

OFFICIAL MICROSOFT LEARNING PRODUCT

20761B

Querying Data with Transact-SQL

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2017 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners

Product Number: 20761B

Part Number (if applicable):

Released: 02/2017

Module 1

Introduction to Microsoft SQL Server 2016

Contents:

Lesson 1: The Basic Architecture of SQL Server	2
Lesson 2: SQL Server Editions and Versions	4
Lesson 3: Getting Started with SQL Server Management Studio	6
Module Review and Takeaways	9

Lesson 1

The Basic Architecture of SQL Server

Contents:

Question and Answers

3

Question and Answers

Put the following T-SQL commands in order by numbering each to create a script that will execute without errors:

	Steps
	<pre>CREATE TABLE HR.Employees (EmployeeID int PRIMARY KEY, LastName nvarchar(25), FirstName nvarchar(25));</pre>
	GO
	<pre>INSERT INTO HR.Employees (EmployeeID, LastName, FirstName) VALUES (121, N'O'Neill, N'Carlene');</pre>
	GO

Answer:

	Steps
1	<pre>CREATE TABLE HR.Employees (EmployeeID int PRIMARY KEY, LastName nvarchar(25), FirstName nvarchar(25));</pre>
2	GO
3	<pre>INSERT INTO HR.Employees (EmployeeID, LastName, FirstName) VALUES (121, N'O'Neill, N'Carlene');</pre>
2	GO

Lesson 2

SQL Server Editions and Versions

Contents:

Question and Answers	5
Resources	5

Question and Answers

SQL Server Versions

Question: Which version of SQL Server are you currently working with? Have you worked with any earlier versions?

Answer: Answers will vary.

SQL Server Editions

Question: You have founded a new company with two friends. Your new application (app) uses a SQL Server database to store information. You are unsure whether your app will be successful but, if it is, you will need both high performance and space for large volumes of data. However, you have not yet launched, so are unsure how many people will use your app. Which edition of SQL Server 2016 should you use for this system?

- ☐ Azure SQL Database
- ☐ Enterprise edition
- ☐ Express edition
- ☐ Business Intelligence edition
- ☐ Any edition is appropriate for these requirements

Answer:

- ☒ Azure SQL Database
- ☐ Enterprise edition
- ☐ Express edition
- ☐ Business Intelligence edition
- ☐ Any edition is appropriate for these requirements

Resources



Additional Reading: For more information on the use of T-SQL on Microsoft Azure SQL Server Database, see the MSDN article *Azure SQL Database Transact-SQL Information* at: <https://azure.microsoft.com/en-gb/documentation/articles/sql-database-transact-sql-information/>

Lesson 3

Getting Started with SQL Server Management Studio

Contents:

Question and Answers	7
Resources	7
Demonstration: Introducing Microsoft SQL Server 2016	7

Question and Answers

Connecting to SQL Server

Question: In your organization, which authentication method do you use to log on to SQL Server?

Answer: Either Windows Authentication or SQL Server Authentication.

Introducing Microsoft SQL Server 2016

Question: A colleague has asked you to run some test queries against the company's scheduling database. Administrators have given you the name of the server where the database is hosted, and the name of the database. Permissions to run the necessary queries have been granted to your Active Directory® account. You are logged on to a client computer with this Active Directory account and have started SQL Server Management Studio. What other information do you need to connect to the database?

- ☐ Your Active Directory account username.
- ☐ Your Active Directory account password.
- ☐ The name of the login created for you in the SQL Server instance.
- ☐ The name of the instance that hosts the database.
- ☐ The name of the user created for you in the SQL Server database.

Answer:

- ☐ Your Active Directory account username.
- ☐ Your Active Directory account password.
- ☐ The name of the login created for you in the SQL Server instance.
- ☒ The name of the instance that hosts the database.
- ☐ The name of the user created for you in the SQL Server database.

Resources

Executing Queries



Additional Reading: For a list of keyboard shortcuts available in SSMS, see *SQL Server Management Studio Keyboard Shortcuts*, in MSDN.

Demonstration: Introducing Microsoft SQL Server 2016

Demonstration Steps

Use SSMS to Connect to an On-premises instance of SQL Server 2016

1. Ensure that the **20761B-MIA-DC** and **20761B-MIA-SQL** virtual machines are running.
2. Log on to **20761B-MIA-SQL** as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
3. In the **D:\Demofiles\Mod01** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
4. In the **User Account Control** dialog box, click **Yes**, press **y** when prompted, and then press Enter.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.

Explore Database and Other Objects

1. In Object Explorer, expand the **Databases** folder to see a list of databases.
2. Expand the **TSQL** database.
3. Expand the **Tables** folder.
4. Expand the **Sales.Customers** table.
5. Expand the **Columns** folders.
6. View the list of columns, and the data type information for each column.
7. Note the data type for the **companyname** column.

Work with T-SQL Scripts

1. If the Solution Explorer pane is not visible, on the **View** menu, click **Solution Explorer**.
2. In Solution Explorer, notice it will be empty.
3. On the **File** menu, point to **New**, and then click **Project**.
4. In the **New Project** dialog box, under **Installed**, click **SQL Server Management Studio Projects**.
5. In the middle pane, click **SQL Server Scripts**.
6. In the **Name** box, type **Module 1 Demonstration**.
7. In the **Location** box, type **D:\Demofiles\Mod01**.
8. Point out the solution name, and then click **OK**.
9. In Solution Explorer, right-click **Queries**, and click **New Query**.
10. Type the following T-SQL code:

```
USE TSQL;  
GO  
SELECT CustID, ShipCountry  
FROM Sales.Orders;
```

11. Select the code and click **Execute**.
12. Point out the **Results** pane.
13. On the **File** menu, click **Save All**.
14. On the **File** menu, click **Close Solution**.
15. On the **File** menu, point to **Recent Projects and Solutions**, and then click **1 D:\...\ Module 1 Demonstration\Module 1 Demonstration.ssmssln**.
16. Point out the **Solution Explorer** pane.
17. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Review Question(s)

Question: Can a SQL Server database be stored across multiple instances?

Answer: No. A database is completely contained within a single instance.

Question: If no T-SQL code is selected in a query window, which code lines will be run when you click the Execute button?

Answer: All statements in the script will be executed.

Question: What does a SQL Server Management Studio solution contain?

Answer: SSMS allows you to organize SQL Scripts so that you can manage large collections of files. Projects can contain scripts, connection strings and other settings. Solutions are collections of projects.

Module 2

Introduction to T-SQL Querying

Contents:

Lesson 1: Introducing T-SQL	2
Lesson 2: Understanding Sets	4
Lesson 3: Understanding Predicate Logic	7
Lesson 4: Understanding the Logical Order of Operations in SELECT Statements	9
Module Review and Takeaways	12

Lesson 1

Introducing T-SQL

Contents:

Question and Answers	3
Demonstration: T-SQL Language Elements	3

Question and Answers

Question: From the following T-SQL elements, select the one that does not contain an expression:

- () SELECT FirstName, LastName, SkillName AS Skill, GetDate() - DOB AS Age
- () WHERE HumanResources.Department.ModifiedDate > (SYSDATETIME() - 31)
- () JOIN HumanResources.Skills ON Employees.ID = Skills.EmployeeID
- () WHERE Skill.Level + Skill.Confidence > 10

Answer:

- () SELECT FirstName, LastName, SkillName AS Skill, GetDate() - DOB AS Age
- () WHERE HumanResources.Department.ModifiedDate > (SYSDATETIME() - 31)
- (√) JOIN HumanResources.Skills ON Employees.ID = Skills.EmployeeID
- () WHERE Skill.Level + Skill.Confidence > 10

Demonstration: T-SQL Language Elements

Demonstration Steps

Use T-SQL Language Elements

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In File Explorer, browse to **D:\Demofiles\Mod02**, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. When the script has finished, press any key.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows® Authentication.
6. On the **File** menu, point to **Open**, and then click **Project/Solution**.
7. In the **Open Project** dialog box, browse to the **D:\Demofiles\Mod02\Demo** folder, and then double-click **Demo.ssmssl.n**.
8. In Solution Explorer, expand **Queries**, and then double-click the **11 - Demonstration A.sql** script file.
9. Select the code under **Step 1**, and then click **Execute**.
10. Select the code under **Step 2**, and then click **Execute**.
11. Select the code under **Step 3**, and then click **Execute**.
12. Select the code under **Step 4**, and then click **Execute**.
13. Select the code under **Step 5**, and then click **Execute**.
14. Select the code under **Step 6**, and then click **Execute**.
15. Select the code under **Step 7**, and then click **Execute**.
16. Select the code under **Step 8**, and then click **Execute**.
17. Select the code under the comment **Cleanup task if needed**, and then click **Execute**.
18. Close SQL Server Management Studio.

Lesson 2

Understanding Sets

Contents:

Question and Answers	5
Resources	6

Question and Answers

Place each employee into the appropriate set. Indicate your answer by writing the set number to the right of each item.

Items	
1	Carolos Lamy Works in: London Skills: JavaScript XML
2	Naiyana Kunakorn Works in: Washington DC Skills: JavaScript SQL Server Administration T-SQL XML
3	Zachary Parsons Works in: Seattle Skills: Active Directory Administration SharePoint Administration SQL Server Administration
4	Patrick Lorenzen Works in: London Skills: SharePoint Administration SQL Server Administration
5	Frederic Towle Works in: Tokyo Skills: Active Directory Administration T-SQL
6	Nickolas McLaughlin Works in: Seattle Skills: C# JavaScript SQL Server Administration
7	Jeanie Sheppard Works in: Buenos Aires Skills: C# JavaScript T-SQL

Category 1		Category 2		Category 3
Employees in London		Employees who know T-SQL		Employees in Seattle who know SQL Server Administration

Answer:

Category 1		Category 2		Category 3
Employees in London		Employees who know T-SQL		Employees in Seattle who know SQL Server Administration
Carolos Lamy Works in: London Skills: JavaScript XML Patrick Lorenzen Works in: London Skills: SharePoint Administration SQL Server Administration		Naiyana Kunakorn Works in: Washington DC Skills: JavaScript SQL Server Administration T-SQL XML Frederic Towle Works in: Tokyo Skills: Active Directory Administration T-SQL Jeanie Sheppard Works in: Buenos Aires Skills: C# JavaScript T-SQL		Zachary Parsons Works in: Seattle Skills: Active Directory Administration SharePoint Administration SQL Server Administration Nickolas McLaughlin Works in: Seattle Skills: C# JavaScript SQL Server Administration

Resources

Set Theory Applied to SQL Server Queries



Additional Reading: For more information on set theory and logical query processing, and its application to SQL queries, see Chapter 1 of Itzik Ben-Gan's *T-SQL Querying* (Microsoft Press, 2015).

Lesson 3

Understanding Predicate Logic

Contents:

Question and Answers

8

Question and Answers

Question: From the following T-SQL elements, select the one that can include a predicate:

- ☐ WHERE clauses
- ☐ JOIN conditions
- ☐ HAVING clauses
- ☐ WHILE statements
- ☐ All of the above

Answer:

- ☐ WHERE clauses
- ☐ JOIN conditions
- ☐ HAVING clauses
- ☐ WHILE statements
- ☒ All of the above

Lesson 4

Understanding the Logical Order of Operations in SELECT Statements

Contents:

Question and Answers	10
Demonstration: Logical Query Processing	10

Question and Answers

Put the following T-SQL elements in order by numbering each to indicate the order that SQL Server will process them in when they appear in a single SELECT statement.

	Steps
	FROM
	WHERE
	GROUP BY
	HAVING
	SELECT
	ORDER BY

Answer:

	Steps
1	FROM
2	WHERE
3	GROUP BY
4	HAVING
5	SELECT
6	ORDER BY

Demonstration: Logical Query Processing

Demonstration Steps

View Query Output That Illustrates Logical Processing Order

1. Start the 20761B-MIA-DC, 20761B-MIA-SQL, MSL-TMG1 virtual machines, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Open SQL Server Management Studio.
3. In the **Connect to Server** dialog box, in the **Server name** box, enter the server you created during preparation. For example, **20761Ba-azure.database.windows.net**.
4. In the **Authentication** list, click **SQL Server Authentication**.
5. In the **Login** box, type **Student**.
6. In the **Password** box, type **Pa\$\$w0rd**, and then click **Connect**.
7. On the **File** menu, point to **Open**, and then click **Project/Solution**.
8. In the **Open Project** dialog box, browse to the **D:\Demofiles\Mod02\Demo** folder, and then double-click **Demo.ssmssl.n**.
9. In Solution Explorer, double-click the **21 - Demonstration B.sql** script file.

10. On the **Query** menu, point to **Connection**, and then click **Change Connection**.
11. In the **Connect to Database Engine** dialog box, in the **Server name** box, enter the server you created during preparation. For example, **20761Ba-azure.database.windows.net**.
12. In the **Authentication** list, click **SQL Server Authentication**.
13. In the **Login** box, type **Student**.
14. In the **Password** box, type **Pa\$\$w0rd**, and then click **Connect**.
15. In the **Available Databases** list, click **AdventureWorksLT**.
16. Select the code under the comment **Step 1**, and then click **Execute**.
17. Select the code under **Step 2**, and then click **Execute**.
18. Select the code under **Step 3**, and then click **Execute**.
19. Select the code under **Step 4**, and then click **Execute**.
20. Select the code under **Step 5**, and then click **Execute**. Note the error message.
21. Select the code under **Step 6**, and then click **Execute**.
22. Select the code under **Step 7**, and then click **Execute**.
23. Select the code under **Step 8**, and then click **Execute**.
24. Close SQL Server Management Studio, without saving any changes.

Module Review and Takeaways

Question: Which category of T-SQL statements concerns querying and modifying data?

Answer: DML.

Question: What are some examples of aggregate functions supported by T-SQL?

Answer: SUM, MIN, COUNT, COUNTBIG, MAX, AVG.

Question: Which SELECT statement element will be processed before a WHERE clause?

Answer: FROM.

Module 3

Writing SELECT Queries

Contents:

Lesson 1: Writing Simple SELECT Statements	2
Lesson 2: Eliminating Duplicates with DISTINCT	4
Lesson 3: Using Column and Table Aliases	6
Lesson 4: Writing Simple CASE Expressions	8
Module Review and Takeaways	10

Lesson 1

Writing Simple SELECT Statements

Contents:

Question and Answers	3
Demonstration: Writing Simple SELECT Statements	3

Question and Answers

Question: You have a table named Sales with the following columns: Country, NumberOfReps, TotalSales.

You want to find out the average amount of sales a sales representative makes in each country. What SELECT query could you use?

Answer: SELECT Country, (TotalSales / NumberOfReps) AS AverageSalesPerRep
FROM Sales;

Demonstration: Writing Simple SELECT Statements

Demonstration Steps

Use Simple SELECT Queries

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod03\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, and press Enter. When the script has completed, press any key.
5. Start SQL Server Management Studio and connect to the **Azure SQL** database engine instance using SQL Server authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod03\Demo** folder.
7. In Solution Explorer, expand **Queries**, and open the **Demonstration A.sql** script file. You may need to enter your password to connect to the **Azure SQL** database engine.
8. In the **Available Databases** list, click **AdventureWorksLT**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Select the code under the comment **Step 5**, and then click **Execute**.
13. Select the code under the comment **Step 6**, and then click **Execute**.
14. Select the code under the comment **Step 7**, and then click **Execute**.
15. On the **File** menu, click **Close**.
16. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Eliminating Duplicates with DISTINCT

Contents:

Question and Answers	5
Demonstration: Eliminating Duplicates with DISTINCT	5

Question and Answers

Question: You have company departments in five countries. You have the following query for the Human Resources database:

```
SELECT DeptName, Country
FROM HumanResources.Departments
```

This returns:

DeptName	Country
-----	-----
Sales	UK
Sales	USA
Sales	France
Sales	Japan
Marketing	USA
Marketing	Japan
Research	USA

You add a **DISTINCT** keyword to the **SELECT** query. How many rows are returned?

Answer: 7

Demonstration: Eliminating Duplicates with DISTINCT

Demonstration Steps

Eliminate Duplicate Rows

1. In Solution Explorer, open the **Demonstration B.sql** script file. You may need to enter your password to connect to the **Azure SQL** database engine.
2. In the **Available Databases** list, click **AdventureWorksLT**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. On the **File** menu, click **Close**.
7. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Using Column and Table Aliases

Contents:

Question and Answers	7
Demonstration: Using Column and Table Aliases	7

Question and Answers

Question: You have the following query:

```
SELECT FirstName LastName
FROM HumanResources.Employees;
```

You are surprised to find that the query returns the following:

LastName

Fred

Rosalind

Anil

Linda

What error have you made in the SELECT query?

Answer: You have omitted a comma between FirstName and LastName.

Use Aliases to Refer to Columns

Question: Which of the following statements use correct column aliases?

SELECT Name AS ProductName FROM Production.Product

SELECT Name = ProductName FROM Production.Product

SELECT ProductName == Name FROM Production.Product

SELECT ProductName = Name FROM Production.Product

SELECT Name AS Product Name FROM Production.Product

Answer: Statements 1 and 4 are correct.

Demonstration: Using Column and Table Aliases

Demonstration Steps

Use Column and Table Aliases

1. In Solution Explorer, open the **Demonstration C.sql** script file. You may need to enter your password to connect to the **Azure SQL** database engine.
2. In the **Available Databases** list, click **AdventureWorksLT**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. On the **File** menu, click **Close**.
8. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Writing Simple CASE Expressions

Contents:

Question and Answers	9
Demonstration: Simple CASE Expressions	9

Question and Answers

Question: You have the following SELECT query:

```
SELECT FirstName, LastName, Sex
FROM HumanResources.Employees;
```

This returns:

```
FirstName LastName Sex
-----
```

```
Maya      Steele    1
Adam      Brookes   0
Naomi     Sharp     1
Pedro     Fielder   0
Zachary   Parsons   0
```

How could you make these results clearer?

Answer: Use the following query:

```
SELECT FirstName, LastName, Gender =
CASE Sex
    WHEN 1 THEN 'Female'
    WHEN 0 THEN 'Male'
    ELSE 'Unspecified'
END
FROM HumanResources.Employees;
```

Demonstration: Simple CASE Expressions

Demonstration Steps

Use a Simple CASE Expression

1. In Solution Explorer, open the **Demonstration D.sql** script file. You may need to enter your password to connect to the **Azure SQL** database engine.
2. In the **Available Databases** list, click **AdventureWorksLT**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Close SQL Server Management Studio, without saving changes.

Module Review and Takeaways

Best Practice

Terminate all T-SQL statements with a semicolon. This will make your code more readable, avoid certain parsing errors, and protect your code against changes in future versions of SQL Server.

Consider standardizing your code on the AS keyword for labeling column and table aliases. This will make it easier to read and avoids accidental aliases.

Review Question(s)

Question: Why is the use of SELECT * not a recommended practice?

Answer: There are two answers:

- 1) * asks for all columns, which is typically too much.
- 2) Query is exposed to changes in the underlying table structure and could return unexpected results.

Real-world Issues and Scenarios

You can create a column alias without using the AS keyword, something you are likely to see in code samples online, or written by developers you work with. While the T-SQL engine will parse this without issue, there is a problem when a comma is omitted between column names—the first column will take the name of the second column as its alias. Not only will the column have a misleading name, but you will also have one column too few in your result set. Always use the AS keyword to avoid this problem.

Module 4

Querying Multiple Tables

Contents:

Lesson 1: Understanding Joins	2
Lesson 2: Querying with Inner Joins	4
Lesson 3: Querying with Outer Joins	6
Lesson 4: Querying with Cross Joins and Self Joins	9
Module Review and Takeaways	11

Lesson 1

Understanding Joins

Contents:

Question and Answers	3
Demonstration: Understanding Joins	3

Question and Answers

Question: You have the following T-SQL query:

```
SELECT o.ID AS OrderID, o.CustomerName, p.ProductName, p.ModelNumber,
FROM Sales.Orders AS o
JOIN Sales.Products AS p
ON o.ProductID = p.ID;
```

Which of the following types of join will the query perform?

- ☐ A cross join
- ☐ An inner join
- ☐ An outer left join
- ☐ An outer right join

Answer:

- ☐ A cross join
- ☒ An inner join
- ☐ An outer left join
- ☐ An outer right join

Demonstration: Understanding Joins

Demonstration Steps

Use Joins

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod04\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, and then press Enter. When the script has completed, press any key.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod04\Demo** folder.
7. In Solution Explorer, expand **Queries**, and then double-click the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Select the code under the comment **Step 5**, and then click **Execute**.
13. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Querying with Inner Joins

Contents:

Question and Answers	5
Demonstration: Querying with Inner Joins	5

Question and Answers

Question: You have the following T-SQL query:

```
SELECT HumanResources.Employees.ID, HumanResources.Employers.ID AS  
CompanyID,  
       HumanResources.Employees.Name, HumanResources.Employers.Name AS  
CompanyName  
FROM HumanResources.Employees  
JOIN HumanResources.Employers  
ON HumanResources.Employees.EmployerID = HumanResources.Employers.ID;
```

How can you improve the readability of this query?

Answer: Use AS to create more readable aliases for the Employees and Employers tables.

Demonstration: Querying with Inner Joins

Demonstration Steps

Use Inner Joins

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Querying with Outer Joins

Contents:

Question and Answers	7
Demonstration: Querying with Outer Joins	7

Question and Answers

Question: You have a table named PoolCars and a table named Bookings in your ResourcesScheduling database. You want to return all the pool cars for which there are zero bookings. Which of the following queries should you use?

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc, ResourcesScheduling.Bookings AS b
WHERE pc.ID = b.CarID;

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
RIGHT OUTER JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID;

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID;

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
LEFT OUTER JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID
WHERE b.BookingID IS NULL;

Answer:

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc, ResourcesScheduling.Bookings AS b
WHERE pc.ID = b.CarID;

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
RIGHT OUTER JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID;

() SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID;

(√) SELECT pc.ID, pc.Make, pc.Model, pc.LicensePlate
FROM ResourcesScheduling.PoolCars AS pc
LEFT OUTER JOIN ResourcesScheduling.Bookings AS b
ON pc.ID = b.CarID
WHERE b.BookingID IS NULL;

Demonstration: Querying with Outer Joins

Demonstration Steps

Use Outer Joins

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.

5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Querying with Cross Joins and Self Joins

Contents:

Question and Answers	10
Demonstration: Querying with Cross Joins and Self Joins	10

Question and Answers

Question: You have two tables named FirstNames and LastNames. You want to generate a set of fictitious full names from this data. There are 150 entries in the FirstNames table and 250 entries in the LastNames table. You use the following query:

```
SELECT (f.Name + ' ' + l.Name) AS FullName  
FROM FirstNames AS f  
CROSS JOIN LastNames AS l
```

How many fictitious full names will be returned by this query?

Answer: 37,500 names

Demonstration: Querying with Cross Joins and Self Joins

Demonstration Steps

Use Self Joins and Cross Joins

1. In Solution Explorer, open the **41 - Demonstration D.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Best Practice

Table aliases should always be defined when joining tables.

Joins should be expressed using SQL-92 syntax, with JOIN and ON keywords.

Review Question(s)

Question: How does an inner join differ from an outer join?

Answer: An inner join filters out rows that do not satisfy the predicate in the ON clause. An outer join includes all rows from both tables and includes NULLs for attributes where no match is found.

Question: Which join types include a logical Cartesian product?

Answer: CROSS, INNER and OUTER

Question: Can a table be joined to itself?

Answer: Yes, as a self join. An alias to at least one table is required in the FROM clause.

Module 5

Sorting and Filtering Data

Contents:

Lesson 1: Sorting Data	2
Lesson 2: Filtering Data with Predicates	4
Lesson 3: Filtering Data with TOP and OFFSET-FETCH	6
Lesson 4: Working with Unknown Values	9
Module Review and Takeaways	11
Lab Review Questions and Answers	12

Lesson 1

Sorting Data

Contents:

Question and Answers	3
Demonstration: Sorting Data	3

Question and Answers

Question: If you declare an alias for a column in the SELECT clause, you cannot use that alias in the WHERE clause—but you can use it in the ORDER BY clause. Why is this?

Answer: This is because of the order in which clauses of a SELECT query are processed. The WHERE clause is processed before the SELECT column list; therefore, column aliases are not available for sorting. The ORDER BY clause is processed last, so column aliases are available and can be used without errors.

Demonstration: Sorting Data

Demonstration Steps

Sort Data Using The ORDER BY Clause

1. Ensure that the MSL-TMG1, 20761B-MIA-DC, and 20761B-MIA-SQL virtual machines are running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Start SQL Server Management Studio and connect to your Azure instance of the **AdventureWorksLT** database engine instance using SQL Server authentication.
3. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod05\Demo** folder.
4. In Solution Explorer, expand **Queries**, and then double-click **11 - Demonstration A.sql**.
5. In the **Available Databases** list, click **ADVENTUREWORKSLT**.
6. Select the code under the comment **Step 1**, and then click **Execute**.
7. Select the code under the comment **Step 2**, and then click **Execute**.
8. Select the code under the comment **Step 3**, and then click **Execute**.
9. Select the code under the comment **Step 4**, and then click **Execute**.
10. Select the code under the comment **Step 5**, and then click **Execute**.
11. Select the code under the comment **Step 6**, and then click **Execute**.
12. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Filtering Data with Predicates

Contents:

Question and Answers	5
Demonstration: Filtering Data with Predicates	5

Question and Answers

Question: You have a table named Employees that includes a column named StartDate. You want to find who started in any year other than 2014. What query would you use?

Answer: Multiple answers are possible. For example:

```
SELECT FirstName, LastName
```

```
FROM Employees
```

```
WHERE Employees.StartDate < '20140101' OR Employees.StartDate >= '20150101';
```

Demonstration: Filtering Data with Predicates

Demonstration Steps

Filter Data in a WHERE Clause

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. In the **Available Databases** list, click **ADVENTUREWORKSLT**.
3. Select the code under the comment **Step 1**, and then click **Execute**.
4. Select the code under the comment **Step 2**, and then click **Execute**.
5. Select the code under the comment **Step 3**, and then click **Execute**. Note the error message.
6. Select the code under the comment **Step 4**, and then click **Execute**.
7. Select the code under the comment **Step 5**, and then click **Execute**.
8. Select the code under the comment **Step 6**, and then click **Execute**.
9. Select the code under the comment **Step 7**, and then click **Execute**.
10. Select the code under the comment **Step 8**, and then click **Execute**.
11. Select the code under the comment **Step 9**, and then click **Execute**.
12. Select the code under the comment **Step 10**, and then click **Execute**.
13. Select the code under the comment **Step 11**, and then click **Execute**.
14. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Filtering Data with TOP and OFFSET-FETCH

Contents:

Question and Answers	7
Demonstration: Filtering Data with TOP and OFFSET-FETCH	7

Question and Answers

Question: You have a table named Products in your Sales database. You are creating a paged display of products in an application that shows 20 products on each page, ordered by name. Which of the following queries would return the third page of products?

- () SELECT ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
OFFSET 60 ROWS FETCH NEXT 20 ROWS ONLY
- () SELECT ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
OFFSET 40 ROWS FETCH NEXT 20 ROWS ONLY;
- () SELECT TOP (20) ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
- () SELECT TOP (20) WITH TIES ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC

Answer:

- () SELECT ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
OFFSET 60 ROWS FETCH NEXT 20 ROWS ONLY
- (v) SELECT ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
OFFSET 40 ROWS FETCH NEXT 20 ROWS ONLY;
- () SELECT TOP (20) ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC
- () SELECT TOP (20) WITH TIES ProductID, ProductName, ProductNumber
FROM Sales.Products
ORDER BY ProductName ASC

Demonstration: Filtering Data with TOP and OFFSET-FETCH

Demonstration Steps

Filter Data Using TOP and OFFSET-FETCH

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. In the **Available Databases** list, ensure **ADVENTUREWORKSLT** is selected.
3. Select the code under the comment **Step 1**, and then click **Execute**.
4. Select the code under the comment **Step 2**, and then click **Execute**.
5. Select the code under the comment **Step 3**, and then click **Execute**.
6. Select the code under the comment **Step 4**, and then click **Execute**.
7. Select the code under the comment **Step 5**, and then click **Execute**.

8. Select the code under the comment **Step 6**, and then click **Execute**.
9. Select the code under the comment **Step 7**, and then click **Execute**.
10. Select the code under the comment **Step 8**, and then click **Execute**.
11. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Working with Unknown Values

Contents:

Question and Answers	10
Demonstration: Working with NULL	10

Question and Answers

Question: You have the following query:

```
SELECT e.Name, e.Age  
FROM HumanResources.Employees AS e  
WHERE YEAR(e.Age) < 1990;
```

Several employees have asked for their age to be removed from the Human Resources database, and this requested action has been applied to the database. Will the above query return these employees?

Answer: No.

Demonstration: Working with NULL

Demonstration Steps

Test for Null

1. In Solution Explorer, open the **41 - Demonstration D.sql** script file.
2. In the **Available Databases** list, ensure **ADVENTUREWORKSLT** is selected.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Close SQL Server Management Studio.

Module Review and Takeaways

Question: Does the physical order of rows in a SQL Server table guarantee any sort order in queries using the table?

Answer: No.

Question: You have the following query:

```
SELECT p.PartNumber, p.ProductName, o.Quantity
```

```
FROM Sales.Products AS p
```

```
LEFT OUTER JOIN Sales.OrderItems AS o
```

```
ON p.ID = o.ProductID
```

```
ORDER BY o.Quantity ASC
```

You have one new product that has yet to receive any orders. Will this product appear at the top or the bottom of the results?

Answer: At the top of the results.

Lab Review Questions and Answers

Lab: Sorting and Filtering Data

Question and Answers

Lab Review

Question: What is the difference between filtering using the TOP option, and filtering using the WHERE clause?

Answer: The TOP option can only be used to filter results based on the columns specified in the ORDER BY clause; and then it can only be used to return a count of rows (or a percentage of rows) from the top of that result set. There is no support for more complex filters that cannot be expressed in terms of sorting. TOP has no facility for filtering NULL values—NULL values in the columns included in the ORDER BY clause are always returned first.

Filters in the WHERE clause have no such limitations; you can specify complex filters and filters that handle NULL.

Module 6

Working with SQL Server 2016 Data Types

Contents:

Lesson 1: Introducing SQL Server 2016 Data Types	2
Lesson 2: Working with Character Data	5
Lesson 3: Working with Date and Time Data	7
Module Review and Takeaways	10

Lesson 1

Introducing SQL Server 2016 Data Types

Contents:

Question and Answers	3
Resources	4
Demonstration: SQL Server Data Types	4

Question and Answers

Place each item into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	tinyint
2	float
3	binary
4	int
5	real
6	varbinary
7	bigint
8	decimal
9	money
10	bit

Category 1		Category 2		Category 3
Exact Numeric Data Types		Approximate Numeric Data Types		Binary Data Types

Answer:

Category 1		Category 2		Category 3
Exact Numeric Data Types		Approximate Numeric Data Types		Binary Data Types
tinyint int bigint decimal money bit		float real		binary varbinary

Resources**Other Data Types**

Additional Reading: See course 20472-2: *Developing Microsoft SQL Server Databases* for additional information on the XML data type.



Additional Reading: See course 20472-2: *Developing Microsoft SQL Server Databases* for additional information on the **hierarchyid** data type.

Demonstration: SQL Server Data Types**Demonstration Steps**

Convert Data Types

1. Ensure that the 20761B-MIA-DC, and 20761B-MIA-SQL virtual machines are running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Start SQL Server Management Studio and connect to your Azure instance of the **AdventureWorksLT** database engine instance using SQL Server authentication.
3. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod06\Demo** folder.
4. In Solution Explorer, expand **Queries**, and then double-click **11 - Demonstration A.sql**.
5. In the **Available Databases** list, click **AdventureWorksLT**.
6. Select the code under the comment **Step 2**, and then click **Execute**.
7. Select the code under the comment **Step 3**, and then click **Execute**. Note the error message.
8. Select the code under the comment **Step 4**, and then click **Execute**.
9. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Working with Character Data

Contents:

Question and Answers	6
Demonstration: Working with Character Data	6

Question and Answers

Question: You have the following query:

```
SELECT FirstName
```

```
FROM HumanResources.Employees
```

```
WHERE FirstName LIKE N'^MA]%'
```

Will the query return an employee with the first name 'Matthew'?

Answer: No.

Demonstration: Working with Character Data

Demonstration Steps

Manipulate Character Data

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. In the **Available Databases** list, click **AdventureWorksLT**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3a**, and then click **Execute**.
5. Select the code under the comment **Step 3b**, and then click **Execute**.
6. Select the code under the comment **Step 4**, and then click **Execute**.
7. Select the code under the comment **Step 5**, and then click **Execute**.
8. Select all the code under the comment **Step 6**, and then click **Execute**.
9. Select the code under the comment **Step 7**, and then click **Execute**.
10. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Working with Date and Time Data

Contents:

Question and Answers	8
Demonstration: Working with Date and Time Data	9

Question and Answers

Place each item into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	datetime
2	datetime2
3	DATEFROMPARTS
4	smalldatetime
5	date
6	EOMONTH
7	time
8	datetimeoffset

Category 1		Category 2		Category 3
Present in all versions of SQL Server		Only present in SQL Server 2008 and later		Only present in SQL Server 2012 and later

Answer:

Category 1		Category 2		Category 3
Present in all versions of SQL Server		Only present in SQL Server 2008 and later		Only present in SQL Server 2012 and later
datetime smalldatetime		datetime2 date time datetimeoffset		DATEFROMPARTS EOMONTH

Demonstration: Working with Date and Time Data**Demonstration Steps**

Query Data and Time Values

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. In the **Available Databases** list, click **AdventureWorksLT**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: Will SQL Server be able to successfully and implicitly convert an **int** data type to a **varchar**?

Answer: No, **int** has a higher type precedence than **varchar**.

Question: What data type is suitable for storing Boolean flag information, such as TRUE or FALSE?

Answer: The **bit** data type.

Question: What logical operators are useful for retrieving ranges of date and time values?

Answer: **>=**, **<**

Module 7

Using DML to Modify Data

Contents:

Lesson 1: Adding Data to Tables	2
Lesson 2: Modifying and Removing Data	6
Lesson 3: Generating Automatic Column Values	9
Lab Review Questions and Answers	12

Lesson 1

Adding Data to Tables

Contents:

Question and Answers	3
Demonstration: Adding Data to Tables	3

Question and Answers

Using SELECT INTO

Question: You want to populate three columns of an existing table with data from another table in the same database. Which of the following types of query should you use?

- () INSERT INTO <TableName> (<Columns,...>) VALUES (<Column Value> ...)
- () INSERT INTO <DestinationTableName> SELECT <Columns> FROM <SourceTableName>
- () INSERT INTO <DestinationTableName> EXECUTE usp_SomeStoredProcedure
- () SELECT <Columns,...> INTO DestinationTableName FROM SourceTableName
- () SELECT <Columns,...> INTO SourceTableName FROM DestinationTableName

Answer:

- (v) INSERT INTO <TableName> (<Columns,...>) VALUES (<Column Value> ...)
- () INSERT INTO <DestinationTableName> SELECT <Columns> FROM <SourceTableName>
- () INSERT INTO <DestinationTableName> EXECUTE usp_SomeStoredProcedure
- () SELECT <Columns,...> INTO DestinationTableName FROM SourceTableName
- () SELECT <Columns,...> INTO SourceTableName FROM DestinationTableName

Demonstration: Adding Data to Tables

Demonstration Steps

INSERT Data into a Table

1. Start the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod07\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. In the Command Prompt window press **y**, and then press Enter.
5. When the script has finished, press Enter.
6. Open **SQL Server Management Studio**, and connect to the **MIA-SQL** database engine instance using Windows authentication.
7. On the **File** menu, point to **Open**, and then click **Project/Solution**.
8. In the **Open Project** dialog box, navigate to the **D:\Demofiles\Mod07\Demo** folder, click **Demo.ssmssl**, and then click **Open**.
9. In Solution Explorer, expand **Queries**, and double-click **11 - Demonstration A.sql**.
10. Highlight the code **USE TSQL GO**, and click **Execute**.

11. First you will populate a table with some data from a stored procedure. Highlight the code under the comment that begins **-- First try the INSERT by stored procedure:**

```
INSERT INTO Production.Products
```

```
(      productID
,      productname
,      supplierid
,      categoryid
,      unitprice)
```

```
EXEC Production.AddNewProducts;
```

12. Click **Execute**. You will receive a message saying that the procedure is not there.
13. Highlight the code below the comment **--Create a backup of the Products with a chosen ID**, and click **Execute**.

```
DROP TABLE IF EXISTS NewProducts
```

```
GO
```

```
SELECT * INTO NewProducts
```

```
FROM PRODUCTION.PRODUCTS WHERE ProductID >= 70
```

You are creating a new table for NewProducts where the Product ID >= 70.

14. You are also going to create a NewOrderDetails table that will contain rows for those products that have been transferred into NewProducts. To do this, highlight the code under the comment **-- Create a backup of the Order Details for the chosen productID**, up to the point shown in the code section for the next step below, and click **Execute**:

```
DROP TABLE IF EXISTS NewOrderDetails
```

```
GO
```

```
SELECT * INTO NewOrderDetails
```

```
FROM SALES.OrderDetails WHERE ProductID >= 70
```

```
-- Delete the copied data from the original tables
```

```
DELETE FROM SALES.OrderDetails
```

```
OUTPUT DELETED.*
```

```
WHERE ProductID >= 70
```

```
DELETE FROM Production.Products
```

```
OUTPUT DELETED.*
```

```
WHERE ProductID >= 70
```

```
-- check that they have been transferred safely
```

```
SELECT * FROM NewProducts
```

```
SELECT * FROM NewOrderDetails
```

```
SELECT * FROM SALES.OrderDetails
```

```
WHERE productid >= 70
```

```
SELECT * FROM Production.Products
WHERE productid >= 70
```

15. Highlight the code below the comment **Now we can put back the rows from the NewTables, using the INSERT statement**, and click **Execute**.

```
DROP PROCEDURE IF EXISTS Production.AddNewProducts
GO
CREATE PROCEDURE Production.AddNewProducts
AS
BEGIN
SELECT Productid, productname, SupplierID, CategoryID, Unitprice FROM NewProducts
END
```

When you click Execute, SQL Server creates the stored procedure that you were missing when you tried to run it at the beginning of the demo.

16. Now you need to populate the original products table with the data within the secondary table as if you were adding new rows. Highlight the code below the comment **Having created it, we can run it to feed the missing rows into the Products table**:

```
INSERT INTO Production.Products (productid, productname, supplierid, categoryid, unitprice)
EXEC Production.AddNewProducts;
SELECT * FROM Production.Products
WHERE productid >= 70
```

17. Click **Execute** to transfer the rows and see that they have been transferred.
18. For the other table, you will use the SELECT INSERT statement. Highlight the code below the comment **-- The OrderDetails will be put back using INSERT .. SELECT**, and click **Execute**:

```
INSERT Sales.OrderDetails (orderid, productid, unitprice, qty, discount)
OUTPUT INSERTED.*
SELECT * FROM NewOrderDetails
```

19. Having seen various ways to add data to a new or existing table, you can clean up the database by dropping the objects used in this demo. Highlight the rest of the code below **-- Clean up the database** and click **Execute**:

```
DROP TABLE NewProducts
GO
DROP TABLE NewOrderDetails
GO
DROP PROCEDURE Production.AddNewProducts
```

20. Close SQL Server Management Studio, without saving any changes.

Lesson 2

Modifying and Removing Data

Contents:

Question and Answers	7
Demonstration: Manipulating Data Using the UPDATE and DELETE Statements and MERGING Data Using Conditional DML	7

Question and Answers

Question: A user cannot delete records in the Cars table by using a DELETE statement. The query was intended to remove all pool cars that have been sold. The query used was:

```
DELETE
```

```
FROM Scheduling.Cars
```

```
WHERE Cars.DateSold <> NULL
```

What mistake did the user make?

Answer: An expression comparison operator ' \neq ' was used instead of the IS NOT NULL clause.

Demonstration: Manipulating Data Using the UPDATE and DELETE Statements and MERGING Data Using Conditional DML

Demonstration Steps

Update and Delete Data in a Table

1. Start the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod07\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. In the Command Prompt window press **y**, and then press Enter.
5. When the script has finished, press Enter.
6. Start **SQL Server Management Studio**, and connect to the **MIA-SQL** database engine instance using Windows authentication.
7. On the **File** menu, point to **Open**, and then click **Project/Solution**.
8. In the **Open Project** dialog box, navigate to the **D:\Demofiles\Mod07\Demo** folder, click **Demo.ssmssln**, and then click **Open**.
9. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
10. Highlight the code **USE AdventureWorks GO**, and click **Execute**.
11. Select the code under **USE AdventureWorks GO**, and then click **Execute**.
12. Select the code under the comment **Remove the copied rows from the store table**, and then click **Execute**.
13. Select the code under the comment **Show that they have been removed**, and then click **Execute**.
14. Select the code under the comment **Use the Merge statement to put them back**, and then click **Execute**.
15. Select the code under the comment **SELECT * FROM Sales.Store where 1 = 0 -- used to extract column names for all columns, without cost of data access**, and then click **Execute**.
16. Select the code under the comment **Use the Merge statement to Change the names back**, and then click **Execute**.
17. Select the code under the comment **Ensure that the environment has been restored to the state it was in before the changes were made**, and then click **Execute**.
18. Select the code under the comment **Clean up the database**, and then click **Execute**.

19. Close SQL Server Management Studio without saving any files.

Lesson 3

Generating Automatic Column Values

Contents:

Question and Answers

10

Question and Answers

Using IDENTITY

Question: You are using an IDENTITY column to store the sequence in which orders were placed in a given year. It is a new year and you want to start the count again from 1. Which of the following statements should you use?

- ☐ OrderSequence int IDENTITY(1,1) NOT NULL
- ☐ SET IDENTITY INSERT
- ☐ SCOPE_IDENTITY()
- ☐ DBCC CHECKIDENT
- ☐ CREATE SEQUENCE

Answer:

- ☒ OrderSequence int IDENTITY(1,1) NOT NULL
- ☐ SET IDENTITY INSERT
- ☐ SCOPE_IDENTITY()
- ☐ DBCC CHECKIDENT
- ☐ CREATE SEQUENCE

Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
You are partway through the exercises and want to start again from the beginning. You run the setup script within the solution and receive lots of error messages. This might occur if you have tried to execute the setup script without running the cleanup script to remove any changes you might have made during the lab.	Run the cleanup script before running the setup script.

Lab Review Questions and Answers

Lab: Using DML to Modify Data

Question and Answers

Lab Review

Question: What attributes of the source columns are transferred to a table created with a SELECT INTO query?

Answer: Name, data type, and whether the column is NULL enabled.

Question: The presence of which constraint prevents TRUNCATE TABLE from executing successfully?

Answer: A foreign key reference to the table.

Module 8

Using Built-In Functions

Contents:

Lesson 1: Writing Queries with Built-In Functions	2
Lesson 2: Using Conversion Functions	5
Lesson 3: Using Logical Functions	7
Lesson 4: Using Functions to Work with NULL	9
Module Review and Takeaways	11

Lesson 1

Writing Queries with Built-In Functions

Contents:

Question and Answers	3
Demonstration: Writing Queries Using Built-in Functions	4

Question and Answers

Categorize each item into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	GETDATE()
2	SUM()
3	OPENDATASOURCE()
4	DATEADD()
5	MIN()
6	OPENQUERY()
7	UPPER()
8	MAX()
9	OPENROWSET()
10	YEAR()
11	COUNT()
12	OPENXML()
13	ABS()
14	AVG()
15	DB_NAME()

Category 1		Category 2		Category 3
Scalar Functions		Aggregate Functions		Rowset Functions

Answer:

Category 1		Category 2		Category 3
Scalar Functions		Aggregate Functions		Rowset Functions
GETDATE() DATEADD() UPPER() YEAR() ABS() DB_NAME()		SUM() MIN() MAX() COUNT() AVG()		OPENDATASOURCE() OPENQUERY() OPENROWSET() OPENXML()

Demonstration: Writing Queries Using Built-in Functions

Demonstration Steps

Use Built-in Scalar Functions

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod08\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, and then press Enter.
5. When the script has finished, press Enter.
6. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
7. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod08\Demo** folder.
8. In Solution Explorer, expand **Queries**, and then double-click **11 - Demonstration A.sql**.
9. Select the code under the comment **Step 1**, and then click **Execute**.
10. Select the code under the comment **Step 2**, and then click **Execute**.
11. Select the code under the comment **Step 3**, and then click **Execute**.
12. Select the code under the comment **Step 4**, and then click **Execute**.
13. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Using Conversion Functions**Contents:**

Question and Answers	6
Demonstration: Using Conversion Functions	6

Question and Answers

Question: You are writing a query against a Human Resources database. You want to ensure that the Employee.StartDate values are displayed in standard British form. What function should you use?

Answer: PARSE() or TRY_PARSE.

Demonstration: Using Conversion Functions

Demonstration Steps

Use Functions to Convert Data

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**. Note the error message.
5. Select the code under the comment **Step 4a**, and then click **Execute**.
6. Select the code under the comment **Step 4b**, and then click **Execute**. Note the error message.
7. Select the code under the comment **Step 5**, and then click **Execute**.
8. Select the code under the comment **Step 6**, and then click **Execute**.
9. Select the code under the comment **Step 7**, and then click **Execute**.
10. Select the code under the comment **Step 8a**, and then click **Execute**. Note the error message.
11. Select the code under the comment **Step 8b**, and then click **Execute**.
12. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Using Logical Functions**Contents:**

Question and Answers	8
Demonstration: Using Logical Functions	8

Question and Answers

Question: You have the following query:

```
SELECT e.FirstName, e.LastName, e.FirstAider  
FROM Employees AS e
```

The FirstAider column contains ones and zeros. How can you change the query to make the results more readable?

Answer: Use an IIF function:

```
SELECT e.FirstName, e.LastName,  
       IIF(e.FirstAider = 1, 'Can administer First Aid', 'No First Aid Skills')  
FROM Employees AS e
```

This will output descriptive text instead of ones and zeros.

Demonstration: Using Logical Functions

Demonstration Steps

Using Logical Functions

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Using Functions to Work with NULL**Contents:**

Question and Answers	10
Demonstration: Using Functions to Work with NULL	10

Question and Answers

Question: You are writing a query against the Employees table in the Human Resources database. The CurrentStatus column can contain the string values "New", "Retired", and "Under Caution". Many employees have this column set to NULL when those statuses do not apply to them. For confidentiality, you want to ensure that the employees currently under caution are displayed like those employees with no applicable status. What function should you use?

- () ISNULL()
- () COALESCE()
- () NULLIF()
- () TRY_PARSE()
- () PARSE()

Answer:

- () ISNULL()
- () COALESCE()
- (√) NULLIF()
- () TRY_PARSE()
- () PARSE()

Demonstration: Using Functions to Work with NULL

Demonstration Steps

Use Functions to Work with NULL

1. In Solution Explorer, open the **41 - Demonstration D.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4a**, and then click **Execute**.
6. Select the code under the comment **Step 4b**, and then click **Execute**.
7. Select the code under the comment **Step 4c**, and then click **Execute**.
8. Select the code under the comment **Step 4d**, and then click **Execute**.
9. Select the code under the comment **Step 5**, and then click **Execute**.
10. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Best Practice

When possible, use standards-based functions, such as CAST or COALESCE, rather than SQL Server-specific functions like NULLIF or CONVERT.

Consider the impact of functions in a WHERE clause on query performance.

Review Question(s)

Question: Which function should you use to convert from an int to a nchar(8)?

Answer: CAST()

Question: Which function will return a NULL, rather than an error message, if it cannot convert a string to a date?

Answer: TRY_CONVERT().

Question: What is the name for a function that returns a single value?

Answer: A scalar function.

Module 9

Grouping and Aggregating Data

Contents:

Lesson 1: Using Aggregate Functions	2
Lesson 2: Using the GROUP BY Clause	5
Lesson 3: Filtering Groups with HAVING	7
Module Review and Takeaways	9

Lesson 1

Using Aggregate Functions

Contents:

Question and Answers	3
Demonstration: Using Aggregate Functions	3

Question and Answers

Question: You have the following query:

```
SELECT COUNT(*) AS RecordCount
```

```
FROM Sales.Products;
```

There are 250 records in the Products table. How many rows will be returned by this query?

Answer: One.

Demonstration: Using Aggregate Functions

Demonstration Steps

Use Built-in Aggregate Functions

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod09\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. Start SQL Server Management Studio and connect to the **MIA-SQL** database instance using Windows authentication.
5. On the **File** menu, point to **Open**, and then click **Project/Solution**.
6. In the **Open Project** dialog box, navigate to the **D:\Demofiles\Mod09\Demo** folder, click **Demo.ssmssln**, and then click **Open**.
7. In Solution Explorer, expand **Queries**, and then double-click **11 - Demonstration A.sql**.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2a**, and then click **Execute**.
10. Select the code under the comment **Step 2b**, and then click **Execute**.
11. Select the code under the comment **Step 2c**, and then click **Execute**.
12. Select the code under the comment **Step 2d**, and then click **Execute**.
13. Select the code under the comment **Step 2e**, and then click **Execute**.
14. Select the code under the comment **Step 2f**, and then click **Execute**.
15. Select the code under the comment **Step 2g**, and then click **Execute**.
16. Select the code under the comment **Step 3a**, and then click **Execute**.
17. Select the code under the comment **Step 3b**, and then click **Execute**.
18. Select the code under the comment **Step 3c**, and then click **Execute**.
19. Select the code under the comment **Step 3d**, and then click **Execute**.
20. Select the code under the comment **Step 3e**, and then click **Execute**.
21. Select the code under the comment **Step 3f**, and then click **Execute**.
22. Select the code under the comment **Step 3g**, and then click **Execute**.
23. Select the code under the comment **Step 3h**, and then click **Execute**.
24. Select the code under the comment **Step 3i**, and then click **Execute**.

25. Select the code under the comment **Step 4**, and then click **Execute**.
26. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Using the GROUP BY Clause

Contents:

Question and Answers	6
Demonstration: Using GROUP BY	6

Question and Answers

Question: You are writing the following T-SQL query to find out how many employees work in each department in your organization:

```
SELECT d.DepartmentID, d.DepartmentName, COUNT(e.EmployeeID) AS EmployeeCount
FROM HumanResources.Departments AS d
INNER JOIN HumanResources.Employees AS e
ON d.DepartmentID = e.DepartmentID
GROUP BY
```

Which columns should be included in the GROUP BY clause?

- ☐ All Columns
- ☐ EmployeeCount
- ☐ DepartmentID, DepartmentName
- ☐ DepartmentID

Answer:

- ☐ All Columns
- ☐ EmployeeCount
- ☒ DepartmentID, DepartmentName
- ☐ DepartmentID

Demonstration: Using GROUP BY

Demonstration Steps

Use the GROUP BY Clause

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2a**, and then click **Execute**.
4. Select the code under the comment **Step 2b**, and then click **Execute**.
5. Select the code under the comment **Step 3**, and then click **Execute**.
6. Select the code under the comment **Step 4a**, and then click **Execute**.
7. Select the code under the comment **Step 4b**, and then click **Execute**.
8. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Filtering Groups with HAVING**Contents:**

Question and Answers	8
Demonstration: Filtering Groups with HAVING	8

Question and Answers

Question: You are writing a query to count the number of orders placed for each product. You have the following query:

```
SELECT p.ProductName, COUNT(*) AS OrderCount
FROM Sales.Products AS p
JOIN Sales.OrderItems AS o
ON p.ProductID = o.ProductID
GROUP BY p.ProductName;
```

You want to change the query to return only products that cost more than \$10. Should you add a HAVING clause or a WHERE clause?

Answer: Add a WHERE clause such as WHERE p.Price > 10.

Demonstration: Filtering Groups with HAVING

Demonstration Steps

Filter Grouped Data Using the HAVING Clause

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2a**, and then click **Execute**.
4. Select the code under the comment **Step 2b**, and then click **Execute**.
5. Select the code under the comment **Step 2c**, and then click **Execute**. Note the error message.
6. Select the code under the comment **Step 2d**, and then click **Execute**.
7. Select the code under the comment **Step 2e**, and then click **Execute**.
8. Select the code under the comment **Step 2f**, and then click **Execute**.
9. Select the code under the comment **Step 2g**, and then click **Execute**.
10. Select the code under the comment **Step 2h**, and then click **Execute**.
11. Select the code under the comment **Step 2i**, and then click **Execute**.
12. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: What is the difference between the COUNT function and the COUNT_BIG function?

Answer: COUNT returns an int; COUNT_BIG returns a big_int.

Question: Can a GROUP BY clause include more than one column?

Answer: Yes, separated by commas.

Question: In a query, can a WHERE clause and a HAVING clause filter on the same column?

Answer: Yes.

Module 10

Using Subqueries

Contents:

Lesson 1: Writing Self-Contained Subqueries	2
Lesson 2: Writing Correlated Subqueries	4
Lesson 3: Using the EXISTS Predicate with Subqueries	6
Module Review and Takeaways	8

Lesson 1

Writing Self-Contained Subqueries

Contents:

Question and Answers	3
Demonstration: Writing Self-Contained Subqueries	3

Question and Answers

Question: You are troubleshooting a query. The outer query contains an inner query in its WHERE clause. The first inner query also contains a second inner query in its WHERE clause. Both inner queries are self-contained. The complete query returns an error. How should you approach this task?

Answer: Break the complete query into its constituent subqueries. Start with the innermost subquery and test to see if it returns the information you expect. Next, test the first inner query. Finally, test the complete query.

Demonstration: Writing Self-Contained Subqueries

Demonstration Steps

Write a Nested Subquery

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod10** 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. At the command prompt, press any key.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows® authentication.
6. On the **File** menu, point to **Open**, and then click **File**.
7. In the **Open File** dialog box, navigate to the **D:\Demofiles\Mod10\Demo** folder, and then double-click **Demo.ssmssl.n**.
8. In Solution Explorer, expand **Queries**, and then double-click **11 - Demonstration A.sql**.
9. Select the code under the comment **Step 1**, and then click **Execute**.
10. Select the code under the comment **Step 2**, and then click **Execute**.
11. Select the code under the comment **Step 3**, and then click **Execute**.
12. Select the code under the comment **Step 4**, and then click **Execute**. Note the error message.
13. Select the code under the comment **Step 5**, and then click **Execute**.
14. Select the code under the comment **Step 6**, and then click **Execute**.
15. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Writing Correlated Subqueries

Contents:

Question and Answers	5
Demonstration: Writing Correlated Subqueries	5

Question and Answers

Question: Which of the following statements about correlated subqueries is correct?

- ☐ () To troubleshoot a correlated subquery, execute the inner query first on its own, before placing it into the outer query.
- ☐ () In a correlated subquery, the inner query is run only once, regardless of the number of rows the outer query returns.
- ☐ () In a correlated subquery, the inner query uses data returned by the outer query.
- ☐ () In a correlated subquery, the inner query is executed first, the outer query second.

Answer:

- ☐ () To troubleshoot a correlated subquery, execute the inner query first on its own, before placing it into the outer query.
- ☐ () In a correlated subquery, the inner query is run only once, regardless of the number of rows the outer query returns.
- ☒ (√) In a correlated subquery, the inner query uses data returned by the outer query.
- ☐ () In a correlated subquery, the inner query is executed first, the outer query second.

Working with Correlated Subqueries

Question: Why can't a correlated subquery be executed separately from the outer query?

Answer: The subquery depends on input from the outer query for its values.

Demonstration: Writing Correlated Subqueries

Demonstration Steps

Write a Correlated Subquery

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Using the EXISTS Predicate with Subqueries

Contents:

Question and Answers	7
Demonstration: Writing Subqueries Using EXISTS	7

Question and Answers

Question: The Human Resources database has recently been extended to record the skills possessed by employees. Employees have added their skills to the database by using a web-based user interface. You want to find employees who have not yet added their skills. You have the following query:

```
SELECT e.EmployeeID, e.FirstName
FROM HumanResources.Employees AS e
WHERE NOT EXISTS (
    SELECT s.EmployeeID, s.SkillName, s.SkillCategory
    FROM HumanResources.Skills AS s
    WHERE e.EmployeeID = s.EmployeeID);
```

How can you improve the query?

Answer: In the inner query SELECT clause, replace the list of columns with `“*”`.

Demonstration: Writing Subqueries Using EXISTS

Demonstration Steps

Write Queries Using EXISTS and NOT EXISTS

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4a**, and then click **Execute**.
6. Select the code under the comment **Step 4b**, and then click **Execute**.
7. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: Can a correlated subquery return a multi-valued set?

Answer: Yes.

Question: What type of subquery may be rewritten as a JOIN?

Answer: Correlated subqueries.

Question: Which columns should appear in the SELECT list of a subquery following the EXISTS predicate?

Answer: Only a * needs to be specified. No actual columns will be retrieved.

Module 11

Using Table Expressions

Contents:

Lesson 1: Using Views	2
Lesson 2: Using Inline TVFs	4
Lesson 3: Using Derived Tables	6
Lesson 4: Using CTEs	8
Module Review and Takeaways	10

Lesson 1

Using Views

Contents:

Question and Answers	3
Resources	3
Demonstration: Using Views	3

Question and Answers

Question: Your DBAs want to grant access to Sales users on the Customers table in the Sales database. However, they also need to prevent Sales users from reading values in the Customers.Relationship column. How can they set up this access?

Answer: Create a view that queries the Customers table but does not include the Relationship column. Grant access to the view and not to the table.

Resources

Writing Queries That Return Results from Views



Additional Reading: For more information on database security, go to course 20764B: *Administering a SQL Database Infrastructure*.

Demonstration: Using Views

Demonstration Steps

Create Views

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod11\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, press Enter, and then wait for the script to finish.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod11\Demo** folder.
7. Open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Using Inline TVFs

Contents:

Question and Answers	5
Demonstration: Inline TVFs	5

Question and Answers

Question: From the following statements, select the one that is true of TVFs but not true of Views.

- ☐ () Stored persistently in the database.
- ☐ () Can accept input parameters.
- ☐ () Can be referred to in a FROM clause, like a table.
- ☐ () Does not store data in the database but queries the database whenever it is called.

Answer:

- ☐ () Stored persistently in the database.
- ☒ (√) Can accept input parameters.
- ☐ () Can be referred to in a FROM clause, like a table.
- ☐ () Does not store data in the database but queries the database whenever it is called.

Demonstration: Inline TVFs

Demonstration Steps

Create Inline TVFs

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Using Derived Tables

Contents:

Question and Answers	7
Demonstration: Using Derived Tables	7

Question and Answers

Question: You are troubleshooting the following query, which returns an error:

```
SELECT orderyear, COUNT(DISTINCT custid) AS cust_count
FROM (
    SELECT YEAR(orderdate) AS orderyear, custid
    FROM Sales.Orders
    WHERE empid = 354
    ORDER BY YEAR(orderdate)
) AS derived_year
GROUP BY orderyear;
```

How can you resolve the error?

Answer: Remove the ORDER BY clause from the derive table query.

Demonstration: Using Derived Tables

Demonstration Steps

Write Queries that Create Derived Tables

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Using CTEs

Contents:

Question and Answers	9
Demonstration: Using CTEs	9

Question and Answers

Question: Which of the following features is required for a CTE query?

- ☐ The query must have a WITH ... AS clause.
- ☐ The query must include a GROUP BY clause.
- ☐ The query must include a CREATE FUNCTION statement.
- ☐ The query must include a nested derived query.

Answer:

- ☒ The query must have a WITH ... AS clause.
- ☐ The query must include a GROUP BY clause.
- ☐ The query must include a CREATE FUNCTION statement.
- ☐ The query must include a nested derived query.

Demonstration: Using CTEs

Demonstration Steps

Write Queries that Create CTEs

1. In Solution Explorer, open the **41 - Demonstration D.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: When would you use a CTE rather than a derived table for a query?

Answer: CTEs may be written once, referenced multiple times in a query.

Question: Which table expressions allow variables to be passed in as parameters to the expression?

Answer: Table-valued functions.

Module 12

Using Set Operators

Contents:

Lesson 1: Writing Queries with the UNION Operator	2
Lesson 2: Using EXCEPT and INTERSECT	4
Lesson 3: Using APPLY	6
Module Review and Takeaways	8

Lesson 1

Writing Queries with the UNION Operator

Contents:

Question and Answers	3
Demonstration: Using UNION and UNION ALL	3

Question and Answers

Question: True or false? The results from a UNION query can contain duplicate rows.

☐ True

☐ False

Answer:

☐ True

☒ False

Question: When combining the output of two sets, UNION and UNION ALL queries cannot include rows with NULL values, because NULL values cannot be compared.

☐ True

☐ False

Answer:

☐ True

☒ False

Demonstration: Using UNION and UNION ALL

Demonstration Steps

1. Ensure that the MSL-TMG1, 20761B-MIA-DC, and 20761B-MIA-SQL virtual machines are running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod12\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, and then press Enter.
5. Wait for the script to finish, and then press any key.
6. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
7. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod12\Demo** folder.
8. In Solution Explorer, expand **Queries**, and double-click the **11 - Demonstration A.sql** script file.
9. Select the code under the comment **Step 1**, and then click **Execute**.
10. Select the code under the comment **Step 2**, and then click **Execute**.
11. Select the code under the comment **Step 3**, and then click **Execute**.
12. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Using EXCEPT and INTERSECT

Contents:

Question and Answers	5
Demonstration: Using EXCEPT and INTERSECT	5

Question and Answers

Question: You have a table of employees and a table of customers, both of which contain a column holding the name of the country where the customer or employee is located. You want to know which countries have at least one customer and at least one employee. Which set operator should you use?

- ☐ UNION ALL
- ☐ UNION
- ☐ EXCEPT
- ☐ INTERSECT
- ☐ None of the above

Answer:

- ☐ UNION ALL
- ☐ UNION
- ☐ EXCEPT
- ☒ INTERSECT
- ☐ None of the above

Demonstration: Using EXCEPT and INTERSECT

Demonstration Steps

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Using APPLY

Contents:

Question and Answers	7
Demonstration: Using CROSS APPLY and OUTER APPLY	7

Question and Answers

Question: What is the difference between CROSS APPLY and CROSS JOIN?

Answer: CROSS JOIN returns all the possible combinations of the left and right table sources; CROSS APPLY returns only the values from the left table source where a value is found in the right table source.

Demonstration: Using CROSS APPLY and OUTER APPLY

Demonstration Steps

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Test with CROSS APPLY**, and then click **Execute**.
6. Select the code under the comment **Step 4**, and then click **Execute**.
7. Select the code under the comment **Step 5**, and then click **Execute**.
8. Select the code under the comment **Use OUTER APPLY to include customers with no orders**, and then click **Execute**.
9. Close SQL Server Management Studio, without saving any changes.

Module Review and Takeaways

Question: Which set operator would you use to combine sets if you knew there were no duplicates and wanted the best possible performance?

Answer: UNION ALL

Question: Which form of the APPLY operator will not return rows from the left table if the result of the right table expression is empty?

Answer: CROSS APPLY

Question: Which form of the APPLY operator can be used to rewrite LEFT OUTER JOIN queries?

Answer: OUTER APPLY

Module 13

Using Window Ranking, Offset, and Aggregate Functions

Contents:

Lesson 1: Creating Windows with OVER	2
Lesson 2: Exploring Window Functions	4
Module Review and Takeaways	7

Lesson 1

Creating Windows with OVER

Contents:

Question and Answers	3
Demonstration: Using OVER and Partitioning	3

Question and Answers

Put the following elements into the logical order in which they are processed in a windowed query by numbering each to indicate the correct order.

	Steps
	The OVER clause
	The PARTITION BY clause
	The ROW or RANGE clause

Answer:

	Steps
1	The OVER clause
2	The PARTITION BY clause
3	The ROW or RANGE clause

Demonstration: Using OVER and Partitioning

Demonstration Steps

Use OVER, PARTITION BY, and ORDER BY Clauses

1. Ensure that the 20761D-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod13\Setup.cmd** as an administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **y**, and then press Enter.
5. Wait for the script to finish, and then press any key.
6. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
7. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod13\Demo** folder.
8. In Solution Explorer, open the **11 - Demonstration A.sql** script file.
9. Select the code under the comment **Step 1**, and then click **Execute**.
10. Select the code under the comment **Step 2**, and then click **Execute**.
11. Select the code under the comment **Step 3**, and then click **Execute**.
12. Select the code under the comment **Step 4**, and then click **Execute**.
13. Select the code under the comment **Step 5**, and then click **Execute**.
14. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Exploring Window Functions

Contents:

Question and Answers	5
Demonstration: Exploring Windows Functions	6

Question and Answers

Place each windowing function into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	SUM()
2	RANK()
3	PERCENT_RANK()
4	MIN()
5	DENSERANK()
6	CUME_DIST()
7	MAX()
8	ROW_NUMBER()
9	PERCENTILE_CONT()
10	NTITLE()
11	PERCENTILE_DISC()

Category 1		Category 2		Category 3
Window Aggregate Functions		Window Ranking Function		Window Distribution Functions

Answer:

Category 1		Category 2		Category 3
Window Aggregate Functions		Window Ranking Function		Window Distribution Functions
SUM() MIN() MAX()		RANK() DENSE_RANK() ROW_NUMBER() NTILE()		PERCENT_RANK() CUME_DIST() PERCENTILE_CONT() PERCENTILE_DISC()

Demonstration: Exploring Windows Functions

Demonstration Steps

Use Window Aggregate, Ranking, and Offset Functions

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Select the code under the comment **Step 8**, and then click **Execute**.
10. Select the code under the comment **Step 9**, and then click **Execute**.
11. Select the code under the comment **Step 10**, and then click **Execute**.
12. Select the code under the comment **Step 11**, and then click **Execute**.
13. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: What results will be returned by a ROW_NUMBER function if there is no ORDER BY clause in the query?

Answer: An unordered set.

Question: Which ranking function would you use to return the values 1,1,3? Which would return 1,1,2?

Answer: RANK, DENSE_RANK.

Question: Can a window frame extend beyond the boundaries of the window partition defined in the same OVER() clause?

Answer: No.

Module 14

Pivoting and Grouping Sets

Contents:

Lesson 1: Writing Queries with PIVOT and UNPIVOT	2
Lesson 2: Working with Grouping Sets	4
Module Review and Takeaways	6

Lesson 1

Writing Queries with PIVOT and UNPIVOT

Contents:

Question and Answers	3
Demonstration: Writing Queries with PIVOT and UNPIVOT	3

Question and Answers

Question: You have the following query:

```
SELECT category, qty, orderyear
FROM Sales.PivotedCategorySales
UNPIVOT(qty FOR orderyear) AS unpvt;
```

In this query, you have provided a name for the new column that will display the unpivoted values ("qty"). You have also provided a name for the column that will display the names of the unpivoted values (orderyear). What else must you provide for the UNPIVOT query to execute?

Answer: You must supply the names of the pivoted columns that will be unpivoted by the query.

Demonstration: Writing Queries with PIVOT and UNPIVOT

Demonstration Steps

Use PIVOT and UNPIVOT

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod14\Setup.cmd** as an administrator.
3. At the command prompt, type **y**, and then press Enter.
4. When the script completes, close the command prompt window.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod14\Demo** folder.
7. In Solution Explorer, open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Select the code under the comment **Step 5**, and then click **Execute**.
13. Select the code under the comment **Step 6**, and then click **Execute**.
14. Select the code under the comment **Step 7**, and then click **Execute**.
15. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Working with Grouping Sets

Contents:

Question and Answers	5
Demonstration: Using Grouping Sets	5

Question and Answers

Question: You have the following query:

```
SELECT e.Department, e.Country, COUNT(EmployeeID) AS Staff
FROM HumanResources.Employees AS e
```

You want to find out how many staff are in each country and how many staff are in each department. You also want to find out how many staff are in Sales in the US, and so on, with all departments in all countries where the company operates. Choose the most succinct grouping technique for this query:

- ☐ GROUPING SETS
- ☐ CUBE
- ☐ ROLLUP
- ☐ You cannot return the required data with GROUPING. Instead, use multiple queries and a UNION element.

Answer:

- ☐ GROUPING SETS
- ☒ CUBE
- ☐ ROLLUP
- ☐ You cannot return the required data with GROUPING. Instead, use multiple queries and a UNION element.

Demonstration: Using Grouping Sets

Demonstration Steps

Use the CUBE and ROLLUP Subclauses

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Select the code under the comment **Step 8**, and then click **Execute**.
10. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: Once a dataset has been pivoted with aggregation, can the original detail rows be restored with an unpivot operation?

Answer: No, the original detail is lost during aggregation.

Question: What are the possible sources of NULLs returned by a query using grouping sets to create aggregations?

Answer: NULLs might be present in the underlying source data, or may be placeholders for rows that do not participate in the group member.

Question: Which subclause infers a hierarchy of columns to create meaningful grouping sets?

Answer: ROLLUP.

Module 15

Executing Stored Procedures

Contents:

Lesson 1: Querying Data with Stored Procedures	2
Lesson 2: Passing Parameters to Stored Procedures	4
Lesson 3: Creating Simple Stored Procedures	7
Lesson 4: Working with Dynamic SQL	9
Module Review and Takeaways	11

Lesson 1

Querying Data with Stored Procedures

Contents:

Question and Answers	3
Demonstration: Querying Data with Stored Procedures	3

Question and Answers

Question: You have the following query, which is intended to call a stored procedure called `HumanResources.FilteredSkills`:

```
EXEC HumanResources.FilteredSkills
    departmentid = @1, skilllevel = @4;
```

Your query returns an error. What should you do to resolve the error?

Answer: You must use the correct syntax to pass the two parameters.

Demonstration: Querying Data with Stored Procedures

Demonstration Steps

Use Stored Procedures

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod15\Setup.cmd** as an administrator.
3. At the command prompt, type **y**, and press Enter.
4. Wait until the script completes, and then press Enter.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod15\Demo** folder.
7. Open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Select the code under the comment **Step 5**, and then click **Execute**.
13. Select the code under the comment **Step 6**, and then click **Execute**.
14. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Passing Parameters to Stored Procedures

Contents:

Question and Answers	5
Demonstration: Passing Parameters to Stored Procedures	5

Question and Answers

Question: A DBA has created a stored procedure in the HumanResources database by executing the following:

```
CREATE PROCEDURE HumanResources.SkillsForEmployee (@empid AS INT)
AS
    SELECT e.ID, e.FirstName, e.LastName, s.SkillName, s.Level
    FROM HumanResources.Employees AS e
    JOIN HumanResources.Skills AS s ON e.ID = s.EmployeeID
    WHERE e.ID = @empid
GO
```

You call the procedure with the following statement:

```
EXEC HumanResources.SkillsForEmployee @empid = N'24'
```

Your query returns an error. What should you do to fix your query?

- ☐ () Pass the @empid parameter as an integer instead of an nvarchar.
- ☐ () Move the position of the "@" symbol to the correct place.
- ☐ () Add a value for the e.ID parameter to your query.
- ☐ () Instead of using the stored procedure, execute your own SELECT query.
- ☐ () Add the OUTPUT keyword to the @empid parameter.

Answer:

- ☒ (v) Pass the @empid parameter as an integer instead of an nvarchar.
- ☐ () Move the position of the "@" symbol to the correct place.
- ☐ () Add a value for the e.ID parameter to your query.
- ☐ () Instead of using the stored procedure, execute your own SELECT query.
- ☐ () Add the OUTPUT keyword to the @empid parameter.

Demonstration: Passing Parameters to Stored Procedures

Demonstration Steps

Pass Parameters to a Stored Procedure

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. In Object Explorer, expand **Databases**, expand **AdventureWorks**, expand **Programmability**, and then expand **Stored Procedures**.
4. Expand any procedure, expand **Parameters**, and then point out list of parameters, data type and direction.
5. Select the code under the comment **Step 3**, and then click **Execute**.
6. Select the code under the comment **Step 4**, and then click **Execute**.
7. Select the code under the comment **Step 5**, and then click **Execute**.

8. Select the code under the comment **Step 6**, and then click **Execute**.
9. Select the code under the comment **Step 7**, and then click **Execute**.
10. Select the code under the comment **Step 8**, and then click **Execute**.
11. Select the code under the comment **Step 9**, and then click **Execute**.
12. Select the code under the comment **Step 10**, and then click **Execute**.
13. Select the code under the comment **Step 11**, and then click **Execute**.
14. Keep SQL Server Management Studio open for the next demonstration.

Lesson 3

Creating Simple Stored Procedures

Contents:

Question and Answers	8
Demonstration: Creating Simple Stored Procedures	8

Question and Answers

Question: The HumanResources.SkillLevelsForDepartment stored procedure is a popular procedure that ensures skills data can be examined in an anonymous form. You have been asked to add a new parameter to the stored procedure. Why should you use ALTER PROCEDURE instead of DROP PROCEDURE followed by CREATE PROCEDURE.

