



70-464^{Q&As}

Developing Microsoft SQL Server Databases

Pass Microsoft 70-464 Exam with 100% Guarantee

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

<https://www.lead4pass.com/70-464.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





QUESTION 1

DRAG DROP

You use the following statement to create a table.

```
CREATE TABLE Employee  
(EmployeeID INT PRIMARY KEY IDENTITY(1,1),  
LastName varchar(50),  
FirstName varchar(50),  
DepartmentId INT,  
SupervisorId INT,  
OfficeId INT,  
Address1 varchar(50),  
Address2 varchar(50),  
City varchar(50),  
State char(2),  
PostalCode varchar(10),  
Country char(2))
```

You have the following queries.

```
SELECT FirstName, LastName, EmployeeId, OfficeID, DepartmentID  
FROM Employee  
WHERE FirstName = 'Ben' AND LastName = 'Smith'  
  
SELECT FirstName, LastName, EmployeeId, OfficeID, DepartmentID  
FROM Employee  
WHERE LastName = 'Smith'
```

You need to create an index to minimize the execution time of the queries.

How should you complete the statement? To answer, drag the appropriate code elements to the correct locations. Each code element 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.

Select and Place:

Code Elements

- DepartmentId
- EmployeeId
- FirstName
- LastName
- OfficeId

Answer Area

```
CREATE INDEX IX_Index3 ON dbo.Employee  
( Code element , Code element )  
INCLUDE ( Code element , Code element )
```



Correct Answer:

Code Elements

EmployeeId

Answer Area

```
CREATE INDEX IX_Index3 ON dbo.Employee  
( LastName , FirstName )  
INCLUDE ( DepartmentId , OfficeId )
```

Box 1: LastName Redesign nonclustered indexes with a large index key size so that only columns used for searching and lookups are key columns. Make all other columns that cover the query into nonkey columns. In this way, you will have all columns needed to cover the query, but the index key itself is small and efficient.

Box 2: FirstName

Box 3: DepartmentID Non-key columns, called included columns, can be added to the leaf level of a nonclustered index to improve query performance by covering the query. That is, all columns referenced in the query are included in the index as either key or non-key columns. This allows the query optimizer to locate all the required information from an index scan; the table or clustered index data is not accessed.

Box 4: OfficeID

QUESTION 2

You need to redesign the system to meet the scalability requirements of the application.

Develop the solution by selecting and arranging the required code blocks in the correct order. You may not need all of the code blocks.

Select and Place:



Code Blocks

```
,  
  UserId int NOT NULL  
  INDEX ix_UserId NONCLUSTERED  
  HASH WITH (BUCKET_COUNT=2),
```

```
,  
  UserId int NOT NULL  
  INDEX x_UserId NONCLUSTERED  
  HASH WITH (BUCKET_COUNT=900000),
```

```
  POSLocation int NOT NULL,  
  StatusID int NOT NULL,  
  CreateDate datetime2 NOT NULL,  
  Price money  
}
```

```
  POSTransactionId int NOT NULL  
  PRIMARY KEY CLUSTERED
```

```
  POSTransactionId int NOT NULL
```

```
ALTER DATABASE CoffeeTransactions  
ADD FILEGROUP [CoffeeTransactions_inmem  
] CONTAINS MEMORY_OPTIMIZED_DATA
```

```
ON [CoffeeTransactions_inmem]
```

```
WITH (MEMORY_OPTIMIZED=ON,  
DURABILITY=SCHEMA_ONLY)
```

```
  POSTransactionId int NOT NULL  
  PRIMARY KEY CLUSTERED  
  HASH WITH (BUCKET_COUNT=1000000)
```

```
,  
  UserId int NOT NULL  
  NONCLUSTERED INDEX ix_UserId,
```

```
CREATE TABLE dbo.POSTransaction (  
  POSTransactionId int NOT NULL  
  PRIMARY KEY NONCLUSTERED  
  HASH WITH (BUCKET_COUNT=1)
```

Answer Area

Correct Answer:



Code Blocks

```
,  
UserId int NOT NULL  
INDEX ix_UserId NONCLUSTERED  
HASH WITH (BUCKET_COUNT=2),
```

```
POSTransactionId int NOT NULL  
PRIMARY KEY CLUSTERED
```

```
POSTransactionId int NOT NULL
```

```
,  
UserId int NOT NULL  
NONCLUSTERED INDEX ix_UserId,
```

```
POSTransactionId int NOT NULL  
PRIMARY KEY NONCLUSTERED  
HASH WITH (BUCKET_COUNT=1)
```

Answer Area

```
ALTER DATABASE CoffeeTransactions  
ADD FILEGROUP [CoffeeTransactions_inmem  
] CONTAINS MEMORY_OPTIMIZED_DATA
```

```
CREATE TABLE dbo.POSTransaction (
```

```
,  
UserId int NOT NULL  
INDEX x_UserId NONCLUSTERED  
HASH WITH (BUCKET_COUNT=900000),  
  
POSTransactionId int NOT NULL  
PRIMARY KEY CLUSTERED  
HASH WITH (BUCKET_COUNT=1000000)
```

```
POSTransactionId int NOT NULL,  
StatusID int NOT NULL,  
CreateDate datetime2 NOT NULL,  
Price money
```

```
WITH (MEMORY_OPTIMIZED=ON,  
DURABILITY=SCHEMA_ONLY)
```

```
ON [CoffeeTransactions_inmem]
```

Note:

*



MEMORY_OPTIMIZED_DATA

First create a memory-optimized data filegroup and add a container to the filegroup.

Then create a memory-optimized table.

*

You must specify a value for the BUCKET_COUNT parameter when you create the memory- optimized table. In most cases the bucket count should be between

1 and 2 times the number of distinct values in the index key.

*

Example:

-- create a durable (data will be persisted) memory-optimized table -- two of the columns are indexed

```
CREATE TABLE dbo.ShoppingCart (
```

```
ShoppingCartId INT IDENTITY(1,1) PRIMARY KEY NONCLUSTERED, UserId INT NOT NULL INDEX ix_UserId  
NONCLUSTERED HASH WITH
```

```
(BUCKET_COUNT=1000000),
```

```
CreatedDate DATETIME2 NOT NULL,
```

```
TotalPrice MONEY
```

```
) WITH (MEMORY_OPTIMIZED=ON)
```

```
GO
```

QUESTION 3

You have a Microsoft SQL Azure database.

You have the following stored procedure:

```
01 CREATE PROCEDURE UpdateContact  
02   @ContactID int,  
03   @LastName nvarchar(50)  
04 AS  
05  
06 SELECT LastName AS OriginalName  
07 FROM Person.Contact  
08  
09 WHERE ContactID = @ContactID;  
10 UPDATE Person.Contact  
11 SET LastName = @LastName  
12  
13 WHERE ContactID = @ContactID;
```



You discover that the stored procedure periodically fails to update Person.Contact.

You need to ensure that Person.Contact is always updated when UpdateContact executes. The solution must minimize the amount of time required for the stored procedure to execute and the number of locks held.

What should you do?

- A. Add the following line of code to line 12: WITH (UPDLOCK)
- B. Add the following line of code to line 05: SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
- C. Add the following line of code to line 08: WITH (UPDLOCK)
- D. Add the following line of code to line 05: SET TRANSACTION ISOLATION LEVEL SNAPSHOT

Correct Answer: C

*

Overall, you should use UPDLOCK when you read a value that you plan to update later in the same transaction to prevent the value from changing.

*

UPDLOCK

Specifies that update locks are to be taken and held until the transaction completes. UPDLOCK takes update locks for read operations only at the row-level or page-level. If UPDLOCK is combined with TABLOCK, or a table-level lock is taken

for some other reason, an exclusive (X) lock will be taken instead.

When UPDLOCK is specified, the READCOMMITTED and READCOMMITTEDLOCK isolation level hints are ignored. For example, if the isolation level of the session is set to SERIALIZABLE and a query specifies (UPDLOCK,

READCOMMITTED), the READCOMMITTED hint is ignored and the transaction is run using the SERIALIZABLE isolation level.

QUESTION 4

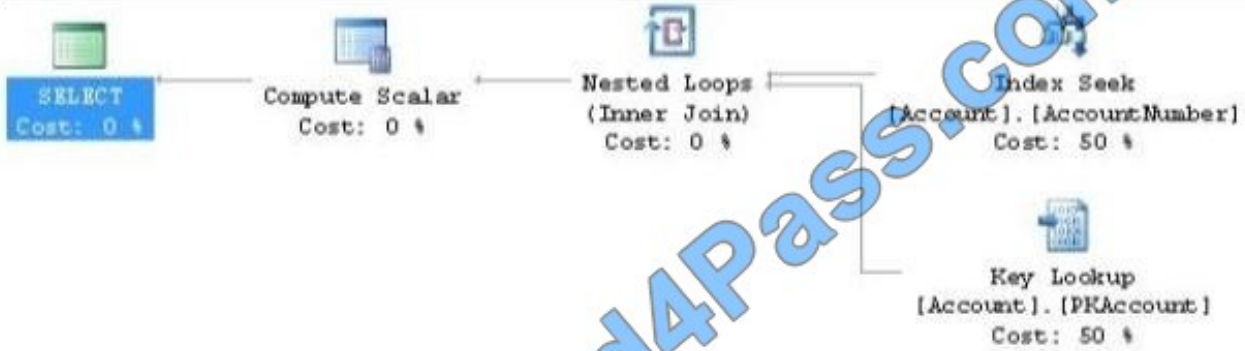
You administer an instance of SQL Server.

