Module 1: SQL Server Security
Lab: Authenticating Users

Exercise 1: Create Logins

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab01\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for the script to finish.

▶ Task 2: Verify the Authentication Mode

1. Start SQL Server Management Studio, and connect to the **MIA-SQL** database engine using Windows authentication.
2. In Object Explorer, right-click the **MIA-SQL** instance, and click then **Properties**.
3. In the **Server Properties - MIA-SQL** dialog box, on the **Security** page, verify that **SQL Server and Windows Authentication mode** is selected, and then click **OK**.

▶ Task 3: Create Logins Based on Windows Authentication

1. In Object Explorer, under **MIA-SQL**, expand **Security**, and expand **Logins** to view the existing logins in the instance.
2. Right-click **Logins**, and click **New Login**.
3. In the **Login - New** dialog box, on the **General** page, in the **Login name** box, type **ADVENTUREWORKS\WebApplicationSvc**.
4. Ensure **Windows authentication** is selected.
5. In the **Default database** list, click **AdventureWorks**, and then click **OK**.
6. In Object Explorer, right-click **Logins**, and then click **New Login**.
7. In the **Login - New** dialog box, on the **General** page, click **Search**.
8. In the **Select User or Group** dialog box, click **Object Types**.
9. In the **Object Types** dialog box, select **Groups**, and then click **OK**.
10. In the **Select User or Group** dialog box, click **Locations**.
11. In the **Locations** dialog box, expand **Entire Directory**, click **adventurework.msft**, and then click **OK**.
12. In the **Select User, Service Account, or Group** dialog box, in the **Enter the object name to select** box, type **IT_Support**, click **Check Names**, and then click **OK**.
13. In the **Login - New** dialog box, ensure that **Windows authentication** is selected.
14. In the **Default database** list, click **AdventureWorks**, and then click **OK**.

▶ Task 4: Create Logins Based on SQL Server Authentication

1. Right-click **Logins**, and click **New Login**.
2. In the **Login - New** dialog box, on the **General** page, in the **Login name** box, type **SalesSupport**.

3. Click **SQL Server authentication**, and in the **Password** and **Confirm Password** boxes, type **Pa55w.rd**.
4. Confirm that **Enforce password policy** is selected. Clear the **Enforce password expiration** check box. The **User must change password at next login** check box will automatically be cleared.
5. In the **Default database** list, click **AdventureWorks**, and then click **OK**.
6. Leave SQL Server Management Studio open for the next exercise.

Results: After this exercise, you should have verified the authentication modes supported by the **MIA-SQL** instance, and created three logins.

Exercise 2: Create Database Users

▶ Task 1: Use Automatic User Creation and Mapping

1. In SQL Server Management Studio, in Object Explorer, under **MIA-SQL**, under **Security**, under **Logins**, right-click **ADVENTUREWORKS\WebApplicationSvc**, and then click **Properties**.
2. In the **Login Properties - ADVENTUREWORKS\WebApplicationSvc** dialog box, on the **User Mapping** page, in the **Users mapped to this login** section, in the **AdventureWorks** database row, select the **Map** check box, and then click **OK**.
3. In Object Explorer, under **MIA-SQL** expand **Databases**, expand **AdventureWorks**, expand **Security**, and then expand **Users**. Notice that a user called **ADVENTUREWORKS\WebApplicationSvc** has been created in the database.

▶ Task 2: Create a User and Map It to a Login

1. In the **AdventureWorks** database, under **Security**, right-click **Users**, and then click **New User**.
2. In the **Database User - New** dialog box, on the **General** page, verify that **SQL user with login** is selected.
3. In the **User name** box, type **ServiceUser**.
4. In the **Login name** box, type **SalesSupport**, and then click **OK**.
5. Under **MIA-SQL**, under **Security**, under **Logins**, right-click **SalesSupport**, and then click **Properties**.
6. In the **Login Properties - SalesSupport** dialog box, on the **User Mapping** page, verify that the login is mapped to the **ServiceUser** user in **AdventureWorks**, and the default schema is **dbo**, and then click **OK**.

▶ Task 3: Create a User Using Transact-SQL

1. In SQL Server Management Studio, on the toolbar, click **New Query**.
2. In the query window, type the following Transact-SQL statement:

```
USE AdventureWorks;
GO
CREATE USER [ITSupport] FOR LOGIN [ADVENTUREWORKS\IT_Support] WITH
DEFAULT_SCHEMA=[dbo]
GO
```

3. Click **Execute**.

4. In Object Explorer, in the **AdventureWorks** database, under **Security**, right-click **Users** then click **Refresh**, verify that the **ITSupport** user appears.

Results: At the end of this exercise, you will have created three database users and mapped them to the logins you created in the previous exercise.

Exercise 3: Correct Application Login Issues

▶ Task 1: Carry Out an Initial Test

1. On the Start page, type **cmd**, and then press Enter to start an instance of the Windows Command Prompt.
2. At the Command Prompt, type the following command, and then press Enter:

```
Sqlcmd -S MIA-SQL -U LegacySalesLogin -P tOps3cr3t
```

3. Notice that the error message presented to the **sqlcmd** is generic, reporting that login failed but giving no further details.
4. In SQL Server Management Studio, in Object Explorer, expand **Management**, expand **SQL Server Logs**, and then double-click the log file whose name begins **Current**.
5. In the **Log File Viewer - MIA SQL** dialog box, in the right-hand pane, look for the topmost log entry that begins **Login failed for user 'LegacySalesLogin'**. The error message states that there was a problem evaluating the login's password.
6. Notice that the next line in the log file contains the following error number:

```
Error: 18456, Severity: 14, State: 7.
```

The documentation for error 18456 indicates that a **State** value of **7** is caused when:

```
Login is disabled, and the password is incorrect.
```

7. In the **Log File Viewer - MIA SQL** dialog box, click **Close**.

The login cannot connect because the account is disabled, and the wrong password is being used.

▶ Task 2: Enable the Login

1. In Object Explorer, under **MIA-SQL**, under **Security**, under **Logins**, right-click **LegacySalesLogin**, and then click **Properties**.
2. In the **Login Properties - LegacySalesLogin** dialog box, on the **Status** page, click **Enabled**, and then click **OK**.
3. In Command Prompt, type the following command, and then press Enter:

```
Sqlcmd -S MIA-SQL -U LegacySalesLogin -P tOps3cr3t
```

4. Notice that the error message presented to the **sqlcmd** is generic, reporting that the login failed but giving no further details.
5. In SQL Server Management Studio, in Object Explorer, under **Management**, under **SQL Server Logs**, double-click the log file whose name begins **Current**.

6. In the **Log File Viewer - MIA SQL** dialog box, in the right-hand pane, look for the topmost log entry that begins **Login failed for user 'LegacySalesLogin'**. Read the rest of the entry to determine the cause of the login failure. Notice that the login failed because the password was not correct.
7. In the **Log File Viewer - MIA SQL** dialog box, click **Close**.

▶ Task 3: Change the Login Password

1. In Object Explorer, under **MIA-SQL**, under **Security**, under **Logins**, right-click **LegacySalesLogin**, and then click **Properties**.
2. In the **Login Properties - LegacySalesLogin** dialog box, on the **General** page, in the **Password** and **Confirm password** boxes, type **t0ps3cr3t**, and then click **OK**.
3. In Command Prompt, type the following command, and then press Enter:

```
Sqlcmd -S MIA-SQL -U LegacySalesLogin -P t0ps3cr3t
```

4. Notice that the error message indicates that the default database cannot be opened.

▶ Task 4: Change the Default Database

1. In Object Explorer, under **MIA-SQL**, under **Security**, under **Logins**, right-click **LegacySalesLogin**, then click **Properties**.
2. In the **Login Properties - LegacySalesLogin** dialog box, on the **General** page, in the **Default database** list, click **AdventureWorks**.
3. On the **User Mapping** page, in the **Users mapped to this login** section, on the row for the **AdventureWorks** database, select the **Map** check box, and then click **OK**.
4. In Command Prompt, type the following command, and then press Enter:

```
Sqlcmd -S MIA-SQL -U LegacySalesLogin -P t0ps3cr3t
```

5. Notice that the login attempt is successful.
6. Close the command prompt window, but leave SQL Server Management Studio open for the next exercise.

Results: After completing this lab, you will be able to:

Correct application login issues.

Exercise 4: Configure Security for Restored Databases

▶ Task 1: Verify Database Users

- In Object Explorer, under **MIA-SQL**, under **Databases**, expand **InternetSales**, expand **Security**, expand **Users**, and note that database users with the following names exist in the database:
  - **ADVENTUREWORKS\WebApplicationSvc**
  - **InternetSalesApplication**

▶ Task 2: Run the Orphaned Users Report

1. On the toolbar, click **New Query**.
2. In the query pane, type the following commands:

```
USE InternetSales
EXEC sp_change_users_login 'Report';
GO
```

3. Select the query you have typed, and click **Execute**.
4. The **InternetSalesApplication** user is reported as orphaned.

▶ Task 3: Repair the Orphaned User

1. On the toolbar, click **New Query**.
2. In the query pane, type the following commands:

```
USE InternetSales
EXEC sp_change_users_login 'Auto_Fix', 'InternetSalesApplication', NULL, NULL
GO
```

3. Select the query you have typed, and click **Execute**. In the output of the query reports, notice that one orphaned user was fixed by updating it.
4. On the toolbar, click **New Query**.
5. In the query pane, type the following commands:

```
USE InternetSales
EXEC sp_change_users_login 'Report';
GO
```

6. Select the query you have typed, and click **Execute**. Notice that no orphaned users are reported.
7. Close SQL Server Management Studio without saving any changes.

Results: At the end of this exercise, the **ADVENTUREWORKS\WebApplicationSvc** and **InternetSalesApplication** logins will have access to the **InternetSales** database.

Module 2: Assigning Server and Database Roles

Lab: Assigning Server and Database Roles

Exercise 1: Assigning Server Roles

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab02\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog, click **Yes** when prompted to confirm that you want to run the command file, and wait for the script to finish.

▶ Task 2: Create a Server Role

1. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, navigate to **D:\Labfiles\Lab02\Starter\Project**, click **Project.ssmssln**, and then click **Open**.
4. In Solution Explorer, expand **Queries**, and then double-click **Lab Exercise 01 - server roles.sql**.
5. Under the heading for **Task 1**, type the following code::

```
CREATE SERVER ROLE database_manager;
```

6. Highlight the code you have typed and click **Execute**.

▶ Task 3: Assign Server-Level Permissions

1. Edit the code under the heading for **Task 2** so that it reads:

```
GRANT ALTER ANY LOGIN TO database_manager;
GRANT VIEW ANY DATABASE TO database_manager;
```

2. Highlight the query you have amended and click **Execute**.

▶ Task 4: Assign Role Membership

1. Under the heading for **Task 3**, type the following code:

```
ALTER SERVER ROLE database_manager ADD MEMBER [ADVENTUREWORKS\Database_Managers];
```

2. Highlight the code you have typed and click **Execute**.
3. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have created the **database_manager** server role, granted permissions to members to alter any login and alter any database, and granted membership to the members of the **Database_Managers** login.

Exercise 2: Assigning Fixed Database Roles

▶ Task 1: Create a Database User and Assign Fixed Database Roles

1. In SQL Server Management Studio, in Object Explorer, expand **Security**, expand **Logins**, right-click **ADVENTUREWORKS\Database_Managers**, and then click **Properties**.
2. In the **Login Properties - ADVENTUREWORKS\Database_Managers** dialog box, on the **User Mapping** page, in the **Users mapped to this login** table, in the **salesapp1** database row, select the **Map** check box.
3. In the **Database role membership for: salesapp1** list, select the **db_accessadmin** and **db_backupoperator** check boxes, and then click **OK**.
4. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have mapped the **Database_Managers** login to the **salesapp1** database and added them to the **db_backupoperator** and **db_accessadmin** roles.

Exercise 3: Assigning User-Defined Database Roles

▶ Task 1: Create a Database Principal in SSMS

1. In SQL Server Management Studio, in Object Explorer, expand **Databases**, expand **salesapp1**, and then expand **Security**.
2. Right-click **Users**, and then click **New User**.
3. In the **Database User - New** dialog box, on the **General** page, ensure that the **User type** box has the value **SQL user with login**.
4. In the **User name** box, type **internetsales_user**, in the **Login name** box, type **ADVENTUREWORKS\InternetSales_Users**, and then click **OK**.

▶ Task 2: Create a User-Defined Database Role in SSMS

1. In Object Explorer, under **salesapp1**, under **Security**, right-click **Roles**, point to **New**, and then click **New Database Role**.
2. In the **Database Role - New** dialog box, on the **General** page, in the **Role name** box, type **sales_reader**, and then click **Add**.
3. In the **Select Database User or Role** dialog box, click **Browse**.
4. In the **Browse for Objects** dialog box, select the **[internetsales_user]** check box, and then click **OK**.
5. In the **Select Database User or Role** dialog box, click **OK**.
6. In the **Database Role - New** dialog box, on the **Securables** page, click **Search**.
7. In the **Add Objects** dialog box, click **Specific objects**, and then click **OK**.
8. In the **Select Objects** dialog box, click **Object Types**.
9. In the **Select Object Types** dialog box, select the **Schemas** check box, then click **OK**.
10. In the **Select Objects** dialog box, click **Browse**.
11. In the **Browse for Objects** dialog box, select the **[Sales]** check box, then click **OK**.
12. In the **Select Objects** dialog box, click **OK**.

13. In the **Database Role - New** dialog box, in the **Permissions for Sales** section, on the **Explicit** tab, in the **Select** row, select the **Grant** check box, and then click **OK**.

▶ Task 3: Create a Database Principal by Using Transact-SQL

1. In Solution Explorer, double-click the query **Lab Exercise 03 - database roles.sql**.
2. Highlight the code under the heading for **Task 3** and click **Execute**.

▶ Task 4: Create User-Defined Database Roles by Using Transact-SQL

1. Under the heading for **Task 4**, type the following code:

```
CREATE ROLE production_reader;
CREATE ROLE sales_order_writer;
GO
```

2. Highlight the code you have just typed and click **Execute**.

▶ Task 5: Grant Permissions to User-Defined Database Roles Using Transact-SQL

1. Under the heading for **Task 5**, type the following code:

```
GRANT SELECT ON SCHEMA::Production TO production_reader;
GO
GRANT UPDATE ON Sales.OrderDetails TO sales_order_writer;
GO
GRANT UPDATE ON Sales.Orders TO sales_order_writer;
GO
```

2. Highlight the query you have typed and click **Execute**.

▶ Task 6: Add a Database Principal to a Role Using Transact-SQL

1. Under the heading for **Task 6**, edit the query so that it reads:

```
ALTER ROLE sales_reader ADD MEMBER internetsales_manager;
ALTER ROLE production_reader ADD MEMBER internetsales_manager;
ALTER ROLE sales_order_writer ADD MEMBER internetsales_manager;
GO
```

2. Highlight the query you have typed and click **Execute**.
3. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have created user-defined database roles and assigned them to database principals.

