

# Database Management System (1333204) - Summer 2024

## Solution Gujarati

Milav Dabgar

June 14, 2024

### પ્રશ્ન 1(અ) [3 ગુણ]

વ્યાખ્યા આપો: DBMS, ઈન્સટન્સ, મેટાડેટા

#### જવાબ

- DBMS (ડેટાબેઝ મેનેજમેન્ટ સિસ્ટમ):** એક સોફ્ટવેર જે વપરાશકતાઓને ડેટાબેઝ બનાવવા, જાળવવા, અને એક્સેસ કરવા સક્ષમ બનાવે છે. જે ડેટા ઓર્ગનાઇઝેશન, સ્ટોરેજ, પુન:પ્રાપ્તિ, સુરક્ષા, અને અખંડતાનું નિયંત્રણ કરે છે.
- ઈન્સટન્સ:** કોઈ ચોક્કસ સમયે ડેટાબેઝમાં સંગ્રહિત વાસ્તવિક ડેટા. તે ડેટાબેઝની વર્તમાન સ્થિતિ અથવા સ્નેપશૉટ છે.
- મેટાડેટા:** ડેટા વિશેનો ડેટા, જે ડેટાબેઝ સ્ટ્રક્ચરનું વર્ણન કરે છે, જેમાં ટેબલ્સ, ફિલ્ડ્સ, સંબંધો, કન્સ્ટ્રીન્ટ્સ, અને ઇન્કેક્સનો સમાવેશ થાય છે.

#### મેમરી ટ્રીક

[title=DIM દૃશ્ય]ડેટાબેઝ સિસ્ટમ, ઈન્સટન્સ સ્નેપશૉટ, મેટાડેટા વર્ણન

### પ્રશ્ન 1(બ) [4 ગુણ]

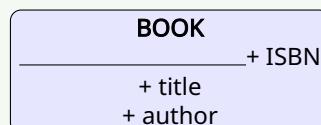
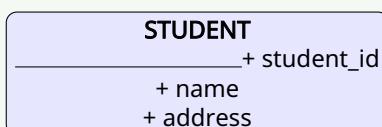
વ્યાખ્યા આપો અને ઉદાહરણ સાથે સમજાવો: 1.Entity 2. Attribute

#### જવાબ

કોષ્ટક: Entity અને Attribute વચ્ચેનો તફાવત

કોન્સોટ	વ્યાખ્યા	ઉદાહરણ
એન્ટિટી	એક વાસ્તવિક દુનિયાની વસ્તુ અથવા ઘાલ જેને સ્પષ્ટપણે ઓળખી શકાય છે	વિદ્યાર્થી (જોન), પુસ્તક (હેરી પોટર), કાર (ટોયોટા કેમરી)
એટ્રિબ્યુટ	એક લક્ષણ અથવા ગુણધર્મ જે એન્ટિટીનું વર્ણન કરે છે	વિદ્યાર્થી: રોલ_નં, નામ, સરનામું પુસ્તક: ISBN, શીર્ષક, લેખક

આકૃતિ:



#### મેમરી ટ્રીક

[title=EA-PC]એન્ટિટીઝ આર ફિઝિકલ/કોન્સેપ્ચ્યુઅલ, એટ્રિબ્યુટ્સ પ્રોવાઇડ કેરેક્ટરિસ્ટિક્સ

## પ્રશ્ન 1(ક) [7 ગુણ]

DBA નું પૂર્ણ નામ લખો. DBAની ભૂમિકા અને જવાબદારીઓ સમજાવો.

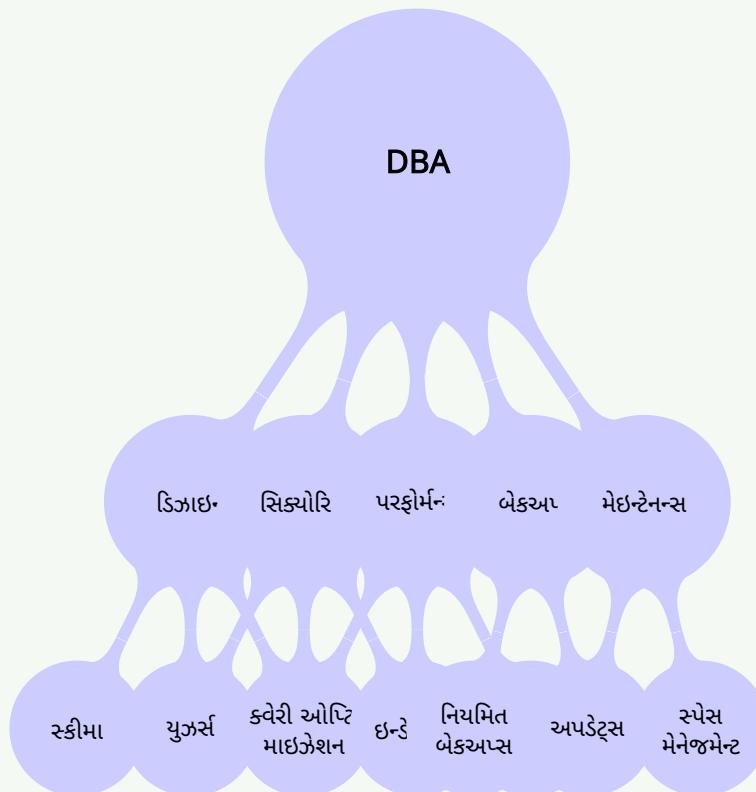
### જવાબ

DBA એટલે ડેટાબેઝ એડમિનિસ્ટ્રેટર.

કોષ્ટક: DBA જવાબદારીઓ

ભૂમિકા	વર્ણન
ડેટાબેઝ ડિઝાઇન	લોજિકલ/ફિઝિકલ ડેટાબેઝ સ્ટ્રક્ચર અને સ્કીમા બનાવે છે
સિક્યુરિટી મેનેજમેન્ટ	યુઝર એકાઉન્ટ્સ અને પરમિશન્સ દ્વારા એક્સેસ નિયંત્રિત કરે છે
પરફોર્મન્સ ટ્યુનિંગ	જડપી ડેટા પુનઃપ્રાપ્તિ માટે કવરીઝ, ઇન્ડેક્સ ઓપ્ટિમાઇઝ કરે છે
બેકઅપ & રિકવરી	ડેટા નુકસાન રોકવા માટેની વ્યૂહરચના અમલમાં મૂકે છે
મેઇન્ટનાન્સ	સોફ્ટવેર અપડેટ કરે છે, પેચિસ લાગુ કરે છે, સ્પેસનું મોનિટરિંગ કરે છે

આકૃતિ:



### મેમરી ટ્રીક

[title=SPMBU]સિક્યુરિટી, પરફોર્મન્સ, મેઇન્ટનાન્સ, બેકઅપ, અપડેટ્સ

## પ્રશ્ન 1(ક) OR [7 ગુણ]

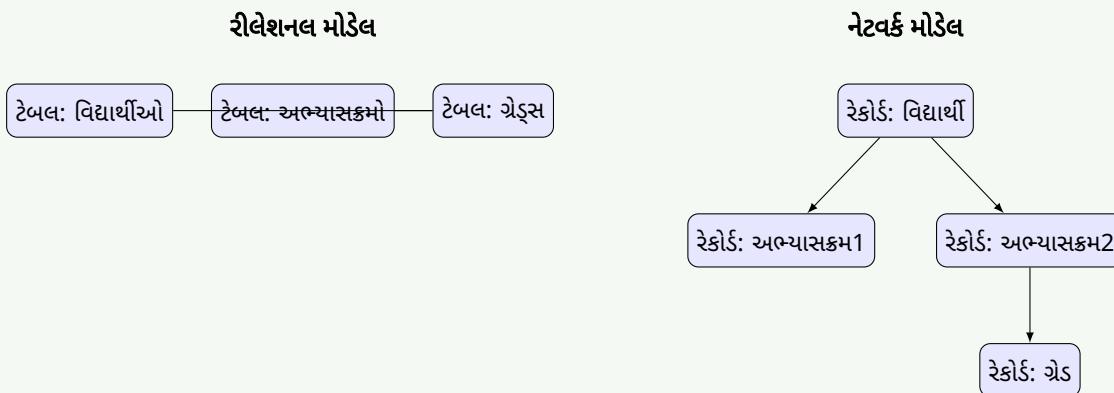
રીલેશનલ અને નેટવર્ક ડેટા મોડેલ વિસ્તારથી સમજાવો.

## જવાબ

કોષ્ટક: રીલેશનલ અને નેટવર્ક ડેટા મોડેલની તુલના

લક્ષણ	રીલેશનલ મોડેલ	નેટવર્ક મોડેલ
સ્ટ્રક્ચર	ટેબલ્સ (રીલેશન્સ) - રો અને કોલમ્સ સાથે	રેકોર્ડ્સ પોઇન્ટર્સ દ્વારા જોડાયેલા જટિલ નેટવર્ક બનાવે છે
સંબંધ	પ્રાઇમરી અને ફોરેન કી દ્વારા જોડાયેલા	પેરટ-ચાઇલડ રેકોર્ડ્સ વચ્ચે ડાયરેક્ટ લિંક્સ
ફોર્માટાબિલિટી	ઉર્ચ - ટેબલ્સ જરૂરિયાત મુજબ જોઈન કરી શકાય છે	સીમિટ - પૂર્વનિર્ધારિત ફિલ્ટર કનેક્શન
ઉદાહરણો	MySQL, Oracle, SQL Server	IDS, IDMS
કવરી લેંગવેજ	SQL (સ્ટ્રક્ચર્ડ કવરી લેંગવેજ)	પ્રોસીજરલ લેંગવેજ

આકૃતિ:



## મેમરી ટ્રીક

[title=RSPEN]રીલેશનલ યુઝિસ સેટ્સ, પોઇન્ટર્સ અને બલ નેટવર્ક્સ

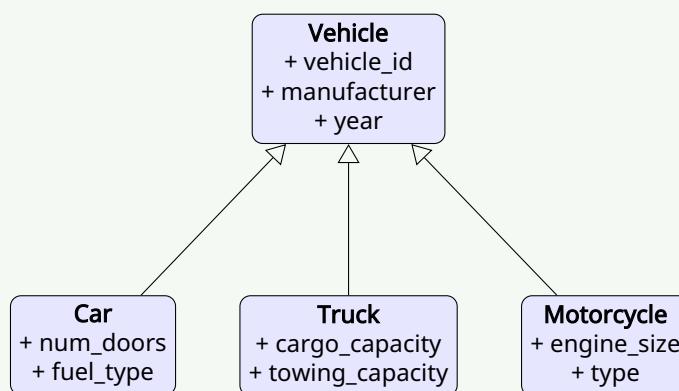
## પ્રશ્ન 2(અ) [3 ગુણ]

Generalization આકૃતિ સાથે સમજાવો.

## જવાબ

Generalization: બે કે વધુ એન્ટિટીઓમાંથી સામાન્ય લક્ષણો કાઢીને નવી ઉર્ચ સ્તરની એન્ટિટી બનાવવાની પ્રક્રિયા.

આકૃતિ:



## મેમરી ટ્રીક

[title=BUSH]બોટમ-અપ શેડ હાયરાર્કી

## પ્રશ્ન 2(બ) [4 ગુણ]

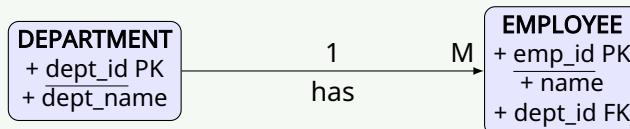
Primary કી અને Foreign કી Constraints સમજાઓ.

## જવાબ

કોષ્ટક: પ્રાઇમરી કી વિ. ફોરેન કી

કન્સ્ટ્રેઇન્ટ	વ્યાખ્યા	ગુણધર્મો	ઉદાહરણ
પ્રાઇમરી કી	ટેબલમાં દરેક રેકૉર્ડને અનન્ય રીતે ઓળખે છે	અનન્ય, નોટ નથી, ટેબલ દીઠ માત્ર એક	વિદ્યાર્થી ટેબલમાં StudentID
ફોરેન કી	ટેબલો વચ્ચે ડેટાને જોડે છે, બીજા ટેબલના પ્રાઇમરી કીનો સંદર્ભ આપે છે	NULL હોઈ શકે, એક ટેબલમાં અનેક હોઈ શકે	અભિયોધી ટેબલમાં DeptID

આકૃતિ:



## મેમરી ટ્રીક

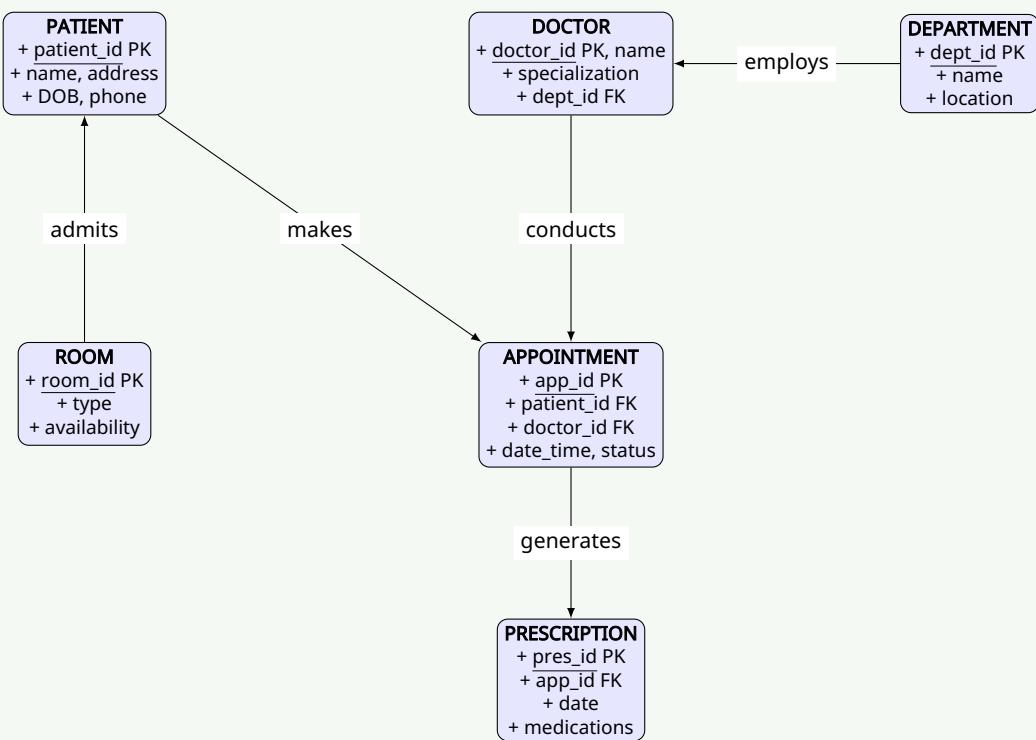
[title=PURE FIRE]પ્રાઇમરી યુનિકલી રેફરન્સિસ એન્ટિટીજ, ફોરેન ઇમ્પોર્ટ્સ રેફર-ન-ડ એન્ટિટીજ

## પ્રશ્ન 2(ક) [7 ગુણ]

હોસ્પિટલ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયાગ્રામ બનાવો

## જવાબ

હોસ્પિટલ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયાગ્રામ:



## મેમરી ટ્રીક

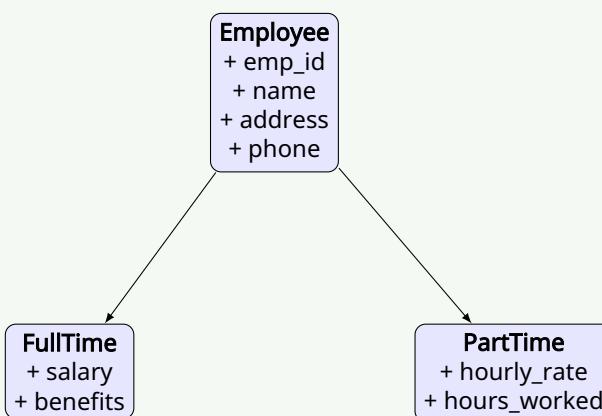
[title=PADRE]પેશાનું અપોઇન્ટમેન્ટ ડોક્ટર રૂમ એન્ટીટીજા

## પ્રશ્ન 2(અ) OR [3 ગુણ]

Specialization આકૃતિ સાથે સમજાવો.

## જવાબ

**Specialization:** હાલની એન્ટીટીમાંથી તેમને અલગ ઓળખવા માટે અન્ય લક્ષણો ઉમેરીને નવી એન્ટીટીઓ બનાવવાની પ્રક્રિયા.  
આકૃતિ:



## મેમરી ટ્રીક

[title=TDSB]ટોપ-ડાઉન સ્પેશલાઇઝડ બ્રેકડાઉન

## પ્રશ્ન 2(બ) OR [4 ગુણ]

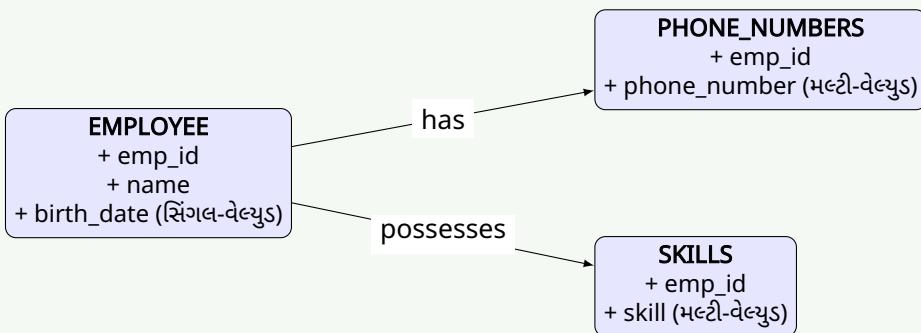
યોગ્ય ઉદાહરણ સાથે સિંગલ વેલ્વુડ અને મલ્ટીવેલ્વુડ એટ્રીબ્યુટ વચ્ચેનો તફાવત સમજાવો.

### જવાબ

કોષ્ટક: સિંગલ-વેલ્વુડ અને મલ્ટી-વેલ્વુડ એટ્રીબ્યુટ્સ

પ્રકાર	વ્યાખ્યા	ઉદાહરણ	ઇમ્પ્લિમેન્ટેશન
સિંગલ-વેલ્વુડ	દરેક એન્ટ્રી ઇન્સ્ટન્સ માટે માત્ર એક જ મૂલ્ય ધરાવે છે	વ્યક્તિની જન-મતારીખ, SSN	સીધા ટેબલ કોલમમાં સંગ્રહિત
મલ્ટી-વેલ્વુડ	એક જ એન્ટ્રી માટે અનેક મૂલ્યો ધરાવી શકે છે	વ્યક્તિની ફુશાળતાઓ, ફોન નંબરો	અલગ ટેબલ અથવા વિશિષ્ટ ફોર્મેટ

આકૃતિ:



### મેમરી ટ્રીક

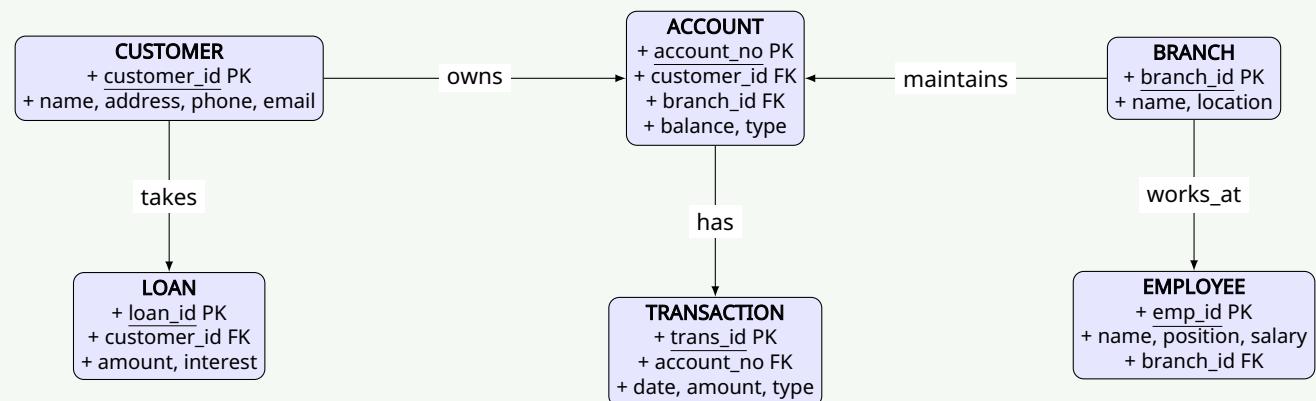
[title=SOME]સિંગલ વન, માન્ટિપલ એન્ટ્રીના

