🔢 Operators in MySql

1. **Arithmetic Operators**

Used to perform mathematical operations.

```sql
SELECT ename, salary, salary + 1000 AS new_salary
FROM emps;
```

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus

---

2. **Comparison Operators**

Used to compare values.

```sql
SELECT ename, salary
FROM emps
WHERE salary > 5000;
```

Operator	Description
=	Equal to
!= / <>	Not equal to
>	Greater than
<	Less than
>=	Greater or equal
<=	Less or equal

---

3. **Logical Operators**

Used to combine multiple conditions.

```
SELECT ename, salary, deptno
FROM emps
WHERE salary > 5000 AND deptno = 20;
```

Operator	Description
AND	All conditions must be true
OR	At least one condition true
NOT	Reverses the condition result

🧮 5 Commonly Used MySQL Functions with Examples

1. **CONCAT()** - Combines multiple strings

```
SELECT empno, CONCAT(ename, ' - ', job_title) AS employee_details
FROM emps;
```

2. **AVG()** - Calculates the average value

```
SELECT deptno, AVG(salary) AS avg_salary
FROM emps
GROUP BY deptno;
```

3. **MIN()** - Returns the minimum value

```
SELECT MIN(salary) AS lowest_salary
FROM emps;
```

4. **MAX()** - Returns the maximum value

```
SELECT MAX(salary) AS highest_salary
FROM emps;
```

5. **NULLIF()** - Returns NULL if two expressions are equal

```
SELECT ename, NULLIF(salary, 1250) AS salary_check
FROM emps;
```

~~~
---

## 🔄 **Usage of ORDER BY, GROUP BY, HAVING with Examples**

### **ORDER BY** - Sorts the result
```sql
SELECT ename, salary
FROM emps
ORDER BY salary DESC;
~~~

➡ *Lists employees sorted by salary in descending order.*

GROUP BY - Groups rows that have the same values

```sql
SELECT deptno, COUNT(empno) AS total_employees
FROM emps
GROUP BY deptno;
```

➡ *Shows the number of employees in each department.*

HAVING - Filters groups created by GROUP BY

```sql
SELECT deptno, AVG(salary) AS avg_salary
FROM emps
GROUP BY deptno
HAVING AVG(salary) > 60000;
```

➡ *Displays departments where the average salary is greater than 60,000.*

☑ Bonus Example Combining All

```sql
SELECT deptno, COUNT(empno) AS total_employees, AVG(salary) AS avg_salary
FROM emps
GROUP BY deptno
HAVING AVG(salary) > 60000
ORDER BY avg_salary DESC;
```

➡ *Groups empss by department, filters departments with average salary > 60K, and sorts the result.*
