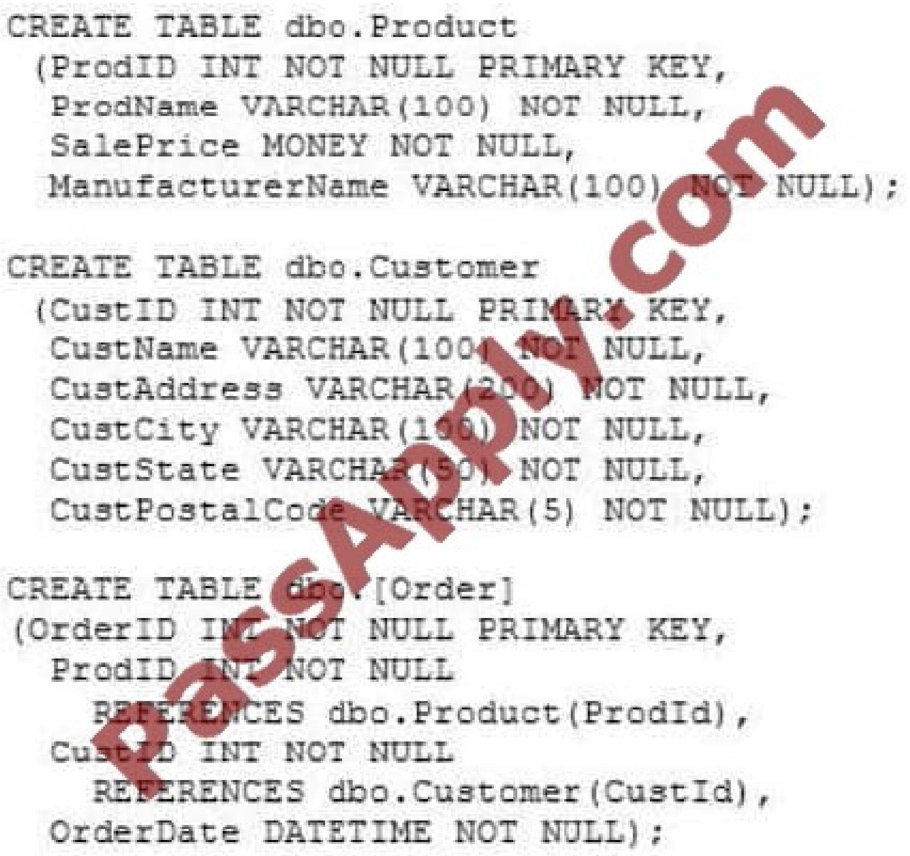
70-433$^{\text{Q\&As}}$

TS: Microsoft SQL Server 2008, Database Development

Pass Microsoft 70-433 Exam with 100% Guarantee

QUESTION 1

You are the database developer for an order-processing application. The database has the following tables:



```
CREATE TABLE dbo.Product
  (ProdID INT NOT NULL PRIMARY KEY,
   ProdName VARCHAR(100) NOT NULL,
   SalePrice MONEY NOT NULL,
   ManufacturerName VARCHAR(100) NOT NULL);

CREATE TABLE dbo.Customer
  (CustID INT NOT NULL PRIMARY KEY,
   CustName VARCHAR(100) NOT NULL,
   CustAddress VARCHAR(200) NOT NULL,
   CustCity VARCHAR(100) NOT NULL,
   CustState VARCHAR(50) NOT NULL,
   CustPostalCode VARCHAR(5) NOT NULL);

CREATE TABLE dbo.[Order]
 (OrderID INT NOT NULL PRIMARY KEY,
   ProdID INT NOT NULL
     REFERENCES dbo.Product(ProdId),
   CustID INT NOT NULL
     REFERENCES dbo.Customer(CustId),
   OrderDate DATETIME NOT NULL);
```

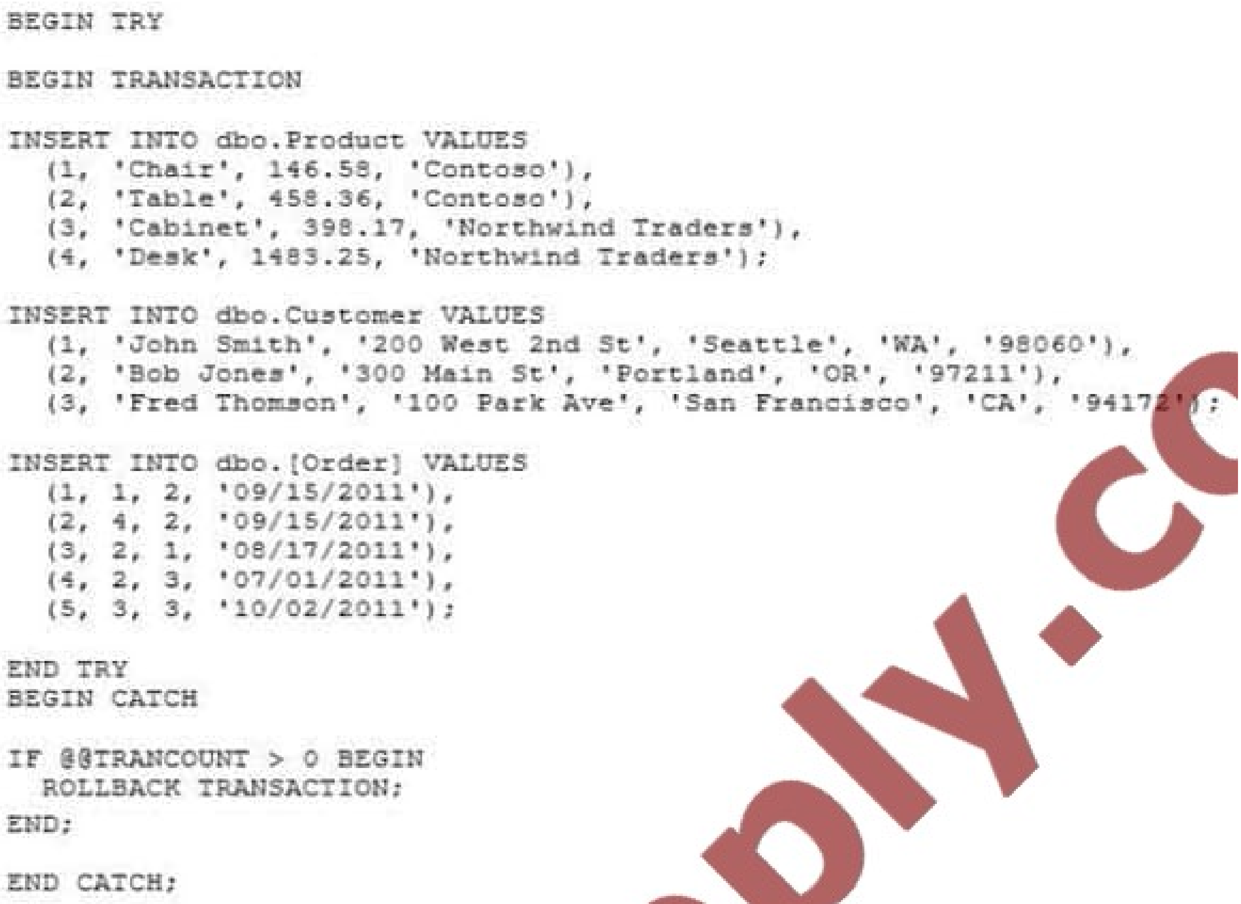
You need to ensure that the following requirements are met:

Data is loaded into the tables.

Data that has been inserted will be removed if any statement fails. No open transactions are performed after the batch has executed.

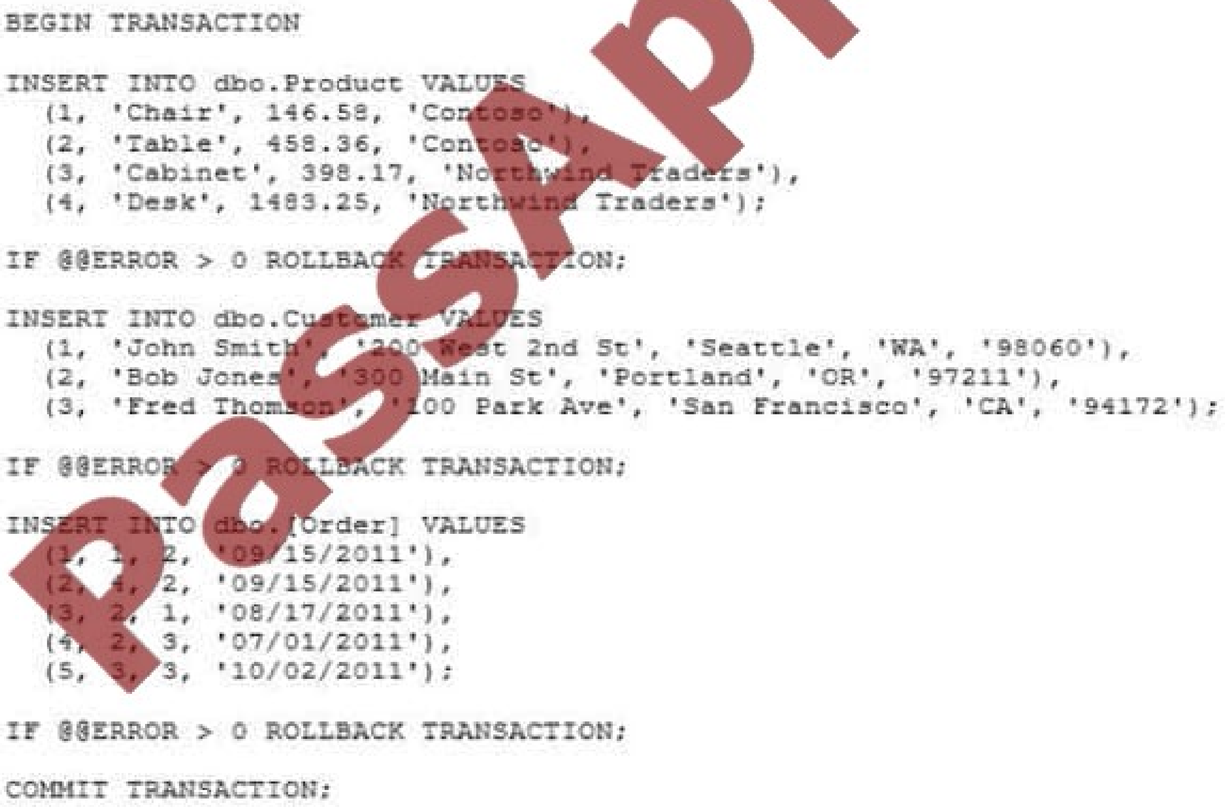
Which Transact-SQL statements should you use?

A.



```
BEGIN TRY

BEGIN TRANSACTION

INSERT INTO dbo.Product VALUES
   (1, 'Chair', 146.58, 'Contoso'),
   (2, 'Table', 458.36, 'Contoso'),
   (3, 'Cabinet', 398.17, 'Northwind Traders'),
   (4, 'Desk', 1483.25, 'Northwind Traders');

INSERT INTO dbo.Customer VALUES
   (1, 'John Smith', '200 West 2nd St', 'Seattle', 'WA', '98060'),
   (2, 'Bob Jones', '300 Main St', 'Portland', 'OR', '97211'),
   (3, 'Fred Thomson', '100 Park Ave', 'San Francisco', 'CA', '94172');

INSERT INTO dbo.[Order] VALUES
   (1, 1, 2, '09/15/2011'),
   (2, 4, 2, '09/15/2011'),
   (3, 2, 1, '08/17/2011'),
   (4, 2, 3, '07/01/2011'),
   (5, 3, 3, '10/02/2011');

END TRY
BEGIN CATCH

IF @@TRANCOUNT > 0 BEGIN
  ROLLBACK TRANSACTION;
END;

END CATCH;
```

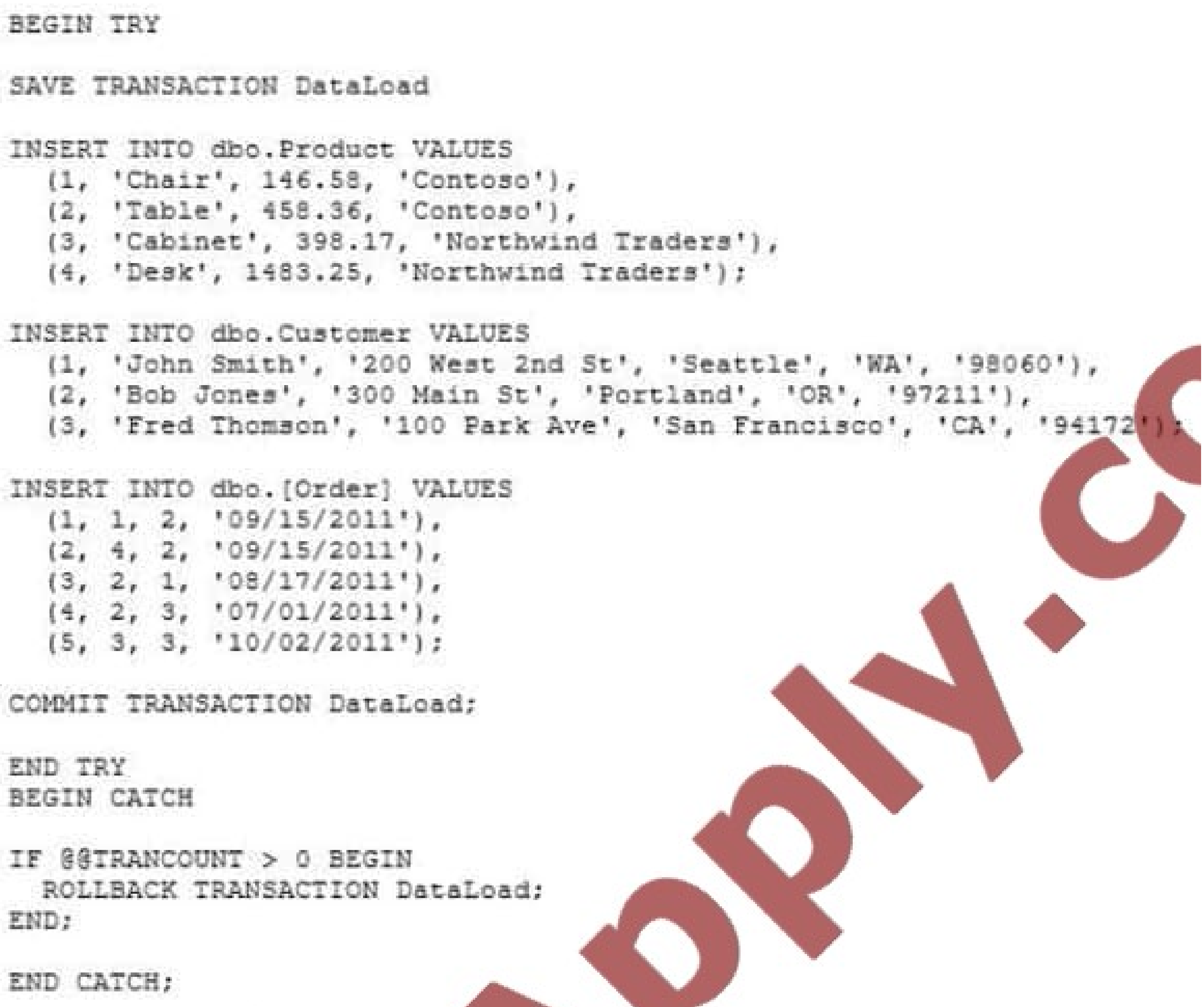
B.



```
BEGIN TRANSACTION

INSERT INTO dbo.Product VALUES
   (1, 'Chair', 146.58, 'Contoso'),
   (2, 'Table', 458.36, 'Contoso'),
   (3, 'Cabinet', 398.17, 'Northwind Traders'),
   (4, 'Desk', 1483.25, 'Northwind Traders');

IF @@ERROR > 0 ROLLBACK TRANSACTION;

INSERT INTO dbo.Customer VALUES
   (1, 'John Smith', '200 West 2nd St', 'Seattle', 'WA', '98060'),
   (2, 'Bob Jones', '300 Main St', 'Portland', 'OR', '97211'),
   (3, 'Fred Thomson', '100 Park Ave', 'San Francisco', 'CA', '94172');

IF @@ERROR > 0 ROLLBACK TRANSACTION;

INSERT INTO dbo.[Order] VALUES
   (1, 1, 2, '09/15/2011'),
   (2, 4, 2, '09/15/2011'),
   (3, 2, 1, '08/17/2011'),
   (4, 2, 3, '07/01/2011'),
   (5, 3, 3, '10/02/2011');

IF @@ERROR > 0 ROLLBACK TRANSACTION;

COMMIT TRANSACTION;
```

