Hidden Features of SQL Server

Asked 16 years, 3 months ago Modified 12 years, 7 months ago

Viewed 61k times

215

votes

🔒 **Locked.** This question and its answers are locked because the question is off-topic but has historical significance. It is not currently accepting new answers or interactions.

What are some hidden features of SQL Server?

For example, undocumented system stored procedures, tricks to do things which are very useful but not documented enough?

Answers

Thanks to everybody for all the great answers!

Stored Procedures

- **sp_msforeachtable:** Runs a command with '?' replaced with each table name (v6.5 and up)
- **sp_msforeachdb:** Runs a command with '?' replaced with each database name (v7 and up)
- **sp_who2:** just like sp_who, but with a lot more info for troubleshooting blocks (v7 and up)

- **sp_helptext:** If you want the code of a stored procedure, view & UDF
- **sp_tables:** return a list of all tables and views of database in scope.
- **sp_stored_procedures:** return a list of all stored procedures
- **xp_sscanf:** Reads data from the string into the argument locations specified by each format argument.
- **xp_fixeddrives:**: Find the fixed drive with largest free space
- **sp_help:** If you want to know the table structure, indexes and constraints of a table. Also views and UDFs. Shortcut is Alt+F1

Snippets

- Returning rows in random order
- All database User Objects by Last Modified Date
- Return Date Only
- Find records which date falls somewhere inside the current week.
- Find records which date occurred last week.
- Returns the date for the beginning of the current week.
- Returns the date for the beginning of last week.

- See the text of a procedure that has been deployed to a server
- Drop all connections to the database
- Table Checksum
- Row Checksum
- Drop all the procedures in a database
- Re-map the login Ids correctly after restore
- Call Stored Procedures from an INSERT statement
- Find Procedures By Keyword
- Drop all the procedures in a database
- Query the transaction log for a database programmatically.

Functions

- HashBytes()
- EncryptByKey
- PIVOT command

Misc

- Connection String extras
- TableDiff.exe
- Triggers for Logon Events (New in Service Pack 2)
- Boosting performance with persisted-computed-columns (pcc).

- DEFAULT_SCHEMA setting in sys.database_principles
- Forced Parameterization
- Vardecimal Storage Format
- Figuring out the most popular queries in seconds
- Scalable Shared Databases
- Table/Stored Procedure Filter feature in SQL Management Studio
- Trace flags
- Number after a `GO` repeats the batch
- Security using schemas
- Encryption using built in encryption functions, views and base tables with triggers

sql-server t-sql

Share

edited May 6, 2012 at 16:53

community wiki
31 revs, 8 users 65%
Sklivvz

4 If known, it would be nice to include the applicable versions with each answer. (2000 and up, 2005, 2000 only, etc.) – b w Sep 1, 2009 at 18:37

There is a lot of goodness in this question. Please do not delete it! :-) – Sklivvz Mar 2, 2012 at 10:34

Comments disabled on deleted / locked posts / reviews

84 Answers

Sorted by: Highest score (default)

1 2 3 Next

91 votes

In Management Studio, you can put a number after a GO end-of-batch marker to cause the batch to be repeated that number of times:

```
PRINT 'X'
GO 10
```

Will print 'X' 10 times. This can save you from tedious copy/pasting when doing repetitive stuff.

Share

answered Sep 26, 2008 at 18:13

community wiki
GilM

70 votes

A lot of SQL Server developers still don't seem to know about the **OUTPUT clause** (SQL Server 2005 and newer) on the DELETE, INSERT and UPDATE statement.

