Module 4 – Introduction to DBMS and SQL Theory Questions

Q1. What is SQL, and why is it essential in database management?

Ans:- SQL (Structured Query Language) is used to create, manage, and query databases. It is essential for working with relational databases.

Q2. Explain the difference between DBMS and RDBMS.

Ans:- DBM**S** stores data in files; RDBMS stores data in tables with relationships between them.

Q3. Describe the role of SQL in managing relational databases.

Ans:- SQL helps in inserting, updating, deleting, and retrieving data in relational databases**.**

Q4. What are the key features of SQL?

Ans:- Key features of SQL:

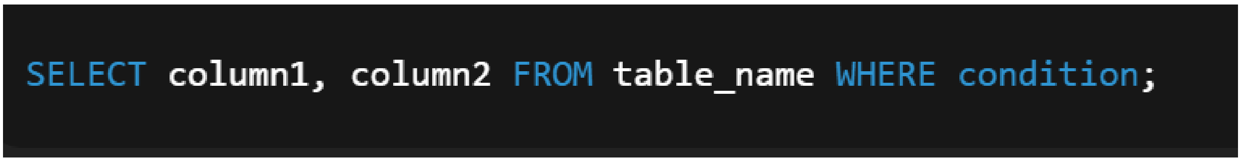
- Easy syntax
- Data control
- Data manipulation
- Standard language
- Supports transactions

Q5. What are the basic components of SQL syntax?

- **Keywords** (SELECT, FROM, WHERE)
- **Clauses**
- **Expressions**
- **Statements**

Q6. Write the general structure of an SQL SELECT statement.

Ans:-



```sql
SELECT column1, column2 FROM table_name WHERE condition;
```

Q7. Explain the role of clauses in SQL statements.

Ans:- **Clauses** add conditions to SQL queries, like filtering with **WHERE**, sorting with **ORDER BY**.

Q8. What are constraints in SQL? List and explain the different types of constraints.

Ans:- **Constraints** are rules on data:

- **NOT NULL** – no empty values
- **UNIQUE** – no duplicate values
- **PRIMARY KEY** – unique + not null
- **FOREIGN KEY** – links two tables
- **CHECK** – limits value range
- **DEFAULT** – sets a default value

Q9. How do PRIMARY KEY and FOREIGN KEY constraints differ?

Ans:- **PRIMARY KEY** uniquely identifies rows

FOREIGN KEY links to another table's primary key

Q10. What is the role of NOT NULL and UNIQUE constraints?

Ans:- **NOT NULL**: prevents empty fields

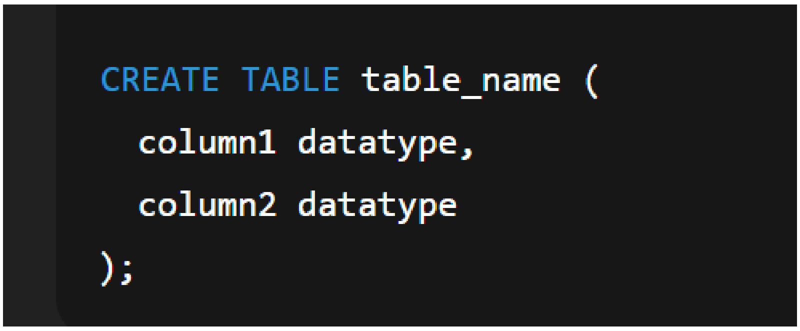
UNIQUE: ensures all values are different

O11. Define the SQL Data Definition Language (DDL).

Ans:- DDL (Data Definition Language) is used to define tables and structures (CREATE, ALTER, DROP).

Q12. Explain the CREATE command and its syntax.

Ans:-



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

Q13. What is the purpose of specifying data types and constraints during table creation?

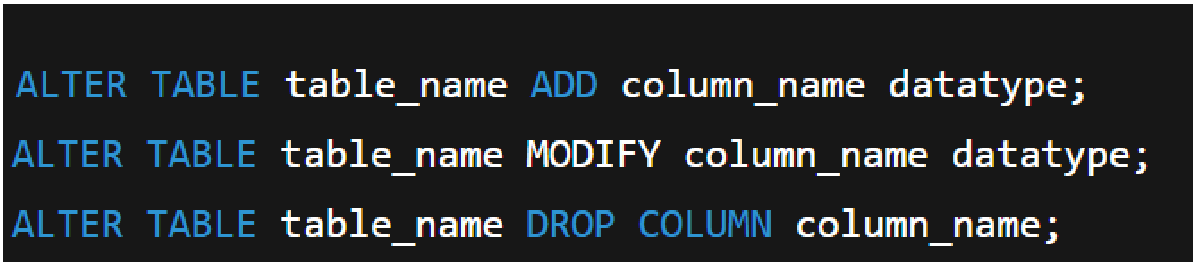
Ans:- To ensure correct data format and apply rules (e.g., age must be a number, ID must be unique).

Q14. What is the use of the ALTER command in SQL?

Ans:- ALTER is used to change the structure of a table (add, modify, or drop columns).

