Question 10

1. batch, script and transaction

BATCH:

a set of Transact-SQL statements that are submitted together and executed as a group, one after the other. A batch is terminated by an end-of-batch signal. the word “go” on a line by itself.

If you have error in one statement in batch, Adaptive Server returns an error message for that statement and returns results for all the others.

Examples: Explicit batches, Stored procedures, Arrays of parameters

Transaction:

sequence of operations performed (using one or more SQL statements) on a database as a single logical unit of work. The effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database). A database transaction must be atomic, consistent, isolated and durable.

Script:

is a collection of structured query language (SQL) commands that are stored in a text file and perform some operation or task. These tasks are usually repetitive. The commands contained in a script can be any supported by the SQL language. When combined, they're typically flexible enough to be used in a variety of related situations. SQL scripts have a number of advantages, including:

Ease of use - The scripts can be saved and loaded when needed.

Consistent operation - If the scripts work correctly the first time, they'll work correctly every time.

Decrease in errors - Manually entered commands are susceptible to human error. Scripts reduce that possibility.

Scheduled operation - Scripts can be scheduled to run at a time when it's convenient, or when a human isn't present.

2. Trigger and stored procedure

Trigger	stored procedure
is a stored procedure that instructs the system to take one or more actions when a specific change is attempted. By preventing incorrect, unauthorized, or inconsistent changes to data, triggers help maintain the integrity of a database.	is a collection of SQL statements and optional control-of-flow statements stored under a name. The creator of a stored procedure can also define parameters to be supplied when the stored procedure is executed.
Doesn't take parameters and no return. Triggers can protect referential integrity—enforcing rules about the relationships among data in different tables. triggers go into effect when a user attempts to modify data with an insert, delete, or update command.	All Transact-SQL extensions support return values from stored procedures, user-defined return status from stored procedures, and the ability to pass parameters from a procedure to its caller.
It can execute automatically based on the events.	It can be invoked explicitly by the user.
We cannot define a trigger inside another trigger. Within a trigger you can call stored procedures	We can define procedures inside another procedure. Also, we can use functions inside the stored procedure. you cannot call l a trigger from a stored procedure

3. Stored procedure and functions

Stored procedure	functions
Stored procedures are precompiled objects which are compiled for the first time and its compiled format is saved, which executes (compiled code) whenever it is called.	A function is compiled and executed every time whenever it is called. A function must return a value and cannot modify the data received as parameters.
does not have a return type. But it returns values using the OUT parameters.	A function has a return type and returns a value.
You can use DML queries such as insert, update, select etc. with procedures.	You cannot use a function with Data Manipulation queries. Only Select queries are allowed in functions.
A procedure allows both input and output parameters.	A function does not allow output parameters
You can call a function from a stored procedure.	You cannot call stored procedures from a function.
You cannot call a procedure using select statements.	You can call a function using a select statement.

4. Drop, Truncate and Delete.

Delete	Drop	Truncate
it is a Data Manipulation Command (DML).	It is a Data Definition Language Command (DDL).	It is also a Data Definition Language Command (DDL).
It is use to delete the one or more tuples of a table.	It is use to drop the whole table.	used to delete all the rows of a relation (table) in one go.
With the help of "DELETE" command we can either delete all the rows in one go or can delete row one by one. i.e., we can use it as per the requirement or the condition using Where clause. It is comparatively slower than TRUNCATE cmd.	With the help of "DROP" command we can drop (delete) the whole structure in one go i.e., it removes the named elements of the schema. By using this command, the existence of the whole table is finished or say lost.	With the help of "TRUNCATE" command we can't delete the single row as here WHERE clause is not used. By using this command, the existence of all the rows of the table is lost. It is comparatively faster than delete command as it deletes all the rows fastly.
Here we can use the "ROLLBACK" command to restore the tuple.	Here we can't restore the table by using the "ROLLBACK" command.	Here we can't restore the tuples of the table by using the "ROLLBACK" command.

5. Select and select into.

Select	Select into
used to select data from a database table	creates a new table and puts the data in it.
The data returned is stored in a result table, called the result-set.	The new table will be created with the column-names and types as defined in the old table.

6. Local and global variables.

Local variables	Global variables
Declared by the user and are often used as counters for while loops or if...else blocks in a batch or stored procedure.	are system-supplied, predefined variables. They are distinguished from local variables by the two @ signs preceding their names.
To declare a local variable's name and datatype use: declare @variable_name datatype	@@error. The two @ signs are considered part of the identifier used to define the global variable.
When you declare a variable, it has the value NULL.	Some global variables provide information to use in transactions.
Assign values to local variables with a select statement: select @a = 1, @b = 2, @c = 3	Users cannot create global variables and cannot update the value of global variables directly in a select statement.

