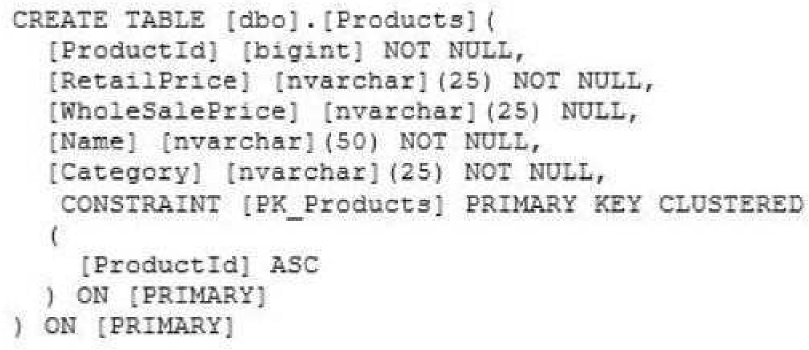
70-761

Number: 70-761
Passing Score: 800
Time Limit: 120 min
File Version: 1.0

Querying Data with Transact-SQL (beta)

Exam A

QUESTION 1
You develop a Microsoft SQL Server 2012 database that contains a table named Products. The Products table has the following definition:



```
CREATE TABLE [dbo].[Products](
  [ProductId] [bigint] NOT NULL,
  [RetailPrice] [nvarchar](25) NOT NULL,
  [WholeSalePrice] [nvarchar](25) NULL,
  [Name] [nvarchar](50) NOT NULL,
  [Category] [nvarchar](25) NOT NULL,
   CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
  (
    [ProductId] ASC
  ) ON [PRIMARY]
) ON [PRIMARY]
```

You need to create an audit record only when either the RetailPrice or WholeSalePrice column is updated.

Which Transact-SQL query should you use?

A. CREATE TRIGGER TrgPriceChange ON Products FOR UPDATE AS IF CCLUMNS_CHANGED(RetailPrice, WholesalePrice)
- - Create Audit Records

B. CREATE TRIGGER TrgPriceChange ON Products FOR UPDATE AS IF EXISTS(SELECT RetailPrice from inserted) OR EXISTS (SELECT WholeSalePnce FROM inserted)
- - Create Audit Records

C. CREATE TRIGGER TrgPriceChange ON Products FOR UPDATE AS IF COLUMNS_UPDATED(RetailPrice, WholesalePrice)
- - Create Audit Records

D. CREATE TRIGGER TrgPriceChange ON Products FOR UPDATE AS IF UPDATE(RetailPrice) OR UPDATE(WholeSalePrice)
- - Create Audit Records

Correct Answer: D

Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/bb510663.aspx Reference: http://msdn.microsoft.com/en-us/library/ms186329.aspx

QUESTION 2
A table named Profits stores the total profit made each year within a territory. The Profits table has columns named Territory, Year, and Profit.

You need to create a report that displays the profits made by each territory for each year and its previous year.

Which Transact-SQL query should you use?

A. SELECT Territory, Year, Profit,
LEAD(Profit, 1, 0) OVER (PARTITION BY Territory ORDER BY Year) AS PrevProfit FROM Profits
B. SELECT Territory, Year, Profit,
LAG(Profit, 1, 0) OVER (PARTITION BY Year ORDER BY Territory) AS PrevProfit FROM Profits
C. SELECT Territory, Year, Profit,
LAG(Profit, 1, 0) OVER (PARTITION BY Territory ORDER BY Year) AS PrevProfit FROM Profits
D. SELECT Territory, Year, Profit,
LEAD(Profit, 1, 0) OVER (PARTITION BY Year ORDER BY Territory) AS PrevProfit FROM Profits

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/hh231256.aspx Reference: http://msdn.microsoft.com/en-us/library/hh213125.aspx

QUESTION 3
You use Microsoft SQL Server 2012 database to develop a shopping cart application. You need to rotate the unique values of the ProductName field of a table-valued expression into multiple columns in the output.

Which Transact-SQL operator should you use?

A. CROSS JOIN
B. CROSS APPLY
C. PIVOT

D. UNPIVOT

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:
http://technet.microsoft.com/en-us/library/ms177634.aspx

QUESTION 4
You administer a Microsoft SQL Server database that supports a shopping application.

You need to retrieve a list of customers who live in territories that do not have a sales person.

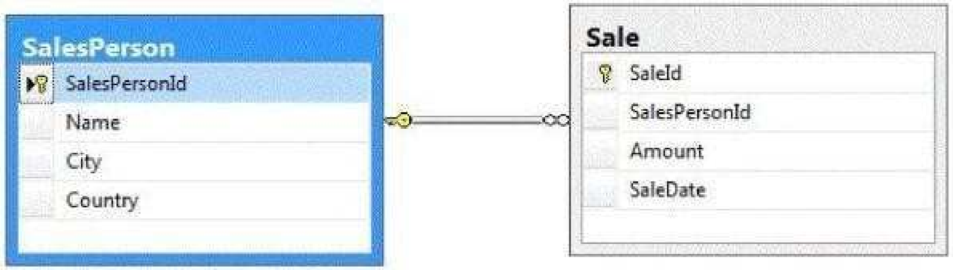
Which Transact- SQL query or queries should you use? (Each correct answer presents a complete solution. Choose all that apply.)

A. SELECT CustomerID FROM Customer
WHERE TerritoryID <>SOME(SELECT TerritoryID FROM Salesperson)
B. SELECT CustomerID FROM Customer
WHERE TerritoryID <>ALL(SELECT TerritoryID FROM Salesperson)
C. SELECT CustomerID FROM Customer
WHERE TerritoryID <>ANY(SELECT TerritoryID FROM Salesperson)
D. SELECT CustomerID FROM Customer
WHERE TerritoryID NOT IN(SELECT TerritoryID FROM Salesperson)

Correct Answer: BD
Section: (none)
Explanation

Explanation/Reference:

QUESTION 5
You support a database structure shown in the exhibit. (Click the Exhibit button.)



You need to write a query that displays the following details:

- Total sales made by sales people, year, city, and country
- Sub totals only at the city level and country level
- A grand total of the sales amount

Which Transact-SQL query should you use?

A. SELECT SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate) AS Year, Sum(Amount) AS Total FROM Sale INNER JOIN SalesPerson
ON Sale.SalesPersonID = SalesPerson.SalesPersonID
GROUP BY GROUPING SETS((SalesPerson.Name, Country, City, DatePart(yyyy, SaleDate)), (Country, City), (Country), ())

B. SELECT SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate) AS Year, Sum(Amount) AS Total FROM Sale INNER JOIN SalesPerson
ON Sale.SalesPersonID = SalesPerson.SalesPersonID
GROUP BY CUBE(SalesPerson.Name, Country, City, DatePart(yyyy, SaleDate))

C. SELECT SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate) AS Year, Sum(Amount) AS Total FROM Sale INNER JOIN SalesPerson
ON Sale.SalesPersonID = SalesPerson.SalesPersonID
GROUP BY CUBE(SalesPerson.Name, DatePart(yyyy, SaleDate), City, Country)

D. SELECT SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate) AS Year, Sum(Amount) AS Total FROM Sale INNER JOIN SalesPerson
ON Sale.SalesPersonID = SalesPerson.SalesPersonID
GROUP BY ROLLUP(SalesPerson.Name, DatePart(yyyy, SaleDate), City, Country)

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Be careful with this question,because on exam can be different options for answer. And none of them is correct : D You should report this question. Reference:

http://www.grapefruitmoon.net/diving-into-t-sql-grouping-sets/ Reference: http://msdn.microsoft.com/en-us/library/ms177673.aspx

QUESTION 6
You are developing a database that will contain price information. You need to store the prices that include a fixed precision and a scale of six digits. Which data type should you use?

A. Float
B. Money
C. Smallmoney
D. Numeric

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Numeric is the only one in the list that can give a fixed precision and scale. Reference: http://msdn.microsoft.com/en-us/library/ms179882.aspx

QUESTION 7
You administer a Microsoft SQL Server database that supports a banking transaction management application.

You need to retrieve a list of account holders who live in cities that do not have a branch location.

Which Transact-SQL query or queries should you use? (Each correct answer presents a complete solution. Choose all that apply.)

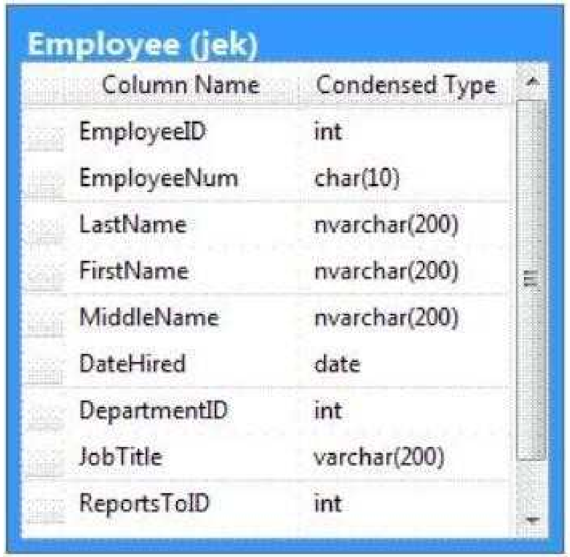
A. SELECT AccountHolderID
FROM AccountHolder
WHERE CityID NOT IN (SELECT CityID FROM BranchMaster)
B. SELECT AccountHolderID
FROM AccountHolder
WHERE CityID <> ALL (SELECT CityID FROM BranchMaster)
C. SELECT AccountHolderID
FROM AccountHolder
WHERE CityID <> SOME (SELECT CityID FROM BranchMaster)
D. SELECT AccountHolderID
FROM AccountHolder
WHERE CityID <> ANY (SELECT CityID FROM BranchMaster)

Correct Answer: AB

Section: (none)
Explanation

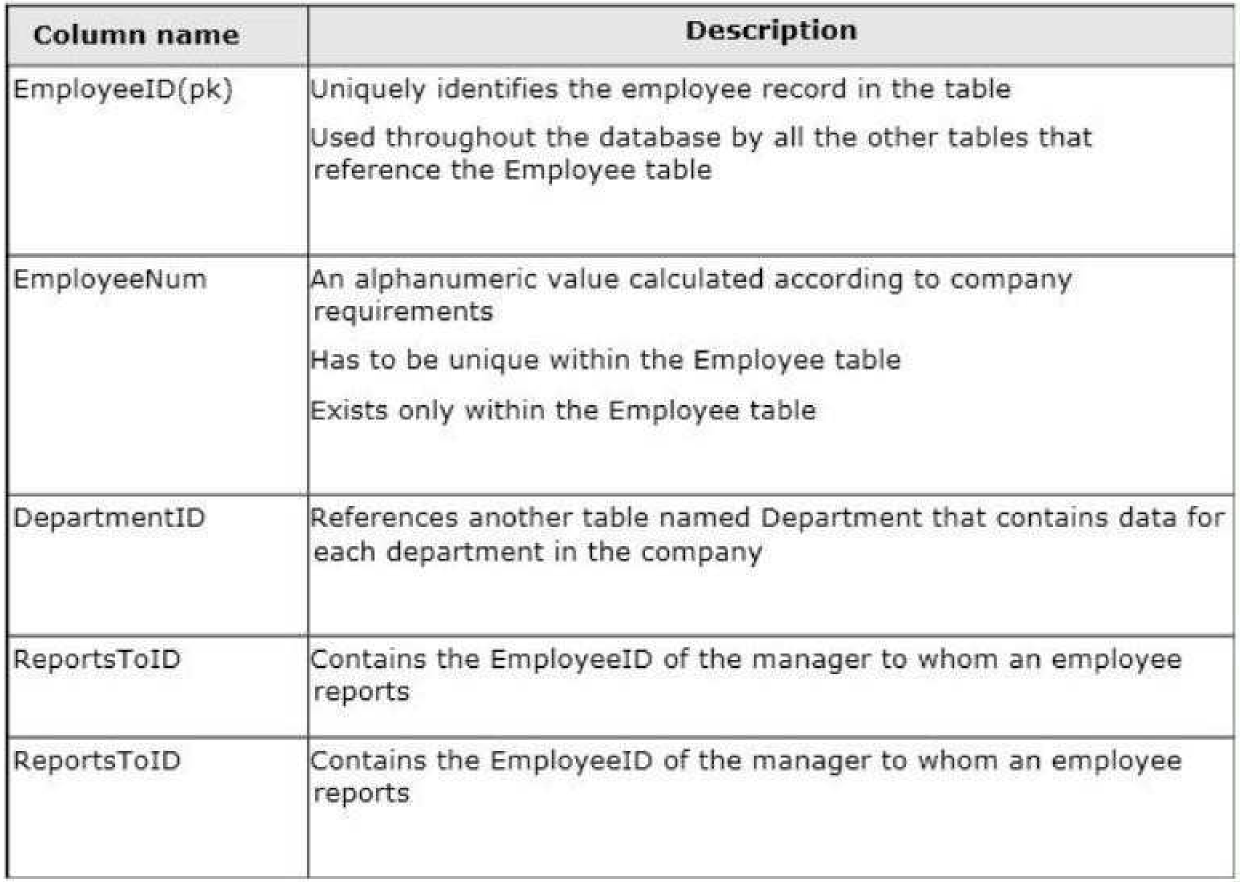
Explanation/Reference:
Explanation:
Verified the answers as correct.
Reference: http://msdn.microsoft.com/en-us/library/ms188047.aspx Reference: http://msdn.microsoft.com/en-us/library/ms177682.aspx Reference: http://msdn.microsoft.com/en-us/library/ms173545.aspx

QUESTION 8
You administer a Microsoft SQL Server 2012 database. The database contains a table named Employee. Part of the Employee table is shown in the exhibit. (Click the Exhibit button.)



Employee (jek)

Column Name	Condensed Type
EmployeeID	int
EmployeeNum	char(10)
LastName	nvarchar(200)
FirstName	nvarchar(200)
MiddleName	nvarchar(200)
DateHired	date
DepartmentID	int
JobTitle	varchar(200)
ReportsToID	int



Column name	Description
EmployeeID(pk)	Uniquely identifies the employee record in the table Used throughout the database by all the other tables that reference the Employee table
EmployeeNum	An alphanumeric value calculated according to company requirements Has to be unique within the Employee table Exists only within the Employee table
DepartmentID	References another table named Department that contains data for each department in the company
ReportsToID	Contains the EmployeeID of the manager to whom an employee reports
ReportsToID	Contains the EmployeeID of the manager to whom an employee reports

Confidential information about the employees is stored in a separate table named EmployeeData. One record exists within EmployeeData for each record in the Employee table. You need to assign the appropriate constraints and table properties to ensure data integrity and visibility. On which column in the Employee table should you a create a unique constraint?

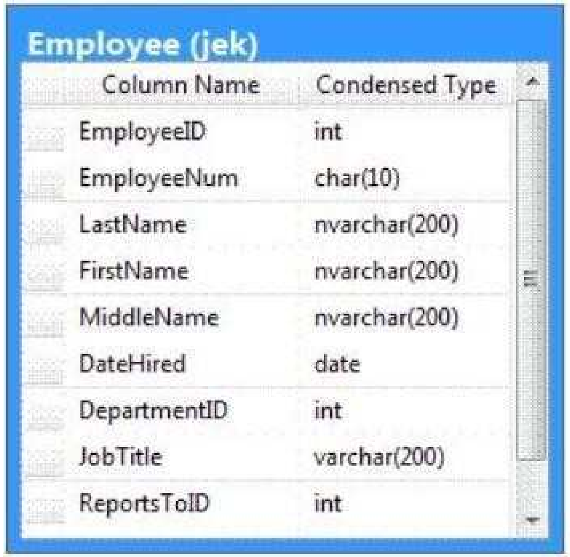
A. DateHired
B. DepartmentID
C. EmployeeID
D. EmployeeNum
E. FirstName
F. JobTitle

G. LastName
H. MiddleName
I. ReportsToID

Correct Answer: D
Section: (none)
Explanation

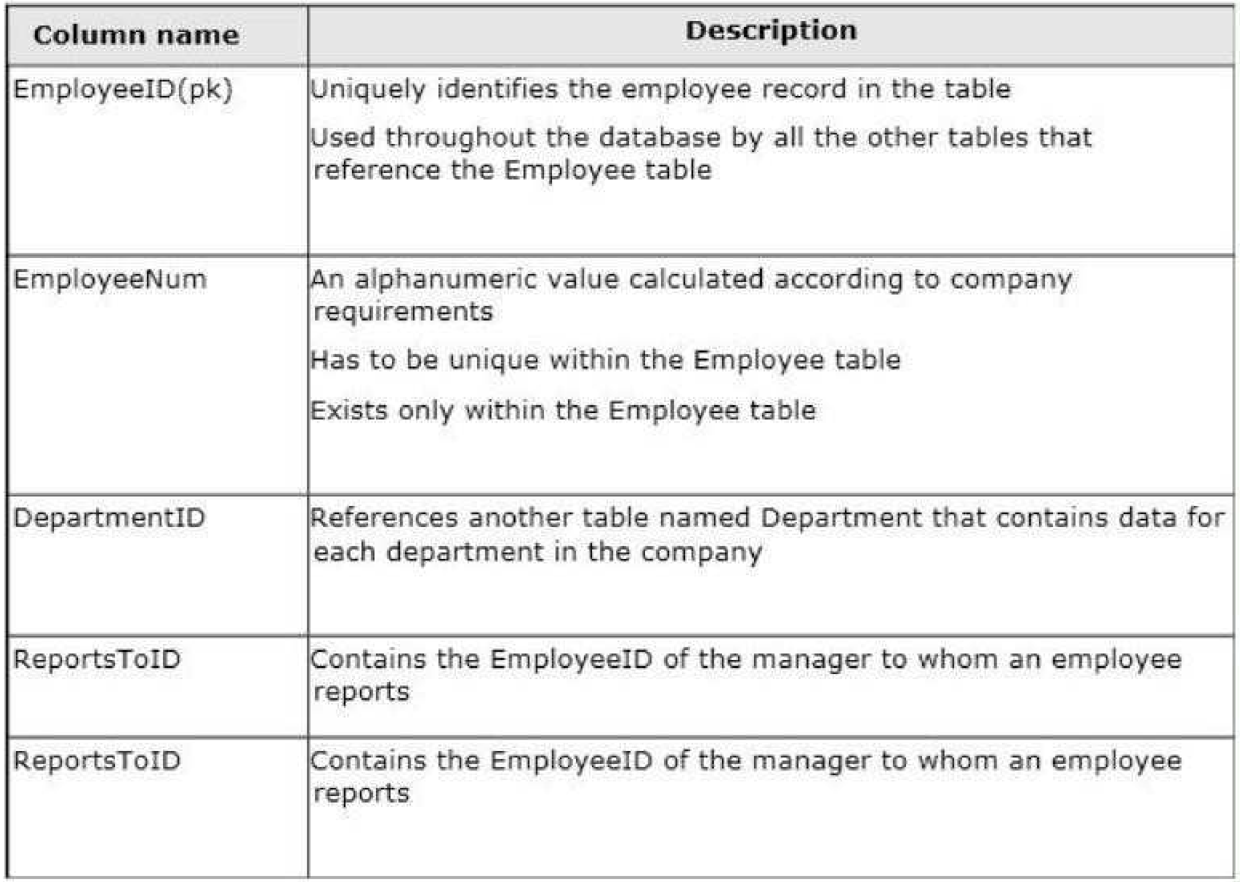
Explanation/Reference:

QUESTION 9
You administer a Microsoft SQL Server 2012 database. The database contains a table named Employee. Part of the Employee table is shown in the exhibit. (Click the Exhibit button.)



Employee (jek)

Column Name	Condensed Type
EmployeeID	int
EmployeeNum	char(10)
LastName	nvarchar(200)
FirstName	nvarchar(200)
MiddleName	nvarchar(200)
DateHired	date
DepartmentID	int
JobTitle	varchar(200)
ReportsToID	int



Column name	Description
EmployeeID(pk)	Uniquely identifies the employee record in the table Used throughout the database by all the other tables that reference the Employee table
EmployeeNum	An alphanumeric value calculated according to company requirements Has to be unique within the Employee table Exists only within the Employee table
DepartmentID	References another table named Department that contains data for each department in the company
ReportsToID	Contains the EmployeeID of the manager to whom an employee reports
ReportsToID	Contains the EmployeeID of the manager to whom an employee reports

Unless stated above, no columns in the Employee table reference other tables.

Confidential information about the employees is stored in a separate table named EmployeeData. One record exists within EmployeeData for each record in the Employee table.

You need to assign the appropriate constraints and table properties to ensure data integrity and visibility.
On which column in the Employee table should you use an identity specification to include a seed of 1,000 and an increment of 1?

A. DateHired
B. DepartmentID
C. EmployeeID

D. EmployeeNum
E. FirstName
F. JobTitle
G. LastName
H. MiddleName
I. ReportsToID

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 10
You administer a Microsoft SQL Server 2012 database that includes a table named Products. The Products table has columns named ProductId, ProductName, and CreatedDateTime.

The table contains a unique constraint on the combination of ProductName and CreatedDateTime.

You need to modify the Products table to meet the following requirements:

- Remove all duplicates of the Products table based on the ProductName column.
- Retain only the newest Products row.

Which Transact-SQL query should you use?

```
A. WITH CTEDupRecords
   AS
   (
   SELECT MAX(CreatedDateTime) AS CreatedDateTime, ProductName FROM Products
   GROUP BY ProductName
   HAVING COUNT(*) > 1
   )
   DELETE p
   FROM Products p
   JOIN CTEDupRecords cte ON
B. ProductName = cte.ProductName
   AND p.CreatedDateTime > cte.CreatedDateTime
C. WITH CTEDupRecords
   AS
   (
   SELECT MAX(CreatedDateTime) AS CreatedDateTime, ProductName FROM Products
   GROUP BY ProductName
   HAVING COUNT(*) > 1
   )
   DELETE p
   FROM Products p
   JOIN CTEDupRecords cte ON
   cte.ProductName = p.ProductName
   AND cte.CreatedDateTime > p.CreatedDateTime
D. WITH CTEDupRecords
   AS
   (
   SELECT MIN(CreatedDateTime) AS CreatedDateTime, ProductName FROM Products
   GROUP BY ProductName
   )
   DELETE p
   FROM Products p
   JOIN CTEDupRecords cte ON
E. ProductName = cte.ProductName
F. WITH CTEDupRecords
   AS
   (
   SELECT MAX(CreatedDateTime) AS CreatedDateTime, ProductName FROM Products
   GROUP BY ProductName
   HAVING COUNT(*) > 1
   )
   DELETE p
   FROM Products p
   JOIN CTEDupRecords cte ON
G. ProductName = cte.ProductName
```

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

QUESTION 11
You develop three Microsoft SQL Server 2012 databases named Database1, Database2, and Database3.

You have permissions on both Database1 and Database2. You plan to write and deploy a stored procedure named dbo.usp_InsertEvent in Database3. dbo.usp_InsertEvent must execute other stored procedures in the other databases.

You need to ensure that callers that do not have permissions on Database1 or Database2 can execute the stored procedure.

Which Transact-SQL statement should you use?

A. USE Database2
B. EXECUTE AS OWNER
C. USE Database1
D. EXECUTE AS CALLER

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms188354.aspx Reference: http://blog.sqlauthority.com/2007/10/06/sql-server-executing-remote-stored-procedure- callingstored-procedure-on-linked-server/

QUESTION 12
You administer a Microsoft SQL Server 2012 database that has multiple tables in the Sales schema. Some users must be prevented from deleting records in any of the tables in the Sales schema. You need to manage users who are prevented from deleting records in the Sales schema.

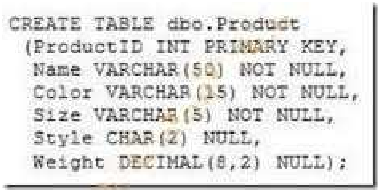
You need to achieve this goal by using the minimum amount of administrative effort. What should you do?

