Store Procedure

Course: INFO6210 Data Management and Database Design
Week: 10+
Instructor: Mutsalklisana

Store Procedure

- A stored procedure is an executable database object that contains one or more SQL statements
- A stored procedure is also called a sproc (pronounced either as one word or as “ess-proc”) or a procedure.
- Stored procedures are *<u>precompiled</u>*
  – That means that the execution plan for the SQL code *<u>is compiled the first time the procedure is executed and is then saved in its compiled form</u>*
  – For this reason, *<u>stored procedures execute faster than an equivalent SQL script</u>*

Source: http://www.w3computing.com/

Store Procedure

- You use the EXEC statement to run, or call a procedure.
  - If this statement is the first line in a batch, you can omit the EXEC keyword and code just the procedure name
  - To make your code easier to read, however, you should always include the EXEC keyword
- You can call a stored procedure from within another stored procedure
  - A stored procedure can be call from within itself
  - This technique, called a recursive call or recursion, is seldom used in SQL programming

Source: http://www.w3computing.com/

Advantage of Store Procedure

- One of the advantages of using procedures is that *application programmers and end users don’t need to know the structure of the database or how to code SQL*
- Another advantage of using procedures is that they *can restrict and control access to a database*
- If you use procedures in this way, you can *prevent both accidental errors and malicious damage*

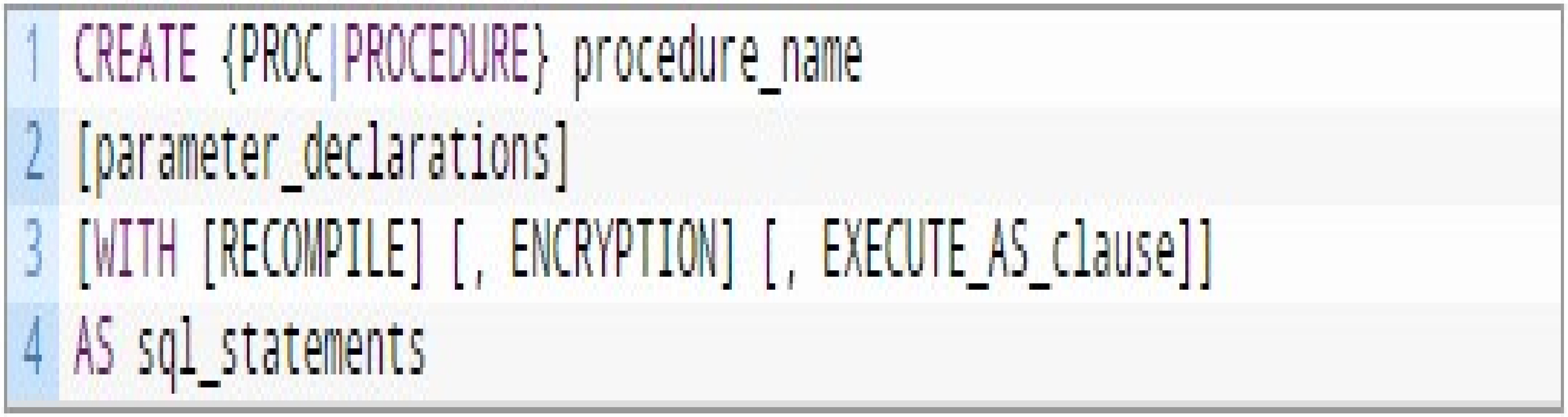
Source: http://www.w3computing.com/

Create a Stored Procedure

- Use **CREATE PROC** statement to create a stored procedure
- Code the name of the procedure in the **CREATE PROC** clause
  - Stored procedure names can't be the same name as the name of any other object in the database
  - Also good practice, to <u>*prefix its name*</u> with the letter **"sp"** to <u>*help distinguish a store procedure from other DB objects*</u>

Create a Stored Procedure

- **Syntax** of CREATE PROC statement

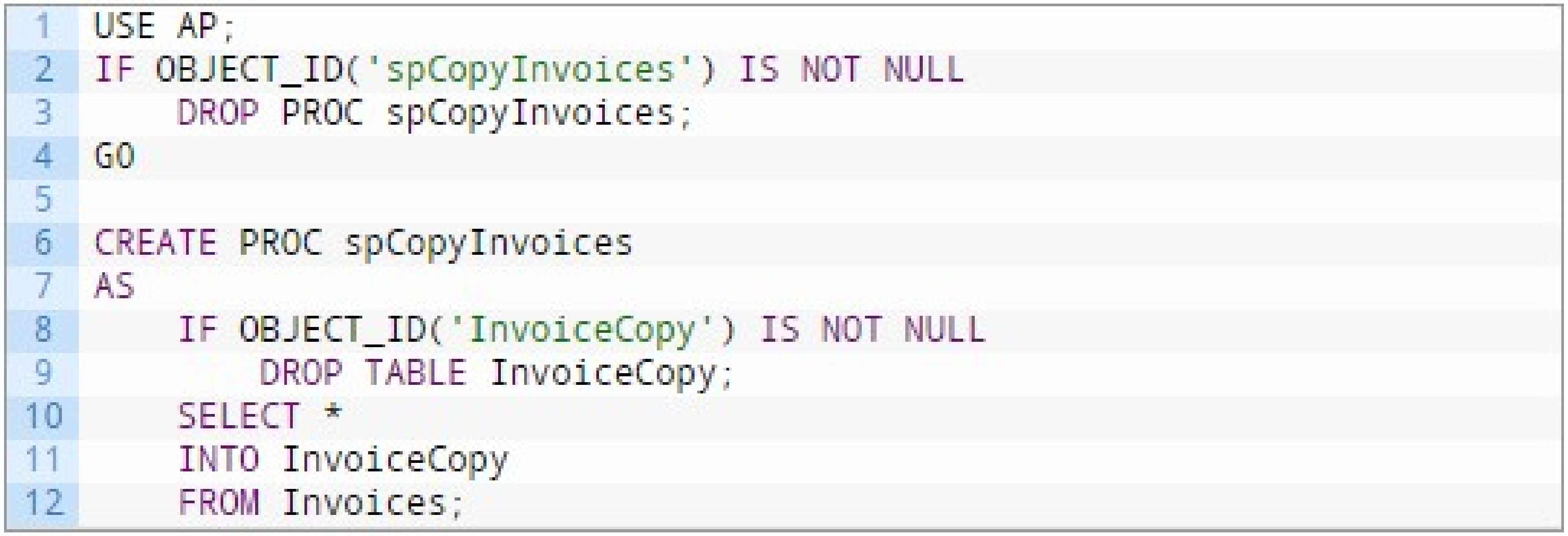


```
CREATE {PROC|PROCEDURE} procedure_name
[parameter_declarations]
[WITH [RECOMPILE] [, ENCRYPTION] [, EXECUTE_AS_clause]]
AS sql_statements
```

