

ડેટાબેઝ મેનેજમેન્ટ (4331603) - ઉનાળો 2025 ઉકેલ

Milav Dabgar

May 15, 2025

પ્રશ્ન 1(a) [3 ગુણ]

નીચેના શબ્દોની વ્યાખ્યા આપો. 1) મેટાડેટા 2) સ્કીમા 3) ડેટા ડિક્ષનરી.

જવાબ

ટેબલ:

કોષ્ટક 1. ડેટાબેઝ શબ્દો

શબ્દ	વ્યાખ્યા
મેટાડેટા	ડેટા વિશેનો ડેટા જે ડેટાબેઝની રૂચના અને વિશેષતાઓ વર્ણવે છે
સ્કીમા	ડેટાબેઝના સંગઠન અને સંબંધોને દર્શાવતી તાકિક રૂચના
ડેટા ડિક્ષનરી	ડેટાબેઝના તત્ત્વો વિશેની માહિતી સંગ્રહિત કરતો કેન્દ્રીય ભંડાર

- મેટાડેટા: ડેટાની લાક્ષણિકતાઓ અને ગુણાધમો વર્ણવતી માહિતી
- સ્કીમા: ડેટાબેઝની રૂચના અને મર્યાદાઓ વ્યાખ્યાયિત કરતો બ્લૂપ્રિન્ટ
- ડેટા ડિક્ષનરી: બધા ડેટાબેઝ ઓફ્જેક્ટ્સ અને તેમના ગુણાધમોની કેટલોગ

મેમરી ટ્રીક

"MSD - My System Dictionary"

પ્રશ્ન 1(b) [4 ગુણ]

ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમના ફાયદા લખો.

જવાબ

ટેબલ:

કોષ્ટક 2. DBMS ના ફાયદા

ફાયદો	વર્ણન
ડેટા સ્વતંત્રતા	એપ્લિકેશન્સ ડેટા સ્ટોરેજથી સ્વતંત્ર
ડેટા આધ્યાત્મિકતા	ડેટાની ચોકસાઈ અને સુસંગતતા જાળવે છે
સુરક્ષા નિયંત્રણ	વપરાશકર્તા પ્રમાણીકરણ અને અધિકરણ
સમવર્તી પહોંચ	અનેક વપરાશકર્તાઓ એકસાથે પહોંચ કરી શકે છે

- ઘટેલી રીડન-ન્સી: ડ્યુપ્લિકેટ ડેટા સ્ટોરેજ દૂર કરે છે
- કેન્દ્રીકૃત નિયંત્રણ: ડેટા મેનેજમેન્ટનું એક જ બિંદુ
- ડેટા વહેંચણી: અનેક એપ્લિકેશન્સ સમાન ડેટાનો ઉપયોગ કરી શકે છે
- બેંકઅપ પુનઃપ્રાપ્તિ: આપોઆપ ડેટા સુરક્ષા પદ્ધતિઓ

મેમરી ટ્રીક

"DISC-RCDB - Database Is Super Cool"

પ્રશ્ન 1(c) [7 ગુણ]

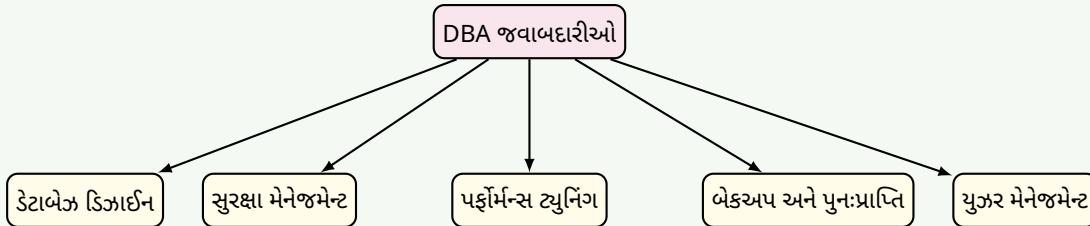
DBA ની જવાબદારીઓ સમજાવો.

જવાબ

ટેબલ:

કોષ્ટક 3. DBA ની જવાબદારીઓ

જવાબદારી	કાર્યો
ડેટાબેઝ ડિઝાઇન	તાર્કિક અને ભૌતિક રચનાઓ બનાવવી
સુરક્ષા મેનેજમેન્ટ	વપરાશકર્તા પહોંચ અને પરવાનગીઓનું નિયંત્રણ
પફોર્મન્સ ટ્યુનિંગ	કવરેજ અને ડેટાબેઝ ઓપરેશન્સને ઓપ્ટિમાઇઝ કરવા
બેકઅપ પુનઃપ્રાપ્તિ	ડેટા સુરક્ષા અને પુનઃસ્થાપન સુનિશ્ચિત કરવું
યુગર મેનેજમેન્ટ	એકાઉન્ટ બનાવવા અને વિશેષાધિકારો અસાઇન કરવા



આકૃતિ 1. DBA ની મુખ્ય જવાબદારીઓ

- ડેટાબેઝ ઇન્સ્ટલેશન: DBMS સોફ્ટવેર સેટઅપ અને કોન્ફિગર કરવું
- ડેટા માઇગ્રેશન: સિસ્ટમની વર્ત્તિ અને ડેટાની સુરક્ષા રીતે ટ્રાન્સફર કરવો
- ડૉક્યુમેન્ટેશન: ડેટાબેઝ સ્કીમા અને પ્રક્રિયાઓ જાળવવી
- મોનિટરિંગ: સિસ્ટમ પફોર્મન્સ અને રિસોર્સ વપરાશ ટ્રૈક કરવું
- ટ્રબ્લશ્યુટિંગ: ડેટાબેઝ સમસ્યાઓ અને ભૂલો ઉકેલવી

મેમરી ટ્રીક

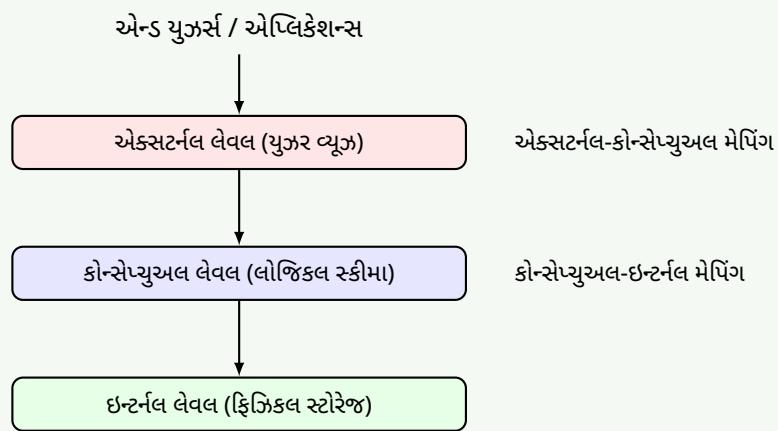
"DSPBU-DMT - DBA Solves Problems By Understanding Database Management Tasks"

પ્રશ્ન 1(c OR) [7 ગુણ]

ડેટા એબ્સ્ટ્રેક્શન શું છે? ત્રણ સ્તરની ANSI SPARC આર્કિટેક્ચરને વિગતવાર સમજાવો.

