# CHHATRAPATI SHAHU JI MAHARAJ UNIVERSITY, KANPUR

ACCREDITED WITH GRADE A++ NAAC

प्रश्नBANK
Bridge of Academic Novelties in Knowledge

## KANPUR UNIVERSITY'S

# QUESTION BANK

Brief and Intensive Notes
Multiple Choice Questions

Based On NEP 2020

# DATABASE MANAGEMENT SYSTEM

**B.Sc. IV SEM**

DR. AGNIVESH

MRS. SANCHITA LAXMI

# Course Title: Database Management System

## Unit I

Introduction: Database System Concepts, File system vs. database system, Database system architecture, Data models and their types, Data base scheme and instances, Data independence, Database Languages and Interfaces. ER model concepts: Notations for ER diagram, Extended E-R diagram, Extended E-R model, E-R model design issues, constraints, and keys: Weak entity set strong entity set, Relationships of higher degree.

### 1. Introduction: Database System Concepts

A **database system** is a collection of data and a set of programs that manage and access that data. It ensures data consistency, integrity, and security, while allowing multiple users to access and manipulate the data efficiently. Key concepts include:

- **Database**: An organized collection of data.
- **DBMS (Database Management System)**: Software that handles data storage, retrieval, and updating.
- **Users**: End users, application programmers, and database administrators.

### 2. File System vs. Database System

| Feature | File System | Database System |
|---|---|---|
| Data Redundancy | High | Low |
| Data Access | Manual and sequential | Efficient and flexible (SQL) |
| Integrity & Security | Hard to enforce | Built-in support |
| Data Independence | Not supported | Supported (logical and physical) |
| Concurrent Access | Difficult to manage | Managed by DBMS |

### 3. Database System Architecture

Refers to the structure and interaction of components in a DBMS:

- **Three-tier architecture**:
  - **External Level**: User views
  - **Conceptual Level**: Logical structure
  - **Internal Level**: Physical storage

- **Two-tier & Client-Server Architecture**: Common in applications where the database runs on a server and clients interact with it remotely.

**4. Data Models and Their Types**

A **data model** defines how data is connected and how it can be processed. Types include:

- **Hierarchical Model**: Tree-like structure (parent-child).
- **Network Model**: More flexible graph structure.
- **Relational Model**: Data in tables (most popular).
- **Object-Oriented Model**: Combines object-oriented programming concepts.
- **Entity-Relationship (ER) Model**: Represents entities and relationships in a visual format.

**5. Database Schema and Instances**

- **Schema**: The logical design of the database (structure, tables, types); fixed over time.
- **Instance**: The actual data in the database at a specific moment; changes frequently.

**6. Data Independence**

The ability to change the schema at one level without affecting the schema at the next higher level:

- **Logical Data Independence**: Change in logical schema (e.g., add a field) doesn't affect applications.
- **Physical Data Independence**: Change in physical storage doesn't affect logical schema.

**7. Database Languages and Interfaces**

- **DDL (Data Definition Language)**: Define database structure (e.g., CREATE, ALTER).
- **DML (Data Manipulation Language)**: Handle data (e.g., SELECT, INSERT).
- **DCL (Data Control Language)**: Permissions and access (e.g., GRANT, REVOKE).
- **TCL (Transaction Control Language)**: Manage transactions (e.g., COMMIT, ROLLBACK).
- **Interfaces**: Command-line tools, GUIs, and application interfaces (e.g., JDBC, ODBC).

**ER Model Concepts**

An **ER (Entity-Relationship) Diagram** is a **visual representation of the data and its relationships** in a database. It's part of the **ER model**, which is used in database design to map out the structure logically before implementing it.

An **entity** is **a real-world object or concept** that we want to store information about in a database.

- Think of it as a **noun** — something you can identify.
- Each entity will eventually become a **table** in the database.

**Examples:**

- A **Student**
- A **Book**
- A **Car**
- A **Teacher**

In ER diagrams, **entities** are represented by **rectangles**.

An **attribute** is a **property or characteristic** of an entity — it describes or gives more details about the entity.

- Think of it as an **adjective** — it tells you more about the noun (entity).
- In databases, attributes become **columns** in the table.

**Examples:** For the entity **Student**:

- Student_ID
- Name
- Date_of_Birth
- Email

In ER diagrams, **attributes** are represented by **ovals** and are connected to their entity.

**1. Notations for ER Diagram**

- **Entity**: Rectangle
- **Attribute**: Oval
- **Relationship**: Diamond
- **Primary Key**: Underlined attribute
- **Multivalued Attribute**: Double oval
- **Derived Attribute**: Dotted oval

**2. Extended E-R (EER) Diagram**

Enhances the ER model with:

- **Generalization**: Combining similar entities into a higher-level entity.
- **Specialization**: Breaking an entity into sub-entities.
- **Aggregation**: Treating a relationship as an entity.
- **Inheritance**: Attributes from super-class are inherited by sub-class.

**3. E-R Model Design Issues**

- Choosing between entity sets and attributes or relationship sets.
- Naming conventions for clarity.
- Determining participation (total vs. partial) and cardinality (1:1, 1:N, M:N).
- Avoiding redundancy and ensuring normalization.

**4. Constraints and Keys**

- **Constraints**: Rules to ensure data integrity.
  - **Key Constraints**: Uniqueness of attributes.
  - **Participation Constraints**: Total (every instance must participate) or partial.
- **Keys**:
  - **Primary Key**: Unique identifier for entities.
  - **Candidate Key**: Other possible keys.
  - **Foreign Key**: Attribute referring to a primary key in another table.

**5. Weak Entity Set and Strong Entity Set**

- **Strong Entity**: Can exist independently with a primary key.
- **Weak Entity**: Depends on a strong entity; lacks a full key.
  - Identified using **partial key** and a **discriminator**.

**6. Relationships of Higher Degree**

- **Binary Relationship**: Between two entities (most common).
  - **Ternary (or higher) Relationship**: Involves three or more entities.
  - More complex and often broken down into binary relationships with care to preserve meaning.

**1. What is a key advantage of using a DBMS over a traditional file system?**

A. DBMS stores data in flat files only
B. File systems provide better data consistency than DBMS
C. DBMS supports data redundancy and inconsistency
D. DBMS allows centralized control and reduces data redundancy

**Answer:** D. DBMS allows centralized control and reduces data redundancy
**Explanation:**
A **DBMS** offers **centralized control** over data, enforcing rules to **reduce redundancy** (duplicate data) and maintain **data integrity and consistency**. In contrast, traditional file systems often suffer from **data duplication and inconsistency** due to lack of centralized management.

**2. How many levels are there in the ANSI/SPARC database architecture?**

A. One
B. Two
C. Three
D. Four

**Answer:** C. Three
**Explanation:**
The **ANSI/SPARC architecture** defines a **three-level** architecture for databases:

- **Internal level** (physical storage)
- **Conceptual level** (logical structure)
- **External level** (user view)

**3. Which level of database architecture defines how the data is physically stored?**

A. External level
B. Conceptual level
C. Internal level
D. Logical level

**Answer:** C. Internal level
**Explanation:**
The **Internal level** describes the **physical storage** of data on hardware such as disks, including indexing, compression, and file organization.

**4. What is the main purpose of the three-schema architecture in DBMS?**

A. To speed up data retrieval
B. To support multiple programming languages
C. To separate user views from physical data storage
D. To provide graphical interfaces for users

**Answer:** C. To separate user views from physical data storage
**Explanation:**
The **three-schema architecture** ensures **data abstraction**, separating the way data is stored (internal level) from the way users view and interact with data (external level), with a **conceptual schema** in between to manage structure and consistency.

**5. Which of the following best describes the difference between a database schema and a database instance?**

A. Schema is temporary; instance is permanent
B. Schema describes the structure, while instance is the data at a given time

C. Schema contains data; instance contains metadata
D. Schema changes frequently; instance remains fixed

**Answer:** B. Schema describes the structure, while instance is the data at a given time
**Explanation:**
A **database schema** is the **logical design or blueprint** of the database, including tables, fields, relationships, etc. A **database instance** refers to the **actual data** stored in the database **at a particular moment**. The schema is generally fixed, while the instance changes as data is added, updated, or deleted.

**6. What is the primary goal of data independence in a DBMS?**

A. To allow multiple users to access data simultaneously
B. To reduce the need for data backups
C. To enable changes in schema at one level without affecting other levels
D. To improve data compression

**Answer:** C. To enable changes in schema at one level without affecting other levels
**Explanation:**
**Data independence** refers to the ability to **modify the schema at one level** (e.g., physical or logical) **without altering the schema at the next higher level**, ensuring flexibility and minimal disruption to applications.

**7. Which type of data independence allows changes in the logical schema without affecting the external schema or application programs?**

A. Physical data independence
B. Logical data independence
C. External data independence
D. Internal data independence

**Answer:** B. Logical data independence
**Explanation:**
**Logical data independence** is the capacity to change the **conceptual (logical) schema**—like adding or removing fields or tables—**without affecting** how users or applications access the data through their **external views**.

**8. Which of the following is a language used to define the structure of a database?**

A. DML (Data Manipulation Language)
B. DCL (Data Control Language)

C. DDL (Data Definition Language)
D. TCL (Transaction Control Language)

**Answer:** C. DDL (Data Definition Language)
**Explanation:**
**DDL** is used to define and modify the **structure** of database objects like tables, indexes, and schemas. Commands include CREATE, ALTER, and DROP.

