



Microsoft

70-464 Exam

Microsoft Developing Microsoft SQL Server Databases Exam

Thank you for Downloading 70-464 exam PDF Demo

You can Buy Latest 70-464 Full Version Download

<https://www.certkillers.net/Exam/70-464>

<https://www.certkillers.net>

Version: 18.0

Question: 1

You are testing disaster recovery procedures.

You attempt to restore DB1 to a different server and you receive the following error message: "Msg 33111.

Level 16, State 3, Line 1

Cannot find server certificate with thumbprint

,0xA694FBEA88C9354E5E2567C30A2A69E8FB4C44A9\

Msg 3013, Level 16, State 1, Line 1

RESTORE DATABASE is terminating abnormally."

You need to ensure that you can restore DB1 to a different server.

Which code segment should you execute?

- A. `RESTORE CERTIFICATE CERT2
FROM FILE='CERT2.CER'
WITH PRIVATE KEY (FILE = 'CERT2.KEY',
DECRYPTION BY PASSWORD='p@ssw0rd1');`
- B. `CREATE CERTIFICATE CERT1
FROM FILE='CERT1.CER'
WITH PRIVATE KEY (FILE = 'CERT1.KEY',
DECRYPTION BY PASSWORD='p@ssw0rd1');`
- C. `CREATE CERTIFICATE CERT2
ENCRYPTION BY PASSWORD='p@ssw0rd1'
WITH SUBJECT = 'EncryptionCertificate';`
- D. `CREATE CERTIFICATE CERT1
ENCRYPTION BY PASSWORD='p@ssw0rd1'
WITH SUBJECT = 'EncryptionCertificate';`

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

Answer: B

Question: 2

You need to create the InvoiceStatus table in DB1.

How should you define the InvoiceID column in the CREATE TABLE statement?

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

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

Answer: C

Question: 3

Which data type should you use for CustomerID?

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

Answer: D

Explanation:

Invoices.xml

All customer IDs are 11 digits. The first three digits of a customer ID represent the customer's country. The remaining eight digits are the customer's account number.

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

bigint: -2^{63} (-9,223,372,036,854,775,808) to $2^{63}-1$ (9,223,372,036,854,775,807)

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

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

Question: 4

You need to modify InsertInvoice to comply with the application requirements. Which code segment should you execute?

- A. `OPEN CERT1;`
`ALTER PROCEDURE Accounting.usp_InsertInvoice`
`WITH ENCRYPTION;`
`CLOSE CERT1;`
- B. `OPEN CERT2;`
`ALTER PROCEDURE Accounting.usp_InsertInvoice`
`WITH ENCRYPTION;`
`CLOSE CERT2;`
- C. `ADD SIGNATURE TO Accounting.usp_InsertInvoice`
`BY CERTIFICATE CERT1;`
- D. `ADD SIGNATURE TO Accounting.usp_InsertInvoice`
`BY CERTIFICATE CERT2;`

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

Answer: D

Question: 5

You attempt to process an invoice by using `usp_InsertInvoice.sql` and you receive the following error message: "Msg 515, Level 16, State 2, Procedure `usp_InsertInvoice`, Line 10 Cannot insert the value NULL into column 'InvoiceDate', table 'DB1.Accounting.Invoices'; column does not allow nulls. INSERT fails."

You need to modify `usp_InsertInvoice.sql` to resolve the error. How should you modify the INSERT statement?

- A. `InvoiceDate varchar(100) 'InvoiceDate',`

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

Answer: C

Question: 6

You need to modify the function in CountryFromID.sql to ensure that the country name is returned instead of the country ID.

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

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

Answer: D

Explanation:

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

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

Question: 7

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

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

Subquery returned more than 1 value. This is not permitted when the subquery follows =, !=, <, <=, >, >= or when the subquery is used as an expression."

You need to ensure that IndexManagement.sql executes properly.

Which WHILE statement should you use at line 18?

- A. WHILE SUM(@RowNumber) < (SELECT @counter FROM @indextable)
- B. WHILE @counter < (SELECT COUNT(RowNumber) FROM @indextable)
- C. WHILE COUNT(@RowNumber) < (SELECT @counter FROM @indextable)
- D. WHILE @counter < (SELECT SUM(RowNumber) FROM @indextable)

Answer: B

Question: 8

You need to convert the functionality of Legacy.sql to use a stored procedure.

Which code segment should the stored procedure contain?

- A.

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

```
CREATE PROC usp_InvoicesByCustomerAboveTotal(  
    @sqlstring AS nvarchar(1000))  
AS  
...
```
- C.

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

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

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

Answer: D

Explanation:

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

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

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

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

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

Question: 9

You need to create a function that filters invoices by CustomerID. The SELECT statement for the function is contained in InvoicesByCustomer.sql.

Which code segment should you use to complete the function?

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

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

Answer: A

Question: 10

DRAG DROP

You need to build a stored procedure that amortizes the invoice amount. Which code segment should you use to create the stored procedure? To answer, move the appropriate code segments from the list of code segments to the answer area and arrange them in the correct order.

Ordered List Title	Answer Choices Title
<div style="border: 1px solid black; height: 313px; width: 296px;"></div>	<pre> RECONFIGURE; EXEC sp_configure 'clr enabled', '1'; EXEC sp_recomple @objname = 'TaxCalc' CREATE PROCEDURE Accounting.Amortize(@total decimal(8,2), @period int) RETURNS decimal(8,2) AS EXTERNAL NAME TaxCalc.TreyResearch.Amortize; CREATE ASSEMBLY TaxCalc FROM 'C:\temp\TreyTax.DLL' CREATE ASSEMBLY TaxCalc FROM 'C:\temp\Amortize.cs' </pre>
	<p style="text-align: center;"> <input type="button" value=" << Move"/> <input type="button" value=" Remove >>"/> </p>

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;

```

Explanation:

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

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

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

Question: 11

You need to provide referential integrity between the Sessions table and Speakers table. Which code segment should you add at line 47 of Tables.sql?

- A. ALTER TABLE dbo.Sessions ADD CONSTRAINT
FK_Sessions_Speakers FOREIGN KEY (SessionID)
REFERENCES dbo.Speakers (SpeakerID);
- B. ALTER TABLE dbo.Sessions ADD CONSTRAINT
FK_Sessions_Speakers FOREIGN KEY (SpeakerID)
REFERENCES dbo.Speakers (SpeakerID);
- C. ALTER TABLE dbo.Speakers ADD CONSTRAINT
FK_Speakers_Sessions FOREIGN KEY (SpeakerID)
REFERENCES dbo.Sessions (SessionID);
- D. ALTER TABLE dbo.Speakers ADD CONSTRAINT
FK_Speakers_Sessions FOREIGN KEY (SessionID)
REFERENCES dbo.Sessions (SessionID);

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

Answer: B

Explanation:

<http://msdn.microsoft.com/en-us/library/ms189049.aspx>
<http://msdn.microsoft.com/en-us/library/ms179610.aspx>
<http://msdn.microsoft.com/en-us/library/ff878370.aspx>

Question: 12

You execute usp_TestSpeakers.

You discover that usp_SelectSpeakersByName uses inefficient execution plans.

You need to update usp_SelectSpeakersByName to ensure that the most efficient execution plan is used.

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

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

Answer: C

Explanation:

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

Question: 13

You need to recommend a solution to ensure that SQL1 supports the auditing requirements of usp_UpdateSpeakerName.
What should you include in the recommendation?

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

Answer: A

Question: 14

You are evaluating the table design.
You need to recommend a change to Tables.sql that reduces the amount of time it takes for usp_AttendeesReport to execute.
What should you add at line 14 of Tables.sql?

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

Answer: D

Explanation:

<http://msdn.microsoft.com/en-us/library/ms188300.aspx>
<http://msdn.microsoft.com/en-us/library/ms191250.aspx>

Question: 15

You need to modify usp_SelectSpeakersByName to support server-side paging. The solution must minimize the amount of development effort required.
What should you add to usp_SelectSpeakersByName?

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

Answer: B

Explanation:

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

Question: 16

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

The solution must meet the following requirements:

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

Which code block should you use?

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

Answer: A

Explanation:

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

Question: 17

You need to create the object used by the parameter of usp_InsertSessions.

Which statement should you use?

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

Answer: B

Question: 18

Developers report that `usp_UpdateSessionRoom` periodically returns error 3960.

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

What should you configure in `Procedures.sql`?

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

Answer: A

Question: 19

You discover that `usp.SelectSpeakersByName` executes slowly if `usp_UpdateSpeakerName` executes simultaneously.

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

What should you update?

- A. `Usp_UpdateSpeakerName` to use the `NOLOCK` query hint
- B. `Usp_UpdateSpeakerName` to use snapshot isolation
- C. `Usp_SelectSpeakersByName` to use the `NOLOCK` query hint
- D. `Usp_SelectSpeakersByName` to use snapshot isolation

Answer: C

`NOLOCK`

Is equivalent to `READUNCOMMITTED`.

`READUNCOMMITTED`

Specifies that dirty reads are allowed.

Question: 20

While testing `usp.GetFutureSessions`, you discover that `IX_Sessions` is accessed by a scan rather than a seek.

You need to minimize the amount of time it takes to execute `usp_GetFutureSessions`.

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

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

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

Answer: BE

Future delivery dates.

Thank You for trying 70-464 PDF Demo

To Buy Latest 70-464 Full Version Download visit link below

<https://www.certkillers.net/Exam/70-464>

Start Your 70-464 Preparation

[Limited Time Offer] Use Coupon “CKNET” for Further discount on your purchase. Test your 70-464 preparation with actual exam questions.

<https://www.certkillers.net>