



Vendor: Microsoft

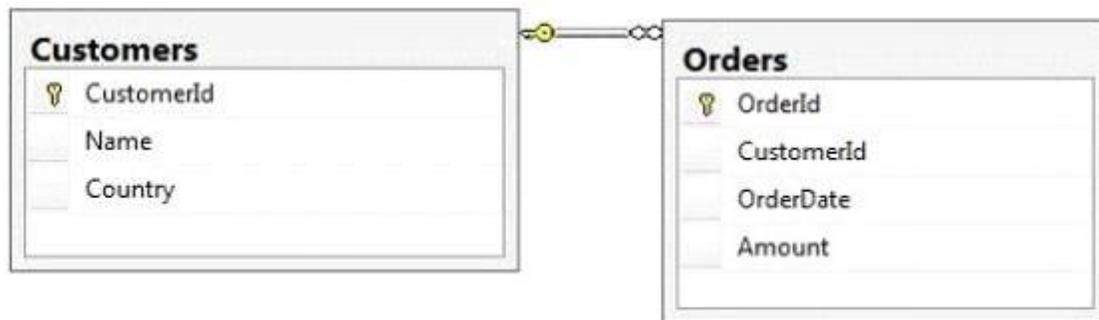
Exam Code: 70-457

**Exam Name: Transition Your MCTS on SQL Server 2008 to
MCSA: SQL Server 2012, Part 1**

Version: Demo

QUESTION 1

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.

```
<row OrderId= "1" orderDate="2000-01-01T00:00:00", Amount="3400.00" Name="Customer A"
Country="Australia" />
```

```
<row OrderId="2" OrderDate="2001-01-01T00:00:00" Amount="4300.00" Name="Customer A"
Country-"Australia" />
```

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')

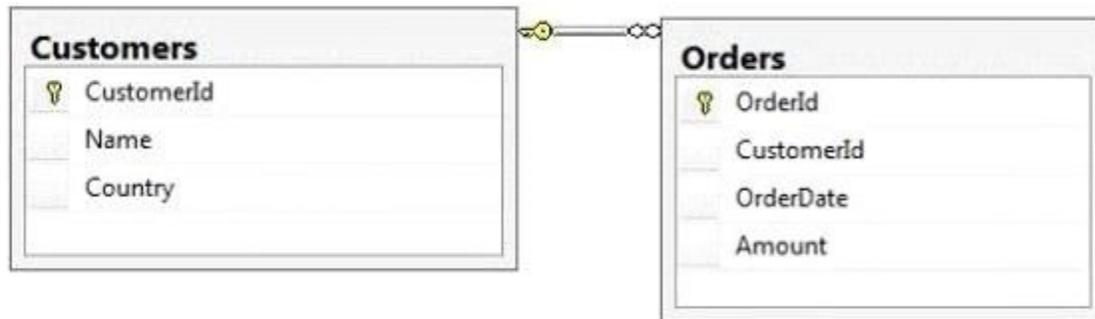
Correct Answer: A

Explanation:

<http://msdn.microsoft.com/en-us/library/bb510464.aspx>

QUESTION 2

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.

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

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')

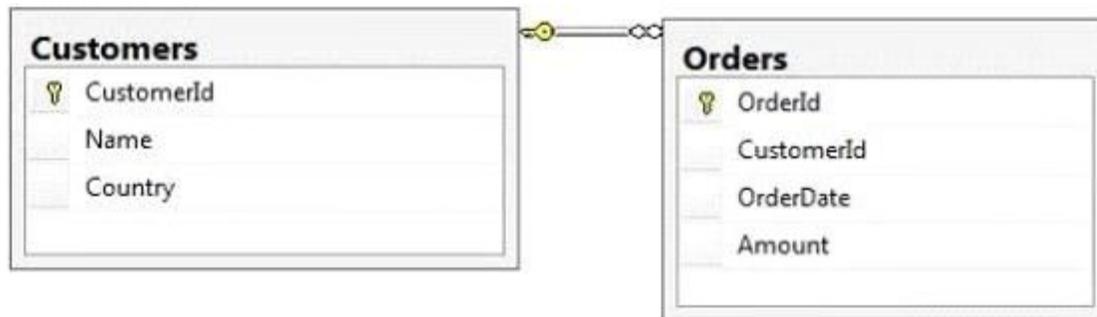
Correct Answer: C

Explanation:

<http://msdn.microsoft.com/en-us/library/ms188273.aspx>

QUESTION 3

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

Which Transact-SQL query should you use?

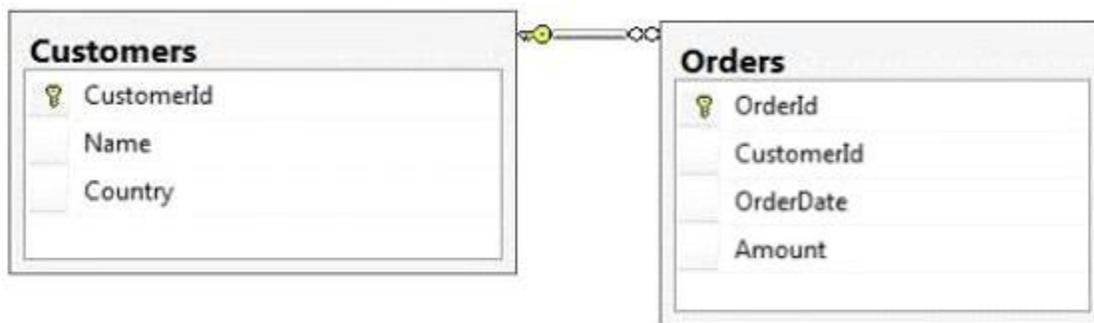
- A. SELECT OrderId, OrderDate, Amount, Name, Country
FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers.CustomerId
WHERE Customers.CustomerId = 1
FOR XML RAW
- B. SELECT OrderId, OrderDate, Amount, Name, Country
FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers.CustomerId

- 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: E

QUESTION 4

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.

```

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

```

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: D

QUESTION 5

You create a table that has the StudentCode, SubjectCode, and Marks columns to record mid-year marks for students. The table has marks obtained by 50 students for various subjects. You need to ensure that the top half of the students arranged by their average marks must be given a rank of 1 and the remaining students must be given a rank of 2. Which Transact-SQL query should you use?

