**TASK 1:**

1. Installed the mysql container image from Docker hub on Docker desktop.
2. Connected to the virtual database using the connection details provided:

**SQL Link:**

DB Name: **project\_medical\_data\_history**

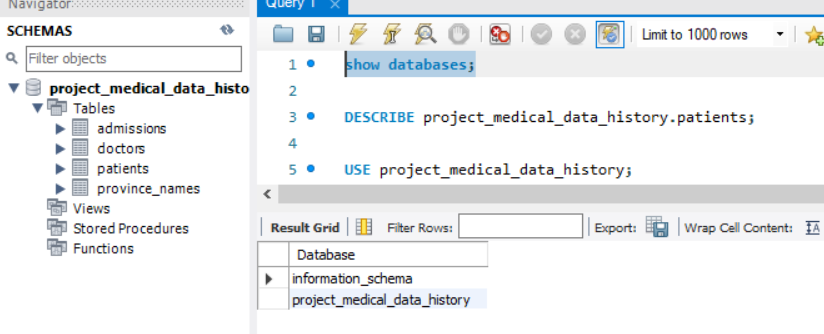
Host: ###########

Domain Name: projects.datamites.com

Username: **dm\_team4**

Password: **################**

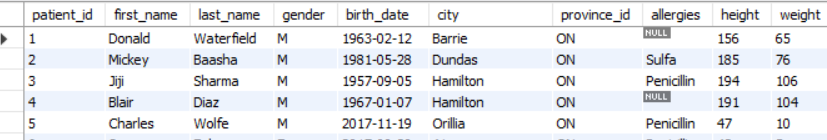
1. show databases;



1. Got the schema for “patients” table using the **DESCRIBE project\_medical\_data\_history.patients;** command.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| patient\_id | int(11) | NO | PRI |  |  |
| first\_name | varchar(150) | YES |  |  |  |
| last\_name | varchar(150) | YES |  |  |  |
| gender | char(1) | YES |  |  |  |
| birth\_date | date | YES |  |  |  |
| city | varchar(150) | YES |  |  |  |
| province\_id | char(2) | YES |  |  |  |
| allergies | varchar(150) | YES |  |  |  |
| height | decimal(3,0) | YES |  |  |  |
| weight | decimal(4,0) | YES |  |  |  |

1. SELECT \* FROM patients;

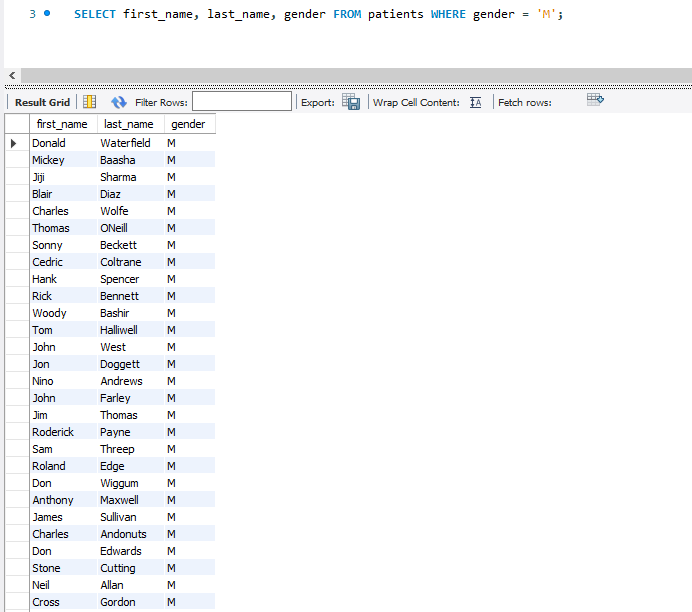


**TASK 2:**

\*Complete query outputs are in attached excel sheet.

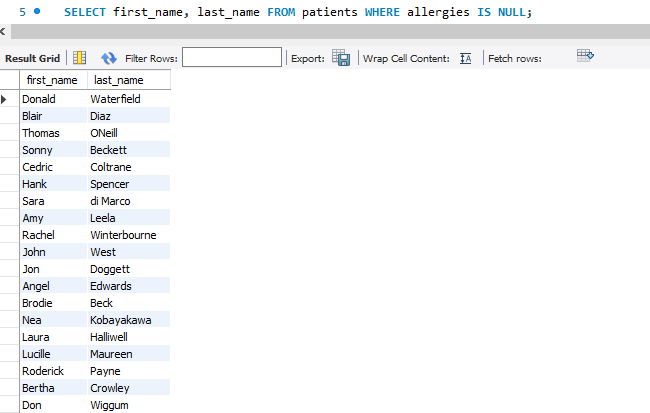
1. **Show first name, last name, and gender of patients whose gender is 'M'**

**Ans.** SELECT first\_name, last\_name, gender FROM patients WHERE gender = 'M';



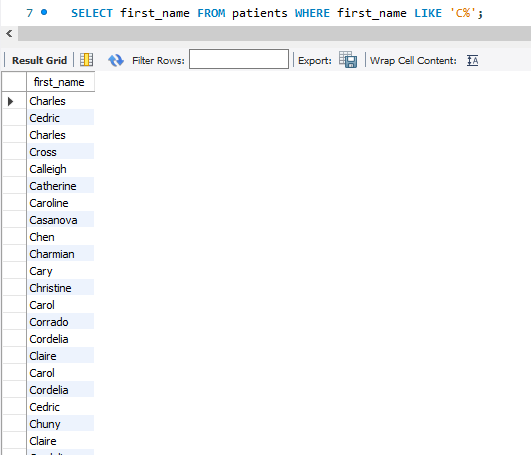
1. **Show first name and last name of patients who do not have allergies.**

