```
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)
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'
update patients
SET allergies = 'NKA'
where allergies IS NULL

--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

--Show first name, last name, and the full province name of each patient.
--Example: 'Ontario' instead of 'ON'
SELECT
first_name, last_name, province_name
FROM patients p
JOIN province_names pn ON p.province_id=pn.province_id
--Show how many patients have a birth_date with 2010 as the birth year.
SELECT
COUNT(patient_id) AS total_patients
FROM patients
WHERE YEAR(birth_date) = 2010
--Show the first_name, last_name, and height of the patient with the greatest height.
SELECT
first_name,last_name, height
FROM patients
ORDER BY height DESC
LIMIT 1
--or
SELECT
first_name, last_name, height
FROM patients
WHERE height = (
  SELECT max(height)
  FROM patients
  )
```

```
--Show all columns for patients who have one of the following patient_ids:
--1,45,534,879,1000
SELECT
*
FROM patients
WHERE patient_id IN (1,45,534,879,1000)

--Show the total number of admissions
SELECT
COUNT(*) as total_number
FROM admissions

--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

--Show the patient id and the total number of admissions for patient_id 579.
SELECT
p.patient_id,
count(*) AS total_admissions
from patients p
JOIN admissions a ON p.patient_id=a.patient_id
WHERE p.patient_id = 579

--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'

--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 where allergies are not null and are from the city of 'Hamilton'
SELECT
first_name,last_name,allergies
from patients
where allergies NOT NULL AND city ='Hamilton'

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

```
SELECT
distinct year(birth_date) AS unique_birth_year
FROM patients
ORDER BY unique_birth_year asc

--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) = 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.
SELECT
patient_id,first_name
from patients
where first_name LIKE 's____%s'
--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 p
JOIN admissions a ON p.patient_id=a.patient_id
WHERE diagnosis ='Dementia'

--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) 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.
SELECT
(SELECT count(*) from patients WHERE gender = 'M') AS male_count,
(SELECT count(*) from patients WHERE gender = 'F') AS female_count

--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 IN ('Penicillin', 'Morphine')
ORDER BY
  allergies,
  first_name,
  last_name;
--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(*)>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.

SELECT
city,
COUNT(patient_id) 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"
SELECT
first_name,last_name, 'Patient' AS role
FROM patients
UNION all
SELECT
first_name,last_name, 'Doctor' AS role
FROM doctors
--Show all allergies ordered by popularity. Remove NULL values from query.
select
allergies,
count(*) AS total_diagnosis
from patients p
where allergies IS not NULL
GROUP BY allergies
ORDER BY total_diagnosis 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.
```

```
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
SELECT
CONCAT(UPPER(last_name),',',LOWER(first_name)) AS full_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.

SELECT
province_id,
SUM(height) AS total_high
FROM patients
group by province_id
having SUM(height) >= 7000

--Show the difference between the largest weight and smallest weight for patients
--with the last name 'Maroni'
SELECT
(MAX(weight) - min(weight)) AS diif_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.
select
DAY(admission_date) AS day_number,
count(*) AS num_of_admissions
FROM admissions
GROUP BY day_number
ORDER BY num_of_admissions DESC
--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

--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.
SELECT
  patient_id,
  attending_doctor_id,
  diagnosis
FROM admissions
WHERE
  (
    attending_doctor_id IN (1, 5, 19)
    AND patient_id % 2 != 0
  )
  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.
SELECT
first_name,last_name,
count(*) AS num_admissions
FROM admissions a
JOIN doctors d ON a.attending_doctor_id=d.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
d.doctor_id,
concat(first_name,' ',last_name) as full_name,
MIN(admission_date) AS first_admission_date,
MAX(admission_date) AS last_admission_date
FROM doctors d
JOIN admissions a ON a.attending_doctor_id=d.doctor_id
group by d.doctor_id

--Display the total amount of patients for each province. Order by descending.
SELECT
  province_name,
  COUNT(*) as patient_count
FROM patients pa
  join province_names pr on pr.province_id = pa.province_id
group by pr.province_id
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.
SELECT
concat(d.first_name,' ',d.last_name) AS full_name_doctor,
diagnosis,
concat(p.first_name,' ',p.last_name) AS full_name_patient
FROM patients p
JOIN admissions a ON a.patient_id=p.patient_id
JOIN doctors d ON a.attending_doctor_id=d.doctor_id

--display the first name, last name and number of duplicate patients
--based on their first name and last name.
--Ex: A patient with an identical name can be considered a duplicate.
SELECT
first_name,last_name,
COUNT(*) AS num_duplicates
FROM patients
GROUP BY first_name,last_name
HAVING COUNT(*) > 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.

select
    concat(first_name, ' ', last_name) AS 'patient_name',
    ROUND(height / 30.48, 1) as 'height "Feet"',
    ROUND(weight * 2.205, 0) AS 'weight "Pounds"', birth_date,
CASE
          WHEN gender = 'M' THEN 'MALE'
  ELSE 'FEMALE'
END AS 'gender_type'
from patients

--Show patient_id, first_name, last_name from patients whose does not have
--any records in the admissions table. (Their patient_id does not exist in
--any admissions.patient_id rows.)
SELECT
patient_id,first_name,last_name
FROM patients
WHERE patient_id not in (SELECT patient_id
                         FROM admissions)


--Display a single row with max_visits, min_visits, average_visits
```

```
--where the maximum, minimum and average number of admissions per day
--is calculated. Average is rounded to 2 decimal places.

  select
          max(number_of_visits) as max_visits,
          min(number_of_visits) as min_visits,
  round(avg(number_of_visits),2) as average_visits
from (
  select admission_date, count(*) as number_of_visits
  from admissions
  group by admission_date
)
--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
--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;
--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 AS attending_doctor_speciality
FROM patients p
join admissions a ON p.patient_id=a.patient_id
```

```
JOIN doctors d ON d.doctor_id=a.attending_doctor_id
WHERE d.first_name = 'Lisa' AND a.diagnosis = 'Epilepsy'
--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
  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
--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 insurance,
    SUM(CASE WHEN patient_id % 2 = 0 THEN 10
       ELSE 50
       END) AS cost_after_insurance
FROM admissions
GROUP BY insurance

--Show the provinces that has more patients identified as 'M' than 'F'.
--Must only show full province_name
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);
--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';
 --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(
    100.0 * COUNT(CASE WHEN gender = 'M' THEN 1 END) / COUNT(*),
    2),'%') AS percent_male
FROM patients;
--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 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.
select province_name
from province_names
order by
  (case when province_name = 'Ontario' then 0 else 1 end),--0 to be first
  province_name
--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 as doctor_id,
  CONCAT(d.first_name,' ', d.last_name) as doctor_name,
  d.specialty,
  YEAR(a.admission_date) as selected_year,
  COUNT(*) as total_admissions
FROM doctors as d
  LEFT JOIN admissions as a ON d.doctor_id = a.attending_doctor_id
GROUP BY
```

```
  doctor_name,
  selected_year
ORDER BY doctor_id, selected_year
```