

OFFICIAL MICROSOFT LEARNING PRODUCT

10985B

Introduction to SQL Databases

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.

© 2016 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: 10985B

Part Number (if applicable):

Released: 10/2016

Module 1

Introduction to Databases

Contents:

Lesson 1: Introduction to Relational Databases	2
Lesson 2: Other Types of Databases and Storage	5
Lesson 3: Data Analysis	7
Lesson 4: Database Languages in SQL Server 2016	9
Module Review and Takeaways	12
Lab Review Questions and Answers	13

Lesson 1

Introduction to Relational Databases

Contents:

Question and Answers	3
Demonstration: Exploring a Relational Database	3

Question and Answers

Question: Which of the following describes the purpose of a foreign key?

- () To ensure that values in a column are unique.
- () To reduce data redundancy.
- () To check that the values added to a column are valid by referencing a different column, usually a column in a different table.

Answer:

- () To ensure that values in a column are unique.
- () To reduce data redundancy.
- (√) To check that the values added to a column are valid by referencing a different column, usually a column in a different table.

Demonstration: Exploring a Relational Database

Demonstration Steps

1. Start the 10985B-MIA-DC and 10985B-MIA-SQL virtual machines, and then log on to 10985B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod01** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for setup to complete.
4. Open Microsoft SQL Server Management Studio, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
5. In Object Explorer, expand **Databases**, expand **AdventureWorks2016**, right-click **Database Diagrams**, and then click **New Database Diagram**.
6. If the **Microsoft SQL Server Management Studio** dialog box appears asking if you wish to create support objects for database diagramming, click **Yes**.
7. In the **Add Table** dialog box, press and hold down the CTRL key, click **Customer (Sales)**, click **SalesOrderDetail (Sales)**, click **SalesOrderHeader (Sales)**, click **ShipMethod (Purchasing)**, click **Add**, and then click **Close**.
8. Review the tables and note the following points:
 - a. The **SalesOrderHeader (Sales)** table contains the **SalesOrderID**, which is the primary key column.
 - b. The **SalesOrderDetail (Sales)** table also contains a **SalesOrderID** column.
9. In the **SalesOrderDetail (Sales)** table, right-click the **SalesOrderID** column, and then click **Properties**.
10. In the **Properties** window, click the **Description** field, and then click the ellipsis button (...).
11. In the **Description Property** dialog box, note that the column is a primary key column, and that there is a foreign key that references the **SalesOrderID** column in the **SalesOrderHeader** column, and then click **Cancel**.
12. Click the line between the **Customer (Sales)** table and the **SalesOrderHeader (Sales)** table. This line represents a foreign key relationship.
13. In the **Properties** window, click **Description**, and then click the ellipsis button (...).

14. In the **Description Property** dialog box, note that the foreign key references the **CustomerID** column in the **Customer (Sales)** table, and then click **Cancel**.
15. Close the database diagram window, and do not save any changes. Close SQL Server Management Studio.

Lesson 2

Other Types of Databases and Storage

Contents:

Question and Answers

6

Question and Answers

Match the types of database to the correct description. Indicate your answer by writing the category number to the right of each item.

Items	
1	Highly structured databases that store data in related tables.
2	A broad category of different types of databases that are often used to store semi-structured and unstructured data, or highly complex data.
3	Analytical databases that contain aggregated data stored in cubes.

Category 1		Category 2		Category 3
Relational Databases		NoSQL Databases		OLAP Databases

Answer:

Category 1		Category 2		Category 3
Relational Databases		NoSQL Databases		OLAP Databases
Highly structured databases that store data in related tables.		A broad category of different types of databases that are often used to store semi-structured and unstructured data, or highly complex data.		Analytical databases that contain aggregated data stored in cubes.

Lesson 3

Data Analysis

Contents:

Question and Answers

8

Question and Answers

Question: Data Quality Services is a tool for analyzing data to gain insights.

☐ True

☐ False

Answer:

☐ True

☒ False

Lesson 4

Database Languages in SQL Server 2016

Contents:

Question and Answers	10
Demonstration: Querying a SQL Server 2016 Database	10

Question and Answers

Question: Which one of the following correctly represents the structure of a SQL Server Transact-SQL SELECT statement?