Ans. SELECT first\_name, last\_name FROM patients WHERE allergies IS NULL;



1. **Show first name of patients that start with the letter 'C'.**

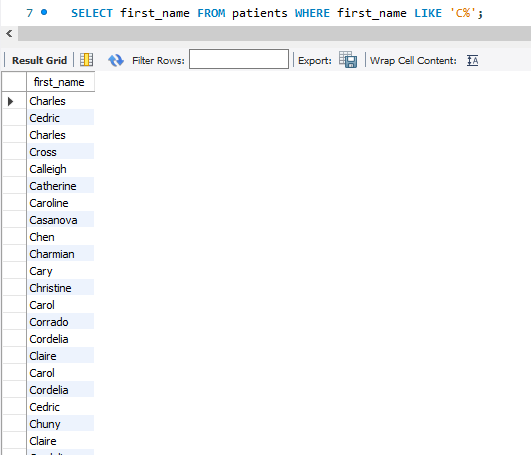
**Ans.** SELECT first\_name FROM patients WHERE first\_name LIKE 'C%';



1. **Show first name and last name of patients that weight within the range of**

**100 to 120 (inclusive)**

**Ans.** SELECT first\_name, last\_name FROM patients WHERE weight BETWEEN 100 AND 120;



1. **Update the patients table for the allergies column. If the patient's allergies is**

**null then replace it with 'NKA'**

**Ans.** UPDATE patients SET allergies = 'NKA' WHERE allergies IS NULL;

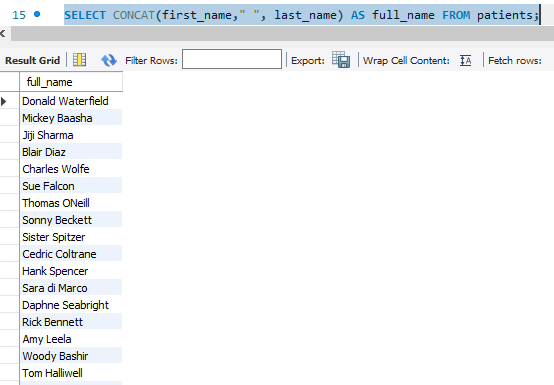
**UPDATE patients SET allergies = 'NKA' WHERE allergies IS NULL Error Code: 1142. UPDATE command denied to user 'dm\_team4'@'106.219.156.221' for table 'patients'**

Need to check with the team if this user has update rights to this database.

1. **Show first name and last name concatenated into one column to show their**

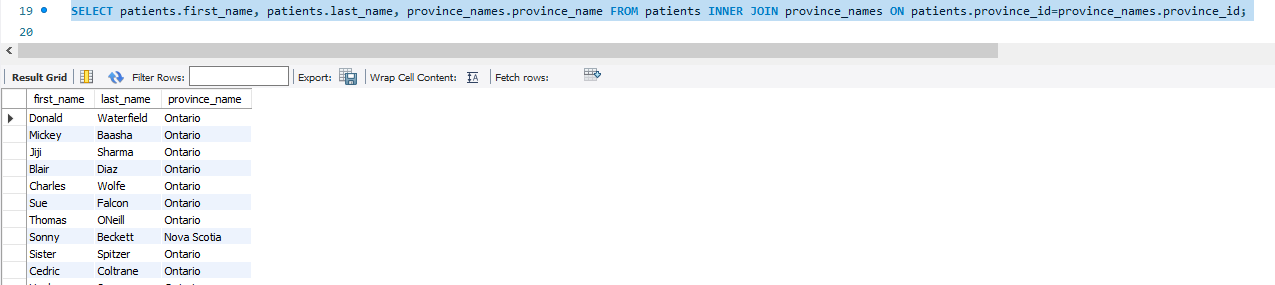
**full name.**

**Ans.** SELECT CONCAT(first\_name," ", last\_name) AS full\_name FROM patients;



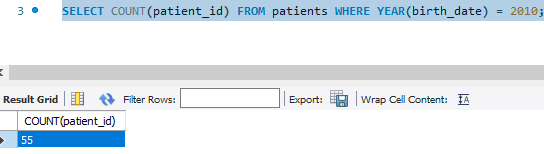
1. **Show first name, last name, and the full province name of each patient.**

**Ans.**  SELECT patients.first\_name, patients.last\_name, province\_names.province\_name FROM patients INNER JOIN province\_names ON patients.province\_id=province\_names.province\_id;



1. **Show how many patients have a birth\_date with 2010 as the birth year.**

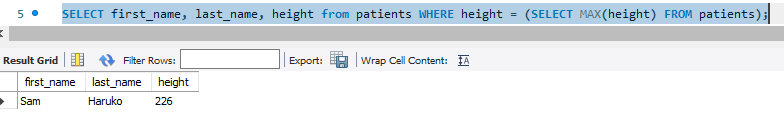
**Ans.** SELECT COUNT(patient\_id) FROM patients WHERE YEAR(birth\_date) = 2010;



