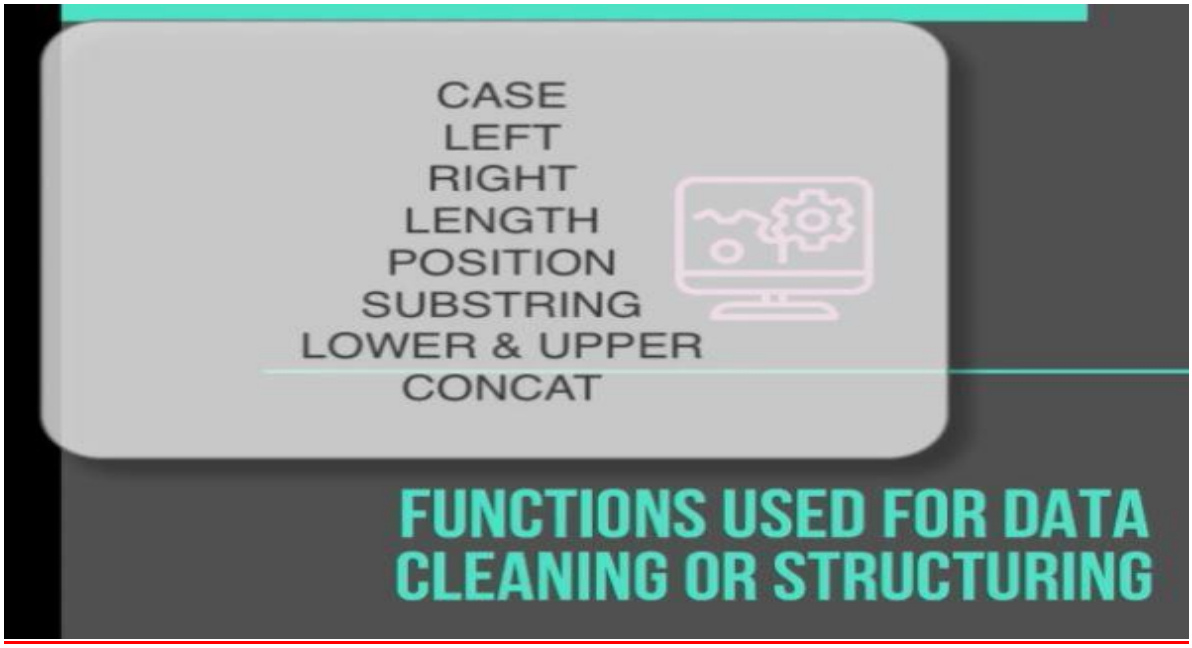
DATA CLEANING AND STRUCTURING



1. CASE Statement

- The **CASE statement** is SQL's way of handling if/then logic.
- The **CASE statement** is followed by at least one pair of **WHEN and THEN** statements- SQL's equivalent of IF/THEN in Excel.
- Every **CASE statement** must end with the **END** statement.
- **ELSE** statement is optional, and captures values not specified in the **WHEN/THEN** statements.

General Form:

```
CASE WHEN (condition) THEN (result if condition is met)
     ELSE (result if previous condition not met)
     END AS new_attribute_name
```

Example 1:

Query Editor Query History

```sql
/* Create a new column that categorizes customers based on the total
   amount they have paid: BRONZE - $0-$70,BRONZE - $0-$70 AND
   GOLD - greater than $150. Call this "customer_category" and sort
   the results by the total amount paid. */

SELECT customer_id, SUM(amount),
       CASE WHEN SUM(amount) BETWEEN 0 AND 70 THEN 'BROZEN'
            WHEN SUM(amount) BETWEEN 71 AND 150 THEN 'SILVER'
            ELSE 'GOLD'
       END AS customer_category
FROM payment
GROUP BY 1
ORDER BY 2;
```

Example 2:

Query Editor Query History

```
/* Count the inventory of each film and then categorizes each film as
   'high' inventory if it's inventory is greater than or equal to 5
    'low' inventory otherwise */

SELECT film_id, COUNT(inventory_id),
       CASE WHEN COUNT(inventory_id) >= 5 THEN 'HIGH'
            ELSE 'LOW'
       END AS inventory_status
FROM inventory
GROUP BY 1;
```

Data Output Explain Messages Notifications

	film_id smallint	count bigint	inventory_status text
1	652	4	LOW
2	273	7	HIGH

Example 3:

Query Editor Query History

```
/* Count the number of rentals for each film and categorize films
   based on number of rentals: 'low' (less than 10 rentals). 'medium'
   (between 10 to 20 rentals) and 'high' (greater than 20 rentals). */

SELECT f.film_id, title, COUNT(rental_id),
       CASE WHEN COUNT(rental_id) < 10 THEN 'LOW'
            WHEN COUNT(rental_id) BETWEEN 10 AND 20 THEN 'MEDIUM'
            ELSE 'HIGH'
       END AS demand_status
FROM film f
JOIN inventory i
ON f.film_id=i.film_id
JOIN rental r
ON i.inventory_id=r.inventory_id
GROUP BY 1;
```

Data Output Explain Messages Notifications

	film_id [PK] integer	title character varying (255)	count bigint	demand_status text
1	652	Pajama Jawbreaker	14	MEDIUM
2	273	Effect Gladiator	25	HIGH
3	51	Balloon Homeward	23	HIGH

2. LEFT, RIGHT & LENGTH Statement

- **LEFT** pulls a specified number of characters for each row in a column starting at the beginning (or from the left). **LEFT (string, number)**
- **RIGHT** pulls a specified number of characters for each row in a column starting at the end (or from the right). **RIGHT (string, number)**
- **LENGTH** provides the number of characters for each row of a specified column. **LENGTH (string)**

Example 1:

Query Editor Query History

```
/* Extract each customer's local phone number including the area code
   (the last 10 digits in the given phone number),e.g. 14033335568
   Hint : Find the phone attribute in address table. */

SELECT RIGHT(phone,10) AS local_phone_number
FROM address;

-- Result was changed from 14033335568 to 4033335568 in no.3
```

Data Output Explain Messages Notifications

	local_phone_number text
1	
2	
3	4033335568
4	6172235589
5	8303384290

Example 2:

Query Editor Query History

```
/* Extract each customer's country code from the given phone number.
   Hint : The country code is either 1-digit or 2-digits If a phone
   number does not include a country code, then write 'NULL'. */

SELECT phone, CASE WHEN LENGTH(phone) = 11 THEN LEFT(phone, 1)
            WHEN LENGTH(phone) = 12 THEN LEFT(phone, 2)
            ELSE 'NULL'
        END AS country_code
FROM address;
```

Data Output Explain Messages Notifications

	phone character varying (20)	country_code text
1		NULL
2		NULL
3	14033335568	1
4	6172235589	NULL
5	28303384290	2
6	838635286649	83

Example 3:

Query Editor Query History

```
/* How many customers have first names starting with a vowel(a,e,i,o,u)
   */
SELECT SUM(firstname_vowel)
FROM (SELECT customer_id,
        CASE WHEN LEFT(first_name, 1) IN ('A','E','I','O','U') THEN 1
            ELSE 0
         END AS firstname_vowel
      FROM customer)sub1;

--Since the values under firstname_vowel are either 1 or 0, so summed up.
```

Data Output Explain Messages Notifications

	sum bigint
1	83

3. POSITION, SUBSTRING, UPPER & LOWER Statement

- **POSITION** provides the position of a string starting from the left. In other words, it provides an index where the character is for each row. Note that the index of the first position is 1 in SQL.

  General form: POSITION ('string' IN attribute)

- Since the **POSITION** is case-sensitive, you may use either **LOWER or UPPER** to make all the characters in either **lowercase or uppercase**.