- () SELECT <table>
FROM <column list>
WHERE <filter that limits the rows returned>
- () SELECT <column list>
FROM <table>
WHERE <filter that limits the rows returned>
- () SELECT <rows>
FROM <table>
WHERE <filter that limits the rows returned>

Answer:

- () SELECT <table>
FROM <column list>
WHERE <filter that limits the rows returned>
- (√) SELECT <column list>
FROM <table>
WHERE <filter that limits the rows returned>
- () SELECT <rows>
FROM <table>
WHERE <filter that limits the rows returned>

Demonstration: Querying a SQL Server 2016 Database

Demonstration Steps

1. Open SQL Server Management Studio, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
2. On the **File** menu, point to **Open**, click **File**, browse to **D:\Demofiles\Mod01**, click **TransactSQLQueries.sql**, and then click **Open**.
3. Select the code under the comment **Return all rows from all columns in the Sales.SalesOrderHeader table**, and then click **Execute**.
4. Review the result set, noting the columns that the query returns and, in the bottom right-hand corner of the results set, the number of rows the query returned.
5. Select the code under the comment **Return all rows from the SalesOrderID and OrderDate columns from the Sales.SalesOrderHeader table**, and then click **Execute**.
6. Review the result set, noting the columns and the number of rows the query returned.
7. Select the code under the comment **Return only rows from the SalesOrderID, OrderDate, and SalesPersonID columns for which the SalespersonID = 282**, and then click **Execute**.
8. Review the result set, noting the columns and the number of rows the query returned. Note that the number of rows returned is much lower than for the previous two queries.
9. Select the code under the comment **Return only rows from the SalesOrderID, OrderDate, and SalesPersonID columns for which the SalespersonID > 282**, and then click **Execute**.

10. Review the result set, noting the columns and the number of rows the query returned. Note that the number of rows returned has increased again.
11. Select the code under the comment **Return only rows from the SalesOrderID and OrderDate columns for which the SalespersonID = 282 and the orderdate is from the year 2013**, and then click **Execute**.
12. Review the result set, noting the columns and the number of rows the query returned. This query returned the lowest number of rows because it contained the most restrictive filters.
13. Close the TransactSQLQueries.sql query window and do not save any changes. Close SQL Server Management Studio.

Module Review and Takeaways

Review Question(s)

Question: What types of databases and RDBMSs are in use in your work place? Do you have any experience of working directly with them?

Answer: Answers will vary because students' experiences are likely to be different.

Lab Review Questions and Answers

Lab: Exploring and Querying SQL Server Databases

Question and Answers

Lab Review

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

Answer: The number of rows varied because the WHERE clause in the queries acts as a filter that limits the rows that the query returns. The WHERE clause in each statement uses different conditions, therefore different numbers of rows are returned.

Module 2

Data Modeling

Contents:

Lesson 1: Data Modeling	2
Lesson 2: ANSI-SPARC Database Model	4
Lesson 3: Entity Relationship Modeling	6
Module Review and Takeaways	8
Lab Review Questions and Answers	9

Lesson 1

Data Modeling

Contents:

Question and Answers

3

Question and Answers

Question: True or false? You can use entity-relationship modeling to show the way the elements in the database are interconnected.

☐ True

☐ False

Answer:

☒ True

☐ False

Lesson 2

ANSI-SPARC Database Model

Contents:

Question and Answers

5

Question and Answers

Question: Where is the physical representation of the data modeled?

- ☐ External level
- ☐ Conceptual level
- ☐ Internal level

Answer:

- ☐ External level
- ☐ Conceptual level
- ☒ Internal level

Lesson 3

Entity Relationship Modeling

Contents:

Question and Answers	7
Resources	7

Question and Answers

Question: True or false? You can remove one or more attributes from a multi-attribute candidate key without detriment.

☐ True

☐ False

Answer:

☐ True

☒ False

Resources

Notations



Additional Reading: *Case*Method TM, Entity-Relationship Modelling* by Richard Barker (Addison Wesley).

Module Review and Takeaways

Best Practice

Always listen to your customers. You will need to extract their requirements because they might not be able to fully describe what they need.

Approach data modeling in a logical manner. Do not try to add detail too early in the process, before you fully understand what is required in general.

Review Question(s)

Question: How would you approach data modeling?

