Name: Ibrahim Mohamed Radi

# Display difference between the following objects in SQL Server

## 1. trigger and stored procedure

| | trigger | stored procedure |
|---|---|---|
| **Basic** | trigger is a stored procedure that runs automatically when various events happen (e.g. update, insert, delete) | Stored procedures are a piece of the code in written in PL/SQL to do some specific task |
| **Running Methodology** | It can execute automatically based on the events | It can be invoked explicitly by the user |
| **Return** | Triggers cannot return values | Stored procedures can return values |
| **Transaction statements** | we can't use transaction statements inside a trigger | We can use transaction statements like begin transaction, commit transaction, and rollback inside a stored procedure |
| **Parameter** | It can not take input as parameter | It can take input as a parameter |

## 2. stored procedure and functions

| | stored procedure | functions |
|---|---|---|
| **return** | **SP can return zero, single or multiple values.** | **Function must return a single value (which may be a scalar or a table).** |
| **transaction** | We can use transaction in SP. | We can't use transaction in UDF. |
| **parameter** | **SP can have input/output parameter.** | **Only input parameter.** |
| **call** | We can call function from SP. | We can't call SP from function. |
| **WHERE/ HAVING statement** | **We can't use SP in SELECT/ WHERE/ HAVING statement.** | **We can use UDF in SELECT/ WHERE/ HAVING statement.** |
| **Try-Catch** | We can use exception handling using Try-Catch block in SP. | We can use exception handling using Try-Catch block in SP. |

## 3. drop and delete statement

| | delete | drop |
|---|---|---|
| **Purpose** | DELETE Command, removes some or all tuples/records from a relation/table | DROP Command, removes named elements of schema like relations/table, constraints or entire schema. |
| **Orientation** | SQL is data-oriented language. | PL/SQL is application-oriented language. |
| **Interaction** | SQL directly interacts with database server. | PL/SQL does not directly interact with database server. |
| **Language** | DELETE is DML. | DROP is DDL. |
| **Clause** | Where clause is used to add filtering. | No where clause is available. |
| **Rollback** | Delete command can be rollbacked as it works on data buffer. | Drop command can't be rollbacked as it works directly on data. |
| **Memory Space** | Table memory space is not free if all records are deleted using Delete Command. | Drop command frees the memory space. |
| **Problem** | DELETE command may face shortage of memory. | DROP Command may cause memory fragmentation. |
| **Objective** | SQL is used to write queries, create and execute DDL and DML statements. | PL/SQL is used to write program blocks, functions, procedures, triggers and packages. |

## 4. select and select into statement

| | select | select into |
|---|---|---|
| **Create** | No Create a new table | Create a new table |
| **Display** | Display Data | No Display Data |

## 5. DDL, DML, DCL and DQL

| | DDL | DML | DCL | DQL |
|---|---|---|---|---|
| **Definition** | Data Definition Language | Data Manipulation Language | Data Control Language | Data Query Language |
| **Example** | Create Alter Drop | Insert Update Delete | Grant Revoke | Select |

## 6. Table valued and multi statement function

| | Table valued<br>ITVF | multi statement<br>MSTVF |
|---|---|---|
| **RETURNS Syntax** | You simply state RETURNS TABLE and the return table's definition will be based on the function's SELECT statement. No need to specify the structure of the return table. | Your RETURNS syntax explicitly specifies the structure of the return table. This is done by declaring a TABLE variable that will be used to store and accumulate the rows that are returned as the value of the function. |
| **The BEGIN/END Syntax** | **ITVFs do not use the BEGIN/END syntax.** | **MSTVFs do use the BEGIN/END syntax.** |
| **Performance** | Generally faster than MTSVFs. | Generally slower than ITVFs. |
| **Data Updates** | **In some cases, it's possible to update data in the underlying tables using an ITFV.** | **You cannot update data in the underlying tables using a MSTVF** |

## 7. Varchar (50) and varchar(max)

| | Varchar (50) | varchar(max) |
|---|---|---|
| **Max Size** | up to 50 B | up to 2 GB |

## 8. SQL and windows Authentication

| | SQL Authentication | windows Authentication |
|---|---|---|
| **login** | Windows principal token in the OS | Username and Password |
| **identity validation** | not ask for a password for identity validation | ask for a password for identity validation |
| **control** | DBAs do not have control over the AD logins and groups | DBAs have control over the AD logins and groups |
| **web applications** | Not supported web applications | support web applications |
| **operating systems** | connect from windows operating systems | connect from cross-operating systems |

## 9. Inline function and view

| | Inline function | view |
|---|---|---|
| **call** | with restriction | without any restriction |
| **Performance** | The same Performance | |
| **return** | returns a value/table | returned from a query |

## 10.Identity and unique Constraint

| | Identity Constraint | unique Constraint |
|---|---|---|
| **value** | unique | unique |
| **Increase** | auto-increasing | No auto-increasing |
| **Count in table** | Only one column at the Table | column at the Table |