70-761 Q&As

Querying Data with Transact-SQL

Pass Microsoft 70-761 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.passapply.com/70-761.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

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

```
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 ALWATS AS ROW END NOT NULL,
      PERIOD FOR SYSTEM_TIME(ValidFrom, ValidTo)
)
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = CustomersHistory))
```


You need to return normalized data for all customers that were added in the year 2014. Which Transact-SQL statement should you run?

```
SELECT FirstName, LastName, SUM(AnnualRevenue)
FROM Customers
GROUP BY GROUPING SETS((FirstName, LastName, AnnualRevenue),())
ORDER BY FirstName, LastName, AnnualRevenue
```

A.
```
SELECT FirstName, LastName, SUM(AnnualRevenue)
FROM Customers
GROUP BY GROUPING SETS((FirstName, LastName, AnnualRevenue),())
ORDER BY FirstName, LastName, 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
```

F.
```
SELECT c.CustomerID, c.FirstName, c.LastName, c.Address, c.ValidFrom, c.ValidTo
FROM Customers AS c ORDER BY c.CustomerID
FOR XML PATH ('CustomerData'), root ('Customers')
```

G.
```
SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers FOR SYSTEM_TIME
BETWEEN '2014-01-01 00:00:00.000000' AND '2015-01-01 00:00:00.000000'
```

H.
```
SELECT CustomerID, FirstName, LastName, TaxIdNumber, Address, ValidFrom, ValidTo
FROM Customers
WHERE DateCreated
BETWEEN '20140101' AND '20141231'
```

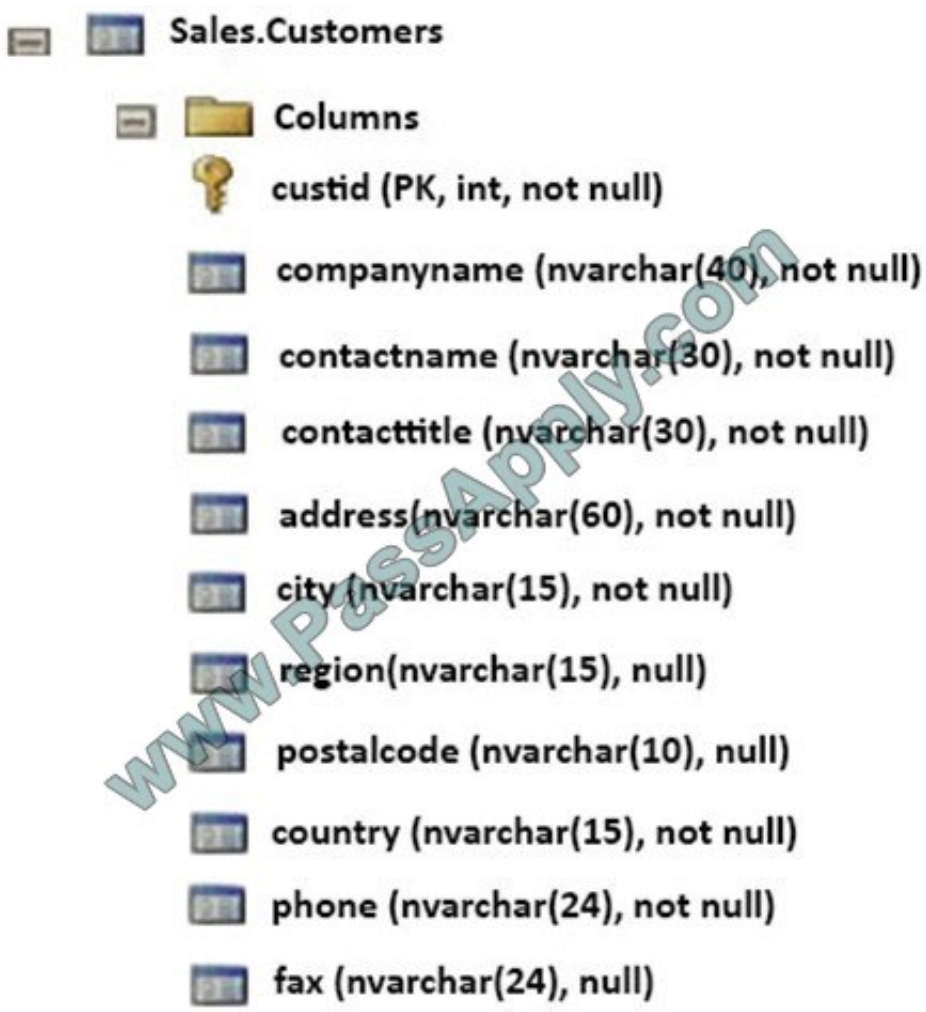
A. B. C. D. E. F. G. H.

