

AdventureWorks Database – 2019

Get the entity relationship diagram: [Click here](#)

1. Write a query returning the SalesOrderID, OrderDate, CustomerID and TotalDue from Sales.SalesOrderHeader table. Including the average order total over all the results.

```
SELECT SalesOrderID,  
       OrderDate,  
       CustomerID,  
       TotalDue,  
       AVG(SalesOrderID) OVER() AS avg_order  
FROM Sales.SalesOrderHeader  
ORDER BY CustomerID
```

2. Add the average TotalDue for each customer to the query in Q1.

```
SELECT SalesOrderID,  
       OrderDate,  
       CustomerID,  
       TotalDue,  
       AVG(SalesOrderID) OVER() AS avg_order,  
       AVG(TotalDue) OVER() AS avg_totaldue  
FROM Sales.SalesOrderHeader  
ORDER BY CustomerID
```

3. Write a query that assigns row numbers to the Production.Product table. Start the numbers over for each ProductSubcategoryID and make sure that the row numbers are in the order of ProductID. Display only rows where the ProductSubcategoryID is not null.

```
SELECT ProductID,  
       Name,  
       ProductSubcategoryID,  
       ROW_NUMBER() OVER (PARTITION BY ProductSubcategoryID ORDER BY ProductID) AS  
RowNumber  
FROM Production.Product  
WHERE ProductSubcategoryID IS NOT NULL;
```

4. Write a query that divides the customers into ten buckets based on the total sales for 2021.

```
SELECT CustomerID,  
       SUM(TotalDue) AS Sum_TotalDue,  
       NTILE(10) OVER(ORDER BY SUM(TotalDue) DESC) AS top_10_buckets  
FROM Sales.SalesOrderHeader  
WHERE YEAR(OrderDate) = 2011  
GROUP BY CustomerID
```

5. Write a query that returns the SalesOrderID, ProductID, and LineTotal from Sales.SalesOrderDetail. Calculate a running total of the LineTotal for each ProductID in order of SalesOrderID. Be sure to use the correct frame.

```
SELECT SalesOrderID,  
       ProductID,  
       LineTotal,  
       SUM(LineTotal) OVER(PARTITION BY ProductID ORDER BY SalesOrderID) AS  
running_total  
FROM Sales.SalesOrderDetail  
ORDER BY ProductID
```

6. Explain why you should specify the frame where it is supported instead of relying on the default.

- Relying on default is suitable for some cases which depend on our purposes (like running total - cumulative summary in Q5). The range for default frame is ROWS UNBOUNDED PRECEDING AND CURRENT ROW.
- In other purposes - when we want to specify the 2nd highest TotalDue for each productID at all rows in new columns, we have to change the default to ROWS BETWEEN.
- we have to change the default to ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING to make sure the 2nd highest totaldue for each product will be added to new column using nth_value(TotalDue, 2) function and so on...

7. Say that you want to Calculate an average of the current row and the row before and after. What would the frame look like in order to accomplish this.

I would like to choose the frame like : ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING:

- When it comes to the 1st value, it doesn't have the value before that, so the average will be counted using current row and 1 row following that.
- It also applies for the last value when it does not have the following value for the next after the last value.