

DATAMITES - INTERNSHIP

MEDICAL DATA HISTORY

Project ID : **PRSQL-02 MEDICAL DATA HISTORY**

Project Team ID : **PTID-CDA-SEP-25-761**

Database Management System(DBMS) : **SQL SERVER**

Language : **SQL**

Project By : **ARUNPANDIYAN R.A**

TABLE OF CONTENTS

	Page No
<i>INTRODUCTION</i>	3
<i>PROJECT OVERVIEW</i>	3
<i>QUERIES NEED TO BE PERFORMED</i>	4
<i>EXPLORING THE DATABASE</i>	6
<i>QUERIES AND THEIR OUTPUTS</i>	11
<i>CONCLUSION</i>	28

INTRODUCTION:

This ‘Medical Data History’ project discusses about history of patients, doctors and their diagnosis in the provinces of Canada.

With all the data’s regarding patients and doctors we need to explore relevant queries so that we can find some valuable insights based on their history. **SQL** is used in this project for all the ‘data interpretations and findings’.

Let’s explore more deeper in this project step by step and will explore their history and get into conclusions.

PROJECT OVERVIEW:

- In this project ‘Medical Data History’ there’s a database provided called **project_medical_data_history** which consists of certain tables like **admissions, doctors, patients and province_names**.
- The **admissions** table consists data’s like
 - patient_id
 - admission_date
 - discharge_date
 - diagnosis
 - attending_doctor_id
- The **doctors** table consists data’s like
 - doctor_id
 - first_name
 - last_name
 - speciality
- The **patients** table consists data’s like
 - patient_id
 - first_name
 - last_name
 - gender
 - birth_date
 - city
 - province_id
 - allergies
 - height
 - weight
- The **province_names** table consists data’s like
 - province_id
 - province_name

With all the data’s in the tables **admissions, doctors, patients, province_names** we need to interpret and find the answers for the queries.

QUERIES NEED TO BE PERFORMED:

1. Show first name, last name, and gender of patients who's gender is 'M'
2. Show first name and last name of patients who does not have allergies.
3. Show first name of patients that start with the letter 'C'
4. Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)
5. Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'
6. Show first name and last name concatenated into one column to show their full name.
7. Show first name, last name, and the full province name of each patient.
8. Show how many patients have a birth_date with 2010 as the birth year.
9. Show the first_name, last_name, and height of the patient with the greatest height.
10. Show all columns for patients who have one of the following patient_ids: 1,45,534,879,1000
11. Show the total number of admissions
12. Show all the columns from admissions where the patient was admitted and discharged on the same day.
13. Show the total number of admissions for patient_id 579.
14. Based on the cities that our patients live in, show unique cities that are in province_id 'NS'?
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
16. Show unique birth years from patients and order them by ascending.

17. 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. Tip: HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

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.

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

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

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

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

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

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

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

26. 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.

27. 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

28. 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.

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

30. 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.

31. 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. e.g. if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

32. Show patient_id, weight, height, isObese from the patients table. Display isObese as a boolean 0 or 1. Obese is defined as $\text{weight(kg)}/(\text{height(m)})$. Weight is in units kg. Height is in units cm.

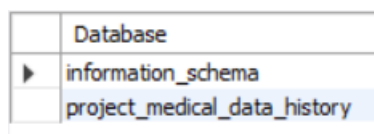
33. Show patient_id, first_name, last_name, and attending doctor's specialty. Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'. Check patients, admissions, and doctors tables for required information.

EXPLORING THE DATABASE:

First we have to Load the database to get a clear information on the data's present in it. We'll explore the steps one by one with SQL queries.

To load the database We have connected to SQL SERVER first to access the database and to perform queries, with the provided login credentials we connected to our project via SQL SERVER database, then we dived deeper into the database exploration which follows the steps.

- `SHOW databases;`



	Database
▶	information_schema
	project_medical_data_history

It displays the database name – 'project_medical_data_history'.

- `USE project_medical_data_history;`

In this query `USE` is used to select a specific database 'project_medical_data_history' to work with in a database management system (DBMS).

- SHOW tables;

	Tables_in_project_medical_data_history
►	admissions
	doctors
	patients
	province_names

This query **displays the tables present in the database**. The displayed tables are **admissions, doctors, patients, province_names,**

We will get use of these tables and dive deeper to find insights with the help of the data's present in the tables.