Correct Answer: G

QUESTION 2

DRAG DROP

You need to create a stored procedure to update a table named Sales.Customers. The structure of the table is shown in the exhibit. (Click the exhibit button.)



The stored procedure must meet the following requirements:

Accept two input parameters.

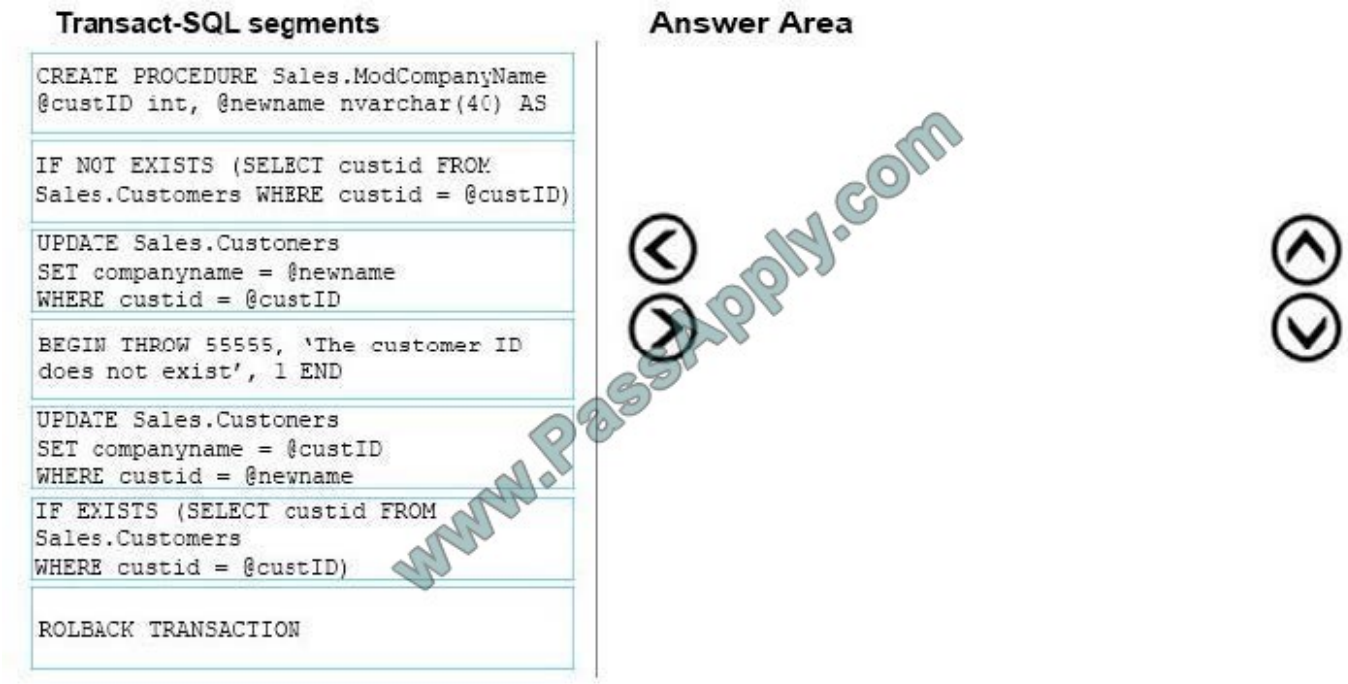
Update the company name if the customer exists.

Return a custom error message if the customer does not exist.