Answer: By understanding the requirements of the business.

To successfully model data so that the result meets the requirements of the business, you first need to understand the requirements. This means talking to the relevant people and extracting the requirements from them. Remember that they may not be totally certain in their own minds about what they need.

This is a fundamental skill of a business analyst.

Real-world Issues and Scenarios

One of the issues you may encounter might involve changes made to the requirements. These are usually driven by business needs or by an incomplete understanding of the requirements at the start of data modeling. By their nature, changes are frustrating and challenging, but they are a part of the process and, as such, need to be handled and planned for.

Tools

There are several ER modeling tools that can be used for the type of modeling discussed in this module. However, for simple models, Microsoft Visio® is sufficiently powerful.

Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
Changes to requirements.	Plan for and handle changes to requirements.

Lab Review Questions and Answers

Lab: Identify Components in Entity Relationship Modeling

Question and Answers

Lab Review

Question: How do you identify entities?

Answer: Entities can be identified as nouns in the data brief.

Entities represent real-world objects about which information is required to be stored. An entity can represent a physical object, such as a person, place, or a thing; or it could represent a virtual object, such as an event.

Question: What is a relationship?

Answer: Relationships exist between entities in a data model. They represent how one entity is related to another at the level of their respective attributes.

In a conceptual model, a relationship can exist between two entities. It is a named, meaningful link between entities. It represents how the attributes in one entity are related to those in the other, and can be thought of as a verb phrase that links the two entities to form a sentence. For example:

- STUDENT attends COURSE PRESENTATION.
- COMPANY sponsors STUDENT.
- COURSE PRESENTATION is one presentation of COURSE TYPE.

Module 3

Normalization

Contents:

Lesson 1: Fundamentals of Normalization	2
Lesson 2: Normal Form	5
Lesson 3: Denormalization	8
Module Review and Takeaways	10
Lab Review Questions and Answers	11

Lesson 1

Fundamentals of Normalization

Contents:

Question and Answers	3
Demonstration: Identifying Candidate Keys	3

Question and Answers

Question: What type of key can you use when you cannot identify an appropriate primary key from the existing columns for a table?

- () Composite primary key
- () Surrogate key
- () Candidate key

Answer:

- () Composite primary key
- (√) Surrogate key
- () Candidate key

Demonstration: Identifying Candidate Keys

Demonstration Steps

1. Start the 10985B-MIA-DC and 10985B-MIA-SQL virtual machines, and then log on to 10985B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod03** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and wait for setup to complete.
4. Open **Microsoft SQL Server Management Studio**, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
5. On the **File** menu, point to **Open**, and then click **File**.
6. In the **File Open** dialog box, navigate to the **D:\Demofiles\Mod03** folder, click **Candidate Keys.sql**, and then click **Open**.
7. Under the comment **View the columns in the Production.Product table and identify the total number of rows**, select the Transact-SQL statement, and then click **Execute**.
8. Review the results, and in the results pane, in the bottom right corner, note the number of rows that the query returned.
9. Under the comment **Assess ProductID as a candidate key**, select the Transact-SQL statement, and then click **Execute**.
10. Review the results, and in the results pane, in the bottom right corner, note that the number of rows that the query returned is the same as for the query in step 7. This means that **ProductID** is a candidate key because it contains a unique value for each of the rows in the table.
11. Under the comment **Assess Name as a candidate key**, select the Transact-SQL statement, and then click **Execute**.
12. Review the results, and in the results pane, in the bottom right corner, note that the number of rows that the query returned is the same as for the query in step 7. This means that **Name** is a candidate key because it contains a unique value for each of the rows in the table.
13. Under the comment **Assess ProductNumber as a candidate key**, select the Transact-SQL statement, and then click **Execute**.
14. Review the results, and in the results pane, in the bottom right corner, note that the number of rows that the query returned is the same as for the query in step 7. This means that **ProductNumber** is a candidate key because it contains a unique value for each of the rows in the table.

15. Under the comment **Assess Color as a candidate key**, select the Transact-SQL statement, and then click **Execute**.
16. Review the results, and in the results pane, in the bottom right corner, note that the number of rows that the query returned is less than the number of rows that the query in step 7 returned. This means that **Color** is not a candidate key because it does not contain a unique value for each of the rows in the table.
17. Close SQL Server Management Studio without saving changes.

