



# Microsoft

## Exam 70-469

### Recertification for MCSE: Data Platform

Version: 5.0

[ Total Questions: 281 ]

### Topic break down

Topic	No. of Questions
Topic 1: Scenario 1	10
Topic 2: Scenario 2	13
Topic 3: Scenario 3	6
Topic 4: Scenario 4	10
Topic 5: Litware, Inc	12
Topic 6: Coho Winery	14
Topic 7: Fourth Coffee	11
Topic 8: Litware, Inc	10
Topic 9: Contoso Ltd	10
Topic 10: Fabrikam, Inc	10
Topic 11: A.Datum	13
Topic 12: Mix Questions Set 1	55
Topic 13: Mix Questions Set 2	107

## Topic 1, Scenario 1

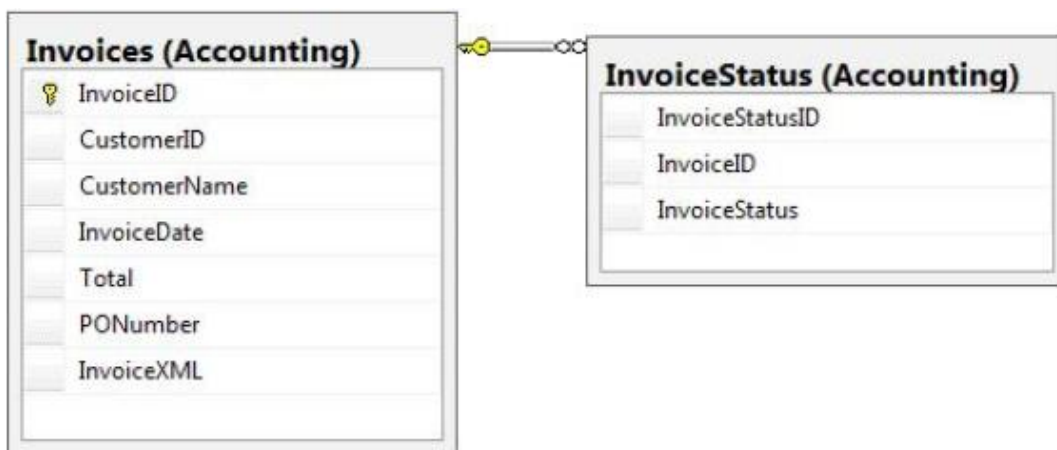
### Application Information

Your 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 Requirements

The 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.xml

All 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:

```

01 <?xml version="1.0"?>
02 <Invoice InvoiceDate="2012-02-20">
03     <Customer ID="00156590099" Name="Litware" />
04     <Total>125</Total>
05     <PONumber>1666</PONumber>
06 </Invoice>

```

### InvoicesByCustomer.sql

```
01 (SELECT CustomerID,  
02     CustomerName,  
03     InvoiceID,  
04     InvoiceDate,  
05     Total,  
06     PONumber  
07     FROM Accounting.Invoices  
08     WHERE CustomerID=@CustID);
```

### Legacy.sql

```
01 DECLARE @sqlstring AS nvarchar(1000);  
02 DECLARE @CustomerID AS varchar(11), @Total AS decimal(8,2);  
03  
04 SET @sqlstring=N'SELECT CustomerID, InvoiceID, Total  
05     FROM Accounting.Invoices  
06     WHERE CustomerID=@CustomerID AND Total > @Total;';  
07  
08 EXEC sys.sp_executesql  
09     @statement=@sqlstring,  
10     @params=N'@CustomerID AS varchar(11), @Total AS decimal(8,2)',  
11     @CustomerID=999, @Total=500;
```

### CountryFromID.sql

```
01 CREATE FUNCTION CountryFromID (@CustomerID varchar(11)) RETURNS varchar(20)  
02 AS  
03 BEGIN  
04     DECLARE @Country varchar(20);  
05     SET @CustomerID = LEFT(@CustomerID,3);  
06     SELECT @Country = CASE @CustomerID  
07         WHEN '001'  
08             THEN 'United States'  
09         WHEN '002'  
10             THEN 'Spain'  
11         WHEN '003'  
12             THEN 'Japan'  
13         WHEN '004'  
14             THEN 'China'  
15         WHEN '005'  
16             THEN 'Brazil'  
17         ELSE 'Other'  
18     END;  
19     RETURN @CustomerID;  
20 END;
```

### IndexManagement.sql

```
01 DECLARE @IndexTable TABLE (  
02   TableName varchar(100), IndexName varchar(100), Fragmentation int, RowNumber int  
03 );  
04 DECLARE @TableName sysname, @IndexName sysname, @Fragmentation int,  
05   @RowNumber int, @sqlcommand varchar(1000);  
06  
07 INSERT INTO @IndexTable (TableName, IndexName, Fragmentation, Rownumber)  
08   SELECT OBJECT_NAME(i.Object_id),  
09     i.name AS IndexName,  
10     indexstats.avg_fragmentation_in_percent,  
11     ROW_NUMBER() OVER(ORDER BY i.name DESC) AS 'RowNumber'  
12 FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, 'DETAILED')  
13 AS indexstats INNER JOIN sys.indexes AS i  
14   ON i.OBJECT_ID = indexstats.OBJECT_ID AND i.index_id = indexstats.index_id;  
15  
16 DECLARE @counter int = 0;  
17  
18 WHILE @counter < (SELECT RowNumber FROM @indextable)  
19 BEGIN  
20   SET @counter = @counter + 1;  
21   WITH t AS (  
22     SELECT TableName, IndexName, Fragmentation  
23     FROM @IndexTable WHERE RowNumber = @counter  
24   )  
25   SELECT  
26     @TableName= TableName,  
27     @IndexName = IndexName,  
28     @Fragmentation = Fragmentation  
29   FROM t;  
30  
31   IF @Fragmentation <= 30  
32     BEGIN  
33       SET @sqlCommand =  
34       N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REORGANIZE';  
35       EXEC sp_executesql @sqlCommand;  
36     END;  
37   ELSE  
38     BEGIN  
39       SET @sqlCommand=N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REBUILD';  
40       EXEC sp_executesql @sqlCommand;  
41     END;  
42   END;
```

### Question No : 1 - (Topic 1)

You 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) NOT NULL,
- 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 A
- B. Option B
- C. Option C
- D. Option D

**Answer: C**

### Question No : 2 - (Topic 1)

You 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 10

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(100) 'InvoiceDate',
- B. InvoiceDate varchar(100) 'Customer/InvoiceDate', '
- C. InvoiceDate date '@InvoiceDate',
- D. InvoiceDate date 'Customer/@InvoiceDate',

**Answer: C**

**Question No : 3 - (Topic 1)**

You 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 A
- B. Option B
- C. Option C
- D. Option D

**Answer: D**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms187926.aspx>

<http://msdn.microsoft.com/en-us/library/ms190782.aspx>

<http://msdn.microsoft.com/en-us/library/bb669091.aspx>

<http://msdn.microsoft.com/en-us/library/windows/desktop/ms709342.aspx>

<http://msdn.microsoft.com/en-us/library/ms188001.aspx>



**Question No : 4 - (Topic 1)**

You 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), PNumber 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), PNumber bigint)
AS
INSERT INTO @tblInvoices
```
- 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), PNumber bigint)
AS
```

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

**Answer: A**

**Question No : 5 DRAG DROP - (Topic 1)**

You 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.

Ordered List Title	Answer Choices Title
<div style="border: 1px solid gray; height: 276px; width: 263px;"></div>	<pre>RECONFIGURE; EXEC sp_configure 'clr enabled', '1'; EXEC sp_recomple @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'</pre>
<p>&lt;&lt; Move Remove &gt;&gt;</p>	

**Answer:**

Ordered List Title	Answer Choices Title
<div style="border: 1px dashed red; padding: 5px;"><pre>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;</pre></div>	<div style="border: 1px dashed green; padding: 5px;"><pre>RECONFIGURE; EXEC sp_configure 'clr enabled', '1'; EXEC sp_recomple @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'</pre></div>
<p>&lt;&lt; Move Remove &gt;&gt;</p>	

**Explanation:**

```
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;
```

D:\Documents and Settings\useralbo\Desktop\1.jpg

<http://msdn.microsoft.com/en-us/library/ms131089.aspx>

<http://msdn.microsoft.com/en-us/library/ms131048.aspx>

<http://msdn.microsoft.com/en-us/library/ms187926.aspx>

### **Question No : 6 - (Topic 1)**

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

**Answer: D**

**Question No : 7 - (Topic 1)**

You execute IndexManagement.sql and you receive the following error message:

"Msg 512, Level 16, State 1, Line 12

Subquery 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 @indextable)

**Answer: B**

### Question No : 8 - (Topic 1)

You 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. 04
- B. 05
- C. 06
- D. 19

**Answer: D**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms186755.aspx>

<http://msdn.microsoft.com/en-us/library/ms191320.aspx>

### Question No : 9 - (Topic 1)

You 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 1

Cannot find server certificate with thumbprint

,0xA694FBEA88C9354E5E2567C30A2A69E8FB4C44A9\

Msg 3013, Level 16, State 1, Line 1

RESTORE 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 A
- B. Option B
- C. Option C
- D. Option D

**Answer: B**

**Question No : 10 - (Topic 1)**

Which data type should you use for CustomerID?

- A. varchar(11)
- B. bigint
- C. nvarchar(11)
- D. char(11)

**Answer: D**

**Explanation:**

Invoices.xml

All 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:  $-2^{31}$  (-2,147,483,648) to  $2^{31}-1$  (2,147,483,647) (just 10 digits max)

bigint:  $-2^{63}$  (-9,223,372,036,854,775,808) to  $2^{63}-1$  (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>

## **Topic 2, Scenario 2**

### **Application Information**

You have two servers named SQL1 and SQL2 that have SQL Server 2012 installed.

You have an application that is used to schedule and manage conferences.

Users report that the application has many errors and is very slow.

You are updating the application to resolve the issues.

You plan to create a new database on SQL1 to support the application. A junior database administrator has created all the scripts that will be used to create the database. The script that you plan to use to create the tables for the new database is shown in Tables.sql. The script that you plan to use to create the stored procedures for the new database is shown in StoredProcedures.sql. The script that you plan to use to create the indexes for the new database is shown in Indexes.sql. (Line numbers are included for reference only.)

A database named DB2 resides on SQL2. DB2 has a table named SpeakerAudit that will audit changes to a table named Speakers.

A stored procedure named usp\_UpdateSpeakersName will be executed only by other stored procedures. The stored procedures executing usp\_UpdateSpeakersName will always handle transactions.

A stored procedure named usp\_SelectSpeakersByName will be used to retrieve the names of speakers. Usp\_SelectSpeakersByName can read uncommitted data.

A stored procedure named usp\_GetFutureSessions will be used to retrieve sessions that will occur in the future.

## Procedures.sql

```
01 CREATE PROCEDURE usp_UpdateSpeakerName
02   @SpeakerID int,
03   @LastName nvarchar(100)
04 AS
05
06 BEGIN TRY
07
08 UPDATE Speakers
09 SET LastName = @LastName
10 WHERE SpeakerID = @SpeakerID;
11
12 INSERT INTO SQL2.DB2.dbo.SpeakerAudit(SpeakerID, LastName)
13 VALUES (@SpeakerID, @LastName);
14
15 END TRY
16 BEGIN CATCH
17
18 END CATCH;
19
20 GO
21
22 CREATE PROCEDURE usp_SelectSpeakersByName
23   @LastName nvarchar(100)
24 AS
25 SELECT SpeakerID,
26        FirstName,
27        LastName
28 FROM Speakers
29 WHERE LastName LIKE @LastName + '%'
30
31 GO
32
33 CREATE PROCEDURE usp_InsertSessions
34   @SessionData SessionDataTable READONLY
35 AS
36 INSERT INTO Sessions
37   (SpeakerID, Title, Absract, DeliveryTime, TitleAndSpeaker)
38 SELECT SpeakerID, Title, Absract, DeliveryTime, TitleAndSpeaker
39 FROM @SessionData;
40 GO
41
42 CREATE PROCEDURE usp_UpdateSessionRoom
43   @RoomID int,
44   @SpeakerID int
45 AS
```



```
46 SET TRANSACTION ISOLATION LEVEL SNAPSHOT
47 BEGIN TRANSACTION;
48
49 SELECT SessionID,
50    Title
51 FROM Sessions
52 WHERE SpeakerID = @SpeakerID;
53
54 UPDATE Sessions
55 SET RoomID = @RoomID
56 WHERE SpeakerID = @SpeakerID;
57
58 COMMIT TRANSACTION;
59
60 CREATE PROCEDURE usp_AttendeesReport
61    @LastName varchar(100)
62 AS
63 SELECT FirstName + ' ' + LastName AS FullName
64 FROM Attendees
65 WHERE LastName = @LastName;
66 GO
67
68 CREATE PROCEDURE usp_GetFutureSessions
69 AS
70 SELECT SpeakerID,
71    RoomID,
72    DeliveryTime
73 FROM Sessions
74
75 GO
76
77 CREATE PROCEDURE usp_TestSpeakers
78 AS
79 EXECUTE usp_SelectSpeakersByName 'a';
80 EXECUTE usp_SelectSpeakersByName 'an';
81 EXECUTE usp_SelectSpeakersByName 'and';
82 EXECUTE usp_SelectSpeakersByName 'ander';
83 EXECUTE usp_SelectSpeakersByName 'anderson';
84 EXECUTE usp_SelectSpeakersByName 'b';
85 EXECUTE usp_SelectSpeakersByName 'bi';
86 ...
87 EXECUTE usp_SelectSpeakersByName 'zzz';
88 GO
```

Indexes.sql

```
01 CREATE INDEX IX_Sessions ON Sessions
02 (SessionID, DeliveryTime)
03 INCLUDE (RoomID)
04
05 GO
06
07 CREATE INDEX IX_Speakers ON Speakers
08 (LastName);
09 GO
10
11 CREATE INDEX IX_Attendees_Name ON Attendees
12 (FirstName, LastName);
13
14 GO
15
16 CREATE INDEX IX_Attendees_Confirmed ON Attendees
17 (Confirmed);
18 GO
```

## Tables.sql

```
01 CREATE DATABASE Conference;
02 GO
03
04 ALTER DATABASE Conference
05 SET READ_COMMITTED_SNAPSHOT ON;
06 GO
07
08 CREATE TABLE Attendees
09 (
10     AttendeeID int IDENTITY (1,1) NOT NULL,
11     FirstName nvarchar(100) NOT NULL,
12     LastName nvarchar(100) NOT NULL,
13     EmailAddress nvarchar(100) NOT NULL,
14
15     CONSTRAINT PK_Attendees_AttendeeID PRIMARY KEY (AttendeeID)
16 );
17 GO
18
19 CREATE TABLE Speakers
20 (
21     SpeakerID int IDENTITY(1,1) NOT NULL,
22     FirstName nvarchar(100) NOT NULL,
23     LastName nvarchar(100) NOT NULL,
24     Photo varbinary(max),
25     CONSTRAINT PK_Speakers_SpeakerID PRIMARY KEY (SpeakerID)
26 );
27 GO
28
29 CREATE TABLE Sessions
30 (
31     SessionID uniqueidentifier NOT NULL
32     CONSTRAINT DF_SessionID DEFAULT (NEWID()),
33     SpeakerID int NOT NULL,
34     Title nvarchar(100) NOT NULL,
35     Abstract nvarchar(max) NOT NULL,
36     DeliveryTime datetime NOT NULL,
37     TitleAndSpeaker nvarchar(200)
38
39 );
40 GO
41
42 CREATE TABLE Rooms
43 (
44     RoomID uniqueidentifier NOT NULL CONSTRAINT DF_RoomID DEFAULT (NEWID()),
45     Location varchar(100) NOT NULL
46 );
```

### Question No : 11 - (Topic 2)

While testing usp.GetFutureSessions, you discover that IX\_Sessions is accessed by a scan rather than a seek.

You need to minimize the amount of time it takes to execute usp\_GetFutureSessions.

What should you do? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Change line 02 of Indexes.sql to:  
`(DeliveryTime, SessionID)`
- B. At line 04 of Indexes.sql, add:  
`WHERE GETDATE() < DeliveryTime;`
- C. Change line 02 of Indexes.sql to:  
`(SpeakerID, RoomID, DeliveryTime)`
- D. Change line 74 of Procedures.sql to:  
`WHERE GETDATE() > DeliveryTime;`
- E. Change line 74 of Procedures.sql to:  
`WHERE GETDATE() < DeliveryTime;`
- F. At line 04 of Indexes.sql, add:  
`WHERE GETDATE() > DeliveryTime;`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer: B,E**

**Explanation:** Future delivery dates.

### Question No : 12 - (Topic 2)

You need to provide referential integrity between the Sessions table and Speakers table.

Which code segment should you add at line 47 of Tables.sql?

- A. 

```
ALTER TABLE dbo.Sessions ADD CONSTRAINT
FK_Sessions_Speakers FOREIGN KEY (SessionID)
REFERENCES dbo.Speakers (SpeakerID);
```
- B. 

```
ALTER TABLE dbo.Sessions ADD CONSTRAINT
FK_Sessions_Speakers FOREIGN KEY (SpeakerID)
REFERENCES dbo.Speakers (SpeakerID);
```
- C. 

```
ALTER TABLE dbo.Speakers ADD CONSTRAINT
FK_Speakers_Sessions FOREIGN KEY (SpeakerID)
REFERENCES dbo.Sessions (SessionID);
```
- D. 

```
ALTER TABLE dbo.Speakers ADD CONSTRAINT
FK_Speakers_Sessions FOREIGN KEY (SessionID)
REFERENCES dbo.Sessions (SessionID);
```

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

**Answer: B**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms189049.aspx>

<http://msdn.microsoft.com/en-us/library/ms179610.aspx>

<http://msdn.microsoft.com/en-us/library/ff878370.aspx>

### Question No : 13 - (Topic 2)

You need to create the object used by the parameter of usp\_InsertSessions.

Which statement should you use?

- A. CREATE XML SCHEMA COLLECTION SessionDataTable



- B. CREATE TYPE SessionDataTable AS Table
- C. CREATE SCHEMA SessionDataTable
- D. CREATE TABLE SessionDataTable

**Answer: B**

### Question No : 14 - (Topic 2)

You need to add a new column named Confirmed to the Attendees table.

The solution must meet the following requirements:

-  Have a default value of false.
-  Minimize the amount of disk space used.

Which code block should you use?

- A. ALTER TABLE Attendees  
ADD Confirmed bit DEFAULT 0;
- B. ALTER TABLE Attendees  
ADD Confirmed char(1) DEFAULT '1';
- C. ALTER TABLE Attendees  
ADD Confirmed bit DEFAULT 1;
- D. ALTER TABLE Attendees  
ADD Confirmed char(1) DEFAULT '1';

**Answer: A**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms177603.aspx>

### Question No : 15 - (Topic 2)

You discover that usp.SelectSpeakersByName executes slowly if usp\_UpdateSpeakerName executes simultaneously.

You need to minimize the execution time of usp.SelectSpeakersByName. The solution must not affect the performance of the other stored procedures.

What should you update?

- A. Usp\_UpdateSpeakerName to use the NOLOCK query hint
- B. Usp\_UpdateSpeakerName to use snapshot isolation
- C. Usp\_SelectSpeakersByName to use the NOLOCK query hint
- D. Usp\_SelectSpeakersByName to use snapshot isolation

**Answer: C**

**Explanation:** NOLOCK

Is equivalent to READUNCOMMITTED.

READUNCOMMITTED

Specifies that dirty reads are allowed.

### Question No : 16 - (Topic 2)

You are evaluating the table design.

You need to recommend a change to Tables.sql that reduces the amount of time it takes for usp\_AttendeesReport to execute.

What should you add at line 14 of Tables.sql?

- A. FullName nvarchar(100) NOT NULL CONSTRAINT DF\_FullName DEFAULT (dbo.CreateFullName (FirstName, LastName)),
- B. FullName AS (FirstName + ' ' + LastName),
- C. FullName nvarchar(100) NOT NULL DEFAULT (dbo.CreateFullName (FirstName, LastName)).
- D. FullName AS (FirstName + ' ' + LastName) PERSISTED,

**Answer: D**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms188300.aspx>

<http://msdn.microsoft.com/en-us/library/ms191250.aspx>

**Question No : 17 - (Topic 2)**

You are evaluating the index design.