A. Create a custom database role that includes the users. Deny Delete permissions on the Sales schema for the custom database role.
B. Include the Sales schema as an owned schema for the db_denydatawriter role. Add the users to the db_denydatawriter role.
C. Deny Delete permissions on each table in the Sales schema for each user.
D. Create a custom database role that includes the users. Deny Delete permissions on each table in the Sales schema for the custom database role.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 13
You administer a Microsoft SQL Server 2012 database. The database contains a Product table created by using the following definition:



```
CREATE TABLE dbo.Product
 (ProductID INT PRIMARY KEY,
  Name VARCHAR(50) NOT NULL,
  Color VARCHAR(15) NOT NULL,
  Size VARCHAR(5) NOT NULL,
  Style CHAR(2) NULL,
  Weight DECIMAL(8,2) NULL);
```

You need to ensure that the minimum amount of disk space is used to store the data in the Product table.

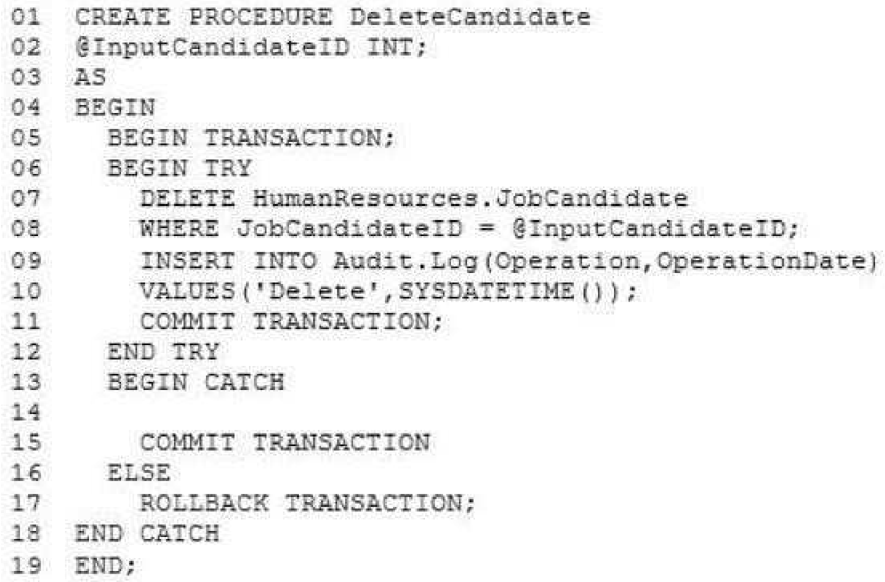
What should you do?

A. Convert all indexes to Column Store indexes.
B. Implement Unicode Compression.
C. Implement row-level compression.
D. Implement page-level compression.

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/cc280449.aspx Reference: http://msdn.microsoft.com/en-us/library/cc280464.aspx Reference: http://msdn.microsoft.com/en-us/library/cc280576.aspx Reference: http://msdn.microsoft.com/en-us/library/ee240835.aspx

QUESTION 14
You use Microsoft SQL Server 2012 to create a stored procedure as shown in the following code segment. (Line numbers are included for reference only.)



```
01  CREATE PROCEDURE DeleteCandidate
02  @InputCandidateID INT;
03  AS
04  BEGIN
05    BEGIN TRANSACTION;
06    BEGIN TRY
07      DELETE HumanResources.JobCandidate
08      WHERE JobCandidateID = @InputCandidateID;
09      INSERT INTO Audit.Log(Operation,OperationDate)
10      VALUES('Delete',SYSDATETIME());
11      COMMIT TRANSACTION;
12    END TRY
13    BEGIN CATCH
14
15      COMMIT TRANSACTION
16    ELSE
17      ROLLBACK TRANSACTION;
18  END CATCH
19  END;
```

The procedure can be called within other transactions.

You need to ensure that when the DELETE statement from the HumanResourcesJobCandidate table succeeds, the modification is retained even if the insert into the Audit.Log table fails.

Which code segment should you add to line 14?

A. IF @@TRANCOUNT = 0

B. IF (XACT_STATE ( )) = 0

C. IF (XACT_STATE ( )) = 1

D. IF @@TRANCOUNT = l

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms189797.aspx Reference: http://msdn.microsoft.com/en-us/library/ms187967.aspx

QUESTION 15
A table named Profits stores the total profit made each year within a territory. The Profits table has columns named Territory, Year, and Profit. You need to create a report that displays the profits made by each territory for each year and its preceding year. Which Transact-SQL query should you use?

A. SELECT Territory, Year, Profit,
LAG(Profit, 1, 0) OVER(PARTITION BY Year ORDER BY Territory) AS NextProfit FROM Profits
B. SELECT Territory, Year, Profit,
LAG(Profit, 1, 0) OVER(PARTITION BY Territory ORDER BY Year) AS NextProfit FROM Profits
C. SELECT Territory, Year, Profit,
LEAD(Profit, 1, 0) OVER(PARTITION BY Territory ORDER BY Year) AS NextProfit FROM Profits
D. SELECT Territory, Year, Profit,
LEAD(Profit, 1, 0) OVER(PARTITION BY Year ORDER BY Territory) AS NextProfit FROM Profits

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/hh231256.aspx Reference: http://msdn.microsoft.com/en-us/library/hh213125.aspx

QUESTION 16
You use Microsoft SQL Server 2012 to develop a database application.

Your application sends data to an NVARCHAR(MAX) variable named @var.

You need to write a Transact-SQL statement that will find out the success of a cast to a decimal (36,9).

Which code segment should you use?

A. BEGIN TRY
SELECT convert(decimal(36,9), @var) AS Value, 'True' AS BadCast END TRY
BEGIN CATCH
SELECT convert(decimal(36,9), @var) AS Value, 'False' AS BadCast END CATCH
B. TRY(
SELECT convert(decimal(36,9), @var)
SELECT 'True' AS BadCast
)
CATCH(
SELECT 'False' AS BadCast
)

C. SELECT

```
CASE
WHEN convert(decimal(36,9), @var) IS NULL
THEN 'True'
ELSE 'False'
END
AS BadCast
```

D. SELECT

```
IIF(TRY_PARSE(@var AS decimal(36,9)) IS NULL, 'True', 'False') AS BadCast
```

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/hh213126.aspx

QUESTION 17
You are writing a set of queries against a FILESTREAM-enabled database.

You create a stored procedure that will update multiple tables within a transaction.

You need to ensure that if the stored procedure raises a runtime error, the entire transaction is terminated and rolled back.

Which Transact-SQL statement should you include at the beginning of the stored procedure?

A. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
B. SET XACT_ABORT OFF
C. SET TRANSACTION ISOLATION LEVEL SNAPSHOT
D. SET IMPLICIT_TRANSACTIONS ON
E. SET XACT_ABORT ON
F. SET IMPLICIT TRANSACTIONS OFF

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms188792.aspx

QUESTION 18
You develop a Microsoft SQL Server 2012 database. The database is used by two web applications that access a table named Products.

You want to create an object that will prevent the applications from accessing the table directly while still providing access to the required data.

You need to ensure that the following requirements are met:

- Future modifications to the table definition will not affect the applications' ability to access data.
- The new object can accommodate data retrieval and data modification.

You need to achieve this goal by using the minimum amount of changes to the applications.

What should you create for each application?

A. Synonyms
B. Common table expressions
C. Views
D. Temporary tables

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms190174.aspx

QUESTION 19
You are developing a database application by using Microsoft SQL Server 2012.

An application that uses a database begins to run slowly.

Your investigation shows the root cause is a query against a read-only table that has a clustered index.

The query returns the following six columns:

- One column in its WHERE clause contained in a non-clustered index
- Four additional columns
- One COUNT (*) column based on a grouping of the four additional columns

You need to optimize the statement.

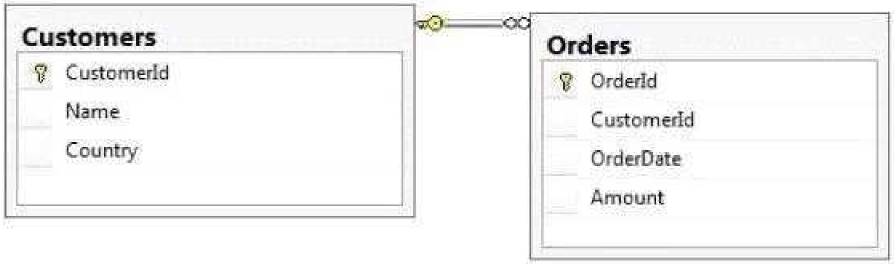
What should you do?

A. Add a HASH hint to the query.
B. Add a LOOP hint to the query.
C. Add a FORCESEEK hint to the query.
D. Add an INCLUDE clause to the index.
E. Add a FORCESCAN hint to the Attach query.
F. Add a columnstore index to cover the query.
G. Enable the optimize for ad hoc workloads option.
H. Cover the unique clustered index with a columnstore index.
I. Include a SET FORCEPLAN ON statement before you run the query.
J. Include a SET STATISTICS PROFILE ON statement before you run the query.
K. Include a SET STATISTICS SHOWPLAN_XML ON statement before you run the query.
L. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query.
M. Include a SET TRANSACTION ISOLATION LEVEL SNAPSHOT statement before you run the query.
N. Include a SET TRANSACTION ISOLATION LEVEL SERIALIZABLE statement before you run the query.

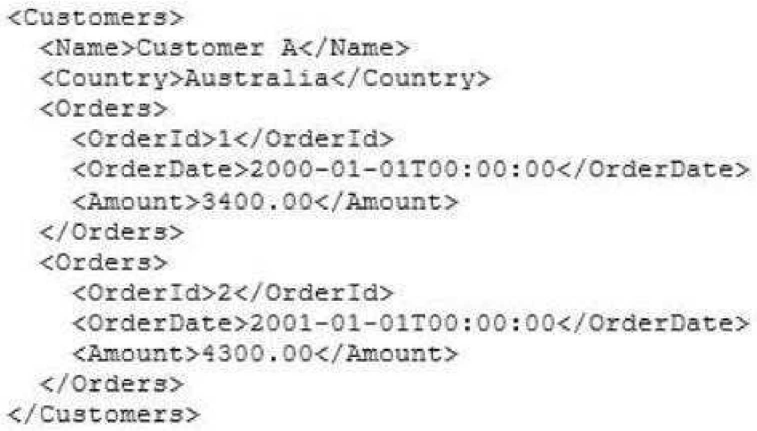
Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:

QUESTION 20
You administer a Microsoft SQL Server 2012 database named ContosoDb. Tables are defined as shown in the exhibit. (Click the Exhibit button.)



You need to display rows from the Orders table for the Customers row having the CustomerId value set to 1 in the following XML format.



```
<Customers>
  <Name>Customer A</Name>
  <Country>Australia</Country>
  <Orders>
    <OrderId>1</OrderId>
    <OrderDate>2000-01-01T00:00:00</OrderDate>
    <Amount>3400.00</Amount>
  </Orders>
  <Orders>
    <OrderId>2</OrderId>
    <OrderDate>2001-01-01T00:00:00</OrderDate>
    <Amount>4300.00</Amount>
  </Orders>
</Customers>
```

Which Transact-SQL query should you use?

A. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers-CustomerId WHERE Customers.CustomerId = 1
FOR XML RAW

B. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers.CustomerId WHERE Customers=CustomerId = 1
FOR XML RAW, ELEMENTS

C. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers.CustomerId WHERE Customers.CustomerId = 1
FOR XML AUTO

D. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId - Customers.CustomerId WHERE Customers.CustomerId= 1

FOR XML AUTO, ELEMENTS

E. SELECT Name, Country, OrderId, OrderDate, Amount FROM Orders INNER JOIN Customers ON Orders.CustomerId= Customers.CustomerId WHERE Customers.CustomerId= 1
FOR XML AUTO

F. SELECT Name, Country, OrderId, OrderDate, Amount FROM Orders INNER JOIN Customers ON Orders.CustomerId= Customers.CustomerId WHERE Customers.CustomerId= 1
FOR XML AUTO, ELEMENTS

G. SELECT Name AS '@Name', Country AS '@Country', OrderId, OrderDate, Amount FROM Orders INNER JOIN Customers ON Orders.CustomerId= Customers.CustomerId WHERE Customers.CustomerId= 1
FOR XML PATH ('Customers')

H. SELECT Name AS 'Customers/Name', Country AS 'Customers/Country', OrderId, OrderDate, Amount FROM Orders
INNER JOIN Customers ON Orders.CustomerId= Customers.CustomerId WHERE Customers.CustomerId= 1
FOR XML PATH ('Customers')

Correct Answer: F
Section: (none)
Explanation

Explanation/Reference:

QUESTION 21

You use Microsoft SQL Server 2012 to develop a database application.

You need to implement a computed column that references a lookup table by using an INNER JOIN against another table.

What should you do?

A. Reference a user-defined function within the computed column.
B. Create a BEFORE trigger that maintains the state of the computed column.
C. Add a default constraint to the computed column that implements hard-coded values.
D. Add a default constraint to the computed column that implements hard-coded CASE statements.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 22
You administer a Microsoft SQL Server 2012 database named ContosoDb. The database contains a table named Suppliers and a column named IsActive in the Purchases schema. You create a new user named ContosoUser in ContosoDb. ContosoUser has no permissions to the Suppliers table. You need to ensure that ContosoUser can delete rows that are not active from Suppliers. You also need to grant ContosoUser only the minimum required permissions. Which Transact-SQL statement should you use?

```
A. GRANT DELETE ON Purchases.Suppliers TO ContosoUser
B. CREATE PROCEDURE Purchases.PurgeInactiveSuppliers
   WITH EXECUTE AS USER = 'dbo'
   AS
   DELETE FROM Purchases.Suppliers WHERE IsActive = 0
   GO
   GRANT EXECUTE ON Purchases.PurgeInactiveSuppliers TO ContosoUser
C. GRANT SELECT ON Purchases.Suppliers TO ContosoUser
D. CREATE PROCEDURE Purchases.PurgeInactiveSuppliers
   AS
   DELETE FROM Purchases.Suppliers WHERE IsActive = 0
   GO
   GRANT EXECUTE ON Purchases.PurgeInactiveSuppliers TO ContosoUser
```

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms188354.aspx Reference: http://msdn.microsoft.com/en-us/library/ms187926.aspx

QUESTION 23
You administer a Microsoft SQL Server 2012 server. You plan to deploy new features to an application. You need to evaluate existing and potential clustered and non-clustered indexes that will improve performance.

What should you do?

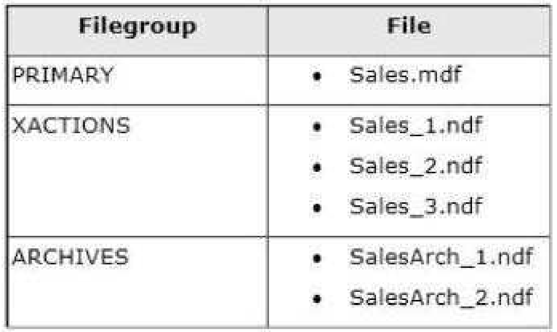
A. Query the sys.dm_db_index_usage_stats DMV.

B. Query the sys.dm_db_missing_index_details DMV.
C. Use the Database Engine Tuning Advisor.
D. Query the sys.dm_db_missing_index_columns DMV.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 24
You administer a Microsoft SQL Server database named Sales. The database is 3 terabytes in size. The Sales database is configured as shown in the following table.



Filegroup	File
PRIMARY	• Sales.mdf
XACTIONS	• Sales_1.ndf • Sales_2.ndf • Sales_3.ndf
ARCHIVES	• SalesArch_1.ndf • SalesArch_2.ndf

You discover that Sales_2.ndf is corrupt. You need to recover the corrupted data in the minimum amount of time. What should you do?

A. Perform a file restore.
B. Perform a transaction log restore.
C. Perform a restore from a full backup.
D. Perform a filegroup restore.

Correct Answer: A
Section: (none)
Explanation

Explanation/Reference:

QUESTION 25
Your database contains a table named SalesOrders. The table includes a DATETIME column named OrderTime that stores the date and time each order is placed. There is a non-clustered index on the OrderTime column.

The business team wants a report that displays the total number of orders placed on the current day.

You need to write a query that will return the correct results in the most efficient manner.

Which Transact-SQL query should you use?

A. SELECT COUNT(*) FROM SalesOrders
WHERE OrderTime = CONVERT(DATE, GETDATE())

B. SELECT COUNT(*) FROM SalesOrders
WHERE OrderTime = GETDATE()

C. SELECT COUNT(*) FROM SalesOrders
WHERE CONVERT(VARCHAR, OrderTime, 112) = CONVERT(VARCHAR, GETDATE(I, 112))

D. SELECT COUNT(*) FROM SalesOrders
WHERE OrderTime >= CONVERT(DATE, GETDATE())
AND OrderTime <DATEADD(DAY, 1, CONVERT(DATE, GETDATE()))

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 26
Your application contains a stored procedure for each country. Each stored procedure accepts an employee identification number through the @EmpID parameter.

You plan to build a single process for each employee that will execute the stored procedure based on the country of residence.

Which approach should you use?

A. A recursive stored procedure
B. Trigger
C. An UPDATE statement that includes CASE
D. Cursor
E. The foreach SQLCLR statement

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 27
You use Microsoft SQL Server 2012 to develop a database application.

You create a stored procedure named dbo.ModifyData that can modify rows.

You need to ensure that when the transaction fails, dbo.ModifyData meets the following requirements:

- Does not return an error
- Closes all opened transactions

Which Transact-SQL statement should you use?

```
A. BEGIN TRANSACTION
   BEGIN TRY
   EXEC dbo.ModifyData
   COMMIT TRANSACTION
   END TRY
   BEGIN CATCH
   IF @@ TRANCOUNT = 0
   ROLLBACK TRANSACTION;
   END CATCH
```

```
B. BEGIN TRANSACTION
   BEGIN TRY
   EXEC dbo.ModifyData
   COMMIT TRANSACTION
   END TRY
   BEGIN CATCH
   IF @@ERROR != 0
   ROLLBACK TRANSACTION;
   THROW;
   END CATCH
```

```
C. BEGIN TRANSACTION
   BEGIN TRY
   EXEC dbo.ModifyData
   COMMIT TRANSACTION
   END TRY
   BEGIN CATCH
   IF @@TRANCOUNT = 0
   ROLLBACK TRANSACTION;
   THROW;
   END CATCH
D. BEGIN TRANSACTION
   BEGIN TRY
   EXEC dbo.ModifyData
   COMMIT TRANSACTION
   END TRY
   BEGIN CATCH
   IF @@ERROR != 0
   ROLLBACK TRANSACTION;
   END CATCH
```

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 28
You are developing a database application by using Microsoft SQL Server 2012.

An application that uses a database begins to run slowly. You discover that during reads, the transaction experiences blocking from concurrent updates.

You need to ensure that throughout the transaction the data maintains the original version.

What should you do?

A. Add a HASH hint to the query.
B. Add a LOOP hint to the query.
C. Add a FORCESEEK hint to the query.
D. Add an INCLUDE clause to the index.
E. Add a FORCESCAN hint to the Attach query.
F. Add a columnstore index to cover the query.
G. Enable the optimize for ad hoc workloads option.
H. Cover the unique clustered index with a columnstore index.

I. Include a SET FORCEPLAN ON statement before you run the query.
J. Include a SET STATISTICS PROFILE ON statement before you run the query.
K. Include a SET STATISTICS SHOWPLAN_XML ON statement before you run the query.
L. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query.
M. Include a SET TRANSACTION ISOLATION LEVEL SNAPSHOT statement before you run the query.
N. Include a SET TRANSACTION ISOLATION LEVEL SERIALIZABLE statement before you run the query.

Correct Answer: M
Section: (none)
Explanation

Explanation/Reference:

QUESTION 29
You are developing a database application by using Microsoft SQL Server 2012.

You have a query that runs slower than expected.

You need to capture execution plans that will include detailed information on missing indexes recommended by the query optimizer.

What should you do?

A. Add a HASH hint to the query.
B. Add a LOOP hint to the query.
C. Add a FORCESEEK hint to the query.
D. Add an INCLUDE clause to the index.
E. Add a FORCESCAN hint to the Attach query.
F. Add a columnstore index to cover the query.
G. Enable the optimize for ad hoc workloads option.
H. Cover the unique clustered index with a columnstore index.
I. Include a SET FORCEPLAN ON statement before you run the query.
J. Include a SET STATISTICS PROFILE ON statement before you run the query.
K. Include a SET STATISTICS SHOWPLAN_XML ON statement before you run the query.
L. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query.
M. Include a SET TRANSACTION ISOLATION LEVEL SNAPSHOT statement before you run the query.
N. Include a SET TRANSACTION ISOLATION LEVEL SERIALIZABLE statement before you run the query.

Correct Answer: K
Section: (none)
Explanation

Explanation/Reference:

QUESTION 30
You are developing a database application by using Microsoft SQL Server 2012.

An application that uses a database begins to run slowly.

You discover that a large amount of memory is consumed by single-use dynamic queries.

You need to reduce procedure cache usage from these statements without creating any additional indexes.

What should you do?

A. Add a HASH hint to the query.
B. Add a LOOP hint to the query.
C. Add a FORCESEEK hint to the query.
D. Add an INCLUDE clause to the index.
E. Add a FORCESCAN hint to the Attach query.
F. Add a columnstore index to cover the query.
G. Enable the optimize for ad hoc workloads option.
H. Cover the unique clustered index with a columnstore index.
I. Include a SET FORCEPLAN ON statement before you run the query.
J. Include a SET STATISTICS PROFILE ON statement before you run the query.
K. Include a SET STATISTICS SHOWPLAN_XML ON statement before you run the query.
L. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query.
M. Include a SET TRANSACTION ISOLATION LEVEL SNAPSHOT statement before you run the query.
N. Include a SET TRANSACTION ISOLATION LEVEL SERIALIZABLE statement before you run the query.

Correct Answer: G
Section: (none)
Explanation

Explanation/Reference:
Explanation:

Reference: http://msdn.microsoft.com/en-us/library/cc645587.aspx

QUESTION 31
You use Microsoft SQL Server 2012 database to develop a shopping cart application.

You need to invoke a table-valued function for each row returned by a query.

Which Transact-SQL operator should you use?

A. CROSS JOIN
B. UNPIVOT
C. PIVOT
D. CROSS APPLY

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:
Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms175156.aspx

QUESTION 32
You develop a Microsoft SQL Server 2012 database. The database is used by two web applications that access a table named Products.

You want to create an object that will prevent the applications from accessing the table directly while still providing access to the required data.

You need to ensure that the following requirements are met:

- Future modifications to the table definition will not affect the applications' ability to access data.
- The new object can accommodate data retrieval and data modification. You need to achieve this goal by using the minimum amount of changes to the existing applications.

What should you create for each application?

A. table partitions

B. views
C. table-valued functions
D. stored procedures

Correct Answer: B
Section: (none)
Explanation

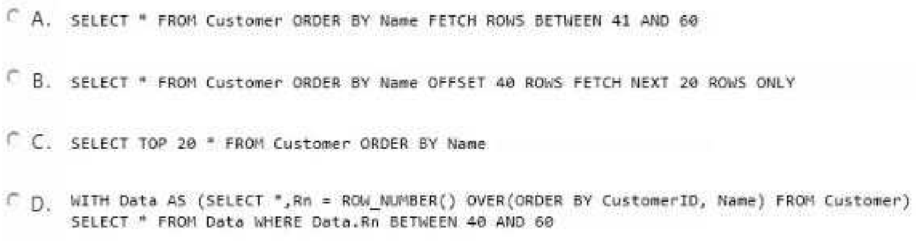
Explanation/Reference:

QUESTION 33
Your database contains a table named Customer that has columns named CustomerID and Name.

You want to write a query that retrieves data from the Customer table sorted by Name listing 20 rows at a time.

You need to view rows 41 through 60.

Which Transact-SQL query should you create?



