DBMS Assignment - Complete Theory Answers

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

SQL (Structured Query Language) is a standard language used to communicate with relational databases. It is essential because it allows users to store, retrieve, manipulate, and manage data efficiently. SQL ensures data integrity, security, and consistency, making it a key tool in database management.

## 2. Explain the difference between DBMS and RDBMS.

DBMS (Database Management System) is software that manages databases and provides operations such as storage, retrieval, and updates. RDBMS (Relational Database Management System) is a type of DBMS that stores data in tabular form with relationships between them. Examples: DBMS – MS Access, RDBMS – MySQL, Oracle, SQL Server.

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

SQL plays a vital role in relational databases by providing commands to create tables, insert data, retrieve data, update records, and enforce constraints. It acts as the medium between users and the database system.

## 4. What are the key features of SQL?

Key features of SQL include:   
- Simple and easy-to-learn syntax  
- Ability to manage large datasets  
- Support for data definition, manipulation, and control  
- High security and integrity through constraints  
- Portability across different RDBMS platforms

## 5. What are the basic components of SQL syntax?

The basic components of SQL syntax include:   
- Keywords (SELECT, INSERT, UPDATE, DELETE)  
- Identifiers (table names, column names)  
- Clauses (WHERE, ORDER BY, GROUP BY)  
- Expressions (mathematical or logical)  
- Statements (complete SQL commands).

## 6. Write the general structure of an SQL SELECT statement.

The general structure is:  
SELECT column1, column2, ...   
FROM table\_name   
[WHERE condition]   
[GROUP BY column]   
[HAVING condition]   
[ORDER BY column];

## 7. Explain the role of clauses in SQL statements.

Clauses are used to provide conditions or additional instructions to SQL statements. For example: WHERE filters data, GROUP BY groups rows, HAVING applies conditions on groups, and ORDER BY sorts the result set.

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

Constraints are rules applied to table columns to ensure data integrity and accuracy. Types of constraints include:  
- NOT NULL: Ensures a column cannot have NULL values  
- UNIQUE: Ensures all values in a column are unique  
- PRIMARY KEY: Uniquely identifies each record  
- FOREIGN KEY: Establishes a relationship between tables  
- CHECK: Ensures values meet a specific condition  
- DEFAULT: Assigns a default value if none is provided.

## 9. How do PRIMARY KEY and FOREIGN KEY constraints differ?

A PRIMARY KEY uniquely identifies each record within a table, while a FOREIGN KEY links a column in one table to the PRIMARY KEY of another, establishing a relationship.

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

NOT NULL ensures that a column cannot have missing (NULL) values, while UNIQUE ensures that all values in a column are different, avoiding duplicates.

## 11. Define the SQL Data Definition Language (DDL).

DDL (Data Definition Language) is a subset of SQL used to define and manage database structures. It includes commands like CREATE, ALTER, DROP, and TRUNCATE.

## 12. Explain the CREATE command and its syntax.

The CREATE command is used to create databases and tables. Syntax:  
CREATE TABLE table\_name (  
 column1 datatype constraint,  
 column2 datatype constraint,  
 ...  
);

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

Data types ensure that only valid types of data are stored in a column (e.g., INT for numbers, VARCHAR for text). Constraints enforce rules such as uniqueness, relationships, and required fields to maintain data integrity.

## 14. What is the use of the ALTER command in SQL?

The ALTER command is used to modify an existing database object such as a table. It can add, modify, or delete columns and constraints.

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

- Add: ALTER TABLE table\_name ADD column\_name datatype;  
- Modify: ALTER TABLE table\_name MODIFY column\_name datatype;  
- Drop: ALTER TABLE table\_name DROP COLUMN column\_name;

## 16. What is the function of the DROP command in SQL?

The DROP command is used to permanently delete database objects such as tables, databases, or views.

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

Dropping a table permanently removes its structure and all stored data. This operation cannot be undone unless a backup exists.

## 18. Define the INSERT, UPDATE, and DELETE commands in SQL.

- INSERT: Adds new records to a table.  
- UPDATE: Modifies existing records in a table.  
- DELETE: Removes records from a table.