**9. Which DBMS language is primarily used to retrieve and manipulate data?**

A. DDL
B. DML
C. DCL
D. HTML

**Answer:** B. DML
**Explanation:**
**DML (Data Manipulation Language)** is used for **retrieving, inserting, updating, and deleting** data in the database. Examples include SELECT, INSERT, UPDATE, and DELETE.

**10. What is the purpose of the GRANT and REVOKE commands in DBMS?**

A. To define new tables
B. To control user access to data
C. To manipulate data in a table
D. To manage transaction rollback

**Answer:** B. To control user access to data
**Explanation:**
**GRANT** and **REVOKE** are part of **DCL (Data Control Language)**, used to **assign or remove permissions** from users on database objects for security and access control.

**11. Which language in DBMS is used to manage transactions like commit and rollback?**

A. DDL
B. DCL
C. DML
D. TCL

**Answer:** D. TCL (Transaction Control Language)
**Explanation:**

**TCL** includes commands like COMMIT, ROLLBACK, and SAVEPOINT, which are used to **manage transactions** and ensure data integrity during operations.

**12. In an ER model, which of the following represents a real-world object or concept?**

A. Attribute
B. Entity
C. Relationship
D. Key

**Answer:** B. Entity
**Explanation:**
An **entity** represents a **real-world object or concept** such as a person, place, or event that can be distinctly identified. Entities are usually represented by rectangles in ER diagrams.

**13. What does a diamond shape represent in an ER diagram?**

A. Attribute
B. Entity
C. Relationship
D. Primary Key

**Answer:** C. Relationship
**Explanation:**
In an ER diagram, a **diamond shape** represents a **relationship** between entities. For example, a "Works_For" relationship might connect the entities "Employee" and "Company".

**14. Which of the following is an example of a derived attribute in an ER model?**

A. Employee ID
B. Date of Birth
C. Age
D. Name

**Answer:** C. Age
**Explanation:**
**Age** is a **derived attribute** because it can be calculated from another stored attribute like **Date of Birth**. Derived attributes are typically shown with a dashed oval in ER diagrams.

**15. Which key uniquely identifies each record in a table?**

A. Foreign Key
B. Composite Key
C. Super Key
D. Primary Key

**Answer:** D. Primary Key
**Explanation:**
A **Primary Key** uniquely identifies each record in a table. It cannot have NULL values and ensures the **uniqueness** of each row.

**16. What is the main purpose of a foreign key in a relational database?**

A. To uniquely identify a tuple
B. To store large text data
C. To enforce referential integrity between tables
D. To sort records in a table

**Answer:** C. To enforce referential integrity between tables
**Explanation:**
A **Foreign Key** links one table to another and ensures **referential integrity** by requiring the values in the foreign key column to match values in the referenced table's primary key column.

**17. Which of the following constraints ensures that a column cannot have NULL values?**

A. UNIQUE
B. DEFAULT
C. NOT NULL
D. CHECK

**Answer:** C. NOT NULL
**Explanation:**
The **NOT NULL** constraint prevents a column from having NULL values, ensuring that a value must always be provided when inserting or updating data.

**18. Which of the following is true about a weak entity set?**

A. It can be uniquely identified by its own attributes
B. It always has a primary key
C. It depends on a strong entity set for identification
D. It does not have any attributes

**Answer:** C. It depends on a strong entity set for identification
**Explanation:**
A **weak entity set** cannot be uniquely identified by its own attributes alone. It requires a **foreign key relationship** with a **strong entity set** and typically uses a **partial key** combined with the strong entity's primary key.

**19. What distinguishes a strong entity set from a weak entity set?**

A. Strong entity sets don't have relationships
B. Strong entity sets must have at least two foreign keys
C. Strong entity sets have a primary key, while weak entity sets do not
D. Weak entity sets can exist independently

**Answer:** C. Strong entity sets have a primary key, while weak entity sets do not
**Explanation:**
A **strong entity set** has a **primary key** that uniquely identifies its entities, whereas a **weak entity set** lacks a primary key and relies on a related strong entity for its identification.

**20. What is a relationship of higher degree in an ER model?**

A. A relationship between two entities
B. A relationship with more than one attribute
C. A relationship involving more than two entity sets
D. A recursive relationship

**Answer:** C. A relationship involving more than two entity sets
**Explanation:**
A **higher-degree relationship** (also called **n-ary relationship**) involves **more than two entity sets**. For example, a ternary relationship involves three entities, such as "Doctor", "Patient", and "Hospital" in a "Treatment" relationship.

**21. In an ER diagram, how is a ternary relationship different from three binary relationships?**

A. It simplifies the diagram
B. It always has a primary key
C. It can capture the interdependency among all three entities
D. It does not involve any attributes

**Answer:** C. It can capture the interdependency among all three entities
**Explanation:**
A **ternary relationship** captures the **combined relationship** among **three entities**

simultaneously, which may not be correctly represented by simply using **three binary relationships**, as it could miss the **interdependence** among all three.

**22. Which of the following is NOT a type of DBMS?**

A. Hierarchical DBMS
B. Network DBMS
C. Relational DBMS
D. Distributed Operating System

**Answer:** D. Distributed Operating System
**Explanation:** Distributed Operating System is not a database management system. The primary types of DBMS are Hierarchical, Network, Relational, and Object-oriented.

**23. Which type of DBMS uses tables to represent data and relationships among those data?**

A. Network DBMS
B. Hierarchical DBMS
C. Relational DBMS
D. Object-oriented DBMS

**Answer:** C. Relational DBMS
**Explanation:** Relational DBMS stores data in tables (also called relations), making it easy to manage, query, and relate different sets of data.

**24. In which DBMS model is data represented in a tree-like structure?**

A. Object-oriented DBMS
B. Relational DBMS
C. Network DBMS
D. Hierarchical DBMS

**Answer:** D. Hierarchical DBMS
**Explanation:** In Hierarchical DBMS, data is organized in a tree format where each child has one parent, representing a one-to-many relationship.

**25. Which type of DBMS is best suited for complex data types like multimedia, images, and custom objects?**

A. Relational DBMS
B. Hierarchical DBMS
C. Object-oriented DBMS
D. Network DBMS

**Answer:** C. Object-oriented DBMS
**Explanation:** Object-oriented DBMS combines features of object-oriented programming and database systems, making it ideal for managing complex and non-traditional data types.

**26.** A major drawback of file systems over database systems is:
A. Fast access
B. High redundancy
C. Better structure
D. Less data
**Answer:** B
**Explanation:** File systems often lead to **data redundancy** due to duplication.

**27.** What does a DBMS offer that a traditional file system does not?
A. Text editing
B. Query processing
C. File encryption
D. Manual sorting
**Answer:** B
**Explanation:** Query processing via SQL is a major advantage of DBMS.

**28.** In three-tier architecture, the **conceptual level** describes:
A. How data is stored physically
B. How users interact with data
C. The logical structure of data
D. The hardware layer
**Answer:** C
**Explanation:** The conceptual level defines the **logical structure** of the database.

**29.** The internal level of database architecture deals with:
A. Security constraints
B. Data storage and indexing
C. User interfaces
D. ER diagrams
**Answer:** B
**Explanation:** The internal level focuses on **physical data storage**.

**30.** Inheritance in the extended ER model means:
A. One entity copies from another
B. Attributes are encrypted
C. Data is hidden
D. Entities are unrelated
**Answer:** A
**Explanation:** Subclasses **inherit** attributes from a parent entity (generalization/specialization).

***31. In the E-R model, which of the following is true about a weak entity set?***

A. It can exist without a related strong entity.
B. It must have a primary key of its own.
C. It has a partial key and needs a strong entity's key for identification.
D. It participates in a one-to-one relationship with its strong entity.

**Answer:** C
**Explanation:** A weak entity doesn't have a primary key of its own. It uses a **partial key** and depends on a **strong entity**'s key via an identifying relationship.

***32. Which of the following is NOT a valid reason for choosing an entity over an attribute in E-R modeling?***

A. The item has multiple descriptive attributes.
B. The item is referenced by other entities.
C. The item is subject to deletion rules.
D. The item is always a single value.

**Answer:** D
**Explanation:** If something is **always a single value**, it's better represented as an **attribute**, not an entity. Entities are chosen when the concept has **complex structure or semantics**.

***33. Which of the following best describes a key constraint in E-R models?***

A. It limits the number of entities in an entity set.
B. It specifies how entities participate in a relationship.
C. It states that no two entities in an entity set can have the same value for a key attribute.
D. It restricts the cardinality of a relationship.

**Answer:** C
**Explanation:** A **key constraint** enforces **uniqueness** among entities by requiring that no two share the same value for their **key attribute(s)**.

***34. Consider the relationship "Teaches" between entities "Professor" and "Course". If every course must be taught by exactly one professor, what constraint applies on the "Teaches" relationship?***

A. Total participation and many-to-one
B. Partial participation and one-to-many
C. Total participation and one-to-many
D. Total participation and many-to-many

**Answer:** A
**Explanation:** Since **every course must** be taught by **one professor**, it is **total participation** from the Course side and a **many-to-one** (many courses per professor, one professor per course) relationship.

***35. Which of the following does not influence the choice between representing something as an entity or as a relationship?***

A. Whether it has multiple attributes
B. Whether it has unique identification
C. Whether it is a physical or abstract object
D. Whether it changes frequently

**Answer:** D
**Explanation: Volatility** (how often something changes) does **not** directly affect the **entity vs. relationship** decision. Structure and semantics do.

