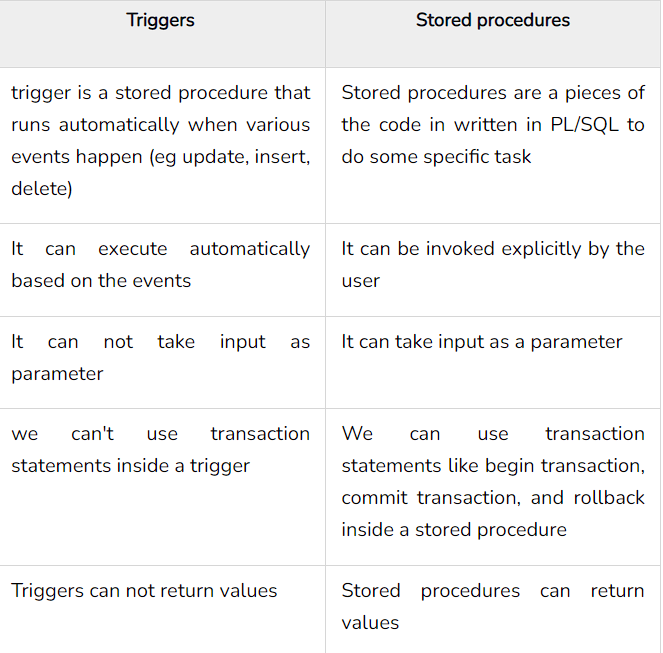
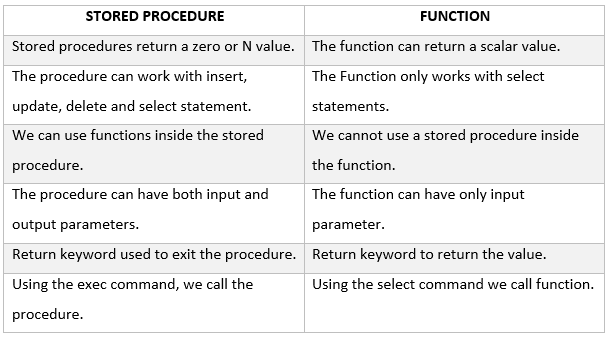
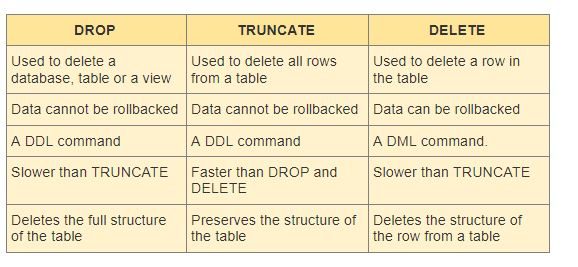
**Batch:** Collection or group of SQL statements. All statements of a batch are compiled into one executional unit called execution plan. All statements are then executed statement by statement.

**Script:**  scripts are the series of commands (sequence of instructions), And these commands could be any combination of DDL (Data Definition Language) or DML (Data Manipulation Language) commands. Therefore, you could change the database structure (CREATE, ALTER, DROP objects) and/or change the data (perform INSERT/UPDATE/DELETE commands).

**Transaction:** is a set of commands that are guaranteed to succeed or fail totally ( it won't complete half the commands and then fail on the rest, if one fails they all fail). And in case of failure it will rollback the executed commands

1. **Trigger and stored Procedure**
2. **Stored Procedure and Functions**
3. **Drop, Truncate and Delete statement**

**Select:** DQL Command used to retrieve data from table or a variable.

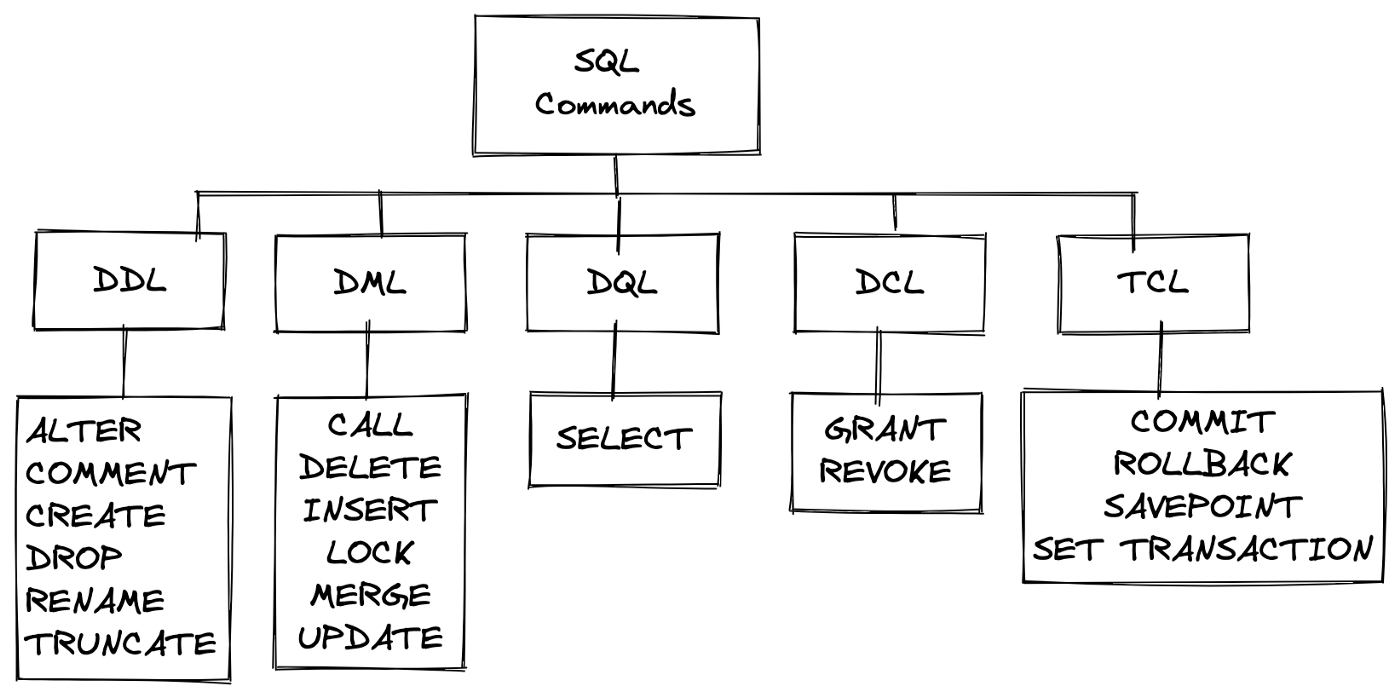
**Select Into:** DDL Command used to create new table based on existing one.

**Global variables:** are pre-defined system variables. It starts with **@@**. It provides information about the present user environment for SQL Server. SQL Server provides multiple global variables, which are very effective to use in Transact-SQL.

**Local** **variables:** local variable is an object that can hold a single data value of a specific type. The life cycle of the variable starts from the point where it is declared and has to end at the end of the batch.

**Convert:** it is function which is a specific function of the SQL server.it does converting and formatting data types at the same time.

**Cast:** The CAST function is ANSI standard and is compatible to use in other databases. it is used to convert a data type without a specific format

1. **DDL,DML,DCL,DQL and TCL**

**XML Raw:**

* Transforms each row in the result set into an XML element.
* You can only present data as elements or attributes.
* It has a problem with join statements so we use auto instead.

**XML Auto:**

* + - * Returns query results in a simple, nested XML tree. Each table in the FROM clause
      * For which at least one column is listed in the SELECT clause is represented as an XML element
      * The columns listed in the SELECT clause are mapped to the appropriate element attributes.

**Table Valued :** In Table Valued functions, you pull *already-existing* columns from *already-existing* tables.

**Multi Statement:** Table Valued function returns the result set of a table variable that is created, defined, and populated within the definition of the function. The advantage of a Multi Statement Table Valued function is you have complete control over the definition of the table variable, including naming your columns, as well as setting up appropriate data types, constraints, indexes.

**11.**

**Varchar(50) :** Used to store strings of variable length in size from 1 to 8000 bytes, Takes up 1 to 4 byte for each character based on collation and requires one or more bytes to store the length of the data.

**varchar(max) :** Similar to the VARCHAR data type in that  supports character strings up to 2 GB (2,147,483,647 bytes) in length. but VARCHAR(MAX) columns cannot be included as a key column of an index and do not allow you to restrict the length of the column.

1. **Datetime, datetime2(7) and datetimeoffset(7)**

**13.**

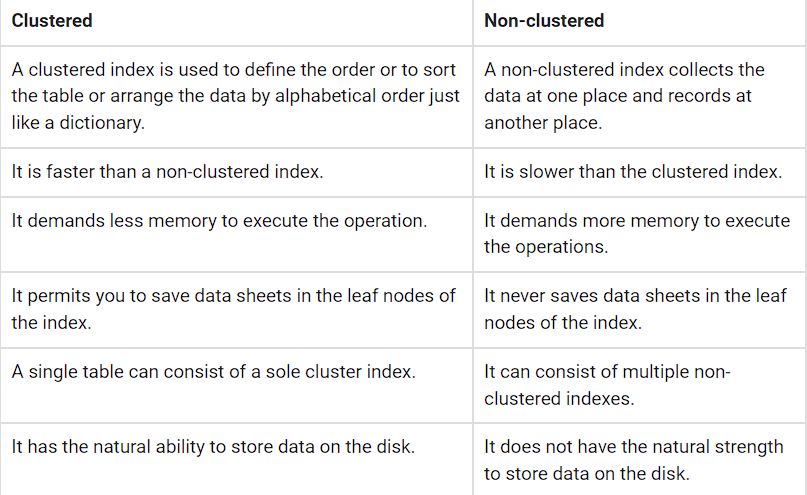
**Default instance:** it is the first SQL service created when installing the SQL server for the first time.

**Named instance:** after creating the default service we can create as many named instanced as we want with different names

**14. SQL and windows Authentication**

**Windows authentication** mode requires users to provide a valid Windows username and password to access the database server. In enterprise environments, these credentials are normally Active Directory domain credentials.

**Mixed** **authentication** mode allows the use of Windows credentials but supplements them with local SQL Server user accounts that the administrator may create and maintain within SQL Server.

**15. Clustered and non-clustered index**

**16.**

**Group by rollup:** The ROLLUP operator is used to calculate sub-totals and grand totals for a set of columns passed to the “GROUP BY ROLLUP” clause.

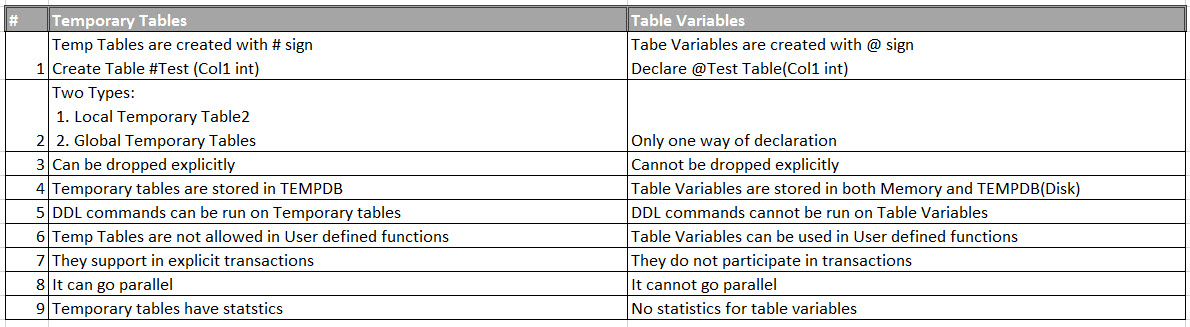
**Group by cube:** The CUBE operator is also used in combination with the GROUP BY clause, however the CUBE operator produces results by generating all combinations of columns specified in the GROUP BY CUBE clause.

**17. Sequence object and identity**

|  |  |
| --- | --- |
| **Sequence** | **Identity** |
| Sequence is a database-level object so it is independent of tables. A Sequence object allows us to synchronize a seed value across the multiple tables. | Identity is a table level object, in other words it is dependent on the table. This property is set or used with the CREATE TABLE and ALTER TABLE statements |
| Using the CYCLE property, we can restart the counter after a specific interval. | We cannot restart the Identity counter after the specified interval. |
| Using the “CACHE” property we can cache a sequence and improve the performance of SQL Server. | We cannot cache the identity column. |
| Using the MAXVALUE property we can define a maximum value for the sequence | We cannot define the maximum value for an Identity column. It is dependent on the data type for the identity column. |
| We can alter the seed well as the Step size of a Sequence object any time | We can reseed an Identity property but we cannot change the step size. |
| We can create a Sequence number in descending order using a sequence object. | We cannot create an Identity property in descending order |

**18. Inline function and view**

|  |  |
| --- | --- |
| **Inline function** | **view** |
| You can't have Update statements in an inline function | but you can write Update statements against them just as though they were an updatable view. |
| Inline function can take parameters | VIEW cannot. |

**19. Table variable and temporary table**

**20.**

**Row\_number() : it doesn’t give the equal rows the same rank.**

**dense\_Rank() : it gives the equal rows the same rank.**