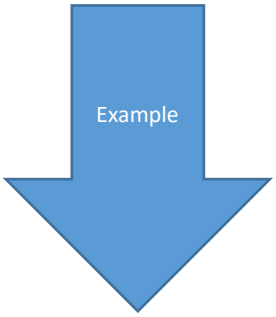
  General form: LOWER (attribute)
  UPPER (attribute)

- **SUBSTRING** is used to extract a string containing a specific number of characters from a particular position of a given string.

  General form: SUBSTRING (attribute, str_pos, ext_char)

  Where,

  - str_pos – (optional) this is the position of the string where extracting starts. If omitted, the substring function will start at positon 1.
  - ext_char – (optional) specifies the number of characters to be extracted from the starting position. If omitted, the substring function will extract until the last character of the string.



Example 1: The email attribute under the customer table is formatted as:

firstname.lastname@sakilacustomer.org

Create a new column showing the firstname.lastname only and call this new column email_name.

```
SELECT LEFT(email, POSITION('@' IN email))
FROM customer;
```

Data Output | Explain | Messages | Notifications

	left text
1	jared.ely@
2	mary.smith@

OR

```
SELECT LEFT(email, POSITION('@' IN email)-1)
FROM customer;
```

Data Output | Explain | Messages | Notifications

	left text
1	jared.ely
2	mary.smith

Example 2: The email attribute under the customer table is formatted as:

firstname.lastname@sakilacustomer.org

Create a new column showing the last name only and call this new column email_lastname.

```
SELECT SUBSTRING(email, POSITION('.' IN email)+1,
                POSITION('@' IN email) - POSITION('.' IN email)-1)
                AS email_lastname
FROM customer;
```

Output | Explain | Messages | Notifications

email_lastname text
ely
smith

Example 3: The email attribute under the customer table is formatted as:

firstname.lastname@sakilacustomer.org

Create a new column showing the first name only and call this new column email_firstname.

```
SELECT LEFT(email, POSITION('.' IN email)-1)
                AS email_firstname
FROM customer;
```

Data Output | Explain | Messages | Notifications

	email_firstname text
1	jared
2	mary

Example 4: The film's title attribute under the film table is formatted as:

"Chamber Italian"

Grab the second word in the film's title (in lowercase) and call this new attribute film_phrase.

```
SELECT RIGHT(lower(title), LENGTH(title)-POSITION(' ' IN title))
        AS titile_phrase
FROM film;
```

Data Output | Explain | Messages | Notifications

	titile_phrase text
1	italian
2	wonderful

4. CONCAT Statement

CONCATE combines values from several columns into one column.

General form: CONCATE (column1, column2)

Note: You may also insert a character or space in between the columns.

Instead of CONCAT, you may use || (pipes), e.g. **column1 ||column2**

Example 1: Combine the first_name and last_name columns into one column full_name. Make sure to put a space between the first name and last name.

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

Data Output | Explain | Messages | Notifications

	full_name text
1	Jared Ely
2	Mary Smith

Example 2: Create a new version of email addresses (name it email_address). It will be of the following format: f.lastname@sakila.org
'f' stands for the first letter of the first name. All characters should be in lowercase.

```
SELECT CONCAT(LOWER(LEFT(first_name,1)),'.',LOWER(last_name),'@sakila.org')
        AS email_address
FROM customer;
```

Data Output Explain Messages Notifications

	email_address text
1	j.ely@sakila.org
2	m.smith@sakila.org

Example 3: Create a column showing the customer's district and country. Make sure to put a comma and a space in between.

For example: Alberta, Canada

```
SELECT district, country, CONCAT(district,',',' ',country)
        AS dist_country
FROM address a
JOIN city ci
ON a.city_id = ci.city_id
JOIN country co
ON ci.country_id = co.country_id;
```

Data Output Explain Messages Notifications

	district character varying (20)	country character varying (50)	dist_country text
1	Alberta	Canada	Alberta, Canada
2	QLD	Australia	QLD, Australia