**SQL Assignment 5**

1. Explain how SQL Query keyword statements are executed in order.

A-

**Query order of execution**

**1. FROM and JOINs**

The **FROM** clause, and subsequent **JOIN**s are first executed to determine the total working set of data that is being queried. This includes sub-queries in this clause, and can cause temporary tables to be created under the hood containing all the columns and rows of the tables being joined.

**2. WHERE**

Once we have the total working set of data, the first-pass **WHERE** constraints are applied to the individual rows, and rows that do not satisfy the constraint are discarded. Each of the constraints can only access columns directly from the tables requested in the **FROM** clause. Aliases in the **SELECT** part of the query are not accessible in most databases since they may include expressions dependent on parts of the query that have not yet executed.

**3. GROUP BY**

The remaining rows after the **WHERE** constraints are applied are then grouped based on common values in the column specified in the **GROUP BY** clause. As a result of the grouping, there will only be as many rows as there are unique values in that column. Implicitly, this means that you should only need to use this when you have aggregate functions in your query.

**4. HAVING**

If the query has a **GROUP BY** clause, then the constraints in the **HAVING** clause are then applied to the grouped rows, discard the grouped rows that don't satisfy the constraint. Like the **WHERE** clause, aliases are also not accessible from this step in most databases.

**5. SELECT**

Any expressions in the **SELECT** part of the query are finally computed.

**6. DISTINCT**

Of the remaining rows, rows with duplicate values in the column marked as **DISTINCT** will be discarded.

**7. ORDER BY**

If an order is specified by the **ORDER BY** clause, the rows are then sorted by the specified data in either ascending or descending order. Since all the expressions in the **SELECT** part of the query have been computed, you can reference aliases in this clause.

**8. LIMIT / OFFSET**

Finally, the rows that fall outside the range specified by the **LIMIT** and **OFFSET** are discarded, leaving the final set of rows to be returned from the query.

1. Explain the advantages of stored procedures and their syntax in relation to recompiling stored procedures.

A-

Use of Recompile makes SQL Server to recompile to SP, hence instead of using existing plan for same SP, Database engine will create new execution plan.

It can be both Automated and Manual process depending upon your requirement.

Too many recompilations can be very bad since for each compilation database engine has to create new execution plan which could be costly operation. So generally automatic recompilation of SP should be avoided, and even manually recompilation should be tested in development environment first before using it production to be sure that it really does help.

1. Give an example of the derived table.

A-

A derived table is a table expression that appears in the FROM clause of a query.

Ex:

CREATE TABLE query (s1 INT, s2 CHAR(5), s3 FLOAT);

INSERT INTO query VALUES (5,'8',1.0);

INSERT INTO query VALUES (6,'3',9.0);

SELECT c1,c2,c3

FROM (SELECT s1 AS c1, s2 AS c2, s3\*2 AS c3 FROM query) AS c

WHERE c1 > 4;

1. What is the database's trigger? Explain the different forms of triggers that can be found in the database.

A-

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs.

Triggers are database objects, actually, a special kind of stored procedure, which “reacts” to certain actions we make in the database. The main idea behind triggers is that they always perform an action in case some event happens.

There are 4 types of triggers:

1. Data Definition Language (DDL) triggers- A DDL trigger executes in response to a change to the structure of a database (for example, CREATE, ALTER, DROP).
2. Data Manipulation Language (DML) triggers- A DML trigger executes in response to a change in data (INSERT, UPDATE, DELETE).
3. CLR triggers- CLR triggers are special types of triggers that are based on the Common Language Runtime (CLR) in the .Net Framework.
4. Logon triggers- This type of trigger is fired against a LOGON event before a user session is established to the SQL Server.
5. What are the benefits and drawbacks of triggers?

A-

Advantages:

1. Helps us to automate the data alterations.

2. Allows us to reuse the queries once written.

3. Provides a method to check the data integrity of the database.

4. Helps us to detect errors on the database level.

5. Allows easy auditing of data.

Disadvantages:

1. Increases the overhead costs of the server.

2. Provides only extended validations i.e. not all validations are accessible in SQL triggers.

3. Troubleshooting errors due to triggers is a tedious job.

4. Can cause logical errors in the application even if a slight mistake in query exists.

5. We could lose the original data if we set a wrong trigger by mistake.

1. Create a stored procedure to call other stored procedures.

A-

Create table employee.tbl(Id INT NOT NULL AUTO\_INCREMENT, Name Varchar(30) NOT NULL, PRIMARY KEY(id))

Create Procedure insert1()

-> BEGIN insert into employee.tbl(name) values ('Ram');

-> END

Create Procedure insert2()

-> BEGIN

-> CALL insert1();

-> Select last\_insert\_id();

-> END

Call insert2();