SQL PROJECT

ID: PTID-CDA-OCT-25-803

PROJECT ID: PRSQL-02-MEDICAL DATA HISTORY

ADITYA HINGRAJIYA

adityahingrajiya7@gmail.com

INDEX

Sr. No	Content	Page no
1	INTRODUCTION • Project Objective • Dataset Overview • Tools & Technologies Used	3
2	SQL QUERIES • Questions • SQL Queries • Results & Screenshots of Queries	4
3	INSIGHTS	25
4	CHALLENGES • Technical Challenges • Solutions/Approach to Fix Them	26
5	SUMMARY	27

INTRODUCTION

The healthcare industry involves the generation of significant structured data on patients, physicians, hospital admissions, and medical services delivered at the regional level. The effective storage and recovery of such data are crucial for keeping track of the medical history and decision-making in the field of healthcare. SQL-based relational modeling offers a robust mechanism for managing the healthcare data in a systematic manner.

The project that catches my interest most is the Medical Data History project. The SQL application of this project is related to modeling and querying the healthcare database that comprises several interrelated tables named Patients, Doctors, Admissions, and Province_Name. The tables are associated with different entities of the medical setup and linked together through primary and foreign keys.

The purpose of this project is to prove the application of conceptual knowledge in SQL, encompassing concepts like join, filter, aggregation, and relationship mapping to a real-world healthcare setting. The database, through well-organized queries, will allow for the processing of patient admission information, doctor engagement, and provincial healthcare information, thus establishing the relevance of using SQL in healthcare data management.

SQL QUERIES

We have solved the provided list of questions as a project and attached the screenshot of the solution:

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

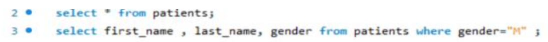
Query:

SELECT * FROM patients;

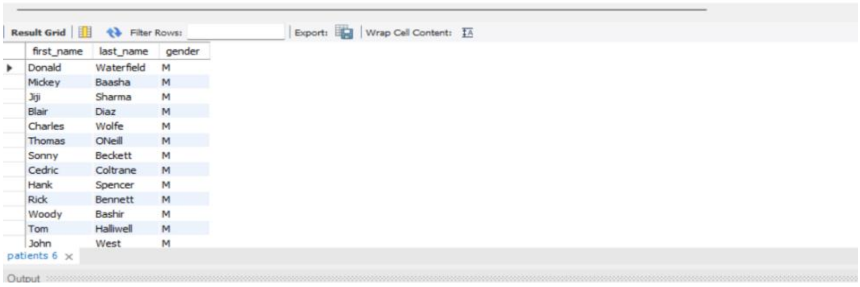
SELECT first_name, last_name, gender

FROM patients

WHERE gender=’M’



```
select * from patients;
select first_name , last_name, gender from patients where gender="M" ;
```



first_name	last_name	gender
Donald	Waterfield	M
Mickey	Baasha	M
Jiji	Sharma	M
Blair	Diaz	M
Charles	Wolfe	M
Thomas	ONeill	M
Sonny	Beckett	M
Cedric	Coltrane	M
Hank	Spencer	M
Rick	Bennett	M
Woody	Bashir	M
Tom	Halliwell	M
John	West	M

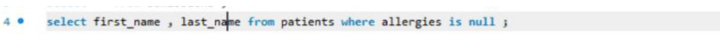
Q.2. Show first name and last name of patients who does not have allergies.

Query:

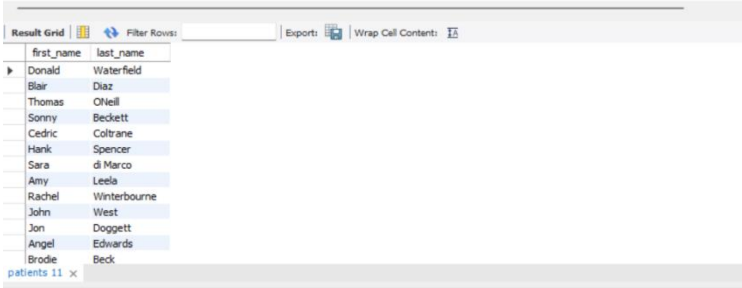
SELECT first_name, last_name

FROM patients

WHERE allergies is null;



```
select first_name , last_name from patients where allergies is null ;
```

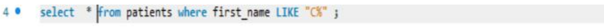


first_name	last_name
Donald	Waterfield
Blair	Diaz
Thomas	ONeill
Sonny	Beckett
Cedric	Coltrane
Hank	Spencer
Sara	di Marco
Amy	Leela
Rachel	Winterbourne
John	West
Jon	Doggett
Angel	Edwards
Brodie	Beck

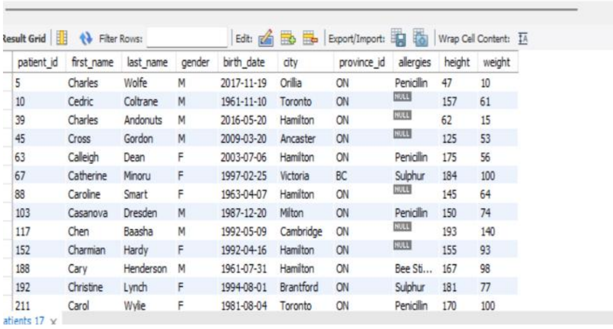
Q.3. Show first name of patients that start with the letter 'C'

Query:

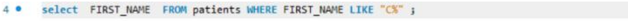
```
SELECT *
FROM patients
WHERE first_name LIKE ‘C%’;
```



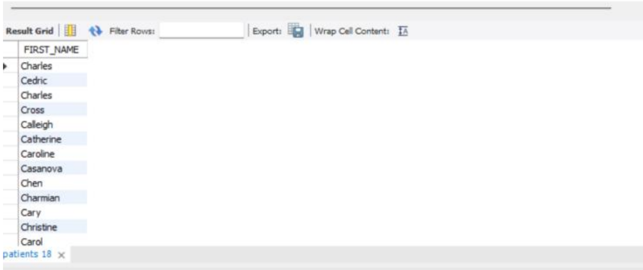
```
select * from patients where first_name LIKE "C%" ;
```



patient_id	first_name	last_name	gender	birth_date	city	province_id	allergies	height	weight
5	Charles	Wolfe	M	2017-11-19	Orillia	ON	Penicillin	47	10
10	Cedric	Coltrane	M	1961-11-10	Toronto	ON	NULL	157	61
39	Charles	Andonuts	M	2016-05-20	Hamilton	ON	NULL	62	15
45	Cross	Gordon	M	2009-03-20	Ancaster	ON	NULL	125	53
63	Calleigh	Dean	F	2003-07-06	Hamilton	ON	Penicillin	175	56
67	Catherine	Minoru	F	1997-02-25	Victoria	BC	Sulphur	184	100
88	Caroline	Smart	F	1963-04-07	Hamilton	ON	NULL	145	64
103	Casanova	Dresden	M	1987-12-20	Milton	ON	Penicillin	150	74
117	Chen	Baasha	M	1992-05-09	Cambridge	ON	NULL	193	140
152	Charmian	Hardy	F	1992-04-16	Hamilton	ON	NULL	155	93
188	Cary	Henderson	M	1961-07-31	Hamilton	ON	Bee Sti...	167	98
192	Christine	Lynch	F	1994-08-01	Brantford	ON	Sulphur	181	77
211	Carol	Wylie	F	1981-08-04	Toronto	ON	Penicillin	170	100



```
select FIRST_NAME FROM patients WHERE FIRST_NAME LIKE "C%" ;
```

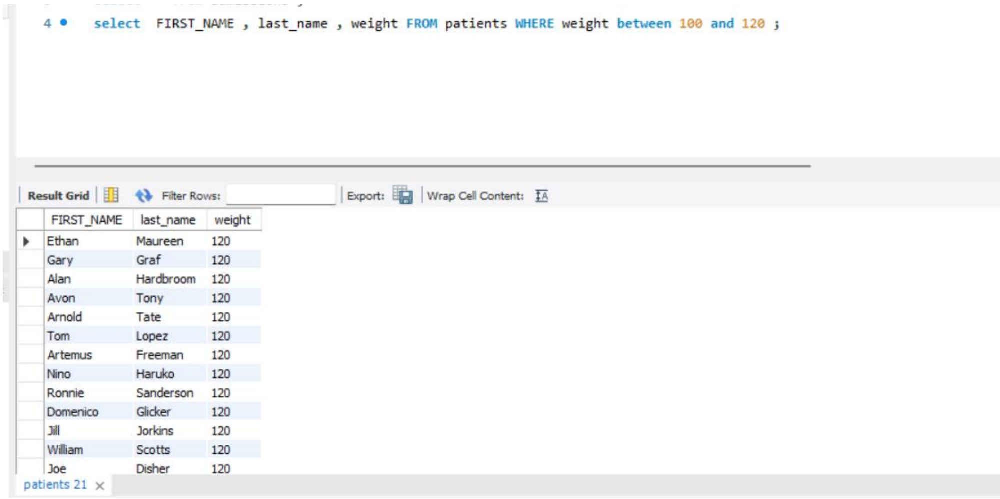


FIRST_NAME
Charles
Cedric
Charles
Cross
Calleigh
Catherine
Caroline
Casanova
Chen
Charmian
Cary
Christine
Carol

Q.4. Show first name and last name of patients that weight within the range of 100 to 120 (inclusive).

