DUMPSARENA

Querying Data with Transact-SQL

Microsoft 70-761

Version Demo

Total Demo Questions: 14

Total Premium Questions: 207

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.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.

```
1    SELECT p.productname
2    FROM Production.Categories AS c
3    inner join production.products as p on c.categoryid*p.categoryid
4    WHERE c.categoryname = 'Beverages'
```

Use the Check Syntax button to verify your work. Any syntax or spelling errors will be reported by line and character position. You may check syntax as many times as needed.

ANSWER: Please see explanation

Explanation:

1 SELECT p.productname

2 FROM Production.categories AS c3 inner join production.products as p on c.categoryid=p.categoryid 4 WHERE c.categoryname = 'Beverages'

Note: On line 3 change * to =

QUESTION NO: 2 - (SIMULATION)

SIMULATION

You create a table named Sales.Orders by running the following Transact-SQL statement:

```
CREATE TABLE Sales.Orders (
    OrderID int NOT NULL,
    OrderDate date NULL,
    ShippedDate date NULL,
    Status varchar(10),
    CONSTRAINT PK_ORDERS PRIMARY KEY CLUSTERED
)
```

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

- removes orders from the table that were placed before January 1, 2012
- uses the date format of YYYYMMDD
- ensures that the order has been shipped before deleting the record

Construct the query using the following guidelines:

- use one-part column names and two-part table names
- do not use functions
- do not surround object names with square brackets
- do not use variables
- do not use aliases for column names and table names

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.

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

ANSWER: See the solution below

Explanation:

DELETE FROM Sales.Orders

WHERE OrderDate < '2012-01-01' AND ShippedDate NOT NULL

References:

https://msdn.microsoft.com/en-us/library/ms189835.aspx https://msdn.microsoft.com/en-us/library/bb630352.aspx

QUESTION NO: 3

You run the following Transact-SQL statement:

```
CREATE TABLE CourseParticipants
 (
  CourseID INT NOT NULL,
  CourseDate DATE NOT NULL,
  LocationDescription VARCHAR(100) NOT NULL,
  NumParticipants INT NOT NULL
 )
```

You need to create a query that returns the total number of attendees for each combination of CourseID, CourseDate, and the following locations: Lisbon, London, and Seattle. The result set should resemble the following:

	CourseID	CourseDate	Lisbon	London	Seattle
1	1	2018-02-01	NULL	NULL	15
2	2	2018-02-01	33	NULL	NULL
3	1	2018-02-02	NULL	20	NULL
4	1	2018-02-03	20	10	NULL
5	2	2018-02-03	NULL	20	NULL

Which Transact-SQL code segment should you run?

A.
```
SELECT *
   FROM CourseParticipants
   PIVOT(SUM(NumParticipants) FOR LocationDescription
   IN (Lisbon, London, Seattle))
```

B.
```
SELECT *
   FROM CourseParticipants
   PIVOT(SUM(NumParticipants) FOR LocationDescription
   IN (Lisbon, London, Seattle)) as PVTTable
```

C.
```
SELECT *
   FROM CourseParticipants
   UNPIVOT(SUM(NumParticipants) FOR LocationDescription
   IN (Lisbon, London, Seattle)
```

