**Patient Data: Diabetes Prediction**

**Introduction**

Hospitals (especially the general hospitals) are always crowded on daily basis. Among the major patients visiting or being admitted in hospitals are those with deadly and serious diseases like hypertension, diabetes etc. In this analysis, patients with diabetes along with its other related problems were queried.

**Analysis Questions:**1. Retrieve the Patient\_id and ages of all patients.  
2. Select all female patients who are older than 40.  
3. Calculate the average BMI of patients.  
4. List patients in descending order of blood glucose levels.  
5. Find patients who have hypertension and diabetes.  
6. Determine the number of patients with heart disease.  
7. Group patients by smoking history and count how many smokers and nonsmokers there are.  
8. Retrieve the Patient\_ids of patients who have a BMI greater than the average BMI.  
9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.  
10. Calculate the age of patients in years (assuming the current date as of now).  
11. Rank patients by blood glucose level within each gender group.  
12. Update the smoking history of patients who are older than 50 to "Ex-smoker."  
13. Insert a new patient into the database with sample data.  
14. Delete all patients with heart disease from the database.  
15. Find patients who have hypertension but not diabetes using the EXCEPT operator.  
16. Define a unique constraint on the "patient\_id" column to ensure its values are unique.  
17. Create a view that displays the Patient\_ids, ages, and BMI of patients.  
18. Suggest improvements in the database schema to reduce data redundancy and improve data integrity.  
19. Explain how you can optimize the performance of SQL queries on this dataset.

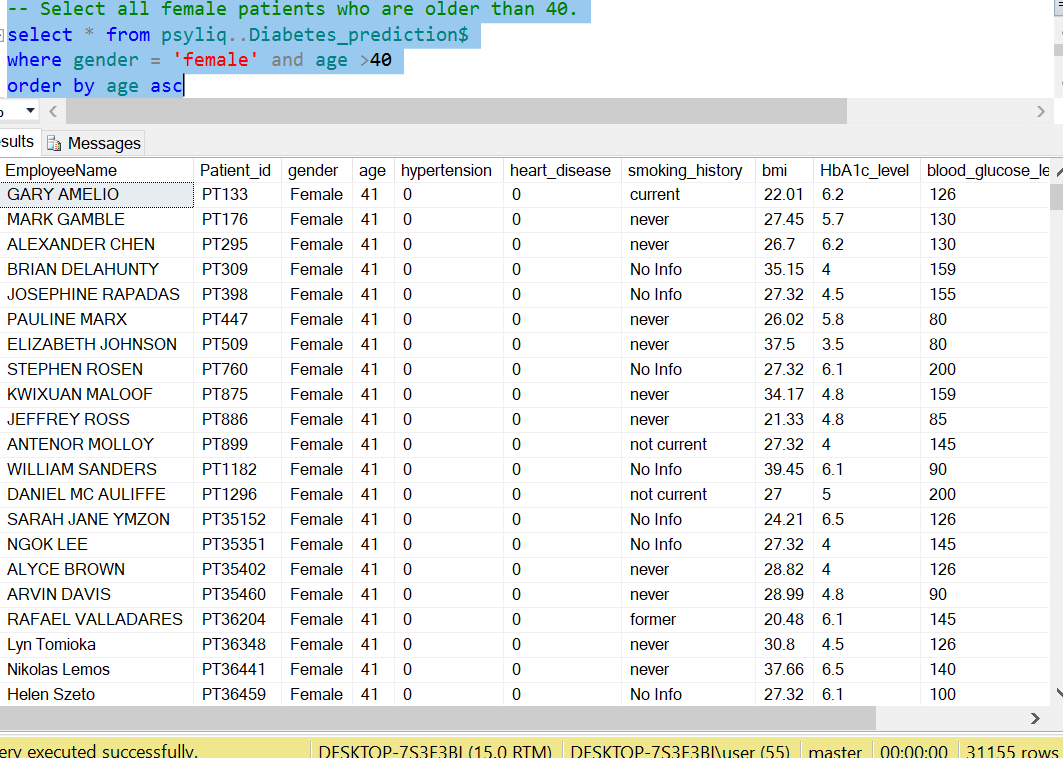
**Project Tool Used:** SQL

**Analysis Insights and Results**

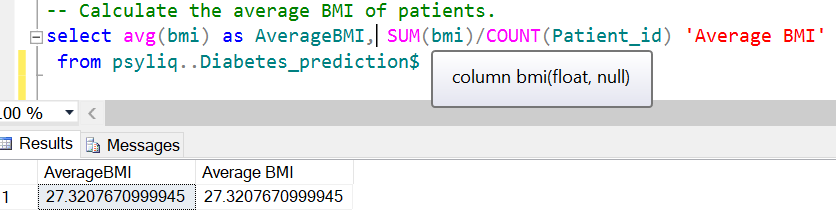
1. The Patient\_id and ages of all patients are shown below. The samples patients are 10,000 in number.