જવાબ

ડેટા એબ્સ્ટ્રેક્શન: વપરાશકર્તાની જાતિની ડેટાબેઝ અમલીકરણ વિગતો છુપાવીને સરળ ઇન્ટરફેસ પ્રદાન કરવું.



ટેબલ:

કોષ્ટક 4. આર્કિટેક્ચર સ્તરો

સ્તર	વર્ણન	વપરાશકર્તાઓ
એક્સ્ટર્નલ લેવલ	વ્યક્તિગત વપરાશકર્તા દૃશ્યો અને એપ્લિકેશન્સ	એન્ડ યુજર્સ
કોન્સેપ્ચુઅલ લેવલ	સંપૂર્ણ તાર્કિક ડેટાબેઝ રચના	ડેટાબેઝ ડિઝાઇનર્સ
ઇન્ટર્નલ લેવલ	ભૌતિક ફાઈલ સંગઠન અને ઇન્ડૉક્સિંગ	સિસ્ટમ પ્રોગ્રામર્સ

- એક્સ્ટર્નલ લેવલ: જટિલતા છુપાવતા અનેક વપરાશકર્તા દૃશ્યો
- કોન્સેપ્ચુઅલ લેવલ: સ્ટોરેજ વિગતો વિના સંપૂર્ણ ડેટાબેઝ સ્કીમા
- ઇન્ટર્નલ લેવલ: ભૌતિક ફાઈલ સંગઠન અને ઇન્ડૉક્સિંગ
- ડેટા સ્વતંત્રતા: એક સ્તરમાં ફેરફારો અન્યને અસર કરતા નથી

મેરી ટ્રીક

"ECI - Every Computer Implements"

પ્રશ્ન 2(a) [3 ગુણા]

સ્કીમા અને ઇન્સ્ટન્સનો તફાવત સમજાવો

જવાબ

ટેબલ:

કોષ્ટક 5. સ્કીમા VS ઇન્સ્ટન્સ

પાસું	સ્કીમા	ઇન્સ્ટન્સ
વ્યાખ્યા	ડેટાબેઝ રચનાનો બ્લુપ્રિન્ટ	ચોક્કસ સમયે વાસ્તવિક ડેટા
પ્રકૃતિ	સ્થિર તાર્કિક ડિઝાઇન	ડાયનામિક ડેટા સામગ્રી
ફેરફારો	ભાગ્યે જ સંશોધિત	વારંવાર અપડેટ

- સ્કીમા: ડેટાબેઝ સંગઠન અને મર્યાદાઓ વાળે છે
- ઇન્સ્ટન્સ: ચોક્કસ ક્ષણે ડેટાબેઝ સામગ્રીનો સ્નેપશૉટ
- સંબંધ: સ્કીમા રચના વ્યાખ્યાપિત કરે છે, ઇન્સ્ટન્સ ડેટા સમાવે છે

મેમરી ટ્રીક

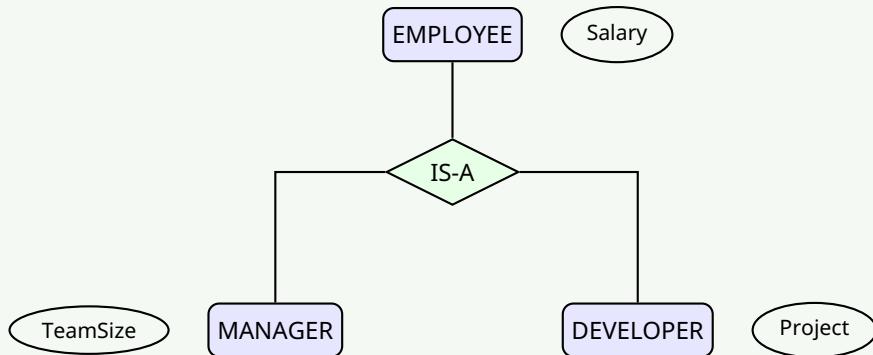
"SI - Structure vs Information"

પ્રશ્ન 2(b) [4 ગુણ]

સ્પેશલાઈઝેશન ઉદાહરણ સાથે સમજાવો.

જવાબ

સ્પેશલાઈઝેશન: ચોક્કસ લાક્ષણિકતાઓના આધારે સુપરકલાસમાંથી સબકલાસ બનાવવાની પ્રક્રિયા.



આકૃતિ 3. સ્પેશલાઈઝેશન હાયરાર્કોર્ડ

- ટોપ-ડાઉન અપ્રોગ્રામ: સામાન્ય એન્ટિટીઓ ચોક્કસ એન્ટિટીઓ તરફ
- ઇનહેરિટન્સ: સબકલાસેસ સુપરકલાસના ગુણધર્મો વારસામાં લે છે
- ડિસજોઇન: મેનેજર અને ડેવલપર અલગ કટેગરી છે
- ઉદાહરણ: એમ્પલોયી મેનેજર અને ડેવલપરમાં વિશેષીકૃત

મેમરી ટ્રીક

"STID - Specialization Takes Inheritance Down"

પ્રશ્ન 2(c) [7 ગુણ]

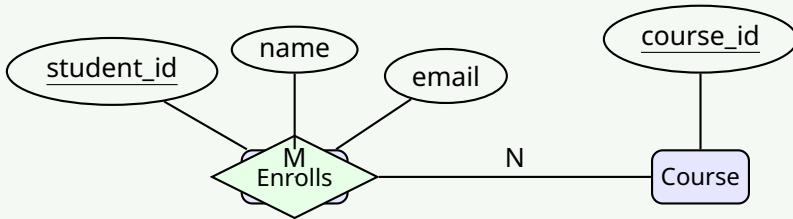
ER ડાયાગ્રામ શું છે? ER ડાયાગ્રામમાં વપરાતા વિવિધ પ્રતીકોને ઉદાહરણ સાથે સમજાવો.

