

70-433 New Added Questions 100 Percent Pass Exam Guaranteed Free Download Provided By Braindump2go Now! (171-180)

Important News: Microsoft 70-433 Exam Questions are been updated recently! Braindumo2go offers the latest up-to-date 70-433 Dumps for free instant download which helps you pass 70-433 Exam in a short time! Our 70-433 Exam Dumps has two version: 70-433 PDF Dumps,70-433 VCE Dumps! Choose one of them according to your own need! 100% New 70-433 Exam Questions from Microsoft Official Exam Center! 100% Pass Microsoft 70-433 Exam! Exam Code: 70-433Exam Name: TS: Microsoft SQL Server 2008, Database DevelopmentCertification Provider: MicrosoftKeywords: 70-433 Exam Dumps,70-433 Practice Tests,70-433 Practice Exams,70-433 Exam Questions,70-433 PDF,70-433 VCE Free,70-433 Book,70-433 E-Book,70-433 Study Guide,70-433 Braindump,70-433 Prep Guide

Compared Before Buying Microsoft 70-433 PDF & VCE!		
Pass4sure	Braindump2go	Test King
	100% Pass OR Money Back	
202 Q&As - Practice	210 Q&As – Real Questions	202 Q&As - Practice
\$125.99	\$99.99	\$124.99
No Discount	Coupon Code: BDNT2014	No Discount

QUESTION 171 You are a database developer working on an application hosted on Microsoft SQL Server 2008 R2. The application regularly imports employee data from XML files. These files are bulk-loaded into the CompanyDoc column of the CompanyXML table. The table has the following definition:

```
CREATE TABLE CompanyXML
(CompanyXMLID int IDENTITY PRIMARY KEY NOT NULL,
 CompanyDoc xml NOT NULL)
```

One of the XML files loaded into this table has the following code fragment.

```
<Company>
  <Employee Name = "Michaeline" Department = "Finance"/>
  <Employee Name = "Chad" Department = "Facilities"/>
  ...
  <Employee Name = "Shelly" Department = "Human Resources"/>
</Company>
```

You need to be able to return the data as shown in the following table:

Name	Department
Michael	Finance
John	Facilities
David	Human Resources

Which Transact-SQL statements should you use?

- A. DECLARE @xml xml
 DECLARE @results table (Name varchar(20), Department varchar(20))
 DECLARE cur_xml CURSOR FAST FORWARD FOR
 SELECT CompanyDoc FROM CompanyXML
 OPEN cur_xml
 FETCH NEXT FROM cur_xml INTO @xml
 WHILE @@FETCH_STATUS = 0
 BEGIN
 INSERT @results
 SELECT Name = @xml.value('/Company[1]/Employee[1]/@Name', 'varchar(20)')
 , Department = @xml.value('/Company[1]/Employee[1]/@Department', 'varchar(20)')
 FETCH NEXT FROM cur_xml INTO @xml
 END
 CLOSE cur_xml
 DEALLOCATE cur_xml
 SELECT * FROM @results
- B. SELECT Name = Company.Employee.value('@Name', 'varchar(20)')
 , Department = Company.Employee.value('@Department', 'varchar(20)')
 FROM CompanyXML c
 CROSS APPLY CompanyDoc.nodes('/Company/Employee') Company(Employee)
- C. DECLARE @xml xml
 DECLARE @results table (Name varchar(20), Department varchar(20))
 DECLARE cur_xml CURSOR FAST FORWARD FOR
 SELECT CompanyDoc FROM CompanyXML
 OPEN cur_xml
 FETCH NEXT FROM cur_xml INTO @xml
 WHILE @@FETCH_STATUS = 0
 BEGIN
 INSERT @results
 SELECT Name = @xml.value('/Company/Employee/@Name', 'varchar(20)')
 , Department = @xml.value('/Company/Employee/@Department', 'varchar(20)')
 FETCH NEXT FROM cur_xml INTO @xml
 END
 CLOSE cur_xml
 DEALLOCATE cur_xml
 SELECT * FROM @results
- D. DECLARE @xmlid int
 DECLARE cur_xml CURSOR FAST FORWARD FOR
 SELECT CompanyXMLID FROM CompanyXML
 OPEN cur_xml
 FETCH NEXT FROM cur_xml INTO @xmlid
 WHILE @@FETCH_STATUS = 0
 BEGIN
 SELECT Name = Company.Employee.query('@Name', 'varchar(20)')
 , Department = Company.Employee.query('@Department', 'varchar(20)')
 FROM CompanyXML
 WHERE CompanyXMLID = @xmlid
 FETCH NEXT FROM cur_xml INTO @xmlid
 END
 CLOSE cur_xml
 DEALLOCATE cur_xml

