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

You use Microsoft SQL Server Profile to evaluate a query named Query1. The Profiler report indicates the following issues:

At each level of the query plan, a low total number of rows are processed.

The query uses many operations. This results in a high overall cost for the query.

You need to identify the information that will be useful for the optimizer.

What should you do?

A. Start a SQL Server Profiler trace for the event class Performance statistics in the Performance event category.

B. Create one Extended Events session with the sqlserver.missing_column_statistics event added.

C. Start a SQL Server Profiler trace for the event class Soft Warnings in the Errors and Warnings event category.

D. Create one Extended Events session with the sqlserver.error_reported event added.

Correct Answer: A

The Performance Statistics event class can be used to monitor the performance of queries, stored procedures, and triggers that are executing. Each of the six event subclasses indicates an event in the lifetime of queries, stored procedures, and triggers within the system. Using the combination of these event subclasses and the associated sys.dm_exec_query_stats, sys.dm_exec_procedure_stats and sys.dm_exec_trigger_stats dynamic management views, you can reconstitute the performance history of any given query, stored procedure, or trigger.

References: https://docs.microsoft.com/en-us/sql/relational-databases/event-classes/performance-statistics-event-class?view=sql-server-2017

QUESTION 3

You are designing a stored procedure for a database named DB1.

The following requirements must be met during the entire execution of the stored procedure:

The stored procedure must only read changes that are persisted to the database.

SELECT statements within the stored procedure should only show changes to the data that are made by the stored procedure.

You need to configure the transaction isolation level for the stored procedure.

Which Transact-SQL statement or statements should you run?

A. SET TRANSACTION ISOLATION LEVEL READ UNCOMMITED ALTER DATABASE DB1 SET READ_COMMITED_SNAPSHOT ON

B. SET TRANSACTION ISOLATION LEVEL READ COMMITED ALTER DATABASE DB1 SET READ_COMMITED_SNAPSHOT OFF

C. SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

D. SET TRANSACTION ISOLATION LEVEL READ UNCOMMITED ALTER DATABASE SET READ_COMMITED_SNAPSHOT OFF

Correct Answer: B

READ COMMITTED specifies that statements cannot read data that has been modified but not committed by other transactions. This prevents dirty reads. Data can be changed by other transactions between individual statements within the

current transaction, resulting in nonrepeatable reads or phantom data. This option is the SQL Server default.

Incorrect Answers:

A, D: READ UNCOMMITTED specifies that statements can read rows that have been modified by other transactions but not yet committed.

References: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/snapshot-isolation-in-sql-server

QUESTION 4

Note: The 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 is independent of the other question in the series. Information and details provided in a question apply only to that question.

You have a database named DB1. The database does not have a memory optimized filegroup. You create a table by running the following Transact-SQL statement:

```
CREATE TABLE tblTransaction(
      [TransactionID] [int] NOT NULL PRIMARY KEY,
      [TransactionDate] [date] NOT NULL,
      [AccountId] [int] NOT NULL,
      [ValueType] [char](3) NOT NULL,
      [Amount] [decimal](20,2) NULL
);
```

The table is currently used for OLTP workloads. The analytics user group needs to perform real-time operational analytics that scan most of the records in the table to aggregate on a number of columns. You need to add the most efficient index to support the analytics workload without changing the OLTP application.

What should you do?

A. Create a clustered indexon 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 hash index on the table.

Correct Answer: E

A nonclustered columnstore index enables real-time operational analytics in which the OLTP workload uses the underlying clustered index, while analytics run concurrently on the columnstore index.

Columnstore indexes can achieve up to 100xbetter performance on analytics and data warehousing workloads and up to 10x better data compression than traditional rowstore indexes. These recommendations will help your queries achieve the very fast query performance that columnstore indexes are designed to provide.

References:https://msdn.microsoft.com/en-us/library/gg492088.aspx

QUESTION 5

You manage a database that supports an Internet of Things (IoS) solution. The database records metrics from over 100 million devices every minute. The database requires 99.995% uptime.

The database uses a table named Checkins that is 100 gigabytes (GB) in size. The Checkins table stores metrics from the devices. The database also has a table named Archive that stores four terabytes (TB) of data. You use stored

procedures for all access to the tables.

You observe that the wait type PAGELATCH_IO causes large amounts of blocking.

You need to resolve the blocking issues while minimizing downtime for the database.

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

A. Convert all stored procedures that access the Checkins table to natively compiled procedures.

B. Convert the Checkins table to an In-Memory OLTP table.

C. Convert all tables to clustered columnstore indexes.

D. Convert the Checkins table to a clustered columnstore index.

Correct Answer: AB

Natively compiled stored procedures are Transact-SQL stored procedures compiled to native code that access memory-optimized tables. Natively compiled stored procedures allow for efficient execution of the queries and business logic in the stored procedure.

SQL Server In-Memory OLTP helps improve performance of OLTP applications through efficient, memory-optimized data access, native compilation of business logic, and lock- and latch free algorithms. The In-Memory OLTP feature includes memory-optimized tables and table types, as well as native compilation of Transact-SQL stored procedures for efficient access to these tables.

References: https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/natively-compiled-stored-procedures

https://docs.microsoft.com/en-us/sql/relational-databases/in-memory-oltp/memory-optimized-tables