

Aggregate query quiz

1. Which one of these is not one of the aggregate functions?

Med (median)

Avg (average)

StDev (Standard deviation)

COUNT_BIG (count used for big integers)

2. Write a query that returns the average TotalDue grouped by CustomerID and the year of OrderDate from the sales.SalesOrderHeader table. How many rows are returned?

26017

```
SELECT CustomerID, AVG(TotalDue) AS AvgTotalDue,  
       YEAR(OrderDate) AS OrderYear  
FROM Sales.SalesOrderHeader  
GROUP BY CustomerID, YEAR(OrderDate);
```

3. Starting with the query written in question 2, add an ORDER BY clause so that the highest average total due is returned first. What is the CustomerID in the row with the highest average total due?

29641

```
SELECT CustomerID, AVG(TotalDue) AS AvgTotalDue,  
       YEAR(OrderDate) AS OrderYear  
FROM Sales.SalesOrderHeader  
GROUP BY CustomerID, YEAR(OrderDate)  
ORDER BY AVG(TotalDUE) DESC;
```

4. Write a query that returns the maximum TotalDue of the orders that were placed in 2011 grouped by the TerritoryID. How many rows are returned?

10

```
SELECT max(totaldue) AS MaxTotalDue, TerritoryID
FROM Sales.SalesOrderHeader
where orderdate between '1/1/2011' and '12/31/2011'
GROUP BY TerritoryID
```

5. Add an order by clause to the query written in question 4 so that the territory with the highest maximum total due is returned first. Which TerritoryID has the highest maximum totalDue in 2011?

5

```
SELECT max(totaldue) MaxTotalDue, TerritoryID
FROM Sales.SalesOrderHeader
where orderdate between '1/1/2011' and '12/31/2011'
GROUP BY TerritoryID
ORDER BY MaxTotalDue DESC;
```

6. Write a query that joins the Sales.SalesOrderHeader table to the Sales.SalesOrderDetail table. The OrderQty column of the Sales.SalesOrderDetail table records how many of the item were sold for that detail row. The query should return the total of items sold for each ProductID per year. How many rows were returned?

610

```
SELECT SUM(SOD.OrderQty) AS TotalOfProductSold,
       ProductID, YEAR(OrderDate) AS OrderYear
FROM Sales.SalesOrderHeader AS SOH
JOIN Sales.SalesOrderDetail AS SOD ON SOH.SalesOrderID = SOD.SalesOrderID
GROUP BY ProductID, YEAR(OrderDate);
```

7. When looking at the results of the query from question 6, how many of ProductID 809 were sold in 2012?

538

8. Starting with the query from Question 6, join the Production.Product table and add the Color column to the SELECT list. What else do you need to do to get the query to run correctly?

Add Color to the GROUP BY clause

Apply an aggregate function to Color

Remove Color

Add Color to the HAVING clause

```

SELECT SUM(SOD.OrderQty) AS TotalOfProductSold,
       SOD.ProductID, YEAR(OrderDate) AS OrderYear,
       Color
FROM Sales.SalesOrderHeader AS SOH
JOIN Sales.SalesOrderDetail AS SOD ON SOH.SalesOrderID = SOD.SalesOrderID
JOIN Production.Product AS Prod ON Prod.ProductID = SOD.ProductID
GROUP BY SOD.ProductID, YEAR(OrderDate),
       Color;

```

9. What do you need to add to the following query so that only customers with at least 10 orders is returned?

```

SELECT CustomerID, SUM(TotalDue) AS TotalSales,
       COUNT(*) OrderCount
FROM Sales.SalesOrderHeader
GROUP BY CustomerID;

```

```
HAVING COUNT(*) >= 10
```

```
HAVING COUNT(*) > 10
```

```
WHERE COUNT(*) >= 10
```

```
WHERE COUNT(*) > 10
```

10. Write a query from the Person.Person table that returns each FirstName and LastName along with a count of the rows. Which FirstName and LastName combination is used most in this table?

Laura Norman