○ A. SELECT * FROM Customer ORDER BY Name FETCH ROWS BETWEEN 41 AND 60

○ B. SELECT * FROM Customer ORDER BY Name OFFSET 40 ROWS FETCH NEXT 20 ROWS ONLY

○ C. SELECT TOP 20 * FROM Customer ORDER BY Name

○ D.
```
WITH Data AS (SELECT *,Rn = ROW_NUMBER() OVER(ORDER BY CustomerID, Name) FROM Customer)
SELECT * FROM Data WHERE Data.Rn BETWEEN 40 AND 60
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: B
Section: (none)
Explanation

Explanation/Reference:

QUESTION 34
You are developing a database that will contain price information.

You need to store the prices that include a fixed precision and a scale of six digits.

Which data type should you use?

A. Real
B. Small money
C. Money
D. Decimal

Correct Answer: D
Section: (none)
Explanation

Explanation/Reference:

QUESTION 35
You develop a database application. You create four tables. Each table stores different categories of products.

You create a Primary Key field on each table.

You need to ensure that the following requirements are met:

- The fields must use the minimum amount of space.
- The fields must be an incrementing series of values.
- The values must be unique among the four tables.

What should you do?

A. Create a ROWVERSION column.
B. Create a SEQUENCE object that uses the INTEGER data type.
C. Use the INTEGER data type along with IDENTITY
D. Use the UNIQUHDENTTFIER data type along with NEWSEQUENTIALID()
E. Create a TIMESTAMP column.

Correct Answer: D
Section: (none)

Explanation

Explanation/Reference:

QUESTION 36
Your application contains a stored procedure for each country. Each stored procedure accepts an employee identification number through the @EmpID parameter. You need to build a single process for each employee that will execute the appropriate stored procedure based on the country of residence.

Which approach should you use?

A. A SELECT statement that includes CASE
B. Cursor
C. BULK INSERT
D. View
E. A user-defined function

Correct Answer: E
Section: (none)
Explanation

Explanation/Reference:
SQL Server user-defined functions are routines that accept parameters, perform an action, such as a complex calculation, and return the result of that action as a value. The return value can either be a single scalar value or a result set.

QUESTION 37
You are developing a database application by using Microsoft SQL Server 2012.

An application that uses a database begins to run slowly.

You discover that the root cause is a query against a frequently updated table that has a clustered index. The query returns four columns: three columns in its WHERE clause contained in a non- clustered index and one additional column.

You need to optimize the statement.

What should you do?

A. Add a HASH hint lo the query.
B. Add a LOOP hint to the query.
C. Add a FORCESEEK hint to the query.

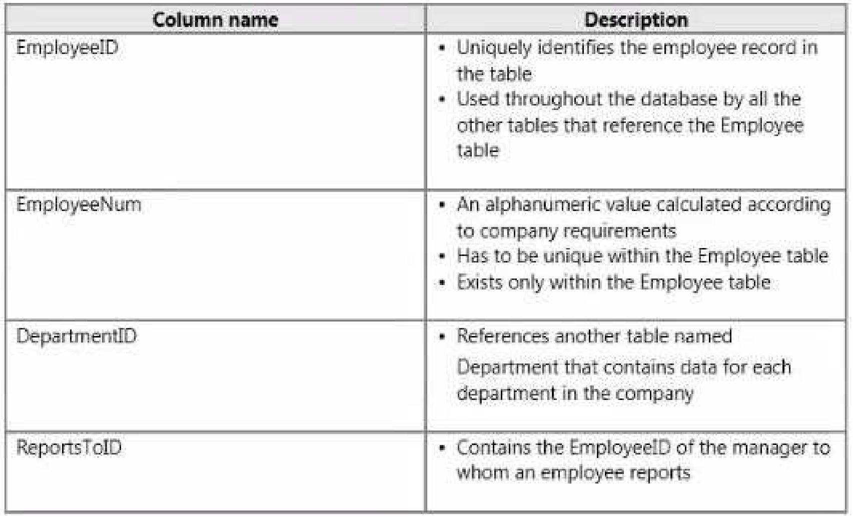
D. Add an INCLUDE clause to the index.
E. Add a FORCESCAN hint to the Attach query.
F. Add a FORCESCAN hint to the Attach query.
G. Add a columnstore index to cover the query.
H. Enable theoptimize for ad hoc workloads option.
I. Cover the unique clustered index with a columnstore index. Include a SET FORCEPLAN ON statement before you run the query.
J. Include a SET STATISTICS PROFILE ON statement before you run the query.
K. Include a SET STATISTICS SHOWPLAN.XML ON statement before you run the query.
L. Include a SET TRANSACTION ISOLATION LEVEL REPEATABLE READ statement before you run the query.
M. Include a SET TRANSACTION ISOLATION LEVEL SNAPSHOT statement before you run the query.
N. Include a SET TRANSACTION ISOLATION LEVEL SERIALIZABLE statement before you run the query.

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 38
You administer a Microsoft SQL Server 2012 database.

The database contains a table named Employee. Part of the Employee table is shown in the exhibit. (Click the Exhibit button.)



Column name	Description
EmployeeID	• Uniquely identifies the employee record in the table • Used throughout the database by all the other tables that reference the Employee table
EmployeeNum	• An alphanumeric value calculated according to company requirements • Has to be unique within the Employee table • Exists only within the Employee table
DepartmentID	• References another table named Department that contains data for each department in the company
ReportsToID	• Contains the EmployeeID of the manager to whom an employee reports

Unless stated above, no columns in the Employee table reference other tables. Confidential information about the employees is stored in a separate table named EmployeeData. One record exists within EmployeeData for each record in the Employee table.

You need to assign the appropriate constraints and table properties to ensure data integrity and visibility.

On which column in the Employee table should you create a Foreign Key constraint that references a different table in the database?

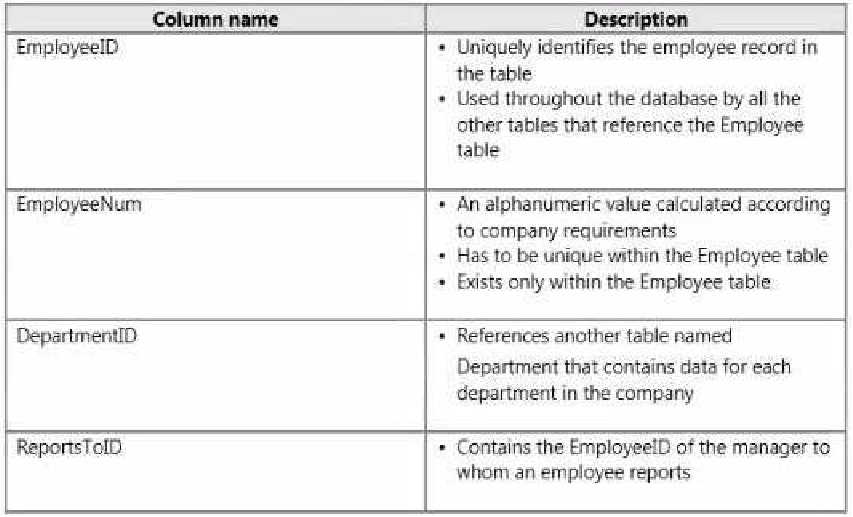
A. DateHired
B. Departments
C. EmployeeID
D. EmployeeNum
E. FirstName
F. JobTitle
G. LastName
H. MiddleName
I. ReportsToID
C

Correct Answer:
Section: (none)
Explanation

Explanation/Reference:
Use the EmployeeID, which would be used as a primary key in the Employee table, when defining a foreign key constraint from another table in the database.

QUESTION 39
You administer a Microsoft SQL Server 2012 database.

The database contains a table named Employee. Part of the Employee table is shown in the exhibit. (Click the Exhibit button.)



Column name	Description
EmployeeID	• Uniquely identifies the employee record in the table • Used throughout the database by all the other tables that reference the Employee table
EmployeeNum	• An alphanumeric value calculated according to company requirements • Has to be unique within the Employee table • Exists only within the Employee table
DepartmentID	• References another table named Department that contains data for each department in the company
ReportsToID	• Contains the EmployeeID of the manager to whom an employee reports

Unless stated above, no columns in the Employee table reference other tables.

Confidential information about the employees is stored in a separate table named EmployeeData. One record exists within EmployeeData for each record in the Employee table.

You need to assign the appropriate constraints and table properties to ensure data integrity and visibility.

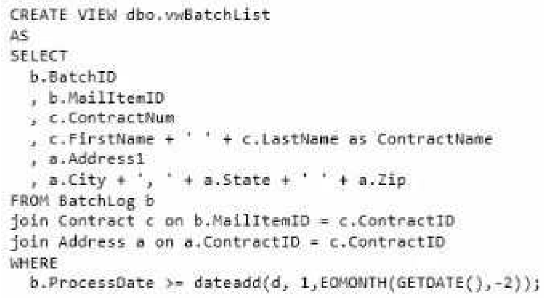
On which column in the Employee table should you create a Primary Key constraint for this table?

A. DateHired
B. Departments
C. EmployeeID
D. EmployeeNum
E. FirstName
F. JobTitle
G. LastName
H. MiddleName
I. ReportsToID

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 40
You create a view based on the following statement:

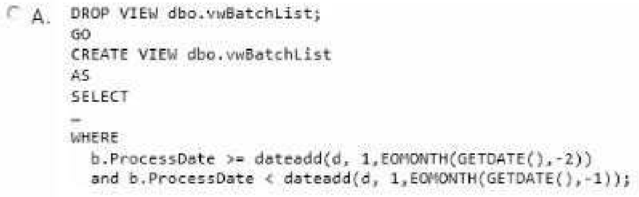


```
CREATE VIEW dbo.vwBatchList
AS
SELECT
  b.BatchID
  , b.MailItemID
  , c.ContractNum
  , c.FirstName + ' ' + c.LastName as ContractName
  , a.Address1
  , a.City + ', ' + a.State + ' ' + a.Zip
FROM BatchLog b
join Contract c on b.MailItemID = c.ContractID
join Address a on a.ContractID = c.ContractID
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2));
```

You grant the Select permission to User1.

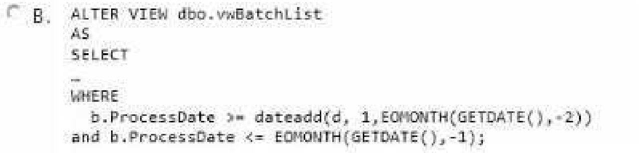
You need to change the view so that it displays only the records that were processed in the month prior to the current month. You need to ensure that after the changes, the view functions correctly for User1.

Which Transact-SQL statement should you use?



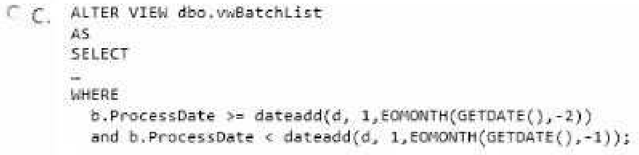
○ A.

```
DROP VIEW dbo.vwBatchList;
GO
CREATE VIEW dbo.vwBatchList
AS
SELECT
…
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2))
  and b.ProcessDate < dateadd(d, 1,EOMONTH(GETDATE(),-1));
```



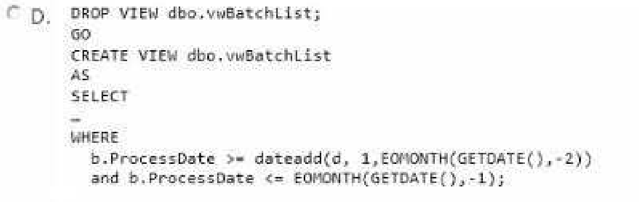
○ B.

```
ALTER VIEW dbo.vwBatchList
AS
SELECT
…
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2))
and b.ProcessDate <= EOMONTH(GETDATE(),-1);
```



○ C.

```
ALTER VIEW dbo.vwBatchList
AS
SELECT
…
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2))
  and b.ProcessDate < dateadd(d, 1,EOMONTH(GETDATE(),-1));
```



○ D.

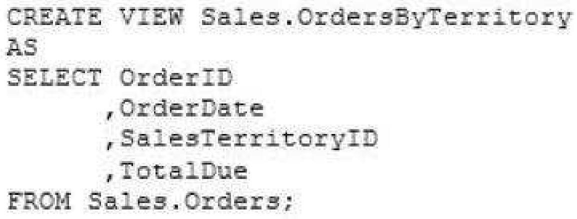
```
DROP VIEW dbo.vwBatchList;
GO
CREATE VIEW dbo.vwBatchList
AS
SELECT
…
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2))
  and b.ProcessDate <= EOMONTH(GETDATE(),-1);
```

A. Option A
B. Option B
C. Option C
D. Option D

Correct Answer: C
Section: (none)
Explanation

Explanation/Reference:

QUESTION 41
You have a view that was created by using the following code:



```
CREATE VIEW Sales.OrdersByTerritory
AS
SELECT OrderID
      ,OrderDate
      ,SalesTerritoryID
      ,TotalDue
FROM Sales.Orders;
```

You need to create an inline table-valued function named Sales.fn_OrdersByTerritory, which must meet the following requirements:

- Accept the @T integer parameter.
- Use one-part names to reference columns.
- Filter the query results by SalesTerritoryID.
- Return the columns in the same order as the order used in OrdersByTerritoryView.

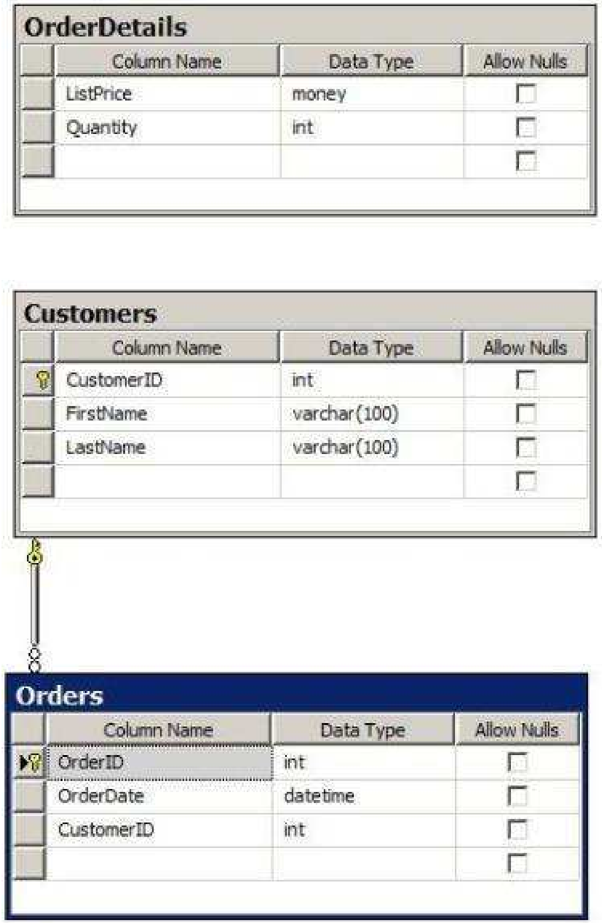
Which code segment should you use? To answer, type the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE FUNCTION Sales.fn_OrdersByTerritory (@T int) RETURNS TABLE AS RETURN (SELECT OrderID,OrderDate,SalesTerrirotyID,TotalDue FROM Sales.OrdersByTerritory WHERE SalesTerritoryID = @T)

QUESTION 42
You have a database that contains the tables shown in the exhibit. (Click the Exhibit button.)



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	□
Quantity	int	□
		□

Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	□
FirstName	varchar(100)	□
LastName	varchar(100)	□
		□

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	□
OrderDate	datetime	□
CustomerID	int	□
		□

You deploy a new server that has SQL Server 2012 installed. You need to create a table named Sales.OrderDetails on the new server. Sales.OrderDetails must meet the following requirements:

- Write the results to a disk.
- Contain a new column named LineItemTotal that stores the product of ListPrice and Quantity for each row.
- The code must NOT use any object delimiters.

The solution must ensure that LineItemTotal is stored as the last column in the table. Which code segment should you use? To answer, type the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer

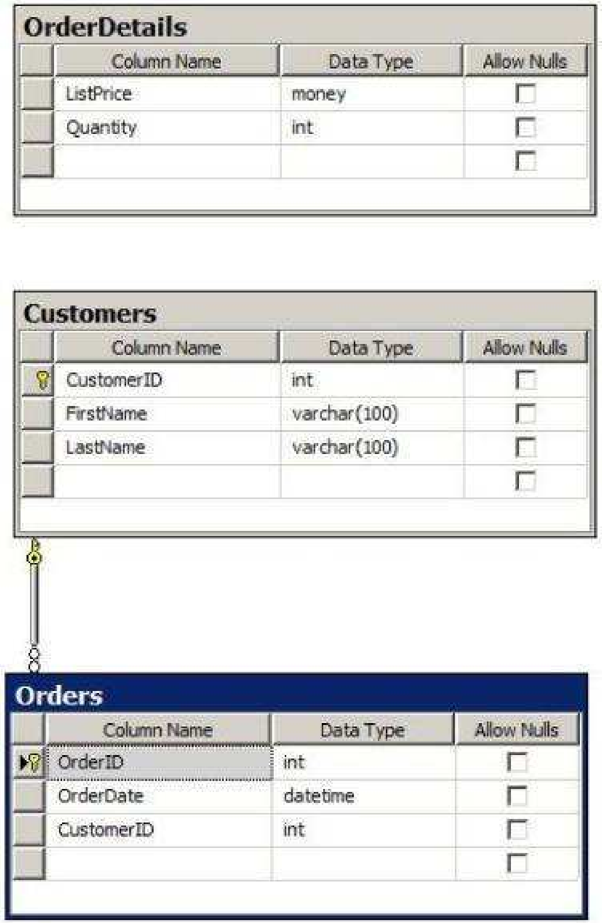
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE TABLE Sales.OrderDetails (ListPrice money not null, Quantity int not null, LineItemTotal as (ListPrice * Quantity) PERSISTED) Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms174979.aspx
Reference: http://technet.microsoft.com/en-us/library/ms188300.aspx

QUESTION 43

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



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	□
Quantity	int	□
		□

Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	□
FirstName	varchar(100)	□
LastName	varchar(100)	□
		□

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	□
OrderDate	datetime	□
CustomerID	int	□
		□

You need to create a view named uv_CustomerFullName to meet the following requirements:

- The code must NOT include object delimiters.
- The view must be created in the Sales schema.
- Columns must only be referenced by using one-part names.
- The view must return the first name and the last name of all customers.
- The view must prevent the underlying structure of the customer table from being changed.
- The view must be able to resolve all referenced objects, regardless of the user's default schema.

Which code segment should you use?
To answer, type the correct code in the answer area.

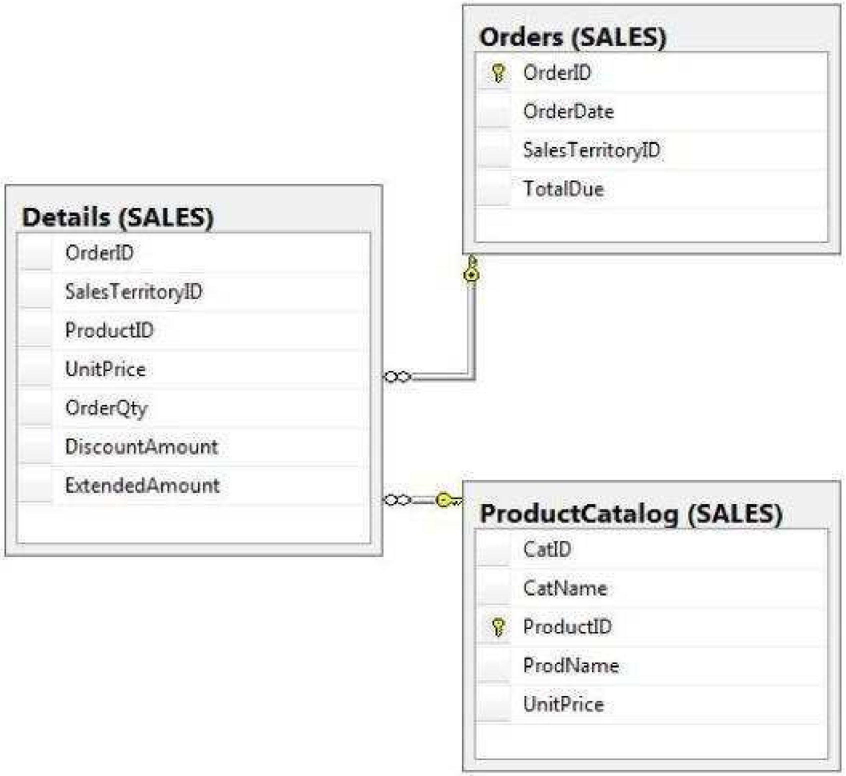
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE VIEW Sales.uv_CustomerFullName WITH SCHEMABINDING AS SELECT FirstName, LastName FROM Sales.Customers Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms187956.aspx

QUESTION 44

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



You need to create a query that calculates the total sales of each OrderId from the Sales.Details table. The solution must meet the following requirements:

- Use one-part names to reference columns.
- Sort the order of the results from OrderId.

- NOT depend on the default schema of a user.
- Use an alias of TotalSales for the calculated ExtendedAmount.
- Display only the OrderId column and the calculated TotalSales column.

Which code segment should you use? To answer, type the correct code in the answer area.

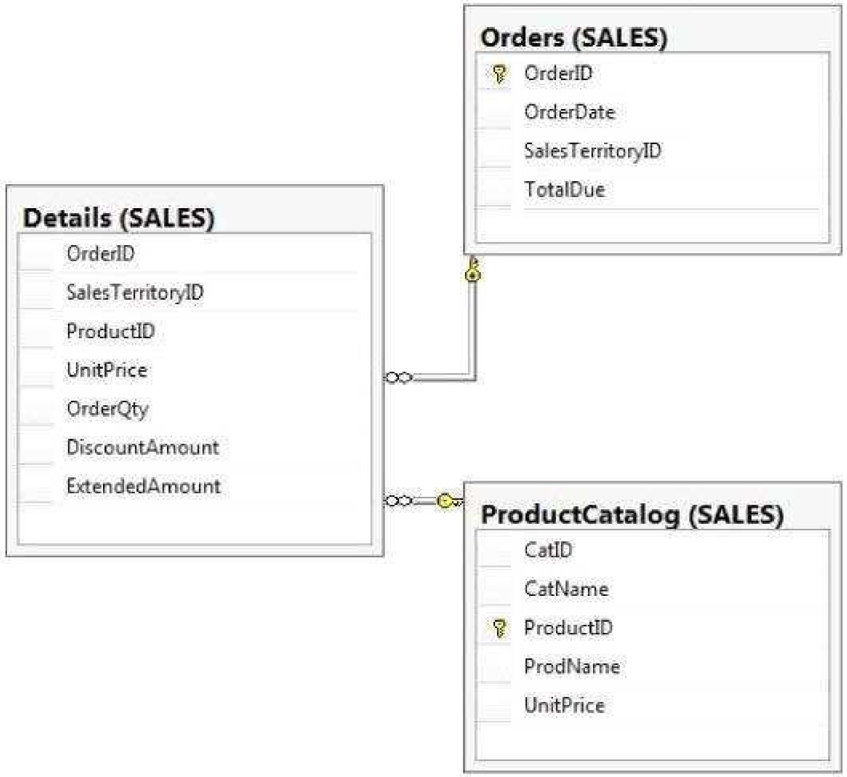
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

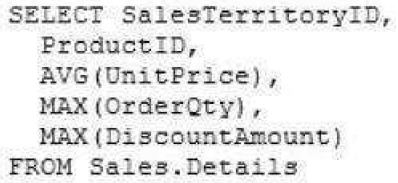
Explanation: SELECT OrderID, SUM(ExtendedAmount) AS TotalSales FROM Sales.Details GROUP BY OrderID ORDER BY OrderID

QUESTION 45

You have a database that contains the tables as shown in the exhibit. (Click the Exhibit button.)



You have the following query:



```
SELECT SalesTerritoryID,
   ProductID,
   AVG(UnitPrice),
   MAX(OrderQty),
   MAX(DiscountAmount)
FROM Sales.Details
```

You need to recreate the query to meet the following requirements:

- Reference columns by using one-part names only.
- Sort aggregates by SalesTerritoryID, and then by ProductID.
- Order the results in descending order from SalesTerritoryID to ProductID.
- The solution must use the existing SELECT clause and FROM clause.

Which code segment should you use? To answer, type the correct code in the answer area.

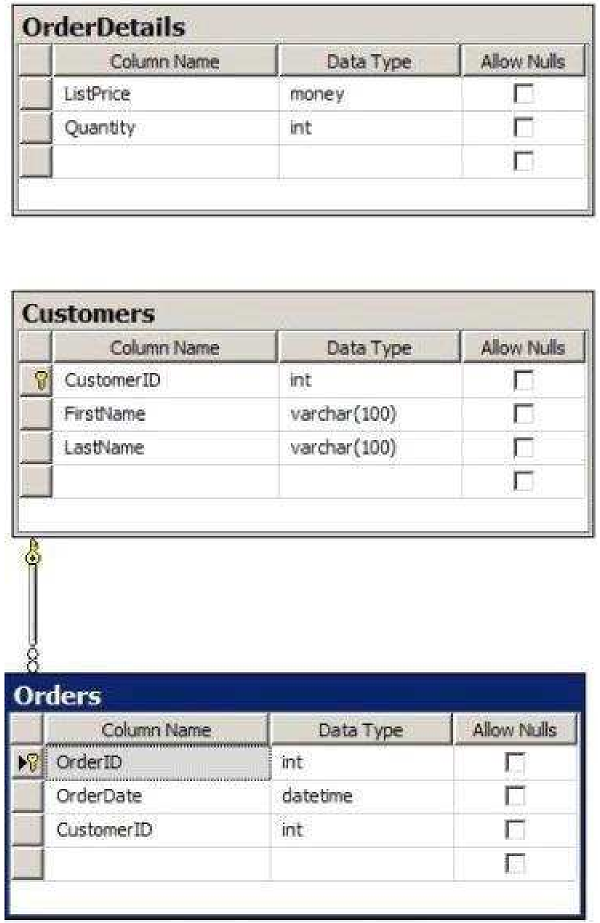
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: SELECT SalesTerritoryID, ProductID, AVG(UnitPrice), MAX(OrderQty), MAX(DiscountAmount) FROM Sales.Details GROUP BY SalesTerritoryID , ProductID ORDER BY SalesTerritoryID DESC, ProductID DESC

QUESTION 46

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



OrderDetails

	Column Name	Data Type	Allow Nulls
	ListPrice	money	□
	Quantity	int	□
			□

Customers

	Column Name	Data Type	Allow Nulls
🔑	CustomerID	int	□
	FirstName	varchar(100)	□
	LastName	varchar(100)	□
			□

Orders

	Column Name	Data Type	Allow Nulls
▶🔑	OrderID	int	□
	OrderDate	datetime	□
	CustomerID	int	□
			□

You need to create a query for a report. The query must meet the following requirements:

- NOT use object delimiters.
- Return the most recent orders first.
- Use the first initial of the table as an alias.
- Return the most recent order date for each customer.
- Retrieve the last name of the person who placed the order.
- Return the order date in a column named MostRecentOrderDate that appears as the last column in the report.

The solution must support the ANSI SQL-99 standard.
Which code segment should you use?

To answer, type the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: SELECT C.LastName, MAX(O.OrderDate) AS MostRecentOrderDate FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID GROUP BY C.LastName ORDER BY MostRecentOrderDate DESC

QUESTION 47

You have an XML schema collection named Sales.InvoiceSchema. You need to declare a variable of the XML type named XML1. The solution must ensure that XML1 is validated by using Sales.InvoiceSchema.

Which code segment should you use? To answer, type the correct code in the answer area.

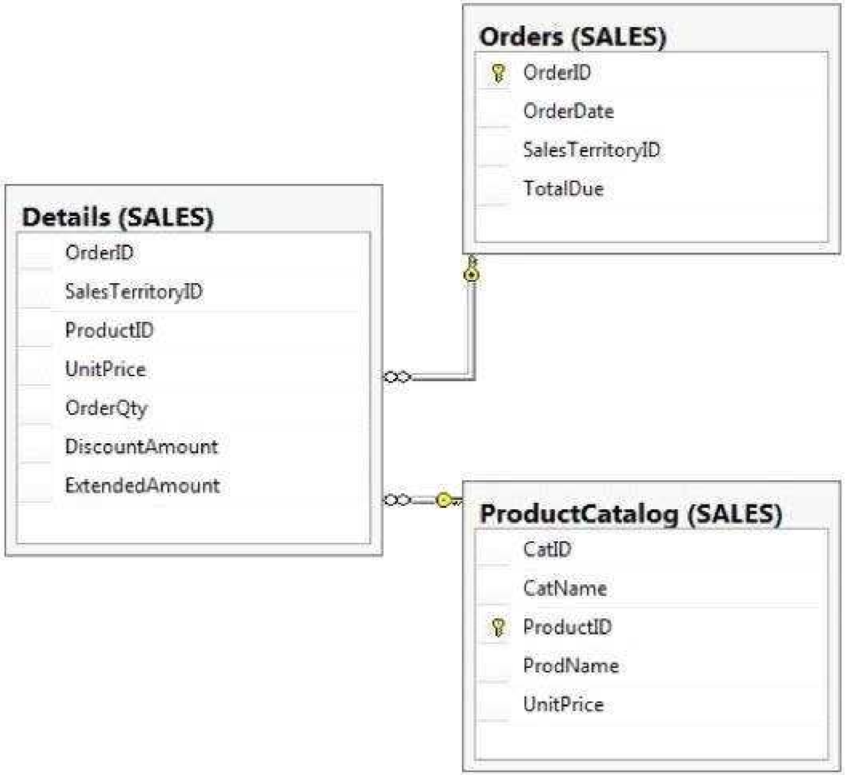
Correct Answer: Answer: DECLARE @XML1 XML(Sales.InvoiceSchema)
Section: (none)
Explanation

Explanation/Reference:

Explanation: Reference: http://msdn.microsoft.com/en-us/library/ms176009.aspx

QUESTION 48

You have a database that contains the tables as shown in the exhibit. (Click the Exhibit button.)



You need to create a query that returns a list of products from Sales.ProductCatalog. The solution must meet the following requirements:

- UnitPrice must be returned in descending order.
- The query must use two-part names to reference the table.
- The query must use the RANK function to calculate the results.
- The query must return the ranking of rows in a column named PriceRank.
- The list must display the columns in the order that they are defined in the table.
- PriceRank must appear last.

Which code segment should you use? To answer, type the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

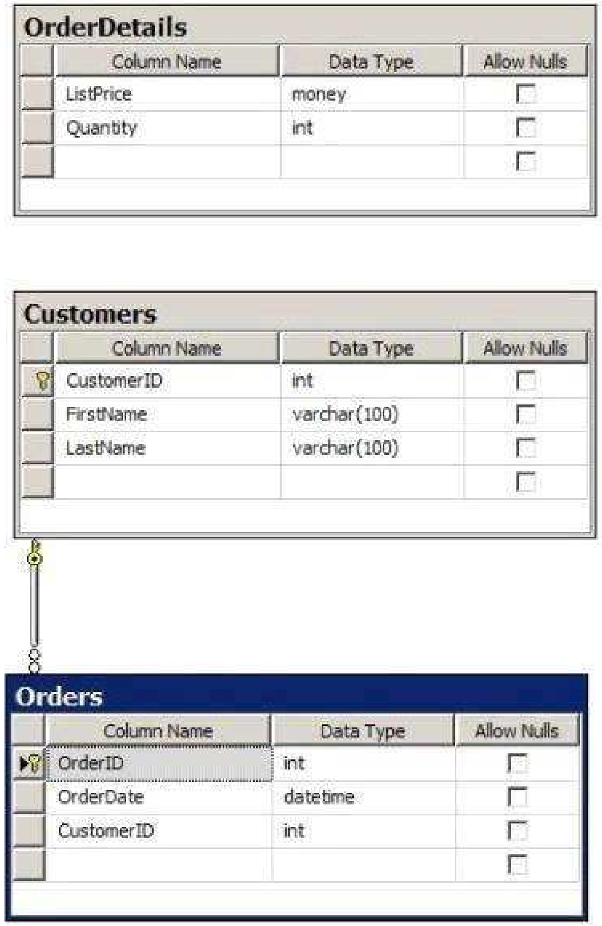
Explanation/Reference:

Explanation:

SELECT ProductCatalog.CatID, ProductCatalog.CatName, ProductCatalog.ProductID, ProductCatalog.ProdName, ProductCatalog.UnitPrice, RANK() OVER (ORDER BY ProductCatalog.UnitPrice DESC) AS PriceRank FROM Sales.ProductCatalog ORDER BY ProductCatalog.UnitPrice DESC

QUESTION 49

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



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	□
Quantity	int	□
		□

Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	□
FirstName	varchar(100)	□
LastName	varchar(100)	□
		□

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	□
OrderDate	datetime	□
CustomerID	int	□
		□

You have an application named Appl. You have a parameter named @Count that uses the int data type. App1 is configured to pass @Count to a stored procedure.

You need to create a stored

procedure named usp_Customers for Appl. Usp_Customers must meet the following requirements:
- NOT use object delimiters.
- Minimize sorting and counting.
- Return only the last name of each customer in alphabetical order.
- Return only the number of rows specified by the @Count parameter.
- The solution must NOT use BEGIN and END statements.

Which code segment should you use? To answer, type the correct code in the answer area.

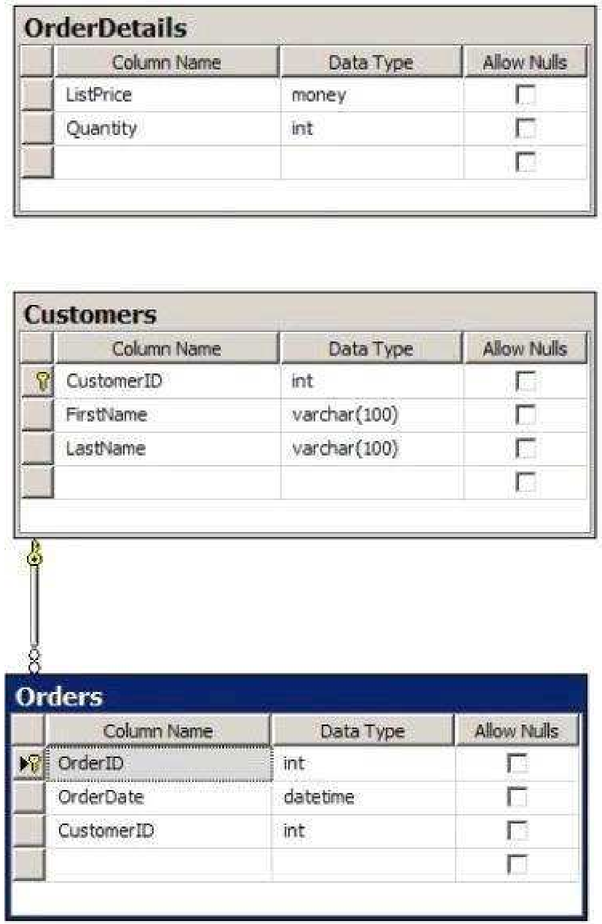
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE PROCEDURE usp_Customers @Count int AS SELECT TOP(@Count) Customers.LastName FROM Customers ORDER BY Customers.LastName

QUESTION 50

You have a database that contains the tables as shown below:



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	□
Quantity	int	□
		□

Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	□
FirstName	varchar(100)	□
LastName	varchar(100)	□
		□

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	□
OrderDate	datetime	□
CustomerID	int	□
		□

You have a stored procedure named Procedure1. Procedure1 retrieves all order ids after a specific date. The rows for Procedure1 are not sorted. Procedure1 has a single parameter named Parameter1. Parameter1 uses the varchar type and is configured to pass the specific date to Procedure1. A database administrator discovers that OrderDate is not being compared correctly to Parameter1 after the data type of the column is changed to datetime. You need to update the SELECT statement to meet the following requirements:

- The code must NOT use aliases.
- The code must NOT use object delimiters.
- The objects called in Procedure1 must be able to be resolved by all users.
- OrderDate must be compared to Parameter1 after the data type of Parameter1 is changed to datetime.

Which SELECT statement should you use? To answer, type the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: SELECT Orders.OrderID FROM Orders WHERE Orders.OrderDate>CONVERT(datetime,@Parameter1)

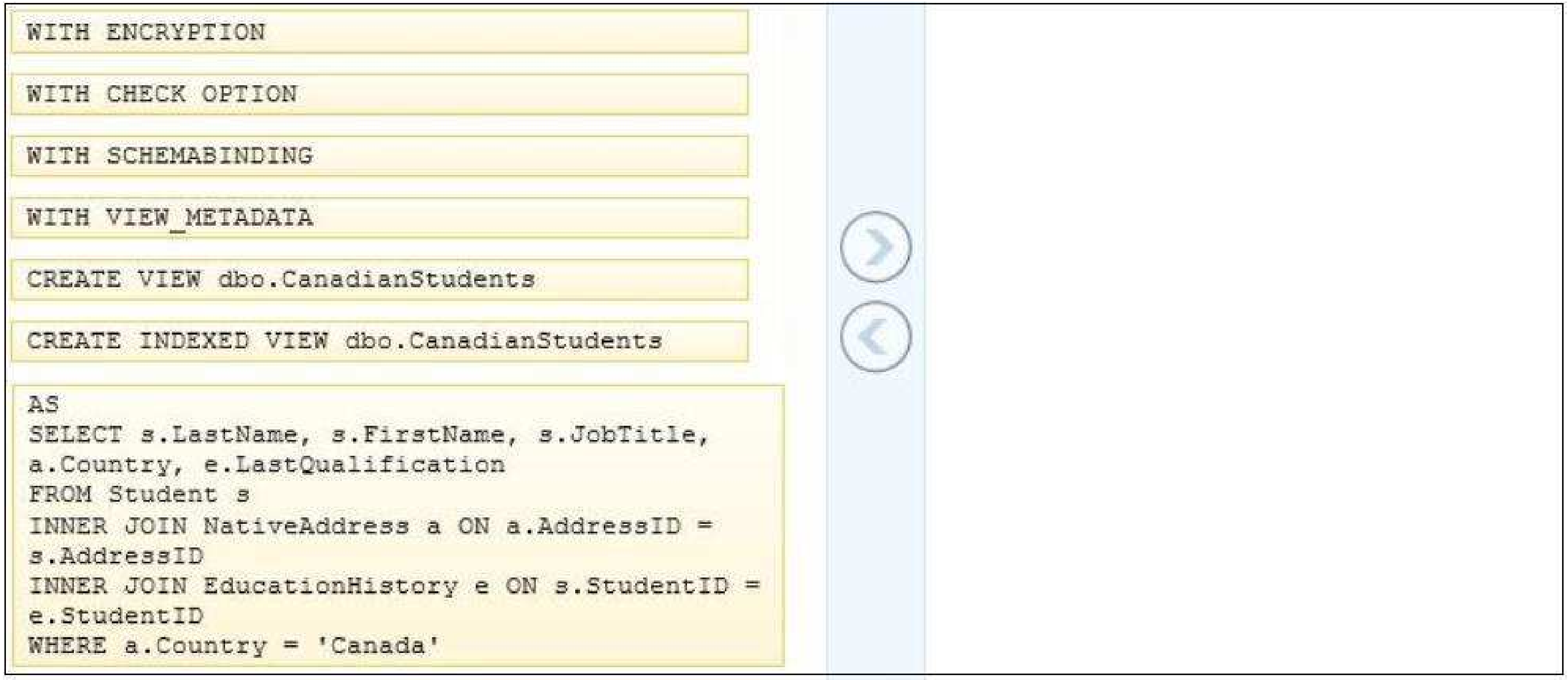
QUESTION 51

You develop a database application for a university. You need to create a view that will be indexed that meets the following requirements:

- Displays the details of only students from Canada.
- Allows insertion of details of only students from Canada.

Which four Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
WITH ENCRYPTION
```

```
WITH CHECK OPTION
```

```
WITH SCHEMABINDING
```

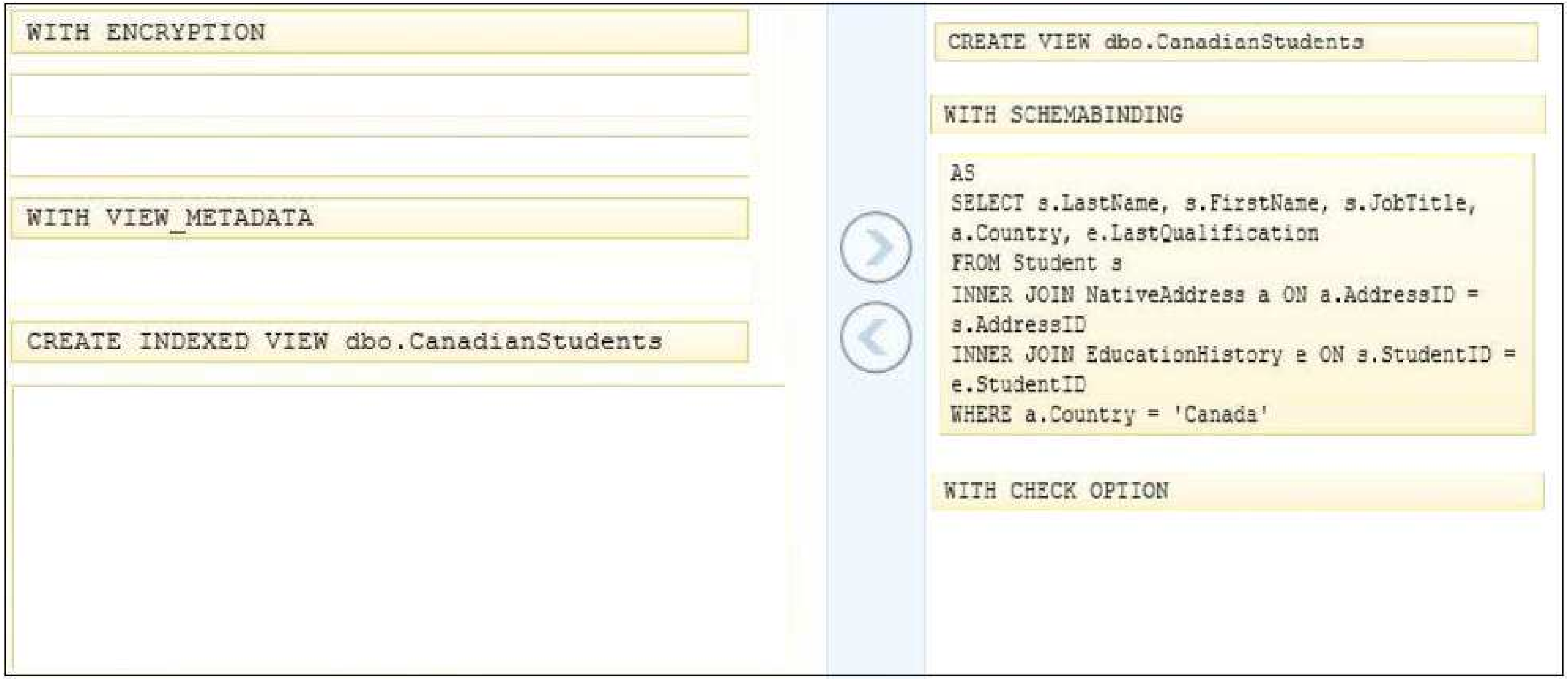
```
WITH VIEW_METADATA
```

```
CREATE VIEW dbo.CanadianStudents
```

```
CREATE INDEXED VIEW dbo.CanadianStudents
```

```
AS
SELECT s.LastName, s.FirstName, s.JobTitle,
a.Country, e.LastQualification
FROM Student s
INNER JOIN NativeAddress a ON a.AddressID =
s.AddressID
INNER JOIN EducationHistory e ON s.StudentID =
e.StudentID
WHERE a.Country = 'Canada'
```

Correct Answer:



WITH ENCRYPTION

WITH VIEW_METADATA

CREATE INDEXED VIEW dbo.CanadianStudents

CREATE VIEW dbo.CanadianStudents

WITH SCHEMABINDING

```
AS
SELECT s.LastName, s.FirstName, s.JobTitle,
a.Country, e.LastQualification
FROM Student s
INNER JOIN NativeAddress a ON a.AddressID =
s.AddressID
INNER JOIN EducationHistory e ON s.StudentID =
e.StudentID
WHERE a.Country = 'Canada'
```

WITH CHECK OPTION

Section: (none)
Explanation

Explanation/Reference:

Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms187956.aspx

QUESTION 52

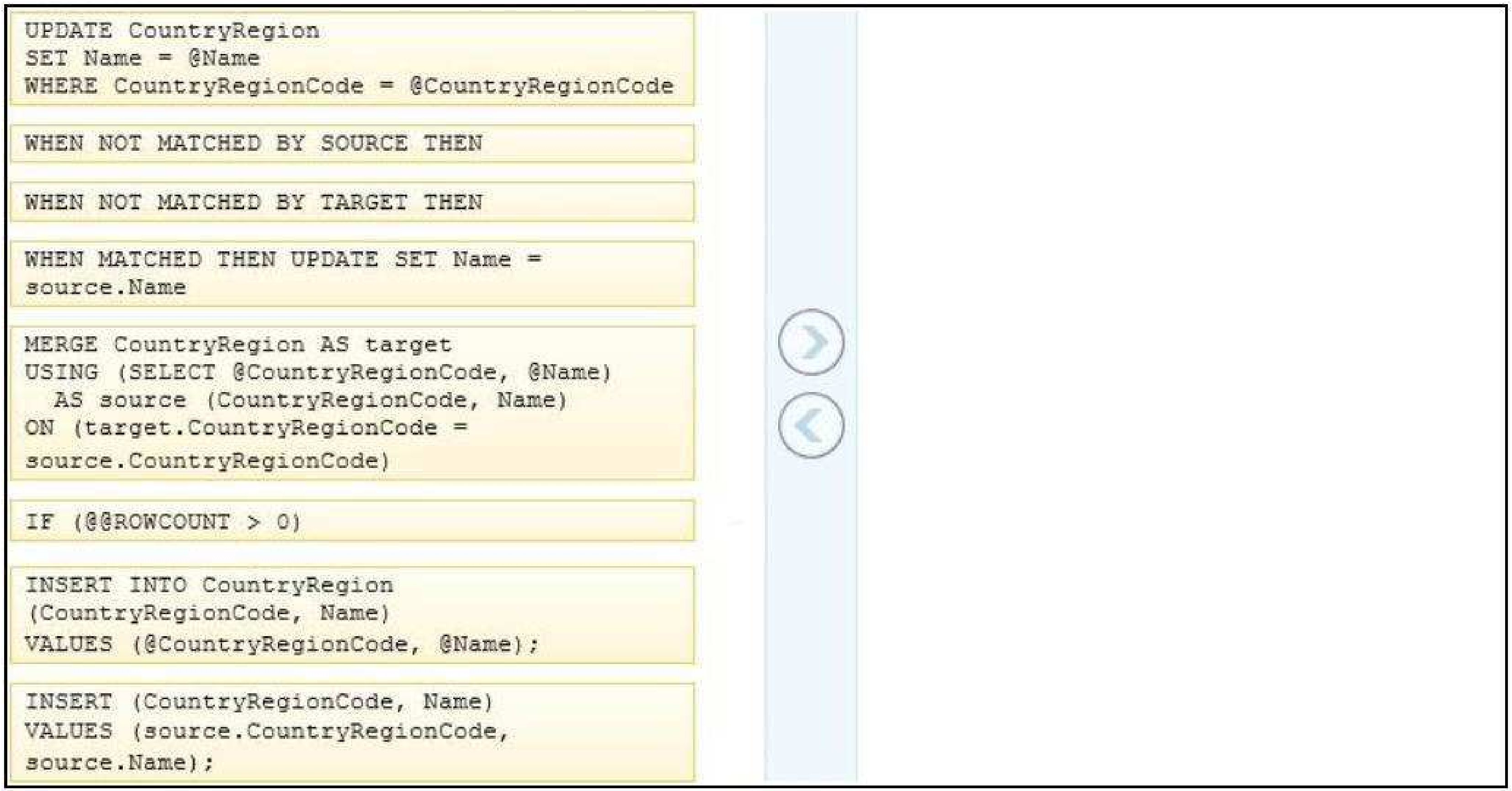
You create the following stored procedure. (Line numbers are included for reference only.) You need to ensure that the stored procedure performs the following tasks:

- If a record exists, update the record.
- If no record exists, insert a new record.

