

## **Practical 1: Defining Schema**

### **1. What is a schema in DBMS?**

A schema is a logical structure of the database that defines how data is organized into tables, columns, and relationships. It acts as a blueprint for the database.

### **2. What are the benefits of schema design?**

**Data Integrity:** Enforces constraints like primary keys.

**Efficiency:** Optimizes query performance.

**Scalability:** Supports future growth.

**Consistency:** Reduces redundancy using normalization.

### **3. What is the difference between schema and instance?**

**Schema:** The design or structure of the database (static).

**Instance:** The actual data stored in the database at a particular time (dynamic).

### **4. What are primary and foreign keys?**

**Primary Key:** A unique identifier for rows in a table.

**Foreign Key:** A column in one table referencing the primary key of another to establish relationships.

5. What is normalization, and why is it necessary?

Normalization organizes data to reduce redundancy and improve integrity by dividing it into smaller tables with meaningful relationships.

---

## **Practical 2: Creating Tables and Data Constraints**

6. What is the CREATE TABLE command in SQL?

It is used to define a new table in the database. Example:

```
CREATE TABLE Users (  
    user_id INT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL  
);
```

7. Explain different types of data constraints.

NOT NULL: Ensures a column cannot have NULL values.

UNIQUE: Ensures all values in a column are unique.

PRIMARY KEY: A combination of NOT NULL and UNIQUE.

FOREIGN KEY: Establishes a relationship between two tables.

CHECK: Enforces conditions on data.

DEFAULT: Assigns a default value if none is provided.

8. How do you rename a table in SQL?

Use the ALTER TABLE command:

```
ALTER TABLE OldTableName RENAME TO NewTableName;
```

9. What is the difference between DELETE and TRUNCATE?

DELETE: Removes specific rows, can use WHERE conditions.

TRUNCATE: Removes all rows and resets auto-increment counters.

---

### **Practical 3: Grouping Data and Aggregate Functions**

10. What is the purpose of aggregate functions in SQL?

Aggregate functions like SUM, AVG, COUNT, MAX, and MIN summarize data. Example:

```
SELECT AVG(salary) FROM Employees;
```

11. Explain the GROUP BY clause.

It groups rows sharing a value in a specified column. Example:

```
SELECT department_id, COUNT(*) AS employee_count  
FROM Employees  
GROUP BY department_id;
```

12. What is the difference between HAVING and WHERE?

WHERE: Filters rows before grouping.

HAVING: Filters groups after applying aggregate functions.

---

#### **Practical 4: Sub-Queries, Joins, and Set Operations**

13. What are sub-queries in SQL?

A sub-query is a query inside another query to fetch data dynamically. Example:

```
SELECT name FROM Employees  
WHERE salary > (SELECT AVG(salary) FROM Employees);
```

14. What is a JOIN? Explain its types.

Joins retrieve data from multiple tables based on a relationship.

INNER JOIN: Returns rows with matching values in both tables.

LEFT JOIN: Returns all rows from the left table, matched rows from the right.

RIGHT JOIN: Opposite of LEFT JOIN.

FULL JOIN: Combines LEFT and RIGHT JOIN.

15. What are set operations in SQL?

UNION: Combines results of two queries without duplicates.

UNION ALL: Combines results with duplicates.

INTERSECT: Returns rows common to both queries.

EXCEPT/MINUS: Returns rows in one query but not in the other.

---

## **Practical 5: Procedures, Triggers, and Cursors**

16. What is a stored procedure?

A stored procedure is a collection of SQL statements stored in the database, executed as a single unit.  
Example:

```
CREATE PROCEDURE CalculateBonus (salary DECIMAL)
BEGIN
    SELECT salary * 0.1 AS bonus;
END;
```

17. What is a trigger?

A trigger is an automated SQL block executed in response to an event like INSERT, UPDATE, or DELETE.

18. What is a cursor? When is it used?

A cursor allows row-by-row processing of a query's result set, useful for operations that require detailed data manipulation.

---

## **Practical 6: Application Design**

19. How do you design a database for applications?

Identify entities (e.g., Users, Orders).

Create tables and define relationships.

Normalize data.

Implement constraints for integrity.

20. What is the importance of foreign keys in application databases?

They maintain data consistency by ensuring that a referenced value exists in another table.

---

### **Practical 7: Forms, Reports, and Query Processing**

21. What are forms in a database?

Forms are user-friendly interfaces for entering, updating, or retrieving data.

22. What is indexing?

Indexing improves query performance by allowing the database to find rows faster. Example:

```
CREATE INDEX idx_name ON Employees(name);
```

23. How do reports help in DBMS?

Reports provide summaries, such as sales totals or employee performance, to aid decision-making.

---

### **Practical 8: Large Objects**

24. What is a BLOB? How does it differ from CLOB?

BLOB: Binary data like images or videos.

CLOB: Large text data like documents.

25. What is the use of a BFILE?

A BFILE stores references to external files without storing them in the database.

---

### **Practical 9: Data Warehousing and Association Rules**

26. What is a data warehouse?

A data warehouse stores historical data from multiple sources, optimized for reporting and analysis.

27. What is the purpose of association rule mining?

It finds relationships between items, e.g., products frequently bought together in market basket analysis.

28. Explain the Apriori algorithm.

It identifies frequent itemsets by iteratively applying a minimum support threshold.

---

### **Practical 10: Distributed Databases and Web Interfaces**

29. What is a distributed database?

A database spread across multiple locations but connected via a network.

30. How do servlets interact with databases?

Servlets use JDBC to connect to databases, execute queries, and generate dynamic responses.

31. What is the two-phase commit protocol?

A distributed transaction protocol ensuring all nodes commit a transaction or none at all

---

#### General DBMS Questions

32. What is a relational database?

A database structured to recognize relationships among stored data using tables.

33. What is ACID in DBMS?

Atomicity: All or nothing execution.

Consistency: Database remains consistent before and after a transaction.

Isolation: Transactions don't interfere with each other.

Durability: Changes persist even after system failures.

34. What is a deadlock in DBMS?

A situation where two or more transactions wait indefinitely for resources locked by each other.

35. Explain the difference between OLTP and OLAP.

OLTP (Online Transaction Processing): Handles daily operations (e.g., banking).



OLAP (Online Analytical Processing): Supports data analysis and decision-making.

---

Let me know if you want more specific questions or answers!