

Module 1: Introduction to Databases

Lab: Exploring and Querying SQL Server Databases

Scenario

You have recently started working as a database developer at Adventure Works, manufacturers and retailers of bicycles and associates products. Your line manager has asked you to familiarize yourself with the databases that you will be working with, and has given your user account the required permissions to do so. You intend to create database diagrams of parts of the OLTP database and data warehouse database, and examine the table

Objectives

After completing this lab, you will be able to:

- Create database diagrams to explore databases.
- Write Transact-SQL SELECT statements that include a WHERE clause.

Lab Setup

Estimated Time: 45 minutes

Before you start this lab:

1. Ensure that the 10985C-MIA-DC and 10985C-MIA-SQL virtual machines are both running.
2. Log on to 10985C-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Ps\$\$w0rd**.
3. Browse to the **D:\Labfiles\Lab01\Starter** folder, and then run **Setup.cmd** as Administrator.

Exercise 1: Exploring an OLTP Schema and a Data Warehouse Schema

Scenario

You will begin your exploration of the databases by creating two database diagrams, one for the Adventure Works OLTP database, and one for the Adventure Works data warehouse.

Task 1: Explore an OLTP Database Schema

1. Open SQL Server Management Studio and connect to the MIA-SQL Database Engine instance by using Windows Authentication.
2. In the **AdventureWorks2016** database, create a database diagram and then add the following tables to it:
 - **Address (Person)**
 - **Customer (Sales)**
 - **SalesOrderDetail (Sales)**
 - **SalesOrderHeader (Sales)**

- **SalesPerson (Sales)**
- **SalesTerritory (Sales)**
- **ShipMethod (Purchasing)**

3. Examine the primary key columns and the foreign key relationships between the tables in the diagram, and note which columns are involved in the relationships.
4. Save the database diagram as **AdventureWorks Diagram**, and then close the diagram window.

Task 2: Explore a Data Warehouse Schema

1. In the **AdventureWorksDW2016** database, create a database diagram and add then the following tables to it:
 - **DimCustomer**
 - **DimDate**
 - **DimProduct**
 - **DimProductCategory**
 - **DimProductSubcategory**
 - **FactInternetSales**
2. Examine the primary key columns and the foreign key relationships between the tables in the diagram, and note which columns are involved in the relationships.
3. Save the database diagram as **AdventureWorks Data Warehouse Diagram**, and then close the diagram window.

Results: In this exercise, you created a database diagram in the AdventureWorks2016 database and a database diagram in the AdventureWorksDW2016 database.

Exercise 2: Querying a Database by Using Transact-SQL

Scenario

You will now continue your exploration of the Adventure Works OLTP database by writing and executing SELECT statements and examining the result sets.

Task 1: Write Transact-SQL SELECT Statements

1. In SQL Server Management Studio, open a new query window connected to the **AdventureWorks2016** database.
2. Write and execute a Transact-SQL query that returns all columns and all rows from the **Sales.SalesOrderHeader** table.
3. In the same query window, under the existing query, write and execute a Transact-SQL statement that returns the **SalesOrderID**, **OrderDate**, **SalesPersonID** columns and all rows from the **Sales.SalesOrderHeader** table.
4. Verify that both queries return the same number of rows.

Task 2: Write Transact-SQL SELECT Statements with a WHERE Clause

1. Under the existing query, write and execute a Transact-SQL statement that returns the **SalesOrderID**, **OrderDate**, **SalesPersonID** columns from the **Sales.SalesOrderHeader** table, and use a WHERE clause to return only the rows with a **SalesPersonID** value of **279**.
2. Review the results and note the number of rows that the query returned.
3. Under the existing query, write and execute a Transact-SQL statement that returns the **SalesOrderID**, **OrderDate**, **SalesPersonID** columns from the **Sales.SalesOrderHeader** table, and use a WHERE clause to return only the rows with a **SalesPersonID** value of **279** or **282**.
4. Review the results and note the number of rows that the query returned.
5. Under the existing query, write and execute a Transact-SQL statement that returns the **SalesOrderID**, **OrderDate** columns from the **Sales.SalesOrderHeader** table, and use a WHERE clause to return only the rows with a **SalesOrderID** value that is between **57000** and **58000**.
6. Review the results and note the number of rows that the query returned. Verify that the second query returned more rows.
7. Under the existing query, write and execute a Transact-SQL statement that returns the **SalesOrderID**, **OrderDate** columns from the **Sales.SalesOrderHeader** table, and use a WHERE clause to return only the rows with a **SalesPersonID** value of **279** and an **OrderDate** value that includes the year **2014**.
8. Review the results and note the number of rows that the query returned.
9. Close SQL Server Management Studio without saving changes.

Results: In this exercise, you:

- Wrote and executed SELECT statements to retrieve all columns and to retrieve specific columns from a table in the Adventure Works OLTP database.
- Wrote and executed SELECT statements that include a WHERE clause to filter the rows that are returned from a table in the Adventure Works OLTP database.

Lab Review

Question Why did the number of rows returned by the queries that you wrote in the lab vary?

©2016 Microsoft Corporation. All rights reserved.

The text in this document is available under the [Creative Commons Attribution 3.0 License](#), additional terms may apply. All other content contained in this document (including, without limitation, trademarks, logos, images, etc.) are **not** included within the Creative Commons license grant. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. Microsoft makes no warranties, express or implied, with respect to the information provided here.