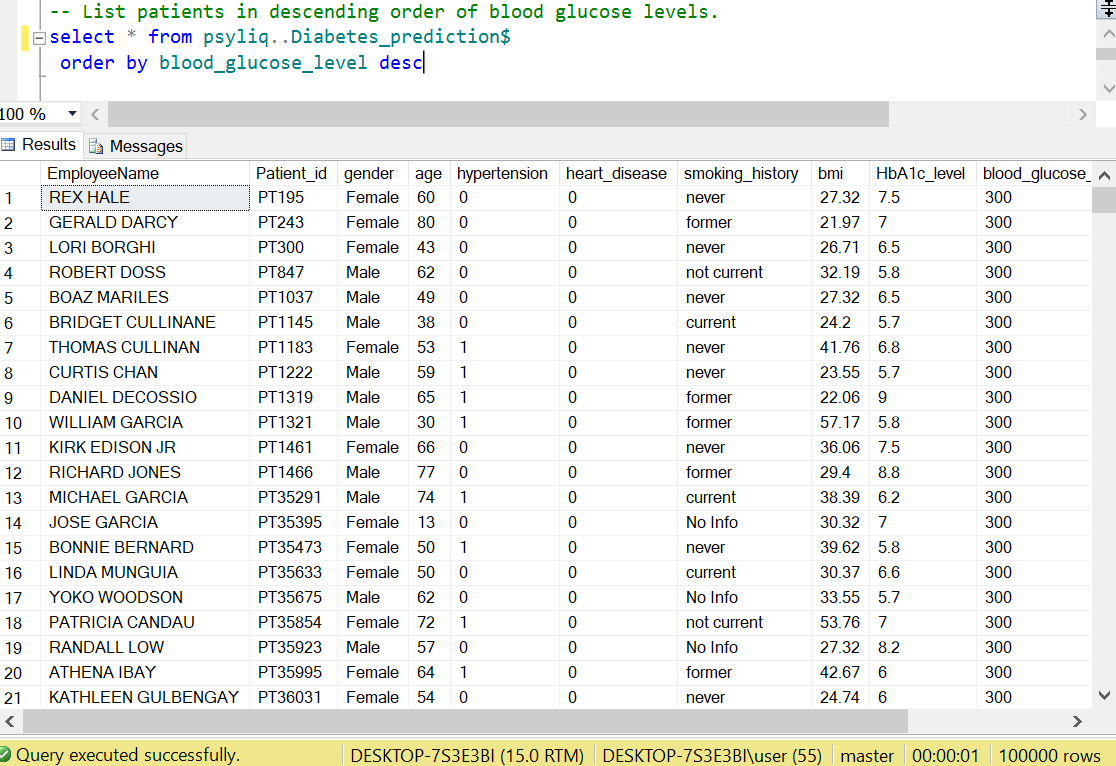
2. Female patients who are older than 40 are 31,155.



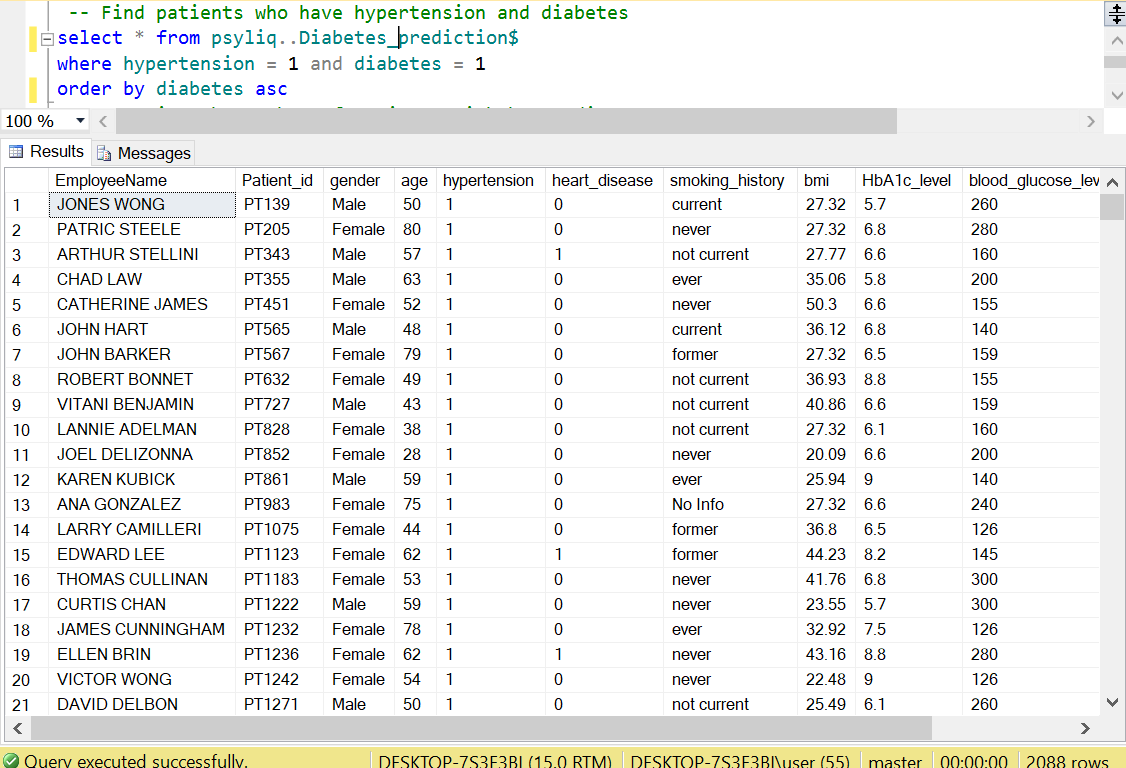
3. The average BMI of patients is calculated to be 27.32



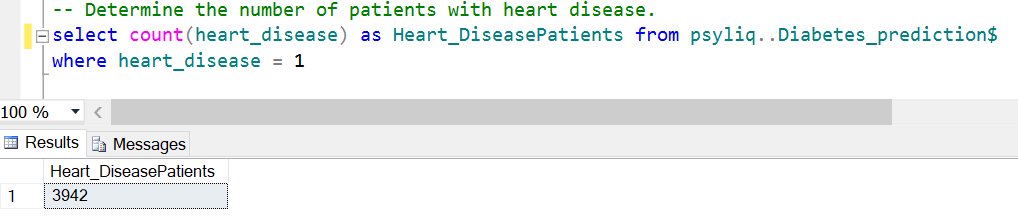
4. Below is the list of patients in descending order of blood glucose levels



