

# Subject Name (Gujarati)

1333204 -- Winter 2024

Semester 1 Study Material

*Detailed Solutions and Explanations*

## પ્રશ્ન 1(અ) [3 માકર્સ]

ફિલ્ડ, રેકૉર્ડ, મેટાડટા ને વ્યાખ્યાયિત કરો.

### જવાબ

- ફિલ્ડ: એન્ટીના એક એટ્રિબ્યુટને રજૂ કરતો ડેટાનો એક એકમ
- રેકૉર્ડ: એન્ટી વિશે ડેટા સંગ્રહિત કરતા સંબંધિત ફિલ્ડ્સનો સમૂહ
- મેટાડટા: ડેટા વિશેની માહિતી જે ડેટાબેઝ ઓફજેક્ટ્સની સંરચના, ગુણાધર્મો અને સંબંધોનું વર્ણન કરે છે

### મેમરી ટ્રીક

"FRaMe" (ફિલ્ડ, રેકૉર્ડ, મેટાડટા)

## પ્રશ્ન 1(બ) [4 માકર્સ]

સ્ટ્રોગ અને વીક entity set ને વ્યાખ્યાયિત કરો.

### જવાબ

એન્ટી પ્રકાર	વર્ણન	ઓળખ	ઉદાહરણ
સ્ટ્રોગ એન્ટી	સ્વતંત્ર રીતે અસ્તિત્વમાં છે	તેની પોતાની પ્રાથમિક કી ધરાવે છે	ગ્રાહક, કર્મચારી
વીક એન્ટી	સ્ટ્રોગ એન્ટી પર આધાર રાખે છે	પેરેન્ટ એન્ટી કી જરૂરી છે	બેંક એકાઉન્ટ, ઓર્ડર આઈટમ

### મેમરી ટ્રીક

"SWing" (Strong is With own identity, weak is Not Getting own identity)

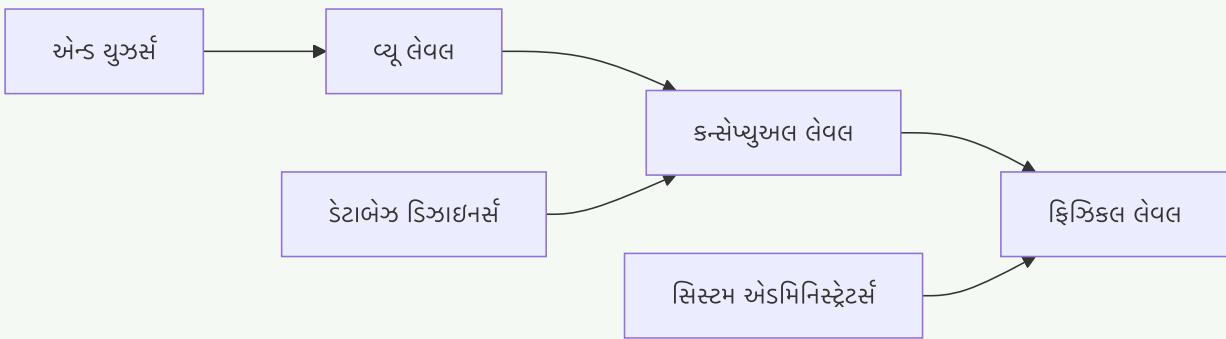
## પ્રશ્ન 1(ક) [7 માકર્સ]

ડેટા ઓફ્સ્ક્રેનના 3 સ્તરો સમજાવો.

### જવાબ

સ્તર	વર્ણન	વપરાશકર્તા
ફિલ્ડ લેવલ	ડેટા ભौતિક રીતે કેવી રીતે સંગ્રહિત થાય છે તે વર્ણવે છે	સિસ્ટમ એડમિનિસ્ટ્રેટર્સ
કન્સેપ્ચુઅલ લેવલ	કયો ડેટા સંગ્રહિત થયેલો છે અને સંબંધોનું વર્ણન કરે છે	ડેટાબેઝ ડિઝાઇનર્સ
વ્યૂ લેવલ	વપરાશકર્તાઓ માટે પ્રસ્તુત ડેટાબેઝનો ભાગ વર્ણવે છે	એન્ડ યુઝર્સ

### ડાયગ્રામ:



### મેમરી ટ્રીક

“PCV” (Physical, Conceptual, View - નીચેથી ઉપર)

### પ્રશ્ન 1(ક) OR [7 માકર્સ]

DBMS ના ફાયદાઓ અને ગેરફાયદાઓ સમજાવો.

#### જવાબ

ફાયદાઓ	ગેરફાયદાઓ
ડેટા રીડન્ડન્સી કંટ્રોલ	સોફ્ટવેર અને હાર્ડવેરની ઊંચી કિમત
ડેટા કન્સિસ્ટન્સી	ડિઝાઇન અને જાળવણીમાં જટિલતા
બહેતર ડેટા સિક્યુરિટી	ભારે ઉપયોગ સાથે પફ્ફાર્ન-સ પર અસર
ડેટા શેરિંગ	સિસ્ટમ નિષ્ફળતાઓથી સંવેદનશીલતા
ડેટા ઇન્ડિપેન્ન્સ	નિષ્ફળતા પછી રિકવરી ચેલેન્જ્યુસ
પ્રમાણભૂત એક્સેસ	વધારેલ તાલીમ આવશ્યકતાઓ

### મેમરી ટ્રીક

“BASIC-DV” (Benefits: Access, Security, Independence, Consistency - Drawbacks: Vulnerability)

### પ્રશ્ન 2(અ) [3 માકર્સ]

રિલેશનલ એલેજેબ્લા નું સિલેક્ટ ઓપરેશન સમજાવો.

#### જવાબ

સિલેક્ટ ઓપરેશન (થ)	વર્ણન
સિન્ટેક્સ	થ(Relation)
કાર્ય	શરત સંતોષતા ટપદ્સ મેળવે છે
ઉદાહરણ	થsalary>30000(Employee)

### મેમરી ટ્રીક

“SERVe” (Select Exactly Required Values)

### પ્રશ્ન 2(બ) [4 માકર્સ]

DBMS માં પ્રાઇમરી, ફોરેઇન, સુપર, કેન્ડીકેટ કી ની વ્યાખ્યા આપો.