Exercise 4: Verifying Security

▶ Task 1: Test IT Support Permissions

1. Minimize SQL Server Management Studio.
2. Click **Start**, type **cmd**, and then press Enter.
3. At the command prompt, type the following command (which opens the **sqlcmd** utility as **ADVENTUREWORKS\AnthonyFrizzell**), and then press Enter:

```
runas /user:adventureworks\anthonyfrizzell /noprofile sqlcmd
```

4. When you are prompted for a password, type **Pa55w.rd**, and then press Enter. Wait for the connection to succeed and the SQLCMD window to open.
5. In the SQLCMD window, at the command prompt, type the following commands to verify your identity, and then press Enter:

```
SELECT SUSER_NAME();
GO
```

   Note that SQL Server identifies Windows group logins using their individual user account, even though there is no individual login for that user. **ADVENTUREWORKS\AnthonyFrizzell** is a member of the **ADVENTUREWORKS\IT_Support** global group, which is in turn a member of the **ADVENTUREWORKS\Database_Managers** domain local group for which you created a login.
6. In the SQLCMD window, at the command prompt, type the following commands to alter the password of the **Marketing_Application** login, and then press Enter:

```
ALTER LOGIN Marketing_Application WITH PASSWORD = 'NewPa55w.rd';
GO
```

7. In the SQLCMD window, at the command prompt, type the following commands to disable the **ADVENTUREWORKS\WebAplicationSvc** login, and then press Enter:

```
ALTER LOGIN [ADVENTUREWORKS\WebApplicationSvc] DISABLE;
GO
```

8. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.
9. In SQL Server Management Studio, in Object Explorer, under **MIA-SQL**, under **Security**, right-click **Logins**, and then click **Refresh**.
10. Under **Logins**, right-click **ADVENTUREWORKS\WebApplicationSvc**, and then click **Properties**.
11. In the **Login Properties - ADVENTUREWORKS\WebApplicationSvc** dialog box, on the **Status** page, click **Enabled**, and then click **OK** to re-enable the login.

▶ Task 2: Test Sales Employee Permissions

1. In the command prompt window, type the following command to run **sqlcmd** as **ADVENTUREWORKS\DanDrayton**, and then press Enter. This user is a member of the **ADVENTUREWORKS\Sales_NorthAmerica** global group, which is in turn a member of the **ADVENTUREWORKS\InternetSales_Users** domain group:

```
runas /user:adventureworks\dandrayton /noprofile sqlcmd
```

2. At the command prompt, when you are prompted for a password, type **Pa55w.rd**, and then press Enter.
3. In the SQLCMD window, at the command prompt, type the following commands to query the **Sales.Orders** table in the **salesapp1** database table, and then press Enter:

```
SELECT TOP (10) * FROM salesapp1.Sales.Orders;
GO
```

4. Verify that the user can query the **Sales.Orders** table.

5. In the SQLCMD window, at the command prompt, type the following commands to update the **Sales.Orders** table in the **salesapp1** database, and then press Enter:

```
UPDATE salesapp1.Sales.Orders SET shippeddate = getdate() WHERE orderid = 10257;
GO
```

6. Verify that the user does **NOT** have UPDATE permission on the **Sales.Orders** table.
7. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.

▶ Task 3: Test Sales Manager Permissions

1. In the command prompt window, at the command prompt, type the following command to run **sqlcmd** as **ADVENTUREWORKS\ DeannaBall**, and then press Enter. This user is a member of the **ADVENTUREWORKS\Sales_Managers** global group, which is in turn a member of the **ADVENTUREWORKS\InternetSales_Managers** domain group:

```
runas /user:adventureworks\deannaball /noprofile sqlcmd
```

2. At the command prompt, when you are prompted for a password, type **Pa55w.rd**, and then press Enter.
3. In the SQLCMD window, at the command prompt, type the following commands to query the **Sales.Orders** table in the **salesapp1** database table, and then press Enter:

```
SELECT TOP (10) * FROM salesapp1.Sales.Orders;
GO
```

4. Verify that the user can query the **Sales.Orders** table.
5. In the SQLCMD window, at the command prompt, type the following commands to query the **Production.Suppliers** table in the **salesapp1** database table, and then press Enter:

```
SELECT TOP (10) * FROM salesapp1.Production.Suppliers;
GO
```

6. Verify that the user can query the **Production.Suppliers** table.
7. In the SQLCMD window, at the command prompt, type the following commands to update the **Sales.Orders** table in the **salesapp1** database, and then press Enter:

```
UPDATE salesapp1.Sales.Orders SET shippeddate = getdate() WHERE orderid = 10257;
GO
```

8. Verify that the user has UPDATE permissions on the **Sales.Orders** table.
9. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.
10. In the Command Prompt window, at the command prompt, type **exit**, and then press Enter.
11. Close SQL Server Management Studio without saving any changes.

Results: At the end of this exercise, you will have verified your new security settings.

Module 3: Authorizing Users to Access Resources

Lab: Authorizing Users to Access Resources

Exercise 1: Granting, Denying, and Revoking Permissions on Objects

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab03\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog, click **Yes** when prompted to confirm that you want to run the command file, and wait for the script to finish.

▶ Task 2: Grant Permissions on Objects

1. Review the supplied security requirements in the scenario for this lab.
2. Determine the permissions that should be assigned at the object level.
3. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
4. On the **File** menu, point to **New**, and then click **Query with Current Connection**.
5. In the new query window, type the following code to grant permissions for the e-commerce application to read data from the **Products.vProductCatalog** view and insert rows into the **Sales.SalesOrderHeader** and **Sales.SalesOrderDetail** tables:

```
USE InternetSales;
GO
GRANT SELECT ON Products.vProductCatalog TO WebApplicationSvc;
GRANT INSERT ON Sales.SalesOrderHeader TO WebApplicationSvc;
GRANT INSERT ON Sales.SalesOrderDetail TO WebApplicationSvc;
GO
```

6. Below the code that you have just entered, type the following code to grant permissions for all sales employees and managers to read data from the **Customer** table:

```
GRANT SELECT ON Customers.Customer TO Database_Managers;
GRANT SELECT ON Customers.Customer TO InternetSales_Managers;
GRANT SELECT ON Customers.Customer TO InternetSales_Users;
GO
```

7. On the toolbar, click **Execute**.
8. Minimize SQL Server Management Studio and open a command prompt.
9. At the command prompt, type the following command to open the sqlcmd utility as **adventureworks\anthonyfrizzell**, who is a member of the **IT_Support** group, and then press Enter:

```
runas /user:adventureworks\anthonyfrizzell /noprofile sqlcmd
```

10. At the command prompt, when you are prompted for a password, type **Pa$$w0rd**, and press Enter.

11. In the SQLCMD window, at the command prompt, type the following command to verify your identity, and then press Enter:

```
SELECT suser_name();
GO
```

12. In the SQLCMD window, at the command prompt, type the following commands to verify that Anthony can access the **Customer** table through his membership of the **IT_Support** global group, and hence the **Database Managers** local group and SQL Server login, and then press Enter:

```
USE InternetSales;
GO
SELECT TOP 5 FirstName, LastName FROM Customers.Customer;
GO
```

▶ Task 3: Deny Permissions on Objects

1. You realize that the **Database_Managers** do not need to access the customer information, so decide to deny them access.
2. In SQL Server Management Studio, in the query window, below the existing code, type the following code to deny the **Database_Managers** user SELECT permissions on the **Customer** table:

```
DENY SELECT ON Customers.Customer TO Database_Managers;
GO
```

3. Select the code that you have just typed, and then click **Execute**.
4. In the SQLCMD window, at the command prompt, type the following command to verify that Anthony is now denied access to the **Customer** table, and then press Enter:

```
SELECT TOP 5 FirstName, LastName FROM Customers.Customer;
GO
```

▶ Task 4: Revoke Permissions on Objects

1. You realize that, although the **Database_Managers** users do not need to access the customer information, Anthony is a member of another group and therefore does need access to the table. You decide to revoke the deny permission that you have implemented, leaving Anthony to inherit permissions from his other group membership.
2. In SQL Server Management Studio, in the query window, below the existing code, type the following code to deny the **Database_Managers** user SELECT permissions on the **Customer** table:

```
REVOKE SELECT ON Customers.Customer TO Database_Managers;
GO
```

3. Select the code that you have just typed, and then click **Execute**.
4. In the SQLCMD window, at the command prompt, type the following command to verify that Anthony can access the **Customer** table through his membership of the **Sales_Managers** global group, and hence the **InternetSales_Managers** local group and SQL Server login, and then press Enter:

```
SELECT TOP 5 FirstName, LastName FROM Customers.Customer;
GO
```

5. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.

6. In SQL Server Management Studio, on the **File** menu, click **Close**.
7. In the **Microsoft SQL Server Management Studio** dialog box, click **No**.
8. Leave SQL Server Management Studio open for the next exercise

Results: After completing this exercise, you will have assigned the required object-level permissions.

Exercise 2: Granting EXECUTE Permissions on Code

▶ Task 1: Grant EXECUTE Permission

1. Review the supplied security requirements in the scenario for this lab.
2. Determine the permissions that should be assigned on code.
3. On the **File** menu, point to **New**, and then click **Query with Current Connection**.
4. In the new query window, type the following code to grant permission for the sales managers to run the **ChangeProductPrice** stored procedure, and then click **Execute**:

```
USE InternetSales;
GO
GRANT EXECUTE ON Products.ChangeProductPrice TO InternetSales_Managers;
GO
```

▶ Task 2: Test the Permission

1. In the command prompt window, at the command prompt, type the following command to open the sqlcmd utility as **adventureworks\deannaball**, who is a member of the **IT_Support** group, and then press Enter:

```
runas /user:adventureworks\deannaball /noprofile sqlcmd
```

2. At the command prompt, when you are prompted for a password, type **Pa$$w0rd**, and then press Enter.
3. In the SQLCMD window, at the command prompt, type the following commands to verify that Deanna can run the stored procedure, and then press Enter:

```
USE InternetSales;
GO
EXECUTE Products.ChangeProductPrice 1, 2;
GO
```

4. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.
5. In SQL Server Management Studio, in the query window, below the existing code, type the following code to check that the stored procedure updated the price:

```
SELECT ListPrice from Products.Product WHERE ProductID<10;
```

6. Select the code that you have just typed, and then click **Execute**.
7. On the **File** menu, click **Close**.
8. In the **Microsoft SQL Server Management Studio** dialog box, click **No**.

9. Leave SQL Server Management Studio open for the next exercise.

Results: After completing this exercise, you will have assigned the required EXECUTE permissions on stored procedures.

Exercise 3: Granting Permissions at the Schema Level

▶ Task 1: Grant Permission on a Schema

1. Review the supplied security requirements in the scenario for this lab.
2. Determine the permissions that should be assigned at the schema level.
3. On the **File** menu, point to **New**, and then click **Query with Current Connection**.
4. In the new query window, type the following code to grant permission for the sales managers to insert and update data in the **Sales** schema, and for the sales employees and managers to read data in the **Sales** schema:

```
USE InternetSales;
GO
GRANT INSERT, UPDATE ON SCHEMA::Sales TO InternetSales_Managers;
GRANT SELECT ON Schema::Sales TO InternetSales_Managers;
GRANT SELECT ON Schema::Sales TO InternetSales_Users;
GO
```

5. On the toolbar, click **Execute**.

▶ Task 2: Test the Permission

1. In the command prompt window, at the command prompt, type the following command to open the sqlcmd utility as **adventureworks\anthonyfrizzell**, who is a member of the **Sales_Managers** group, and then press Enter:

```
runas /user:adventureworks\anthonyfrizzell /noprofile sqlcmd
```

2. At the command prompt, when you are prompted for a password, type **Pa$$w0rd**, and then press Enter.
3. In the SQLCMD window, at the command prompt, type the following commands to verify that Anthony can access and update sales data, and then press Enter:

```
USE InternetSales;
GO
SELECT TOP 5 SalesOrderID, CustomerID FROM Sales.SalesOrderHeader;
GO
UPDATE Sales.SalesOrderHeader SET CustomerID=28389 WHERE SalesOrderID=43697;
GO
SELECT TOP 5 SalesOrderID, CustomerID FROM Sales.SalesOrderHeader;
GO
```

4. In the SQLCMD window, at the command prompt, type **exit**, and then press Enter.
5. In the command prompt window, at the command prompt, type **exit**.
6. In SQL Server Management Studio, on the **File** menu, click **Close**.
7. In the **Microsoft SQL Server Management Studio** dialog box, click **No**.

8. On the **File** menu, click **Exit**.
9. In the **Microsoft SQL Server Management Studio** dialog box, click **No**.

Results: After completing this exercise, you will have assigned the required schema-level permissions.

Module 4: Protecting Data with Encryption and Auditing

Lab: Using Auditing and Encryption

Exercise 1: Working with SQL Server Audit

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **MT17B-WS2016-NAT**, **20764C-MIA-DC**, and **20764C-MIA-SQL** virtual machines are running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab04\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.

▶ Task 2: Create a Server Audit

1. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
2. In Object Explorer, expand the **Security** node, right-click the **Audits** node, and then click **New Audit**.
3. In the **Create Audit** dialog box, in the **Audit name** box, type **activity_audit**.
4. In the **File path** box, type **D:\Labfiles\Lab04\Starter\Audit**, and then click **OK**.
5. In Object Explorer, expand the **Audits** node, right-click the **activity_audit** node, and then click **Enable Audit**.
6. In the **Enable Audit** dialog box, click **Close**.

▶ Task 3: Create a Server Audit Specification

1. In Object Explorer, right-click the **Server Audit Specifications** node, and then click **New Server Audit Specification**.
2. In the **Create Server Audit Specification** dialog box, in the **Name** box, type **audit_logins**.
3. In the **Audit** box, type **activity_audit**.
4. In the **Actions** box, in the **Audit Action Type** list, select the **SUCCESSFUL_LOGIN_GROUP** value, and then click **OK**.
5. In Object Explorer, expand the **Server Audit Specifications** node, right-click the **audit_logins** node, and then click **Enable Server Audit Specification**.
6. In the **Enable Server Audit Specification** dialog box, click **Close**.
7. In Object Explorer, collapse the **Security** node.

▶ Task 4: Create a Database Audit Specification

1. In Object Explorer, expand the **Databases** node, expand the **salesapp1** node, and then expand the **Security** node.
2. Right-click the **Database Audit Specifications** node, and click **New Database Audit Specification**.
3. In the **Create Database Audit Specification** dialog box, in the **Name** box, type **employees_change_audit**, and in the **Audit** box, type **activity_audit**.
4. In the **Actions** box, in the **Audit Action Type** list, click the **INSERT** value.

5. In the **Object Class** list, on the first row, click **OBJECT**.
6. In the **Object Name** column, in the first row, click the ellipsis (**...**).
7. In the **Select Objects** dialog box, in the **Enter the object names to select (examples)** box, type **HR.Employees**, and then click **OK**.
8. In the **Principal Name** column, in the first row, click the ellipsis (**...**).
9. In the **Select Objects** dialog box, in the **Enter the object names to select (examples)** box, type **public**, and then click **OK**.
10. On the second row, in the **Audit Action Type** list, click the **UPDATE** value.
11. In the **Object Class** list on the second row, click **OBJECT**.
12. In the **Object Name** column, in the second row, click the ellipsis (**...**).
13. In the **Select Objects** window, in the **Enter the object names to select (examples)** box, type **HR.Employees**, and then click **OK**.
14. In the **Principal Name** column, in the second row, click the ellipsis (**...**).
15. In the **Select Objects** window, in the **Enter the object names to select (examples)** box, type **public**, and then click **OK**.
16. In the **Create Database Audit Specification** dialog box, click **OK**.
17. In Object Explorer, expand the **Database Audit Specifications** node, right-click the **employees_change_audit** node, and then click **Enable Database Audit Specification**.
18. In the **Enable Database Audit Specification** dialog box, click **Close**.