- SELECT * FROM admissions LIMIT 5;

	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
	1	2018-11-06	2018-11-08	Ovarian Dermoid-Cyct	21
	3	2018-10-21	2018-10-27	Congestive Heart Failure	8
	3	2019-01-24	2019-01-29	Cardiac Arrest	2
	6	2018-06-13	2018-06-15	Asthma Exacerbation	3
✱	NULL	NULL	NULL	NULL	NULL

It displays the columns and the data's present in the table with the first five rows. It displays the data's of **patient_id, admission_date, discharge_date, diagnosis, attending_doctor_id.**

- SELECT patient_id, admission_date, discharge_date, diagnosis, attending_doctor_id as doctor_id FROM admissions LIMIT 5;

	patient_id	admission_date	discharge_date	diagnosis	doctor_id
►	1	2018-09-20	2018-09-20	Ineffective Breathin Pattern R/T Fluid Accumulatio	24
	1	2018-11-06	2018-11-08	Ovarian Dermoid-Cyct	21
	3	2018-10-21	2018-10-27	Congestive Heart Failure	8
	3	2019-01-24	2019-01-29	Cardiac Arrest	2
	6	2018-06-13	2018-06-15	Asthma Exacerbation	3
✱	NULL	NULL	NULL	NULL	NULL

It displays the columns and the data's present in the table with the first five rows. It displays the data's of **patient_id, admission_date, discharge_date, diagnosis,** and replaces **attending_doctor_id** as **doctor_id.**

- `SELECT count(*) FROM admissions;`

	count(*)
►	5067

It displays the **no of rows present** in the **admissions** table as **5067**.

- `SELECT * FROM doctors LIMIT 5;`

	doctor_id	first_name	last_name	specialty
►	1	Claude	Walls	Internist
	2	Joshua	Green	Cardiologist
	3	Miriam	Tregre	General Surgeon
	4	James	Russo	Obstetrician/Gynecologist
	5	Scott	Hill	Gastroenterologist
*	NULL	NULL	NULL	NULL

It displays the columns and the data's present in the **doctors** table with the first five rows. It displays the data's of **doctor_id**, **first_name**, **last_name**, **speciality**.

- `SELECT count(*) FROM doctors;`

	count(*)
►	27

It displays the **no of rows present** in the **doctors** table as **27**.

- `SELECT * FROM patients LIMIT 5;`

	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
	2	Mickey	Baasha	M	1981-05-28	Dundas	ON	Sulfa	185	76
	3	Jiji	Sharma	M	1957-09-05	Hamilton	ON	Penicillin	194	106
	4	Blair	Diaz	M	1967-01-07	Hamilton	ON	NULL	191	104
	5	Charles	Wolfe	M	2017-11-19	Orillia	ON	Penicillin	47	10
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

It displays the columns and the data's present in the **patients** table with the first five rows. It displays the data's of **patient_id**, **first_name**, **last_name**, **gender**, **birth_date**, **city**, **province_id**, **allergies**, **height**, **weight**.

- SELECT count(*) FROM patients;

	count(*)
▶	4530

It displays the **no of rows present** in the **patients** table as **4530**.

- SELECT * FROM province_names LIMIT 5;

	province_id	province_name
▶	AB	Alberta
	BC	British Columbia
	MB	Manitoba
	NB	New Brunswick
	NL	Newfoundland and Labrador
•	NULL	NULL

It displays the columns and the data's present in the **province_names** table with the first five rows. It displays the data's of **province_id**, **province_name**.

- SELECT count(*) FROM province_names;

	count(*)
▶	13

It displays the **no of rows present** in the **province_names** table as **13**.

- DESCRIBE admissions;

	Field	Type	Null	Key	Default	Extra
▶	patient_id	int(11)	NO	PRI	NULL	
	admission_date	date	NO	PRI	NULL	
	discharge_date	date	YES		NULL	
	diagnosis	varchar(50)	YES		NULL	
	attending_doctor_id	int(11)	YES		NULL	

It describes about the columns present in the table **admissions** and describes the datatypes of the fields, NULL values and the Primary Key columns present (int – integer, date – YYYY-MM-DD, varchar -variables and characters) . The values (11), (50) represents the space allocated for those integers, variables and characters.

