

Subject Name (Gujarati)

1333204 -- Winter 2023

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

વ્યાખ્યા આપો: ફિલ્ડ, રેકૉર્ડ, મેટાડેટા

જવાબ

| શબ્દ | વ્યાખ્યા |
|----------|--|
| ફિલ્ડ | ડેટાબેઝ ટેબલમાં ચોક્કસ એટ્રિબ્યુટને રજૂ કરતી ડેટાની એક એકલ એકમ (દા.ત. નામ, ઉંમર, ID) |
| રેકૉર્ડ | સંબંધિત ફિલ્ડ્સનો સંપૂર્ણ સેટ જે એક એન્ટિટી ઇન્સ્ટન્સને રજૂ કરે છે (ટેબલમાં એક રો) |
| મેટાડેટા | ડેટા જે અન્ય ડેટાની રચના, ગુણધર્મો અને સંબંધોનું વર્ણન કરે છે ("ડેટા વિશે ડેટા") |

મેમરી ટ્રીક

"FRM: ફિલ્ડ્સ રો-અપ એઝ મેટાડેટા"

પ્રશ્ન 1(બ) [4 ગુણ]

વ્યાખ્યા લખો (i) E-R મોડલ (ii) એન્ટિટી (iii) એન્ટિટી સેટ અને (iv) એટ્રિબ્યુટ્સ

જવાબ

| શબ્દ | વ્યાખ્યા |
|--------------|--|
| E-R મોડલ | ડેટાબેઝ ડિઝાઇનનો ગ્રાફિકલ અભિગમ જે એન્ટિટીઝ, તેમના એટ્રિબ્યુટ્સ અને રિલેશનશીપને મોડેલ કરે છે |
| એન્ટિટી | એક વાસ્તવિક-વિશ્વ વસ્તુ, વિચાર અથવા ચીજ જેનું સ્વતંત્ર અસ્તિત્વ છે |
| એન્ટિટી સેટ | સમાન એન્ટિટીઓનો સંગ્રહ જે સમાન એટ્રિબ્યુટ્સ ધરાવે છે (ટેબલ તરીકે રજૂ કરાય છે) |
| એટ્રિબ્યુટ્સ | ગુણધર્મો અથવા લક્ષણો જે એન્ટિટીનું વર્ણન કરે છે (ટેબલના કોલમ તરીકે રજૂ કરાય છે) |

ENTITY_SET



contains



ENTITY

string

attribute1

number

attribute2

મેમરી ટ્રીક

“EAAA: એન્ટિટીસ એક્ટિવિસ્ટ એઝ એટ્રિબ્યુટ્સ”

પ્રશ્ન 1(ક) [7 ગુણ]

DBMS નાં ફાયદા અને ગેરફાયદા જણાવો.

જવાબ

| ફાયદા | ગેરફાયદા |
|--|---|
| ડેટા શેરિંગ: ઘણા વપરાશકર્તાઓ એક સાથે એક્સેસ કરી શકે છે | ખર્ચ: મોંઘા હાર્ડવેર/સોફ્ટવેર જરૂરિયાતો |
| ડેટા ઇન્ટિગ્રિટી: કન્સ્ટ્રેન્ટ્સ દ્વારા ચોકસાઈ જાળવે છે | જટિલતા: વિશિષ્ટ તાલીમની જરૂર પડે છે |
| ડેટા સિક્યુરિટી: પરમિશન દ્વારા એક્સેસ નિયંત્રિત કરે છે | પ્રદર્શન: મોટા ડેટાબેઝ માટે ધીમું હોઈ શકે છે |
| ડેટા ઇન્ડિપેન્ડન્સ: સ્ટોરેજ બદલવાથી એપ્લિકેશન પર અસર થતી નથી | નબળાઈ: કેન્દ્રીય નિષ્ફળતા બિંદુ ડેટા લોસનું જોખમ છે |
| ઘટાડેલ રિડન્ડન્સી: ડુપ્લીકેટ ડેટા દૂર કરે છે | કન્વર્ઝન ખર્ચ: ફાઇલ સિસ્ટમથી માઇગ્રેટ કરવું ખર્ચાળ છે |

મેમરી ટ્રીક

“SIDSR vs CCPVC” (શેરિંગ, ઇન્ટિગ્રિટી, ડેટા ઇન્ડિપેન્ડન્સ, સિક્યુરિટી, રિડન્ડન્સી vs કોસ્ટ, કોમ્પ્લેક્સિટી, પરફોર્મન્સ, વલ્નરેબિલિટી, કન્વર્ઝન)

પ્રશ્ન 1(ક) OR [7 ગુણ]

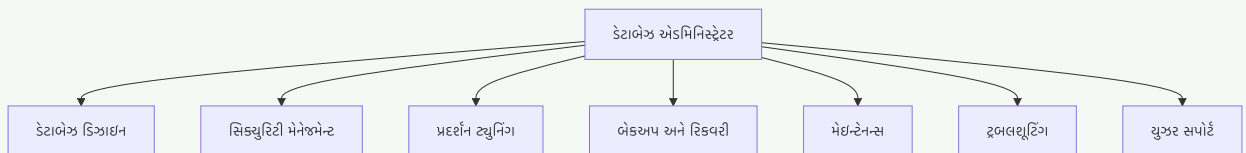
DBA નું પુરુનામ લખો. DBAની ભૂમિકા અને જવાબદારીઓ સમજાવો.

જવાબ

DBA: Database Administrator (ડેટાબેઝ એડમિનિસ્ટ્રેટર)

DBAની જવાબદારીઓ

ડેટાબેઝ ડિઝાઇન: કાર્યક્ષમ ડેટાબેઝ સ્કીમા બનાવે છે
 સિક્યુરિટી મેનેજમેન્ટ: યુઝર એક્સેસ કંટ્રોલ સેટ કરે છે
 પ્રદર્શન ટ્યુનિંગ: ક્વેરી અને ઇન્ડેક્સને ઓપ્ટિમાઇઝ કરે છે
 બેકઅપ અને રિકવરી: ડેટા સુરક્ષા યોજનાઓ લાગુ કરે છે
 મેઇન્ટેનન્સ: સોફ્ટવેર અપડેટ કરે છે અને પેચ લાગુ કરે છે
 ટ્રબલશૂટિંગ: ડેટાબેઝ સમસ્યાઓનો ઉકેલ કરે છે
 યુઝર સપોર્ટ: ડેટાબેઝ વપરાશકર્તાઓને તાલીમ આપે છે અને સહાય કરે છે



મેમરી ટ્રીક

“SPBT-MUS” (સિક્યુરિટી, પરફોર્મન્સ, બેકઅપ, ટ્રબલશૂટિંગ, મેઇન્ટેનન્સ, યુઝર સપોર્ટ)

પ્રશ્ન 2(અ) [3 ગુણ]

યોગ્ય ઉદાહરણ સાથે સિંગલ વેલ્યુડ અને મલ્ટી વેલ્યુડ એટ્રિબ્યુટ્સ વચ્ચેનો તફાવત સમજાવો

| એટ્રિબ્યુટ પ્રકાર | વર્ણન | ઉદાહરણો |
|-------------------|---|---------------------------------|
| સિંગલ-વેલ્યુડ | દરેક એન્ટિટી ઇન્સ્ટન્સ માટે માત્ર એક મૂલ્ય ધરાવે છે | Employee ID, જન્મતારીખ, નામ |
| મલ્ટી-વેલ્યુડ | એક જ એન્ટિટી માટે ઘણા મૂલ્યો ધરાવી શકે છે | ફોન નંબર, કૌશલ્યો, ઇમેઇલ એડ્રેસ |

| EMPLOYEE | |
|----------|---------------|
| string | emp_id |
| string | name |
| date | birth_date |
| string | phone_numbers |
| string | skills |

