

70-762 Dumps

Developing SQL Databases (beta)

<https://www.certleader.com/70-762-dumps.html>



NEW QUESTION 1

You manage a database with tables named Invoice and InvoiceDetails. Each invoice may have multiple records.

Users update the InvoiceDetails table by using a .NET web application. The application retrieves records from both tables and updates the tables by running an inline update statement.

Users experience slow performance when updating records in the application. The solution must meet the following requirements:

Must use a stored procedure.

Must not use inline update statements

Must use a table-valued parameter.

Must call the stored procedure to update all records. You need to optimize performance.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Actions

Create a stored procedure and use invDetails as a read-only input parameter.

Create a stored procedure and use invDetails as the input parameter.

Send invoice detail records to the stored procedure as a string value for the input parameter.

Create a user-defined table type named invDetails like schema of InvoiceDetails table.

Create an alias type from VARCHAR (MAX) Named invDetails.

Send invoice detail records to the stored procedure as a ADO.NET table for the input parameter

Answer Area



Answer:

Explanation: Box 1: Create a user-defined table type...

Table-valued parameters are declared by using user-defined table types. You can use table-valued parameters to send multiple rows of data to a Transact-SQL statement or a routine, such as a stored procedure or function, without creating a temporary table or many parameters.

Box 2: ...read-only input parameter.

Table-valued parameters must be passed as input READONLY parameters to Transact-SQL routines. Box 3:

Example

The following example uses Transact-SQL and shows you how to create a table-valued parameter type, declare a variable to reference it, fill the parameter list, and then pass the values to a stored procedure.

```
USE AdventureWorks2012;
```

```
/* Create a table type. */
```

```
CREATE TYPE LocationTableType AS TABLE ( LocationName VARCHAR(50)
```

```
, CostRate INT ); GO
```

```
/* Create a procedure to receive data for the table-valued parameter. */ CREATE PROCEDURE dbo. usp_InsertProductionLocation
```

```
@TVP LocationTableType READONLY Etc."
```

```
/* Declare a variable that references the type. */ DECLARE @LocationTVP AS LocationTableType;
```

```
/* Add data to the table variable. */
```

```
INSERT INTO @LocationTVP (LocationName, CostRate) SELECT Name, 0.00
```

```
FROM AdventureWorks2012.Person.StateProvince;
```

```
/* Pass the table variable data to a stored procedure. */ EXEC usp_InsertProductionLocation @LocationTVP; GO
```

References:

<https://docs.microsoft.com/en-us/sql/relational-databases/tables/use-table-valued-parameters-database-engine?vi>

NEW QUESTION 2

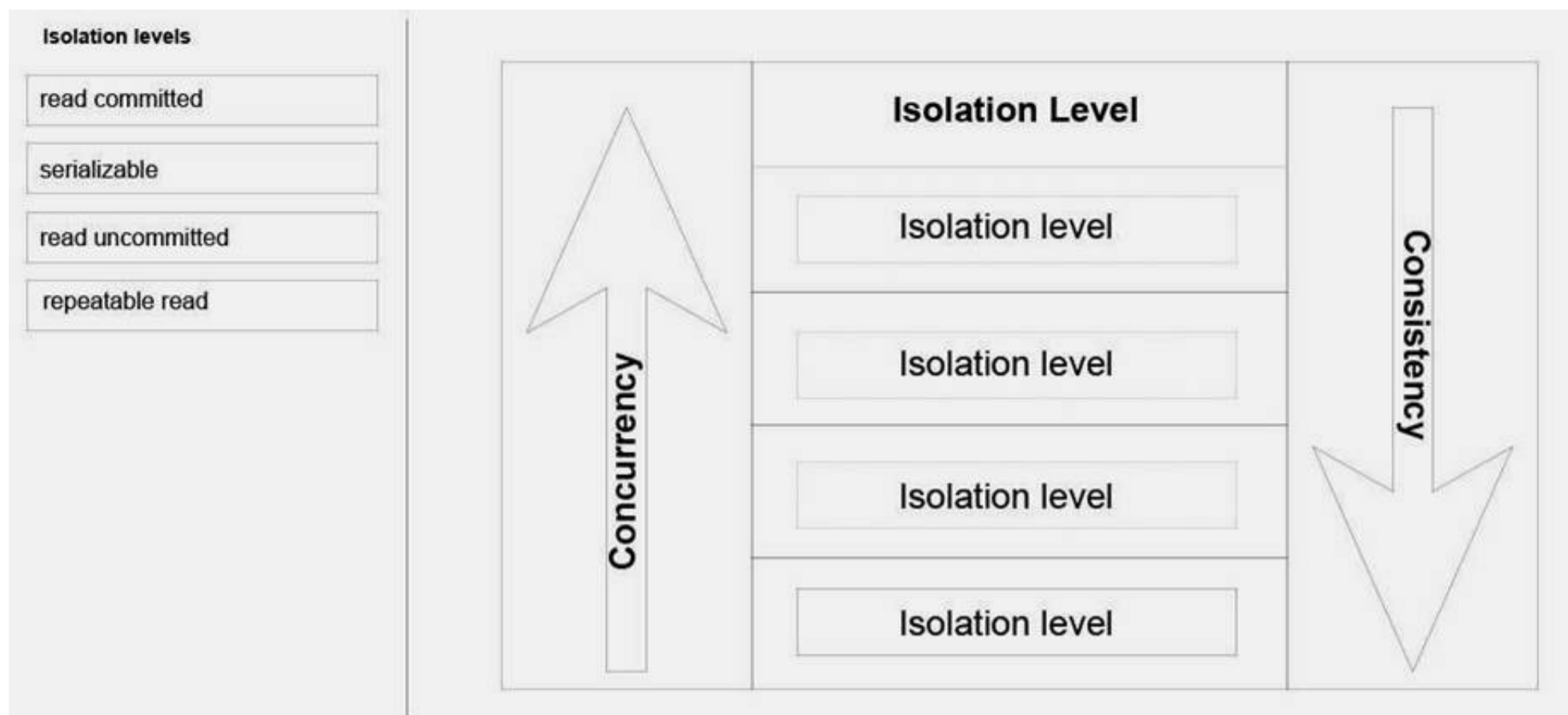
You are analyzing the performance of a database environment.

Applications that access the database are experiencing locks that are held for a large amount of time. You are experiencing isolation phenomena such as dirty, nonrepeatable and phantom reads.

You need to identify the impact of specific transaction isolation levels on the concurrency and consistency of data.

What are the consistency and concurrency implications of each transaction isolation level? To answer, drag the appropriate isolation levels to the correct locations.

Each isolation level may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.



Answer:

Explanation: Read Uncommitted (aka dirty read): A transaction T1 executing under this isolation level can access data changed by concurrent transaction(s).

Pros: No read locks needed to read data (i.e. no reader/writer blocking). Note, T1 still takes transaction duration locks for any data modified.

Cons: Data is not guaranteed to be transactionally consistent.

Read Committed: A transaction T1 executing under this isolation level can only access committed data. Pros: Good compromise between concurrency and consistency.

Cons: Locking and blocking. The data can change when accessed multiple times within the same transaction.

Repeatable Read: A transaction T1 executing under this isolation level can only access committed data with an additional guarantee that any data read cannot change (i.e. it is repeatable) for the duration of the transaction.

Pros: Higher data consistency.

Cons: Locking and blocking. The S locks are held for the duration of the transaction that can lower the concurrency. It does not protect against phantom rows.

Serializable: A transaction T1 executing under this isolation level provides the highest data consistency including elimination of phantoms but at the cost of reduced concurrency. It prevents phantoms by taking a range lock or table level lock if range lock can't be acquired (i.e. no index on the predicate column) for the duration of the transaction.

Pros: Full data consistency including phantom protection.

Cons: Locking and blocking. The S locks are held for the duration of the transaction that can lower the concurrency.

References:

<https://blogs.msdn.microsoft.com/sqlcat/2011/02/20/concurrency-series-basics-of-transaction-isolation-levels/>

NEW QUESTION 3

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 reporting database that includes a non-partitioned fact table named Fact_Sales. The table is persisted on disk.

Users report that their queries take a long time to complete. The system administrator reports that the table takes too much space in the database. You observe that there are no indexes defined on the table, and many columns have repeating values.

You need to create the most efficient index on the table, minimize disk storage and improve reporting query performance.

What should you do?

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

Answer: D

Explanation: The columnstore index is 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, and up to 10x data compression over the uncompressed data size.

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

NEW 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 is independent of the other questions in the series. Information and details provided in a question apply only to that question.

You are developing an application to track customer sales.

You need to create an object that meets the following requirements:

- Run managed code packaged in an assembly that was created in the Microsoft.NET Framework and uploaded in Microsoft SQL Server.
- Run within a transaction and roll back if a failure occurs.

- Run when a table is created or modified. What should you create?

- A. extended procedure
- B. CLR procedure
- C. user-defined procedure
- D. DML trigger
- E. scalar-valued function
- F. table-valued function
- G. DDL trigger

Answer: B

Explanation: The common language runtime (CLR) is the heart of the Microsoft .NET Framework and provides the execution environment for all .NET Framework code. Code that runs within the CLR is referred to as managed code.

With the CLR hosted in Microsoft SQL Server (called CLR integration), you can author stored procedures, triggers, user-defined functions, user-defined types, and user-defined aggregates in managed code. Because managed code compiles to native code prior to execution, you can achieve significant performance increases in some scenarios.

NEW QUESTION 5

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.

The Account table was created by using the following Transact-SQL statement:

```
CREATE TABLE Account
(
    AccountNumber int NOT NULL,
    ProductCode char(2) NOT NULL,
    Status tinyint NOT NULL,
    OpenDate date NOT NULL,
    CloseDate date,
    Balance decimal(15,2),
    AvailableBalance decimal(15,2)
);
```

There are more than 1 billion records in the Account table. The Account Number column uniquely identifies each account. The ProductCode column has 100 different values. The values are evenly distributed in the table. Table statistics are refreshed and up to date.

You frequently run the following Transact-SQL SELECT statements:

```
SELECT ProductCode, SUM(Balance) AS TotalSUM FROM Account WHERE ProductCode
<> 'CD' GROUP BY ProductCode;
SELECT AccountNumber, Balance FROM Account WHERE Production = 'CD'
```

You must avoid table scans when you run the queries. You need to create one or more indexes for the table. Solution: You run the following Transact-SQL statement:

```
CREATE CLUSTERED INDEX PK_Account On Account(AccountNumber);
CREATE NONCLUSTERED INDEX IX_Account_ProductCode On Account(ProductCode)
INCLUDE (Balance);
```

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

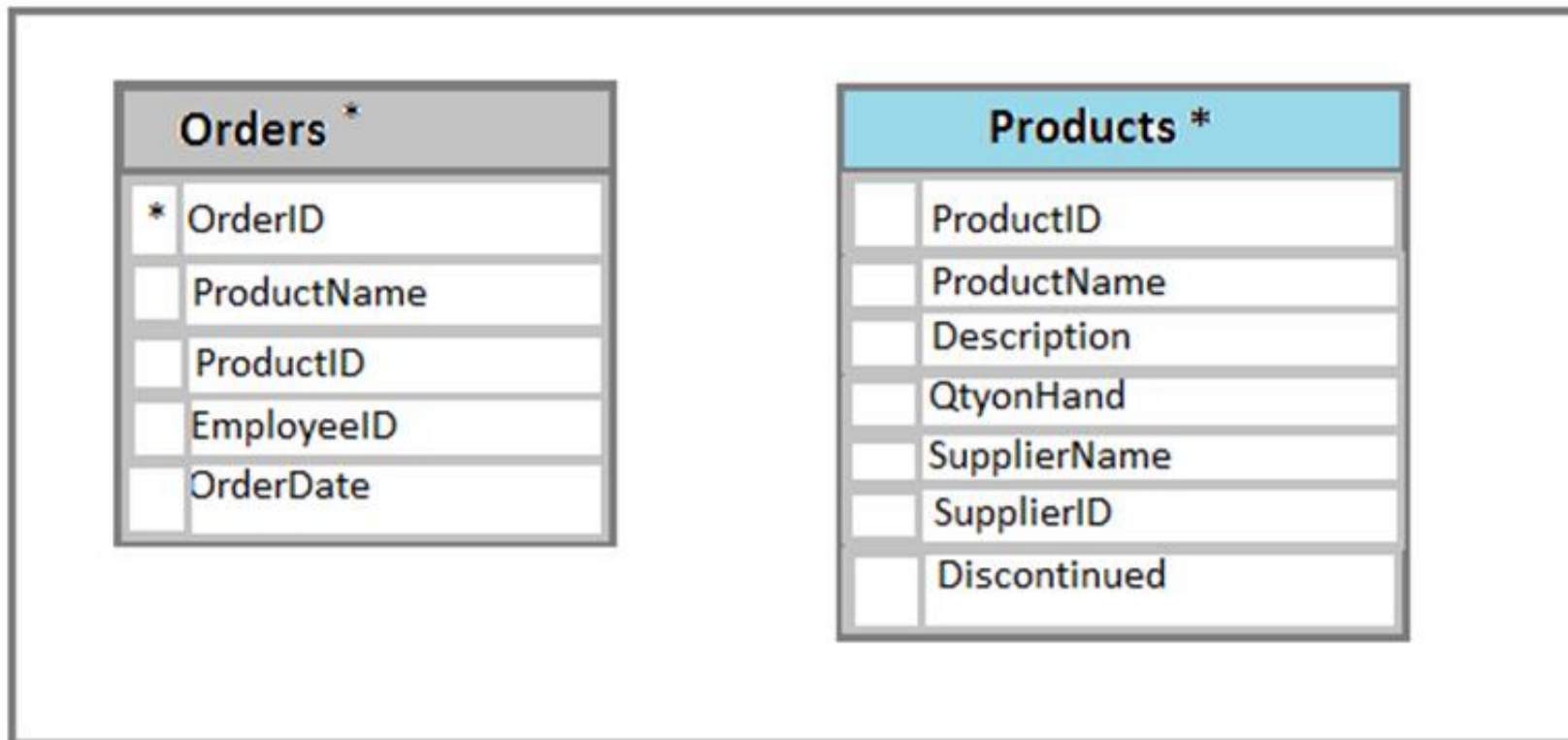
Explanation: Create a clustered index on the AccountNumber column as it is unique. Create a nonclustered index that includes the ProductCode column.

References: <https://msdn.microsoft.com/en-us/library/ms190457.aspx>

NEW QUESTION 6

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 the series.

You have a database named Sales that contains the following database tables. Customer, Order, and Products. The Products table and the order table shown in the following diagram.



The Customer table includes a column that stores the date for the last order that the customer placed.

You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.

You need to begin to modify the table design to adhere to third normal form.

Which column should you remove for each table? To answer? drag the appropriate column names to the correct locations. Each column name may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Columns

ProductID

ProductName

Description

EmployeeID

OrderDate

SupplierName

SupplierID

Discontinued

Answer Area

	Table	Column to remove
Products		Column
Orders		Column

Answer:

Explanation: In the Products table the SupplierName is dependent on the SupplierID, not on the ProductID. In the Orders table the ProductName is dependent on the ProductID, not on the OrderID. Note:

A table is in third normal form when the following conditions are met:

* It is in second normal form.

* All non-primary fields are dependent on the primary key.

Second normal form states that it should meet all the rules for First 1Normal Form and there must be no partial dependencies of any of the columns on the primary key.

First normal form (1NF) sets the very basic rules for an organized database:

* Define the data items required, because they become the columns in a table. Place related data items in a table.

* Ensure that there are no repeating groups of data.

* Ensure that there is a primary key.

References: <https://www.tutorialspoint.com/sql/third-normal-form.htm>

NEW QUESTION 7

You have two database tables. Table1 is a partitioned table and Table 2 is a non-partitioned table.

Users report that queries take a long time to complete. You monitor queries by using Microsoft SQL Server Profiler. You observe lock escalation for Table1 and Table 2.

You need to allow escalation of Table1 locks to the partition level and prevent all lock escalation for Table2. Which Transact-SQL statement should you run for each table? To answer, drag the appropriate Transact-SQL statements to the correct tables. Each command may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Transact-SQL statements

DBCC TRACEON(1211, -1)

DBCC TRACEON(1224, -1)

ALTER TABLE Table1 SET (LOCK_ESCALATION - DISABLE)

ALTER TABLE Table2 SET (LOCK_ESCALATION - DISABLE)

ALTER TABLE Table1 SET (LOCK_ESCALATION - AUTO)

ALTER TABLE Table2 SET (LOCK_ESCALATION - AUTO)

ALTER TABLE Table1 SET (LOCK_ESCALATION - TABLE)

ALTER TABLE Table2 SET (LOCK_ESCALATION - TABLE)

Answer Area

Table	Transaction
Table1	Transact-SQL statement
Table2	Transact-SQL statement

Answer:

Explanation: Since SQL Server 2008 you can also control how SQL Server performs the Lock Escalation – through the ALTER TABLE statement and the property LOCK_ESCALATION. There are 3 different options available:

Box 1: Table1, Auto

The default option is TABLE, means that SQL Server *always* performs the Lock Escalation to the table level –even when the table is partitioned. If you have your table partitioned, and you want to have a Partition Level Lock Escalation (because you have tested your data access pattern, and you don't cause deadlocks with it), then you can change the option to AUTO. AUTO means that the Lock Escalation is performed to the partition level, if the table is partitioned, and otherwise to the table level.

Box 2: Table 2, DISABLE

With the option DISABLE you can completely disable the Lock Escalation for that specific table.

For partitioned tables, use the LOCK_ESCALATION option of ALTER TABLE to escalate locks to the HoBT level instead of the table or to disable lock escalation.

References:

<http://www.sqlpassion.at/archive/2014/02/25/lock-escalations/>

NEW QUESTION 8

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;
```

You need to create one nonclustered na index that contains all of the columns in the above table. You must minimize index

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

Answer: D

NEW QUESTION 9

You have a database named Sales.

You need to create a table named Customer that includes the columns described in the following table:

Column Name	Description	Masking requirement
CustomerID	An integer primary key for the customer record	None
FirstName	A string value that stored the first name of the customer	None
LastName	A string value that stored the last name of the customer	None
CreditLimit	A monetary value that stored the customer's credit limit	The entire field must be masked with a value between 1 and 100.
MobileNo	A ten-digit string value that stored the mobile number of the customer	The entire field must be masked with the exception of the first three digits.
Email	A string value that stored the email address of the customer	The entire field must be masked except the first letter of the user name, the @ symbol, and the top-level domain.

How should you complete the Transact SQL statement? To answer, select the appropriate Transact-SQL segments in the answer area.

Answer Area

```
CREATE TABLE Customer
```

```
(
```

```
    CustomerID int IDENTITY PRIMARY KEY,
```

```
    FirstName varchar(100),
```

```
    LastName varchar(100) NOT NULL,
```

```
    CreditLimit money
```

MASKED WITH (FUNCTION = 'partial(1, "000", 100)')

MASKED WITH (FUNCTION = 'default()')

MASKED WITH (FUNCTION = 'random(1,100)')

```
    MobileNO varchar(10)
```

MASKED WITH (FUNCTION = 'partial(3, "XXXXXXX", 0)')

MASKED WITH (FUNCTION = 'default()')

MASKED WITH (FUNCTION = 'random(1,3)')

NULL,

```
    Email varchar(100)
```

MASKED WITH (FUNCTION = 'email()')

MASKED WITH (FUNCTION = 'default()')

MASKED WITH (FUNCTION = 'partial(1, "XXXXXXX", 4)')

MASKED WITH (FUNCTION = 'partial(1, "@", 4)')

NULL,

```
);
```

Answer:

Explanation: Box 1: MASKED WITH (FUNCTION = 'default()')

The Default masking method provides full masking according to the data types of the designated fields. Example column definition syntax: Phone# varchar(12)

MASKED WITH (FUNCTION = 'default()') NULL Box 2: MASKED WITH (FUNCTION = 'partial(3, "XXXXXX", 0)')

The Custom String Masking method exposes the first and last letters and adds a custom padding string in the middle. prefix,[padding],suffix examples:

PhoneNumber varchar(10) MASKED WITH (FUNCTION = 'partial(5, "XXXXXX", 0)') Box 3: MASKED WITH (FUNCTION = 'email()')

The Email masking method which exposes the first letter of an email address and the constant suffix ".com", in the form of an email address. .aXXX@XXX.com.

Example definition syntax: Email varchar(100) MASKEDWITH (FUNCTION = 'email()') NULL References: <https://msdn.microsoft.com/en-us/library/mt130841.aspx>

NEW QUESTION 10

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

You are developing an application to track customer sales.

You need to create a database object that meets the following requirements:

- Return a value of 0 if data is inserted successfully into the Customers table.
- Return a value of 1 if data is not inserted successfully into the Customers table.
- Support logic that is written by using managed code.
- Support TRY...CATCH error handling.

What should you create?

- A. extended procedure
- B. CLR procedure
- C. user-defined procedure
- D. DML trigger
- E. DDL trigger
- F. scalar-valued function
- G. table-valued function

Answer: B

Explanation: DML triggers is a special type of stored procedure that automatically takes effect when a data manipulation language (DML) event takes place that affects the table or view defined in the trigger. DML events include INSERT, UPDATE, or DELETE statements. DML triggers can be used to enforce business rules and data integrity, query other tables, and include complex Transact-SQL statements.

A CLR trigger is a type of DDL trigger. A CLR Trigger can be either an AFTER or INSTEAD OF trigger. A CLR trigger can also be a DDL trigger. Instead of executing a Transact-SQL stored procedure, a CLR trigger executes one or more methods written in managed code that are members of an assembly created in the .NET Framework and uploaded in SQL Server.

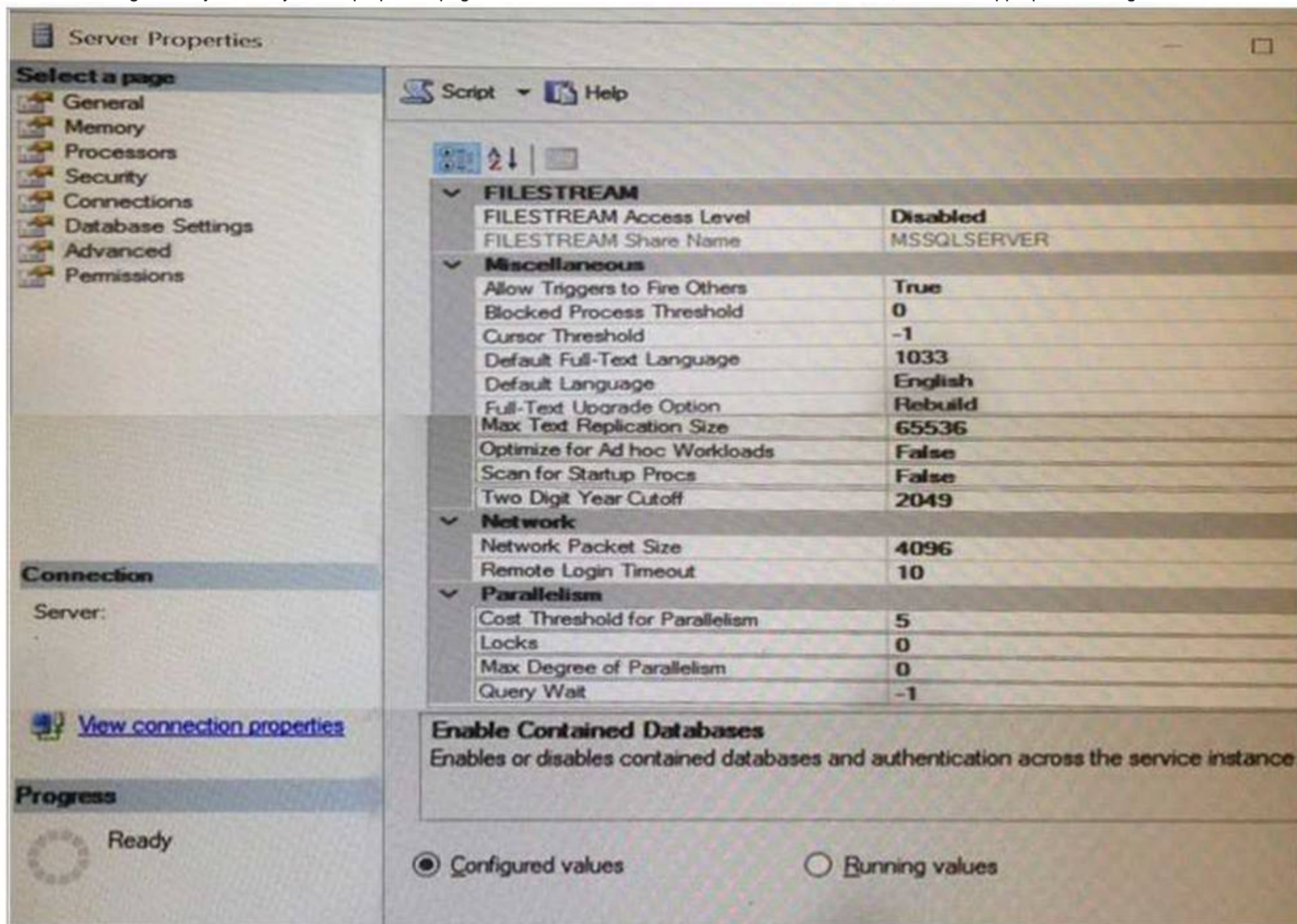
References: <https://msdn.microsoft.com/en-us/library/ms178110.aspx>

NEW QUESTION 10

You are reviewing the execution plans in the query plan cache. You observe the following:

- There are a large number of single use plans.
- There are a large number of simple execution plans that use multiple CPU cores. You need to configure the server to optimize query plan execution.

Which two setting should you modify on the properties page for the Microsoft SQL Server instance? To answer, select the appropriate settings in the answer area.



Answer:

Explanation: * Optimize for ad hoc workloads

The optimize for ad hoc workloads option is used to improve the efficiency of the plan cache for workloads that contain many single use ad hoc batches. When this option is set to 1, the Database Engine stores a small compiled plan stub in the plan cache when a batch is compiled for the first time, instead of the full compiled plan. This helps to relieve memory pressure by not allowing the plan cache to become filled with compiled plans that are not reused.

* Cost Threshold for Parallelism

Use the cost threshold for parallelism option to specify the threshold at which Microsoft SQL Server creates and runs parallel plans for queries. SQL Server creates and runs a parallel plan for a query only when the estimated cost to run a serial plan for the same query is higher than the value set in cost threshold for parallelism. The cost refers to an estimated elapsed time in seconds required to run the serial plan on a specific hardware configuration.

5 means 5 seconds, but is 5 seconds on a machine internal to Microsoft from some time in the 1990s. There's no way to relate it to execution time on your current machine, so we treat it as a pure number now. Raising it to 50 is a common suggestion nowadays, so that more of your simpler queries run on a single thread.

NEW QUESTION 13

You use Query Store to optimize a query in a database. The query has two execution plans:

Plan 2 is shown in the Plan 2 Execution Plan exhibit.

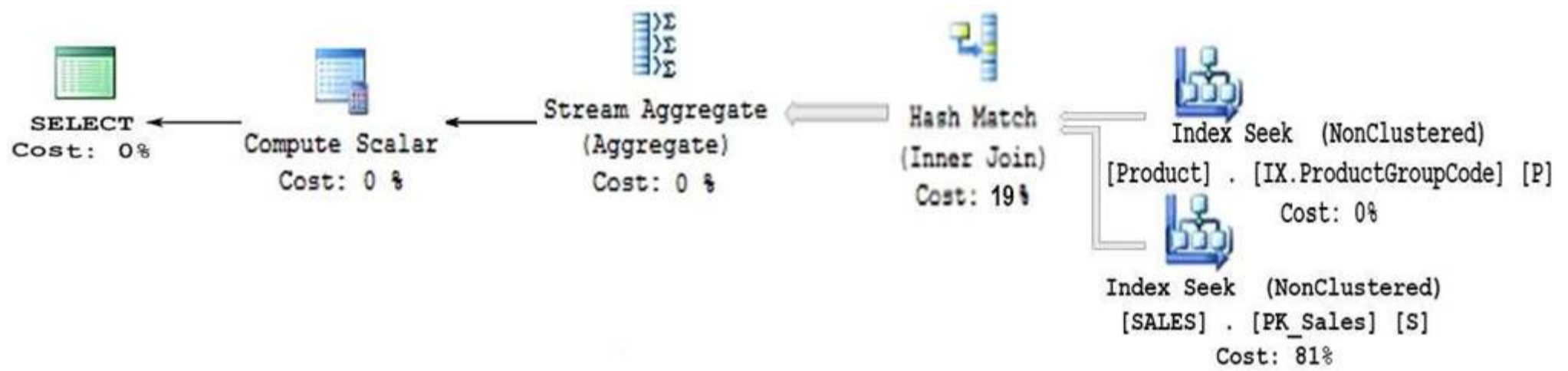
Plan 10 is shown in the Plan 10 Execution Plan exhibit.

You create an index at 22:24 based on the missing index suggestion in Plan 2.

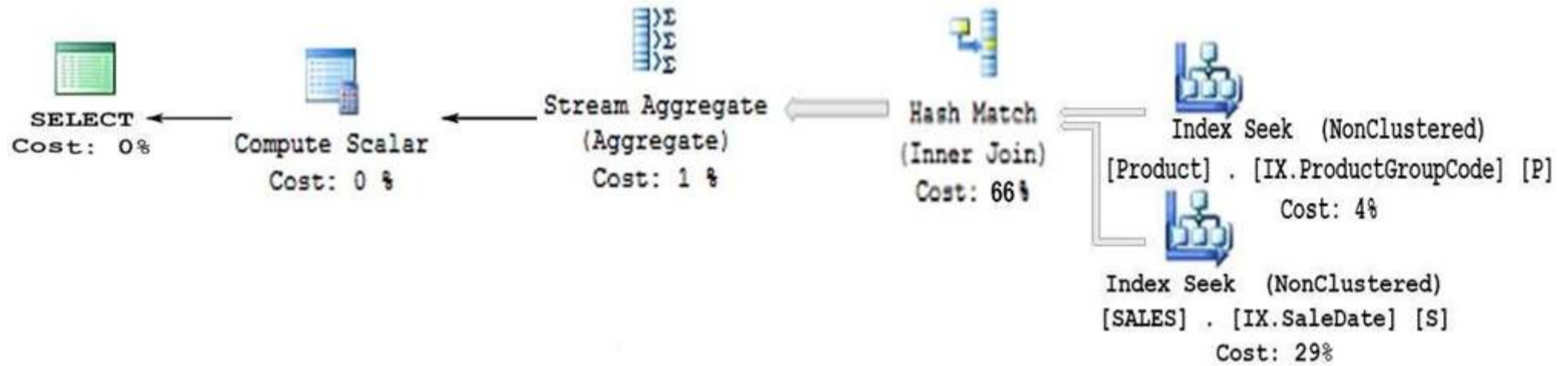
The average duration statistics for the query is shown in the Tracked Queries exhibit. (Click the Exhibit button.)

You need to analyze the operators in the two execution plans.

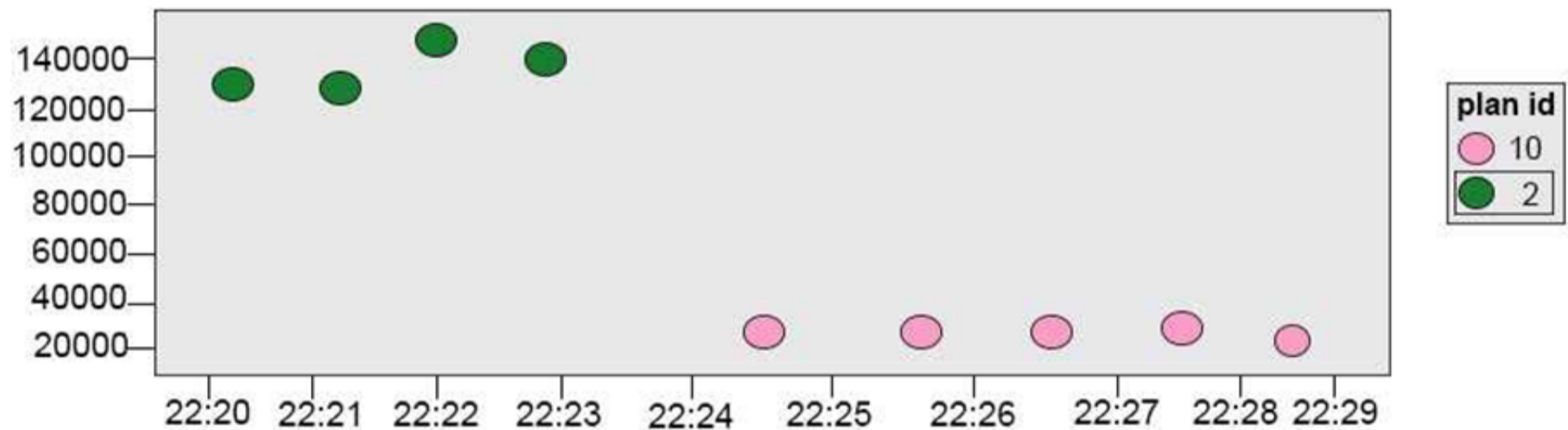
For each of the following statements, select Yes if the statement is true. Otherwise, select No. Plan 2 Execution Plan



Plan 10 Execution Plan



Tracked Queries



Answer Area

	Yes	No
The Hash Match operator in Plan 10 processed more data than the operator in Plan 2	<input type="radio"/>	<input type="radio"/>
The Clustered Index Scan operator in Plan 2 was less efficient than the Nonclustered Index Seek operator in Plan 10	<input type="radio"/>	<input type="radio"/>
Query throughput will increase by using Plan 10 instead of Plan 2	<input type="radio"/>	<input type="radio"/>

Answer:

Explanation:

Answer Area

The Hash Match operator in Plan 10 processed more data than the operator in Plan 2

The Clustered Index Scan operator in Plan 2 was less efficient than the Nonclustered Index Seek operator in Plan 10

Query throughput will increase by using Plan 10 instead of Plan 2

Yes

No

☒
☐
☒
☐
☐
☒

NEW QUESTION 15

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 have a database that contains a table named Employees. The table stores information about the employees of your company.

You need to implement and enforce the following business rules:

- Limit the values that are accepted by the Salary column.
- Prevent salaries less than \$15,000 and greater than \$300,000 from being entered.
- Determine valid values by using logical expressions.
- Do not validate data integrity when running DELETE statements. Solution: You implement a FOR UPDATE trigger on the table. Does the solution meet the goal?

A. Yes

B. No

Answer: B

Explanation: References:

<http://stackoverflow.com/questions/16081582/difference-between-for-update-of-and-for-update>

NEW QUESTION 19

You have a database named MyDatabase. You must monitor all the execution plans in XML format by using Microsoft SQL Trace. The trace must meet the following requirements:

- Capture execution plans only for queries that run the MyDatabase database.
- Filter out plans with event duration of less than or equal to 100 microseconds.
- Save trace results to a disk on the server. You need to create the trace.

In which order should you arrange the Transact-SQL segments to develop the solution? To answer, move all Transact-SQL segments to the answer area and arrange them in the correct order.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

DECLARE @traceEventId int = 122;

DECLARE @traceColumnIdForTextData int = 1; DECLARE @durationFilter bigint = 100 DECLARE @databaseId int;

SELECT @databaseId = DB_ID('MyDatabase');

Transact-SQL segments

```
EXEC sp_trace_setfilter @TraceID, 13, 0, 2, @duration-
Filter;
EXEC sp_trace_setfilter @TraceID, 3, 0, 0, @databaseId;
```

```
EXEC sp_trace_setevent @TraceID, @traceEventId, 1, 1;
EXEC sp_trace_setevent @TraceID, @traceEventId, 12, 1;
EXEC sp_trace_setevent @TraceID, @traceEventId, 13, 1;
```

```
EXEC sp_trace_setstatus @TraceID, 1;
```

```
EXEC sp_trace_create @TraceID OUTPUT, 2, @fileName, @max-
filesize, NULL @fileCount;
```

```
DECLARE @ traceID int;
DECLARE @maxFileSize bigint = 20;
DECLARE @fileCount INT = 15;
DECLARE @fileName NVARCHAR(245) = 'D:\SQL Trace\TraceRe
sult;
DECLARE @traceEventId int = 122;
DECLARE @traceColumnIdForTextData int = 1;
DECLARE @durationFilter bigint = 100;
DECLARE @databaseID int;
('SELECT @databaseId = DB_ID('MyDatabase');
```

Answer Area



Answer:

Explanation: The following system stored procedures are used to define and manage traces:

- * sp_trace_create is used to define a trace and specify an output file location as well as other options that I'll cover in the coming pages. This stored procedure returns a handle to the created trace, in the form of an integer trace ID.
 - * sp_trace_setevent is used to add event/column combinations to traces based on the trace ID, as well as to remove them, if necessary, from traces in which they have already been defined.
 - * sp_trace_setfilter is used to define event filters based on trace columns.
 - * sp_trace_setstatus is called to turn on a trace, to stop a trace, and to delete a trace definition once you're done with it. Traces can be started and stopped multiple times over their lifespan.
- References: <https://msdn.microsoft.com/en-us/library/cc293613.aspx>

NEW QUESTION 23

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.

You need to ensure that the performance of each instance is consistent for the same queried and query plans. 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.

Answer: H

Explanation: Advanced Viewing of Target Data from Extended Events in SQL Server

When your event session is currently active, you might want to watch the event data in real time, as it is received by the target.

Management > Extended Events > Sessions > [your-session] > Watch Live Data.

The query_post_execution_showplan extended event enables you to see the actual query plan in the SQL Server Management Studio (SSMS) UI. When the Details pane is visible, you can see a graph of the query plan on the Query Plan tab. By hovering over a node on the query plan, you can see a list of property names and their values for the node.

The screenshot displays the SQL Server Management Studio (SSMS) interface. The main window shows a query plan for 'Query 1: Query cost (relative to the batch): 100%'. The query plan is a 'Nested Loops (Left Outer Join)' with a 'Clustered Index Seek (Clustered)' node highlighted. The 'Properties' pane on the right shows details for the 'Clustered Index Seek (Clustered)' node.

Clustered Index Seek (Clustered)	
Scanning a particular range of rows from a clustered index.	
Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Actual Execution Mode	Row
Estimated Execution Mode	Row
Storage	RowStore
Actual Number of Rows	0
Actual Number of Batches	0
Estimated Operator Cost	0.0033263 (4%)
Estimated I/O Cost	0.003125
Estimated Subtree Cost	0.0033263
Estimated CPU Cost	0.0001581
Estimated Number of Executions	1.272901
Number of Executions	1
Estimated Number of Rows	1
Estimated Row Size	139 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	228
Object [InMemTest2].[sys].[sysclsobjs].[clst] [s]	
Output List [InMemTest2].[sys].[sysclsobjs].name	
Seek Predicates Seek Keys[1]: Prefix: [InMemTest2].[sys].[sysclsobjs].class, [InMemTest2].[sys].[sysclsobjs].id = Scalar Operator((50)), Scalar Operator([InMemTest2].[sys].[sysclsobjs].[nsid] as [s], [nsid])	

References: <https://msdn.microsoft.com/en-us/library/mt752502.aspx>

NEW QUESTION 24

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 observe that many deadlocks appear to be happening during specific times of the day.
You need to monitor the SQL environment and capture the information about the processes that are causing the deadlocks.
What should you do?

- A. Create a sys.dm_os_waiting_tasks query.
- B. Create a sys.dm_exec_sessions query.
- C. Create a PerformanceMonitor 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.

Answer: F

Explanation: To view deadlock information, the Database Engine provides monitoring tools in the form of two trace flags, and the deadlock graph event in SQL Server Profiler.

Trace Flag 1204 and Trace Flag 1222

When deadlocks occur, trace flag 1204 and trace flag 1222 return information that is captured in the SQL Server error log. Trace flag 1204 reports deadlock information formatted by each node involved in the deadlock. Trace flag 1222 formats deadlock information, first by processes and then by resources. It is possible to enable both trace flags to obtain two representations of the same deadlock event.

References: [https://technet.microsoft.com/en-us/library/ms178104\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms178104(v=sql.105).aspx)

NEW QUESTION 26

You have a database that users query frequently.

The users report that during peak business hours, the queries take longer than expected to execute.

A junior database administrator uses Microsoft SQL Server Profiler on the database server to trace the session activities.

While performing the trace, the performance of the database server worsens, and the server crashes.

You need to recommend a solution to collect the query run times. The solution must minimize the impact on the resources of the database server.

What should you recommend?

- A. Increase the free space on the system drive of the database server, and then use SQL Server Profiler on the server to trace the session activities.
- B. Collect session activity data by using SQL Server Extended Events.
- C. Clean up tempdb, and then use SQL Server Profiler on the datafile server to trace the session activities.
- D. Collect performance data by using a Data Collector Set (DCS) in Performance Monitor.

Answer: A

NEW QUESTION 28

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 have a database named DB1 that includes a table named sales . Orders. You grant a user named User1 select permissions on the sales schema.

You need to ensure that User1 can select data from the sales, orders table without specifying the schema name in any Transact SQL statements.

Solution: You move the sales.orders table to the dbo schema. Does the solution meet the goal?

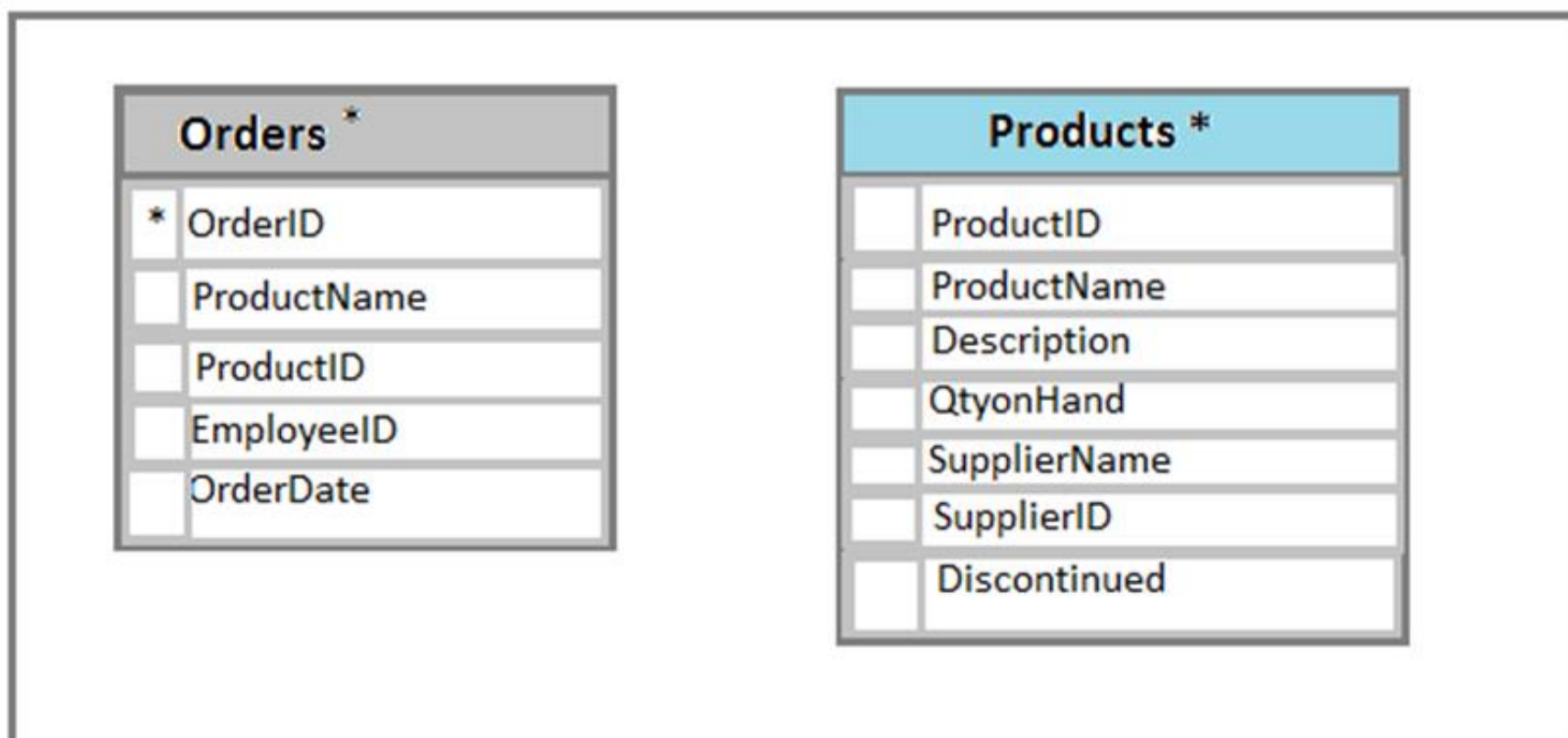
- A. Yes
- B. No

Answer: B

NEW QUESTION 32

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 Sales that contains the following database tables: Customer, Order, and Products. The Products table and the Order table are shown in the following diagram.



The customer table includes a column that stores the data for the last order that the customer placed.

You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.

Changes to the price of any product must be less a 25 percent increase from the current price. The shipping department must be notified about order and shipping details when an order is entered into the database.

You need to implement the appropriate table objects.

Which object should you use for each table? To answer, drag the appropriate objects to the correct tables. Each object may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Objects		Answer Area	
		Table	Objects
Foreign key constraint	Instead of trigger	Orders	
Check constraint	Primary key constraint	Products	
Unique constraint	After insert trigger		

Answer:

Explanation: The Products table needs a primary key constraint on the ProductID field.

The Orders table needs a foreign key constraint on the ProductID field, with a reference to the ProductID field in the Products table.

NEW QUESTION 34

You have a database that contains three encrypted store procedures named dbo.Proc1, dbo.Proc2 and dbo.Proc3. The stored procedures include INSERT, UPDATE, DELETE and BACKUP DATABASE statements.

You have the following requirements:

- * You must run all the stored procedures within the same transaction.
- * You must automatically start a transaction when stored procedures include DML statements.
- * You must not automatically start a transaction when stored procedures include DDL statements. You need to run all three stored procedures.

Which four Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments to the answer area and arrange them in the correct order.

Transact-SQL segments

```
BEGIN CATCH
IF (XACT_STATE() != 0)
    ROLLBACK TRANSACTION
END CATCH
```

```
IF (@TRANCOUNT > 0)
    ROLLBACK TRANSACTION
```

```
BEGIN TRAN
```

```
EXEC dbo.Proc1
EXEC dbo.Proc2
EXEC dbo.Proc3
```

```
SET IMPLICIT_TRANSACTIONS OFF
```

```
SET IMPLICIT_TRANSACTIONS ON
```

```
COMMIT TRANSACTION
```

```
BEGIN TRY
    EXEC dbo.Proc1
    EXEC dbo.Proc2
    EXEC dbo.Proc3
    IF (XACT_STATE() = 1)
        COMMIT TRANSACTION;
END TRY
```

Answer Area



Answer:

Explanation: Note:

Implicit transaction mode remains in effect until the connection executes a SET IMPLICIT_TRANSACTIONS OFF statement, which returns the connection to

autocommit mode. In autocommit mode, all individual statements are committed if they complete successfully.

When a connection is in implicit transaction mode and the connection is not currently in a transaction, executing any of the following statements starts a transaction:

Note 2: XACT_STATE returns the following values.

1 The current request has an active user transaction. The request can perform any actions, including writing data and committing the transaction. The transaction is committable.

-1 The current request has an active user transaction, but an error has occurred that has caused the transaction to be classified as an uncommittable transaction. The transaction is uncommittable and should be rolled back.

0 There is no active user transaction for the current request. A commit or rollback operation would generate an error.

References:

[https://technet.microsoft.com/en-us/library/ms187807\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms187807(v=sql.105).aspx) [https://technet.microsoft.com/en-us/library/ms189797\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/ms189797(v=sql.110).aspx)

NEW QUESTION 39

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.

Your company has employees in different regions around the world.

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

- Employee ID
- date and time employee checked in to work
- 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,
    DateChekedIn datetimeoffset(0) NOT NULL,
    DateCheclOut datetimeoffset(0) NOT NULL)
```

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation: Datetimeoffset defines a date that is combined with a time of a day that has time zone awareness and is based on a 24-hour clock.

Syntax: datetimeoffset [(fractional seconds precision)]

Forthe use"datetimeoffset(0)", the Fractional seconds precision is 0, which is required here. References: <https://msdn.microsoft.com/en-us/library/bb630289.aspx>

NEW QUESTION 41

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 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 tables described in the following table:

Table name	Description
TBL1	<ul style="list-style-type: none"> Column1 is configured as the primary key. The table will contain 20 million records. The table will contain historical data. Most queries of TBL1 return a high percentage of rows from the table with aggregates.
TBL2	<ul style="list-style-type: none"> Column1 is configured as the primary key. The table will contain 25 million records. The frequency of updates and deletes to records in TBL2 is low. Most queries of TBL2 return a low percentage of rows and a high percentage of columns.

There are no foreign key relationships between TBL1 and TBL2.

You need to minimize the amount of time required for queries that use data from TBL1 and TBL2 to return data.

What should you do?

- 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 columns 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 inde
- F. Create a nonclustered columnstore index on TBL1. Create noncluster index on TBL2.
- G. Drop existing indexes on TBL1 and then create a clustered columnstore inde
- H. Create a nonclustered columnstore index on TBL1 Make no changes to TBL2.

Answer: B

NEW QUESTION 42

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.

The Account table was created by using the following Transact-SQL statement:

```
CREATE TABLE Account
(
    AccountNumber int NOT NULL,
    ProductCode char(2) NOT NULL,
    Status tinyint NOT NULL,
    OpenDate date NOT NULL,
    CloseDate date,
    Balance decimal(15,2),
    AvailableBalance decimal(15,2)
);
```

There are more than 1 billion records in the Account table. The Account Number column uniquely identifies each account. The ProductCode column has 100 different values. The values are evenly distributed in the table. Table statistics are refreshed and up to date.

You frequently run the following Transact-SQL SELECT statements:

```
SELECT ProductCode, SUM(Balance) AS TotalSUM FROM Account WHERE ProductCode
<> 'CD' GROUP BY ProductCode;
SELECT AccountNumber, Balance FROM Account WHERE Production = 'CD'
```

You must avoid table scans when you run the queries. You need to create one or more indexes for the table. Solution: You run the following Transact-SQL statement:

CREATE NONCLUSTERED INDEX IX_Account_ProductCode ON Account(ProductCode); Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation: References: <https://msdn.microsoft.com/en-za/library/ms189280.aspx>

NEW QUESTION 44

You need to build a function that meets the following requirements:

How should you complete the Transact-SQL statement? To answer, drag the appropriate Transact-SQL statements to the correct locations. Each Transact-SQL segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Transact-SQL segments

RETURNS TABLE AS

```
BEGIN
INSERT INTO @Prices
SELECT ListPrice FROM Product
WHERE ProductId = @ProductId
RETURN
END
```

RETURNS @Prices TABLE (ListPrice
money) AS

```
RETURN (
SELECT ListPrice FROM Product
WHERE ProductId = @ProductId
)
```

RETURNS [Money] AS

```
BEGIN
SELECT @ListPrice = ListPrice FROM Product
WHERE ProductId = @ProductId
RETURN @ListPrice
END
```



Answer area

```
CREATE FUNCTION dbo.GetPrice (@ProductId int)
```

Transact-SQL segment

Transact-SQL segment

Answer:

Explanation: References: [https://technet.microsoft.com/en-us/library/ms189294\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms189294(v=sql.105).aspx)

NEW QUESTION 45

You are developing an app that allows users to query historical company financial data. You are reviewing email messages from the various stakeholders for a project.

The message from the security officer is shown in the Security Officer Email exhibit below. TO: Database developer

From: Security Officer

Subject: SQL object requirements

We need to simplify the security settings for the SQL objects. Having a assign permissions at every object in SQL is tedious and leads to a problem.

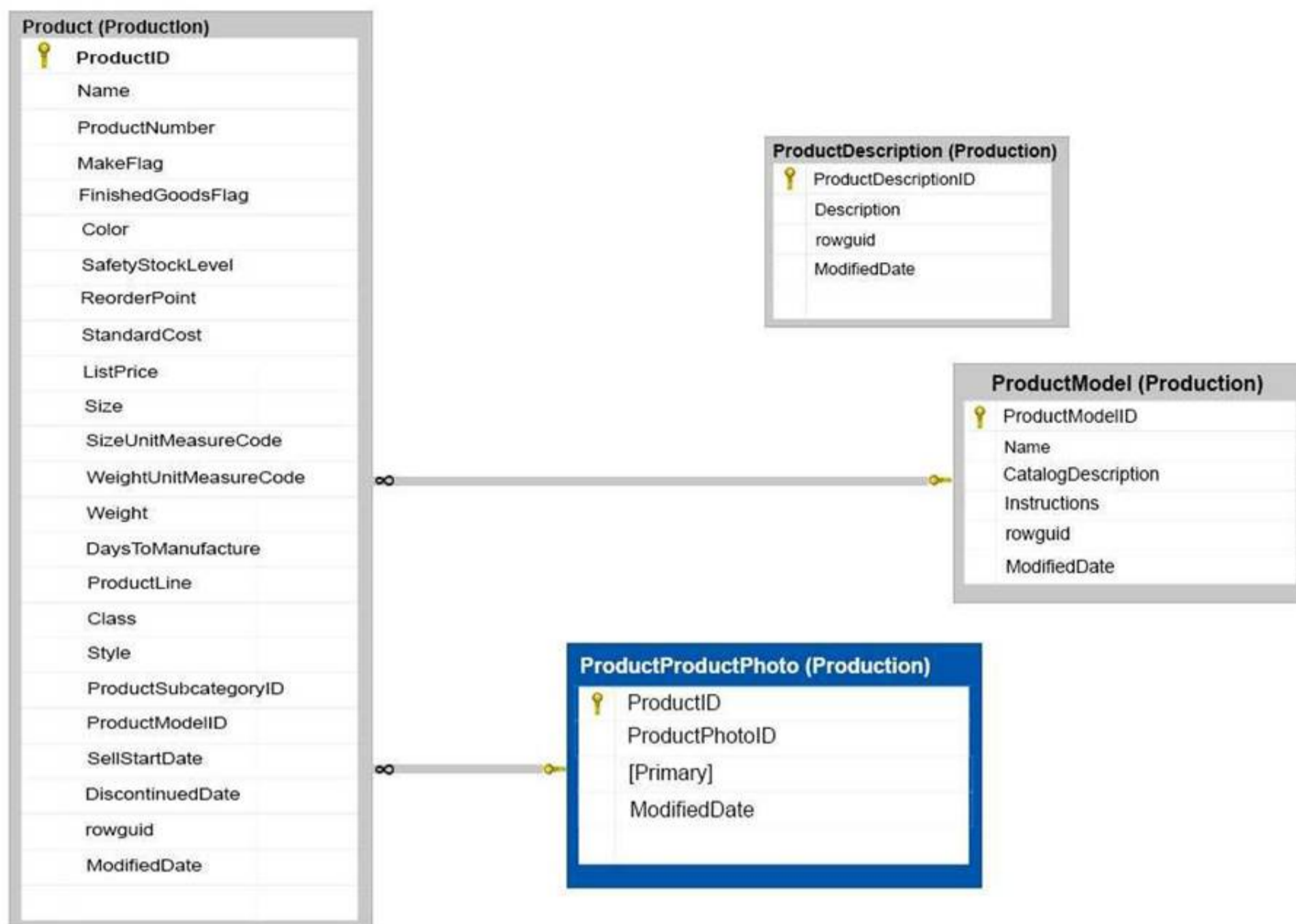
Documentation is also more difficult when we have to assign permissions at multiple levels. We need to assign the required permissions at one object, even though that object may be obtaining from other objects.

The message from the sales manager is shown in the Sales Manager Email exhibit below. TO: Database developer

From: Sales Manager Subject: Needed SQL objects

When creating objects for our use, they need to be flexible. We will be changing the base infrastructure frequently. We need components in SQL that will provide backward compatibility to our front end applications as the environments change so that do not need to modify the front end applications. We need objects that can provide a filtered set of the data. The data may be coming from multiple tables and we need an object that can provide access to all of the data through a single object reference.

This is an example of the types of data we need to be able to have queries against without having to change the front end applications.



The message from the web developer is shown in the Web Developer Email exhibit below. TO: Database developer

From: Web Developer

Subject: SQL Object component

Whatever you will be configuring to provide access to data in SQL, it needs to connect using the items referenced in this interface. We have been using this for a long time, and we cannot change this from end easily. Whatever objects are going to be used in SQL they must work using object types this interface references.




Database Name: 

Table Name: 

Column Name: 

You need to create one or more objects that meet the needs of the security officer, the sales manager and the web developer. For each of the following statements, select Yes if the statement is true. Otherwise, select No.

Answer Area

Yes No

You must create a stored procedure

☐
☐

You must create a trigger

☐
☐

You must create a view

☐
☐

Answer:

Explanation: * Stored procedure: Yes

A stored procedure to implement the following:

Whatever you will be configuring to provide access to data in SQL, it needs to connect using the items referenced in this interface. We have been using this for a long time, and we cannot change this from end easily. Whatever objects are going to be used in SQL they must work using object types this interface references.

* Trigger: No

No requirements are related to actions taken when changing the data.

* View: Yes

Because: We need objects that can provide a filtered set of the data. The data may be coming from multiple tables and we need an object that can provide access to all of the data through a single object reference.

NEW QUESTION 49

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.

The Account table was created by using the following Transact-SQL statement:

```
CREATE TABLE Account
(
    AccountNumber int NOT NULL,
    ProductCode char(2) NOT NULL,
    Status tinyint NOT NULL,
    OpenDate date NOT NULL,
    CloseDate date,
    Balance decimal(15,2),
    AvailableBalance decimal(15,2)
);
```

There are more than 1 billion records in the Account table. The Account Number column uniquely identifies each account. The ProductCode column has 100 different values. The values are evenly distributed in the table. Table statistics are refreshed and up to date.

You frequently run the following Transact-SQL SELECT statements:

```
SELECT ProductCode, SUM(Balance) AS TotalSUM FROM Account WHERE ProductCode
<> 'CD' GROUP BY ProductCode;
SELECT AccountNumber, Balance FROM Account WHERE Production = 'CD'
```

You must avoid table scans when you run the queries. You need to create one or more indexes for the table. Solution: You run the following Transact-SQL statement:

```
CREATE NONCLUSTERED INDEX PK_Account ON Account(AccountNumber);
CREATE NONCLUSTERED INDEX IX_Account_ProductCode ON Account(Product-
Code) INCLUDE (Balance);
```

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation: Create a clustered index on the AccountNumber column as it is unique, not a non nonclustered one. References: <https://msdn.microsoft.com/en-us/library/ms190457.aspx>

NEW QUESTION 53

You are creating the following two stored procedures:

A natively-compiled stored procedure

An interpreted stored procedure that accesses both disk-based and memory-optimized tables

Both stored procedures run within transactions.

You need to ensure that cross-container transactions are possible. Which setting or option should you use?

- A. the SET TRANSACTION_READ_COMMITTED isolation level for the connection
- B. the SERIALIZABLE table hint on disk-based tables
- C. the SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT=ON option for the database
- D. the SET MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT=OFF option for the database

Answer: C

Explanation: Provide a supported isolation level for the memory-optimized table using a table hint, such as WITH (SNAPSHOT). The need for the WITH (SNAPSHOT) hint can be avoided through the use of the database option MEMORY_OPTIMIZED_ELEVATE_TO_SNAPSHOT. When this option is set to ON, access to a memory-optimized table under a lower isolation level is automatically elevated to SNAPSHOT isolation.

NEW QUESTION 54

You have a trigger named CheckTriggerCreation that runs when a user attempts to create a trigger. The CheckTriggerCreation trigger was created with the ENCRYPTION option and additional proprietary business logic.

You need to prevent users from running the ALTER and DROP statements or the sp_tableoption stored procedure.

Which three Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

Commands

```
DISABLE TRIGGER CheckTriggerCreation ON  
DATABASE;
```

```
ENABLE TRIGGER CheckTriggerCreation ON  
DATABASE;
```

```
CREATE TRIGGER CheckTriggerCreation  
ON DATABASE  
FOR CREATE_TRIGGER  
AS  
    RAISERROR ('Error message', 10, 1)  
    ROLLBACK  
GO
```

```
CREATE TRIGGER CheckTableChanges  
ON DATABASE  
FOR ALTER_TABLE, DROP_TABLE  
AS  
    RAISERROR ('Error message', 10, 1)  
    ROLLBACK  
GO
```

```
DROP TRIGGER CheckTriggerCreation ON  
DATABASE;
```

```
CREATE TRIGGER CheckTableChanges  
ON DATABASE  
FOR DDL_TABLE_EVENTS  
AS  
    RAISERROR ('Error message', 10, 1)  
    ROLLBACK  
GO
```

Answer Area



Answer:

Explanation:

Commands

```
DISABLE TRIGGER CheckTriggerCreation ON
DATABASE;
```

```
ENABLE TRIGGER CheckTriggerCreation ON
DATABASE;
```

```
CREATE TRIGGER CheckTriggerCreation
ON DATABASE
FOR CREATE_TRIGGER
AS
    RAISERROR ('Error message', 10, 1)
    ROLLBACK
GO
```

```
CREATE TRIGGER CheckTableChanges
ON DATABASE
FOR ALTER_TABLE, DROP_TABLE
AS
    RAISERROR ('Error message', 10, 1)
    ROLLBACK
GO
```

```
DROP TRIGGER CheckTrigerCreation ON
DATABASE;
```

```
CREATE TRIGGER CheckTableChanges
ON DATABASE
FOR DDL_TABLE_EVENTS
AS
    RAISERROR ('Error message', 10, 1)
    ROLLBACK
GO
```

Answer Area

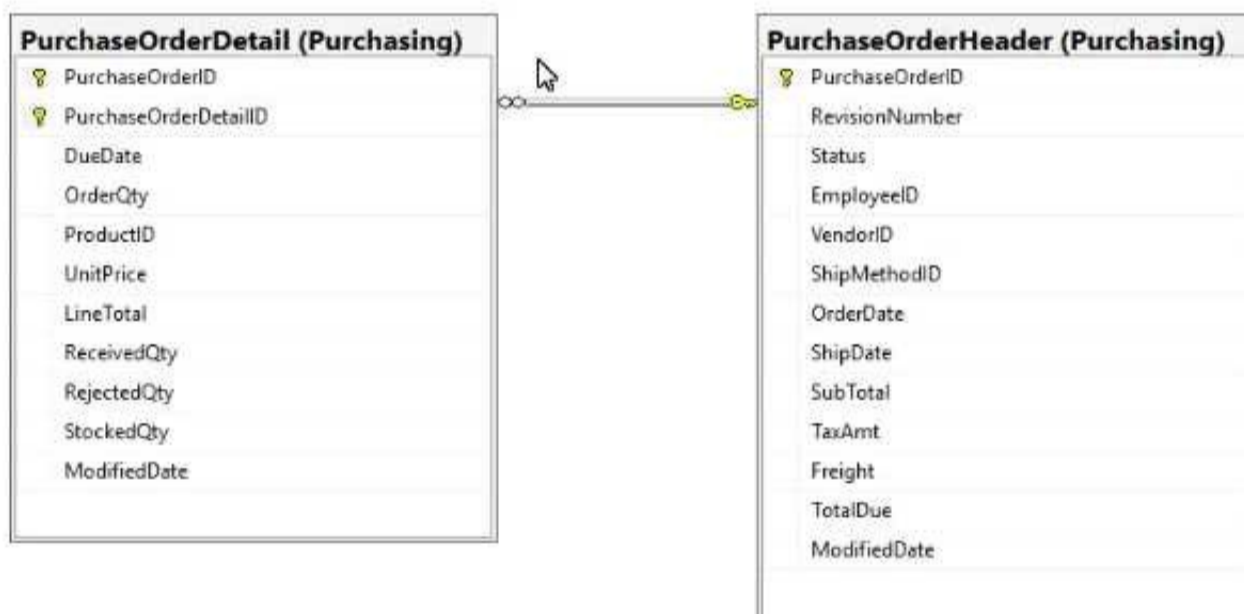
```
DISABLE TRIGGER CheckTriggerCreation ON
DATABASE;
```

```
CREATE TRIGGER CheckTableChanges
ON DATABASE
FOR DDL_TABLE_EVENTS
AS
    RAISERROR ('Error message', 10, 1)
    ROLLBACK
GO
```

```
ENABLE TRIGGER CheckTriggerCreation ON
DATABASE;
```

NEW QUESTION 58

You manage a database that includes the tables shown in the exhibit (Click the Exhibit button.)



You plan to create a DML trigger that reads the value of the LineTotal column for each row in the PurchaseOrderDetail table. The trigger must add the value obtained to the value in the SubTotal column of the PurchaseOrderHeader table.

You need to organize the list to form the appropriate Transact-SQL statement.

Which five Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

Transact-SQL segments

CREATE TRIGGER NewPODetail

UPDATE PurchaseOrderDetail

UPDATE PurchaseOrderHeader
SET SubTotal = SubTotal + LineTotal
FROM inserted

UPDATE PurchaseOrderDetail
SET SubTotal = SubTotal + LineTotal
FROM inserted

ON PurchaseOrderDetail

UPDATE PurchaseOrderHeader
SET SubTotal = LineTotal
FROM inserted

AFTER INSERT AS

WHERE PurchaseOrderHeader.PurchaseOrderID =
inserted.PurchaseOrderID

Answer area

➤
➤

⬅
⬅

⬆
⬆

Answer:

Explanation:

Answer area

1	AFTER INSERT AS
2	CREATE TRIGGER NewPODetail
3	ON PurchaseOrderDetail

NEW QUESTION 60

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 named 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	decimal(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?

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

Answer: C

Explanation: 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.

NEW QUESTION 62

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 is independent of the other questions in this series. Information and details provided in a question apply only to that question.

You have a database named DB1. There is no memory-optimized filegroup in the database.

You have a table and a stored procedure that were created by running the following Transact-SQL statements:

```
CREATE TABLE Employee
(
    EmployeeId int NOT NULL PRIMARY KEY,
    FirstName varchar(20),
    LastName varchar(20),
    Status char(1),
    Address varchar(100),
    Department int NOT NULL
);

CREATE PROCEDURE uspSelectEmployeeDetails
(
    @LastName varchar(20)
)
AS
BEGIN
    SELECT e.FirstName, e.LastName, d.DepartmentName
    FROM Employee e
    JOIN Department d on e.DepartmentId = d.DepartmentId
    WHERE e.Status = 'T' AND e.LastName = @LastName;
END;
```

The Employee table is persisted on disk. You add 2,000 records to the Employee table. You need to create an index that meets the following requirements:

- Optimizes the performance of the stored procedure.
- Covers all the columns required from the Employee table.
- Uses FirstName and LastName as included columns.
- Minimizes index storage size and index key size. What should you do?

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

Answer: B

Explanation: References: [https://technet.microsoft.com/en-us/library/jj835095\(v=sql.110\).aspx](https://technet.microsoft.com/en-us/library/jj835095(v=sql.110).aspx)

NEW QUESTION 64

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.

Your company has employees in different regions around the world.

