

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

Ans.

SQL stands for Structured Query Language. It is a standard programming language used to interact with databases. SQL is used to create, read, update, and delete (CRUD) data in a database.

Store & fetch data (SELECT, INSERT)

Update & delete data (UPDATE, DELETE)

Create & manage tables (CREATE, ALTER, DROP)

Ensure data accuracy & security (constraints, permissions)

Works with almost all relational databases .

Que. Explain the difference between DBMS and RDBMS.

Ans.

DBMS (Database Management System)

A software system that manages databases.

Stores data as files or collections of records.

Does not enforce relationships between data.

Example: Microsoft Access, File System, dBase

RDBMS (Relational Database Management System)

A type of DBMS that stores data in tables (rows and columns).

Enforces relationships between tables using keys (Primary Key, Foreign Key).

Supports SQL for querying.

Example: MySQL, SQLite, PostgreSQL, Oracle, SQL Server

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

Ans.

SQL (Structured Query Language) is the standard language used to manage relational databases (RDBMS).

Create, modify, and delete tables.

Fetch specific information using conditions.

Use constraints primary key, foreign key, unique to maintain accuracy.

Manage permissions with SQL commands grant, revoke.

Que. What are the key features of SQL?

Ans.

Data definition.

Data manipulation.

Data retrieval.

Data integrity.

Data security.

Portability

Transaction control.

Que. What are the basic components of SQL syntax?

Ans.

Statements / Commands

SQL is made up of statements like SELECT, INSERT, UPDATE, DELETE, CREATE, etc.

Clauses

Add conditions or details to statements.

Common clauses: WHERE, ORDER BY, GROUP BY, HAVING.

Expressions

Produce values using operators, functions, or calculations.

Predicates

Conditions that evaluate to TRUE/FALSE/UNKNOWN.

Queries

Used to fetch data SELECT statement.

Semicolon

Ends an SQL statement.

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

Ans.

SELECT columns

FROM table_name

WHERE condition

ORDER BY column;

Que. Explain the role of clauses in SQL statements.

Ans.

clauses are used to add conditions or modify the behavior of a statement. They help in filtering, grouping, and sorting the data.

WHERE : Filters rows based on a condition.

GROUP BY : Groups rows that have the same values.

HAVING : Filters groups.

ORDER BY : Sorts the result set.

LIMIT / OFFSET : Restrict number of rows.

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

Ans.

Constraints are rules applied to table columns in SQL. They ensure the validity, accuracy, and reliability of the data in a database.

PRIMARY KEY

Uniquely identifies each record in a table.

Cannot be NULL.

FOREIGN KEY

Links one table to another.

Ensures values match a primary key in another table.

UNIQUE

Ensures all values in a column are different.

NOT NULL

Ensures a column cannot have NULL values.

CHECK

Ensures values meet a specific condition.

DEFAULT

Provides a default value if no value is given.

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

Ans.

PRIMARY KEY	FOREIGN KEY
Uniquely identifies each record in a table.	Creates a relationship between two tables.
Must be unique and not NULL.	Can have duplicate values.
Cannot contain NULL.	Can contain NULL.
Only one primary key per table	Can have multiple foreign keys in a table.

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

Ans.

NOT NULL Constraint

Ensures that a column cannot have empty values.

Every row must have a value in that column.

UNIQUE Constraint

Ensures that all values in a column are different.

Unlike PRIMARY KEY, a table can have multiple UNIQUE constraints.

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

Ans.

DDL is a part of SQL used to define and manage the structure of a database.

It deals with creating, altering, and deleting database objects like tables, schemas, indexes, etc.

CREATE → Creates a new database object.

ALTER → Modifies an existing object.

DROP → Deletes an object permanently.

TRUNCATE → Removes all rows from a table but keeps the structure.

Que. Explain the CREATE command and its syntax.

Ans.

The CREATE command is part of DDL (Data Definition Language).

It is used to create new database objects like databases, tables, views, indexes, etc.

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

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

Ans.

When you create a table, data types define what kind of values can be stored in each column.

Ensures data consistency

Helps the database use memory efficiently.

Prevents invalid data entry.

Constraints enforce *rules* to keep data valid and reliable.

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

Ans.

The ALTER command is part of DDL (Data Definition Language).

It is used to modify the structure of an existing table without deleting it.

Add a new column, Modify an existing column, Rename a column, Drop a column, Add or remove constraints

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

Ans.

ADD:

```
ALTER TABLE Students  
ADD phone_number VARCHAR(15);
```

Modify:

```
ALTER TABLE Students  
MODIFY name VARCHAR(100) NOT NULL;
```

Drop column:

ALTER TABLE Students

DROP COLUMN age;

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

Ans.

The DROP command is a DDL (Data Definition Language) statement.

It is used to permanently delete database objects such as:

Tables

Databases

Views

Indexes

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

Ans.

Permanent Data Loss

Loss of Table Structure

Permissions Are Removed

Cannot Be Rolled Back

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

Ans.

INSERT Command

Used to add new rows (records) into a table.

UPDATE Command

Used to modify existing records in a table.

Often used with WHERE to update specific rows.

DELETE Command

Used to remove records from a table.

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

Ans.

Without WHERE Clause

All rows in the table are affected.

Can lead to accidental data loss or unwanted updates.

With WHERE Clause

Only rows that meet the condition are affected.

Makes operations safe and precise.

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

Ans.

The SELECT statement is used to retrieve data from a database table. It allows you to choose which columns to display and filter or sort the results.

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

Ans.

WHERE Clause

Filters rows based on a condition.

Retrieve only the data you need.

ORDER BY Clause

Sorts the result set by one or more columns.

Display data in ascending (ASC) or descending (DESC) order.

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

Ans.