- DESCRIBE doctors;

	Field	Type	Null	Key	Default	Extra
▶	doctor_id	int(11)	NO	PRI	HULL	
	first_name	varchar(130)	YES		HULL	
	last_name	varchar(130)	YES		HULL	
	specialty	varchar(125)	YES		HULL	

It describes about the columns present in the table **doctors** and describe the datatypes of the fields, NULL values and the primary Key columns present (int – integer, date – YYYY-MM-DD, varchar -variables and characters). The values (130) (125) represents the space allocated for the first_name, last_name and specialty.

- DESCRIBE patients;

	Field	Type	Null	Key	Default	Extra
▶	patient_id	int(11)	NO	PRI	HULL	
	first_name	varchar(150)	YES		HULL	
	last_name	varchar(150)	YES		HULL	
	gender	char(1)	YES		HULL	
	birth_date	date	YES		HULL	
	city	varchar(150)	YES		HULL	
	province_id	char(2)	YES		HULL	
	allergies	varchar(150)	YES		HULL	
	height	decimal(3,0)	YES		HULL	
	weight	decimal(4,0)	YES		HULL	

It describes about the columns present in the table **patients** and describe the datatypes of the fields, NULL values and the primary Key columns present (int – integer, date – YYYY-MM-DD, varchar -variables and characters, decimal – values in decimals). The values (11) (150) (3,0) (4,0) represents the space allocated for the respective fields.

- DESCRIBE province_names;

	Field	Type	Null	Key	Default	Extra
▶	province_id	char(2)	NO	PRI	HULL	
	province_name	varchar(30)	NO		HULL	

It describes about the columns present in the table **province_names** and describe the datatypes of the fields, NULL values and the primary Key columns present (char – characters, varchar -variables and characters). The values (2) (30) represents the space allocated for the respective fields.

So, this is how we get the overall view of the respective fields among the tables **admissions**, **doctors**, **patients**, **province_names** and their datatypes.

The Querying process follows from next step using the available tables and the fields.

QUERIES AND THEIR OUTPUTS:

1. Show first name, last name, and gender of patients who's gender is 'M'.

- `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

.... Table continues

Explanation: Filters and displays the 'first_name', 'last_name' and the 'gender' of the patients from **patients** table.

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

- `SELECT first_name, last_name
FROM patients
WHERE allergies IS NULL OR allergies = '';`

	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

.... Table continues

Explanation: Filters and displays the 'first_name', 'last_name' of the patients from the **patients** table who doesn't have allergies.

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

- `SELECT first_name`
`FROM patients`
`WHERE first_name LIKE "C%";`

	first_name
▶	Charles
	Cedric
	Charles
	Cross
	Calleigh
	Catherine
	Caroline
	Casanova
	Chen
	Charmian
	Cary
	Christine

....Table continues

Explanation: Filters the first_name which startws with letter 'c' from **patients** table.

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

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

	first_name	last_name
▶	Jiji	Sharma
	Blair	Diaz
	Thomas	ONeill
	Sonny	Beckett
	Tom	Halliwell
	Jon	Doggett
	Angel	Edwards
	John	Farley
	Temple	Russert
	Don	Edwards
	Gary	Ramotswe
	Phil	Chester

... Table continues

Explanation: Filters the first_name and last_name of the patients whose weight is between 100 to 120 from **patients** table.

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

- `SELECT patient_id, first_name, last_name, birth_date, gender, height, weight, province_id, city, CASE WHEN allergies IS NULL THEN 'NKA' ELSE allergies END AS allergies FROM patients;`

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

...Table continues

Explanation: We can't able to update the table in the database permanently since the database is in Read-Only format.

But we can able to show a table by Replacing patients whose allergies is NULL with NKA.

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

- `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 O'Neill
	Sonny Beckett
	Sister Spitzer
	Cedric Coltrane
	Hank Spencer
	Sara di Marco

... Table continues

Explanation: It combines the names present in 'first_name' 'last_name' and displays 'full_name' as output.

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

- SELECT p.first_name, p.last_name, pr.province_name
FROM patients p
join province_names pr on p.province_id=pr.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	O'Neill	Ontario
	Sonny	Beckett	Nova Scotia
	Sister	Spitzer	Ontario
	Cedric	Coltrane	Ontario
	Hank	Spencer	Ontario
	Sara	di Marco	Ontario

... Table continues

Explanation: It joins 'province_id' from 'province_names' table to 'patients' table's 'province_id' because 'province_id' is the primary key in both the tables. Then it filters 'province_name' and displays along with 'first_name', 'last_name'.

