SQL | Dynamic SQL exercise | Basic dynamic SQL to pass in table name

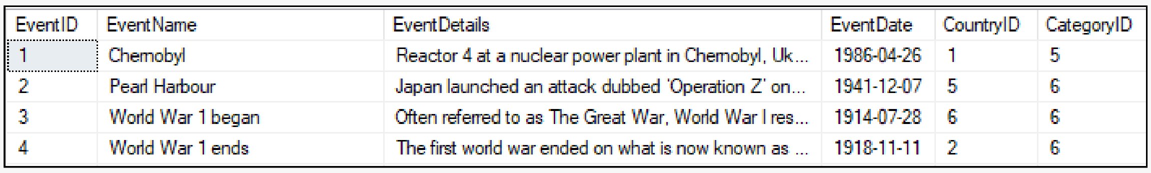
Software ==>	SQL (198 exercises)
Version ==>	Any version of SQL Server
Topic ==>	Dynamic SQL (4 exercises)
Level ==>	Relatively easy
Subject ==>	SQL training

Before you can do this exercise, you'll need to download and unzip this file (if you have any problems doing this, click here for help). Once you've done this:

1. Go into SQL Server Management Studio;
2. Open the SQL file you've just unzipped (you can press CTRL + O to do this); then
3. Execute this script.

This will generate the database that you'll need to use in order to do this exercise (note that the database and script are only to be used for exercises published on this website, and may not be reused or distributed in any form without the prior written permission of Wise Owl).

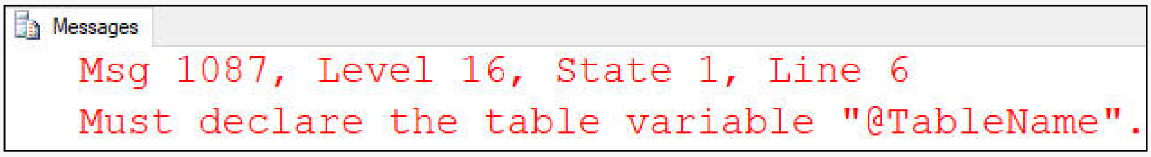
The aim of this exercise is to be able to pass different table names to a select statement, to show different sets of rows. First create a stored procedure which selects everything from the table **tblEvent**:



EventID	EventName	EventDetails	EventDate	CountryID	CategoryID
1	Chernobyl	Reactor 4 at a nuclear power plant in Chernobyl, Uk...	1986-04-26	1	5
2	Pearl Harbour	Japan launched an attack dubbed 'Operation Z' on...	1941-12-07	5	6
3	World War 1 began	Often referred to as The Great War, World War I res...	1914-07-28	6	6
4	World War 1 ends	The first world war ended on what is now known as ...	1918-11-11	2	6

The first few events.

Now add a **varchar(max)** parameter called **@TableName** to hold the name of the table from which you want to extract data, and use its value in place of **tblEvent** in the **FROM** clause. Sadly this doesn't work and results in an error:



```
Messages
Msg 1087, Level 16, State 1, Line 6
Must declare the table variable "@TableName".
```

It takes a while to accept that this can't work!

Declare a **varchar(max)** variable called **@SQL**, and set it equal to the SQL script up to the **FROM** keyword, then add your parameter value on the end:

```
DECLARE @SQL VARCHAR(MAX) =
'SELECT * FROM ' +
   @TableName
```

Then finally either use **EXEC (@SQL)** or **Exec SP_SQLEXEC @SQL** at the bottom of the stored procedure to run the select statement contained in your variable.

Now try running the stored procedure:

```
exec uspChangingTables 'tblevent'
```

*Try passing in each of the **tblEvent**, **tblCountry** and **tblContinent** table names.*

Optionally save this as **Turning the table.sql**, and close it down.

You can unzip this file to see the answers to this exercise, although please remember this is for your personal use only.

0 threads Add post