Answer: Because permissions assigned to the stored procedure will be maintained if you use ALTER PROCEDURE.

Demonstration: Creating Simple Stored Procedures

Demonstration Steps

Create a Stored Procedure

1. In Solution Explorer, open the **31 - Demonstration C.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Keep SQL Server Management Studio open for the next demonstration.

Lesson 4

Working with Dynamic SQL

Contents:

Question and Answers	10
Demonstration: Working with Dynamic SQL	10

Question and Answers

Question: You want to execute dynamic SQL with a single parameter named @skillname. In addition to the parameter itself, what other parameters should you send to sp_executesql?

Answer: @statement and @params.

Demonstration: Working with Dynamic SQL

Demonstration Steps

Execute Dynamic SQL Queries

1. In Solution Explorer, open the **41 - Demonstration D.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: What benefits do stored procedures provide for data retrieval that views do not?

Answer: Answers may vary, but ability to accept parameters is a correct answer.

Question: What form should parameter and value pairs take when passed to a stored procedure in the EXECUTE statement?

Answer: @NAME = VALUE.

Question: Which method for constructing dynamic SQL allows parameters to be passed at runtime?

Answer: Using sp_executesql.

Module 16

Programming with T-SQL

Contents:

Lesson 1: T-SQL Programming Elements	2
Lesson 2: Controlling Program Flow	5
Module Review and Takeaways	7

Lesson 1

T-SQL Programming Elements

Contents:

Question and Answers	3
Demonstration: T-SQL Programming Elements	3

Question and Answers

Question: You have the following T-SQL script:

```
INSERT INTO HumanResources.PossibleSkills (SkillName, Category, Credit)
VALUES('Database Administration', 'IT Professional', 5);

INSERT INTO HumanResources.PossibleSkills (SkillName, Category, Credit)
VALUES('C#.NET', 'Developer', 4);

INSERT INTO HumanResources.PossibleSkills (SkillName, Category, Credit)
VALUES('Project Management', 'Management', 'Two');

GO
```

The script generates an error on the third INSERT statement. How many new rows do you expect to find in the PossibleSkills table after this error?

Answer: Two new rows.

Demonstration: T-SQL Programming Elements

Demonstration Steps

Control Batch Execution and Variable Usage

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod16\Setup.cmd** as an administrator.
3. At the command prompt, type **y**, and then press Enter.
4. When the script completes, close the command prompt window.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod16\Demo** folder.
7. Open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **-- Create a proc to search for category**, and then click **Execute**.
11. Select the code under the comment **-- Set up table for batch demos**, and then click **Execute**.
12. Select the code under the comment **Step 3**, and then click **Execute**.
13. Select the code under the comment **-- Show that the batch was successful**, and then click **Execute**.
14. Select the code under the comment **Step 4**, and then click **Execute**.
15. Select the code under the comment **Step 5**, and then click **Execute**. Note the error message.
16. Select the code under the comment **--Show that no rows were inserted**, and then click **Execute**.
17. Select the code under the comment **Step 6**, and then click **Execute**.

18. Select the code under the comment **--Run the following batch in its entirety to show the choices,** and then click **Execute**.
19. Select the code under the comment **Step 7,** and then click **Execute**.
20. Select the code under the comment **-- Declare a parameter to search for category,** and then click **Execute**.
21. Select the code under the comment **-- Test it locally,** and then click **Execute**.
22. Select the code under the comment **Step 8,** and then click **Execute**.
23. Select the code under the comment **Step 9,** and then click **Execute**.
24. Select the code under the comment **Step 10,** and then click **Execute**.
25. Select the code under the comment **Step 11,** and then click **Execute**.
26. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Controlling Program Flow

Contents:

Question and Answers	6
Demonstration: Controlling Program Flow	6

Question and Answers

Question: You want to populate a table by creating 15 new rows. Before you create the rows, you need to check that the table exists. From the following T-SQL keywords, choose the one that you will NOT need to use.

- ☐ IF
- ☐ WHILE
- ☐ BEGIN
- ☐ END
- ☐ INSERT

Answer:

- ☐ IF
- ☒ WHILE
- ☐ BEGIN
- ☐ END
- ☐ INSERT

Demonstration: Controlling Program Flow

Demonstration Steps

Control the Flow of Execution

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: Can you declare a variable in one batch and reference it in multiple batches?

Answer: No, variables are local to the batch in which they are declared.

Question: Can you create a synonym that references an object that does not yet exist?

Answer: Yes, resolution doesn't occur until the synonym is used.

Question: Will a WHILE loop exit when the predicate evaluates to NULL?

Answer: Yes.

Module 17

Implementing Error Handling

Contents:

Lesson 1: Implementing T-SQL Error Handling	2
Lesson 2: Implementing Structured Exception Handling	4
Module Review and Takeaways	6

Lesson 1

Implementing T-SQL Error Handling

Contents:

Question and Answers	3
Demonstration: Handling Errors Using T-SQL	3

Question and Answers

Question: You are writing some error handling in a T-SQL script. If a problem arises, you want to raise an error with a severity of 20. Should you use RAISERROR or THROW for this error handling?

Answer: Use RAISERROR

Demonstration: Handling Errors Using T-SQL

Demonstration Steps

Handle Errors

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod17\Setup.cmd** as an administrator.
3. At the command prompt, type **y**, and then press Enter.
4. When the script completes, close the command prompt window.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod17\Demo** folder.
7. Open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **--Capture @@ERROR into a variable**, and then click **Execute**.
11. Select the code under the comment **--Create a custom error message**, and then click **Execute**.
12. Select the code under the comment **--Use a custom error message**, and then click **Execute**.
13. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Implementing Structured Exception Handling

Contents:

Question and Answers	5
Demonstration: Using a TRY/CATCH Block	5

Question and Answers

Question: You have the following T-SQL script:

```
BEGIN TRY
    INSERT INTO HumanResources.PossibleSkills(SkillName, Category)
    VALUES ('Database Administration', 'IT Professional');
END TRY
DECLARE @prefix AS NVARCHAR(50) = 'There has been an error: ';
BEGIN CATCH
    PRINT @prefix + ERROR_MESSAGE();
    THROW;
END CATCH;
GO
```

The code will not compile and execute. What should you do to troubleshoot this code?

Answer: Move the line that declares and assigns the @prefix variable.

Demonstration: Using a TRY/CATCH Block

Demonstration Steps

Use a TRY/CATCH Block

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: Which error types cannot be caught by structured exception handling?

Answer: Compile/syntax errors, as well as some delayed name resolution errors.

Question: Can TRY/CATCH blocks be nested?

Answer: Yes.

Question: How can you use THROW outside of a CATCH block?

Answer: With arguments that raise a user-defined error.

Module 18

Implementing Transactions

Contents:

Lesson 1: Transactions and the Database Engine	2
Lesson 2: Controlling Transactions	5
Module Review and Takeaways	7

Lesson 1

Transactions and the Database Engine

Contents:

Question and Answers	3
Demonstration: Transactions and the Database Engine	4

Question and Answers

Place each T-SQL keyword into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	BEGIN TRANSACTION
2	BEGIN TRY
3	COMMIT TRANSACTION
4	END TRY
5	END TRANSACTION
6	BEGIN CATCH
7	ROLLBACK TRANSACTION
8	END CATCH
9	INSERT
10	RAISERROR

Category 1		Category 2
TCL		Non-TCL

Answer:

Category 1		Category 2
TCL		Non-TCL
BEGIN TRANSACTION COMMIT TRANSACTION END TRANSACTION ROLLBACK TRANSACTION		BEGIN TRY END TRY BEGIN CATCH END CATCH INSERT RAISERROR

Demonstration: Transactions and the Database Engine**Demonstration Steps**

Use Transactions

1. Ensure that the 20761B-MIA-DC and 20761B-MIA-SQL virtual machines are both running, and then log on to 20761B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Run **D:\Demofiles\Mod18\Setup.cmd** as an administrator.
3. At the command prompt, type **y**, and then press Enter.
4. When the script completes, close the command prompt window.
5. Start SQL Server Management Studio and connect to the **MIA-SQL** database engine instance using Windows authentication.
6. Open the **Demo.ssmssl** solution in the **D:\Demofiles\Mod18\Demo** folder.
7. Open the **11 - Demonstration A.sql** script file.
8. Select the code under the comment **Step 1**, and then click **Execute**.
9. Select the code under the comment **Step 2**, and then click **Execute**.
10. Select the code under the comment **Step 3**, and then click **Execute**. Note the error message.
11. Select the code under the comment **Step 4**, and then click **Execute**.
12. Select the code under the comment **Step 5**, and then click **Execute**.
13. Keep SQL Server Management Studio open for the next demonstration.

Lesson 2

Controlling Transactions**Contents:**

Question and Answers	6
Demonstration: Controlling Transactions	6

Question and Answers

Question: You have executed the following batch of T-SQL statements:

```
BEGIN TRY
    BEGIN TRANSACTION;
        INSERT INTO dbo.SimpleOrders(custid, empid, orderdate)
            VALUES (68,9,'2006-07-12');
        INSERT INTO dbo.SimpleOrderDetails(orderid,productid,unitprice,qty)
            VALUES (1, 2,15.20,20);
    END TRY
BEGIN CATCH
    SELECT ERROR_NUMBER() AS ErrNum, ERROR_MESSAGE() AS ErrMsg;
    ROLLBACK TRANSACTION;
END CATCH;
```

A fellow query writer is now receiving errors resulting from locks on database records. What can you do to troubleshoot this problem?

Answer: Issue a COMMIT TRANSACTION statement.

Demonstration: Controlling Transactions

Demonstration Steps

Control Transactions

1. In Solution Explorer, open the **21 - Demonstration B.sql** script file.
2. Select the code under the comment **Step 1**, and then click **Execute**.
3. Select the code under the comment **Step 2**, and then click **Execute**.
4. Select the code under the comment **Step 3**, and then click **Execute**.
5. Select the code under the comment **Step 4**, and then click **Execute**.
6. Select the code under the comment **Step 5**, and then click **Execute**.
7. Select the code under the comment **Step 6**, and then click **Execute**. Note the error message.
8. Select the code under the comment **Step 7**, and then click **Execute**.
9. Select the code under the comment **Step 8**, and then click **Execute**.
10. Close SQL Server Management Studio without saving any files.

Module Review and Takeaways

Question: What happens to a nested transaction when the outer transaction is rolled back?

Answer: The inner transaction is also rolled back, so nested transactions are not typically useful in user code.

Question: When a runtime error occurs in a transaction and SET XACT_ABORT is ON, is the transaction always automatically rolled back?

Answer: No, not if the error occurs within a TRY block.