Query:

```
SELECT first_name, last_name, weight
FROM patients
WHERE weight BETWEEN 100 AND 120;
```



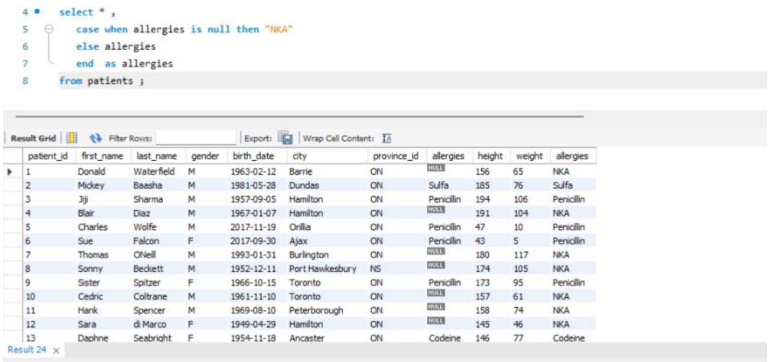
```
select  FIRST_NAME , last_name , weight FROM patients WHERE weight between 100 and 120 ;
```

FIRST_NAME	last_name	weight
Ethan	Maureen	120
Gary	Graf	120
Alan	Hardbroom	120
Avon	Tony	120
Arnold	Tate	120
Tom	Lopez	120
Artemus	Freeman	120
Nino	Haruko	120
Ronnie	Sanderson	120
Domenico	Glicker	120
Jill	Jorkins	120
William	Scotts	120
Joe	Disher	120

Q.5. Update the patients table for the allergy’s column. If the patient's allergies is null then replace it with 'NKA'

Query:

```
SELECT *
        CASE WHEN allergies IS NULL THEN ‘NKA’
        ELSE allergies
        END AS allergies
FROM patients;
```



```
select * ,
  case when allergies is null then "NKA"
  else allergies
  end  as allergies
from patients ;
```

patient_id	first_name	last_name	gender	birth_date	city	province_id	allergies	height	weight	allergies
1	Donald	Waterfield	M	1963-02-12	Barrie	ON	NULL	156	65	NKA
2	Mickey	Baasha	M	1981-05-28	Dundas	ON	Sulfa	185	76	Sulfa
3	Jiji	Sharma	M	1957-09-05	Hamilton	ON	Penicillin	194	106	Penicillin
4	Blair	Diaz	M	1967-01-07	Hamilton	ON	NULL	191	104	NKA
5	Charles	Wolfe	M	2017-11-19	Orillia	ON	Penicillin	47	10	Penicillin
6	Sue	Falcon	F	2017-09-30	Ajax	ON	Penicillin	43	5	Penicillin
7	Thomas	ONeill	M	1993-01-31	Burlington	ON	NULL	180	117	NKA
8	Sonny	Beckett	M	1952-12-11	Port Hawkesbury	NS	NULL	174	105	NKA
9	Sister	Spitzer	F	1966-10-15	Toronto	ON	Penicillin	173	95	Penicillin
10	Cedric	Coltrane	M	1961-11-10	Toronto	ON	NULL	157	61	NKA
11	Hank	Spencer	M	1969-08-10	Peterborough	ON	NULL	158	74	NKA
12	Sara	di Marco	F	1949-04-29	Hamilton	ON	NULL	145	46	NKA
13	Daphne	Seabright	F	1954-11-18	Ancaster	ON	Codeine	146	77	Codeine

Q.6. Show first name and last name concatenated into one column to show their full name.

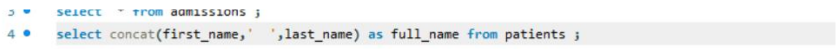
Query:

SELECT *

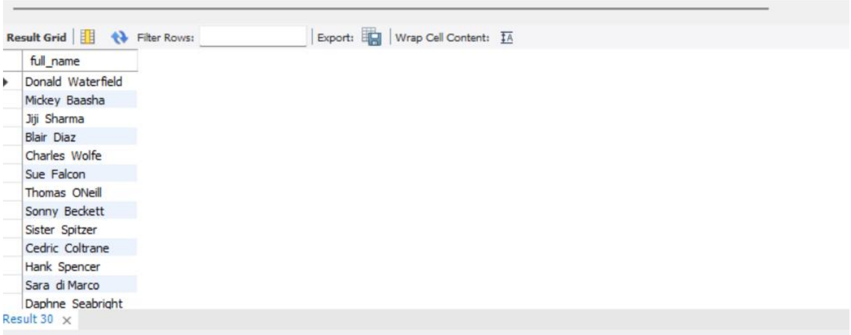
FROM admissions;

SELECT concat (first_name,' ', last_name') AS full_name

FROM patients;



```
select * from admissions ;
select concat(first_name,'  ',last_name) as full_name from patients ;
```



full_name
Donald Waterfield
Mickey Baasha
Jiji Sharma
Blair Diaz
Charles Wolfe
Sue Falcon
Thomas ONeill
Sonny Beckett
Sister Spitzer
Cedric Coltrane
Hank Spencer
Sara di Marco
Daphne Seabright

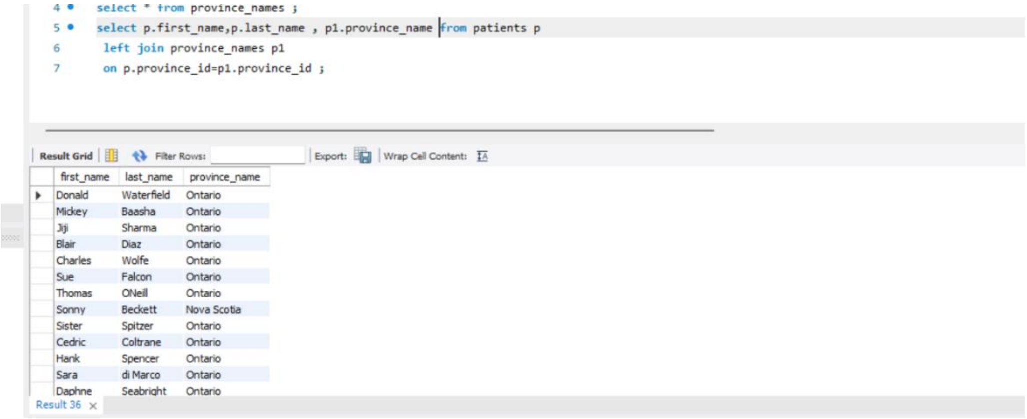
Q.7. Show first name, last name, and the full province name of each patient.

Query:

SELECT * FROM province_name;

SELECT p.first_name, p.last_name, p1.province_name

FROM patients;



```
select * from province_names ;
select p.first_name,p.last_name , p1.province_name from patients p
 left join province_names p1
 on p.province_id=p1.province_id ;
```

first_name	last_name	province_name
Donald	Waterfield	Ontario
Mickey	Baasha	Ontario
Jiji	Sharma	Ontario
Blair	Diaz	Ontario
Charles	Wolfe	Ontario
Sue	Falcon	Ontario
Thomas	ONeill	Ontario
Sonny	Beckett	Nova Scotia
Sister	Spitzer	Ontario
Cedric	Coltrane	Ontario
Hank	Spencer	Ontario
Sara	di Marco	Ontario
Daphne	Seabright	Ontario

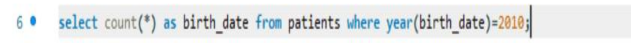
Q.8. Show how many patients have a birth_date with 2010 as the birth year.

