# Find an object in SQL Server (cross-database)

18

7

If I've been told a table (or proc) name, but not which connected database the object is located in, is there any simple script to search for it? Maybe search somewhere in the System Databases? (I'm using SQL Server 2005)

sql-server sql-server-2005

edited Apr 6 '10 at 20:04
John Saunders
149k 22 207 369

asked Jun 21 '09 at 23:33
Margaret
2,195 17 45 63

1 Can't test this on 2005, but this should find the object in any database: ghostbin.com/paste/cwyxt – jumxozizi Apr 5 '18 at 13:59

## 5 Answers

32

There is an Information_Schema schema which is a set of views on tables from the SYS schema that you can query to get what you want.

The downside of the Information_Schema is that you have to write one query for each type of object. Upside is that the Information_Schema is more friendly to read as well.

The Sys schema will initially seem a little cryptic but it has the same information in a single spot.

Basically, there is a table called SysObjects in each database that has the names of all objects and their types.

So, you would want to search in a database as follows:

```
Select [name] as ObjectName, Type as ObjectType
```

```
Select [name] as ObjectName, Type as ObjectType
From Sys.Objects
Where 1=1
and [Name] like '%YourObjectName%'
and Type in ('U', 'P')
```

If you look up object types, you will find a whole list for views, triggers, etc.

Now, if you want to search for this in each database, you will have to iterate through the databases. You can do one of the following:

If you want to search through each database without any clauses, then use the sp_MSforeachdb as shown in an answer here.

If you only want to search specific databases, use the "USE DBName" and then search command.

You will benefit greatly from having it parameterized in that case. Note that the name of the database you are searching in will have to be replaced in each query (DatabaseOne, DatabaseTwo...). Check this out:

```
Declare @ObjectName VarChar (100)

Set @ObjectName = '%Customer%'

Select 'DatabaseOne' as DatabaseName, [name] as ObjectName, Type as ObjectType
From DatabaseOne.Sys.Objects
Where 1=1
and [Name] like @ObjectName
and Type in ('U', 'P')
UNION ALL
Select 'DatabaseTwo' as DatabaseName, [name] as ObjectName, Type as ObjectType
From DatabaseTwo.Sys.Objects
Where 1=1
and [Name] like @ObjectName
and Type in ('U', 'P')
UNION ALL
Select 'DatabaseThree' as DatabaseName, [name] as ObjectName, Type as ObjectType
From DatabaseThree.Sys.Objects
Where 1=1
and [Name] like @ObjectName
and Type in ('U', 'P')
```

4 What's the point of `WHERE 1=1` ? – itsho Dec 7 '14 at 7:57

3 @itsho, if the universe breaks, Raj More doesn't want to waste time running database queries. :) – user1172763 Feb 27 '15 at 15:43

4 @itsho it's for code readability. Each AND is on a separate line, and although it's not obvious in the way SO has displayed the code, they would often be indented. Also, it makes it ever-so-easier to comment out any/all conditions. Without the 1=1, you can't comment out the first condition without also removing the 'AND' on the next line. So, it does nothing to affect the query, but makes it easier for the user running the command. – Michael Bray Feb 11 '16 at 18:38

9

```
sp_MSforeachdb 'select db_name(), * From ?..sysobjects where xtype in (''U'', ''P'') And
name = ''ObjectName'''
```

Instead of 'ObjectName' insert object you are looking for. First column will display name of database where object is located at.

answered Jun 22 '09 at 0:13

Andrija Cacanovic
**8,046** 14 46 66

8 db_name() is the current database, "?" will be the database being searched. So the first column will be the same for all records, if you need to know which database the object is in use: sp_MSforeachdb 'select "?", * From ?..sysobjects where xtype in ("U", "P") And name = "ObjectName"' – David Martin Nov 12 '12 at 12:38

Best answer according to me. However, I'd remove the type filter in the case you don't know the type of the object you're searching. – Psddp Sep 10 '18 at 20:04

3

Easiest way is to hit up the information_schemas...

```
SELECT *
FROM information_schema.Tables
WHERE [Table_Name]='????'

SELECT *
```

answered Jun 21 '09 at 23:50

Alan Barber
**926** 4 15

1

You can use sp_MSforeachdb to search all databases.

declare @RETURN_VALUE int

declare @command1 nvarchar(2000)

set @command1 = "Your command goes here"

exec @RETURN_VALUE = sp_MSforeachdb @command1 = @command1

Raj

answered Jun 21 '09 at 23:46

Raj
**8,929** 2 38 49

0

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go


/*****************************************************************
Naziv procedure     : sp_rfv_FIND
Ime i prezime autora: Srdjan Nadrljanski
Datum kreiranja     : 13.06.2013.
Namena              : Traži sql objekat na celom serveru
Tabele              :
Ulazni parametri    :
```

```
                                        )
AS


declare @text varchar(1500),@textinit varchar (1500)
set @textinit=
'USE @sifra

insert into ##temp2
select ''@sifra''as dbName,a.[Object Name],a.[Object Type]
from(
 SELECT DISTINCT sysobjects.name AS [Object Name]   ,
case
when sysobjects.xtype = ''C'' then ''CHECK constraint''
when sysobjects.xtype = ''D'' then ''Default or DEFAULT constraint''
when sysobjects.xtype = ''F'' then ''Foreign Key''
when sysobjects.xtype = ''FN'' then ''Scalar function''
when sysobjects.xtype = ''P'' then ''Stored Procedure''
when sysobjects.xtype = ''PK'' then ''PRIMARY KEY constraint''
when sysobjects.xtype = ''S'' then ''System table''
when sysobjects.xtype = ''TF'' then ''Function''
when sysobjects.xtype = ''TR'' then ''Trigger''
when sysobjects.xtype = ''U'' then ''User table''
when sysobjects.xtype = ''UQ'' then ''UNIQUE constraint''
when sysobjects.xtype = ''V'' then ''View''
when sysobjects.xtype = ''X'' then ''Extended stored procedure''
end as [Object Type]
FROM sysobjects
WHERE
sysobjects.type in
(''C'',''D'',''F'',''FN'',''P'',''K'',''S'',''TF'',''TR'',''U'',''V'',''X'')
AND sysobjects.category = 0
AND CHARINDEX(''@SEARCHSTRING'',sysobjects.name)>0
AND ((CHARINDEX(''@notcontain'',sysobjects.name)=0 or
CHARINDEX(''@notcontain'',sysobjects.name)<>0))
)a'

    set @textinit=replace(@textinit,'@SEARCHSTRING',@SEARCHSTRING)
    set @textinit=replace(@textinit,'@notcontain',@notcontain)


SELECT name AS dbName,cast(null as varchar(255)) as ObjectName,cast(null as
varchar(255)) as ObjectType
```

```
declare @sifra VARCHAR(255),@suma int,@brojac int

set @suma=(select count(dbName) from ##temp1)

DECLARE c_k CURSOR LOCAL FAST_FORWARD FOR
SELECT dbName FROM ##temp1 ORDER BY dbName DESC

OPEN c_k
FETCH NEXT FROM c_K INTO @sifra
SET @brojac = 1
WHILE (@@fetch_status = 0 ) AND (@brojac <= @suma)
BEGIN

    set @text=replace(@textinit,'@sifra',@sifra)

    exec (@text)

    SET @brojac = @brojac +1

    DELETE FROM ##temp1 WHERE dbName = @sifra

       FETCH NEXT FROM c_k INTO @sifra
END
close c_k
DEALLOCATE c_k

select * from ##temp2
order by dbName,ObjectType
drop table ##temp2
drop table ##temp1
```

answered Jun 18 '13 at 11:44

preterani
**1** 1

Remember than any other process can read or modify global temp tables (prefixed by ##). You might get some unexpected results or errors if you are unlucky. In an unlikely scenario this can even compromise security. – jumxozizi Apr 5 '18 at 14:01