Show first name, last name, and gender of patients who's gender is 'M' [EASY]

```
select
        first_name,
   last_name,
   gender
from
        patients
where
        gender = "M"
```

Show first name and last name of patients who does not have allergies. (null) [EASY]

```
select
        first_name,
   last_name
from
        patients
where
        allergies is null
```

Show first name of patients that start with the letter 'C'[EASY]

```
select
        first_name
from
        patients
where
        first_name like 'C%'
```

Show first name and last name of patients that weight within the range of 100 to 120 (inclusive) [EASY]

```
select
        first_name,
   last_name
from
        patients
where
        weight between 100 and 120
```

Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'[EASY]

```
update patients
set allergies = 'NKA'
where allergies is null
```

Show first name and last name concatinated into one column to show their full name.[EASY]

```
select
        concat(first_name, " ", last_name)
FROM
        patients
```

note that select first_name || ‘ ‘ || last_name from patients; also works. It depends on the type of database we’re pulling from.

Show first name, last name, and the **full** province name of each patient. [EASY]

Example: 'Ontario' instead of 'ON'

```
select
        pat.first_name,
    pat.last_name,
    prov.province_name
from
        patients as pat
join
        province_names AS prov on prov.province_id = pat.province_id
```

Show how many patients have a birth_date with 2010 as the birth year. [EASY]

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

Honestly saying YEAR(birth_date) = 2010; would’ve just as good. I was just being kind of lazy in this regard. Especially since the year is first. I treated it as a string and not a date object. Better to see things objectively.

Show the first_name, last_name, and height of the patient with the greatest height.[EASY]

```
select
        first_name,
    last_name,
    max(height)
 from
        patients
```

Show all columns for patients who have one of the following patient_ids:
1,45,534,879,1000 [EASY]

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

Show the total number of admissions[EASY]

```
select
        COUNT(*) as total_admissions
 from
        admissions
```

Show all the columns from admissions where the patient was admitted and discharged on the same day. [EASY]

```
select
        *
 from
        admissions
where
        admission_date = discharge_date
```

Show the patient id and the total number of admissions for patient_id 579.[EASY]
```
select
        patient_id,
   count(admission_date)
from
        admissions
where patient_id = 579
```

Based on the cities that our patients live in, show unique cities that are in province_id 'NS'?[EASY]
```
select
        distinct(city)
from
        patients
where
        province_id = "NS"
```

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
```

Write a query to find list of patients first_name, last_name, and allergies from Hamilton where allergies are not null [EASY]
```
select
        first_name,
   last_name,
   allergies
 from
        patients
where
        city = "Hamilton"
   and
        allergies is not null
```

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. [EASY]

```
select
        distinct(city)
from
        patients
where
        city like 'a%'
   or
   city like 'e%'
   or
   city like 'i%'
   or
   city like 'o%'
   or
   city like 'u%'
order by city asc
```

Show unique birth years from patients and order them by ascending. [MEDIUM]

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

we can also use group by birthyear. Though I'm not a fan of going that route. The `GROUP BY` statement is often used with aggregate functions (`COUNT()`, `MAX()`, `MIN()`, `SUM()`, `AVG()`) to group the result-set by one or more columns.

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. [MEDIUM]

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

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

```
select
        patient_id,
   first_name
```

```
from
	patients
where
	first_name like "s%____%s"
```
note that

```
 first_name LIKE 's%s'
  AND len(first_name) >= 6;
```
also works.
Show patient_id, first_name, last_name from patients whos diagnosis is 'Dementia'.

Primary diagnosis is stored in the admissions table. [MEDIUM]
```
select
	p.patient_id,
    p.first_name,
    p.last_name
from
	patients p
join admissions ad on ad.patient_id = p.patient_id
where ad.diagnosis = "Dementia"
```

note that
```
WHERE patient_id IN (
    SELECT patient_id
    FROM admissions
    WHERE diagnosis = 'Dementia'
  );
```
also works.

Display every patient's first_name.
Order the list by the length of each name and then by alphabetically [MEDIUM]
```
select
	first_name
from
	patients
order by
	len(first_name) asc, first_name asc
```

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. [MEDIUM]
```
select
	(select
	count(*)
    from
	patients
    where
	gender = "M") as M,
    (select
	count(*)
```

```
    from
        patients
    where
        gender = "F") as F
```

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 [MEDIUM]

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

Show patient_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.[MEDIUM]

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

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

```
select
    city,
    count(*) as num_patients
from patients
group by city
order by num_patients desc,city asc
```

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

```
select
    first_name,
    last_name,
    'Patient'
from
    patients
union all
select
    first_name,
    last_name,
    'Doctor'
from
```

```
        doctors
```

Show all allergies ordered by popularity. Remove NULL values from query.[MEDIUM]
```
select
        allergies,
   count(allergies)
from
        patients
where
        allergies not null
group by
        allergies
order by
        count(allergies) DESC
```

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. [MEDIUM]
```
select
        first_name,
   last_name,
   birth_date
from
        patients
where
        year(birth_date) between 1970 and 1979
order by birth_date asc
```

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 [MEDIUM]
```
select
        concat(upper(last_name), ",", LOWER(first_name)) as name
from
        patients
order by
        first_name desc
```

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. [MEDIUM]
```
select
        province_id,
   sum(height)
from
        patients
group by
        province_id
having
        sum(height) >= 7000
```

```
order by
        sum(height)
```

Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni' [MEDIUM]
```
select
        max(weight) - min(weight)
from
        patients
where
        last_name = "Maroni"
```

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.[MEDIUM]

```
select
        day(admission_date) as day_number,
    count(admission_date) as number_of_admissions
from
        admissions
group by
        day(admission_date)
order by
        number_of_admissions desc
```

Show all columns for patient_id 542's most recent admission_date.[MEDIUM]

```
select
        *
from
        admissions
where
        patient_id = 542
group by
        patient_id
having
        admission_date = max(admission_date)
```

Show patient_id, attending_doctor_id, and diagnosis for admissions that match one of the two criteria:
1. patient_id is an odd number and attending_doctor_id is either 1, 5, or 19.
2. attending_doctor_id contains a 2 and the length of patient_id is 3 characters.
[MEDIUM]
```
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
    )
```

Show first_name, last_name, and the total number of admissions attended for each doctor.

Every admission has been attended by a doctor. [MEDIUM]

```
select
        first_name,
   last_name,
   count(attending_doctor_id) as attended
from
        admissions ad
join
        doctors doc
   on
   doc.doctor_id = ad.attending_doctor_id
group by
        attending_doctor_id
```

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

Select [MEDIUM]

```
        doc.doctor_id,
   concat(doc.first_name, ' ', doc.last_name),
   min(ad.admission_date) as first_admission_date,
   max(ad.admission_date) as last_admission_date
from
        doctors doc
join
        admissions ad
   on
   ad.attending_doctor_id = doc.doctor_id
group by
        doctor_id
```

Display the total amount of patients for each province. Order by descending. [MEDIUM]

```
select
        prov.province_name,
        count(pat.patient_id) as patient_count
from
```

```
        patients pat
join
        province_names prov
   on
   prov.province_id = pat.province_id
group by
        prov.province_name
order by
        patient_count desc
```

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

```
select
        pat.first_name || ' ' || pat.last_name as patient_name,
   ad.diagnosis,
   doc.first_name || ' ' || doc.last_name as doctor_name
from
        patients as pat
join
        admissions as ad
   on
   ad.patient_id = pat.patient_id
join
        doctors as doc
   on
   doc.doctor_id = ad.attending_doctor_id
```

---

display the number of duplicate patients based on their first_name and last_name.[MEDIUM]

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

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.[MEDIUM]

```
select
```

```
        first_name || ' ' || last_name as patient_name,
        round(height/30.48, 1) as height,
    round(weight * 2.205, 0) as weight,
    birth_date,
    case
        when gender = "M" then "MALE"
       else "FEMALE"
    end as gender_type
from
        patients
```

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.[HARD]
```
select
        count(*) as patients_in_group,
    floor(weight / 10) * 10 as weight_group
from
        patients
group by
        weight_group
order by
        weight_group desc
```

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.[HARD]
```
select
        patient_id,
    weight,
    height,
    case
        when weight/(power(height*0.01,2)) >= 30
        then 1
       else 0
    end as isObese
from
        patients
```

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. [HARD]

```
select
        pat.patient_id,
    pat.first_name,
    pat.last_name,
    doc.specialty
from
        patients as pat
join
        admissions as ad
    on
    ad.patient_id = pat.patient_id
join
        doctors as doc
    on
    doc.doctor_id = ad.attending_doctor_id
where
        ad.diagnosis = "Epilepsy" and doc.first_name = "Lisa"
```

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[HARD]

```
select
        distinct(pat.patient_id),
    round(
      concat(pat.patient_id,
      len(pat.last_name),
      year(pat.birth_date)),
     0) as temp_password
from
        patients as pat
join
        admissions as ad
    on
    ad.patient_id = pat.patient_id
```

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.[HARD]

```
select
        case
        when patient_id % 2 = 1 then "No"
      else "Yes"
   end as has_insurance,
   sum(case
        when patient_id % 2 = 1 then 50
      else 10
   end) cost_after_insurance
from admissions

group by
        has_insurance
```

Show the provinces that has more patients identified as 'M' than 'F'. Must only show full province_name [HARD]
```
select
        prov.province_name
from
        province_names as prov
join
        patients as pat
   on
   pat.province_id = prov.province_id
group by
        prov.province_name
having
        count( case when gender = "M" then 1 end) > count( case when gender = "F" then 1 end)
```

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' [HARD]

```
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"
```

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

```
select
        concat(round(100*avg(gender="M"),2), "%") as percent_of_male_patients
from patients
```

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

```
select
        admission_date,
    count(admission_date) as admission_day,
    count(admission_date) - lag(count(admission_date)) over (order by admission_date) as
admission_count_change
from
        admissions
group by
        admission_date
```

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

```
select
        province_name
from
        province_names
order by
        province_name = "Ontario" desc,
    province_name asc
```