Databases Exercise

To create a MySQL server and client, we will be using a tool called `docker`. We aren't going to go into the details of how it works in this module, however we do have an entire module dedicated to it soon after this one. For now, all you need to know is that Docker can help us install software like a database in a contained way, away from our main system.

Part 1

1. Ensure you have Docker Desktop installed and running (you can check with `docker -v`).
2. Ensure you have the `docker-setup` directory provided by your instructor.
3. Run the following command **inside** the directory in a terminal. This will create both the client and server for us which is running on localhost.

```
$ docker-compose up -d
```

You should get the following output:

```
Creating mysql_container   ... done
Creating adminer_container ... done
```

4. Navigate to the following URL to ensure that you can see the `Adminer` interface:

http://localhost:8080/

5. Fill in the username (`root`) and password field (`password`), leave the database field blank.
6. Select `SQL Command` on the left.
7. We'll create our own database with:

```
CREATE DATABASE test;
```

7. Select `test` in the DB dropdown on the left.
8. Now we'll create our first table with:

```
CREATE TABLE person (
  person_id INT NOT NULL AUTO_INCREMENT,
  first_name VARCHAR(255) NOT NULL,
  last_name VARCHAR(255) NOT NULL,
  age INT,
  PRIMARY KEY(person_id)
);
```

9. Let's alter the table by adding a new field:

```
ALTER TABLE person
ADD email varchar(255);
```

You now have your very own database, with a single table called `person`. There's no data in there yet, but you can verify it exists by navigating to `http://localhost:8080/?server=db&username=root&db=test&table=person`.

Part 2

1. Insert rows into the `person` table.
2. Try and update some of the rows.
3. Try and delete some rows you created.
4. Build up a SELECT statement one part at a time and start to refine your query (use all of the keywords `SELECT, FROM, WHERE, ORDER BY, LIMIT`).

Part 3

1. Activate a virtual environment (refer back to the Python Ecosystem module if you forgot).
2. Install the dependencies from `requirements.txt` with `pip install -r requirements.txt`.
3. Run the `mysql_example.py` file. Inspect the file contents to get an understanding of how it works.
4. Using the original Python file as inspiration, update `insert_intro_db.py` to **insert** a new record into your database. Here is a hint.
5. You can verify this works by running `SELECT * FROM person` in the Adminer web page.