DUMPSARENA

Querying Microsoft SQL Server 2012

Microsoft 70-461

Version Demo

Total Demo Questions: 15

Total Premium Questions: 254

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

You administer a Microsoft SQL Server database that has Trustworthy set to On. You create a stored procedure that returns database-level information from Dynamic Management Views. You grant User1 access to execute the stored procedure. You need to ensure that the stored procedure returns the required information when User1 executes the stored procedure. You need to achieve this goal by granting the minimum permissions required. What should you do? (Each correct answer presents a complete solution. Choose all that apply.)

A. Create a SQL Server login that has VIEW SERVER STATE permissions. Create an application role and a secured password for the role.

B. Modify the stored procedure to include the EXECUTE AS OWNER statement. Grant VIEW SERVER STATE permissions to the owner of the stored procedure.

C. Create a SQL Server login that has VIEW SERVER STATE permissions. Modify the stored procedure to include the EXECUTE AS {newlogin} statement.

D. Grant the db_owner role on the database to User1.

E. Grant the sysadmin role on the database to User1.

ANSWER: D E

QUESTION NO: 2

You develop a Microsoft SQL Server database.

You need to create and call a stored procedure that meets the following requirements:

- Accepts a single input parameter for CustomerID.
- Returns a single integer to the calling application.

Which two Transact-SQL statements should you use? Each correct answer presents part of the solution.

A.
```
CREATE PROCEDURE dbo.GetCustomerRating
@CustomerID INT,
@CustomerRating INT OUTPUT
AS
SET NOCOUNT ON
SELECT @CustomerRating = CustomerOrders/CustomerValue
FROM Customers
WHERE CustomerID = @CustomerID
RETURN
GO
```

B.
```
EXECUTE dbo.GetCustomerRating 1745
```

C.
```
DECLARE @CustomerRatingByCustomer INT
DECLARE @Result INT
EXECUTE @Result = dbo.GetCustomerRating
```

1745,
@CustomerRatingByCustomer

D. CREATE PROCEDURE dbo.GetCustomerRating
@CustomerID INT,
@CustomerRating INT OUTPUT
AS
SET NOCOUNT ON
SELECT @Result = CustomerOrders/CustomerValue
FROM Customers
WHERE CustomerID = @CustomerID
RETURN @Result
GO

E. DECLARE @CustomerRatingByCustomer INT
EXECUTE dbo.GetCustomerRating
@CustomerID = 1745,
@CustomerRating = @CustomerRatingByCustomer OUTPUT

F. CREATE PROCEDURE dbo.GetCustomerRating
@CustomerID INT
AS
DECLARE @Result INT
SET NOCOUNT ON
SELECT @Result = CustomerOrders/CustomerValue
FROM Customers
WHERE CustomerID = @CustomerID

ANSWER: A E

QUESTION NO: 3 - (SIMULATION)

SIMULATION

The following is a series of questions in which you are required to input one or more lines of code.

To input your response

Type your response into the text entry field in the Answer Area. You may input one or more lines of code. More than one solution may be correct. You will receive credit if your solution matches any of the correct solutions.

To validate code syntax

After entering your code, click the Check Syntax button. This validates code syntax (such as SQL commands) and values (such as table names and variable names) used in your solution. If there are any errors, they will appear in the window next to the Check Syntax button. You may change your code and re-validate the syntax as many times as you want.

Note that Check Syntax does NOT validate whether you have answered the question correctly. It simply validates the accuracy of your syntax.

To view available command keywords

Click the Keywords button to view a list of command keywords. This is a general list provided for reference and is not limited to commands used in the question.

You have a SQL database that contains a table named Products.

You are implementing a stored procedure that retrieves the list of products, performs custom business logic and then retrieve the list of products again.

The custom business logic in the stored procedure does not modify data from the Products table.

The stored procedure contains the following:

```
01
02 GO
03 BEGIN TRANSACTION;
04 GO
05 SELECT *
06    FROM Products
07         WHERE ProductID=123 ;
08 GO
09 ...
10 SELECT *
11    FROM Products
12         WHERE ProductID=123;
13 GO
14 ...
15 COMMIT TRANSACTION ;
16 GO
```

You need to complete line 01 of the stored procedure to ensure that when the transaction occurs, the data read from the SELECT * FROM Products statement on line 05 is identical to the data read from the SELECT * FROM Products statement on line 10. The solution must maximize concurrency.

Part of the correct Transact-SQL has been provided in the answer area below. Enter the code in the answer area that resolves the problem and meets the stated goals or requirements. You can add code within the code that has been provided as well as below it.

```
1 SET TRANSACTION ISOLATION LEVEL ;
```

Keywords

ADD
ALL
ALTER
AND
ANY
AS
ASC
AUTHORIZATION
AVG
BACKUP
BEGIN
BETWEEN
BREAK
BROWSE
BULK
BY
CASCADE
CASE
CAST
CATCH
CHECK
CHECKPOINT
CLOSE
CLUSTERED
COALESCE
COLLATE
COLUMN
COMMIT
COMPUTE
CONSTRAINT
CONTAINS
CONTAINSTABLE
CONTINUE
CONVERT
CREATE
CROSS
CURRENT
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP
CURRENT_USER
CURSOR
DATABASE
DATETIME
DBCC
DEALLOCATE
DECLARE
DEFAULT
DELETE
DENSE_RANK
DENY
DESC
DISK
DISTINCT
DISTRIBUTED
DOUBLE
DROP
DUMP
ELSE
END
ERRLVL
ERROR_NUMBER
ESCAPE
EXCEPT
EXEC
EXECUTE
EXISTS
EXIT
EXTERNAL
FETCH
FILE
FILESTREAM
FILLFACTOR
FOR
FOREIGN
FREETEXT
FREETEXTTABLE
FROM
FULL
FUNCTION
GETDATE
GO
GOTO
GRANT
GROUP
HAVING
HOLDLOCK
IDENTITY
IDENTITY_INSERT
IDENTITYCOL
IF
IFF
IN
INDEX
INNER
INSERT
INT
INTERSECT
INTO
IS
ISNULL
JOIN
KEY
KILL
LEFT
LIKE
LINENO
LOAD
MAX
MERGE
NATIONAL
NOCHECK
NONCLUSTERED
NOT
NULL
NULLIF
OF
OFF
OFFSETS
ON
OPEN
OPENDATASOURCE
OPENQUERY
OPENROWSET
OPENXML
OPTION
OR
ORDER
OUTER
OVER
PERCENT
PERSISTED
PIVOT
PLAN
PRECISION
PRIMARY
PRINT
PROC
PROCEDURE
PUBLIC
RAISERROR
RANK
READ
READTEXT
RECONFIGURE
REFERENCES
REPEATABLE
REPLICATION
RESTORE
RESTRICT
RETURN
RETURNS
REVERT
REVOKE
RIGHT
ROLLBACK
ROWCOUNT
ROW_NUMBER
ROWGUIDCOL
RULE
SAVE
SCHEMA
SCHEMABINDING
SECURITYAUDIT
SELECT
SEMANTICKEYPHRASETABLE
SEMANTICSIMILARITYDETAILSTABLE
SEMANTICSIMILARITYTABLE
SESSION_USER
SET
SETUSER
SHUTDOWN
SNAPSHOT
SOME
STATISTICS
SYSTEM_USER
TABLE
TABLESAMPLE
TEXTSIZE
THEN
TO
TOP
TRAN
TRANSACTION
TRIGGER
TRUNCATE
TRY
TRY_CONVERT
TSEQUAL
UNION
UNIQUE
UNPIVOT
UPDATE
UPDATETEXT
USE
USER
VALUES
VARYING
VIEW
WAITFOR
WHEN
WHERE
WHILE
WITH
WITHIN GROUP
WRITETEXT
XML

ANSWER: Please review the explanation part for this answer.

Explanation:

Add REPEATABLE READ to line 1 to get:

SET TRANSACTIONISOLATION LEVEL REPEATABLE READ;

REPEATABLE READ specifies that statements cannot read data that has been modified but not yet committed by other transactions and that no other transactions can modify data that has been read by the current transaction until the current transaction completes.

Incorrect Answers:

READ UNCOMMITTED specifies that statements can read rows that have been modified by other transactions but not yet committed.

READ COMMITTED specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads. Data can be changed by other transactions between individual statements within the current transaction, resulting in nonrepeatable reads or phantom data. References: https://msdn.microsoft.com/en-us/library/ms173763.aspx

QUESTION NO: 4

You are working with a table that has an XML column that contains information about books. Each book may have an associated price.

You need to write a query that returns each author on a separate row in XML format.

Which XML method should you use?

A. Value()

B. Nodes()

C. Query()

D. Exist()

ANSWER: B

Explanation:

The nodes() method is useful when you want to shred an xml data type instance into relational data. It allows you to identify nodes that will be mapped into a new row.

The result of the nodes() method is a rowset that contains logical copies of the original XML instances. In these logical copies, the context node of every row instance is set to one of the nodes identified with the query expression, so that subsequent queries can navigate relative to these context nodes.

Incorrect Answers:

A: The Value() method performs an XQuery against the XML and returns a value of SQL type. This method returns a scalar value.

C: The Query() method specifies an XQuery against an instance of the xml data type. The result is of xml type. The method returns an instance of untyped XML.

D: The Exists() method returns a bit that represents one of the following conditions:

1, representing True, if the XQuery expression in a query returns a nonempty result. That is, it returns at least one XML node.

0, representing False, if it returns an empty result.

NULL if the xml data type instance against which the query was executed contains NULL.

References: https://docs.microsoft.com/en-us/sql/t-sql/xml/nodes-method-xml-data-type?view=sql-server-2017

QUESTION NO: 5

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

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

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

A. SELECT CustomerID FROM Customer
WHERE TerritoryID <> SOME(SELECT TerritoryID FROM Salesperson)

B. SELECT CustomerID FROM Customer
WHERE TerritoryID <> ALL(SELECT TerritoryID FROM Salesperson)

C. SELECT CustomerID FROM Customer
WHERE TerritoryID <> ANY(SELECT TerritoryID FROM Salesperson)

D. SELECT CustomerID FROMCustomer
WHERE TerritoryID NOT IN(SELECT TerritoryID FROM Salesperson)

ANSWER: B D

QUESTION NO: 6 - (SIMULATION)

SIMULATION

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

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

You need to create a stored procedure named usp_Customers for App1 that returns only the number of rows specified by the @Count parameter.

The solution must NOT use BEGIN, END, or DECLARE statements.

Part of the correct Transact-SQL statement has been provided in the answer area. Complete the Transact-SQL statement

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

ANSWER: Please review the explanation part for this answer.

Explanation:

SELECT TOP(@Count) LastName

Change the second line to: SELECT TOP(@Count) LastName

The complete code is:

CREATE PROCEDURE usp_Customers @Count int

SELECT TOP(@Count) LastName

FROM Customers

ORDER BY Customers.LastName

QUESTION NO: 7

You have a vendor application that uses a scalar function.

You discover that the queries for the application run slower than expected.

You need to gather the runtime information of the scalar function.

What are two possible ways to achieve this goal? Each correct answer presents a complete solution.

A. Enable the Include Actual Execution Plan option.

B. Enable the Display Estimated Execution Planoption.

C. Create and then enable a profiler trace.

D. Create and then enable an extended events trace.

E. Run the Database Engine Tuning Advisor.

ANSWER: A D

Explanation:

A: An execution plan is the result of the query optimizer's attempt to calculate the mostefficient way to implement the request represented by the T-SQL query you submitted. To generate the first execution plan, you can enable the Include Actual Execution Plan option.

D: SQL Server Extended Events can be used to capture User Defined Function(UDF) counts

References:https://www.mssqltips.com/sqlservertip/4100/how-to-find-udfs-causing-sql-server-performance-issues/

QUESTION NO: 8 - (DRAG DROP)

DRAG DROP

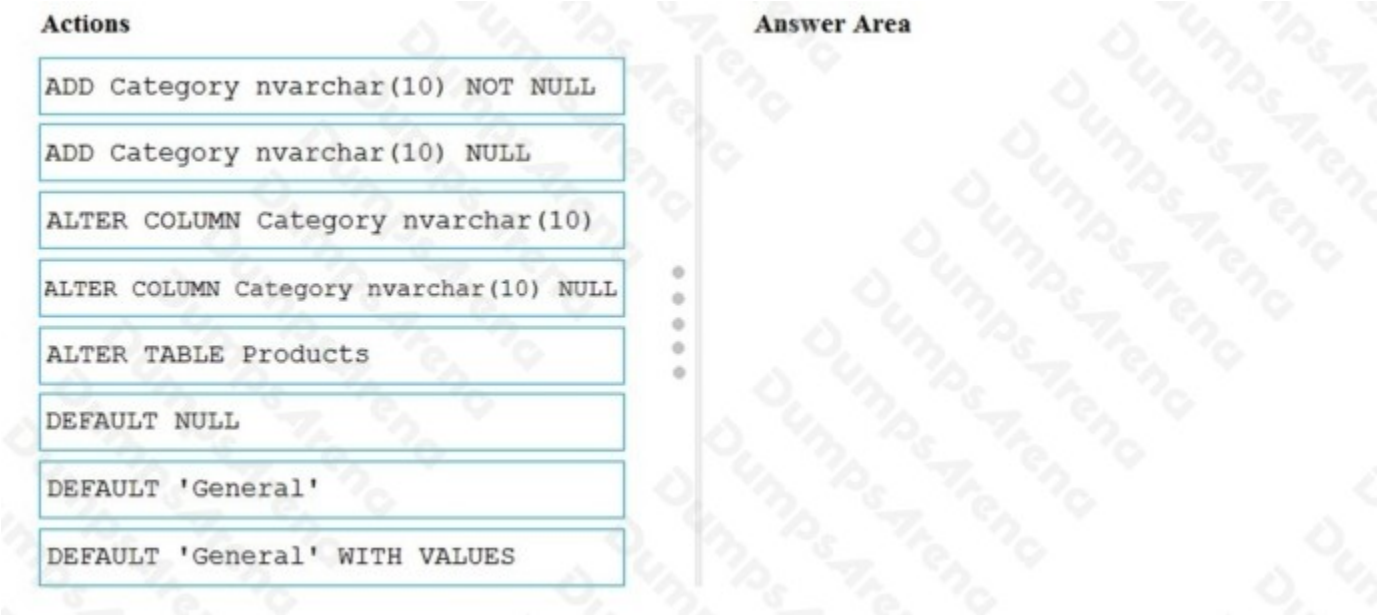
You use a Microsoft Azure SQL Database instance named ContosoDb. ContosoDb contains a table named Products that has existing records.

The Products table has columns named Code and QuantityOnHand.

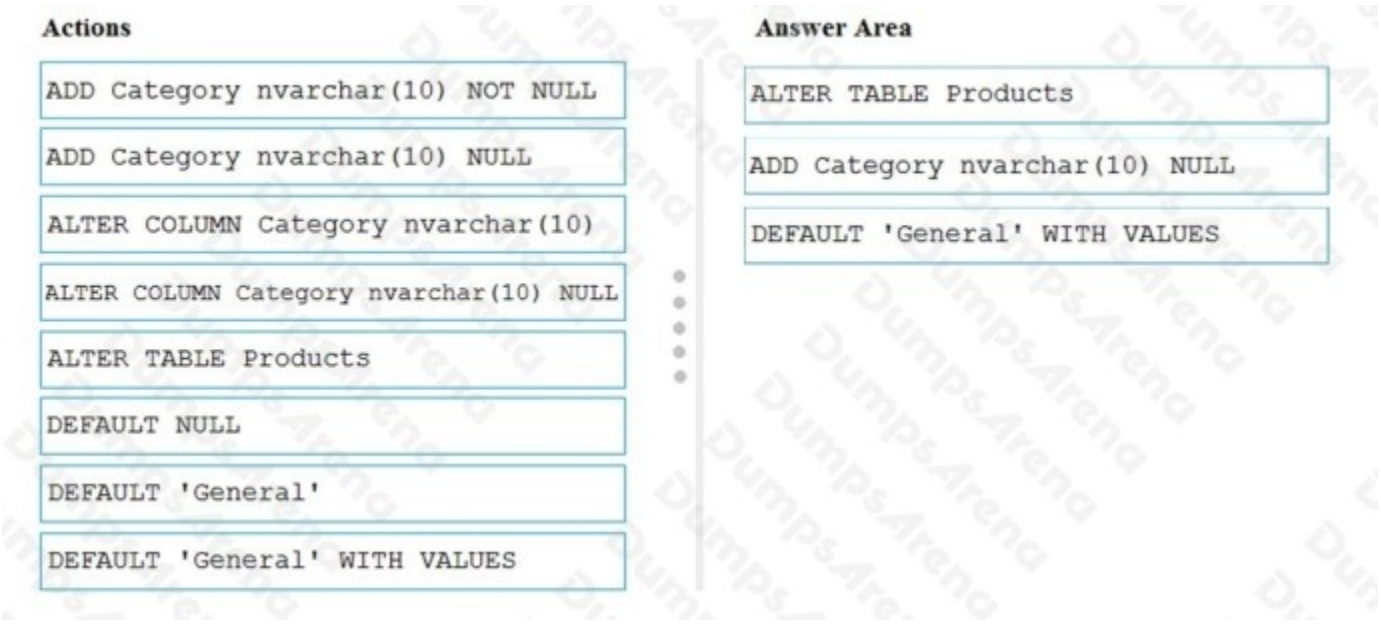
You need to create a new column in the Products table named Category that allows null values and sets the value of the Category column to General for all existing records.

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

Select and Place:



ANSWER:



Explanation:

NULL or NOT NULL specifies whether the column can accept null values. Columns that do not allow null values can be added with ALTER TABLE only if they have a default specified or if the table is empty. NOT NULL can be specified for computed columns only if PERSISTED is also specified. If the new column allows null values and no default is specified, the new column contains a null value for each row in the table. If the new column allows null values and a default definition is added with the new column, WITH VALUES can be used to store the default value in the new column for each existing row in the table.

Reference:

https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-table-transact-sql

QUESTION NO: 9

You are designing a Microsoft SQL Server database named Orders.

You create a table by running the following Transact-SQL statement:

```
CREATE TABLE Customers
(CustomerID int UNIQUE NOT NULL,
 UserName varchar(50) UNIQUE NOT NULL,
 CustomerName varchar(255) NOT NULL,
 Address varchar(255) NOT NULL,
 City varchar(50) NOT NULL,
 State char(2) NOT NULL,
 Phone char(12) NOT NULL)
```

You create a stored procedure to be used by an ASP.NET application that runs the following statement:

```
SELECT CustomerID,
  CustomerName,
  Address,
  City,
  State,
  Phone,
FROM Customers
WHERE UserName = @name
```

You need to ensure that the query runs as efficiently as possible.

Which Transact-SQL statement should you run?

A.
```
CREATE NONCLUSTERED INDEX IX_Customers
ON Customers (UserName)
```

B.
```
ALTER TABLE Customers
ADD CONSTRAINT IX_Customers PRIMARY KEY CLUSTERED (CustomerID)
```

C.
```
CREATE NONCLUSTERED INDEX IX_Customers
ON CUSTOMERS (Customer ID)
```

D.
```
CREATE CLUSTERED INDEX IX_Customers
ON Customers (UserName)
```

A. Option A

B. Option B

C. Option C

D. Option D

ANSWER: B

Explanation:

Creating a primary key automatically creates a corresponding unique clustered index, or a nonclustered index if specified as such.

Example: To create a primary key in an existing table

The following example creates a primary key on the column TransactionID in the AdventureWorks database.

ALTER TABLE Production.TransactionHistoryArchive

ADD CONSTRAINT PK_TransactionHistoryArchive_TransactionID PRIMARY KEY CLUSTERED (TransactionID);

References:

https://docs.microsoft.com/en-us/sql/relational-databases/tables/create-primary-keys

QUESTION NO: 10

You administer a Microsoft SQL Server database that contains a table named OrderDetail. You discover that the NCI_OrderDetail_CustomerID non-clustered index is fragmented. You need to reduce fragmentation.

You need to achieve this goal without taking the index offline. Which Transact-SQL batch should you use?

A. CREATE INDEX NCI_OrderDetail_CustomerID ON OrderDetail.CustomerID WITH DROP EXISTING

B. ALTER INDEX NCI_OrderDetail_CustomerID ON OrderDetail.CustomerID REORGANIZE

C. ALTER INDEX ALL ON OrderDetail REBUILD

D. ALTER INDEX NCI_OrderDetail_CustomerID ON OrderDetail.CustomerID REBUILD

ANSWER: B

Explanation:

Reference: http://msdn.microsoft.com/en-us/library/ms188388.aspx

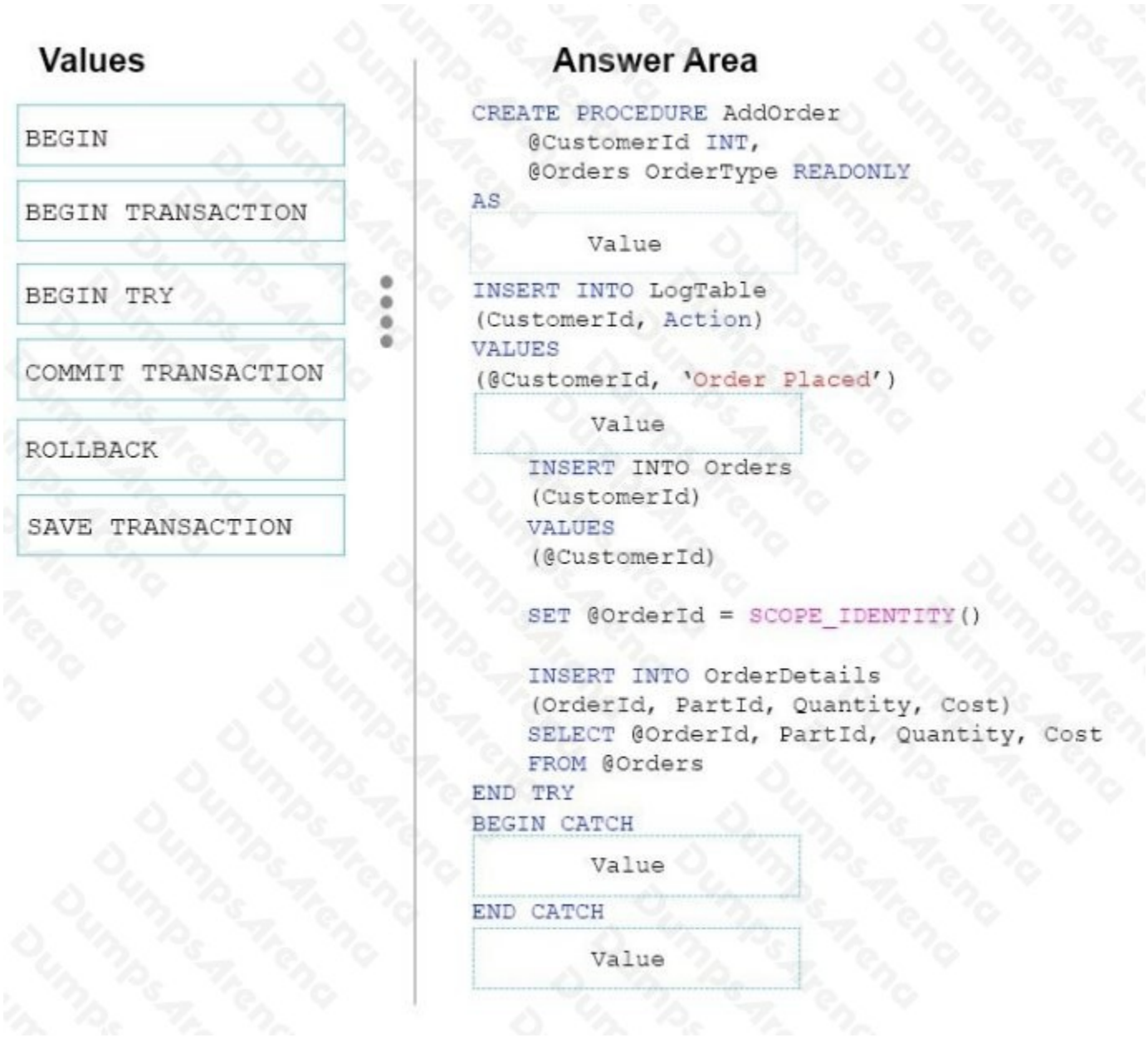
QUESTION NO: 11 - (DRAG DROP)

DRAG DROP

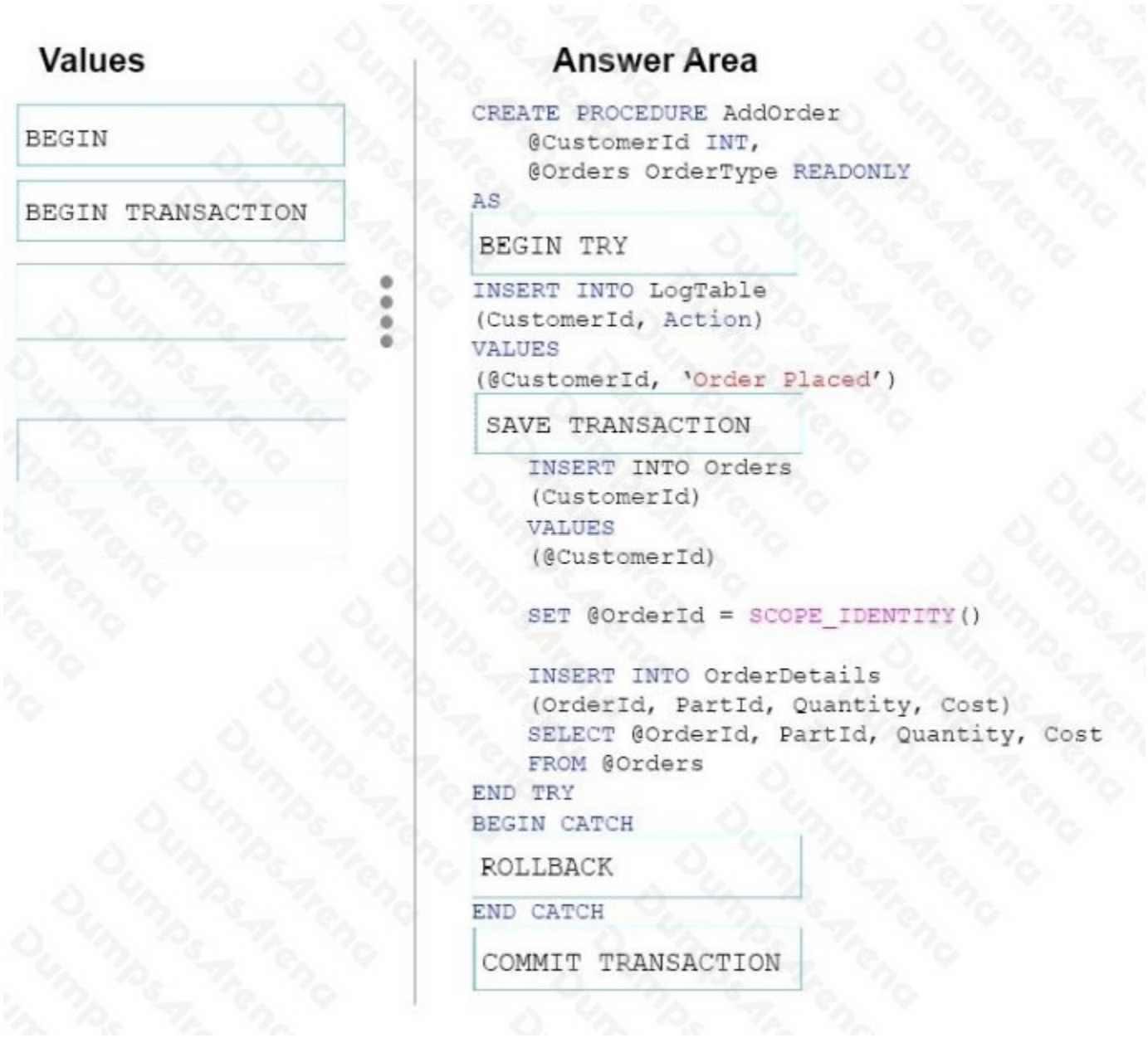
You need to create a stored procedure that enters values into multiple tables. The solution must ensure that if a single insert fails, none of the values are inserted into the tables.

How should you complete the stored procedure? To answer, drag the appropriate values to the correct locations. Each value may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Select and Place:


Values
BEGIN
BEGIN TRANSACTION
BEGIN TRY
COMMIT TRANSACTION
ROLLBACK
SAVE TRANSACTION
Answer Area
CREATE PROCEDURE AddOrder
@CustomerId INT,
@Orders OrderType READONLY
AS
Value
INSERT INTO LogTable
(CustomerId, Action)
VALUES
(@CustomerId, 'Order Placed')
Value
INSERT INTO Orders
(CustomerId)
VALUES
(@CustomerId)
SET @OrderId = SCOPE_IDENTITY()
INSERT INTO OrderDetails
(OrderId, PartId, Quantity, Cost)
SELECT @OrderId, PartId, Quantity, Cost
FROM @Orders
END TRY
BEGIN CATCH
Value
END CATCH
Value

ANSWER:



Explanation:

References: https://msdn.microsoft.com/en-us/library/ms188378.aspx

QUESTION NO: 12

You have several SQL Server queries.

You plan to optimize the queries to improve performance.

You run the queries in SQL Server Management Studio.

You need to compare query runs to expose the indexing issues of the queries.

Which three actions should you perform from SQL Server Management Studio? Each correct answer presents part of the solution.

A. Enable the Debug option.

B. Add the STATISTICS TIME execution setting to the query.

C. Add the STATISTICS IO execution setting to the query.

D. Add the STATISTICS PROFILE execution setting to the query.

E. Enable the Include Actual Execution Plan option.

ANSWER: B C E

Explanation:

E: An execution plan is theresult of the query optimizer's attempt to calculate the most efficient way to implement the request represented by the T-SQL query you submitted. To generate the first execution plan, you can enable the Include Actual Execution Plan option.

B: SET STATISTICS TIME displays the number of milliseconds required to parse, compile, and execute each statement.

C: STATISTICS IO causes SQL Server to display information regarding the amount of disk activity generated by Transact-SQL statements. This is useful information for optimizing queries.

The information include Scan count:

Number of seeks/scans started after reaching the leaf level in any direction to retrieve all the values to construct the final dataset for the output.

Scan count is 0 if the index used is a unique index or clustered index on a primary key and you are seeking for only one value. For example WHERE Primary_Key_Column = .

Scant count is 1 when you are searching for one value using a non-unique clustered index which is defined on anon-primary key column. This is done to check for duplicate values for the keyvalue that you are searching for. For example WHERE Clustered_Index_Key_Column = .

Scan count is N when N is the number of different seek/scan started towards the leftor right side at the leaf level after locating a key value using the index key.

Incorrect Answers:

A: The Debug option is used the find programming errors, and is not used to increase performance.

D: Graphical Plans are quick and easy to read but the detailed data for the plan is masked. Both Estimated and Actual execution plans can be viewed in graphical format. Text plans are a bit harder to read, but more information is immediately available. There are three text plan formats:

- SHOWPLAN_ALL: a reasonably complete set of data showing the Estimated execution plan for the query

- SHOWPLAN_TEXT: provides a very limited set of data for use with tools like osql.exe. It too only shows the Estimated execution plan - STATISTICS PROFILE: similar to SHOWPLAN_ALL except it represents the data for the Actual execution plan

References: https://www.simple-talk.com/sql/performance/execution-plan-basics/ https://msdn.microsoft.com/en-us/library/ms184361.aspx

QUESTION NO: 13 - (HOTSPOT)

HOTSPOT

You are developing an SQL Server database. The database contains two tables and a function that are defined by the following Transact-SQL statements.

```
CREATE TABLE [dbo].[SalesOrderDetail](
    [SalesOrderID] [int] NOT NULL,
    [SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
    [OrderQty] [smallint] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [LineTotal] [numeric](38, 6) NOT NULL,
CONSTRAINT [PK_SalesOrderDetail] PRIMARY KEY CLUSTERED
(
[SalesOrderDetailID] ASC
))

CREATE TABLE [dbo].[SalesOrderHeader](
    [SalesOrderID] [int] IDENTITY(1,1) NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [Status] [tinyint] NOT NULL,
    [PurchaseOrderNumber] [nvarchar](25) NULL,
    [AccountNumber] [nvarchar](15) NULL,
    [CustomerID] [int] NOT NULL,
    [TotalDue] [money] NOT NULL,
CONSTRAINT [PK_SalesOrderHeader] PRIMARY KEY CLUSTERED
(
[SalesOrderID] ASC
))

CREATE FUNCTION TopSellingProducts
(
    @date datetime
)
RETURNS TABLE
AS
RETURN
(
    SELECT TOP 5
        COUNT([SalesOrderDetail].ProductID) [count],
        [SalesOrderDetail].ProductID
    FROM [SalesOrderHeader]
    INNER JOIN [SalesOrderDetail] ON [SalesOrderHeader].[SalesOrderID] = [SalesOrderDetail].[SalesOrderID]
    WHERE [OrderDate] >= dateadd(day,datediff(day,1,@date),0)
      AND [OrderDate] < dateadd(day,datediff(day,0,@date),0)
    GROUP BY [SalesOrderDetail].ProductID
    ORDER BY COUNT ([SalesOrderDetail].ProductID) DESC
)
```

You need to create a query to determine the total number of products that are sold each day for the five top-selling products on that particular day.

How should you complete the Transact-SQL script? To answer, select the appropriate Transact-SQL statements in the answer area.

Hot Area:

Answer Area

[▼]
- JOIN OrderDates (OrderDate)
- WITH OrderDates (OrderDate)
- APPLY OrderDates (OrderDate)
- SELECT OrderDates (OrderDate)

```
AS
(
```

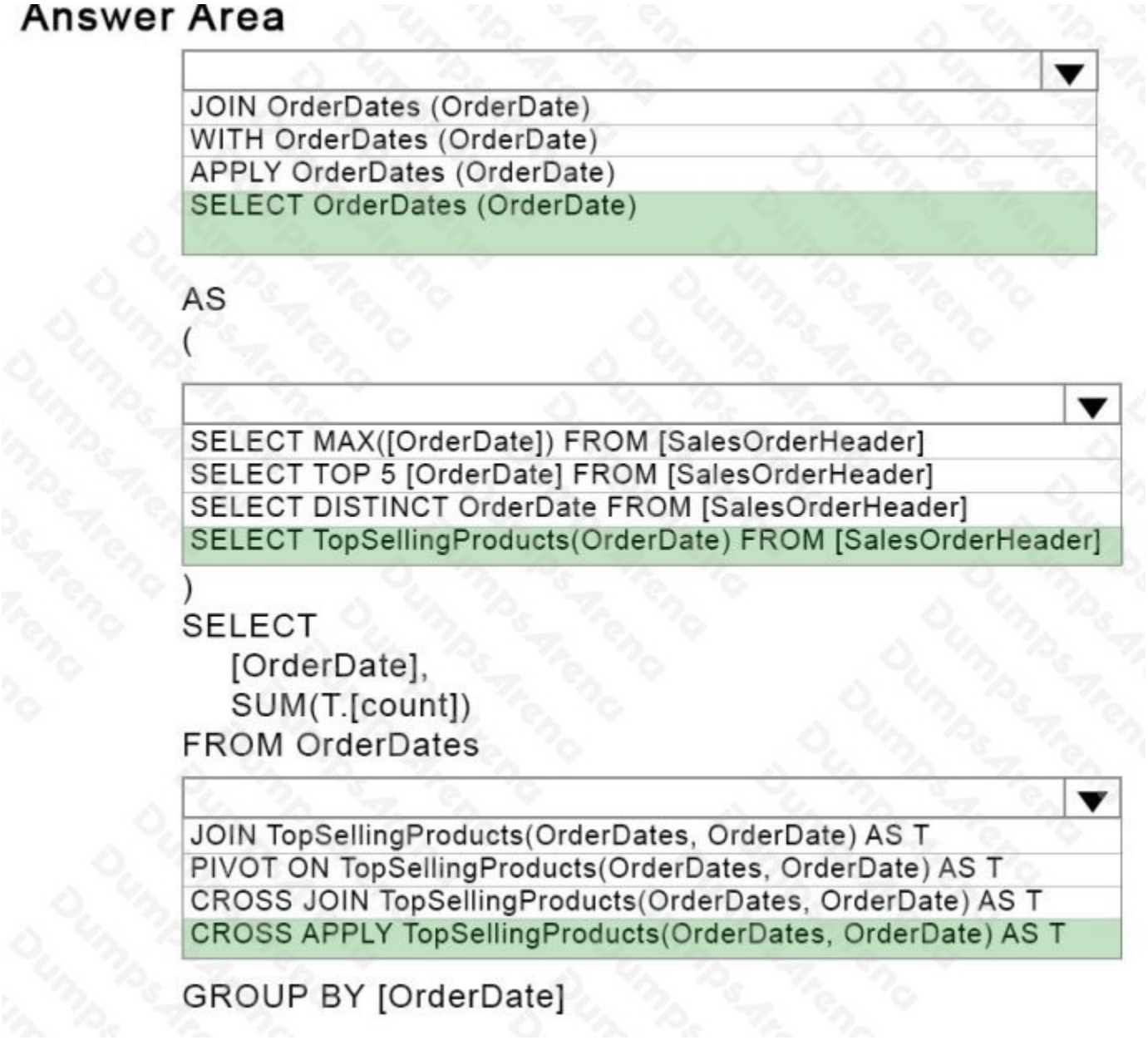
[▼]
- SELECT MAX([OrderDate]) FROM [SalesOrderHeader]
- SELECT TOP 5 [OrderDate] FROM [SalesOrderHeader]
- SELECT DISTINCT OrderDate FROM [SalesOrderHeader]
- SELECT TopSellingProducts(OrderDate) FROM [SalesOrderHeader]

```
)
SELECT
    [OrderDate],
    SUM(T.[count])
FROM OrderDates
```

