

33) How do you write a query in tuple relational calculus?

2M

Ans:

In Tuple Relational Calculus (TRC), a query is written as:

$$\{t \mid P(t)\}$$

Where:

- $t \rightarrow$ a tuple variable.
- $P(t) \rightarrow$ a predicate (condition) that must be true for tuple t .

Example: To get names of students from relation STUDENT(name, age, dept):

$$\{t.name \mid STUDENT(t) \wedge t.dept = "CSE"\}$$

👉 So, the general form is: { $t \mid$ condition on t }.

36) Write the importance of Logical data independence.

2M

Ans:

- 1) Logical independence allows changes in the logical schema (like adding/removing attributes, new relationships) without affecting the external schema or application programs.
- 2) It ensures data abstraction and flexibility, so users and applications remain unaffected even if the database structure is modified at the logical level

37) Comment on “Entity verses Relationship”.

2M

Ans:

- Entity represents a real-world object with independent existence (e.g., Student, Teacher).
- Relationship represents an association between two or more entities (e.g., Student-enrolls-Course).

In short: *Entities are things, Relationships show how those things are connected.*

38) List out the Various types of Joins.

2M

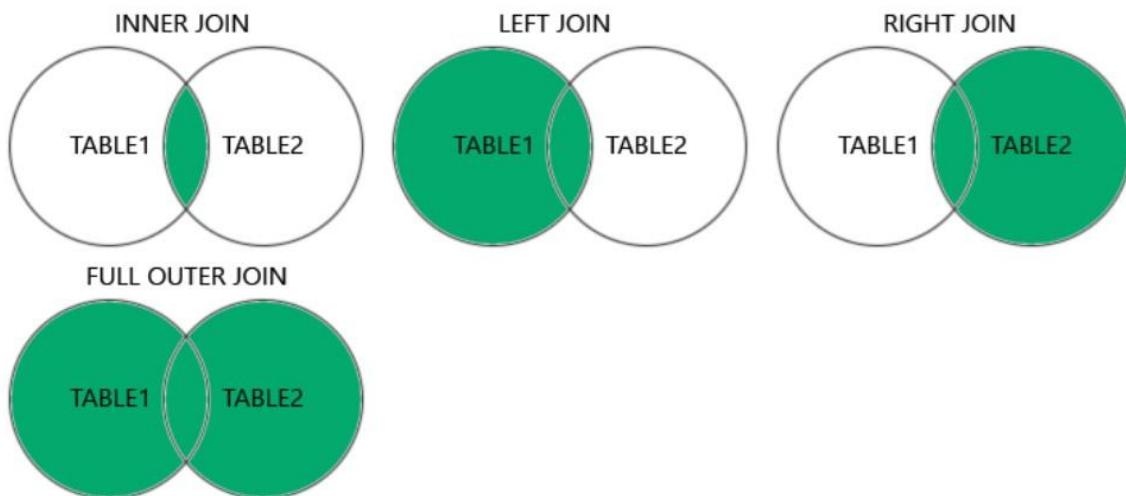
Ans: Types of Joins (2 Marks):

1. Inner Join
2. Left Outer Join
3. Right Outer Join
4. Full Outer Join
5. Cross Join (Cartesian Product)
6. Self Join
7. Natural Join

Different Types of SQL JOINS

Here are the different types of the JOINS in SQL:

- **(INNER) JOIN**: Returns records that have matching values in both tables
- **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
- **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
- **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table



39) Differentiate between Procedural Query Language and Declarative Query Language 2M

Ans:

- **Procedural Query Language:** Specifies what data is required and how to retrieve it (step-by-step).
Example: Relational Algebra.
- **Declarative Query Language:** Specifies what data is required, but not how to retrieve it.
Example: SQL, Relational Calculus.

In short: *Procedural = What + How, Declarative = Only What.*

40) Define Schema and Instance. 2M

Ans:

Schema and Instance (2 Marks):

- **Schema:** The overall structure/blueprint of a database that defines tables, attributes, and relationships (logical design).
Example: *Student(Roll, Name, Dept).*
- **Instance:** The actual data present in the database at a particular moment of time.
Example: *rows of Student table.*

41) What are integrity constraints? Define the terms primary key constraint and foreign key constraint

2M

Ans:

Integrity constraints are rules in a database that ensure the accuracy and consistency of data.

- Primary Key Constraint: Ensures that each record in a table is unique and not NULL.
- Foreign Key Constraint: Ensures referential integrity by linking a column in one table to the primary key of another table.