You are tasked with tuning a common set of queries. You have the results of several test executions, along with query plans. The schema and the data for all database object(s) used remain unchanged between executions. The QueryTime column is defined as a computed column that uses the GETDATE() system function. The query plans and results are shown below:



```
SELECT *
FROM dbo.Account
WHERE AccountNumber = 'A10000001'
```

Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [dbo].[Account] WHERE [AccountNumber]=@1

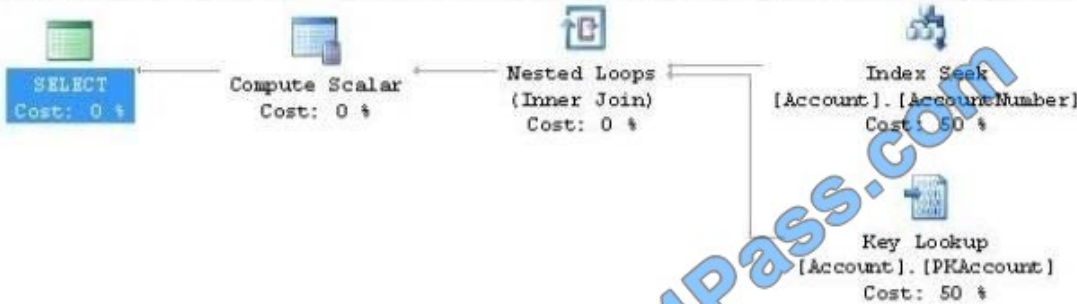


AccountID	AccountNumber	Name	QueryTime
-----	-----	-----	-----
0F63B176-7257-4480-9D0E-126C45 CEFFF1	A10000001	Don Hall	2014-01-29 18:01:50.923

```
SELECT *
FROM dbo.Account
WHERE AccountNumber IN( 'A10000001', 'Q88700323', 'R00000012' )
GO
```




Query 1: Query cost (relative to the batch): 100%
 SELECT * FROM dbo.Account WHERE AccountNumber IN('A10000001', 'Q88700323', 'R00000012')



AccountID	AccountNumber	Name	QueryTime
----- -----	----- -----	----- -----	----- -----
0F63B176-7257-4480-9D0E-126C45 CEFFF1	A10000001	Don Hall	2014-01-29 20:14:05.660
337227AA-3A4B-4B28-8E02-0ADEAD 06EA10	Q88700323	Darren Parker	2014-01-29 20:14:05.660
C4980E64-874E-4640-8826- BAF35D8FB845	R00000012	Carol Philips	2014-01-29 20:14:05.660

You need to make an initial diagnosis of the situation, based solely on this input

Which two statements can you make about the performance characteristics of this query? Each correct answer presents a complete solution. Choose two.

- A. The queries would perform better if the index named AccountNumber included the Name and QueryTime column.
- B. The queries would perform worse if the index named AccountNumber included the NameColumn.
- C. The queries would perform better if the index named AccountNumber included the Name column.
- D. The object Account is a table, with an index having a leading column of AccountNumber and a Clustered Index named PKAccount.
- E. The object Account is an indexed view, with an index having a leading column of AccountNumber and a Clustered Index named PKAccount.
- F. The object Account is a view, joining the Account-AccountNumber and Account.PKAccount objects together.

Correct Answer: BD

QUESTION 5

You need to create a function that will use a SELECT statement in ProductsByProductType.sql. Which code segment should you use to complete the function?



- A. `CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS @tblInvoices TABLE (ProductID bigint, ProductType varchar(11), CreationDate
date)
AS
INSERT INTO @tblInvoices`
- B. `CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS TABLE
AS
RETURN`
- C. `CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS @TblInvoices TABLE (ProductID bigint, ProductType varchar(11), CreationDate
date)
AS`
- D. `CREATE FUNCTION Production.fnProductsByProductType (@ProductType varchar(11))
RETURNS xml
AS
RETURN`

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: B

<http://msdn.microsoft.com/en-us/library/ms191320.aspx> <http://msdn.microsoft.com/en-us/library/ms186755.aspx>

[Latest 70-464 Dumps](#)

[70-464 VCE Dumps](#)

[70-464 Practice Test](#)



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](#)

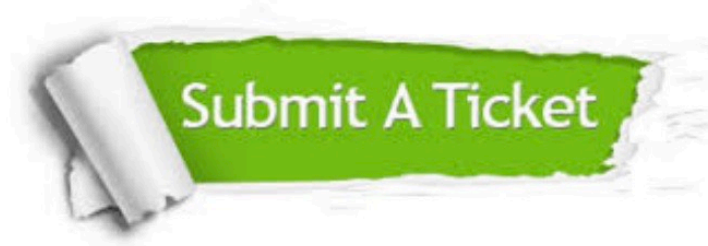
We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications. You can view Vendor list of All Certification Exams offered:

<https://www.lead4pass.com/allproducts>

Need Help

Please provide as much detail as possible so we can best assist you.

To update a previously submitted ticket:



 <p>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.</p>	 <p>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.</p>	 <p>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.</p>
---	---	--

Any charges made through this site will appear as Global Simulators Limited.

All trademarks are the property of their respective owners.

Copyright © lead4pass, All Rights Reserved.