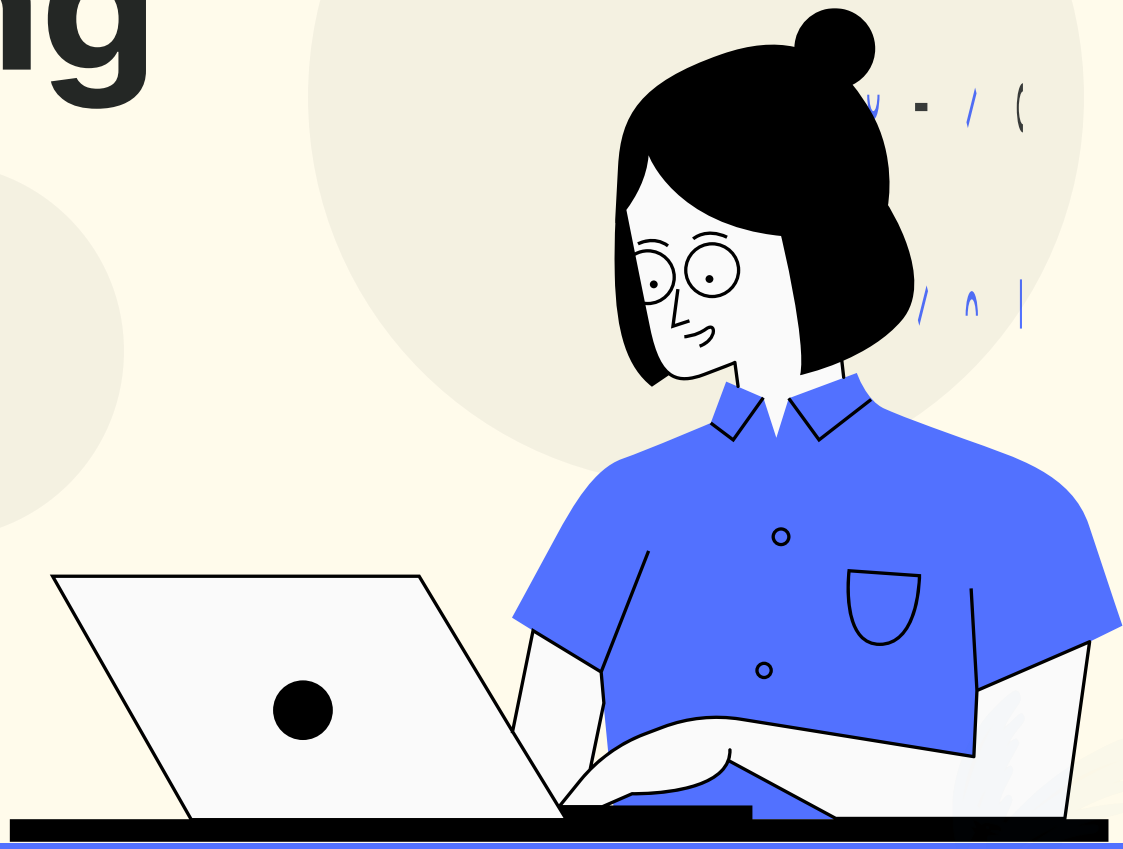
Data Cleaning using SQL



- By Sakshi Yadav

Missing Values

Find Missing Values: Use **IS NULL** to find missing values.

```
SELECT * FROM your_table WHERE column_name IS NULL;
```

Missing Values

Replace Missing Values: Use **COALESCE** or **ISNULL** to provide default values.

```
UPDATE your_table
SET column_name = COALESCE(column_name, 'default_value')
WHERE column_name IS NULL;
```

Remove Duplicates

Find Duplicates

```sql
SELECT column_name, COUNT(*) FROM your_table
GROUP BY column_name
HAVING COUNT(*) > 1;
```

Remove Duplicates

Delete Duplicates (keeping the first occurrence**)**

```sql
DELETE FROM your_table
WHERE id NOT IN (
  SELECT MIN(id)
  FROM your_table
  GROUP BY column_name
);
```

• Standardize Data Formats •

Update Data Formats: Ensure consistency in data formats, such as date formats or phone numbers.

```
UPDATE your_table
SET date_column = TO_DATE(date_column, 'YYYY-MM-DD')
WHERE date_column IS NOT NULL;
```

• Remove Unnecessary Data •

Delete Unnecessary Rows

```
DELETE FROM your_table
WHERE condition_to_remove_rows;
```

Correct Data Inconsistencies

Correct Specific Issues

```
UPDATE your_table
SET column_name = 'correct_value'
WHERE column_name = 'incorrect_value';
```

Normalize Data

Create New Tables for Normalization: Break down large tables into smaller related tables.

```sql
CREATE TABLE new_table AS
SELECT DISTINCT column_name
FROM your_table;
```

Validate Data

Check Constraints: Ensure data adheres to business rules.

```sql
SELECT *
FROM your_table
WHERE column_name NOT BETWEEN lower_bound AND upper_bound;
```

• Handle Outliers •

Identify Outliers

```sql
SELECT *
FROM your_table
WHERE
column_name > (SELECT AVG(column_name) + 3 * STDDEV(column_name)
FROM your_table)
OR
column_name < (SELECT AVG(column_name) - 3 * STDDEV(column_name)
FROM your_table);
```