DUMPSBOSS.

Querying Data with Transact-SQL

Microsoft 70-761

Version Demo

Total Demo Questions: 14

Total Premium Questions: 207

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

QUESTION NO: 1 - (SIMULATION)

SIMULATION

You have a database that includes the following tables. All of the tables are in the Production schema.

You need to create a query that returns a list of product names for all products in the Beverages category.

Construct the query using the following guidelines:

- Use the first letter of the table name as the table alias.
- Use two-part column names.
- Do not surround object names with square brackets.
- Do not use implicit joins.
- Do not use variables.
- Use single quotes to surround literal values.

Part of the correct Transact-SQL has been provided in the answer area below. Enter the code in the answer area that resolves the problem and meets the stated goals or requirements. You can add code within the code that has been provided as well as below it.

Keywords

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

```
1 SELECT p.productname
2 FROM Production.Categories AS c
3
4 WHERE c.categoryname = 'Beverages'
```

ANSWER: See the explanation below.

Explanation:

1. SELECT p.productname

2. FROM Production.Categories As c

3. JOIN Production.Products As p

ON (c.categoryid = p.categoryid)

4. WHERE c.categoryname = 'Beverages'

References:

https://docs.microsoft.com/en-us/sql/relational-databases/performance/joins?view=sql-server-2017

QUESTION NO: 2 - (SIMULATION)

SIMULATION

You have a table named Cities that has the following two columns: CityID and CityName. The CityID column uses the int data type, and CityName uses nvarchar(max).

You have a table named RawSurvey. Each row includes an identifier for a question and the number of persons that responded to that question from each of four cities. The table contains the following representative data:

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

A reporting table named SurveyReport has the following columns: CityID, QuestionID, and RawCount, where RawCount is the value from the RawSurvey table.

You need to write a Transact-SQL query to meet the following requirements:

- Retrieve data from the RawSurvey table in the format of the SurveyReport table.
- The CityID must contain the CityID of the city that was surveyed.
- The order of cities in all SELECT queries must match the order in the RawSurvey table. - The order of cities in all IN statements must match the order in the RawSurvey table.

Construct the query using the following guidelines:

- Use one-part names to reference tables and columns, except where not possible.
- ALL SELECT statements must specify columns.
- Do not use column or table aliases, except those provided. - Do not surround object names with square brackets.

Keywords

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

Part of the correct Transact-SQL has been provided in the answer area below. Enter the code in the answer area that resolves the problem and meets the stated goals or requirements. You can add code within the code that has been provided as well as below it.

```
1 SELECT CityID, QuestionID, RawCount
2 AS t1
3 AS t2
4 JOIN
```

Use the Check Syntax button to verify your work. Any syntax or spelling errors will be reported by line and character position.

ANSWER: Please see explanation

Explanation:

1 Select CityID, QuestionID, RawCount

2 FROM (SELECT QuestionId, Tokyo, Boston, London, NewYork FROM RawSurvey) AS t1

3 UNPIVOT (RawCount FOR CityName IN (Tokyo, Boston, London, NewYork)) AS t2 4 JOIN Cities ON t2.CityName = t1.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

QUESTION NO: 3

Note: This question is part of a series of questions that use the same or similar answer choices. An answer choice may be correct for more than one question in the series. Each question is independent of the other questions in this series. Information and details provided in a question apply only to that question.

You have a database that contains tables named Customer_CRMSystem and Customer_HRSystem. Both tables use the following structure:

Column name	Data type	Allow null
CustomerID	int	No
CustomerCode	char(4)	Yes
CustomerName	varchar(50)	No

The tables include the following records:

Customer_CRMSystem

CustomerID	CustomerCode	CustomerName
1	CUS1	Roya
2	CUS9	Almudena
3	CUS4	Jack
4	NULL	Jane
5	NULL	Francisco

Customer_HRSystem

```
SELECT c.CustomerCode, c.CustomerName
FROM Customer_CRMSystem c
INNER JOIN Customer_HRSystem h
ON c.CustomerCode = h.CustomerCode AND c.CustomerName = h.CustomerName
```

Records that contain null values for CustomerCode can be uniquely identified by CustomerName. You need to display a Cartesian product, combining both tables.

Which Transact-SQL statement should you run?

A.
```
SELECT c.CustomerCode, c.CustomerName
FROM Customer_CRMSystem c
INNER JOIN Customer_HRSystem h
ON c.CustomerCode = h.CustomerCode AND c.CustomerName = h.CustomerName
```

B.
```
SELECT CustomerCode, CustomerName
FROM Customer_CRMSystem
INTERSECT
SELECT CustomerCode, CustomerName
FROM Customer_HRSystem
```

C.
```
SELECT c.CustomerCode, c.CustomerName
FROM Customer_CRMSystem c
LEFT OUTER JOIN Customer_HRSystem h
ON c.CustomerCode = h.CustomerCode
WHERE h.CustomerCode IS NULL AND c.CustomerCode IS NOT NULL
```

D.
```
SELECT CustomerCode, CustomerName
FROM Customer_CRMSystem
EXCEPT
SELECT CustomerCode, CustomerName
FROM Customer_HRSystem
```

E.
```
SELECT CustomerCode, CustomerName
FROM Customer_CRMSystem
UNION
SELECT CustomerCode, CustomerName
FROM Customer_HRSystem
```

