

# Subject Name Solutions

1333204 – Winter 2024

Semester 1 Study Material

*Detailed Solutions and Explanations*

## Question 1(a) [3 marks]

Define: Field, Record, Metadata

### Solution

- **Field:** A single unit of data representing one attribute of an entity
- **Record:** Collection of related fields that store data about an entity
- **Metadata:** Data about data that describes the structure, properties, and relationships of database objects

### Mnemonic

“FRaMe” (Field, Record, Metadata)

## Question 1(b) [4 marks]

Define: strong and weak entity set.

### Solution

Entity Type	Description	Identification	Example
<b>Strong Entity</b>	Exists independently	Has its own primary key	Customer, Employee
<b>Weak Entity</b>	Depends on strong entity	Requires parent entity key	Bank Account, Order Item

### Mnemonic

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

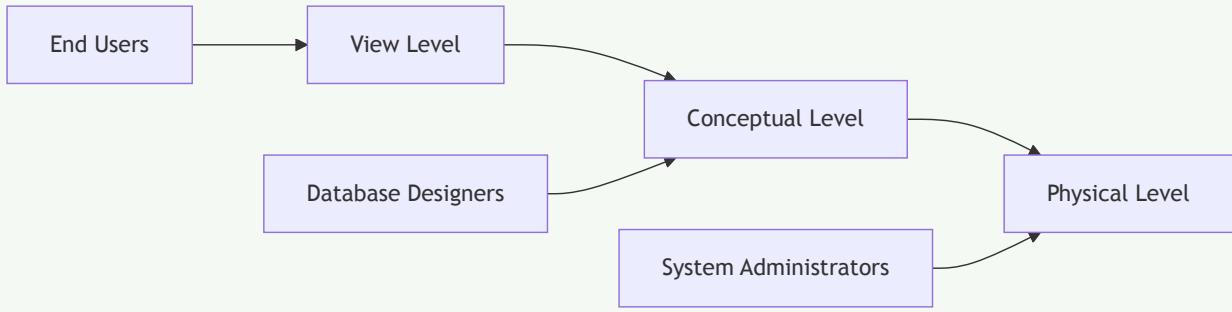
## Question 1(c) [7 marks]

Explain 3 Levels of Data Abstraction

### Solution

Level	Description	Used By
<b>Physical Level</b>	Describes how data is stored physically	System Administrators
<b>Conceptual Level</b>	Describes what data is stored and relationships	Database Designers
<b>View Level</b>	Describes part of database relevant to users	End Users

Diagram:



Mnemonic

“PCV” (Physical, Conceptual, View - bottom to top)

### Question 1(c) OR [7 marks]

Explain advantages and disadvantages of DBMS.

Solution

Advantages	Disadvantages
<b>Data Redundancy Control</b>	<b>High Cost</b> of software and hardware
<b>Data Consistency</b>	<b>Complexity</b> in design and maintenance
<b>Improved Data Security</b>	<b>Performance Impact</b> with heavy usage
<b>Data Sharing</b>	<b>Vulnerability</b> to system failures
<b>Data Independence</b>	<b>Recovery Challenges</b> after failure
<b>Standardized Access</b>	<b>Increased Training Requirements</b>

Mnemonic

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

### Question 2(a) [3 marks]

Explain select operation in relational algebra with example

Solution

Select Operation ( )	Description
<b>Syntax</b>	(Relation)
<b>Function</b>	Retrieves tuples satisfying condition
<b>Example</b>	$\text{salary} > 30000(\text{Employee})$

Mnemonic

“SERVe” (Select Exactly Required Values)

### Question 2(b) [4 marks]

Define Primary, Foreign, Super, Candidate Keys in DBMS.