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

- SELECT COUNT(*) AS patient_count
FROM patients

WHERE YEAR(birth_date) = 2010;

	patient_count
▶	55

Explanation: It displays the total count of the patients who born in the year 2010.

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

- SELECT first_name, last_name, height
FROM patients
WHERE height = (SELECT MAX(height) FROM patients);

	first_name	last_name	height
▶	Sam	Haruko	226

Explanation: It displays the **first_name**, **last_name** and **height** of the patient with greatest height(cm).

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

- SELECT * FROM patients
WHERE patient_id IN (1,45,534,879,1000);

	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

Explanation: It filters and displays all the columns in the patients table whose patient id's are **1,45,534,879,1000**.

11. Show the total number of admissions

- SELECT count(*) from admissions;

	count(*)
▶	5067

Explanation: It counts the total number of admissions and displays the value.

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

- `SELECT * FROM admissions`
`WHERE DATE(admission_date) = 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 Arthritis Knee	5

Explanation: It filters and displays all the columns from the **admissions** table by which patients was admitted and discharged in the same day.

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

- `SELECT count(*) AS total_count_in_id_579`
`FROM admissions`
`WHERE patient_id=579;`

	total_count_in_id_579
▶	2

Explanation: It filters and displays the total number of admissions in the **patient_id** 579.

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

- `SELECT DISTINCT city`
`FROM patients`
`WHERE province_id='NS';`

	city
▶	Port Hawkesbury
	Halifax
	Inverness

Explanation: It filters and displays the unique cities that are in the **province_id** 'NS'.

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.

- `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

... Table continues

Explanation: It filters and displays **first_name**, **last_name** and **birth_date** of the patients who have height more than 160 and weight more than 70.

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

- `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

... Table continues

Explanation: It filters and displays **birth_year** of the patients from the **patients** table in ascending order.

17. 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. Tip: HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

- ```
SELECT COUNT(*) AS unique_first_name_count
FROM (
 SELECT first_name
 FROM patients
 GROUP BY first_name
 HAVING COUNT(*) = 1
) AS unique_names;
```

|   | unique_first_name_count |
|---|-------------------------|
| ▶ | 319                     |

**Explanation:** It filters the **unique first\_name** of patients from **patients** table which doesn't repeat again and counts the total and displays the value.

**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.**

- ```
SELECT patient_id, first_name
```

```
FROM patients
WHERE first_name LIKE 's%s' AND LENGTH(first_name) >= 6;
```

	patient_id	first_name
▶	496	Spiros
	629	Spiros
	648	Stanislaus
	1273	Stanislaus
	1789	Seamus
	1926	Stanislaus
	1996	Stanislaus
	2258	Spiros
	2378	Stanislaus
	2771	Stanislaus
	3487	Stanislaus
★	HULL	HULL

Explanation: It filters and displays the **patient_id** and **first_name** of the patients from **patients** table whose first_name starts and ends with 's' which has atleast 6 characters in length.

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

- ```
SELECT p.patient_id, p.first_name, p.last_name
FROM patients p
join admissions a ON p.patient_id = a.patient_id
WHERE a.diagnosis = 'Dementia';
```

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

**Explanation:** It filters and displays **patient\_id**, **first\_name** and **last\_name** of the patients from **patients** table who diagnosis is 'Dementia'.

**20. Display every patient's first\_name. Order the list by the length of each name and then by alphabetically.**

- ```
SELECT first_name
FROM patients
ORDER BY LENGTH(first_name), first_name;
```

	first_name
▶	Al
	Al
	Al
	Al
	Al
	Al
	Al
	Bo
	Bo
	Bo
	Bo
	Bo
	Bo
	...

... Table continues

Explanation: It filters and displays the **first_name** of the patients from **patients** table based on their length and also alphabetical order.

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

- ```
SELECT
SUM(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) AS
male_count,
SUM(CASE WHEN gender = 'F' THEN 1 ELSE 0 END) AS
female_count
FROM patients;
```

|   | male_count | female_count |
|---|------------|--------------|
| ▶ | 2468       | 2062         |

**Explanation:** It filters and displays the total count of male and female patients from **patients** table.

22. Show patient\_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

- SELECT patient\_id, diagnosis  
FROM admissions  
GROUP BY patient\_id, diagnosis  
HAVING COUNT(\*) > 1;

|   | 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          |

**Explanation:** It filters **patient\_id** and **diagnosis** from patient table who admitted multiple times for the same diagnosis.

