



# Microsoft

## EXAM 70-433

*TS: Microsoft SQL Server 2008, Database Development*

Total Q&A: 145

Exam Version: 7.60

<http://www.exam1pass.com/70-433-exam.html>

**Question: 1**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE

```

[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[Active] [bit] NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
[ProductID] ASC
) ON [PRIMARY]) ON [PRIMARY]GO

```

You want to add a new field to the Product table to meet the following requirements:

- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support Unicode string values that have a minimum length of two characters and a maximum length of 100 characters.

Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT
- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME

R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)

**Answer: J**

### Question: 2

You are the database developer for a Microsoft SQL Server 2008 database that contains tables named order and product.

The tables have the following definitions: CREATE TABLE [order]

(OrderID INT,

ProductID INT,

CustomerID INT,

OrderDate DATETIME); CREATE TABLE product

(ProductID INT,

ProductName VARCHAR(100),

SalePrice money,

ManufacturerName VARCHAR(100)); You need to write a query that will extract a valid XML result

set of all ordered products. You also need to ensure that the query conforms to the following

schema: <?xml version="1.0" encoding="utf-16"?><xsd:schema

attributeFormDefault="unqualified" elementFormDefault="qualified" version="1.0"

xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<xsd:element name="OrderedProducts">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="ProductID" type="xsd:int" />

<xsd:element name="ProductName" type="xsd:string" />

<xsd:element name="SalePrice" type="xsd:decimal" />

<xsd:element name="ManufacturerName" type="xsd:string" />

<xsd:element name="OrderDate" type="xsd:dateTime" />

</xsd:sequence>

</xsd:complexType>

</xsd:element></xsd:schema> Which SQL query should you use?

A.

SELECT p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate FROM product  
p INNER JOIN [order] o ON p.ProductID = o.ProductID FOR XML AUTO ('OrderedProducts');

B.

SELECT p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate FROM product  
p INNER JOIN [order] o ON p.ProductID = o.ProductID FOR XML AUTO;

C.

SELECT p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate FROM product  
p INNER JOIN [order] o ON p.ProductID = o.ProductID FOR XML PATH ('OrderedProducts');

D.

SELECT '<OrderedProducts>', p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate, '</OrderedProducts>' FROM product p INNER JOIN [order] o ON p.ProductID =  
o.ProductID FOR XML PATH;

E. SELECT 1 as Tag, 0 as Parent, p.ProductID as  
 [OrderedProducts!1!ProductID!ELEMENT], p.ProductName as  
 [OrderedProducts!1!ProductName!ELEMENT], p.SalePrice as  
 [OrderedProducts!1!SalePrice!ELEMENT], p.ManufacturerName as  
 [OrderedProducts!1!ManufacturerName!ELEMENT], o.OrderDate as  
 [OrderedProducts!1!OrderDate!ELEMENT] FROM product p INNER JOIN [order] o ON p.ProductID =  
 o.ProductID FOR XML EXPLICIT;

F. SELECT 1 as Tag, 0 as Parent, p.ProductID as  
 [OrderedProducts!1!ProductID!ELEMENT], p.ProductName as  
 [OrderedProducts!1!ProductName!ELEMENT], p.SalePrice as  
 [OrderedProducts!1!SalePrice!ELEMENT], p.ManufacturerName as  
 [OrderedProducts!1!ManufacturerName!ELEMENT], o.OrderDate as  
 [OrderedProducts!1!OrderDate!ELEMENT] FROM product p INNER JOIN [order] o ON p.ProductID =  
 o.ProductID FOR XML EXPLICIT ('OrderedProducts');

G.  
 SELECT p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate FROM product  
 p INNER JOIN [order] o ON p.ProductID = o.ProductID FOR XML RAW;

H.  
 SELECT p.ProductID, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate FROM product  
 p INNER JOIN [order] o ON p.ProductID = o.ProductID FOR XML RAW ('OrderedProducts');

**Answer: C OR E**

### Question: 3

You are a database developer writing reports for a sales management application. A customer table has the following definition: CREATE TABLE customer

(CustomerID INT,  
 FirstName VARCHAR(30),  
 LastName VARCHAR(50),  
 StreetAddress VARCHAR(100),  
 City VARCHAR(100),  
 [State] VARCHAR(25),  
 PostalCode VARCHAR(5));

An order table has the following definition: CREATE TABLE [order]

(OrderID INT,  
 ProductID INT,  
 CustomerID INT,  
 OrderDate DATETIME);

You need to write a report that contains the following columns:

Column name	Description
CustomerID	The CustomerID
FullName	Concatenated first and last names
Postalcode	Customer's Postal Code
OrderCount	Number of orders for this customer
EarliestOrderDate	The OrderDate from the earliest (by OrderDate) order for this customer

You also need to ensure that the report meets the following requirements:

- Contains only customers who have placed orders
- Contains only customers who have Postal Codes beginning with 89
- Returns only one record for each customer
- The report is ordered by the CustomerID in ascending order Which Transact-SQL query should you use?

A. SELECT c.CustomerID,c.FirstName + ' ' + c.LastName AS FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHEREc.PostalCode LIKE '89%'GROUP BY c.CustomerId, c.FirstName + ' ' + c.LastName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

B. SELECT c.CustomerID,c.FirstName + ' ' + c.LastName AS FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHERE c.PostalCode = '89%'GROUP BY c.CustomerId, c.FirstName + ' ' + c.LastName, c.PostalCode, o.CustomerIdORDER BY FullName

C. WITH FullNames AS(SELECT CustomerId,FirstName + ' ' + LastName AS FullNameFROM customer)SELECT c.CustomerID,fn.FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN FullNames fn ON fn.CustomerID = c.CustomerIDINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHERE c.PostalCode LIKE '89%'GROUP BY c.CustomerId, fn.FullName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

D. SELECT c.CustomerID,c.FirstName + ' ' + c.LastName AS FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cLEFT OUTER JOIN [order] o ON c.CustomerId = o.CustomerIdWHERE c.PostalCode LIKE '89%'GROUP BY c.CustomerId, c.FirstName + ' ' + c.LastName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

E. USING FullNames AS( SELECTCustomerId,FirstName + ' ' + LastName AS FullNameFROM customer)SELECT c.CustomerID,fn.FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN FullNames fn ON fn.CustomerID = c.CustomerIDINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHERE c.PostalCode LIKE '89%'GROUP BY c.CustomerId, fn.FullName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

F. SELECT c.CustomerID,c.FirstName & ' ' & c.LastName AS FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHERE c.PostalCode LIKE '89%'GROUP BY c.CustomerId, c.FirstName & ' ' & c.LastName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

G. SELECTc.CustomerID,c.FirstName + ' ' + c.LastName AS FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN [order] o ON c.CustomerId = o.CustomerIdWHEREc.PostalCode LIKE '89%'ORDER BY c.CustomerID

H. WITH FullNames AS( SELECTCustomerId,FirstName + ' ' + LastName AS FullNameFROM customer)SELECTc.CustomerID,fn.FullName,c.PostalCode,COUNT(\*) AS OrderCount,MIN(o.OrderDate) AS EarliestOrderDateFROM Customer cINNER JOIN FullNames fn ON fn.CustomerID = c.CustomerIDINNER JOIN [order] o ON c.CustomerId = o.CustomerIdGROUP BY c.CustomerId, fn.FullName, c.PostalCode, o.CustomerIdORDER BY c.CustomerID

**Answer: A OR C**

**Question: 4**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK\_Product\_ProductID and a non-clustered index named AK\_Product\_ProductNumber. Both indexes have been created on a single primary partition.

The table has the following definition: `CREATE TABLE [Production].[Product](  
[ProductID] [int] IDENTITY(1,1) NOT NULL,  
[Name] [nvarchar](50) NOT NULL,  
[ProductNumber] [nvarchar](25) NOT NULL,  
[Color] [nvarchar](15) NULL,  
[Class] [nchar](2) NULL,  
[Style] [nchar](2) NULL,  
[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED  
(  
[ProductID] ASC`

`) ON [PRIMARY]) ON [PRIMARY]GO` The index has the following definition: `CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_ProductNumber] ON [Production].[Product] (  
[ProductNumber] ASC) ON [PRIMARY]GO` The Production.Product table contains 1 million rows. You want to ensure that data retrieval takes the minimum amount of time when the queries executed against the Production.Product table are ordered by product number or filtered by class. You need to find out the degree of fragmentation for the indexes on the Production.Product table.

Which Transact-SQL statement should you use?

- A. `ALTER STATISTICS Production.Product`
- B. `SELECT * FROM sys.indexes where name=N'Production.Product'`
- C. `CREATE STATISTICS ProductClass_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class is not null`
- D. `SELECT * FROM sys.dm_db_index_physical_stats (DB_ID(),  
OBJECT_ID(N'Production.Product'),NULL, NULL, NULL)`
- E. `SELECT * FROM STATS WHERE name='AK_Product_ProductNumber'`
- F. `DBCC SHOW_STATISTICS ('Production.Product', AK_Product_ProductNumber)`
- G. `UPDATE INDEX AK_Product_ProductNumber ON Production.Product SET  
(STATISTICS_NORECOMPUTE = ON)`
- H. `ALTER INDEX AK_Product_ProductNumber ON Production.Product REBUILD`
- I. `SELECT * FROM sys.dm_db_index_operational_stats (DB_ID(),  
OBJECT_ID(N'Production.Product'),NULL, NULL)`
- J. `SELECT * FROM SYS.STATS WHERE name='AK_Product_ProductNumber'`
- K. `EXEC sys.sp_configure 'index create memory', 1`
- L. `CREATE STATS ProductClass_StatsON Production.Product (Name, ProductNumber, Class)WHERE  
Class is not nullWITH SAMPLE 100 PERCENT`

- M. ALTER DATABASE [AdventureWorks] SET AUTO\_UPDATE\_STATISTICS ON
- N. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD Partition = 1
- O. UPDATE STATISTICS Production.Product
- P. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REORGANIZE
- Q. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class <> null

---

**Answer: D**

---

**Question: 5**

A table named Contacts includes a column named SmtetAddress. 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 SmtetAddress like '%@%.org'
- B. select \* from Contacts where SmtetAddress like '\_%@\_%.org'
- C. select \* from Contacts where SmtetAddress like '%@\_%.org'
- D. select \* from Contacts where SmtetAddress like '%@%[.]org'

---

**Answer: B**

---

**Question: 6**

You have a column named TelephoneNumber that stores numbers as varchar(20). You need to write a query that returns the first three characters of a telephone number. Which expression should you use?

- A. CHARINDEX('[0-9][0-9][0-9]', TelephoneNumber, 3)
- B. SUBSTRING(TelephoneNumber, 3, 3)
- C. SUBSTRING(TelephoneNumber, 3, 1)
- D. LEFT(TelephoneNumber, 3)

---

**Answer: D**

---

**Question: 7**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK\_Product\_ProductID and a non-clustered index named AK\_Product\_ProductNumber. Both indexes have been created on a single primary partition. The table has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
```



[Style] [nchar](2) NULL,  
[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK\_Product\_ProductID] PRIMARY KEY CLUSTERED  
(  
[ProductID] ASC  
) ON [PRIMARY]) ON [PRIMARY]GO The index has the following definition: CREATE UNIQUE  
NONCLUSTERED INDEX [AK\_Product\_ProductNumber] ON [Production].[Product] (  
[ProductNumber] ASC) ON [PRIMARY]GO The Production.Product table contains 1 million rows.  
You want to ensure that data retrieval takes the minimum amount of time when the queries  
executed against the Production.Product table are ordered by product number or filtered by class.  
You need to build new query optimization statistics on the Production.Product table to support  
queries that filter data rows where the class field has a null value. Which Transact-SQL statement  
should you use?

A. DBCC SHOW\_STATISTICS ('Production.Product', AK\_Product\_ProductNumber)  
B. EXEC sys.sp\_configure 'index create memory', 1  
C. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REORGANIZE  
D. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD Partition = 1  
E. CREATE STATS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE  
Class is not nullWITH SAMPLE 100 PERCENT  
F. SELECT \* FROM sys.dm\_db\_index\_physical\_stats (DB\_ID(),  
OBJECT\_ID(N'Production.Product'),NULL, NULL, NULL)  
G. UPDATE INDEX AK\_Product\_ProductNumber ON Production.Product SET  
(STATISTICS\_NORECOMPUTE = ON)  
H. UPDATE STATISTICS Production.Product  
I. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber,  
Class)WHERE Class <> null  
J. SELECT \* FROM sys.dm\_db\_index\_operational\_stats (DB\_ID(),  
OBJECT\_ID(N'Production.Product'),NULL, NULL)  
K. SELECT \* FROM SYS.STATS WHERE name='AK\_Product\_ProductNumber'  
L. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD  
M. ALTER STATISTICS Production.Product  
N. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber,  
Class)WHERE Class is not null  
O. SELECT \* FROM sys.indexes where name=N'Production.Product'  
P. ALTER DATABASE [AdventureWorks] SET AUTO\_UPDATE\_STATISTICS ON  
Q. SELECT \* FROM STATS WHERE name='AK\_Product\_ProductNumber'

---

**Answer: N**

---

**Question: 8**

You are using SQL Server Profiler to gather deadlock information. You need to capture an XML description of a deadlock. Which event should you use?

- A. Showplan XML  
B. Lock:Deadlock



- C. Deadlock Graph
- D. Lock:Deadlock Chain

---

**Answer: C**

---

**Question: 9**

You develop a new stored procedure for an existing database. You create two tables named Customer and Orders.

The tables have the following definitions: CREATE TABLE Customer

(CustomerID int NOT NULL PRIMARY KEY CLUSTERED,

CustomerName nvarchar(255) NOT NULL,

CustomerAddress nvarchar (1024) NOT NULL) CREATE TABLE Orders

(OrderID int NOT NULL PRIMARY KEY CLUSTERED,

CustomerID int NOT NULL FOREIGN KEY REFERENCES Customer(CustomerID),

OrderDetails nvarchar(MAX)) Users are restricted from accessing table objects directly. You need

to ensure that users are able to retrieve customer data. You need to create a stored procedure

that meets the following requirements:

- Returns a row that contains the name, address, and number of orders made by a customer by specifying the CustomerID value.

- Returns a row even if the customer has made no orders.

- Does not return a row if the CustomerID does not exist. Which Transact-SQL statement or statements should you use?

A. INSERT INTO Tickets VALUES (4, 'sales', 'open', 0), (5, 'support', 'open', 0), (6, 'support', 'open', 1)

B. UPDATE Tickets SET IsArchived = 1 WHERE TicketId IN (1, 2, 3)

C. CREATE PROCEDURE p\_GetTotalOrdersByCustomer (@customerid int) AS SELECT

c.CustomerName, c.CustomerAddress, TotalOrders = COUNT(o.OrderID) FROM Customer c INNER

JOIN Orders o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerName,

c.CustomerAddress WHERE c.CustomerID = @customerid GO

D. CREATE PROCEDURE p\_GetTotalOrdersByCustomer (@customerid int) AS SELECT

c.CustomerName, c.CustomerAddress, TotalOrders = SUM(o.OrderID) FROM Customer c INNER

JOIN Orders o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerName, c.CustomerAddress

GO

E. INSERT INTO SupportTickets VALUES (4, 'support', 'open', 0), (5, 'support', 'in progress', 0), (6,

'support', 'closed', 0)

F. INSERT INTO Tickets VALUES (4, 'support', 'open', 0), (5, 'support', 'in progress', 0), (6, 'support',

'closed', 0)

G. CREATE PROCEDURE p\_GetTotalOrdersByCustomer (@customerid int) AS SELECT

c.CustomerName, c.CustomerAddress, TotalOrders = ISNULL(o.TotalOrders, 0) FROM Customer

c INNER JOIN (SELECT CustomerID, TotalOrders = COUNT(OrderID) FROM Orders WHERE

CustomerID = @customerid GROUP BY CustomerID) o ON c.CustomerID = o.CustomerID WHERE

c.CustomerID = @customerid GO

H. CREATE PROCEDURE p\_GetTotalOrdersByCustomer (@customerid int) AS SELECT

c.CustomerName, c.CustomerAddress, TotalOrders = ISNULL(o.TotalOrders, 0) FROM Customer

cINNER JOIN (SELECT CustomerID, TotalOrders = SUM(OrderID)FROM OrdersWHERE CustomerID = @customeridGROUP BY CustomerID) oON c.CustomerID = o.CustomerIDWHERE c.CustomerID = @customerid GO

---

**Answer: B OR F**

---

**Question: 10**

You administer a Microsoft SQL Server 2008 instance that has two databases. The first database named AdventureWorks contains a table named Sales.SalesOrders.

The Sales.SalesOrders table has the following definition: CREATE TABLE [Sales].[SalesOrderDetail]([SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL, [ProductID] [int] NOT NULL, [OrderQty] [smallint] NOT NULL, [OrderDate] [datetime] NOT NULL, CONSTRAINT [PK\_SalesOrderDetail] PRIMARY KEY CLUSTERED

(  
[SalesOrderDetailID] ASC

)) ON [PRIMARY] The second database named AdventureWorksDW contains a table named dbo.SalesOrderSummary.

The dbo.SalesOrderSummary table has the following definition: CREATE TABLE

[dbo].[SalesOrderSummary] (

ProductID [int] NOT NULL,

OrderQty [int] NOT NULL,

OrderYear [int] NOT NULL,

CONSTRAINT [PK\_SalesOrderSummary] PRIMARY KEY CLUSTERED

(  
OrderYear ASC,  
ProductID ASC

)) ON [PRIMARY] You plan to migrate sales data for the year 2011 from the SalesOrderDetail table into the SalesOrderSummary table. You need to ensure that the following requirements are met:

- All data is removed from the SalesOrderSummary table before migrating data.
- A subset of data is migrated from the SalesOrderDetail table in the AdventureWorks database to the SalesOrderSummary table in the AdventureWorksDW database.
- Migrated data summarizes order quantity in one row per product for the year 2011. 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.)

A. USE AdventureWorksGO

B. USE AdventureWorksDWGO

C. DELETE TABLE dbo.SalesOrderSummaryGO

D. TRUNCATE TABLE dbo.SalesOrderSummaryGO

E. SELECTProductID,SUM(OrderQty) as OrderQty,YEAR(OrderDate) as OrderYear

INTOdbo.SalesOrderSummaryFROMAdventureWorks.Sales.SalesOrderDetailWHEREYEAR(OrderDate)=2011GROUP BYProductID,YEAR(OrderDate)GO

F. SELECTProductID,SUM(OrderQty) AS OrderQty,YEAR(2011) AS

OrderYearINTOdbo.SalesOrderSummaryFROMAdventureWorks.Sales.SalesOrderDetailGROUP  
BYProductIDHAVINGYEAR(2011)=2011GO  
G. INSERT dbo.SalesOrderSummary (ProductID, OrderQty,  
OrderYear)SELECTProductID,SUM(OrderQty) AS OrderQty,YEAR(OrderDate) AS OrderYear  
FROMAdventureWorks.Sales.SalesOrderDetailWHEREYEAR(OrderDate)=2011GROUP  
BYProductID,YEAR(OrderDate)GO  
H. INSERT dbo.SalesOrderSummary (ProductID, OrderQty,  
OrderYear)SELECTProductID,SUM(OrderQty),YEAR(2011)  
FROMAdventureWorks.Sales.SalesOrderDetailGROUP BYProductIDHAVINGYEAR(2011)=2011GO

**Answer: (B BEFORE D) AND  
(D BEFORE G) AND ONLY (B,  
D, G)**

**Question: 11**

A database contains tables named Sales and SalesArchive. SalesArchive contains historical sales data. You configure Change Tracking on the Sales table. The minimum valid version of the Sales table is 10. You need to write a query to export only sales data that changed since version 10, including the primary key of deleted rows.

Which method should you use?

- A. FROM SalesRIGHT JOIN CHANGETABLE (CHANGES Sales, 10) AS C ...
- B. FROM SalesINNER JOIN CHANGETABLE (CHANGES Sales, 10) AS C ...
- C. FROM SalesRIGHT JOIN CHANGETABLE (CHANGES SalesArchive, 10) AS C ...
- D. FROM SalesINNER JOIN CHANGETABLE (CHANGES SalesArchive, 10) AS C ...

**Answer: A**

**Question: 12**

You administer a Microsoft SQL Server 2008 database that contains a stored procedure named dbo.SalesOrderDetails. The stored procedure has following definition: CREATE PROCEDURE dbo.SalesOrderDetails

@CustomerID int,

@OrderDate datetime,

@SalesOrderID intASSELECT

h.SalesOrderID,

h.OrderDate,

d.OrderQty,

d.ProductIDFROM

Sales.SalesOrderHeader h

INNER JOIN

Sales.SalesOrderDetail d

ON d.SalesOrderID = h.SalesOrderIDWHERE

h.CustomerID = @CustomerID

or h.OrderDate > @OrderDate

or h.SalesOrderID > @SalesOrderIDGO Parameter values passed to the stored procedure largely

vary. You discover that the stored procedure executes quickly for some parameters but slowly for other parameters. You need to ensure that the query plan generated is optimized to provide the most consistent execution times for any set of parameters passed to the stored procedure. Which query hint should you use?

- A. OPTION (KEEP PLAN)
- B. OPTION (KEEPFIXED PLAN)
- C. OPTION (OPTIMIZE FOR UNKNOWN)
- D. OPTION (NOLOCK)
- E. OPTION (RECOMPILE)
- F. OPTION (FAST 25)
- G. OPTION (MAXDOP 25)
- H. OPTION (ROBUST PLAN)

**Answer: C OR E**

### Question: 13

You notice that a database server is responding slowly to queries. You run the following dynamic management views (DMV) query on the server. SELECT TOP (10) wait\_type, wait\_time\_ms FROM sys.dm\_os\_wait\_stats ORDER BY wait\_time\_ms DESC; The query returns a top wait type of SOS\_SCHEDULER\_YIELD. You need to identify what is causing the server response issues. Which resource should you investigate first?

- A. Memory
- B. Network
- C. CPU
- D. Disk

**Answer: C**

### Question: 14

You administer a Microsoft SQL Server 2008 database that contains a table named dbo.[order]. There are no triggers on the table. You plan to create a stored procedure that will have the following parameters:

·@ProdId int

·@CustId int You need to ensure that the following requirements are met:

·The OrderID and ProdID values of each modified row are captured into a local table variable before data is modified.

·The ProdID is modified to @ProdID where CustID is equal to @CustId. Which Transact-SQL statement should you use?

- A. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID = @ProdId OUTPUT DELETED.OrderID, DELETED.ProdID INTO @OrderIDs WHERE CustID = @CustId;
- B. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID = @ProdId WHERE CustID = @CustId; INSERT into @OrderIDs SELECT SCOPE\_IDENTITY(OrderId), SCOPE\_IDENTITY(ProdId);
- C. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID =

@ProdIdOUTPUT INSERTED.OrderID, INSERTED.ProdID WHERE CustID = @CustId;  
D. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID = @ProdIdOUTPUT INSERTED.OrderID, INSERTED.ProdID INTO @OrderIDs WHERE CustID = @CustId;  
E. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID = @ProdIdOUTPUT SELECT d.OrderID, d.ProdID FROM DELETED d INTO @OrderIDs WHERE CustID = @CustId;  
F. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE dbo.[order] SET ProdID = @ProdIdOUTPUT INSERTED.OrderID, DELETED.ProdID INTO @OrderIDs WHERE CustID = @CustId;  
G. UPDATE [order] SET ProdID = @ProdIdOUTPUT DELETED.OrderID, DELETED.ProdID INTO @OrderIDs (OrderID INT, ProdID INT) WHERE CustID = @CustId;  
H. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT); UPDATE [order] SET ProdID = @CustIdOUTPUT #INSERTED.OrderID, #INSERTED.ProdID INTO @OrderIDs WHERE CustID = @CustId;

---

**Answer: A OR F**

---

**Question: 15**

You have a table named Inventory. You open a Microsoft Windows PowerShell session at the following location by using the SQL Server Windows PowerShell provider. PS  
SQLSERVER:\SQL\CONTOSO\DEFAULT\Databases\ReportServer\Tables\dbo.Inventory\Columns>  
Using the SQL Server Windows PowerShell provider, you need to query all the columns in the table. Which cmdlet should you use?

- A. Get-ChildItem
- B. Get-ItemProperty
- C. Get-Location
- D. Get-Item

---

**Answer: A**

---

**Question: 16**

You need to ensure that tables are not dropped from your database. What should you do?

- A. Create a DML trigger that contains ROLLBACK.
- B. Create a DML trigger that contains COMMIT.
- C. Create a DDL trigger that contains COMMIT.
- D. Create a DDL trigger that contains ROLLBACK.

---

**Answer: D**

---

**Question: 17**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE [Production].[Product](  
[ProductID] [int] IDENTITY(1,1) NOT NULL,

```
[Name] [nvarchar](50) NOT NULL,  
[ProductNumber] [nvarchar](25) NOT NULL,  
[Color] [nvarchar](15) NULL,  
[Class] [nchar](2) NULL,  
[Style] [nchar](2) NULL,  
[Active] [bit] NOT NULL,  
[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED  
(  
[ProductID] ASC  
) ON [PRIMARY]) ON [PRIMARY]GO
```

You want to add a new field to the Product table to meet the following requirements:

- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support integer values ranging from -10 through 10.

Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)

**Answer: F**

#### Question: 18

You need to round the value 1.75 to the nearest whole number. Which code segment should you use?

- A. Select ROUND(1.75,1.0)
- B. Select ROUND(1.75,2)

- C. Select ROUND(1.75,0)  
D. Select ROUND(1.75,2.0)

**Answer: C**

**Question: 19**

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 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 TRYBEGIN CATCH IF @@TRANCOUNT > 0 BEGINROLLBACK TRANSACTION DataLoad;END; END CATCH;
- B. 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'); COMMIT TRANSACTION; END TRYBEGIN CATCH IF @@TRANCOUNT > 0 BEGINROLLBACK TRANSACTION;END; END CATCH;
- C. BEGIN TRY BEGIN TRANSACTION customers; 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'); SAVE TRANSACTION customers; END TRYBEGIN CATCH IF @@TRANCOUNT > 0 BEGINROLLBACK TRANSACTION customers;END; END CATCH;
```

```
D. 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'); SAVE TRANSACTION; END TRYBEGIN CATCH IF @@TRANCOUNT > 0 BEGINROLLBACK TRANSACTION;END; END CATCH;
```

```
E. 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 TRYBEGIN CATCH IF @@TRANCOUNT > 0 BEGINROLLBACK TRANSACTION;END; END CATCH;
```

```
F. 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;
```

```
G. SET XACT_ABORT ONBEGIN 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;
```

```
H. 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'); IF @@ERROR >
```

ROLLBACK TRANSACTIONELSECOMMIT TRANSACTION;

**Answer: B OR G**

### Question: 20

You are a developer for a Microsoft SQL Server 2008 R2 database instance. You want to add functionality to an existing application that uses the Microsoft SQL Server Profiler tool to capture trace information. You need to ensure that the following requirements are met:

- Users are able to use the SQL Server Profiler tool.
- Users are able to capture trace information and read saved trace files.
- Users are granted the minimum permissions to achieve these goals. Which three actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

- A. Create a database role named traceusers.
- B. Add each user of the application to the sysadmin fixed-server role.
- C. Add each user of the application to the traceusers database role.
- D. Grant the ALTER TRACE permissions to the traceusers database role.
- E. Grant the CONTROL SERVER permissions to the traceusers database role.
- F. Create a server role named traceusers.