Query_1:

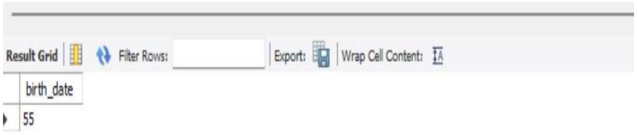
SELECT COUNT (*) AS birth_date

FROM patients

WHERE year(birth_date) =2010;



```
select count(*) as birth_date from patients where year(birth_date)=2010;
```



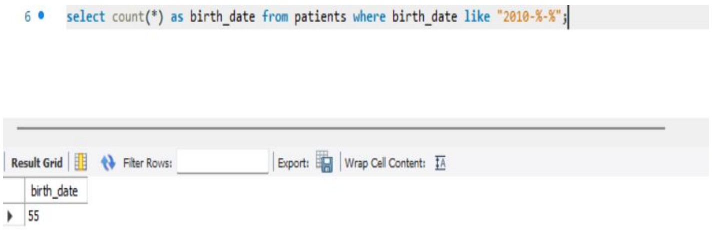
birth_date
55

Query_2:

SELECT COUNT (*) AS birth_date

FROM patients

WHERE birth_date LIKE “2010-%-%”;



```
select count(*) as birth_date from patients where birth_date like "2010-%-%";
```

birth_date
55

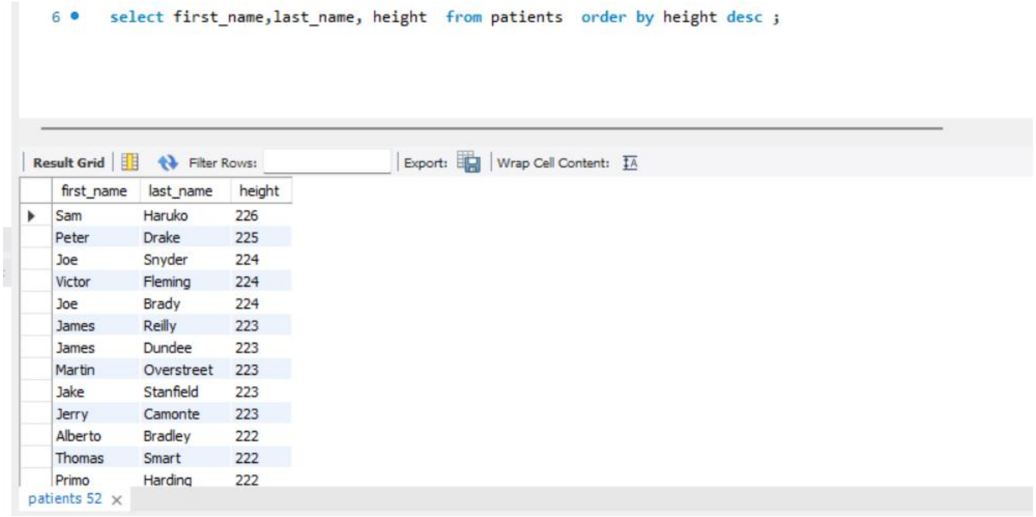
Q.9. Show the first_name, last_name, and height of the patient with the greatest height.

Query:

SELECT first_name, last_name, height

FROM patients

ORDER BY height DESC;



```
select first_name,last_name, height  from patients  order by height desc ;
```

first_name	last_name	height
Sam	Haruko	226
Peter	Drake	225
Joe	Snyder	224
Victor	Fleming	224
Joe	Brady	224
James	Reilly	223
James	Dundee	223
Martin	Overstreet	223
Jake	Stanfield	223
Jerry	Camonte	223
Alberto	Bradley	222
Thomas	Smart	222
Primo	Harding	222

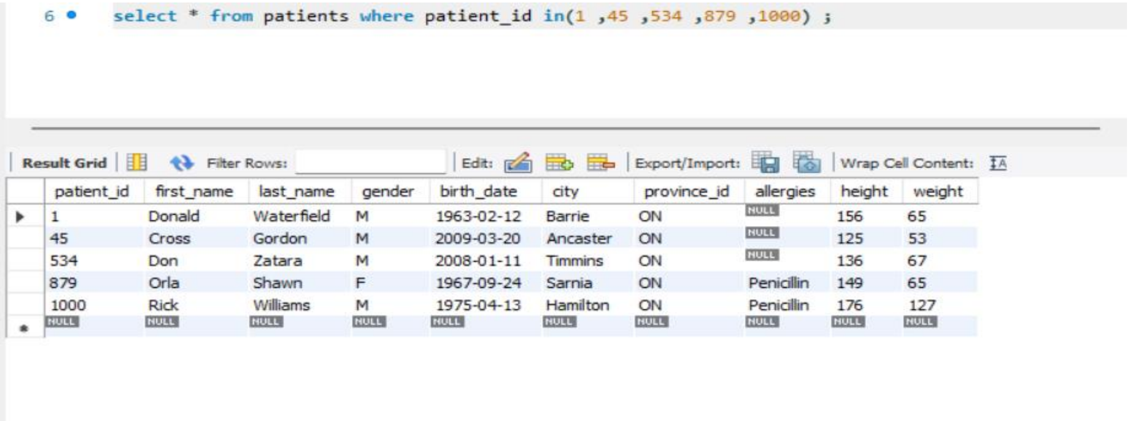
Q.10. Show all columns for patients who have one of the following patient_ids: 1,45,534,879,1000

Query:

SELECT *

FROM patients

WHERE patient_id IN (1, 45, 534, 879, 1000);



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

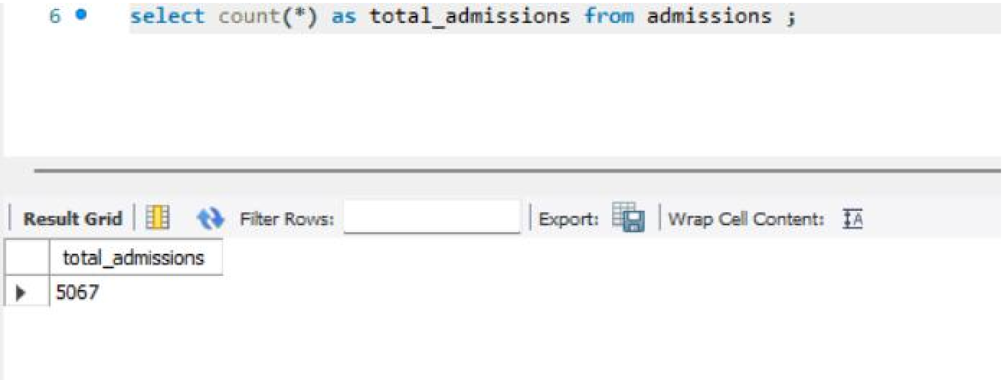
patient_id	first_name	last_name	gender	birth_date	city	province_id	allergies	height	weight
1	Donald	Waterfield	M	1963-02-12	Barrie	ON	NULL	156	65
45	Cross	Gordon	M	2009-03-20	Ancaster	ON	NULL	125	53
534	Don	Zatara	M	2008-01-11	Timmins	ON	NULL	136	67
879	Orla	Shawn	F	1967-09-24	Sarnia	ON	Penicillin	149	65
1000	Rick	Williams	M	1975-04-13	Hamilton	ON	Penicillin	176	127
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Q.11. Show the total number of admissions.

Query:

SELECT COUNT (*) AS total_admissions

FROM admissions;



```
6 •   select count(*) as total_admissions from admissions ;
```

total_admissions
5067

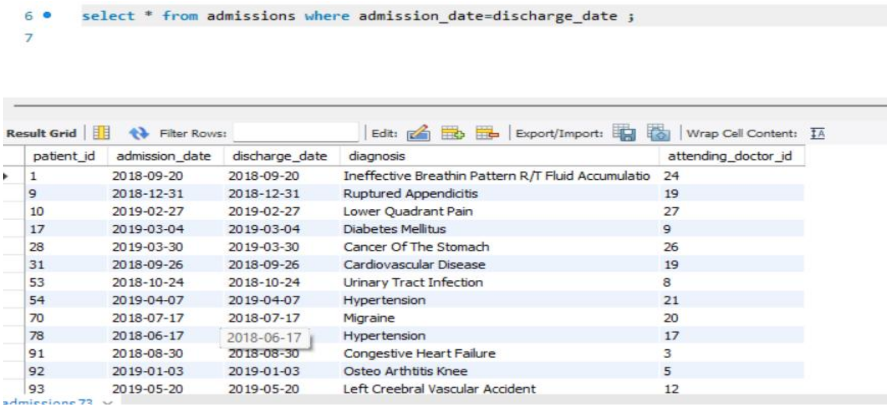
Q.12. Show all the columns from admissions where the patient was admitted and discharged on the same day.

Query:

SELECT *

FROM admissions

WHERE admission_date= discharge_date;



```
select * from admissions where admission_date=discharge_date ;
```

patient_id	admission_date	discharge_date	diagnosis	attending_doctor_id
1	2018-09-20	2018-09-20	Ineffective Breathin Pattern R/T Fluid Accumulatio	24
9	2018-12-31	2018-12-31	Ruptured Appendicitis	19
10	2019-02-27	2019-02-27	Lower Quadrant Pain	27
17	2019-03-04	2019-03-04	Diabetes Mellitus	9
28	2019-03-30	2019-03-30	Cancer Of The Stomach	26
31	2018-09-26	2018-09-26	Cardiovascular Disease	19
53	2018-10-24	2018-10-24	Urinary Tract Infection	8
54	2019-04-07	2019-04-07	Hypertension	21
70	2018-07-17	2018-07-17	Migraine	20
78	2018-06-17	2018-06-17	Hypertension	17
91	2018-08-30	2018-08-30	Congestive Heart Failure	3
92	2019-01-03	2019-01-03	Osteo Arthtitis Knee	5
93	2019-05-20	2019-05-20	Left Creebral Vascular Accident	12

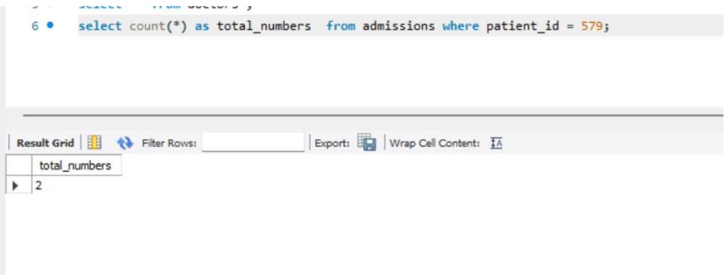
Q.13. Show the total number of admissions for patient_id 579.