## Solution

Key Type	Description
<b>Primary Key</b>	Unique identifier for each record
<b>Foreign Key</b>	Attribute linking to primary key in another table
<b>Super Key</b>	Set of attributes that can uniquely identify records
<b>Candidate Key</b>	Minimal super key that can be primary key

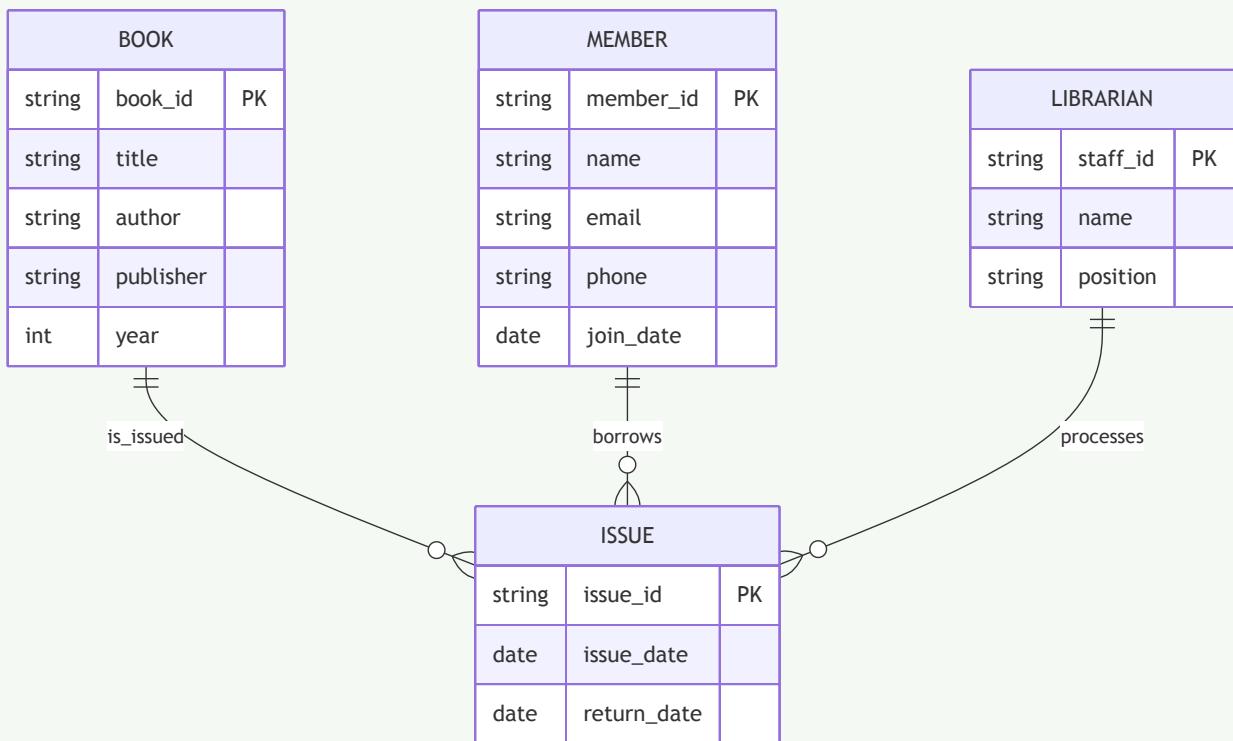
## Mnemonic

“PFSC” (Person First Shows Credentials)

## Question 2(c) [7 marks]

Draw E R Diagram of Library Management System.

## Solution



## Mnemonic

“LIMB” (Library Items, Members, Borrowing)

## Question 2(a) OR [3 marks]

Explain union operation in relational algebra with example.

## Solution

Union Operation ( $\cup$ )	Description
<b>Syntax</b>	$Relation1 \cup Relation2$
<b>Function</b>	Combines tuples from both relations

**Requirement**

Both relations must be union-compatible

**Example:** Students\_CS  $\cup$  Students\_IT**Mnemonic**

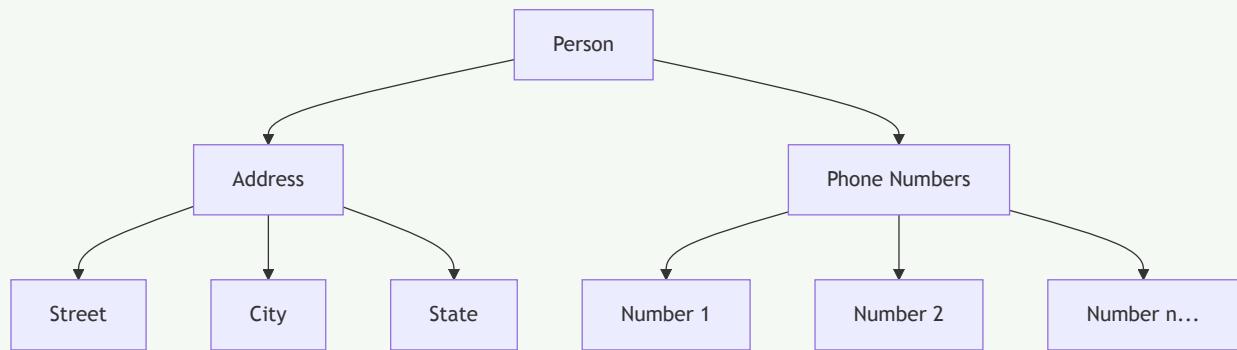
“CUP” (Combining Union of Parts)