***36. In E-R modeling, total participation is represented by:***

A. Dashed line
B. Diamond with double border
C. Double rectangle
D. Double line between entity and relationship

**Answer:** D
**Explanation: Total participation** means every entity instance must be related and is denoted by a **double line** between the entity and the relationship.

***37. An identifying relationship is always:***

A. One-to-one
B. Weak
C. Between a strong and weak entity
D. Multivalued

**Answer:** C
**Explanation:** An **identifying relationship** connects a **weak entity** to its **owning strong entity**, helping form the complete key for the weak entity.

***38. If an entity has a composite key, which of the following is true?***

A. It must be a weak entity
B. It cannot be part of a relationship
C. Its key consists of multiple attributes
D. It cannot participate in generalization

**Answer:** C
**Explanation:** A **composite key** means the **key is formed by a combination** of two or more attributes. It doesn't imply weakness or restriction on relationships.

***39. Which of these E-R model features cannot be directly mapped to a relational model?***

A. Multivalued attributes
B. Composite attributes
C. Weak entities
D. Primary keys

**Answer:** B
**Explanation: Composite attributes** (e.g., Name with subparts First, Last) are **not directly supported** in relational models and must be **flattened** or **broken into atomic parts**.

***40. Which of the following statements is true about participation constraints?***

A. They define whether an attribute can be null
B. They dictate how many entities can participate in a relationship
C. They define uniqueness of keys
D. They are the same as cardinality ratios

**Answer:** B
**Explanation: Participation constraints** define **how entities participate** in relationships — **totally** (every instance must participate) or **partially**.

***41. Which of the following must be true for a relationship between a strong and weak entity set?***

A. The weak entity has total participation in the relationship.
B. The strong entity has total participation in the relationship.
C. The relationship must be many-to-many.
D. The relationship can have no attributes.

**Answer:** A
**Explanation:** A weak entity **must** be associated with a strong entity to exist, hence it has **total participation** in the **identifying relationship** (shown with a double line in E-R diagrams).

42**. Which statement about weak entities is FALSE?**

A. A weak entity depends on a strong entity for existence.
B. A weak entity always has a composite attribute as a key.
C. A weak entity's key includes the primary key of a strong entity.
D. A weak entity uses an identifying relationship to connect to a strong entity.

**Answer:** B
**Explanation:** A weak entity uses a **partial key**, but it doesn't **have to be composite** in terms of multiple fields within itself. It becomes unique only when combined with the key of the strong entity, not necessarily from composite attributes.

***43. In a DBMS architecture, the mapping between the conceptual schema and the internal schema is needed because:***

A. Users directly access physical data structures.
B. It allows logical data independence.
C. It allows physical data independence.
D. It provides views for users.

**Answer:** C
**Explanation:** The **mapping between the conceptual and internal schema** enables **physical data independence** — allowing changes in the internal storage structure **without affecting the logical structure** of the database.

***44. Which of the following is NOT typically a function of the DBMS's query processor component?***

A. Query parsing and validation
B. Optimization of query execution plans
C. Transaction logging and recovery
D. Translating SQL queries into low-level instructions

**Answer:** C
**Explanation: Transaction logging and recovery** are handled by the **storage manager** and **recovery manager**, not the **query processor**. The query processor handles parsing, validation, optimization, and execution of queries.

***45. Which of the following best describes a ternary relationship in the E-R model?***

A. A relationship involving three attributes
B. A recursive relationship
C. A relationship involving three distinct entity sets
D. A composite attribute across three tables

**Answer:** C
**Explanation:** A **ternary relationship** connects **three different entity sets** (e.g., Doctor, Patient, Medication in a "Prescribes" relationship). It's not about attributes, but **entities**.

***46. In which of the following cases is a ternary relationship more appropriate than three binary relationships?***

A. When all participating entities are weak
B. When the meaning of the relationship **depends on the combination** of all three entities
C. When each binary relationship is independent
D. When the entities share a common attribute

**Answer:** B
**Explanation:** A **ternary relationship** is needed when the **semantics** depend on the **joint participation** of all three entities. Modeling it with binary relationships can **lose important constraints**.

***47. Which of the following is a disadvantage of replacing a ternary relationship with binary ones?***

A. It simplifies the schema
B. It can introduce redundancy
C. It preserves all constraints
D. It always improves performance

**Answer:** B
**Explanation:** Decomposing ternary relationships into binary can **introduce redundancy and ambiguity**, and may **fail to enforce constraints** that depend on the **three-way combination**.

***48. Consider a ternary relationship among Supplier, Part, and Project. Which of the following is likely to be true?***

A. Each part must belong to only one supplier and one project
B. A supplier may supply a part to multiple projects
C. A part is associated with only one project
D. The ternary relationship can always be converted into binary ones without loss

**Answer:** B
**Explanation:** In a ternary relationship like Supplies(Supplier, Part, Project), a **supplier can supply the same part to multiple projects**, and multiple combinations can exist. **Binary reduction** is often **not accurate** here.

***49. In a ternary relationship, how many entity sets are involved and what is the minimum number of participating instances needed to establish a valid relationship?***

A. 2 entity sets, 2 instances
B. 3 entity sets, 1 instance from each
C. 1 entity set, 3 instances
D. 3 attributes, 2 instances

**Answer:** B
**Explanation:** A ternary relationship involves **3 entity sets**, and a valid relationship requires **one instance from each** to form a complete relationship tuple.

***50. When mapping a ternary relationship to a relational schema, how is it typically represented?***

A. By creating three binary relations
B. By adding foreign keys to one of the participating entity tables
C. By creating a separate relation with foreign keys referencing all three entities
D. By combining all attributes into a composite attribute

**Answer:** C
**Explanation:** A ternary (or n-ary) relationship is typically mapped by creating a **separate relation** that includes **foreign keys referencing the primary keys** of each participating entity set.

**Unit II**

Introduction to SQL: Basic Structure of SQL Query, Set operators, SELECT, UNION, INTERSECT, and EXCEPT, Nested queries, Aggregate function, Null values, Derived Relations, Modification of the Database, Joined relations and up-dates in SQL.

**Introduction to SQL (Structured Query Language)**

**SQL** is the standard language used to communicate with relational databases. It is used to perform tasks such as querying data, updating records, and managing database structures.

1. Basic Structure of an SQL Query

The most common SQL query uses the SELECT statement:

Sql:

```
SELECT column1, column2
FROM table_name
WHERE condition
GROUP BY column
HAVING condition
ORDER BY column;
```

- **SELECT**: Specifies columns to retrieve.
- **FROM**: Specifies the table(s).
- **WHERE**: Filters records.
- **GROUP BY**: Groups rows sharing values.
- **HAVING**: Filters groups.
- **ORDER BY**: Sorts results.

**2. Set Operators in SQL**

Set operators combine the results of two or more queries.

- **UNION**: Combines results of two queries, removing duplicates.
- **UNION ALL**: Same as UNION but includes duplicates.
- **INTERSECT**: Returns only rows present in both queries.
- **EXCEPT** (or **MINUS** in some systems): Returns rows from the first query that are **not** in the second.

Example:

Sql:

```
SELECT name FROM students
UNION
SELECT name FROM teachers;
```

**3. SELECT Statement**

SELECT is used to retrieve data from a database.

Basic usage:

Sql:

```
SELECT * FROM employees;
```

With conditions:

Sql:

```
SELECT name, salary FROM employees
WHERE department = 'HR';
```

**4. UNION, INTERSECT, and EXCEPT**

These are used to combine the results of two SELECT statements.

- **UNION** → Adds and removes duplicates.
- **INTERSECT** → Only common rows.
- **EXCEPT** → Rows in first query, not in second.

All involved queries must have **same number of columns** and **compatible data types**.

**5. Nested Queries (Subqueries)**

A **subquery** is a query within another query.

Example:

Sql:

```
SELECT name FROM employees
WHERE dept_id IN (
  SELECT dept_id FROM departments WHERE location = 'New York'
);
```

Types:

- **Single-row subquery**: Returns one value.
- **Multi-row subquery**: Returns multiple values.
- **Correlated subquery**: Refers to columns of the outer query.

**6. Aggregate Functions**

Used to perform calculations on data sets:

- SUM() – Total
- AVG() – Average
- COUNT() – Number of rows
- MAX() – Maximum value
- MIN() – Minimum value

Example:

Sql:

SELECT AVG(salary) FROM employees;

**7. Null Values**

- **NULL** represents missing or unknown data.
- You cannot compare NULL using = or !=.
- Use IS NULL or IS NOT NULL.

Example:

Sql:

SELECT * FROM employees WHERE bonus IS NULL;

**8. Derived Relations (Subqueries in FROM Clause)**

A **derived relation** is a subquery used as a temporary table in the FROM clause.

Example:

Sql:

SELECT dept, avg_salary

```
FROM (
  SELECT department AS dept, AVG(salary) AS avg_salary
  FROM employees
  GROUP BY department
) AS dept_avg;
```

**9. Modification of the Database**

SQL provides commands to **modify** data:

- **INSERT**: Add new records

  Sql:

  ```
  INSERT INTO employees (name, salary) VALUES ('Alice', 50000);
  ```

- **UPDATE**: Modify existing data

  Sql:

  ```
  UPDATE employees SET salary = salary + 1000 WHERE dept_id = 2;
  ```

- **DELETE**: Remove data

  Sql:

  ```
  DELETE FROM employees WHERE emp_id = 101;
  ```

**10. Joined Relations (Joins) and Updates in SQL**

**Joins** combine rows from two or more tables based on a related column.

Types of Joins:

- **INNER JOIN**: Only matching rows.
- **LEFT JOIN**: All rows from the left table + matching rows from the right.
- **RIGHT JOIN**: All rows from the right table + matching from the left.
- **FULL JOIN**: All rows from both tables.

