Shashidhar M
Data Science & Analytics Intern @ Psyliq

Task: SQL Diabetes Prediction

1.

These are the patient id with their age

```
15  -- 1. Retrieve the Patient_id and ages of all patients
16  SELECT patient_id,age
17  FROM diabetes_predictions;
18
19  -- 2. Select all female patients who are older than 40
20  SELECT *
21  FROM diabetes_predictions
22  WHERE age>40 and gender='Female'
23
24  -- 3. Calculate the average BMI of patients.
25  SELECT avg(bmi) AS bmi_avg
26  FROM diabetes_predictions
27
```

Data Output | Messages | Notifications

	patient_id character varying	age numeric (5,2)
1	PT100101	22.00
2	PT101	80.00
3	PT102	54.00
4	PT103	28.00
5	PT104	36.00
6	PT105	76.00
7	PT106	20.00
8	PT107	44.00
9	PT108	79.00
10	PT109	42.00

iabetes_Healthcare > Schemas > public > Tables > diabetes_predictions > Columns > bmi

2.

- all female patients who are older than 40

```sql
-- 2. Select all female patients who are older than 40
SELECT *
FROM diabetes_predictions
WHERE age>40 and gender='Female'

-- 3. Calculate the average BMI of patients.
SELECT avg(bmi) AS bmi_avg
FROM diabetes_predictions
```

Output | Messages | Notifications

employeename character varying	patient_id character varying	gender character varying	age numeric (5,2)	hypertension integer	heart_disease integer	smoking_history character varying	bmi numeric (5,2)	hba1c_level numeric (4,2)	blood_glucose_level integer	diabetes integer
NATHANIEL FORD	PT101	Female	80.00	0	1	never	25.19	6.60	140	
GARY JIMENEZ	PT102	Female	54.00	0	0	No Info	27.32	6.60	80	
ALSON LEE	PT107	Female	44.00	0	0	never	19.31	6.50	200	
DAVID KUSHNER	PT108	Female	79.00	0	0	No Info	23.86	5.70	85	
ARTHUR KENNEY	PT111	Female	53.00	0	0	never	27.32	6.10	85	
PATRICIA JACKSON	PT112	Female	54.00	0	0	former	54.70	6.00	100	
EDWARD HARRINGTON	PT113	Female	78.00	0	0	former	36.05	5.00	130	
JOHN MARTIN	PT114	Female	67.00	0	0	never	25.69	5.80	200	
DAVID FRANKLIN	PT115	Female	76.00	0	0	No Info	27.32	5.00	160	
SEBASTIAN WONG	PT118	Female	42.00	0	0	never	24.48	5.70	158	

Healthcare > Schemas > public > Tables > diabetes_predictions > Columns > bmi

Ln 19, Col

3. The average BMI of patients.

- the average BMI of patients.

```
23
24  -- 3. Calculate the average BMI of patients.
25  SELECT avg(bmi) AS bmi_avg
26  FROM diabetes_predictions
27
28  -- 4. List patients in descending order of blood glucos
29  SELECT *
30  FROM diabetes_predictions
31  ORDER BY blood_glucose_level DESC
32
```

Data Output | Messages | Notifications

	bmi_avg numeric
1	27.3207294927050729

4. Patients in descending order of blood glucose levels

```
27
28  -- 4. List patients in descending order of blood glucose levels.
29  SELECT *
30  FROM diabetes_predictions
31  ORDER BY blood_glucose_level DESC
32
33  -- 5. Find patients who have hypertension and diabetes.
34  SELECT *
35  FROM diabetes_predictions
36  WHERE hypertension=1 AND diabetes=1;
37
38  -- 6. Determine the number of patients with heart disease.
```

Data Output | Messages | Notifications

	employeename character varying	patient_id character varying	gender character varying	age numeric (5,2)	hypertension integer	heart_disease integer	smoking_history character varying	bmi numeric (5,2)	hba1c_level numeric (4,2)	blood_glucose_level integer	diabetes integer
1	TERRENCE HONG	PT8557	Female	80.00	0	0	former	32.05	9.00	300	1
2	ABDELLATIF HABEK	PT20781	Male	51.00	1	0	not current	32.88	6.20	300	1
3	JOANNA CHAN	PT24450	Female	60.00	1	0	No Info	27.32	5.80	300	1
4	BOAZ MARILES	PT1037	Male	49.00	0	0	never	27.32	6.50	300	1
5	STEFANO MORONI	PT26989	Female	80.00	1	0	never	27.53	6.10	300	1
6	THOMAS HULL	PT3865	Female	39.00	0	0	former	35.50	9.00	300	1
7	MARLEN SANCHEZ	PT16787	Female	50.00	1	0	never	61.67	6.10	300	1
8	ANDREI AFANASIEV	PT17504	Male	60.00	0	0	current	37.46	9.00	300	1
9	MARGARET GROENINGER	PT19665	Male	54.00	1	0	current	27.32	5.80	300	1
10	HOWARD CONROY	PT8894	Male	42.00	0	0	never	49.20	9.00	300	1

5. Patients who have hypertension and diabetes.

- patients who have hypertension and diabetes.

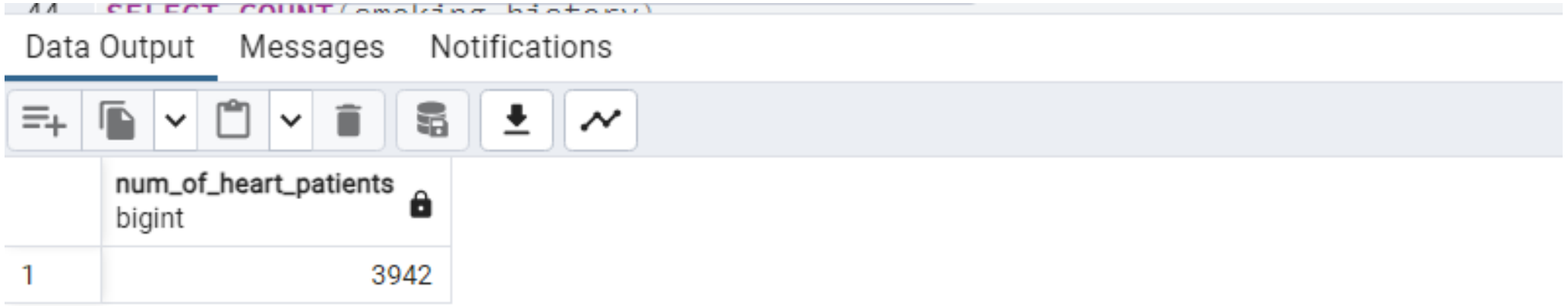
```sql
-- 5. Find patients who have hypertension and diabetes.
SELECT *
FROM diabetes_predictions
WHERE hypertension=1 AND diabetes=1;

-- 6. Determine the number of patients with heart disease.
SELECT COUNT(patient_id) AS num_of_heart_patients
FROM diabetes_predictions
WHERE heart_disease=1;

-- 7. Group patients by smoking history and count how many smokers and non-smokers there are
```

Data Output | Messages | Notifications

	employeename character varying	patient_id character varying	gender character varying	age numeric (5,2)	hypertension integer	heart_disease integer	smoking_history character varying	bmi numeric (5,2)	hba1c_level numeric (4,2)	blood_glucose_level integer	diabetes integer
1	NES WONG	PT139	Male	50.00	1	0	current	27.32	5.70	260	1
2	TRIC STEELE	PT205	Female	80.00	1	0	never	27.32	6.80	280	1
3	THUR STELLINI	PT343	Male	57.00	1	1	not current	27.77	6.60	160	1
4	AD LAW	PT355	Male	63.00	1	0	ever	35.06	5.80	200	1
5	THERINE JAMES	PT451	Female	52.00	1	0	never	50.30	6.60	155	1
6	HN HART	PT565	Male	48.00	1	0	current	36.12	6.80	140	1
7	HN BARKER	PT567	Female	79.00	1	0	former	27.32	6.50	159	1
8	BERT BONNET	PT632	Female	49.00	1	0	not current	36.93	8.80	155	1
9	TANI BENJAMIN	PT727	Male	43.00	1	0	not current	40.86	6.60	159	1