મેમરી ટ્રીક

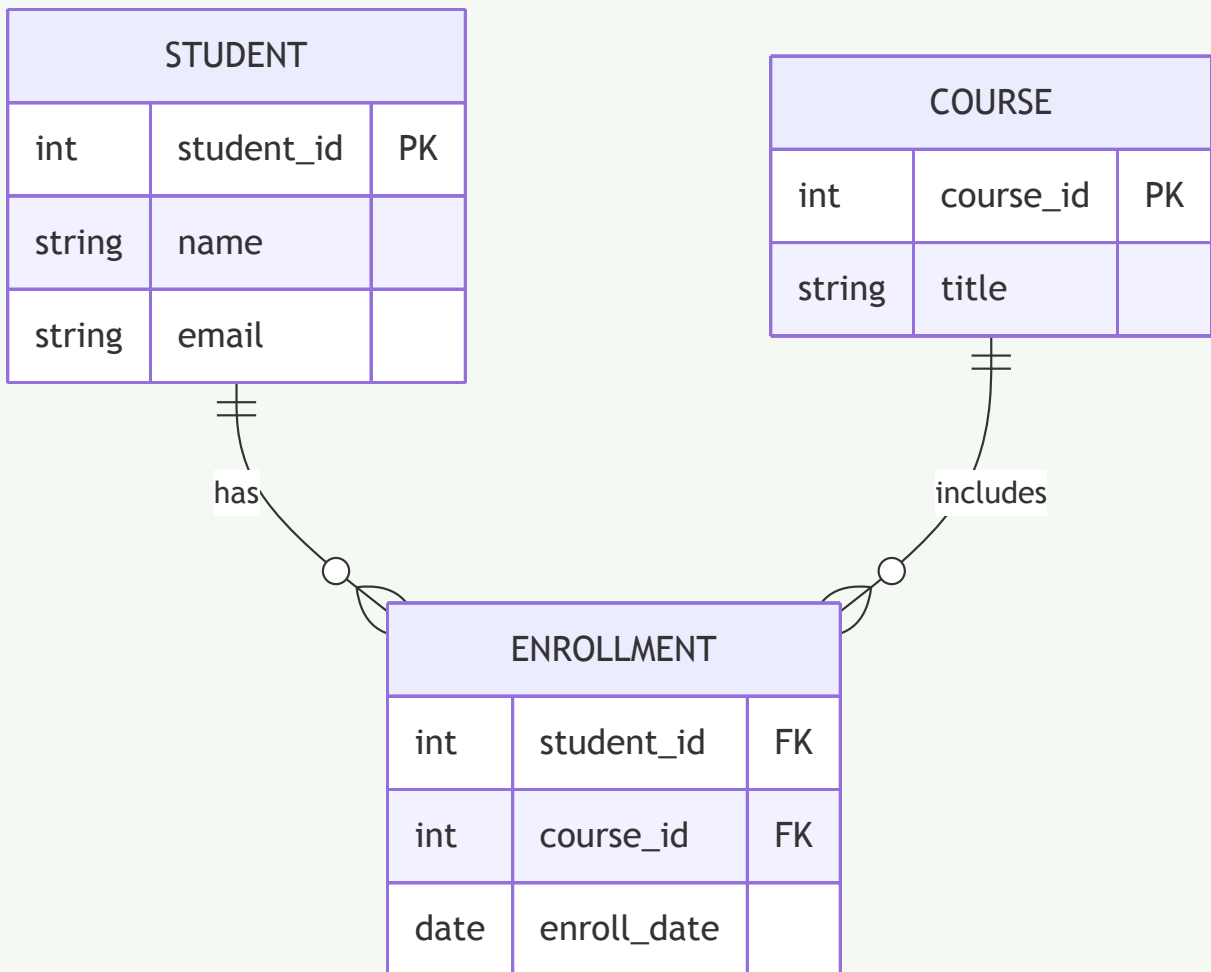
``SIM: સિંગલ ઇઝ મિનિમલ, મલ્ટી ઇઝ મેની``

પ્રશ્ન 2(બ) [4 ગુણ]

E-R ડાયાગ્રામ માટે કી કન્સ્ટ્રેન્ટ્સ સમજાવો

જવાબ

| કી કન્સ્ટ્રેન્ટ | વર્ણન |
|-----------------|--|
| પ્રાઇમરી કી | એન્ટિટી સેટમાં દરેક એન્ટિટીને અનન્ય રીતે ઓળખે છે |
| કેન્ડિડેટ કી | કોઈપણ એટ્રિબ્યુટ જે પ્રાઇમરી કી તરીકે કામ કરી શકે |
| ફોરેન કી | અન્ય એન્ટિટી સેટની પ્રાઇમરી કીનો સંદર્ભ આપે છે |
| સુપર કી | એટ્રિબ્યુટ્સનો કોઈપણ સેટ જે અનન્ય રીતે એન્ટિટીને ઓળખે છે |

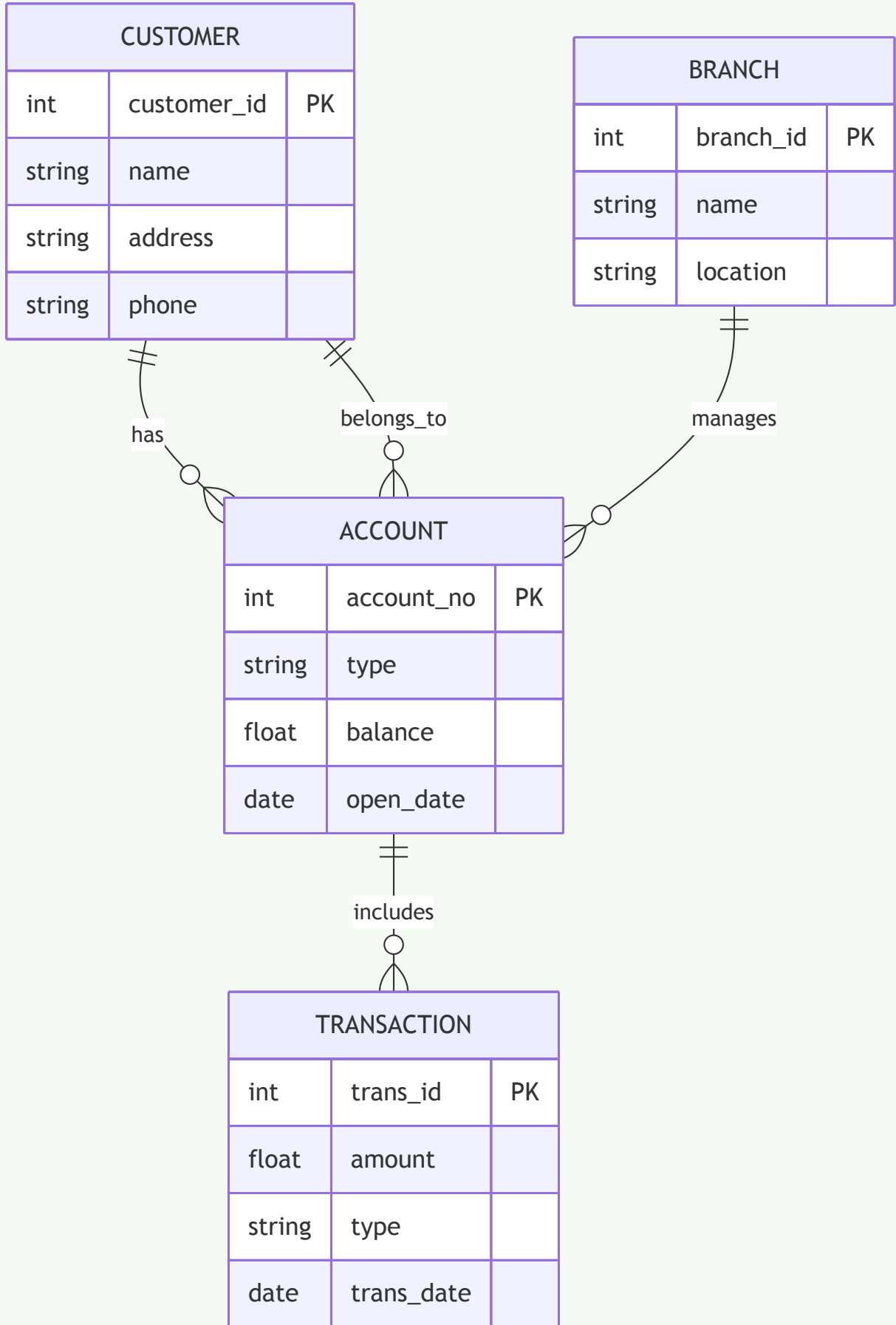


મેમરી ટ્રીક

``PCFS: પ્રાઇમરી કેન્ડિડેટ્સ ફાઇન્ડ સુપરકીઝ``

પ્રશ્ન 2(ક) [7 ગુણ]

