Intelligent Data Management with SQL Server

Session: 13

Programming Transact-SQL

Objectives

- Describe Transact-SQL programming
- Describe program flow statements
- Describe various Transact-SQL functions
- Explain the procedure to create and alter User-Defined Functions (UDFs)
- Explain creation of windows with OVER
- Describe window functions

Introduction

- Transact-SQL programming is a procedural language extension to SQL.
- Transact-SQL programming is extended by adding the subroutines and programming structures similar to high-level languages.
- Like high-level languages, Transact-SQL programming also has rules and syntax that control and enable programming statements to work together.
- Users can control the flow of programs by using conditional statements such as IF and loops such as WHILE.

Transact-SQL Programming Elements 1-3

- Transact-SQL programming elements enable to perform various operations that cannot be done in a single statement.
- Users can group several Transact-SQL statements together by using one of the following ways:

Batches

- A batch is a collection of one or more Transact-SQL statements that are sent as one unit from an application to the server.

Stored Procedures

- A stored procedure is a collection of Transact-SQL statements that are precompiled and predefined on the server.

Transact-SQL Programming Elements 2-3

Triggers

- A trigger is a special type of stored procedure that is executed when the user performs an event such as an INSERT, DELETE, or UPDATE operation on a table.

Scripts

- A script is a chain of Transact-SQL statements stored in a file that is used as input to the SSMS code editor or sqlcmd utility.

Transact-SQL Programming Elements 3-3

Following features enable users to work with Transact-SQL statements:

Variables

- A variable allows a user to store data that can be used as input in a Transact-SQL statement.

Control-of-flow

- Control-of-flow is used for including conditional constructs in Transact-SQL.

Error Handling

- Error handling is a mechanism that is used for handling errors and provides information to the users about the error occurred.

Transact-SQL Batches 1-2

A Transact-SQL batch is a group of one or more Transact-SQL statements sent to the server as one unit from an application for execution.

In the execution plan, the SQL statements are executed one by one.
It should be terminated with a semicolon.

A compile error such as syntax error restricts the compilation of the execution plan.
So, if a compile-time error occurs, no statements in the batch are executed.

Transact-SQL Batches 2-2

A run-time error such as a constraint violation or an arithmetic overflow has one of the following effects:

- Most of the run-time errors stop the current statement and the statements that follow in the batch.
- A specific run-time error such as a constraint violation stops only the existing statement and the remaining statements in the batch are executed.

Transact-SQL Variables 1-3

- Variables allow users to store data for using as input in a Transact-SQL statement.