Query:

SELECT COUNT (*) AS total_numbers

FROM admissions

WHERE patient_id= 579;



```
select count(*) as total_numbers  from admissions where patient_id = 579;
```

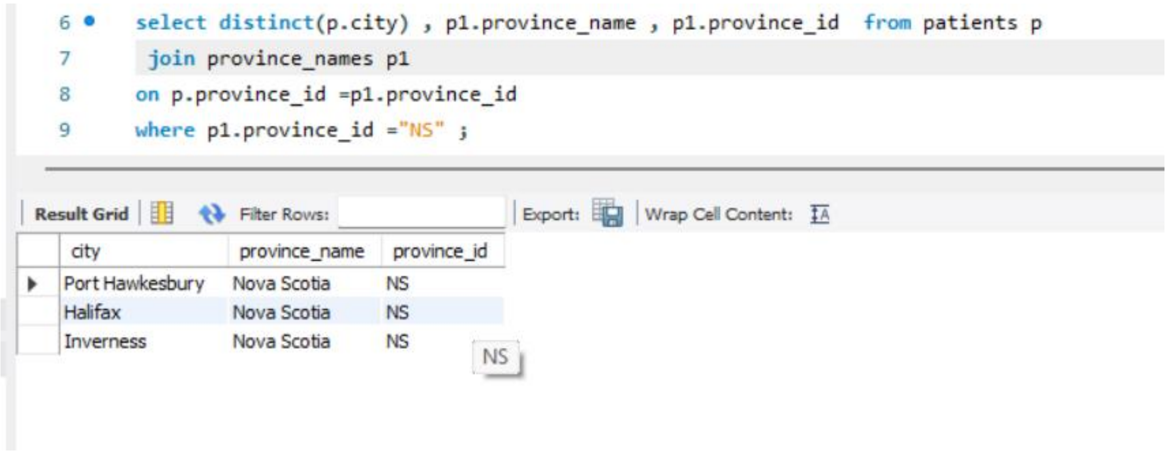
total_numbers
2

Q.14. Based on the cities that our patients live in, show unique cities that are in province_id 'NS'?

Query:

SELECT DISTINCT (p.city), p1.province_name, p1.province_id

FROM patients p;



```
select distinct(p.city) , p1.province_name , p1.province_id  from patients p
 join province_names p1
on p.province_id =p1.province_id
where p1.province_id ="NS" ;
```

city	province_name	province_id
Port Hawkesbury	Nova Scotia	NS
Halifax	Nova Scotia	NS
Inverness	Nova Scotia	NS

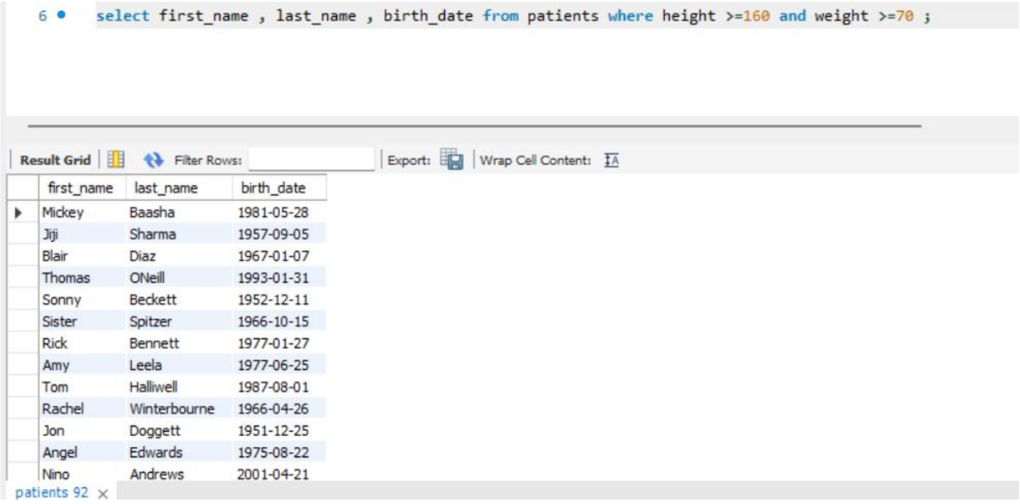
Q.15. Write a query to find the first_name, last name and birth date of patients who have height more than 160 and weight more than 70?

Query:

SELECT first_name, last_name, birth_date

FROM patients

WHERE height>=160 AND weight>=70;



```
select first_name , last_name , birth_date from patients where height >=160 and weight >=70 ;
```

first_name	last_name	birth_date
Mickey	Baasha	1981-05-28
Jiji	Sharma	1957-09-05
Blair	Diaz	1967-01-07
Thomas	ONeill	1993-01-31
Sonny	Beckett	1952-12-11
Sister	Spitzer	1966-10-15
Rick	Bennett	1977-01-27
Amy	Leela	1977-06-25
Tom	Halliwell	1987-08-01
Rachel	Winterbourne	1966-04-26
Jon	Doggett	1951-12-25
Angel	Edwards	1975-08-22
Nino	Andrews	2001-04-21

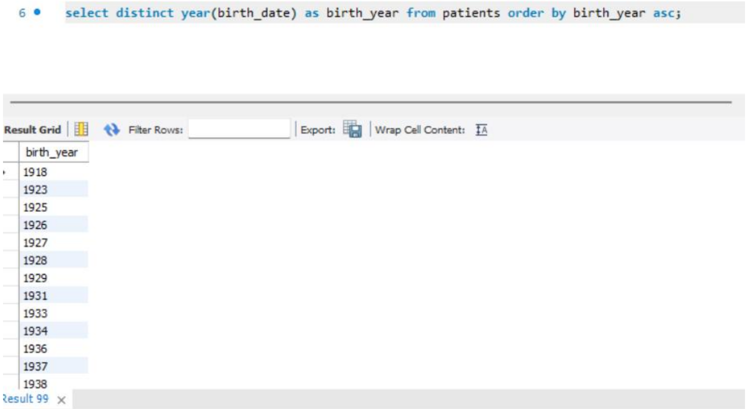
Q.16. Show unique birth years from patients and order them by ascending.

Query:

SELECT DISTINCT year(birth_date) AS birth_year

FROM patients

ORDER BY birth_year ASC;



```
select distinct year(birth_date) as birth_year from patients order by birth_year asc;
```

birth_year
1918
1923
1925
1926
1927
1928
1929
1931
1933
1934
1936
1937
1938

Q.17. Show unique first names from the patients table which only occurs once in the list.

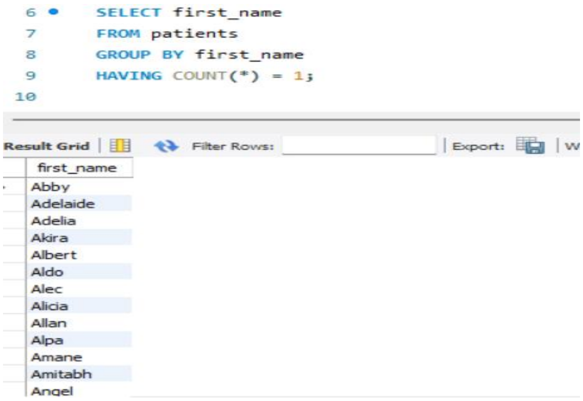
Query:

SELECT first_name

FROM patients

GROUP BY first_name

HAVING COUNT (*) =1;



```
SELECT first_name
FROM patients
GROUP BY first_name
HAVING COUNT(*) = 1;
```

```
first_name
Abby
Adelaide
Adelia
Akira
Albert
Aldo
Alec
Alicia
Allan
Alpa
Amane
Amitabh
Angel
```

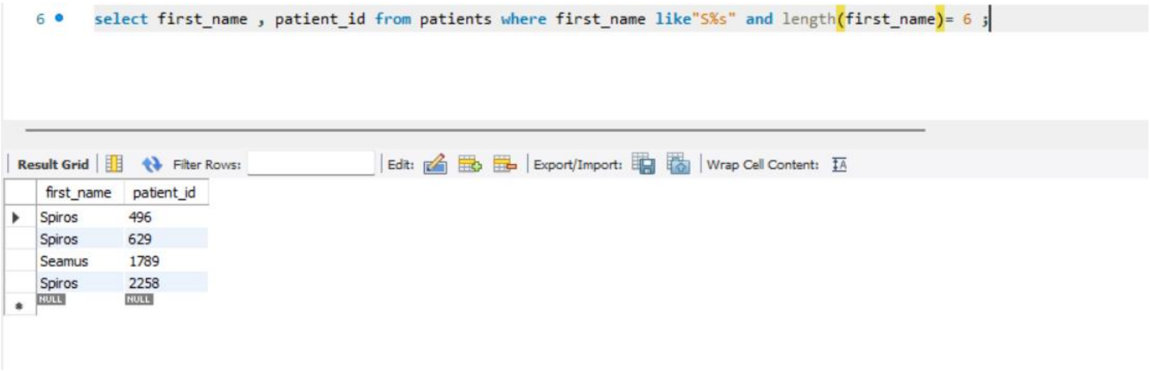
Q.18. Show patient_id and first_name from patients where their first_name start and ends with 's' and is at least 6 characters long.

