Diabetes Prediction

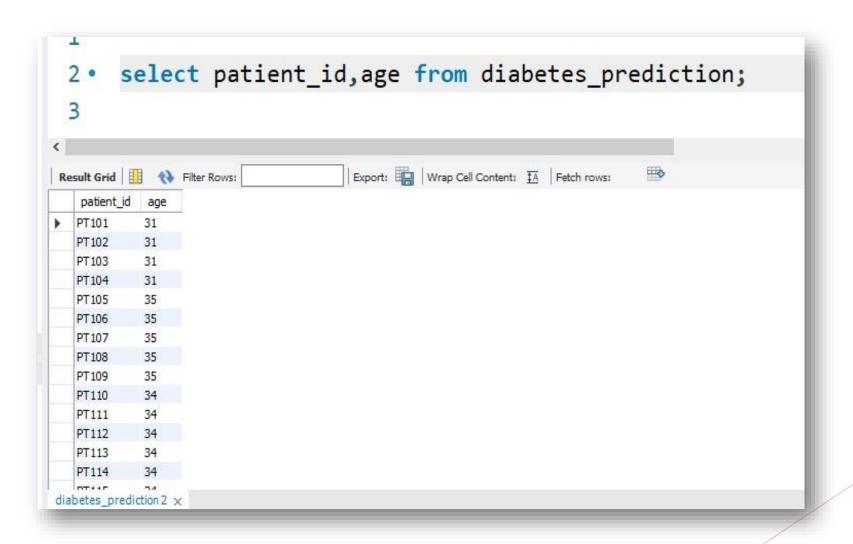


Internship Project [PSYLIQ]

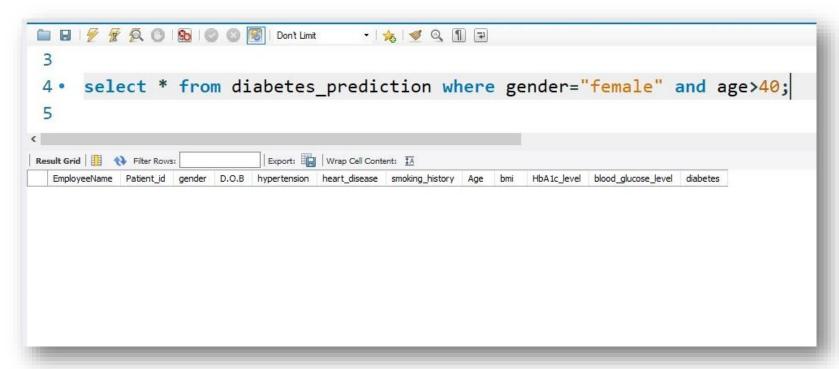
Introduction

This dataset is provided by PSYLIQ, In this project, I will use the Diabetes Prediction data set to explore various aspects Diabetes and how they affect patients. The data set contains information about 100000 patients which are diabetes patients and their details such as Patient_id,gender, age, hypertension, heart_disease,smoking_history, bmi, HbA1c_level blood_glucose_level, diabetes.

