Laporan Praktikum
Mata Kuliah Sistem Manajemen Basis Data
SQL Practice

Dwi Cahya Ramadani

25 April 2022

1 Soal Easy

- Show first name, last name, and gender of patients who's gender is 'M'
  Query :

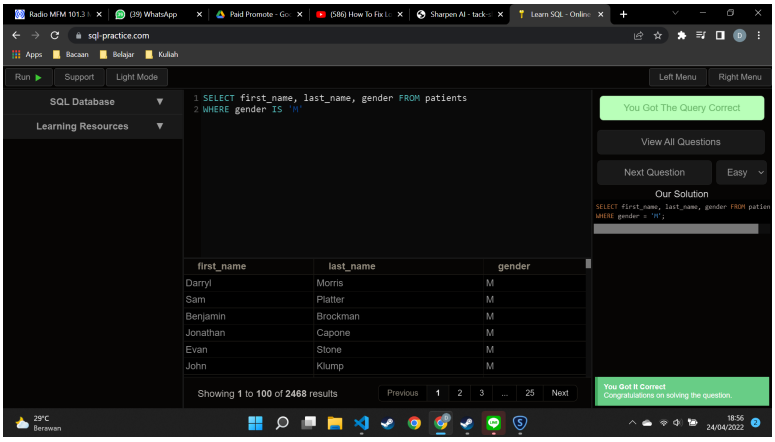
Listing 1: Easy - 1

```
1 SELECT first_name, last_name, gender FROM patients
2 Where gender is 'M'
```

Penjelasan :

- Baris 1 : Mengambil data first_name, last_name, dan gender dari tabel patients
- Baris 2 : Dengan kondisi dimana gendernya adalah "M"

Screenshot :



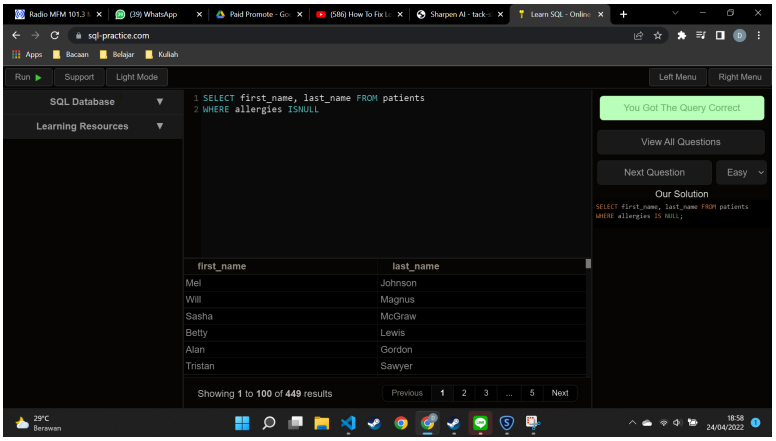
- Show first name and last name of patients who does not have allergies (null)
  Query :

Listing 2: Easy - 2

```
1 SELECT first_name, last_name FROM patients
2 WHERE allergies isnull
```

  - Baris 1 : Mengambil data first_name dan last_name dari tabel patients
  - Baris 2 : Dengan kondisi dimana allergies dari pasien bernilai null (tidak memiliki allergi)

  Screenshot :



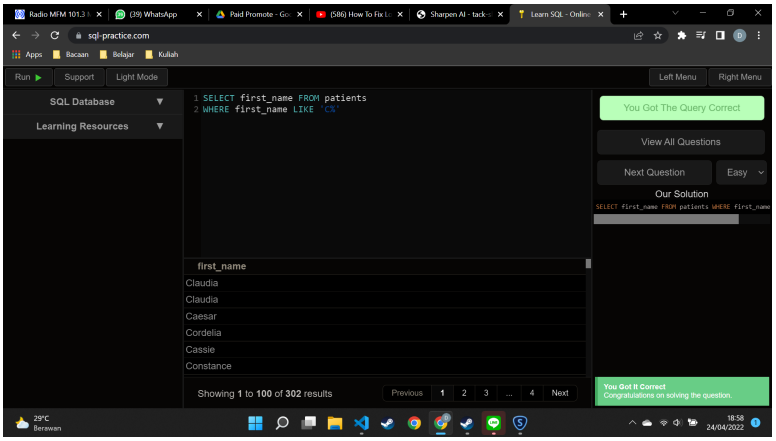
- Show first name of patients that start with the letter 'C'
  Query :

Listing 3: Easy - 3

```
1 SELECT first_name FROM patients
2 WHERE first_name LIKE 'C%'
```

  - Baris 1 : Mengambil data first_name dari tabel patients
  - Baris 2 : Dengan kondisi dimana first_name dari pasien diawali dengan huruf "C"

Screenshot :



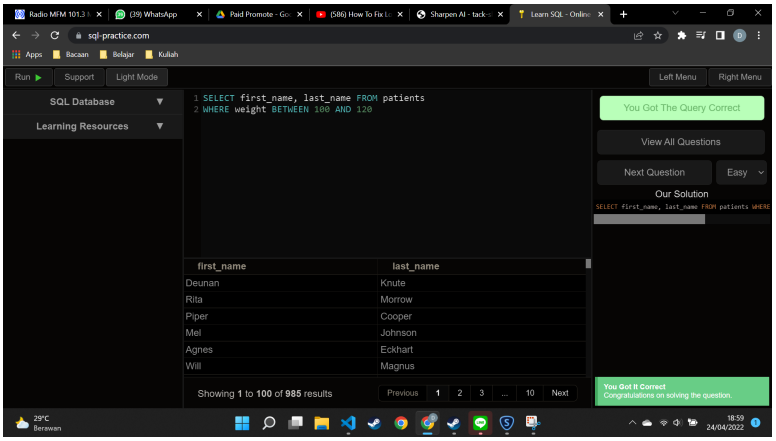
- Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)
  Query :

Listing 4: Easy - 4

```
1 SELECT first_name, last_name FROM patients
2 Where weight between 100 AND 120
```

  - Baris 1 : Mengambil data first_name dan last_name dari tabel patients
  - Baris 2 : Dengan kondisi dimana weight dari pasien bernilai diantara 100 dan 120

  Screenshot :



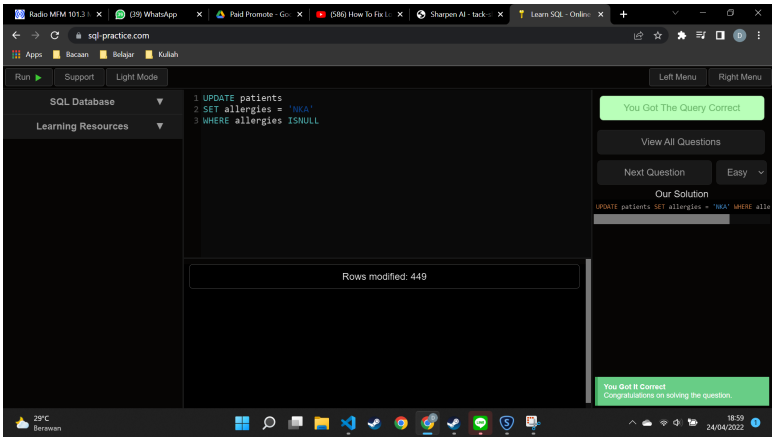
- Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'
  Query :

Listing 5: Easy - 5

```
1 update patients
2 set allergies = 'NKA'
3 where allergies isnull
```

  - Baris 1 : Mengupdate tabel patients
  - Baris 2 : Kolom allergies akan diisi dengan "NKA"
  - Baris 3 : Pada data dimana allergies dari pasien bernilai null (tidak memiliki allergi)

  Screenshot :