Lesson 2

Normal Form

Contents:

Question and Answers	6
Demonstration: Normalizing to Third Normal Form	7

Question and Answers

Match each description with the correct level of normal form. Indicate your answer by writing the category number to the right of each item.

Items	
1	Provide a primary key as a unique identifier for rows, and eliminate data groups that repeat across columns.
2	Ensure that all non-key columns are functionally dependent on the whole of the primary key.
3	Eliminate attributes that are only transitively dependent on the primary key.

Category 1		Category 2		Category 3
First normal form		Second normal form		Third normal form

Answer:

Category 1		Category 2		Category 3
First normal form		Second normal form		Third normal form
Provide a primary key as a unique identifier for rows, and eliminate data groups that repeat across columns.		Ensure that all non-key columns are functionally dependent on the whole of the primary key.		Eliminate attributes that are only transitively dependent on the primary key.

Demonstration: Normalizing to Third Normal Form

Demonstration Steps

1. In the **D:\Demofiles\Mod03** folder, double-click the **Normalize Book Data.xlsx** file.
2. On the **Raw Data** worksheet, review the data and note the following points:
 - a. The worksheet contains data about a collection of books.
 - b. There are multiple copies of the same book titles, and each copy might be a different pressing. This information is stored in repeating data group columns called **Copy 1**, **Copy 2**, and **Copy 3**.
3. On the **1NF** worksheet, note the changes that have been made to bring the table into first normal form. These changes include:
 - a. Adding a composite primary key consisting of the **Book Title**, **Copy Number**, and **Pressing** columns.
 - b. The repeating data group columns have been removed.
4. Note that the columns **Year Published**, **Author**, **Author Date of Birth**, and **Author Age** all contain repeating data values. This is because these columns are not functionally dependent on the whole of the primary key.
5. On the **2NF** worksheet, note the changes that have been made to conform to second normal form. The table has been decomposed into four tables: **Book**, **Author**, **Copy**, and **Pressing**. The columns in each table depend on the whole of the primary key in their respective tables.
6. On the **3NF** worksheet, note the change that has been made to ensure that the **Author** table conforms to third normal form. The **Author Age** column has been removed because it was transitively dependent on the primary key, through the **Author Date of Birth** non-key column. If required, the age of each author can be calculated by using the date of birth value, so it not necessary to store it.

Lesson 3

Denormalization

Contents:

Question and Answers

9

Question and Answers

Question: True or false? A denormalized database and an unnormalized database are essentially the same thing.

☐ True

☐ False

Answer:

☐ True

☒ False

Module Review and Takeaways

Review Question(s)

Question: Can you think of a data set that you are familiar with, from your place of work or elsewhere, that you could normalize? What steps would you take to normalize to third normal form?

Answer: Answers will vary because students will have access to different data sets, but the basic process should be the same:

1. Eliminate repeating data groups and ensure that every row is uniquely identifiable.
2. Ensure that all attributes depend on the whole of the primary key.
3. Eliminate transitive dependencies.

Lab Review Questions and Answers

Lab: Normalizing Data

Question and Answers

Lab Review

Question: In the suggested solution, the Orders table has a surrogate key as its primary key. Why do you think this key was chosen instead of a composite candidate key, such as the OrderDate and Customer columns?

Answer: There are two good reasons for this choice:

- The primary key in Orders is referenced from the OrderDetails table. Using the composite candidate key would mean that these larger key values would have to be repeated multiple times in the OrderDetails table, which is inefficient. Using the surrogate key instead means that the OrderDetails table references the Orders table by using a much smaller key.
- It is logically possible for a customer to place more than one order in a day, so the combination of OrderDate and Customer is not actually a valid candidate key. This is another example of the importance of understanding the business processes in addition to the data.

Question: In the suggested solution, in the Customers table, are there any other columns that you could break down even further? If so, why might you do this?

Answer: You could break down the following columns:

- Customer could be split into multiple columns, including Title, FirstName, MiddleName, and LastName.
- Address could be split into multiple columns, including Number, Street, City, PostalCode, State, and Country.