## જવાબ

કી પ્રકાર	વર્ણન
પ્રાઇમરી કી	દરેક રેકૉર્ડ માટે યુનિક ઓળખકર્તા
ફોરેઇન કી	અન્ય ટેબલમાં પ્રાઇમરી કી સાથે જોડાતું એટ્રિબ્યુટ
સુપર કી	એટ્રિબ્યુટ્સનો સેટ જે રેકૉર્ડ્સને યુનિક રીતે ઓળખી શકે છે
કેન્ડિક્ટ કી	મિનિમલ સુપર કી જે પ્રાઇમરી કી બની શકે છે

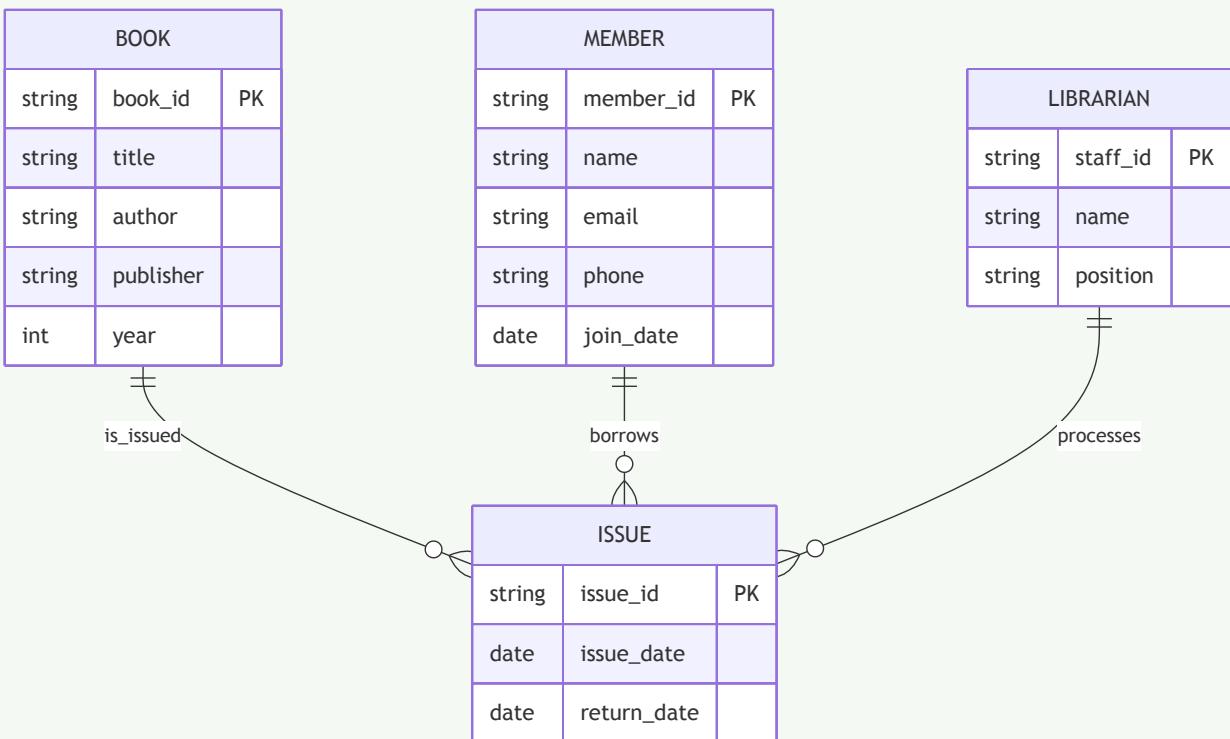
## મેમરી ટ્રીક

“PFSC” (Person First Shows Credentials)

## પ્રશ્ન 2(ક) [7 માક્સ્]

Library Management System નો E R Diagram દોરો.

## જવાબ



## મેમરી ટ્રીક

“LIMB” (Library Items, Members, Borrowing)

## પ્રશ્ન 2(અ) OR [3 માક્સ્]

રિલેશનલ એન્જેચા નું યુનિયન ઓપરેશન સમજાવો.

## જવાબ

$$\begin{array}{c}
 \text{યુનિયન ઓપરેશન (U)} \quad \text{વર્ણન} \\
 \hline
 \text{સિન્ક્રેસ} \quad \text{Relation1} \cup \text{Relation2}
 \end{array}$$

કાર્ય  
આવશ્યકતા

બંને સંબંધોમાંથી ટપલ્સ એકત્રિત કરે છે  
બંને સંબંધો યુનિયન-સંગત હોવા જોઈએ

ઉદાહરણ: Students\_CS  $\cup$  Students\_IT

મેમરી ટ્રીક

“CUP” (Combining Union of Parts)

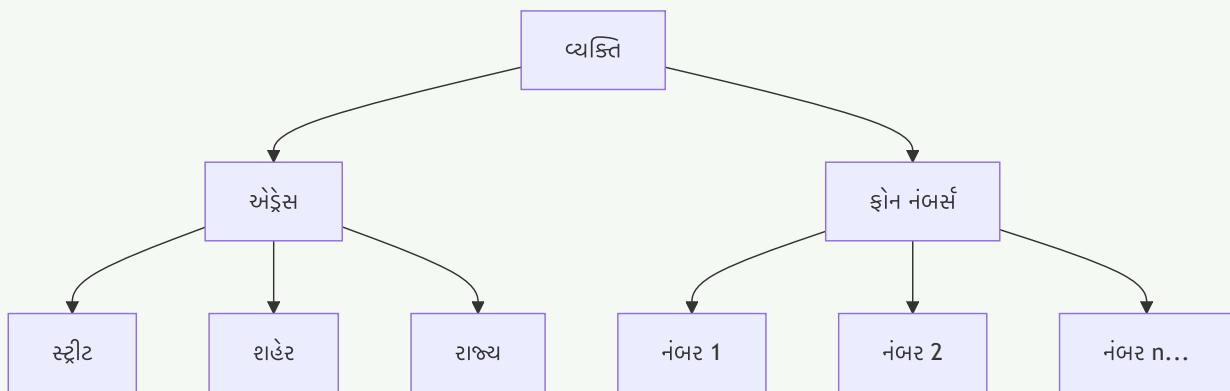
### પ્રશ્ન 2(બ) OR [4 માક્સ]

ઉદાહરણ સાથે કંપોઝિટ એટ્રિબ્યુટ અને મલ્ટીવેલ્યુડ એટ્રિબ્યુટ ને વ્યાખ્યાયિત કરો.