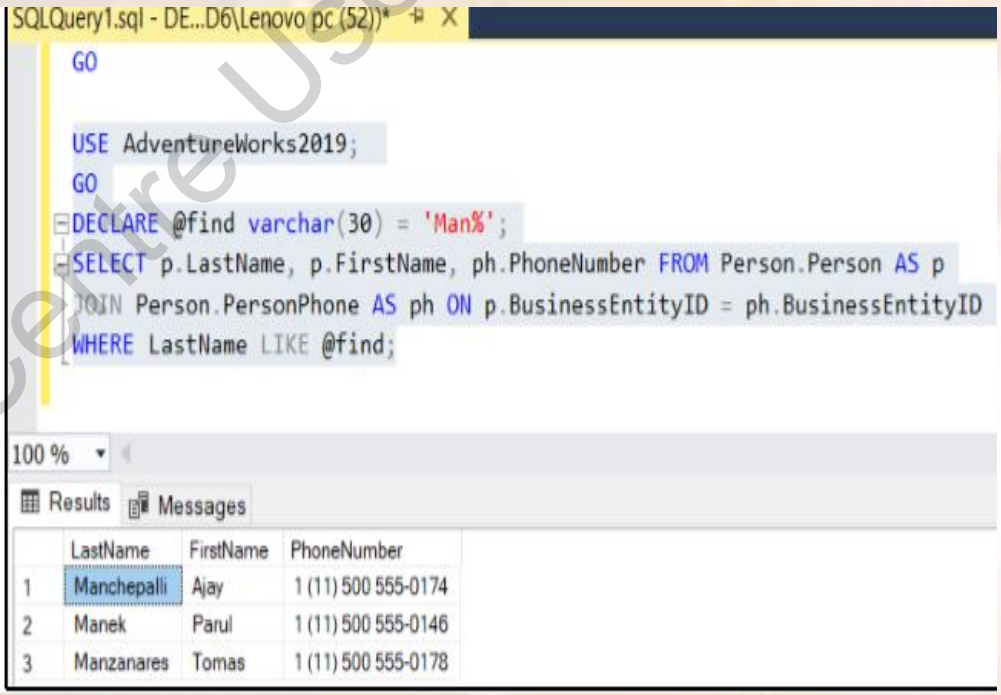
For example,

- Users can create a query that requires various types of data values specified in WHERE clause each time the query is executed.
- Here, the users can write logic to store variables with appropriate data.

Transact-SQL Variables 2-3

DECLARE

- Variables are declared with the DECLARE statement in the body of a batch.
- These variables are assigned values by using the SELECT or SET statement.
- The variables are initialized with NULL values if the user has not provided a value at the time of the declaration.



Contact Information

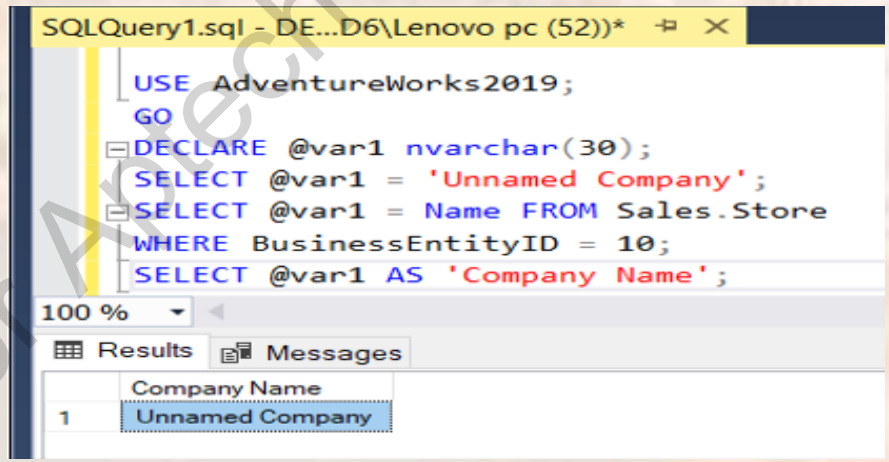
Transact-SQL Variables 3-3

SET

- The SET statement sets the local variable created by the DECLARE statement to the specified value.

SELECT

- The SELECT statement indicates that the specified local variable that was created using DECLARE should be set to the given expression.



Generic Name

Synonyms 1-5

Synonyms are database objects that serve following purposes:

They offer another name for a different database object, also called as the base object.

They present a layer of abstraction that guards a client application from the modifications made to the location and the name of the base object.

Database Objects

Database Objects
Extended stored procedure
SQL table-valued function
SQL stored procedure
Table (User-defined)
Replication-filter-procedure
SQL scalar function
SQL inline-tabled-valued function
View

Synonyms 2-5

- Suppose users want to create a synonym and have a default schema that is not owned by them.
- In such a case, they can qualify the synonym name with the schema name that they actually own.

