

ValidTorrent

TRY BEFORE YOU BUY

Download a free sample of any of our exam questions and answers

- > 24/7 customer support, Secure shopping site
- > Free One year updates to match real exam scenarios
- > If you failed your exam after buying our products we will refund the full amount back to you.

Over 61842+ Satisfied Customers

[About Us](#)

Try Before You Buy

Download a free sample of any of our exam questions and answers

- > 24/7 customer support, Secure shopping site
- > Free One year updates to match real exam scenarios
- > If you failed your exam after buying our products we will refund the full amount back to you.

[Free Download](#)

<http://www.validtorrent.com>

High-quality valid dumps & reliable dumps torrent & useful PDF torrent

Exam : **070-761-Deutsch**

Title : Querying Data with
Transact-SQL (70-761
Deutsch Version)

Vendor : Microsoft

Version : DEMO

QUESTION NO: 1

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben, Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie haben eine Tabelle mit dem Namen "Produkte", in der Informationen zu Produkten gespeichert sind, die Ihr Unternehmen verkauft. Die Tabelle enthält eine Spalte mit dem Namen ListPrice, in der die Einzelhandelspreisinformationen für Produkte gespeichert sind. Einige Produkte werden nur firmenintern verwendet. Die Aufzeichnungen für diese Produkte werden in der Tabelle "Produkte" zu Inventur Zwecken geführt. Der Preis für jedes dieser Produkte beträgt 0,00 USD. Kunden dürfen diese Produkte nicht bestellen.

Sie müssen den Listenpreis für Produkte, die weniger als 100 US-Dollar kosten, um 10 Prozent erhöhen. Sie müssen die Preise nur für Produkte erhöhen, die Kunden bestellen dürfen.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```
UPDATE Production.Product
SET ListPrice = ListPrice + 1.1
WHERE ListPrice
BETWEEN 0 and 100
```

Entspricht die Lösung dem Ziel?

A. Nein

B. Ja

Answer: A

Explanation

Products with a price of \$0.00 would also be increased.

QUESTION NO: 2

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben, können Sie NICHT mehr darauf zurückkommen. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie erstellen Indizes in einem Data Warehouse.

Sie haben eine Dimensionstabelle mit dem Namen Table1, die 10.000 Zeilen enthält. Die Zeilen werden verwendet, um mehrere Berichte zu generieren.

Die Berichte werden einer Spalte zugeordnet, die der Primärschlüssel ist.

Der Ausführungsplan enthält Lesezeichensuche für Tabelle1.

Sie stellen fest, dass die Berichte langsamer als erwartet ausgeführt werden.

Sie müssen die Zeit reduzieren, die zum Ausführen der Berichte erforderlich ist.

Lösung: Sie erstellen einen Clustered-Index für die Primärschlüsselspalte.

Erfüllt dies das Ziel?

A. Nein

B. Ja

Answer: B

QUESTION NO: 3

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie haben eine Tabelle mit dem Namen "Produkte", in der Informationen zu Produkten gespeichert sind, die Ihr Unternehmen verkauft. Die Tabelle enthält eine Spalte mit dem Namen ListPrice, in der die Einzelhandelspreisinformationen für Produkte gespeichert sind. Einige Produkte werden nur firmenintern verwendet. Die Aufzeichnungen für diese Produkte werden in der Tabelle "Produkte" zu Inventur Zwecken geführt. Der Preis für jedes dieser Produkte beträgt 0,00 USD. Kunden dürfen diese Produkte nicht bestellen.

Sie müssen den Listenpreis für Produkte, die weniger als 100 US-Dollar kosten, um 10 Prozent erhöhen. Sie müssen die Preise nur für Produkte erhöhen, die Kunden bestellen dürfen.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```
UPDATE Production.Products
SET ListPrice = ListPrice * 1.1
WHERE ListPrice
BETWEEN 0 and 100
```

Entspricht die Lösung dem Ziel?

A. Ja

B. Nein

Answer: B

QUESTION NO: 4

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden nicht in der Lage sein, darauf zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie haben eine Datenbank, die Bestellungen und Lieferungen für Kunden in Nordamerika

verfolgt. Die Datenbank enthält die folgenden Tabellen:
Vertrieb.Kunden

Column	Data type	Notes
CustomerID	int	primary key
CustomerCategoryID	int	foreign key to the Sales.CustomerCategories table
PostalCityID	int	foreign key to the Application.Cities table
DeliveryCityID	int	foreign key to the Application.Cities table
AccountOpenedDate	datetime	does not allow new values
StandardDiscountPercentage	int	does not allow new values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow new values
DeliveryLocation	geography	does not allow new values
PhoneNumber	nvarchar(20)	does not allow new values data is formatted as follows: 425-555-0187

Anwendung.Städte

Column	Data type	Notes
CityID	int	primary key
LatestRecordedPopulation	bigint	null values are permitted

Sales.CustomerCategories

Column	Data type	Notes
CustomerCategoryID	int	primary key
CustomerCategoryName	nvarchar(50)	does not allow null values

Das Entwicklungsteam des Unternehmens entwirft eine Kundenverzeichnis-anwendung. Die Anwendung muss Kunden nach der Vorwahl ihrer Telefonnummer auflisten. Die Vorwahl ist als die ersten drei Zeichen der Telefonnummer definiert.

Die Hauptseite der Anwendung basiert auf einer indizierten Ansicht, die den Bereich und die Telefonnummer für alle Kunden enthält.

