70-761$^{\text{Q\&As}}$

Querying Data with Transact-SQL

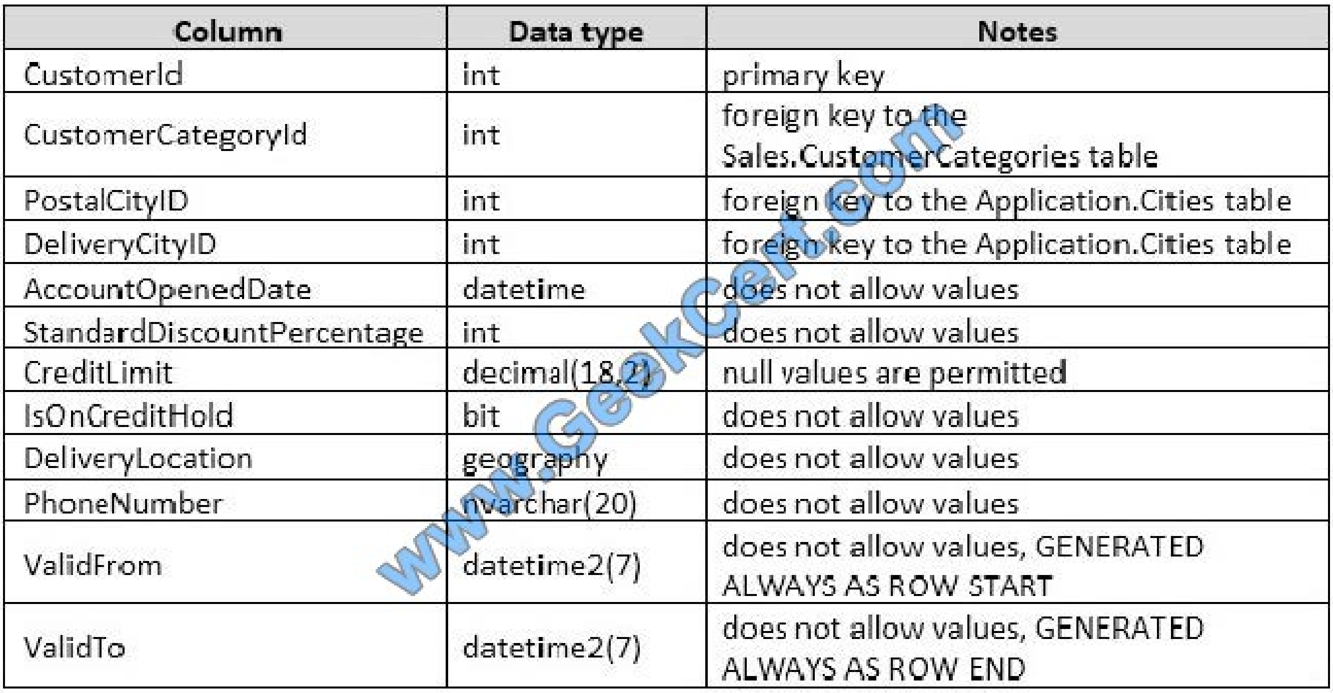
Pass Microsoft 70-761 Exam with 100% Guarantee

QUESTION 1

Note: This question is part of a series of questions that use the same scenario. For your convenience, the scenario is repeated in each question. Each question presents a different goal and answer choices, but the text of the scenario is exactly the same in each question on this series.

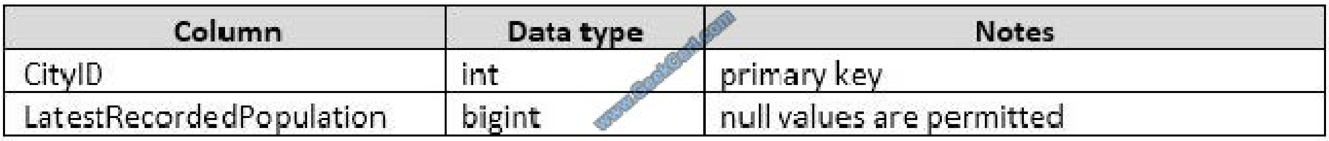
You have a database that tracks orders and deliveries for customers in North America. System versioning is enabled for all tables. The database contains the Sales.Customers, Application.Cities, and Sales.CustomerCategories tables.

