

Class Notes for Session 3 (SQL–TL3): Transact-SQL

Module

Data Management with SQL Server

Session Title: SQL–TL3

Source: SQL Server–The Definitive Guide, Session 4

Duration: 2 hours

Objective: By the end of this session, students will be able to:

- Understand Transact-SQL (T-SQL) and its role in SQL Server.
- Identify and categorize T-SQL statements.
- Describe T-SQL data types and language elements.
- Explain sets, predicate logic, and the logical order of operators in the SELECT statement.
- Write and execute basic T-SQL queries using the AdventureWorks 2022 database.

Sub-Session 1: Introduction to Transact-SQL

Definition

Transact-SQL (T-SQL) is Microsoft's proprietary extension to SQL, used to interact with SQL Server databases. It includes standard SQL for querying and managing data, plus additional features like procedural programming, error handling, and advanced functions.

Explanation

- **Purpose:** T-SQL enables users to create, query, modify, and manage databases in SQL Server.
- **Key Features:**
 - Supports Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and procedural constructs (e.g., loops, variables).
 - Extends SQL with features like stored procedures, triggers, and user-defined functions.

- **Usage:** Executed via SSMS, command-line tools, or application code to perform tasks like retrieving data, updating records, or defining database structures.

Example

- **T-SQL Query Example** (using AdventureWorks 2022):

```
USE AdventureWorks2022;  
SELECT FirstName, LastName  
FROM Person.Person  
WHERE Title = 'Mr.';
```

This retrieves names of individuals with the title "Mr." from the Person table.

Class Work

1. **Discussion:** Explain the difference between standard SQL and T-SQL in terms of functionality (e.g., T-SQL's procedural programming vs. SQL's declarative nature).
2. **Lab Exercise:** Using SSMS, connect to the AdventureWorks 2022 database and write a T-SQL query to retrieve the top 10 records from the `Production.Product` table. Save the query as `IntroQuery.sql`.

Sub-Session 2: Categories of Transact-SQL Statements

Definition

T-SQL statements are categorized into three main types: **Data Definition Language (DDL)**, **Data Manipulation Language (DML)**, and **Data Control Language (DCL)**, each serving distinct purposes in database management.

Explanation

- **Data Definition Language (DDL):**
 - Used to define or modify database structures (e.g., tables, schemas).
 - Examples: `CREATE`, `ALTER`, `DROP`.
- **Data Manipulation Language (DML):**
 - Used to manipulate data within tables (e.g., insert, update, delete, query).
 - Examples: `SELECT`, `INSERT`, `UPDATE`, `DELETE`.

- **Data Control Language (DCL):**
 - Used to manage permissions and security.
 - Examples: GRANT , REVOKE , DENY .

Example

- **DDL Example:**

```
CREATE TABLE TestTable (  
    ID INT PRIMARY KEY,  
    Name NVARCHAR(50)  
);
```

Creates a new table named TestTable .

- **DML Example:**

```
INSERT INTO TestTable (ID, Name)  
VALUES (1, 'Sample Product');
```

Inserts a record into TestTable .

- **DCL Example:**

```
GRANT SELECT ON TestTable TO User1;
```

Grants User1 permission to query TestTable .

Class Work

1. **Categorization Activity:** Provide a list of T-SQL statements (e.g., CREATE , SELECT , GRANT , UPDATE , DROP) and categorize each as DDL, DML, or DCL.
2. **Lab Exercise:** Write and execute a T-SQL script in SSMS that:
 - Creates a table named SampleCustomers in AdventureWorks 2022.
 - Inserts one record into it.
 - Grants SELECT permission to a hypothetical user (TestUser).Save the script as StatementCategories.sql .

Sub-Session 3: Data Types Supported by Transact-SQL

Definition

T-SQL data types define the type of data a column or variable can store in SQL Server, such as numbers, text, dates, or binary data.

