Writeup for Chorus SQL Questions

**Question 1: Retrieve all active patients. Write a query to return all patients who are active.**

Assumption: Since the columns aren’t listed, return all columns.

*SELECT \* FROM public."Patient" WHERE active = true;*

**Question 2: Find encounters for a specific patient. Given a patient\_id, retrieve all encounters for that patient, including the status and encounter date.**

*SELECT \* FROM public."Encounter" WHERE patient\_id = '395714a3-216f-4526-b648-d8b24ef324e1';*

**Question 3: List all observations recorded for a patient. Write a query to fetch all observations for a given patient\_id, showing the observation type, value, unit, and recorded date.**

*SELECT*

*id AS observation\_id*

*, patient\_id*

*, type AS observation\_type*

*, value*

*, unit*

*, recorded\_at*

*FROM public."Observation" WHERE patient\_id ='395714a3-216f-4526-b648-d8b24ef324e1';*

**Question 4: Find the most recent encounter for each patient. Retrieve each patient’s most recent encounter (based on encounter\_date). Return the patient\_id, encounter\_date, and status.**

*WITH cte as (*

*SELECT*

*patient\_id*

*,encounter\_date*

*, status*

*, RANK() OVER (PARTITION BY patient\_id ORDER BY encounter\_date DESC) ranks*

*FROM public."Encounter"*

*)*

*SELECT*

*patient\_id*

*, encounter\_date*

*, status*

*FROM cte*

*where ranks = 1*

*order by patient\_id desc*

**Question 5: Find patients who have had encounters with more than one practitioner.**

**Write a query to return a list of patient IDs who have had encounters with more than one distinct practitioner.**

*WITH cte AS (*

*SELECT*

*patient\_id*

*, COUNT(DISTINCT practitioner\_id) counts*

*FROM public."Encounter"*

*GROUP BY patient\_id*

*)*

*SELECT patient\_id FROM cte WHERE counts>1;*

**Question 6: Find the top 3 most prescribed medication. Write a query to find the three most commonly prescribed medications from the MedicationRequest table, sorted by the number of prescriptions.**

*SELECT*

*medication\_name*

*, COUNT(DISTINCT id) AS num\_prescriptions*

*FROM public."MedicationRequest"*

*GROUP BY medication\_name*

*ORDER BY 2 DESC*

*LIMIT 3*

**Question 7: Get practitioners who have never prescribed any medication. Write a query to find all practitioners who do not appear in the MedicationRequest table as a prescribing practitioner.**

Assumption: The columns weren’t specified, so I am returning all columns.

First Solution:

*SELECT \* FROM public."Practitioner"*

*WHERE id NOT IN (SELECT DISTINCT practitioner\_id AS id FROM public."MedicationRequest");*

Second Solution

*SELECT \* FROM public."Practitioner"*

*WHERE name NOT IN (SELECT DISTINCT name FROM public."MedicationRequest");*

**Question 8: Find the average number of encounters per patient. Calculate the average number of encounters per patient, rounded to two decimal places.**

Solution One: If we include patients with 0 encounters in the calculation:

*WITH patients\_w\_encounters AS (*

*SELECT*

*patient\_id*

*, COUNT(DISTINCT id) counts*

*FROM public."Encounter"*

*GROUP BY 1*

*),*

*patients\_w\_no\_encounters AS (*

*SELECT*

*id AS patient\_id*

*, 0 counts*

*FROM public."Patient"*

*WHERE id NOT IN (SELECT patient\_id AS id FROM public."Encounter")*

*),*

*combined AS (*

*SELECT \* FROM patients\_w\_no\_encounters*

*UNION ALL*

*SELECT \* FROM patients\_w\_encounters*

*)*

*SELECT ROUND(AVG(counts),2) FROM combined;*

Solution Two: If we don't include patients with 0 encounters:

*WITH cte as (*

*SELECT*

*patient\_id*

*, COUNT(DISTINCT id) counts*

*FROM public."Encounter"*

*GROUP BY 1*

*)*

*SELECT ROUND(AVG(counts),2) FROM cte;*

**Question 9: Identify patients who have never had an encounter but have a medication request. Write a query to find patients who have a record in the MedicationRequest table but no associated encounters in the Encounter table.**

*SELECT \* FROM public."Patient"*

*WHERE id NOT IN (SELECT patient\_id AS id FROM public."Encounter") AND id IN (SELECT patient\_id AS id FROM public."MedicationRequest");*

**Question 10: Determine patient retention by cohort. Write a query to count how many patients had their first encounter in each month (YYYY-MM format) and still had at least one encounter in the following six months.**

*WITH firsts AS (*

*SELECT*

*patient\_id,*

*MIN(encounter\_date) AS first\_encounter\_date*

*FROM "Encounter"*

*GROUP BY patient\_id*

*),*

*seconds AS (*

*SELECT DISTINCT fe.patient\_id*

*FROM firsts fe*

*JOIN "Encounter" e*

*ON fe.patient\_id = e.patient\_id*

*AND e.encounter\_date > fe.first\_encounter\_date*

*AND e.encounter\_date <= fe.first\_encounter\_date + INTERVAL '6 months'*

*)*

*SELECT*

*DATE\_TRUNC('month', fe.first\_encounter\_date) AS encounter\_month,*

*COUNT(DISTINCT fe.patient\_id) AS patient\_count*

*FROM firsts fe*

*JOIN seconds se ON fe.patient\_id = se.patient\_id*

*GROUP BY encounter\_month*

*ORDER BY encounter\_month;*

**Note: We can pass parameters for the Questions 2 and 3 like this:**

Question 2

*DEALLOCATE ALL;*

*PREPARE my\_query (UUID) AS*

*SELECT \* FROM public."Encounter" WHERE patient\_id = $1;*

*EXECUTE my\_query('395714a3-216f-4526-b648-d8b24ef324e1'); -- you can replace this variable with any ID*

Question 3

*DEALLOCATE ALL;*

*PREPARE my\_query (UUID) AS*

*SELECT*

*id AS observation\_id*

*, patient\_id*

*, type AS observation\_type*

*, value*

*, unit*

*, recorded\_at*

*FROM public."Observation" WHERE patient\_id = $1;*

*EXECUTE my\_query('395714a3-216f-4526-b648-d8b24ef324e1');*