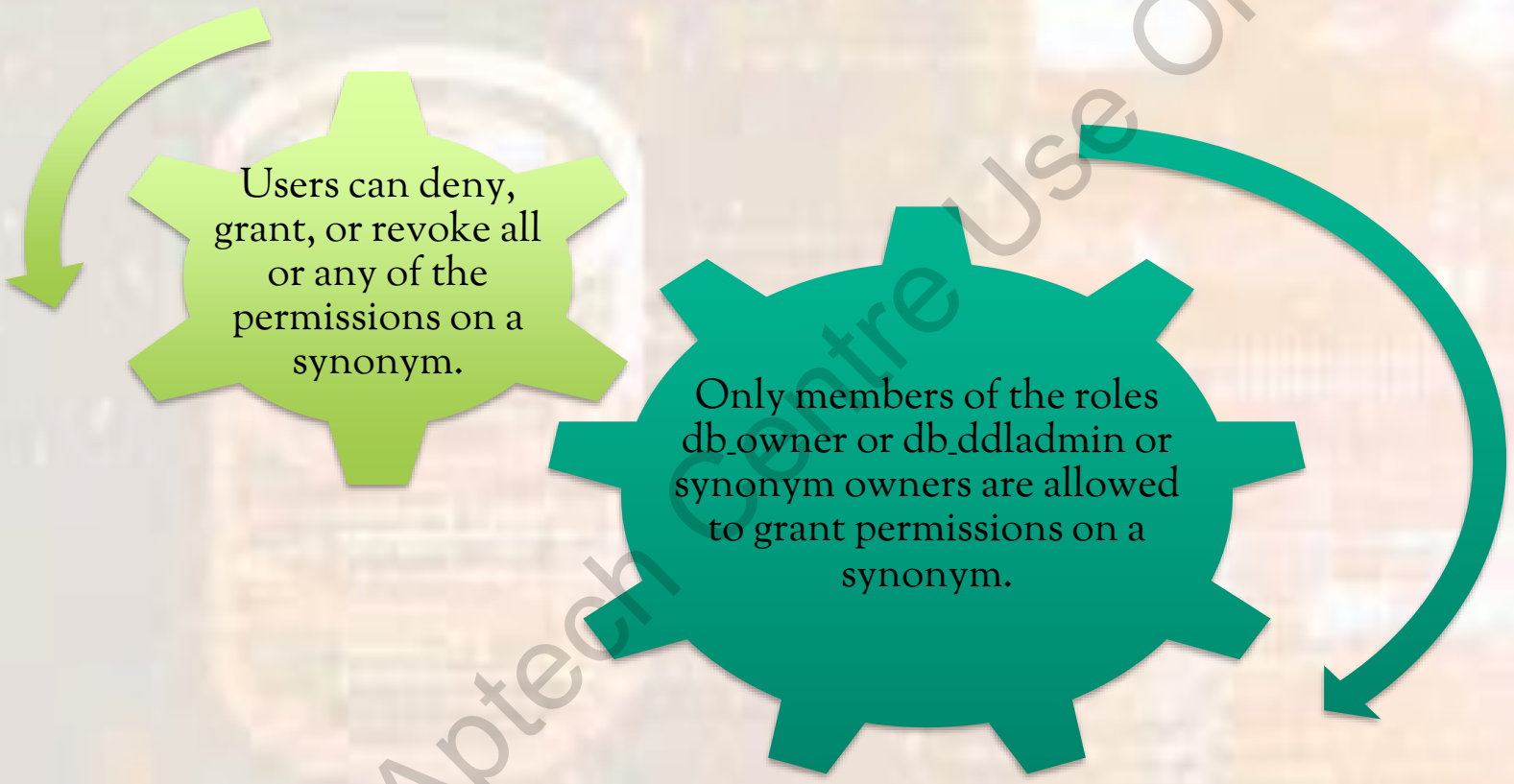
Synonyms and Schemas

For example,

- A user owns a schema Resources, but Materials is the user's default schema.
- If this user wants to create a synonym, he/she must prefix the name of the synonym with the schema Resources.