Details for the Sales.Customers table are shown in the following table:



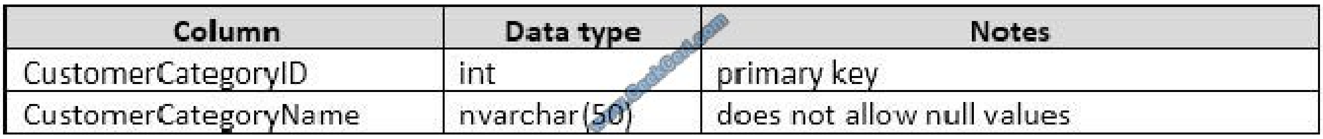
Column	Data type	Notes
CustomerId	int	primary key
CustomerCategoryId	int	foreign key to the Sales.CustomerCategories table
PostalCityID	int	foreign key to the Application.Cities table
DeliveryCityID	int	foreign key to the Application.Cities table
AccountOpenedDate	datetime	does not allow values
StandardDiscountPercentage	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 for the Application.Cities table are shown in the following table:



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

Details for the Sales.CustomerCategories table are shown in the following table:



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

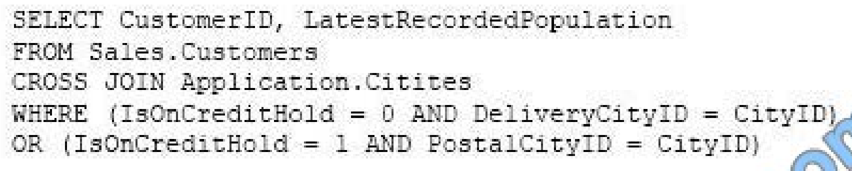
You need to create a query that meets the following requirements:

For customers that are not on a credit hold, return the CustomerID and the latest recorded population for the delivery city that is associated with the customer.

For customers that are on a credit hold, return the CustomerID and the latest recorded population for the postal city that is associated with the customer.

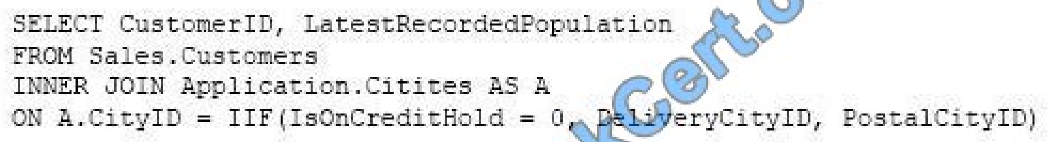
Which two Transact-SQL queries will achieve the goal? Each correct answer presents a complete solution.

A.



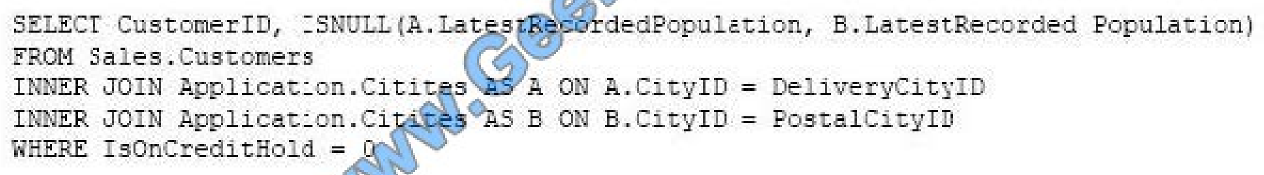
```
SELECT CustomerID, LatestRecordedPopulation
FROM Sales.Customers
CROSS JOIN Application.Citites
WHERE (IsOnCreditHold = 0 AND DeliveryCityID = CityID)
OR (IsOnCreditHold = 1 AND PostalCityID = CityID)
```

B.



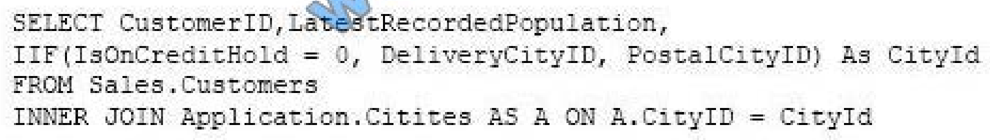
```
SELECT CustomerID, LatestRecordedPopulation
FROM Sales.Customers
INNER JOIN Application.Citites AS A
ON A.CityID = IIF(IsOnCreditHold = 0, DeliveryCityID, PostalCityID)
```

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. B. C. D.

Correct Answer: A

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.

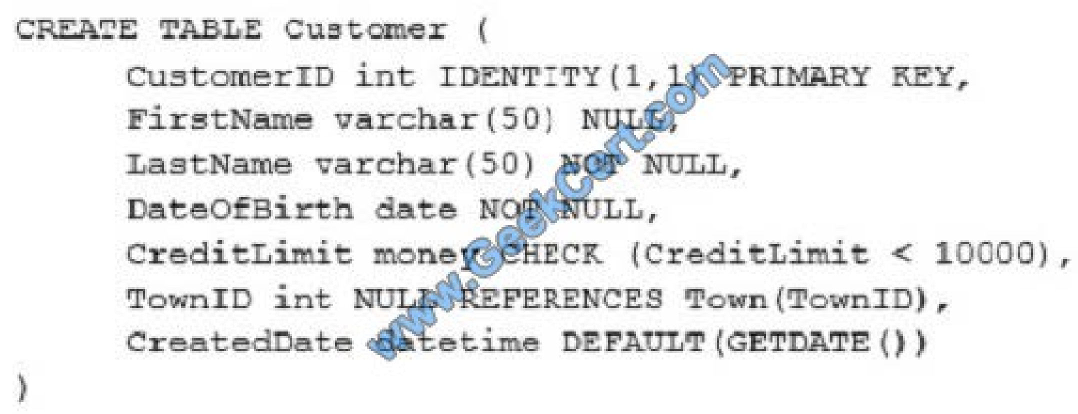
References: https://technet.microsoft.com/en-us/library/ms190690(v=sql.105).aspx https://msdn.microsoft.com/en-us/library/hh213574.aspx