D.
```
SELECT *
   FROM CourseParticipants
   UNPIVOT(SUM(NumParticipants) FOR LocationDescription
   IN (Lisbon, London, Seattle) AS PVTTable
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: B

Explanation:

References: https://www.techonthenet.com/sql_server/pivot.php

QUESTION NO: 4 - (DRAG DROP)

DRAG DROP

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 are developing a database to track customer orders. The database contains the following tables: Sales.Customers, Sales.Orders, and Sales.OrderLines. The following table describes the columns in Sales.Customers.

Column name	Data type	Constraints
CustomerID	int	primary key
CustomerName	nvarchar(100)	does not allow null values
PhoneNumber	nvarchar(20)	does not allow null values
AccountOpenedDate	date	does not allow null values
StandardDiscountPercentage	decimal(18,3)	does not allow null values
CreditLimit	decimal(18,2)	null values are permitted
IsOnCreditHold	bit	does not allow null values
DeliveryLocation	geography	does not allow null values
PhoneNumber	nvarchar(20)	does not allow null values

The following table describes the columns in Sales.Orders.

Column name	Data type	Constraints
OrderID	int	primary key
CustomerID	int	foreign key to the Sales.Customers table
OrderDate	date	does not allow null values

The following table describes the columns in Sales.OrderLines.

Column name	Data type	Constraints
OrderLineID	int	primary key
OrderID	int	foreign key to the Sales.Orders table
Quantity	int	does not allow null values
UnitPrice	decimal(18,2)	null values are permitted
TaxRate	decimal(18,2)	does not allow null values

You need to create a stored procedure that inserts data into the Customers table. The stored procedure must meet the following requirements:

- Data changes occur as a single unit of work.
- Data modifications that are successful are committed and a value of 0 is returned to the calling procedure.
- Data modifications that are unsuccessful are rolled back. You must display a message that uses severity level 16 and a value of -1.
- The stored procedure uses a built-in scalar function to evaluate the current condition of data modifications.
- The entire unit of work is terminated and rolled back if a run-time error occurs during execution of the stored procedure.

How should complete the stored procedure definition? To answer, drag the appropriate Transact-SQL segments to the correct targets. Each TransactSQL segment 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.

NOTE: Each correct selection is worth one point.

Select and Place:

ANSWER:

Explanation:

Box 1: XACT_ABORT

XACT_ABORT specifies whether SQL Server automatically rolls back the current transaction when a Transact-SQL statement raises a run-time error. When SET XACT_ABORT is ON, if a Transact-SQL statement raises a run-timeerror, the entire transaction is terminated and rolled back.

Box 2: COMMIT

Commit the transaction.

Box 3: XACT_STATE

Box 4: ROLLBACK Rollback the transaction

Box 5: THROW

THROW raises an exception and the severity is set to 16.

Requirement: Data modifications that are unsuccessful are rolled back. The exception severity level is set to 16 and a value of -1 is returned.

References: https://msdn.microsoft.com/en-us/library/ms188792.aspx https://msdn.microsoft.com/en-us/library/ee677615.aspx

QUESTION NO: 5

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

QUESTION NO: 6

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:

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 create a list of all unique customers that appear in either table.

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: E

Explanation:

UNION combines the results of two or more queries into a single result set that includes all the rows that belong to all queries in the union. The UNION operation is different from using joins that combine columns from two tables.

Incorrect Answers:

F: UNION ALL incorporates all rows into the results. This includes duplicates. If not specified, duplicate rows are removed.
References: https://msdn.microsoft.com/en-us/library/ms180026.aspx

QUESTION NO: 7

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:

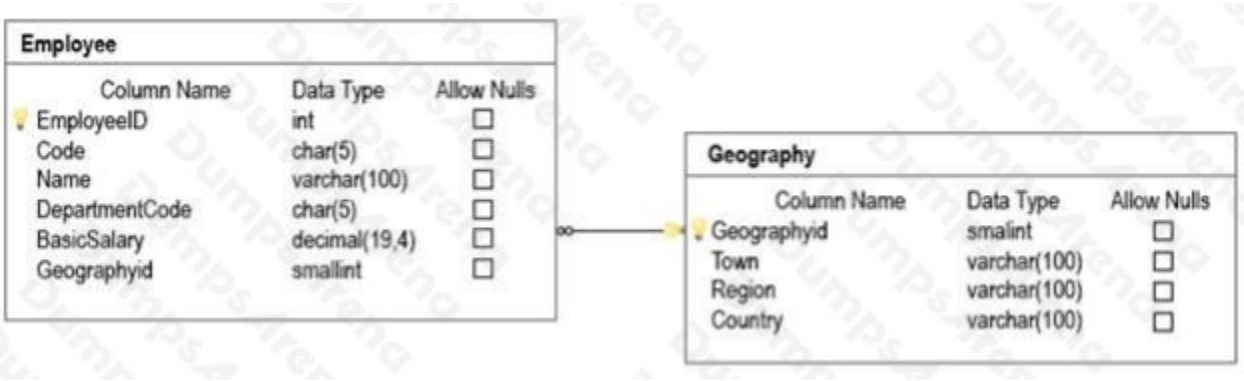
A: A table valued function cannot be called from indexed view C: The Stored procedure cannot be called inside of a View.

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

QUESTION NO: 8 - (HOTSPOT)

HOTSPOT

You have two tables as shown in the following image:



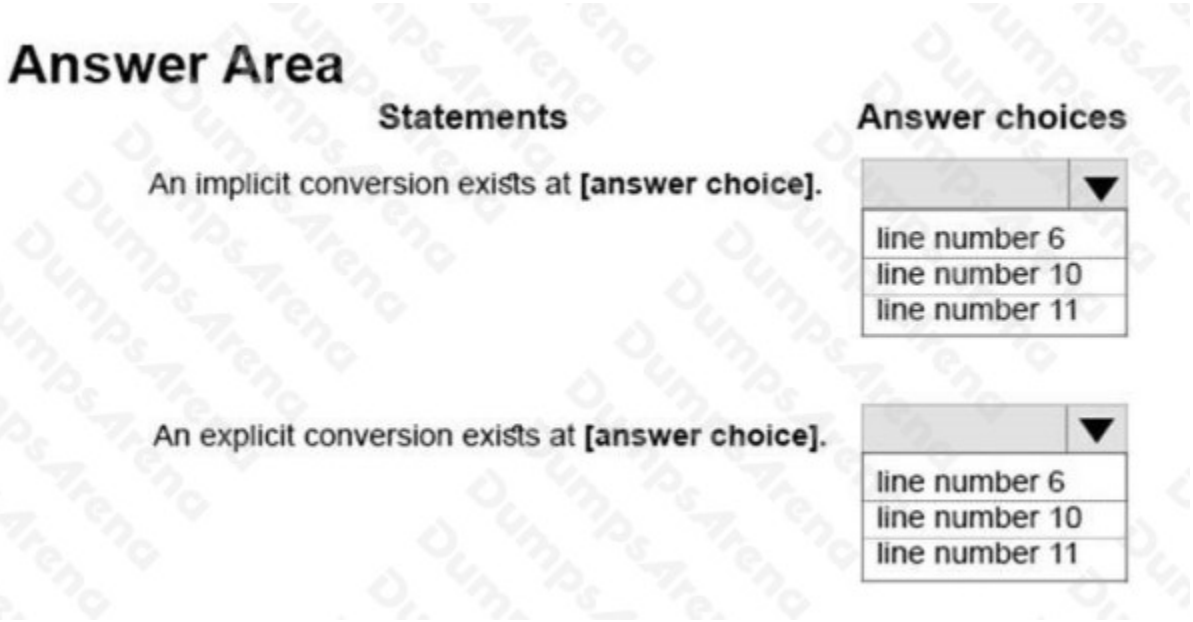
You need to analyze the following query. (Line numbers are included for reference only.)