## પ્રશ્ન 2(ક) OR [7 ગુણ]

બેન્કિંગ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયગ્રામ બનાવો

### જવાબ

બેન્કિંગ મેનેજમેન્ટ સિસ્ટમ માટે E-R ડાયગ્રામ:



## મેમરી ટ્રીક

[title=CABLE]કસ્ટમર્સ અકાઉન્ટ્સ બ્રાન્ચિસ લોન્સ એપ્પલોયીજ

## પ્રશ્ન 3(અ) [3 ગુણ]

WHERE અને DESC કલોઝ ઉદાહરણ સાથે સમજાવો.

## જવાબ

કોષ્ટક: WHERE અને DESC કલોઝનો ઉપયોગ

કલોઝ	હેતુ	સિન્ટેક્સ	ઉદાહરણ
WHERE	ચોક્કસ શરત પર આધારિત રો ફિલ્ટર કરે છે	SE... FROM ... WHERE condition	SELECT * FROM employees WHERE salary > 50000
DESC	પરિણામોને ઉત્તરતા કમમાં ગોઠવે છે	SE... ORDER BY ... DESC	SELECT * FROM products ORDER BY price DESC

## Example Data Operation

```

1 -- Original
2 | ID | Name | Marks |
3 | 1 | Alice | 85 |
4 | 2 | Bob | 92 |
5 | 3 | Carol | 78 |
6
7 -- WHERE Marks > 80
8 | 1 | Alice | 85 |
9 | 2 | Bob | 92 |
10
11 -- ORDER BY Marks DESC
12 | 2 | Bob | 92 |
13 | 1 | Alice | 85 |
14 | 3 | Carol | 78 |

```

## મેમરી ટ્રીક

[title=WDF]Where ડેટા ફિલ્ટર કરે છે, DESC ઉચ્ચતમ પહેલા કમ આપે છે

## પ્રશ્ન 3(બ) [4 ગુણ]

DDL કમાન્ડની યાદી બનાવો. કોઈ પણ ૨ DDL કમાન્ડ ઉદાહરણ સાથે સમજાવો.

## જવાબ

DDL (ડેટા ડેફીનિશન લેંગ્વેજ) કમાન્ડ્સ:

1. CREATE
2. ALTER
3. DROP
4. TRUNCATE
5. RENAME

કોષ્ટક: CREATE અને ALTER કમાન્ડ્સ

ક્રમાંક	હેતુ	સિન્ટેક્સ	ઉદાહરણ
CREATE	ટેબલ, વ્યુ, ઇન્ડેક્સ જેવા ડેટાબેઝ ઓફજોક્ટ્સ બનાવે છે	CREATE TABLE ...	CREATE TABLE students (id INT...)
ALTER	હાલના ડેટાબેઝ ઓફજોક્ટની સ્ટ્રક્ચર સુધારે છે	ALTER TABLE ...	ALTER TABLE students ADD COLUMN...

```

1 -- CREATE ઉદાહરણ
2 CREATE TABLE employees (
3   emp_id INT PRIMARY KEY,
4   name VARCHAR(50) NOT NULL,
5   dept VARCHAR(30),
6   salary DECIMAL(10,2)
7 );
8
9 -- ALTER ઉદાહરણ
10 ALTER TABLE employees
11 ADD COLUMN hire_date DATE;

```

### મેમરી ટ્રીક

[title=CADTR]Create Alter Drop Truncate Rename

## પ્રશ્ન 3(ક) [૭ ગુણ]

eno, ename, salary, dept ફિલ્ડ ધરાવતા Company ટેબલ પર નીચેની Query perform કરો.

પ્રશ્નો:

- Company ટેબલના તમામ રેકૉર્ડ ડિસ્પલે કરો.
- ડુપ્લિકેટ વેલ્યુ સિવાય માત્ર dept ડિસ્પલે કરો.
- ename ના ઉત્તરતા કર્મચારીઓનાં નામ ડિસ્પલે કરો.
- શહેરનું નામ સ્ટોર કરવા માટે "cityname" નામથી નવી કોલમ ઉમેરો.
- "Mumbai" શહેરમાં ન રહેતા હોય તેવા તમામ કર્મચારીઓનાં નામ ડિસ્પલે કરો.
- ૧૦૦૦૦ કરતા ઓછું પગાર ધરાવતા તમામ કર્મચારીઓને ડિલીટ કરો.
- "A" થી શરૂ થતા તમામ કર્મચારીઓના નામ ડિસ્પલે કરો.

### જવાબ

```

1 -- 1. Company ટેબલના તમામ રેકૉર્ડ ડિસ્પલે કરો
2 SELECT * FROM Company;
3
4 -- 2. ડુપ્લિકેટ વેલ્યુ સિવાય માત્ર dept ડિસ્પલે કરો
5 SELECT DISTINCT dept FROM Company;
6
7 -- 3. ename ના ઉત્તરતા કર્મચારીઓનાં નામ ડિસ્પલે કરો
8 SELECT * FROM Company ORDER BY ename DESC;
9
10 -- 4. શહેરનું નામ સ્ટોર કરવા માટે "cityname" નામથી નવી કોલમ ઉમેરો
11 ALTER TABLE Company ADD COLUMN cityname VARCHAR(50);
12
13 -- 5. "Mumbai" શહેરમાં ન રહેતા હોય તેવા તમામ કર્મચારીઓનાં નામ ડિસ્પલે કરો
14 SELECT ename FROM Company WHERE cityname != 'Mumbai';
15
16 -- 6. ૧૦૦૦૦ કરતા ઓછું પગાર ધરાવતા તમામ કર્મચારીઓને ડિલીટ કરો
17 DELETE FROM Company WHERE salary < 10000;

```

18  
19     -- એ. "A" થી શરૂ થતા તમામ કર્મચારીઓના નામ ડિસ્પ્લે કરો  
20     SELECT ename FROM Company WHERE ename LIKE 'A%';