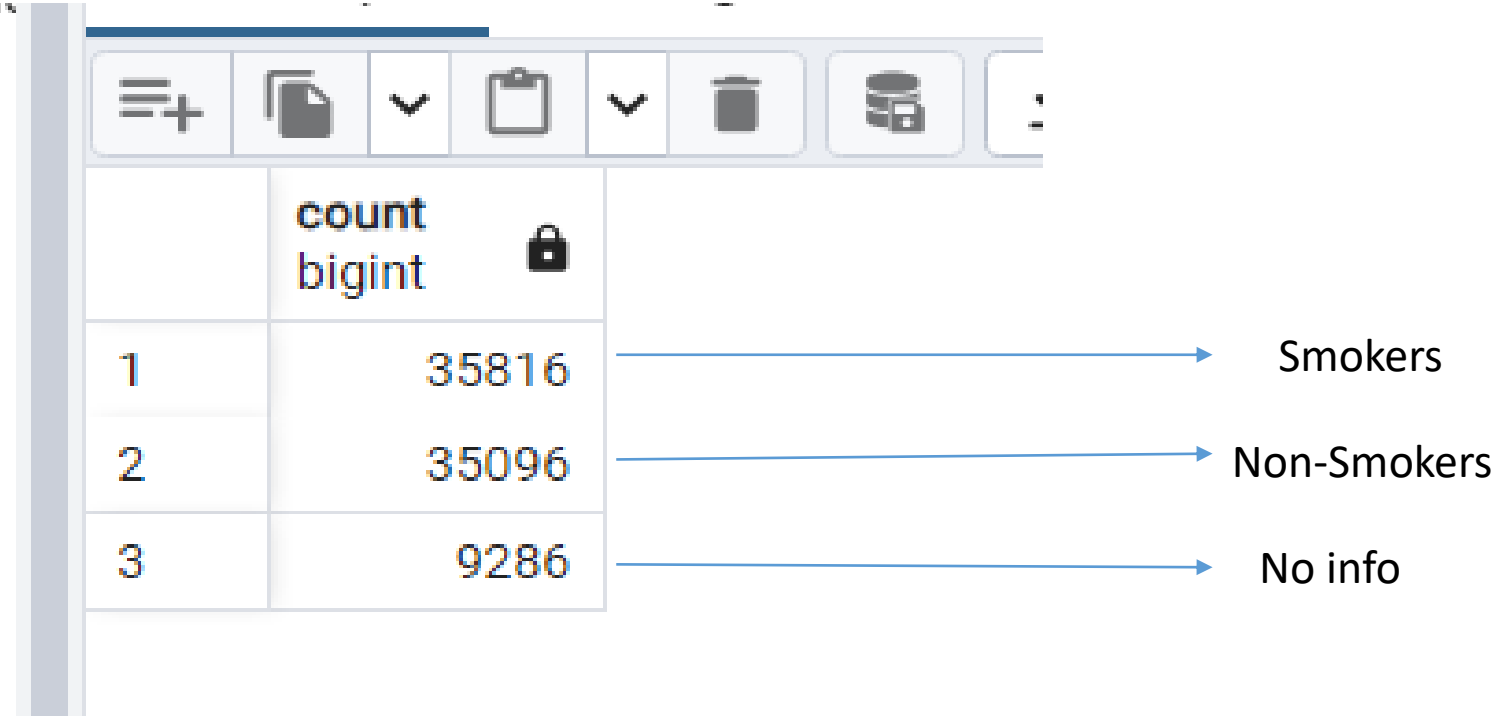
6. The number of patients with heart disease

- the number of patients with heart disease are 3942



7.

