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.geekcert.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

You run the following Transact-SQL statement:

```
CREATE TABLE OrderLines(
   OrderLineID INT NOT NULL IDENTITY PRIMARY KEY CLUSTERED,
   OrderID INT NOT NULL,
   StockItemID INT NOT NULL,
   Description NVARCHAR(100) NOT NULL,
   Quantity INT NOT NULL,
   UnitPrice DECIMAL(18, 2) NULL
)
```

There are multiple unique OrderID values. Most of the UnitPrice values for the same OrderID are different.

You need to create a single index seek query that does not use the following operators: Nested loop Sort Key lookup

Which Transact-SQL statement should you run?

A. CREATE INDEX IX_OrderLines_1 ON OrderLines (OrderID, UnitPrice) INCLUDE (Description, Quantity)

B. CREATE INDEX IX_OrderLines_1 ON OrderLines (OrderID, UnitPrice) INCLUDE (Quantity)

C. CREATE INDEX IX_OrderLines_1 ON OrderLines (OrderID, UnitPrice, Quantity)

D. CREATE INDEX IX_OrderLines_1 ON OrderLines (UnitPrice, OrderID) INCLUDE (Description, Quantity)

Correct Answer: A

An index with nonkey columns can significantly improve query performance when all columns in the query are included in the index either as key or nonkey columns. Performance gains are achieved because the query optimizer can locate all

the column values within the index; table or clustered index data is not accessed resulting in fewer disk I/O operations.

Note: All data types except text, ntext, and image can be used as nonkey columns.

Incorrect Answers:

C: Redesign nonclustered indexes with a large index key size so that only columns used for searching and lookups are key columns.

D: The most unique column should be the first in the index.

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

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

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.

Your company has employees in different regions around the world.

You need to create a database table that stores the following employee attendance information:

1.

Employee ID

2.

Date and time employee checked in to work

3.

Date and time employee checked out of work

Date and time information must be time zone aware and must not store fractional seconds.

Solution: You run the following Transact-SQL statement:

```
CREATE TABLE [dbo].[EmployeeAttendance](
    EmployeeID int NOT NULL,
    DateCheckedIn datetime2(0) NOT NULL,
    DateCheckedOut datetime2(0) NOT NULL )
```

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

Datetime2 stores fractional seconds.

Datetime2 defines a date that is combined with a time of day that is based on 24-hour clock. datetime2 can be considered as an extension of the existing datetime type that has a larger date range, a larger default fractional precision, and

optional user-specified precision.

Reference:

https://docs.microsoft.com/en-us/sql/t-sql/data-types/datetime2-transact-sql?view=sql-server-2017

https://msdn.microsoft.com/en-us/library/bb677335.aspx

QUESTION 4

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 the SQL Server instances in real time and optimize the server to maximize throughput, response time, and overall SQL performance.

What should you do?

A. A. Create asys.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: B

sys.dm_exec_sessions returns one row per authenticated session on SQL Server. sys.dm_exec_sessions is a server-scope view that shows information about all active user connections and internal tasks. This information includes client

version, client program name, client login time, login user, current session setting, and more. Use sys.dm_exec_sessions to first view the current system load and to identify a session of interest, and then learn more information about that session by using other dynamic management views or dynamic management functions.

Examples of use include finding long-running cursors, and finding idle sessions that have open transactions.

QUESTION 5

You have the following stored procedure:

```
CREATE PROCEDURE AddNextNumber @Number INT
AS
BEGIN
   SET ANSI_DEFAULTS ON
   INSERT INTO Numbers (Number) VALUES (@Number)
END
```

The Numbers table becomes unavailable when you run the stored procedure. The stored procedure obtains an exclusive lock on the table and does not release the lock.

What are two possible ways to resolve the issue? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. Remove the implicit transaction and the SET ANSI_DEFAULTS ON statement.

B. Set the ANSI_DEFAULT statement to OFF and add a COMMIT TRANSACTION statement after the INSERT statement.

C. Add a COMMIT TRANSACTION statement after the INSERT statement.

D. Remove the SET ANSI_DEFAULTS ON statement.

Correct Answer: CD

SET ANSI_DEFAULTS is a server-side setting that the client does not modify. When enabled (ON), this option enables SET IMPLICIT_TRANSACTIONS (and some other options).

The SET IMPLICIT_TRANSACTIONS, when ON, the system is in implicit transaction mode.

This means that if @@TRANCOUNT = 0, any of the following Transact-SQL statements begins a new transaction. It is equivalent to an unseen BEGIN TRANSACTION being executed first: ALTER TABLE, FETCH, REVOKE, BEGIN

TRANSACTION, GRANT, SELECT, CREATE, INSERT, TRUNCATE TABLE, DELETE, OPEN, UPDATE, DROP.

References: https://docs.microsoft.com/en-us/sql/t-sql/statements/set-implicit-transactions-transact-sql?view=sql-server-2017