### કોષ્ટક: SQL ઓપરેશન્સ

ઓપરેશન	SQL ક્રમાંસ	હેતુ
SELECT	SELECT * FROM Company	બધો ડેટા મેળવે છે
DISTINCT	SELECT DISTINCT dept	ડુલિકેટ દૂર કરે છે
ORDER BY	ORDER BY ename DESC	ઉત્તરતા ક્રમમાં ગોઠવે છે
ALTER	ALTER TABLE ADD COLUMN	નવી કોલમ ઉમેરે છે
WHERE	WHERE cityname != 'Mumbai'	ફિલ્ટર શરત
DELETE	DELETE FROM WHERE	રેકૉર્ડ દૂર કરે છે
LIKE	WHERE ename LIKE 'A%'	પેર્ન મેયિંગ

### મેમરી ટ્રીક

[title=SODA-WDL]Select Order Distinct Alter - Where Delete Like

## પ્રશ્ન 3(અ) OR [3 ગુણ]

SELECT અને DISTINCT કલોઝ ઉદાહરણ સાથે સમજાવો.

### જવાબ

#### કોષ્ટક: SELECT અને DISTINCT કલોઝનો ઉપયોગ

કલોઝ	હેતુ	સિન્કેસ	ઉદાહરણ
SELECT	ડેટાબેઝમાંથી ડેટા મેળવે છે	SELECT columns FROM table	SELECT name, age FROM students
DISTINCT	ડુલિકેટ મૂલ્યો દૂર કરે છે	SELECT DISTINCT columns FROM table	SELECT DISTINCT department FROM employees

```

1 -- Original: Sales, IT, HR, IT, Sales
2
3 -- SELECT dept_name
4 Sales
5 IT
6 HR
7 IT
8 Sales
9
10 -- SELECT DISTINCT dept_name
11 Sales
12 IT
13 HR

```

### મેમરી ટ્રીક

[title=SUD]Select Unique with Distinct

## પ્રશ્ન 3(બ) OR [4 ગુણ]

DML કમાન્ડની યાદી બનાવો. કોઈ પણ ૨ DML કમાન્ડ ઉદાહરણ સાથે સમજાવો.

### જવાબ

**DML (દેશા મેન્યુલેશન લોગેજ) કમાન્ડ્સ:**

1. INSERT
2. UPDATE
3. DELETE
4. SELECT

**કોષ્ટક: INSERT અને UPDATE કમાન્ડ્સ**

કમાન્ડ	હેતુ	સિન્ક્રેસન	ઉદાહરણ
INSERT	ટેબલમાં નવા રેકૉર્ડ ઉમેરે છે	INSERT INTO ... VALUES	INSERT INTO students VALUES (1, 'John', 85)
UPDATE	હાલના રેકૉર્ડમાં ફેરફાર કરે છે	UPDATE ... SET ... WHERE	UPDATE students SET marks=90 WHERE id=1

```

1 -- INSERT ઉદાહરણ
2 INSERT INTO employees (emp_id, name, dept, salary)
3   VALUES (101, 'John Smith', 'IT', 65000);
4
5 -- UPDATE ઉદાહરણ
6 UPDATE employees
7   SET salary = 70000
8 WHERE emp_id = 101;

```

### મેમરી ટ્રીક

[title=IUDS]Insert Update Delete Select

## પ્રશ્ન 3(ક) OR [7 ગુણ]

નીચેની Query ના આઉટપુટ લખો.

### જવાબ

**કોષ્ટક: SQL ફંક્શન આઉટપુટ**

ફંક્શન	વર્ણન	આઉટપુટ
ABS(-34), ABS(16)	નિરપેક્ષ મૂલ્ય	34, 16
SQRT(16), SQRT(64)	વર્ગમૂળ	4, 8
POWER(5,2), POWER(2,4)	પાવર ફંક્શન	25, 16
MOD(15,3), MOD(13,3)	મોડ્યુલસ (બાકી)	0, 1
ROUND(123.456,1)	દશાંશ સ્થાન સુધી રાઉન્ડ	123.5
ROUND(123.456,2)	(૨ દશાંશ)	123.46
CEIL(122.6)	ઉપર રાઉન્ડ	123
CEIL(-122.6)	(નેગેટિવ)	-122
FLOOR(-157.5)	નીચે રાઉન્ડ	-158
FLOOR(157.5)	(પોઝિટિવ)	157

## મેમરી ટ્રીક

[title=ASPRCF]Absolute Square Power Remainder Ceiling Floor

## પ્રશ્ન 4(અ) [૩ ગુણ]

SQLમાં ડેટા ટાઈપની યાદી બનાવો. 1.VARCHAR() અને 2.INT() ડેટા ટાઈપ ઉદાહરણ સાથે સમજાવો.

## જવાબ

SQL ડેટા ટાઈપ કેટેગરીઓ:

1. ન્યુમેરિક (INT, FLOAT, DECIMAL)
2. કેરેક્ટર (CHAR, VARCHAR)
3. ડેટ/ટાઈમ (DATE, TIME, DATETIME)
4. બાઇનરી (BLOB, BINARY)
5. બૂલિયન (BOOL)

કોષ્ટક: VARCHAR અને INT ડેટા ટાઈપ્સ

ડેટા ટાઈપ	વર્ણન	સાઈઝ	ઉદાહરણ
VARCHAR(n)	વેરિએબલ-લેન્થ કેરેક્ટર સ્ટ્રિંગ	n કેરેક્ટર સુધી	નામ, ઈમેલ માટે VARCHAR(50)
INT	ઇન્ટિજર ન્યુમેરિક ડેટા	સામાન્ય રીતે 4 બાઈટ્સ	ID, કાઉન્ટ માટે INT

```

1 CREATE TABLE students (
2   student_id INT PRIMARY KEY,
3   name VARCHAR(50) NOT NULL,
4   age INT,
5   email VARCHAR(100)
6 );

```

## મેમરી ટ્રીક

[title=VIA]Variable strings, Integers for Ages

## પ્રશ્ન 4(બ) [૪ ગુણ]

2NF (સેક્ન્ડ નોર્મલ ફોર્મ) ઉદાહરણ અને ઉકેલ સાથે સમજાવો.

### જવાબ

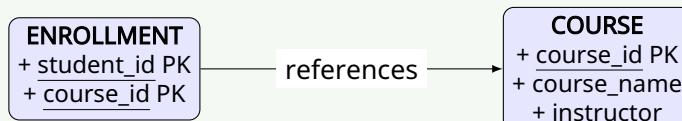
**2NF વ્યાખ્યા:** એક સંબંધ 2NF માં છે જો તે 1NF માં હોય અને કોઈપણ નોન-પ્રાઇમ એટ્રિબ્યુટ કોઈપણ કેન્ડિક્ટ કીના સબસેટ પર આધારિત ન હોય.

