1. batch, script and transaction

|  |  |  |
| --- | --- | --- |
| **Batch** | **Script** | **Transaction** |
| Is just that a collection of commands that need to be executed without guaranteed of success or fail. | Is a set of SQL commands saved as a file in SQL Scripts. A SQL script can contain one or more SQL statements or PL/SQL blocks. | a collection of commands that are guaranteed to succeed or fail totally |
| Don’t allow rollback | You can use SQL Scripts to create, edit, view, run, and delete database objects. | allows you to rollback/undo changes. |

1. trigger and stored procedure

|  |  |
| --- | --- |
| **Trigger** | **Stored Procedure** |
| Is a stored procedure that runs automatically when events (update, insert, delete) happen | are a piece of the code in written in SQL to do some specific task |
| can execute(fire) automatically based on the events | can be invoked explicitly by the user |
| cannot take input as parameter | Can take input as parameter |
| cannot return values | can return values |

1. stored procedure and functions

|  |  |
| --- | --- |
| **Functions** | **Stored Procedures** |
| A function has a return type and returns a value. | A procedure does not have a return type. But it returns values using the OUT parameters. |
| You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions. | You can use DML queries such as insert, update, select etc… with procedures. |
| A function does not allow output parameters | A procedure allows both input and output parameters. |
| You cannot manage transactions inside a function. | You can manage transactions inside a procedure. |
| You can call a function using a select statement. | You cannot call a procedure using select statements. |

1. drop, truncate and delete statement

|  |  |  |
| --- | --- | --- |
| DELETE | DROP | TRUNCATE |
| DML | DDL | DDL |
| The delete statement is used to remove single or multiple records from an existing table depending on the specified condition. | Command drops the complete table from the database. | The truncate command removes the complete data from an existing table but not the table itself. It preserves the table structure or schema. |
| Can restore any deleted row or multiple rows from the database using the ROLLBACK command. | Can not get the complete table deleted from the database using the ROLLBACK command. | Can not restore all the deleted rows from the database using the ROLLBACK command. |
| The DELETE command deletes each record individually, making it slower than a TRUNCATE command. | The TRUNCATE command is faster than both DROP and DELETE commands. | DROP is quick to execute but slower than TRUNCATE because of its complexities. |

1. select and select into statement

|  |  |
| --- | --- |
| SELECT | SELECT INTO |
| used to select data from a database.  The data returned is stored in a result table, called the result-set. | copies data from one table by select statment into a new table. |

1. local and global variables

|  |  |
| --- | --- |
| **Local** | **Global** |
| declared inside a function. | declared outside the function. |
| Local variables are created when the function starts its execution and are lost when the function ends | Global variables are created as the execution of the program begins and are lost when the program is ended |
| the value of the local variable is modified in one function, the changes are not visible in another function. | he value of the global variable is modified in one function changes are visible in the rest of the program |
| Accessible inside the function | Accessible throughout the code |

1. convert and cast statements

|  |  |
| --- | --- |
| **Convert** | **Cast** |
| The Convert statement is used to convert expressions from one type to another, in many cases there will be a need within a stored procedure | The Cast statement  is used to convert a data type variable or data from one data type to another data type |

1. DDL, DML, DCL, DQL and TCL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DDL** | **DML** | **DCL** | **DQL** | **TCL** |
| consists of the SQL commands that can be used to define the database schema | commands that deals with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements | includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system. | statements are used for performing queries on the data within schema objects.  The purpose of the DQL Command is to get some schema relation based on the query passed to it. | a set of tasks into a single execution unit. Each transaction begins with a specific task and ends when all the tasks in the group successfully complete |
| Commands :  CREATE: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).  DROP: This command is | Commands:  INSERT : It is used to insert data into a table.  UPDATE: It is used to update existing data within a table.  DELETE : It is used to delete records from a database table. | Commands:  GRANT: This command gives users access privileges to the database  REVOKE: This command withdraws the user’s access privileges given by using the GRANT command. | SELECT: It is used to retrieve data from the database. | COMMIT  Commits a Transaction  ROLLBACK: Rollbacks a transaction in case of any error occurs. |

1. For xml raw and for xml auto

|  |  |
| --- | --- |
| **For** | **Auto** |
| Is responsible for generating the XML content | Converts each column in the SQL table into an attribute in the corresponding XML document. |

1. Table valued and multi statement function

|  |  |
| --- | --- |
| **Table valued** | **multi statement** |
| simply state RETURNS TABLE and the return table’s definition will be based on the function’s SELECT statement | Your RETURNS syntax explicitly specifies the structure of the return table |
| do not use the BEGIN/END syntax | use the BEGIN/END syntax. |
| Generally faster than MTSVFs. | Generally slower than ITVFs. |

1. Varchar (50) and varchar(max)

|  |  |
| --- | --- |
| Varchar (50) | varchar(max) |
| Non-Unicode **Var**iable Length **char**acter data type. | Non-Unicode **large** **Var**iable Length **char**acter data type. |
| Can range in size from 1 to 8000 bytes | it supports character strings up to 2 GB (2,147,483,647 bytes) in length |
| the length of the data needs to be stored along with the data | columns cannot be included as a key column of an index. |
| You can create index on Varchar column data type. | Index can’t be created on a Varchar(Max) data type columns. |

12. Datetime, datetime2(7) and datetimeoffset(7)

|  |  |  |
| --- | --- | --- |
| **Datetime** | **datetime2()** | **datetimeoffset** |
| YYYY-MM-DD hh-mm-ss.nnn | n represents fractional seconds precision  YYYY-MM-DD hh-mm-ss.nnnnnnn | stores the date-time value, as well as an offset indicating how far that date-time departs from UTC.  DATETIMEOFFSET(fractional seconds precision) |
| Character length :  19 positions min  23 max  Storage : 8 bytes | 19 positions min  27 Max  Storage 6 – 8 bytes | 26 Position min  34 max  8 – 10 bytes |

13. Default instance and named instance

|  |  |
| --- | --- |
| **Default instance** | **named instance** |
| One instance can be the default instance  The default instance has no name | A named instance is one where you specify an instance name when installing the instance |

14. SQL and windows Authentication

|  |  |
| --- | --- |
| **Windows Authentication** | **SQL Server Authentication** |
| SQL Server does not ask for a password for identity validation. Therefore, Windows confirms users’ identities for authentication. | Users need to specify the login and password while connecting to SQL Server |

15. Clustered and non-clustered index

|  |  |
| --- | --- |
| Clustered index | non-clustered index |
| It is faster than a non-clustered index. | It is slower than the clustered index. |
| A single table can consist of a sole cluster index. | It can consist of multiple non-clustered indexes. |
| It has the natural ability to store data on the disk. | It does not have the natural strength to store data on the disk. |

16. Group by rollup and group by cube.

|  |  |
| --- | --- |
| CUBE | ROLLUP |
| Produces all possible combinations of subtotals specified in GROUP BY clause and a Grand Total | Produces only some possible subtotal combinations. |

17. Sequence object and identity

|  |  |
| --- | --- |
| SEQUENCE | IDENTITY |
| not need to use insert statment to be increased because it is used with update  cycle | is tied to a particular table |
| shared between tables | cannot be shared among multiple tables since it is a table column property. |
| here is no (insent identity on and off) not needed | To generate the next IDENTITY value, a new row has to be inserted into the table |
| SEQUENCE object can be reset | IDENTITY property cannot be reset to its initial value. |

18. Inline function and view

|  |  |
| --- | --- |
| Inline function | view |
| run statements that return a single value | Views are similar to inline table valued function - they allow you centralize a query in an object that can be easily called from other queries |
| Accept parameter | virtual table created from a result set of a SQL query |
| Can’t have trigger | Can have trigger |

19. Table variable and temporary table

|  |  |
| --- | --- |
| Table variable | temporary table |
| does not have any effect on the Transaction. | can be part of Transaction. |
| will store in the physical memory for some of the data | Temp table will be stored in the tempdb |

20. Row\_number() and dense\_Rank() function

|  |  |
| --- | --- |
| Row\_number() | dense\_Rank() |
| RowNumber returns the sequential number of a row within a partition of a result set, starting at 1 for the first row in each partition. | returns the rank of each row within the partition of a result set |