QUESTION 2

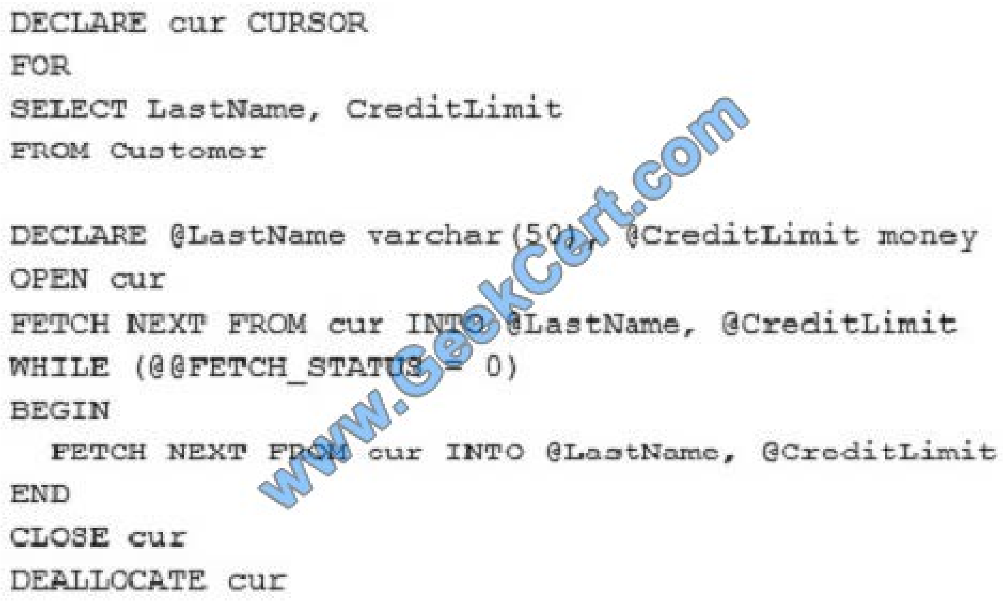
Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You create a table named Customer by running the following Transact-SQL statement: You create a cursor by running the following Transact-SQL statement:



```
CREATE TABLE Customer (
      CustomerID int IDENTITY(1,1) PRIMARY KEY,
      FirstName varchar(50) NULL,
      LastName varchar(50) NOT NULL,
      DateOfBirth date NOT NULL,
      CreditLimit money CHECK (CreditLimit < 10000),
      TownID int NULL REFERENCES Town(TownID),
      CreatedDate datetime DEFAULT(GETDATE())
)
```



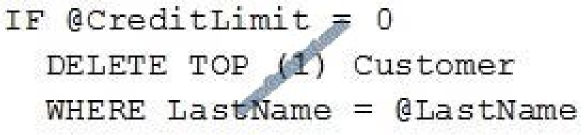
```
DECLARE cur CURSOR
FOR
SELECT LastName, CreditLimit
FROM Customer

DECLARE @LastName varchar(50), @CreditLimit money
OPEN cur
FETCH NEXT FROM cur INTO @LastName, @CreditLimit
WHILE (@@FETCH_STATUS = 0)
BEGIN
  FETCH NEXT FROM cur INTO @LastName, @CreditLimit
END
CLOSE cur
DEALLOCATE cur
```

If the credit limit is zero, you must delete the customer record while fetching data.

You need to add the DELETE statement.

Solution: You add the following Transact-SQL statement:



```
IF @CreditLimit = 0
  DELETE TOP (1) Customer
  WHERE LastName = @LastName
```

Does the solution meet the goal?