[▼]
- JOIN TopSellingProducts(OrderDates, OrderDate) AS T
- PIVOT ON TopSellingProducts(OrderDates, OrderDate) AS T
- CROSS JOIN TopSellingProducts(OrderDates, OrderDate) AS T
- CROSS APPLY TopSellingProducts(OrderDates, OrderDate) AS T

```
GROUP BY [OrderDate]
```

ANSWER:



Explanation:

The APPLY operator allows you to invoke a table-valued function for each row returned by an outer table expression of a query. T

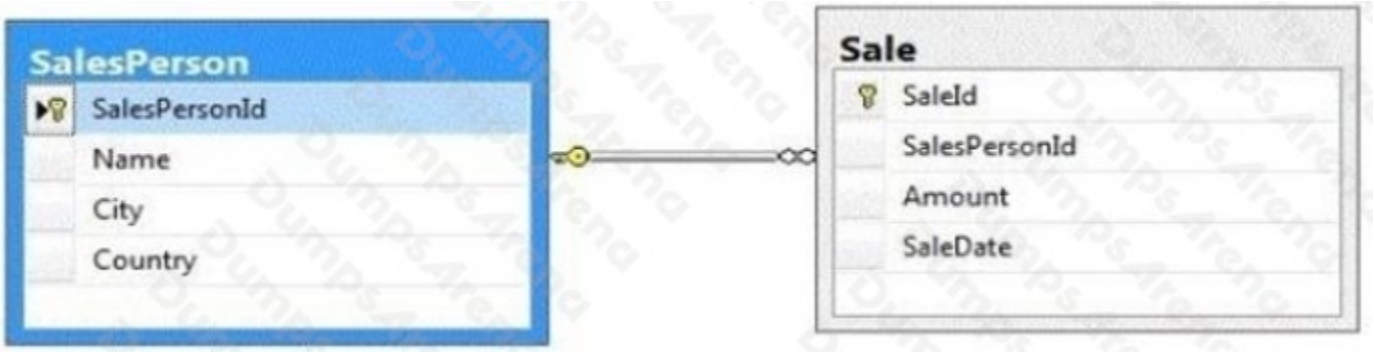
There are two forms of APPLY: CROSS APPLY and OUTER APPLY. CROSS APPLYreturns only rows from the outer table that produce a result set from the table-valued function. OUTER APPLY returns both rows that produce a result set, and rows that do not, with NULL values in the columns produced by the table-valued function.

Reference: Using APPLY

https://technet.microsoft.com/en-us/library/ms175156(v=sql.105).aspx

QUESTION NO: 14

You support a database structure shown in the exhibit.



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

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

Which Transact-SQL query should you use?

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

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

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

D. SELECT SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate), Sum(Amount)
FROM Sale
INNER JOIN SalesPerson
ON Sale.SalesPersonId = SalesPerson.SalesPersonId
GROUP BY SalesPerson.Name, Country, City,
DatePart(yyyy, SaleDate),
GROUPING SETS((Country, City, DatePart(yyyy, SaleDate)),
(Country, City) (Country),())

ANSWER: D

Explanation:

- Total sales made by sales people, year, city, and country is implemented with the following code:

GROUP BY SalesPerson.Name, Country, City,

DatePart(yyyy, SaleDate),

- Sub totals only at the city level and country level

- A grand total of the sales amount is implemented with

GROUPING SETS((Country, City, DatePart(yyyy, SaleDate)), (Country, City) (Country),())

Note: GROUP BY ()

Specifies the empty group which generates the grand total. This is useful as one of the elements of a GROUPING SET. For example, this statement gives the total sales for each country and then gives the grand-total for all countries.

SELECT Country, SUM(Sales) AS TotalSales

FROM Sales

GROUP BY GROUPING SETS (Country, ());

Incorrect Answers:

B, C: GROUP BY ROLLUP creates a group for each combination of column expressions. In addition, it "rolls up" the results into subtotals and grand totals.

D: The GROUPING SETS option gives you the ability to combine multiple GROUP BY clauses into one GROUP BY clause. The results are the equivalent of UNION ALL of the specified groups.

References:

https://docs.microsoft.com/en-us/sql/t-sql/queries/select-group-by-transact-sql

QUESTION NO: 15

You use a Microsoft Azure SQL Database instance. The instance contains a table named Customers that has columns named Id, Name, and IsPriority.

You need to create a view named VwPriorityCustomers that:

- returns rows from Customer that have a value of True in the IsPriority column, and - does not allow columns to be altered or dropped in the underlying table.

Which Transact-SQL statement should you run?

A. CREATE VIEW VwPriorityCustomers
AS
SELECT Id, Name FROM dbo.Customers WHERE IsPriority=1 WITH CHECK OPTION

B. CREATE VIEW VwPriorityCustomers
WITH VIEW_METADATA
AS
SELECT Id, Name FROM dbo.Customers WHERE IsPriority=1

C. CREATE VIEW VwPriorityCustomers
WITH ENCRYPTION
AS
SELECT Id, Name FROM dbo.Customers WHERE IsPriority=1

D. CREATE VIEW VwPriorityCustomers
WITH SCHEMABINDING
AS SELECT Id, Name FROM dbo.Customers WHERE IsPriority=1

ANSWER: D

Explanation:

SCHEMABINDING binds the view to the schema of the underlying table or tables. When SCHEMABINDING is specified, the base table or tables cannot be modified in a way that would affect the view definition.

References: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017