જવાબ

ER ડાયાગ્રામ: ડેટાબેઝ ડિગ્રાઇનમાં એન્ટિટીઝ, એટ્રિબ્યુટ્સ અને સંબંધો દર્શાવતી ગ્રાફિકલ પ્રતિનિધિત્વ. ટેબલ:

કોષ્ટક 6. ER ડાયાગ્રામ પ્રતીકો

પ્રતીક	આકાર	હેતુ	ઉદાહરણ
એન્ટિટી	લંબચોર્સ	વાસ્તવિક વિશ્વનો ઓફ્જેક્ટ	Student, Course
એટ્રિબ્યુટ	અંડાકાર	એન્ટિટીના ગુણધર્મો	Name, Age, ID
સંબંધ	હીરો	એન્ટિટી કોન્ફેન્સન્સ	Enrolls, Takes
પ્રાઇમરી ક્રી	અન્ડરલાઇન અંડાકાર	યુનિક આઇડેન્ટિફિક્યુર	Student_ID



આકૃતિ 4. ER ડાયાગ્રામ ઉદાહરણ

- એન્ટિટી સેટ્સ: સમાન ગુણધર્મો ધરાવતી સમાન એન્ટિટીઓનો સંગ્રહ
- વીક એન્ટિટી: ઓળખ માટે સ્ટ્રોંગ એન્ટિટી પર આધારિત
- કાર્ડિનાલિટી: સંબંધ સહભાગિતા વ્યાખ્યાપિત કરે છે (1:1, 1:M, M:N)
- પાર્ટિસિપેશન: ટોટલ (ડબલ લાઇન) અથવા પાર્શ્વિયલ (સિંગલ લાઇન)

મેમરી ટ્રીક

"EARP - Entities And Relationships Program"

પ્રશ્ન 2(a OR) [3 ગુણ]

DA અને DBA નો તફાવત સમજાવો.

જવાબ

ટેબલ:

કોષ્ટક 7. DA vs DBA

પાસું	ડેટા એડમિનિસ્ટ્રેટર (DA)	ડેટાબેઝ એડમિનિસ્ટ્રેટર (DBA)
ફોકસ	ડેટા પોલિસીઝ અને સ્ટાન્ડર્ડ્સ	તકનીકી ડેટાબેઝ ઓપરેશન્સ
સ્તર	વ્યૂહાત્મક આયોજન	ઓપરેશનલ અમલીકરણ
સ્કોપ	સંસ્થા-વ્યાપી ડેટા	ચોક્કસ ડેટાબેઝ સિસ્ટમ્સ

- DA: સંસ્થાકીય સંસાધન તરીકે ડેટાનું સંચાલન કરે છે
- DBA: તકનીકી ડેટાબેઝ જાળવણી અને પર્ફોર્મન્સ સંભાળે છે
- સહયોગ: DA નીતિઓ સેટ કરે છે, DBA તેમને અમલમાં મૂકે છે

મેમરી ટ્રીક

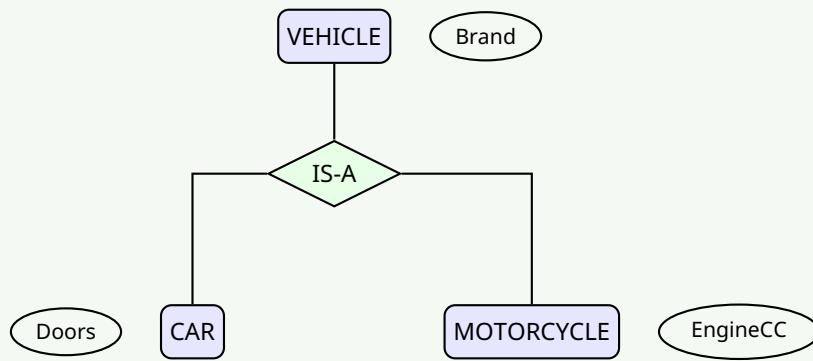
"DA-DBA: Design Authority - Database Builder Administrator"

પ્રશ્ન 2(b OR) [4 ગુણ]

જનરલાઈઝન ઉદાહરણ સાથે સમજાવો.

જવાબ

જનરલાઈઝન: સમાન એન્ટિટીને સામાન્ય સુપરકલાસમાં જોડવાની બોટમ-અપ પ્રક્રિયા.



આકૃતિ 5. જનરલાઇઝેશન હાયરકોર્ટી

- બોટમ-અપ અપ્રોચે: ચોક્કસ એન્ટિટીઝથી સામાન્ય એન્ટિટી તરફ
- કોમન એટ્રિબ્યુટ્સ: સહેજ ગુણધર્મો સુપરક્લાસમાં ખસોડાય છે
- સ્પેશયલાઇઝેશન રિવર્સ્: સ્પેશયલાઇઝેશન પ્રક્રિયાનું વિપરીત
- ઉદાહરણ: કાર અને મોટરસાઇકલ વાહનમાં સામાન્યીકૃત

મેમરી ટ્રીક

“GBCS - Generalization Brings Common Superclass”

પ્રશ્ન 2(c OR) [7 ગુણ]

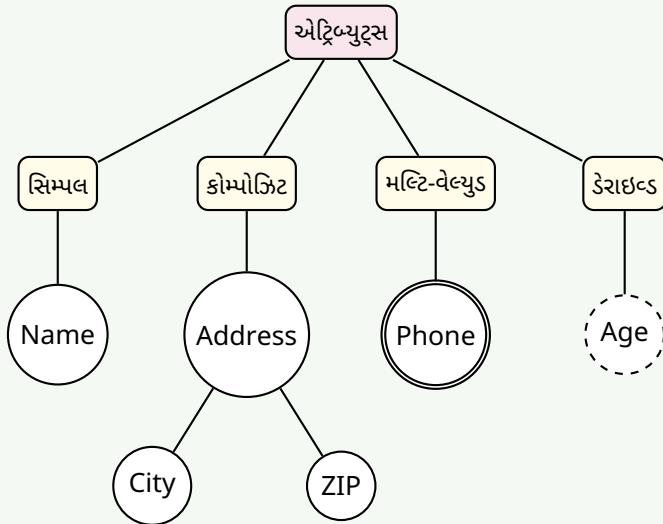
એટ્રિબ્યુટ શું છે? વિવિધ પ્રકારના એટ્રિબ્યુટ્સ ઉદાહરણ સાથે સમજાવો.

