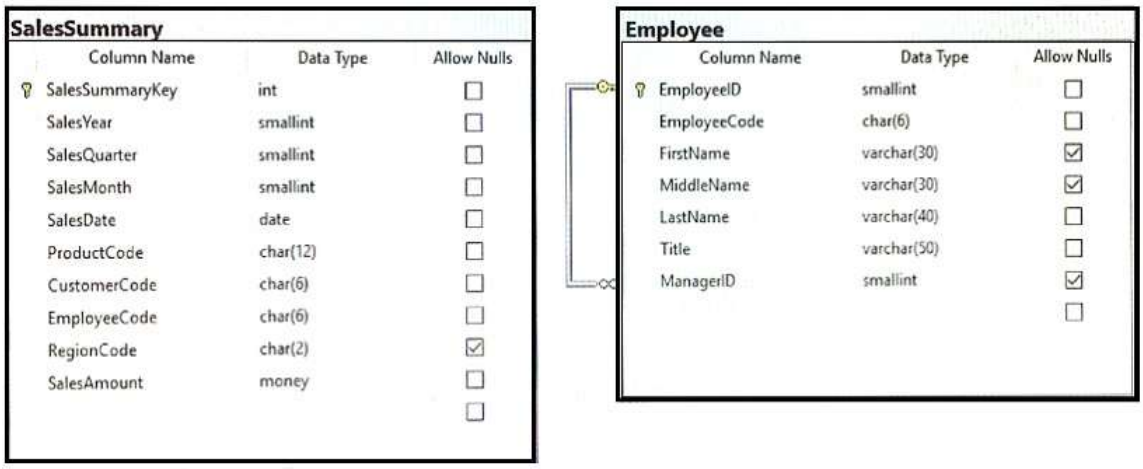
2018 NEW Microsoft 70-761: Querying Data with Transact-SQL Exam Questions and Answers RELEASED in Braindump2go.com Online IT Study Website Today!

2018 Braindump2go Valid Microsoft 70-761 Exam Preparation Materials:

2018 Latest 135Q&As 70-761 PDF Dumps and VCE Dumps:

https://www.braindump2go.com/70-761.html

QUESTION 96
Drag and Drop Question
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.
Start of repeated scenario
You have a database that contains the tables shown in the exhibit. (Click the Exhibit button.)



You review the Employee table and make the following observations:
- Every record has a value in the ManagerID except for the Chief Executive Officer (CEO).
- The FirstName and MiddleName columns contain null values for some

records.
- The valid values for the Title column are Sales Representative manager, and CEO.

You review the SalesSummary table and make the following observations:
- The ProductCode column contains two parts: The first five digits represent a product code, and the last seven digits represent the unit price. The unit price uses the following pattern: ####.##.
- You observe that for many records, the unit price portion of the ProductCode column contains values.
- The RegionCode column contains NULL for some records.
- Sales data is only recorded for sales representatives.

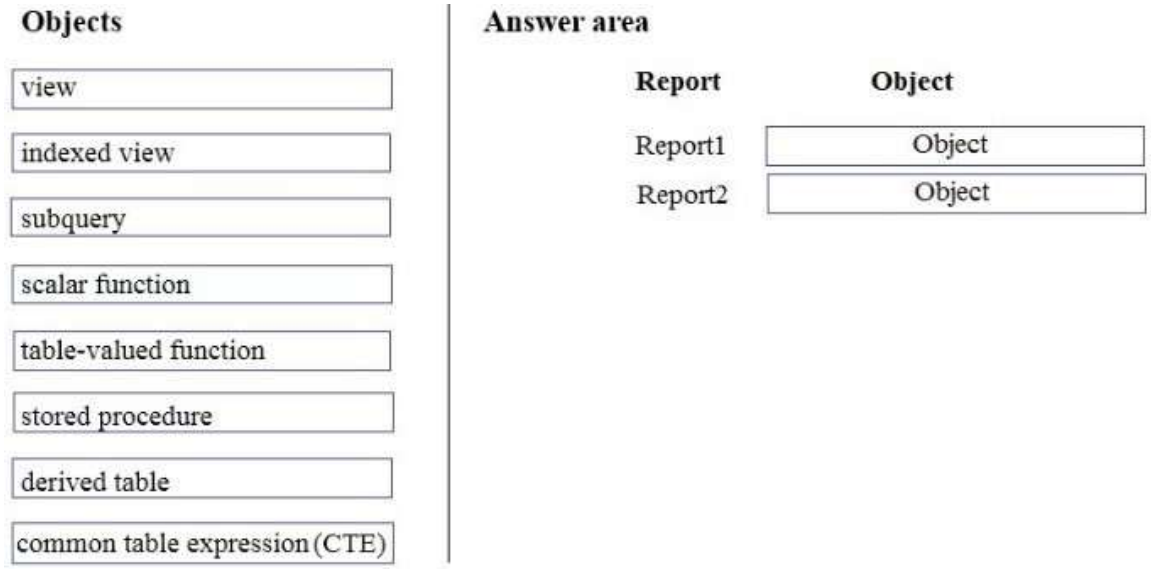
You are developing a series of reports and procedures to support the business.
Details for each report or procedure follow.
Sales Summary report: This report aggregates data by year and quarter.
The report must resemble the following table.

SalesYear	SalesQuarter	YearSalesAmount	QuarterSalesAmount
2015	1	2000.00	1000.00
2015	2	2000.00	500.00
2015	3	2000.00	250.00
2015	4	2000.00	250.00
2016	1	3500.00	500.00
2016	2	3500.00	1000.00

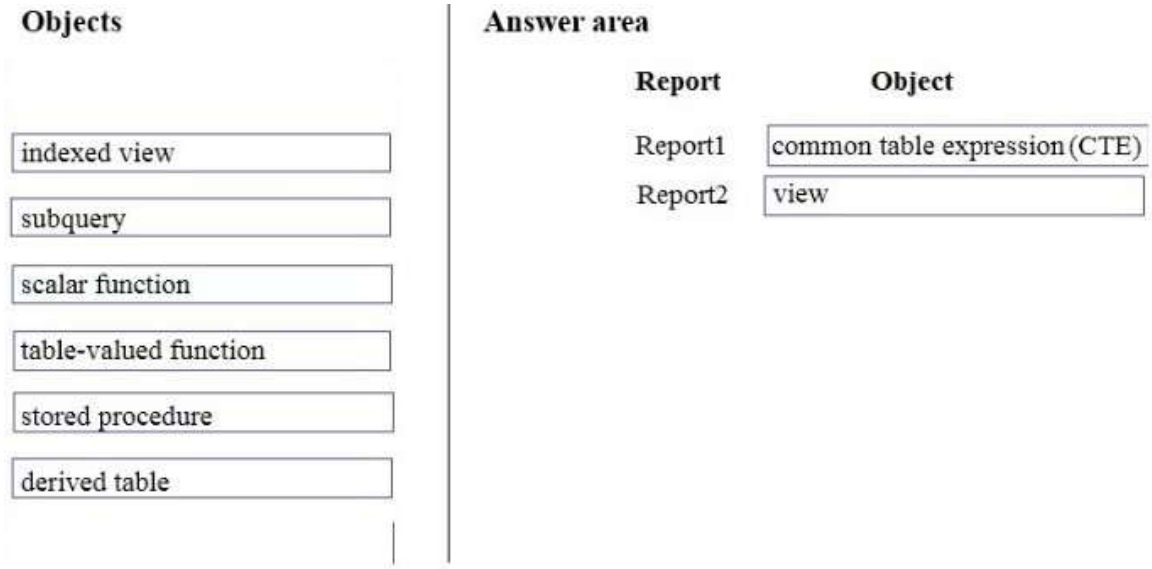
Sales Manager report: This report lists each sales manager and the total sales amount for all employees that report to the sales manager.
Sales by Region report: This report lists the total sales amount by employee and by region. The report must include the following columns: EmployeeCode, MiddleName, LastName, RegionCode, and SalesAmount. If MiddleName is NULL, FirstName must be displayed. If both FirstName and MiddleName have null values, the world Unknown must be displayed/ If RegionCode is NULL, the word Unknown must be displayed.
Report1: This report joins data from SalesSummary with the Employee table and other tables.
You plan to create an object to support Report1. The object has the following requirements:
- be joinable with the SELECT statement that supplies data for the report
- can be used multiple times with the SELECT statement for the report
- be usable only with the SELECT statement for the report
- not be saved as a permanent object

Report2: This report joins data from SalesSummary with the Employee table and other tables.
You plan to create an object to support Report1. The object has the following requirements:
Sales Hierarchy report. This report aggregates rows, creates subtotal rows, and super-aggregates rows over the SalesAmount column in a single result-set. The report uses SaleYear, SaleQuarter, and SaleMonth as a hierarchy. The result set must not contain a grand total or cross-tabulation aggregate rows.
Current Price Stored Procedure: This stored procedure must return the unit price for a product when a product code is supplied. The unit price must include a dollar sign at the beginning. In addition, the unit price must contain a comma every three digits to the left of the decimal point, and must display two digits to the left of the decimal point. The stored procedure must not throw errors, even if the product code contains invalid data.
End of Repeated Scenario
You are creating the queries for Report1 and Report2.
You need to create the objects necessary to support the queries.

Which object should you use to join the SalesSummary table with the other tables that each report uses? To answer, drag the appropriate objects to the correct reports. each object 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.



Answer:



Explanation:

Box 1: common table expression (CTE)

A common table expression (CTE) can be thought of as a temporary result set that is defined within the execution scope of a single SELECT, INSERT, UPDATE, DELETE, or CREATE VIEW statement. A CTE is similar to a derived table in that it is not stored as an object and lasts only for the duration of the query. Unlike a derived table, a CTE can be self-referencing and can be referenced multiple times in the same query.

A CTE can be used to:

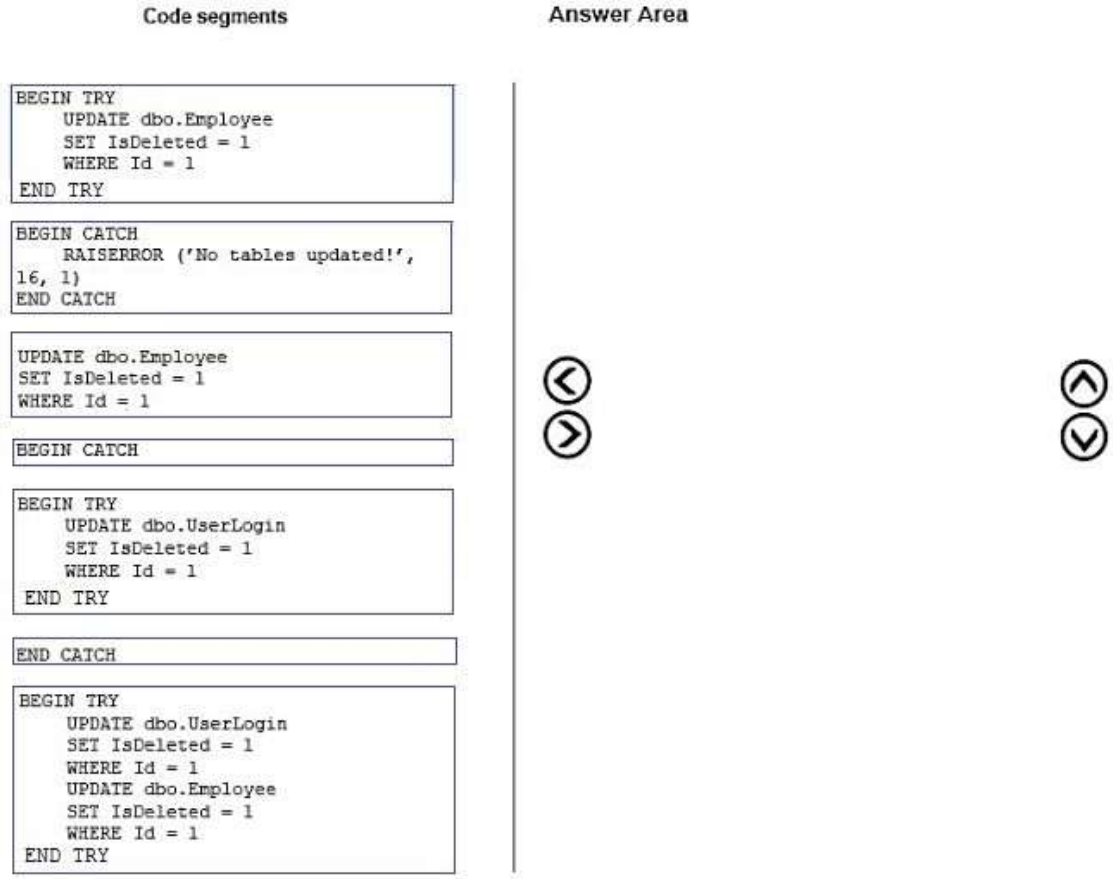
From Scenario: Report1: This report joins data from SalesSummary with the Employee table and other tables. You plan to create an object to support Report1. The object has the following

requirements:
Box 2: view
From scenario: Report2: This report joins data from SalesSummary with the Employee table and other tables.
You plan to create an object to support Report1. The object has the following requirements:
References: https://technet.microsoft.com/en-us/library/ms190766(v=sql.105).aspx

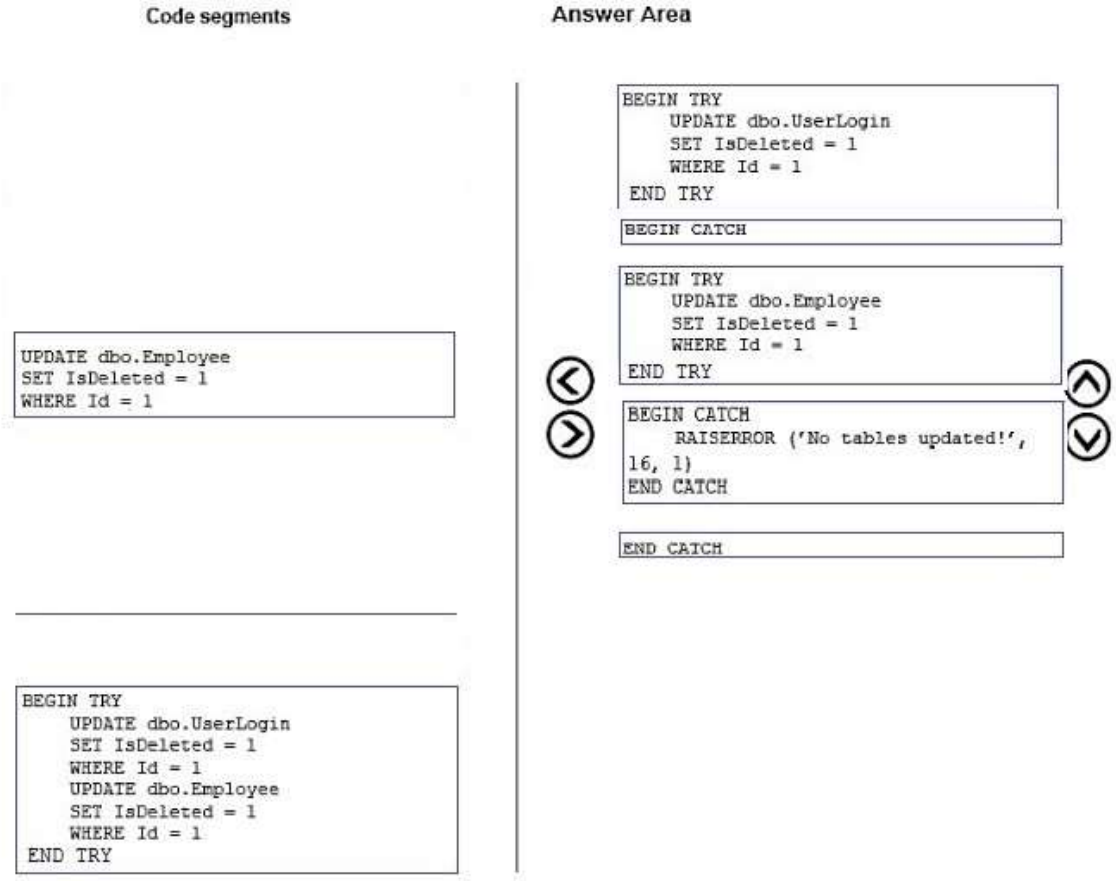
QUESTION 97
Drag and Drop Question
You have two tables named UserLogin and Employee respectively.
You need to create a Transact-SQL script that meets the following requirements:

```
- 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 to 1.
- 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.



Answer:



QUESTION 98

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 table named Products that stores information about products your company sells. The table has a column named ListPrice that stores retail pricing information for products.
Some products are used only internally by the company. Records for these products are maintained in the Products table for inventory purposes. The price for each of these products is $0.00. Customers are not permitted to order these products.
You need to increase the list price for products that cost less than $100 by 10 percent. You must only increase pricing for products that customers are permitted to order.
Solution: You run the following Transact-SQL statement:

```
UPDATE Production.Product
SET ListPrice = ListPrice + 1.1
WHERE ListPrice < 100
```

Does the solution meet the goal?

A. Yes
B. No

Answer: B
Explanation:
Products with a price of $0.00 would also be increased.

QUESTION 99
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 table named Products that stores information about products your company sells. The table has a column named ListPrice that stores retail pricing information for products.
Some products are used only internally by the company. Records for these products are maintained in the Products table for inventory purposes. The price for each of these products is $0.00. Customers are not permitted to order these products.
You need to increase the list price for products that cost less than $100 by 10 percent. You must only increase pricing for products that customers are permitted to order.
Solution: You run the following Transact-SQL statement:

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

Does the solution meet the goal?

A. Yes
B. No

Answer: B
Explanation:
Products with a price of $0.00 would also be increased.

QUESTION 100
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 table named Products that stores information about products your company sells. The table has a column named ListPrice that stores retail pricing information for products.
Some products are used only internally by the company. Records for these products are maintained in the Products table for inventory purposes. The price for each of these products is $0.00. Customers are not permitted to order these products.
You need to increase the list price for products that cost less than $100 by 10 percent. You must

only increase pricing for products that customers are permitted to order.
Solution: You run the following Transact-SQL statement:

```
UPDATE Production.Product
SET ListPrice = ListPrice + 1.1
WHERE ListPrice
BETWEEN .01 and 99.99
```

Does the solution meet the goal?

A. Yes
B. No

Answer: B
Explanation:
Products with a price between $0.00 and $100 will be increased, while products with a price of $0.00 would not be increased.

QUESTION 101
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 table that was created by running the following Transact-SQL statement:

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

The Products table includes the data shown in the following table:

ProductID	ProductName	UnitPrice	UnitsInStock	UnitsOnOrder
1	ProductA	10.00	10	15
2	ProductB	30.00	20	Null
3	ProductC	15.00	5	20

TotalUnitPrice is calculated by using the following formula:
`TotalUnitPrice = UnitPrice * (UnitsInStock + UnitsOnOrder)`
You need to ensure that the value returned for TotalUnitPrice for ProductB is equal to 600.00.
Solution: You run the following Transact-SQL statement:

```
SELECT ProductName, UnitPrice*(UnitsInStock+ISNULL(UnitsOnOnrder,0)) AS
TotalUnitPrice FROM Products
```

Does the solution meet the goal?

A. Yes
B. No

Answer: A
Explanation:
ISNULL (check_expression , replacement_value)
Arguments:
check_expression
Is the expression to be checked for NULL. check_expression can be of any type.
replacement_value
Is the expression to be returned if check_expression is NULL. replacement_value must be of a type that is implicitly convertible to the type of check_expresssion.
References: https://docs.microsoft.com/en-us/sql/t-sql/functions/isnull-transact-sql

QUESTION 102
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 table that was created by running the following Transact-SQL statement:

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

The Products table includes the data shown in the following table:

ProductID	ProductName	UnitPrice	UnitsInStock	UnitsOnOrder
1	ProductA	10.00	10	15
2	ProductB	30.00	20	Null
3	ProductC	15.00	5	20

TotalUnitPrice is calculated by using the following formula:
`TotalUnitPrice = UnitPrice * (UnitsInStock + UnitsOnOrder)`
You need to ensure that the value returned for TotalUnitPrice for ProductB is equal to 600.00.
Solution: You run the following Transact-SQL statement:

```
SELECT ProductName, UnitPrice*(UnitsInStock+COALESCE(UnitsOnOnrder,0)) AS
TotalUnitPrice FROM Products
```

Does the solution meet the goal?

A. Yes
B. No

Answer: A
Explanation:
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 103
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 table that was created by running the following Transact-SQL statement:

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

The Products table includes the data shown in the following table:

ProductID	ProductName	UnitPrice	UnitsInStock	UnitsOnOrder
1	ProductA	10.00	10	15
2	ProductB	30.00	20	Null
3	ProductC	15.00	5	20

TotalUnitPrice is calculated by using the following formula:
`TotalUnitPrice = UnitPrice * (UnitsInStock + UnitsOnOrder)`
You need to ensure that the value returned for TotalUnitPrice for ProductB is equal to 600.00.
Solution: You run the following Transact-SQL statement:

```
SELECT ProductName, UnitPrice*(UnitsInStock+UnitsOnOrder) AS
TotalUnitPrice FROM Products
```

Does the solution meet the goal?

A. Yes

B. No

Answer: B
Explanation:
The NULL value in the UnitsOnOrder field would cause a runtime error.

QUESTION 104
You have a database that stores information about server and application errors. The database contains the following table:
Servers

Column	Data type	Notes
ServerID	int	This is the primary key for the table.
DNS	Nvarchar(100)	Null values are not permitted for this column.

Errors

Column	Data type	Notes
ErrorID	int	This is the primary key for the table.
ServerID	int	Null values are not permitted for this column. This column is a foreign key that is related for the `ServerID` column in the `Servers` table.
Occurrences	int	Null values are not permitted for this column.
LogMessage	nvarchar(max)	Null values are not permitted for this column.

You need to return all unique error log messages and the server where the error occurs most often.
Which Transact-SQL statement should you run?

A.
```
SELECT DISTINCT ServerID, LogMessage FROM Errors AS e1
WHERE LogMessage IN (
       SELECT TOP 1 e2.LogMessage FROM Errors AS e2
       WHERE e2.LogMessage = e1.LogMessage AND e2.ServerID <> e1.ServerID
       ORDER BY e2.Occurrences
)
```

B.
```
SELECT DISTINCT ServerID, LogMessage FROM Errors AS e1
WHERE Occurrences > ALL (
       SELECT e2.LogMessage FROM Errors AS e2
       WHERE e2.LogMessage = e1.LogMessage AND e2.ServerID <> e1.ServerID
)
```

C.
```
SELECT DISTINCT ServerID, LogMessage FROM Errors AS e1
GROUP BY ServerID, LogMessage
HAVING MAX(Occurrences) = 1
```

D.

```
SELECT ServerID, LogMessage FROM Errors AS e1
GROUP BY ServerID, LogMessage, Occurrences
HAVING COUNT(*) = 1
ORDER BY Occurrences
```

Answer: A

QUESTION 105
You have a database that includes the following tables.
HumanResources.Employee

Column	Data type	Notes
BusinessEntityID	int	primary key

Sales.SalesPerson

Column	Data type	Notes
BusinessEntityID	int	primary key
CommissionPct	smallmoney	does not allow null values

The HumanResources.Employee table has 2,500 rows, and the Sales.SalesPerson table has 2,000 rows.
You review the following Transact-SQL statement:

```
SELECT e.BusinessEntityID
FROM HumanResources.Employee AS e
WHERE 0.015 IN
      (SELECT CommissionPct
      FROM Sales.SalesPerson AS sp
      WHERE e.BusinessEntityID = sp.BusinessEntityID)
```

You need to determine the performance impact of the query.
How many times will a lookup occur on the primary key index on the Sales.SalesPerson table?

A. 200
B. 2,000
C. 2,500
D. 5,500

Answer: C

QUESTION 106
Hotspot Question
You have a table named HumanResources.Department that was created with the query shown in the exhibit. (Click the Exhibit button.)

```
CREATE TABLE HumanResources.Department
(
    DepID int IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED
  , DeptName varchar(50) NOT NULL
  , ManagerID INT NULL
  , ParentDeptID int NULL
  , SysStartTime datetime2 GENERATED ALWAYS AS ROW START NOT TULL
  , SysEndTime datetime2 GENERATED ALWAYS AS ROW END NOT NULL
  , PERIOD FOR SYSTEM_TIME (SysStartTime,SysEndTime)
)
WITH (SYSTEM_VERSIONING = ON)
;
```

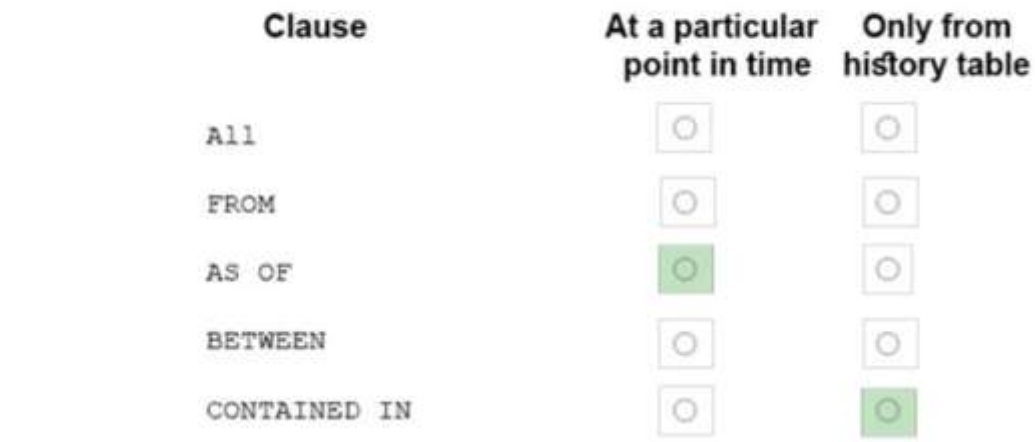
You need to query temporal data in the table.
In the table below, identify the Transact-SQL segments that must be used to retrieve the appropriate data. **NOTE:** Make only one selection in each column.

Answer Area

Clause	At a particular point in time	Only from history table
All	○	○
FROM	○	○
AS OF	○	○
BETWEEN	○	○
CONTAINED IN	○	○

Answer:

Answer Area



Explanation:
AS OF: Returns a table with a rows containing the values that were actual (current) at the specified point in time in the past.
CONTAINED IN: If you search for non-current row versions only, we recommend you to use CONTAINED IN as it works only with the history table and will yield the best query performance.
Incorrect Answers:
Not ALL: Returns the union of rows that belong to the current and the history table.
References: https://docs.microsoft.com/en-us/sql/relational-databases/tables/querying-data-in-a-system- versioned-temporal-table