Data Engineering 101

Data Cleaning using SQL

Shwetank Singh
GritSetGrow - GSGLearn.com

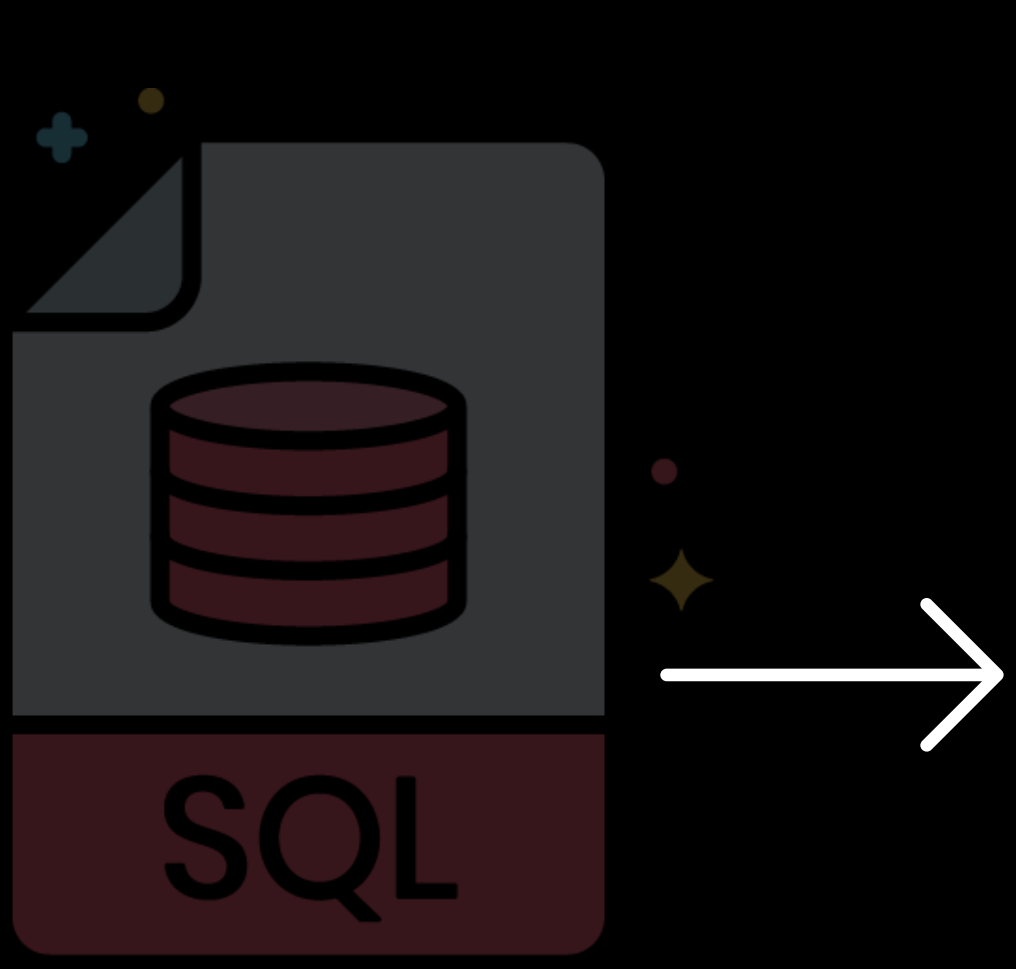
Data Engineering 101: Data Cleaning by SQL

Remove leading/trailing spaces

```
UPDATE table_name
SET column_name =TRIM(column_name);
```

Uses the TRIM function to remove spaces from both ends of the string in the specified column.

Shwetank Singh
GritSetGrow - GSGLearn.com



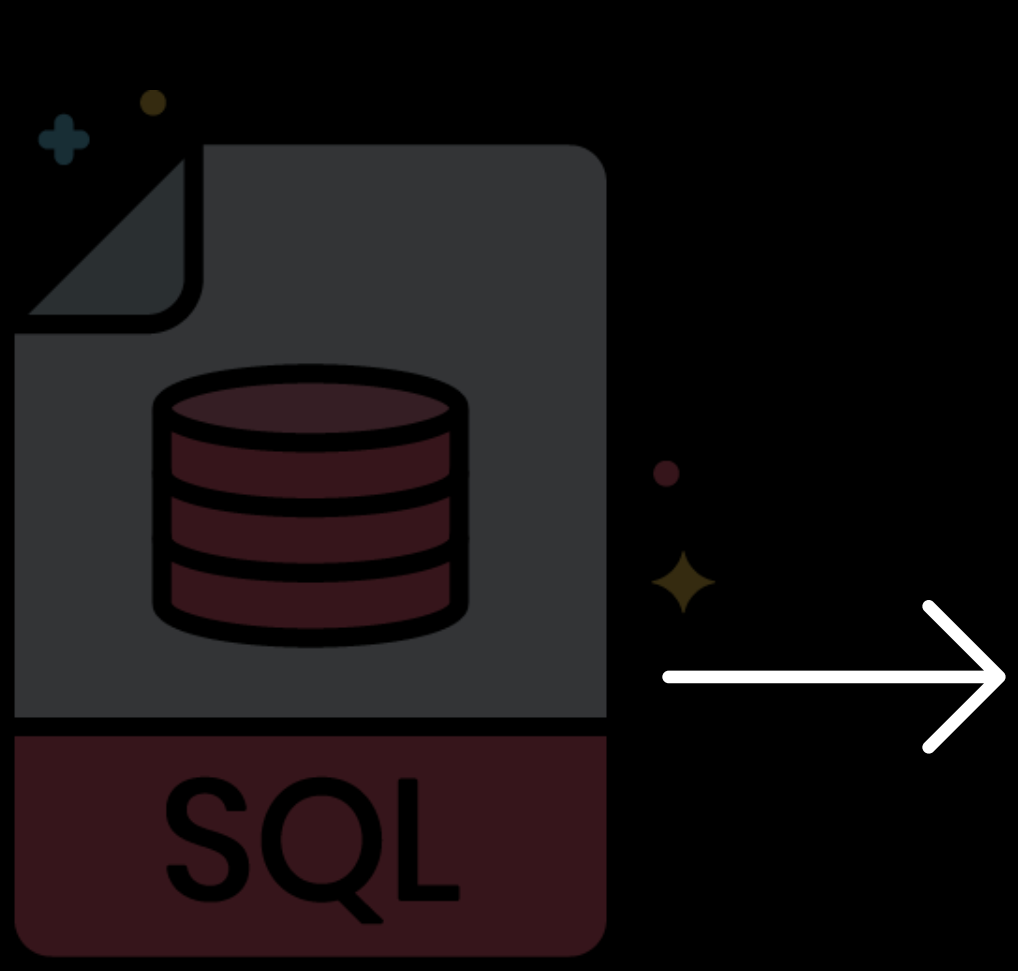
Data Engineering 101: Data Cleaning by SQL

Convert text to uppercase

```
UPDATE table_name
SET column_name = UPPER(column_name);
```

Converts all characters in the specified column to uppercase using the UPPER function.

Shwetank Singh
GritSetGrow - GSGLearn.com



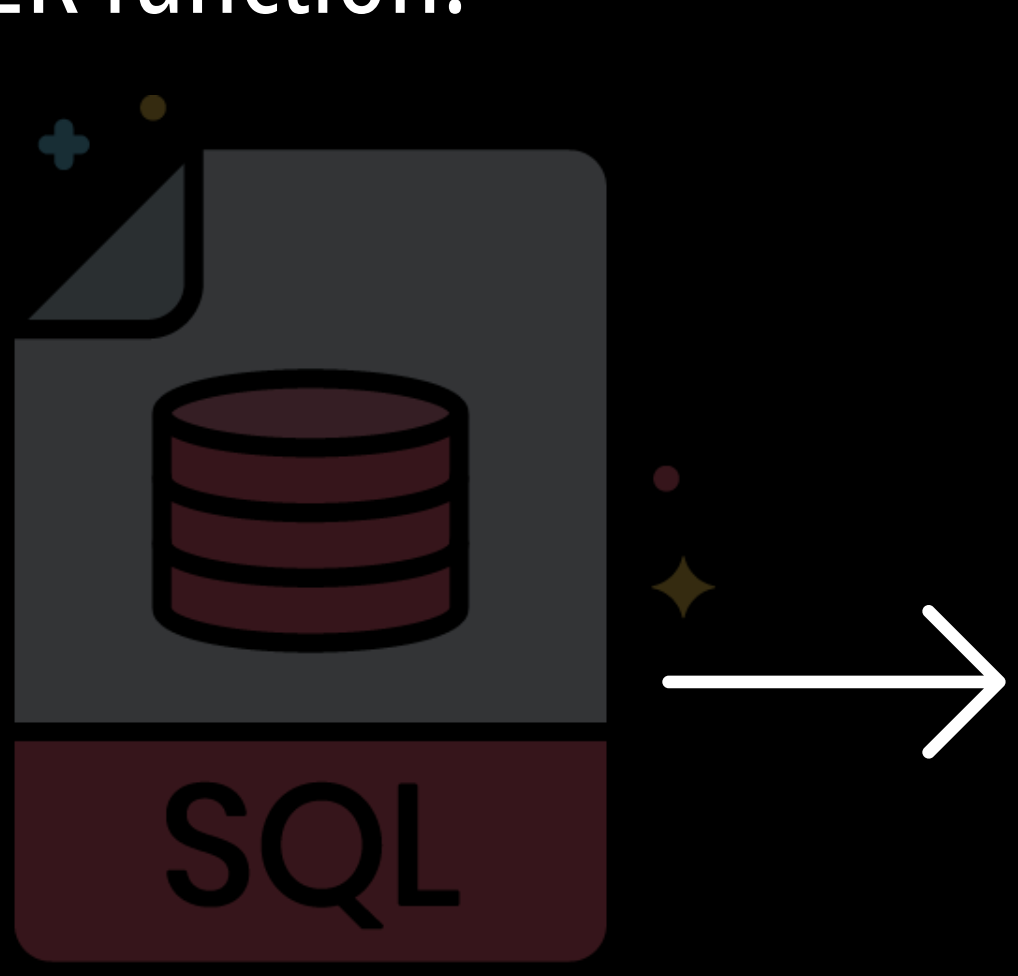
Data Engineering 101: Data Cleaning by SQL

Convert text to lowercase

```
UPDATE table_name
SET
column_name = LOWER(column_name);
```

Converts all characters in the specified column to lowercase using the LOWER function.

Shwetank Singh
GritSetGrow - GSGLearn.com



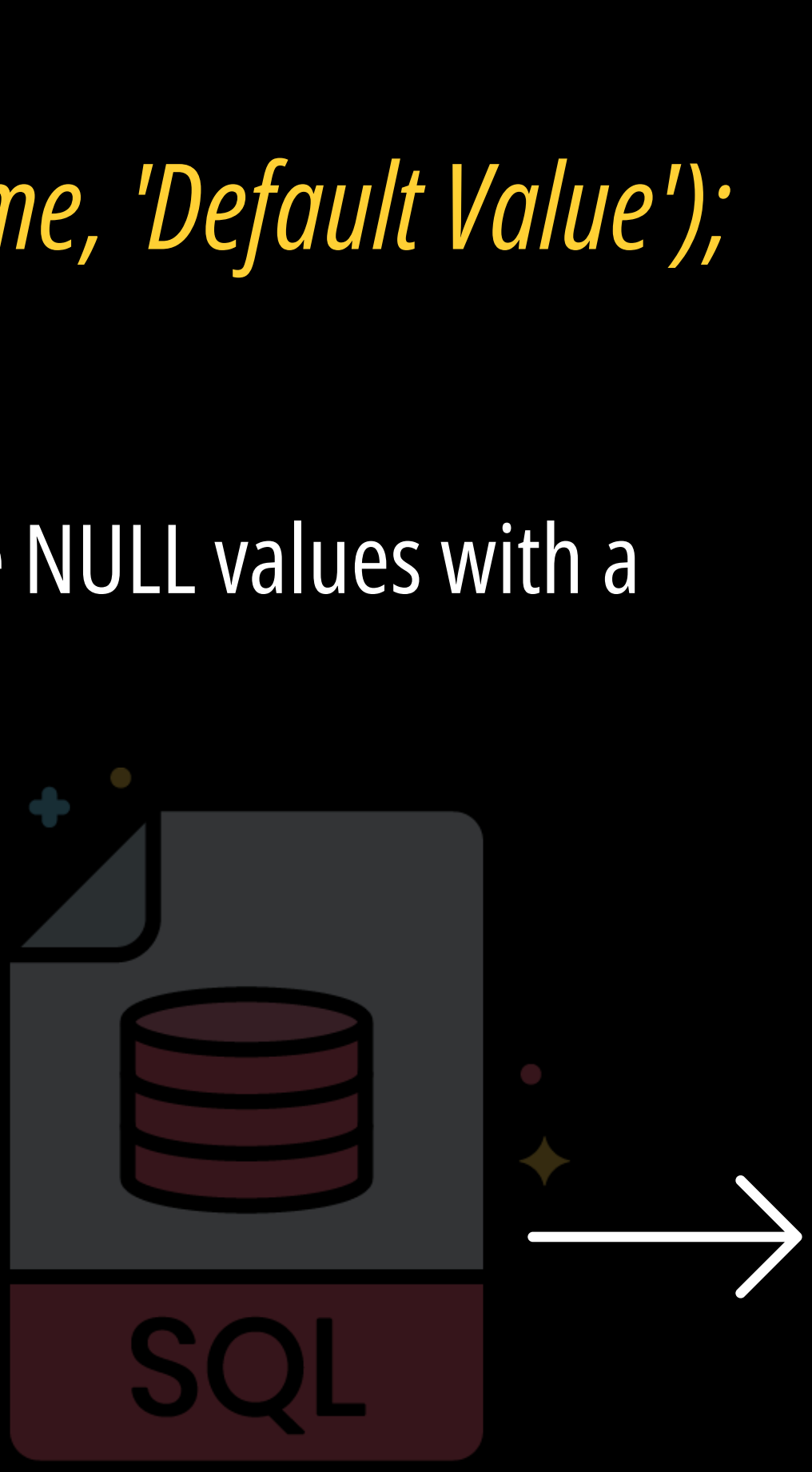
Data Engineering 101: Data Cleaning by SQL

Replace NULL values with a default value

```
UPDATE table_name
SET column_name =
COALESCE(column_name, 'Default Value');
```

Uses COALESCE to replace NULL values with a specified default value.

Shwetank Singh
GritSetGrow - GSGLearn.com

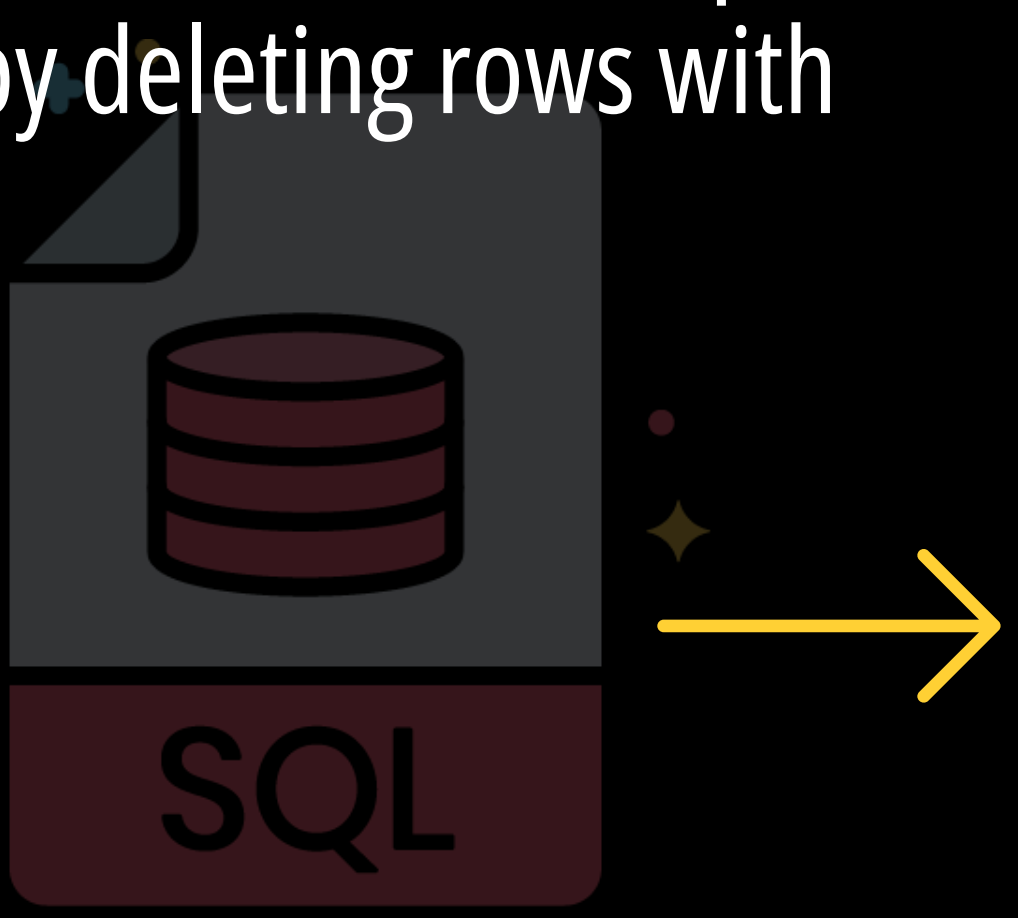


Data Engineering 101: Data Cleaning by SQL

Remove duplicate rows

```
DELETE FROM table_name
WHERE id
NOT IN (SELECT MIN(id)
FROM table_name
GROUP BY column1, column2, column3);
```

Keeps only the first occurrence of each unique combination of columns by deleting rows with higher IDs.