Query:

SELECT first_name, patient_name,

FROM patients

WHERE first_name LIKE ‘S%s’ AND length(first_name) =6;



```
select first_name , patient_id from patients where first_name like"S%s" and length(first_name)= 6 ;
```

```
first_name  patient_id
Spiros      496
Spiros      629
Seamus      1789
Spiros      2258
NULL        NULL
```

Q.19. Show patient_id, first_name, last_name from patients whose diagnosis is 'Dementia'. Primary diagnosis is stored in the admissions table.

Query:

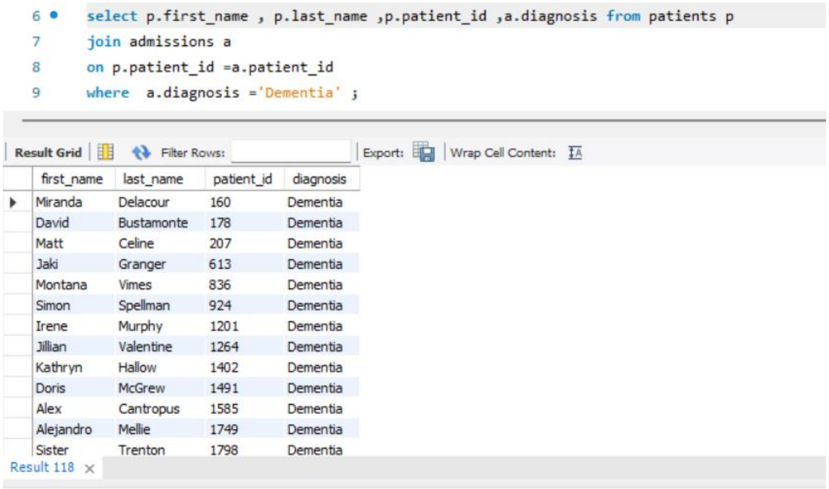
SELECT p.first_name, p.last_name, p.patient_id, a.diagnosis

FROM patients p

JOIN admissions a

ON p.patient_id =a.patient_id

WHERE a.diagnosis=’Dementia’;



```
select p.first_name , p.last_name ,p.patient_id ,a.diagnosis from patients p
join admissions a
on p.patient_id =a.patient_id
where  a.diagnosis ='Dementia' ;
```

first_name	last_name	patient_id	diagnosis
Miranda	Delacour	160	Dementia
David	Bustamonte	178	Dementia
Matt	Celine	207	Dementia
Jaki	Granger	613	Dementia
Montana	Vimes	836	Dementia
Simon	Spellman	924	Dementia
Irene	Murphy	1201	Dementia
Jillian	Valentine	1264	Dementia
Kathryn	Hallow	1402	Dementia
Doris	McGrew	1491	Dementia
Alex	Cantropus	1585	Dementia
Alejandro	Mellie	1749	Dementia
Sister	Trenton	1798	Dementia

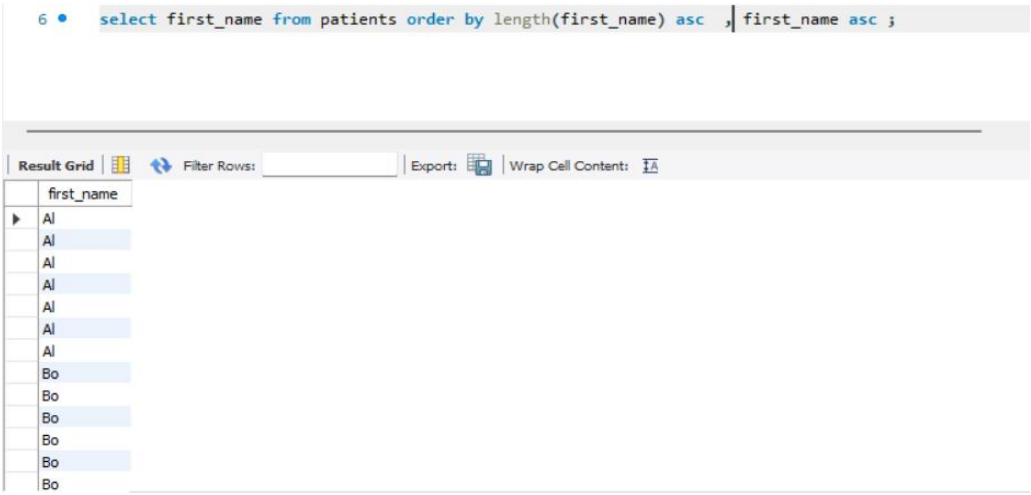
Q.20. Display every patient's first_name. Order the list by the length of each name and then by alphabetically.

Query:

SELECT first_name

FROM patients

ORDER BY length(first_name) ASC, first_name ASC;



```
select first_name from patients order by length(first_name) asc  , first_name asc ;
```

first_name
Al
Al
Al
Al
Al
Al
Al
Bo
Bo
Bo
Bo
Bo
Bo

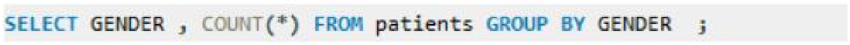
Q.21. Show the total amount of male patients and the total amount of female patients in the patient table. Display the two results in the same row.

Query:

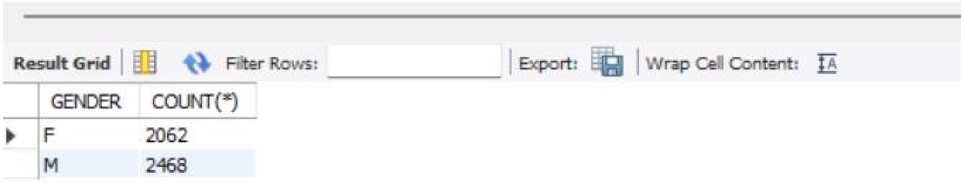
SELECT gender, COUNT (*)

FROM patients

GROUP BY gender;



```
SELECT GENDER , COUNT(*) FROM patients GROUP BY GENDER  ;
```



GENDER	COUNT(*)
F	2062
M	2468

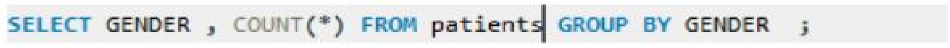
Q.22. Show the total amount of male patients and the total amount of female patients in the patient table. Display the two results in the same row.

Query:

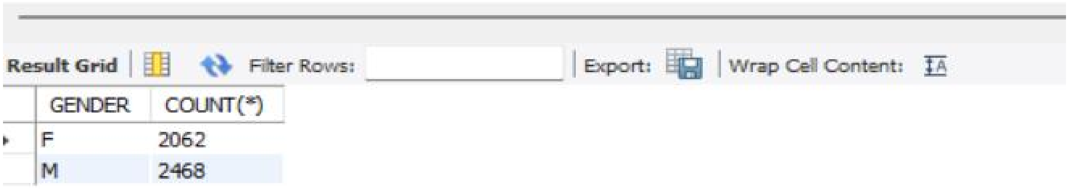
SELECT gender, COUNT (*)

FROM patients

GROUP BY gender;



```
SELECT GENDER , COUNT(*) FROM patients GROUP BY GENDER  ;
```



GENDER	COUNT(*)
F	2062
M	2468

Q.23. Show patient_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

Query:

SELECT p.patient_id, a.diagnosis

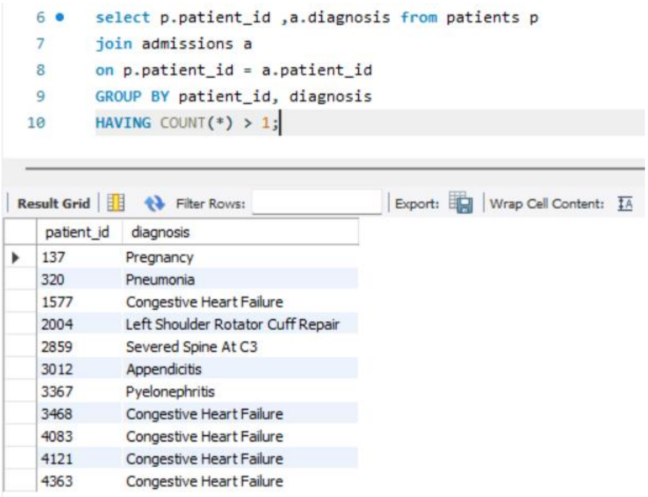
FROM patients p

JOIN admissions a

ON p.patients_id = a.patinet_id

GROUP BY patient_id, diagnosis

HAVING COUNT (*)>1;



