
📚 SQL Practice Course: Using the `HAVING` Clause

📖 Module 1: Understanding the `HAVING` Clause

✅ What is `HAVING` in SQL?

- The `HAVING` clause is used to **filter groups** after the `GROUP BY` clause.
- Difference between **WHERE vs HAVING**
- `HAVING` works with **aggregate functions** like `COUNT()`, `SUM()`, `AVG()`, etc.

✅ Syntax of `HAVING`

```sql
SELECT column, AGG_FUNCTION(column)
FROM table
GROUP BY column
HAVING condition;
```

✅ Basic Example

Find departments where the **average salary is greater than 5000**:

```sql
SELECT Department, AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Department
HAVING AVG(Salary) > 5000;
```

📖 Module 2: Multi-Table Setup for Practice

We'll use **three tables**:

🔹 Employees Table

```sql
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DepartmentID INT,
    Salary DECIMAL(10,2)
);
```

🔹 Departments Table

```sql
CREATE TABLE Departments (
    DepartmentID INT PRIMARY KEY,
    DepartmentName VARCHAR(50)
);
```

◇ Projects Table

```sql
CREATE TABLE Projects (
    ProjectID INT PRIMARY KEY,
    ProjectName VARCHAR(50),
    DepartmentID INT
);
```

◇ Basic `HAVING` Clause Practice

1️⃣ Get all departments where **more than 5 employees** work.

```sql
SELECT DepartmentID, COUNT(EmployeeID) AS EmployeeCount
FROM Employees
GROUP BY DepartmentID
HAVING COUNT(EmployeeID) > 5;
```

2️⃣ Find departments where the **average salary is above 6000**.
3️⃣ Show departments where **total salary exceeds 50,000**.
4️⃣ Find employees who appear **more than once** in the table.
5️⃣ List departments where **minimum salary is below 3000**.

◇ `HAVING` with Multiple Joins

6️⃣ Find departments where **more than 3 projects are assigned**.

```sql
SELECT d.DepartmentName, COUNT(p.ProjectID) AS ProjectCount
FROM Departments d
JOIN Projects p ON d.DepartmentID = p.DepartmentID
GROUP BY d.DepartmentName
HAVING COUNT(p.ProjectID) > 3;
```

7️⃣ Show departments where the **total salary is greater than 100,000**.
8️⃣ List projects where the **average employee salary is more than 7000**.

9️⃣ Find departments with **at least one project and more than 10 employees**.
🔟 Show projects where the **sum of employee salaries exceeds 50,000**.

◇ Advanced Queries with `HAVING` and Aggregations

1️⃣1️⃣ Show departments with **more than 2 employees earning above 8000**.
1️⃣2️⃣ Find projects where the **highest-paid employee earns over 9000**.
1️⃣3️⃣ Get projects where the **average salary of employees is below 4000**.
1️⃣4️⃣ Find departments where **more than half the employees earn above 5000**.
1️⃣5️⃣ Show departments where **total number of employees is between 5 and 15**.

◇ `HAVING` with Nested Queries

1️⃣6️⃣ Find projects where **total employee salary is greater than the average salary of all employees**.
1️⃣7️⃣ Show departments where the **number of employees is more than the average number of employees per department**.
1️⃣8️⃣ List departments where **total salary is above the company's average salary**.
1️⃣9️⃣ Find projects where **employees work on more than one project**.
2️⃣0️⃣ Show departments where the **highest salary is more than twice the lowest salary**.

◇ Complex Queries with Multiple Conditions

2️⃣1️⃣ Find projects where **more than 3 employees earn above 6000 and total salary is above 30,000**.
2️⃣2️⃣ Show departments where **the sum of salaries is greater than the average salary multiplied by the number of employees**.
2️⃣3️⃣ Find employees working on projects **that have an average salary below 5000**.
2️⃣4️⃣ Show projects where **the number of employees is between 5 and 10 and total salary exceeds 40,000**.
2️⃣5️⃣ List departments where **no employee earns below 4000**.

◇ `HAVING` with Date Filters

2️⃣6️⃣ Show projects started in the last 2 years where **total salary exceeds 100,000**.
2️⃣7️⃣ Find departments where **employees hired in the last 5 years have an average salary above 7000**.
2️⃣8️⃣ List employees who joined before 2020 and earn above the average salary.
2️⃣9️⃣ Show departments where the **oldest employee has worked for more than 10 years**.
3️⃣0️⃣ Find projects with employees who joined **after 2018 and have an average salary above 6000**.

🎯 Learning Outcome

✔ **Understand how `HAVING` filters grouped results**
✔ **Use `HAVING` with multiple joins and aggregate functions**
✔ **Write advanced queries for real-world scenarios**
✔ **Optimize queries using indexes and execution plans**

Would you like me to create **solutions** for all 30 questions? 🚀