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 to check if a column exists in a SQL Server table?

Asked 10 years, 11 months ago Active today Viewed 1.1m times

1760

326

I need to add a specific column if it does not exist. I have something like the following, but it always returns false:

```
IF EXISTS(SELECT *
          FROM   INFORMATION_SCHEMA.COLUMNS
          WHERE  TABLE_NAME = 'myTableName'
                 AND COLUMN_NAME = 'myColumnName')
```

How can I check if a column exists in a table of the SQL Server database?

sql-server sql-server-2008 tsql sql-server-2012 sql-server-2016

edited Jun 3 '18 at 11:12
Rann Lifshitz
3,356 4 16 37

asked Sep 25 '08 at 12:34
Maciej
9,124 3 17 23

12 I don't actually think there's anything wrong with the code in the question: Works finely for me in 2008 R2. (Maybe you were running it in the wrong database? Maybe your database was case-sensitive and you didn't have the case right in your myTableName / myColumnName strings? This type of query seems more flexible than the COL_LENGTH solution: I'm able to run it against a different database and even over a database link by suitably prefixing "INFORMATION_SCHEMA". Couldn't see how to do that with the COL_LENGTH metadata-function. – mwardm Jun 13 '13 at 14:57

3 @mwardm - `COL_LENGTH('AdventureWorks2012.HumanResources.Department ','ModifiedDate')` works fine. – Martin Smith Sep 12 '13 at 16:38

6 Little related hint: if you want to update a column right after column addition(I believe many users were searching this article for that purpose), you could use `EXEC sp_executesql` with formed `UPDATE` statement. – cassandrad Apr 16 '15 at 15:02

The real answer is you should add the database you are checking against so it's `FROM [YourDatabase].INFORMATION_SCHEMA.COLUMNS` – Alex Kwitny Jun 25 '15 at 22:35

20 Answers

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

1942

SQL Server 2005 onwards:

```
IF EXISTS(SELECT 1 FROM sys.columns
          WHERE Name = N'columnName'
          AND Object_ID = Object_ID(N'schemaName.tableName'))
BEGIN
    -- Column Exists
END
```

Martin Smith's version is shorter:

```
IF COL_LENGTH('schemaName.tableName', 'columnName') IS NOT NULL
BEGIN
    -- Column Exists
END
```

edited Apr 12 '17 at 3:35

answered Sep 25 '08 at 12:39

Mitch Wheat
263k 36 413 506

In Martin Smith's version, one thing to mention is not to include columnName within square brackets [ ]. When columnName is inside square brackets [ ], it'll give null even if the column exists in the table – Hemendra Jun 24 at 7:57

@HemendraSinghChauhan - that's because they aren't part of the name. You will also find that when comparing with the name in `sys.columns` – Martin Smith Jun 24 at 7:59

@MartinSmith didn't knew that, I was using your answer and came across this. Generally I use square brackets during adding columns, so I used them inside COL_LENGTH function too. My code was like this: `Alter table Table_Name Add [ColumnName] NVarchar(max) NULL; Select COL_LENGTH('[TABLE_NAME]', '[COLUMN_NAME]')` – Hemendra Jun 24 at 8:08

yes that isn't valid. The arguments to `COL_LENGTH` need to be unquoted. It is theoretically possible for someone to create a column that does actually have a name of `[COLUMN_NAME]` - e.g. `CREATE TABLE #T([[COLUMN_NAME]]] INT); SELECT * FROM #T` and then it would be ambiguous if this was not the rule. – Martin Smith Jun 24 at 8:12

A more concise version

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
END
```

The point about permissions on viewing metadata applies to all answers not just this one.

Note that the first parameter table name to `COL_LENGTH` can be in one, two, or three part name format as required.

An example referencing a table in a different database is

