Intelligent Data Management with SQL Server

Session: 4

Transact-SQL

Objectives

- Explain Transact-SQL
- List different categories of Transact-SQL statements
- Explain various data types supported by Transact-SQL
- Explain Transact-SQL language elements
- Explain sets and predicate logic
- Describe logical order of operators in the SELECT statement

Introduction

- SQL is the universal language used in the database world.
  - Most modern RDBMS products use some type of SQL dialect as their primary query language.
  - SQL can be used to create or destroy objects such as tables on the database server and to manipulate those objects, such as adding data into them or retrieving data from them.
- Transact-SQL is Microsoft's implementation of the standard SQL.
- Usually referred to as T-SQL, this language implements a standardized way to communicate to the database.
- It provides a comprehensive language that supports defining tables, inserting, deleting, updating, and accessing the data in the table.

Transact-SQL 1-2

Transact-SQL is a powerful language offering features such as:

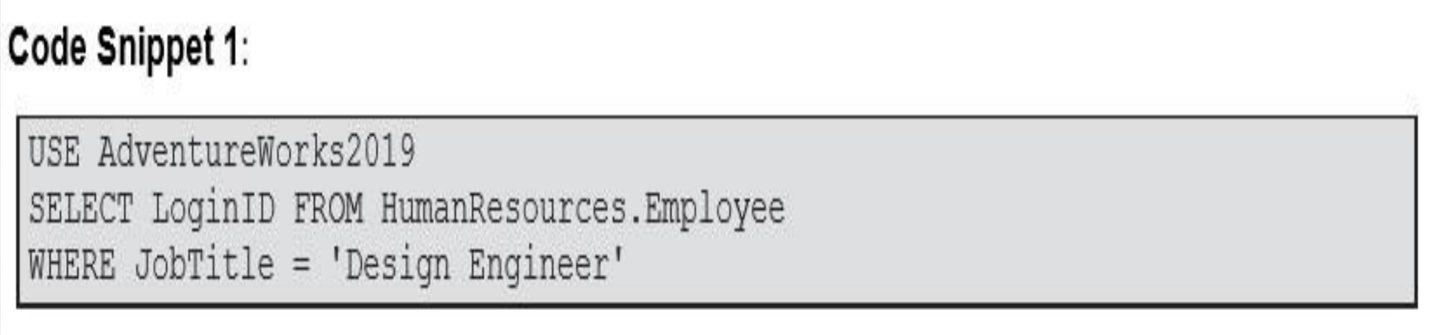
- data types
- temporary objects
- extended stored procedures
- Scrollable cursors
- conditional processing
- transaction control
- exception and error-handling

The Transact-SQL language also provides:

- improved performance
- increased functionality
- enhanced features.

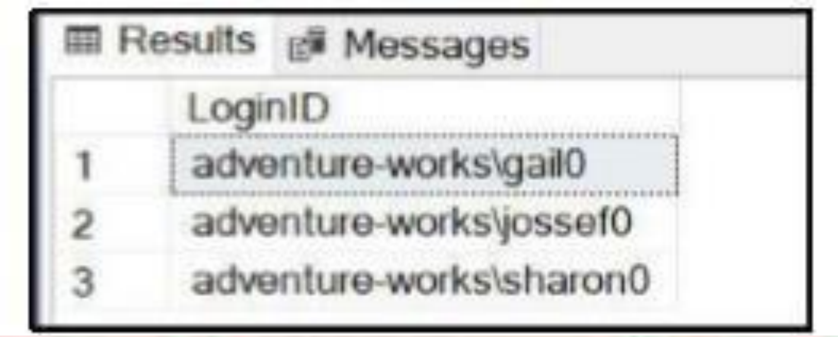
Enhancements include scalar functions, paging, sequences, meta-data discovery, and better error handling support.

Transact-SQL 2-2



Code Snippet 1:

```
USE AdventureWorks2019
SELECT LoginID FROM HumanResources.Employee
WHERE JobTitle = 'Design Engineer'
```



```
     LoginID
1    adventure-works\gail0
2    adventure-works\jossef0
3    adventure-works\sharon0
```

- ➢ Transact-SQL includes many syntax elements that are used by or that influence most statements.
- ➢ These elements include data types, predicates, functions, variables, expressions, control-of-flow, comments, and batch separators.

Different Categories of Transact-SQL Statements

SQL Server supports three types of Transact-SQL statements, namely

- DDL (Data Definition Language)
- DML (Data Manipulation Language)
- DCL (Data Control Language)

Data Definition Language (DDL)

DDL is used to define and manage all attributes and properties of a database which includes:

- row layouts
- column definitions
- key columns
- file locations
- storage strategy

Most DDL statements take the following form, where object_name can be a table, view, trigger, stored procedure, and so on:

➢ CREATE object_name
➢ ALTER object_name
➢ DROP object_name

Data Manipulation Language(DML)

DML is used to select, insert, update, or delete data in the objects defined with DDL

All database users can use these commands during the routine operations on adatabase

Different DML statements are as follows:

- ➢ SELECT statement
- ➢ INSERT statement
- ➢ UPDATE statement
- ➢ DELETE statement

Data Control Language (DCL)

Data control language is used to control permissions on database objects

DCL statements are also used for securing the database

The three basic DCL statements are as follows:

- ➢ GRANT statement
- ➢ REVOKE statement
- ➢ DENY statement

Data Types 1-3

- A data type is an attribute defining the type of data that an object can contain.
- Data types must be provided for columns, parameters, variables, and functions that return data values, and stored procedures that have a return code.

Following objects have datatypes:

➢ Columns present in tables and views

➢ Parameters in stored procedures

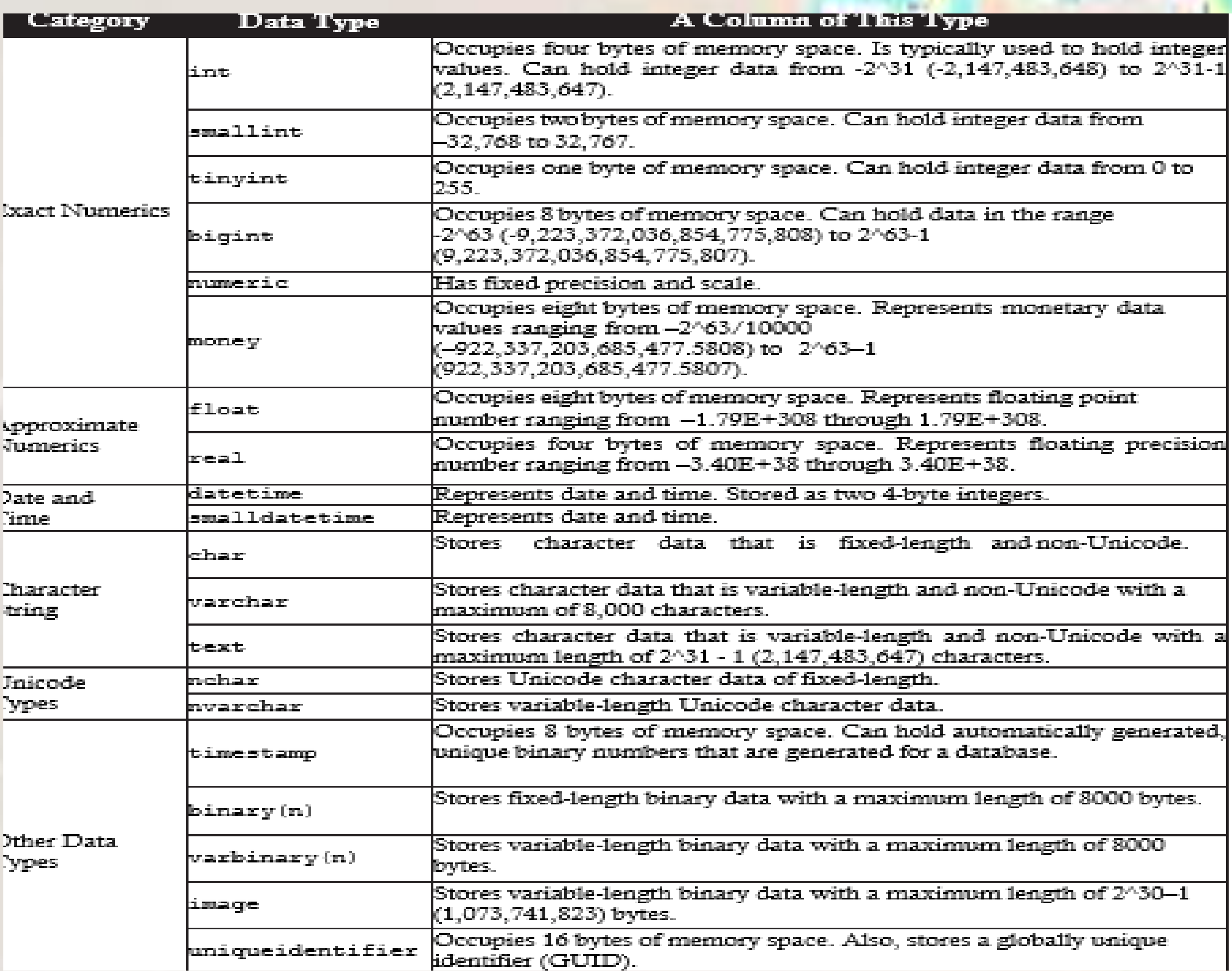
➢ Variables

➢Transact-SQL functions that return one or more data values of a specific data type

➢ Stored procedures that have a return code belonging to the integer data type

Data Types 2-3

System-defined datatypes

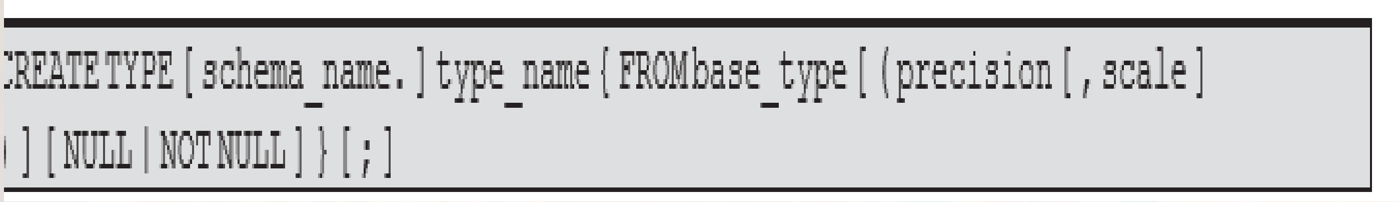


Category	Data Type	A Column of This Type
Exact Numerics	int	Occupies four bytes of memory space. Is typically used to hold integer values. Can hold integer data from -2^31 (-2,147,483,648) to 2^31-1 (2,147,483,647).
	smallint	Occupies two bytes of memory space. Can hold integer data from –32,768 to 32,767.
	tinyint	Occupies one byte of memory space. Can hold integer data from 0 to 255.
	bigint	Occupies 8 bytes of memory space. Can hold data in the range -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807).
	numeric	Has fixed precision and scale.
	money	Occupies eight bytes of memory space. Represents monetary data values ranging from –2^63/10000 (–922,337,203,685,477.5808) to 2^63–1 (922,337,203,685,477.5807).
Approximate Numerics	float	Occupies eight bytes of memory space. Represents floating point number ranging from –1.79E+308 through 1.79E+308.
	real	Occupies four bytes of memory space. Represents floating precision number ranging from –3.40E+38 through 3.40E+38.
Date and Time	datetime	Represents date and time. Stored as two 4-byte integers.
	smalldatetime	Represents date and time.
Character String	char	Stores character data that is fixed-length and non-Unicode.
	varchar	Stores character data that is variable-length and non-Unicode with a maximum of 8,000 characters.
	text	Stores character data that is variable-length and non-Unicode with a maximum length of 2^31 - 1 (2,147,483,647) characters.
Unicode Types	nchar	Stores Unicode character data of fixed-length.
	nvarchar	Stores variable-length Unicode character data.
Other Data Types	timestamp	Occupies 8 bytes of memory space. Can hold automatically generated, unique binary numbers that are generated for a database.
	binary(n)	Stores fixed-length binary data with a maximum length of 8000 bytes.
	varbinary(n)	Stores variable-length binary data with a maximum length of 8000 bytes.
	image	Stores variable-length binary data with a maximum length of 2^30–1 (1,073,741,823) bytes.
	uniqueidentifier	Occupies 16 bytes of memory space. Also, stores a globally unique identifier (GUID).

Data Types 3-3

Alias data types

- These are based on the system-supplied data types.
- Alias data types are used when more than one table stores the same type of data in a column and has similar characteristics such as length, nullability.



```
CREATE TYPE [ schema_name. ] type_name { FROM base_type [ ( precision [ , scale ]
) ] [ NULL | NOT NULL ] } [ ; ]
```

User-defined types

These are created using programming languages supported by the .NET Framework.

Transact-SQL Language Elements

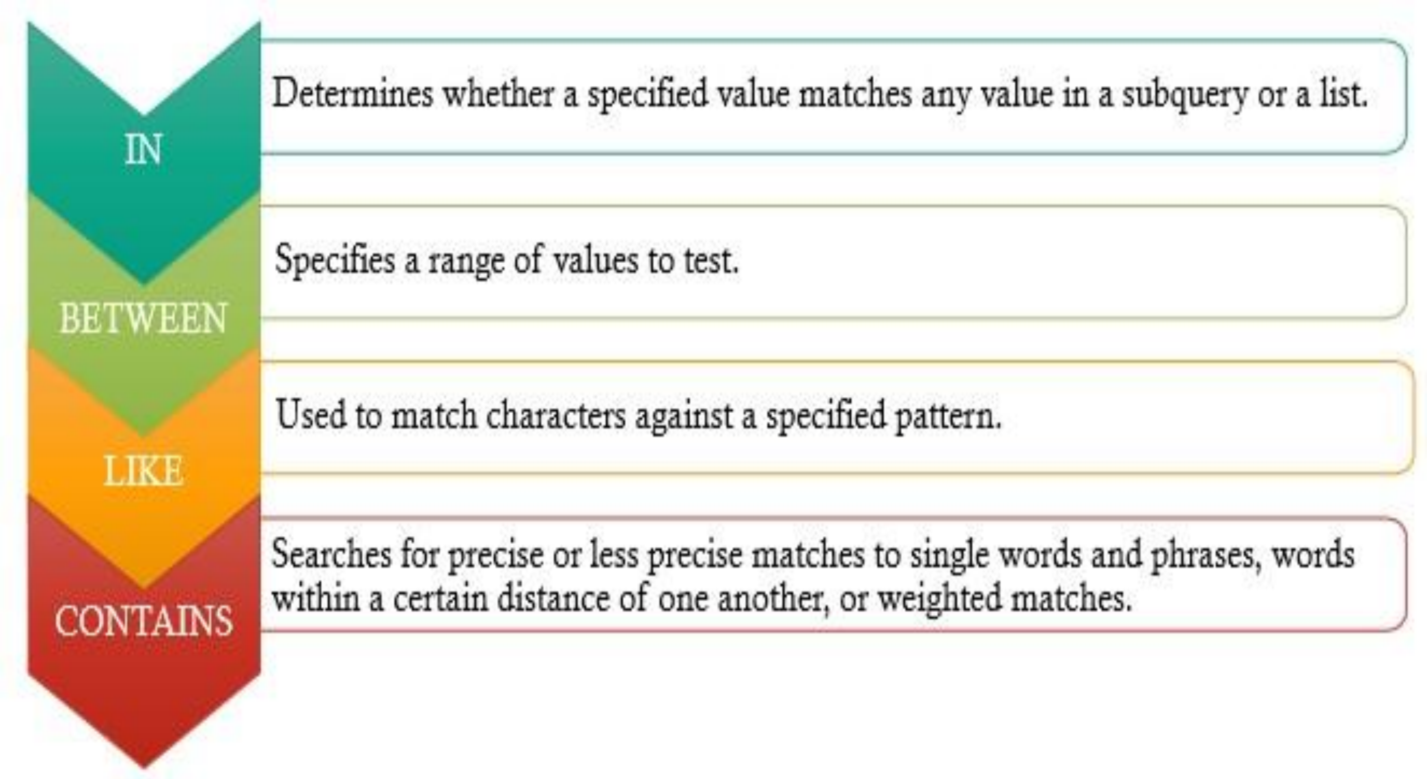
- The Transact-SQL language elements are used in SQL Server 2019 for working on the data that is entered in SQL Server database.

The Transact-SQL language elements includes:

- Predicates
- Operators
- Functions
- Variables
- Expressions
- control-of-flow
- Errors and
- Transactions, comments, and batch separators.

Predicates and Operators 1-2

Predicates are used to evaluate whether an expression is TRUE, FALSE, or UNKNOWN.



IN

Determines whether a specified value matches any value in a subquery or a list.

BETWEEN

Specifies a range of values to test.

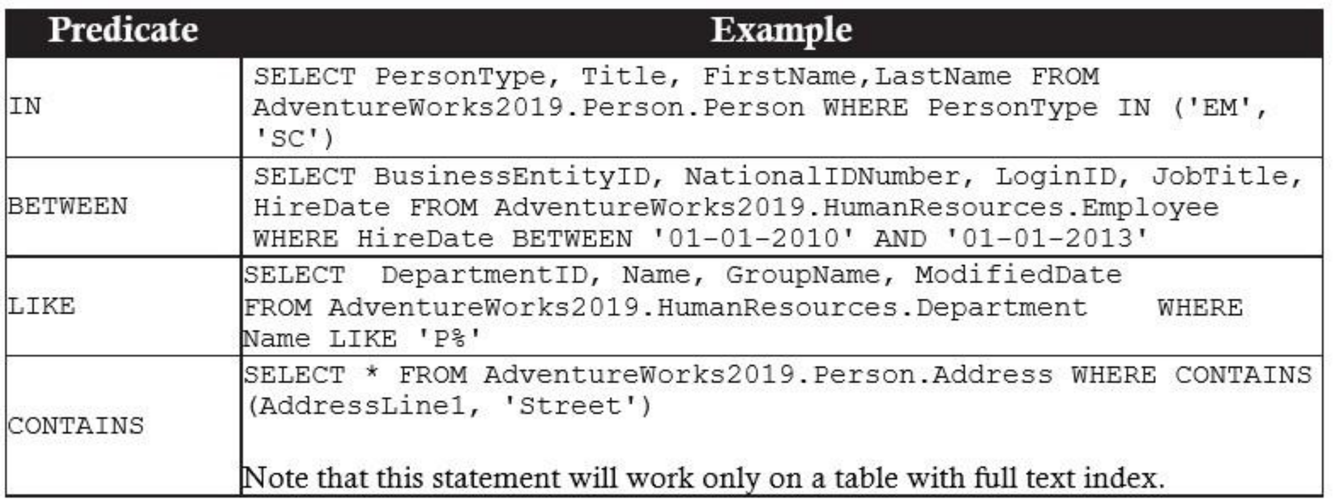
LIKE

Used to match characters against a specified pattern.

CONTAINS

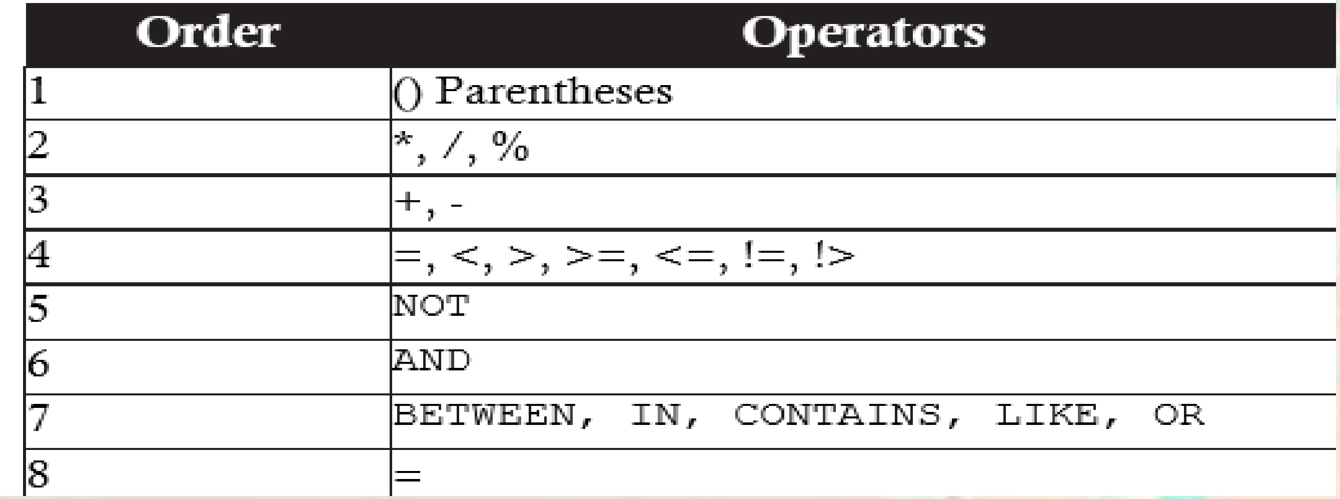
Searches for precise or less precise matches to single words and phrases, words within a certain distance of one another, or weighted matches.

Predicates and Operators 2-2



Predicate	Example
IN	SELECT PersonType, Title, FirstName,LastName FROM AdventureWorks2019.Person.Person WHERE PersonType IN ('EM', 'SC')
BETWEEN	SELECT BusinessEntityID, NationalIDNumber, LoginID, JobTitle, HireDate FROM AdventureWorks2019.HumanResources.Employee WHERE HireDate BETWEEN '01-01-2010' AND '01-01-2013'
LIKE	SELECT DepartmentID, Name, GroupName, ModifiedDate FROM AdventureWorks2019.HumanResources.Department WHERE Name LIKE 'P%'
CONTAINS	SELECT * FROM AdventureWorks2019.Person.Address WHERE CONTAINS (AddressLine1, 'Street') Note that this statement will work only on a table with full text index.

Predicate Examples



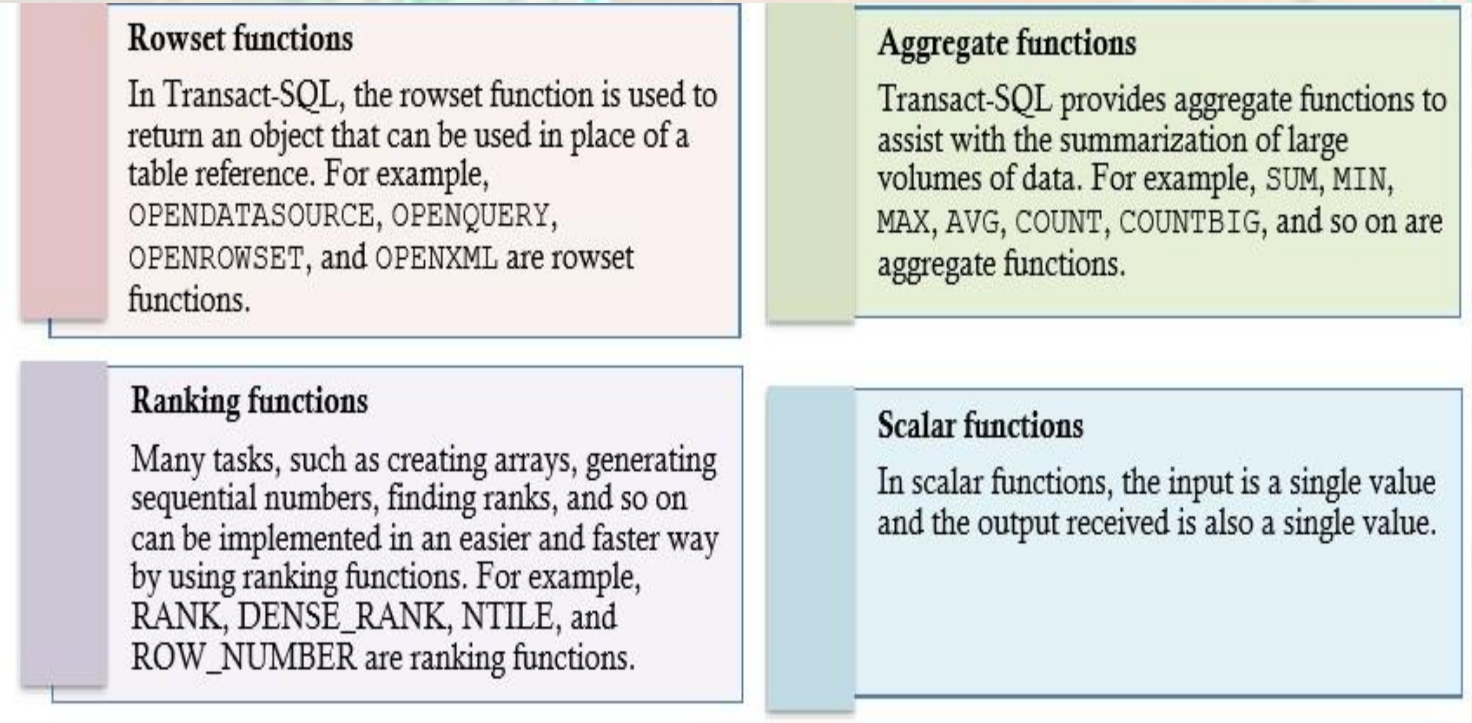
Order	Operators
1	() Parentheses
2	*, /, %
3	+, -
4	=, <, >, >=, <=, !=, !>
5	NOT
6	AND
7	BETWEEN, IN, CONTAINS, LIKE, OR
8	=

Precedence of Predicates and Operators

Functions 1-2

A function is a set of Transact-SQL statements that is used to perform some task. These functions can be useful when data is calculated or manipulated.

The four types of functions in SQL Server 2019 are as follows:



Rowset functions

In Transact-SQL, the rowset function is used to return an object that can be used in place of a table reference. For example, `OPENDATASOURCE`, `OPENQUERY`, `OPENROWSET`, and `OPENXML` are rowset functions.

Aggregate functions

Transact-SQL provides aggregate functions to assist with the summarization of large volumes of data. For example, `SUM`, `MIN`, `MAX`, `AVG`, `COUNT`, `COUNTBIG`, and so on are aggregate functions.

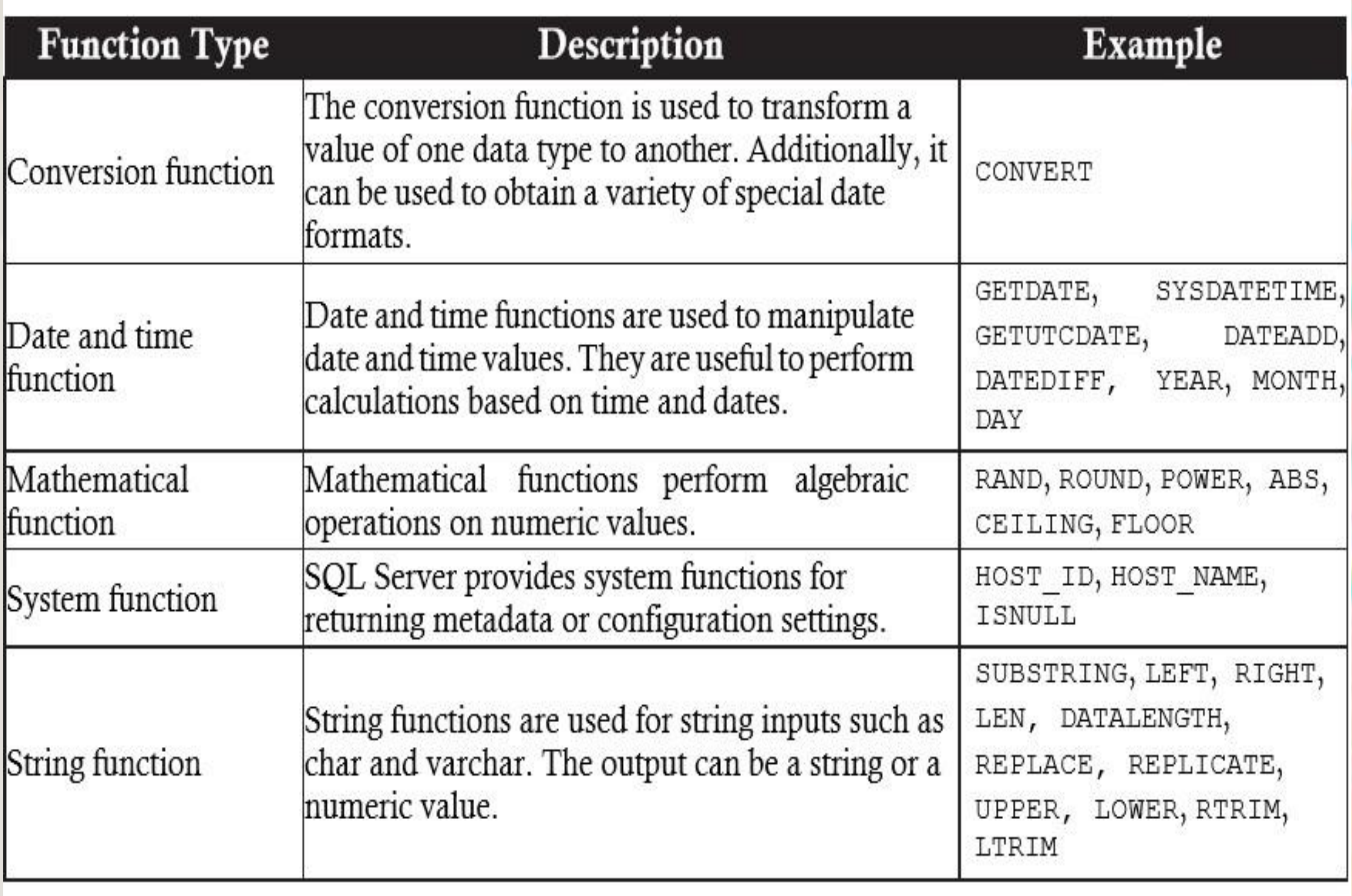
Ranking functions

Many tasks, such as creating arrays, generating sequential numbers, finding ranks, and so on can be implemented in an easier and faster way by using ranking functions. For example, RANK, DENSE_RANK, NTILE, and ROW_NUMBER are ranking functions.

Scalar functions

In scalar functions, the input is a single value and the output received is also a single value.

Functions 2-2



Function Type	Description	Example
Conversion function	The conversion function is used to transform a value of one data type to another. Additionally, it can be used to obtain a variety of special date formats.	CONVERT
Date and time function	Date and time functions are used to manipulate date and time values. They are useful to perform calculations based on time and dates.	GETDATE, SYSDATETIME, GETUTCDATE, DATEADD, DATEDIFF, YEAR, MONTH, DAY
Mathematical function	Mathematical functions perform algebraic operations on numeric values.	RAND, ROUND, POWER, ABS, CEILING, FLOOR
System function	SQL Server provides system functions for returning metadata or configuration settings.	HOST_ID, HOST_NAME, ISNULL
String function	String functions are used for string inputs such as char and varchar. The output can be a string or a numeric value.	SUBSTRING, LEFT, RIGHT, LEN, DATALENGTH, REPLACE, REPLICATE, UPPER, LOWER, RTRIM, LTRIM

Scalar Functions

Variables

A variable is an object that can hold a data value. In Transact-SQL, variables can be classified into local and global variables.

- In Transact-SQL, local variables are created and used for temporary storage while SQL statements are executed.
- Data can be passed to SQL statements using local variables.
- The name of a local variable must be prefixed with '@' sign.

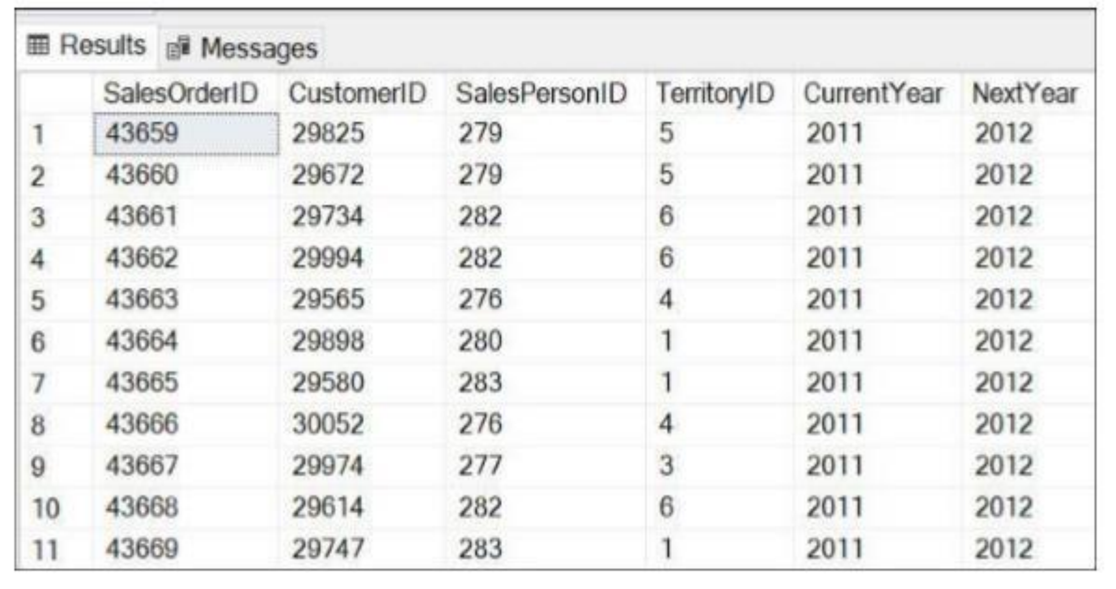
For example,

```
DECLARE @SearchWord NVARCHAR(30)
SET @SearchWord = N'performance'
```

Expressions

An expression is a combination of identifiers, values, and operators that SQL Server can evaluate in order to obtain a result.

Expressions can be used in several different places when accessing or changing data.



	SalesOrderID	CustomerID	SalesPersonID	TerritoryID	CurrentYear	NextYear
1	43659	29825	279	5	2011	2012
2	43660	29672	279	5	2011	2012
3	43661	29734	282	6	2011	2012
4	43662	29994	282	6	2011	2012
5	43663	29565	276	4	2011	2012
6	43664	29898	280	1	2011	2012
7	43665	29580	283	1	2011	2012
8	43666	30052	276	4	2011	2012
9	43667	29974	277	3	2011	2012
10	43668	29614	282	6	2011	2012
11	43669	29747	283	1	2011	2012

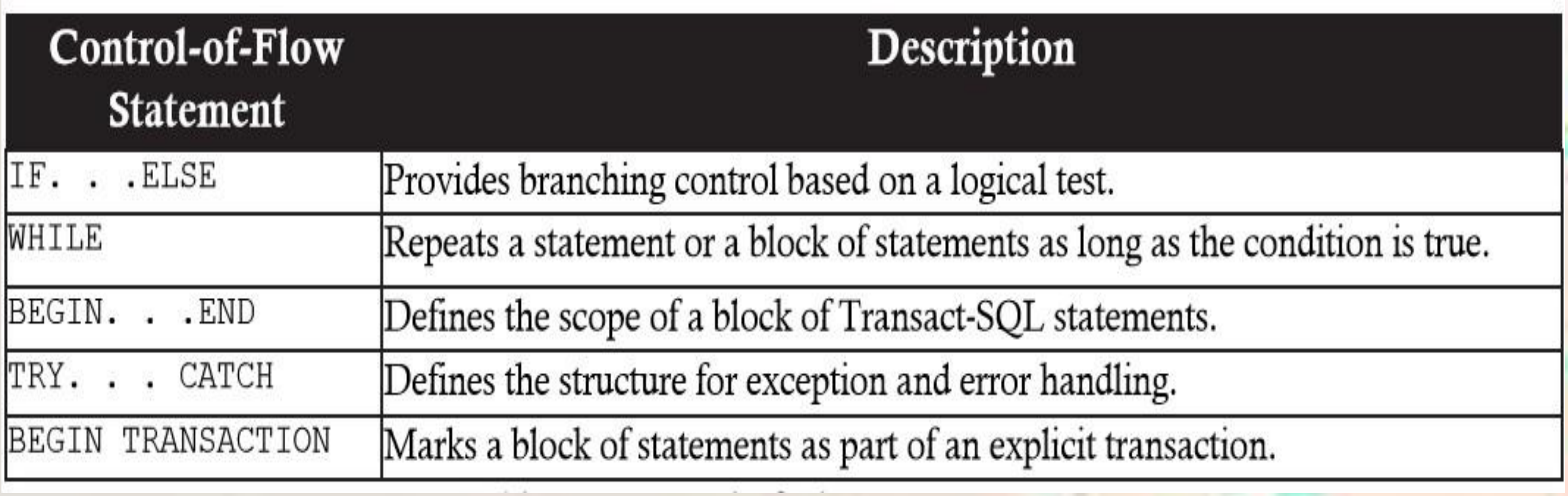
Expression Result

Control-of-Flow, Errors, and Transactions

Although Transact-SQL is primarily a data retrieval language, it supports control-of-flow statements for executing and finding errors.

Control-of-flow language determines the execution flow of

- Transact-SQL statements
- statement blocks
- user-defined functions
- and stored procedures



Control-of-Flow Statement	Description
`IF. . .ELSE`	Provides branching control based on a logical test.
`WHILE`	Repeats a statement or a block of statements as long as the condition is true.
`BEGIN. . .END`	Defines the scope of a block of Transact-SQL statements.
`TRY. . . CATCH`	Defines the structure for exception and error handling.
`BEGIN TRANSACTION`	Marks a block of statements as part of an explicit transaction.

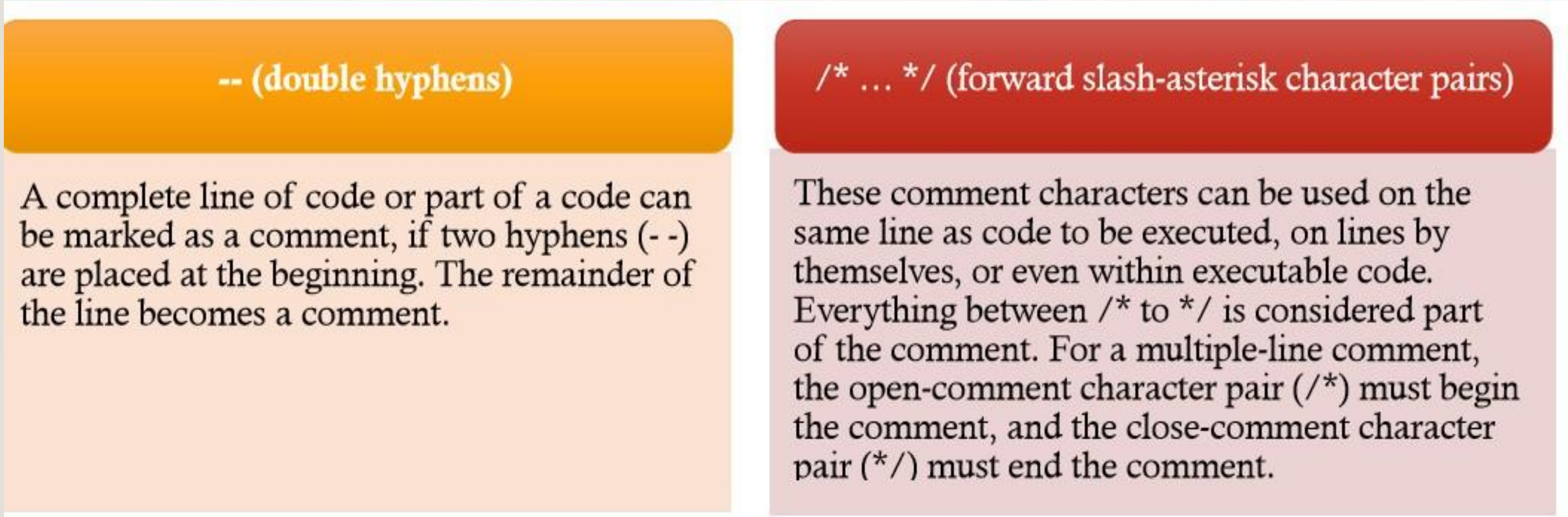
Control-of-Flow Statements

Comments

Comments are descriptive text strings, also known as remarks, in program code that will be ignored by the compiler.

Can be included inside the source code of a single statement, a batch, or a stored procedure.

SQL Server supports two types of commenting styles:



-- (double hyphens)	/* … */ (forward slash-asterisk character pairs)
A complete line of code or part of a code can be marked as a comment, if two hyphens (- -) are placed at the beginning. The remainder of the line becomes a comment.	These comment characters can be used on the same line as code to be executed, on lines by themselves, or even within executable code. Everything between /* to */ is considered part of the comment. For a multiple-line comment, the open-comment character pair (/*) must begin the comment, and the close-comment character pair (*/) must end the comment.

Batch Separators

A batch is a collection of one or more Transact-SQL statements sent at one time from an application to SQL Server for execution.

The Transact-SQL statements in a batch are compiled into a single executable unit, called an execution plan.

The process wherein a set of commands are processed one at a time from a batch of commands is called batch processing.

A batch separator is handled by SQL Server client tools such as SSMS to execute commands.

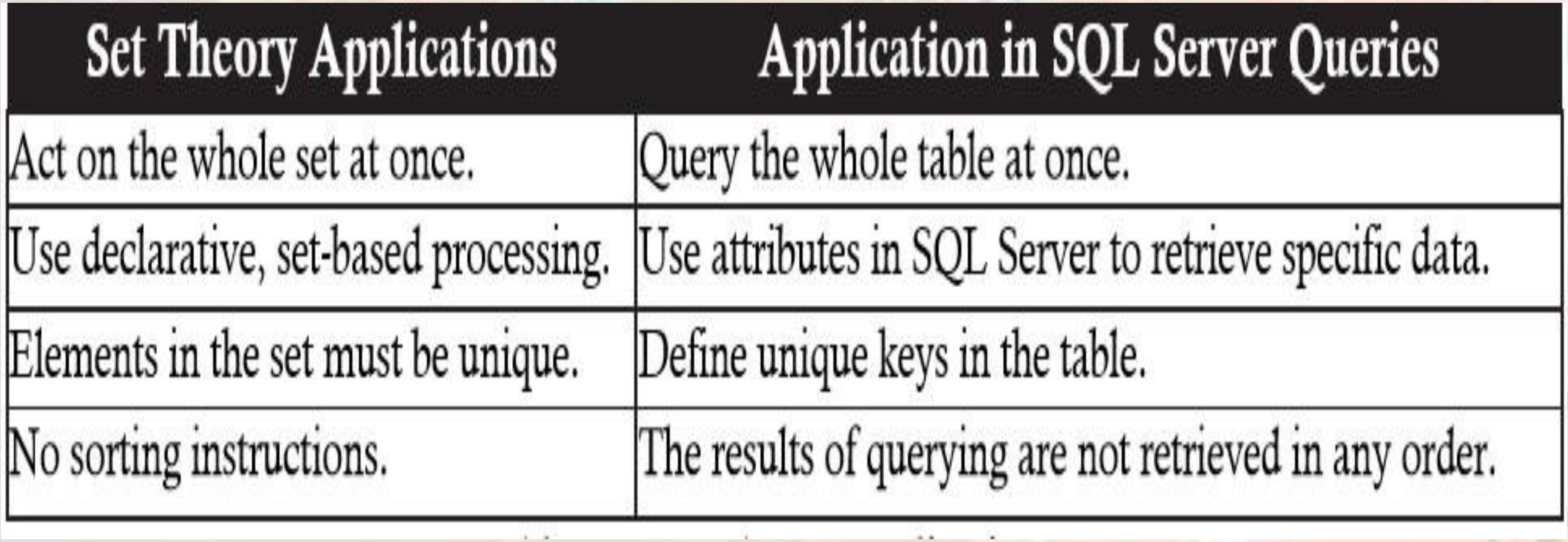
Sets and Predicate Logic

- Sets and Predicate Logic are the two mathematical fundamentals that are used in SQL Server 2019.
- Both these theories are used in querying of data in SQL Server 2019.

Set Theory

Set theory is a mathematical foundation used in relational database model. A set is a collection of distinct objects considered as a whole.

For example, all the employees under an Employee table can be considered as one set. Employees are different objects that form a part of the set in the Employee table.

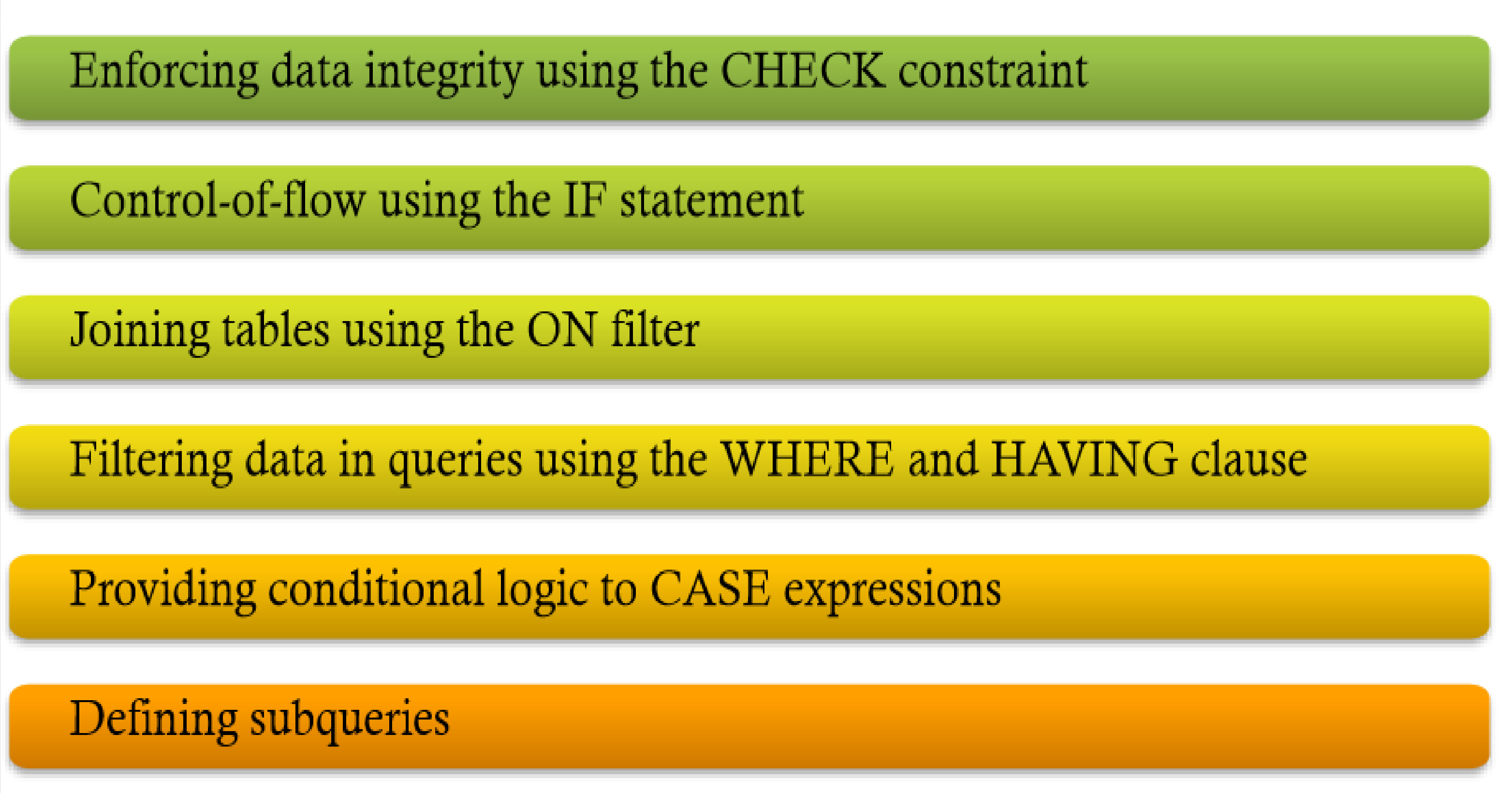


Set Theory Applications	Application in SQL Server Queries
Act on the whole set at once.	Query the whole table at once.
Use declarative, set-based processing.	Use attributes in SQL Server to retrieve specific data.
Elements in the set must be unique.	Define unique keys in the table.
No sorting instructions.	The results of querying are not retrieved in any order.

Predicate Logic

Predicate logic is a mathematical framework that consists of logical tests that gives a result. The results are always displayed as either true or false.

Some of the applications of predicate logic in Transact-SQL are as follows:

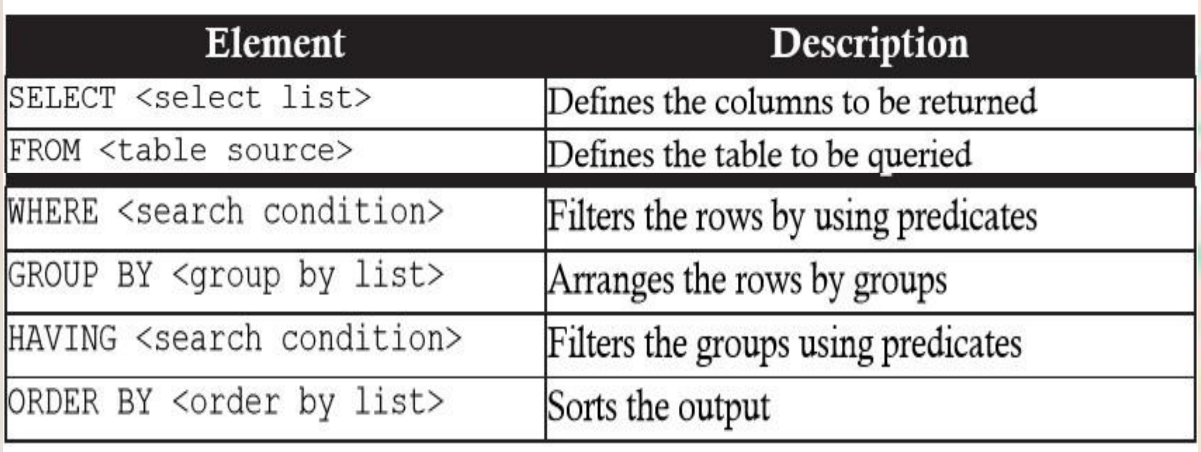


- Enforcing data integrity using the CHECK constraint
- Control-of-flow using the IF statement
- Joining tables using the ON filter
- Filtering data in queries using the WHERE and HAVING clause
- Providing conditional logic to CASE expressions
- Defining subqueries

Logical Order of Operators in SELECT Statement 1-3

Along with syntax of different SQL Server elements, an SQL Server user must also know the process of how the entire query is executed.

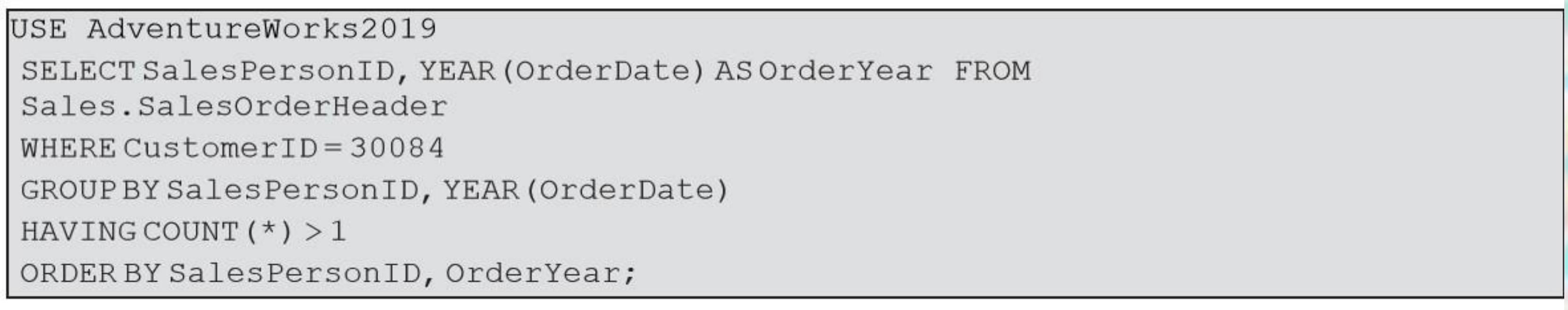
This process is a logical process that breaks the query and executes the query according to a predefined sequence in SQL Server 2019.



Element	Description
`SELECT <select list>`	Defines the columns to be returned
`FROM <table source>`	Defines the table to be queried
`WHERE <search condition>`	Filters the rows by using predicates
`GROUP BY <group by list>`	Arranges the rows by groups
`HAVING <search condition>`	Filters the groups using predicates
`ORDER BY <order by list>`	Sorts the output

Elements of SELECT Statement

Logical Order of Operators in SELECT Statement 2-3



```
USE AdventureWorks2019
 SELECT SalesPersonID, YEAR(OrderDate) AS OrderYear FROM
 Sales.SalesOrderHeader
 WHERE CustomerID = 30084
 GROUP BY SalesPersonID, YEAR(OrderDate)
 HAVING COUNT(*) > 1
 ORDER BY SalesPersonID, OrderYear;
```

In the example, the order in which SQL Server will execute the SELECT statement is asfollows:

1.First, the FROM clause is evaluated to define the source table that will be queried.

2. Next, the WHERE clause is evaluated to filter the rows in the source table. This filtering is defined by the predicate mentioned in the WHERE clause.

3.After this, the GROUP BY clause is evaluated. This clause arranges the filtered values received from the WHERE clause.

4.Next, the HAVING clause is evaluated based on the predicate that is provided.

5. Next, the SELECT clause is executed to determine the columns that will appear in the query results.

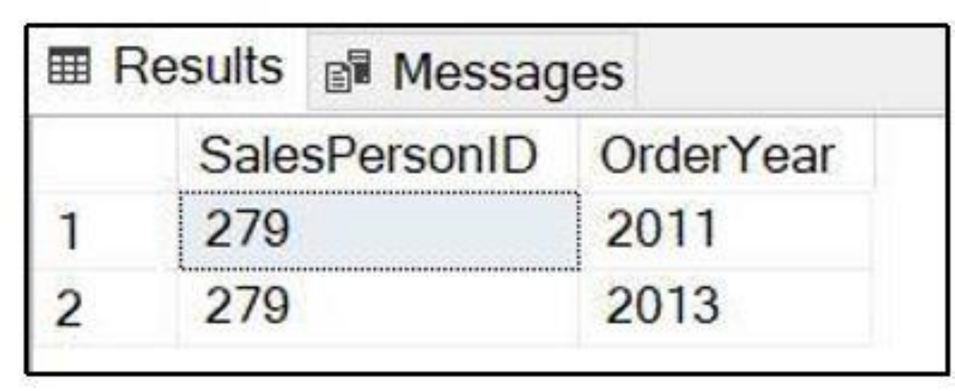
6. Finally, the ORDER BY statement is evaluated to display the output.

Logical Order of Operators in SELECT Statement 3-3

The order of execution for the SELECT statement would be as follows:

5.SELECT SalesPersonID, YEAR(OrderDate) AS OrderYear

1.FROM SalesOrderHeader
2.WHERE CustomerID = 30084
3.GROUP BY SalesPersonID, YEAR(OrderDate)
4.HAVING COUNT(*) > 1
6. ORDER BY SalesPersonID, OrderYear;



	SalesPersonID	OrderYear
1	279	2011
2	279	2013

SELECT Statement Result

Summary

- Transact-SQL is apowerful language which offers features such asdata types, temporary objects, and extended stored procedures.
- SQL Server supports three types of Transact-SQL statements, namely, DDL, DML, and DCL.
- A data type is an attribute defining the type of data that an object can contain.
- The Transact-SQL language elements includes predicates, operators, functions, variables, expressions, control-of-flow, errors, and transactions, comments, and batch separators.
- Sets and Predicate Logic are the two mathematical fundamentals that are used in SQL Server 2019.
- Set theory is a mathematical foundation used in relational database model, where a set is a collection of distinct objects considered as a whole.
- Predicate logic is a mathematical framework that consists of logical tests that gives aresult.