**SQL Assignment 3**

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

Ans. A procedure can be executed inside the function with the help of OPENROWSET() using OLE DB provider connection MSDASQL. Users can define the OPENROWSET() connection with the necessary details of SQL Server instance with declared Linked Server and credentials that has access to the database. Here, we will have an error handling and different scenario with INSERT and SELECTS operation in the procedure and calling it by SQL Server function. Let’s start by calling a procedure with the OPENROWSET T-SQL statement.

Syntax:-

SELECT \*

FROM OPENROWSET('MSDASQL','DRIVER={SQL Server}; SERVER=localhost; Trusted\_Connection=yes', 'EXEC AdventureWorks..[usp\_get]');

1. How to inspect the query's execution plan?
2. Ans. On the toolbar, click Database Engine Query. You can also open an existing query and display the estimated execution plan by clicking the Open File toolbar button and locating the existing query.
3. Enter the query for which you would like to display the estimated execution plan.
4. On the Query menu, click Display Estimated Execution Plan or click the Display Estimated Execution Plan toolbar button. The estimated execution plan is displayed on the Execution Plan tab in the results pane.
5. To alter the display of the execution plan, right-click the execution plan and select **Zoom In**, **Zoom Out**, **Custom Zoom**, or **Zoom to Fit**. **Zoom In** and **Zoom Out** allow you to magnify or reduce the execution plan by fixed amounts. **Custom Zoom** allows you to define your own display magnification, such as zooming at 80 percent. **Zoom to Fit** magnifies the execution plan to fit the result pane. Alternatively, use a combination of the CTRL key and your mouse wheel to activate **dynamic zoom**.
6. To navigate the display of the execution plan, use the vertical and horizontal scroll bars, or **click and hold on any blank area** of the execution plan, and **drag your mouse**. Alternatively, click and hold the plus (+) sign in the right lower corner of the execution plan window, to display a miniature map of the entire execution plan.
7. What is the purpose of the MAXDOP and recompiling keywords in SQL queries?

Ans.

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

Ans. The CREATE TABLE is a DDL statement which is used to create tables in the database.The table gets created as soon as the CREATE TABLE script is executed and is ready to hold the data onwards.The user must have the CREATE TABLE system privilege to create the table in its own schema.But to create a table in any user's schema, user must have CREATE ANY TABLE schema.

Here is the syntax of a basic CREATE TABLE statement.There may be many additional clauses to explicitly provide the storage specifications or segment values.

CREATE TABLE [schema.]table

( { column datatype [DEFAULT expr] [column\_constraint] ...

| table\_constraint}

[, { column datatype [DEFAULT expr] [column\_constraint] ...

| table\_constraint} ]...)

[AS subquery]

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

Ans. SQL UPDATE JOIN means we will update one table using another table and join condition.

UPDATE customer\_table

INNER JOIN

Customer\_table

ON customer\_table.rel\_cust\_name = customer\_table.cust\_id

SET customer\_table.rel\_cust\_name = customer\_table.cust\_name

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

**Ans. The DROP command** removes a table from the database.

1. All the tables' rows, indexes, and privileges will also be removed.
2. No DML triggers will be fired.
3. The operation cannot be rolled back.
4. DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command.
5. DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back

**TRUNCATE TABLE**

1. TRUNCATE is a DDL command
2. TRUNCATE is executed using a table lock and the whole table is locked to remove all records.
3. We cannot use the WHERE clause with TRUNCATE.
4. TRUNCATE removes all rows from a table.
5. Minimal logging in the transaction log, so it is faster performance-wise.
6. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.
7. Identify the column is reset to its seed value if the table contains an identity column.
8. To use Truncate on a table you need at least ALTER permission on the table.
9. Truncate uses less transaction space than the Delete statement.
10. Truncate cannot be used with indexed views.
11. TRUNCATE is faster than DELETE.

**DELETE**

1. DELETE is a DML command.
2. DELETE is executed using a row lock, each row in the table is locked for deletion.
3. We can use where clause with DELETE to filter & delete specific records.
4. The DELETE command is used to remove rows from a table based on WHERE condition.
5. It maintains the log, so it slower than TRUNCATE.
6. The DELETE statement removes rows one at a time and records an entry in the transaction log for each deleted row.
7. Identity of column keep DELETE retains the identity.
8. To use Delete you need DELETE permission on the table.
9. Delete uses more transaction space than the Truncate statement.
10. The delete can be used with indexed views.