W03 Paper Case Study

You have had a profitable second week at your new company. You now know how to use Structured Query Language (SQL) mechanics to write queries.

Your boss gave you a new task. This week, you should learn how to explain the various filtering techniques used in the WHERE clause. Your boss would like you to be able to compare and contrast the use of equality comparisons, inequality comparisons, range comparisons, and lookup operators like the IN operator.

You should return and report with a 3–5 paragraph report that clearly explains what you learned about the various comparison operators in the WHERE clause. This paper should qualify what you learned by experimenting with the technology.

Report

This week, I learned about using filtering techniques in SQL through the WHERE clause. Among the most useful operators are equality comparison (=), inequality (!=), range comparison (>, <, <=, >=), and lookup operators such as IN, BETWEEN, LIKE, AND, OR, as well as handling null values with IS NULL and IS NOT NULL.

The equality operator (=) is the most straightforward for searching for an exact match. The inequality operator (!=) is used when we want to exclude a specific value from the query. The range comparison operators (>, <, <=, >=) allow searches within set ranges, such as numbers, dates, or strings. This helps list products with prices above or below a value or find transactions before or after a particular date. Additionally, the AND operator ensures that both conditions are met, while the OR operator requires at least one of the conditions to be met.

The IN operator simplifies the search for values within a predefined list, which is useful for filtering by a limited set of categories. The BETWEEN operator defines a range between two values, such as prices or dates, and is efficient when handling large volumes of data. The LIKE operator is ideal for text searches. Unlike numeric operators, LIKE allows for searching patterns within text columns, such as names that begin, contain, or end with certain letters using wildcards like '\_' to match a single character, and '%' to match any number of characters. This is complemented using the REGEX operator, which offers more complex searches based on specific patterns, such as '^' to mark the beginning of a string, '$' for the end, '.' representing any character, and '[]' to define a set of characters.

In addition, handling null values with IS NULL and IS NOT NULL is helpful in large databases that may contain incomplete records. These operators allow you to identify or exclude data that has no values assigned to it.