



# The Ultimate Oracle SQL Course

## Group Operations

### Section Recap

We started this section by looking at AGGREGATE functions, which are functions that operate on a group of rows, as opposed to single-row functions like NVL or COALESCE.

These functions return only one row for the whole group on which they were applied.

The more commonly used aggregate functions are MIN and MAX, which get the minimum and maximum values from the group, and SUM, COUNT and AVG, which sum, count or calculate the average value of the group to which they were applied.

Most aggregate functions, which includes the ones we covered here, ignore nulls, so if you apply the COUNT function to a column that can contain nulls, the function will count only the rows in which that column is not null.

If you want to count all of the rows, regardless of the existence of nulls, you have to use COUNT(\*)

Many aggregate functions, which includes the ones we covered here as well, allow you to add the DISTINCT keyword before the parameter expression, and that tells Oracle to only consider distinct values of the argument expression in their calculation.

In this section, you also learned that the GROUP BY clause allows you to separate the rows from the resultset into different groups, and most of the times the GROUP BY clause is used in conjunction with aggregate functions, which makes Oracle apply the function to each group, and then return one row per group.

A very important rule about grouped queries is that every expression that appears in the SELECT list that is not an aggregate function or an expression including an aggregate function, or a literal/constant, MUST appear in the group BY clause, if you ignore this rule you will get an error.

It would also be good to remember that you can nest aggregate functions.

And finally, you learned that there is a clause that allows you to filter the results of queries that include the GROUP BY clause, and it is the HAVING clause.

The role this clause plays is similar to that of the WHERE clause, but the HAVING clause is applied to the results of grouped queries.

You can have both, though. In that case, the WHERE clause is applied to the individual rows BEFORE grouping them, and the HAVING clause is applied after the groups are created by the GROUP BY clause and the aggregate functions have been calculated, so, you can filter based on the results of an aggregate function in the HAVING clause.

Here is an example that applies the concepts covered in this section:

```
SELECT department_id,  
        MIN(salary),  
        MAX(salary),  
        AVG(bonus)  
FROM employee  
WHERE bonus IS NOT NULL  
GROUP BY department_id  
HAVING MIN(salary) < 2000  
        OR MAX(salary) > 4000  
ORDER BY MIN(salary) DESC;
```

Congratulations on finishing another section of the course!

I will see you in the next one.