GRANT

Gives specific privileges to a user or role.

Allow users to perform actions like SELECT, INSERT, UPDATE, DELETE, etc.

REVOKE

Removes previously granted privileges from a user or role.

Prevent users from performing certain actions.

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

Ans.

COMMIT

Permanently saves all changes made in the current transaction.

Make sure your INSERT, UPDATE, DELETE operations are applied to the database.

ROLLBACK

Undo all changes made in the current transaction.

Cancel operations if something goes wrong, preventing errors from affecting the database.

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

Ans.

JOIN is used to combine rows from two or more tables based on a related column (usually a primary key and foreign key).

Helps in retrieving related data from multiple tables in a single query.

INNER JOIN

Returns only the rows that have matching values in both tables.

Rows without a match are not included.

LEFT JOIN

Returns all rows from the left table, and matching rows from the right table.

If no match, columns from the right table are NULL.

RIGHT JOIN

Returns all rows from the right table, and matching rows from the left table.

If no match, columns from the left table are NULL.

FULL OUTER JOIN

Returns all rows from both tables.

If there's no match, missing columns are NULL.

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

Ans.

JOINS allow you to combine rows from two or more tables based on a related column.

This is useful because relational databases store related data in separate tables to avoid duplication. JOINS help retrieve combined meaningful data in a single query.

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

Ans.

GROUP BY	ORDER BY
Groups rows that have the same value in a column.	Sorts the result set in ascending or descending order.
Often used with aggregate functions like COUNT(), SUM(), AVG().	Used to display data in a specific order.
Combines multiple rows into summary rows.	Does not combine rows just reorders them.
SELECT product, SUM(quantity) FROM Orders GROUP BY product;	SELECT name, age FROM Students ORDER BY age DESC;

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

Ans.

A stored procedure is a precompiled set of SQL statements stored in the database.

It can accept parameters, perform operations, and return results.

Helps in reusing code, improving performance, and enhancing security.

Stored in the database for reuse.

Can contain multiple SQL statements.

Can include control-of-flow statements like IF, WHILE.

Can accept input/output parameters.

Que. Explain the advantages of using stored procedures.

Ans.

Reusability, performance, reduce network traffic, security, maintainability, consistency.

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

Ans.

Table	View
A physical database object that stores data.	A virtual table created from a SQL query.
Stores data physically on disk.	Does not store data it only stores query definition.
Can be updated directly.	not be updatable.
Used for storing and managing data.	Used for simplifying queries, providing security and abstraction.

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

Ans.

Data Abstraction & Simplification

Security & Access Control

Consistency

Reusability

Logical Data Independence

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

Ans.

A trigger is like an automatic action in SQL.

It runs by itself when something happens in a table like INSERT, UPDATE, or DELETE.

Types of Triggers

1. BEFORE Trigger

Runs before data is inserted, updated, deleted.

Used for checking or changing data before saving.

2. AFTER Trigger

Runs after data is inserted, updated, deleted.

Used for logging, updating other tables, or audit purposes.

3. INSTEAD OF Trigger

Runs instead of the action.

Used when you want to control how data is updated in a view.

When to Use Triggers?

- To check data before saving.
- To keep logs of changes.
- To update related tables automatically.

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

Ans.

INSERT Trigger

- Runs when new data is added into a table.
- You can check, modify, or log inserted data.

UPDATE Trigger

- Runs when existing data is changed in a table.
- You can track old vs. new values.

DELETE Trigger

- Runs when data is removed from a table.
- You can stop deletion or keep a backup of deleted rows.

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

Ans.

PL/SQL is Oracle's extension of SQL.

SQL is good for working with data (like SELECT, INSERT, UPDATE, DELETE).

But SQL is not a full programming language → it can't do loops, conditions, or error handling.

PL/SQL adds programming features to SQL, so you can write procedures, functions, loops, conditions, and triggers inside the database.

Logic Control, Variables and Constants, Error Handling, Stored Procedures & Functions, Cursors, Triggers

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

Ans.

Combines SQL with Programming Logic

Improved Performance

Reusability with Procedures & Functions

Better Security

Error Handling

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

Ans.

Control structures are programming instructions that control the flow of execution in a PL/SQL block.

Conditional: IF-THEN, IF-THEN-ELSE, CASE

Iterative: LOOP, WHILE LOOP, FOR LOOP

Sequential: Statements run one after another in order

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

Ans.

A cursor in PL/SQL is like a pointer to the result set of a SQL query.

When you run a query like SELECT, Oracle creates a cursor to store the rows returned.

With cursors, you can fetch rows one by one and process them in PL/SQL.

Implicit Cursor	Explicit Cursor
Oracle automatically	Programmer explicitly declares it
Single-row queries (SELECT INTO)	Multi-row queries (loop row by row)
No control over fetch/open/close	Full control (OPEN, FETCH, CLOSE)
Faster for simple queries	Needed for complex/multi-row queries

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

Ans.

When Query Returns Multiple Rows.

You Need Row-by-Row Processing

You Want More Control

Complex Business Logic

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

Ans.

A SAVEPOINT is like a checkpoint inside a transaction. if something goes wrong, you can rollback only to that savepoint instead of undoing the whole transaction.

1. COMMIT

- Makes all changes permanent in the transaction (from the beginning or from the last commit).
- Once committed, you cannot rollback to a savepoint anymore.

2. ROLLBACK

- Undoes changes.
- If used without savepoint → undoes the entire transaction.
- If used with savepoint → undoes only changes made after that savepoint.

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

Ans.

You use savepoints when you want more control inside a transaction so you can undo only part of the work instead of the whole thing.

Complex Transactions with Multiple Steps

Error Handling in Programs

Partial Rollbacks

Want error recovery checkpoints