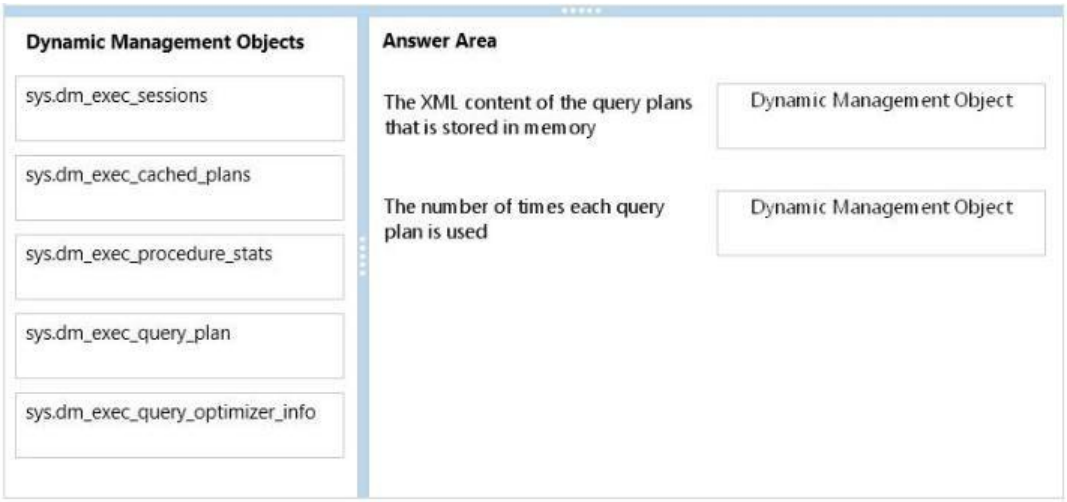
Microsoft

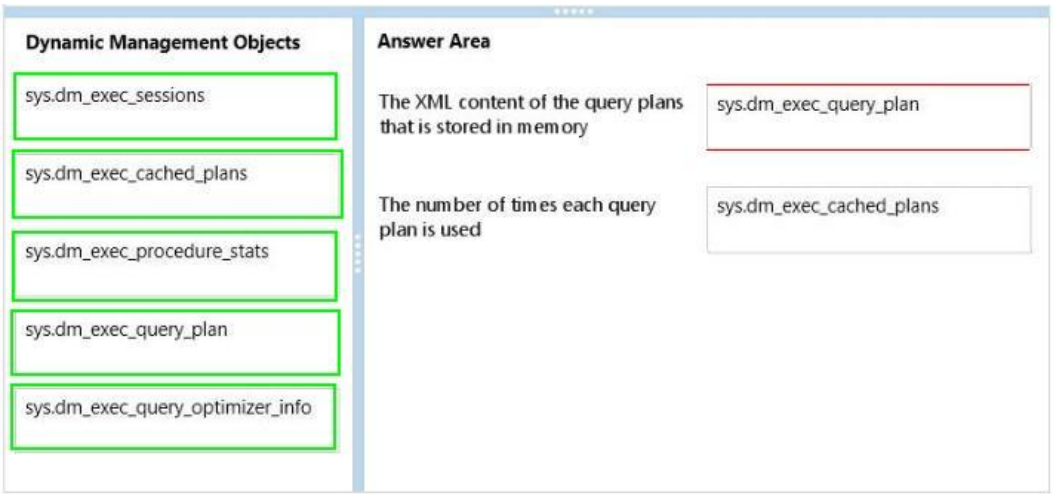
70-469

Recertification for MCSE: Data Platform

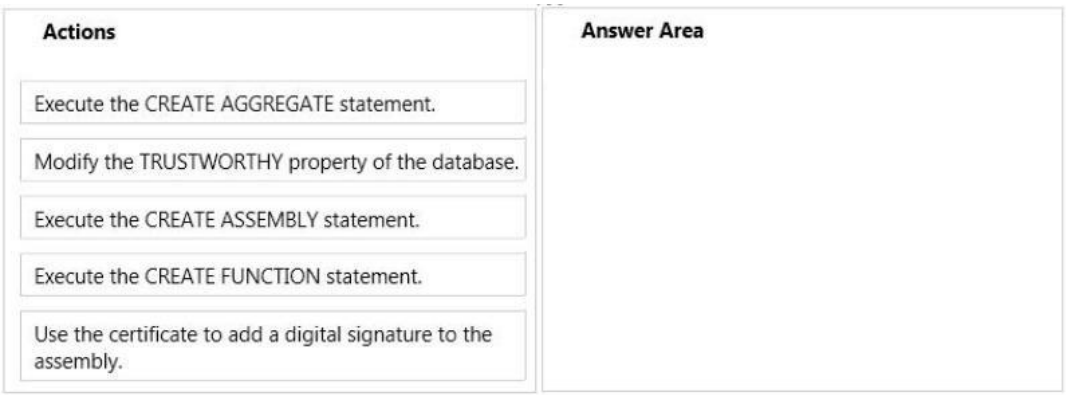
the answer area.



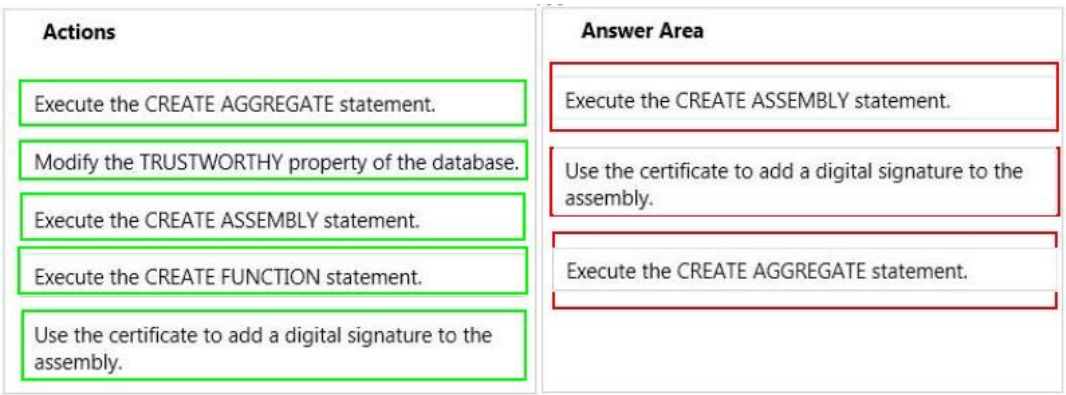
Answer:
Exhibit



QUESTION: 101
DRAG DROP
You plan to create a custom aggregation function named Median.
You plan to deploy Median to a SQL Server 2014 server named Server1.
You need to ensure that Median can access a web service named WebApp1. The solution must minimize the number of changes made to the database.
You create a Microsoft .NET Framework class that contains the definition of Median. You upload a certificate to Server1.
What three tasks should you perform next?
To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



Answer:
Exhibit



QUESTION: 102
You have a Microsoft SQL Azure database named DBAzurel. DBAzurel contains a table named Orders that stores sales data. Each order has a sales total that can only be discovered by querying multiple tables.
You need to ensure that the value of the sales total is returned by executing a query on Orders. What should you create?

A. A calculated column that uses a scalar function
B. A trigger that uses a table-valued function
C. A calculated column that uses a table-valued function
D. A trigger that uses a ranking function

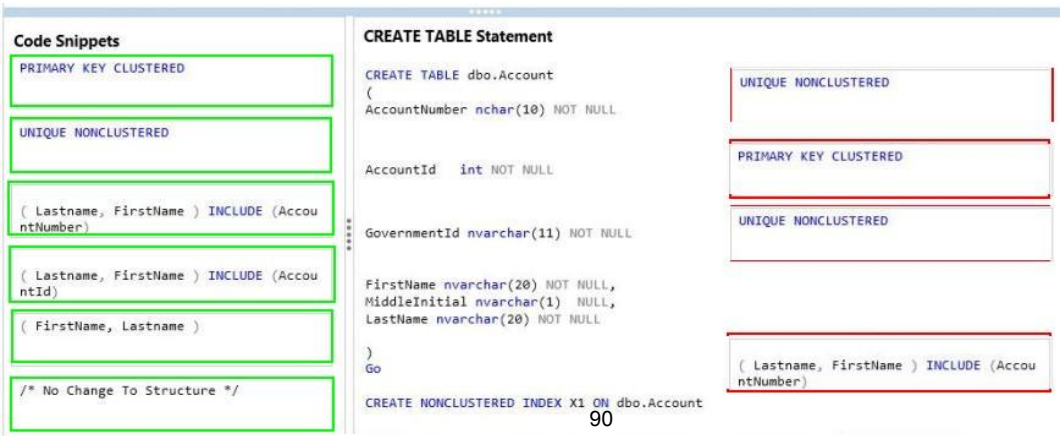
Answer: C

Explanation:
A table-valued parameter is scoped to the stored procedure, function, or dynamic Transact-SQL text, exactly like other parameters. Similarly, a variable of table type has scope like any other local variable that is created by using a DECLARE statement. You can declare table-valued variables within dynamic Transact-SQL statements and pass these variables as table-valued parameters to stored procedures and functions.

Table-valued parameters offer more flexibility and in some cases better performance than temporary tables or other ways to pass a list of parameters.
Incorrect:
Not A: A scalar function would only be able to use other columns from the same table.

QUESTION: 103
DRAG DROP
You administer a SQL Server 2014 instance.
You have been assigned to determine the cause of frequent long-running transactions that have been tracked to the dbo.Account table, where there are many cases of blocking and deadlocks. The dbo.Account table contains more than one million rows.
Users and processes frequently search for and update data by using the AccountId column, and less frequently the AccountNumber and GovernmentId columns, all of which contain only unique values. Users frequently get lists of AccountNumber values by searching on Last Name and then First Name.
You need to modify the structure of the dbo.Account table to alleviate the issues.
How should you complete the table definition to reduce contention on the table structure?
To answer, drag the appropriate code snippets to the correct locations in the CREATE TABLE statement. Each code snippet 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.