જવાબ

એટ્રિબ્યુટ પ્રકાર	વર્ણન	ઉદાહરણ
કંપોઝિટ	નાના સબપાર્ટ્સમાં વિભાજિત થઈ શકે છે	એડ્રેસ (સ્ટ્રીટ, શહેર, રાજ્ય, પિન)
મલ્ટીવેલ્યુડ	એક કરતાં વધુ મૂલ્ય ધરાવી શકે છે	ફોન નંબર્સ, ઈમેલ એડ્રેસિસ

ડાયાગ્રામ:



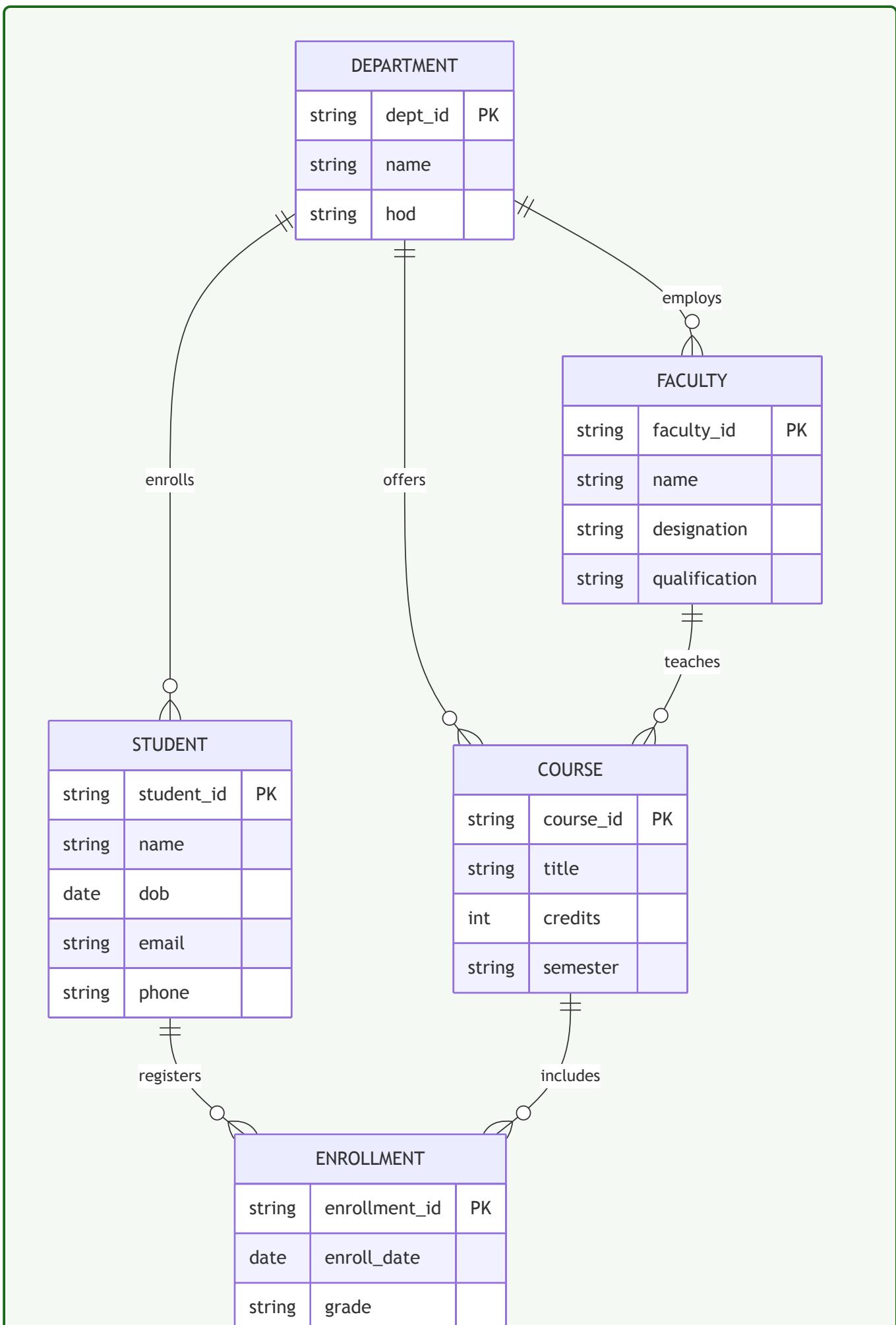
મેમરી ટ્રીક

“CoMbo” (Composite has Multiple components)

### પ્રશ્ન 2(ક) OR [7 માક્સ]

College Management System નો E R Diagram દોરો.





