70-762$^{\text{Q\&As}}$

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

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

https://www.leads4pass.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 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.

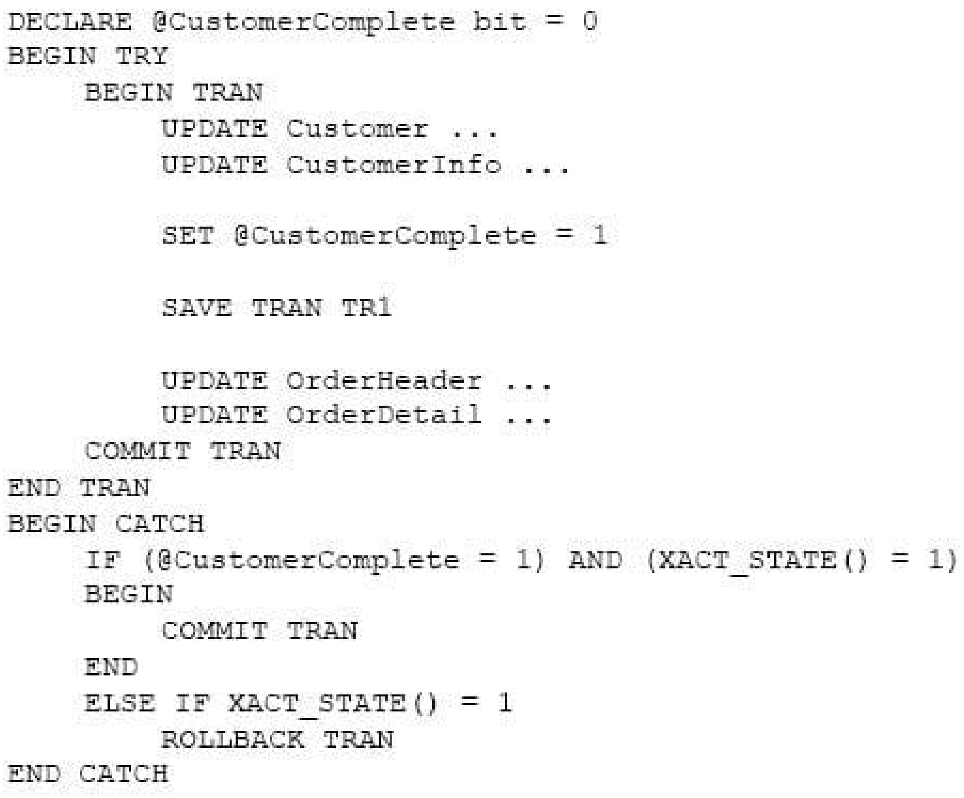
You need to ensure that the stored procedure:

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 2

You have a table that stores transactions partitioned by year. Users occasionally experience performance issues when they access the table. The table is locked exclusively when the records are updated. You need to prevent exclusive locks

on the table and maintain data integrity.

What should you do?

A. Set the LOCK_EXCALATION property to DISABLE

B. Set the DATA_COMPRESSION property to ROW at the partition level

C. Set the LOCK_EXCALATION property to AUTO

D. Set the LOCK_EXCALATION property to TABLE

E. Set the DATA_COMPRESSION property to PAGE at the partition level

Correct Answer: C

The default lock escalation mode is called TABLE, it implements SQL Server\\'s lock escalation on all types of tables whether partitioned or not partitioned.

There are two more lock escalation modes: AUTO and DISABLE.

The AUTO mode enables lock escalation for partitioned tables only for the locked partition. For non-partitioned tables it works like TABLE.

The DISABLE mode removes the lock escalation capability for the table and that is important when concurrency issues are more important than memory needs for specific tables.

References:

https://www.mssqltips.com/sqlservertip/4359/altering-lock-escalation-for-sql-server-tables/

QUESTION 3

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 must monitor and optimize the SQL Server to maximize throughput, response time, and overall SQL performance.

You need to examine delays in executed threads, including errors with queries and batches.

What should you do?

A. Create a sys.dm_os_waiting_tasks query.

B. Create a sys.dm_exec_sessions query.

C. Create a Performance Monitor 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: G

sys.dm_os_wait_stats returns information about all the waits encountered by threads that executed. You can use this aggregated view to diagnose performance issues with SQL Server and also with specific queries and batches. Incorrect Answers:

A: sys.dm_os_waiting_tasks returns information about the wait queue of tasks that are waiting on some resource.

References: https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-os-wait-stats-transact-sql

QUESTION 4

You are developing a database reporting solution for a table that contains 900 million rows and is 103 GB.

The table is updated thousands of times a day, but data is not deleted.

The SELECT statements vary in the number of columns used and the amount of rows retrieved.

You need to reduce the amount of time it takes to retrieve data from the table. The must prevent data duplication.

Which indexing strategy should you use?

A. a nonclustered index for each column in the table

B. a clustered columnstore index for the table

C. a hash index for the table

D. a clustered index for the table and nonclustered indexes for nonkey columns

Correct Answer: B

Columnstore indexes are the standard for storing and querying large data warehousing fact tables. It uses column-based data storage and query processing to achieve up to 10x query performance gains in your data warehouse over

traditional row-oriented storage.

A clustered columnstore index is the physical storage for the entire table.

Generally, you should define the clustered index key with as few columns as possible.

A nonclustered index contains the index key values and row locators that point to the storage location of the table data. You can create multiple nonclustered indexes on a table or indexed view. Generally, nonclustered indexes should be

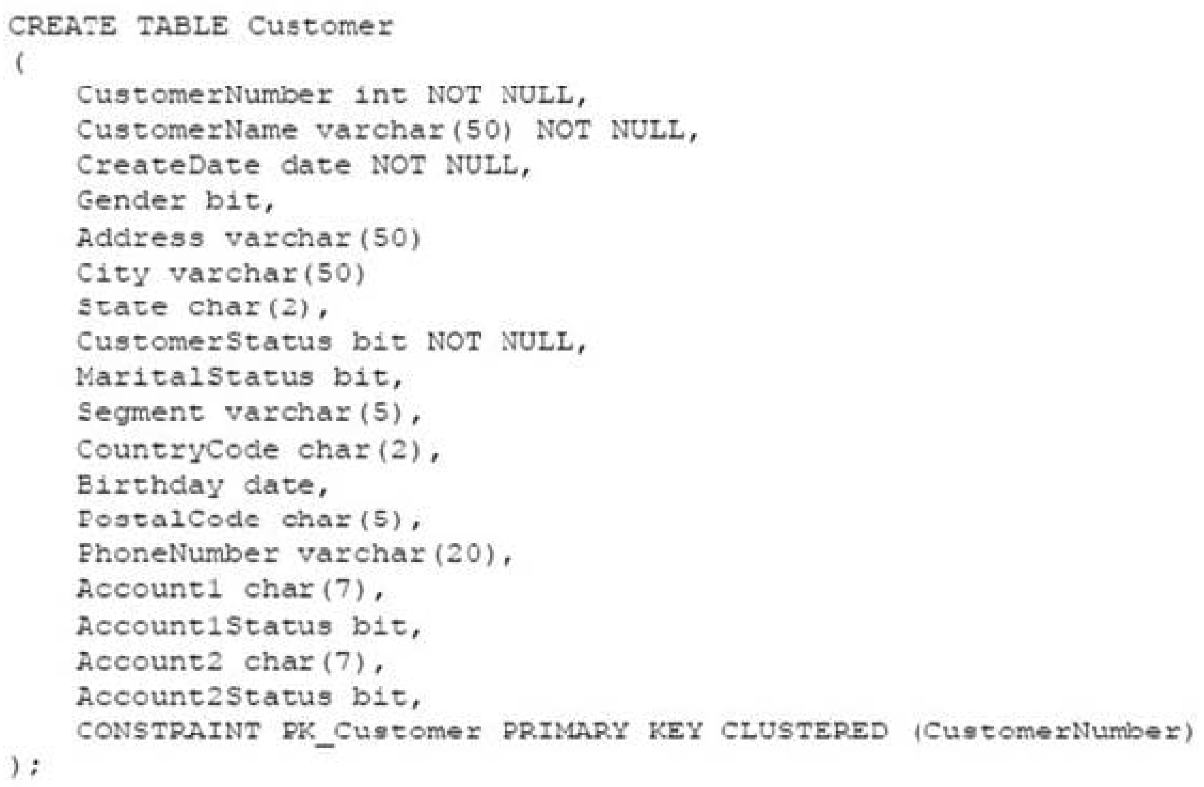
designed to improve the performance of frequently used queries that are not covered by the clustered index.

References: https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview?view=sql-server-2017

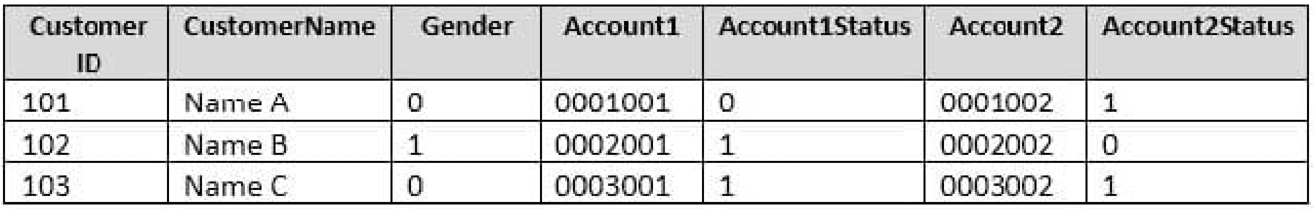
QUESTION 5

Note: This question is part of a series of questions that use the same scenario. For your convenience, the scenario is repeated in each question. Each question presents a different goal and answer choices, but the text of the scenario is exactly the same in each question in this series.

You have a database named DB1 that contains the following tables: Customer, CustomerToAccountBridge, and CustomerDetails. The three tables are part of the Sales schema. The database also contains a schema named Website. You create the Customer table by running the following Transact-SQL statement: The value of the CustomerStatus column is equal to one for active customers. The value of the Account1Status and Account2Status columns are equal to one for active accounts. The following table displays selected columns and rows from the Customer table.



```
CREATE TABLE Customer
(
    CustomerNumber int NOT NULL,
    CustomerName varchar(50) NOT NULL,
    CreateDate date NOT NULL,
    Gender bit,
    Address varchar(50)
    City varchar(50)
    State char(2),
    CustomerStatus bit NOT NULL,
    MaritalStatus bit,
    Segment varchar(5),
    CountryCode char(2),
    Birthday date,
    PostalCode char(5),
    PhoneNumber varchar(20),
    Account1 char(7),
    Account1Status bit,
    Account2 char(7),
    Account2Status bit,
    CONSTRAINT PK_Customer PRIMARY KEY CLUSTERED (CustomerNumber)
);
```



Customer ID	CustomerName	Gender	Account1	Account1Status	Account2	Account2Status
101	Name A	0	0001001	0	0001002	1
102	Name B	1	0002001	1	0002002	0
103	Name C	0	0003001	1	0003002	1

You plan to create a view named Website.Customer and a view named Sales.FemaleCustomers.

Website.Customer must meet the following requirements:

Allow users access to the CustomerName and CustomerNumber columns for active customers.

Allow changes to the columns that the view references. Modified data must be visible through the view.

Prevent the view from being published as part of Microsoft SQL Server replication.

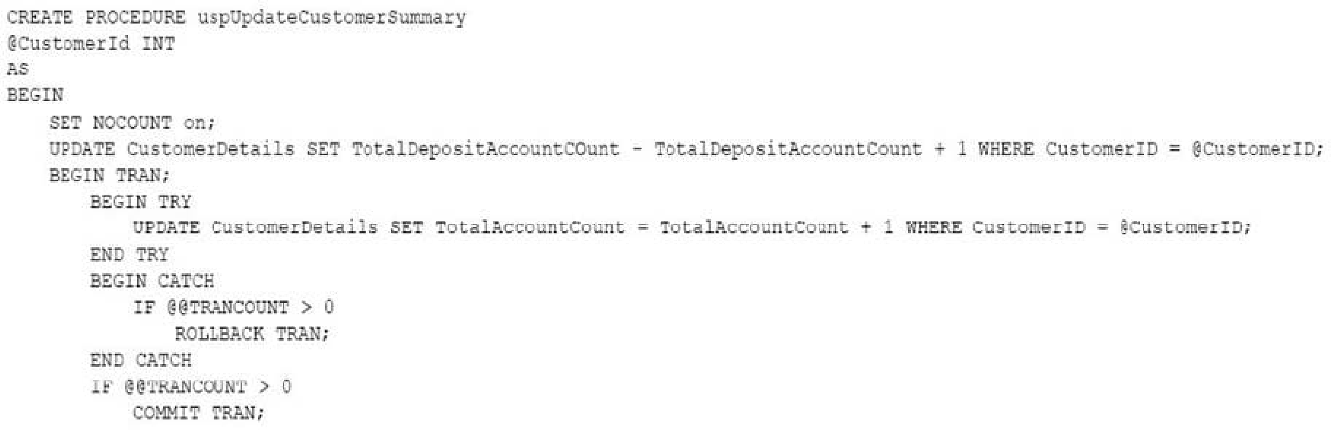
Sales.Female.Customers must meet the following requirements:

Allow users access to the CustomerName, Address, City, State and PostalCode columns.

Prevent changes to the columns that the view references.

Only allow updates through the views that adhere to the view filter.

You have the following stored procedures: spDeleteCustAcctRelationship and spUpdateCustomerSummary. The spUpdateCustomerSummary stored procedure was created by running the following Transacr-SQL statement:



```
CREATE PROCEDURE uspUpdateCustomerSummary
@CustomerId INT
AS
BEGIN
    SET NOCOUNT on;
    UPDATE CustomerDetails SET TotalDepositAccountCOunt - TotalDepositAccountCount + 1 WHERE CustomerID = @CustomerID;
    BEGIN TRAN;
        BEGIN TRY
            UPDATE CustomerDetails SET TotalAccountCount = TotalAccountCount + 1 WHERE CustomerID = @CustomerID;
        END TRY
        BEGIN CATCH
            IF @@TRANCOUNT > 0
                ROLLBACK TRAN;
        END CATCH
        IF @@TRANCOUNT > 0
            COMMIT TRAN;
```

You run the uspUpdateCustomerSummary stored procedure to make changes to customer account summaries. Other stored procedures call the spDeleteCustAcctRelationship to delete records from the CustomerToAccountBridge table.

When you start uspUpdateCustomerSummary, there are no active transactions. The procedure fails at the second update statement due to a CHECK constraint violation on the TotalDepositAccountCount column.

What is the impact of the stored procedure on the CustomerDetails table?

A. A. The value of the TotalAccountCount column decreased.

B. The value of the TotalDepositAccountCount column is not changed.

C. The statement that modifies TotalDepositAccountCount is excluded from the transaction.

D. The value of the TotalAccountCount column is not changed.

E. The value of the TotalDepositAccountCount column is increased.

F. The statement that modifies TotalAccountCount column is excluded from the transaction.

G. The value of the TotalDepositAcountCount column is decreased.

Correct Answer: D

70-762 PDF Dumps 70-762 Study Guide 70-762 Braindumps