

SeQuEL 10 - Aggregating Query

by Clark Anderson

We begin with a query that uses some of the Aggregating Functions and supplies a text constant, 'Grand Total' for the Name column. This query summarizes all of the selected values for Quantity. The WHERE clause is a filter for the original data.

```
SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS AvgQty, COUNT(*) AS NRec, 'Grand Total' AS BagelName
FROM BagelsBaked
WHERE Quantity > 0
```

BagelsBaked Name Grand Total Quantity Stats						
SumQty	MaxQty	MinQty	AvgQty	NRec	BagelName	
5820	360	80	135	43	Grand Total	

Next, we have modified the query to list individual Name values and added the 'GROUP BY Name' clause to create the aggregate statistics for each individual name value collection.

The ORDER BY clause sorts the GROUP BY result set collection in ascending order.

```
SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS AvgQty, COUNT(*) AS NRec, Name AS BagelName
FROM BagelsBaked
WHERE Quantity > 0
GROUP BY Name
ORDER BY SUM(Quantity)
```

BagelsBaked Name Quantity Stats						
SumQty	MaxQty	MinQty	AvgQty	NRec	BagelName	
420	170	120	140	3	Cinnamon Raisin	
660	360	100	165	4	Plain	
850	150	90	121	7	Whole Wheat	
870	140	100	124	7	Everything	
900	240	100	150	6	Sesame Seed	
1030	240	90	128	8	Egg	
1090	240	80	136	8	Poppy Seed	

We can unite these first two queries to make a good summary report.

```
SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS AvgQty, COUNT(*) AS NRec, Name AS BagelName
FROM BagelsBaked
WHERE Quantity > 0
GROUP BY Name
ORDER BY SUM(Quantity)
UNION
SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS AvgQty, COUNT(*) AS NRec, 'Grand Total' AS BagelName
FROM BagelsBaked
WHERE Quantity > 0
```

```

-----
|           BagelsBaked Name Quantity Stats Report           |
-----
|SumQty |MaxQty |MinQty |AvgQty | NRec  | BagelName |
-----
|  420  |  170  |  120  |  140  |   3   |Cinnamon Raisin|
|  660  |  360  |  100  |  165  |   4   | Plain         |
|  850  |  150  |   90  |  121  |   7   | Whole Wheat   |
|  870  |  140  |  100  |  124  |   7   | Everything     |
|  900  |  240  |  100  |  150  |   6   | Sesame Seed   |
| 1030  |  240  |   90  |  128  |   8   | Egg           |
| 1090  |  240  |   80  |  136  |   8   | Poppy Seed    |
-----
| 5820  |  360  |   80  |  135  |  43   | Grand Total   |
-----

```

Now, we modify the WHERE clause to be more specific about which records to include in the query results.

```

SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS
AvgQty, COUNT(*) AS NRec , Name AS BagelName
FROM BagelsBaked
WHERE Quantity > 0 AND Name LIKE '*SEED*'
GROUP BY Name

```

```

-----
|           BagelsBaked Name Seed Quantity Stats           |
-----
|SumQty |MaxQty |MinQty |AvgQty | NRec  | BagelName |
-----
| 1090  |  240  |   80  |  136  |   8   | Poppy Seed   |
|  900  |  240  |  100  |  150  |   6   | Sesame Seed  |
-----

```

Here we can add 'NOT' to the WHERE clause and reverse the choice of included records.

```

SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS
AvgQty, COUNT(*) AS NRec , Name AS BagelName
FROM BagelsBaked
WHERE Quantity > 0 AND Name NOT LIKE '*SEED*'
GROUP BY Name

```

```

-----
|           BagelsBaked Name NOT Seed Quantity Stats           |
-----
|SumQty |MaxQty |MinQty |AvgQty | NRec  | BagelName |
-----
|  420  |  170  |  120  |  140  |   3   |Cinnamon Raisin|
| 1030  |  240  |   90  |  128  |   8   | Egg           |
|  870  |  140  |  100  |  124  |   7   | Everything     |
|  660  |  360  |  100  |  165  |   4   | Plain         |
|  850  |  150  |   90  |  121  |   7   | Whole Wheat   |
-----

```

The HAVING clause is used to filter the results of the GROUP BY clause collection. This is the HAVING clause's unique purpose.

```
SELECT SUM(Quantity) AS SumQty, MAX(Quantity) AS MaxQty, MIN(Quantity) AS MinQty, INT(AVG(Quantity)) AS
AvgQty, COUNT(*) AS NRec , Name AS BagelName
FROM [D:\PROfssn\ACCESSUG\Articles\10SQLAggr10\ASQLUni6.mdb].BagelsBaked
WHERE Quantity > 0 AND Name NOT LIKE '*SEED*'
GROUP BY Name
HAVING COUNT(*) > 3 AND MIN(Quantity) > 90
```

```
-----
| BagelsBaked NOT Seed Stats NRec gt 3 MinQty gt 90      |
-----
|SumQty |MaxQty |MinQty |AvgQty | NRec  | BagelName |
-----
|  870  |  140  |  100  |  124  |    7  | Everything |
|  660  |  360  |  100  |  165  |    4  | Plain      |
-----
```

One of my initial goals was to assemble a reference for future use. I surprised myself in how much new knowledge and better understanding I gained in the process.