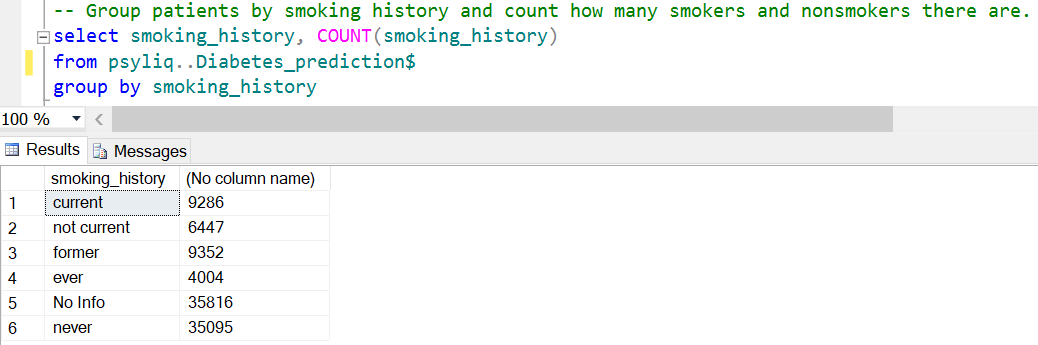
5. Patients with hypertension and diabetes are two thousand and eighty-eight (2,088)



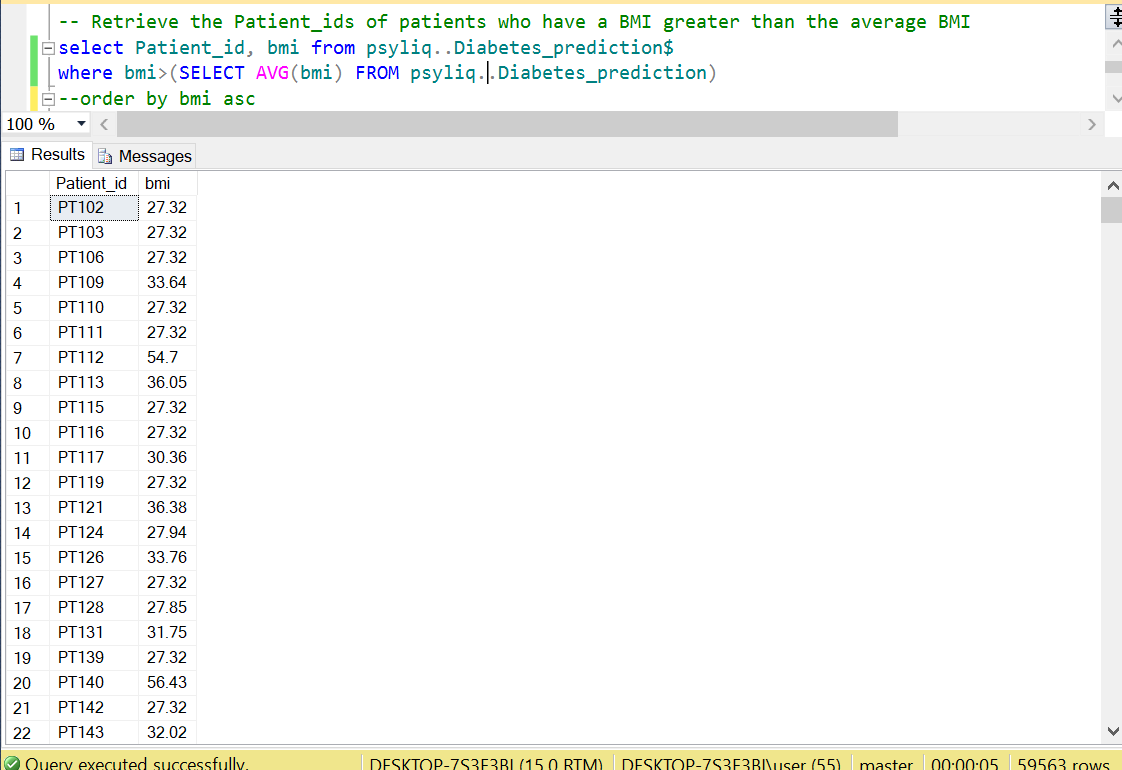
6. The patients with heart disease are three thousand, nine hundred and forty-two (3,942).