▶ Task 5: Generate Audited Activity

1. In SQL Server Management Studio, on the **File** menu, point to **Open**, and click **Project/Solution**.
2. In the **Open Project** window, browse to **D:\Labfiles\Lab04\Starter\Project**, and double-click **Project.ssmssln**.
3. In Solution Explorer, expand **Queries**, and then double-click the **Lab Exercise 01 - audit activity.sql** query.
4. Highlight the code under the heading **Task 1**, and click **Execute**.

▶ Task 6: Review Audit Data

1. Under the heading **Task 2**, type:

```
SELECT *
FROM sys.fn_get_audit_file(' D:\Labfiles\Lab04\Starter\Audit\*',
default,default)
WHERE session_id = @@SPID;
```

2. Highlight the code you have typed and click **Execute**.

▶ Task 7: Disable the Audit

1. In Object Explorer, under **MIA-SQL**, expand the **Security** node, expand the **Audits** node, right-click **activity_audit**, and then click **Disable Audit**.
2. In the **Disable Audit** dialog box, click **Close**.

Results: After completing this exercise, you will be able to:

Create a server audit.

Create a server audit specification.

Create a database audit specification.

Retrieve audit data.

Exercise 2: Encrypt a Column with Always Encrypted

▶ Task 1: Encrypt a Column

1. In SQL Server Management Studio, under **MIA-SQL**, under **Databases**, under **salesapp1**, expand the **Tables** node, expand the **Sales.Customers** node, and then expand the **Columns** node.
2. Right-click **phone (nvarchar(24), not null)**, and click **Encrypt Column**.
3. In the **Always Encrypted** dialog box, on the **Introduction** page, click **Next**.
4. On the **Column Selection** page, under **Sales.Customers**, select the **phone** row.
5. Change the value of the **Encryption Type** box to **Randomized**, and then click **Next**.
6. On the **Master Key Configuration** page, click **Next**.
7. On the **Run Settings** page, click **Next**.
8. On the **Summary** page, click **Finish**.
9. When the encryption process has finished, click **Close**.
10. In Object Explorer, under **salesapp1,** under the**Tables** node, right-click **Sales.Customers**, and then click **Select top 1000 rows**. Notice that the values in the **phone** column are encrypted.

▶ Task 2: View Always Encrypted Data from an Application

1. In File Explorer, navigate to **D:\Labfiles\Lab04\Starter**, right-click **query_encrypted_columns.ps1**, and then click **Run with PowerShell**.
2. Review the output of the script. The script demonstrates how a change in the connection string to enable the **Column Encryption Setting** property allows it to decrypt the Always Encrypted column. This is possible because the script has access to the column master key in the local Windows key store.
3. When you have finished reviewing the results, press Enter to close the PowerShell window.

Results: After completing this exercise, you will be able to:

Implement Always Encrypted.

Exercise 3: Encrypt a Database Using TDE

▶ Task 1: Create a Service Master Key

1. In SQL Server Management Studio, in Solution Explorer, double-click the **Lab Exercise 03 - TDE.sql** query.
2. Highlight the code under the heading for **Task 1**, and click **Execute**.

▶ Task 2: Back Up the Service Master Key

1. Edit the query under the heading for **Task 2** so that it reads:

```
BACKUP MASTER KEY TO FILE = 'D:\Labfiles\Lab04\Starter\Audit\smk.bak'
    ENCRYPTION BY PASSWORD = 'iD2Z1i85saFyAiK7auzn$';
```

2. Highlight the query you have amended and click **Execute**.

▶ Task 3: Create and Back Up a Server Certificate

1. Highlight the code under the heading for **Task 3**, and click **Execute**.
2. Edit the query under the heading for **Task 4** so that it reads:

```
BACKUP CERTIFICATE TDE_cert
TO FILE = 'D:\Labfiles\Lab04\Starter\Audit\TDE_cert.bak'
WITH PRIVATE KEY
(
    FILE = 'D:\Labfiles\Lab04\Starter\Audit\TDE_cert_pk.bak',
    ENCRYPTION BY PASSWORD = '*R8vkULA5aKhp3ekGg1o3'
);
GO
```

3. Highlight the query you have amended and click **Execute**.

▶ Task 4: Create a Database Encryption Key and Encrypt the salesapp1 Database

1. Edit the query under the heading for **Task 5** so that it reads:

```
USE salesapp1;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDE_cert;
GO
```

2. Highlight the query you have amended and click **Execute**.
3. Highlight the code under the heading for **Task 6**, and click **Execute**.
4. Highlight the code under the heading for **Task 7**, click **Execute**, and then review the results.

▶ Task 5: Move the salesapp1 Database

1. In Object Explorer, under the **Databases** node, right-click **salesapp1**, point to **Tasks**, and then click **Detach**.
2. In the **Detach Database** dialog box, select the **Drop Connections** check box, and then click **OK**.
3. In Object Explorer, in the **Connect** list, click **Database Engine**.
4. Connect to **MIA-SQL\SQL2** using Windows authentication.
5. In Object Explorer, under the **MIA-SQL\SQL2** instance, right-click **Databases** and click **Attach**.
6. In the **Attach Databases** dialog box, click **Add**.

7. In the **Attach Databases** dialog box, navigate to the **D:\Labfiles\Lab04\Starter\Setupfiles** folder, click the **salesapp1.mdf** file, and then click **OK**.
8. In the **Microsoft SQL Server Management Studio** dialog box, notice that an error message is displayed because the certificate with which the database encryption key is protected does not exist on the **MIA-SQL\SQL2** instance. Because of this, the data file cannot be attached, click **OK**.
9. In the **Attach Databases** dialog box, click **Cancel**.
10. In Solution Explorer, double-click the **Lab Exercise 03 - move DB.sql** query.
11. On the **Query** menu, point to **Connection**, and then click **Change Connection**.
12. In the **Connect to Database Engine** dialog box, connect to the **MIA-SQL\SQL2** database engine using Windows authentication.
13. Highlight the code under the heading for **Task 10**, and click **Execute**. This creates a server master key on **MIA-SQL\SQL2**.
14. Highlight the code under the heading for **Task 11**, and click **Execute**. This creates a certificate in the **master** database on **MIA-SQL\SQL2** from the backup files you created previously.
15. In Object Explorer, under the **MIA-SQL\SQL2** instance, right-click **Databases** and click **Attach**.
16. In the **Attach Databases** dialog box, click **Add**.
17. In the **Attach Databases** dialog box, navigate to the **D:\Labfiles\Lab04\Starter\Setupfiles** folder, click the **salesapp1.mdf** file, and then click **OK**.
18. In the **Attach Databases** dialog box, click **OK**.
19. Highlight the code under the heading for **Task 12**, and click **Execute**. This queries the **Sales.Customers** table in the salesapp1 database.
20. Review the query results then close SQL Server Management Studio without saving any files.

Results: After completing this exercise, you will be able to:

Encrypt a database using TDE.

Move an encrypted database to another SQL Server instance.

Module 5: Recovery Models and Backup Strategies

Lab: Understanding SQL Server Recovery Models

Exercise 1: Plan a Backup Strategy

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab05\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes** to run the command file, and wait for the script to finish.

▶ Task 2: Review the Business Requirements

- Review the supplied business requirements in the supporting documentation for this exercise.

▶ Task 3: Determine an Appropriate Backup Strategy

- The options provided below are an example strategy that would meet the requirements. Review these suggestions.

For the MarketDev database:

Full Recovery Model

Notes:

- Full database backup should complete in approximately 3.4 hours (20 GB/100 MB per minute). This means that you cannot employ a full database backup strategy only as it would not meet the RPO. The full database backup should complete by 22:24, which is within the available time window for backups.
- Each log backup should complete in approximately 3.4 minutes (1 GB per hour/3 log backups per hour/100 MB per minute). This fits within the 20-minute interval and meets the RPO.
- Full database restore should complete in approximately 4.3 hours (20 GB/80 MB per minute). Log file restore should complete in approximately 2.1 hours (10 hours * 1 GB per hour/80 MB per minute) and meet the RTO.

For the Research Database:

Simple Recovery Model

Type of Backup	Schedule
Full	18:30 daily

Notes:

- Recovery to last full daily database backup complies with the RPO.
- Daily backup should complete in approximately 2 minutes (200 MB/100 MB per minute).

- Full restore should complete in approximately 2.5 minutes and complies with RTO (200 MB/80 MB per minute).

Results: At the end of this exercise, you will have created a plan to back up two databases.

Exercise 2: Configure Database Recovery Models

▶ Task 1: Review and Adjust Database Recovery Models

1. On the taskbar, click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** box, type **MIA-SQL**, and then click **Connect**.
3. In Object Explorer, expand **Databases**, right-click **MarketDev**, and then click **Properties**.
4. On the **Options** page, in the **Recovery model** list, click **Full**, and then click **OK**.
5. In Object Explorer, under **Databases**, right-click **Research**, and then click **Properties**.
6. On the **Options** page, verify that the **Recovery model** is set to **Simple**, and then click **Cancel**.
7. Close SSMS without saving any changes.

Results: At the end of this exercise, you will have modified the database recovery models where required.

Exercise 3: Challenge Exercise: Review Recovery Models and Strategy (if time permits)

▶ Task 1: Review the RPO and RTO Requirements for the Databases

- The supporting documentation includes details of the business continuity requirements for the databases. Review this documentation.

▶ Task 2: Review the Existing Recovery Models and Backup Strategies

- The supporting documentation also includes details of the backup strategy for the databases. Review this documentation.

▶ Task 3: Indicate Whether or Not the Strategy Would Be Successful

1. For the **CreditControl** database, a full backup would take approximately 3.4 hours (20 GB/100 MB per minute). This would not satisfy the requirements for the time window, because the Wednesday 06:00 full backup would not complete before office hours start.
2. For the **PotentialIssue** database, the 15-minute log backups would meet the RPO. A full restore should take approximately 24 minutes ((200 MB + (7 days * 24 hours) * 10 MB per hour)/80 MB per minute) which meets the RTO. The full database backup would complete in approximately two minutes (200 MB/100 MB per minute) which means that the full database would complete in the available time window. The backup strategy for the **PotentialIssue** database meets the business requirements.

Results: At the end of this exercise, you will have assessed the backup strategy.

Module 6: Backing Up SQL Server Databases

Lab: Backing Up Databases

Exercise 1: Backing Up Databases

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab06\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for the script to finish.
4. In File Explorer, navigate to **D:**.
5. On the **Home** ribbon, click **New folder**.
6. Type **Backups**, and press Enter.

▶ Task 2: Set the Recovery Model

1. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
2. In Object Explorer, expand **Databases**, right-click **AdventureWorks**, and then click **Properties**.
3. In the **Database Properties - AdventureWorks** dialog box, on the **Options** page, in the **Recovery model** drop-down list, click **Simple**, and then click **OK**.

▶ Task 3: Perform a Full Database Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, ensure that **Backup type** is set to **Full**.
3. In the **Destination** section, click the existing file path, click **Remove**, and then click **Add**.
4. In the **Select Backup Destination** dialog box, in the **File name** box, type **D:\Backups\AdventureWorks.bak**, and then click **OK**.
5. In the **Back Up Database - AdventureWorks** dialog box, on the **Media Options** page, click **Back up to a new media set, and erase all existing backup sets**.
6. In the **New media set name** box, type **AdventureWorks Backup**.
7. On the **Backup Options** page, note the default backup name.
8. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
9. When the backup has completed successfully, click **OK**.
10. In File Explorer, in the **D:\Backups** folder, verify that the backup file **AdventureWorks.bak** has been created, and note its size.

▶ Task 4: Modify Data in the Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**.

```
UPDATE HumanResources.Employee
SET VacationHours = VacationHours + 10 WHERE SickLeaveHours < 30;
```

3. Note the number of rows affected, and then close the query pane without saving the file.

▶ Task 5: Perform Another Full Database Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, ensure that **Backup type** is set to **Full**.
3. In the **Destination** section, verify that **D:\Backups\AdventureWorks.bak** is listed.
4. On the **Media Options** page, ensure that **Back up to the existing media set** and **Append to the existing backup set** are selected.
5. On the **Backup Options** page, in the **Name** box, type **AdventureWorks-Full Database Backup 2**.
6. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
7. When the backup has completed successfully, click **OK**.
8. In File Explorer, in the **D:\Backups** folder, verify the **AdventureWorks.bak** backup file has increased in size.

▶ Task 6: View the Backup and Restore Events Report

1. In SQL Server Management Studio, in Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Reports**, point to **Standard Reports**, and then click **Backup and Restore Events**.
2. In the **Backup and Restore Events [AdventureWorks]** report, expand **Successful Backup Operations** and view the backup operations that have been performed for this database.
3. In the **Device Type** column, expand each of the **Disk (temporary)** entries to view details of the backup media set file.
4. Close the report pane.

Results: At the end of this exercise, you will have backed up the **AdventureWorks** database to D:\Backups\AdventureWorks.bak using the simple recovery model.

Exercise 2: Performing Database, Differential and Transaction Log Backups

▶ Task 1: Set the Recovery Model

1. In SQL Server Management Studio, in Object Explorer, under **Databases**, right-click **AdventureWorks**, and then click **Properties**.
2. In the **Database Properties - AdventureWorks** dialog box, on the **Options** page, in the **Recovery model** drop-down list, ensure that **Full** is selected, and then click **OK**.

▶ Task 2: Perform a Full Database Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks,** point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, ensure that **Backup type** is set to **Full**.
3. In the **Destination** section, click the existing file path, click **Remove**, and then click **Add**.
4. In the **Select Backup Destination** dialog box, in the **File name** box, type **D:\Backups\AWNational.bak**, and then click **OK**.
5. In the **Back Up Database - AdventureWorks** dialog box, on the **Media Options** page, click **Back up to a new media set, and erase all existing backup sets**.
6. In the **New media set name** box, type **AdventureWorks Backup**.
7. On the **Backup Options** page, note the default backup name.
8. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
9. When the backup has completed successfully, click **OK**.
10. In File Explorer, in the **D:\Backups** folder, verify that the backup file **AWNational.bak** has been created, and note its size.

▶ Task 3: Modify Data in the Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**.

```
UPDATE HumanResources.Employee
SET VacationHours = VacationHours + 10 WHERE SickLeaveHours < 30;
```

3. Note the number of rows affected, and then close the query pane without saving the file.

▶ Task 4: Perform a Transaction Log Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, in the **Backup type** list, click **Transaction Log**.
3. In the **Destination** section, verify that **D:\Backups\AWNational.bak** is listed.
4. On the **Media Options** page, ensure that **Back up to the existing media set** and **Append to the existing backup set** are selected.
5. On the **Backup Options** page, in the **Name** box, type **AdventureWorks-Transaction Log Backup**.
6. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
7. When the backup has completed successfully, click **OK**.
8. In File Explorer, in the **D:\Backups** folder, verify the **AWNational.bak** backup file has increased in size.

▶ Task 5: Modify Data in the Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**.

```
UPDATE HumanResources.Employee
SET VacationHours = VacationHours + 10 WHERE SickLeaveHours < 30;
```

3. Note the number of rows affected, and then close the query pane without saving the file.

▶ Task 6: Perform a Differential Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, in the **Backup type** list, click **Differential**.
3. In the **Destination** section, verify that **D:\Backups\AWNational.bak** is listed.
4. On the **Media Options** page, ensure that **Back up to the existing media set** and **Append to the existing backup set** are selected.
5. On the **Backup Options** page, in the **Name** box, type **AdventureWorks-Differential Backup**.
6. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
7. When the backup has completed successfully, click **OK**.
8. In File Explorer, in the **D:\Backups** folder, verify the **AWNational.bak** backup file has increased in size.

▶ Task 7: Modify Data in the Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**.

```
UPDATE HumanResources.Employee
SET VacationHours = VacationHours + 10 WHERE SickLeaveHours < 30;
```

3. Note the number of rows affected, and then close the query pane without saving the file.

▶ Task 8: Perform Another Transaction Log Backup

1. In Object Explorer, under **Databases**, right-click **AdventureWorks**, point to **Tasks**, and then click **Back Up**.
2. In the **Back Up Database - AdventureWorks** dialog box, in the **Backup type** list, click **Transaction Log**.
3. In the **Destination** section, verify that **D:\Backups\AWNational.bak** is listed.
4. On the **Media Options** page, ensure that **Back up to the existing media set** and **Append to the existing backup set** are selected.
5. On the **Backup Options** page, in the **Name** box, type **AdventureWorks-Transaction Log Backup 2**.
6. In the **Set backup compression** list, click **Compress backup**, and then click **OK**.
7. When the backup has completed successfully, click **OK**.
8. In File Explorer, in the **D:\Backups** folder, verify the **AWNational.bak** backup file has increased in size.

▶ Task 9: Verify Backup Media

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**:

```
RESTORE HEADERONLY
FROM DISK = 'D:\Backups\AWNational.bak';
GO
```

3. Verify that the backups you performed in this exercise are all listed.
4. Modify the Transact-SQL code as follows, and then click **Execute**:

```
RESTORE FILELISTONLY
FROM DISK = 'D:\Backups\AWNational.bak';
GO
```

5. Note the database files that are included in the backups.
6. Modify the Transact-SQL code as follows, and then click **Execute**:

```
RESTORE VERIFYONLY
FROM DISK = 'D:\Backups\AWNational.bak';
GO
```

7. Verify that the backup is valid then close the query pane without saving the file.

Results: At the end of this exercise, you will have backed up the national database to D:\Backups\AWNational.bak.

Exercise 3: Performing a Partial Backup

▶ Task 1: Create Read-Only Filegroup

1. In SQL Server Management Studio, open the **CreateReadOnly.sql** script file from the **D:\Labfiles\Lab06\Starter\Setupfiles** folder, and then click **Execute**.
2. This script creates the required read-only components you need for this Lab, and then close the query pane without saving the file.

▶ Task 2: Back Up the Read-Only Filegroup

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**:

```
BACKUP DATABASE AdventureWorks FILEGROUP = 'ReadOnlyFG' TO DISK =
'D:\Labfiles\Lab06\Starter\AWReadOnly.bak' WITH FORMAT, INIT, NAME = 'AdventureWorks
Read Only FG Backup', COMPRESSION;
```

3. In File Explorer, in the **D:\Labfiles\Lab06\Starter** folder, verify that the backup file **AWReadOnly.bak** has been created.

▶ Task 3: Perform a Partial Backup

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**:

```
BACKUP DATABASE AdventureWorks READ_WRITE_FILEGROUPS TO DISK =
'D:\Labfiles\Lab06\Starter\AWPartial.bak' WITH FORMAT, INIT, NAME = 'AdventureWorks
Partial Backup', COMPRESSION;
```

3. In File Explorer, in the **D:\Labfiles\Lab06\Starter** folder, verify that the backup file **AWPartial.bak** has been created.

▶ Task 4: Modify Data in the Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**.

```
UPDATE HumanResources.Employee
SET VacationHours = VacationHours + 10 WHERE SickLeaveHours < 30;
```

3. Note the number of rows affected, and then close the query pane without saving the file.

▶ Task 5: Perform a Differential Partial Backup

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code to perform a differential partial backup, and then click **Execute**:

```
BACKUP DATABASE AdventureWorks
READ_WRITE_FILEGROUPS
TO DISK = 'D:\Labfiles\Lab06\Starter\AWPartialDifferential.bak'
WITH DIFFERENTIAL, NOFORMAT, NOINIT, NAME = 'AdventureWorks-Active Data Diff',
COMPRESSION;
```

3. In File Explorer, in the **D:\Labfiles\Lab06\Starter** folder, verify that the backup file **AWPartialDifferential.bak** has been created.

▶ Task 6: Verify Backup Media

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code, and then click **Execute**:

```
RESTORE HEADERONLY FROM DISK = 'D:\Labfiles\Lab06\Starter\AWReadOnly.bak';
GO
```

3. View the backups on this backup media, and scroll to the right to view the **BackupTypeDescription** column.
4. Modify the Transact-SQL code as follows, and then click **Execute**:

```
RESTORE HEADERONLY FROM DISK = 'D:\Labfiles\Lab06\Starter\AWPartial.bak';
GO
```

5. View the backups on this backup media, and scroll to the right to view the **BackupTypeDescription** column.
6. Close SQL Server Management Studio without saving any script files.

Results: At the end of this exercise, you will have backed up the read-only filegroup in the **AdventureWorks** database to **D:\Backups\AWReadOnly.bak**; and you will have backed up the writable filegroups in the **AdventureWorks** database to **D:\Backups\AWReadWrite.bak**.

Module 7: Restoring SQL Server Databases

Lab: Restoring SQL Server Databases

Exercise 1: Restoring a Database Backup

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab07\Starter** folder, right-click **Setup.cmd** file, and click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. When you are prompted to continue this operation, type **Y**, press Enter, wait for the script to finish, and then press any key to close the command prompt.

▶ Task 2: Determine the Cause of the Failure

1. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
2. In Object Explorer, expand **Databases**, and note that the **HumanResources** database is in a **Recovery Pending** state.
3. In SQL Server Management Studio, click **New Query**.
4. In the query pane, type the following Transact-SQL code to attempt to bring the database online:

```
ALTER DATABASE HumanResources SET ONLINE;
```

5. Click **Execute**, and note the error message that is displayed. The database cannot be brought online because the primary data file is lost.
6. View the contents of the **D:\Labfiles\Lab07\Starter\Setupfiles** folder to verify that the **HumanResources.mdf** file is not present.

▶ Task 3: Restore the HumanResources Database

1. In SQL Server Management Studio, click **New Query**.
2. In the query pane, type the following Transact-SQL code to attempt to restore the database, and then click **Execute**:

```
RESTORE DATABASE HumanResources FROM DISK =
'D:\Labfiles\Lab07\Backups\HumanResources.bak';
```

3. In Object Explorer, verify that the **HumanResources** database is now recovered and ready to use. You may need to click the **Refresh** button.

Results: After this exercise, you should have restored the **HumanResources** database.

Exercise 2: Restoring Database, Differential, and Transaction Log Backups

▶ Task 1: Determine the Cause of the Failure

1. In SQL Server Management Studio, in Object Explorer, under **Databases**, note that the **InternetSales** database is in a **Recovery Pending** state.
2. Click **New Query**.
3. In the query pane, type the following Transact-SQL code to attempt to bring the database online:

```
ALTER DATABASE InternetSales SET ONLINE;
```

4. Click **Execute**, and note the error message that is displayed. There is a problem with the primary data file.
5. View the contents of the **D:\Labfiles\Lab07\Starter\Setupfiles** folder to verify that the **InternetSales.mdf** file is present. This file has become corrupt, and has rendered the database unusable.

▶ Task 2: Perform a Tail-log Backup

1. View the contents of the **D:\Labfiles\Lab07\Starter\Setupfiles** folder and verify that the **InternetSales.ldf** file is present.
2. In SQL Server Management Studio, click **New Query**.
3. In the query pane, type the following Transact-SQL code to back up the tail of the transaction log:

```
USE master;
BACKUP LOG InternetSales TO DISK = 'D:\Labfiles\Lab07\Backups\InternetSales.bak'
WITH NO_TRUNCATE;
```

4. Click **Execute**, and view the resulting message to verify that the backup is successful.

▶ Task 3: Restore the InternetSales Database to Another Instance

1. In Object Explorer, click **Connect**, and then click **Database Engine**.
2. In **Server name**, type **MIA-SQL\SQL2** and click **Connect**.
3. In Object Explorer, under **MIA-SQL\SQL2**, right-click **Databases**, and then click **Restore Database**.
4. In the **Restore Database** dialog box, in the **Source** section, click **Device**, and then click the ellipsis (**...**) button.
5. In the **Select backup devices** dialog box, click **Add**.
6. In the **Locate Backup File - MIA-SQL\SQL2** dialog box, select **D:\Labfiles\Lab07\Backups\InternetSales.bak**, and then click **OK**.
7. In the **Select backup devices** dialog box, click **OK**.
8. Note that the backup media contains a full backup, and a transaction log backup (these are the planned backups in **InternetSales.bak**), in addition to a copy-only transaction log backup (which is the tail-log backup). All of these are automatically selected in the **Restore** column.
9. On the **Files** page, select the **Relocate all files to folder** check box.
10. In the **Data file folder**, delete the existing text, and type **D:\Labfiles\Lab07\Backups**.
11. In the **Log file folder**, delete the existing text, and type **D:\Labfiles\Lab07\Backups**.
12. On the **Options** page, ensure that the **Recovery state** is set to **RESTORE WITH RECOVERY**.

13. In the **Script** drop-down list, click **New Query Editor Window**, and then click **OK**.
14. When the database has been restored successfully, click **OK**.
15. View the Transact-SQL code that was used to restore the database, noting that the full backup, the differential backup, and the first transaction log backup, were restored using the NORECOVERY option. The restore operation for the tail-log backup used the default RECOVERY option to recover the database.
16. In Object Explorer, under **MIA-SQL\SQL2**, verify that the **InternetSales** database is now recovered and ready to use.

Results: After this exercise, you should have restored the **InternetSales** database.

Exercise 3: Performing a Piecemeal Restore

▶ Task 1: Begin a Piecemeal Restore

1. In SQL Server Management Studio, in Object Explorer, under the **MIA-SQL\SQL2** instance, under **Databases**, verify that the **AWDataWarehouse** database is not listed.
2. Click **New Query**.
3. In the query pane, type the following Transact-SQL code to start a partial restore of the database from the full backup set in position 1, in the **AWDataWarehouse.bak** media set:

```
USE master;
RESTORE DATABASE AWDataWarehouse FILEGROUP='Current'
FROM DISK = 'D:\Labfiles\Lab07\Backups\AWDataWarehouse.bak'
WITH REPLACE, PARTIAL, FILE = 1, NORECOVERY;
```

4. Click **Execute**, and view the resulting message to verify that the restore is successful.
5. In Object Explorer, under the **MIA-SQL2** instance, right-click the **Databases** folder, and then click **Refresh**; verify that **AWDataWarehouse** is listed with a "Restoring" status.

▶ Task 2: Restore Read/Write Filegroups and Bring the Database Online

1. In SQL Server Management Studio, under the existing code in the query pane, type the following Transact-SQL code to restore the differential backup set in position 2, in the **AWDataWarehouse.bak** media set:

```
RESTORE DATABASE AWDataWarehouse
FROM DISK = 'D:\Labfiles\Lab07\Backups\AWDataWarehouse.bak'
WITH FILE = 2, RECOVERY;
```

2. Select the code you just entered, and click **Execute**, and view the resulting message to verify that the restore is successful.
3. In Object Explorer, under **MIA-SQL2** instance, right-click the **Databases** folder, click **Refresh**, and then verify that **AWDataWarehouse** is now shown as online.
4. Expand the **AWDataWarehouse** database, expand **Tables**, right-click **dbo.FactInternetSales**, and then click **Select Top 1000 Rows**. Note that you can retrieve data from this table, which is stored in the read/write **Current** filegroup.

5. In Object Explorer, right-click **dbo.FactInternetSalesArchive**, and then click **Select Top 1000 Rows**. Note that you cannot retrieve data from this table, which is stored in the read-only **Archive** filegroup.

Results: After this exercise, you will have restored the **AWDataWarehouse** database.

Module 8: Automating SQL Server Management

Lab: Automating SQL Server Management

Exercise 1: Create a SQL Server Agent Job

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab08\Starter** folder, right-click the **Setup.cmd** file and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for the script to finish.

▶ Task 2: Create a Backup Job

1. Start SQL Server Management Studio, and connect to the **MIA-SQL** database engine using Windows authentication.
2. In Object Explorer, expand **SQL Server Agent**, and then expand **Jobs** to view any existing jobs.
3. Right-click **Jobs**, and then click **New Job**.
4. In the **New Job** dialog box, on the **General** page, in the **Name** box, type **Backup HumanResources**.
5. On the **Steps** page, click **New**.
6. In the **New Job Step** dialog box, on the **General** page, in the **Step name** box, type **Back Up Database**.
7. In the **Database** list, click **HumanResources**, and, in the **Command** box, type the following command:

```
BACKUP DATABASE HumanResources TO DISK = 'D:\HumanResources.bak';
```

8. In the **New Job Step** dialog box, on the **Advanced** page, in the **Output file** box, type **D:\Labfiles\Lab08\Starter\BackupLog.txt**, and then click **OK**.
9. In the **New Job** dialog box, on the **Steps** page, click **New**.
10. In the **New Job Step** dialog box, on the **General** page, in the **Step name** box, type **Move Backup File**.
11. In the **Type** list, click **Operating system (CmdExec)**, and in the **Command** box, type the following command, which moves the backup file to the **D:\Labfiles\Lab08\Starter** folder:

```
Move /Y D:\HumanResources.bak D:\Labfiles\Lab08\Starter\HumanResources.bak /Y
```

12. In the **New Job Step** dialog box, click **OK**.
13. In the **New Job** dialog box, on the **Steps** page, verify that the **Start step** is set to **1:Back Up Database**, and in the **Job step list** table, note the **On Success** and **On Failure** actions for the steps in the job.
14. In the **New Job** dialog box, click **OK**.
15. In Object Explorer, in the Jobs folder, verify that the job appears.
16. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have created a job named **Backup HumanResources**.

Exercise 2: Test a Job

▶ Task 1: Test the Backup Job

1. In Object Explorer, under **Jobs**, right-click **Backup HumanResources**, and then click **Start Job at Step**.
2. In the **Start Job on 'MIA-SQL'** dialog box, click **Start**.
3. Note that the **Start Jobs - MIA-SQL** dialog shows an error, and then click **Close**.

▶ Task 2: Investigate the Error

1. In Object Explorer, right-click **Backup HumanResources**, and then click **View History**.
2. In the **Log File Viewer - MIA-SQL** dialog box, expand the first row by clicking the date cell.
3. Note that **Step ID 1** has succeeded.
4. Click the **Step ID 2** cell, resize the lower panel that contains the text **Selected row details:** and scroll to the bottom.
5. The error message text should read:

```
Executed as user: ADVENTUREWORKS\ServiceAcct. The syntax of the command is incorrect.
Process Exit Code 1.  The step failed.
```

6. The move command in Step 2 of the job has a syntax error.
7. In the **Log File Viewer - MIA-SQL** dialog box, click **Close**.

▶ Task 3: Resolve the Error and Retest

1. In Object Explorer, right-click **Backup HumanResources**, and then click **Properties**.
2. In the **Job Properties - Backup HumanResources** dialog box, on the **Steps** page, click **Move Backup File**, and then click **Edit**.
3. In the **Job Step Properties - Move Backup File** dialog box, in the **Command** box, delete the existing text, replace it with the following command, and then click **OK**:

```
Move /Y D:\HumanResources.bak D:\Labfiles\Lab08\Starter\HumanResources.bak
```

4. In the **Job Properties - Backup HumanResources** dialog box, click **OK**.
5. In Object Explorer, right-click **Backup HumanResources**, and then click **Start Job at Step**.
6. In the **Start Job on 'MIA-SQL'** dialog box, click **Start**.
7. Note that in the **Start Jobs - MIA-SQL** dialog box the job finished with success, and then click **Close**.
8. In File Explorer, navigate to **D:\Labfiles\Lab08\Starter** and note that the **HumanResources.bak** file is present.
9. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have tested the SQL Server Agent job and confirmed that it executes successfully.

Exercise 3: Schedule a Job

▶ Task 1: Add a Schedule to the Job

1. Right-click the taskbar and click **Settings**.

   Scroll down to the Notification area section, and then click **Turn system icons on or off**.

2. On the Turn system icons on or off page, on the **Clock** row, make sure it is switched **On**.
3. Close the settings window.
4. Note that the clock now appears in the taskbar.
5. In Microsoft SQL Server Management Studio, in Object Explorer, double-click **Backup HumanResources**.
6. In the **Job properties - Backup HumanResources** dialog box, on the **Schedules** page, click **New**.
7. In the **New Job Schedule** dialog box, in the **Name** box, type **Daily Backup**, and then in the **Frequency** area, in the **Occurs** list, click **Daily**.
8. In the **Daily frequency** section, in the **Occurs once at** section, set the time to two minutes from the current system time as shown in the clock in the notification area, and then click **OK**.
9. In the **Job properties - Backup HumanResources** dialog box, click **OK**, and wait until the system clock shows the scheduled time.

▶ Task 2: Verify Scheduled Job Execution

1. In Object Explorer, double-click **Job Activity Monitor**, and note the **Status** of the **Backup HumanResources** job.
2. If the job is still running, click **Refresh** until the **Status** changes to **Idle**.
3. Verify that the **Last Run Outcome** for the job is **Succeeded**, that the **Last Run** time is the time that you scheduled previously, and then click **Close**.
4. Leave SQL Server Management Studio open for the next exercise.

Results: At the end of this exercise, you will have created a schedule for the **Backup HumanResources** job.

Exercise 4: Configure Master and Target Servers

▶ Task 1: Configure the Master and Target Servers

1. In Object Explorer, right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Make this a Master**.
2. In the **Master Server Wizard - MIA-SQL** dialog box, click **Next**.
3. On the **Master Server Operator** page, click **Next**.

4. On the **Target Servers** page, in the **Registered servers** section, expand **Database Engine**, and expand **Local Server Groups**.
5. Click **mia-sql\sql2**, and click **>**, click **mia-sql\sql3**, and click **>**, and then click **Next**.
6. In the **Checking Server Compatibility** dialog box, click **Close**.
7. On the **Master Server Login Credentials** page, click **Next**.
8. On the **Complete the Wizard** page, click **Finish**.
9. On the **Enlist TSX Progress** page, click **Close**.

▶ Task 2: Create a Maintenance Plan

1. In Object Explorer, expand **Management**, right-click **Maintenance Plans**, and then click **Maintenance Plan Wizard**.
2. In the **Maintenance Plan Wizard** dialog box, click **Next**.
3. On the **Select Plan Properties** page, in the **Name** box, type **System Database Backups**.
4. In the **Schedule** section, click **Change**.
5. In the **New Job Schedule** dialog box, in the **Frequency** section, in the **Occurs** list, click **Daily**.
6. In the **Daily frequency** section, in the **Occurs once at** box, set the time to five minutes from the current system time as shown in the clock in the notification area, and then click **OK**.
7. On the **Select Plan Properties** page, click **Next**.
8. On the **Select Target Servers** page, select the boxes next to **MIA-SQL\SQL2** and **MIA-SQL\SQL3**, and then click **Next**.
9. On the **Select Maintenance Tasks** page, check **Back Up Database (Full)**, and then click **Next**.
10. On the **Select Maintenance Task Order** page, click **Next**.
11. On the **Define Backup Database (Full) Task** page, on the **General** tab, in the **Database(s)** list, click **System databases**, and then click **OK**.
12. On the **Destination** tab, select the **Create a sub-directory for each database** check box, in the **Folder** box, type **D:**, and then click **Next**.
13. On the **Select Report Options** page, in the **Folder location** box, type **D:**, and then click **Next**.
14. On the **Complete the Wizard** page, click **Finish**.
15. On the **Maintenance Plan Wizard Progress** page, wait for the operation to complete, and then click **Close**.
16. In Object Explorer, right-click **SQL Server Agent**, point to **Multi Server Administration**, and then click **Manage Target Servers**.
17. In the **Target Server** list, click **MIA-SQL\SQL2**, and then click **Force Poll**.
18. In the **Force Target Server to Poll MSX Immediately** dialog box, click **OK**.
19. In the **Target Server** list, click **MIA-SQL\SQL3**, and then click **Force Poll**.
20. In the **Force Target Server to Poll MSX Immediately** dialog box, click **OK**.

 **Note:** By forcing the target servers to poll the master server, they will get a copy of the System Database Backup jobs.

21. In the **Target Server Status - MIA-SQL** dialog box, click **Close**.
22. In File Explorer, view the **D:** folder. Log files should have been created from each server and for each system database; a folder should have been created, holding backup files from each SQL Server instance.

▶ Task 3: Monitor the Job History

1. In SQL Server Management Studio, in Object Explorer, under **SQL Server Agent**, double-click **Job Activity Monitor**.
2. In the **Job Activity Monitor - MIA-SQL** window, note the rows in the **Agent Job Activity** table for:
   - **System Database Backups.Subplan_1**
   - **System Database Backups.Subplan_1 (Multi-Server)**
3. Right-click **System Database Backups.Subplan_1**, and then click **View history**.
4. In the **Log File Viewer - MIA-SQL** dialog box, review the history information, and then click **Close**.
5. Right-click **System Database Backups.Subplan_1 (Multi-Server)**, and then click **View history**.
6. Note that there is less detail in the Log File Viewer for the multiserver jobs. For more detailed information, use the Job Activity Monitor on the target server.
7. In the **Log File Viewer - MIA-SQL** dialog box, click **Close**.
8. In the **Job Activity Monitor - MIA-SQL** dialog box, click **Close**.
9. In Object Explorer, click **Connect**, and then click **Database Engine**.
10. In the **Connect to Server** dialog, in the **Server name** box, type **MIA-SQL\SQL2**, and then click **Connect**.
11. In Object Explorer, under **MIA-SQL\SQL2**, expand **SQL Server Agent**, and then double-click **Job Activity Monitor**.
12. In the **Job Activity Monitor - MIA-SQL\SQL2** dialog box, right-click **System Database Backups.Subplan_1 (Multi-Server)**, and then click **View history**.
13. Note that there is more information shown on the target server.
14. In the **Log File Viewer - MIA-SQL\SQL2** dialog box, click **Close**.
15. In the **Job Activity Monitor - MIA-SQL\SQL2** dialog box, click **Close**.
16. Close SQL Server Management Studio.

Results: At the end of this exercise, you will have created and executed a maintenance plan across multiple servers.

Module 9: Configuring Security for SQL Server Agent

Lab: Configuring Security for SQL Server Agent

Exercise 1: Analyzing Security Problems in SQL Server Agent

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab09\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.
4. Press any key to close the window.

▶ Task 2: Examine the Job History for the Failing Job

1. Start Microsoft SQL Server Management Studio, and then connect to the **MIA-SQL** Database Engine instance by using Windows authentication.
2. In Object Explorer, expand **SQL Server Agent**, expand **Jobs**, right-click **Generate Sales Log**, and then click **View History**.
3. In the **Log File Viewer - MIA-SQL** window, expand the first job execution by clicking the plus sign on a row in the right pane, and then scroll the window to the right so that the **Message** column is visible. (The job is started on a schedule, so one or more rows of job history might be visible.)
4. Notice that the failure message for the job step reads as follows:

```
Non-SysAdmins have been denied permission to run DTS Execution job steps without a
proxy account.  The step failed.
```

5. Click **Close**.
6. In Object Explorer, right-click **Generate Sales Log**, and then click **Properties**.
7. In the **Job Properties - Generate Sales Log** window, notice that the owner of the job is the **PromoteApp09** login, and then click **Cancel**.
8. Leave SQL Server Management Studio open for the next exercise. The job step is failing because the job is owned by a login who is not a member of the **sysadmin** role.

Results: After completing this exercise, you should have identified the cause of the job failure.

Exercise 2: Configuring a Credential

▶ Task 1: Create a Credential

1. In Object Explorer, under **MIA-SQL**, expand **Security**, right-click **Credentials**, and then click **New Credential**.
2. In the **New Credential** dialog box, in the **Credential name** box, type **ExtractUser**.
3. In the **Identity** box, click the ellipsis (**...**) button.
4. In the **Select User or Group** dialog box, click **Locations**.
5. In the **Locations** dialog box, click **Entire Directory**, and then click **OK**.
6. In the **Select User, Service Account, or Group** dialog box, in the **Enter the object name to select** box, type **Student**, click **Check Names**, and then click **OK**.
7. In the **New Credential** window, in the **Password** and **Confirm password** boxes, type **Pa55w.rd**, and then click **OK**.
8. In Object Explorer, expand **Credentials** to verify that **ExtractUser** appears.
9. Leave SQL Server Management Studio open for the next exercise.

Results: After completing this exercise, you should have created a credential that references the **ADVENTUREWORKS\Student** Windows account.

Exercise 3: Configuring a Proxy Account

▶ Task 1: Create a Proxy Account

1. In Object Explorer, under **SQL Server Agent**, right-click **Proxies**, and then click **New Proxy**.
2. In the **New Proxy Account** window, on the **General** page, in the **Proxy name** box, type **ExtractProxy**.
3. In the **Credential name** box, click the ellipsis (**...**) button.
4. In the **Select Credential** dialog box, click **Browse**.
5. In the **Browse for Objects** dialog box, select **ExtractUser**, and then click **OK**.
6. In the **Select Credential** dialog box, click **OK**.
7. In the **New Proxy Account** window, in the **Active to the following subsystems** box, select **SQL Server Integration Services Package**.
8. On the **Principals** page, click **Add**.
9. In the **Add Principal** dialog box, verify that **Principal type** has the value **SQL Login**, select **PromoteApp09**, and then click **OK**.
10. In the **New Proxy Account** window, click **OK**.
11. In Object Explorer, expand **Proxies**, and then expand **SSIS Package Execution** to verify that **ExtractProxy** appears.
12. Leave SQL Server Management Studio open for the next exercise.

Results: After completing this exercise, you should have created a proxy account that is suitable for correcting the problem with the SQL Server Agent job called **Generate Sales Log**.

Exercise 4: Configuring and Testing the Security Context of the Job

▶ Task 1: Configure the Job to Use a Proxy Account

1. In Object Explorer, under **SQL Server Agent**, under **Jobs**, right-click **Generate Sales Log**, and then click **Properties**.
2. In the **Job Properties - Generate Sales Log** window, on the **Steps** page, click **Edit**.
3. On the **Job Step Properties - Execute Package** page, in the **Run as** box, click **ExtractProxy**, and then click **OK**.
4. In the **Job Properties - Generate Sales Log** window, click **OK**.

▶ Task 2: Test the Configuration

1. Right-click **Generate Sales Log**, and then click **Start Job at Step**.
2. In the **Start Jobs - MIA-SQL** dialog box, make sure that the job ran successfully, and then click **Close**.
3. In File Explorer, browse to **D:\Labfiles\Lab09\Starter\SalesLog**, and then verify that **sales_log.csv** has been generated.
4. Close File Explorer, and then close SQL Server Management Studio without saving any changes.

Results: After completing this exercise, the **Generate Sales Log** SQL Server Agent job should be working correctly, and the **sales_log.csv** file should be generated to **D:\Labfiles\Lab09\Starter\SalesLog** each time the job runs.

Module 10: Monitoring SQL Server with Alerts and Notifications

Lab: Monitoring SQL Server with Alerts and Notifications

Exercise 1: Configuring Database Mail

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab10\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes** and wait for the script to finish.

▶ Task 2: Configure Database Mail

1. Start SQL Server Management Studio and connect to the **MIA-SQL** Database Engine instance using Windows authentication.
2. In Object Explorer, expand **Management**, right-click **Database Mail**, and then click **Configure Database Mail**.
3. On the **Welcome to Database Mail Configuration Wizard** page, click **Next**.
4. On the **Select Configuration Task** page, click **Set up Database Mail by performing the following tasks:**, and then click **Next**.
5. On the **New Profile** page, in the **Profile name** box, type **SQL Server Agent Profile**, and then click **Add**.
6. In the **Add Account to profile 'SQL Server Agent Profile'** dialog box, click **New Account**.
7. In the **New Database Mail Account** dialog box, enter the following details and click **OK**:
   - **Account name**: AdventureWorks Administrator
   - **Email address**: administrator@adventureworks.msft
   - **Display name**: Administrator (AdventureWorks)
   - **Reply email**: administrator@adventureworks.msft
   - **Server name**: mia-sql.adventureworks.msft
8. On the **New Profile** page, click **Next**.
9. On the **Manage Profile Security** page, select **Public** for the **SQL Server Agent Profile** profile, set its **Default Profile** setting to **Yes**, and then click **Next**.
10. On the **Configure System Parameters** page, click **Next**.
11. On the **Complete the Wizard** page, click **Finish**, and when configuration is complete, click **Close**.

▶ Task 3: Test Database Mail

1. In Object Explorer, right-click **Database Mail**, and then click **Sent Test E-Mail**.
2. In the **Send Test E-Mail from MIA-SQL** dialog box, ensure that the **SQL Server Agent Profile** database mail profile is selected.
3. In the **To** box, type **student@adventureworks.msft**, and then click **Send Test Email**.
4. In File Explorer, navigate to the **C:\inetpub\mailroot\Drop** folder, and verify that an email message has been created.
5. Double-click the message to view it in Outlook. When you have read the message, close it and minimize the Drop folder window.
6. In the **Database Mail Test E-Mail** dialog box (which might be behind SQL Server Management Studio), click **OK**.
7. In SQL Server Management Studio, on the toolbar, click **New Query**.
8. In the query pane, type the following Transact-SQL code, and then click **Execute**:

```
SELECT * FROM msdb.dbo.sysmail_event_log;
SELECT * FROM msdb.dbo.sysmail_mailitems;
```

9. Review the results. The first result shows system events for Database Mail, and the second shows records of email messages that have been sent.
10. Close the query pane without saving changes.
11. Leave SSMS open for the next exercise.

Results: After this exercise, you should have configured Database Mail with a new profile named **SQL Server Agent Profile**.

Exercise 2: Configuring Operators

▶ Task 1: Create the Student Operator

1. In Object Explorer, expand **SQL Server Agent**, right-click **Operators**, and then click **New Operator**.
2. In the **New Operator** dialog box, in the **Name** box, type **Student**, in the **E-mail name** box, type **student@adventureworks.msft**, and then click **OK**.

▶ Task 2: Create the DBA Team Operator

1. In Object Explorer, under **SQL Server Agent**, right-click **Operators** and click **New Operator**.
2. In the **New Operator** dialog box, in the **Name** box, type **DBA Team**, in the **E-mail name** box, type **dba@adventureworks.msft**, and then click **OK**.

▶ Task 3: Configure the SQL Server Agent Mail Profile

1. In Object Explorer, right-click **SQL Server Agent**, and then click **Properties**.
2. In the **SQL Server Agent Properties** dialog box, on the **Alert System** page, select **Enable mail profile**.
3. In the **Mail profile** drop-down list, click **SQL Server Agent Profile**.

4. Select **Enable fail-safe operator**, in the **Operator** drop-down list, click **DBA Team**.
5. In the **Notify using** section, select **E-mail**, and then click **OK**.
6. In Object Explorer, right-click **SQL Server Agent**, and then click **Restart**.
7. In the **User Account Control** dialog box, click **Yes**.
8. In the **Microsoft SQL Server Management Studio** dialog box, click **Yes**.
9. Leave SSMS open for the next exercise.

Results: After this exercise, you should have created operators named **Student** and **DBA Team**, and configured the SQL Server Agent service to use the **SQL Server Agent Profile** Database Mail profile.

Exercise 3: Configuring Alerts and Notifications

▶ Task 1: Create an Alert

1. In Object Explorer, expand **SQL Server Agent**, right-click **Alerts**, and then click **New Alert**.
2. In the **New Alert** dialog box, on the **General** page, in the **Name** box, type **InternetSales Log Full Alert**.
3. In the **Database name** list, click **InternetSales**.
4. Click **Error number**, and type the error number **9002**.
5. On the **Response** page, select **Execute job**, and in the drop-down list, click **Back Up Log - InternetSales ([Uncategorized (Local)])**.
6. Select **Notify operators**, and then select **E-mail** for the **Student** operator.
7. On the **Options** page, under **Include alert error text in**, select **E-mail**, and then click **OK**.

▶ Task 2: Configure Job Notifications

1. In Object Explorer, under **SQL Server Agent**, expand **Jobs** and view the existing jobs.
2. Right-click **Back Up Database - AWDataWarehouse**, and then click **Properties**.
3. In the **Job Properties - Back Up Database - AWDataWarehouse** dialog box, on the **Notifications** page, select **E-mail**.
4. In the first drop-down list, click **Student**.
5. In the second drop-down list, click **When the job fails**, and then click **OK**.
6. Right-click **Back Up Database - HumanResources**, and then click **Properties**.
7. In the **Job Properties - Back Up Database - HumanResources** dialog box, on the **Notifications** page, select **E-mail**.
8. In the first drop-down list, click **Student**.
9. In the second drop-down list, click **When the job fails**, and then click **OK**.
10. Right-click **Back Up Database - InternetSales**, and then click **Properties**.
11. In the **Job Properties - Back Up Database - InternetSales** dialog box, on the **Notifications** page, select **E-mail**.

12. In the first drop-down list, click **Student**.
13. In the second drop-down list, click **When the job completes**, and then click **OK**.
14. Right-click the **Back Up Log - InternetSales** job, and then click **Properties**.
15. In the **Job Properties - Back Up Log - InternetSales** dialog box, on the **Notifications** page, select **E-mail**.
16. In the first drop-down list, click **Student**.
17. In the second drop-down list, click **When the job completes**, and then click **OK**.
18. Expand the **Operators** folder, right-click **Student**, and then click **Properties**.
19. In the **Student Properties** dialog box, on the **Notifications** page, click **Jobs**, verify the job notifications that have been defined for this operator, and then click **Cancel**.
20. Leave SSMS open for the next exercise.

Results: After this exercise, you should have created an alert named **InternetSales Log Full Alert**.

Also, you should have configured the **Back Up Database - AWDataWarehouse**, **Back Up Database - HumanResources**, **Back Up Database - InternetSales**, and **Back Up Log - InternetSales** jobs to send notifications.

Exercise 4: Testing Alerts and Notifications

▶ Task 1: Test the Alert

1. On the **File** menu, point to **Open**, and then click **File**.
2. In the **Open File** dialog box, navigate to the **D:\Labfiles\Lab10\Starter** folder, click **TestAlert.sql**, and then click **Open**.
3. On the toolbar, click **Execute** and wait for the script to finish. When the log file for the **InternetSales** database is full, error 9002 occurs.
4. In Object Explorer, expand **Alerts**, right-click **InternetSales Log Full Alert**, and then click **Properties**.
5. In the **'InternetSales Log Full Alert' alert properties** dialog box, on the **History** page, note the **Date of last alert** and **Date of last response** values, and then click **Cancel**.
6. In File Explorer, in the **C:\inetpub\mailroot\Drop** folder, verify that four new email messages have been created.
7. Double-click the new email messages to view them in Outlook. They should include a notification that the transaction log was filled, and a notification that the **Back Up Log - InternetSales** job completed. When you have finished checking them, close all email messages and minimize the **Drop** window.

▶ Task 2: Test Job Notifications

1. In Object Explorer, under **Jobs**, right-click **Back Up Database - AWDataWarehouse**, and click **Start Job at Step**.
2. When the job has completed, note that it failed and click **Close**.
3. In Object Explorer, under **Jobs**, right-click **Back Up Database - HumanResources**, and click **Start Job at Step**.

4. When the job has completed, note that it succeeded and click **Close**.
5. In Object Explorer, under **Jobs**, right-click **Back Up Database - InternetSales**, and click **Start Job at Step**.
6. When the job has completed, note that it succeeded and click **Close**.
7. Under the **Operators** folder, right-click **Student**, and then click **Properties**.
8. In the **Student Properties** dialog box, on the **History** page, note the date and time of the most recent notification by email, and then click **Cancel**.
9. In File Explorer, in the **C:\inetpub\mailroot\Drop** folder, verify that new email messages have been created.
10. Open each of the messages and verify that they include a failure notification for the **Back Up Database - AWDataWarehouse** job and a completion notification for the **Back Up Database - InternetSales** job, but no notification regarding the **Back Up Database - HumanResources job**.
11. When you have read the messages, close them and close the **Drop** window.
12. Close SQL Server Management Studio without saving any files.

Results: After this exercise, you will have verified that the notifications you configured for backups of the **AWDataWarehouse**, **HumanResources**, and **InternetSales** databases work as expected.

You will also verify that an alert is triggered when the transaction log of the **Internet Sales** database is full.

Module 11: Introduction to Managing SQL Server Using PowerShell

Lab: Using PowerShell to Manage SQL Server

Exercise 1: Getting Started with PowerShell

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab11\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for the script to finish.

▶ Task 2: Using PowerShell Help

1. On the Start menu, type, **Windows PowerShell**, right click, **Windows PowerShell**, then click Run as administrator.
2. In the **User Account Control** dialog box, click **Yes**.
3. In the console, type **Update-Help**, and then press Enter. Wait for the help files to update.
4. At the command prompt, type **Get-Help**, and then press Enter to display the summary help page.
5. At the command prompt, type **Get-Help Get-ChildItem**, and then press Enter to display help for the cmdlet.
6. At the command prompt, type **Get-Command Remove-***, and then press Enter to get a list of all the cmdlets that start with the verb **Remove-**.
7. At the command prompt, type **Get-Command *sql***, and then press Enter to get a list of all the cmdlets that include "sql".
8. At the command prompt, type **Get-Item**, and then press TAB to view tab-completion of the cmdlet.
9. Press ESC to clear the command.

▶ Task 3: Using PowerShell Providers

1. At the command prompt, type **Get-PsProvider**, and then press Enter to list the PowerShell providers and valid locations.
2. At the command prompt, type **Set-Location Env:**, and then press Enter to go to the Environment location.
3. At the command prompt, type **Get-ChildItem**, and then press Enter to list the contents of the location.
4. At the command prompt, type **Get-PSDrive**, and then press Enter to list the available drives.
5. Compare the PSDrives with the PSProviders.
6. At the command prompt, type **Import-Module SQLPS**, and then press Enter to import the SQL PowerShell module. You can ignore any warnings that appear.

7. At the command prompt, type **Get-PsProvider**, and then press Enter to list the PowerShell providers. Note that SqlServer now appears in the list.
8. At the command prompt, type **Get-PSDrive**, and then press Enter to list the available drives. Note that SqlServer now appears in the list.
9. At the command prompt, type **Get-Help Get-PSProvider**, press Enter, and then read the returned information to learn more about the PSProvider cmdlet.
10. At the command prompt, type **Get-Help Get-PSDrive**, press Enter, and then read the returned information to learn more about the PSDrive cmdlet.
11. At the command prompt, type **Exit** to close the console.

Results: After completing this exercise, you will have investigated PowerShell help and the SQL PowerShell provider.

Exercise 2: Using PowerShell to Change SQL Server Settings

▶ Task 1: View Database Settings

1. On the Start menu, type Windows PowerShell, right-click **Windows PowerShell**, and then click **Run ISE as Administrator**.
2. In the **User Account Control** dialog box, click **Yes**.
3. On the **File** menu, click **Open**.
4. In the **Open** dialog box, navigate to **D:\Labfiles\Lab11\Starter**, click **DisplayProperties.ps1**, and then click **Open**.
5. Select the code under the **#1#** comment, and then on the toolbar, click **Run Selection** to display "import the SQL module". Ignore any warnings that may appear.
6. Select the code under the **#2#** comment, and then on the toolbar, click **Run Selection** to change the location.
7. Select the code under the **#3#** comment, and then on the toolbar, click **Run Selection** to get the database object.
8. Select the code under the **#4#** comment, and then on the toolbar, click **Run Selection** to display the database properties.
9. Select the code under the **#5#** comment, and then on the toolbar, click **Run Selection** to display the database option.

▶ Task 2: Amend Database Settings

1. On the **File** menu, click **Open**.
2. In the **Open** dialog box, navigate to **D:\Labfiles\Lab11\Starter**, click **ChangeProperties.ps1**, and then click **Open**.
3. Select the code under the **#1#** comment, and then on the toolbar, click **Run Selection** to prepare the script.
4. Select the code under the **#2#** comment, and then on the toolbar, click **Run Selection** to change the location.

5. Select the code under the **#3#** comment, and then on the toolbar, click **Run Selection** to get the database object.
6. Select the code under the **#4#** comment, and then on the toolbar, click **Run Selection** to change the compatibility level of the AdventureWorks2016 database.
7. Select the code under the **#5#** comment, and then on the toolbar, click **Run Selection** to change the ANSI nulls, autoshrink, read only, and recovery model options.

▶ Task 3: View and Amend Server Settings

1. On the **File** menu, click **Open**.
2. In the **Open** dialog box, navigate to **D:\Labfiles\Lab11\Starter**, click **ChangeServerSettings.ps1**, and then click **Open**.
3. Select the code under the **#1#** comment, and then on the toolbar, click **Run Selection** to display "import the SQL module". Ignore any warnings that may appear.
4. Select the code under the **#2#** comment, and then on the toolbar, click **Run Selection** to change the location.
5. Select the code under the **#3#** comment, and then on the toolbar, click **Run Selection** to get the server object.
6. Select the code under the **#4#** comment, and then on the toolbar, click **Run Selection** to display the **Settings** object.
7. Select the code under the **#5#** comment, and then on the toolbar, click **Run Selection** to change the login mode to mixed authentication.
8. Select the code under the **#6#** comment, and then on the toolbar, click **Run Selection** to change the login mode back to integrated authentication.
9. Select the code under the **#7#** comment, and then on the toolbar, click **Run Selection** to examine the **UserOptions** object.
10. Select the code under the **#8#** comment, and then on the toolbar, click **Run Selection** to change some server settings.
11. Select the code under the **#9#** comment, and then on the toolbar, click **Run Selection** to reset the server settings.
12. On the **File** menu, click **Exit**.

Results: After completing this lab exercise, you will have PowerShell scripts to show the IT Director.

Module 12: Tracing Access to SQL Server with Extended Events

Lab: Extended Events

Exercise 1: Using the system_health Extended Events Session

▶ Task 1: Prepare the Lab Environment

1. Ensure the **MT17B-WS2016-NAT**, **20764C-MIA-DC**, and **20764C-MIA-SQL** virtual machines are running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab12\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.

▶ Task 2: Run a Workload

1. In File Explorer, navigate to the **D:\Labfiles\Lab12\Starter** folder.
2. Right-click **start_load_1.ps1**, and then click **Run with PowerShell**.
3. If a message is displayed asking you to confirm a change in execution policy, type **Y**, and then press Enter.
4. Wait for the workload to complete, which should take about a minute, and then press Enter to close the Windows PowerShell window.

▶ Task 3: Query the system_health Extended Events Session

1. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine using Windows authentication.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, navigate to the **D:\Labfiles\Lab12\Starter\Project** folder, click **Project.ssmssln**, and then click **Open**.
4. In Solution Explorer, expand **Queries**, and then double-click **Exercise 01 - system_health.sql**.
5. Edit the code under the comment that begins **Task 2** so that it reads as follows:

```
SELECT CAST(event_data AS xml) AS xe_data
FROM sys.fn_xe_file_target_read_file('system_health*.xel', NULL, NULL, NULL);
```

6. Select the query that you edited in the previous step, and then click **Execute**.

▶ Task 4: Extract Deadlock Data

1. Edit the query under the comment that begins **Task 3** so that it reads as follows:

```
SELECT xe_event.c.value('@timestamp', 'datetime2(3)') AS event_time,
xe_event.c.query('/event/data/value/deadlock') AS deadlock_data
FROM
(
   SELECT CAST(event_data AS xml) AS xe_data
   FROM sys.fn_xe_file_target_read_file('system_health*.xel', NULL, NULL, NULL)
) AS xe_data
CROSS APPLY xe_data.nodes('/event') AS xe_event(c)
WHERE xe_event.c.value('@name', 'varchar(100)') = 'xml_deadlock_report'
ORDER BY event_time;
```

2. Select the query that you edited in the previous step, and then click **Execute**.
3. In the **Results** pane, click any of the row values in the **deadlock_data** column to view the deadlock XML in detail.
4. On the **File** menu, click **Close**.
5. Leave SSMS open for the next exercise.

Results: After completing this exercise, you will have extracted deadlock data from the SQL Server.

Exercise 2: Tracking Page Splits Using Extended Events

▶ Task 1: Create an Extended Events Session to Track Page Splits

1. In Solution Explorer, double-click **Exercise 02 - page splits.sql**.
2. In Object Explorer, under **MIA-SQL**, expand **Management**, expand **Extended Events**, right-click **Sessions**, and then click **New Session Wizard**.
3. In the **New Session Wizard**, on the **Introduction** page, click **Next**.
4. On the **Set Session Properties** page, in the **Session name** box, type **track page splits**, and then click **Next**.
5. On the **Choose Template** page, click **Next**.
6. On the **Select Events To Capture** page, in the **Event library** section, in the **Channel** column header, click the drop-down button, and then select the **Debug** check box.
7. In the **Search Events** box, type **transaction_log**, double-click the **transaction_log** row in the **Event Library** list, which will add it to the **Selected events** list, and then click **Next**.
8. On the **Capture Global Fields** page, click **Next**.
9. On the **Set Session Event Filters** page, click **Click here to add a clause**.
10. In the **Field** drop-down list, click **sqlserver.database_name**.
11. In the **Value** box, type **AdventureWorks**, and then click **Finish**.
12. On the **Create Event Session** page, click **Close**.
13. In Object Explorer, expand **Sessions**, right-click **track page splits**, and then click **Properties**.
14. In the **Session Properties** dialog box, on the **Events** page, click **Configure**.

