્રશ્નનું નિવારણ કરવાનું હોય તેનું કારણ સમજવું.

(d) પ્રશ્નના સમાધાન માટે જે વેલ્યુની પુનઃપ્રાપ્તી કરવાની હોય તેનાં ગુણધર્મ જોવા.

આરોપણ કરવામાં આવેલા એક્સેસ સ્ટ્રક્ચરની વ્યાખ્યા માટે (b) અને (c) ઉમેદવાર છે. દરેક અપડેટ આદાન- પ્રદાન માટે આપણે નિર્દેશ કરવો જોઈએ -

(a) ફાઈલને અપડેટ કરવી.

(b) દરેક ફાઈલના સંચાલન માટેની પ્રક્રિયા (ઈનસર્ટ, અપડેટ અને ડીલીટ)

(c) ડીલીટ અથવા અપડેટ માટેની પસંદગીની શરતો સ્પષ્ટ કરવી.

(d) અપડેટ પ્રક્રિયા દ્વારા કોની વેલ્યુ બદલાય છે તેનું આરોપણ.

અહીં આરોપણ કરવામાં આવેલા C એ એક્સેસ સ્ટ્રક્ચર અને આરોપીત d માટે ઉમેદવાર છે. જ્યારે d એ એક્સેસ સ્ટ્રક્ચરને નિવારવા માટે ઉમેદવાર છે. કારણ તેમાં સુધારા - વધારા કરવા માટે ચોક્કસ સ્ટ્રક્ચરને અપડેટ કરવાની જરૂર પડશે.

2. પ્રશ્નો અને આદાન - પ્રદાન માટે અપેક્ષિત ફ્રિકવન્સીનું પૃથક્કરણ :

દરેક ફાઈલમાં સમગ્રતયા પ્રશ્નો અને આદાન - પ્રદાનના પૃથક્કરણ અથવા સંયુક્ત પૃથક્કરણ અપેક્ષિત ફ્રિકવન્સીનો સમાવેશ થાય છે.

3. પ્રશ્નોના સમાધાન અને આદાન- પ્રદાન માટે સમય સંકોચનું પૃથક્કરણ :

સમયબદ્ધતા માટે કેટલાક પ્રશ્નો અને આદાન - પ્રદાન માટે સમયના અભાવનો નિયમબદ્ધ તણાવ ઉભો કરે છે. પ્રાયમરી એક્સેસ સ્ટ્રક્ચર માટે આરોપણની પસંદગી પ્રશ્નોત્તર અને આદાન - પ્રદાન માટે સમયબદ્ધતા એ પ્રાયમરી એક્સેસ સ્ટ્રક્ચર માટે મુખ્ય અગત્યતા ધરાવે છે.

4. અપડેટ ઓપરેશન માટે અપેક્ષિત ફ્રિકવન્સીનું પૃથક્કરણ :

(એનાલાઈઝીંગ ધી એક્સપેકેટડ ફ્રિક્વન્સીસ ઓફ અપડેટ ઓપરેશન્સ.)

જે ફાઈલને અવાર- નવાર અપડેટ કરવાની જરૂર પડે તેને ઓછામાં ઓછા નંબરને પ્રાપ્ત કરવા માટે પથને સૂચક કરવા જોઈએ. કારણ કે એક્સેસ પથને અપડેટ કરવા માટે તે પોતે જ અપડેટ પ્રક્રિયાને ધીમી પાડે છે.

અગાઉની માહિતીના આધારે ઈન્ડેક્ષીંગ માટે ફિઝીકલ ડેટાબેઝ ડિઝાઈનનો નિર્ણય સંબોધી શકે છે. આરોપણ કે જેની વેલ્યુ સમાનતા અથવા પંક્તિની પરિસ્થિતિમાં (સીલેકશન ઓપરેશન) અને તેઓ કે જે ચાવીરૂપ છે અને અથવા સંયુક્ત પરિસ્થિતિમાં ભાગ લે છે (જોઈન ઓપરેશન) એક્સેસ પથની જરૂર પડે છે. પ્રશ્નોત્તરની કાર્યવાહી ઈન્ડેક્ષીંસ અથવા હેશીંગ યોજના પસંદગી અને જોડાણને વેગ આપવા અસ્તિત્વ ધરાવે છે. બીજી બાજુ ઈન્સર્ટ, ડીલીટ અને અપડેટ પ્રક્રિયા દરમ્યાન ઈન્ડેક્ષીસની હાજરી સમગ્રતયા વધારો કરે છે.

