

Use Stack Overflow for Teams at work to find answers in a private and secure environment. Get your first 10 users free. [Sign up.](#)

How do I perform an IF...THEN in an SQL SELECT?

Asked 11 years ago Active 3 months ago Viewed 3.5m times

▲ How do I perform an IF...THEN in an SQL SELECT statement?

1415 For example:

▼ `SELECT IF(Obsolete = 'N' OR InStock = 'Y' ? 1 : 0) AS Saleable, * FROM Product`

★ 297 sql sql-server tsql if-statement case

edited Mar 26 '18 at 6:09



DineshDB

4,071 4 24 39

asked Sep 15 '08 at 14:34



Eric Labashosky

12.5k 12 34 31

12 You may want to take a look at this [link](#). Regarding: SQL WHERE clauses: Avoid CASE, use Boolean logic – [Somebody](#) Feb 14 '12 at 21:21

3 @Somebody: not really relevant because the article talks about using logical rewrite rules to convert an implication into a disjunction. The clue is the word 'logical' i.e. something that resolves to true or false, which doesn't apply to projection. TL;DR article applies to WHERE and CHECK but not SELECT. – [onedaywhen](#) May 11 '16 at 16:06

4 @MartinSmith's answer is the most elegant - use IIF in SQL 2012+. – [Murray Foxcroft](#) Mar 27 '17 at 13:06

25 Answers

▲ The CASE statement is the closest to IF in SQL and is supported on all versions of SQL Server.

1000

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

ELSE 0



```
END AS bit) AS Saleable, *
FROM Product
```

You only need to do the `CAST` if you want the result as a Boolean value. If you are happy with an `int`, this works:

```
SELECT CASE
    WHEN Obsolete = 'N' OR InStock = 'Y'
    THEN 1
    ELSE 0
END AS Saleable, *
FROM Product
```

`CASE` statements can be embedded in other `CASE` statements and even included in aggregates.

SQL Server Denali (SQL Server 2012) adds the `IIF` statement which is also available in [access](#) (pointed out by [Martin Smith](#)):

```
SELECT IIF(Obsolete = 'N' OR InStock = 'Y', 1, 0) AS Saleable, * FROM Product
```

edited Jun 3 at 15:49



Peter Mortensen

14.4k 19 88 117

answered Sep 15 '08 at 14:38



Darrel Miller

118k 28 173 228

53 Just a additional word of caution don't enclose your conditions in brakets when using case. Took quite a bit of time to realise that :) – [Archan Mishra](#) Sep 13 '11 at 15:41

17 and don't forget the END – [Simon_Weaver](#) Jan 28 '14 at 10:52

8 and the AS bit! – [Cas Bloem](#) Apr 22 '14 at 12:49

8 Case, When, Else and End should be indented parallel (along same line) - and only then should be indented further inwards - works best for me. – [Ujjwal Singh](#) Jul 1 '14 at 21:24

5 @ReeveStrife Only iif SQL Server 2012+ – [stuartdotnet](#) May 30 '16 at 3:26

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

SELECT CASE <variable> WHEN <value>      THEN <returnvalue>
                WHEN <othervalue> THEN <returnthis>
                ELSE <returndefaultcase>

        END AS <newcolumnname>
FROM <table>

```

The extended case:

```

SELECT CASE WHEN <test>      THEN <returnvalue>
        WHEN <othertest> THEN <returnthis>
        ELSE <returndefaultcase>

        END AS <newcolumnname>
FROM <table>

```

You can even put case statements in an order by clause for really fancy ordering.

edited Oct 28 '15 at 0:07



Christoph

3,014 2 20 37

answered Sep 15 '08 at 15:10



Jonathan

16k 10 58 78

-
- 30 I know this is old, but i think it should be noted that you can add a AS Col_Name after the END to name the resulting column – Ben Jun 18 '12 at 10:22
-
- 8 I always feel like the 2nd one is simpler. – Hogan Apr 15 '16 at 20:32
-
- 3 Agreed, I almost always end up using the extended case statement because the conditions I want to test on are always more complex than just the one variable itself. It also just feels easier to me to read. – magnum_pi May 18 '16 at 16:34
-
- 1 Good explanation of both situations, with or without variable. With the variable the condition needs to satisfy an equality between the variable after the case statement and the one you base your condition on, without variable you can add a self-sufficient condition to test. – Remus.A Feb 23 '18 at 11:24
-
- I am more convenient with the second option. The two are equally fine. – Stanley Okpala Nwosa Oct 20 '18 at 9:30
-

From SQL Server 2012 you can use the [IIF function](#) for this.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

This is effectively just a shorthand (albeit not standard SQL) way of writing `CASE` .

I prefer the conciseness when compared with the expanded `CASE` version.

Both `IIF()` and `CASE` resolve as expressions within a SQL statement and can only be used in well-defined places.

The `CASE` expression cannot be used to control the flow of execution of Transact-SQL statements, statement blocks, user-defined functions, and stored procedures.

If your needs can not be satisfied by these limitations (for example, a need to return differently shaped result sets dependent on some condition) then SQL Server does also have a procedural `IF` keyword.