1. **Show the first\_name, last\_name, and height of the patient with the greatest**

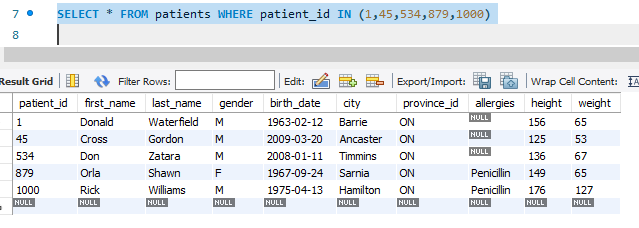
**height.**

**Ans.** SELECT first\_name, last\_name, height from patients WHERE height = (SELECT MAX(height) FROM patients);



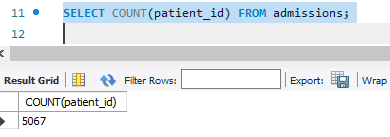
1. **Show all columns for patients who have one of the following patient\_ids: 1,45,534,879,1000**

**Ans.** SELECT \* FROM patients WHERE patient\_id IN (1,45,534,879,1000)



1. **Show the total number of admissions**

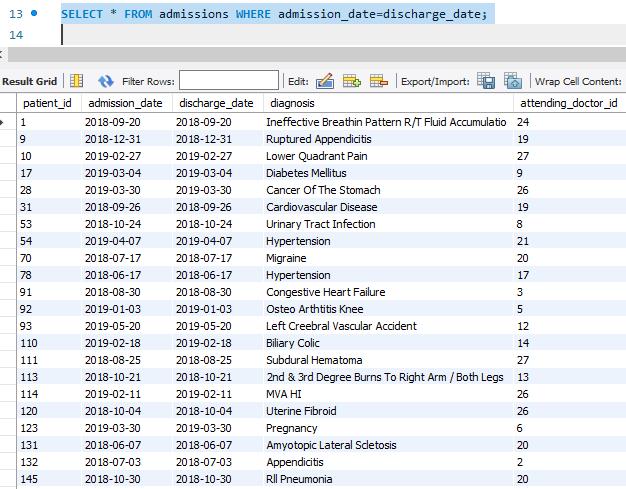
**Ans.** SELECT COUNT(patient\_id) FROM admissions;



1. **Show all the columns from admissions where the patient was admitted**

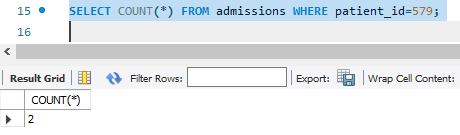
**and discharged on the same day.**

**Ans.** SELECT \* FROM admissions WHERE admission\_date=discharge\_date;



1. **Show the total number of admissions for patient\_id 579.**

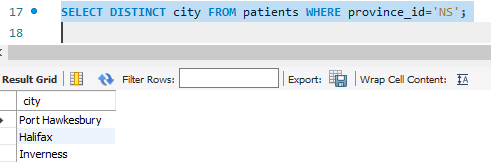
**Ans.** SELECT COUNT(\*) FROM admissions WHERE patient\_id=579;



1. **Based on the cities that our patients live in, show unique cities that are in**

**province\_id 'NS'?**

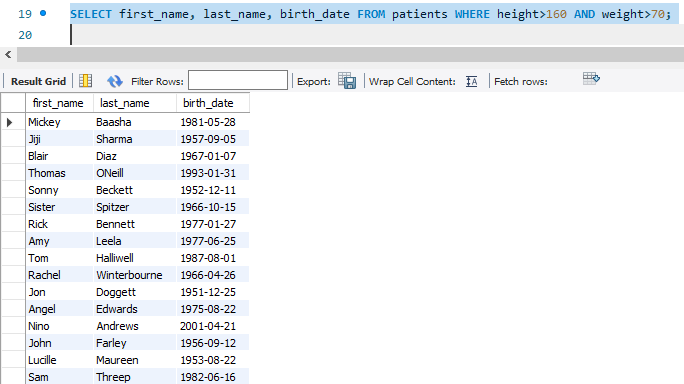
**Ans.** SELECT DISTINCT city FROM patients WHERE province\_id='NS';



1. **Write a query to find the first\_name, last name and birth date of patients**

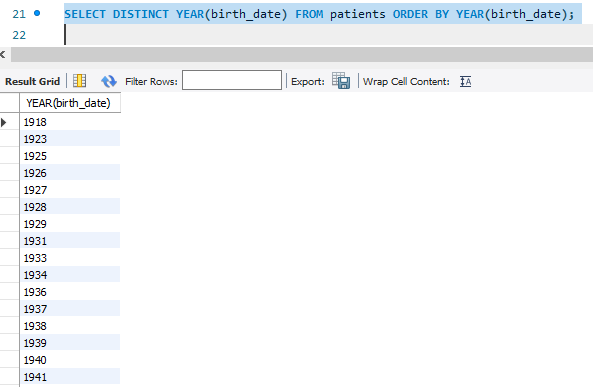
**who have height more than 160 and weight more than 70.**