A. Yes

B. No

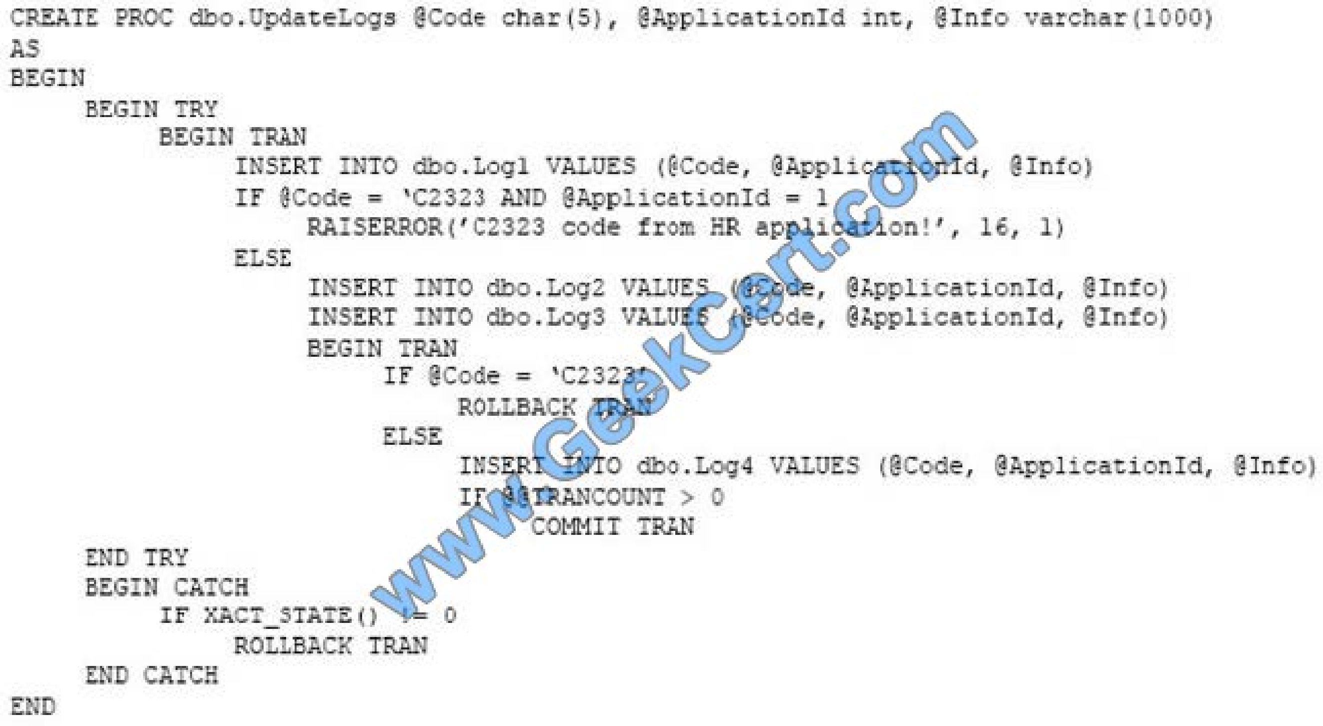
Correct Answer: B

Use a WHERE CURRENT OF clause, which deletes at the current position of the specified cursor. References: https://docs.microsoft.com/en-us/sql/t-sql/statements/delete-transact-sql

QUESTION 3

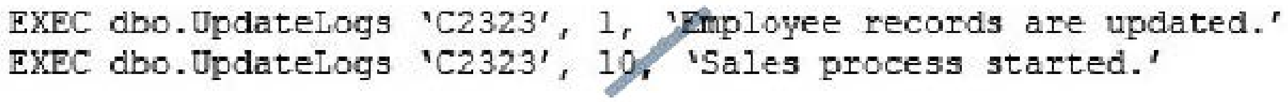
HOTSPOT

You have the following stored procedure:



```
CREATE PROC dbo.UpdateLogs @Code char(5), @ApplicationId int, @Info varchar(1000)
AS
BEGIN
     BEGIN TRY
          BEGIN TRAN
               INSERT INTO dbo.Log1 VALUES (@Code, @ApplicationId, @Info)
               IF @Code = 'C2323 AND @ApplicationId = 1
                    RAISERROR('C2323 code from HR application!', 16, 1)
               ELSE
                    INSERT INTO dbo.Log2 VALUES (@Code, @ApplicationId, @Info)
                    INSERT INTO dbo.Log3 VALUES (@Code, @ApplicationId, @Info)
                    BEGIN TRAN
                         IF @Code = 'C2323'
                              ROLLBACK TRAN
                         ELSE
                              INSERT INTO dbo.Log4 VALUES (@Code, @ApplicationId, @Info)
                              IF @@TRANCOUNT > 0
                                   COMMIT TRAN
     END TRY
     BEGIN CATCH
          IF XACT_STATE() != 0
               ROLLBACK TRAN
     END CATCH
END
```

