

## Pass Microsoft 70-459 Exam

Number: 70-459  
Passing Score: 700  
Time Limit: 180 min  
File Version: 36.7



## Pass Microsoft 70-459 Exam

**Exam Name:** Transition Your MCITP: Database Administrator 2008 or MCITP: Database Developer 2008  
to MCSE: Data Platform

## Exam A

### QUESTION 1

You have an application that uses a view to access data from multiple tables. You need to ensure that you can insert rows into the underlying tables by using the view. What should you do?

- A. Materialize the view.
- B. Create an INSTEAD OF trigger on the view.
- C. Define the view by using the CHECK option.
- D. Define the view by using the SCHEMABINDING option.

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms180800.aspx>

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>

### QUESTION 2

You create a view by using the following code:

```
CREATE VIEW dbo.View1
WITH VIEW_METADATA
AS
SELECT t1.col1, t1.col2, t2.*
FROM dbo.Table1 AS t1 JOIN dbo.Table2 AS t2 ON t1.col1=t2.col2;
```

Several months after you create the view, users report that the view has started to return unexpected results. You discover that the design of Table2 was modified since you created the view. You need to ensure that the view returns the correct results. Which code segment should you run?

- A. EXEC sp\_refreshview @viewname = 'dbo.View1';
- B. ALTER VIEW dbo.View1 WITH SCHEMABINDING, VIEW\_METADATA  
AS  
SELECT t1.col1, t1.col2, t2.\*  
FROM dbo.Table1 AS t1 JOIN dbo.Table2 AS t2  
ON t1.col1 = t2.col2;
- C. DROP VIEW dbo.View1;  
GO  
CREATE VIEW dbo.View1 WITH SCHEMABINDING, VIEW\_METADATA  
AS  
SELECT t1.col1, t1.col2, t2.\*  
FROM dbo.Table1 AS t1 JOIN dbo.Table2 AS t2  
ON t1.col1 = t2.col2;
- D. EXEC sp\_refreshsqlmodule @name = 'dbo.Table2';

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms173846.aspx>  
<http://msdn.microsoft.com/en-us/library/ms187956.aspx>  
<http://msdn.microsoft.com/en-us/library/ms187821.aspx>  
<http://msdn.microsoft.com/en-us/library/bb326754.aspx>

**QUESTION 3**

You are creating a table named Orders. You need to ensure that every time a new row is added to the Orders table, a user-defined function is called to validate the row before the row is added to the table. What should you use? More than one answer choice may achieve the goal. Select the BEST answer.

- A. a Data Definition Language (DDL) trigger
- B. a Data Manipulation Language (DML) trigger
- C. a DEFAULT constraint
- D. a FOREIGN KEY constraint
- E. a CHECK constraint

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to this reference, this answer looks correct.

Reference:

<http://www.techrepublic.com/blog/programming-and-development/comparing-sql-server-constraints-and-dml-triggers/402>  
<http://msdn.microsoft.com/en-us/library/ms178110.aspx>

**QUESTION 4**

You have an index for a table in a SQL Azure database. The database is used for Online Transaction Processing (OLTP). You discover that the index consumes more physical disk space than necessary. You need to minimize the amount of disk space that the index consumes. What should you set from the index options?

- A. STATISTICS\_NORECOMPUTE = OFF
- B. STATISTICS\_NORECOMPUTE = ON
- C. FILLFACTOR = 0
- D. FILLFACTOR = 80

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

Reference:

<http://msdn.microsoft.com/en-us/library/ms177459.aspx>  
<http://msdn.microsoft.com/en-us/library/ms188783.aspx>

**QUESTION 5**

Your company has a SQL Azure subscription. You implement a database named Database1. In Database1, you create two tables named Table1 and Table2. You create a stored procedure named sp1. Sp1 reads data from Table1 and inserts data into Table2. A user named User1 informs you that he is unable to run sp1. You verify that User1 has the SELECT permission on Table1 and Table2. You need to ensure that User1 can run sp1. The solution must minimize the number of permissions assigned to User1. What should you do?

- A. Grant User1 the INSERT permission on Table2.
- B. Add User1 to the db\_datawriter role.
- C. Grant User1 the EXECUTE permission on sp1.
- D. Change sp1 to run as the sa user.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to this reference, the answer looks correct.

Reference:

<http://msdn.microsoft.com/en-us/library/ms191291.aspx>

### QUESTION 6

You are designing a SQL Server database for an order fulfillment system. You create a table named Sales.Orders by using the following script:

```
CREATE TABLE Sales.Orders
(
    OrderID int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    OrderDate date NOT NULL,
    CustomerID int NOT NULL
);
```

Each order is tracked by using one of the following statuses:

- Fulfilled
- Shipped
- Ordered
- Received

You need to design the database to ensure that you can retrieve the status of an order on a given date. The solution must ensure that new statuses can be added in the future. What should you do? More than one answer choice may achieve the goal. Select the BEST answer.

- A. To the Sales.Orders table, add a column named Status that will store the order status. Update the Status column as the order status changes.
- B. To the Sales.Orders table, add three columns named FulfilledDate, ShippedDate, and ReceivedDate. Update the value of each column from null to the appropriate date as the order status changes.
- C. Implement change data capture on the Sales.Orders table.
- D. Create a new table named Sales.OrderStatus that contains three columns named OrderID, StatusDate, and Status. Insert new rows into the table as the order status changes.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms191178.aspx>

<http://msdn.microsoft.com/en-us/library/cc645937.aspx>

### QUESTION 7

You are troubleshooting an application that runs a query. The application frequently causes deadlocks. You need to identify which transaction causes the deadlock. What should you do? More than one answer choice may achieve the goal. Select the BEST answer.

- A. Query the sys.dm\_exec\_sessions dynamic management view.
- B. Create an extended events session to capture deadlock information.
- C. Query the sys.dm\_exec\_requests dynamic management view.
- D. Create a trace in SQL Server Profiler that contains the Deadlock graph event.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://www.sqlservercentral.com/blogs/james-sql-footprint/2012/08/12/monitor-deadlock-in-sql-2012/>  
[http://blogs.technet.com/b/mspfe/archive/2012/06/28/how\\_2d00\\_to\\_2d00\\_monitor\\_2d00\\_deadlocks\\_2d00\\_in\\_2d00\\_sql\\_2d00\\_server.aspx](http://blogs.technet.com/b/mspfe/archive/2012/06/28/how_2d00_to_2d00_monitor_2d00_deadlocks_2d00_in_2d00_sql_2d00_server.aspx)  
<http://msdn.microsoft.com/en-us/library/ms177648.aspx>  
<http://msdn.microsoft.com/en-us/library/ms176013.aspx>  
<http://msdn.microsoft.com/en-us/library/ms188246.aspx>

### QUESTION 8

You have a SQL Azure database. You need to identify which keyword must be used to create a view that will be indexed. Which keyword should you identify?

- A. DISTINCT
- B. DEFAULT
- C. SCHEMABINDING
- D. VIEW\_METADATA

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>  
<http://msdn.microsoft.com/en-us/library/ms191432.aspx>

### QUESTION 9

You have an index for a table in a SQL Azure database. The database is used for Online Transaction Processing (OLTP). You discover that many page splits occur when records are inserted or updated in the table. You need to minimize the number of page splits. What should you set from the index options?

- A. FILLFACTOR = 0
- B. STATISTICS\_NORECOMPUTE = ON
- C. STATISTICS\_NORECOMPUTE = OFF
- D. FILLFACTOR = 80

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms188783.aspx>

<http://msdn.microsoft.com/en-us/library/ms177459.aspx>

#### **QUESTION 10**

You are creating a table named Orders. You need to ensure that every time a new row is added to the Orders table, a table that is used for auditing is updated. What should you use? More than one answer choice may achieve the goal. Select the BEST answer.

- A. a DEFAULT constraint
- B. a Data Definition Language (DDL) trigger
- C. a CHECK constraint
- D. a FOREIGN KEY constraint
- E. a data manipulation language (DML) trigger

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://www.techrepublic.com/blog/programming-and-development/comparing-sql-server-constraints-and-dml-triggers/402>

<http://msdn.microsoft.com/en-us/library/ms178110.aspx>

#### **QUESTION 11**

You have a SQL Server 2012 database named DB1. You have a backup device named Device1. You discover that the log file for the database is full. You need to ensure that DB1 can complete transactions. The solution must not affect the chain of log sequence numbers (LSNs). Which code segment should you execute?

- A. `BACKUP LOG DB1 TO Device1 WITH TRUNCATE_ONLY`
- B. `BACKUP LOG DB1 TO Device1 WITH COPY_ONLY`
- C. `BACKUP LOG DB1 TO Device1 WITH NORECOVERY`
- D. `BACKUP LOG DB1 TO Device1`

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms186865.aspx>

<http://msdn.microsoft.com/en-us/library/ms179478.aspx>

<http://msdn.microsoft.com/en-us/library/ms190925.aspx>

#### **QUESTION 12**

You are creating a table to support an application that will cache data outside of SQL Server. The application will detect whether cached values were changed before it updates the values. You need to create the table, and then verify that you can insert a row into the table. Which code segment should you use?

- A. 

```
CREATE TABLE Table1 (  
    ID int IDENTITY(1,1),  
    Name varchar(100),  
    Version uniqueidentifier DEFAULT NEWID()  
INSERT INTO Table1 (Name, Version)  
VALUES ('Smith, Ben')
```
- B. 

```
CREATE TABLE Table1 (  
    ID int IDENTITY(1,1),  
    Name varchar(100),  
    Version rowversion  
INSERT INTO Table1 (Name, Version)  
VALUES ('Smith, Ben', NEWID())
```
- C. 

```
CREATE TABLE Table1 (  
    ID int IDENTITY(1,1),  
    Name varchar(100),  
    Version uniqueidentifier DEFAULT NEWID()  
INSERT INTO Table1 (Name, Version)  
VALUES ('Smith, Ben', NEWID())
```
- D. 

```
CREATE TABLE Table1 (  
    ID int IDENTITY(1,1),  
    Name varchar(100),  
    Version rowversion  
INSERT INTO Table1 (Name)  
VALUES ('Smith, Ben')
```

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms182776.aspx>

<http://msdn.microsoft.com/en-us/library/ms187942.aspx>

<http://msdn.microsoft.com/en-us/library/ms190348.aspx>

### QUESTION 13

You use SQL Server 2012 to maintain the data used by the applications at your company. You plan to create a table named Table1 by using the following statement. (Line numbers are included for reference only.)

```
01 CREATE TABLE dbo.table1(  
02 ID int IDENTITY(1,1) NOT NULL,  
03  
04 Email varchar(100) NULL,  
05 CONSTRAINT PK_table1 PRIMARY KEY CLUSTERED(ID ASC)  
06 )
```

You need to ensure that Table1 contains a column named UserName. The UserName column will:

- Store string values in any language.
- Accept a maximum of 200 characters.
- Be case-sensitive and accent-sensitive.