## મેમરી ટ્રીક

“DECFS” (Departments, Enrollments, Courses, Faculty, Students)

### પ્રશ્ન 3(અ) [3 માફર્સ]

SQL માં વિવિધ ડેટા ટાઈપ્સ ની યાદી બનાવો અને ટુંક માં સમજાવો

#### જવાબ

ડેટા ટાઈપ કેટેગરી	ઉદાહરણો	ઉપયોગ
નિયમેરિક	INT, FLOAT, DECIMAL	સંખ્યાઓ સંગ્રહ કરવા
ક્રેક્ટર	CHAR, VARCHAR, TEXT	ટેક્સ્ટ સંગ્રહ કરવા
ડેટ/ટાઈમ	DATE, TIME, TIMESTAMP	સમય સંબંધિત ડેટા સંગ્રહ કરવા
બૂલિયન	BOOLEAN	સાચા/ખોટા મૂલ્યો સંગ્રહ કરવા
બાઇનરી	BLOB, BINARY	બાઇનરી ડેટા સંગ્રહ કરવા

## મેમરી ટ્રીક

“NCDBB” (Numbers, Characters, Dates, Booleans, Binaries)

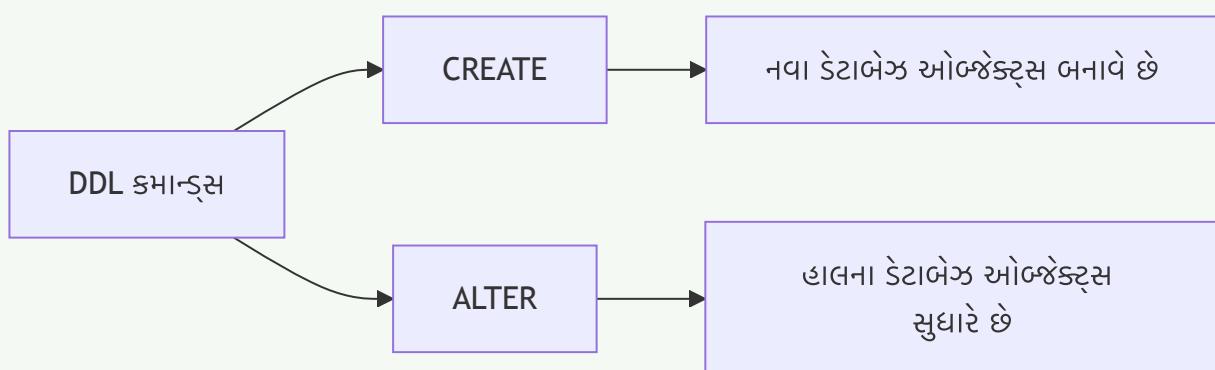
### પ્રશ્ન 3(બ) [4 માફર્સ]

કોઈ પણ બે DDL કમાન્ડ્સ સિંટેક્ષ અને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

કમાન્ડ	સિન્ટેક્ષ	ઉદાહરણ
CREATE	CREATE TABLE table_name (column_definitions);	CREATE TABLE Student (id INT PRIMARY KEY, name VARCHAR(50));
ALTER	ALTER TABLE table_name ADD/DROP/MODIFY column_name data_type;	ALTER TABLE Student ADD email VARCHAR(100);

#### ડાયાગ્રામ:



## મેમરી ટ્રીક

“CAD” (Create And Define)

### પ્રશ્ન 3(ક) [7 માંકર્સ]

નીચે ની કવરી નું આઉટપુટ લખો. a. CEIL(123.57), CEIL(4.1) b. MOD(12,4), MOD(10,4) c. POWER(2,3), POWER(3,3) d. ROUND(121.413,1), ROUND(121.413,2) e. FLOOR(25.3), FLOOR(25.7) f. LENGTH('AHMEDABAD') g. ABS(-25), ABS(36)

#### જવાબ

ફંક્શન	પરિણામ	સમજૂતી
CEIL(123.57)	124	123.57 થી મોટી કે સમાન સૌથી નાની પૂર્ણ સંખ્યા
CEIL(4.1)	5	4.1 થી મોટી કે સમાન સૌથી નાની પૂર્ણ સંખ્યા
MOD(12,4)	0	12 ÷ 4
MOD(10,4)	2	10 ÷ 4
POWER(2,3)	8	2 ને 3 ની ઘાત
POWER(3,3)	27	3 ને 3 ની ઘાત
ROUND(121.413,1)	121.4	1 દશાંશ સ્થાન સુધી રાઉન્ડ
ROUND(121.413,2)	121.41	2 દશાંશ સ્થાન સુધી રાઉન્ડ
FLOOR(25.3)	25	25.3 થી નાની કે સમાન સૌથી મોટી પૂર્ણ સંખ્યા
FLOOR(25.7)	25	25.7 થી નાની કે સમાન સૌથી મોટી પૂર્ણ સંખ્યા
LENGTH('AHMEDABAD')	9	અક્ષરોની સંખ્યા
ABS(-25)	25	-25 ની નિરપેક્ષ કિંમત
ABS(36)	36	36 ની નિરપેક્ષ કિંમત

#### મેમરી ટ્રીક

“CMPRFLA” (Ceiling, Modulus, Power, Round, Floor, Length, Absolute)

### પ્રશ્ન 3(અ) OR [3 માંકર્સ]

કોઈ પણ ત્રાણ ડેટ ફંક્શન સમજાવો.

#### જવાબ

ડેટ ફંક્શન	હેતુ	ઉદાહરણ	પરિણામ
ADD_MONTHS	તારીખમાં મહિના ઉમેરે છે	ADD_MONTHS('01-JAN-2023', 3)	01-APR-2023
MONTHS_BETWEEN	બે તારીખો વર્ચ્યેના મહિના ગણે છે	MONTHS_BETWEEN('01-2-MAR-2023', '01-JAN-2023')	
SYSDATE	વર્તમાન તારીખ અને સમય આપે છે	SYSDATE	વર્તમાન સિસ્ટમ તારીખ/સમય

#### મેમરી ટ્રીક

“AMS” (Add\_months, Months\_between, Sysdate)

### પ્રશ્ન 3(બ) OR [4 માંકર્સ]

કોઈ પણ બે DML કમાન્ડ સિંટેક્ષન અને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