**Question 2(b) OR [4 marks]**

Define Composite attribute and Multivalued attribute with example

**Solution**

Attribute Type	Description	Example
<b>Composite</b>	Can be divided into smaller subparts	Address (street, city, state, zip)
<b>Multivalued</b>	Can have more than one value	Phone numbers, Email addresses

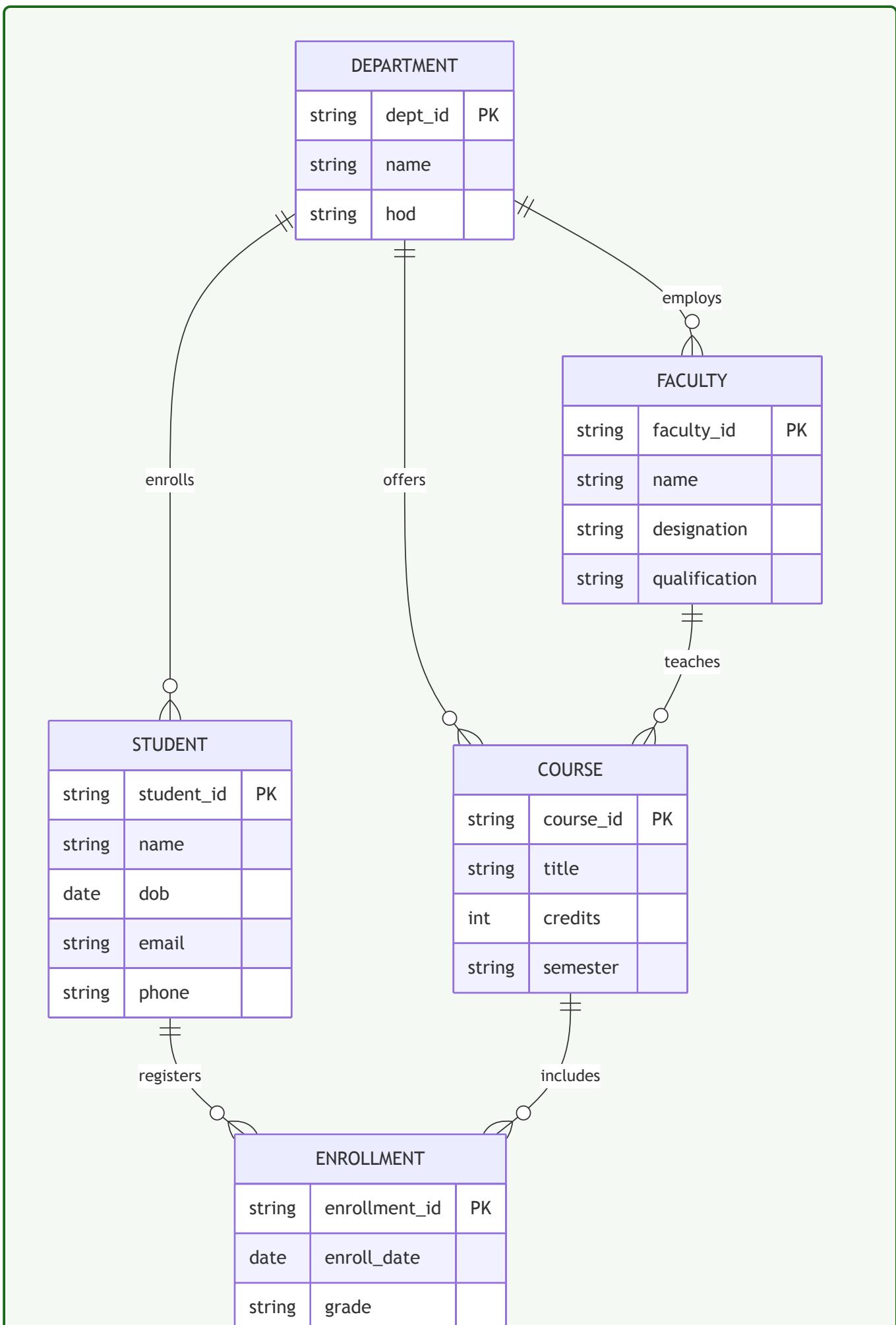
**Diagram:****Mnemonic**

“CoMbo” (Composite has Multiple components)

**Question 2(c) OR [7 marks]**

Draw E R Diagram of College Management System.

## Solution



### Mnemonic

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

### Question 3(a) [3 marks]

List different data types in SQL and Explain in brief

### Solution

Data Type Category	Examples	Usage
Numeric	INT, FLOAT, DECIMAL	Store numbers
Character	CHAR, VARCHAR, TEXT	Store text
Date/Time	DATE, TIME, TIMESTAMP	Store temporal data
Boolean	BOOLEAN	Store true/false values
Binary	BLOB, BINARY	Store binary data

### Mnemonic

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

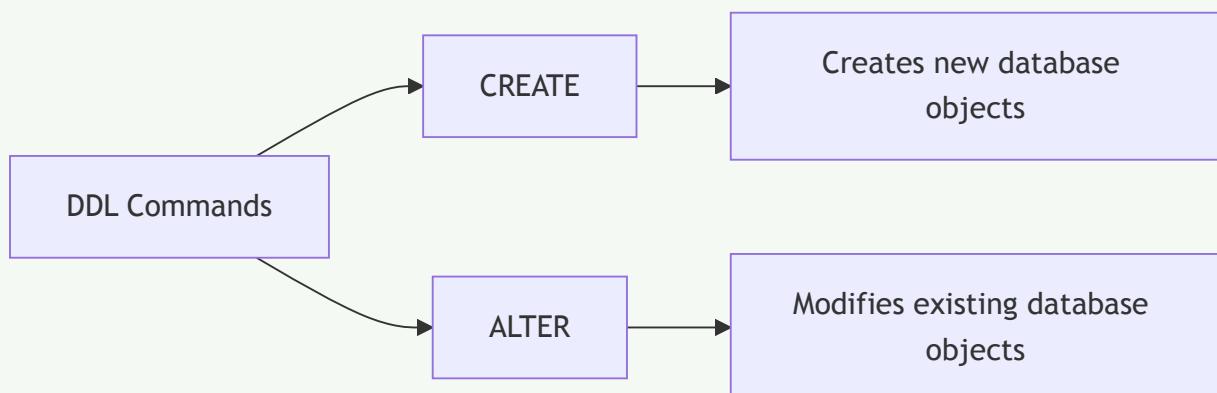
### Question 3(b) [4 marks]

Explain any two DDL Commands with Syntax and Example

### Solution

Command	Syntax	Example
<b>CREATE</b>	CREATE TABLE table_name (column_definitions);	CREATE TABLE Student (id INT PRIMARY KEY, name VARCHAR(50));
<b>ALTER</b>	ALTER TABLE table_name ADD/DROP/MODIFY column_name data_type;	ALTER TABLE Student ADD email VARCHAR(100);

Diagram:



### Mnemonic

“CAD” (Create And Define)

### Question 3(c) [7 marks]

Write the Output of Following Query. 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)

#### Solution

Function	Result	Explanation
CEIL(123.57)	124	Smallest integer $\geq 123.57$
CEIL(4.1)	5	Smallest integer $\geq 4.1$
MOD(12,4)	0	Remainder of $12 \div 4$
MOD(10,4)	2	Remainder of $10 \div 4$
POWER(2,3)	8	2 raised to power 3
POWER(3,3)	27	3 raised to power 3
ROUND(121.413,1)	121.4	Round to 1 decimal place
ROUND(121.413,2)	121.41	Round to 2 decimal places
FLOOR(25.3)	25	Largest integer $\leq 25.3$
FLOOR(25.7)	25	Largest integer $\leq 25.7$
LENGTH('AHMEDABAD')	9	Number of characters
ABS(-25)	25	Absolute value of -25
ABS(36)	36	Absolute value of 36

#### Mnemonic

"CMPRFLA" (Ceiling, Modulus, Power, Round, Floor, Length, Absolute)

### Question 3(a) OR [3 marks]

Explain any three Date Functions.

#### Solution

Date Function	Purpose	Example	Result
ADD_MONTHS	Adds months to date	ADD_MONTHS('01-JAN-2023', 3)	01-APR-2023
MONTHS_BETWEEN	Calculates months between dates	MONTHS_BETWEEN('01-MAR-2023', '01-JAN-2023')	
SYSDATE	Returns current date and time	SYSDATE	Current system date/time

#### Mnemonic

"AMS" (Add\_months, Months\_between, Sysdate)

### Question 3(b) OR [4 marks]

Explain any two DML Commands with Syntax and Example

#### Solution

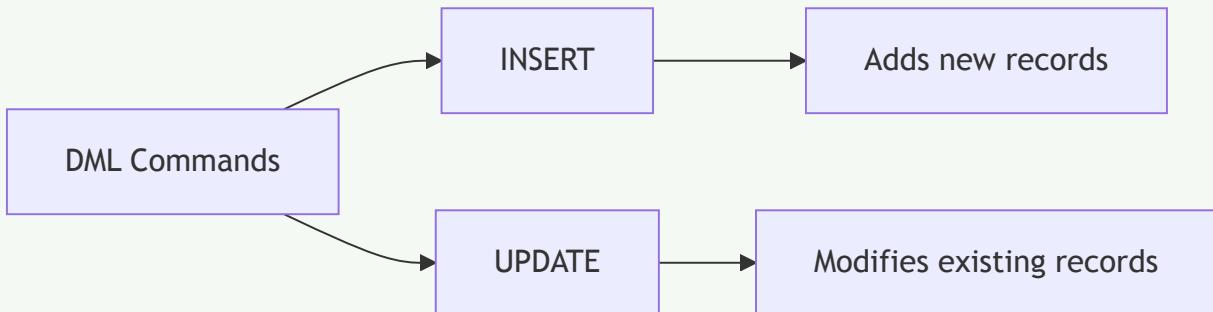
Command	Syntax	Example
INSERT	INSERT INTO table_name VALUES (value1, value2,...);	INSERT INTO Student VALUES (1, 'Raj', 'raj@example.com');

### UPDATE

UPDATE table\_name SET  
column=value WHERE  
condition;

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

Diagram:



### Mnemonic

"IUM" (Insert, Update, Manipulate)

## Question 3(c) OR [7 marks]

For the table: EMP(emp\_no, emp\_name, designation, salary, deptno), Write SQL commands for following operations.

### Solution

Operation	SQL Command
Create table EMP	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;
Give the emp_no, emp_name, designation, salary, deptno of EMP Display information of all employees whose name starts with 'p'	SELECT * FROM EMP WHERE emp_name LIKE 'p%';
Display department wise salary total	SELECT deptno, SUM(salary) AS total_salary FROM EMP GROUP BY deptno;
Add new column email_id in EMP table Change the column name "designation" to "post"	ALTER TABLE EMP ADD email_id VARCHAR(100); ALTER TABLE EMP RENAME COLUMN designation TO post;
Delete all the records from the table person	DELETE FROM person;

### Mnemonic

"CSDAACD" (Create, Select, Display, Aggregate, Add, Change, Delete)

## Question 4(a) [3 marks]

List different aggregate functions and explain any one with syntax and example.

## Solution

Aggregate Function	Purpose
<b>SUM</b>	Calculates total
<b>AVG</b>	Calculates average
<b>COUNT</b>	Counts number of rows
<b>MAX</b>	Finds maximum value
<b>MIN</b>	Finds minimum value

### Example for AVG:

`AVG(column\_name)` - Calculates average of values in column  
`SELECT AVG(salary) FROM Employee;` - Returns average salary

## Mnemonic

“SCAMM” (Sum, Count, Avg, Max, Min)

## Question 4(b) [4 marks]

Define the transaction with example.

## Solution

Transaction Concept	Description
<b>Definition</b>	Logical unit of work that must be completely processed or completely fail
<b>Properties</b>	ACID (Atomicity, Consistency, Isolation, Durability)
<b>States</b>	Active, Partially Committed, Committed, Failed, Aborted

### Example:

```
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;
```

## Mnemonic

“TAPS” (Transaction As Process Set)

## Question 4(c) [7 marks]

What is an Operator in SQL? Explain Arithmetic and Logical operators with Syntax and Example

## Solution

Type	Operators	Example	Result
<b>Arithmetic</b>	<code>+</code> (Addition)	<code>5 + 3</code>	8
	<code>-</code> (Subtraction)	<code>5 - 3</code>	2
	<code>*</code> (Multiplication)	<code>5 * 3</code>	15
	<code>/</code> (Division)	<code>15 / 3</code>	5
	<code>%</code> (Modulus)	<code>5 % 2</code>	1
<b>Logical</b>	<code>AND</code>	<code>salary &gt; 30000 AND dept = 'IT'</code>	True if both conditions true

OR	salary > 50000 OR dept = 'HR'	True if either condition true
NOT	NOT (salary < 20000)	True if salary not less than 20000

### SQL Examples:

```

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

```

### Mnemonic

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

### Question 4(a) OR [3 marks]

List different numeric functions and explain any one with syntax and example.

#### Solution

Numeric Function	Purpose
<b>ROUND</b>	Rounds a number to specified decimal places
<b>TRUNC</b>	Truncates a number to specified decimal places
<b>CEIL</b>	Returns smallest integer greater than or equal to number
<b>FLOOR</b>	Returns largest integer less than or equal to number
<b>ABS</b>	Returns absolute value

#### Example for ROUND:

ROUND(number, decimal\\_places) - Rounds number to specified decimal places  
SELECT ROUND(125.679, 2) FROM DUAL; - Returns 125.68

### Mnemonic

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

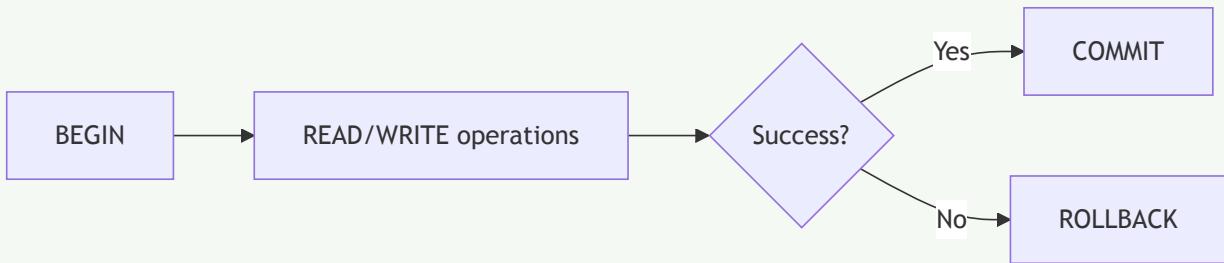
### Question 4(b) OR [4 marks]

List various database operations of a transaction.

#### Solution

Operation	Description
<b>BEGIN/START</b>	Marks transaction start point
<b>READ</b>	Retrieves data from database
<b>WRITE</b>	Modifies data in database
<b>COMMIT</b>	Makes changes permanent
<b>ROLLBACK</b>	Undoes changes and returns to start point
<b>SAVEPOINT</b>	Creates points to rollback partially

Diagram:



Mnemonic

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

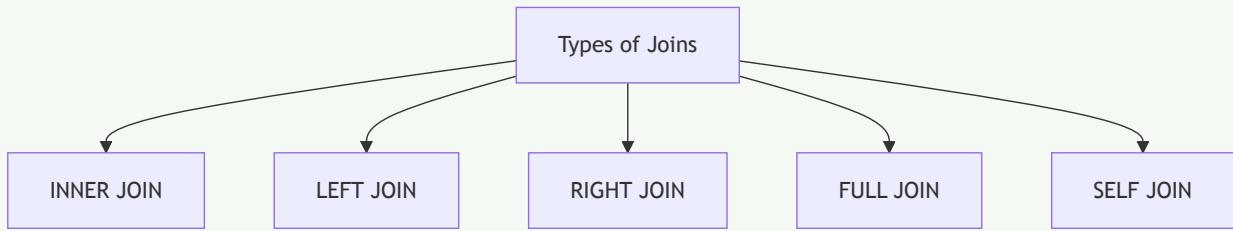
### Question 4(c) OR [7 marks]

What is join? Explain different types of joins with syntax and example.

Solution

Join Type	Description	Syntax Example
<b>INNER JOIN</b>	Returns rows when there is a match in both tables	SELECT * FROM TableA INNER JOIN TableB ON TableA.id = TableB.id;
<b>LEFT JOIN</b>	Returns all rows from left table and matched rows from right	SELECT * FROM TableA LEFT JOIN TableB ON TableA.id = TableB.id;
<b>RIGHT JOIN</b>	Returns all rows from right table and matched rows from left	SELECT * FROM TableA RIGHT JOIN TableB ON TableA.id = TableB.id;
<b>FULL JOIN</b>	Returns rows when there is a match in one of the tables	SELECT * FROM TableA FULL JOIN TableB ON TableA.id = TableB.id;
<b>SELF JOIN</b>	Joins a table to itself	SELECT * FROM Employee e1 JOIN Employee e2 ON e1.manager_id = e2.emp_id;

Diagram:



Mnemonic

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

### Question 5(a) [3 marks]

Convert the customer relation into 1NF shown below. Customer

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

## Solution

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

## Mnemonic

“AFM” (Atomic values, Flatten Multivalued attributes)

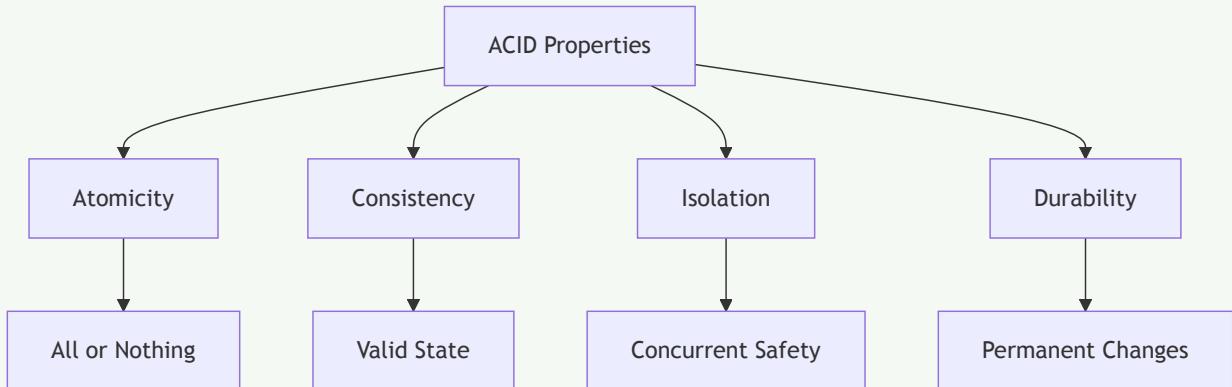
## Question 5(b) [4 marks]

List and Explain ACID properties of transaction.

## Solution

ACID Property	Description
<b>Atomicity</b>	Transaction executes completely or not at all
<b>Consistency</b>	Database remains consistent before and after transaction
<b>Isolation</b>	Concurrent transactions don't interfere with each other
<b>Durability</b>	Committed changes are permanent even after system failure

## Diagram:



## Mnemonic

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

## Question 5(c) [7 marks]

List different types of functional dependencies and explain each using example.

## Solution

Functional Dependency	Description	Example
<b>Trivial FD</b>	$X \rightarrow Y$ where $Y$ is a subset of $X$	{StudentID, Name} $\rightarrow$ {Name}

**Non-trivial FD**

X  
 $\rightarrow Y$  where  $Y$  is not a subset of  $X$   
Part of composite key  
determines non-key attribute

{StudentID}

$\rightarrow \{Name\}$

{CourseID,

StudentID}

$\rightarrow \{CourseName\}$

**Partial FD**

X  
 $\rightarrow Y$  and  $Y \rightarrow Z$  implies  $X \rightarrow Z$

{StudentID}  $\rightarrow$

{DeptID} and {DeptID}  $\rightarrow$

{DeptName}

**Transitive FD**

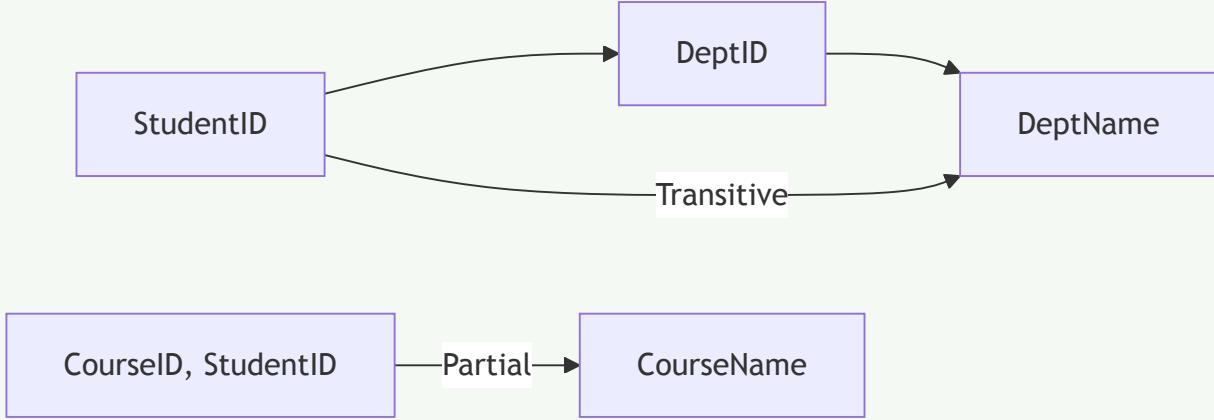
One attribute determines set  
of values for another

{CourseID}

$\rightarrow \rightarrow \{TextbookID\}$

**Multivalued FD**

Diagram:



**Mnemonic**

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

### Question 5(a) OR [3 marks]

Convert the Depositor\_Account relation into 2NF shown below. Where functional dependencies(FD) are as under, FD1: {cid, ano}  $\rightarrow$  {access\_date, balance, bname}

**Solution**

**Account Table (2NF):**

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

**Depositor Table (2NF):**

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

**Mnemonic**

“RPKD” (Remove Partial Key Dependencies)

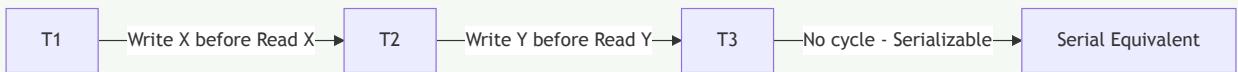
### Question 5(b) OR [4 marks]

Explain conflict serializability.

## Solution

Concept	Description
<b>Definition</b>	Schedule is conflict serializable if equivalent to some serial schedule
<b>Conflict Operations</b>	Read-Write, Write-Read, Write-Write operations on same data item
<b>Conflict Graph</b>	Directed graph showing conflicts between transactions
<b>Testing</b>	Schedule is conflict serializable if conflict graph has no cycles

### Diagram:



## Mnemonic

“COGS” (Conflict Operations Graph Serializable)

## Question 5(c) OR [7 marks]

Explain 3NF normalization with example

## Solution

Normal Form	Definition	Example
<b>1NF</b>	Atomic values, no repeating groups	Student(ID, Name, Phone1, Phone2) → <i>Student(ID, Name, Phone)</i>
<b>2NF</b>	1NF + No partial dependencies	Order(OrderID, ProductID, CustomerID, ProductName) → <i>Order(OrderID, ProductID, CustomerID)</i> Product(ProductID, ProductName)
<b>3NF</b>	2NF + No transitive dependencies	Student(ID, DeptID, DeptName) → <i>Student(ID, DeptID)</i> + <i>Department(DeptID, DeptName)</i>

### Violation Example:

1 Employee(EmpID, EmpName, DeptID, DeptName, Location)

### 3NF Conversion:

1 Employee(EmpID, EmpName, DeptID)  
2 Department(DeptID, DeptName, Location)

### Diagram:



## Mnemonic

“APTN” (Atomic values, Partial dependencies removed, Transitive dependencies removed, Normalized)