Shwetank Singh
GritSetGrow - GSGLearn.com

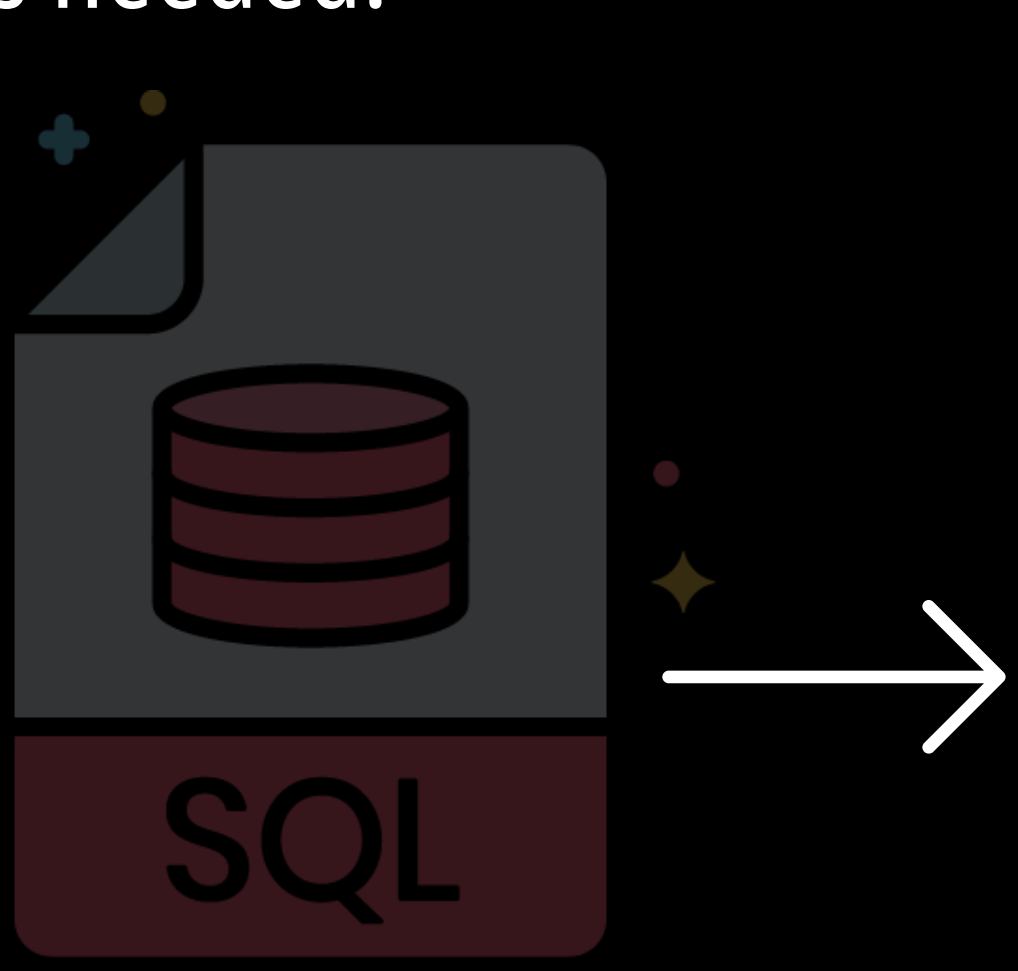
Data Engineering 101: Data Cleaning by SQL

Convert date format

```
UPDATE table_name
SET date_column =
TO_DATE(date_column, 'MM/DD/YYYY');
```

Converts a string date to a proper DATE format.
Adjust the format string as needed.

Shwetank Singh
GritSetGrow - GSGLearn.com



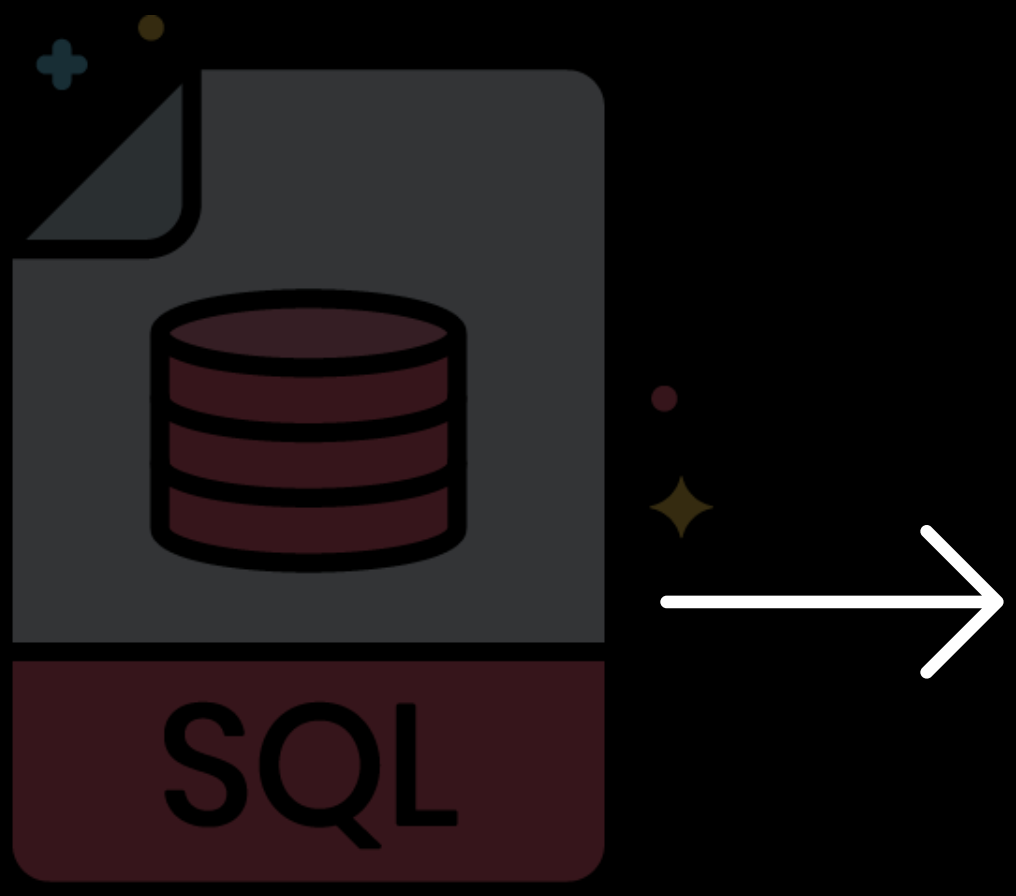
Data Engineering 101: Data Cleaning by SQL

Extract year from date

```
SELECT
EXTRACT(YEAR FROM date_column)
AS year
FROM table_name;
```

Extracts the year from a date column using the EXTRACT function.

Shwetank Singh
GritSetGrow - GSGLearn.com



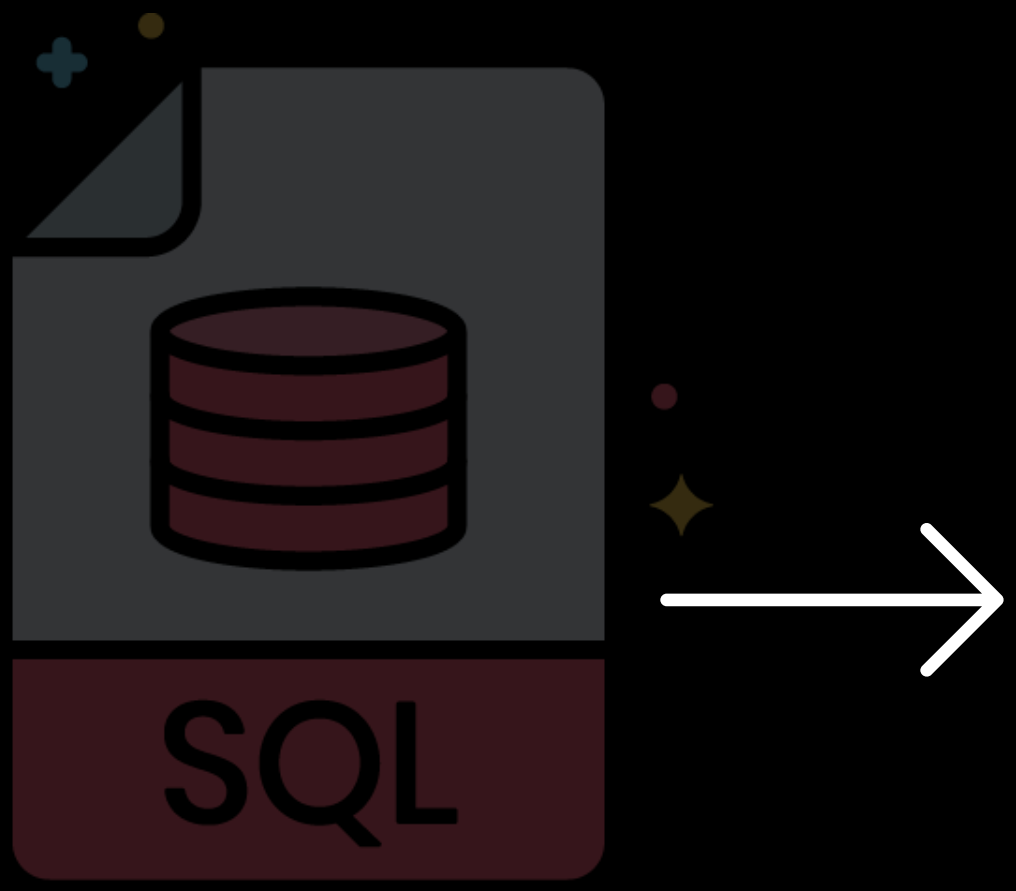
Data Engineering 101: Data Cleaning by SQL

Standardize phone numbers

```
UPDATE table_name
SET phone_column =
REGEXP_REPLACE(phone_column, '[^0-9]', '');
```

Removes all non-numeric characters from phone numbers using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com



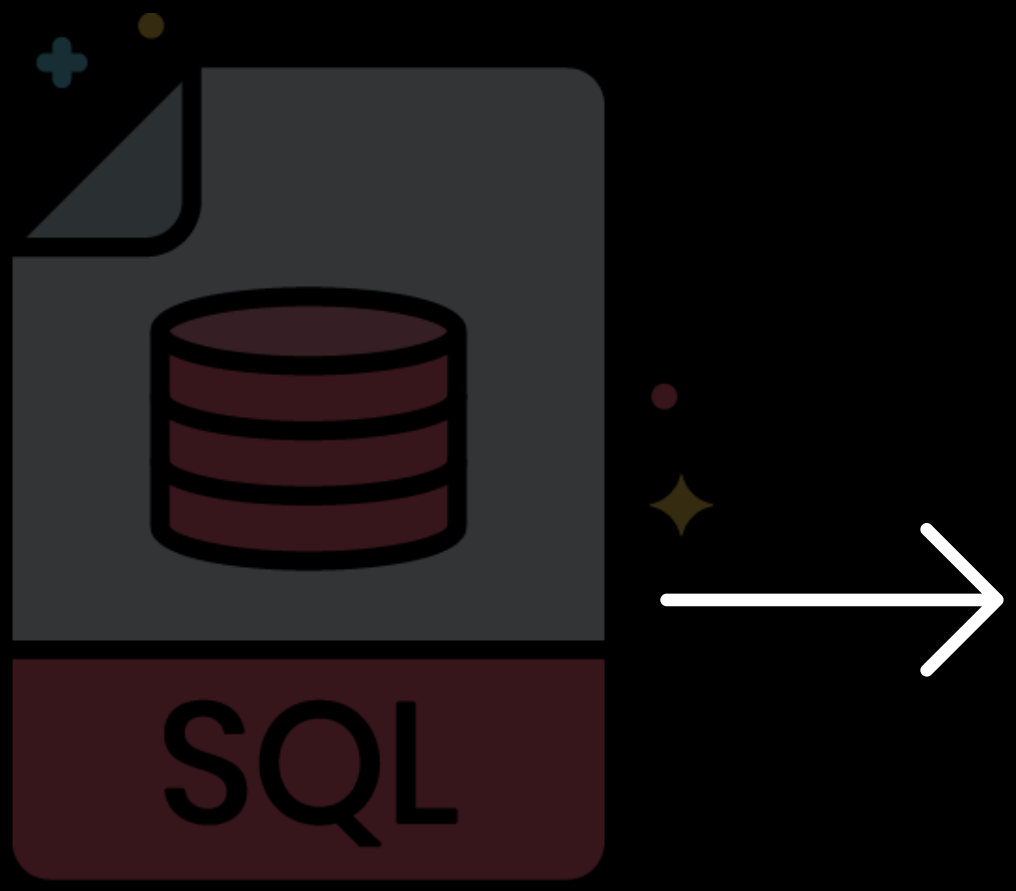
Data Engineering 101: Data Cleaning by SQL

Split full name into first and last name

```
UPDATE table_name
SET first_name = SUBSTRING_INDEX(full_name, ' ', 1),
last_name = SUBSTRING_INDEX(full_name, ' ', -1);
```

Splits a full name into first and last name using SUBSTRING_INDEX (MySQL syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



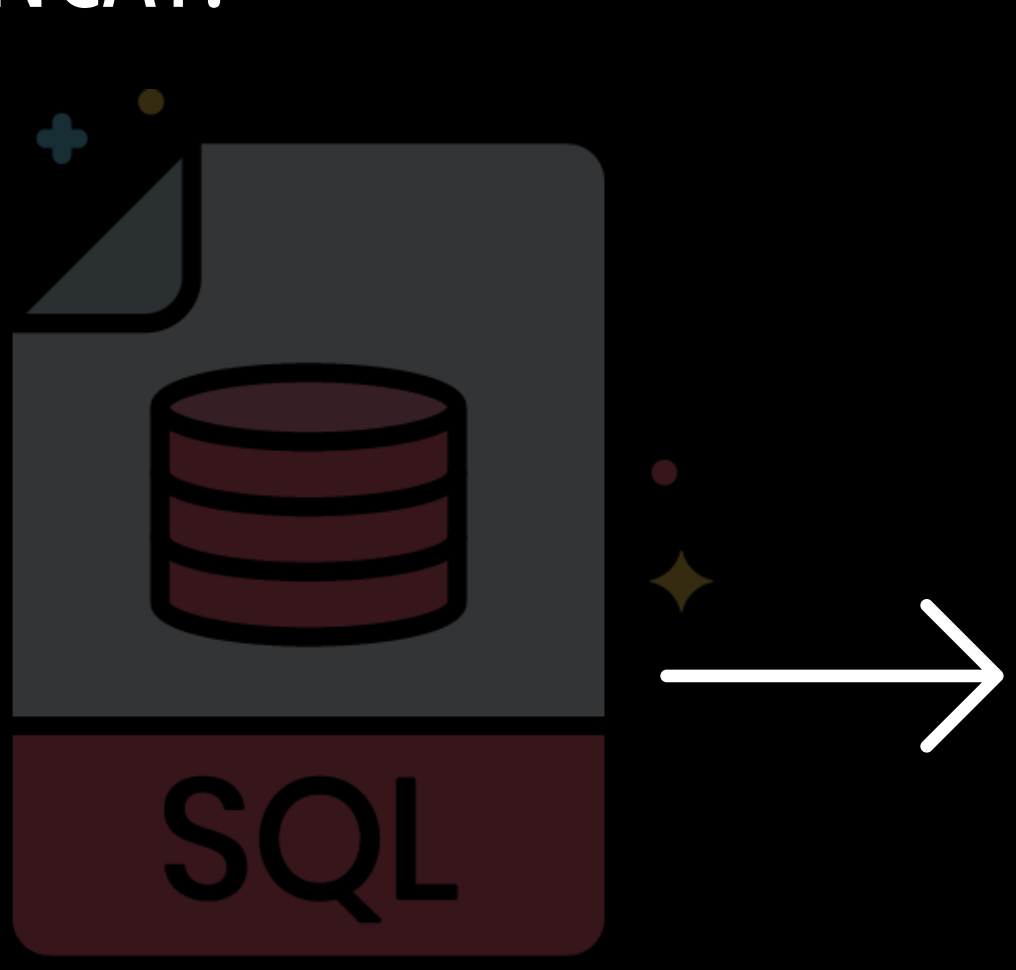
Data Engineering 101: Data Cleaning by SQL

Combine columns

```
UPDATE table_name
SET
full_address = CONCAT(street, ', ', city, ',
', state, ' ', zip_code);
```

Concatenates multiple columns into a single full address column using CONCAT.