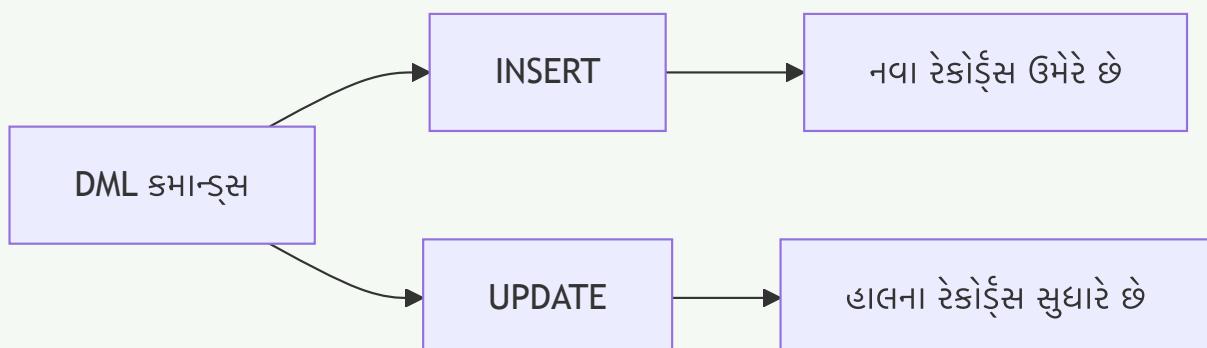
કમાન્ડ	સિંટેક્ષન	ઉદાહરણ
INSERT	INSERT INTO table_name VALUES (value1, value2,...);	INSERT INTO Student VALUES (1, 'Raj', 'raj@example.com');

**UPDATE**

UPDATE table\_name SET  
column=value WHERE  
condition;

UPDATE Student SET  
email='new@example.com'  
WHERE id=1;

ડાયાગ્રામ:



મેમરી ટ્રીક

“IUM” (Insert, Update, Manipulate)

### પ્રશ્ન 3(ક) OR [7 માંકર્સ]

ટેબલ EMP(emp\_no, emp\_name, designation, salary, deptno) ને ધ્યાને લઈ ને નીચે આપેલા operations માટે SQL commands લખો.

જવાબ

ઓપરેશન

SQL કમાન્ડ

EMP ટેબલ ને ફિલે કરો

CREATE TABLE EMP (emp\_no INT PRIMARY KEY,

emp\_name VARCHAR(50), designation VARCHAR(30),

salary DECIMAL(10,2), deptno INT);

SELECT emp\_no, emp\_name, designation, salary,

deptno FROM EMP;

SELECT \* FROM EMP WHERE emp\_name LIKE `p%`;

emp\_no, emp\_name, designation, salary,

deptno ને EMP ને આપો

જેમના નામ 'p' થી શરૂ થતાં હોય તેવા બધા એમણ્ણોથી ની

માહિતી દરખાસ્તો

Department wise salary total દરખાસ્તો

SELECT deptno, SUM(salary) AS total\_salary FROM

EMP GROUP BY deptno;

ALTER TABLE EMP ADD email\_id VARCHAR(100);

ALTER TABLE EMP RENAME COLUMN designation TO

post;

DELETE FROM person;

EMP table માં નવી કોલમ email\_id ઉમેરો

કોલમ નામ "designation" ને "post" તરીકે બદલાવો

टેબલ person ના તમામ records delete કરો

મેમરી ટ્રીક

“CSDAACD” (Create, Select, Display, Aggregate, Add, Change, Delete)

### પ્રશ્ન 4(અ) [3 માંકર્સ]

વિવિધ aggregate functions ની યાદી બનાવો અને કોઈ એક ને syntax અને ઉદાહરણ સાથે સમજાવો.

## જવાબ

એગ્રીગેટ ફંક્શન	હેતુ
SUM	કુલ ગણતરી કરે છે
AVG	સરેરાશ ગણતરી કરે છે
COUNT	રો ની સંખ્યા ગણે છે
MAX	મહત્તમ મૂલ્ય શોધે છે
MIN	લઘુત્તમ મૂલ્ય શોધે છે

### AVG માટે ઉદાહરણ:

AVG(column\\_name) - કોલમમાં મૂલ્યોની સરેરાશ ગણતરી કરે છે  
SELECT AVG(salary) FROM Employee; - સરેરાશ પગાર આપે છે

## મેમરી ટ્રીક

“SCAMM” (Sum, Count, Avg, Max, Min)

## પ્રશ્ન 4(બ) [4 માક્સ]

ટ્રાન્સેક્શન ને ઉદાહરણ સાથે વ્યાખ્યાયિત કરો.

## જવાબ

ટ્રાન્સેક્શન કન્સેપ્ટ	વર્ણન
વ્યાખ્યા	કાર્યનો તાર્કિક એકમ જે સંપૂર્ણપણે પ્રોસેસ થવો જોઈએ અથવા સંપૂર્ણપણે નિષ્ફળ જવો જોઈએ
ગુણધર્મો	ACID (એટોમિસિટી, કન્સિસ્ટન્સી, આઈસોલેશન, ડ્યુરેબિલિટી)
સ્થિતિઓ	એક્ટિવ, પાર્શ્વયલી કમિટેડ, કમિટેડ, ફેલ, એબોર્ટ
ઉદાહરણ:	

```

1 BEGIN TRANSACTION;
2   UPDATE Accounts SET balance = balance - 5000 WHERE acc_no = 'A123';
3   UPDATE Accounts SET balance = balance + 5000 WHERE acc_no = 'B456';
4 COMMIT;

```

## મેમરી ટ્રીક

“TAPS” (Transaction As Process Set)

## પ્રશ્ન 4(ક) [7 માક્સ]

SQL માં ઓપરેટર શું છે? એરિથમેટિક અને લોજિકલ ઓપરેટર ઉદાહરણ સાથે સમજાવો.

## જવાબ

પ્રકાર	ઓપરેટર્સ	ઉદાહરણ	પરિણામ
એરિથમેટિક	+ (ઉમેરો) - (બાદબાકી) * (ગુણાકાર) / (ભાગાકાર) % (મોડ્યુલસ)	5 + 3 5 - 3 5 * 3 15 / 3 5 % 2	8 2 15 5 1

લોજિકલ	AND	salary > 30000 AND dept = 'IT'	બંને શરતો સાચી હોય તો સાચું
	OR	salary > 50000 OR dept = 'HR'	કોઈપણ એક શરત સાચી હોય તો સાચું
	NOT	NOT (salary < 20000)	જો પગાર 20000 થી ઓછો ન હોય તો સાચું

### SQL ઉદાહરણો:

```

1 -- 
2 SELECT product_name, price * 1.18 AS price_with_tax FROM Products;
3 --
4 --
5 SELECT * FROM Employees WHERE (salary > 30000 AND dept = 'IT') OR (experience > 5);

```

### મેમરી ટ્રીક

“ASMDOLA” (Add, Subtract, Multiply, Divide, OR, AND, NOT)

### પ્રશ્ન 4(અ) OR [3 માકર્સ]

વિવિધ numeric functions ની યાદી બનાવો અને કોઈ એક ને syntax અને ઉદાહરણ સાથે સમજાવો.

