<https://mva.microsoft.com/en-US/training-courses/querying-microsoft-sql-server-2012-databases-jump-start-8241>

--ddl or data definition language: create alter drop

--dml or data manipulation language: select insert update delete (CRUD operations)

--dcl or data control language: grant revoke deny

**T-SQL enforces operator precedence**

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

**T-SQL functions**:

String functions: SUBSTRING,LEFT, RIGHT,LEN,DATALENGTH, REPLACE,REPLICATE,UPPER, LOWER,RTRIM, LTRIM

Date and time functions: GETDATE,SYSTDATETIME,GETUTCDATE,DATEADD,DATEDIFF,YEAR,MONTH,DAY

Aggregate functions: SUM,MIN,MAX,AVG,COUNT

**T-SQL 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, can declare and initialize in the same statement

**DECLARE @MyVar int = 30;**

**T-SQL 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 OrderQty \* UnitPrice ...**

**T-SQL 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 a SQL Server T-SQL command

**T-SQL flow control, errors and transactions**:

Used in programmatic code objects such as stored procedures, triggers, statement blocks

Flow control: IF...ELSE,WHILE,BREAK,CONTINUE,BEGIN...END

Error handling: TRY…CATCH

Transaction control: BEGIN TRANSACTION,COMMIT TRANSACTION,ROLLBACK TRANSACTION

The order in which a query is written is not the order in which it is evaluated by the server. The order is:

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>

USE AdventureWorks2014;

SELECT SalesPersonID, YEAR(OrderDate) AS OrderYear

FROM Sales.SalesOrderHeader

WHERE CustomerID = 29974

GROUP BY SalesPersonID, YEAR(OrderDate)

HAVING COUNT(\*) > 1

ORDER BY SalesPersonID, OrderYear;

When performance tuning, using SELECT 1 gives you stats for just speaking to sql server on network(rather than what happens inside the database engine)