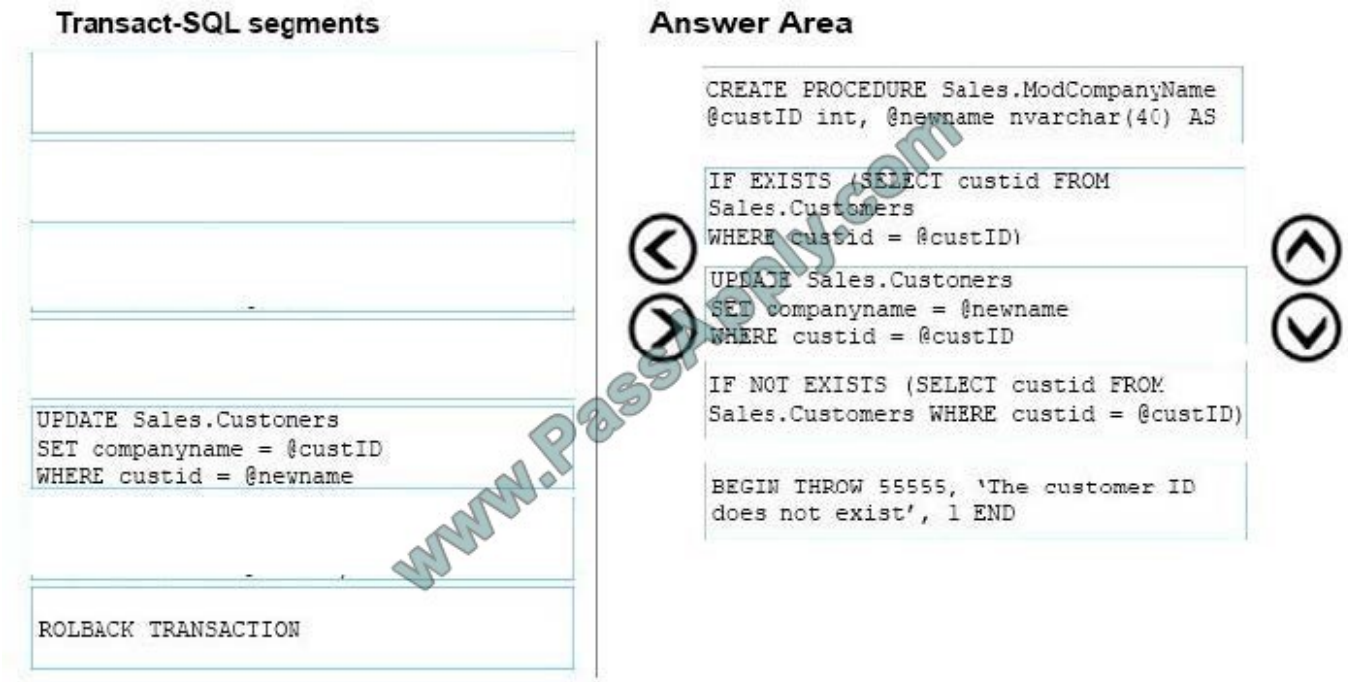
Which five Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Select and Place:



Correct Answer:



QUESTION 3

You have a database named DB1 that contains a temporal table named Sales.Customers.

You need to create a query that returns the credit limit that was available to each customer in DB1 at the beginning of

2017.

Which query should you execute?

```
SELECT
      CustomerID,
      CustomerName,
      CreditLimit
FROM
      Sales.Customers
            FOR SYSTEM_TIME CONTAINED IN ('2017-01-01 00:00:00');
```

A.

```
SELECT
      CustomerID,
      CustomerName,
      CreditLimit
FROM
      Sales.Customers
            FOR SYSTEM_TIME CONTAINED IN ('2017-01-01 00:00:00');
```

B.

```
SELECT
      CustomerID,
      CustomerName,
      CreditLimit
FROM
      Sales.Customers
            FOR SYSTEM_TIME CONTAINED IN ('2017-01-01');
```

C.

```
SELECT
      CustomerID,
      CustomerName,
      CreditLimit
FROM
      Sales.Customers
            FOR SYSTEM_TIME AS OF '2017-01-01';
```

D.

```
SELECT
      CustomerID,
      CustomerName,
      CreditLimit
FROM
      Sales.Customers
            FOR SYSTEM_TIME ALL;
```

A. B. C. D.

Correct Answer: C

AS OF: Returns a table with a rows containing the values that were actual (current) at the specified point in time in the past.