જવાબ

એટ્રિબ્યુટ: એન્ટિટીનું વર્ણન કરતી ગુણવત્તા અથવા લાક્ષણિકતા.
ટેબલ:

કોષ્ટક 8. એટ્રિબ્યુટ પ્રકારો

એટ્રિબ્યુટ પ્રકાર	વર્ણન	ઉદાહરણ
સિમ્પલ	વધુ વિભાજિત કરી શકાતું નથી	Age, Name
કોમ્પોઝિટ	ઉપવિભાગ કરી શકાય છે	Address (Street, City, ZIP)
સિંગલ-વેલ્યુડ	એન્ટિટી દીઠ એક મૂલ્ય	Student_ID
મલ્ટિ-વેલ્યુડ	અનેક મૂલ્યો શક્ય	Phone_numbers
ડેરાઇસ	અન્ય એટ્રિબ્યુટ્સમાંથી ગણાય છે	Age from Birth_date



આકૃતિ 6. એટ્રિબ્યુટ્સના પ્રકાર

- કી એટ્રિબ્યુટ: એન્ટિટી ઇન્સ્ટન્સેસને ચુનિકળી ઓળખે છે
- નાલ વેલ્યુડ: એટ્રિબ્યુટ્સ કે જેમાં કોઈ મૂલ્ય ન હોઈ શકે
- ડિફોલ્ટ વેલ્યુડ: નિર્દિષ્ટ ન હોય ત્યારે પૂર્વનિર્ધારિત મૂલ્યો
- કન્સ્ટ્રેઇન્ડ્સ: એટ્રિબ્યુટ મૂલ્યોને સંચાલિત કરતા નિયમો

મેમરી ટ્રીક

“SCSMD-K - Simple Composite Single Multi Derived Key”

પ્રશ્ન 3(a) [3 ગુણ]

SQL માં GRANT અને REVOKE સ્ટેટમેન્ટ સમજાવો.

જવાબ

ટેબલ:

કોષ્ટક 9. GRANT અને REVOKE

સ્ટેટમેન્ટ	હેતુ	સિન્ક્રેક્સ ઉદાહરણ
GRANT	વપરાશકર્તાનોને વિશેષાધિકારો પ્રદાન કરે છે	GRANT SELECT ON table TO user
REVOKE	વપરાશકર્તા વિશેષાધિકારો દૂર કરે છે	REVOKE INSERT ON table FROM user

```

1   -- Grant privileges
2   GRANT SELECT, INSERT ON employees TO john;
3   GRANT ALL PRIVILEGES ON database TO admin;
4
5   -- Revoke privileges
6   REVOKE DELETE ON employees FROM john;
7   REVOKE ALL ON database FROM user;
  
```

- વિશેષાધિકારો: SELECT, INSERT, UPDATE, DELETE, ALL
- ઓફ્જેક્ટ્સ: ટેબલ્સ, વ્યુડ, ડેટાબેઝિસ, પ્રોસીજર્સ
- સુરક્ષા: ડેટા એક્સેસ અનુ મોડિફિકેશન રાઇટ્સનું નિયંત્રણ

મેમરી ટ્રીક

``GR - Grant Rights, Remove Rights''

પ્રશ્ન 3(b) [4 ગુણ]

નીચેના Character functions સમજાવો. 1) INITCAP 2) SUBSTR

જવાબ

ટેબલ:

કોષ્ટક 10. Character ફંક્શન્સ

ફંક્શન	હેતુ	સિન્ક્રેસ	ઉદાહરણ
INITCAP	દેરેક શબ્દનો પહેલો અક્ષર મોટો કરે છે	INITCAP(string)	INITCAP('hello world') = 'Hello World'
SUBSTR	સ્ટ્રિંગમાંથી સબસ્ટ્રિંગ કાઢે છે	SUBSTR(string, start, length)	SUBSTR('Database', 1, 4) = 'Data'

```

1 -- INITCAP examples
2 SELECT INITCAP('database management') FROM dual; -- Database Management
3 SELECT INITCAP('gtu university') FROM dual; -- Gtu University
4
5 -- SUBSTR examples
6 SELECT SUBSTR('Programming', 1, 7) FROM dual; -- Program
7 SELECT SUBSTR('Database', 5) FROM dual; -- base

```

- INITCAP: સ્ટ્રિંગને યોગ્ય કેસ ફોર્મેટમાં કન્વર્ટ કરે છે
- SUBSTR: પેરામીટર્સ છે સ્ટ્રિંગ, શરૂઆતની સ્થિતિ, વૈકલ્પિક લંબાઈ
- વપરાશ: ટેક્સ્ટ ફોર્મેટિંગ અને સ્ટ્રિંગ મેનિપ્યુલેશન ઓપરેશન્સ

મેમરી ટ્રીક

``IS - Initialize String, Split String''

પ્રશ્ન 3(c) [7 ગુણ]

નીચે દર્શાવેલ ટેબલને ધ્યાનમાં લઈ આપેલ કવેરીજના જવાબ લખો. stud_master (enroll_no, name, city, dept)

જવાબ

```

1 -- 1. IT dept માં અભ્યાસ કરતા બધા વદ્દ્યાર્થીઓની વિભિન્નો દર્શાવો
2 SELECT * FROM stud_master
3 WHERE dept = 'IT';
4
5 -- 2. 'p' થી શરૂ થતા નામ વશીની બધી માહિતી મેળવો
6 SELECT * FROM stud_master
7 WHERE name LIKE 'p%';
8
9 -- 3. ટેબલમાં નવો વદ્દ્યાર્થી દાખલ કરો
10 INSERT INTO stud_master (enroll_no, name, city, dept)
11 VALUES ('202501', 'John Smith', 'Mumbai', 'CS');

```

```

12 -- 4. stud_master ટેબલમાં gender નામનું નવું કોલમ ઉમેરો
13 ALTER TABLE stud_master
14 ADD gender VARCHAR(10);
15
16 -- 5. stud_master ટેબલની પંકૃતિઓની સંખ્યા ગણો
17 SELECT COUNT(*) FROM stud_master;
18
19 -- 6. enroll_no ના અવરોહી ફ્રેમમાં બધી વદ્દ્યારથી વળિતો દર્શાવો
20 SELECT * FROM stud_master
21 ORDER BY enroll_no DESC;
22
23 -- 7. ડેટા સાથે stud_master ટેબલનો નાશ કરો
24 DROP TABLE stud_master;
25

```

ટેબલ:

કોષ્ટક 11. SQL કવરીઝ

કવરી પ્રકાર	SQL કમાન્ડ	હેતુ
SELECT	ડેટા મેળવે છે	રેકૉર્ડ્સ દર્શાવે છે
INSERT	નવો ડેટા ઉમેરે છે	રેકૉર્ડ્સ બનાવે છે
ALTER	રચના સંશોધિત કરે છે	કોલમ્સ ઉમેરે છે
COUNT	એગ્રિગેટ ફંક્શન	પંકૃતિઓ ગણો છે

મેમરી ટ્રીક

“SIAC-DOC - SQL Is A Complete Database Operations Collection”

પ્રશ્ન 3(a OR) [3 ગુણ]

SQL માં equi join ઉદાહરણ સાથે સમજાવો.