**કોષ્ટક:** 2NF પહેલાં

student_id	course_id	course_name	instructor
S1	C1	Database	Prof. Smith
S1	C2	Networking	Prof. Jones
S2	C1	Database	Prof. Smith
S3	C3	Programming	Prof. Wilson

**સમસ્યા:** નોન-પ્રાઇમ એટ્રિબ્યુટ્સ (course\_name, instructor) માત્ર course\_id પર આધારિત છે, સંપૂર્ણ કી (student\_id, course\_id) પર નહીં.

**આફ્ટિસ:** 2NF ઉકેલ



**કોષ્ટક:** 2NF પછી

Enrollment ટેબલ:

student_id	course_id
S1	C1
S1	C2
S2	C1
S3	C3

Course ટેબલ:

course_id	course_name	instructor
C1	Database	Prof. Smith
C2	Networking	Prof. Jones
C3	Programming	Prof. Wilson

### મેમરી ટ્રીક

[title=PFPK]Partial Functional dependency on Primary Key

## પ્રશ્ન 4(ક) [૭ ગુણા]

Function dependency સમજાવો. Partial function dependency ઉદાહરણ સાથે સમજાવો.

### જવાબ

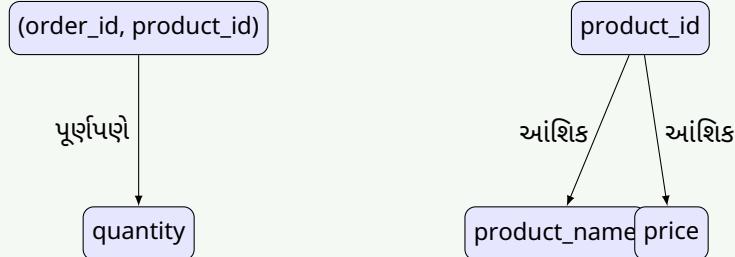
**Functional Dependency:** એટ્રિબ્યુટ્સ વર્ચનો સંબંધ જ્યાં એક એટ્રિબ્યુટનું મૂલ્ય બીજા એટ્રિબ્યુટના મૂલ્યને નક્કી કરે છે.  
નોટેશન:  $X \rightarrow Y$  ( $X$   $Y$  ને નક્કી કરે છે)

**Partial Functional Dependency:** જ્યારે નોન-પ્રાઇમ એટ્રિબ્યુટ કંપોઝિટ કીના સંપૂર્ણ કરતાં ભાગ પર આધારિત હોય.  
કોષ્ટક: Order Details (નોર્મલાઇઝેશન પહેલાં)

order_id	product_id	quantity	product_name	price
O1	P1	5	Keyboard	50
O1	P2	2	Mouse	25
O2	P1	1	Keyboard	50
O3	P3	3	Monitor	200

**Functional Dependencies:**

- $(\text{order\_id}, \text{product\_id}) \rightarrow \text{quantity}$
- $\text{product\_id} \rightarrow \text{product\_name}$
- $\text{product\_id} \rightarrow \text{price}$

**આકૃતિ:****ઉક્ત નોર્મલાઈઝ ટેબલ્સ:**

Orders ટેબલ:

order_id	product_id	quantity
O1	P1	5
O1	P2	2
O2	P1	1
O3	P3	3

Products ટેબલ:

product_id	product_name	price
P1	Keyboard	50
P2	Mouse	25
P3	Monitor	200

**મેમરી ટ્રીક**

[title=PDPK]Partial Dependency on Part of Key

**પ્રશ્ન 4(અ) OR [3 ગુણ]**

કમાન્ડ સમજાવવો: 1) To\_Char() 2) To\_Date()

**જવાબ**

કોષ્ટક: કન્વર્ન ફંક્શન્સ

ફંક્શન	હેતુ	સિન્ટેક્સ	ઉદાહરણ
TO_CHAR()	ડેટ/નંબરને કેરેક્ટર સ્ટ્રિંગમાં રૂપાંતરિત કરે છે	TO_CHAR(val, fmt)	TO_CHAR(SYSDATE, 'DD-MON')
TO_DATE()	કેરેક્ટર સ્ટ્રિંગને ડેટમાં રૂપાંતરિત કરે છે	TO_DATE(str, fmt)	TO_DATE('14-JUN', 'DD-MON')

1 `SELECT TO_CHAR(SYSDATE, 'DD-MON-YYYY') FROM DUAL;`  
 2 `SELECT TO_DATE('2024-06-14', 'YYYY-MM-DD') FROM DUAL;`

### મેમરી ટ્રીક

[title=DCS]Date Conversion Strings

## પ્રશ્ન 4(વ) OR [4 ગુણ]

Full function dependency ઉદાહરણ સાથે સમજાવો.

### જવાબ

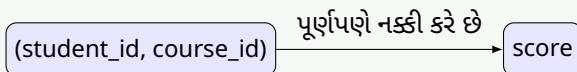
**Full Functional Dependency:** જ્યારે એક એટ્રિબ્યુટ કંપોઝિટ કી પર ફંક્શનલી ડિપેન્ડન્ટ હોય, અને માત્ર ભાગ પર નહીં પણ સંપૂર્ણ કી પર આધારિત હોય.

**કોષ્ટક:** Exam Results

student_id	course_id	exam_date	score
S1	C1	2024-05-10	85
S1	C2	2024-05-15	92
S2	C1	2024-05-10	78
S2	C2	2024-05-15	88

**Full Functional Dependency:**

- $(\text{student\_id}, \text{course\_id}) \rightarrow \text{score}$  (સ્કોર વિદ્યાર્થી અને કોર્સ બંને પર આધારિત છે)
- આકૃતિ:



**સમજૂતી:** સ્કોર એટ્રિબ્યુટ સંપૂર્ણ રીતે કંપોઝિટ કી ( $\text{student\_id}, \text{course\_id}$ ) પર આધારિત છે કારણ કે:

- અલગ અલગ વિદ્યાર્થીઓના એક જ કોર્સ માટે અલગ અલગ સ્કોર હોઈ શકે છે
- એક જ વિદ્યાર્થીના અલગ અલગ કોર્સ માટે અલગ અલગ સ્કોર હોઈ શકે છે
- ચોક્કસ સ્કોર જાણવા માટે આપણાને  $\text{student\_id}$  અને  $\text{course\_id}$  બનેની જરૂર પડે છે

### મેમરી ટ્રીક

[title=FCEK]Fully dependent on Complete/Entire Key

## પ્રશ્ન 4(ક) OR [7 ગુણ]

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

## જવાબ

**નોર્મલાઇઝેશન:** ડેટા રિડન્સી ઘટાડવા, ડેટા અખંડતા સુધારવા અને એનોમલીઓને દૂર કરવા માટે મોટા ટેબલને નાના સંબંધિત ટેબલમાં વિભાજિત કરીને ડેટાને વ્યવસ્થિત કરવાની પ્રક્રિયા.