A. Option AB. Option BC. Option CD. Option D Answer: C QUESTION 172 You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK_Product_ProductID and a non-clustered index named AK_Product_ProductNumber. Both indexes have been created on a single primary partition. The table has the following definition:

```
CREATE TABLE [Production].[Product] (
  [ProductID] [int] IDENTITY(1,1) NOT NULL,
  [Name] [nvarchar](50) NOT NULL,
  [ProductNumber] [nvarchar](25) NOT NULL,
  [Color] [nvarchar](15) NULL,
  [Class] [nchar](2) NULL,
  [Style] [nchar](2) NULL,
  [ModifiedDate] [datetime] NOT NULL,
  CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
  [ProductID] ASC
) ON [PRIMARY]
) ON [PRIMARY]
GO

The index has the following definition:

CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_ProductNumber] ON [Production].[Product]
(
  [ProductNumber] ASC
) ON [PRIMARY]
GO

The Production.Product table contains 1 million rows.
```

You want to ensure that data retrieval takes the minimum amount of time when the queries executed against the Production.Product table are ordered by product number or filtered by class. You observe that the average fragmentation for the AK_Product_ProductNumber index is 24 percent. You need to reduce fragmentation. You need to achieve this goal without blocking access to table data. Which Transact-SQL statement should you use?

- A. ALTER DATABASE (AdventureWorks) SET AUTO_UPDATE_STATISTICS ON
- B. UPDATE STATISTICS Production.Product
- C. UPDATE INDEX AK_Product_ProductNumber ON Production.Product SET (STATISTICS_IORECOMPUTE = ON)
- D. ALTER STATISTICS Production.Product
- E. ALTER INDEX AK_Product_ProductNumber ON Production.Product REBUILD Partition = 1
- F. ALTER INDEX AK_Product_ProductNumber ON Production.Product REORGANIZE
- G. EXEC sys.sp_configure 'index create memory', 1
- H. SELECT * FROM STATS WHERE name='AK_Product_ProductNumber'
- I. SELECT * FROM sys.dm_db_index_physical_stats (DB_ID(), OBJECT_ID('Production.Product'), NULL, NULL, NULL)
- J. SELECT * FROM sys.dm_db_index_operational_stats (DB_ID(), OBJECT_ID('Production.Product'), NULL, NULL)
- K. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WHERE Class <> null
- L. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WHERE Class is not null
- M. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WITH SAMPLE 100 PERCENT
- N. DBCC SHOW_STATISTICS ('Production.Product', AK_Product_ProductNumber)

