**SQL Server IF…ELSE Condition Statement: T-SQL Select Query Example**

**Why do you need Conditional Statements?**

Conditional statements in the SQL server help you to define different logics and actions for different conditions. It allows you to perform different actions based on conditions defined within the statement. In real life, you perform many actions dependent on the outcome of some other activity or situation.

Some real-time examples of SQL case statement are:

* If it rains tomorrow, I will plan on a road trip.
* If flight tickets are less than $400 from my city, then I will go on vacation in Europe, else I will prefer some nearby tourist spot.

Here, you can see that one action, like Road trip above, is **conditionally dependent** on the outcome of another activity which is "whether it will rain or not tomorrow!"

Similarly, MS SQL also provides the capability to execute T-SQL statement conditionally.

In this tutorial, you will learn:

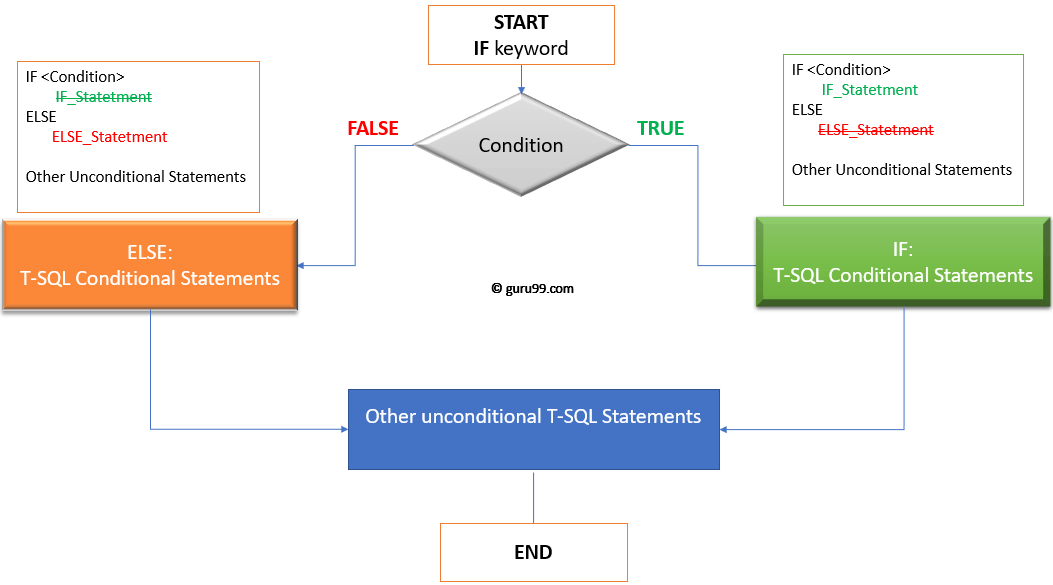
* [IF… Else statement](https://www.guru99.com/sql-server-if-else.html#1)
  + [IF...ELSE with the variable in Boolean Expression.](https://www.guru99.com/sql-server-if-else.html#2)
  + [IF...ELSE with Begin End](https://www.guru99.com/sql-server-if-else.html#3)
* [IF statement with No Else](https://www.guru99.com/sql-server-if-else.html#4)
* [Nested IF…Else Statements](https://www.guru99.com/sql-server-if-else.html#5)

**IF… Else statement in SQL Server**

In MS SQL, **IF…ELSE** is a type of **Conditional statement**

Any T-SQL statement can be executed conditionally using **IF… ELSE**.

Below figure explains IF ELSE in SQL server:

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF1.png)How if and else works

* If the condition evaluates to **True,** then T-SQL statements followed by **IF** condition in SQL server will be executed.
* If the condition evaluates to **False,** then T-SQL statements followed by **ELSE** keyword will be executed.
* Once, either IF T-SQL statements or ELSE T-SQL statement is executed, then other unconditional T-SQL statements continues execution.

### IF… Else Syntax and rules in SQL

**Syntax:**

IF <Condition>

{Statement | Block\_of\_statement}

[ ELSE

{Statement | Block\_of\_statement}]

**Rules:**

* The condition should be **Boolean Expression, i.e.,**condition results in Boolean value when evaluated.
* IF ELSE statement in SQL can conditionally handle a single T-SQL statement or block of T-SQL statements.
* Block of statement should start with keyword BEGIN and close with keyword END.
* Using BEGIN and END helps SQL server to identify statement block that needs to be executed and separate it from rest of the T-SQL statements which are not part of IF…ELSE T-SQL block.
* ELSE is optional.

