

## Who ran the DELETE statements on SQL Server.??

It's commonly happened at all work place where an user ran a DELETE command by mistake or intensely on a SQL Server and no one will be accepting who did this. Here, I would like to demonstrate a way using the transaction log to track down helpful information related to this fact.

### Step 1

Before moving forward, we will create a database and a table on which I will delete some data. Run the below SQL code to create a database and table.

```
--Create DB.
USE [master];
GO
CREATE DATABASE ReadingDBLog;
GO
-- Create tables.
USE ReadingDBLog;
GO
CREATE TABLE [Location] (
    [Sr.No] INT IDENTITY,
    [Date] DATETIME DEFAULT GETDATE (),
    [City] CHAR (25) DEFAULT 'Bangalore');
```

### Step 2

We have created a database named "ReadingDBLog" and a table 'Location' with three columns. Now we will insert a 100 rows into the table.

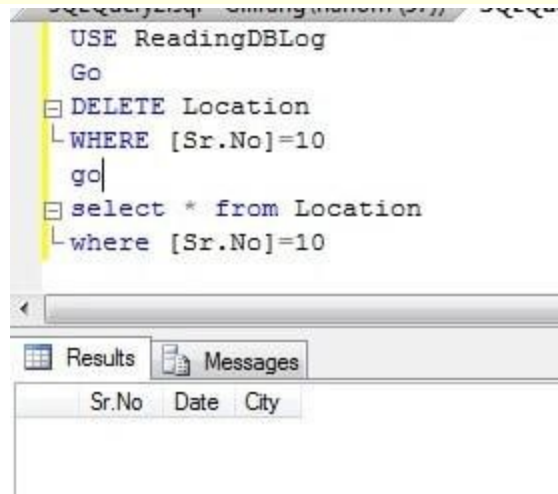
```
USE ReadingDBLog
GO
INSERT INTO Location DEFAULT VALUES ;
GO 100
```

### Step 3

Now go ahead and delete some rows to check who has deleted your data.

```
USE ReadingDBLog
GO
DELETE Location WHERE [Sr.No]=10
```

```
GO
SELECT * FROM Location WHERE [Sr.No]=10
GO
```



You can see in the above screenshot that a row has been deleted from the table "Location". I also ran a SELECT statement to verify the data has been deleted.

#### Step 4

Now we have to search the transaction log file to find the info about the deleted rows. Run the below command to get info about all deleted transactions.

```
USE ReadingDBLog
GO
SELECT
    [Transaction ID],
    Operation,
    Context,
    AllocUnitName

FROM
    fn_dblog(NULL, NULL)
WHERE
    Operation = 'LOP_DELETE_ROWS'
```

	Transaction ID	Operation	Context	AllocUnitName
1	0000:00000445	LOP_DELETE_ROWS	LCX_MARK_AS_GHOST	sys.sysprfiles.clst
2	0000:00000445	LOP_DELETE_ROWS	LCX_MARK_AS_GHOST	sys.sysprfiles.clst
3	0000:00000445	LOP_DELETE_ROWS	LCX_MARK_AS_GHOST	sys.sysclsobjs.nc
4	0000:00000445	LOP_DELETE_ROWS	LCX_MARK_AS_GHOST	sys.sysowners.nc2
5	0000:000004ce	LOP_DELETE_ROWS	LCX_HEAP	dbo.Location

All transactions which have executed a DELETE statement will display by running the above command and we can see this in the above screenshot. As we are searching for deleted data in table Location, we can see this in the last row. We can find the table name in the "AllocUnitName" column. The last row says a DELETE statement has been performed on a HEAP table 'dbo.Location' under transaction ID 0000:000004ce. Now capture the transaction ID from here for our next command.

### Step 5

We found the transaction ID from the above command which we will use in the below command to get the transaction SID of the user who has deleted the data.

```
USE ReadingDBLog
GO
SELECT
    Operation,
    [Transaction ID],
    [Begin Time],
    [Transaction Name],
    [Transaction SID]
FROM
    fn_dblog(NULL, NULL)
WHERE
    [Transaction ID] = '0000:000004ce'
AND
    [Operation] = 'LOP_BEGIN_XACT'
```

```

use ReadingDBLog
go
SELECT
    Operation,
    [Transaction ID],
    [Begin Time],
    [Transaction Name],
    [Transaction SID]
FROM
    fn_dblog(NULL, NULL)
WHERE
    [Transaction ID] = '0000:000004ce'
AND
    [Operation] = 'LOP_BEGIN_XACT'

```

	Operation	Transaction ID	Begin Time	Transaction Name	Transaction SID
1	LOP_BEGIN_XACT	0000:000004ce	2013/10/14 12:55:17:630	DELETE	0x01050000000000005150000009F11BA296C79F97398D0CF19E8030000

Here, we can see the [Begin Time] of this transaction which will also help filter out the possibilities in finding the exact info like when the data was deleted and then you can filter on the base of begin time when that command was executed.

We can read the above output as "A DELETE statement began at 2013/10/14 12:55:17:630 under transaction ID 0000:000004ce by user transaction SID 0x01050000000000005150000009F11BA296C79F97398D0CF19E8030000.

Now our next step is to convert the transaction SID hexadecimal value into text to find the real name of the user.

## Step 6

Now we will figure out who ran the DELETE command. We will copy the hexadecimal value from the transaction SID column for the DELETE transaction and then pass that value into the SUSER\_SNAME () function.

```

USE MASTER
GO
SELECT
    SUSER_SNAME(0x01050000000000005150000009F11BA296C79F97398D0CF19E8030000)

```



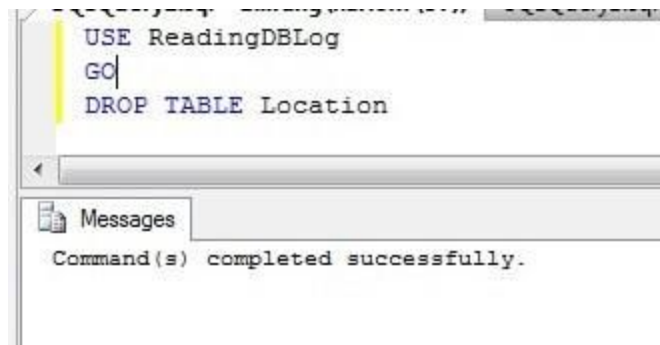
Now we have found the user that did the delete.

## Finding a user who ran a DROP statement

### Step 1

Here I am going to drop table Location.

```
USE ReadingDBLog
GO
DROP TABLE Location
```



### Step 2

Similarly if you drop any object or you perform anything operation in your database it will get logged in the transaction log file which will be visible by using this function fn\_dblog.

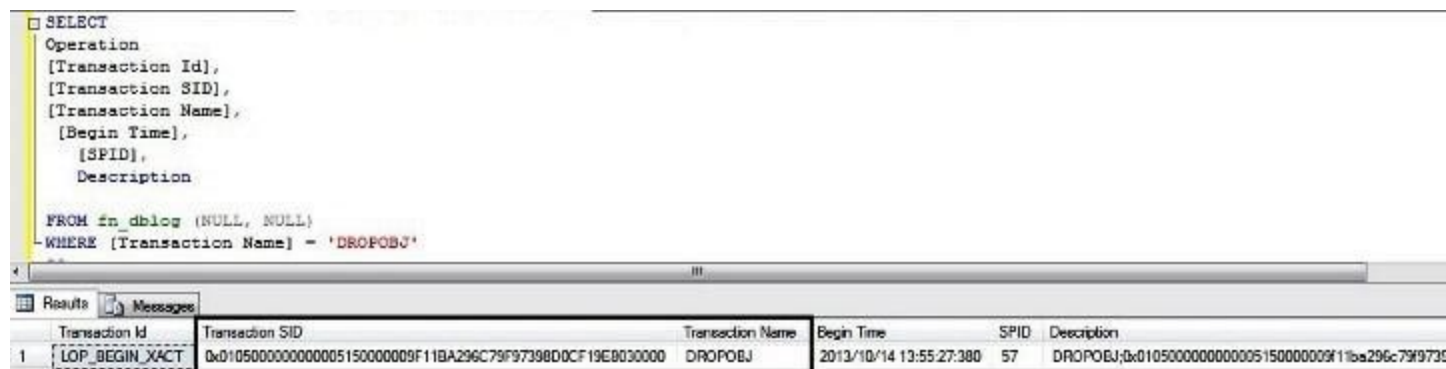
Run the below script to display all logs which have been logged under DROPOBJ statement.

```
USE ReadingDBLog
GO
SELECT
Operation,
[Transaction Id],
[Transaction SID],
[Transaction Name],
```

```

[Begin Time],
[SPID],
Description
FROM fn_dblog (NULL, NULL)
WHERE [Transaction Name] = 'DROPOBJ'
GO

```



The screenshot shows a SQL query window with the following query:

```

SELECT
  Operation
  [Transaction Id],
  [Transaction SID],
  [Transaction Name],
  [Begin Time],
  [SPID],
  Description
FROM fn_dblog (NULL, NULL)
WHERE [Transaction Name] = 'DROPOBJ'

```

The Results tab shows a single row of data:

Transaction Id	Transaction SID	Transaction Name	Begin Time	SPID	Description
1 LOP_BEGIN_XACT	0x0105000000000005150000009F11BA296C79F97398D0CF19E8030000	DROPOBJ	2013/10/14 13:55:27:380	57	DROPOBJ;0x0105000000000005150000009F11ba296c79f97398d0cf19e8030000

Here we can find the transaction SID and all required info which we need to find the user.

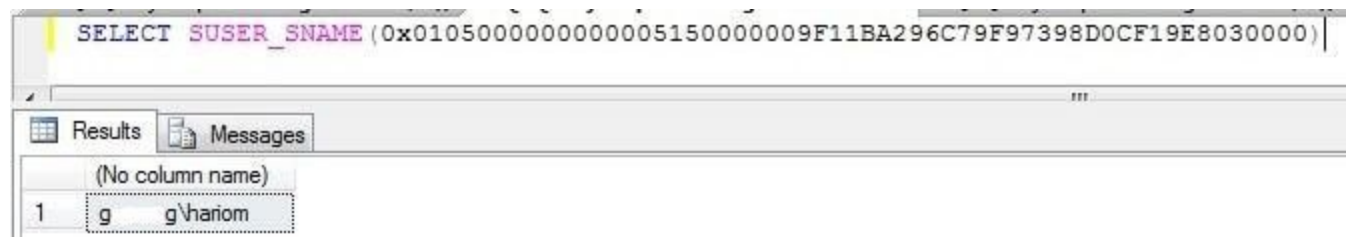
### Step 3

Now we can pass the transaction SID into system function SUSER\_SNAME () to get the exact user name.

```

SELECT
SUSER_SNAME(0x0105000000000005150000009F11BA296C79F97398D0CF19E8030000)

```



The screenshot shows a SQL query window with the following query:

```

SELECT SUSER_SNAME (0x0105000000000005150000009F11BA296C79F97398D0CF19E8030000)

```

The Results tab shows a single row of data:

(No column name)
g g\harm

Once again, we found the user in question.

Reference:

<https://www.mssqltips.com/sqlservertip/3090/how-to-find-user-who-ran-drop-or-delete-statements-on-your-sql-server-objects/>