7. Convert and cast statements.

functions are both used to convert data from one data type to another

Convert statements	Cast statements
CONVERT is SQL implementation-specific	CAST is part of the ANSI-SQL specification
it accepts an optional style parameter that is used for formatting. For example, when converting a Date Time datatype to Varchar, you can specify the resulting date's format, such as YYYY/MM/DD or MM/DD/YYYY..	It can't do that.
in the CONVERT function, there is an additional parameter called style specifies the format of the data type after conversion.	It doesn't have a formatting style, as a result there is no parameters can do this.

8. DDL, DML, DCL, DQL and TCL.

DDL (Data Definition Language) :

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

Examples of DDL commands:

a. CREATE – is used to create the database or its objects (like table, index, function, views, store procedure and triggers).
b. DROP – is used to delete objects from the database.
c. ALTER-is used to alter the structure of the database.
d. TRUNCATE–is used to remove all records from a table, including all spaces allocated for the records are removed.
e. COMMENT –is used to add comments to the data dictionary.
f. RENAME –is used to rename an object existing in the database.

DQL (Data Query Language):

DML statements are used for performing queries on the data within schema objects. The purpose of DQL Command is to get some schema relation based on the query passed to it.

Example of DQL:

SELECT – is used to retrieve data from a database.

DML (Data Manipulation Language):

The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements.

Examples of DML:

1. INSERT – is used to insert data into a table.
2. UPDATE – is used to update existing data within a table.
3. DELETE – is used to delete records from a database table.

DCL (Data Control Language):

DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

Examples of DCL commands:

a. GRANT-gives user’s access privileges to database.
b. REVOKE-withdraw user’s access privileges given by using the GRANT command.

TCL (transaction Control Language):

TCL commands deals with the transaction within the database.

Examples of TCL commands:

a. COMMIT– commits a Transaction.
b. ROLLBACK– rollbacks a transaction in case of any error occurs.
c. SAVEPOINT–sets a save point within a transaction.
d. SET TRANSACTION–specify characteristics for the transaction.

9. For xml row and for xml auto

For xml row	For xml auto
Each column becomes an attribute in the result set, and each row becomes an element with the generic name of row.	each row is named after the table

10. Table valued and multi statements function

Table valued	Multi statements function
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.
ITVFs do not use the BEGIN/END syntax.	MSTVFs do use the BEGIN/END syntax
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.

11. Varchar (50) and varchar(max)

Varchar (50)	Varchar(max)
Non-Unicode Variable Length character data type. Example: DECLARE @Name VARCHAR (50) = 'BASAVARAJ' SELECT @Name	Non-Unicode large Variable Length character data type. Example: DECLARE @Name VARCHAR(Max) = 'BASAVARAJ' SELECT @Name
It can store maximum 8000 NonUnicode characters (i.e., maximum storage capacity is 8000 bytes of storage). Optional Parameter n value can be from 1 to 8000.	it can store maximum of 2 147 483 647 Non-Unicode characters (i.e., maximum storage capacity is: 2GB).
If we know that data to be stored in the column or variable is less than or equal to 8000 characters, then we can use this data type. For example, First Name, Last Name etc., columns value can't cross the max 8000 characters limit, in such scenario's it is better to use this data type.	If we know that the data to be stored in the column or variable can cross an 8KB Data page, then we can use this data type.

12. Datetime (3), datetime2(7) and datetimeoffset (7)

Datetimeoffset	Datetime2	Datetime
Date Range: 0001-01-01 through 9999-12-31	0001-01-01 through 9999-12-31	1753-01-01 thru 9999-12-31
Time zone offset range: -14:00 through +14:00	None	None
Storage Size: 8 to 10 bytes, depending on the precision* * Plus 1 byte to store the precision	6 to 8 bytes, depending on the precision* * Plus 1 byte to store the precision	8 bytes
Character Length: 26 positions minimum 34 maximum	19 positions minimum 27 maximum	19 positions minimum 23 maximum
Accuracy: 100 nanoseconds	100 nanoseconds	Rounded to increments of .000, .003, or .007 seconds
2025-05-21 10:15:30.5555555 +07:30	2025-05-21 10:15:30.5555555	2025-05-21 0:15:30.557

13. Default instance and named instance

Named instance	Default instance
the user determines a named instance during the setup. Also, it is possible to install SQL Server as a named instance without installing the default instance first.	If there is no SQL server installed yet, a default instance will be created unless the user specifies a named instance. There can only be one default instance. The default instance name is MSSQLSERVER.
instance where the user specifies an instance name when installing the instance.	used when installing a single instance of SQL server.
can have multiple named instances.	There is only one default instance

14. SQL and windows Authentication

SQL	Windows
SQL Authentication is the typical authentication used for various database systems, composed of a username and a password.	When you are accessing SQL Server from the same computer it is installed on, you shouldn't be prompted to type in a username and password.
shared servers where different users should have access to different databases, SQL authentication should be used.	with Windows Authentication, the SQL Server service already knows that someone is logged in into the operating system with the correct credentials, and it uses these credentials to allow the user into its databases.
can be used in all the other type of cases different from Intranet type	must be used when working in an Intranet type of an environment.