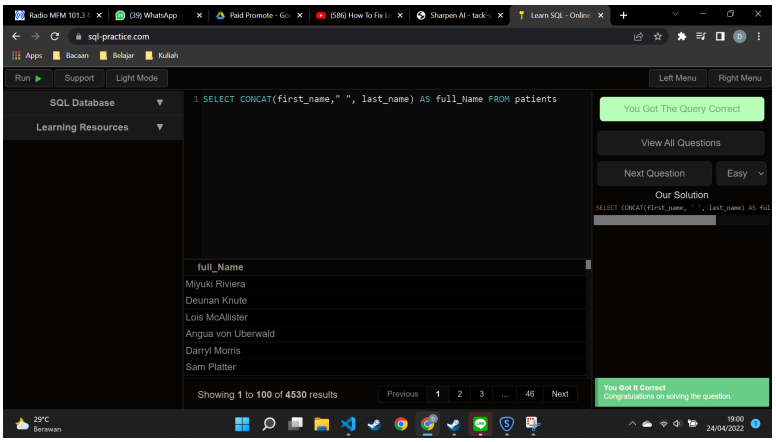
- Show first name and last name concatinated into one column to show their full name
  Query :

Listing 6: Easy - 6

```
1 SELECT concat(first_name," ", last_name) as full_Name FROM
      patients
```

  - Baris 1 : Mengambil data gabungan dari first_name dan last_name yang akan disebut sebagai full_name dari data pada tabel patients

Screenshot :



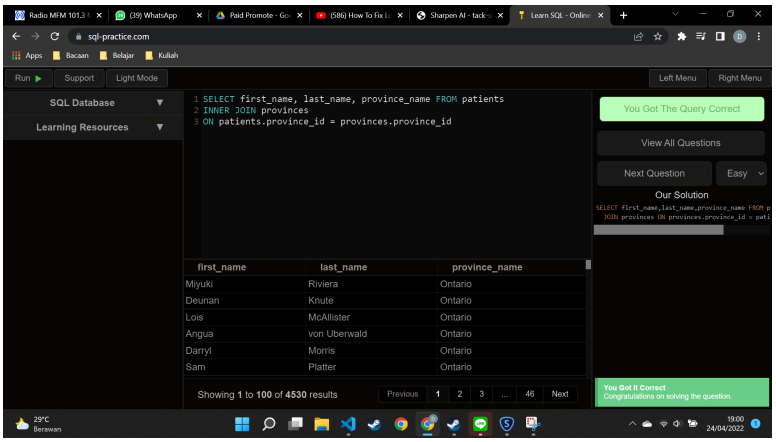
- Show first name, last name, and the full province name of each patient. Example: 'Ontario' instead of 'ON'
  Query :

Listing 7: Easy - 7

```
1 select first_name, last_name, province_name from patients
2 inner join provinces
3 on patients.province_id = provinces.province_id
```

  - Baris 1 : Mengambil data first_name, last_name, dan provincename dari tabel patients
  - Baris 2 : Menggabungkan tabel provinces sesuai dengan data tabel patients
  - Baris 3 : Dengan acuan province_id pada tabel patients merujuk pada province_id pada tabel provinces

Screenshot :



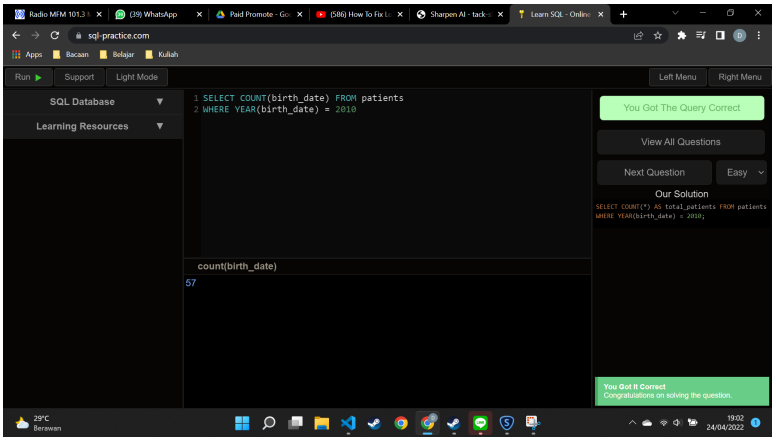
- Show how many patients have a birth_date with 2010 as the birth year
  Query :

Listing 8: Easy - 8

```
1 select count(birth_date) from patients
2 where year(birth_date) = 2010
```

  - Baris 1 : Mengambil banyaknya data pada kolom brthdate dari tabel patients
  - Baris 2 : Dengan kondisi dimana tahun birth_date dari pasien adalah 2010

Screenshot :



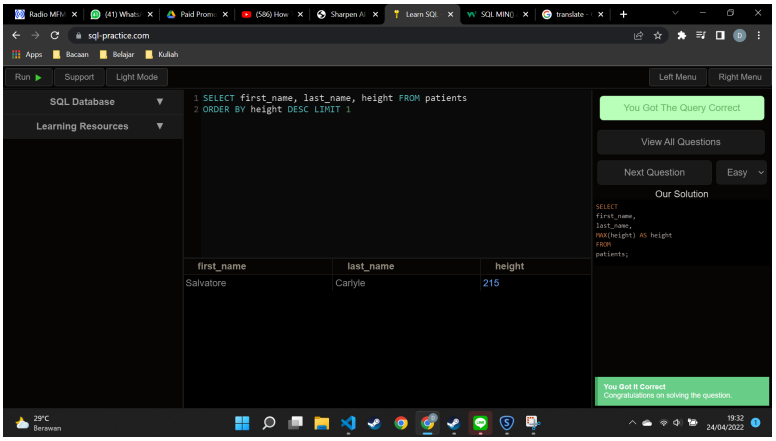
- Show the first_name, last_name, and height of the patient with the greatest height
  Query :

Listing 9: Easy - 9

```
1 select first_name, last_name, height from patients
2 order by height desc limit 1
```

  - Baris 1 : Mengambil data first_name, last_name, dan height dari tabel patients
  - Baris 2 : Mengurutkan data berdasarkan height secara descending (dari height terbesar), dan ambil 1 data pertama

  Screenshot :



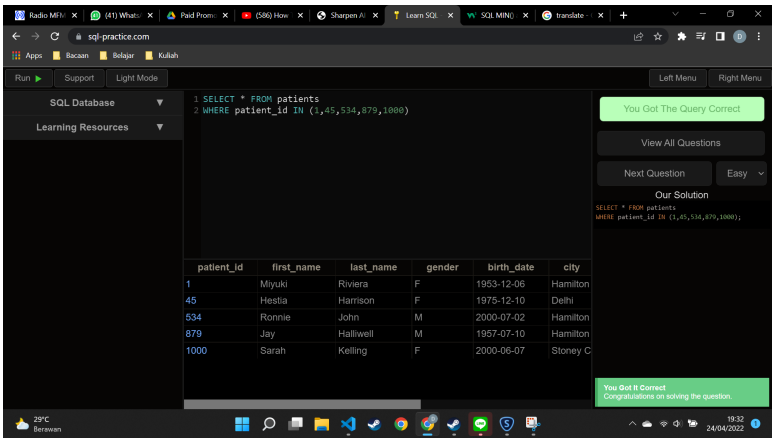
- Show all columns for patients who have one of the following patient_ids: 1,45,534,879,1000
  Query :

Listing 10: Easy - 10

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

  - Baris 1 : Mengambil semua data dari tabel patients
  - Baris 2 : Dengan kondisi dimana patient_id nya termasuk diantara 1, 45, 534, 879, 1000

Screenshot :



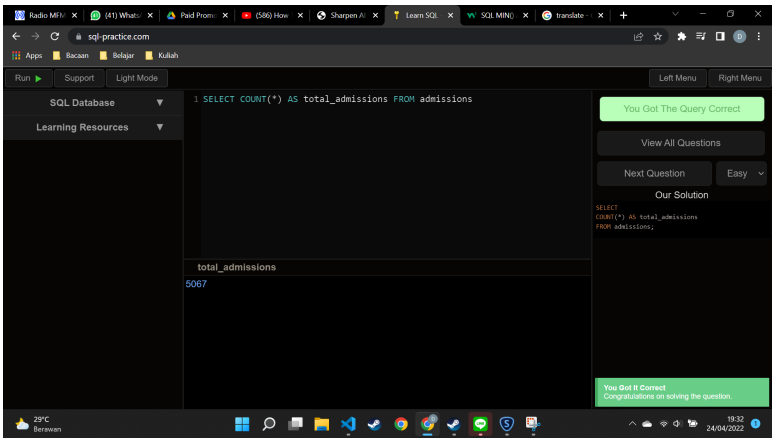
- Show the total number of admissions
  Query :