Shwetank Singh
GritSetGrow - GSGLearn.com



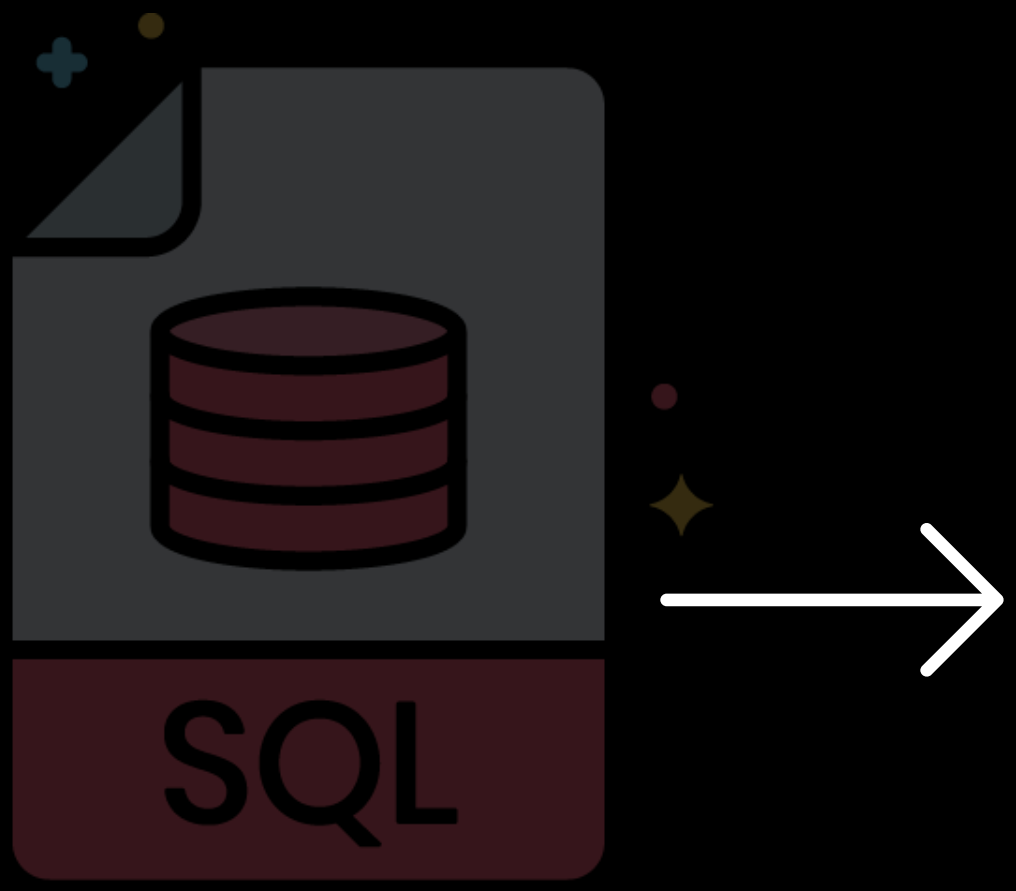
Data Engineering 101: Data Cleaning by SQL

Remove special characters

```
UPDATE table_name
SET column_name =
REGEXP_REPLACE(column_name, '[^a-zA-Z0-9 ]', '');
```

Removes all characters that are not alphanumeric or spaces using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com

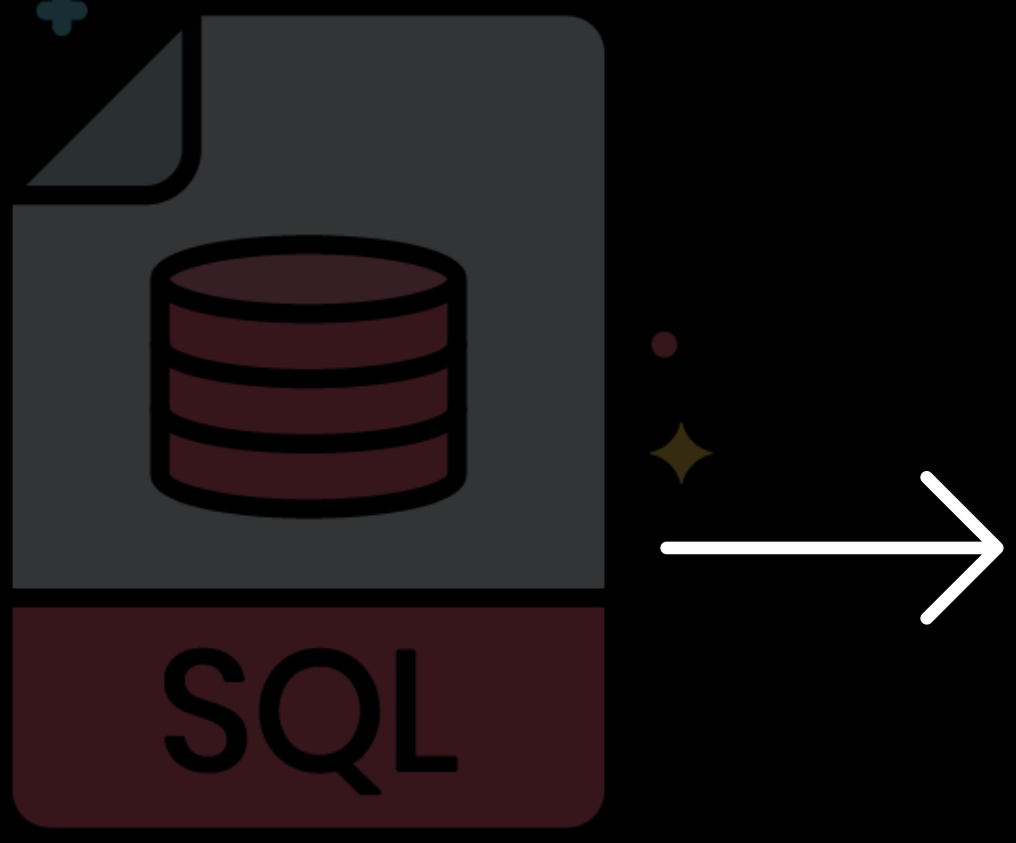


Data Engineering 101: Data Cleaning by SQL

Standardize state abbreviations

```
UPDATE table_name
SET state = CASE
WHEN state = 'California' THEN 'CA'
WHEN state = 'New York' THEN 'NY' ...
END;
```

Uses a CASE statement to convert full state names to standard abbreviations.



Shwetank Singh
GritSetGrow - GSGLearn.com

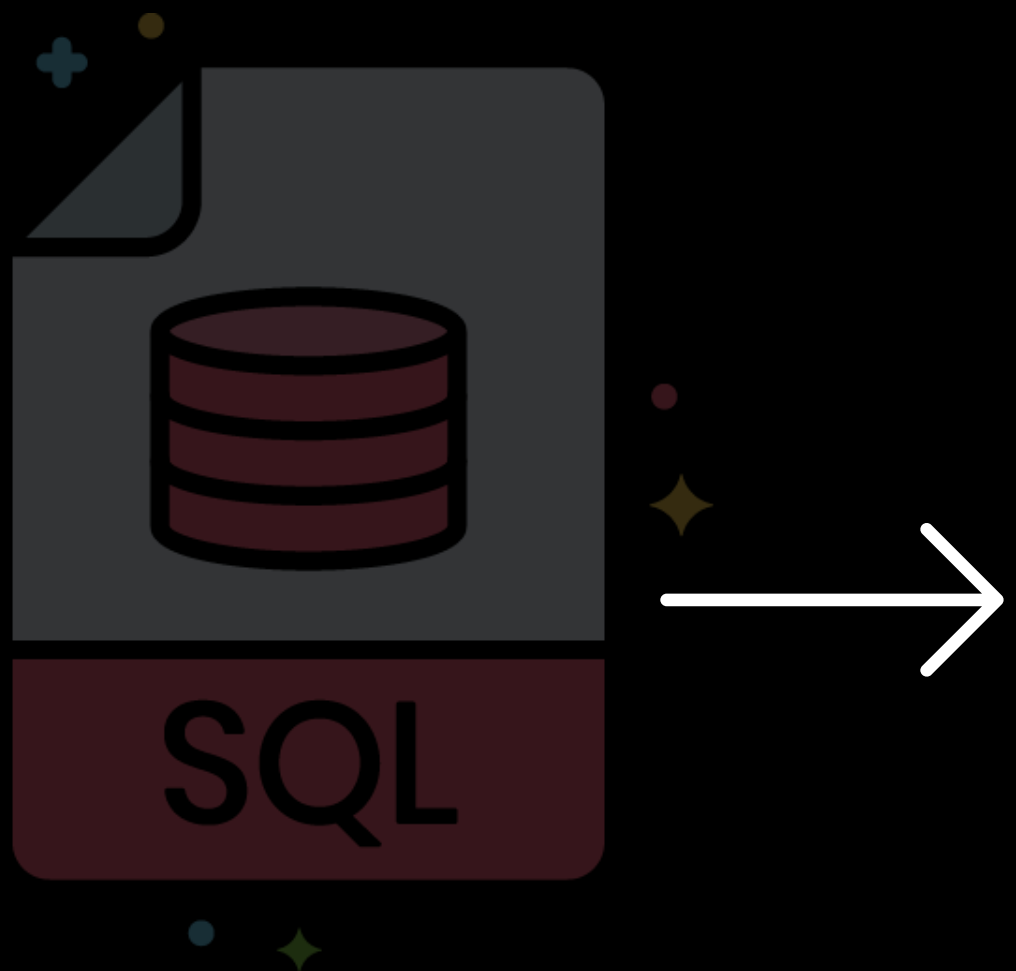
Data Engineering 101: Data Cleaning by SQL

Fix capitalization of proper nouns

```
UPDATE table_name
SET city_name = INITCAP(city_name);
```

Capitalizes the first letter of each word in the city name using INITCAP (Oracle syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



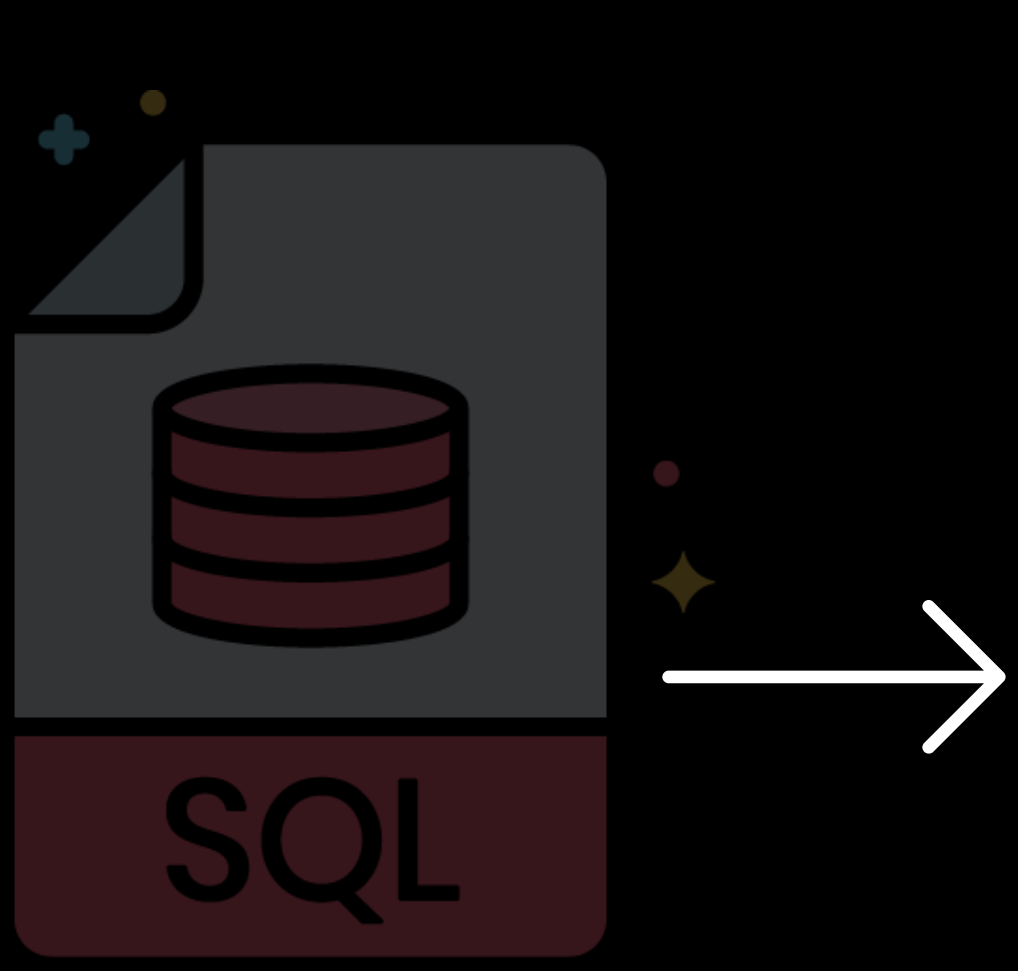
Data Engineering 101: Data Cleaning by SQL

Convert empty strings to NULL

```
UPDATE table_name
SET
column_name = NULLIF(column_name, '');
```

Replaces empty strings with NULL values using the NULLIF function.

Shwetank Singh
GritSetGrow - GSGLearn.com



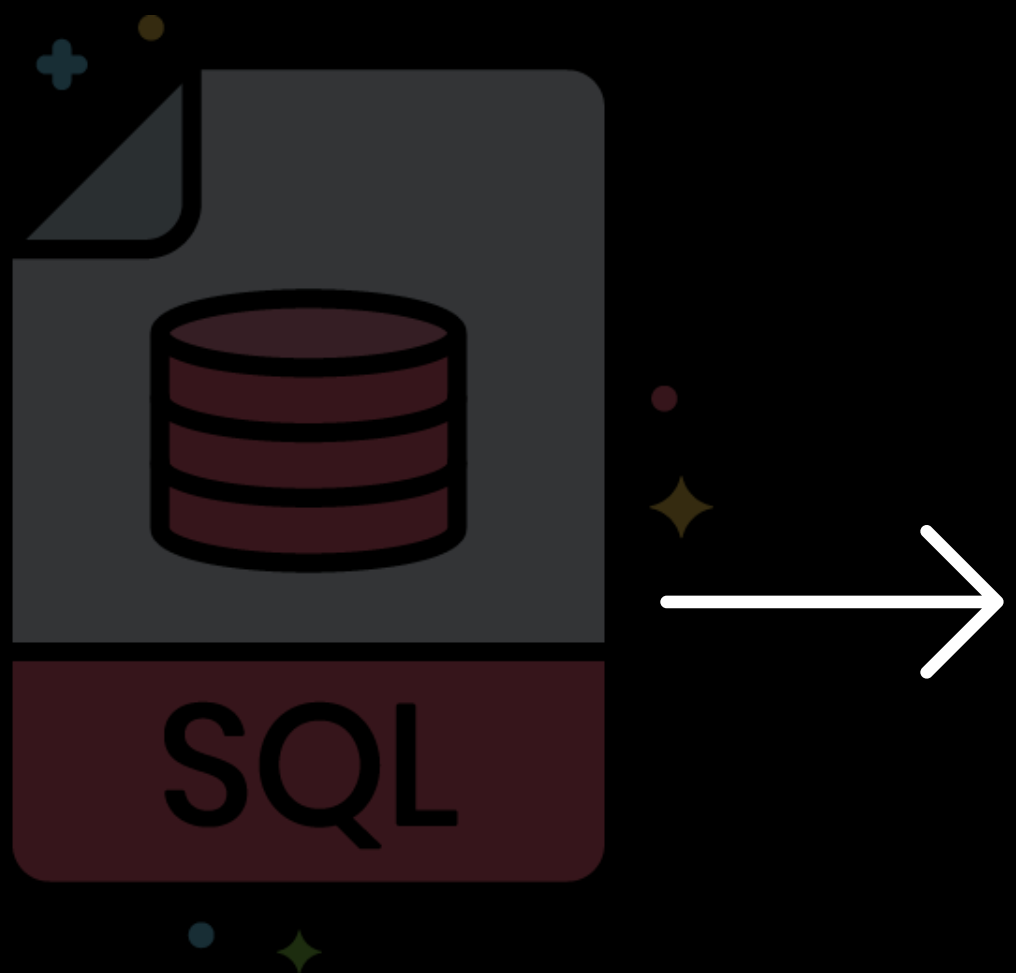
Data Engineering 101: Data Cleaning by SQL

Round numeric values

```
UPDATE table_name
SET
numeric_column = ROUND(numeric_column, 2);
```

Rounds numeric values to a specified number of decimal places using the ROUND function.

Shwetank Singh
GritSetGrow - GSGLearn.com



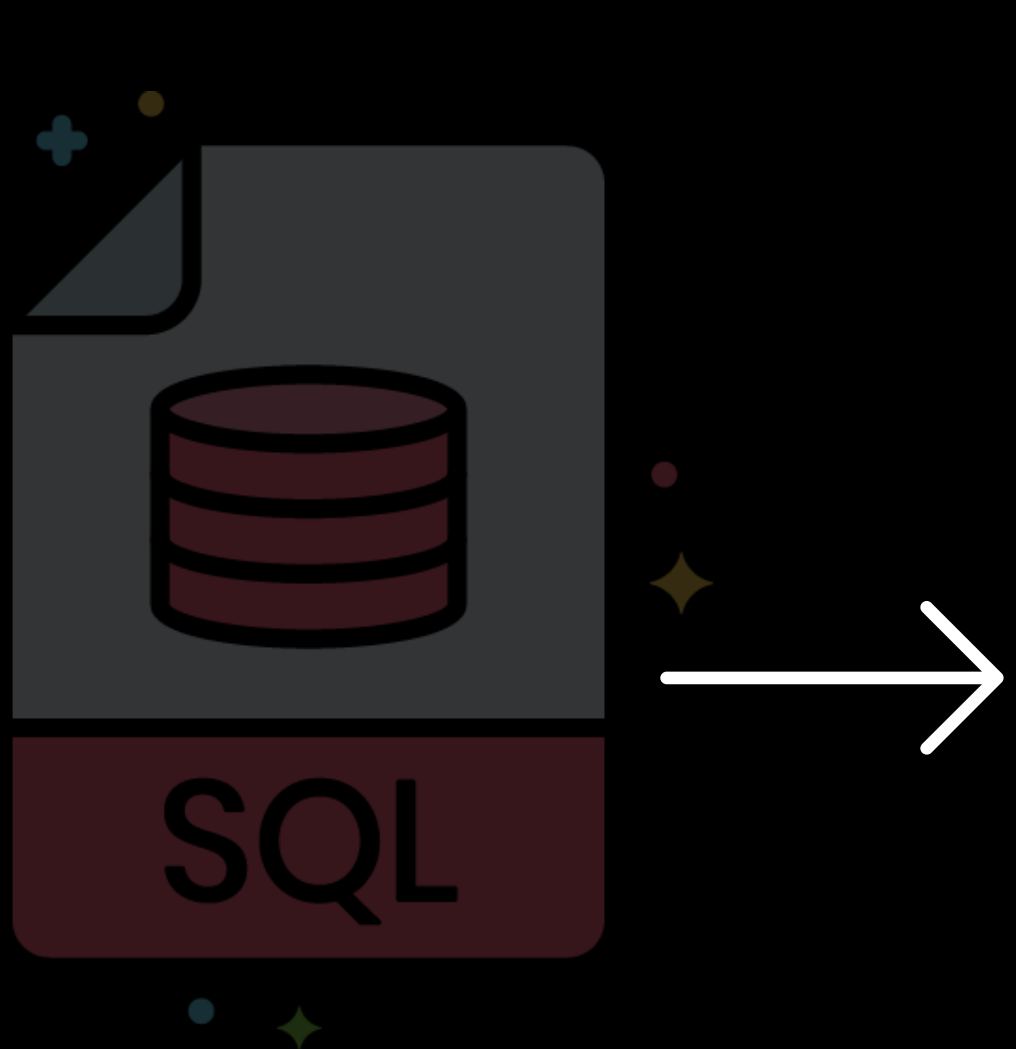
Data Engineering 101: Data Cleaning by SQL

Remove rows with all NULL values

```
DELETE FROM table_name
WHERE (column1 IS NULL AND column2
IS NULL AND column3 IS NULL);
```

Deletes rows where all specified columns contain NULL values.

Shwetank Singh
GritSetGrow - GSGLearn.com



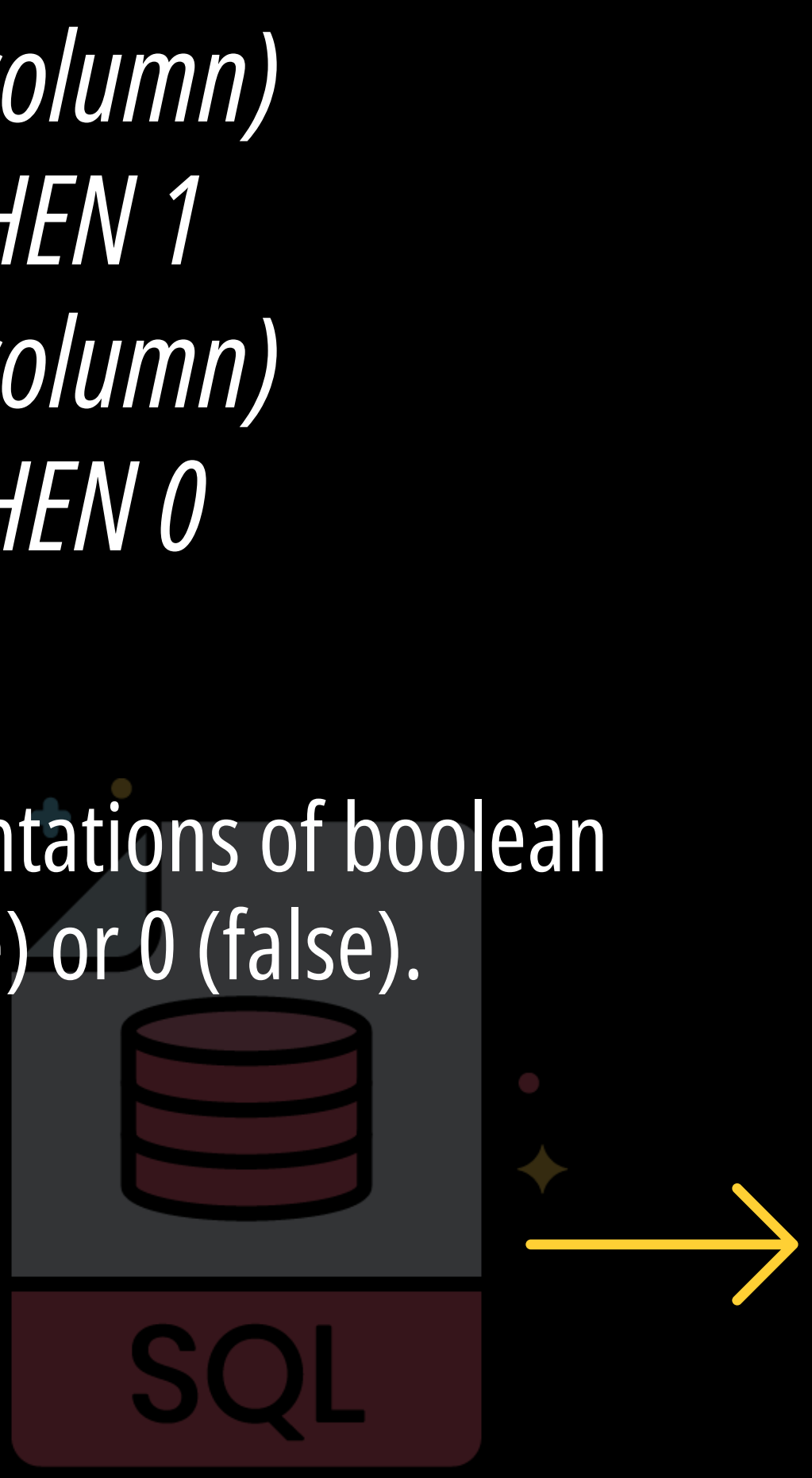
Data Engineering 101: Data Cleaning by SQL

Standardize boolean values

```
UPDATE table_name
SET bool_column = CASE
WHEN LOWER(bool_column)
IN ('yes', 'true', '1') THEN 1
WHEN LOWER(bool_column)
IN ('no', 'false', '0') THEN 0
ELSE NULL END;
```

Converts various representations of boolean values to standard 1 (true) or 0 (false).

Shwetank Singh
GritSetGrow - GSGLearn.com



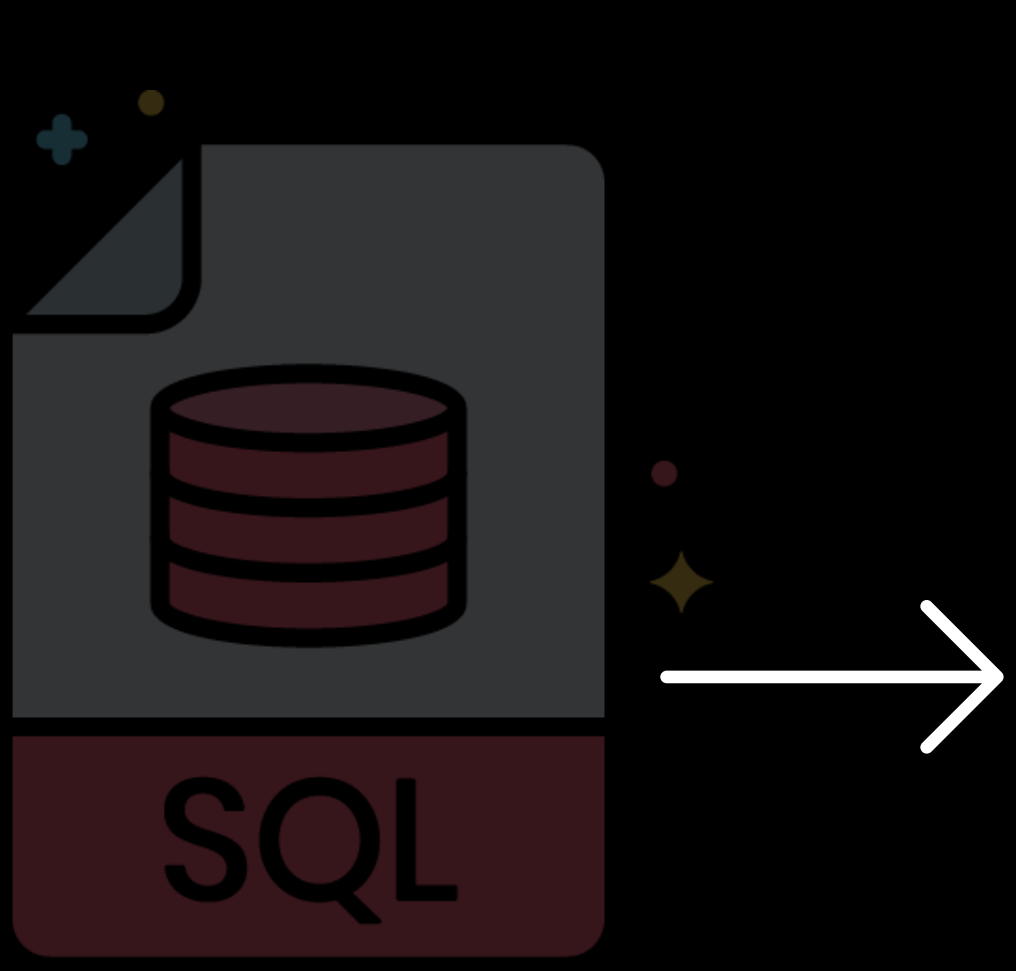
Data Engineering 101: Data Cleaning by SQL

Remove HTML tags

```
UPDATE table_name
SET
html_column =
REGEXP_REPLACE(html_column, '<[^>]+>', '');
```

Removes HTML tags from text using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com



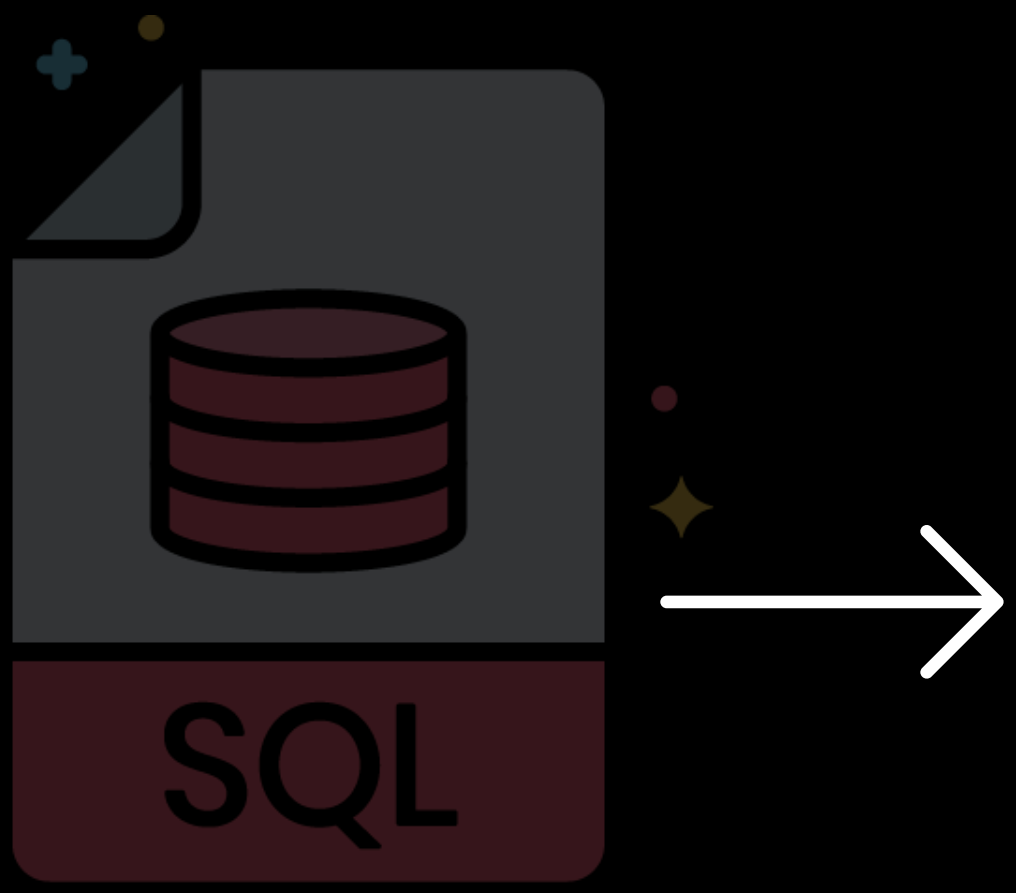
Data Engineering 101: Data Cleaning by SQL

Convert timezone

```
UPDATE table_name
SET timestamp_column =
timestamp_column AT TIME ZONE 'UTC'
AT TIME ZONE 'US/Pacific';
```

Converts timestamps from one timezone to another (PostgreSQL syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Handle misspellings

```
UPDATE table_name
SET product_name = CASE
WHEN product_name LIKE '%labtop%'
THEN REPLACE(product_name, 'labtop', 'laptop')
ELSE product_name END;
```

Corrects common misspellings using CASE and REPLACE functions.

Shwetank Singh
GritSetGrow - GSGLearn.com

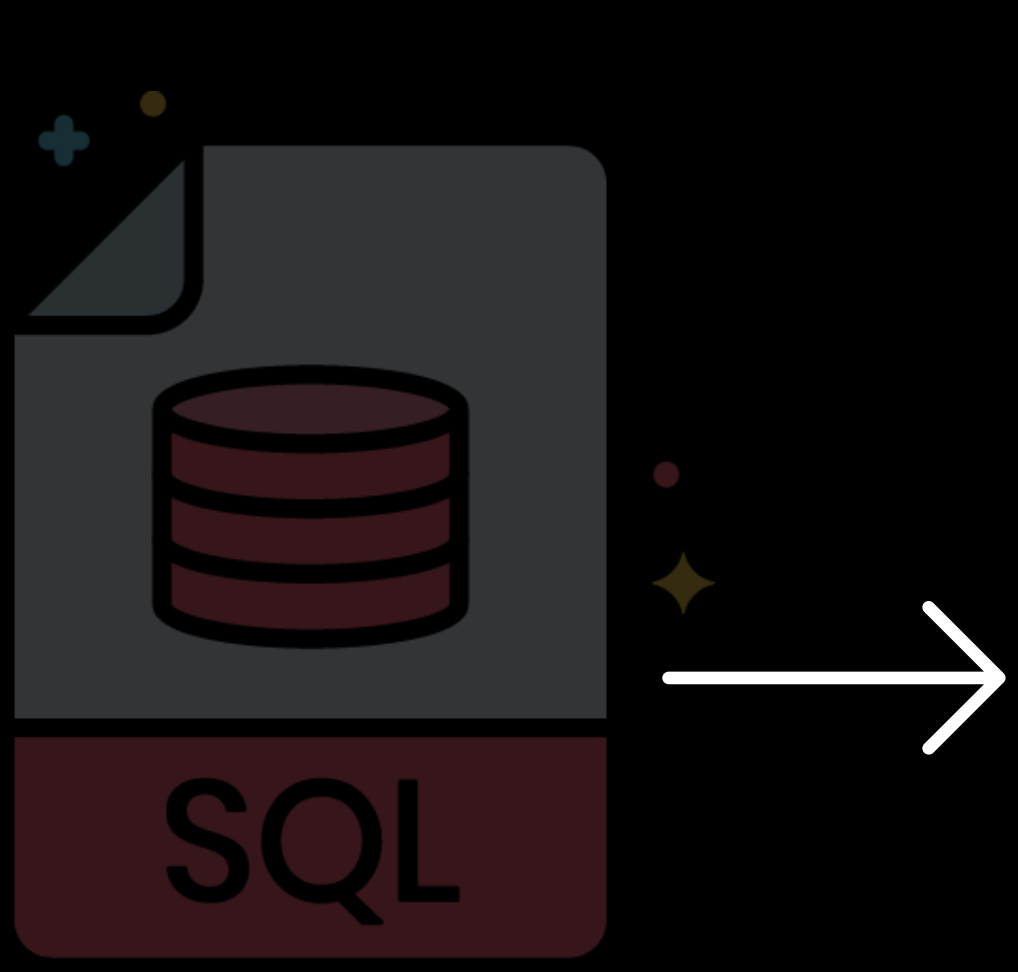
Data Engineering 101: Data Cleaning by SQL

