SQL exclude a column using SELECT * [except columnA] FROM tableA?

623

88

We all know that to select all columns from a table, we can use

```
SELECT * FROM tableA
```

Is there a way to exclude column(s) from a table without specifying all the columns?

```
SELECT * [except columnA] FROM tableA
```

The only way that I know is to manually specify all the columns and exclude the unwanted column. This is really time consuming so I'm looking for ways to save time and effort on this, as well as future maintenance should the table has more/less columns.

thanks!

sql sql-server tsql

edited Apr 5 '18 at 8:49
a_horse_with_no_name
316k 48 482 588

asked Apr 8 '09 at 9:15
uuɐwəʃəs

16 It would be very convenient to have this feature, not to put in production code, but for troubleshooting purposes. Example: I have a table that has several columns I query, but I want to quickly omit a text column or two. – Micah B. Jul 10 '13 at 16:18

I had a need for this when working with openquery (though I neede the functionality in MySQL rather than SQL Server). I had to query an MySQL database using SQL Server. Because a MySQL table had fixed width char columns, I couldn't use a `SELECT *` query (OLE DB has issues mapping those). I couldn't specify the right columns because I had no direct access to the MySQL database, however SQL Server was kind enough to inform me of the names of the fixed width char columns... – jahu Dec 4 '14 at 9:26

2 possible duplicate of SELECT * EXCEPT – Glen Solsberry Apr 1 '15 at 23:32

I'd like to add another reason to do this: `SELECT DISTINCT *` except for the key column to work without duplicate rows someone else created

36 Answers

1 2 next

395

I agree with everyone... but if I was going to do something like this I might do it this way:

```
/* Get the data into a temp table */
SELECT * INTO #TempTable
FROM YourTable
/* Drop the columns that are not needed */
ALTER TABLE #TempTable
DROP COLUMN ColumnToDrop
/* Get results and drop temp table */
SELECT * FROM #TempTable
DROP TABLE #TempTable
```

edited Nov 17 '16 at 21:42

radtek
17.1k 6 91 80

answered Nov 11 '09 at 0:35

Norman Skinner
4,502 1 14 18

185 Inefficient...but very creative :) – Guillermo Gutiérrez Nov 15 '11 at 20:26

1 Beautiful. I often need to include join two temp tables or a temp to another table where I don't need all the columns in the temp - especially because grouping will be involved. – VISQL Oct 9 '12 at 21:41

2 Very nice. Sure solves the problem of abstracting out the column names. – Toaster Feb 20 '13 at 12:44

1 @CeesTimmerman - if you have a query involving duplicate column names, that is a separate problem, REGARDLESS of which approach you are taking. Google "SQL column rename" or "SQL column alias". Something like `SELECT table1.ID AS table1ID ...` , IIRC. – ToolmakerSteve Aug 21 '14 at 2:44

3 @ToolmakerSteve The idea of this question is to only specify columns you don't need. Naming columns would require specifying all, say, 20+ columns of a particular table. – Cees Timmerman Aug 21 '14 at 8:04

- This makes your contract between client and database stable. Same data, every time
- Performance, covering indexes

Edit (July 2011):

If you drag from Object Explorer the `Columns` node for a table, it puts a CSV list of columns in the Query Window for you which achieves one of your goals

edited Mar 6 '12 at 10:03

demoncodemonkey
8,878 6 48 89

answered Apr 8 '09 at 9:19

gbn
352k 60 495 586

1 i never knew the thing about dragging columsn thank you GBN you've again saved the day. you must make a lot of $$ due to your sql expertise – |--"""---------""""""""" Oct 21 '12 at 18:26

2 there are valid scenarios with SELECT *, especially in ETL routines. I think the best answer here is the one with dynamic SQL. – mishkin Feb 5 '14 at 20:16

1 There are cases where you want to select say all data for a student for statistical exploration but not bring down the wire the student id itself to bolster privacy – George Birbilis Jul 11 '14 at 16:59

1 if for nothing else, i am glad i read this post to find out about the CSV trick. that is why giving background and info around the subject is important than just straight answers. +1 from me for that. thanks – Andrei Bazanov Jun 21 '16 at 9:06

