**SQL Assignment 5**

1. Explain how SQL Query keyword statements are executed in order.
2. Explain the advantages of stored procedures and their syntax in relation to recompiling stored procedures.
3. Give an example of the derived table.
4. What is the database's trigger? Explain the different forms of triggers that can be found in the database.
5. What are the benefits and drawbacks of triggers?
6. Create a stored procedure to call other stored procedures.

**Here’s a breakdown of your SQL assignment questions:**

**### 1. SQL Query Execution Order**

**SQL queries are executed in a specific order, which generally follows this sequence:**

**1. \*\*FROM\*\*: Specifies the tables from which to retrieve data. This step is executed first to gather the data.**

**2. \*\*JOIN\*\*: Combines rows from two or more tables based on related columns.**

**3. \*\*WHERE\*\*: Filters the results based on specified conditions.**

**4. \*\*GROUP BY\*\*: Groups the filtered results based on one or more columns.**

**5. \*\*HAVING\*\*: Filters groups based on specified conditions (used with aggregate functions).**

**6. \*\*SELECT\*\*: Specifies which columns to return in the result set.**

**7. \*\*DISTINCT\*\*: Removes duplicate rows from the result set.**

**8. \*\*ORDER BY\*\*: Sorts the final result set based on one or more columns.**

**9. \*\*LIMIT / OFFSET\*\*: Restricts the number of rows returned by the query.**

**### 2. Advantages of Stored Procedures and Syntax**

**\*\*Advantages:\*\***

**- \*\*Performance\*\*: Stored procedures are precompiled and can reduce execution time.**

**- \*\*Reusability\*\*: They can be reused across multiple applications.**

**- \*\*Maintainability\*\*: Changes can be made in one place, improving manageability.**

**- \*\*Security\*\*: Stored procedures can restrict direct access to tables.**

**- \*\*Reduced Network Traffic\*\*: Multiple SQL statements can be executed in a single call.**

**\*\*Syntax for Creating a Stored Procedure:\*\***

**```sql**

**CREATE PROCEDURE procedure\_name**

**AS**

**BEGIN**

**-- SQL statements**

**END;**

**```**

**To recompile a stored procedure, you can use the following command:**

**```sql**

**EXEC sp\_recompile 'procedure\_name';**

**```**

**### 3. Example of a Derived Table**

**A derived table is a subquery that is used in the `FROM` clause. Here’s an example:**

**```sql**

**SELECT dt.DepartmentID, dt.TotalEmployees**

**FROM (SELECT DepartmentID, COUNT(\*) AS TotalEmployees**

**FROM Employees**

**GROUP BY DepartmentID) AS dt;**

**```**

**### 4. Database Triggers**

**A trigger is a special type of stored procedure that automatically executes in response to certain events on a particular table or view.**

**\*\*Different Forms of Triggers:\*\***

**- \*\*DML Triggers\*\*: Execute in response to Data Manipulation Language events (INSERT, UPDATE, DELETE).**

**- \*\*BEFORE Triggers\*\*: Execute before the DML event.**

**- \*\*AFTER Triggers\*\*: Execute after the DML event.**

**- \*\*INSTEAD OF Triggers\*\*: Execute in place of the DML event (commonly used for views).**

**- \*\*DDL Triggers\*\*: Execute in response to Data Definition Language events (CREATE, ALTER, DROP).**

**- \*\*LOGON/LOGOFF Triggers\*\*: Execute in response to user logon or logoff events.**

**### 5. Benefits and Drawbacks of Triggers**

**\*\*Benefits:\*\***

**- \*\*Automated Actions\*\*: Triggers can enforce business rules automatically.**

**- \*\*Data Integrity\*\*: They help maintain data integrity by performing checks or constraints.**

**- \*\*Auditing\*\*: Triggers can log changes or actions for auditing purposes.**

**\*\*Drawbacks:\*\***

**- \*\*Performance Overhead\*\*: They can slow down DML operations due to additional processing.**

**- \*\*Complexity\*\*: Overuse can lead to complex interdependencies that are hard to manage.**

**- \*\*Hidden Logic\*\*: Logic within triggers can be less visible, making it harder to debug or understand the flow.**

**### 6. Creating a Stored Procedure to Call Other Stored Procedures**

**Here’s an example of a stored procedure that calls other stored procedures:**

**```sql**

**CREATE PROCEDURE MainProcedure**

**AS**

**BEGIN**

**-- Call first stored procedure**

**EXEC FirstStoredProcedure;**

**-- Call second stored procedure**

**EXEC SecondStoredProcedure;**

**END;**

**```**

**Replace `FirstStoredProcedure` and `SecondStoredProcedure` with the actual names of your stored procedures.**

**Feel free to ask if you need more details on any of these topics!**