**CSCI 341 Homework 2**

You are given a description of the database tables below. The tables are simplified and slightly modified versions of those ones you have worked on HW1. You are supposed to create a relational database (**part 1**) and write and execute queries that will extract and update some data from the database (**part 2**).

**Part 1 (30 points)**

Construct a physical database according to the given schema. To create the physical database, it is required to use a database management system. You will use the free DBMS tool, PostgreSQL.

*The schema is:*

* **DiseaseType**(id:integer, description:varchar(140))
* **Country**(cname:varchar(50), population:bigint)
* **Disease**(disease code:varchar(50), pathogen:varchar(20), description:varchar(140), id:integer) References **DiseaseType** (id)
* **Discover**(cname:varchar(50), disease code:varchar(50), first enc date:date) References **Disease** (disease code), Country (cname)
* **Users**(email:varchar(60), name:varchar(30), surname:varchar(40), salary:integer, phone:varchar(20), cname:varchar(50)) References **Country** (cname)
* **PublicServant**(email:varchar(60), department:varchar(50)) References **Users** (email)
* **Doctor**(email:varchar(60), degree:varchar(20)) References **Users** (email)
* **Specialize**(id:integer, email:varchar(60)) References **DiseaseType** (id), **Doctor** (email)
* **Record**(email:varchar(60), cname:varchar(50), disease code:varchar(50), total deaths:integer, total patients:integer) References **Disease** (disease code), **Country** (cname), **PublicServant** (email)

After constructing the database model, you should create tables and insert data instances on the tables (at least 10 instances for each table). Please note that the result set of the queries from **part 2** should be non-empty. (This means that you must think of data instances you are going to insert).

**Part 2 (70 points)**

In this part, you are going to connect to your PostgreSQL database and execute some queries and updates. You are asked to use Python3 and SQLAlchemy library to connect and interact with your database. Please refer to the SQLAlchemy documentation to start working (<https://docs.sqlalchemy.org/en/14/intro.html>). You can use SQLAlchemy ORM or execute the raw SQL scripts using Textual SQL ([https://docs.sqlalchemy.org/en/13/core/tutorial.html#using-textual-sql](https://docs.sqlalchemy.org/en/13/core/tutorial.html)). Chose any approach you prefer as long as it is easy to demonstrate. You can also use IDEs to help you organize your work. Some of the options are DataGrip and Pycharm with Database extensions.

**SQL Queries:**

* List the disease code and the description of diseases that are caused by “bacteria” (pathogen) and were discovered before 1990.
* List the name, surname and degree of doctors who are not specialized in “infectious diseases”.
* List the name, surname and degree of doctors who are specialized in more than 2 disease types.
* For each country list the cname and average salary of doctors who are specialized in “virology”.
* List the departments of public servants who report “covid-19” cases in more than one country and the number of such public servants who work in these departments. (i.e “Dept1 3” means that in the “Dept1” department there are 3 such employees.)
* Double the salary of public servants who have recorded covid-19 patients more than 3 times.
* Delete the users whose surname contain the substring “bek” (e.g. Alibek, bek)
* Create an index namely “idx pathogen” on the “pathogen” field.
* List the email, name, and department of public servants who have created records where the number of patients is between 100000 and 999999
* List the top 5 counties with the highest number of total patients recorded.
* Group the diseases by disease type and the total number of patients treated.

**Submission**

Export your database in .sql format (you can use pg\_dump in order to do this).

Submit a zip file on Moodle that will consist of the following files:

* .sql file of your database.
* .py file that will connect to your database and execute all the queries.