70-762 Q&As

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4itsure.com/70-762.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

Note: This question is part of a series of questions that use the same or similar answer choices. An Answer choice may be correct for more than one question in the series. Each question independent of the other questions in this series.

Information and details provided in a question apply only to that question.

You are a database developer for a company. The company has a server that has multiple physical disks. The disks are not part of a RAID array. The server hosts three Microsoft SQL Server instances. There are many SQL jobs that run

during off-peak hours.

You observe that many deadlocks appear to be happening during specific times of the day.

You need to monitor the SQL environment and capture the information about the processes that are causing the deadlocks. Captured information must be viewable as the queries are running.

What should you do?

A. A. Create a sys.dm_os_waiting_tasks query.

B. Create a sys.dm_exec_sessions query.

C. Create a PerformanceMonitor Data Collector Set.

D. Create a sys.dm_os_memory_objects query.

E. Create a sp_configure `max server memory\\' query.

F. Create a SQL Profiler trace.

G. Create a sys.dm_os_wait_stats query.

H. Create an Extended Event.

Correct Answer: F

To view deadlock information, the Database Engine provides monitoring tools in the form of two trace flags, and the deadlock graph event in SQL Server Profiler.

Trace Flag 1204 and Trace Flag 1222 When deadlocks occur, trace flag 1204 and trace flag 1222 return information that is captured in the SQL Server error log. Trace flag 1204 reports deadlock information formatted by each nodeinvolved in the deadlock. Trace flag 1222 formats deadlock information, first by processes and then by resources. It is possible to enable bothtrace flags to obtain two representations of the same deadlock event.

References:https://technet.microsoft.com/en-us/library/ms178104(v=sql.105).aspx

QUESTION 2

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You have a database that contains a table named Employees. The table stored information about the employees of your

company. You need to implement the following auditing rules for the Employees table:

- Record any changes that are made to the data in the Employees table.

-

Customize the data recorded by the audit operations.

Solution: You implement a stored procedure on the Employees table.

Does the solution meet the goal?

A.

Yes

B.

No

Correct Answer: B

We should use table-valued functions, not procedures, to customize the recorded change data. References: https://msdn.microsoft.com/en-us/library/cc645858.aspx

QUESTION 3

Note: This question is part of a series of questions that present the same scenario. Each question in this series contains a unique solution. Determine whether the solution meets the stated goals. The Account table was created by using the following Transact-SQL statement:

```
CREATE TABLE Account
(
    AccountNumber int NOT NULL,
    ProductCode char(2) NOT NULL,
    Status tinyint NOT NULL,
    OpenDate date NOT NULL,
    CloseDate date,
    Balance decimal(15,2),
    AvailableBalance decimal(15,2)
);
```

There are more than 1 billion records in the Account table. The Account Number column uniquely identifies each account. The ProductCode column has 100 different values. The values are evenly distributed in the table. Table statistics are refreshed and up to date.

You frequently run the following Transact-SQL SELECT statements:

```
SELECT ProductCode, SUM(Balance) AS TotalSUM FROM Account WHERE ProductCode
<> 'CD' GROUP BY ProductCode;
SELECT AccountNumber, Balance FROM Account WHERE ProductCode = 'CD'
```

You must avoid table scans when you run the queries.

You need to create one or more indexes for the table.

Solution: You run the following Transact-SQL statement:

CREATE NONCLUSTERED INDEX IX_Account_ProductCode ON Account(ProductCode);

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

References: https://msdn.microsoft.com/en-za/library/ms189280.aspx

QUESTION 4

You have a view that includes an aggregate.

You must be able to change the values of columns in the view. The changes must be reflected in the tables that the view uses.

You need to ensure that you can update the view.

What should you create?

A. a nonclustered index

B. a schema-bound view

C. a stored procedure

D. an INSTEAD OF trigger

Correct Answer: B

Binds the view to the schema of the underlying table or tables. When SCHEMABINDING is specified, the base table or tables cannot be modified in a way that would affect the view definition. Views or tables that participate in a view created with the SCHEMABINDING clause cannot be dropped unless that view is dropped or changed so that it no longer has schema binding.

References: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql

QUESTION 5

You are optimizing the performance of a batch update process. You have tables and indexes that were created by running the following Transact-SQL statements:

```
CREATE TABLE Invoices (
    InvoiceID INT NOT NULL IDENTITY PRIMARY KEY CLUSTERED,
    CustomerID INT NOT NULL,
    OrderID INT NULL,
    IsCreditNote BIT NOT NULL,
    IsCreditValidated BIT NOT NULL DEFAULT Ø
)

CREATE INDEX IX_invoices_CustomerID_Filter_IsCreditValidated ON Invoices
(CustomerID) WHERE IsCreditValidated = 1

CREATE TABLE CreditValidation (
   CreditValidationID INT NOT NULL IDENTITY PRIMARY KEY CLUSTERED,
   CustomerID INT NOT NULL,
   ValidationDate DATETIME NOT NULL
)
```

The following query runs nightly to update the isCreditValidated field:

```
UPDATE I
SET IsCreditValidated = 1
FROM Invoices I
WHERE EXISTS (SELECT Ø FROM CreditValidation CV WHERE CV.CustomerID =
I.CustomerID AND CV.ValidationDate >= I.InvoiceDate)
AND I.IsCreditNote = 1
AND I.IsCreditValidated = Ø
AND I.InvocieDate >= DATEADD (DD, -7, GETDATE () )
```

You review the database and make the following observations:

Most of the IsCreditValidated values in the Invoices table are set to a value of 1.

There are many unique InvoiceDate values.

The CreditValidation table does not have an index.

Statistics for the index IX_invoices_CustomerID_Filter_IsCreditValidated indicate there are no individual seeks but multiple individual updates.

You need to ensure that any indexes added can be used by the update query. If the

IX_invoices_CustomerId_Filter_IsCreditValidated index cannot be used by the query, it must be removed. Otherwise, the query must be modified to use with

the index.

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

NOTE: Each correct selection is worth one point.

A. Add a filtered nonclustered index to Invoices on InvoiceDate that selects where IsCreditNote= 1 and IsCreditValidated = 0.

B. Rewrite the update query so that the condition for IsCreditValidated = 0 precedes the condition for IsCreditNote = 1.

C. Create a nonclustered index for invoices in IsCreditValidated, InvoiceDate with an include statement using IsCreditNote and CustomerID.

D. Add a nonclustered index for CreditValidation on CustomerID.

E. Drop the IX_invoices_CustomerId_Filter_IsCreditValidatedIndex.

Correct Answer: ABE

A filtered index is an optimized nonclustered index especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve query performance as well as reduce index maintenance and storage costs compared with full-table indexes.

References: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-filtered-indexes