Introduction to SQL and Advanced Functions Assignment

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

Answer: DDL (Data Definition Language) commands are used to define or modify the structure of database objects like tables and schemas. These commands affect the database structure permanently.
DML (Data Manipulation Language) commands are used to insert, update, or delete data stored inside tables. They change the actual records but not the structure.
DQL (Data Query Language) commands are mainly used to retrieve data from the database without modifying it.
DDL example: CREATE TABLE Students (...)
DML example: INSERT INTO Students VALUES (...)
DQL example SELECT * FROM Students ;

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.

Answer : SQL constraints are rules applied to table columns to ensure the accuracy, consistency, and reliability of data.
They prevent invalid, duplicate, or inconsistent data from being entered into the database.
Constraints help maintain data integrity and enforce business rules at the database level.
A **PRIMARY KEY** constraint uniquely identifies each record in a table and does not allow NULL values.
It is useful for uniquely identifying records such as CustomerID or ProductID.
A **NOT NULL** constraint ensures that a column always contains a value.
For example, ProductName should not be left empty.
A **FOREIGN KEY** constraint establishes a relationship between two tables.
It ensures that values in one table match values in another table.
For example, CategoryID in the Products table must exist in the Categories table.

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?

Answer : The LIMIT clause in SQL is used to control the number of rows returned by a query.
It is commonly used when only a specific number of records are required.
The OFFSET clause specifies how many rows should be skipped before starting to return records.
OFFSET is mainly used together with LIMIT for pagination purposes.
LIMIT defines the page size, while OFFSET defines the starting position.
For example, displaying results page by page on a website uses these clauses.
If each page contains 10 records, the first page has OFFSET 0.
The second page has OFFSET 10, and the third page has OFFSET 20.
Using both clauses helps improve performance by fetching only required data.
This approach avoids loading large result sets unnecessarily.
Example :

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

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.

Answer : A Common Table Expression (CTE) is a temporary result set that is defined within the execution scope of a single SQL statement . It is created using a WITH keyword and can be referenced like a table in the main query. A CTE exists only for the duration of query and does not get stored permanently in the database.

Benefits of CTE :

- It exists only for the duration of a single query execution.
- CTEs make complex queries easier to read and understand.
- They allow developers to break large queries into smaller, logical sections.
- CTEs improve code readability compared to nested subqueries.
- They can be referenced multiple times within the same query.
- CTEs are especially useful for hierarchical and recursive queries.
- They also help in writing cleaner analytical queries.

- CTEs do not store data permanently in the database.
- They act like a temporary view for query execution.

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

Answer : SQL Normalization is a systematic process of organizing data in a relational database to reduce redundancy and dependency by dividing large tables into smaller, well-structured tables and defining relationships between them. The primary goal of normalization is to ensure data consistency, integrity, and efficient storage, while avoiding anomalies during insert, update, and delete operations.
Its primary goal is to eliminate duplicate data and improve data consistency.
The First Normal Form (1NF) ensures that a table contains only atomic (indivisible) values and that each field holds a single value. It also requires that each record can be uniquely identified using a primary key. For example, a table should not store multiple phone numbers in a single column.
The Second Normal Form (2NF) builds on 1NF by eliminating partial dependency. This means that all non-key attributes must be fully dependent on the entire primary key, not just a part of it. This rule is especially important in tables with composite primary keys, such as an Order table where product details depend only on ProductID and not the entire key.
The Third Normal Form (3NF) ensures that there is no transitive dependency between non-key attributes. In other words, non-key attributes should depend only on the primary key and not on other non-key attributes. For example, storing department name separately instead of repeating it with employee details helps achieve 3NF.