**Lab Exercises**

**Towns Addresses**

SELECT

    t.town\_id,

    t.name AS "town\_name",

    a.address\_text

FROM

    addresses AS a

JOIN

    towns AS t

USING

    (town\_id)

WHERE

    t.name IN ('San Francisco', 'Sofia', 'Carnation')

ORDER BY

    a.town\_id,

    a.address\_id

;

**Managers**

SELECT

    e.employee\_id,

    CONCAT (e.first\_name, ' ', e.last\_name) AS "full\_name",

    d.department\_id,

    d.name AS "department\_name"

FROM

    employees AS e

JOIN

    departments AS d

ON

    e.employee\_id = d.manager\_id

ORDER BY

    e.employee\_id

LIMIT 5

;

**Employees Projects**

SELECT

    e.employee\_id,

    CONCAT (e.first\_name, ' ', e.last\_name) AS "full\_name",

    ep.project\_id,

    p.name AS "project\_name"

FROM

    employees AS e

JOIN

    employees\_projects AS ep

USING

    (employee\_id)

JOIN

    projects AS p

USING

    (project\_id)

WHERE

    ep.project\_id = 1

;

**Higher Salary**

SELECT

    COUNT (\*)

FROM

    employees

WHERE

    salary > (SELECT AVG(salary) FROM employees)

**Homework Exercises**

**First 10 Apartments Booked At**

SELECT

    a.name,

    a.country,

    CAST(b.booked\_at AS DATE)

FROM

    apartments AS a

LEFT JOIN

    bookings AS b

USING

    (booking\_id)

LIMIT 10

;

**First 10 Customers with Bookings**

SELECT

    b.booking\_id,

    CAST(b.starts\_at AS DATE),

    b.apartment\_id,

    CONCAT (c.first\_name, ' ', c.last\_name) AS "customer\_name"

FROM

    bookings AS b

RIGHT JOIN

    customers AS c

USING

    (customer\_id)

ORDER BY

    "customer\_name"

LIMIT 10

;

**Booking Information**

SELECT

    b.booking\_id,

    a.name AS "apartment\_owner",

    a.apartment\_id,

    CONCAT (c.first\_name, ' ', c.last\_name) AS "customer\_name"

FROM

    bookings AS b

FULL JOIN

    apartments AS a

USING

    (booking\_id)

FULL JOIN

    customers AS c

USING

    (customer\_id)

ORDER BY

    b.booking\_id,

    "apartment\_owner",

    "customer\_name"

;

**Unassigned Apartments**

SELECT

    b.booking\_id,

    b.apartment\_id,

    c.companion\_full\_name

FROM

    bookings AS b

JOIN

    customers AS c

USING

    (customer\_id)

WHERE

    b.apartment\_id IS NULL

;

**Bookings Made by Lead**

SELECT

    b.apartment\_id,

    b.booked\_for,

    c.first\_name,

    c.country

FROM

    bookings AS b

JOIN

    customers AS c

USING

    (customer\_id)

WHERE

    c.job\_type IN ('Lead')

;

**Total Sum of Nights**

SELECT

    a.name,

    SUM(b.booked\_for)

FROM

    apartments AS a

JOIN

    bookings AS b

USING

    (apartment\_id)

GROUP BY

    a.name

ORDER BY

    a.name

;

**Popular Vacation Destination**

SELECT

    a.country,

    COUNT(b.booking\_id) AS "booking\_count"

FROM

    bookings AS b

JOIN

    apartments AS a

USING

    (apartment\_id)

WHERE

    b.booked\_at > '2021-05-18 07:52:09.904+03'

    AND

    b.booked\_at < '2021-09-17 19:48:02.147+03'

GROUP BY

    a.country

ORDER BY

    "booking\_count" DESC

;

**Bulgaria's Peaks Higher than 2835 Meters**

SELECT

    mc.country\_code,

    m.mountain\_range,

    p.peak\_name,

    p.elevation

FROM

    mountains AS m

JOIN

    peaks AS p

ON

    m.id = p.mountain\_id

JOIN

    mountains\_countries AS mc

USING

    (mountain\_id)

WHERE

    p.elevation > 2835

    AND

    mc.country\_code = 'BG'

ORDER BY

    p.elevation DESC

;

**Count Mountain Ranges**

SELECT

    mc.country\_code,

    COUNT(m.mountain\_range) AS "mountain\_range\_count"

FROM

    mountains\_countries AS MC

JOIN

    mountains AS m

ON

    m.id = mc.mountain\_id

WHERE

    mc.country\_code IN ('US', 'RU', 'BG')

GROUP BY

    mc.country\_code

ORDER BY

    "mountain\_range\_count" DESC

;

**Rivers in Africa**

SELECT

    c.country\_name,

    r.river\_name

FROM

    countries AS c

LEFT JOIN

    countries\_rivers AS cr

USING

    (country\_code)

LEFT JOIN

    rivers AS r

ON

    cr.river\_id = r.id

WHERE

    c.continent\_code = 'AF'

ORDER BY

    c.country\_name

LIMIT 5

;

**Minimum Average Area Across Continents**

SELECT

    MIN("average\_area") AS "min\_average\_area"

FROM (

    SELECT

        AVG(area\_in\_sq\_km) AS "average\_area"

    FROM

        countries

    GROUP BY

        continent\_code

) AS "min\_average\_area"

;

**Countries Without Any Mountains**

SELECT

    COUNT(\*) AS "countries\_without\_mountains"

FROM

    countries AS c

LEFT JOIN

    mountains\_countries AS mc

USING

    (country\_code)

WHERE

    mc.mountain\_id IS NULL

;

**Monasteries by Country**

UPDATE

    countries

SET

    country\_name = 'Burma'

WHERE

    country\_name = 'Myanmar'

;

INSERT INTO monasteries(monastery\_name, country\_code)

VALUES

    ('Hanga Abbey', (SELECT country\_code FROM countries WHERE country\_name = 'Tanzania')),

    ('Myin-Tin-Daik', (SELECT country\_code FROM countries WHERE country\_name = 'Myanmar'))

;

SELECT

    continent\_name,

    country\_name,

    COUNT(m.country\_code) AS "monasteries\_count"

FROM

    continents AS c

JOIN

    countries AS countr

USING

    (continent\_code)

LEFT JOIN

    monasteries AS m

USING

    (country\_code)

WHERE

    NOT countr.three\_rivers

GROUP BY

    countr.country\_name,

    c.continent\_name

ORDER BY

    "monasteries\_count" DESC,

    countr.country\_name

;

**Monasteries by Continents and Countries**

UPDATE

    countries

SET

    country\_name = 'Burma'

WHERE

    country\_name = 'Myanmar'

;

INSERT INTO monasteries(monastery\_name, country\_code)

VALUES

    ('Hanga Abbey', (SELECT country\_code FROM countries WHERE country\_name = 'Tanzania')),

    ('Myin-Tin-Daik', (SELECT country\_code FROM countries WHERE country\_name = 'Myanmar'))

;

SELECT

    continent\_name,

    country\_name,

    COUNT(m.country\_code) AS "monasteries\_count"

FROM

    continents AS c

JOIN

    countries AS countr

USING

    (continent\_code)

LEFT JOIN

    monasteries AS m

USING

    (country\_code)

WHERE

    NOT countr.three\_rivers

GROUP BY

    countr.country\_name,

    c.continent\_name

ORDER BY

    "monasteries\_count" DESC,

    countr.country\_name

;