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

- SELECT city, COUNT(\*) AS patient\_count  
FROM patients  
GROUP BY city  
ORDER BY patient\_count DESC, city ASC;

|   | city         | patient_count |
|---|--------------|---------------|
| ▶ | Hamilton     | 1938          |
|   | Toronto      | 317           |
|   | Burlington   | 276           |
|   | Brantford    | 147           |
|   | Ancaster     | 117           |
|   | Stoney Creek | 107           |
|   | Cambridge    | 79            |
|   | Dundas       | 79            |
|   | Milton       | 65            |
|   | Paris        | 58            |
|   | Grimsby      | 55            |
|   | Timmins      | 53            |
|   | Ottawa       | 51            |

.... Table continues

**Explanation:** It filters **city** and **patient\_count** from **patients** table, groups and aggregates the city based on the **patient\_count** in Descending Order.  
The group of the city changes based on the order of **patient\_count** only.

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

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

|  | first_name | last_name | role    |
|--|------------|-----------|---------|
|  | Faye       | Soong     | Patient |
|  | Christine  | Turtle    | Patient |
|  | Armando    | Matthews  | Patient |
|  | Karen      | Costello  | Patient |
|  | Deanna     | Barnes    | Patient |
|  | Molly      | Hanover   | Patient |
|  | Constance  | Yang      | Patient |
|  | Claude     | Walls     | Doctor  |
|  | Joshua     | Green     | Doctor  |
|  | Miriam     | Tregre    | Doctor  |
|  | James      | Russo     | Doctor  |
|  | Scott      | Hill      | Doctor  |
|  | Tasha      | Phillips  | Doctor  |

.... Table continues

**Explanation:** It filters and displays the **first\_name**, **last\_name** and **role** either its **patient** or **doctor**.

**25. Show all allergies ordered by popularity. Remove NULL values from query.**

- `SELECT allergies, COUNT(*) AS allergy_count  
FROM patients  
WHERE allergies IS NOT NULL AND allergies != 'NKA'  
GROUP BY allergies  
ORDER BY allergy_count DESC;`

|   | allergies     | allergy_count |
|---|---------------|---------------|
| ► | Penicillin    | 1082          |
|   | Codeine       | 305           |
|   | Sulfa         | 157           |
|   | ASA           | 99            |
|   | Sulfa Drugs   | 71            |
|   | Peanuts       | 52            |
|   | Iodine        | 48            |
|   | Tylenol       | 42            |
|   | Bee Stings    | 40            |
|   | Valporic Acid | 38            |
|   | Tetracycline  | 34            |
|   | Wheat         | 33            |
|   | Peanut Butter | 29            |

.... Table continues

**Explanation:** It filters and displays the **allergies** and **allergy\_count** from **patients** table by grouping the allergies and ordering **allergy\_count** in descending order. (It removes the NULL values and gives the results)

**26. Show all patient's first\_name, last\_name, and birth\_date who were born in the 1970s decade. Sort the list starting from the earliest birth\_date.**

- `SELECT first_name, last_name, birth_date  
FROM patients  
WHERE YEAR(birth_date) BETWEEN 1970 AND 1979  
ORDER BY birth_date ASC;`

|   | 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     | Brearily   | 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 |
|   | Santino    | Hawker     | 1970-04-08 |
|   | Andy       | Guarnaccia | 1970-04-10 |

... Table continues

**Explanation:** It filters and displays the **first\_name**, **last\_name** and **birth\_date** who born in the 1970's and sorts their date of birth in ascending order.

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

- SELECT CONCAT(UPPER(last\_name), ', ', LOWER(first\_name)) AS full\_name  
FROM patients  
ORDER BY first\_name DESC;

|   | full_name           |
|---|---------------------|
| ▶ | MILLER,zoe          |
|   | CORBIE,ziva         |
|   | KOBAYAKAWA,zenigata |
|   | OVERSTREET,zenigata |
|   | BENNETT,zen         |
|   | MEPHESTO,zelda      |
|   | MORRIS,zelda        |
|   | THOMAS,zefram       |
|   | FLUTE,zefram        |
|   | MARONEY,zatanna     |
|   | TYRELL,zatanna      |
|   | CHE,zane            |
|   | WONG,zane           |

.... Table continues

**Explanation:** It filters and displays the **first\_name**, **last\_name** by the order first\_name in upper case separated by comma and last\_name in lowercase in descending order.

