**GROUP BY and HAVING in SQL Server**

**1. GROUP BY Clause**

The GROUP BY clause in SQL Server is used to group rows with the same values in specified columns and perform aggregate functions like COUNT(), SUM(), AVG(), MAX(), MIN(), etc.

**Syntax:**

SELECT column\_name, AGGREGATE\_FUNCTION(column\_name)

FROM table\_name

GROUP BY column\_name;

**Example:**

SELECT Department, COUNT(EmployeeID) AS TotalEmployees

FROM Employees

GROUP BY Department;

This query groups employees by department and counts the number of employees in each department.

**2. HAVING Clause**

The HAVING clause is used to filter the results after grouping, similar to the WHERE clause but specifically for aggregate functions.

**Syntax:**

SELECT column\_name, AGGREGATE\_FUNCTION(column\_name)

FROM table\_name

GROUP BY column\_name

HAVING condition;

**Example:**

SELECT Department, COUNT(EmployeeID) AS TotalEmployees

FROM Employees

GROUP BY Department

HAVING COUNT(EmployeeID) > 5;

This query retrieves departments where the number of employees is greater than 5.

**Difference Between WHERE and HAVING**

| **Feature** | **WHERE** | **HAVING** |
| --- | --- | --- |
| Used For | Filtering individual rows before grouping | Filtering groups after aggregation |
| Can Use Aggregate Functions | ❌ No | ✅ Yes |
| Works Before/After GROUP BY | Before GROUP BY | After GROUP BY |

**Example with both WHERE and HAVING:**

SELECT Department, COUNT(EmployeeID) AS TotalEmployees

FROM Employees

WHERE Salary > 50000 -- Filters employees before grouping

GROUP BY Department

HAVING COUNT(EmployeeID) > 5; -- Filters groups after aggregation

This filters out employees with a salary below 50,000, then groups by department, and finally selects only the departments with more than 5 employees.

**TASK**

**Question 1**

**Table:** Employees

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| EmployeeID | INT | Unique ID for each employee |
| Name | VARCHAR(100) | Employee name |
| Department | VARCHAR(50) | Department name |
| Salary | DECIMAL(10,2) | Employee salary |

**Question:**  
Write an SQL query to display each department and the total number of employees where the salary is greater than 50,000. Show only those departments that have more than 5 employees.

**Question 2**

**Table:** Students

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| StudentID | INT | Unique ID for each student |
| StudentName | VARCHAR(100) | Name of the student |
| CourseName | VARCHAR(50) | Course enrolled |
| Marks | INT | Marks obtained |

**Question:**  
Write an SQL query to display each course and the average marks of students who scored 40 or more marks. Show only those courses whose average marks are greater than 75.

**Question 3**

**Table:** Sales

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| SaleID | INT | Unique sale ID |
| Category | VARCHAR(50) | Product category |
| SalesAmount | DECIMAL(10,2) | Amount of sale |
| OrderDate | DATE | Date of the order |

**Question:**  
Write an SQL query to display each product category and the total sales amount for sales made on or after January 1, 2025. Show only the categories with total sales exceeding 100,000.

**Question 4**

**Table:** Customers

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| CustomerID | INT | Unique customer ID |
| CustomerName | VARCHAR(100) | Name of the customer |
| City | VARCHAR(50) | City of the customer |
| Country | VARCHAR(50) | Country of the customer |

**Question:**  
Write an SQL query to display each city and the total number of customers living in it, considering only customers from the USA. Show only cities with more than 10 customers.

**Question 5**

**Table:** Projects

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| ProjectID | INT | Unique project ID |
| EmployeeID | INT | ID of employee working on project |
| HoursWorked | INT | Total hours worked |
| Status | VARCHAR(20) | Status of the project (e.g., Completed, Ongoing) |

**Question:**  
Write an SQL query to display each project and the total hours worked, considering only completed projects. Show only projects that have more than 500 total hours worked.