Example:

Sql:

```
SELECT e.name, d.name
```

```
FROM employees e
JOIN departments d ON e.dept_id = d.dept_id;
```

**Updates using joins** (may differ by SQL dialect):

Sql:

```
UPDATE employees
SET employees.salary = employees.salary + 500
FROM departments
WHERE employees.dept_id = departments.dept_id AND departments.name = 'IT';
```

**1. Which SQL command is used to retrieve data from a database?**

**A.** SELECT
B. GET
C. FETCH
D. RETRIEVE

**Answer:** A. SELECT

**Explanation:** The SELECT statement is used to retrieve data from a database table. It is the most commonly used SQL command to fetch data.

**2. Which of the following is the correct SQL statement to update a record in a table?**

A.UPDATE table_name SET column1=value1 WHERE condition;
B.CHANGE table_name SET column1=value1 WHERE condition;
C.MODIFY table_name SET column1=value1 WHERE condition;
D.ALTER table_name SET column1=value1 WHERE condition;

**Answer:** A. UPDATE table_name SET column1=value1 WHERE condition;

**Explanation:** The UPDATE statement is used to modify existing records in a table based on a specified condition (WHERE clause).

**3. What is the purpose of the WHERE clause in SQL?**

A. To specify the table to be updated
B. To filter records based on a condition
C. To sort the result set
D. To create a new database

**Answer:** B. To filter records based on a condition

**Explanation:** The WHERE clause is used to filter records based on specific conditions (e.g., WHERE age > 30).

**4. Which SQL command is used to delete a record from a table?**

A. REMOVE
B. DELETE
C. REMOVE RECORD
D. DROP

**Answer:** B. DELETE

**Explanation:** The DELETE statement is used to remove one or more records from a table based on a specified condition. It does not remove the table itself.

**5. Which of the following SQL commands is used to create a new table?**

A. CREATE TABLE
B. MAKE TABLE
C. NEW TABLE
D. ADD TABLE

**Answer:** A. CREATE TABLE

**Explanation:** The CREATE TABLE statement is used to define a new table along with its columns and data types.

**6. What does the ALTER TABLE command do in SQL?**

A. Deletes a table
B. Modifies the structure of a table
C. Creates a new table
D. Renames a table

**Answer:** B) Modifies the structure of a table

**Explanation:** The ALTER TABLE command is used to add, modify, or delete columns in an existing table.

**7. Which keyword is used to sort the result set in SQL?**

A. ORDER BY
B. SORT BY
C. GROUP BY
D. FILTER BY

**Answer:** A. ORDER BY

**Explanation:** The ORDER BY keyword is used to sort the result set in either ascending or descending order.

**8. Which SQL statement is used to remove all rows from a table without deleting the table itself?**

A. DELETE
B. TRUNCATE
C. REMOVE
D. DROP

**Answer:** B. TRUNCATE

**Explanation:** The TRUNCATE statement removes all rows from a table but does not delete the table itself. It is faster than DELETE because it does not log individual row deletions.

**9. Which SQL statement is used to create a view?**

A. CREATE VIEW
B. CREATE TABLE
C. CREATE REPORT
D. VIEW TABLE

**Answer:** A. CREATE VIEW

**Explanation:** The CREATE VIEW statement is used to create a view, which is a virtual table based on the result of a SELECT query.

**10. Which of the following SQL commands is used to remove a table from a database?**

A. DELETE TABLE
B. REMOVE TABLE
C. DROP TABLE
D. ERASE TABLE

**Answer:** C. DROP TABLE

**Explanation:** The DROP TABLE statement is used to delete a table and its data from the database permanently.

**11. Which of the following aggregate functions is used to find the total sum of a numeric column in SQL?**

A. COUNT()

B. AVG()
C. SUM()
D. MAX()

**Answer:** C. SUM()

**Explanation:** The SUM() function is used to calculate the total sum of a numeric column. It only works on columns that contain numeric data types (e.g., integers, decimals).

**12. Which of the following functions returns the average value of a numeric column in SQL?**

A. COUNT()
B. MAX()
C. AVG()
D. SUM()

**Answer:** C. AVG()

**Explanation:** The AVG() function calculates the average (mean) of a numeric column by dividing the total sum of the values by the number of values.

**13. Which aggregate function is used to count the number of rows in a SQL query result?**

A. COUNT()
B. SUM()
C. GROUP_CONCAT()
D. MIN()

**Answer:** A. COUNT()

**Explanation:** The COUNT() function counts the number of rows in a table or the number of non-NULL values in a column. It is often used with the DISTINCT keyword to count unique values.

**14. Which of the following SQL aggregate functions returns the maximum value in a column?**

A. COUNT()
B. MAX()
C. MIN()
D. AVG()

**Answer:** B. MAX()

**Explanation:** The MAX() function returns the highest value in a specified column. It can be used on numeric, date, or string columns.

**15. What is the purpose of the GROUP BY clause when using aggregate functions in SQL?**

A. To filter rows based on a condition
B. To group rows that have the same values in specified columns
C. To join multiple tables
D. To order the result set in ascending or descending order

**Answer:** B. To group rows that have the same values in specified columns

**Explanation:** The GROUP BY clause is used to group rows based on one or more columns, allowing aggregate functions (like SUM(), AVG(), COUNT(), etc.) to be applied to each group. For example, you might use GROUP BY to calculate the total sales for each salesperson.

**16. Which statement is true about a composite key in a relational database?**

A. It is a key that is composed of a single column
B. It is always used as a primary key
C. It is a key that is formed by combining two or more columns
D. It is a key that does not allow NULL values

**Answer:** C. It is a key that is formed by combining two or more columns

**Explanation:** A **composite key** is a key that consists of two or more columns in a table, which together uniquely identify a record. It is used when no single column can uniquely identify a record on its own.

**17. Which SQL join returns all records from the left table and the matched records from the right table?**

A. INNER JOIN
B. LEFT JOIN
C. RIGHT JOIN
D. FULL JOIN

**Answer:** B. LEFT JOIN

**Explanation:**
A **LEFT JOIN** returns all records from the left table and matched records from the right table. If there is no match, NULLs are returned for columns from the right table.

**18. What will an INNER JOIN return?**

A. Only unmatched rows from both tables
B. All rows from both tables
C. Only matched rows from both tables
D. All rows from the left table

**Answer:** C. Only matched rows from both tables

**Explanation:**
An **INNER JOIN** returns only the rows where there is a match in both joined tables based on the given condition.

**19. Which join should be used to include all records from both tables, regardless of whether there is a match?**

A. LEFT JOIN
B. RIGHT JOIN
C. FULL OUTER JOIN
D. CROSS JOIN

**Answer:** C. FULL OUTER JOIN

**Explanation:**
A **FULL OUTER JOIN** returns all records from both tables. Where there is no match, it returns NULLs in the columns of the table without a matching row.

**20. What does a CROSS JOIN do?**

A. Matches rows using a specified condition
B. Returns only matching rows
C. Returns the Cartesian product of two tables
D. Returns only unmatched rows

**Answer: C.** Returns the Cartesian product of two tables

**Explanation:**
A **CROSS JOIN** returns the Cartesian product of the two tables—each row from the first table is paired with all rows from the second table.

**21. What type of join is typically used when combining records based on a foreign key relationship?**

**A. CROSS JOIN**
**B. INNER JOIN**
**C. FULL OUTER JOIN**
**D. NATURAL JOIN**

**Answer: B.** INNER JOIN

**Explanation:**
An **INNER JOIN** is commonly used when combining records from two tables that share a foreign key-primary key relationship, returning only matching rows.

**22.** What is the correct order of clauses in a SQL SELECT statement?
A. SELECT, WHERE, FROM, GROUP BY
B. FROM, SELECT, WHERE, GROUP BY
C. SELECT, FROM, WHERE, GROUP BY
D. SELECT, GROUP BY, WHERE, FROM
**Answer:** C
**Explanation:** The correct sequence is: SELECT → FROM → WHERE → GROUP BY → HAVING → ORDER BY.

**23.** What does SELECT * FROM employees; do?
A. Deletes all records
B. Selects all records
C. Selects all columns
D. Selects only unique records
**Answer:** C
**Explanation:** * selects **all columns** from the table.

**24.** To eliminate duplicate records, use:
A. UNIQUE
B. PRIMARY KEY
C. DISTINCT
D. ALL
**Answer:** C
**Explanation:** DISTINCT removes duplicate rows from the result set.

**25.** What does UNION do in SQL?
A. Combines results and keeps duplicates
B. Filters duplicates
C. Combines and removes duplicates
D. Subtracts one query from another
**Answer:** C
**Explanation:** UNION **combines two result sets and removes duplicates**.

**26.** Which operator includes duplicates?
A. UNION
B. UNION ALL

C. INTERSECT
D. EXCEPT

**Answer:** B
**Explanation:** UNION ALL includes all results including duplicates.

**27.** INTERSECT returns:
A. All rows from first query
B. All rows from both queries
C. Rows common to both queries
D. All unique rows
**Answer:** C
**Explanation:** INTERSECT gives **common rows** from both result sets.

