

KTest

更に上のクオリティ 更に上のサービス



問題集

<http://www.ktest.jp>

1年で無料進級することに提供する

Exam : 070-464J

**Title : Developing Microsoft SQL
Server 2012 Databases**

Version : DEMO

トピック 1、シナリオ 1

アプリケーション情報

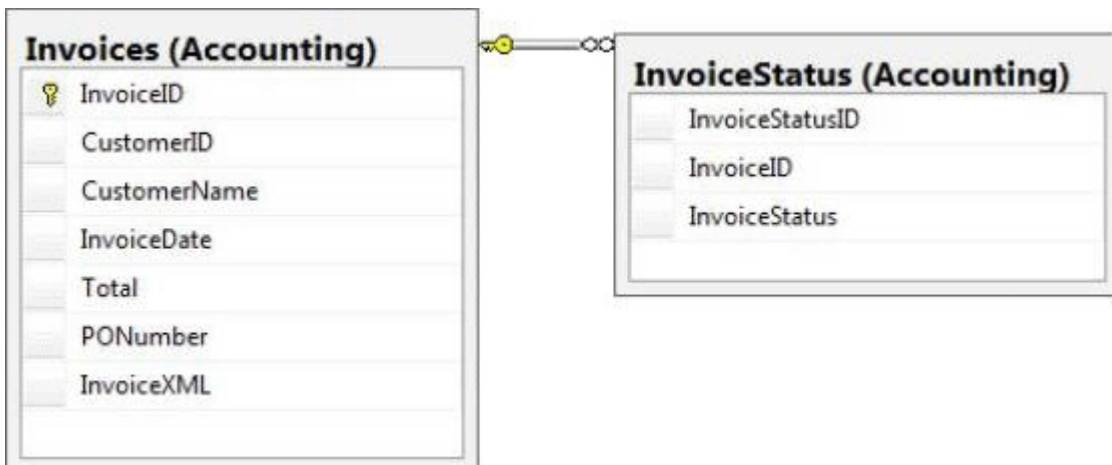
あなたの会社は顧客から XML 形式で請求書を受け取ります。現在は、請求書をファイルとして格納され、デスクトップアプリケーションによって処理される。アプリケーションは、いくつかのパフォーマンスとセキュリティの問題があります。

アプリケーションは、**SQL Server** ベースのソリューションに移行されている。請求書スキーマというスキーマは、請求書の XML 用に作成されています。

請求書のデータは、時々不完全です。不完全なデータが保存され、そのまま処理されなければならない。ユーザーはビューを提供されたデータをフィルタリングすることはできません。

あなたは、プロセスを受け、安全に請求書データを格納するために使用される **SQL Server** データベース **DB1** という名前を設計している。サードパーティ製の **Microsoft .NET Framework** コンポーネントは、税の計算を実行するために購入される。サードパーティの税コンポーネントは **Treytax.dll** という名前の **DLL** ファイルと **Amortize.cs** という名前のソース・コード・ファイルとして提供されます。コンポーネントは **Trey Research** 社という名前のクラスおよび償却（という名前のメソッドを公開します。ファイルは次の場所にあり **c:/temp/**。

次の図は計画されたテーブルを示している：



あなたは **Accounting.InvoiceID_Seq** という配列を有する。

あなたは **CERT1** と **CERT2** という名前の二つの証明書を作成する予定。あなたはマスターに **CERT1** が作成されます。あなたは **DB1** に **CERT2** が作成されます。

あなたは **DB1** に対して、動的 **T-SQL** 文を生成する機能を必要とするレガシーアプリケーションがあります。レガシー・アプリケーションによって生成されたクエリのサンプルが **Legacy.sql** に表示されます。

募集要項

計画されたデータベースには、次の要件があり：

- すべてのストアドプロシージャは、署名する必要があります。
- 元の XML 請求書をデータベースに格納する必要があります。
- XML スキーマは、請求書データを検証するために使用されなければなりません。
- ダイナミック **T-SQL** ステートメントは、ストアドプロシージャに変換する必要があります。
- .NET Framework** の税コンポーネントへのアクセスは、**T-SQL** オブジェクトに対して使用可能である必要があります。
- 列は、それぞれのテーブルで使用されるスペースの量を最小限にデータ型を使用して定義しなければなら

