Group by and having clouse complex example

Certainly! Here's a more complex example of a SQL query that makes use of both the **GROUP BY** and **HAVING** clauses:

Suppose you have a table named "sales" with columns "region", "salesperson", "product", "month", and "sales". The table contains information about sales made by different salespersons in different regions for different products in different months. Here is a sample of the data in the "sales" table:

create table employee\_sales(

emplyee\_id int primary key identity,

employee\_name varchar(25),

department varchar(25),

region varchar(25),

[month] varchar(25),

sales int

)

INSERT INTO employee\_sales (employee\_name, department, region, month, sales)

VALUES

('Alice', 'Sales', 'North', 'Jan-20', 1000),

('Bob', 'Sales', 'South', 'Feb-20', 3000),

('Charlie', 'Sales', 'South', 'March-20', 1500),

('Dave', 'Sales', 'North', 'App-20', 1000),

('Eve', 'Sales', 'North', 'May-20', 3500),

('Frank', 'Support', 'North', 'App-20', 1000),

('Grace', 'Support', 'North', 'May-20', 1000);

--- get Those Employee jin ke AVG(sales) >= 1500 AND COUNT(DISTINCT emplyee\_id) >= 2 ORDER BY avg\_sale DESC;

SELECT

department,

region,

AVG(sales) AS avg\_sale,

COUNT(DISTINCT emplyee\_id) AS employee\_count

FROM employee\_sales

GROUP BY department, region

HAVING AVG(sales) >= 1500 AND COUNT(DISTINCT emplyee\_id) >= 2

ORDER BY avg\_sale DESC;

1. Selects the **region**, **product**, **total\_sales** (sum of the **sales** column using the **SUM** function), and **avg\_sale** (average of the **sales** column using the **AVG** function).
2. Groups the data by **region** and **product** to create summarized rows.
3. Filters the grouped rows using the **HAVING** clause to only include rows where the **total\_sales** is greater than or equal to 2500.
4. Orders the result set by **total\_sales** in descending order using the **ORDER BY** clause.