Best Practices and Techniques

Data Cleaning in SQL

Presented by: Omotosho Taiwo Data Scientist

Date: April 03, 2025

Introduction

- Data cleaning is essential for ensuring that data is accurate, consistent, and reliable. In SQL, various techniques are used to clean data, including handling missing values, removing duplicates, standardizing formats, and more.

Handling Missing Data

- Missing data is a common issue in datasets. Use SQL functions to identify and handle missing values.
- **Identifying NULL values:**

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

- **Replacing NULL values with a default value:**

```
SELECT COALESCE(column_name, 'Default Value') AS cleaned_column FROM table_name;
```

```
SELECT AVG(column_name) FROM table_name WHERE column_name IS NOT NULL;
```

Removing Duplicates

- Duplicates can distort analysis. SQL provides methods to identify and remove them.

- **Identifying duplicates:**

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

Removing duplicates:

```
WITH CTE AS (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY column_name ORDER BY column_name) AS row_num
    FROM table_name
)
DELETE FROM CTE WHERE row_num > 1;
```

Standardizing Data Formats

- **Standardizing data formats ensures consistency across the dataset.**

- **Date Formatting:**

```
SELECT DATE_FORMAT(date_column, '%Y-%m-%d') AS standardized_date FROM table_name;
```

- **Phone Number Standardization:**

```
SELECT REGEXP_REPLACE(phone_number, '[^\d]', '',
                      'g') AS cleaned_phone_number
FROM table_name;
```

Outlier Detection and Removal

- Outliers can skew your analysis. Detect and remove them using statistical methods.

- **Identifying outliers based on z-scores:**

```
SELECT column_name
FROM table_name
WHERE ABS(column_name -
            (SELECT AVG(column_name)
             FROM table_name)) >
            3 * (SELECT STDDEV(column_name)
                 FROM table_name);
```

Removing outliers

```
DELETE FROM table_name
WHERE ABS(column_name -
    (SELECT AVG(column_name)
     FROM table_name)) >
    3 * (SELECT STDDEV(column_name)
         FROM table_name);
```


```
DELETE FROM table_name
WHERE ABS(column_name -
    (SELEC
     FROM
    3 * (SELECT STDDEV(column_name)
         FROM table_name);
```

Handling Inconsistent Data

- Inconsistent data entries can be fixed by standardizing values and categories.

- Standardizing text values:

```
SELECT TRIM(LOWER(column_name)) AS cleaned_column

FROM table_name;
```

Correcting inconsistent categories:

```sql
SELECT CASE
            WHEN column_name IN ('cat', 'Cat', 'cats')
            THEN 'Cat'
            WHEN column_name IN ('dog', 'Dog', 'dogs')
            THEN 'Dog'
            ELSE column_name
        END AS standardized_column
FROM table_name;
```

Validation Checks

- SQL allows you to implement validation checks to ensure data quality.
- **Checking for invalid data ranges:**

```sql
SELECT *
FROM table_name
WHERE column_name < 0 OR
      column_name > 100;
```

Checking data integrity:

```sql
SELECT *
FROM orders o
LEFT JOIN customers c
ON o.customer_id = c.customer_id
WHERE c.customer_id IS NULL;
```

Conclusion

- Data cleaning in SQL is essential for ensuring accurate and reliable data. By using the right techniques, you can prepare your data for deeper analysis and decision-making.