**T-SQL Quering** = Transact-Structured Query Language

Source: <https://www.dataquest.io/blog/sql-vs-t-sql/>

**SQL vs T-SQL: Understanding the Differences**

*SQL or T-SQL — which one do you need to learn?*

SQL ​and T-SQL both see heavy use in the database and data science industries. But what exactly are they? These two query languages are very similar, both in name and in what they can do, so the distinction between them can be difficult to understand.

In this post, we're going to: define what Standard SQL and T-SQL are, investigate the differences between them, provide examples of each, and summarize which you should be learning and why.

* Define what Standard SQL and T-SQL are
* Investigate the differences between them
* Provide examples of each
* Summarize which you should be learning and why

**What is Standard SQL?**

Standard SQL, usually referred to simply as "SQL," is a type of programming language called a query language. Query languages are used for communicating with a database.

SQL is used for adding, retrieving, or updating data stored in a database. It is used across many different types of databases. Meaning, if you learn the [basics of SQL](https://www.dataquest.io/blog/sql-basics/), you will be in a good position for a career in data.

Databases and the data stored within them are a core part of how many companies operate. An easy example is with a retailer that might store order or customer information in a database. SQL is a programming language that allows the company to work with that data.

**What is T-SQL?**

T-SQL, which stands for Transact-SQL and is sometimes referred to as TSQL, is an extension of the SQL language used primarily within Microsoft SQL Server. This means that it provides all the functionality of SQL but with some added extras.

You can think of it a bit like a SQL dialect — it's very similar to regular SQL, but it has a few extras and differences that make it unique.

Despite​ the clear and rigid specifications of standard SQL, it does allow for database companies to add their own extensions to set them apart from other products. T-SQL is an example of this for Microsoft SQL Server databases — ​T-SQL is central to the software and runs most operations within it.

​Most major database vendors offer their own SQL language extensions for their own products, and T-SQL is one of the most widely-used examples of these (because Microsoft SQL server is popular).

Put simply: when you are writing queries within Microsoft SQL Server, you are effectively using T-SQL. All applications that communicate with SQL Server, regardless of the application's user interface, do so by sending T-SQL statements to the server.

However, ​in addition to SQL Server, other database management systems (DBMS) also support T-SQL. Another Microsoft product, Microsoft Azure SQL Database​, supports most features of T-SQL.

T-SQL has been designed to make working with those databases that support it easier and more efficient​.

## What is the difference between SQL and T-SQL?

Now we have covered the basics of both, let's take a look at the main differences:

### Difference #1

The obvious difference is in what they are designed for: SQL is a​ query language used for manipulating data stored in a database. T-SQL is also a query language, but it's an extension of SQL that is primarily used in Microsoft SQL Server databases and software.

### Difference #2

SQL is open-source​. T-SQL is developed and owned by Microsoft.

### Difference #3

SQL statements are executed one at a time, also known as "non-procedural." T-SQL executes statements in a "procedural" way, meaning that the code will be processed as a block, logically and in a structured order.

There are advantages and disadvantages to each approach, but from a learner perspective, this difference isn't too important. You'll be able to get and work with the data you want in either language, ​it's just that the way you go about doing that will vary a bit depending on which language you're using and the specifics of your query.

### Difference #4

On top of these more general differences, SQL and T-SQL also have some slightly different command key words. T-SQL also features functions that are not part of regular SQL.

An example of this is how in ​we ​select the top ​X number of rows.​ In standard SQL, ​we ​would use the LIMIT keyword​. In T-SQL, we use the TOP keyword.

Both of these​ commands do the same thing​, as we can see in the examples below. Both queries will return the top ten rows in the users table ordered by the age column.

#### SQL Example

SELECT \*

FROM users

ORDER BY age

LIMIT 10;

#### T-SQL Example

SELECT TOP 10 (\*)

FROM users

ORDER BY age;

### Difference #5

Finally, and as referenced before, T-SQL offers functionality that does not appear in regular SQL. One example is the ISNULL function. This will replace NULL values coming from a specific column. The below would return an age of “0” for any rows that have a value of NULL in the age column.

SELECT ISNULL(0, age)

FROM users;

(There are [ways of doing this in standard SQL too](https://stackoverflow.com/questions/9877533/replace-nulls-values-in-sql-using-select-statement), of course, but the commands are slightly different.)