1. Retrieve the Patient_id and ages of all patients.

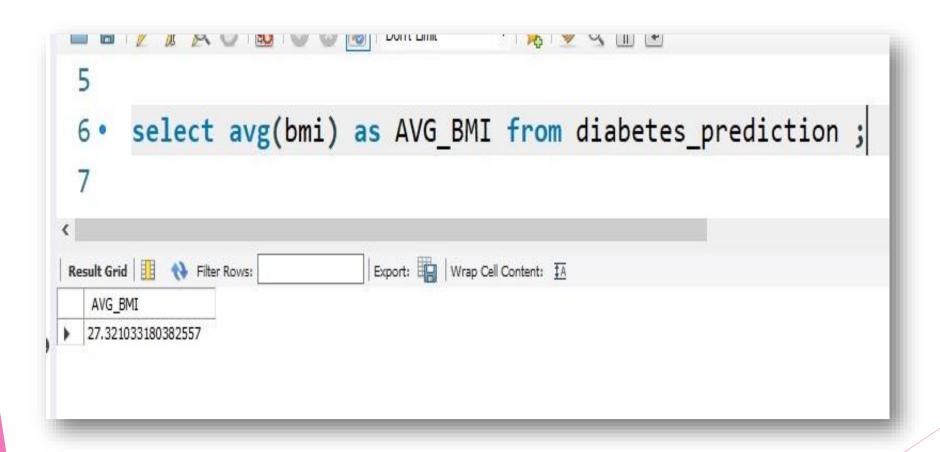


2. Select all female patients who are older than 40.

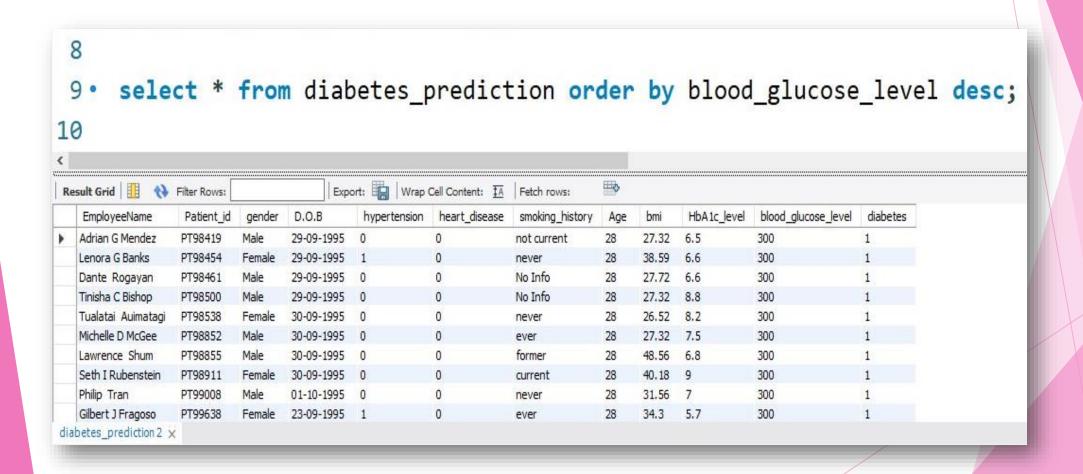


Because I don't have age greater than 40. so it retrieve nothing after execution

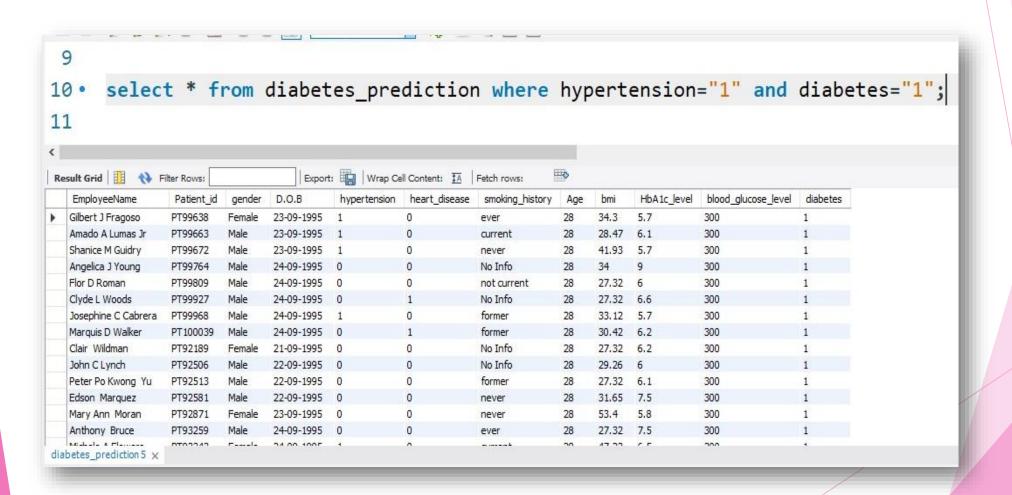
3. Calculate the average BMI of patients.



