```
- CONCAT():
      CONCAT(first_name, ' ', last_name) AS "Full Name"

- CONCAT_WS():
      CONCAT_WS(' ', first_name, middle_name, last_name) AS "Full Name"

- SUBSTRING():
      SUBSTRING(string, start_position, length)
      e.g.:
      SELECT SUSTRING('Mili Vanili', 1, 4)
      --> Returns 'Mili'

      SELECT SUBSTRING('Mili Vanili', 'Mili')
      --> Returns 'Mili'

      SELECT SUBSTRING("River Information" FROM '([0-9]{1,4})')

      SELECT SUBSTRING('Mili Vanili', 'Vili')
      --> Returns Null

      SELECT SUBSTRING('koko123456pepi', '[1-9]+')
      means check if RegEx '[1-9]+' in 'koko123456pepi'
      --> Returns 123456

- LEFT():
      LEFT(sting, count)
      e.g.:
      SELECT LEFT('Zaio Baio', 4)
      --> Returns 'Zaio'

- RIGHT():
      RIGHT(string, count)'e.g.:
      e.g.:
      SELECT RIGHT('Zaio Baio', 4)
      --> Returns 'Baio'

- REPLACE():
      REPLACE(string, pattern, replacement)

- UPPER():
      ELECT UPPER('mimi')
      --> Returns MIMI

- LOWER():
      SELECT LOWER('MIMI')
      --> Returns mimi

- TRIM():
      SELECT TRIM(string)
      e.g.:
      SELECT TRIM(' koko ')
      --> Returns koko without space around it'
      SELECT LTRIM...
      e.g.:
      ELECT LTRIM('+3588612351', '+359')
      SELECT RTRIM...
      SELECT TRIM(BOTH ' ' FROM ' Uni ')
      SELECT TRIM(LEADING FROM string)
```

```
      SELECT TRIM(TRAILING FROM string)

- REVERSE():
      SELECT REVERSE(string)

- REPEAT():
      SELECT REPEAT(string, count)

- LENGTH() can use also CHAR_LENGTH():
      SELECT LENGTH(string)

- BIT_LENGTH():ca
      BIT_LENGTH(string)

- TRANSLATE():
      SELECT TRANSLATE('12312', '123', 'alb');
      --> Returns 'albal'

- POSITION():
      SELECT POSITION('moia' IN 'Mila moia mamo')
      --> Returns 6, the position of the first character
-----------------------------------------------------
- FLOOR():
      SELECT FLOOR(2.68)
      --> Returns 2

- CEIL():
      SELECT CEIL(2.68)
      --> Returns 3

- ROUND():
      SELECT ROUND(33.68)
      --> Returns 34
      SELECT ROUND(33.688, 2)
      --> Returns 33.69
      SELECT ROUND(33.688, -1)
      --> Returns 30

- TRUNC():
      Truncate to n decimal places
      SELECT TRUNC(12.588)
      --> Returns 12
      SELECT TRUNC(12.588, 1)
      --> Returns 12.5

- COALESCE(argument1, argument2...):
      Accepts an unlimited number of arguments. It returns the first argument that
is not null. If all arguments are null, the COALESCE function will return null.
      !! Arguments should be of the same data type !!
      e.g.:
      COALESCE(AGE(died, born)::VARCHAR(20), CONCAT('Alive: ', AGE(now(),
born)::VARCHAR(30))) AS lifespan
-----------------------------------------------------
- EXTRACT('part' FROM date):
      extract a segment from a date as an integer
      part can be second, minute, hour, day, week, month, year
      EXTRACT('day' FROM date)

- AGE(first_date, second_date):
```

```
      find the difference between two dates

- NOW():
      obtains the current date and time, including time zone

- CURRENT_DATE:
      SELECT CURRENT_DATE:

- CURRENT_TIME:
      SELECT CURRENT_TIME:

- TO_CHAR():
      formats the date value according to the format
      SELECT TO_CHAR(NOW(), 'DD Month YYY') AS "Date";
      --> Returns 23 February 2023

- DATE_PART():
      SELECT DATE_PART('year', now())

- INTERVAL:
      SELECT now() + INTERVAL '3 hours'

-------------------------------------------------------------
- % - represents zero, one or multiple characters

- _ - represents a signle character

- LIKE operator in a WHERE clause
      SELECT *
      FROM authors
      WHERE firs_name LIKE 'Aga%'
      --> Agatha will be in the result
      or
      WHERE first_name LIKE '_gatha'

      ILIKE is case-insensitive

-------------------------------------------------------------
- CAST():
      CAST(table_name AS data_type)
      e.g.:
      CAST(population AS VARCHAR)
```