PRSQL – 02 Medical History Data
SQL Documentation
Team ID: PTID-CDA-NOV-25-826

Purpose:

Provided runnable SQL queries used for analysis and exercises on project_medical_data_history database. This document explains each query, expected output, assumptions, and recommendations for indexing, testing and hardening.

Environment:

Database: project_medical_data_history (explicitly selected in the SQL file with USE project_medical_data_history;)

SQL dialect: MySQL-compatible(queries se functions such as YEAR(), DAY(), CONCAT() LENGTH(), SUBSTRING_INDEX() AND FLOOR()).

Permissions: Some queries assume SELECT privileges; one UPDATE is present and may require write privileges.

Inferred Scheme (Columns Referenced):

The SQL file references the following tables and columns (inferred from queries):

- **patients**
  - Patient_id, first_name, last_name, gender, birth_datem weight, height, allergies, province_id, city.
- **admissions**
  - admission_id (inferred), patient_id, admission_date, diagnosis, attending_doctor_id
- **doctors**
  - doctor_id, first_name, last_name, role(inferred)
- **province_names**
  - province_id, province_name

High-level summary of file contents:

The SQL file contains a sequence of commented queries (numbered exercises) that run SELECT statements across the four tables above. It includes simple retrievals, aggregations, string and date manipulations, grouping, joins, and one UPDATE statement for filling null allergies values.

Query reference – purpose & output

Below are the extracted queries from Medical_history.sql with a short purpose the SQL, the expected output columns, and brief notes/recommendations.

Query 0 – Verify tables / sample rows

Purpose: Inspect table availability, structure and sample data before running reports.

Query: select * from admissions;

select * from doctors;

select * from patients;

select * from province_names;

Output:

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
6	2018-11-08	2018-11-09	Uterine Fibroid	22
7	2018-06-24	2018-07-03	Cancer	8
8	2018-09-18	2018-09-21	Amigima	6
9	2018-12-31	2018-12-31	Ruptured Appendicitis	19
9	2019-03-02	2019-03-09	Osteoarthritis	8
10	2018-12-30	2019-01-05	Zenkers Diverticulitis	14

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
6	Tasha	Phillips	Psychiatrist
7	Hazel	Patterson	Oncologist
8	Mickey	Duval	Pediatrician
9	Jon	Nelson	Neurologist
10	Monica	Singleton	Orthopaedic Surgeon
11	Douglas	Brooks	Respirologist

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
6	Sue	Falcon	F	2017-09-30	Ajax	ON	Penicillin	43	5
7	Thomas	ONeill	M	1993-01-31	Burlington	ON	NULL	180	117
8	Sonny	Beckett	M	1952-12-11	Port Hawkesbury	NS	NULL	174	105
9	Sister	Spitzer	F	1966-10-15	Toronto	ON	Penicillin	173	95
10	Cedric	Coltrane	M	1961-11-10	Toronto	ON	NULL	157	61
11	Hank	Spencer	M	1969-08-10	Peterborough	ON	NULL	158	74

province_id	province_name
AB	Alberta
BC	British Columbia
MB	Manitoba
NB	New Brunswick
NL	Newfoundland and Labrador
NS	Nova Scotia
NT	Northwest Territories
NU	Nunavut
ON	Ontario
PE	Prince Edward Island
QC	Quebec

Query 1 – List male patients

Purpose: Retrieve first and last names for male patients.

Query: SELECT

first_name,

last_name,

```sql
  CONCAT(first_name, ' ', last_name) AS Full_name,

  gender

FROM

  patients

WHERE

  gender = 'M';
```

	first_name	last_name	Full_name	gender
▶	Donald	Waterfield	Donald Waterfield	M
	Mickey	Baasha	Mickey Baasha	M
	Jiji	Sharma	Jiji Sharma	M
	Blair	Diaz	Blair Diaz	M
	Charles	Wolfe	Charles Wolfe	M
	Thomas	ONeill	Thomas ONeill	M
	Sonny	Beckett	Sonny Beckett	M
	Cedric	Coltrane	Cedric Coltrane	M
	Hank	Spencer	Hank Spencer	M