Synonyms 3-5

Granting Permissions on Synonyms



Permissions

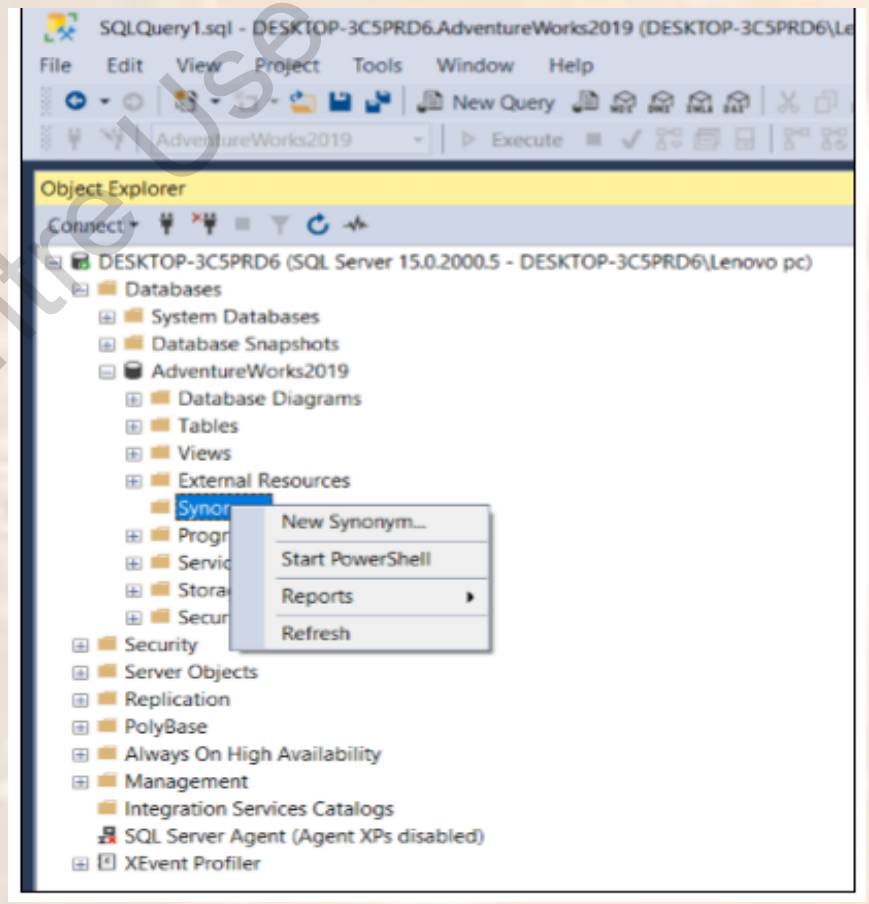
Permissions
DELETE
INSERT
TAKE OWNERSHIP
VIEW DEFINITION
CONTROL
EXECUTE
SELECT
UPDATE

Synonyms 4-5

Working with Synonyms

To create a synonym using SSMS, perform the following steps:

- In Object Explorer, expand the database to create a new synonym.
- Select the Synonyms folder, right-click it and then, click New Synonym.



Creating a New Synonym

Synonyms 5-5

Synonym name	New name for the object. Here, Emp is the name.
Synonym schema	Is the new name for the schema object. Here, HumanResources is used for the synonym and the object type.
Server name	Name of the server to be connected. Here, the server name is specified as 10.2.110.140.
Database name	Connects the object. Here, AdventureWorks2019 is the database name.

Program Flow Statements 1-2

Different types of program flow statements and functions supported by Transact-SQL are as follows:

- **Transact-SQL Control-of-Flow language**

  - Control-of-flow language determines the execution flow of Transact-SQL statements, statement blocks, user-defined functions, and stored procedures.

Control-Of-Flow Language Keywords
RETURN
THROW
TRY....CATCH
WAITFOR
WHILE
BEGIN....END
BREAK
CONTINUE
GOTO label
IF...ELSE

Keywords

Program Flow Statements 2-2

➢ BEGIN....END

- The BEGIN...END statements with a series of Transact-SQL statements so that a group of Transact-SQL statements is executed.

➢ IF...ELSE

- The BEGIN...END statements with a series of Transact-SQL statements so that a group of Transact-SQL statements is executed.

➢ WHILE

- The WHILE statement specifies a condition for the repetitive execution of the statement block.