F.
```
SELECT CustomerCode, CustomerName
FROM Customer_CRMSystem
UNION ALL
SELECT CustomerCode, CustomerName
FROM Customer_HRSystem
```

G.
```
SELECT c.CustomerCode, c.CustomerName, h.CustomerCode, h.CustomerName
FROM Customer_CRMSystem c
CROSS JOIN Customer_HRSystem h
```

H.
```
SELECT c.CustomerCode, c.CustomerName, h.CustomerCode, h.CustomerName
FROM Customer_CRMSystem c
FULL OUTER JOIN Customer_HRSystem h
ON c.CustomerCode = h.CustomerCode AND c.CustomerName = h.CustomerName
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

F. Option F

G. Option G

H. Option H

ANSWER: G

Explanation:

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.

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

QUESTION NO: 4 - (DRAG DROP)

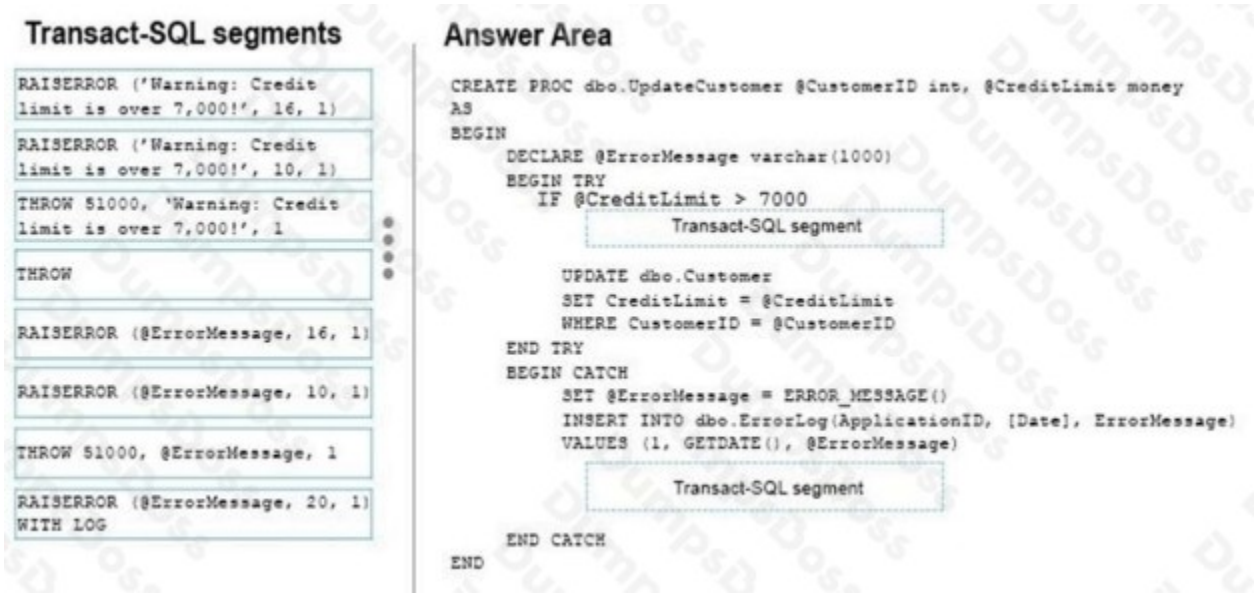
DRAG DROP

You need to create a stored procedure that meets the following requirements:

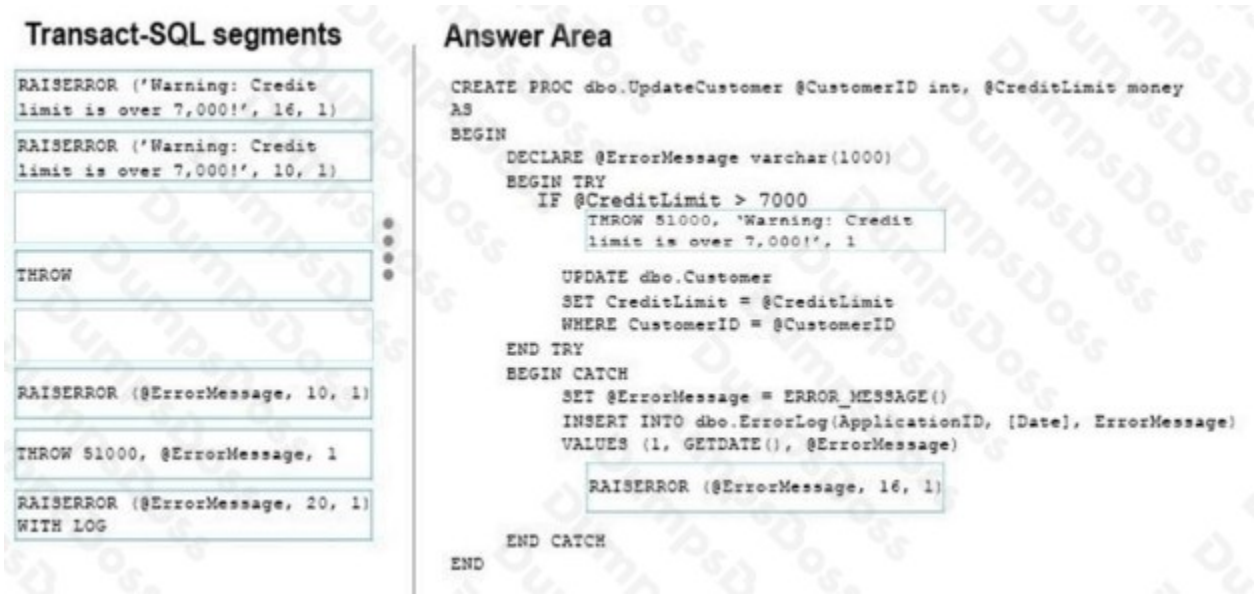
- Produces a warning if the credit limit parameter is greater than 7,000
- Propagates all unexpected errors to the calling process

How should you complete the Transact-SQL statement? To answer, drag the appropriate Transact-SQL segments to the correct locations. Each Transact-SQL segments may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Select and Place:



ANSWER:



Explanation:

Box 1: THROW 51000, 'Warning: Credit limit is over 7,000!",1

THROW raises an exception and transfers execution to a CATCH block of a TRY…CATCH construct in SQL Server.

THROW syntax:

THROW [{ error_number | @local_variable },

{ message | @local_variable },

{ state | @local_variable }]

[;]

Box2: RAISERROR (@ErrorMessage, 16,1)

