Group Functions

Introduction

SQL has the ability to group records and display a summary of a column(s). This is accomplished with the use of Group functions, also called multiple-row or aggregate functions. For instance, finding the highest or lowest value in a column, totaling the values in a column, averaging the values in a column, or counting the values stored in a column. A single value for the entire table can be returned, or rows can be grouped together and a value displayed for each group.

Refer to your textbook or the Oracle SQL Reference Manual pdf for excellent examples of the following functions.

Group Functions

Oracle supports the following Group functions: SUM, AVG, COUNT, MAX, MIN, STDDEV, and VARIANCE. SUM, AVG, STDDEV and VARIANCE use numeric columns only, whereas COUNT, MAX and MIN can use numeric or character columns. Note that these functions ignore NULL values when calculating their answer. The optional keywords DISTINCT and ALL can be used to include or omit duplicate values in a column. ALL is the default.

The SUM function returns the accumulated value of a column or mathematical expression. To total the quantity column in the OrderItems table we would code: SELECT SUM(Quantity) “Total Quantity on Order” FROM OrderItems;. You can also use a mathematical expression such as SUM(Quantity * 2).

The AVG function returns the average value of a column or mathematical expression. To find the average cost of books in inventory we would code: SELECT AVG(Cost) “Average Book Cost” FROM Books;. Special consideration should be given to numeric columns that can be NULL.

The COUNT function returns a value that represents how many rows were retrieved. This can be used to tell us how many rows are stored in the table by using the * operator: SELECT COUNT(*) FROM Books; would give a result of 14. Or, COUNT can be used to tell us how many non-NULL values are in a specific column. If we wanted to know how many customers have been referred: SELECT COUNT(Referred) FROM Customers; would give a result of 5.

The MAX and MIN functions return the largest and smallest values, respectively, in a column. If used on a non-numeric column, the value with the highest or lowest “sort” order will be returned. If we wanted to know what the largest order quantity is, we would code: SELECT MAX(Quantity) FROM OrderItems;. On the other hand, if we needed to determine the first publisher name alphabetically, we would code: SELECT MIN(Name) FROM Publisher;.

More than one group function can be used on the SELECT, and group functions can be used in a mathematical expression or even nested. For example, if we wanted to find the range between the largest order quantity and smallest order quantity, we could code:

SELECT MAX(Quantity), MIN(Quantity), MAX(Quantity) - MIN(Quantity) “Quantity Range” FROM OrderItems;.

Grouping Data

When a group function is coded on a SELECT, a single value (row) is returned. It is sometimes necessary to show a value for each “group” of records. For instance, say we wanted to display each state and a count of the number of customers in that state. By viewing the data in the Customers table, we can see that Florida has 4 customers, California has 3 customers, and so forth. To display this query by state, we would need to group the data by state and show a count:

```
SELECT State, COUNT(State)
FROM Customers
GROUP BY State;
```

The HAVING clause can filter which groups get displayed. Back to the example above, if we only wanted those states that have a count of customers greater than 2, we would code:

```
SELECT State, COUNT(State)
FROM Customers
GROUP BY State
HAVING COUNT(State) > 2;
```

If necessary, we can filters records before the grouping process by applying the WHERE clause. Using the same example, let’s limit our query to only New York and Texas:

```
SELECT State, COUNT(State)
FROM Customers
WHERE State IN (‘NY’, ‘TX’) GROUP
BY State
HAVING COUNT(State) > 2;
```

If we wanted to sort the data, say by largest count to smallest, then we could add an ORDER BY:

```
SELECT State, COUNT(State)
FROM Customers
WHERE State IN (‘NY’, ‘TX’)
GROUP BY State
HAVING COUNT(State) > 2
ORDER BY 2 DESC;
```

Keep in mind that if a group function is used, the only other column(s) allowed on the SELECT is another group function OR a column that is being grouped on (referenced in a GROUP BY). In the examples above, we would receive an error if we did not include the GROUP BY. Also, more than two columns can be coded on the GROUP BY, with the major group field first, followed by the minor group field.

Advanced grouping can be performed by using GROUPING SETS. Refer to the SQL Reference manual for further discussion on this topic.