Source: http://www.w3computing.com/

Example#1: Create a Stored Procedure

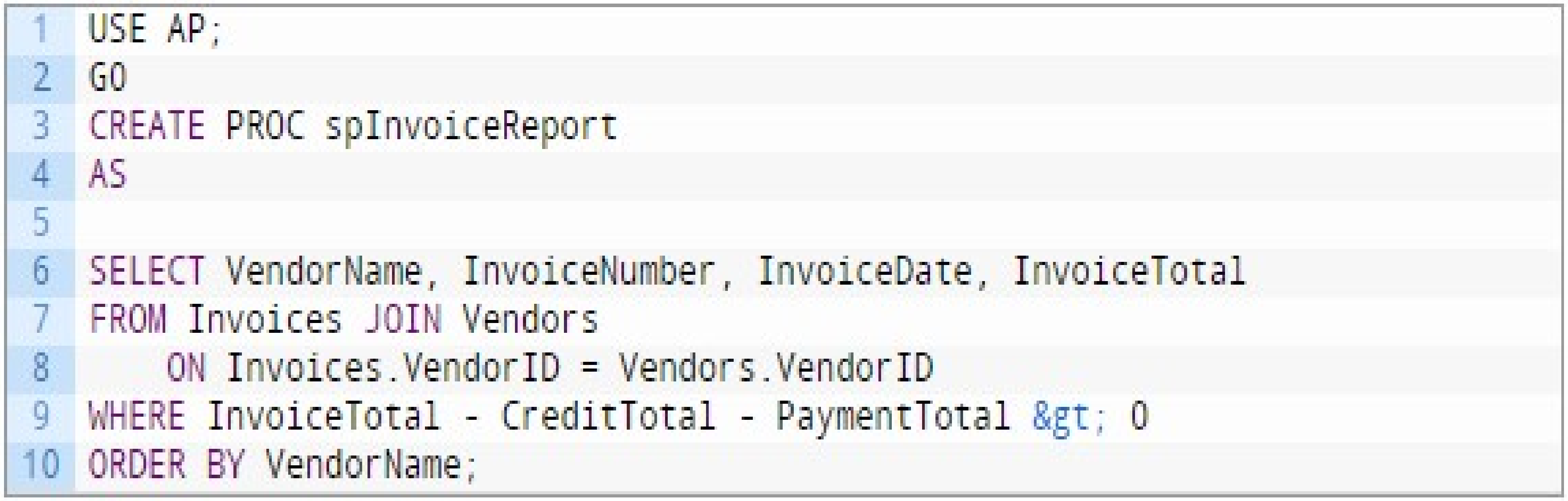
- Sample script that creates a stored procedure that <u>*copies a table*</u>



```
USE AP;
IF OBJECT_ID('spCopyInvoices') IS NOT NULL
    DROP PROC spCopyInvoices;
GO

CREATE PROC spCopyInvoices
AS
    IF OBJECT_ID('InvoiceCopy') IS NOT NULL
        DROP TABLE InvoiceCopy;
    SELECT *
    INTO InvoiceCopy
    FROM Invoices;
```

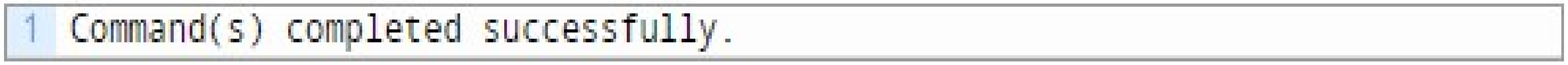
Source: http://www.w3computing.com/

Example#2: Create a Stored Procedure



```
USE AP;
GO
CREATE PROC spInvoiceReport
AS

SELECT VendorName, InvoiceNumber, InvoiceDate, InvoiceTotal
FROM Invoices JOIN Vendors
    ON Invoices.VendorID = Vendors.VendorID
WHERE InvoiceTotal - CreditTotal - PaymentTotal &gt; 0
ORDER BY VendorName;
```

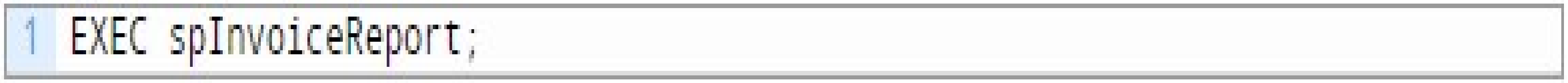
The response from the system



```
Command(s) completed successfully.
```