2 "specify only the required columns" - hard coding nonsence – RM Haley Apr 30 '18 at 17:35

63

The automated way to do this in SQL (SQL Server) is:

```
declare @cols varchar(max), @query varchar(max);
SELECT  @cols = STUFF
    (
        (
            SELECT DISTINCT '], [' + name
            FROM sys.columns
            where object_id = (
```

```
    ) + ']';

SELECT @query = 'select ' + @cols + ' from MyTable';
EXEC (@query);
```

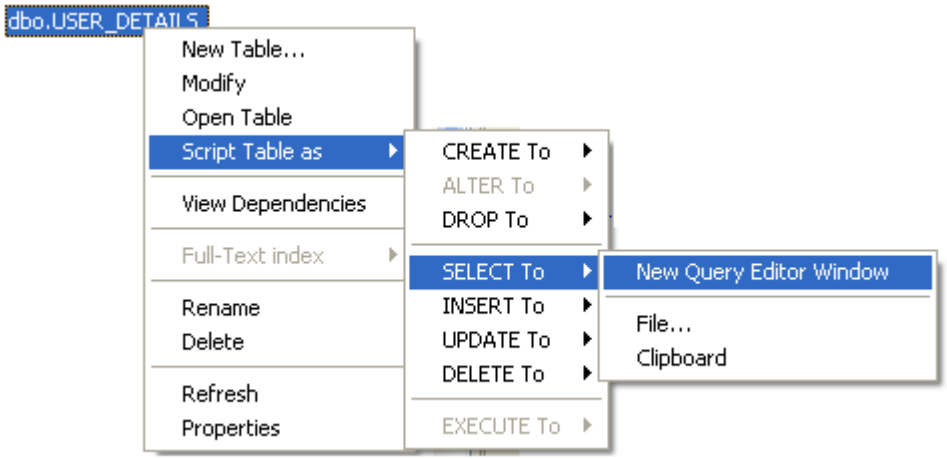
edited Dec 29 '15 at 17:13 — Community ♦ — 1 1

answered Jul 14 '11 at 10:19 — pl80 — 628 5 5

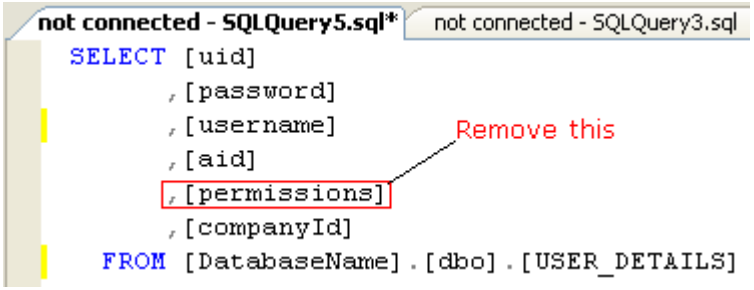
What if you are querying from the result of a CTE or other subquery? A simple example: you might want to make a subquery that appends the result of `row_number()` to each row, then perform a join by `row_number`, then select everything *excluding the* `row_number` from the result of the join. – ely Mar 18 '16 at 5:38

58

If you don't want to write each column name manually you can use `Script Table As` by right clicking on *table* or *view* in *SSMS* like this:



Then you will get whole select query in *New Query Editor Window* then remove unwanted column like this:



Done

edited Aug 2 '12 at 4:44

answered Jul 13 '12 at 7:03

hims056

23.5k 23 93 120

2 This is more easy than selecting all column names then add commas to it. – stom Dec 12 '16 at 7:35

35

You could create a view that has the columns you wish to select, then you can just select * from the view...

answered Apr 8 '09 at 9:33

campo

999 1 11 19

5 At first my reaction was "how is this easier than just specifying the columns"? But then I decided that from a maintenance perspective, it might be an improvement. – ToolmakerSteve Aug 21 '14 at 2:15

1 This also supports the query optimiser rather than counteracting it with a temp table or dynamic SQL. – underscore_d Jun 26 '16 at 7:26