You might break down these columns if it made sense for business processes. For example, updating a customer's name if it is misspelt, or if they change their name because of marriage, is easier and less vulnerable to error if you only need to update the last name, instead of the full name. Alternatively, it might be important to be able to query customers based on a part of their address, such as City or Postal Code. In this case, breaking down the address makes these types of queries easier to write and will be more efficient.

Module 4

Relationships

Contents:

Lesson 1: Introduction to Relationships	2
Lesson 2: Planning Referential Integrity	4
Module Review and Takeaways	7
Lab Review Questions and Answers	8

Lesson 1

Introduction to Relationships

Contents:

Question and Answers

3

Question and Answers

Question: You are planning a database that tracks the courses that students attend at a college. Each course can include up to 25 students, and students can enroll on multiple courses at the same time. You have created an ERD that includes the entities Students and Courses. What type of relationship exists between these entities?

- ☐ A one-to-many relationship.
- ☐ A one-to-one relationship.
- ☐ A many-to-many relationship.
- ☐ A super-type/sub-type relationship.

Answer:

- ☐ A one-to-many relationship.
- ☐ A one-to-one relationship.
- ☒ A many-to-many relationship.
- ☐ A super-type/sub-type relationship.

Lesson 2

Planning Referential Integrity

Contents:

Question and Answers	5
Demonstration: Implementing Referential Integrity by Using a Foreign Key	5

Question and Answers

Question: A junior administrator reported that, when she deleted a row in a table in your organization's OLTP database, the action unexpectedly caused the deletion of more than 100 rows. You successfully verify that the necessary foreign key constraints exist. What was the most likely cause of the unexpected deletions?

- ☐ A foreign key constraint was configured with the ON DELETE SET NULL option.
- ☐ A foreign key constraint was configured with the ON DELETE SET DEFAULT option.
- ☐ A foreign key constraint was configured with the ON DELETE CASCADE option.
- ☐ A foreign key constraint was configured with the ON UPDATE CASCADE option.

Answer:

- ☐ A foreign key constraint was configured with the ON DELETE SET NULL option.
- ☐ A foreign key constraint was configured with the ON DELETE SET DEFAULT option.
- ☒ A foreign key constraint was configured with the ON DELETE CASCADE option.
- ☐ A foreign key constraint was configured with the ON UPDATE CASCADE option.

Demonstration: Implementing Referential Integrity by Using a Foreign Key

Demonstration Steps

1. Start the 10985B-MIA-DC and 10985B-MIA-SQL virtual machines, and then log on to 10985B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod04** folder, run **Setup.cmd** as Administrator.
3. Open **Microsoft SQL Server Management Studio** and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
4. On the **File** menu, point to **Open**, click **File**, browse to **D:\Demofiles\Mod04**, click **Referential Integrity.sql**, and then click **Open**.
5. In the query window, under the comment **Create a database and change database context**, select the Transact-SQL statement and then click **Execute**.
6. In the query window, under the comment **Create Customers table with a primary key constraint and four rows**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
7. In the query window, under the comment **Create a table with a foreign key constraint**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
8. In the query window, under the comment **Test foreign key constraint by adding an order with a valid CustomerID value**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
9. Note that the statement completes successfully.
10. In the query window, under the comment **Add an order with an invalid CustomerID value**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
11. In the Results pane, review the message that states that the insert statement conflicted with the foreign key constraint.

12. In the query window, under the comment **Update a CustomerID value in Customers that has no matching value in Orders**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
13. Note that the statement completes successfully.
14. In the query window, under the comment **Update a CustomerID value in Customers that has a matching value in Orders**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
15. In the Results pane, review the message that states that the update statement conflicted with the constraint.
16. In the query window, under the comment **Drop foreign key and add new foreign key that specifies cascading referential integrity**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
17. In the query window, under the comment **Test cascading referential integrity**, review the Transact-SQL statement, select the Transact-SQL statement, and then click **Execute**.
18. In the Results pane, review the results, noting that the row in the **Orders** table was successfully updated.
19. Close SQL Server Management Studio without saving changes.

Module Review and Takeaways

Question: Think about the kinds of processes that exist in your organization. What entities can you identify and what types of relationships exist between them? Think of three or four specific entities—what constraints would you implement to enforce referential integrity, and would you use options such as CASCADE or SET NULL?