Sie müssen die Vorwahl aus dem Feld Telefonnummer zurückgeben.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```
CREATE FUNCTION AreaCode (
    @phoneNumber nvarchar(20)
)
RETURNS
TABLE
WITH SCHEMABINDING
AS
RETURN (
    SELECT TOP 1 @phoneNumber as PhoneNumber, VALUE as AreaCode
    FROM STRING_SPLIT(@phoneNumber, '-')
)
```

Entspricht die Lösung dem Ziel?

A. Nein

B. Ja

Answer: B

Explanation

As the result of the function will be used in an indexed view we should use schemabinding.

References: <https://sqlstudies.com/2014/08/06/schemabinding-what-why/>

QUESTION NO: 5

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungsbildschirm angezeigt.

Sie erstellen eine Tabelle mit dem Namen Products, indem Sie die folgende Transact-SQL-Anweisung ausführen:

```
CREATE TABLE Products (  
    ProductID int IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    ProductName nvarchar(100) NULL,  
    UnitPrice decimal(18, 2) NOT NULL,  
    UnitsInStock int NOT NULL,  
    UnitsOnOrder int NULL  
)
```

Sie haben die folgende gespeicherte Prozedur:

```
CREATE PROCEDURE InsertProduct  
    @ProductName nvarchar(100),  
    @UnitPrice decimal(18,2),  
    @UnitsInStock int,  
    @UnitsOnOrder int  
AS  
BEGIN  
    INSERT INTO Products(ProductName, ProductPrice, ProductsInStock, ProductsOnOrder)  
    VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)  
END
```

Sie müssen die gespeicherte Prozedur ändern, um die folgenden neuen Anforderungen zu erfüllen:

- Produktdatensätze als einzelne Arbeitseinheit einfügen.
- Geben Sie die Fehlernummer 51000 zurück, wenn ein Produkt nicht in die Datenbank eingefügt werden kann.
- Wenn ein Vorgang zum Einfügen eines Produktdatensatzes fehlschlägt, dürfen die Produktinformationen nicht dauerhaft in die Datenbank geschrieben werden.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```

ALTER PROCEDURE InsertProduct
@ProductName nvarchar(100),
@UnitPrice decimal(18,2),
@UnitsInStock int,
@UnitsOnOrder int
AS
BEGIN
    SET XACT_ABORT ON
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Products (ProductName, ProductPrice, ProductsInStock, ProductsOnOrder)
            VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)
        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        IF XACT_STATE() <> 0 ROLLBACK TRANSACTION
        THROW 51000, 'The product could not be created.', 1
    END CATCH
END

```

Entspricht die Lösung dem Ziel?

A. Nein

B. Ja

Answer: A

Explanation

With X_ABORT ON the INSERT INTO statement and the transaction will be rolled back when an error is raised, it would then not be possible to ROLLBACK it again in the IF XACT_STATE() <> 0 ROLLACK TRANSACTION statement.

Note: A transaction is correctly defined for the INSERT INTO ..VALUES statement, and if there is an error in the transaction it will be caught ant he transaction will be rolled back, finally an error 51000 will be raised.

Note: When SET XACT_ABORT is ON, if a Transact-SQL statement raises a run-time error, the entire transaction is terminated and rolled back.

XACT_STATE is a scalar function that reports the user transaction state of a current running request.

XACT_STATE indicates whether the request has an active user transaction, and whether the transaction is capable of being committed.

The states of XACT_STATE are:

0 There is no active user transaction for the current request.

1 The current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction.

2 The current request has an active user transaction, but an error has occurred that has caused the transaction to be classified as an committable transaction.

References:

<https://msdn.microsoft.com/en-us/library/ms188792.aspx>

<https://msdn.microsoft.com/en-us/library/ms189797.aspx>

QUESTION NO: 6

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung,

während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie haben eine Tabelle mit dem Namen "Produkte", in der Informationen zu Produkten gespeichert sind, die Ihr Unternehmen verkauft. Die Tabelle enthält eine Spalte mit dem Namen ListPrice, in der die Einzelhandelspreisinformationen für Produkte gespeichert sind. Einige Produkte werden nur firmenintern verwendet. Die Aufzeichnungen für diese Produkte werden in der Tabelle "Produkte" zu Inventur Zwecken geführt. Der Preis für jedes dieser Produkte beträgt 0,00 USD. Kunden dürfen diese Produkte nicht bestellen.

Sie müssen den Listenpreis für Produkte, die weniger als 100 US-Dollar kosten, um 10 Prozent erhöhen. Sie müssen die Preise nur für Produkte erhöhen, die Kunden bestellen dürfen.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```
UPDATE Production.Products
SET ListPrice = (ListPrice* .1)
WHERE ListPrice <100
```

Entspricht die Lösung dem Ziel?

A. Nein

B. Ja

Answer: A

Explanation

Mathematical equation will only return 10 % of the value.

QUESTION NO: 7

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben, können Sie NICHT mehr darauf zurückkommen. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Sie haben eine Datenbank mit dem Namen DB1, die zwei Tabellen mit den Namen Sales.Customers und Sales.Orders enthält.

Sales.Customers hat eine Fremdschlüsselbeziehung zu einer Spalte namens CustomerID in Sales.Orders.

Sie müssen eine Abfrage empfehlen, die alle Kunden zurückgibt. Die Abfrage muss auch die Anzahl der Bestellungen zurückgeben, die jeder Kunde im Jahr 2016 aufgegeben hat.

Lösung: Sie empfehlen folgende Abfrage:

```
SELECT
    Cust.CustomerName,
    NumberOfOrders = COUNT (Ord.OrderID)
FROM
    Sales.Customers Cust
LEFT JOIN
    Sales.Orders Ord
        ON Cust.CustomerID = Ord.OrderID
GROUP BY
    Cust.CustomerName ;
```

Erfüllt dies das Ziel?

A. Nein

B. Ja

Answer: A

QUESTION NO: 8

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, bei denen die gleichen oder ähnliche Antwortmöglichkeiten verwendet werden. Eine Antwortauswahl kann für mehr als eine Frage in der Reihe richtig sein. Jede Frage ist unabhängig von den anderen Fragen in dieser Reihe. In einer Frage angegebene Informationen und Details gelten nur für diese Frage.

Sie erstellen eine Tabelle, indem Sie die folgende Transact-SQL-Anweisung ausführen:

```
CREATE TABLE Customers (
    CustomerID int NOT NULL PRIMARY KEY CLUSTERED,
    FirstName nvarchar(100) NOT NULL,
    LastName nvarchar(100) NOT NULL,
    TaxIdNumber varchar(20) NOT NULL,
    Address nvarchar(1024) NOT NULL,
    AnnualRevenue decimal(19,2) NOT NULL,
    DateCreated datetime2(2) NOT NULL,
    ValidFrom datetime2(2) GENERATED ALWAYS AS ROW START NOT NULL,
    ValidTo datetime2(2) GENERATED ALWAYS AS ROW END NOT NULL,
    PERIOD FOR SYSTEM_TIME(ValidFrom, ValidTo)
)
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = CustomersHistory))
```

Sie müssen für alle Kunden, die im Jahr 2014 hinzugefügt wurden, normalisierte Daten zurückgeben.

Welche Transact-SQL-Anweisung sollten Sie ausführen?

- A `SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, AnnualRevenue, DateCreated
FROM Customers
GROUP BY GROUPING SETS((FirstName, LastName), (Address), (CustomerID, AnnualRevenue), (CustomerID), ())
ORDER BY CustomerID, FirstName, LastName, Address, AnnualRevenue`
- B `SELECT FirstName, LastName, Address
FROM Customers
FOR SYSTEM_TIME ALL ORDER BY ValidFrom`
- C `SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c
ORDER BY c.CustomerID
FOR JSON AUTO, ROOT('Customers')`
- D `SELECT * FROM (SELECT CustomerID, FirstName, LastName, Address, AnnualRevenue, DateCreated
FROM Customers) AS Customers PIVOT(AVG(AnnualRevenue)
FOR DateCreated IN([2014])) AS PivotCustomers
ORDER BY LastName, FirstName`
- E `SELECT CustomerID, AVG(AnnualRevenue)
AS AverageAnnualRevenue, FirstName, LastName, Address, DateCreated
FROM Customers WHERE YEAR(DateCreated) >= 2014
GROUP BY CustomerID, FirstName, LastName, Address, DateCreated`

- A. Option
- B. Option C
- C. Option B
- D. Option
- E. Option A
- F. Option
- G. Option
- H. Option

Answer: H

Explanation

The following query searches for row versions for Employee row with EmployeeID = 1000 that were active at least for a portion of period between 1st January of 2014 and 1st January 2015 (including the upper boundary):

```
SELECT * FROM Employee  
FOR SYSTEM_TIME  
BETWEEN '2014-01-01 00:00:00.0000000' AND '2015-01-01 00:00:00.0000000' WHERE  
EmployeeID = 1000 ORDER BY ValidFrom; References: https://msdn.microsoft.com/en-us/library/dn935015.aspx
```

QUESTION NO: 9

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im

Überprüfungsbildschirm angezeigt.

Sie erstellen eine Tabelle mit dem Namen Products, indem Sie die folgende Transact-SQL-Anweisung ausführen:

```
CREATE TABLE Products (  
    ProductID int IDENTITY (1, 1), NOT NULL PRIMARY KEY,  
    ProductName nvarchar (100), NULL,  
    UnitPrice decimal (18, 2) NOT NULL,  
    UnitsInStock int NOT NULL,  
    UnitsOnOrder int NULL  
)
```

Sie haben die folgende gespeicherte Prozedur:

```
CREATE PROCEDURE InsertProduct  
    @ProductName nvarchar(100),  
    @UnitPrice decimal (18, 2),  
    @UnitsInStock int,  
    @UnitsOnOrder int  
AS  
BEGIN  
    INSERT INTO Products (ProductName, UnitPrice, UnitsInStock, UnitsOnOrder)  
    VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)  
END
```

Sie müssen die gespeicherte Prozedur ändern, um die folgenden neuen Anforderungen zu erfüllen:

- Produktdatensätze als einzelne Arbeitseinheit einfügen.
- Geben Sie die Fehlernummer 51000 zurück, wenn ein Produkt nicht in die Datenbank eingefügt werden kann.
- Wenn ein Vorgang zum Einfügen eines Produktdatensatzes fehlschlägt, dürfen die Produktinformationen nicht dauerhaft in die Datenbank geschrieben werden.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```
ALTER PROCEDURE InsertProduct
@ProductName nvarchar (100),
@UnitPrice decimal (18, 2),
@UnitsInStock int,
@UnitsOnOrder int
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
            INSERT INTO Products (ProductName, UnitPrice, UnitsInStock, UnitsOnOrder)
            VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)
        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION
        RAISERROR (51000,16, 1)
    END CATCH
END
```

Entspricht die Lösung dem Ziel?

A. Nein

B. Ja

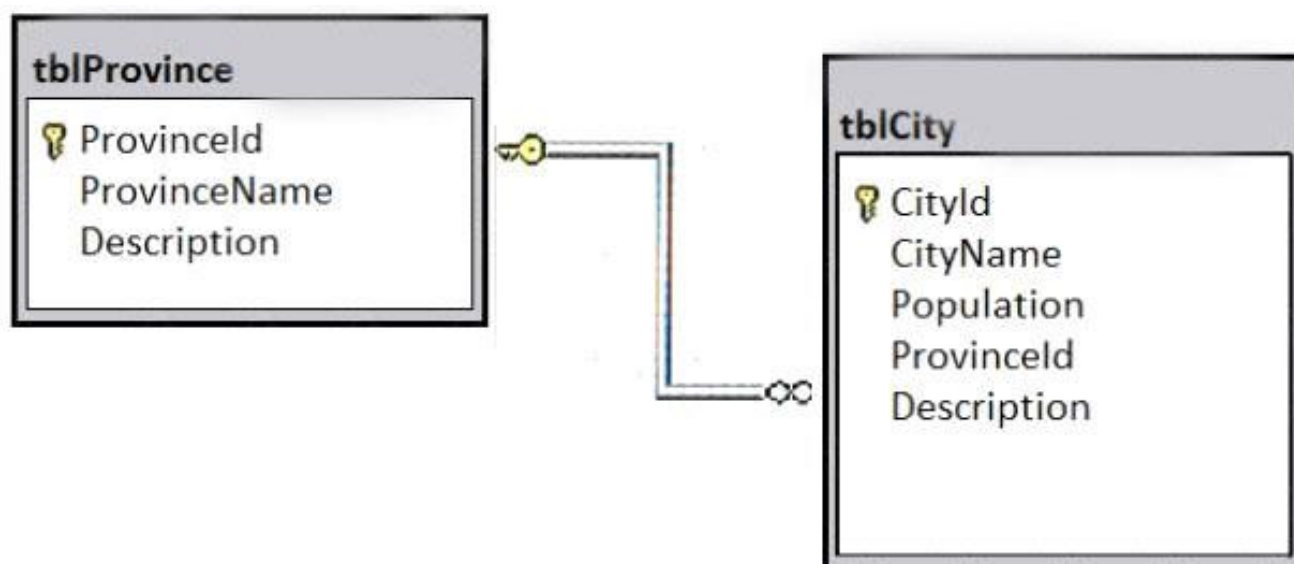
Answer: A

QUESTION NO: 10

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungs Bildschirm angezeigt.