Yes it's possible (but not recommended).

```
SELECT @columns = ISNULL(@columns + ', ','') + QUOTENAME(column_name)
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'contact' AND COLUMN_NAME <> 'dob'
ORDER BY ORDINAL_POSITION

EXEC ('SELECT ' + @columns + ' FROM contact')
```

Explanation of the code:

1. Declare a variable to store a comma separated list of column names. This defaults to NULL.
2. Use a system view to determine the names of the columns in our table.
3. Use `SELECT @variable = @variable + ... FROM` to concatenate the column names. This type of `SELECT` does not not return a result set. This is perhaps undocumented behaviour but works in every version of SQL Server. As an alternative you could use `SET @variable = (SELECT ... FOR XML PATH(''))` to concatenate strings.
4. Use the `ISNULL` function to prepend a comma only if this is not the first column name. Use the `QUOTENAME` function to support spaces and punctuation in column names.
5. Use the `WHERE` clause to hide columns we don't want to see.
6. Use `EXEC (@variable)` , also known as *dynamic SQL*, to resolve the column names at runtime. This is needed because we don't know the column names at compile time.

edited Mar 29 '16 at 9:30

answered Apr 13 '10 at 9:58

Anthony Faull

13.7k 5 37 65

Thanks! This was the key to solve my problem. – Antonio Sep 9 '15 at 15:05

I have one question: for which reason the second select has to be put into an EXEC statement? I have seen that it is effectively necessary, but I wonder why I cannot simply write `SELECT @columns FROM contact` – Antonio Sep 9 '15 at 16:10

Like the others have said there is no way to do this, but if you're using Sql Server a trick that I use is to change the output to comma separated, then do

and cut the whole list of columns from the output window. Then you can choose which columns you want without having to type them all in.

edited Jul 6 '10 at 20:15
Tom H
41.4k 10 70 116

answered Apr 8 '09 at 9:21
MrTelly
13.2k 39 76

11 See my tip about dragging from SSMS – gbn Nov 15 '11 at 20:38

11

Basically, you cannot do what you would like - but you can get the right tools to help you out making things a bit easier.

If you look at Red-Gate's SQL Prompt, you can type "SELECT * FROM MyTable", and then move the cursor back after the "*", and hit <TAB> to expand the list of fields, and remove those few fields you don't need.

It's not a perfect solution - but a darn good one! :-) Too bad MS SQL Server Management Studio's Intellisense still isn't intelligent enough to offer this feature.....

Marc

answered Apr 8 '09 at 12:37
marc_s
592k 133 1134 1279

This is good, but the problem is your query can become huge. It would be nice to have the "except" feature, not for prod code, but ad-hoc querying. – Micah B. Jul 10 '13 at 16:20

7

no there is no way to do this. maybe you can create custom views if that's feasible in your situation

EDIT May be if your DB supports execution of dynamic sql u could write an SP and pass the columns u don't want to see to it and let it

12 It is doable but I would fire the person doing that. – Lieven Keersmaekers Apr 8 '09 at 9:29

Classic cleanup problem: Copy the dfata you want to keep into a temp table, truncate the big one, re-fill with the temp table. You need to exclude the identity column. – Volker Mar 9 '18 at 23:06

7

In SQL Management Studio you can expand the columns in Object Explorer, then drag the `Columns` tree item into a query window to get a comma separated list of columns.

answered May 19 '10 at 7:55

cjk
39.8k 6 70 100

7

If you are using SQL Server Management Studio then do as follows:

1. Type in your desired tables name and select it
2. Press Alt + F1
3. o/p shows the columns in table.
4. Select the desired columns
5. Copy & paste those in your select query
6. Fire the query.

Enjoy.

edited Oct 12 '15 at 5:42

shA.t
13.3k 4 39 75

answered Aug 30 '11 at 12:16

asdasdasd
79 1 1

answered May 6 '10 at 19:03

Shrage Smilowitz
14.6k 2 22 31

Can you use some variation of "innername.*" to represent the inner columns, similar to "SELECT table1.* ..." when doing a join? – ToolmakerSteve Aug 21 '14 at 2:17