You need to recommend a change to Indexes.sql that will minimize the amount of time it takes for usp\_AttendeesReport to execute. The solution must minimize the amount of database fragmentation.

Which line of code should you use to replace line 12 of Indexes.sql?

- A. (LastName);
- B. (FirstName) INCLUDE (LastName);
- C. (LastName, FirstName);
- D. (LastName) INCLUDE (FirstName);

**Answer: C**

**Question No : 18 - (Topic 2)**

You need to create the object used by the parameter of usp\_InsertSessions.

Which statement should you use?

- A. CREATE SCHEMA SessionDataTable
- B. CREATE TYPE SessionDataTable AS Table
- C. CREATE TABLE SessionDataTable
- D. CREATE XML SCHEMA COLLECTION SessionDataTable

**Answer: A**

**Question No : 19 - (Topic 2)**

You execute usp\_TestSpeakers.

You discover that usp\_SelectSpeakersByName uses inefficient execution plans.

You need to update usp\_SelectSpeakersByName to ensure that the most efficient



execution plan is used.

What should you add at line 30 of Procedures.sql?

- A. OPTION (FORCESCAN)
- B. OPTION (FORCESEEK)
- C. OPTION (OPTIMIZE FOR UNKNOWN)
- D. OPTION (OPTIMIZE FOR (@LastName= 'Anderson'))

**Answer: C**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms181714.aspx>

### **Question No : 20 - (Topic 2)**

You need to recommend a solution to ensure that SQL1 supports the auditing requirements of usp\_UpdateSpeakerName.

What should you include in the recommendation?

- A. The Distributed Transaction Coordinator (DTC)
- B. Transactional replication
- C. Change data capture
- D. Change tracking

**Answer: A**

### **Question No : 21 - (Topic 2)**

You need to modify usp\_SelectSpeakersByName to support server-side paging. The solution must minimize the amount of development effort required.

What should you add to usp\_SelectSpeakersByName?

- A. A table variable

- B. An OFFSET-FETCH clause
- C. The ROWNUMBER keyword
- D. A recursive common table expression

**Answer: B**

**Explanation:**

<http://www.mssqltips.com/sqlservertip/2696/comparing-performance-for-different-sql-serverpaging-methods/>

<http://msdn.microsoft.com/en-us/library/ms188385.aspx>

<http://msdn.microsoft.com/en-us/library/ms180152.aspx>

<http://msdn.microsoft.com/en-us/library/ms186243.aspx>

<http://msdn.microsoft.com/en-us/library/ms186734.aspx>

<http://www.sqlserver-training.com/how-to-use-offset-fetch-option-in-sql-server-order-byclause/>

[http://www.sqlservercentral.com/blogs/juggling\\_with\\_sql/2011/11/30/using-offset-and-fetch/](http://www.sqlservercentral.com/blogs/juggling_with_sql/2011/11/30/using-offset-and-fetch/)

### **Question No : 22 - (Topic 2)**

You need to ensure that if any of the statements in usp\_UpdateSpeakerName return an error message, all of the changes executed by usp\_UpdateSpeakerName are not committed to the database.

What should you do in Procedures.sql? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Add the following at line 17:  
`ROLLBACK TRANSACTION`
- B. Add the following at line 05:  
`BEGIN TRANSACTION SpeakerUpdate`
- C. Add the following at line 05:  
`SAVE TRANSACTION SpeakerUpdate`
- D. Add the following at line 17:  
`ROLLBACK TRANSACTION SpeakerUpdate`
- E. Add the following at line 07:  
`BEGIN TRANSACTION`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer: B,D**

**Question No : 23 - (Topic 2)**

Developers report that `usp_UpdateSessionRoom` periodically returns error 3960.

You need to prevent the error from occurring. The solution must ensure that the stored procedure returns the original values to all of the updated rows.

What should you configure in `Procedures.sql`?

- A. Replace line 46 with the following code:  
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
- B. Replace line 46 with the following code:  
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
- C. Move the SELECT statement at line 49 to line 57.
- D. Move the SET statement at line 46 to line 53.

**Answer: A**

### **Topic 3, Scenario 3**

#### **Application Information**

You have two servers named SQL1 and SQL2. SQL1 has SQL Server 2012 Enterprise installed. SQL2 has SQL Server 2008 Standard installed.

You have an application that is used to manage employees and office space. Users report that the application has many errors and is very slow.

You are updating the application to resolve the issues.

You plan to create a new database on SQL1 to support the application. The script that you plan to use to create the tables for the new database is shown in Tables.sql. The script that you plan to use to create the stored procedures for the new database is shown in StoredProcedures.sql. The script that you plan to use to create the indexes for the new database is shown in Indexes.sql.

A database named DB2 resides on SQL2. DB2 has a table named EmployeeAudit that will audit changes to a table named Employees.

A stored procedure named usp\_UpdateEmployeeName will be executed only by other stored procedures. The stored procedures executing usp\_UpdateEmployeeName will always handle transactions.

A stored procedure named usp\_SelectEmployeesByName will be used to retrieve the names of employees. Usp\_SelectEmployeesByName can read uncommitted data.

A stored procedure named usp\_GetFutureOfficeAssignments will be used to retrieve office assignments that will occur in the future.

#### **StoredProcedures.sql**

```
01 CREATE PROCEDURE usp_UpdateEmployeeName
02   @EmployeesInfo EmployeesInfo READONLY
03 AS
04
05 BEGIN TRY
06
07 UPDATE Employees
08 SET LastName = ei.LastName
09 FROM Employees e
10   INNER JOIN @ EmployeesInfo ei ON e.EmployeeID = ei.EmployeeID;
11
12 INSERT INTO SQL2.DB2.dbo.EmployeeAudit (EmployeeID, LastName)
13 SELECT EmployeeID, LastName
14 FROM @EmployeesInfo;
15
16 END TRY
17 BEGIN CATCH
18
19 END CATCH;
20
21 GO
22
23 CREATE PROCEDURE usp_SelectEmployeesByName
24   @LastName nvarchar(100)
25 AS
26 SELECT EmployeeID,
27   FirstName,
28   LastName
29 FROM Employees
30 WHERE LastName LIKE @LastName + '%'
31
32 GO
33
34 CREATE PROCEDURE usp_UpdateOffice
35   @OfficeID int,
36   @EmployeeID int
37 AS
38 SET TRANSACTION ISOLATION LEVEL SNAPSHOT
39 BEGIN TRANSACTION;
40
41 SELECT OfficeID,
42   OfficeName
43 FROM Offices
44 WHERE EmployeeID = @EmployeeID;
45
46 UPDATE Offices
47 SET EmployeeID = @EmployeeID,
48   StartDate = GETDATE()
49 WHERE OfficeID = @OfficeID;
50
51 COMMIT TRANSACTION;
52
53 CREATE PROCEDURE usp_GetFutureOfficeAssignments
54 AS
55 SELECT EmployeeID,
56   OfficeID,
57   StartDate
58 FROM Offices
59 WHERE StartDate > GETDATE();
60 GO
61
```

Indexes.sql

```
01 CREATE INDEX IX_Offices ON Offices
02 (EmployeeID, StartDate)
03 INCLUDE (OfficeID)
04
05 GO
06
07 CREATE INDEX IX_Employees ON Employees
08 (LastName);
09 GO
10
```

### Tables.sql

```
01 CREATE DATABASE HumanResources;
02 GO
03
04 ALTER DATABASE HumanResources
05 SET ALLOW_SNAPSHOT_ISOLATION ON;
06 GO
07
08 USE HumanResources
09 GO
10
11 CREATE TABLE Employees
12 (
13     EmployeeID int IDENTITY(1,1) NOT NULL,
14     FirstName nvarchar(100) NOT NULL,
15     LastName nvarchar(100) NOT NULL,
16
17 );
18 GO
19
20 CREATE TABLE Offices
21 (
22     OfficeID int IDENTITY(1,1) NOT NULL,
23     EmployeeID int NOT NULL,
24     OfficeName nvarchar(100) NOT NULL,
25     StartDate datetime NOT NULL
26 );
27 GO
```

You need to create the object used by the parameter of usp\_UpdateEmployeeName.

Which code segment should you use?

- A. CREATE XML SCHEMA COLLECTION EmployeesInfo
- B. CREATE TYPE EmployeesInfo AS Table
- C. CREATE SCHEMA EmployeesInfo
- D. CREATE TABLE EmployeesInfo

**Answer: B**

**Explanation:**

Example Usage of Table-Valued Parameters (Database Engine)

<http://msdn.microsoft.com/en-us/library/bb510489.aspx> (Benefits of using Table-Valued Parameters)

```
/* Create a table type. */
```

```
CREATE TYPE LocationTableType AS TABLE
```

```
( LocationName VARCHAR(50)
```

```
, CostRate INT );
```

```
GO
```

```
/* Create a procedure to receive data for the table-valued parameter. */
```

```
CREATE PROCEDURE dbo. usp_InsertProductionLocation
```

```
@TVP LocationTableType READONLY
```

```
AS
```

```
SET NOCOUNT ON
```

```
INSERT INTO AdventureWorks2012.Production.Location
```

```
(Name
```

```
,CostRate
```

```
,Availability
```

```
,ModifiedDate)
```

```
SELECT *, 0, GETDATE()
```

```
FROM @TVP;
```

```
GO
```

Also:

<http://msdn.microsoft.com/en-us/library/ms175007.aspx>(CREATE TYPE \*tabletypename\*

```
AS
```

```
TABLE)
```

<http://msdn.microsoft.com/en-us/library/ms175010.aspx>(table data types)

Wrong Answers:

<http://msdn.microsoft.com/en-us/library/ms174979.aspx>(CREATE TABLE)

<http://msdn.microsoft.com/en-us/library/ms189462.aspx>(CREATE SCHEMA)

<http://msdn.microsoft.com/en-us/library/ms176009.aspx>(CREATE XML SCHEMA COLLECTION)

**Question No : 25 - (Topic 3)**

You need to modify usp\_SelectEmployeesByName to support server-side paging. The solution must minimize the amount of development effort required.

What should you add to usp\_SelectEmployeesByName?

- A. A table variable
- B. The ROWNUMBER keyword
- C. An OFFSET-FETCH clause
- D. A recursive common table expression

**Answer: C**

**Explanation:**

<http://www.mssqltips.com/sqlservertip/2696/comparing-performance-for-different-sql-serverpaging-methods/>

<http://msdn.microsoft.com/en-us/library/ms188385.aspx>

<http://msdn.microsoft.com/en-us/library/ms180152.aspx>

<http://msdn.microsoft.com/en-us/library/ms186243.aspx>



<http://msdn.microsoft.com/en-us/library/ms186734.aspx>

<http://www.sqlserver-training.com/how-to-use-offset-fetch-option-in-sql-server-order-byclause/>

[http://www.sqlservercentral.com/blogs/juggling\\_with\\_sql/2011/11/30/using-offset-and-fetch/](http://www.sqlservercentral.com/blogs/juggling_with_sql/2011/11/30/using-offset-and-fetch/)

**Question No : 26 - (Topic 3)**

You need to add a new column named Confirmed to the Employees table. The solution must meet the following requirements:

-  Have a default value of TRUE.
-  Minimize the amount of disk space used.



Which code segment should you use?

- A. 

```
ALTER TABLE Employees  
ADD Confirmed char(1) DEFAULT '1';
```
- B. 

```
ALTER TABLE Employees  
ADD Confirmed char(1) DEFAULT '0';
```
- C. 

```
ALTER TABLE Employees  
ADD Confirmed bit DEFAULT 0;
```
- D. 

```
ALTER TABLE Employees  
ADD Confirmed bit DEFAULT 1;
```

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

**Answer: D**

**Question No : 27 - (Topic 3)**

You need to provide referential integrity between the Offices table and Employees table.

Which code segment or segments should you add at line 27 of Tables.sql? (Each correct answer presents part of the solution. Choose all that apply.)

- A. ALTER TABLE dbo.Offices ADD CONSTRAINT  
PK\_Offices\_EmployeeID PRIMARY KEY (EmployeeID);
- B. ALTER TABLE dbo.Employees ADD CONSTRAINT  
FK\_Employees\_Offices FOREIGN KEY (OfficeID)  
REFERENCES dbo.Offices (OfficeID);
- C. ALTER TABLE dbo.Employees ADD CONSTRAINT  
PK\_Employees\_EmployeeID PRIMARY KEY (EmployeeID);
- D. ALTER TABLE dbo.Offices ADD CONSTRAINT  
FK\_Offices\_Employees FOREIGN KEY (EmployeeID)  
REFERENCES dbo.Employees (EmployeeID);

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

**Answer: C,D**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms189049.aspx>

### **Question No : 28 - (Topic 3)**

You need to recommend a solution to ensure that SQL1 supports the auditing requirements of usp\_UpdateEmployeeName.

What should you include in the recommendation?

- A. Change data capture
- B. Change tracking
- C. Transactional replication
- D. The Distributed Transaction Coordinator (DTC)

**Answer: D**

**Question No : 29 - (Topic 3)**

You execute usp\_SelectEmployeesByName multiple times, passing strings of varying lengths to @LastName. You discover that usp\_SelectEmployeesByName uses inefficient execution plans.

You need to update usp\_SelectEmployeesByName to ensure that the most efficient execution plan is used.

What should you add at line 31 of StoredProcedures.sql?

- A. OPTION (ROBUST PLAN)
- B. OPTION (OPTIMIZE FOR UNKNOWN)
- C. OPTION (KEEP PLAN)
- D. OPTION (KEEPFIXED PLAN)

**Answer: B**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms181714.aspx>

**Topic 4, Scenario 4**

**Application Information**

You are a database administrator for a manufacturing company.

You have an application that stores product data. The data will be converted to technical diagrams for the manufacturing process.

The product details are stored in XML format. Each XML must contain only one product that has a root element named Product. A schema named Production.ProductSchema has been created for the products.xml.

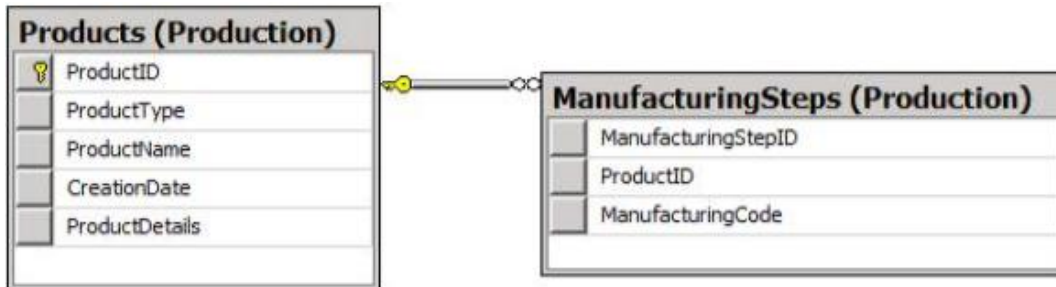
You develop a Microsoft .NET Framework assembly named ProcessProducts.dll that will be used to convert the XML files to diagrams. The diagrams will be stored in the database as images. ProcessProducts.dll contains one class named ProcessProduct that has a method name of Convert(). ProcessProducts.dll was created by using a source code file named ProcessProduct.cs.

All of the files are located in C:\Products\.

The application has several performance and security issues.

You will create a new database named ProductsDB on a new server that has SQL Server 2012 installed. ProductsDB will support the application.

The following graphic shows the planned tables for ProductsDB:



You will also add a sequence named Production.ProductID\_Seq.

You plan to create two certificates named DBCert and ProductsCert. You will create ProductsCert in master. You will create DBCert in ProductsDB.

You have an application that executes dynamic T-SQL statements against ProductsDB. A sample of the queries generated by the application appears in Dynamic.sql.

### Application Requirements

The planned database has the following requirements:

- ✍ All stored procedures must be signed.
- ✍ The amount of disk space must be minimized.
- ✍ Administrative effort must be minimized at all times.
- ✍ The original product details must be stored in the database.
- ✍ An XML schema must be used to validate the product details.
- ✍ The assembly must be accessible by using T-SQL commands.
- ✍ A table-valued function will be created to search products by type.
- ✍ Backups must be protected by using the highest level of encryption.
- ✍ Dynamic T-SQL statements must be converted to stored procedures.
- ✍ Indexes must be optimized periodically based on their fragmentation.
- ✍ Manufacturing steps stored in the ManufacturingSteps table must refer to a product by the same identifier used by the Products table.

### ProductDetails\_Insert.sql

```
01 CREATE PROCEDURE Production.ProductDetails_Insert @XML nvarchar(1000)
02 AS
03 DECLARE @handle INT;
04 DECLARE @document nvarchar(1000);
05 SET @document = @XML;
06
07 EXEC sp_xml_preparedocument @handle OUTPUT, @document;
08
09 INSERT INTO PRODUCTSDB.Production.Invoices (
10     ProductID,
11     ProductDetails,
12     ProductType,
13     ProductName,
14     CreationDate
15 )
16 SELECT (NEXT VALUE FOR Production.ProductID_Seq),
17     @XML, * FROM OPENXML (@handle, '/Invoice',2)
18     WITH (
19         ProductType nvarchar(11) 'ProductType/ID',
20         ProductName nvarchar(50) '@ProductName',
21         CreationDate date 'CreationDate'
22     );
23
24 EXEC sp_xml_removedocument @handle;
```

### Product, xml

All product types are 11 digits. The first five digits of the product id reference the category of the product and the remaining six digits are the subcategory of the product.

The following is a sample customer invoice in XML format:

```
01 <?xml version="1.0"?>
02 <Product ProductName="Widget">
03     <ProductType ID="00156590099" />
04     <CreationDate>2011-08-05</CreationDate>
05 </Invoice>
```

### ProductsByProductType.sql

```
01 (SELECT ProductID,
02     ProductType,
03     CreationDate
04     FROM Production.Products
05     WHERE ProductType=@ProductType);
```

### Dynamic.sql

```
01 DECLARE @tsql AS nvarchar(500);
02 DECLARE @ProductType AS varchar(11), @CreationDate AS date;
03
04 SET @sqlstring=N'SELECT ProductID, ProductType, CreationDate
05   FROM Production.Product
06   WHERE ProductID=@ProductID AND CreationDate > @CreationDate;';
07
08 EXEC sys.sp_executesql
09   @statement=@sqlstring,
10   @params=N'@ ProductType AS varchar(11), @CreationDate AS date',
11   @ProductType=00125061246, @Total='2012-05-10';
```

## Category FromType.sql

```
01 CREATE FUNCTION CategoryFromType (@Type varchar(11)) RETURNS nvarchar(20)
02 AS
03 BEGIN
04   DECLARE @Category AS varchar(20);
05   SET @Category = LEFT(@Category,5);
06   SELECT @Category = CASE @Type
07     WHEN '00001'
08       THEN 'Bikes'
09     WHEN '00002'
10       THEN 'Wheels'
11     ...
12     ELSE 'Other'
13   END;
14 RETURN @Category;
15 END;
```

## IndexManagement.sql

```

01 DECLARE @IndexTable TABLE (
02   TableName varchar(100), IndexName varchar(100), Fragmentation int, RowNumber int
03 );
04 DECLARE @TableName sysname, @IndexName sysname, @Fragmentation int,
05   @RowNumber int, @sqlcommand varchar(1000);
06
07 INSERT INTO @IndexTable (TableName, IndexName, Fragmentation, Rownumber)
08   SELECT OBJECT_NAME(i.Object_id),
09     i.name AS IndexName,
10     indexstats.avg_fragmentation_in_percent,
11     ROW_NUMBER() OVER(ORDER BY i.name DESC) AS 'RowNumber'
12   FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, 'DETAILED')
13   AS indexstats INNER JOIN sys.indexes AS i
14     ON i.OBJECT_ID = indexstats.OBJECT_ID AND i.index_id = indexstats.index_id;
15
16 DECLARE @counter int = 0;
17
18 WHILE @counter < (SELECT RowNumber FROM @indextable)
19   BEGIN
20     SET @counter = @counter + 1;
21     WITH t AS (
22       SELECT TableName, IndexName, Fragmentation
23       FROM @IndexTable WHERE RowNumber = @counter
24     )
25     SELECT
26       @TableName= TableName,
27       @IndexName = IndexName,
28       @Fragmentation = Fragmentation
29     FROM t;
30
31     IF @Fragmentation <= 30
32       BEGIN
33         SET @sqlCommand =
34           N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REORGANIZE';
35         EXEC sp_executesql @sqlCommand;
36       END;
37     ELSE
38       BEGIN
39         SET @sqlCommand=N'ALTER INDEX '+@indexName+N' ON '+@TableName+N' REBUILD';
40         EXEC sp_executesql @sqlCommand;
41       END;
42     END;

```

### Question No : 30 - (Topic 4)

You execute IndexManagement.sql and you receive the following error message: "Msg 512, Level 16, State 1, Line 12

Subquery 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 SUM(RowNumber) FROM @indextable)
- C. WHILE COUNT(@RowNumber) < (SELECT @counter FROM @indextable)

D. WHILE @counter < (SELECT COUNT(LineNumber) FROM @indextable)

**Answer: D**

**Question No : 31 - (Topic 4)**

An administrator provides a digital certificate named ServerCert.