Listing 11: Easy - 11

```
1 select count(*) as total_admissions from admissions
```

  – Baris 1 : Mengambil banyaknya data dari admissions (akan disebut sebagai total_admissions)

Screenshot :



2 Soal Medium

- Show unique birth years from patients and order them by ascending
  Query :

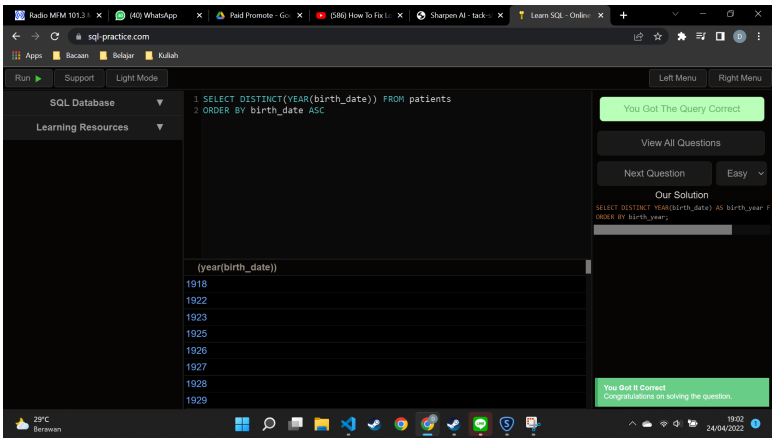
Listing 12: Medium - 1

```
1 select distinct(year(birth_date)) from patients
2 order by birth_date ASC
```

Penjelasan :

- Baris 1 : Mengambil data tahun yang unik (distinct) pada kolom birth_date dari tabel patients
- Baris 2 : Mengurutkan secara ascending berdasarkan birth_date

Screenshot :



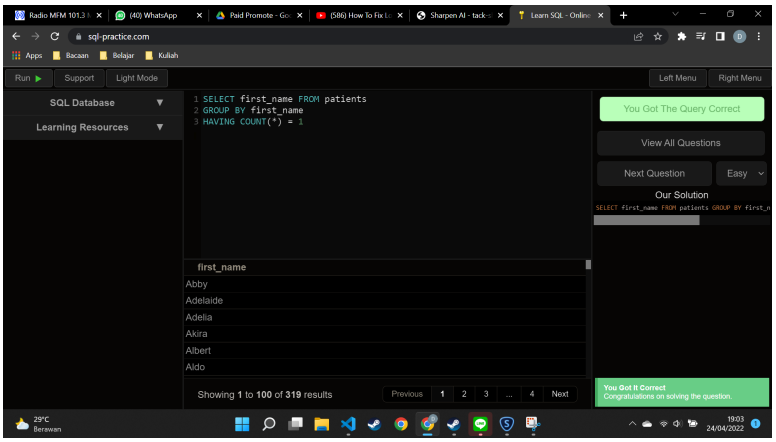
- Show unique first names from the patients table which only occurs once in the list. For example, if two or more people are named 'John' in the first_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output.
  Query :

Listing 13: Medium - 2

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

- Baris 1 : Mengambil data first_name dari tabel patients
- Baris 2 : Mengelompokkan data berdasarkan first_name
- Baris 3 : Mengambil data dengan jumlah data dalam kelompok adalah 1

Screenshot :



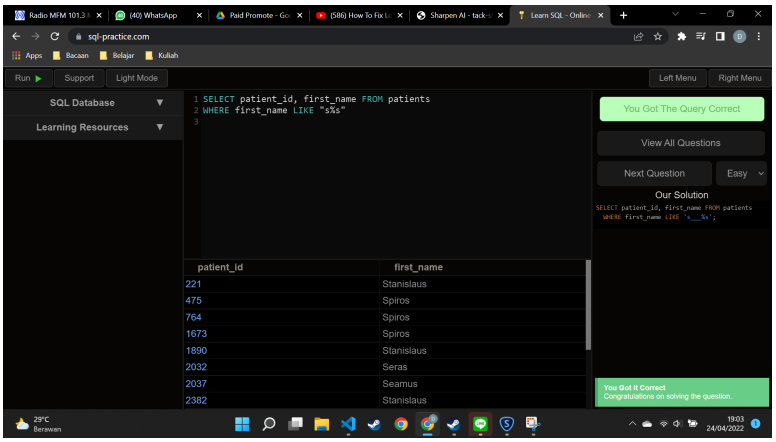
- Show patient_id and first_name from patients where their first_name start and ends with 's' and is atleast 5 characters long.
  Query :

Listing 14: Medium - 3

```
1 select patient_id, first_name from patients
2 where first_name Like "s%s"
```

  - Baris 1 : Mengambil data patient_id dan first_name dari tabel patients
  - Baris 2 : Dengan kondisi dimana first_name dari pasien diawali dengan huruf "s" yang diikuti minimal 3 karakter dan diakhiri dengan huruf "s"

Screenshot :



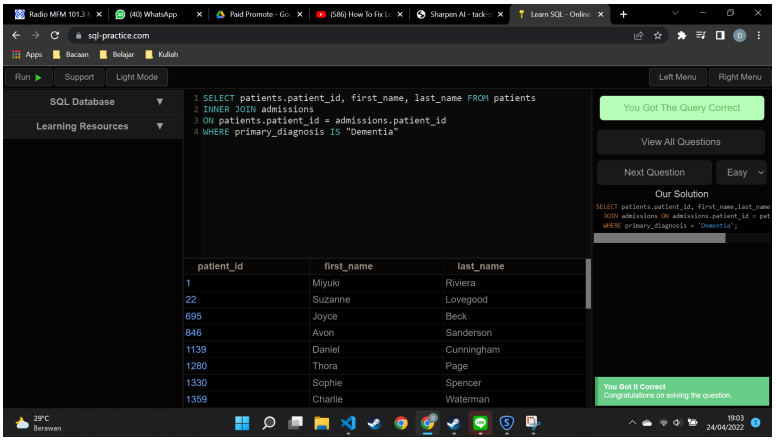
- Show patient_id, first_name, last_name from patients whos primary_diagnosis is 'Dementia'. Primary diagnosis is stored in the admissions table
  Query :

Listing 15: Medium - 4

```
1 select patients.patient_id, first_name, last_name from
      patients
2 inner join admissions
3 on patients.patient_id = admissions.patient_id
4 where primary_diagnosis is "Dementia"
```

  - Baris 1 : Mengambil data patient_id, first_name dan last_name dari tabel patients
  - Baris 2 : Menggabungkan tabel admissions sesuai data dari tabel patients
  - Baris 3 : Dengan acuan patient_id pada tabel patients merujuk pada patient_id pada tabel admissions
  - Baris 4 : Mengambil data dengan kondisi dimana primary_diagnosis dari pasien adalah "Dementia"

Screenshot :



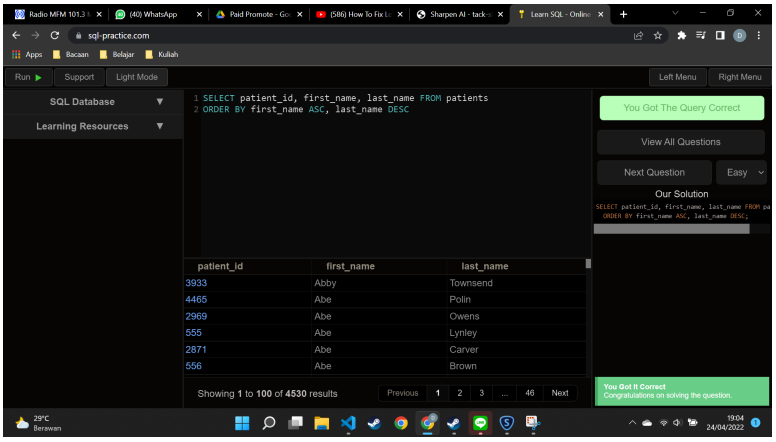
- Show patient_id, first_name, last_name from the patients table. Order the rows by the first_name ascending and then by the last_name descending Query :