Output:

Query 2 – Show first & last names of patients with NULL allergies

Purpose: Identify patients missing allergy information

Query: SELECT

```sql
  first_name, last_name, allergies

FROM

  patients

WHERE

  allergies IS NULL;
```

	first_name	last_name	allergies
▶	Donald	Waterfield	NULL
	Blair	Diaz	NULL
	Thomas	ONeill	NULL
	Sonny	Beckett	NULL
	Cedric	Coltrane	NULL
	Hank	Spencer	NULL
	Sara	di Marco	NULL
	Amy	Leela	NULL
	Rachel	Winterbourne	NULL

Output:

Query 3 - Show first names that start with 'C'

Purpose: Demonstrate string filtering using prefix matching
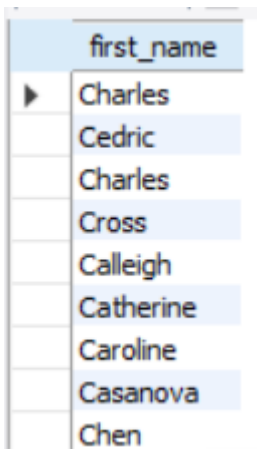
Query: SELECT

  first_name

FROM

  patients

WHERE

  first_name LIKE 'C%';

Output:

first_name
Charles
Cedric
Charles
Cross
Calleigh
Catherine
Caroline
Casanova
Chen

Query 4 – Show patients weighing between 100 and 120 kg

Purpose: Retrieve medium-heigh weight range patients for analysis

Query: SELECT

  first_name, last_name, weight

FROM

  patients

WHERE

weight BETWEEN 100 AND 120;

	first_name	last_name	weight
▶	Jiji	Sharma	106
	Blair	Diaz	104
	Thomas	ONeill	117
	Sonny	Beckett	105
	Tom	Halliwell	114
	Jon	Doggett	116
	Angel	Edwards	106
	John	Farley	104
Output: | Temple | Russert | 102 |

Query 5 – Update NULL allergies to 'NKA'

Purpose: Data Cleaning – replace missing allergy values with 'No Known Allergies' *(Note: Fails as user lacks UPDATE permission)*

Query: UPDATE patients

SET

   allergies = 'NKA'

WHERE

   allergies IS NULL;

Output: *Permission Denied for UPDATE*

Query 6 – Show full name using CONCAT()

Purpose: Create user-friendly full-name column (first_name + last_name).

Query: SELECT

   first_name,

   last_name,

   CONCAT(first_name, ' ', last_name) AS Full_name

FROM

   patients;

first_name	last_name	Full_name
Donald	Waterfield	Donald Waterfield
Mickey	Baasha	Mickey Baasha
Jiji	Sharma	Jiji Sharma
Blair	Diaz	Blair Diaz
Charles	Wolfe	Charles Wolfe
Sue	Falcon	Sue Falcon
Thomas	ONeill	Thomas ONeill
Sonny	Beckett	Sonny Beckett
Sister	Spitzer	Sister Spitzer

Output:

Query 7 – Join patients with province_names

Purpose: Display patients with their province names using a foreign-key lookup.

Query: SELECT

  p.first_name, p.last_name, pn.province_name

FROM

  patients p

    JOIN

  province_names pn ON p.province_id = pn.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

Output: Result 10

Query 8 – Count patients born in 2010

Purpose: Determine number of patients from a specific birth year.

Query: SELECT

  COUNT(patient_id) AS Count_of_patients_born_in_2010

FROM

  patients

WHERE

  YEAR(birth_date) = 2010;

Output:

Count_of_patients_born_in_2010
55

Query 9 – Show tallest patient

Purpose: Retrieve the maximum height entry with patient identification

Query: SELECT

  first_name, last_name, height

FROM

  patients

ORDER BY height DESC

LIMIT 1;

Output:

first_name	last_name	height
Sam	Haruko	226

Query 10 – Show all columns for patient IDs(1, 45, 534, 879, 1000)

Purpose: Direct lookup for specific patient IDs for testing or validation.