Answer: Answers will vary, depending on students' knowledge, opinions, and experience.

Lab Review Questions and Answers

Lab: Planning and Implementing Referential Integrity

Question and Answers

Lab Review

Question: Do you think that it was a good idea to implement the ON DELETE CASCADE and the ON DELETE SET DEFAULT options in the final exercise in the lab? What problems might this potentially cause? What might you have done instead to prevent these problems?

Answer: Cascading the delete to the CustomerDetails table removes all customer information for the deleted customers, which means that it is more difficult to perform data analysis later on, when it is based on customer demographics. Using the SET NULL option in Orders results in order rows that are not associated with a specific customer, but instead have a default catch-all CustomerID value. Removing the context for an order in this way can make it very difficult to aggregate data in a meaningful manner, which again can reduce the usefulness of the data for analytics.

One possible solution would be to use triggers to make copies of the data before deleting it from the tables. This data could then be used for analytical purposes. If it is necessary to eliminate data that identifies a customer, you could selectively save the non-identifying elements, such as the city where the customer lives, or their age, and not save the identifying data, such as their name, phone number, or street address.

Module 5

Performance

Contents:

Lesson 1: Indexing	2
Lesson 2: Query Performance	4
Lesson 3: Concurrency	6
Module Review and Takeaways	8

Lesson 1

Indexing

Contents:

Question and Answers	3
Demonstration: Testing Index Performance	3

Question and Answers

Question: How do you choose between index types?

Answer: How you choose between index types depends on data structure, table sizes, and speed required.

A clustered index causes the data to be sorted in a logical order on one column. There can only be one clustered index per table, so care must be taken in selecting the appropriate column.

A nonclustered index is a separate structure that contains one or more columns as a key, in addition to a pointer back to the source row in the table. Nonclustered indexes may be created on tables that are heaps, or on tables organized by clustered indexes. Therefore, there can be many nonclustered indexes per table.

Nonclustered indexes are often added to tables to improve specific query performance and avoid resource-intensive table scans.

Demonstration: Testing Index Performance

Demonstration Steps

1. In the virtual machine, on the taskbar, click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** box, type **MIA-SQL**. In the **Authentication** box, ensure **Windows Authentication** is selected, and then click **Connect**.
3. On the **File** menu, point to **Open**, click **File**, navigate to the **D:\Demofiles\Mod05\Demo1.sql** script file, and then click **Open**.
4. To turn statistics on, highlight the code under **Step 1** and click **Execute**.
5. To run a query with no indexes, highlight the code under **Step 2** and click **Execute**.
6. To create indexes, highlight the code under **Step 3** and click **Execute**.
7. To run a query with indexes, highlight the code under **Step 4** and click **Execute**.
8. To drop the indexes, highlight the code under **Step 5** and click **Execute**.
9. Close SQL Server Management Studio without saving any changes.

Lesson 2

Query Performance

Contents:

Question and Answers	5
Demonstration: Nonsargable Queries	5

Question and Answers

Question: True or false? The use of the * predicate in SELECT statements has an adverse effect on query performance.

☐ True

☐ False

Answer:

☒ True

☐ False

Demonstration: Nonsargable Queries

Demonstration Steps

1. On the taskbar, click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** box, type **MIA-SQL**. Ensure **Windows Authentication** is selected in the **Authentication** box, and then click **Connect**.
3. On the **File** menu, point to **Open**, click **File**, navigate to the **D:\Demofiles\Mod05\Demo2.sql** script file, and then click **Open**.
4. To turn statistics on, highlight the code under **Step 1** and click **Execute**.
5. To run a query with no indexes, highlight the code under **Step 2** and click **Execute**.
6. To create indexes, highlight the code under **Step 3** and click **Execute**.
7. To run a query with indexes, highlight the code under **Step 4** and click **Execute**.
8. To drop the indexes, highlight the code under **Step 5** and click **Execute**.
9. Close SQL Server Management Studio without saving any changes.

Lesson 3

Concurrency

Contents:

Question and Answers	7
Resources	7
Demonstration: Locking and Blocking	7

Question and Answers

Question: True or false? Isolation levels affect the locks imposed when a transaction is modifying data.

☐ True

☐ False

Answer:

☐ True

☒ False

Resources

Transactions



Additional Reading: For more information, see the *Isolation Level* and *Locking* topics later in this module.

