SQL Practice Questions

EASY

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 does not have allergies. (null).

```
SELECT first_name, last_name
FROM patients
where allergies is 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
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'.

```
update patients
set allergies = "NKA"
where allergies is null;
```

6. Show first name and last name concatinated into one column to show their full name.

```
select concat(first_name, " ", last_name) as full_name
from patients;
```

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

```
select first_name, last_name, province_name
from patients
   join province_names ON patients.province_id = province_names.province_id;
```

8. Show how many patients have a birth_date with 2010 as the birth year.

```
select count(birth_date)
from patients
where birth_date like "%2010%";
```

9. Show the first_name, last_name, and height of the patient with the greatest height.

```
select first_name, last_name, max(height)
from patients
group by first_name, last_name
order by max(height) desc
limit 1;
```

OR if you want to use AS:

```
select first_name, last_name, max(height) as height
from patients
group by first_name, last_name
order by height desc
limit 1;
```

10. Show all columns for patients who have one of these patient_ids: 1,45,534,879,1000

```
select *
from patients
where patient_id in (1,45,534,879,1000);
```

11. Show the total number of admissions.

```
select count(admission_date)
from admissions;
```

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 patient id and the total number of admissions for patient_id 579.

```
select patient_id, count(admission_date)
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 distinct(city)
    from patients
    where province_id = "NS";
    ```

15. Write a query to find the first_name, last name and birth date of patients who has height greater than 160 and weight greater than 70.

    ```
    select first_name, last_name, birth_date
    from patients
    where height > 160 and weight > 70;
    ```

16. Write a query to find list of patients first_name, last_name, and allergies where allergies are not null and are from the city of 'Hamilton'

    ```
    select first_name, last_name, allergies
    from patients
    where allergies is not null and city = "Hamilton";
    ```

    ==========

MEDIUM

17. Show unique birth years from patients and order them by ascending.

    ```
    select distinct(year(birth_date)) as birth_year
    from patients
    order by birth_year;
    ```

18. 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.

    ```
    select first_name
    from patients
    group by first_name
    having count(first_name ="Leo") = 1;
    ```

19. Show patient_id and first_name from patients where their first_name start and ends with 's' and is at least 6 characters long.

    ```
    select patient_id, first_name
    ```

```
from patients
where first_name like "s%" and first_name like "%s" and first_name like "%_____%";
OR

select patient_id, first_name
from patients
where first_name like "s%s" and first_name like "%_____%";

OR

SELECT patient_id, first_name
FROM patients
WHERE first_name LIKE "s____%s";
```

20. Show patient_id, first_name, last_name from patients whos diagnosis is 'Dementia'. Primary diagnosis is stored in the admissions table.

```
select p.patient_id, p.first_name, p.last_name
from patients as p
        join admissions as a
  on p.patient_id = a.patient_id
where diagnosis = "Dementia";
```

21. 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 len(first_name), first_name asc;
```

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

```
select count(gender = "M") as Male,
          count(gender = "F") as Female
from patients;
```

23. Show first and last name, allergies from patients which have allergies to either 'Penicillin' or 'Morphine'. Show results ordered ascending by allergies then by first_name then by last_name.

```
select first_name ,last_name, allergies
from patients
where allergies = "Penicillin" or allergies = "Morphine"
order by allergies, first_name, last_name;
```

24. Show patient_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

```
select patient_id, diagnosis
from admissions
group by patient_id, diagnosis
having count(patient_id = diagnosis) > 1;
```

25. 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(*) as number_of_patients
from patients
group by city
order by number_of_patients desc, city;
```

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

```
select first_name, last_name, "Patient" as role
from patients
union all
select first_name, last_name, "Doctor" as role
from doctors;
```

27. Show all allergies ordered by popularity. Remove NULL values from query.

```
select allergies, count(*) as popular_allergies
from patients
where allergies is not null
group by allergies
order by popular_allergies desc;
```

28. 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 like "%197%"
order by birth_date asc;
```

OR

```
select first_name, last_name, birth_date
from patients
where Year(birth_date) between 1970 and 1979
order by birth_date asc;
```

29. 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 decending order. EX: SMITH,jane

```
select concat(upper(last_name), "," ,lower(first_name)) as full_name
from patients
order by first_name desc;
```

30. 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;
```

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

```
select   (max(weight) - min(weight)) as weight_diff
from patients
where last_name = "Maroni";
```

32. 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 day(admission_date) as day_num, count(patient_id) as num_of_addmission
from admissions
group by day_num
order by num_of_addmission Desc;
```

33. Show all columns for patient_id 542's most recent admission_date.

```
select *
from admissions
where patient_id = 542
order by admission_date desc
limit 1;
```

34. Show patient_id, attending_doctor_id, and diagnosis for admissions that match one of the two criteria: (A). patient_id is an odd number and attending_doctor_id is either 1, 5, or (B). attending_doctor_id contains a 2 and the length of patient_id is 3 characters.

```
select patient_id, attending_doctor_id, diagnosis
from admissions
where patient_id % 2 = 1 and attending_doctor_id in (1,5,19)
        or attending_doctor_id like "%2%" and len(patient_id) = 3;
```

35. Show first_name, last_name, and the total number of admissions attended for each doctor. Every admission has been attended by a doctor.

```
select first_name, last_name, count(admission_date) as admissions_attended
from admissions a
        join doctors d
        on a.attending_doctor_id = d.doctor_id
group by doctor_id;
```

36. For each doctor, display their id, full name, and the first and last admission date they attended.

```
select doctor_id,
        concat("first_name", " ", "last_name") as full_name,
        min(admission_date) as first_date_attended,
        max(admission_date) as last_date_attended
from admissions a
        join doctors d
        on a.attending_doctor_id = d.doctor_id
group by doctor_id;
```

37. Display the total amount of patients for each province. Order by descending.