#### જવાબ

ન્યૂમેરિક ફંક્શન	હેતુ
ROUND	સંખ્યાને નિર્દિષ્ટ દશાંશ સ્થાનો સુધી રાઉન્ડ કરે છે
TRUNC	સંખ્યાને નિર્દિષ્ટ દશાંશ સ્થાનો સુધી ટ્રંક્ટ કરે છે
CEIL	સંખ્યાથી મોટી કે સમાન સૌથી નાની પૂર્ણ સંખ્યા આપે છે
FLOOR	સંખ્યાથી નાની કે સમાન સૌથી મોટી પૂર્ણ સંખ્યા આપે છે
ABS	નિરપેક્ષ મૂલ્ય આપે છે

#### ROUND માટે ઉદાહરણ:

ROUND(number, decimal\\_places) - સંખ્યાને નિર્દિષ્ટ દશાંશ સ્થાનો સુધી રાઉન્ડ કરે છે

SELECT ROUND(125.679, 2) FROM DUAL; - 125.68 આપે છે

### મેમરી ટ્રીક

“RTCF” (Round, Truncate, Ceiling, Floor, Absolute)

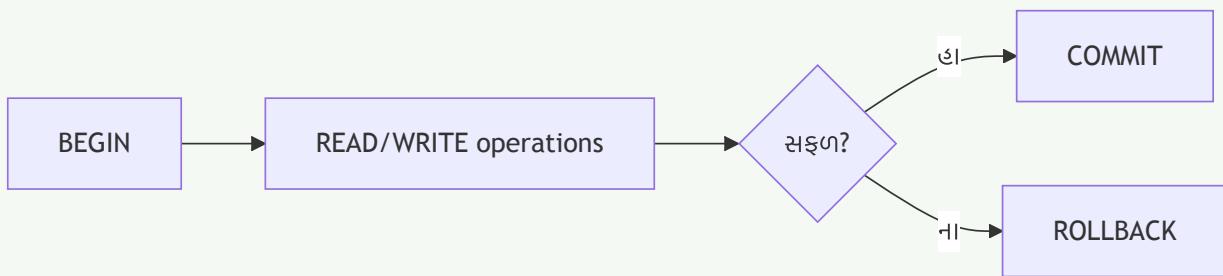
### પ્રશ્ન 4(બ) OR [4 માકર્સ]

ટ્રાન્સેક્શન માટે વિવિધ database operations ની યાદી બનાવો.

#### જવાબ

ઓપરેશન	વર્ણન
BEGIN/START	ટ્રાન્સેક્શન શરૂઆત બિંદુ ચિહ્નિત કરે છે
READ	ડેટાબેઝમાંથી ડેટા મેળવે છે
WRITE	ડેટાબેઝમાં ડેટા સુધારે છે
COMMIT	ફેરફારો કાયમી બનાવે છે
ROLLBACK	ફેરફારો રદ કરે છે અને પ્રારંભિક બિંદુ પર પાછા ફરાર કરે છે
SAVEPOINT	અંશિક રૂપે પાછા ફરવા માટે બિંદુઓ બનાવે છે

### ડાયાગ્રામ:



### મેમરી ટ્રીક

“BRWCRS” (Begin, Read, Write, Commit, Rollback, Savepoint)

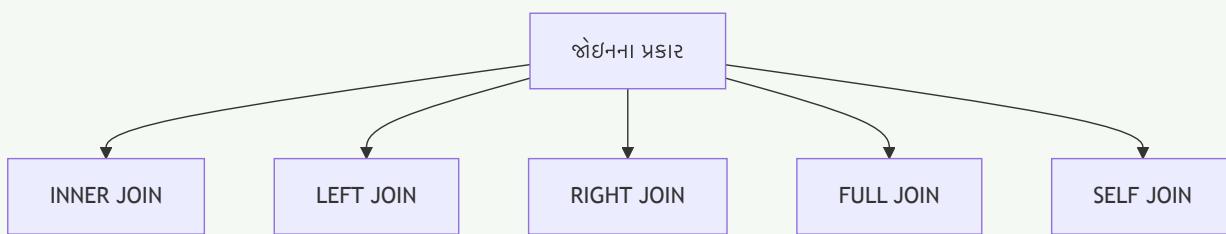
### પ્રશ્ન 4(ક) OR [7 માંકર્સ]

જોઇન શું છે? વિવિધ પ્રકાર ના જોઇન ને syntax અને ઉદાહરણ સાથે સમજાવો.

### જવાબ

જોઇન પ્રકાર	વર્ણન	સિન્ટેક્સ ઉદાહરણ
INNER JOIN	બંને ટેબલમાં મેચ હોય ત્યારે રો આપે છે	SELECT * FROM TableA INNER JOIN TableB ON TableA.id = TableB.id;
LEFT JOIN	ડાબા ટેબલના બધા રો અને જમણા ટેબલના મેચ થતા રો આપે છે	SELECT * FROM TableA LEFT JOIN TableB ON TableA.id = TableB.id;
RIGHT JOIN	જમણા ટેબલના બધા રો અને ડાબા ટેબલના મેચ થતા રો આપે છે	SELECT * FROM TableA RIGHT JOIN TableB ON TableA.id = TableB.id;
FULL JOIN	કોઈપણ એક ટેબલમાં મેચ હોય ત્યારે રો આપે છે	SELECT * FROM TableA FULL JOIN TableB ON TableA.id = TableB.id;
SELF JOIN	ટેબલને તેની જાત સાથે જોડે છે	SELECT * FROM Employee e1 JOIN Employee e2 ON e1.manager_id = e2.emp_id;

### ડાયાગ્રામ:



### મેમરી ટ્રીક

“ILRFS” (Inner, Left, Right, Full, Self)

### પ્રશ્ન 5(અ) [3 માંકર્સ]

નીચે આપેલા customer relation ને 1NF માં બદલાવો.