Which code segment should you add at line 03?

- A. `UserName nvarchar(200) COLLATE Latin1_General_CI_AI NOT NULL,`
- B. `UserName varchar(200) COLLATE Latin1_General_CI_AI NOT NULL,`
- C. `UserName nvarchar(200) COLLATE Latin1_General_CS_AS NOT NULL,`
- D. `UserName varchar(200) COLLATE Latin1_General_CS_AS NOT NULL,`
- E. `UserName nvarchar(200) COLLATE Latin1_General_CI_AS NOT NULL,`
- F. `UserName varchar(200) COLLATE Latin1_General_CI_AS NOT NULL,`

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

A lot of the questions had a minor change that you need to pay attention to. Otherwise if you simply remember the answer, you will get it wrong.

example 1. a question that had the table column specifications of "english" would normally result in the answer being "varchar", but the question now changed to "international", so you need to know that the answer is "nvarchar".

example 2. "case sensitive, accent sensitive" would usually be "CS\_AS", but now the question changed to "case insensitive, accent sensitive" so its "CI\_AS".

So while to the untrained eye, the questions appear the same. They are only 99% the same, but the specifications may have changed slightly. So pay attention.

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms184391.aspx>

<http://msdn.microsoft.com/en-us/library/ms143726.aspx>

<http://msdn.microsoft.com/en-us/library/ff848763.aspx>

#### **QUESTION 14**

You are creating a database that will store usernames and credit card numbers for an application. You need to recommend a solution to store the credit card numbers in the database. What should you recommend? More than one answer choice may achieve the goal. Select the BEST answer.

- A. One-way encryption
- B. Reversible encryption
- C. Encrypting File System (EFS)
- D. Transparent Data Encryption (TDE)

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:



[http://technet.microsoft.com/en-us/library/hh994559\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/hh994559(v=ws.10).aspx)  
<http://msdn.microsoft.com/en-us/library/bb964742.aspx>  
<http://msdn.microsoft.com/en-us/library/bb510663.aspx>

#### QUESTION 15

You have a database named DB1. You plan to create a stored procedure that will insert rows into three different tables. Each insert must use the same identifying value for each table, but the value must increase from one invocation of the stored procedure to the next. Occasionally, the identifying value must be reset to its initial value. You need to design a mechanism to hold the identifying values for the stored procedure to use. What should you do? More than one answer choice may achieve the goal. Select the BEST answer.

- A. Create a sequence object that holds the next value in the sequence. Retrieve the next value by using the stored procedure. Reset the value by using an ALTER SEQUENCE statement as needed.
- B. Create a fourth table that holds the next value in the sequence. At the end each transaction, update the value by using the stored procedure. Reset the value as needed by using an UPDATE statement.
- C. Create a sequence object that holds the next value in the sequence. Retrieve the next value by using the stored procedure. Increment the sequence object to the next value by using an ALTER SEQUENCE statement. Reset the value as needed by using a different ALTER SEQUENCE statement.
- D. Create an identity column in each of the three tables. Use the same seed and the same increment for each table. Insert new rows into the tables by using the stored procedure. Use the DBCC CHECKIDENT command to reset the columns as needed.

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ff878091.aspx>  
<http://msdn.microsoft.com/en-us/library/ms176057.aspx>  
<http://msdn.microsoft.com/en-us/library/ff878572.aspx>  
<http://msdn.microsoft.com/en-us/library/ff878058.aspx>

#### QUESTION 16

**DRAG DROP**

You plan to install two SQL Server 2012 environments named Environment1 and Environment2. Your company identifies the following availability requirements for each environment:

- Environment1 must have Mirroring with automatic failover implemented.
- Environment2 must have AlwaysOn with automatic failover implemented.

You need to identify the minimum number of SQL Server 2012 servers that must be deployed to each environment to ensure that all data remains available if a physical server fails. How many servers should you identify? To answer, drag the appropriate number to the correct environment in the answer area.

**Select and Place:**

Number of Servers	Answer Area
2 3 4	Environment1 #
	Environment2 #

**Correct Answer:**

Number of Servers	Answer Area
2 3 4	Environment1 3
	Environment2 2

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms189852.aspx>

<http://msdn.microsoft.com/en-us/library/hh510230.aspx>

### QUESTION 17

DRAG DROP

You plan to deploy SQL Server 2012. You identify the following security requirements for the deployment:

- Users must be prevented from intercepting and reading the T-SQL statements sent from the clients to the database engine.
- All database files and log files must be encrypted if the files are moved to another disk on another server.

You need to identify which feature meets each security requirement. The solution must minimize processor overhead. Which features should you identify? To answer, drag the appropriate feature to the correct requirement in the answer area.

**Select and Place:**

## Features

Encrypting File System (EFS)

Policy-Based Management

Secure Socket Layer (SSL)

Transparent Data Encryption (TDE)

Windows BitLocker Drive Encryption (BitLocker)

## Answer Area

Users must be prevented from intercepting and reading the T-SQL statements sent from the clients to the database engine.

Feature

All database files and log files must be encrypted if the files are moved to another disk on another server.

Feature

**Correct Answer:**

## Features

Encrypting File System (EFS)

Policy-Based Management

Secure Socket Layer (SSL)

Transparent Data Encryption (TDE)

Windows BitLocker Drive Encryption (BitLocker)

## Answer Area

Users must be prevented from intercepting and reading the T-SQL statements sent from the clients to the database engine.

Secure Socket Layer (SSL)

All database files and log files must be encrypted if the files are moved to another disk on another server.

Transparent Data Encryption (TDE)

**Section: (none)**

## Explanation

### Explanation/Reference:

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/windows/desktop/aa364223.aspx>

<http://msdn.microsoft.com/en-us/library/bb510667.aspx>

<http://msdn.microsoft.com/en-us/library/bb879935.aspx>

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

<http://msdn.microsoft.com/en-us/library/windows/hardware/gg487306.aspx>

<http://msdn.microsoft.com/en-us/library/ff773063.aspx>

## QUESTION 18

### DRAG DROP

You execute the following code:

```
CREATE TABLE Customers(  
ID int PRIMARY KEY,  
Name nchar(10))  
GO
```

You discover that the Customers table was created in the dbo schema. You need to create a code segment to move the table to another schema named Schema2. What should you create? To answer, drag the appropriate code segments to the correct location in the answer area.

### Build List and Reorder:

Ordered List Title	Answer Choices Title
<div><div>▲</div><div>▼</div></div> <div></div>	<div>ALTER SCHEMA</div> <div>ALTER TABLE</div> <div>dbo</div> <div>dbo.Customers</div> <div>EXEC sp_rename</div> <div>TRANSFER</div> <div>Schema2</div> <div>SET SCHEMA =</div>
	<div>&lt;&lt; Move</div> <div>Remove &gt;&gt;</div>

### Correct Answer:

```
ALTER SCHEMA  
Schema2  
TRANSFER  
dbo.Customers
```

### Section: (none)

### Explanation

### Explanation/Reference:

According to these references, the answer looks correct.

Reference:

<http://msdn.microsoft.com/en-us/library/ms173423.aspx>

### QUESTION 19

#### DRAG DROP

You plan to deploy SQL Server 2012. Your company identifies the following monitoring requirements for the database:

- An e-mail message must be sent if the SQL Server Authentication mode changes.
- An e-mail message must be sent if CPU utilization exceeds 90 percent.

You need to identify which feature meets each monitoring requirement. Which features should you identify? To answer, drag the appropriate feature to the correct monitoring requirement in the answer area.

#### Select and Place:

Features	Answer Area
Policy-Based Management	An e-mail message must be sent if the SQL Server Authentication mode changes. <div>Feature</div>
a SQL Server Agent alert	An e-mail message must be sent if CPU utilization exceeds 90 percent. <div>Feature</div>
SQL Server Integration Services (SSIS)	
trace flags	

#### Correct Answer:

Features	Answer Area
Policy-Based Management	An e-mail message must be sent if the SQL Server Authentication mode changes. <div>Policy-Based Management</div>
a SQL Server Agent alert	An e-mail message must be sent if CPU utilization exceeds 90 percent. <div>a SQL Server Agent alert</div>
SQL Server Integration Services (SSIS)	
trace flags	

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/bb510667.aspx>

<http://msdn.microsoft.com/en-us/library/ms180982.aspx>

<http://msdn.microsoft.com/en-us/library/ms141026.aspx>

<http://msdn.microsoft.com/en-us/library/ms188396.aspx>

**QUESTION 20**

**DRAG DROP**

You are designing two stored procedures named Procedure1 and Procedure2. You identify the following requirements:

- Procedure1 must take a parameter that ensures that multiple rows of data can pass into the stored procedure.
- Procedure2 must use business logic that resides in a Microsoft .NET Framework assembly.

You need to identify the appropriate technology for each stored procedure. Which technologies should you identify? To answer, drag the appropriate technology to the correct stored procedure in the answer area. (Answer choices may be used once, more than once, or not at all.)

**Select and Place:**

Technologies		Answer Area
Common language runtime (CLR)		Procedure1 Technology
Extensible Markup Language (XML)		Procedure2 Technology
a table-valued parameter (TVP)		

**Correct Answer:**

Technologies		Answer Area
Common language runtime (CLR)		Procedure1 a table-valued parameter
Extensible Markup Language (XML)		Procedure2 Common language runtime
a table-valued parameter (TVP)		

**Section: (none)**

**Explanation**



**Explanation/Reference:**

According to these references, the answer looks correct.

**References:**

<http://msdn.microsoft.com/en-us/library/ms131102.aspx>

<http://msdn.microsoft.com/en-us/library/bb522446.aspx>

<http://msdn.microsoft.com/en-us/library/bb510489.aspx>

**QUESTION 21****DRAG DROP**

You are designing a database for a university. The database will contain two tables named Classes and StudentGrades that have the following specifications:

- Classes will store brochures in the XPS format.
- The brochures must be structured in folders and must be accessible by using UNC paths.
- StudentGrades must be backed up on a separate schedule than the rest of the database.

You need to identify which SQL Server technology meets the specifications of each table. Which technologies should you identify? To answer, drag the appropriate technology to the correct table in the answer area.