Q15. How can you add, modify, and drop columns from a table using ALTER?

Ans:-



```
ALTER TABLE table_name ADD column_name datatype;
ALTER TABLE table_name MODIFY column_name datatype;
ALTER TABLE table_name DROP COLUMN column_name;
```

Q16. What is the function of the DROP command in SQL?

Ans:- **DROP** deletes a table and all its data permanently.

Q17. What are the implications of dropping a table from a database?

Ans:- Once a table is dropped, all data is **lost** and cannot be recovered unless backed up.

Q18. Define the INSERT, UPDATE, and DELETE commands in SQL.

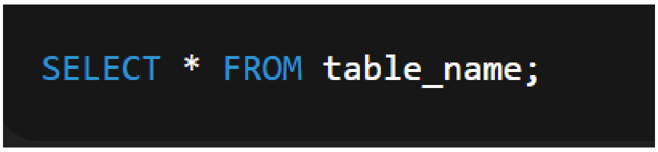
Ans:-
- **INSERT** adds new rows
- **UPDATE** modifies rows
- **DELETE** removes rows

Q19. What is the importance of the WHERE clause in UPDATE and DELETE operations?

Ans:- **WHERE** ensures only specific rows are updated or deleted, avoiding unwanted changes.

Q20. What is the SELECT statement, and how is it used to query data?

Ans:- **SELECT** retrieves data from tables.



```sql
SELECT * FROM table_name;
```

Q22. Explain the use of the ORDER BY and WHERE clauses in SQL queries.

Ans:-
- **ORDER BY** sorts results
- **WHERE** filters rows

Q23. What is the purpose of GRANT and REVOKE in SQL?

Ans:- **GRANT** gives user rights; **REVOKE** removes them.

Q24. How do you manage privileges using these commands?

Ans:- Use these to manage who can SELECT, INSERT, UPDATE, DELETE, etc.

Q25. What is the purpose of the COMMIT and ROLLBACK commands in SQL?

Ans:- • **COMMIT** saves changes

- **ROLLBACK** cancels uncommitted changes

Q26. Explain how transactions are managed in SQL databases.

Ans:- Transactions are handled using COMMIT, ROLLBACK, and SAVEPOINT to ensure data consistency.

Q27. Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

Ans:- • **JOIN** combines rows from tables

- **INNER JOIN**: common data only
- **LEFT JOIN**: all from left + match
- **RIGHT JOIN**: all from right + match
- **FULL JOIN**: all data, matched or no

Q28. How are joins used to combine data from multiple tables?

Ans:- Joins help in fetching related data from multiple tables using common fields.

Q29. What is the GROUP BY clause in SQL? How is it used with aggregate functions?

Ans:- **GROUP BY** groups rows with same values to use with aggregate functions (SUM, AVG, etc.).

Q30. Explain the difference between GROUP BY and ORDER BY.

Ans:- • **GROUP BY**: for aggregation

- **ORDER BY**: for sorting results

Q31. What is a view in SQL, and how is it different from a table?

Ans:- A **view** is a virtual table based on a query. It does not store data like a table.

Q32. Explain the advantages of using views in SQL databases.

Ans:- Advantages:

- Simplifies queries
- Hides complexity
- Adds security

Q33. What is a trigger in SQL? Describe its types and when they are used.

Ans:- A **trigger** is an automatic action on INSERT, UPDATE, DELETE.
Types:

- BEFORE
- AFTER
- INSTEAD OF

Q34. Explain the difference between INSERT, UPDATE, and DELETE triggers.

Ans:- • **INSERT Trigger** – fires on new data

- **UPDATE Trigger** – fires on data change

- **DELETE Trigger** – fires on row delete

Q35. What is PL/SQL, and how does it extend SQL's capabilities?

Ans:- **PL/SQL** is Oracle's extension of SQL. It adds programming features like loops, conditions.

Q36. List and explain the benefits of using PL/SQL.

Ans:- Benefits:

- Better performance
- Easy error handling
- Supports logic and loops

Q37. What are control structures in PL/SQL? Explain the IF-THEN and LOOP control structures.

Ans:- • **F-THEN**: decision-making

- **LOOP**: repeats a block of code

Q38. How do control structures in PL/SQL help in writing complex queries?

Ans:- They allow complex logic inside SQL, like checking conditions or repeating steps.

Q39. What is a cursor in PL/SQL? Explain the difference between implicit and explicit cursors.

Ans:- • **Cursor** lets you process rows one-by-one.

- **Implicit**: auto-handled
- **Explicit**: defined by the user

Q40. When would you use an explicit cursor over an implicit one?

Ans:- Use **explicit cursors** when you need control over row-by-row processing.

Q41. Explain the concept of SAVEPOINT in transaction management. How do ROLLBACK and COMMIT interact with savepoints?

Ans:- **SAVEPOINT** lets you mark a point to **ROLLBACK** to without undoing all changes.

Q42. When is it useful to use savepoints in a database transaction?

Ans:- Use savepoints to undo parts of a transaction while keeping earlier actions.