બૅકિંગ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયાગ્રામ બનાવો



મુખ્ય એન્ટિટીઝ અને રિલેશનશિપ્સ:

- ગ્રાહક: ગ્રાહક માહિતી સંગ્રહિત કરે છે

- એકાઉન્ટ: વિવિધ એકાઉન્ટ પ્રકારો (સેવિંગ્સ, ચેકિંગ)
- ટ્રાન્ઝેક્શન: ડિપોઝિટ, વિડ્રોઅલ રેકૉર્ડ કરે છે
- બ્રાન્ચ: વિવિધ બેંક સ્થાનો
- રિલેશનશિપ્સ: ગ્રાહકો પાસે એકાઉન્ટ છે, એકાઉન્ટમાં ટ્રાન્ઝેક્શન છે, બ્રાન્ચ એકાઉન્ટ મેનેજ કરે છે

મેમરી ટ્રીક

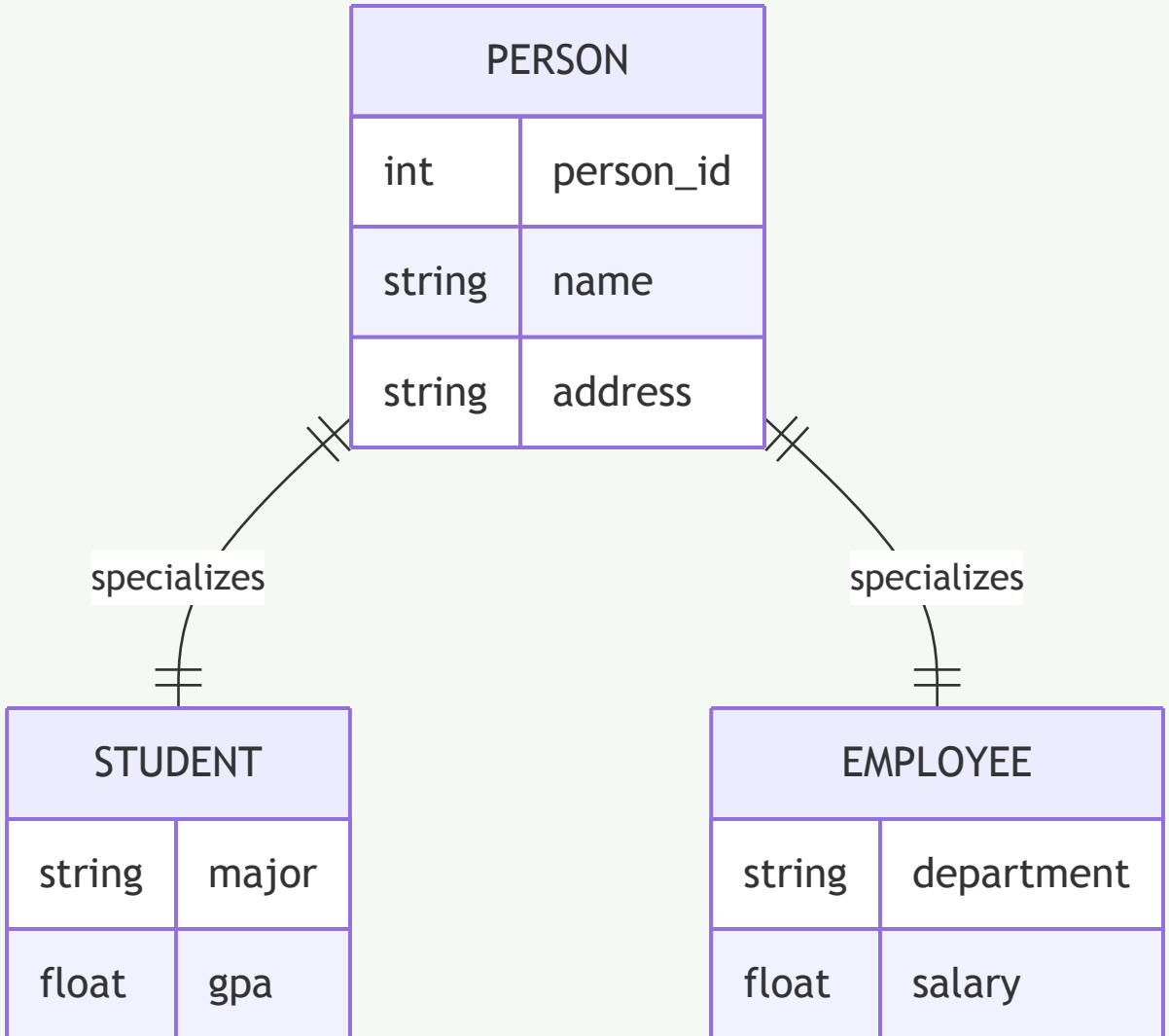
"CATB: કસ્ટમર્સ એક્સેસ ટ્રાન્ઝેક્શન્સ એટ બ્રાન્ચીસ"

પ્રશ્ન 2(અ) OR [3 ગુણ]

યોગ્ય ઉદાહરણ સાથે સ્પેશિયલાઈઝેશન અને જનરલાઈઝેશન વચ્ચેનો તફાવત સમજાવો

જવાબ

| વિચાર | દિશા | વર્ણન | ઉદાહરણ |
|----------------|----------|---|-------------|
| સ્પેશિયલાઈઝેશન | ટોપ-ડાઉન | સામાન્ય એન્ટિટીને વધુ ચોક્કસ સબ-એન્ટિટીઓમાં વિભાજિત કરવું | વ્યક્તિ → , |
| જનરલાઈઝેશન | બોટમ-અપ | સમાન એન્ટિટીઓને ઉચ્ચ-સ્તરીય એન્ટિટીમાં જોડવું | કાર, ટ્રક → |



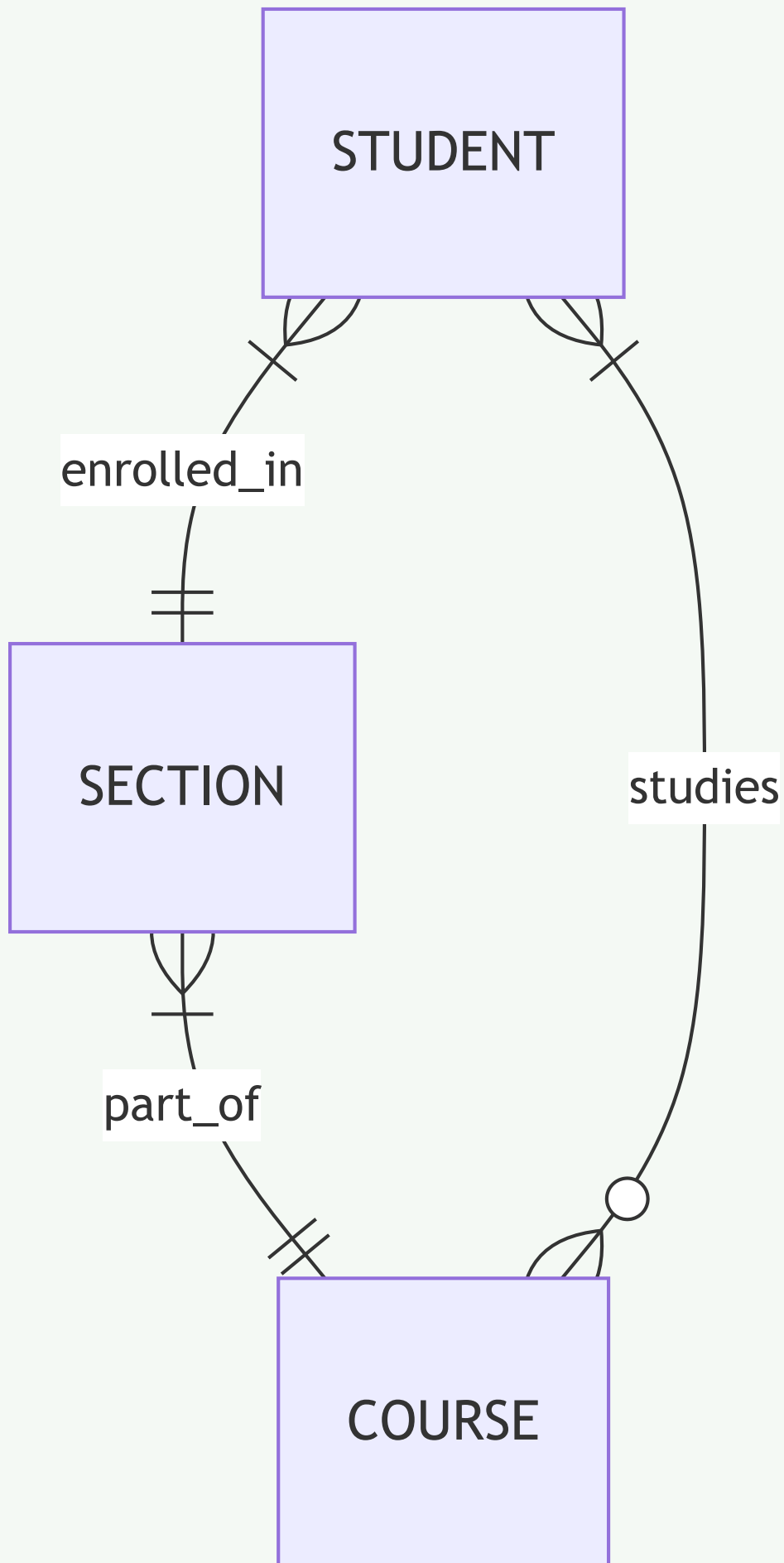
પ્રશ્ન 2(બ) OR [4 ગુણ]

ચાસ્પ ટ્રેપની વ્યાખ્યા લખો. તે ક્યારે ઉદ્ભવે છે તે સમજાવો. ચાસ્પ ટ્રેપ માટેનો ઉપાય સમજાવો

જવાબ

ચાસ્પ ટ્રેપ: ER ડાયાગ્રામમાં ઉદ્ભવતી સમસ્યા જ્યારે એન્ટિટીઓ વચ્ચે મલ્ટિપલ પાથ હોય છે, જેથી રિલેશનશિપના અર્થઘટનમાં અસ્પષ્ટતા આવે છે.

| પાસું | વર્ણન |
|--------|--|
| ઉદ્ભવ | જ્યારે એન્ટિટી પ્રકારો વચ્ચે બે અથવા વધુ અલગ પાથ હોય જે ચક્ર બનાવે છે |
| સમસ્યા | અયોગ્ય અથવા અસ્પષ્ટ ક્વેરી પરિણામો તરફ દોરી જાય છે |
| ઉકેલ | એક રિલેશનશિપને તોડવું અથવા ઇચ્છિત પાથને સ્પષ્ટ કરવા માટે કન્સ્ટ્રેન્ટ્સ ઉમેરવા |

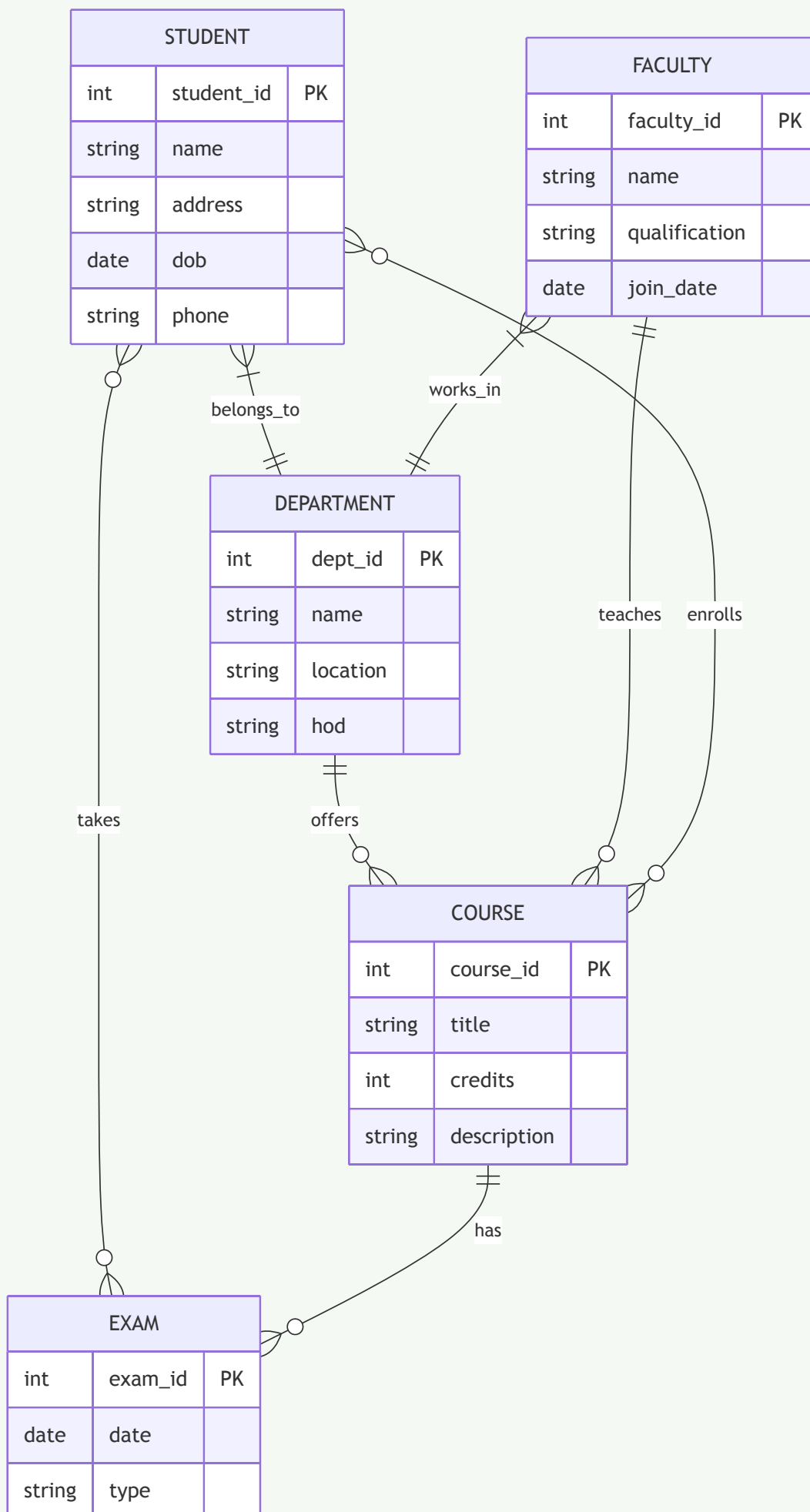


મેમરી ટ્રીક

“COP: સાયકલ્સ ઓફ પાથસ નીડ બ્રેકિંગ”

પ્રશ્ન 2(ક) OR [7 ગુણ]

કોલેજ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયાગ્રામ બનાવો



મુખ્ય એન્ટિટીઝ અને રિલેશનશિપ્સ:

- વિદ્યાર્થી: વિદ્યાર્થી વિગતો સંગ્રહિત કરે છે
- વિભાગ: શૈક્ષણિક વિભાગો
- ફેકલ્ટી: શિક્ષકો અને પ્રોફેસરો
- કોર્સ: ભણાવવામાં આવતા વિષયો
- પરીક્ષા: મૂલ્યાંકન કાર્યક્રમો
- રિલેશનશિપ્સ: વિદ્યાર્થીઓ કોર્સમાં એનરોલ થાય છે, ફેકલ્ટી કોર્સ શીખવે છે, વિભાગો કોર્સ ઓફર કરે છે

મેમરી ટ્રીક

``SDFCE: સ્ટુડન્ટ્સ ડિલાઇટ ફેકલ્ટી બાય કમ્પ્લીટિંગ એકઝામ્સ``

પ્રશ્ન 3(અ) [3 ગુણ]

GROUP BY ક્લોઝ ઉદાહરણ સાથે સમજાવો.