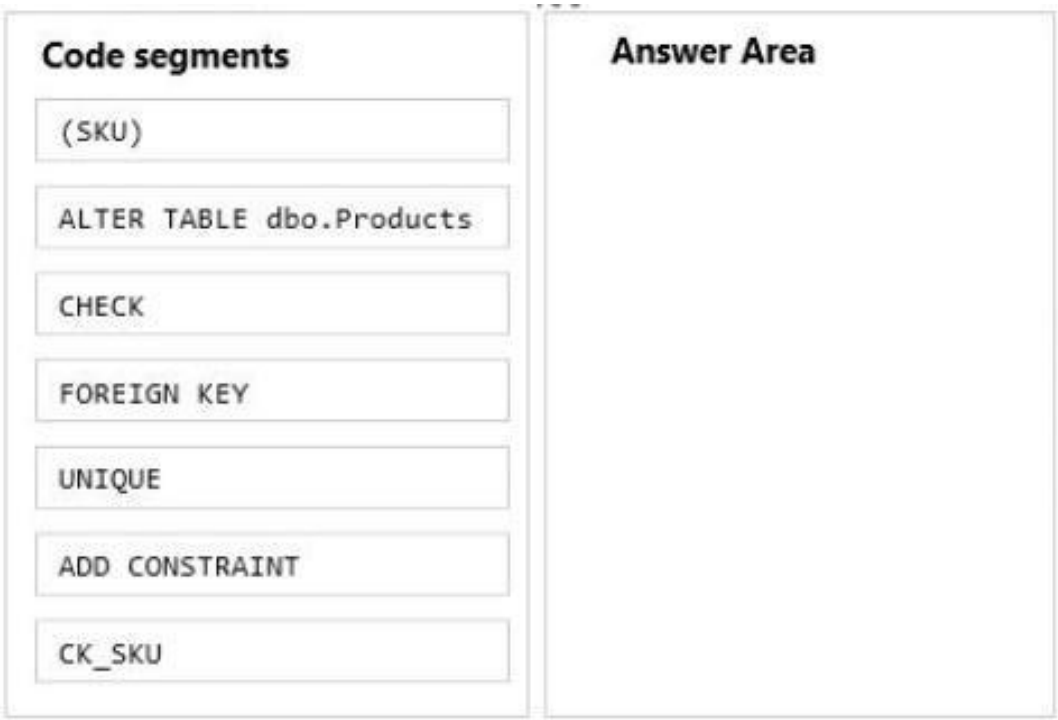
Answer:
Exhibit



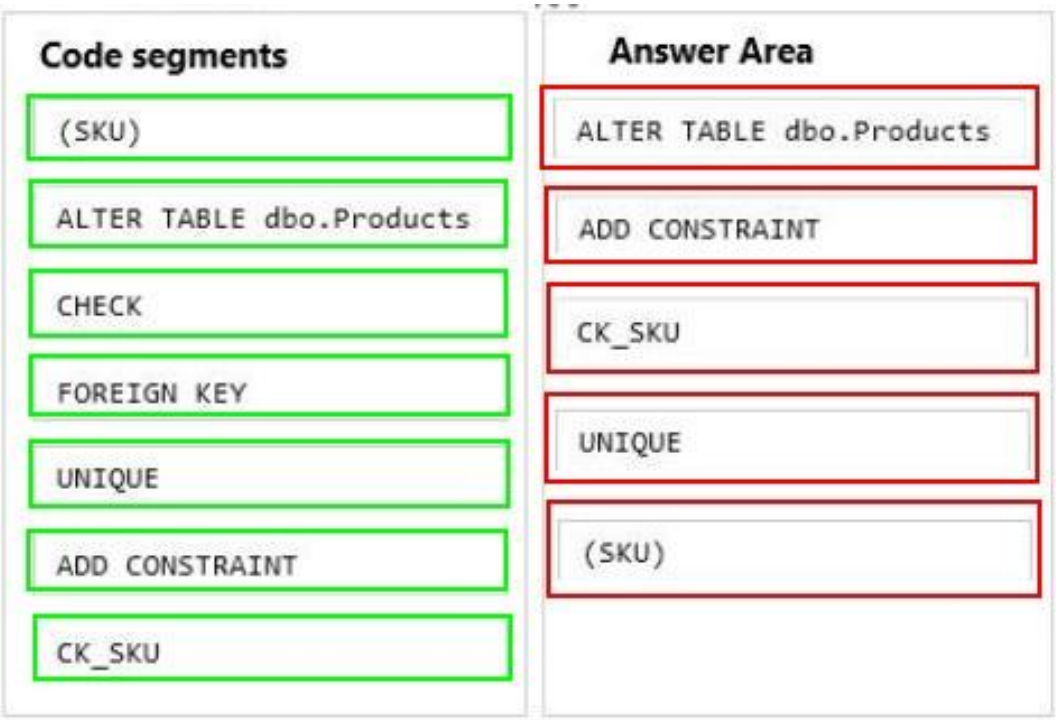
QUESTION: 104
DRAG DROP
You create a disk-based table that contains the following script:

```
CREATE TABLE dbo.Products
(
  ProductId bigint IDENTITY(1,1),
  Name nvarchar(50) NULL,
  Description nvarchar(max) NULL,
  SKU char(10) NULL,
            CONSTRAINT PK_Products PRIMARY KEY CLUSTERED (ProductId)
)  ON [PRIMARY]
GO
```

You need to prevent duplicate values in the SKU field. Which five code segments should you use?
To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.



Answer:
Exhibit



QUESTION: 105
Your network contains a server that has SQL Server 2014 installed. You create a table by using the following script:

```
CREATE TABLE dbo.Products
(
  id int NOT NULL,
  ProductName nvarchar(50) NULL,
  ProductManufacturer nvarchar(50) NULL,
  ProductDescription nvarchar(200) NULL,
    CONSTRAINT PK_Products PRIMARY KEY CLUSTERED (id)
)
  ON [PRIMARY]
GO
```

You need to recommend a solution to ensure that each combination of ProductName and ProductManufacturer is not duplicated. What should you recommend creating?

A. A UNIQUE constraint
B. A filtered index
C. A columnstore index
D. A CHECK constraint

Answer: A

QUESTION: 106
You have a Microsoft SQL Azure database. You have the following stored procedure:

```
01 CREATE PROCEDURE UpdateContact
02   @ContactID int,
03   @LastName nvarchar(50)
04 AS
05
06 SELECT LastName AS OriginalName
07 FROM Person.Contact
08
09 WHERE ContactID = @ContactID;
10 UPDATE Person.Contact
11 SET LastName = @LastName
12
13 WHERE ContactID = @ContactID;
```

You discover that the stored procedure periodically fails to update Person.Contact.
You need to ensure that Person.Contact is always updated when UpdateContact executes. The solution must minimize the amount of time required for the stored procedure to execute and the number of locks held. What should you do?

A. Add the following line of code to line 12: WITH (UPDLOCK)
B. Add the following line of code to line 05:
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
C. Add the following line of code to line 08: WITH (UPDLOCK)
D. Add the following line of code to line 05:
SET TRANSACTION ISOLATION LEVEL SNAPSHOT

Answer: C

Explanation:
* Overall, you should use UPDLOCK when you read a value that you plan to update later in the same transaction to prevent the value from changing.
* UPDLOCK
Specifies that update locks are to be taken and held until the transaction completes. UPDLOCK takes update locks for read operations only at the row-level or page-level. If UPDLOCK is combined with TABLOCK, or a table-level lock is taken for some other reason, an exclusive (X) lock will be taken instead.
When UPDLOCK is specified, the READCOMMITTED and READCOMMITTEDLOCK isolation level hints are ignored. For example, if the isolation level of the session is set to SERIALIZABLE and a query specifies (UPDLOCK, READCOMMITTED), the

READCOMMITTED hint is ignored and the transaction is run using the SERIALIZABLE isolation level.

QUESTION: 107
You use SQL Server 2014. The physical server is a dedicated database server that has 120GB of RAM available. There is approximately 50GB of storage space available on a slow local disk. You create a new stored procedure. You decide you need to temporarily hold approximately 300,000 rows from two tables, from which you will compute two complex business scores. The stored procedure will use temporary storage defined as follows:

```
AccountNumber char(10) NOT NULL
YearToDateSalesTotal decimal(15,2) NULL
SalesScore int NULL
FutureSalesExpectationScore int NULL
```

The code will make several passes through the data, applying complex calculations before writing the data to a permanent disk-based table in the same database from which it reads the data. For this stored procedure, you need to deal with temporary data in the most efficient way to minimize physical disk pressure. What should you do? More than one answer choice may achieve the goal. Select the BEST answer.

A.
```
CREATE TYPE dbo.AccountScoringModel as TABLE
(
     AccountNumber char(10) COLLATE Latin1_General_100_BIN2 NOT NULL ,
     YearToDateSalesTotal decimal(15,2) NULL,
     SalesScore int NULL,
     FutureSalesExpectationScore int NULL,
     INDEX AccountNumber HASH (AccountNumber) WITH (BUCKET_COUNT = 25000)
) WITH ( MEMORY_OPTIMIZED = ON )
GO
DECLARE @AccountScoring as dbo.AccountScoringModel
```

B.
```
DECLARE @AccountScoring as TABLE
(
     AccountNumber char(10) NOT NULL,
     YearToDateSalesTotal decimal(15,2) NULL,
     SalesScore int NULL,
     FutureSalesExpectationScore int NULL
)
```

C.
```
CREATE TABLE #AccountScoring
(
     AccountNumber char(10) NOT NULL,
     YearToDateSalesTotal decimal(15,2) NULL,
     SalesScore int NULL,
     FutureSalesExpectationScore int NULL
)
```

D.
```
CREATE TYPE dbo.AccountScoringModel as TABLE
(
     AccountNumber char(10) COLLATE Latin1_General_100_BIN2 NOT NULL ,
     YearToDateSalesTotal decimal(15,2) NULL,
     SalesScore int NULL,
     FutureSalesExpectationScore int NULL,
     INDEX AccountNumber HASH (AccountNumber) WITH (BUCKET_COUNT = 120)
) WITH ( MEMORY_OPTIMIZED = ON )
GO
DECLARE @AccountScoring as dbo.AccountScoringModel
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: A

Explanation:
* You must specify a value for the BUCKET_COUNT parameter when you create the memory-optimized table. In most cases the bucket count should be between 1 and 2 times the number of distinct values in the index key. If the index key contains a lot of duplicate values, on average there are more than 10 rows for each index key value, use a nonclustered index instead
You may not always be able to predict how many values a particular index key may have or will have. Performance should be acceptable if the BUCKET_COUNT value is within 5

times of the actual number of key values.