### IF…ELSE with the only numeric value in Boolean Expression.

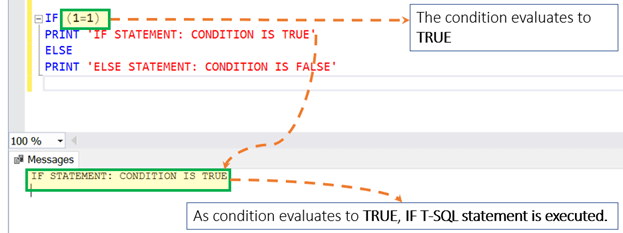
**Condition: TRUE**

IF (1=1)

PRINT 'IF STATEMENT: CONDITION IS TRUE'

ELSE

PRINT 'ELSE STATEMENT: CONDITION IS FALSE'

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF2.png)

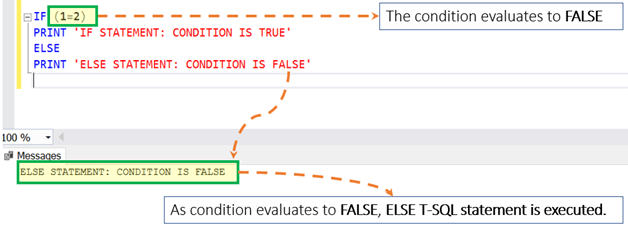
**Condition: FLASE**

IF (1=2)

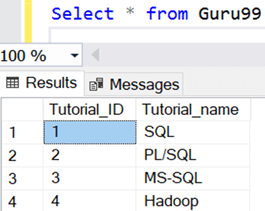
PRINT 'IF STATEMENT: CONDITION IS TRUE'

ELSE

PRINT 'ELSE STATEMENT: CONDITION IS FALSE'

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF3.png)

**Assumption:** Assume that you have the table as **'Guru99'** with two columns and four rows as displayed below:

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF4.png)

We will use**'Guru99'** table in further examples

### IF...ELSE with the variable in Boolean Expression.

**Condition: TRUE**

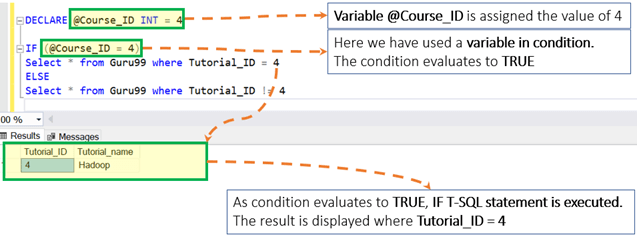
DECLARE @Course\_ID INT = 4

IF (@Course\_ID = 4)

Select \* from Guru99 where Tutorial\_ID = 4

ELSE

Select \* from Guru99 where Tutorial\_ID != 4

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF5.png)

**Condition: FLASE**

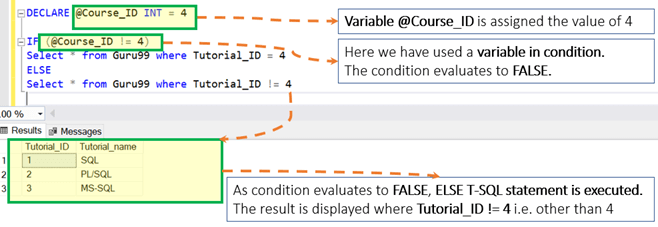
DECLARE @Course\_ID INT = 4

IF (@Course\_ID != 4)

Select \* from Guru99 where Tutorial\_ID = 4

ELSE

Select \* from Guru99 where Tutorial\_ID != 4

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF6.png)

### IF...ELSE with Begin End

**Condition: TRUE**

DECLARE @Course\_ID INT = 2

IF (@Course\_ID <=2)

