70-762^{Q&As}

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.passapply.com/70-762.html

100% Passing Guarantee
100% Money Back Assurance

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

Instant Download After Purchase

100% Money Back Guarantee

365 Days Free Update

800,000+ Satisfied Customers

QUESTION 1

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.

You are developing a new application that uses a stored procedure. The stored procedure inserts thousands of records as a single batch into the Employees table.

Users report that the application response time has worsened since the stored procedure was updated. You examine disk-related performance counters for the Microsoft SQL Server instance and observe several high values that include a

disk performance issue. You examine wait statistics and observe an unusually high WRITELOG value.

You need to improve the application response time.

Solution: You replace the stored procedure with a user-defined function.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

References: https://msdn.microsoft.com/en-us/library/ms345075.aspx

---

QUESTION 2

Note: This question is part of a series of questions that use the same or similar answer choices. As answer choice may be correct for more than one question in the series. Each question is independent of the other questions in this series. Information and details provided in a question apply only to that question.

You have a Microsoft SQL Server database named DB1 that contains the following tables:

Table name	Description
TBL1	• Column1 is configured as the primary key. • Column2 will store the year. • TBL1 only stores data for the year 2016. • The table will contain 1 million records.
TBL2	• Column1 is configured as the primary key. • Column2 will store the year. • TBL2 only stores data for the year 2015. • The table will contain 1 million records.

Users frequently run the following query. The users report that the query takes a long time to return results.

```
SELECT Column1, Column2, Column3
FROM (
    SELECT Column1, Column2, Column3
    FROM TBL1
    UNION ALL
    SELECT Column1, Column2, Column3
    FROM TBL2)
WHERE Column2 = <year> AND Column3 = 1
```

You need to minimize the amount of time required for the query to return data.

A. Create clustered indexes on TBL1 and TBL2.

B. Create a clustered index on TBL1.Create a nonclustered index on TBL2 and add the most frequently queried column as included columns.

C. Create a nonclustered index on TBL2 only.

D. Create UNIQUE constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.

E. Drop existing indexes on TBL1 and then create a clustered columnstore index. Create a nonclustered columnstore index on TBL1.Create a nonclustered index on TBL2.

F. Drop existing indexes on TBL1 and then create a clustered columnstore index. Create a nonclustered columnstore index on TBL1.Make no changes to TBL2.

G. Create CHECK constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.

H. Create an indexed view that combines columns from TBL1 and TBL2.

Correct Answer: D

A partitioned view is a view defined by a UNION ALL of member tables structured in the same way, but stored separately as multiple tables in either the same instance of SQL Server or in a group of autonomous instances of SQL Server servers, called federated database servers.

References: https://docs.microsoft.com/en-us/sql/t-sql/statements/create-view-transact-sql?view=sql-server-2017#partitioned-views

QUESTION 3

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 need to create a stored procedure that updates the Customer, CustomerInfo, OrderHeader, and OrderDetails tables in order.

You need to ensure that the stored procedure:

Runs within a single transaction.

Commits updates to the Customer and CustomerInfo tables regardless of the status of updates to the OrderHeader and OrderDetail tables.

Commits changes to all four tables when updates to all four tables are successful.

Solution: You create a stored procedure that includes the following Transact-SQL segment:

```
DECLARE @CustomerComplete bit = 0
BEGIN TRY
    BEGIN TRAN
        UPDATE Customer ...
        UPDATE CustomerInfo ...

        SET @CustomerComplete = 1

        SAVE TRAN TR1

        UPDATE OrderHeader ...
        UPDATE OrderDetail ...
    COMMIT TRAN
END TRAN
BEGIN CATCH
    IF (@CustomerComplete = 1) AND (XACT_STATE() = 1)
    BEGIN
        COMMIT TRAN
    END
    ELSE IF XACT_STATE() = 1
        ROLLBACK TRAN
END CATCH
```

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

Need to handle the case where the first two updates (OrderHeader, OrderDetail) are successful, but either the 3rd or the 4th (OrderHeader, OrderDetail) fail. We add the @CustomerComplete variable in the BEGIN TRY block, and test it in the BEGIN CATCH block.

Note: XACT_STATE indicates whether the request has an active user transaction, and whether the transaction is capable of being committed. XACT_STATE =1: the current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction.

References: https://docs.microsoft.com/en-us/sql/t-sql/functions/xact-state-transact-sql

QUESTION 4

You have the following stored procedure that is called by other stored procedures and applications:

```
CREATE PROCEDURE UpdateCustomer @CustomerId INT
AS
BEGIN
    EXEC ProcessCustomer_Internal @CustomerId
    DECLARE @Status INT
    SELECT @Status = Status FROM Customer WHERE CustomerId =
@CustomerId
END
```

You need to modify the stored procedure to meet the following requirements:

Always return a value to the caller.

Return 0 if @Status is NULL.

Callers must be able to use @Status as a variable.

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

NOTE: Each correct selection is worth one point.

A. Replace NULL values with 0. Add a PRINT statement to return @Status.

B. Add a RETURN statement.

C. Replace NULL values with 0. Add an output parameter to return @Status.

D. Replace NULL values with 0. Add a SELECT statement to return @Status.

E. Add a PRINT statement.

F. Add a SELECT statement to return @Status.

G. Add an output parameter to return @Status.

Correct Answer: BC

There are three ways of returning data from a procedure to a calling program: result sets, output parameters, and return codes.

References: https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/return-data-from-a-stored-procedure?view=sql-server-2017

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