ઈન્ડેક્ષીંગ માટેના નિર્ણય લેતી વખતે નીચેના મુદ્દાઓ ધ્યાનમાં રાખવા જોઈએ.

(i) ઈન્ડેક્ષ કરવામાં આવનાર પસંદગીની શરતો અથવા જોડાણમાં ઉપયોગ કરવામાં આવે છે તેનું આરોપણ તે ચાવી (કી) અથવા તો કેટલાંક પ્રશ્નોત્તર હોવા જોઈએ.

(ii) ઈન્ડેક્ષ એક અથવા મલ્ટીપલ આરોપણ રચી જો એક સંબંધી એક સાથે ઘણાં બધાં પ્રશ્નોનું આરોપણ કરવામાં આવે તો વિશદ્ આરોપણ ઈન્ડેક્ષની જરૂર પડે છે.

(iii) ઈન્ડેક્ષનું ઝુંમખું (નોન - કી - ફિલ્ડ પર ઈન્ડેક્ષનું નિર્માણ - દા.ત. જો ફાઈલમાં ઘણાં રેકોર્ડ ફિલ્ડ માટે સમાન વેલ્યુના હોઈ શકે.) પ્રશ્નોની હારમાળા માટે ઘણાં ઉપયોગી થઈ શકે, જો વધારે આરોપણની જરૂર પડે તો, આ નિરૂપણથી ઝૂમખું (CLUSTER) બનાવવામાં આવે છે. તેનાં નિર્ણય પહેલાં અવલોકન કરવું પડે. ટેબલ દીઠ એક ઈન્ડેક્ષ પ્રાયમરી અથવા ઈન્ડેક્ષનું ઝૂમખું વધારેમાં વધારે હોઈ શકે.

(iv) RIBMSs સામાન્ય રીતે B+ટ્રીનો ઉપયોગ કરે છે, અને હેશ ઈન્ડેક્ષીસ પણ કેટલીક સિસ્ટમમાં ઉપયોગમાં લેવાય છે. B+ટ્રી સમાનતા અને પ્રશ્નોની હારમાળા બન્નેને મદદરૂપ સર્ચ કી દ્વારા આરોપણ કરવામાં આવે છે. હેશ ઈન્ડેક્ષીસ ખાસ કરીને જોડાણ દરમ્યાન, સમાન પરિસ્થિતિમાં સારું કામ કરે છે.

2.6 સારાંશ (SUMMARY)

આ પ્રકરણમાં ડેટા સ્ટ્રક્ચરને ફાઈલ સંગઠન અને ફિઝીકલ ડેટાબેઝ ડિઝાઈન સંબંધિત કેટલાંક પ્રશ્નોના ઉકેલને આવરી લેવામાં આવ્યા છે. અને તેની સમજની સરળતા માટે જરૂરી પાર્શ્વભૂમિ આપવામાં આવી છે. એક્સેસ મેથડની કાર્યદક્ષતા પર ડેટા સ્ટ્રક્ચર અને ફાઈલ સંગઠનનો મુદ્દો સમજાવવામાં આવ્યો છે. રેઈડ RAID ટેકનોલોજી, બાયનરી સર્ચ અને ઈન્ડેક્ષની એક્સેસ ગતિને કેવી રીતે સુધારી શકાય તેને ઉદાહરણ સહિત ચર્ચવામાં આવ્યું છે. નમૂના રૂપ ડેટા સ્ટ્રક્ચર (લીંક્ડ લીસ્ટ, ઈનવરટેડ લીસ્ટ અને Bટ્રી) અને ફાઈલ સંગઠન ટેક્નિકને SAM, ISAM અને DAM સમજાવવામાં આવી છે. ફિઝીકલ ડેટાબેઝ ડિઝાઈન અને એક્સેસ સ્ટ્રક્ચર પર પરિણામને સ્પર્શતા મુદ્દાને સમજાવવામાં આવ્યા છે.

આ વિષયને જટીલ કી સંબંધિત અભિગમને સમજાવવા માટે આ પ્રકરણ પાયો નાંખે છે.

2.7 તમારી પ્રગતિ ચકાસોના ઉત્તરો

1. યોગ્ય ઈન્ડેક્ષીસનું અસ્તિત્વ ડેટા આરોપણ માટે સંવેદનશીલ છે. પ્રશ્નોત્તર માટેના સમાધાન માટે સમય મુખ્ય ભાગ ભજવે છે.માહિતી મેળવવા ગતિ આપવા માટે ઈન્ડેક્ષનો ઉપયોગ કરવામાં આવે છે. આમ છતાં ઈન્ડેક્ષની હાજરી અપડેટીંગ પ્રક્રિયાને ધીમી કરવાનું વલણ ધરાવે છે.

2. મોટે પાયે વ્યાવહારિક ધોરણે પ્રાપ્ય DBMS પાસે B-ટ્રી / B+ટ્રી ઈન્ડેક્ષ રચતા સાધનો, સિસ્ટમમાં બાંધવામાં આવે છે. 1.0. ગતિ વધારવા અને પ્રશ્નોના નિરાકરણ માટે B+ટ્રી ઈન્ડેક્ષ સીસ્ટમ સહજતાથી વેગ આપે છે. ORACLE અને SYBASE એ B-ટ્રી / B+ટ્રીને મદદ કરનાર રીલેશનલ ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમના ઉદાહરણ છે.

2.8 ચાવીરૂપ શબ્દો (KEY WORDS)

Cache Memory : કોમ્પ્યુટરમાં ચાલતી પ્રક્રિયાઓ દરમ્યાન કેન્દ્રિય પ્રોસેસીંગ એકમમાં આવેલ ઝડપી સંગ્રહ ક્ષમતા ધરાવતો ભાગ.

Binary Search : અનુક્રમિત થયેલ માહિતીને શોધવાની પદ્ધતિ.

B-tree : અનુક્રમિત માહિતી સંગ્રહ કરવાની પદ્ધતિ જે ખૂબ કાર્યક્ષમ હોય અને માહિતી એક્સેસ કરવા વિવિધ શ્રેણીઓ ધરાવતી હોય.

B+ tree : B+ treeના બંધારણની વિવિધતા કે જે માહિતીને અનુક્રમિત તેમજ ઝડપી અનુક્રમણિતા આધારિત એક્સેસ આપી શકે.

Hashing : માહિતીને પુનઃ પ્રાપ્ત કરવાની પદ્ધતિ કે જે શોધ ચાવીને સંગ્રહિત સરનામું તેમજ સંગ્રહિત માહિતીને ઝડપી પુનઃ પ્રાપ્તિ કરી આપે છે.

Index : માહિતીનું તાર્કિક તેમજ ભૌતિક સરનામું.

RAID : સ્વતંત્ર ડીસ્કનું માધ્યમ કે જે એકથી વધુ ડ્રાઈવ ધરાવતું હોય અને તેના નિયામક પણ વિવિધ હોય, જેનો મુખ્ય ઉદ્દેશ્ય માહિતીનું વિભાજન કરીને ઝડપી પુનઃપ્રાપ્તિ કરાવીને માહિતીનું પુનરાવર્તન અટકાવવું.

Sequential Access : કોઈપણ માહિતીને ક્રમમાં ગોઠવી પ્રથમ, દ્વિતીય તે ક્રમમાં ગોઠવણી કરે તે.

2.9 સંદર્ભ અને વિશેષ વાચન (REFERENCE AND FURTHER READING)

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

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

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

Folk,, Michael, J. [et.al.] (2004). *File Structures: An Object-oriented Approach with C++.* New Delhi: Pearson Education.

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

O'Brien, James A. (1997). *Introduction to Information Systems.* Irwin: The McGraw-Hill Company.