- Group patients by smoking history and count how many smokers and non-smokers there are:



	count bigint
1	35816
2	35096
3	9286

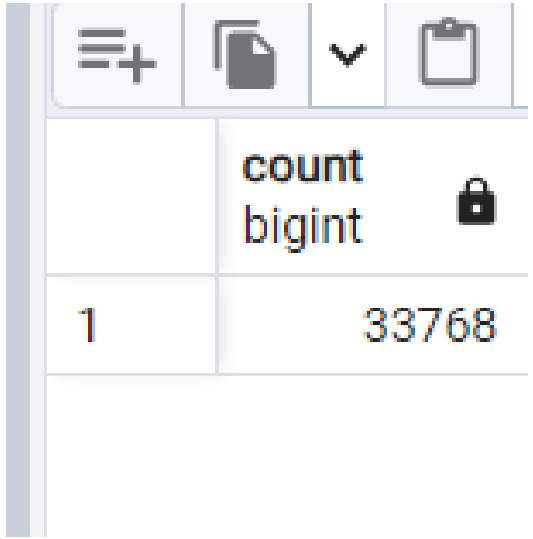
Smokers

Non-Smokers

No info

8.

- BMI greater than the average BMI
- There are upto 33,768 patient Ids whose BMIs are greater than avg BMI.

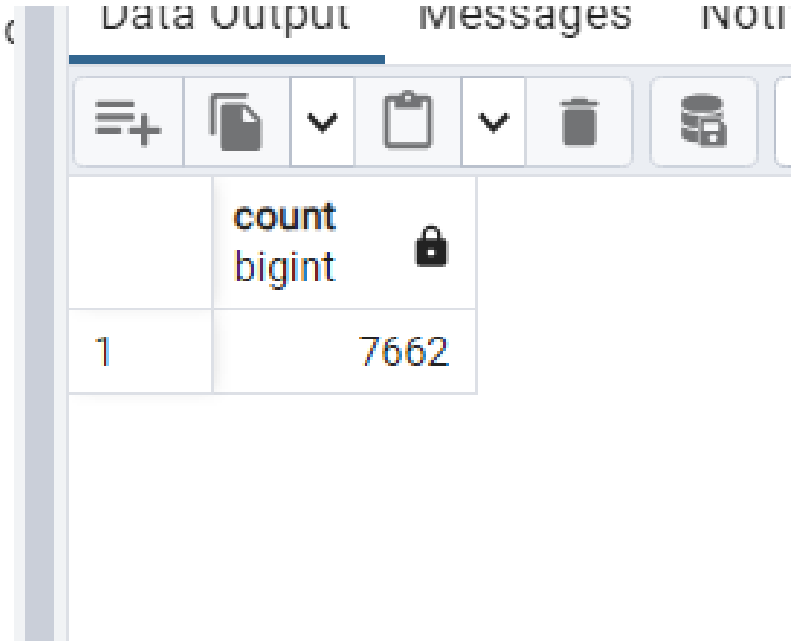


	count bigint
1	33768

9.

- patient with highest hba1c level and lowest hba1c level
- 654 patients with highest hba1c level
- Upto 7662 patients with lowest hba1c level

	patient_id character varying	hba1c_level numeric (4,2)
645	PT98911	9.00
646	PT99155	9.00
647	PT99175	9.00
648	PT99266	9.00
649	PT99298	9.00
650	PT99442	9.00
651	PT99613	9.00
652	PT99764	9.00
653	PT99807	9.00
654	PT99841	9.00



10.

- the age of patients in years (assuming the current date as of now).