15. Clustered and non-clustered index

Clustered	Non-clustered
faster	slower
requires less memory for operations.	requires more memory for operations.
In clustered index, index is the main data	In Non-Clustered index, index is the copy of data.
A table can have only one clustered index.	A table can have multiple non-clustered indexes.
Clustered index has inherent ability of storing data on the disk.	Non-Clustered index does not have inherent ability of storing data on the disk
store pointers to block not data.	store both value and a pointer to actual row that holds data.
In Clustered index leaf nodes are actual data itself.	leaf nodes are not the actual data itself rather they only contain included columns.
In Clustered index, Clustered key defines order of data within table.	In Non-Clustered index, index key defines order of data within index.
is a type of index in which table records are physically reordered to match the index.	is a special type of index in which logical order of index does not match physical stored order of the rows on disk.

16. Group by rollup and group by cube

rollup	cube
ROLLUP operator is used to calculate sub-totals and grand totals for a set of columns passed to the "GROUP BY ROLLUP" clause.	the CUBE operator produces results by generating all combinations of columns specified in the GROUP BY CUBE clause.
It's an additional switch to GROUP BY clause. It can be applied to all aggregation functions to return cross tabular result sets.	It's an extension to GROUP BY clause. It's used to extract statistical and summarized information from result sets. It creates groupings and then applies aggregation functions on them.

17. Sequence object and identity

Sequence object	Identity object
On the flip side the SEQUENCE object is defined by the user and can be shared by multiple tables since is it is not tied to any table.	IDENTITY property is tied to a particular table and cannot be shared among multiple tables since it is a table column property.
On the other hand, the next VALUE for a SEQUENCE object can simply be generated using the NEXT VALUE FOR clause with the sequence object.	To generate the next IDENTITY value, a new row has to be inserted into the table.
the value for the SEQUENCE object can be reset.	The value for the IDENTITY property cannot be reset to its initial value
the maximum value for a SEQUENCE object can be defined.	A maximum value cannot be set for the IDENTITY property.

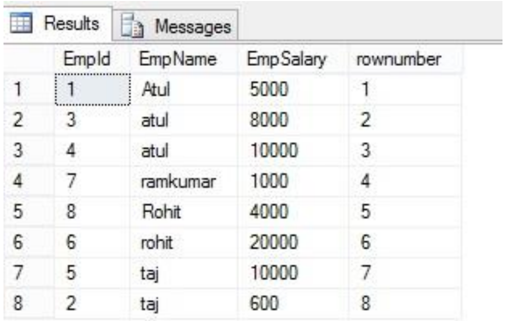
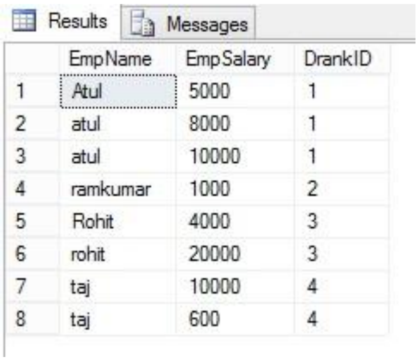
18. Inline table function and view

Inline function	View
Can't be indexed and its performance decrease when number of rows increase	View can be materialized (indexed view) and hence Performs better
Can't have triggers	Views can have triggers since they can be used to change underlying tables (instead of triggers)
The inline table function accepts parameters	Don't accept parameters
CROSS APPLY can be used with it	CROSS APPLY can't be used with it

19. Table variable and temporary table

Table variable	temporary table
Table Variable acts like a variable and exists for a particular batch of query execution. It gets dropped once it comes out of batch. It is created in the memory database but may be pushed out to tempdb.	Temporary Tables are physically created in the tempdb database. These tables act as the normal table and also can have constraints, index like normal tables.
Its scope is within the patch	the scope of any particular temporary table is the session in which it is created

20. Row_number() and dense_Rank() function

Row_number	Dense_Rank
Row_Number() will generate a unique number for every row, even if one or more rows has the same value.	DENSE_RANK () will assign the same number for the row which contains the same value without skipping the next number.
select *, ROW_NUMBER() over(order by EmpName) as row number from Employe	SELECT EmpName ,EmpSalary ,DENSE_RANK() over(order by EmpName) as DrankID from Employe
	

	EmpId	EmpName	EmpSalary	rownumber
1	1	Atul	5000	1
2	3	atul	8000	2
3	4	atul	10000	3
4	7	ramkumar	1000	4
5	8	Rohit	4000	5
6	6	rohit	20000	6
7	5	taj	10000	7
8	2	taj	600	8

	EmpName	EmpSalary	DrankID
1	Atul	5000	1
2	atul	8000	1
3	atul	10000	1
4	ramkumar	1000	2
5	Rohit	4000	3
6	rohit	20000	3
7	taj	10000	4
8	taj	600	4