70-762^Q&As

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

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

https://www.passapply.com/70-762.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft
Official Exam Center

⚙ **Instant Download** After Purchase

⚙ **100% Money Back** Guarantee

⚙ **365 Days** Free Update

⚙ **800,000+** Satisfied Customers

🏅 SATISFACTION GUARANTEED 100%

QUESTION 1

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.

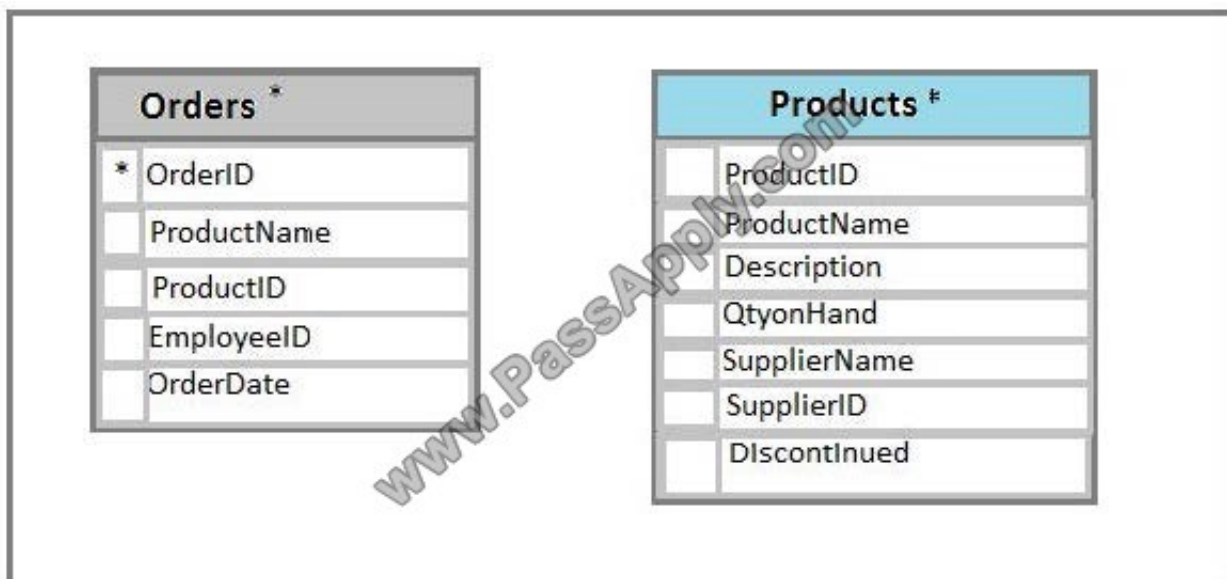Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

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

---

QUESTION 2

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 modify the database design to meet the following requirements:

In the table below, identify the constraint that must be configured for each table.

NOTE: Make only one selection in each column.

Hot Area:

Answer Area

Constraint	Orders table	Products table
Check constraint on **OrderID**	●	●
Foreign key constraint on **ProductID**	●	●
Check constraint on **ProductID**	●	●
Foreign key constraint on **OrderID**	●	●

Correct Answer:

Answer Area

Constraint	Orders table	Products table
Check constraint on **OrderID**	●	●
Foreign key constraint on **ProductID**	● (selected)	●
Check constraint on **ProductID**	●	● (selected)
Foreign key constraint on **OrderID**	●	●

A FOREIGN KEY in one table points to a PRIMARY KEY in another table. Here the foreign key constraint is put on the ProductIDin the Orders, and points to the ProductID of the Products table.

With a check constraint on the ProductID we can ensure that the Products table contains only unique rows.

References:http://www.w3schools.com/sql/sql_foreignkey.asp

QUESTION 3

DRAG DROP

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.

Table diagram

PurchaseOrderDetail (Purchasing)
🔑 PurchaseOrderID
🔑 PurchaseOrderDetailID
DueDate
OrderQty
ProductID
UnitPrice
LineTotal
ReceivedQty
RejectedQty
StockedQty
ModifiedDate

PurchaseOrderHeader (Purchasing)
🔑 PurchaseOrderID
RevisionNumber
Status
EmployeeID
VendorID
ShipMethodID
OrderDate
ShipDate
SubTotal
TaxAmt
Freight
TotalDue
ModifiedDate

Select and Place:

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

Correct Answer:

Transact-SQL segments

```
UPDATE PurchaseOrderDetail
```

```
UPDATE PurchaseOrderDetail
SET SubTotal = SubTotal + LineTotal
FROM inserted
```

```
UPDATE PurchaseOrderHeader
SET SubTotal = LineTotal
FROM inserted
```

Answer area

```
CREATE TRIGGER NewPODetail
```

```
ON PurchaseOrderDetail
```

```
AFTER INSERT AS
```

```
UPDATE PurchaseOrderHeader
SET SubTotal = SubTotal + LineTotal
FROM inserted
```

```
WHERE PurchaseOrderHeader.PurchaseOrderID =
inserted.PurchaseOrderID
```

Explanation:

Box 1: Create TRIGGER NewPODetail

Box 2: ON PurchaseDetail

Box 3: AFTER INSERT AS

Basic syntax:

CREATE TRIGGER trigger_name

ON { table | view }

{ FOR | AFTER | INSTEAD OF }

{ [INSERT] [,] [UPDATE] [,] [DELETE] }

AS

{ sql_statement [;] [,...n] | EXTERNAL NAME }

Box 4: Update PurchaseOrderHeader

SET SubTotal = SubTotal + LineTotal

FROM inserted

Box 5: WHERE PurchaserOrderHeader.PurchaseOrderID=

Inserted.PurchaseOrderID

The trigger must read 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.

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

QUESTION 4

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

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

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

What should you do?

A. Use the NOLOCK option.

B. Execute the DBCC UPDATEUSAGE statement.

C. Use the max worker threads option.

D. Use a table-valued parameter.

E. Set SET ALLOW_SNAPSHOT_ISOLATION to ON.

F. Set SET XACT_ABORT to ON.

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

H. Use the OUTPUT parameters.

Correct Answer: A

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

Incorrect Answers:

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

G: DISABLE, not AUTO, would be better.

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

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

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

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

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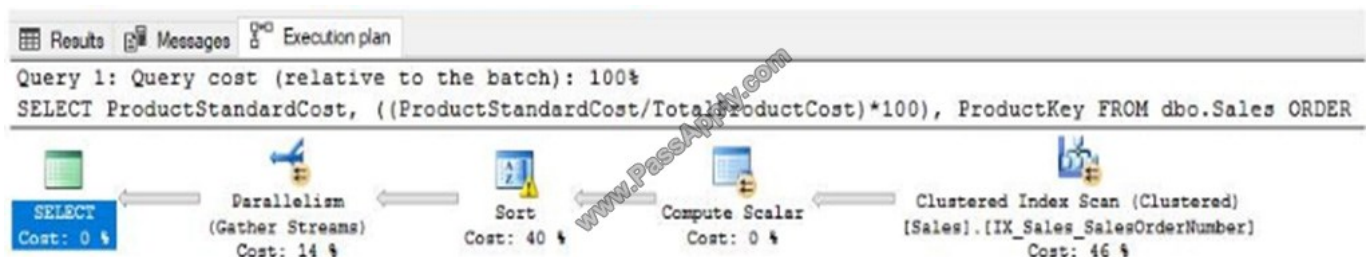

QUESTION 5

HOTSPOT

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). Execution Plan

You need to add an index that optimizes performance.

Hot Area:

Answer Area

```
CREATE INDEX IX_Sales_01
ON Sales (                                    ▼  )
```

| ProductStandardCost DESC |
| ProductStandardCost ASC, ProductKey ASC |
| ProductStandardCost ASC, ProductKey DESC |
| ProductStandardCost DESC, ProductKey DESC |

| ▼ |

| WITH (ONLINE = ON) |
| INCLUDE (TotalProductCost) |
| WITH (SORT_IN_TEMPDB = ON) |
| WITH (SOER IN TEMPDB = OFF) |

Correct Answer:

Answer Area

```
CREATE INDEX IX_Sales_01
ON Sales (                                      ▼ )
```

ProductStandardCost DESC
ProductStandardCost ASC, ProductKey ASC
ProductStandardCost ASC, ProductKey DESC
ProductStandardCost DESC, ProductKey DESC

▼
WITH (ONLINE = ON)
INCLUDE (TotalProductCost)
WITH (SORT_IN_TEMPDB = ON)
WITH (SOER IN TEMPDB = OFF)

Explanation:

Box 1:

Specifying the order in which key values are stored in an index is useful when queries referencing the table have ORDER BY clauses that specify different directions for the key column or columns in that index. In these cases, the index can

remove the need for a SORT operator in the query plan; therefore, this makes the query more efficient.

The Database Engine can move equally efficiently in either direction. An index defined as (RejectedQty DESC, ProductID ASC) can still be used for a query in which the sort direction of the columns in the ORDER BY clause are reversed. For

example, a query with the ORDER BY clause ORDER BY RejectedQty ASC, ProductID DESC can use the index.

Box 2:

You can include nonkey columns in a nonclustered index to avoid exceeding the current index size limitations of a maximum of 16 key columns and a maximum index key size of 900 bytes. The Database Engine does not consider nonkey

columns when calculating the number of index key columns or index key size.

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

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle
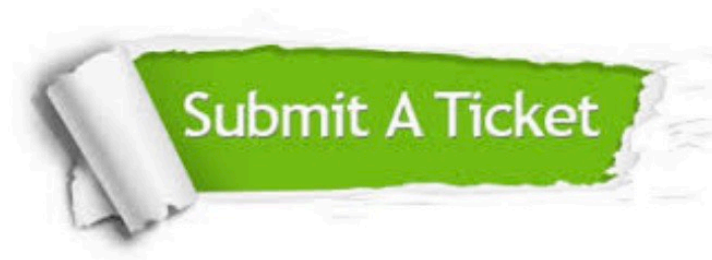
We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.passapply.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket:

Submit A Ticket

One Year Free Update
Free update is available within One Year after your purchase. After One Year, you will get 50% discounts for updating. And we are proud to boast a 24/7 efficient Customer Support system via Email.

Money Back Guarantee
To ensure that you are spending on quality products, we provide 100% money back guarantee for 30 days from the date of purchase.

Security & Privacy
We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.