**Answer: (C BEFORE D) AND  
(A BEFORE C) AND ONLY (A,  
C, D)**

### Question: 21

You are a database developer responsible for maintaining an application.

The application has a table named Programs that has the following definition: CREATE TABLE

[dbo].[Customers]([ID] int NOT NULL IDENTITY(1,1),

[Name] varchar(100) NOT NULL,

[Address] varchar(255) NOT NULL,

[City] varchar(100) NOT NULL,

[State] char(2) NOT NULL,

[Postal] varchar(5) NOT NULL,

[Priority] char(1) NOT NULL,

[Active] bit NOT NULL)GOALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers

PRIMARY KEY NONCLUSTERED (Name)GO You need to modify the Customers table to meet the following requirements:

- ID must be the Primary Key.
- The clustered index must be on the ID column.
- The Active column must have a default value of 1, and must allow values of only 0 or 1.
- The Priority column must have values of "a", "b", or "c".
- The Postal column must contain a correctly formatted five-digit numeric Postal Code. Which Transact-SQL statement or statements should you use?

- A. ALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY CLUSTERED (ID)
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Priority CHECK ([Priority] IN ('a', 'b',

- 'c'))ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Postal CHECK ([Postal] LIKE '0-9][0-9][0-9][0-9][0-9]')
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT DF\_Active DEFAULT (1) FOR [Active]ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Active CHECK ([Active] IN (0,1))
- B. ALTER TABLE [dbo].[Customers] DROP CONSTRAINT PK\_CustomersALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY CLUSTERED (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Priority CHECK ([Priority] IN ('a', 'b', 'c'))ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Postal CHECK ([Postal] LIKE '0-9][0-9][0-9][0-9][0-9]')
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT DF\_Active DEFAULT (1) FOR [Active]ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Active CHECK ([Active] IN (0,1))
- C. ALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY CLUSTERED (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Priority VALIDATE ([Priority] IN ('a', 'b', 'c'))ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Postal VALIDATE ([Postal] LIKE '0-9][0-9][0-9][0-9][0-9]')
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT DF\_Active DEFAULT (1) FOR [Active]ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Active VALIDATE ([Active] IN (0,1))
- D. ALTER TABLE [dbo].[Customers] DROP CONSTRAINT PK\_CustomersALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY NONCLUSTERED (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Priority CHECK ([Priority] IN ('a', 'b', 'c'))ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Postal CHECK ([Postal] LIKE '0-9][0-9][0-9][0-9][0-9]')
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT DF\_Active DEFAULT (1) FOR [Active]ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Active CHECK ([Active] IN (0,1))
- E. ALTER TABLE [dbo].[Customers] DROP CONSTRAINT PK\_CustomersALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_DataChecks CHECK ([Priority] LIKE 'abc') AND ISNUMERIC([Postal]) = 1 AND LEN([Postal]) = 5 AND [Active] IN (0,1))
- F. ALTER CONSTRAINT PK\_Customers SET CLUSTERED (ID)ALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY CLUSTERED (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Priority CHECK ([Priority] IN ('a', 'b', 'c'))ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Postal CHECK ([Postal] LIKE '0-9][0-9][0-9][0-9][0-9]')
- ALTER TABLE [dbo].[Customers] ADD CONSTRAINT DF\_Active DEFAULT (1) FOR [Active]ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_Active CHECK ([Active] IN (0,1))
- G. ALTER TABLE [dbo].[Customers] DROP CONSTRAINT PK\_CustomersALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_DataChecks CHECK ([Priority] = 'abc') AND LEN([Postal]) = 5 AND [Active] IN (0,1))
- H. ALTER TABLE [dbo].[Customers] DROP CONSTRAINT PK\_CustomersALTER TABLE [dbo].[Customers] ADD CONSTRAINT PK\_Customers PRIMARY KEY (ID) ALTER TABLE [dbo].[Customers] ADD CONSTRAINT CK\_DataChecks CHECK ([Priority] IN 'abc') AND ISNUMERIC([Postal]) = 1 AND [Active] IN (0,1))

---

**Answer: B OR E**

---

**Question: 22**

You administer a Microsoft SQL Server 2008 database that contains a table named Sales.SalesOrderHeader. The Sales.SalesOrderHeader table has the following definition: CREATE TABLE [Sales].[SalesOrderHeader](  
[SalesOrderID] [int] IDENTITY(1,1) NOT NULL,  
[OrderDate] [datetime] NOT NULL,  
[DueDate] [datetime] NOT NULL,  
[ShipDate] [datetime] NULL,  
[Status] [tinyint] NOT NULL,  
[CustomerID] [int] NOT NULL,  
[SalesPersonID] [int] NULL,  
[SubTotal] [money] NOT NULL,  
[TaxAmt] [money] NOT NULL,  
CONSTRAINT [PK\_OrderHeader\_SalesOrderID]  
PRIMARY KEY CLUSTERED  
(  
[SalesOrderID] ASC  
) ON [PRIMARY]GO

You want to generate an execution plan of XML output document for a query that displays all Sales.SalesOrderHeader records containing orders that have not been shipped. You need to ensure that the following requirements are met:

- An XML document is provided only for this query.
- An XML document that contains the estimated execution plan is returned. 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.)

- A. SET SHOWPLAN\_XML ON;GO
- B. SET SHOWPLAN\_XML OFF;GO
- C. SELECT \*FROM Sales.SalesOrderHeaderWHERE ShipDate = NULLGO
- D. SELECT \*FROM Sales.SalesOrderHeaderWHERE ShipDate IS NULLGO
- E. GRANT SHOWPLAN XML ON;GO
- F. GRANT SHOWPLAN XML OFF;GO
- G. SET STATISTICS XML ON;GO
- H. SET STATISTICS XML OFF;GO

**Answer: (A BEFORE D) AND  
(D BEFORE B) AND ONLY (A,  
D, B)**

**Question: 23**

You have a table named Customers that has an XML column named CustomerData. There are currently no indexes on the table. You use the following WHERE clause in a query: WHERE CustomerData.exist ('/CustomerDemographic/@Age[.>="21"]') = 1 You need to create indexes for the query.

Which Transact-SQL statements should you use?

- A. CREATE PRIMARY XML INDEX PXML\_IDX\_CustomerON Customers(CustomerData); CREATE XML

INDEX SXML\_IDX\_Customer ON Customer(CustomerData) USING XML INDEX  
PXML\_IDX\_Customer FOR PATH;  
B. CREATE PRIMARY XML INDEX PXML\_IDX\_Customer ON Customers(CustomerData); CREATE XML  
INDEX SXML\_IDX\_Customer ON Customer(CustomerData) USING XML INDEX  
PXML\_IDX\_Customer FOR VALUE;  
C. CREATE CLUSTERED INDEX CL\_IDX\_Customer ON Customers(CustomerID); CREATE PRIMARY  
XML INDEX PXML\_IDX\_Customer ON Customers(CustomerData); CREATE XML INDEX  
SXML\_IDX\_Customer\_Property ON Customer(CustomerData) USING XML INDEX  
PXML\_IDX\_Customer FOR VALUE;  
D. CREATE CLUSTERED INDEX CL\_IDX\_Customer ON Customers(CustomerID); CREATE PRIMARY  
XML INDEX PXML\_IDX\_Customer ON Customers(CustomerData); CREATE XML INDEX  
SXML\_IDX\_Customer ON Customer(CustomerData) USING XML INDEX PXML\_IDX\_Customer FOR  
PATH;

---

**Answer: D**

---

**Question: 24**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application.

You create tables named complaint, customer, and product as follows: CREATE TABLE

[dbo].[complaint] ([ComplaintID] [int],

[ProductID] [int],

[CustomerID] [int],

[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],

[CustomerName] [varchar](100),

[Address] [varchar](200),

[City] [varchar](100),

[State] [varchar](50),

[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],

[ProductName] [varchar](100),

[SalePrice] [money],

[ManufacturerName] [varchar](100)); You need to write a query to identify all customers who

have complained about products that have an average sales price of 500 or more from September 01, 2011. Which SQL query should you use?

A. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

B. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());

C. SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) > 10 GROUP BY c.CustomerName;

- D. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500
- E. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;
- F. SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;
- G. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));
- H. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;
- I. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500
- J. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);

---

**Answer: I**

---

**Question: 25**

You have two tables named dbo.Products and dbo.PriceChange. Table dbo.Products contains ten products. Five products are priced at \$20 per unit and have PriceIncrease set to 1. The other five products are priced at \$10 per unit and have PriceIncrease set to 0. You have the following query:

```
INSERT dbo.PriceChange (ProductID, Change, ChangeDate) SELECT ProductID, inPrice - delPrice, SYSDATETIME() FROM (
```

```
UPDATE dbo.Products
```

```
SET Price *= 1.1
```

```
OUTPUT inserted.ProductID, inserted.Price, deleted.Price
```

```
WHERE PriceIncrease = 1) p (ProductID, inPrice, delPrice);
```

You need to predict the results of the query. Which results should the query produce?

- A. Five rows are updated in dbo.Products. No rows are inserted into dbo.PriceChange.
- B. No rows are updated in dbo.Products. Five rows are inserted into dbo.PriceChange.
- C. No rows are updated in dbo.Products. No rows are inserted into dbo.PriceChange.
- D. Five rows are updated in dbo.Products. Five rows are inserted into dbo.PriceChange.

---

**Answer: D**

---

**Question: 26**

You are tasked to create a table that has a column that must store the current time accurate to



ten microseconds. You need to use a system function in conjunction with the DEFAULT option in the column definition. Which system function should you use?

- A. DATEADD
- B. SYSDATETIME
- C. CURRENT\_TIMESTAMP
- D. GETUTCDATE

---

**Answer: B**

---

**Question: 27**

You are a developer of a Microsoft SQL Server 2008 R2 database instance that supports a web-based order-entry application. You need to create a server-side trace that meets the following requirements:

- Captures performance information from 05:00 hours to 05:30 hours daily.
  - Stores trace information in a format that can be used even while users are disconnected from the network. Which four actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)
- A. Create a server-side trace by using the SQL Server Profiler tool. Configure the trace to start at 05:00 hours and to stop at 05:30 hours.
  - B. Save the trace information to a SQL table.
  - C. Save the trace information to a trace file.
  - D. Export the trace file to a SQL script.
  - E. Create a server-side trace by using the SQL Server Profiler tool. Configure the trace to stop at 05:30 hours.
  - F. Create a SQL Server Agent job by using the exported SQL script. Schedule the job to start at 05:00 hours daily.

---

**Answer: (E BEFORE C) AND (C BEFORE D) AND (D BEFORE F) AND ONLY (E, C, D, F)**

---

**Question: 28**

You administer a Microsoft SQL Server 2008 database that contains a table named Sales.SalesOrderHeader. The Sales.SalesOrderHeader table has the following definition: CREATE

```
TABLE [Sales].[SalesOrderHeader](  
[SalesOrderID] [int] IDENTITY(1,1) NOT NULL,  
[OrderDate] [datetime] NOT NULL,  
[DueDate] [datetime] NOT NULL,  
[ShipDate] [datetime] NULL,  
[Status] [tinyint] NOT NULL,  
[CustomerID] [int] NOT NULL,  
[SalesPersonID] [int] NULL,  
[SubTotal] [money] NOT NULL,
```



```
[TaxAmt] [money] NOT NULL,  
CONSTRAINT [PK_OrderHeader_SalesOrderID]  
PRIMARY KEY CLUSTERED  
(
```

```
[SalesOrderID] ASC
```

```
) ) ON [PRIMARY]GO
```

You plan to perform the following tasks:

- Display all Sales.SalesOrderHeader records that contain orders that have not been shipped.
- Generate an execution plan of XML output document for the query. You need to ensure that the following requirements are met:
  - Data is returned for all fields in the Sales.SalesOrderHeader table where the shipping date is unknown.
  - An XML document is provided only for this query.
  - An XML document that contains the actual execution plan is returned. 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.)

- A. SET SHOWPLAN\_XML ON;GO
- B. SET SHOWPLAN\_XML OFF;GO
- C. SELECT \*FROM Sales.SalesOrderHeaderWHERE ShipDate = NULLGO
- D. SELECT \*FROM Sales.SalesOrderHeaderWHERE ShipDate IS NULLGO
- E. GRANT SHOWPLAN XML ON;GO
- F. GRANT SHOWPLAN XML OFF;GO
- G. SET STATISTICS XML ON;GO
- H. SET STATISTICS XML OFF;GO

**Answer: (G BEFORE D) AND  
(D BEFORE H) AND ONLY (G,  
D, H)**

#### Question: 29

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application. You create tables named complaint, customer, and product as follows: CREATE TABLE

```
[dbo].[complaint] ([ComplaintID] [int],  
[ProductID] [int],  
[CustomerID] [int],  
[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],  
[CustomerName] [varchar](100),  
[Address] [varchar](200),  
[City] [varchar](100),  
[State] [varchar](50),  
[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],  
[ProductName] [varchar](100),  
[SalePrice] [money],
```

[ManufacturerName] [varchar](100)); You need to write a query to sum the sales made to each customer who has made a complaint by the following entries:

·The customer name and product name

·The grand total of all sales Which SQL query should you use?

- A. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());`
- B. `SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));`
- C. `SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;`
- D. `SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;`
- E. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500`
- F. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());`
- G. `SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) > 10 GROUP BY c.CustomerName;`
- H. `SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;`
- I. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500`
- J. `SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);`

**Answer: F**

### Question: 30

You administer a Microsoft SQL Server 2008 R2 database instance named AdventureWorks. A user who has the db\_datareader permissions on the AdventureWorks database wants to view detailed information about how the following query will be executed:

`SELECT * FROM Sales.SalesOrderHeader WHERE OnlineOrderFlag = 1 AND SubTotal > 500` You need to ensure that the user can view the following information in a data grid without executing

the query:

- Estimated number of rows of output
  - Estimated I/O cost
  - Estimated CPU cost Which two actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)
- A. Grant the following permission to the user: GRANT SHOWPLAN
- B. Grant the following permission to the user: GRANT EXECUTE ON XML SCHEMA COLLECTION
- C. Grant the following permission to the user: GRANT SELECT ON OBJECT::Sales.SalesOrderHeader
- D. Request the user to run the following command: SET SHOWPLAN\_XML ON
- E. Request the user to run the following command: SET SHOWPLAN\_ALL ON
- F. Request the user to run the following command: SET STATISTICS IO ON
- G. Request the user to run the following command: SET STATISTICS XML ON

**Answer: (A BEFORE E) AND ONLY (A, E)**

#### Question: 31

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,t.Name AS [Territory]FROM Sales.SalesPerson pCROSS JOIN Sales.SalesTerritory tWHERE p.TerritoryId = t.TerritoryId
- B. SELECT p.SalesPersonId,t.Name AS [Territory]FROM Sales.SalesPerson pCROSS JOIN Sales.SalesTerritory t
- C. SELECT p.SalesPersonId,t.Name AS [Territory]FROM Sales.SalesPerson pINNER JOIN Sales.SalesTerritory tON p.TerritoryId = t.TerritoryId
- D. SELECT p.SalesPersonId,t.Name AS [Territory]FROM Sales.SalesPerson pFULL JOIN Sales.SalesTerritory tON p.TerritoryId = t.TerritoryId

**Answer: B**

#### Question: 32

You are given a database design to evaluate. All of the tables in this database should have a clustered index. You need to determine the tables that are missing a clustered index by using the system catalog views. Which Transact-SQL statement should you use?

- A. SELECT name AS table\_nameFROM sys.tablesWHERE OBJECTPROPERTY(object\_id,'TableHasClustIndex') = 0ORDER BY name;
- B. SELECT name AS table\_nameFROM sys.tablesWHERE OBJECTPROPERTY(object\_id,'TableHasClustIndex') = 0 AND OBJECTPROPERTY(object\_id,'TableHasUniqueCnst') = 1ORDER BY name;
- C. SELECT name AS table\_nameFROM sys.tablesWHERE OBJECTPROPERTY(object\_id,'TableHasUniqueCnst') = 0ORDER BY name;
- D. SELECT name AS table\_nameFROM sys.tablesWHERE

OBJECTPROPERTY(object\_id,'TableHasClustIndex') = 1 AND  
OBJECTPROPERTY(object\_id,'TableHasUniqueCnst') = 1 ORDER BY name;

**Answer: A**

**Question: 33**

You work for an international charity organization.

You are writing a query to list the highest 100 different amounts that were donated. You have written the following code segment: 01 SELECT \*02 FROM (SELECT Customer.CustomerID, SUM(TotalDue) AS TotalGiven, 03 04

FROM Customer05

JOIN SalesOrder 06

ON Customer.CustomerID = SalesOrder.CustomerID07

GROUP BY Customer.CustomerID) AS DonationsToFilter08 WHERE FilterCriteria <= 100 You need to insert a Transact-SQL clause in line 03 to complete the query. Which Transact-SQL clause should you insert?

- A. NTILE(100) OVER (ORDER BY SUM(TotalDue) DESC) AS FilterCriteria
- B. DENSE\_RANK() OVER (ORDER BY SUM(TotalDue) DESC) AS FilterCriteria
- C. ROW\_NUMBER() OVER (ORDER BY SUM(TotalDue) DESC) AS FilterCriteria
- D. RANK() OVER (ORDER BY SUM(TotalDue) DESC) AS FilterCriteria

**Answer: B**

**Question: 34**

You are a database developer for your organization. You have an application hosted on Microsoft SQL Server 2008 R2.

One of the tables in the application database has the following definition: CREATE TABLE XMLData(XMLPage xml NOT NULL) No indexes, keys, or constraints are defined along with the table. The table currently contains more than one million entries. An example of the XML data is shown below: Company

<Employee Name = "Mark" = "Department Finance"/>

<Employee Name = "Peter" = "Department Sales"/>

<Employee Name = "Susan" = "Department Facilities"/>Company

A stored procedure is executed in the application.

The stored procedure has the following definition:

CREATE PROCEDURE p\_GetFinanceUser(@Name varchar(20))AS SELECT Name =

Company.Employee.value('@Name','varchar(20)')

,Department = Company.Employee.value('@Department','varchar(20)')FROM XMLData cCROSS

APPLY XMLPage.nodes('/Company/Employee') Department(Employee)WHERE

Company.Employee.exist('@Department[.="Finance"]') = 1 Users report that the response time of the stored procedure has decreased. You need to ensure that the response time of the stored procedure is improved. 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.)

- A. CREATE INDEX idx\_XMLPage ON XMLData (XMLPage)

- B. CREATE PRIMARY XML INDEX idx\_XMLPage ON XMLData (XMLPage)
- C. CREATE XML INDEX idx\_XMLPage\_PATH ON XMLData(XMLPage) USING XML INDEX idx\_XMLPageFOR PATH
- D. CREATE XML INDEX idx\_XMLPage\_VALUE ON XMLData(XMLPage) USING XML INDEX idx\_XMLPageFOR VALUE
- E. CREATE XML INDEX idx\_XMLPage\_PROPERTY ON XMLData(XMLPage) USING XML INDEX idx\_XMLPageFOR PROPERTY
- F. ALTER TABLE XMLData ADD XMLKey int IDENTITY (1,1) UNIQUE NOT NULL
- G. ALTER TABLE XMLData ADD XMLKey int IDENTITY (1,1) PRIMARY KEY CLUSTERED NOT NULL

**Answer: (G BEFORE B) AND  
(B BEFORE C) AND ONLY (G,  
B, C)**

**Question: 35**

You have a table named Books that has columns named BookTitle and Description. There is a full-text index on these columns. You need to return rows from the table in which the word 'computer' exists in either column. Which code segment should you use?

- A. SELECT \* FROM Books WHERE BookTitle LIKE '%computer%'
- B. SELECT \* FROM Books WHERE BookTitle = '%computer%' OR Description = '%computer%'
- C. SELECT \* FROM Books WHERE FREETEXT(BookTitle, 'computer')
- D. SELECT \* FROM Books WHERE FREETEXT(\*, 'computer')

**Answer: D**

**Question: 36**

You administer a Microsoft SQL Server 2008 database that includes a table named ProductDetails.

The Products table has the following schema: CREATE TABLE dbo.ProductDetails(

ProductID nchar(4) NOT NULL,

ProductName nvarchar(50) NULL,

ProductDescription nvarchar(50) NULL,

UnitCost money NULL,

UnitPrice money NULL,

ReOrderLevel int NOT NULL,

CONSTRAINT PK\_ProductDetails PRIMARY KEY CLUSTERED

(

ProductID ASC

)) GO You create a User-Defined Function (UDF) in the same database.

The UDF has the following schema: CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4)) RETURNS Money WITH SCHEMABINDING AS BEGIN

DECLARE @Profit Money;

SELECT @Profit = UnitPrice - UnitCost FROM dbo.ProductDetails WHERE ProductID =

@ProductID;

RETURN @Profit; END You need to meet the following requirements:

-Ensure that the UnitPrice column in the ProductDetails table does not accept NULL values.

- Use the CalculateProductProfit() UDF to calculate profit for each product.
  - Avoid accidental drop or change of the Product table. 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.)
- A. DROP TABLE ProductDetails
- B. DROP FUNCTION CalculateProductProfit
- C. ALTER TABLE ProductDetails ALTER COLUMN UnitPrice Money NULL
- D. ALTER TABLE ProductDetails ALTER COLUMN UnitPrice Money NOT NULL
- E. CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyWITH SCHEMABINDINGASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM dbo.ProductDetails WHERE ProductID = @ProductID;RETURN @Profit;END
- F. CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM dbo.ProductDetails WHERE ProductID = @ProductID;RETURN @Profit;END
- G. CREATE FUNCTION CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyWITH SCHEMABINDINGASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM dbo.ProductDetails WHERE ProductID = @ProductID;RETURN @Profit;END

**Answer: (B BEFORE D) AND (D BEFORE E) AND ONLY (B, D, E)**

**Question: 37**

A table named Locations contains 5000 locations and the number of sales made at each location. You need to display the top 5 percent of locations by sales made. Which Transact-SQL code segment should you use?

- A. WITH Percentages AS ( SELECT \*, NTILE(20) OVER (ORDER BY SalesMade) AS groupingColumn FROM Locations) SELECT \* FROM Percentages WHERE groupingColumn = 20;
- B. WITH Percentages AS (SELECT \*, NTILE(5) OVER (ORDER BY SalesMade) AS groupingColumn FROM Locations) SELECT \* FROM Percentages WHERE groupingColumn = 1;
- C. WITH Percentages AS (SELECT \*, NTILE(20) OVER (ORDER BY SalesMade) AS groupingColumn FROM Locations) SELECT \* FROM Percentages WHERE groupingColumn = 1;
- D. WITH Percentages AS (SELECT \*, NTILE(5) OVER (ORDER BY SalesMade) AS groupingColumn FROM Locations) SELECT \* FROM Percentages WHERE groupingColumn = 5;

**Answer: A**

**Question: 38**

Your company manufactures and distributes bowling balls. You have a full-text catalog named ftCatalog which contains the ftInventory index on the Products table. Your marketing department has just inserted a new bowling ball into the Inventory table. You notice only the new bowling ball is not being included in the results of the full-text searches. You have confirmed that the row exists in the Products table.

You need to update the full-text catalog in the least amount of time. Which Transact-SQL statement should you use?



- A. ALTER FULLTEXT INDEX ON ftInventoryRESUME POPULATION
- B. ALTER FULLTEXT CATALOG ftCatalog REBUILD
- C. ALTER FULLTEXT INDEX ON ftInventorySTART UPDATE POPULATION
- D. ALTER FULLTEXT INDEX ON ftInventorySTART FULL POPULATION

---

**Answer: C**

---

**Question: 39**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application. You create tables named complaint, customer, and product as follows: CREATE TABLE

[dbo].[complaint] ([ComplaintID] [int],

[ProductID] [int],

[CustomerID] [int],

[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],

[CustomerName] [varchar](100),

[Address] [varchar](200),

[City] [varchar](100),

[State] [varchar](50),

[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],

[ProductName] [varchar](100),

[SalePrice] [money],

[ManufacturerName] [varchar](100)); You need to write a query to sum the monthly sales of each product. Which SQL query should you use?

A. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;

B. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());

C. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);

D. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500

E. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));

F. SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;



G. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500

H. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

I. SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) > 10 GROUP BY c.CustomerName;

J. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;

**Answer: A**

#### Question: 40

You have the following two tables. Products

ProductID	ProductName	VendorID1
1	Product1	02
2	Product2	13
3	Product3	14

ProductChanges

ProductID	ProductName	VendorID1
1	Product1	12
2	Product2	13
3	NewProduct3	25
4	Product4	05
5	Product5	1

You execute the following statement. MERGE Products USING ProductChanges ON (Products.ProductID = ProductChanges.ProductID) WHEN MATCHED AND Products.VendorID = 0 THEN DELETE WHEN MATCHED THEN UPDATE SET Products.ProductName = ProductChanges.ProductName, Products.VendorID = ProductChanges.VendorID; You need to identify the rows that will be displayed in the Products table. Which rows will be displayed?

A. ProductID ProductName VendorID1 Product1 12 Product2 13 NewProduct3 24 Product4 05 Product5 1

B. ProductID ProductName VendorID1 Product1 12 Product2 13 NewProduct3 25 Product5 1

C. ProductID ProductName VendorID2 Product2 13 NewProduct3 24 Product4 0

D. ProductID ProductName VendorID2 Product2 13 NewProduct3 2

**Answer: C**

#### Question: 41

You have a database that contains two tables named ProductCategory and ProductSubCategory. You need to write a query that returns a list of product categories that contain more than ten

sub-categories. Which query should you use?

- A. SELECT [Name]FROM ProductSubCategoryWHERE ProductCategoryID IN (SELECT ProductCategoryIDFROM ProductCategory)GROUP BY [Name]HAVING COUNT(\*) > 10
- B. SELECT [Name]FROM Product Category cWHERE NOT EXISTS (SELECT ProductCategoryIDFROM ProductSubCategoryWHERE ProductCategoryID = c.ProductCategoryIDGROUP BY ProductCategoryIDHAVING COUNT(\*) > 10)
- C. SELECT [Name]FROM ProductSubCategoryWHERE ProductCategoryID NOT IN (SELECT ProductCategoryIDFROM ProductCategory)GROUP BY [Name]HAVING COUNT(\*) > 10
- D. SELECT [Name]FROM Product Category cWHERE EXISTS (SELECT ProductCategoryIDFROM ProductSubCategoryWHERE ProductCategoryID = c.ProductCategoryIDGROUP BY ProductCategoryIDHAVING COUNT(\*) > 10)

**Answer: D**

#### Question: 42

You are a developer of a Microsoft SQL Server 2008 R2 database instance that supports a web-based order-entry application. You need to create a server-side trace that meets the following requirements:

- Captures performance information from 07:00 hours to 07:30 hours daily.
  - Stores trace information in a portable format. Which four actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)
- A. Create a server-side trace by using the SQL Server Profiler tool. Configure the trace to start at 07:00 hours and to stop at 07:30 hours.
  - B. Save the trace information to a SQL table.
  - C. Save the trace information to a trace file.
  - D. Export the trace file to a SQL script.
  - E. Create a server-side trace by using the SQL Server Profiler tool. Configure the trace to stop at 07:30 hours.
  - F. Create a SQL Server Agent job by using the exported SQL script. Schedule the job to start at 07:00 hours daily.

**Answer: (E BEFORE C) AND (C BEFORE D) AND (D BEFORE F) AND ONLY (E, C, D, F)**

#### Question: 43

You have two views named Sales.SalesSummaryOverall and Sales.CustomerAndSalesSummary.

They are defined as follows: CREATE VIEW Sales.SalesSummaryOverallAS

SELECT CustomerId, SUM(SalesTotal) AS OverallTotal

FROM Sales.SalesOrder

GROUP BY CustomerIdGOCREATE VIEW Sales.CustomerAndSalesSummaryAS

SELECT Customer.Name,

SalesSummaryOverall.OverallTotal,

```
(SELECT AVG(OverallTotal)
FROM Sales.SalesSummaryOverall
WHERE SalesSummaryOverall.CustomerId =
Customer.CustomerId) AS avgOverallTotal,
(SELECT MAX(OverallTotal)
FROM Sales.SalesSummaryOverall
WHERE SalesSummaryOverall.CustomerId =
Customer.CustomerId) AS maxOverallTotal,
FROM Sales.Customer
LEFT OUTER JOIN Sales. Sales.SalesSummaryOverall
ON SalesSummaryByYear.CustomerId = Customer.CustomerId
```

GO You have been tasked to modify the Sales.CustomerAndSalesSummary view to remove references to other views. You need to identify a feature to use in the modified version of the Sales.CustomerAndSalesSummary object to achieve the task. Which feature should you use?

- A. Table variables
- B. Common table expressions
- C. Temporary tables
- D. User-defined table types

---

**Answer: B**

---

**Question: 44**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[Active] [bit] NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
[ProductID] ASC
)
```

ON [PRIMARY]) ON [PRIMARY]GO You want to add a new field to the Product table to meet the following requirements:

- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support U.S. dollar currency values that

have two decimals and as large as 500,000 U.S. dollars. Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE
- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY
- R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT

**Answer: Q**

#### Question: 45

You administer a Microsoft SQL Server 2008 database that contains a table named Sales.SalesOrderDetail and a view named Sales.ProductOrders.

The view has the following definition: CREATE VIEW Sales.ProductOrders AS SELECT ProductID,

COUNT(\*) as NbrOfOrders,

SUM(OrderQty) as TotalOrderQtyFROM

Sales.SalesOrderDetailGROUP BY

ProductIDGO The Sales.SalesOrderDetail table contains 5 million rows. Report queries that join to this view consume excessive disk I/O. You need to create an index on the view. Which

Transact-SQL statement or statements should you use?

- A. ALTER VIEW Sales.ProductOrders WITH SCHEMABINDING ASSELECTProductID,COUNT(\*) as NbrOfOrders,SUM(OrderQty) as TotalOrderQtyFROMSales.SalesOrderDetailGROUP BYProductIDGOCREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO
- B. ALTER VIEW Sales.ProductOrders WITH SCHEMABINDING ASSELECTProductID,COUNT\_BIG(\*) as NbrOfOrders,SUM(OrderQty) as TotalOrderQtyFROMSales.SalesOrderDetail WITH (NOLOCK)GROUP BYProductIDGOCREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO
- C. ALTER VIEW Sales.ProductOrders WITH SCHEMABINDING ASSELECTProductID,COUNT\_BIG(\*) as NbrOfOrders,SUM(OrderQty) as TotalOrderQtyFROMSales.SalesOrderDetailGROUP BYProductIDGOCREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders

(ProductID)GO

D. ALTER VIEW Sales.ProductOrders AS SELECT ProductID, COUNT\_BIG(\*) as NbrOfOrders, SUM(OrderQty) as TotalOrderQty FROM Sales.SalesOrderDetail GROUP BY ProductID GO CREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO

E. IF EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[Sales].[ProductOrders]')) DROP VIEW [Sales].[ProductOrders] GO CREATE VIEW Sales.ProductOrders WITH SCHEMABINDING AS SELECT ProductID, COUNT\_BIG(\*) as NbrOfOrders, SUM(OrderQty) as TotalOrderQty FROM Sales.SalesOrderDetail GROUP BY ProductID GO CREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO

F. IF EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[Sales].[ProductOrders]')) DROP VIEW [Sales].[ProductOrders] GO CREATE VIEW Sales.ProductOrders AS SELECT ProductID, COUNT\_BIG(\*) as NbrOfOrders, SUM(OrderQty) as TotalOrderQty FROM Sales.SalesOrderDetail GROUP BY ProductID GO CREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO

G. IF EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[Sales].[ProductOrders]')) DROP VIEW [Sales].[ProductOrders] GO CREATE VIEW Sales.ProductOrders WITH SCHEMABINDING AS SELECT ProductID, COUNT\_BIG(\*) as NbrOfOrders, SUM(OrderQty) as TotalOrderQty FROM Sales.SalesOrderDetail WITH (NOLOCK) GROUP BY ProductID GO CREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO

H. IF EXISTS (SELECT \* FROM sys.views WHERE object\_id = OBJECT\_ID(N'[Sales].[ProductOrders]')) DROP VIEW [Sales].[ProductOrders] GO CREATE VIEW Sales.ProductOrders WITH SCHEMABINDING AS SELECT ProductID, COUNT(\*) as NbrOfOrders, SUM(OrderQty) as TotalOrderQty FROM Sales.SalesOrderDetail GROUP BY ProductID GO CREATE UNIQUE CLUSTERED INDEX IX\_V\_ProductOrders ON Sales.ProductOrders (ProductID)GO

---

**Answer: C OR E**

---

**Question: 46**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[Active] [bit] NOT NULL,
```

```
[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED  
(  
[ProductID] ASC  
) ON [PRIMARY]) ON [PRIMARY]GO
```

You want to add a new field to the Product table to meet the following requirements:

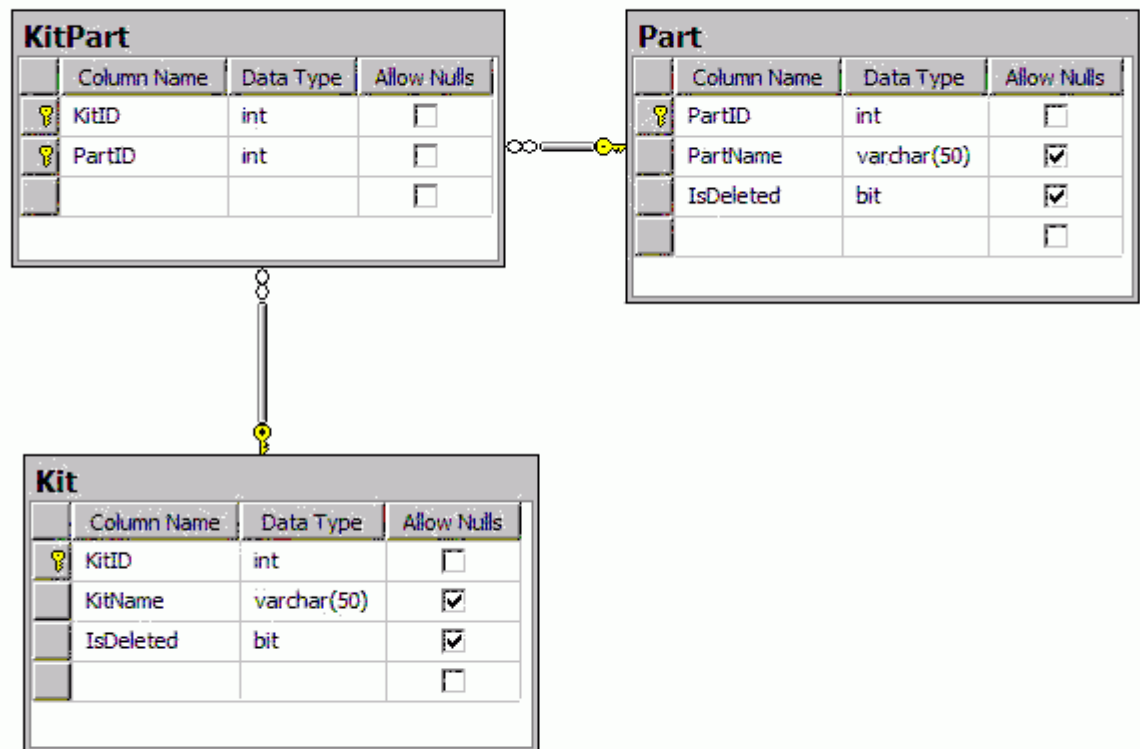
- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support date values that are no more precise than to the hour.

Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME
- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT
- Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)
- R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE

**Answer: O**

**Question: 47**



You are developing a database using Microsoft SQL Server 2008. The database contains the tables shown in the following exhibit. (Click the Exhibit button.) You are required to prevent parts from being deleted if they belong to a kit. If a part belongs to a kit, the delete should not occur and the IsDeleted column for the row should be changed to 'True'. Parts should be deleted if they do not belong to a kit. You have the following Transact-SQL statement to be used in a trigger:

```
UPDATE p
SET IsDeleted = 1
FROM KitPart kp
JOIN deleted d ON kp.PartID = d.PartID
JOIN Part p ON kp.PartID = p.PartID;
DELETE FROM p
FROM Part p
JOIN deleted d ON p.PartID = d.PartID
LEFT OUTER JOIN KitPart kp ON p.PartID = kp.PartID
WHERE kp.KitID IS NULL;
```

You need to implement the Transact-SQL statement in a trigger. Which trigger syntax should you use?

- A. CREATE TRIGGER tr\_Part\_d ON PartINSTEAD OF DELETE AS BEGIN ...END
- B. CREATE TRIGGER tr\_Part\_d ON PartAFTER DELETE AS BEGIN ...END



- C. CREATE TRIGGER tr\_KitPart\_d ON KitPartAFTER DELETE AS BEGIN ...END  
D. CREATE TRIGGER tr\_KitPart\_d ON KitPartINSTEAD OF DELETE AS BEGIN ...END

**Answer: A**

**Question: 48**

You are using TRY...CATCH error handling. You need to raise an error that will pass control to the CATCH block. Which severity level should you use?

- A. 10  
B. 16  
C. 9  
D. 0

**Answer: B**

**Question: 49**

You need to alter stored procedures to use the WITH RECOMPILE option.

Which types of stored procedures should you alter? (Each correct answer represents a complete solution. Choose two.)

- A. Stored procedures implemented from CLR assemblies.  
B. Stored procedures that require the FOR REPLICATION option.  
C. Stored procedures that require the WITH ENCRYPTION option.  
D. Stored procedures that contain queries that use the OPTION (RECOMPILE) hint.

**Answer: C AND D**

**Question: 50**

Your company uses an application that passes XML to the database server by using stored procedures. The database server has a large number of XML handles that are currently active. You determine that the XML is not being flushed from SQL Server memory. You need to identify the system stored procedure to flush the XML from memory. Which Transact-SQL statement should you use?

- A. sp\_xml\_preparedocument  
B. sp\_xml\_removedocument  
C. DBCC DROPCLEANBUFFERS  
D. DBCC FREEPROCACHE

**Answer: B**

**Question: 51**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance. You create tables named order, customer, and product as follows: CREATE TABLE [dbo].[order] ([OrderID] [int], [ProductID] [int],

[CustomerID] [int],  
[OrderDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],  
[CustomerName] [varchar](100),  
[Address] [varchar](200),  
[City] [varchar](100),  
[State] [varchar](50),  
[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],  
[ProductName] [varchar](100),  
[SalePrice] [money],  
[ManufacturerName] [varchar](100)); You need to write a query to return all customer names and total number of orders for customers who have placed more than 10 orders. Which SQL query should you use?

- A. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500
- B. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' AND AVG(p.SalePrice) >= 500
- C. SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, DATEPART(mm, o.OrderDate);
- D. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());
- E. SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE;
- F. SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, o.OrderDate));
- G. SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID WHERE COUNT(o.OrderID) > 10 GROUP BY c.CustomerName;
- H. SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, OrderMonth;
- I. SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerName HAVING COUNT(o.OrderID) > 10;
- J. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

---

**Answer: I**

---

**Question: 52**

You are using Microsoft SQL Server 2008 Enterprise Edition. You need to maintain a history of all data modifications made to a table, including the type of modification and the values modified.

Which tracking method should you use?

- A. Database Audit
- B. Change Data Capture
- C. Change Tracking
- D. C2 Audit Tracing

---

**Answer: B**

---

**Question: 53**

You need to build CREATE INDEX statements for all the missing indexes that SQL Server has identified. Which dynamic management view should you use?

- A. sys.dm\_db\_missing\_index\_columns
- B. sys.dm\_db\_index\_usage\_stats
- C. sys.dm\_db\_missing\_index\_group\_stats
- D. sys.dm\_db\_missing\_index\_details

---

**Answer: D**

---

**Question: 54**

You need to write a query that uses a ranking function that returns the sequential number of a row within a partition of a result set, starting at 1 for the first row in each partition. Which Transact-SQL statement should you use?

- A. ROW\_NUMBER
- B. DENSE\_RANK
- C. RANK
- D. NTILE(10)

---

**Answer: A**

---

**Question: 55**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance. You create tables named order, customer, and product as follows: CREATE TABLE [dbo].[order] ([OrderID] [int], [ProductID] [int], [CustomerID] [int], [OrderDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int], [CustomerName] [varchar](100), [Address] [varchar](200), [City] [varchar](100), [State] [varchar](50), [ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int], [ProductName] [varchar](100),

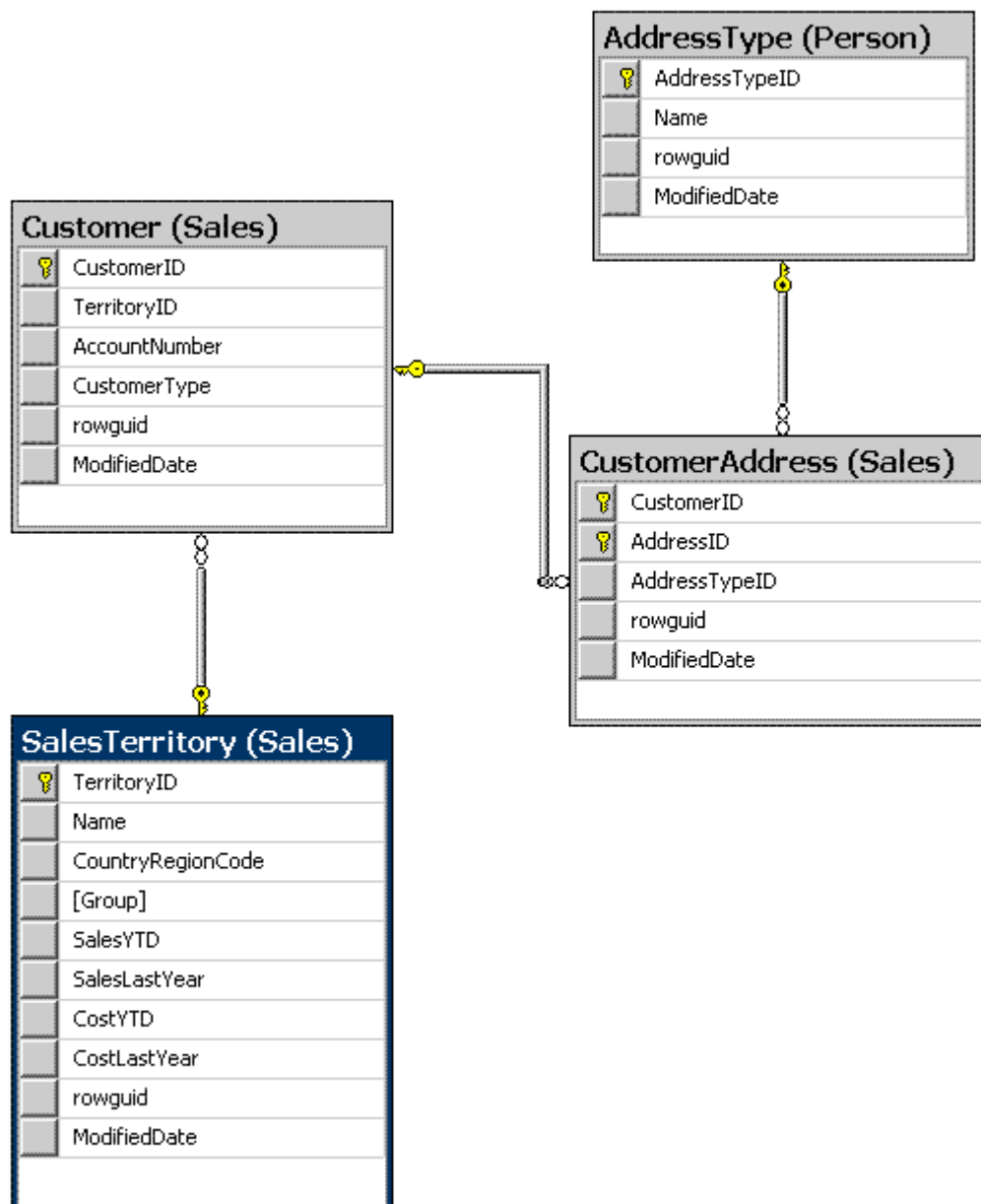
[SalePrice] [money],

[ManufacturerName] [varchar](100)); You need to write a query to identify all customers who have ordered for an average amount of more than 500 or more from September 01, 2011. Which SQL query should you use?

- A. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, DATEPART(mm, o.OrderDate);`
- B. `SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID WHERE COUNT(o.OrderID) > 10 GROUP BY c.CustomerName;`
- C. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500`
- D. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, o.OrderDate));`
- E. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE;`
- F. `SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerName HAVING COUNT(o.OrderID) > 10;`
- G. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, OrderMonth;`
- H. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());`
- I. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());`
- J. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' AND AVG(p.SalePrice) >= 500`

**Answer: C**

**Question: 56**



You administer a Microsoft SQL Server 2008R2 database that hosts a customer relationship management (CRM) application.

The application supports the following two types of customers as shown in the exhibit. (Click the Exhibit button.)

- Business customers who have shipments sent to their office locations
  - Residential customers who have shipments sent to their home address
- You need to generate a list of residential customers who live in North America. 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.)

A. DECLARE @CustomerListHomeAddress TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListHomeAddressSELECTCustomerIdFROMSales.CustomerAddressWHEREAddress

sTypeID = (SELECT AddressTypeID FROM Person.AddressType WHERE Name = 'Home')

B. DECLARE @CustomerListHomeAddress TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListHomeAddressSELECTCustomerIdFROMSales.CustomerAddressWHERE EXISTS (SELECT AddressTypeID FROM Person.AddressType WHERE Name = 'Home')

C. DECLARE @CustomerListNorthAmerica TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListNorthAmericaSELECTCustomerIdFROMSales.CustomerEXCEPT(SELECTCustomerIdFROMSales.CustomerINNER JOINSales.SalesTerritoryONCustomer.TerritoryID = SalesTerritory.TerritoryIDWHERESalesTerritory.[Group] = 'North America')

D. DECLARE @CustomerListNorthAmerica TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListNorthAmericaSELECTCustomerIdFROMSales.CustomerINTERSECT(SELECTCustomerIdFROMSales.CustomerINNER JOINSales.SalesTerritoryONCustomer.TerritoryID = SalesTerritory.TerritoryIDWHERESalesTerritory.[Group] = 'North America')

E. SELECTDISTINCT HomeAddress.CustomerIdFROM@CustomerListHomeAddress AS HomeAddressWHERECustomerId IN (SELECT CustomerID FROM @CustomerListNorthAmerica)

F. SELECTDISTINCT HomeAddress.CustomerIdFROM@CustomerListHomeAddress AS HomeAddressWHERECustomerId NOT IN (SELECT CustomerID FROM @CustomerListNorthAmerica)

**Answer: (A BEFORE D) AND (D BEFORE E) AND ONLY (A, D, E)**

**Question: 57**

You administer a Microsoft SQL Server 2008 database.

The database contains tables named Customer, Subscriptions, and Orders that have the following definitions: CREATE TABLE dbo.Customer (CustomerId int NOT NULL PRIMARY KEY,

FirstName varchar(255) NOT NULL,

LastName varchar(255) NOT NULL,

CustomerAddress varchar(1024)) CREATE TABLE dbo.Subscriptions (SubscriptionID int NOT NULL PRIMARY KEY

SubscriptionName varchar(255) NOT NULL

CustomerId int FOREIGN KEY NOT NULL REFERENCES Customer(CustomerID)) CREATE TABLE

dbo.Orders (OrderID int NOT NULL PRIMARY KEY

OrderText varchar(255) NOT NULL

CustomerId int FOREIGN KEY NOT NULL REFERENCES Customer(CustomerID)) Customers are considered active if they meet the following requirements:

- Placed an order for a subscription that is recorded in the Subscriptions table.

- Placed an order for an individual product that is recorded in the Orders table. You need to create a view that shows unique rows where the customer either has made an Order or has a Subscription. Which Transact-SQL statement should you use?

A. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT

c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cWHERE

dbo.CustomerId in (SELECT CustomerIdFROM dbo.Subscriptions sUNION ALLSELECT

CustomerIdFROM dbo.Orders o)

- B. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cLEFT OUTER  
JOIN dbo.Subscriptions sON c.CustomerID = s.CustomerIDLEFT OUTER JOIN dbo.Orders oON  
c.CustomerID = o.CustomerID
- C. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cWHERE  
CustomerId in (SELECT CustomerIdFROM dbo.Subscriptions sINTERSECTSELECT CustomerIdFROM  
dbo.Orders o)
- D. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cINNER JOIN  
dbo.Subscriptions sON c.CustomerID = s.CustomerIDEXCEPTSELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cINNER JOIN  
dbo.Orders oON c.CustomerID = o.CustomerID
- E. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cLEFT OUTER  
JOIN dbo.Subscriptions sON c.CustomerID = s.CustomerIDLEFT OUTER JOIN dbo.Orders oON  
c.CustomerID = o.CustomerIDWHERE s.CustomerID IS NOT NULLOR o.CustomerID IS NOT NULL
- F. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cINNER OUTER  
JOIN dbo.Subscriptions sON c.CustomerID = s.CustomerIDINNER OUTER JOIN dbo.Orders oON  
c.CustomerID = o.CustomerID
- G. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cINNER JOIN  
dbo.Subscriptions sON c.CustomerID = s.CustomerIDUNION ALLSELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cINNER JOIN  
dbo.Orders oON c.CustomerID = o.CustomerID
- H. CREATE VIEW dbo.vw\_ActiveCustomersAS SELECT DISTINCT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM dbo.Customer cWHERE EXISTS  
(SELECT \*FROM dbo.subscriptions sWHERE c.CustomerID = s.CustomerID)OR EXISTS (SELECT  
\*FROM dbo.orders oWHERE c.CustomerID = o.CustomerID)

---

**Answer: A OR H**

---

**Question: 58**

You are the database developer for an order-processing application. After a customer places an order, a confirmation message must be sent to the customer.

