

Laporan Praktikum

Mata Kuliah Sistem Manajemen Basis Data

SQL Practice

Dwi Cahya Ramadani

23 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"
- 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 alergi)
- 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"

- 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

- 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 alergi)

- Show first name and last name concatenated 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

- 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

- 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

2 Soal Medium

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

Query :

Listing 9: 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

- 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 10: 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

- 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 11: 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"

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

Query :

Listing 12: 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"

- 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 13: 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
- Show the total amount of male patients and the total amount of female patients in the patients table
Query :

Listing 14: 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
- 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 15: 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

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

Query :

Listing 16: 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)

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

Query :

Listing 17: 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

- 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 18: Medium - 10

```
1 alter table patients
2 add Role varchar(100);
3 Alter table physicians
4 add Role varchar(100);
5 update patients
6 set Role = "Patient";
7 Update physicians
8 set Role = "Physician";
9 insert into patients (first_name, last_name, Role, Gender)
10 select first_name, last_name, Role, Role from physicians;
11 select first_name, last_name, Role from patients
```

```

12 order by first_name ASC, last_name ASC
13
14 '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
- Show all allergies ordered by popularity. Remove 'NKA' and NULL values from query
Query :

Listing 19: 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

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 20: 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
- Show patient_id, weight, height, isObese from the patients table. Display isObese as a boolean 0 or 1. Obese is defined as $\text{weight(kg)} / (\text{height(m)}^2)$

≥ 30 . weight is in units kg. height is in units cm

Query :

Listing 21: Hard - 2

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

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

- 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 22: 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"

- 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 23: 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
- 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 24: Hard - 5

```
1 alter table admissions
2 add column [has_insurance] varchar(3);
3 alter table admissions
4 add column [cost_after_insurance] int;
5 update admissions
6 set has_insurance =
7 (case
8 when patient_id % 2 = 0 then "Yes"
9 when patient_id % 2 != 0 then "No"
10 end),
11 cost_after_insurance =
12 (case
13 when patient_id % 2 = 0 then 10
14 when patient_id % 2 != 0 then 50
15 end);
16 select has_insurance, sum(cost_after_insurance) from
   admissions
17 group by has_insurance
18
19 '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`