28. Show the province\_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000.

- SELECT province\_id, SUM(height) AS total\_height  
FROM patients  
GROUP BY province\_id  
HAVING SUM(height) >= 7000;

|   | province_id | total_height |
|---|-------------|--------------|
| ▶ | BC          | 7720         |
|   | NS          | 9765         |
|   | ON          | 678037       |

**Explanation:** It filters and displays the **province\_id** and **total\_height** by grouping the **province\_id** based on the height greater than equal to 7000.



**29. Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'**

- `SELECT MAX(weight) - MIN(weight) AS weight_difference  
FROM patients  
WHERE last_name = 'Maroni';`

|   | weight_difference |
|---|-------------------|
| ▶ | 71                |

**Explanation:** It filters and displays the **maximum weight** and **minimum weight** whose last name is 'Maroni'.

**30. Show all of the days of the month (1-31) and how many admission\_dates occurred on that day. Sort by the day with most admissions to least admissions.**

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

|   | day_of_month | admission_count |
|---|--------------|-----------------|
| ▶ | 11           | 184             |
|   | 4            | 184             |
|   | 9            | 183             |
|   | 2            | 180             |
|   | 12           | 179             |
|   | 6            | 179             |
|   | 16           | 177             |
|   | 21           | 174             |
|   | 28           | 173             |
|   | 13           | 173             |
|   | 7            | 167             |
|   | 3            | 166             |
|   | 8            | 166             |

... Table continues

**Explanation:** It filters and displays the **day\_of\_month** and **admission\_count** from **admissions** table by grouping the day and ordering the admission count in descending format.

**31. 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 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.**

- ```
SELECT FLOOR(weight / 10) * 10 AS weight_group, COUNT(*) AS
patient_count
FROM patients
GROUP BY weight_group
ORDER BY weight_group DESC;
```

	weight_group	patient_count
▶	140	6
	130	59
	120	191
	110	426
	100	507
	90	403
	80	478
	70	633
	60	685
	50	443
	40	202
	30	126
	20	165
	10	114
	0	92

Explanation: It filters and displays the **weight_group** and **patient_count** by grouping the weights and orders it in Descending and displays the count of the patients.

32. 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.

- ```
SELECT patient_id, weight, height,
CASE WHEN weight / POWER(height / 100, 2) >= 30 THEN 1 ELSE 0 END
AS isObese
FROM patients;
```

|   | patient_id | weight | height | isObese |
|---|------------|--------|--------|---------|
| ▶ | 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       |
|   | 9          | 95     | 173    | 1       |
|   | 10         | 61     | 157    | 0       |
|   | 11         | 74     | 158    | 0       |
|   | 12         | 46     | 145    | 0       |
|   | 13         | 77     | 146    | 1       |
|   | 14         | 95     | 220    | 0       |
|   | 15         | 72     | 172    | 0       |
|   | 16         | 59     | 153    | 0       |

**Explanation:** It filters and displays the **patient\_id**, **weight**, **height** and **isObese** . The obese is calculated based on BMI, If the BMI is  $\geq 30$  then the patient is obese.

In general **BMI=weight (kg)/height (m)<sup>2</sup>**

**33. Show patient\_id, first\_name, last\_name, and attending doctor's specialty. Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'. Check patients, admissions, and doctors tables for required information.**

- SELECT p.patient\_id, p.first\_name, p.last\_name, d.specialty  
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 a.diagnosis = 'Epilepsy' AND d.first\_name = 'Lisa';

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

**Explanation:** It filters and displays the **patient\_id**, **first\_name**, **last\_name** and attending doctor's **specialty** with conditions whose diagnosis is 'Epilepsy' and doctor's first name is 'Lisa'.

## CONCLUSION:

All the given Queries have been performed and found the solutions based on the provided database '**Medical\_data\_history**' on the Provinces of Canada.

From **Loading the Database, exploring datatypes of all the tables and executed the queries by interpreting the data's in appropriate way to find the solutions for each and every queries.**

In this project we not only found the solution for queries more than that **we have gained knowledge of how to join the multiple tables for a particular solution, how to group and order, how the cases should be used, how to calculate the obesity of the patients using BMI formula, how to aggregate the patients based on their id's, based on names, based on sorting in ascending and descending order, grouping patients based on weight category, grouped doctors with respect to their attributes, speciality , aggregated the provinces etc etc.**

So that we have performed all queries and got solutions and explained this project '**Medical\_data\_history**' process step by step.