Listing 16: Medium - 5

```
1 select patient_id, first_name, last_name from patients
2 order by first_name ASC, last_name DESC
```

  - Baris 1 : Mengambil data patient_id, first_name, dan last_name dari tabel patients
  - Baris 2 : Mengurutkan berdasarkan first_name secara ascending lalu mengurutkan berdasarkan last_name secara descending

Screenshot :



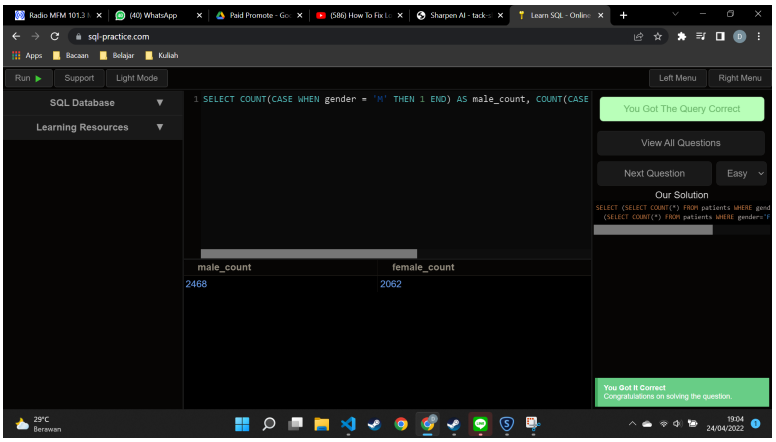
- Show the total amount of male patients and the total amount of female patients in the patients table
  Query :

Listing 17: Medium - 6

```
1 select count(case when gender = 'M' then 1 end) as male_count,
      count(case when gender = 'F' then 2 end) as female_count
    from patients
```

  – Baris 1 : Mengambil banyaknya data yang memiliki gender "M" (selanjutnya akan disebut male_count) dan banyaknya data yang memiliki gender "F" (selanjutnya akan disebut female_count) dari tabel patients

  Screenshot :



- 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.
  Query :

Listing 18: Medium - 7

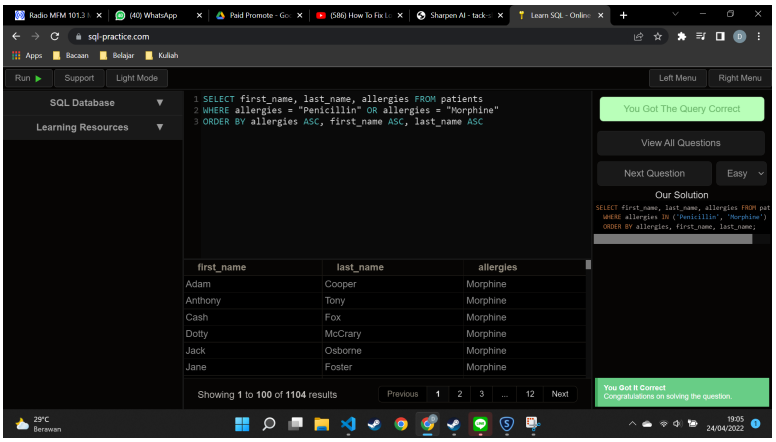
```
1 select first_name, last_name, allergies from patients
2 where allergies = "Penicillin" OR allergies = "Morphine"
3 order by allergies ASC, first_name ASC, last_name ASC
```

  – Baris 1 : Mengambil data first_name, last_name, dan allergies dari tabel patients

  – Baris 2 : Dengan kondisi dimana allergies yang dimiliki pasien adalah "Penicillin" atau "Morphine"

– Baris 3 : Mengurutkan data berdasarkan allergies secara ascending, lalu mengurutkan data berdasarkan first_name secara ascending, dan terakhir mengurutkan data berdasarkan last_name secara ascending

Screenshot :



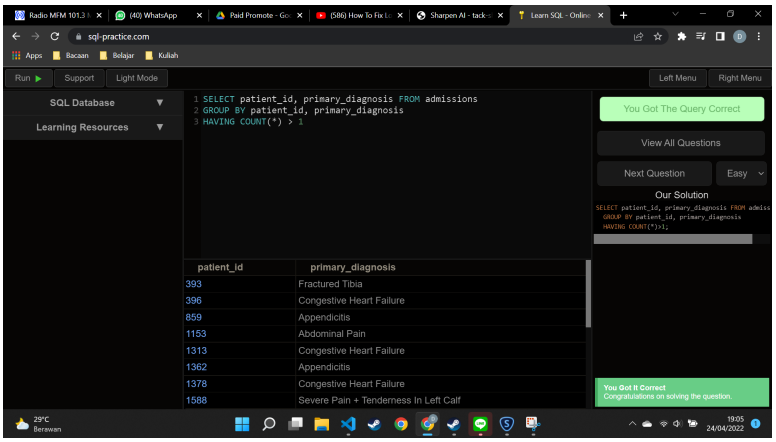
- Show patient_id, primary_diagnosis from admissions. Find patients admitted multiple times for the same primary_diagnosis
  Query :

Listing 19: Medium - 8

```
1 select patient_id, primary_diagnosis from admissions
2 group by patient_id, primary_diagnosis
3 having count(*) > 1
```

  – Baris 1 : Mengambil data patient_id dan primary_diagnosis dari tabel admissions
  – Baris 2 : Mengelompokkan berdasarkan patient_id, lalu Mengelompokkan berdasarkan primary_diagnosis
  – Baris 3 : Mengambil data dalam kelompok yang memiliki anggota lebih dari 1 (pasien yang sama terkena penyakit yang sama lebih dari sekali)

Screenshot :



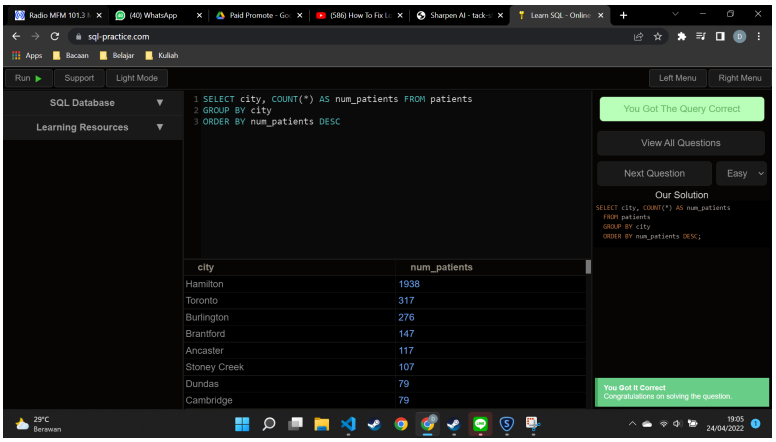
- Show the city and the total number of patients in the city in the order from most to least patients
  Query :

Listing 20: Medium - 9

```
1 select city, count(*) as num_patients from patients
2 group by city
3 order by num_patients DESC
```

  - Baris 1 : Mengambil data city dan banyaknya data (akan disebut sebagai num_patients) dari tabel patients
  - Baris 2 : Mengelompokkan data berdasarkan city
  - Baris 3 : Mengurutkan data berdasarkan num_patients secara descending

Screenshot :



- Show first name, last name and role of every person that is either patient or physician. The roles are either "Patient" or "Physician"
  Query :

