

Introduction to DBMS & SQL

1. What is SQL, and why is it essential in database management?

Answer: SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases. It is essential because it allows users to perform operations such as querying, updating, inserting, and deleting data, as well as defining and controlling database structure and access.

2. Explain the difference between DBMS and RDBMS.

Answer: A DBMS (Database Management System) manages data without enforcing relationships among tables. An RDBMS (Relational Database Management System) manages data with enforced relationships, typically using primary and foreign keys, and follows the relational model introduced by E.F. Codd.

3. Describe the role of SQL in managing relational databases.

Answer: SQL provides the tools to define, manipulate, and query relational databases. It allows users to create tables, insert and retrieve data, update records, enforce constraints, and control access and transactions.

4. What are the key features of SQL?

Answer: Key features include:

- Data Querying
- Data Manipulation
- Data Definition
- Data Control
- Transaction Control
- Use of constraints and joins
- Standardized language for relational databases

SQL Syntax

1. What are the basic components of SQL syntax?

Answer: Basic components include:

- **Keywords** (e.g., SELECT, INSERT)
- **Identifiers** (e.g., table/column names)
- **Expressions** (e.g., arithmetic or logical)
- **Clauses** (e.g., WHERE, ORDER BY)
- **Operators** (e.g., =, <, >)
- **Comments**

2. Write the general structure of an SQL SELECT statement.

Answer:

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

3. Explain the role of clauses in SQL statements.

Answer: Clauses define the conditions and scope of an SQL query, such as filtering rows (WHERE), sorting (ORDER BY), grouping (GROUP BY), and limiting results (LIMIT).

SQL Constraints

1. What are constraints in SQL? List and explain the different types of constraints.

Answer: Constraints enforce rules on table data. Types include:

- **NOT NULL:** Ensures a column cannot have NULL values.
- **UNIQUE:** Ensures all values in a column are unique.
- **PRIMARY KEY:** Uniquely identifies each row.
- **FOREIGN KEY:** Ensures referential integrity between tables.
- **CHECK:** Validates data based on a condition.
- **DEFAULT:** Sets a default value if none is provided.

2. How do PRIMARY KEY and FOREIGN KEY constraints differ?

Answer: A PRIMARY KEY uniquely identifies rows within its table. A FOREIGN KEY creates a relationship between two tables and ensures the value exists in the referenced table.

3. What is the role of NOT NULL and UNIQUE constraints?

Answer:

- **NOT NULL** prevents a column from accepting NULL values.
 - **UNIQUE** ensures all values in a column are distinct.
-

Main SQL Commands and Sub-commands (DDL)

1. Define the SQL Data Definition Language (DDL).

Answer: DDL is a subset of SQL used to define and modify database structures like tables, schemas, and indexes using commands such as CREATE, ALTER, and DROP.

2. Explain the CREATE command and its syntax.

Answer: CREATE is used to create a database or table.

Syntax:

```
sql
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint
);
```

3. What is the purpose of specifying data types and constraints during table creation?

Answer: It ensures data integrity, storage optimization, and enforces rules on what type of data can be stored in each column.

ALTER Command

1. What is the use of the ALTER command in SQL?

Answer: ALTER modifies the structure of an existing table, such as adding, modifying, or deleting columns or constraints.

2. How can you add, modify, and drop columns from a table using ALTER?

Answer:

- Add: ALTER TABLE table_name ADD column_name datatype;
 - Modify: ALTER TABLE table_name MODIFY column_name new_datatype;
 - Drop: ALTER TABLE table_name DROP COLUMN column_name;
-

DROP Command

1. What is the function of the DROP command in SQL?

Answer: DROP removes database objects like tables, views, or databases permanently.

2. What are the implications of dropping a table from a database?

Answer: All data and structure of the table are permanently deleted, and it cannot be recovered unless a backup exists.

Data Manipulation Language (DML)

1. Define the INSERT, UPDATE, and DELETE commands in SQL.

Answer:

- **INSERT** adds new rows to a table.
- **UPDATE** modifies existing data.
- **DELETE** removes rows from a table.

2. What is the importance of the WHERE clause in UPDATE and DELETE operations?

Answer: It specifies which rows should be updated or deleted. Without WHERE, all rows are affected.

Data Query Language (DQL)

1. What is the SELECT statement, and how is it used to query data?

Answer: SELECT retrieves data from tables. It allows filtering, sorting, and aggregation of results.

2. Explain the use of the ORDER BY and WHERE clauses in SQL queries.

Answer:

- **WHERE** filters rows based on conditions.
 - **ORDER BY** sorts the result set in ascending or descending order.
-

Data Control Language (DCL)

1. What is the purpose of GRANT and REVOKE in SQL?

Answer: They manage user permissions. GRANT gives access; REVOKE removes it.

2. How do you manage privileges using these commands?

Answer: Use GRANT to assign specific rights (e.g., SELECT, INSERT) to users and REVOKE to take them away.

Transaction Control Language (TCL)

1. What is the purpose of the COMMIT and ROLLBACK commands in SQL?

Answer:

- **COMMIT** saves all changes made in a transaction.
- **ROLLBACK** undoes changes since the last commit.

2. Explain how transactions are managed in SQL databases.

Answer: Transactions group one or more SQL statements into a single unit of work. They follow the ACID properties to ensure data integrity.

SQL Joins

1. Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

Answer: Joins combine rows from two or more tables based on related columns.

- **INNER JOIN:** Matches rows in both tables.
- **LEFT JOIN:** All rows from the left table and matched rows from the right.
- **RIGHT JOIN:** All rows from the right table and matched from the left.
- **FULL OUTER JOIN:** All rows when there is a match in either table.

2. How are joins used to combine data from multiple tables?

Answer: By using a common key, joins allow combining data from different tables into a single result set.

SQL GROUP BY

1. What is the GROUP BY clause in SQL? How is it used with aggregate functions?

Answer: GROUP BY groups rows with the same values and allows aggregate functions (like COUNT, SUM) to be applied to each group.

2. Explain the difference between GROUP BY and ORDER BY.

Answer:

- GROUP BY groups records based on column values.

- ORDER BY sorts the result set.
-

SQL Stored Procedure

1. What is a stored procedure in SQL, and how does it differ from a standard SQL query?

Answer: A stored procedure is a saved SQL code block that performs operations. Unlike a single query, it can include logic, loops, and multiple SQL statements.

2. Explain the advantages of using stored procedures.

Answer:

- Code reuse
 - Improved performance
 - Better security
 - Centralized logic
-

SQL View

1. What is a view in SQL, and how is it different from a table?

Answer: A view is a virtual table created from a SQL query. Unlike a table, it does not store data physically.

2. Explain the advantages of using views in SQL databases.

Answer:

- Simplify complex queries
 - Enhance security
 - Provide data abstraction
 - Present different data perspectives
-

SQL Triggers

1. What is a trigger in SQL? Describe its types and when they are used.

Answer: A trigger is a stored procedure that executes automatically in response to certain events (INSERT, UPDATE, DELETE). Types:

- **BEFORE Trigger**
- **AFTER Trigger**
- **INSTEAD OF Trigger**

2. Explain the difference between INSERT, UPDATE, and DELETE triggers.

Answer:

- **INSERT Trigger** activates on new row insertions.
 - **UPDATE Trigger** fires when data is modified.
 - **DELETE Trigger** responds to data deletion.
-

Introduction to PL/SQL

1. What is PL/SQL, and how does it extend SQL's capabilities?

Answer: PL/SQL (Procedural Language/SQL) extends SQL with procedural features like variables, loops, conditions, and functions, enabling more powerful programs.

2. List and explain the benefits of using PL/SQL.

Answer:

- Supports procedural logic
 - Code modularity and reuse
 - Enhances performance
 - Integrates seamlessly with SQL
-

PL/SQL Control Structures

1. What are control structures in PL/SQL? Explain the IF-THEN and LOOP control structures.

Answer: Control structures manage the flow of execution.

- **IF-THEN** executes code if a condition is true.
- **LOOP** repeats a set of statements until a condition is met.

2. How do control structures in PL/SQL help in writing complex queries?

Answer: They enable conditional logic and iteration, making it easier to handle complex business logic in a structured way.

SQL Cursors

1. What is a cursor in PL/SQL? Explain the difference between implicit and explicit cursors.

Answer: A cursor is a pointer to a result set.

- **Implicit:** Automatically created for SQL statements.
- **Explicit:** Declared by the user for custom control over result sets.

2. When would you use an explicit cursor over an implicit one?

Answer: When you need to process each row individually or perform complex row-by-row operations.

Rollback and Commit Savepoint

1. Explain the concept of SAVEPOINT in transaction management. How do ROLLBACK and COMMIT interact with savepoints?

Answer: A SAVEPOINT marks a point in a transaction. ROLLBACK can undo operations to a specific SAVEPOINT, and COMMIT makes all changes permanent.

2. When is it useful to use savepoints in a database transaction?

Answer: When you want to partially rollback a transaction without losing all changes—useful for error handling and complex operations.