W04 Paper: Case Study JOINS in SQL queries

You have had a profitable third week at your new company. You built on your skill at filtering data into and out of a query's result set and learned how to join multiple tables this week.

Your boss gave you a new task. You should qualify what you learned about redundant data in the context of equi join (equality comparison) statements with tables and subqueries as tables. You also should qualify the importance of table aliases in self joins.

You should return and report with a 3–5 paragraph report that clearly explains what you learned while mastering command line semantics. This paper should qualify what you learned by experimenting with the technology.

Report

This week I learned about equality comparison in joins. When performing a join, equality comparison (=) is critical for matching common values between tables, allowing only rows that meet the equality condition to be included in the results.

In addition, I learned about the different types of joins in SQL, following the ANSI SQL 91 standard. Among them, I studied the CROSS JOIN, which combines all the rows of two tables in a Cartesian product, the INNER JOIN, which returns only the rows that have matches in both tables, and the LEFT JOIN and RIGHT JOIN, which include all the rows of the left or right table, even if they have no matches in the other table. In addition, I reviewed the SELF JOIN, which allows you to join a table with itself to compare rows from the same table, something very useful in data analysis.

One very important thing I learned was to use table aliases, especially when performing a self-join. Since in this type of join, a table is compared to itself, aliases are essential to distinguish between the different instances of the table within the query. Without aliases, it would be impossible to correctly refer to the columns of each instance of the table in the same query, which would cause errors. Additionally proper use of aliases facilitates the readability and accuracy of SQL queries when working with multiple complex joins.