Listing 21: Medium - 10

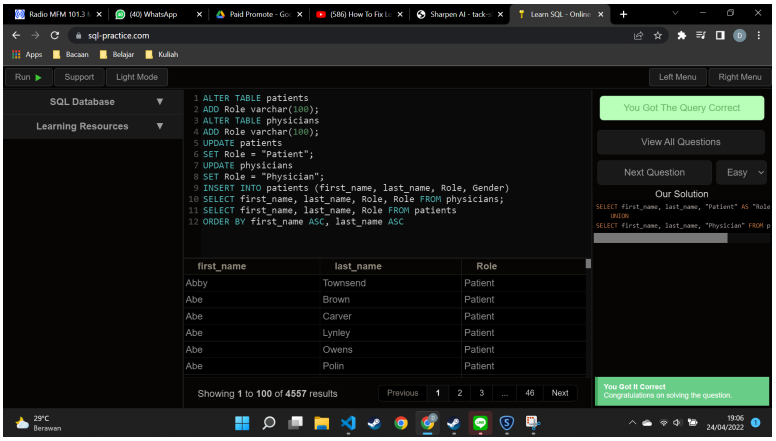
```
alter table patients
add Role varchar(100);
Alter table physicians
add Role varchar(100);
update patients
set Role = "Patient";
Update physicians
set Role = "Physician";
insert into patients (first_name, last_name, Role, Gender)
select first_name, last_name, Role, Role from physicians;
select first_name, last_name, Role from patients
order by first_name ASC, last_name ASC

'CARA LAMBAT'
```

  - Baris 1 : Mengambil tabel patients
  - Baris 2 : Menambahkan kolom Role sebagai tipe data varchar
  - Baris 3 : Mengambil tabel physicians
  - Baris 4 : Menambahkan kolom Role sebagai tipe data varchar
  - Baris 5 : Mengupdate tabel patients
  - Baris 6 : Mengubah nilai Role pada tabel patients dengan "Patient"
  - Baris 7 : Mengupdate tabel patients
  - Baris 8 : Mengubah nilai Role pada tabel physicians dengan "Physician"

- Baris 9 - 10 : Menambahkan nilai dari tabel physicians ke tabel patients
- Baris 11 : Mengambil data first_name, last_name, dan Role dari tabel patients
- Baris 12 : Mengurutkan berdasarkan first_name secara ascending, lalu mengurutkan berdasarkan last_name secara ascending

Screenshot :



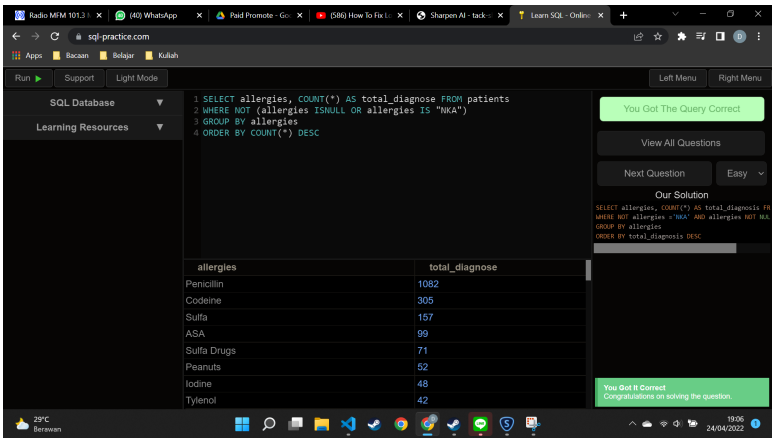
- Show all allergies ordered by popularity. Remove 'NKA' and NULL values from query
  Query :

Listing 22: Medium - 11

```
1 select allergies, Count(*) as total_diagnose from patients
2 where not (allergies isNULL OR allergies is "NKA")
3 group by allergies
4 order by count(*) DESC
```

- Baris 1 : Mengambil data allergies dan banyaknya data sebagai total_diagnose dari tabel patients
- Baris 2 : Dengan kondisi dimana allergies yang dimiliki adalah selain NKA dan tidak bernilai null
- Baris 3 : Mengelompokkan data berdasarkan allergies
- Baris 4 : Mengurutkan data berdasarkan banyaknya data dari tiap kelompok secara descending

Screenshot :



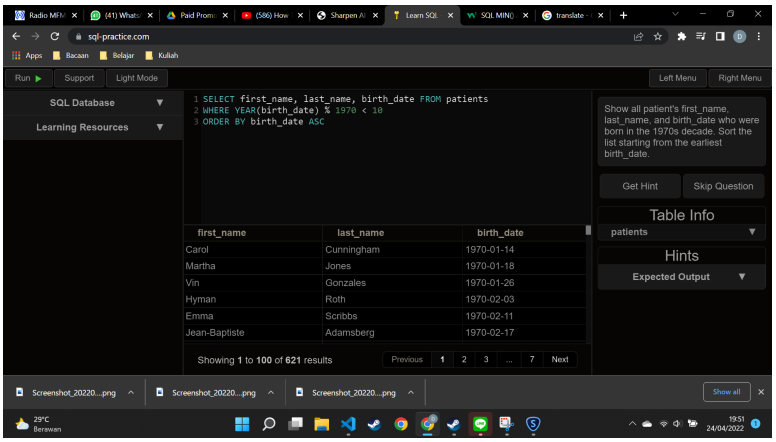
- Show all patient's first_name, last_name, and birth_date who were born in the 1970s decade. Sort the list starting from the earliest birth_date.
  Query :

Listing 23: Medium - 12

```
1 select first_name, last_name, height from patients
2 where year(birth_date) % 1970 < 10
3 order by birth_date asc
```

  - Baris 1 : Mengambil data first_name, last_name, dan height dari tabel patients
  - Baris 2 : Dengan kondisi dimana modulus dari tahun lahir dengan 1970 adalah kurang dari 10 (1970 - 1979)
  - Baris 3 : Mengurutkan data berdasarkan birth_date secara ascending

Screenshot :



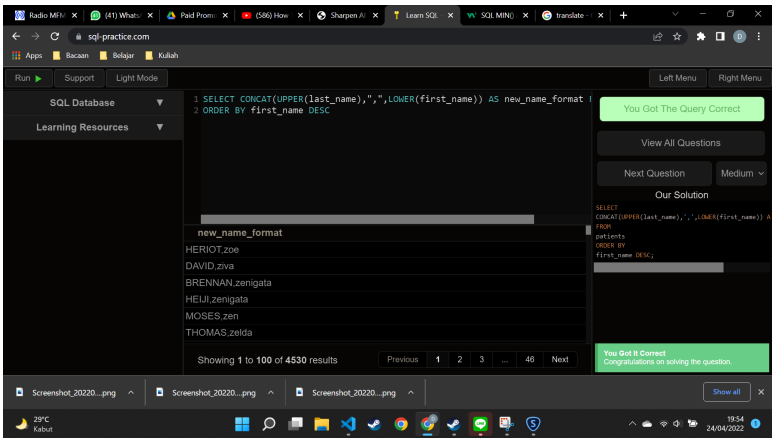
- We want to display each patient's full name in a single column. Their last_name in all upper letters must appear first, then first_name in all lower case letters. Separate the last_name and first_name with a comma. Order the list by the first_name in decending order. EX: SMITH,jane
  Query :

Listing 24: Medium - 13

```
1 select concat(Upper(last_name),",",lower(first_name)) as
      new_name_format from patients
2 order by first_name desc
```

  - Baris 1 : Mengambil data gabungan dari last_name yang diubah uppercase, tanda ",", dan first_name yang diubah lowercase (akan disebut sebagai new_name_format) dari tabel patients
  - Baris 2 : Mengurutkan data berdasarkan first_name secara descending

Screenshot :



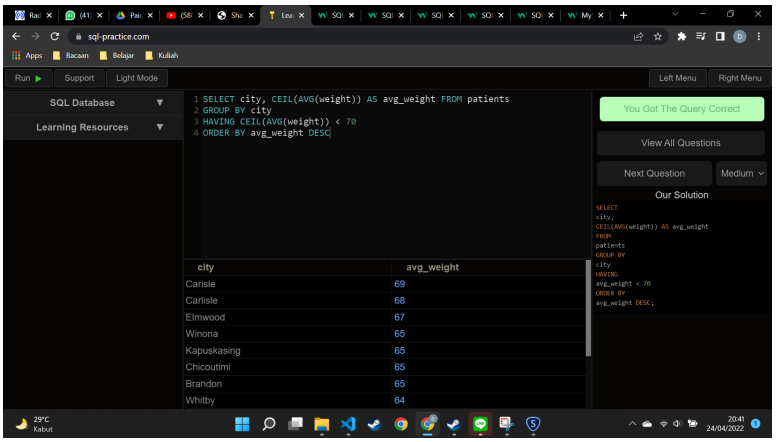
- Show the cities where the patient's average weight, rounded-up, is less than 70kg. Sort the list by highest to lowest avg_weight.
  Query :

Listing 25: Medium - 14

```
1 select city, ceil(avg(weight)) as avg_weight from patients
2 group by city
3 having ceil(avg(weight)) < 70
4 order by avg_weight desc
```

  - Baris 1 : Mengambil data city, pembulatan keatas dari rata-rata weight (akan disebut sebagai avg_weight) dari tabel patients
  - Baris 2 : Mengelompokkan data berdasarkan city
  - Baris 3 : Dengan kondisi dimana hasil pembulatan keatas dari rata-rata weight kurang dari 70 kg
  - Baris 4 : Mengurutkan data berdasarkan avg_weight secara descending

Screenshot :



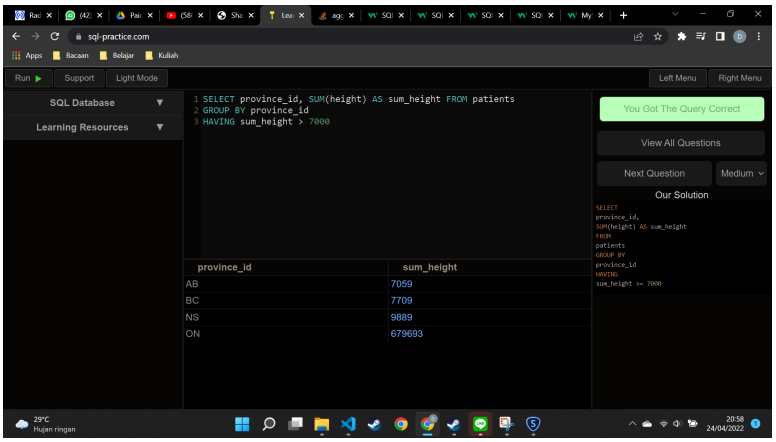
- Show the province_id(s) where the total sum of its patient's height is greater than or equal to 7,000.
  Query :

Listing 26: Medium - 15

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

  - Baris 1 : Mengambil data province_id, jumlah dari height (akan disebut sebagai sum_height) dari tabel patients
  - Baris 2 : Mengelompokkan data berdasarkan province_id
  - Baris 3 : Dengan kondisi dimana sum_height lebih dari 7000

Screenshot :



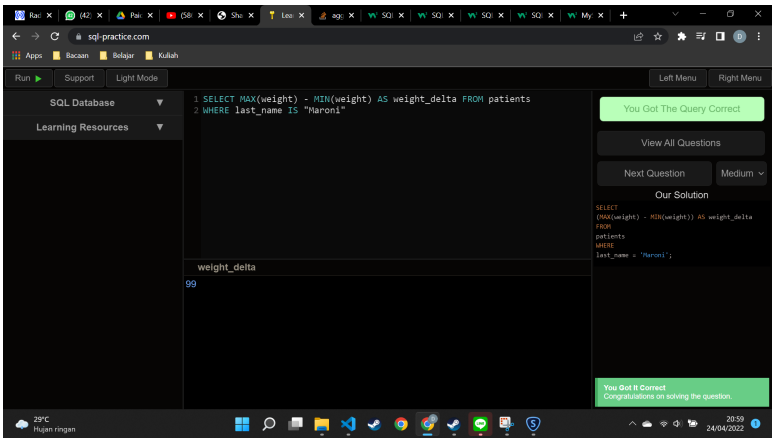
- Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'
  Query :

Listing 27: Medium - 16

```
1 select max(weight) - min(weight) as weight_delta from patients
2 where last_name is "Maroni"
```

  - Baris 1 : Mengambil data selisih dari nilai maximal dengan nilai minimal dari kolom weight (akan disebut sebagai weight_delta) dari tabel patients
  - Baris 2 : Dengan kondisi dimana last_name dari pasien adalah "Maroni"

Screenshot :



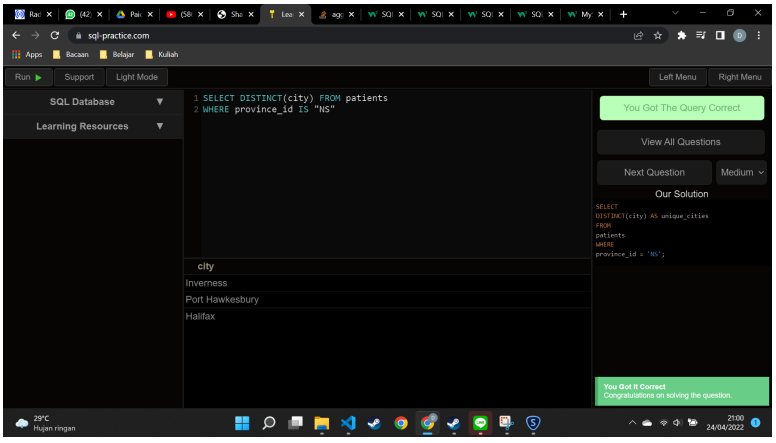
- Based on the cities that our patients live in, show unique cities that are in province_id 'NS'?
  Query :

Listing 28: Medium - 17

```
1 select distinct(city) from patients
2 where province_id is "NS"
```

  - Baris 1 : Mengambil data yang unik pada kolom city dari tabel patients
  - Baris 2 : Dengan kondisi dimana province_id dari pasien adalah "NS"

Screenshot :



3 Soal Hard

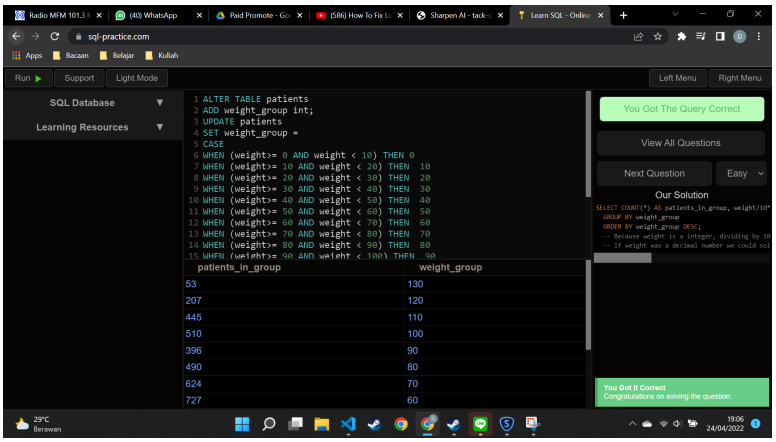
- Show all of the patients grouped into weight groups. Show the total amount of patients in each weight group. Order the list by the weight group decending. For example, if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc
  Query :

Listing 29: Hard - 1

```
1  alter table patients
2  add weight_group int;
3  update patients
4  set weight_group =
5  case
6  when (weight >= 0 and weight < 10) then 0
7  when (weight >= 10 and weight < 20) then  10
8  when (weight >= 20 and weight < 30) then  20
9  when (weight >= 30 and weight < 40) then  30
10 when (weight >= 40 and weight < 50) then  40
11 when (weight >= 50 and weight < 60) then  50
12 when (weight >= 60 and weight < 70) then  60
13 when (weight >= 70 and weight < 80) then  70
14 when (weight >= 80 and weight < 90) then  80
15 when (weight >= 90 and weight < 100) then  90
16 when (weight >= 100 and weight < 110) then 100
17 when (weight >= 110 and weight < 120) then 110
18 when (weight >= 120 and weight < 130) then 120
19 when (weight >= 130 and weight < 140) then 130
20 end;
21 select count(*) as patients_in_group, weight_group from
       patients
22 group by weight_group
23 order by weight_group DESC
24
25 'CARA LAMBAT'
```

  - Baris 1 : Mengambil data table patients
  - Baris 2 : Menambahkan kolom weight_group sebagai tipe data integer pada table patients
  - Baris 3 : Mengupdate table patients
  - Baris 4 - 20 : Mengubah nilai weight_group sesuai dengan berat pasien (menurut aturan dari soal)
  - Baris 21 : Mengambil banyaknya data (akan disebut sebagai patients_in_group) dan weight_group dari tabel patients
  - Baris 22 : Mengelompokkan data berdasarkan weight_group
  - Baris 23 : Mengurutkan data berdasarkan weight_group secara descending

Screenshot :



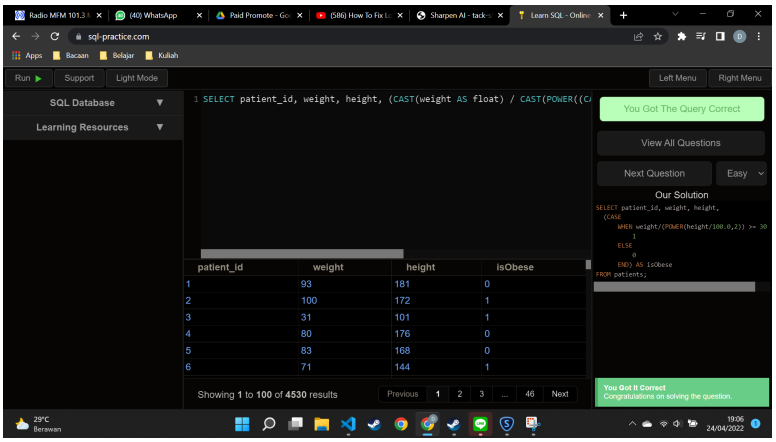
- Show patient_id, weight, height, isObese from the patients table. Display isObese as a boolean 0 or 1. Obese is defined as weight(kg)/(height(m)^2) >= 30. weight is in units kg. height is in units cm
  Query :

Listing 30: Hard - 2

```
select patient_id, weight, height, (cast(weight as float) /
    power((cast(height as float)/100),2) >=30) as isObese from
     patients
```

  - Baris 1 : Mengambil data patient_id, weight, height, dan $(weight/(height/100)^2)$ >= 30 (akan disebut sebagai isObese) dari tabel patients

Screenshot :



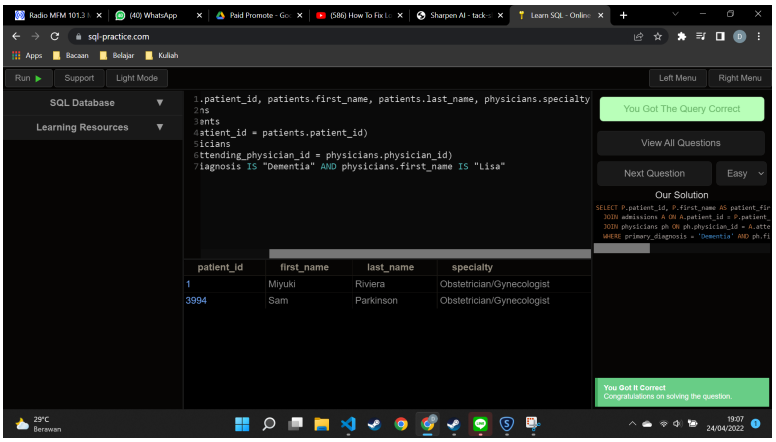
- Show patient_id, first_name, last_name, and attending physician's specialty. Show only the patients who has a primary_diagnosis as 'Dementia' and the physician's first name is 'Lisa'. Check patients, admissions, and physicians tables for required information.
  Query :

Listing 31: Hard - 3

```
1 select patients.patient_id, patients.first_name, patients.
      last_name, physicians.specialty
2 from ((admissions
3 inner join patients
4 on admissions.patient_id = patients.patient_id)
5 Inner join physicians
6 on admissions.attending_physician_id = physicians.physician_id
      )
7 where primary_diagnosis is "Dementia" and physicians.
      first_name is "Lisa"
```

  - Baris 1 : Mengambil data patient_id, first_name, last_name, dan specialty
  - Baris 2 - 4 : Dari tabel admissions yang digabung dengan tabel patients dengan acuan patient_id pada tabel admissions merujuk ke patient_id pada tabel patients
  - Baris 5 - 6 : Lalu digabung lagi dengan tabel physicians dengan acuan attending_physician_id pada tabel admissions merujuk ke physician_id pada tabel physicians
  - Baris 7 : Dengan kondisi dimana primary_diagnosis dari pasien adalah "Dementia" dan first_name dari physician adalah "Lisa"

Screenshot :



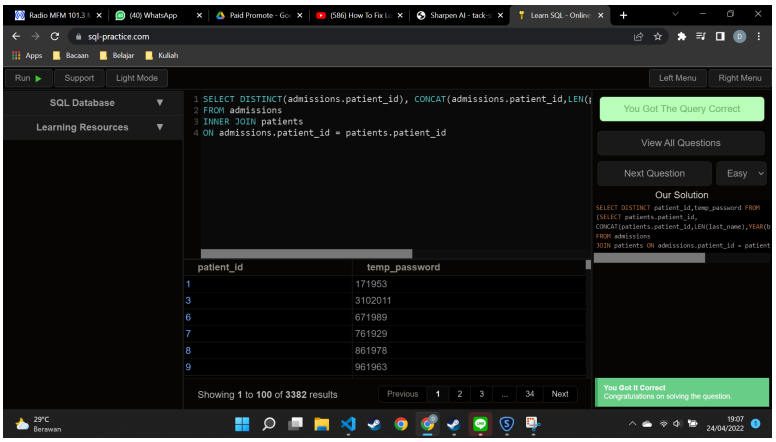
- All patients who have gone through admissions, can see their medical documents on our site. Those patients are given a temporary password after their first admission. Show the patient_id and temp_password. The password must be the following, in order: 1. patient_id 2. the numerical length of patient's last_name 3. year of patient's birth_date
  Query :

Listing 32: Hard - 4

```
1 select distinct(admissions.patient_id), concat(admissions.
      patient_id,LEN(patients.last_name),year(patients.
      birth_date)) as temp_password
2 from admissions
3 inner join patients
4 on admissions.patient_id = patients.patient_id
```

  - Baris 1 : Mengambil data unik dari kolom patient_id, gabungan dari patient_id, panjang last_name dan tahun lahir (akan disebut sebagai temp_password)
  - Baris 2 : Dari tabel admissions
  - Baris 3 : Menggabungkan tabel patients sesuai dengan data pada tabel admissions
  - Baris 4 : Dengan acuan patient_id pada tabel admissions merujuk ke patient_id pada tabel patients

Screenshot :



- Each admission costs 50 dollars for patients without insurance, and 10 dollars 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.
  Query :

Listing 33: Hard - 5

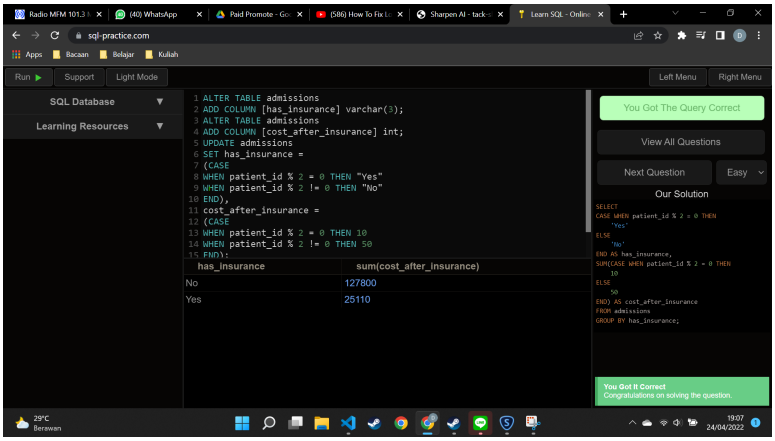
```
alter table admissions
add column [has_insurance] varchar(3);
alter table admissions
add column [cost_after_insurance] int;
update admissions
set has_insurance =
(case
when patient_id % 2 = 0 then "Yes"
when patient_id % 2 != 0 then "No"
end),
cost_after_insurance =
(case
when patient_id % 2 = 0 then 10
when patient_id % 2 != 0 then 50
end);
select has_insurance, sum(cost_after_insurance) from
    admissions
group by has_insurance

'CARA LAMBAT'
```

  - Baris 1 : Mengambil data tabel admissions
  - Baris 2 : Menambahkan kolom has_insurance sebagai tipe data varchar

- Baris 3 : mengambil data tabel admissions
- Baris 4 : Menambahkan kolom cost_after_insurance sebagai tipe data integer
- Baris 5 : Mengupdate tabel admissions
- Baris 6 - 10 : Mengubah nilai dari has_insurance sesuai dengan patient_id, ketika patient_id genap maka bernilai "Yes", ketika patient_id ganjil maka bernilai "No"
- Baris 11 - 15 : Mengubah nilai dari cost_after_insurance sesuai dengan patient_id, ketika patient_id genap maka bernilai 10, ketika patient_id ganjil maka bernilai 50
- Baris 16 : Mengambil data has_insurance dan jumlah dari cost_after_insurance dari tabel admissions
- Baris 17 : Mengelompokkan data berdasarkan has_insurance

Screenshot :



- Show the province that has more patients identified as 'M' than 'F'. Must only show full province_name
  Query :

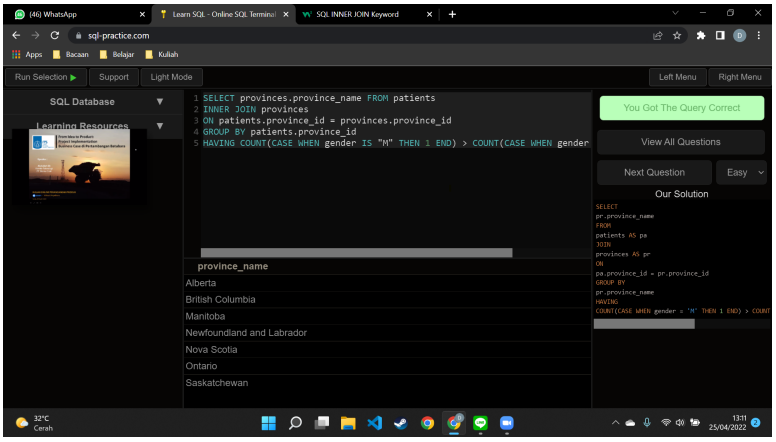
Listing 34: Hard - 6

```
select provinces.province_name from patients
inner join provinces
on patients.province_id = provinces.province_id
group by patients.province_id
having count(case when gender is "M" then 1 end) > count(case
    when gender is "F" then 1 end)
```

- Baris 1 : Mengambil data province_name dari tabel patients

- Baris 2 : Menggabungkan data pada tabel provinces sesuai dengan data pada tabel patients
- Baris 3 : Dengan acuan province_id pada tabel patients merujuk ke province_id pada tabel provinces
- Baris 4 : Mengelompokkan data berdasarkan province_id
- Baris 5 : Dengan kondisi dimana banyaknya data dengan gender "M" lebih banyak dibandingkan data dengan gender "F"

Screenshot :



- 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 'Halifax'

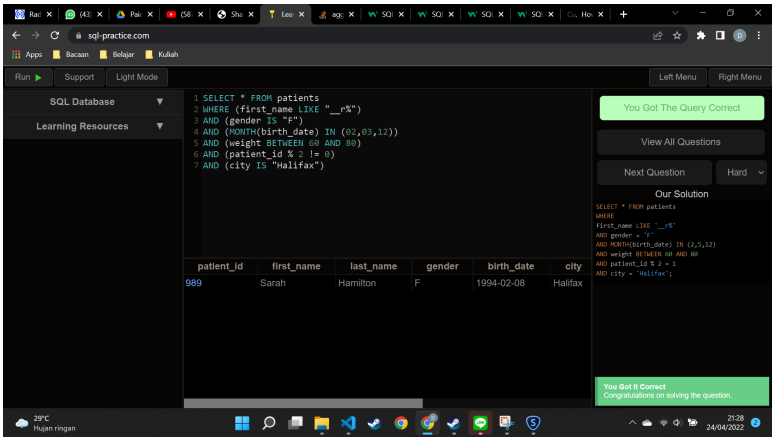
Query :

Listing 35: Hard - 7

```
select * from patients
where (first_name like "__r%")
and (gender is "F")
AND (month(birth_date) in (02,03,12))
and (weight between 60 and 80)
and (patient_id % 2 != 0)
and (city is "Halifax")
```

- Baris 1 : Mengambil semua data dari tabel patients
- Baris 2 : Dengan kondisi dimana first_name yang mengandung huruf "r" pada huruf ketiga
- Baris 3 : Dan bergender "F"
- Baris 4 : Dan memiliki bulan lahir yang termasuk dalam (2, 3, 12)
- Baris 5 : Dan memiliki weight diantara 60 dan 80
- Baris 6 : Dan memiliki patient_id ganjil
- Baris 7 : Dan berasal dari kota "Halifax"

Screenshot :



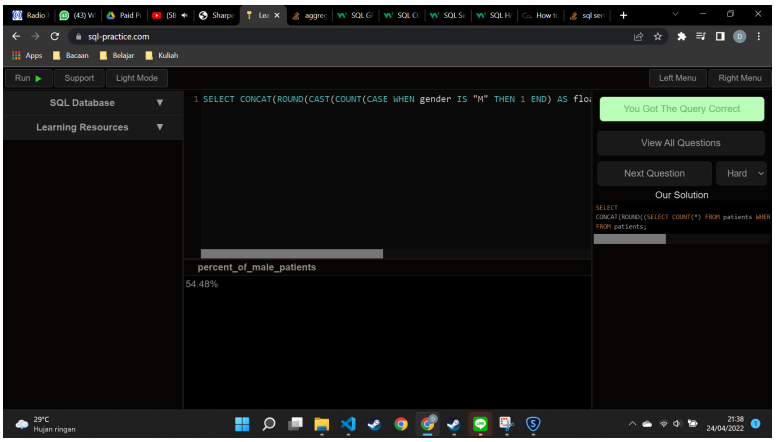
- Show the percent of patients that have 'M' as their gender. Round the answer to the nearest hundreth number and in percent form.
  Query :

Listing 36: Hard - 8

```
select concat(Round(cast(count(case when gender is "M" then 1
    end) as float) / cast(count(*) as float) * 100,2),"%") as
    percent_of_male_patients from patients
```

- Baris 1 : Mengambil data gabungan dari pembulatan pada 2 angka dibelakang koma dari perhitungan banyaknya data bergender "M" dibagi total banyaknya data dikali 100 (perhitungan persentase), dan tanda "%" (akan disebut sebagai percent_of_male_patients) dari tabel patients

Screenshot :


sql-practice.com
Run
Support
Light Mode
Left Menu
Right Menu
SQL Database
Learning Resources
1 SELECT CONCAT(ROUND(CAST(COUNT(CASE WHEN gender IS "M" THEN 1 END) AS flo
You Got The Query Correct
View All Questions
Next Question
Hard
Our Solution
percent_of_male_patients
54.48%