りません。

-請求書のステータステーブルに格納されている請求書は、請求書テーブルで 사용되는のと同じ識別子で請求書を参照する必要があります。

-バックアップディスクの盗難から保護するために、請求書データは、暗号化の最高レベルを使用して保護されなければなりません。

-ソリューションは、顧客が請求書をフィルタリングする機能をユーザに提供し、テーブル値関数を提供する必要があります。

-インデックスは管理作業の最小量を使用することで、断片化に基づいて、定期的に最適化する必要があります。

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

すべての顧客 ID は 11 桁です。顧客 ID の最初の 3 桁は、お客様の国を表しています。残りの 8 桁は、顧客の口座番号です。

以下は、XML 形式で顧客の請求書のサンプル:

```
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,
03     RowNumber int
04 );
05 DECLARE @TableName sysname, @IndexName sysname, @Fragmentation int,
06     @RowNumber int, @sqlcommand varchar(1000);
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;

```

1. あなたは、障害回復手順をテストしています。

あなたが別のサーバーに DB1 を復元しようとし、次のエラーメッセージが表示される:

"Msg 33111.

レベル 16、状態 3、行 1

拇印を使用してサーバー証明書が見つかりません。

,0xA694FBEA88C9354E5E2567C30A2A69E8FB4C44A9\

メッセージ 3013、レベル 16、状態 1、行 1

RESTORE DATABASE が異常終了しました。"

あなたが別のサーバーに DB1 を復元できることを確認する必要があります。

あなたはどちらのコードセグメントを実行する必要がありますか？

- ☐ 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

2. あなたは DB1 に InvoiceStatus テーブルを作成する必要があります。

どのようにして、CREATE TABLE ステートメントで InvoiceID 列を定義する必要がありますか？

- ☐ 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

3. どのデータ型には、得意先のために使うべきでしょうか？

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

Answer: D

説明:

Invoices.xml

すべての顧客 ID は 11 桁です。顧客 ID の最初の 3 桁は customer'scountry を表しています。残りの 8 桁は、顧客の口座番号です。

int: -2³¹ (-2,147,483,648) to 2³¹-1 (2,147,483,647) (just 10 digits max)

bigint: -2⁶³ (-9,223,372,036,854,775,808) to 2⁶³-1 (9,223,372,036,854,775,807)

<http://msdn.microsoft.com/en-us/library/ms176089.aspx>

<http://msdn.microsoft.com/en-us/library/ms187745.aspx>

4. あなたは、アプリケーションの要件に準拠する InsertInvoice を変更する必要があります。
あなたはどちらのコードセグメントを実行する必要がありますか？

- ☐ 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

5. あなたは `usp_InsertInvoice.sql` を使用して請求書処理しようとし、次のエラーメッセージが表示される:

"MSG515、レベル 16、状態 2、手順 `usp_InsertInvoice`、10 行目

列に NULL 値を挿入できません」を `invoiceDate`、テーブル `DB1.Accounting.Invoices` を;列には、NULL 値を許可しません。INSERT は失敗します。"

あなたは、エラーを解決するために `usp_InsertInvoice.sql` を変更する必要があります。

どのようにして、INSERT 文を変更する必要がありますか?

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

Answer: B

6. あなたは `CountryFromID` 内の関数を変更する必要があります。国名ではなく国 ID が返されることを保証するためにでしょうか?

あなたは `CountryFromID` のどのコード行を変更する必要がありますか?

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

Answer: D

説明:

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

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

7. あなたは `IndexManagement.sql` を実行し、次のエラーメッセージが表示される:

"MSG512、レベル 16、状態 1、行 12

サブクエリは複数の値を返しました。サブクエリが=これは、許可されていません! =、<、<=、>、>=、またはサブクエリが式として使用されている場合。"

あなたは `IndexManagement.sql` が適切に実行されることを確認する必要があります。

その WHILE ステートメントを使用すると、行 18 で使用する必要がありますか?

- 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 6indextable)`

Answer: B

8. あなたは .NET Framework プロセス製品コンポーネントを使用するようにデータベースを準備する必要があります。

あなたはどちらのコードセグメントを実行する必要がありますか? (各正解はソリューションの一部を提供します。当てはまるものをすべて選択してください。)

- ☐ A. `CREATE ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.DLL'`
- ☐ B. `RECONFIGURE;`
- ☐ C. `EXEC sp_recompile @objname = 'Production.ProcessProduct';`
- ☐ D. `CREATE TYPE Production.ProcessProduct
EXTERNAL NAME ProductionAssembly.ProcessProducts.Process;`
- ☐ E. `Exec SP_CONFIGURE 'clr enabled', '1';`
- ☐ F. `CREATE PROCEDURE Production.ProcessProduct(
 @ProductID int, @ProductType varchar(11)
)
AS EXTERNAL NAME ProductionAssembly.ProcessProducts.Process;`
- ☐ G. `CREATE ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.cs';`

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

Answer: A, B, D, E

説明:

<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>

9. ストアドプロシージャを使用する **Legacy.sql** の機能を変換する必要があります。
ストアドプロシージャは、どのコードセグメントが含まれている必要がありますか？

- ☐ 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

説明:

<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>

10. あなたが得意で請求書をフィルタリングする機能を作成する必要があります。関数の **SELECT** 文は `InvoicesByCustomer.sql` に含まれています。

どのコードセグメントは、あなたが機能を完了するために使用する必要がありますか？

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

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

Answer: A

11.DRAG DROP

あなたは、請求書の金額を償却ストアプロシージャを構築する必要があります。

どのコードには、ストアプロシージャを作成するために使用 **segmentsshould** でしょうか？（答える解答エリアに適切なコード **segmentsfrom** コードセグメントのリストを移動し、正しい順序でそれらを配置する。）

Ordered List Title	Answer Choices Title
<div> <div> <div></div> <div></div> </div> <div></div> </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>
	<div><< Move</div> <div>Remove >></div>

Answer:

```

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

```

説明:

<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>

トピック 2、シナリオ 2

アプリケーション情報

あなたは、SQL Server2012 がインストールされてい SQL1 と SQL2 という名前の二つのサーバーがある。あなたが会議をスケジュールし、管理するために使用されているアプリケーションがあります。ユーザーは、アプリケーションが多くのエラーがあり、非常に遅いことを報告しています。あなたが問題を解決するためにアプリケーションを更新しています。

アプリケーションをサポートするための **SQL1** 上に新しいデータベースを作成する予定です。ジュニアデータベース管理者はデータベースを作成するために使用されるすべてのスクリプトを作成しました。新しいデータベースのテーブルを作成するために使用する予定のスクリプトは **Tables.sql** に示されている。新しいデータベースのストアドプロシージャを作成するために使用する予定のスクリプトは **StoredProcedures.sql** に示されている。新しいデータベースのインデックスを作成するために使用する予定のスクリプトは **Indexes.sql** に示されている。(行番号は、参考のために含まれています。)

DB2 という名前のデータベースが **SQL2** に常駐します。 **DB2** は表の名前スピーカーへの変更を監査するという名前のテーブル **SpeakerAudit** を持っています。

usp_UpdateSpeakersName という名前のストアドプロシージャは、他のストアドプロシージャによって実行されます。 **usp_UpdateSpeakersName** の実行ストアドプロシージャは、常にトランザクションを処理します。

usp_SelectSpeakersByName という名前のストアドプロシージャは、話者の名前を取得するために使用されます。 **Usp_SelectSpeakersByName** はコミットされていないデータを読み取ることができます。

usp_GetFutureSessions という名前のストアドプロシージャは、将来的に発生するセッションを取得するために使用されます。

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, Abstract, DeliveryTime, TitleAndSpeaker)
38 SELECT SpeakerID, Title, Abstract, 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 );
```

1. あなたは、セッションテーブルとスピーカーテーブル間の参照整合性を提供する必要があります。
あなたは Tables.sql の 47 行で追加する必要があります、どのコードセグメントでしょうか？

- ☐ 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

説明:

<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>

2. あなたは `usp_TestSpeakers` を実行します。あなたは `usp_SelectSpeakersByName` は非効率的な実行計画を使用していることを発見する。あなたが最も効率的な実行計画が使用されることを保証するために `usp_SelectSpeakersByName` を更新する必要があります。

あなたは `Procedures.sql` の行 30 に追加すべきでしょうか？

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

Answer: C

説明:

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

3. あなたは `SQL1` は `usp_UpdateSpeakerName` の監査要件をサポートしていることを確実にするためのソリューションをお勧めする必要があります。

勧告で何を含めるべきでしょうか？

- A. 分散トランザクションコーディネータ (DTC)
B. トランザクションレプリケーション
C. データ・キャプチャを変更する
D. 追跡を変更する

Answer: A

4. あなたは、テーブルデザインを評価している。あなたは `usp_AttendeesReport` を実行するのにかかる時間が短縮さ `Tables.sql` への変更を推奨する必要があります。

あなたは `Tables.sql` の行 14 に追加すべきでしょうか？

- A. `FullNamenvarchar (100)NOTNULLCONSTRAINTDF_FullNameDEFAULT{dbo.CreateFullName (FirstName,LastName)},`
- B. `FullNameAS (FirstName+' '+LastName),`
- C. `FullNamenvarchar (100) NOTNULLDEFAULT (dbo.CreateFullName(FirstName,LastName)).`
- D. `FullName AS (FirstName + ' ' + LastName)PERSISTED,`

Answer: D

説明:

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

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

5. あなたは、サーバー側のページングをサポートするために `usp_SelectSpeakersByName` を変更する必要があります。解決策は、必要な開発作業の量を最小限に抑える必要があります。

あなたは `usp_SelectSpeakersByName` に何を追加する必要がありますか？

- A. テーブル変数
- B. `OFFSET-FETCH` 句
- C. `ROWNUMBER` キーワード
- D. 再帰共通テーブル式

Answer: B

説明:

<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/

6. あなたは、出席者のテーブルが確認さという名前の新しい列を追加する必要があります。

ソリューションは、次の要件を満たしている必要があります:

-偽のデフォルト値を持っています。

-使用されるディスク領域の量を最小限に抑えます。

あなたはどちらのコードブロックを使用する必要がありますか？

- 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 (l) DEFAULT '1';

Answer: A

説明:

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

7. あなたは `usp_InsertSessions` のパラメータで使用されるオブジェクトを作成する必要があります。あなたはどちらのステートメントを使用する必要がありますか？

- A. `CREATEXML SCHEMA COLLECTION SessionDataTable`
- B. `CREATETYPE SessionDataTable AS Table`
- C. `CREATE SCHEMA SessionDataTable`
- D. `CREATETABLE Sessior.EataTable`

Answer: B

8. 開発者は、`usp_UpdateSessionRoom` が定期的にエラー3960 を返すことを報告している。あなたは、エラーが発生しないようにする必要があります。解決策は、ストアドプロシージャが更新された行のすべてに元の値を返すことを確認する必要があります。

あなたは `Procedures.sql` に何を設定する必要がありますか？

- A. 次のコードで 46 行を置き換えます。
`SET TRANSACTION ISOLATION LEVEL SERIALIZABLE`
- B. 次のコードで 46 行を置き換えます。
`SET TRANSACTION ISOLATION LEVEL REPEATABLE READ`
- C. 57 行目に 49 行で `SELECT` 文を移動します。
- D. 53 行目に 46 行で、`SET` ステートメントを移動します。

Answer: A

9. あなたは `usp_UpdateSpeakerName` が同時に実行された場合 `usp.SelectSpeakersByName` がゆっくりと実行されることを発見します。あなたは `usp.SelectSpeakersByName` の実行時間を最小限に抑える必要がある。解決策は、他のストアドプロシージャのパフォーマンスに影響を及ぼしてはなりません。

あなたは何を更新する必要がありますか？

- A. `Usp_UpdateSpeakerName` は `NOLOCK` クエリヒントを使用する
- B. `Usp_UpdateSpeakerName` は、スナップショット・アイソレーションを使用する
- C. `Usp_SelectSpeakersByName` は `NOLOCK` クエリヒントを使用する
- D. `Usp_SelectSpeakersByName` は、スナップショット・アイソレーションを使用する

Answer: C

説明:

`NOLOCK`

`READUNCOMMITTED` に相当します。

`READUNCOMMITTED`

汚い読み込みが許可されるように指定します。