A. Option AB. Option BC. Option CD. Option DE. Option EF. Option FG. Option GH. Option HI. Option IJ. Option JK. Option KL. Option LM. Option MN. Option NO. Option OP. Option PQ. Option Q Answer: G QUESTION 173 You administer a Microsoft SQL Server 2008 database named AdventureWorks that contains a table named Production.Product. The table contains a primary key named PK_Product_ProductID and a non-clustered index named AK_Product_ProductNumber. Both indexes have been created on a single primary partition. The table has the following definition:

```
CREATE TABLE [Production].[Product] (
  [ProductID] [int] IDENTITY(1,1) NOT NULL,
  [Name] [nvarchar](50) NOT NULL,
  [ProductNumber] [nvarchar](25) NOT NULL,
  [Color] [nvarchar](15) NULL,
  [Class] [nchar](2) NULL,
  [Style] [nchar](2) NULL,
  [ModifiedDate] [datetime] NOT NULL,
  CONSTRAINT [PK_Product_ProductID] PRIMARY KEY CLUSTERED
(
  [ProductID] ASC
) ON [PRIMARY]
) ON [PRIMARY]
GO

The index has the following definition:

CREATE UNIQUE NONCLUSTERED INDEX [AK_Product_ProductNumber] ON [Production].[Product]
(
  [ProductNumber] ASC
) ON [PRIMARY]
GO

The Production.Product table contains 1 million rows.
```

You want to ensure that data retrieval takes the minimum amount of time when the queries executed against the Production.Product table are ordered by product number or filtered by class. You need to find out the degree of fragmentation for the indexes on the

Production.Product table. Which Transact-SQL statement should you use?

- A. ALTER DATABASE [AdventureWorks] SET AUTO_UPDATE_STATISTICS ON
- B. UPDATE STATISTICS Production.Product
- C. UPDATE INDEX AK_Product_ProductNumber ON Production.Product SET (STATISTICS_NORECOMPUTE = ON)
- D. ALTER STATISTICS Production.Product
- E. ALTER INDEX AK_Product_ProductNumber ON Production.Product REBUILD
- F. ALTER INDEX AK_Product_ProductNumber ON Production.Product REBUILD PARTITION = 1
- G. ALTER INDEX AK_Product_ProductNumber ON Production.Product REORGANIZE
- H. EXEC sp_configure 'index create memory', 1
- I. SELECT * FROM SYS.STATS WHERE name='AK_Product_ProductNumber'
- J. SELECT * FROM SYS.STATS WHERE name='AK_Product_ProductNumber'
- K. SELECT * FROM SYS.STATS WHERE name='AK_Product_ProductNumber'
- L. SELECT * FROM sys.dm_db_index_physical_state (DB_ID(), OBJECT_ID('Production.Product'), NULL, NULL, NULL)
- M. SELECT * FROM sys.dm_db_index_operational_state (DB_ID(), OBJECT_ID('Production.Product'), NULL, NULL)
- N. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WHERE Class <> null
- O. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WHERE Class IS NOT NULL
- P. CREATE STATISTICS ProductClass_State ON Production.Product (Name, ProductNumber, Class) WHERE Class IS NOT NULL WITH SAMPLE 100 PERCENT
- Q. SHOW STATISTICS ('Production.Product', AK_Product_ProductNumber)

A. Option AB. Option BC. Option CD. Option DE. Option EF. Option FG. Option GH. Option HI. Option IJ. Option JK. Option KL. Option LM. Option MN. Option NO. Option OP. Option PQ. Option Q Answer: L QUESTION 174 Drag and Drop Question You administer a Microsoft SQL Server 2008 instance that has two databases. The first database named AdventureWorks contains a table named Sales.SalesOrders. The Sales.SalesOrders table has the following definition:

```
CREATE TABLE [Sales].[SalesOrderDetail]
([SalesOrderDetailID] [int] NOT NULL,
[ProductID] [int] NOT NULL,
[OrderQty] [smallint] NOT NULL,
[OrderYear] [int] NOT NULL,
CONSTRAINT [PK_SalesOrderDetail] PRIMARY KEY CLUSTERED
([SalesOrderDetailID]
)) ON [PRIMARY]
```