Example: *Student(RollNo PRIMARY KEY)* and *Marks(RollNo FOREIGN KEY references Student)*

42) Define Domain, Tuple and Relation.

2M

Ans:

Relation: STUDENT

Roll	Name	Dept
101	Amit	CSE
102	Sara	ECE
103	Raj	ME

Domain: The set of all possible values an attribute can have.

Example: Dept domain = {CSE,ECE,ME}.

Tuple: A single row/record in a relation, representing data about one entity.

Example: (101, "Amit", "CSE").

Relation: A table consisting of rows (tuples) and columns (attributes).

Example: STUDENT(Relation) with attributes (Roll, Name, Dept).

43) Write about on Entity and Entity-Set

2M

Ans:

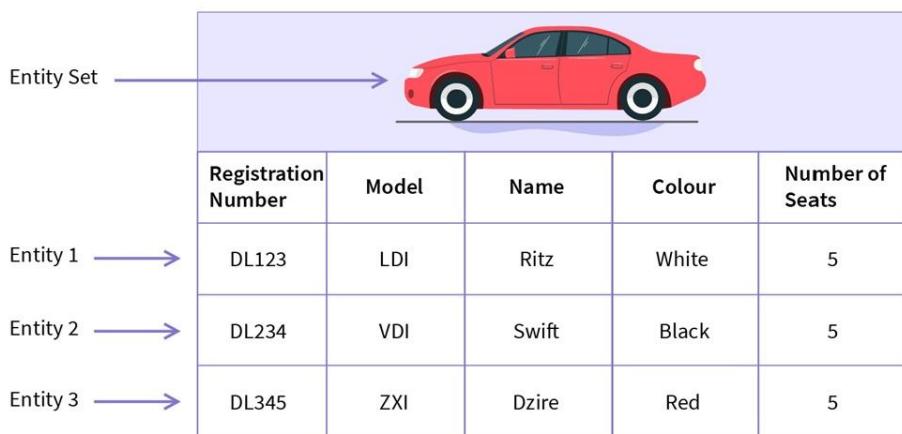
- Entity: A real-world object that has a distinct existence and can be uniquely identified.

Example: *Student (Roll=101, Name=Rahul)*.

- Entity Set: A collection of similar entities that share the same attributes.

Example: *STUDENT entity set = {all students in database}*.

In short: *Entity = single object, Entity Set = group of similar objects*.



44) What are different Data Base Languages? What are the different SQL constructs(keywords) used in these Languages? 2M

Ans:

1. **Data Definition Language (DDL):** Defines structure of database objects.
 - o **SQL Keywords:** CREATE, ALTER, DROP, TRUNCATE.
2. **Data Manipulation Language (DML):** Used to access and modify data.
 - o **SQL Keywords:** SELECT, INSERT, UPDATE, DELETE.
3. **Data Control Language (DCL):** Controls access and permissions.
 - o **SQL Keywords:** GRANT, REVOKE.
4. **Transaction Control Language (TCL):** Manages transactions.
 - o **SQL Keywords:** COMMIT, ROLLBACK, SAVEPOINT.

In short: *DDL = structure, DML = data, DCL = access, TCL = transactions.*

45) Write about inheritance, super class and sub class. 2M

Ans:

Inheritance is a concept where a subclass automatically acquires (inherits) the attributes and relationships of its superclass.

It allows data sharing, reusability, and specialization in databases

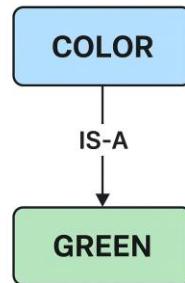
A Superclass is a higher-level (generalized) entity set that contains the common attributes and relationships shared by one or more lower-level entities (subclasses).

A Subclass is a lower-level (specialized) entity set that inherits attributes and relationships from its superclass and may also have its own additional attributes.

The relation between super class and sub class is “IS-A” relation

Example:

- Superclass: COLOR
- Subclass: GREEN inherits properties of COLOR and has its own attributes.



46) Compare UNIQUE and NOT NULL Constraints.

2M

Ans:

- **UNIQUE Constraint:** Ensures that all values in a column are distinct (no duplicate values allowed). It can accept one NULL value.
- **NOT NULL Constraint:** Ensures that a column cannot have NULL values, but duplicate values are allowed.

In short:

UNIQUE → No duplicates (NULL allowed once).

NOT NULL → No NULL values (duplicates allowed).