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.

Severity levels from 0 through 18 can be specified by any user. Severity levels from 19through 25 can only be specified by members of the sysadmin fixed server role or users with ALTER TRACE permissions. For severity levels from 19 through 25, the WITH LOG option is required.

On Severity level 16. Using THROW to raise an exception

The following example shows how to use the THROW statement to raise an exception.

Transact-SQL

THROW 51000, 'The record does not exist.', 1;

Here is the result set.

Msg 51000, Level 16, State 1, Line 1 The record does not exist.

Note: RAISERROR syntax:

RAISERROR({ msg_id | msg_str | @local_variable }

{ ,severity ,state }

[,argument [,...n]])

[WITH option [,...n]]

Note: The ERROR_MESSAGE function returns the message text of the error that caused the CATCH block of a TRY…CATCH construct to be run.

References:

https://msdn.microsoft.com/en-us/library/ms178592.aspx https://msdn.microsoft.com/en-us/library/ms190358.aspx https://msdn.microsoft.com/en-us/library/ee677615.aspx

QUESTION NO: 5 - (HOTSPOT)

HOTSPOT

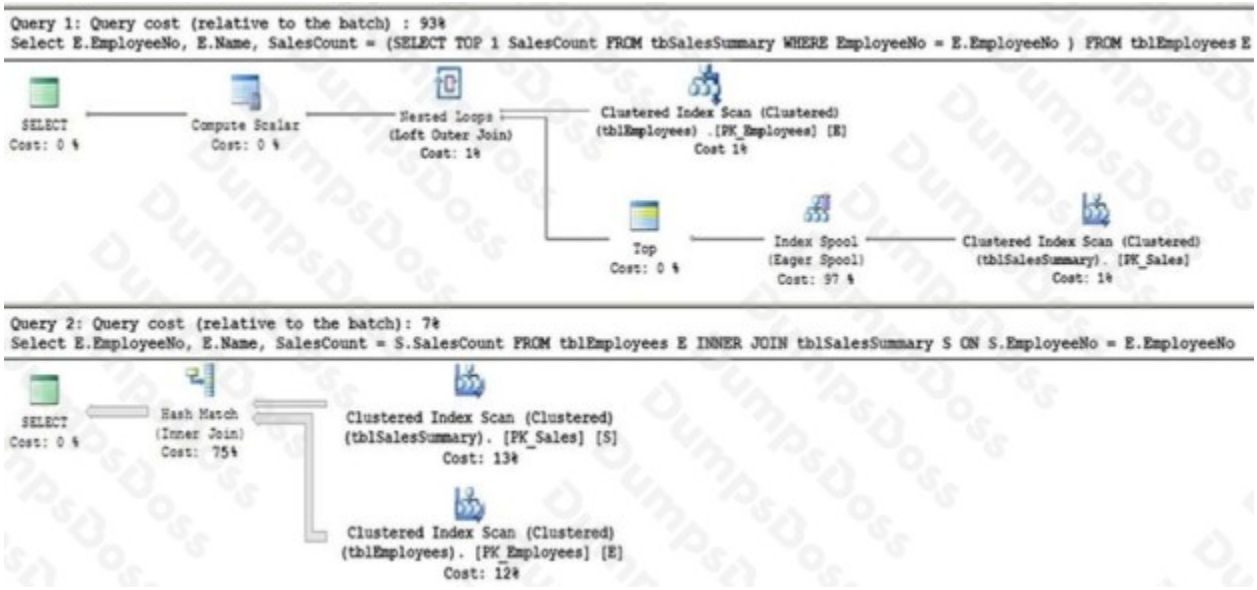
You have a database that contains the following tables: tblEmployees and tblSalesSummary. Each record contains approximately one million records.

You use Microsoft SQL Server Management Studio (SSMS) to run two queries. The Include Actual Execution Plan option is enabled. Both queries return the same results. SSMS generates the execution plans shown in the exhibit. (Click the Exhibit button.)

You need to troubleshoot the queries.

How should you interpret the execution plans? To answer, select the appropriate options in the answer area.