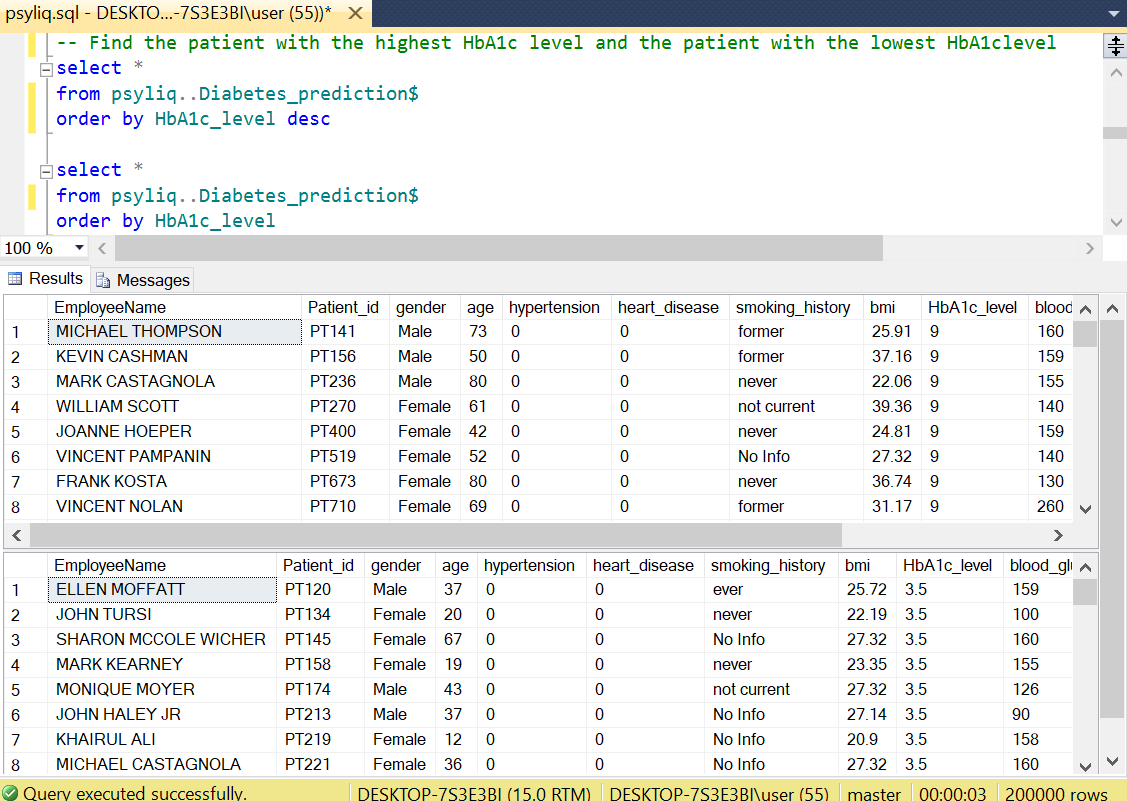
7. By grouping the patients based on smoking history, smokers are found to be 29,089 while non-smokers are found to be 35,095. Therefore, majority of the sampled patients are non-smokers.



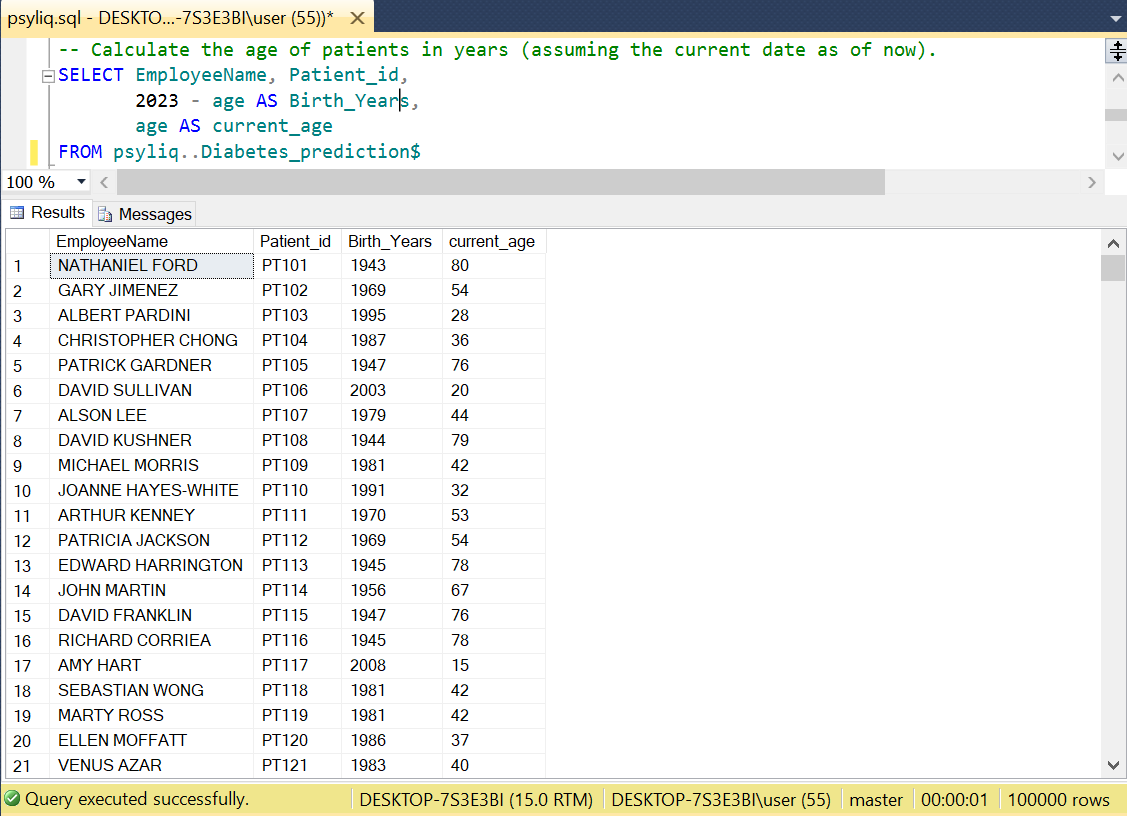
8. The patients (with BMI) above the average BMI are 59,563 and this means they are more than those below the average BMI.



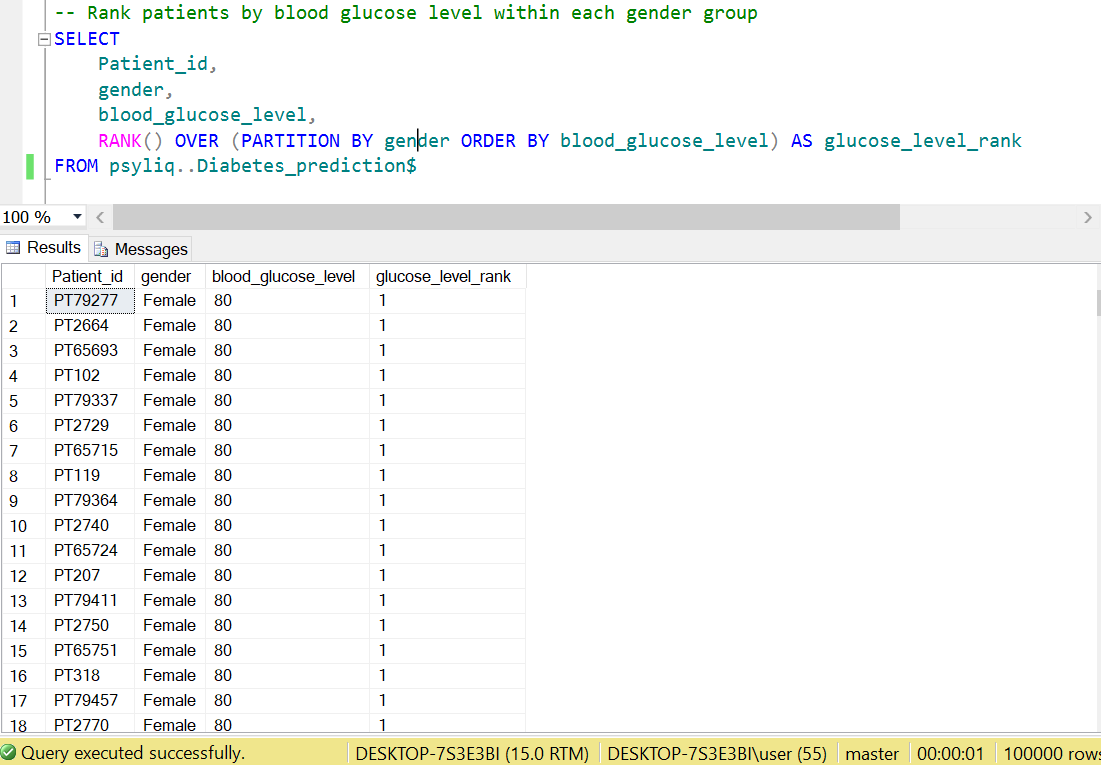
9. The patient with the highest HbA1c level among others is Michael Thompson while the patient with the lowest HbA1c level is Ellen Moffatt. Those with lowest HbA1c level (3.5) are 7,451 while those with highest HbA1c level (9) are 556.



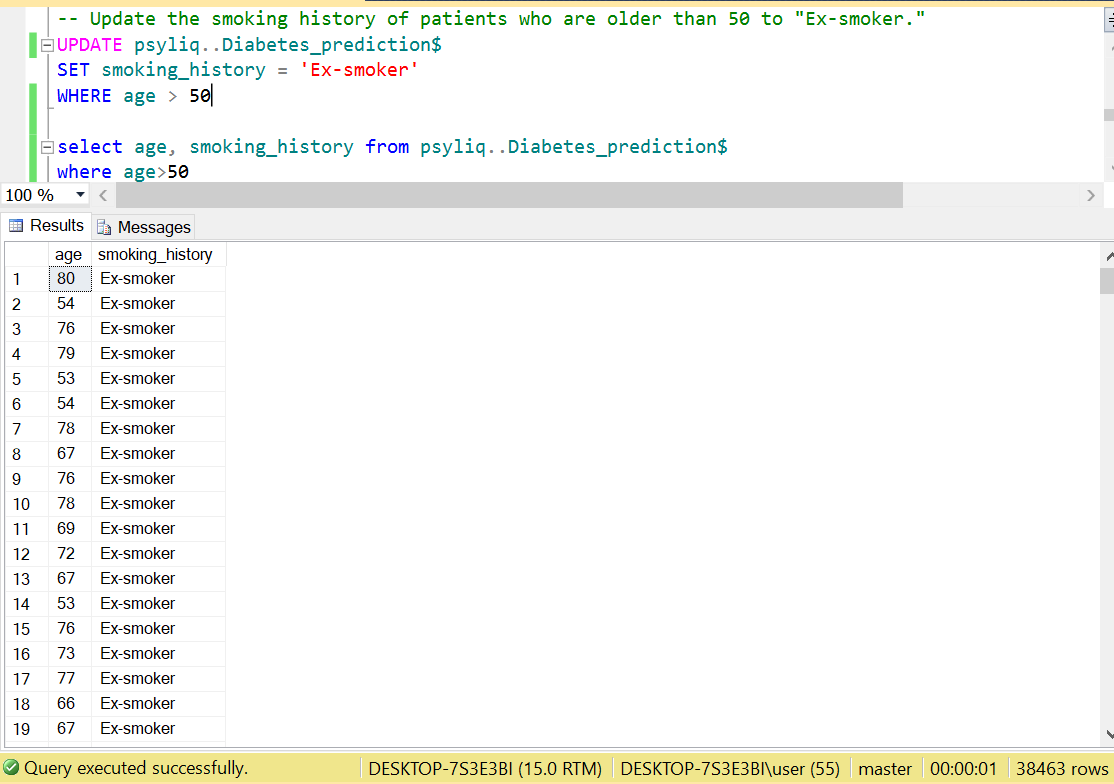
10. The oldest patients are found to be 80years with 1984 as their year of birth



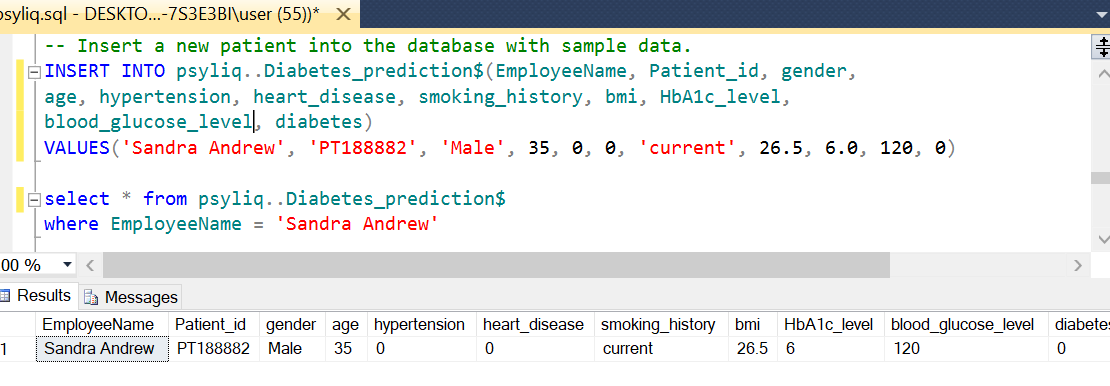
11. More females have lesser blood glucose level than males.



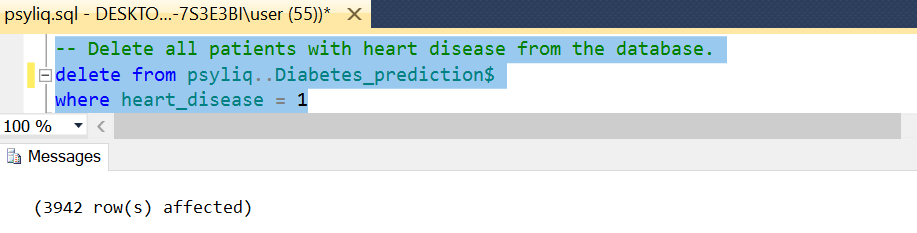
12. The patients who are older than 50 are 38,463 in number and their smoking history was updated to "Ex-smoker" as shown below.



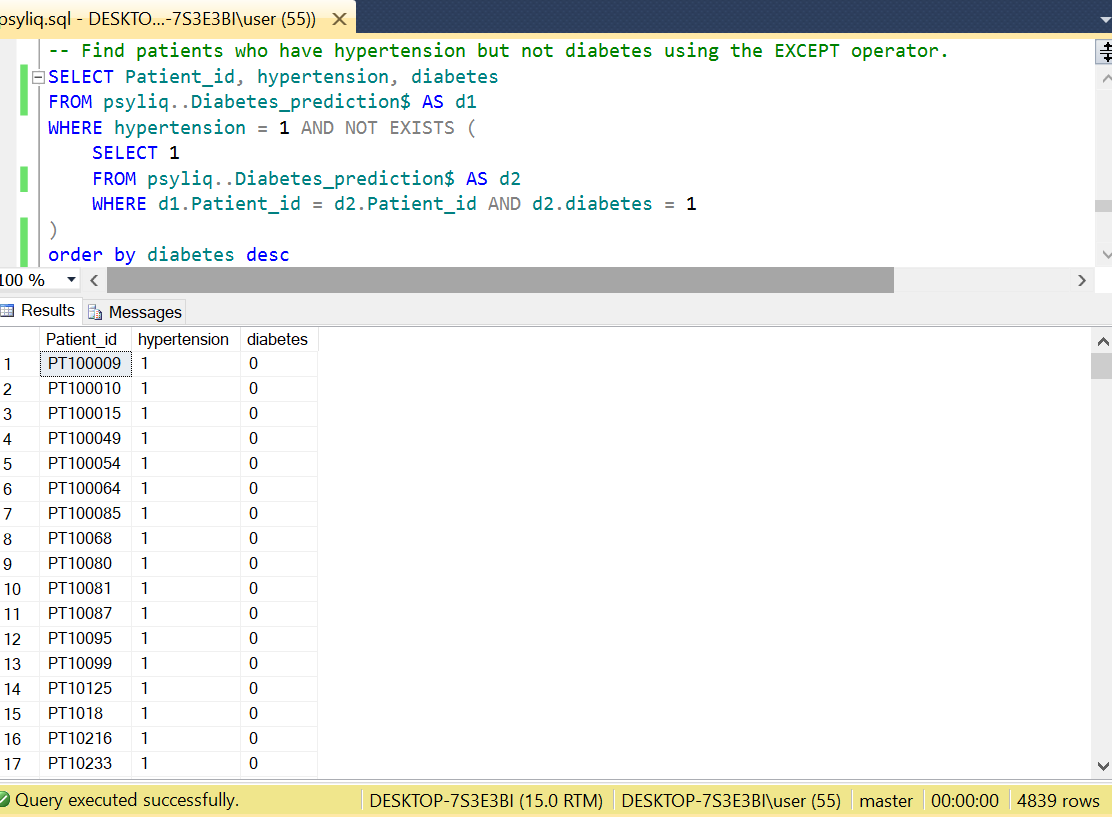
13. A new patient with the patient\_id “PT188882” is inserted into the database as shown below.



