

KillTest

更に上のクオリティ 更に上のサービス



問題集

<http://www.killtest.jp>

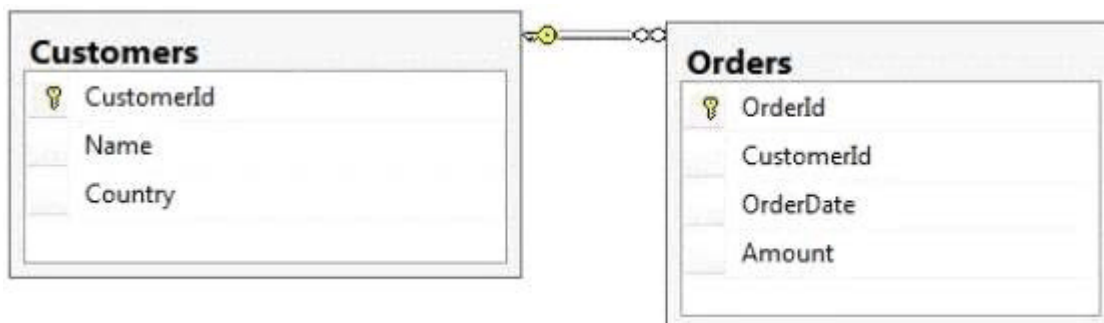
1年で無料進級することに提供する

Exam : **070-457J**

Title : Transition Your MCTS on
SQL Server 2008 to MCSA:
SQL Server 2012, Part 1

Version : DEMO

1. あなたが ContosoDb 付きの Microsoft SQL Server 2012 のデータベースを管理します。展示に示すように、テーブルが定義されています。(図表ボタンをクリックします。)



あなたは、顧客の注文テーブルから行を表示する必要があると、次の XML 形式で 1 に設定 CustomerId の値を持つ行:

```
<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
```

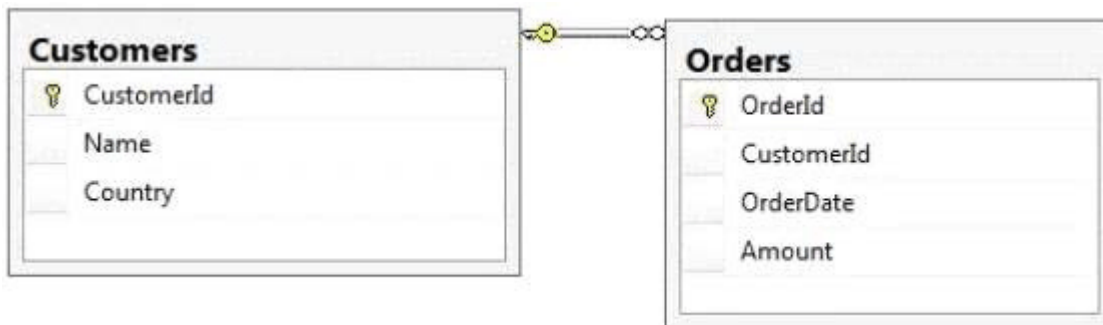
```

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')

```

Answer: A

2. あなたが ContosoDb 付きの Microsoft SQL Server 2012 のデータベースを管理する。展示に示すように、テーブルが定義されている。(図表ボタンをクリックします。)



あなたは、顧客の注文テーブルから行を表示する必要があると、次のXML形式で1にCustomerIdvalueセットを持つ行です。

```

<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>

```

あなたはどちらの Transact-SQL クエリを使うべきでしょうか？

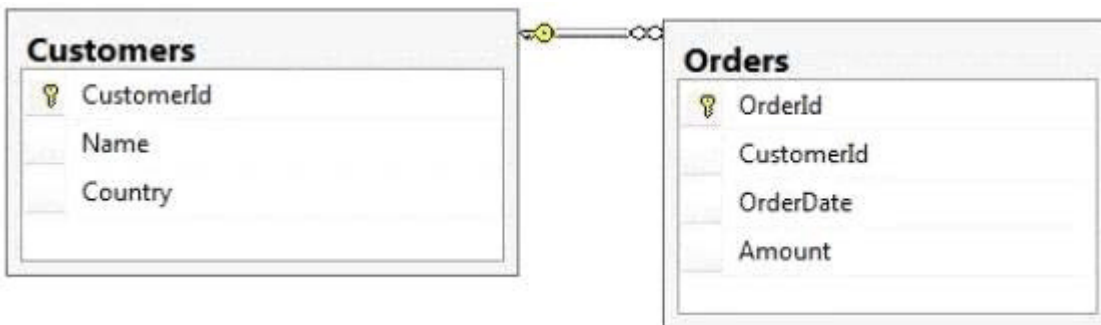
- A. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers. CustomerId WHERE Customers.CustomerId = 1 FOR XML RAW
- B. SELECT OrderId, OrderDate, Amount, Name, Country FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers. CustomerId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

Answer: C

3. あなたが ContosoDb 付きの Microsoft SQL Server 2012 のデータベースを管理します。展示に示すように、テーブルが定義されている。(図表ボタンをクリックします。)



あなたは、顧客の注文テーブルから行を表示する必要があると、次の XML 形式で 1 に設定 CustomerId の値を持つ行です。

```

<CUSTOMERS Name="Customer A" Country="Australia">
  <ORDERS OrderID="1" OrderDate="2001-01-01" Amount="3400.00" />
  <ORDERS OrderID="2" OrderDate="2002-01-01" Amount="4300.00" />
</CUSTOMERS>
  
```

あなたはどちらの Transact-SQL クエリを使うべきでしょうか？

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

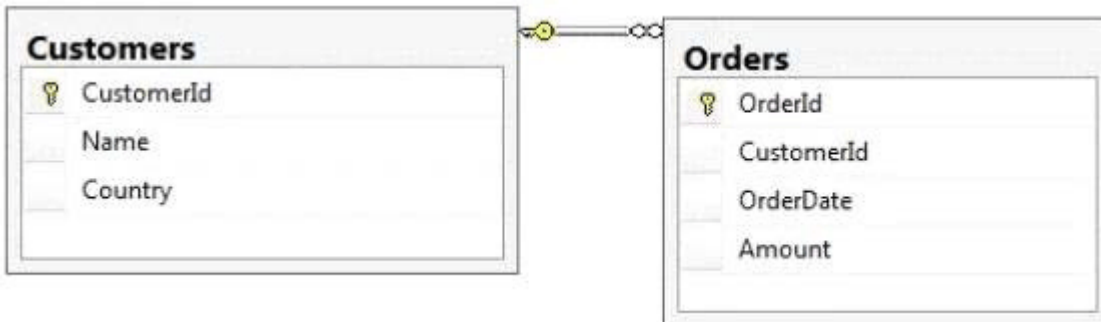
E. SELECT Name, Country, OrderId, OrderDate, Amount FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers. CustomerId WHERE Customers.CustomerId = 1 FOR XML AUTO
 F. SELECT Name, Country, OrderId, OrderDate, Amount FROM Orders INNER JOIN Customers ON Orders.CustomerId = Customers. CustomerId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

Answer: E

4. あなたが ContosoDb 付きの Microsoft SQL Server 2012 のデータベースを管理します。展示に示すように、テーブルが定義されています。(図表ボタンをクリックします。)



あなたは、顧客の注文テーブルから行を表示する必要があると、次のXML形式で1に設定CustomerIdの値を持つ行です。

```

<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>
  
```

あなたはどちらの Transact-SQL クエリを使うべきでしょうか？

- 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')

Answer: D

5. あなたは、アプリケーションをサポートする Microsoft SQL Server 2012 のサーバーのデータベースを開発します。

アプリケーションは、次の定義を持つ表が含まれている：

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

あなたは、行ごとに ItemsInStore と ItemsInWarehouse 値の総和を返す計算列を作成する必要があります。新しい列が重く照会することが期待、そしてあなたは、インデックス列にできるようにする必要があります。

あなたはどちらの Transact-SQL ステートメントを使用する必要がありますか？

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)

Answer: B

6. あなたは、Customers という名前のテーブルを含む Microsoft SQL Server 2012 のデータベースを開発する。

Customers テーブルには、次の定義が含まれている：

```

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]

```

あなたは、どちら MobileNumber または HomeNumber 列が更新されている場合にのみ監査レコードを作成する必要があります。

あなたはどちらの Transact-SQL クエリを使うべきでしょうか？

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

Answer: D

7. あなたが SavingAccounts と LoanAccounts という名前の二つのテーブルを持つ Microsoft SQL Server 2012 のデータベースを開発しています。両方のテーブルのデータ型は nvarchar データ型の AccountNumber をという名前の列があります。あなたは、トランザクションの名前の 3 番目のテーブルを使用しているの transactionID、AccountNumber を、金額、および TransactionDate という名前の列があります。あなたは、複数のレコードがトランザクションテーブルに挿入されたとき、または SavingAccounts LoanAccounts で有効 AccountNumber をを持っている唯一のレコードが挿入されていることを確認する必要があります。

あなたはどちらの Transact-SQL ステートメントを使用する必要がありますか？

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
```

Answer: A

8. あなたは、Microsoft SQL Server 2012 のデータベースを開発する。

あなたは、次のタスクを実行するビューを作成する：

- 50 万レコードの各まで含む 8 のテーブルを結合する。

- 5 フィールドの集計を行います。

ビューは頻繁にいくつかのレポートで使用されます。 、レポートの性能を向上させる必要があります。

あなたはどうすればいいのでしょうか？

A. テーブル値関数にビューに変換します。

B. 共通テーブル式 (CTE) にビューに変換します。

C. インデックス付きビューにビューに変換します。

D. ストアドプロシージャにビューを変換し、ストアドプロシージャからの一時テーブルに結果を取得します。

Answer: C

9. あなたは、Microsoft SQL Server 2012 のデータベースのデータベース開発者である。

データベースには、次の定義を持つ Customers という名前のテーブルが含まれている：

```
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))
```

あなたは、受注テーブルの CustomerID 列は、Customer テーブルの CustomerID 列に存在する値のみが含まれていることを確認する必要があります。

あなたはどちらの Transact-SQL ステートメントを使用する必要がありますか？

A. ALTER TABLE Orders ADD CONSTRAINT FK_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_Orders_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 (OrderID) REFERENCES Orders (OrderID);

E. ALTER TABLE Orders ADD CONSTRAINT PK_Orders_CustomerID PRIMARY KEY (CustomerID)

Answer: A

10. あなたは、ベンダー、顧客、およびエージェントのデータを含む三つのテーブルを持っている。あなたは、これらの企業のための電話番号をルックアップするために使用されるビューを作成します。ビューには、次の定義があります：あなたは、ユーザーは、このビューを使用することによってのみ、電話番号を更新することができることを確認する必要があります。

あなたはどうすればいいのでしょうか？

```
Create view apt.vwCompanyPhoneList
(Source, CompanyID, CompanyNumber,
 LastName, FirstName, BusinessName, Phone)
as

SELECT 'Customer' as Source
  , CustomerID
  , CustomerNumber
  , CustomerLastName
  , CustomerFirstName
  , CustomerBusinessName
  , Phone
FROM apt.Customer
UNION ALL
SELECT 'Agent' as Source
  , AgentID
  , AgentNumber
  , AgentLastName
  , AgentFirstName
  , AgentBusinessName
  , Phone
FROM apt.Agent
UNION ALL
SELECT 'Vendor' as Source
  , VendorID
  , VendorNumber
  , VendorLastName
  , VendorFirstName
  , VendorBusinessName
  , Phone
FROM apt.Vendor
GO
```

- A. ビューを変更します。各 SELECT 文とともに VIEWS クエリヒントを EXPAND 使用します。
- B. ビューに INSTEAD OF UPDATE トリガを作成します。
- C. ビューを削除します。 SCHEMABINDING 句を使用してビューを再作成し、ビューにインデックスを作成します。
- D. ビューの AFTER UPDATE トリガーを作成します。

Answer: B

11. あなたは、Microsoft SQL Server 2012 のデータベースを開発する。あなたは、注文からビューを作成し、次の定義を使用してテーブルと OrderDetails です。

```

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

```

ユーザーがビューを使用してデータを変更することができるようにする必要があります。
あなたはどうすればいいのでしょうか？

- A. ビュー上の AFTER トリガーを作成します。
- B. VIEW_METADATA 句を使用するビューを変更します。
- C. ビューに INSTEAD OF トリガーを作成します。
- D. インデックス付きビューにビューを変更します。

Answer: C

12. 次のコードを使用して作成されたビューがあり：

```

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

```

あなたは、次の要件を満たす必要があり Sales.fn_OrdersByTerritory 名前インラインテーブル値関数を作成する必要があります：

- @T の整数パラメーターを受け入れます。
- 参照列に一つの分名を使用します。
- SalesTerritoryID によってクエリ結果をフィルタリングします。
- OrdersByTerritoryView で使用順序と同じ順序で列を返します。

あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）

```

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

```

Answer: A

13. あなたは、展示に示すテーブルを含むデータベースを持っている。(図表ボタンをクリックします。)

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"/>

あなたは、SQL Server2012 がインストールされている新しいサーバーを展開。あなたは、Sales という名前のテーブルを作成する必要があります。

新しいサーバーに OrderDetails テーブル。Sales.OrderDetails は、次の要件を満たしている必要があります：
-ディスクに結果を書き込みます。

- ListPrice と各行の数量の製品を格納 LineltemTotal という名前の新しい列が含まれています。

-コードは任意のオブジェクトのデリミタを使用しないでください。

解決策は、LineltemTotal がテーブルの最後の列として格納されていることを確認する必要があります。あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）


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))

Answer: A

14. あなたは、展示に示すテーブルを含むデータベースを持っている。(図表ボタンをクリックします。)

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"/>

あなたは、次の要件を満たすために `uv_CustomerFullName` という名前のビューを作成する必要があります :

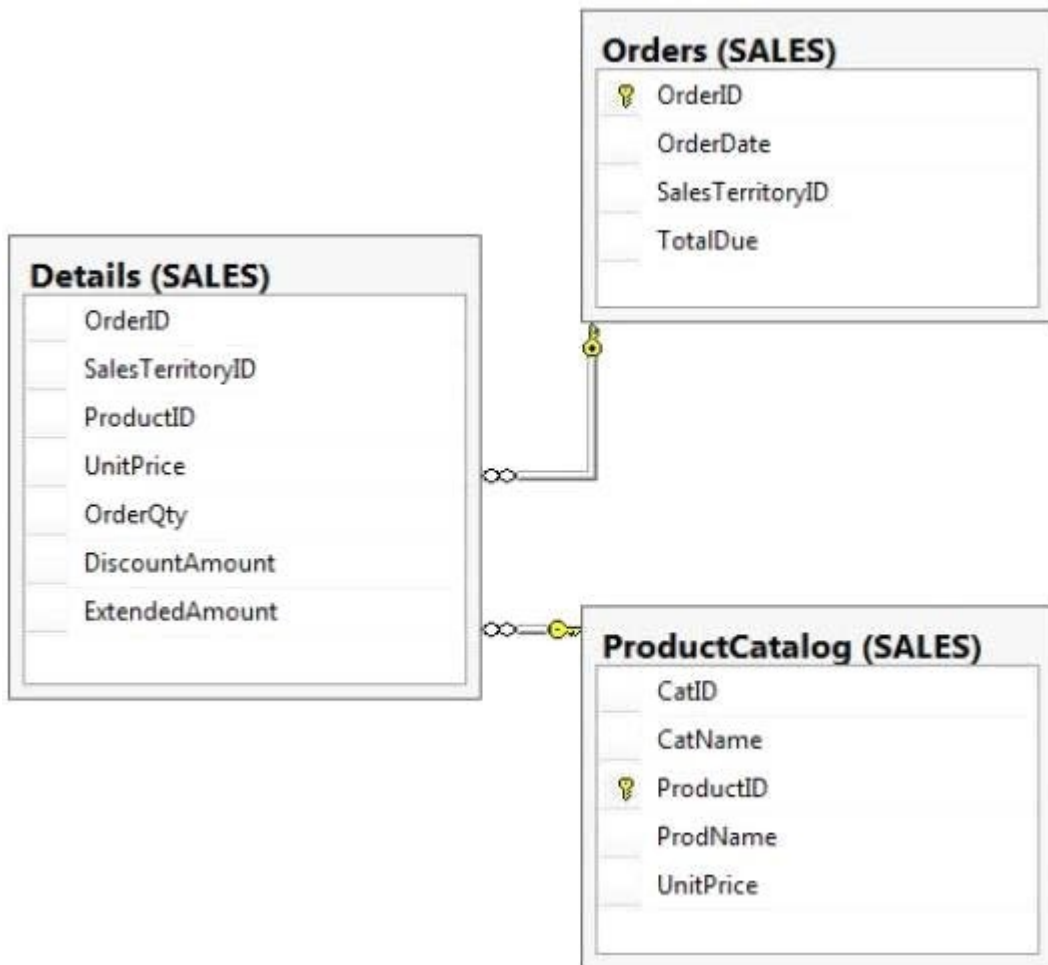
- コードは、オブジェクトの区切り文字を含めることはできません。
- ビューは、**Sales** スキーマ内に作成する必要があります。
- 列は一つの分名を使用して参照される必要があります。
- ビューには、最初の名前とすべての顧客の姓を返す必要があります。
- ビューが変更される顧客表の基礎構造を防ぐ必要があります。
- ビューには、関係なく、ユーザーのデフォルト・スキーマの、すべての参照オブジェクトを解決できる必要があります。

あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）

- 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`

Answer: A

15. あなたは、展示に示すテーブルを含むデータベースを持っている。(図表ボタンをクリックします。)



あなたが **Sales.Details** テーブルから各受注コードの総売上高を計算するクエリを作成する必要があります。

解決策は、次の要件を満たしている必要があります：

- 参照列に一つの分名を使用します。
- 受注コードからの結果の順序を並べ替えます。
- ユーザーの既定のスキーマに依存しないものとします。
- 計算 **ExtendedAmount** ため **TotalSales** のエイリアスを使用します。
- のみ受注コード列と計算 **TotalSales** 列を表示します。

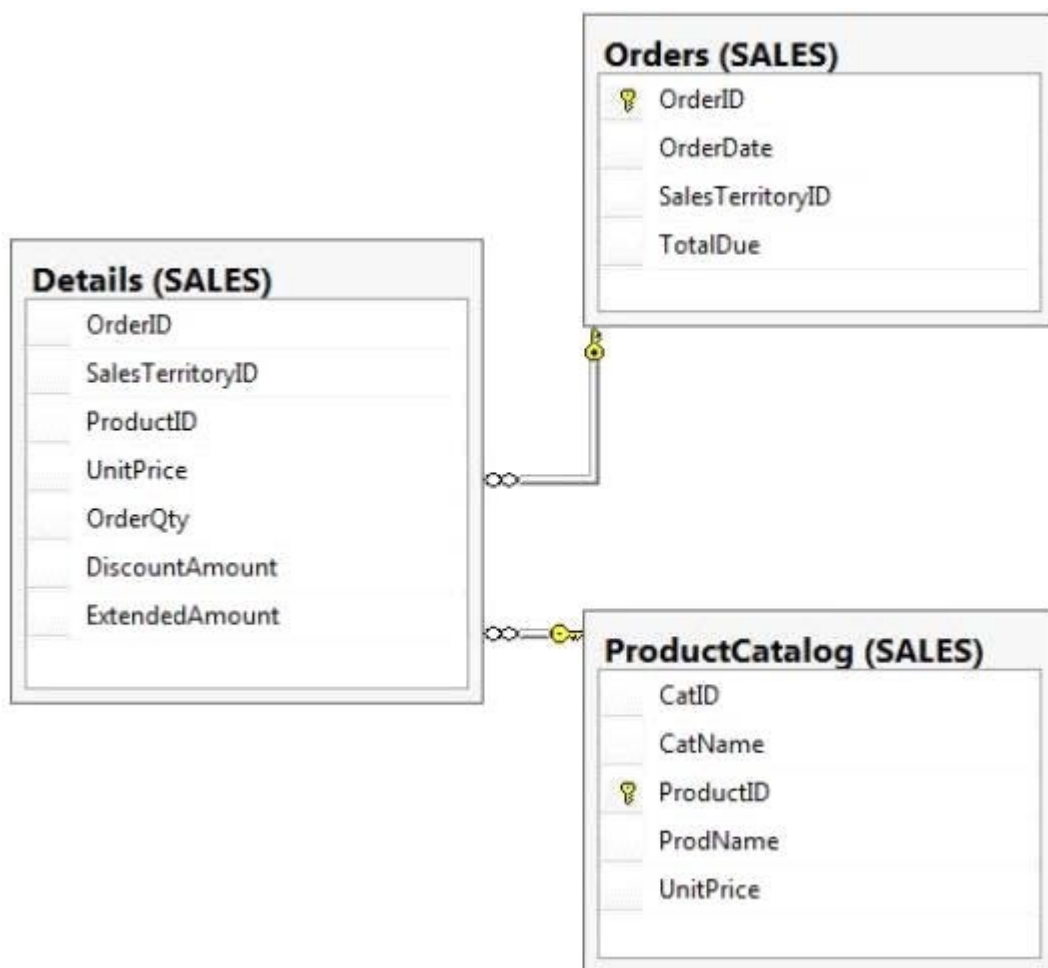
あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）

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`

Answer: A

16. あなたは、展示に示されているようにテーブルを含むデータベースを持っている。（図表ボタンをクリックします。）



次のクエリを持っている:

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

あなたは、次の要件を満たすために、クエリを再作成する必要があります:

- 一液名のみを使用して参照列です。

- 並べ替え SalesTerritoryID によって集計、およびの ProductID によってです。

- SalesTerritoryID からの ProductID に降順で結果をご注文ください。

- ソリューションは、既存の SELECT 句と FROM 句を使用する必要があります。

あなたはどちらのコードセグメントを使うべきでしょうか? に答えるために、答えエリアで正しいコードを入力します。


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

Answer: A

17. あなたは、展示に示すテーブルを含むデータベースを持っている。(図表ボタンをクリックします)。

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"/>

あなたは、レポートのクエリを作成する必要があります。クエリは、次の要件を満たしている必要があります：

- オブジェクト区切り文字を使用しないでください。
 - 最初に最も最近の注文を返します。
 - 別名として、表の最初の頭文字を使用してください。
 - 各顧客の最新の注文日を返します。
 - 注文した人の最後の名前を取得します。
 - レポートの最後の列として表示されます **MostRecentOrderDate** という名前の列で注文日を返します。
- 解決策は、ANSI SQL-99 標準をサポートする必要があります。

あなたはどちらのコードセグメントを使うべきでしょうか？ (答えるために、答えエリアで正しいコードを入力します。)

A. `SELECT c.CustomerID -- optional c.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

B. 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

C. 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

Answer: A

18. あなたが Sales.InvoiceSchema という名前の XML スキーマコレクションを持っている。あなたが XML1 名前の XML 型の変数を宣言する必要があります。解決策は、XML1 が Sales.InvoiceSchema 使用して検証されることを確認する必要があります。

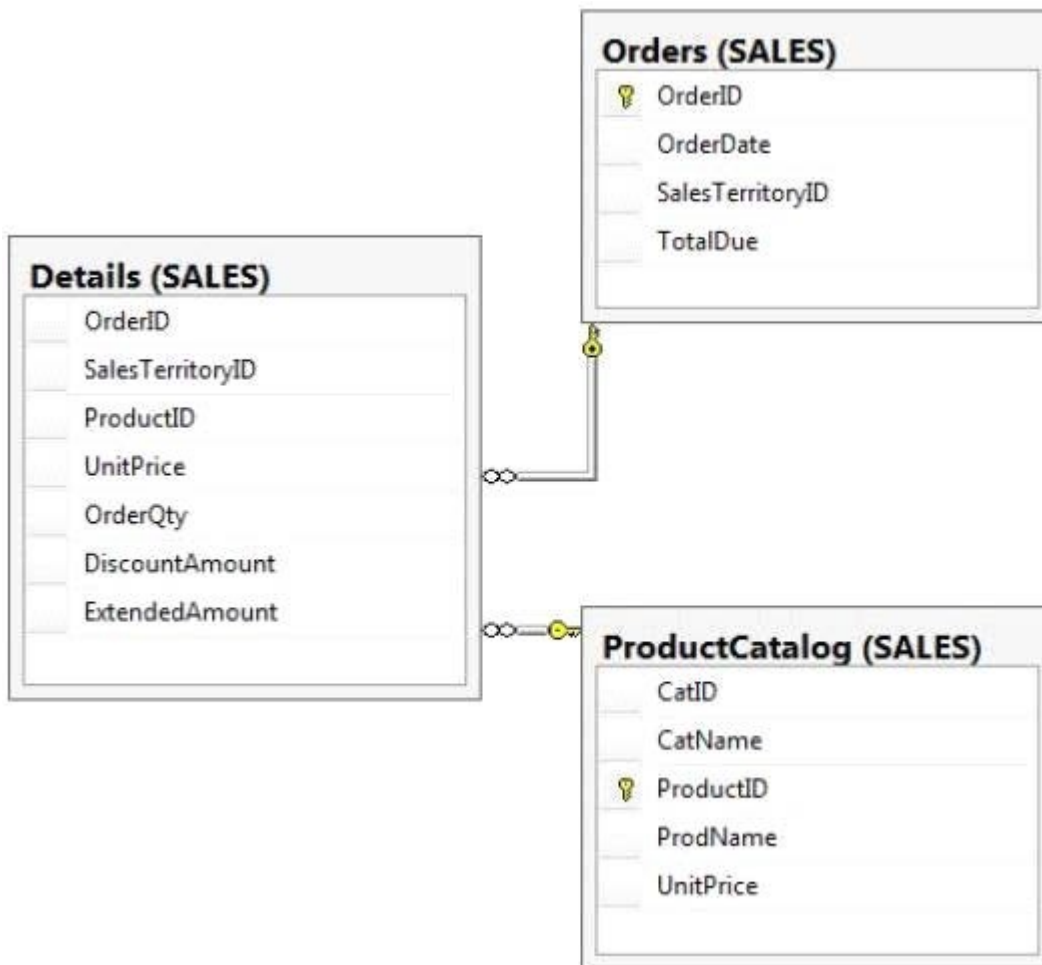
あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）

A. Declare @XML1=XML(Sales.InvoiceSchema)

B. DECLARE @XML1 XML @XML1 = Sales.InvoiceSchema CREATE XML SCHEMA COLLECTION XML1 AS @XML1

Answer: A

19. あなたは、展示に示されているようにテーブルを含むデータベースを持っている。（図表ボタンをクリックします。）



あなたが **Sales.ProductCatalog** から製品のリストを返すクエリを作成する必要があります。

解決策は、次の要件を満たしている必要があります：

- 単価は降順で戻らなければなりません。
- クエリは、テーブルを参照する二つの部分の名前を使用する必要があります。
- クエリは、結果を計算するために **RANK** 関数を使用する必要があります。
- クエリは **PriceRank** という名前の列の行の順位を返す必要があります。
- リストには、表に定義されている順序で列を表示する必要があります。
- **PriceRank** は、最後に記述する必要があります。

あなたはどちらのコードセグメントを使うべきでしょうか？（答えるために、答えエリアで正しいコードを入力します。）

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`

Answer: A

20. あなたは、展示に示すテーブルを含むデータベースを持っている。(図表ボタンをクリックします。)

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"/>

あなたは、出願という名前のアプリケーションを持っている。あなたは、int データ型を使用しています @カウントという名前のパラメータを持っている。

APP1 は、ストアプロシージャに @カウントを通過させるように構成されている。

あなたは、出願のため usp_Customers という名前のストアプロシージャを作成する必要があります。

Usp_Customers は、次の要件を満たしている必要があります：

- オブジェクト区切り文字を使用しないでください。
- ソート、カウントを最小限に抑えます。
- 唯一のアルファベット順に各顧客の姓を返します。
- @ Count パラメータで指定された行の数だけを返します。
- 解決策は、文を BEGIN と END 使用してはいけません。

あなたはどちらのコードセグメントを使うべきでしょうか？ (答えるために、答えエリアで正しいコードを入力します。)

A. CREATE PROCEDURE usp_Customers @Count int AS SELECT TOP(@Count) LastName FROM

Customers ORDER BY LastName

Answer: A