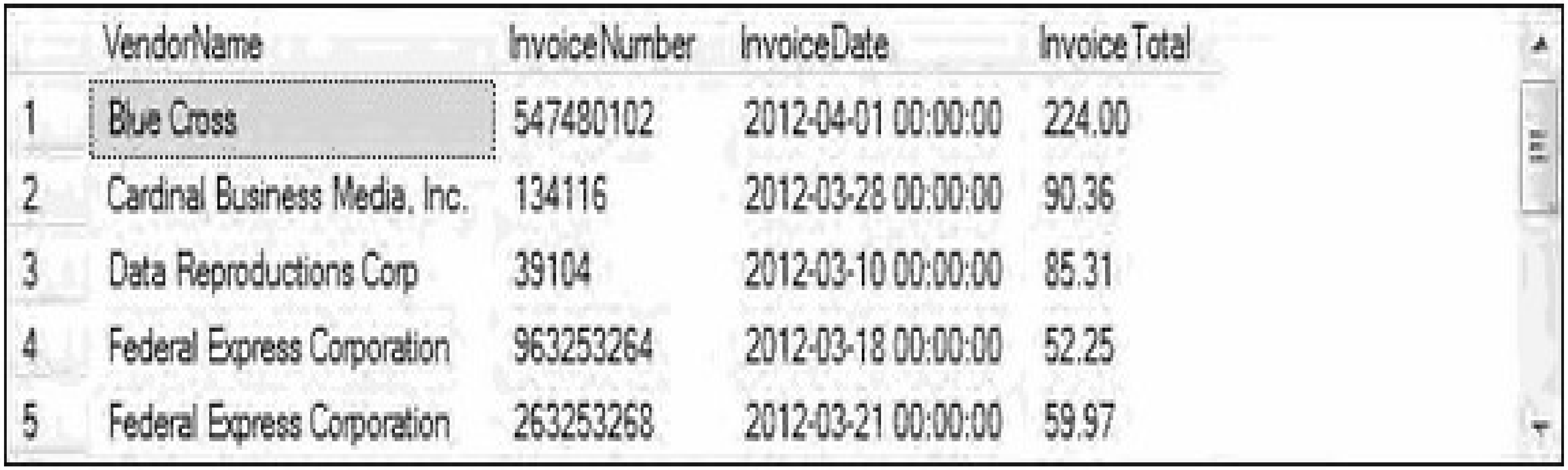
Source: http://www.w3computing.com/

Example#2: Create a Stored Procedure



```
EXEC spInvoiceReport;
```

The result set created by the procedure



	VendorName	InvoiceNumber	InvoiceDate	InvoiceTotal
1	Blue Cross	547480102	2012-04-01 00:00:00	224.00
2	Cardinal Business Media, Inc.	134116	2012-03-28 00:00:00	90.36
3	Data Reproductions Corp	39104	2012-03-10 00:00:00	85.31
4	Federal Express Corporation	963253264	2012-03-18 00:00:00	52.25
5	Federal Express Corporation	263253268	2012-03-21 00:00:00	59.97

Source: http://www.w3computing.com/

Create a Stored Procedure

- When CREATE PROC statement is executed,
  - *Syntax of the SQL statement within procedure is checked*
  - In case there is a coding error, system will respond with error message and the procedure stops

Source: http://www.w3computing.com/

Create a Stored Procedure

- Need to change the database context by coding a USE statement before the CREATE PROC statement
  - Because stored procedure is created in the current DB
  - **CREATE PROC** *must be first and only statement in the batch*
    - *(Looking at the example in the following slides, the script creates a procedure after a USE -DROP PROC statement then GO command, the CREATE PROC statement is the first to follow)*

Source: http://www.w3computing.com/

Create a ‘temporary’ Stored Procedure

- In addition to stored procedures that are stored in the current DB, you can create temporary stored procedures that are stored in the tempdb database
- These *procedures exist only while the current DB session is open*, so they **are not used often**
- To identify a temporary stored procedure,
  - Prefix the name with *one number sign (#)* for a *local procedure* and *(##) for a global procedure*

Source: http://www.w3computing.com/

OPTION: RECOMPILE

- **RECOMPILE** prevents the system from precompiling the procedure
  - The execution plan for the procedure must be compiled each time it’s executed; will slow down most procedures
  - Generally, **should omit this option**

Source: http://www.w3computing.com/

OPTION: ENCRYPTION

- **ENCRYPTION** is a security option that *prevents the user from being able to view the declaration of a store procedure*
  - System stores the procedure as an object in DB; it also stores code for the procedure
  - *Use Encryption option, if you want to block user to examine information embedded in the code*

OPTION: EXECUTE_AS clause

- **EXECUTE AS** clause *allow users to execute the stored procedure with a specified security context*
- For example, you can use this clause to allow user to execute the stored procedure with same security permissions as you
  - That way, you can be sure that the stored procedure will work for the caller even if the caller doesn’t have permission to access all of the objects that you used within the stored procedure

Source: http://www.w3computing.com/

Stored Procedure - Return Values

- In addition to passing output parameters back to the calling program, stored procedures also pass back a return value
- *Default return value is zero*
- However, use **RETURN** statement to return a number
- **Syntax:**

  RETURN [integer_expression]

Return Value: Example

- For example, if a stored procedure updates rows, you may want to return the number of rows that have been updated
- To do that, you can use the @@ROWCOUNT function to return the number of rows that have been updated

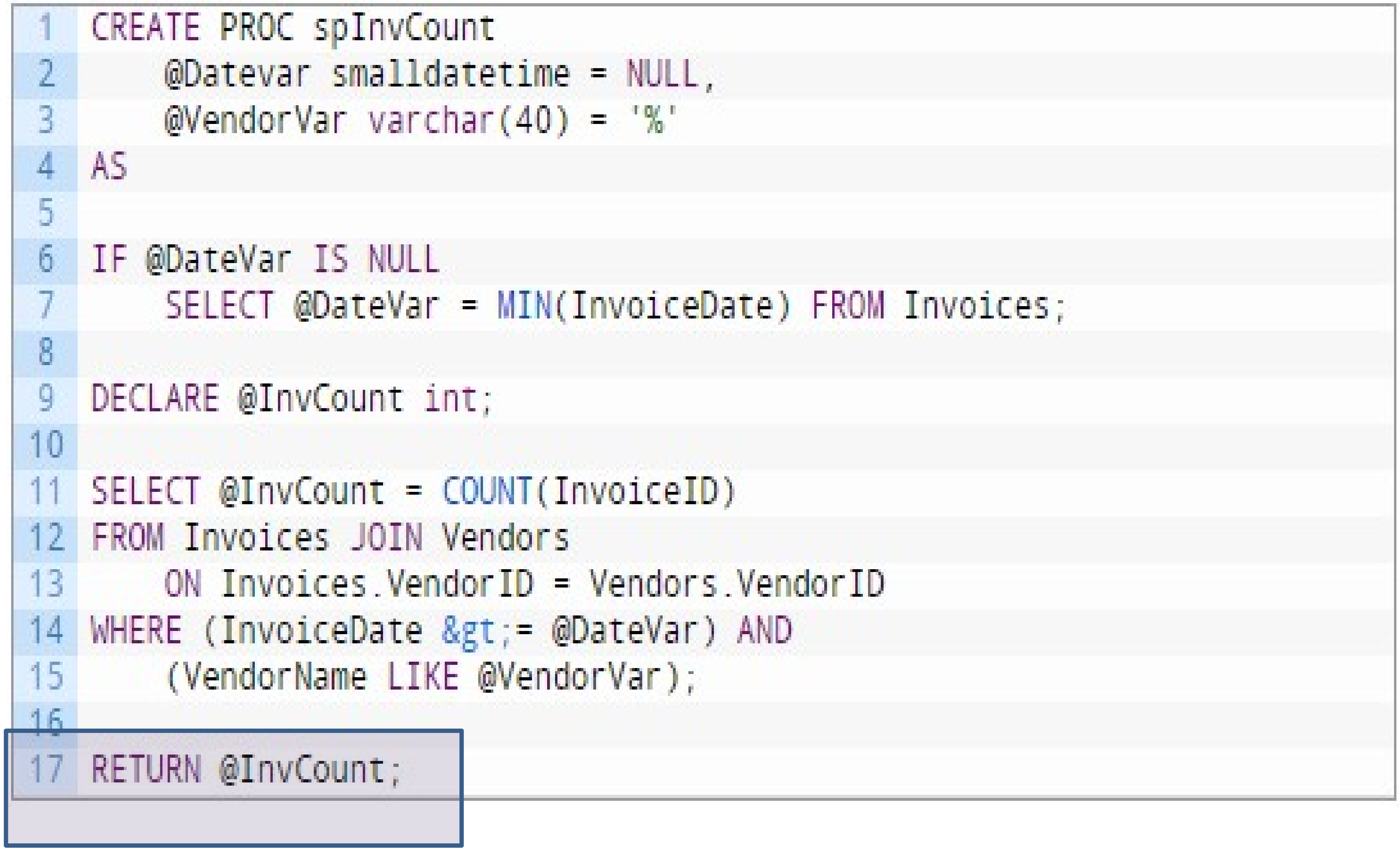
