SQL HAVING Clause

The `HAVING` clause is used to filter groups of data created by the `GROUP BY` clause based on aggregate functions. Unlike the `WHERE` clause, which filters rows before grouping, the `HAVING` clause filters groups after the aggregation has been performed.

Key Features of HAVING

1. **Purpose:** Filter aggregated results (e.g., totals, averages, counts).
2. **Works Only with Aggregates:** Operates on aggregated values (e.g., `SUM`, `AVG`, `COUNT`).
3. **Used with GROUP BY:** Typically used after `GROUP BY` to filter groups.
4. **Syntax:**

SQL

```sql
SELECT column1, AGGREGATE_FUNCTION(column2)
FROM table_name
GROUP BY column1
HAVING condition;

```

Examples

1. Filter Groups Based on Aggregated Values

Example: Find departments with a total salary greater than 100,000.

SQL

```sql
SELECT department_id, SUM(salary) AS total_salary
FROM employees
GROUP BY department_id
HAVING SUM(salary) > 100000;

```

Result: Only departments where the total salary exceeds 100,000 are shown.

---

2. Filter Groups with `COUNT`

Example: Find job titles with more than 5 employees.

SQL

```sql
SELECT job_title, COUNT(employee_id) AS total_employees
FROM employees
GROUP BY job_title
HAVING COUNT(employee_id) > 5;

```

Result: Only job titles with more than 5 employees are shown.

---

3. Filter Groups with `AVG`

Example: Find departments with an average salary less than 50,000.

SQL

```sql
SELECT department_id, AVG(salary) AS average_salary
FROM employees
GROUP BY department_id
HAVING AVG(salary) < 50000;

```

Result: Only departments with an average salary below 50,000 are shown.

Using HAVING with Multiple Conditions

You can use logical operators (`AND` , `OR`) to filter groups based on multiple conditions.

Example: Find departments where the total salary is above 100,000 and the average salary is above 40,000.

SQL

```sql
SELECT department_id,
       SUM(salary) AS total_salary,
       AVG(salary) AS average_salary
FROM employees
GROUP BY department_id
HAVING SUM(salary) > 100000 AND AVG(salary) > 40000;

```

Difference Between WHERE and HAVING

	Feature	WHERE Clause	HAVING Clause
1	**Purpose**	Filters rows before grouping	Filters groups after aggregation
2	**Used With**	Columns, constants	Aggregate functions, columns
3	**Example**	`WHERE salary > 50000`	`HAVING AVG(salary) > 50000`

+ Neu

Using HAVING Without GROUP BY

Although rare, you can use `HAVING` without `GROUP BY` to filter results based on aggregate functions applied to the entire result set.

Example: Check if the total salary across all employees is greater than 500,000.

SQL

```sql
SELECT SUM(salary) AS total_salary
FROM employees
HAVING SUM(salary) > 500000;

```

Combining WHERE and HAVING

The `WHERE` clause filters rows before grouping, while `HAVING` filters groups after aggregation.

Example: Find departments where the total salary is above 100,000, but only consider employees earning more than 30,000.

SQL

```sql
SELECT department_id, SUM(salary) AS total_salary
FROM employees
WHERE salary > 30000
GROUP BY department_id
HAVING SUM(salary) > 100000;

```

Order of Execution in SQL

1. **FROM:** Tables and joins are processed first.
2. **WHERE:** Filters rows before aggregation.
3. **GROUP BY:** Groups rows.
4. **HAVING:** Filters aggregated results.
5. **SELECT:** Selects columns or aggregates to display.
6. **ORDER BY:** Sorts the final result.

Practice Query

Imagine an `orders` table with columns: `order_id`, `customer_id`, `order_date`, and `order_total`.

Task: Find customers who placed more than 5 orders and whose total order value exceeds 10,000.

SQL

```sql
SELECT customer_id, COUNT(order_id) AS total_orders, SUM(order_total) AS total_value
FROM orders
GROUP BY customer_id
HAVING COUNT(order_id) > 5 AND SUM(order_total) > 10000;

```

Would you like to explore more complex examples or practice queries?