Transact-SQL Functions 1-3

Deterministic Built-in Functions	Non-Deterministic Built-in Functions
POWER	@@TOTAL_WRITE
ROUND	CURRENT_TIMESTAMP
RADIANS	GETDATE
EXP	GETUTCDATE
FLOOR	GET_TRANSMISSION_STATUS
SQUARE	NEWID
SQRT	NEWSEQUENTIALID
LOG	@@CONNECTIONS
YEAR	@@CPU_BUSY
ABS	@@DBTS
ASIN	@@IDLE
ACOS	@@IOBUSY
SIGN	@@PACK_RECEIVED
SIN	@@PACK_SENT

Deterministic and Non-deterministic Built-in Functions

Transact-SQL Functions 2-3

Some functions are not always deterministic, but you can use them in indexed views if they are given in a deterministic manner.

Function	Description
CONVERT	Is deterministic only if one of these conditions exists: • Has an sql_variant source type. • Has an sql_variant target type and source type is non-deterministic. • Has its source or target type as smalldatetime or datetime, has the other source or target type as a character string, and has a non-deterministic style specified. The style parameter must be a constant to be deterministic.
CAST	Is deterministic only if it is used with smalldatetime, sql_variant, or datetime.
ISDATE	Is deterministic unless used with the CONVERT function, the CONVERT style parameter is specified, and style is not equal to 0, 100, 9, or 109.
CHECKSUM	Is deterministic, with the exception of CHECKSUM(*).

Deterministic Functions

Transact-SQL Functions 3-3

Calling Extended Stored Procedures from Functions

- They are non-deterministic because the extended stored procedures may result in side effects on the database.

Scalar-valued Functions

- A Scalar-valued Function (SVF) always returns an int, bit, or string value.

Table-valued Functions

- Table-valued functions are user-defined functions that return a table.

Altering User-defined Functions 1-2

- Users can modify the user-defined functions in SQL Server 2019 by using the Transact-SQL or SSMS.

- Changing the user-defined functions does not modify the functions' permissions, nor will it affect any stored procedures, triggers, or functions.

Limitations and Restrictions

The ALTER FUNCTION does not allow users to perform the following actions:

- Modify a scalar-valued function to a table-valued function.
- Modify an inline function to a multi-statement function.
- Modify a Transact-SQL to a CLR function.

Permissions

The ALTER permission is required on the schema or the function. If the function specifies a user-defined type, then it requires the EXECUTE permission on the type.

Altering User-defined Functions 2-2

Modifying a User-defined function using SSMS

1. Click the plus (+) symbol beside the database that contains the function to be modified.
2. Click the plus (+) symbol next to the Programmability folder.
3. Click the plus (+) symbol next to the folder, which contains the function to be modified. There are four folder types as follows:
   - Table-valued Functions
   - Scalar-valued Functions
   - Aggregate Functions
   - System Functions
4. Right-click the function to be modified and then, select **Modify**. The code for the function appears in a query editor window.
5. In the query editor window, make the required changes to the `ALTER FUNCTION` statement body.
6. Click **Execute** on the toolbar to execute the `ALTER FUNCTION` statement.

Modifying a User-defined function using Transact-SQL

- In the **Object Explorer**, connect to the Database Engine instance.
- On the **Standard bar**, click **New Query**.
- Type the `ALTER FUNCTION` code in the **Query Editor**.
- Click **Execute** on the toolbar to execute the `ALTER FUNCTION` statement.

Creation of Windows with OVER

- In Transact-SQL, the OVER clause is used to define a window within a query resultset.

- Using windows and OVER clause with functions provides several advantages.

For instance,

They help to calculate aggregated values. They also enable row numbers in a resultset to be generated easily.

Windowing Components

- A window function is a function that applies to a collection of rows.

