

Developing SQL Databases

Lab 5 – Creating Views

# Overview

A new web-based stock promotion is being tested at the Adventure Works Bicycle Company. Your manager is worried that providing access from the web-based system directly to the database tables will be insecure, so she has asked you to design some views that the web-based system will use.

The Sales department has also asked you to create a view that enables a temporary worker to enter new customer data without viewing credit card, email address, or phone number information.

Before starting this lab, you should view **Module 5 – Implementing Views** in the course *Developing SQL Databases*. 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

# Challenge 1: Create Standard Views

The web-based stock promotion requires two new views: OnlineProducts and Available Models. The documentation for each view is shown in the following tables:

View 1: OnlineProducts

|  |  |
| --- | --- |
| View Column | Table Column |
| ProductID | SalesLT.Product,ProductID |
| Name | SalesLT.Product,Name |
| Product Number | SalesLT.Product,ProductNumber |
| Color | SalesLT.Product.Color. If NULL, return ‘N/A’ |
| Size | SalesLT.Product.Size |
| Price | SalesLT.Product.ListPrice |
| Weight | SalesLT.Product.Weight |

This view is based on the SalesLT.Product table. Products should be displayed only if the product is on sale, which can be determined using the SellStartDate and SellEndDate columns.

View 2: Available Models

|  |  |
| --- | --- |
| View Column | Table Column |
| Product ID | SalesLT.Product.ProductID |
| Product Name | SalesLT.Product.Name |
| Product Model ID | SalesLT.ProductModel.ProductModelID |
| Product Model | SalesLT.ProductMode.Name |

This view is based on two tables: SalesLT.Product and SalesLT.ProductModel. Products should be displayed only if the product is on sale, which can be determined using the SellStartDate and SellEndDate columns.

## Design and Implement the Views

## Review the documentation for the new views.

## Using SSMS, connect to AdventureWorksLT.

## Open a new query window.

1. Write and execute scripts to create the new views.

# Challenge 2: Create an Updateable View

The Sales department has asked you to create an updateable view based on the SalesLT.Customer table, enabling a temporary worker to enter a batch of new customers while keeping the credit card, email and phone number information secure.

The view must contain five columns from the SalesLT.Customer table: CustomerID, FirstName, LastName, PasswordHash, and PasswordSalt. You must be able to update the view with new customers.

|  |  |
| --- | --- |
| View Columns | Table Columns |
| CustomerID | SalesLT.Customer.CustomerID |
| FirstName | SalesLT.Customer.FirstName |
| LastName | SalesLT.Customer.LastName |
| PasswordHash | SalesLT.Customer.PasswordHash |
| PasswordSalt | SalesLT.Customer.PasswordSalt |

## Design and Implement the Updateable View

## Review the requirements for the updateable view.

1. Write and execute a script to create the new view.

## Test the Updateable View

## Write and execute a SELECT query to check that the view returns the correct columns. Order the result set by CustomerID.

## Write and execute an INSERT statement to add a new record to the view with the following values:

|  |  |  |  |
| --- | --- | --- | --- |
| FirstName | LastName | PasswordHash | PasswordSalt |
| Ed | Kish | Uw8sEe4ZGPvigEQEiSJ57Bd77SB77S | cjsKU4w= |

## Check that the new record appears in the view results.

1. Close SSMS without saving any changes