The following Transact-SQL batch has been run in the database: ALTER DATABASE NORTHWIND SET ENABLE\_BROKER; CREATE MESSAGE TYPE EmailMessageVALIDATION = NONE; CREATE CONTRACT EmailContract(EmailMessage SENT BY INITIATOR); CREATE QUEUE EmailSendQueue; CREATE QUEUE EmailReceiveQueue; CREATE SERVICE EmailSendServiceON QUEUE EmailSendQueue (EmailContract); CREATE SERVICE EmailReceiveServiceON QUEUE EmailReceiveQueue (EmailContract); You need to place the message in the EmailSendQueue for the email system to process. Which Transact-SQL batch should you use?

- A. DECLARE@EmailDialog UNIQUEIDENTIFIER,@Message NVARCHAR(128);BEGIN DIALOG



CONVERSATION @EmailDialogFROM SERVICE EmailSendServiceTO SERVICE  
'EmailReceiveService'ON CONTRACT EmailContractWITH ENCRYPTION = OFF;SET @Message =  
N'Dear Sir/Madam. Your order has been received.';SEND ON CONVERSATION  
@EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

B. DECLARE@EmailDialog UNIQUEIDENTIFIER,@Message NVARCHAR(128);BEGIN DIALOG  
@EmailDialogFROM SERVICE EmailSendServiceTO SERVICE 'EmailReceiveService'ON CONTRACT  
EmailContract;SET @Message = N'Dear Sir/Madam. Your order has been received.';SEND ON  
CONVERSATION @EmailDialog (@Message);

C. DECLARE@EmailDialog UNIQUEIDENTIFIER,@Message NVARCHAR(128);BEGIN DIALOG  
@EmailDialogFROM SERVICE EmailSendServiceTO SERVICE EmailReceiveServiceON CONTRACT  
EmailContract;SET @Message = N'Dear Sir/Madam. Your order has been received.';SEND ON  
CONVERSATION @EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

D. DECLARE@EmailDialog UNIQUEIDENTIFIER,@Message NVARCHAR(128);BEGIN DIALOG  
@EmailDialogFROM SERVICE EmailSendServiceTO SERVICE 'EmailReceiveService'ON CONTRACT  
EmailContractWITH LIFETIME = 1000;SET @Message = N'Dear Sir/Madam. Your order has been  
received.';SEND ON CONVERSATION @EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

E. DECLARE@EmailDialog BIGINT,@Message NVARCHAR(128);BEGIN DIALOG @EmailDialogTO  
SERVICE 'EmailReceiveService'ON CONTRACT EmailContract;SET @Message = N'Dear Sir/Madam.  
Your order has been received.';SEND ON CONVERSATION @EmailDialogMESSAGE TYPE  
EmailMessage (@Message);GO

F. DECLARE@EmailDialog BIGINT,@Message NVARCHAR(128);BEGIN DIALOG @EmailDialogTO  
SERVICE EmailReceiveServiceFROM SERVICE 'EmailSendService'ON CONTRACT EmailContract;SET  
@Message = N'Dear Sir/Madam. Your order has been received.';SEND ON CONVERSATION  
@EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

G. DECLARE@EmailDialog BIGINT,@Message NVARCHAR(128);BEGIN DIALOG  
@EmailDialogFROM SERVICE EmailSendServiceTO SERVICE 'EmailReceiveService'ON CONTRACT  
EmailContract;SET @Message = N'Dear Sir/Madam. Your order has been received.';SEND ON  
CONVERSATION @EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

H. DECLARE@EmailDialog BIGINT,@Message NVARCHAR(128);BEGIN DIALOG @EmailDialogTO  
SERVICE 'EmailReceiveService'FROM SERVICE EmailSendServiceON CONTRACT EmailContract;SET  
@Message = N'Dear Sir/Madam. Your order has been received.';SEND ON CONVERSATION  
@EmailDialogMESSAGE TYPE EmailMessage (@Message);GO

---

**Answer: A OR D**

---

**Question: 59**

You have a table named Sales.PotentialClients. This table contains a column named EmailAddress. You are tasked to develop a report that returns valid ".com" email addresses from Sales.PotentialClients. A valid email address must have at least one character before the @ sign, and one character after the @ sign and before the ".com." You need to write a Transact-SQL statement that returns data to meet the business requirements. Which Transact-SQL statement should you use?

- A. SELECT\* FROMSales.PotentialClientsWHEREEmailAddress LIKE'\_%@\_.com'
- B. SELECT \* FROM Sales.PotentialClientsWHERE EmailAddress LIKE '\_%@\_%.com'

- C. SELECT \* FROM Sales.PotentialClients WHERE EmailAddress LIKE '%@%.com'
- D. SELECT \* FROM Sales.PotentialClients WHERE EmailAddress LIKE '%@%[.]com'

---

**Answer: B**

---

**Question: 60**

Your server collation is SQL\_Latin1\_General\_CP1\_CI\_AS. You have a database named Contoso that has a collation setting of SQL\_Scandinavian\_Cp850\_CI\_AS.

You create and populate a temporary table #Person from table dbo.Person in Contoso using the following statements: USE Contoso; CREATE TABLE #Person (LastName nchar(128)); INSERT INTO #Person SELECT LastName FROM dbo.Person; You then run the following command: SELECT \* FROM dbo.Person a JOIN #Person b

ON a.LastName = b.LastName; This command returns the following error: Cannot resolve the collation conflict between "SQL\_Latin1\_General\_CP1\_CI\_AS" and "SQL\_Scandinavian\_Cp850\_CI\_AS" in the equal to operation.

You need to resolve the collation conflict. Which Transact-SQL statement should you use?

- A. CREATE TABLE #Person (LastName nvarchar(128) COLLATE database\_default);
- B. CREATE TABLE #Person (LastName nvarchar(128) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS);
- C. CREATE TABLE #Person (LastName nvarchar(128) SPARSE);
- D. CREATE TABLE tmpPerson (LastName nvarchar(128) COLLATE SQL\_Latin1\_General\_CP1\_CI\_AS);

---

**Answer: A**

---

**Question: 61**

You have a server named Contoso with multiple databases. You have been tasked to write a PowerShell script to determine which databases on the server are larger than 100GB.

You open PowerShell from SQL Server Management Studio. You create two variables as follows:

```
PS SQLSERVER:\SQL\Contoso> $MultipleOfGB = 1024 * 1024
PS SQLSERVER:\SQL\Contoso> $Server = Get-Item
```

You need to determine which script will produce the desired list of databases. What script should you use?

- A. \$Server | Where-Object{(\$\_.DatabaseSize \* \$MultipleOfGB) -match 100GB} | Select-Object Name, DatabaseSize
- B. \$Server.Databases | Where-Object{(\$\_.Size \* \$MultipleOfGB) -gt 100GB} | Select-Object Name, Size
- C. \$Server.Databases | Where-Object{(\$\_.Size \* \$MultipleOfGB) -match 100GB} | Select-Object Name, Size
- D. \$Server | Where-Object{(\$\_.DatabaseSize \* \$MultipleOfGB) -gt 100GB} | Select-Object Name, DatabaseSize

---

**Answer: B**

---

**Question: 62**

You are updating a database table. You need to partition the table to store only the last 1000 rows of data in the table. What should you do?

- A. Create the partition function, the partition scheme, and the distributed partitioned view.
- B. Create the partition function and the partition scheme, and update the table.
- C. Add a secondary file to the primary filegroups, update the table, and create the distributed partitioned view.
- D. Create the partition function, update the table, and create a filtered index.

---

**Answer: B**

---

**Question: 63**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK\_Product\_ProductID and a non-clustered index named AK\_Product\_ProductNumber. Both indexes have been created on a single primary partition.

The table has the following definition: `CREATE TABLE [Production].[Product](  
[ProductID] [int] IDENTITY(1,1) NOT NULL,  
[Name] [nvarchar](50) NOT NULL,  
[ProductNumber] [nvarchar](25) NOT NULL,  
[Color] [nvarchar](15) NULL,  
[Class] [nchar](2) NULL,  
[Style] [nchar](2) NULL,  
[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED  
(  
[ProductID] ASC`

`) ON [PRIMARY]) ON [PRIMARY]GO` The index has the following definition: `CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_ProductNumber] ON [Production].[Product] (  
[ProductNumber] ASC) ON [PRIMARY]GO` The Production.Product table contains 1 million rows. You want to ensure that data retrieval takes the minimum amount of time when the queries executed against the Production.Product table are ordered by product number or filtered by class. You need to refresh statistics on the Production.Product table.

Which Transact-SQL statement should you use?

- A. `ALTER INDEX AK_Product_ProductNumber ON Production.Product REORGANIZE`
- B. `SELECT * FROM sys.dm_db_index_physical_stats (DB_ID(),  
OBJECT_ID(N'Production.Product'),NULL, NULL, NULL)`
- C. `ALTER DATABASE [AdventureWorks] SET AUTO_UPDATE_STATISTICS ON`
- D. `UPDATE INDEX AK_Product_ProductNumber ON Production.Product SET  
(STATISTICS_NORECOMPUTE = ON)`
- E. `ALTER INDEX AK_Product_ProductNumber ON Production.Product REBUILD Partition = 1`
- F. `SELECT * FROM sys.dm_db_index_operational_stats (DB_ID(),  
OBJECT_ID(N'Production.Product'),NULL, NULL)`
- G. `SELECT * FROM sys.indexes where name=N'Production.Product'`
- H. `CREATE STATISTICS ProductClass_StatsON Production.Product (Name, ProductNumber,`

Class)WHERE Class <> null

I. EXEC sys.sp\_configure 'index create memory', 1

J. SELECT \* FROM STATS WHERE name='AK\_Product\_ProductNumber'

K. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD

L. UPDATE STATISTICS Production.Product

M. ALTER STATISTICS Production.Product

N. DBCC SHOW\_STATISTICS ('Production.Product', AK\_Product\_ProductNumber)

O. SELECT \* FROM SYS.STATS WHERE name='AK\_Product\_ProductNumber'

P. CREATE STATS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class is not nullWITH SAMPLE 100 PERCENT

Q. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class is not null

**Answer: L**

#### Question: 64

You have two tables named dbo.CurrentProducts and dbo.ArchiveProducts. You have the following query: SELECT ProductID, Name FROM dbo.CurrentProducts UNION ALL SELECT ProductID, Name FROM dbo.ArchiveProducts; You need to predict the list of products that the query will produce. Which list of products should the query return?

- A. Products that appear in dbo.CurrentProducts or dbo.ArchiveProducts but not in both.
- B. Products that appear in dbo.CurrentProducts or dbo.ArchiveProducts. Products that appear in both tables are listed multiple times.
- C. Products that have a matching ProductID and Name in dbo.CurrentProducts or dbo.ArchiveProducts.
- D. Products that appear in dbo.CurrentProducts or dbo.ArchiveProducts. Products that appear in both tables are listed only once.

**Answer: B**

#### Question: 65

You need to identify which products will be inserted when you execute the following code block.

BEGIN TRANSACTION

INSERT INTO Product (ProductName) VALUES ('food')

BEGIN TRANSACTION

INSERT INTO Product (ProductName) VALUES ('beverage')

COMMIT TRANSACTION ROLLBACK TRANSACTION Which products will be inserted?

- A. None
- B. Beverage
- C. Food and beverage
- D. Food

**Answer: A**

#### Question: 66

You administer a Microsoft SQL Server database named AdventureWorks2008R2. The database has a table that has the following definition: CREATE TABLE Production.Location( Name nvarchar(100) NOT NULL PRIMARY KEY, StartDate datetime2 NOT NULL

CHECK (StartDate >= '2011-01-01')) You plan to implement custom error handling for INSERT commands. The error number for a duplicate key is 2627. The error number for a NULL violation is 515. You need to ensure that an INSERT statement meets the following requirements:

- If a duplicate row is detected during insertion, no error message must be raised.
- For NULL errors, the prefix "NULL:" must be added to the message; all others errors must begin with "OTHER:".

·Return all errors as error number 50000. Which Transact-SQL statement should you use?

- A. INSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-18') IF @@error = 515RAISERROR ('NULL:' + Error\_Message(),16,1)
- B. DECLARE @msg nvarchar(2000)INSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-18') IF @@error = 515BEGINSET @msg = 'NULL:' + Error\_Message()RAISERROR (@msg,16,1)ENDEELSEBEGINSET @msg = 'OTHER:' + Error\_Message()RAISERROR (@msg,16,1)END
- C. BEGIN TRYINSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-10')END TRYBEGIN CATCHIF ERROR\_NUMBER() <> 2627BEGINDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR (@msg,16,1)ENDEND CATCH
- D. BEGIN TRYINSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-10')END TRYBEGIN CATCHDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR (@msg,16,1)END CATCH
- E. BEGIN TRYINSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-10')END TRYBEGIN CATCHIF ERROR\_NUMBER() = 2627BEGINDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR (@msg,16,1)ENDEND CATCH
- F. BEGIN TRYINSERT Production.Location (Name,StartDate)VALUES ('Paint','2001-10-10')END TRYBEGIN CATCHIF ERROR\_NUMBER() <> 2627BEGINDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR 50010 @msgENDEND CATCH
- G. BEGIN TRYINSERT Production.Location (Name,StartDate)SELECT 'Paint','2011-10-10'WHERE NOT EXISTS (SELECT \*FROM Production.Location as LWHERE L.Name = 'Paint')END TRYBEGIN CATCHDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR (@msg,16,1)END CATCH
- H. BEGIN TRYINSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-10')END TRYBEGIN CATCHIF ERROR\_NUMBER() = 2627BEGINDELETE FROM Production.LocationWHERE Name = 'Paint'INSERT Production.Location (Name,StartDate)VALUES ('Paint','2011-10-10')ENDEELSEBEGINDECLARE @msg nvarchar(2000)SET @msg = case ERROR\_NUMBER()WHEN 515 THEN 'NULL:'ELSE 'OTHER:' END + ERROR\_MESSAGE()RAISERROR (@msg,16,1)ENDEND CATCH

---

**Answer: C OR G**

---

**Question: 67**

You are the database administrator for an order management system.

The database has the following two schemas:

·The dbo schema that is used by the main data processing group  
·A Reporting schema that is used by the reporting group The application contains a product table that has the following definition: `CREATE TABLE dbo.Product( ProductID INT, ProductName VARCHAR(100), SalePrice MONEY, ManufacturerName VARCHAR(150));` The application contains an order table that has the following definition: `CREATE TABLE dbo.[Order]( OrderID INT, ProductID INT, CustomerID INT, OrderDate DATETIME2);` The application also contains a customer table that has the following definition: `CREATE TABLE dbo.Customer (CustomerID INT, CustomerName VARCHAR(100), [Address] VARCHAR(200), City VARCHAR(100), [State] VARCHAR(50), ZipCode VARCHAR(5));` You want to load a subset of data from the product and order tables in the dbo schema to a new ProductOrderHistory table in the Reporting schema. You need to ensure that the table meets the following requirements:  
·Includes rows for every product that has been manufactured by "Contoso".  
·Includes rows for every product that was ordered from March 1, 2011 to May 31, 2011. The Reporting.ProductOrderHistory columns have the following types: CustomerName VARCHAR(100)ProductName VARCHAR(100)SalePrice MONEYManufacturerName VARCHAR(150)ProductOrderDate DATETIME Which Transact-SQL query should you use?  
A. `SELECT c.CustomerName,p.ProductName,p.SalePrice,p.ManufacturerName,CAST(o.OrderDate as datetime) as ProductOrderDateINTO Reporting.ProductOrderHistoryFROMdbo.product p LEFT JOIN(dbo.[order] o INNER JOINdbo.customer c ON o.CustomerID = c.CustomerID) ON p.ProductID = o.ProductID WHEREp.ManufacturerName = 'Contoso' OR(o.OrderDate >= '20110301' ANDo.OrderDate < '20110601')`  
B. `SELECT c.CustomerName,p.ProductName,p.SalePrice,p.ManufacturerName,o.OrderDate as ProductOrderDateINTO Reporting.ProductOrderHistoryFROMdbo.product p INNER JOINdbo.[order] o ON p.ProductID = o.ProductID INNER JOINdbo.customer c ON o.CustomerID = c.CustomerID WHEREp.ManufacturerName = '%Contoso%' OR(o.OrderDate >= '20110301'AND o.OrderDate < '20110601')`  
C. `SELECT c.CustomerName,p.ProductName,p.SalePrice,p.ManufacturerName,CAST(o.OrderDate as datetime) as ProductOrderDateINTO Reporting.ProductOrderHistoryFROMdbo.Product p LEFT JOIN(dbo.[Order] o INNER JOINdbo.Customer c ON o.CustomerID = c.CustomerID) ON p.ProductID = o.ProductID WHEREp.ManufacturerName = 'Contoso' ORo.OrderDate BETWEEN '20110301' AND '20110531';`



- D. `SELECT c.CustomerName, p.ProductName, p.SalePrice, p.ManufacturerName, CAST(o.OrderDate as datetime) as ProductOrderDate INTO reporting.ProductOrderHistory FROM dbo.product p LEFT JOIN dbo.[order] o ON p.ProductID = o.ProductID LEFT JOIN dbo.customer c ON o.CustomerID = c.CustomerID WHERE p.ManufacturerName = 'Contoso' OR (o.OrderDate between '20110301' AND '20110531 23:59:59.999');`
- E. `SELECT c.CustomerName, p.ProductName, p.SalePrice, p.ManufacturerName, CAST(o.OrderDate as datetime) as ProductOrderDate INTO TABLE reporting.ProductOrderHistory FROM dbo.product p LEFT JOIN dbo.[order] o ON p.ProductID = o.ProductID LEFT JOIN dbo.customer c ON o.CustomerID = c.CustomerID WHERE p.ManufacturerName LIKE '%contoso%' OR o.OrderDate >= '20110301' AND o.OrderDate < '20110601'`
- F. `SELECT c.CustomerName, p.ProductName, p.SalePrice, p.ManufacturerName, o.OrderDate as ProductOrderDate INTO reporting.ProductOrderHistory FROM dbo.product p RIGHT JOIN dbo.[order] o ON p.ProductID = o.ProductID INNER JOIN dbo.customer c ON o.CustomerID = c.CustomerID WHERE p.ManufacturerName = 'Contoso' OR (o.OrderDate >= '20110301' AND o.OrderDate < '20110601')`
- G. `SELECT c.CustomerName, p.ProductName, p.SalePrice, p.ManufacturerName, CAST(o.OrderDate as datetime) as ProductOrderDate INTO reporting.ProductOrderHistory FROM dbo.product p CROSS JOIN dbo.[order] o ON p.productID = o.productID CROSS JOIN dbo.customer c ON o.CustomerID = c.CustomerID WHERE p.ManufacturerName = 'Contoso' OR (o.OrderDate >= '20110301' AND o.OrderDate < '20110601')`
- H. `SELECT c.CustomerName, p.ProductName, p.SalePrice, p.ManufacturerName, CAST(o.OrderDate as datetime) as ProductOrderDate INTO reporting.ProductOrderHistory FROM dbo.product p LEFT JOIN dbo.[order] o ON p.ProductID = o.ProductID LEFT JOIN dbo.customer c ON o.CustomerID = c.CustomerID WHERE p.ManufacturerName = 'Contoso' OR o.OrderDate >= '20110301' AND o.OrderDate < '20110601'`

---

**Answer: A OR D**

---

**Question: 68**

You have tables named Sales.SalesOrderDetails and Sales.SalesOrderHeader. You have been tasked to update the discount amounts for the sales of a particular salesperson.

You need to set UnitPriceDiscount to .1 for all entries in Sales.SalesOrderDetail that only correspond to SalesPersonID 290.

Which Transact-SQL statement should you use?

- A. `UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 WHERE EXISTS (SELECT * FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290);`
- B. `UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290;`
- C. `UPDATE d SET UnitPriceDiscount = .1 FROM Sales.SalesOrderDetail d INNER JOIN Sales.SalesOrderHeader h ON h.SalesOrderID = d.SalesOrderID WHERE h.SalesPersonID = 290;`
- D. `UPDATE Sales.SalesOrderDetail SET UnitPriceDiscount = .1 FROM Sales.SalesOrderDetail d WHERE EXISTS (SELECT * FROM Sales.SalesOrderHeader h WHERE h.SalesPersonID = 290);`

---

**Answer: C**

---



**Question: 69**

You have a third-party application that inserts data directly into a table. You add two new columns to the table. These columns cannot accept NULL values and cannot use default constraints. You need to ensure that the new columns do not break the third-party application. What should you do?

- A. Create an INSTEAD OF INSERT trigger.
- B. Create a stored procedure.
- C. Create a DDL trigger.
- D. Create an AFTER INSERT trigger.

---

**Answer: A**

---

**Question: 70**

You are a developer for a Microsoft SQL Server 2008 R2 database instance that hosts an application. You are designing some new tables. You need to choose the most appropriate data types. Which data types should you use? (To answer, drag the appropriate data type to the correct data in the answer area. Each data type may be used once, more than once, or not at all. Each data may be used once or not at all. Additionally, you may need to drag the split bar between panes or scroll to view content.)

1.  
XML
2.  
nvarchar(max)
3.  
varbinary(max) FILESTREAM
4.  
datetime2
5.  
timestamp  
none.gif

DataXML data validated by an XSD  
DatetimeInternational text data longer than 5,000 characters  
XML data that must be indexed and searched by using XPATH queries  
XML data that must be preserved exactly as it was received  
MP3 audio files  
The date and time a row was last modified  
HTML fragments

---

**Answer: G5 AND B4 AND F3  
AND H2 AND E2 AND C2  
AND D1 AND A1**

---

**Question: 71**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[Active] [bit] NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
[ProductID] ASC
) ON [PRIMARY]) ON [PRIMARY]GO
```

You want to add a new field to the Product table to meet the following requirements:

- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support decimal values between 0 and 10,000 that have up to six decimal points. Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)
- Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY
- R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME

---

**Answer: C**

---

**Question: 72**

You have a database that uses stored procedures to perform INSERT, UPDATE, DELETE, and SELECT statements.

You are tasked with providing a recommendation of indexes to be created and dropped from the database.

You need to select the appropriate method to accomplish the task. Which method should you use?

- A. SQL Server Profiler
- B. Database Engine Tuning Advisor
- C. Index Usage DMVs
- D. Missing Index DMVs

**Answer: B****Question: 73**

You have an application that is used by international clients. All clients connect by using Windows Authentication. You need to ensure that system and user-defined error messages are displayed in the localized language for the clients.

What should you do? (Each correct answer represents part of the solution. Choose two.)

- A. Use @@LANGUAGE function
- B. Use default language for each login
- C. Use @lang parameter of sp\_addmessage
- D. Use the "set language" option of sp\_configure

**Answer: B AND C****Question: 74**

You are a database developer working on an application hosted on Microsoft SQL Server 2008 R2. The application regularly imports employee data from XML files. These files are bulk-loaded into the CompanyDoc column of the CompanyXML table. The table the following definition: CREATE TABLE CompanyXML(CompanyXMLID int IDENTITY PRIMARY KEY NOT NULL, CompanyDoc xml NOT NULL) One of the XML files loaded into this table has the following code fragment.

