batch	script	transaction
Number of statement run together but they independent from each other If one fails it executes the others	Number of statement separated by go	Dependent queries run as single unite of work If any error happens, it stops and roll back the work.
trigger	**stored procedure**	
-special type of stored procedure that automatically fires when an event occurs in the database server (update, insert, delete) -Can't be invoked explicitly by the user -Calling a trigger inside another trigger is not possible. -ROLLBACK, COMIT are allowed in triggers. -No parameters	-Some of SQL statements that you can save, so the code can be reused over and over again -can be invoked explicitly by the user. -Procedures can be called inside another given procedure -ROLLBACK and COMMIT are allowed in procedures. -Parameters can be taken as input by a stored procedure.	
stored procedure	**functions**	
-object in DB that is compiled one time and its compiled format is saved, which executes compiled code whenever it is called. -optional to return value -can have input or output parameters - return int -allows SELECT as well as DML(INSERT/UPDATE/DELETE) statement in it. -allow call a function from a stored procedure. -can use Transactions in Procedure.	-A function is compiled and executed every time whenever it is called. -must return a value -can have only input parameters. -return tables or scalar value. -allows SELECT statement. -does not allow call a stored procedure from a function. -can't use Transactions in Function	
drop	**truncate**	**delete**

-DDL -used to drop the whole table -cannot use ROLLBACK -slower than truncate - It does not work with the WHERE clause.	-DDL -used to delete all the rows of a relation - cannot use ROLLBACK -faster than drop - It does not work with the WHERE clause.	-DML -used to delete one or more tuples of a table. -Can restore the data using the COMMIT or ROLLBACK -slow than truncate because it writes in log file - It works with the WHERE clause.
select	**Select into**	
- DQL - used to retrieve data from the database.	- DML - allows you to create a new table and populate it with the result set of a SELECT statement	
local variable	**Global variable**	
- is created when the function is executed, and once the execution is finished, the variable is destroyed. - can only be accessed within the function in which they are declared. - Can be accessed only in its batch	- exists in the program for the entire time the program is executed - declared outside of all the functions or block and can be accessed globally in a program. - It can be accessed by any function present in the program	
convert	**cast**	
CONVERT (type [(length)], expression [, style]) - accepts an optional style parameter that is used for formatting.	CAST (expression AS data_type [(length)])	
DDL	**DML**	**DQL**
- used to define the database structure or schema. - Create, Alter, Drop, Truncate	- used for modifying data with in schema objects - Select into, Insert, Update, Delete, Merge, Call	- used for performing queries on the data within schema objects. - Select statement
DCL	**TCL**	
- used to control access to data stored in a database - Grant, Revoke	- used to manage transactions in the database. - Commit, Rollback	
For xml raw	**for xml auto**	
- returns column values as attributes and wraps each row in a generic row element.	- returns column values as attributes and wraps each row	

- each column value in the rowset that is not NULL is mapped to an attribute of the element.	in an element named after the table from which it came. - This doesn't provide much control over the shape of the XML generated from a query result.	
Table valued	multi statement function	
- user-defined function that returns a table. - state RETURNS TABLE and the return table's definition will be based on the function's SELECT statement.	- user-defined function combines the scalar function's capability to contain complex code with the inline table-valued function's capability to return a result set. - RETURNS syntax explicitly specifies the structure of the return table.	
Varchar(50)	varchar(max)	
stores a maximum of 50 characters.	- Varchar(max) stores a maximum of 2,147,483,647 = 2^31-1 characters (2 GB) VARCHAR(MAX) columns cannot be included as a key column of an index.	
Datetime	datetime2(7)	datetimeoffset(7)
YYYY-MM-DD hh:mm:ss.nnn - Not user-defined fractional second precision	YYYY-MM-DD hh:mm:ss[.nnnnnnn] - User-defined fractional second precision	YYYY-MM-DD hh:mm:ss[.nnnnnnn] [+|-]hh:mm
Default instance	named instance	
- Only one default instance - Type of instance when installing a single instance of SQL server	- Multiple named instances - user specifies an instance name when installing the instance	
SQL	windows Authentication	
- Less secure -We mention the Server name as well as user name along with the password. - you have limited access to the databases inside the server.	- Much more secure than SQL Server Authentication. - Need to mention server name only. It won't require a username and password. - You have all read/write access.	

Clustered and	non-clustered index	
- used to define the order or to sort the table or arrange the data by alphabetical order just like a dictionary. - Faster than non clustered - It demands less memory to execute the operation - single table can consist of one cluster index. - index contains pointer to block but not direct data	- collects the data at one place and records at another place. - Slower than clustered - It demands more memory to execute the operations. -It can consist of multiple non-clustered indexes. - index store both value and a pointer to actual row that holds data.	
Group by rollup	group by cube	
- creates subtotals at any level of aggregation needed, from the most detailed up to a grand total. - generates a result set that shows aggregates for a hierarchy of values in the selected columns	- is an extension similar to ROLLUP, enabling a single statement to calculate all possible combinations of subtotals. - generates a result set that shows aggregates for all combinations of values in the selected columns.	
Sequence object	identity	
- Generates sequential values upon request that can be used in any SQL statement - is a user-defined database object and it is not tied to any specific table meaning its value can be shared by multiple tables. - Can specify start value, increment value, minimum value, maximum value, and cycle option.	- Automatically generates values for a column - is a column property meaning it is tied to the table and can't be shared between different tables and columns - Generates a unique sequential value for a column	
Inline function	view	
-Can't have triggers -Can be built with a parameter - Can be used in the FROM clause of a SELECT statement	-Can have triggers -Can't be built with a parameter - Virtual table defined using a SELECT statement	
Table variable	temporary table	

- involves the effort when you usually create the normal tables. -can be used by the current user only. - resides in the memory and exists only for the duration of a query or stored procedure execution. - Function allows us to use the table variable.	- easy to create and back up data. - can be used by multiple users. - exists temporarily on the SQL Server and can be used to hold temporary data. - Functions cannot use the temp variable	
Row_number()	dense_Rank()	
- generates a unique ranking even with duplicate records	always generate a contiguous sequence of ranks like (1,2,3,...)	