SQL Queries

SQL (Structured Query Language) allows you to interact with databases by writing queries to retrieve and manipulate data. In this section, we will discuss complex SELECT statements, SQL aggregation functions, and how to use GROUP BY and HAVING clauses.

1. Complex SELECT Statements

A SELECT statement retrieves data from one or more tables in a database. You can make SELECT queries more complex by adding multiple conditions and sorting the results.

Multiple Conditions (Using WHERE Clause)

You can filter the data based on more than one condition. This is done using logical operators such as `AND`, `OR`, and `NOT`.

- AND: All conditions must be true.
- OR: At least one condition must be true.
- NOT: Reverses the condition.

Example:

```
SELECT name, age, salary
FROM employees
WHERE age > 30 AND salary > 50000;
```

This query retrieves the names, ages, and salaries of employees who are older than 30 and have a salary greater than 50,000.

Sorting Data (Using ORDER BY Clause)

You can sort the result set by one or more columns using the `ORDER BY` clause.

- ASC: Sorts data in ascending order (default).
- DESC: Sorts data in descending order.

Example:

```
SELECT name, age, salary
FROM employees
WHERE age > 30
ORDER BY salary DESC;
```

This query retrieves the names, ages, and salaries of employees older than 30, sorted by salary in descending order.

2. SQL Functions for Data Aggregation

SQL provides several functions to perform calculations on data. These functions are commonly used in combination with `GROUP BY` to aggregate data in a meaningful way.

COUNT() Function
- COUNT counts the number of rows in a table or group.

Example:

```
SELECT department, COUNT(*) AS num_employees
FROM employees
GROUP BY department;
```

This query counts the number of employees in each department.

SUM() Function
- SUM adds up the values of a specified column.

Example:

```
SELECT department, SUM(salary) AS total_salary
FROM employees
GROUP BY department;
```

This query calculates the total salary for each department.

AVG() Function
- AVG calculates the average value of a specified column.

Example:

```
SELECT department, AVG(salary) AS average_salary
FROM employees
GROUP BY department;
```

This query calculates the average salary for each department.

MIN() and MAX() Functions
- MIN returns the smallest value in a column.
- MAX returns the largest value in a column.

Example:

```
SELECT department, MIN(salary) AS lowest_salary, MAX(salary) AS
highest_salary
FROM employees
GROUP BY department;
```

This query finds the lowest and highest salaries in each department.

3. GROUP BY Clause

The GROUP BY clause is used to group rows that have the same values in specified columns into summary rows. It is often used with aggregate functions like `COUNT()`, `SUM()`, `AVG()`, etc.

Example:

```
SELECT department, COUNT(*) AS num_employees, AVG(salary) AS avg_salary
FROM employees
GROUP BY department;
```

This query groups the employees by department and calculates the number of employees and the average salary for each department.

Using GROUP BY with Multiple Columns

You can group data by more than one column. This can be useful when you want to group by combinations of values.

Example:

```
SELECT department, job_title, AVG(salary) AS avg_salary
FROM employees
GROUP BY department, job_title;
```

This query groups employees by both department and job title, and calculates the average salary for each combination.

4. HAVING Clause

The HAVING clause is used to filter groups created by the GROUP BY clause. You can use it to filter the results based on aggregated data (like `COUNT()`, `SUM()`, `AVG()`) which cannot be filtered using the WHERE clause.

Example of HAVING with COUNT()

```
SELECT department, COUNT(*) AS num_employees
FROM employees
GROUP BY department
HAVING COUNT(*) > 10;
```

This query groups employees by department and only includes departments with more than 10 employees.

Example of HAVING with SUM()

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

This query calculates the total salary for each department and only includes departments where the total salary is greater than 100,000.

In Summary:

- Complex SELECT statements can retrieve data based on multiple conditions and sort the results.

- SQL functions like `COUNT()`, `SUM()`, and `AVG()` are used to perform calculations and aggregate data.
- The GROUP BY clause groups rows by column values and is often used with aggregate functions to summarize data.
- The HAVING clause filters groups based on aggregated data, while WHERE filters individual rows before grouping.