Query: SELECT

  *

FROM

  patients

WHERE

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

Output:

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

Query 11 – Count total number of admissions

Purpose: Basic dataset size metric for admissions.

Query: SELECT

    COUNT(*) AS Total_admissions

FROM

    admissions;

Total_admissions
▶ 5067

Output:

Query 12 – Show all columns for patients admitted and discharged on same day.

Purpose: Retrieve all admissions where patients were admitted and discharged on the same day, indicating same-day hospital visits.

Query: SELECT

    *

FROM

    admissions

WHERE

    admission_date = discharge_date;

Output:

	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

admissions 17 ⌄

Query 13 – Show total number of admission for patient_id = 579

Purpose: Count how many times patient_id 579 is admitted.

Query: SELECT

   COUNT(patient_id)

FROM

   admissions

WHERE

   patient_id = 579;

	COUNT(patient_id)
▶	2

Output:

Query 14 – Show unique cities that are in province_name 'NS'

Purpose: Retrieve all unique cities in which patients reside within the province_id 'NS'

Query: SELECT DISTINCT

   (city) AS unique_cities_in_NS, province_id

FROM

   patients

WHERE

   province_id = 'NS';

unique_cities_in_NS	province_id
Port Hawkesbury	NS
Halifax	NS
Inverness	NS

Output:

Query 15 – Patients above 160 cm Height and 70kg weight

Purpose: Retrieve patients whose heights exceeds 160cm and weight 70kg, along with their basic personal details

Query: SELECT

  first_name, last_name, birth_date, height, weight

FROM

  patients

WHERE

  height > 160 AND weight > 70;

first_name	last_name	birth_date	height	weight
Mickey	Baasha	1981-05-28	185	76
Jiji	Sharma	1957-09-05	194	106
Blair	Diaz	1967-01-07	191	104
Thomas	ONeill	1993-01-31	180	117
Sonny	Beckett	1952-12-11	174	105
Sister	Spitzer	1966-10-15	173	95
Rick	Bennett	1977-01-27	220	95
Amy	Leela	1977-06-25	172	72
Tom	Halliwell	1987-08-01	179	114

Output:

Query 16 – Unique Patient Birth years

Purpose: Retrieve all distinct birth year from the patients table and display them in ascending order.

Query: SELECT DISTINCT

  (YEAR(birth_date)) AS Unique_Birth_Years

FROM

  patients

ORDER BY YEAR(birth_date) ASC;

Unique_Birth_Years
1918
1923
1925
1926
1927
1928
1929
1931
1933

Output: Result 21 ×

Query 17 – First Names occurring only once

Purpose: Retrieve first names that appear exactly once in the patients table.

Query: SELECT

   first_name AS Unique_first_names

FROM

   patients

GROUP BY first_name

HAVING COUNT(*) = 1

ORDER BY first_name;

Unique_first_names
Abby
Adelaide
Adelia
Akira
Albert
Aldo
Alec
Alicia
Allan

Output:

Query 18 – First Names starting and ending with 's'

Purpose: Retrieve patient IDs and first names where the name starts with 's', ends with 's' and has a minimum length of six characters.

Query: SELECT

   patient_id, first_name

FROM

   patients

WHERE

   first_name LIKE 's%s'

     AND LENGTH(first_name) >= 6

ORDER BY patient_id;

Output:

patient_id	first_name
496	Spiros
629	Spiros
648	Stanislaus
1273	Stanislaus
1789	Seamus
1926	Stanislaus
1996	Stanislaus
2258	Spiros
2378	Stanislaus

Query 19 – Patients Diagnosed with Dementia

Purpose: Retrieve patient details for individuals whose primary diagnosis in the admissions table is recorded as 'Dimentia'

