

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```



```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic

-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.

-- sysname is used to reference database object names.


-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'


-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic

-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.

-- sysname is used to reference database object names.


-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'


-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic

-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.

-- sysname is used to reference database object names.


-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'


-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```



```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname = 'Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```

```
-- sample #2
DECLARE @TableName sysname ='Sales.SalesOrderHeader'
EXECUTE ('SELECT * FROM '+@TableName) -- T-SQL string with
variable - dynamic
-- sysname is a system-supplied user-defined data type that is
functionally equivalent to nvarchar(128) except that it is not
nullable.
-- sysname is used to reference database object names.

-- sample #3
EXECUTE sp_executeSQL N'SELECT * FROM
Purchasing.PurchaseOrderHeader'

-- sample #4
DECLARE @SQL varchar(256), @Table sysname;
SET @SQL='SELECT * FROM'; SET @Table = 'Production.Product'
SET @SQL = @SQL+' '+@Table
PRINT @SQL -- for debugging dynamic SQL prior to execution of
generated static code
EXEC (@SQL) -- SQL Server 2005 execute dynamic SQL
```

Dynamic queries with local variables and parameter(s)

t-sql parameterized query

```
-- Parameters are NOT ALLOWED everywhere in an SQL statement -
hence dynamic SQL
DECLARE @Color varchar(16) = 'Yellow'
SELECT Color=@Color, ProductCount=COUNT(Color)
FROM AdventureWorks2008.Production.Product
WHERE Color = @Color
/*      Color      ProductCount
      Yellow          36          */

-- SQL Server 2012 sp_executeSQL usage with input and output
```