Eine Datenbank verfügt über zwei Tabellen, wie im folgenden Datenbankdiagramm dargestellt:



Sie müssen alle Provinzen auflisten, die mindestens zwei große Städte haben. Eine Großstadt hat mindestens eine Million Einwohner. Die Abfrage muss die folgenden Spalten zurückgeben:

- tblProvince.Provinceld
- tblProvince.ProvinceName
- a derived column named LargeCityCount that presents the total count of large cities for the province

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```

SELECT P.ProvinceId, P.ProvinceName, CitySummary.LargeCityCount
FROM tblProvince P
CROSS APPLY (
    SELECT COUNT(*) AS LargeCityCount FROM tblCity C
    WHERE C.Population >= 1000000 AND C.ProvinceId = P.ProvinceId
) CitySummary
WHERE CitySummary.LargeCityCount >= 2
  
```

Entspricht die Lösung dem Ziel?

- A. Ja
- B. Nein

Answer: A

Explanation

The requirement to list all provinces that have at least two large cities is met by the WHERE CitySummary.LargeCityCount >=2 clause.

CROSS APPLY will work fine here.

Note:

The APPLY operator allows you to invoke a table-valued function for each row returned by an outer table expression of a query. The table-valued function acts as the right input and the outer table expression acts as the left input. The right input is evaluated for each row from the left input and the rows produced are combined for the final output. The list of columns produced by the APPLY operator is the set of columns in the left input followed by the list of columns returned by the right input.

There are two forms of APPLY: CROSS APPLY and OUTER APPLY. CROSS APPLY returns only rows from the outer table that produce a result set from the table-valued function.

OUTER APPLY returns both rows that produce a result set, and rows that do not, with NULL values in the columns produced by the table-valued function.

References: [https://technet.microsoft.com/en-us/library/ms175156\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175156(v=sql.105).aspx)

QUESTION NO: 11

Sie führen die folgende Transact-SQL-Anweisung aus:

```
CREATE TABLE Employees (  
    EmployeeID int IDENTITY(1, 1) PRIMARY KEY NOT NULL,  
    FirstName nvarchar(30) NOT NULL,  
    LastName nvarchar(40) NOT NULL,  
    Title nvarchar(50) NOT NULL,  
    DepartmentID smallint NOT NULL,  
    ManagerID int NULL  
)
```

Sie müssen eine gespeicherte Prozedur erstellen, die die folgenden Anforderungen erfüllt:

- * Fügt Daten in die Employees-Tabelle ein.
 - * Verarbeitet alle Datenänderungen als eine Arbeitseinheit.
 - * Stellt den Schweregrad der Ausnahme auf 16 und eine Fehlernummer von 60.000 ein, wenn ein Fehler auftritt.
 - * Wenn eine Transact-SQL-Anweisung einen Laufzeitfehler auslöst, wird die gesamte Arbeitseinheit beendet und zurückgesetzt, und die Zeilennummer in der Anweisung wird angegeben, in der der Fehler aufgetreten ist.
 - * Fügt den Wert Neuer Mitarbeiter für die Spalte Titel ein, wenn kein Titel angegeben ist.
- Wie soll die Transact-SQL-Anweisung ausgefüllt werden? Ziehen Sie zum Antworten das entsprechende Transact-SQL-Segment auf das richtige Ziel. Jedes Transact-SQL-Segment kann einmal, mehrmals oder gar nicht verwendet werden.
- Möglicherweise müssen Sie die geteilte Leiste zwischen den Fenstern ziehen oder einen Bildlauf durchführen, um den Inhalt anzuzeigen.
- HINWEIS: Jede richtige Auswahl ist einen Punkt wert.

Transact-SQL segments

RAISERROR (60000, 16, 1)

THROW 60000, 'The record was not added.', 1

IF XACT_STATE () <> 0 ROLLBACK TRANSACTION

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION

SAVE TRANSACTION AddEmployee

COMMIT TRANSACTION

Answer Area

```
CREATE PROCEDURE ADDEmployee
    @FirstName nvarchar(30),
    @LastName nvarchar(40),
    @Title nvarchar(50) = 'New Employee',
    @DepartmentID smallint,
    @ManagerID int
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO Employees(FirstName, LastName, Title, DepartmentID, ManagerID)
        VALUES (@FirstName, @LastName, @Title, @DepartmentID, @ManagerID)
        Transact-SQL segment
    END TRY
    BEGIN CATCH
        Transact-SQL segment
        Transact-SQL segment
    END CATCH
```

Answer:

Transact-SQL segments

RAISERROR (60000, 16, 1)

THROW 60000, 'The record was not added.', 1

IF XACT_STATE () <> 0 ROLLBACK TRANSACTION

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION

SAVE TRANSACTION AddEmployee

COMMIT TRANSACTION

Answer Area

```
CREATE PROCEDURE ADDEmployee
    @FirstName nvarchar(30),
    @LastName nvarchar(40),
    @Title nvarchar(50) = 'New Employee',
    @DepartmentID smallint,
    @ManagerID int
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO Employees(FirstName, LastName, Title, DepartmentID, ManagerID)
        VALUES (@FirstName, @LastName, @Title, @DepartmentID, @ManagerID)
        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION
        RAISERROR (60000, 16, 1)
    END CATCH
```