જવાબ

GROUP BY ક્લોઝ સમાન મૂલ્યો ધરાવતી રો સારાંશ રોમાં જૂથ કરે છે.

| ફીચર | વર્ણન |
|-----------|---|
| હેતુ | એકસરખા ડેટાને એગ્રીગેટ ફંક્શન માટે જૂથોમાં ગોઠવે છે |
| ઉપયોગ | એગ્રીગેટ ફંક્શન (COUNT, SUM, AVG, MAX, MIN) સાથે વપરાય છે |
| સિન્ટેક્સ | SELECT column1, COUNT(*) FROM table GROUP BY column1; |

```
1 SELECT department, AVG(salary)
2 FROM employees
3 GROUP BY department;
```

મેમરી ટ્રીક

``GAS: ગ્રુપ એન્ડ સમરાઈઝ``

પ્રશ્ન 3(બ) [4 ગુણ]

Data Definition Language (DDL) કમાન્ડની યાદી બનાવો. કોઈ પણ ૨ DDL કમાન્ડ ઉદાહરણ સાથે સમજાવો.

જવાબ

DDL કમાન્ડ્સ: CREATE, ALTER, DROP, TRUNCATE, RENAME

| કમાન્ડ | વર્ણન | ઉદાહરણ |
|----------|---|---|
| CREATE | ડેટાબેઝ ઓબ્જેક્ટ્સ જેમ કે ટેબલ, વ્યૂ, ઇન્ડેક્સ બનાવે છે | CREATE TABLE students (id INT PRIMARY KEY, name VARCHAR(50)); |
| ALTER | મૌજૂદ ડેટાબેઝ ઓબ્જેક્ટ્સ સુધારે છે | ALTER TABLE students ADD COLUMN email VARCHAR(100); |
| DROP | ડેટાબેઝ ઓબ્જેક્ટ્સ દૂર કરે છે | DROP TABLE students; |
| TRUNCATE | ટેબલમાંથી બધા રેકૉર્ડ્સ દૂર કરે છે | TRUNCATE TABLE students; |

મેમરી ટ્રીક

``CADTR: ક્રિએટ, ઓલ્ટર, ડ્રોપ, ટ્રન્કેટ, રીનેમ``

પ્રશ્ન 3(ક) [7 ગુણ]

enr_no, name, percent, branch ફિલ્ડ ધરાવતા Students ટેબલ પર નીચેની Query perform કરો.

જવાબ

```
1  -- . Students
2  SELECT * FROM Students;
3
4  -- . branch
5  SELECT DISTINCT branch FROM Students;
6
7  -- . name
8  SELECT * FROM Students ORDER BY name DESC;
9
10 -- . "address"
11 ALTER TABLE Students ADD address VARCHAR(100);
12
13 -- . "ICT"
14 SELECT * FROM Students WHERE branch = 'ICT';
15
16 -- . percent
17 DELETE FROM Students WHERE percent < 60;
18
19 -- . "S"
20 SELECT * FROM Students WHERE name LIKE 'S%';
```

| કવેરી | હેતુ |
|-------------|--|
| SELECT | ટેબલમાંથી ડેટા મેળવે છે |
| DISTINCT | ડુપ્લિકેટ મૂલ્યો દૂર કરે છે |
| ORDER BY | પરિણામોને ચોક્કસ ક્રમમાં ગોઠવે છે |
| ALTER TABLE | ટેબલ સ્ટ્રક્ચર સુધારે છે |
| WHERE | શરતો પર આધારિત રેકૉર્ડ્સ ફિલ્ટર કરે છે |
| DELETE | શરતો મેળવતા રેકૉર્ડ્સ દૂર કરે છે |
| LIKE | સ્ટ્રિંગ તુલનામાં પેટર્ન મેચિંગ |

મેમરી ટ્રીક

``SDOAWDL: સિલેક્ટ ડિસ્ટિંક્ટ ઓર્ડર ઓલ્ટર વ્હેર ડિલીટ લાઇક``

પ્રશ્ન 3(અ) OR [3 ગુણ]

સિન્ટેક્સ અને ઉદાહરણ સાથે GRANT કમાન્ડ સમજાવો.

જવાબ

GRANT કમાન્ડ વપરાશકર્તાઓને ડેટાબેઝ ઓબ્જેક્ટ્સ પર ચોક્કસ અધિકારો આપે છે.

| ઘટક | વર્ણન |
|-------------|---|
| સિન્ટેક્સ | GRANT privilege(s) ON object TO user [WITH GRANT OPTION]; |
| પ્રિવિલેજીસ | SELECT, INSERT, UPDATE, DELETE, ALL PRIVILEGES |
| ઓબ્જેક્ટ્સ | ટેબલ્સ, વ્યૂ, સિક્વેન્સિસ, વગેરે |

```
1 GRANT SELECT, UPDATE ON employees TO user1;
2 GRANT ALL PRIVILEGES ON database_name.* TO user2 WITH GRANT OPTION;
```

મેમરી ટ્રીક

“GPO: ગ્રાન્ટ પ્રિવિલેજીસ ટુ અધર્સ”

પ્રશ્ન 3(બ) OR [4 ગુણ]

Truncate અને Drop કમાન્ડનો તફાવત લખો.

જવાબ

| ફીચર | TRUNCATE | DROP |
|-------------------|---|--|
| હેતુ સ્ટ્રક્ચર | ટેબલથી બધી પંક્તિઓ દૂર કરે છે ટેબલ સ્ટ્રક્ચર જાળવી રાખે છે | સંપૂર્ણ ટેબલ સ્ટ્રક્ચર દૂર કરે છે ટેબલની વ્યાખ્યા સંપૂર્ણપણે દૂર કરે છે |
| રિકવરી | સરળતાથી રોલબેક નથી કરી શકાતું | કમિટ થાય ત્યાં સુધી પુનઃપ્રાપ્ત કરી શકાય છે |
| સ્પીડ ટ્રિગર્સ | DELETE કરતાં ઝડપી ટ્રિગર્સ સક્રિય કરતું નથી | ઝડપી ઓપરેશન ટ્રિગર્સ સક્રિય કરતું નથી |

```

1 -- Truncate
2 TRUNCATE TABLE students;
3
4 -- Drop
5 DROP TABLE students;
```

મેમરી ટ્રીક

“TRC-DST: ટ્રન્કેટ રિમૂવ્સ કન્ટેન્ટ્સ, ડ્રોપ ડિસ્ટ્રોય્સ સ્ટ્રક્ચર ટોટલી”

પ્રશ્ન 3(ક) OR [7 ગુણ]

નીચેની Query ના આઉટપુટ લખો.

જવાબ

| ક્વેરી | આઉટપુટ | સમજૂતી |
|---------------------------------------|---------------|---|
| ABS(-23), ABS(49) | 23, 49 | નિરપેક્ષ મૂલ્ય પાછું આપે છે |
| SQRT(25), SQRT(81) | 5, 9 | વર્ગમૂળ પાછું આપે છે |
| POWER(3,2), POWER(-2,3) | 9, -8 | x^y (પ્રથમ મૂલ્યને બીજા મૂલ્યની પાવર સુધી ઉંચકે છે) |
| MOD(15,4), MOD(21,3) | 3, 0 | વિભાજન પછી શેષ પાછો આપે છે |
| ROUND(123.446,1), ROUND(123.456,2) | 123.4, 123.46 | ચોક્કસ દશાંશ જગ્યાઓ પર રાઉન્ડ કરે છે |
| CEIL(234.45), CEIL(-234.45) | 235, -234 | નજીકના પૂર્ણાંક સુધી ઉપર રાઉન્ડ કરે છે |
| FLOOR(-12.7), FLOOR(12.7) | -13, 12 | નજીકના પૂર્ણાંક સુધી નીચે રાઉન્ડ કરે છે |

```

1 SELECT ABS(-23), ABS(49);           -- 23, 49
2 SELECT SQRT(25), SQRT(81);          -- 5, 9
3 SELECT POWER(3,2), POWER(-2,3);      -- 9, -8
4 SELECT MOD(15,4), MOD(21,3);         -- 3, 0
5 SELECT ROUND(123.446,1), ROUND(123.456,2); -- 123.4, 123.46
6 SELECT CEIL(234.45), CEIL(-234.45);  -- 235, -234
7 SELECT FLOOR(-12.7), FLOOR(12.7);    -- -13, 12

```

મેમરી ટ્રીક

“ASPMRCF: એબ્સોલ્યુટ સ્ક્વેર પાવર મોડ્યુલો રાઉન્ડ સીલિંગ ફ્લોર”

પ્રશ્ન 4(અ) [3 ગુણ]

SQLમાં ડેટા ટાઈપની યાદી બનાવો. કોઈ પણ ૨ ડેટા ટાઈપ ઉદાહરણ સાથે સમજાવો.

જવાબ

SQL ડેટા ટાઈપ: INTEGER, FLOAT, VARCHAR, CHAR, DATE, DATETIME, BOOLEAN, BLOB

| ડેટા ટાઈપ | વર્ણન | ઉદાહરણ |
|-----------|---|---------------------------------|
| INTEGER | દશાંશ પોઇન્ટ વિના પૂર્ણ સંખ્યાઓ | id INTEGER = 101 |
| VARCHAR | પરિવર્તનશીલ-લંબાઈ સ્ટ્રિંગ | name VARCHAR(50) = 'John' |
| DATE | તારીખ મૂલ્યો સંગ્રહિત કરે છે (YYYY-MM-DD) | birth_date DATE = '2000-05-15' |
| FLOAT | ફ્લોટિંગ પોઇન્ટ સાથે દશાંશ સંખ્યાઓ | salary FLOAT = 45000.50 |

```

1 CREATE TABLE employees (
2     id INTEGER,
3     name VARCHAR(50),
4     salary FLOAT
5 );

```

મેમરી ટ્રીક

“IVDB: ઇન્ટિજર અને વારચાર આર ડેટાબેઝ બેસિક્સ”

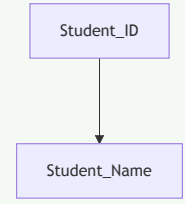
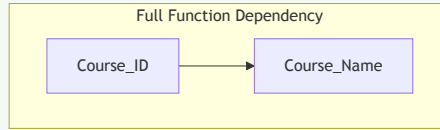
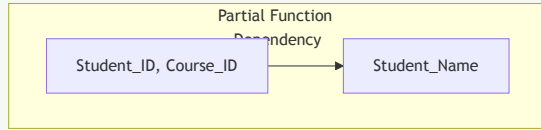
પ્રશ્ન 4(બ) [4 ગુણ]

Full function dependency ઉદાહરણ સાથે સમજાવો.

જવાબ

Full Function Dependency: જ્યારે Y, X પર ફંક્શનલી ડિપેન્ડન્ટ હોય, પરંતુ X ના કોઈ સબસેટ પર નહીં.

| વિચાર | વર્ણન | ઉદાહરણ |
|------------|---|--|
| વ્યાખ્યા | એટ્રિબ્યુટ B, A પર પૂર્ણપણે ફંક્શનલી ડિપેન્ડન્ટ છે જો B સંપૂર્ણ A પર આધાર રાખે છે | Student_ID → Name() |
| નોન-ઉદાહરણ | જ્યારે એટ્રિબ્યુટ કોમ્પોઝિટ કીના માત્ર ભાગ પર આધાર રાખે છે | {Student_ID, Course_ID} → Student_Name() |



મેમરી ટ્રીક

“FFD: ફુલ, નોટ ફંક્શન ઓફ ડિપેન્ડન્સી”

પ્રશ્ન 4(ક) [7 ગુણ]

નોર્મલાઇઝેશનની વ્યાખ્યા આપો. 2NF (સેકન્ડ નોર્મલ ફોર્મ) ઉદાહરણ અને ઉકેલ સાથે સમજાવો.

જવાબ

નોર્મલાઇઝેશન: ડેટાબેઝની રચના કરવાની પ્રક્રિયા જેથી મોટા ટેબલને નાના ટેબલોમાં વિભાજિત કરીને અને તેમની વચ્ચે સંબંધો વ્યાખ્યાયિત કરીને, રિડન્ડન્સી અને ડિપેન્ડન્સી ઘટાડવામાં આવે.

2NF (સેકન્ડ નોર્મલ ફોર્મ):

- ટેબલ 2NF માં છે જો તે 1NF માં હોય અને કોઈ નોન-પ્રાઇમ એટ્રિબ્યુટ કેન્ડિડેટ કીના કોઈ પણ યોગ્ય સબસેટ પર આધાર રાખતું ન હોય.

2NF પહેલાં

સમસ્યા

Order(Order_ID, Product_ID, Product_Name, Quantity, Price)

Product_Name માત્ર Product_ID પર આધાર રાખે છે, સંપૂર્ણ કી પર નહીં

2NF પછી

Order(Order_ID, Product_ID, Quantity)
Product(Product_ID, Product_Name, Price)

ઉકેલ

માત્ર પૂર્ણ કી ડિપેન્ડન્સી
પ્રોડક્ટ વિગતો માત્ર Product_ID પર આધાર રાખે છે

| ORDER | |
|-------|------------|
| int | order_id |
| int | product_id |
| int | quantity |



contains



| PRODUCT | |
|---------|--------------|
| int | product_id |
| string | product_name |
| float | price |

મેમરી ટ્રીક

``2NF-PPD: પાર્શિયલ ડિપેન્ડન્સી પ્રોબ્લેમ્સ ડિવાઇડેડ``

પ્રશ્ન 4(અ) OR [3 ગુણ]

ક્રમાંક સમજાવવો. ૧) To_Number (), ૨) To_Char()

જવાબ

| ફંક્શન | હેતુ | સિન્ટેક્સ | ઉદાહરણ |
|-------------|---|------------------------------|---------------------------------|
| TO_NUMBER() | સ્ટ્રિંગને નંબરમાં રૂપાંતરિત કરે છે | TO_NUMBER(string, [format]) | TO_NUMBER('123.45') = 123.45 |
| TO_CHAR() | નંબર/તારીખને સ્ટ્રિંગમાં રૂપાંતરિત કરે છે | TO_CHAR(value, [format]) | TO_CHAR(1234, '9999') = '1234' |

```

1  --
2  SELECT TO_NUMBER('123.45') FROM dual;  -- 123.45
3
4  --
5  SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') FROM dual;  -- 20-JAN-2024
6
7  --
8  SELECT TO_CHAR(1234.56, '$9,999.99') FROM dual;  -- $1,234.56

```

મેમરી ટ્રીક

``NC: નંબર્સ એન્ડ કેરેક્ટર્સ કન્વર્ઝન``

પ્રશ્ન 4(બ) OR [4 ગુણ]

1NF (ફર્સ્ટ નોર્મલ ફોર્મ) ઉદાહરણ અને ઉકેલ સાથે સમજાવો.

જવાબ

1NF (ફર્સ્ટ નોર્મલ ફોર્મ): એક રિલેશન 1NF માં છે જો તેમાં કોઈ રિપીટિંગ ગ્રુપ્સ અથવા એરે ન હોય.

| 1NF પહેલાં | સમસ્યા |
|---|-----------------------------------|
| Student(ID, Name, Courses) | Courses કોલમમાં મલ્ટિપલ વેલ્યુ છે |
| ઉદાહરણ: (101, John, ``Math,Science,History``) | મલ્ટી-વેલ્યુડ એટ્રિબ્યુટ |

1NF પછી

ઉકેલ

Student(ID, Name, Course)

દરેક રોમાં એક કોર્સ

ઉદાહરણો: (101, John, Math), (101, John, Science), (101, John, History)

એટોમિક વેલ્યુઝ

| STUDENT_BEFORE | |
|----------------|---------|
| int | id |
| string | name |
| string | courses |

| STUDENT_AFTER | |
|---------------|--------|
| int | id |
| string | name |
| string | course |

મેમરી ટ્રીક

"1NF-ARM: એટોમિક વેલ્યુઝ રિમૂવ મલ્ટિવેલ્યુઝ"

પ્રશ્ન 4(ક) OR [7 ગુણ]

SQL માં Function dependency સમજાવો. Partial function dependency ઉદાહરણ સાથે સમજાવો.

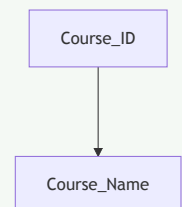
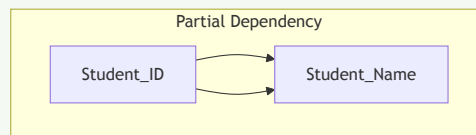
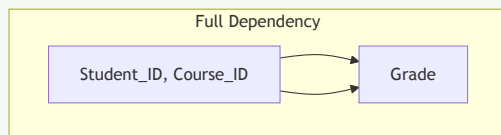
જવાબ

ફંક્શનલ ડિપેન્ડન્સી: એક સંબંધ જ્યાં એક એટ્રિબ્યુટ બીજા એટ્રિબ્યુટનું મૂલ્ય નક્કી કરે છે.

નોટેશન: $X \rightarrow Y(X, Y)$

પાર્શિયલ ફંક્શનલ ડિપેન્ડન્સી: જ્યારે એક એટ્રિબ્યુટ કમ્પોઝિટ પ્રાઇમરી કીના માત્ર એક ભાગ પર આધાર રાખે છે.

| વિચાર | ઉદાહરણ | સમજૂતી |
|----------------------|--|--|
| કમ્પોઝિટ કી | {Student_ID, Course_ID} | સાથે મળીને પ્રાઇમરી કી બનાવે છે |
| પાર્શિયલ ડિપેન્ડન્સી | {Student_ID, Course_ID} \rightarrow Student_Name | Student_Name માત્ર Student_ID પર આધાર રાખે છે |
| સમસ્યા | અપડેટ એનોમલીઝ, ડેટા રિડન્ડન્સી | એક જ વિદ્યાર્થીનું નામ ઘણા બધા કોર્સ માટે પુનરાવર્તિત થાય છે |



ઉકેલ: અલગ ટેબલોમાં વિભાજિત કરો જ્યાં દરેક નોન-કી એટ્રિબ્યુટ કી પર સંપૂર્ણપણે આધારિત હોય.

મેમરી ટ્રીક

"PD-CPK: પાર્શિયલ ડિપેન્ડન્સી - કમ્પોનન્ટ ઓફ પ્રાઇમરી કી"

પ્રશ્ન 5(અ) [3 ગુણ]

Transaction નાં ગુણધર્મો ઉદાહરણ સાથે સમજાવો.

જવાબ

Transaction ગુણધર્મો (ACID):

| ગુણધર્મ | વર્ણન | ઉદાહરણ |
|--------------|---|---|
| એટોમિસિટી | બધા ઓપરેશનો સફળતાપૂર્વક પૂર્ણ થાય છે અથવા એક પણ થતું નથી | બેંક ટ્રાન્સફર: ડેબિટ અને ક્રેડિટ બંને થાય અથવા બંને ન થાય |
| કન્સિસ્ટન્સી | ડેટાબેઝ પહેલા અને પછી માન્ય સ્થિતિમાં રહે છે | એકાઉન્ટ બેલેન્સ કન્સ્ટ્રેન્ટ્સ માન્ય રહે છે |
| આઇસોલેશન | ટ્રાન્ઝેક્શન એવી રીતે એક્ઝિક્યુટ થાય છે જાણે તે એકમાત્ર હોય | બે યુઝર એક જ રેકૉર્ડ અપડેટ કરી રહ્યા હોય ત્યારે દખલ કરતા નથી |
| ડ્યુરેબિલિટી | કમિટ કરેલા ફેરફારો સિસ્ટમ નિષ્ફળતા પછી પણ ટકી રહે છે | એકવાર પુષ્ટિ થઈ જાય, પછી વીજળી જતી રહે તો પણ ડિપોઝિટ યથાવત રહે છે |

```

graph LR
    A[START TRANSACTION] --> B[Debit Account A]
    B --> C[Credit Account B]
    C --> D{Successful?}
    D -- Yes --> E[COMMIT]
    D -- No --> F[ROLLBACK]
    
```

મેમરી ટ્રીક

“ACID: એટોમિસિટી, કન્સિસ્ટન્સી, આઇસોલેશન, ડ્યુરેબિલિટી”

પ્રશ્ન 5(બ) [4 ગુણ]

ઉપર Q.5 (b) માં આપેલ “Students” અને “CR” ટેબલનો ઉપયોગ કરીને સેટ ઓપરેટર દ્વારા નીચેની Query લખો.

જવાબ

```

1  -- . Students    CR
2  SELECT Stnd_Name FROM Student
3  UNION
4  SELECT CR_Name FROM CR;
5
6  -- . Students    CR
7  SELECT Stnd_Name FROM Student
8  INTERSECT
9  SELECT CR_Name FROM CR;
10
11 -- . Students    CR
12 SELECT Stnd_Name FROM Student
13 MINUS
14 SELECT CR_Name FROM CR;
15
16 -- . CR          Student
17 SELECT CR_Name FROM CR
18 MINUS
19 SELECT Stnd_Name FROM Student;
    
```

| સેટ ઓપરેટર | હેતુ | ઉદાહરણ માટે પરિણામ |
|------------|-----------------------------|---|
| UNION | બધી અલગ રો જોડે છે | Manoj, Rahil, Jiya, Rina, Jitesh, Priya |
| INTERSECT | માત્ર સામાન્ય રો પરત કરે છે | Manoj, Rina |

MINUS

પ્રથમ સેટમાં હોય પણ બીજા સેટમાં ન હોય તે રો

Rahil, Jiya

MINUS (ભિન્ન)

બીજા સેટમાં હોય પણ પહેલા સેટમાં ન હોય તે રો

Jitesh, Priya

મેમરી ટ્રીક

“UIMD: યુનિયન ઇન્કલુડ્સ, માઈનસ ડિવાઈડ્સ”

પ્રશ્ન 5(ક) [7 ગુણ]

Conflict serializability વિસ્તારથી સમજાવો.

જવાબ

Conflict Serializability: એક શેડ્યુલ કન્ફ્લિક્ટ સીરિયલાઇઝેબલ છે જો તેને નોન-કન્ફ્લિક્ટિંગ ઓપરેશન્સને સ્વેપ કરીને સીરિયલ શેડ્યુલમાં રૂપાંતરિત કરી શકાય.