Query: SELECT

   p.patient_id, p.first_name, p.last_name, ad.diagnosis

FROM

   patients p

     JOIN

   admissions ad ON p.patient_id = ad.patient_id

WHERE

ad.diagnosis = 'Dementia';

patient_id	first_name	last_name	diagnosis
160	Miranda	Delacour	Dementia
178	David	Bustamonte	Dementia
207	Matt	Celine	Dementia
613	Jaki	Granger	Dementia
836	Montana	Vimes	Dementia
924	Simon	Spellman	Dementia
1201	Irene	Murphy	Dementia
1264	Jillian	Valentine	Dementia
1402	Kathryn	Hallow	Dementia

Output:

Query 20 – First Names Ordered by Length and Alphabetically

Purpose: Display all patient first names and sorted first by name length and then alphabetically

Query: SELECT

  first_name

FROM

  patients

ORDER BY LENGTH(first_name) , first_name ASC;

first_name
Al
Al
Al
Al
Al
Al
Al
Bo

Output:

Query 21 – Male and Female patient count in a single row

Purpose: Retrieve the total number of male and female patients from the patients table and display both counts in a single result using conditional aggregation.

Query: SELECT

   COUNT(CASE

      WHEN gender = 'M' THEN 1

   END) AS Count_of_Male_patients,

   COUNT(CASE

      WHEN gender = 'F' THEN 1

   END) AS Count_of_Female_patients

FROM

   patients;

Count_of_Male_patients	Count_of_Female_patients
2468	2062

Output:

Query 22 – Total Male and Female Patient Count

Purpose: Retrieve the total number of male patients and the total number of female patients from the patients table using conditional counting.

Query: SELECT

   COUNT(CASE

      WHEN gender = 'M' THEN 1

   END) AS Count_of_Male_Patients,

   COUNT(CASE

      WHEN gender = 'F' THEN 1

   END) AS Count_of_Female_Patients

FROM

   patients;

Count_of_Male_Patients	Count_of_Female_Patients
2468	2062

Output:

Query 23 – Patients admitted multiple times for the same diagnosis

Purpose: Identify patients who have been admitted more than once with the same diagnosis by grouping admissions records and returning only those with multiple occurrences.

Query: 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

Output:

Query 24 – Patient count by city

Purpose: Display each city along with the total number of patients living there, ordered first by highest patient count and then alphabetically by city name.

Query: SELECT

   city, COUNT(*) AS Total_patients

FROM

   patients

GROUP BY city

ORDER BY Total_patients DESC , city ASC;

city	Total_patients
Hamilton	1938
Toronto	317
Burlington	276
Brantford	147
Ancaster	117
Stoney Creek	107
Cambridge	79
Dundas	79
Milton	65

Output:

Query 25 – Combined List of all patients and doctors with roles

Purpose: Retrieve the first name, last name and assigned role ("Doctor" or "Patient") for every individual in both the doctors and patients tables by merging the two datasets.

Query: SELECT

   first_name, last_name, 'Doctor' AS Role

FROM

   doctors

UNION ALL SELECT

   first_name, last_name, 'Patient' AS Role

FROM

   patients;

first_name	last_name	Role
Jeanette	Sites	Doctor
Larry	Miller	Doctor
Donna	Greenwood	Doctor
Donald	Waterfield	Patient
Mickey	Baasha	Patient
Jiji	Sharma	Patient
Blair	Diaz	Patient
Charles	Wolfe	Patient
Sue	Falcon	Patient

Output:

Query 26 – Allergies Order by Popularity

Purpose: Retrieve all non-null allergies types and rank them by how frequently they occur among patients, from most to least common.

Query: SELECT

  allergies, COUNT(allergies) AS Allergies_by_popularity

FROM

  patients

WHERE

  allergies IS NOT NULL

GROUP BY allergies

ORDER BY Allergies_by_popularity DESC;

Output:

allergies	Allergies_by_popularity
Penicillin	1082
Codeine	305
Sulfa	157
ASA	99
Sulfa Drugs	71
Peanuts	52
Iodine	48
Tylenol	42
Bee Stings	40

Query 27 – Patients Born in the 1970s

Purpose: Retrieve all patients born between 1970 and 1979 and sort them from the earliest to latest birth date.