Remove non-printable characters

```
UPDATE table_name
SET text_column =
REGEXP_REPLACE(text_column, '[^[:print:]]', '');
```

Removes non-printable characters from text using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com



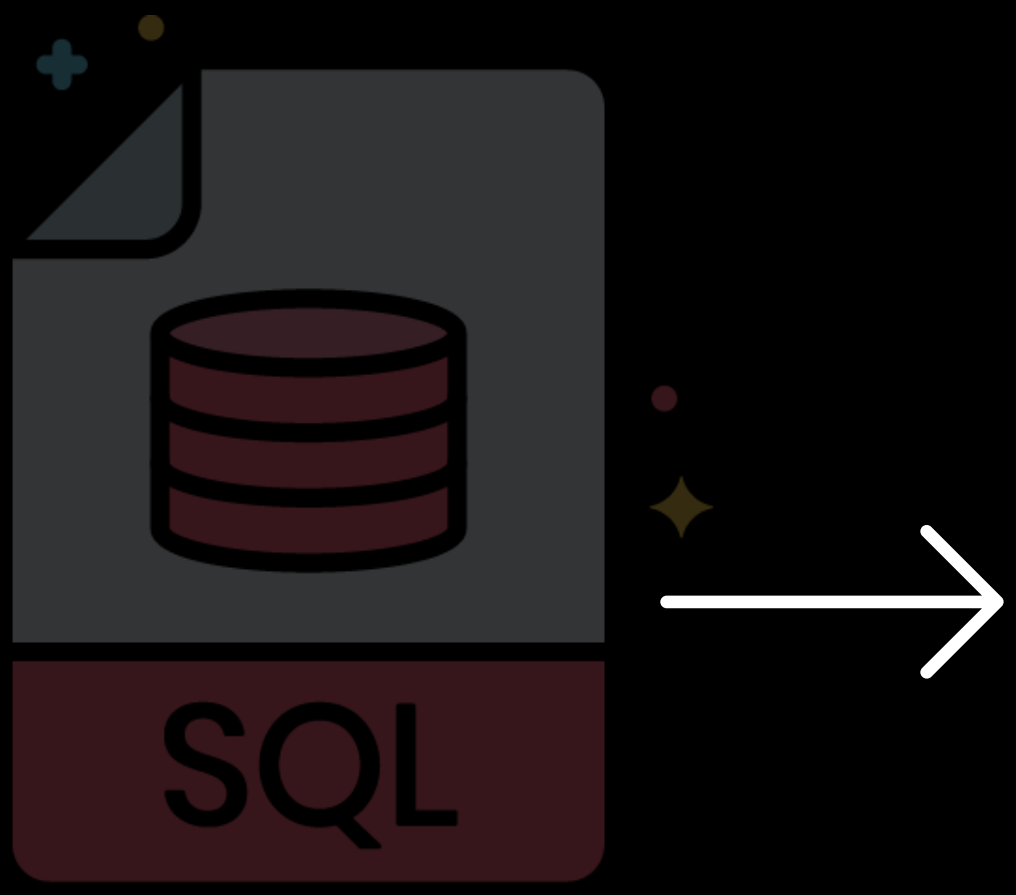
Data Engineering 101: Data Cleaning by SQL

Standardize date/time values

```
UPDATE table_name
SET
date_column = DATE_TRUNC('day', date_column);
```

Truncates date/time values to remove time information, standardizing to midnight (PostgreSQL syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



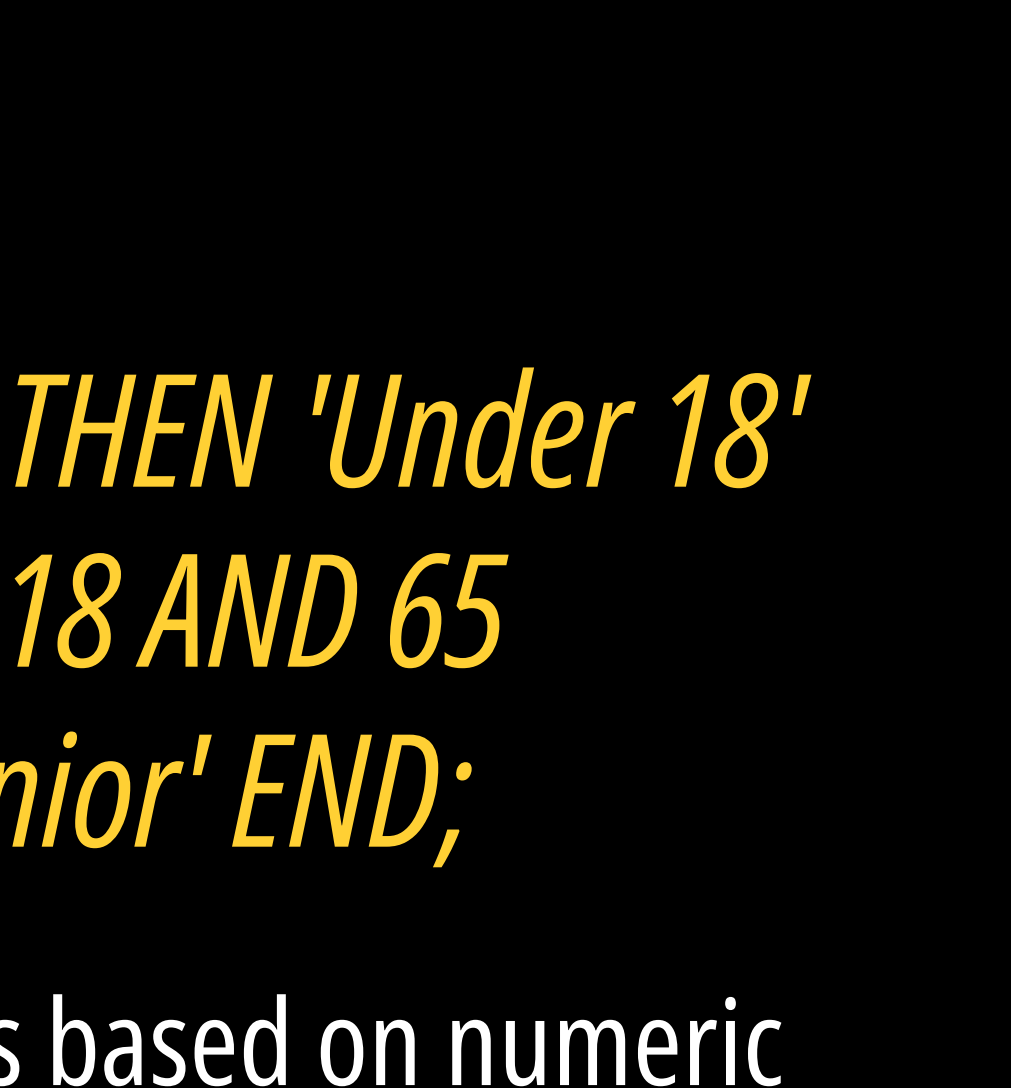
Data Engineering 101: Data Cleaning by SQL

Convert numeric to categorical

```
UPDATE table_name
SET age_group =
CASE WHEN age < 18 THEN 'Under 18'
WHEN age BETWEEN 18 AND 65
THEN 'Adult' ELSE 'Senior' END;
```

Creates categorical groups based on numeric values using a CASE statement.

Shwetank Singh
GritSetGrow - GSGLearn.com



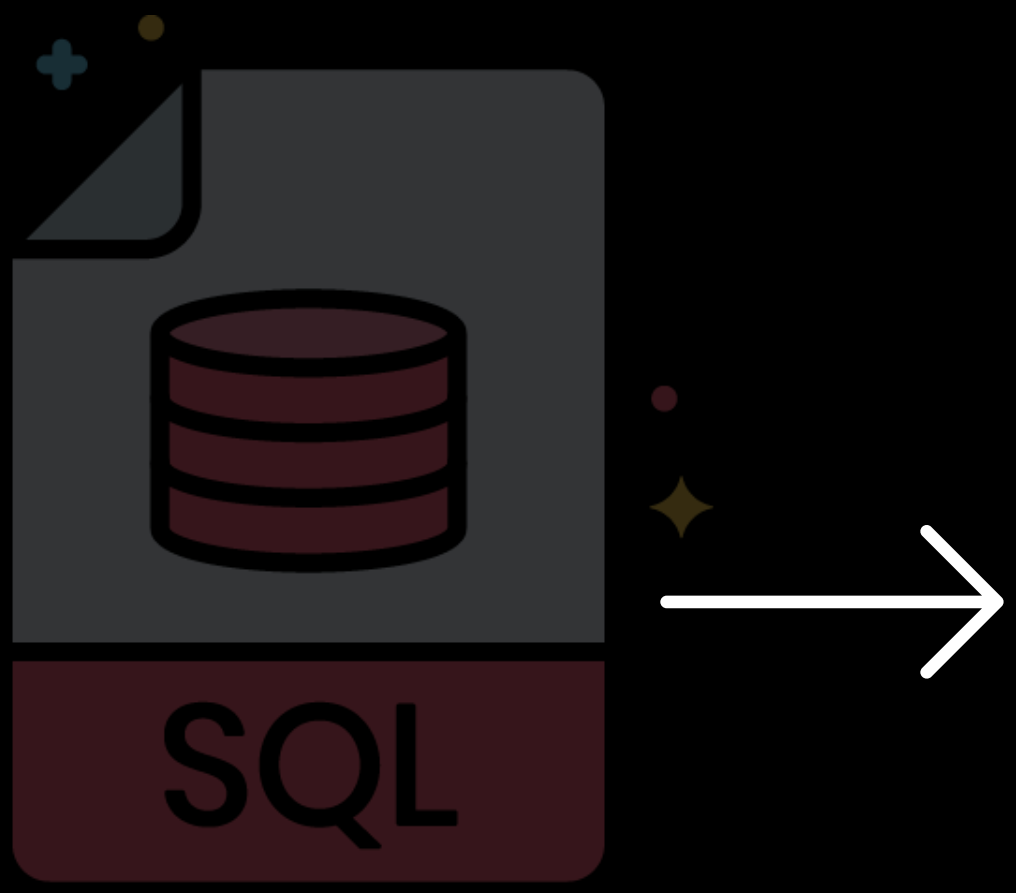
Data Engineering 101: Data Cleaning by SQL

Remove rows with future dates

```
DELETE FROM table_name
WHERE date_column > CURRENT_DATE;
```

Deletes rows where the date is in the future, assuming these are data entry errors.

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Standardize currency values

```
UPDATE table_name
SET price = price * 100
WHERE currency = 'cents';
```

Converts prices in cents to dollars for consistency across the table.

Shwetank Singh
GritSetGrow - GSGLearn.com

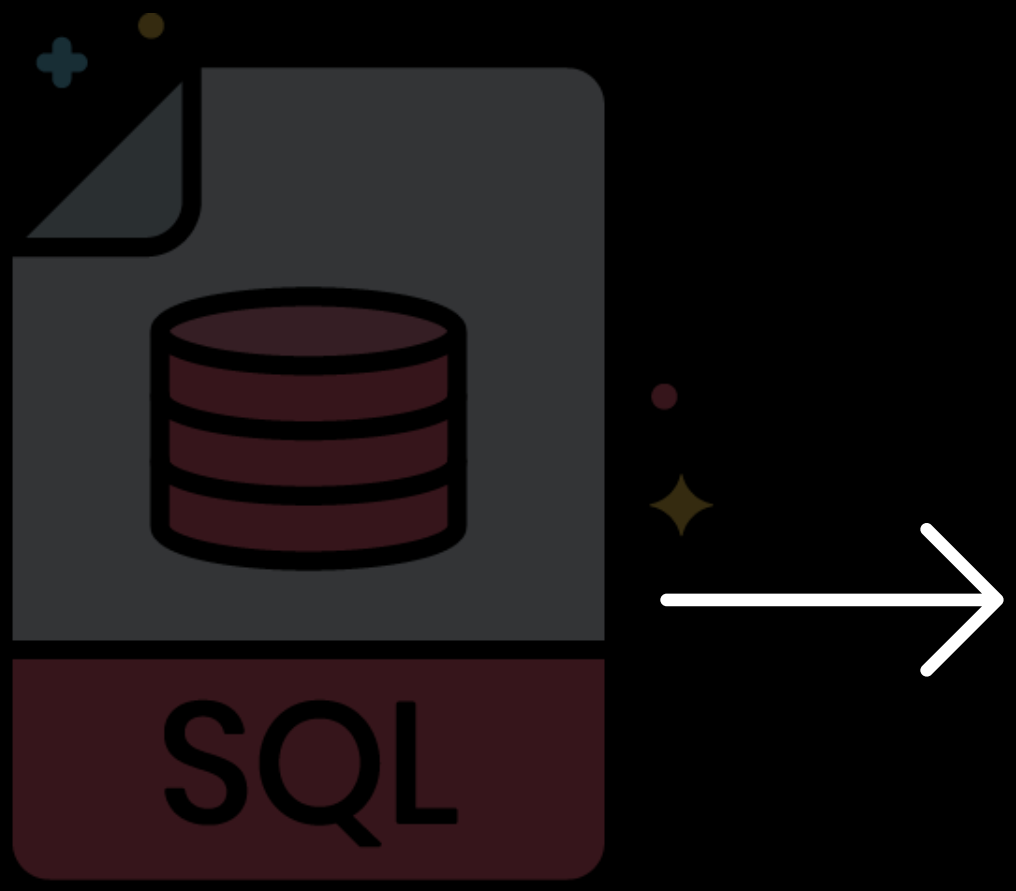
Data Engineering 101: Data Cleaning by SQL

Handle missing foreign keys

```
DELETE FROM child_table
WHERE parent_id
NOT IN (SELECT id FROM parent_table);
```

Removes rows from a child table that reference non-existent parent records.

Shwetank Singh
GritSetGrow - GSGLearn.com



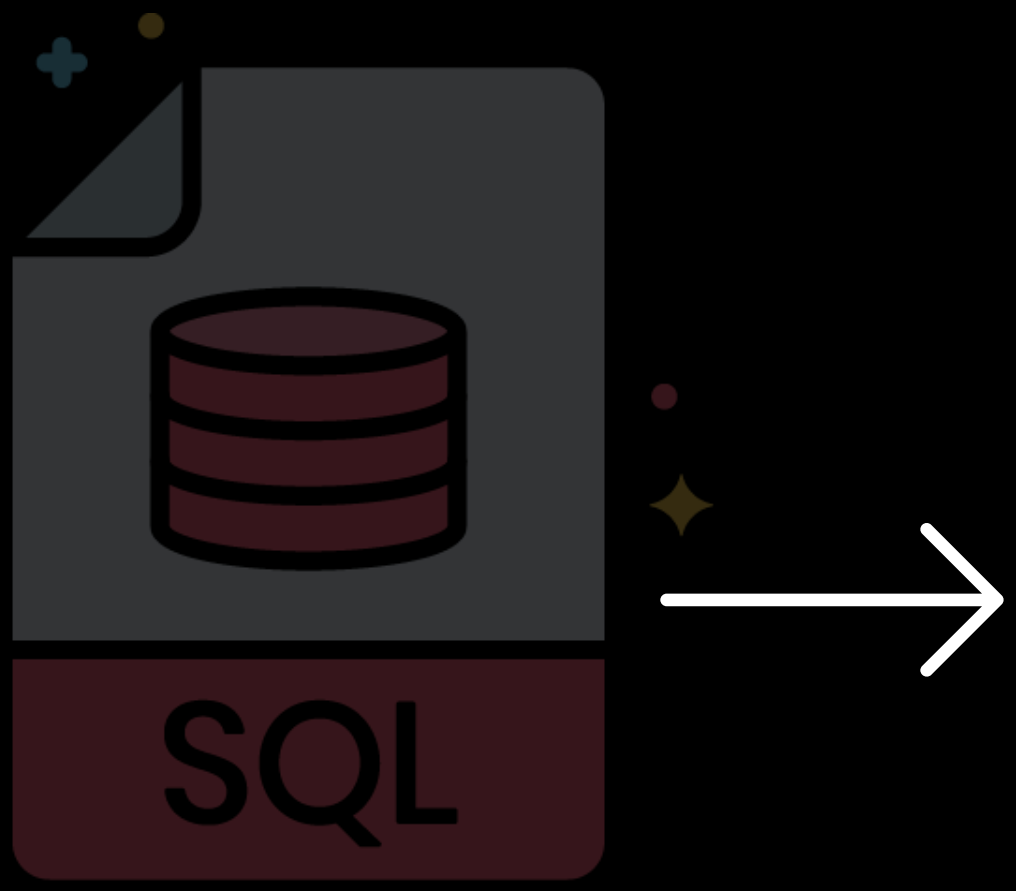
Data Engineering 101: Data Cleaning by SQL