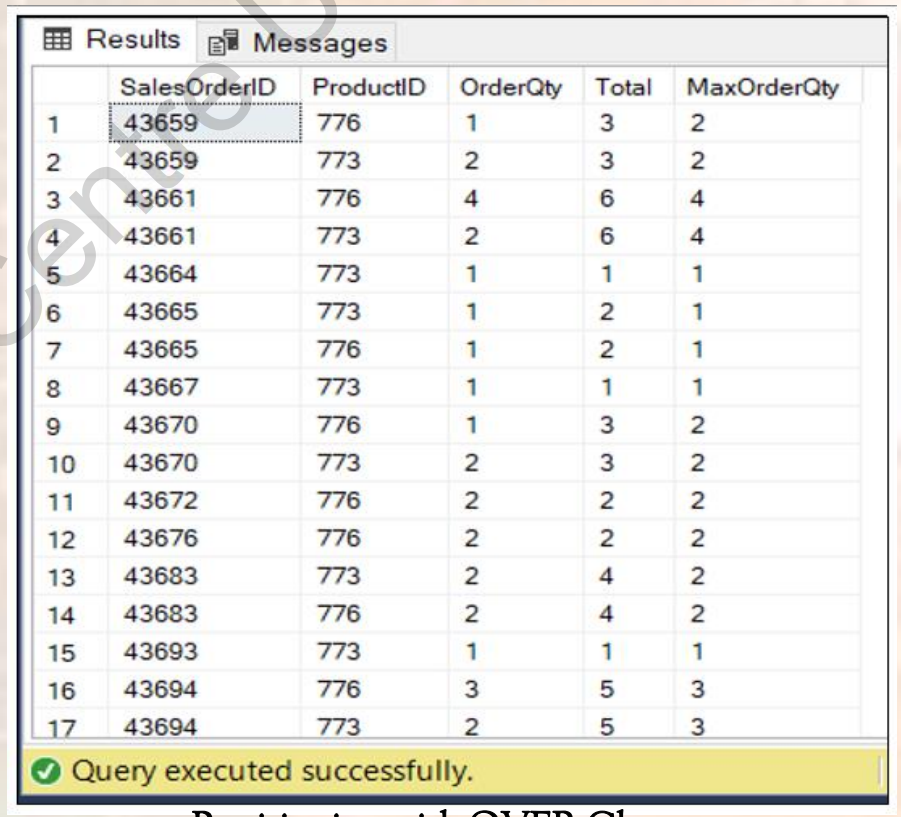
- In Transact-SQL, the OVER clause is used to define a window within a query resultset.
- Using windows and the OVER clause with functions provides several advantages.

Windowing Components 1-3

Three core components of creating windows with the OVER clause are as follows:

Partitioning

- Partitioning is a feature that limits the window of the recent calculation to only those rows from the resultset that contains the same values in the partition columns as in the existing row.
- It uses the PARTITION BY clause.



Results | Messages

	SalesOrderID	ProductID	OrderQty	Total	MaxOrderQty
1	43659	776	1	3	2
2	43659	773	2	3	2
3	43661	776	4	6	4
4	43661	773	2	6	4
5	43664	773	1	1	1
6	43665	773	1	2	1
7	43665	776	1	2	1
8	43667	773	1	1	1
9	43670	776	1	3	2
10	43670	773	2	3	2
11	43672	776	2	2	2
12	43676	776	2	2	2
13	43683	773	2	4	2
14	43683	776	2	4	2
15	43693	773	1	1	1
16	43694	776	3	5	3
17	43694	773	2	5	3

Query executed successfully.

Partitioning with OVER Clause

Windowing Components 2-3

Ordering

- The ordering element defines the ordering for calculation in the partition.
- The ordering element has different meaning to some extent for different function categories. With ranking functions, ordering is spontaneous.

	CustomerID	StoreID	Rnk_All	Rnk_Cust
1	701	844	813	1
2	700	1030	633	2
3	699	842	815	3
4	698	640	1009	4
5	697	1032	631	5
6	696	840	817	6
7	695	638	1011	7
8	694	1034	629	8
9	693	838	819	9
10	692	802	855	10
11	691	1036	627	11
12	690	836	821	12
13	689	1402	278	13
14	688	1038	625	14
15	687	834	823	15
16	686	1400	279	16
17	685	1040	623	17

Query executed successfully.

