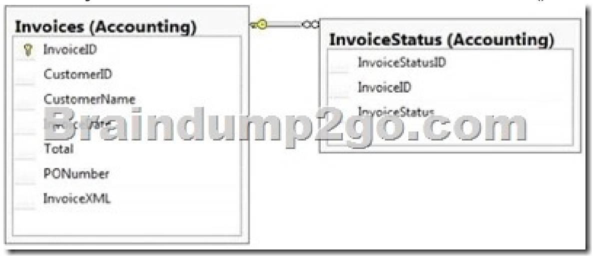
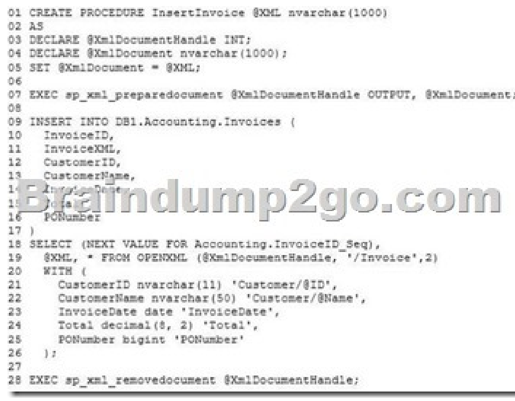
2015 Braindump2go 70-469 Dumps PDF Free Download (1-10)

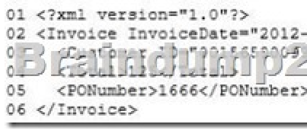
New Braindump2go 70-469 Exam Questions Updated Today! Want to know New Questions in 2015 70-469 Exam? Download Free Braindump2go 70-469 Exam Preparation Materials Now! Vendor: MicrosoftExam Code: 70-469Exam Name: Recertification for MCSE: Data PlatformKeywords: 70-469 Exam Dumps,70-469 Practice Tests,70-469 Practice Exams,70-469 Exam Questions,70-469 PDF,70-469 VCE Free,70-469 Book,70-469 E-Book,70-469 Study Guide,70-469 Braindump,70-469 Prep Guide

Compare
Pass4
281 Q&As –
$124.99
No Discount

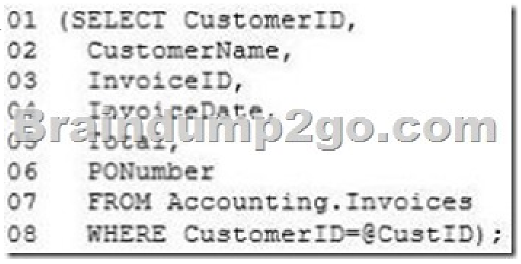
Case Study 1 - Scenario 1 (Question 1 - Question 10)Application InformationYour company receives invoices in XML format from customers. Currently, the invoices are stored as files and processed by a desktop application. The application has several performance and security issues. The application is being migrated to a SQL Server-based solution. A schema named InvoiceSchema has been created for the invoices xml.The data in the invoices is sometimes incomplete. The incomplete data must be stored and processed as-is. Users cannot filter the data provided through views.You are designing a SQL Server database named DB1 that will be used to receive, process, and securely store the invoice data. A third-party Microsoft .NET Framework component will be purchased to perform tax calculations. The third-party tax component will be provided as a DLL file named Treytax.dll and a source code file named Amortize.cs. The component will expose a class named TreyResearch and a method named Amortize().The files are located in c:temp.The following graphic shows the planned tables: 

You have a sequence named Accounting.InvoiceID_Seq.You plan to create two certificates named CERT1 and CERT2. You will create CERT1 in master. You will create CERT2 in DB1.You have a legacy application that requires the ability to generate dynamic T-SQL statements against DB1. A sample of the queries generated by the legacy application appears in Legacy.sql.Application RequirementsThe planned database has the following requirements:- All stored procedures must be signed.- The original XML invoices must be stored in the database. - An XML schema must be used to validate the invoice data.- Dynamic T-SQL statements must be converted to stored procedures. - Access to the .NET Framework tax components must be available to T-SQL objects.- Columns must be defined by using data types that minimize the amount of space used by each table.- Invoices stored in the InvoiceStatus table must refer to an invoice by the same identifier used by the Invoice table.- To protect against the theft of backup disks, invoice data must be protected by using the highest level of encryption.- The solution must provide a table-valued function that provides users with the ability to filter invoices by customer.- Indexes must be optimized periodically based on their fragmentation by using the minimum amount of administrative effort.Usp_InsertInvoices.sql 

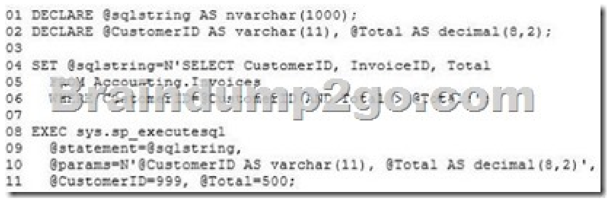
```
01 CREATE PROCEDURE InsertInvoice @XML nvarchar(1000)
02 AS
03 DECLARE @XmlDocumentHandle INT;
04 DECLARE @XmlDocument nvarchar(1000);
05 SET @XmlDocument = @XML;
06
07 EXEC sp_xml_preparedocument @XmlDocumentHandle OUTPUT, @XmlDocument;
08
09 INSERT INTO DB1.Accounting.Invoices (
10   InvoiceID,
11   InvoiceXML,
12   CustomerID,
13   CustomerName,
14   InvoiceDate,
15   Total,
16   PONumber
17 )
18 SELECT (NEXT VALUE FOR Accounting.InvoiceID_Seq),
19   @XML, * FROM OPENXML (@XmlDocumentHandle, '/Invoice',2)
20   WITH (
21     CustomerID nvarchar(11) 'Customer/@ID',
22     CustomerName nvarchar(50) 'Customer/@Name',
23     InvoiceDate date 'InvoiceDate',
24     Total decimal(8, 2) 'Total',
25     PONumber bigint 'PONumber'
26   );
27
28 EXEC sp_xml_removedocument @XmlDocumentHandle;
```

Invoices.xmlAll customer IDs are 11 digits. The first three digits of a customer ID represent the customer's country. The remaining eight digits are the customer's account number.The following is a sample of a customer invoice in XML format: 

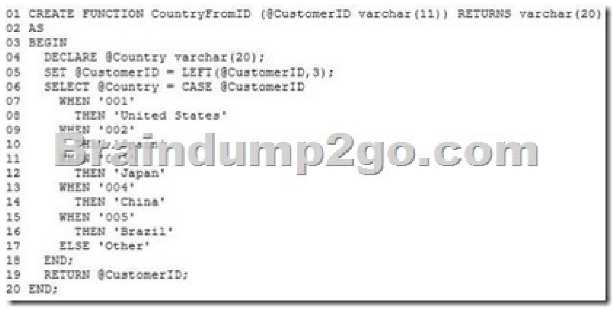
```
<?xml version="1.0"?>
<Invoice InvoiceDate="2012-
[illegible]
[illegible]
  <PONumber>1666</PONumber>
</Invoice>
```

InvoicesByCustomer.sql 

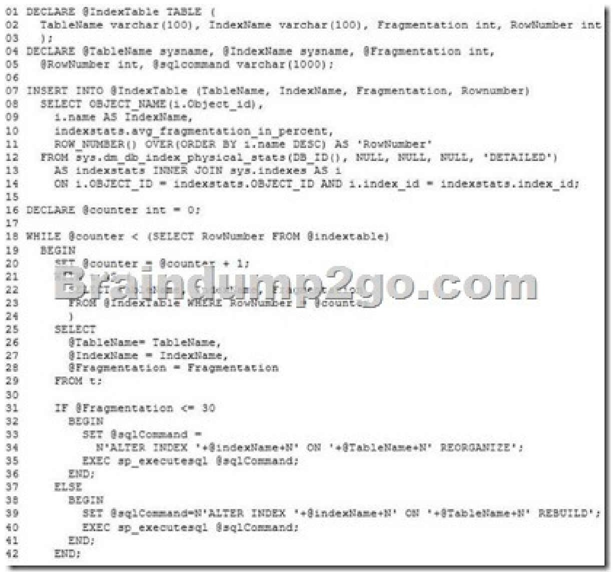
```
(SELECT CustomerID,
  CustomerName,
  InvoiceID,
  InvoiceDate,
  Total,
  PONumber
  FROM Accounting.Invoices
  WHERE CustomerID=@CustID);
```

Legacy.sql 

```
DECLARE @sqlstring AS nvarchar(1000);
DECLARE @CustomerID AS varchar(11), @Total AS decimal(8,2);

SET @sqlstring=N'SELECT CustomerID, InvoiceID, Total
  FROM Accounting.Invoices
  [illegible]

EXEC sys.sp_executesql
  @statement=@sqlstring,
  @params=N'@CustomerID AS varchar(11), @Total AS decimal(8,2)',
  @CustomerID=999, @Total=500;
```

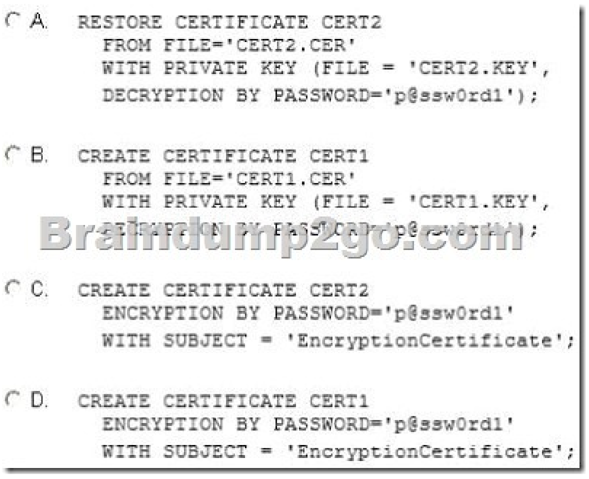
CountryFromID.sql 

```
CREATE FUNCTION CountryFromID (@CustomerID varchar(11)) RETURNS varchar(20)
AS
BEGIN
  DECLARE @Country varchar(20);
  SET @CustomerID = LEFT(@CustomerID,3);
  SELECT @Country = CASE @CustomerID
    WHEN '001'
      THEN 'United States'
    WHEN '002'
      [illegible]
    [illegible]
      THEN 'Japan'
    WHEN '004'
      THEN 'China'
    WHEN '005'
      THEN 'Brazil'
    ELSE 'Other'
  END;
  RETURN @CustomerID;
END;
```

IndexManagement.sql 

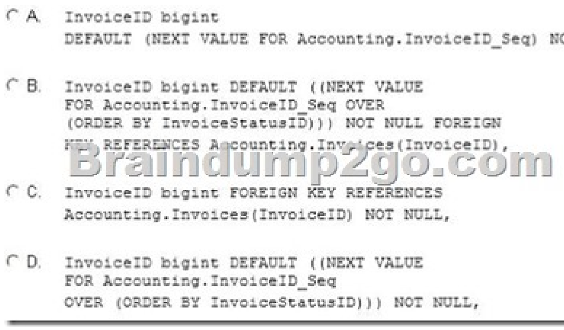
```
DECLARE @IndexTable TABLE (
  TableName varchar(100), IndexName varchar(100), Fragmentation int, RowNumber int
  );
DECLARE @TableName sysname, @IndexName sysname, @Fragmentation int,
  @RowNumber int, @sqlcommand varchar(1000);

INSERT INTO @IndexTable (TableName, IndexName, Fragmentation, Rownumber)
  SELECT OBJECT_NAME(i.Object_id),
    i.name AS IndexName,
    indexstats.avg_fragmentation_in_percent,
    ROW_NUMBER() OVER(ORDER BY i.name DESC) AS 'RowNumber'
  FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, 'DETAILED')
    AS indexstats INNER JOIN sys.indexes AS i
    ON i.OBJECT_ID = indexstats.OBJECT_ID AND i.index_id = indexstats.index_id;

DECLARE @counter int = 0;

WHILE @counter < (SELECT RowNumber FROM @indextable)
  BEGIN
    SET @counter = @counter + 1;
    [illegible]
      [illegible]
      FROM @IndexTable WHERE RowNumber = @counter
      )
    SELECT
      @TableName= TableName,
      @IndexName = IndexName,
      @Fragmentation = Fragmentation
    FROM t;

    IF @Fragmentation <= 30
      BEGIN
        SET @sqlCommand =
          N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REORGANIZE';
        EXEC sp_executesql @sqlCommand;
      END;
    ELSE
      BEGIN
        SET @sqlCommand=N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REBUILD';
        EXEC sp_executesql @sqlCommand;
      END;
    END;
```

QUESTION 1You are testing disaster recovery procedures.You attempt to restore DB1 to a different server and you receive the following error message: "Msg 33111.Level 16, State 3, Line 1Cannot find server certificate with thumbprint ,0xA694FBEA88C9354E5E2567C30A2A69E8FB4C44A9Msg 3013, Level 16, State 1, Line 1RESTORE DATABASE is terminating abnormally."You need to ensure that you can restore DB1 to a different server.Which code segment should you execute?



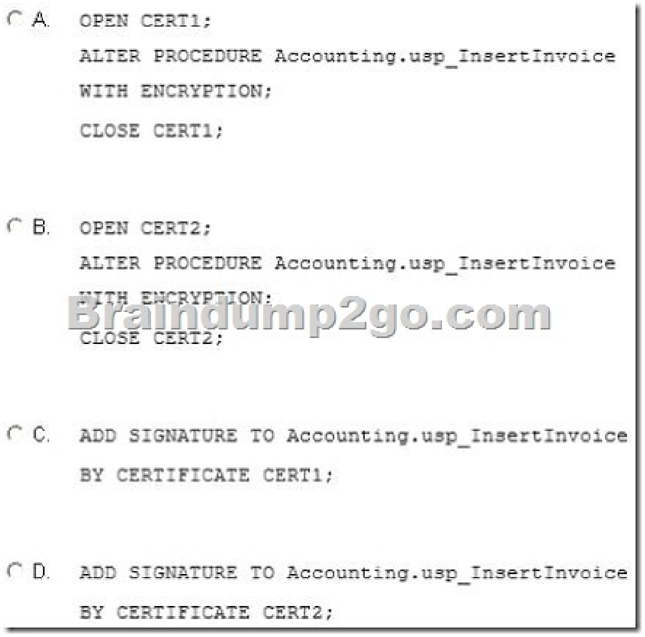
```
A. RESTORE CERTIFICATE CERT2
     FROM FILE='CERT2.CER'
     WITH PRIVATE KEY (FILE = 'CERT2.KEY',
     DECRYPTION BY PASSWORD='p@ssw0rd1');

B. CREATE CERTIFICATE CERT1
     FROM FILE='CERT1.CER'
     WITH PRIVATE KEY (FILE = 'CERT1.KEY',
     DECRYPTION BY PASSWORD='p@ssw0rd1');

C. CREATE CERTIFICATE CERT2
     ENCRYPTION BY PASSWORD='p@ssw0rd1'
     WITH SUBJECT = 'EncryptionCertificate';

D. CREATE CERTIFICATE CERT1
     ENCRYPTION BY PASSWORD='p@ssw0rd1'
     WITH SUBJECT = 'EncryptionCertificate';
```

A. Option AB. Option BC. Option CD. Option D Answer: B QUESTION 2You need to create the InvoiceStatus table in DB1.How should you define the InvoiceID column in the CREATE TABLE statement?



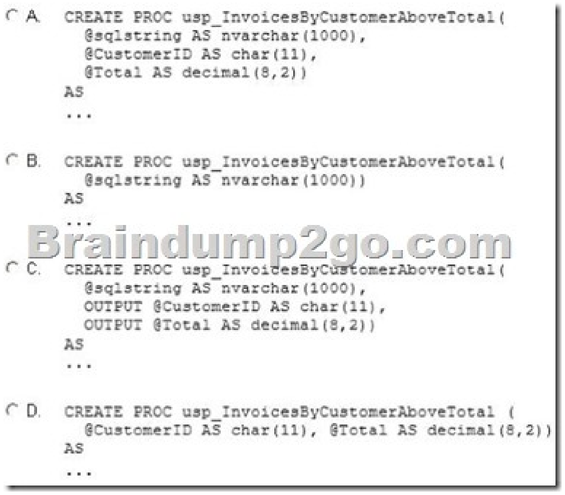
```
A. InvoiceID bigint
   DEFAULT (NEXT VALUE FOR Accounting.InvoiceID_Seq) NO

B. InvoiceID bigint DEFAULT ((NEXT VALUE
   FOR Accounting.InvoiceID_Seq OVER
   (ORDER BY InvoiceStatusID))) NOT NULL FOREIGN
   KEY REFERENCES Accounting.Invoices(InvoiceID),

C. InvoiceID bigint FOREIGN KEY REFERENCES
   Accounting.Invoices(InvoiceID) NOT NULL,

D. InvoiceID bigint DEFAULT ((NEXT VALUE
   FOR Accounting.InvoiceID_Seq
   OVER (ORDER BY InvoiceStatusID))) NOT NULL,
```

A. Option AB. Option BC. Option CD. Option D Answer: C QUESTION 3You need to modify InsertInvoice to comply with the application requirements.Which code segment should you execute?

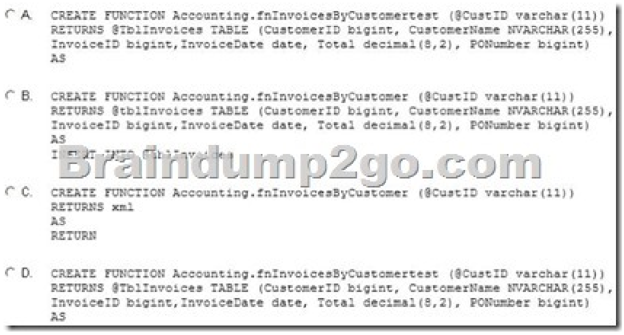


```
A. OPEN CERT1;
   ALTER PROCEDURE Accounting.usp_InsertInvoice
   WITH ENCRYPTION;
   CLOSE CERT1;

B. OPEN CERT2;
   ALTER PROCEDURE Accounting.usp_InsertInvoice
   WITH ENCRYPTION;
   CLOSE CERT2;

C. ADD SIGNATURE TO Accounting.usp_InsertInvoice
   BY CERTIFICATE CERT1;

D. ADD SIGNATURE TO Accounting.usp_InsertInvoice
   BY CERTIFICATE CERT2;
```

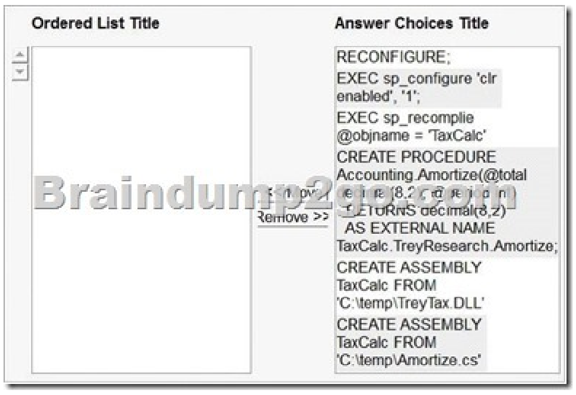
A. Option AB. Option BC. Option CD. Option D Answer: D QUESTION 4Which data type should you use for CustomerID? A. varchar(11)B. bigintC. nvarchar(11)D. char(11) Answer: DExplanation:Invoices.xmlAll customer IDs are 11 digits. The first three digits of a customer ID represent the customer's country. The remaining eight digits are the customer's account number. int: -231 (-2,147,483,648) to 231-1 (2,147,483,647) (just 10 digits max) bigint: -263 (-9,223,372,036,854,775,808) to 263-1 (9,223,372,036,854,775,807) http://msdn.microsoft.com/en-us/library/ms176089.aspx http://msdn.microsoft.com/en-us/library/ms187745.aspx QUESTION 5You attempt to process an invoice by using usp_InsertInvoice.sql and you receive the following error message: "Msg 515, Level 16, State 2, Procedure usp_InsertInvoice, Line Cannot insert the value NULL into column 'InvoiceDate', table 'DB1.Accounting.Invoices'; column does not allow nulls. INSERT fails."You need to modify usp_InsertInvoice.sql to resolve the error.How should you modify the INSERT statement? A. InvoiceDate varchar(l00) 'InvoiceDate',B. InvoiceDate varchar(100) 'Customer/InvoiceDate', 'C. InvoiceDate date '@InvoiceDate',D. InvoiceDate date 'Customer/@InvoiceDate', Answer: C QUESTION 6You need to modify the function in CountryFromID.sql to ensure that the country name is returned instead of the country ID.Which line of code should you modify in CountryFromID.sql? A. 04B. 05C. 06D. 19 Answer: DExplanation:http://msdn.microsoft.com/en-us/library/ms186755.aspx http://msdn.microsoft.com/en-us/library/ms191320.aspx QUESTION 7You execute IndexManagement.sql and you receive the following error message:"Msg 512, Level 16, State 1, Line 12Subquery returned more than 1 value. This is not permitted when the subquery follows =,! =, <, <= ,>, > = or when the subquery is used as an expression."You need to ensure that IndexManagement.sql executes properly.Which WHILE statement should you use at line 18? A. WHILE SUM(@RowNumber) < (SELECT @counter FROM @indextable)B. WHILE @counter < (SELECT COUNT(RowNumber) FROM @indextable)C. WHILE COUNT(@RowNumber) < (SELECT @counter FROM @indextable)D. WHILE @counter < (SELECT SUM(RowNumber) FROM @indextabie) Answer: B QUESTION 8You need to convert the functionality of Legacy.sql to use a stored procedure.Which code segment should the stored procedure contain?


```
A.  CREATE PROC usp_InvoicesByCustomerAboveTotal(
      @sqlstring AS nvarchar(1000),
      @CustomerID AS char(11),
      @Total AS decimal(8,2))
    AS
    ...

B.  CREATE PROC usp_InvoicesByCustomerAboveTotal(
      @sqlstring AS nvarchar(1000))
    AS
    ...

C.  CREATE PROC usp_InvoicesByCustomerAboveTotal(
      @sqlstring AS nvarchar(1000),
      OUTPUT @CustomerID AS char(11),
      OUTPUT @Total AS decimal(8,2))
    AS
    ...

D.  CREATE PROC usp_InvoicesByCustomerAboveTotal (
      @CustomerID AS char(11), @Total AS decimal(8,2))
    AS
    ...
```

A. Option AB. Option BC. Option CD. Option D Answer: DExplanation: http://msdn.microsoft.com/en-us/library/ms187926.aspxhttp://msdn.microsoft.com/en-us/library/ms190782.aspx http://msdn.microsoft.com/en-us/library/bb669091.aspxhttp://msdn.microsoft.com/en-us/library/windows/desktop/ms709342.aspx http://msdn.microsoft.com/en-us/library/ms188001.aspx QUESTION 9You need to create a function that filters invoices by CustomerID. The SELECT statement for the function is contained in InvoicesByCustomer.sql.Which code segment should you use to complete the function?

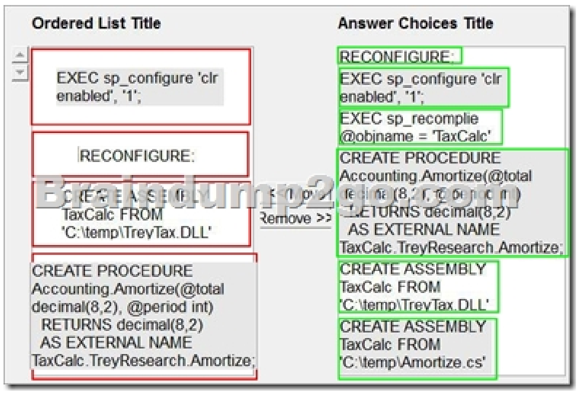

```
A.  CREATE FUNCTION Accounting.fnInvoicesByCustomertest (@CustID varchar(11))
    RETURNS @TblInvoices TABLE (CustomerID bigint, CustomerName NVARCHAR(255),
    InvoiceID bigint,InvoiceDate date, Total decimal(8,2), PONumber bigint)
    AS

B.  CREATE FUNCTION Accounting.fnInvoicesByCustomer (@CustID varchar(11))
    RETURNS @tblInvoices TABLE (CustomerID bigint, CustomerName NVARCHAR(255),
    InvoiceID bigint,InvoiceDate date, Total decimal(8,2), PONumber bigint)
    AS
    [illegible]

C.  CREATE FUNCTION Accounting.fnInvoicesByCustomer (@CustID varchar(11))
    RETURNS xml
    AS
    RETURN

D.  CREATE FUNCTION Accounting.fnInvoicesByCustomertest (@CustID varchar(11))
    RETURNS @TblInvoices TABLE (CustomerID bigint, CustomerName NVARCHAR(255),
    InvoiceID bigint,InvoiceDate date, Total decimal(8,2), PONumber bigint)
    AS
```

A. Option AB. Option BC. Option CD. Option D Answer: A QUESTION 10Drag and Drop QuestionsYou need to build a stored procedure that amortizes the invoice amount. Which code segment should you use to create the stored procedure? To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.



```
RECONFIGURE;
EXEC sp_configure 'clr enabled', '1';
EXEC sp_recomplie @objname = 'TaxCalc'
CREATE PROCEDURE Accounting.Amortize(@total decimal(8,2), @period int)
  RETURNS decimal(8,2)
  AS EXTERNAL NAME TaxCalc.TreyResearch.Amortize;
CREATE ASSEMBLY TaxCalc FROM 'C:\temp\TreyTax.DLL'
CREATE ASSEMBLY TaxCalc FROM 'C:\temp\Amortize.cs'
```

Answer:



```
EXEC sp_configure 'clr enabled', '1';
RECONFIGURE;
CREATE ASSEMBLY TaxCalc FROM 'C:\temp\TreyTax.DLL'
CREATE PROCEDURE Accounting.Amortize(@total decimal(8,2), @period int)
  RETURNS decimal(8,2)
  AS EXTERNAL NAME TaxCalc.TreyResearch.Amortize;
```