Correct data type mismatch

```
UPDATE table_name
SET numeric_column =
CAST(text_column AS DECIMAL(10,2))
WHERE ISNUMERIC(text_column) = 1;
```

Converts text values to numeric type where possible (SQL Server syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



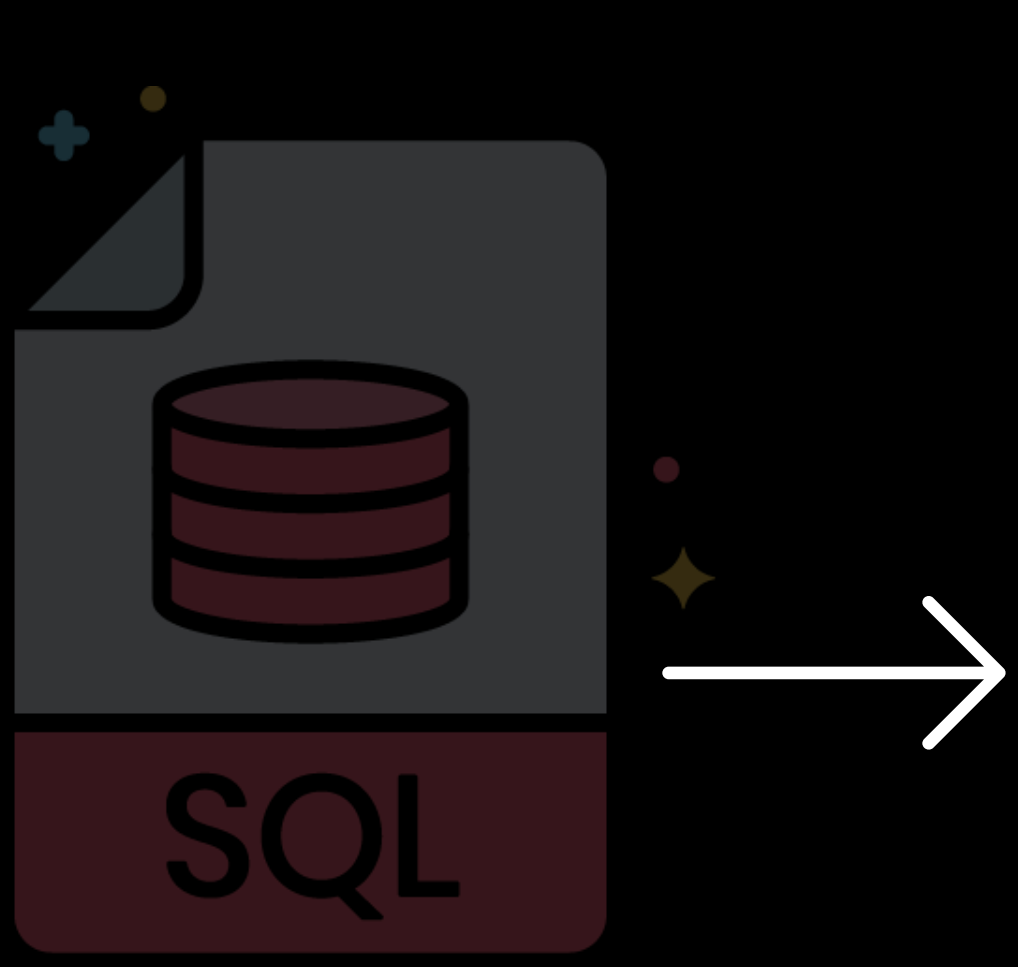
Data Engineering 101: Data Cleaning by SQL

Remove control characters

```
UPDATE table_name
SET text_column =
REGEXP_REPLACE(text_column, '[\x00-\x1F\x7F]', '');
```

Removes ASCII control characters from text using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com

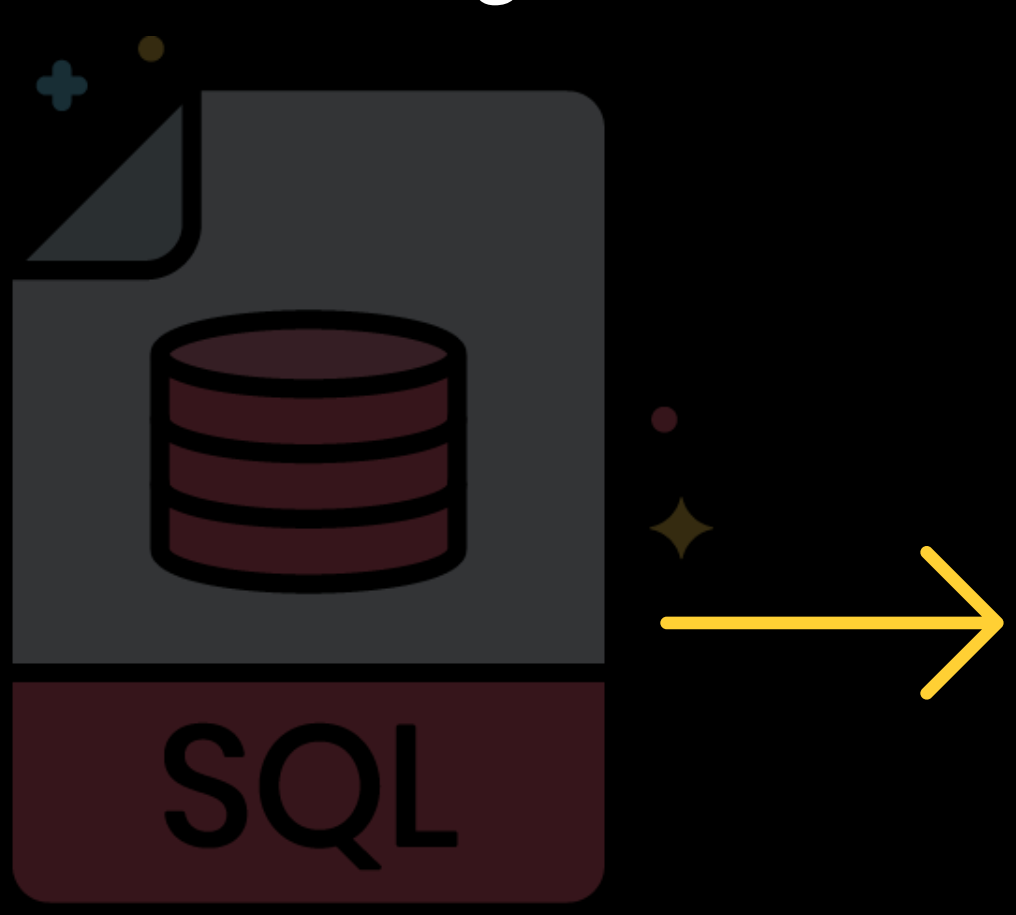


Data Engineering 101: Data Cleaning by SQL

Standardize units of measurement

```
UPDATE table_name
SET weight =
CASE WHEN unit = 'lbs'
THEN weight * 0.453592
ELSE weight END, unit = 'kg';
```

Converts weights from pounds to kilograms and updates the unit column.



Shwetank Singh
GritSetGrow - GSGLearn.com

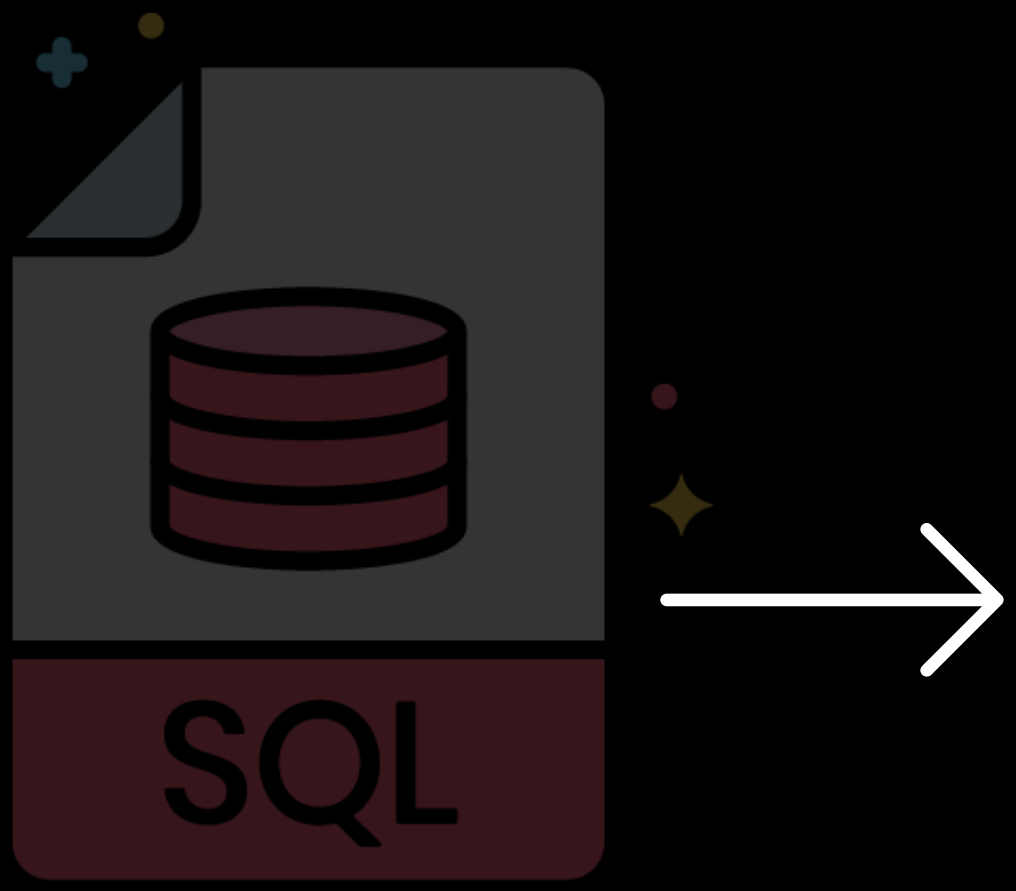
Data Engineering 101: Data Cleaning by SQL

Remove duplicate spaces

```
UPDATE table_name
SET text_column =
REGEXP_REPLACE(text_column, '\s+', ' ');
```

Replaces multiple spaces with a single space using REGEXP_REPLACE.

Shwetank Singh
GritSetGrow - GSGLearn.com



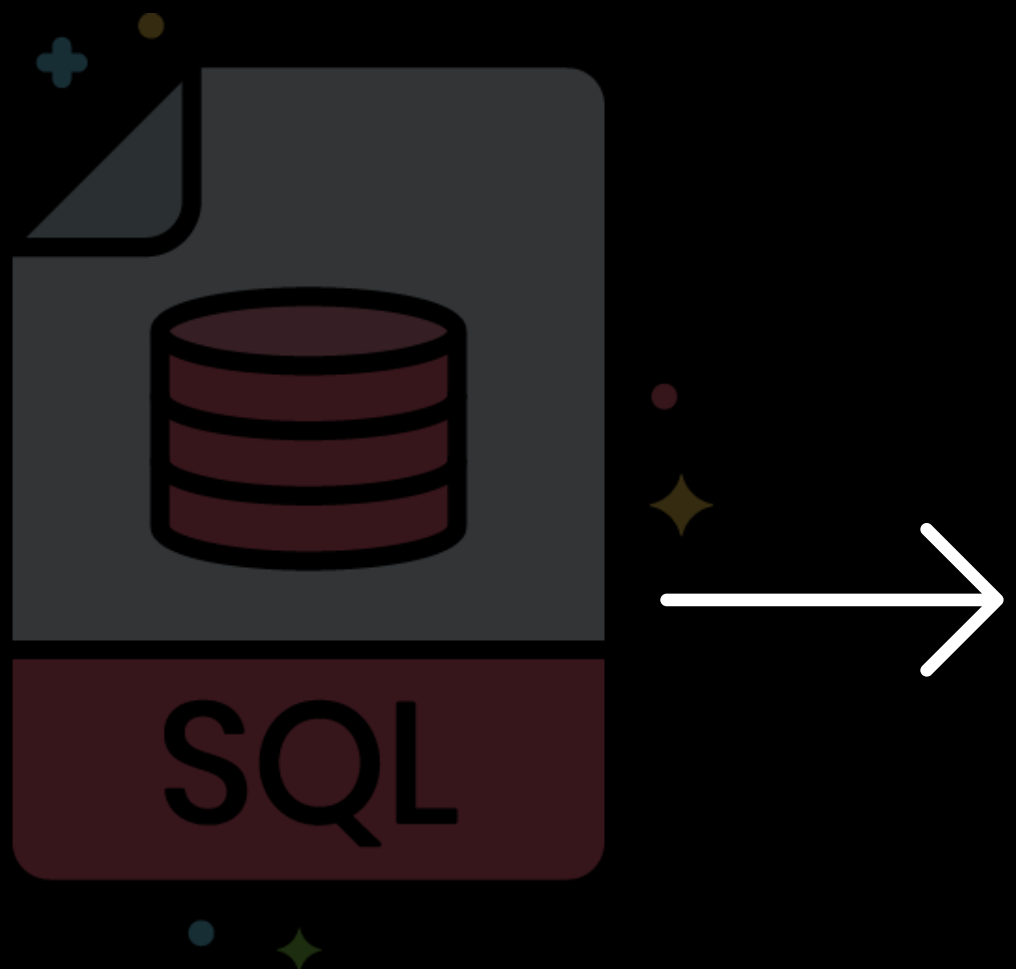
Data Engineering 101: Data Cleaning by SQL

Standardize phone number format

```
UPDATE table_name
SET phone = CONCAT('(', SUBSTRING(phone, 1,
3), ') ', SUBSTRING(phone, 4, 3), '-',
SUBSTRING(phone, 7, 4))
WHERE LENGTH(phone) = 10;
```

Formats 10-digit phone numbers to (XXX) XXX-XXXX format.

Shwetank Singh
GritSetGrow - GSGLearn.com



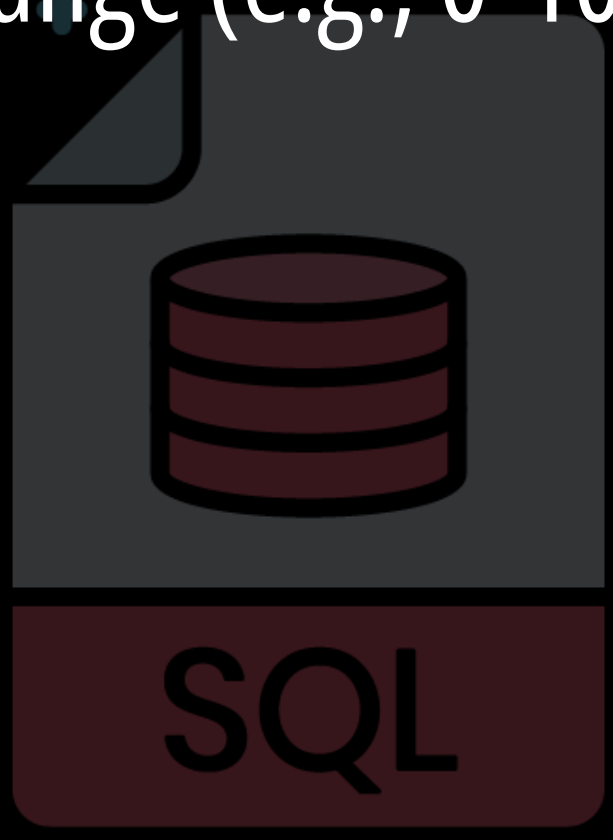
Data Engineering 101: Data Cleaning by SQL

Handle out-of-range values

```
UPDATE table_name
SET score = CASE
WHEN score < 0
THEN 0
WHEN score > 100
THEN 100 ELSE score END;
```

Clamps values to a valid range (e.g., 0-100 for a percentage).

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Standardize gender values

```
UPDATE table_name
SET gender = CASE WHEN LOWER(gender)
IN ('m', 'male')
THEN 'M' WHEN LOWER(gender)
IN ('f', 'female') THEN 'F' ELSE 'Other' END;
```

Standardizes various gender inputs to consistent values.