```
<Company><Employee Name = "Michaeline" Department = "Finance"/><Employee Name = "Chad" Department = "Facilities"/>
```

...<Employee Name = "Shelly" Department = "Human Resources"/></Company> You need to be able to return the data as shown in the following table:

Name	Department
Michael	Finance
John	Facilities
David	Human Resources

Which Transact-SQL statements should you use?

- A. SELECT Name = Company.Employee.query('@Name','varchar(20)'), Department = Company.Employee.query('@Department','varchar(20)')FROM CompanyXML cCROSS APPLY CompanyDoc.nodes('/Company/Employee') Company(Employee)

B. DECLARE @xmlid int  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyXMLID  
 FROM CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xmlid  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 SELECT Name = Company.Employee.value('@Name','varchar(20)'),  
 Department = Company.Employee.value('@Department','varchar(20)')  
 FROM CompanyXML  
 WHERE CompanyXMLID = @xmlid  
 FETCH NEXT FROM cur\_xml INTO @xmlid  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml

C. SELECT Name = Company.Employee.value('@Name','varchar(20)'), Department =  
 Company.Employee.value('@Department','varchar(20)')  
 FROM CompanyXML  
 CROSS APPLY CompanyDoc.nodes('/Company/Employee') Company(Employee)

D. DECLARE @xml xml  
 DECLARE @results table (Name varchar(20), Department  
 varchar(20))  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyDoc FROM  
 CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xml  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 INSERT @results  
 SELECT Name = Company.Employee.query('@Name','varchar(20)'),  
 Department = Company.Employee.query('@Department','varchar(20)')  
 FROM @xml.nodes('/Company/Employee') Company(Employee)  
 FETCH NEXT FROM cur\_xml INTO @xml  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml  
 SELECT \* FROM @results

E. DECLARE @xmlid int  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyXMLID  
 FROM CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xmlid  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 SELECT Name = Company.Employee.query('@Name','varchar(20)'),  
 Department = Company.Employee.query('@Department','varchar(20)')  
 FROM CompanyXML  
 WHERE CompanyXMLID = @xmlid  
 FETCH NEXT FROM cur\_xml INTO @xmlid  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml

F. DECLARE @xml xml  
 DECLARE @results table (Name varchar(20), Department  
 varchar(20))  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyDoc FROM  
 CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xml  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 INSERT @results  
 SELECT Name = @xml.value('/Company[1]/Employee[1]/@Name',  
 'varchar(20)'), Department = @xml.value('/Company[1]/Employee[1]/@Department',  
 'varchar(20)')  
 FETCH NEXT FROM cur\_xml INTO @xml  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml  
 SELECT \* FROM @results

G. DECLARE @xml xml  
 DECLARE @results table (Name varchar(20), Department  
 varchar(20))  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyDoc FROM  
 CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xml  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 INSERT @results  
 SELECT Name = @xml.value('/Company/Employee/@Name',  
 'varchar(20)'), Department = @xml.value('/Company/Employee/@Department', 'varchar(20)')  
 FETCH NEXT FROM cur\_xml INTO @xml  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml  
 SELECT \* FROM @results

H. DECLARE @xml xml  
 DECLARE @results table (Name varchar(20), Department  
 varchar(20))  
 DECLARE cur\_xml CURSOR FAST\_FORWARD FORSELECT CompanyDoc FROM  
 CompanyXML  
 OPEN cur\_xml  
 FETCH NEXT FROM cur\_xml INTO @xml  
 WHILE @@FETCH\_STATUS = 0  
 BEGIN  
 INSERT @results  
 SELECT Name = Company.Employee.value('@Name','varchar(20)'),  
 Department = Company.Employee.value('@Department','varchar(20)')  
 FROM @xml.nodes('/Company/Employee') Company(Employee)  
 FETCH NEXT FROM cur\_xml INTO @xml  
 END  
 CLOSE cur\_xml  
 DEALLOCATE cur\_xml  
 SELECT \* FROM @results

**Answer: C OR H**

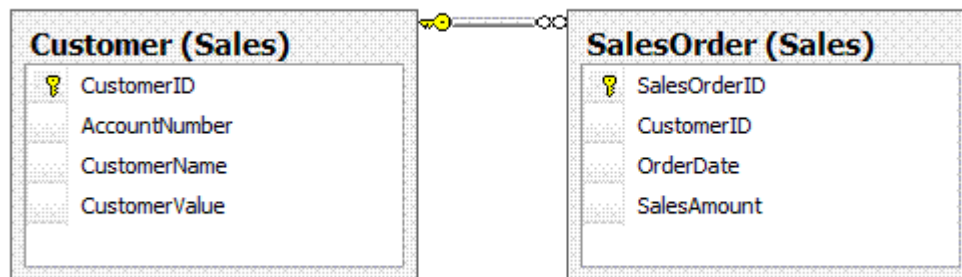
**Question: 75**

You have implemented change tracking on a table named Sales.SalesOrder. You need to determine all columns that have changed since the minimum valid version. Which function should you use?

- A. CHANGETABLE with the VERSION argument
- B. CHANGETABLE with the CHANGES argument
- C. CHANGE\_TRACKING\_CURRENT\_VERSION
- D. CHANGE\_TRACKING\_IS\_COLUMN\_IN\_MASK

**Answer: B**

**Question: 76**



You administer a Microsoft SQL Server 2008 database that contains tables named Sales.Customer and Sales.SalesOrder.

A diagram of the tables is shown in the exhibit. (Click the Exhibit button.) You need to execute a query to update the value of the CustomerValue field to HV when a customer has more than 5 orders for a total sales amount of more than 5,000 U.S. dollars. Which Transact-SQL statement should you use?

- A. UPDATE Sales.Customer SETCustomerValue = 'HV'FROMSales.CustomerWHERESalesAmount > 5000AND CustomerID IN(SELECTc.CustomerIDFROMSales.Customer cINNER JOINSales.SalesOrder o ON o.CustomerID=c.CustomerIDGROUP BYc.CustomerIDHAVINGCOUNT(\*) > 5)
- B. UPDATE Sales.Customer SETCustomerValue = 'HV'FROMSales.CustomerWHERECustomerID IN(SELECTc.CustomerIDFROMSales.Customer cINNER JOINSales.SalesOrder o ON o.CustomerID=c.CustomerIDGROUP BYc.CustomerIDHAVINGCOUNT(\*) > 5AND SUM(SalesAmount) > 5000)
- C. UPDATE c SETCustomerValue = 'HV'FROMSales.SalesOrder oINNER JOINSales.Customer c ON c.CustomerID=o.CustomerIDGROUP BYc.CustomerIDHAVINGCOUNT(\*) > 5AND SUM(SalesAmount) > 5000
- D. UPDATE c SETCustomerValue = 'HV'FROMSales.SalesOrder oINNER JOINSales.Customer c ON c.CustomerID=o.CustomerIDWHERESalesAmount > 5000GROUP BYc.CustomerIDHAVINGCOUNT(\*) > 5
- E. UPDATE u SETCustomerValue = 'HV'FROMSales.Customer

uWHERE EXISTS(SELECT c.CustomerID FROM Sales.Customer c INNER JOIN Sales.SalesOrder o ON  
o.CustomerID = c.CustomerID WHERE c.CustomerID = u.CustomerID GROUP  
BY c.CustomerID HAVING COUNT(\*) > 5 AND SUM(SalesAmount) > 5000)  
F. UPDATE u SET CustomerValue = 'HV' FROM Sales.Customer u WHERE SalesAmount > 5000 AND  
EXISTS(SELECT c.CustomerID FROM Sales.Customer c INNER JOIN Sales.SalesOrder o ON  
o.CustomerID = c.CustomerID WHERE c.CustomerID = u.CustomerID GROUP  
BY c.CustomerID HAVING COUNT(\*) > 5)  
G. UPDATE Sales.Customer SET CustomerValue = 'HV' FROM Sales.SalesOrder o INNER  
JOIN Sales.Customer c ON c.CustomerID = o.CustomerID GROUP  
BY c.CustomerID HAVING COUNT(\*) > 5 AND SUM(SalesAmount) > 5000  
H. UPDATE Sales.Customer SET CustomerValue = 'HV' FROM Sales.SalesOrder o INNER  
JOIN Sales.Customer c ON c.CustomerID = o.CustomerID WHERE SalesAmount > 5000 GROUP  
BY c.CustomerID HAVING COUNT(\*) > 5

---

**Answer: B OR E**

---

**Question: 77**

© 2012 Microsoft Corporation. All rights reserved. Welcome! Welcome to this Microsoft Certification exam. The following information is provided to help you use your exam time most efficiently.

**Exam Format and Question Formats** This exam session includes multiple sections. You will see introductory sections, the exam content, and post-exam sections. This exam might contain several question formats. The Instructions button located at the bottom of each question screen provides information on the question format shown.

**Timing** The amount of time that you have to complete this exam was provided on the Welcome Screen.

The amount of time you have remaining in each section is provided in the upper-right corner of each screen. Any delays, such as the loading of the next screen, will not be counted against your exam time.

**Number of Questions** The total number of questions on this exam is provided on the "Get Ready to Take the Exam!" screen. The number of questions remaining in each section is provided in the upper-left corner of each question.

**Reviewing/Commenting** The Mark for review or comment check box located in the upper-left corner of each question screen can be used to flag a question that you want to review. At the end of this exam, you will see a screen that shows you which questions you have marked for review. The Mark for review or comment check box can also be used to flag a question for which you want to provide a comment. Note that you cannot make comments during the exam. The comment period is timed separately; any time you spend making comments does not affect the time limit for the exam itself.

**Important Changes to Your Exam Experience** **Scrolling** You will no longer be notified that you have not viewed the entire question when you move to another question. In other words, you will not be prompted to scroll to the end of a question before you move to the next one. **Keyboard Shortcuts** Functionality, such as keyboard shortcuts, can be used during the exam. Additional information about the keyboard controls that are available within the exam can be found by

clicking the Help button associated with each question. Case Studies This exam might contain case studies. We no longer time each case study separately. In other words, the amount of time you are given to complete this exam can be used at your discretion. You can spend as much time as you like on any given case or question, but be sure to manage your time accordingly.

**Question: 78**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application. You create tables named complaint, customer, and product as follows: CREATE TABLE

```
[dbo].[complaint] ([ComplaintID] [int],
```

```
[ProductID] [int],
```

```
[CustomerID] [int],
```

```
[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],
```

```
[CustomerName] [varchar](100),
```

```
[Address] [varchar](200),
```

```
[City] [varchar](100),
```

```
[State] [varchar](50),
```

```
[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],
```

```
[ProductName] [varchar](100),
```

```
[SalePrice] [money],
```

```
[ManufacturerName] [varchar](100)); You need to write a query to sum the sales made to each customer who has made a complaint by the following entries:
```

- Each customer name

- Each product name

- The grand total of all sales Which SQL query should you use?

A. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

B. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));

C. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500

D. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;

E. SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;



F. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);

G. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500

H. SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) > 10 GROUP BY c.CustomerName;

I. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;

J. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());

**Answer: J****Question: 79**

You have a table named Employee. You document your company's organizational hierarchy by inserting the EmployeeID of each employee's manager in the ReportsTo column. You need to write a recursive query that produces a list of employees and their manager. The query must also include the employee's level in the hierarchy. You write the following code segment. (Line numbers are included for reference only.) 01 WITH EmployeeList (EmployeeID, FullName, ManagerName, Level) 02 AS ( 03 04 ) 05 SELECT EmployeeID, FullName, ManagerName, Level 06 FROM EmployeeList;

Which code segment should you insert at line 3?

A. SELECT EmployeeID, FullName, " AS [Reports To], 1 AS [Level] FROM Employee UNION ALL SELECT emp.EmployeeID, emp.FullName, mgr.FullName, 1 + 1 AS [Level] FROM Employee emp LEFT JOIN Employee mgr ON emp.ReportsTo = mgr.EmployeeID

B. SELECT EmployeeID, FullName, " AS [Reports To], 1 AS [Level] FROM Employee WHERE ReportsTo IS NULL UNION ALL SELECT emp.EmployeeID, emp.FullName, mgr.FullName, mgr.Level + 1 FROM EmployeeList mgr JOIN Employee emp ON emp.ReportsTo = mgr.EmployeeID

C. SELECT EmployeeID, FullName, " AS [Reports To], 1 AS [Level] FROM Employee UNION ALL SELECT emp.EmployeeID, emp.FullName, mgr.FullName, mgr.Level + 1 FROM EmployeeList mgr JOIN Employee emp ON emp.ReportsTo = mgr.EmployeeID

D. SELECT EmployeeID, FullName, " AS [Reports To], 1 AS [Level] FROM Employee WHERE ReportsTo IS NULL UNION ALL SELECT emp.EmployeeID, emp.FullName, mgr.FullName, 1 + 1 AS [Level] FROM Employee emp JOIN Employee mgr ON emp.ReportsTo = mgr.EmployeeID

**Answer: B****Question: 80**

You are a developer for a Microsoft SQL Server 2008 R2 database instance. You plan to add

functionality to an existing application that uses the Microsoft SQL Server Profiler tool to capture trace information. You need to ensure that the following requirements are met:

- Users are able to use the new application functionality.
- Users are able to capture trace information and read saved trace files.
- Users are granted the minimum permissions to achieve these goals. Which three actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

- A. Create a database role named traceusers.
- B. Add each user of the application to the sysadmin fixed-server role.
- C. Add each user of the application to the traceusers database role.
- D. Grant the ALTER TRACE permissions to the traceusers database role.
- E. Grant the CONNECT permission to the traceusers database role.
- F. Create a server role named traceusers.

**Answer: (C BEFORE D) AND  
(A BEFORE C) AND ONLY (A,  
C, D)**

#### Question: 81

You are tasked with creating a workload that will be used by the Database Engine Tuning Advisor (DTA). You need to create a workload in an appropriate format. Which format should you choose? (Each correct answer represents a complete solution. Choose three.)

- A. XML File
- B. Transact-SQL Script
- C. SQL Server Event Log
- D. SQL Server Transaction Log
- E. SQL Server Profiler Trace File
- F. Performance Counter Log File

**Answer: A AND B AND E**

#### Question: 82

You administer a SQL Server 2008 instance. The instance contains a database table named Sales.SalesOrderDetail.

The table has the following definition: CREATE TABLE [Sales].[SalesOrderDetail](  
[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] NOT NULL,  
[UnitPrice] [money] NOT NULL,  
[UnitPriceDiscount] [money] NOT NULL,  
[LineTotal] AS (isnull([UnitPrice]\*((1.0)-[UnitPriceDiscount]))\*[OrderQty],(0.0))),  
[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,

[ModifiedDate] [datetime] NOT NULL,  
CONSTRAINT [PK\_SalesOrderDetail\_SalesOrderID\_SalesOrderDetailID] PRIMARY KEY CLUSTERED (  
[SalesOrderID] ASC,  
[SalesOrderDetailID] ASC)WITH (DATA\_COMPRESSION = ROW) ON [PRIMARY]) ON [PRIMARY]GO  
The table includes the following index: CREATE NONCLUSTERED INDEX  
[IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail] (  
[ProductID] ASC) ON [PRIMARY]GO You need to enable row compression for the index. Which  
Transact-SQL statement or statements should you use?

A. ALTER INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]REBUILD WITH  
(FILLFACTOR = 50)GO  
B. ALTER INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]REBUILD WITH  
(DATA\_COMPRESSION = ROW) GO  
C. ALTER INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]REORGANIZE WITH  
(DATA\_COMPRESSION = ROW) GO  
D. ALTER INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]REORGANIZE WITH  
(DATA\_COMPRESSION = ON) GO  
E. ALTER INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]REBUILD WITH  
(ALLOW\_ROW\_LOCKS = ON, MAXDOP = 50)GO  
F. CREATE NONCLUSTERED INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]  
([ProductID] ASC)WITH (DATA\_COMPRESSION = ROW, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING =  
ON) ON [PRIMARY]GO  
G. CREATE NONCLUSTERED INDEX [IX\_SalesOrderDetail\_ProductID] ON [Sales].[SalesOrderDetail]  
([ProductID] ASC)WITH (DATA\_COMPRESSION = ROW, SORT\_IN\_TEMPDB = ON, DROP\_EXISTING =  
OFF) ON [PRIMARY]GO  
H. SELECT 50 AS fill\_factorFROM sys.dm\_db\_index\_operational\_stats (DB\_ID(),  
OBJECT\_ID(N'Sales.SalesOrderDetail'), NULL, NULL) AS a JOIN sys.indexes AS b ON a.object\_id =  
b.object\_id AND a.index\_id = b.index\_idWHERE name='IX\_SalesOrderDetail\_ProductID'GO

---

**Answer: B OR F**

---

**Question: 83**

You need to write a query that allows you to rank total sales for each salesperson into four groups, where the top 25 percent of results are in group 1, the next 25 percent are in group 2, the next 25 percent are in group 3, and the lowest 25 percent are in group 4. Which Transact-SQL statement should you use?

- A. NTILE(100)
- B. NTILE(25)
- C. NTILE(4)
- D. NTILE(1)

---

**Answer: C**

---

**Question: 84**

You administer a Microsoft SQL Server 2008 database that includes a table named Products.  
The Products table has the following schema: CREATE TABLE dbo.Products(

```
ProductID nchar(4) NOT NULL,  
ProductName nvarchar(50) NULL,  
ProductDescription nvarchar(50) NULL,  
UnitCost money NULL,  
UnitPrice money NULL,  
CONSTRAINT PK_Products PRIMARY KEY CLUSTERED  
(
```

```
ProductID ASC
```

)) GO You create a User-Defined Function (UDF) in the same database.

The UDF has the following schema: CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyWITH SCHEMABINDINGASBEGIN

```
DECLARE @Profit Money;
```

```
SELECT @Profit = UnitPrice - UnitCost FROM dbo.Products WHERE ProductID = @ProductID;
```

```
RETURN @Profit;END
```

You need to meet the following requirements:

- Ensure that the UnitPrice column does not accept NULL values.
- Use the CalculateProductProfit() UDF.
- Avoid accidental drop or change of the Product table. 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.)

A. DROP TABLE Products

B. DROP FUNCTION CalculateProductProfit

C. ALTER TABLE Products ALTER COLUMN UnitPrice Money NULL

D. ALTER TABLE Products ALTER COLUMN UnitPrice Money NOT NULL

E. CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyWITH SCHEMABINDINGASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM dbo.Products WHERE ProductID = @ProductID;RETURN @Profit;END

F. CREATE FUNCTION dbo.CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM dbo.Products WHERE ProductID = @ProductID;RETURN @Profit;END

G. CREATE FUNCTION CalculateProductProfit(@ProductID nchar(4))RETURNS MoneyWITH SCHEMABINDINGASBEGINDECLARE @Profit Money;SELECT @Profit = UnitPrice - UnitCost FROM Products WHERE ProductID = @ProductID;RETURN @Profit;END

**Answer: (B BEFORE D) AND  
(D BEFORE E) AND ONLY (B,  
D, E)**

#### Question: 85

You are required to modify a table named Sales.SalesOrder. The table has change tracking enabled on it. You need to disable change tracking prior to modifying the Sales.SalesOrder table. Which Transact-SQL statement should you use?

A. EXEC sys.sp\_cdc\_disable\_db

B. ALTER TABLE Sales.SalesOrderDISABLE CHANGE\_TRACKING

C. ALTER DATABASE ContosoSET CHANGE\_TRACKING = OFF

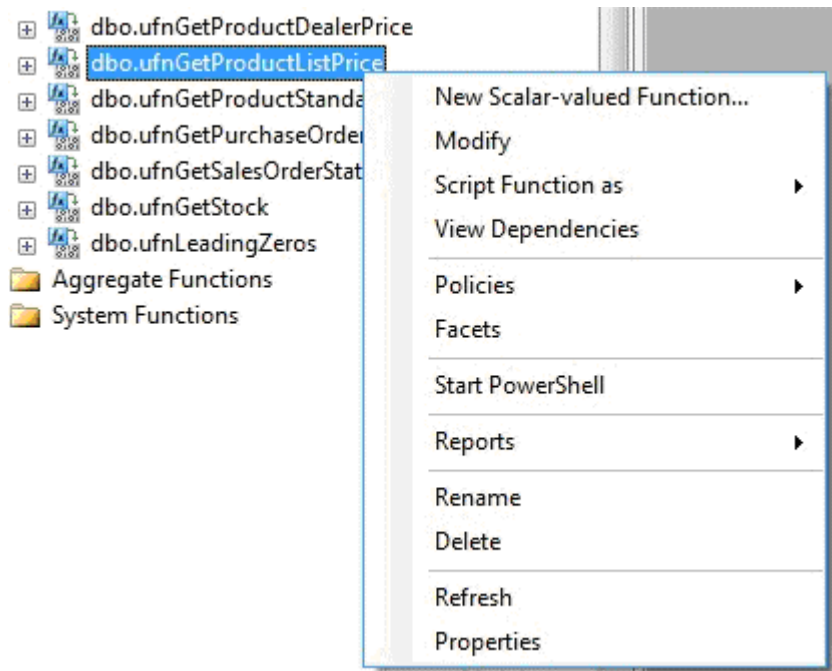
D. EXEC sys.sp\_cdc\_disable\_table@source\_schema = N'Sales',@source\_name = N'SalesOrder',@capture\_instance = N'Sales\_SalesOrder'

**Answer: B**

**Question: 86**

You want to delete a User-Defined Function (UDF) named ufnGetProductListPrice from the AdventureWorks2008R2 database. You need to perform the following tasks before executing DROP FUNCTION on ufnGetProductListPrice:

- View a list of objects that depend on ufnGetProductListPrice.
- View a list of objects on which ufnGetProductListPrice depends. What should you do? (To answer, select the appropriate option or options in the answer area.)



Work Area

**Answer: D**

**Question: 87**

You are a database developer for your organization. You create an application that uses a Transact-SQL variable to store user input data. The database collation is case-insensitive. The variable is constructed as shown in the following statement: DECLARE @content varchar(64) SELECT @content = 'The Advanced Research Projects Agency Network (ARPANET), was the worlds first operational packet switching network and the core network of a set that came to compose the global Internet.' You need to implement a keyword search that meets the following requirements:

- Searches for the existence of the word ARPANET within the user-entered content.
- If the search term is found, the statement must return its starting position, and 0 otherwise.
- Performs a case-sensitive search for the given search term. Which Transact-SQL statement should you use?

- A. SELECT CHARINDEX(UPPER'arpanet', @content, 1)GO
- B. SELECT CHARINDEX('ARPANET', @content COLLATE Latin1\_General\_CS\_AS)GO
- C. SELECT CHARINDEX('arpanet', @content)GO
- D. SELECT CHARINDEX('ARPANET', @content, COLLATE Latin1\_General\_CS\_AS)GO
- E. SELECT CHARINDEX(@content, 'ARPANET', COLLATE Latin1\_General\_CS\_AS)GO
- F. SELECT CHARINDEX('ARPANET', @content COLLATE Latin1\_General\_CS\_AS, 0)GO
- G. SELECT CHARINDEX(@content COLLATE Latin1\_General\_CS\_AS, 'ARPANET', 0)GO
- H. SELECT CHARINDEX(@content, 'ARPANET')GO

---

**Answer: B OR F**

---

**Question: 88**

You create a database named AdventureWorks. You want to create a new table to store customer reviews for all products within the database. The table must meet the following requirements:

- Stores information for the Product ID, Customer ID, Rating, and Reviews columns.
- The Reviews column can contain NULL values.
- The Reviews column is optimized to store NULL values. 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.)

- A. USE AdventureWorksGO
- B. ALTER TABLE ProductReviewsALTER COLUMN Review NOT NULL ADD SPARSE;GO
- C. ALTER TABLE ProductReviewsALTER COLUMN Review ADD SPARSE ;GO
- D. CREATE TABLE ProductReviews(CustomerID nchar(6) NOT NULL,ProductID nchar(6) NOT NULL,Rating smallint NOT NULL,Review varchar(500) NOT NULL,CONSTRAINT PK\_ProductReviews PRIMARY KEY CLUSTERED(CustomerID ASC,ProductID ASC))GO
- E. CREATE TABLE ProductReviews(CustomerID nchar(6) NOT NULL,ProductID nchar(6) NOT NULL,Rating smallint NULL,Review varchar(500) NULL,CONSTRAINT PK\_ProductReviews PRIMARY KEY CLUSTERED(CustomerID ASC,ProductID ASC))GO
- F. CREATE TABLE ProductReviews(CustomerID nchar(6) NOT NULL,ProductID nchar(6) NOT NULL,Rating smallint NOT NULL,Review varchar(500) SPARSE NOT NULL,CONSTRAINT PK\_ProductReviews PRIMARY KEY CLUSTERED(CustomerID ASC,ProductID ASC))GO

---

**Answer: (A BEFORE E) AND  
(E BEFORE C) AND ONLY (A,  
E, C)**

---

**Question: 89**

You have a transaction that uses the repeatable read isolation level. This transaction causes frequent blocking problems. You need to reduce blocking. You also need to avoid dirty reads and non-repeatable reads. Which transaction isolation level should you use?

- A. SNAPSHOT
- B. SERIALIZABLE
- C. READ UNCOMMITTED
- D. READ COMMITTED

---

**Answer: A**

---

**Question: 90**

You have a table named Orders. OrderID is defined as an IDENTITY(1,1). OrderDate has a default value of 1. You need to write a query to insert a new order into the Orders table for CustomerID 45 with today's date and a cost of 89.00. Which statement should you use?

- A. INSERT INTO Orders(OrderID, CustomerId, OrderDate, Cost)VALUES (1, 45, DEFAULT, 89.00);
- B. INSERT INTO Orders(OrderID, CustomerId, OrderDate, Cost)VALUES (1, 45, CURRENT\_TIMESTAMP, 89.00);
- C. INSERT INTO Orders(CustomerId, OrderDate, Cost)VALUES (45, CURRENT\_TIMESTAMP, 89.00);
- D. INSERT INTO Orders(CustomerId, OrderDate, Cost)VALUES (45, DEFAULT, 89.00);

---

**Answer: C**

---

**Question: 91**

A table named Shirts includes an XML column named SizeAndColors that contains the sizes and colors of each shirt, as shown in the following code segment. <shirt size="XS" colors="Red" /><shirt size="S" colors="Red, Blue" />...<shirt size="XL" colors="Blue, Green" /> You need to return a list of shirt colors available in a specific size. Which code segment should you use?

- A. DECLARE @Size VARCHAR(10) = 'L' SELECT ShirtStyleName,SizeAndColors.value('/shirt[1]/@colors','color')FROM ShirtsWHERE SizeAndColors.exist('/shirt[@size=sql:variable("@Size")]') = 1
- B. DECLARE @Size VARCHAR(10) = 'L' SELECT ShirtStyleName,SizeAndColors.query('data(/shirt[@size=sql:variable("@Size")]/@colors)')FROM Shirts
- C. DECLARE @Size VARCHAR(10) = 'L' SELECT ShirtStyleName,SizeAndColors.value('/shirt[1] [@size =sql:variable("@Size")]/@ colors, 'color')FROM Shirts
- D. DECLARE @Size VARCHAR(10) = 'L' SELECT ShirtStyleName,SizeAndColors.value('/shirt[1]/@colors','color')FROM ShirtsWHERE SizeAndColors.value('/shirt[1]/@size','varchar(20)') = @Size

---

**Answer: B**

---

**Question: 92**

You have the following XML document that contains Product information. DECLARE @prodList xml ='<ProductList xmlns="urn:Wide\_World\_Importers/schemas/Products">

<Product Name="Product1" Category="Food" Price="12.3" />

<Product Name="Product2" Category="Drink" Price="1.2" />

<Product Name="Product3" Category="Food" Price="5.1" />

...</ProductList>'; You need to return a list of products that contains the Product Name, Category, and Price of each product. Which query should you use?

- A. SELECT prod.value('.[1]/@Name','varchar(100)'),prod.value('.[1]/@Category','varchar(20)'),prod.value('.[1]/@Price','money')FROM @prodList.nodes('/ProductList/Product') ProdList(prod);



B. SELECT

```
prod.value('@Name','varchar(100)'),prod.value('@Category','varchar(20)'),prod.value('@Price','money')FROM @prodList.nodes('/ProductList/Product') ProdList(prod);
```

C. WITH XMLNAMESPACES(DEFAULT 'urn:Wide\_World\_Importers/schemas/Products')SELECT prod.value('./@Name','varchar(100)'),prod.value('./@Category','varchar(20)'),prod.value('./@Price','money')FROM @prodList.nodes('/ProductList/Product') ProdList(prod);

D. WITH XMLNAMESPACES(DEFAULT 'urn:Wide\_World\_Importers/schemas/Products' as o)SELECT prod.value('Name[1]','varchar(100)'),prod.value('Category[1]','varchar(20)'),prod.value('Price[1]','money')FROM @prodList.nodes('/o:ProductList/o:Product') ProdList(prod);

---

**Answer: C**

---

**Question: 93**

You need to capture and record a workload for analysis by the Database Engine Tuning Advisor (DTA). Which tool should you use?

- A. Performance Monitor
- B. SQL Server Profiler
- C. DTA utility
- D. Activity Monitor

---

**Answer: B**

---

**Question: 94**

You have a table named Subscribers. You receive an updated list of subscribers. You need to remove subscribers that are no longer on the list. Which clause should you use to remove the subscribers from the table?

- A. WHEN NOT MATCHED BY TARGET
- B. WHEN MATCHED
- C. WHEN NOT MATCHED BY SOURCE
- D. WHEN NOT MATCHED

---

**Answer: C**

---

**Question: 95**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application. You create tables named complaint, customer, and product as follows: CREATE TABLE

```
[dbo].[complaint] ([ComplaintID] [int],  
[ProductID] [int],  
[CustomerID] [int],  
[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],  
[CustomerName] [varchar](100),  
[Address] [varchar](200),  
[City] [varchar](100),
```

[State] [varchar](50),  
[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],  
[ProductName] [varchar](100),  
[SalePrice] [money],  
[ManufacturerName] [varchar](100)); You need to write a query to sum the sales of all products that have complaints by the following entries:

- The product name
- The month the product had a complaint
- The product name and the month the product had a complaint
- The grand total of all sales Which SQL query should you use?

A. SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) > 10 GROUP BY c.CustomerName;

B. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);

C. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;

D. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());

E. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

F. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500

G. SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;

H. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500

I. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));

J. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;

---

**Answer: I**

---

**Question: 96**

Your company's database contains Customers and Orders tables. You have been tasked to write a SELECT statement that outputs customer and order data as a valid and well-formed XML document. You are required to mix attribute and element based XML within the document. You have determined that using the FOR XML AUTO clause will not be suitable. You need to identify the correct FOR XML clause to meet the requirement. Which FOR XML statement should you use? (Each correct answer represents a complete solution. Choose two.)

- A. FOR BROWSE
- B. FOR XML RAW
- C. FOR XML PATH
- D. FOR XML EXPLICIT

---

**Answer: C AND D**

---

**Question: 97**

You have two tables named Customer and SalesOrder. You need to identify all customers that have not yet made any purchases and those that have only made orders with an OrderTotal less than 100. Which query should you use?

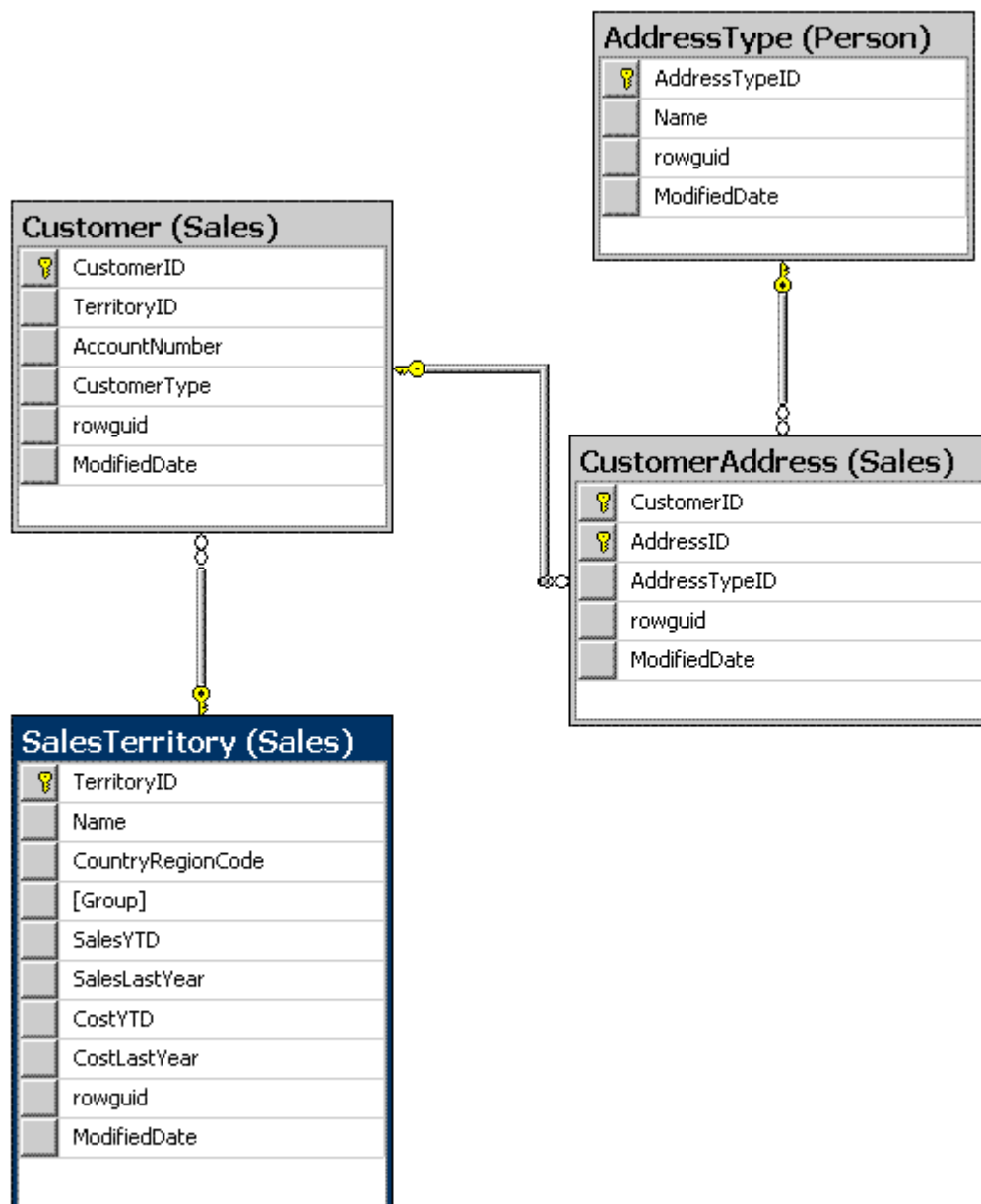
- A. SELECT \*FROM CustomerWHERE 100 > SOME (SELECT OrderTotalFROM SalesOrderWHERE Customer.CustomerID = SalesOrder.CustomerID)
- B. SELECT \*FROM CustomerWHERE EXISTS (SELECT SalesOrder.CustomerIDFROM SalesOrderWHERE Customer.CustomerID = SalesOrder.CustomerIDAND SalesOrder.OrderTotal <= 100)
- C. SELECT \*FROM CustomerWHERE 100 > ALL (SELECT OrderTotalFROM SalesOrderWHERE Customer.CustomerID = SalesOrder.CustomerID)
- D. SELECT \*FROM CustomerWHERE 100 > (SELECT MAX(OrderTotal)FROM SalesOrderWHERE Customer.CustomerID = SalesOrder.CustomerID)

---

**Answer: C**

---

**Question: 98**



You administer a Microsoft SQL Server 2008R2 database that hosts a customer relationship management (CRM) application.

The application supports the following two types of customers as shown in the exhibit. (Click the Exhibit button.)

- Business customers who have shipments sent to their office locations
  - Residential customers who have shipments sent to their home address
- You need to generate a list of residential customers who live outside North America. 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.)

A. DECLARE @CustomerListHomeAddress TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListHomeAddressSELECTCustomerIdFROMSales.CustomerAddressWHEREAddress

sTypeID = (SELECT AddressTypeID FROM Person.AddressType WHERE Name = 'Home')

B. DECLARE @CustomerListHomeAddress TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListHomeAddressSELECTCustomerIdFROMSales.CustomerAddressWHEREEXISTS (SELECT AddressTypeID FROM Person.AddressType WHERE Name = 'Home')

C. DECLARE @CustomerListNorthAmerica TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListNorthAmericaSELECTCustomerIdFROMSales.CustomerEXCEPT(SELECTCusto merIDFROMSales.CustomerINNER JOINSales.SalesTerritoryONCustomer.TerritoryID = SalesTerritory.TerritoryID)

D. DECLARE @CustomerListNorthAmerica TABLE (CustomerId INT NOT NULL)INSERT INTO@CustomerListNorthAmericaSELECTCustomerIdFROMSales.CustomerINTERSECT(SELECTCus tomerIDFROMSales.CustomerINNER JOINSales.SalesTerritoryONCustomer.TerritoryID = SalesTerritory.TerritoryIDWHERESalesTerritory.[Group] = 'North America')

E. SELECTDISTINCT HomeAddress.CustomerIdFROM@CustomerListHomeAddress AS HomeAddressWHERECustomerId IN (SELECT CustomerID FROM @CustomerListNorthAmerica)

F. SELECTDISTINCT HomeAddress.CustomerIdFROM@CustomerListHomeAddress AS HomeAddressWHERECustomerId NOT IN (SELECT CustomerID FROM @CustomerListNorthAmerica)

**Answer: (A BEFORE D) AND (D BEFORE F) AND ONLY (A, D, F)**

**Question: 99**

You are tasked to analyze blocking behavior of the following query: SET TRANSACTION ISOLATION LEVEL SERIALIZABLEWITH Customers AS ( SELECT \*

FROM Customer ),SalesTotal AS ( SELECT CustomerId, SUM(OrderTotal) AS AllOrderTotal FROM SalesOrder)SELECT CustomerId, AllOrderTotalFROM

SalesTotalWHERE AllOrderTotal > 10000.00; You need to determine if other queries that are using the Customer table will be blocked by this query. You also need to determine if this query will be blocked by other queries that are using the Customer table. What behavior should you expect?

- A. The other queries will not be blocked by this query.This query will not be blocked by the other queries.
- B. The other queries will be blocked by this query.This query will not be blocked by the other queries.
- C. The other queries will not be blocked by this query.This query will be blocked by the other queries.
- D. The other queries will be blocked by this query.This query will be blocked by the other queries.

**Answer: A**

**Question: 100**

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 PreferredCustomerList.  
Data will be added to the PreferredCustomerList table regularly.  
The PreferredCustomerList table has the following definition: CREATE TABLE PreferredCustomerList  
(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 also present in the PreferredCustomerList table. Which Transact-SQL statement should you use?

A. CREATE VIEW vw\_ValidCustomerASSELECT FirstName,LastNameFROM Customer cEXCEPTSELECT FirstName,LastNameFROM PreferredCustomerList

B. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

C. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cLEFT OUTER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastNameWHERE cel.FirstName IS NULL

D. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameINNER JOIN PreferredCustomerList celON c.LastName = cel.LastName

E. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cEXCEPTSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

F. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINTERSECTSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

G. CREATE VIEW vw\_ValidCustomerASSELECT  
c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cLEFT OUTER JOIN PreferredCustomerList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastNameWHERE cel.LastName IS NULL

H. CREATE VIEW vw\_ValidCustomerASSELECT  
CustomerID,FirstName,LastName,CustomerAddressFROM Customer cEXCEPTSELECT  
CustomerID,FirstName,LastName,CustomerAddressFROM PreferredCustomerList

**Answer: B OR F****Question: 101**

You have a database server that has four quad-core processors. This database server executes

complex queries that are used to generate reports. You need to force a query to use only one processor core without affecting other queries. Which option should you use?

- A. OPTION (MAXDOP 4)
- B. OPTION (MAXDOP 0)
- C. OPTION (MAXDOP 16)
- D. OPTION (MAXDOP 1)

---

**Answer: D**

---

**Question: 102**

You need to implement a common table expression (CTE). Which code segment should you use?

- A. SELECT Year,Region,Total FROM (SELECT Year,Region,SUM(OrderTotal) AS TotalFROM OrdersGROUP BY Year, Region) AS [SalesByYear];
- B. CREATE VIEW SalesByYearASSELECT Year,Region,SUM(OrderTotal)FROM OrdersGROUP BY Year, Region; GO SELECT Year,Region,TotalFROM SalesByYear;
- C. WITH SalesByYear(Year,Region,Total)AS (SELECT Year,Region,SUM(OrderTotal)FROM OrdersGROUP BY Year,Region) SELECT Year,Region,TotalFROM SalesByYear;
- D. SELECT DISTINCT Year,Region,(SELECT SUM(OrderTotal)FROM Orders SalesByYearWHERE Orders.Year = SalesByYear.YEARAND Orders.Region = SalesByYear.Region) AS [Total]FROM Orders;

---

**Answer: C**

---

**Question: 103**

You have a single CLR assembly in your database. The assembly only references blessed assemblies from the Microsoft .NET Framework and does not access external resources. You need to deploy this assembly by using the minimum required permissions. You must ensure that your database remains as secure as possible. Which options should you set?

- A. PERMISSION\_SET = SAFE TRUSTWORTHY OFF
- B. PERMISSION\_SET = SAFE TRUSTWORTHY ON
- C. PERMISSION\_SET = UNSAFE TRUSTWORTHY ON
- D. PERMISSION\_SET = EXTERNAL\_ACCESS TRUSTWORTHY OFF

---

**Answer: A**

---

**Question: 104**

You have a table named ProductCounts that contains 1000 products as well as the number of units that have been sold for each product. You need to write a query that displays the top 5% of products that have been sold most frequently.Which Transact-SQL code segments should you use?

- A. WITH Percentages AS (SELECT \*, NTILE(5) OVER (ORDER BY UnitsSold) AS groupingColumnFROM ProductCounts)SELECT \*FROM percentagesWHERE groupingColumn =1;
- B. WITH Percentages AS ( SELECT \*, NTILE(20) OVER (ORDER BY UnitsSold) AS groupingColumnFROM ProductCounts)SELECT \*FROM PercentagesWHERE groupingColumn = 20;
- C. WITH Percentages AS (SELECT \*, NTILE(20) OVER (ORDER BY UnitsSold) AS groupingColumnFROM ProductCounts)SELECT \*FROM PercentagesWHERE groupingColumn = 1;



D. WITH Percentages AS (SELECT \*, NTILE(5) OVER (ORDER BY UnitsSold) AS groupingColumn FROM ProductCounts) SELECT \* FROM Percentages WHERE groupingColumn = 5;

**Answer: B**

#### Question: 105

You have two tables named MainTable and ArchiveTable. You need to move data older than 30 days from MainTable into ArchiveTable. Which code segment should you use?

- A. INSERT INTO ArchiveTable SELECT \* FROM MainTable WHERE RecordDate < DATEADD(D,-30,GETDATE())
- B. DELETE FROM MainTable OUTPUT DELETED.\* INTO ArchiveTable WHERE RecordDate < DATEADD(D,-30,GETDATE())
- C. DELETE FROM MainTable OUTPUT deleted.\* WHERE RecordDate < DATEADD(D,-30,GETDATE())
- D. INSERT INTO ArchiveTable SELECT \* FROM MainTable WHERE RecordDate < DATEADD(D,-30,GETDATE()) DELETE FROM MainTable

**Answer: B**

#### Question: 106

Your database contains sales information for millions of orders. You need to identify the orders with the highest average unit price and an order total greater than 10,000. The list should contain no more than 20 orders. Which query should you use?

- A. SELECT TOP (20) o.SalesOrderId, o.OrderDate, o.Total, SUM(od.QTY \* od.UnitPrice) / SUM(od.Qty) AS [AvgUnitPrice] FROM Sales.SalesOrderHeader o JOIN SALES.SalesOrderDetail od ON o.SalesOrderId = od.SalesOrderId WHERE o.Total > 10000 GROUP BY o.SalesOrderId, o.OrderDate, o.Total ORDER BY AvgUnitPrice;
- B. SELECT TOP (20) o.SalesOrderId, o.OrderDate, o.Total, (SELECT SUM(od.Qty \* od.UnitPrice) / SUM(od.Qty) FROM Sales.SalesOrderDetail od WHERE o.SalesOrderId = od.SalesOrderId) AS [AvgUnitPrice] FROM Sales.SalesOrderHeader o WHERE o.Total > 10000 ORDER BY o.Total DESC, AvgUnitPrice;
- C. SELECT TOP (20) o.SalesOrderId, o.OrderDate, o.Total, (SELECT SUM(od.Qty \* od.UnitPrice) / SUM(od.QTY) FROM Sales.SalesOrderDetail od WHERE o.SalesOrderId = od.SalesOrderId) AS [AvgUnitPrice] FROM Sales.SalesOrderHeader o WHERE o.Total > 10000 ORDER BY AvgUnitPrice DESC;
- D. SELECT TOP (20) o.SalesOrderId, o.OrderDate, o.Total, SUM(od.Qty \* od.UnitPrice) / SUM(od.Qty) AS [AvgUnitPrice] FROM Sales.SalesOrderHeader o JOIN Sales.SalesOrderDetail od ON o.SalesOrderId = od.SalesOrderId WHERE o.Total > 10000 GROUP BY o.SalesOrderId, o.OrderDate, o.Total ORDER BY Total DESC;

**Answer: C**

#### Question: 107

Note: This question is part of a series of questions that use the same set of answer choices. Each answer choice may be used once, more than once, or not at all. You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application.

You create tables named complaint, customer, and product as follows: CREATE TABLE

```
[dbo].[complaint] ([ComplaintID] [int],  
[ProductID] [int],  
[CustomerID] [int],  
[ComplaintDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],  
[CustomerName] [varchar](100),  
[Address] [varchar](200),  
[City] [varchar](100),  
[State] [varchar](50),  
[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],  
[ProductName] [varchar](100),  
[SalePrice] [money],  
[ManufacturerName] [varchar](100));
```

You need to write a query to return all customer names and total number of complaints for customers who have made more than 10 complaints. Which SQL query should you use?

- A. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());
- B. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500
- C. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID WHERE com.ComplaintDate > '09/01/2011' AND AVG(p.SalePrice) >= 500
- D. SELECT c.CustomerName, COUNT(com.ComplaintID) AS complaints FROM customer c INNER JOIN complaint com ON c.CustomerID = com.CustomerID GROUP BY c.CustomerName HAVING COUNT(com.ComplaintID) > 10;
- E. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, ComplaintMonth;
- F. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);
- G. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID INNER JOIN customer c ON com.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());
- H. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));
- I. SELECT p.ProductName, DATEPART(mm, com.ComplaintDate) ComplaintMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN complaint com ON p.ProductID = com.ProductID GROUP BY CUBE;
- J. SELECT c.CustomerName, COUNT(com.ComplaintID) AS Complaints FROM customer c INNER

JOINcomplaint com ON c.CustomerID = com.CustomerID WHERE COUNT(com.ComplaintID) >  
10 GROUP BY c.CustomerName;

**Answer: D**

**Question: 108**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database for an inventory management system.

The application contains a product table that has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[Active] [bit] NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
[ProductID] ASC
)
```

ON [PRIMARY]) ON [PRIMARY]GO You want to add a new field to the Product table to meet the following requirements:

- Allows user-specified information that will be added to records in the Product table.
- Supports the largest storage size needed for the field.
- Uses the smallest data type necessary to support the domain of values that will be entered by users. You need to add a field named User\_Data\_1 to support only values that are 1 or 0. Which SQL statement should you use?

- A. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLDATETIME
- B. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(11,6)
- C. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] TINYINT
- D. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NVARCHAR(100)
- E. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIGINT
- F. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLMONEY
- G. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NCHAR(100)
- H. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATE
- I. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] VARCHAR(100)
- J. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME
- K. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] SMALLINT
- L. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] INT
- M. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(6,11)
- N. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] DATETIME2
- O. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] CHAR(100)

- P. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] BIT  
Q. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] NUMERIC(5,6)  
R. ALTER TABLE [Production].[Product] ADD [User\_Data\_1] MONEY

---

**Answer: P**

---

**Question: 109**

You are responsible for a SQL Server database.

You require the tables to be added or altered only on the first day of the month.

You need to ensure that if the tables are attempted to be modified or created on any other day, an error is received and the attempt is not successful. Which Transact-SQL statement should you use?

- A. CREATE TRIGGER TRG\_TABLES\_ON\_FIRSTON DATABASE FOR CREATE\_TABLE,ALTER\_TABLE AS IF DATEPART(day,getdate())>1BEGINRAISERROR ('Must wait til next month.', 16, 1) END  
B. CREATE TRIGGER TRG\_TABLES\_ON\_FIRSTON DATABASE FOR CREATE\_TABLE AS IF DATEPART(day,getdate())>1BEGINRAISERROR ('Must wait til next month.', 16, 1) END  
C. CREATE TRIGGER TRG\_TABLES\_ON\_FIRSTON ALL SERVER FOR ALTER\_DATABASE AS IF DATEPART(day,getdate())>1BEGINROLLBACK RAISERROR ('Must wait til next month.', 16, 1)END  
D. CREATE TRIGGER TRG\_TABLES\_ON\_FIRSTON DATABASE FOR CREATE\_TABLE,ALTER\_TABLE AS IF DATEPART(day,getdate())>1BEGINROLLBACK RAISERROR ('Must wait til next month.', 16, 1) END

---

**Answer: D**

---

**Question: 110**

You have a table named Employees. You want to identify the supervisor to which each employee reports. You write the following query. SELECT e.EmployeeName AS [EmployeeName], s.EmployeeName AS [SuperVisorName]FROM Employees e You need to ensure that the query returns a list of all employees and their respective supervisor. Which join clause should you use to complete the query?

- A. RIGHT JOIN Employees sON e.ReportsTo = s.EmployeeId  
B. INNER JOIN Employees sON e.EmployeeId = s.EmployeeId  
C. LEFT JOIN Employees sON e.ReportsTo = s.EmployeeId  
D. LEFT JOIN Employees sON e.EmployeeId = s.EmployeeId

---

**Answer: C**

---

**Question: 111**

You have a table named Stores that has an XML column named OpenHours. This column contains the opening and closing times.

<hours dayofWeek="Monday" open="8:00" closed="18:00" />

<hours dayofWeek="Tuesday" open="8:00" closed="18:00" />

...

<hours dayofWeek="Saturday" open="8:00" closed="18:00" /> You need to write a query that returns a list of stores and their opening time for a specified day. Which code segment should you use?

- A. DECLARE @Day VARCHAR(10) = 'Tuesday' SELECT  
StoreName,OpenHours.value('/hours[1][@dayofWeek=sql:variable("@Day")]/@open','time')FROM  
M Stores
- B. DECLARE @Day VARCHAR(10) = 'Tuesday' SELECT  
StoreName,OpenHours.value('/hours[1]/@open','time')FROM StoresWHERE  
OpenHours.exist('/hours[@dayofWeek=sql:variable("@Day")])' = 1
- C. DECLARE @Day VARCHAR(10) = 'Tuesday' SELECT  
Storename,OpenHours.query('data(/hours[@dayofWeek=sql:variable("@Day")]/@open)')FROM  
Stores
- D. DECLARE @Day VARCHAR(10) = 'Tuesday' SELECT  
StoreName,OpenHours.value('/hours[1]/@open','time')FROM StoresWHERE  
OpenHours.value('/hours[1]/@dayofWeek','varchar(20)') = @Day

**Answer: C**

### Question: 112

Results		Messages	
	Ranking	StateProvinceName	TotalOrders
1	1	California	5466
2	2	Washington	2754
3	3	Oregon	1297
4	4	Illinois	6
5	4	Ohio	6
6	6	Florida	5
7	6	Texas	5
8	8	Georgia	4

You administer a Microsoft SQL Server 2008 database that contains a table named `dbo.SalesOrders`. The table has the following definition: `CREATE TABLE [dbo].[SalesOrder]([SalesOrderNumber] NVARCHAR(20) NOT NULL, [FullDateAlternateKey] DATETIME NOT NULL, [CustomerName] NVARCHAR(100) NOT NULL, [AddressLine] NVARCHAR(120) NOT NULL, [City] NVARCHAR(30) NOT NULL, [StateProvinceName] NVARCHAR(50) NOT NULL, [CountryName] NVARCHAR(50) NOT NULL, [SalesAmount] MONEY NOT NULL, CONSTRAINT [PK_SalesOrderNumber] PRIMARY KEY CLUSTERED ([SalesOrderNumber] ASC) ON [PRIMARY]) ON [PRIMARY]GO` The SalesOrder table contains one million rows. You want to create a report that meets the following requirements:

- Only the states of the Unites States are ranked against each other based on the total number of orders received from each state.
- When two states have the same rank, the rank of the subsequent state is one plus the number of ranks that come before that row, as shown in the exhibit. (Click the Exhibit button.) You need to

execute a Transact-SQL query to generate the report. Which Transact-SQL query should you use?

- A. `SELECT RANK() OVER (ORDER BY StateProvinceName DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT StateProvinceName, count(*) AS  
TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY StateProvinceName) AS`
- B. `SELECT DENSE_RANK() OVER (ORDER BY StateProvinceName DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT StateProvinceName, count(*) AS  
TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY StateProvinceName) AS`
- C. `SELECT DENSE_RANK() OVER (ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT StateProvinceName, count(*) AS  
TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY StateProvinceName) AS`
- D. `SELECT RANK() OVER (ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT StateProvinceName, count(*) AS  
TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY StateProvinceName) AS`
- E. `SELECT RANK() OVER (PARTITION BY CountryName ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT CountryName, StateProvinceName, count(*)  
AS TotalOrders FROM dbo.SalesOrder GROUP BY CountryName, StateProvinceName) AS`
- F. `SELECT DENSE_RANK() OVER (PARTITION BY CountryName ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT CountryName, StateProvinceName, count(*)  
AS TotalOrders FROM dbo.SalesOrder GROUP BY CountryName, StateProvinceName) AS`
- G. `SELECT RANK() OVER (PARTITION BY CountryName ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT CountryName, StateProvinceName, count(*)  
AS TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY CountryName, StateProvinceName) AS`
- H. `SELECT DENSE_RANK() OVER (PARTITION BY CountryName ORDER BY TotalOrders DESC) AS  
Ranking, StateProvinceName, TotalOrders FROM (SELECT CountryName, StateProvinceName, count(*)  
AS TotalOrders FROM dbo.SalesOrder WHERE CountryName = 'United States' GROUP  
BY CountryName, StateProvinceName) AS`

**Answer: D OR G**

### Question: 113

You have a table named Orders. You have been tasked to modify your company's main database to remove all inactive order rows. You are developing a stored procedure that will enable you to delete these rows. You have written the following code segment to accomplish this task.

```
01 BEGIN TRY 02  
DECLARE @RowCount INT = 1000 03  
WHILE @RowCount = 1000 04  
BEGIN 05  
DELETE TOP (1000) FROM Orders WHERE Status = 'Inactive'; 06  
SET @RowCount = @@ROWCOUNT 07
```



...08

END 09 END TRY 10 BEGIN CATCH 11

PRINT ERROR\_MESSAGE()12 END CATCH

You need to insert a Transact-SQL statement that will notify you immediately after each batch of rows is deleted.

Which Transact-SQL statement should you insert at line 07?

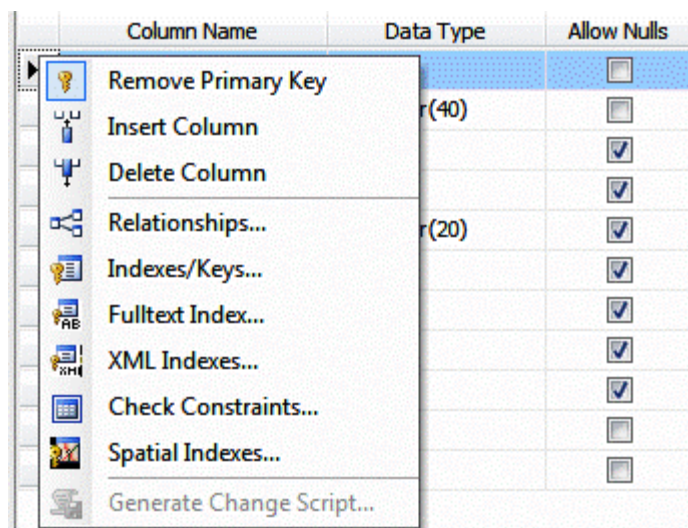
- A. RAISERROR ('Deleted %i rows', 11, 1, @RowCount) WITH NOWAIT
- B. RAISERROR ('Deleted %i rows', 16, 1, @RowCount)
- C. RAISERROR ('Deleted %i rows', 6, 1, @RowCount)
- D. RAISERROR ('Deleted %i rows', 10, 1, @RowCount) WITH NOWAIT

**Answer: D**

#### Question: 114

You administer a Microsoft SQL Server 2008 database that contains two tables named Products and Suppliers. You want to implement referential integrity between the Products and Suppliers tables. You want to create a new Foreign Key constraint on the Products table. The new Foreign Key constraint must meet the following requirements:

- It must be created on the SupplierID column on the Products table to refer to the SupplierID column on the Suppliers table.
- It must prevent the deletion of rows from the Suppliers table whenever an attempt is made to delete a Supplier by using a SupplierID referenced by Foreign Keys in existing rows in the Products table. You need to be able to create a new Foreign Key constraint by using Microsoft SQL Server Management Studio. What should you do? (To answer, configure the appropriate option or options in the dialog box in the answer area.)



Work Area

**Answer: D**

#### Question: 115



ParentSiteURL	SiteURL
NULL	
http://www.contoso.com/index.htm	
http://www.contoso.com/index.htm	http://www.contoso.com/finance/index.htm
http://www.contoso.com/finance/index.htm	http://www.contoso.com/finance/reports/index.htm
http://www.contoso.com/finance/index.htm	http://www.contoso.com/finance/main/index.htm
http://www.contoso.com/index.htm	http://www.contoso.com/marketing/index.htm
http://www.contoso.com/marketing/index.htm	http://www.contoso.com/marketing/reports/index.htm
http://www.contoso.com/marketing/index.htm	http://www.contoso.com/marketing/main/index.htm
http://www.contoso.com/index.htm	http://www.contoso.com/sales/index.htm

You have the following xml: <Site URL="http://www.contoso.com/index.htm">

<Site URL="http://www.contoso.com/finance/index.htm">

<Site URL="http://www.contoso.com/finance/reports/index.htm" />

<Site URL="http://www.contoso.com/finance/main/index.htm" />

</Site>

<Site URL="http://www.contoso.com/marketing/index.htm">

<Site URL="http://www.contoso.com/marketing/reports/index.htm" />

<Site URL="http://www.contoso.com/marketing/main/index.htm" />

</Site>

<Site URL="http://www.contoso.com/sales/index.htm" /></Site> You are tasked to query the sites listed in the xml by using OPENXML. The results will have two columns, ParentSiteURL and SiteURL. The ParentSiteURL column should contain the URL attribute of the parent site. The SiteURL column should contain the URL attribute of the site itself. The output should look like as shown in the following the exhibit. (Click the Exhibit button.) You need to write the OPENXML query. Which Transact-SQL statement should you use?

- A. SELECT ParentSiteURL, SiteURL FROM OPENXML (@XMLDocHandle, '//@URL', 1) WITH (ParentSiteURL nVarChar(512) ' ../URL', SiteURL nVarChar(512) 'URL')
- B. SELECT ParentSiteURL, SiteURL FROM OPENXML (@XMLDocHandle, '//@URL', 1) WITH (ParentSiteURL nVarChar(512) ' ../@URL', SiteURL nVarChar(512) '@URL')
- C. SELECT ParentSiteURL, SiteURL FROM OPENXML (@XMLDocHandle, '//@Site', 1) WITH (ParentSiteURL nVarChar(512) ' ../URL', SiteURL nVarChar(512) 'URL')
- D. SELECT ParentSiteURL, SiteURL FROM OPENXML (@XMLDocHandle, '//@Site', 1) WITH (ParentSiteURL nVarChar(512) ' ../@URL', SiteURL nVarChar(512) '@URL')

**Answer: D**

### Question: 116

You create and populate two tables by using the following Transact-SQL statements: CREATE TABLE CurrentStudents (LastName VARCHAR(50), FirstName VARCHAR(50), Address VARCHAR(100), Age INT); INSERT INTO CurrentStudents VALUES ('Fritz', 'David', '181 Kline Street', 14), ('Reese', 'Paul', '4429 South Union', 14), ('Brown', 'Jake', '5401 Washington Ave', 14), ('Smith', 'Tom', '124 Water St', 14)

```
,('Holtz', 'Mary', '984 Mass Ct', 14)
,('Robbins', 'Jan', '4449 Union Ave', 14)
,('Larsen', 'Frank', '5812 Meadow St', 14)
,('Bishop', 'Cathy', '14429 Skyhigh Ave', 14)
,('Francis', 'Thomas', '15401 120th St', 14)
CREATE TABLE NewYearRoster(LastName VARCHAR(50),
FirstName VARCHAR(50),
Address VARCHAR(100),
Age INT);INSERT INTO NewYearRoster
VALUES ('Fritz', 'David', '181 Kline Street', 15)
,('Reese', 'Paul', '1950 Grandview Place', 15)
,('Adams', 'Wilbur', '4231 W. 93rd', 15)
,('Adams', 'Norris', '100 1st Ave', 15)
,('Thomas', 'Paul', '18176 Soundview Dr', 15)
,('Linderson', 'Danielle', '941 W. 37 Ave', 15)
,('Moore', 'Joshua', '2311 10st Ave', 15)
,('Dark', 'Shelby', '1987 Fifth Ave', 15)
,('Scharp', 'Mary', '1902 W. 303rd', 15)
,('Morris', 'Walt', '100 12st St', 15); You run the following MERGE statement to update, insert and
delete rows in the CurrentStudents table MERGE TOP (3) CurrentStudents AS T USING
NewYearRoster AS SON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN
MATCHED AND NOT (T.Age = S.Age OR T.Address = S.Address) THEN
UPDATE SET Address = S.Address,
Age = S.Age WHEN NOT MATCHED BY TARGET THEN
INSERT (LastName, FirstName, Address, Age)
VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN
DELETE; You need to identify the total number of rows that are updated, inserted, and deleted in
the CurrentStudent table. Which total number of rows should you choose?
A. 9
B. 0
C. 3
D. 6
```

---

**Answer: C**

---

**Question: 117**

You administer a Microsoft SQL Server 2008 R2 database instance named AdventureWorks. A user who has the db\_datareader permissions on the AdventureWorks database wants to view the estimated execution plan XML output document for the following query:

```
SELECT * FROM Sales.SalesOrderHeader WHERE OnlineOrderFlag = 1 AND SubTotal > 500
```

You need to ensure that the user can view the document. Which two actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

A. Grant the following permission to the user: GRANT SHOWPLAN

- B. Grant the following permission to the user: GRANT EXECUTE ON XML SCHEMA COLLECTION
- C. Grant the following permission to the user: GRANT SELECT ON  
OBJECT::Sales.SalesOrderHeader
- D. Request the user to run the following command: SET SHOWPLAN\_XML ON
- E. Request the user to run the following command: SET SHOWPLAN\_ALL ON
- F. Request the user to run the following command: SET STATISTICS IO ON
- G. Request the user to run the following command: SET STATISTICS XML ON

**Answer: (A BEFORE D) AND  
ONLY (A, D)**

**Question: 118**

You administer a Microsoft SQL Server 2008 instance that has two databases. The first database named AdventureWorks contains a table named Sales.SalesOrders.

The Sales.SalesOrders table has the following definition: CREATE TABLE [Sales].[SalesOrderDetail]([SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,  
[ProductID] [int] NOT NULL,  
[OrderQty] [smallint] NOT NULL,  
[OrderDate] [datetime] NOT NULL,  
CONSTRAINT [PK\_SalesOrderDetail] PRIMARY KEY CLUSTERED

(  
[SalesOrderDetailID] ASC

)) ON [PRIMARY] The second database named AdventureWorksDW contains a table named dbo.SalesOrderSummary.

The dbo.SalesOrderSummary table has the following definition: CREATE TABLE

[dbo].[SalesOrderSummary] (  
ProductID [int] NOT NULL,  
OrderQty [int] NOT NULL,  
OrderYear [int] NOT NULL,  
CONSTRAINT [PK\_SalesOrderSummary] PRIMARY KEY CLUSTERED  
(

OrderYear ASC,  
ProductID ASC

)) ON [PRIMARY] You plan to migrate sales data for the year 2011 from the SalesOrderDetail table into the SalesOrderSummary table. You need to ensure that the following requirements are met:

- All data is removed from the SalesOrderSummary table before migrating data.
- A subset of data is migrated from the SalesOrderDetail table in the AdventureWorks database to the SalesOrderSummary table in the AdventureWorksDW database.
- Migrated data summarizes order quantity in one row per product for the year 2011. 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.)

- A. USE AdventureWorksGO
- B. USE AdventureWorksDWGO
- C. TRUNCATE TABLE Sales.SalesOrderDetailGO

- D. TRUNCATE TABLE dbo.SalesOrderSummaryGO
- E. MERGE dbo.SalesOrderSummary AS dstUSING (SELECTProductID,SUM(OrderQty) AS OrderQty,YEAR(OrderDate) AS OrderYearFROMAdventureWorks.Sales.SalesOrderDetailWHEREYEAR(OrderDate)=2011GROUP BYProductID,YEAR(OrderDate)) AS srcON dst.ProductID = src.ProductIDWHEN NOT MATCHED BY SOURCE THENINSERT(ProductID, OrderQty, OrderYear)VALUES(src.ProductID, src.OrderQty, src.OrderYear);GO
- F. MERGE dbo.SalesOrderSummary AS dstUSING (SELECTProductID,SUM(OrderQty) AS OrderQty,YEAR(OrderDate) AS OrderYearFROMAdventureWorks.Sales.SalesOrderDetailWHEREYEAR(OrderDate)=2011GROUP BYProductID,YEAR(OrderDate)) AS srcON dst.ProductID = src.ProductIDWHEN MATCHED THENDELETE;GO
- G. INSERT dbo.SalesOrderSummary (ProductID, OrderQty, OrderYear)SELECTProductID,SUM(OrderQty) AS OrderQty,YEAR(OrderDate) AS OrderYear FROMAdventureWorks.Sales.SalesOrderDetailWHEREYEAR(OrderDate)=2011GROUP BYProductID,YEAR(OrderDate)GO
- H. INSERT dbo.SalesOrderSummary (ProductID, OrderQty, OrderYear)SELECTProductID,SUM(OrderQty),YEAR(2011) FROMAdventureWorks.Sales.SalesOrderDetailGROUP BYProductIDHAVINGYEAR(2011)=2011GO

**Answer: (B BEFORE D) AND (D BEFORE G) AND ONLY (B, D, G)**

### Question: 119

You have a table named Subcategories that contains subcategories for socks, vests and helmets. You have another table named Products that contains products only from the subcategories socks and vests.

You have the following query: SELECT s.Name, p.Name AS ProductNameFROM Subcategories s OUTER APPLY (SELECT \*

FROM Products pr

WHERE pr.SubcategoryID = s.SubcategoryID) p

WHERE s.Name IS NOT NULL; You need to predict the results of the query. What results should the query produce?

- A. Name ProductName-----Socks Mountain Bike Socks,Socks Mountain Bike Socks,Socks Racing Socks, MSocks Racing Socks, LVests Classic Vest, SVests Classic Vest, MVests Classic Vest, LHelmets NULLNULL NULL
- B. Name ProductName-----Socks Mountain Bike Socks,Socks Mountain Bike Socks,Socks Racing Socks, MSocks Racing Socks, LVests Classic Vest, SVests Classic Vest, MVests Classic Vest, LHelmets NULL
- C. Name ProductName-----Socks Mountain Bike Socks,Socks Mountain Bike Socks,Socks Racing Socks, MSocks Racing Socks, LVests Classic Vest, SVests Classic Vest, MVests Classic Vest, LNULL Mountain Bike Socks,NULL Mountain Bike Socks,NULL Racing Socks, MNULL

Racing Socks, LNULL Classic Vest, SNULL Classic Vest, MNULL Classic Vest, LHelmets NULLNULL NULL  
D. Name ProductName-----Socks Mountain Bike Socks,Socks Mountain Bike Socks,Socks Racing Socks, MSocks Racing Socks, LVests Classic Vest, SVests Classic Vest, MVests Classic Vest, L

---

**Answer: B**

---

**Question: 120**

You are working with a SQL Server 2008 instance that is configured to use the Icelandic\_CS\_AS collation. You create a database by using code that includes the following statements. CREATE DATABASE InternationalDB COLLATE Japanese\_CS\_AS;GOUSE InternationalDB;GOCREATE TABLE AppPermTable (PrimaryKey int PRIMARY KEY, AppID nchar ); You implement a temporary table named #AppTempTable that uses the following code. USE InternationalDB;GOCREATE TABLE #AppTempTable (PrimaryKey int PRIMARY KEY, AppID nchar );INSERT INTO #AppTempTable SELECT \* FROM AppPermTable; You need to identify the collation that will be assigned to #AppTempTable. Which collation will be assigned?

- A. Icelandic\_CS\_AS
- B. No collation
- C. The collation selected by the Windows system locale of the server
- D. Japanese\_CS\_AS

---

**Answer: A**

---

**Question: 121**

You administer a Microsoft SQL Server 2008 database named AdventureWork that contains a table named Production.Product.

The table has the following definition: CREATE TABLE [Production].[Product](

[ProductID] [int] IDENTITY(1,1) NOT NULL,

[ProductName] [nvarchar](50) NOT NULL,

[ListPrice] [money] NOT NULL,

[ProductionDate] [datetime] NULL,

[UserCreated] [nvarchar](128) NOT NULL,

[DateCreated] [datetime] NOT NULL DEFAULT GETDATE(),

CONSTRAINT [PK\_Product\_ProductID] PRIMARY KEY CLUSTERED (

[ProductID] ASC) ON [PRIMARY]) ON [PRIMARY]GO You want to add a new product named

Widget and a list price of 10.50 U.S. dollars to the product table. You need to add a record for the product information. You also need to set the DateCreated field to the current date and the UserCreated field to your Windows login identification name. Which Transact-SQL statement should you use?

- A. INSERT INTO [Production].[Product](ProductID, ProductName, ListPrice, UserCreated, DateCreated)values(1, 'Widget', 10.50, system\_user, GETDATE())
- B. INSERT INTO [Production].[Product](ProductName, ListPrice, UserCreated)values('Widget', 10.50, system\_user)
- C. INSERT INTO [Production].[Product](ProductName, ListPrice, UserCreated,

ProductionDate)values('Widget', 10.50, user\_id(), GETDATE())  
D. INSERT INTO [Production].[Product](ProductName, ListPrice, DateCreated)values('Widget', 10.50, GETDATE())  
E. INSERT INTO [Production].[Product](ProductID, ProductName, ListPrice, UserCreated, DateCreated)select 1, 'Widget', 10.50, system\_user, GETDATE()  
F. INSERT INTO [Production].[Product](ProductName, ListPrice, UserCreated)select 'Widget', 10.50, system\_user  
G. INSERT INTO [Production].[Product](ProductName, ListPrice, UserCreated, ProductionDate)select'Widget', 10.50, user\_id(), GETDATE()  
H. INSERT INTO [Production].[Product](ProductName, ListPrice, DateCreated)select 'Widget', 10.50, GETDATE()

---

**Answer: B OR F**

---

**Question: 122**

You have a table named Person that contains a nvarchar column named Surname. The Person table currently has a clustered index on PersonID. The Surname column contains Russian and Japanese characters. The following code segment will be used to search by Surname.

```
IF @lang = 'Russian'
```

```
SELECT PersonID, Surname
```

```
FROM Person
```

```
WHERE Surname = @SearchName COLLATE Cyrillic_General_CI_AS IF @lang = 'Japanese'
```

```
SELECT PersonID, Surname
```

```
FROM Person
```

```
WHERE Surname = @SearchName COLLATE Japanese_CI_AS_KS
```

You need to enable SQL Server to perform an index seek for these queries. What should you do?

- A. Create a computed column for each collation that needs to be searched. Create an index on each computed column.
- B. Create a new column for each collation that needs to be searched and copy the data from the Surname column. Create an index on each new column.
- C. Create a computed column for each collation that needs to be searched. Create an index on the Surname column.
- D. Create an index on the Surname column.

---

**Answer: A**

---

**Question: 123**

You create a stored procedure that contains proprietary formulas. You need to ensure that no users are able to access Microsoft SQL Server Management Studio to view the definition of the stored procedure. Which code segment should you use?

- A. CREATE PROCEDURE Sales.uspFormulaWITH EXECUTE AS SELF
- B. CREATE PROCEDURE Sales.uspFormula WITH RECOMPILE
- C. CREATE PROCEDURE Sales.uspFormula WITH EXECUTE AS OWNER
- D. CREATE PROCEDURE Sales.uspFormulaWITH ENCRYPTION



**Answer: D****Question: 124**

You administer a Microsoft SQL Server 2008 R2 instance configured to use Windows Authentication.

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

```
CREATE TABLE dbo.CustomerTransaction(CustomerTransactionId int NOT NULL PRIMARY KEY,
```

```
CustomerId int NOT NULL, TransactionAmount money NOT NULL )
```

 You define the following table:

```
CREATE TABLE dbo.CustomerWarningLog(
```

```
CustomerWarningLogId int NOT NULL identity PRIMARY KEY,
```

```
CustomerId int NOT NULL,
```

```
Balance money NOT NULL,
```

```
LogTime datetime2(0) NOT NULL,
```

```
LogUserName nvarchar(128) NOT NULL)
```

 You need to ensure that the following requirements are met:

- An entry is logged in CustomerWarningLog when a customer's account balance is less than 100.00.

- The TransactionLogUserName is set to the login name of the user who modifies the CustomerTransaction table. Which Transact-SQL statement or statements should you use?

A. CREATE TRIGGER dbo.CustomerTransaction\_InsertUpdateDeleteTriggerON

```
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEAS IF UPDATE(CustomerID)WITH TotalCTE AS(SELECT CustomerId, SUM(TransactionAmount) AS BalanceFROM  
dbo.CustomerTransactionWHERE CustomerId in (SELECT CustomerIdFROM insertedUNION  
ALLSELECT CustomerIdFROM deleted)GROUP BY CustomerId)INSERT dbo.  
CustomerWarningLog(CustomerId, Balance, LogTime, LogUserName)SELECT CustomerId, Balance,  
SYSDATETIME(), SUSER_SNAME()FROM TotalForCustomerWHERE TotalCTE.Amount < 100 GO
```

B. CREATE TRIGGER dbo.CustomerTransaction\_InsertUpdateTriggerON

```
dbo.CustomerTransactionFOR INSERT, UPDATEAS IF UPDATE(CustomerID) or  
UPDATE(TransactionAmount)WITH TotalCTE AS(SELECT CustomerId, SUM(TransactionAmount) AS  
BalanceFROM dbo.CustomerTransactionWHERE CustomerId in (SELECT CustomerIdFROM  
insertedUNION ALLSELECT CustomerIdFROM deleted)GROUP BY CustomerId)INSERT dbo.  
CustomerWarningLog(CustomerId, Balance, LogTime, LogUserName)SELECT CustomerId, Balance,  
SYSDATETIME(), USER_NAME()FROM TotalForCustomerWHERE TotalCTE.Amount < 100 GO
```

C. CREATE TRIGGER dbo.CustomerTransaction\_InsertUpdateDeleteTriggerON

```
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEASDECLARE @CustomerId int, @Balance  
moneyWITH TotalCTE AS(SELECT CustomerId, SUM(TransactionAmount) AS BalanceFROM  
dbo.CustomerTransactionWHERE CustomerId in (SELECT CustomerIdFROM insertedUNION  
ALLSELECT CustomerIdFROM deleted)GROUP BY CustomerId)SELECT @CustomerId =  
CustomerId,@Balance = TransactionAmountFROM TotalForCustomerWHERE TotalCTE.Amount <  
100IF @@rowcount > 0INSERT dbo. CustomerWarningLog(CustomerId, Balance, LogTime,  
LogUserName)SELECT @CustomerId, @Balance, SYSDATETIME(), SUSER_SNAME() GO
```

D. CREATE TRIGGER dbo.CustomerTransaction\_InsertUpdateDeleteTriggerON

```
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEAS IF UPDATE(CustomerID) or
```



```
UPDATE(TransactionAmount)WITH TotalCTE AS(SELECT CustomerId, SUM(TransactionAmount) AS
BalanceFROM dbo.CustomerTransactionWHERE CustomerId in (SELECT CustomerIdFROM
insertedUNION ALLSELECT CustomerIdFROM deleted)GROUP BY CustomerId)INSERT dbo.
CustomerWarningLog(CustomerId, Balance, LogTime, LogUserName)SELECT CustomerId, Balance,
SYSDATETIME(), SUSER_SNAME()FROM TotalForCustomerWHERE TotalCTE.Amount < 100 GO
E. CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTriggerON
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEASDECLARE @CustomerId int, @Balance
moneyWITH TotalCTE AS(SELECT CustomerId, SUM(TransactionAmount) AS BalanceFROM
dbo.CustomerTransactionWHERE CustomerId in (SELECT CustomerIdFROM insertedUNION
ALLSELECT CustomerIdFROM deleted)GROUP BY CustomerId)SELECT @CustomerId =
CustomerId,@Balance = TransactionAmountFROM TotalForCustomerWHERE TotalCTE.Amount <
100IF @CustomerId IS NULLINSERT dbo. CustomerWarningLog(CustomerId, Balance, LogTime,
LogUserName)SELECT @CustomerId, @Balance, SYSDATETIME(), USER_NAME() GO
F. CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTriggerON
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEASWITH TotalCTE AS(SELECT CustomerId,
SUM(TransactionAmount) AS BalanceFROM dbo.CustomerTransactionWHERE CustomerId in
(SELECT CustomerIdFROM inserted)GROUP BY CustomerId)INSERT dbo.
CustomerWarningLog(CustomerId, Balance, LogTime, LogUserName)SELECT CustomerId, Balance,
SYSDATETIME(), SUSER_SNAME()FROM TotalForCustomerWHERE TotalCTE.Amount <= 100 GO
G. CREATE TRIGGER dbo.CustomerTransaction_UpdateDeleteTriggerON
dbo.CustomerTransactionFOR UPDATE, DELETEASWITH TotalCTE AS(SELECT CustomerId,
SUM(TransactionAmount) AS BalanceFROM dbo.CustomerTransactionWHERE TransactionAmount
< 100AND CustomerId in (SELECT CustomerIdFROM deleted)GROUP BY CustomerId)INSERT dbo.
CustomerWarningLog(CustomerId, Balance, LogTime, LogUserName)SELECT CustomerId, Balance,
SYSDATETIME(), SUSER_SNAME()FROM TotalForCustomer GO
H. CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTriggerON
dbo.CustomerTransactionFOR INSERT, UPDATE, DELETEASWITH TotalCTE AS(SELECT CustomerId,
SUM(TransactionAmount) AS BalanceFROM dbo.CustomerTransactionWHERE CustomerId in
(SELECT CustomerIdFROM insertedUNION ALLSELECT CustomerIdFROM deleted)GROUP BY
CustomerId)INSERT dbo. CustomerWarningLog(CustomerId, Balance, LogTime,
LogUserName)SELECT CustomerId, Balance, SYSDATETIME(), SUSER_SNAME()FROM
TotalForCustomerWHERE TotalCTE.Amount < 100 GO
```

**Answer: D OR H****Question: 125**

You are writing a batch that contains multiple UPDATE statements to modify existing products. You have placed these updates into one explicit transaction. You need to set an option at the beginning of the transaction to roll back all changes if any of the updates in the transaction fail. Which option should you enable?

