

Creating Programmatic SQL Database Objects

Lab 1 – Implementing Stored Procedures

# Overview

In this lab, you need to create a set of stored procedures in the **AdventureWorksLT** database to support a new reporting application. The procedures will be created within a new Reports schema.

Before starting this lab, you should view **Module 1 – Creating Stored Procedures** in the course *Creating Programmatic SQL Database Objects*. Then, if you have not already done so, follow the instructions in the **Getting Started** document for this course to set up the lab environment.

If you find some of the challenges difficult, don’t worry – you can find suggested solutions for all of the challenges in the **Lab Solution** folder for this module.

# What You’ll Need

To complete the labs, you will need the following:

* An Azure SQL Database instance with the AdventureWorksLT sample database. Review the Getting Started document for information about how to provision this.
* The lab files for this course

# Setup

# Using SQL Server Management Studio, connect to the AdventureWorksLT database.

# Open Lab1Setup.sql from the Setup folder for this course and run the following Transact-SQL:

CREATE SCHEMA Reports;

GO

# Challenge 1: Create Stored Procedures

In this exercise, you will create two stored procedures to support one of the new reports.

## Review the Reports.GetProductColors Stored Procedure Specification

Review the following design requirements for your stored procedure:

|  |  |
| --- | --- |
| Stored Procedure: | Reports.GetProductColors |
| Input Parameters: | None |
| Output Parameters: | None |
| Output Columns: | Color (from SalesLT.Product) |
| Notes: | Colors should not be returned more than once in the output. NULL values should not be returned. |

## Create the Reports.GetProductColors Stored Procedure

Design, implement, and execute the stored procedure in accordance with the design specifications.

## Review the Reports.GetProductsAndModels Stored Procedure Specification

1. Review the following design requirements for your stored procedure:

|  |  |
| --- | --- |
| Stored Procedure: | Reports.GetProductsAndModels |
| Input Parameters: | None |
| Output Parameters: | None |
| Output Columns: | ProductID, Name, ProductNumber, SellStartDate, SellEndDate and Color (from SalesLT.Product), ProductModelID (from SalesLT.ProductModel), Description (from SalesLT.ProductDescription). |
| Output Order: | ProductID, ProductModelID |
| Notes: | For descriptions, return the Description column from the SalesLT.ProductDescription table. |

## Create the Reports.GetProductsandModels Stored Procedure

1. Design, implement, and execute the stored procedure in accordance with the design specifications.

# Challenge 2: Create Parameterized Stored Procedures

In this exercise, you will create a stored procedure to support one of the new reports.

## Review the Reports.GetProductsByColor Stored Procedure specification

1. Review the following design requirements for your stored procedure:

|  |  |
| --- | --- |
| Stored Procedure | Reports.GetProductsByColor |
| Input parameters | @Color (same data type as the Color column in the Production.Product table) |
| Output parameters | None |
| Output columns | ProductID, Name, ListPrice (returned as a column named Price), Color, and Size (from Production.Product) |
| Output order | Name |
| Notes | The procedure should return products that have no Color if the parameter is NULL. |

## Create the Reports.GetProductsByColor Stored Procedure

1. Design and create the Reports.GetProductsByColor stored procedure.
2. Execute the Reports.GetProductsByColor stored procedure with a color of ‘Blue’.
3. Execute the Reports.GetProductsByColor stored procedure with a color of NULL.