

SQL Question and Answers.

Easy

1. Show first name, last name, and gender of patients who's gender is 'M'.

a. `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).

a. `SELECT first_name,last_name FROM patients where allergies is null;`

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

a. `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).

a. `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'

a. `Update patients set allergies = 'NKA' where allergies is null;`

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

a. `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. Example: 'Ontario' instead of 'ON'

a. `select p.first_name,p.last_name,prov.province_name FROM patients as p JOIN province_names as prov where p.province_id = prov.province_id;`

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

a. `select count(*) from patients where Year(birth_date) = 2010;`

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

a. `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.

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

11. Show the total number of admissions.
a. `select count((patient_id)) from admissions;`
12. Show all the columns from admissions where the patient was admitted and discharged on the same day.
a. `select * from admissions where admission_date = discharge_date;`
13. Show the patient id and the total number of admissions for patient_id 579.
a. `select patient_id, count(*) as total_admission 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'?
a. `select distinct(city) as unique_cities 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.
a. `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 from city 'Hamilton' where allergies is not null.

- a. `select first_name, last_name, allergies from patients where city = 'Hamilton' and allergies is not NULL;`
- 17. Based on cities where our patient lives in, write a query to display the list of unique city starting with a vowel (a, e, i, o, u). Show the result order in ascending by city.
 - a. `SELECT distinct(city) FROM Customers WHERE City LIKE '[aeiou]%' order by city;`

Medium

1. 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.
 - a) `select distinct(first_name)
from patients
group by first_name
having count(*) = 1;`
2. Show unique birth years from patients and order them by ascending.
 - a) `select distinct(Year(birth_date)) as byear from patients order by byear;`

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

a) `select patient_id,first_name from patients
where first_name like 's%s' and
LEngth(first_name)>5;`

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

a) `select p.patient_id, p.first_name, p.last_name
from patients p LEFT JOIN admissions a where
p.patient_id = a.patient_id and a.diagnosis =
'Dementia';`

5. Display every patient's first_name. Order the list by the length of each name and then by alphabetically.

a) `select first_name from patients order by
length(first_name),first_name asc;`

6. 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.

a) `select count(gender) from patients group by
gender;`

b) `SELECT (SELECT count(*) FROM patients WHERE
gender='M') AS male_count, (SELECT count(*)`

```
FROM patients WHERE gender='F') AS  
female_count;
```

7. 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.

```
a) select first_name, last_name, allergies from  
patients where allergies in  
( 'Penicillin', 'Morphine' ) order by  
allergies, first_name, last_name;
```

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

```
a) select patient_id, diagnosis from admissions  
group by patient_id, diagnosis having  
Count(diagnosis) > 1 ;
```

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

```
a) select city , count(patient_id) as num_patients  
from patients group by city order by  
num_patients desc, city;
```

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

```
a) select first_name, last_name, 'Patient' as role
    from patients
    UNION all
    select first_name, last_name, 'Doctor' from
    doctors ;
```

--- union all because it gives distinct values.

11. Show all allergies ordered by popularity.

Remove NULL values from query.

```
a) select allergies, count(allergies) as
    total_diagnosis from patients where allergies is
    not null group by allergies order by
    total_diagnosis DESC;
```

12. 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.

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

13. 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

```
a) select  
    CONCAT(upper(last_name),',',lower(first_name)  
    ) as new_name_format from patients order by  
    first_name desc;
```

14. 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.

```
a) select province_id, sum(height) as sumheight  
    from patients group by province_id having  
    sumheight>7000;
```

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

```
a) select max(weight) - min(weight) from patients  
    where last_name = 'Maroni';
```

16. 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.


```
a) SELECT
    DAY(admission_date) AS day_number,
    COUNT(*) AS number_of_admissions
FROM admissions
GROUP BY day_number
ORDER BY number_of_admissions DESC
```

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

```
a) select * from admissions where patient_id = 542
AND admission_date = (SELECT
max(admission_date)
FROM admissions
WHERE patient_id = 542
);
```

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

a) Select patient_id, attending_doctor_id, diagnosis from admissions where (mod(patient_id, 2) <> 0 AND attending_doctor_id in (1,5,19)) or (attending_doctor_id like '%2%' and length(patient_id) = 3);

19. 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.

a) select first_name || ' ' || last_name ,
round(height/30.48,1),
round(weight*2.205,0),birth_date,case when
gender= 'M' then 'MALE' when gender= 'F' then
'FEMALE' End as Gender1 from patients;

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

a) select
doctors.first_name,doctors.last_name,count(ad
missions.attending_doctor_id) from doctors
JOIN admissions where

```
admissions.attending_doctor_id =  
doctors.doctor_id group by attending_doctor_id  
;
```

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

```
a)select d.doctor_id,Concat(d.first_name,'  
,d.last_name), min(a.admission_date),  
max(a.admission_date) from doctors d join  
admissions a on d.doctor_id =  
a.attending_doctor_id group by d.doctor_id;
```

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

```
a)select pro.province_name , count(p.province_id)  
as patient_count from province_names pro join  
patients p where p.province_id =  
pro.province_id group by pro.province_id order  
by patient_count desc ;
```

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

```
a)select p.first_name || ' ' || p.last_name,  
a.diagnosis , d.first_name || ' ' || d.last_name  
from patients p JOIN admissions a JOIN doctors
```

d where p.patient_id = a.patient_id and
a.attending_doctor_id = d.doctor_id;

24. display the number of duplicate patients based on their first_name and last_name.

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

HARD

1. 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.

a. `select count(patient_id), floor(weight/10) *10 as weight_group from patients group by weight_group order by weight_group desc;`

2. Show patient_id, weight, height, isObese from the patients table.

Display isObese as a boolean 0 or 1.

Obese is defined as $\text{weight(kg)} / (\text{height(m)}^2) \geq 30$.

weight is in units kg.

height is in units cm.

a. `select patient_id, weight, height, case when (weight/power(height,2))*power(10,4) >=30 then 1 else 0 end from patients;`

3. 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.

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

4. 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: 1. patient_id 2. the numerical length of patient's last_name 3. year of patient's birth_date

```
a. select p.patient_id ,  
       FLOOR(p.patient_id || length(p.last_name) || year(p.birth_date)) as temp_password from  
       patients p JOIN admissions a where p.patient_id  
       = a.patient_id group by p.patient_id;
```

5. 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 admission_total cost for each has_insurance group.

```
a. select case when patient_id %2 = 0 then 'Yes'  
       else 'No' End has_insurance, case when  
       patient_id %2 = 0 then count(*)*10 else  
       Count(*)*50 end from admissions group by  
       has_insurance;
```

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

```
a. SELECT pr.province_name FROM patients AS pa
JOIN province_names AS pr ON pa.province_id =
pr.province_id GROUP BY pr.province_name
HAVING COUNT( CASE WHEN gender = 'M'
THEN 1 END) > COUNT( CASE WHEN gender = 'F'
THEN 1 END);
```

7. 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'

```
a. select *
from patients
where
first_name like '__r%'
AND gender = 'F'
```

```
and weight between 60 and 80
and patient_id % 2 != 0
and city = 'Kingston';
```

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

```
a. select ROUND(((select COunt(*) from patients
  where gender = 'M')/cast(count(*) as
  float))*100,2) || '%' from patients;
```

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

```
a. SELECT
  admission_date,
  count(admission_date) as admission_day,
  count(admission_date) -
  LAG(count(admission_date)) OVER() AS
  admission_count_change
FROM admissions
  group by admission_date;
```

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


```
a. select province_name from province_names  
   Order by case when province_name = 'Ontario'  
   then 0 else 1 End;
```