

[2016-Oct.-NewBraindump2go 70-461 Exam VCE 179q&as [NQ95-NQ100 Download

2016/10 New Microsoft 70-461: Querying Microsoft SQL Server 2012/2014 Exam Questions Updated Today! Free Instant Download 70-461 Exam Dumps (PDF & VCE) 179Q&As from Braindump2go.com Today! **100% Real Exam Questions! 100% Exam Pass Guaranteed!** 1. | 2016/10 Latest 70-461 Exam Dumps (PDF & VCE) 179Q&As Download: <http://www.braindump2go.com/70-461.html> 2. | 2016/10 Latest 70-461 Exam Questions & Answers: <https://drive.google.com/folderview?id=0B75b5xYLjSSNflp4NUtxTHJkb0hXTWtYMmdnbjBpVjNqUVV3NjNDcW1qOWVLMUQ3cUpENU0&usp=sharing> QUESTION 95 You create the following stored procedure. (Line numbers are included for reference only.)

```
01 CREATE PROCEDURE dbo.InsertCountryRegion
02     @CountryRegionCode nvarchar(3),
03     @Name nvarchar(50)
04 AS
05 BEGIN
06     SET NOCOUNT ON;
07     ...
08 END;
```

You need to ensure that the stored procedure performs the following tasks: - If a record exists, update the record. - If no record exists, insert a new record. Which four Transact-SQL statements should you insert at line 07? (To answer, move the appropriate statements from the list of statements to the answer area and arrange them in the correct order.)

```
UPDATE CountryRegion
SET Name = @Name
WHERE CountryRegionCode = @CountryRegionCode

WHEN NOT MATCHED BY SOURCE THEN
...

WHEN NOT MATCHED BY TARGET THEN
...

WHEN MATCHED THEN UPDATE SET Name =
source.Name

MERGE CountryRegion AS target
USING (SELECT @CountryRegionCode, @Name)
AS source (CountryRegionCode, Name)
ON (target.CountryRegionCode =
source.CountryRegionCode)

IF (@@ROWCOUNT > 0)

INSERT INTO CountryRegion
(CountryRegionCode, Name)
VALUES (@CountryRegionCode, @Name);

INSERT (CountryRegionCode, Name)
VALUES (source.CountryRegionCode,
source.Name);
```

Answer:

```
MERGE CountryRegion AS target
USING (SELECT @CountryRegionCode, @Name)
AS source (CountryRegionCode, Name)
ON (target.CountryRegionCode =
source.CountryRegionCode)

WHEN NOT MATCHED BY SOURCE THEN
...

WHEN NOT MATCHED BY TARGET THEN
...

WHEN MATCHED THEN UPDATE SET Name =
source.Name
```

Explanation: <http://technet.microsoft.com/en-us/library/bb510625.aspx> QUESTION 96 You use Microsoft SQL Server 2012 to develop a database that has two tables named Div1Cust and Div2Cust. Each table has columns named DivisionID and CustomerID. None of the rows in Div1Cust exist in Div2Cust. You need to write a query that meets the following requirements: - The rows in Div1Cust must be combined with the rows in Div2Cust. - The result set must have columns named Division and Customer. - Duplicates must be retained. 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.)

```
EXCEPT

SELECT DivisionID, CustomerID
FROM Div2Cust

SELECT DISTINCT DivisionID, CustomerID
FROM Div1Cust, Div2Cust

INTERSECT

SELECT DivisionID AS Division, CustomerID AS
Customer
FROM Div1Cust, Div2Cust

INNER JOIN

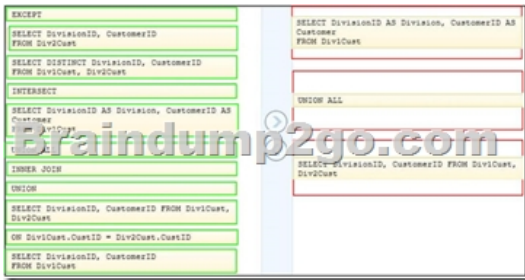
UNION

SELECT DivisionID, CustomerID FROM Div1Cust,
Div2Cust

ON Div1Cust.CustomerID = Div2Cust.CustomerID

SELECT DivisionID, CustomerID
FROM Div1Cust
```

Answer:



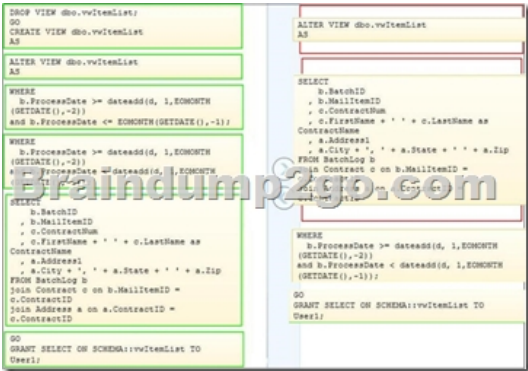
Explanation: <http://msdn.microsoft.com/en-us/library/ms180026.aspx> <http://msdn.microsoft.com/en-us/library/ms188055.aspx>
 QUESTION 97 You create a view based on the following statement:

```
CREATE VIEW dbo.vwItemList
AS
SELECT
    b.BatchID
    , b.MailItemID
    , c.ContractNum
    , c.FirstName + ' ' + c.LastName as ContractName
    , a.Address1
    , a.City + ', ' + a.State + ' ' + a.Zip
FROM BatchLog b
join Contract c on b.MailItemID = c.ContractID
join Address a on a.ContractID = c.ContractID
WHERE
    b.ProcessDate >= dateadd(d, 1, EOMONTH(GETDATE(), -2));
```

You grant the Select permission to User1 for this view. You need to change the view so that it displays only the records that were processed in the month prior to the current month. You need to ensure that after the changes, the view functions correctly for User1. Which four Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)



Answer:



Explanation: <http://msdn.microsoft.com/en-us/library/hh213020.aspx> <http://msdn.microsoft.com/en-us/library/ms186819.aspx> <http://msdn.microsoft.com/en-us/library/ms173846.aspx>
 QUESTION 98 You use Microsoft SQL Server 2012 to develop a database application. You create a table by using the following definition: CREATE TABLE Prices (PriceId int IDENTITY(1,1) PRIMARY KEY, ActualPrice NUMERIC(16,9), PredictedPrice NUMERIC(16,9)) You need to create a computed column based on a user-defined function named udf_price_index. You also need to ensure that the column supports an index. Which three Transact-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

```

CREATE FUNCTION udf_price_index
    (@Actualprice FLOAT, @Predictedprice
    FLOAT)
    RETURNS FLOAT
AS
ALTER TABLE Prices ADD (PriceIndex)
AS dbo.udf_price_index(@ActualPrice,
[PredictedPrice]) PERSISTED
ALTER TABLE Prices ADD (PriceIndex)
AS dbo.udf_price_index(@ActualPrice,
[PredictedPrice])
AS
BEGIN
    SELECT @PriceIndex = CASE
        WHEN @Predictedprice = 0 THEN 0
        ELSE @Actualprice/@Predictedprice
    END
RETURN @PriceIndex
END
GO

CREATE FUNCTION udf_price_index
    (@Actualprice NUMERIC(14,9),
    @Predictedprice NUMERIC(14,9))
    RETURNS NUMERIC(14,9)
    WITH SCHEMABINDING
AS
BEGIN
    DECLARE @PriceIndex NUMERIC(14,9)
    SELECT @PriceIndex = CASE
        WHEN @Predictedprice = 0 THEN 0
        ELSE @Actualprice/@Predictedprice
    END
RETURN @PriceIndex
END
GO
    
```

Answer:

```

CREATE FUNCTION udf_price_index
    (@Actualprice FLOAT, @Predictedprice
    FLOAT)
    RETURNS FLOAT
AS
ALTER TABLE Prices ADD (PriceIndex)
AS dbo.udf_price_index(@ActualPrice,
[PredictedPrice]) PERSISTED
ALTER TABLE Prices ADD (PriceIndex)
AS dbo.udf_price_index(@ActualPrice,
[PredictedPrice])
AS
BEGIN
    SELECT @PriceIndex = CASE
        WHEN @Predictedprice = 0 THEN 0
        ELSE @Actualprice/@Predictedprice
    END
RETURN @PriceIndex
END
GO

CREATE FUNCTION udf_price_index
    (@Actualprice NUMERIC(14,9),
    @Predictedprice NUMERIC(14,9))
    RETURNS NUMERIC(14,9)
    WITH SCHEMABINDING
AS
BEGIN
    DECLARE @PriceIndex NUMERIC(14,9)
    SELECT @PriceIndex = CASE
        WHEN @Predictedprice = 0 THEN 0
        ELSE @Actualprice/@Predictedprice
    END
RETURN @PriceIndex
END
GO
    
```

QUESTION 99 You use a Microsoft SQL Server 2012 database. You need to create an indexed view within the database for a report that displays Customer Name and the total revenue for that customer. Which four T-SQL statements should you use? (To answer, move the appropriate SQL statements from the list of statements to the answer area and arrange them in the correct order.)

Answer:

```

CREATE VIEW Sales.vwCustomerRevenue
AS
WITH SCHEMABINDING
CREATE VIEW
Sales.vwCustomerRevenue
WITH SCHEMABINDING
AS
SELECT
    C.CustomerID
    , C.CustomerName
    , SUM(O.SubTotal) as CustomerTotal
    , COUNT_BIG(*) as RecCount
FROM Sales.Customer as C on C.CustomerID
= O.CustomerID
GROUP BY
    C.CustomerID
    , C.CustomerName
GO
CREATE UNIQUE CLUSTERED INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
GO
CREATE UNIQUE INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
    
```

Explanation: <http://msdn.microsoft.com/en-us/library/ms191432.aspx> QUESTION 100 You administer a Microsoft SQL Server 2012 database. You use an OrderDetail table that has the following definition:

```

CREATE TABLE [dbo].[OrderDetail]
(
    [SalesOrderID] [int] NOT NULL,
    [SalesOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
    [ProductID] [int] NOT NULL,
    [SpecialOfferID] [int] NULL,
    [UnitPrice] [money] NOT NULL);
    
```

You need to create a non-clustered index on the SalesOrderID column in the OrderDetail table to include only rows that contain a value in the SpecialOfferID column. Which four 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.)

Where	
FILTER ON	
Special Offer ID is not NULL	
ON dbo.OrderDetail(SalesOrderID)	
ON dbo.OrderDetail(SalesOrderID) AS FILTERED_INDEX	
CREATE NONCLUSTERED INDEX Findx_SpecialOfferID	
CREATE NONCLUSTERED FILTERED INDEX Findx_SpecialOrderID	

Answer:

Where	CREATE NONCLUSTERED INDEX Findx_SpecialOfferID
FILTER ON	ON dbo.OrderDetail(SalesOrderID)
Special Offer ID is not NULL	Where
ON dbo.OrderDetail(SalesOrderID)	Special Offer ID is not NULL
ON dbo.OrderDetail(SalesOrderID) AS FILTERED_INDEX	
CREATE NONCLUSTERED INDEX Findx_SpecialOfferID	
CREATE NONCLUSTERED FILTERED INDEX Findx_SpecialOrderID	

!!!RECOMMEND!!! 1. [Braindump2go|2016/10 Latest 70-461 Exam Dumps \(PDF & VCE\) 179Q&As Download:](http://www.braindump2go.com/2016/10/Latest-70-461-Exam-Dumps-(PDF-&-VCE)-179Q&As-Download/)
<http://www.braindump2go.com/70-461.html> 2. [Braindump2go|2016/10 Latest 70-461 Exam Questions & Answers:](https://drive.google.com/folderview?id=0B75b5xYLjSSNflp4NUtxTHJkb0hXTWtYMmdnbjBpVjNqUVV3NjNDcW1qOWVLMUQ3cUpENU0&usp=sharing)
<https://drive.google.com/folderview?id=0B75b5xYLjSSNflp4NUtxTHJkb0hXTWtYMmdnbjBpVjNqUVV3NjNDcW1qOWVLMUQ3cUpENU0&usp=sharing>