Incorrect Answers:

A, B: CONTAINED IN has two parameters: CONTAINED IN (,)

References: https://docs.microsoft.com/en-us/sql/relational-databases/tables/querying-data-in-a-system-versioned-temporal-table

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 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,ProductPrice,ProductsInStock,ProductsOnOrder)
     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
     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
```


Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

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:

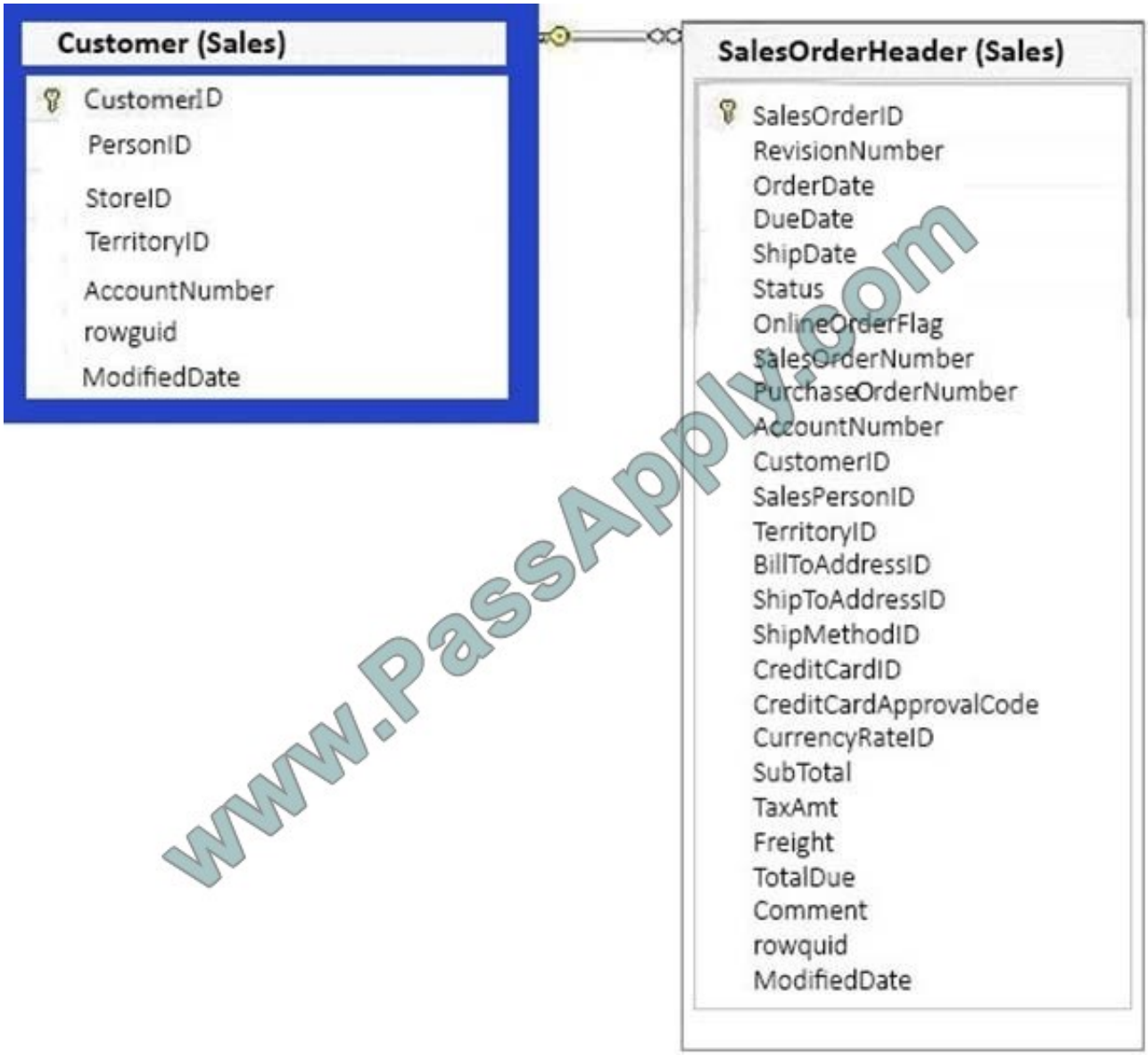
You can implement error handling for the INSERT statement by specifying the statement in a TRY...CATCH construct.

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 5

You have a database that includes the tables shown in the exhibit. (Click the exhibit button.)



You need to create a list of all customers and the date that the customer placed their last order. For customers who have not placed orders, you must substitute a zero for the order ID and 01/01/1990 for the date. Which Transact-SQL statement should you run?

