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.

```
SELECT
 COUNT(*) AS patients_in_group,
 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 weight(kg)/(height(m)^2) >= 30.
weight is in units kg.
height is in units cm.

```
SELECT patient_id, weight, height,
(CASE
WHEN weight/(POWER(height/100.0,2)) >= 30 THEN
1
ELSE
0
END) AS isObese
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'

```
select p.patient_id , p.first_name, p.last_name ,d.specialty
FROM patients p JOIN admissions a
ON p.patient_id = a.patient_id
join doctors d
ON a.attending_doctor_id = d.doctor_id
where a.diagnosis = 'Epilepsy' AND d.first_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

```
select a.patient_id ,
concat(a.patient_id ,len(p.last_name),year(p.birth_date)) AS temp_password
FROm admissions a join patients p
ON a.patient_id = p.patient_id
group by a.patient_id
```

```
SELECT
  DISTINCT P.patient_id,
  CONCAT(
    P.patient_id,
    LEN(last_name),
    YEAR(birth_date)
  ) AS temp_password
FROM patients P
  JOIN admissions A ON A.patient_id = 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.

```
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_after_insurance
FROM admissions
GROUP BY has_insurance;
```

```
select has_insurance ,
case
```

```
WHEN has_insurance = 'YES' THEN count(has_insurance)*10
ELSE count(has_insuarnce )*50
END As cost_after_insurance
(
select
CASE when patient_id % 2 == 0 THen 'YES ' ELSE 'No' END AS has_insurance
FROM admissions)
group by has_insurance


select has_insurance,sum(admission_cost) as admission_total
from
(
  select patient_id,
  case when patient_id % 2 = 0 then 'Yes' else 'No' end as has_insurance,
  case when patient_id % 2 = 0 then 10 else 50 end as admission_cost
  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

```
select pn.province_name FROM patients p
JOIN province_names pn
ON p.province_id = pn.province_id
group by pn.province_name
HAVING COUNT(CASE WHEN p.gender = 'M' THEN 1 END)  > count(case WHEN p.gender = 'F'
THEN 1 END) ;

SELECT province_name
FROM (
   SELECT
     province_name,
     SUM(gender = 'M') AS n_male,
     SUM(gender = 'F') AS n_female
   FROM patients pa
     JOIN province_names pr ON pa.province_id = pr.province_id
   GROUP BY province_name
 )
WHERE n_male > n_female
```

```
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
  SUM(gender = 'M') > SUM(gender = 'F');
```

7)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 END)
                /CAST(count(*) AS float),4 ) *100,'%')
          as per FROM patients
```

8)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_day,
COUNT(admission_date) - LAG(COUNT(admission_date))
OVER (order by admission_date) AS Change
FROM admissions group by admission_date
```

9)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
(case WHEN province_name  ='Ontario' THEN 0 ELSE 1 END),
province_name;

select province_name
from province_names
order by
  (not province_name = 'Ontario'),
  province_name
```

10)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 !=0 and city = 'Kingston'
```