

9 SQL Techniques for Effective Data Cleaning



Sai Kumar Bysani

1. Replace NULL values with a default value or remove them based on the context.

...

```
SELECT COALESCE(column_name, 'DefaultValue') FROM table_name;
```

2. Identify and delete duplicate rows based on key columns.

```
...  
WITH CTE AS (  
    SELECT  
        id,  
        column1, column2,  
        ROW_NUMBER() OVER (PARTITION BY column1, column2 ORDER BY id) AS row_num  
    FROM table_name  
)  
DELETE FROM table_name  
WHERE id IN (  
    SELECT id  
    FROM CTE  
    WHERE row_num > 1  
);
```

3. Convert text to lower or upper case to ensure consistency.

```
SELECT LOWER(column_name) AS cleaned_column FROM table_name;
```

4. Remove extra spaces from text fields.

```
...  
SELECT TRIM(column_name) FROM table_name;
```

5. Convert date strings into a consistent date format.

```
...  
SELECT STR_TO_DATE(column_name, '%m/%d/%Y') FROM table_name;
```

6. Identify and manage outliers in numerical data.

```
...
```

```
SELECT * FROM table_name WHERE column_name BETWEEN lower_limit AND  
upper_limit;
```

7. Replace or remove special characters in text fields.

```
...  
SELECT REGEXP_REPLACE(column_name, '[^a-zA-Z0-9]', '') AS cleaned_column  
FROM table_name;
```

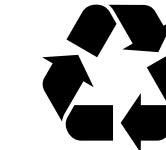
8. Standardize values in categorical columns

```
UPDATE table_name  
SET column_name = 'Male'  
WHERE column_name IN ('M', 'male');
```

9. Use statistical methods to fill in missing numerical data.

```
...  
UPDATE table_name  
SET column_name = (SELECT AVG(column_name) FROM table_name)  
WHERE column_name IS NULL;
```

If you found this helpful...

 **React**
 **Comment**
 **Share**



Sai Kumar Bysani