These are just a couple of code differences to give you an idea of how the two compare, but of course, there are many more. You can learn more about [SQL commands](https://www.dataquest.io/blog/sql-commands/) with our extensive guide. And of course, Microsoft has [documentation for working with T-SQL](https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15).

## Which is better to learn?

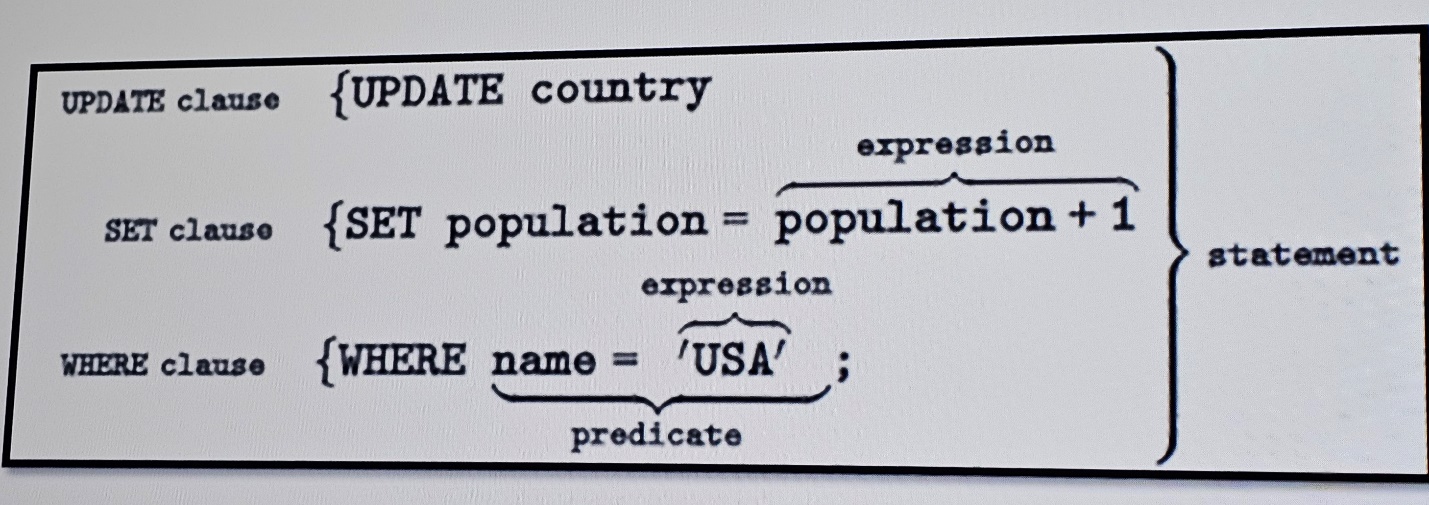
If you want to work with databases in any way, or if you're seeking a data job, [learning SQL](https://www.dataquest.io/blog/why-sql-is-the-most-important-language-to-learn/) is a necessity.

As T-SQL is an extension of SQL, you will need to learn the basics of SQL before starting. If you learn T-SQL first, you will end up picking up knowledge of standard SQL anyway.

With most things, which you choose to learn should depend on what you are trying to achieve. If you are going to be working with Microsoft SQL server, then it is worth learning more about T-SQL. If you are a beginner looking to get started in using databases, then begin with learning about SQL.

**T-SQL Quering** = Transact-Structured Query Language

* Clauses
* Expressions
* Predicates
* Queries
* Statements
* SQL statements also include the semicolon (“;”) statement terminator



**SQL Statement Types**

|  |  |  |
| --- | --- | --- |
| **Data Manipulation Language (DML)** | **Data Definition Language (DDL)** | **Data Control Language (DCL)** |
| Statements for quering and modifying data:   * **SELECT** * **INSERT** * **UPDATE** * **DELETE** | Statements for defining database objects:   * **CREATE** * **ALTER** * **DROP** | Statements for assigning security permissions:   * **GRANT** * **REVOKE** * **DENY** |

**T-SQL Elements:**

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

**T-SQL Elements – Predicates**

* a predicates is an expression that evaluates to TRUE, FALSE or UNKNOWN
* predicates are used in the search condition oh WHERE clauses and HAVING clauses, the join conditions of FROM clauses and other constructs where a Boolean value is required
* some of the predicates are ALL, ANY, BETWEEN, IN , LIKE, OR, SOME

**T-SQL Elements – Operators**

* Arithmetic Operators

+ (Add), - (Subtract), \* (Multiply), / (Divide), % (Modulo = restul împărțirii)

* Assignment Operator

the equal sign (=)

* Bitwise Operators

& (Bitwise AND), &= (Bitwise AND EQUALS), | (Bitwise OR), ...