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



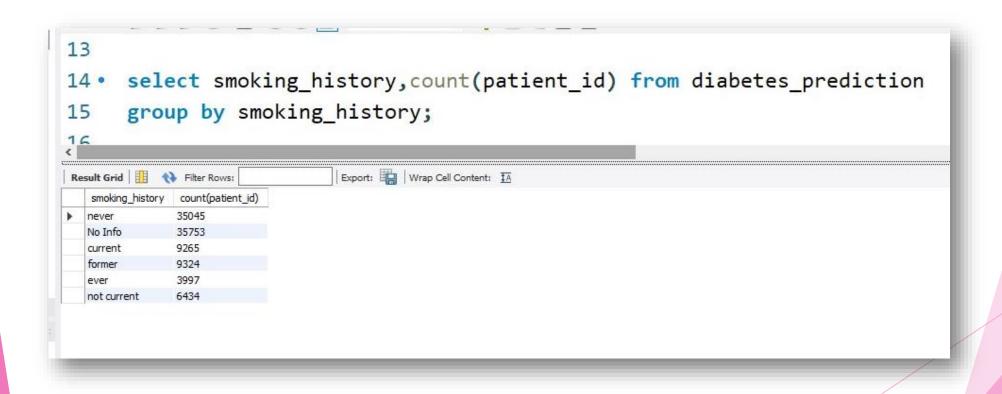
5. Find patients who have hypertension and diabetes.



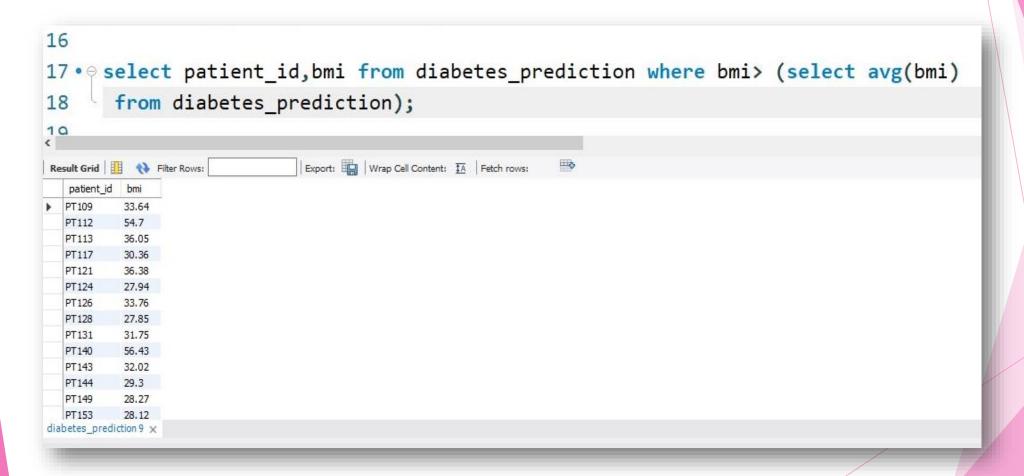
6. Determine the number of patients with heart disease.

```
11
      select count(patient_id) from diabetes_prediction where heart_disease=1;
13
                            Export: Wrap Cell Content: IA
  count(patient_id)
3937
```

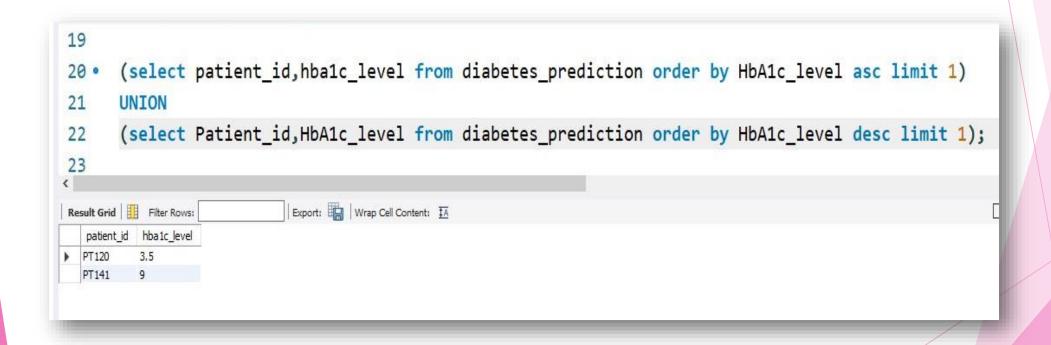
7. Group patients by smoking history and count how many smokers and non-smokers there are