Source: http://www.w3computing.com/

Example: spInvCount

- The stored procedure named spInvCount returns a count of the number of invoices that meet the conditions specified by the input parameters
- These parameters are identical to the input parameters used by the stored procedure

Source: http://www.w3computing.com/

Example of Stored Proc w/ Return Val

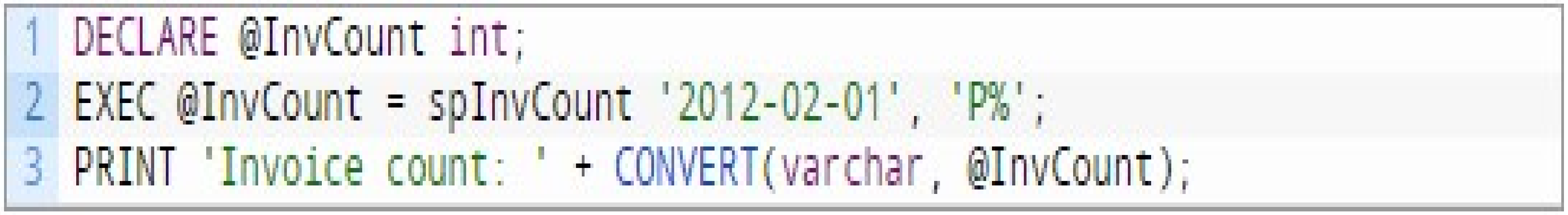


```
CREATE PROC spInvCount
    @Datevar smalldatetime = NULL,
    @VendorVar varchar(40) = '%'
AS

IF @DateVar IS NULL
    SELECT @DateVar = MIN(InvoiceDate) FROM Invoices;

DECLARE @InvCount int;

SELECT @InvCount = COUNT(InvoiceID)
FROM Invoices JOIN Vendors
    ON Invoices.VendorID = Vendors.VendorID
WHERE (InvoiceDate &gt;= @DateVar) AND
    (VendorName LIKE @VendorVar);

RETURN @InvCount;
```

Source: http://www.w3computing.com/

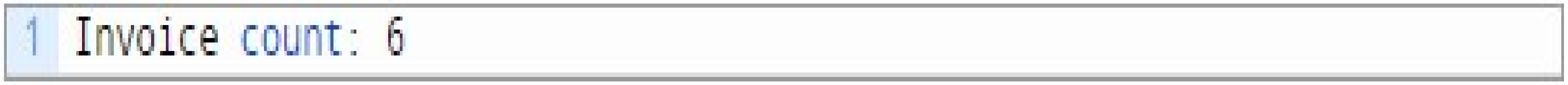
Example – Cont'd

- Script that calls the stored procedure:



```
DECLARE @InvCount int;
EXEC @InvCount = spInvCount '2012-02-01', 'P%';
PRINT 'Invoice count: ' + CONVERT(varchar, @InvCount);
```

The response from the system



```
Invoice count: 6
```

Source: http://www.w3computing.com/

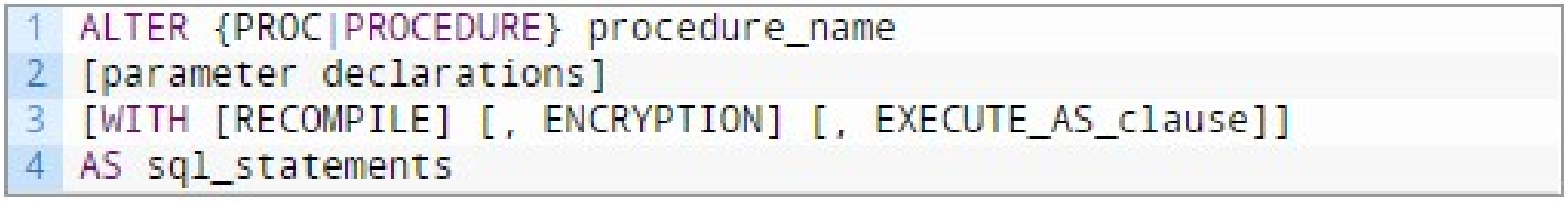
Delete Stored Procedure

- **DROP PROC** is used to delete one or more stored procedures from the database
  - Deletion is permanent
- **Syntax:**

DROP {PROC|PROCEDURE} procedure_name [,…]

Change a Stored Procedure

- **ALTER PROC** is used to redefine an existing stored procedure
  - **Completely replace the previous definition for the stored definition**; <u>*Best practice, to just delete the stored procedure and then recreate it*</u>
  - Exception: In the case of security permission, if you have assigned setting to restrict the user who can call the procedure; permissions are lost when you delete the procedure; **Best option is to use ALTER PROC instead**

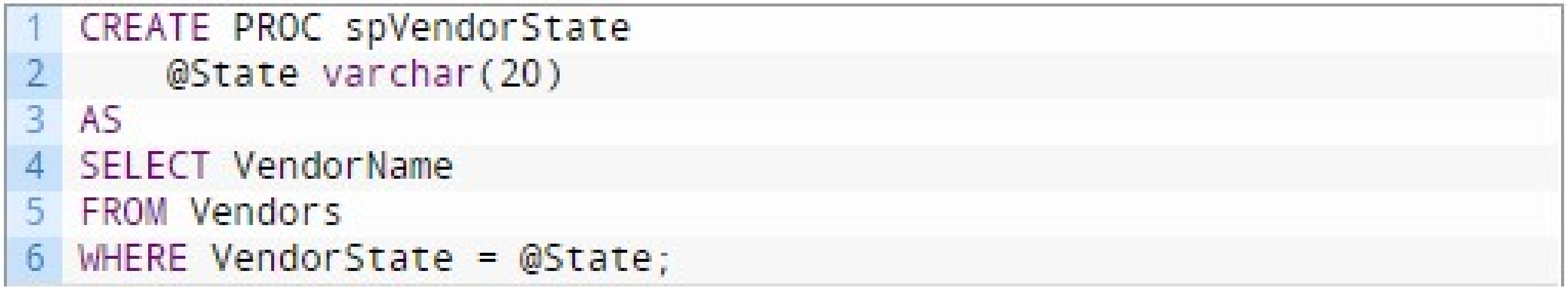


```
ALTER {PROC|PROCEDURE} procedure_name
[parameter declarations]
[WITH [RECOMPILE] [, ENCRYPTION] [, EXECUTE_AS_clause]]
AS sql_statements
```

Source: http://www.w3computing.com/

Example

- Create spVendorState Procedure

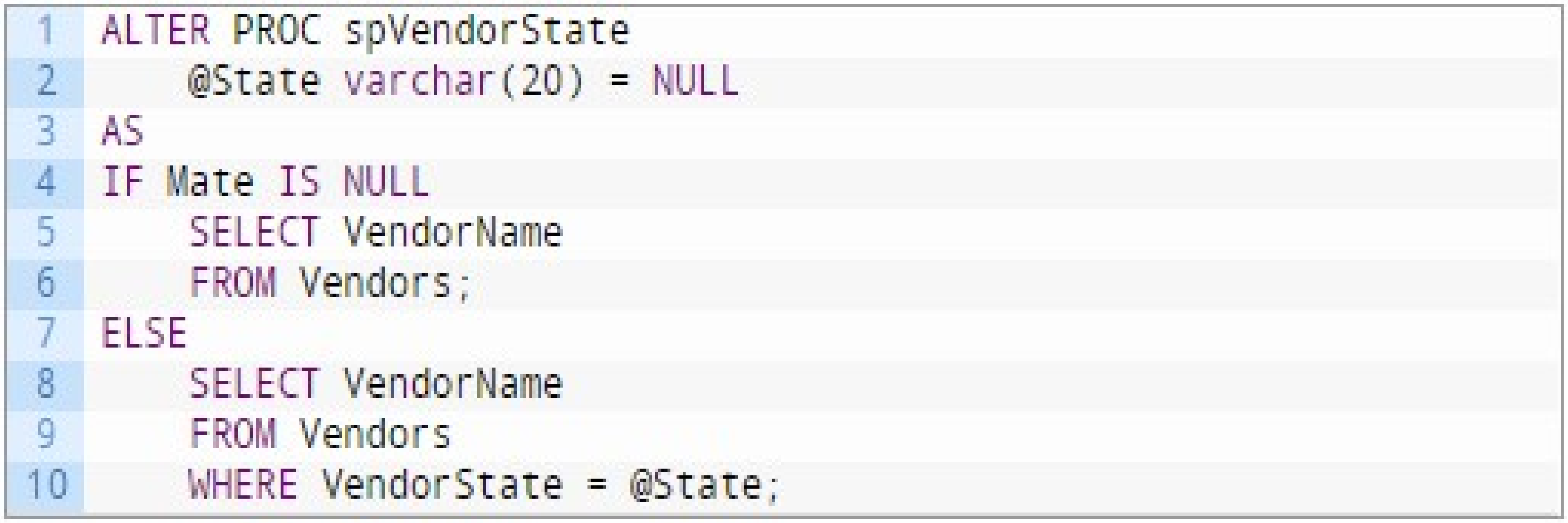


```sql
CREATE PROC spVendorState
    @State varchar(20)
AS
SELECT VendorName
FROM Vendors
WHERE VendorState = @State;
```

Source: *http://www.w3computing.com/*

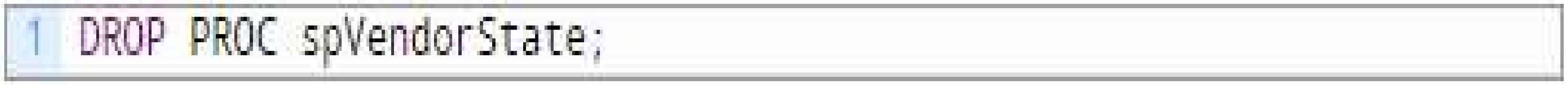
Example of ALTER PROC & DROP PROC

- Statement that changes the parameter defined by the procedure



```
ALTER PROC spVendorState
    @State varchar(20) = NULL
AS
IF Mate IS NULL
    SELECT VendorName
    FROM Vendors;
ELSE
    SELECT VendorName
    FROM Vendors
    WHERE VendorState = @State;
```

- Statement that delete the procedure



```
DROP PROC spVendorState;
```

References

- http://www.w3computing.com/sqlserver/create-stored-procedure/
- http://www.tutorialspoint.com/listtutorial/An-Introduction-to-Stored-Procedures-in-SQL-2012/4702
- http://www.tutorialspoint.com/listtutorial/How-to-Create-a-Stored-Procedure-in-SQL-2012/4703
- http://technet.microsoft.com/en-us/library/aa174792(v=sql.80).aspx
- http://www.mssqltips.com/sqlservertutorial/160/sql-server-stored-procedure/
- http://www.w3schools.com/sql/sql_groupby.asp

Source: http://www.w3computing.com/