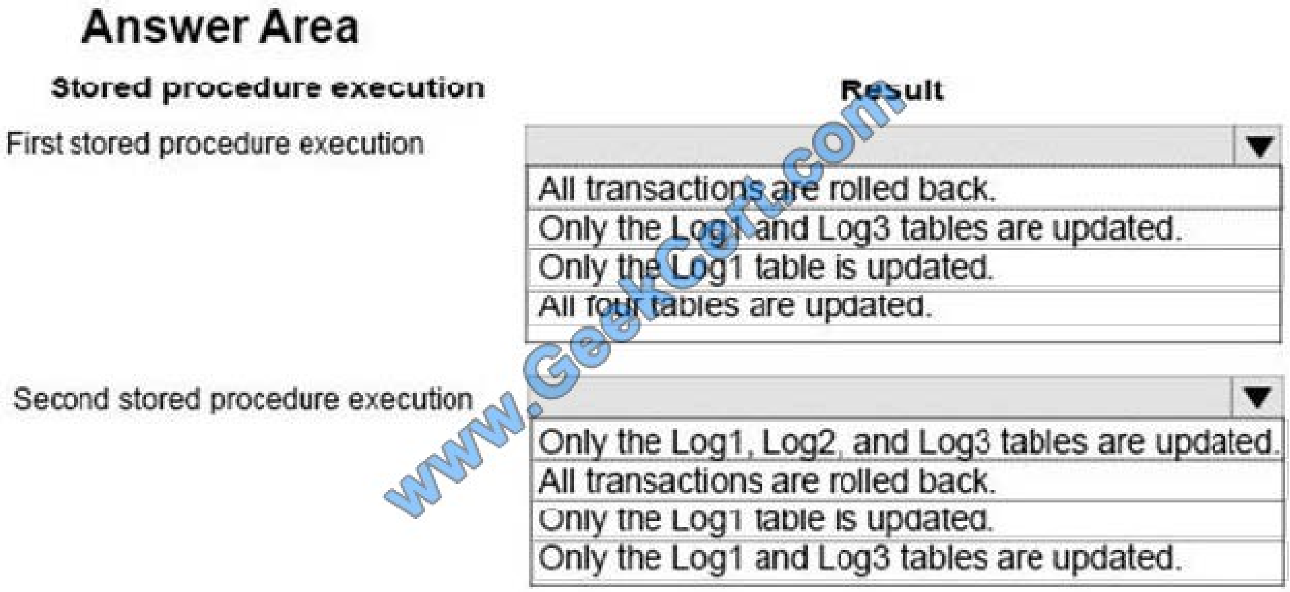
You run the following Transact-SQL statements:



```
EXEC dbo.UpdateLogs 'C2323', 1, 'Employee records are updated.'
EXEC dbo.UpdateLogs 'C2323', 10, 'Sales process started.'
```

What is the result of each Transact-SQL statement? To answer, select the appropriate options in the answer area.

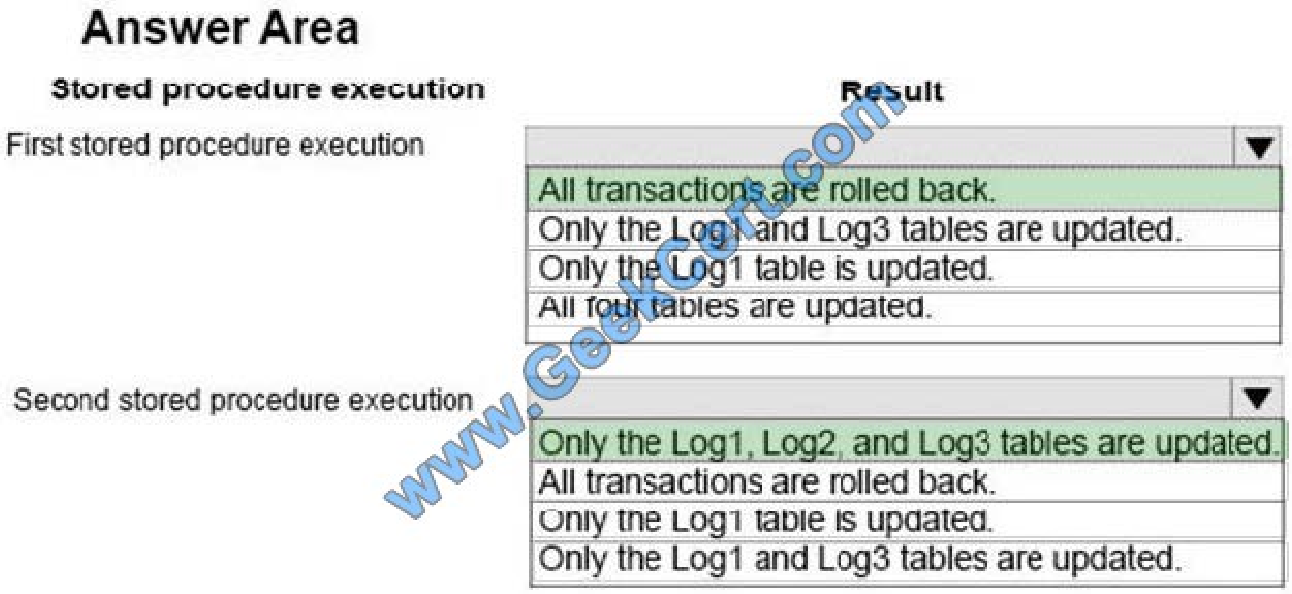
Hot Area:



Answer Area

Stored procedure execution	Result
First stored procedure execution	All transactions are rolled back. Only the Log1 and Log3 tables are updated. Only the Log1 table is updated. All four tables are updated.
Second stored procedure execution	Only the Log1, Log2, and Log3 tables are updated. All transactions are rolled back. Only the Log1 table is updated. Only the Log1 and Log3 tables are updated.

Correct Answer:



Answer Area

Stored procedure execution	Result
First stored procedure execution	All transactions are rolled back. Only the Log1 and Log3 tables are updated. Only the Log1 table is updated. All four tables are updated.
Second stored procedure execution	Only the Log1, Log2, and Log3 tables are updated. All transactions are rolled back. Only the Log1 table is updated. Only the Log1 and Log3 tables are updated.