```
SELECT C.CustomerID, COALESCE(MAX(OrderDate), '19000101')
FROM Sales.Customer C LEFT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID
ORDER BY C.CustomerID
```

A.
```
SELECT C.CustomerID, COALESCE(MAX(OrderDate), '19000101')
FROM Sales.Customer C LEFT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID
ORDER BY C.CustomerID
```

B.
```
SELECT C.CustomerID, MAX(OrderDate)
FROM Sales.Customer C LEFT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID
ORDER BY C.CustomerID
```

C.
```
SELECT C.CustomerID, MAX(OrderDate)
FROM Sales.Customer C CROSS JOIN Sales.SalesOrderHeader SOH
GROUP BY C.CustomerID
ORDER BY C.CustomerID
```

D.
```
SELECT C.CustomerID, SOH.SalesOrderID, MAX(OrderDate)
FROM Sales.Customer C LEFT OUTER JOIN Sales.SalesOrderHeader SOH
ON C.CustomerID = SOH.CustomerID
GROUP BY C.CustomerID, SOH.SalesOrderID
ORDER BY C.CustomerID
```

A. B. C. D.

Correct Answer: A

COALESCE evaluates the arguments in order and returns the current value of the first expression that initially does not evaluate to NULL. References: https://docs.microsoft.com/en-us/sql/t-sql/language-elements/coalesce-transact-sql

QUESTION 6

You have a database that tracks customer complaints.

The database contains a table named Complaints that includes the following columns:

Column name	Column description
ComplaintID	This is a unique identifier for a complaint record.
CustomerTranscript	This column stores a transcribed verbatim record of a customer complaint.

You need to create a query that lists complaints about defective products. The report must include complaints where the exact phrase "defective product" occurs, as well as complaints where similar phrases occur.

A.
```
SELECT ComplaintID, CustomerTranscript FROM Complaints
INNER JOIN FREETEXTABLE (Complaints, CustomerTranscript, 'defective product') AS Matches
ON Complaints.ComplaintID = Matches.[KEY]
```

B.
```
SELECT ComplaintID, CustomerTranscript FROM Complaints
INNER JOIN CONTAINSTABLE(Complaints, CustomerTranscript, '%defective% product%') AS Matches
ON Complaints.ComplaintID = Matches.[KEY]
```

C.
```
SELECT ComplaintID, CustomerTranscript, FROM Complaints
WHERE CONTAINS(CustomerTranscript, 'defective product')
```

D.
```
SELECT ComplaintID, ComplaintTranscript FROM Complaints