**Ans.** SELECT first\_name, last\_name, birth\_date FROM patients WHERE height>160 AND weight>70;



1. **Show unique birth years from patients and order them by ascending.**

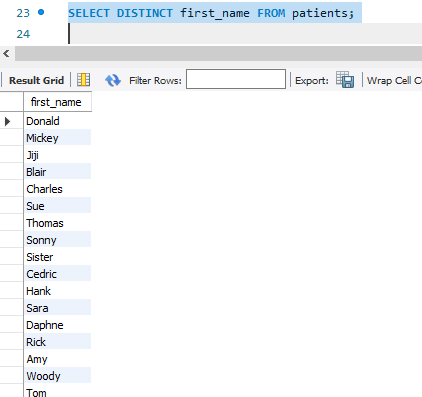
**Ans.** SELECT DISTINCT YEAR(birth\_date) FROM patients ORDER BY YEAR(birth\_date);



1. **Show unique first names from the patients table which only occurs once in**

**the list.**

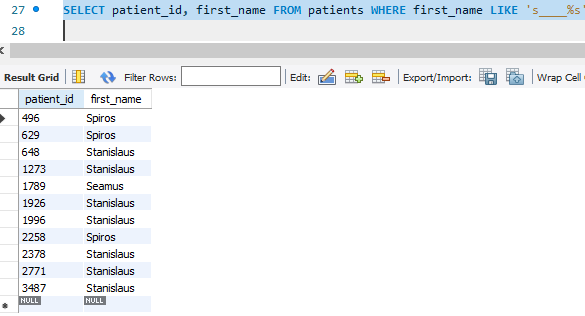
**Ans.** SELECT DISTINCT first\_name FROM patients;



1. **Show patient\_id and first\_name from patients where their first\_name starts**

**and ends with 's' and is at least 6 characters long.**

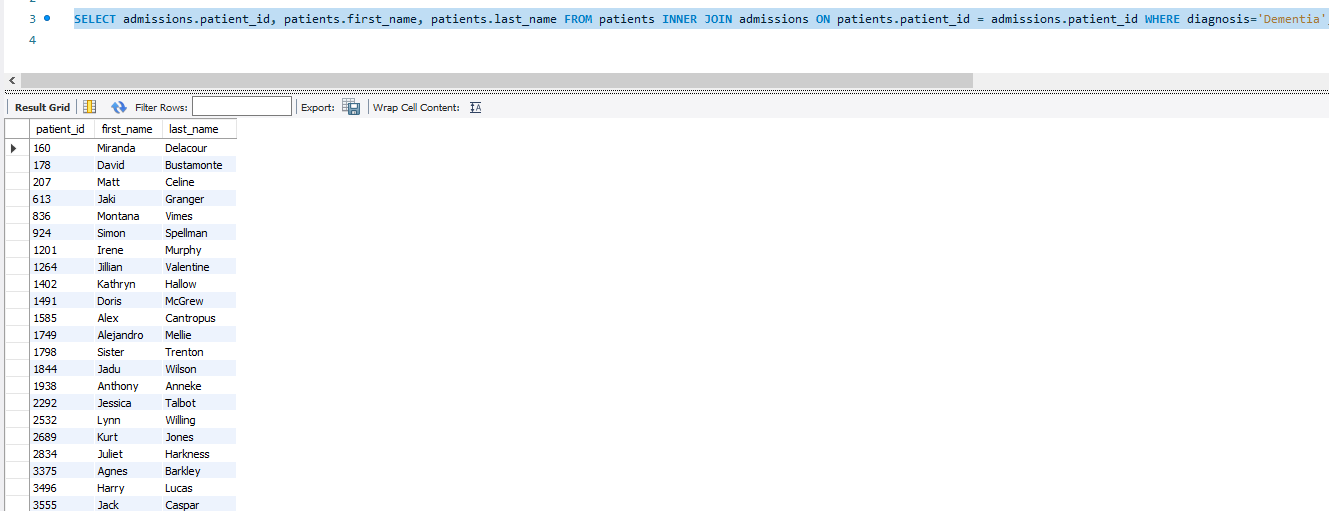
**Ans.** SELECT patient\_id, first\_name FROM patients WHERE first\_name LIKE 's\_\_\_\_%s';



1. **Show patient\_id, first\_name, last\_name from patients whose diagnosis is**

**'Dementia'. Primary diagnosis is stored in the admissions table.**

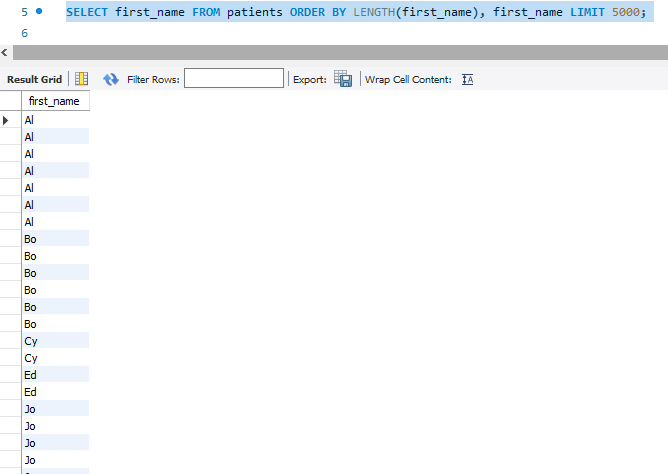
**Ans.** SELECT admissions.patient\_id, patients.first\_name, patients.last\_name FROM patients INNER JOIN admissions ON patients.patient\_id = admissions.patient\_id WHERE diagnosis='Dementia';



1. **Display every patient's first\_name. Order the list by the length of each**

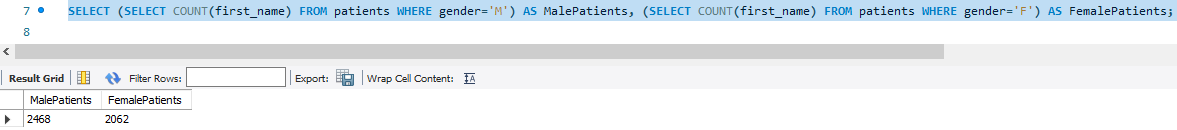
**name and then by alphabetically.**

**Ans.** SELECT first\_name FROM patients ORDER BY LENGTH(first\_name), first\_name LIMIT 5000;



1. **Show the total number of male patients and the total number of female**
2. **patients in the patients table. Display the two results in the same row.**

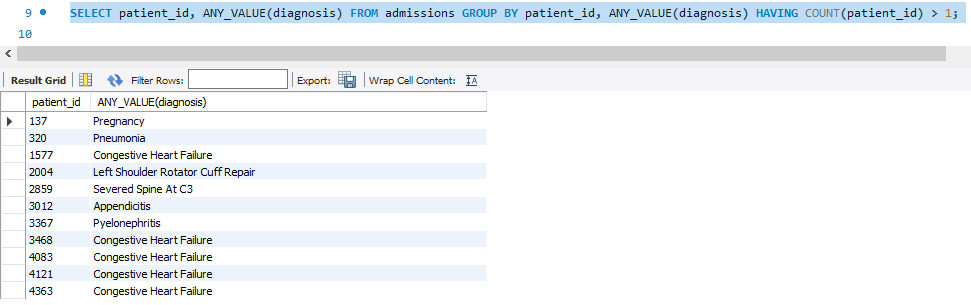
**Ans.** SELECT (SELECT COUNT(first\_name) FROM patients WHERE gender='M') AS MalePatients, (SELECT COUNT(first\_name) FROM patients WHERE gender='F') AS FemalePatients;



1. **Show patient\_id, diagnosis from admissions. Find patients admitted**

**multiple times for the same diagnosis.**

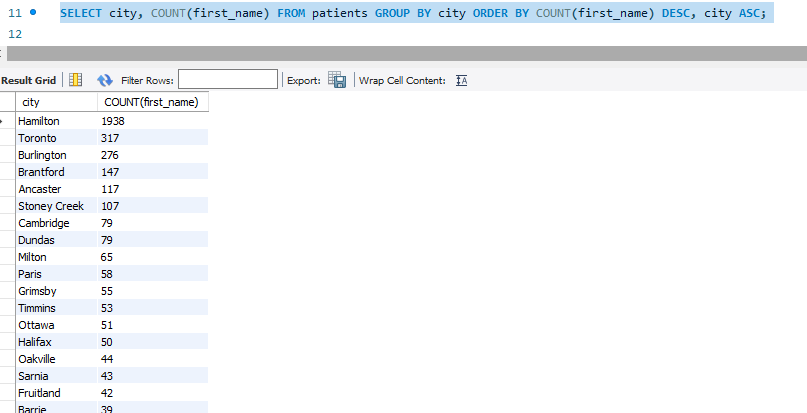
**Ans.** SELECT patient\_id, ANY\_VALUE(diagnosis) FROM admissions GROUP BY patient\_id, ANY\_VALUE(diagnosis) HAVING COUNT(patient\_id) > 1;



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.**

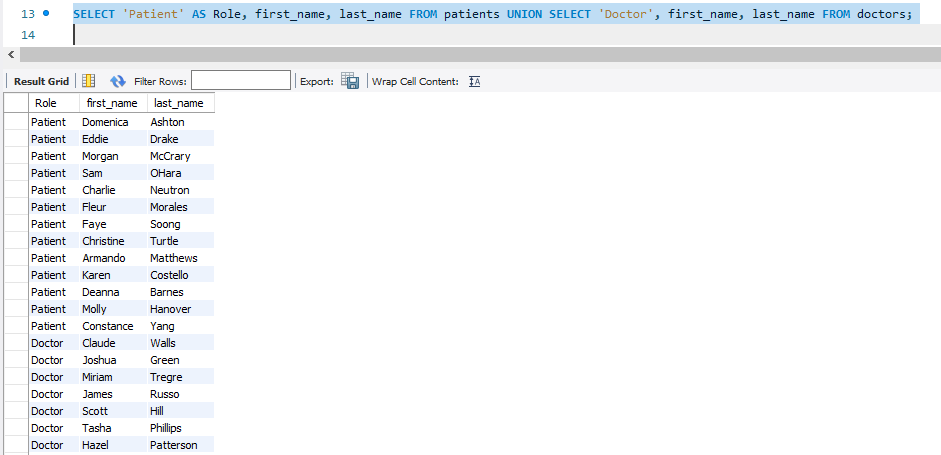
**Ans.** SELECT city, COUNT(first\_name) FROM patients GROUP BY city ORDER BY COUNT(first\_name) DESC, city ASC;



1. **Show first name, last name and role of every person that is either patient**

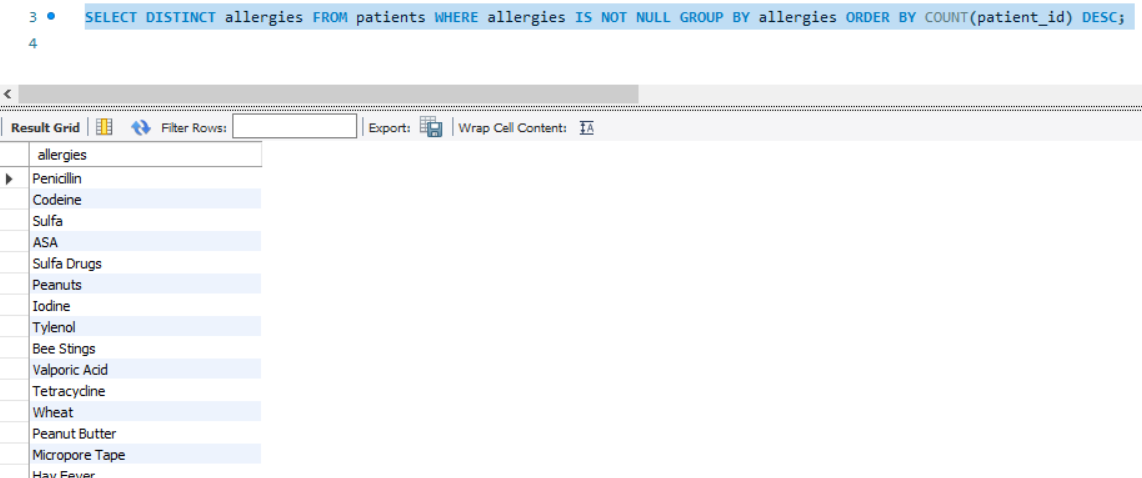
**or doctor. The roles are either "Patient" or "Doctor"**

**Ans.** SELECT 'Patient' AS Role, first\_name, last\_name FROM patients UNION SELECT 'Doctor', first\_name, last\_name FROM doctors;



1. **Show all allergies ordered by popularity. Remove NULL values from the**

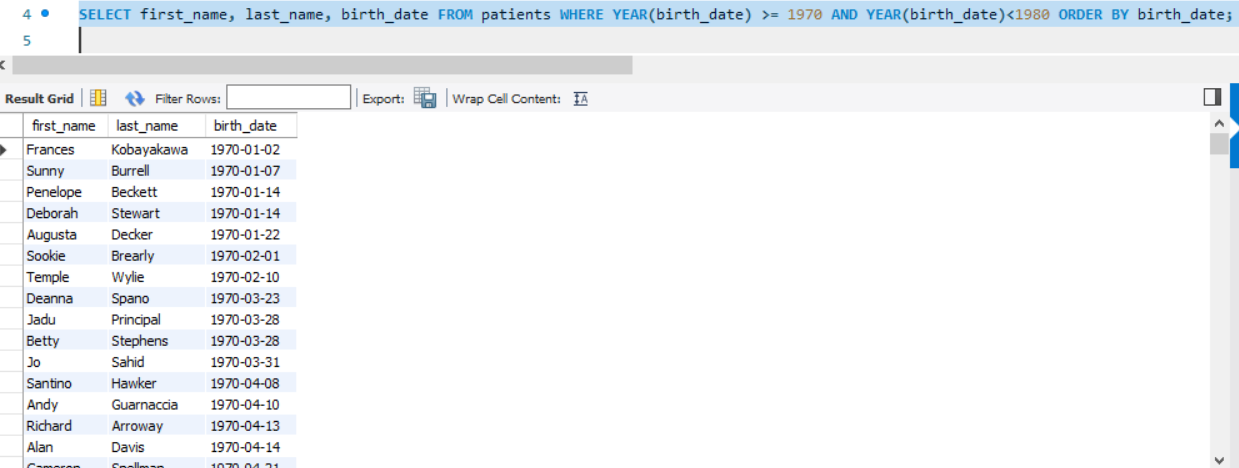
**query.**

**Ans**. SELECT DISTINCT allergies FROM patients WHERE allergies IS NOT NULL GROUP BY allergies ORDER BY COUNT(patient\_id) DESC**;**

1. **Show all patient's first\_name, last\_name, and birth\_date who were born in**

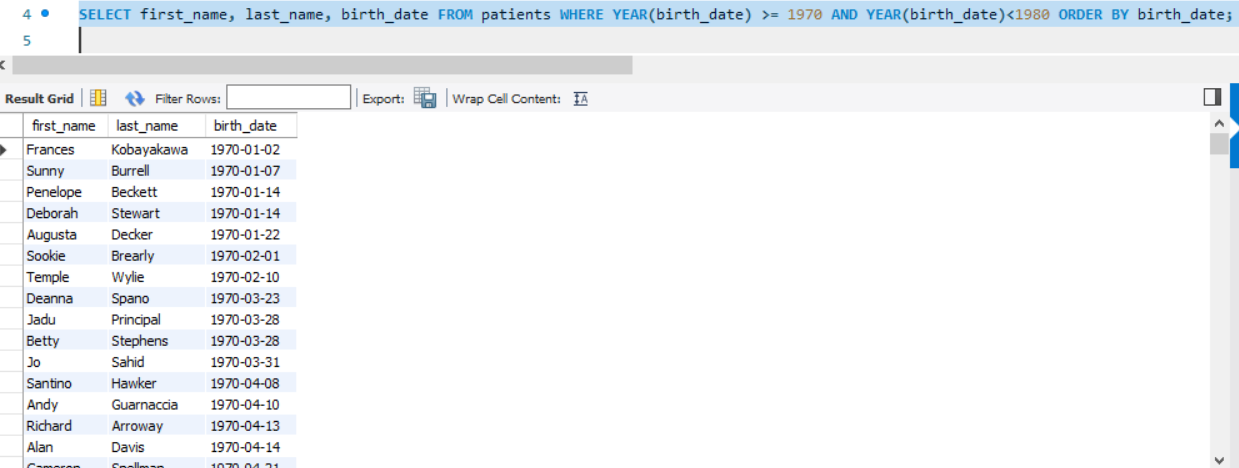
**the 1970s decade. Sort the list starting from the earliest birth\_date.**

**Ans.** SELECT first\_name, last\_name, birth\_date FROM patients WHERE YEAR(birth\_date) >= 1970 AND YEAR(birth\_date)<1980 ORDER BY birth\_date;



1. **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 lowercase letters. Separate the last\_name and first\_name with a comma. Order thelist by the first\_name in descending order EX: SMITH,jane**

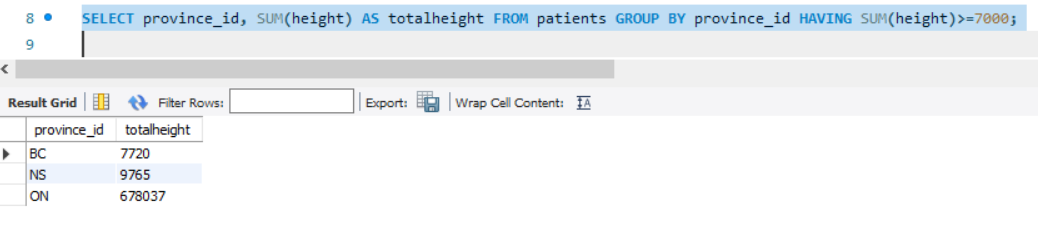
**Ans.** SELECT CONCAT(UPPER(last\_name),",", LOWER(first\_name)) AS full\_name FROM patients ORDER BY first\_name DESC LIMIT 5000;



1. **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.**

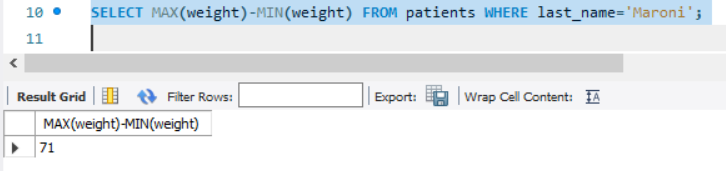
**Ans.** SELECT province\_id, SUM(height) AS totalheight FROM patients GROUP BY province\_id HAVING SUM(height)>=7000;



1. **Show the difference between the largest weight and smallest weight for**

**patients with the last name 'Maroni'**

**Ans.** SELECT MAX(weight)-MIN(weight) FROM patients WHERE last\_name='Maroni';

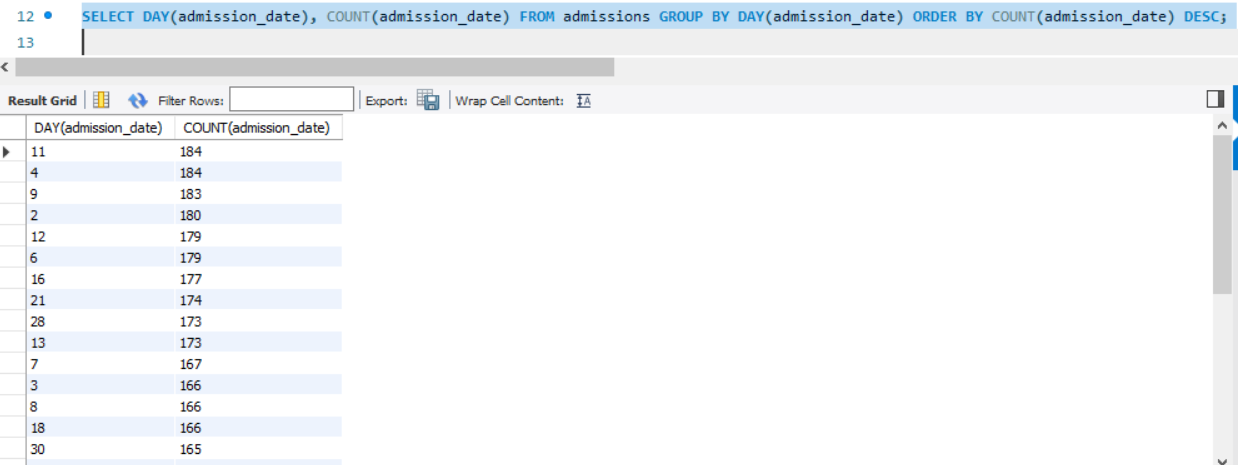


1. **Show all of the days of the month (1-31) and how many admission\_dates**

**occurred on that day. Sort by the day with most admissions to least**

**admissions.**

**Ans.** SELECT DAY(admission\_date), COUNT(admission\_date) FROM admissions GROUP BY DAY(admission\_date) ORDER BY COUNT(admission\_date) DESC;



1. **Show all of the patients grouped into weight groups. Show the total**

**number of patients in each weight group. Order the list by the weight group**

**descending. e.g. if they weigh 100 to 109 they are placed in the 100 weight**

**group, 110-119 = 110 weight group, etc.**

**Ans.** SELECT CASE

WHEN weight BETWEEN 0 AND 25 THEN '0-25'

WHEN weight BETWEEN 26 AND 50 THEN '26-50'

WHEN weight BETWEEN 51 AND 75 THEN '51-75'

WHEN weight BETWEEN 76 AND 100 THEN '76-100'

WHEN weight BETWEEN 100 AND 25 THEN '100-125'

ELSE '126-150'

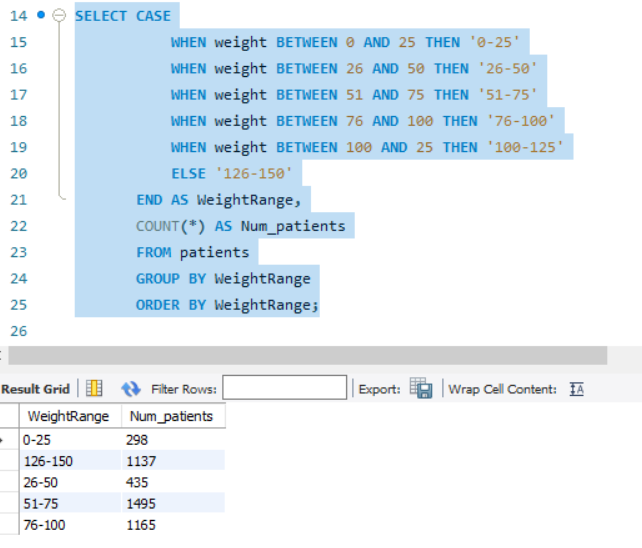
END AS WeightRange,

COUNT(\*) AS Num\_patients

FROM patients

GROUP BY WeightRange

ORDER BY WeightRange;



1. **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).**

**Weight is in units kg. Height is in units cm.**

**Ans.** SELECT patient\_id, weight, height,

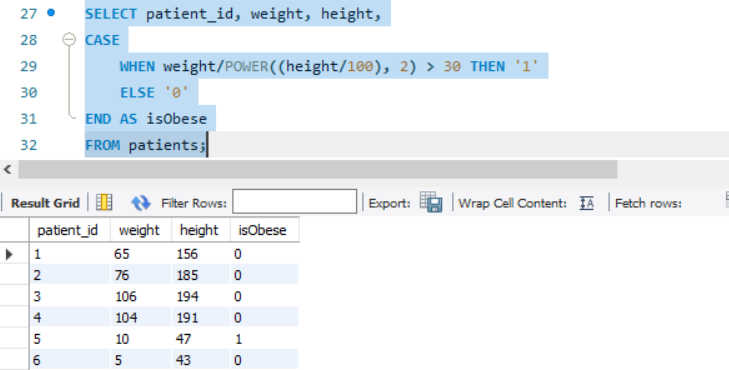
CASE

WHEN weight/POWER((height/100), 2) > 30 THEN '1'

ELSE '0'

END AS isObese

FROM patients;



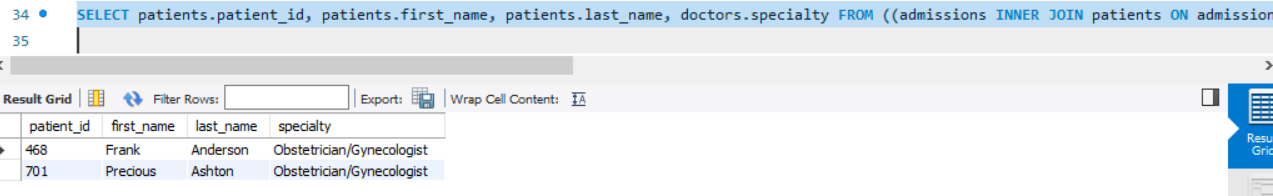
1. **Show patient\_id, first\_name, last\_name, and attending doctor's specialty.**

**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.**

**Ans.** SELECT patients.patient\_id, patients.first\_name, patients.last\_name, doctors.specialty FROM ((admissions INNER JOIN patients ON admissions.patient\_id = patients.patient\_id) INNER JOIN doctors ON admissions.attending\_doctor\_id = doctors.doctor\_id) WHERE diagnosis='Epilepsy' AND doctors.first\_name='Lisa';



1. **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.**

**Ans.** SELECT patient\_id, CONCAT(patient\_id, LENGTH(last\_name), YEAR(birth\_date)) AS temp\_password FROM patients;

