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

QUESTION 1

Database users report that SELECT statements take a long time to return results. You run the following Transact-SQL statement:

```
SELECT OBJECT_NAME([object_id]) AS [object_name],
d.equality_columns, d.inequality_columns, d.included_columns
FROM sys.dm_db_missing_index_details;
```

Object_name	Equality_columns	Inequality_columns	Included_columns
[Users]	[CountryCode]	[UserStatus]	[UserName]

You need to create one nonclustered covering index that contains all of the columns in the above table. You must minimize index key size. Which Transact-SQL statement should you run?

A. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserName);

B. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserStatus) INCLUDE (UserName);

C. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserStatus, UserName);

D. CREATE NONCLUSTERED INDEX IX_User ON Users (UserStatus, CountryCode) INCLUDE (UserName);

Correct Answer: D

Use the UserStatus as the first column in the index, as it is an in_equality column. Incorrect Answers:

A: UserStatus is not included.

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

QUESTION 2

Note: This question is part of a series of questions that use the same answer choices. An answer choice may be correct for more than one question on 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 work on an OLTP database that has no memory-optimized file group defined.

You have a table names tblTransaction that is persisted on disk and contains the information described in the following table:

Item	Name	Data Type	Nullable	Notes
Column	TransactionDate	Date	No	For each transaction date, there are only about 100,000 records. The table contains over one billion records in total.
Column	SequenceNo	bigint	No	Uniquely identifies a transaction record within a date
Column	AccountId	int	No	
Column	ValueType	char(3)	No	
Column	Amount	decimail(20.2)	Yes	
	IX_ValueType			Nonclustered columnstore index on the ValueType column.

Users report that the following query takes a long time to complete.

```
SELECT TransactionDate, COUNT(*) AS TotalCount FROM tblTransaction
WHERE TransactionDate - DATEADD(D, -1, CONVERT(DATE,CONVERT(VARCHAR(8),
GETDATE(),112)112))
GROUP BY TransactionDate;
```

You need to create an index that:
- improves the query performance
- does not impact the existing index
- minimizes storage size of the table (inclusive of index pages).

What should you do?

Users report that the following query takes a long time to complete.

```
SELECT TransactionDate, COUNT(*) AS TotalCount FROM tblTransaction
WHERE TransactionDate - DATEADD(D, -1, CONVERT(DATE,CONVERT(VARCHAR(8),
GETDATE(),112)112))
GROUP BY TransactionDate;
```

You need to create an index that:

-improves the query performance

-does not impact the existing index

- minimizes storage size of the table (inclusive of index pages).

What should you do?

A. Create aclustered index on the table.

B. Create a nonclustered index on the table.

C. Create a nonclustered filtered index on the table.

D. Create a clustered columnstore index on the table.

E. Create a nonclustered columnstore index on the table.

F. Create a hashindex on the table.

Correct Answer: C

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, reduce index maintenance costs, and reduce index storage costs compared with full-table indexes.

QUESTION 3

You have several real-time applications that constantly update data in a database. The applications run more than 400 transactions per second that insert and update new metrics from sensors.

A new web dashboard is released to present the data from the sensors. Engineers report that the applications take longer than expected to commit updates.

You need to change the dashboard queries to improve concurrency and to support reading uncommitted data.

What should you do?

A. Use the NOLOCK option.

B. Execute the DBCC UPDATEUSAGE statement.

C. Use the max worker threads option.

D. Use a table-valued parameter.

E. Set SET ALLOW_SNAPSHOT_ISOLATION to ON.

F. Set SET XACT_ABORT to ON.

G. Execute the ALTER TABLE T1 SET (LOCK_ESCALATION = AUTO); statement.

H. Use the OUTPUT parameters.

Correct Answer: A

The NOLOCK hint allows SQL to read data from tables by ignoring any locks and therefore not being blocked by other processes. This can improve query performance, but also introduces the possibility of dirty reads.

Incorrect Answers:

F: When SET XACT_ABORT is ON, if a Transact-SQL statement raises a run-time error, the entire transaction is terminated and rolled back.

G: DISABLE, not AUTO, would be better.

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.

Note: SQL Server\\'s locking mechanism uses memory resources to maintain locks. In situations where the number of row or page locks increases to a level that decreases the server\\'s memory resources to a minimal level, SQL Server\\'s locking strategy converts these locks to entire table locks, thus freeing memory held by the many single row or page locks to one table lock. This process is called lock escalation, which frees memory, but reduces table concurrency.

References: https://www.mssqltips.com/sqlservertip/2470/understanding-the-sql-server-nolock-hint/

QUESTION 4

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:

```
BEGIN TRY
    BEGIN TRAN
        UPDATE Customer ...
        UPDATE CustomerInfo ...
        UPDATE OrderHeader ...
        UPDATE OrderDetail ...
    COMMIT TRAN
END TRAN
BEGIN CATCH
    IF XACT STATE() = 1
        ROLLBACK TRAN
END CATCH
```

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

All four tables are updated in a single transaction.

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

References: https://docs.microsoft.com/en-us/sql/t-sql/language-elements/begin-transaction-transact-sql

QUESTION 5

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 identify previous situations where a modification has prevented queries from selecting data in tables.

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.