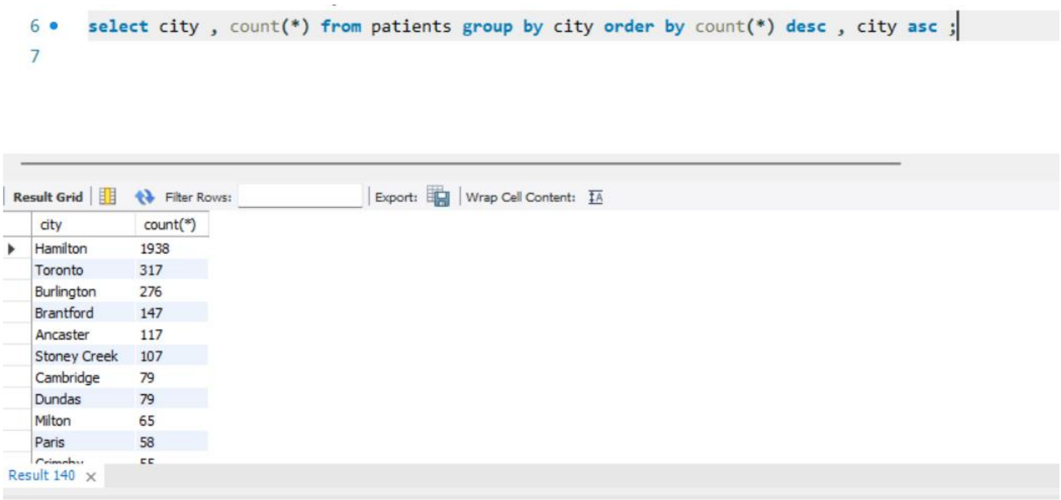
```
select p.patient_id ,a.diagnosis from patients p
join admissions a
on p.patient_id = a.patient_id
GROUP BY patient_id, diagnosis
HAVING COUNT(*) > 1;
```

patient_id	diagnosis
137	Pregnancy
320	Pneumonia
1577	Congestive Heart Failure
2004	Left Shoulder Rotator Cuff Repair
2859	Severed Spine At C3
3012	Appendicitis
3367	Pyelonephritis
3468	Congestive Heart Failure
4083	Congestive Heart Failure
4121	Congestive Heart Failure
4363	Congestive Heart Failure

Q.24. Show the city and the total number of patients in the city. Order from most to least patients and then by city name ascending.

Query:

```
SELECT city, COUNT (*)
FROM patient
GROUP BY city
ORDER BY COUNT (*) DESC, city ASC;
```



```
select city , count(*) from patients group by city order by count(*) desc , city asc ;
```

city	count(*)
Hamilton	1938
Toronto	317
Burlington	276
Brantford	147
Ancaster	117
Stoney Creek	107
Cambridge	79
Dundas	79
Milton	65
Paris	58

Q.25. Show first name, last name and role of every person that is either patient or doctor. The roles are either "Patient" or "Doctor".

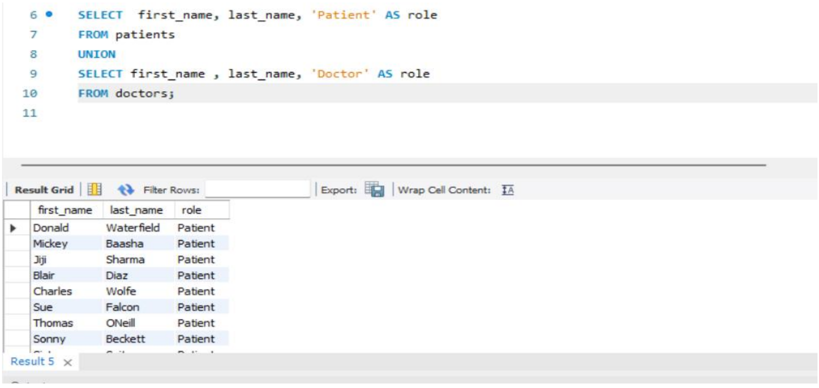
```
SELECT fiest_name, last_name, ‘Patient’ AS role

FROM patient

UNION

SELECT first_name, last_name, ‘Doctor’ AS role

FROM doctors;
```



```
SELECT  first_name, last_name, 'Patient' AS role
FROM patients
UNION
SELECT first_name , last_name, 'Doctor' AS role
FROM doctors;
```

first_name	last_name	role
Donald	Waterfield	Patient
Mickey	Baasha	Patient
Jiji	Sharma	Patient
Blair	Diaz	Patient
Charles	Wolfe	Patient
Sue	Falcon	Patient
Thomas	ONeill	Patient
Sonny	Beckett	Patient

Q.26. Show all allergies ordered by popularity. Remove NULL values from query.

Query:

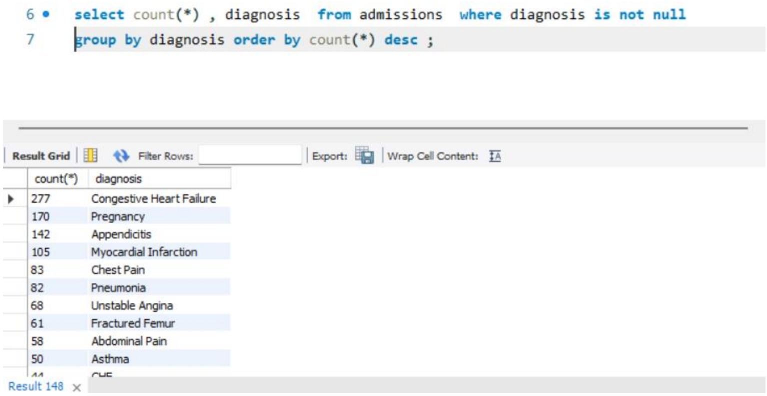
```
SELECT COUNT (*), diagnosis

FROM admissions

WHERE diagnosis IS NOT NULL

GROUP BY diagnosis

ORDER BY COUNT (*) DESC;
```



```
select count(*) , diagnosis  from admissions  where diagnosis is not null
group by diagnosis order by count(*) desc ;
```

count(*)	diagnosis
277	Congestive Heart Failure
170	Pregnancy
142	Appendicitis
105	Myocardial Infarction
83	Chest Pain
82	Pneumonia
68	Unstable Angina
61	Fractured Femur
58	Abdominal Pain
50	Asthma
44	CHF

Q.27. 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.

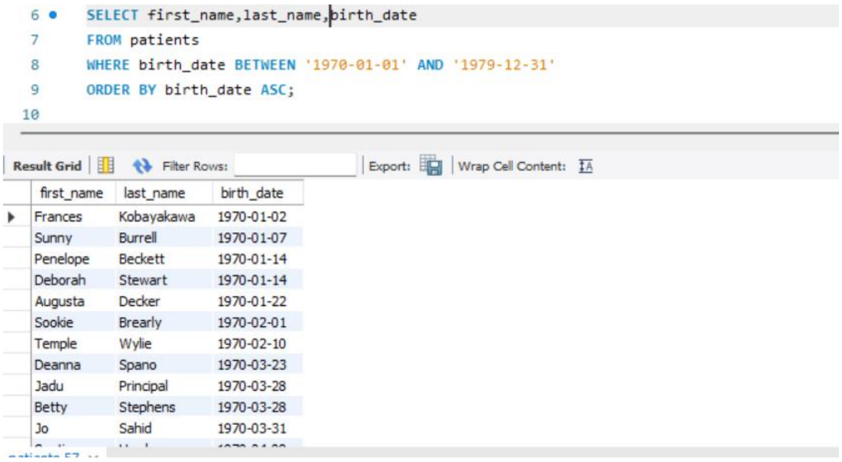
Query:

SELECT first_name, last_name, birth_date

FROM patients

WHERE birth_date BETWEEN ‘1970-01-01’ AND ‘1979-12-31’

ORDER BY birth_date ASC;



```
SELECT first_name,last_name,birth_date
FROM patients
WHERE birth_date BETWEEN '1970-01-01' AND '1979-12-31'
ORDER BY birth_date ASC;
```

first_name	last_name	birth_date
Frances	Kobayakawa	1970-01-02
Sunny	Burrell	1970-01-07
Penelope	Beckett	1970-01-14
Deborah	Stewart	1970-01-14
Augusta	Decker	1970-01-22
Sookie	Brearly	1970-02-01
Temple	Wylie	1970-02-10
Deanna	Spano	1970-03-23
Jadu	Principal	1970-03-28
Betty	Stephens	1970-03-28
Jo	Sahid	1970-03-31

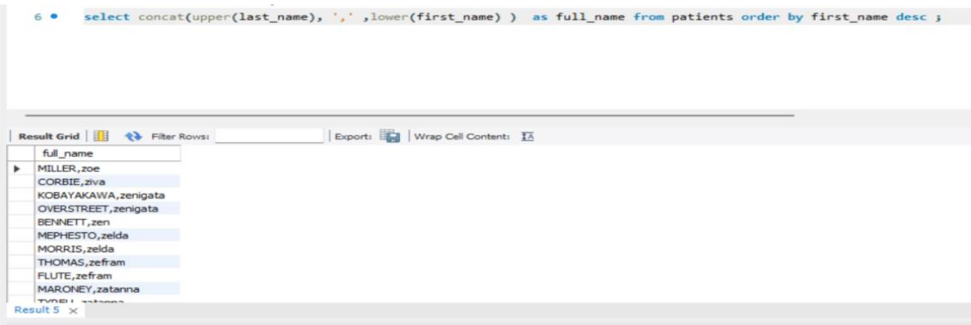
Q.28. We want to display each patient's full name in a single column. Their last_name in all upper letters must appear first, then first_name in all lowercase letters. Separate the last_name and first_name with a comma. Order the list by the first_name in descending order   EX: SMITH, jane

Query:

SELECT concat (upper (last_name), ‘,’, lower (first_name) AS full_name

FROM patients

ORDER BY first_name DESC;



```sql
select concat(upper(last_name), ',' ,lower(first_name) )  as full_name from patients order by first_name desc ;
```

```
full_name
MILLER,zoe
CORBIE,ziva
KOBAYAKAWA,zenigata
OVERSTREET,zenigata
BENNETT,zen
MEPHESTO,zelda
MORRIS,zelda
THOMAS,zefram
FLUTE,zefram
MARONEY,zatanna
```

Q.29. 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.

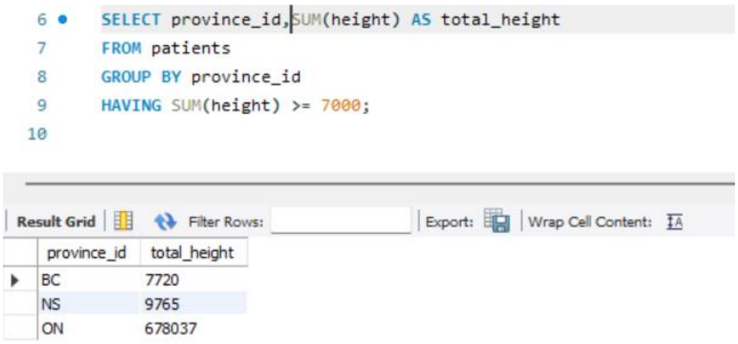
Query:

SELECT province_id, SUM (height) AS total_height

FROM patients

GROUP BY province_id

HAVING SUM (height) >= 7000;



```sql
SELECT province_id,SUM(height) AS total_height
FROM patients
GROUP BY province_id
HAVING SUM(height) >= 7000;
```

```
province_id  total_height
BC           7720
NS           9765
ON           678037
```

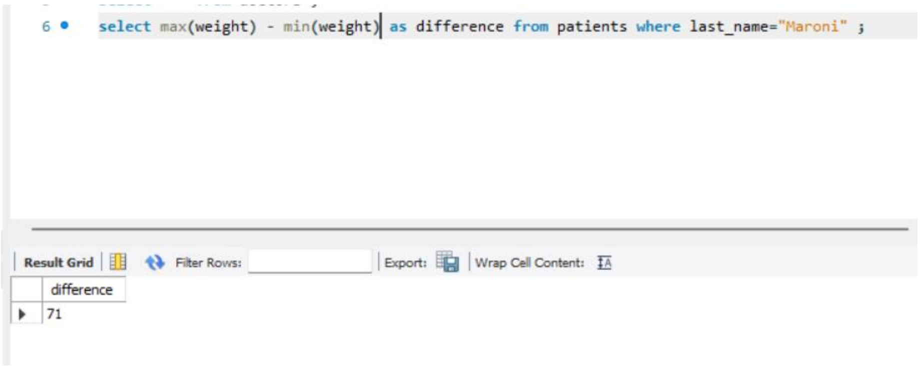
Q.30. Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni’.

Query:

SELECT MAX (weight) – MIN (weight) AS difference

FROM patients

WHERE last_name=’Maroni’;



```
select max(weight) - min(weight) as difference from patients where last_name="Maroni" ;
```

difference
71

Q.31. 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.

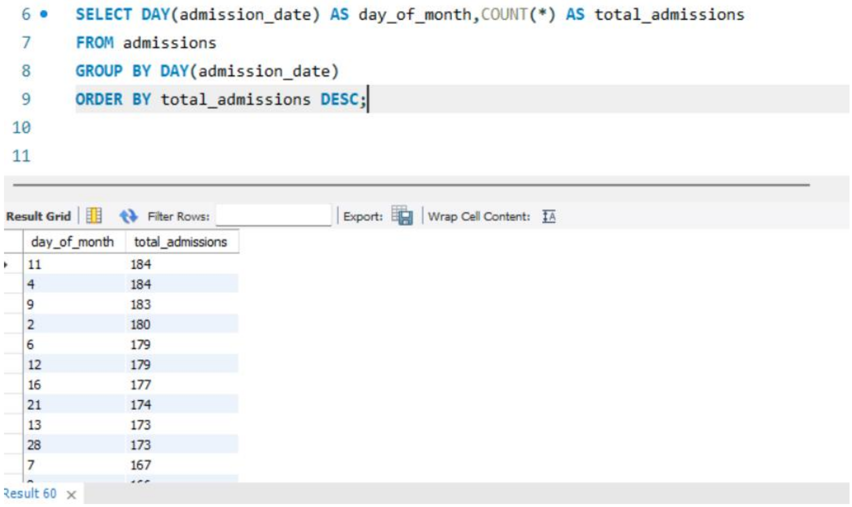
Query:

```
SELECT DAY (admission_date) AS day_of_month, COUNT (*) AS total_admissions

FROM admissions

GROUP BY DAY (admission_date)

ORDER BY total_admissions DESC;
```



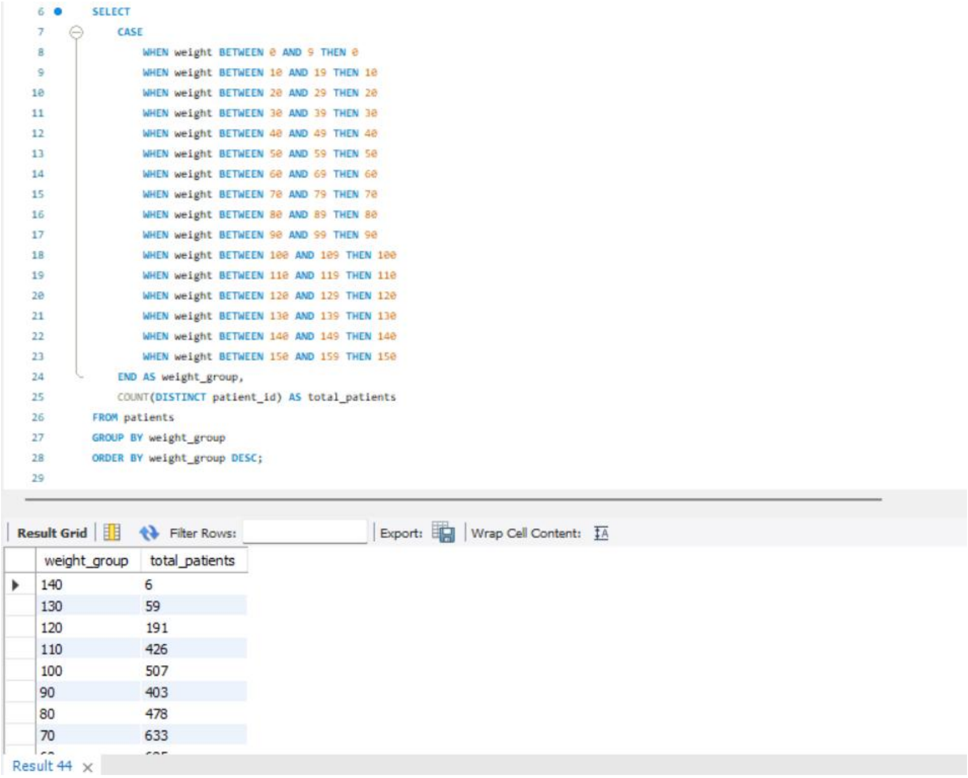
```
SELECT DAY(admission_date) AS day_of_month,COUNT(*) AS total_admissions
FROM admissions
GROUP BY DAY(admission_date)
ORDER BY total_admissions DESC;
```

day_of_month	total_admissions
11	184
4	184
9	183
2	180
6	179
12	179
16	177
21	174
13	173
28	173
7	167

Q.32. 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 descending. e.g. if they weight 100 to109 they are placed in the 100-weight group, 110-119 = 110 weight group, etc.

Query:

```
SELECT
  CASE
    WHEN weight BETWEEN 0 AND 9 THEN 0
    WHEN weight BETWEEN 10 AND 19 THEN 10
    WHEN weight BETWEEN 20 AND 29 THEN 20
    WHEN weight BETWEEN 30 AND 39 THEN 30
    WHEN weight BETWEEN 40 AND 49 THEN 40
    WHEN weight BETWEEN 50 AND 59 THEN 50
    WHEN weight BETWEEN 60 AND 69 THEN 60
    WHEN weight BETWEEN 70 AND 79 THEN 70
    WHEN weight BETWEEN 80 AND 89 THEN 80
    WHEN weight BETWEEN 90 AND 99 THEN 90
    WHEN weight BETWEEN 100 AND 109 THEN 100
    WHEN weight BETWEEN 110 AND 119 THEN 110
    WHEN weight BETWEEN 120 AND 129 THEN 120
    WHEN weight BETWEEN 130 AND 139 THEN 130
    WHEN weight BETWEEN 140 AND 149 THEN 140
    WHEN weight BETWEEN 150 AND 159 THEN 150
  END AS weight_group,
  COUNT (DISTINCT patient_id) AS total_patients
FROM patients
GROUP BY weight_group
ORDER BY weight_group DESC;
```



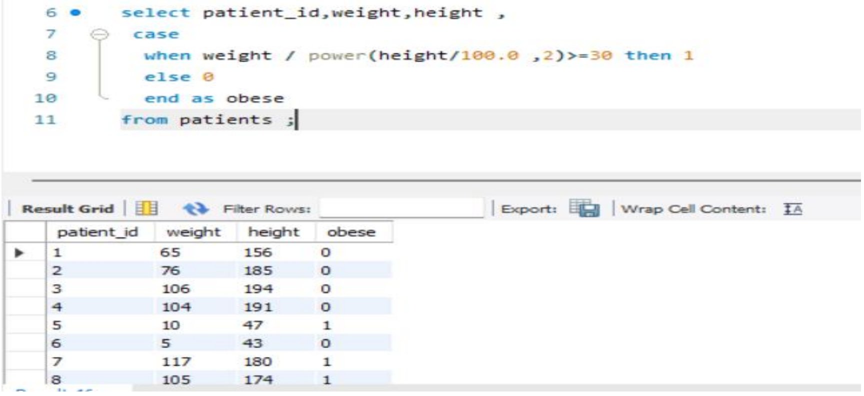
```sql
SELECT
    CASE
        WHEN weight BETWEEN 0 AND 9 THEN 0
        WHEN weight BETWEEN 10 AND 19 THEN 10
        WHEN weight BETWEEN 20 AND 29 THEN 20
        WHEN weight BETWEEN 30 AND 39 THEN 30
        WHEN weight BETWEEN 40 AND 49 THEN 40
        WHEN weight BETWEEN 50 AND 59 THEN 50
        WHEN weight BETWEEN 60 AND 69 THEN 60
        WHEN weight BETWEEN 70 AND 79 THEN 70
        WHEN weight BETWEEN 80 AND 89 THEN 80
        WHEN weight BETWEEN 90 AND 99 THEN 90
        WHEN weight BETWEEN 100 AND 109 THEN 100
        WHEN weight BETWEEN 110 AND 119 THEN 110
        WHEN weight BETWEEN 120 AND 129 THEN 120
        WHEN weight BETWEEN 130 AND 139 THEN 130
        WHEN weight BETWEEN 140 AND 149 THEN 140
        WHEN weight BETWEEN 150 AND 159 THEN 150
    END AS weight_group,
    COUNT(DISTINCT patient_id) AS total_patients
FROM patients
GROUP BY weight_group
ORDER BY weight_group DESC;
```

weight_group	total_patients
140	6
130	59
120	191
110	426
100	507
90	403
80	478
70	633

Q.33. Show patient_id, weight, height, is Obese from the patient table. Display is Obese as a Boolean 0 or 1. Obese is defined as weight(kg)/(height(m). Weight is in units kg. Height is in units cm.

Query:

```
SELCET patient_id, weight, height,
    CASE
        WHEN weight / power (height/100.0, 2)>=30 THEN 1
        ELSE 0
        END AS obese
    FROM patients;
```



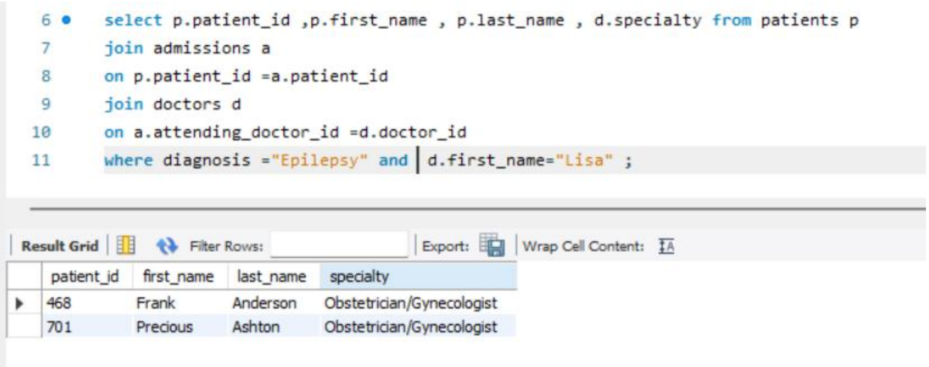
```sql
select patient_id,weight,height ,
  case
    when weight / power(height/100.0 ,2)>=30 then 1
    else 0
    end as obese
from patients ;
```

patient_id	weight	height	obese
1	65	156	0
2	76	185	0
3	106	194	0
4	104	191	0
5	10	47	1
6	5	43	0
7	117	180	1
8	105	174	1

Q.34. 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.

Query:

```
SELECT p.patient_id, p.first_name, p.last_namr, d.speciality
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 diagnosis = ‘Epilepsy’ AND d.first_name=’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 diagnosis ="Epilepsy" and d.first_name="Lisa" ;
```

patient_id	first_name	last_name	specialty
468	Frank	Anderson	Obstetrician/Gynecologist
701	Precious	Ashton	Obstetrician/Gynecologist

INSIGHTS

- The database contains **4,530 unique patients** and **27 doctors**, with a total of **5,067 admission records**, indicating repeat hospital visits for several patients.

- Gender analysis shows **2,468 male patients** and **2,062 female patients**, reflecting a slightly higher male patient population.

- The most frequently reported diagnosis is **Congestive Heart Failure (277 cases)**, followed by **Pregnancy (170 cases)** and **Appendicitis (142 cases)**.

- A total of **481 admissions** show patients being admitted and discharged on the **same day**, indicating short-duration or emergency treatments.

- There are **11 cases** where patients were admitted **multiple times for the same diagnosis**, suggesting chronic or recurring medical conditions.

- City-wise analysis shows **Hamilton** has the highest number of patients (**1,938 patients**), followed by **Toronto (317)** and **Burlington (276)**.

- Obesity analysis based on BMI calculation reveals **2,222 patients** fall under the **obese category**, highlighting a major health risk trend.

- Age distribution analysis shows **55 patients** were born in the year **2010**, representing younger age-group admissions.

- Province-based mapping confirms patient data is spread across **13 provinces**, enabling region-wise healthcare analysis.

- Doctor-admission linkage shows that patient diagnoses are distributed across multiple medical specialties, supporting comprehensive treatment coverage.

CHALLENGES

Challenges faced during this project :

1. **Challenge:** Repeating Query Errors
   How We Overcame: We carefully analysed SQL error messages, executed queries step-by-step, and tested individual clauses separately to identify and fix syntax and logical mistakes.

2. **Challenge:** Query Optimization and Method Selection
   How We Overcame: Different approaches were evaluated, and the most efficient method was chosen by optimizing joins, reducing unnecessary subqueries, and improving overall query performance.

3. **Challenge:** Database Design and Relationship Issues
   How We Overcame: Tables were redesigned using normalization techniques, and correct primary and foreign key relationships were implemented to ensure data integrity.

4. **Challenge:** Handling Inconsistent and Missing Data
   How We Overcame: Data cleaning techniques such as filtering NULL values, removing duplicates, and validating data formats were applied to maintain accuracy.

5. **Challenge:** Software and Tool-Related Issues
   How We Overcame: Software issues were resolved by updating the SQL environment, restarting connections, and maintaining backups to prevent data loss.

6. **Challenge:** Complex Join Conditions
   How We Overcame: Join logic was tested incrementally, and table relationships were thoroughly analysed to avoid duplicate or missing records.

7. **Challenge:** Debugging Logical Errors
   How We Overcame: Query results were verified using sample data and manual checks to ensure the logic produced correct and meaningful outputs.

8. **Challenge:** Learning Advanced SQL Concepts
   How We Overcame: Advanced SQL features were understood through documentation, practice queries, and gradual implementation within the project.

SUMMARY

The Medical History SQL project demonstrates effective use of relational databases to store, manage, and analyse healthcare data. By integrating patient, doctor, admission, and province information, the project provides meaningful insights into patient demographics, disease patterns, hospital admissions, and regional healthcare distribution.

The analysis highlights key trends such as frequently diagnosed conditions, repeat admissions, obesity prevalence, and city-wise patient concentration.

Overall, the project showcases practical application of SQL concepts including joins, aggregations, filtering, and data normalization, emphasizing the importance of structured data analysis in healthcare decision-making.