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

The WHERE clause ensures that only specific rows are updated or deleted. Without it, all rows in the table would be affected, which may cause data loss.

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

The SELECT statement retrieves data from one or more tables. It can select specific columns or all columns and apply filters, sorting, and grouping.

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

- ORDER BY: Sorts the result set in ascending or descending order.  
- WHERE: Filters records based on a condition.

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

GRANT provides specific privileges to users (like SELECT, INSERT), while REVOKE removes previously granted privileges.

## 23. How do you manage privileges using these commands?

By using GRANT to assign permissions and REVOKE to withdraw them, administrators manage access control in databases.

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

COMMIT saves all changes made during a transaction permanently, while ROLLBACK undoes changes made since the last COMMIT or SAVEPOINT.

## 25. Explain how transactions are managed in SQL databases.

Transactions are managed using ACID properties (Atomicity, Consistency, Isolation, Durability). They group multiple operations into a single unit of work to ensure data integrity.

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

A JOIN combines rows from multiple tables based on related columns.  
- INNER JOIN: Returns matching rows from both tables.  
- LEFT JOIN: Returns all rows from the left table and matching rows from the right.  
- RIGHT JOIN: Returns all rows from the right table and matching rows from the left.  
- FULL OUTER JOIN: Returns all rows when there is a match in one of the tables.

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

Joins are used to retrieve related data stored in different tables by linking them through primary and foreign keys.

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

GROUP BY groups rows that have the same values into summary rows. It is often used with aggregate functions like COUNT, SUM, AVG, MIN, and MAX.

## 29. Explain the difference between GROUP BY and ORDER BY.

GROUP BY groups rows based on column values for aggregation, while ORDER BY sorts the results in ascending or descending order.

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

A stored procedure is a precompiled collection of SQL statements stored in the database. Unlike a standard query, it can accept parameters and be reused multiple times.

## 31. Explain the advantages of using stored procedures.

Advantages include:   
- Better performance (precompiled)  
- Reusability and modular programming  
- Improved security with restricted access  
- Reduced network traffic

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

A view is a virtual table created from a query result. Unlike tables, views do not store data physically but provide a way to simplify complex queries.

## 33. Explain the advantages of using views in SQL databases.

Advantages include:  
- Simplifies complex queries  
- Provides data security by restricting access to specific columns  
- Enhances data independence  
- Can be used for reusability and abstraction

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

A trigger is a set of SQL instructions that automatically executes when a specific event occurs on a table. Types include BEFORE, AFTER, and INSTEAD OF triggers. They are used for enforcing business rules and logging changes.

## 35. Explain the difference between INSERT, UPDATE, and DELETE triggers.

- INSERT Trigger: Executes when a new row is added.  
- UPDATE Trigger: Executes when a row is modified.  
- DELETE Trigger: Executes when a row is removed.

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

PL/SQL (Procedural Language for SQL) extends SQL by adding procedural programming features such as loops, conditions, and exception handling.

## 37. List and explain the benefits of using PL/SQL.

Benefits include:  
- Combines SQL with procedural features  
- Enhances performance with block structures  
- Improves security with stored code  
- Provides modularity and reusability

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

Control structures are constructs that control the flow of execution. - IF-THEN: Executes statements based on a condition.  
- LOOP: Repeats a block of code until an exit condition is met.

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

They allow conditional execution and repetition, enabling developers to implement business logic directly inside the database.

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

A cursor is a pointer to the result set of a query. - Implicit cursors are automatically created for single SQL statements.  
- Explicit cursors are declared by the user to handle complex queries with multiple rows.

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

Explicit cursors are used when you need to fetch multiple rows one by one, especially in loops.

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

A SAVEPOINT is a marker within a transaction that allows partial rollback. ROLLBACK can undo changes up to a SAVEPOINT, while COMMIT saves all changes permanently.

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

Savepoints are useful in long transactions where only part of the changes may need to be undone without affecting the entire transaction.