The second database named AdventureWorksDW contains a table named dbo.SalesOrderSummary. The dbo.SalesOrderSummary table has the following definition:

```
CREATE TABLE [dbo].[SalesOrderSummary] (
ProductID [int] NOT NULL,
OrderQty [int] NOT NULL,
OrderYear [int] NOT NULL,
CONSTRAINT [PK_SalesOrderSummary] PRIMARY KEY CLUSTERED
(
OrderYear ASC,
ProductID ASC
)
) ON [PRIMARY]
```

You plan to migrate sales data for the year 2011 from the SalesOrderDetail table into the SalesOrderSummary table. You need to ensure that the following requirements are met: Which three Transact-SQL statements should you use? (To answer, move the appropriate statements from the list of statements to the answer area and arrange them in the correct order.)

```
USE AdventureWorks
GO

USE AdventureWorksDW
GO

DELETE TABLE dbo.SalesOrderSummary
GO

TRUNCATE TABLE dbo.SalesOrderSummary
GO

SELECT
ProductID,
SUM(OrderQty) AS OrderQty,
YEAR(OrderDate) AS OrderYear
INTO
dbo.SalesOrderSummary
FROM
AdventureWorks.Sales.SalesOrderDetail
WHERE
YEAR(OrderDate)=2011
GROUP BY
ProductID,
YEAR(OrderDate)
GO

SELECT
ProductID,
SUM(OrderQty) AS OrderQty,
YEAR(2011) AS OrderYear
INTO
dbo.SalesOrderSummary
FROM
AdventureWorks.Sales.SalesOrderDetail
GROUP BY
ProductID
HAVING
YEAR(2011)=2011
GO

INSERT dbo.SalesOrderSummary
(ProductID, OrderQty, OrderYear)
SELECT
ProductID,
SUM(OrderQty) AS OrderQty,
YEAR(OrderDate) AS OrderYear
FROM
AdventureWorks.Sales.SalesOrderDetail
WHERE
YEAR(OrderDate)=2011
GROUP BY
ProductID,
YEAR(OrderDate)
GO

INSERT dbo.SalesOrderSummary
(ProductID, OrderQty, OrderYear)
SELECT
ProductID,
SUM(OrderQty),
YEAR(2011)
FROM
AdventureWorks.Sales.SalesOrderDetail
GROUP BY
ProductID
HAVING
YEAR(2011)=2011
GO
```

Answer:

<pre>USE AdventureWorks GO</pre>	<pre>USE AdventureWorksDW GO</pre>
<pre>USE AdventureWorksDW GO</pre>	
<pre>DELETE TABLE dbo.SalesOrderSummary GO</pre>	<pre>TRUNCATE TABLE dbo.SalesOrderSummary GO</pre>
<pre>TRUNCATE TABLE dbo.SalesOrderSummary GO</pre>	
<pre>SELECT ProductID, SUM(OrderQty) AS OrderQty, YEAR(OrderDate) AS OrderYear INTO dbo.SalesOrderSummary FROM AdventureWorks.Sales.SalesOrderDetail WHERE YEAR(OrderDate)=2011 GROUP BY ProductID, YEAR(OrderDate) GO</pre>	<pre>INSERT dbo.SalesOrderSummary (ProductID, OrderQty, OrderYear) SELECT ProductID, SUM(OrderQty) AS OrderQty, YEAR(OrderDate) AS OrderYear FROM AdventureWorks.Sales.SalesOrderDetail WHERE YEAR(OrderDate)=2011 GROUP BY ProductID, YEAR(OrderDate) GO</pre>
<pre>SELECT ProductID, SUM(OrderQty) AS OrderQty, YEAR(2011) AS OrderYear INTO dbo.SalesOrderSummary FROM AdventureWorks.Sales.SalesOrderDetail GROUP BY ProductID HAVING YEAR(2011)=2011 GO</pre>	
<pre>INSERT dbo.SalesOrderSummary (ProductID, OrderQty, OrderYear) SELECT ProductID, SUM(OrderQty) AS OrderQty, YEAR(OrderDate) AS OrderYear FROM AdventureWorks.Sales.SalesOrderDetail WHERE YEAR(OrderDate)=2011 GROUP BY ProductID, YEAR(OrderDate) GO</pre>	
<pre>INSERT dbo.SalesOrderSummary (ProductID, OrderQty, OrderYear) SELECT ProductID, SUM(OrderQty), YEAR(2011) FROM AdventureWorks.Sales.SalesOrderDetail GROUP BY ProductID HAVING YEAR(2011)=2011 GO</pre>	

