Qventus Data Platform Assignment

For this project, you'll be using a data sources provided to you in a csv format.
You'll download the data file and write SQL queries and Python program to manipulate the data. We use SQL and Python for our backend code primarily at Qventus so we expect part 1 and 2 to use similar languages.

● When you are done with both parts of this project, zip up your project directory and upload it to Google Drive. Make sure we have access and send us a link.
● We expect this to take 2 - 3 hours to complete. If it seems like it will take significantly longer than that to complete, please ask us first to make sure you aren't over-complicating the project.

At any point, you have any questions, please don't hesitate to reach out to us.

Part 0: Get the data

1. Download the following excel files. This will be used in part 1 and part 2 of the assignment.
    a. Patient Data CSV
    b. Proecdure Orders CSV

Part 1: Answer questions in SQL

1. Create the Schema (Schema is given below)
2. Upload the data from the CSV file given from Part 0 into these tables

```
CREATE TABLE `procedure_orders` (
  `id` int NOT NULL AUTO_INCREMENT,
  `procedure_id` int NOT NULL,
  `encounter_id` int NOT NULL,
  `order_time` timestamp NULL DEFAULT NULL,
  `procedure_name` varchar(128) DEFAULT NULL,
  `order_class` varchar(128) DEFAULT NULL,
  `order_variety` varchar(128) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `procedure_id` (`procedure_id`),
```

```
  KEY `encounter_id` (`encounter_id`,`order_time`)
)

CREATE TABLE `patient_data` (
  `id` int NOT NULL AUTO_INCREMENT,
  `encounter_id` int DEFAULT NULL,
  `admit_time` timestamp NULL DEFAULT NULL,
  `discharge_time` timestamp NULL DEFAULT NULL,
  `discharge_department` varchar(128) DEFAULT NULL,
  `final_drg` varchar(128) DEFAULT NULL,
  `admitting_ICD10` varchar(128) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `encounter_id` (`encounter_id`)
);
```

Note: For each question, write down your SQL query and results you see from executing the query. Make sure to copy all the sql answers to a part1.sql file. If also possible please explain what your sql is doing (this is completely up to you on how you want to share this explanation).

1. What were the top 5 most placed order class in the last month of the data set?
2. On average, how many procedure orders were placed per patient within 24 hours of their admit time by month in the last year of the dataset?
3. On average, how many orders were placed per patient within 24 hours of their admit split by admit on a weekend vs. weekday by year
4. How would you visualize the result in question 3? Who would you expect your audience to be? Why do you think it might be useful? You can either create a visualization or tell us how you want to visualize this data.

Part 2: Write a python script

Using the data provided from the Part 0 CSVs, please write a python script to import the CSVs and answer the following questions. You are allowed to use any type of library to import the csv into the application.

1. Create a function called **search_procedure_orders** that takes in a search pattern and the procedure_orders_data and returns relevant procedure orders as an array.

Example:

```
search_procedure_orders('hospitalist%', procedure_orders_data) →
['hospitalist ip consult']
```

2. Create a function called **map_orders_to_patients** that takes in patients data and procedure orders and returns number of orders per each patient

Example:

```
map_procedure_orders_to_patients(patient_data, procedure_orders_data)
→ [{encounter_id: 1, procedure_orders: 10}]
```

3. Create a function called **patients_with_procedure** that takes in patient_data, procedure_orders_data, and a search pattern and returns the number of matching orders per each patient, raising an error if there are no patients with such a procedure

Example:

```
patients_with_procedure(patient_data, procedure_orders_data,
'hospitalist%') → [{encounter_id: 1, procedure_orders: 10}]
```

4. How would you make **patients_with_procedure** simpler?

Evaluation

- Accuracy
  - We'll evaluate the accuracy of your answers.
  - For python, you can choose to change the datastructure, just let us know if you do have change it to something else.
- Performance
  - All results should return within 1 second.
- Code Quality

- We are looking for clear, easy to understand code. Write code that makes your thinking algorithm as obvious as possible.
- Prioritize readability over cleverness. We care less that you can code a solution to this problem in one line, and more that we are able to easily follow your code.
- Do your best to make the code readable without requiring reading lots of documentation.