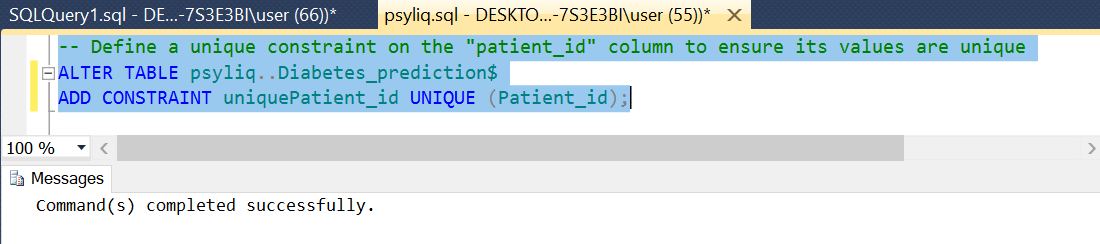
14. All patients with heart disease (3942) were deleted from the database with the query below.

,.

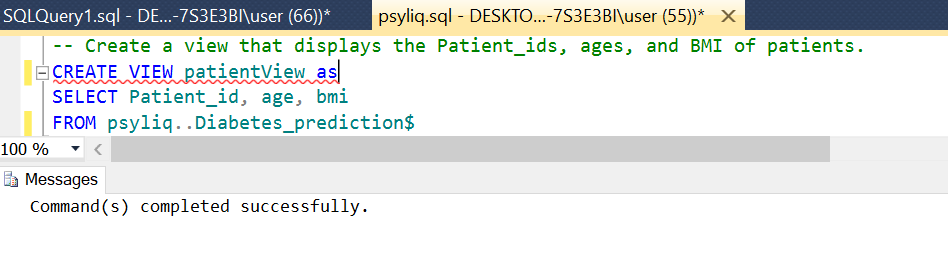
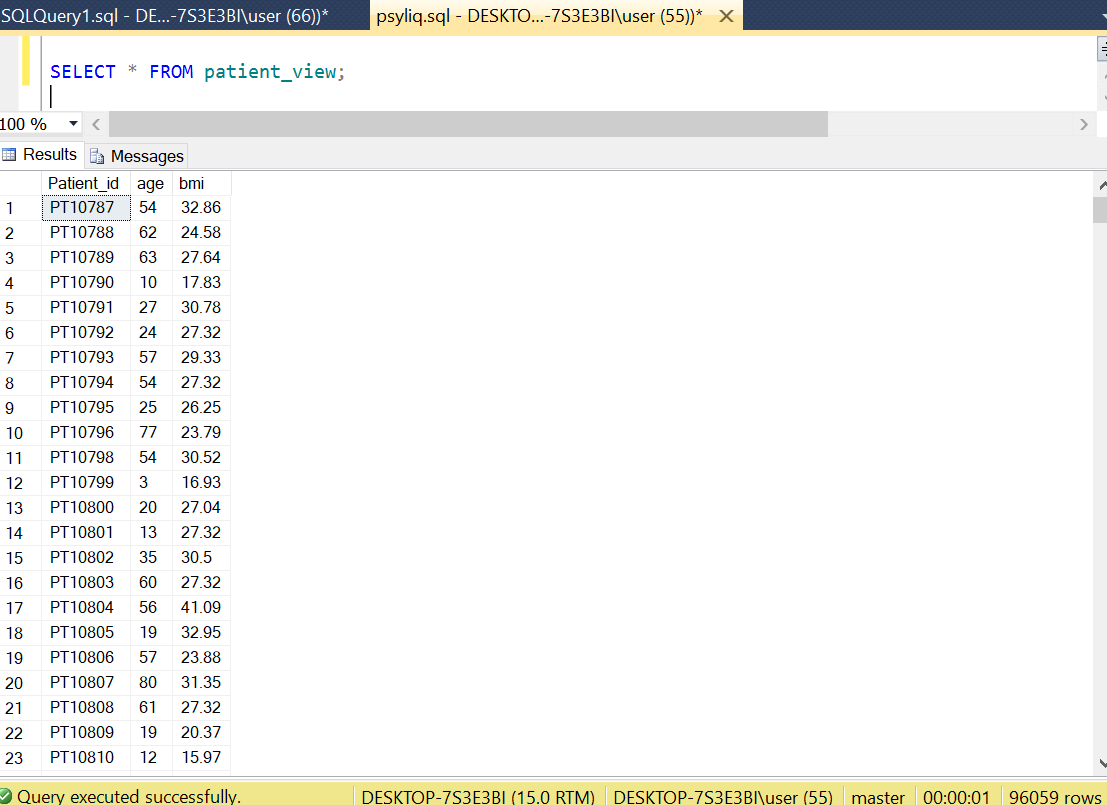
15. The patients with hypertension but not diabetes are 4,839



16. A unique constraint on the "patient\_id" was done with the following query.



17. A view that displays the Patient\_ids, ages, and BMI of patients

18. The current diabetes database schema exhibits a solid foundation, yet there’s room for optimization. The database schema can be improved through the following:

i. Splitting Patient Information: Patient information could be broken into separate tables to avoid redundant data. A patients’ table could be created to store general patient details (name, gender, age) and a health\_metrics’ table to store health-related information.

ii. Data Validation for Precision: Cconstraints or triggers to ensure age and health metrics fall within reasonable ranges can be implemented to promote accurate and reliable data.

iii. Documentation for Clarity: Relationships, foreign keys, and constraints should be documented to facilitate understanding for future developers working with the schema.

iv. Constraints for Data Integrity: Appropriate constraints, such as foreign key constraints should be added to maintain referential integrity between tables.

v. Review and Refine Data Types: Data types should be evaluated and refined to ensure they align optimally with the nature of the stored data.