**28.** EXCEPT operator returns:
A. All common rows
B. Only duplicates
C. Rows from first query not in second
D. All rows from both
**Answer:** C
**Explanation:** EXCEPT gives rows in **first query not in the second**.

**29.** A subquery is:
A. A query that runs in background
B. A query within another query
C. Always returns multiple rows
D. Always runs first
**Answer:** B
**Explanation:** A **subquery** is a query inside another query.

**30.** Subqueries are enclosed within:
A. Curly braces
B. Square brackets
C. Parentheses
D. Quotes
**Answer:** C
**Explanation:** Subqueries are enclosed in **parentheses** ().

**31.** Which keyword is used to compare a subquery returning multiple values?
A. IN
B. =

C. <>
D. BETWEEN
**Answer:** A
**Explanation:** Use IN when subquery returns **multiple values**.

**32.** What is a correlated subquery?
A. Independent of outer query
B. Executes only once
C. Refers to column from outer query
D. Returns only NULL
**Answer:** C
**Explanation:** A **correlated subquery** references outer query's columns.

**33.** Which of the following is not an aggregate function?
A. COUNT()
B. SUM()
C. AVG()
D. GETDATE()
**Answer:** D
**Explanation:** GETDATE() is a **date function**, not an aggregate function.

**34.** What does COUNT(*) do?
A. Counts only non-null values
B. Counts unique rows
C. Counts all rows
D. Counts columns
**Answer:** C
**Explanation:** COUNT(*) returns the **total number of rows**, including nulls.

**35.** Which function returns the average value?
A. SUM()
B. COUNT()
C. AVG()
D. MAX()
**Answer:** C
**Explanation:** AVG() returns the **average** of values.

**36.** What does NULL represent?
A. 0
B. Empty string
C. Unknown or missing value

D. False
**Answer:** C
**Explanation:** NULL means **unknown or missing** data.

**37.** Which comparison is valid for NULL?
A. = NULL
B. != NULL
C. IS NULL
D. <> NULL
**Answer:** C
**Explanation:** Use IS NULL to check for NULL values.

**38.** What will NULL + 10 return?
A. 10
B. NULL
C. 0
D. Error
**Answer:** B
**Explanation:** Any operation with NULL results in **NULL**.

**39.** Derived relations are created using:
A. JOIN
B. SELECT clause
C. Subquery in FROM clause
D. Temporary table
**Answer:** C
**Explanation:** A **derived relation** is a subquery used as a temporary table in the FROM clause.

**40.** A derived table must have:
A. A WHERE clause
B. An alias
C. A primary key
D. A HAVING clause
**Answer:** B
**Explanation:** A derived table **must be given an alias** to reference it.

**41.** Which command is used to insert data into a table?
A. UPDATE
B. INSERT
C. MODIFY
D. APPEND

**Answer:** B
**Explanation:** INSERT is used to add new records.

**42.** What does this command do? DELETE FROM students;
A. Deletes the table
B. Deletes the schema
C. Deletes all records
D. Deletes one record
**Answer:** C
**Explanation:** DELETE FROM table_name; removes **all rows** from the table.

**43.** Which command modifies existing data?
A. ALTER
B. CREATE
C. UPDATE
D. DROP
**Answer:** C
**Explanation:** UPDATE modifies **existing data** in a table.

**44.** Which clause should be used with UPDATE to prevent affecting all rows?
A. GROUP BY
B. ORDER BY
C. WHERE
D. HAVING
**Answer:** C
**Explanation:** WHERE is used to target specific rows during updates.

**45.** What is the result of a LEFT JOIN?
A. Only matched rows
B. All rows from right table
C. All rows from left table + matched from right
D. All unmatched rows only
**Answer:** C
**Explanation:** LEFT JOIN returns **all rows from the left**, and matched rows from the right.

**46.** Which of these is not a valid JOIN type in SQL?
A. INNER
B. CROSS
C. OUTER
D. CIRCULAR

**Answer:** D
**Explanation:** CIRCULAR JOIN does not exist in SQL.

**47.** To update a table using data from another table, you typically use:
A. JOIN
B. GROUP BY
C. ROLLUP
D. HAVING
**Answer:** A
**Explanation:** You use a JOIN to update data using another table's values.

**48.** Which of these SQL commands affects **both data and structure**?
A. SELECT
B. UPDATE
C. DELETE
D. ALTER
**Answer:** D
**Explanation:** ALTER modifies **table structure**, like adding/removing columns.

49. Consider two tables: Students (columns: StudentID, Name, MajorID) and Majors (columns: MajorID, MajorName). You need to retrieve the names of all students and their corresponding major names. However, if a student hasn't been assigned a major yet (i.e., MajorID is NULL in the Students table), you should still include their name in the result, but the MajorName should be displayed as 'Undeclared'. Which type of JOIN and accompanying logic would best achieve this?

(a) SELECT S.Name, M.MajorName FROM Students S INNER JOIN Majors M ON S.MajorID = M.MajorID; (b) SELECT S.Name, COALESCE(M.MajorName, 'Undeclared') FROM Students S LEFT JOIN Majors M ON S.MajorID = M.MajorID; (c) SELECT S.Name, CASE WHEN S.MajorID IS NULL THEN 'Undeclared' ELSE M.MajorName END FROM Students S JOIN Majors M ON S.MajorID = M.MajorID; (d) SELECT S.Name, M.MajorName FROM Students S FULL OUTER JOIN Majors M ON S.MajorID = M.MajorID WHERE S.MajorID IS NULL OR S.MajorID IS NOT NULL;

**Answer:** (b) SELECT S.Name, COALESCE(M.MajorName, 'Undeclared') FROM Students S LEFT JOIN Majors M ON S.MajorID = M.MajorID;

**Explanation:** This question tests your understanding of LEFT JOIN and handling NULL values.

- A LEFT JOIN ensures that all rows from the Students table are included in the result. When there's no matching MajorID in the Majors table, the columns from Majors will have NULL values.
- The COALESCE() function is crucial here. It checks the first argument (M.MajorName). If it's not NULL (meaning there's a matching major), it returns the major name. If it is NULL (meaning the student has no assigned major), it returns the second argument, 'Undeclared'.

- Option (a) uses an INNER JOIN, which would exclude students with a NULL MajorID.
- Option (c) uses an INNER JOIN as well, so students with NULL MajorID would be excluded before the CASE statement could apply.
- Option (d) uses a FULL OUTER JOIN, which would include all students and all majors. The WHERE clause doesn't correctly handle the 'Undeclared' requirement.

**50.** You have two tables: Events (columns: EventID, EventName, StartDate, EndDate) and Attendees (columns: AttendeeID, EventID, RegistrationDate). You need to find the names of all events and the total number of attendees registered for each event that started in the year 2024. Events with no attendees started in 2024 should also be listed with an attendee count of 0. Which of the following SQL structures is the most appropriate?

(a) SELECT E.EventName, COUNT(A.AttendeeID) FROM Events E INNER JOIN Attendees A ON E.EventID = A.EventID WHERE YEAR(E.StartDate) = 2024 GROUP BY E.EventName; (b) SELECT E.EventName, COUNT(A.AttendeeID) FROM Events E LEFT JOIN Attendees A ON E.EventID = A.EventID WHERE YEAR(E.StartDate) = 2024 GROUP BY E.EventName; (c) SELECT E.EventName, COUNT(A.AttendeeID) FROM Attendees A RIGHT JOIN Events E ON A.EventID = E.EventID WHERE YEAR(E.StartDate) = 2024 GROUP BY E.EventName; (d) SELECT E.EventName, COUNT(*) FROM Events E LEFT JOIN Attendees A ON E.EventID = A.EventID WHERE YEAR(E.StartDate) = 2024 GROUP BY E.EventName;

**Answer:** (b) SELECT E.EventName, COUNT(A.AttendeeID) FROM Events E LEFT JOIN Attendees A ON E.EventID = A.EventID WHERE YEAR(E.StartDate) = 2024 GROUP BY E.EventName;

**Explanation:** This question tests your understanding of LEFT JOIN in conjunction with aggregate functions and filtering.

- We need all events that started in 2024, so the WHERE YEAR(E.StartDate) = 2024 clause is necessary and should be applied to the Events table (the left table in our desired join).
- A LEFT JOIN from Events to Attendees ensures that all events meeting the year criteria are included, even if they have no attendees.
- COUNT(A.AttendeeID) counts the number of non-NULL AttendeeID values. When an event has no attendees, the columns from the Attendees table will be NULL for that event, resulting in a count of 0.
- Option (a) uses an INNER JOIN, which would exclude events from 2024 that have no attendees.
- Option (c) uses a RIGHT JOIN. While it might technically work if the WHERE clause is applied correctly, it's less intuitive as we want all events matching the criteria, making Events the logical left table.
- Option (d) uses COUNT(*), which would count all rows resulting from the LEFT JOIN. For events with no attendees, it would count one row (with NULLs for attendee columns), resulting in a count of 1, which is incorrect. We need to specifically count the attendees.

**Unit III**

Database Design Functional dependencies, Normal forms, First, second, and third normal forms, BCNF, Multi-valued dependencies and Fourth Normal form, Join Dependencies and Fifth Normal form.

**Database Design** is the process of organizing data efficiently to avoid redundancy and ensure data integrity. It typically involves:

- Choosing appropriate data types
- Identifying relationships between data
- Applying **normalization** to reduce redundancy

The design moves from a **conceptual model** (like ER diagrams) to a **logical model** (like relational schemas), and then to a **physical model** (actual implementation).