Which four Transact-SQL statements should you insert at line 07? (To answer, move the appropriate statements from the list of statements to the answer area and

arrange them in the correct order.)
Answer:

Select and Place:



```
UPDATE CountryRegion
SET Name = @Name
WHERE CountryRegionCode = @CountryRegionCode
```

```
WHEN NOT MATCHED BY SOURCE THEN
```

```
WHEN NOT MATCHED BY TARGET THEN
```

```
WHEN MATCHED THEN UPDATE SET Name =
source.Name
```

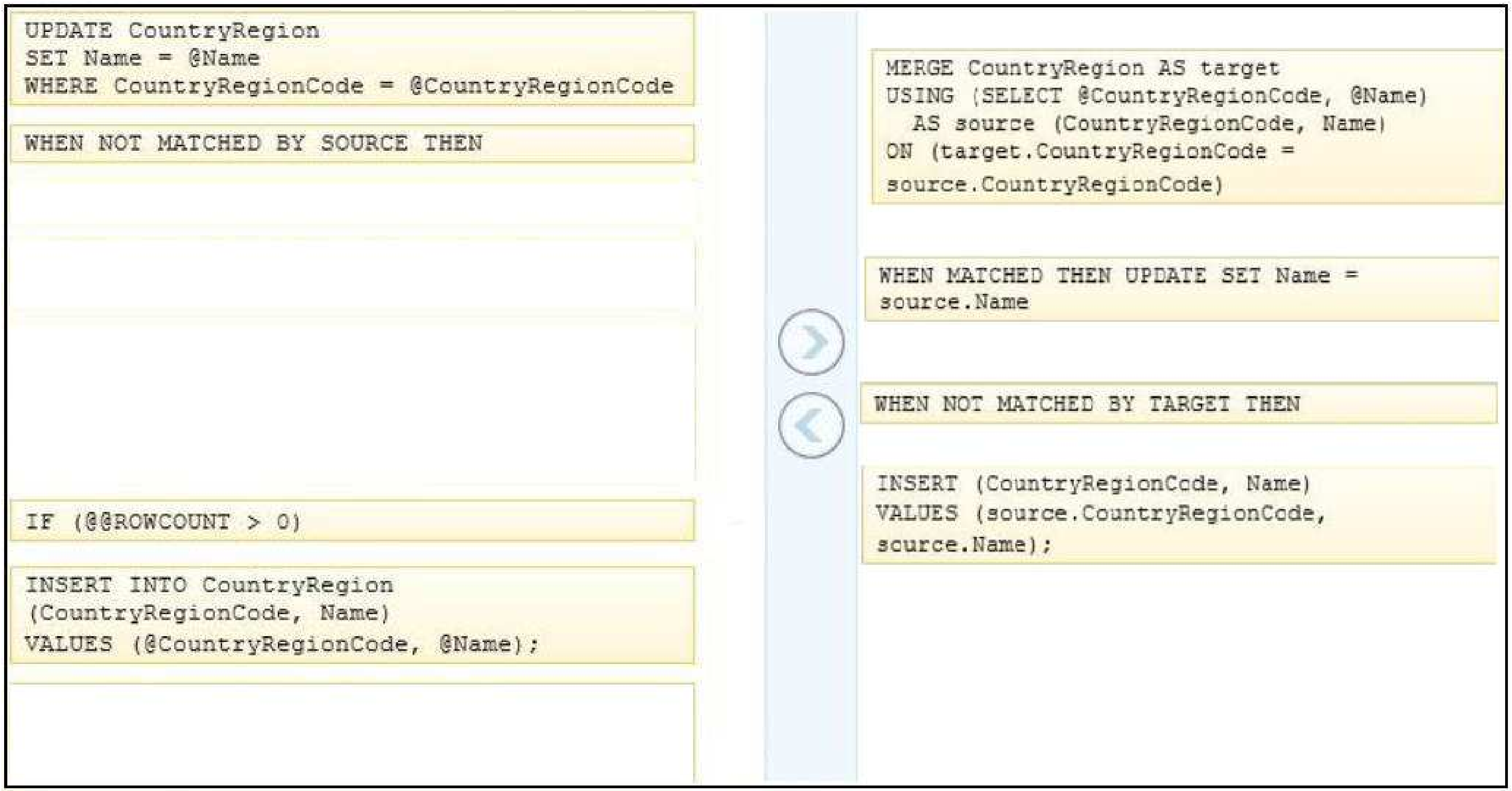
```
MERGE CountryRegion AS target
USING (SELECT @CountryRegionCode, @Name)
  AS source (CountryRegionCode, Name)
ON (target.CountryRegionCode =
source.CountryRegionCode)
```

```
IF (@@ROWCOUNT > 0)
```

```
INSERT INTO CountryRegion
(CountryRegionCode, Name)
VALUES (@CountryRegionCode, @Name);
```

```
INSERT (CountryRegionCode, Name)
VALUES (source.CountryRegionCode,
source.Name);
```

Correct Answer:



```
UPDATE CountryRegion
SET Name = @Name
WHERE CountryRegionCode = @CountryRegionCode
```

```
WHEN NOT MATCHED BY SOURCE THEN
```

```
IF (@@ROWCOUNT > 0)
```

```
INSERT INTO CountryRegion
(CountryRegionCode, Name)
VALUES (@CountryRegionCode, @Name);
```

```
MERGE CountryRegion AS target
USING (SELECT @CountryRegionCode, @Name)
  AS source (CountryRegionCode, Name)
ON (target.CountryRegionCode =
source.CountryRegionCode)
```

```
WHEN MATCHED THEN UPDATE SET Name =
source.Name
```

```
WHEN NOT MATCHED BY TARGET THEN
```

```
INSERT (CountryRegionCode, Name)
VALUES (source.CountryRegionCode,
source.Name);
```

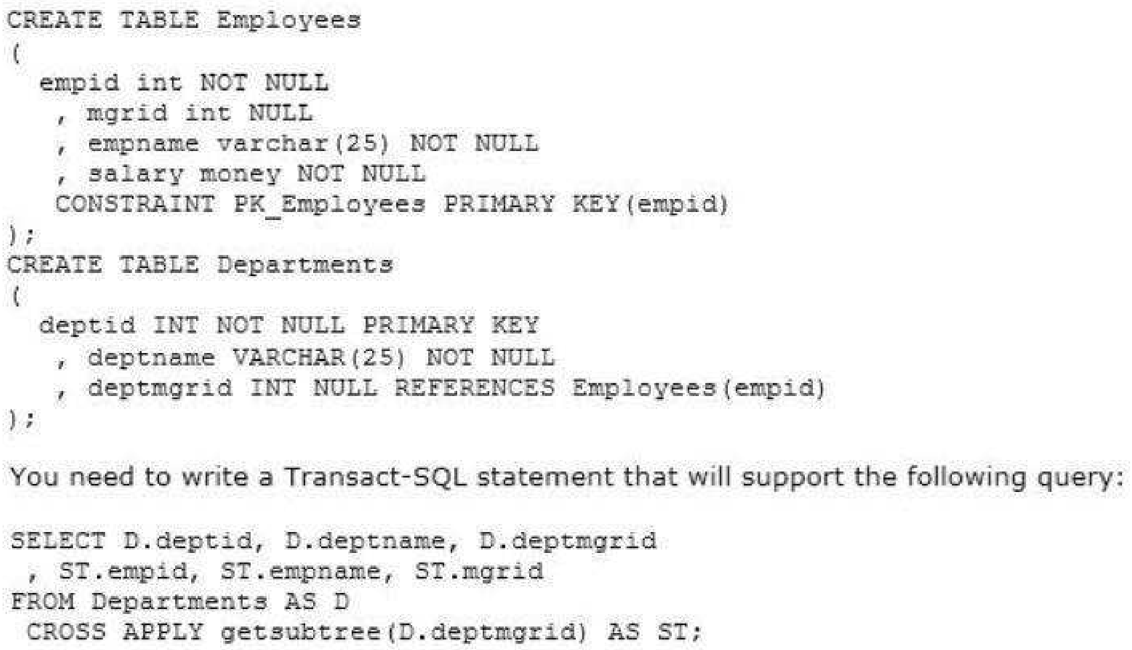
Section: (none)
Explanation

Explanation/Reference:

Explanation:
Reference: http://technet.microsoft.com/en-us/library/bb510625.aspx

QUESTION 53

You use Microsoft SQL Server 2012 to develop a database application. You create two tables by using the following table definitions.



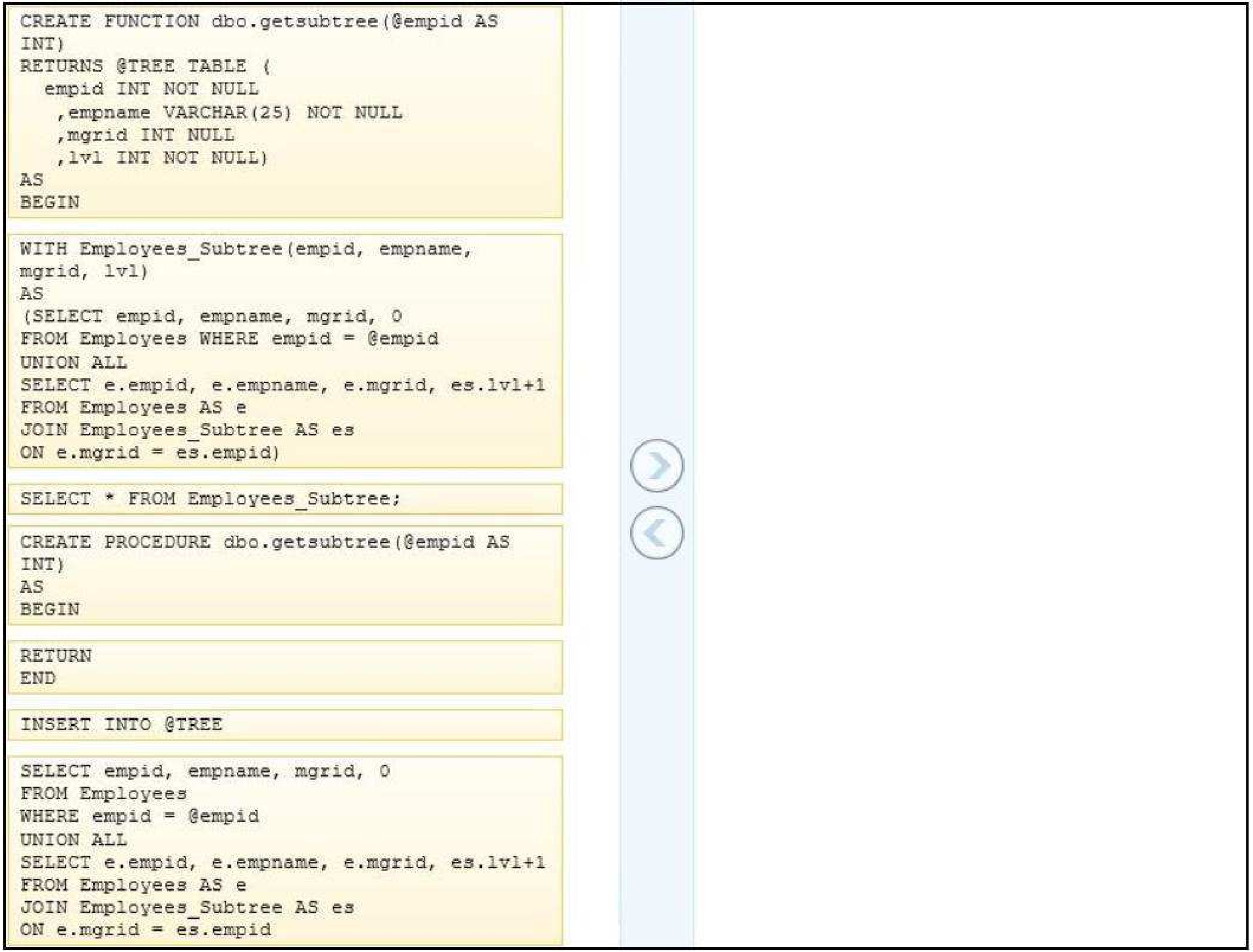
```
CREATE TABLE Employees
(
  empid int NOT NULL
   , mgrid int NULL
   , empname varchar(25) NOT NULL
   , salary money NOT NULL
   CONSTRAINT PK_Employees PRIMARY KEY(empid)
);
CREATE TABLE Departments
(
  deptid INT NOT NULL PRIMARY KEY
   , deptname VARCHAR(25) NOT NULL
   , deptmgrid INT NULL REFERENCES Employees(empid)
);
```

You need to write a Transact-SQL statement that will support the following query:

```
SELECT D.deptid, D.deptname, D.deptmgrid
 , ST.empid, ST.empname, ST.mgrid
FROM Departments AS D
 CROSS APPLY getsubtree(D.deptmgrid) AS ST;
```

Which six Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
CREATE FUNCTION dbo.getsubtree(@empid AS
INT)
RETURNS @TREE TABLE (
  empid INT NOT NULL
   ,empname VARCHAR(25) NOT NULL
   ,mgrid INT NULL
   ,lvl INT NOT NULL)
AS
BEGIN
```

```
WITH Employees_Subtree(empid, empname,
mgrid, lvl)
AS
(SELECT empid, empname, mgrid, 0
FROM Employees WHERE empid = @empid
UNION ALL
SELECT e.empid, e.empname, e.mgrid, es.lvl+1
FROM Employees AS e
JOIN Employees_Subtree AS es
ON e.mgrid = es.empid)
```

```
SELECT * FROM Employees_Subtree;
```

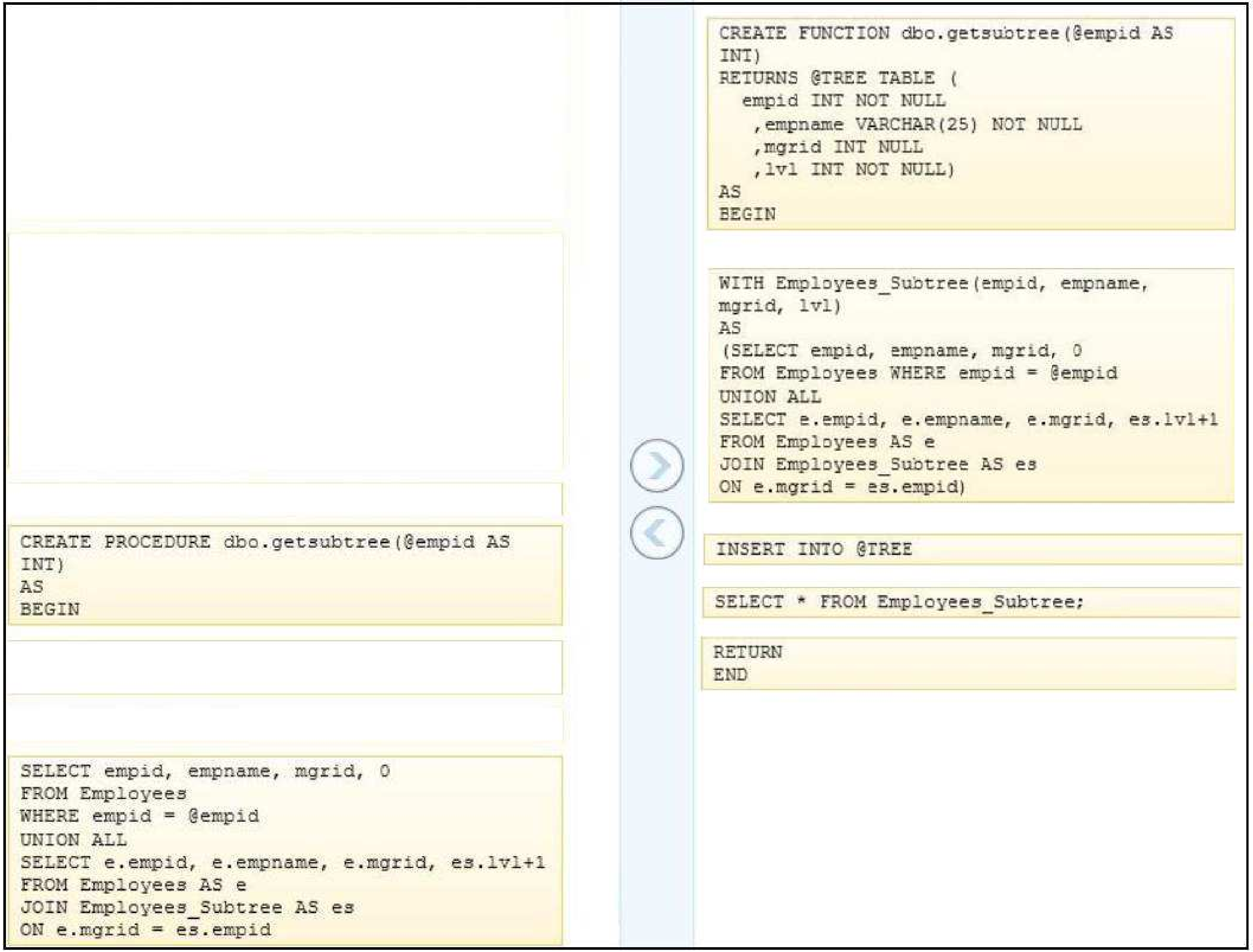
```
CREATE PROCEDURE dbo.getsubtree(@empid AS
INT)
AS
BEGIN
```

```
RETURN
END
```

```
INSERT INTO @TREE
```

```
SELECT empid, empname, mgrid, 0
FROM Employees
WHERE empid = @empid
UNION ALL
SELECT e.empid, e.empname, e.mgrid, es.lvl+1
FROM Employees AS e
JOIN Employees_Subtree AS es
ON e.mgrid = es.empid
```

Correct Answer:



```
CREATE PROCEDURE dbo.getsubtree(@empid AS
INT)
AS
BEGIN
```

```
SELECT empid, empname, mgrid, 0
FROM Employees
WHERE empid = @empid
UNION ALL
SELECT e.empid, e.empname, e.mgrid, es.lvl+1
FROM Employees AS e
JOIN Employees_Subtree AS es
ON e.mgrid = es.empid
```

```
CREATE FUNCTION dbo.getsubtree(@empid AS
INT)
RETURNS @TREE TABLE (
  empid INT NOT NULL
   ,empname VARCHAR(25) NOT NULL
   ,mgrid INT NULL
   ,lvl INT NOT NULL)
AS
BEGIN
```

```
WITH Employees_Subtree(empid, empname,
mgrid, lvl)
AS
(SELECT empid, empname, mgrid, 0
FROM Employees WHERE empid = @empid
UNION ALL
SELECT e.empid, e.empname, e.mgrid, es.lvl+1
FROM Employees AS e
JOIN Employees_Subtree AS es
ON e.mgrid = es.empid)
```

```
INSERT INTO @TREE
```

```
SELECT * FROM Employees_Subtree;
```

```
RETURN
END
```

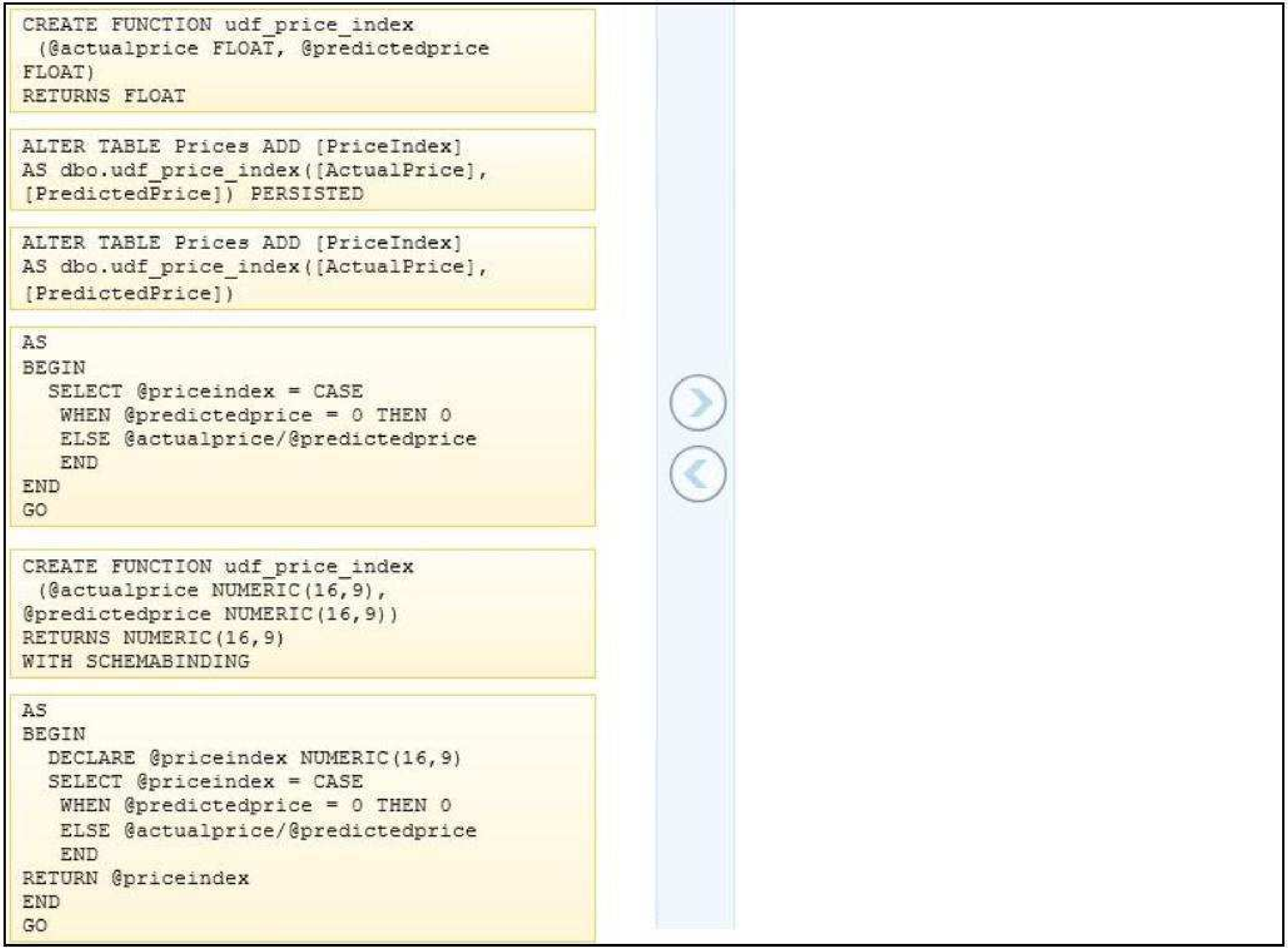
Section: (none)
Explanation

Explanation/Reference:

QUESTION 54

You use Microsoft SQL Server 2012 to develop a database application. You create a table by using the following definition: CREATE TABLE Prices (PriceId int IDENTITY(1,1) PRIMARY KEY, ActualPrice NUMERIC(16,9), PredictedPrice NUMERIC(16,9)) You need to create a computed column based on a user-defined function named udf_price_index. You also need to ensure that the column supports an index. Which three Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
CREATE FUNCTION udf_price_index
 (@actualprice FLOAT, @predictedprice
FLOAT)
RETURNS FLOAT
```

```
ALTER TABLE Prices ADD [PriceIndex]
AS dbo.udf_price_index([ActualPrice],
[PredictedPrice]) PERSISTED
```

