**Lab Exercises**

**Mountains and Peaks**

CREATE TABLE mountains(

    id SERIAL PRIMARY KEY,

    name VARCHAR(50)

)

;

CREATE TABLE peaks(

    id SERIAL PRIMARY KEY,

    name VARCHAR(50),

    mountain\_id INT,

    CONSTRAINT

        fk\_peaks\_mountains

        FOREIGN KEY(mountain\_id)

            REFERENCES mountains(id)

)

;

**Trip Organization**

SELECT

    v.driver\_id,

    v.vehicle\_type,

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

FROM

    campers AS c

JOIN

    vehicles AS v

ON

    c.id = v.driver\_id

;

**SoftUni Hiking**

SELECT

    r.start\_point,

    r.end\_point,

    r.leader\_id,

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

FROM

    campers AS c

JOIN

    routes AS r

ON

    c.id = r.leader\_id

;

**Delete Mountains**

CREATE TABLE mountains(

    id SERIAL PRIMARY KEY,

    name VARCHAR

)

;

CREATE TABLE peaks(

    id SERIAL PRIMARY KEY,

    name VARCHAR,

    mountain\_id INT,

    CONSTRAINT fk\_mountain\_id

    FOREIGN KEY(mountain\_id)

    REFERENCES mountains(id)

    ON DELETE CASCADE

)

;

**Homework Exercises**

**PRIMARY KEY**

CREATE TABLE products(

    product\_name VARCHAR(100)

)

;

INSERT INTO

    products (product\_name)

VALUES

    ('Broccoli'),

    ('Shampoo'),

    ('Toothpaste'),

    ('Candy')

;

ALTER TABLE products

ADD COLUMN id SERIAL PRIMARY KEY

;

**Remove Primary Key**

ALTER TABLE products

DROP CONSTRAINT products\_pkey

;

**Customs**

CREATE TABLE passports(

    id INT GENERATED ALWAYS AS IDENTITY(START WITH 100 INCREMENT 1) PRIMARY KEY,

    nationality VARCHAR(50)

)

;

INSERT INTO passports(nationality)

VALUES

    ('N34FG21B'),

    ('K65LO4R7'),

    ('ZE657QP2')

;

CREATE TABLE people(

    id SERIAL PRIMARY KEY,

    first\_name VARCHAR(50),

    salary DECIMAL(10, 2),

    passport\_id INT,

    CONSTRAINT fk\_passports

        FOREIGN KEY(passport\_id)

            REFERENCES passports(id)

)

;

INSERT INTO

    people(first\_name, salary, passport\_id)

VALUES

    ('Roberto', 43300.0000, 101),

    ('Tom', 56100.0000, 102),

    ('Yana', 60200.0000, 100)

;

**Car Manufacture**

CREATE TABLE manufacturers(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    name VARCHAR(50)

);

CREATE TABLE models(

    id INT GENERATED ALWAYS AS IDENTITY(START WITH 1000 INCREMENT 1) PRIMARY KEY,

    model\_name VARCHAR(50),

    manufacturer\_id INT,

    CONSTRAINT fk\_models\_manufacturers

        FOREIGN KEY(manufacturer\_id)

            REFERENCES manufacturers(id)

);

CREATE TABLE production\_years(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    established\_on DATE,

    manufacturer\_id INT,

    CONSTRAINT fk\_production\_years\_manufacturers

        FOREIGN KEY(manufacturer\_id)

            REFERENCES manufacturers(id)

);

INSERT INTO manufacturers(name)

VALUES

    ('BMW'),

    ('Tesla'),

    ('Lada')

;

INSERT INTO models(model\_name, manufacturer\_id)

VALUES

    ('X1', 1),

    ('i6', 1),

    ('Model S', 2),

    ('Model X', 2),

    ('Model 3', 2),

    ('Nova', 3)

;

INSERT INTO production\_years(established\_on, manufacturer\_id)

VALUES

    ('1916-03-01', 1),

    ('2003-01-01', 2),

    ('1966-05-01', 3)

;

**Photo Shooting**

CREATE TABLE customers(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    name VARCHAR(50),

    date DATE

);

CREATE TABLE photos(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    url VARCHAR(50),

    place VARCHAR(50),

    customer\_id INT,

    CONSTRAINT fk\_photos\_customers

        FOREIGN KEY (customer\_id)

            REFERENCES customers(id)

);

INSERT INTO customers(name, date)

VALUES

    ('Bella', '2022-03-25'),

    ('Philip', '2022-07-05')

;

INSERT INTO photos(url, place, customer\_id)

VALUES

    ('bella\_1111.com',  'National Theatre', 1),

    ('bella\_1112.com',  'Largo', 1),

    ('bella\_1113.com',  'The View Restaurant', 1),

    ('philip\_1121.com', 'Old Town', 2),

    ('philip\_1122.com', 'Rowing Canal', 2),

    ('philip\_1123.com', 'Roman Theater', 2)

;

**Study Session**

CREATE TABLE students(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    student\_name VARCHAR(50)

);

CREATE TABLE exams(

    id INT GENERATED ALWAYS AS IDENTITY (START WITH 101 INCREMENT 1) PRIMARY KEY,

    exam\_name VARCHAR(50)