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

- Employee ID
- date and time employee checked in to work
- 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,
    DateChekedIn datetimeoffset NOT NULL,
    DateCheclOut datetimeoffset NOT NULL)
```

Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation: Datetimeoffset, not datetimeoffset, defines a date that is combined with a time of a day that has time zone awareness and is based on a

24-hourclock.

Syntax: datetimeoffset [(fractional seconds precision)]

For the use "datetimeoffset", the Fractional seconds precision is 7. References: <https://msdn.microsoft.com/en-us/library/bb630289.aspx>

NEW QUESTION 65

You have a view that includes an aggregate.

You must be able to change the values of columns in the view. The changes must be reflected in the tables that the view uses.

You need to ensure that you can update the view. What should you create?

- A. table-valued function
- B. a schema-bound view
- C. a partitioned view
- D. a DML trigger

Answer: B

Explanation: When you use the SchemaBinding keyword while creating a view or function you bind the structure of any underlying tables or views. It means that as long as that schemabound object exists as a schemabound object (ie you don't remove schemabinding) you are limited in changes that can be made to the tables or views that it refers to.

References: <https://sqlstudies.com/2014/08/06/schemabinding-what-why/>

NEW QUESTION 70

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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and SalesPerson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```

You must modify the ProductReview Table to meet the following requirements:

- * The table must reference the ProductID column in the Product table
- * Existing records in the ProductReview table must not be validated with the Product table.
- * Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- * Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- * Create new rows in the table without granting INSERT permissions to the table.
 - * Notify the sales person who places an order whether or not the order was completed.
- You must add the following constraints to the SalesHistory table:
- * a constraint on the SaleID column that allows the field to be used as a record identifier
 - * a constant that uses the ProductID column to reference the Product column of the ProductTypes table
 - * a constraint on the CategoryID column that allows one row with a null value in the column
 - * a constraint that limits the SalePrice column to values greater than four
- Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.
- You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:
- * The table must hold 10 million unique sales orders.
 - * The table must use checkpoints to minimize I/O operations and must not use transaction logging.
 - * Data loss is acceptable.
- Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.
- You need to create a stored procedure named spDeleteCategory to delete records in the database. The stored procedure must meet the following requirements:
- * Delete records in both the BlogEntry and BlogCategory tables where CategoryId equals parameter @CategoryId.
 - * Avoid locking the entire table when deleting records from the BlogCategory table.
 - * If an error occurs during a delete operation on either table, all changes must be rolled back, otherwise all changes should be committed.
- How should you complete the procedure? To answer, select the appropriate Transact-SQL segments in the answer area.

Answer Area

```
CREATE PROCEDURE spDeleteCategory
(@CategoryId int)
AS
BEGIN
    SET NOCOUNT ON;

    SET IMPLICIT_TRANSACTIONS ON
    SET IMPLICIT_TRANSACTIONS OFF
    SET TRANSACTION ISOLATION LEVEL READ COMMITTED
    SET TRANSACTION ISOLATION LEVEL SNAPSHOT

    BEGIN TRY
        DELETE FROM BlogEntry WHERE CategoryID = @CategoryId;
        ...
        DELETE FROM BlogCategory
        WITH ( ) WHERE CategoryId = @CategoryId;

        IF @@TRANCOUNT > 0
        BEGIN
            COMMIT
        END

    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            COMMIT
            ROLLBACK
        END
    END
```

Answer:

Explanation: Box 1: SET TRANSACTION ISOLATION LEVEL READ COMMITTED

You can minimize locking contention while protecting transactions from dirty reads of uncommitted data modifications by using either of the following:

- * The READ COMMITTED isolation level with the READ_COMMITTED_SNAPSHOT database option set ON.
- * The SNAPSHOT isolation level.

With ROWLOCK we should use READ COMMITTED Box 2: ROWLOCK

Requirement: Avoid locking the entire table when deleting records from the BlogCategory table

ROWLOCK specifies that row locks are taken when page or table locks are ordinarily taken. When specified in transactions operating at the SNAPSHOT isolation level, row locks are not taken unless ROWLOCK is combined with other table hints that require locks, such as UPDLOCK and HOLDLOCK.

Box 3: COMMIT

Box 4: ROLLBACK

NEW QUESTION 72

You are developing an ETL process to cleanse and consolidate incoming data. The ETL process will use a reference table to identify which data must be cleansed in the target table. The server that hosts the tables restarts daily.

You need to minimize the amount of time it takes to execute the query and the amount of time it takes to populate the reference table.

What should you do?

- Convert the target table to a memory-optimized table
- Create a natively compiled stored procedure to cleanse and consolidate the data.
- Convert the reference table to a memory-optimized table
- Set the durability option to SCHEMA_AND_DATA
- Create a native compiled stored procedure to implement the ETL process for both tables.

- F. Convert the reference table to a memory-optimized table
G. Set the durability option to SCHEMA_ONLY.

Answer: D

NEW QUESTION 74

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 have a table that has a clustered index and a nonclustered index. The indexes use different columns from the table. You have a query named Query1 that uses the nonclustered index.

Users report that Query1 takes a long time to report results. You run Query1 and review the following statistics for an index seek operation:

Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Actual Number of Rows	3571454
Actual Number of Batches	0
Estimated I/O Cost	0.0093577
Estimated Operator Cost	0.0107304 (0%)
Estimated CPU Cost	0.0013727
Estimated Subtree Cost	0.0107304
Estimated Number of Executions	1
Number of Executions	8
Estimated Number of Rows	0
Estimated Row Size	19 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	100

You need to resolve the performance issue. Solution: You defragment both indexes. Does the solution meet the goal?

- A. Yes
B. No

Answer: B

Explanation: We see Actual Number of Row is 3571454, while Estimated Number of Rows is 0. This indicates that the statistics are old, and need to be updated.

NEW QUESTION 77

You are planning a set of stored procedures that must be able to access memory-optimized tables. You need to optimize the performance of the stored procedures. Which statement should you include in the stored procedure definitions?

- A. WITH RECOMPILE
B. WITH NATIVE_COMPILATION
C. WITH EXECUTE AS SELF
D. WITH NO INFOMSG5

Answer: D

NEW QUESTION 79

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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and SalesPerson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```

You must modify the ProductReview Table to meet the following requirements:

- * The table must reference the ProductID column in the Product table
- * Existing records in the ProductReview table must not be validated with the Product table.
- * Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- * Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- * Create new rows in the table without granting INSERT permissions to the table.
- * Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- * a constraint on the SaleID column that allows the field to be used as a record identifier
- * a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- * a constraint on the CategoryID column that allows one row with a null value in the column
- * a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- * The table must hold 10 million unique sales orders.
- * The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- * Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to update the SalesHistory table

How should you complete the Transact_SQL statement? To answer? select the appropriate Transact-SQL, segments in the answer area.

Answer Area

```
IF OBJECT_id(*SalesHistory*)>0 DROP TABLE SalesHistory
GO
IF OBJECT_ID(*ProductTypes*)>0 DROP TABLE ProductTypes
GO
CREATE TABLE ProductTypes
(
    ProductID SMALLINT,
    ProductDescription VARCHAR(255),
    CONSTRAINT pk_ProductID PRIMARY KEY (ProductID)
)
GO
CREATE TABLE [dbp].[SalesHistoryK]
[SalesID] [int]
```

IDENTITY(1,4)
IDENTITY(1,4) NOT NULL PRIMARY KEY
UNIQUE

```
[ProductID] SMALLINT NULL ,
[SaleDate] [datetime] NULL
```

```
[SalePrice] [money]
```

NOT NULL
NULL CHECK (SalesPrice > 4)
UNIQUE

```
[CategoryID] [smallint]
```

NOT NULL
NULL CHECK (SalesPrice > 4)
UNIQUE

CONSTRAINT fk_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES SalesHistory(CategoryID)
CONSTRAINT fk_SalesHistoryProductID FOREIGN KEY (ProductID) REFERENCES ProductTypes(ProductID)

```
)
GO
```

Answer:

Explanation: Box 1:

SaleID must be the primary key, as a constraint on the SaleID column that allows the field to be used as a record identifier is required.

Box2:

A constraint that limits the SalePrice column to values greater than four. Box 3: UNIQUE

A constraint on the CategoryID column that allows one row with a null value in the column. Box 4:

A foreign key constraint must be put on the productID referencing the ProductTypes table, as a constraint that uses the ProductID column to reference the Product column of the ProductTypes table is required.

Note: Requirements are:

You must add the following constraints to the SalesHistory table:

NEW QUESTION 84

You are developing an application that connects to a database. The application runs the following jobs:

Job	Transact-SQL statement	Description
JobA	Exec uspDeletePrevRecords	The stored procedure deletes all records from a table named tblBalanceTransactions that were created before the current month by using a single DELETE statement. Approximately 10 million records are deleted each time you run this stored procedure
JobB	Exec uspUpdateCurRecords	This stored procedure updates records in the tblBalanceTransaction table that were created in the current month. Only a few hundred records are updated each time you run this stored procedure.

The READ_COMMITTED_SNAPSHOT database option is set to OFF, and auto-content is set to ON. Within the stored procedures, no explicit transactions are defined.

If JobB starts before JobA, it can finish in seconds. If JobA starts first, JobB takes a long time to complete. You need to use Microsoft SQL Server Profiler to determine whether the blocking that you observe in JobB is caused by locks acquired by JobA.

Which trace event class in the Locks event category should you use?

- A. LockAcquired
- B. LockCancel
- C. LockDeadlock
- D. LockEscalation

Answer: A

Explanation: The Lock:Acquired event class indicates that acquisition of a lock on a resource, such as a data page, has been achieved.

The Lock:Acquired and Lock:Released event classes can be used to monitor when objects are being locked, the type of locks taken, and for how long the locks were retained. Locks retained for long periods of time may cause contention issues and should be investigated.

NEW QUESTION 85

Case Study Background

You have a database named HR1 that includes a table named Employee.

You have several read-only, historical reports that contain regularly changing totals. The reports use multiple queries to estimate payroll expenses. The queries run concurrently. Users report that the payroll estimate reports do not always run. You must monitor the database to identify issues that prevent the reports from running.

You plan to deploy the application to a database server that supports other applications. You must minimize the amount of storage that the database requires.

Employee Table

You use the following Transact-SQL statements to create, configure, and populate the Employee table:

```
CREATE TABLE dbo.Employee
(
    EmployeeId INT PRIMARY KEY,
    LastName varchar(50),
    FirstName varchar(50),
    DepartmentId int,
    HireDate datetime,
    TerminationDate datetime,
    SupervisorId int,
    CostCenterNumber int,
    EmployeeStatus int,
    EmployeePayRate int
)
GO

CREATE INDEX IX_1 on dbo.Employee (LastName, FirstName, DepartmentId) INCLUDE (HireDate)
CREATE INDEX IX_2 on dbo.Employee (LastName) INCLUDE (EmployeeId, FirstName, DepartmentId)
CREATE INDEX IX_3 on dbo.Employee (LastName, FirstName) INCLUDE (DepartmentId)
CREATE INDEX IX_4 on dbo.Employee (LastName, FirstName) INCLUDE (HireDate, DepartmentId)
GO

INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1001, 'Employee A', 3001001)
INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1002, 'Employee B', 3001001)
GO
```

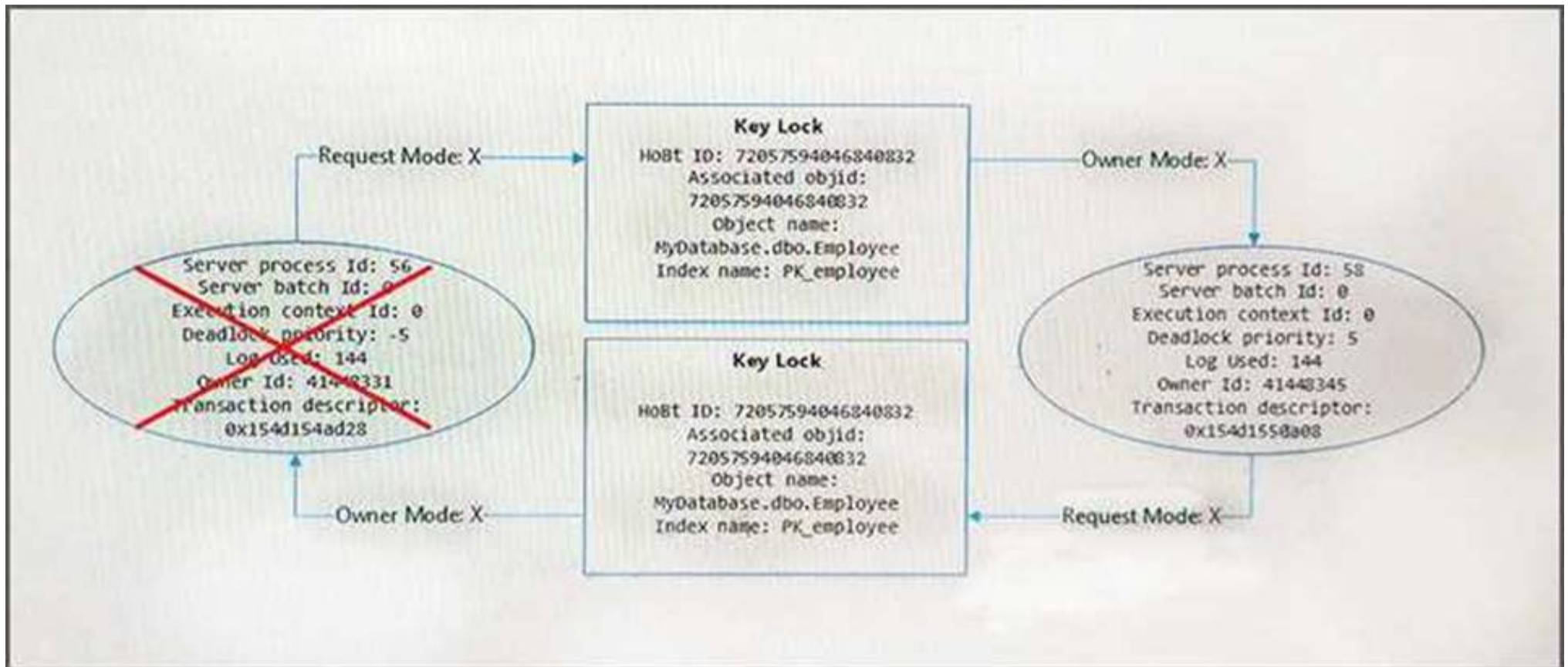
Application

You have an application that updates the Employees table. The application calls the following stored procedures simultaneously and asynchronously:

- UspA: This stored procedure updates only the EmployeeStatus column.
- UspB: This stored procedure updates only the EmployeePayRate column.

The application uses views to control access to data. Views must meet the following requirements:

- Allow user access to all columns in the tables that the view accesses.
- Restrict updates to only the rows that the view returns. Exhibit



Both of the stored procedures experience blocking issues. UspB must not abort if UspA commits changes to a row before UspB commits changes to the same row. UspA must not abort if UspB commits changes to a row before UspA commits changes to the same row.

You need to specify the transaction isolation levels to enable row versioning.

How should you complete the Transact-SQL statements? To answer, drag the Transact-SQL segments to the correct locations. Each Transact-SQL segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Transact-SQL segments

SET ALLOW_SNAPSHOT_ISOLATION OFF;

SET TRANSACTION ISOLATION LEVEL
READ COMMITTED;

SET TRANSACTION ISOLATION LEVEL
READ UNCOMMITTED;

SET TRANSACTION ISOLATION LEVEL
REPEATABLE READ;

SET TRANSACTION ISOLATION LEVEL
SERIALIZABLE;

SET READ_COMMITTED_SNAPSHOT ON;

SET READ_COMMITTED_SNAPSHOT OFF;

Answer area

ALTER DATABASE HR1

Transact-SQL segment

CREATE PROCEDURE UspA
AS
BEGIN

SET NOCOUNT ON;

Transact-SQL segment

BEGIN TRANSACTION;

...

COMMIT TRANSACTION;

END;

CREATE PROCEDURE UspB
AS
BEGIN

Transact-SQL segment

BEGIN TRANSACTION;

...

COMMIT TRANSACTION;

Answer:

Explanation: References: [https://technet.microsoft.com/en-us/library/ms175095\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175095(v=sql.105).aspx) [https://technet.microsoft.com/en-us/library/ms173763\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms173763(v=sql.105).aspx)

NEW QUESTION 87

You create a database table named FactSales by running the following Transact-SQL statements:

```
CREATE TABLE FactSales (
DateKey [int] NOT NULL,
Productkey [int] NOT NULL,
CustomerKey [int] NOT NULL,
EmployeeKey [int] NOT NULL,
SalesAmount money,
SalesCost money
INDEX ix_FactSales CLUSTERED (DateKey));

CREATE INDEX ix_FactSales_ProductKey ON FactSales (ProductKey);
CREATE INDEX ix_FactSales_CustomerKey ON FactSales (CustomerKey);
CREATE INDEX ix_FactSales_EmployeeKey ON FactSales (EmployeeKey);
```

You must optimize the indexes without making changes to the ix_FactSales_EmployeeKey index. You need to implement a columnstore index for the table. How should you complete the Transact-SQL code? To answer, select the appropriate Transact-SQL segments in the answer area.
NOTE: Each correct selection is worth one point.

Answer Area

	▼
DROP INDEX ix_FactSales_ProductKey ON FactSales	
DROP INDEX ix_FactSales_EmployeeKey ON FactSales	
DROP INDEX ix_FactSales ON FactSales	

	▼
CREATE CLUSTERED COLUMNSTORE INDEX ix FactSales ON FactSales	
DROP INDEX ix_FactSales_CustomerKey ON FactSales	
DROP INDEX ix_FactSales_EmployeeKey ON FactSales	
DROP INDEX ix_FactSales ON FactSales	

	▼
CREATE CLUSTERED COLUMNSTORE INDEX ix FactSales ON FactSales	
CREATE CLUSTERED COLUMNSTORE INDEX ix_FactSales ON FactSales WITH (DROP_EXISTING = ON)	
ALTER INDEX ix_FactSales_EmployeeKey ON FactSales REBUILD	
CREATE INDEX ix_FactSales_EmployeeKey ON FactSales (EmployeeKey);	

Answer:

Explanation:

Answer Area

▼

DROP INDEX ix_FactSales_ProductKey ON FactSales
DROP INDEX ix_FactSales_EmployeeKey ON FactSales
DROP INDEX ix_FactSales ON FactSales

▼

CREATE CLUSTERED COLUMNSTORE INDEX ix FactSales ON FactSales
DROP INDEX ix_FactSales_CustomerKey ON FactSales
DROP INDEX ix_FactSales_EmployeeKey ON FactSales
DROP INDEX ix_FactSales ON FactSales

▼

CREATE CLUSTERED COLUMNSTORE INDEX ix FactSales ON FactSales
CREATE CLUSTERED COLUMNSTORE INDEX ix FactSales ON FactSales WITH (DROP_EXISTING = ON)
ALTER INDEX ix_FactSales_EmployeeKey ON FactSales REBUILD
CREATE INDEX ix_FactSales_EmployeeKey ON FactSales (EmployeeKey);

NEW QUESTION 92

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 have a database that contains a table named Employees. The table stores information about the employees of your company.

You need to implement the following auditing rules for the Employees table:

- Record any changes that are made to the data in the Employees table.
- Customize the data recorded by the audit operations.

Solution: You implement a user-defined function on the Employees table. Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation: SQL Server 2016 provides two features that track changes to data in a database: change data capture and change tracking. These features enable applications to determine the DML changes (insert, update, and delete operations) that were made to user tables in a database.

Change data is made available to change data capture consumers through table-valued functions (TVFs). References: <https://msdn.microsoft.com/en-us/library/cc645858.aspx>

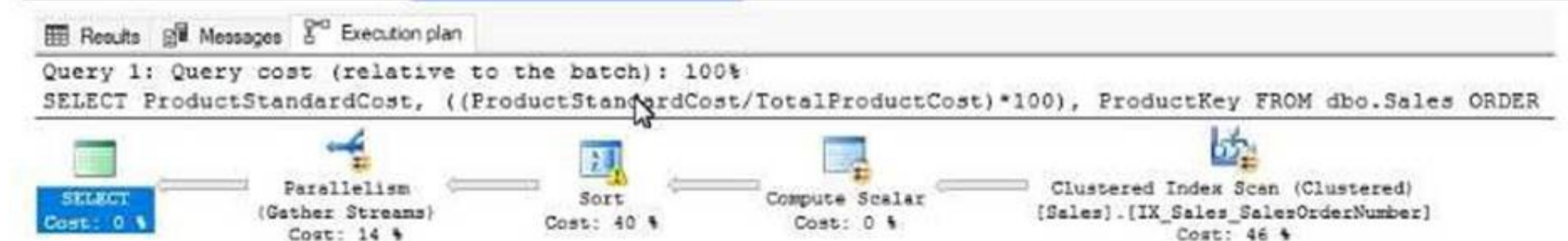
NEW QUESTION 94

You have a Microsoft SQL Server database that has a table named Sales. The table is used for retrieving data and is updated during non business hours.

You run the following Transact-SQL statement:

```
SELECT ProductStandardCost, ((ProductStandardCost/TotalProductCost)*100) StandardCostRatio, ProductKey
FROM Sales
ORDER BY ProductStandardCost DESC, ProductKey
```

You analyze the execution plan for the statement. (Click the Exhibit button).



You need to add an index that optimizes performance. How should you complete the Transact-SQL statement?

CREATE INDEX IX_Sales_01

ProductStandardCost DESC

ProductStandardCost ASC, ProductKey ASC

ProductStandardCost ASC, ProductKey DESC

ProductStandardCost DESC, ProductKey DESC

ON Sales (

WITH (ONLINE = ON)

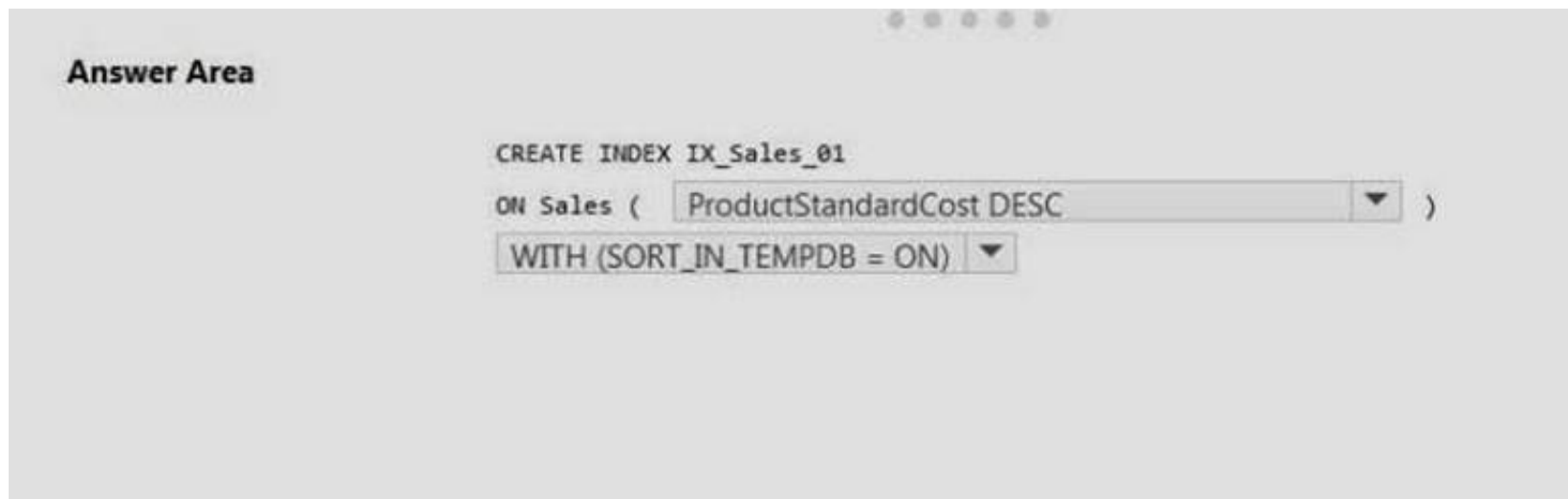
INCLUDE (TotalProductCost)

WITH (SORT_IN_TEMPDB = ON)

WITH (SORT_IN_TEMPDB = OFF)

Answer:

Explanation:



NEW QUESTION 96

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some questions sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a 3-TB database. The database server has 64 CPU cores. You plan to migrate the database to Microsoft Azure SQL Database.

You need to select the service tier for the Azure SQL database. The solution must meet or exceed the current processing capacity.

Solution: You select the Standard service tier. Does this meet the goal?

A. Yes

B. No

Answer: B

Explanation: Premium service is required for 3 TB of storage. Single database DTU and storage limits

	Basic	Standard	Premium
Maximum storage size	2 GB	1 TB	4 TB
Maximum DTUs	5	3000	4000

References: <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-dtu>

NEW QUESTION 99

You suspect deadlocks on a database.

Which two trace flags in the Microsoft SQL Server error log should you locate? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. 1204

B. 1211

C. 1222

D. 2528

E. 3205

Answer: AC

Explanation: Trace flag 1204 returns the resources and types of locks participating in a deadlock and also the current command affected.

Trace flag 1222 returns the resources and types of locks that are participating in a deadlock and also the current command affected, in an XML format that does not comply with any XSD schema.

References:

<https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-traceon-trace-flags-transact-sql?vie>

NEW QUESTION 104

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 is independent of the other questions in this series. Information and details provided in a question apply only to that question.

Table name	Description
TBL1	<ul style="list-style-type: none"> The table has 25 columns. The table will contain 10 million records. Approximately 100,000 records will be inserted monthly.
TBL2	<ul style="list-style-type: none"> The table has 25 columns. The table has 25 columns. The table will contain 100,000 records. The frequency of inserting, updating, and deleting records is low.

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

```
SELECT *
FROM TBL1
WHERE Column1 BETWEEN '01/01/2016' AND '30/04/2016'

SELECT Column5, Column6
FROM TBL2
WHERE Column2 = 'ABC156XYZ'
```

There are no foreign key relationships between TBL1 and TBL2.

You need to minimize the amount of time required for the two queries to return records from the tables. What should you do?

- Create clustered indexes on TBL1 and TBL2.
- Create a clustered index on TBL1 Create a nonclustered index on tbl2 and add the most frequently queried columns as included columns.
- Create a nonclustered index on tbl2 only.
- Create unique constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- 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.
- Drop existing indexes on TBL1 and then create a clustered columnstore index.
- Create a nonclustered columnstore index on TBL1. Make no changes to TBL2.
- Create check constraints on both TBL1 and tbl2. Create a partitioned view that combines columns from TBL1 and tbl2.
- Create an indexed view that combines columns from TBL1 and TBL2.

Answer: F

NEW QUESTION 109

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 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 are developing an application to track customer sales. You create tables to support the application. You need to create a database object that meets the following data entry requirements:

Table name	Data entry requirements
Customers	Inserts must be grouped in a batch of Transact-SQL statements.
CustomerTransactions	Inserts must be grouped in a batch of Transact-SQL statements. Inserts must be grouped in a batch of Transact-SQL statements.
Invoices	All of the data modifications made must become a permanent part of the database if data entry is successful. All data entry modifications must be erased if an error occurs.
Orders	Inserts must be grouped in a batch of Transact-SQL statements. All of the data modifications made must become a permanent part of the database if data entry is successful. All data entry modifications must be erased if an error occurs.

What should you create?

- extended procedure
- CLR procedure
- user-defined procedure
- DML trigger
- DDL trigger
- scalar-valued function
- table-valued function

Answer: C

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

NEW QUESTION 113

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:

```
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)
);
```

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.

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 Transact-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;
    END
```

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

You must update the design of the Customer table to meet the following requirements.

- * You must be able to store up to 50 accounts for each customer.
- * Users must be able to retrieve customer information by supplying an account number.
- * Users must be able to retrieve an account number by supplying customer information.

Which three Transact-SQL segments should you use to develop the solution? To answer, move the appropriate code blocks from the list of code blocks to the answer area and arrange them in the correct order.

Transact-SQL segments

Answer Area

WHERE CustomerStatus <> 1;

ON Customer(CreateDate, CustomerStatus) INCLUDE
(Segmentm ContryCode, PhoneNumber)

WHERE CustomerID = 1;

CREATE NONCLUSTERED INDEX IX_Customer_CreatedDate

ON Customer(CreateDate) INCLUDE (Segment, Country-
Code, PhoneNumber)

CREATE CLUSTERED INDEX IX_Customer_CreatedDate



Answer:

Explanation: Box 1: Clustered Index

With the same size of keys, the nonclustered indexes need more space than clustered indexes. Box 2, Box 3:

Include the CustomerStatus column in the index, and only when CustomerStatus not equal to 1 (the active customers).

References:

<http://www.sqlserverlogexplorer.com/overview-of-cluster-and-noncluster-index/>

NEW QUESTION 118

You have a table named Person.Address that includes the following columns:

AddressID
AddressLine1
AddressLine2
City
StateProvinceID
PostalCode
RowGuid
ModifiedDate

You need to create a nonclustered index on PostalCode named IX_Address_PostalCode that uses the following included columns:

AddressLine1
AddressLine2
City
StateProvinceID

How should you complete the Transact-SQL statement? To answer, select the appropriate Transact_SQL segments in the answer are.

Answer Area

CREATE NONCLUSTERED IX_Address_PostalCode

ON

INCLUDE

INDEX

WHERE

(Person.Address (PostalCode

(AddressLine1, AddressLine2, City, StateProvinceID)

ON

INCLUDE

INDEX

WHERE

Answer:

Explanation: Box 1: INDEX

Box 2: ON

Box 3: INCLUDE

INCLUDE (column [,... n]) specifies the non-key columns to be added to the leaf level of the nonclustered index. The nonclustered index can be unique or non-unique.

References:

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

NEW QUESTION 119

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 use a memory-optimized filegroup. The database contains a table named Table1. The table must support the following workloads:

Workload	Type	Description
Reporting	Existing	The reporting workload must scan most of the records in the table to aggregate on a number of columns. A clustered columnstore index is already created on the table to support this workload.
OLTP	New	The OLTP workload must support 3,000 transactions per second. Rows are identified by using two columns. The filter is variant on one of the two columns while constant on the other. Only a small number of records with a few columns are returned by the query.

You need to add the most efficient index to support the new OLTP workload, while not deteriorating the existing Reporting query performance. What should you do?

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

Answer: C

Explanation: 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. References: [https://technet.microsoft.com/en-us/library/cc280372\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/cc280372(v=sql.105).aspx)

NEW QUESTION 124

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

- A. CREATE INDEX IX_OrderLines_I ON OrderLines (OrderID, UnitPrice) INCLUDE(Description, Quantity)
- B. CREATE INDEX IX_OrderLines_I ON OrderLines (OrderID, UnitPrice) INCLUOE(Quantity)
- C. CREATE INDEX IX_OrderLines_I ON OrderLines (OrderID, UnitPrice, Quantity)
- D. CREATE INDEX IX_OrderLines_I ON OrderLines (UnitPrice, OrderID) INCLUDE(Description, Quantity)

Answer: A

NEW QUESTION 126

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 have a database named DB1 that includes a table named Sales-Orders. You grant a user named User1 select permissions on the sales schema.

You need to ensure that User1 can select data from the sales.orders table without specifying the schema name in any Transact-SQL statements.

Solution: You create a stored procedure in the sales schema that selects the data from the sales.orders table.

Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

NEW QUESTION 128

Case study Background

You have a database named HR1 that includes a table named Employee.

You have several read-only, historical reports that contain regularly changing totals. The reports use multiple queries to estimate payroll expenses. The queries run concurrently. Users report that the payroll estimate reports do not always run. You must monitor the database to identify issues that prevent the reports from running.

You plan to deploy the application to a database server that supports other applications. You must minimize the amount of storage that the database requires.

Employee Table

You use the following Transact-SQL statements to create, configure, and populate the Employee table:

```
CREATE TABLE dbo.Employee
(
    EmployeeId INT PRIMARY KEY,
    LastName varchar(50),
    FirstName varchar(50),
    DepartmentId int,
    HireDate datetime,
    TerminationDate datetime,
    SupervisorId int,
    CostCenterNumber int,
    EmployeeStatus int,
    EmployeePayRate int
)
GO

CREATE INDEX IX_1 on dbo.Employee (LastName, FirstName, DepartmentId) INCLUDE (HireDate)
CREATE INDEX IX_2 on dbo.Employee (LastName) INCLUDE (EmployeeId, FirstName, DepartmentId)
CREATE INDEX IX_3 on dbo.Employee (LastName, FirstName) INCLUDE (DepartmentId)
CREATE INDEX IX_4 on dbo.Employee (LastName, FirstName) INCLUDE (HireDate, DepartmentId)
GO

INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1001, 'Employee A', 3001001)
INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1002, 'Employee B', 3001001)
GO
```

Application

You have an application that updates the Employees table. The application calls the following stored procedures simultaneously and asynchronously:

UspA: This stored procedure updates only the EmployeeStatus column.

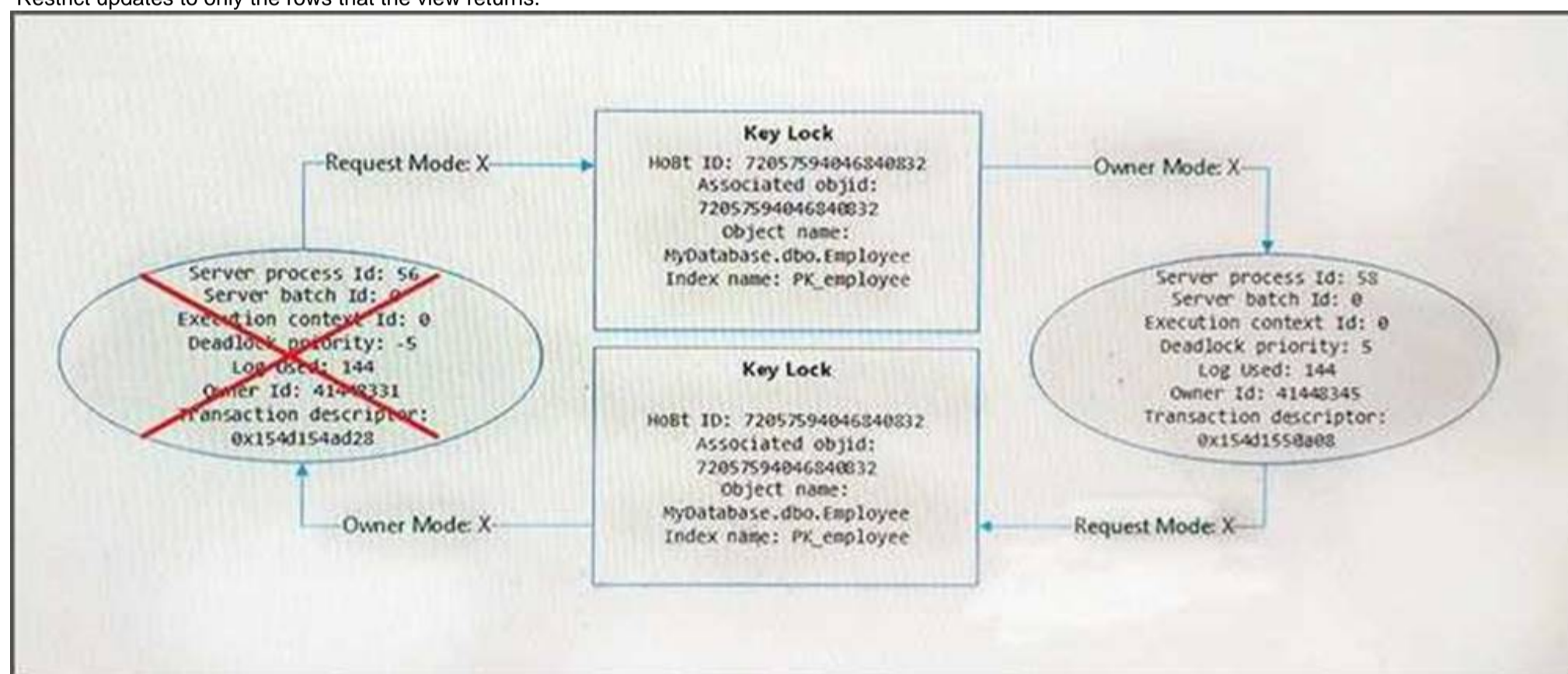
UspB: This stored procedure updates only the EmployeePayRate column.

The application uses views to control access to data.

Views must meet the following requirements:

Allow user access to all columns in the tables that the view accesses.

Restrict updates to only the rows that the view returns.



You observe that the four indexes require a large amount of disk space. You must reduce the amount of disk space that the indexes are using.

You need to create a single index to replace the existing indexes.

Which three Transact-SQL segments should you use to develop the solution? To answer, move the appropriate Transact-SQL segments from the list of Transact-SQL segments to the answer area and arrange them in the correct order.

NOTE: More than one order of answer choices is correct. You will receive credit for any of the correct orders you select.

Transact-SQL segments

(INCLUDE (HireDate, DepartmentId)

(LastName, FirstName)

(FirstName, LastName, DepartmentId)

INCLUDE (LastName, DepartmentId,
EmployeeId)

CREATE INDEX IX_New on dbo.Employee

INCLUDE (HireDate, DepartmentId
EmployeeId)

(FirstName, LastName)

CREATE CLUSTERED INDEX IX_New on
dbo. Employee

Answer Area



Answer:

Explanation: Scenario: You observe that the four indexes require a large amount of disk space. You must reduce the amount of disk space that the indexes are using.

Current indexes:

```
CREATE INDEX IX_1 on dbo.Employee (LastName, FirstName, DepartmentId) INCLUDE (HireDate)
CREATE INDEX IX_2 on dbo.Employee (LastName) INCLUDE (EmployeeId, FirstName, DepartmentId)
CREATE INDEX IX_3 on dbo.Employee (LastName, FirstName) INCLUDE (DepartmentId)
CREATE INDEX IX_4 on dbo.Employee (LastName, FirstName) INCLUDE (HireDate, DepartmentId)
GO
```

NEW QUESTION 129

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 add a unique clustered index to the Employees table. Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation: References: <https://msdn.microsoft.com/en-us/library/ms190457.aspx>

NEW QUESTION 130

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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and SalesPerson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```

You must modify the ProductReview Table to meet the following requirements:

- * The table must reference the ProductID column in the Product table
- * Existing records in the ProductReview table must not be validated with the Product table.
- * Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- * Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- * Create new rows in the table without granting INSERT permissions to the table.
- * Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- * a constraint on the SaleID column that allows the field to be used as a record identifier
- * a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- * a constraint on the CategoryID column that allows one row with a null value in the column
- * a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- * The table must hold 10 million unique sales orders.
- * The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- * Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to modify the design of the Orders table. What should you create?

- A. a stored procedure with the RETURN statement
- B. a FOR UPDATE trigger
- C. an AFTER UPDATE trigger
- D. a user defined function

Answer: D

Explanation: Requirements: You must modify the Orders table to meet the following requirements:

1. Create new rows in the table without granting INSERT permissions to the table.
2. Notify the sales person who places an order whether or not the order was completed. References: <https://msdn.microsoft.com/en-us/library/ms186755.aspx>

NEW QUESTION 132

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.

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 Transact-SQL statement:

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 line 09 due to a CHECK constraint violation on the TotalDepositAccountCount column.

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

- 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.

Answer: D

NEW QUESTION 136

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 update the application to use implicit transactions when connecting to the database. Does the solution meet the goal?

- A. Yes
- B. No

Answer: B

Explanation: References:

<http://sqltouch.blogspot.co.za/2013/05/writelog-waittype-implicit-vs-explicit.html>

NEW QUESTION 140

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 that contains the following tables: BlogCategory, BlogEntry, ProductReview, Product, and SalesPerson. The tables were created using the following Transact SQL statements:

```
CREATE TABLE BlogCategory
(
    CategoryID int NOT NULL PRIMARY KEY,
    CategoryName nvarchar (20)
);

CREATE TABLE BlogEntry
(
    Entry int NOT PRIMARY KEY,
    Entrytitle nvarchar (50),
    Category int NOT NULL FOREIGN KEY REFERENCES BlogCategory
(CategoryID)
);

CREATE TABLE dbo.ProductReview
(
    ProductReviewID IDENTITY(1,1) PRIMARY KEY,
    Product int NOT NULL,
    Review varchar (1000) NOT NULL
);

CREATE TABLE dbo.Product
(
    ProductID int Identity(1,1) PRIMARY KEY,
    Name varchar(1000) NOT NULL
);

CREATE TABLE dbo.SalesPerson
(
    SalesPersonID int IDENTITY(1,1) PRIMARY KEY,
    Name varchar (1000) NOT NULL,
    SalesID Money
)
```

You must modify the ProductReview Table to meet the following requirements:

- * The table must reference the ProductID column in the Product table
- * Existing records in the ProductReview table must not be validated with the Product table.
- * Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
- * Changes to records in the Product table must propagate to the ProductReview table.

You also have the following database tables: Order, ProductTypes, and SalesHistory, The transact-SQL statements for these tables are not available.

You must modify the Orders table to meet the following requirements:

- * Create new rows in the table without granting INSERT permissions to the table.
- * Notify the sales person who places an order whether or not the order was completed.

You must add the following constraints to the SalesHistory table:

- * a constraint on the SaleID column that allows the field to be used as a record identifier
- * a constant that uses the ProductID column to reference the Product column of the ProductTypes table
- * a constraint on the CategoryID column that allows one row with a null value in the column
- * a constraint that limits the SalePrice column to values greater than four

Finance department users must be able to retrieve data from the SalesHistory table for sales persons where the value of the SalesYTD column is above a certain threshold.

You plan to create a memory-optimized table named SalesOrder. The table must meet the following requirements:

- * The table must hold 10 million unique sales orders.
- * The table must use checkpoints to minimize I/O operations and must not use transaction logging.
- * Data loss is acceptable.

Performance for queries against the SalesOrder table that use Where clauses with exact equality operations must be optimized.

You need to enable referential integrity for the ProductReview table.

How should you complete the relevant Transact-SQL statement? To answer? select the appropriate Transact-SQL segments in the answer area. Select two alternatives.

- A. For the first selection select: WITH CHECK
- B. For the first selection select: WITH NOCHECK
- C. For the second selection select: ON DELETE NO ACTION ON UPDATE CASCADE
- D. For the second selection select: ON DELETE CASCADE ON UPDATE CASCADE
- E. For the second selection select: ON DELETE NO ACTION ON UPDATE NO ACTION
- F. For the second selection select: ON DELETE CASCADE ON UPDATE NO ACTION

Answer: BC

Explanation: B: We should use WITH NOCHECK as existing records in the ProductReview table must not be validated with the Product table.

C: Deletes should not be allowed, so we use ON DELETE NO ACTION. Updates should be allowed, so we use ON DELETE NO CASCADE

NO ACTION: the Database Engine raises an error, and the update action on the row in the parent table is rolled back.

CASCADE: corresponding rows are updated in the referencing table when that row is updated in the parent table.

Note: ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Specifies what action happens to rows in the table that is altered, if those rows have a referential relationship and the referenced row is deleted from the parent table. The default is NO ACTION.

ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT }

Specifies what action happens to rows in the table altered when those rows have a referential relationship and the referenced row is updated in the parent table. The default is NO ACTION.

Note: You must modify the ProductReview Table to meet the following requirements:

1. The table must reference the ProductID column in the Product table
2. Existing records in the ProductReview table must not be validated with the Product table.
3. Deleting records in the Product table must not be allowed if records are referenced by the ProductReview table.
4. Changes to records in the Product table must propagate to the ProductReview table. References: <https://msdn.microsoft.com/en-us/library/ms190273.aspx>

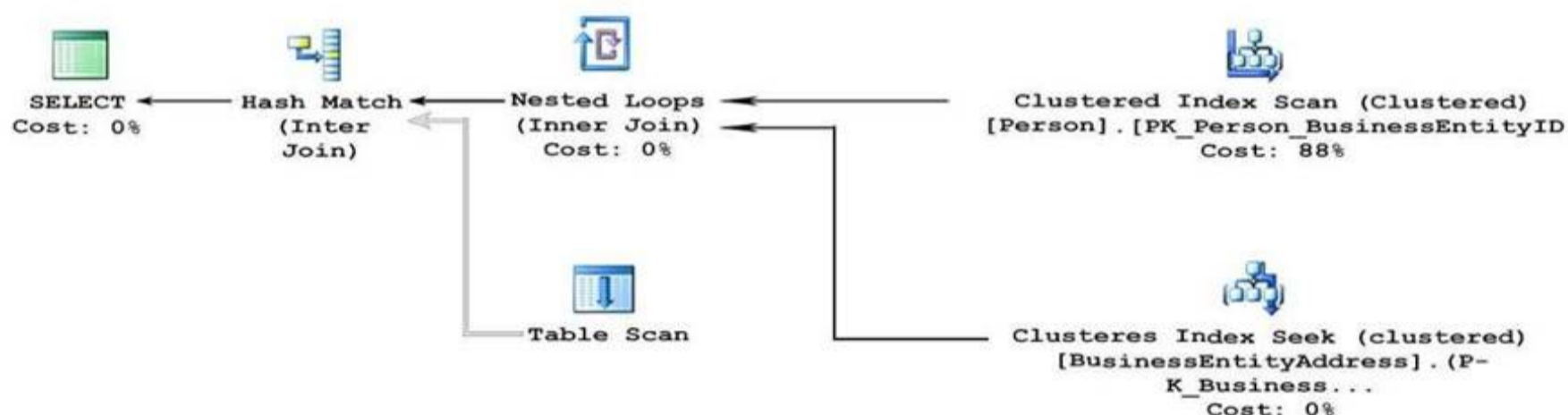
<https://msdn.microsoft.com/en-us/library/ms188066.aspx>

NEW QUESTION 144

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 is independent of the other questions in this series. Information and details provided in a question apply only to that question. You have a database named DB1. There is no memory-optimized filegroup in the database. You run the following query:

```
DECLARE @LastName varchar(20) = "contoso";
SELECT P.FirstName, Addr.*
FROM Person.Person P
JOIN Person.BusinessEntityAddress ON P.BusinessEntityID =
BusinessEntityAddress.BusinessEntityID
JOIN Person.[Address] ADDR ON BusinessEntityAddress.AddressID =
Addr.AddressID
WHERE P.LastName = @LastName;
```

The following image displays the execution plan the query optimizer generates for this query:



Users frequently run the same query with different values for the local variable @lastName. The table named Person is persisted on disk. You need to create an index on the Person.Person table that meets the following requirements:

- All users must be able to benefit from the index.
- FirstName must be added to the index as an included column. What should you do?

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

Answer: B

Explanation: By including nonkey columns, you can create nonclustered indexes that cover more queries. This is because the nonkey columns have the following benefits:

They can be data types not allowed as index key columns.

They are not considered by the Database Engine when calculating the number of index key columns or index key size.

NEW QUESTION 145

You have a data warehouse fact table that has a clustered columnstore index. You have multiple CSV files that contain a total of 3 million rows of data. You need to upload the data to the fact table. The solution must avoid the delta group when you import the data. Which solution will achieve the goal in the least amount of time?

- A. Load the source data to a staging table.
- B. Load the data to the fact table by using the insert_select statement and specify the Tablock option on the staging table.
- C. Load the source data to a staging table that has a clustered index on the primary key.
- D. Copy the data to the fact table by using the insert_select statement.
- E. Load the source data to the fact table by running bcp.exe and specify the _Tablock option.
- F. Load the source data to the fact table by using the bulk insert statement and specify the Tablock option.

Answer: D

NEW QUESTION 149

You are analyzing the performance of a database environment. You need to find all unused indexes in the current database.

How should you complete the Transact-SQL statement? To answer, select the appropriate Transact-SQL segments in the answer area.

Answer Area

```
SELECT a.name as TableName
       b.name as IndexName

FROM sys.indexes b

INNER JOIN sys.dm_exec_query_stats s ON s.object_id = b.object_id AND s.index_id = b.index_id

INNER JOIN sys.dm_db_index_usage_stats u ON u.object_id = b.object_id

WHERE ((user_seeks = 0 AND user_scans = 0 AND user_lookups = 0) OR s.object_id is null)
```

Answer:

Explanation: Example: Following query helps you to find all unused indexes within database using sys.dm_db_index_usage_stats DMV.

-- Ensure a USE statement has been executed first. SELECT u.*

FROM [sys].[indexes] i

INNER JOIN [sys].[objects] o ON (i.OBJECT_ID = o.OBJECT_ID)

LEFT JOIN [sys].[dm_db_index_usage_stats] u ON (i.OBJECT_ID = u.OBJECT_ID) AND i.[index_id] = u.[index_id]

AND u.[database_id] = DB_ID() --returning the database ID of the current database WHERE o.[type] <>'S' --shouldn't be a system base table

AND i.[type_desc] <>'HEAP' AND i.[name] NOT LIKE 'PK_'

AND u.[user_seeks] + u.[user_scans] + u.[user_lookups] = 0 AND u.[last_system_scan] IS NOT NULL

ORDER BY 1 ASC

References: https://basitaalishan.com/2012/06/15/find-unused-indexes-using-sys-dm_db_index_usage_stats/

NEW QUESTION 154

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 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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. What should you do?

- Create clustered indexes on TBL1 and TBL2.
- Create a clustered index on TBL1. Create a nonclustered index on TBL2 and add the most frequently queried columns as included columns.
- Create a nonclustered index on TBL2 only.
- Create unique constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- 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.
- Drop existing indexes on TBL1 and then create a clustered columnstore index.
- Create a nonclustered columnstore index on TBL1.
- Make no changes to TBL2.
- Create CHECK constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- Create an indexed view that combines columns from TBL1 and TBL2.

Answer: H

NEW QUESTION 155

You are profiling a frequently used database table named UserEvents. The READ_COMMITTED_SNAPSHOT database option is set to OFF. In the trace results, you observe that lock escalation occurred for one stored procedure even though the number of locks in the database did not exceed memory or configuration thresholds. Events details are provided in the following table:

Attribute	Value
EventClass	LockEscalation
Mode	5 - X (X represents exclusive)
ObjectID	274100017 (objectID of table UserEvents)
Type	5 - OBJECT
TextData	<pre>CREATE PROCEDURE uspDeleteEvents (@EventType tinyint) AS BEGIN SET NOCOUNT ON; SET TRAN ISOLATION LEVEL READ COMMITTED; LABEL_DELETE: DELETE FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType; END;</pre>

You need to modify the uspDeleteEvents stored procedure to avoid lock escalation. How should you modify the stored procedure? To answer, select the appropriate Transact-SQL segments in the answer area.