A **functional dependency** occurs when one attribute uniquely determines another attribute.

**Notation:**
If A → B, then for any two rows with the same value of A, the value of B will also be the same.

**Example:**
If student_id → student_name, then knowing a student_id lets us determine the student_name.

Functional dependencies help identify:

- **Primary keys**
- **Normalization rules**
- **Data redundancy**

**Normalization** is the step-by-step process of organizing a database to:

- Remove redundancy
- Prevent anomalies (update, insert, delete anomalies)
- Improve data integrity

Each **normal form (NF)** builds on the previous one.

**First Normal Form (1NF)**

A relation is in **1NF** if:

- All attributes are **atomic** (no multiple values or arrays in a single cell)
- No repeating groups

**Example Violation:**
A column storing multiple phone numbers like: "1234, 5678" □
□ **Fix:** Split into separate rows or a new table.

**Second Normal Form (2NF)**

A relation is in **2NF** if:

1. It is already in **1NF**
2. All **non-prime attributes** are **fully functionally dependent** on the **entire primary key**

Applies to **composite keys**

**Violation Example:**
If course_id, student_id → grade and student_name depends only on student_id, then it violates 2NF.

**Third Normal Form (3NF)**

A relation is in **3NF** if:

1. It is in **2NF**
2. No **transitive dependencies** (i.e., non-prime attributes should not depend on other non-prime attributes)

**Example:**
If student_id → department_id and department_id → department_name, then student_id → department_name is a **transitive dependency**.

□ Fix: Break into two tables.

**Boyce-Codd Normal Form (BCNF)**

A stricter version of 3NF.

A relation is in **BCNF** if:

- For every functional dependency X → Y, X must be a **super key**

**When is BCNF needed?**
When **non-trivial FDs** exist where the determinant is **not a superkey**.

**Example:**
If course → instructor and multiple courses can have same instructor, but instructor is not a key, it violates BCNF.

**Multi-Valued Dependencies (MVDs)**

An **MVD** exists when one attribute in a table determines multiple independent values of another attribute.

**Example:**
If a student can have multiple phone numbers **and** multiple hobbies, and they are **independent** of each other:

Student_ID →→ Phone
Student_ID →→ Hobby

This can cause unnecessary combinations and redundancy.

**Fourth Normal Form (4NF)**

A relation is in **4NF** if:

1. It is in **BCNF**
2. It has **no multi-valued dependencies (MVDs)**

**Violation Example:**
If a student has multiple hobbies **and** multiple phone numbers in the same table → MVDs → violates 4NF

 Fix: Split into two separate tables for phone and hobby.

**Join Dependencies**

A **join dependency (JD)** exists when a relation can be split into **two or more relations** and can be perfectly **reconstructed** using a **natural join**.

General form:
R1, R2, ..., Rn is a join dependency of R if:
R = R1 ⋈ R2 ⋈ ... ⋈ Rn

**Fifth Normal Form (5NF) / Project-Join Normal Form (PJNF)**

A relation is in **5NF** if:

1. It is in **4NF**
2. It **cannot be decomposed** into smaller relations without losing data
3. It has **no join dependencies** that are not implied by candidate keys

Very rare in practice. Most designs stop at **3NF or BCNF**.

**Example (complex):** If a vendor supplies multiple parts to multiple projects and each combination is valid independently, decomposing into three relations may lose some info unless in 5NF.

**1. What does a functional dependency A → B signify in a relational database?**

A. B is a primary key
B. B uniquely determines A
C. A uniquely determines B
D. A and B are always equal

**Answer: C.** A uniquely determines B

**Explanation:**
A → B means that for each value of **A**, there is exactly one corresponding value of **B**. So, **A functionally determines B**.

**2. Which normal form removes partial dependencies?**

A. First Normal Form (1NF)
B. Second Normal Form (2NF)
C. Third Normal Form (3NF)
D. Boyce-Codd Normal Form (BCNF)

**Answer: B.** Second Normal Form (2NF)

**Explanation:**
**2NF** eliminates **partial dependencies**, i.e., when a non-prime attribute is dependent on only a part of a candidate key.

**3. Which of the following is true about Third Normal Form (3NF)?**

A. It allows transitive dependencies
B. It ensures no repeating groups
C. It removes transitive dependencies
D. It allows partial dependencies

**Answer: C.** It removes transitive dependencies

**Explanation:**
**3NF** removes **transitive dependencies**, where a non-prime attribute is functionally dependent on another non-prime attribute.

**4. If a relation is in 2NF and has no transitive dependencies, it is in which normal form?**

A. 1NF
B. 2NF
C. 3NF
D. BCNF

**Answer: C. 3NF**

**Explanation:**
A relation in **2NF** with **no transitive dependencies** satisfies the condition for **Third Normal Form (3NF)**.

**5. Which of the following best describes a transitive dependency?**

A. A → B and B → C implies A → C
B. A → A
C. A → B and B → A
D. A → B, B → C, but not A → C

**Answer: A.** A → B and B → C implies A → C

**Explanation:**
A **transitive dependency** occurs when one attribute depends on another through a third attribute, like **A → B** and **B → C**, which implies **A → C**.

**6.** A functional dependency is a relationship between:
A. Tables
B. Rows
C. Columns
D. Keys
**Answer:** C
**Explanation:** A functional dependency is a relationship between attributes (columns), where one determines the other.

**7.** If A → B and B → C, then A → C. This is called:
A. Reflexive rule
B. Augmentation rule
C. Transitive rule
D. Pseudo-transitive rule

**Answer:** C
**Explanation:** This is the **transitive rule** of functional dependency.

**8.** Which of the following is a valid functional dependency?
A. A ← B
B. A → B
C. A ↔ B
D. A & B
**Answer:** B
**Explanation:** Functional dependencies are written as A → B.

**9.** If X → Y, and Y is a subset of Z, then:
A. X → Z
B. X → YZ
C. Z → Y
D. No conclusion
**Answer:** D
**Explanation:** We cannot conclude anything about X → Z just from X → Y.

**10.** In a functional dependency, the left-hand side should be:
A. A primary key
B. A candidate key
C. A super key
D. A set of attributes
**Answer:** D
**Explanation:** The determinant can be any set of attributes, not necessarily a key.

**11.** Which of the following violates 1NF?
A. Single value in each cell
B. Multiple phone numbers in one cell
C. Primary key exists
D. Unique values in a column
**Answer:** B
**Explanation:** 1NF requires atomic (indivisible) values.

**12.** Which condition is essential for 1NF?
A. Transitive dependency
B. Atomicity
C. Candidate keys
D. Functional dependency

**Answer:** B
**Explanation:** 1NF requires all attributes to hold atomic values.

**13.** A table with multi-valued attributes is:
A. In 1NF
B. In 2NF
C. Not in 1NF
D. In BCNF
**Answer:** C
**Explanation:** Multi-valued attributes violate 1NF.

**14.** A student table with multiple hobbies in a single column breaks which normal form?
A. 1NF
B. 2NF
C. 3NF
D. BCNF
**Answer:** A
**Explanation:** 1NF is violated if columns contain multiple values.

**15.** Which of the following is NOT true about 1NF?
A. No repeating groups
B. Only atomic values
C. No transitive dependencies
D. No multi-valued attributes
**Answer:** C
**Explanation:** Transitive dependency is handled in 3NF, not 1NF.

**16.** 2NF removes:
A. Transitive dependencies
B. Multi-valued attributes
C. Partial dependencies
D. Functional dependencies
**Answer:** C
**Explanation:** 2NF removes **partial dependencies**.

**17.** Which is required for 2NF?
A. 1NF + no partial dependencies
B. 3NF + atomicity
C. 1NF + no multi-valued dependencies
D. 2NF + transitive dependencies

**Answer:** A
**Explanation:** 2NF = 1NF + no partial dependencies.

**18.** Partial dependency means:
A. Attribute depends on a part of composite key
B. Attribute depends on another attribute
C. Attribute depends on whole key
D. Attribute depends on primary key only
**Answer:** A
**Explanation:** Partial dependency = dependency on part of a composite key.

**19.** Which violates 2NF?
A. A single attribute key
B. Partial dependency on composite key
C. Transitive dependency
D. Atomic value in each field
**Answer:** B
**Explanation:** Partial dependencies are the main target of 2NF.

**20.** If a table is in 1NF and has only single-attribute primary key, it is:
A. Not in 2NF
B. Always in 2NF
C. In 3NF
D. Needs decomposition
**Answer:** B
**Explanation:** Partial dependencies occur only with composite keys. So, single-attribute key = 2NF.

**21.** 3NF removes:
A. Atomicity
B. Transitive dependencies
C. Partial dependencies
D. Functional dependencies
**Answer:** B
**Explanation:** 3NF eliminates transitive dependencies.

**22.** A table in 2NF, with transitive dependencies, is:
A. Also in 3NF
B. Not in 3NF
C. In BCNF
D. In 4NF

**Answer:** B
**Explanation:** Transitive dependency violates 3NF.

**23.** Transitive dependency occurs when:
A. A → B and B → C
B. A → B and A → C
C. A → B and B → A
D. A → A
**Answer:** A
**Explanation:** A → B → C implies A → C transitively.

**24.** In 3NF, every non-prime attribute must:
A. Depend only on candidate keys
B. Depend on foreign keys
C. Be multi-valued
D. Be a key
**Answer:** A
**Explanation:** 3NF = no transitive dependency on non-prime attributes.

