**VCE & PDF**
**Pass4Lead.com**

# 70-461<sup>Q&As</sup>

## Querying Microsoft SQL Server 2012/2014

## Pass Microsoft 70-461 Exam with 100% Guarantee

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

**https://www.pass4lead.com/70-461.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**

What is the difference between the COUNT(*) aggregate function and the COUNT() general set function?

A. COUNT(*) counts rows; COUNT() counts rows where is not NULL.

B. COUNT(*) counts columns; COUNT() counts rows.

C. COUNT(*) returns a BIGINT; COUNT() returns an INT.

D. There\\'s no difference between the functions.

Correct Answer: A

**QUESTION 2**

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.)

Select and Place:

```
CREATE VIEW Sales.vwCustomerRevenue
AS
WITH SCHEMABINDING

CREATE VIEW
Sales.vwCustomerRevenue
WITH SCHEMABINDING
AS

SELECT
  O.CustomerID
    , C.CustomerName
    , SUM(O.SubTotal) as CustomerTotal
    , COUNT_BIG(*) as RecCount
FROM Sales.SalesOrderHeader AS O
JOIN Sales.Customer as C on C.CustomerID =
O.CustomerID

GROUP BY
  O.CustomerID
    , C.CustomerName

GO
CREATE UNIQUE CLUSTERED INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);

GO
CREATE UNIQUE INDEX idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

Correct Answer:

```
CREATE VIEW Sales.vwCustomerRevenue
AS
WITH SCHEMABINDING
```

```
CREATE VIEW
Sales.vwCustomerRevenue
WITH SCHEMABINDING
AS
```

```
SELECT
  O.CustomerID
   , C.CustomerName
   , SUM(O.SubTotal) as CustomerTotal
   , COUNT_BIG(*) as RecCount
FROM Sales.SalesOrderHeader AS O
JOIN Sales.Customer as C on C.CustomerID =
O.CustomerID
```

```
GO
CREATE UNIQUE CLUSTERED INDEX
idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

```
GROUP BY
  O.CustomerID
   , C.CustomerName
```

```
GO
CREATE UNIQUE INDEX idx_vwCustomerRevenue
ON Sales.vwCustomerRevenue (CustomerID);
```

**QUESTION 3**

Your database contains tables named Products and ProductsPriceLog. The Products table contains columns named ProductCode and Price. The ProductsPriceLog table contains columns named ProductCode, OldPrice, and NewPrice. The ProductsPriceLog table stores the previous price in the OldPrice column and the new price in the NewPrice column. You need to increase the values in the Price column of all products in the Products table by 5 percent. You also need to log the changes to the ProductsPriceLog table. Which Transact-SQL query should you use?

A. UPDATE Products SET Price = Price * 1.05 OUTPUT inserted.ProductCode, deleted.Price, inserted.Price INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

B. UPDATE Products SET Price = Price * 1.05 OUTPUT inserted.ProductCode, inserted.Price, deleted.Price INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

C. UPDATE Products SET Price = Price * 1.05 OUTPUT inserted.ProductCode, deleted.Price, inserted.Price * INTO ProductsPriceLog(ProductCode, OldPrice, NewPrice)

D. UPDATE Products SET Price = Price * 1.05 INSERT INTO ProductsPriceLog (ProductCode, CldPnce, NewPrice; SELECT ProductCode, Price, Price * 1.05 FROM Products

Correct Answer: A

http://msdn.microsoft.com/en-us/library/ms177564.aspx

**QUESTION 4**

You develop an SQL Server database. The database contains a table that is defined by the following T-SQL statements:

```
CREATE TABLE Employees
(employeeNumber INT,
    surName VARCHAR(100),
    givenName VARCHAR(25),
    dateOfBirth DATE,
    workPhone VARCHAR(12));
```

The table contains duplicate records based on the combination of values in the surName, givenName, and dateOfBirth fields.

You need to remove the duplicate records.

How should you complete the relevant Transact-SQL statements? To answer, drag the appropriate code segment or segments to the correct location or locations in the answer area. Each code 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.

Select and Place:

**Transact-SQL segments**

| Row_number() |
| Rank() |
| PARTITION BYsurName |
| PARTITION BY employeeNumber |
| ORDER BY |
| GROUP BY |
| DELETE FROM CTE WHERE Ct > 1 |
| DELETE FROM CTE WHERE Ct= 1 |

**Answer Area**

```
WITH CTE
    AS (
        SELECT surName,
                givenName,
                DateOfBirth,
                [Transact-SQL segment]
        Over (
            [Transact-SQL segment]  , givenName, DateOfBirth
            [Transact-SQL segment]  (SELECT 1)) AS Ct
        FROM   dbo.Employees)
    [Transact-SQL segment]
```

Correct Answer:

**Transact-SQL segments**

| Rank() |

| PARTITION BY employeeNumber |

| GROUP BY |

| DELETE FROM CTE WHERE Ct= 1 |

**Answer Area**

```
WITH CTE
    AS (
        SELECT surName,
            givenName
            DateOfBirth,
            Row_number()
    Over
        PARTITION BYsurName                    givenName, DateOfBirth
        ORDER BY                               ELECT 1)) AS Ct
    FROM    dbo.Employees)
DELETE FROM CTE WHERE Ct > 1
```

let us write a query which will delete all duplicate data in one shot. We will use a CTE (Common Table Expression) for this purpose. We will read in future posts what a CTE is and why it is used. On a lighter note, CTE\\'s can be imagined as equivalent to temporary result sets that can be used only in an underlying SELECT, INSERT, UPDATE, DELETE or CREATE VIEW statement. ;WITH CTE AS ( SELECT Name , City , [State] , ROW_NUMBER() OVER(PARTITION BY Name, City, [State] ORDER BY [Name]) AS Rnum FROM Persons ) DELETE FROM CTE WHERE Rnum 1 In the code by saying WHERE Rnum 1, we are asking SQL Server to keep all the records with Rank 1, which are not duplicates, and delete any other record. After executing this query in SQL Server Management Studio, you will end up with no duplicates in your table. To confirm that just run a simple query against your table.

**QUESTION 5**

You work as a SQL Server 2012 database developer at ABC.com.

You are developing a batch process that accepts an input parameter and allows the result set to be joined with a table.

How would you accomplish this?

A. You should make use of a stored procedure to perform the batch process.

B. You should make use of a table-valued user-defined function.

C. You should make use of the SET XACT_ABORT ON statement.

D. You should make use of a scalar user-defined function.

E. You should make use of a schema-bound user-defined function.

Correct Answer: B

Ref: h http://msdn.microsoft.com/en-us/library/ms191007.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.pass4lead.com/allproducts

## Need Help

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