```
COL_LENGTH('AdventureWorks2012.HumanResources.Department','ModifiedDate')
```

One difference with this answer compared to using the metadata views is that metadata functions such as `COL_LENGTH` always only return data about committed changes irrespective of the isolation level in effect.

edited Oct 22 '18 at 7:19
Vadim Ovchinnikov
8,090 4 31 55

answered Mar 20 '11 at 14:47
Martin Smith
363k 63 613 717

11 This is less readable than some of the other answers, probably why it's not as highly rated. – Bill Yang Nov 30 '11 at 22:09

38 @Bill - Less readable in what way? Looks fine in Firefox. This answer was posted more than 2 years later than the accepted one, which explains the rating IMO. If you meant less clear that it is an existence check this type of idiom is quite common in SQL Server. e.g. using `IF OBJECT_ID('TableName','U') IS NULL` to check object existence or `DB_ID('foo')` to check database existence. – Martin Smith Nov 30 '11 at 22:31

58 @MartinSmith I'm sure he meant less readable because if you didn't know this idiom, and you inherited this code from someone else, you would not immediately understand what the code does. Kind of like writing `x>>2` instead of `x/4` in C++. The more verbose code (`if exists (select column_name from information_schema ...)`) takes a lot more space, but no one would ever scratch their head trying to figure out what it does. – Kip Aug 20 '13 at 16:49

20 Besides more concise this is a way faster solution. Accessing `INFORMATION_SCHEMA` views or `sys.columns` hits disk, while `COL_LENGTH` uses cached database metadata. – wqw Jan 13 '14 at 8:49

7 This is probably not the most highly rated answer because it was given 2.5 years after the other one. That's why I always check the dates when comparing the ratings on two answers. It takes a lot longer to overcome an answer that was given much earlier. ;) – Sean Feb 28 '14 at 19:35

Tweak the below to suit your specific requirements:

130

```
if not exists (select
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
            where
                  table_name = 'MyTable'
                  and column_name = 'MyColumn')
    alter table MyTable add MyColumn int
```

Edit to deal with edit to question: That should work - take a careful look over your code for stupid mistakes; are you querying INFORMATION_SCHEMA on the same database as your insert is being applied to for example? Do you have a typo in your table/column name in either statement?

edited Dec 31 '14 at 8:57

answered Sep 25 '08 at 12:35
Luke Bennett
29.3k 3 25 54

3 I just found out that adding TABLE_SCHEMA = 'mySchema' after where clause fixes the problem. – Maciej Sep 25 '08 at 17:01

11 -1: does not answer OP's question, only adds the new information on how to add a new collumn despite OP not asking about that at all, does not address OP's comment. – ANeves Nov 2 '11 at 11:46

72

Try this...

```
IF NOT EXISTS(
  SELECT TOP 1 1
  FROM INFORMATION_SCHEMA.COLUMNS
  WHERE
    [TABLE_NAME] = 'Employees'
    AND [COLUMN_NAME] = 'EmployeeID')
BEGIN
  ALTER TABLE [Employees]
    ADD [EmployeeID] INT NULL
END
```

edited Nov 14 '11 at 7:18
Soner Gönül
83.4k 27 161 293

answered Sep 25 '08 at 12:35
Leon Tayson
3,458 5 31 35

6 This method also works with SQL CE, whereas some of the other methods mentioned do not. – Sandra Walters Nov 14 '13 at 20:00

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

4 Within an `EXISTS` statement SQL automatically optimizes the columns away (much like `count(*)`) so `SELECT *` will suffice. – Marc L. Mar 1 '16 at 19:49

45

I'd prefer `INFORMATION_SCHEMA.COLUMNS` over a system table because Microsoft does not guarantee to preserve the system tables between versions. For example, `dbo.syscolumns` does still work in SQL 2008, but it's deprecated and could be removed at any time in future.

answered Jun 26 '09 at 8:58
Christian Hayter
27.3k 5 61 91

8 "INFORMATION_SCHEMA views could be incomplete since they are not updated for all new features" – onedaywhen Feb 26 '13 at 15:45

5 Well yes, that goes without saying since `INFORMATION_SCHEMA` views contain only ANSI-standard metadata. However, that is sufficient for an existence test. – Christian Hayter Feb 26 '13 at 17:15

3 Microsoft says "In future releases of SQL Server, Microsoft may augment the definition of any system catalog view by adding columns to the end of the column list. We recommend against using the syntax SELECT * FROM sys.catalog_view_name in production code because the number of columns returned might change and break your application." That implies that they won't remove columns or change their order. That's good enough backward compatibility for all but edge cases. – siride Jul 12 '13 at 21:26

42

You can use the information schema system views to find out pretty much anything about the tables you're interested in:

```
SELECT *
  FROM INFORMATION_SCHEMA.COLUMNS
 WHERE TABLE_NAME = 'yourTableName'
 ORDER BY ORDINAL_POSITION
```

You can also interrogate views, stored procedures and pretty much anything about the database using the Information_schema views.

answered Sep 25 '08 at 12:38
community wiki
anonymous

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

36 From **SQL Server 2016** you can use new DIE statements instead of big `IF` wrappers

```
ALTER TABLE Table_name DROP COLUMN IF EXISTS Column_name
```

edited Sep 5 at 4:08

answered May 1 '16 at 4:46
Pரதீப்
77.7k 12 88 126

3 Underrated answer in 2019 – ColinMac Sep 4 at 17:01

33 First check if the `table` / `column` (`id` / `name`) combination exists in `dbo.syscolumns` (an internal SQL Server table that contains field definitions), and if not issue the appropriate `ALTER TABLE` query to add it. For example:

```
IF NOT EXISTS ( SELECT  *
            FROM    syscolumns
            WHERE   id = OBJECT_ID('Client')
                    AND name = 'Name' )
ALTER TABLE Client
ADD Name VARCHAR(64) NULL
```

edited Jun 15 '15 at 12:29
shA.t
13.5k 4 39 77

answered Sep 25 '08 at 12:37
mdb
45.7k 9 61 62

31 Try something like:

```
CREATE FUNCTION ColumnExists(@TableName varchar(100), @ColumnName varchar(100))
RETURNS varchar(1) AS
BEGIN
DECLARE @Result varchar(1);
IF EXISTS (SELECT * FROM INFORMATION_SCHEMA.Columns WHERE TABLE_NAME = @TableName AND
COLUMN_NAME = @ColumnName)
BEGIN
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
BEGIN
    SET @Result = 'F'

END
RETURN @Result;
END
GO

GRANT EXECUTE ON  [ColumnExists] TO [whoever]
GO
```

Then use it like this:

```
IF ColumnExists('xxx', 'yyyy') = 'F'
BEGIN
  ALTER TABLE xxx
  ADD yyyyy varChar(10) NOT NULL
END
GO
```

It should work on both SQL Server 2000 & SQL Server 2005. Not sure about SQL Server 2008, but don't see why not.

edited Jun 15 '15 at 12:27
shA.t
13.5k 4 39 77

answered Sep 25 '08 at 12:38
Matt Lacey
60.5k 10 84 140

26

A good friend and colleague of mine showed me how you can also use an `IF` block with SQL functions `OBJECT_ID` and `COLUMNPROPERTY` in SQL SERVER 2005+ to check for a column. You can use something similar to the following:

You can see for yourself here

```
IF (OBJECT_ID(N'[dbo].[myTable]') IS NOT NULL AND
    COLUMNPROPERTY( OBJECT_ID(N'[dbo].[myTable]'), 'ThisColumnDoesNotExist', 'ColumnId')
IS NULL)
BEGIN
    SELECT 'Column does not exist -- You can add TSQL to add the column here'
END
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

1 And of course, if you are confident that the table exists, you can leave out the first part of the condition and check on `COLUMNPROPERTY` only. – Ruud Helderman Dec 12 '14 at 12:52

25

```
declare @myColumn   as nvarchar(128)
set @myColumn = 'myColumn'
if not exists (
    select  1
    from    information_schema.columns columns
    where   columns.table_catalog   = 'myDatabase'
        and columns.table_schema    = 'mySchema'
        and columns.table_name      = 'myTable'
        and columns.column_name     = @myColumn
    )
begin
    exec('alter table myDatabase.mySchema.myTable add'
    +'    ['+@myColumn+'] bigint       null')
end
```

answered Mar 3 '11 at 15:49

Tuomo Kämäräinen

251 3 2

21

Try this

```
SELECT COLUMNS.*
FROM   INFORMATION_SCHEMA.COLUMNS COLUMNS,
       INFORMATION_SCHEMA.TABLES TABLES
WHERE  COLUMNS.TABLE_NAME = TABLES.TABLE_NAME
       AND Upper(COLUMNS.COLUMN_NAME) = Upper('column_name')
```