**25.** Which condition makes a schema not in 3NF?
A. Atomic values
B. Dependency of non-key on key
C. Dependency of non-key on non-key
D. Full functional dependency
**Answer:** C
**Explanation:** Transitive dependency between non-keys violates 3NF.

**26.** A relation in BCNF must also be in:
A. 1NF only
B. 2NF only
C. 3NF
D. None of the above
**Answer:** C
**Explanation:** BCNF is a stricter version of 3NF.

**27.** BCNF removes:
A. Multi-valued dependency
B. Join dependency
C. Anomalies caused by non-superkey FDs
D. Partial dependency
**Answer:** C
**Explanation:** In BCNF, every determinant must be a superkey.

**28.** A table with overlapping candidate keys and FDs not on keys violates:
A. 2NF
B. 3NF
C. BCNF
D. 4NF
**Answer:** C
**Explanation:** FDs where determinant is not a super key → violates BCNF.

**29.** BCNF violation is possible even if relation is in:
A. 1NF
B. 2NF
C. 3NF
D. 5NF
**Answer:** C
**Explanation:** A relation can be in 3NF but not in BCNF.

**30.** Which of the following is always in BCNF?
A. Relation with only one candidate key
B. Relation with atomic values
C. Relation with no functional dependencies
D. Relation with no multi-valued dependencies
**Answer:** C
**Explanation:** No FDs means BCNF is automatically satisfied.

**31.** A multi-valued dependency is denoted by:
A. →
B. ←
C. →→
D. ⇔
**Answer:** C
**Explanation:** MVDs are denoted using →→.

**32.** MVDs occur when:
A. Attribute is partially dependent
B. Attribute is transitively dependent
C. One attribute determines a set of values independently
D. Data is repeated
**Answer:** C
**Explanation:** MVD = independent multi-values per key.

**33.** Which normal form removes MVDs?
A. 2NF
B. 3NF
C. BCNF
D. 4NF
**Answer:** D
**Explanation:** 4NF removes multi-valued dependencies.

**34.** MVDs often lead to:
A. Transitive dependencies
B. Inconsistent updates
C. Data loss
D. Partial dependencies
**Answer:** B
**Explanation:** MVDs cause **data redundancy and update anomalies**.

**35.** If a relation has more than one independent multi-valued dependency, it should be:
A. Converted to 2NF
B. Split into separate tables
C. Deleted
D. Replaced with a view
**Answer:** B
**Explanation:** Decompose to remove MVDs → 4NF.

**36.** A relation is in 4NF if it is in BCNF and has:
A. No partial dependency
B. No MVDs
C. No join dependency
D. Composite key
**Answer:** B
**Explanation:** 4NF = BCNF + No MVDs.

**37.** Which dependency is targeted in 5NF?
A. Functional
B. Join
C. Multi-valued
D. Partial
**Answer:** B
**Explanation:** 5NF removes join dependencies.

**38.** Join dependency occurs when:
A. A table can be recombined from its projections
B. A column is not atomic
C. There is a transitive dependency
D. Two foreign keys are used
**Answer:** A
**Explanation:** JD = a table can be **losslessly reconstructed** from joins of subsets.

**39.** 5NF is also known as:
A. Transitive normal form
B. BCNF
C. Project-Join Normal Form
D. Multi-value normal form
**Answer:** C
**Explanation:** 5NF = **Project-Join Normal Form (PJNF)**.

**40.** 5NF is needed when:
A. Transitive dependencies exist
B. Partial dependencies exist
C. MVDs exist
D. Complex joins may cause anomalies
**Answer:** D
**Explanation:** 5NF removes **join anomalies** from complex decompositions.

**41.** The main goal of normalization is to:
A. Improve performance
B. Reduce redundancy
C. Increase storage
D. Encrypt data
**Answer:** B
**Explanation:** Normalization removes redundancy and anomalies.

**42.** Which is the highest normal form commonly used in practice?
A. 2NF
B. 3NF
C. 5NF
D. 4NF
**Answer:** B
**Explanation:** Most databases are normalized up to **3NF or BCNF**.

**43.** Which normal form focuses on atomicity?
A. 2NF
B. 3NF
C. 1NF
D. 4NF
**Answer:** C
**Explanation:** 1NF requires atomic values.

**44.** Which normal form removes both partial and transitive dependencies?
A. 3NF
B. 1NF
C. 2NF
D. BCNF
**Answer:** A
**Explanation:** 3NF removes both **partial and transitive** dependencies.

**45.** The primary key is used to:
A. Store images
B. Identify records uniquely
C. Encrypt columns
D. Perform joins only
**Answer:** B
**Explanation:** A primary key **uniquely identifies** each row.

**46.** A candidate key is:
A. Always primary key
B. Never used
C. A potential primary key
D. A foreign key
**Answer:** C
**Explanation:** A **candidate key** is a set of attributes that could serve as the primary key.

**47.** Decomposition should always be:
A. Lossy
B. Lossless
C. Temporary
D. Avoided
**Answer:** B
**Explanation:** A **lossless decomposition** preserves all original data.

**48.** Which normal form deals with redundancy due to join dependencies?
A. 3NF
B. 4NF
C. 5NF
D. BCNF
**Answer:** C
**Explanation:** 5NF removes redundancy due to join dependencies.

**49.** Which of the following anomalies are reduced by normalization?
A. Update anomaly
B. Insertion anomaly
C. Deletion anomaly
D. All of the above
**Answer:** D
**Explanation:** Normalization removes all major anomalies.

**50.** A relation in 3NF and BCNF may still need further normalization if it has:
A. Partial dependency
B. MVDs
C. No key
D. Too many joins
**Answer:** B
**Explanation:** MVDs are not handled by 3NF or BCNF → require 4NF.

**Unit IV**

Transaction, Query Processing Transaction and system concepts: transaction states, ACID properties of transactions, Transaction Management concurrent execution schedules and Recoverability, Serializability. Concurrency Control: Concurrency Control Techniques: Two phase Locking Techniques for Concurrency Control; Time stamping in Concurrency control.

***1. A transaction is a:***

A. Set of SQL queries
B. Logical unit of work
C. Concurrent process
D. Set of indexes

**Answer:** B
**Explanation:** A transaction is a logical unit of work that must be either entirely completed or aborted.

***2. Which of the following is NOT a property of a transaction?***

A. Atomicity
B. Consistency
C. Divisibility
D. Durability

**Answer:** C
**Explanation:** The correct ACID properties are Atomicity, Consistency, Isolation, and Durability. Divisibility is not part of ACID.

***3. In the transaction states, what does the 'active' state mean?***

A. Transaction has finished
B. Transaction is in execution
C. Transaction is waiting
D. Transaction has been committed

**Answer:** B
**Explanation:** The active state indicates the transaction is currently executing its operations.

***4. Which state follows the 'partially committed' state in a successful transaction?***

A. Failed
B. Committed
C. Aborted
D. Active

**Answer:** B
**Explanation:** If no failure occurs, a transaction moves from partially committed to committed.

***5. What does the 'durability' property ensure in a transaction?***

A. Changes are temporary
B. Changes are lost on failure
C. Changes are permanent even after system failure
D. Changes are rolled back

**Answer:** C
**Explanation:** Durability ensures that once a transaction is committed, its effects persist even after a crash.

**6. If a transaction fails after modification but before commit, the state is:**

A. Committed
B. Failed
C. Aborted
D. Active

**Answer:** C
**Explanation:** The transaction is rolled back to ensure atomicity, so it is considered aborted.

***7. Which property ensures that concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially?***

A. Consistency
B. Isolation
C. Atomicity
D. Serializability

**Answer:** D
**Explanation:** Serializability ensures the correctness of concurrent transaction schedules.

***8. A transaction that reads uncommitted changes made by another transaction is said to be experiencing a:***

A. Dirty read
B. Phantom read
C. Non-repeatable read
D. Lost update

**Answer:** A
**Explanation:** A dirty read occurs when a transaction reads data that has not been committed by another transaction.

***9. Which of the following ensures consistency in concurrent transaction executions?***

A. Buffering
B. Paging
C. Concurrency control
D. Indexing

**Answer:** C
**Explanation:** Concurrency control mechanisms ensure consistent and correct results in concurrent transactions.

***10. In which schedule can operations of different transactions be interleaved, but the final outcome is the same as if transactions were executed one by one?***

A. Non-serial
B. Serializable
C. Irrecoverable
D. Strict

**Answer:** B
**Explanation:** Serializable schedules preserve the consistency of the database.

***11. Which of the following is NOT a valid schedule in transaction execution?***

A. Serializable
B. Recoverable
C. Irrecoverable
D. Non-deterministic

**Answer:** D
**Explanation:** Non-deterministic is not a standard classification of transaction schedules.

***12. A schedule in which a transaction reads a value written by an uncommitted transaction is:***

A. Strict
B. Recoverable
C. Irrecoverable
D. Serializable

**Answer:** C
**Explanation:** Irrecoverable schedules risk reading uncommitted data and crashing if the writer aborts.

***13. What ensures that once a transaction reads or modifies data, no other transaction can access it until the first completes?***

A. Atomicity
B. Serializability
C. Isolation
D. Consistency

**Answer:** C
**Explanation:** Isolation prevents other transactions from accessing intermediate states of a transaction.

***14. Two-phase locking protocol is used to:***

A. Detect deadlock
B. Prevent lost updates
C. Ensure serializability
D. Recover from failure

**Answer:** C
**Explanation:** 2PL guarantees conflict-serializability by dividing execution into growing and shrinking phases.

***15. In 2PL, once a transaction releases a lock, it cannot:***

A. Abort
B. Request a new lock
C. Commit
D. Read data

**Answer:** B
**Explanation:** In the shrinking phase, no new locks can be acquired.

***16. A transaction using 2PL protocol that holds all locks until it commits is using:***

A. Conservative 2PL
B. Rigorous 2PL
C. Strict 2PL
D. Basic 2PL

**Answer:** C
**Explanation:** Strict 2PL holds all exclusive locks until commit or abort.

***17. Which of the following can result from poor concurrency control?***

A. Data loss
B. Data inconsistency
C. Deadlock
D. All of the above

**Answer:** D
**Explanation:** Without proper control, concurrency can lead to multiple issues including deadlocks and inconsistencies.

***18. Which timestamp ordering rule is violated if a transaction reads a data item modified by a newer transaction?***

A. Read Rule
B. Write Rule
C. Timestamp Rule
D. Recoverability Rule

**Answer:** A
**Explanation:** In timestamp ordering, a transaction must read only versions committed before its timestamp.

***19. Which approach avoids cascading aborts?***

A. Irrecoverable schedule
B. Non-serializable schedule
C. Strict schedule
D. Permissive schedule

**Answer:** C
**Explanation:** Strict schedules ensure no transaction can read or write a data item modified by an uncommitted transaction.

***20. Time-stamp ordering protocol ensures:***

A. Deadlock prevention
B. Starvation
C. Non-serializability
D. File locking

**Answer:** A
**Explanation:** Time-stamp ordering prevents deadlocks by determining a serial order using timestamps.

***21. In timestamp ordering, each transaction is assigned:***

A. A unique identifier
B. A priority level
C. A system log
D. A page number

**Answer:** A
**Explanation:** Each transaction gets a unique timestamp that determines its order of execution.

***22. What does the read-timestamp (RTS) of a data item indicate?***

A. The last time it was written
B. The earliest transaction to access it
C. The largest timestamp of any transaction that successfully read the item
D. The number of reads performed

**Answer:** C
**Explanation:** RTS records the most recent successful read timestamp.

***23. What is the purpose of write-timestamp (WTS)?***

A. Log write operations
B. Denote the size of the transaction
C. Keep track of which transaction last wrote to the item
D. Count the number of writes

**Answer:** C
**Explanation:** WTS shows the timestamp of the most recent write to the data item.

***24. What happens if a transaction tries to read a data item with a WTS greater than its own timestamp?***

A. It's allowed
B. It waits
C. It is rolled back
D. It commits

**Answer:** C
**Explanation:** This would violate serializability, so the transaction must be aborted and restarted.

***25. Which technique avoids deadlock entirely?***

A. Wait-die
B. Wound-wait

C. Timestamp ordering
D. Strict 2PL

**Answer:** C
**Explanation:** Timestamp ordering does not use locks, so deadlock cannot occur.

***26. Which protocol can result in starvation?***

A. Basic 2PL
B. Strict 2PL
C. Timestamp ordering
D. Conservative 2PL

**Answer:** C
**Explanation:** Repeated aborts due to timestamp conflicts can cause starvation.

***27. What does Conservative 2PL do before transaction starts?***

A. Locks nothing
B. Takes all locks it needs
C. Requests no locks
D. Unlocks all data

**Answer:** B
**Explanation:** Conservative 2PL avoids deadlock by locking all required data items in advance.

***28. In wound-wait scheme, what happens if an older transaction requests a lock held by a younger one?***

A. Older waits
B. Younger rolls back
C. Deadlock occurs
D. Both continue

**Answer:** B
**Explanation:** In wound-wait, the younger transaction is aborted to avoid deadlock.

***29. Which of the following is* not *a locking protocol?***

A. Two-phase locking
B. Strict locking
C. Timestamp ordering
D. Conservative locking

**Answer:** C
**Explanation:** Timestamp ordering does not rely on locks.

***30. Which concurrency control protocol may cause cascading rollbacks?***

A. Strict 2PL
B. Basic 2PL
C. Conservative 2PL
D. Timestamp ordering

**Answer:** B
**Explanation:** Basic 2PL may allow reads of uncommitted data, leading to cascading rollbacks.

***31. A recoverable schedule ensures:***

A. Transactions can't abort
B. No transaction reads from an uncommitted one
C. All transactions are serial
D. All transactions are deadlock-free

**Answer:** B
**Explanation:** Recoverable schedules prevent transactions from reading uncommitted changes.

***32. In a cascadeless schedule:***

A. Transactions can read uncommitted data
B. Only committed data is read
C. Transactions execute in parallel
D. Deadlocks are allowed

**Answer:** B
**Explanation:** Cascadeless schedules ensure no transaction reads data from another uncommitted transaction.

***33. What is the main goal of concurrency control?***

A. Increase speed
B. Prevent deadlock
C. Maintain consistency
D. Save memory

**Answer:** C
**Explanation:** Concurrency control ensures database consistency with simultaneous transactions.

***34. What is a conflict in transaction schedules?***

A. Two operations access the same data item
B. Two operations access different items
C. Reading from a committed item
D. Writing only once

**Answer:** A
**Explanation:** A conflict occurs when two operations on the same item are not commutative (e.g., read/write or write/write).

***35. Which of the following schedules is guaranteed to be serializable?***

A. Irrecoverable
B. Strict
C. Conflict-serializable
D. Non-serializable

**Answer:** C
**Explanation:** Conflict-serializability guarantees equivalence to some serial schedule.

***36. Serial schedules:***

A. Allow concurrent reads
B. Execute transactions one after another
C. Cannot maintain consistency
D. Are always slower

**Answer:** B
**Explanation:** Serial schedules execute one transaction completely before starting another.

***37. In conflict-serializability, operations are reordered to:***

A. Match disk speed
B. Improve indexing
C. Match a serial schedule
D. Create deadlocks

**Answer:** C
**Explanation:** Conflict-serializable schedules are equivalent to some serial execution in terms of conflicting operations.

***38. Dirty read is possible in which isolation level?***

A. Read committed
B. Serializable
C. Repeatable read
D. Read uncommitted

**Answer:** D
**Explanation:** Read uncommitted allows transactions to read uncommitted changes from others.

***39. What is the effect of locking at a high granularity (e.g., entire database)?***

A. Increases concurrency
B. Increases deadlocks
C. Reduces concurrency
D. Has no effect

**Answer:** C
**Explanation:** High granularity locking reduces the number of concurrent transactions.

***40. Lock escalation involves:***

A. Breaking a transaction
B. Changing multiple fine-grained locks to a coarser one
C. Committing automatically
D. Reducing lock time

**Answer:** B
**Explanation:** Lock escalation reduces overhead by replacing many fine locks with a larger coarse lock.

***41. What happens in a deadlock?***

A. One transaction runs forever
B. Transactions wait on each other in a cycle
C. Transactions restart automatically
D. Data is overwritten

**Answer:** B
**Explanation:** Deadlocks occur when transactions form a cycle of waiting for resources held by each other.

***42. Which of the following can be used to detect deadlocks?***

A. Transaction log
B. Wait-for graph
C. Recovery manager
D. Isolation protocol

**Answer:** B
**Explanation:** Wait-for graphs help identify cycles, which indicate deadlocks.

***43. Which schedule allows recovery even if a transaction fails?***

A. Recoverable schedule
B. Irrecoverable schedule
C. Strict schedule
D. Non-serializable schedule

**Answer:** A
**Explanation:** Recoverable schedules ensure a transaction only commits if the ones it depends on have committed.

***44. Phantom reads occur when:***

A. A read reads uncommitted data
B. New rows are added between reads
C. A transaction is aborted
D. Two reads return the same result

**Answer:** B
**Explanation:** Phantom reads happen when new rows satisfy a condition between two reads of the same query.

***45. Which isolation level prevents phantom reads?***

A. Read committed
B. Repeatable read
C. Serializable
D. Read uncommitted

**Answer:** C
**Explanation:** Only the serializable level can prevent phantom reads.

***46. Which of the following helps in maintaining database consistency during failure?***

A. Indexing
B. Caching
C. Logging
D. Pipelining

**Answer:** C
**Explanation:** Logging ensures that the database can be recovered after crashes.

***47. Write-ahead logging ensures:***

A. Log is updated after write
B. Log is written before data
C. Logs are not needed
D. Transactions can skip logs

**Answer:** B
**Explanation:** WAL ensures logs are written before the actual data write to maintain durability.

***48. What is ARIES used for?***

A. Indexing
B. Deadlock resolution
C. Concurrency control
D. Recovery management

**Answer:** D
**Explanation:** ARIES (Algorithm for Recovery and Isolation Exploiting Semantics) is used for recovery in DBMS.

***49. In concurrency control, starvation means:***

A. A transaction is aborted
B. A transaction never gets resources
C. System fails
D. Transactions are all blocked

**Answer:** B
**Explanation:** Starvation occurs when a transaction is perpetually denied access due to other transactions being prioritized.

***50. To prevent starvation, systems can implement:***

A. Fair scheduling
B. Timestamp ordering
C. Deadlock prevention
D. Lock escalations

**Answer:** A
**Explanation:** Fair scheduling policies ensure all transactions get access eventually, preventing starvation.