BEGIN

Select \* from Guru99 where Tutorial\_ID = 1

Select \* from Guru99 where Tutorial\_ID = 2

END

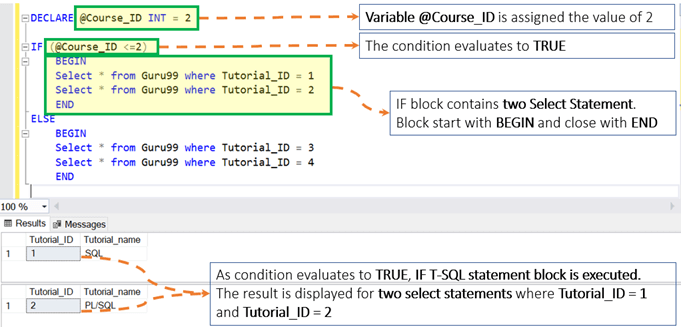
ELSE

BEGIN

Select \* from Guru99 where Tutorial\_ID = 3

Select \* from Guru99 where Tutorial\_ID = 4

END

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF7.png)

**Condition: FALSE**

DECLARE @Course\_ID INT = 2

IF (@Course\_ID >=3)

BEGIN

Select \* from Guru99 where Tutorial\_ID = 1

Select \* from Guru99 where Tutorial\_ID = 2

END

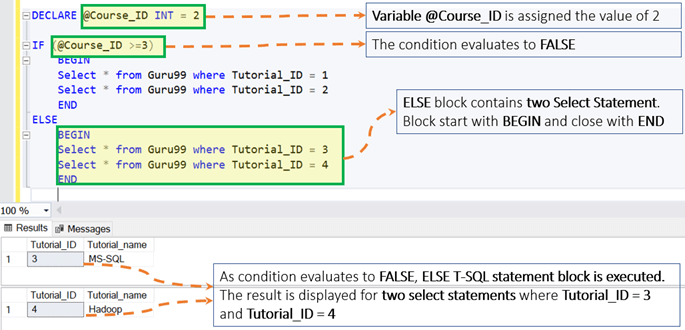
ELSE

BEGIN

Select \* from Guru99 where Tutorial\_ID = 3

Select \* from Guru99 where Tutorial\_ID = 4

END

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF8.png)

## IF statement with No Else

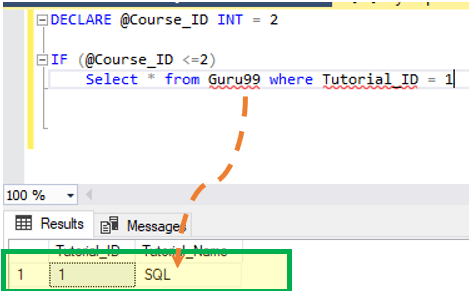
You can use an IF statement in SQL without an ELSE part. Remember we stated that the ELSE part is optional. For example:

DECLARE @Course\_ID INT = 2

IF (@Course\_ID <=2)

Select \* from Guru99 where Tutorial\_ID = 1

It prints the following:

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF9.png)

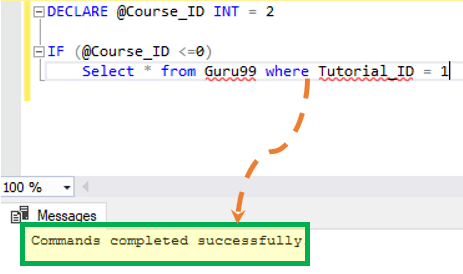
Executing the false condition will give no output. Consider the following query

DECLARE @Course\_ID INT = 2

IF (@Course\_ID <=0)

Select \* from Guru99 where Tutorial\_ID = 1

The result it

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF10.png)

## Nested IF…Else Statements

Unlike other [programming languages](https://www.guru99.com/best-programming-language.html), you cannot add an ELSE IF statement within an IF ELSE condition in SQL. This is why you can nest IF ELSE in SQL query statements. It is demonstrated below:

DECLARE @age INT;

SET @age = 60;

IF @age < 18

PRINT 'underage';

ELSE

BEGIN

IF @age < 50

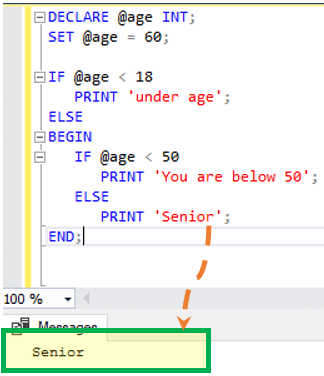
PRINT 'You are below 50';

ELSE

PRINT 'Senior';

END;

* In this example, the code will print underage if the value of @age is below 18.
* If not, the ELSE part will be executed. The ElSE part has a nested IF…ELSE.
* If the value of @age is below 50, this will print You are below 50. If none of these conditions is true, the code will print Senior.

[](https://www.guru99.com/images/1/030819_0910_SQLServerIF11.png)