**Lab Exercises**

**Find Book Titles**

SELECT

    title

FROM

    books

WHERE

    SUBSTRING(title, 1, 3) = 'The'

ORDER BY

    id

;

**Replace Titles**

SELECT REPLACE(title, 'The', '\*\*\*')

FROM

    books

WHERE LEFT(title, 3) = 'The'

ORDER BY

    id

;

**Triangles on Bookshelves**

SELECT

    id,

    side \* height / 2 AS area

FROM

    triangles

ORDER BY

    id

;

**Format Costs**

SELECT

    title,

    TRUNC(cost, 3) AS "modified\_price"

FROM

    books

ORDER BY

    id

;

**Year of Birth**

SELECT

    first\_name,

    last\_name,

    EXTRACT(YEAR FROM born) AS "year"

FROM authors

;

**Format Date of Birth**

SELECT

    last\_name AS "Last Name",

    TO\_CHAR(born, 'DD (Dy) Mon YYYY') AS "Date of Birth"

FROM authors

;

**Harry Potter Books**

SELECT

    title

FROM

    books

WHERE

    title LIKE '%Harry Potter%'

;

**Homework Exercises**

**River Info**

CREATE VIEW view\_river\_info AS

SELECT

    CONCAT\_WS(' ', 'The river', river\_name, 'flows into the', outflow, 'and is', "length", 'kilometers long.') AS "River Information"

FROM

    rivers

ORDER BY

    river\_name

;

**Concatenate Geography Data**

CREATE VIEW view\_continents\_countries\_currencies\_details AS

SELECT

    CONCAT(TRIM(cont.continent\_name), ': ', cont.continent\_code) AS continent\_details,

    CONCAT\_WS(' - ', countr.country\_name, countr.capital, countr.area\_in\_sq\_km, 'km2') AS country\_information,

    CONCAT(curr.description, ' (', curr.currency\_code, ')') AS currencies

FROM

    countries AS countr,

    continents AS cont,

    currencies AS curr

WHERE

    cont.continent\_code = countr.continent\_code

        AND

    countr.currency\_code = curr.currency\_code

ORDER BY

    country\_information ASC,

    currencies ASC

;

**Capital Code**

ALTER TABLE

    countries

ADD COLUMN

    capital\_code CHAR (2)

;

UPDATE

    countries

SET

    capital\_code = SUBSTRING(capital, 1, 2)

;

**(Descr)iption**

SELECT

    SUBSTRING(description, 5)

FROM

    currencies

;

**Substring River Length**

SELECT

    SUBSTRING("River Information", '([0-9]{1,4})') AS river\_length

FROM

    view\_river\_info

;

**Replace A**

SELECT

    REPLACE(mountain\_range, 'a', '@') AS replace\_a,

    REPLACE(mountain\_range, 'A', '$') AS replace\_A

FROM

    mountains

;

**Character Length and Bits**

SELECT

    CONCAT(m.mountain\_range, ' ', p.peak\_name) AS mountain\_information,

    LENGTH(CONCAT(m.mountain\_range, ' ', p.peak\_name)) AS characters\_length,

    BIT\_LENGTH(CONCAT(m.mountain\_range, ' ', p.peak\_name)) AS bits\_of\_a\_tring

FROM

    mountains AS m,

    peaks AS p

WHERE

    m.id = p.mountain\_id

;

**Length of a Number**

SELECT

    population,

    LENGTH(CAST(population AS VARCHAR)) AS length

FROM

    countries

;

**Positive and Negative LEFT**

SELECT

    peak\_name,

    LEFT(peak\_name, 4) AS positive\_left,

    LEFT(peak\_name, -4) AS negative\_left

FROM

    peaks

;

SELECT

    peak\_name,

    RIGHT(peak\_name, 4) AS positive\_right,

    RIGHT(peak\_name, -4) AS negative\_right

FROM

    peaks

;

**Positive and Negative RIGHT**

SELECT

    peak\_name,

    RIGHT(peak\_name, 4) AS positive\_right,

    RIGHT(peak\_name, -4) AS negative\_right

FROM

    peaks

;

**Update iso\_code**

UPDATE

    countries

SET

    iso\_code = UPPER(LEFT(country\_name, 3))

WHERE

    iso\_code IS NULL

;

**REVERSE country\_code**

UPDATE

    countries

SET

    country\_code = LOWER(REVERSE(country\_code))

;

**Elevation --->> Peak Name**

SELECT

    CONCAT(elevation, ' ', REPEAT('-', 3), REPEAT('>', 2), ' ', peak\_name) AS "Elevation --->> Peak Name"

FROM

    peaks

WHERE

    elevation >= 4884

;

**Arithmetical Operators**

CREATE TABLE

    bookings\_calculation

AS SELECT

    booked\_for

FROM

    bookings

WHERE

    apartment\_id = 93

;

ALTER TABLE

    bookings\_calculation

ADD COLUMN

    multiplication NUMERIC,

ADD COLUMN

    modulo NUMERIC

;

UPDATE

    bookings\_calculation

SET

    multiplication = booked\_for \* 50,

    modulo = booked\_for % 50

;

**ROUND vs TRUNC**

SELECT

    latitude,

    ROUND(latitude, 2),

    TRUNC(latitude, 2)

FROM

    apartments

;

**EXTRACT Booked At**

SELECT

    EXTRACT(YEAR from booked\_at) AS "YEAR",

    EXTRACT(MONTH from booked\_at) AS "MONTH",

    EXTRACT(DAY from booked\_at) AS "DAY",

    EXTRACT(HOUR from booked\_at) AS "HOUR",

    EXTRACT(MINUTE from booked\_at) AS "MINUTE",

    CEILING(EXTRACT(SECOND from booked\_at)) AS "SECOND"

FROM

    bookings

;

**Match or Not**

SELECT

    companion\_full\_name,

    email

FROM

    users

WHERE

    companion\_full\_name ILIKE '%and%'

        AND

    email NOT LIKE '%@gmail'

;

**COUNT by Initial**

SELECT

LEFT(first\_name, 2) AS initials,

COUNT('initials') AS user\_count

FROM

users

GROUP BY

initials

ORDER BY

user\_count DESC,

initials ASC

;

**Translate**

SELECT

    capital,

    TRANSLATE(capital, 'áãåçéíñóú', 'aaaceinou') AS translated\_name

FROM

    countries

;

**LTRIM & RTRIM**

SELECT

    LTRIM(peak\_name, 'M') AS left\_trim,

    RTRIM(peak\_name, 'm') AS right\_trim

FROM

    peaks

;