```

AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
        INSERT INTO Employees(FirstName, LastName, Title, DepartmentID, ManagerID
        VALUES (@FirstName, @LastName, @Title, @DepartmentID, @ManagerID
        COMMIT TRANSACTION
    END TRY

    BEGIN CATCH
        IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION
        RAISERROR (60000, 16, 1)
    END CATCH
END

```

QUESTION NO: 12

Sie haben eine Tabelle mit dem Namen Cities, die die folgenden zwei Spalten enthält: CityID und CityName. Die Spalte CityID verwendet den Datentyp int und CityName verwendet nvarchar (max).

Sie haben eine Tabelle mit dem Namen RawSurvey. Jede Zeile enthält eine Kennung für eine Frage und die Anzahl der Personen, die auf diese Frage aus jeder der vier Städte geantwortet haben. Die Tabelle enthält die folgenden repräsentativen Daten:

QuestionID	Tokyo	Boston	London	New York
Q1	1	42	48	51
Q2	22	39	58	42
Q3	29	41	61	33
Q4	62	70	60	50
Q5	63	31	41	21
Q6	32	1	16	34

Eine Berichtstabelle mit dem Namen SurveyReport enthält die folgenden Spalten: CityID, QuestionID und RawCount, wobei RawCount der Wert aus der RawSurvey-Tabelle ist. Sie müssen eine Transact-SQL-Abfrage schreiben, um die folgenden Anforderungen zu erfüllen:

* Abrufen von Daten aus der RawSurvey-Tabelle im Format der SurveyReport-Tabelle.

- * Die CityID muss die CityID der befragten Stadt enthalten.
- * Die Reihenfolge der Städte in allen SELECT-Abfragen muss mit der Reihenfolge in der RawSurvey-Tabelle übereinstimmen.
- * Die Reihenfolge der Städte in allen IN-Anweisungen muss mit der Reihenfolge in der RawSurvey-Tabelle übereinstimmen.

Erstellen Sie die Abfrage anhand der folgenden Richtlinien:

- * Verwenden Sie einteilige Namen, um auf Tabellen und Spalten zu verweisen, sofern dies nicht möglich ist.
- * ALL SELECT-Anweisungen müssen Spalten angeben.
- * Verwenden Sie keine Spalten- oder Tabellen-Aliase, außer den angegebenen.
- * Umschließen Sie Objektnamen nicht mit eckigen Klammern.

Keywords

ADD	EXIT	PROC
ALL	EXTERNAL	PROCEDURE
ALTER	FETCH	PUBLIC
AND	FILE	RAISERROR
ANY	FILLFACTOR	READ
AS	FORFOREIGN	READTEXT
ASC	FREETEXT	RECONFIGURE
AUTHORIZATION	FREETEXTTABLE	REFERENCES
BACKUP	FROM	REPLICATION
BEGIN	FULL	RESTORE
BETWEEN	FUNCTION	RESTRICT
BREAK	GOTO	RETURN
BROWSE	GRANT	REVERT
BULK	GROUP	REVOKE
BY	HAVING	RIGHT
CASCADE	HOLDLOCK	ROLLBACK
CASE	IDENTITY	ROWCOUNT
CHECK	IDENTITY_INSERT	ROWGUIDCOL
CHECKPOINT	IDENTITYCOL	RULE
CLOSE	IF	SAVE
CLUSTERED	IN	SCHEMA
COALESCE	INDEX	SECURITYAUDIT
COLLATE	INNER	SELECT
COLUMN	INSERT	SEMANTICKEYPHRASETABLE
COMMIT	INTERSECT	SEMANTICSIMILARITYDETAILSTABLE
COMPUTE	INTO	SEMANTICSIMILARITYTABLE
CONCAT	IS	SESSION_USER
CONSTRAINT	JOIN	SET
CONTAINS	KEY	SETUSER
CONTAINSTABLE	KILL	SHUTDOWN
CONTINUE	LEFT	SOME
CONVERT	LIKE	STATISTICS
CREATE	LINENO	SYSTEM_USER
CROSS	LOAD	TABLE
CURRENT	MERGE	TABLESAMPLE
CURRENT_DATE	NATIONAL	TEXTSIZE
CURRENT_TIME	NOCHECK	THEN
CURRENT_TIMESTAMP	NONCLUSTERED	TO
CURRENT_USER	NOT	TOP
CURSOR	NULL	TRAN
DATABASE	NULLIF	TRANSACTION
DBCC	OF	TRIGGER
DEALLOCATE	OFF	TRUNCATE
DECLARE	OFFSETS	TRY_CONVERT
DEFAULT	ON	TSEQUAL
DELETE	OPEN	UNION
DENY	OPENDATASOURCE	UNIQUE
DESC	OPENQUERY	UNPIVOT
DISK	OPENROWSET	UPDATE
DISTINCT	OPENXML	UPDATETEXT
DISTRIBUTED	OPTION	USE
DOUBLE	OR	USER
DROP	ORDER	VALUES
DUMP	OUTER	VARYING
ELSE	OVER	VIEW
END	PERCENT	WAITFOR
ERRLVL	PIVOT	WHEN
ESCAPE	PLAN	WHERE
ESCEPT	PRECISION	WHILE
EXEC	PRIMARY	WITH
EXECUTE	PRINT	WITHIN GROUP
EXISTS		WRITETEXT

Ein Teil des richtigen Transact-SQL wurde im Antwortbereich unten bereitgestellt. Geben Sie den Code in den Antwortbereich ein, der das Problem löst und die angegebenen Ziele oder Anforderungen erfüllt. Sie können Code sowohl innerhalb als auch unterhalb des bereitgestellten Codes hinzufügen.