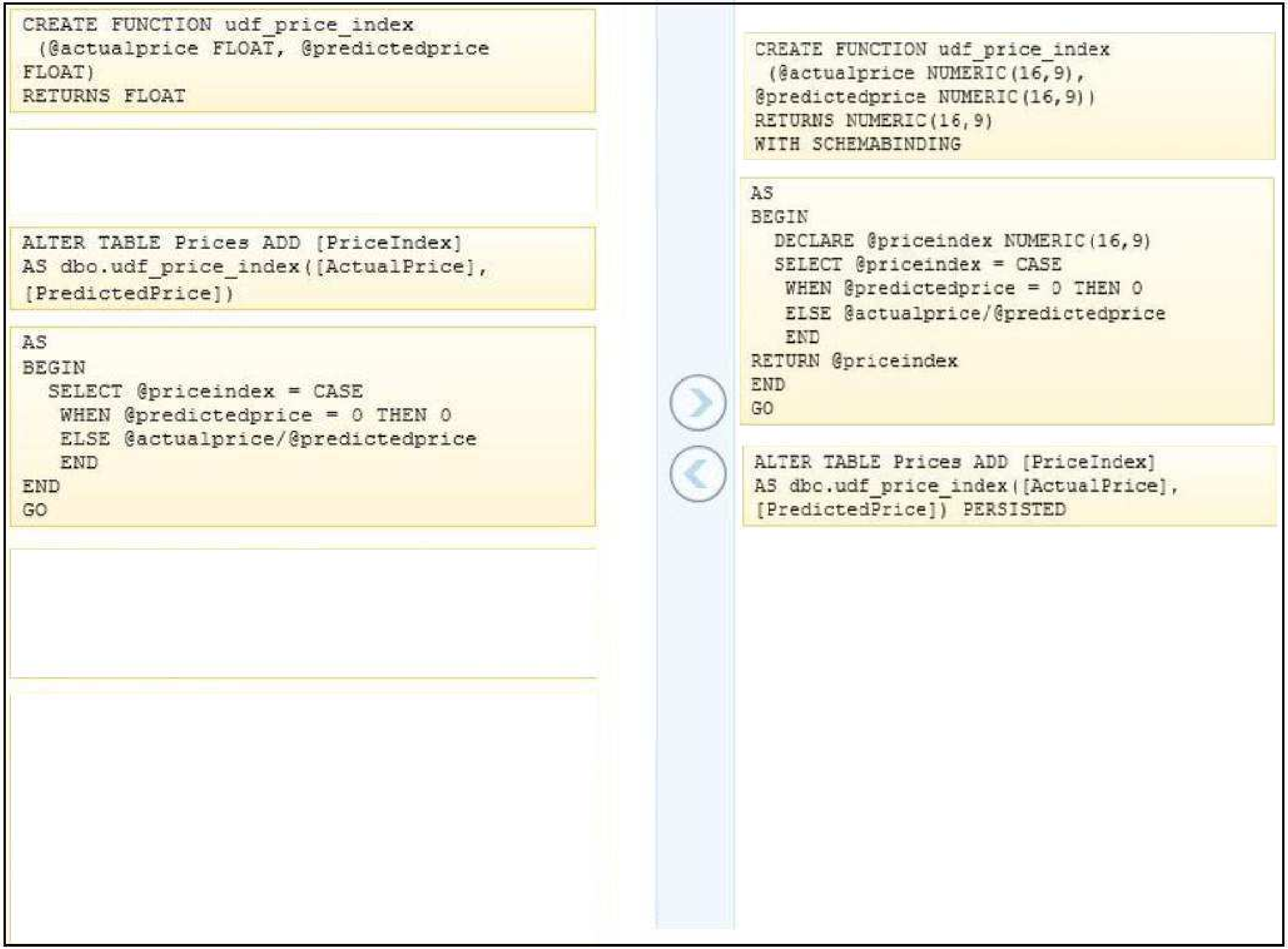
```
ALTER TABLE Prices ADD [PriceIndex]
AS dbo.udf_price_index([ActualPrice],
[PredictedPrice])
```

```
AS
BEGIN
  SELECT @priceindex = CASE
   WHEN @predictedprice = 0 THEN 0
   ELSE @actualprice/@predictedprice
   END
END
GO
```

```
CREATE FUNCTION udf_price_index
 (@actualprice NUMERIC(16,9),
@predictedprice NUMERIC(16,9))
RETURNS NUMERIC(16,9)
WITH SCHEMABINDING
```

```
AS
BEGIN
  DECLARE @priceindex NUMERIC(16,9)
  SELECT @priceindex = CASE
   WHEN @predictedprice = 0 THEN 0
   ELSE @actualprice/@predictedprice
   END
RETURN @priceindex
END
GO
```

Correct Answer:



```
CREATE FUNCTION udf_price_index
 (@actualprice FLOAT, @predictedprice
FLOAT)
RETURNS FLOAT
```

```
ALTER TABLE Prices ADD [PriceIndex]
AS dbo.udf_price_index([ActualPrice],
[PredictedPrice])
```

```
AS
BEGIN
  SELECT @priceindex = CASE
   WHEN @predictedprice = 0 THEN 0
   ELSE @actualprice/@predictedprice
   END
END
GO
```

```
CREATE FUNCTION udf_price_index
 (@actualprice NUMERIC(16,9),
@predictedprice NUMERIC(16,9))
RETURNS NUMERIC(16,9)
WITH SCHEMABINDING
```

```
AS
BEGIN
  DECLARE @priceindex NUMERIC(16,9)
  SELECT @priceindex = CASE
   WHEN @predictedprice = 0 THEN 0
   ELSE @actualprice/@predictedprice
   END
RETURN @priceindex
END
GO
```

```
ALTER TABLE Prices ADD [PriceIndex]
AS dbc.udf_price_index([ActualPrice],
[PredictedPrice]) PERSISTED
```

Section: (none)
Explanation

Explanation/Reference:

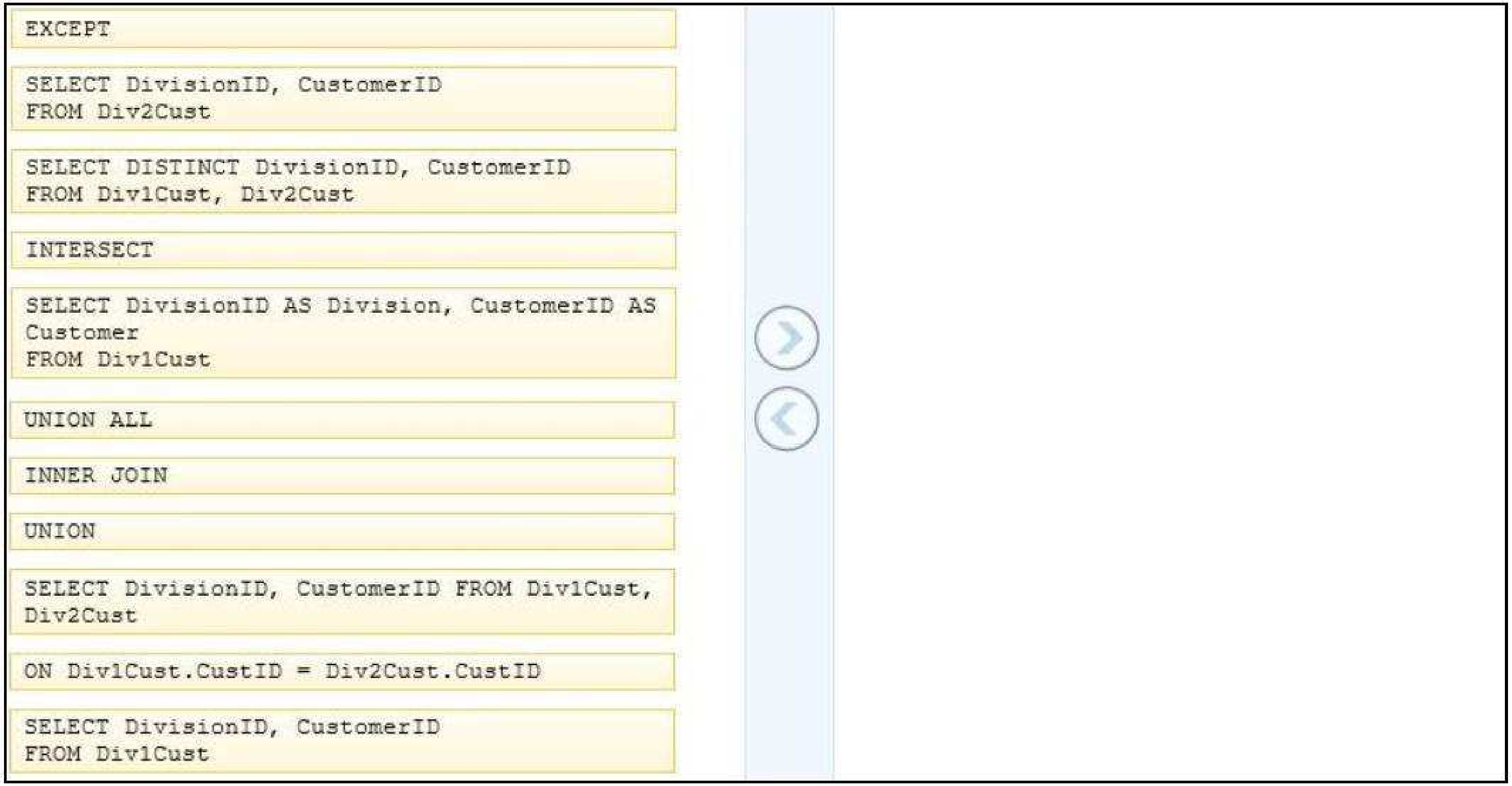
QUESTION 55

You use Microsoft SQL Server 2012 to develop a database that has two tables named Div1Cust and Div2Cust. Each table has columns named DivisionID and CustomerId . None of the rows in Div1Cust exist in Div2Cust. You need to write a query that meets the following requirements:

- The rows in Div1Cust must be combined with the rows in Div2Cust.
- The result set must have columns named Division and Customer.
- Duplicates must be retained.

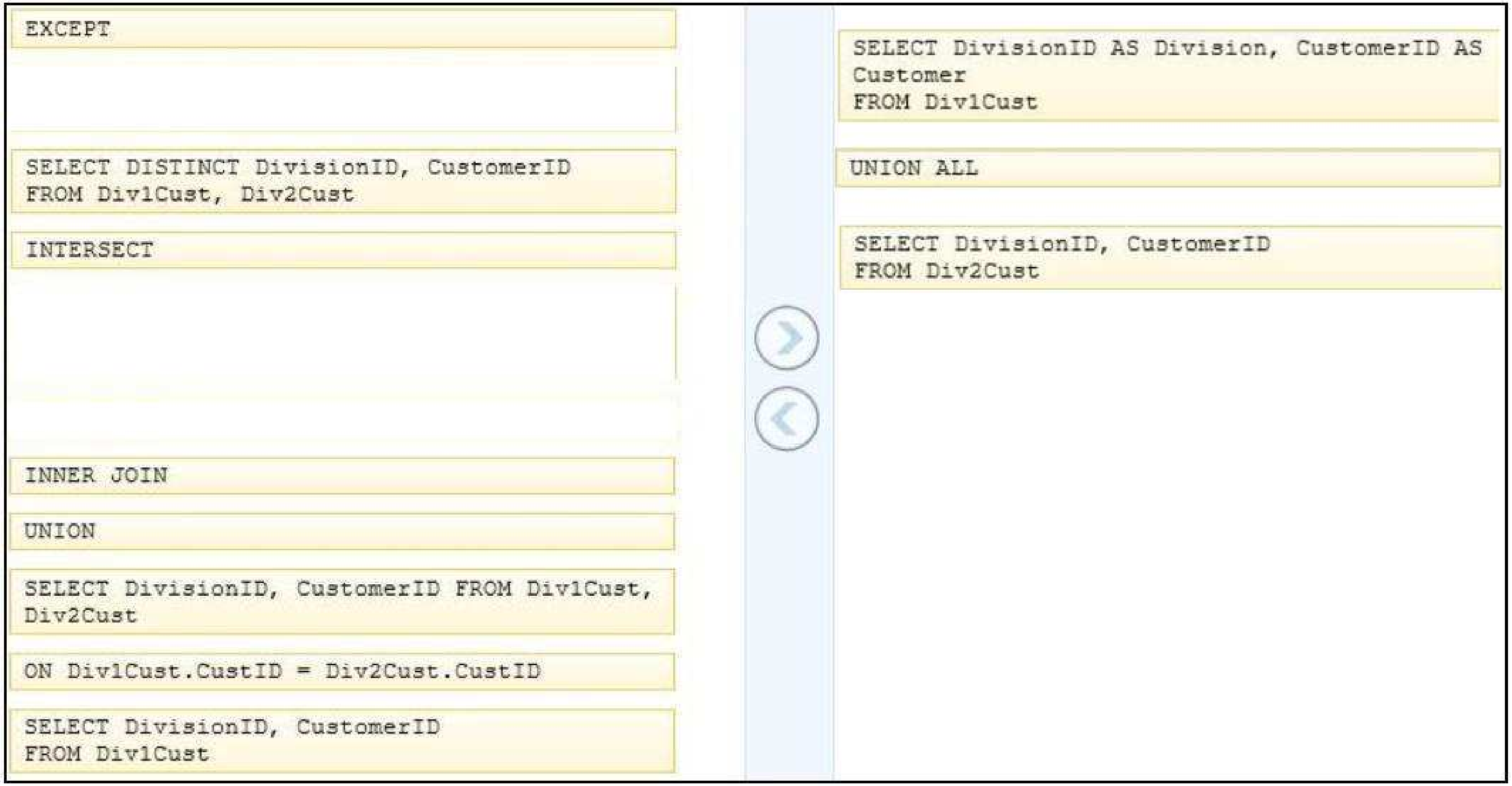
Which three Transact-SQL statements should you use? (To answer, move the appropriate statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
EXCEPT

SELECT DivisionID, CustomerID
FROM Div2Cust

SELECT DISTINCT DivisionID, CustomerID
FROM Div1Cust, Div2Cust

INTERSECT

SELECT DivisionID AS Division, CustomerID AS
Customer
FROM Div1Cust

UNION ALL

INNER JOIN

UNION

SELECT DivisionID, CustomerID FROM Div1Cust,
Div2Cust

ON Div1Cust.CustID = Div2Cust.CustID

SELECT DivisionID, CustomerID
FROM Div1Cust
```

Correct Answer:



```
EXCEPT
```

```
SELECT DISTINCT DivisionID, CustomerID
FROM Div1Cust, Div2Cust
```

```
INTERSECT
```

```
INNER JOIN
```

```
UNION
```

```
SELECT DivisionID, CustomerID FROM Div1Cust,
Div2Cust
```

```
ON Div1Cust.CustID = Div2Cust.CustID
```

```
SELECT DivisionID, CustomerID
FROM Div1Cust
```

```
SELECT DivisionID AS Division, CustomerID AS
Customer
FROM Div1Cust
```

```
UNION ALL
```

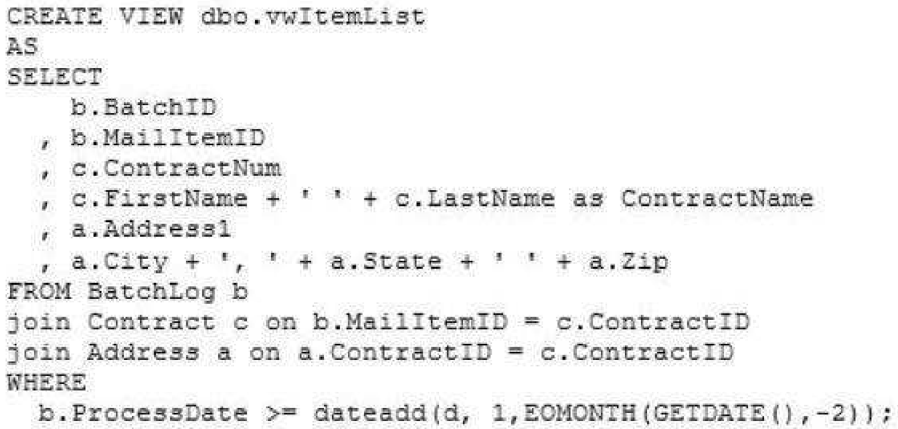
```
SELECT DivisionID, CustomerID
FROM Div2Cust
```

Section: (none)
Explanation

Explanation/Reference:

QUESTION 56

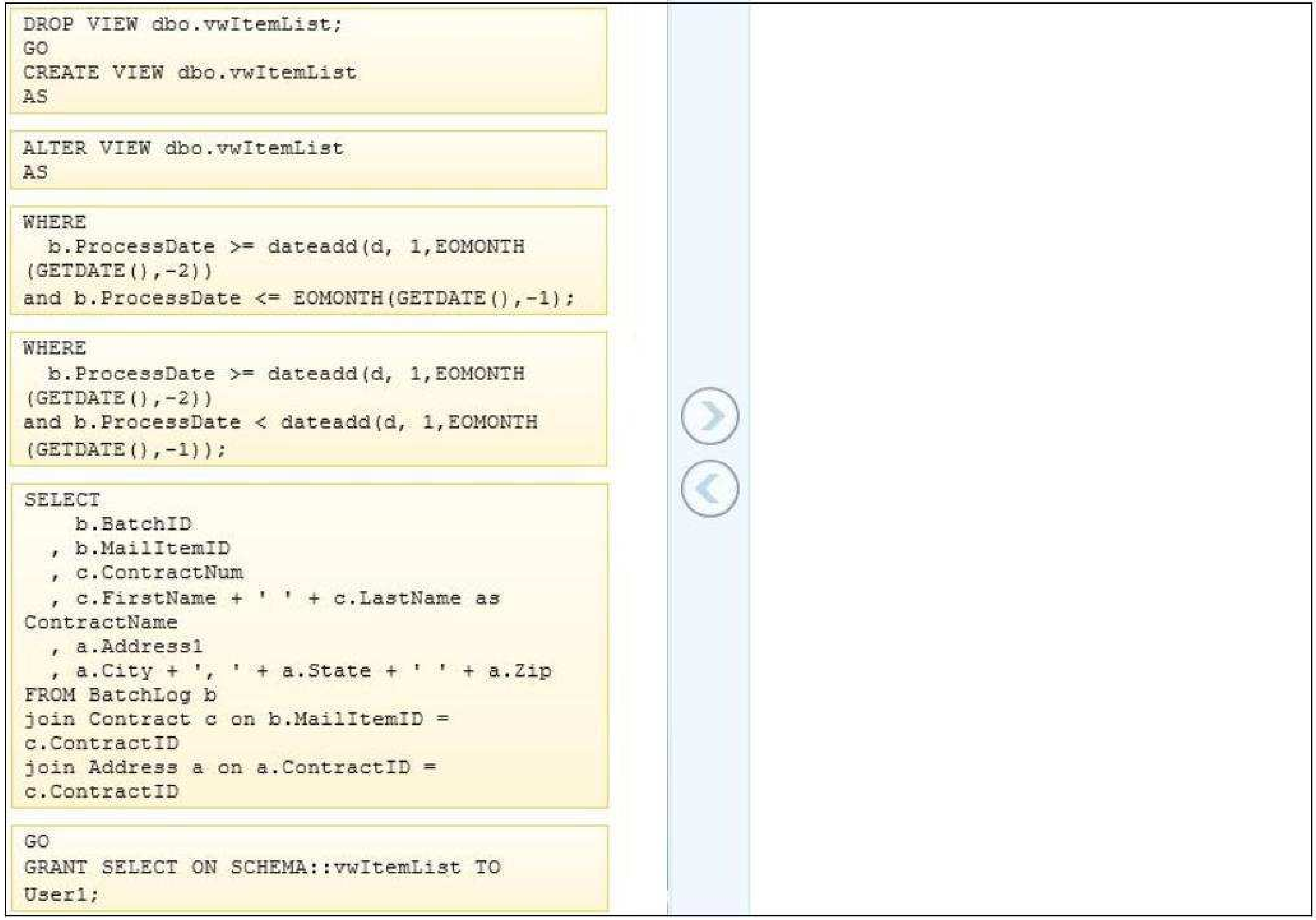
You create a view based on the following statement:



```
CREATE VIEW dbo.vwItemList
AS
SELECT
    b.BatchID
  , b.MailItemID
  , c.ContractNum
  , c.FirstName + ' ' + c.LastName as ContractName
  , a.Address1
  , a.City + ', ' + a.State + ' ' + a.Zip
FROM BatchLog b
join Contract c on b.MailItemID = c.ContractID
join Address a on a.ContractID = c.ContractID
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH(GETDATE(),-2));
```

You grant the Select permission to User1 for this view. You need to change the view so that it displays only the records that were processed in the month prior to the current month. You need to ensure that after the changes, the view functions correctly for User1. Which four Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
DROP VIEW dbo.vwItemList;
GO
CREATE VIEW dbo.vwItemList
AS
```

```
ALTER VIEW dbo.vwItemList
AS
```

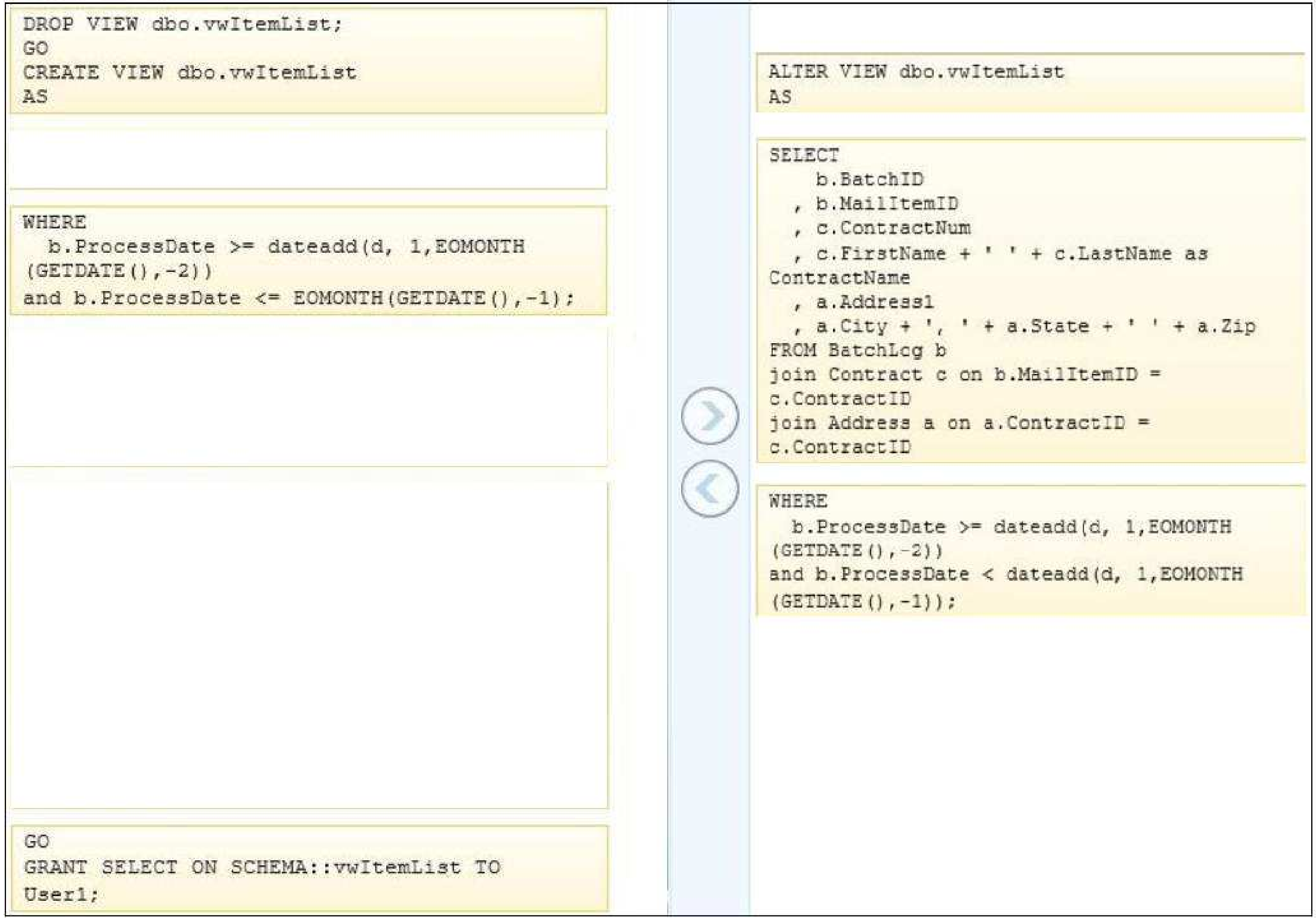
```
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH
(GETDATE(),-2))
and b.ProcessDate <= EOMONTH(GETDATE(),-1);
```

```
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH
(GETDATE(),-2))
and b.ProcessDate < dateadd(d, 1,EOMONTH
(GETDATE(),-1));
```

```
SELECT
    b.BatchID
  , b.MailItemID
  , c.ContractNum
  , c.FirstName + ' ' + c.LastName as
ContractName
  , a.Address1
  , a.City + ', ' + a.State + ' ' + a.Zip
FROM BatchLog b
join Contract c on b.MailItemID =
c.ContractID
join Address a on a.ContractID =
c.ContractID
```

```
GO
GRANT SELECT ON SCHEMA::vwItemList TO
User1;
```

Correct Answer:



```
DROP VIEW dbo.vwItemList;
GO
CREATE VIEW dbo.vwItemList
AS
```

```
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH
(GETDATE(),-2))
and b.ProcessDate <= EOMONTH(GETDATE(),-1);
```

```
GO
GRANT SELECT ON SCHEMA::vwItemList TO
User1;
```

```
ALTER VIEW dbo.vwItemList
AS
```

```
SELECT
    b.BatchID
  , b.MailItemID
  , c.ContractNum
  , c.FirstName + ' ' + c.LastName as
ContractName
  , a.Address1
  , a.City + ', ' + a.State + ' ' + a.Zip
FROM BatchLog b
join Contract c on b.MailItemID =
c.ContractID
join Address a on a.ContractID =
c.ContractID
```

```
WHERE
  b.ProcessDate >= dateadd(d, 1,EOMONTH
(GETDATE(),-2))
and b.ProcessDate < dateadd(d, 1,EOMONTH
(GETDATE(),-1));
```

Section: (none)
Explanation

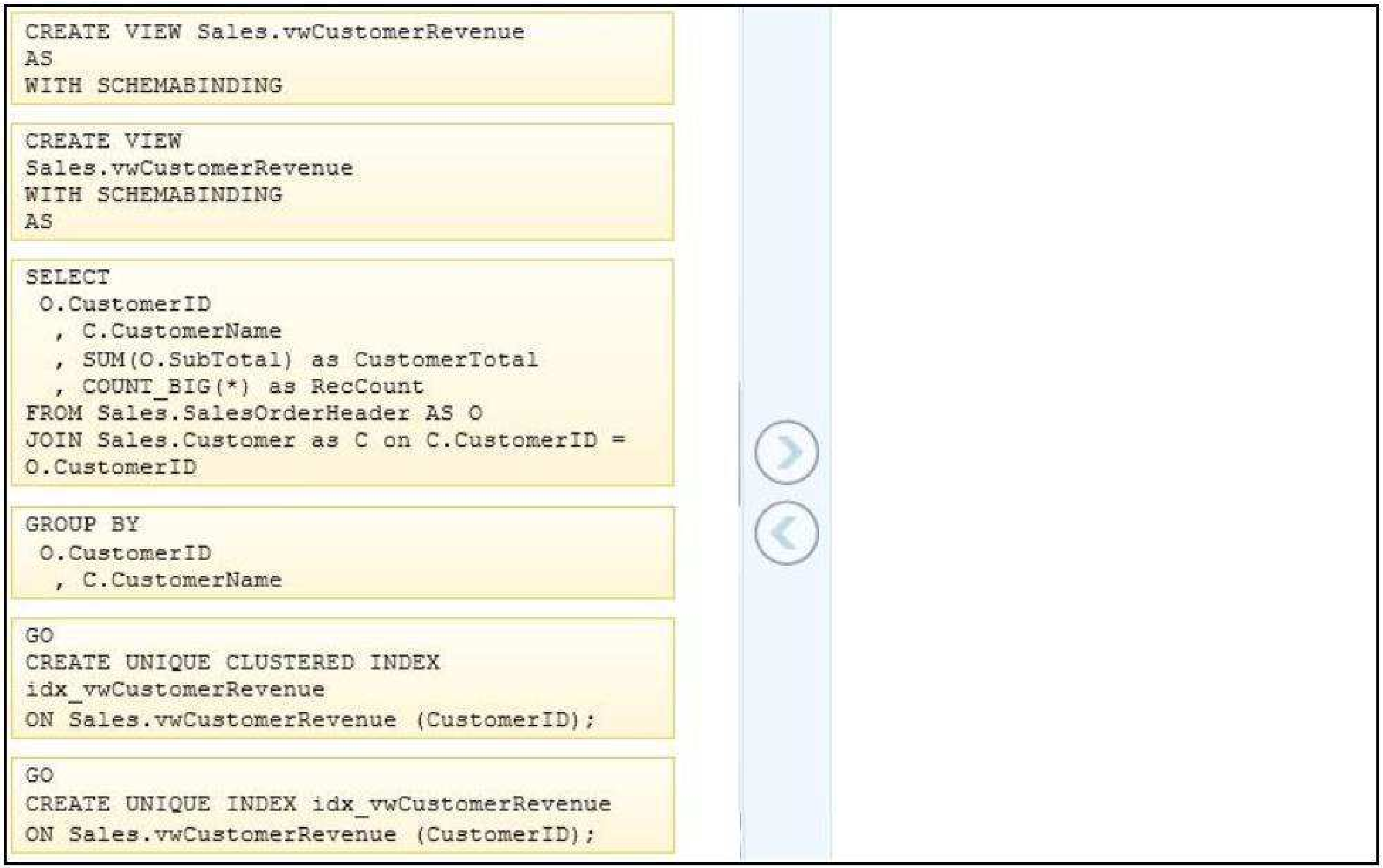
Explanation/Reference:

Explanation:
Reference: http://msdn.microsoft.com/en-us/library/hh213020.aspx
Reference: http://msdn.microsoft.com/en-us/library/ms186819.aspx
Reference: http://msdn.microsoft.com/en-us/library/ms173846.aspx

QUESTION 57

You use a Microsoft SQL Server 2012 database. You need to create an indexed view within the database for a report that displays Customer Name and the total revenue for that customer. Which four T-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
CREATE VIEW Sales.vwCustomerRevenue
AS
WITH SCHEMABINDING
```

```
CREATE VIEW
Sales.vwCustomerRevenue
WITH SCHEMABINDING
AS
```

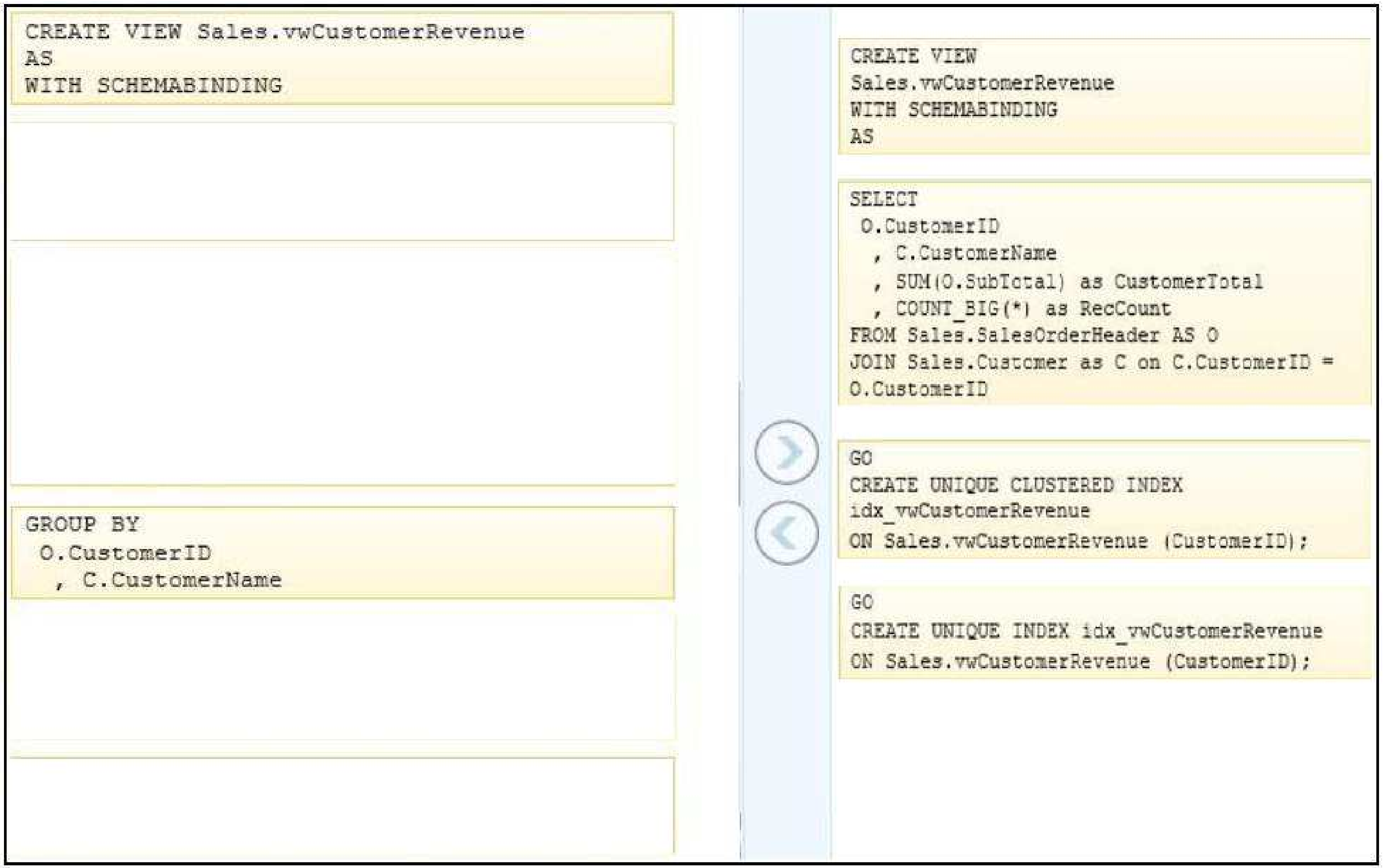
```
SELECT
 O.CustomerID
  , C.CustomerName
  , SUM(O.SubTotal) as CustomerTotal
  , COUNT_BIG(*) as RecCount
FROM Sales.SalesOrderHeader AS O
JOIN Sales.Customer as C on C.CustomerID =
O.CustomerID
```

```
GROUP BY
 O.CustomerID
  , C.CustomerName
```

```
GO
CREATE UNIQUE CLUSTERED INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

```
GO
CREATE UNIQUE INDEX idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

Correct Answer:



```
CREATE VIEW Sales.vwCustomerRevenue
AS
WITH SCHEMABINDING
```

```
GROUP BY
 O.CustomerID
  , C.CustomerName
```

```
CREATE VIEW
Sales.vwCustomerRevenue
WITH SCHEMABINDING
AS
```

```
SELECT
 O.CustomerID
  , C.CustomerName
  , SUM(O.SubTotal) as CustomerTotal
  , COUNT_BIG(*) as RecCount
FROM Sales.SalesOrderHeader AS O
JOIN Sales.Customer as C on C.CustomerID =
O.CustomerID
```

```
GO
CREATE UNIQUE CLUSTERED INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

```
GO
CREATE UNIQUE INDEX idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

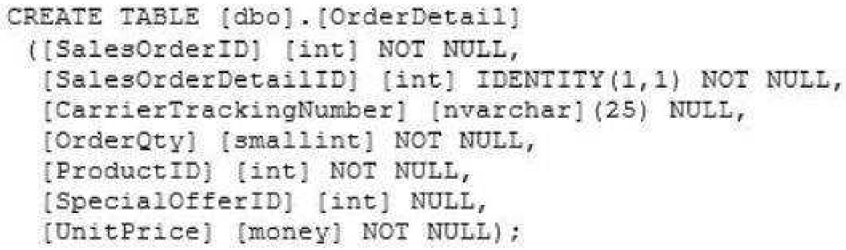
Section: (none)
Explanation

Explanation/Reference:

Explanation:
Reference: http://msdn.microsoft.com/en-us/library/ms191432.aspx Read all restrictions for indexed views. Also read this useful question: http://stackoverflow.com/questions/12419330/how-to-create-indexed-view-with-select-distinct-statement-insql-2005

QUESTION 58

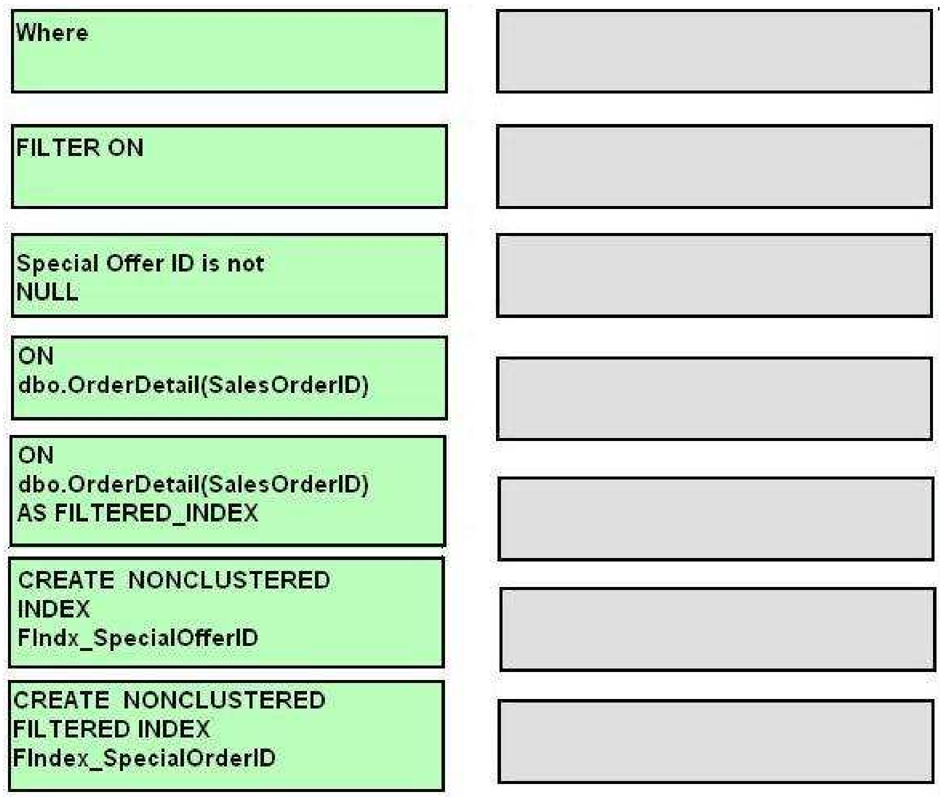
You administer a Microsoft SQL Server 2012 database. You use an OrderDetail table that has the following definition:



```
CREATE TABLE [dbo].[OrderDetail]
 ([SalesOrderID] [int] NOT NULL,
  [SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
  [CarrierTrackingNumber] [nvarchar](25) NULL,
  [OrderQty] [smallint] NOT NULL,
  [ProductID] [int] NOT NULL,
  [SpecialOfferID] [int] NULL,
  [UnitPrice] [money] NOT NULL);
```

You need to create a non-clustered index on the SalesOrderID column in the OrderDetail table to include only rows that contain a value in the SpecialOfferID column. Which four Transact-SQL statements should you use? (To answer, move the appropriate statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



Where

FILTER ON

Special Offer ID is not NULL

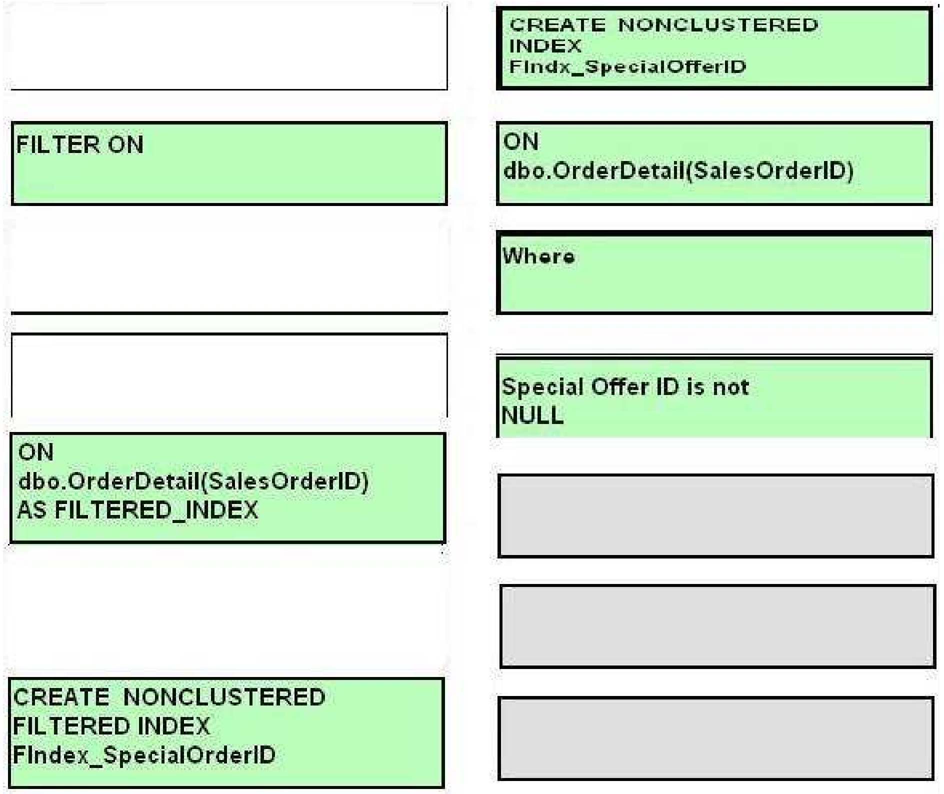
ON dbo.OrderDetail(SalesOrderID)

ON dbo.OrderDetail(SalesOrderID) AS FILTERED_INDEX

CREATE NONCLUSTERED INDEX FIndx_SpecialOfferID

CREATE NONCLUSTERED FILTERED INDEX FIndex_SpecialOrderID

Correct Answer:



FILTER ON

ON
dbo.OrderDetail(SalesOrderID)
AS FILTERED_INDEX

CREATE NONCLUSTERED
FILTERED INDEX
FIndex_SpecialOrderID

CREATE NONCLUSTERED
INDEX
FIndx_SpecialOfferID

ON
dbo.OrderDetail(SalesOrderID)

Where

Special Offer ID is not
NULL

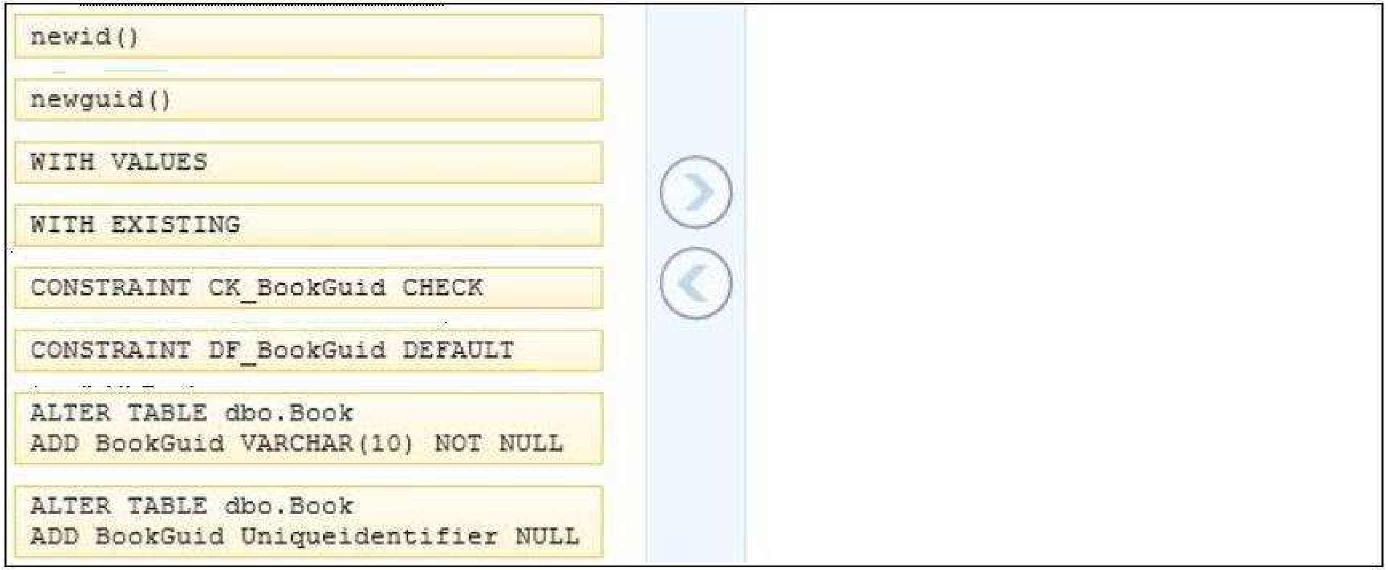
Section: (none)
Explanation

Explanation/Reference:

QUESTION 59

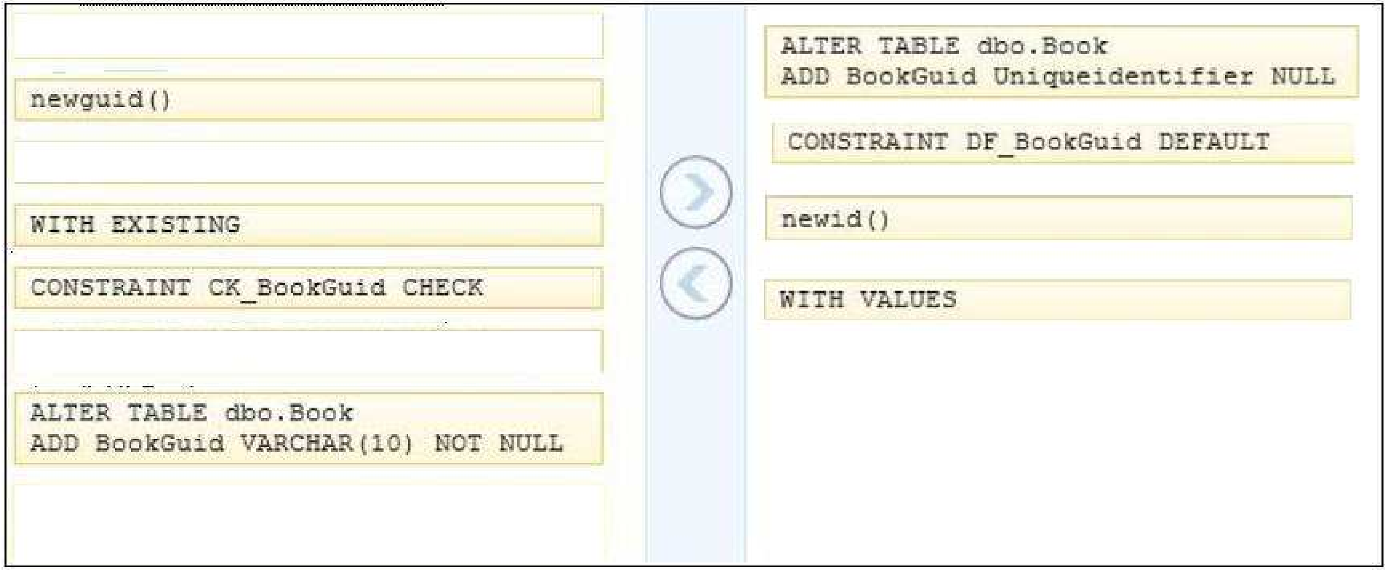
You want to add a new GUID column named BookGUID to a table named dbo.Book that already contains data. BookGUID will have a constraint to ensure that it always has a value when new rows are inserted into dbo.Book. You need to ensure that the new column is assigned a GUID for existing rows. Which four Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



```
newid()

newguid()

WITH VALUES

WITH EXISTING

CONSTRAINT CK_BookGuid CHECK

CONSTRAINT DF_BookGuid DEFAULT

ALTER TABLE dbo.Book
ADD BookGuid VARCHAR(10) NOT NULL

ALTER TABLE dbo.Book
ADD BookGuid Uniqueidentifier NULL
```

Correct Answer:



```
newguid()

WITH EXISTING

CONSTRAINT CK_BookGuid CHECK

ALTER TABLE dbo.Book
ADD BookGuid VARCHAR(10) NOT NULL
```

```
ALTER TABLE dbo.Book
ADD BookGuid Uniqueidentifier NULL

CONSTRAINT DF_BookGuid DEFAULT

newid()

WITH VALUES
```

Section: (none)
Explanation

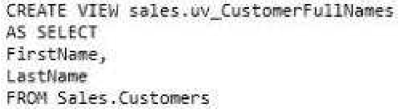
Explanation/Reference:

Explanation:
Actually, in the real world, you don't have to use WITH VALUES at the end of the statement and it works just as well. But because the question specifically states which FOUR TSQL statements to use, we have to include it.

Reference: http://msdn.microsoft.com/en-us/library/ms190273.aspx

QUESTION 60

You need to create a view named uv_CustomerFullNames. The view must prevent the underlying structure of the customer table from being changed. Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
CREATE VIEW sales.uv_CustomerFullNames
AS SELECT
FirstName,
LastName
FROM Sales.Customers
```

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE VIEW sales.uv_CustomerFullNames WITH SCHEMABINDING AS SELECT FirstName, LastName FROM Sales.Customers

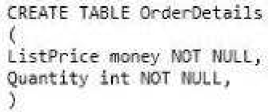
QUESTION 61

You need to create a table named OrderDetails on a new server. OrderDetails must meet the following requirements:

- Contain a new column named LineItemTotal that stores the product of ListPrice and Quantity for each row.
- The calculation for a line item total must not be run every time the table is queried.
- The code must NOT use any object delimiters.