**Select and Place:**

Technologies	Answer Area	
FileStream	Technology	Classes
FileTable	Technology	StudentGrades
Filegroup		
Partitioned views		

**Correct Answer:**

Technologies	Answer Area	
FileStream	FileTable	Classes
FileTable	Filegroup	StudentGrades
Filegroup		
Partitioned views		

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/gg471497.aspx>

<http://msdn.microsoft.com/en-us/library/ff929144.aspx>

<http://msdn.microsoft.com/en-us/library/ms189563.aspx>

<http://msdn.microsoft.com/en-us/library/ms190174.aspx>

<http://msdn.microsoft.com/en-us/library/ms187956.aspx>

**QUESTION 22**

**DRAG DROP**

You have a SQL Azure database named Database1. You need to design the schema for a table named table1. Table1 will have less than one million rows. Table1 will contain the following information for each row:

Column	Description
ID	An incremental numeric value used to identify the row
Name	A string in English
Code	An alphanumeric code that has five characters
ModifiedDate	The date of the last modification

The solution must minimize the amount of space used to store each row. Which data types should you recommend for each column? To answer, drag the appropriate data type to the correct column in the answer area.

**Select and Place:**



Data Types	Answer Area	
int	ID	Data type
bigint	Name	Data type
varchar	Code	Data type
nvarchar	ModifiedDate	Data type
char		
smalldatetime		
date		

Correct Answer:

Data Types	Answer Area	
int	ID	int
bigint	Name	varchar
varchar	Code	char
nvarchar	ModifiedDate	date
char		
smalldatetime		
date		

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to this reference, the answer looks correct.

Reference:

<http://msdn.microsoft.com/en-US/library/ms187752.aspx>

**QUESTION 23**

**DRAG DROP**

You are designing an authentication strategy for a new server that has SQL Server 2012 installed. The strategy must meet the following business requirements:

- The account used to generate reports must be allowed to make a connection during certain hours only.
- Failed authentication requests must be logged.

You need to recommend a technology that meets each business requirement. The solution must minimize the amount of events that are logged. Which technologies should you recommend? To answer, drag the appropriate solution to the correct business requirement in the answer area.

**Select and Place:**

Technologies	Answer Area
login auditing	The account used to generate reports must be allowed to make a connection during certain hours only.
logon triggers	Technology
C2 audit tracing	Failed authentication requests must be logged.
Policy-Based Management	Technology

**Correct Answer:**

Technologies	Answer Area
login auditing	The account used to generate reports must be allowed to make a connection during certain hours only.
logon triggers	logon triggers
C2 audit tracing	Failed authentication requests must be logged.
Policy-Based Management	login auditing

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms175850.aspx>

<http://msdn.microsoft.com/en-us/library/bb326598.aspx>

<http://msdn.microsoft.com/en-us/library/ms187634.aspx>

<http://msdn.microsoft.com/en-us/library/bb510667.aspx>

#### **QUESTION 24**

You have a SQL Server 2012 database named Database1. You execute the following code:

```

CREATE TABLE Sales
(
    ID int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    OrderDate char(10) NOT NULL,
    Amount decimal
);
GO

CREATE INDEX IX_Sales_OrderDate
ON Sales(OrderDate)
INCLUDE (ID, Amount);
GO

CREATE PROC usp_Proc1(
    @date1 datetime,
    @date2 datetime
)
AS
SELECT ID, OrderDate, Amount
FROM Sales
WHERE CAST(OrderDate AS datetime)
    BETWEEN @date1 AND @date2
ORDER BY ID;
GO

```

You insert 3 million rows into Sales. You need to reduce the amount of time it takes to execute Proc1. What should you do?

- A. Run the following: ALTER TABLE Sales ALTER COLUMN OrderDate datetime NOT NULL;
- B. Change the WHERE clause to the following: WHERE OrderDate BETWEEN CAST(@date1, char(10)) AND CAST(@date2, char(10))
- C. Remove the ORDER BY clause from the stored procedure.
- D. Run the following:
 

```

DROP INDEX IX_Sales_OrderDate;
GO
CREATE INDEX IX_Sales_OrderDate ON Sales(OrderDate);
GO

```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Thiago from Brazil - Aug 06 2013, 2:43 PM Report Spam

Exam A / Q39 -> is wrong

C is correct (Remove the ORDER BY clause from the stored procedure.)

-- BURGOS

I Disagree with previous answer (D) because it will force a lookup (by retiring "amount" column) and proc will use NonSARG and will continue to perform a index scan anyway. Predicting a selective range of dates, to

prevent Index Scan (in 3MM rows) is better change procedure (B).

In fact, I think that would be better the following steps:

- drop index and PK;
- change datatype of OrderDate to datetime;
- recreate PK as CLUSTERED;
- recreate index without ID in INCLUDE Clause (because ID is part of cluster now)
- recreate procedure without any "cast" function and without order by (because ID is natural order).

According to these references, the answer looks correct.

Reference:

[http://www.c-sharpcorner.com/UploadFile/skumaar\\_mca/good-practices-to-write-the-stored-procedures-in-sql-server/](http://www.c-sharpcorner.com/UploadFile/skumaar_mca/good-practices-to-write-the-stored-procedures-in-sql-server/)

### QUESTION 25

You plan to design an application that temporarily stores data in a SQL Azure database. You need to identify which types of database objects can be used to store data for the application. The solution must ensure that the application can make changes to the schema of a temporary object during a session. Which type of objects should you identify?

- A. Common table expressions (CTEs)
- B. Temporary tables
- C. Table variables
- D. Temporary stored procedures

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms175972.aspx>

<http://msdn.microsoft.com/en-us/library/ms189084.aspx>

<http://msdn.microsoft.com/en-us/library/ms175010.aspx>

<http://msdn.microsoft.com/en-us/library/bb510489.aspx>

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

<http://zacksfiasco.com/post/2010/01/21/SQL-Server-Temporary-Stored-Procedures.aspx>

### QUESTION 26

You have a table named Table1 that contains 1 million rows. Table1 contains a column named Column1 that stores sensitive information. Column1 uses the nvarchar(16) data type. You have a certificate named Cert1. You need to replace Column1 with a new encrypted column that uses one-way hashing. Which code segment should you execute before you remove Column1? To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.

**Build List and Reorder:**

Ordered List Title	Answer Choices Title
<div> <div>▲</div> <div>▼</div> </div> <div></div>	<pre> ALTER TABLE Table1 ADD Column2 nvarchar(256); ALTER TABLE Table1 ADD Column2 varbinary(256); UPDATE Table1 SET Column2 = EncryptByKey (Key_GUID('Key1'),Column1); CREATE SYMMETRIC KEY Key1 WITH ALGORITHM = SHA1 ENCRYPTION BY CERTIFICATE Cert1; CREATE SYMMETRIC KEY Key1 WITH ALGORITHM = AES_256 ENCRYPTION BY CERTIFICATE Cert1; CREATE CREDENTIAL Cred1 WITH IDENTITY = 'User1', SECRET = 'P@ssw0rd'; OPEN SYMMETRIC KEY Key1 DECRYPTION BY CERTIFICATE Cert1; CLOSE SYMMETRIC KEY; </pre>
	<div>&lt;&lt; Move</div> <div>Remove &gt;&gt;</div>

**Correct Answer:**

```

ALTER TABLE Table1 ADD Column2
varbinary(256);
OPEN SYMMETRIC KEY Key1 DECRYPTION
BY CERTIFICATE Cert1;
CREATE SYMMETRIC KEY Key1 WITH
ALGORITHM = SHA1 ENCRYPTION BY
CERTIFICATE Cert1;
UPDATE Table1 SET Column2 = EncryptByKey
(Key_GUID('Key1'),Column1);
CLOSE SYMMETRIC KEY;

```

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, the answer looks correct.

References:

<http://www.databasejournal.com/features/mssql/article.php/3922881/Column-Level-Encryption-in-SQL-Server.htm>

<http://msdn.microsoft.com/en-us/library/bb510663.aspx>

<http://msdn.microsoft.com/en-us/library/ms179331.aspx>

<http://msdn.microsoft.com/en-us/library/ms175491.aspx>

<http://msdn.microsoft.com/en-us/library/ms181860.aspx>

<http://msdn.microsoft.com/en-us/library/ms174361.aspx>

<http://msdn.microsoft.com/en-us/library/ms190499.aspx>

<http://msdn.microsoft.com/en-us/library/ms177938.aspx>  
<http://msdn.microsoft.com/en-us/library/ms345262.aspx>  
<http://msdn.microsoft.com/en-us/library/ms188357.aspx>  
<http://msdn.microsoft.com/en-us/library/ms175491.aspx>

#### QUESTION 27

You have a SQL Server 2012 database named Database1. You execute the following code:

```
CREATE TABLE Sales
(
    ID int IDENTITY(1,1) NOT NULL PRIMARY KEY,
    OrderDate char(10) NOT NULL,
    Amount decimal
);
GO

CREATE INDEX IX_Sales_OrderDate
ON Sales(OrderDate)
INCLUDE (ID, Amount);
GO

CREATE PROC usp_Proc1(
    @date1 datetime,
    @date2 datetime
)
AS
SELECT ID, OrderDate, Amount
FROM Sales
WHERE CAST(OrderDate AS datetime)
    BETWEEN @date1 AND @date2
ORDER BY ID;
GO
```

You insert 3 million rows into Sales. You need to reduce the amount of time it takes to execute Proc1. What should you do?

- A. Run the following: `ALTER TABLE Sales ALTER COLUMN OrderDate datetime NOT NULL;`
- B. Change the WHERE clause to the following: `WHERE OrderDate > @date1 AND OrderDate < @date2`
- C. Remove the ORDER BY clause from the stored procedure.
- D. Run the following:  
`DROP INDEX IX_Sales_OrderDate;`  
`GO`  
`CREATE INDEX IX_Sales_OrderDate ON Sales(OrderDate);`  
`GO`

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

I'm not sure about the exact wording of the answers, but I remember that the CAST-BETWEEN part was replaced by > @date1 AND < @date2 and this was the answer that made most sense to me.



## Case Study 1 - Datum

### QUESTION 1

#### Case Study 1: A Datum

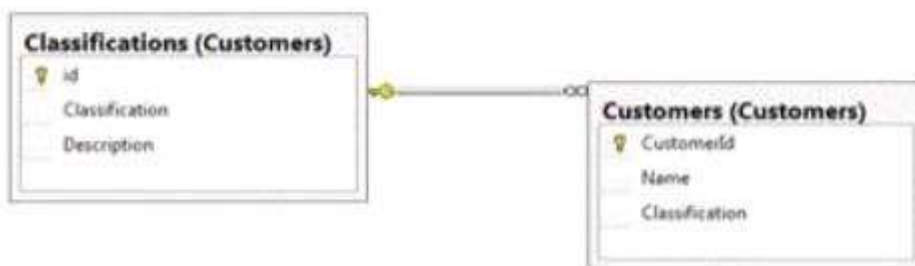
##### Overview

##### General Overview

A. Datum Corporation has offices in Miami and Montreal. The network contains a single Active Directory forest named adatum.com. The offices connect to each other by using a WAN link that has a 5-ms latency. A. Datum standardizes its database platform by using SQL Server 2012 Standard edition.

##### Databases

Each office contains databases named Sales, Inventory, Customers, Products, Personnel, and Dev. Servers and databases are managed by a team of database administrators. Currently, all of the database administrators have the same level of permissions on all of the servers and all of the databases. The Customers database contains two tables named Customers and Classifications. The following graphic shows the relevant portions of the tables:



The following table shows the current data in the Classifications table:

id	Classification	Description
1	Platinum	Yearly sales over 1,000,000
2	Gold	Yearly sales over 500,000
3	Silver	Yearly sales over 100,000

The Inventory database is used mainly for reports. The database is recreated every day. A full backup of the database currently takes three hours to complete.

##### Stored Procedures

A stored procedure named sp1 generates millions of rows of data for multiple reports. Sp1 combines data from five different tables from the Sales and Customers databases in a table named Table1. After Table1 is created, the reporting process reads data from a table in the Products database and searches for information in Table1 based on input from the Products table. After the process is complete, Table1 is deleted. A stored procedure named sp2 is used to generate a product list. Sp2 takes several minutes to run due to locks on the tables the procedure accesses. A stored procedure named sp3 is used to update prices. Sp3 is composed of several UPDATE statements called in sequence from within a transaction. Currently, if one of the UPDATE statements fails, the stored procedure continues to execute. A stored procedure named sp4 calls stored procedures in the Sales, Customers, and Inventory databases. The nested stored procedures read tables from the Sales, Customers, and Inventory databases. Sp4 uses an EXECUTE AS clause. A stored procedure named sp5 changes data in multiple databases. Security checks are performed each time sp5 accesses a database. You suspect that the security checks are slowing down the performance of sp5. All stored procedures accessed by user applications call nested stored procedures. The nested stored procedures are never called directly.

##### Design Requirements

##### Data Recovery

You must be able to recover data from the Inventory database if a storage failure occurs. You have a Recovery Point Objective (RPO) of one hour. You must be able to recover data from the Dev database if data is lost

accidentally. You have a Recovery Point Objective (RPO) of one day.

### Classification Changes

You plan to change the way customers are classified. The new classifications will have four levels based on the number of orders. Classifications may be removed or added in the future. Management requests that historical data be maintained for the previous classifications.

### Security

A group of junior database administrators must be able to view the server state of the SQL Server instance that hosts the Sales database. The junior database administrators will not have any other administrative rights.

### Question

You need to recommend a solution for the planned changes to the customer classifications. What should you recommend? (Each correct answer presents part of the solution. Choose all that apply.)

- A. Add a table to track any changes made to the classification of each customer.
- B. Add columns for each classification to the Customers table.
- C. Implement change data capture.
- D. Add a row to the Customers table each time a classification changes.
- E. Add a column to the Classifications table to track the status of each classification.

**Correct Answer:** AC

**Section:** (none)

**Explanation**

### Explanation/Reference:

According to this reference, this answer looks correct.

Reference:

<http://msdn.microsoft.com/en-us/library/cc645937.aspx>

## QUESTION 2

### Case Study 1: A Datum

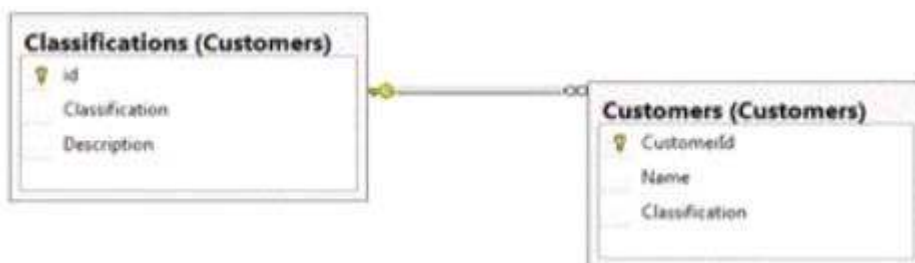
#### Overview

#### General Overview

A. Datum Corporation has offices in Miami and Montreal. The network contains a single Active Directory forest named adatum.com. The offices connect to each other by using a WAN link that has a 5-ms latency. A. Datum standardizes its database platform by using SQL Server 2012 Standard edition.

### Databases

Each office contains databases named Sales, Inventory, Customers, Products, Personnel, and Dev. Servers and databases are managed by a team of database administrators. Currently, all of the database administrators have the same level of permissions on all of the servers and all of the databases. The Customers database contains two tables named Customers and Classifications. The following graphic shows the relevant portions of the tables:



The following table shows the current data in the Classifications table:

id	Classification	Description
1	Platinum	Yearly sales over 1,000,000
2	Gold	Yearly sales over 500,000
3	Silver	Yearly sales over 100,000

The Inventory database is used mainly for reports. The database is recreated every day. A full backup of the database currently takes three hours to complete.

### Stored Procedures

A stored procedure named sp1 generates millions of rows of data for multiple reports. Sp1 combines data from five different tables from the Sales and Customers databases in a table named Table1. After Table1 is created, the reporting process reads data from a table in the Products database and searches for information in Table1 based on input from the Products table. After the process is complete, Table1 is deleted. A stored procedure named sp2 is used to generate a product list. Sp2 takes several minutes to run due to locks on the tables the procedure accesses. A stored procedure named sp3 is used to update prices. Sp3 is composed of several UPDATE statements called in sequence from within a transaction. Currently, if one of the UPDATE statements fails, the stored procedure continues to execute. A stored procedure named sp4 calls stored procedures in the Sales, Customers, and Inventory databases. The nested stored procedures read tables from the Sales, Customers, and Inventory databases. Sp4 uses an EXECUTE AS clause. A stored procedure named sp5 changes data in multiple databases. Security checks are performed each time sp5 accesses a database. You suspect that the security checks are slowing down the performance of sp5. All stored procedures accessed by user applications call nested stored procedures. The nested stored procedures are never called directly.

### Design Requirements

#### Data Recovery

You must be able to recover data from the Inventory database if a storage failure occurs. You have a Recovery Point Objective (RPO) of one hour. You must be able to recover data from the Dev database if data is lost accidentally. You have a Recovery Point Objective (RPO) of one day.

#### Classification Changes

You plan to change the way customers are classified. The new classifications will have four levels based on the number of orders. Classifications may be removed or added in the future. Management requests that historical data be maintained for the previous classifications.

### Security

A group of junior database administrators must be able to view the server state of the SQL Server instance that hosts the Sales database. The junior database administrators will not have any other administrative rights.

-----

#### Question

You need to recommend a solution to meet the security requirements of the junior database administrators. What should you include in the recommendation?

- A. a shared login
- B. a database role
- C. a credential
- D. a server role

**Correct Answer: D**

**Section: (none)**

**Explanation**

#### Explanation/Reference:

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms188659.aspx>  
<http://msdn.microsoft.com/en-us/library/ms189121.aspx>  
<http://msdn.microsoft.com/en-us/library/ms161950.aspx>  
<http://msdn.microsoft.com/en-us/library/ms188642.aspx>  
<http://msdn.microsoft.com/en-us/library/aa337562.aspx>

### QUESTION 3

#### Case Study 1: A Datum

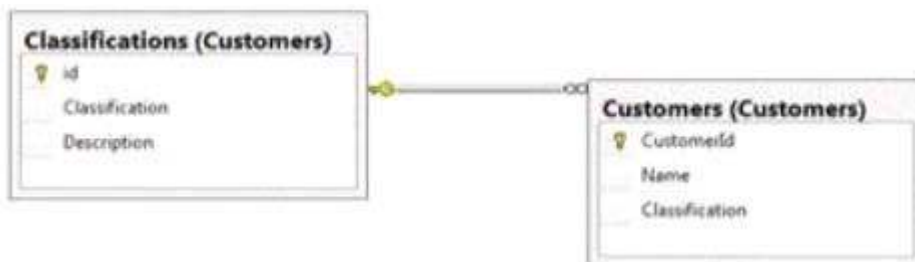
##### Overview

##### General Overview

A. Datum Corporation has offices in Miami and Montreal. The network contains a single Active Directory forest named adatum.com. The offices connect to each other by using a WAN link that has a 5-ms latency. A. Datum standardizes its database platform by using SQL Server 2012 Standard edition.

##### Databases

Each office contains databases named Sales, Inventory, Customers, Products, Personnel, and Dev. Servers and databases are managed by a team of database administrators. Currently, all of the database administrators have the same level of permissions on all of the servers and all of the databases. The Customers database contains two tables named Customers and Classifications. The following graphic shows the relevant portions of the tables:



The following table shows the current data in the Classifications table:

id	Classification	Description
1	Platinum	Yearly sales over 1,000,000
2	Gold	Yearly sales over 500,000
3	Silver	Yearly sales over 100,000

The Inventory database is used mainly for reports. The database is recreated every day. A full backup of the database currently takes three hours to complete.

##### Stored Procedures

A stored procedure named sp1 generates millions of rows of data for multiple reports. Sp1 combines data from five different tables from the Sales and Customers databases in a table named Table1. After Table1 is created, the reporting process reads data from a table in the Products database and searches for information in Table1 based on input from the Products table. After the process is complete, Table1 is deleted. A stored procedure named sp2 is used to generate a product list. Sp2 takes several minutes to run due to locks on the tables the procedure accesses. A stored procedure named sp3 is used to update prices. Sp3 is composed of several UPDATE statements called in sequence from within a transaction. Currently, if one of the UPDATE statements fails, the stored procedure continues to execute. A stored procedure named sp4 calls stored procedures in the Sales, Customers, and Inventory databases. The nested stored procedures read tables from the Sales, Customers, and Inventory databases. Sp4 uses an EXECUTE AS clause. A stored procedure named sp5 changes data in multiple databases. Security checks are performed each time sp5 accesses a database. You suspect that the security checks are slowing down the performance of sp5. All stored procedures accessed by user applications call nested stored procedures. The nested stored procedures are never called directly.

## Design Requirements

### Data Recovery

You must be able to recover data from the Inventory database if a storage failure occurs. You have a Recovery Point Objective (RPO) of one hour. You must be able to recover data from the Dev database if data is lost accidentally. You have a Recovery Point Objective (RPO) of one day.

### Classification Changes

You plan to change the way customers are classified. The new classifications will have four levels based on the number of orders. Classifications may be removed or added in the future. Management requests that historical data be maintained for the previous classifications.

### Security

A group of junior database administrators must be able to view the server state of the SQL Server instance that hosts the Sales database. The junior database administrators will not have any other administrative rights.

### Question

You need to recommend a disaster recovery strategy for the Inventory database. What should you include in the recommendation?

- A. Log shipping
- B. AlwaysOn Availability Groups
- C. SQL Server Failover Clustering
- D. Peer-to-peer replication

**Correct Answer:** A

**Section:** (none)

**Explanation**

### Explanation/Reference:

-- BURGOS

LogShipping is available in SQL Standard Edition and may be used for reporting purposes. Inventory have NO changes during the day, so, Log Shipping won't cause significant traffic.

AlwaysOn is wrong because is unavailable on SQL Standard Edition.

FailOver would need a SAN.

--\BURGOS.

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/cc645993.aspx>

<http://msdn.microsoft.com/en-us/library/ms187103.aspx>

<http://msdn.microsoft.com/en-us/library/ms190640.aspx>

## QUESTION 4

### Case Study 1: A Datum

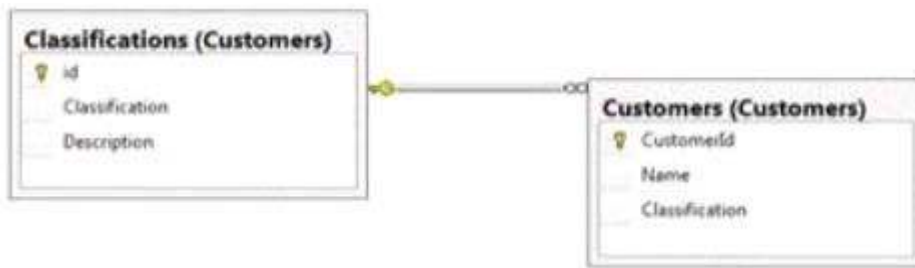
#### Overview

#### General Overview

A. Datum Corporation has offices in Miami and Montreal. The network contains a single Active Directory forest named adatum.com. The offices connect to each other by using a WAN link that has a 5-ms latency. A. Datum standardizes its database platform by using SQL Server 2012 Standard edition.

#### Databases

Each office contains databases named Sales, Inventory, Customers, Products, Personnel, and Dev. Servers and databases are managed by a team of database administrators. Currently, all of the database administrators have the same level of permissions on all of the servers and all of the databases. The Customers database contains two tables named Customers and Classifications. The following graphic shows the relevant portions of the tables:



The following table shows the current data in the Classifications table:

id	Classification	Description
1	Platinum	Yearly sales over 1,000,000
2	Gold	Yearly sales over 500,000
3	Silver	Yearly sales over 100,000

The Inventory database is used mainly for reports. The database is recreated every day. A full backup of the database currently takes three hours to complete.

### Stored Procedures

A stored procedure named sp1 generates millions of rows of data for multiple reports. Sp1 combines data from five different tables from the Sales and Customers databases in a table named Table1. After Table1 is created, the reporting process reads data from a table in the Products database and searches for information in Table1 based on input from the Products table. After the process is complete, Table1 is deleted. A stored procedure named sp2 is used to generate a product list. Sp2 takes several minutes to run due to locks on the tables the procedure accesses. A stored procedure named sp3 is used to update prices. Sp3 is composed of several UPDATE statements called in sequence from within a transaction. Currently, if one of the UPDATE statements fails, the stored procedure continues to execute. A stored procedure named sp4 calls stored procedures in the Sales, Customers, and Inventory databases. The nested stored procedures read tables from the Sales, Customers, and Inventory databases. Sp4 uses an EXECUTE AS clause. A stored procedure named sp5 changes data in multiple databases. Security checks are performed each time sp5 accesses a database. You suspect that the security checks are slowing down the performance of sp5. All stored procedures accessed by user applications call nested stored procedures. The nested stored procedures are never called directly.

### Design Requirements

#### Data Recovery

You must be able to recover data from the Inventory database if a storage failure occurs. You have a Recovery Point Objective (RPO) of one hour. You must be able to recover data from the Dev database if data is lost accidentally. You have a Recovery Point Objective (RPO) of one day.

#### Classification Changes

You plan to change the way customers are classified. The new classifications will have four levels based on the number of orders. Classifications may be removed or added in the future. Management requests that historical data be maintained for the previous classifications.

#### Security

A group of junior database administrators must be able to view the server state of the SQL Server instance that hosts the Sales database. The junior database administrators will not have any other administrative rights.

#### Question

You need to recommend a solution to ensure that sp4 adheres to the security requirements. What should you include in the recommendation?

A. Configure data manipulation language (DML) triggers.

- B. Enable SQL Server Audit.
- C. Enable trace flags.
- D. Enable C2 audit tracing.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms178110.aspx>

<http://msdn.microsoft.com/en-us/library/cc280386.aspx>

<http://msdn.microsoft.com/en-us/library/ms188396.aspx>

<http://msdn.microsoft.com/en-us/library/ms187634.aspx>

## QUESTION 5

### Case Study 1: A Datum

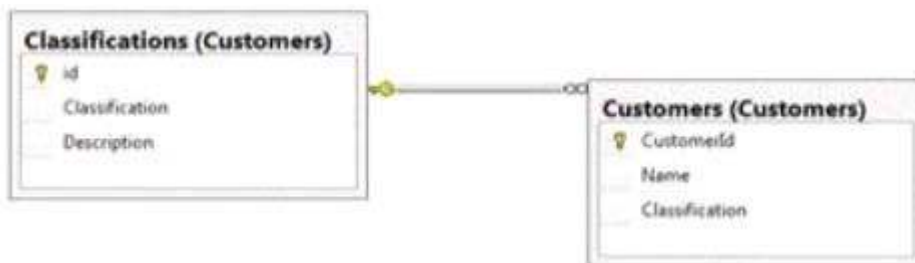
#### Overview

#### General Overview

A. Datum Corporation has offices in Miami and Montreal. The network contains a single Active Directory forest named adatum.com. The offices connect to each other by using a WAN link that has a 5-ms latency. A. Datum standardizes its database platform by using SQL Server 2012 Standard edition.

#### Databases

Each office contains databases named Sales, Inventory, Customers, Products, Personnel, and Dev. Servers and databases are managed by a team of database administrators. Currently, all of the database administrators have the same level of permissions on all of the servers and all of the databases. The Customers database contains two tables named Customers and Classifications. The following graphic shows the relevant portions of the tables:



The following table shows the current data in the Classifications table:

id	Classification	Description
1	Platinum	Yearly sales over 1,000,000
2	Gold	Yearly sales over 500,000
3	Silver	Yearly sales over 100,000

The Inventory database is used mainly for reports. The database is recreated every day. A full backup of the database currently takes three hours to complete.

#### Stored Procedures

A stored procedure named sp1 generates millions of rows of data for multiple reports. Sp1 combines data from five different tables from the Sales and Customers databases in a table named Table1. After Table1 is created,

the reporting process reads data from a table in the Products database and searches for information in Table1 based on input from the Products table. After the process is complete, Table1 is deleted. A stored procedure named sp2 is used to generate a product list. Sp2 takes several minutes to run due to locks on the tables the procedure accesses. A stored procedure named sp3 is used to update prices. Sp3 is composed of several UPDATE statements called in sequence from within a transaction. Currently, if one of the UPDATE statements fails, the stored procedure continues to execute. A stored procedure named sp4 calls stored procedures in the Sales, Customers, and Inventory databases. The nested stored procedures read tables from the Sales, Customers, and Inventory databases. Sp4 uses an EXECUTE AS clause. A stored procedure named sp5 changes data in multiple databases. Security checks are performed each time sp5 accesses a database. You suspect that the security checks are slowing down the performance of sp5. All stored procedures accessed by user applications call nested stored procedures. The nested stored procedures are never called directly.

## **Design Requirements**

### **Data Recovery**

You must be able to recover data from the Inventory database if a storage failure occurs. You have a Recovery Point Objective (RPO) of one hour. You must be able to recover data from the Dev database if data is lost accidentally. You have a Recovery Point Objective (RPO) of one day.

### **Classification Changes**

You plan to change the way customers are classified. The new classifications will have four levels based on the number of orders. Classifications may be removed or added in the future. Management requests that historical data be maintained for the previous classifications.

### **Security**

A group of junior database administrators must be able to view the server state of the SQL Server instance that hosts the Sales database. The junior database administrators will not have any other administrative rights.

### **Question**

You need to recommend a change to sp3 to ensure that the procedure continues to execute even if one of the UPDATE statements fails. Which change should you recommend?

- A. Set the IMPLICIT\_TRANSACTIONS option to on.
- B. Set the XACT\_ABORT option to off.
- C. Set the IMPLICIT\_TRANSACTIONS option to off.
- D. Set the XACT\_ABORT option to on.

**Correct Answer: B**

**Section: (none)**

**Explanation**

### **Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms188792.aspx>

<http://msdn.microsoft.com/en-us/library/ms188317.aspx>

<http://msdn.microsoft.com/en-us/library/ms187807.aspx>



## Case Study 2 - Contoso

### QUESTION 1

#### Case Study 2: 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 2012. 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. The previous version of the ERP application relied on table-level security.

##### 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

You need to recommend a solution that addresses the concurrency requirement. What should you recommend?

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

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

#### References:

<http://msdn.microsoft.com/en-us/library/ms191242%28v=SQL.105%29.aspx>

<http://msdn.microsoft.com/en-us/library/bb677357.aspx>

<http://msdn.microsoft.com/en-us/library/ms378149.aspx>

## QUESTION 2

### Case Study 2: 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 2012. 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. The previous version of the ERP application relied on table-level security.

#### 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

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. filegroups
- B. indexed views
- C. table partitioning
- D. indexes

**Correct Answer: A**

**Section: (none)****Explanation****Explanation/Reference:**

According to these references, this answer looks correct.

**References:**

<http://msdn.microsoft.com/en-us/library/ms187048.aspx>

<http://msdn.microsoft.com/en-us/library/ms189563.aspx>

<http://msdn.microsoft.com/en-us/library/ms190174.aspx>

<http://msdn.microsoft.com/en-us/library/ms190787.aspx>

<http://msdn.microsoft.com/en-us/library/ms175049.aspx>

**QUESTION 3****Case Study 2: 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 2012. 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. The previous version of the ERP application relied on table-level security.

**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

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

**Correct Answer:** BD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

-- BURGOS

According previous answer, "C" is correct too. I AGREE, if you change datatype from uniqueidentifier to bigint you will decrease the key length and the fragmentation, but, you CAN do that? Would be better to drop the index, or drop database, but, you wouldn't to do that. So, I think that this column would be with original datatype.

--\BURGOS

According to these references, this answer looks correct.

References:

<http://technet.microsoft.com/en-us/library/ms190639.aspx>

<http://msdn.microsoft.com/en-us/library/ms190457.aspx>

<http://msdn.microsoft.com/en-us/library/ms186342.aspx>

<http://msdn.microsoft.com/en-us/library/ms189280.aspx>

<http://stackoverflow.com/questions/255569/sql-data-type-for-primary-key-sql-server>

<http://sqlservernet.blogspot.com/2012/01/which-data-type-is-good-for-primary-key.html>

<http://msdn.microsoft.com/en-us/library/jj591574.aspx>

<http://www.informit.com/articles/article.aspx?p=25862&seqNum=5>

## QUESTION 4

### Case Study 2: 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 2012. 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. The previous version of the ERP application relied on table-level security.

#### 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**

You need to recommend a solution that meets the data recovery requirement. What should you include in the recommendation?

- A. a differential backup
- B. snapshot isolation
- C. a transaction log backup
- D. a database snapshot

**Correct Answer: D**

**Section: (none)**

**Explanation**

### **Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ms175158.aspx>

<http://msdn.microsoft.com/en-us/library/ms378149.aspx>

<http://msdn.microsoft.com/en-us/library/ms187048.aspx>

## **QUESTION 5**

### **Case Study 2: 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 2012. 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. The previous version of the ERP application relied on table-level security.

## 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**

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

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ff848763.aspx>

<http://msdn.microsoft.com/en-us/library/ms143726.aspx>

<http://msdn.microsoft.com/en-us/library/ms190920.aspx>

## Case Study 3 - Litware

### QUESTION 1

#### Case Study 3: Litware, Inc

##### Overview

You are a database administrator for a company named Litware, Inc. Litware is a book publishing house. Litware has a main office and a branch office. You are designing the database infrastructure to support a new web-based application that is being developed. The web application will be accessed at [www.litwareinc.com](http://www.litwareinc.com). Both internal employees and external partners will use the application. You have an existing desktop application that uses a SQL Server 2005 database named App1\_DB. App1\_DB will remain in production.

##### Requirements

##### Planned Changes

You plan to deploy a SQL Server 2012 instance that will contain two databases named Database1 and Database2. All database files will be stored in a highly available SAN. Database1 will contain two tables named Orders and OrderDetails. Database1 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. Database1 will contain several queries that access data in the Database2 tables. 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. 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. All data in Database2 is recreated each day and does not change until the next data creation process. 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. Litware plans to use offsite storage for all SQL Server 2012 backups.

##### Business Requirements

You have the following requirements:

- Costs for new licenses must be minimized.
- Private information that is accessed by Application1 must be stored in a secure format.
- Development effort must be minimized whenever possible.
- The storage requirements for databases must be minimized.
- System administrators must be able to run real-time reports on disk usage.
- The databases must be available if the SQL Server service fails.
- Database administrators must receive a detailed report that contains allocation errors and data corruption.
- Application developers must be denied direct access to the database tables.
- Applications must be denied direct access to the tables.
- You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

##### Question

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

- A. repeatable read
- B. serializable
- C. read uncommitted
- D. read committed

**Correct Answer: A**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

#### References:

<http://msdn.microsoft.com/en-us/library/ms378149.aspx>

<http://msdn.microsoft.com/en-us/library/ms173763.aspx>

## QUESTION 2

### Case Study 3: Litware, Inc

#### Overview

You are a database administrator for a company named Litware, Inc. Litware is a book publishing house. Litware has a main office and a branch office. You are designing the database infrastructure to support a new web-based application that is being developed. The web application will be accessed at [www.litwareinc.com](http://www.litwareinc.com). Both internal employees and external partners will use the application. You have an existing desktop application that uses a SQL Server 2005 database named App1\_DB. App1\_DB will remain in production.

#### Requirements

##### Planned Changes

You plan to deploy a SQL Server 2012 instance that will contain two databases named Database1 and Database2. All database files will be stored in a highly available SAN. Database1 will contain two tables named Orders and OrderDetails. Database1 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. Database1 will contain several queries that access data in the Database2 tables. 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. 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. All data in Database2 is recreated each day and does not change until the next data creation process. 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. Litware plans to use offsite storage for all SQL Server 2012 backups.

#### Business Requirements

You have the following requirements:

- Costs for new licenses must be minimized.
- Private information that is accessed by Application1 must be stored in a secure format.
- Development effort must be minimized whenever possible.
- The storage requirements for databases must be minimized.
- System administrators must be able to run real-time reports on disk usage.
- The databases must be available if the SQL Server service fails.
- Database administrators must receive a detailed report that contains allocation errors and data corruption.
- Application developers must be denied direct access to the database tables.
- Applications must be denied direct access to the tables.
- You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

#### Question

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 subquery
- C. a common table expression
- D. a cursor

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

Explanation missing!

### QUESTION 3

#### Case Study 3: Litware, Inc

##### Overview

You are a database administrator for a company named Litware, Inc. Litware is a book publishing house. Litware has a main office and a branch office. You are designing the database infrastructure to support a new web-based application that is being developed. The web application will be accessed at [www.litwareinc.com](http://www.litwareinc.com). Both internal employees and external partners will use the application. You have an existing desktop application that uses a SQL Server 2005 database named App1\_DB. App1\_DB will remain in production.

##### Requirements

##### Planned Changes

You plan to deploy a SQL Server 2012 instance that will contain two databases named Database1 and Database2. All database files will be stored in a highly available SAN. Database1 will contain two tables named Orders and OrderDetails. Database1 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. Database1 will contain several queries that access data in the Database2 tables. 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. 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. All data in Database2 is recreated each day and does not change until the next data creation process. 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. Litware plans to use offsite storage for all SQL Server 2012 backups.

##### Business Requirements

You have the following requirements:

- Costs for new licenses must be minimized.
- Private information that is accessed by Application1 must be stored in a secure format.
- Development effort must be minimized whenever possible.
- The storage requirements for databases must be minimized.
- System administrators must be able to run real-time reports on disk usage.
- The databases must be available if the SQL Server service fails.
- Database administrators must receive a detailed report that contains allocation errors and data corruption.
- Application developers must be denied direct access to the database tables.
- Applications must be denied direct access to the tables.
- You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

---

##### Question

You need to recommend a solution to allow application users to perform UPDATE operations on the database tables. The solution must meet the business requirements. What should you recommend?

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

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

#### References:

<http://msdn.microsoft.com/en-us/library/ms188354.aspx>  
<http://msdn.microsoft.com/en-us/library/ms189121.aspx>  
<http://msdn.microsoft.com/en-us/library/ms131287.aspx>  
<http://msdn.microsoft.com/en-us/library/ms186755.aspx>  
<http://msdn.microsoft.com/en-us/library/ms191320.aspx>  
<http://msdn.microsoft.com/en-us/library/bb510667.aspx>

## QUESTION 4

### Case Study 3: Litware, Inc

#### Overview

You are a database administrator for a company named Litware, Inc. Litware is a book publishing house. Litware has a main office and a branch office. You are designing the database infrastructure to support a new web-based application that is being developed. The web application will be accessed at [www.litwareinc.com](http://www.litwareinc.com). Both internal employees and external partners will use the application. You have an existing desktop application that uses a SQL Server 2005 database named App1\_DB. App1\_DB will remain in production.

#### Requirements

##### Planned Changes

You plan to deploy a SQL Server 2012 instance that will contain two databases named Database1 and Database2. All database files will be stored in a highly available SAN. Database1 will contain two tables named Orders and OrderDetails. Database1 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. Database1 will contain several queries that access data in the Database2 tables. 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. 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. All data in Database2 is recreated each day and does not change until the next data creation process. 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. Litware plans to use offsite storage for all SQL Server 2012 backups.

#### Business Requirements

You have the following requirements:

- Costs for new licenses must be minimized.
- Private information that is accessed by Application1 must be stored in a secure format.
- Development effort must be minimized whenever possible.
- The storage requirements for databases must be minimized.
- System administrators must be able to run real-time reports on disk usage.
- The databases must be available if the SQL Server service fails.
- Database administrators must receive a detailed report that contains allocation errors and data corruption.
- Application developers must be denied direct access to the database tables.
- Applications must be denied direct access to the tables.
- You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

#### Question

You need to recommend a solution for the deployment of SQL Server 2012. The solution must meet the business requirements. What should you include in the recommendation?

- A. Deploy two servers that have SQL Server 2012 installed. Implement AlwaysOn Availability Groups on both servers.
- B. Upgrade the existing SQL Server 2005 instance to SQL Server 2012. Deploy a new server that has SQL Server 2012 installed. Implement AlwaysOn.
- C. Install a new instance of SQL Server 2012 on the server that hosts the SQL Server 2005 instance. Deploy a new server that has SQL Server 2012 installed. Implement AlwaysOn.

D. Deploy two servers that have SQL Server 2012 installed and implement Failover Clustering.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/bb677622.aspx>

<http://msdn.microsoft.com/en-us/library/ff877884.aspx>

## QUESTION 5

### Case Study 3: Litware, Inc

#### Overview

You are a database administrator for a company named Litware, Inc. Litware is a book publishing house. Litware has a main office and a branch office. You are designing the database infrastructure to support a new web-based application that is being developed. The web application will be accessed at [www.litwareinc.com](http://www.litwareinc.com). Both internal employees and external partners will use the application. You have an existing desktop application that uses a SQL Server 2005 database named App1\_DB. App1\_DB will remain in production.

#### Requirements

##### Planned Changes

You plan to deploy a SQL Server 2012 instance that will contain two databases named Database1 and Database2. All database files will be stored in a highly available SAN. Database1 will contain two tables named Orders and OrderDetails. Database1 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. Database1 will contain several queries that access data in the Database2 tables. 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. 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. All data in Database2 is recreated each day and does not change until the next data creation process. 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. Litware plans to use offsite storage for all SQL Server 2012 backups.

#### Business Requirements

You have the following requirements:

- Costs for new licenses must be minimized.
- Private information that is accessed by Application1 must be stored in a secure format.
- Development effort must be minimized whenever possible.
- The storage requirements for databases must be minimized.
- System administrators must be able to run real-time reports on disk usage.
- The databases must be available if the SQL Server service fails.
- Database administrators must receive a detailed report that contains allocation errors and data corruption.
- Application developers must be denied direct access to the database tables.
- Applications must be denied direct access to the tables.
- You must encrypt the backup files to meet regulatory compliance requirements. The encryption strategy must minimize changes to the databases and to the applications.

#### Question

You need to recommend a solution to synchronize Database2 to App1\_Db1. What should you recommend?

- A. Change data capture
- B. Snapshot replication



- C. Transactional replication
- D. Master Data Services

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://msdn.microsoft.com/en-us/library/ee633752.aspx>

<http://msdn.microsoft.com/en-us/library/ms151198.aspx>

<http://msdn.microsoft.com/en-us/library/cc645937.aspx>

## Case Study 4 - APP

### QUESTION 1

#### Case Study 4: Application Scenario

##### 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

```

#### Question

You need to provide referential integrity between the Offices table and Employees table. Which code segment or segments should you add at line 28 of Tables.sql? (Each correct answer presents part of the solution. Choose all that apply.)

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

**Correct Answer:** AC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to this reference, this answer looks correct.

## References:

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

## QUESTION 2

### Case Study 4: Application Scenario

#### 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

```

#### Question

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

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

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

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

### **QUESTION 3**

#### **Case Study 4: Application Scenario**

##### **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

```

#### Question

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. an OFFSET-FETCH clause
- B. a recursive common table expression
- C. a table variable
- D. the ROWNUMBER keyword

**Correct Answer: A**

**Section: (none)**

**Explanation**

#### Explanation/Reference:

According to these references, this answer looks correct.

References:

<http://www.mssqltips.com/sqlservertip/2696/comparing-performance-for-different-sql-server-paging-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-by-clause/>-  
[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/)

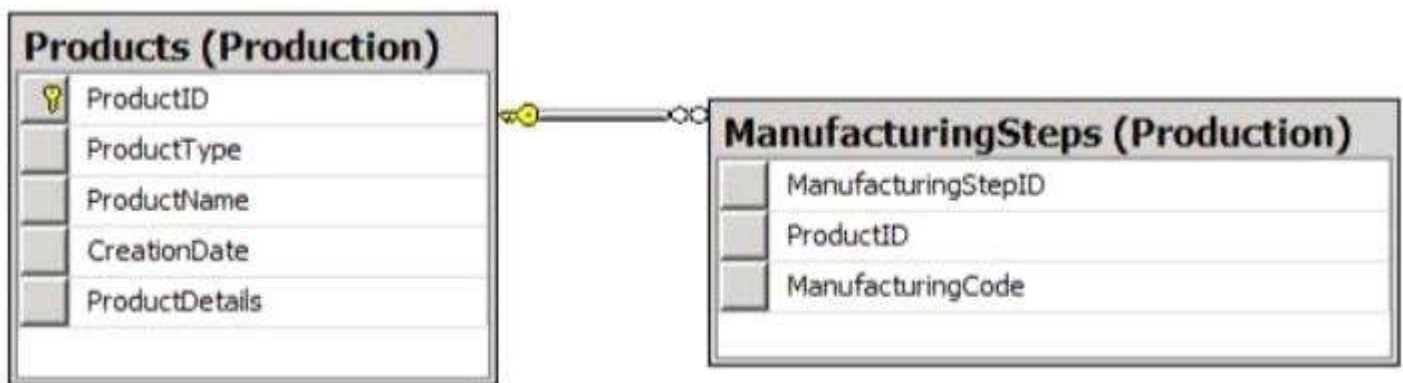
## Case Study 5 - Manufacturing

### QUESTION 1

#### Case Study 5: Manufacturing Company

##### 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.es. 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 ProductID.

##### 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';

```

#### CategoryFromType.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

You need to modify Production.ProductDetails\_Insert to comply with the application requirements. Which code segment should you execute?

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



```
D. OPEN PRODUCTSCERT;
   ALTER PROCEDURE Production.ProductDetails_Insert
   WITH ENCRYPTION;
   CLOSE PRODUCTSCERT;
```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to this reference, this answer looks correct.

References:

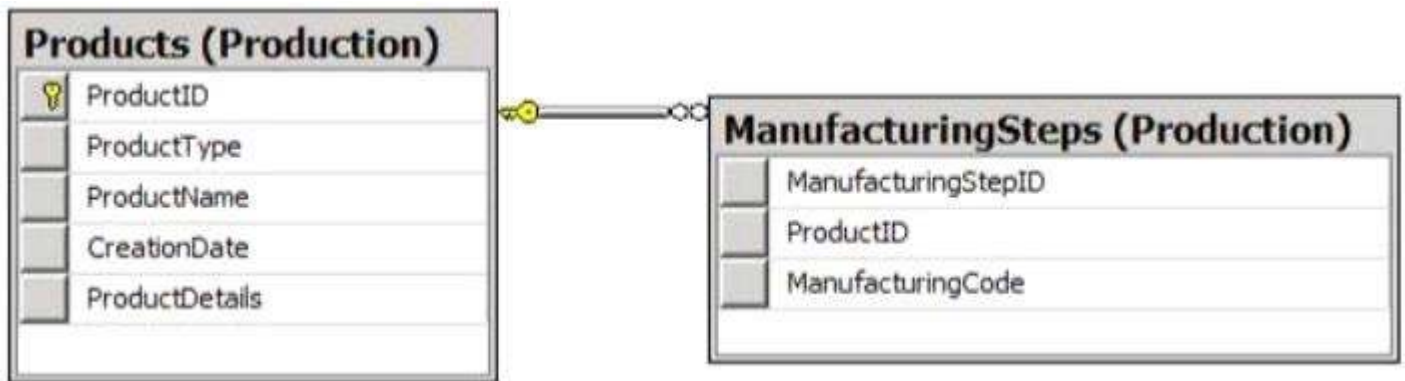
<http://msdn.microsoft.com/en-us/library/bb669102.aspx>

## QUESTION 2

### Case Study 5: Manufacturing Company

#### 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.es. 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 ProductID.

#### 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';

```

#### CategoryFromType.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

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 ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.DLL';
- B. EXEC sp\_recompile @objname = 'Production.ProcessProduct';
- C. RECONFIGURE;
- D. EXEC sp\_configure 'clr enabled', '1';
- E. CREATE ASSEMBLY ProductionAssembly FROM 'C:\Products\ProcessProducts.cs';
- F. CREATE PROCEDURE Production.ProcessProduct (
 @ProductID int, @ProductType varchar(11)

```

)
AS EXTERNAL NAME ProductionAssembly.ProcessProduct.Process;
G. CREATE TYPE Production.ProcessProduct
EXTERNAL NAME ProductionAssembly.ProcessProducts.Process;

```

**Correct Answer:** ACDG

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to the reference, this answer looks correct.

Reference:

<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>

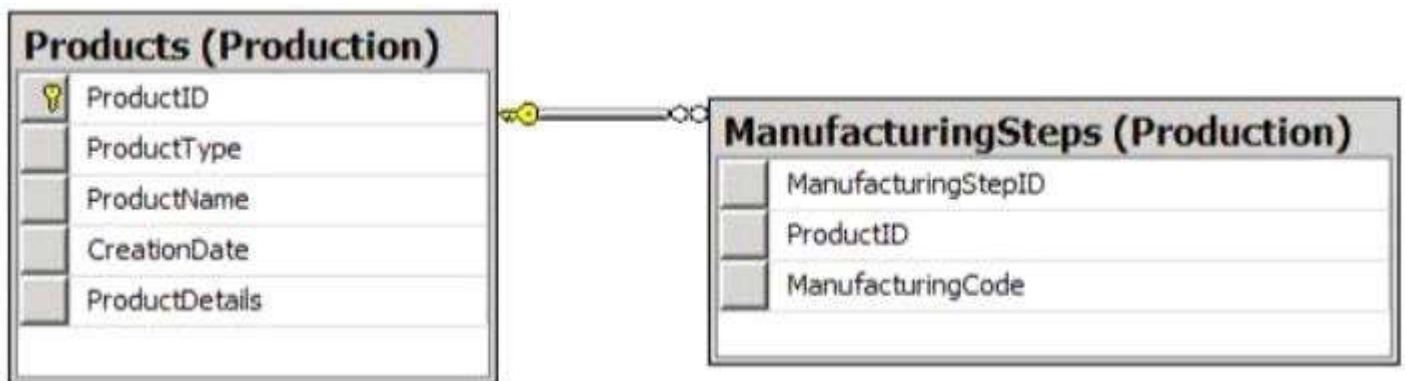
<http://msdn.microsoft.com/en-us/library/ms131107.aspx>

### QUESTION 3

#### Case Study 5: Manufacturing Company

##### 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.es. 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 ProductID.

#### 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';

```

#### CategoryFromType.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

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
```

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to this reference, this answer looks correct.

Reference:

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

## **Case Study 6 - DB APP**

### **QUESTION 1**

#### **Case Study 6: Database Application Scenario**

##### **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
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

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.Speakers ADD CONSTRAINT  
FK\_Speakers\_Sessions FOREIGN KEY (SessionID)  
REFERENCES dbo.Sessions (SessionID);
- C. ALTER TABLE dbo.Sessions ADD CONSTRAINT  
FK\_Sessions\_Speakers FOREIGN KEY (SpeakerID)  
REFERENCES dbo.Speakers (SpeakerID);
- D. ALTER TABLE dbo.Speakers ADD CONSTRAINT  
FK\_Speakers\_Sessions FOREIGN KEY (SpeakerID)  
REFERENCES dbo.Sessions (SessionID);

**Correct Answer: C**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<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 2**

### **Case Study 6: Database Application Scenario**

#### **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
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

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

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<http://www.mssqltips.com/sqlservertip/2696/comparing-performance-for-different-sql-server-paging-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-by-clause/>

[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 3**

#### **Case Study 6: Database Application Scenario**

##### **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
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

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

**Correct Answer: B**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to this reference, this answer looks correct.

References:

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

## Case Study 7 - Invoice

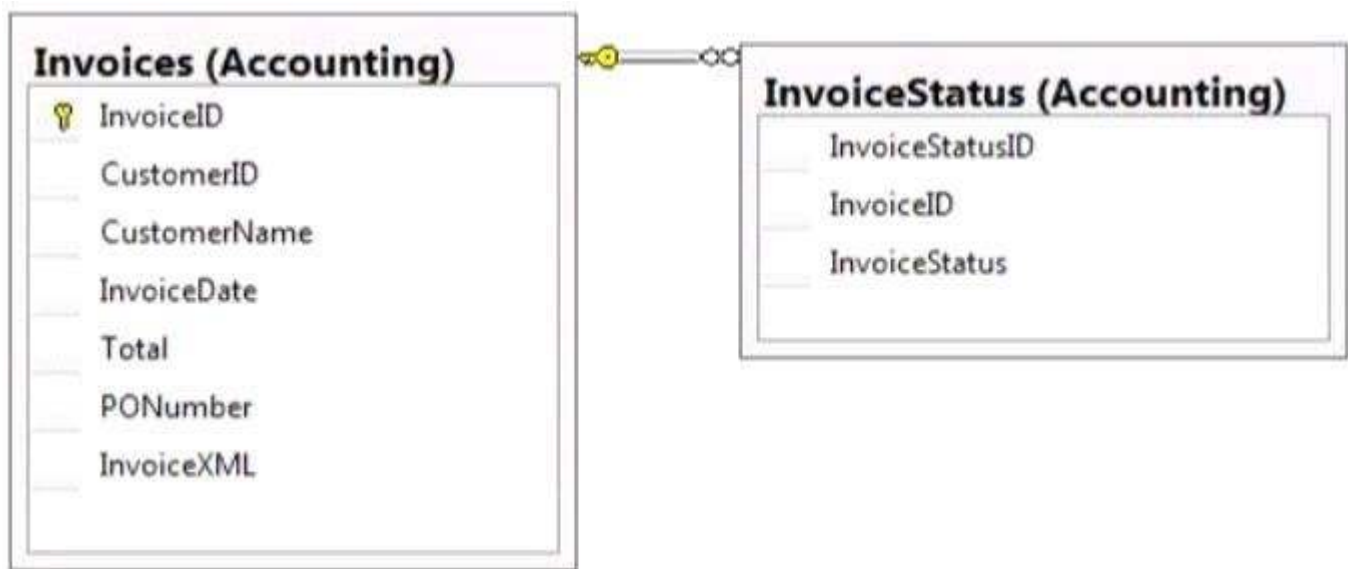
### QUESTION 1

#### Case Study 7: Invoice Schema Scenario

##### 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 @Country;
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

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. 06
- C. 19
- D. 05

**Correct Answer: C**  
**Section: (none)**

## Explanation

### Explanation/Reference:

According to these references, this answer looks correct.

References:

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

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

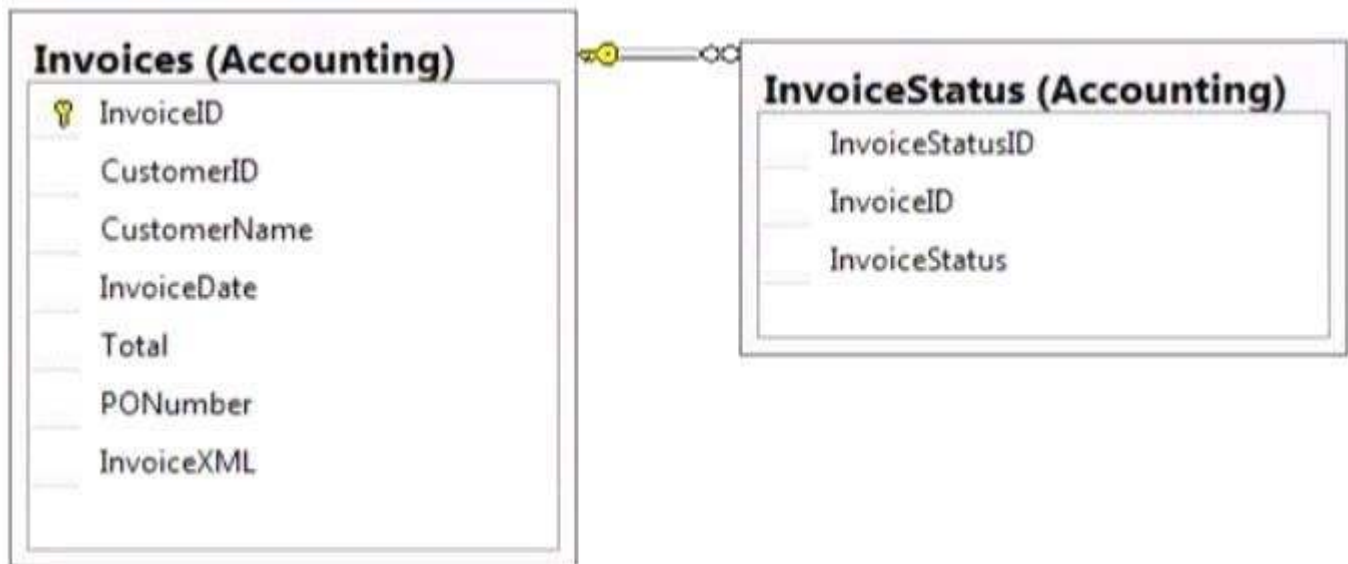
## QUESTION 2

### Case Study 7: Invoice Schema Scenario

#### 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

Which data type should you use for CustomerID?

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

**Correct Answer:** B

**Section:** (none)

**Explanation**

### Explanation/Reference:

#### 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)

According to these references, this answer looks correct.

#### References:

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

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

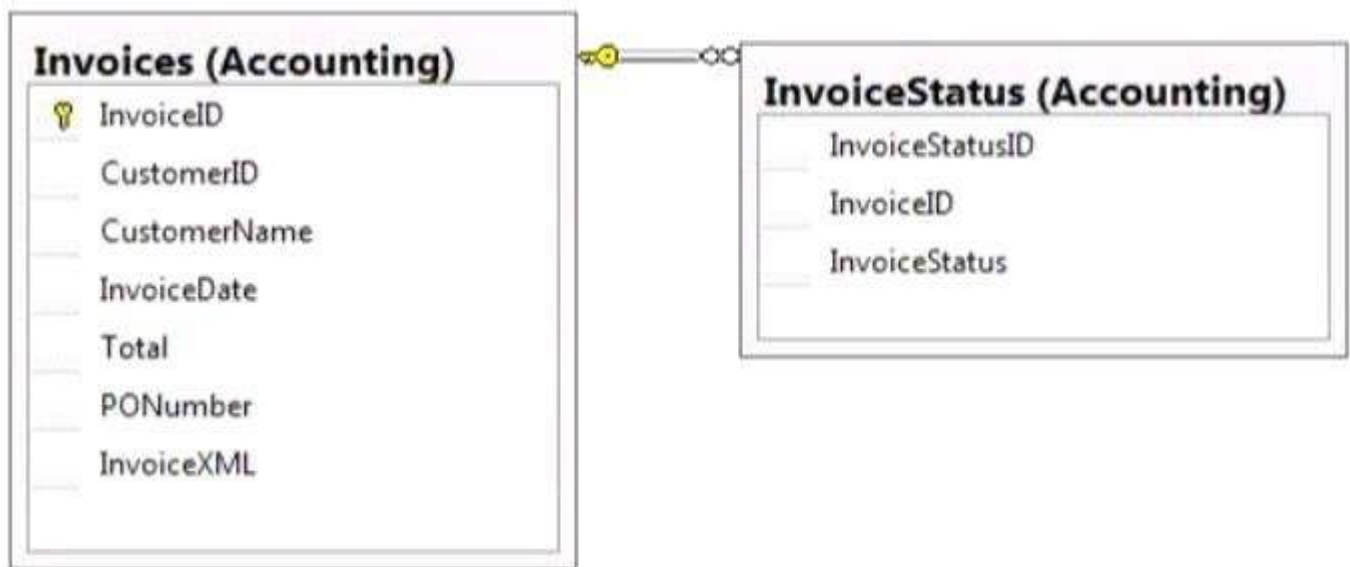
### QUESTION 3

#### Case Study 7: Invoice Schema Scenario

##### 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

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),
 OUTPUT @CustomerID AS char(11),
 OUTPUT @Total AS decimal(8,2))
 AS
 ...
- B. CREATE PROC usp\_InvoicesByCustomerAboveTotal (
 @sqlstring AS nvarchar(1000),

```

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

```

**Correct Answer: D**

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<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 4**

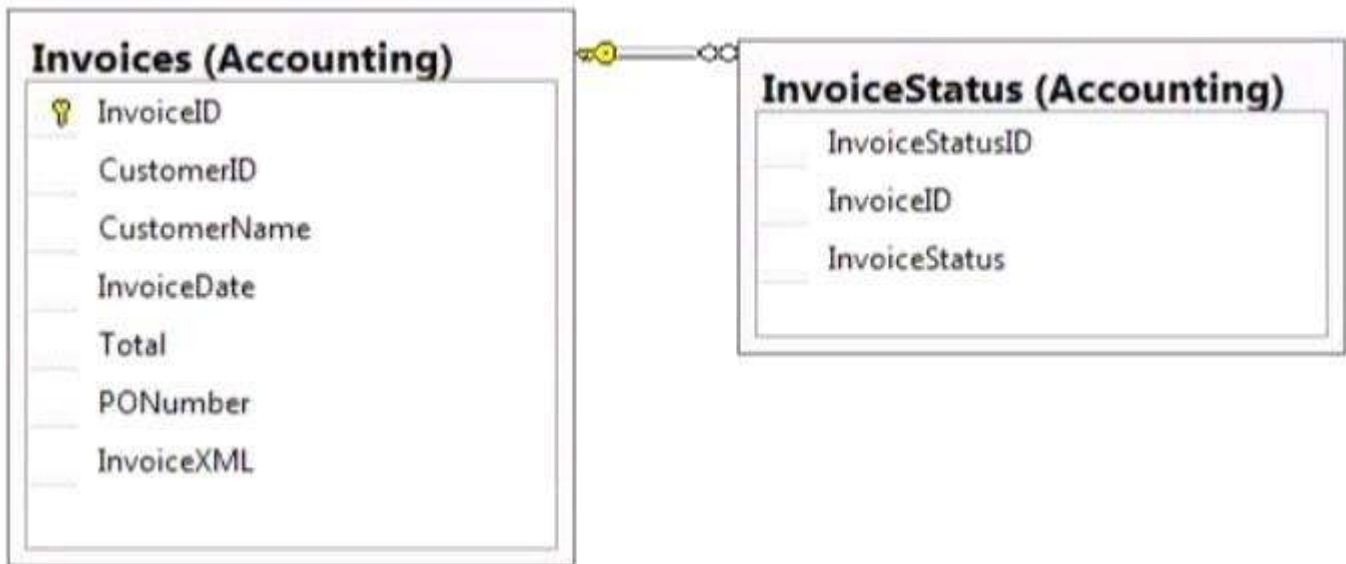
##### **Case Study 7: Invoice Schema Scenario**

##### **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 @Country;
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

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.

**Build List and Reorder:**

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_recompile @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>&lt;&lt; Move</div> <div>Remove &gt;&gt;</div>

**Correct 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;

```

**Section: (none)**

**Explanation**

**Explanation/Reference:**

According to these references, this answer looks correct.

References:

<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>