You need to implement Transparent Data Encryption (TDE) on ProductsDB.

Which code segment should you use?

**A.** USE PRODUCTSDB;

GO

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = TRIPLE_DES_3KEY  
ENCRYPTION BY SERVER CERTIFICATE DBCERT;
```

GO

```
ALTER DATABASE PRODUCTSDB SET ENCRYPTION ON;
```

GO

**B.** USE PRODUCTSDB;

GO

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = TRIPLE_DES_3KEY  
ENCRYPTION BY SERVER CERTIFICATE PRODUCTSCERT;
```

GO

```
ALTER DATABASE PRODUCTSDB SET ENCRYPTION ON;
```

GO

**C.** USE PRODUCTSDB;

GO

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE PRODUCTSCERT;
```

GO

```
ALTER DATABASE PRODUCTSDB SET ENCRYPTION ON;
```

GO

**D.** USE PRODUCTSDB;

GO

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE DBCERT;
```

GO

```
ALTER DATABASE PRODUCTSDB SET ENCRYPTION ON;
```

GO

**Answer: C**



**Explanation:**

<http://msdn.microsoft.com/en-us/library/bb934049.aspx>

**Question No : 32 - (Topic 4)**

You are planning the ManufacturingSteps table.

You need to define the ProductID column in the CREATE TABLE statement.

Which code segment should you use?

- A. `ProductID bigint  
DEFAULT (NEXT VALUE FOR Production.ProductID_Seq) NOT NULL,`
- B. `ProductID bigint FOREIGN KEY REFERENCES  
Production.Product(ProductID) NOT NULL,`
- C. `ProductID bigint DEFAULT  
((NEXT VALUE FOR Production.ProductID_Seq OVER  
(ORDER BY ManufacturingStepID))) NOT NULL,`
- D. `ProductID bigint DEFAULT  
((NEXT VALUE FOR Production.ProductID_Seq OVER  
(ORDER BY ManufacturingStepID)))  
NOT NULL FOREIGN KEY REFERENCES  
Production.Product(ProductID),`

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

**Answer: B**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms189049.aspx>

<http://msdn.microsoft.com/en-us/library/ms179610.aspx>

<http://msdn.microsoft.com/en-us/library/ff878370.aspx>

**Question No : 33 - (Topic 4)**

You are testing disaster recovery procedures.

When you attempt to restore ProductsDB to another server, you receive the following error message: "Msg

33111, Level 16, State 3, Line 5

Cannot find server certificate with thumbprint '  
0x9D876A3468B911EIBA4FCBF4724019B\

Msg 3013, Level 16, State 1, Line 5

RESTORE DATABASE is terminating abnormally."

You need to ensure that you can restore ProductsDB to another server.

Which code segment should you execute on the other server?

- A. 

```
RESTORE CERTIFICATE DBCERT
FROM FILE='DBCERT.CER'
WITH PRIVATE KEY (FILE = 'c:\DBCERT.KEY',
DECRYPTION BY PASSWORD = 'SecretP@ss');
```
- B. 

```
CREATE CERTIFICATE PRODUCTSCERT
ENCRYPTION BY PASSWORD = 'SecretP@ss'
WITH SUBJECT = 'SecurityCertificate';
```
- C. 

```
CREATE CERTIFICATE DBCERT
ENCRYPTION BY PASSWORD = 'SecretP@ss'
WITH SUBJECT = 'SecurityCertificate';
```
- D. 

```
CREATE CERTIFICATE PRODUCTSCERT
FROM FILE='PRODUCTSCERT.CER'
WITH PRIVATE KEY (FILE = 'c:\PRODUCTSCERT.KEY',
DECRYPTION BY PASSWORD = 'SecretP@ss');
```

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

**Answer: D**

### Question No : 34 - (Topic 4)

You need to prepare the database to use the .NET Framework ProcessProducts component.

Which code segments should you execute? (Each correct answer presents part of the solution. Choose all that apply.)

- A. 

```
CREATE PROCEDURE Production.ProcessProduct (
    @ProductID int, @ProductType varchar(11)
)
AS EXTERNAL NAME ProductionAssembly.ProcessProducts.Process;
```
- B. 

```
EXEC sp_recompile @objname = 'Production.ProcessProduct';
```
- C. 

```
RECONFIGURE;
```
- D. 

```
Exec SP_CONFIGURE 'clr enabled', '1';
```
- E. 

```
CREATE ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.DLL'
```
- F. 

```
CREATE ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.cs';
```
- G. 

```
CREATE TYPE Production.ProcessProduct
EXTERNAL NAME ProductionAssembly.ProcessProductss.Process;
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F
- G. Option G

**Answer: A,C,D,E**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms131048.aspx>

<http://msdn.microsoft.com/en-us/library/ms131052.aspx>

<http://msdn.microsoft.com/en-us/library/ms189524.aspx>

<http://msdn.microsoft.com/en-us/library/ms345106.aspx>

<http://msdn.microsoft.com/en-us/library/ms131107.aspx>

**Question No : 35 - (Topic 4)**

Which data type should you use for ProductType?

- A. varchar(11)
- B. nvarchar(11)
- C. char(11)
- D. bigint

**Answer: C**

**Question No : 36 - (Topic 4)**

While testing the CategoryFromType function, you discover that the function is returning 'Other'.

You need to update CategoryFromType to return the category name.

Which line of code should you modify in CategoryFromType.sql?

- A. 04
- B. 05
- C. 12
- D. 14

**Answer: B**

**Question No : 37 - (Topic 4)**

You need to create a function that will use a SELECT statement in ProductsByProductType.sql.

Which code segment should you use to complete the function?

- A. 

```
CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS @tblInvoices TABLE (ProductID bigint, ProductType varchar(11), CreationDate
date)
AS
INSERT INTO @tblInvoices
```
- B. 

```
CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS TABLE
AS
RETURN
```
- C. 

```
CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS @tblInvoices TABLE (ProductID bigint, ProductType varchar(11), CreationDate
date)
AS
```
- D. 

```
CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS xml
AS
RETURN
```

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

**Answer: B**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/ms191320.aspx>

<http://msdn.microsoft.com/en-us/library/ms186755.aspx>

### Question No : 38 - (Topic 4)

You need to modify Production.ProductDetails\_Insert to comply with the application requirements.

Which code segment should you execute?

- A. OPEN PRODUCTSCERT;  
ALTER PROCEDURE Production.ProductDetails\_Insert  
WITH ENCRYPTION;  
CLOSE PRODUCTSCERT;
- B. OPEN DBCERT;  
ALTER PROCEDURE Production.ProductDetails\_Insert  
WITH ENCRYPTION;  
CLOSE DBCERT;
- C. ADD SIGNATURE TO Production.ProductDetails\_Insert  
BY CERTIFICATE DBCERT;
- D. ADD SIGNATURE TO Production.ProductDetails\_Insert  
BY CERTIFICATE PRODUCTSCERT;

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

**Answer: C**

**Explanation:**

<http://msdn.microsoft.com/en-us/library/bb669102.aspx>

#### **Question No : 39 - (Topic 4)**

Which code segment should you use to define the ProductDetails column?

- A. ProductDetails xml (DOCUMENT Production.ProductDetailsSchema) NULL
- B. ProductDetails xml NULL
- C. ProductDetails xml (CONTENT Production.ProductDetailsSchema) NULL
- D. ProductDetails varchar(MAX) NULL

**Answer: D**

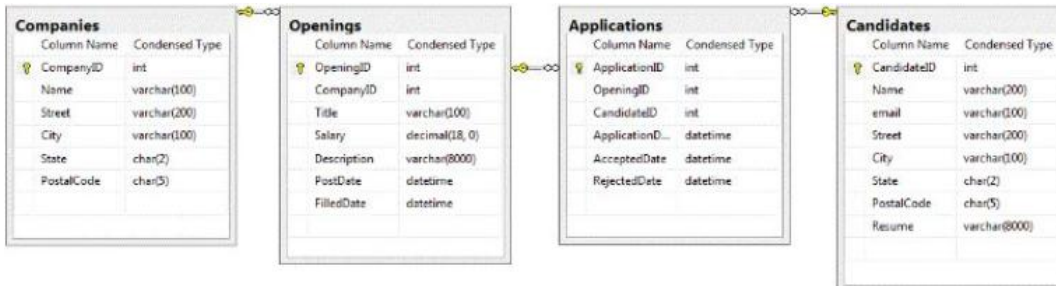
**Topic 5, Litware, Inc**

## Overview

### General Overview

You are a database developer for a company named Litware, Inc. Litware has a main office in Miami.

Litware has a job posting web application named WebApp1. WebApp1 uses a database named DB1. DB1 is hosted on a server named Server1. The database design of DB1 is shown in the exhibit. (Click the Exhibit button.)



WebApp1 allows a user to log on as a job poster or a job seeker. Candidates can search for job openings based on keywords, apply to an opening, view their application, and load their resume in Microsoft Word format. Companies can add a job opening, view the list of candidates who applied to an opening, and mark an application as denied.

### Users and Roles

DB1 has five database users named Company, CompanyWeb, Candidate, CandidateWeb, and Administrator.

DB1 has three user-defined database roles. The roles are configured as shown in the following table.

Role name	Role member
Companies	Company Administrator CompanyWeb
Candidates	Candidate Administrator CandidateWeb
Administrators	Administrator

### Keyword Search

The keyword searches for the job openings are performed by using the following stored procedure named usp\_GetOpenings:

```
01 CREATE PROCEDURE usp_GetOpenings
02     @keyword varchar(max),
03     @minsalary decimal(18,0) = 0
04 AS
05 DECLARE @plural varchar(max);
06 DECLARE @ing varchar(max);
07 SET @plural = @keyword + 's';
08 SET @ing = @keyword + 'ing';
09 SELECT o.Title, o.Salary, c.Name, o.Description
10 FROM Openings o
11 INNER JOIN Companies c ON c.CompanyID = o.CompanyID
12 WHERE (o.Description LIKE '%'+@keyword+'%'
13        OR o.Description LIKE '%'+@plural+'%'
14        OR o.Description LIKE '%'+@ing+'%')
15 AND o.Salary >= @minsalary;
```

### Opening Update

Updates to the Openings table are performed by using the following stored procedure named usp\_UpdateOpening:

```
01 CREATE PROCEDURE usp_UpdateOpening
02     @openingID int,
03     @title varchar(100),
04     @salary decimal(18,0),
05     @description varchar(8000)
06 AS
07 UPDATE Openings
08 SET Title = @title,
09     Salary = @salary,
10     Description = @description
11 WHERE OpeningID = @openingID;
```

## Problems and Reported Issues

### Concurrency Problems

You discover that deadlocks frequently occur.

You identify that a stored procedure named usp\_AcceptCandidate and a stored procedure named usp\_UpdateCandidate generate deadlocks. The following is the code for usp\_AcceptCandidate:



```
01 CREATE PROCEDURE usp_AcceptCandidate
02     @applicationID int
03 AS
04     DECLARE @date datetime;
05     SET @date = GETDATE();
06     UPDATE Applications
07         SET AcceptedDate = @date
08         WHERE ApplicationID = @applicationID;
09     SELECT Name, email
10     FROM Candidates c
11     INNER JOIN Applications a
12         ON a.CandidateID = c.CandidateID
13     WHERE a.AcceptedDate IS NOT NULL;
```

### Salary Query Issues

Users report that when they perform a search for job openings without specifying a minimum salary, only job openings that specify a minimum salary are displayed.

### Log File Growth Issues

The current log file for DB1 grows constantly. The log file fails to shrink even when the daily SQL Server Agent Shrink Database task runs.

### Performance Issues

You discover that a stored procedure named usp\_ExportOpenings takes a long time to run and executes a table scan when it runs.

You also discover that the usp\_GetOpenings stored procedure takes a long time to run and that the non-clustered index on the Description column is not being used.

### Page Split Issues

On DB1, many page splits per second spike every few minutes.

### Requirements

#### Security and Application Requirements

Litware identifies the following security and application requirements:

- ✍ Only the Administrator, Company, and CompanyWeb database users must be able to execute the usp\_UpdateOpening stored procedure.
- ✍ Changes made to the database must not affect WebApp1.

#### Locking Requirements

Litware identifies the following locking requirements:

- ✍ The usp\_GetOpenings stored procedure must not be blocked by the usp\_UpdateOpening stored procedure.
- ✍ If a row is locked in the Openings table, usp\_GetOpenings must retrieve the latest

version of the row, even if the row was not committed yet.

### Integration Requirements

Litware exports its job openings to an external company as XML data. The XML data uses the following format:

```
<Opening title="web programmer" salary="75000">  
  This is the description of the opening  
</Opening>
```

A stored procedure named `usp_ExportOpenings` will be used to generate the XML data. The following is the code for `usp_ExportOpenings`:


```
01 CREATE PROCEDURE usp_ExportOpenings  
02   @lastPost datetime  
03 AS  
04 SELECT Description  
05    , Title  
06    , Salary  
07 FROM Openings  
08 WHERE PostDate > @lastPost  
09    AND FilledDate IS NULL
```


The stored procedure will be executed by a SQL Server Integration Services (SSIS) package named `Package1`.

The XML data will be written to a secured folder named `Folder1`. Only a dedicated Active Directory account named `Account1` is assigned the permissions to read from or write to `Folder1`.

### Refactoring Requirements

Litware identifies the following refactoring requirements:

 New code must be written by reusing the following query:

 The results from the query must be able to be joined to other queries.

### Upload Requirements

Litware requires users to upload their job experience in a Word file by using WebApp1. WebApp1 will send the Word file to DB1 as a stream of bytes. DB1 will then convert the Word file to text before the contents of the Word file is saved to the Candidates table.

A database developer creates an assembly named Conversions that contains the following:

- ✍ A class named Convert in the SqlConversions namespace
- ✍ A method named ConvertToText in the Convert class that converts Word files to text

The ConvertToText method accepts a stream of bytes and returns text. The method is used in the following stored procedure:

```
01 CREATE PROCEDURE usp_UpdateCandidate
02     @candidateID int,
03     @wordResume varbinary(max)
04 AS
05 DECLARE @textResume varchar(8000);
06 SET @textResume = ConvertToText(@wordResume);
07 UPDATE Candidates SET Resume = @textResume
08     WHERE CandidateID = @candidateID;
09 SELECT OpeningID, ApplicationDate
10 FROM Applications
11 WHERE CandidateID = @candidateID;
```

### Job Application Requirements

A candidate can only apply to each job opening once.

### Data Recovery Requirements

All changes to the database are performed by using stored procedures. WebApp1 generates a unique transaction ID for every stored procedure call that the application makes to the database.

If a server fails, you must be able to restore data to a specified transaction.

### Question No : 40 - (Topic 5)

You need to implement a change to usp\_ExportOpenings that meets the integration requirements.

What should you modify in usp\_ExportOpenings? (Each correct answer presents part of the solution. Choose all that apply?)

A. To the end of line 04, add [Opening].

- B. To the end of line 05, add [Opening! title].
- C. To line 10, add FOR XML RAW.
- D. To line 10, add FOR XMLEXPLICIT.
- E. To line 10, add FOR XML AUTO.
- F. To the end of line 04, add [Opening!ELEMENT].
- G. To the end of line 06, add [Opening!salary!ELEMENT].
- H. To the end of line 05, add [Opening!title!ELEMENT].
- I. To the end of line 06, add [Opening! salary].

**Answer: A,B,E,I**

**Explanation:**

The AUTO mode generates nesting in the resulting XML by using heuristics based on the way the SELECT statement is specified. You have minimal control over the shape of the XML generated. The nested FOR XML queries can be written to generate XML hierarchy beyond the XML shape that is generated by AUTO mode heuristics.

**Question No : 41 - (Topic 5)**

You need to implement a solution that resolves the salary query issue.

Which statement should you execute on DB1?

- A. UPDATE Openings SET Salary=0 WHERE Salary IS NULL;  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
MODIFY COLUMN Salary NOT NULL;  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
ADD CONSTRAINT DF\_SALARY  
DEFAULT 0 FOR Salary;  
GO
- B. ALTER TABLE Openings  
WITH NOCHECK  
ADD CONSTRAINT DF\_SALARY  
DEFAULT 0 FOR Salary;  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
MODIFY COLUMN Salary NULL;  
GO  
UPDATE Openings SET Salary=0 WHERE Salary IS NULL;  
GO
- C. UPDATE Openings SET Salary=0 WHERE Salary IS NULL;  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
ADD CONSTRAINT CT\_SALARY  
CHECK (Salary>=0);  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
MODIFY COLUMN Salary NOT NULL;  
GO
- D. ALTER TABLE Openings  
WITH NOCHECK  
ADD CONSTRAINT CT\_SALARY  
CHECK (Salary>=0);  
GO  
ALTER TABLE Openings  
WITH NOCHECK  
MODIFY COLUMN Salary NOT NULL;  
GO  
UPDATE Openings SET Salary=0 WHERE Salary IS NULL;  
GO

A. Option A

- B. Option B
- C. Option C
- D. Option D

**Answer: A**

**Question No : 42 - (Topic 5)**

You need to identify the cause of the page split issues.

Which SQL Server feature should you use?

- A. DBCC REINDEX
- B. SQL Server Profiler
- C. Extended Events
- D. DBCC TRACEOFF

**Answer: C**

**Question No : 43 - (Topic 5)**

You need to implement a solution that addresses the upload requirements.

Which code segment should you use to implement the Conversions assembly?

- A. 

```
CREATE FUNCTION ConvertToText (@wordResume varchar(8000))
    RETURNS varbinary(max)
    AS EXTERNAL NAME SqlConversions.Conversions.ConvertToText;
```
- B. 

```
CREATE PROCEDURE ConvertToText (@wordResume varbinary(max))
    AS EXTERNAL NAME Conversions.SqlConversions.ConvertToText;
```
- C. 

```
CREATE PROCEDURE ConvertToText (@wordResume varchar(8000))
    AS EXTERNAL NAME SqlConversions.Conversions.ConvertToText;
```
- D. 