Box 1: All transactions are rolled back.

The first IF-statement, IF @CODE = \\'C2323\\' AND @ApplicationID = 1, will be true, an error will be raised, the error will be caught in the CATCH block, and the only transaction that has been started will be rolled back.

Box 2: Only Log1, Log2, and Log3 tables are updated.

The second IF-statement, IF @Code = \\'C2323\\', will be true, so the second transaction will be rolled back, but log1, log2, and log3 was updated before the second transaction.

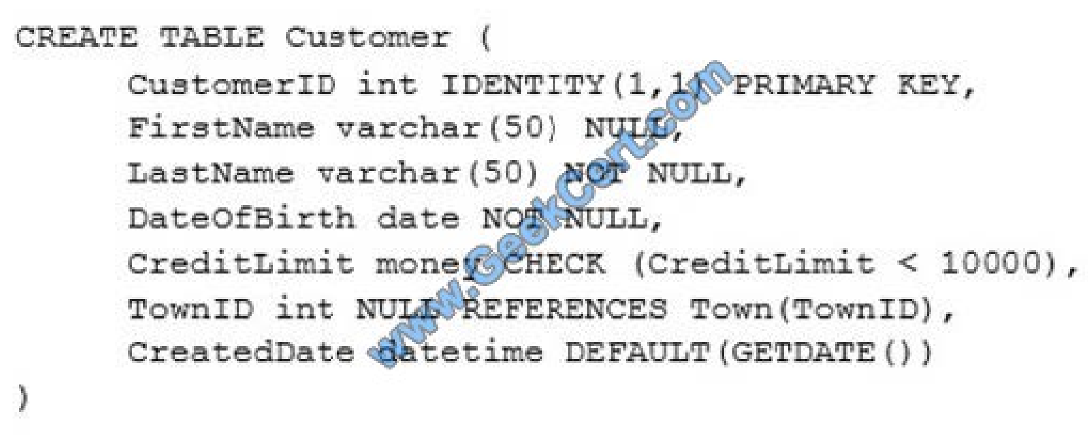
QUESTION 4

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while

others might not have a correct solution.

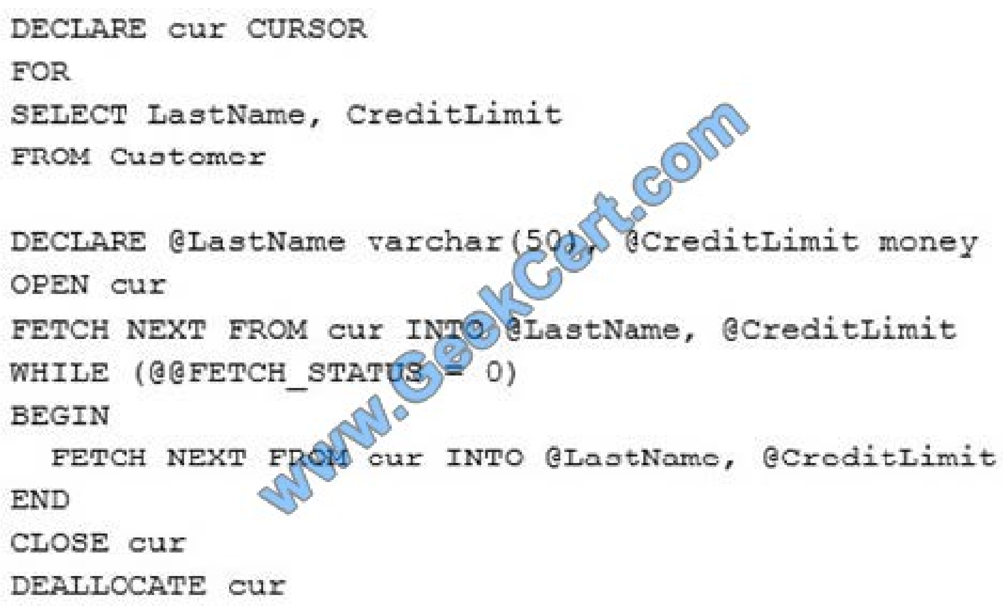
After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You create a table named Customer by running the following Transact-SQL statement:



```
CREATE TABLE Customer (
      CustomerID int IDENTITY(1,1) PRIMARY KEY,
      FirstName varchar(50) NULL,
      LastName varchar(50) NOT NULL,
      DateOfBirth date NOT NULL,
      CreditLimit money CHECK (CreditLimit < 10000),
      TownID int NULL REFERENCES Town(TownID),
      CreatedDate datetime DEFAULT(GETDATE())
)
```

You create a cursor by running the following Transact-SQL statement:

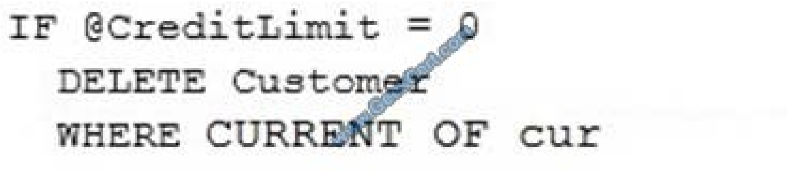


```
DECLARE cur CURSOR
FOR
SELECT LastName, CreditLimit
FROM Customer

DECLARE @LastName varchar(50), @CreditLimit money
OPEN cur
FETCH NEXT FROM cur INTO @LastName, @CreditLimit
WHILE (@@FETCH_STATUS = 0)
BEGIN
  FETCH NEXT FROM cur INTO @LastName, @CreditLimit
END
CLOSE cur
DEALLOCATE cur
```

If the credit limit is zero, you must delete the customer record while fetching data. You need to add the DELETE statement.