Ordering

	TerritoryID	Name	SalesYTD	Rnk_One	Rnk_Two
1	1	Northwest	7887186.7882	2	1
2	2	Northeast	2402176.8476	10	1
3	3	Central	3072175.118	8	1
4	4	Southwest	10510853.8739	1	1
5	5	Southeast	2538667.2515	9	1
6	6	Canada	6771829.1376	3	1
7	7	France	4772398.3078	6	1
8	8	Germany	3805202.3478	7	1
9	9	Australia	5977814.9154	4	1
10	10	United Kingdom	5012905.3656	5	1

Partitioning and Ranking

Windowing Components 3-3

Framing

- Framing is a feature that enables you to specify a further division of rows within a window partition.
- This is done by assigning upper and lower boundaries for the window frame that presents rows to the window function.

	ProductID	Shelf	Quantity	RunQty
1	1	A	408	408
2	1	B	324	732
3	1	A	353	1085
4	2	A	427	427
5	2	B	318	745
6	2	A	364	1109
7	3	A	585	585
8	3	B	443	1028
9	3	A	324	1352
10	4	A	512	512
11	4	B	422	934
12	4	A	388	1322
13	316	A	532	532
14	316	B	388	920
15	316	B	441	1361
16	317	C	283	283
17	317	A	158	441

Query executed successfully.

Framing

Window Functions 1-5

Some of the different types of window functions are as follows:

Ranking functions

- These functions return a rank value for each row in a partition.
- Based on the function that is used, many rows will return the same value as the other rows.
- Ranking functions are non-deterministic.

Ranking Functions	Description
NTILE	Spreads rows in an ordered partition into a given number of groups, beginning at 1. For each row, the function returns the number of the group to which the row belongs.
ROW NUMBER	Retrieves the sequential number of a row in a partition of a resultset, starting at 1 for the first row in each partition.
DENSE RANK	Returns the rank of rows within the partition of a resultset, without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the row in question.

Ranking Functions

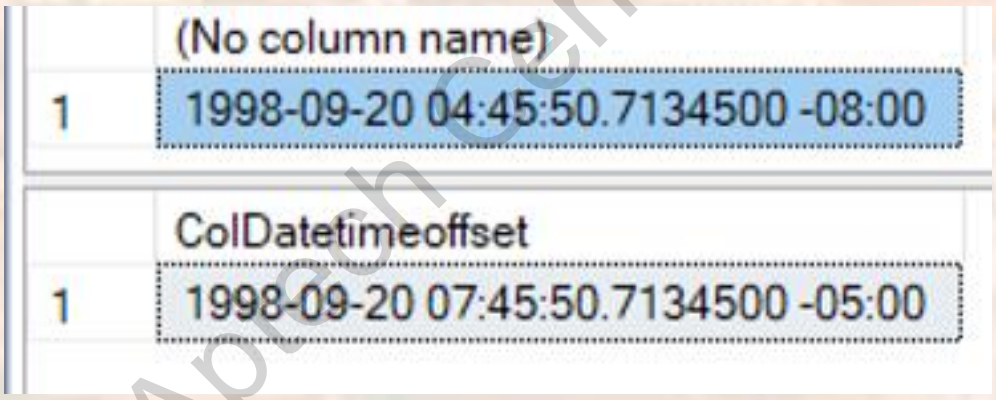
Window Functions 2-5

OFFSET functions

Different types of offset functions are as follows:

- **SWITCHOFFSET**

This function returns a DATETIMEOFFSET value that is modified from the stored time zone offset to a specific new time zone offset.