```
CREATE FUNCTION ConvertToText (@wordResume varbinary(max))
    RETURNS varchar(8000)
    AS EXTERNAL NAME Conversions.SqlConversions.ConvertToText;
```

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

**Answer: D**

**Question No : 44 - (Topic 5)**

You need to implement a solution that meets the security requirements.

Which statement should you execute?

- A. `REVOKE EXEC ON usp_UpdateOpening FROM Candidates;`
- B. `DENY EXEC ON usp_UpdateOpening TO Candidates;`
- C. `ALTER PROCEDURE usp_UpdateOpening  
@openingIDint,  
@titlevarchar(100),  
@salarydecimal(18,0),  
@descriptionvarchar(8000)  
WITH EXECUTE AS Administrator  
AS  
...`
- D. `ALTER PROCEDURE usp_UpdateOpening  
@openingIDint,  
@titlevarchar(100),  
@salarydecimal(18,0),  
@descriptionvarchar(8000)  
WITH EXECUTE AS Company  
AS  
...`

- A. Option A
- B. Option B
- C. Option C

D. Option D

**Answer: A**

**Question No : 45 - (Topic 5)**

You need to implement a solution that meets the data recovery requirements.

You update each stored procedure to accept a parameter named @transactionID.

What should you add next to the beginning of each stored procedure?

- A. SAVE TRANSACTION WITH MARK @transactionID
- B. ROLLBACK DISTRIBUTED TRANSACTION @transactionID
- C. BEGIN TRANSACTION WITH MARK @transactionID
- D. COMMIT TRANSACTION @transactionID

**Answer: C**

**Question No : 46 - (Topic 5)**

You need to create a script that automates the export of the XML data. The script must meet the integration requirements.

What should you include in the script?

- A. The CREATE SERVER ROLE command and the sp\_reassign\_proxy, sp\_add\_job, sp\_add\_jobstep, and sp\_grant\_login\_to\_proxy system stored procedures.
- B. The CREATE CREDENTIAL command and the sp\_add\_proxy, sp\_add\_job, sp\_add\_jobstep, and sp\_grant\_proxy\_to\_subsystem system stored procedures.
- C. The CREATE CREDENTIAL command and the sp\_reassign\_proxy, sp\_add\_job, sp\_add\_jobstep, and sp\_grant\_login\_to\_proxy system stored procedures.
- D. The CREATE SERVER ROLE command and the sp\_add\_proxy, sp\_add\_job, sp\_add\_jobstep, and sp\_grant\_proxy\_to\_subsystem system stored procedures.

**Answer: B**



**Question No : 47 - (Topic 5)**

You need to implement a solution that meets the locking requirements.

Which line of code should you modify?

- A. Change line 07 in usp\_UpdateOpening to:  
UPDATE Openings WITH (UPDLOCK)
- B. Change line 09 in usp\_GetOpenings to:  
FROM Openings o (ROWLOCK)
- C. Change line 07 in usp\_UpdateOpening to:  
UPDATE Openings WITH (READPAST)
- D. Change line 09 in usp\_GetOpenings to:  
FROM Openings o (NOLOCK)

**Answer: D**

**Question No : 48 - (Topic 5)**

You need to resolve the performance issues of the usp\_ExportOpenings stored procedure. The solution must minimize the amount of hard disk space used.

Which statement should you execute on DB1?

- A. EXEC sp\_dboption 'DB1', 'auto create statistics', 'TRUE';
- B. CREATE INDEX IX\_Exp\_Openings ON Openings(PostDate, FilledDate) INCLUDE (Description, Title, Salary);
- C. CREATE INDEX IX\_Exp\_Openings ON Openings(PostDate) INCLUDE (Description, Title, Salary) WHERE FilledDate IS NULL;
- D. EXEC sp\_recompile 'usp\_ExportOpenings';

**Answer: C**

**Question No : 49 - (Topic 5)**

You need to recommend a solution that meets the concurrency problems.

What should you include in the recommendation?

- A. Modify the stored procedures to use the SERIALIZABLE isolation level.
- B. Modify the order in which usp\_AcceptCandidate accesses the Applications table and the Candidates table.
- C. Modify the order in which usp\_UpdateCandidate accesses the Applications table and the Candidates table.
- D. Modify the stored procedures to use the REPEATABLE READ isolation level.

**Answer: C**

### **Question No : 50 - (Topic 5)**

You need to implement a solution that meets the job application requirements.

What should you do?

- A. Create a one-to-one relationship between the Openings table and the Applications table.
- B. Create a one-to-one relationship between the Candidates table and the Applications table.
- C. Add a UNIQUE constraint to the Applications table on the ApplicationID column and CandidateID column.
- D. Add a UNIQUE constraint to the Applications table on the OpeningID column and the CandidateID column.

**Answer: D**

### **Question No : 51 - (Topic 5)**

You need to design a solution that meets the refactoring requirements.

Which type of object should you include in the solution?

- A. An indexed view
- B. An aggregate function
- C. A distributed view
- D. A table-valued function

**Answer: D**

## **Topic 6, Coho Winery**

### **Overview**

You are a database developer for a company named Coho Winery. Coho Winery has an office in London.

Coho Winery has an application that is used to process purchase orders from customers and retailers in 10 different countries.

The application uses a web front end to process orders from the Internet. The web front end adds orders to a database named Sales. The Sales database is managed by a server named Server1.

An empty copy of the Sales database is created on a server named Server2 in the London office. The database will store sales data for customers in Europe.

A new version of the application is being developed. In the new version, orders will be placed either by using the existing web front end or by loading an XML file.

Once a week, you receive two files that contain the purchase orders and the order details of orders from offshore facilities.

You run the usp\_ImportOrders stored procedure and the usp\_ImportOrderDetails stored procedure to copy the offshore facility orders to the Sales database.

The Sales database contains a table named Orders that has more than 20 million rows.

### **Database Definitions**

#### **Database and Tables**

The following scripts are used to create the database and its tables:

```
01 CREATE DATABASE Sales;
02 GO
03 USE Sales;
04 GO
05 CREATE TABLE Products
06 (
07     ProductID int IDENTITY(1,1) NOT NULL,
08     Name nvarchar(100) NOT NULL,
09     UnitPrice decimal(18,2) NOT NULL,
10     Discontinued bit NOT NULL DEFAULT 0,
11     CONSTRAINT PK_Products PRIMARY KEY (ProductID)
12 );
13 GO
14
15 CREATE TABLE Customers
16 (
17     CustomerID int IDENTITY(1,1) NOT NULL,
18     Name nvarchar(200) NOT NULL,
19     Email nvarchar(200) NOT NULL,
20     Phone nvarchar(10) NOT NULL,
21     Address1 nvarchar(200) NOT NULL,
22     Address2 nvarchar(200) NULL,
23     City nvarchar(200) NOT NULL,
24     State char(2) NOT NULL,
25     ZIP char(5) NOT NULL,
26     CONSTRAINT PK_Customers PRIMARY KEY (CustomerID)
27 );
28 GO
29
30 CREATE TABLE Orders
31 (
32     OrderID int IDENTITY(1,1) NOT NULL,
33     CustomerID int NOT NULL,
34     OrderDate datetime NOT NULL DEFAULT GETDATE(),
35     DeliveryDate datetime NOT NULL,
36     ShipDate datetime NULL,
37     Amount decimal(18,2) NOT NULL,
38     CONSTRAINT PK_Orders PRIMARY KEY(OrderID)
39 );
40 GO
41
42 ALTER TABLE Orders
43     ADD CONSTRAINT FK_Orders_Customers
44     FOREIGN KEY(CustomerID)
45     REFERENCES Customers(CustomerID);
46 GO
47
48 CREATE TABLE OrderDetails
49 (
50     OrderID int NOT NULL,
51     LineItem int NOT NULL,
52     ProductID int NOT NULL,
53     Quantity int NOT NULL,
54     UnitPrice decimal(18,2) NOT NULL,
55     Total decimal(18,2) NOT NULL,
56     Discount decimal(18,2) NULL,
57     CONSTRAINT PK_OrderDetails PRIMARY KEY(OrderID, LineItem)
58 );
59 GO
60
61 ALTER TABLE OrderDetails
62     ADD CONSTRAINT FK_OrderDetails_Orders
63     FOREIGN KEY(OrderID)
64     REFERENCES Orders(OrderID);
65 GO
66
67 ALTER TABLE OrderDetails
68     ADD CONSTRAINT FK_OrderDetails_Products
69     FOREIGN KEY(ProductID)
70     REFERENCES Products(ProductID);
71 GO
```

## Stored Procedures

The following are the definitions of the stored procedures used in the database:

```
01 CREATE PROCEDURE usp_AddOrder
02     @customerID int,
03     @deliveryDate datetime,
04     @items ItemsTable READONLY,
05     @orderID int OUTPUT
06 AS
07 SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
08 BEGIN TRANSACTION;
09     DECLARE @amount decimal(18,2);
10     SELECT @amount = SUM(Quantity * UnitPrice) FROM @items;
11     INSERT INTO Orders (CustomerID, DeliveryDate, Amount)
12     VALUES (@customerID, @deliveryDate, @amount);
13     SELECT @orderID = @@IDENTITY;
14     INSERT INTO OrderDetails
15     SELECT @orderID, LineItem, ProductID, Quantity,
16     UnitPrice, Total, Discount
17     FROM @items;
18 COMMIT TRANSACTION;
19 GO

20 CREATE PROCEDURE usp_AddXMLOrder
21     @customerID int,
22     @deliverDate datetime,
23
24     @orderID int OUTPUT
25 AS
26 SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
27 BEGIN TRANSACTION;
28     DECLARE @itemsTable ItemsTable;
29     EXEC usp_ValidateAndGetItems @schema, @items, @itemsTable;
30     UPDATE Orders SET originalOrder = @items
31     WHERE OrderID = @orderID;
32 COMMIT TRANSACTION;
33 GO

34 CREATE PROCEDURE usp_ValidateAndGetItems
35     @schema sysname,
36     @items XML,
37     @itemsTable ItemsTable Output
38 AS

39 CREATE PROCEDURE usp_GetOrdersAndItems
40 AS
41
42     SELECT o.OrderID, o.OrderDate, o.DeliveryDate, o.Amount,
43     od.LineItem, od.Quantity, od.UnitPrice, p.Name
44     FROM Orders o
45     INNER JOIN OrderDetails od ON od.OrderID=o.OrderID
46     INNER JOIN Products p ON p.ProductID=od.ProductID
47     WHERE o.ShipDate IS NULL
48     AND o.DeliveryDate >= GETDATE() - 30;
49 GO
```

```
50 CREATE PROCEDURE usp_GetOrders
51 AS
52     SELECT OrderID, DeliveryDate, Amount
53     FROM Orders
54     WHERE ShipDate IS NULL
55     ORDER BY DeliveryDate;
56 GO
57
58 CREATE PROCEDURE usp_GetOrdersByProduct
59     @productID int
60
61 AS
62     SELECT OrderID, LineItem, Quantity,
63     UnitPrice, Total, Discount
64     FROM OrderDetails
65
66     WHERE ProductID = @productID;
67 GO
68
69 CREATE PROCEDURE usp_ImportOrders
70 AS
71     BULK INSERT Orders
72     FROM 'f:\orders\orders.tbl'
73     WITH
74     (
75         FIELDTERMINATOR = ' |',
76         ROWTERMINATOR = ' |\n'
77     );
78 GO
79
80 CREATE PROCEDURE usp_ImportOrderDetails
81     @firstRow int
82 AS
83     BULK INSERT OrderDetails
84     FROM 'f:\orders\details.tbl'
85     WITH
86     (
87         FIRSTROW = @firstRow,
88         FIELDTERMINATOR = ' |',
89         ROWTERMINATOR = ' |\n'
90     );
91 GO
```

## Indexes

The following indexes are part of the Sales database:

```
01 CREATE INDEX IX_Orders_ShipDate
02   ON Orders (Shipdate)
03
04   INCLUDE (CustomerID, OrderDate, Amount);
05 GO
```

## Data Import

The XML files will contain the list of items in each order. Each retailer will have its own XML schema and will be able to use different types of encoding. Each XML schema will use a default namespace. The default namespaces are not guaranteed to be unique.

For testing purposes, you receive an XSD file from a customer.





For testing purposes, you also create an XML schema collection named ValidateOrder. ValidateOrder contains schemas for all of the retailers.

The new version of the application must validate the XML file, parse the data, and store the parsed data along with the original XML file in the database. The original XML file must be stored without losing any data.

## Reported Issues

### Performance Issues

You notice the following for the usp\_GetOrdersAndItems stored procedure:

-  The stored procedure takes a long time to complete.
-  Less than two percent of the rows in the Orders table are retrieved by usp\_GetOrdersAndItems.
-  A full table scan runs when the stored procedure executes.
-  The amount of disk space used and the amount of time required to insert data are very high.

You notice that the usp\_GetOrdersByProduct stored procedure uses a table scan when the stored procedure is executed.

### Page Split Issues

Updates to the Orders table cause excessive page splits on the IX\_Orders\_ShipDate index.

## Requirements

### Site Requirements

Users located in North America must be able to view sales data for customers in North America and Europe in a single report. The solution must minimize the amount of traffic over the WAN link between the offices.

### **Bulk Insert Requirements**

The usp\_ImportOrderDetails stored procedure takes more than 10 minutes to complete. The stored procedure runs daily. If the stored procedure fails, you must ensure that the stored procedure restarts from the last successful set of rows.

### **Index Monitoring Requirements**

The usage of indexes in the Sales database must be monitored continuously. Monitored data must be maintained if a server restarts. The monitoring solution must minimize the usage of memory resources and processing resources.

### **Question No : 52 - (Topic 6)**

You need to modify usp\_GetOrdersAndItems to ensure that an order is NOT retrieved by usp\_GetOrdersAndItems while the order is being updated.

What should you add to usp\_GetOrdersAndItems?

- A. Add SET TRANSACTION ISOLATION LEVEL SERIALIZABLE to line 03.
- B. Add SET TRANSACTION ISOLATION LEVEL SNAPSHOT to line 03.
- C. Add (UPDLOCK) to the end of line 06.
- D. Add (READPAST) to the end of line 06.

**Answer: D**

### **Question No : 53 - (Topic 6)**

You need to implement a solution that addresses the index monitoring requirements.

What should you do?

- A. Schedule a SQL Server Agent job that saves data from the dynamic management views to a table in the database.
- B. Create a SQL Server Audit that saves data to a log file, and then create a SQL Server Audit Specification that gathers data from the DATABASE\_OPERATION group.
- C. Create a performance monitor Data Collector Set (DCS) that monitors the SQL Server counters.
- D. Schedule a SQL Server Profiler trace, and then save the trace data to a table in the database.



**Answer: A**

**Question No : 54 - (Topic 6)**

You need to implement a solution that solves the performance issues of usp\_GetOrdersAndItems.

Which statements should you execute?

- A.** CREATE INDEX IX\_Orders\_Active ON Orders(ShipDate, DeliveryDate, Amount)
- B.** CREATE INDEX IX\_Orders\_Active ON Orders(DeliveryDate) INCLUDE(Amount) WHERE ShipDate IS NULL
- C.** CREATE INDEX IX\_Orders\_Active ON Orders(DeliveryDate, Amount) WHERE ShipDate IS NULL
- D.** CREATE INDEX IX\_Orders\_Active ON Orders(ShipDate, DeliveryDate) INCLUDE( Amount)

**Answer: B**

**Question No : 55 - (Topic 6)**

You discover that the usp\_GetOrdersAndItems stored procedure takes a long time to complete while usp\_AddOrder or usp\_AddXMLOrder run.

You need to ensure that usp\_GetOrdersAndItems completes as quickly as possible.

What should you do? (Each correct answer presents part of the solution. Choose all that apply.)

- A.** Set the isolation level of the usp\_GetOrdersAndItems stored procedure to SERIALIZABLE.
- B.** Execute the ALTER DATABASE Sales SET ALLOW\_SNAPSHOT\_ISOLATION ON statement.
- C.** Set the isolation level of the usp\_AddOrder stored procedure to SERIALIZABLE.
- D.** Set the isolation level of the usp\_GetOrdersAndItems stored procedure to SNAPSHOT.
- E.** Set the isolation level of the usp\_AddOrder stored procedure to SNAPSHOT.

F. Execute the ALTER DATABASE Sales SET ALLOW\_SNAPSHOT\_ISOLATION OFF statement.

**Answer: B,D**

**Question No : 56 - (Topic 6)**

You need to implement a solution that addresses the performance issues of the usp\_GetOrdersByProduct stored procedure.

Which statement should you execute?

- A. 

```
CREATE INDEX IX_OrderDetails_ByProduct
ON OrderDetails (ProductID)
INCLUDE (OrderID, LineItem, UnitPrice, Total, Discount)
```
- B. 

```
CREATE INDEX IX_OrderDetails_ByProduct
ON OrderDetails (ProductID)
INCLUDE (LineItem, Quantity, UnitPrice, Total, Discount)
```
- C. 

```
CREATE INDEX IX_OrderDetails_ByProduct
ON OrderDetails (ProductID)
```
- D. 

```
CREATE INDEX IX_OrderDetails_ByProduct
ON OrderDetails (ProductID)
INCLUDE (LineItem, Quantity, UnitPrice, Discount)
```

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

**Answer: C**

**Question No : 57 - (Topic 6)**

You need to modify usp.GetOrdersAndItems to ensure that an order is NOT retrieved by usp\_GetOrdersAndItems while the order is being updated.

What should you add to usp.GetOrdersAndItems?

- A. Add WITH (NOLOCK) to the end of line 47.
- B. Add SET TRANSACTION ISOLATION LEVEL READ COMMITTED to line 44.
- C. Add SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED to line 44.
- D. Add WITH (READPAST) to the end of line 47.

**Answer: B**

### **Question No : 58 - (Topic 6)**

You need to implement a solution that addresses the page split issues.

Which statement should you execute?

- A. ALTER INDEX IX\_Orders\_ShipDate ON Orders REBUILD WITH (PAD\_INDEX=OFF, DROP\_EXISTING = ON);
- B. ALTER INDEX IX\_Orders\_ShipDate ON Orders REBUILD WITH (FILLFACTOR=50, DROP\_EXISTING = ON);
- C. ALTER INDEX IX\_Orders\_ShipDate ON Orders REBUILD WITH (FILLFACTOR = 0, DROP\_EXISTING = ON);
- D. ALTER INDEX IX\_Orders\_ShipDate ON Orders REBUILD WITH (PAD\_INDEX=ON, DROP\_EXISTING = ON);

**Answer: B**

### **Question No : 59 - (Topic 6)**

You need to modify the Orders table to store the XML data used by the retailers.

Which statement should you execute?

- A. ALTER Orders ADD originalOrder XML (ValidateOrder);
- B. ALTER Orders

ADD originalOrder XML;

C. ALTER Orders

ADD originalOrder varchar(max);

D. ALTER Orders

ADD originalOrder varbinary(max);

**Answer: D**

### Question No : 60 - (Topic 6)

You plan to create a stored procedure that inserts data from an XML file to the OrderDetails table. The following is the signature of the stored procedure:

```
CREATE PROCEDURE usp_InsertItems
    @items XML (ValidateOrder)
```

The following is the XSD file used to create the ValidateOrder schema collection:

```
<?xml version="1.0" encoding="UTF-16"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
<xsd:element name="root">
  <xsd:complexType mixed="true">
    <xsd:sequence>
      <xsd:element name="Product"
        minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType mixed="true">
          <xsd:sequence>
            <xsd:element name="UnitPrice" type="xsd:decimal"
              minOccurs="1" maxOccurs="1" />
            <xsd:element name="Quantity" type="xsd:integer"
              minOccurs="1" maxOccurs="1" />
          </xsd:sequence>
          <xsd:attribute name="lineItem"
            type="xsd:integer" use="required"/>
          <xsd:attribute name="productID"
            type="xsd:integer" use="required"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="numberItems"
      type="xsd:integer" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```

You develop a code segment that retrieves the number of items and loops through each item. Each time the loop runs, a variable named @itemNumber is incremented.

You need to develop a code segment that retrieves the product ID of each item number in the loop.

Which code segment should you develop?

- A. SET @productID = @items.value'/Root/Product/productID', int)
- B. SET @productID = @items.value'/Root/Product['+ @itemNumber+ ']/@productID', int)
- C. SET @productID = @items.value'/Root/Product['+ @itemNumber+ ']/productID', int)
- D. SET @productID = @items.value'/Root/Product/@productID', int)

**Answer: B**

#### **Question No : 61 - (Topic 6)**

You need to ensure that a new execution plan is used by usp\_GetOrdersByProduct each time the stored procedure runs.

What should you do?

- A. Execute sp\_help usp\_GetOrdersByProduct\
- B. Add WITH (FORCESEEK) to line 69 in usp.GetOrdersByProduct.
- C. Add WITH RECOMPILE to line 64 in usp.GetOrdersByProduct.
- D. Execute sp\_recompile usp.GetOrdersByProduct'.

