**GROUP BY**

The **GROUP BY** statement in SQL is used to organize rows in a dataset into groups based on one or more columns. It's commonly paired with aggregate functions to perform calculations on each group.

To group rows with the same values in specified columns.

To calculate summaries (like totals, averages, counts) for each group.

SELECT column1, column2, aggregate\_function(column3)

FROM table\_name

GROUP BY column1, column2;

EXAMPLE:

Consider a table Sales

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Product** |  |  | | --- | --- | --- |  |  |  | | --- | --- | |  |  | | | **Region** | | --- | | | **Amount** | | --- | |
| A | EAST | |  | | --- | | 100 | |
| B | WEST | 200 |
| A | EAST | 150 |
| B  A | WEST  WEST | 250  300 |

SELECT Product, Region, SUM(Amount) AS RegionalSales

FROM Sales

GROUP BY Product, Region;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| | **Product** | | --- | | | **Region** | | --- | | | **RegionalSales** | | --- | |
| A | EAST | 250 |
| A | WEST | 300 |
| B | WEST | 450 |

SELECT Product, COUNT(\*) AS NumberOfSales

FROM Sales

GROUP BY Product;

**HAVING**

The **HAVING** clause in SQL is used to filter groups of data created by the GROUP BY clause based on aggregate function conditions. It is similar to the WHERE clause but is applied after data has been grouped and aggregated.

 The WHERE clause filters rows **before** grouping.

 The HAVING clause filters groups **after** aggregation.

SELECT column1, aggregate\_function(column2)

FROM table\_name

GROUP BY column1

HAVING condition;

EXAMPLE:

TABLE –SALES

|  |  |  |
| --- | --- | --- |
| PRODUCT | REGION | AMOUNT |
| A | EAST | 100 |
| B | WEST | 200 |
| A | EAST | 150 |
| B | WEST | 250 |
| A | WEST | 300 |

SELECT Product, SUM(Amount) AS TotalSales

FROM Sales

GROUP BY Product

HAVING SUM(Amount) > 500;

|  |  |
| --- | --- |
| PRODUCT | TOTALSALES |
| A | 550 |

## EXISTS

## The EXISTS operator in SQL is used to test for the existence of rows in a subquery. It returns TRUE if the subquery returns at least one row, and FALSE otherwise.

 To check if a condition is met by any row in a related table or dataset.

##  Often used in conjunction with WHERE or as part of conditional logic.

## SELECT column1, column2, ...

## FROM table\_name

## WHERE EXISTS (

## SELECT 1

## FROM another\_table

## WHERE condition);

## EXAMPLE:

## CUSTOMER TABLE

|  |  |
| --- | --- |
| CustomerID | Name |
| 1 | ALICE |
| 2 | BOB |
| 3 | CHARLIE |

## ORDER TABLE

|  |  |  |
| --- | --- | --- |
| OrderID | CustomerID | Amount |
| 101 | 1 | 500 |
| 102 | 2 | 300 |

## Query:Find customers who have placed at least one order.

## SELECT Name

## FROM Customers

## WHERE EXISTS (

## SELECT 1

## FROM Orders

## WHERE Orders.CustomerID = Customers.CustomerID

## );

## OUTPUT:

## ALICE

## BOB