Explanation

- **Categories of Data Types:**
 - **Numeric:** INT , DECIMAL , FLOAT (e.g., for IDs, prices).
 - **Character:** CHAR , VARCHAR , NVARCHAR (e.g., for names, descriptions).
 - **Date and Time:** DATE , DATETIME , TIME (e.g., for order dates).
 - **Binary:** BINARY , VARBINARY (e.g., for images or files).
 - **Other:** XML , JSON , UNIQUEIDENTIFIER (e.g., for structured data or GUIDs).
- **Key Considerations:**
 - Choose appropriate data types to optimize storage and performance.
 - NVARCHAR supports Unicode for international characters, unlike VARCHAR .
 - Precision and scale are critical for DECIMAL (e.g., DECIMAL(10,2) for currency).

Example

- **Data Type Example** (using AdventureWorks 2022):

```
USE AdventureWorks2022;  
SELECT ProductID AS ID, -- INT  
       Name AS ProductName, -- NVARCHAR  
       ListPrice AS Price, -- MONEY  
       SellStartDate AS StartDate -- DATETIME  
FROM Production.Product  
WHERE ListPrice > 500;
```

This query uses columns with different data types: INT , NVARCHAR , MONEY , and DATETIME .

Class Work

1. **Identification Task:** List 5 T-SQL data types (e.g., INT , NVARCHAR , DATETIME , DECIMAL , VARBINARY) and provide one example use case for each (e.g., INT for product IDs).
2. **Lab Exercise:** Write a T-SQL script to create a table in AdventureWorks 2022 called OrderSummary with columns: OrderID (INT), CustomerName (NVARCHAR), OrderDate (DATE), and TotalAmount

(DECIMAL). Insert one sample record. Save as `DataTypes.sql` .

Sub-Session 4: Transact-SQL Language Elements

Definition

T-SQL language elements include components like variables, operators, predicates, expressions, and comments used to construct queries and scripts.

Explanation

- **Variables:** Store temporary data (e.g., `DECLARE @Count INT;`).
- **Operators:** Perform operations (e.g., `+` for addition, `=` for comparison).
- **Predicates:** Conditions that evaluate to true/false (e.g., `WHERE Price > 100`).
- **Expressions:** Combinations of values, operators, and functions (e.g., `ListPrice * 1.1`).
- **Comments:** Used for documentation (`-- Single-line` or `/* Multi-line */`).
- **Batches:** Groups of T-SQL statements executed together, separated by `GO` .

Example

- **T-SQL Language Elements Example:**

```
USE AdventureWorks2022;
-- Declare a variable
DECLARE @MinPrice DECIMAL(10,2) = 1000.00;
-- Use a predicate and expression in a query
SELECT ProductID, Name, ListPrice
FROM Production.Product
WHERE ListPrice > @MinPrice; -- Predicate
/* This query retrieves products priced above the variable value */
```

Class Work

1. **Identification Activity:** Provide a T-SQL script snippet and identify variables, operators, predicates, expressions, and comments within it.
2. **Lab Exercise:** Write a T-SQL script that:
 - Declares a variable `@MaxPrice` (DECIMAL).
 - Uses it in a WHERE clause to filter `Production.Product` records.

- Includes both single-line and multi-line comments.

Save as `LanguageElements.sql` .

Sub-Session 5: Sets and Predicate Logic

Definition

- **Sets:** Collections of data (e.g., rows in a table) manipulated using T-SQL operations like UNION, INTERSECT, or EXCEPT.
- **Predicate Logic:** Logical conditions in T-SQL queries (e.g., WHERE , HAVING) that filter data based on true/false evaluations.

Explanation

- **Sets:**
 - SQL Server operates on sets of data, not individual rows.
 - Set operations combine results (e.g., UNION combines rows from two queries).
- **Predicate Logic:**
 - Predicates are conditions (e.g., Price > 100 AND Category = 'Bikes').
 - Evaluated using logical operators: AND , OR , NOT .
 - Used in WHERE or HAVING clauses to filter rows or groups.

Example

- **Set Operation Example:**

```
USE AdventureWorks2022;
SELECT ProductID, Name
FROM Production.Product
WHERE ProductSubcategoryID = 1
UNION
SELECT ProductID, Name
FROM Production.Product
WHERE ProductSubcategoryID = 2;
```

Combines products from two subcategories.

- **Predicate Logic Example:**

```
SELECT ProductID, Name, ListPrice
FROM Production.Product
WHERE ListPrice > 500 AND Color = 'Red';
```

Class Work

1. **Set Operation Lab:** Write a T-SQL query using `UNION` to combine products from `Production.Product` where `Color = 'Black'` and `Color = 'Silver'`. Save as `SetOperation.sql`.
2. **Predicate Logic Exercise:** Write a query to retrieve `Sales.SalesOrderHeader` records where `TotalDue > 1000` and `OrderDate` is in 2020. Use `AND` and `OR` operators. Save as `PredicateLogic.sql`.

Sub-Session 6: Logical Order of Operators in the SELECT Statement

Definition

The logical order of operators in a T-SQL `SELECT` statement defines the sequence in which clauses are processed by SQL Server to produce query results.

Explanation

- **Logical Order:**
 - i. **FROM:** Specifies the table(s) to query.
 - ii. **WHERE:** Filters rows based on conditions.
 - iii. **GROUP BY:** Groups rows by specified columns.
 - iv. **HAVING:** Filters groups based on conditions.
 - v. **SELECT:** Specifies columns to retrieve.
 - vi. **ORDER BY:** Sorts the result set.
- **Key Points:**
 - The logical order differs from the written order in a query (e.g., `SELECT` is written first but processed later).
 - Understanding this order helps optimize queries and predict results.

Example

- **SELECT Statement Example:**

```
USE AdventureWorks2022;  
SELECT ProductSubcategoryID, AVG(ListPrice) AS AvgPrice  
FROM Production.Product  
WHERE ListPrice > 0  
GROUP BY ProductSubcategoryID  
HAVING AVG(ListPrice) > 500  
ORDER BY AvgPrice DESC;
```

Logical Processing:

- i. FROM: Access `Production.Product` .
- ii. WHERE: Filter rows where `ListPrice > 0` .
- iii. GROUP BY: Group by `ProductSubcategoryID` .
- iv. HAVING: Keep groups where average price > 500.
- v. SELECT: Compute `AVG(ListPrice)` .
- vi. ORDER BY: Sort by `AvgPrice` descending.

Class Work

1. **Diagram Activity:** Draw a flowchart of the logical order of a `SELECT` statement, labeling each step (FROM, WHERE, etc.).
2. **Lab Exercise:** Write a T-SQL query for AdventureWorks 2022 that:
 - Retrieves `CustomerID` and total `TotalDue` from `Sales.SalesOrderHeader` .
 - Groups by `CustomerID` .
 - Filters groups with total `TotalDue > 5000` .
 - Orders by total `TotalDue` descending.Save as `SelectOrder.sql` . Explain the logical order of execution.

Additional Class Work Summary

1. **Group Project:** Divide students into groups to create a presentation summarizing T-SQL statement categories, data types, and the logical order of `SELECT` statements. Each group covers one sub-session.
2. **Practical Assignment:** Write 3 T-SQL queries for AdventureWorks 2022:
 - A DDL query to create a table.

- A DML query to insert and select data.
 - A query using predicate logic and set operations.
- Save as separate .sql files and execute in SSMS. Submit scripts and results.

3. Quiz:

- What is the difference between VARCHAR and NVARCHAR ? (Answer: NVARCHAR supports Unicode.)
- Name one DML statement. (Answer: SELECT .)
- What is the first step in the logical order of a SELECT statement? (Answer: FROM.)

Deliverables Mapping

- **Student Guide (SG):** Session 4
- **Exercise Package (XP):** Session 4
- **Trainer Guide (TG):** Session 4

Resources

- **OnlineVarsity:**
 - Download the *SQL Server–The Definitive Guide* eBook.
 - Access Glossary for terms like “T-SQL,” “DDL,” “DML,” “Predicate.”
 - Use Practice 4 Me for self-assessment questions.
- **Library References:**
 - *Murach’s SQL Server 2022 for Developers* by Bryan Syverson.
 - *T-SQL Fundamentals, 4th Edition* by Itzik Ben-Gan.

Notes for Instructors

- Use the AdventureWorks 2022 database for all hands-on exercises to leverage its pre-installed schema.
- Demonstrate T-SQL queries live in SSMS, highlighting syntax and execution.
- Ensure students save scripts with meaningful names and proper comments for organization.
- Monitor lab exercises to assist with syntax errors or connection issues.