Answer Area

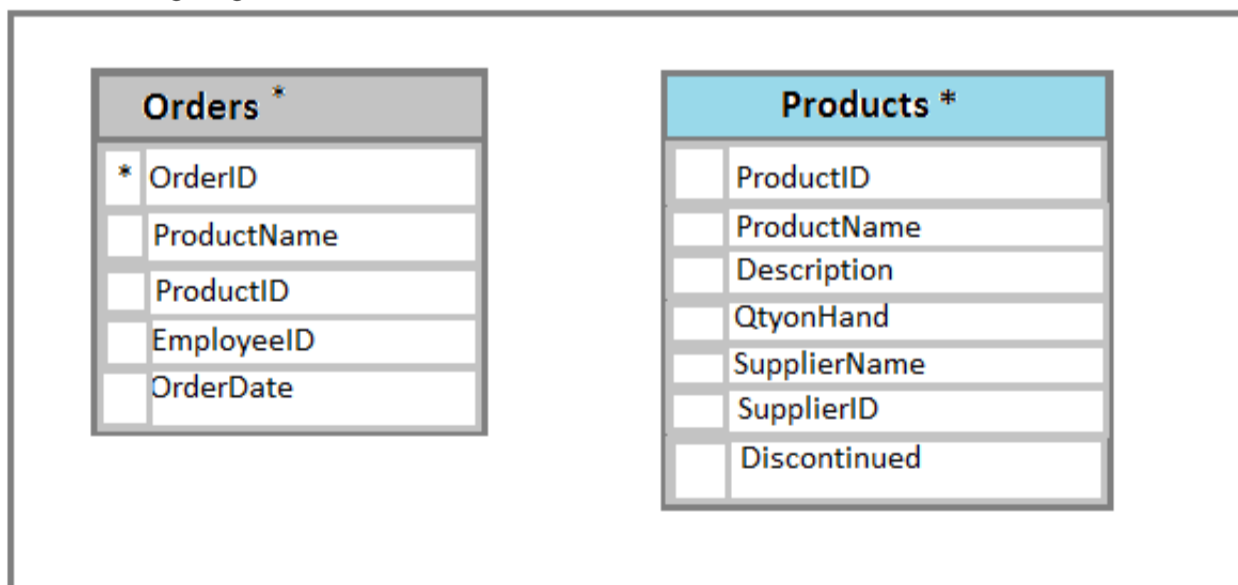
```
CREATE PROCEDURE uspDeleteEvents (@EventType tinyint)
AS
BEGIN
    SET NOCOUNT ON;
    SET TRAN ISOLATION LEVEL READ COMMITTED;
    LABEL_DELETE_DELETE_TOP(4000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE FROM UserEvents WITH (PAGLOCK) WHERE EventType = @EventType
    LABEL_DELETE_DELETE TOP(400000) FROM UserEvents WITH (ROWLOCK) WHERE EventType = @EventType
    IF @@TRANCOUNT > 0 COMMIT TRAN
    WHILE @@ROWCOUNT > 0 GOTO LABEL_DELETE
    IF @@TRANCCOUNT > 4000 COMMIT TRAN
    SET ROWCOUNT 4000
END;
```

Answer:

Explanation: Delete up to 4000 rows at a time. Keep doing it until all rows have been deleted. Note that @@ROWCOUNT returns the number of rows affected by the last statement. References: <https://msdn.microsoft.com/en-us/library/ms187316.aspx>

NEW QUESTION 160

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 Sales that contains the following database tables: Customer, Order, and Products. The Products table and the Order table are shown in the following diagram.



The customer table includes a column that stores the data for the last order that the customer placed. You plan to create a table named Leads. The Leads table is expected to contain approximately 20,000 records. Storage requirements for the Leads table must be minimized.

You need to create triggers that meet the following requirements:

- * Optimize the performance and data integrity of the tables.
- * Provide a custom error if a user attempts to create an order for a customer that does not exist.
- * In the Customers table, update the value for the last order placed.
- * Complete all actions as part of the original transaction.

In the table below, identify the trigger types that meet the requirements.

NOTE: Make only selection in each column. Each correct selection is worth one point.

Answer Area

Trigger type	Provide custom	Update Customer table
AFTER INSERT trigger	<input type="radio"/>	<input type="radio"/>
INSTEAD OF INSERT trigger	<input type="radio"/>	<input type="radio"/>
AFTER UPDATE trigger	<input type="radio"/>	<input type="radio"/>
INSTEAD OF UPDATE trigger	<input type="radio"/>	<input type="radio"/>

Answer:

Explanation: INSTEAD OF INSERT triggers can be defined on a view or table to replace the standard action of the INSERT statement. AFTER specifies that the DML trigger is fired only when all operations specified in the triggering SQL statement have executed successfully.
References: [https://technet.microsoft.com/en-us/library/ms175089\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms175089(v=sql.105).aspx)

NEW QUESTION 164

You are experiencing performance issues with the database server.

You need to evaluate schema locking issues, plan cache memory pressure points, and backup I/O problems. What should you create?

- A. a System Monitor report
- B. a sys.dm_tran_database_transaction dynamic management view query
- C. an Extended Events session that uses Query Editor
- D. an Activity Monitor session in Microsoft SQL Management Studio.

Answer: D

Explanation: References: <https://msdn.microsoft.com/en-us/library/hh212951.aspx>

NEW QUESTION 168

You are performing a code review for Transact-SQL statements.

What action does the constraint in each statement perform? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

• • • • •

Answer Area

SQL Statement

Action

```
CREATE TABLE Persons (
    ID INT NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT,
    CONSTRAINT PK_Person PRIMARY KEY (ID, LastName)
)
```

Enforce unique values for columns.
Enforce a reference to a primary key.
Ensure unique values within a column.
Ensure values are acceptable for columns based on an evaluation.

```
CREATE TABLE Orders (
    OrderID INT NOT NULL,
    OrderNumber INT NOT NULL,
    PersonID INT,
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
    REFERENCES Persons(PersonID)
)
```

Enforce unique values for columns.
Enforce a reference to a primary key.
Ensure unique values within columns.
Ensure values are acceptable for columns based on an evaluation.

```
CREATE TABLE Persons (
    ID INT NOT NULL UNIQUE,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT
)
```

Enforce unique values for a columns.
Enforce a reference to a primary key.
Ensure unique values within columns.
Ensure values are acceptable for columns based on an evaluation.

```
CREATE TABLE Persons (
    ID INT NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT CHECK (Age >= 18)
)
```

Enforce unique values for a columns.
Enforce a reference to a primary key.
Ensure unique values within columns.
Ensure values are acceptable for columns based on an evaluation.

Answer:

Explanation:

SQL Statement

Action

```
CREATE TABLE Persons (
    ID INT NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT,
    CONSTRAINT PK_Person PRIMARY KEY (ID, LastName)
)
```

Ensure unique values within a column.

```
CREATE TABLE Orders (
    OrderID INT NOT NULL,
    OrderNumber INT NOT NULL,
    PersonID INT,
    PRIMARY KEY (OrderID),
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)
    REFERENCES Persons(PersonID)
)
```

Enforce a reference to a primary key.

```
CREATE TABLE Persons (
    ID INT NOT NULL UNIQUE,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT
)
```

Enforce unique values for a columns.

```
CREATE TABLE Persons (
    ID INT NOT NULL,
    LastName VARCHAR(255) NOT NULL,
    FirstName VARCHAR(255),
    Age INT CHECK (Age >= 18)
)
```

Ensure values are acceptable for columns based on an evaluation.

NEW QUESTION 170

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 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	<ul style="list-style-type: none"> The table will contain 10 million records. The frequency of inserting, updating, and deleting records is low.
TBL2	<ul style="list-style-type: none"> The table will contain 1 million records.

Users frequently run the following query:

```
SELECT TBL1.Column2, TBL2.Column2, SUM(TBL1.Column3), SUM(TBL1.Column4 * TBL1.Column5 * (TBL2.Column4 - TBL2.Column3))
FROM TBL1
INNER JOIN TBL2 TBL1.Column1 = TBL2.Column1
GROUP BY TBL1.Column2, TBL2.Column2
```

Users report that the query takes a long time to return results.

You need to minimize the amount of time required for the query to return data. What should you do?

- Create clustered indexes on TBL1 and TBL2.
- Create a clustered index on TBL1. Create a nonclustered index on TBL2 and add the most frequently queried columns as included columns.
- Create a nonclustered index on TBL2 only.
- Create unique constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- 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.
- Drop existing indexes on TBL1 and then create a clustered columnstore index.
- Create a nonclustered columnstore index on TBL1.
- Make no changes to TBL2.
- Create CHECK constraints on both TBL1 and TBL2. Create a partitioned view that combines columns from TBL1 and TBL2.
- Create an indexed view that combines columns from TBL1 and TBL2.

Answer: D

NEW QUESTION 171

You maintain a Microsoft Azure SQL Database instance.

You grant User1 the SELECT and EXECUTE permissions for all objects in the dbo schema. You must create a stored procedure that allows User1 to view the following information:

details for each connection to the database

a list of all active user connections and internal tasks

You need to create the stored procedure for User1 and ensure that User1 can run the stored procedure without any error.

How should you complete the Transact-SQL statements? To answer, drag the appropriate Transact-SQL segments to the correct locations. Each Transact-SQL segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: More than one combination of answer choices is correct. You will receive credit for any of the correct combinations you select.

Transact-SQL segments

GRANT SELECT ON SCHEMA: :sys TO User1

GRANT VIEW DATABASE STATE TO User1

GRANT VIEW DEFINITION TO User1

SELECT * FROM sys.dm_exec_connections

SELECT * FROM sys.dm_exec_requests

SELECT * FROM sys.dm_exec_sessions

Answer Area

CREATE PROCEDURE ViewConnections

AS

Transact-SQL segment

Transact-SQL segment

GO

Transact-SQL segment

Answer:

Explanation: Box 1: Sys.dm_exec_connections

Sys.dm_exec_connections returns information about the connections established to this instance of SQL Server and the details of each connection. Returns server wide connection information for SQL Server. Returns current database connection information for SQL Database.

Box 2: sys.dm_exec_sessions

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.

Box 3: GRANT VIEW DATABASE STATE To User1

SQL Database: Requires VIEW DATABASE STATE to see all connections to the current database. VIEW DATABASE STATE cannot be granted in the master database.

NEW QUESTION 175

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

Answer: B

Explanation: References:

<http://stackoverflow.com/questions/11444923/stored-procedure-to-update-multiple-tables>

NEW QUESTION 176

You have a database that is experiencing deadlock issues when users run queries. You need to ensure that all deadlocks are recorded in XML format. What should you do?

- A. Create a Microsoft SQL Server Integration Services package that uses sys.dm_tran_locks.
- B. Enable trace flag 1224 by using the Database Consistency Checker(BDCC).
- C. Enable trace flag 1222 in the startup options for Microsoft SQL Server.
- D. Use the Microsoft SQL Server Profiler Lock:Deadlock event class.

Answer: C

Explanation: When deadlocks occur, trace flag 1204 and trace flag 1222 return information that is captured in the SQL Server error log. Trace flag 1204 reports deadlock information formatted by each node involved in the deadlock. Trace flag 1222 formats deadlock information, first by processes and then by resources. The output format for Trace Flag 1222 only returns information in an XML-like format. References: [https://technet.microsoft.com/en-us/library/ms178104\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms178104(v=sql.105).aspx)

NEW QUESTION 180

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 have a table that has a clustered index and a nonclustered index. The indexes use different columns from the table. You have a query named Query1 that uses the nonclustered index.

Users report that Query1 takes a long time to report results. You run Query1 and review the following statistics for an index seek operation:

Index Seek (NonClustered)

Scan a particular range of rows from a nonclustered index.

Physical Operation	Index Seek
Logical Operation	Index Seek
Actual Execution Mode	Row
Actual Number of Rows	3571454
Actual Number of Batches	0
Estimated I/O Cost	0.0093577
Estimated Operator Cost	0.0107304 (0%)
Estimated CPU Cost	0.0013727
Estimated Subtree Cost	0.0107304
Estimated Number of Executions	1
Number of Executions	8
Estimated Number of Rows	0
Estimated Row Size	19 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	True
Node ID	100

You need to resolve the performance issue.

Solution: You update statistics for the nonclustered index. Does the solution meet the goal?

- A. Yes
- B. No

Answer: A

Explanation: We see Actual Number of Row is 3571454, while Estimated Number of Rows is 0. This indicates that the statistics are old, and need to be updated.

NEW QUESTION 182

You need to build a function that meets the following requirements:

*Returns multiple rows

*Optimizes the performance of the query within the function

How should you complete the Transact-SQL statement? To answer, drag the appropriate Transact-SQL segments to the correct locations. Each Transact-SQL segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Transact-SQL segments

RETURNS TABLE AS

```
BEGIN
INSERT INTO @Prices
SELECT ListPrice FROM Product
WHERE ProductId = @ProductId
RETURN
END
```

RETURNS @Prices TABLE (ListPrice money) AS

RETURNS [Money] AS

```
BEGIN
SELECT @ListPrice = ListPrice FROM Product
WHERE ProductId = @ProductId
RETURN @ListPrice
END
```

Answer area

CREATE FUNCTION dbo.GetPrice (@Product

Transact-SQL segment

Transact-SQL segment

Answer:

Explanation:

NEW QUESTION 185

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

Answer: B

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

NEW QUESTION 187

You run the following Transact-SQL following statement:

```
CREATE TABLE Customer(
    CustomerId INT IDENTITY (1, 1) PRIMARY KEY,
    Code CHAR(5) NOT NULL,
    FirstName VARCHAR (50) NOT NULL,
    LastName VARCHAR (50) NOT NULL
)
```

Customer records may be inserted individually or in bulk from an application. You observe that the application attempts to insert duplicate records. You must ensure that duplicate records are not inserted and bulk insert operations continue without notifications. Which Transact-SQL statement should you run?

- A. CREATE UNIQUE NONCLUSTERED INDEX IX_Customer_Code ON Customer (Code) WITH(ONLINE = OFF)
- B. CREATE UNIQUE INDEX IX_CUSTOMER_Code O Customer (Code) WITH (IGNORE_DUP_KEY= ON)
- C. CREATE UNIQUE INDEX IX Customer Code ON Customer (Code) WITH (IGNORE DUP KEY=OFF)
- D. CREATE UNIQUE NONCLUSTERED INDEX IX_Customer_Code ON Customer (Code)
- E. CREATE UNIQUE NONCLUSTERED INDEX IX_Customer_Code ON Customer (Code) WITH (ONLINE = ON)

Answer: B

Explanation: IGNORE_DUP_KEY = { ON | OFF } specifies the error response when an insert operation attempts to insert duplicate key values into a unique index. The IGNORE_DUP_KEY option applies only to insert operations after the index is created or rebuilt. The option has no effect when executing CREATE INDEX, ALTER INDEX, or UPDATE. The default is OFF.

NEW QUESTION 188

You are maintaining statistics for a database table named tblTransaction. The table contains more than 10 million records. You need to create a stored procedure that meets the following requirements:

- On weekdays, update statistics for a sample of the total number of records in the table.
 - On weekends, update statistics by sampling all rows in the table. A maintenance task will call this stored procedure daily.
- How should you complete the stored procedure? To answer, select the appropriate Transact-SQL segments in the answer area.
NOTE: Each correct selection is worth one point.

```
CREATE PROCEDURE uspUpdateTxnStats
AS
BEGIN
    SET NOCOUNT ON;
    SET DATEFIRST 1;
    DECLARE @isWeekDay bit;
    SELECT @isWeekday = CASE WHEN DATEPART (dw, GETDATE()) <=5 THEN 1 ELSE 0 END;
    IF @isWEEKday = 1
    BEGIN
        UPDATE STATISTICS
        SET STATISTICS
        UPDATE TOP(20) STATISTICS
        UPDATE #STATISTICS

    END
    ELSE
    BEGIN
        UPDATE STATISTICS
        SET STATISTICS
        UPDATE TOP(20) STATISTICS
        UPDATE #STATISTICS

    END;
END;
```

UPDATE STATISTICS
 SET STATISTICS
 UPDATE TOP(20) STATISTICS
 UPDATE #STATISTICS

tblTransaction

WITH FULLSCAN
 WITH RESAMPLE
 WITH SAMPLE 20 PERCENT
 WITH SAMPLE 200000 ROWS
 SELECT TOP(20) PERCENT

UPDATE STATISTICS
 SET STATISTICS
 UPDATE TOP(20) STATISTICS
 UPDATE #STATISTICS

tblTransaction

WITH FULLSCAN
 WITH RESAMPLE
 WITH SAMPLE 20 PERCENT
 WITH SAMPLE 200000 ROWS
 SELECT TOP(20) PERCENT

Answer:

Explanation: Box 1: UPDATE STATISTICS Box 2: SAMPLE 20 PERCENT
UPDATE STATISTICS tablenameSAMPLE number { PERCENT | ROWS }

Specifies the approximate percentage or number of rows in the table or indexed view for the query optimizer to use when it updates statistics. For PERCENT, number can be from 0 through 100 and for ROWS, number can be from 0 to the total number of rows.

Box 3: UPDATE STATISTICS Box 4: WITH FULLSCAN

FULLSCAN computes statistics by scanning all rows in the table or indexed view. FULLSCAN and SAMPLE 100 PERCENT have the same results. FULLSCAN cannot be used with the SAMPLE option.

References: <https://msdn.microsoft.com/en-us/library/ms187348.aspx>

NEW QUESTION 192

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your 70-762 Exam with Our Prep Materials Via below:

<https://www.certleader.com/70-762-dumps.html>