How do I check if a Sql server string is null or empty

208

I want to check for data but ignore it if it's null or empty. Currently the query is as follows...

```
Select
Coalesce(listing.OfferText, company.OfferText, '') As Offer_Text,
from tbl_directorylisting listing
  Inner Join tbl_companymaster company
   On listing.company_id= company.company_id
```

45

But I want to get company.OfferText if listing.Offertext is an empty string, as well as if it's null.

What's the best performing solution?
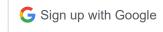
sql-server-2005

asked Dec 2 '08 at 14:31

digiguru
7,149   16   54   83

15 Answers

397

I think this:

```
SELECT
    ISNULL(NULLIF(listing.Offer_Text, ''), company.Offer_Text) AS Offer_Text
FROM ...
```

is the most elegant solution.

```
    temp := null;
  else
    temp := listing.Offer_Text; // may now be null or non-null, but not ''
// b) ISNULL:
if (temp is null)
  result := true;
else
  result := false;
```

1   I couldn't decide if I should upvote your answer or unclefofa's, since he seems to have answered first, but his answer has been edited after you
    answered. I ended upvoting both. – Zecc Mar 22 '11 at 10:04 ✏

    if listing.Offer_Text = ' ', it passes the NULLIF condition. I am sadfused. – Merritt Nov 9 '12 at 23:16

    As long as company.Offer_Text isn't null but then this will complicate things... =) – Coops Jan 3 '13 at 12:38

2   Shouldn't we use trim to make sure everything goes as planned – irfandar Aug 26 '13 at 11:00 ✏

5   @irfandar - Well, if you want to treat a string with all spaces as empty, go ahead use trim. Otherwise a string with all spaces is not empty. – Martin Ba
    Sep 2 '13 at 12:13

▲

50

▼

```
SELECT
    CASE WHEN LEN(listing.OfferText) > 0 THEN listing.OfferText
        ELSE COALESCE(Company.OfferText, '') END
    AS Offer_Text,

...
```

In this example, if `listing.OfferText` is NULL, the LEN() function should also return NULL, but that's still not > 0.

Update

This is similar to the accepted answer, but it also has a fallback in case `Company.OfferText` is also null. None of the other current answers using `NULLIF()` also do this.

edited Mar 18 '15 at 21:45   answered Dec 2 '08 at 14:34

Joel Coehoorn
318k   99   505   740

This got upvoted today, so it's indexed somewhere. I now know a better way to do this than the original answer. – Joel Coehoorn Mar 18 '15 at 21:46

34

```sql
Select
CASE
    WHEN listing.OfferText is null or listing.OfferText = '' THEN company.OfferText
    ELSE COALESCE(Company.OfferText, '')
END As Offer_Text,
from tbl_directorylisting listing
 Inner Join tbl_companymaster company
  On listing.company_id= company.company_id
```

answered Dec 2 '08 at 14:36

Patrick Harrington
33.3k   5   19   19

Here is another solution:

18

```sql
SELECT Isnull(Nullif(listing.offertext, ''), company.offertext) AS offer_text,
FROM   tbl_directorylisting listing
        INNER JOIN tbl_companymaster company
          ON listing.company_id = company.company_id
```
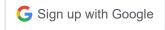
edited Jul 13 '10 at 8:30   answered Mar 24 '09 at 13:35

Join Stack Overflow to learn, share knowledge, and build your career.

| Sign up with email | G Sign up with Google | Sign up with Facebook | ✕ |

12

You can use `ISNULL` and check the answer against the known output:

```
SELECT case when ISNULL(col1, '') = '' then '' else col1 END AS COL1 FROM TEST
```

edited Mar 7 '12 at 15:26   answered Dec 2 '08 at 14:38

Adrian Carneiro   Code Trawler
46.1k   12   77   118   **263**   2   7

11

In SQL Server 2012 you have `IIF` , e.g you can use it like

```
SELECT IIF(field IS NULL, 1, 0) AS IsNull
```

The same way you can check if field is empty.

edited Mar 18 '16 at 16:54   answered Aug 1 '14 at 6:13

ΩmegaMan   lkurylo
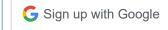**17.2k**   5   51   65   **1,099**   24   49

5

Use the LEN function to check for null or empty values. You can just use LEN(@SomeVarcharParm) > 0. This will return false if the value is NULL, '', or ' '. This is because LEN(NULL) returns NULL and NULL > 0 returns false. Also, LEN(' ') returns 0. See for yourself run:

```
SELECT
  CASE WHEN NULL > 0 THEN 'NULL > 0 = true' ELSE 'NULL > 0 = false' END,
  CASE WHEN LEN(NULL) > 0 THEN 'LEN(NULL) = true' ELSE 'LEN(NULL) = false' END,
  CASE WHEN LEN('') > 0 THEN 'LEN('''') > 0 = true' ELSE 'LEN('''') > 0 = false' END,
  CASE WHEN LEN(' ') > 0 THEN 'LEN('' '') > 0 = true' ELSE 'LEN('' '') > 0 = false' END,
  CASE WHEN LEN(' test ') > 0 THEN 'LEN('' test '') > 0 = true' ELSE 'LEN('' test '') > 0
= false' END
```

Join Stack Overflow to learn, share knowledge, and build your career.

When you say "... LEN(NULL) returns NULL and NULL > 0 returns false...", the true rule is that every test or comparaison with NULL return NULL ! – Didier68 Feb 17 '16 at 11:23

Right that is worth noting, but this works as a shortcut because the comparison makes the null coalesce to the correct boolean so this wouldn't work for the inverse comparison of LEN(NULL) = 0 when we want to return true for null or empty. – Zach Johnson Mar 22 '17 at 20:19

4

```
Select
Coalesce(NullIf(listing.OfferText, ''), NullIf(company.OfferText, ''), '') As
Offer_Text,
from tbl_directorylisting listing
 Inner Join tbl_companymaster company
  On listing.company_id= company.company_id
```

answered Sep 10 '13 at 5:52

Muhammad Sharjeel Ahsan
402 3 11 24

Plus one for the first answer (5 years later) to use both `NULLIF()` and coalesce to an empty string if `company.OfferText` is null. However, the 2nd `NULLIF()` call here serves no purpose, as if that value an empty string you're just going to coalesce back to an empty string anyway. – Joel Coehoorn Mar 18 '15 at 21:50

4

This simple combination of COALESCE and NULLIF should do the trick:

```
SELECT
   Coalesce(NULLIF(listing.OfferText, ''), company.OfferText) As Offer_Text
...
```

Note: Add another empty string as the last COALESCE argument if you want the statement to return an empty string instead of NULL if both values are NULL.

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email   G Sign up with Google   Sign up with Facebook   ✕

Here's a solution, but I don't know if it's the best....

3
```
Select
Coalesce(Case When Len(listing.Offer_Text) = 0 Then Null Else listing.Offer_Text End,
company.Offer_Text, '') As Offer_Text,
from tbl_directorylisting listing
 Inner Join tbl_companymaster company
  On listing.company_id= company.company_id
```

answered Dec 2 '08 at 14:39

digiguru
7,149   16   54   83

3
```
SELECT
    COALESCE(listing.OfferText, 'company.OfferText') AS Offer_Text,
FROM
    tbl_directorylisting listing
    INNER JOIN tbl_companymaster company ON listing.company_id= company.company_id
```

edited Jan 15 '13 at 23:23   answered Jan 15 '13 at 23:09

Bo Persson   Anoop Verma
79.8k   17   126   186   **1,265**   11   17

---

this syntax :

3
```
SELECT *
FROM tbl_directorylisting listing
WHERE (civilite_etudiant IS NULL)
```
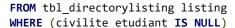
worked for me in Microsoft SQL Server 2008 (SP3)

Join Stack Overflow to learn, share knowledge, and build your career.

| Sign up with email | G Sign up with Google | Sign up with Facebook | ✕ |

I know this is an old thread but I just saw one of the earlier posts above and it is not correct.

3

If you are using **LEN(...)** to determine whether the field is **NULL or EMPTY** then you need to use it as follows:

```
...WHEN LEN(ISNULL(MyField, '')) < 1 THEN NewValue...
```

edited May 16 at 11:00   answered Aug 28 '17 at 18:02

Mansoor   Milan
147   10   **2,190**   1   25   37

To prevent the records with `Empty` or `Null` value in SQL result

2

we can simply add `.....` `WHERE Column_name != '' or 'null'`

edited Jul 28 '12 at 15:02   answered Jul 28 '12 at 11:02

Rndm   Code Crawler
3,201   7   32   56   **38**   5

> This would be the first port of call but if you are doing things like concatenating from multiple columns in a table for example, then this would exclude the row rather than just display a blank for that column – Coops Jan 3 '13 at 12:42

```
[Column_name] IS NULL OR LEN(RTRIM(LTRIM([Column_name]))) = 0
```

2

edited Jul 17 '17 at 21:42   answered Jul 17 '17 at 21:28

Paul Roub   user3829854
33.2k   8   61   76   **175**   2   3

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email   G Sign up with Google   Sign up with Facebook   ✕