QUESTION 175 You administer a Microsoft SQL Server 2008 R2 database that has a table named Customer. The table has the following definition:

```
CREATE TABLE Customer
(CustomerID int NOT NULL PRIMARY KEY,
FirstName varchar(255) NOT NULL,
LastName varchar(255) NOT NULL,
CustomerAddress varchar(1024))
```

The database also has a table named CustomerExclusionList. Data will be added to the CustomerExclusionList table regularly. The CustomerExclusionList table has the following definition:

```
CREATE TABLE CustomerExclusionList
(CustomerID int NOT NULL,
LastName varchar(255) NOT NULL)
```

You need to create a view that returns all records and columns of the Customer table that are not present in the CustomerExclusionList table. Which Transact-SQL statement should you use?

- A. CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
c.FirstName,
c.LastName,
c.CustomerAddress
FROM Customer c
INNER JOIN CustomerExclusionList cel
ON c.Firstname = cel.FirstName
INNER JOIN CustomerExclusionList cel
ON c.LastName = cel.LastName
- B. CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
c.FirstName,
c.LastName,
c.CustomerAddress
FROM Customer c
LEFT OUTER JOIN CustomerExclusionList cel
ON c.Firstname = cel.FirstName
AND c.LastName = cel.LastName
WHERE cel.FirstName IS NULL
- C. CREATE VIEW vw_ValidCustomer
AS
SELECT CustomerID,
FirstName,
LastName,
CustomerAddress
FROM Customer c
EXCEPT
SELECT CustomerID,
FirstName,
LastName,
CustomerAddress
FROM CustomerExclusionList
- D. CREATE VIEW vw_ValidCustomer
AS
SELECT c.CustomerID,
c.FirstName,
c.LastName,
c.CustomerAddress
FROM Customer c
INNER JOIN CustomerExclusionList cel
ON c.Firstname = cel.FirstName
AND c.LastName = cel.LastName

A. Option AB. Option BC. Option CD. Option D Answer: C QUESTION 176 You are a database developer writing reports for a sales management application. A customer table has the following definition:

```
CREATE TABLE customer  
(CustomerID INT,  
FirstName VARCHAR(30),  
LastName VARCHAR(50),  
City VARCHAR(100),  
[State] VARCHAR(25),  
PostalCode VARCHAR(5));
```

An order table has the following definition:

```
CREATE TABLE [order]  
(OrderID INT,  
CustomerID INT,  
OrderDate DATETIME);
```

You need to write a report that contains the following columns:

Column name	Description
CustomerID	The CustomerID
FullName	Concatenated first and last names
PostalCode	Customer's Postal Code
OrderCount	Number of orders for this customer
EarliestOrderDate	The OrderDate from the earliest (by OrderDate) order for this customer

You also need to ensure that the report meets the following requirements: Which Transact-SQL query should you use?

```

A. SELECT c.CustomerID,
    c.FirstName + ' ' + c.LastName
    c.PostalCode,
    COUNT(*) AS OrderCount,
    MIN(o.OrderDate) AS EarliestOrderDate
FROM Customer c
INNER JOIN [order] o ON c.CustomerID = o.CustomerID
WHERE c.PostalCode LIKE '894'
GROUP BY c.CustomerID, c.FirstName + ' ' + c.LastName
ORDER BY c.CustomerID

B. SELECT c.CustomerID,
    c.FirstName + ' ' + c.LastName
    c.PostalCode,
    COUNT(*) AS OrderCount,
    MIN(o.OrderDate) AS EarliestOrderDate
FROM Customer c
LEFT OUTER JOIN [order] o ON c.CustomerID = o.CustomerID
WHERE c.PostalCode LIKE '894'
GROUP BY c.CustomerID, c.FirstName + ' ' + c.LastName
ORDER BY c.CustomerID

C. SELECT c.CustomerID,
    c.FirstName + ' ' + c.LastName
    c.PostalCode,
    COUNT(*) AS OrderCount,
    MIN(o.OrderDate) AS EarliestOrderDate
FROM Customer c
INNER JOIN [order] o ON c.CustomerID = o.CustomerID
WHERE c.PostalCode LIKE '894'
ORDER BY c.CustomerID

D. SELECT c.CustomerID,
    c.FirstName + ' ' + c.LastName
    c.PostalCode,
    COUNT(*) AS OrderCount,
    MIN(o.OrderDate) AS EarliestOrderDate
FROM Customer c
INNER JOIN [order] o ON c.CustomerID = o.CustomerID
WHERE c.PostalCode = '894'
GROUP BY c.CustomerID, c.FirstName + ' ' + c.LastName
ORDER BY FullName
    
```

A. Option AB. Option BC. Option CD. Option D Answer: A QUESTION 177A table named Contacts includes a column named SmtAddress. You must develop a report that returns e-mail addresses from the Contacts table that have the following format: at least one character, the at sign (@), at least one character, and then ".org". You need to return data that meets the requirements. Which Transact-SQL statement should you use? A. Select * from Contacts where SmtAddress like '%@%[.]org'B. Select * from Contacts where SmtAddress like '%@%.org'C. Select * from Contacts where SmtAddress like '%@_%.org'D. Select * from Contacts where SmtAddress like '%@%.org' Answer: B QUESTION 178You administer a Microsoft SQL Server 2008 database that contains a table named dbo.[order]. There are no triggers on the table. You plan to create a stored procedure that will have the following parameters: You need to ensure that the following requirements are met: Which Transact-SQL statement should you use?

```

A. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);
UPDATE [order]
SET
    ProdID = @CustId
OUTPUT #INSERTED.OrderID, #INSERTED.ProdID INTO @OrderIDs
WHERE
    CustID = @CustId;

B. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);
UPDATE dbo.[order]
SET
    ProdID = @ProdId
OUTPUT INSERTED.OrderID, INSERTED.ProdID INTO @OrderIDs
WHERE
    CustID = @CustId;

C. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);
UPDATE dbo.[order]
SET
    ProdID = @ProdId
OUTPUT DELETED.OrderID, DELETED.ProdID INTO @OrderIDs
WHERE
    CustID = @CustId;

D. DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);
UPDATE dbo.[order]
SET
    ProdID = @ProdId
OUTPUT SELECT d.OrderID, d.ProdID FROM DELETED d INTO @OrderIDs
WHERE
    CustID = @CustId;
    
```

A. Option AB. Option BC. Option CD. Option D Answer: A QUESTION 179You administer a SQL Server 2008 instance. The instance contains a database table named Sales.SalesOrderDetail. The table has the following definition:

```
CREATE TABLE [Sales].[SalesOrderDetail](
    [SalesOrderID] [int] NOT NULL,
    [SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
    [CarrierTrackingNumber] [nvarchar](25) NULL,
    [OrderQty] [smallint] NOT NULL,
    [ProductID] [int] NOT NULL,
    [SpecialOfferID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [UnitPriceDiscount] [money] NOT NULL,
    [LineTotal] AS ([smallint](((UnitPrice)*((1.0)-(UnitPriceDiscount)))*[OrderQty],(0.0))),
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
    CONSTRAINT PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID PRIMARY KEY CLUSTERED
) ON [PRIMARY]
WITH (DATA_COMPRESSION = ROW) ON ([PRIMARY])
) ON ([PRIMARY])
GO

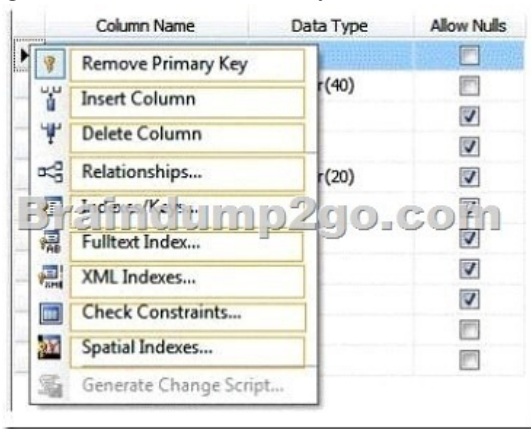
The table includes the following index:

CREATE NONCLUSTERED INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrderDetail]
(
    [ProductID] ASC
) ON ([PRIMARY])
GO
```

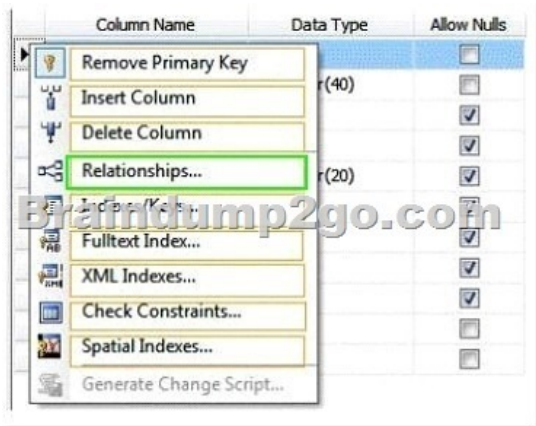
You need to enable row compression for the index. Which Transact-SQL statement or statements should you use?

- A. CREATE NONCLUSTERED INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrderDetail] ([ProductID] ASC) WITH (DATA_COMPRESSION = ROW, SORT_IN_ORDER = ON) GO
- B. ALTER INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrderDetail] REBUILD WITH (ALLOW_ROW_LOCKS = ON, MAX_ROW_SIZE = 8096, SORT_IN_ORDER = ON, DATA_COMPRESSION = ROW) GO
- C. ALTER INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrderDetail] REBUILD WITH (DATA_COMPRESSION = ROW) GO
- D. ALTER INDEX [IX_SalesOrderDetail_ProductID] ON [Sales].[SalesOrderDetail] REORGANIZE WITH (DATA_COMPRESSION = ON) GO

A. Option AB. Option BC. Option CD. Option D Answer: B QUESTION 180 Hotspot Question You administer a Microsoft SQL Server 2008 database that contains two tables named Products and Suppliers. You want to implement referential integrity between the Products and Suppliers tables. You want to create a new Foreign Key constraint on the Products table. The new Foreign Key constraint must meet the following requirements: You need to be able to create a new Foreign Key constraint by using Microsoft SQL Server Management Studio. What should you do? (To answer, configure the appropriate option or options in the dialog box in the answer area.)



Answer:



100% 70-433 Complete Success & Money Back Guarantee! By utilizing Braindump2go high quality Microsoft 70-433 Exam

Dumps Products, You can surely pass 70-433 certification 100%! Braindump2go also offers 100% money back guarantee to individuals in case they fail to pass Microsoft 70-433 in one attempt.

Compared Before Buying Microsoft 70-433 PDF & VCE!

Pass4sure	Braindump2go 100% Pass OR Money Back	Test King
202 Q&As - Practice	210 Q&As – Real Questions	202 Q&As - Practice
\$125.99	\$99.99	\$124.99
No Discount	Coupon Code: BDNT2014	No Discount

<http://www.braindump2go.com/70-433.html>