It can be extremely useful to know which rows have been INSERTed, UPDATEd, or DELETEd, and the OUTPUT clause allows to do this very easily - it allows access to the "virtual" tables called `inserted` and `deleted` (like in triggers):

```
DELETE FROM (table)
OUTPUT deleted.ID, deleted.Description
WHERE (condition)
```

If you're inserting values into a table which has an INT IDENTITY primary key field, with the OUTPUT clause, you can get the inserted new ID right away:

```
INSERT INTO MyTable(Field1, Field2)
OUTPUT inserted.ID
VALUES (Value1, Value2)
```

And if you're updating, it can be extremely useful to know what changed - in this case, `inserted` represents the new values (after the UPDATE), while `deleted` refers to the old values before the UPDATE:

```
UPDATE (table)
SET field1 = value1, field2 = value2
OUTPUT inserted.ID, deleted.field1, inserted.field1
WHERE (condition)
```

If a lot of info will be returned, the output of OUTPUT can also be redirected to a temporary table or a table variable (`OUTPUT INTO @myInfoTable`).

Extremely useful - and very little known!

Marc

Share

answered Aug 7, 2009 at 9:03

community wiki
marc_s

52
votes

`sp_msforeachtable`: Runs a command with '?' replaced with each table name. e.g.

```
exec sp_msforeachtable "dbcc dbreindex('?')"
```

You can issue up to 3 commands for each table

```
exec sp_msforeachtable
    @Command1 = 'print ''reindexing table ?''',
    @Command2 = 'dbcc dbreindex(''?'')',
    @Command3 = 'select count (*) [?] from ?'
```

Also, `sp_MSforeachdb`

Share

edited Feb 17, 2010 at 2:34

community wiki
5 revs, 3 users 46%
Mitch Wheat

2 You can get the name of the table in the query by using single quotes around the question mark. sp_msforeachtable "select

count(*), '?' as tabenm from ?" – Jody Oct 29, 2008 at 13:35

51 votes

Connection String extras:

MultipleActiveResultSets=true;

This makes ADO.Net 2.0 and above read multiple, forward-only, read-only results sets on a single database connection, which can improve performance if you're doing a lot of reading. You can turn it on even if you're doing a mix of query types.

Application Name=MyProgramName

Now when you want to see a list of active connections by querying the sysprocesses table, your program's name will appear in the program_name column instead of ".Net SqlClient Data Provider"

Share

answered Sep 23, 2008 at 15:21

community wiki
Chris Wenham

7 I made Application Name a requirement at my company. Every new app must have a unique name. Makes tracking down which app locked/broke something a lot easier. – Neil N Dec 22, 2009 at 20:31

2 Application Name is also available as a filter in profiler. It helps a lot if you want to only see your queries and not the queries of

your coworkers. – Mathias F Jul 20, 2010 at 18:14

33

votes

TableDiff.exe

- Table Difference tool allows you to discover and reconcile differences between a source and destination table or a view. Tablediff Utility can report differences on schema and data. The most popular feature of tablediff is the fact that it can generate a script that you can run on the destination that will reconcile differences between the tables.

Link

Share

edited Jun 20, 2020 at 9:12

community wiki
2 revs
Sklivvz

31

votes

A less known TSQL technique for returning rows in random order:

```
-- Return rows in a random order
SELECT
    SomeColumn
FROM
    SomeTable
ORDER BY
    CHECKSUM(NEWID())
```

Share

edited May 24, 2009 at 14:59

community wiki
2 revs, 2 users 94%
Mitch Wheat

6 Great for small result sets. I wouldn't use it on a table with more than 10000 rows unless you've got time to spare
– John Sheehan Sep 23, 2008 at 15:18

I've used it on tables much larger than that, and it wasn't too slow. – Mitch Wheat Sep 23, 2008 at 15:25

What's the purpose of the CHECKSUM()? You can order by just NEWID(). – Jonas Lincoln Oct 9, 2008 at 15:01

6 I've even seen decent results on 100,000,000 (100 mil) rows, w/o CHECKSUM(). Also, I have to ask as well, why not just ORDER BY NEWID? – Troy DeMonbreun Oct 14, 2008 at 16:40

5 @GateKiller: I've rolled back your edit, because the Checksum() is not a mistake; it reduces the size of the sort column. – Mitch Wheat May 24, 2009 at 15:00

30
votes

In Management Studio, you can quickly get a comma-delimited list of columns for a table by :

1. In the Object Explorer, expand the nodes under a given table (so you will see folders for Columns, Keys, Constraints, Triggers etc.)

2. Point to the Columns folder and drag into a query.

This is handy when you don't want to use heinous format returned by right-clicking on the table and choosing Script Table As..., then Insert To... This trick does work with the other folders in that it will give you a comma-delimited list of names contained within the folder.

Share

answered Jul 20, 2010 at 15:22

community wiki
Thomas

23

votes

Row Constructors

You can insert multiple rows of data with a single insert statement.

```
INSERT INTO Colors (id, Color)
VALUES (1, 'Red'),
       (2, 'Blue'),
       (3, 'Green'),
       (4, 'Yellow')
```

Share

answered Dec 18, 2009 at 19:39

community wiki
Rob Boek

I voted this up, but then tried it in MSSQL 2005, and it doesn't work. 2008 only? – richardtallent Dec 22, 2009 at 21:08

11 Yes, it's a new feature of 2008. – Rob Boek Dec 22, 2009 at 22:09

2 This was a feature that I missed when I came from DB2 to SQL Server. In DB2, there was a significant speed improvement when using this instead of individual insert statements – Nathan Koop Aug 30, 2010 at 19:15

22
votes

If you want to know the table structure, indexes and constraints:

```
sp_help 'TableName'
```

Share

answered Sep 26, 2008 at 14:54

community wiki
Eduardo Molteni

Combine this tip with its shortcut key! First highlight a tablename and then hit ALT+F1 – Michael J Swart Nov 25, 2011 at 19:20

22
votes

HashBytes() to return the MD2, MD4, MD5, SHA, or SHA1 hash of its input.

Share

edited Apr 28, 2009 at 8:23

community wiki
3 revs, 2 users 88%
Joel Coehoorn

Nice one! The correct link is msdn.microsoft.com/en-us/library/ms174415(SQL.90).aspx (2005 version) – Sklivvz
Sep 23, 2008 at 14:24

You're right, that was the 2008 version of the docs, even though the pages are pretty much identical. Fixed now.
– Joel Coehoorn Sep 23, 2008 at 14:29

20
votes

Figuring out the most popular queries

- With sys.dm_exec_query_stats, you can figure out many combinations of query analyses by a single query.

Link with the commnad

```
select * from sys.dm_exec_query_stats
order by execution_count desc
```

Share

edited Oct 31, 2010 at 13:34

community wiki
2 revs, 2 users 63%
Sklivvz

17

votes

[The spatial results tab can be used to create art](#).

[enter link description here http://michaeljswart.com/wp-content/uploads/2010/02/venus.png](http://michaeljswart.com/wp-content/uploads/2010/02/venus.png)

Share

answered Mar 22, 2011 at 12:35

community wiki
Martin Smith

7 I saw Jesus in my query results! – P Daddy Mar 22, 2011 at 15:14

6 Pfff... What chump wastes his time messing with the spatial results tab. Oh wait... You know, I thought that post looked familiar, now I remember why. – Michael J Swart Mar 24, 2011 at 14:21

16

votes

[EXCEPT and INTERSECT](#)

Instead of writing elaborate joins and subqueries, these two keywords are a much more elegant shorthand and readable way of expressing your query's intent when comparing two query results. New as of SQL Server 2005, they strongly complement UNION which has already existed in the TSQL language for years.

The concepts of EXCEPT, INTERSECT, and UNION are fundamental in set theory which serves as the basis and foundation of relational modeling used by all modern RDBMS. Now, Venn diagram type results can be more intuitively and quite easily generated using TSQL.

Share

edited Nov 25, 2008 at 23:02

community wiki
2 revs
Ray Vega

16
votes

I know it's not exactly hidden, but not too many people know about the PIVOT command. I was able to change a stored procedure that used cursors and took 2 minutes to run into a speedy 6 second piece of code that was one tenth the number of lines!

Share

edited Apr 28, 2009 at 8:31

community wiki
2 revs, 2 users 67%
Ronald Wildenberg

16
votes

useful when restoring a database for Testing purposes or whatever. Re-maps the login ID's correctly:

```
EXEC sp_change_users_login 'Auto_Fix', 'Mary', NULL, 'B3
```

Share

edited Mar 22, 2011 at 13:37

community wiki
2 revs, 2 users 86%
Kolten

I have had this proc not work before, and I had to change the objects ownership to a temp user, drop the original user, re - add the original and assign the ownership back. Ugh...
– StingyJack Nov 25, 2008 at 20:09

15

votes

Drop all connections to the database:

```
Use Master
Go

Declare @dbname sysname

Set @dbname = 'name of database you want to drop connect

Declare @spid int
Select @spid = min(spid) from master.dbo.sysprocesses
where dbid = db_id(@dbname)
While @spid Is Not Null
Begin
        Execute ('Kill ' + @spid)
        Select @spid = min(spid) from master.dbo.sysproc
        where dbid = db_id(@dbname) and spid > @spid
End
```

Share

answered Sep 23, 2008 at 15:56

community wiki
GateKiller

Is there a one-liner, or a drop database parameter that does this for me? I notice that if you attempt to 'delete database' through the ui, there's a checkbox for 'close existing connections' which implies that it's a boolean parameter. – DevinB Apr 16, 2009 at 16:14

1 Actually, I just found a two line solution. ALTER DATABASE [@DATABASE_NAME@] SET READ_ONLY WITH ROLLBACK IMMEDIATE --this disconnects all users ALTER DATABASE [@DATABASE_NAME@] SET READ_WRITE WITH ROLLBACK IMMEDIATE DROP DATABASE [@DATABASE_NAME@] – DevinB Apr 16, 2009 at 17:02

1 `ALTER DATABASE MyDB SET SINGLE_USER WITH ROLLBACK IMMEDIATE` will prevent any new connections from occurring, too. – ErikE Sep 8, 2010 at 23:02

15
votes

Table Checksum

```
Select CheckSum_Agg(Binary_CheckSum(*)) From Table With
```

Row Checksum

```
Select CheckSum_Agg(Binary_CheckSum(*)) From Table With
Value
```

Share

answered Sep 23, 2008 at 15:56

community wiki
GateKiller

2 These allow you to produce a checksum for all the data in the table. It is a simple and quick way to check if two rows or two tables are the same. – GateKiller Sep 23, 2008 at 17:55

15
votes

I'm not sure if this is a hidden feature or not, but I stumbled upon this, and have found it to be useful on many occassions. You can concatonate a set of a field in a single select statement, rather than using a cursor and looping through the select statement.

Example:

```
DECLARE @nvcConcatonated nvarchar(max)
SET @nvcConcatonated = ''

SELECT @nvcConcatonated = @nvcConcatonated + C.CompanyNa
FROM tblCompany C
WHERE C.CompanyID IN (1,2,3)

SELECT @nvcConcatonated
```

Results:

```
Acme, Microsoft, Apple,
```

Share

edited Nov 9, 2011 at 16:34

community wiki
2 revs, 2 users 94%
Sheki

2 you can also use COALESCE() to do the same thing without the need to initialize the variable. SELECT @nvcConcatonated = COALESCE(@nvcConcatonated+',','')+CAST(C.CompanyName as VARCHAR(255)) FROM... – Christopher Klein Sep 28, 2009 at 20:45

This also works in an update statement. Sometimes useful for doing things like concatenating a list of ID's that were updated. – EBarr Sep 14, 2011 at 21:35

14 votes

If you want the code of a stored procedure you can:

```
sp_helptext 'ProcedureName'
```

(not sure if it is hidden feature, but I use it all the time)

Share

answered Sep 23, 2008 at 16:56

community wiki
Eduardo Molteni

Don't know why, but sp_helptext output is a bit goofy on any overly long lines in the original. When scripting Sprocs this doens't happen, so maybe there is another, more robust, export mechanism? sp_helptext 'MyView' also useful. – Kristen Feb 16, 2009 at 15:24

I'm not sure what you mean. For me, the SPs code are outputed with the same format I have scripted them in the original file (with all the CRs, etc) – Eduardo Molteni Feb 17, 2009 at 2:23

I don't recall the *exact* details, but it has to do with the way the text is stored--something about page size, I believe. The output is *mostly* correct, but every now & then you get an extra line break. – RolandTumble Jul 29, 2009 at 23:24

13

votes

A stored procedure trick is that you can call them from an INSERT statement. I found this very useful when I was working on an SQL Server database.

```
CREATE TABLE #toto (v1 int, v2 int, v3 char(4), status c
INSERT #toto (v1, v2, v3, status) EXEC dbo.sp_fulubulu(s
SELECT * FROM #toto
DROP TABLE #toto
```

Share

edited Nov 1, 2008 at 4:45

community wiki
2 revs, 2 users 91%
edomaur

1 Sadly can't be used with @TableVariable – Kristen Feb 16, 2009 at 15:29

The pain with this very useful technique is that unlike most #tables, you have to fully define all the columns. The lazy way of doing this is to create the #table inside the proc you are calling right at the end, then sp_help in tempdb, copy and paste, remove code from proc. Done – adolf garlic Mar 31, 2009 at 13:45

12

votes

In SQL Server 2005/2008 to show row numbers in a SELECT query result:

```
SELECT ( ROW_NUMBER() OVER (ORDER BY OrderId) ) AS RowNu
        GrandTotal, CustomerId, PurchaseDate
FROM Orders
```

ORDER BY is a compulsory clause. The OVER() clause tells the SQL Engine to sort data on the specified column (in this case OrderId) and assign numbers as per the sort results.

Share

edited Dec 19, 2009 at 3:56

community wiki
2 revs, 2 users 86%
Binoj Antony

wouldn't be simpler if they used syntactic suger in sql engine to parse it syntax word as "RowNumberInTable" – none Oct 31, 2010 at 13:40

1 +1 for window functions. You can do things OVER a subset of records by using OVER (PARTITION BY ...) msdn.microsoft.com/en-us/library/ms189461%28v=SQL.100%29.aspx – Matt Stephenson Dec 7, 2011 at 16:56

10

votes

Useful for parsing stored procedure arguments: xp_sscanf

> Reads data from the string into the argument locations specified by each format argument.
>
> The following example uses xp_sscanf to extract two values from a source string based on their positions in the format of the source string.

```
DECLARE @filename varchar (20), @message varchar (20)
EXEC xp_sscanf 'sync -b -fproducts10.tmp -rrandom', 'syr
  @filename OUTPUT, @message OUTPUT
SELECT @filename, @message
```

> Here is the result set.

```
-------------------- --------------------
products10.tmp       random
```

Share

edited Sep 24, 2008 at 8:18

community wiki
2 revs
Sklivvz

4 I must be having a dumb moment (no, really). Can you tell me where we can use this? – Raj More Sep 1, 2009 at 19:51

9

votes

Return Date Only

```
Select Cast(Floor(Cast(Getdate() As Float))As Datetime)
```

or

```
Select DateAdd(Day, 0, DateDiff(Day, 0, Getdate()))
```

Share

edited Feb 17, 2009 at 11:28

community wiki
3 revs
GateKiller

Short version - SELECT CAST(FLOOR(CAST(@DateTime AS FLOAT))AS DATETIME) – Meff Oct 1, 2008 at 21:33

Hell yes. CASTFLOORCAST rules. – StingyJack Nov 25, 2008 at 20:11

Can't find a reference to it, but I seem to remember tests that suggested SELECT DateAdd(Day, 0, DateDiff(Day, 0, @DateTime)) was faster. Happy to be enlightened, either way! – Kristen Feb 16, 2009 at 15:44

Found this sqlteam.com/forums/topic.asp?TOPIC_ID=35296#107617 but it didn't include the CAST/FLOOR method. An informal test on a medium sized recordset suggests DATEADD may be about 7% faster than CAST/FLOOR - not enough to worry about for most situations – Kristen Feb 16, 2009 at 16:00

I've added the other method, however; my quick testing shows that the cast floor method is 800 Nanoseconds quicker. So nothing in it really. – GateKiller Feb 17, 2009 at 11:40

9

votes

dm_db_index_usage_stats

This allows you to know if data in a table has been updated recently even if you don't have a DateUpdated column on the table.

```
SELECT OBJECT_NAME(OBJECT_ID) AS DatabaseName, last_user
FROM sys.dm_db_index_usage_stats
WHERE database_id = DB_ID( 'MyDatabase')
AND OBJECT_ID=OBJECT_ID('MyTable')
```

Code from: http://blog.sqlauthority.com/2009/05/09/sql-server-find-last-date-time-updated-for-any-table/

Information referenced from: SQL Server - What is the date/time of the last inserted row of a table?

Available in SQL 2005 and later

Share

edited May 23, 2017 at 11:33

community wiki
2 revs
Nathan Koop

7

votes

Here are some features I find useful but a lot of people don't seem to know about:

```
sp_tables
```

> Returns a list of objects that can be queried in the current environment. This means any object that can appear in a FROM clause, except synonym objects.

Link

```
sp_stored_procedures
```

> Returns a list of stored procedures in the current environment.

Link

Share

answered Sep 23, 2008 at 14:18

community wiki
Sklivvz

7

votes

Find records which date falls somewhere inside the current week.

```
where dateadd( week, datediff( week, 0, TransDate ), 0 )
dateadd( week, datediff( week, 0, getdate() ), 0 )
```

Find records which date occurred last week.

```
where dateadd( week, datediff( week, 0, TransDate ), 0 )
dateadd( week, datediff( week, 0, getdate() ) - 1, 0 )
```

Returns the date for the beginning of the current week.

```
select dateadd( week, datediff( week, 0, getdate() ), 0
```

Returns the date for the beginning of last week.

```
select dateadd( week, datediff( week, 0, getdate() ) - 1
```

Share

answered Sep 23, 2008 at 15:55

community wiki
GateKiller

Fine but index on TransDate would not be used. I would rather write – vaso Nov 18, 2010 at 4:44

where TransDate >= convert(datetime, floor(convert(float, dateadd(day, -datepart(weekday, @date)+1, @date)))) and TransDate >= convert(datetime, floor(convert(float, dateadd(day, 7-datepart(weekday, @date)+1, @date)))) – vaso Nov 18, 2010 at 4:45

correction: where TransDate >= convert(datetime, floor(convert(float, dateadd(day, -datepart(weekday, @date)+1, @date)))) and TransDate < convert(datetime, floor(convert(float, dateadd(day, 7-datepart(weekday, @date)+1, @date)))) – vaso Nov 18, 2010 at 4:46

7

votes

Not so much a hidden feature but setting up key mappings in Management Studio under Tools\Options\Keyboard: Alt+F1 is defaulted to sp_help "selected text" but I cannot live without the adding Ctrl+F1 for sp_helptext "selected text"

Share

answered Sep 23, 2008 at 23:17

community wiki
JohnD

I use to configure the USE command also, for moving along the db's – Jhonny D. Cano -Leftware- Jun 30, 2009 at 13:21

7

votes

Persisted-computed-columns

- Computed columns can help you shift the runtime computation cost to data modification phase. The computed column is stored with the rest of the row and is transparently utilized when the expression on the computed columns and the query matches. You can

> also build indexes on the PCC’s to speed up filtrations and range scans on the expression.

Link

Share

edited Jun 20, 2020 at 9:12

community wiki
Sklivvz

7
votes

There are times when there's no suitable column to sort by, or you just want the default sort order on a table and you want to enumerate each row. In order to do that you can put "(select 1)" in the "order by" clause and you'd get what you want. Neat, eh?

```
select row_number() over (order by (select 1)), * from c
```

Share

answered Sep 5, 2010 at 15:37

community wiki
Denis Valeev

6
votes

Simple encryption with EncryptByKey

Share

answered Sep 23, 2008 at 14:34

community wiki
John Sheehan

1 2 3 Next