Solution: You add the following Transact-SQL statement:



```
IF @CreditLimit = 0
  DELETE Customer
  WHERE CURRENT OF cur
```

Does the solution meet the goal?

A. Yes

B. No

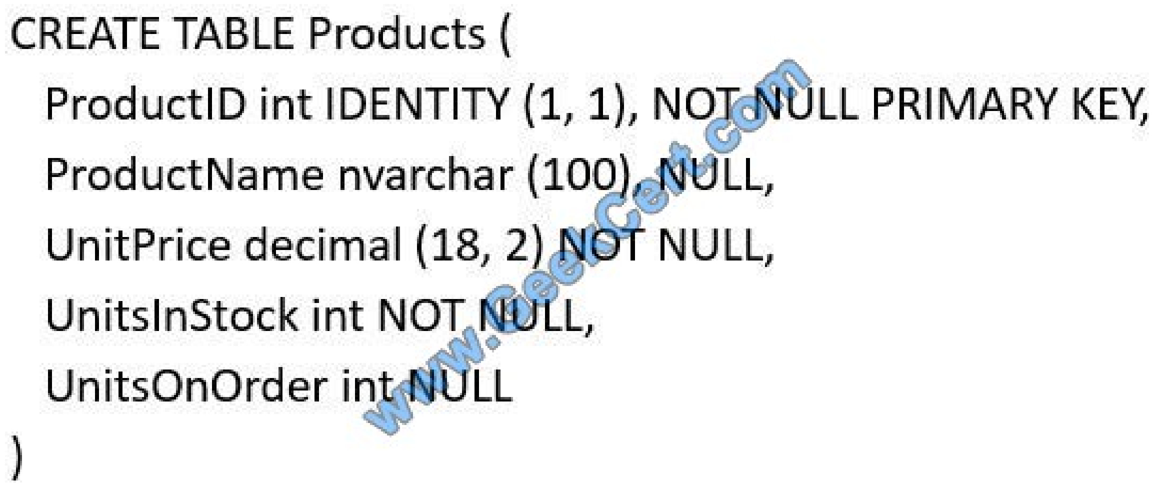
Correct Answer: B

QUESTION 5

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

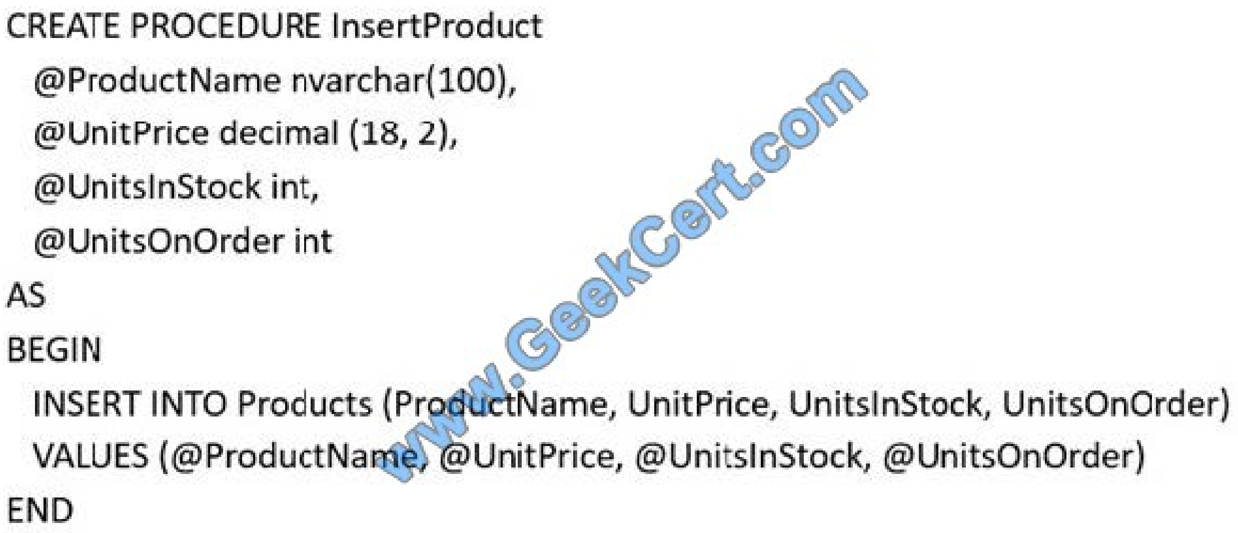
After you answer a question in this section. You will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You create a table named Products by running the following Transact-SQL statement:



```
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
)
```

