TABLE RELATIONS

1. Database Desing: fundamental concepts
    1.1 Identification of the entities
    1.2 Defining table columns
    2.3 Defining primary keys:
        - always define an additional column for the PK
        - do not use existing one(e.g.: name)
        - can be an integer number
        - must be declared as PRIMARY KEY
        - put the PK in the first column
        - exceptions: entities that have a well-known ID
          e.g.: countries(BG, DE, US), currencies(USD, EUR, BGN)
    2.4 Modelling relationships
    2.5 Defining constraints
    3.6 Filling test data

2. Table Relations: relational DB model in action
    - Relationships between tables are based on interconnections:
    PRIMARY KEY / FOREIGN KEY

    -PRIMARY KEY:

```
id INT PRIMARY KEY

id SERIAL PRIMARY KEY

id INT GENERATED ALWAYS AS IDENTITY --> cannot provide an explicit value

id INT GENERATED BY DEFAULT AS IDENTITY --> cannot guarantee uniqueness
```

    - FOREIGN KEY:
        - The FK is an identifier of a record located in another table(usually its PK)
        - By using relationships, we avoid repeating data in the DB

            - one-to-many - e.g.: mountains / peaks
            - many-to-many - e.g.: students / courses
            - one-to-one - e.g.: country/capital

```
CREATE TABLE clients(
      id SERIAL PRIMARY KEY,
      name VARCHAR(30)
);
CREATE TABLE orders(
      id SERIAL PRIMARY KEY,
      client_id INT REFERENCES clients(id)
);
CREATE TABLE orders(
      id INT PRIMARY KEY,
      client_id INT,
      CONSTRAINT fk_orders_clients
            FOREIGN KEY(client_id)
                  REFERENCES clients(id)
);

#1.

CRATE 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 (mointain_id)
            REFERENCES mountains(id)
);
```

ADD CONSTRAINT AFTER THE TABLE HAS BEEN CREATED

```
ALTER TABLE
      peaks
ADD CONSTRAINT fk_peaks_mountains
FOREIGN KEY (mountain_id)
REFERENCES mountains(id)
```

2.2   MANY-TO-MANY:

```
SELECT
      CONCAT(m.fist_name, ' ', m.last_name)
      CONCAT(w.fist_name, ' ', w.last_name)
FROM men as m,
JOIN men_woman
ON m.id = men_woman.men_id
      JOIN woman AS w
      ON men_woman.woman_id = w.id
```

3. CASCADE:
- ON UPDATE CASCADE;
- ON DELETE CASCADE;