WHERE CONTAINS(CustomerTranscript,'defective') AND CONTAINS (CustomerTranscript, 'product')
```

A. B. C. D.

Correct Answer: D

References: https://docs.microsoft.com/en-us/sql/t-sql/queries/contains-transact-sql?view=sql-server-2017

QUESTION 7

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

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

Correct Answer: B

QUESTION 8

You are building a stored procedure that will update data in a table named Table1 by using a complex query as the data source.

You need to ensure that the SELECT statement in the stored procedure meets the following requirements:

Data being processed must be usable in several statements in the stored procedure.

Data being processed must contain statistics.

What should you do?

A. Update Table1 by using a common table expression (CTE).

B. Insert the data into a temporary table, and then update Table1 from the temporary table.

C. Place the SELECT statement in a derived table, and then update Table1 by using a JOIN to the derived table.

D. Insert the data into a table variable, and then update Table1 from the table variable.

Correct Answer: B

Temp Tables... Are real materialized tables that exist in tempdb Have dedicated stats generated by the engine Can be indexed Can have constraints Persist for the life of the current CONNECTION Can be referenced by other queries or subproce Incorrect Answers:

A: CTEs do not have dedicated stats. They rely on stats on the underlying objects

C: Unlike a derived table, a CTE can be self-referencing and can be referenced multiple times in the same query.

References: https://technet.microsoft.com/en-us/library/ms190766(v=sql.105).aspx
https://dba.stackexchange.com/questions/13112/whats-the-difference-between-a-cte-and-a-temp-table

QUESTION 9

You have a disk-based table that contains 15 columns.

You query the table for the number of new rows created during the current day.

You need to create an index for the query. The solution must generate the smallest possible index.

Which type of index should you create?

A. clustered

B. filtered nonclustered with a getdate() predicate in the WHERE statement clause

C. hash

D. nonclustered with compression enabled

Correct Answer: B

A filtered index is an optimized nonclustered index especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve

query performance as well as reduce index maintenance and storage costs compared with full-table indexes.

Creating a filtered index can reduce disk storage for nonclustered indexes when a full-table index is not necessary.

References: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-filtered-indexes

QUESTION 10

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 are building a stored procedure that will be used by hundreds of users concurrently.

You need to store rows that will be processed later by the stored procedure. The object that stores the rows must meet the following requirements:

Be indexable

Contain up-to-date statistics

Be able to scale between 10 and 100,000 rows

The solution must prevent users from accessing one another\\'s data.

Solution: You create a table variable in the stored procedure.

Does this meet the goal?

A.

Yes

B.

No

Correct Answer: B

QUESTION 11

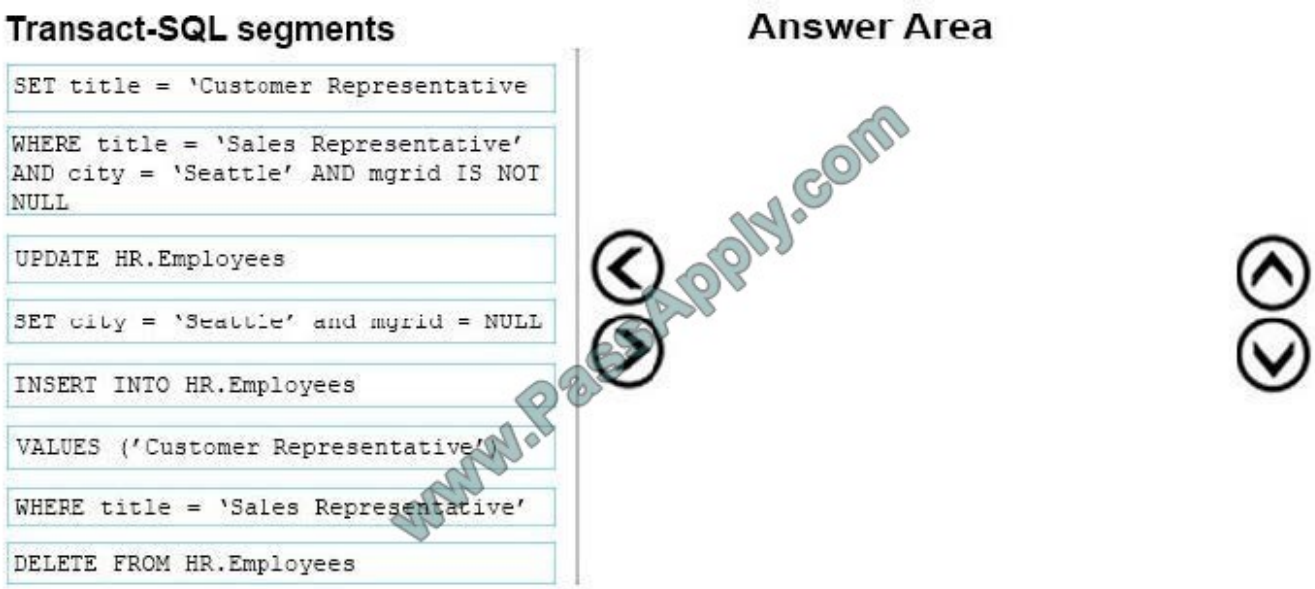
DRAG DROP

You have a table named HR.Employees as shown in the exhibit. (Click the exhibit button.)

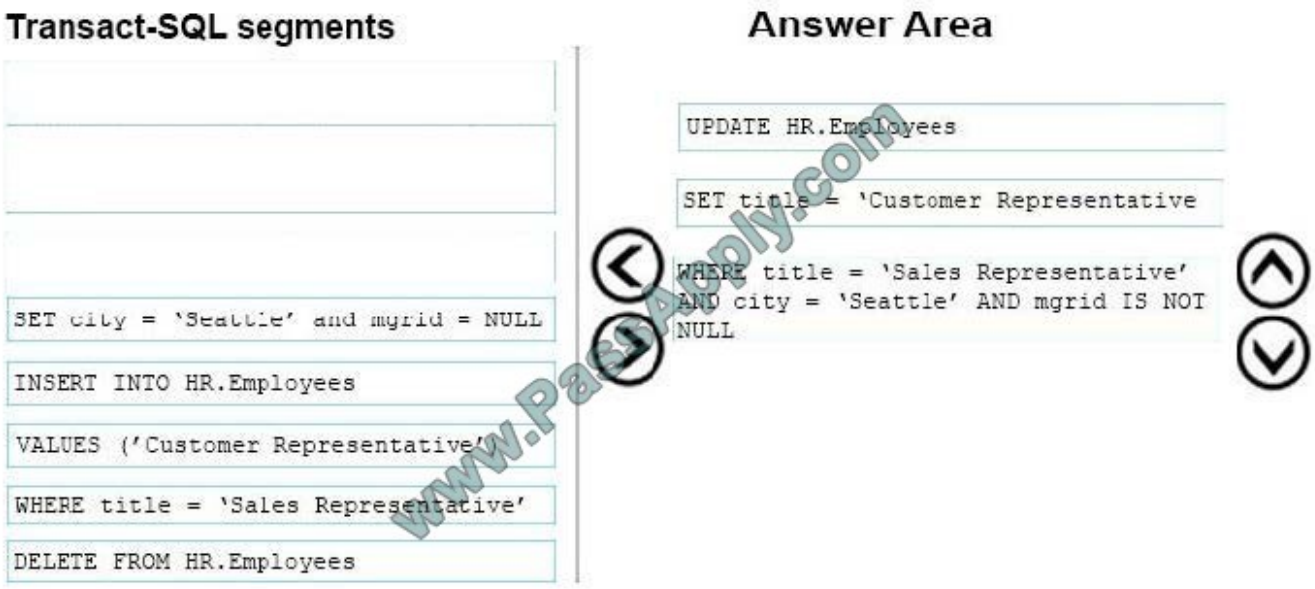
You need to write a query that will change the value of the job title column to Customer Representative for any employee who lives in Seattle and has a job title of Sales Representative. If the employee does not have a manager defined, you must not change the title.

Which three 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:



Correct Answer:



References: https://msdn.microsoft.com/en-us/library/ms177523.aspx

QUESTION 12

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.

Multiple processes use the data from a table named Sales and place it in other databases across the organization. Some of the processes are not completely aware of the data types in the Sales table. This leads to data type conversion errors.

You need to implement a method that returns a NULL value id data conversion fails instead of throwing an error.

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

Correct Answer: H

TRY_CONVERT returns a value cast to the specified data type if the cast succeeds; otherwise, returns null.
References: https://docs.microsoft.com/en-us/sql/t-sql/functions/try-convert-transact-sql

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.passapply.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: