Video-10

Topics to cover:

- Functions in SQL

# Functions

SQL functions are ready-made tools in SQL that help you do calculations, change text, work with dates, or summarize data.

Think of them like shortcuts — instead of writing long code, you just call a function.

# Types of Functions

1. Built-in Functions: These are ready-made functions that come with SQL. You don't need to create them just use them directly.

a. String Functions (e.g., CONCAT, LENGTH, SUBSTRING)

b. Numeric Functions (e.g., ABS, ROUND, CEIL)

c. Date and Time Functions (e.g., NOW, DATE_FORMAT, DATEDIFF)

d. Aggregate Functions (e.g., COUNT, SUM, AVG)

2. User-Defined Functions (UDFs) : These are custom functions that created by users to perform specific operations.

# Creating employees table to understand different Commands

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    salary DECIMAL(10,2),
    hire_date DATE,
    department VARCHAR(50)
);

INSERT INTO employees (emp_id, first_name, last_name, salary, hire_date, department)
VALUES
(101, 'Amit', 'Sharma', 45000, '2020-01-15', 'IT'),
(102, 'Priya', 'Mehta', 52000, '2019-03-10', 'HR'),
(103, 'Raj', 'Verma', 61000, '2021-07-25', 'Finance'),
(104, 'Sneha', 'Kapoor', 48000, '2022-11-05', 'IT'),
(105, 'Arjun', 'Singh', 70000, '2018-06-30', 'Finance');

select * from employees;
```

# String Functions (work with text)

```
-- 1.Upper : Converts names to uppercase.
SELECT *, UPPER(first_name)
FROM employees;

-- 2.Lower : Converts names to lowercase.
SELECT *, LOWER(last_name)
FROM employees;

-- 3.Length : Shows length of first name.
SELECT *, LENGTH(first_name)
FROM employees;

-- 4. CONCAT : Joins two strings.
SELECT *, CONCAT(first_name," ",last_name) as full_name
FROM employees;
```

# Numeric Functions (work with numbers)

-- 1. Round : Rounds monthly salary

```
SELECT *, ROUND(salary/12, 2) AS monthly_salary
FROM employees;
```

-- 2. ABS : Shows absolute value

```
SELECT *, ABS(salary - 50000) AS diff
FROM employees;
```

-- 3. Ceil : Rounds a number up to the nearest integer.

```
SELECT CEIL(4.2) AS ceil_example1;

SELECT CEIL(7.9) AS ceil_example2;
```

-- 4. Floor : Rounds a number down to the nearest integer.

```
SELECT FLOOR(4.2) AS floor_example1;

SELECT FLOOR(7.9) AS floor_example2;
```

# Date/Time Functions (work with dates)

-- 1. NOW() → current date and time
SELECT NOW() AS current_datetime;

-- 2. CURDATE() → current date only
SELECT CURDATE() AS today_date;

-- 3. YEAR - extracts year
SELECT *, YEAR(hire_date) AS hire_year
FROM employees;

-- 4. MONTH(join_date) → extracts month
SELECT *, MONTH(hire_date) AS hire_month_number
FROM employees;

SELECT *, MONTHNAME(hire_date) AS hire_month_name
FROM employees;

# Date/Time Functions (work with dates)

-- 5. DATEDIFF(end_date, start_date) → difference in days

```
SELECT *, DATEDIFF(CURDATE(), hire_date) AS days_worked
FROM employees;

SELECT *, TIMESTAMPDIFF(YEAR, hire_date,CURDATE()) AS years_worked
FROM employees;

SELECT *, TIMESTAMPDIFF(MONTH, hire_date,CURDATE()) AS months_worked
FROM employees;
```

```sql
# Aggregate Functions (work on groups of rows)

-- COUNT(*) → counts rows
SELECT COUNT(*) AS total_employees
FROM employees;

-- SUM(salary) → adds all salaries
SELECT SUM(salary) AS total_salary
FROM employees;

-- AVG(salary) → average salary
SELECT AVG(salary) AS avg_salary
FROM employees;

-- MAX(salary) → highest salary
SELECT MAX(salary) AS max_salary
FROM employees;

-- MIN(salary) → lowest salary
SELECT MIN(salary) AS min_salary
FROM employees;
```