The solution must ensure that LineItemTotal is stored as the last column in the table.

Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
CREATE TABLE OrderDetails
(
ListPrice money NOT NULL,
Quantity int NOT NULL,
)
```

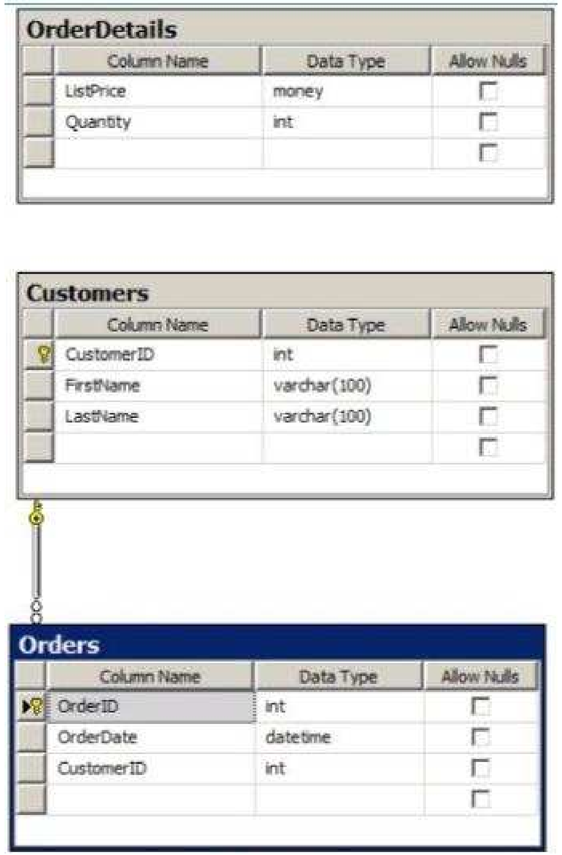
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE TABLE OrderDetails (ListPrice money NOT NULL, Quantity int NOT NULL, LineItemTotal AS (ListPrice * Quantity) PERSISTED)

QUESTION 62

You have a database named Sales that contains the tables shown in the exhibit. (Click the Exhibit button.)



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	☐
Quantity	int	☐
		☐

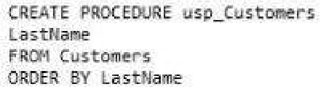
Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	☐
FirstName	varchar(100)	☐
LastName	varchar(100)	☐
		☐

Orders

Column Name	Data Type	Allow Nulls
OrderID	int	☐
OrderDate	datetime	☐
CustomerID	int	☐
		☐

You have an application named Appl. You have a parameter named @Count that uses the int data type. App1 is configured to pass @Count to a stored procedure. You need to create a stored procedure named usp_Customers for App1 that returns only the number of rows specified by the @Count parameter. The solution must NOT use BEGIN and END statements. Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
CREATE PROCEDURE usp_Customers
LastName
FROM Customers
ORDER BY LastName
```

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: CREATE PROCEDURE usp_Customers @Count int SELECT TOP(@Count) Customers.LastName FROM Customers ORDER BY Customers.LastName

QUESTION 63

You need to create a query that calculates the total sales of each OrderID from a table named Sales.Details. The table contains two columns named OrderID and ExtendedAmount. The solution must meet the following requirements:

- Use one-part names to reference columns.
- Start the order of the results from OrderID.
- NOT depend on the default schema of a user.
- Use an alias of TotalSales for the calculated ExtendedAmount.
- Display only the OrderID column and the calculated TotalSales column.

Provide the correct code in the answer area.

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

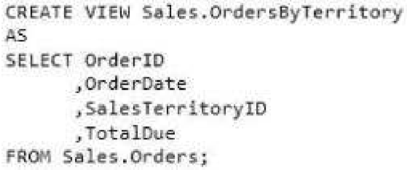
Explanation: SELECT OrderID, SUM(ExtendedAmount) AS TotalSales FROM Sales.Details GROUP BY OrderID ORDER BY OrderID

QUESTION 64
You have an XML schema collection named Sales.InvoiceSchema. You need to declare a variable of the XML type named invoice. The solution must ensure that the invoice is validated by using Sales.InvoiceSchema. Provide the correct code in the answer area.

Correct Answer: Answer: DECLARE @invoice XML(Sales.InvoiceSchema)
Section: (none)
Explanation

Explanation/Reference:

QUESTION 65
You have a view that was created by using the following code:

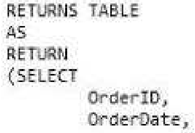


```
CREATE VIEW Sales.OrdersByTerritory
AS
SELECT OrderID
       ,OrderDate
       ,SalesTerritoryID
       ,TotalDue
FROM Sales.Orders;
```

You need to create an inline table-valued function named Sales.fn_OrdersByTerritory. Sales.fn_OrdersByTerritory must meet the following requirements:

- Use one-part names to reference columns.
- Return the columns in the same order as the order used in OrdersByTerritoryView.

Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
RETURNS TABLE
AS
RETURN
(SELECT
        OrderID,
        OrderDate,
```

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

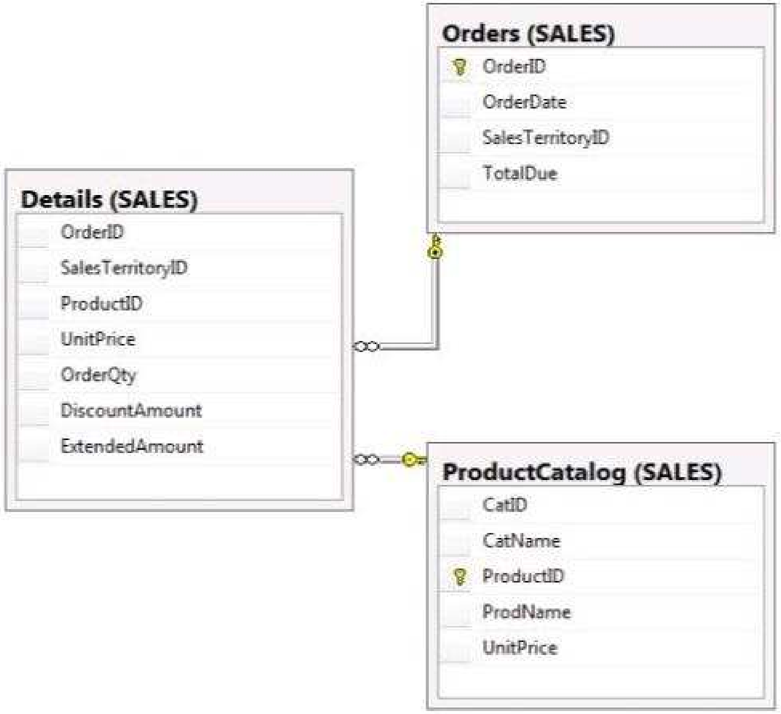
Explanation/Reference:

Explanation: CREATE FUNCTION Sales.fn_OrdersByTerritory (@T int) RETURNS TABLE AS RETURN (SELECT OrderID, OrderDate, SalesTerritoryID, TotalDue

FROM Sales.OrdersByTerritory WHERE SalesTerritoryID=@T)

QUESTION 66

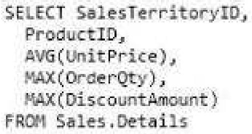
You have a database named Sales that contains the tables as shown in the exhibit. (Click the Exhibit button.)



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

- References columns by using one-part names only.
- Groups aggregates by SalesTerritoryID, and then by ProductID.
- Orders the results in descending order by SalesTerritoryID and then by ProductID.

Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
SELECT SalesTerritoryID,
  ProductID,
  AVG(UnitPrice),
  MAX(OrderQty),
  MAX(DiscountAmount)
FROM Sales.Details
```

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

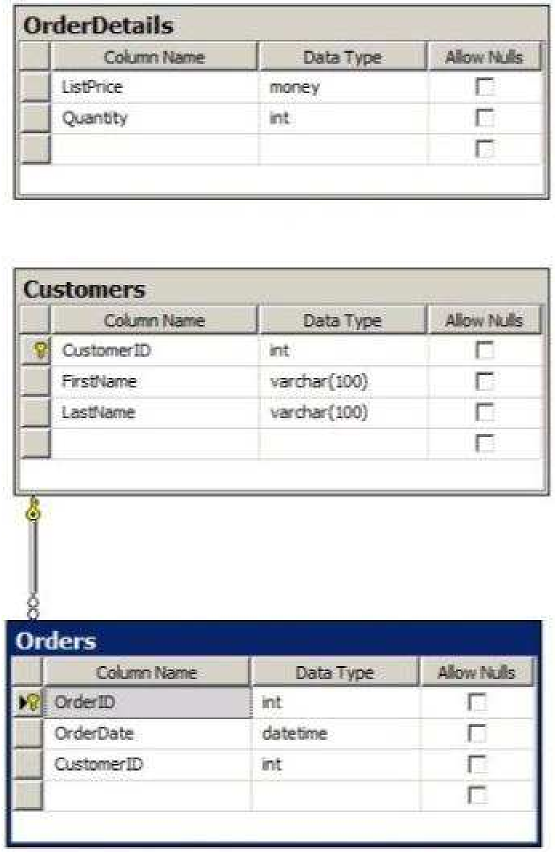
Explanation/Reference:

Explanation: SELECT SalesTerritoryID,

ProductID, AVG(UnitPrice), MAX(OrderQty) MAX(DiscountAmount) FROM Sales.Details GROUP BY SalesTerritoryID, ProductID ORDER BY SalesTerritoryID DESC, ProductID DESC

QUESTION 67

You have a database named Sales that contains the tables shown in the exhibit. (Click the Exhibit button).



OrderDetails

Column Name	Data Type	Allow Nulls
ListPrice	money	□
Quantity	int	□
		□

Customers

Column Name	Data Type	Allow Nulls
CustomerID	int	□
FirstName	varchar(100)	□
LastName	varchar(100)	□
		□

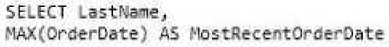
Orders

Column Name	Data Type	Allow Nulls
OrderID	int	□
OrderDate	datetime	□
CustomerID	int	□
		□

You need to create a query for a report. The query must meet the following requirements:

- NOT use object delimiters.
- Use the first initial of the table as an alias.
- Return the most recent order date for each customer.
- Retrieve the last name of the person who placed the order.

The solution must support the ANSI SQL-99 standard. Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



```
SELECT LastName,
MAX(OrderDate) AS MostRecentOrderDate
```

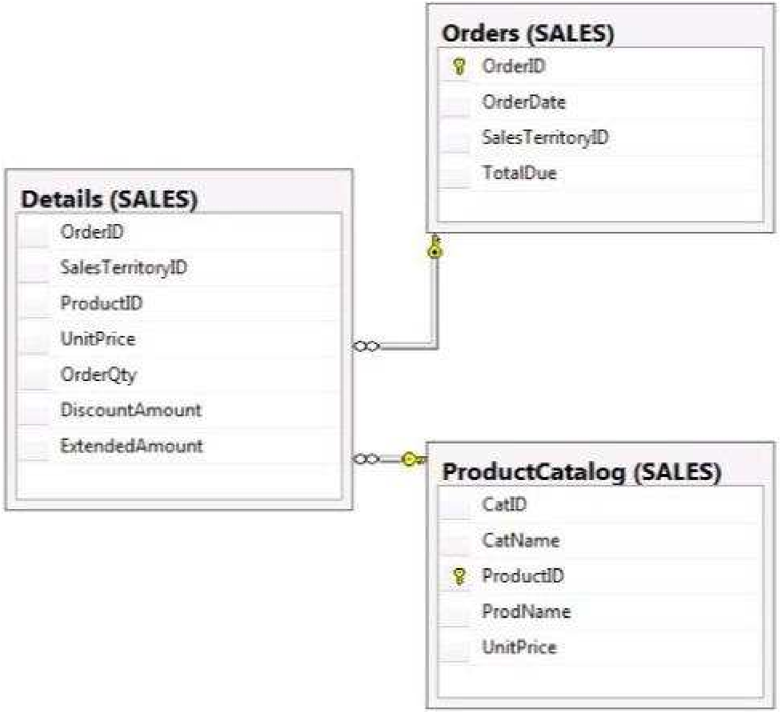
Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: SELECT C.LastName, MAX(O.OrderDate) AS MostRecentOrderDate FROM Customers AS C INNER JOIN Orders AS O ON C.CustomerID=O.CustomerID GROUP BY C.Lastname ORDER BY MAX (O.OrderDate) DESC

QUESTION 68

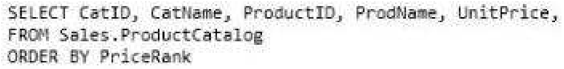
You have a database named Sales that contains the tables as shown in the exhibit. (Click the Exhibit button.)



You need to create a query that returns a list of products from Sales.ProductCatalog. The solution must meet the following requirements:

- UnitPrice must be returned in descending order.
- The query must use two-part names to reference the table.
- The query must use the RANK function to calculate the results.
- The query must return the ranking of rows in a column named PriceRank.
- The list must display the columns in the order that they are defined in the table. PriceRank must appear last.

Part of the correct T-SQL statement has been provided in the answer area. Provide the complete code.



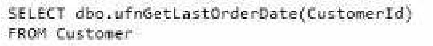
```
SELECT CatID, CatName, ProductID, ProdName, UnitPrice,
FROM Sales.ProductCatalog
ORDER BY PriceRank
```

Correct Answer: Answer: Please review the explanation part for this answer
Section: (none)
Explanation

Explanation/Reference:

Explanation: SELECT ProductCatalog.CatID, ProductCatalog.CatName, ProductCatalog.ProductID, ProductCatalog.ProdName, ProductCatalog.UnitPrice, RANK() OVER (ORDER BY ProductCatalog.UnitPrice DESC) AS PriceRank FROM Sales.ProductCatalog ORDER BY ProductCatalog.UnitPrice DESC

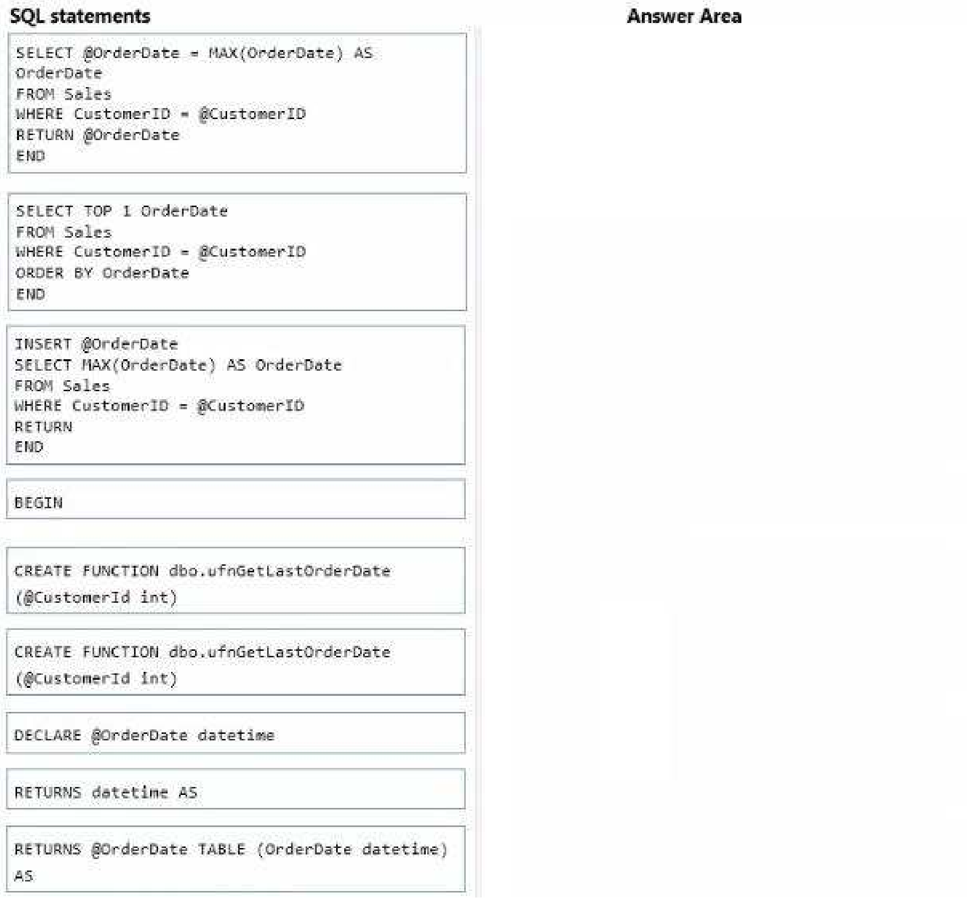
QUESTION 69
You write the following SELECT statement to get the last order date for a particular customer.



```
SELECT dbo.ufnGetLastOrderDate(CustomerId)
FROM Customer
```

You need to create the user-defined function to return the last order date for the specified customer. Which five Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Select and Place:



SQL statements

Answer Area

```
SELECT @OrderDate = MAX(OrderDate) AS
OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
RETURN @OrderDate
END
```

```
SELECT TOP 1 OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
ORDER BY OrderDate
END
```

```
INSERT @OrderDate
SELECT MAX(OrderDate) AS OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
RETURN
END
```

```
BEGIN
```

```
CREATE FUNCTION dbo.ufnGetLastOrderDate
(@CustomerId int)
```

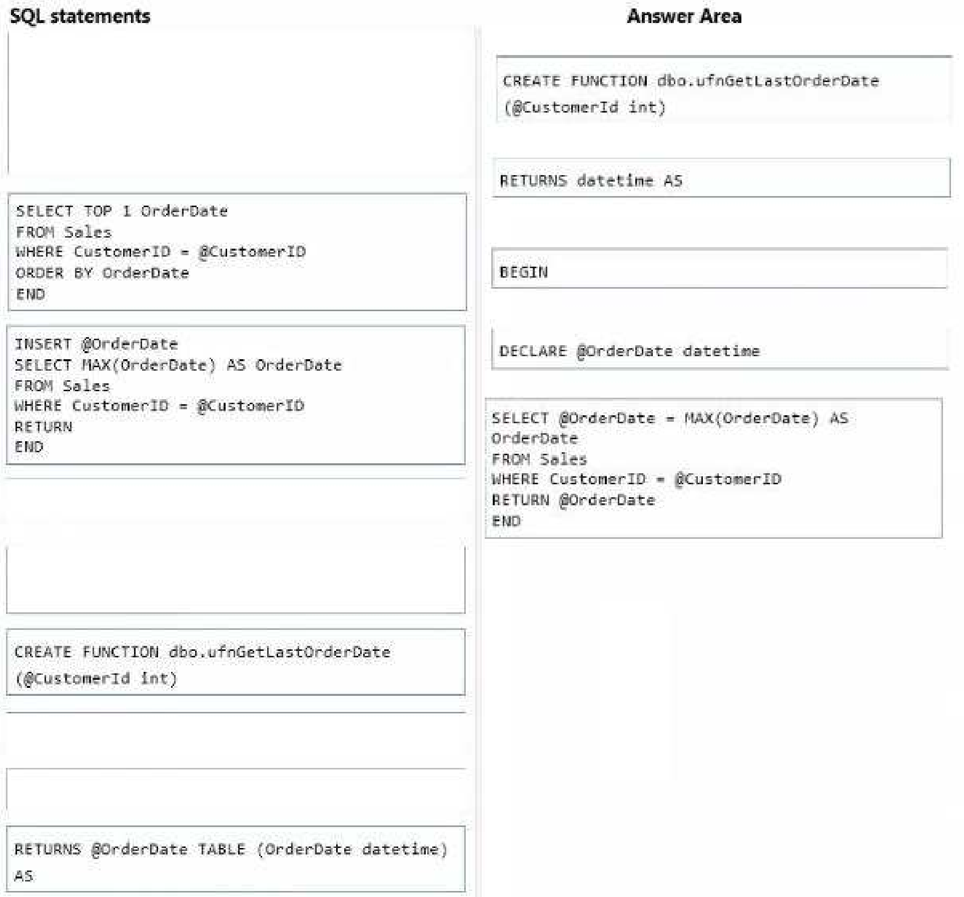
```
CREATE FUNCTION dbo.ufnGetLastOrderDate
(@CustomerId int)
```

```
DECLARE @OrderDate datetime
```

```
RETURNS datetime AS
```

```
RETURNS @OrderDate TABLE (OrderDate datetime)
AS
```

Correct Answer:



```
SELECT TOP 1 OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
ORDER BY OrderDate
END
```

```
INSERT @OrderDate
SELECT MAX(OrderDate) AS OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
RETURN
END
```

```
CREATE FUNCTION dbo.ufnGetLastOrderDate
(@CustomerId int)
```

```
RETURNS @OrderDate TABLE (OrderDate datetime)
AS
```

```
CREATE FUNCTION dbo.ufnGetLastOrderDate
(@CustomerId int)
```

```
RETURNS datetime AS
```

```
BEGIN
```

```
DECLARE @OrderDate datetime
```

```
SELECT @OrderDate = MAX(OrderDate) AS
OrderDate
FROM Sales
WHERE CustomerID = @CustomerID
RETURN @OrderDate
END
```

Section: (none)
Explanation

Explanation/Reference:

Note: * First function header * Then declare that the function returns a datetime
* Thirdly begin the function body.
* Fourthly declare the return variable
* At last include the code that retrieves the required date.

QUESTION 70

You are using SQL Server Management Studio (SSMS) to configure the backup. You need to meet the technical requirements.

Which two backup options should you configure? (Choose two).

A. Enable encryption of the backup file.
B. Enable compression of the backup file.
C. Disable encryption of the backup file.
D. Disable compression of the backup file.

Correct Answer: BC
Section: (none)
Explanation

Explanation/Reference: