**SQL Assignment 3**

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

Create a function and then call another function from within it. this process is called as

Stored Procedures.

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

1. How to inspect the query's execution plan?

To view a visual explain execution plan, execute your query from the SQL editor and then select Execution Plan within the query results tab. The execution plan defaults to Visual Explain , but it also includes a Tabular Explain view that is similar to what you see when executing EXPLAIN in the MySQL client.

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

The maximum degree of parallelism (MAXDOP) is **a server configuration option for running SQL Server on multiple CPUs**. It controls the number of processors used to run a single statement in parallel plan execution. The default value is 0, which enables SQL Server to use all available processors.

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

Create Table :-

create table customer

(

cust\_id int not null,

cust\_name varchar(50),

contact\_name varchar(50),

address varchar(50),

postalcode varchar(50),

country varchar(10),

PRIMARY KEY(cust\_id)

)

select \* from customer

Alter Query:-

ALTER table

customer

ADD gender

varchar(1)

select \* from customer

Drop Query:-

drop table customer

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

CREATE TABLE table1

(column1 INT,

column2 INT,

column3 VARCHAR (100)

)

select \* from table1

INSERT INTO table1 (column1, column2, column3)

SELECT 1, 11, 'FIRST'

UNION ALL

SELECT 11,12, 'SECOND'

UNION ALL

SELECT 21, 13, 'THIRD'

UNION ALL

SELECT 31, 14, 'FOURTH'

CREATE TABLE table2 (column1 INT, column2 INT, column3 VARCHAR (100))

INSERT INTO table2 (column1, column2, column3)

SELECT 1, 21, 'TWO-ONE'

UNION ALL

SELECT 11, 22, 'TWO-TWO'

UNION ALL

SELECT 21, 23, 'TWO-THREE'

UNION ALL

SELECT 31, 24, 'TWO-FOUR'

select \* from table2

UPDATE table1

SET column2 = table2.column2,

column3 = table2.column3

FROM table1

INNER JOIN table2 ON table1.column1 = table2.column1

WHERE table1.column1 IN (21,31)

**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.

DROP is a DDL Command. Objects deleted using DROP are permanently lost and it cannot be rolled back. Unlike TRUNCATE which only deletes the data of the tables,

the DROP command deletes the data of the table as well as removes the entire schema/structure of the table from the database

CREATE TABLE table2 (column1 INT, column2 INT, column3 VARCHAR (100))

INSERT INTO table2 (column1, column2, column3)

SELECT 1, 21, 'TWO-ONE'

UNION ALL

SELECT 11, 22, 'TWO-TWO'

UNION ALL

SELECT 21, 23, 'TWO-THREE'

UNION ALL

SELECT 31, 24, 'TWO-FOUR'

select \* from table2

#Delete Query

DELETE FROM table2 WHERE column3='TWO-TWO';

#Truncate Query

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself.

TRUNCATE TABLE table2;

# Drop Query :-

Drop table Table2