મુખ્ય વિચારો

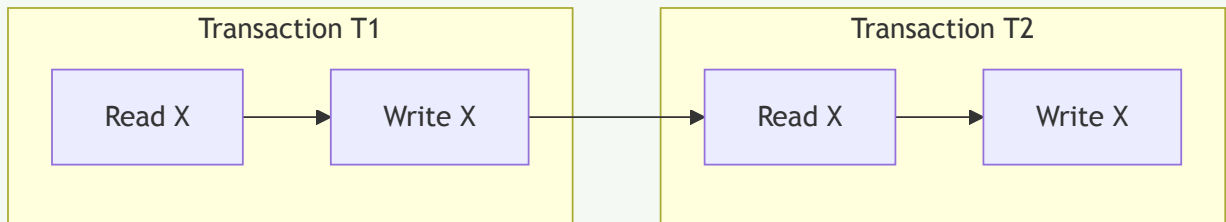
વર્ણન

કન્ફ્લિક્ટ ઓપરેશન્સ

બે ઓપરેશન કન્ફ્લિક્ટ કરે છે જો તેઓ એક જ ડેટા આઇટમને એક્સેસ કરે છે અને ઓછામાં ઓછું એક રાઇટ હોય
ટ્રાન્ઝેક્શન વચ્ચેના કન્ફ્લિક્ટને દર્શાવતો ડાયરેક્ટેડ ગ્રાફ
જો પ્રીસિડન્સ ગ્રાફમાં કોઈ સાયકલ ન હોય, તો શેડ્યુલ કન્ફ્લિક્ટ સીરિયલાઇઝેબલ છે

પ્રીસિડન્સ ગ્રાફ
સીરિયલાઇઝેબલ

Conflicts



ઉદાહરણ:

- T1: R(X), W(X)
- T2: R(X), W(X)

સીરિયલાઇઝેબલ શેડ્યુલ્સ:

- T1 બાદ T2: R1(X), W1(X), R2(X), W2(X)
- T2 બાદ T1: R2(X), W2(X), R1(X), W1(X)

નોન-સીરિયલાઇઝેબલ: R1(X), R2(X), W1(X), W2(X) - પ્રીસિડન્સ ગ્રાફમાં સાયકલ બનાવે છે

મેમરી ટ્રીક

“COPS: કન્ફ્લિક્ટ ઓપરેશન્સ પ્રોડ્યુસ સીરિયલાઇઝેબિલિટી”

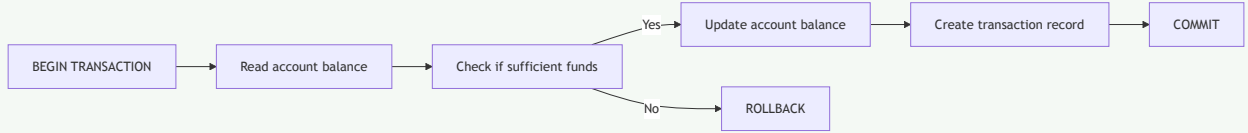
પ્રશ્ન 5(અ) OR [3 ગુણ]

Transaction નો concept ઉદાહરણ સાથે સમજાવો.

જવાબ

ટ્રાન્ઝેક્શન: કામની એક તાર્કિક એકમ જે સંપૂર્ણપણે કરવું અથવા સંપૂર્ણપણે અનડૂ કરવું આવશ્યક છે.

| ટ્રાન્ઝેક્શન તબક્કાઓ | વર્ણન | ઉદાહરણ |
|-------------------------------------|--|---|
| BEGIN ઓપરેશન-સ એક્ઝિક્યુટ | ટ્રાન્ઝેક્શનની શરૂઆત ચિહ્નિત કરે છે ડેટાબેઝ ઓપરેશન-સ (રીડ/રાઇટ) | START TRANSACTION UPDATE account SET balance = balance - 1000 WHERE id = 123 |
| COMMIT/ROLLBACK | સફળતા/નિષ્ફળતા સાથે ટ્રાન્ઝેક્શન સમાપ્ત કરે છે | COMMIT અથવા ROLLBACK |



ઉદાહરણ:

```

1 BEGIN TRANSACTION;
2 UPDATE accounts SET balance = balance - 1000 WHERE acc_no = 123;
3 UPDATE accounts SET balance = balance + 1000 WHERE acc_no = 456;
4 COMMIT;
  
```

મેમરી ટ્રીક

“BEC: બિગિન, એક્ઝિક્યુટ, કમિટ”

પ્રશ્ન 5(બ) OR [4 ગુણ]

Equi-join સિન્ટેક્સ અને ઉદાહરણ સાથે સમજાવો.

જવાબ

Equi-join: એક જોઈન ઓપરેશન જે સમાનતા કમ્પેરિઝન ઓપરેટરનો ઉપયોગ કરે છે.

| ફીચર | વર્ણન |
|----------------------|--|
| સિન્ટેક્સ | SELECT columns FROM table1, table2 WHERE table1.column = table2.column; |
| હેતુ વૈકલ્પિક | મેચિંગ કોલમ વેલ્યુના આધારે બે ટેબલમાંથી રો જોડે છે SELECT columns FROM table1 INNER JOIN table2 ON table1.column = table2.column; |

```

1 --
2 SELECT s.name, d.dept_name
3 FROM students s, departments d
4 WHERE s.dept_id = d.dept_id;
5
6 -- INNER JOIN
7 SELECT s.name, d.dept_name
8 FROM students s INNER JOIN departments d
9 ON s.dept_id = d.dept_id;
  
```

મેમરી ટ્રીક

“EQ-ME: ઇક્વાલિટી મેચિસ એન્ટ્રીસ”

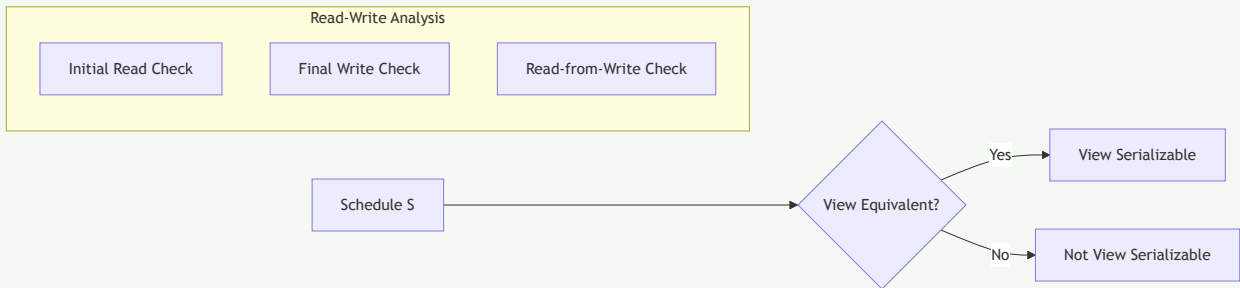
પ્રશ્ન 5(ક) OR [7 ગુણ]

View serializability વિસ્તારથી સમજાવો.

જવાબ

View Serializability: એક શેડ્યુલ વ્યૂ સીરિયલાઇઝેબલ છે જો તે કોઈ સીરિયલ શેડ્યુલ સાથે વ્યૂ ઇક્વિવેલન્ટ હોય.

| શરત | વર્ણન |
|-------------------------|---|
| ઇનિશિયલ રીડ | જો T1 શેડ્યુલ S માં ડેટા આઇટમ X ની પ્રારંભિક વેલ્યુ વાંચે છે, તો તેણે S' શેડ્યુલમાં પણ પ્રારંભિક વેલ્યુ વાંચવી જોઈએ |
| ફાઇનલ રાઇટ | જો T1, S માં ડેટા આઇટમ X નું અંતિમ લખાણ કરે છે, તો તેણે S' માં પણ અંતિમ લખાણ કરવું જોઈએ |
| ડિપેન્ડન્સી પ્રિઝર્વેશન | જો T1, S માં T2 દ્વારા લખાયેલ X ની વેલ્યુ વાંચે છે, તો તેણે S' માં પણ T2 પાસેથી વાંચવું જોઈએ |



તુલના:

- કન્ફ્લિક્ટ સીરિયલાઇઝેબિલિટી: વધુ પ્રતિબંધિત, પરીક્ષણ કરવું સરળ (પ્રીસિડન્સ ગ્રાફ)
- વ્યૂ સીરિયલાઇઝેબિલિટી: વધુ સામાન્ય, પરીક્ષણ કરવું વધુ મુશ્કેલ (NP-કમ્પ્લીટ)

વ્યૂ સીરિયલાઇઝેબલ પરંતુ કન્ફ્લિક્ટ સીરિયલાઇઝેબલ નહીં તેનું ઉદાહરણ:

- T1: W(X)
- T2: W(X)
- T3: R(X)
- શેડ્યુલ: W1(X), W2(X), R3(X) - સીરિયલ શેડ્યુલ T2,T1,T3 સાથે વ્યૂ ઇક્વિવેલન્ટ

મેમરી ટ્રીક

“VIR-FF: વ્યૂ પ્રિઝર્વેશન ઇનિશિયલ રીડ્સ એન્ડ ફાઇનલ રાઇટ્સ”