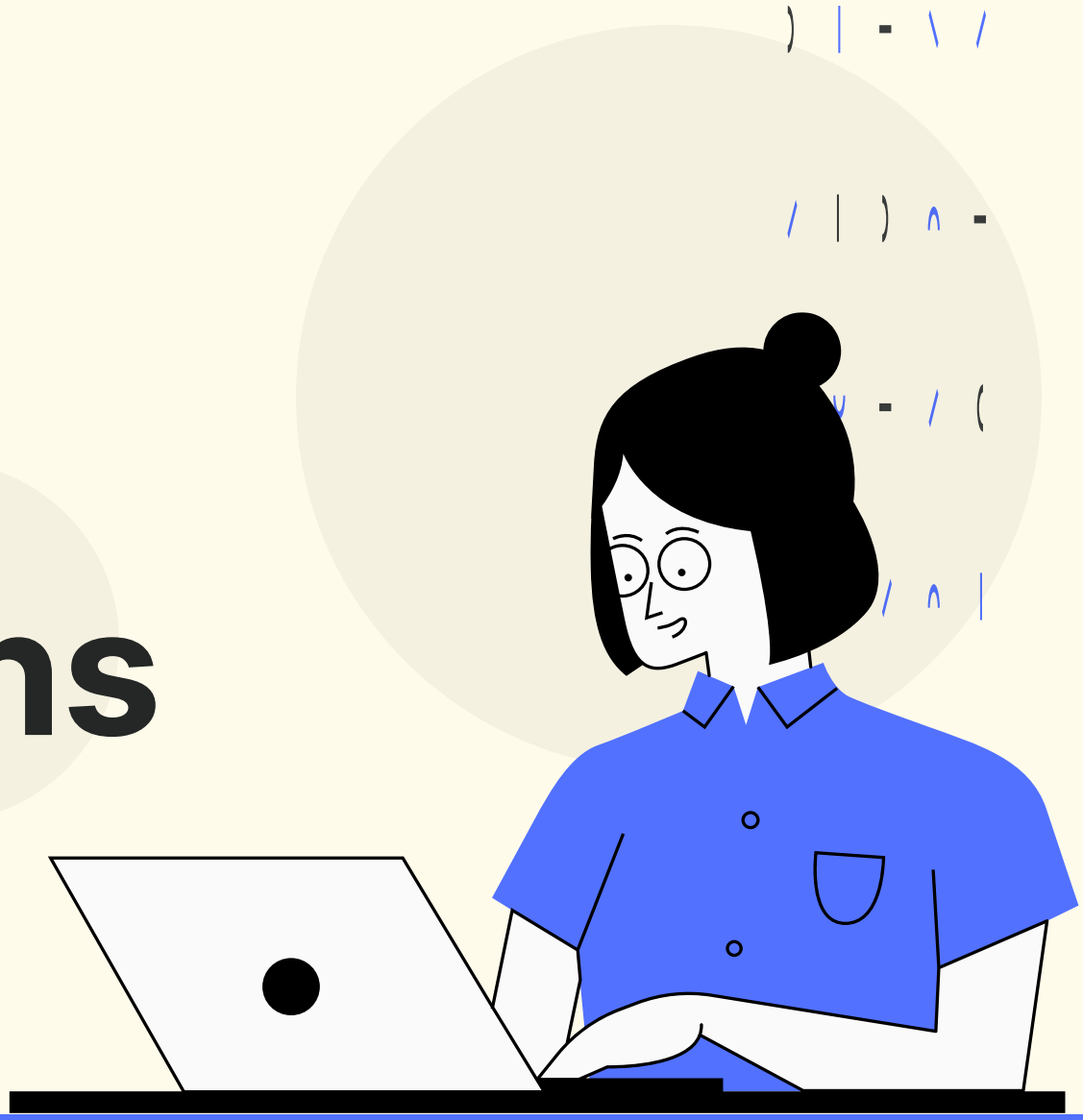
Mostly Asked SQL functions



- By Sakshi Yadav

COUNT()

Use Case: Counts the number of rows in a dataset.

```sql
SELECT COUNT(*) AS total_employees
FROM employees;
```

SUM()

Use Case: Adds up values in a numeric column.

```sql
SELECT SUM(salary) AS total_salary
FROM employees;
```

AVG()

Use Case: Calculates the average value of a numeric column.

```sql
SELECT AVG(salary) AS average_salary
FROM employees;
```

MAX()

Use Case: Finds the maximum value in a column.

```sql
SELECT MAX(salary) AS highest_salary
FROM employees;
```

MIN()

Use Case: Finds the minimum value in a column.

```sql
SELECT MIN(salary) AS lowest_salary
FROM employees;
```

ROUND()

Use Case: Rounds a numeric value to a specified number of decimal places.

```
SELECT ROUND(AVG(salary), 2) AS average_salary
FROM employees;
```

CONCAT()

Use Case: Combines two or more strings into one string.

```sql
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM employees;
```

LEN()

Use Case: Returns the length of a string.

```
SELECT LEN(first_name) AS name_length
FROM employees;
```

• SUBSTRING() •

Use Case: Extracts a portion of a string.

```sql
SELECT SUBSTRING(first_name, 1, 3) AS short_name
FROM employees;
```

TRIM()

Use Case: Removes leading and trailing spaces from a string.

```
SELECT TRIM(name) AS clean_name
FROM employees;
```

• UPPER() / LOWER() •

Use Case: Converts a string to uppercase or lowercase.

```
SELECT UPPER(first_name) AS uppercase_name
FROM employees;
```

COALESCE()

Use Case: Returns the first non-null value from a list of expressions.

```sql
SELECT COALESCE(phone_number, 'N/A') AS contact_number
FROM employees;
```

CASE

Use Case: Creates conditional logic in SQL queries.

```sql
SELECT first_name,
         CASE
           WHEN salary > 50000 THEN 'High'
           WHEN salary BETWEEN 30000 AND 50000 THEN 'Medium'
           ELSE 'Low'
         END AS salary_band
  FROM employees;
```

DATEPART()

Use Case: Extracts a specific part of a date.

```
SELECT DATEPART(year, hire_date) AS hire_year
FROM employees;
```

DATEDIFF()

Use Case: Calculates the difference between two dates.

```
SELECT DATEDIFF(day, hire_date, GETDATE()) AS days_since_hired
FROM employees;
```