Shwetank Singh
GritSetGrow - GSGLearn.com

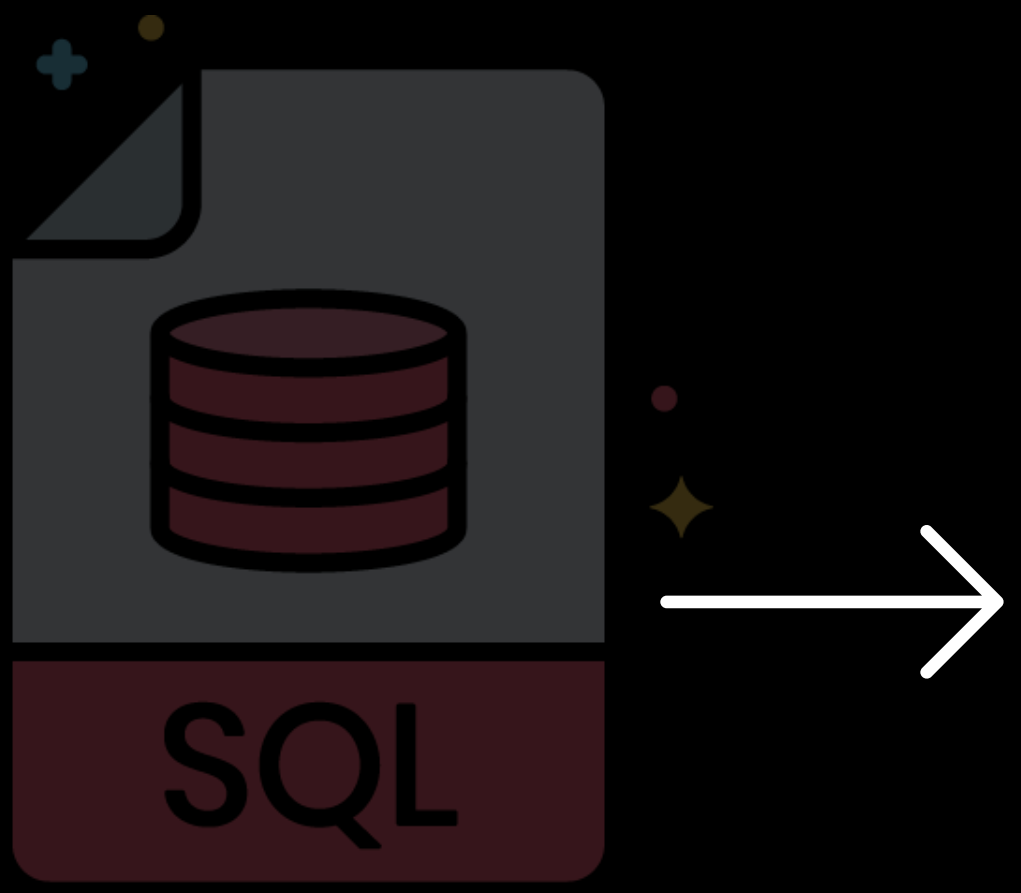
Data Engineering 101: Data Cleaning by SQL

Remove rows with impossible combinations

```
DELETE FROM sales
WHERE sale_date < hire_date;
```

Deletes records where logical rules are violated (e.g., sale date before hire date).

Shwetank Singh
GritSetGrow - GSGLearn.com



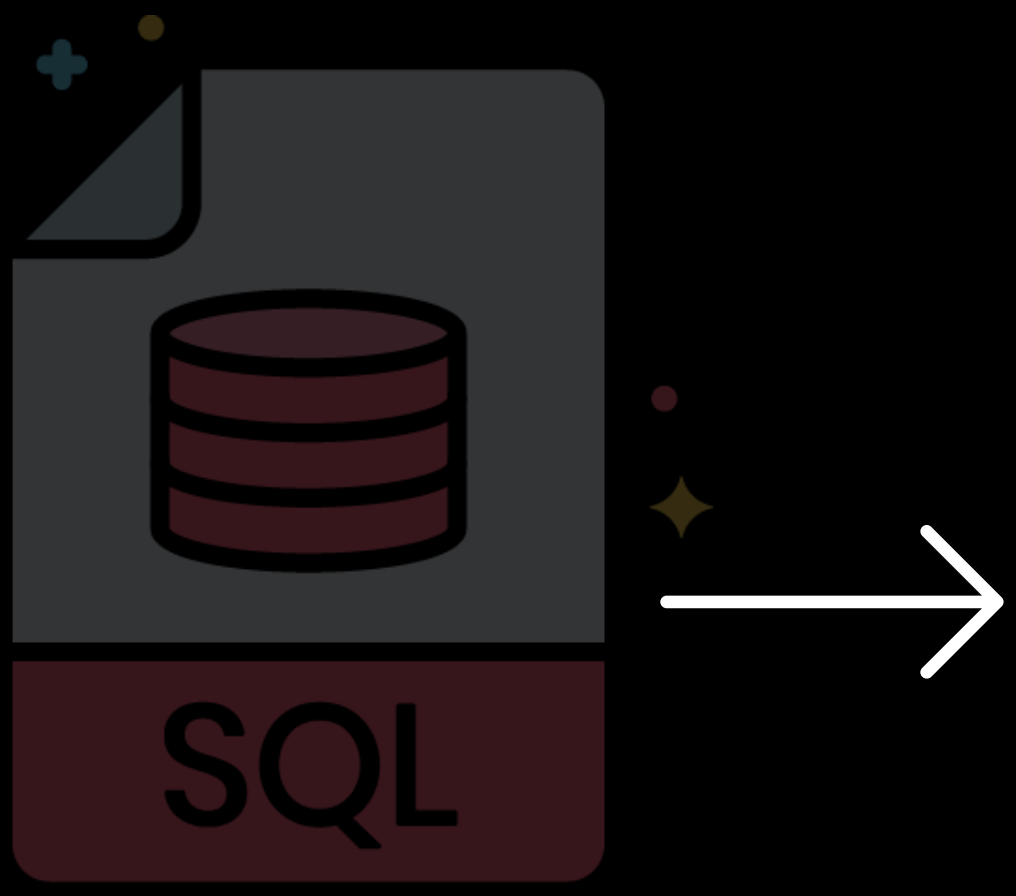
Data Engineering 101: Data Cleaning by SQL

Handle inconsistent NULL representations

```
UPDATE table_name
SET column_name = NULL
WHERE column_name
IN ('N/A', 'NULL', 'None');
```

Replaces various text representations of NULL with actual NULL values.

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Correct data entry errors in dates

```
UPDATE table_name
SET date_column = DATE_ADD(date_column,
INTERVAL 100 YEAR)
WHERE YEAR(date_column) < 1950;
```

Adds 100 years to dates likely entered without the correct century.

Shwetank Singh
GritSetGrow - GSGLearn.com

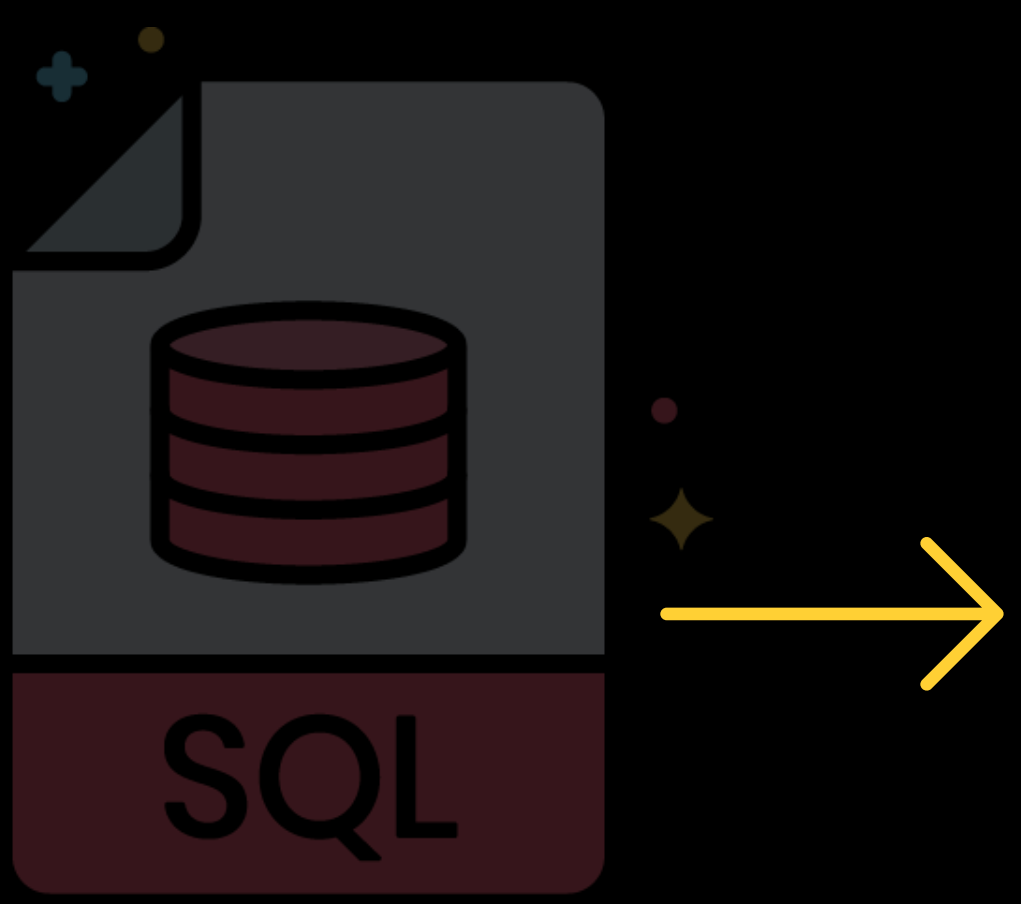
Data Engineering 101: Data Cleaning by SQL

Standardize country codes

```
UPDATE table_name
SET country_code = CASE
WHEN country = 'United States'
THEN 'USA'
WHEN country = 'United Kingdom'
THEN 'GBR' ... END;
```

Maps full country names to standard ISO country codes.

Shwetank Singh
GritSetGrow - GSGLearn.com



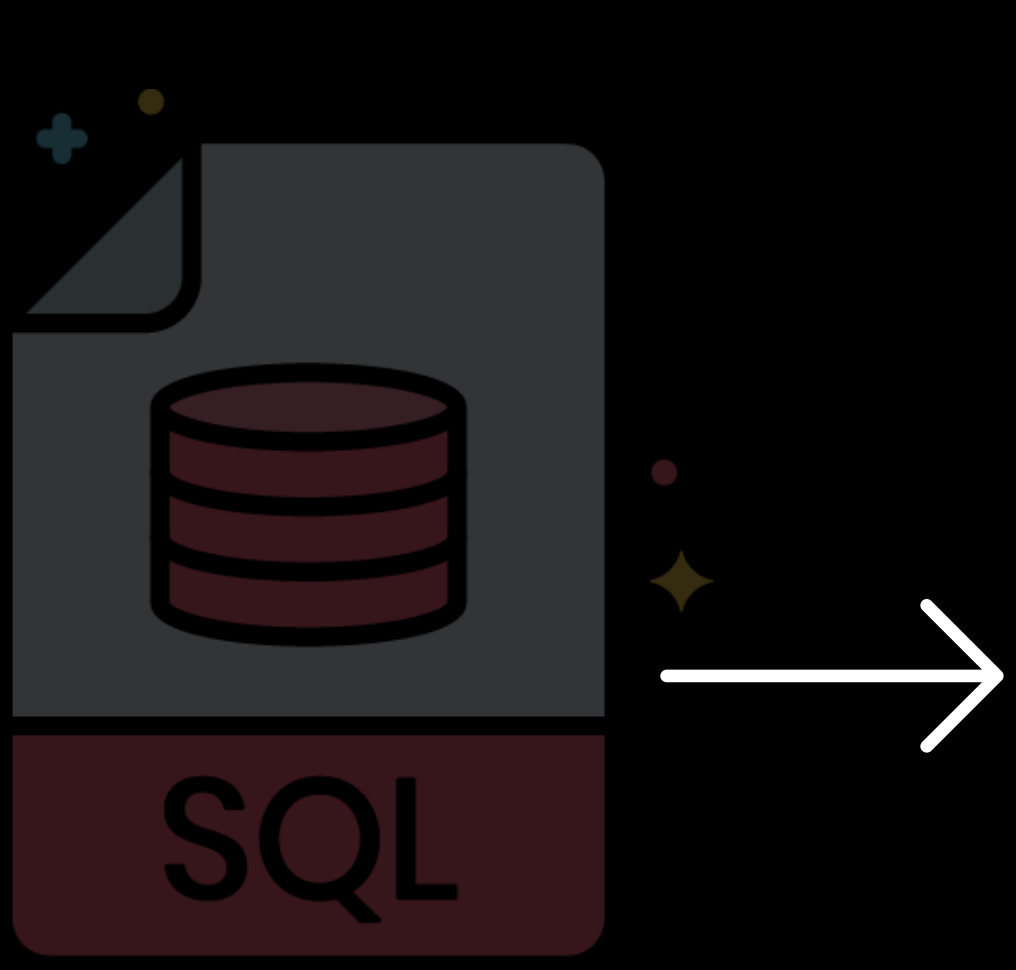
Data Engineering 101: Data Cleaning by SQL

Remove time component from date

```
UPDATE table_name
SET date_only = DATE(datetime_column);
```

Extracts only the date part from a datetime column.

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Handle inconsistent decimal separators

```
UPDATE table_name
SET numeric_column =
REPLACE(REPLACE(numeric_column, '.', ''), ',', '.')
WHERE numeric_column LIKE '%,%';
```

Converts comma decimal separators to periods for consistency.

Shwetank Singh
GritSetGrow - GSGLearn.com

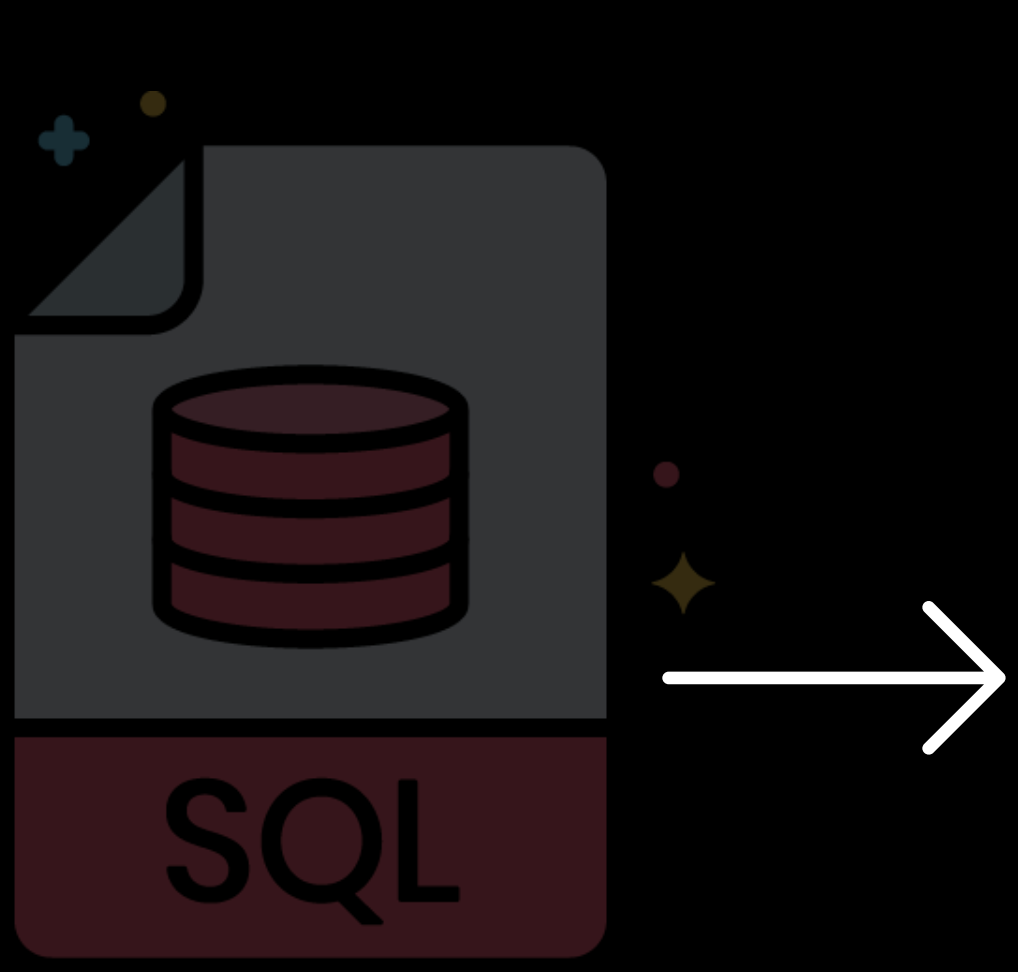
Data Engineering 101: Data Cleaning by SQL

Standardize product codes

```
UPDATE table_name
SET product_code = LPAD(product_code, 8, '0');
```

Pads product codes with leading zeros to ensure consistent length.

Shwetank Singh
GritSetGrow - GSGLearn.com



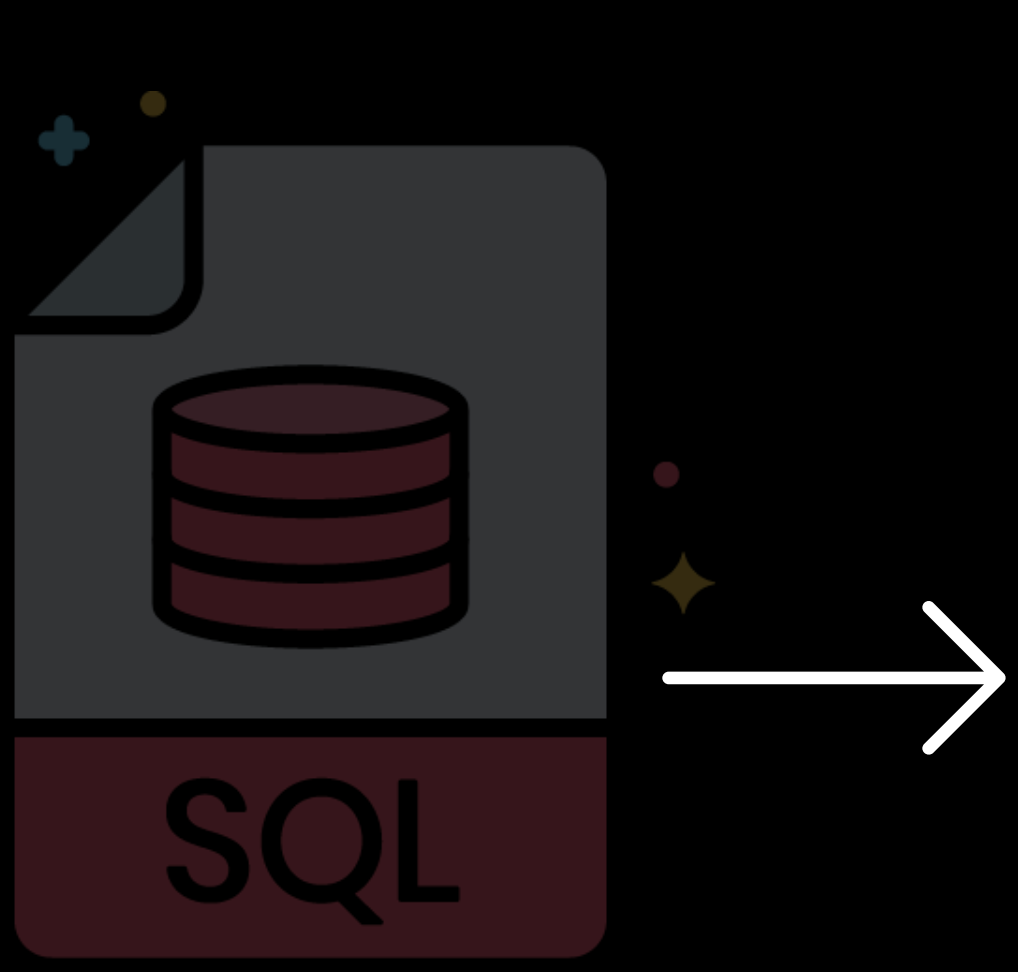
Data Engineering 101: Data Cleaning by SQL

Remove soft hyphens

```
UPDATE table_name
SET
text_column = REPLACE(text_column, CHR(173), '');
```

Removes soft hyphen characters (often invisible) from text.

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Handle inconsistent true/false values

```
UPDATE table_name
SET bool_column = CASE
WHEN bool_column
IN (1, '1', 'T', 'TRUE', 'Y', 'YES')
THEN TRUE ELSE FALSE END;
```

Standardizes various true/false representations to boolean values.

Shwetank Singh
GritSetGrow - GSGLearn.com

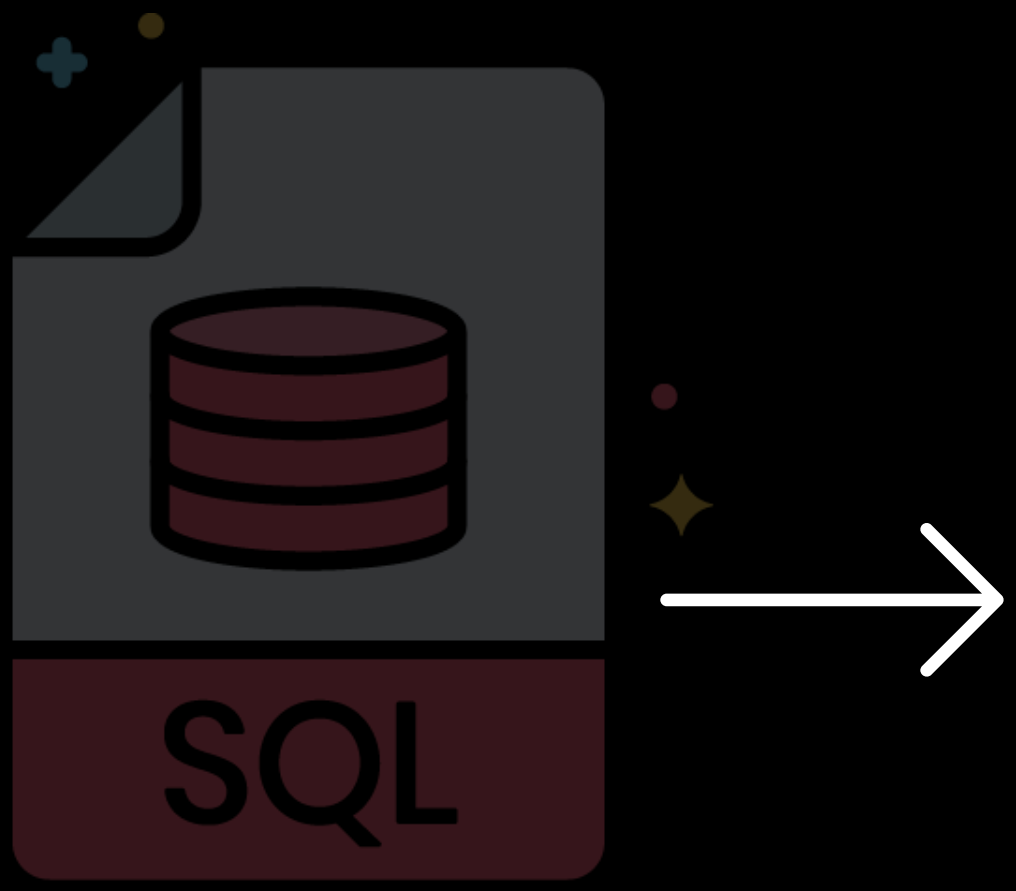
Data Engineering 101: Data Cleaning by SQL

Correct common typos

```
UPDATE table_name
SET city = CASE
WHEN city = 'New Yrok' THEN 'New York'
WHEN city = 'Chicagp' THEN 'Chicago' ...
END;
```

Corrects common typos in city names.

Shwetank Singh
GritSetGrow - GSGLearn.com



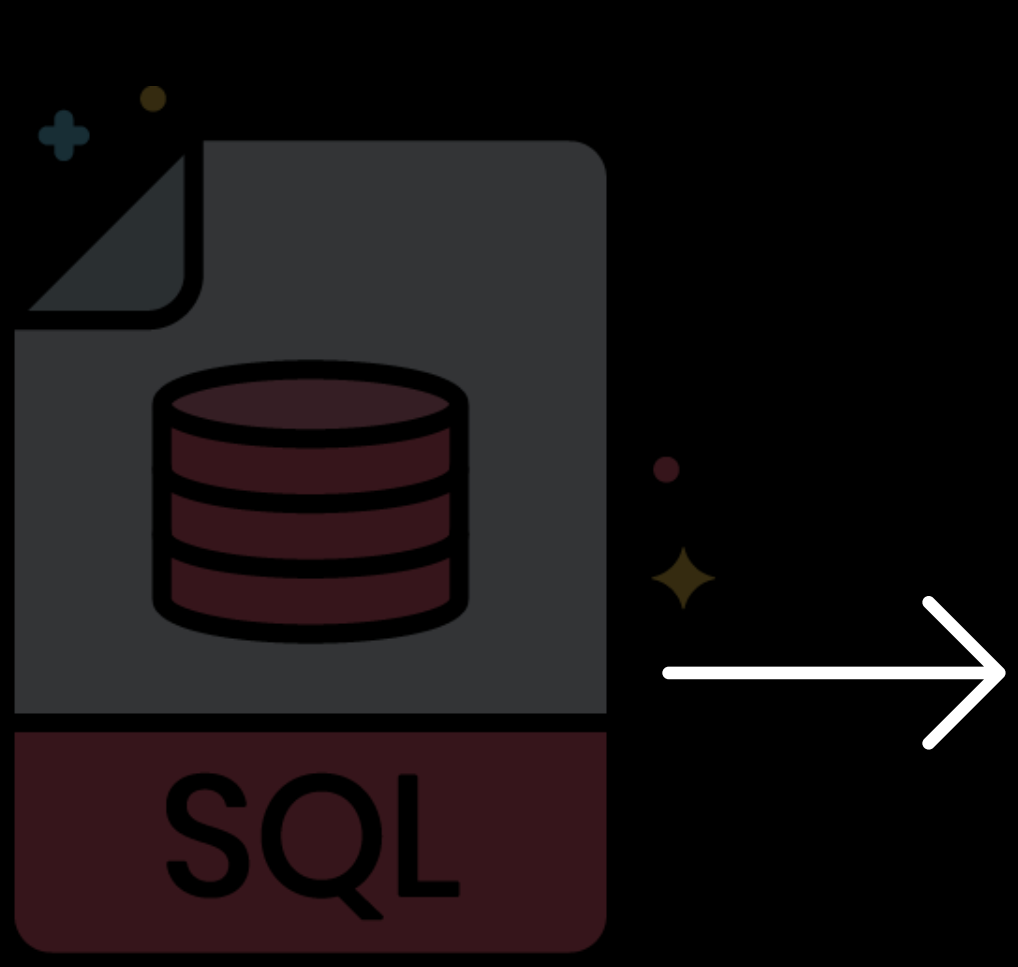
Data Engineering 101: Data Cleaning by SQL

Remove duplicate words

```
UPDATE table_name
SET
description = REGEXP_REPLACE(description,
'(\w+)(\s+\1)+', '\1', 'g');
```

Removes immediately repeated words in text (PostgreSQL syntax).

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Standardize ZIP codes

```
UPDATE table_name
SET
zip_code = LPAD(zip_code, 5, '0')
WHERE LENGTH(zip_code) < 5;
```

Ensures all ZIP codes are 5 digits, padding with leading zeros if necessary.

Shwetank Singh
GritSetGrow - GSGLearn.com

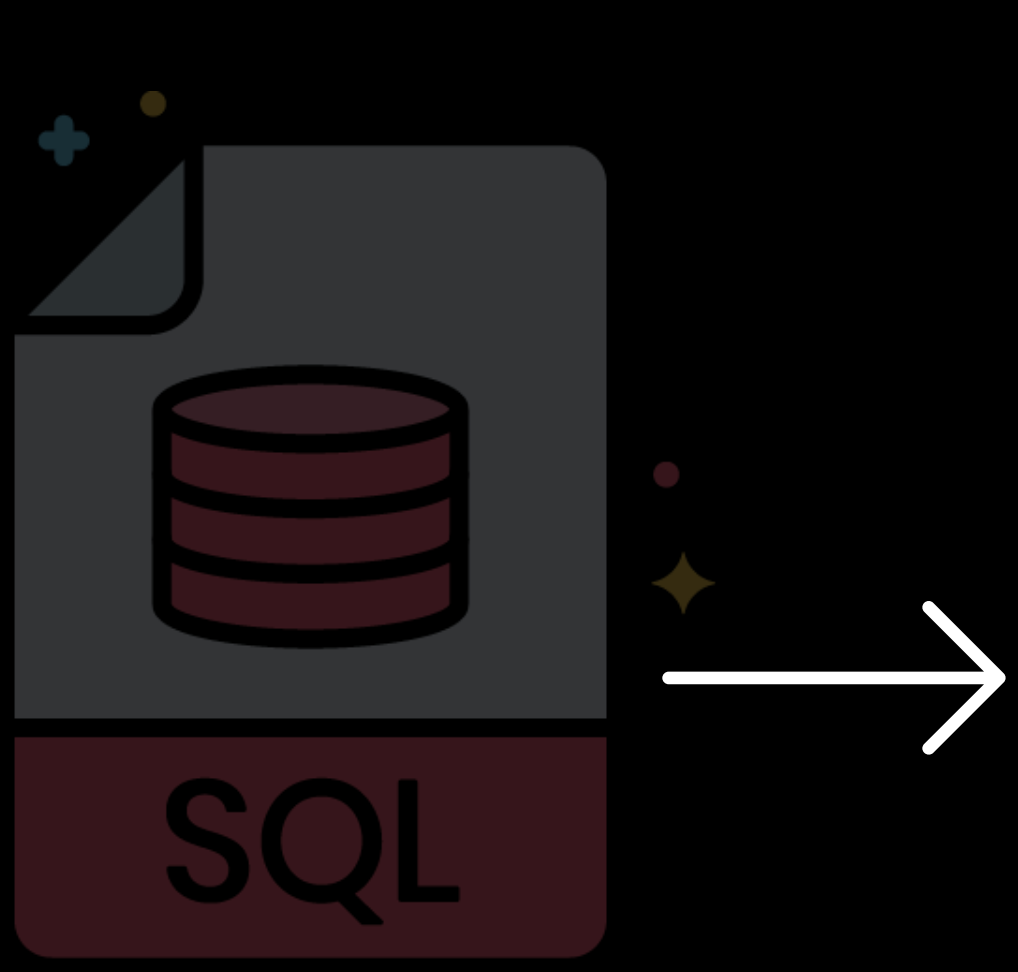
Data Engineering 101: Data Cleaning by SQL

Handle inconsistent date separators

```
UPDATE table_name
SET date_column =
REPLACE(REPLACE(date_column, '/', '-'), '.', '-');
```

Replaces various date separators with a standard separator (-).

Shwetank Singh
GritSetGrow - GSGLearn.com



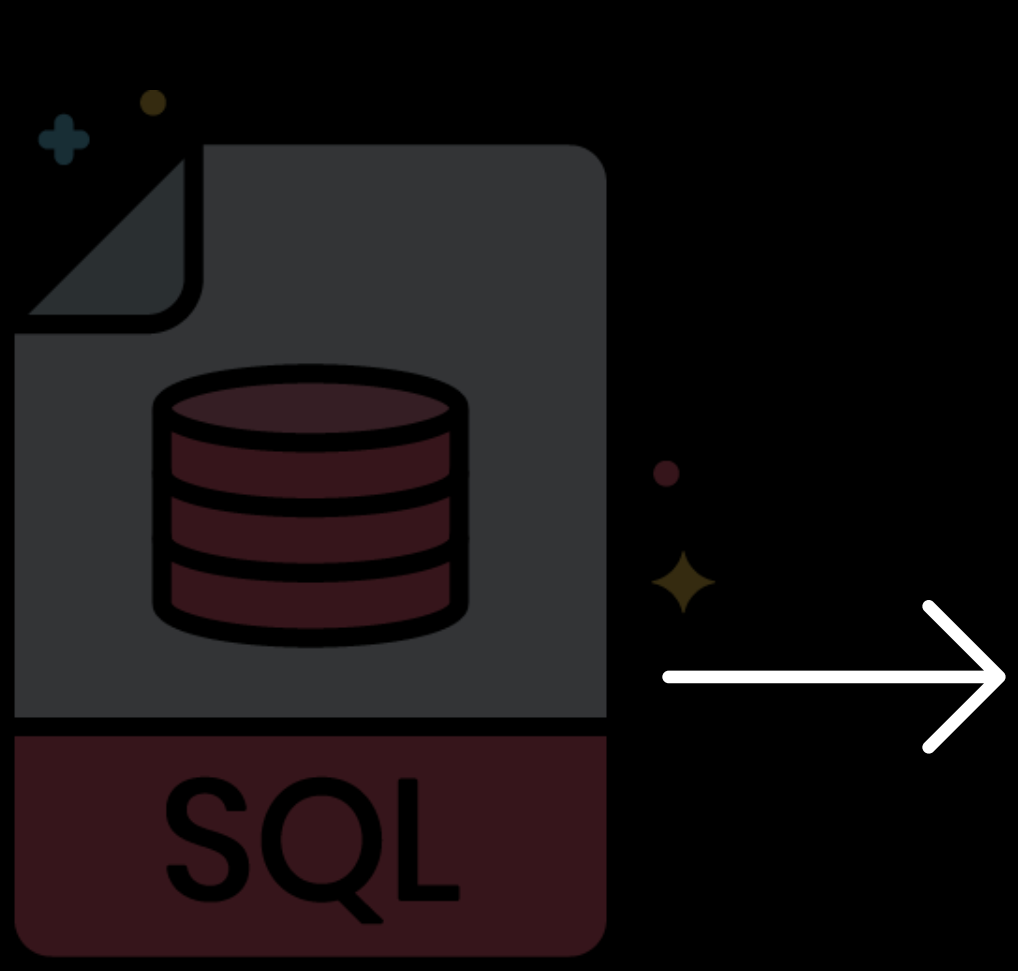
Data Engineering 101: Data Cleaning by SQL

Remove rows with suspicious patterns

```
DELETE FROM table_name
WHERE email LIKE '%@%.%;
```

Removes rows where the email contains multiple @ symbols or semicolons.

Shwetank Singh
GritSetGrow - GSGLearn.com



Data Engineering 101: Data Cleaning by SQL

Standardize numeric scale

```
UPDATE table_name
SET
temperature = temperature * 1.8 + 32,
unit = 'F'
WHERE unit = 'C';
```

Converts temperatures from Celsius to Fahrenheit and updates the unit.

Shwetank Singh
GritSetGrow - GSGLearn.com

Thank you