```
IF @IncludeExtendedInformation = 1
BEGIN
    SELECT A,B,C,X,Y,Z
    FROM T
END
ELSE
BEGIN
    SELECT A,B,C
    FROM T
END
```

[Care must sometimes be taken to avoid parameter sniffing issues with this approach however.](#)

edited Jun 3 at 15:54



Peter Mortensen

14.4k 19 88 117

answered Jul 20 '11 at 23:39



Martin Smith

363k 63 613 717

4 This should be the answer if you want a IF.. then statement in SQL. – Mr.J Mar 16 '17 at 3:27



You can find some nice examples in [The Power of SQL CASE Statements](#), and I think the statement that you can use will be something like this (from [4guysfromrolla](#)):

86

`SELECT`

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
WHEN 'M' THEN 'Male'
WHEN 'F' THEN 'Female'
```

```
END
FROM Employees
```

edited May 24 '11 at 17:31



Peter Mortensen

14.4k 19 88 117

answered Sep 15 '08 at 14:39



sven

11.5k 9 45 61

- 3 see: meta.stackexchange.com/questions/103053/... for an interesting discussion. I the two links you provide do add extra context, which I support. – Sam Saffron Aug 19 '11 at 2:47
- 1 The refer is really useful and highly recommended in case of additional details – baymax Dec 7 '16 at 14:56

Use CASE. Something like this.

75

```
SELECT Salable =
    CASE Obsolete
    WHEN 'N' THEN 1
    ELSE 0
END
```

answered Sep 15 '08 at 14:37



palehorse

19.3k 3 35 45

48

```
SELECT
(CASE
    WHEN (Obsolete = 'N' OR InStock = 'Y') THEN 'YES'
    ELSE 'NO'
END) as Salable
, *
FROM Product
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



1,156 2 20 36



65.8k 26 151 187



45



```
SELECT
CASE
  WHEN OBSOLETE = 'N' or InStock = 'Y' THEN 'TRUE'
  ELSE 'FALSE'
END AS Salable,
*
FROM PRODUCT
```

answered Sep 15 '08 at 14:37

[Santiago Cepas](#)
3,517 2 19 28

Microsoft SQL Server (T-SQL)

44



In a select , use:

```
select case when Obsolete = 'N' or InStock = 'Y' then 'YES' else 'NO' end
```

In a where clause, use:

```
where 1 = case when Obsolete = 'N' or InStock = 'Y' then 1 else 0 end
```

edited Jun 3 at 15:50

[Peter Mortensen](#)
14.4k 19 88 117

answered Sep 15 '08 at 14:40

[user7658](#)
854 7 7

why wouldn't you just do where Obsolete = 'N' or InStock = 'Y' and cut the where in half practically – [maksymiuk](#) Jun 7 at 22:04

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

44

```

IF EXISTS(SELECT *
          FROM   Northwind.dbo.Customers
          WHERE  CustomerId = 'ALFKI')
    PRINT 'Need to update Customer Record ALFKI'
ELSE
    PRINT 'Need to add Customer Record ALFKI'

IF EXISTS(SELECT *
          FROM   Northwind.dbo.Customers
          WHERE  CustomerId = 'LARSE')
    PRINT 'Need to update Customer Record LARSE'
ELSE
    PRINT 'Need to add Customer Record LARSE'

```

Isn't this good enough for T-SQL?

edited Jun 3 at 15:52



Peter Mortensen

14.4k 19 88 117

answered Jan 6 '10 at 1:02



Ken

1,081 11 21

-
- 3 This isn't what the requester wanted, but is very useful to know that you can use if statements *outside* a select statement. – [Jonathan](#) Apr 10 '13 at 8:06
-
- 2 EXISTS is good because it kicks out of the search loop if item is found. A COUNT runs until the end of table rows. Nothing to do with question, but something to know. – [JustJohn](#) Feb 11 '16 at 21:12
-

Simple if-else statement in SQL Server:

30

```

DECLARE @val INT;
SET @val = 15;

IF @val < 25
    PRINT 'Hi Ravi Anand';
ELSE
    PRINT 'By Ravi Anand.';

GO

```

Nested If else statement in SQL Server -

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

SET @val = 15;

IF @val < 25
PRINT 'Hi Ravi Anand.';
ELSE
BEGIN
IF @val < 50
    PRINT 'what''s up?';
ELSE
    PRINT 'Bye Ravi Anand.';
END;

GO

```

edited Jun 3 at 15:57



Peter Mortensen

14.4k 19 88 117

answered Dec 17 '15 at 17:24



Ravi Anand

1,809 3 22 43

2 Late but is it usable inside SELECT as OP asked? – [abdul qayyum](#) Feb 17 at 10:23

Use a CASE statement:

24

```

SELECT CASE
    WHEN (Obsolete = 'N' OR InStock = 'Y')
    THEN 'Y'
    ELSE 'N'
END as Available

etc...

```

edited Nov 4 '16 at 12:45



Krizzy

77 1 3 6

answered Sep 15 '08 at 14:40

Christopher

A new feature, [IF](#) (that we can simply use), was added in SQL Server 2012:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

edited Sep 30 '17 at 0:12



twoleggedhorse

4,362 2 18 35

answered Apr 26 '16 at 15:10



sandeep rawat

3,494 1 9 31

1 This answer repeats (with less detail) what was [already provided in the answer by Martin Smith](#) several years ago. – [jk7](#) Oct 31 '18 at 19:47

1 @jk7 this was the first answer to the question . – [sandeep rawat](#) Nov 1 '18 at 2:46

2 Not from what I see. It says your answer was posted Apr 26 '16 and Martin's was posted Jul 20 '11. – [jk7](#) Nov 2 '18 at 14:40

Use pure bit logic:

24

```

DECLARE @Product TABLE (
    id INT PRIMARY KEY IDENTITY NOT NULL
    ,Obsolote CHAR(1)
    ,Instock CHAR(1)
)

INSERT INTO @Product ([Obsolote], [Instock])
VALUES ('N', 'N'), ('N', 'Y'), ('Y', 'Y'), ('Y', 'N')

;
WITH cte
AS
(
    SELECT
        'CheckIfInstock' = CAST(ISNULL(NULLIF(ISNULL(NULLIF(p.[Instock], 'Y'), 1), 'N'),
0) AS BIT)
        , 'CheckIfObsolote' = CAST(ISNULL(NULLIF(ISNULL(NULLIF(p.[Obsolote], 'N'), 0),
'Y'), 1) AS BIT)
        ,*
    FROM
        @Product AS p
)
SELECT
    'Salable' = c.[CheckIfInstock] & ~c.[CheckIfObsolote]
    ,*
FROM

```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

See [working demo. It then without case in SQL Server.](#)

For start, you need to work out the value of `true` and `false` for selected conditions. Here comes two [NULLIF](#):

```
for true: ISNULL(NULLIF(p.[Instock], 'Y'), 1)
for false: ISNULL(NULLIF(p.[Instock], 'N'), 0)
```

combined together gives 1 or 0. Next use [bitwise operators](#).

It's the most [WYSIWYG](#) method.

edited Jun 3 at 15:56



Peter Mortensen

14.4k 19 88 117

answered Jun 8 '13 at 22:21



Tomasito

1,430 18 37

-
- 17 -1 for Code Obfuscation. Seriously, this is about as far from WYSIWYG as you can get! A ruddy unreadable mess, and if I had to work on your code, I would be cursing all day long... sorry :-/ – [Heliac](#) Jun 22 '13 at 8:47
-
- 2 @Heliac put cte part in View and you will never see the mess. For long and complicated AND, OR, NOT it is more readable than CASE (that part outside cte of course). – [Tomasito](#) Jun 23 '13 at 9:52
-
- 1 I've given this a +1 for the neatness, once it's in a cte, but note that the answer is currently wrong for the question. You need an 'I' not an '&'. – [Mark Hurd](#) Nov 16 '16 at 0:19
-
- 2 Totally agree with @Heliac. While it is syntactically correct and works fine it is just not easily supportable. Putting it in a CTE will just move that piece of un-readable code somewhere else. – [objectNotFound](#) Nov 24 '17 at 17:04
-
- 1 The table method of checking combination could have its advantages. Using a table variable and joining it to the existing query could provide a set-based solution without a case. This answer is a poor example, but the table idea itself has merit. – [Suncat2000](#) Nov 30 '18 at 13:19
-



19

```
SELECT 1 AS Saleable, *
FROM @Product
WHERE ( Obsolete = 'N' OR InStock = 'Y' )
UNION
SELECT 0 AS Saleable, *
FROM @Product
WHERE NOT ( Obsolete = 'N' OR InStock = 'Y' )
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



17

```
SELECT CASE WHEN profile.nrefillno = 0 THEN 'N' ELSE 'R'END as newref
From profile
```

answered Aug 25 '15 at 9:44



atik sarker

3,076 27 34

Can you elaborate? – [Peter Mortensen](#) Jun 3 at 15:57

14

case statement some what similar to if in SQL server

```
SELECT CASE
    WHEN Obsolete = 'N' or InStock = 'Y'
    THEN 1
    ELSE 0
END as Saleable, *
FROM Product
```

answered Dec 9 '15 at 12:15



Chanukya

5,090 1 13 29

2 Could you please put some explanation on how this answers the question asked? – [Guanxi](#) Dec 9 '15 at 13:32@Guanxi: though not my answer, a 'case' generalizes an 'if-then-else' (from 2 cases to many) – [JosephDoggie](#) Oct 2 '18 at 16:39Can you elaborate? – [Peter Mortensen](#) Jun 3 at 15:57By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

CASE orweb2.dbo.Inventory.RegulatingAgencyName
  WHEN 'Region 1'
    THEN orweb2.dbo.CountyStateAgContactInfo.ContactState
  WHEN 'Region 2'
    THEN orweb2.dbo.CountyStateAgContactInfo.ContactState
  WHEN 'Region 3'
    THEN orweb2.dbo.CountyStateAgContactInfo.ContactState
  WHEN 'DEPT OF AGRICULTURE'
    THEN orweb2.dbo.CountyStateAgContactInfo.ContactAg
  ELSE (
    CASE orweb2.dbo.CountyStateAgContactInfo.IsContract
      WHEN 1
        THEN orweb2.dbo.CountyStateAgContactInfo.ContactCounty
      ELSE orweb2.dbo.CountyStateAgContactInfo.ContactState
    END
  )
END AS [County Contact Name]

```

edited Oct 11 '16 at 8:00



NzGuy

362 3 14

answered Feb 11 '16 at 21:35



JustJohn

681 13 31

-
- 1 The edit that reformatted the Case statements is all fine and dandy and makes it more understandable but the SQL would still all lump up in the view that is using it. – [JustJohn](#) Oct 12 '16 at 16:30
-
- 1 I'm just wandering why CASE become upvoted and marked as a answer instead of the IF which should have been the answer, like this one, this is still a CASE statement, not an IF. – [Mr.J](#) Mar 22 '17 at 7:16
-
- @Mr.J: though not my answer, a 'case' generalizes an 'if-then-else' (from 2 cases to many) – [JosephDoggie](#) Oct 2 '18 at 16:40
-

If you're inserting results into a table for the first time, rather than transferring results from one table to another, this works in Oracle 11.2g:

12

```


INSERT INTO customers (last_name, first_name, city)
  SELECT 'Doe', 'John', 'Chicago' FROM dual
  WHERE NOT EXISTS
    (SELECT '1' from customers
     where last_name = 'Doe'
     and first_name = 'John')

```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Oct 26 '12 at 15:30

Robert B. Grossman

16112

4 the tags say SQL Server, TSQL – [Malachi](#) Dec 27 '12 at 15:46

As an alternative solution to the CASE statement, a table-driven approach can be used:

10


```
DECLARE @Product TABLE (ID INT, Obsolete VARCHAR(10), InStock VARCHAR(10))
INSERT INTO @Product VALUES
(1, 'N', 'Y'),
(2, 'A', 'B'),
(3, 'N', 'B'),
(4, 'A', 'Y')

SELECT P.* , ISNULL(Stmt.Saleable,0) Saleable
FROM
    @Product P
    LEFT JOIN
        ( VALUES
            ( 'N', 'Y', 1 )
        ) Stmt (Obsolete, InStock, Saleable)
    ON P.InStock = Stmt.InStock OR P.Obsolete = Stmt.Obsolete
```

Result:

ID	Obsolete	InStock	Saleable
1	N	Y	1
2	A	B	0
3	N	B	1
4	A	Y	1

edited Jun 3 at 16:00

Peter Mortensen

14.4k1988117

answered Jan 31 '18 at 11:31

Serkan Arslan

9,39321536

SELECT CASE WHEN Obsolete = 'N' **or** InStock = 'Y' **THEN** 1 **ELSE** 0
END AS Saleable, *
FROM Product

8

answered Aug 8 '17 at 21:33
user8422856

For those who uses SQL Server 2012, IIF is a feature that has been added and works as an alternative to Case statements.

6

SELECT IIF(Obsolete = 'N' **OR** InStock = 'Y', 1, 0) **AS** Saleable, *
FROM Product

edited Apr 7 '14 at 14:16



demongolem

7,143 12 74 94

answered Jan 8 '14 at 10:18



Dibin

1,157 1 11 17

1 This answer repeats (with less detail) what was [already provided in the answer by Martin Smith](#) several years ago. – jk7 Oct 31 '18 at 19:44

SELECT IIF(Obsolete = 'N' **OR** InStock = 'Y',1,0) **AS** Saleable, * **FROM** Product

6

answered Nov 30 '16 at 11:05



SURJEET SINGH Bisht

798 8 14

- 6 Hi Surjeet Singh Bisht; your code might be correct, but with some context it would make a better answer; for example, you could explain how and why this proposed change would resolve the questioner's problem, perhaps including a link to the relevant documentation. That would make it more useful to them, and also more useful to other site readers who are looking for solutions to similar problems. – Vince Bowdren Nov 30 '16 at 16:27
- 5 This answer doesn't add anything new. In fact this exact same line has been part of the accepted answer [for over 5 years](#). – S.L. Barth Nov 30 '16 at 19:20

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



Question:

3

```
SELECT IF(Obsolete = 'N' OR InStock = 'Y' ? 1 : 0) AS Saleable, * FROM Product
```



ANSI:

```
Select  
  case when p.Obsolete = 'N'  
    or p.InStock = 'Y' then 1 else 0 end as Saleable,  
  p.*  
FROM  
  Product p;
```

Using aliases -- p in this case -- will help prevent issues.

edited Mar 6 at 7:32



Bikramjeet Singh

407 1 4 13

answered Nov 2 '18 at 15:44



David Cohn

49 4



You can have two choices for this to actually implement:

3

1. Using IIF, which got introduced from SQL Server 2012:

```
SELECT IIF ( (Obsolete = 'N' OR InStock = 'Y'), 1, 0) AS Saleable, * FROM Product
```



2. Using Select Case :

```
SELECT CASE  
  WHEN Obsolete = 'N' or InStock = 'Y'  
    THEN 1  
    ELSE 0  
  END as Saleable, *  
FROM Product
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



1

```
SELECT
  CAST(
    CASE WHEN Obsolete = 'N'
    or InStock = 'Y' THEN ELSE 0 END AS bit
  ) as Saleable, *
FROM
  Product
```

edited Mar 6 at 7:49



Bikramjeet Singh

407 1 4 13

answered Oct 8 '18 at 6:34



gii96

31 5

- 7 From Review: Welcome to Stack Overflow! Please don't answer just with source code. Try to provide a nice description about how your solution works. See: [How do I write a good answer?](#). Thanks – [seuniojeqep](#) Oct 8 '18 at 6:52
- 2 I think you will find this doesn't execute, because it is missing any output following the 'THEN' keyword. – [Dodecaphone](#) Apr 5 at 14:00
- Can you elaborate? – [Peter Mortensen](#) Jun 3 at 16:01