**SQL Assignment 3**

1. Create a function and then call another function from within it. What is this process called?
2. The process of calling one function from within another function is known as "function invocation" or "calling a function." In programming, a function is a self-contained block of code that performs a specific task. When you define a function, you are creating a reusable piece of code that can be called and executed whenever needed.

Here's a simple example in Python:

def outer\_function():

print("This is the outer function.")

def inner\_function():

print("This is the inner function.")

# Call the inner function from within the outer function

inner\_function()

# Call the outer function

outer\_function()

2)How to inspect the query's execution plan?

Inspecting the execution plan of a query is crucial for database optimization and performance tuning. The method to inspect the execution plan depends on the specific database system you are using. Here are some general guidelines for popular database systems:

1. Microsoft SQL Server (T-SQL):

In SQL Server Management Studio (SSMS), you can use the SHOWPLAN or SHOWPLAN\_XML statement or use the "Display Estimated Execution Plan" or "Include Actual Execution Plan" options.’

-- To display the estimated execution plan

SET SHOWPLAN\_ALL ON;

GO

-- Your SQL query here

-- To turn off the display of the estimated execution plan

SET SHOWPLAN\_ALL OFF;

GO

2. MySQL:

In MySQL, you can use the EXPLAIN statement before your query to get the execution plan.

Example:

EXPLAIN SELECT \* FROM your\_table WHERE your\_condition;

3. PostgreSQL:

In PostgreSQL, you can use the EXPLAIN statement similarly to MySQL.

Example:

EXPLAIN SELECT \* FROM your\_table WHERE your\_condition;

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

MAXDOP (Maximum Degree of Parallelism):

Purpose: The MAXDOP option controls the maximum number of processors that can be used for the parallel execution of a single query. Parallel execution is a feature that allows a single query to be divided into multiple tasks that can be executed simultaneously by different processors.

Usage:

SELECT \* FROM your\_table

OPTION (MAXDOP 4);

OPTION (RECOMPILE):

Purpose: The OPTION (RECOMPILE) hint is used to force the SQL Server query optimizer to recompile the query plan every time the query is executed. This can be useful when the optimal plan for a query can vary based on the parameter values passed to the query.

Usage:

SELECT \* FROM your\_table

OPTION (RECOMPILE);

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

Building Data Definition Language (DDL) statements from an existing database table involves extracting the structure and constraints of the table in a format that can be used to recreate the table. Here are the steps you can follow to generate DDL statements for an existing database table:

1. Identify the Database and Table:

Connect to the database management system (DBMS) where your table is located.

Identify the database containing the table of interest.

Identify the specific table for which you want to generate DDL statements.

2. Use SQL Statements to Extract Table Structure:

Execute SQL queries to retrieve information about the table structure. Different DBMS systems have different ways to obtain this information. Below are examples for various databases:

1. MySQL:

SHOW CREATE TABLE your\_table\_name;

1. PostgreSQL:

\d your\_table\_name

1. SQL Server:

sp\_help 'your\_table\_name';

1. Oracle:

DESC your\_table\_name;

3. Capture and Save the Output:

The output of the SQL queries will contain the DDL statements needed to recreate the table. Capture and save this output for further use.

4. Adjust DDL Statements (Optional):

Depending on your requirements, you may need to adjust the generated DDL statements. For example, you might want to modify the table name, change data types, or remove specific constraints.

5. Execute DDL Statements:

Use the captured and adjusted DDL statements to recreate the table in another database or environment.

Be cautious when executing DDL statements, especially in a production environment, as they can modify the database schema.

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, you can use the UPDATE statement along with the FROM clause to specify the table and the join condition. Here's an example using a hypothetical scenario where you want to update values in one table based on matching records in another table:

-- Example: Updating data in a table using an inner join

-- Suppose you have two tables: 'orders' and 'order\_updates'.

-- You want to update the 'status' column in the 'orders' table based on matching 'order\_id' in the 'order\_updates' table.

-- Sample 'orders' table

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

status VARCHAR(50)

);

INSERT INTO orders (order\_id, status)

VALUES (1, 'Pending'),

(2, 'Shipped'),

(3, 'Processing');

-- Sample 'order\_updates' table

CREATE TABLE order\_updates (

order\_id INT PRIMARY KEY,

new\_status VARCHAR(50)

);

INSERT INTO order\_updates (order\_id, new\_status)

VALUES (1, 'Completed'),

(2, 'Delivered');

-- Use an inner join in the UPDATE statement to match records based on 'order\_id'

UPDATE orders

SET status = ou.new\_status

FROM order\_updates ou

WHERE orders.order\_id = ou.order\_id;

-- View the updated 'orders' table

SELECT \* FROM orders;

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

TRUNCATE, DELETE, and DROP are SQL commands used for different purposes in managing data and database objects. Here's a brief differentiation along with examples:

1. TRUNCATE:

Purpose: The TRUNCATE statement is used to remove all rows from a table quickly. It is a DDL (Data Definition Language) statement and is more efficient than DELETE for removing all records from a table because it doesn't log individual row deletions.

Example:

-- Truncate all data from the 'employees' table

TRUNCATE TABLE employees;

2. DELETE:

Purpose: The DELETE statement is used to remove specific rows from a table based on a condition. It is a DML (Data Manipulation Language) statement and provides more flexibility than TRUNCATE by allowing you to specify a condition for which rows to delete.

Example:

-- Delete employees with a salary less than 50000 from the 'employees' table

DELETE FROM employees

WHERE salary < 50000;

3. DROP:

Purpose: The DROP statement is used to remove database objects such as tables, indexes, or views. It is a DDL statement, and once executed, the entire object is deleted along with all associated data.

Example:

-- Drop the 'employees' table and all its data

DROP TABLE employees;