- A.

```
SELECT StudentCode as Code,
RANK ( ) OVER (ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks GROUP BY StudentCode
```
- B.

```
SELECT Id, Name, Marks, DENSE_RANK ( ) OVER (ORDER BY Marks DESC) AS Rank
FROM StudentMarks
```
- C.

```
SELECT StudentCode as Code,
DENSE_RANK ( ) OVER (ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks GROUP BY StudentCode
```
- D.

```
SELECT StudentCode as Code,
NTILE (2) OVER (ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks GROUP BY StudentCode
```
- E.

```
SELECT StudentCode AS Code,Marks AS Value
FROM (SELECT StudentCode, Marks AS Marks,
RANK ( ) OVER (PARTITION BY SubjectCode ORDER BY Marks ASC) AS Rank FROM
StudentMarks)
tmp WHERE Rank = 1
```
- F.

```
SELECT StudentCode AS Code,Marks AS Value FROM (
SELECT StudentCode, Marks AS Marks,
RANK() OVER (PARTITION BY SubjectCode ORDER 3Y Marks DESC) AS Rank FROM
StudentMarks)
tmp WHERE Rank = 1
```
- G.

```
SELECT StudentCode AS Code,Marks AS Value
FROM (SELECT StudentCode, Marks AS Marks,
RANK ( ) OVER (PARTITION BY StudentCode ORDER BY Marks ASC) AS Rank FROM
StudentMarks)
tmp WHERE Rank = 1
```
- H.

```
SELECT StudentCode AS Code,Marks AS Value
FROM (SELECT StudentCode, Marks AS Marks,
RANXO OVER (PARTITION BY StudentCode ORDER BY Marks DESC) AS Rank FROM
StudentMarks)
tmp WHERE Rank = 1
```

Correct Answer: D

QUESTION 6

You create a table that has the StudentCode, SubjectCode, and Marks columns to record mid-year marks for students. The table has marks obtained by 50 students for various subjects.

You need to ensure that the following requirements are met:

- Students must be ranked based on their average marks.
- If one or more students have the same average, the same rank must be given to these students.
- Consecutive ranks must be skipped when the same rank is assigned.

Which Transact-SQL query should you use?

- A.

```
SELECT StudentCode as Code,
RANK() OVER(ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- B.

```
SELECT Id, Name, Marks,
DENSE_RANK() OVER(ORDER BY Marks DESC) AS Rank
FROM StudentMarks
```
- C.

```
SELECT StudentCode as Code,
DENSE_RANK() OVER(ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- D.

```
SELECT StudentCode as Code,
NTILE(2) OVER(ORDER BY AVG (Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- E.

```
SELECT StudentCode AS Code,Marks AS Value FROM (
SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY SubjectCode ORDER BY Marks ASC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- F.

```
SELECT StudentCode AS Code,Marks AS Value FROM (
SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY SubjectCode ORDER BY Marks DESC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- G.

```
SELECT StudentCode AS Code,Marks AS Value FROM (
SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY StudentCode ORDER BY Marks ASC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- H.

```
SELECT StudentCode AS Code,Marks AS Value FROM (
SELECT StudentCode, Marks AS Marks,
RANXO OVER(PARTITION BY StudentCode ORDER BY Marks DESC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```

Correct Answer: A

Explanation:

<http://msdn.microsoft.com/en-us/library/ms189798.aspx>

QUESTION 7

You create a table that has the StudentCode, SubjectCode, and Marks columns to record mid-year marks for students. The table has marks obtained by 50 students for various subjects. You need to retrieve the students who scored the highest marks for each subject along with the marks. Which Transact-SQL query should you use?

- A.

```
SELECT StudentCode as Code, RANK() OVER(ORDER BY AVG(Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- B.

```
SELECT Id, Name, Marks, DENSE_RANK() OVER(ORDER BY Marks DESC) AS Rank
FROM StudentMarks
```
- C.

```
SELECT StudentCode as Code, DENSE_RANK() OVER(ORDER BY AVG(Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- D.

```
SELECT StudentCode as Code, NTILE(2) OVER(ORDER BY AVG(Marks) DESC) AS Value
FROM StudentMarks
GROUP BY StudentCode
```
- E.

```
SELECT StudentCode AS Code, Marks AS Value FROM ( SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY SubjectCode ORDER BY Marks ASC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- F.

```
SELECT StudentCode AS Code, Marks AS Value FROM ( SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY SubjectCode ORDER BY Marks DESC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- G.

```
SELECT StudentCode AS Code, Marks AS Value FROM ( SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY StudentCode ORDER BY Marks ASC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```
- H.

```
SELECT StudentCode AS Code, Marks AS Value FROM ( SELECT StudentCode, Marks AS Marks,
RANK() OVER(PARTITION BY StudentCode ORDER BY Marks DESC) AS Rank
FROM StudentMarks) tmp
WHERE Rank = 1
```

Correct Answer: F

QUESTION 8

DRAG DROP

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.

```

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:

<pre> CREATE VIEW Sales.vwCustomerRevenue AS WITH SCHEMABINDING </pre>	<pre> CREATE VIEW Sales.vwCustomerRevenue AS WITH SCHEMABINDING </pre>
<pre> CREATE VIEW Sales.vwCustomerRevenue WITH SCHEMABINDING AS </pre>	<pre> 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 </pre>
<pre> 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 </pre>	<pre> GROUP BY O.CustomerID , C.CustomerName </pre>
<pre> GO CREATE UNIQUE CLUSTERED INDEX idx_vwCustomerRevenue ON Sales.vwCustomerRevenue (CustomerID); </pre>	<pre> GO CREATE UNIQUE CLUSTERED INDEX idx_vwCustomerRevenue ON Sales.vwCustomerRevenue (CustomerID); </pre>
<pre> GO CREATE UNIQUE INDEX idx_vwCustomerRevenue ON Sales.vwCustomerRevenue (CustomerID); </pre>	

QUESTION 9

You develop a Microsoft SQL Server 2012 server database that supports an application. The application contains a table that has the following definition:

```
CREATE TABLE Inventory (  
    ItemID int NOT NULL PRIMARY KEY,  
    ItemsInStore int NOT NULL,  
    ItemsInWarehouse int NOT NULL)
```

You need to create a computed column that returns the sum total of the ItemsInStore and ItemsInWarehouse values for each row.

The new column is expected to be queried heavily, and you need to be able to index the column.

Which Transact-SQL statement should you use?

- A. ALTER TABLE Inventory
ADD TotalItems AS ItemsInStore + ItemsInWarehouse
- B. ALTER TABLE Inventory
ADD TotalItems AS ItemsInStore + ItemsInWarehouse PERSISTED
- C. ALTER TABLE Inventory
ADD TotalItems AS SUM(ItemsInStore, ItemsInWarehouse) PERSISTED
- D. ALTER TABLE Inventory
ADD TotalItems AS SUM(ItemsInStore, ItemsInWarehouse)

Correct Answer: B

Explanation:

<http://msdn.microsoft.com/en-us/library/ms174979.aspx>

QUESTION 10

You develop a Microsoft SQL Server 2012 database that contains a table named Customers.

The Customers table has the following definition:

```
CREATE TABLE [dbo].[Customers] (  
    [CustomerId] [bigint] NOT NULL,  
    [MobileNumber] [nvarchar](25) NOT NULL,  
    [HomeNumber] [nvarchar](25) NULL,  
    [Name] [nvarchar](50) NOT NULL,  
    [Country] [nvarchar](25) NOT NULL,  
    CONSTRAINT [PK_Customers] PRIMARY KEY CLUSTERED  
    (  
        [CustomerId] ASC  
    ) ON [PRIMARY]  
    ) ON [PRIMARY]
```

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

Which Transact-SQL query should you use?

- A. CREATE TRIGGER TrgPhoneNumberChange
ON Customers FOR UPDATE
AS
IF COLUMNS_UPDATED (HomeNumber, MobileNumber) - Create Audit Records
- B. CREATE TRIGGER TrgPhoneNumberChange
ON Customers FOR UPDATE
AS
IF EXISTS(SELECT HomeNumber FROM inserted) OR
EXISTS (SELECT MobileNumber FROM inserted) - Create Audit Records
- C. CREATE TRIGGER TrgPhoneNumberChange
ON Customers FOR UPDATE
AS
IF COLUMNS_CHANGED (HomeNumber, MobileNumber) - Create Audit Records
- D. CREATE TRIGGER TrgPhoneNumberChange
ON Customers FOR UPDATE
AS
IF UPDATE (HomeNumber) OR UPDATE (MobileNumber) - Create Audit Records

Correct Answer: D

Explanation:

<http://msdn.microsoft.com/en-us/library/bb510663.aspx>

<http://msdn.microsoft.com/en-us/library/ms186329.aspx>

QUESTION 11

You develop a Microsoft SQL Server 2012 database that has two tables named SavingAccounts and LoanAccounts. Both tables have a column named AccountNumber of the nvarchar data type. You use a third table named Transactions that has columns named TransactionId, AccountNumber, Amount, and TransactionDate. You need to ensure that when multiple records are inserted in the Transactions table, only the records that have a valid AccountNumber in the SavingAccounts or LoanAccounts are inserted. Which Transact-SQL statement should you use?

- A. CREATE TRIGGER TrgValidateAccountNumber
ON Transactions
INSTEAD OF INSERT
AS
BEGIN
INSERT INTO Transactions
SELECT TransactionID,AccountNumber,Amount,TransactionDate FROM inserted
WHERE AccountNumber IN
(SELECT AccountNumber FROM LoanAccounts
UNION SELECT AccountNumber FROM SavingAccounts))
END
- B. CREATE TRIGGER TrgValidateAccountNumber
ON Transactions
FOR INSERT
AS
BEGIN
INSERT INTO Transactions
SELECT TransactionID,AccountNumber,Amount,TransactionDate FROM inserted
WHERE AccountNumber IN
(SELECT AccountNumber FROM LoanAccounts

- ```
UNION SELECT AccountNumber FROM SavingAccounts))
END
C. CREATE TRIGGER TrgValidateAccountNumber
ON Transactions
INSTEAD OF INSERT
AS
BEGIN
IF EXISTS (
SELECT AccountNumber FROM inserted EXCEPT
(SELECT AccountNumber FROM LoanAccounts
UNION SELECT AccountNumber FROM SavingAccounts)) BEGIN
ROLLBACK TRAN
END
END
D. CREATE TRIGGER TrgValidateAccountNumber
ON Transactions
FOR INSERT
AS
BEGIN
IF EXISTS (
SELECT AccountNumber FROM inserted EXCEPT
(SELECT AccountNumber FROM LoanAccounts
UNION SELECT AccountNumber FROM SavingAccounts)) BEGIN
ROLLBACK TRAN
END
END
```

**Correct Answer:** A

### QUESTION 12

You develop a Microsoft SQL Server 2012 database.

You create a view that performs the following tasks:

- Joins 8 tables that contain up to 500,000 records each.
- Performs aggregations on 5 fields.

The view is frequently used in several reports. You need to improve the performance of the reports.

What should you do?

- Convert the view into a table-valued function.
- Convert the view into a Common Table Expression (CTE).
- Convert the view into an indexed view.
- Convert the view into a stored procedure and retrieve the result from the stored procedure into a temporary table.

**Correct Answer:** C

#### Explanation:

<http://msdn.microsoft.com/en-us/library/ms191432.aspx>

### QUESTION 13

You are a database developer of a Microsoft SQL Server 2012 database.

The database contains a table named Customers that has the following definition:

```
CREATE TABLE Customer
(CustomerID INT NOT NULL PRIMARY KEY,
 CustomerName VARCHAR(255) NOT NULL,
 CustomerAddress VARCHAR(1000) NOT NULL)
```

You are designing a new table named Orders that has the following definition:

```
CREATE TABLE Orders
(OrderID INT NOT NULL PRIMARY KEY,
 CustomerID INT NOT NULL,
 OrderDescription VARCHAR(2000))
```

You need to ensure that the CustomerId column in the Orders table contains only values that exist in the CustomerId column of the Customer table.

Which Transact-SQL statement should you use?

- A. ALTER TABLE Orders  
ADD CONSTRAINT FX\_Orders\_CustomerID FOREIGN KEY (CustomerId) REFERENCES  
Customer  
(CustomerId)
- B. ALTER TABLE Customer  
ADD CONSTRAINT FK\_Customer\_CustomerID FOREIGN KEY (CustomerId) REFERENCES  
Orders  
(CustomerId)
- C. ALTER TABLE Orders  
ADD CONSTRAINT CK\_Crders\_CustomerID  
CHECK (CustomerId IN (SELECT CustomerId FROM Customer))
- D. ALTER TABLE Customer  
ADD OrderId INT NOT NULL;  
ALTER TABLE Customer  
ADD CONSTRAINT FK\_Customer\_OrderID FOREIGN KEY (CrderID) REFERENCES Orders  
(CrderID);
- E. ALTER TABLE Orders  
ADD CONSTRAINT PK Orders CustomerId PRIMARY KEY (CustomerId)

**Correct Answer:** A

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms189049.aspx>

### QUESTION 14

You have three tables that contain data for dentists, psychiatrists, and physicians. You create a view that is used to look up their email addresses and phone numbers.

The view has the following definition:

```
Create view apt.vwProviderList
(Specialty, CompanyID, CompanyNumber, LastName,
 FirstName, BusinessName, Email, Phone)
as

SELECT 'Dentist' as Specialty
, DentistID
, DentistNumber
, DentistLastName
, DentistFirstName
, DentistBusinessName
, Email
, Phone
FROM apt.Dentist
UNION ALL
SELECT 'Psychiatrist' as Specialty
, PsychiatristID
, PsychiatristNumber
, PsychiatristLastName
, PsychiatristFirstName
, PsychiatristBusinessName
, Email
, Phone
SELECT 'Physician' as Specialty
, PhysicianID
, PhysicianNumber
, PhysicianLastName
, PhysicianFirstName
, PhysicianBusinessName
, Email
, Phone
FROM apt.Physician
GO
```

You need to ensure that users can update only the phone numbers and email addresses by using this view.

What should you do?

- A. Alter the view. Use the EXPAND VIEWS query hint along with each SELECT statement.
- B. Create an INSTEAD OF UPDATE trigger on the view.
- C. Drop the view. Re-create the view by using the SCHEMABINDING clause, and then create an index on the view.
- D. Create an AFTER UPDATE trigger on the view.

**Correct Answer: B**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>

**QUESTION 15**

You develop a Microsoft SQL Server 2012 database. You create a view from the Orders and OrderDetails tables by using the following definition.

```
CREATE VIEW vOrders
WITH SCHEMABINDING
AS
SELECT o.ProductID,
 o.OrderDate,
 SUM(od.UnitPrice * od.OrderQty) AS Amount
FROM OrderDetails AS od INNER JOIN
 Orders AS o ON od.OrderID = o.OrderID
WHERE od.SalesOrderID = o.SalesOrderID
GROUP BY o.OrderDate, o.ProductID
GO
```

You need to ensure that users are able to modify data by using the view.

What should you do?

- A. Create an AFTER trigger on the view.
- B. Modify the view to use the WITH VIEW\_METADATA clause.
- C. Create an INSTEAD OF trigger on the view.
- D. Modify the view to an indexed view.

**Correct Answer: C**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>

**QUESTION 16**

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:

- Accept the @T integer parameter.
- Use one-part names to reference columns.

- Filter the query results sorted 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:**

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

**QUESTION 17**

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     | <input type="checkbox"/> |
|              | Quantity    | int       | <input type="checkbox"/> |
|              |             |           | <input type="checkbox"/> |

| Customers |             |              |                          |
|-----------|-------------|--------------|--------------------------|
|           | Column Name | Data Type    | Allow Nulls              |
| 🔑         | CustomerID  | int          | <input type="checkbox"/> |
|           | FirstName   | varchar(100) | <input type="checkbox"/> |
|           | LastName    | varchar(100) | <input type="checkbox"/> |
|           |             |              | <input type="checkbox"/> |

| Orders |             |           |                          |
|--------|-------------|-----------|--------------------------|
|        | Column Name | Data Type | Allow Nulls              |
| 🔑      | OrderID     | int       | <input type="checkbox"/> |
|        | OrderDate   | datetime  | <input type="checkbox"/> |
|        | CustomerID  | int       | <input type="checkbox"/> |
|        |             |           | <input type="checkbox"/> |

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 LineltemTotal that stores the product of ListPrice and Quantity for each row.
- The code must NOT use any object delimiters.

The solution must ensure that LineltemTotal 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.

- A. CREATE TABLE Sales.OrderDetails (  
ListPrice money not null,  
Quantity int not null,  
LineltemTotal as (ListPrice \* Quantity) PERSISTED)
- B. CREATE TABLE Sales.OrderDetails (  
ListPrice money not null,  
Quantity int not null,  
LineltemTotal as (ListPrice \* Quantity))

**Correct Answer:** A

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms174979.aspx>

<http://technet.microsoft.com/en-us/library/ms188300.aspx>

#### **QUESTION 18**

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     | <input type="checkbox"/> |
|              | Quantity    | int       | <input type="checkbox"/> |
|              |             |           | <input type="checkbox"/> |

| Customers |             |              |                          |
|-----------|-------------|--------------|--------------------------|
|           | Column Name | Data Type    | Allow Nulls              |
| ?         | CustomerID  | int          | <input type="checkbox"/> |
|           | FirstName   | varchar(100) | <input type="checkbox"/> |
|           | LastName    | varchar(100) | <input type="checkbox"/> |
|           |             |              | <input type="checkbox"/> |

| Orders |             |           |                          |
|--------|-------------|-----------|--------------------------|
|        | Column Name | Data Type | Allow Nulls              |
| ?      | OrderID     | int       | <input type="checkbox"/> |
|        | OrderDate   | datetime  | <input type="checkbox"/> |
|        | CustomerID  | int       | <input type="checkbox"/> |
|        |             |           | <input type="checkbox"/> |

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.

- A. CREATE VIEW Sales.uv\_CustomerFullName  
with Schemabinding  
AS  
SELECT FirstName, LastName  
FROM Customers
- B. CREATE VIEW Sales.uv\_CustomerFullName  
AS

```
SELECT FirstName, LastName
FROM Customers
```

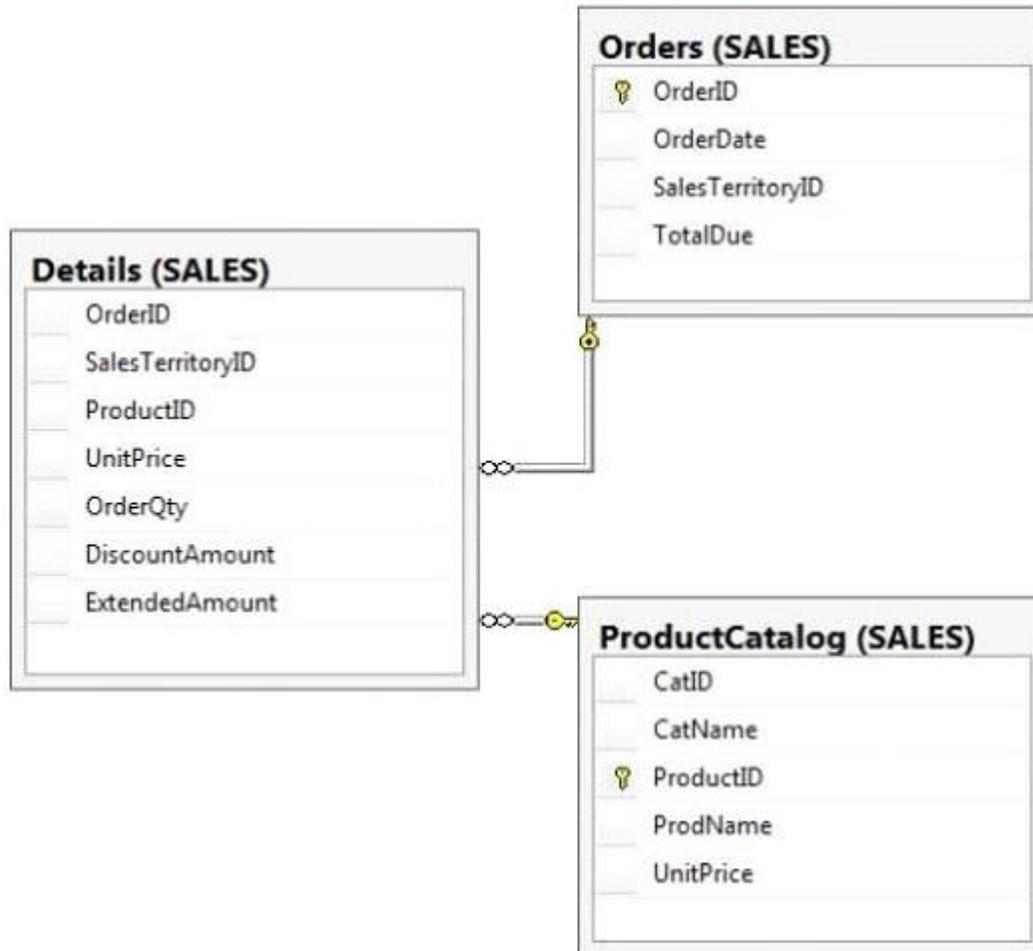
**Correct Answer:** A

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>

### QUESTION 19

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.
- 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.

Which code segment should you use?

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

- A. 

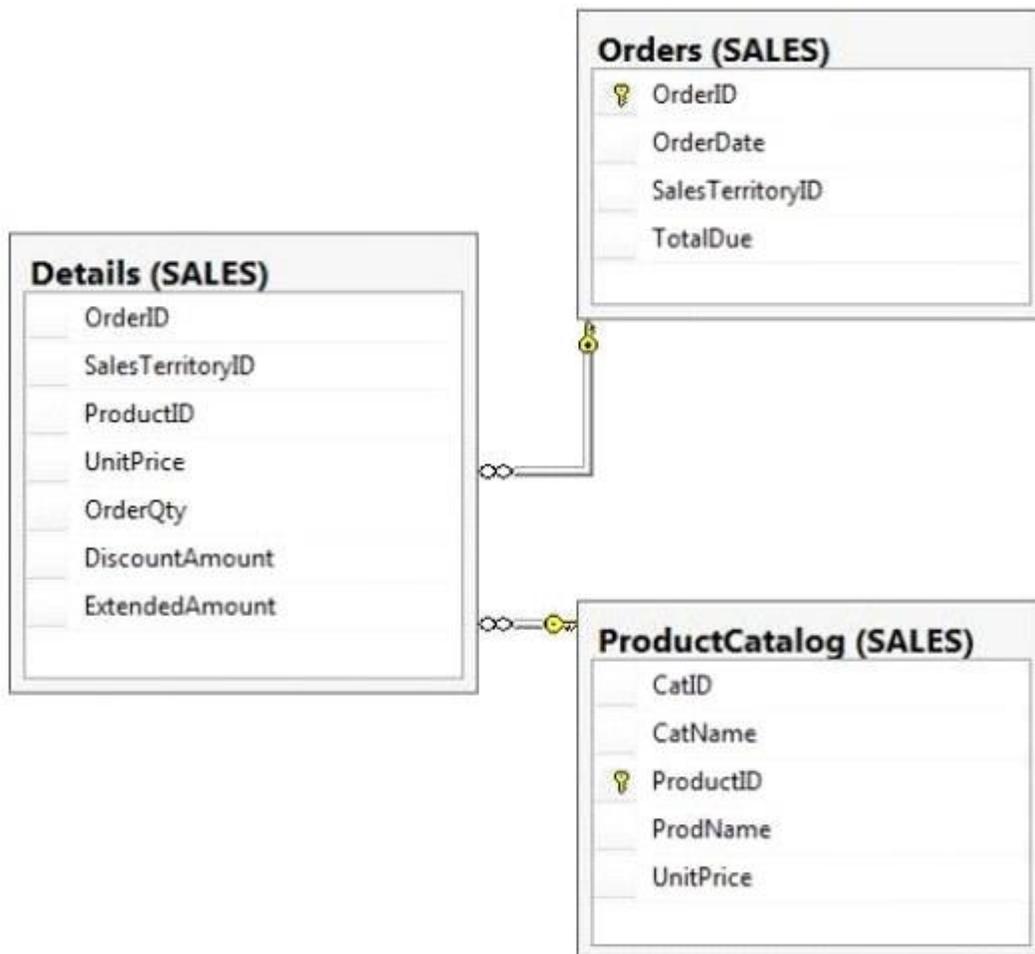
```
SELECT OrderID, SUM(ExtendedAmount) AS TotalSales
FROM Sales.Details
GROUP BY OrderID
ORDER BY OrderID
```
- B. 

```
SELECT OrderID, SUM(ExtendedAmount) AS TotalSales
FROM Sales.Details
ORDER BY OrderID
```

**Correct Answer:** A

**QUESTION 20**

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.

- A. 

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

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

**Correct Answer:** A

#### QUESTION 21

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     | <input type="checkbox"/> |
|              | Quantity    | int       | <input type="checkbox"/> |
|              |             |           | <input type="checkbox"/> |

| Customers |             |              |                          |
|-----------|-------------|--------------|--------------------------|
|           | Column Name | Data Type    | Allow Nulls              |
| 🔑         | CustomerID  | int          | <input type="checkbox"/> |
|           | FirstName   | varchar(100) | <input type="checkbox"/> |
|           | LastName    | varchar(100) | <input type="checkbox"/> |
|           |             |              | <input type="checkbox"/> |

| Orders |             |           |                          |
|--------|-------------|-----------|--------------------------|
|        | Column Name | Data Type | Allow Nulls              |
| 🔑      | OrderID     | int       | <input type="checkbox"/> |
|        | OrderDate   | datetime  | <input type="checkbox"/> |
|        | CustomerID  | int       | <input type="checkbox"/> |
|        |             |           | <input type="checkbox"/> |

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.

- A. SELECT c.CustomerID -- optional
- B. LastName, max(o.OrderDate) 'MostRecentOrderDate'  
FROM Customer c LEFT OUTER JOIN Orders o ON o.CustomerID = c.CustomerID  
GROUP BY c.CustomerID, c.LastName  
ORDER BY 3 DESC
- C. select C.Lastname, P.MostRecentOrderDate  
from customers AS C inner join  
(  
select customID, MostRecentOrderDate=max(orderDate) from orders  
group by customID  
) P  
on C.customerID=P.CustomerID  
order by P.MostRecentOrderDate desc
- D. SELECT C.LastName, O.OrderDate AS MostRecentOrderDate  
FROM Customers AS C INNER JOIN Orders AS O  
ON C.CustomerID = O.CustomerID  
ORDER BY O.OrderDate DESC

**Correct Answer:** A

#### QUESTION 22

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.

- A. Declare @XML1=XML(Sales.InvoiceSchema)
- B. DECLARE @XML1 XML  
@XML1 = Sales.InvoiceSchema  
CREATE XML SCHEMA COLLECTION XML1 AS @XML1

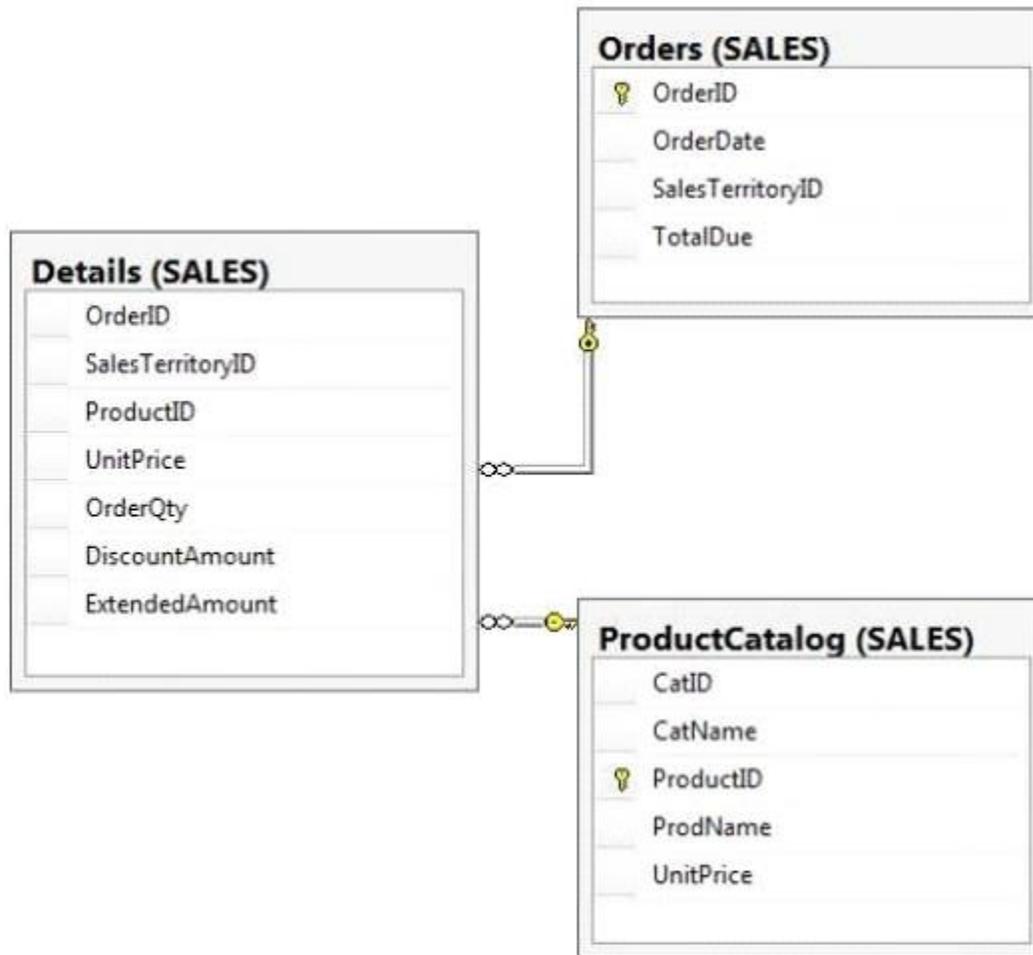
**Correct Answer:** A

#### Explanation:

<http://msdn.microsoft.com/en-us/library/ms176009.aspx>

#### QUESTION 23

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.

- A. `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`
- B. `SELECT ProductCatalog.CatID, ProductCatalog.CatName, ProductCatalog.ProductID, ProductCatalog.ProdName, ProductCatalog.UnitPrice, RANK() OVER (PARTITION BY ProductCatalog.UnitPrice ORDER BY ProductCatalog.`

```
UnitPrice DESC) AS PriceRank
FROM Sales.ProductCatalog
ORDER BY ProductCatalog.UnitPrice DESC
```

**Correct Answer:** A

**QUESTION 24**

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     | <input type="checkbox"/> |
|              | Quantity    | int       | <input type="checkbox"/> |
|              |             |           | <input type="checkbox"/> |

| Customers |             |              |                          |
|-----------|-------------|--------------|--------------------------|
|           | Column Name | Data Type    | Allow Nulls              |
| 🔑         | CustomerID  | int          | <input type="checkbox"/> |
|           | FirstName   | varchar(100) | <input type="checkbox"/> |
|           | LastName    | varchar(100) | <input type="checkbox"/> |
|           |             |              | <input type="checkbox"/> |

| Orders |             |           |                          |
|--------|-------------|-----------|--------------------------|
|        | Column Name | Data Type | Allow Nulls              |
| 🔑      | OrderID     | int       | <input type="checkbox"/> |
|        | OrderDate   | datetime  | <input type="checkbox"/> |
|        | CustomerID  | int       | <input type="checkbox"/> |
|        |             |           | <input type="checkbox"/> |

You have an application named App1. 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. 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:**

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

**QUESTION 25**

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     | <input type="checkbox"/> |
|              | Quantity    | int       | <input type="checkbox"/> |
|              |             |           | <input type="checkbox"/> |

| Customers |             |              |                          |
|-----------|-------------|--------------|--------------------------|
|           | Column Name | Data Type    | Allow Nulls              |
|           | CustomerID  | int          | <input type="checkbox"/> |
|           | FirstName   | varchar(100) | <input type="checkbox"/> |
|           | LastName    | varchar(100) | <input type="checkbox"/> |
|           |             |              | <input type="checkbox"/> |

| Orders |             |           |                          |
|--------|-------------|-----------|--------------------------|
|        | Column Name | Data Type | Allow Nulls              |
|        | OrderID     | int       | <input type="checkbox"/> |
|        | OrderDate   | datetime  | <input type="checkbox"/> |
|        | CustomerID  | int       | <input type="checkbox"/> |
|        |             |           | <input type="checkbox"/> |

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.

- A. 

```
SELECT OrderID
FROM Orders
WHERE OrderDate>CONVERT(datetime,@Parameter1)
```
- B. 

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

**Correct Answer:** A

#### QUESTION 26

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

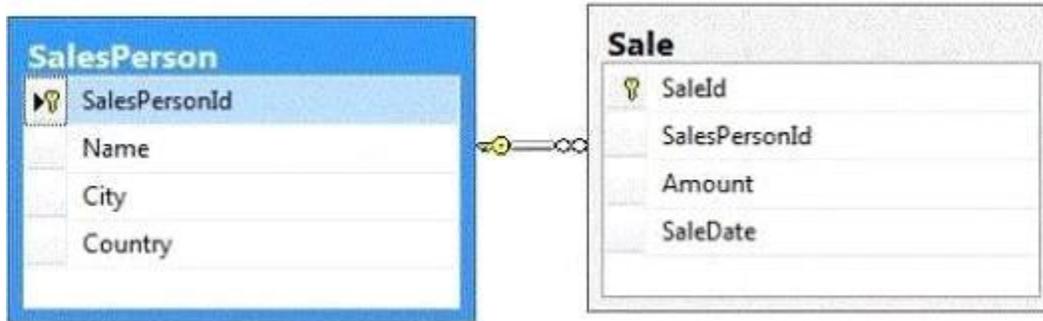
**Correct Answer:** D

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms175156.aspx>

**QUESTION 27**

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

**Explanation:**

<http://www.grapefruitmoon.net/diving-into-t-sql-grouping-sets/>

<http://msdn.microsoft.com/en-us/library/ms177673.aspx>

### QUESTION 28

You are a database developer for an application hosted on a Microsoft SQL Server 2012 server.

The database contains two tables that have the following definitions:

```
CREATE TABLE Customer
(CustomerID int NOT NULL PRIMARY KEY,
 CustomerName varchar(50) NOT NULL)
```

```
CREATE TABLE Orders
(OrderID int NOT NULL PRIMARY KEY,
 CustomerID int NOT NULL FOREIGN KEY REFERENCES Customer (CustomerID),
 OrderAmount money NOT NULL,
 ShippingCountry varchar(50) NOT NULL)
```

Global customers place orders from several countries.

You need to view the country from which each customer has placed the most orders.

Which Transact-SQL query do you use?

- A. 

```
SELECT c.CustomerID, c.CustomerName, o.ShippingCountry
FROM Customer c
INNER JOIN
(SELECT CustomerID, ShippingCountry,
RANK() OVER (PARTITION BY CustomerID
ORDER BY COUNT(OrderAmount) DESC) AS Rnk
FROM Orders
GROUP BY CustomerID, ShippingCountry) AS o
ON c.CustomerID = o.CustomerID
WHERE o.Rnk = 1
```
- B. 

```
SELECT c.CustomerID, c.CustomerName, o.ShippingCountry
FROM
(SELECT c.CustomerID, c.CustomerName, o.ShippingCountry,
RANK() OVER (PARTITION BY CustomerID
ORDER BY COUNT(o.OrderAmount) ASC) AS Rnk
FROM Customer c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
GROUP BY c.CustomerID, c.CustomerName, o.ShippingCountry) cs
WHERE Rnk = 1
```
- C. 

```
SELECT c.CustomerID, c.CustomerName, o.ShippingCountry
FROM Customer c
INNER JOIN
(SELECT CustomerID, ShippingCountry,
RANK() OVER (PARTITION BY CustomerID
ORDER BY OrderAmount DESC) AS Rnk
FROM Orders
GROUP BY CustomerID, ShippingCountry) AS o
ON c.CustomerID = o.CustomerID
WHERE o.Rnk = 1
```
- D. 

```
SELECT c.CustomerID, c.CustomerName, o.ShippingCountry
FROM Customer c
INNER JOIN
(SELECT CustomerID, ShippingCountry,
```

```
COUNT(OrderAmount) DESC) AS OrderAmount
FROM Orders
GROUP BY CustomerID, ShippingCountry) AS o
ON c.CustomerID = o.CustomerID
ORDER BY OrderAmount DESC
```

**Correct Answer:** A

#### QUESTION 29

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

**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 30

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

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms188047.aspx>

<http://msdn.microsoft.com/en-us/library/ms177682.aspx>

<http://msdn.microsoft.com/en-us/library/ms173545.aspx>

## EnsurePass.com Members Features:

1. Verified Answers researched by industry experts.
2. Q&As are downloadable in PDF and VCE format.
3. 98% success Guarantee and **Money Back** Guarantee.
4. Free updates for **180** Days.
5. **Instant Access to download the Items**

View list of All Exam provided:

<http://www.ensurepass.com/certifications?index=A>

To purchase Lifetime Full Access Membership click here:

<http://www.ensurepass.com/user/register>

**Valid Discount Code for 2015: JREH-G1A8-XHC6**

To purchase the HOT Microsoft Exams:

| <u>Microsoft</u>       |                        |                        |                        |
|------------------------|------------------------|------------------------|------------------------|
| <a href="#">70-243</a> | <a href="#">70-347</a> | <a href="#">70-466</a> | <a href="#">70-515</a> |
| <a href="#">70-246</a> | <a href="#">70-410</a> | <a href="#">70-467</a> | <a href="#">70-516</a> |
| <a href="#">70-247</a> | <a href="#">70-411</a> | <a href="#">70-480</a> | <a href="#">70-519</a> |
| <a href="#">70-321</a> | <a href="#">70-412</a> | <a href="#">70-483</a> | <a href="#">70-583</a> |
| <a href="#">70-331</a> | <a href="#">70-413</a> | <a href="#">70-484</a> | <a href="#">70-640</a> |
| <a href="#">70-332</a> | <a href="#">70-414</a> | <a href="#">70-485</a> | <a href="#">70-649</a> |
| <a href="#">70-336</a> | <a href="#">70-417</a> | <a href="#">70-486</a> | <a href="#">70-668</a> |
| <a href="#">70-337</a> | <a href="#">70-461</a> | <a href="#">70-487</a> | <a href="#">70-680</a> |
| <a href="#">70-341</a> | <a href="#">70-462</a> | <a href="#">70-488</a> | <a href="#">70-687</a> |
| <a href="#">70-342</a> | <a href="#">70-463</a> | <a href="#">70-489</a> | <a href="#">70-688</a> |
| <a href="#">70-346</a> | <a href="#">70-464</a> | <a href="#">70-513</a> | <a href="#">70-689</a> |