	(No column name)
1	1998-09-20 04:45:50.7134500 -08:00

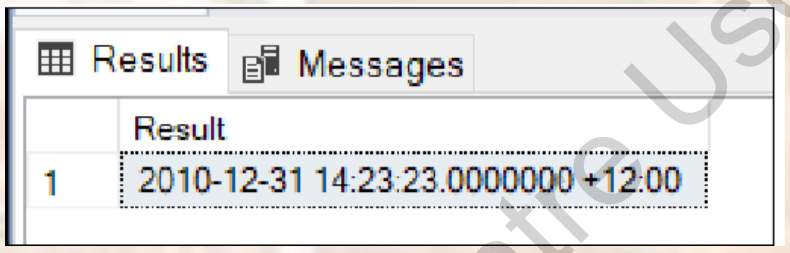
	ColDatetimeoffset
1	1998-09-20 07:45:50.7134500 -05:00

Use of SWITCHOFFSET Function

Window Functions 3-5

➢ DATETIMEOFFSETFROMPARTS

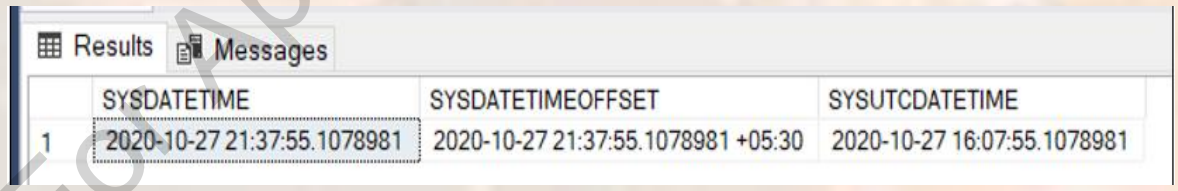
This function returns a datetimeoffset value for the specified date and time with specified precision and offset.



Use of DATETIMEOFFSETFROMPARTS Function

➢ SYSDATETIMEOFFSET

These functions returns datetimeoffset(7) value which contains the date and time of the computer on which the instance of SQL Server is running.



Use of SYSDATETIMEOFFSET Function

Window Functions 4-5

Analytic Functions

SQL Server 2019 supports several analytic functions. These functions compute aggregate value based on a group of rows. Analytic functions compute running totals, moving averages, or top-N results within a group.

Function	Description
LEAD	Provides access to data from a subsequent row in the same resultset without using a self-join.
LAST_VALUE	Retrieves the last value in an ordered set of values.
LAG	Provides access to data from a previous row in the same resultset without using a self-join.
FIRST_VALUE	Retrieves the first value in an ordered set of values.
CUME_DIST	Computes the cumulative distribution of a value in a group of values.
PERCENTILE_ CONT	Computes a percentile based on a continuous distribution of the column value in SQL.
PERCENTILE_DISC	Calculates a particular percentile for sorted values in an entire rowset or within distinct partitions of a rowset.

Analytic Functions

Window Functions 5-5

	Name	ListPrice	LessExpensive
1	Patch Kit/8 Patches	2.29	Patch Kit/8 Patches
2	Road Tire Tube	3.99	Patch Kit/8 Patches
3	Touring Tire Tube	4.99	Patch Kit/8 Patches
4	Mountain Tire Tube	4.99	Patch Kit/8 Patches
5	LL Road Tire	21.49	Patch Kit/8 Patches
6	ML Road Tire	24.99	Patch Kit/8 Patches
7	LL Mountain Tire	24.99	Patch Kit/8 Patches
8	Touring Tire	28.99	Patch Kit/8 Patches
9	ML Mountain Tire	29.99	Patch Kit/8 Patches
10	HL Road Tire	32.60	Patch Kit/8 Patches
11	HL Mountain Tire	35.00	Patch Kit/8 Patches

First Value() Function

Summary

- Transact-SQL provides basic programming elements such as variables, control-of-flow elements, conditional, and loop constructs.
- A batch is a collection of one or more Transact-SQL statements that are sent as one unit from an application to the server.
- Variables allow users to store data for using as input in other Transact-SQL statements.
- Synonyms provide a way to have an alias for a database object that may exist on a remote or local server.
- Deterministic functions always return same result each time when they are called with a definite set of input values.
- Non-deterministic functions return different results every time they are called with specified set of input values even though the database that is accessed remains the same.
- A window function is a function that applies to a collection of rows.