);

CREATE TABLE study\_halls(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    study\_hall\_name VARCHAR(50),

    exam\_id INT,

    CONSTRAINT fk\_study\_halls\_exams

        FOREIGN KEY (exam\_id)

            REFERENCES exams(id)

);

CREATE TABLE students\_exams(

    student\_id INT,

    exam\_id INT,

    CONSTRAINT fk\_students\_exams\_students

        FOREIGN KEY (student\_id)

            REFERENCES students(id),

    CONSTRAINT fk\_students\_exams\_exams

        FOREIGN KEY (exam\_id)

            REFERENCES exams(id)

);

INSERT INTO students(student\_name)

VALUES

    ('Mila'),

    ('Toni'),

    ('Ron')

;

INSERT INTO exams(exam\_name)

VALUES

    ('Python Advanced'),

    ('Python OOP'),

    ('PostgreSQL')

;

INSERT INTO study\_halls(study\_hall\_name, exam\_id)

VALUES

    ('Open Source Hall', 102),

    ('Inspiration Hall', 101),

    ('Creative Hall', 103),

    ('Masterclass Hall', 103),

    ('Information Security Hall', 103)

;

INSERT INTO students\_exams(student\_id, exam\_id)

VALUES

    (1, 101),

    (1, 102),

    (2, 101),

    (3, 103),

    (2, 102),

    (2, 103)

;

**Online Store**

CREATE TABLE item\_types(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    item\_type\_name VARCHAR(50)

);

CREATE TABLE items(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    item\_name VARCHAR(50),

    item\_type\_id INT,

    CONSTRAINT fk\_items\_item\_types

        FOREIGN KEY (item\_type\_id)

            REFERENCES item\_types(id)

);

CREATE TABLE cities(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    city\_name VARCHAR(50)

)

;

CREATE TABLE customers(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    customer\_name VARCHAR(50),

    birthday DATE,

    city\_id INT,

    CONSTRAINT fk\_customers\_cities

        FOREIGN KEY (city\_id)

            REFERENCES cities(id)

);

CREATE TABLE orders(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    customer\_id INT,

    CONSTRAINT fk\_orders\_customers

        FOREIGN KEY (customer\_id)

            REFERENCES customers(id)

);

CREATE TABLE order\_items(

    order\_id INT,

    item\_id INT,

    CONSTRAINT fk\_order\_items\_items

        FOREIGN KEY (item\_id)

            REFERENCES items(id),

    CONSTRAINT fk\_order\_items\_orders

        FOREIGN KEY (order\_id)

            REFERENCES orders(id)

);

**Delete Cascade**

ALTER TABLE countries

ADD

    CONSTRAINT fk\_countries\_continents

        FOREIGN KEY (continent\_code)

            REFERENCES continents (continent\_code) ON DELETE CASCADE

;

ALTER TABLE countries

ADD

    CONSTRAINT fk\_countries\_currencies

        FOREIGN KEY (currency\_code)

            REFERENCES currencies (currency\_code) ON DELETE CASCADE

;

**Update Cascade**

ALTER TABLE countries\_rivers

ADD

    CONSTRAINT fk\_countries\_rivers\_countries

        FOREIGN KEY (country\_code)

            REFERENCES countries(country\_code) ON UPDATE CASCADE

;

ALTER TABLE countries\_rivers

ADD

    CONSTRAINT fk\_countries\_rivers\_rivers

        FOREIGN KEY (river\_id)

            REFERENCES rivers(id) ON UPDATE CASCADE

;

**SET NULL**

CREATE TABLE customers(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    customer\_name VARCHAR(50)

)

;

CREATE TABLE contacts(

    id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

    contact\_name VARCHAR(50),

    phone VARCHAR(50),

    email VARCHAR(50),

    customer\_id INT,

    FOREIGN KEY (customer\_id)

        REFERENCES customers (id)

            ON DELETE SET NULL

            ON UPDATE CASCADE

)

;

INSERT INTO customers(customer\_name)

VALUES

    ('BlueBird Inc'),

    ('Dolphin LLC')

;

INSERT INTO contacts(contact\_name, phone, email, customer\_id)

VALUES

    ('John Doe', '(408)-111-1234', 'john.doe@bluebird.dev', 1),

    ('Jane Doe', '(408)-111-1235',  'jane.doe@bluebird.dev', 1),

    ('David Wright', '(408)-222-1234',  'david.wright@dolphin.dev', 2)

;

DELETE FROM

    customers

WHERE

    id = 1

;

**Peaks in Rila**

SELECT

    m.mountain\_range,

    p.peak\_name,

    p.elevation

FROM

    mountains AS m

JOIN

    peaks AS p

ON

    m.id = p.mountain\_id

WHERE

    m.mountain\_range LIKE ('Rila')

ORDER BY

    p.elevation DESC

;

**Countries Without Any Rivers**

SELECT

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

FROM

    countries AS c

LEFT JOIN

    countries\_rivers AS cr

ON

    c.country\_code = cr.country\_code

WHERE

    cr.river\_id IS NULL

;