```
select pr.province_name, count(p.patient_id) as total_patients
from patients as p
        join province_names as pr
        on p.province_id = pr.province_id
group by pr.province_name
order by total_patients desc;
```

38. For every admission, display the patient's full name, their admission diagnosis, and their doctor's full name who diagnosed their problem.

```
select
concat(p.first_name, " ", p.last_name) as patient_full_name,
a.diagnosis,
concat(d.first_name, " ", d.last_name) as doc_full_name
from patients as p
        join admissions as a
                on p.patient_id = a.patient_id
         join doctors as d
                on d.doctor_id = a.attending_doctor_id;
```

39. display the first name, last name and number of duplicate patients based on their first name and last name.

```
select first_name, last_name, count(*) as num_of_duplicates
from patients
group by first_name, last_name
having count(*) > 1;
```

40. Display patient's full name, height in the units feet rounded to 1 decimal, weight in the unit pounds rounded to 0 decimals, birth_date, gender non abbreviated. Convert CM to feet by dividing by 30.48. Convert KG to pounds by multiplying by 2.205.

```
select
concat(first_name, " ", last_name) as patient_full_name,
round((height/30.48), 1) as height,
round((weight*2.205), 0) as weight,
birth_date,
        case
                when gender = "M" then "Male"
                 when gender = "F" then "Female"
        end as gender
from patients;
```

41. Show patient_id, first_name, last_name from patients who do not have any records in the admissions table. (Their patient_id does not exist in any admissions.patient_id rows.)

```
select p.patient_id, p.first_name, p.last_name
from patients as p
        Left join admissions as a
        on p.patient_id = a.patient_id
where a.patient_id is null;
```

==========

HARD

42. Show all of the patients grouped into weight groups. Show the total amount of patients in each weight group. Order the list by the weight group decending. For example, if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

```
select (weight/10) * 10 as weight_group, count(*) as no_of_patients_in_grp
from patients
group by weight_group
order by weight_group desc;
```

43. 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)2) >= 30. Weight is in units kg. Height is in units cm.

```
// Comment:  To convert height (CM) to height (M): divide the height by 100.00
(height/100.00) //

select patient_id, weight, height,
Case
        when weight/power(height/100.00,2) > 30 then 1
        else 0
End as isObese
from patients;
```

44. 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 p.patient_id, p.first_name, p.last_name, d.specialty
from patients as p
        join admissions as a
                on p.patient_id = a.patient_id
         join doctors as d
                on d.doctor_id = a.attending_doctor_id
where a.diagnosis = "Epilepsy" and d.first_name = "Lisa";
```

45. 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. The password must be the following, in order: (A). patient_id (B). the numerical length of patient's last_name (C). year of patient's birth_date.

```
select distinct(p.patient_id),
concat(p.patient_id,len(p.last_name),year(p.birth_date)) as temp_password
from patients as p
        join admissions as a
        on p.patient_id = a.patient_id;
```

46. Each admission costs $50 for patients without insurance, and $10 for patients with insurance. All patients with an even patient_id have insurance. Give each patient a 'Yes' if they have insurance, and a 'No' if they don't have insurance. Add up the dmission_total cost for each has_insurance group.

```
select
case
        when patient_id % 2 = 0 then "Yes"
        else "No"
end as has_insurance,

sum(case
            when patient_id % 2 = 0 then 10
            else 50
        end) as cost_as_per_insurance_availability
from admissions
group by has_insurance;
```

47. Show the provinces that has more patients identified as 'M' than 'F'. Must only show full province_name.

```
select pn.province_name
from patients as p
        join province_names as pn
  on p.province_id = pn.province_id
group by province_name
having sum(case
            when p.gender = "M" then 1 else 0
          end) >
       sum(case
            when p.gender = "F" then 1 else 0
          end);
```

48. We are looking for a specific patient. Pull all columns for the patient who matches the following criteria:- First_name contains an 'r' after the first two letters.- Identifies their gender as 'F'- Born in February, May, or December- Their weight would be between 60kg and 80kg- Their patient_id is an odd number- They are from the city 'Kingston'.

```
select *
from patients
where
        first_name like "__r%" and
        gender = "F" and
        month(birth_date) in (2, 5, 12) and
        weight between 60 and 80 and
        patient_id % 2 = 1 and
        city = "Kingston";
```

49. Show the percent of patients that have 'M' as their gender. Round the answer to the nearest hundreth number and in percent form.

```
select
concat(round((sum(case when gender = "M" then 1 else 0 end) *100.00 / count(*)), 2),
"%") as male_percentage
 from patients;
```

50. For each day display the total amount of admissions on that day. Display the amount changed from the previous date.

```
SELECT admission_date,
        COUNT(admission_date) AS admission_count,
        COUNT(admission_date) - LAG(COUNT(admission_date)) OVER (ORDER BY
       admission_date) AS admission_count_change
FROM admissions
GROUP BY admission_date;
```

51. Sort the province names in ascending order in such a way that the province 'Ontario' is always on top.

```
SELECT province_name
FROM province_names
ORDER BY (province_name = "Ontario") desc, province_name asc;
```

52. We need a breakdown for the total amount of admissions each doctor has started each year. Show the doctor_id, doctor_full_name, specialty, year, total_admissions for that year.

```
select d.doctor_id,
        concat(d.first_name, " ", d.last_name) as Doc_full_name,
        d.specialty,
        year(a.admission_date) as the_year,
        count(*) as total_admissions_started
from admissions as a
      join doctors as d
      on a.attending_doctor_id = d.doctor_id
group by d.doctor_id, the_year;
```

By Ajay Dimri: [LinkedIn]