જવાબ

Equi Join: સમાન કોલમ્સના આધારે ટેબલ્સને જોડવા માટે સમાનતા શરતનો ઉપયોગ કરતું જોઈન ઓપરેશન.

```

1 -- Equi Join ઉદાહરણ
2 SELECT s.name, c.course_name
3   FROM students s, courses c
4 WHERE s.course_id = c.course_id;
5
6 -- JOIN સનિયુક્તસનો ઉપયોગ
7 SELECT s.name, c.course_name
8   FROM students s
9 JOIN courses c ON s.course_id = c.course_id;

```

- સમાનતા ઓપરેટર: કોલમ મૂલ્યો મેચ કરવા માટે = નો ઉપયોગ
- કોમન કોલમ્સ: ટેબલ્સમાં સંબંધિત એટ્રિબ્યુટ્સ હોવા જોઈએ
- પરિણામ: મેચના આધારે અનેક ટેબલ્સમાંથી સંયુક્ત ડેટા

મેમરી ટ્રીક

“EJ - Equal Join”

પ્રશ્ન 3(b OR) [4 ગુણ]

નીચેના Aggregate functions સમજાવો. 1) MAX 2) SUM

જવાબ

ટેબલ:

કોષ્ટક 12. Aggregate ફંક્શન્સ

ફંક્શન	હેતુ	સિન્ક્રેસ	ઉદાહરણ
MAX	મહત્તમ મૂલ્ય પરત કરે છે	MAX(column)	MAX(salary) = 50000
SUM	મૂલ્યોનો કુલ સરવાળો પરત કરે છે	SUM(column)	SUM(marks) = 450

```

1 -- MAX ઉદાહરણો
2 SELECT MAX(salary) FROM employees; -- સૌથી વધુ પગાર
3 SELECT MAX(age) FROM students; -- સૌથી જૂના વાહિયાર્થીની ઉંમર
4
5 -- SUM ઉદાહરણો
6 SELECT SUM(credits) FROM courses; -- કુલ ફેરેડટિસ
7 SELECT SUM(price * quantity) FROM orders; -- કુલ ઓર્ડર મૂલ્ય

```

- એગ્રિગેટ ફંક્શન્સ: અનેક પંક્તિઓ પર કામ કરે છે, એક મૂલ્ય પરત કરે છે
- NULL હેન્ડલિંગ: ગણતરીમાં NULL મૂલ્યોને અવગણો છે
- GROUP BY: કેટેગરી-વાઇડ પરિણામો માટે ગુપ્પિંગ સાથે ઉપયોગ કરી શકાય છે

મેમરી ટ્રીક

``MS - Maximum Sum''

પ્રશ્ન 3(c OR) [7 ગુણ]

નીચે દર્શાવેલ ટેબલ માટે SQL કવરેજ લખો: PRODUCT_Master: (prod_no, prod_name, profit, quantity, sell_price, cost_price)

જવાબ

```

1 -- 1. PRODUCT_Master ટેબલ બનાવો
2 CREATE TABLE PRODUCT_Master (
3     prod_no VARCHAR(10) PRIMARY KEY,
4     prod_name VARCHAR(50),
5     profit NUMBER(10,2),
6     quantity NUMBER,
7     sell_price NUMBER(10,2),
8     cost_price NUMBER(10,2)
9 );
10
11 -- 2. આ ટેબલમાં એક રેકોર્ડ દાખલ કરો
12 INSERT INTO PRODUCT_Master VALUES
13 ('P001', 'Laptop', 15000, 10, 45000, 30000);
14
15 -- 3. 20000 થી વધુ નફો ધરાવતા પ્રોડક્ટ શોધો
16 SELECT * FROM PRODUCT_Master
17 WHERE profit > 20000;
18
19 -- 4. 5 થી ઓછી quantity ધરાવતા પ્રોડક્ટ ડલીટ કરો

```

```

20  DELETE FROM PRODUCT_Master
21  WHERE quantity < 5;
22
23  -- 5. 5000 થી વધુ sell_price ધરાવતા પ્રોડક્ટમાં 2% નફો ઉમેરો
24  UPDATE PRODUCT_Master
25  SET profit = profit * 1.02
26  WHERE sell_price > 5000;
27
28  -- 6. PRODUCT_Master માં total_price નામનું નવું ફીલ્ડ ઉમેરો
29  ALTER TABLE PRODUCT_Master
30  ADD total_price NUMBER(10,2);
31
32  -- 7. કોઈ દુપ્લિકેટ ડેટા વગર પ્રોડક્ટ નામ શોધો
33  SELECT DISTINCT prod_name FROM PRODUCT_Master;

```

મેમરી ટ્રીક

“CIDFAUD - Create Insert Delete Find Add Update Distinct”

પ્રશ્ન 4(a) [3 ગુણ]

fully functional dependency ઉદાહરણ સાથે સમજાવો.

જવાબ

Fully Functional Dependency: એટ્રિબ્યુટ સંપૂર્ણ રીતે ફંક્શનલ ડિપેન્ડન્ટ છે જો તે સંપૂર્ણ પ્રાઇમરી કી પર આધારિત હોય, આંશિક કી પર નહીં.

ટેબલ:

કોષ્ટક 13. ડિપેન્ડન્સી પ્રકાર

ડિપેન્ડન્સી પ્રકાર	વ્યાખ્યા	ઉદાહરણ
Full FD	સંપૂર્ણ કી પર આધારિત	(Student_ID, Course_ID) → Grade
Partial FD	કીના ભાગ પર આધારિત	(Student_ID, Course_ID) → Student_Name

ઉદાહરણ: Student_Course(Student_ID, Course_ID, Student_Name, Grade)

Full FD: (Student_ID, Course_ID) → Grade Partial FD: Student_ID → Student_Name

- સંપૂર્ણ કી: કોમ્પોઝિટ પ્રાઇમરી કીના બધા એટ્રિબ્યુટ્સ જરૂરી
- નોન-કી એટ્રિબ્યુટ: સંપૂર્ણ પ્રાઇમરી કી કોમ્પ્લિનેશન પર આધારિત
- 2NF જરૂરિયાત: આંશિક ડિપેન્ડન્સીઝ દૂર કરે છે

મેમરી ટ્રીક

“FFD - Full Function Dependency”

પ્રશ્ન 4(b) [4 ગુણ]

નીચે દર્શાવેલ રિલેશનલ સ્કીમાને ધ્યાનમાં લઈ રિલેશનલ અલજીબ્રા એક્સપ્રેશન આપો: Employee (Emp_name, Emp_id, birth_date, Post, salary)

જવાબ

1. Post="Clerk" ધરાવતા બધા કર્મચારીઓની યાદી બનાવો

$$\sigma_{Post='Clerk'}(Employee)$$

2. salary > 2000 અને post='Manager' ધરાવતા Emp_id અને Emp_name શોધો

$$\pi_{Emp_id, Emp_name}(\sigma_{salary > 2000 \wedge Post='Manager'}(Employee))$$

ટેબલ:

કોષ્ટક 14. રિલેશનલ અલજુભા પ્રતીકો

પ્રતીક	ઓપરેશન	હેતુ
σ	સિલેક્શન	શરત આધારિત પંક્તિઓ ફિલ્ટર કરે છે
π	પ્રોજેક્શન	ચોક્કસ કોલમ્સ પરસંદ કરે છે
\wedge	AND	તાર્કિક સંયોજન

- સિલેક્શન (σ): નિર્દિષ્ટ શરતો પૂરી કરતી પંક્તિઓ પરસંદ કરે છે
- પ્રોજેક્શન (π): પરિણામમાંથી જરૂરી કોલમ્સ પરસંદ કરે છે
- સંયુક્ત ઓપરેશન્સ: અનેક ઓપરેશન્સનો એકસાથે ઉપયોગ કરી શકાય છે

મેમરી ટ્રીક

"SPA - Select Project And"

પ્રશ્ન 4(c) [7 ગુણ]

2NF ના કાઈટેરિયા શું છે? આપેલ રિલેશનમાં વિવિધ ફંક્શનલ ડિપેન્ડ-સીજ શોધો અને તેને 2NF માં નોર્મલાઇઝ કરો.

જવાબ

2NF કાઈટેરિયા:

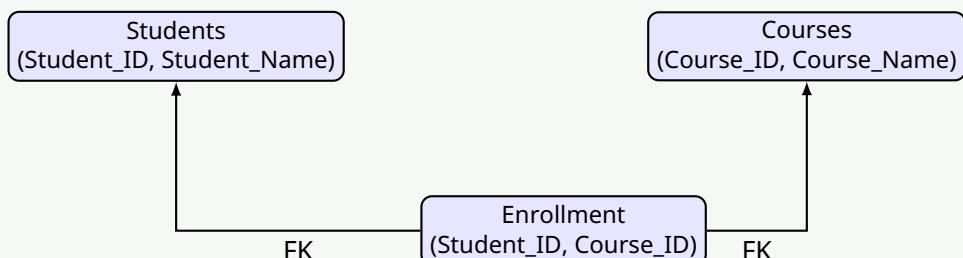
- 1NF માં હોવું જોઈએ
- પ્રાઇમરી કી પર કોઈ આંશિક ફંક્શનલ ડિપેન્ડ-સીજ ન હોવી જોઈએ

આપેલ ટેબલ: Student_Course(Student_ID, Course_ID, Student_Name, Course_Name)

ફંક્શનલ ડિપેન્ડ-સીજ્સ:

- Student_ID → Student_Name (Partial FD)
- Course_ID → Course_Name (Partial FD)
- (Student_ID, Course_ID) → (Student_Name, Course_Name) (Full FD)

2NF નોર્મલાઇઝનાન:



આકૃતિ 7. 2NF ડિક્રમ્પોઝિશન

1 ટેબલ 1: Students

2 Students(Student_ID, Student_Name)

3

- 4 - ટેબલ 2: Courses
Courses(Course_ID, Course_Name)
- 5 - ટેબલ 3: Enrollment
Enrollment(Student_ID, Course_ID)

મેમરી ટ્રીક

``2NF - Two Normal Form removes partial dependencies''

પ્રશ્ન 4(a OR) [3 ગુણ]

3NF ઉદાહરણ સાથે સમજાવો.

જવાબ

3NF (Third Normal Form): 2NF માં હોય અને પ્રાઇમરી કી પર કોઈ ટ્રાન્ઝિટિવ ડિપેન્ડન્સીઝ ન હોય તેવું ટેબલ.

ટેબલ:

કોષ્ટક 15. 3NF જરૂરિયાત

નોર્મલ ફોર્મ	જરૂરિયાત	નાબૂદ કરે છે
3NF	2NF માં + કોઈ ટ્રાન્ઝિટિવ ડિપેન્ડન્સીઝ નહીં	ટ્રાન્ઝિટિવ FD

ઉદાહરણ: Employee(Emp_ID, Dept_ID, Dept_Name)

ટ્રાન્ઝિટિવ ડિપેન્ડન્સી: Emp_ID → Dept_ID → Dept_Name

3NF ઉકેલ: Employee(Emp_ID, Dept_ID) Department(Dept_ID, Dept_Name)

- ટ્રાન્ઝિટિવ ડિપેન્ડન્સી: $A \rightarrow B \rightarrow C$ જ્યાં A પ્રાઇમરી કી છે
- નોન-કી ટુ નોન-કી: નોન-કી એટ્રિબ્યુટ્સ વચ્ચે ડિપેન્ડન્સી
- ડિકમ્પોઝિશન: ટ્રાન્ઝિટિવ ડિપેન્ડન્સીઝ દૂર કરવા માટે ટેબલ વિભાજિત કરવું

મેમરી ટ્રીક

``3NF - Third Normal Form removes Transitive dependencies''

પ્રશ્ન 4(b OR) [4 ગુણ]

નીચે દર્શાવેલ રિલેશનલ સ્કીમને ધ્યાનમાં લઈ રિલેશનલ અલજુબ્રા એક્સપ્રેશન આપો: Students (Name, SPI, DOB, Enrollment No)

જવાબ

1. SPI 7.0 થી વધુ હોય તેવા બધા વિદ્યાર્થીઓની યાદી બનાવો

$$\sigma_{SPI > 7.0}(\text{Students})$$

2. enrollment number 007 હોય તેવા વિદ્યાર્થીનું name, DOB દર્શાવો

$$\pi_{Name, DOB}(\sigma_{Enrollment_No='007'}(\text{Students}))$$

ટેબલ:

કોષ્ટક 16. રિલેશનલ અલજુબ્રા

કવેરી	રિલેશનલ અલજીબા	હેતુ
ફિલ્ટર	σ (condition)	પંક્તિઓ પસંદ કરે છે
પ્રોજેક્ટ	π (attributes)	કોલમ્સ પસંદ કરે છે

- પહેલા સિલેક્શન: પ્રોજેક્શન પહેલાં શરતો લાગુ કરો
- ચોક્કસ મૂલ્ય: સ્ટ્રાંગ લિટરલ્સ માટે કવોટ્સનો ઉપયોગ કરો
- કોલમ નામો: ચોક્કસ એટ્રિબ્યુટ નામો જરૂરી

મેમરી ટ્રીક

“SPI-DOB: Select Project Information - Display Output Better”

પ્રશ્ન 4(c OR) [7 ગુણ]

1NF ના કાઈટેરિયા શું છે? આપેલ ટેબલને બે અલગ અલગ પદ્ધતિથી 1NF માં નોર્મલાઇઝ કરો.

જવાબ

1NF કાઈટેરિયા:

- દરેક સેલમાં એક જ અણુ મૂલ્ય હોય જોઈએ
- કોઈ પુનરાવર્તિ જૂથો અથવા એરેઝ નહીં
- દરેક પંક્તિ અનાન્ય હોવી જોઈએ

આપેલ ટેબલ:

EnrollmentNo	Name	Subjects
001	DEF	Maths,Physics,Chemistry
002	XYZ	History,Biology,English

પદ્ધતિ 1 - અલગ પંક્તિઓ:

કોષ્ટક 17. 1NF - અલગ પંક્તિઓ

EnrollmentNo	Name	Subject
001	DEF	Maths
001	DEF	Physics
001	DEF	Chemistry
002	XYZ	History
002	XYZ	Biology
002	XYZ	English

પદ્ધતિ 2 - અલગ ટેબલ્સ:

```

1 -- Students ટેબલ
2 Students(EnrollmentNo, Name)
3
4 -- Subjects ટેબલ
5 Subjects(SubjectID, SubjectName)
6
7 -- Student_Subjects ટેબલ
8 Student_Subjects(EnrollmentNo, SubjectID)

```

મેમરી ટ્રીક

“1NF - One Normal Form creates Atomic values”

પ્રશ્ન 5(a) [3 ગુણ]

ટ્રાન્ઝેક્શનની ACID પ્રોપર્ટીઝ સમજાવો.

જવાબ

ટેબલ:

કોષ્ટક 18. ACID પ્રોપર્ટીઝ

પ્રોપર્ટી	વર્ણન	હેતુ
Atomicity	સંપૂર્ણ અથવા કંઈ જ એક્ઝિક્યુશન	ટ્રાન્ઝેક્શન સંપૂર્ણતા
Consistency	ડેટાબેઝ માન્ય રહે છે	ડેટા અખંડિતતા
Isolation	સમવર્તી ટ્રાન્ઝેક્શન્સ સ્વતંત્ર	હસ્તક્ષેપ ટાળવો
Durability	કમિટ થયેલા ફેરફારો કાયમી	ડેટા સ્થિરતા

- **Atomicity:** ટ્રાન્ઝેક્શન સંપૂર્ણ રીતે એક્ઝિક્યુટ થાય અથવા બિલકુલ નહીં
- **Consistency:** ટ્રાન્ઝેક્શન પહેલાં/પછી ડેટાબેઝ કન્સ્ટ્રેઇન્ડ્સ જાળવાય છે
- **Isolation:** ટ્રાન્ઝેક્શન્સ એક્બીજા સાથે હસ્તક્ષેપ કરતા નથી
- **Durability:** એકવાર કમિટ થયા પછી, ફેરફારો સિસ્ટમ ફેઇલ્યુર્સમાં ટકી રહે છે

મેમરી ટ્રીક

“ACID - All Consistent Isolated Durable”

પ્રશ્ન 5(b) [4 ગુણ]

નીચે દર્શાવેલ સ્પેસિફિકેશન મુજબ ટેબલ બનાવો: STUDENT: (stu_id, stu_name, Address, City, contact_no, Branch_name)

જવાબ

```

1 CREATE TABLE STUDENT (
2   stu_id VARCHAR(10) PRIMARY KEY,
3   stu_name VARCHAR(50) NOT NULL,
4   Address VARCHAR(100),
5   City VARCHAR(30),
6   contact_no NUMBER(10),
7   Branch_name VARCHAR(20) CHECK (Branch_name IN ('IT', 'Computer', 'Electrical', 'Civil'))
8 );

```

ટેબલ:

કોષ્ટક 19. ટેબલ કન્સ્ટ્રેઇન્ડ્સ

કન્સ્ટ્રેઇન્ડ	હેતુ	અમલીકરણ
NOT NULL	ફરજિયાત ફીલ્ડ	stu_name NOT NULL
CHECK	વેલ્યુ વેલિડેશન	Branch_name IN (...)

- પ્રાઇમરી કી: stu_id દરેક વિદ્યાર્થીને અન્ન્ય રીતે ઓળખે છે

- NOT NULL: stu_name ખાલી હોઈ શકે નહીં
- CHECK કન્સ્ટ્રેઇન્ટ: Branch_name નિર્દિષ્ટ મૂલ્યો સુધી મર્યાદિત
- ડેટા ટાઇપ્સ: દરેક ફીલ્ડ માટે યોગ્ય સાઇઝ

મેમરી ટ્રીક

“CNPD - Constraints Names Primary Datatypes”

પ્રશ્ન 5(c) [7 ગુણા]

ટ્રિગર શું છે? Oracle માં ટ્રિગર બનાવવા માટે સિન્ટેક્સ લખો. સિમ્પલ ટ્રિગર બનાવો.

જવાબ

ટ્રિગર: વિશેષ સ્ટોર્ડ પ્રોસીજર જે ડેટાબેઝ ઇવેન્ટ્સના પ્રતિભાવમાં આપોઆપ એક્ઝિક્યુટ થાય છે.
Oracle ટ્રિગર સિન્ટેક્સ:

```

1 CREATE [OR REPLACE] TRIGGER trigger_name
2 {BEFORE | AFTER | INSTEAD OF} {INSERT | UPDATE | DELETE}
3 ON table_name
4 [FOR EACH ROW]
5 [WHEN condition]
6 DECLARE
7   -- Variable declarations
8 BEGIN
9   -- Trigger logic
10 END;

```

સિમ્પલ ટ્રિગર ઉદાહરણ:

```

1 CREATE OR REPLACE TRIGGER display_student_trigger
2 BEFORE INSERT ON STUDENT
3 FOR EACH ROW
4 BEGIN
5   DBMS_OUTPUT.PUT_LINE('Inserting student: ' || :NEW.stu_name ||
6     ' with ID: ' || :NEW.stu_id);
7 END;

```

ટેબલ:

કોષ્ટક 20. ટ્રિગર પ્રકારો

ટ્રિગર પ્રકાર	ક્યારે એક્ઝિક્યુટ થાય	હેતુ
BEFORE	DML ઓપરેશન પહેલાં	વેલિડેશન, મોડિફિકેશન
AFTER	DML ઓપરેશન પછી	લોગિંગ, ઓડિટિંગ
FOR EACH ROW	રો-લેવલ ટ્રિગર	પ્રતિ પંક્તિ એક્ઝિક્યુશન

- :NEW: દાખલ/અપડેટ કરવામાં આવતા નવા મૂલ્યોનો સંદર્ભ
- :OLD: ડિલીટ/અપડેટ કરવામાં આવતા જૂના મૂલ્યોનો સંદર્ભ
- આપોઆપ એક્ઝિક્યુશન: નિર્દિષ્ટ ઇવેન્ટ્સ પર આપોઆપ ફાયર થાય છે
- બિઝનેસ લોજિક: જાટેલ બિઝનેસ નિયમો લાગુ કરે છે

મેમરી ટ્રીક

“TBA-FEN - Triggers Before After For Each New”

પ્રશ્ન 5(a OR) [3 ગુણ]

ટ્રાન્ઝેક્શનમાં કન્કરન્સી કંટ્રોલના પ્રોબ્લેમ્સ સમજાવો.

જવાબ

ટેબલ:

કોષ્ટક 21. કન્કરન્સી કંટ્રોલ સમસ્યાઓ

સમસ્યા	વર્ણન	ઉદાહરણ
Lost Update	એક ટ્રાન્ઝેક્શન બીજાના ફેરફારો પર લખે છે	T1, T2 સમાન રેકૉર્ડ અપડેટ કરે છે
Dirty Read	અનકમિટ ડેટા વાંચવો	T1 T2 ના અનકમિટ ફેરફારો વાંચે છે
Unrepeatable Read	સમાન કવેરી અલગ પરિણામો આપે છે	T1 વાંચે, T2 અપડેટ કરે, T1 ફરી વાંચે

- **Phantom Read:** સમાન ટ્રાન્ઝેક્શનમાં કવેરીઝ વર્ચ્યે નવી પંક્તિઓ દેખાય છે
- **Deadlock:** બે ટ્રાન્ઝેક્શન્સ એકબીજાના લોક્સની રાહ જુઓ છે
- **Inconsistent Analysis:** ડેટા સંશોધિત થતો હોય ત્યારે વાંચવો

મેમરી ટ્રીક

“LDU-PID - Lost Dirty Unrepeatable Phantom Inconsistent Deadlock”

પ્રશ્ન 5(b OR) [4 ગુણ]

નીચે દર્શાવેલ સ્પેસિફિકેશન મુજબ ટેબલ બનાવો: STUDENT: (stu_id, stu_name, Address, City, contact_no, Branch_name)

જવાબ

```

1 CREATE TABLE STUDENT (
2   stu_id VARCHAR(10) PRIMARY KEY CHECK (stu_id LIKE 'S%'),
3   stu_name VARCHAR(50),
4   Address VARCHAR(100),
5   City VARCHAR(30),
6   contact_no NUMBER(10),
7   Branch_name VARCHAR(20)
8 );

```

ટેબલ:

કોષ્ટક 22. ટેબલ કન્સ્ટ્રેઇન્ટ્સ

કન્સ્ટ્રેઇન્ટ	અમલીકરણ	હેતુ
PRIMARY KEY	stu_id PRIMARY KEY	અનન્ય ઓળખ
CHECK	stu_id LIKE 'S%'	'S' થી શરૂ થવું જોઈએ

- પ્રાઇમરી કી: stu_id અનન્ય આઇડેન્ટિફિયર તરીકે કામ કરે છે
- પેર્સન ચેક: stu_id અક્ષર 'S' થી શરૂ થવું જોઈએ
- ડેટા ટાઇપ્સ: યોગ્ય ફીલ્ડ સાઇઝ અને ટાઇપ્સ
- કન્સ્ટ્રેઇન્ટ વેલિડેશન: ડેટાબેઝ આપોઆપ નિયમો લાગુ કરે છે

મેમરી ટ્રીક

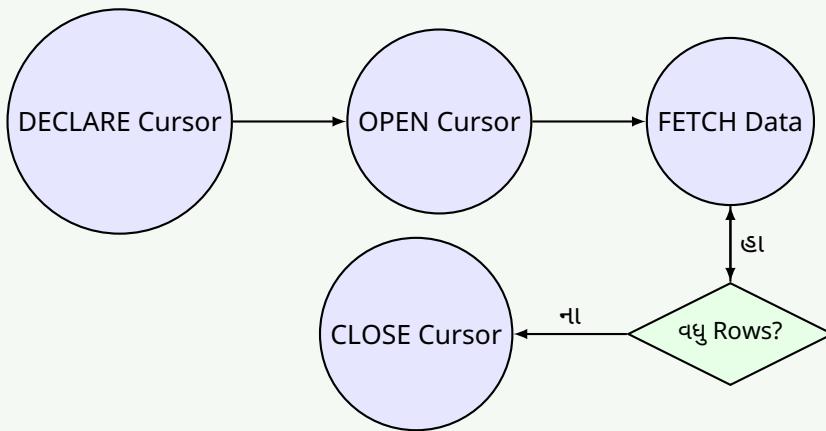
“PKC-ST - Primary Key Check Starts”

પ્રશ્ન 5(c OR) [7 ગુણ]

એક્સપ્લિસિટ કર્સર શું છે? એક્સપ્લિસિટ કર્સર ઉદાહરણ સાથે સમજાવો.

જવાબ

એક્સપ્લિસિટ કર્સર: અનેક પંક્તિઓ પરત કરતા SELECT સ્ટેટમેન્ટ્સ હેન્ડલ કરવા માટે પ્રોગ્રામેટિક કંટ્રોલ સાથે વપરાશકર્તા-વ્યાખ્યાયિત કર્સર. કર્સર ઓપરેશન્સ:



આફ્ટિ 8. એક્સપ્લિસિટ કર્સર લાઈફસાયકલ

```

1  -- ડાફુલ રેશન
2  DECLARE
3    CURSOR student_cursor IS
4      SELECT stu_id, stu_name FROM STUDENT WHERE city = 'Ahmedabad';
5      v_id STUDENT.stu_id%TYPE;
6      v_name STUDENT.stu_name%TYPE;
7  BEGIN
8    -- કર્સર ઓપન કરો
9    OPEN student_cursor;
10
11   -- ડેટા ફેચ કરો
12  LOOP
13    FETCH student_cursor INTO v_id, v_name;
14    EXIT WHEN student_cursor%NOTFOUND;
15
16    DBMS_OUTPUT.PUT_LINE('ID: ' || v_id || ', Name: ' || v_name);
17  END LOOP;
18
19   -- કર્સર બંધ કરો
20  CLOSE student_cursor;
21  END;
  
```

ટેબલ:

કોષ્ટક 23. કર્સર કમાન્ડ્સ

ઓપરેશન	હેતુ	સિન્ટેક્સ
DECLARE	કર્સર ડિફાઇન કરવું	CURSOR name IS SELECT...
OPEN	કર્સર ઇનિશિયલાઇઝ કરવું	OPEN cursor_name
FETCH	ડેટા મેળવવો	FETCH cursor INTO variables
CLOSE	રિસોર્સ છોડવા	CLOSE cursor_name

- મેન્યુઅલ કંટ્રોલ: પ્રોગ્રામર કર્સર ઓપરેશનને નિયંત્રિત કરે છે
- મેમરી મેનેજમેન્ટ: સ્પષ્ટ રીતે ઓપન અને કલોઝ કરવું જોઈએ
- લૂપ પ્રોસેસિંગ: સામાન્ય રીતે અનેક પંક્તિઓ માટે લૂપ્સ સાથે ઉપયોગ થાય છે
- કર્સર એટ્રિબ્યુટ્સ: %FOUND, %NOTFOUND, %ROWCOUNT

મેમરી ટ્રીક

“DOFC - Declare Open Fetch Close”