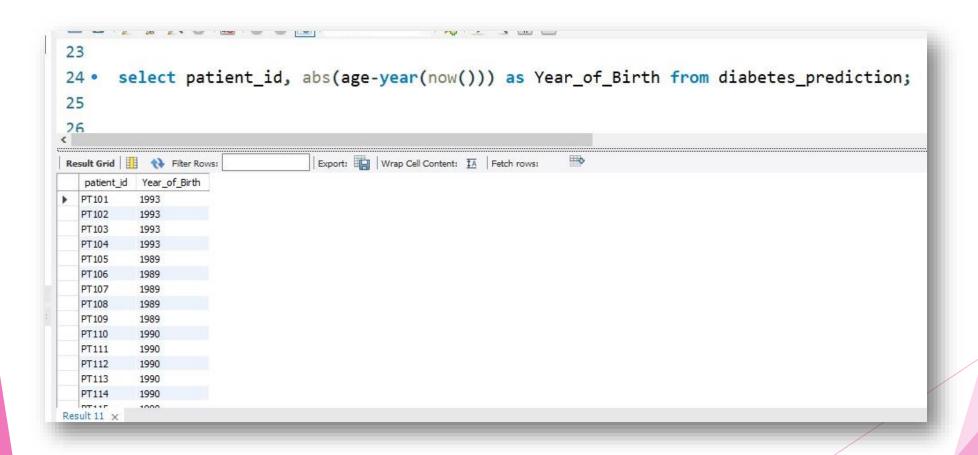
8. Retrieve the Patient_ids of patients who have a BMI greater than the average BMI.



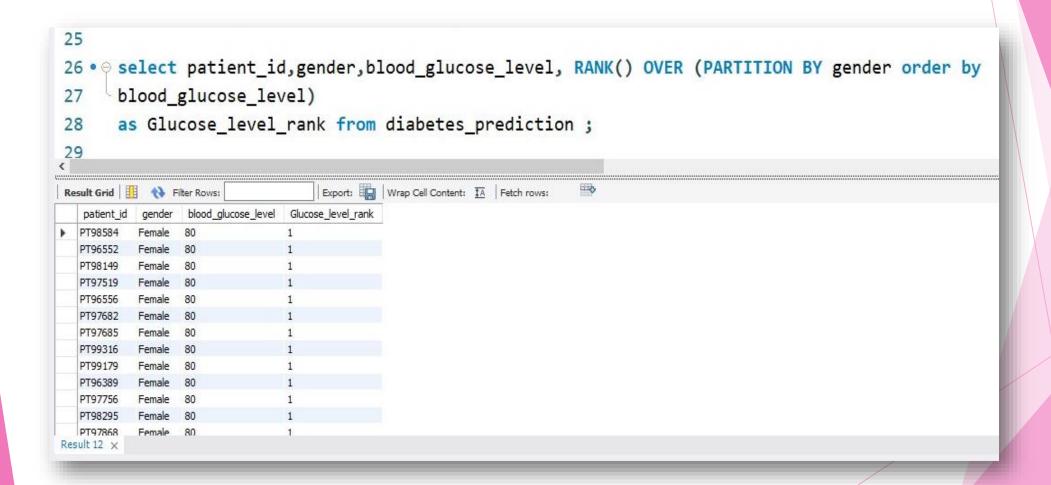
9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.



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."

```
Dont Limit

SET SQL_SAFE_UPDATES = 0;

update diabetes_prediction set smoking_history="Ex-smoker" where age>50;

where age>50;
```

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

```
insert into diabetes_prediction
values("Vyas Kumar","PT101010","Male","30-03-1999",0,0,"never",25.5,6.1,120,0,24);
9 • select * from diabetes prediction;
 Export: Wrap Cell Content: TA Fetch rows:
                                                                                         blood glucose level
  EmployeeName
                 Patient id
                                        hypertension heart_disease smoking_history
                                                                                HbA1c level
  Ruth S Bacuvani
                                                                          40.69
  Jessica K Aldaz
                                                          No Info
                 PT100095
                                                                          24.6
  William Chun
                                                          No Info
                                                                          27.32 6.2
  Antoinette L Wells
                                                          No Info
                               24-09-1995
  Richard D Swart
                                                          former
```

current

35.42 4

22.43 6.6

Vivian Chu

/vas Kumar

Savitree Satram

PT100100

PT101010

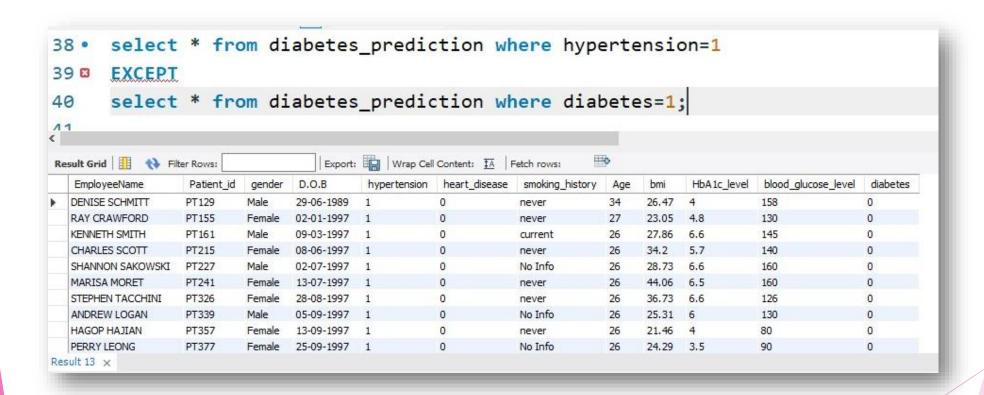
25-09-1995

30-03-1999

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

delete from diabetes_prediction where heart_disease=1;

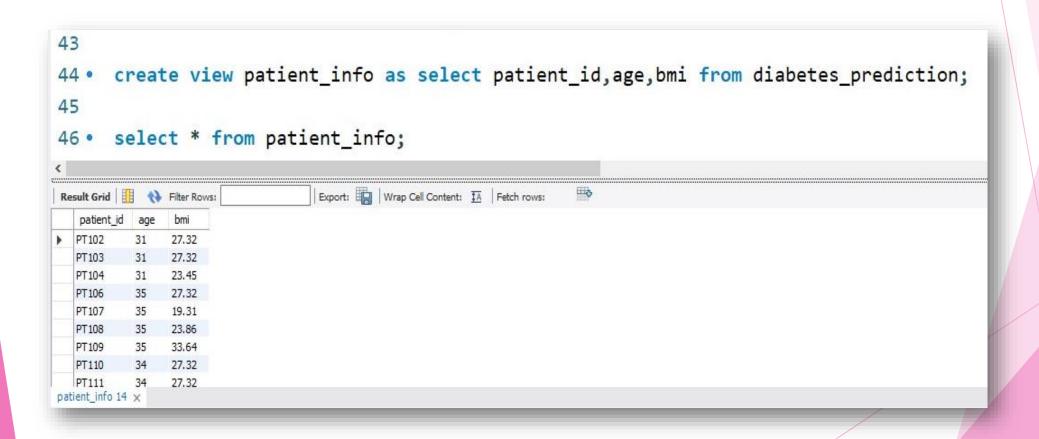
15. Find patients who have hypertension but not diabetes using the EXCEPT operator.



16. Define a unique constraint on the "patient_id" column to ensure its values are unique.

alter table diabetes_prediction modify Patient_id varchar(255) 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.

To reduce data redundancy and improve data integrity in a database schema, you can consider the following best practices:

- ▶ 1. Normalization: Break down large tables, apply normalization techniques (1NF, 2NF, 3NF).
- ▶ 2. Primary Keys: Ensure each table has a unique primary key.
- > 3. Foreign Keys: Establish relationships between tables for referential integrity.
- ▶ 4. Avoid Redundant Columns: Refrain from duplicating data across multiple tables.
- ▶ 5. Data Types: Use appropriate data types for columns.
- ▶ 6. Default Values and Constraints: Set defaults, use constraints to enforce rules.
- > 7. Indexes: Apply indexes for frequently queried columns (considering trade-offs).
- ▶ 8. Avoid Nulls: Minimize NULL values; use defaults or separate tables.
- 9. Data Validation: Enforce validation at both application and database levels.
- 10. Use of Views: Presents data from multiple tables without duplicating information, simplifying queries, and maintaining a consistent view.

19. Explain how you can optimize the performance of SQL queries on this dataset.

Indexing:

- Create indexes on key columns for faster data retrieval.
- Query Optimization:
 - Simplify queries by minimizing joins and subqueries.
- Result Set Management:
 - ▶ Use the LIMIT clause to restrict returned rows.
 - ► Fetch only necessary columns, avoiding SELECT *.
- Stored Procedures:
 - Encapsulate frequent queries in stored procedures.
- Statistics Update:
 - Regularly update table statistics for accurate query optimization.
- Distinct Usage:
 - ▶ Minimize the use of SELECT DISTINCT; explore alternatives like GROUP BY or refined logic.

