**PRCE\_001\_MEDICAL DATA HISTORY**

**use project\_medical\_data\_history;**

**select \* from admissions;**

**select \* from doctors;**

**select \* from patients;**

**select \* from province\_names;**

**-- 1. Show first name, last name, and gender of patients whose gender is 'M'**

**select first\_name,last\_name,gender from patients where gender='M';**

**-- 2. Show first name and last name of patients who do not have allergies.**

**select first\_name,last\_name,gender from patients where allergies !='Null';**

**select first\_name,last\_name,gender from patients where allergies is not null;**

**-- 3. Show first name of patients that start with the letter 'C'**

**select first\_name from patients where first\_name like 'c%';**

**-- 4. Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)**

**select first\_name,last\_name,weight from patients where weight between 100 and 120 ;**

**-- 5. Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'**

**select \* from patients where allergies is null;**

**select count(\*) from patients where allergies is null;**

**select count(\*) from patients ;**

**update allergies set allergies = 'NKA' where allergies is null;**

**-- 6. Show first name and last name concatenated into one column to show their full name.**

**select concat ( first\_name , last\_name) as result from patients;**

**-- 7. Show first name, last name, and the full province name of each patient.**

**select \* from province\_names;**

**select \* from patients;**

**select pa.first\_name,pa.last\_name,pr.province\_name from patients pa inner join province\_names pr on pa.province\_id = pr.province\_id;**

**-- 8. Show how many patients have a birth\_date with 2010 as the birth year.**

**select count(\*) from patients where birth\_date between '2010-01-01' and '2010-12-31';**

**-- 9. Show the first\_name, last\_name, and height of the patient with the greatest height.**

**select max(height) from patients;**

**select first\_name,last\_name,height from patients where height= (select max(height) from patients);**

**-- 10. Show all columns for patients who have one of the following patient\_ids: 1,45,534,879,1000**

**select \* from patients where patient\_id = '1' or patient\_id = '45' or patient\_id = '534' or patient\_id = '879' or patient\_id = '1000' ;**

**-- 11. Show the total number of admissions**

**select count(\*) from admissions;**

**select \* from admissions;**

**select \* from doctors;**

**-- 12. Show all the columns from admissions where the patient was admitted and discharged on the same day.**

**select \* from admissions where admission\_date=discharge\_date;**

**-- 13. Show the total number of admissions for patient\_id 579.**

**select count(patient\_id) from admissions where patient\_id = '579';**

**-- 14. Based on the cities that our patients live in, show unique cities that are in province\_id 'NS'?**

**select \* from patients;**

**select city, province\_id from patients where province\_id = 'NS';**

**-- 15. Write a query to find the first\_name, last name and birth date of patients who have height more than 160 and weight more than 70**

**select first\_name,last\_name,birth\_date,height,weight from patients where height>160 and weight>70;**

**-- 16. Show unique birth years from patients and order them by ascending.**

**select Distinct year (birth\_date) as a from patients order by a asc;**

**select date\_format(birth\_date,"%Y") from patients;**

**-- 17. Show unique first names from the patients table which only occurs once in the list. For example, if two or more people are named 'John' in the first\_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output. Tip: HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.**

**select Distinct (first\_name) from patients;**

**select first\_name from patients group by first\_name having count(first\_name)=1;**

**-- 18. Show patient\_id and first\_name from patients where their first\_name starts and ends with 's' and is at least 6 characters long.**

**select patient\_id,first\_name from patients where length (first\_name)>5 and first\_name like 's%%s';**

**-- 19. Show patient\_id, first\_name, last\_name from patients whose diagnosis is 'Dementia'. Primary diagnosis is stored in the admissions table.**

**select \* from patients;**

**select \* from admissions;**

**select patient\_id,diagnosis from admissions where diagnosis='Dementia';**

**select pa.patient\_id,pa.first\_name,pa.last\_name, a.diagnosis from patients as pa inner join admissions as a on pa.patient\_id=a.patient\_id where diagnosis = 'Dementia';**

**-- 20. Display every patient's first\_name. Order the list by the length of each name and then by alphabetically.**

**select first\_name from patients order by length(first\_name), first\_name asc;**

**-- 21. Show the total number of male patients and the total number of female patients in the patients table. Display the two results in the same row.**

**select count(patient\_id) from patients as pa where gender= 'M';**

**select count(patient\_id) from patients where gender= 'F';**

**select count(\*) from patients;**

**select(select count(\*) from patients where gender = "M") as "Male\_Patients",**

**(select count(\*) from patients where gender = "F") as "Female\_Patients" ; #nested select**

**-- 22. Show the total number of male patients and the total number of female patients in the patients table. Display the two results in the same row.**

**-- 23. Show patient\_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.**

**select patient\_id,diagnosis,count(\*) from admissions group by patient\_id,diagnosis having count(patient\_id)!=1; # admitted for multiple times with same diagnosis**

**select patient\_id,count(\*) from admissions group by patient\_id having count(patient\_id)!=1; # admitted for multiple times not for same diagnosis**

**select patient\_id,diagnosis from admissions;**

**-- 24. Show the city and the total number of patients in the city. Order from most to least patients and then by city name ascending.**

**select city,count(patient\_id) from patients group by city order by count(patient\_id) desc;**

**select city,count(patient\_id) from patients group by city order by city;**

**-- 25. Show first name, last name and role of every person that is either patient or doctor. The roles are either "Patient" or "Doctor"**

**select pa.first\_name,pa.last\_name,'Patient' as Role from patients pa**

**union all**

**select da.first\_name,da.last\_name,'Doctor' as Role from doctors da ;**

**select \* from patients;**

**select \* from doctors;**

**select \* from admissions;**

**-- 26. Show all allergies ordered by popularity. Remove NULL values from the query.**

**select allergies,count(patient\_id) as popularity from patients group by allergies having allergies is not null order by popularity desc ;**

**-- 27. Show all patient's first\_name, last\_name, and birth\_date who were born in the 1970s decade. Sort the list starting from the earliest birth\_date.**

**select first\_name,last\_name,birth\_date from patients where birth\_date between '1970-01-01' and '1979-12-31' order by birth\_date desc;**

**-- 28. We want to display each patient's full name in a single column. Their last\_name in all upper letters must appear first, then first\_name in all lower case letters. Separate the last\_name and first\_name with a comma. Order the list by the first\_name in descending order EX: SMITH,jane**

**select concat (UPPER(last\_name),",",LOWER(first\_name)) as result from patients order by LOWER(first\_name) desc;**

**select upper(last\_name) from patients;**

**select lower(first\_name) from patients;**

**-- 29. Show the province\_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000**

**select province\_id,sum(height) from patients group by province\_id having sum(height)>=7000;**

**-- 30. Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'**

**select last\_name,weight from patients where last\_name = 'Maroni';**

**select max(weight)from patients where last\_name = 'Maroni';**

**select min(weight) from patients where last\_name = 'Maroni';**

**select max(weight) - min(weight) from patients where last\_name = 'Maroni';**

**-- 31. Show all of the days of the month (1-31) and how many admission\_dates occurred on that day. Sort by the day with most admissions to least admissions.**

**select admission\_date,count(admission\_date) as admission\_count from admissions group by admission\_date order by admission\_count desc;**

**-- 32. Show all of the patients grouped into weight groups. Show the total number of patients in each weight group. Order the list by the weight group descending. e.g. if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.**

**select patient\_id,first\_name,last\_name,weight from patients where weight between '100' and '110';**

**select w.weight as weight\_range,count(\*) as number\_of\_weight from**

**(select case**

**when weight between 0 and 49 then '0-50'**

**when weight between 50 and 100 then '51-100'**

**else '101-200' end as weight from patients) as w group by w.weight order by number\_of\_weight desc**

**-- 33. Show patient\_id, weight, height, isObese from the patients table. Display isObese as a boolean 0 or 1. Obese is defined as weight(kg)/(height(m). Weight is in units kg. Height is in units cm.**

**select patient\_id,first\_name,last\_name,height,weight,weight\*100/height as bmi,**

**case**

**when weight\*100/height > 30 then '1'**

**else '0' end as isobese from patients;**

**select patient\_id,first\_name,last\_name,height,weight,weight\*100/height as bmi from patients;**

**-- 34. Show patient\_id, first\_name, last\_name, and attending doctor's specialty. Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'. Check patients, admissions, and doctors tables for required information.**

**select \* from patients;**

**select \* from admissions;**

**select \* from doctors;**

**select pa.patient\_id,pa.first\_name,pa.last\_name,ad.diagnosis,ad.attending\_doctor\_id,doc.first\_name,doc.last\_name from patients as pa**

**inner join admissions as ad on pa.patient\_id=ad.patient\_id**

**inner join doctors as doc on doc.doctor\_id=ad.attending\_doctor\_id where ad.diagnosis='Epilepsy' and doc.first\_name='Lisa';**

**select \* from admissions where diagnosis='Epilepsy';**

**-- 35. All patients who have gone through admissions, can see their medical documents on our site. Those patients are given a temporary password after their first admission. Show the patient\_id and temp\_password**

**select patient\_id,concat(patient\_id,length(last\_name),date\_format(birth\_date,"%Y")) as tem\_password from patients;**

**CHALLENGES FACED:**

* **I am not able to update the null values in the data basics. So as per the mentor guidance, I supposed to write the query and add it in the documents.**
* **The concept related to Nested select are challenging.**
* **The concept which use based on case are new and difficult to understand.**
* **Query related to grouping based on BMI (weight in kgs/height in m) are able to get the in conceptual thinking. While implement it takes time to grouping based on the requirement. I was a really a good to question which I came across.**