**Answer: B**

#### **Question No : 62 - (Topic 6)**

You need to ensure that a new execution plan is used by usp\_GetOrdersByProduct each time the stored procedure runs.

What should you do?

- A. Execute sp\_help 'usp\_GetOrdersByProduct'.

- B. Execute sp\_recompile 'usp\_GetOrdersByProduct'.
- C. Add WITH RECOMPILE to line 03 in usp\_GetOrdersByProduct.
- D. Add WITH (FORCESEEK) to line 07 in usp\_GetOrdersByProduct.

**Answer: C**

**Explanation:**

Ref: [http://msdn.microsoft.com/en-us/library/ms190439\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms190439(v=sql.90).aspx)

**Question No : 63 - (Topic 6)**

You need to ensure that usp\_AddXMLOrder can be used to validate the XML input from the retailers.

Which parameters should you add to usp\_AddXMLOrder on line 04 and line 05? (Each correct answer presents part of the solution. Choose all that apply.)

- A. @schema varbinary(100).
- B. @items varchar(max).
- C. @schema sysname.
- D. @items varbinary(max).
- E. @items xml.
- F. @schema xml.

**Answer: C,E**

**Question No : 64 - (Topic 6)**

You need to implement a solution that meets the site requirements.

What should you implement?

- A. A non-indexed view on Server1
- B. A non-indexed view on Server2
- C. A distributed view on Server1
- D. A distributed view on Server2

**Answer: C**

### Question No : 65 - (Topic 6)

You need to implement a solution that addresses the bulk insert requirements.

What should you add to line 08 in usp\_ImportOrderDetails?

- A. LASTROW=0.
- B. BATCHSIZE=0.
- C. BATCHSIZE=1000.
- D. LASTROW = 1000.

**Answer: C**

### Topic 7, Fourth Coffee Background

#### Corporate Information

Fourth Coffee is global restaurant chain. There are more than 5,000 locations worldwide.




#### Physical Locations

Currently a server at each location hosts a SQL Server 2012 instance. Each instance contains a database called StoreTransactions that stores all transactions from point of sale and uploads summary batches nightly.

Each server belongs to the COFFECORP domain. Local computer accounts access the StoreTransactions database at each store using sysadmin and datareaderwriter roles.

#### Planned changes

Fourth Coffee has three major initiatives:

-  The IT department must consolidate the point of sales database infrastructure.
-  The marketing department plans to launch a mobile application for micropayments.
-  The finance department wants to deploy an internal tool that will help detect fraud.

Initially, the mobile application will allow customers to make micropayments to buy coffee and other items on the company web site. These micropayments may be sent as gifts to other users and redeemed within an hour of ownership transfer. Later versions will generate profiles based on customer activity that will push texts and ads generated by an analytics application.

When the consolidation is finished and the mobile application is in production, the micropayments and point of sale transactions will use the same database.

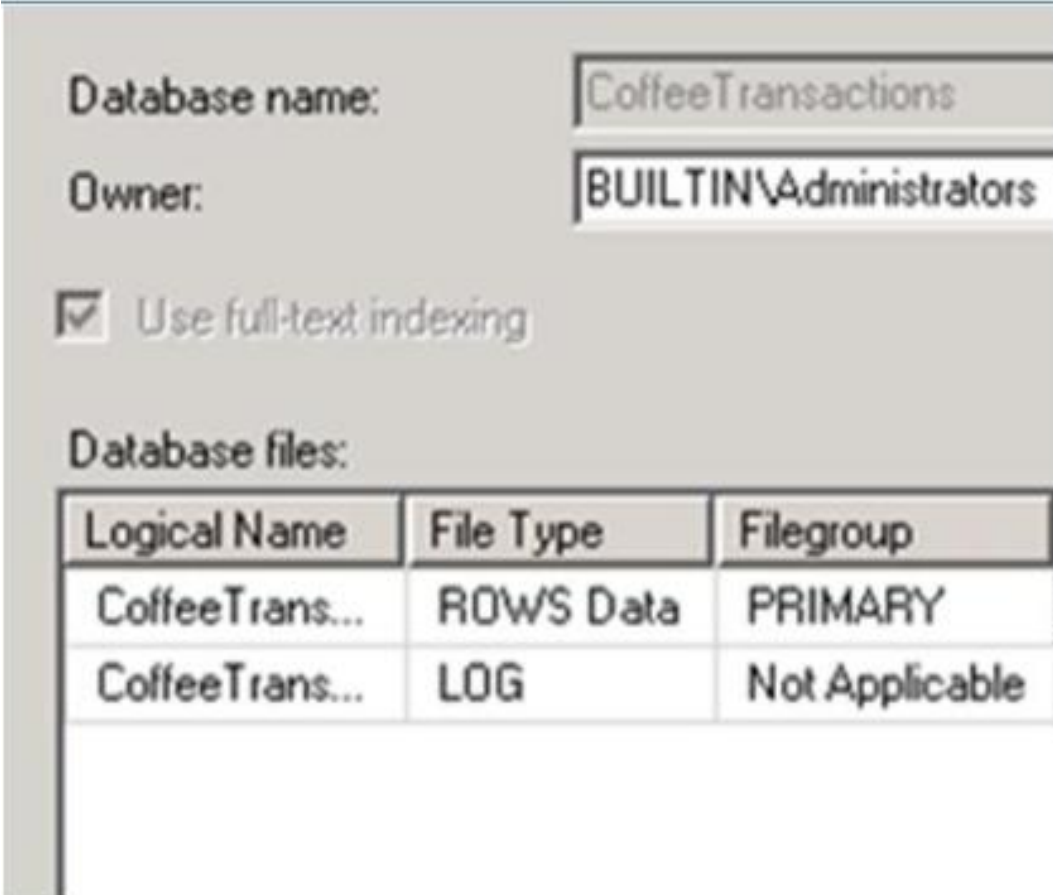
## Existing Environment

### Existing Application Environment

Some stores have been using several pilot versions of the micropayment application. Each version currently is in a database that is independent from the point of sales systems. Some versions have been used in field tests at local stores, and others are hosted at corporate servers. All pilot versions were developed by using SQL Server 2012.

### Existing Support Infrastructure

The proposed database for consolidating micropayments and transactions is called CoffeeTransactions. The database is hosted on a SQL Server 2014 Enterprise Edition instance and has the following file structures:



Database name: CoffeeTransactions

Owner: BUILTIN\Administrators

Use full-text indexing

Database files:

Logical Name	File Type	Filegroup
CoffeeTrans...	ROWS Data	PRIMARY
CoffeeTrans...	LOG	Not Applicable

## Business Requirements

### General Application Solution Requirements

The database infrastructure must support a phased global rollout of the micropayment application and consolidation.

The consolidated micropayment and point of sales database will be into a CoffeeTransactions database. The infrastructure also will include a new CoffeeAnalytics



database for reporting on content from CoffeeTransactions.

Mobile applications will interact most frequently with the micropayment database for the following activities:

- ✍ Retrieving the current status of a micropayment;
- ✍ Modifying the status of the current micropayment; and
- ✍ Canceling the micropayment.

The mobile application will need to meet the following requirements:

- ✍ Communicate with web services that assign a new user to a micropayment by using a stored procedure named `usp_AssignUser`.
- ✍ Update the location of the user by using a stored procedure named `usp_AddMobileLocation`.

The fraud detection service will need to meet the following requirements:

- ✍ Query the current open micropayments for users who own multiple micropayments by using a stored procedure named `usp.LookupConcurrentUsers`.
- ✍ Persist the current user locations by using a stored procedure named `usp_MobileLocationSnapshot`.
- ✍ Look at the status of micropayments and mark micropayments for internal investigations.
- ✍ Move micropayments to `dbo.POSException` table by using a stored procedure named `usp_DetectSuspiciousActivity`.
- ✍ Detect micropayments that are flagged with a `StatusId` value that is greater than 3 and that occurred within the last minute.

The CoffeeAnalytics database will combine imports of the `POSTransaction` and `MobileLocation` tables to create a `UserActivity` table for reports on the trends in activity. Queries against the `UserActivity` table will include aggregated calculations on all columns that are not used in filters or groupings.

Micropayments need to be updated and queried for only a week after their creation by the mobile application or fraud detection services.

## Performance

The most critical performance requirement is keeping the response time for any queries of the `POSTransaction` table predictable and fast.

Web service queries will take a higher priority in performance tuning decisions over the fraud detection agent queries.

## Scalability

Queries of the user of a micropayment cannot return while the micropayment is being updated, but can show different users during different stages of the transaction.

The fraud detection service frequently will run queries over the micropayments that occur over different time periods that range between 30 seconds and ten minutes.

The `POSTransaction` table must have its structure optimized for hundreds of thousands of active micropayments that are updated frequently.

All changes to the POSTransaction table will require testing in order to confirm the expected throughput that will support the first year's performance requirements.

Updates of a user's location can tolerate some data loss.

Initial testing has determined that the POSTransaction and POSException tables will be migrated to an in-memory optimized table.

### **Availability**

In order to minimize disruption at local stores during consolidation, nightly processes will restore the databases to a staging server at corporate headquarters.

### **Technical Requirements**

#### **Security**

The sensitive nature of financial transactions in the store databases requires certification of the COFFECORP\Auditors group at corporate that will perform audits of the data. Members of the COFFECORP\Auditors group cannot have sysadmin or datawriter access to the database.

Compliance requires that the data stewards have access to any restored StoreTransactions database without changing any security settings at a database level.

Nightly batch processes are run by the services account in the COFFECORP\StoreAgent group and need to be able to restore and verify the schema of the store databases match.

No Windows group should have more access to store databases than is necessary.

#### **Maintainability**

You need to anticipate when POSTransaction table will need index maintenance.

When the daily maintenance finishes, micropayments that are one week old must be available for queries in UserActivity table but will be queried most frequently within their first week and will require support for in-memory queries for data within first week.

The maintenance of the UserActivity table must allow frequent maintenance on the day's most recent activities with minimal impact on the use of disk space and the resources available to queries. The processes that add data to the UserActivity table must be able to update data from any time period, even while maintenance is running.

The index maintenance strategy for the UserActivity table must provide the optimal structure for both maintainability and query performance.

All micropayments queries must include the most permissive isolation level available for the maximum throughput.

In the event of unexpected results, all stored procedures must provide error messages in text message to the calling web service.

Any modifications to stored procedures will require the minimal amount of schema changes necessary to increase the performance.

### **Performance**

Stress testing of the mobile application on the proposed CoffeeTransactions database uncovered performance bottlenecks. The sys.dm\_os\_wait\_stats Dynamic Management View (DMV) shows high wait\_time values for WRTTELOG and PAGEIOLATCHJJP wait types when updating the MobileLocation table.

Updates to the MobileLocation table must have minimal impact on physical resources.

### **Supporting Infrastructure**

The stored procedure usp\_LookupConcurrentUsers has the current implementation:

```

CREATE PROCEDURE usp_LookupConcurrentUsers
AS BEGIN
    --summary table
    CREATE TABLE #POSTransactionTemp (
        POSTransactionId int NOT NULL,
        UserId int NOT NULL,
        StatusID int NOT NULL,
        POSLocation int NOT NULL,
        CreateDate datetime2 NOT NULL,
        Price money
    )
    DECLARE @timewindow datetime2
    SET @timewindow = GETDATE();

    WITH concurrentusers
    AS
    (
        SELECT UserId, COUNT(*) concurrentsessions
        FROM dbo.POSTransaction
        WHERE CreateDate >= dateadd(second,-60, @timewindow )
        GROUP BY UserId
        HAVING COUNT(*) > 1
    )
    INSERT INTO #POSTransactionTemp
    (
        POSTransactionId, UserId,
        StatusID, POSLocation,
        CreateDate, Price
    )

    SELECT d.*
    FROM dbo.POSTransaction d
    JOIN concurrentusers c
    on d.UserID = c.UserId
    WHERE d.CreateDate >= dateadd(second,-60, @timewindow )
    ...
    SELECT * FROM #POSTransactionTemp

END

```

The current stored procedure for persisting a user location is defined in the following code:

```
CREATE PROCEDURE dbo.usp_MobileLocationSnapshot
AS
BEGIN

INSERT INTO CoffeeAnalytics.dbo.MobileLocationLog
SELECT * FROM CoffeeTransactions.dbo.MobileLocation

END
```

The current stored procedure for managing micropayments needing investigation is defined in the following code:

```
01 CREATE PROCEDURE dbo.usp_DetectSuspiciousActivity
02 WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER
03 AS
04 BEGIN ATOMIC
05 WITH (TRANSACTION ISOLATION LEVEL = SNAPSHOT,
06 LANGUAGE = 'us_english')
07 IF EXISTS(SELECT POSTransactionId FROM dbo.POSTransaction
08 WHERE StatusID >= 4 and CreateDate >= dateadd(second,-60,
09 GETDATE() ))
10 MERGE dbo.POSException AS target
11 USING (SELECT POSTransactionId, StatusID, UserId,
12 POSLocation, CreateDate, Price FROM dbo.POSTransaction
13 WHERE StatusID >= 4 and
14 CreateDate >= dateadd(second,-60, GETDATE() ))
15 AS source (POSTransactionId, StatusID, UserId,
16 POSLocation, CreateDate, Price)
17 ON (target.POSTransactionId = source.POSTransactionId)
18 WHEN MATCHED THEN
19 UPDATE SET StatusID = source.StatusID
20 WHEN NOT MATCHED THEN
21 INSERT (POSTransactionId, StatusID, UserId,
22 POSLocation, CreateDate, Price)
23 VALUES (source.POSTransactionId, source.StatusID,
24 source.UserId, source.POSLocation,
25 source.CreateDate, source.Price);
26 END
```

The current table, before implementing any performance enhancements, is defined as follows:

```
CREATE TABLE dbo.POSTransaction (  
    POSTransactionId int NOT NULL PRIMARY KEY,  
    UserId int NOT NULL,  
    POSLocation int NOT NULL,  
    StatusID int NOT NULL,  
    CreateDate datetime2 NOT NULL,  
    Price money  
)  
CREATE INDEX ix_UserID on dbo.POSTransaction(UserId)
```

**Question No : 66 DRAG DROP - (Topic 7)**

You need to redesign the system to meet the scalability requirements of the application.

Develop the solution by selecting and arranging the required code blocks in the correct order.

You may not need all of the code blocks.

Code Blocks	Answer Area
<pre>    ,     UserId int NOT NULL     INDEX ix_UserId NONCLUSTERED     HASH WITH (BUCKET_COUNT=2),</pre>	
<pre>    ,     UserId int NOT NULL     INDEX x_UserId NONCLUSTERED     HASH WITH (BUCKET_COUNT=900000),</pre>	
<pre>    POSLocation int NOT NULL,     StatusID int NOT NULL,     CreateDate datetime2 NOT NULL,     Price money     )</pre>	
<pre>    POSTransactionId int NOT NULL     PRIMARY KEY CLUSTERED</pre>	
<pre>    POSTransactionId int NOT NULL</pre>	
<pre>ALTER DATABASE CoffeeTransactions ADD FILEGROUP [CoffeeTransactions_inmem ] CONTAINS MEMORY_OPTIMIZED_DATA</pre>	
<pre>ON [CoffeeTransactions_inmem]</pre>	
<pre>WITH (MEMORY_OPTIMIZED=ON, DURABILITY=SCHEMA_ONLY)</pre>	
<pre>    POSTransactionId int NOT NULL     PRIMARY KEY CLUSTERED     HASH WITH (BUCKET_COUNT=1000000)</pre>	
<pre>    ,     UserId int NOT NULL     NONCLUSTERED INDEX ix_UserId,</pre>	
<pre>CREATE TABLE dbo.POSTransaction (</pre>	
<pre>    POSTransactionId int NOT NULL     PRIMARY KEY NONCLUSTERED     HASH WITH (BUCKET_COUNT=1)</pre>	

Answer:

Code Blocks	Answer Area
<pre> , UserId int NOT NULL INDEX ix_UserId NONCLUSTERED HASH WITH (BUCKET_COUNT=2), </pre>	<pre> ALTER DATABASE CoffeeTransactions ADD FILEGROUP [CoffeeTransactions_inmem ] CONTAINS MEMORY_OPTIMIZED_DATA </pre>
<pre> , UserId int NOT NULL INDEX x_UserId NONCLUSTERED HASH WITH (BUCKET_COUNT=900000), </pre>	<pre> CREATE TABLE dbo.POSTransaction ( </pre>
<pre> POSLocation int NOT NULL, StatusID int NOT NULL, CreateDate datetime2 NOT NULL, Price money ) </pre>	<pre> , UserId int NOT NULL INDEX x_UserId NONCLUSTERED HASH WITH (BUCKET_COUNT=900000), </pre>
<pre> POSTransactionId int NOT NULL PRIMARY KEY CLUSTERED </pre>	<pre> POSTransactionId int NOT NULL PRIMARY KEY CLUSTERED HASH WITH (BUCKET_COUNT=1000000) </pre>
<pre> POSTransactionId int NOT NULL </pre>	<pre> ) POSLocation int NOT NULL, StatusID int NOT NULL, CreateDate datetime2 NOT NULL, Price money ) </pre>
<pre> ALTER DATABASE CoffeeTransactions ADD FILEGROUP [CoffeeTransactions_inmem ] CONTAINS MEMORY_OPTIMIZED_DATA </pre>	<pre> WITH (MEMORY_OPTIMIZED=ON, DURABILITY=SCHEMA_ONLY) </pre>
<pre> ON [CoffeeTransactions_inmem] </pre>	<pre> ON [CoffeeTransactions_inmem] </pre>
<pre> WITH (MEMORY_OPTIMIZED=ON, DURABILITY=SCHEMA_ONLY) </pre>	
<pre> POSTransactionId int NOT NULL PRIMARY KEY CLUSTERED HASH WITH (BUCKET_COUNT=1000000) </pre>	
<pre> , UserId int NOT NULL NONCLUSTERED INDEX ix_UserId, </pre>	
<pre> CREATE TABLE dbo.POSTransaction ( </pre>	
<pre> POSTransactionId int NOT NULL PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT=1) </pre>	

Explanation: Box 1:

```

ALTER DATABASE CoffeeTransactions
ADD FILEGROUP [CoffeeTransactions_inmem
] CONTAINS MEMORY_OPTIMIZED_DATA

```

Box 2:

```

CREATE TABLE dbo.POSTransaction (

```

Box 3:



```
UserId int NOT NULL  
INDEX x_UserId NONCLUSTERED  
HASH WITH (BUCKET_COUNT=900000),
```

Box 4:

```
POSTransactionId int NOT NULL  
PRIMARY KEY CLUSTERED  
HASH WITH (BUCKET_COUNT=1000000)
```

Box 5:

```
POSLocation int NOT NULL,  
StatusID int NOT NULL,  
CreateDate datetime2 NOT NULL,  
Price money  
)
```

Box 6:

```
WITH (MEMORY_OPTIMIZED=ON,  
DURABILITY=SCHEMA_ONLY)
```

Box 7:

```
ON [CoffeeTransactions_inmem]
```

Note:

\* MEMORY\_OPTIMIZED\_DATA

First create a memory-optimized data filegroup and add a container to the filegroup.  
Then create a memory-optimized table.

\* 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.

\* Example:

```
-- create a durable (data will be persisted) memory-optimized table  
-- two of the columns are indexed
```

```
CREATE TABLE dbo.ShoppingCart (  
ShoppingCartId INT IDENTITY(1,1) PRIMARY KEY NONCLUSTERED,  
UserId INT NOT NULL INDEX ix_UserId NONCLUSTERED HASH WITH  
(BUCKET_COUNT=1000000),  
CreateDate DATETIME2 NOT NULL,  
TotalPrice MONEY  
) WITH (MEMORY_OPTIMIZED=ON)  
GO
```

### Question No : 67 - (Topic 7)

You need to optimize the index structure that is used by the tables that support the fraud detection services.

What should you do?

- A. Add a hashed nonclustered index to CreateDate.
- B. Add a not hash nonclustered index to CreateDate.
- C. Add a not hash clustered index on POSTransactionId and CreateDate.
- D. Add a hashed clustered index on POSTransactionId and CreateDate.

**Answer: A**

**Explanation:** The fraud detection service will need to meet the following requirement (among others):

\* Detect micropayments that are flagged with a StatusId value that is greater than 3 and that occurred within the last minute.

### Question No : 68 DRAG DROP - (Topic 7)

You need to create the usp.AssignUser stored procedure.

Develop the solution by selecting and arranging the required code blocks in the correct order. You may not need all of the code blocks.

Code Blocks	Answer Area
<pre>IF @StatusID IS NULL   RAISERROR (N'The transaction does not exist.',16,1)</pre>	
<pre>WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER</pre>	
<pre>CREATE PROCEDURE dbo.usp_AssignUser @UserId int, @POSTransactionId int</pre>	
<pre>WITH (TRANSACTION ISOLATION LEVEL = READ COMMITTED, LANGUAGE = N'us_english')</pre>	
<pre>UPDATE dbo.POSTransaction SET UserId=@UserId WHERE POSTransactionId=@POSTransactio nId END</pre>	
<pre>AS BEGIN</pre>	
<pre>DECLARE @StatusID int SELECT @StatusID=StatusId FROM dbo.POSTransaction WHERE POSTransactionId=@POSTransactionI d</pre>	
<pre>IF @StatusID IS NULL   THROW 51000, N'The transaction does not exist.', 1</pre>	
<pre>WITH (TRANSACTION ISOLATION LEVEL = REPEATABLE READ, LANGUAGE = N'us_english')</pre>	
<pre>AS BEGIN ATOMIC</pre>	

Answer:

Code Blocks	Answer Area
<pre>IF @StatusID IS NULL RAISERROR (N'The transaction does not exist.',16,1)</pre>	<pre>CREATE PROCEDURE dbo.usp_AssignUser @UserId int, @POSTransactionId int</pre>
<pre>WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER</pre>	<pre>WITH NATIVE_COMPILATION, SCHEMABINDING, EXECUTE AS OWNER</pre>
<pre>CREATE PROCEDURE dbo.usp_AssignUser @UserId int, @POSTransactionId int</pre>	<pre>AS BEGIN ATOMIC</pre>
<pre>WITH (TRANSACTION ISOLATION LEVEL = READ COMMITTED, LANGUAGE = N'us_english')</pre>	<pre>WITH (TRANSACTION ISOLATION LEVEL = REPEATABLE READ, LANGUAGE = N'us_english')</pre>
<pre>UPDATE dbo.POSTransaction SET UserId=@UserId WHERE POSTransactionId=@POSTransactio nId END</pre>	<pre>UPDATE dbo.POSTransaction SET UserId=@UserId WHERE POSTransactionId=@POSTransactio nId END</pre>
<pre>AS BEGIN</pre>	
<pre>DECLARE @StatusID int SELECT @StatusID=StatusId FROM dbo.POSTransaction WHERE POSTransactionId=@POSTransactionI d</pre>	<pre>DECLARE @StatusID int SELECT @StatusID=StatusId FROM dbo.POSTransaction WHERE POSTransactionId=@POSTransactionI d</pre>
<pre>IF @StatusID IS NULL THROW 51000, N'The transaction does not exist.', 1</pre>	<pre>IF @StatusID IS NULL THROW 51000, N'The transaction does not exist.', 1</pre>
<pre>WITH (TRANSACTION ISOLATION LEVEL = REPEATABLE READ, LANGUAGE = N'us_english')</pre>	
<pre>AS BEGIN ATOMIC</pre>	

Explanation: Box 1:

```
CREATE PROCEDURE dbo.usp_AssignUser
@UserId int, @POSTransactionId int
```

Box 2:

```
WITH
NATIVE_COMPILATION, SCHEMABINDING,
EXECUTE AS OWNER
```

Box 3:

```
AS
BEGIN ATOMIC
```

Box 4:

```
WITH (TRANSACTION ISOLATION LEVEL =  
REPEATABLE READ, LANGUAGE  
= N'us_english')
```

Box 5:

```
UPDATE dbo.POSTransaction  
SET UserId=@UserId  
WHERE POSTransactionId=@POSTransactio  
nId  
END
```

Box 6:

```
DECLARE @StatusID int  
SELECT @StatusID=StatusId  
FROM dbo.POSTransaction  
WHERE POSTransactionId=@POSTransactionI  
d
```

Box 7:

```
IF @StatusID IS NULL  
THROW 51000, N'The transaction  
does not exist.', 1
```

Note:

\* From scenario: The mobile application will need to meet the following requirements:  
/Communicate with web services that assign a new user to a micropayment by using a  
stored procedure named usp\_AssignUser.

\* Example:

```
create procedure dbo.OrderInsert(@OrdNo integer, @CustCode nvarchar(5))
```

```
with native_compilation, schemabinding, execute as owner
```

```
as
```

```
begin atomic with
```

```
(transaction isolation level = snapshot,
```

```
language = N'English')
```

```
declare @OrdDate datetime = getdate();
```

```
insert into dbo.Ord (OrdNo, CustCode, OrdDate) values (@OrdNo, @CustCode,
```

```
@OrdDate);  
end  
go
```

\* Natively compiled stored procedures are Transact-SQL stored procedures compiled to native code that access memory-optimized tables. Natively compiled stored procedures allow for efficient execution of the queries and business logic in the stored procedure.

\* READ COMMITTED versus REPEATABLE READ

Read committed is an isolation level that guarantees that any data read was committed at the moment it is read. It simply restricts the reader from seeing any intermediate, uncommitted, 'dirty' read. It makes no promise whatsoever that if the transaction re-issues the read, it will find the same data, data is free to change after it was read.

Repeatable read is a higher isolation level, that in addition to the guarantees of the read committed level, it also guarantees that any data read cannot change, if the transaction reads the same data again, it will find the previously read data in place, unchanged, and available to read.

\* Both RAISERROR and THROW statements are used to raise an error in SQL Server. The journey of RAISERROR started from SQL Server 7.0, whereas the journey of THROW statement has just begun with SQL Server 2012. Obviously, Microsoft is suggesting us to start using the THROW statement instead of RAISERROR. The THROW statement seems to be simple and easy to use than RAISERROR.

\* Explicit transactions. The user starts the transaction through an explicit BEGIN TRAN or BEGIN ATOMIC. The transaction is completed following the corresponding COMMIT and ROLLBACK or END (in the case of an atomic block).

### **Question No : 69 DRAG DROP - (Topic 7)**

You need to design the UserActivity table.

Which three steps should you perform in sequence? To answer, move the appropriate three actions from the list of actions to the answer area and arrange them in the correct order.

Actions	Answer Area
Create a nonclustered hash index.	
Create a clustered columnstore index.	
Create a partitioning scheme for use by the table.	
Use an ALTER INDEX REBUILD on a specific partition.	
Use an ALTER INDEX REORGANIZE on a specific partition.	

**Answer:**

Actions	Answer Area
Create a nonclustered hash index.	Create a clustered columnstore index.
Create a clustered columnstore index.	
Create a partitioning scheme for use by the table.	Create a partitioning scheme for use by the table.
Use an ALTER INDEX REBUILD on a specific partition.	
Use an ALTER INDEX REORGANIZE on a specific partition.	Use an ALTER INDEX REORGANIZE on a specific partition.

**Explanation:** Box 1:

Create a clustered columnstore index.

Box 2:

Create a partitioning scheme for use by the table.

Box 3:

Use an ALTER INDEX REORGANIZE on a specific partition.

Note:

\* Creating a partitioned table or index typically happens in four parts:

- ✍ Create a filegroup or filegroups and corresponding files that will hold the partitions specified by the partition scheme.
- ✍ Create a partition function that maps the rows of a table or index into partitions based on the values of a specified column.
- ✍ Create a partition scheme that maps the partitions of a partitioned table or index to the new filegroups.
- ✍ Create or modify a table or index and specify the partition scheme as the storage location.

\* Reorganizing an index uses minimal system resources.

\* From scenario:

/ The index maintenance strategy for the UserActivity table must provide the optimal structure for both maintainability and query performance.

/ The CoffeeAnalytics database will combine imports of the POSTransaction and MobileLocation tables to create a UserActivity table for reports on the trends in activity. Queries against the UserActivity table will include aggregated calculations on all columns that are not used in filters or groupings.

/ When the daily maintenance finishes, micropayments that are one week old must be available for queries in UserActivity table but will be queried most frequently within their first week and will require support for in-memory queries for data within first week.

The maintenance of the UserActivity table must allow frequent maintenance on the day's most recent activities with minimal impact on the use of disk space and the resources available to queries. The processes that add data to the UserActivity table must be able to update data from any time period, even while maintenance is running.

\* Columnstore indexes work well for mostly read-only queries that perform analysis on large data sets. Often, these are queries for data warehousing workloads. Columnstore indexes give high performance gains for queries that use full table scans, and are not well-suited for queries that seek into the data, searching for a particular value.

### **Question No : 70 - (Topic 7)**

You need to monitor the health of your tables and indexes in order to implement the required index maintenance strategy.

What should you do?



- A. Query system DMVs to monitor avg\_chain\_length and max\_chain\_length. Create alerts to notify you when these values converge.
- B. Create a SQL Agent alert when the File Table: Avg time per file I/O request value is increasing.
- C. Query system DMVs to monitor total\_bucket\_count. Create alerts to notify you when this value increases.
- D. Query system DMVs to monitor total\_bucket\_count. Create alerts to notify you when this value decreases.

**Answer: A**

**Explanation:** From scenario:

- \* You need to anticipate when POSTransaction table will need index maintenance.
- \* The index maintenance strategy for the UserActivity table must provide the optimal structure for both maintainability and query performance.

### Question No : 71 - (Topic 7)

You need to modify the stored procedure usp\_LookupConcurrentUsers.

What should you do?

- A. Add a clustered index to the summary table.
- B. Add a nonclustered index to the summary table.
- C. Add a clustered columnstore index to the summary table.
- D. Use a table variable instead of the summary table.

**Answer: A**

**Explanation:** Scenario: Query the current open micropayments for users who own multiple micropayments by using a stored procedure named usp.LookupConcurrentUsers

### Question No : 72 DRAG DROP - (Topic 7)

You need to implement a new version of usp\_AddMobileLocation. Develop the solution by selecting and arranging the required code blocks in the correct order. You may not need all

of the code blocks.

Code Blocks	Answer Area
<pre>DELAYED_DURABILITY = ON ,TRANSACTION ISOLATION LEVEL = SNAPSHOT</pre>	
<pre>CREATE PROCEDURE dbo.usp_AddMobileLocation @POSTransactionId int, @Longitude float, @Latitude float, @Lat float WITH</pre>	
<pre>NATIVE_COMPILATION ...</pre>	
<pre>DELAYED_DURABILITY = OFF ,TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED</pre>	
<pre>DELAYED_DURABILITY = ON ,TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED</pre>	
<pre>Insert into dbo.MobileLocation ( POSTransactionId, Longitude, Latitude, CreateDate ) VALUES ( @POSTransactionId, @Longitude, @Latitude, GETDATE() ) END</pre>	
<pre>,LANGUAGE = N'English' )</pre>	
<pre>AS BEGIN ATOMIC WITH (</pre>	
<pre>DELAYED_DURABILITY = OFF ,TRANSACTION ISOLATION LEVEL = SNAPSHOT</pre>	

Answer:

Code Blocks	Answer Area
<pre>DELAYED_DURABILITY = ON ,TRANSACTION ISOLATION LEVEL = SNAPSHOT</pre>	<pre>CREATE PROCEDURE dbo.usp_AddMobileLocat ion @POSTransactionId int, @Long float, @Lat float WITH</pre>
<pre>CREATE PROCEDURE dbo.usp_AddMobileLocat ion @POSTransactionId int, @Long float, @Lat float WITH</pre>	<pre>NATIVE_COMPILATION ...</pre>
<pre>NATIVE_COMPILATION ...</pre>	<pre>AS BEGIN ATOMIC WITH (</pre>
<pre>DELAYED_DURABILITY = OFF ,TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED</pre>	<pre>DELAYED_DURABILITY = OFF ,TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED</pre>
<pre>DELAYED_DURABILITY = ON ,TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED</pre>	<pre>,LANGUAGE = N'English' )</pre>
<pre>Insert into dbo.MobileLocation ( POSTransactionId, Longitude, Latitude, CreateDate ) VALUES ( @POSTransactionId, @Long, @Lat, GETDATE() ) END</pre>	<pre>Insert into dbo.MobileLocation ( POSTransactionId, Longitude, Latitude, CreateDate ) VALUES ( @POSTransactionId, @Long, @Lat, GETDATE() ) END</pre>
<pre>,LANGUAGE = N'English' )</pre>	
<pre>AS BEGIN ATOMIC WITH (</pre>	
<pre>DELAYED_DURABILITY = OFF ,TRANSACTION ISOLATION LEVEL = SNAPSHOT</pre>	

Explanation: Box 1:

```
CREATE PROCEDURE dbo.usp_AddMobileLocat
ion @POSTransactionId int, @Long
float, @Lat float
WITH
```

Box 2:

```
NATIVE_COMPILATION
...
```

Box 3:

```
AS  
BEGIN ATOMIC WITH  
(
```

Box 4:

```
DELAYED_DURABILITY = OFF  
, TRANSACTION ISOLATION LEVEL  
= READ UNCOMMITTED
```

Box 5:

```
, LANGUAGE = N'English'  
)
```

Box 6:

```
Insert into dbo.MobileLocation  
(  
POSTransactionId,  
Longitude,  
Latitude,  
CreateDate  
)  
VALUES  
(  
@POSTransactionId,  
@Long,  
@Lat,  
GETDATE()  
)  
END
```

Note:

\* From scenario:

The mobile application will need to meet the following requirements:

- Update the location of the user by using a stored procedure named usp\_AddMobileLocation.

\* DELAYED\_DURABILITY

SQL Server transaction commits can be either fully durable, the SQL Server default, or

delayed durable (also known as lazy commit).

Fully durable transaction commits are synchronous and report a commit as successful and return control to the client only after the log records for the transaction are written to disk.

Delayed durable transaction commits are asynchronous and report a commit as successful before the log records for the transaction are written to disk. Writing the transaction log entries to disk is required for a transaction to be durable. Delayed durable transactions become durable when the transaction log entries are flushed to disk.

### Question No : 73 DRAG DROP - (Topic 7)

You need to optimize the index and table structures for POSTransaction.

Which task should you use with each maintenance step? To answer, drag the appropriate tasks to the correct maintenance steps. Each task 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.

Tasks	Maintenance Steps
an identity for UserActivityID starting at the next value	Convert UserActivity to use <input type="text" value="Task"/>
a sequence for UserActivityID starting at the next value	
on-disk tables using the partitioning scheme	Copy UserActivity metadata to create UserActivity_Archive as <input type="text" value="Task"/>
in-memory tables using the partitioning scheme	After copying UserActivity metadata to create UserActivity_Staging, create a view on top of <input type="text" value="Task"/>
UserActivity and UserActivity_Archive	After switching a new partition from UserActivity_Staging into UserActivity_Archive, <input type="text" value="Task"/>
UserActivity, UserActivity_Staging, and UserActivity_Archive	
Alter the partition function and UserActivity_Staging constraints	
Alter the partition function and UserActivity_Archive constraints	

**Answer:**

## Dumps with VCE and PDF (+Free VCE Software)

Tasks	Maintenance Steps
an identity for UserActivityID starting at the next value	Convert UserActivity to use
a sequence for UserActivityID starting at the next value	
on-disk tables using the partitioning scheme	Copy UserActivity metadata to create UserActivity_Archive as
in-memory tables using the partitioning scheme	After copying UserActivity metadata to create UserActivity_Staging, create a view on top of
UserActivity and UserActivity_Archive	After switching a new partition from UserActivity_Staging into UserActivity_Archive,
UserActivity, UserActivity_Staging, and UserActivity_Archive	
Alter the partition function and UserActivity_Staging constraints	
Alter the partition function and UserActivity_Archive constraints	

### Explanation:

#### Maintenance Steps

Convert UserActivity to use

a sequence for UserActivityID starting at the next value

Copy UserActivity metadata to create UserActivity\_Archive as

on-disk tables using the partitioning scheme

After copying UserActivity metadata to create UserActivity\_Staging, create a view on top of

UserActivity and UserActivity\_Archive

After switching a new partition from UserActivity\_Staging into UserActivity\_Archive,

Alter the partition function and UserActivity\_Archive constraints

C:\Users\Kamran\Desktop\image.jpg

### Question No : 74 - (Topic 7)

You need to implement security for the restore and audit process. What should you do?

- A.** Grant the COFFECORP\Auditors group ALTER ANY CONNECTION and SELECT ALL USER SECURABLES permissions. Grant the COFFECORP\StoreAgent group ALTER ANY CONNECTION and IMPERSONATE ANY LOGIN permissions.
- B.** Grant the COFFECORP\Auditors group CONNECT ANY DATABASE and IMPERSONATE ANY LOGIN permissions. Grant the COFFECORP\StoreAgent group

CONNECT ANY DATABASE and SELECT ALL USER SECURABLES permissions.

**C.** Grant the COFFECORP\Auditors group ALTER ANY CONNECTION and IMPERSONATE ANY LOGIN permissions. Grant the COFFECORP\StoreAgent group ALTER ANY CONNECTION and SELECT ALL USER SECURABLES permissions.

**D.** Grant the COFFECORP\Auditors group CONNECT ANY DATABASE and SELECT ALL USER SECURABLES permissions. Grant the COFFECORP\StoreAgent group CONNECT ANY DATABASE and IMPERSONATE ANY LOGIN permissions.

**Answer: A**

### Question No : 75 - (Topic 7)

You need to modify the usp\_DetectSuspiciousActivity stored procedure.

Which two actions should you perform? Each correct answer presents part of the solution. Choose two.

A. Replace lines 04-06 with the following code:

```
BEGIN ATOMIC WITH
(
    DELAYED_DURABILITY = ON,
    TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED,
    LANGUAGE = N'English'
)
```

B. Replace lines 04-06 with the following code:

```
BEGIN ATOMIC WITH
(
    DELAYED_DURABILITY = ON,
    TRANSACTION ISOLATION LEVEL = REPEATABLE READ
)
```

C. Change the logic of the stored procedure to use separate UPDATE and INSERT statements.

D. Replace lines 07-09 with the following code:

```
DECLARE @exists BIT = 0
IF EXISTS ( SELECT TOP 1 * FROM POSTransaction (NOLOCK) WHERE StatusID = 4 and CreateDate
>= dateadd(second,-60, GETDATE() ))
```

E. Replace lines 04-06 with the following code:

```
BEGIN ATOMIC WITH
(
    TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED,
    LANGUAGE = N'English'
)
```

F. Replace lines 07-09 with the following code:

```
DECLARE @exists BIT = 0
SELECT TOP 1 @exists = 1 FROM POSTransaction WHERE StatusID >= 4 and CreateDate >= dateadd
(second,-60, GETDATE() )
IF @exists = 1
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer: D,E**

**Explanation:**

Note:

\* Move micropayments to dbo.POSException table by using a stored procedure named ups\_DetectSuspiciousActivity.

### **Question No : 76 - (Topic 7)**

You need to modify the stored procedure usp\_LookupConcurrentUsers.

What should you do?

- A. Use the summary table as an in-memory optimized table with a non-hash clustered index.
- B. Use the summary table as an in-memory optimized table with a non-hash nonclustered index.
- C. Use a type variable instead of the summary table.
- D. Add a clustered index to the summary table.

**Answer: A**

### **Question No : 77 - (Topic 8)**

You need to recommend a solution for the deployment of SQL Server 2014. The solution must meet the business requirements.

What should you include in the recommendation?



- A. Create a new instance of SQL Server 2014 on the server that hosts the SQL Server 2008 instance.
- B. Upgrade the existing SQL Server 2008 instance to SQL Server 2014.
- C. Deploy two servers that have SQL Server 2014 installed and implement Failover Clustering.
- D. Deploy two servers that have SQL Server 2014 installed and implement database mirroring.

**Answer: C**

**Explanation:** Scenario: The databases must be available if the SQL Server service fails.

Failover Clustering Overview

Windows Server Failover Clustering (WSFC) with SQL Server

### Question No : 78 - (Topic 8)

You need to recommend a feature to support your backup solution.

What should you include in the recommendation?

- A. Transparent Data Encryption (TDE)
- B. Column-level encryption
- C. An NTFS file permission
- D. A Secure Sockets Layer (SSL)

**Answer: A**

**Explanation:** \* Scenario: •You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

\* Transparent data encryption (TDE) performs real-time I/O encryption and decryption of the data and log files. The encryption uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery.

Transparent Data Encryption (TDE)

### Question No : 79 - (Topic 8)

You need to recommend a solution for Application1 that meets the security requirements.

What should you include in the recommendation?

- A. Signed stored procedures
- B. Certificate Authentication
- C. Encrypted columns
- D. Secure Socket Layer (SSL)

**Answer: A**

**Explanation:** \* Scenario:

/ Data from Database2 will be accessed periodically by an external application named Application1

/ Application developers must be denied direct access to the database tables. Applications must be denied direct access to the tables.

Tutorial: Signing Stored Procedures with a Certificate

### Question No : 80 - (Topic 8)

You need to recommend a solution to allow application users to perform tables. The solution must meet the business requirements.

What should you recommend?

- A. Create a Policy-Based Management Policy.
- B. Create a user-defined database role and add users to the role.
- C. Create stored procedures that use EXECUTE AS clauses.
- D. Create functions that use EXECUTE AS clauses.

**Answer: D**

**Explanation:** \* c Clause (Transact-SQL)

In SQL Server you can define the execution context of the following user-defined modules: functions (except inline table-valued functions), procedures, queues, and triggers.

Using EXECUTE AS in Modules

**Question No : 81 - (Topic 8)**

You need to recommend an isolation level for usp\_UpdateOrderDetails.

Which isolation level should recommend?

- A. Read committed
- B. Repeatable read
- C. Read uncommitted
- D. Serializable

**Answer: B**

**Explanation:** \* Scenario: Databasel will also contain a stored procedure named usp\_UpdateOrderDetails. The stored procedure is used to update order information. The stored procedure queries the Orders table twice each time the procedure executes. The rows returned from the first query must be returned on the second query unchanged along with any rows added to the table between the two read operations.

\* 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.

Transaction Isolation Levels

**Question No : 82 - (Topic 8)**

You need to recommend a solution to synchronize Database2 to App1\_Db1.

What should you recommend?

- A. Change data capture
- B. Snapshot replication
- C. Master Data Services
- D. Transactional replication

**Answer: D**

**Explanation:** Scenario:

\* Data from Database2 will be accessed periodically by an external application named Application1. The data from Database2 will be sent to a database named App1\_Db1 as soon as changes occur to the data in Database2.

\* All data in Database2 is recreated each day and does not change until the next data creation process.

Transactional Replication

### Question No : 83 - (Topic 8)

You need to recommend a disk monitoring solution that meets the business requirements.

What should you include in the recommendation?

- A. An audit
- B. A dynamic management view
- C. A maintenance plan
- D. A SQL Server Agent alert

**Answer: B**

**Explanation:** Dynamic Management Views and Functions (Transact-SQL)

### Question No : 84 - (Topic 8)

You need to recommend a solution to improve the performance of usp.UpdateInventory. The solution must minimize the amount of development effort.

What should you include in the recommendation?

- A. A table variable
- B. A common table expression
- C. A subquery
- D. A cursor

**Answer: A**

**Explanation:** \*Scenario: Database2 will contain a stored procedure named usp\_UpdateInventory. Usp\_UpdateInventory will manipulate a table that contains a self-join that has an unlimited number of hierarchies.

\* A table variable can be very useful to store temporary data and return the data in the table format.

table

\* Example: The following example uses a self-join to find the products that are supplied by more than one vendor.

Because this query involves a join of the ProductVendor table with itself, the ProductVendor table appears in two roles. To distinguish these roles, you must give the ProductVendor table two different aliases (pv1 and pv2) in the FROM clause. These aliases are used to qualify the column names in the rest of the query. This is an example of the self-join Transact-SQL statement:

```
USE AdventureWorks2008R2;
```

```
GO
```

```
SELECT DISTINCT pv1.ProductID, pv1.VendorID
```

```
FROM Purchasing.ProductVendor pv1
```

```
INNER JOIN Purchasing.ProductVendor pv2
```

```
ON pv1.ProductID = pv2.ProductID
```

```
AND pv1.VendorID <> pv2.VendorID
```

```
ORDER BY pv1.ProductID
```

Incorrect:

Not B: Using a CTE offers the advantages of improved readability and ease in maintenance of complex queries. The query can be divided into separate, simple, logical building blocks. These simple blocks can then be used to build more complex, interim CTEs until the final result set is generated.

**Question No : 85 - (Topic 8)**

You need to recommend a database reporting solution that meets the business requirements.

What should you include in the recommendation?

- A. Data collection
- B. Performance Monitor
- C. A maintenance plan
- D. A dynamic management view

**Answer: A**

**Explanation:** \* Scenario: System administrators must be able to run real-time reports on disk usage.

\* The data collector provides an historical report for each of the System Data collection sets. Each of the following reports use data that is stored in the management data warehouse:

Disk Usage Summary

Query Statistics History

Server Activity History

You can use these reports to obtain information for monitoring system capacity and troubleshooting system performance.

System Data Collection Set Reports

### Question No : 86 - (Topic 8)

During performance testing, you discover that database INSERT operations against the Inventory table are slow.

You need to recommend a solution to reduce the amount of time it takes to complete the INSERT operations.

What should you recommend?

- A. Partition the nonclustered index.
- B. Partition the Inventory table.snapshot replication
- C. Create a column store index.Master Data Services
- D. Drop the clustered index.change data capture

**Answer: A**

**Explanation:** Scenario: Database2 will contain a table named Inventory. Inventory will contain over 100 GB of data. The Inventory table will have two indexes: a clustered index on the primary key and a nonclustered index. The column that is used as the primary key will use the identity property.

## Topic 9, Contoso Ltd

### Overview

#### Application Overview

Contoso, Ltd., is the developer of an enterprise resource planning (ERP) application. Contoso is designing a new version of the ERP application. The previous version of the ERP application used SQL Server 2008 R2. The new version will use SQL Server 2014. The ERP application relies on an import process to load supplier data. The import process updates thousands of rows simultaneously, requires exclusive access to the database, and runs daily.

You receive several support calls reporting unexpected behavior in the ERP application. After analyzing the calls, you conclude that users made changes directly to the tables in the database.

#### Tables

**The current database schema contains a table named OrderDetails. The OrderDetails table contains information about the items sold for each purchase order.**

**OrderDetails stores the product ID, quantities, and discounts applied to each product in a purchase order. The product price is stored in a table named Products.**

The Products table was defined by using the SQL\_Latin1\_General\_CP1\_CI\_AS collation. A column named ProductName was created by using the varchar data type.

The database contains a table named Orders. Orders contains all of the purchase orders from the last 12 months. Purchase orders that are older than 12 months are stored in a table named OrdersOld.

#### Stored Procedures

The current version of the database contains stored procedures that change two tables. The following shows the relevant portions of the two stored procedures:

```
CREATE PROC Sales.Proc1
AS
BEGIN TRAN
UPDATE Sales.Table1 ...
UPDATE Sales.Table2 ...
COMMIT TRAN
GO
```

```
CREATE PROC Sales.Proc2
AS
BEGIN TRAN
UPDATE Sales.Table2 ...
UPDATE Sales.Table1 ...
COMMIT TRAN
GO
```

## Customer Problems

### Installation Issues

The current version of the ERP application requires that several SQL Server logins be set up to function correctly. Most customers set up the ERP application in multiple locations and must create logins multiple times.

### Index Fragmentation Issues

Customers discover that clustered indexes often are fragmented. To resolve this issue, the customers defragment the indexes more frequently.

All of the tables affected by fragmentation have the following columns that are used as the clustered index key:

Column	Data type
id	uniqueidentifier
lastModified	datetime
modifiedBy	varchar(200)

### Backup Issues

Customers who have large amounts of historical purchase order data report that backup time is unacceptable.

### Search Issues



Users report that when they search product names, the search results exclude product names that contain accents, unless the search string includes the accent.

### **Missing Data Issues**

Customers report that when they make a price change in the Products table, they cannot retrieve the price that the item was sold for in previous orders.

### **Query Performance Issues**

Customers report that query performance degrades very quickly. Additionally, the customers report that users cannot run queries when SQL Server runs maintenance tasks.

### **Import Issues**

During the monthly import process, database administrators receive many supports call from users who report that they cannot access the supplier data. The database administrators want to reduce the amount of time required to import the data.

### **Design Requirements**

#### **File Storage Requirements**

The ERP database stores scanned documents that are larger than 2 MB. These files must only be accessed through the ERP application. File access must have the best possible read and write performance.

#### **Data Recovery Requirements**

If the import process fails, the database must be returned to its prior state immediately.

#### **Security Requirements**

You must provide users with the ability to execute functions within the ERP application, without having direct access to the underlying tables.

#### **Concurrency Requirements**

You must reduce the likelihood of deadlocks occurring when Sales.Proc1 and Sales.Proc2 execute.

### **Question No : 87 - (Topic 9)**

You need to recommend a solution that meets the data recovery requirement.

What should you include in the recommendation?

- A. A differential backup
- B. A transaction log backup

- C. Snapshot isolation
- D. A database snapshot

**Answer: D**

**Explanation:** How Database Snapshots Work

**Question No : 88 - (Topic 9)**

You need to recommend a solution that reduces the time it takes to import the supplier data.

What should you include in the recommendation?

- A. Enable instant file initialization.
- B. Reorganize the indexes.
- C. Disable Resource Governor.
- D. Enable Auto Update Statistics.

**Answer: C**

**Explanation:** \* The ERP application relies on an import process to load supplier data. The import process updates thousands of rows simultaneously, requires exclusive access to the database, and runs daily.

Resource Governor

**Question No : 89 - (Topic 9)**

You need to recommend a solution that addresses the concurrency requirement.

What should you recommend?

- A. Break each stored procedure into two separate procedures, one that changes Sales.Table1 and one that changes Sales.Table2.
- B. Make calls to Sales.Proc1 and Sales.Proc2 synchronously.
- C. Call the stored procedures in a Distributed Transaction Coordinator (DTC) transaction.

D. Modify the stored procedures to update tables in the same order for all of the stored procedures.

**Answer: D**

**Explanation: \* Concurrency Requirements**

You must reduce the likelihood of deadlocks occurring when Sales.Proc1 and Sales.Proc2 execute.

**Question No : 90 - (Topic 9)**

You need to recommend a solution that addresses the index fragmentation and index width issue.

What should you include in the recommendation? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Change the data type of the lastModified column to smalldatetime.
- B. Remove the lastModified column from the clustered index.
- C. Change the data type of the modifiedBy column to tinyint.
- D. Change the data type of the id column to bigint.
- E. Remove the modifiedBy column from the clustered index.
- F. Remove the id column from the clustered index.

**Answer: B,E**

**Explanation:** Scenario: Index Fragmentation Issues

Customers discover that clustered indexes often are fragmented. To resolve this issue, the customers defragment the indexes more frequently.

All of the tables affected by fragmentation have the following columns that are used as the clustered index key:

Column	Data type
id	uniqueidentifier
lastModified	datetime
modifiedBy	varchar(200)

**Question No : 91 - (Topic 9)**

You need to recommend changes to the ERP application to resolve the search issue. The solution must minimize the impact on other queries generated from the ERP application.

What should you recommend changing?

- A. The collation of the Products table
- B. The index on the ProductName column
- C. The collation of the ProductName column
- D. The data type of the ProductName column

**Answer: C**

**Explanation:** [http://technet.microsoft.com/en-us/library/aa214408\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa214408(v=sql.80).aspx)

**Question No : 92 - (Topic 9)**

You need to recommend a solution that addresses the backup issue. The solution must minimize the amount of development effort.

What should you include in the recommendation?

- A. Indexed views
- B. Filegroups
- C. Table partitioning
- D. Indexes

**Answer: B**

**Explanation:** \* Backup Issues

Customers who have large amounts of historical purchase order data report that backup time is unacceptable.

\* For very large databases (and by that, I mean, at least 500gb, but more like 5-10tb or more), it can become too expensive to regularly run a straight full backup . So, where needed, you can choose to backup smaller pieces of the database by choosing to back up

one of the files or file groups that make up a database.

Using Filegroups and Files to Store Data

**Question No : 93 - (Topic 9)**

You need to recommend a solution that resolves the missing data issue. The solution must minimize the amount of development effort.

What should you recommend?

- A. Denormalize the Products table.
- B. Denormalize the OrderDetails table.
- C. Normalize the OrderDetails table.
- D. Normalize the Products table.

**Answer: D**

**Explanation:** \* Scenario:

/ Missing Data Issues

Customers report that when they make a price change in the Products table, they cannot retrieve the price that the item was sold for in previous orders.

/ The current database schema contains a table named OrderDetails. The OrderDetails table contains information about the items sold for each purchase order. OrderDetails stores the product ID, quantities, and discounts applied to each product in a purchase order. The product price is stored in a table named Products.

[http://technet.microsoft.com/en-us/library/aa933055\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa933055(v=sql.80).aspx)

**Question No : 94 - (Topic 9)**

You need to recommend a solution that addresses the file storage requirements.

What should you include in the recommendation?

- A. FileStream
- B. FileTable
- C. The varbinary data type
- D. The image data type

**Answer: B**

**Explanation:** \* Scenario: File Storage Requirements

The ERP database stores scanned documents that are larger than 2 MB. These files must only be accessed through the ERP application. File access must have the best possible read and write performance.

\* FileTables remove a significant barrier to the use of SQL Server for the storage and management of unstructured data that is currently residing as files on file servers. Enterprises can move this data from file servers into FileTables to take advantage of integrated administration and services provided by SQL Server. At the same time, they can maintain Windows application compatibility for their existing Windows applications that see this data as files in the file system.

FileTables (SQL Server)

### Question No : 95 - (Topic 9)

You need to recommend a solution that addresses the installation issues.

What should you include in the recommendation?

- A. Windows logins
- B. Server roles
- C. Contained users
- D. Database roles

**Answer: C**

**Explanation:** \* Scenario: **Installation Issues**

The current version of the ERP application requires that several SQL Server logins be set up to function correctly. Most customers set up the ERP application in multiple locations and must create logins multiple times.

\* Creating contained users enables the user to connect directly to the contained database. This is a very significant feature in high availability and disaster recovery scenarios such as in an AlwaysOn solution. If the users are contained users, in case of failover, people would

be able to connect to the secondary without creating logins on the instance hosting the secondary. This provides an immediate benefit.

Contained Databases

### Question No : 96 - (Topic 9)

You need to recommend a solution that addresses the security requirement.

What should you recommend?

- A. Revoke user permissions on the tables. Create stored procedures that manipulate data. Grant the users the EXECUTE permission on the stored procedures.
- B. Grant the users the SELECT permission on the tables. Create views that retrieve data from the tables. Grant the users the SELECT permission on the views.
- C. Deny the users SELECT permission on the tables. Create views that retrieve data from the tables. Grant the users the SELECT permission on the views.
- D. Deny the users the SELECT permission on the tables. Create stored procedures that manipulate data. Grant the users the EXECUTE permission on the stored procedures.

**Answer: C**

**Explanation:** \* Security Requirements

You must provide users with the ability to execute functions within the ERP application, without having direct access to the underlying tables.

### Topic 10, Fabrikam, Inc

#### Background

#### Corporate Information

Fabrikam, Inc. is a retailer that sells electronics products on the Internet. The company has a headquarters site and one satellite sales office.

You have been hired as the database administrator, and the company wants you to change the architecture of the Fabrikam ecommerce site to optimize performance and reduce downtime while keeping capital expenditures to a minimum. To help with the solution, Fabrikam has decided to use cloud resources as well as on-premise servers.

## Physical Locations

All of the corporate executives, product managers, and support staff are stationed at the headquarters office. Half of the sales force works at this location.

There is also a satellite sales office. The other half of the sales force works at the satellite office in order to have sales people closer to clients in that area. Only sales people work at the satellite location.

## Problem Statement

To be successful, Fabrikam needs a website that is fast and has a high degree of system uptime. The current system operates on a single server and the company is not happy with the single point of failure this presents. The current nightly backups have been failing due to insufficient space on the available drives and manual drive cleanup often needing to happen to get past the errors. Additional space will not be made available for backups on the HQ or satellite servers. During your investigation, you discover that the sales force reports are causing significant contention.

## Configuration

### Windows Logins

The network administrators have set up Windows groups to make it easier to manage security. Users may belong to more than one group depending on their role.

The groups have been set up as shown in the following table:

Group	Members
OurDomain\Management	All corporate executives
OurDomain\SalesStaff	All sales people
OurDomain\ProductionStaff	All product managers and support staff
OurDomain\AllUsers	Everyone
OurDomain \CustomerSupport	Customer support representatives

### Server Configuration

The IT department has configured two physical servers with Microsoft Windows Server 2012 R2 and SQL Server 2014 Enterprise Edition and one Windows Azure Server. There are two tiers of storage available for use by database files only a fast tier and a slower tier. Currently the data and log files are stored on the fast tier of storage only. If a possible use case exists, management would like to utilize the slower tier storage for data files.

The servers are configured as shown in the following table:



Location	Server
Company headquarters	HQ_Server
Satellite sales office	Satellite_Server
Microsoft Windows Azure (cloud)	Cloud_File Server

### Database

Currently all information is stored in a single database called ProdDB, created with the following script:

```
CREATE DATABASE ProdDB
GO
```

```
ALTER DATABASE ProdDB SET RECOVERY SIMPLE
GO
```

The Product table is in the Production schema owned by the ProductionStaff Windows group. It is the main table in the system so access to information in the Product table should be as fast as possible.

The columns in the Product table are defined as shown in the following table:

Column	Data type
ProductID	INT
ProductName	VARCHAR(100)
ProductDescription	VARCHAR(MAX)
ProductPrice	SMALLMONEY
QuantityOnHand	INT
ProductCost	SMALLMONEY
ProductSupplierID	INT

The SalesOrderDetail table holds the details about each sale. It is in the Sales schema owned by the SalesStaff Windows group.

This table is constantly being updated, inserted into, and read.

The columns in the SalesOrderDetail table are defined as shown in the following table:

Column	Data type
SalesOrderDetailID	INT
ProductID	INT
SalePrice	SMALLMONEY
SaleQuantity	INT

### Database Issues

The current database does not perform well. Additionally, a recent disk problem caused the system to go down, resulting in lost sales revenue. In reviewing the current system, you found that there are no automated maintenance procedures. The database is severely fragmented, and everyone has read and write access.

### Requirements

#### Database

The database should be configured to maximize uptime and to ensure that very little data is lost in the event of a server failure. To help with performance, the database needs to be modified so that it can support in-memory data, specifically for the Product table, which the CIO has indicated should be a memory-optimized table. The auto-update statistics option is set off on this database.

Only product managers are allowed to add products or to make changes to the name, description, price, cost, and supplier. The changes are made in an internal database and pushed to the Product table in ProdDB during system maintenance time. Product managers and others working at the headquarters location also should be able to generate reports that include supplier and cost information.

#### Customer data access

Customers access the company's website to order products, so they must be able to read product information such as name, description, and price from the Product table. When customers place orders, stored procedures called by the website update product quantity-on-hand values. This means the product table is constantly updated at random times.

#### Customer support data access

Customer support representatives need to be able to view and not update or change product information. Management does not want the customer support representatives to be able to see the product cost or any supplier information.

### Sales force data access

Sales people at both the headquarters office and the satellite office must generate reports that read from the Product and SalesOrderDetail tables. No updates or inserts are ever made by sales people. These reports are run at random times and there can be no reporting downtime to refresh the data set except during the monthly maintenance window. The reports that run from the satellite office are process intensive queries with large data sets. Regardless of which office runs a sales force report, the SalesOrderDetail table should only return valid, committed order data; any orders not yet committed should be ignored.

### Historical Data

The system should keep historical information about customers who access the site so that sales people can see how frequently customers log in and how long they stay on the site. The information should be stored in a table called Customer Access. Supporting this requirement should have minimal impact on production website performance.

### Backups

The recovery strategy for Fabrikam needs to include the ability to do point in time restores and minimize the risk of data loss by performing transaction log backups every 15 minutes.

### Database Maintenance

The company has defined a maintenance window every month when the server can be unavailable. Any maintenance functions that require exclusive access should be accomplished during that window.

### Project milestones completed

- ✍ Revoked all existing read and write access to the database, leaving the schema ownership in place.
- ✍ Configured an Azure storage container secured with the storage account name MyStorageAccount with the primary access key StorageAccountKey on the cloud file server.
- ✍ SQL Server 2014 has been configured on the satellite server and is ready for use.
- ✍ On each database server, the fast storage has been assigned to drive letter F:, and the slow storage has been assigned to drive letter D:.

### Question No : 97 DRAG DROP - (Topic 10)

The business requires a satellite office to have a local copy of the data to report against.

You want to implement a solution to support the requirements. You need to establish a new Availability Group between the two servers.

Develop the solution by selecting and arranging the required code blocks in the correct

### Microsoft Exams List

<a href="#">70-246 Dump PDF VCE</a>	<a href="#">70-485 Dump PDF VCE</a>	<a href="#">70-742 Dump PDF VCE</a>	<a href="#">98-366 Dump PDF VCE</a>
<a href="#">70-247 Dump PDF VCE</a>	<a href="#">70-486 Dump PDF VCE</a>	<a href="#">70-743 Dump PDF VCE</a>	<a href="#">98-367 Dump PDF VCE</a>
<a href="#">70-331 Dump PDF VCE</a>	<a href="#">70-487 Dump PDF VCE</a>	<a href="#">70-744 Dump PDF VCE</a>	<a href="#">98-368 Dump PDF VCE</a>
<a href="#">70-332 Dump PDF VCE</a>	<a href="#">70-488 Dump PDF VCE</a>	<a href="#">70-761 Dump PDF VCE</a>	<a href="#">98-369 Dump PDF VCE</a>
<a href="#">70-333 Dump PDF VCE</a>	<a href="#">70-489 Dump PDF VCE</a>	<a href="#">70-762 Dump PDF VCE</a>	<a href="#">98-372 Dump PDF VCE</a>
<a href="#">70-334 Dump PDF VCE</a>	<a href="#">70-490 Dump PDF VCE</a>	<a href="#">70-765 Dump PDF VCE</a>	<a href="#">98-373 Dump PDF VCE</a>
<a href="#">70-339 Dump PDF VCE</a>	<a href="#">70-491 Dump PDF VCE</a>	<a href="#">70-768 Dump PDF VCE</a>	<a href="#">98-374 Dump PDF VCE</a>
<a href="#">70-341 Dump PDF VCE</a>	<a href="#">70-492 Dump PDF VCE</a>	<a href="#">70-980 Dump PDF VCE</a>	<a href="#">98-375 Dump PDF VCE</a>
<a href="#">70-342 Dump PDF VCE</a>	<a href="#">70-494 Dump PDF VCE</a>	<a href="#">70-981 Dump PDF VCE</a>	<a href="#">98-379 Dump PDF VCE</a>
<a href="#">70-345 Dump PDF VCE</a>	<a href="#">70-496 Dump PDF VCE</a>	<a href="#">70-982 Dump PDF VCE</a>	<a href="#">MB2-700 Dump PDF VCE</a>
<a href="#">70-346 Dump PDF VCE</a>	<a href="#">70-497 Dump PDF VCE</a>	<a href="#">74-343 Dump PDF VCE</a>	<a href="#">MB2-701 Dump PDF VCE</a>
<a href="#">70-347 Dump PDF VCE</a>	<a href="#">70-498 Dump PDF VCE</a>	<a href="#">74-344 Dump PDF VCE</a>	<a href="#">MB2-702 Dump PDF VCE</a>
<a href="#">70-348 Dump PDF VCE</a>	<a href="#">70-499 Dump PDF VCE</a>	<a href="#">74-409 Dump PDF VCE</a>	<a href="#">MB2-703 Dump PDF VCE</a>
<a href="#">70-354 Dump PDF VCE</a>	<a href="#">70-517 Dump PDF VCE</a>	<a href="#">74-678 Dump PDF VCE</a>	<a href="#">MB2-704 Dump PDF VCE</a>
<a href="#">70-383 Dump PDF VCE</a>	<a href="#">70-532 Dump PDF VCE</a>	<a href="#">74-697 Dump PDF VCE</a>	<a href="#">MB2-707 Dump PDF VCE</a>
<a href="#">70-384 Dump PDF VCE</a>	<a href="#">70-533 Dump PDF VCE</a>	<a href="#">77-420 Dump PDF VCE</a>	<a href="#">MB2-710 Dump PDF VCE</a>
<a href="#">70-385 Dump PDF VCE</a>	<a href="#">70-534 Dump PDF VCE</a>	<a href="#">77-427 Dump PDF VCE</a>	<a href="#">MB2-711 Dump PDF VCE</a>
<a href="#">70-410 Dump PDF VCE</a>	<a href="#">70-640 Dump PDF VCE</a>	<a href="#">77-600 Dump PDF VCE</a>	<a href="#">MB2-712 Dump PDF VCE</a>
<a href="#">70-411 Dump PDF VCE</a>	<a href="#">70-642 Dump PDF VCE</a>	<a href="#">77-601 Dump PDF VCE</a>	<a href="#">MB2-713 Dump PDF VCE</a>
<a href="#">70-412 Dump PDF VCE</a>	<a href="#">70-646 Dump PDF VCE</a>	<a href="#">77-602 Dump PDF VCE</a>	<a href="#">MB2-714 Dump PDF VCE</a>
<a href="#">70-413 Dump PDF VCE</a>	<a href="#">70-673 Dump PDF VCE</a>	<a href="#">77-603 Dump PDF VCE</a>	<a href="#">MB2-715 Dump PDF VCE</a>
<a href="#">70-414 Dump PDF VCE</a>	<a href="#">70-680 Dump PDF VCE</a>	<a href="#">77-604 Dump PDF VCE</a>	<a href="#">MB2-716 Dump PDF VCE</a>
<a href="#">70-417 Dump PDF VCE</a>	<a href="#">70-681 Dump PDF VCE</a>	<a href="#">77-605 Dump PDF VCE</a>	<a href="#">MB2-717 Dump PDF VCE</a>
<a href="#">70-461 Dump PDF VCE</a>	<a href="#">70-682 Dump PDF VCE</a>	<a href="#">77-881 Dump PDF VCE</a>	<a href="#">MB2-718 Dump PDF VCE</a>
<a href="#">70-462 Dump PDF VCE</a>	<a href="#">70-684 Dump PDF VCE</a>	<a href="#">77-882 Dump PDF VCE</a>	<a href="#">MB5-705 Dump PDF VCE</a>
<a href="#">70-463 Dump PDF VCE</a>	<a href="#">70-685 Dump PDF VCE</a>	<a href="#">77-883 Dump PDF VCE</a>	<a href="#">MB6-700 Dump PDF VCE</a>
<a href="#">70-464 Dump PDF VCE</a>	<a href="#">70-686 Dump PDF VCE</a>	<a href="#">77-884 Dump PDF VCE</a>	<a href="#">MB6-701 Dump PDF VCE</a>
<a href="#">70-465 Dump PDF VCE</a>	<a href="#">70-687 Dump PDF VCE</a>	<a href="#">77-885 Dump PDF VCE</a>	<a href="#">MB6-702 Dump PDF VCE</a>
<a href="#">70-466 Dump PDF VCE</a>	<a href="#">70-688 Dump PDF VCE</a>	<a href="#">77-886 Dump PDF VCE</a>	<a href="#">MB6-703 Dump PDF VCE</a>
<a href="#">70-467 Dump PDF VCE</a>	<a href="#">70-689 Dump PDF VCE</a>	<a href="#">77-887 Dump PDF VCE</a>	<a href="#">MB6-704 Dump PDF VCE</a>
<a href="#">70-469 Dump PDF VCE</a>	<a href="#">70-692 Dump PDF VCE</a>	<a href="#">77-888 Dump PDF VCE</a>	<a href="#">MB6-705 Dump PDF VCE</a>
<a href="#">70-470 Dump PDF VCE</a>	<a href="#">70-695 Dump PDF VCE</a>	<a href="#">77-891 Dump PDF VCE</a>	<a href="#">MB6-884 Dump PDF VCE</a>
<a href="#">70-473 Dump PDF VCE</a>	<a href="#">70-696 Dump PDF VCE</a>	<a href="#">98-349 Dump PDF VCE</a>	<a href="#">MB6-885 Dump PDF VCE</a>
<a href="#">70-480 Dump PDF VCE</a>	<a href="#">70-697 Dump PDF VCE</a>	<a href="#">98-361 Dump PDF VCE</a>	<a href="#">MB6-886 Dump PDF VCE</a>
<a href="#">70-481 Dump PDF VCE</a>	<a href="#">70-698 Dump PDF VCE</a>	<a href="#">98-362 Dump PDF VCE</a>	<a href="#">MB6-889 Dump PDF VCE</a>
<a href="#">70-482 Dump PDF VCE</a>	<a href="#">70-734 Dump PDF VCE</a>	<a href="#">98-363 Dump PDF VCE</a>	<a href="#">MB6-890 Dump PDF VCE</a>
<a href="#">70-483 Dump PDF VCE</a>	<a href="#">70-740 Dump PDF VCE</a>	<a href="#">98-364 Dump PDF VCE</a>	<a href="#">MB6-892 Dump PDF VCE</a>
<a href="#">70-484 Dump PDF VCE</a>	<a href="#">70-741 Dump PDF VCE</a>	<a href="#">98-365 Dump PDF VCE</a>	<a href="#">MB6-893 Dump PDF VCE</a>

### Cisco Exams List

<a href="#">010-151 Dump PDF VCE</a>	<a href="#">350-018 Dump PDF VCE</a>	<a href="#">642-737 Dump PDF VCE</a>	<a href="#">650-667 Dump PDF VCE</a>
<a href="#">100-105 Dump PDF VCE</a>	<a href="#">352-001 Dump PDF VCE</a>	<a href="#">642-742 Dump PDF VCE</a>	<a href="#">650-669 Dump PDF VCE</a>
<a href="#">200-001 Dump PDF VCE</a>	<a href="#">400-051 Dump PDF VCE</a>	<a href="#">642-883 Dump PDF VCE</a>	<a href="#">650-752 Dump PDF VCE</a>
<a href="#">200-105 Dump PDF VCE</a>	<a href="#">400-101 Dump PDF VCE</a>	<a href="#">642-885 Dump PDF VCE</a>	<a href="#">650-756 Dump PDF VCE</a>
<a href="#">200-120 Dump PDF VCE</a>	<a href="#">400-151 Dump PDF VCE</a>	<a href="#">642-887 Dump PDF VCE</a>	<a href="#">650-968 Dump PDF VCE</a>
<a href="#">200-125 Dump PDF VCE</a>	<a href="#">400-201 Dump PDF VCE</a>	<a href="#">642-889 Dump PDF VCE</a>	<a href="#">700-001 Dump PDF VCE</a>
<a href="#">200-150 Dump PDF VCE</a>	<a href="#">400-251 Dump PDF VCE</a>	<a href="#">642-980 Dump PDF VCE</a>	<a href="#">700-037 Dump PDF VCE</a>
<a href="#">200-155 Dump PDF VCE</a>	<a href="#">400-351 Dump PDF VCE</a>	<a href="#">642-996 Dump PDF VCE</a>	<a href="#">700-038 Dump PDF VCE</a>
<a href="#">200-310 Dump PDF VCE</a>	<a href="#">500-006 Dump PDF VCE</a>	<a href="#">642-997 Dump PDF VCE</a>	<a href="#">700-039 Dump PDF VCE</a>
<a href="#">200-355 Dump PDF VCE</a>	<a href="#">500-007 Dump PDF VCE</a>	<a href="#">642-998 Dump PDF VCE</a>	<a href="#">700-101 Dump PDF VCE</a>
<a href="#">200-401 Dump PDF VCE</a>	<a href="#">500-051 Dump PDF VCE</a>	<a href="#">642-999 Dump PDF VCE</a>	<a href="#">700-104 Dump PDF VCE</a>
<a href="#">200-601 Dump PDF VCE</a>	<a href="#">500-052 Dump PDF VCE</a>	<a href="#">644-066 Dump PDF VCE</a>	<a href="#">700-201 Dump PDF VCE</a>
<a href="#">210-060 Dump PDF VCE</a>	<a href="#">500-170 Dump PDF VCE</a>	<a href="#">644-068 Dump PDF VCE</a>	<a href="#">700-205 Dump PDF VCE</a>
<a href="#">210-065 Dump PDF VCE</a>	<a href="#">500-201 Dump PDF VCE</a>	<a href="#">644-906 Dump PDF VCE</a>	<a href="#">700-260 Dump PDF VCE</a>
<a href="#">210-250 Dump PDF VCE</a>	<a href="#">500-202 Dump PDF VCE</a>	<a href="#">646-048 Dump PDF VCE</a>	<a href="#">700-270 Dump PDF VCE</a>
<a href="#">210-255 Dump PDF VCE</a>	<a href="#">500-254 Dump PDF VCE</a>	<a href="#">646-365 Dump PDF VCE</a>	<a href="#">700-280 Dump PDF VCE</a>
<a href="#">210-260 Dump PDF VCE</a>	<a href="#">500-258 Dump PDF VCE</a>	<a href="#">646-580 Dump PDF VCE</a>	<a href="#">700-281 Dump PDF VCE</a>
<a href="#">210-451 Dump PDF VCE</a>	<a href="#">500-260 Dump PDF VCE</a>	<a href="#">646-671 Dump PDF VCE</a>	<a href="#">700-295 Dump PDF VCE</a>
<a href="#">210-455 Dump PDF VCE</a>	<a href="#">500-265 Dump PDF VCE</a>	<a href="#">646-985 Dump PDF VCE</a>	<a href="#">700-501 Dump PDF VCE</a>
<a href="#">300-070 Dump PDF VCE</a>	<a href="#">500-275 Dump PDF VCE</a>	<a href="#">648-232 Dump PDF VCE</a>	<a href="#">700-505 Dump PDF VCE</a>
<a href="#">300-075 Dump PDF VCE</a>	<a href="#">500-280 Dump PDF VCE</a>	<a href="#">648-238 Dump PDF VCE</a>	<a href="#">700-601 Dump PDF VCE</a>
<a href="#">300-080 Dump PDF VCE</a>	<a href="#">500-285 Dump PDF VCE</a>	<a href="#">648-244 Dump PDF VCE</a>	<a href="#">700-602 Dump PDF VCE</a>
<a href="#">300-085 Dump PDF VCE</a>	<a href="#">500-290 Dump PDF VCE</a>	<a href="#">648-247 Dump PDF VCE</a>	<a href="#">700-603 Dump PDF VCE</a>
<a href="#">300-101 Dump PDF VCE</a>	<a href="#">500-801 Dump PDF VCE</a>	<a href="#">648-375 Dump PDF VCE</a>	<a href="#">700-701 Dump PDF VCE</a>
<a href="#">300-115 Dump PDF VCE</a>	<a href="#">600-199 Dump PDF VCE</a>	<a href="#">648-385 Dump PDF VCE</a>	<a href="#">700-702 Dump PDF VCE</a>
<a href="#">300-135 Dump PDF VCE</a>	<a href="#">600-210 Dump PDF VCE</a>	<a href="#">650-032 Dump PDF VCE</a>	<a href="#">700-703 Dump PDF VCE</a>
<a href="#">300-160 Dump PDF VCE</a>	<a href="#">600-211 Dump PDF VCE</a>	<a href="#">650-042 Dump PDF VCE</a>	<a href="#">700-801 Dump PDF VCE</a>
<a href="#">300-165 Dump PDF VCE</a>	<a href="#">600-212 Dump PDF VCE</a>	<a href="#">650-059 Dump PDF VCE</a>	<a href="#">700-802 Dump PDF VCE</a>
<a href="#">300-180 Dump PDF VCE</a>	<a href="#">600-455 Dump PDF VCE</a>	<a href="#">650-082 Dump PDF VCE</a>	<a href="#">700-803 Dump PDF VCE</a>
<a href="#">300-206 Dump PDF VCE</a>	<a href="#">600-460 Dump PDF VCE</a>	<a href="#">650-127 Dump PDF VCE</a>	<a href="#">810-403 Dump PDF VCE</a>
<a href="#">300-207 Dump PDF VCE</a>	<a href="#">600-501 Dump PDF VCE</a>	<a href="#">650-128 Dump PDF VCE</a>	<a href="#">820-424 Dump PDF VCE</a>
<a href="#">300-208 Dump PDF VCE</a>	<a href="#">600-502 Dump PDF VCE</a>	<a href="#">650-148 Dump PDF VCE</a>	<a href="#">840-425 Dump PDF VCE</a>
<a href="#">300-209 Dump PDF VCE</a>	<a href="#">600-503 Dump PDF VCE</a>	<a href="#">650-159 Dump PDF VCE</a>	
<a href="#">300-210 Dump PDF VCE</a>	<a href="#">600-504 Dump PDF VCE</a>	<a href="#">650-281 Dump PDF VCE</a>	
<a href="#">300-320 Dump PDF VCE</a>	<a href="#">640-692 Dump PDF VCE</a>	<a href="#">650-393 Dump PDF VCE</a>	
<a href="#">300-360 Dump PDF VCE</a>	<a href="#">640-875 Dump PDF VCE</a>	<a href="#">650-472 Dump PDF VCE</a>	
<a href="#">300-365 Dump PDF VCE</a>	<a href="#">640-878 Dump PDF VCE</a>	<a href="#">650-474 Dump PDF VCE</a>	
<a href="#">300-370 Dump PDF VCE</a>	<a href="#">640-911 Dump PDF VCE</a>	<a href="#">650-575 Dump PDF VCE</a>	
<a href="#">300-375 Dump PDF VCE</a>	<a href="#">640-916 Dump PDF VCE</a>	<a href="#">650-621 Dump PDF VCE</a>	
<a href="#">300-465 Dump PDF VCE</a>	<a href="#">642-035 Dump PDF VCE</a>	<a href="#">650-663 Dump PDF VCE</a>	
<a href="#">300-470 Dump PDF VCE</a>	<a href="#">642-732 Dump PDF VCE</a>	<a href="#">650-665 Dump PDF VCE</a>	
<a href="#">300-475 Dump PDF VCE</a>	<a href="#">642-747 Dump PDF VCE</a>	<a href="#">650-754 Dump PDF VCE</a>	

## HOT EXAMS

### Cisco

[100-105 Dumps VCE PDF](#)  
[200-105 Dumps VCE PDF](#)  
[300-101 Dumps VCE PDF](#)  
[300-115 Dumps VCE PDF](#)  
[300-135 Dumps VCE PDF](#)  
[300-320 Dumps VCE PDF](#)  
[400-101 Dumps VCE PDF](#)  
[640-911 Dumps VCE PDF](#)  
[640-916 Dumps VCE PDF](#)

### Microsoft

[70-410 Dumps VCE PDF](#)  
[70-411 Dumps VCE PDF](#)  
[70-412 Dumps VCE PDF](#)  
[70-413 Dumps VCE PDF](#)  
[70-414 Dumps VCE PDF](#)  
[70-417 Dumps VCE PDF](#)  
[70-461 Dumps VCE PDF](#)  
[70-462 Dumps VCE PDF](#)  
[70-463 Dumps VCE PDF](#)  
[70-464 Dumps VCE PDF](#)  
[70-465 Dumps VCE PDF](#)  
[70-480 Dumps VCE PDF](#)  
[70-483 Dumps VCE PDF](#)  
[70-486 Dumps VCE PDF](#)  
[70-487 Dumps VCE PDF](#)

### CompTIA

[220-901 Dumps VCE PDF](#)  
[220-902 Dumps VCE PDF](#)  
[N10-006 Dumps VCE PDF](#)  
[SY0-401 Dumps VCE PDF](#)