Demonstration: Locking and Blocking

Demonstration Steps

1. On the taskbar, click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** box, type **MIA-SQL**. Ensure **Windows Authentication** is selected in the **Authentication** box, and then click **Connect**.
3. On the **File** menu, point to **Open**, click **File**, navigate to the **D:\Demofiles\Mod05\Demo3A.sql** script file, and then click **Open**.
4. On the **File** menu, point to **Open**, click **File**, navigate to the **D:\Demofiles\Mod05\Demo3B.sql** script file, and then click **Open**.
5. To start the first query, switch to the **Demo3A** tab, highlight the code under **Step 1** and click **Execute**.
6. Immediately switch to the **Demo3B** tab, highlight the code under **Step 2** and click **Execute**.
7. Note that the query does not complete because it is being blocked.
8. Switch to the **Demo3A** tab and, on the toolbar, click **Cancel Executing Query**.
9. Wait until the cancellation has completed and switch to the **Demo3B** tab.
10. Notice that the query has now completed because the locks have been released.
11. Close SQL Server Management Studio without saving any changes.

Module Review and Takeaways

Best Practice

Indexes, joins and the types of search that you perform have a huge effect on query performance so you should consider index and query design carefully. Furthermore, most databases will have many concurrent users—you should consider whether a database that performs optimally in a test environment with one use, would perform optimally in a production environment.

Module 6

Database Objects

Contents:

Lesson 1: Tables	2
Lesson 2: Views	5
Lesson 3: Stored Procedures, Triggers, and Functions	7
Module Review and Takeaways	9
Lab Review Questions and Answers	10

Lesson 1

Tables

Contents:

Question and Answers	3
Demonstration: Creating Databases and Tables	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	Checks the values that are added to a column and prevents the insertion of duplicate values.
2	Inserts a pre-configured value into a column when an INSERT statement does not provide a value for that column.
3	Compares a value to a pre-configured condition to assess whether the value is valid.

Category 1		Category 2		Category 3
Unique constraint		Default constraint		Check constraint

Answer:

Category 1		Category 2		Category 3
Unique constraint		Default constraint		Check constraint
Checks the values that are added to a column and prevents the insertion of duplicate values.		Inserts a pre-configured value into a column when an INSERT statement does not provide a value for that column.		Compares a value to a pre-configured condition to assess whether the value is valid.

Demonstration: Creating Databases and Tables

Demonstration Steps

1. Start the 10985B-MIA-DC and 10985B-MIA-SQL virtual machines, and then log on to 10985B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. Open **Microsoft SQL Server Management Studio**, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
3. On the **File** menu, point to **Open**, click **File**, browse to **D:\Demofiles\Mod06**, click **Create Tables.sql**, and then click **Open**.
4. In the query window, under the comment **Create a database and change database context**, review the Transact-SQL statement, select it, and then click **Execute**.
5. In the query window, under the comment **Create schemas**, review the Transact-SQL statement, select it, and then click **Execute**.
6. In the query window, under the comment **Create tables**, review the Transact-SQL statement, select it, and then click **Execute**.
7. In the query window, under the comment **Add CHECK constraint to Sales.Order**, review the Transact-SQL statement, select it, and then click **Execute**.
8. In the query window, under the comment **Test CHECK constraint**, review the Transact-SQL statement, select it, and then click **Execute**.
9. Review the results, and note that the insert fails because the order date is yesterday's date.
10. In the query window, under the comment **Add DEFAULT constraint to Person.CustomerDemographics**, review the Transact-SQL statement, select it, and then click **Execute**.
11. In the query window, under the comment **Test DEFAULT constraint**, review the Transact-SQL statement, select it, and then click **Execute**.
12. Review the results, and note that the insert succeeds and the default value 'Unknown' is entered.
13. In the query window, under the comment **Add UNIQUE constraint to Person.Employee**, review the Transact-SQL statement, select it, and then click **Execute**.
14. In the query window, under the comment **Test UNIQUE constraint**, review the Transact-SQL statement, select it, and then click **Execute**.
15. Review the results, and note that the first insert succeeds, but the second fails because it contains the duplicate **SSN** value '123'.
16. Close SQL Server Management Studio without saving changes.