19. Optimizing the performance of SQL queries on this dataset involves various strategies to enhance efficiency and reduce query execution times. Below are some key considerations and optimization techniques:

i. Indexing Columns: Frequently queried columns can be identified and indexed to enhance query performance.

ii. Selective Retrieval for Precision: When crafting SELECT statements, the needed columns should be queried. Instead of opting for SELECT \*, only the necessary fields should be selectively retrieved. This reduces data transfer volumes and speeds up query execution.

iii. Avoid SELECT DISTINCT: While SELECT DISTINCT is a powerful tool, it can be resource-intensive. Its usage should be avoided while alternative approaches can be explored if possible.

iv. Limiting Results with LIMIT: For queries that potentially return large result sets, LIMIT clause should be used to restrict the number of rows returned. This is particularly beneficial for paginated displays and can significantly enhance query performance.

v. Harnessing the Power of Caching: Caching mechanisms at the database or application level can be used to store frequently used query results. Caching is a valuable strategy for queries involving relatively static data.

vi. Parameterized Queries for Security and Efficiency: To safeguard against SQL injection risks and to improve the efficiency of query execution plans, parameterized queries can be embraced especially for frequently reused queries with different parameter values.

**Conclusion**

In this analysis, diabetes\_prediction data has been explored with SQL along with schema improvement, and query optimization. From different insights in patient records to create a more efficient database structure and fine-tuning SQL queries, each step contributes to a more insightful and responsive data experience.

**The Overall SQL Queries**

-- Retrieving the overall raw data

select \* from psyliq..Diabetes\_prediction$

--Retrieve the Patient\_id and ages of all patients

select Patient\_id, age

from psyliq..Diabetes\_prediction$

-- Select all female patients who are older than 40.

select \* from psyliq..Diabetes\_prediction$

where gender = 'female' and age >40

order by age asc

-- Calculate the average BMI of patients.

select avg(bmi) as AverageBMI, SUM(bmi)/COUNT(Patient\_id) 'Average BMI'

from psyliq..Diabetes\_prediction$

-- List patients in descending order of blood glucose levels.

select \* from psyliq..Diabetes\_prediction$

order by blood\_glucose\_level desc

-- Find patients who have hypertension and diabetes

select \* from psyliq..Diabetes\_prediction$

where hypertension = 1 and diabetes = 1

order by diabetes asc

-- Determine the number of patients with heart disease.

select count(heart\_disease) as Heart\_DiseasePatients from psyliq..Diabetes\_prediction$

where heart\_disease = 1

-- Group patients by smoking history and count how many smokers and nonsmokers there are.

select smoking\_history, COUNT(smoking\_history)

from psyliq..Diabetes\_prediction$

group by smoking\_history

-- Retrieve the Patient\_ids of patients who have a BMI greater than the average BMI

select Patient\_id, bmi from psyliq..Diabetes\_prediction$

where bmi>(SELECT AVG(bmi) FROM psyliq..Diabetes\_prediction)

--order by bmi asc

-- Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel

select \*

from psyliq..Diabetes\_prediction$

-- where HbA1c\_level = 9

order by HbA1c\_level desc

select \*

from psyliq..Diabetes\_prediction$

-- where HbA1c\_level = 3.5

order by HbA1c\_level

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

SELECT EmployeeName, Patient\_id,

2023 - age AS Birth\_Years,

age AS current\_age

FROM psyliq..Diabetes\_prediction$

order by Birth\_Years

-- Rank patients by blood glucose level within each gender group

SELECT

Patient\_id,

gender,

blood\_glucose\_level,

RANK() OVER (PARTITION BY gender ORDER BY blood\_glucose\_level) AS glucose\_level\_rank

FROM psyliq..Diabetes\_prediction$

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

UPDATE psyliq..Diabetes\_prediction$

SET smoking\_history = 'Ex-smoker'

WHERE age > 50

select age, smoking\_history from psyliq..Diabetes\_prediction$

where age>50

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

INSERT INTO psyliq..Diabetes\_prediction$(EmployeeName, Patient\_id, gender,

age, hypertension, heart\_disease, smoking\_history, bmi, HbA1c\_level,

blood\_glucose\_level, diabetes)

VALUES('Sandra Andrew', 'PT188882', 'Male', 35, 0, 0, 'current', 26.5, 6.0, 120, 0)

select \* from psyliq..Diabetes\_prediction$

where EmployeeName = 'Sandra Andrew'

-- Delete all patients with heart disease from the database.

delete from psyliq..Diabetes\_prediction$

where heart\_disease = 1

-- Find patients who have hypertension but not diabetes using the EXCEPT operator.

SELECT Patient\_id, hypertension, diabetes

FROM psyliq..Diabetes\_prediction$ AS d1

WHERE hypertension = 1 AND NOT EXISTS (

SELECT 1

FROM psyliq..Diabetes\_prediction$ AS d2

WHERE d1.Patient\_id = d2.Patient\_id AND d2.diabetes = 1

)

order by diabetes desc

-- Define a unique constraint on the "patient\_id" column to ensure its values are unique

ALTER TABLE psyliq..Diabetes\_prediction$

ADD CONSTRAINT uniquePatient\_id UNIQUE (Patient\_id);

-- Create a view that displays the Patient\_ids, ages, and BMI of patients.

CREATE VIEW patientView as

SELECT Patient\_id, age, bmi

FROM psyliq..Diabetes\_prediction$

SELECT \* FROM patient\_view;