6

```
DECLARE @SQL VARCHAR(max), @TableName sysname = 'YourTableName'

SELECT @SQL = COALESCE(@SQL + ', ', '') + Name
FROM sys.columns
WHERE OBJECT_ID = OBJECT_ID(@TableName)
AND name NOT IN ('Not This', 'Or that');

SELECT @SQL = 'SELECT ' + @SQL + ' FROM ' + @TableName

EXEC (@SQL)
```

UPDATE:

You can also create a stored procedure to take care of this task if you use it more often. In this example I have used the built in STRING_SPLIT() which is available on SQL Server 2016+, but if you need there are pleanty of examples of how to create it manually on SO.

```
CREATE PROCEDURE [usp_select_without]
@schema_name sysname = N'dbo',
@table_name sysname,
@list_of_columns_excluded nvarchar(max),
@separator nchar(1) = N','
AS
BEGIN
 DECLARE
 @SQL nvarchar(max),
 @full_table_name nvarchar(max) = CONCAT(@schema_name, N'.', @table_name);
```

```
 SELECT @SQL = N'SELECT ' + @SQL + N' FROM ' + @full_table_name;
 EXEC(@SQL)
END
```

And then just:

```
EXEC [usp_select_without]
@table_name = N'Test_Table',
@list_of_columns_excluded = N'ID, Date, Name';
```

edited Oct 9 '18 at 10:35

answered Oct 16 '17 at 15:58

Bartosz X

1,267 15 22

5

If we are talking of Procedures, it works with this trick to generate a new query and **EXECUTE IMMEDIATE** it:

```
SELECT LISTAGG((column_name), ', ') WITHIN GROUP (ORDER BY column_id)
INTO var_list_of_columns
FROM ALL_TAB_COLUMNS
WHERE table_name = 'PUT_HERE_YOUR_TABLE'
AND column_name NOT IN ('dont_want_this_column','neither_this_one','etc_column');
```

edited Mar 26 '15 at 9:07

Nadeem_MK

6,156 7 35 51

answered May 21 '14 at 13:49

AlexandreWL

51 1 1

5

> Is there a way to exclude column(s) from a table without specifying all the columns?

Using declarative SQL in the usual way, no.

However, SQL's `SELECT *` already gets a lot a flak (@Guffa's answer here is a typical objection), so I don't think `SELECT ALL BUT` will get into the SQL Standard anytime soon.

I think the best 'work around' is to create a `VIEW` with only the columns you desire then `SELECT * FROM ThatView` .

edited Jun 27 '16 at 12:18

answered Mar 24 '11 at 12:48

onedaywhen
44.7k 10 79 124

1 @underscore_d: I've now revised my answer. – onedaywhen Jun 27 '16 at 12:53

Cool, I agree with your last 2 paragraphs. The info about Tutorial D is interesting, although I tend to agree with those who think `select *` is questionable - very useful for ad hoc stuff and programs that need to generically handle data tables, but not so much for building (in lack of a better word) 'pure' queries. Still, not being in the ANSI standard doesn't mean Microsoft can't add it to their dialect, as with many other things, but I'm sceptical whether they would. – underscore_d Jun 27 '16 at 13:40

3

I do not know of any database that supports this (SQL Server, MySQL, Oracle, PostgreSQL). It is definitely not part of the SQL standards so I think you have to specify only the columns you want.

You could of course build your SQL statement dynamically and have the server execute it. But this opens up the possibility for SQL injection..

answered Apr 8 '09 at 9:19

Ronald Wildenberg
27.1k 10 74 119

3

I know this is a little old, but I had just run into the same issue and was looking for an answer. Then I had a senior developer show me a very simple trick.

If you are using the management studio query editor, expand the database, then expand the table that you are selecting from so that you can see the columns folder.

imz -- Ivan Zakharyaschev
2,920 5 33 85

anonymous
31 1

1 Yes, but what if you have 5 joins, the idea would be to do `SELECT * Except(tableName.ColumnName) FROM ...` – Pawel Cioch Sep 30 '14 at 19:54

I find this useful :) I haven't known about that, nevertheless this is not an answer for topic question. – Fka Feb 10 '15 at 11:40

2

Well, it is a common best practice to specify which columns you want, instead of just specifying *. So you should just state which fields you want your select to return.

answered Apr 8 '09 at 9:24

Gustavo
301 2 5

2

A colleage advised a good alternative:

- Do SELECT INTO in your preceding query (where you generate or get the data from) into a table (which you will delete when done). This will create the structure for you.
- Do a script as CREATE to new query window.
- Remove the unwanted columns. Format the remaining columns into a 1 liner and paste as your column list.
- Delete the table you created.

Done...

This helped us a lot.

answered Feb 2 '12 at 13:02

Charl
832 5 12

2

pls refer: http://www.postgresonline.com/journal/archives/41-How-to-SELECT-ALL-EXCEPT-some-columns-in-a-table.html

The Information Schema Hack Way

```
SELECT 'SELECT ' || array_to_string(ARRAY(SELECT 'o' || '.' || c.column_name
        FROM information_schema.columns As c
            WHERE table_name = 'officepark'
            AND  c.column_name NOT IN('officeparkid', 'contractor')
    ), ',') || ' FROM officepark As o' As sqlstmt
```

The above for my particular example table - generates an sql statement that looks like this

SELECT o.officepark,o.owner,o.squarefootage FROM officepark As o

edited Feb 15 '16 at 12:01

answered Feb 15 '16 at 11:39

user3393089

69 1 9

Whilst this may theoretically answer the question, it would be preferable to include the essential parts of the answer here, and provide the link for reference. – Bhargav Rao ♦ Feb 15 '16 at 11:42

The best way to solve this is using view you can create view with required columns and retrieve data form it

2

example

```
mysql> SELECT * FROM calls;
+----+------------+---------+
| id | date       | user_id |
+----+------------+---------+
|  1 | 2016-06-22 |       1 |
|  2 | 2016-06-22 |    NULL |
|  3 | 2016-06-22 |    NULL |
|  4 | 2016-06-23 |       2 |
|  5 | 2016-06-23 |       1 |
```

```
    ->     SELECT id,date from calls;
Query OK, 0 rows affected (0.20 sec)

mysql> select * from C_VIEW;
+----+------------+
| id | date       |
+----+------------+
|  1 | 2016-06-22 |
|  2 | 2016-06-22 |
|  3 | 2016-06-22 |
|  4 | 2016-06-23 |
|  5 | 2016-06-23 |
|  6 | 2016-06-23 |
|  7 | 2016-06-23 |
+----+------------+
7 rows in set (0.00 sec)
```

answered Jun 30 '16 at 12:37

Mahesh Madushanka
2,494 1 6 22

2 What if the number of columns is large, say 100 & we wanted to SELECT all columns, but one. Is there a better approach? – KartikKannapur Oct 25 '16 at 16:22

3 This one kind of misses the point. You are selecting "id,date" in the second call, which, if you were going to do that, just do it in the first place. – keithpjolley May 20 '17 at 15:02

1

Right click table in Object Explorer, Select top 1000 rows

It'll list all columns and not *. Then remove the unwanted column(s). Should be much faster than typing it yourself.

Then when you feel this is a bit too much work, get Red Gate's SQL Prompt, and type ssf from tbl, go to the * and click tab again.

answered Mar 24 '11 at 12:57

cairnz
3,626 1 13 19

1

```
declare @colnames varchar(max)=''

select @colnames=@colnames+','+name from syscolumns where object_id(tablename)=id and
name not in (column3,column4)

SET @colnames=RIGHT(@colnames,LEN(@colnames)-1)
```

`@colnames` looks like `column1,column2,column5`

edited Mar 3 '15 at 12:37
Egor Shkorov
1,668 1 16 22

answered Mar 3 '15 at 11:52
Valeriy
11 1

1

Sometimes the same program must handle different database stuctures. So I could not use a column list in the program to avoid errors in `select` statements.

`*` gives me all the optional fields. I check if the fields exist in the data table before use. This is my reason for using `*` in `select` .

This is how I handle excluded fields:

```
Dim da As New SqlDataAdapter("select * from table", cn)
da.FillSchema(dt, SchemaType.Source)
Dim fieldlist As String = ""
For Each DC As DataColumn In DT.Columns
   If DC.ColumnName.ToLower <> excludefield Then
    fieldlist = fieldlist &  DC.Columnname & ","
   End If
  Next
```

edited Oct 1 '15 at 10:39
spenibus
3,764 11 19 30

answered Oct 1 '15 at 10:19
Dusan
11 1

0

```
sp_help <table_name>
```

-Click on the 'Column_name' column> Copy> Paste (creates a vertical list) into a New Query window and just type commas in front of each column value... comment out the columns you don't want... far less typing than any code offered here and still manageable.

answered May 12 '11 at 15:43

plo
9 1

"far less typing than any code offered here" and far more hassle than dragging the Columns node into the query editor or any of the other superior ways of getting the list. I mean what about tables with many 10s of columns? – underscore_d Jun 26 '16 at 7:34

0

You can get SQL Complete from devart.com, which not just expands the * wildcard just like SQL Prompt from Red Gate does (as described in cairnz's answer), but also provides a column picker drop down with checkboxes in which you can check all the columns that you want in the select list and they will be inserted automatically for you (and if you then uncheck a column it will be automatically removed from the select list).

edited Mar 27 '12 at 0:37

answered Mar 27 '12 at 0:21

yoel halb
8,785 3 40 41

0

In SSMS there is an easier way with **IntelliSense** and **Aliasing**. Try this

1. Right-Click in the text editor and make sure **IntelliSense** is enabled.
2. Type the query with an alias [SELECT **t.*** FROM tablename t].
3. Go the text **t.*** and delete the * ,and SSMS will auto-list the columns of the **f** aliased table.

You can then quickly specify **only the columns you want** w/o having to use SSMS to write a select to another script and then do more copy/paste operations. I use this all the time.

Could You specify Your answer to column exclusion, please? – Kamiccolo Mar 19 '14 at 15:41

@Kamiccolo - what is being described by DuckWork is a MANUAL action. Cutting and pasting the desired column names. He's merely saying this is a way to more easily get at the names, without a lot of typing. It doesn't help you write a query that says "exclude this column". It just helps you create the desired list of columns, which you then paste into your query. – ToolmakerSteve Aug 21 '14 at 2:34

0

This won't save time on loading from the database. But, you could always unset the column you don't want in the array it's placed in. I had several columns in a table but didn't want one particular. I was too lazy to write them all out in the SELECT statement.

```
$i=0;
$row_array = array();

while($row = mysqli_fetch_assoc($result)){

  $row_array[$i]=$row;
  unset($row_array[$i]['col_name']);
  $i++;
}
```

answered Aug 3 '14 at 19:36

user3904583
17 1

Thanks - that is a useful idea, in my situation. – ToolmakerSteve Aug 21 '14 at 2:35

Very poor idea, you still select more than you need which is wasteful od database resources. And it is simply a bad idea not to select specific columns in the query for production except in some edge cases. You are recommending the use of a SQL antipattern – HLGEM Apr 29 '15 at 14:49

an SQL antipattern but a code pattern. The alternative is pretty much to select everything but manually. – evandentremont Jul 15 '15 at 18:20

I did it like this and it works just fine (version 5.5.41):

```
# set sql command using prepared columns
SET @sql = CONCAT("SELECT ", @dyn_colums, " FROM table_name");

# prepare and execute
PREPARE statement FROM @sql;
EXECUTE statement;
```

edited May 2 '15 at 10:24

answered May 2 '15 at 10:17

Vladimir Djuricic

3,061 1 10 15

0

If anyone here is using MySql like I was use this:

```
CREATE TABLE TempTable AS SELECT * FROM #YourTable;

ALTER TABLE TempTable
DROP COLUMN #YourColumn;

SELECT * FROM TempTable;
DROP TABLE TempTable;
```

edited Feb 13 '16 at 23:16

Bartłomiej Semańczyk

35.5k 27 173 247

answered Feb 13 '16 at 22:58

Felix

31 1

The question is specifically about SQL Server, but I'll withhold the downvote since the same syntax will work for both. – underscore_d Jun 26 '16 at 7:38

Creating a whole table (not even temporary) just to lose a column seems wasteful. – Reuben Thompson Sep 8 '16 at 12:42

I think the better way to is to create a permanent view instead of a table – Alexander Mills Feb 16 at 3:28

1 2 next