**1NF વ્યાખ્યા:** એક સંબંધ 1NF માં છે જો તેના બધા એટ્રિબ્યુટ્સ માત્ર અવિભાજ્ય (એટોમિક) મૂલ્યો ધરાવતા હોય.

**કોષ્ટક:** 1NF પહેલાં

student_id	name	courses
S1	John	Math, Physics
S2	Mary	Chemistry, Biology, Physics
S3	Tim	Computer Science

**સમસ્યાઓ:**

- નોન-એટોમિક મૂલ્યો (એક સેલમાં અનેક કોર્સી)
- ચોક્કસ કોર્સને કવેરી કે અપડેટ કરવું સરળ નથી

**આકૃતિ:**



**કોષ્ટક:** 1NF પછી

student_id	name	course
S1	John	Math
S1	John	Physics
S2	Mary	Chemistry
S2	Mary	Biology
S2	Mary	Physics
S3	Tim	Computer Science

## મેમરી ટ્રીક

[title=ASAV]Atomic Single-value Attributes only Valid

## પ્રશ્ન 5(અ) [3 ગુણ]

Transaction નો concept ઉદાહરણ સાથે સમજાવો.

## જવાબ

**Transaction:** એક લોજિકલ કાર્ય એકમ જે સંપૂર્ણપણે અમલમાં મૂકવામાં આવે અથવા સંપૂર્ણપણે રદ કરવામાં આવે.

**કોષ્ટક:** Transaction ગુણધર્મો

ગુણધર્મ	વર્ણન
Atomicity	બધા ઓપરેશન સફળતાપૂર્વક પૂર્ણ થાય અથવા કોઈ નહીં
Consistency	સુસંગત સ્થિતિમાં રહે
Isolation	એકબીજામાં દખલ ન કરે
Durability	ફેરફાર ટકી રહે

1 BEGIN TRANSACTION;

```

2   -- અકાઉન્ટ A માંથી 500 કાઢવા
3   UPDATE accounts SET balance = balance - 500 WHERE account_id = 'A';
4   -- અકાઉન્ટ B માં 500 ઉમેરવા
5   UPDATE accounts SET balance = balance + 500 WHERE account_id = 'B';
6   COMMIT;

```

## મેમરી ટ્રીક

[title=ACID]Atomicity Consistency Isolation Durability

## પ્રશ્ન 5(બ) [4 ગુણ]

equi join સિન્ટેક્સ અને ઉદાહરણ સાથે સમજાવો.

### જવાબ

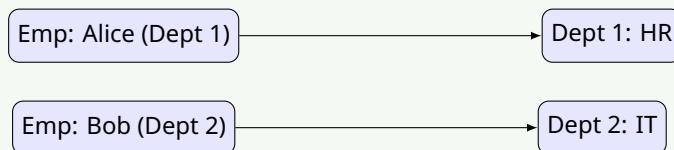
**Equi Join:** સમાનતા તુલના ઓપરેટરનો ઉપયોગ કરે છે.

```

1   SELECT e.name, d.dept_name
2   FROM employees e, departments d
3   WHERE e.dept_id = d.dept_id;

```

આફ્ટરિટી:



## મેમરી ટ્રીક

[title=MEET]Match Equal Elements Every Table

## પ્રશ્ન 5(ક) [7 ગુણ]

Conflict serializability વિસ્તારથી સમજાવો.

### જવાબ

**Conflict Serializability:** સમાંતર ટ્રોન્કેશનની સાચી કાર્યપ્રણાલી સુનિશ્ચિત કરવાની એક રીત, જે એ ગેરંટી આપે છે કે એક્ઝિક્યુશન શેડ્યુલ કોઈ સીરિયલ એક્ઝિક્યુશનના સમકક્ષ છે.

**કોષ્ટક:** Conflict Serializability ના મુખ્ય ઘાલો

ઘાલ	વર્ણન
Conflicting Operations	બે ઓપરેશન કોન્ફિલક્ટ કરે છે જો તેઓ એક જ ડેટા આઇટમ એક્સેસ કરે અને ઓછામાં ઓછું એક રાઇટ હોય
Precedence Graph	સંધર્ષો દર્શાવતો ડાયરેક્ટેડ ગ્રાફ
Conflict Serializable	શેડ્યૂલ conflict serializable છે જો તેનો precedence graph એસાઇક્લિક હોય

આફ્ટરિટી:



$T_1 \xrightarrow{\text{Serializable}} T_2$

$T_4 \xrightarrow{\text{Circular}} T_5$

સાયકલ (Serializable નથી)

**ઉદાહરણ:** ટ્રાન્ઝેક્શન  $T_1$  અને  $T_2$  ધ્યાનમાં લો:

- $T_1$ : Read(A), Write(A)
- $T_2$ : Read(A), Write(A)

શેડ્યુલ S1:  $R_1(A), W_1(A), R_2(A), W_2(A)$  - Serializable ( $T_1 \rightarrow T_2$  સમકક્ષ) શેડ્યુલ S2:  $R_1(A), R_2(A), W_1(A), W_2(A)$  - Not serializable (precedence ગ્રાફમાં સાયકલ છે)

**Conflict Serializability નક્કી કરવાના પગલાં:**

1. બધા કોન્ફિલિક્ટિંગ ઓપરેશન જોડીઓ ઓળખો
2. precedence ગ્રાફ બનાવો
3. ચેક કરો કે ગ્રાફમાં સાયકલ છે કે નહીં
4. જો સાયકલ ન હોય, તો શેડ્યુલ conflict serializable છે

### મેમરી ટ્રીક

[title=COPS]Conflicts, Operations, Precedence, Serializability

## પ્રશ્ન 5(અ) OR [3 ગુણ]

Transaction નાં ગુણધર્મો ઉદાહરણ સાથે સમજાવો.

