**GROUP BY** in SQL is a statement used to organize data into groups based on one or more columns. It is commonly used with aggregate functions (e.g., COUNT, SUM, AVG, MAX, MIN) to perform calculations on each group of data.

**Key Features of GROUP BY:**

1. **Grouping Data**: Rows with the same values in the specified columns are grouped together.
2. **Aggregate Functions**: Applied to each group to calculate summary statistics (e.g., total sales per region).
3. **Used with SELECT**: Often combined with aggregate functions to retrieve meaningful insights from grouped data.

**Syntax:**

SELECT column1, column2, aggregate\_function(column3)

FROM table\_name

GROUP BY column1, column2;

* column1, column2: Columns by which the data is grouped.
* aggregate\_function: Any function like SUM, COUNT, AVG, MAX, MIN.

**Example Use Cases:**

**1. Counting Records in Groups:**

SELECT department, COUNT(employee\_id) AS employee\_count

FROM employees

GROUP BY department;

* Groups employees by department.
* Counts the number of employees in each department.

**2. Summing Values in Groups:**

SELECT customer\_id, SUM(order\_total) AS total\_spent

FROM orders

GROUP BY customer\_id;

* Groups orders by customer ID.
* Calculates the total amount spent by each customer.

**3. Combining with WHERE:**

SELECT category, AVG(price) AS average\_price

FROM products

WHERE price > 10

GROUP BY category;

* Filters products with a price greater than 10.
* Groups by category and calculates the average price for each.

**4. Using HAVING with GROUP BY:**

HAVING filters groups after they are created.

SELECT department, AVG(salary) AS average\_salary

FROM employees

GROUP BY department

HAVING AVG(salary) > 50000;

* Groups employees by department.
* Displays only those departments where the average salary exceeds 50,000.

**Best Practices:**

1. **Columns in SELECT Must Be in GROUP BY or Aggregated**:
   * Columns not part of an aggregate function must appear in the GROUP BY clause.
   * Example (Invalid):
   * SELECT department, salary
   * FROM employees
   * GROUP BY department;
     + salary must be aggregated or removed.
2. **Order of Execution**:
   * SQL processes GROUP BY after WHERE and before ORDER BY.
3. **Nulls in Grouping**:
   * Rows with NULL in the grouping column are treated as a single group.

**Advanced Features:**

1. **Grouping by Expressions**:
2. SELECT YEAR(order\_date), SUM(order\_total) AS yearly\_sales
3. FROM orders
4. GROUP BY YEAR(order\_date);
   * Groups by a calculated value (YEAR(order\_date)).
5. **Rollups and CUBE (if supported)**:
   * Perform hierarchical or multidimensional grouping.

SELECT department, job\_title, COUNT(\*)

FROM employees

GROUP BY ROLLUP(department, job\_title);

* Provides subtotals and grand totals for groups.

The GROUP BY statement is an essential tool for analyzing and summarizing data in SQL.