edited Dec 21 '14 at 14:35

Pரதீப்

77.7k 12 88 126

answered Sep 30 '11 at 13:08

Douglas Tondo

211 2 2

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

column names in separate tables ;). – shA.t Jun 15 '15 at 12:34

21

This worked for me in SQL 2000:

```
IF EXISTS
(
    SELECT *
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE table_name = 'table_name'
    AND column_name = 'column_name'
)
BEGIN
...
END
```

edited Mar 23 '17 at 19:31

answered Aug 2 '11 at 19:54
Joe M
1,159 2 18 51

18

I needed similar for SQL SERVER 2000 and, as @Mitch points out, this only works inm 2005+.

Should it help anyone else, this is what worked for me in the end:

```
if exists (
    select *
    from
        sysobjects, syscolumns
    where
        sysobjects.id = syscolumns.id
        and sysobjects.name = 'table'
        and syscolumns.name = 'column')
```

edited Mar 16 '17 at 7:30
SuicideSheep
2,174 11 48 86

answered May 4 '12 at 14:14
FrostbiteXIII
698 7 21

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

15

```
if exists (
  select *
  from INFORMATION_SCHEMA.COLUMNS
  where TABLE_NAME = '<table_name>'
  and COLUMN_NAME = '<column_name>'
) begin
  print 'Column you have specified exists'
end else begin
  print 'Column does not exist'
end
```

edited Mar 7 at 6:06
Dale Burrell
4,870 5 28 57

answered May 8 '14 at 7:15
BYRAKUR SURESH BABU
161 1 2

12

```
IF NOT EXISTS( SELECT NULL
            FROM INFORMATION_SCHEMA.COLUMNS
           WHERE table_name = 'TableName'
             AND table_schema = 'SchemaName'
             AND column_name = 'ColumnName')  BEGIN

  ALTER TABLE [SchemaName].[TableName] ADD [ColumnName] int(1) NOT NULL default '0';

END;
```

edited 31 mins ago
DeanOC
5,622 6 33 50

answered Sep 12 '13 at 12:30
Na30m
156 3 10

2 I think you meant table_schema='schema_name'. – Tab Alleman Jul 28 '14 at 13:17

```
select distinct object_name(sc.id)
from syscolumns sc,sysobjects so
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

edited Dec 6 '15 at 15:27 answered May 23 '15 at 10:25

Uwe Keim
28.3k 33 142 228

Nishad
322 6 18

10

A temp table version of the accepted answer:

```
if (exists(select 1
             from tempdb.sys.columns
            where Name = 'columnName'
              and Object_ID = object_id('tempdb..#tableName')))
begin
...
end
```

edited May 23 '17 at 11:55

Community ♦
1 1

answered Jan 7 '15 at 23:56

crokusek
3,460 1 30 49

1 How is that different from the accepted answer? Would a temp table not work in the accepted answer? – John Saunders Jan 8 '15 at 1:24

1 Correct. The accepted answer does not work for temp tables because 'sys.columns' must be specified as 'tempdb.sys.columns' and the table name must be preceeded by 'tempdb..'. – crokusek Jan 8 '15 at 20:08

7

Wheat's answer is good, but assumes you do not have any identical table name / column name pairs in any schema or database. To make it safe for that condition use this...

```
select *
from Information_Schema.Columns
where Table_Catalog = 'DatabaseName'
  and Table_Schema = 'SchemaName'
  and Table_Name = 'TableName'
  and Column_Name = 'ColumnName'
```

answered Mar 26 '15 at 17:42

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

7

There are several ways to check the existence of a column. I would strongly recommend to use `INFORMATION_SCHEMA.COLUMNS` as it is created in order to communicate with user. Consider following tables:

```
sys.objects
sys.columns
```

and even some other access methods available to check `system catalog.`

Also, no need to use `SELECT *` , simply test it by `NULL value`

```
IF EXISTS(
           SELECT NULL
           FROM INFORMATION_SCHEMA.COLUMNS
           WHERE
             TABLE_NAME = 'myTableName'
             AND COLUMN_NAME = 'myColumnName'
         )
```

edited Apr 28 '18 at 14:23
user9713753

answered Feb 15 '16 at 20:45
Ali Elmi
221 3 8

1 No matter even if you `SELECT *` with `EXISTS` , because when exists is used it doesn't really select all the rows and all the columns, internally it just checks for the existence and not actually checks for all rows and columns – Pawan Nogariya Nov 23 '18 at 8:55

6

One of the most simple and understandable solution is:

```
IF COL_LENGTH('Table_Name','Column_Name') IS NULL
 BEGIN
    -- Column Not Exists, implement your logic
 END
ELSE
 BEGIN
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

answered May 24 '17 at 7:17

Arsman Ahmad
984 11 19

6

Here is a simple script I use to manage addition of columns in the database:

```
IF NOT EXISTS (
        SELECT *
        FROM sys.Columns
        WHERE Name = N'QbId'
            AND Object_Id = Object_Id(N'Driver')
        )
BEGIN
    ALTER TABLE Driver ADD QbId NVARCHAR(20) NULL
END
ELSE
BEGIN
    PRINT 'QbId is already added on Driver'
END
```

In this example, the `Name` is the `ColumnName` to be added and `Object_Id` is the `TableName`

edited Jan 28 '18 at 4:41

Brien Foss
2,611 3 13 27

answered Nov 16 '16 at 12:21

UJS
680 1 7 16

1 Perfect answer because generally we need to check column only if we have a corresponding alter table command :) – vibs2006 Jun 1 '18 at 7:14

4

Below query can be used to check whether searched column exists or not in the table. We can take decision based on the searched result also as shown below.

```
IF EXISTS (SELECT 'Y' FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = <YourTableName>
AND COLUMN_NAME = <YourColumnName>)
  BEGIN
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```
  BEGIN


    ALTER TABLE <YourTableName> ADD <YourColumnName> <DataType>[Size]
  END
```

answered Oct 30 '18 at 14:31

Suraj Kumar
2,823 5 11 26

3

Yet another variation...

```
SELECT
  Count(*) AS existFlag
FROM
  sys.columns
WHERE
  [name] = N 'ColumnName'
  AND [object_id] = OBJECT_ID(N 'TableName')
```

edited Mar 7 at 6:05

Bikramjeet Singh
407 1 4 13

answered Jul 10 '14 at 11:01

Manuel Alves
2,180 19 16

1

table -->script table as -->new windows - you have design script. check and find column name in new windows

answered May 10 at 15:56

arnav
1,854 15 17

Execute the below query to check if the column exists in the given table:

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

answered May 29 at 13:43

S Krishna

323 3 7

0

```
IF EXISTS(SELECT 1 FROM sys.columns
      WHERE Name = N'columnName'
      AND Object_ID = Object_ID(N'schemaName.tableName'))
```

> This should be the fairly easier way and straight forward solution to this problem. I have used this multiple times for similar scenarios. It works like a charm, no doubts on that.

answered Apr 3 at 9:45

Ilangeeran

51 5

0

```
IF EXISTS (
SELECT *
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_CATALOG = 'Database Name'
and TABLE_SCHEMA = 'Schema Name'
and TABLE_NAME = 'Table Name'
and COLUMN_NAME = 'Column Name'
and DATA_TYPE = 'Column Type') -- Where statement lines can be deleted.

BEGIN
--COLUMN EXISTS IN TABLE
END

ELSE BEGIN
--COLUMN DOES NOT EXISTS IN TABLE
END
```

answered Jul 8 at 10:58

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

0

Another contribute is the following sample that add the column if not exist.

```
USE [Northwind]
GO

IF NOT EXISTS(SELECT * FROM INFORMATION_SCHEMA.COLUMNS
                WHERE TABLE_NAME = 'Categories'
                    AND COLUMN_NAME = 'Note')
BEGIN

ALTER TABLE Categories ADD Note NVARCHAR(800) NULL

END
GO
```

Hope it helps. Simone

answered Jul 11 at 23:18

Simone Spagna
123 8

protected by bummi Oct 13 '14 at 11:02

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.