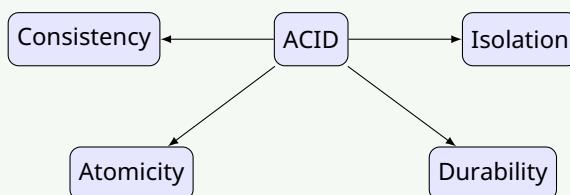
### જવાબ

ટ્રાન્ઝેક્શનના ACID ગુણધર્મો:

કોઈક: ACID ગુણધર્મો

ગુણધર્મ	વર્ણન	ઉદાહરણ
Atomicity	બધા ઓપરેશન સફળતાપૂર્વક પૂર્ણ થાય અથવા કોઈ નહીં	બેંક ટ્રાન્સફર - ડેબિટ અને કેર્ડિટ બંને એકસાથે સફળ થવા જોઈએ અથવા નિષ્ફળ થવા જોઈએ
Consistency	ટ્રાન્ઝેક્શન પહેલાં અને પછી ડેટાબેઝ સુસંગત રિસ્ટિમાં રહે	\$100 ટ્રાન્સફર કર્યા પછી, સિસ્ટમમાં કુલ પૈસા અપરિવર્તિત રહે
Isolation	સમાંતર ટ્રાન્ઝેક્શન એકબીજામાં દખલ ન કરે	ટ્રાન્ઝેક્શન A ટ્રાન્ઝેક્શન B ના આંશિક પરિણામો જોતું નથી
Durability	એકવાર કમિટ થયા પછી, ફેરફારો કાયમી છે	પાવર ફેલ્યોર પણ કમિટેડ ટ્રાન્ઝેક્શનને ખોવાતું નથી

આકૃતિ:



ઉદાહરણ:

```

1 -- ATM Withdrawal ટ્રાન્ઝેક્શન
2 BEGIN TRANSACTION;
3   -- બેલેન્સ ચેક કરો
4   SELECT balance FROM accounts WHERE account_id = 'A123';
5
6   -- જો પૂર્તું હોય, તો બેલેન્સ અપડેટ કરો
7   UPDATE accounts SET balance = balance - 100 WHERE account_id = 'A123';
8
9   -- ઉપાડની નોંધ કરો
10  INSERT INTO transactions (account_id, type, amount, date)
11    VALUES ('A123', 'WITHDRAWAL', 100, SYSDATE);
12
13   -- જો બધા ઓપરેશન સફળ હોય તો
14   COMMIT;
15   -- જો કોઈ ઓપરેશન નષ્ટિકુળ જાય તો
16   -- ROLLBACK;
17 END TRANSACTION;

```

## મેમરી ટ્રીક

[title=ACID]Atomicity Consistency Isolation Durability

## પ્રશ્ન 5(બ) OR [4 ગુણ]

ઉપર Q.5 (b) માં આપેલ "Faculty" અને "CT" ટેબલનો ઉપયોગ કરીને...

### જવાબ

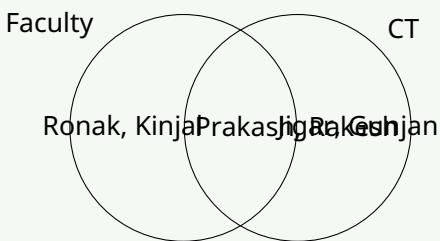
#### Queries:

1. Faculty અથવા CT (UNION)
2. Faculty અને CT (INTERSECT)
3. માત્ર Faculty (MINUS)
4. માત્ર CT (MINUS)

```

1 SELECT FacultyName FROM Faculty UNION SELECT CTName FROM CT;
2 SELECT FacultyName FROM Faculty INTERSECT SELECT CTName FROM CT;
3 SELECT FacultyName FROM Faculty MINUS SELECT CTName FROM CT;
4 SELECT CTName FROM CT MINUS SELECT FacultyName FROM Faculty;

```



## મેમરી ટ્રીક

[title=UIMM]Union Intersect Minus Minus

## પ્રશ્ન 5(ક) OR [7 ગુણ]

View serializability વિસ્તારથી સમજાવો.

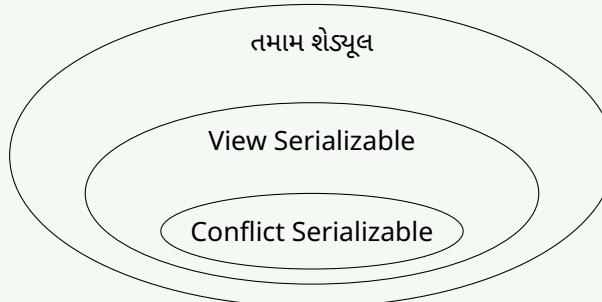
### જવાબ

**View Serializability:** એક શેડ્યુલ view serializable છે જો તે કોઈ સીરિયલ શેડ્યૂલના view equivalent હોય, એટલે કે તે ડેટાબેઝની એક જ દૃશ્ય (અથવા અંતિમ સ્થિતી) ઉત્પન્ન કરે.

**કોષ્ટક:** Conflict Serializability સાથે તુલના

પાસું	View Serializability	Conflict Serializability
વ્યાપ્તા	રીડ અને રાઇટના અંતિમ પરિણામો પર આધારિત	ઓપરેશન વર્ચ્યોના કોન્ફિલક્ટ પર આધારિત
શરત	પ્રારંભિક રીડ, અંતિમ લખાણ, અને રીડ-રાઇટ ડિપેન્ડન્સી જાળવે છે	ઓપરેશન વર્ચ્યોના બધા કોન્ફિલક્ટ જાળવે છે
સ્કોપ	શેડ્યૂલનો વ્યાપક વર્ગ	view serializable શેડ્યૂલનો સબસેટ
ટેરિંગ	પરીક્ષણ વધુ જટિલ	precedence ગ્રાફ વડે ટેસ્ટ કરી શકાય

આકૃતિ:



### View Equivalence શરતો:

- પ્રારંભિક રીડ: જો T1 શેડ્યૂલ S1 માં ડેટા આઇટમ A ની પ્રારંભિક વેલ્યુ વાંચે છે, તો તેણે S2 માં પણ પ્રારંભિક વેલ્યુ વાંચવી જોઈએ.
- અંતિમ રાઇટ: જો T1 શેડ્યૂલ S1 માં ડેટા આઇટમ A પર અંતિમ લખાણ કરે છે, તો તેણે S2 માં પણ અંતિમ લખાણ કરવું જોઈએ.
- રીડ-રાઇટ ડિપેન્ડન્સી: જો T1 શેડ્યૂલ S1 માં T2 દ્વારા લખાયેલ A ની વેલ્યુ વાંચે છે, તો તેણે S2 માં પણ T2 દ્વારા લખાયેલ વેલ્યુ વાંચવી જોઈએ.

ઉદાહરણ - View Serializable પરંતુ Conflict Serializable નહીં: બ્લાઇન્ડ રાઇટ (વાંચયા વિના લખાણ) ધરાવતા ટ્રાન્કેશન ધ્યાનમાં લો:

- T1: W1(A)
- T2: W2(A)

શેડ્યૂલ S: W1(A), W2(A) - T1 → T2 અને T2 → T1 બંને માટે view serializable છે (અંતિમ લખાણ હંમેશા T2 દ્વારા થાય છે) પરંતુ W1(A) અને W2(A) કોન્ફિલક્ટ કરે છે, એટલે કોન્ફિલક્ટ ગ્રાફમાં બંને દિશામાં એજ હશે.

### મેમરી ટ્રીક

[title=IRF]Initial reads, Result writes, Final view