15. In the **Selected events** list, click **transaction_log**.
16. On the **Filter (Predicate)** tab, click **Click here to add a clause**.
17. In the **And/Or** box, ensure that the value is **And**.
18. In the **Field** list, click **operation**.
19. In the **Operator** list, click **=**.
20. In the **Value** list, click **LOP_DELETE_SPLIT**.
21. On the **Data Storage** page, click **Click here to add a target**.
22. In the **Type** list, click **histogram**.
23. In the **Event to filter on** list, click **transaction_log**.
24. In the **Base buckets on** section, click **Field**, in the **Field** list, click **alloc_unit_id**, and then click **OK**.
25. In Object Explorer, right-click **track page splits**, and then click **Start Session**.

▶ Task 2: Run a Workload

1. In File Explorer, navigate to the **D:\Labfiles\Lab12\Starter** folder, right-click **start_load_2.ps1**, and then click **Run with PowerShell**.
2. If a message is displayed asking you to confirm a change in execution policy, type **Y**, and then press Enter.
3. Wait for the workload to complete. This should take about 60 seconds.

▶ Task 3: Query the Session

1. In SQL Server Management Studio, in the query window for **Exercise 02 - page splits.sql**, edit the code under the comment that begins **Task 3** so that it reads as follows:

```
USE AdventureWorks;
SELECT CAST(target_data AS XML) AS target_data
FROM sys.dm_xe_sessions AS xs
JOIN sys.dm_xe_session_targets xt
ON xs.address = xt.event_session_address
WHERE xs.name = 'track page splits'
AND xt.target_name = 'histogram';
```

2. Select the query that you edited in the previous step, and then click **Execute**.
3. In the results pane, click the returned XML to review the data.
4. On the **File** menu, click **Close**.

▶ Task 4: Extract alloc_unit_id and Count Values

1. In the **Exercise 02 - page splits.sql** pane, edit the code under the comment that begins **Task 4** so that it reads as follows:

```
SELECT xe_node.value('(value)[1]', 'bigint') AS alloc_unit_id,
xe_node.value('(@count)[1]', 'bigint') AS split_count
FROM (     SELECT CAST(target_data AS XML) AS target_data
FROM sys.dm_xe_sessions AS xs
JOIN sys.dm_xe_session_targets xt
ON xs.address = xt.event_session_address
WHERE xs.name = 'track page splits'
AND xt.target_name = 'histogram')
AS xe_data
CROSS APPLY target_data.nodes('HistogramTarget/Slot') AS xe_xml (xe_node);
```

2. Select the query that you edited in the previous step, and then click **Execute**.
3. Review the number of splits in each node.

▶ Task 5: Return Object Names

1. Edit the code under the comment that begins **Task 5** so that it reads as follows:

```
SELECT OBJECT_SCHEMA_NAME(sp.object_id) AS object_schema,
          OBJECT_NAME(sp.object_id) AS object_name,
          si.name AS index_name,
          xe.split_count
FROM (    SELECT xe_node.value('(value)[1]', 'bigint') AS alloc_unit_id,
                        xe_node.value('(@count)[1]', 'bigint') AS split_count
          FROM (  SELECT CAST(target_data AS XML) AS target_data
                        FROM sys.dm_xe_sessions AS xs
                        JOIN sys.dm_xe_session_targets xt
                        ON xs.address = xt.event_session_address
                        WHERE xs.name = 'track page splits'
                        AND xt.target_name = 'histogram') AS xe_data
          CROSS APPLY target_data.nodes('HistogramTarget/Slot') AS xe_xml (xe_node))
AS xe
JOIN sys.allocation_units AS sau
ON sau.allocation_unit_id = xe.alloc_unit_id
JOIN sys.partitions AS sp
ON sp.partition_id = sau.container_id
JOIN sys.indexes AS si
ON si.object_id = sp.object_id
AND si.index_id = sp.index_id;
```

2. Select the query that you edited in the previous step, and then click **Execute**.
3. Review the objects affected by page splits.

▶ Task 6: Delete the Session

1. In Object Explorer, under **Sessions**, right-click **track page splits**, and then click **Delete**.
2. In the **Delete Object** dialog box, click **OK**.
3. Close SSMS without saving changes.

Results: After completing this exercise, you will have extracted page split data from SQL Server.

Module 13: Monitoring SQL Server
Lab: Monitoring SQL Server

Exercise 1: Create and configure a Management Data Warehouse

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **20764C-MIA-DC** and **20764C-MIA-SQL** virtual machines are both running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab13\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, leave the window open as it is creating a load on the database.

▶ Task 2: Create a Management Data Warehouse

1. Start **SQL Server Management Studio**, and then connect to the **MIA-SQL** database engine instance by using Windows authentication.
2. In Object Explorer, right-click on **MIA-SQL**, then click **Activity Monitor**.
3. In the **MIA-SQL – Activity Monitor** tab, click on **Recent Expensive Queries** row to expand it.
4. Note the first row in the list starts with, **SELECT total.name**.
5. In the **Object Explorer** pane, expand the **Management** folder, right-click **Data Collection**, click **Tasks**, then click **Configure Management Data Warehouse**.
6. In the **Configure Management Data Warehouse Wizard** dialog, click **Next**.
7. On the **Configure Management Data Warehouse Storage** page, next to Database name, click **New**.
8. In the **New Database** dialog, in the Database name textbox, type **MDW**.
9. Click **OK**.
10. On the **Configure Management Data Warehouse Storage** page, click **Next**.
11. On the **Map Logins and Users** page, click **Next**.
12. On the **Complete the Wizard** page, click **Finish**.
13. On the **Configure Data Collection Wizard Progress** page, click **Close**.
14. Note in the Object Explorer, a new **MDW** database has been created.
15. Leave SQL Server Management Studio open for the next task.

▶ Task 3: Capture data into the Management Data Warehouse

1. In the **Object Explorer** pane, expand the **Management** folder, right-click **Data Collection**, click **Tasks**, then click **Configure Data Collection**.
2. In the **Configure Data Collection Wizard** dialog box, click **Next**.
3. On the **Setup Data Collection Sets** page, next to the **Server name** textbox, click the **…** button.
4. In the **Connect to Server** dialog, check the Server name is **MIA-SQL**, then click **Connect**.
5. On the **Setup Data Collection Sets** page, in the **Database name** dropdown, select **MDW**.

6. In the **Select data collector sets you want to enable**, check **System Data Collection Sets**, and then click **Next**.
7. On the **Complete the Wizard** page, click **Finish**.
8. On the **Configure Data Collection Wizard Progress** page, click **Close**.
9. In the **Object Explorer** pane, expand **Management**, expand **Data Collection**, expand **System Data Collection Sets**, right-click **Query Statistics**, then click **Properties**.
10. In the **Data Collection Set Properties** dialog, under **Data Collection and upload**, select **Non-cached – Collect and upload data on the same schedule**.
11. Then next to **Schedule**, click **Pick**.
12. In the **Pick Schedule for job** dialog, click the row with **ID 2, Occurs every day every 5 minutes**.
13. Click **OK**.
14. In the **Data Collection Set Properties** dialog, click **OK**.
15. In the **Object Explorer** pane, expand the **Databases** folder, right-click the **MDW** database, click **Reports**, click **Management Data Warehouse**, then click **Management Data Warehouse Overview**.
16. On the **Management Data Warehouse Overview: MDW** report, under the **Query Statistics** column, click the date hyperlink.
17. On the **Query Statics History** report, in the **Query #** table, click the **SELECT total.Name** hyperlink.
18. On the **Query Details** report, scroll to the bottom, in the **Top Query Plans by Average CPU Per Execution** table, in the **Plan #** column, click the **1** hyperlink.
19. On the **Query Plan Details** report, at the bottom of the page, click the **View graphical query execution plan** hyperlink.
20. Note there are no performance recommendations.
21. Close the **ExecutionPlan1.sqlplan** tab.
22. Close the **Query Plan Details** tab.
23. Leave SQL Server Management Studio open for the next exercise.

Results: After completing this exercise, you should have configured a Management Data Warehouse called MDW on the MIA-SQL instance.

Exercise 2: Use the MDW to diagnose performance issues

▶ Task 1: Prepare the lab environment

1. In the **D:\Labfiles\Lab13\Starter** folder, right-click **Execerise2.cmd**, and then click **Run as administrator**.
2. In the **User Account Control** dialog box, click **Yes**, leave the window open as it is creating a load on the database.

▶ Task 2: Investigate execution plans

1. In SQL Server Management Studio, in the **Object Explorer** pane, right-click **MIA-SQL**, then click **Activity Monitor**.
2. Click **Recent Expensive Queries** to show a list of queries.
3. Right-click the top query starting **SELECT total.name**, then click **Show Execution Plan**.
4. Note that SQL Server has identified that there is a missing index that if created could improve performance.
5. Close the **ExecutionPlan** tab.
6. In the **Object Explorer** pane, expand the **Management** folder, right-click **Data Collection**, click **Reports**, click **Management Data Warehouse**, and then click **Query Statistics History**.
7. Note the time at the top of the report.
8. In the **Object Explorer** pane, expand the **Management** folder, expand **Data Collection**, expand the **System Data Collection Sets** folder, right-click on **Query Statistics**, then click **Collect and Upload Now**.
9. In the Collect and upload Data Collection Sets dialog, click **Close**.
10. In the Query Statistics History report, at the top of the report, click the **Refresh**.
11. Note the time at the top has updated.
12. In the table below, click the top hyperlink starting **SELECT total.name**.
13. On the **Query Details** report, scroll to the bottom of the report, in the **Top Query Plans By Average CPU Per Execution** table, in the Plan # column, click the hyperlink **1**.
14. Note that there is a **Missing Indexes** textbox with a suggestion to create an index to improved performance.

Results: At the end of this exercise you will be able to:

Use Activity Monitor to see which running queries are the most expensive

Module 14: Troubleshooting SQL Server

Lab: Troubleshooting Common Issues

Exercise 1: Troubleshoot and Resolve a SQL Login Issue

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **MT17B-WS2016-NAT**, **20764C-MIA-DC**, and **20764C-MIA-SQL** virtual machines are running, and then log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
2. In the **D:\Labfiles\Lab14\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.

▶ Task 2: Review the Exercise Scenario

- Review the exercise scenario; make sure that you understand the problem you are trying to resolve.

▶ Task 3: Troubleshoot and Resolve the Issue

1. On the taskbar, click **Microsoft SQL Server Management Studio**.
2. Connect to the **MIA-SQL** database engine using Windows authentication.
3. In Object Explorer, expand the **MIA-SQL** server, expand **Security**, and then expand **Logins**.
4. Note the **PromoteApp** login has an icon with a red down arrow—the login is disabled.
5. Right-click **PromoteApp**, and click **Properties**.
6. In the **Login Properties - PromoteApp** dialog box, on the **Status** page, click **Enabled**, and then click **OK**.
7. In Object Explorer, right-click **Logins**, and click **Refresh**. Note that the login is now enabled.
8. Leave SQL Server Management Studio open for the next exercise.

Results: After this exercise, you will have investigated and resolved a SQL login issue; the **PromoteApp** login will be functioning properly.

Exercise 2: Troubleshoot and Resolve a Service Issue

▶ Task 1: Review the Exercise Scenario

- Review the exercise scenario; make sure that you understand the problem you are trying to resolve.

▶ Task 2: Troubleshoot and Resolve the Issue

Gather More Information

1. Using Internet Explorer, browse to **http://mia-sql/Reports_SQL2/Pages/Folder.aspx**.
2. Read the error message, and notice that it includes more information than users reported. Particularly, it says:

```
The report server can't connect to its database. Make sure the database is running
and accessible.
```

3. This might indicate a problem with the **MIA-SQL\SQL2** database engine instance.
4. Click **Start**, type **Event Viewer**, and press Enter.
5. In Event Viewer, in the left pane, expand **Windows Logs**, and then click **System**.
6. Click on each of the three most recent messages with a **Level** of **Error** and a **Source** of **Service Control Manager**. Notice that the oldest of the three messages contains the following message:

```
The MSSQL$SQL2 service was unable to log on as ADVENTUREWORKS\ServiceAcct with the
currently configured password due to the following error:
The user name or password is incorrect.
```

7. Close Event Viewer.

Resolve the Issue

1. Click the **Start** button, type **Configuration Manager**, and then click **SQL Server Configuration Manager**.
2. In the **User Account Control** dialog box, click **Yes**.
3. In Sql Server Configuration Manager, in the left pane, click **SQL Server Services**.
4. In the right pane, note that the **SQL Server (SQL2)** service is not running.
5. Right-click **SQL Server (SQL2)**, and then click **Properties**.
6. In the **SQL Server (SQL2) Properties** dialog box, on the **Log On** tab, in the **Password** and **Confirm password** boxes, type **Pa55w.rd**, and then click **OK**.
7. Right-click **SQL Server (SQL2)**, and click **Start**. Notice that the service starts normally.
8. Close SQL Server Configuration Manager.
9. In Internet Explorer, click the **Refresh** button. Notice that Reporting Services is now accessible.
10. Close Internet Explorer.

Results: After this exercise, the Reporting Services instance at http://mia-sql/Reports_SQL2/Pages/Folder.aspx will be functioning properly.

Exercise 3: Troubleshoot and Resolve a Windows Login Issue

▶ Task 1: Simulate the Issue

1. In the **D:\Labfiles\Lab14\Starter** folder, right-click **FrizzellQuery.cmd**, and then click **Open**.
2. When the script completes, press any key to close the command prompt window.

▶ Task 2: Review the Exercise Scenario

- Review the exercise scenario; make sure you understand the problem you are trying to resolve.

▶ Task 3: Troubleshoot and Resolve the Issue

1. In SQL Server Management Studio, in Object Explorer, under **MIA-SQL**, under **Security**, and under **Logins**, right-click **ADVENTUREWORKS\AnthonyFrizzell** and then click **Properties**.
2. On the **General** page, notice that the login is configured for Windows authentication.
3. On the **Status** page, notice that the login has permission to connect to the database engine, and that it is enabled. Click **Cancel**.
4. In Object Explorer, expand **Management**, expand **SQL Server Logs**, right-click the log with the name that begins **Current**, and then click **View SQL Server Log**.
5. In the **Log File Viewer – MIA-SQL** window, in the right pane, locate the first entry with a **Source** of **Logon**. Notice that the error message begins:

   ```
   Login failed for user 'Adventureworks\AnthonyFrizzell'.
   ```

6. Click **Close**.
7. Leave SQL Server Management Studio open for the next exercise.
8. The user's attempt to log in is failing because he is attempting to provide a username and password, when he should be using trusted authentication. He should use **sqlcmd** with the **-E** switch to use trusted authentication, not the **-U** and **-P** switches—which are used to provide a user name and password for SQL Server authentication.