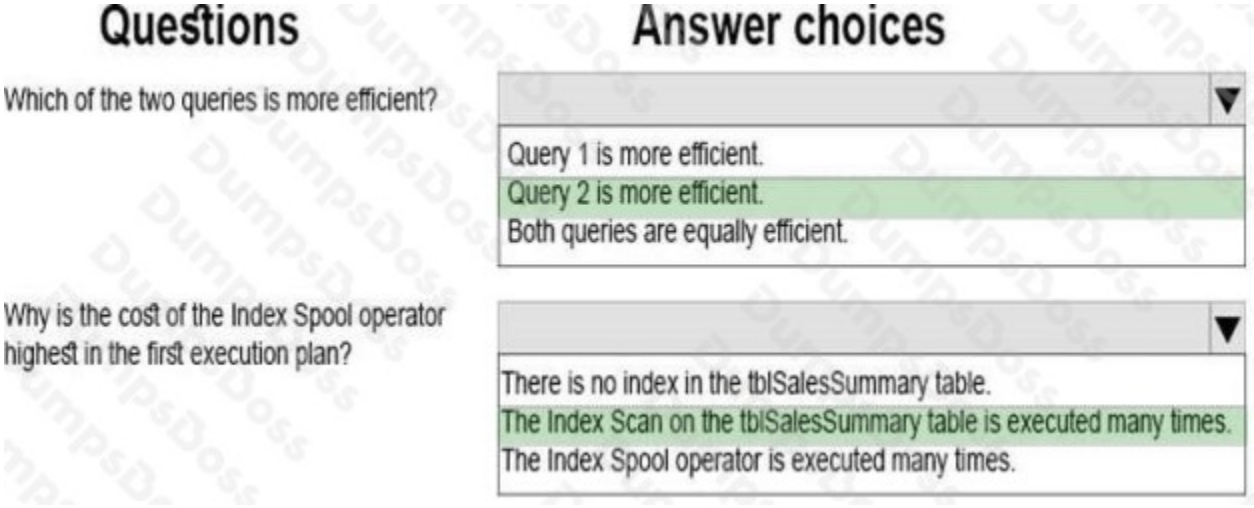
NOTE: Each correct selection is worth one point.



Hot Area:

Questions	Answer choices
Which of the two queries is more efficient?	Query 1 is more efficient. Query 2 is more efficient. Both queries are equally efficient.
Why is the cost of the Index Spool operator highest in the first execution plan?	There is no index in the tblSalesSummary table. The Index Scan on the tblSalesSummary table is executed many times. The Index Spool operator is executed many times.

ANSWER:



Explanation:

References: https://docs.microsoft.com/en-us/sql/relational-databases/showplan-logical-and-physical-operators-reference?view=sql-server-2017

QUESTION NO: 6

You need to create an indexed view that requires logic statements to manipulate the data that the view displays.

Which two database objects should you use? Each correct answer presents a complete solution.

A. a user-defined table-valued function

B. a CRL function

C. a stored procedure

D. a user-defined scalar function

ANSWER: B D

Explanation:

You can create a database object inside an instance of SQL Server that is programmed in an assembly created in the Microsoft .NET Framework common language runtime (CLR).

Incorrect Answers:

A: A table valued function cannot be called from indexed view C: The Stored procedure cannot be called inside of a View.

References: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql

QUESTION NO: 7 - (DRAG DROP)

DRAG DROP

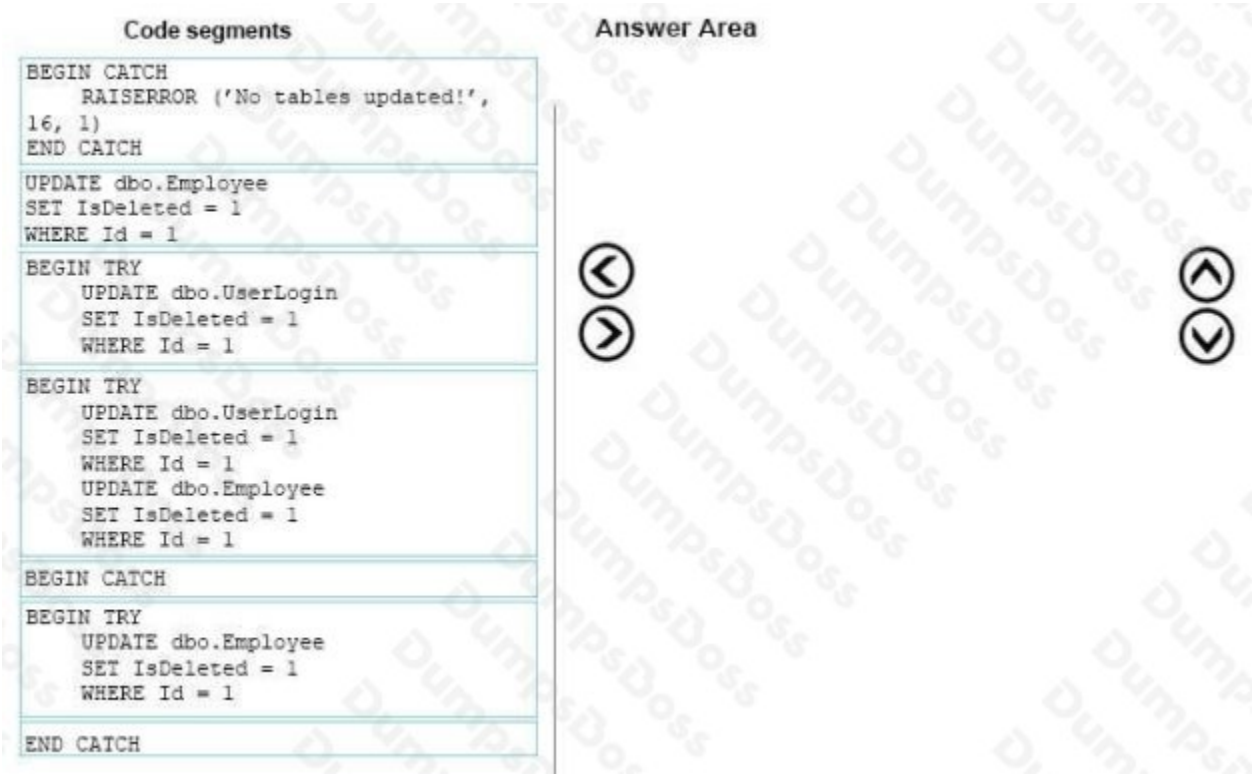
You have two tables named UserLogin and Employee respectively.

You need to create a Transact-SQL script that meets the following requirements:

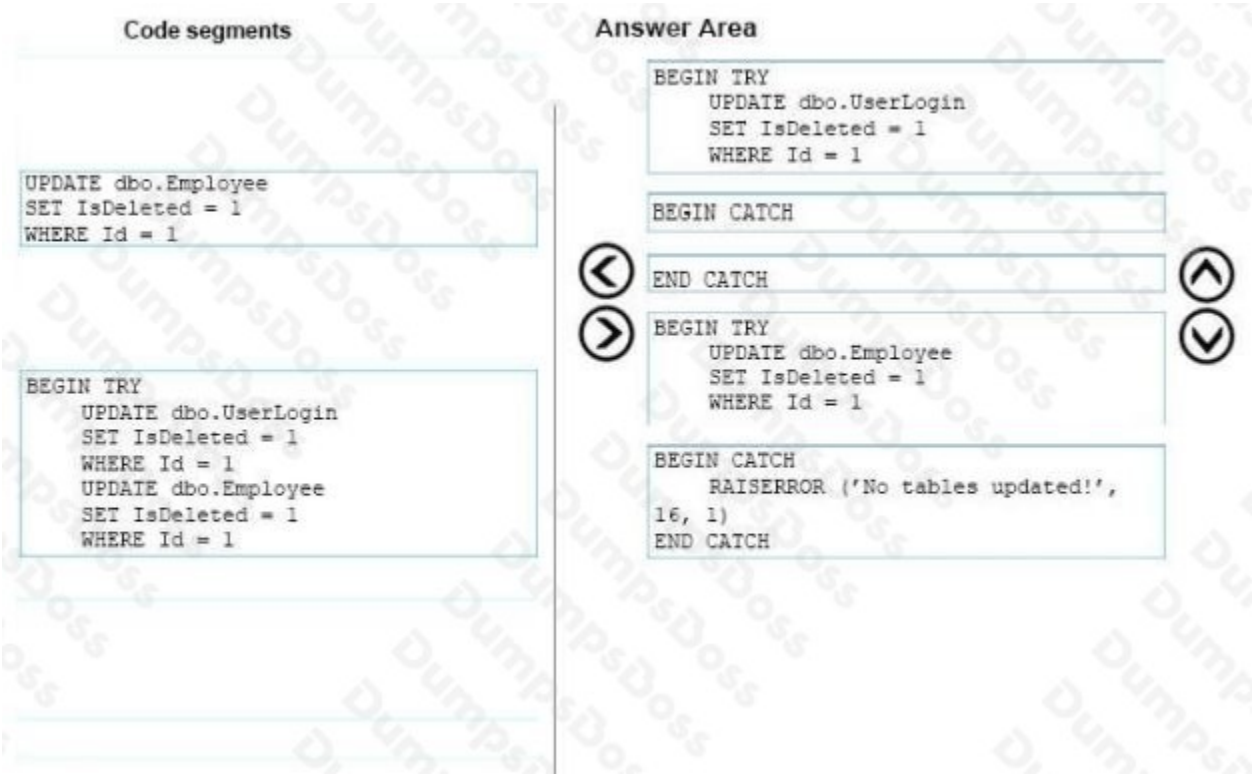
- The script must update the value of the IsDeleted column for the UserLogin table to 1 if the value of the Id column for the UserLogin table is equal to1.
- The script must update the value of the IsDeleted column of the Employee table to 1 if the value of the Id column is equal to 1 for the Employee table when an update to the UserLogin table throws an error.
- The error message “No tables updated!” must be produced when an update to the Employee table throws an error.

Which five Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

Select and Place:



ANSWER:



Explanation:

A TRY block must be immediately followed by an associated CATCH block. Including any other statements between the END TRY and BEGIN CATCH statements generates a syntax error.

Note: END TRY statements are missing in the options, which makes this question confusing.

References: https://msdn.microsoft.com/en-us/library/ms175976.aspx

QUESTION NO: 8

You have a date related query that would benefit from an indexed view.

You need to create the indexed view.

Which two Transact-SQL functions can you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. DATEADD

B. AT TIME ZONE

C. GETUTCDATE

D. DATEDIFF

ANSWER: A D

Explanation:

An indexed view will accept only deterministic functions.

Incorrect Answers:

C: GETUTCDATE is not deterministic.

References:

https://docs.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-2017#DateandTimeFunctions

QUESTION NO: 9 - (HOTSPOT)

HOTSPOT

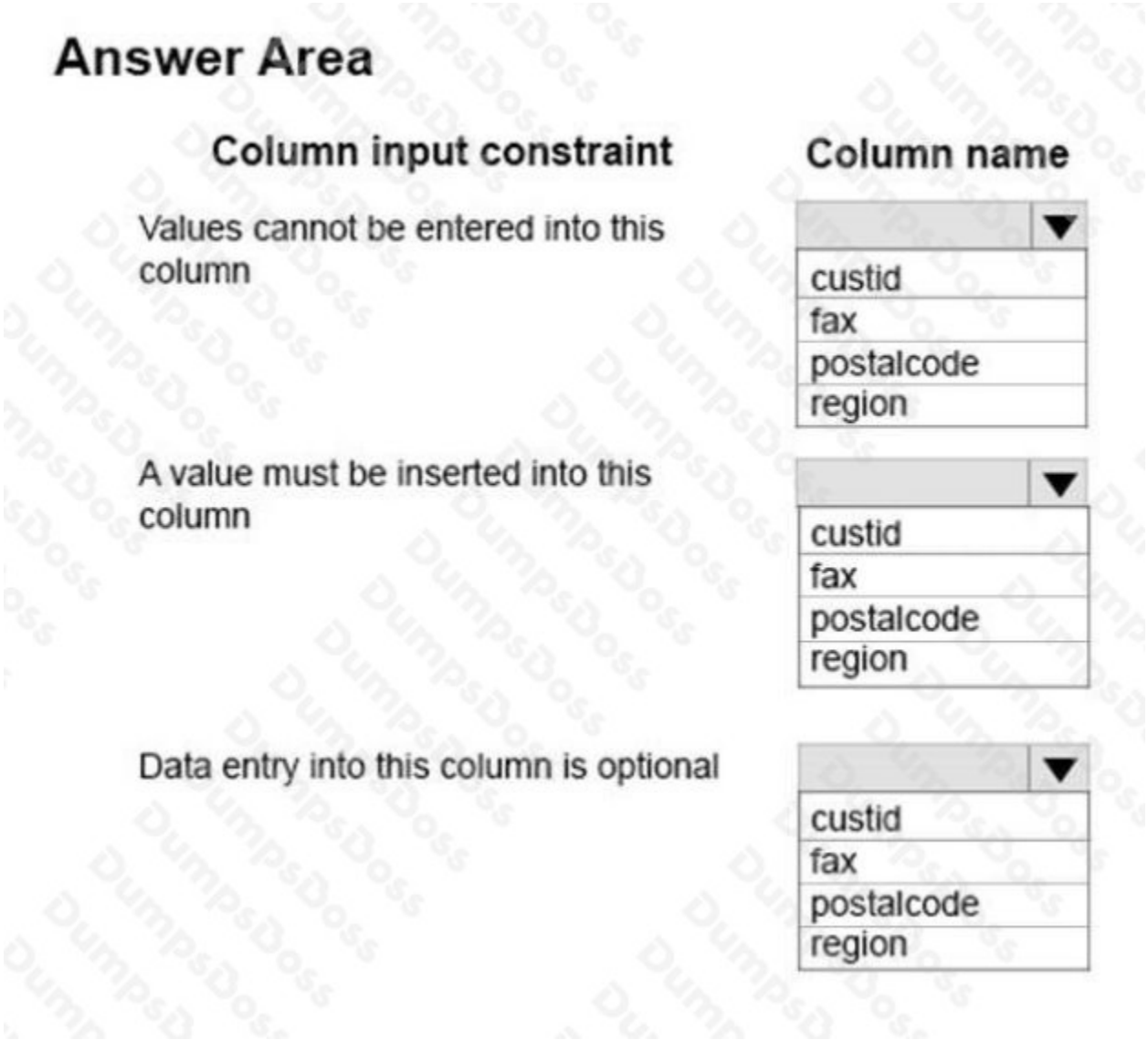
You run the following Transact-SQL statement:

```
CREATE TABLE Sales.Customers(
    custid int IDENTITY(1,1) NOT NULL,
    companyname nvarchar(50) NULL,
    contacttitle nvarchar(30) NOT NULL,
    address nvarchar(60) NOT NULL,
    postalcode nvarchar(10) NOT NULL,
    region nvarchar(15) NULL,
    phone nvarchar(24) NOT NULL,
    fax nvarchar(24) NULL,
) ON PPRIMARY
```

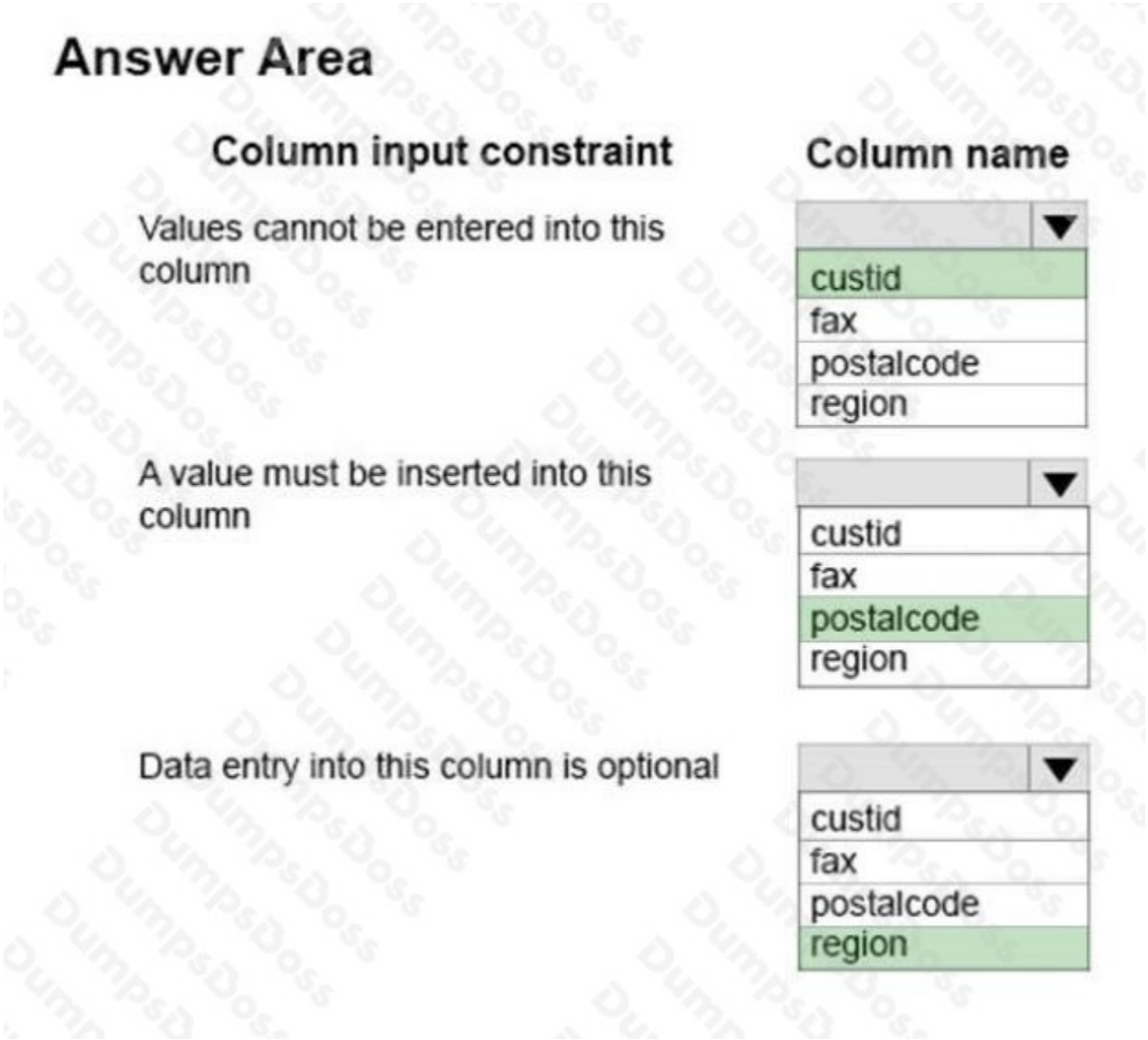
You need to ensure that you can insert data into the table.

What are the characteristics of the data? To answer, select the appropriate options in the answer area.

Hot Area:


Answer Area
Column input constraint
Column name
Values cannot be entered into this column
custid
fax
postalcode
region
A value must be inserted into this column
custid
fax
postalcode
region
Data entry into this column is optional
custid
fax
postalcode
region

ANSWER:



Explanation:

Box 1: custid

IDENTITY indicates that the new column is an identity column. When a new row is added to the table, the Database Engine provides a unique, incremental value for the column. Identity columns are typically used with PRIMARY KEY constraints to serve as the unique row identifier for the table.

Box2: postalcode postalcode is declared as NOT NULL, which means that a value must be inserted.

Box 3: region

Fax is also a correct answer. Both these two columns are declared as NULL, which means that data entry is optional.
References: https://msdn.microsoft.com/en-us/library/ms174979.aspx

QUESTION NO: 10

Note: This question is part of a series of questions that use the same or similar answer choices. An answer choice may be correct for more than one question in the series. Each question is independent of the other questions in this series. Information and details provided in a question apply only to that question.

You have a table named Products that contains information about the products that your company sells. The table contains many columns that do not always contain values.

You need to implement an ANSI standard method to convert the NULL values in the query output to the phrase “Not Applicable”.

What should you implement?

A. the COALESCE function

B. a view

C. a table-valued function

D. the TRY_PARSE function

E. a stored procedure

F. the ISNULL function

G. a scalar function

H. the TRY_CONVERT function

ANSWER: A

Explanation:

COALESCE evaluates the arguments in order and returns the current value of the first expression that initially doesn't evaluate to NULL.

Incorrect Answers:

F: ISNULL is not a ANSI standard function. The COALESCE function is preferred.

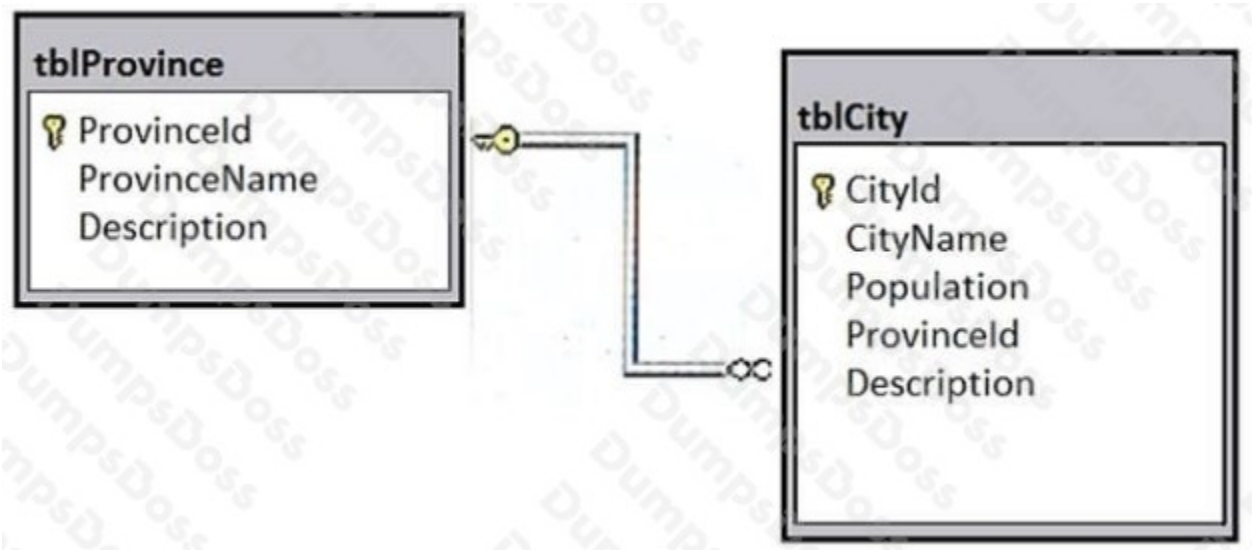
References: https://docs.microsoft.com/en-us/sql/t-sql/language-elements/coalesce-transact-sql?view=sql-server-2017

QUESTION NO: 11

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.

A database has two tables as shown in the following database diagram:



You need to list all provinces that have at least two large cities. A large city is defined as having a population of at least one million residents. The query must return the following columns:

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

Solution: You run the following Transact-SQL statement:

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

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

We should use CROSS APPLY rather than OUTER APPLY.

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

QUESTION NO: 12

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 have a database that tracks orders and deliveries for customers in North America. The database contains the following tables:

Sales.Customers

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

Application.Cities

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

Your company is developing a new social application that connects customers to each other based on the distance between their delivery locations.

You need to write a query that returns the nearest customer.

Solution: You run the following Transact-SQL statement:

```
SELECT TOP 1 B.CustomerID, A.DeliveryLocation.STDistance(B.DeliveryLocation) AS Dist
FROM Sales.Customers AS A
CROSS JOIN Sales.Customers AS B
WHERE A.CustomerID = @custID AND A.CustomerID <> B.CustomerID
ORDER BY Dist
```

The variable @custID is set to a valid customer.

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

Instead, use the ShortestLineTo (geometry Data Type) which returns a LineString instance with two points that represent the shortest distance between the two geometry instances. The length of the LineString instance returned is the distance between the two geometry instances.

Note: STDistance returns the shortest distance between a point in a geography instance and a point in another geography instance.

References:

https://docs.microsoft.com/en-us/sql/t-sql/spatial-geometry/shortestlineto-geometry-data-type https://docs.microsoft.com/en-us/sql/t-sql/spatial-geography/stdistance-geography-data-type

QUESTION NO: 13

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 have a database that tracks orders and deliveries for customers in North America. The database contains the following tables:

Sales.Customers

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

Application.Cities

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

The company's development team is designing a customer directory application. The application must list customers by the area code of their phone number. The area code is defined as the first three characters of the phone number.

The main page of the application will be based on an indexed view that contains the area and phone number for all customers.

You need to return the area code from the PhoneNumber field.

Solution: You run the following Transact-SQL statement:

```
CREATE FUNCTION AreaCode (
     @phoneNumber nvarchar(20)
)
RETURNS nvarchar(10)
WITH SCHEMABINDING
AS
BEGIN
     DECLARE @areaCode nvarchar(max)
     SELECT @areaCode = value FROM STRING_SPLIT(@phoneNumber, '-')
     RETURN @areaCode
END
```

Does the solution meet the goal?

A. Yes

B. No

ANSWER: B

Explanation:

We need SELECT TOP 1 @areacode =.. to ensure that only one value is returned.

QUESTION NO: 14

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 in 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 null values
StandardDiscountPercentage	int	does not allow null values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow null values, a value of 1 indicates that the account is on a credit hold
DeliveryLocation	geography	does not allow null values
PhoneNumber	nvarchar(20)	does not allow null values
ValidFrom	datetime2(7)	does not allow null values, GENERATED ALWAYS AS ROW START
ValidTo	datetime2(7)	does not allow null 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.Cities
WHERE (IsOnCreditHold = 0 AND DeliveryCityID = CityID)
OR (IsOnCreditHold = 1 AND PostalCityID = CityID)
```

B.
```
SELECT CustomerID, LatestRecordedPopulation
FROM Sales.Customers
INNER JOIN Application.Cities 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.Cities AS A ON A.CityID = DeliveryCityID
INNER JOIN Application.Cities 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.Cities AS A ON A.CityID = CityId
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: A B

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