

# Assignment

## Theory Questions

**Question 1 : Explain the fundamental differences between DDL, DML, and DQL commands in SQL. Provide one example for each type of command.**

### 1.DDL (Data Definition Language)

Definition:

DDL (Data Definition Language) consists of commands used to define, create, modify, and delete the structure of database objects such as tables, schemas, and databases. These commands deal with the overall structure of the database rather than the actual data stored inside the tables. When a table is created or its structure is changed, DDL commands are used.

Common DDL Commands:

CREATE – Creates a new table or database.

DROP – Deletes an existing table or database completely.

ALTER – Modifies the structure of an existing table.

TRUNCATE – Removes all records from a table but keeps its structure.

RENAME – Changes the name of a table.

Example:

```
CREATE TABLE Employees (  
    emp_id INT,  
    name VARCHAR(50),  
    salary INT  
);
```

This command creates a new table named Employees.

### 2.DML (Data Manipulation Language)

Definition:

DML (Data Manipulation Language) consists of commands used to insert, update, and delete data within database tables. These commands operate on the records (rows) stored in the table. After a table is created using DDL, DML commands are used to manage the data inside it.

Common DML Commands:

INSERT – Adds new records into a table.

UPDATE – Modifies existing records in a table.

DELETE – Removes records from a table.

Example:

```
INSERT INTO Employees VALUES (101, 'Rahul', 50000);
```

This command inserts a new record into the Employees table.

### 3.DQL (Data Query Language)

Definition:

DQL (Data Query Language) consists of commands used to retrieve data from the database. These commands allow users to fetch specific records from one or more tables based on given conditions. DQL does not modify the database structure or data; it only retrieves information.

Main DQL Command:

SELECT – Retrieves data from one or more tables.

Example:

```
SELECT * FROM Employees;
```

This command retrieves all records from the Employees table.

**Question 2 : What is the purpose of SQL constraints? Name and describe three common types of constraints, providing a simple scenario where each would be useful.**

SQL constraints are rules applied to table columns to enforce data integrity, accuracy, and reliability in a database. They restrict the type of data that can be inserted into a table and ensure that invalid or inconsistent data is not stored. Constraints are defined at the time of table creation or can be added later using the ALTER command.

Three common types of SQL constraints are:

#### 1. PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table. A primary key column cannot contain NULL values and must contain unique values. Each table can have only one primary key.

Scenario:

In an Orders table, the Order\_ID column can be defined as a primary key. This ensures that every order has a unique identifier and no two orders share the same ID.

Example:

```
Order_ID INT PRIMARY KEY
```

## 2. UNIQUE Constraint

The UNIQUE constraint ensures that all values in a column are different from each other. Unlike the primary key, a UNIQUE column can contain one NULL value (depending on the database system), but duplicate values are not allowed.

Scenario:

In an Employees table, the Email column can be defined as UNIQUE to ensure that no two employees register with the same email address.

Example:

Email VARCHAR(100) UNIQUE

## 3. CHECK Constraint

The CHECK constraint is used to enforce a specific condition on a column. It ensures that all values inserted into the column satisfy the defined condition.

Scenario:

In a Students table, the Marks column can have a CHECK constraint to ensure that marks are between 0 and 100.

Example:

Marks INT CHECK (Marks >= 0 AND Marks <= 100)

**Question 3 : Explain the difference between LIMIT and OFFSET clauses in SQL. How would you use them together to retrieve the third page of results, assuming each page has 10 records?**

The LIMIT and OFFSET clauses are used in SQL to control the number of records returned in a query result. They are commonly used for pagination, where data is divided into multiple pages.

LIMIT is used to specify the maximum number of rows to return in the result set. It restricts the total number of records retrieved from the query.

OFFSET is used to skip a specified number of rows before starting to return records. It determines where the result set should begin.

Difference:

- LIMIT controls how many rows are returned.

- OFFSET controls how many rows are skipped before returning results.

Using LIMIT and OFFSET Together:

If each page contains 10 records and we want to retrieve the third page:

- Page 1 → records 1–10
- Page 2 → records 11–20
- Page 3 → records 21–30

To retrieve the third page, we must:

- Skip the first 20 records (OFFSET 20)
- Return the next 10 records (LIMIT 10)

SQL Query:

```
SELECT * FROM table_name  
LIMIT 10 OFFSET 20;
```

This query skips the first 20 rows and retrieves the next 10 rows, which represent the third page of results.

**Question 4 : What is a Common Table Expression (CTE) in SQL, and what are its main benefits? Provide a simple SQL example demonstrating its usage.**

A Common Table Expression (CTE) is a temporary result set defined using the WITH keyword before the main query. It allows a query to be written in a structured and readable way by creating a temporary named result that can be referenced within the same query. A CTE exists only during the execution of that query and is not stored permanently in the database.

### **Simple Example Demonstrating Its Usage**

Suppose we want to find employees whose salary is greater than 50000.

WITH HighSalaryEmployees AS (

SELECT name, salary

FROM Employees

WHERE salary > 50000

)

SELECT \*

FROM HighSalaryEmployees;

Explanation:

The WITH keyword creates a CTE named HighSalaryEmployees.

The query inside the parentheses selects employees with salary greater than 50000.

The main query then selects all records from the CTE.

The CTE exists only for this query execution.

**Question 5 : Describe the concept of SQL Normalization and its primary goals. Briefly explain the first three normal forms (1NF, 2NF, 3NF).**

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It divides large tables into smaller related tables and removes unnecessary duplication of data.

Primary Goals of Normalization:

To reduce duplicate data

To avoid insertion, update, and deletion anomalies

To maintain data consistency

To organize data efficiently

-First Normal Form (1NF)

A table is in 1NF if:

Each column contains only atomic (single) values

There are no repeating groups or multiple values in one column

Example: A student table should not store multiple phone numbers in one column.

## -Second Normal Form (2NF)

A table is in 2NF if:

- It is already in 1NF

- All non-key attributes are fully dependent on the entire primary key

This removes partial dependency, especially when a composite key is used.

## -Third Normal Form (3NF)

A table is in 3NF if:

- It is already in 2NF

- There is no transitive dependency

This means non-key attributes should depend only on the primary key, not on other non-key attributes.