**SQL Server Rename Table**

**SQL Rename table using Transact SQL**

SQL Server does not have any statement that directly renames a table. However, it does provide you with a stored procedure named sp\_rename that allows you to change the name of a table.

The following shows the syntax of using the sp\_rename stored procedure for changing the name of a table:

|  |  |
| --- | --- |
| 1 | EXEC sp\_rename 'old\_table\_name', 'new\_table\_name' |

Note that both old and new name of the table whose name is changed must be enclosed in single quotations.

#### To rename a column

1. In **Object Explorer**, connect to an instance of Database Engine.
2. On the Standard bar, click **New Query**.
3. The following example renames the column TerritoryID in the table Sales.SalesTerritory to TerrID. Copy and paste the following example into the query window and click **Execute**.

Copy

USE AdventureWorks2012;

GO

EXEC sp\_rename 'Sales.SalesTerritory.TerritoryID', 'TerrID', 'COLUMN';

GO

To rename procedure

--Rename the stored procedure.

EXEC sp\_rename 'HumanResources.uspGetAllEmployeesTest', 'uspEveryEmployeeTest';

#### Query All Database Files

List all the Data and Log files for your database with the file sizes.

|  |  |  |
| --- | --- | --- |
| 1 | select d.name, m.name, m.physical\_name, m.type\_desc, m.size | |
| 2 | from sys.master\_files m |

|  |  |
| --- | --- |
| 3 | inner join sys.databases d |
| 4 | on (m.database\_id = d.database\_id) | |

|  |  |
| --- | --- |
| 5 | order by 1, 2 |

#### Index Last Rebuild Date

Use this query to check when an Index was last updated.

|  |  |  |
| --- | --- | --- |
| 1 | SELECT OBJECT\_NAME(object\_id) [TableName], | |
| 2 | name [IndexName], |

|  |  |  |
| --- | --- | --- |
| 3 | STATS\_DATE(object\_id, stats\_id) [LastUpdated] | |
| 4 | FROM sys.stats |

|  |  |
| --- | --- |
| 5 | WHERE name NOT LIKE '\_WA%' |
| 6 | AND STATS\_DATE(object\_id, stats\_id) IS NOT NULL | |

|  |  |  |
| --- | --- | --- |
| 7 | AND OBJECTPROPERTY(object\_id, 'IsMSShipped') = 0 | |
| 8 | ORDER BY [LastUpdated] desc |

#### Table And Index Size

If you need to find your largest tables this query will show how big each one is.

|  |  |  |
| --- | --- | --- |
| 1 | SELECT s.Name AS SchemaName, t.NAME AS TableName, | |
| 2 | p.rows AS RowCounts, |

|  |  |
| --- | --- |
| 3 | SUM(a.total\_pages) \* 8 AS TotalSpaceKB, |
| 4 | SUM(a.used\_pages) \* 8 AS UsedSpaceKB, | |

|  |  |  |
| --- | --- | --- |
| 5 | (SUM(a.total\_pages) - SUM(a.used\_pages)) \* 8 AS UnusedSpaceKB | |
| 6 | FROM sys.tables t |

|  |  |  |
| --- | --- | --- |
| 7 | INNER JOIN sys.schemas s ON s.schema\_id = t.schema\_id | |
| 8 | INNER JOIN sys.indexes i ON t.OBJECT\_ID = i.object\_id |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | INNER JOIN sys.partitions p ON i.object\_id = p.OBJECT\_ID AND i.index\_id = p.index\_id | | |
| 10 | | INNER JOIN sys.allocation\_units a ON p.partition\_id = a.container\_id |

|  |  |
| --- | --- |
| 11 | WHERE |
| 12 | t.NAME NOT LIKE 'dt%' AND t.is\_ms\_shipped = 0 | |

|  |  |
| --- | --- |
| 13 | AND i.OBJECT\_ID &gt; 255 |
| 14 | GROUP BY t.Name, s.Name, p.Rows | |

|  |  |
| --- | --- |
| 15 | ORDER BY TotalSpaceKB desc |

#### Shrink Transaction Logs

If you need to quickly need to shrink a transaction log. Use the query above to get the logical file name.

|  |  |  |
| --- | --- | --- |
| 1 | ALTER DATABASE [Database\_Name] SET RECOVERY SIMPLE | |
| 2 | DBCC SHRINKFILE(Database\_Logfile\_Name) |

|  |  |
| --- | --- |
| 3 | ALTER DATABASE [Database\_Name] SET RECOVERY FULL |

#### Query Database Backup History

Query backup history for specific database. This shows whether it was a full or transaction log backup, start and end times, and backup size.

|  |  |
| --- | --- |
| 1 | SELECT upper(name) as 'Backup Type', |
| 2 | CONVERT(VARCHAR(20), backup\_start\_date, 100) as 'Start', | |

|  |  |  |
| --- | --- | --- |
| 3 | CONVERT(VARCHAR(20), backup\_finish\_date, 100) as 'Finish', | |
| 4 | backup\_size as 'Size', |

|  |  |  |
| --- | --- | --- |
| 5 | recovery\_model as 'Recovery Model' | |
| 6 | FROM msdb..backupset |

|  |  |  |
| --- | --- | --- |
| 7 | WHERE database\_name = 'Database\_Name' | |
| 8 | ORDER by backup\_finish\_date DESC |

#### Grant Execute Permission To All Stored Procedures

Use this command to grant EXECUTE permission to all the stored procedures in a database.

|  |  |
| --- | --- |
| 1 | GRANT EXECUTE TO [DOMAIN\USER] |

#### Search The Contents Of SQL Objects

To search the contents of Stored Procedures, Functions, Triggers, or Views use this query:

|  |  |  |
| --- | --- | --- |
| 1 | DECLARE @SEARCH VARCHAR(255), @notcontain Varchar(255) | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | SELECT @SEARCH = 'specOuterPkgMaterialJoin' | |
| 4 |  |

|  |  |  |
| --- | --- | --- |
| 5 | SELECT sysobjects.name AS [Object Name] , sysobjects.xtype [Type] | |
| 6 | FROM sysobjects,syscomments |

|  |  |
| --- | --- |
| 7 | WHERE sysobjects.id = syscomments.id |
| 8 | AND CHARINDEX(@SEARCH,syscomments.text)>0 | |

To search the Command Text of SQL Agent Jobs for a string use the following query:

|  |  |  |
| --- | --- | --- |
| 1 | USE [msdb] | |
| 2 | GO |

|  |  |  |
| --- | --- | --- |
| 3 | SELECT servers.srvname [Server], jobs.name [Job Name], | |
| 4 | steps.step\_id [Step ID], steps.command [Command], |

|  |  |
| --- | --- |
| 5 | jobs.enabled [Enabled] |
| 6 | FROM dbo.sysjobs jobs | |

|  |  |
| --- | --- |
| 7 | JOIN dbo.sysjobsteps steps ON steps.job\_id = jobs.job\_id |
| 8 | JOIN master.dbo.sysservers servers ON servers.srvid = jobs.originating\_server\_id | |

|  |  |
| --- | --- |
| 9 | WHERE steps.command LIKE N'%SEARCH\_TERM%' |

#### Simple Backup And Restore Of Tables

Use this to quickly backup a table, and update records from backup if needed.

|  |  |
| --- | --- |
| 1 | -- Table Backup |
| 2 | select \* into MyBackup from MyTable; | |

|  |  |
| --- | --- |
| 3 |  |
| 4 | -- Table Restore | |

|  |  |
| --- | --- |
| 5 | Update MyTable set Enabled = 1 where Id in (select Id from MyBackup where status = 1) |

#### SQL Login Creation

Use this to quickly create a SQL Login and SQL User, then grant datareader and datawriter, and also grant Execute access to certain Stored Procedures.

**view source**

**print?**

|  |  |
| --- | --- |
| 1 | -- Create db login |
| 2 | CREATE LOGIN [DOMAIN\ServiceIdX] FROM WINDOWS; | |

|  |  |  |
| --- | --- | --- |
| 3 | GO | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | -- Create db user and grant role membership |
| 6 | CREATE USER [DOMAIN\ServiceIdX] FOR LOGIN [DOMAIN\ServiceIdX] WITH DEFAULT\_SCHEMA = dbo; | |

|  |  |  |
| --- | --- | --- |
| 7 | GO | |
| 8 |  |

|  |  |  |
| --- | --- | --- |
| 9 | -- Add to db roles | |
| 10 | | EXEC sp\_addrolemember N'db\_datareader', N'DOMAIN\ServiceIdX' | |

|  |  |  |
| --- | --- | --- |
| 11 | EXEC sp\_addrolemember N'db\_datawriter', N'DOMAIN\ServiceIdX' | |
| 12 |  |

|  |  |
| --- | --- |
| 13 | -- Grant Exec on Stored Procedure |
| 14 | GRANT EXEC on sp\_MyStoredProc to [DOMAIN\ServiceIdX] | |