Query: SELECT

  first_name, last_name, birth_date

FROM

  patients

WHERE

  YEAR(birth_date) BETWEEN 1970 AND 1979

ORDER BY birth_date;

Output:

	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

Query 28 – Formatted Full Name (LastName, FirstName) Display

Purpose: Generate each patient's full name in a single formatted column where the last name appears in uppercase, the first name in lowercase, separated by a comma, and sort the output by first name in descending order.

Query: SELECT

CONCAT(UPPER(last_name), ',', LOWER(first_name)) AS Full_name

FROM

patients

ORDER BY LOWER(first_name) DESC;

	Full_name
▶	MILLER,zoe
	CORBIE,ziva
	KOBAYAKAWA,zenigata
	OVERSTREET,zenigata
	BENNETT,zen
	MEPHESTO,zelda
	MORRIS,zelda
	THOMAS,zefram
	FLUTE,zefram

Output:

Query 29 – Provinces with total patient height>=7000

Purpose: Retrieve each province_id where the combine height of all patients in that province is at least 7000 units.

Query: SELECT

   province_id, SUM(height) AS Sum_of_height

FROM

   patients

GROUP BY province_id

HAVING SUM(height) >= 7000;

province_id	Sum_of_height
▶ BC	7720
NS	9765
ON	678037

Output:

Query 30 – Weight range for patients with last name 'Maroni'

Purpose: Calculate the difference between the maximum and minimum weight among the patients whose last name is 'Maroni'.

Query: SELECT

   MAX(weight) - MIN(weight) AS Weight_difference

FROM

   patients

WHERE

   last_name = 'Maroni';

Weight_difference
▶ 71

Output:

Query 31 – Admission Count by Day of the Month

Purpose: Display each calendar day (1-31) along with the number of admissions occurring on that day, sorted from the highest to lowest admission count.

Query: SELECT

   DAY(admission_date) AS Day_of_month,

   COUNT(*) AS Number_of_admissions

FROM

   admissions

GROUP BY DAY(admission_date)

ORDER BY Number_of_admissions DESC;

Output:

Day_of_month	Number_of_admissions
11	184
4	184
9	183
2	180
12	179
6	179
16	177
21	174
28	173

Query 32 – Patient Count by 10-Kilogram Weight Groups

Purpose: Group patients into 10-kg weight ranges (eg. 100-109,110-119) and display how many patients fall into each group sorted from the heaviest weight group to the lightest.

Query: SELECT

   weight_range, COUNT(*) AS total_patients

FROM

   (SELECT

     CONCAT(FLOOR(weight / 10) * 10, ' to', FLOOR(weight / 10) * 10 + 9) AS weight_range

   FROM

     patients

WHERE

weight IS NOT NULL) AS t

GROUP BY weight_range

ORDER BY CAST(SUBSTRING_INDEX(weight_range, ' ', 1) AS UNSIGNED) DESC;

	weight_range	total_patients
▶	140 to 149	6
	130 to 139	59
	120 to 129	191
	110 to 119	426
	100 to 109	507
	90 to 99	403
	80 to 89	478
	70 to 79	633
	60 to 69	685

Output:

Query 33 – BMI Calculation and Obesity Flag

Purpose: Display each patient's ID, weight, height, and a Boolean value indicating obesity (1 for obese, 0 for not obese) based on BMI >= 30, where BMI is calculated using weight in kilograms and height in centimetres

Query: SELECT

patient_id,

weight,

height,

CASE

WHEN weight / ((height / 100.0) * (height / 100.0)) >= 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

Output:

Query 34 – Epilepsy Patients Treated by Doctor Lisa

Purpose: Retrieve patients diagnosed with Epilepsy along with their attending doctor's speciality, but only when the doctor's first name is Lisa, using data joined from the patients, admissions, and doctors tables.

Query: SELECT

p.patient_id, p.first_name, p.last_name, d.specialty

FROM

patients as p

JOIN

admissions as a ON p.patient_id = a.patient_id

JOIN

doctors as 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

Output: