

# T-SQL SET Statements

[www.tsq.info](http://www.tsq.info)

On Transact sql language the SET statements allow you to change the current session handling of specific information like: dateformat, system language, lock timeout, rowcount.

## Date and time

- SET Datefirst
- SET Dateformat

## Locks

- SET Deadlock\_priority
- SET Lock\_timeout

## Miscellaneous

- SET Concat\_null\_yields\_null
- SET Cursor\_close\_on\_commit
- SET Identity\_insert
- SET Language

## Query Execution

- SET Rowcount
- SET Noexec

## SET Datefirst

SET Datefirst - sets the first day of the week to a number from 1 through 7.

### SET Datefirst Syntax:

```
SET DATEFIRST { number | @number_variable } ;
```

### SET Datefirst Example:

```
SET DATEFIRST 1 ;
```

### **Messages:**

Command(s) completed successfully.

```
SELECT @@DATEFIRST AS 'First Day';
```

### **Result:**

```
1
```

## **SET Dateformat**

SET Dateformat - sets the order of the month, day, and year date parts.

### **SET Dateformat Syntax:**

```
SET DATEFORMAT { format | @format_variable } ;
```

### **SET Dateformat Example:**

```
SET DATEFORMAT dmy;  
GO  
DECLARE @date_variable datetime2 = '31/08/2014 10:11:43.1234567';  
SELECT @date_variable;  
GO
```

### **Result:**

```
2014-08-31 10:11:43.1234567
```

## **SET Deadlock\_priority**

SET Deadlock\_priority - sets the importance of the current session if it is deadlocked with another session.

### **SET Deadlock\_priority Syntax:**

```
SET DEADLOCK_PRIORITY { LOW | NORMAL | HIGH | <numeric-priority> |  
@deadlock_variable } ;  
<numeric-priority> ::= { -10 | -9 | ... | 0 | ... | 9 | 10 }
```

## **SET Deadlock\_priority Example:**

```
SET DEADLOCK_PRIORITY NORMAL;  
GO
```

### **Messages:**

Command(s) completed  
successfully.

## **SET Lock\_timeout**

SET Lock\_timeout - sets the number of milliseconds of statement that waits for a lock to be released.

### **SET Lock\_timeout Syntax:**

```
SET LOCK_TIMEOUT milliseconds_number ;
```

### **SET Lock\_timeout Example:**

```
SET LOCK_TIMEOUT 3600;  
GO
```

### **Messages:**

Command(s) completed  
successfully.

## **SET Concat\_null\_yields\_null**

SET Concat\_null\_yields\_null - Checks whether concatenation results are treated as null or empty string values.

### **SET Concat\_null\_yields\_null Syntax:**

```
SET CONCAT_NULL_YIELDS_NULL { ON | OFF } ;
```

## **SET Concat\_null\_yields\_null Example:**

```
USE model;  
GO  
SET CONCAT_NULL_YIELDS_NULL ON;  
GO  
SELECT 'test' + NULL;  
GO
```

### **Results**

NULL

```
USE model;  
GO  
SET CONCAT_NULL_YIELDS_NULL OFF;  
GO  
SELECT 'test' + NULL;  
GO
```

### **Results**

test

## **SET Cursor\_close\_on\_commit**

SET Cursor\_close\_on\_commit - The default value for CURSOR\_CLOSE\_ON\_COMMIT is OFF.

With CURSOR\_CLOSE\_ON\_COMMIT set OFF the server will not close cursors when you commit a transaction.

### **SET Cursor\_close\_on\_commit Syntax:**

```
SET CURSOR_CLOSE_ON_COMMIT { ON | OFF } ;
```

### **SET Cursor\_close\_on\_commit Example:**

```
USE model;  
GO  
CREATE TABLE my_table (a INT, b CHAR(10));  
GO  
INSERT INTO my_table VALUES (1,'a'), (2,'b');  
GO
```

```
SET CURSOR_CLOSE_ON_COMMIT OFF;  
GO  
PRINT 'BEGIN TRANSACTION';  
BEGIN TRAN;  
PRINT 'Declare cursor';  
DECLARE my_cursor CURSOR FOR SELECT * FROM my_table;  
PRINT 'Open cursor';  
OPEN my_cursor;  
PRINT 'COMMIT TRANSACTION';  
COMMIT TRAN;  
PRINT 'Use cursor after commit transaction';  
FETCH NEXT FROM my_cursor;  
CLOSE my_cursor;  
DEALLOCATE my_cursor;  
GO
```

### **Messages**

```
BEGIN TRANSACTION  
Declare cursor  
Open cursor  
COMMIT TRANSACTION  
Use cursor after commit transaction
```

## **SET Identity\_insert**

SET Identity\_insert - allow to be inserted explicit values into the identity column of a table.

### **SET Identity\_insert Syntax:**

```
SET IDENTITY_INSERT [ database_name . [ schema_name ] . ] table { ON | OFF } ;
```

### **SET Identity\_insert Example:**

```
USE model;  
GO  
CREATE TABLE Department(  
ID INT IDENTITY NOT NULL PRIMARY KEY, Name VARCHAR(250) NOT NULL);  
  
GO  
INSERT INTO Department(Name)
```

```
VALUES ('Anthropology'), ('Biology'), ('Chemistry'), ('Computer Science'),
('Economics');
GO
DELETE FROM Department WHERE name='Biology';
GO
SELECT * FROM Departments;
GO
```

<b>ID</b>	<b>Name</b>
1	Anthropology
3	Chemistry
4	Computer Science
5	Economics

```
USE model;
GO
INSERT INTO Departments (ID, Name) VALUES (2, 'Biology');
GO
```

#### **Messages**

Msg 544, Level 16, State 1, Line 1  
Cannot insert explicit value for identity column in table 'Departments' when  
IDENTITY\_INSERT is set to OFF.

```
USE model;
GO
SET IDENTITY_INSERT Departments ON;
GO
```

#### **Messages**

Command(s) completed  
successfully.

```
USE model;
GO
INSERT INTO Departments (ID, Name) VALUES (2, 'Biology');
GO
```

#### **Messages**

(1 row(s) affected)

## **SET Language**

SET Language - sets the language of session.  
The session language establish the format of date and system messages.

## **SET Language Syntax:**

```
SET LANGUAGE { [ N ] 'language' | @language_variable } ;
```

## **SET Language Example:**

```
USE model;
GO
DECLARE @MyDay DATETIME;
SET @MyDay = '06/21/2014';

SET LANGUAGE French;
SELECT DATENAME(month, @MyDay) AS 'French Month';

SET LANGUAGE English;
SELECT DATENAME(month, @MyDay) AS 'English Month' ;
GO
```

```
French Month
Juin
English Month
June
```

## **SET Rowcount**

SET Rowcount - sets the number of rows for sql query.  
When the specified number of rows are returned the execution of query stops.

## **SET Rowcount Syntax:**

```
SET ROWCOUNT { number | @number_variable } ;
```

## **SET Rowcount Example:**

```
USE model;
GO
SET ROWCOUNT 3;
```

```
GO
SELECT * FROM Departments WHERE id <=3;
GO
```

<b>ID</b>	<b>Name</b>
1	Anthropology
2	Biology
3	Chemistry

```
USE model;
GO
SET ROWCOUNT 2;
GO
SELECT * FROM Departments;
GO
```

<b>ID</b>	<b>Name</b>
1	Anthropology
2	Biology

## **SET Noexec**

SET Noexec - sets the compile of each query but does not execute the queries.

### **SET Noexec Syntax:**

```
SET NOEXEC { ON | OFF } ;
```

### **SET Noexec Example:**

```
USE model;
GO
PRINT 'OK';
GO
SET NOEXEC ON;
GO
SELECT * FROM Departments WHERE id > 3;
GO
SET NOEXEC OFF;
GO
```



**Messages**  
OK

## **Resources:**

[www.tsql.info/set/set-statements.php](http://www.tsql.info/set/set-statements.php)