

# Module 2

## Introduction to T-SQL Querying



**Zyeed Ahmed.**  
**Aspiring To Learning Data Engineer.**

# Module Overview

- Introducing T-SQL
- Understanding Sets
- Understanding Predicate Logic
- Understanding the Logical Order of Operations in SELECT Statements



Zyeed Ahmed  
Aspiring To Learning Data Engineer.

# Lesson 1: Introducing T-SQL

- About T-SQL
- Categories of T-SQL Statements
- T-SQL Language Elements
- T-SQL Language Elements: Predicates and Operators
- T-SQL Language Elements: Functions
- T-SQL Language Elements: Variables
- T-SQL Language Elements: Expressions
- T-SQL Language Elements: Control of Flow, Errors, and Transactions
- T-SQL Language Elements: Comments
- T-SQL Language Elements: Batch Separators
- Demonstration: T-SQL Language Elements

# About T-SQL

- Structured Query Language (SQL)
  - Developed by IBM in 1970s
  - Adopted as a standard by ANSI and ISO standards bodies
  - Widely used in industry
    - PL/SQL, SQL Procedural Language, Transact-SQL
- Microsoft's implementation is Transact-SQL
  - Referred to as T-SQL
  - Query language for SQL Server 2012
- SQL is declarative, not procedural
  - Describe what you want, don't specify steps

# Categories of T-SQL Statements

## Data Manipulation Language (DML\*)

- Statements for querying and modifying data
- SELECT, INSERT, UPDATE, DELETE

## Data Definition Language (DDL)

- Statements for object definitions
- CREATE, ALTER, DROP

## Data Control Language (DCL)

- Statements for security permissions
- GRANT, REVOKE, DENY

\* DML with SELECT is the focus of this course



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.

# T-SQL Language Elements

- Predicates and Operators
- Functions
- Variables
- Expressions
- Batch Separators
- Control of Flow
- Comments



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.

# T-SQL Language Elements: Predicates and Operators

Elements:	Predicates and Operators:
Predicates	IN, BETWEEN, LIKE
Comparison Operators	=, >, <, >=, <=, <>, !=, !>, !<
Logical Operators	AND, OR, NOT
Arithmetic Operators	+, -, *, /, %
Concatenation	+

**T-SQL enforces operator precedence**

# T-SQL Language Elements: Functions

## String Functions

- SUBSTRING
- LEFT, RIGHT
- LEN
- DATALENGTH
- REPLACE
- REPLICATE
- UPPER, LOWER
- RTRIM, LTRIM

## Date and Time Functions

- GETDATE
- SYSDATETIME
- GETUTCDATE
- DATEADD
- DATEDIFF
- YEAR
- MONTH
- DAY

## Aggregate Functions

- SUM
- MIN
- MAX
- AVG
- COUNT



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.



# T-SQL Language Elements: Variables

- Local variables in T-SQL temporarily store a value of a specific data type
- Name begins with single @ sign
  - @@ reserved for system functions
- Assigned a data type
- Must be declared and used within the same batch
- In SQL Server 2008 and later, you can declare and initialize in the same statement

```
DECLARE @MyVar int = 30;
```

# T-SQL Language Elements: Expressions

- Combination of identifiers, values, and operators evaluated to obtain a single result
- Can be used in SELECT statements
  - SELECT clause
  - WHERE clause
- Can be single constant, single-valued function, or variable
- Can be combined if expressions have same the data type

```
SELECT YEAR(orderdate) + 1 ...
```

```
SELECT qty * unitprice ...
```

# T-SQL Language Elements: Control of Flow, Errors, and Transactions

- Allow you to control the flow of execution within code, handle errors, and maintain transactions
- Used in programmatic code objects
  - Stored procedures, triggers, statement blocks

## Control of Flow

- IF...ELSE
- WHILE
- BREAK
- CONTINUE
- BEGIN...END

## Error Handling

- TRY...CATCH

## Transaction Control

- BEGIN TRANSACTION
- COMMIT TRANSACTION
- ROLLBACK TRANSACTION

# T-SQL Language Elements: Comments

- Marks T-SQL code as a comment:
  - For a block, enclose it between `/*` and `*/` characters

```
/*  
    This is a block  
    of commented code  
*/
```

- For inline text, precede the comments with `--`

```
-- This line of text will be ignored
```

- T-SQL Editors such as SSMS will typically color-code comments, as shown above

# T-SQL Language Elements: Batch Separators

- Batches are sets of commands sent to SQL Server as a unit
- Batches determine variable scope, name resolution
- To separate statements into batches, use a separator:
  - SQL Server tools use the GO keyword
  - GO is not an SQL Server T-SQL command!
  - GO [count] times feature added in SQL Server 2005



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.

# Demonstration: T-SQL Language Elements

In this demonstration, you will see how to:

- Use T-SQL language elements



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.

# Lesson 2: Understanding Sets

- Set Theory and SQL Server
- Set Theory Applied to SQL Server Queries



Zyeed Ahmed.  
Aspiring To Learning Data Engineer.

# Set Theory and SQL Server

- Set theory is a mathematical basis for the relational database model
- What is a set?
  - A collection of distinct objects considered as a whole
  - "All customers who live in Portland"

Characteristics of Set Element	Example
Elements of a set called members	Customer as member of set Customers
Elements of a set are described by attributes	Customer first name, last name, birthdate
Elements must be distinct, or unique	Customer ID



# Set Theory Applied to SQL Server Queries

Application of Set Theory	Comments
Act on all elements of a set at once	Query the whole table at once
Use declarative, set-based processing	Tell the engine what you want to retrieve
Avoid cursor-based processing	Do not tell the engine how to retrieve it. Do not process each result individually
Elements of set must be unique	Define unique keys in a table
No defined order to result set	Items may be returned in any order. This requires explicit sort instructions if an order is desired

# Set Theory Applied to SQL Server Queries

- Employees (Set of all employees)
- Customers (Set of all customers)
- Orders (Set of all orders)
- Customers located in Portland
- Employees hired between 2002 and 2008
- Customers who placed orders in 2006

# Lesson 3: Understanding Predicate Logic

- Predicate Logic and SQL Server
- Predicate Logic Applied to SQL Server Queries

# Predicate Logic and SQL Server

- Predicate logic is a mathematical basis for the relational database model
- In theory, a predicate is a property or expression that is either true or false
- Predicate is also referred to as a Boolean expression

# Predicate Logic Applied to SQL Server Queries

- In SQL Server, a predicate is a property or expression that evaluates to true, false, or unknown (NULL)

## Uses for Predicates

- Filtering data in queries (WHERE and HAVING clauses)
- Providing conditional logic to CASE expressions
- Joining tables (ON filter)
- Defining subqueries
- Enforcing data integrity (CHECK constraints)
- Control of flow (IF statement)

# Predicate Logic Applied to SQL Server Queries

```
WHERE city = 'Portland'
```

```
WHERE orderdate < '20090101'
```

```
FROM t1 JOIN t2 on t1.c1 = t2.c2
```

# Lesson 4: Understanding the Logical Order of Operations in SELECT Statements

- Elements of a SELECT Statement
- Logical Query Processing
- Applying the Logical Order of Operations to Writing SELECT Statements
- Demonstration: Logical Query Processing

# Elements of a SELECT Statement

Element	Expression	Role
SELECT	<select list>	Defines which columns to return
FROM	<table source>	Defines table(s) to query
WHERE	<search condition>	Filters rows using a predicate
GROUP BY	<group by list>	Arranges rows by groups
HAVING	<search condition>	Filters groups using a predicate
ORDER BY	<order by list>	Sorts the output



# Logical Query Processing

- The order in which a query is written is not the order in which it is evaluated by SQL Server.

5: SELECT <select list>

1: FROM <table source>

2: WHERE <search condition>

3: GROUP BY <group by list>

4: HAVING <search condition>

6: ORDER BY <order by list>

# Applying the Logical Order of Operations to Writing SELECT Statements

USE TSQL;

```
SELECT empid, YEAR(orderdate) AS orderyear  
FROM Sales.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderdate)  
HAVING COUNT(*) > 1  
ORDER BY empid, orderyear;
```



Zyeed Ahmed .  
Aspiring To Learning Data Engineer.

# Demonstration: Logical Query Processing

In this demonstration, you will see how to:

- View query output that illustrates logical processing order



Zyeed Ahmed .  
Aspiring To Learning Data Engineer.