* Comparison Operators

= (Equals), > (Greater Than), < (Less Than), >= (Greater Than or Equal To), <= (Less Than or Equal To), <> (Not Equal To)

* Compound Operators

+= (Add EQUALS), -= (Subtract EQUALS), \*= (Multiply EQUALS), /= (Divide EQUALS), %= (Modulo EQUALS),...

Example:

--Compound Operators

DECLARE @x1 int = 27;

SET @x1 += 2;

SELECT @x1 AS Added\_2;

--Logical Operators

SELECT Name

FROM sys.system\_views

WHERE Name LIKE 'dm%';

**T-SQL Elements – Functions**

|  |  |  |
| --- | --- | --- |
| **String Functions** | **Date and Time** | **Agregate Functions** |
| SUBSTRING  LEFT, RIGHT  LEN  REPLACE  REPLICATE  UPPER, LOWER  LTRIM, RTRIM  STUFF  SOUNDEX | GETDATE  SYSDATETIME  DATEADD  DATEDIFF  YEAR  MONTH  DAY  DATENAME  DATEPART  ISDATE | SUM  MIN  MAX  AVG  COUNT  COUNT\_BIG  STDEV  STDEVP  VAR |

**String Functions**

SUBSTRING = extracts some characters from a string.

SUBSTRING (expression, start, length)

SUBSTRING('abcdef', 2, 3); 🡪 bcd

LEFT, RIGHT = extracts a number of characters from a string (starting from right)

RIGHT (character\_expression, integer\_expression)

SELECT RIGHT('abcdefg', 2); 🡪 fg

LEN = returns the length of a string (ignorând spațiile de la început și de la sfârșit)

LEN (string\_expression)

LEN('a b'); 🡪 3

REPLACE = replaces all occurrences of a substring within a string, with a new substring

REPLACE ( string\_expression , string\_pattern , string\_replacement )

SELECT REPLACE('abcdefghicde','cde','xxx'); 🡪 abxxxfghixxx

REPLICATE = repeats a string a specified number of times

REPLICATE ( string\_expression , integer\_expression )

UPPER = converts a string to upper-case, LOWER = converts a string to lower-case

LTRIM = removes leading spaces from a string, RTRIM = removes trailing spaces from a string

STUFF = deletes a part of a string and then inserts another part into the string, starting at a specified position

**Tip:** Also look at the [REPLACE()](https://www.w3schools.com/sql/func_sqlserver_replace.asp) function.

STUFF (character\_expression , start , length , replace\_with\_expression)

SELECT STUFF('abcdef', 2, 3, 'ijklmn'); 🡪 aijklmnef

SOUNDEX = returns a four-character (SOUNDEX) code to evaluate the similarity of two strings

The SOUNDEX() converts the string to a four-character code based on how the string sounds when spoken

**Tip:** Also look at the [DIFFERENCE()](https://www.w3schools.com/sql/func_sqlserver_difference.asp) function.

Example:

* The following example shows the SOUNDEX function and the related DIFFERENCE function. In the first example, the standard SOUNDEX values are returned for all consonants. Returning the SOUNDEX for Smith and Smythe returns the same SOUNDEX result because all vowels, the letter y, doubled letters, and the letter h, are not included.

-- Using SOUNDEX

SELECT SOUNDEX ('Smith'), SOUNDEX ('Smythe');

Here is the result set. Valid for a Latin1\_General collation:

S530 S530

* The DIFFERENCE function compares the difference of the SOUNDEX pattern results. The following example shows two strings that differ only in vowels. The difference returned is 4, the lowest possible difference.

-- Using DIFFERENCE

SELECT DIFFERENCE('Smithers', 'Smythers');

Here is the result set. Valid for a Latin1\_General collation:

4

SELECT DIFFERENCE('Anothers', 'Brothers'); 🡪 2