Results: At the end of this exercise, you will be able to explain to the user why he cannot connect to the database.

Exercise 4: Troubleshoot and Resolve a Job Execution Issue

▶ Task 1: Review the Exercise Scenario

- Review the exercise scenario; make sure that you understand the problem you are trying to resolve.

▶ Task 2: Execute the Failing Job

1. In SQL Server Management Studio, in Object Explorer, under **MIA-SQL**, expand **SQL Server Agent**, expand **Jobs**, right-click **Generate File List**, and then click **Start Job at Step**.
2. In the **Start Jobs - MIA-SQL** dialog box, note the failure, and then click **Close**.

▶ Task 3: Troubleshoot and Resolve the Issue

1. In Object Explorer, under **SQL Server Agent**, under **Jobs**, right-click **Generate File List**, and then click **View History**.
2. In the **Log File Viewer - MIA-SQL** window, expand the first entry.
3. Click on the row with a **Step ID** value of **1**, in the bottom pane, scroll down to view the error message. Notice that the job is attempting to call a stored procedure which doesn't exist. Click **Close**.
4. Right-click **Generate File List**, and click **Properties**.
5. In the **Job Properties - Generate File List** window, on the **Steps** page, click **Edit**.
6. In the **Job Step Properties - Execute Procedure** window, in the **Database** and **Command** boxes, observe that the job is attempting to execute a stored procedure in the **InternetSales** database.
7. In Object Explorer, under **MIA-SQL**, expand **Databases**, expand **Internet Sales**, expand **Programmability**, and then expand **Stored Procedures**. Observe that a stored procedure exists in the database with the name **dbo.usp_GenerateFileList**.
8. In the **Job Step Properties - Execute Procedure** window, edit the content of the **Command** box so that it reads the following, and then click **OK**:

```
EXECUTE dbo.usp_GenerateFileList
GO
```

9. In the **Job Properties - Generate File List** window, click **OK**.
10. To test the fix, in Object Explorer, under **SQL Server Agent**, under **Jobs**, right-click **Generate File List**, and then click **Start Job at Step**. Observe that the job completes successfully, and then click **Close**.
11. Leave SQL Server Management Studio open for the next exercise.

Results: After this exercise, you will have investigated and resolved a job execution issue.

Exercise 5: Troubleshoot and Resolve a Performance Issue

▶ Task 1: Simulate the Issue

1. In the **D:\Labfiles\Lab14\Starter** folder, right-click **Performance.cmd**, and then click **Open**.
2. Notice that three command prompt windows are opened by the script; these windows represent users of the **InternetSales** database. Do not interact with or close these windows; two of them will close when the issue is resolved.

▶ Task 2: Review the Exercise Scenario

- Review the exercise scenario; make sure that you understand the problem you are trying to resolve.

▶ Task 3: Troubleshoot and Resolve the Issue

1. In SQL Server Management Studio, in Object Explorer, right-click **MIA-SQL,** and then click **Activity Monitor**.
2. Expand **Processes**, on the **Database Name** column, click the **Filter** icon. The **Filter** icon is a small square to the right of the column name.

3. In the filter list, click **InternetSales**. This filters the list to show only connections to the **InternetSales** database.
4. The filtered list will include three entries where the value of the **Application Name** column is **SQLCMD**. Notice that for one of these rows (the row with the lowest **SID** value) the value of the **Head Blocker** column is **1**, and that the other two rows have values in the **Blocked By** column.
5. Click the row that has a **Head Blocker** value of **1**, right-click the row, and then click **Details**.
6. In the **Session details...** dialog box, the text of the Transact-SQL command being executed by the session is displayed. Also notice that the command includes a BEGIN TRANSACTION statement without a corresponding COMMIT or ROLLBACK; this open transaction is blocking other sessions from accessing the **Sales.Orders** table.
7. To enable other users to continue to work with the application, click **Kill Process** to kill the blocking session.
8. In the **Kill Process** dialog box, click **Yes**.
9. After a few seconds (the polling interval of Activity Monitor) notice the **SQLCMD** connections disappear from the list. Sessions are no longer blocked by the hanging transaction.
10. Close all the application windows.

Results: At the end of this exercise, you will have resolved a performance issue.

Module 15: Importing and Exporting Data

Lab: Importing and Exporting Data

Exercise 1: Import Excel Data Using the Import Wizard

▶ Task 1: Prepare the Lab Environment

1. Ensure that the **MT17B-WS2016-NAT**, **20764C-MIA-DC**, and **20764C-MIA-SQL** virtual machines are running.
2. Log on to **20764C-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa55w.rd**.
3. In the **D:\Labfiles\Lab15\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
4. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.

Enable Excel data import

1. On the taskbar click Internet Explorer.
2. In the address bar type the following URL, and then press **enter**:

Microsoft Access Database Engine 2016 Redistributable download

https://aka.ms/Tjh400

3. On the webpage click **Download**.
4. On the **Choose the download you want** page, click **AccessDatabaseEngine_X64.exe**, then click **Next**.
5. On the popup at the bottom of the browser, click **Run**.
6. In the **User Account Control** dialog box, click **Yes**.
7. In the **Microsoft Access Database engine 2016 (English) Setup** dialog, click **Next**.
8. On the **End-User License Agreement page**, click **I accept the terms in the License Agreement**, then click **Next**.
9. Click **Install**.
10. Click **OK**.

▶ Task 2: Examine the Excel File

1. In File Explorer, navigate to **D:\Labfiles\Lab15\Starter\Import**, then double-click the **currency_codes.csv** file to open it.
2. Review the contents of the file. Close the file when you have finished your review.

▶ Task 3: Run the Data Import Wizard

1. On the **Start** menu, type **SQL Server 2017 Import and Export Data (64-bit)**, and then press Enter.
2. In the **SQL Server Import and Export Wizard** window, click **Next**.
3. On the **Choose a Data Source** page, in the **Data Source** box, click **Microsoft Excel**.
4. In the **Excel file path** box, type **D:\Labfiles\Lab15\Starter\Import\currency_codes.xlsx**.
5. In the **Excel Version** list, click **Microsoft Excel 2016**, and then click **Next**.

6. On the **Choose a Destination** page, in the **Destination** list, click **SQL Server Native Client 11.0**.
7. Verify that the **Server Name** box contains the value **(local)**, and that **Use Windows Authentication** is selected.
8. In the **Database** box, click **AdventureWorks**, and then click **Next**.
9. On the **Specify Table Copy or Query** page, verify that **Copy data from one or more tables or views** is selected, and then click **Next**.
10. On the **Select Source Tables and Views** page, in the first **Destination: (local)** list, click the dropdown and select **[Accounts].[CurrencyCode]**, and then click **Edit Mappings**.
11. In the **Column Mappings** dialog box, verify that **Append rows to destination table** is selected.
12. In the **Mappings** box, in the first **Destination** box to the right of **alphabetic_code**, click **currency_code**.
13. In the second **Destination** box to the right of **numeric_code**, click **currency_numeric_code**, and then click **OK**.
14. On the **Select Source Tables and Views** page, click **Next**.
15. On the **Review Data Type Mapping** page, click **Next**.
16. On the **Save and Run Package** page, verify that **Run immediately** is selected, then click **Finish**.
17. On the **Complete the Wizard** page, click **Finish**.
18. Note that **178 rows** have been transferred.
19. When execution of the package has finished, click **Close**.
20. On the taskbar, click **Microsoft SQL Server Management Studio**.
21. In the **Connect to Server** dialog box, click **Connect**.
22. On the **File** menu, point to **Open**, and click **Project/Solution**.
23. In the **Open Project** dialog box, navigate to **D:\Labfiles\Lab15\Starter\Project**, and then double-click **Project.ssmssln**.
24. In Solution Explorer, expand **Queries**, and then double-click **Lab Exercise 01 - currency codes.sql**.
25. Click **Execute** to review the data you have loaded.
26. On the **File** menu, click **Close**.
27. Leave SQL Server Management Studio open for the next exercise.

Results: After completing this exercise, you will be able to:

Use the Import and Export Wizard to import data.

Exercise 2: Import a Delimited Text File Using bcp

▶ Task 1: Import Data Using bcp

1. On the **Start** screen, type **cmd**, and then click **Command Prompt**.
2. At the command prompt, type the following code, and then press Enter:

```
bcp AdventureWorks.Accounts.ExchangeRate in
D:\Labfiles\Lab15\Starter\Import\currency_exchange_rates.txt -S MIA-SQL -T -c -t ,
```

3. Wait for the load to complete.

▶ Task 2: Trusting Foreign Key Constraints

1. In SQL Server Management Studio, in Solution Explorer, double-click **Lab Exercise 02 - bcp.sql**.
2. Highlight the query under the heading for **Task 1** and click **Execute**. Note that the foreign key constraints are not trusted.
3. Highlight the query under the heading for **Task 2** and click **Execute**.
4. Highlight the query under the heading for **Task 1** and click **Execute**. Note that the foreign key constraints are now trusted.

Results: At the end of this exercise, you will be able to:

Import a delimited text file using **bcp**.

Exercise 3: Import a Delimited Text File Using BULK INSERT

▶ Task 1: Clear Down the Accounts.ExchangeRate Table

1. In Solution Explorer, double-click **Lab Exercise 03 - BULK INSERT.sql**.
2. Highlight the query under the heading for **Task 1** and click **Execute**.

▶ Task 2: Import Data Using BULK INSERT

1. Edit the query under the heading for **Task 2**, so that it reads:

```
BULK INSERT AdventureWorks.Accounts.ExchangeRate
FROM 'D:\Labfiles\Lab15\Starter\Import\currency_exchange_rates.txt'
WITH ( FIELDTERMINATOR =',',
       ROWTERMINATOR ='\n',
       CHECK_CONSTRAINTS
     );
GO
```

2. Highlight the query you have amended and click **Execute**.

▶ Task 3: Check Foreign Key Trust Status

- Highlight the query under the heading for **Task 3** and click **Execute**. Note that the foreign key constraints are trusted, because you used the CHECK_CONSTRAINT option in your BULK INSERT query.

Results: After completing this exercise, you will be able to:

Import a delimited text file using BULK INSERT.

Exercise 4: Create and Test an SSIS Package to Extract Data

▶ Task 1: Create an SSIS Package

1. Start Visual Studio® 2015.
2. On the **File** menu, point to **Open**, and then click **Project/Solution**.
3. In the **Open Project** dialog box, navigate to **D:\Labfiles\Lab15\Starter\prospect_export**, and then double-click **prospect_export.sln**.
4. In Solution Explorer, right-click **SSIS Packages**, and then click **New SSIS Package**.
5. Right-click **Package1.dtsx**, click **Rename**, type **prospects**, and then press Enter.
6. In the **prospects.dtsx [Design]** pane, click **Control Flow**.
7. In SSIS Toolbox, double-click **Data Flow Task**.
8. Right-click **Data Flow Task**, click **Rename**, type **Export Prospects**, and then press Enter.
9. Double-click **Export Prospects** to open the **Data Flow** page.

▶ Task 2: Add a Data Source

1. In SSIS Toolbox, double-click **Source Assistant**.
2. In the **Source Assistant - Add New Source** dialog box, in the **Select source type** box, click **SQL Server**.
3. In the **Select connection managers** box, click **New**, and then click **OK**.
4. In the **Connection Manager** dialog box, in the **Server name** box, type **MIA-SQL**, and verify that **Use Windows Authentication** is selected.
5. In the **Select or enter a database name** list, click **AdventureWorks**, and then click **OK**.
6. Right-click **OLE DB Source**, click **Rename**, type **usp_prospect_list**, and then press Enter.
7. Double-click **usp_prospect_list** to configure the source.
8. In the **OLE DB Source Editor** dialog box, on the **Connection Manager** page, in the **Data access mode** list, click **SQL command**.
9. In the **SQL command text** box, type **EXEC Sales.usp_prospect_list;**, and then click **OK**.

▶ Task 3: Add a File Target

1. In SSIS Toolbox, expand **Other Destinations**, click and drag **Flat File Destination** to the **Data Flow** page.
2. Right-click **Flat File Destination**, click **Rename**, type **prospects file**, and then press Enter.
3. Click **usp_prospect_list**, click on the left (blue) arrow underneath the **usp_prospect_list** object, and then click **prospects file**.
4. Double-click **prospects file** to configure it.
5. In the **Flat File Destination Editor** dialog box, click **New**.

6. In the **Flat File Format** dialog box, click **Delimited**, and then click **OK**.
7. In the **Flat File Connection Manager Editor** dialog box, on the **General** page, in the **Connection manager name** box, type **prospect file connection**.
8. In the **File name** box, type **D:\Labfiles\Lab15\Starter\Export\prospects.txt**.
9. Select the **Column names in the first data row** check box.
10. On the **Columns** page, in the **Row Delimiter** box, verify that **{CR}{LF}** is selected.
11. In the **Column delimiter** box, verify that **Comma {,}** is selected, and then click **OK**.
12. In the **Flat File Destination Editor** dialog box, on the **Mappings** page, note the mappings between the **Input Column** and the **Destination Column**, and then click **OK**.

▶ Task 4: Test the Package

1. To test the package, on the **Debug** menu, click **Start Debugging**. The package will run almost instantly.
2. On the **Debug** menu, click **Stop Debugging** to return to design mode.
3. In File Explorer, navigate to **D:\Labfiles\Lab15\Starter\Export**, double-click **prospects.txt**, and verify that it contains data.
4. When you have finished your review, close Notepad, and then close Visual Studio, without saving any changes.

Results: At the end of this exercise, you will be able to:

Create and test an SSIS package.

Exercise 5: Deploy a Data-Tier Application

▶ Task 1: Review the Contents of a DACPAC

1. In File Explorer, navigate to **D:\Labfiles\Lab15\Starter\FixedAssets**, right-click **FixedAssets_1.0.9.1.dacpac**, and then click **Unpack**.
2. In the **Unpack Microsoft SQL Server DAC Package File** dialog box, in the **Files will be unpacked to this folder** box, verify the location is **D:\Labfiles\Lab15\Starter\FixedAssets\FixedAssets_1.0.9.1**, and then click **Unpack**.
3. When unpacking is complete, in the **FixedAssets_1.0.9.1** window, double-click the **model.sql** script to review how the DACPAC is going to behave.

▶ Task 2: Deploy the Data-Tier Application

1. In SQL Server Management Studio, in Object Explorer, under **MIA-SQL**, right-click **Databases**, and then click **Deploy Data Tier Application**.
2. In the **Deploy Data-tier Application** dialog box, on the **Introduction** page, click **Next**.
3. On the **Select Package** page, in the **DAC package (file name with the .dacpac extension)** box, type **D:\Labfiles\Lab15\Starter\FixedAssets\FixedAssets_1.0.9.1.dacpac**, and then click **Next**.
4. On the **Update Configuration** page, click **Next**.
5. On the **Summary** page, click **Next**.

6. On the **Deploy DAC** page, when deployment is complete, click **Finish**.
7. In Object Explorer, under **MIA-SQL**, right-click **Databases**, and then click **Refresh**.
8. Expand **Databases**, and verify that the **FixedAssets_1.0.9.1** database has been created successfully.

Results: After completing this exercise, you will be able to:

Review the contents of a DACPAC.

Deploy a data-tier application.