**Problem Set 1: Basic SQL and OMOP Fundamentals**

**Part I: Basic SQL**

Solve the SQL Murder Mystery at <https://mystery.knightlab.com/>. Copy and paste the queries that you used to arrive at your answer here.

**Jeremy Bowers committed the murder; Miranda Priestly was the brains behind the murder!**

**SELECT \***

**FROM crime\_scene\_report**

**WHERE type = "murder"**

**AND city = "SQL City"**

**AND date = 20180115;**

**SELECT \***

**FROM person p,**

**(SELECT max(address\_number) as max\_add**

**FROM person**

**WHERE address\_street\_name = "Northwestern Dr") p2**

**WHERE address\_street\_name = "Northwestern Dr"**

**AND address\_number = p2.max\_add;**

**SELECT \***

**FROM person**

**WHERE address\_street\_name = "Franklin Ave"**

**AND name LIKE ('Annabel%');**

**SELECT \***

**FROM interview**

**WHERE person\_id IN (14887,16371);**

**SELECT name**

**FROM person p,**

**(SELECT id**

**FROM drivers\_license**

**WHERE plate\_number LIKE "%H42W%") dl**

**WHERE license\_id = dl.id**

**INTERSECT**

**SELECT name**

**FROM get\_fit\_now\_member**

**WHERE id LIKE "48Z%";**

**SELECT \***

**FROM interview**

**WHERE person\_id = 67318;**

**SELECT p.id**

**FROM person p,**

**(SELECT \***

**FROM drivers\_license**

**WHERE hair\_color = "red"**

**AND height BETWEEN 65 AND 67**

**AND car\_make = "Tesla"**

**AND car\_model = "Model S") dl**

**WHERE p.license\_id = dl.id**

**INTERSECT**

**SELECT DISTINCT fb.person\_id**

**FROM facebook\_event\_checkin fb,**

**(SELECT person\_id, COUNT(\*) as c**

**FROM facebook\_event\_checkin**

**WHERE event\_name = "SQL Symphony Concert"**

**AND date LIKE "2017%"**

**GROUP BY person\_id) fb2**

**WHERE fb.person\_id = fb2.person\_id**

**AND fb2.c = 3;**

**SELECT name**

**FROM person**

**WHERE id = 99716;**

**Part II: OMOP Fundamentals**

*For this part of the problem set, you will use the OMOP database on EduHeLx.*

Answer each of the following questions. For each question, provide the SQL query or queries you used to arrive at your answer.

**1.** Every time a person has a healthcare encounter, they get a new row in the *visit\_occurrence* table. Use the *person* and *visit\_occurrence* tables to answer the following questions:

**a**. How many visits occurred in the year 2019?

**SELECT COUNT(\*)**

**FROM visit\_occurrence**

**WHERE EXTRACT(YEAR FROM visit\_start\_date) = 2019;**

**b.** How many unique patients (i.e., people in the person table) had a visit in the year 2019?

**SELECT COUNT(DISTINCT p.person\_id)**

**FROM person p**

**LEFT JOIN visit\_occurrence v on p.person\_id = v.person\_id**

**WHERE EXTRACT(YEAR FROM v.visit\_start\_date) = 2019;**

**c.** How many unique patients have at least one visit at any time?

**SELECT COUNT(DISTINCT p.person\_id)**

**FROM person p**

**LEFT JOIN visit\_occurrence v on p.person\_id = v.person\_id**

**WHERE v.person\_id IS NOT NULL;**

**d.** How many unique patients have no visits?

**SELECT COUNT(DISTINCT p.person\_id)**

**FROM person p LEFT JOIN visit\_occurrence v ON p.person\_id = v.person\_id**

**WHERE v.person\_id IS NULL;**

**2.** Write a query to pull back the person ID, birth date, and death date of any patient who is deceased. (Use the *person* table and the *death* table to find this information.) Include a fourth column in your result that calculates the patient’s age at the date of their death.

**SELECT DISTINCT p.person\_id, p.birth\_datetime, d.death\_datetime, EXTRACT(YEAR FROM AGE(d.death\_datetime, p.birth\_datetime)) as age\_of\_death**

**FROM person p**

**LEFT JOIN death d on p.person\_id = d.person\_id**

**WHERE d.person\_id IS NOT NULL**

**ORDER BY p.person\_ID;**

**3.** The *measurement* table contains items like laboratory test results and vital sign measurements. The type of measurement (weight, height, temperature, etc.) is stored in the measurement\_concept\_id column. 3020891 is the measurement\_concept\_id for body temperature.

**a.** How many body temperatures have been recorded with a value greater than or equal to 37.8 degrees Celsius?

**SELECT COUNT(\*)**

**FROM measurement**

**WHERE measurement\_concept\_id = 3020891**

**AND value\_as\_number >= 37.8;**

**b.** What is the lowest body temperature recorded in the *measurement* table? What is the highest?

**SELECT MIN(value\_as\_number) AS min\_temp, MAX(value\_as\_number) AS max\_temp**

**FROM measurement**

**WHERE measurement\_concept\_id = 3020891;**