Lesson 2

Views

Contents:

Question and Answers	6
Demonstration: Creating and Using Views	6

Question and Answers

Question: True or false? To create an indexed view, you must include the WITH SCHEMABINDING option in the view definition.

☐ True

☐ False

Answer:

☒ True

☐ False

Demonstration: Creating and Using Views

Demonstration Steps

1. In the **D:\Demofiles\Mod06** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
2. In the **User Account Control** dialog box, click **Yes**, and wait for setup to complete.
3. Open **Microsoft SQL Server Management Studio**, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
4. On the **File** menu, point to **Open**, click **File**, browse to **D:\Demofiles\Mod06**, click **Create Views.sql**, and then click **Open**.
5. Under the comment **Create a view**, review the Transact-SQL statements, select them, and then click **Execute**.
6. Under the comment **Query the view**, review the Transact-SQL statements, select them, and then click **Execute**.
7. Under the comment **Query the view with a WHERE clause**, review the Transact-SQL statements, select them, and then click **Execute**.
8. Under the comment **Create an index on the view**, review the Transact-SQL statements, select them, and then click **Execute**.
9. In Object Explorer, expand **Databases**, expand **Demo**, expand **Views**, and note the view that you created in step 5.
10. Expand **dbo.VW_CustomerOrders**, expand **Indexes**, and note the index that you created in step 8.
11. Close SQL Server Management Studio without saving changes.

Lesson 3

Stored Procedures, Triggers, and Functions

Contents:

Question and Answers	8
Demonstration: Creating and Using a Stored Procedure	8

Question and Answers

Question: You want to create a database object to encapsulate Transact-SQL statements in a database to ensure consistency and make it easier for users to access data without having to write complex Transact-SQL statements. The Transact-SQL in the database object that you create will reference multiple tables, must accept input parameters, and will be referenced by users in the FROM clause of SELECT statements. What type of database object should you create?

- () An inline user-defined table-valued function.
- () A stored procedure.
- () A view.
- () An indexed view.
- () A multistatement user-defined table-valued function.

Answer:

- () An inline user-defined table-valued function.
- () A stored procedure.
- () A view.
- () An indexed view.
- (√) A multistatement user-defined table-valued function.

Demonstration: Creating and Using a Stored Procedure

Demonstration Steps

1. In the **D:\Demofiles\Mod06** folder, right-click **Setup.cmd**, click **Run as Administrator**.
2. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to complete.
3. Open **Microsoft SQL Server Management Studio**, and then connect to the **MIA-SQL** instance of the database engine by using Windows Authentication.
4. On the **File** menu, point to **Open**, click **File**, browse to **D:\Demofiles\Mod06**, click **Create Stored Procedure.sql**, and then click **Open**.
5. Under the comment **Create the stored procedure**, review the Transact-SQL statements, select them, and then click **Execute**.
6. Under the comment **Execute the stored procedure**, review the Transact-SQL statement, select it, and then click **Execute**.
7. Under the comment **View the results**, review the Transact-SQL statement, select it, and then click **Execute**.
8. Review the results, noting that a row has been added to the **Person.Customer** table with the values specified in the EXEC statement.
9. Close SQL Server Management Studio without saving any changes.

Module Review and Takeaways

Review Question(s)

Question: Think of a table in a database that you are familiar with, perhaps a database in your place of work. Imagine that you need to create a stored procedure to input new rows into the table. Which input parameters would you need to include, and what data types would you assign to these parameters?

Answer: Answers will vary, depending on students' opinions and experiences.

Lab Review Questions and Answers

Lab: Using SQL Server

Question and Answers

Lab Review

Question: The PRIMARY KEY column in the Sales.Orders table does not have the IDENTITY property set. If the IDENTITY property were set for this column, how would you have to modify the USP_InsertOrders stored procedure?

Answer: You would need to modify the procedure by removing the @OrderID input parameter and the reference to the OrderID column in the INSERT statement.

Question: You created a stored procedure to insert new rows into the Sales.Orders table. What else would you need to do to ensure that new orders are recorded in full, including details of the products in each order, prices, and so on?

Answer: You would need to update the Sales.LineItems table with the appropriate details for each order. You could extend the existing stored procedure to do this.