Customer

cid	name	address	Contact_no
CO1	Riya	Amu aavas, Anand	{5322332123}
CO2	Jiya	Sardar colony, Ahmedabad	{5326521456, 5265232849}

### જવાબ

Customer Table (1NF):

cid	name	society	city	Contact_no
CO1	Riya	Amu aavas	Anand	5322332123
CO2	Jiya	Sardar colony	Ahmedabad	5326521456
CO2	Jiya	Sardar colony	Ahmedabad	5265232849

### મેમરી ટ્રીક

“AFM” (Atomic values, Flatten Multivalued attributes)

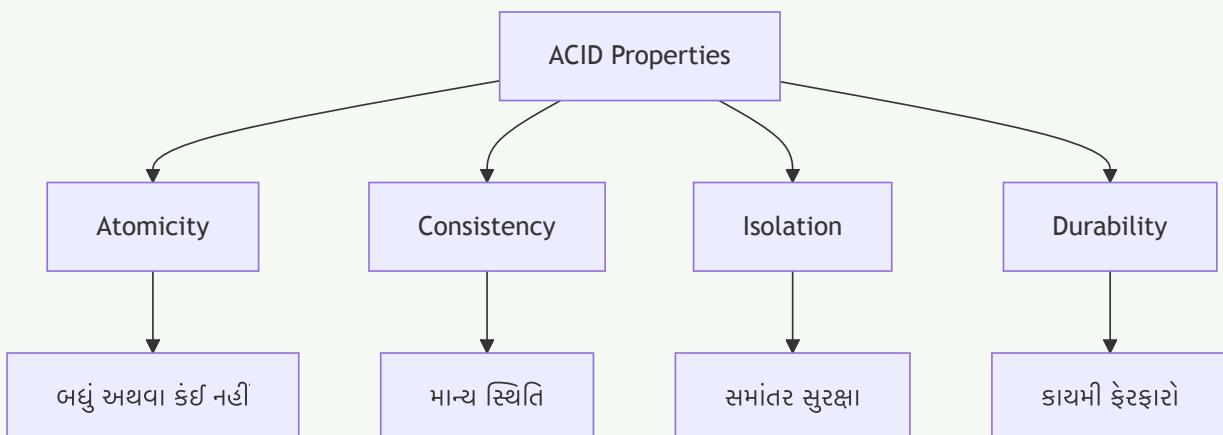
### પ્રશ્ન 5(બ) [4 માંકર્સ]

ટ્રાન્સેક્શન ની ACID properties ની યાદી બનાવો અને સમજાવો.

### જવાબ

ACID Property	વર્ણન
Atomicity	ટ્રાન્સેક્શન સંપૂર્ણપણે ચાલે છે અથવા બિલકુલ નહીં
Consistency	ડેટાબેઝ ટ્રાન્સેક્શન પહેલાં અને પછી સુસંગત રહે છે
Isolation	સમાંતર ટ્રાન્સેક્શન એકબીજા સાથે દખલ કરતા નથી
Durability	કમિટેડ ફરફારો સિસ્ટમ નિષ્ફળતા પછી પણ કાયમી રહે છે

### ડાયાગ્રામ:



### મેમરી ટ્રીક

“ACID” (Atomicity, Consistency, Isolation, Durability)

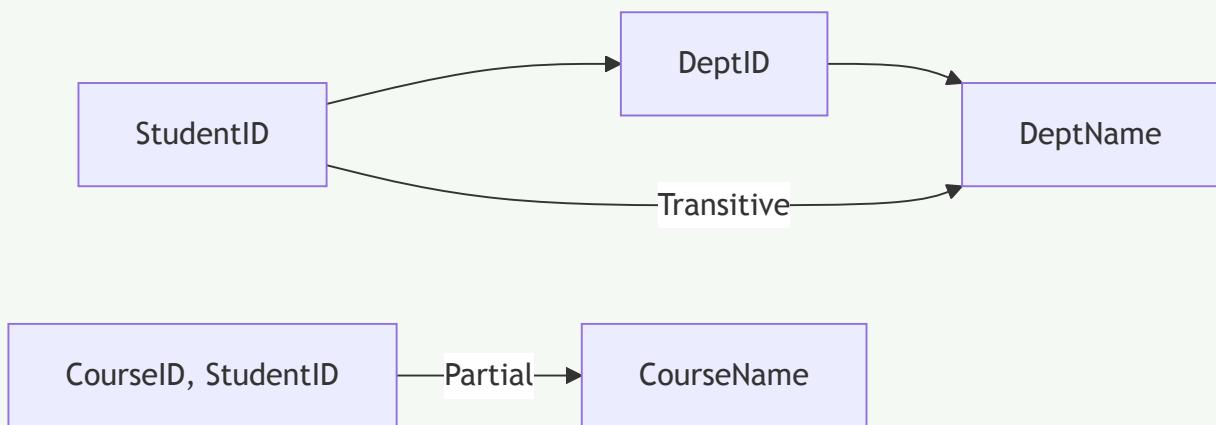
### પ્રશ્ન 5(ક) [7 માંકર્સ]

વિવિધ functional dependencies ની યાદી બનાવો અને દરેક ને ઉદાહરણ સાથે સમજાવો.