```
1  SELECT Rawcount
2  from (select cityid,questionid,rawcount) AS t1
3  unpivot
4  (rawcount for questionid in (QuestionID)) AS t2
```

Verwenden Sie die Schaltfläche Syntax überprüfen, um Ihre Arbeit zu überprüfen. Alle Syntax- oder Rechtschreibfehler werden nach Zeilen- und Zeichenposition gemeldet.

Answer:

Please see explanation for answer.

Explanation

1 SELECT Rawcount

2 from (select cityid,questioned,rawcount) AS t1

3 unpivot

4 (rawcount for questioned in (QuestionID)) AS t2

5 JOIN t2

6. ON t1.CityName = t2.cityName

UNPIVOT must be used to rotate columns of the Rawsurvey table into column values.

References: [https://technet.microsoft.com/en-us/library/ms177410\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms177410(v=sql.105).aspx)

QUESTION NO: 13

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario darstellen. Jede Frage in der Reihe enthält eine eindeutige Lösung, mit der die angegebenen Ziele erreicht werden können. Einige Fragensätze haben möglicherweise mehr als eine richtige Lösung, während andere möglicherweise keine richtige Lösung haben.

Nachdem Sie eine Frage in diesem Abschnitt beantwortet haben. Sie werden NICHT in der Lage sein, dorthin zurückzukehren. Infolgedessen werden diese Fragen nicht im Überprüfungsbildschirm angezeigt.

Sie erstellen eine Tabelle mit dem Namen Products, indem Sie die folgende Transact-SQL-Anweisung ausführen:

```
CREATE TABLE Products (
    ProductID int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    ProductName nvarchar(100) NULL,
    UnitPrice decimal(18, 2) NOT NULL,
    UnitsInStock int NOT NULL,
    UnitsOnOrder int NULL
)
```

Sie haben die folgende gespeicherte Prozedur:

```

CREATE PROCEDURE InsertProduct
    @ProductName nvarchar(100),
    @UnitPrice decimal(18,2),
    @UnitsInStock int,
    @UnitsOnOrder int
AS
BEGIN
    INSERT INTO Products(ProductName, ProductPrice, ProductsInStock, ProductsOnOrder)
    VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)
END

```

Sie müssen die gespeicherte Prozedur ändern, um die folgenden neuen Anforderungen zu erfüllen:

- Produktdatensätze als einzelne Arbeitseinheit einfügen.
- Geben Sie die Fehlernummer 51000 zurück, wenn ein Produkt nicht in die Datenbank eingefügt werden kann.
- Wenn ein Vorgang zum Einfügen eines Produktdatensatzes fehlschlägt, dürfen die Produktinformationen nicht dauerhaft in die Datenbank geschrieben werden.

Lösung: Sie führen die folgende Transact-SQL-Anweisung aus:

```

ALTER PROCEDURE InsertProduct
    @ProductName nvarchar(100),
    @UnitPrice decimal(18,2),
    @UnitsInStock int,
    @UnitsOnOrder int
AS
BEGIN
    BEGIN TRY
        INSERT INTO Products(ProductName, ProductPrice, ProductsInStock, ProductsOnOrder)
        VALUES (@ProductName, @UnitPrice, @UnitsInStock, @UnitsOnOrder)
    END TRY
    BEGIN CATCH
        THROW 51000, 'The product could not be created.', 1
    END CATCH
END

```

Entspricht die Lösung dem Ziel?

A. Ja

B. Nein

Answer: A

Explanation

If the INSERT INTO statement raises an error, the statement will be caught and an error 51000 will be thrown.

In this case no records will have been inserted.

Note:

You can implement error handling for the INSERT statement by specifying the statement in a TRY...CATCH construct.

If an INSERT statement violates a constraint or rule, or if it has a value incompatible with the data type of the column, the statement fails and an error message is returned.

References: <https://msdn.microsoft.com/en-us/library/ms174335.aspx>

QUESTION NO: 14

Hinweis: Diese Frage ist Teil einer Reihe von Fragen, die dasselbe Szenario verwenden. Zur

Vereinfachung wird das Szenario in jeder Frage wiederholt. Jede Frage stellt ein anderes Ziel und eine andere Antwortauswahl dar, aber der Text des Szenarios ist in jeder Frage dieser Reihe genau der gleiche.

Sie haben eine Datenbank, die Bestellungen und Lieferungen für Kunden in Nordamerika verfolgt. Die Systemversionierung ist für alle Tabellen aktiviert. Die Datenbank enthält die Tabellen Sales.Customers, Application.Cities und Sales.CustomerCategories.

Details zur Sales.Customers-Tabelle finden Sie in der folgenden Tabelle:

Column	Data type	Notes
CustomerId	int	primary key
CustomerCategoryId	int	foreign key to the Sales.CustomerCategories table
PostalCityID	int	foreign key to the Application.Cities table
DeliveryCityID	int	foreign key to the Application.Cities table
AccountOpenedDate	datetime	does not allow values
StandardDiscountPercentage	int	does not allow values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow values
DeliveryLocation	geography	does not allow values
PhoneNumber	nvarchar(20)	does not allow values
ValidFrom	datetime2(7)	does not allow values, GENERATED ALWAYS AS ROW START
ValidTo	datetime2(7)	does not allow values, GENERATED ALWAYS AS ROW END

Details zur Application.Cities-Tabelle finden Sie in der folgenden Tabelle:

Column	Data type	Notes
CityID	int	primary key
LatestRecordedPopulation	bigint	null values are permitted

Details zur Tabelle "Sales.CustomerCategories" werden in der folgenden Tabelle angezeigt:

Column	Data type	Notes
CustomerCategoryID	int	primary key
CustomerCategoryName	nvarchar(50)	does not allow null values

Sie müssen eine Abfrage erstellen, die die folgenden Anforderungen erfüllt:

- Geben Sie für Kunden, die sich nicht in einem Guthaben befinden, die Kunden-ID und die zuletzt erfasste Bevölkerung für den Lieferort zurück, der dem Kunden zugeordnet ist.
- Geben Sie für Kunden, die sich in einem Guthaben befinden, die Kunden-ID und die zuletzt erfasste Bevölkerung für die dem Kunden zugeordnete Poststadt zurück.

Welche zwei Transact-SQL-Abfragen erreichen das Ziel? Jede richtige Antwort bietet eine vollständige Lösung.

- A**
- ```
SELECT CustomerID, LatestRecordedPopulation
FROM Sales.Customers
CROSS JOIN Application.Citites
WHERE (IsOnCreditHold = 0 AND DeliveryCityID = CityID)
OR (IsOnCreditHold = 1 AND PostalCityID = CityID)
```
- B**
- ```
SELECT CustomerID, LatestRecordedPopulation
FROM Sales.Customers
INNER JOIN Application.Citites AS A
ON A.CityID = IIF(IsOnCreditHold = 0, DeliveryCityID, PostalCityID)
```
- C**
- ```
SELECT CustomerID, ISNULL(A.LatestRecordedPopulation, B.LatestRecorded Population)
FROM Sales.Customers
INNER JOIN Application.Citites AS A ON A.CityID = DeliveryCityID
INNER JOIN Application.Citites AS B ON B.CityID = PostalCityID
WHERE IsOnCreditHold = 0
```
- D**
- ```
SELECT CustomerID, LatestRecordedPopulation,
IIF(IsOnCreditHold = 0, DeliveryCityID, PostalCityID) As CityId
FROM Sales.Customers
INNER JOIN Application.Citites AS A ON A.CityID = CityId
```

- A. Option C
B. Option D
C. Option B
D. Option A

Answer: C,D

Explanation

Using Cross Joins

A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table.

However, if a WHERE clause is added, the cross join behaves as an inner join.

B: You can use the IIF in the ON-statement.

IIF returns one of two values, depending on whether the Boolean expression evaluates to true or false in SQL Server.

References:

[https://technet.microsoft.com/en-us/library/ms190690\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms190690(v=sql.105).aspx)

<https://msdn.microsoft.com/en-us/library/hh213574.aspx>

QUESTION NO: 15

Sie haben eine Datenbank mit dem Namen DB1, die eine Tabelle mit dem Namen HR.Employees enthält. HR.Employees enthält zwei Spalten mit den Namen ManagerID und EmployeeID. ManagerID bezieht sich auf EmployeeID.

Sie müssen eine Abfrage erstellen, die eine Liste aller Mitarbeiter, den Manager jedes Mitarbeiters und die numerische Ebene jedes Mitarbeiters in der Hierarchie Ihrer Organisation zurückgibt.

Welche fünf Anweisungen sollten Sie der Abfrage nacheinander hinzufügen? Verschieben Sie zum Beantworten die entsprechenden Anweisungen aus der Liste der Anweisungen in den Antwortbereich und ordnen Sie sie in der richtigen Reihenfolge an.

Statements