Data Output | Messages | Notifications

	patient_id character varying	current_age numeric
1	PT100101	22.00
2	PT101	80.00
3	PT102	54.00
4	PT103	28.00
5	PT104	36.00
6	PT105	76.00
7	PT106	20.00
8	PT107	44.00
9	PT108	79.00
10	PT109	42.00

11

- Rank patients by blood glucose level within each gender group
- The top and bottom rows are as follows

	gender character varying	patient_id character varying	blood_glucose_level integer	rank bigint
1	Female	PT20528	300	1
2	Female	PT25656	300	1
3	Female	PT46292	300	1
4	Female	PT36797	300	1
5	Female	PT47173	300	1
6	Female	PT36431	300	1
7	Female	PT25576	300	1
8	Female	PT39188	300	1
9	Female	PT24031	300	1
10	Female	PT45322	300	1

	gender character varying	patient_id character varying	blood_glucose_level integer	rank bigint
42991	Female	PT76696	100	41929
42992	Female	PT60216	100	41929
42993	Female	PT67342	100	41929
42994	Female	PT49406	100	41929
42995	Female	PT87616	100	41929
42996	Female	PT81101	100	41929
42997	Female	PT73827	100	41929
42998	Female	PT56946	100	41929
42999	Female	PT84207	100	41929
43000	Female	PT61746	100	41929

Total rows: 44000 of 100001 Query complete 00:00:01.122

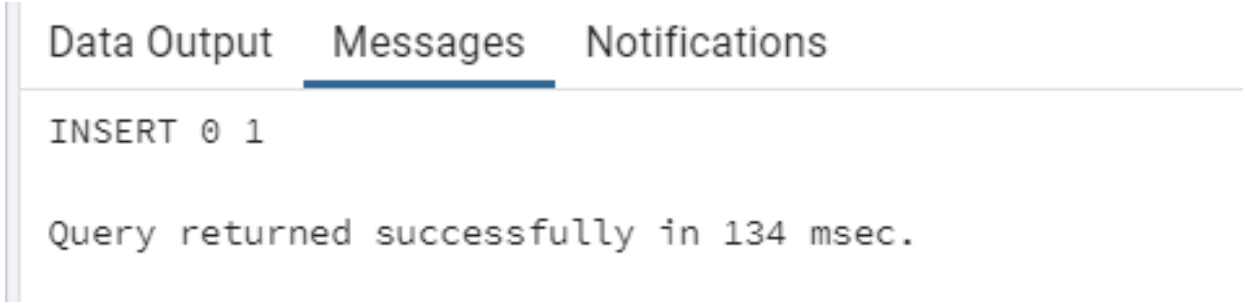
12

- Update the smoking history of patients who are older than 50 to "Ex-smoker."

	smoking_history character varying
1	never
2	Ex-smoker
3	Ex-smoker
4	Ex-smoker
5	Ex-smoker
6	Ex-smoker
7	Ex-smoker
8	Ex-smoker
9	Ex-smoker
10	Ex-smoker

13

- Insert a new patient into the database with sample data.



14

- all patients with heart disease from the database deleted
- We have 96060 patients that do not have heart disease.

	count bigint
1	96060

15

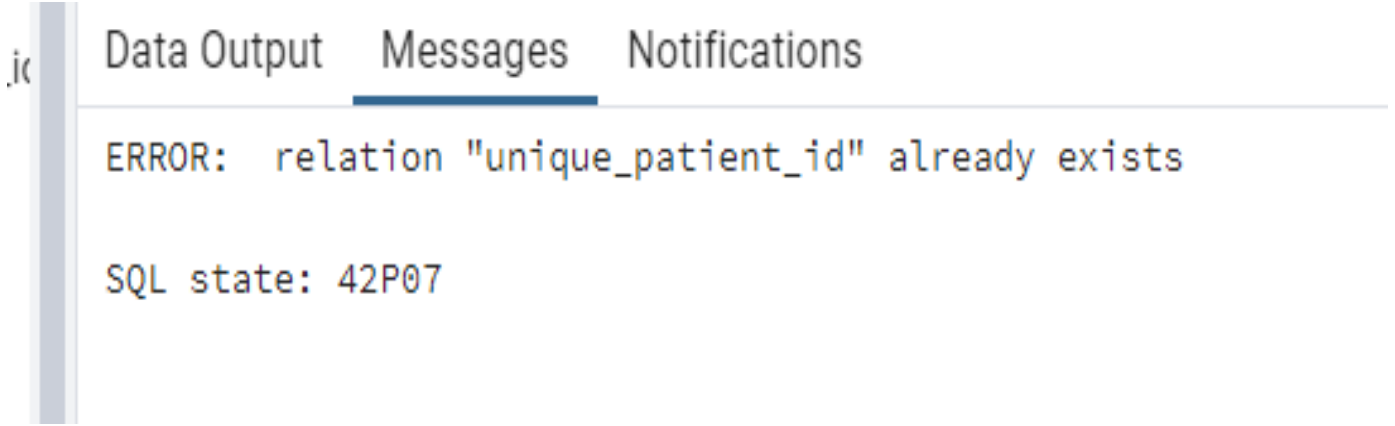
- Patients who have hypertension but not diabetes using the EXCEPT operator