## જવાબ

Functional Dependency	વર્ણન	ઉદાહરણ
Trivial FD	$X \rightarrow YYX$	{StudentID, Name} $\rightarrow \{Name\}$
Non-trivial FD	$X \rightarrow YYX$	{StudentID} $\rightarrow \{Name\}$
Partial FD	કમોઝિટ કી નો ભાગ નોન-કી એટ્રિબ્યુટ નક્કી કરે છે	{CourseID, StudentID} $\rightarrow \{CourseName\}$
Transitive FD	$X \rightarrow YY \rightarrow ZX \rightarrow Z$	{StudentID} $\rightarrow \{DeptID\}$ {DeptID} $\rightarrow \{DeptName\}$
Multivalued FD	એક એટ્રિબ્યુટ બીજા એટ્રિબ્યુટના મૂલ્યોનો સેટ નક્કી કરે છે	{CourseID} $\rightarrow \rightarrow \{TextbookID\}$

## ડાયગ્રામ:



## મેમરી ટ્રીક

“TNPTMV” (Trivial, Non-trivial, Partial, Transitive, Multivalued)

## પ્રશ્ન 5(અ) OR [3 માંકસ]

નીચે આપેલા Depositor\_Account relation ને 2NF માં બદલાવો. જ્યાં functional dependencies(FD) નીચે મુજબ છે. FD1: {cid, ano}  $\rightarrow \{access\_date, balance, bname\}$

## જવાબ

### Account Table (2NF):

ano	balance	bname
-----	---------	-------

### Depositor Table (2NF):

cid	ano	access_date
-----	-----	-------------

## મેમરી ટ્રીક

“RPKD” (Remove Partial Key Dependencies)

## પ્રશ્ન 5(બ) OR [4 માક્સ]

Conflict serializability સમજવો.

જવાબ	
<p>કન્સેપ્ટ વ્યાખ્યા કન્ફિલક્ટ ઓપરેશન્સ કન્ફિલક્ટ ગ્રાફ ટેરિસ્ટિંગ</p> <p>ડાયાગ્રામ:</p> <pre> graph LR     T1[Transaction T1] -- "Write X then Read X" --&gt; T2[Transaction T2]     T2 -- "Write Y then Read Y" --&gt; T3[Transaction T3]     T3 -- "Read X and Y" --&gt; Conclusion[Serializable Transaction]     style Conclusion fill:#e0e0ff     Conclusion --- Note["કોઈ એક નથી - સીરિયલાઇઝેબલ"]   </pre>	<p>વર્ણન</p> <p>સિરિયલ શેડ્યુલ સાથે સમકક્ષ હોય તો શેડ્યુલ કન્ફિલક્ટ સીરિયલાઇઝેબલ છે એક જ ડેટા આઇટમ પર રીડ-રાઇટ, રાઇટ-રીડ, રાઇટ-રાઇટ ઓપરેશન્સ ટ્રાન્સેક્શન વર્ચેના કન્ફિલક્ટ દર્શાવતો ડાયરેક્ટેડ ગ્રાફ જો કન્ફિલક્ટ ગ્રાફમાં ચક ન હોય તો શેડ્યુલ કન્ફિલક્ટ સીરિયલાઇઝેબલ છે</p>

### મેમરી ટ્રીક

“COGS” (Conflict Operations Graph Serializable)

## પ્રશ્ન 5(ક) OR [7 માક્સ]

ઉદાહરણ સાથે 3NF normalization સમજવો.

જવાબ												
<table border="1"> <thead> <tr> <th>Normal Form</th> <th>વ્યાખ્યા</th> <th>ઉદાહરણ</th> </tr> </thead> <tbody> <tr> <td><b>1NF</b></td> <td>એટોમિક વેલ્યુ, કોઈ રિપીટિંગ ગ્રુપ નહીં</td> <td>Student(ID, Name, Phone1, Phone2) → <i>Student(ID, Name, Phone)</i></td> </tr> <tr> <td><b>2NF</b></td> <td>1NF + કોઈ પાર્શ્વયલ ડિપેન્ડન્સી નહીં</td> <td>Order(OrderID, ProductID, CustomerID, ProductName) → <i>Order(OrderID, ProductID, CustomerID, ProductName)</i></td> </tr> <tr> <td><b>3NF</b></td> <td>2NF + કોઈ ટ્રાન્ઝિટિવ ડિપેન્ડન્સી નહીં</td> <td>Student(ID, DeptID, DeptName) → <i>Student(ID, DeptID)</i> + <i>Department(DeptID, DeptName)</i></td> </tr> </tbody> </table>	Normal Form	વ્યાખ્યા	ઉદાહરણ	<b>1NF</b>	એટોમિક વેલ્યુ, કોઈ રિપીટિંગ ગ્રુપ નહીં	Student(ID, Name, Phone1, Phone2) → <i>Student(ID, Name, Phone)</i>	<b>2NF</b>	1NF + કોઈ પાર્શ્વયલ ડિપેન્ડન્સી નહીં	Order(OrderID, ProductID, CustomerID, ProductName) → <i>Order(OrderID, ProductID, CustomerID, ProductName)</i>	<b>3NF</b>	2NF + કોઈ ટ્રાન્ઝિટિવ ડિપેન્ડન્સી નહીં	Student(ID, DeptID, DeptName) → <i>Student(ID, DeptID)</i> + <i>Department(DeptID, DeptName)</i>
Normal Form	વ્યાખ્યા	ઉદાહરણ										
<b>1NF</b>	એટોમિક વેલ્યુ, કોઈ રિપીટિંગ ગ્રુપ નહીં	Student(ID, Name, Phone1, Phone2) → <i>Student(ID, Name, Phone)</i>										
<b>2NF</b>	1NF + કોઈ પાર્શ્વયલ ડિપેન્ડન્સી નહીં	Order(OrderID, ProductID, CustomerID, ProductName) → <i>Order(OrderID, ProductID, CustomerID, ProductName)</i>										
<b>3NF</b>	2NF + કોઈ ટ્રાન્ઝિટિવ ડિપેન્ડન્સી નહીં	Student(ID, DeptID, DeptName) → <i>Student(ID, DeptID)</i> + <i>Department(DeptID, DeptName)</i>										
<b>ઉલ્લંઘન ઉદાહરણ:</b>												
1 Employee(EmpID, EmpName, DeptID, DeptName, Location)												
<b>3NF રૂપાંતરણ:</b>												
1 Employee(EmpID, EmpName, DeptID) 2 Department(DeptID, DeptName, Location)												
<b>ડાયાગ્રામ:</b>												