- A. REMOTE\_PROC\_TRANSACTIONS
- B. ARITHABORT
- C. XACT\_ABORT

## D. IMPLICIT\_TRANSACTIONS

---

**Answer: C**

---

**Question: 126**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK\_Product\_ProductID and a non-clustered index named AK\_Product\_ProductNumber. Both indexes have been created on a single primary partition. The table has the following definition: CREATE TABLE

```
[Production].[Product](
[ProductID] [int] IDENTITY(1,1) NOT NULL,
[Name] [nvarchar](50) NOT NULL,
[ProductNumber] [nvarchar](25) NOT NULL,
[Color] [nvarchar](15) NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
[ProductID] ASC
```

```
) ON [PRIMARY]) ON [PRIMARY]GO The index has the following definition: CREATE UNIQUE
NONCLUSTERED INDEX [AK_Product_ProductNumber] ON [Production].[Product] (
[ProductNumber] ASC) ON [PRIMARY]GO The Production.Product table contains 1 million rows.
You want to ensure that data retrieval takes the minimum amount of time when the queries
executed against the Production.Product table are ordered by product number or filtered by class.
You observe that the average fragmentation for the AK_Product_ProductNumber index is 24
percent.
```

You need to reduce fragmentation. You need to achieve this goal without blocking access to table data. Which Transact-SQL statement should you use?

- A. EXEC sys.sp\_configure 'index create memory', 1
- B. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class <> null
- C. UPDATE INDEX AK\_Product\_ProductNumber ON Production.Product SET (STATISTICS\_NORECOMPUTE = ON)
- D. ALTER DATABASE [AdventureWorks] SET AUTO\_UPDATE\_STATISTICS ON
- E. DBCC SHOW\_STATISTICS ('Production.Product', AK\_Product\_ProductNumber)
- F. CREATE STATISTICS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class is not null
- G. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REORGANIZE
- H. CREATE STATS ProductClass\_StatsON Production.Product (Name, ProductNumber, Class)WHERE Class is not nullWITH SAMPLE 100 PERCENT
- I. SELECT \* FROM sys.dm\_db\_index\_operational\_stats (DB\_ID(),

- OBJECT\_ID(N'Production.Product'),NULL, NULL)  
J. SELECT \* FROM STATS WHERE name='AK\_Product\_ProductNumber'  
K. SELECT \* FROM SYS.STATS WHERE name='AK\_Product\_ProductNumber'  
L. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD Partition = 1  
M. UPDATE STATISTICS Production.Product  
N. SELECT \* FROM sys.dm\_db\_index\_physical\_stats (DB\_ID(),  
OBJECT\_ID(N'Production.Product'),NULL, NULL, NULL)  
O. ALTER INDEX AK\_Product\_ProductNumber ON Production.Product REBUILD  
P. ALTER STATISTICS Production.Product  
Q. SELECT \* FROM sys.indexes where name=N'Production.Product'

---

**Answer: G**

---

**Question: 127**

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\_ValidCustomerASSELECT FirstName,LastNameFROM Customer cEXCEPTSELECT FirstName,LastNameFROM CustomerExclusionList

B. CREATE VIEW vw\_ValidCustomerASSELECT c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN CustomerExclusionList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

C. CREATE VIEW vw\_ValidCustomerASSELECT c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cLEFT OUTER JOIN CustomerExclusionList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastNameWHERE cel.FirstName IS NULL

D. CREATE VIEW vw\_ValidCustomerASSELECT c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN CustomerExclusionList celON c.Firstname = cel.FirstNameINNER JOIN CustomerExclusionList celON c.LastName = cel.LastName

E. CREATE VIEW vw\_ValidCustomerASSELECT c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cEXCEPTSELECT c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN CustomerExclusionList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

F. CREATE VIEW vw\_ValidCustomerASSELECT  
 c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINTERSECTSELECT  
 c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cINNER JOIN  
 CustomerExclusionList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastName

G. CREATE VIEW vw\_ValidCustomerASSELECT  
 c.CustomerID,c.FirstName,c.LastName,c.CustomerAddressFROM Customer cLEFT OUTER JOIN  
 CustomerExclusionList celON c.Firstname = cel.FirstNameAND c.LastName = cel.LastNameWHERE  
 cel.FirstName IS NOT NULL

H. CREATE VIEW vw\_ValidCustomerASSELECT  
 CustomerID,FirstName,LastName,CustomerAddressFROM Customer cEXCEPTSELECT  
 CustomerID,FirstName,LastName,CustomerAddressFROM CustomerExclusionList

**Answer: C OR E**

### Question: 128

You work for a company that provides marketing data to other companies. You have the following

Transact-SQL statement: DECLARE @CustomerDemographics XMLSET

@CustomerDemographics=N'<CustomerDemographics>

<Customer CustomerID="1" Age="21" Education="High School">

<IsCoffeeDrinker>0</IsCoffeeDrinker>

</Customer>

<Customer CustomerID="2" Age="27" Education="College">

<IsCoffeeDrinker>1</IsCoffeeDrinker>

<IsFriendly>1</IsFriendly>

</Customer>

<Customer CustomerID="3" Age="35" Education="Unknown">

<IsCoffeeDrinker>1</IsCoffeeDrinker>

<IsFriendly>1</IsFriendly>

</Customer></CustomerDemographics>'DECLARE @OutputAgeOfCoffeeDrinkers XMLSET

@OutputAgeOfCoffeeDrinkers = @CustomerDemographics.query('

for \$output in /child::CustomerDemographics/child::Customer[( child::IsCoffeeDrinker[1] cast as  
 xs:boolean ? )]

return

<CoffeeDrinkingCustomer>

{ \$output/attribute::Age \}

</CoffeeDrinkingCustomer>')SELECT @OutputAgeOfCoffeeDrinkers You need to determine the  
 result of the query. What result should you expect?

A. <CoffeeDrinkingCustomer Age="27" /><CoffeeDrinkingCustomer Age="35" />

B. <CustomerDemographics><Customer><CoffeeDrinkingCustomer Age="21"  
 /></Customer></CustomerDemographics>

C. <CustomerDemographics><Customer><CoffeeDrinkingCustomer Age="27"  
 /></Customer><Customer><CoffeeDrinkingCustomer Age="35"

/></Customer></CustomerDemographics>

D. <CoffeeDrinkingCustomer Age="21" />

---

**Answer: A**

---

**Question: 129**

You have a SQL Server database. The database contains two schemas named Marketing and Sales. The Marketing schema is owned by a user named MarketingManager. The Sales schema is owned by a user named SalesManager. A user named John must be able to access the Sales.Orders table by using a stored procedure named Marketing.GetSalesSummary. John is not granted a SELECT permission on the Sales.Orders table. A user named SalesUser does have SELECT permission on the Sales.Orders table.

You need to implement appropriate permissions for John and the stored procedure Marketing.GetSalesSummary. What should you do?

- A. Marketing.GetSalesSummary should be created by using the EXECUTE AS CALLER clause. John should be granted IMPERSONATE permission for the user named SalesUser.
- B. Marketing.GetSalesSummary should be created by using the EXECUTE AS OWNER clause. John should be granted EXECUTE WITH GRANT OPTION on Marketing.GetSalesSummary.
- C. Marketing.GetSalesSummary should be created by using the EXECUTE AS 'SalesUser' clause. John should be granted EXECUTE permission on Marketing.GetSalesSummary.
- D. Marketing.GetSalesSummary should be created without an EXECUTE AS clause. John should be granted SELECT permission on the Sales.Orders table.

---

**Answer: C**

---

**Question: 130**

You use a table that stores movie information in your database.

The table has the following schema: CREATE TABLE Movies(

MovieID nchar(4) NOT NULL,

Title nvarchar(25) NOT NULL,

MovieDescription nvarchar(50) NULL,

Rating smallint NOT NULL CONSTRAINT Check\_Rating CHECK (Rating between 1 AND 7),

CONSTRAINT [PK\_Movies] PRIMARY KEY CLUSTERED

(

[MovieID] ASC

))GO You plan to insert a new record for a movie that will be released by using the following information:

Column	Data
MovieID	M999
Title	Movie to be released
MovieDescription	This movie is yet to be released
Rating	0

You need to ensure that the following requirements are met:

- Only this movie record gets a Rating of 0.
- Other movie records that will be inserted in the future must get a Rating of 1 through 7. 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.)

- A. ALTER TABLE Movies NOCHECK CONSTRAINT Check\_Rating;GO
- B. ALTER TABLE Movies WITH CHECK CONSTRAINT Check\_Rating;GO
- C. ALTER TABLE Movies DROP CONSTRAINT Check\_Rating;GO
- D. ALTER TABLE Movies CHECK CONSTRAINT Check\_Rating;GO
- E. ALTER TABLE Movies WITH NOCHECK CONSTRAINT Check\_Rating;GO
- F. INSERT INTO Movies VALUES ('M999','Movie To Be Released','This movie yet to be released',0)GO
- G. INSERT INTO Movies VALUES ('M999','Movie To Be Released','This movie yet to be released', DEFAULT)GO

**Answer: (A BEFORE F) AND  
(F BEFORE D) AND ONLY (A,  
D, F)**

#### Question: 131

You are the administrator of a SQL Server database. Database table modifications and additions must occur only between 11:00 P.M. and midnight. You need to ensure that if database table modifications or additions are attempted at any other time, an error is raised and the attempt is not successful. Which Transact-SQL statement should you use?

- A. CREATE TRIGGER TRG\_TABLES\_ON\_LAST\_HOURL ON ALL SERVER FOR ALTER\_DATABASE AS IF DATEPART(hour, getdate()) <> 1 BEGIN ROLLBACK RAISERROR ('Must wait.', 16, 1) END
- B. CREATE TRIGGER TRG\_TABLES\_ON\_LAST\_HOURL ON DATABASE FOR CREATE\_TABLE AS IF DATEPART(hour, getdate()) <> 23 BEGIN RAISERROR ('Must wait.', 16, 1) END
- C. CREATE TRIGGER TRG\_TABLES\_ON\_LAST\_HOURL ON DATABASE FOR CREATE\_TABLE, ALTER\_TABLE AS IF DATEPART(hour, getdate()) <> 23 BEGIN RAISERROR ('Must wait.', 16, 1) END
- D. CREATE TRIGGER TRG\_TABLES\_ON\_LAST\_HOURL ON DATABASE FOR CREATE\_TABLE, ALTER\_TABLE AS IF DATEPART(hour, getdate()) <> 1 BEGIN ROLLBACK RAISERROR ('Must wait.', 16, 1) END

**Answer: D**

#### Question: 132

You need to generate the following XML document.

<ProductExport>

<Product Price="99">Product1</Product>

<Product Price="199">Product2</Product>

<Product Price="299">Product3</Product>

<Product Price="399">Product4</Product></ProductExport> Which query should you use?

- A. SELECT Price, ProductName FROM Products FOR XML AUTO, ROOT('ProductExport')
- B. SELECT Price [@Price], ProductName AS [\*] FROM Products FOR XML PATH('Product'), ROOT('ProductExport')
- C. SELECT Price, ProductName FROM Products AS ProductExport FOR XML PATH('Product')
- D. SELECT Price [@Price], ProductName AS [\*] FROM Products AS ProductExport FOR XML AUTO,

## ELEMENTS

---

**Answer: B**

---

**Question: 133**

You have a table named Customer that has an XML column named Locations. This column stores an XML fragment that contains details of one or more locations, as show in the following examples. <Location City="Sydney" Address="..." PhoneNumber="..." /><Location City="Chicago" Address="..." PhoneNumber="..." /><Location City="London" Address="..." PhoneNumber="..." /> You need to write a query that returns a row for each of the customer's locations. Each resulting row must include the customer name, city, and an XML fragment that contains the location details. Which query should you use?

- A. SELECT CustomerName,Loc.value('@City','varchar(100)'),Loc.query('.')FROM CustomerCROSS APPLY Customer.Locations.nodes ('/Location') Locs(Loc)
- B. SELECT CustomerName,Locations.query('for \$i in /Location return data(\$i/@City)'),Locations.query('for \$i in /Location return \$i')FROM Customer
- C. SELECT CustomerName,Locations.query('data(/Location/@City)'),Locations.query('/Location')FROM Customer
- D. SELECT CustomerName,Locations.query('for \$i in /Location return element Location {\$i/@City,\$i}')FROM Customer

---

**Answer: A**

---

**Question: 134**

Your company manufactures and distributes bicycle parts. You have a full-text catalog on the Inventory table which contains the PartName and Description columns. You also use a full-text thesaurus to expand common bicycle terms.

You need to write a full-text query that will not only match the exact word in the search, but also the meaning. Which Transact-SQL statement should you use?

- A. SELECT \* FROM InventoryWHERE Description LIKE '%cycle%'
- B. SELECT \* FROM InventoryWHERE FREETEXT (\*, 'cycle')
- C. SELECT \* FROM InventoryWHERE CONTAINS (\*, 'FormsOf(Inflectional, cycle)')
- D. SELECT \* FROM InventoryWHERE CONTAINS (\*, 'cycle')

---

**Answer: B**

---

**Question: 135**

You are using the Database Engine Tuning Advisor (DTA) to analyze a workload. You need to save the recommendations generated by the DTA. Which command should you use?

- A. Export Session Results
- B. Export Session Definition
- C. Import Session Definition
- D. Preview Workload Table



---

**Answer: A**

---

**Question: 136**

You have two tables. A table named Student.CurrentStudents contains the names of all students enrolled for the current year. Another table named Student.NewYearRoster contains the names of students who have enrolled for the upcoming year. You have been tasked to write a MERGE statement to:

- Insert into Student.CurrentStudents the names of students who are enrolled for the upcoming year but not for the current year.

- Update information in Student.CurrentStudents for students who are enrolled both in the current year and in the upcoming year.

- Delete from Student.CurrentStudents the names of students who are not enrolled for the upcoming year. You need to write the appropriate MERGE statement. Which Transact-SQL statement should you use?

A. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS SON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED THENUPDATE SET Address = S.Address, Age = S.AgeWHEN NOT MATCHED BY TARGET THENINSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN DELETE;

B. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS SON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED AND NOT T.Address = S.Address AND NOT T.Age = S.Age THENUPDATE SET T.Age = S.Age, T.Address = S.AddressWHEN NOT MATCHED BY TARGET THENINSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN DELETE;

C. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS SON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED AND NOT T.Address = S.Address OR NOT T.Age = S.Age THENUPDATE SET T.Address = S.Address, T.Age = S.AgeWHEN NOT MATCHED THENINSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age)WHEN MATCHED THENDELETE;

D. MERGE Student.CurrentStudents AS T USING Student.NewYearRoster AS SON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED THENDELETEWHEN NOT MATCHED THENINSERT (LastName, FirstName, Address, Age) VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN UPDATE SET Address = T.Address, Age = T.Age;

---

**Answer: A**

---

**Question: 137**

You have the following rows in the Customer Table: CustomerId Status-----1

Active2

Active3

Inactive4

NULL5

Dormant6

Dormant You write the following query to return all customers that do not have NULL or

'Dormant' for their status: SELECT \*FROM

CustomerWHERE Status NOT IN (NULL, 'Dormant')You need to identify the results of the query.Which result should you expect?

- A. CustomerId Status-----1 Active2 Active3 Inactive4 NULL
- B. CustomerId Status-----1 Active2 Active3 Inactive4 NULL5 Dormant6 Dormant
- C. CustomerId Status-----
- D. CustomerId Status-----1 Active2 Active3 Inactive

**Answer: C**

### Question: 138

You are the database developer for a customer service application. The database has the following table and view. CREATE TABLE Tickets (

TicketId int,

Department varchar(200),

Status varchar(200),

IsArchived bit); GO CREATE VIEW SupportTickets (

TicketId,

Department,

Status,

IsArchived)AS SELECT

t.TicketId,

t.Department,

t.Status,

t.IsArchived FROM Tickets t WHERE t.Department = 'support'

AND t.IsArchived = 0 WITH CHECK OPTION; GO INSERT INTO Tickets VALUES (1, 'support', 'open',

0), (2, 'support', 'open', 0), (3, 'support', 'open', 0); GO You are assigned to review another

developer's code. You need to verify that the entire code is functional. Which Transact-SQL statements will fail?

- A. INSERT INTO Tickets VALUES (4, 'sales', 'open', 0), (5, 'support', 'open', 0), (6, 'support', 'open', 1)
- B. UPDATE Tickets SET IsArchived = 1 WHERE TicketId IN (1, 2, 3)
- C. UPDATE SupportTickets SET IsArchived = 1 WHERE TicketId IN (1, 2, 3)
- D. UPDATE SupportTickets SET Status = 'closed' WHERE TicketId IN (1, 2, 3)
- E. INSERT INTO SupportTickets VALUES (4, 'support', 'open', 0), (5, 'support', 'in progress', 0), (6, 'support', 'closed', 0)
- F. INSERT INTO Tickets VALUES (4, 'support', 'open', 0), (5, 'support', 'in progress', 0), (6, 'support', 'closed', 0)
- G. INSERT INTO SupportTickets VALUES (4, 'support', 'open', 1), (5, 'support', 'in progress', 1), (6, 'support', 'closed', 1)
- H. DELETE FROM SupportTickets WHERE TicketId IN (1, 2, 3)

**Answer: C OR G**

### Question: 139

You administer a database named Contoso running on a Microsoft SQL Server 2008 R2 instance. You plan to implement custom error handling in your application. You need to implement custom error handling that meets the following requirements:

- The custom message is a reusable user-defined error message.
- The message returned to the application is an informational message that returns status information or error that is not severe.
- The custom message returned indicates that an error has occurred in the current database and current session. 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.)

- A. DECLARE @ProcessID INT; DECLARE @ProcessName NVARCHAR(150); SET @ProcessID = @@SPID; SET @ProcessName = DB\_NAME();
- B. DECLARE @ProcessID INT; DECLARE @ProcessName INT; SET @ProcessID = @@SPID; SET @ProcessName = @@DB\_NAME;
- C. EXECUTE sp\_addmessage 50025, 10, N'Session ID: %d returned an error in the database: %s.'
- D. EXECUTE sp\_addmessage 50000, 16, N'Session ID: %d returned an error in the database: %s.'
- E. RAISERROR (50000, @ProcessID, @ProcessName, 10, 1)
- F. RAISERROR (50025, 10, 1, @ProcessID, @ProcessName)
- G. RAISERROR (50000, @ProcessID, @ProcessName, 16, 1)
- H. RAISERROR(50025, 10, 1, @@SPID, DB\_NAME() )

**Answer: (C BEFORE A) AND  
(A BEFORE F) AND ONLY (C,  
A, F)**

#### Question: 140

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance. You create tables named order, customer, and product as follows: CREATE TABLE [dbo].[order] ([OrderID] [int],

[ProductID] [int],

[CustomerID] [int],

[OrderDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],

[CustomerName] [varchar](100),

[Address] [varchar](200),

[City] [varchar](100),

[State] [varchar](50),

[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],

[ProductName] [varchar](100),

[SalePrice] [money],

[ManufacturerName] [varchar](100)); You need to write a query to sum the sales made to each customer by the following entries:

- The Customer name and product name
- The grand total of all sales Which SQL query should you use?

- A. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, DATEPART(mm, o.OrderDate);`
- B. `SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID WHERE COUNT(o.OrderID) > 10 GROUP BY c.CustomerName;`
- C. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500`
- D. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());`
- E. `SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());`
- F. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE;`
- G. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, o.OrderDate));`
- H. `SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' AND AVG(p.SalePrice) >= 500`
- I. `SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY p.ProductName, OrderMonth;`
- J. `SELECT c.CustomerName, COUNT(o.OrderID) AS Orders FROM customer c INNER JOIN [order] o ON c.CustomerID = o.CustomerID GROUP BY c.CustomerName HAVING COUNT(o.OrderID) > 10;`

**Answer: D**

### Question: 141

You have a table named Sales.SalesOrderHeader and a table named Person.Person. You are tasked to write a query that returns SalesOrderID and SalesPersonName that have an OrderDate greater than 20040101. SalesPersonName should be made up by concatenating the columns named FirstName and LastName from the table named Person.Person. You need to write a query to return data, sorted in alphabetical order, by the concatenation of FirstName and LastName. Which Transact-SQL statement should you use?

A. `SELECT SalesOrderID, FirstName + ' ' + LastName as SalesPersonName FROM Sales.SalesOrderHeader H JOIN Person.Person P ON P.BusinessEntityID = H.SalesPersonID WHERE OrderDate > '20040101' ORDER BY SalesPersonName DESC`

B. `SELECT SalesOrderID, FirstName + ' ' + LastName as SalesPersonName FROM Sales.SalesOrderHeader H JOIN Person.Person P ON P.BusinessEntityID = H.SalesPersonID WHERE OrderDate > '20040101' ORDER BY FirstName ASC, LastName ASC`

C. SELECT SalesOrderID, FirstName + ' ' + LastName as SalesPersonName FROM  
Sales.SalesOrderHeader H JOIN Person.Person P on P.BusinessEntityID = H.SalesPersonID WHERE  
OrderDate > '20040101' ORDER BY SalesPersonName ASC

D. SELECT SalesOrderID, FirstName + ' ' + LastName as SalesPersonName FROM  
Sales.SalesOrderHeader H JOIN Person.Person P on P.BusinessEntityID = H.SalesPersonID WHERE  
OrderDate > '20040101' ORDER BY FirstName DESC, LastName DESC

---

**Answer: C**

---

**Question: 142**

Note: This question is part of a series of questions that use the same set of answer choices. An answer choice may be correct for more than one question in the series. You are a developer for a Microsoft SQL Server 2008 R2 database instance. You create tables named order, customer, and product as follows: CREATE TABLE [dbo].[order] ([OrderID] [int],

[ProductID] [int],

[CustomerID] [int],

[OrderDate] [datetime]); CREATE TABLE [dbo].[customer] ([CustomerID] [int],

[CustomerName] [varchar](100),

[Address] [varchar](200),

[City] [varchar](100),

[State] [varchar](50),

[ZipCode] [varchar](5)); CREATE TABLE [dbo].[product] ([ProductID] [int],

[ProductName] [varchar](100),

[SalePrice] [money],

[ManufacturerName] [varchar](100)); You need to write a query to sum the sales made to each customer by the following entries:

·Each customer name

·Each product name

·The grand total of all sales Which SQL query should you use?

A. SELECT p.ProductName, DATEPART(mm, o.OrderDate) OrderMonth, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID GROUP BY CUBE(p.ProductName, DATEPART(mm, o.OrderDate));

B. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());

C. SELECT c.CustomerName, p.ProductName, SUM(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());

D. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID = c.CustomerID WHERE o.OrderDate > '09/01/2011' GROUP BY c.CustomerName HAVING AVG(p.SalePrice) >= 500

E. SELECT c.CustomerName, AVG(p.SalePrice) AS Sales FROM product p INNER JOIN [order] o ON p.ProductID = o.ProductID INNER JOIN customer c ON o.CustomerID =

c.CustomerIDWHEREo.OrderDate > '09/01/2011' ANDAVG(p.SalePrice) >= 500  
F. SELECTc.CustomerName,COUNT(o.OrderID) AS OrdersFROMcustomer c INNER JOIN[order] o  
ON c.CustomerID = o.CustomerIDWHERECOUNT(o.OrderID) > 10GROUP BYc.CustomerName;  
G. SELECTp.ProductName,DATEPART(mm, o.OrderDate) OrderMonth,SUM(p.SalePrice) AS  
SalesFROMproduct p INNER JOIN[order] o ON p.ProductID = o.ProductID GROUP BY  
p.ProductName, DATEPART(mm, o.OrderDate);  
H. SELECTp.ProductName,DATEPART(mm, o.OrderDate) OrderMonth,SUM(p.SalePrice) AS  
SalesFROMproduct p INNER JOIN[order] o ON p.ProductID = o.ProductID GROUP BY CUBE;  
I. SELECTp.ProductName,DATEPART(mm, o.OrderDate) OrderMonth,SUM(p.SalePrice) AS  
SalesFROMproduct p INNER JOIN[order] o ON p.ProductID = o.ProductID GROUP BY  
p.ProductName, OrderMonth;  
J. SELECTc.CustomerName,COUNT(o.OrderID) AS OrdersFROMcustomer c INNER JOIN[order] o  
ON c.CustomerID = o.CustomerIDGROUP BYc.CustomerNameHAVINGCOUNT(o.OrderID) > 10;

---

**Answer: B**

---

**Question: 143**

You have two tables named Customer and SalesOrder. In the Customer table you have 1000 customers, of which 900 customers have orders in the SalesOrder table. You execute the following query to list all customers that have had at least one sale.

SELECT \*FROM

CustomerWHERE Customer.CustomerID IN (SELECT SalesOrder.CustomerID  
FROM SalesOrder) You need to identify the results of the query. Which results will the query return?

- A. No rows
- B. The 900 rows in the Customer table with matching rows in the SalesOrder table
- C. The 1000 rows in the Customer table
- D. A warning message

---

**Answer: B**

---

**Question: 144**

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. CREATE EXTERNAL ACCESS ASSEMBLY
- B. SELECT
- C. CONTROL
- D. VIEW SERVER STATE

---

**Answer: D**

---

**Question: 145**

You have created an assembly that utilizes unmanaged code to access external resources. You need to deploy the assembly with the appropriate permissions. Which permission set should you use?

- A. UNSAFE
- B. EXTERNAL\_ACCESS
- C. SAFE
- D. Default permission set

---

**Answer: A**

---