1) Difference between Stored procedure and User defined functions

Function (UDF)	**Stored Procedure**
Must return a value.	Return value is optional. Can return zero or more values.
Must be a part of an SQL statement to get executed.	Can be executed using EXECUTE or EXEC command.
Functions can be called from Procedure.	Procedures cannot be called from UDF.
Can have only input parameters.	Can have input/output parameters.
UDF can be used in the SQL statements anywhere in the WHERE / HAVING / SELECT sections.	Cannot be used in the SQL statements anywhere in the WHERE / HAVING / SELECT sections.
Inline functions support only one SELECT statement.	Supports any number of select statements.
Allows only SELECT statement in it.	Allows SELECT as well as DML statements.

2) What is Cartesian product of the table?

The CARTESIAN JOIN or CROSS JOIN returns the Cartesian product of the sets of records from two or more joined tables. Thus, it equates to an inner join where the join-condition always evaluates to either True or where the join-condition is absent from the statement.

Ex:

```
SELECT Student.FirstName, emp.empID
FROM Student
CROSS JOIN emp
```

FirstName	empID
Vikas	101
Piyush	101
Rahul	101
Arjun	101
Vikas	103

Piyush	103
Rahul	103
Arjun	103

3) What is Script and batches?

Script:

A SQL script 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. You can use SQL Scripts to create, edit, view, run, and delete script files.

Batches:

A batch of SQL statements is a group of two or more SQL statements or a single SQL statement that has the same effect as a group of two or more SQL statements. In some implementations, the entire batch statement is executed before any results are available. This is often more efficient than submitting statements separately, because network traffic can often be reduced and the data source can sometimes optimize execution of a batch of SQL statements. In other implementations, calling SQLMoreResults triggers the execution of the next statement in the batch.

4) Explore classes and its methods of Environment class

Namespace: System
Provides information about, and means to manipulate, the current environment and platform. This class cannot be inherited.

ExpandEnvironmentVariables(String)	Replaces the name of each environment variable embedded in the specified string with the string Equivalent of the value of the variable, then returns the resulting string.
FailFast(String)	Immediately terminates a process after writing a message to the Windows Application event log, and then includes the message in error reporting to Microsoft.
FailFast(String, Exception)	Immediately terminates a process after writing a message to the Windows Application event log, and then includes the message and exception information in error reporting to Microsoft.
GetCommandLineArgs()	Returns a string array containing the command-line arguments for the current process.

GetEnvironmentVariable(String)	Retrieves the value of an environment variable from the current process.
GetEnvironmentVariable(String, EnvironmentVariableTarget)	Retrieves the value of an environment variable from the current process or from the Windows operating system registry key for the current user or local machine.
GetEnvironmentVariables()	Retrieves all environment variable names and their values from the current process.
GetEnvironmentVariables (EnvironmentVariableTarget)	Retrieves all environment variable names and their values from the current process, or from the Windows operating system registry key for the current user or local machine.
GetFolderPath(Environment+SpecialFolder)	Gets the path to the system special folder that is identified by the specified enumeration.
GetFolderPath(Environment+SpecialFolder, Environment+SpecialFolderOption)	Gets the path to the system special folder that is identified by the specified enumeration, and uses a specified option for accessing special folders.
GetLogicalDrives()	Returns an array of string containing the names of the logical drives on the current computer.
SetEnvironmentVariable(String, String)	Creates, modifies, or deletes an environment variable stored in the current process.
SetEnvironmentVariable(String, String, EnvironmentVariableTarget)	Creates, modifies, or deletes an environment variable stored in the current process or in the Windows operating system registry key reserved for the current user or local machine.

5) Explore classes and its methods of Object

Namespace: System
Supports all classes in the .NET class hierarchy and provides low-level services to derived classes. This is the ultimate base class of all .NET classes; it is the root of the type hierarchy.

Equals(Object)	Determines whether the specified object is equal to the current object.
Equals(Object, Object)	Determines whether the specified object instances are considered equal.
Finalize()	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.
GetHashCode()	Serves as the default hash function.
GetType()	Gets the Type of the current instance.

MemberwiseClone()	Creates a shallow copy of the current Object.
ReferenceEquals(Object, Object)	Determines whether the specified Object instances are the same instance.
ToString()	Returns a string that represents the current object.