SQL Assignment 3

# 1. Create a function and then call another function from within it. What is this process called?

The process of calling one function from within another is known as Function Nesting or Function Invocation.

# 2. How to inspect the query's execution plan?

To inspect a query's execution plan in SQL, you can use the 'EXPLAIN' statement before the SQL query. For example: 'EXPLAIN SELECT \* FROM employees;' This will show the execution steps SQL Server would take to execute the query.

# 3. What is the purpose of the MAXDOP and recompiling keywords in SQL queries?

MAXDOP (Maximum Degree of Parallelism) limits the number of processors used by SQL Server for executing a query. By default, SQL Server uses all available processors, but MAXDOP can be used to control it. The RECOMPILE keyword forces SQL Server to recompile the query plan every time it is executed. This can be useful when query performance varies significantly with different parameter values.

# 4. How to build DDL statements from an existing database table, write steps for it?

To build DDL (Data Definition Language) statements from an existing table, follow these steps:  
1. Use SQL Server Management Studio (SSMS) to script the table as a CREATE statement.  
2. Right-click the table in SSMS > Script Table as > CREATE To > New Query Editor Window.  
3. The generated script will be a CREATE TABLE statement containing all the table's columns, data types, constraints, etc.  
4. You can also use the 'Generate Scripts' option in SSMS to generate DDL for other objects like indexes, views, etc.

# 5. How to update data in a table using an inner join, write an example?

To update data in a table using an inner join, use the following SQL query:  
Example:  
UPDATE employees  
SET employees.salary = departments.salary  
FROM employees  
INNER JOIN departments ON employees.department\_id = departments.id  
WHERE employees.job\_title = 'Manager';  
This query updates the salary in the employees table based on a join with the departments table.

# 6. Differentiate between truncate, delete, and drop with a suitable example.

TRUNCATE: Removes all rows from a table, but does not log individual row deletions. It cannot be rolled back. It resets identity columns.  
Example: 'TRUNCATE TABLE employees;'  
  
DELETE: Removes rows from a table based on a condition. Each row deletion is logged, and DELETE can be rolled back.  
Example: 'DELETE FROM employees WHERE job\_title = 'Manager';'  
  
DROP: Removes a table or other database object entirely, including all data, structure, and constraints. It cannot be rolled back.  
Example: 'DROP TABLE employees;'