You have the following stored procedure:



```
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
```

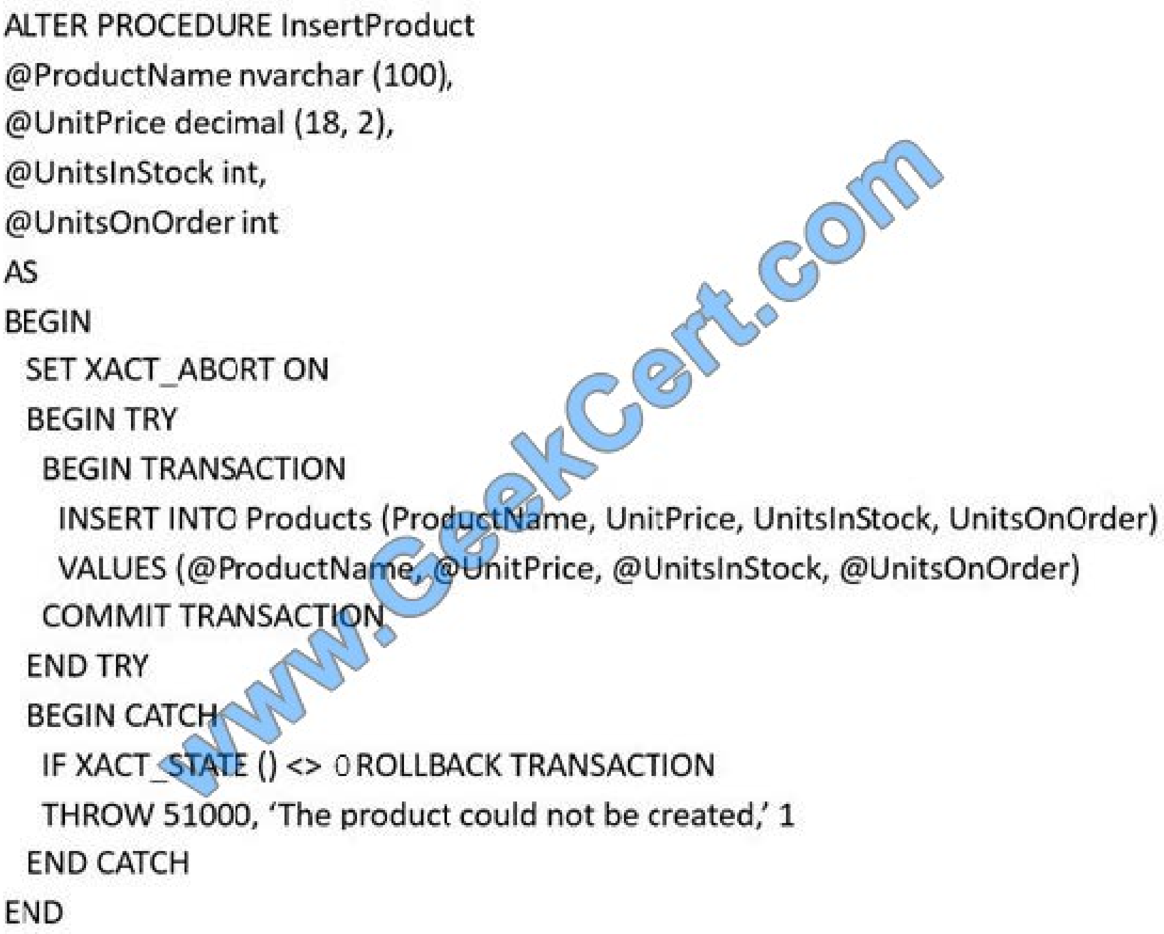
You need to modify the stored procedure to meet the following new requirements:

Insert product records as a single unit of work.

Return error number 51000 when a product fails to insert into the database.

If a product record insert operation fails, the product information must not be permanently written to the database.

Solution: You run the following Transact-SQL statement:



```
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, UnitPrice, UnitsInStock, UnitsOnOrder)
      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
```

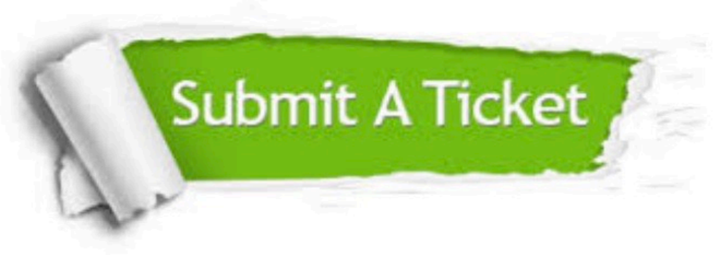
Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

[70-761 PDF Dumps](#) [70-761 VCE Dumps](#) [70-761 Practice Test](#)


Submit A Ticket