SQL SELECT WHERE field contains words



I need a select which would return results like this:

431

SELECT * FROM MyTable WHERE Column1 CONTAINS 'word1 word2 word3'



And I need all results, i.e. this includes strings with 'word2 word3 word1' or 'word1 word3 word2' or any other combination of the three.



All words need to be in the result.

69

sql select



asked Jan 12 '13 at 6:19



15 Answers



Rather slow, but working method to include any of words:



SELECT * FROM mytable
WHERE column1 LIKE '%word1%'
 OR column1 LIKE '%word2%'
 OR column1 LIKE '%word3%'



If you need **all** words to be present, use this:

SELECT * **FROM** mytable

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







edited Jan 12 '13 at 6:27

answered Jan 12 '13 at 6:21



- + 1 I agree it's slower but it can be mitigated with good indexing Preet Sangha Jan 12 '13 at 6:22 🖍
- @PreetSangha Indexing when you're searching for LIKE beginning with a wild card? Please show me how! Popnoodles Jan 12 '13 at 6:24 🖍
- In PostgreSQL 9.1 and later, you can create trigram index which can index such searches. myp Oct 20 '14 at 16:26
- @AquaAlex: your statement will fail if text has word3 word2 word1 . mvp Mar 30 '16 at 21:24
- Another downside of this approach: '%word%' will also find 'words', 'crosswordpuzzle' and 'sword' (just as an example). I'd have to do a column1 LIKE 'word' OR column1 LIKE 'word %' OR column1 LIKE '% word' OR column1 LIKE ' word ' to just find exact word matches - and it would still fail for entries where words are not just separated with spaces. - BlaM Apr 18 '17 at 15:51



Note that if you use LIKE to determine if a string is a substring of another string, you must escape the pattern matching characters in your search string.

If your SQL dialect supports CHARINDEX, it's a lot easier to use it instead:



```
SELECT * FROM MyTable
WHERE CHARINDEX('word1', Column1) > 0
 AND CHARINDEX('word2', Column1) > 0
 AND CHARINDEX('word3', Column1) > 0
```

Also, please keep in mind that this and the method in the accepted answer only cover substring matching rather than word matching. So, for example, the string 'word1word2word3' would still match.

edited Sep 5 '14 at 0:38

answered Sep 5 '14 at 0:21



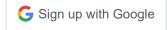
24.1k 20

127

This seems much easier if your search term is a variable rather than having to add the 1% chars before searching. ShapeRlake Nov. 14 14 at 20:30

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

Sign up with email







Function

16

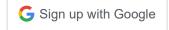
Query

```
INSERT INTO @FilterTable (Data)
SELECT DISTINCT S.Data
FROM fnSplit(' ', 'word1 word2 word3') S -- Contains words

SELECT DISTINCT
        T.*
FROM
        MyTable T
        INNER JOIN @FilterTable F1 ON T.Column1 LIKE '%' + F1.Data + '%'
        LEFT JOIN @FilterTable F2 ON T.Column1 NOT LIKE '%' + F2.Data + '%'
WHERE
F2 Data IS NULL
```

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

Sign up with email





Exellent! How to start to learn about this function, Sir? what is Pieces? and can You tell me pseudocode about this line? SUBSTRING(@str, start, CASE WHEN stop > 0 THEN stop - start ELSE 512 END) AS Data - Khaneddy2013 Feb 25 '16 at 3:27



Instead of SELECT * FROM MyTable WHERE Column1 CONTAINS 'word1 word2 word3', add And in between those words like:

10

SELECT * FROM MyTable WHERE Column1 CONTAINS 'word1 And word2 And word3'



for details, see here https://msdn.microsoft.com/en-us/library/ms187787.aspx

UPDATE

For selecting phrases, use double quotes like:

```
SELECT * FROM MyTable WHERE Column1 CONTAINS '"Phrase one" And word2 And "Phrase Two"'
```

p.s. you have to first enable Full Text Search on the table before using contains keyword. for more details, See here https://docs.microsoft.com/en-us/sql/relational-databases/search/get-started-with-full-text-search

edited Mar 20 '18 at 12:22

answered Jul 26 '16 at 16:42





SELECT * FROM MyTable WHERE
Column1 LIKE '%word1%'
AND Column1 LIKE '%word2%'
AND Column1 LIKE '%word3%'



Changed or to AND based on edit to question.

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

Sign up with email





I need all words to be contained in the result in any combination - Mario M Jan 12 '13 at 6:25



If you are using **Oracle Database** then you can achieve this using <u>contains</u> query. Contains querys are faster than like query.

5

If you need all of the words



SELECT * FROM MyTable WHERE CONTAINS(Column1, 'word1 and word2 and word3', 1) > 0

If you need any of the words

SELECT * FROM MyTable WHERE CONTAINS(Column1, 'word1 or word2 or word3', 1) > 0

Contains need index of type **CONTEXT** on your column.

CREATE INDEX SEARCH IDX ON MyTable(Column) INDEXTYPE IS CTXSYS.CONTEXT

answered Jun 29 '15 at 12:15



mirmdasif

3.684 1 17 23

- 1 @downvoters A comment is appreciated telling what is wrong with the answer. This same query is running in our enterprise solution more than 1000 times per day, without any issues:) mirmdasif Nov 12 '17 at 6:43
- OP does not specify which database is using and everyone has assumed that is Sql Server. But since you have specified Oracle in your response I don't understand downvoters. EAmez Feb 1 at 11:00



If you just want to find a match.

4

SELECT * FROM MyTable WHERE INSTR('word1 word2 word3',Column1)<>0

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

Sign up with email



To get exact match. Example (';a;ab;ac;',';b;') will not get a match.

```
SELECT * FROM MyTable WHERE INSTR(';word1;word2;word3;',';'||Column1||';')<>0
```

edited Feb 9 '16 at 15:08



answered Nov 11 '15 at 20:32



'INSTR' is not a recognized built-in function name. In my SQL Server. – Durgesh Pandey Sep 12 '17 at 5:21



try to use the "tesarus search" in full text index in MS SQL Server. This is much better than using "%" in search if you have millions of records. tesarus have a small amount of memory consumption than the others. try to search this functions:)





answered Mar 31 '17 at 2:35 **Daryl Arenas**



best way is making full-text index on a column in table and use contain instead of LIKE



```
SELECT * FROM MyTable WHERE
contains(Column1 , N'word1' )
AND contains(Column1 , N'word2' )
AND contains(Column1 , N'word3' )
```

answered Oct 14 '17 at 6:34



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

Sign up with email







```
SELECT * FROM MyTable WHERE CONTAINS(Column1, 'word1 NEAR word2 NEAR word3')
SELECT * FROM MyTable WHERE CONTAINS(Column1, 'word1 ~ word2 ~ word3')
```

In addition, CONTAINSTABLE returns a rank for each document based on the proximity of "word1", "word2" and "word3". For example, if a document contains the sentence, "The word1 is word2 and word3," its ranking would be high because the terms are closer to one another than in other documents.

One other thing that I would like to add is that we can also use proximity_term to find columns where the words are inside a specific distance between them inside the column phrase.

answered May 1 '18 at 16:06



Anastasios Selmanis 2.184 3 19 36



This should ideally be done with the help of sql server full text search if using. However, if you can't get that working on your DB for some reason, here is a performance intensive solution:-



RECTN

```
-- table to search in

CREATE TABLE dbo.myTable

(
    myTableId int NOT NULL IDENTITY (1, 1),
    code varchar(200) NOT NULL,
    description varchar(200) NOT NULL -- this column contains the values we are going to

search in
    ) ON [PRIMARY]

GO

-- function to split space separated search string into individual words

CREATE FUNCTION [dbo].[fnSplit] (@StringInput nvarchar(max),

@Delimiter nvarchar(1))

RETURNS @OutputTable TABLE (
    id nvarchar(1000)
)

AS
```

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

Sign up with email





```
1, -1),
   LEN(@StringInput)));
   SET @StringInput = SUBSTRING(@StringInput, ISNULL(NULLIF(CHARINDEX)
   @Delimiter, @StringInput
   ),
   ), LEN
   @StringInput)
   + 1, LEN(@StringInput));
   INSERT INTO @OutputTable (id)
     VALUES (@String);
  END;
 RETURN;
END;
GO
-- this is the search script which can be optionally converted to a stored procedure
/function
declare @search varchar(max) = 'infection upper acute genito'; -- enter your search
string here
-- the searched string above should give rows containing the following
-- infection in upper side with acute genitointestinal tract
-- acute infection in upper teeth
-- acute genitointestinal pain
if (len(trim(@search)) = 0) -- if search string is empty, just return records ordered
alphabetically
begin
select 1 as Priority ,myTableid, code, Description from myTable order by Description
return;
end
declare @splitTable Table(
wordRank int Identity(1,1), -- individual words are assinged priority order (in order of
occurence/position)
word varchar(200)
```

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

Sign up with email



```
insert into @nonWordTable values
('of'),
('with'),
('at'),
('in'),
('for'),
('on'),
('by'),
('like'),
('up'),
('off'),
('near'),
('is'),
('are'),
(','),
(':'),
(';')
insert into @splitTable
select id from dbo.fnSplit(@search,' '); -- this function gives you a table with rows
containing all the space separated words of the search like in this e.g., the output
will be -
-- id
_____
-- infection
-- upper
-- acute
-- genito
delete s from @splitTable s join @nonWordTable n on s.word = n.id; -- trimming out non-
words here
declare @countOfSearchStrings int = (select count(word) from @splitTable); -- count of
space separated words for search
declare @highestPriority int = POWER(@countOfSearchStrings,3);
with plainMatches as
select myTableid, @highestPriority as Priority from myTable where Description like
@search -- exact matches have highest priority
union
select myTableid, @highestPriority-1 as Priority from myTable where Description like
@search + '%' -- then with something at the end
union
```

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

Sign up with email



```
splitWordMatches as( -- give each searched word a rank based on its position in the
searched string
                     -- and calculate its char index in the field to search
select myTable.myTableid, (@countOfSearchStrings - s.wordRank) as Priority, s.word,
wordIndex = CHARINDEX(s.word, myTable.Description) from myTable join @splitTable s on
myTable.Description like '%'+ s.word + '%'
-- and not exists(select myTableid from plainMatches p where p.myTableId =
myTable.myTableId) -- need not look into myTables that have already been found in
plainmatches as they are highest ranked
                                                                              -- this
one takes a long time though, so commenting it, will have no impact on the result
matchingRowsWithAllWords as (
select myTableid, count(myTableid) as myTableCount from splitWordMatches group
by(myTableid) having count(myTableid) = @countOfSearchStrings
, -- trim off the CTE here if you don't care about the ordering of words to be
considered for priority
wordIndexRatings as( -- reverse the char indexes retrived above so that words occuring
earlier have higher weightage
                     -- and then normalize them to sequential values
select s.myTableid, Priority, word, ROW NUMBER() over (partition by s.myTableid order by
wordindex desc) as comparativeWordIndex
from splitWordMatches s join matchingRowsWithAllWords m on s.myTableId = m.myTableId
wordIndexSequenceRatings as ( -- need to do this to ensure that if the same set of words
from search string is found in two rows.
                              -- their sequence in the field value is taken into account
for higher priority
    select w.myTableid, w.word, (w.Priority + w.comparativeWordIndex +
coalesce(sequncedPriority ,0)) as Priority
   from wordIndexRatings w left join
    select w1.myTableid, w1.priority, w1.word, w1.comparativeWordIndex,
count(w1.myTableid) as seguncedPriority
    from wordIndexRatings w1 join wordIndexRatings w2 on w1.myTableId = w2.myTableId
and w1.Priority > w2.Priority and w1.comparativeWordIndex>w2.comparativeWordIndex
    group by w1.myTableid, w1.priority,w1.word, w1.comparativeWordIndex
    sequencedPriority on w.myTableId = sequencedPriority.myTableId and w.Priority =
sequencedPriority.Priority
),
```

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

Sign up with email



```
completeSet as (
select myTableid, priority from plainMatches -- get plain matches which should be highest ranked union
select myTableid, OverallPriority as priority from prioritizedSplitWordMatches -- get ranked split word matches (which are ordered based on word rank in search string and sequence)
),
maximizedCompleteSet as( -- set the priority of a field value = maximum priority for that field value
select myTableid, max(priority) as Priority from completeSet group by myTableId
)
select priority, myTable.myTableid , code, Description from maximizedCompleteSet m join myTable on m.myTableId = myTable.myTableId
order by Priority desc, Description -- order by priority desc to get highest rated items on top
--offset 0 rows fetch next 50 rows only -- optional paging
```

edited Feb 15 at 14:23

answered Feb 13 at 12:17



why not use "in" instead?



Select *
from table
where columnname in (word1, word2, word3)



edited Nov 10 '17 at 2:31



Arun Vinoth

1.4k 13 30 81

answered Nov 9 '17 at 23:41



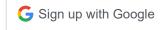
Michael Angerbauer
23 1

Because it doesn't work. Have you actually tried it? - mvp Nov 14 '17 at 22:40

2 I believe this will return only exact matches. – Murray Jan 18 '18 at 22:43

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

Sign up with email







SELECT * FROM MyTable WHERE Column1 Like "*word*"



This will display all the records where column1 has a partial value contains word.



edited Dec 27 '16 at 8:40







DECLARE @SearchStr nvarchar(100)
SET @SearchStr = ' '

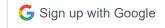
-2



```
CREATE TABLE #Results (ColumnName nvarchar(370), ColumnValue nvarchar(3630))
SET NOCOUNT ON
DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110)
SET @TableName = ''
SET @SearchStr2 = QUOTENAME('%' + @SearchStr + '%',''')
WHILE @TableName IS NOT NULL
BEGIN
   SET @ColumnName = ''
   SET @TableName =
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
                 INFORMATION SCHEMA. TABLES
        FROM
                      TABLE TYPE = 'BASE TABLE'
        WHERE
                   QUOTENAME(TABLE SCHEMA) + '.' + QUOTENAME(TABLE NAME) > @TableName
            AND
                   OBJECTPROPERTY(
            AND
                    OBJECT ID(
                        QUOTENAME (TABLE SCHEMA) + '.' + QUOTENAME (TABLE NAME)
                         ), 'IsMSShipped'
                           ) = 0
```

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

Sign up with email





```
SELECT MIN(QUOTENAME(COLUMN NAME))
            FROM
                     INFORMATION SCHEMA.COLUMNS
           WHERE
                          TABLE SCHEMA
                                          = PARSENAME(@TableName, 2)
                       TABLE NAME
                                   = PARSENAME(@TableName, 1)
                AND
                       DATA TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'int',
                AND
'decimal')
                       QUOTENAME(COLUMN NAME) > @ColumnName
                AND
        IF @ColumnName IS NOT NULL
        BEGIN
            INSERT INTO #Results
            EXEC
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName
+ ', 3630) FROM ' + @TableName + ' (NOLOCK) ' +
                ' WHERE ' + @ColumnName + ' LIKE ' + @SearchStr2
        END
    END
END
SELECT ColumnName, ColumnValue FROM #Results
DROP TABLE #Results
```

edited Mar 5 '18 at 11:02

Hemant Parmar 2,560 7 16 41 answered Mar 5 '18 at 10:41



user2274887

Thank you for this code snippet, which might provide some limited, immediate help. A <u>proper explanation would greatly improve its long-term value</u> by showing *why* this is a good solution to the problem, and would make it more useful to future readers with other, similar questions. Please <u>edit</u> your answer to add some explanation, including the assumptions you've made. – <u>Mogsdad Mar 5 '18 at 15:53</u>



select * from table where name regexp '^word[1-3]\$'

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

Sign up with email



edited Oct 29 '14 at 0:26

answered Jan 12 '13 at 7:27 vidyadhar 2,110 5 14 27

- 3 Is "regexp" standard SQL? Peter Mortensen Jan 31 '14 at 23:27
- 2 For the second query, shouldn't the word be quoted? Peter Mortensen Jan 31 '14 at 23:28
- 1 This code seems to check if the column *equals* one of the three words. The question is about checking if the column *contains all* of the three words. Sam Sep 5 '14 at 0:07
- Hiya, this may well solve the problem... but it'd be good if you could *edit your answer* and provide a little explanation about how and why it works:) Don't forget there are heaps of newbies on Stack overflow, and they could learn a thing or two from your expertise what's obvious to you might not be so to them. Taryn East Sep 5 '14 at 1:33

protected by Mogsdad Mar 5 '18 at 15:53

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?

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

Sign up with email