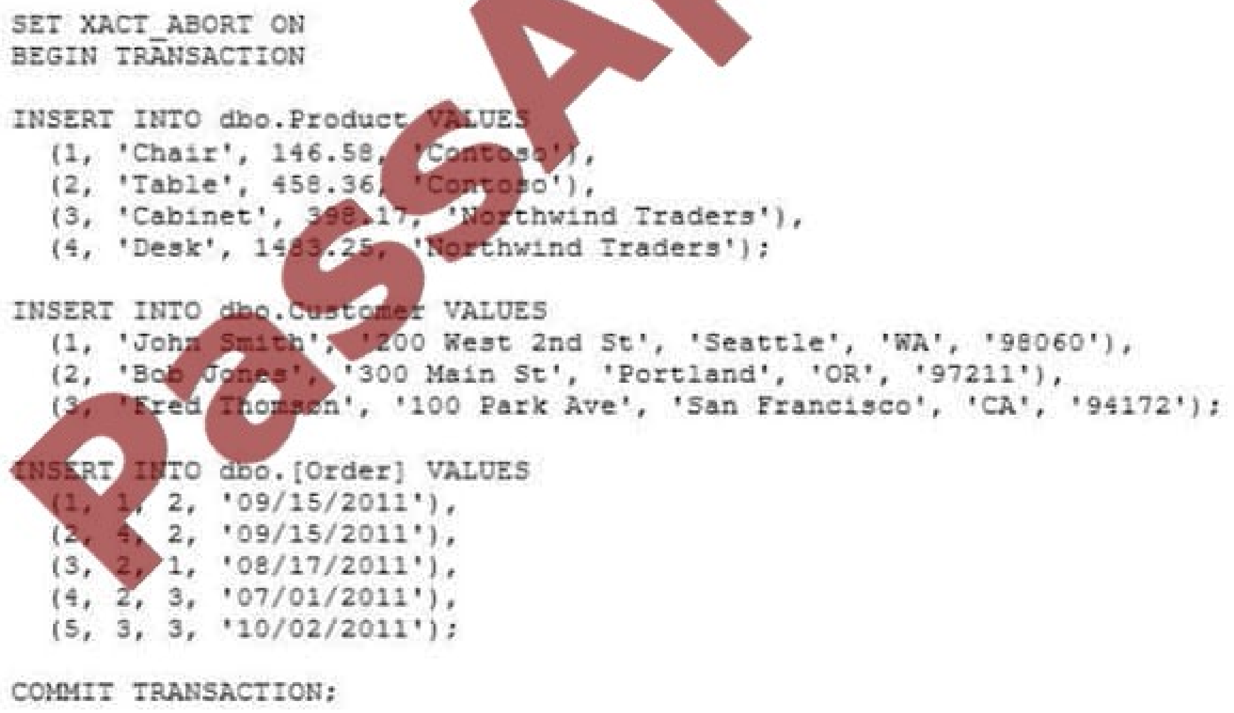
A. B.

C.



```
BEGIN TRY

SAVE TRANSACTION DataLoad

INSERT INTO dbo.Product VALUES
   (1, 'Chair', 146.58, 'Contoso'),
   (2, 'Table', 458.36, 'Contoso'),
   (3, 'Cabinet', 398.17, 'Northwind Traders'),
   (4, 'Desk', 1483.25, 'Northwind Traders');

INSERT INTO dbo.Customer VALUES
   (1, 'John Smith', '200 West 2nd St', 'Seattle', 'WA', '98060'),
   (2, 'Bob Jones', '300 Main St', 'Portland', 'OR', '97211'),
   (3, 'Fred Thomson', '100 Park Ave', 'San Francisco', 'CA', '94172');

INSERT INTO dbo.[Order] VALUES
   (1, 1, 2, '09/15/2011'),
   (2, 4, 2, '09/15/2011'),
   (3, 2, 1, '08/17/2011'),
   (4, 2, 3, '07/01/2011'),
   (5, 3, 3, '10/02/2011');

COMMIT TRANSACTION DataLoad;

END TRY
BEGIN CATCH

IF @@TRANCOUNT > 0 BEGIN
  ROLLBACK TRANSACTION DataLoad;
END;

END CATCH;
```

D.



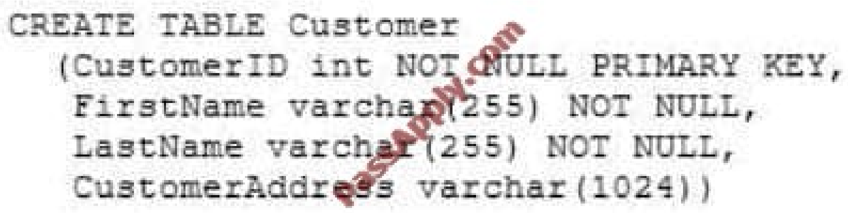
```
SET XACT_ABORT ON
BEGIN TRANSACTION

INSERT INTO dbo.Product VALUES
   (1, 'Chair', 146.58, 'Contoso'),
   (2, 'Table', 458.36, 'Contoso'),
   (3, 'Cabinet', 398.17, 'Northwind Traders'),
   (4, 'Desk', 1483.25, 'Northwind Traders');

INSERT INTO dbo.Customer VALUES
   (1, 'John Smith', '200 West 2nd St', 'Seattle', 'WA', '98060'),
   (2, 'Bob Jones', '300 Main St', 'Portland', 'OR', '97211'),
   (3, 'Fred Thomson', '100 Park Ave', 'San Francisco', 'CA', '94172');

INSERT INTO dbo.[Order] VALUES
   (1, 1, 2, '09/15/2011'),
   (2, 4, 2, '09/15/2011'),
   (3, 2, 1, '08/17/2011'),
   (4, 2, 3, '07/01/2011'),
   (5, 3, 3, '10/02/2011');

COMMIT TRANSACTION;
```

C. D.

Correct Answer: B

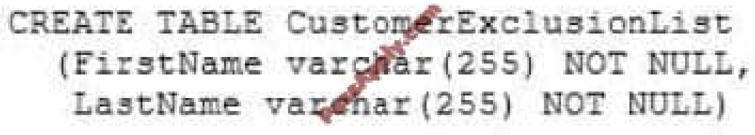
QUESTION 2

You administer a Microsoft SQL Server 2008 R2 database that has a table named Customer. The table has the following definition:



```
CREATE TABLE Customer
  (CustomerID int NOT NULL PRIMARY KEY,
   FirstName varchar(255) NOT NULL,
   LastName varchar(255) NOT NULL,
   CustomerAddress varchar(1024))
```

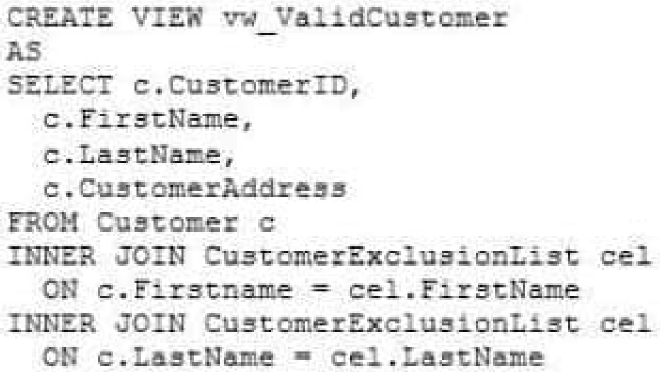
The database also has a table named CustomerExclusionList. Data will be added to the CustomerExclusionList table regularly. The CustomerExclusionList table has the following definition:



```
CREATE TABLE CustomerExclusionList
  (FirstName varchar(255) NOT NULL,
   LastName varchar(255) NOT NULL)
```

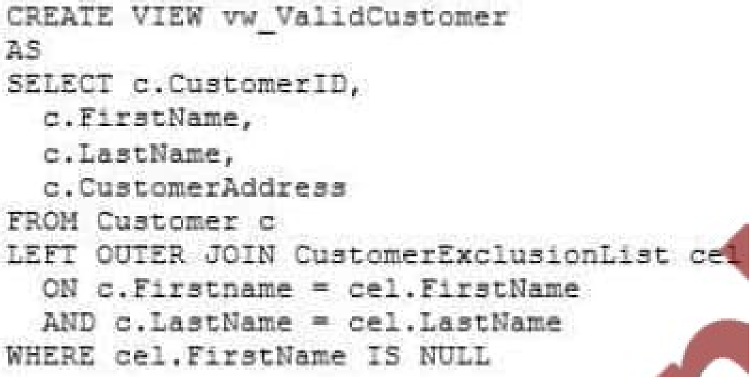
You need to create a view that returns all records and columns of the Customer table that are not present in the CustomerExclusionList table. Which Transact-SQL statement should you use?

A.



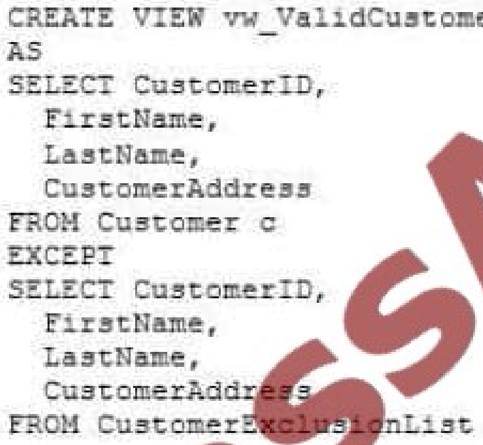
```
CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
  c.FirstName,
  c.LastName,
  c.CustomerAddress
FROM Customer c
INNER JOIN CustomerExclusionList cel
  ON c.Firstname = cel.FirstName
INNER JOIN CustomerExclusionList cel
  ON c.LastName = cel.LastName
```

B.



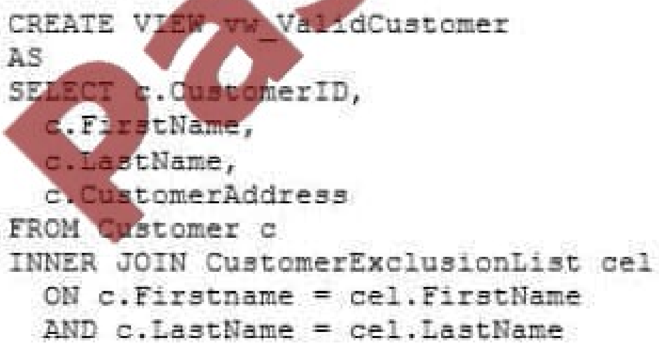
```
CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
  c.FirstName,
  c.LastName,
  c.CustomerAddress
FROM Customer c
LEFT OUTER JOIN CustomerExclusionList cel
  ON c.Firstname = cel.FirstName
  AND c.LastName = cel.LastName
WHERE cel.FirstName IS NULL
```

C.



```
CREATE VIEW vw_ValidCustomer
AS
SELECT CustomerID,
  FirstName,
  LastName,
  CustomerAddress
FROM Customer c
EXCEPT
SELECT CustomerID,
  FirstName,
  LastName,
  CustomerAddress
FROM CustomerExclusionList
```

D.



```
CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
  c.FirstName,
  c.LastName,
  c.CustomerAddress
FROM Customer c
INNER JOIN CustomerExclusionList cel
  ON c.Firstname = cel.FirstName
  AND c.LastName = cel.LastName
```

A. B. C. D.

Correct Answer: C

QUESTION 3

You attempt to query sys.dm_db_index_usage_stats to check the status on the indexes in the Contoso database. The query fails and you receive the following error:

"The user does not have permission to perform this action."

You need to have the least amount of permissions granted to access the dynamic management views.

Which permissions should be granted?

A. SELECT

B. CONTROL

C. VIEW SERVER STATE

D. CREATE EXTERNAL ACCESS ASSEMBLY

Correct Answer: C

QUESTION 4

A table named Contacts includes a column named SmtpAddress.

You must develop a report that returns e-mail addresses from the Contacts table that have the following format: at least one character, the at sign (@), at least one character, and then ".org". You need to return data that meets the requirements.

Which Transact-SQL statement should you use?

A. Select * from Contacts where SmtpAddress like \\'%@%[.]org\\'

B. Select * from Contacts where SmtpAddress like \\'_%@_%.org\\'

C. Select * from Contacts where SmtpAddress like \\'_%@_.org\\'

D. Select * from Contacts where SmtpAddress like \\'%@%.org\\'

Correct Answer: B

QUESTION 5

You have two tables named SalesPerson and SalesTerritory.

You need to create sample data by using a Cartesian product that contains the data from the SalesPerson and SalesTerritory tables.

Which code segment should you use?

A. SELECT p.SalesPersonId,

B. Name AS [Territory] FROM Sales.SalesPerson p FULL JOIN Sales.SalesTerritory t ON p.TerritoryId = t.TerritoryId

C. SELECT p.SalesPersonId,

D. Name AS [Territory] FROM Sales.SalesPerson p INNER JOIN Sales.SalesTerritory t ON p.TerritoryId = t.TerritoryId

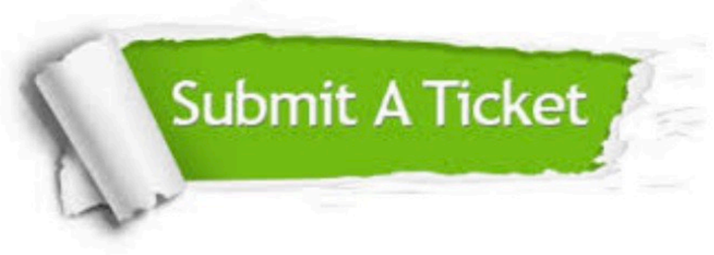
E. SELECT p.SalesPersonId,

F. Name AS [Territory] FROM Sales.SalesPerson p CROSS JOIN Sales.SalesTerritory t WHERE p.TerritoryId = t.TerritoryId

G. SELECT p.SalesPersonId,

H. Name AS [Territory] FROM Sales.SalesPerson p CROSS JOIN Sales.SalesTerritory t;

Correct Answer: D


Submit A Ticket