

Database Management System (1333204) - Summer 2025 Solution

Milav Dabgar

May 17, 2025

પ્રશ્ન 1(અ) [3 ગુણ]

ટૂંકી નોંધ લખો: ડેટા ડિક્શનરી

જવાબ

ડેટા ડિક્શનરી એ કેન્દ્રીય ભંડાર છે જે ડેટાબેઝ બંધારણ, તત્વો અને સંબંધો વિશે મેટાડેટા સંગ્રહિત કરે છે.

કોષ્ટક 1. ડેટા ડિક્શનરી ઘટકો

ઘટક	વર્ણન
ટેબલ નામો	ડેટાબેઝમાં બધા ટેબલોની યાદી
કોલમ વિગતો	ડેટા પ્રકારો, મર્યાદાઓ, લંબાઈ
સંબંધો	ફોરેન કી કનેક્શન્સ
ઇન્ડેક્સ	પ્રદર્શન ઓપ્ટિમાઇઝેશન બંધારણો

મુખ્ય લક્ષણો:

- મેટાડેટા સ્ટોરેજ: ડેટા બંધારણ વિશે માહિતી સમાવે છે
- ડેટા અખંડિતતા: સુસંગતતા નિયમો અને મર્યાદાઓ જાળવે છે
- દસ્તાવેજીકરણ: વ્યાપક ડેટાબેઝ દસ્તાવેજીકરણ પ્રદાન કરે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "ડેટા ડિક્શનરી વિગતો આપે"

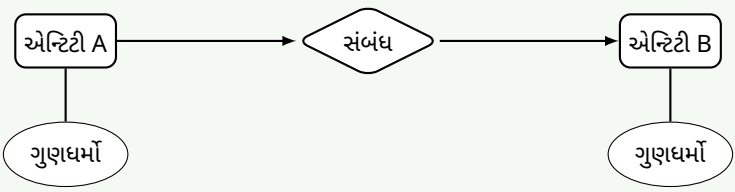
પ્રશ્ન 1(બ) [4 ગુણ]

વ્યાખ્યા આપો (i) E-R મોડેલ (ii) એન્ટિટી (iii) એન્ટિટી સેટ અને (iv) ગુણધર્મો

જવાબ

કોષ્ટક 2. ER મોડેલ વ્યાખ્યાઓ

શબ્દ	વ્યાખ્યા
E-R મોડેલ	એન્ટિટી અને સંબંધોનો ઉપયોગ કરતો કન્સેપ્ચ્યુઅલ ડેટા મોડેલ
એન્ટિટી	સ્વતંત્ર અસ્તિત્વ ધરાવતો વાસ્તવિક વિશ્વનો ઓબ્જેક્ટ
એન્ટિટી સેટ	સમાન પ્રકારની સમાન એન્ટિટીઓનો સંગ્રહ
ગુણધર્મો	એન્ટિટીની લાક્ષણિકતાઓનું વર્ણન કરતા ગુણધર્મો



આકૃતિ 1. ER મોડેલ ઘટકો

મુખ્ય મુદ્દાઓ:

- કન્સેપ્ચ્યુઅલ ડિઝાઇન: ઉચ્ચ સ્તરનો ડેટાબેઝ ડિઝાઇન અભિગમ
- વિઝ્યુઅલ રજૂઆત: સ્પષ્ટ સમજ માટે આકૃતિઓનો ઉપયોગ

મેમરી ટ્રીક

મેમરી ટ્રીક: "એન્ટિટી સંબંધો અર્થપૂર્ણ રીતે"

પ્રશ્ન 1(ક) [7 ગુણ]

DBMS ના ફાયદા સમજાવો

જવાબ

કોષ્ટક 3. DBMS ફાયદા

ફાયદો	લાભ
ડેટા સ્વતંત્રતા	એપ્લિકેશન ડેટા સ્ટ્રક્ચર ફેરફારોથી અલગ
ડેટા શેરિંગ	બહુવિધ વપરાશકર્તાઓ એકસાથે સમાન ડેટા એક્સેસ કરે
ડેટા સુરક્ષા	એક્સેસ કંટ્રોલ અને પ્રમાણીકરણ પદ્ધતિઓ
ડેટા અખંડિતતા	મર્યાદાઓ દ્વારા સુસંગતતા જાળવવામાં આવે છે
બેકઅપ અને રિકવરી	આપોઆપ ડેટા સંરક્ષણ અને પુનઃસ્થાપન
ઘટાડેલી રીડ-ડેન્સી	ડુપ્લિકેટ ડેટા સ્ટોરેજ દૂર કરે છે

મુખ્ય લાભો:

- કેન્દ્રીકૃત નિયંત્રણ: ડેટા વ્યવસ્થાપનનો એક બિંદુ
- ખર્ચ અસરકારકતા: વિકાસ અને જાળવણીનો ખર્ચ ઘટાડે છે
- ડેટા સુસંગતતા: એપ્લિકેશન-સમાં એકસમાન ડેટા સુનિશ્ચિત કરે છે
- સંગામિત એક્સેસ: બહુવિધ વપરાશકર્તાઓ એકસાથે કામ કરી શકે છે
- ક્વેરી ઓપ્ટિમાઇઝેશન: કાર્યક્ષમ ડેટા પુનઃપ્રાપ્તિ પદ્ધતિઓ

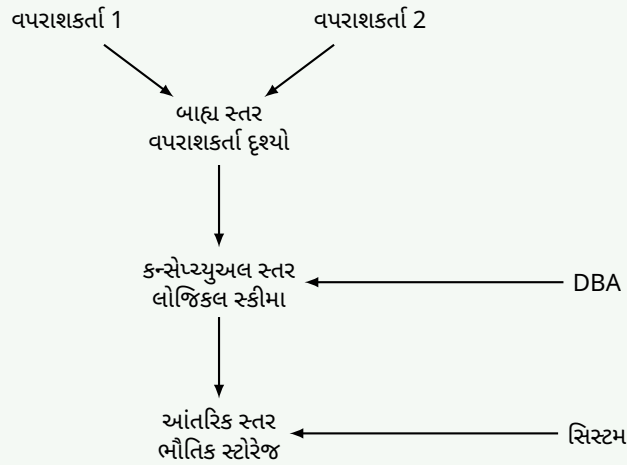
મેમરી ટ્રીક

મેમરી ટ્રીક: "ડેટાબેઝ બિઝનેસને બહેતર બનાવે"

પ્રશ્ન 1(ક) અથવા [7 ગુણ]

DBMS નું આર્કિટેક્ચર સમજાવો

જવાબ



આકૃતિ 2. ત્રણ-સ્તરીય DBMS આર્કિટેક્ચર

કોષ્ટક 4. આર્કિટેક્ચર સ્તરો

સ્તર	હેતુ	વપરાશકર્તાઓ
બાહ્ય	વ્યક્તિગત વપરાશકર્તા દૃશ્યો	અંતિમ વપરાશકર્તાઓ, એપ્લિકેશન્સ
કન્સેપ્ચ્યુઅલ	સંપૂર્ણ લોજિકલ બંધારણ	ડેટાબેઝ એડમિનિસ્ટ્રેટર
આંતરિક	ભૌતિક સ્ટોરેજ વિગતો	સિસ્ટમ પ્રોગ્રામર્સ

મુખ્ય લક્ષણો:

- ડેટા સ્વતંત્રતા: એક સ્તરે ફેરફારો અન્યને અસર કરતા નથી
- સુરક્ષા: વિવિધ વપરાશકર્તાઓ માટે વિવિધ એક્સેસ સ્તરો
- અમૂર્તતા: વપરાશકર્તાઓથી જટિલતા છુપાવે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "બાહ્ય કન્સેપ્ચ્યુઅલ આંતરિક આર્કિટેક્ચર"

પ્રશ્ન 2(અ) [3 ગુણ]

UNIQUE KEY અને PRIMARY KEY સમજાવો

જવાબ

કોષ્ટક 5. કી સરખામણી

લક્ષણ	PRIMARY KEY	UNIQUE KEY
Null મૂલ્યો	મંજૂર નથી	એક null મંજૂર
ટેબલ દીઠ સંખ્યા	માત્ર એક	બહુવિધ મંજૂર
ઇન્ડેક્સ બનાવટ	આપોઆપ clustered	આપોઆપ non-clustered
હેતુ	એન્ટિટી ઓળખ	ડેટા વિશિષ્ટતા

મુખ્ય તફાવતો:

- પ્રાથમિક કી: દરેક રેકૉર્ડને વિશિષ્ટ રીતે ઓળખે છે, null હોઈ શકતી નથી
- યુનિક કી: વિશિષ્ટતા સુનિશ્ચિત કરે છે પણ એક null મૂલ્યની મંજૂરી આપે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "પ્રાથમિક નલને અટકાવે, યુનિક નલને સમજે"

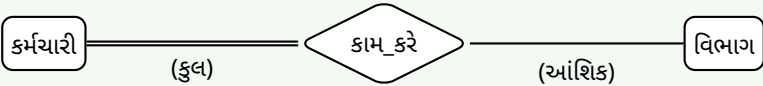
પ્રશ્ન 2(બ) [4 ગુણ]

ER ડાયાગ્રામમાં એન્ટિટીની Participation પર ટૂંકી નોંધ લખો

જવાબ

કોષ્ટક 6. Participation પ્રકારો

પ્રકાર	વર્ણન	પ્રતીક
કુલ Participation	દરેક એન્ટિટી સહભાગી થવી જ જોઈએ	ડબલ લાઇન
આંશિક Participation	કેટલીક એન્ટિટી સહભાગી ન પણ થઈ શકે	સિંગલ લાઇન



આકૃતિ 3. Participation ઉદાહરણ

મુખ્ય સંકેતો:

- ફરજિયાત Participation: દરેક ઇન્સ્ટન્સ સંકળાયેલું હોવું જ જોઈએ
- વૈકલ્પિક Participation: કેટલાક ઇન્સ્ટન્સ સંકળાયેલા ન હોઈ શકે
- બિઝનેસ નિયમો: વાસ્તવિક વિશ્વની મર્યાદાઓને પ્રતિબિંબિત કરે છે

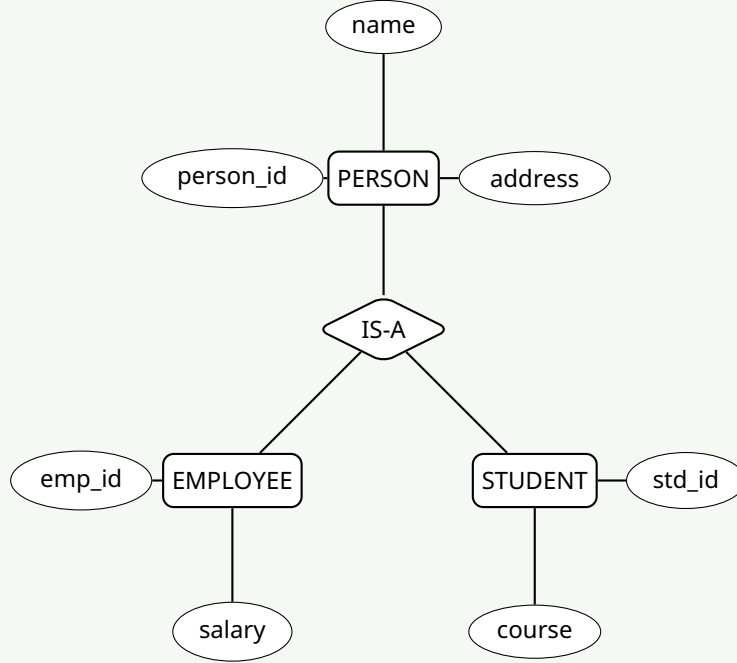
મેમરી ટ્રીક

મેમરી ટ્રીક: "કુલ Participation બધાની જરૂર"

પ્રશ્ન 2(ક) [7 ગુણ]

ER ડાયાગ્રામ માટે Generalization concept વિગતવાર વર્ણન કરો

જવાબ



આકૃતિ 4. Generalization ઉદાહરણ

કોષ્ટક 7. Generalization લાક્ષણિકતાઓ

પાસું	વર્ણન
બોટમ-અપ પ્રક્રિયા	સમાન એન્ટિટીઓને સુપરકલાસમાં જોડે છે
વારસાગતતા	સબકલાસ સુપરકલાસ ગુણધર્મો વારસે મેળવે છે
વિશેષીકરણ	Generalization ની વિપરીત પ્રક્રિયા
ઓવરલેપ મર્યાદાઓ	અલગ અથવા ઓવરલેપિંગ સબકલાસ

મુખ્ય લક્ષણો:

- ગુણધર્મ વારસાગતતા: સામાન્ય ગુણધર્મો સુપરકલાસમાં ખસેડવામાં આવે છે
- સંબંધ વારસાગતતા: સંબંધો પણ વારસામાં મળે છે
- મર્યાદા પ્રકારો: કુલ/આંશિક, અલગ/ઓવરલેપિંગ
- ISA સંબંધ: "is-a" કનેક્શનને રજૂ કરે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "સામાન્યીકરણ સમાન એન્ટિટીઓને જૂથ બનાવે"

પ્રશ્ન 2(અ) અથવા [3 ગુણ]

ER ડાયાગ્રામમાં મેપિંગ કાર્ડિનાલિટી સમજાવો

જવાબ

કોષ્ટક 8. કાર્ડિનાલિટી પ્રકારો

પ્રકાર	વર્ણન	ઉદાહરણ
એક-થી-એક (1:1)	એક એન્ટિટી અન્ય એક સાથે સંબંધિત	વ્યક્તિ-પાસપોર્ટ
એક-થી-ઘણા (1:M)	એક એન્ટિટી ઘણા અન્ય સાથે સંબંધિત	વિભાગ-કર્મચારી
ઘણા-થી-એક (M:1)	ઘણી એન્ટિટી એક સાથે સંબંધિત	કર્મચારી-વિભાગ
ઘણા-થી-ઘણા (M:N)	ઘણી એન્ટિટી ઘણા સાથે સંબંધિત	વિદ્યાર્થી-કોર્સ

મુખ્ય સંકેતો:

- સંબંધ મર્યાદાઓ: એન્ટિટી કેવી રીતે સંબંધિત થઈ શકે છે તે વ્યાખ્યાયિત કરે છે
- બિઝનેસ નિયમો: વાસ્તવિક વિશ્વ સંબંધ મર્યાદાઓને પ્રતિબિંબિત કરે છે

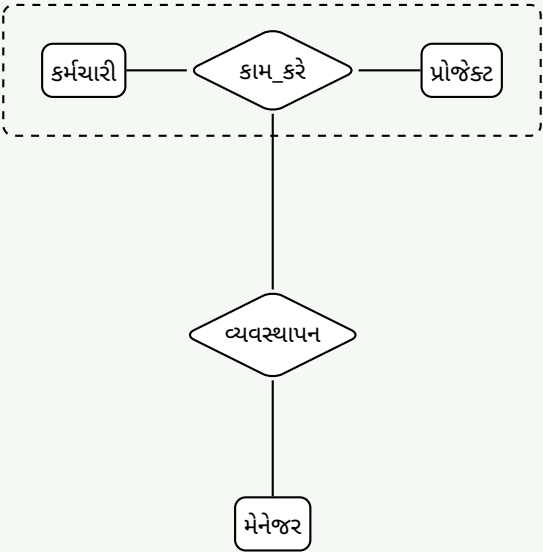
મેમરી ટ્રીક

મેમરી ટ્રીક: "એક કે ઘણા મેપિંગ મહત્વપૂર્ણ"

પ્રશ્ન 2(બ) અથવા [4 ગુણ]

E-R સાચાગ્રામમાં Aggregation સમજાવો

જવાબ



આકૃતિ 5. Aggregation ઉદાહરણ

મુખ્ય લક્ષણો:

- સંબંધ એન્ટિટી તરીકે: સંબંધ સેટને એન્ટિટી તરીકે ગણે છે
- ઉચ્ચ સ્તરના સંબંધો: સંબંધો વચ્ચે સંબંધોની મંજૂરી આપે છે
- જટિલ મોડેલિંગ: અદ્યતન બિઝનેસ દૃશ્યોને હેન્ડલ કરે છે
- અમૂર્ત પદ્ધતિ: જટિલ સંબંધોને સરળ બનાવે છે

કોષ્ટક 9. Aggregation લાભો

લાભ	વર્ણન
મોડેલિંગ લવચીકતા	જટિલ સંબંધોને હેન્ડલ કરે છે
અર્થપૂર્ણ સ્પષ્ટતા	બિઝનેસ નિયમોની સ્પષ્ટ રજૂઆત
ડિઝાઇન સરળતા	મોડેલ જટિલતા ઘટાડે છે

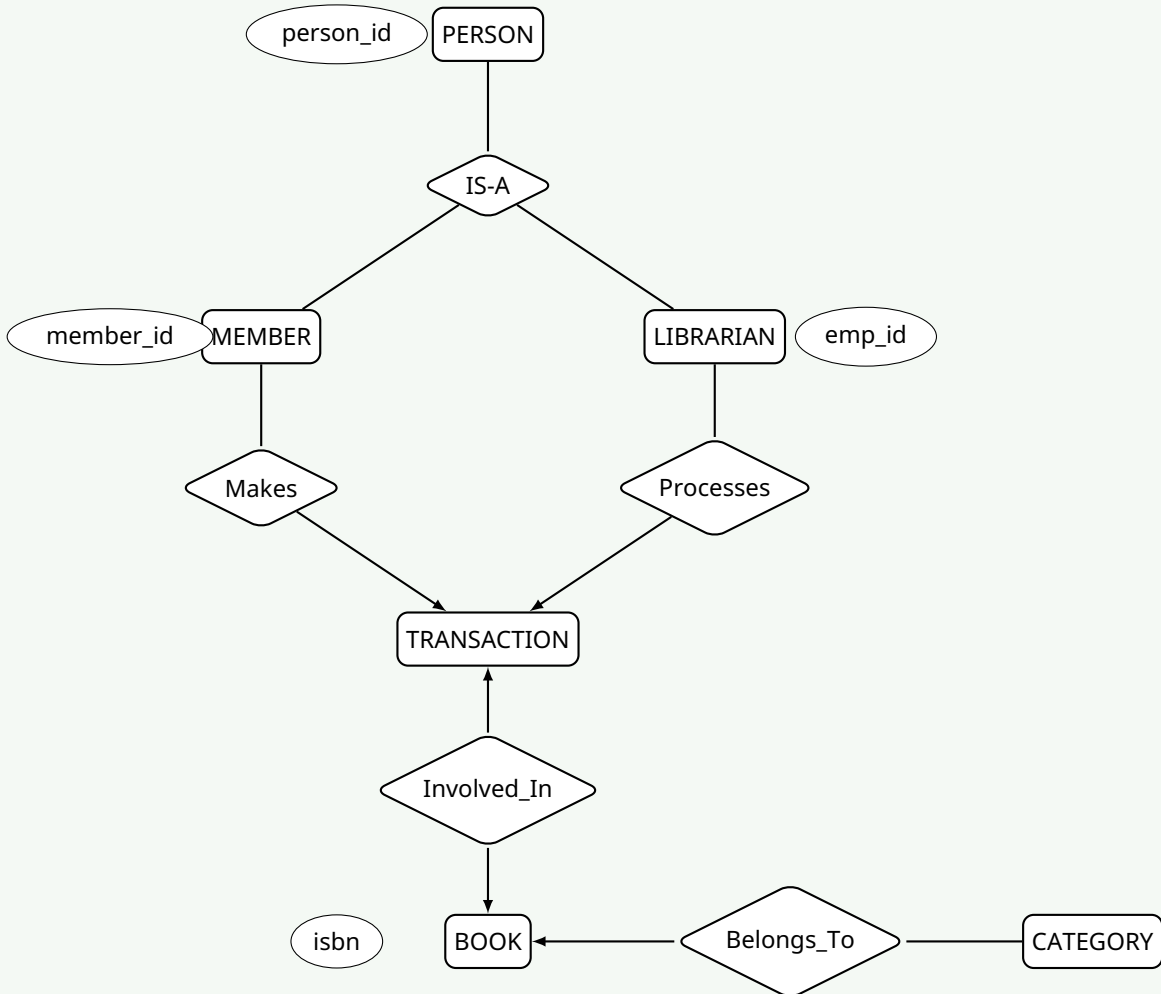
મેમરી ટ્રીક

મેમરી ટ્રીક: "એકીકરણ અઘતન સંગઠનોને અમૂર્ત બનાવે"

પ્રશ્ન 2(ક) અથવા [7 ગુણ]

Enhanced ER મોડેલનો ઉપયોગ કરીને લાઇબ્રેરી મેનેજમેન્ટ સિસ્ટમનો ER ડાયાગ્રામ દોરો

જવાબ



આકૃતિ 6. લાઇબ્રેરી મેનેજમેન્ટ સિસ્ટમ

વપરાયેલ Enhanced ER લક્ષણો:

- સામાન્યીકરણ: મેમ્બર અને લાઇબ્રેરિયન સબકલાસ સાથે વ્યક્તિ સુપરકલાસ

- વિશેષીકરણ: વિવિધ વ્યક્તિ પ્રકારો માટે વિવિધ ગુણધર્મો
- એકીકરણ: બહુવિધ એન્ટિટી સાથે Transaction સંબંધ
- બહુવિધ વારસાગતતા: જટિલ સંબંધ હેન્ડલિંગ

મેમરી ટ્રીક

મેમરી ટ્રીક: "લાઇબ્રેરી સાહિત્યને તાર્કિક રીતે જોડે"

પ્રશ્ન 3(અ) [3 ગુણ]

SQL ડેટા પ્રકાર સમજાવો

જવાબ

કોષ્ટક 10. સામાન્ય SQL ડેટા પ્રકારો

કેટેગરી	ડેટા પ્રકાર	વર્ણન
સંખ્યાત્મક	INT, DECIMAL, FLOAT	સંખ્યાઓ સંગ્રહિત કરે
અક્ષર	CHAR, VARCHAR, TEXT	ટેક્સ્ટ સંગ્રહિત કરે
તારીખ/સમય	DATE, TIME, DATETIME	સમયગત ડેટા સંગ્રહિત કરે
બુલિયન	BOOLEAN	સાચું/ખોટું સંગ્રહિત કરે

મુખ્ય મુદ્દાઓ:

- ડેટા અખંડિતતા: યોગ્ય ડેટા સ્ટોરેજ સુનિશ્ચિત કરે છે
- સ્ટોરેજ ઓપ્ટિમાઇઝેશન: યોગ્ય કદ ફાળવણી
- માન્યતા: આપોઆપ ડેટા પ્રકાર તપાસ

મેમરી ટ્રીક

મેમરી ટ્રીક: "ડેટા પ્રકારો સ્ટોરેજ વ્યાખ્યાયિત કરે"

પ્રશ્ન 3(બ) [4 ગુણ]

DROP અને TRUNCATE COMMAND સરખામણી કરો

જવાબ

કોષ્ટક 11. DROP vs TRUNCATE સરખામણી

લક્ષણ	DROP	TRUNCATE
ઓપરેશન	ટેબલ સ્ટ્રક્ચર દૂર કરે	માત્ર બધો ડેટા દૂર કરે
રોલબેક	રોલબેક કરી શકાતું નથી	રોલબેક કરી શકાય (ટ્રાન્ઝેક્શનમાં)
ઝડપ	ધીમું	ઝડપી
ટ્રિગર્સ	ટ્રિગર્સ ચલાવે છે	ટ્રિગર્સ ચલાવતું નથી
વ્હેર ક્લોઝ	લાગુ નથી	સપોર્ટ કરતું નથી
ઓટો-ઇન્ક્રિમેન્ટ	રીસેટ થાય છે	પ્રારંભિક વેલ્યુ પર રીસેટ થાય છે

કોડ ઉદાહરણો:

```

1 -- DROP command
2 DROP TABLE student;
3
4 -- TRUNCATE command
5 TRUNCATE TABLE student;
```

મુખ્ય તફાવતો:

- સ્ટ્રક્ચર પ્રભાવ: DROP બધું દૂર કરે છે, TRUNCATE સ્ટ્રક્ચર રાખે છે
- પ્રદર્શન: TRUNCATE મોટા ટેબલો માટે ઝડપી છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "DROP નાશ કરે, TRUNCATE કાપે"

પ્રશ્ન 3(ક) [7 ગુણ]

નીચેના Relational Schema અને નીચેના પ્રશ્નો માટે Relational Algebra Expression આપો વિદ્યાર્થીઓ (નામ, SPI, DOB, નોંધણી નંબર)

જવાબ

રિલેશનલ આલ્જિબ્રા એક્સપ્રેશન્સ:

i) એવા તમામ વિદ્યાર્થીઓની યાદી બનાવો કે જેમનું SPI 6.0 કરતાં ઓછું છે:

$$\sigma_{SPI < 6.0}(\text{વિદ્યાર્થીઓ})$$

ii) વિદ્યાર્થીનું નામ જેની નોંધણી નંબર 006 ધરાવે છે:

$$\pi_{\text{નામ}}(\sigma_{\text{નોંધણી_નંબર LIKE '%006\%'}}(\text{વિદ્યાર્થીઓ}))$$

iii) સમાન DOB ધરાવતા તમામ વિદ્યાર્થીઓની યાદી બનાવો:

$$\text{વિદ્યાર્થીઓ} \bowtie_{\substack{\text{વિદ્યાર્થીઓ.DOB} = S2.DOB \\ \wedge \text{વિદ્યાર્થીઓ.નોંધણી_નંબર} \neq S2.નોંધણી_નંબર}} (\rho_{S2}(\text{વિદ્યાર્થીઓ}))$$

iv) સમાન અક્ષરથી શરૂ થતા વિદ્યાર્થીઓનું નામ દર્શાવો:

$$\pi_{\text{નામ}}(\text{વિદ્યાર્થીઓ} \bowtie_{\substack{\text{SUBSTR}(\text{વિદ્યાર્થીઓ.નામ}, 1, 1) = \text{SUBSTR}(S2.નામ, 1, 1) \\ \wedge \text{વિદ્યાર્થીઓ.નોંધણી_નંબર} \neq S2.નોંધણી_નંબર}} (\rho_{S2}(\text{વિદ્યાર્થીઓ})))$$

કોષ્ટક 12. વપરાયેલ રિલેશનલ આલ્જિબ્રા ઓપરેટર્સ

ઓપરેટર	પ્રતીક	હેતુ
પસંદગી	σ	શરત આધારિત પંક્તિઓ ફિલ્ટર કરે
પ્રોજેક્શન	π	ચોક્કસ કોલમ પસંદ કરે
જોઇન	\bowtie	સંબંધિત ટ્યુપલ્સ સંયોજિત કરે
નામ બદલવું	ρ	રિલેશન્સ/એટ્રિબ્યુટ્સનું નામ બદલે

મેમરી ટ્રીક

મેમરી ટ્રીક: "પસંદ કરો પ્રોજેક્ટ કરો જોડો નામ બદલો"

પ્રશ્ન 3(અ) અથવા [3 ગુણ]

ઉદાહરણ સાથે Grant અને Revoke આદેશનો ઉપયોગ સમજાવો

જવાબ

કોડ ઉદાહરણો:

```
1 -- GRANT command
2 GRANT SELECT, INSERT ON student TO user1;
3 GRANT ALL PRIVILEGES ON database1 TO user2;
4
5 -- REVOKE command
6 REVOKE INSERT ON student FROM user1;
7 REVOKE ALL PRIVILEGES ON database1 FROM user2;
```

મુખ્ય લક્ષણો:

- એક્સેસ કંટ્રોલ: વપરાશકર્તા અનુમતિઓ સંચાલિત કરે છે
- સુરક્ષા: અનધિકૃત એક્સેસ અટકાવે છે
- ગ્રેન્યુલર કંટ્રોલ: ચોક્કસ વિશેષાધિકાર અસાઇનમેન્ટ

કોષ્ટક 13. સામાન્ય વિશેષાધિકારો

વિશેષાધિકાર	વર્ણન
SELECT	ડેટા વાંચે
INSERT	નવા રેકૉર્ડ ઉમેરે
UPDATE	હાલનો ડેટા બદલે
DELETE	રેકૉર્ડ દૂર કરે
ALL	સંપૂર્ણ એક્સેસ

મેમરી ટ્રીક

મેમરી ટ્રીક: "Grant આપે, Revoke દૂર કરે"

પ્રશ્ન 3(બ) અથવા [4 ગુણ]

ઉદાહરણ સાથે DML આદેશોનું વર્ણન કરો

જવાબ

કોષ્ટક 14. DML આદેશો

આદેશ	હેતુ	ઉદાહરણ
INSERT	નવા રેકૉર્ડ ઉમેરે	INSERT INTO student...
UPDATE	હાલનો ડેટા બદલે	UPDATE student SET...
DELETE	રેકૉર્ડ દૂર કરે	DELETE FROM student...
SELECT	ડેટા પુનઃપ્રાપ્ત કરે	SELECT * FROM student...

કોડ ઉદાહરણો:

```
1 -- INSERT command
```

```

2 INSERT INTO Students (name, spi, dob)
3 VALUES ('Alice', 8.5, '2000-05-15');
4
5 -- UPDATE command
6 UPDATE Students SET spi = 9.0
7 WHERE name = 'Alice';
8
9 -- DELETE command
10 DELETE FROM Students
11 WHERE spi < 6.0;
12
13 -- SELECT command
14 SELECT name, spi FROM Students
15 WHERE spi > 8.0;

```

મુખ્ય લક્ષણો:

- ડેટા મેનિપ્યુલેશન: મુખ્ય ડેટાબેઝ ઓપરેશન્સ
- ટ્રાન્ઝેક્શન સપોર્ટ: રોલબેક કરી શકાય છે
- શરતી ઓપરેશન્સ: WHERE ક્લોઝ સપોર્ટ

મેમરી ટ્રીક

મેમરી ટ્રીક: "Insert Update Delete Select"

પ્રશ્ન 3(ક) અથવા [7 ગુણ]

DBMS ના તમામ કન્વર્ઝન ફંક્શનની યાદી બનાવો અને તેમાંથી કોઈપણ ત્રણને વિગતવાર સમજાવો

જવાબ

કોષ્ટક 15. કન્વર્ઝન ફંક્શન્સ

ફંક્શન	હેતુ	ઉદાહરણ
TO_CHAR	કેરેક્ટરમાં કન્વર્ટ કરે	TO_CHAR(sysdate)
TO_DATE	તારીખમાં કન્વર્ટ કરે	TO_DATE('15-05-2025')
TO_NUMBER	નંબરમાં કન્વર્ટ કરે	TO_NUMBER('123.45')
CAST	સામાન્ય કન્વર્ઝન	CAST('123' AS INT)
CONVERT	ડેટા પ્રકાર કન્વર્ઝન	CONVERT(varchar, 123)

ત્રણ ફંક્શન્સની વિગતવાર સમજૂતી:

1. TO_CHAR ફંક્શન:

- હેતુ: તારીખો અને નંબરોને કેરેક્ટર સ્ટ્રિંગમાં કન્વર્ટ કરે છે
- સિન્ટેક્સ: TO_CHAR(value, format)
- ઉપયોગ: તારીખ ફોર્મેટિંગ, ચોક્કસ પેટર્ન સાથે નંબર ફોર્મેટિંગ

2. TO_DATE ફંક્શન:

- હેતુ: કેરેક્ટર સ્ટ્રિંગને તારીખ વેલ્યુમાં કન્વર્ટ કરે છે
- સિન્ટેક્સ: TO_DATE(string, format)
- ઉપયોગ: ચોક્કસ ફોર્મેટ સાથે સ્ટ્રિંગ થી તારીખ કન્વર્ઝન

3. TO_NUMBER ફંક્શન:

- હેતુ: કેરેક્ટર સ્ટ્રિંગને સંખ્યાત્મક વેલ્યુમાં કન્વર્ટ કરે છે
- સિન્ટેક્સ: TO_NUMBER(string, format)
- ઉપયોગ: ગણતરીઓ માટે સ્ટ્રિંગ થી નંબર કન્વર્ઝન

મુખ્ય લાભો:

- ડેટા પ્રકાર લવચીકતા: પ્રકારો વચ્ચે સહજ કન્વર્ઝન
- ફોર્મેટ કંટ્રોલ: ચોક્કસ ફોર્મેટિંગ વિકલ્પો
- એરર હેન્ડલિંગ: કન્વર્ઝન દરમિયાન માન્યતા

મેમરી ટ્રીક

મેમરી ટ્રીક: "કેરેક્ટર્સ તારીખો નંબર્સ કન્વર્ટ કરો"

પ્રશ્ન 4(અ) [3 ગુણ]**ટૂંકી નોંધ લખો: Domain Integrity Constraint****જવાબ**

ડોમેન ઇન્ટીગ્રિટી કન્સ્ટ્રેઇન્ટ્સ સુનિશ્ચિત કરે છે કે ડેટા મૂલ્યો ચોક્કસ એટ્રિબ્યુટ્સ માટે સ્વીકાર્ય રેન્જ અને ફોર્મેટમાં છે.

કોષ્ટક 16. ડોમેન કન્સ્ટ્રેઇન્ટ પ્રકારો

કન્સ્ટ્રેઇન્ટ	હેતુ	ઉદાહરણ
CHECK	વેલ્યુ રેન્જ માન્યતા	CHECK (age >= 0 AND age <= 100)
NOT NULL	null વેલ્યુ અટકાવે	name VARCHAR(50) NOT NULL
DEFAULT	ડિફોલ્ટ વેલ્યુ સેટ કરે	status VARCHAR(10) DEFAULT 'Active'

મુખ્ય લક્ષણો:

- ડેટા માન્યતા: એન્ટ્રી સમયે ડેટા ગુણવત્તા સુનિશ્ચિત કરે છે
- બિઝનેસ નિયમો: ડોમેન-વિશિષ્ટ નિયમો લાગુ કરે છે
- આપોઆપ તપાસ: DML ઓપરેશન્સ દરમિયાન માન્યતા થાય છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "ડોમેન ડેટા સીમાઓ વ્યાખ્યાયિત કરે"

પ્રશ્ન 4(બ) [4 ગુણ]**DBMS માં તમામ JOIN ની યાદી બનાવો અને કોઈપણ બે સમજાવો**

જવાબ

કોષ્ટક 17. JOIN ના પ્રકારો

JOIN પ્રકાર	વર્ણન
INNER JOIN	બંને ટેબલમાંથી મેચ થતા રેકૉર્ડ્સ આપે છે
LEFT JOIN	ડાબા ટેબલમાંથી તમામ રેકૉર્ડ્સ આપે છે
RIGHT JOIN	જમણા ટેબલમાંથી તમામ રેકૉર્ડ્સ આપે છે
FULL OUTER JOIN	બંને ટેબલમાંથી તમામ રેકૉર્ડ્સ આપે છે
CROSS JOIN	બંને ટેબલનું કાર્ટેશિયન ગુણાકાર
SELF JOIN	ટેબલ તેની પોતાની સાથે જોડાય છે

વિગતવાર સમજૂતી:

1. INNER JOIN:

```

1 SELECT s.name, c.course_name
2 FROM students s
3 INNER JOIN courses c ON s.course_id = c.course_id;

```

- બંને ટેબલમાંથી માત્ર મેચ થતા રેકૉર્ડ્સ આપે છે
- સૌથી સામાન્ય રીતે વપરાતો જોઈન પ્રકાર

2. LEFT JOIN:

```

1 SELECT s.name, c.course_name
2 FROM students s
3 LEFT JOIN courses c ON s.course_id = c.course_id;

```

- તમામ વિદ્યાર્થીઓ આપે છે, ભલે કોર્સ અસાઇન ન હોય
- મેચ ન થતા રેકૉર્ડ્સ માટે NULL વેલ્યુ

મેમરી ટ્રીક

મેમરી ટ્રીક: "ટેબલોને વિચારપૂર્વક જોડો"

પ્રશ્ન 4(ક) [7 ગુણ]

Functional Dependency નો કોન્સેપ્ટ વિગતવાર સમજાવો

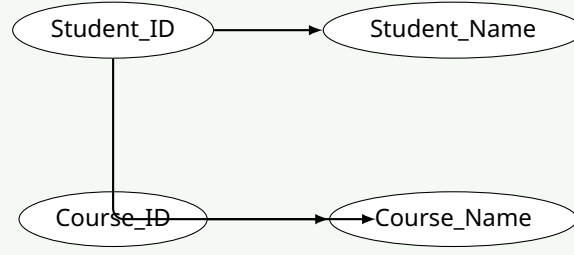
જવાબ

ફંક્શનલ ડિપેન્ડન્સી ત્યારે થાય છે જ્યારે એક એટ્રિબ્યુટનું મૂલ્ય અન્ય એટ્રિબ્યુટના મૂલ્યને વિશિષ્ટ રીતે નક્કી કરે છે.

નોટેશન: $A \rightarrow B$ (A ફંક્શનલી B ને નક્કી કરે છે)

કોષ્ટક 18. ફંક્શનલ ડિપેન્ડન્સીના પ્રકારો

પ્રકાર	વ્યાખ્યા	ઉદાહરણ
સંપૂર્ણ FD	LHS માં તમામ એટ્રિબ્યુટ્સ જરૂરી	{Student_ID, Course_ID} → Grade
આંશિક FD	કેટલાક LHS એટ્રિબ્યુટ્સ વધારાના	{Student_ID, Course_ID} → Student_Name
ટ્રાન્ઝિટિવ FD	અન્ય એટ્રિબ્યુટ દ્વારા પરોક્ષ નિર્ભરતા	Student_ID → Dept_ID → Dept_Name



આકૃતિ 7. ફંક્શનલ ડિપેન્ડન્સી ઉદાહરણ

મુખ્ય ગુણધર્મો:

- રીફ્લેક્સિવિટી: $A \rightarrow A$ (trivial dependency)
- ઓગમેન્ટેશન: જો $A \rightarrow B$, તો $AC \rightarrow BC$
- ટ્રાન્ઝિટિવિટી: જો $A \rightarrow B$ અને $B \rightarrow C$, તો $A \rightarrow C$
- ડિકમ્પોઝિશન: જો $A \rightarrow BC$, તો $A \rightarrow B$ અને $A \rightarrow C$

મેમરી ટ્રીક

મેમરી ટ્રીક: "ફંક્શનલ ડિપેન્ડન્સી નક્કી કરે છે"

પ્રશ્ન 4(અ) અથવા [3 ગુણ]

ટૂંકી નોંધ લખો: Referential integrity Constraints

જવાબ

રેફરન્શિયલ ઇન્ટીગ્રિટી સુનિશ્ચિત કરે છે કે એક ટેબલમાં ફોરેન કી મૂલ્યો સંદર્ભિત ટેબલમાં હાલના પ્રાથમિક કી મૂલ્યોને અનુરૂપ છે.

કોષ્ટક 19. રેફરન્શિયલ ઇન્ટીગ્રિટી નિયમો

નિયમ	વર્ણન	ક્રિયા
INSERT નિયમ	ફોરેન કી પેરેન્ટમાં હોવી જોઈએ	અમાન્ય inserts નકારો
DELETE નિયમ	પેરેન્ટ રેકૉર્ડ ડિલીશન હેન્ડલ કરો	CASCADE, RESTRICT, SET NULL
UPDATE નિયમ	પ્રાથમિક કી અપડેટ્સ હેન્ડલ કરો	CASCADE, RESTRICT

કોડ ઉદાહરણ:

```

1 ALTER TABLE Orders
2 ADD CONSTRAINT FK_Customer
3 FOREIGN KEY (customer_id)
4 REFERENCES Customers(customer_id);
  
```

મુખ્ય લક્ષણો:

- ફોરેન કી કન્સ્ટ્રેઇન્ટ: સંબંધિત ટેબલોને જોડે છે
- ડેટા સુસંગતતા: ઓર્ફન રેકૉર્ડ્સ અટકાવે છે
- સંબંધ જાળવણી: ટેબલ સંબંધો સાચવે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "રેફરન્સને રિલેટેડ રેકૉર્ડ્સની જરૂર"

પ્રશ્ન 4(બ) અથવા [4 ગુણ]

રિલેશનલ આલ્જિબ્રાના union અને intersection ઓપરેશન્સ સમજાવો

જવાબ

કોષ્ટક 20. સેટ ઓપરેશન્સ સરખામણી

ઓપરેશન	પ્રતીક	વર્ણન	જરૂરિયાત
UNION	\cup	બંને રિલેશન્સમાંથી તમામ ટ્યુપલ્સ જોડે છે	યુનિયન સુસંગત
INTERSECTION	\cap	બંને રિલેશન્સમાં સામાન્ય ટ્યુપલ્સ	યુનિયન સુસંગત

Union ઓપરેશન:

- સિન્ટેક્સ: $R \cup S$
- પરિણામ: R અને S માંથી તમામ ટ્યુપલ્સ (ડુપ્લિકેટ્સ દૂર)
- જરૂરિયાત: એટ્રિબ્યુટ્સની સમાન સંખ્યા અને પ્રકારો

Intersection ઓપરેશન:

- સિન્ટેક્સ: $R \cap S$
- પરિણામ: R અને S બંનેમાં અસ્તિત્વ ધરાવતા ટ્યુપલ્સ
- જરૂરિયાત: યુનિયન સુસંગત રિલેશન્સ

ઉદાહરણ:

- 1 Students_CS \cup Students_IT = બંને વભિાગના તમામ વહિયાર્થીઓ
- 2 Students_CS \cap Students_IT = બંને વભિાગના વહિયાર્થીઓ

મેમરી ટ્રીક

મેમરી ટ્રીક: "યુનિયન જોડે, ઇન્ટરસેક્શન સામાન્ય ઓળખે"

પ્રશ્ન 4(ક) અથવા [7 ગુણ]

DBMS માં Normalization નો કોન્સેપ્ટ વિગતવાર સમજાવો

જવાબ

નોર્મલાઇઝેશન એ ડેટા રિડન્ડન્સી ઘટાડવા અને ડેટા અખંડિતતા સુધારવા માટે ડેટાબેઝ ટેબલોને વ્યવસ્થિત કરવાની પ્રક્રિયા છે.

કોષ્ટક 21. નોર્મલ ફોર્મ્સ

નોર્મલ ફોર્મ	જરૂરિયાતો	દૂર કરે છે
1NF	એટોમિક વેલ્યુ, રિપીટિંગ ગ્રુપ નહીં	મલ્ટીવેલ્યુડ એટ્રિબ્યુટ્સ
2NF	1NF + આંશિક નિર્ભરતા નહીં	આંશિક ફંક્શનલ ડિપેન્ડન્સી
3NF	2NF + ટ્રાન્ઝિટિવ નિર્ભરતા નહીં	ટ્રાન્ઝિટિવ ડિપેન્ડન્સી
BCNF	3NF + દરેક ડિટર્મિનન્ટ કેન્ડિડેટ કી	બાકીની વિસંગતતાઓ

નોર્મલાઇઝેશન પ્રક્રિયા:

સ્ટેપ 1 - ફર્સ્ટ નોર્મલ ફોર્મ (1NF):

- રિપીટિંગ ગ્રુપ્સ દૂર કરો
- દરેક સેલ સિંગલ વેલ્યુ ધરાવે છે
- દરેક રેકૉર્ડ વિશિષ્ટ છે

સ્ટેપ 2 - સેકન્ડ નોર્મલ ફોર્મ (2NF):

- 1NF માં હોવું જોઈએ
- આંશિક નિર્ભરતા દૂર કરો
- નોન-કી એટ્રિબ્યુટ્સ સંપૂર્ણપણે પ્રાથમિક કી પર નિર્ભર

સ્ટેપ 3 - થર્ડ નોર્મલ ફોર્મ (3NF):

- 2NF માં હોવું જોઈએ
- ટ્રાન્ઝિટિવ નિર્ભરતા દૂર કરો
- નોન-કી એટ્રિબ્યુટ્સ અન્ય નોન-કી એટ્રિબ્યુટ્સ પર નિર્ભર નથી

નોર્મલાઇઝેશનના ફાયદા:

- ઘટાડેલી રિડન્ડન્સી: ડુપ્લિકેટ ડેટા દૂર કરે છે
- ડેટા અખંડિતતા: સુસંગતતા જાળવે છે
- સ્ટોરેજ કાર્યક્ષમતા: સ્ટોરેજ જગ્યા ઘટાડે છે
- અપડેટ વિસંગતતાઓ: અસંગત અપડેટ્સ અટકાવે છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "નોર્મલાઇઝેશન સુધાર ટેબલો બનાવે"

પ્રશ્ન 5(અ) [3 ગુણ]

DBMS માં Normalization ની જરૂરિયાત વર્ણવો

જવાબ**કોષ્ટક 22. નોર્મલાઇઝેશન દ્વારા ઉકેલાતી સમસ્યાઓ**

સમસ્યા	વર્ણન	ઉકેલ
ઇન્સર્શન વિસંગતતા	સંપૂર્ણ માહિતી વિના ડેટા દાખલ કરી શકાતો નથી	અલગ ટેબલો
અપડેટ વિસંગતતા	એક ફેરફાર માટે બહુવિધ અપડેટ્સ	રિડન્ડન્સી દૂર કરો
ડિલીશન વિસંગતતા	ડિલીટ કરતી વખતે મહત્વપૂર્ણ ડેટા ગુમાવવો	નિર્ભરતા સાચવો

મુખ્ય જરૂરિયાતો:

- ડેટા સુસંગતતા: ડેટાબેઝમાં એકસમાન ડેટા સુનિશ્ચિત કરે છે
- સ્ટોરેજ ઓપ્ટિમાઇઝેશન: બિનજરૂરી સ્ટોરેજ ઘટાડે છે
- જાળવણી સરળતા: ડેટાબેઝ અપડેટ્સ સરળ બનાવે છે

ફાયદા:

- સુધારેલી ડેટા ગુણવત્તા: ભૂલો અને અસંગતતાઓ ઘટાડે છે
- લવચીક ડિઝાઇન: ફેરફાર અને વિસ્તરણ માટે સરળ
- વધુ સારું પ્રદર્શન: અપડેટ ઓપરેશન્સ માટે

મેમરી ટ્રીક

મેમરી ટ્રીક: "નોર્મલાઇઝેશન વ્યવસ્થિત સંસ્થાની જરૂર"

પ્રશ્ન 5(બ) [4 ગુણ]

DBMS માં Transaction ના ગુણધર્મો સમજાવો

જવાબ

કોષ્ટક 23. ACID ગુણધર્મો

ગુણધર્મ	વર્ણન	હેતુ
Atomicity	તમામ ઓપરેશન્સ સફળ થાય અથવા તમામ નિષ્ફળ	પૂર્ણતા સુનિશ્ચિત કરે
Consistency	ડેટાબેઝ માન્ય સ્થિતિમાં રહે છે	અખંડિતતા જાળવે
Isolation	સંગામી ટ્રાન્ઝેક્શન્સ દખલ કરતા નથી	સંઘર્ષ અટકાવે
Durability	કમિટ થયેલા ફેરફારો કાયમી છે	સ્થાયીતા સુનિશ્ચિત કરે

વિગતવાર સમજૂતી:

- **Atomicity:** ટ્રાન્ઝેક્શન અવિભાજ્ય એકમ છે. કાં તો બધું પૂર્ણ થાય અથવા કંઈ નહીં.
- **Consistency:** ડેટાબેઝ એક માન્ય સ્થિતિમાંથી બીજી સ્થિતિમાં જાય છે. તમામ અખંડિતતા મર્યાદાઓ જળવાય છે.
- **Isolation:** સંગામી ટ્રાન્ઝેક્શન્સ ક્રમિક રીતે ચાલતા હોય તેવું લાગે છે. મધ્યવર્તી સ્થિતિઓ અન્ય ટ્રાન્ઝેક્શન્સને દેખાતી નથી.
- **Durability:** એકવાર કમિટ થઈ ગયા પછી, ફેરફારો સિસ્ટમ નિષ્ફળતામાં પણ ટકી રહે છે. ડેટા કાયમી ધોરણે સંગ્રહિત થાય છે.

મેમરી ટ્રીક

મેમરી ટ્રીક: "ACID સાચો ડેટાબેઝ સુનિશ્ચિત કરે"

પ્રશ્ન 5(ક) [7 ગુણ]

View Serializability વિગતવાર સમજાવો

જવાબ

View Serializability નક્કી કરે છે કે શું સમવર્તી શિડ્યુલ સીરીયલ શિડ્યુલ જેવું જ પરિણામ આપે છે કે કેમ, તે રીડ અને રાઈટ ઓપરેશન્સ તપાસીને.

કોષ્ટક 24. વ્યુ ઇન્કિવેલેન્સ શરતો

શરત	વર્ણન
પ્રારંભિક વાંચન	સમાન ટ્રાન્ઝેક્શન્સ પ્રારંભિક મૂલ્યો વાંચે છે
અંતિમ લેખન	સમાન ટ્રાન્ઝેક્શન્સ અંતિમ લેખન કરે છે
મધ્યવર્તી વાંચન	સમાન લખનાર ટ્રાન્ઝેક્શન્સમાંથી મૂલ્યો વાંચો

મુખ્ય ખ્યાલો:

વ્યુ ઇન્કિવેલેન્સ શિડ્યુલ્સ: બે શિડ્યુલ્સ વ્યુ ઇન્કિવેલેન્સ છે જો:

1. દરેક ડેટા આઇટમ માટે, જો ટ્રાન્ઝેક્શન T એક શિડ્યુલમાં પ્રારંભિક મૂલ્ય વાંચે, તો તે બીજામાં પ્રારંભિક મૂલ્ય વાંચે
2. દરેક રીડ ઓપરેશન માટે, જો T એક શિડ્યુલમાં T' દ્વારા લખાયેલું મૂલ્ય વાંચે, તો બીજામાં પણ તે જ હોવું જોઈએ
3. દરેક ડેટા આઇટમ માટે, જો T એક શિડ્યુલમાં અંતિમ લેખન કરે, તો તે બીજામાં પણ અંતિમ લેખન કરે

View Serializability પરીક્ષણ:

- પ્રેસિડન્સ ગ્રાફ: ડાયરેક્ટેડ ગ્રાફ બનાવો
- સાયકલ ડિટેક્શન: ગ્રાફમાં સાયકલ તપાસો
- સંઘર્ષ વિશ્લેષણ: રીડ-રાઈટ સંઘર્ષો તપાસો

ઉદાહરણ વિશ્લેષણ:

- 1 Schedule S1: R1(X) W1(X) R2(X) W2(X)
- 2 Schedule S2: R1(X) R2(X) W1(X) W2(X)

Conflict Serializability સાથે સરખામણી:

- View serializability ઓછું પ્રતિબંધિત છે
- કેટલાક view serializable શિડ્યુલ્સ conflict serializable હોતા નથી
- પરીક્ષણ કરવું વધુ જટિલ

મેમરી ટ્રીક

મેમરી ટ્રીક: "વ્યુ માન્ય શિડ્યુલ્સ ચકાસે"

પ્રશ્ન 5(અ) અથવા [3 ગુણ]

કોઈપણ ડેટાબેઝ પર 2NF કરો

જવાબ

ઉદાહરણ: Student Course Database

મૂળ ટેબલ (2NF માં નથી):

- 1 Student_Course (Student_ID, Student_Name, Course_ID, Course_Name, Grade, Instructor)
- 2 Primary Key: {Student_ID, Course_ID}

ફંક્શનલ ડિપેન્ડન્સી:

- Student_ID → Student_Name (આંશિક નિર્ભરતા)
- Course_ID → Course_Name, Instructor (આંશિક નિર્ભરતા)
- {Student_ID, Course_ID} → Grade

2NF વિભાજન:

ટેબલ 1: Students

- 1 Students (Student_ID, Student_Name)
- 2 Primary Key: Student_ID

ટેબલ 2: Courses

- 1 Courses (Course_ID, Course_Name, Instructor)
- 2 Primary Key: Course_ID

ટેબલ 3: Enrollments

- 1 Enrollments (Student_ID, Course_ID, Grade)
- 2 Primary Key: {Student_ID, Course_ID}
- 3 Foreign Keys: Student_ID → Students, Course_ID → Courses

પરિણામ: તમામ આંશિક નિર્ભરતા દૂર કરવામાં આવી, હવે 2NF માં છે.

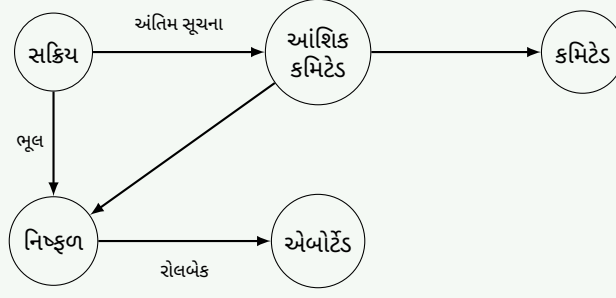
મેમરી ટ્રીક

મેમરી ટ્રીક: "બીજું નોર્મલ ફોર્મ નિર્ભરતા અલગ કરે"

પ્રશ્ન 5(બ) અથવા [4 ગુણ]

Transaction ના સ્ટેટસ સમજાવો

જવાબ



આકૃતિ 8. ટ્રાન્ઝેક્શન સ્ટેટ ડાયાગ્રામ

કોષ્ટક 25. ટ્રાન્ઝેક્શન સ્ટેટ્સ

સ્ટેટ	વર્ણન	ક્રિયાઓ
સક્રિય	ટ્રાન્ઝેક્શન ચાલી રહ્યું છે	રીડ/રાઈટ ઓપરેશન્સ
આંશિક કમિટેડ	અંતિમ સૂચના અમલમાં	કમિટની રાહ જોઈ રહ્યું છે
કમિટેડ	ટ્રાન્ઝેક્શન સફળતાપૂર્વક પૂર્ણ થયું	ફેરફારો કાયમી
નિષ્ફળ	સામાન્ય રીતે આગળ વધી શકતું નથી	ભૂલ આવી
એબોર્ટેડ	ટ્રાન્ઝેક્શન રોલબેક થયું	તમામ ફેરફારો રદ કર્યા

મુખ્ય મુદ્દાઓ:

- રિકવરી: સિસ્ટમ નિષ્ફળ સ્ટેટ્સમાંથી પુનઃપ્રાપ્ત કરી શકે છે
- ટકાઉપણું: કમિટ થયેલા ફેરફારો કાયમી છે
- એટોમિસિટી: એબોર્ટેડ ટ્રાન્ઝેક્શન્સ કોઈ નિશાન છોડતા નથી

મેમરી ટ્રીક

મેમરી ટ્રીક: "ટ્રાન્ઝેક્શન્સ સ્ટેટ્સમાંથી પસાર થાય"

પ્રશ્ન 5(ક) અથવા [7 ગુણ]

Conflict Serializability વિગતવાર સમજાવો

જવાબ

Conflict Serializability સુનિશ્ચિત કરે છે કે સમવર્તી શિડ્યુલ કોઈ સીરીયલ શિડ્યુલ સમાન છે, જે સંઘર્ષમય ઓપરેશન્સનું વિશ્લેષણ કરીને નક્કી થાય છે.

કોષ્ટક 26. સંઘર્ષમય ઓપરેશન્સ

ઓપરેશન જોડી	સંઘર્ષ પ્રકાર	કારણ
Read-Write	RW Conflict	લખતા પહેલા વાંચો
Write-Read	WR Conflict	વાંચતા પહેલા લખો
Write-Write	WW Conflict	બહુવિધ લખવાનું

Conflict Serializability પરીક્ષણ:

સ્ટેપ 1: સંઘર્ષો ઓળખો

- સમાન ડેટા આઈટમ પર ઓપરેશન્સની જોડી શોધો

- તપાસો કે ઓપરેશન્સ અલગ ટ્રાન્ઝેક્શન્સના છે કે નહીં
- નક્કી કરો કે ઓપરેશન્સ સંઘર્ષ કરે છે કે નહીં

સ્ટેપ 2: પ્રેસિડન્સ ગ્રાફ બનાવો

- નોડ્સ ટ્રાન્ઝેક્શન્સ રજૂ કરે છે
- ડાયરેક્ટેડ એજિસ સંઘર્ષો રજૂ કરે છે
- T_i થી T_j સુધીની એજ જો T_i T_j સાથે સંઘર્ષ કરે છે

સ્ટેપ 3: સાયકલ્સ માટે તપાસો

- જો ગ્રાફમાં કોઈ સાયકલ નથી → Conflict serializable
- જો ગ્રાફમાં સાયકલ છે → Conflict serializable નથી

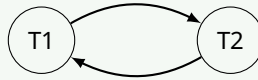
ઉદાહરણ વિશ્લેષણ:

```

1 Schedule: R1(A) W1(A) R2(A) W2(B) R1(B) W1(B)
2 Conflicts:
3 - W1(A) conflicts with R2(A) -> T1 before T2
4 - W2(B) conflicts with R1(B) -> T2 before T1
5 - W2(B) conflicts with W1(B) -> T2 before T1

```

પ્રેસિડન્સ ગ્રાફ:



આકૃતિ 9. પ્રેસિડન્સ ગ્રાફ (સાયકલ)

પરિણામ: સાયકલ ધરાવે છે, તેથી Conflict serializable નથી.

View Serializability સાથે સરખામણી:

- Conflict serializability વધુ પ્રતિબંધિત છે
- તમામ conflict serializable શિફ્ટ્સ view serializable છે
- View serializability કરતાં પરીક્ષણ કરવું સરળ છે

મેમરી ટ્રીક

મેમરી ટ્રીક: "સંઘર્ષો સાયકલ બનાવે, કાળજીપૂર્વક તપાસો"