	patient_id character varying
1	PT17788
2	PT55242
3	PT2860
4	PT91488
5	PT56202
6	PT81513
7	PT82537
8	PT2287
9	PT1401
10	PT68942

Total rows: 2000 of 4839 | Query complete 00:00:00.355

16.

- unique constraint on the "patient_id"



17.

- Create a view that displays the Patient_ids, ages, and BMI of patients.

```
ERROR:  relation "patient_view" already exists

SQL state: 42P07
```

18. Improvements in the database schema to reduce data redundancy and improve data integrity.

- Normalization: Normalize your database to eliminate redundant data. This involves dividing your database into two or more related tables and defining relationships between the tables. The main aim of normalization is to add, delete, and modify data without causing data anomalies.

- Use of Primary Keys: Ensure every table has a primary key. This will help maintain the integrity of your database by avoiding duplicate entries.

- Use of Foreign Keys: Use foreign keys whenever relationships exist between tables. This ensures referential integrity in the relationship where a foreign key correctly points to a candidate key.

- Use of Constraints: Use constraints like UNIQUE, NOT NULL, and CHECK to ensure data integrity. These constraints ensure that the data adheres to the defined rules.

- Avoid Null Values: Avoid permitting null values whenever possible. This will make it easier to perform calculations, comparisons, or concatenations with the data.

Cont.d

- Use of Indexes: Use indexes for frequently searched columns to speed up read operations. Be careful, as excessive use of indexes can slow down write operations.
- Consistent Structure: Ensure that all instances of a repeating group (e.g., multiple addresses for a customer) are structured consistently.
- Data Types: Ensure data types are appropriate for the data being stored. This can prevent the possibility of storing inconsistent types of data in the same column.
- Use of Views: Use views to encapsulate the queries that access the structural part of the database. This can help protect the integrity of the underlying data.
- Regular Audits: Regularly audit the data to ensure it adheres to the business rules and constraints.

How you can optimize the performance of SQL queries on this dataset.

- Use Indexes: Indexes can significantly improve the performance of data retrieval queries. However, they can slow down data modification statements (INSERT, UPDATE, DELETE), so use them judiciously.
- Avoid SELECT: Instead of using SELECT *, specify the columns you need. This reduces the amount of data that needs to be sent from the database to the client.
- Use WHERE instead of HAVING for row filtering: WHERE clause is more efficient than HAVING clause. HAVING should only be used for conditions on aggregate functions.
- Use LIMIT: If you only need a certain number of rows, use the LIMIT clause to restrict the data retrieved from the database.
- Use JOINs wisely: Avoid unnecessary JOINs as they can result in large amounts of unnecessary data. Also, use INNER JOIN instead of OUTER JOIN whenever possible.
- Avoid Correlated Subqueries: Correlated subqueries can slow down queries as they can result in repeated executions.

Cont.d

- Use EXPLAIN PLAN: The EXPLAIN PLAN statement can be used to determine the execution plan that the PostgreSQL planner generates for a given SQL statement. This can help identify bottlenecks.
- Database Maintenance: Regular database maintenance like updating statistics, rebuilding indexes, and vacuuming can help improve query performance.
- Normalize Your Data: Normalization can lead to more efficient storage by eliminating redundant data.
- Use Appropriate Data Types: Using the correct data type can save storage and improve performance.