```
01  DECLARE @DepartmentCode nchar(5) = N'DEP01'
02  DECLARE @RoundedUpSalary int
03  DECLARE @EmployeeName nvarchar(100)
04  SELECT
05      Name,
06      CONVERT(int, Code) EmployeeCode,
07      BasicSalary
08  FROM dbo.Employee e
09  INNER JOIN dbo.Geography g
10  ON e.GeographyId = g.GeographyId
11  WHERE DepartmentCode = @DepartmentCode
```

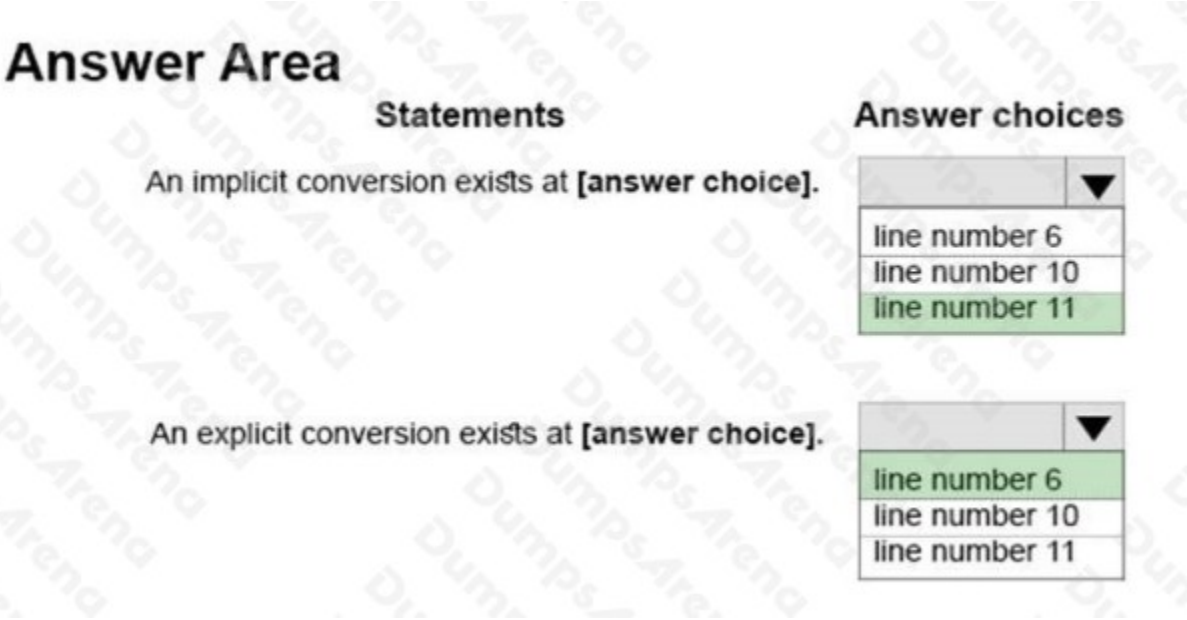
Use the drop-down menus to select the answer choice that completes each statement based on the information presented in the graphic.

NOTE: Each correct selection is worth one point.

Hot Area:



ANSWER:



Explanation:

To compare char(5) and nchar(5) an implicit conversion has to take place.

Explicit conversions use the CAST or CONVERT functions, as in line number 6.

References: https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-type-conversion-database-engine#implicit-and-explicit-conversion

QUESTION NO: 9

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 following records exist in the tables:

Customer_CRMSystem

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

Customer_HRSystem

CustomerID	CustomerCode	CustomerName
1	CUS1	Roya
2	CUS2	Jose
3	CUS9	Yossi
4	NULL	Jane

Records that contain null values for CustomerCode can be uniquely identified by CustomerName.

You need to display distinct customers that appear in 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: B

Explanation:

You need to display distinct customers that appear in both tables.

INTERSECT returns distinct rows that are output by both the left and right input queries operator.

Incorrect Answers:

A: Inner joins return rows only when there is at least one row from both tables that matches the join condition. Inner joins eliminate the rows that do not match with a row from the other table.

D: EXCEPT returns distinct rows from the left input query that aren't output by the right input query.

E: UNION specifies that multiple result sets are to be combined and returned as a single result set, but this will not work here as the CustomerID column values do not match.

F: UNION ALL incorporates all rows into the results. This includes duplicates. If not specified, duplicate rows are removed.

G: A cross join would produce the Cartesian product of the two tables.

H: To retain the nonmatching information by including nonmatching rows in the results of a join, use a full outer join. SQL Server provides the full outer join operator, FULL OUTER JOIN, which includes all rows from both tables, regardless of whether or not the other table has a matching value.

References:

https://technet.microsoft.com/en-us/library/ms187518(v=sql.105).aspx

QUESTION NO: 10 - (DRAG DROP)

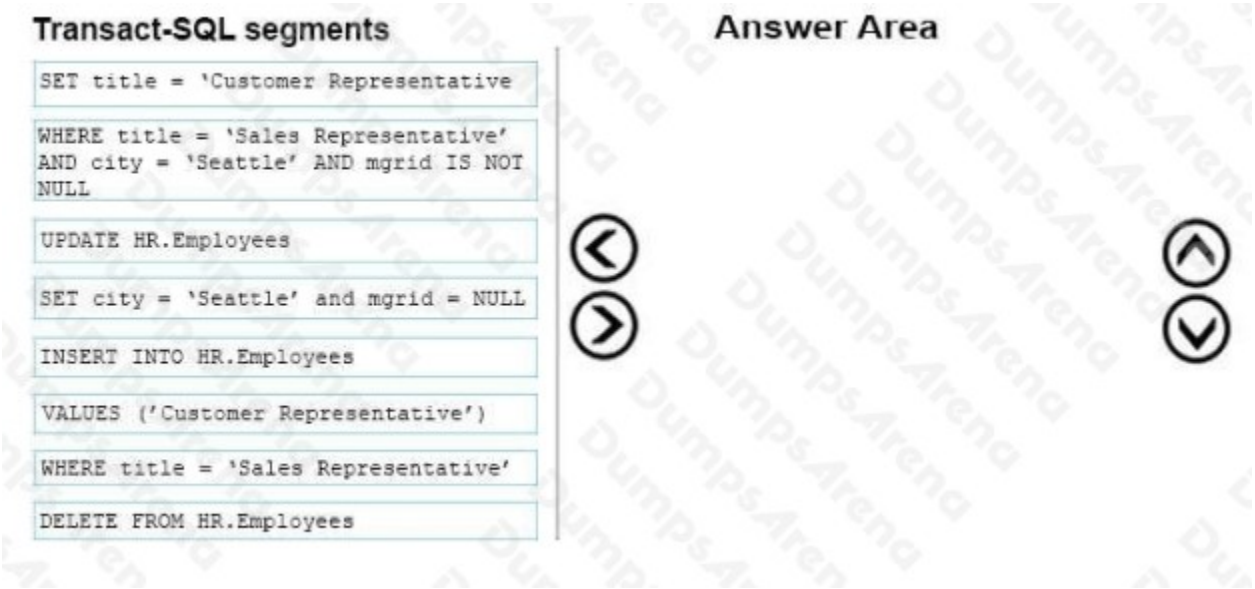
DRAG DROP

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

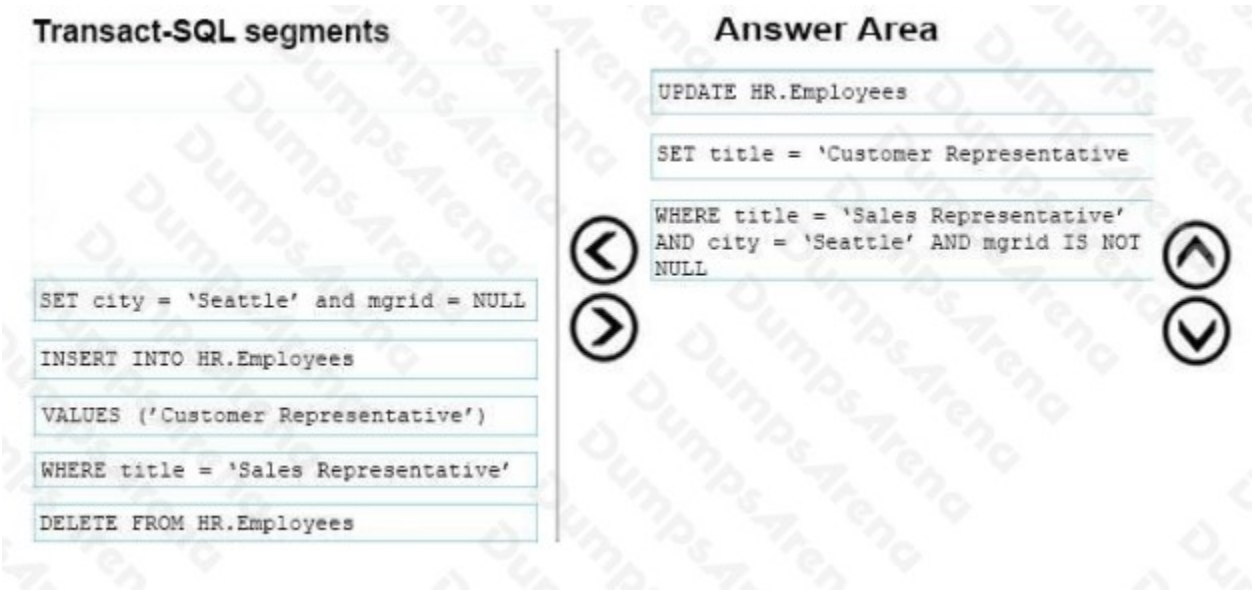
You need to write a query that will change the value of the job title column to Customer Representative for any employee who lives in Seattle and has a job title of Sales Representative. If the employee does not have a manager defined, you must not change the title.

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:



ANSWER:



Explanation:

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

QUESTION NO: 11

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: 12 - (HOTSPOT)

HOTSPOT

You have a database that contains the following tables: tblRoles, tblUsers, and tblUsersInRoles.

The table tblRoles is defined as follows.

Column name	Data type	Nullable	Primary key
RoleID	int	No	Yes
RoleName	varchar(20)	No	No

You have a function named ufnGetRoleActiveUsers that was created by running the following Transact-SQL statement:

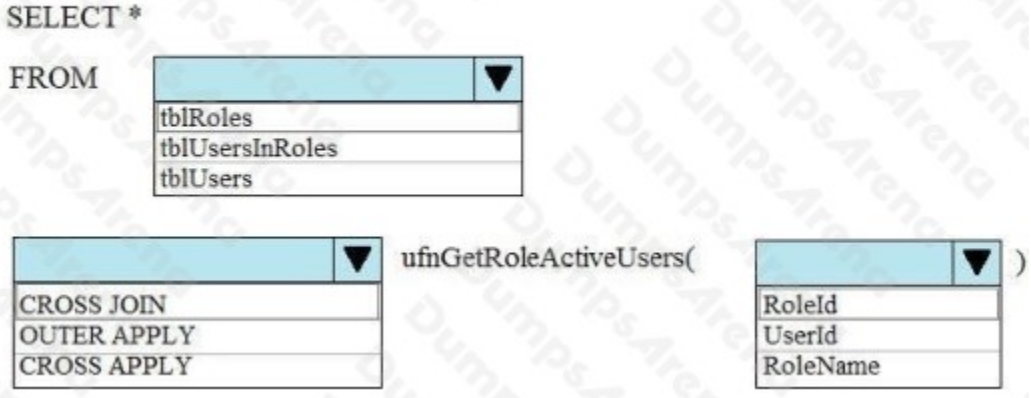
```
CREATE FUNCTION ufnGetRoleActiveUsers(@RoleId AS int)
RETURNS @roleSummary TABLE(UserName varchar (20))
AS
BEGIN
  INSERT INTO @roleSummary
  SELECT U.UserName FROM tblUsersInRoles BRG
  INNER JOIN tblUsers U
  ON U.UserId = BRG.UserId
  WHERE BRG.RoleId = @RoleId AND U.IsActive = 1
  RETURN
END
```

You need to list all roles and their corresponding active users. The query must return the RoleId, RoleName, and UserName columns. If a role has no active users, a NULL value should be returned as the UserName for that role.

How should you complete the Transact-SQL statement? To answer, select the appropriate Transact-SQL segments in the answer area.

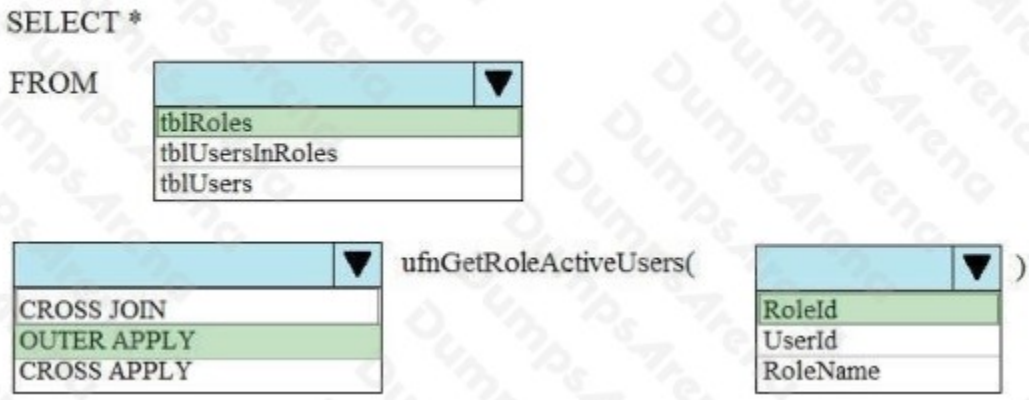
Hot Area:

Answer area



ANSWER:

Answer area



Explanation:

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 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

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

You are performing a code review of stored procedures. Code at line SP03 fails to run (Line numbers are included for reference only.)

```
SP01 BEGIN TRY
SP02  BEGIN TRANSACTION
SP03  . . .
SP04  COMMIT TRANSACTION
SP05 END TRY
SP06 BEGIN CATCH
SP07
SP08  ROLLBACK TRANSACTION
SP09 END CATCH
```

You need to ensure that transactions are rolled back when an error occurs.

Which Transact-SQL segment should you insert at line SP07?

A. If @@Error <> 0

B. If @@ TRANCOUNT = 0

C. If @@ TRANCOUNT > 0

D. If @@ Error = 0

ANSWER: C

Explanation:

Using TRY...CATCH in a transaction

The following example shows how a TRY...CATCH block works inside a transaction. The statement inside the TRY block generates a constraint violation error.

BEGIN TRANSACTION;

BEGIN TRY

-- Generate a constraint violation error.

DELETE FROM Production.Product

WHERE ProductID = 980;

END TRY

BEGIN CATCH

SELECT

ERROR_NUMBER() AS ErrorNumber

,ERROR_SEVERITY() AS ErrorSeverity

,ERROR_STATE() AS ErrorState

```
,ERROR_PROCEDURE() AS ErrorProcedure
,ERROR_LINE() AS ErrorLine
,ERROR_MESSAGE() AS ErrorMessage;
IF @@TRANCOUNT > 0
ROLLBACK TRANSACTION;
END CATCH;
IF @@TRANCOUNT > 0
COMMIT TRANSACTION;
GO
```

References: https://docs.microsoft.com/en-us/sql/t-sql/language-elements/try-catch-transact-sql