1) Date Functions:

- CURRENT_TIMESTAMP – Returns the current date and time in the format (YYYY – MM - DD), (HH.MM.SS.MMM).

- DATEADD – Adds time / date interval to a date and returns the new date. Its syntax is DATEADD (interval, number, date).

  The interval is of various formats like – YY, QQ, MM, DY, WW, DW, HH, MM, SS, MS. If given positive value to get future date and negative value to get date in the past.

- DATEDIFF – Subtracts 2 dates for the given interval. The interval is same as for the DATEADD. Its syntax is DATEDIFF (interval, date1, date2).

- DATEFROMPARTS – Returns date from the specified parts of year, month and day. Its syntax is DATEFROMPARTS (year, month, day). The year should be in YYYY format.

- DATENAME – Returns the specified part of the date. Its syntax is DATENAME (interval, date). It returns the output as a string.

- DATEPART – Returns the specified part of the date. Its syntax is DATEPART (interval, date). It returns the output as a int.

- DAY – Returns the day of the month for a date. Its syntax is DAY (date).

- GETDATE – Returns the current date along with the time.

- GETUTCDATE – Returns current database UTC date and time.

- ISDATE – Checks is the date is valid and returns 1 if its current and 0 if its false.

- MONTH – Returns the month of the year for a date. Its syntax is MONTH (date).

- SYSDATE – Returns the current system date and time while the SQL server is running.

- YEAR – Returns the current year for a date. Its syntax is YEAR (date).

2) String Functions:

- ASCII – Returns the ASCII value of the first character in a column value. Its syntax is ASCII (character).

- CHAR – Returns the character based on the ASCII code. Its syntax is CHAR (code).

- CHARINDEX – Searches for a substring and returns the position. Its syntax is CHARINDEX (substring, string, start).

- CONCAT – Adds two or more string and displays the output. Its syntax is CONCAT (string1...string n).

- Concat with '+'- Adds two or more string and displays the output. Its syntax is string1+. +stringn.

- CONCAT_WS – Adds two or more string with a separator. Its syntax is CONCAT_WS (separator, string1..., string n).

- DATALENGTH – Returns the length of an expression along with the blank space in bytes.

- DIFFERENCE – Finds the difference between the 2 SOUNDEX values and gives output between (0 - 4). 0 say less similarity while 4 says strong similarity. Its syntax is DIFFERENCE (string1, string2).

- FORMAT – Used to specify a value in a specific format. Its syntax is FORMAT (value, format, culture).

- LEFT – Extracts a number of characters from a string, starting from the left. Its syntax is LEFT (string, number_of_characters).

- LEN – Returns the length of a string. Trailing spaces at the end of the string is not included. Its syntax is LEN (string).

- LOWER – Converts it into a lower class.

- LTRIM – Removes the leading spaces from a string.

- NCHAR – Returns Unicode character based on number code. Its syntax is NCHAR (number code).

- SOUNDEX – Returns the sound ex value for an expression. Its syntax is SOUNDEX (expression).

- SPACE – Returns a string with the specified number of space characters. Its syntax is SPACE (number).

- STR – Returns a number as a string. Its syntax is STR (number, length, decimals).

- STUFF – Deletes a part of a string and insert a string at that position specified. Its syntax is STUFF (string, start, length, new string).

3) Ranking Functions:

- ROW_NUMBER () - This is a ranking function which returns a unique sequential number for each row within the partition of the specified window. It starts at 1 for the first row and increments without skipping or duplicate values.

- RANK () - This returns a unique rank number for each distinct row within the partition according to the specified column value. It starts at 1 for the first row in each partition, with the same rank for duplicate values and leaving gaps between the ranks. This gap appears in the sequence after the duplicate values.

- DENSE_RANK () - It also generates unique rank for each number for each distinct row like RANK (). It starts at 1 for the first row in each partition, ranking the rows with equal values with the same rank number, except that it does not skip any rank, leaving no gaps between the ranks.

- NTILE () - It is used to distribute the rows in the rows set into a specified number of groups, providing each row in the row set with a unique group number, starting with the number 1 that shows the group this row belongs to, where N is a positive number, which defines the number of groups you need to distribute the rows set into.

4) Procedures:

A stored procedure is a prepared SQL code that you can save, so the code can be reused repeatedly. So, if you have an SQL query that you write repeatedly, 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 that is passed.

SYNTAX: CREATE PROCEDURE procedure_name GO;

EXEC procedure_name;

5) Functions:

A function is a stored program that you can pass parameters into and return a value. The types of functions are,

- **Aggregate functions:** These functions are used to do operations from the values of the column and a single value is returned. AVG (), COUNT (), FIRST (), LAST (), MAX (), MIN ().
- **Scalar functions:** These functions are based on user input. These too returns single value. UCASE (), LCASE (), MID (), LEN (), ROUND (), NOW ().

6) Triggers:

The various types of triggers: Instead of, for, after, before.

Instead Of: An INSTEAD OF trigger is a trigger that allows you to skip an INSERT, DELETE, or UPDATE statement to a table or a view and execute other statements defined in the trigger instead. The actual insert, delete, or update operation does not occur at all.

For: A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

After: AFTER triggers are often used for complex data validation. These triggers can rollback, or undo, the insert, update, or delete if the code inside the trigger doesn't like the operation. The code can also do something else or even fail the transaction. But if the trigger doesn't explicitly ROLLBACK the transaction, the data modification operation will go as it was originally intended. AFTER triggers report an error code if an operation is rolled back. AFTER trigger takes place after the modification but before the implicit commit, so the transaction is still open when the AFTER trigger is fired, that is what the main advantage of using AFTER trigger.

Logon: Logon triggers are a special type of triggers that gets executed when a LOGON event of SQL Server is raised. This event is raised when a user session is being established with SQL Server that is made after the authentication phase finishes, but before the user session is established. Hence, all messages that we define in the trigger, such as error messages, will be redirected to the SQL Server error log.

7) Magic Tables:

Magic tables are nothing but inserted and deleted which are temporary objects created by the server internally. This is created to hold recently deleted values. On insertion of a record into that table, the inserted table will be created automatically by the database. On deletion of record from that table, the deleted table will be created automatically by the database.