```
SELECT Employees.ManagerId, Employees.EmployeeId,
EmployeeLevel+1
FROM Employees
JOIN Managers ON Employees.EmployeeId =
Managers.ManagerId)
```

```
WITH Managers AS (
```

```
SELECT*
FROM Managers
ORDER BY ManagerID
```

```
SELECT ManagerId, EmployeeId, 0 AS EmployeeLevel
FROM Employees
WHERE ManagerId IS NULL
```

```
UNION ALL
```

```
UNION
```

Answer Area



Answer:

Statements

```
SELECT Employees.ManagerId, Employees.EmployeeId,
EmployeeLevel+1
FROM Employees
JOIN Managers ON Employees.EmployeeId =
Managers.ManagerId)
```

```
WITH Managers AS (
```

```
SELECT*
FROM Managers
ORDER BY ManagerID
```

```
SELECT ManagerId, EmployeeId, 0 AS EmployeeLevel
FROM Employees
WHERE ManagerId IS NULL
```

```
UNION ALL
```

```
UNION
```

Answer Area



The Answer Area contains five code blocks, each enclosed in a dashed red border. The first four blocks are also enclosed in a larger dashed red border. The code blocks are:

```
SELECT Employees.ManagerId, Employees.EmployeeId,
EmployeeLevel+1
FROM Employees
JOIN Managers ON Employees.EmployeeId =
Managers.ManagerId)
```

```
WITH Managers AS (
```

```
SELECT*
FROM Managers
ORDER BY ManagerID
```

```
SELECT ManagerId, EmployeeId, 0 AS EmployeeLevel
FROM Employees
WHERE ManagerId IS NULL
```

```
UNION ALL
```

Explanation

Answer Area

```
SELECT Employees.ManagerId, Employees.EmployeeId,  
EmployeeLevel+1  
FROM Employees  
JOIN Managers ON Employees.EmployeeId =  
Managers.ManagerId)
```

```
WITH Managers AS (
```

```
SELECT*  
FROM Managers  
ORDER BY ManagerID
```

```
SELECT ManagerId, EmployeeId, 0 AS EmployeeLevel  
FROM Employees  
WHERE ManagerId IS NULL
```

```
UNION ALL
```

References:

<https://blog.sqlauthority.com/2012/04/24/sql-server-introduction-to-hierarchical-query-using-a-recursive-cte-a-pr>

QUESTION NO: 16

Sie müssen eine Transact-SQL-Anweisung entwickeln, die die folgenden Anforderungen erfüllt:

- * Die Anweisung muss einen benutzerdefinierten Fehler zurückgeben, wenn beim Aktualisieren einer Tabelle Probleme auftreten.
- * Die Fehlernummer muss der Wert sein
- * Der Schweregrad des Fehlers muss sein
- * Eine Microsoft SQL Server-Warnung muss ausgelöst werden, wenn der Fehler auftritt.

Welches Transact-SQL-Segment sollten Sie für jede Anforderung verwenden? Wählen Sie zum Beantworten die entsprechenden Transact-SQL-Segmente im Antwortbereich aus.

Answer Area

Requirement	Transact-SQL segment
Check for error condition	<pre>BEGIN TRANSACTION...END TRANSACTOIN TRY_PARSE BEGIN...END BEGIN CATCH...END CATCH</pre>
Custom error implementation	<pre>THROW 50555, 'The update failed.', 1 RAISERROR (50555,14,1 'The update failed.') WITH LOG RAISERROR (50555,14,1 'The update failed.') WITH NOWAIT RAISERROR (50555, 'The update failed.')</pre>

Answer:

Answer Area

Requirement	Transact-SQL segment
Check for error condition	<pre>BEGIN TRANSACTION...END TRANSACTOIN TRY_PARSE BEGIN...END BEGIN CATCH...END CATCH</pre>
Custom error implementation	<pre>THROW 50555, 'The update failed.', 1 RAISERROR (50555,14,1 'The update failed.') WITH LOG RAISERROR (50555,14,1 'The update failed.') WITH NOWAIT RAISERROR (50555, 'The update failed.')</pre>

Explanation

Requirement	Transact-SQL segment
Check for error condition	<pre>BEGIN TRANSACTION...END TRANSACTOIN TRY_PARSE BEGIN...END BEGIN CATCH...END CATCH</pre>
Custom error implementation	<pre>THROW 50555, 'The update failed.', 1 RAISERROR (50555,14,1 'The update failed.') WITH LOG RAISERROR (50555,14,1 'The update failed.') WITH NOWAIT RAISERROR (50555, 'The update failed.')</pre>

RAISERROR generates an error message and initiates error processing for the session. RAISERROR can either reference a user-defined message stored in the sys.messages

catalog view or build a message dynamically. The message is returned as a server error message to the calling application or to an associated CATCH block of a TRY...CATCH construct. New applications should use THROW instead.

Note: RAISERROR syntax:

```
RAISERROR( { msg_id | msg_str | @local_variable }
{ ,severity ,state }
[ ,argument [ ,...n ] ] )
[ WITH option [ ,...n ] ]
```

The LOG option logs the error in the error log and the application log for the instance of the Microsoft SQL Server Database Engine.

References:

<https://msdn.microsoft.com/en-us/library/ms178592.aspx>

QUESTION NO: 17

Sie haben eine Datenbank, die die folgenden Tabellen enthält:

Kunde

Column name	Data type	Nullable	Default value
CustomerId	int	No	Identity property
FirstName	varchar(30)	Yes	
LastName	varchar(30)	No	
CreditLimit	money	No	

CustomerAudit

Column name	Data type	Nullable	Default value
CustomerId	int	No	
DateChanged	datetime	No	GETDATE()
OldCreditLimit	money	No	
NewCreditLimit	money	No	
ChangedBy	varchar(100)	No	SYSTEM USER

Wenn der Wert der Spalte CustomerID gleich 3 ist, müssen Sie den Wert der Spalte CreditLimit für den Kunden auf 1000 aktualisieren. Sie müssen sicherstellen, dass die Änderung des Datensatzes in der Customer-Tabelle in der CustomerAudit-Tabelle aufgezeichnet wird.

Welche Transact-SQL-Anweisung sollten Sie ausführen?

- A**
- ```
UPDATE Customer
SET CreditLimit = 1000
WHERE CustomerId = 3
INSERT INTO dbo.CustomerAudit (CustomerId, OldCreditLimit, NewCreditLimit)
SELECT CustomerId, CreditLimit, CreditLimit
FROM Customer
WHERE CustomerId = 3
```
- B**
- ```
UPDATE Customer
SET CreditLimit = 1000
WHERE CustomerId = 3
INSERT INTO dbo.CustomerAudit (CustomerId, OldCreditLimit, NewCreditLimit)
SELECT CustomerId, CreditLimit, CreditLimit
FROM Customer
```
- C**
- ```
UPDATE Customer
SET CreditLimit = 1000
OUTPUT inserted.CustomerId, inserted.CreditLimit, deleted.CreditLimit
INTO CustomerAudit (CustomerId, OldCreditLimit, NewCreditLimit)
WHERE CustomerId = 3
```
- D**
- ```
UPDATE Customer
SET CreditLimit = 1000
OUTPUT inserted.CustomerId, deleted.CreditLimit, inserted.CreditLimit
INTO CustomerAudit (CustomerId, OldCreditLimit, NewCreditLimit)
WHERE CustomerId = 3
```

- A. Option C
- B. Option A
- C. Option B
- D. Option D

Answer: D

Explanation

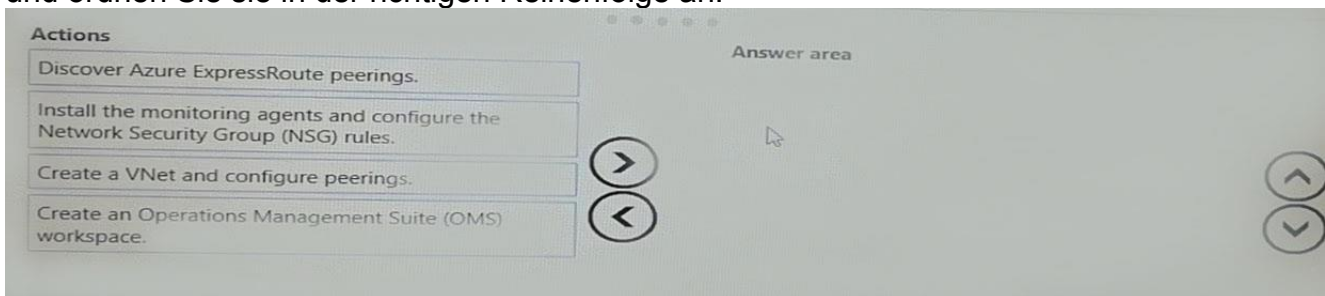
The OUTPUT Clause returns information from, or expressions based on, each row affected by an INSERT, UPDATE, DELETE, or MERGE statement. These results can be returned to the processing application for use in such things as confirmation messages, archiving, and other such application requirements. The results can also be inserted into a table or table variable. Additionally, you can capture the results of an OUTPUT clause in a nested INSERT, UPDATE, DELETE, or MERGE statement, and insert those results into a target table or view. Note: If the column modified by the .RITE clause is referenced in an OUTPUT clause, the complete value of the column, either the before image in deleted.column_name or the after image in inserted.column_name, is returned to the specified column in the table variable.

QUESTION NO: 18

Ihr Unternehmen plant, Network Performance Monitor (NPM) für eine vorhandene Azure ExpressRoute-Verbindung zu verwenden.

Sie müssen NPM konfigurieren.

Welche drei Aktionen sollten Sie nacheinander ausführen? Verschieben Sie zum Beantworten die entsprechenden Aktionen aus der Liste der Aktionen in den Antwortbereich und ordnen Sie sie in der richtigen Reihenfolge an.



Answer:

