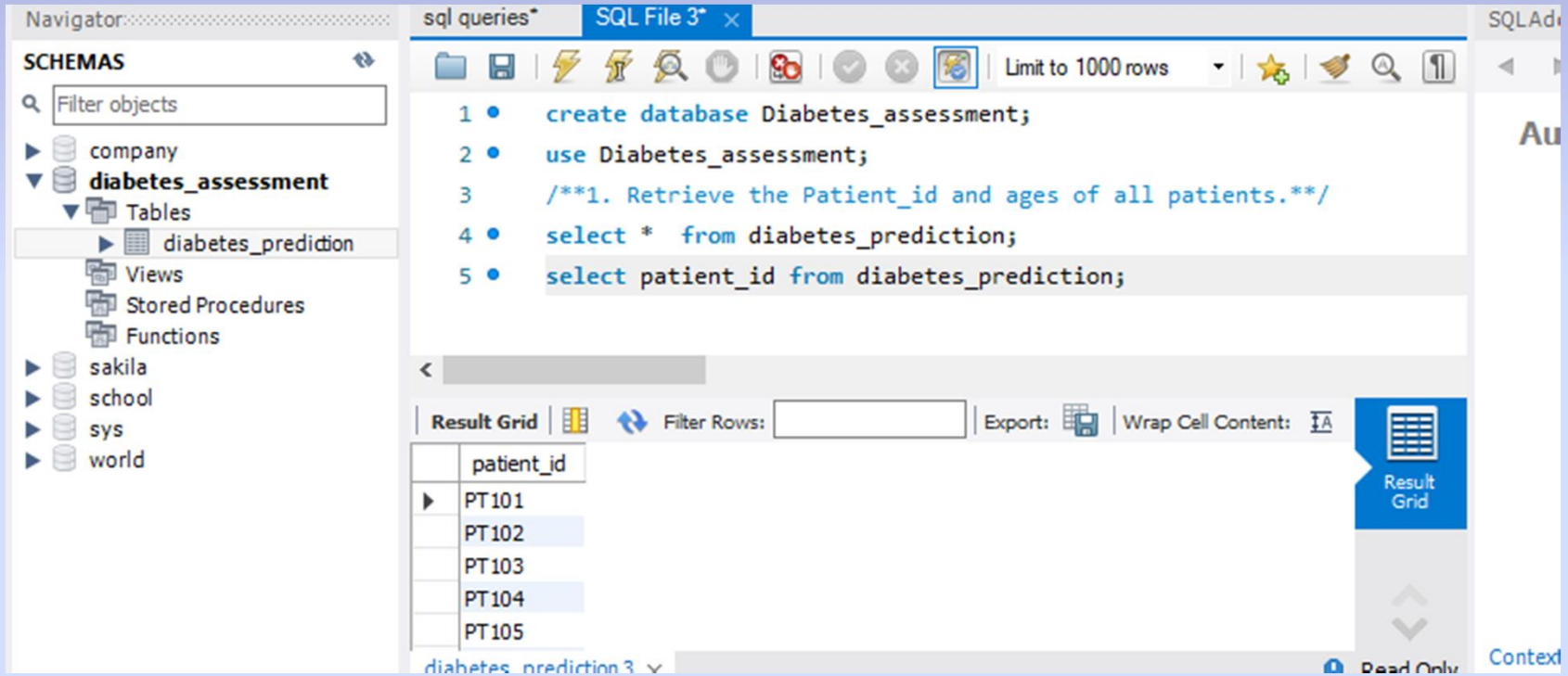
Diabetes Prediction

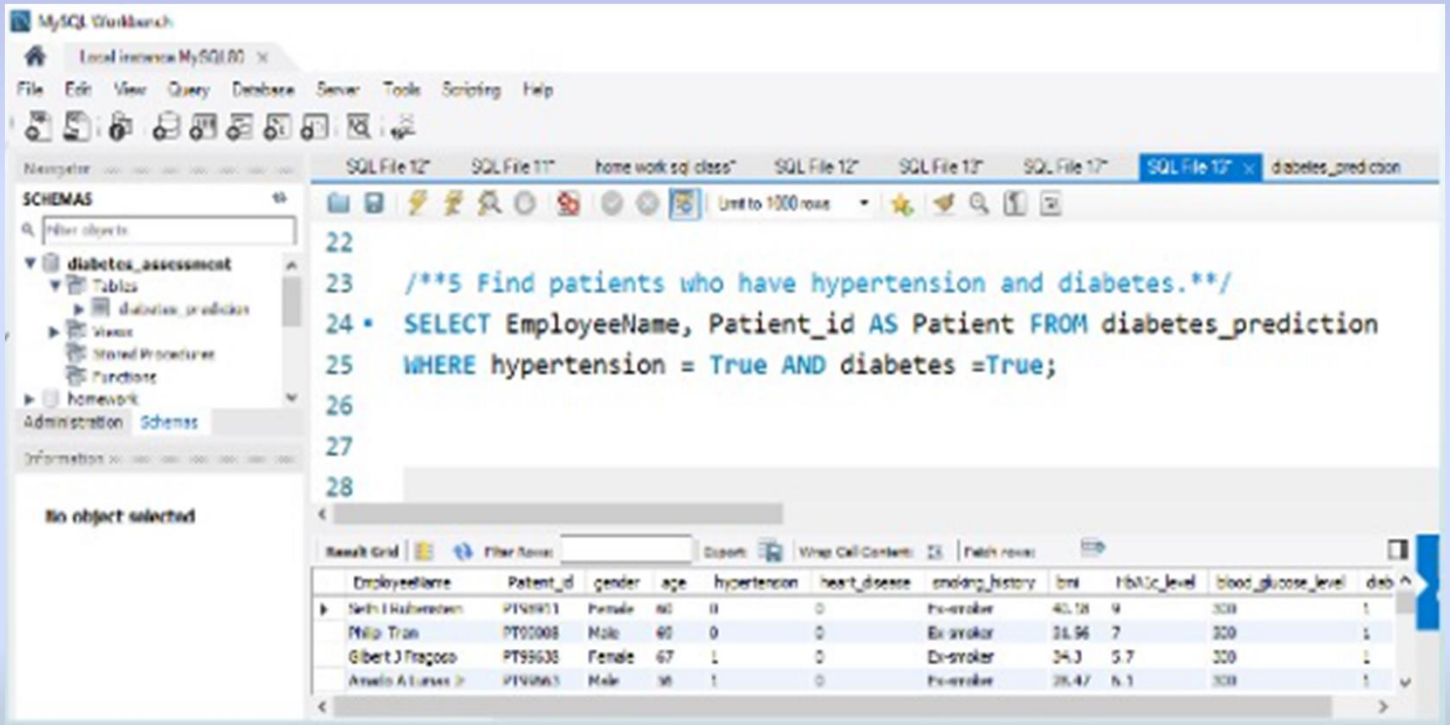
PSYLIQ DATA ANALYST INTERNSHIP
NEHA DANDEKAR
TASK 2

1. Retrieve the Patient_id of all patients.

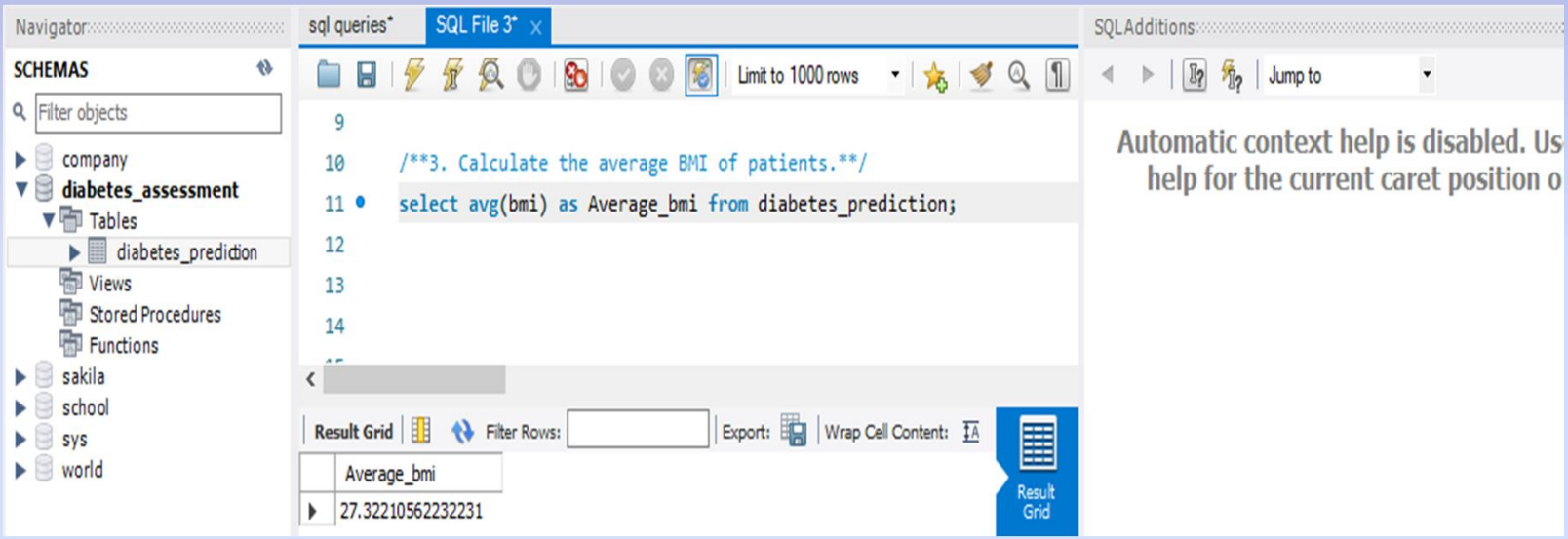


2. Select all female patients who are older than 40.

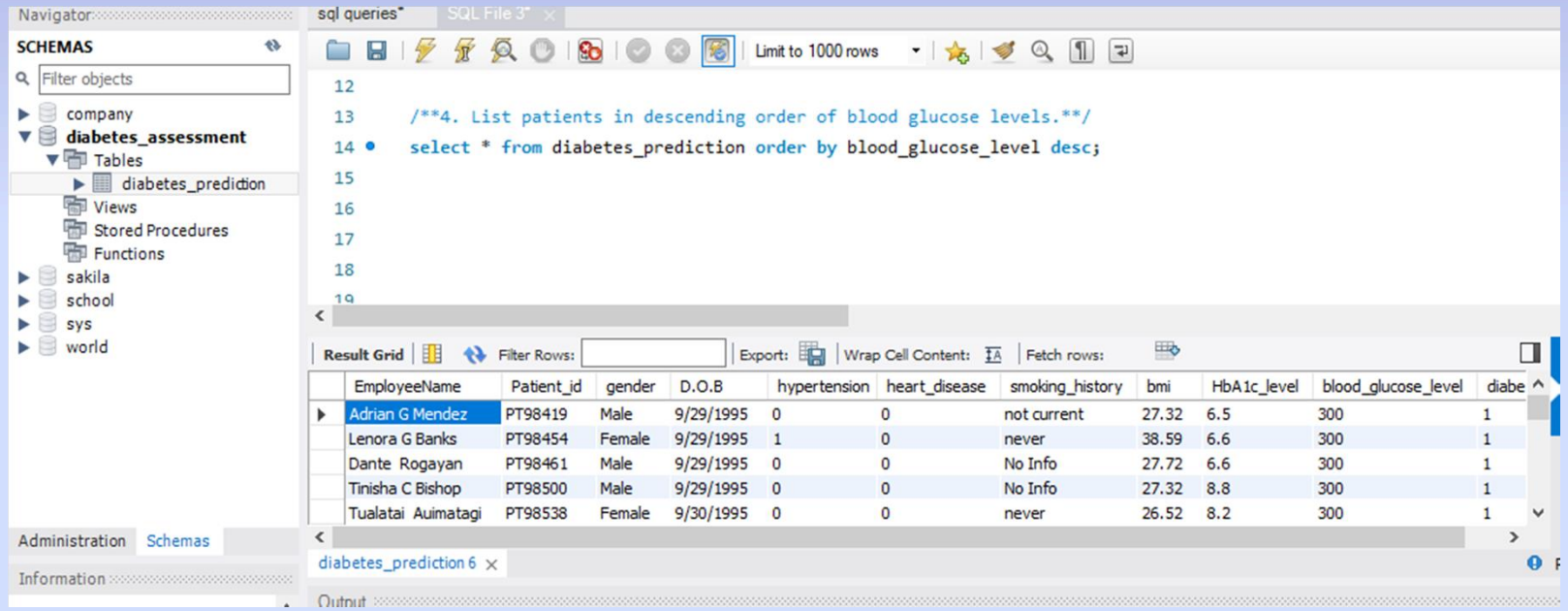
- select * from diabetes_prediction WHERE gender= 'Female' AND age > 40;



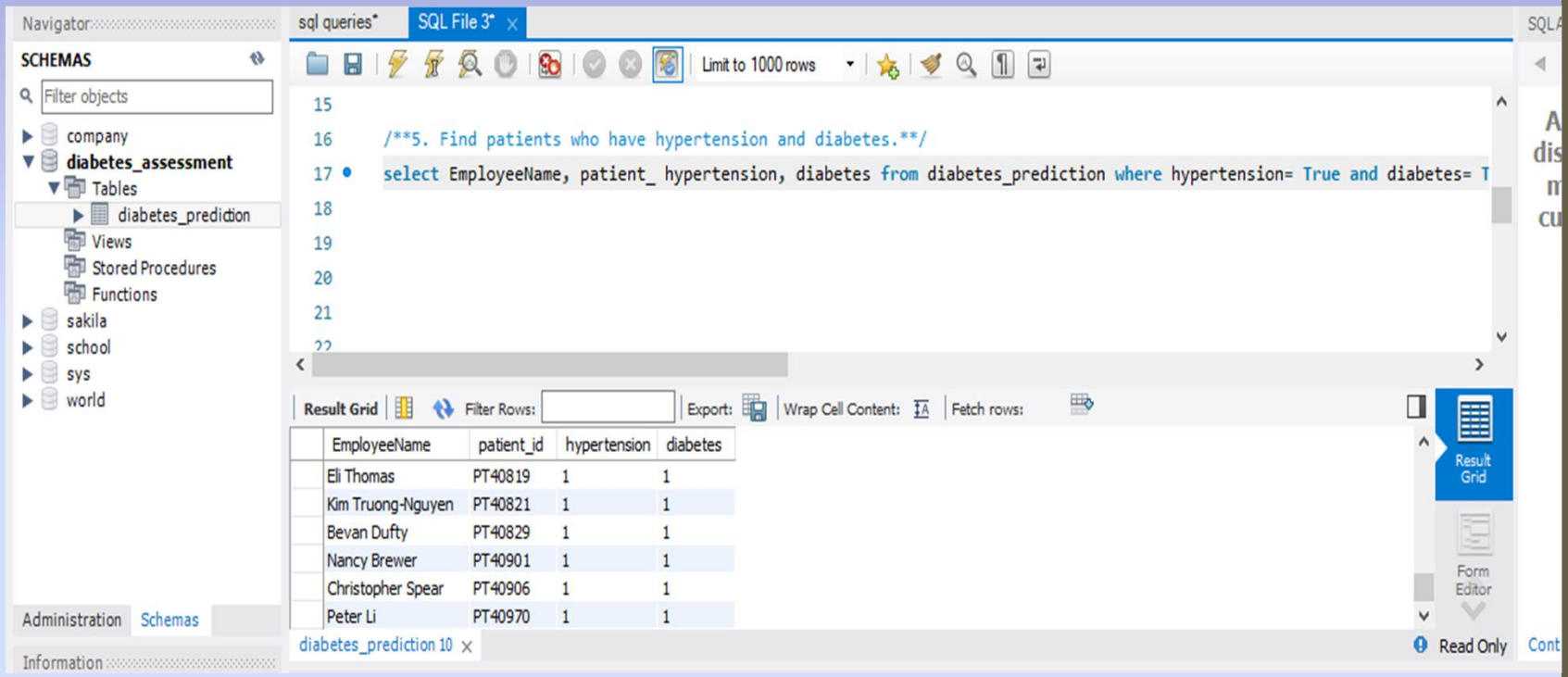
3. Calculate the average BMI of patients.



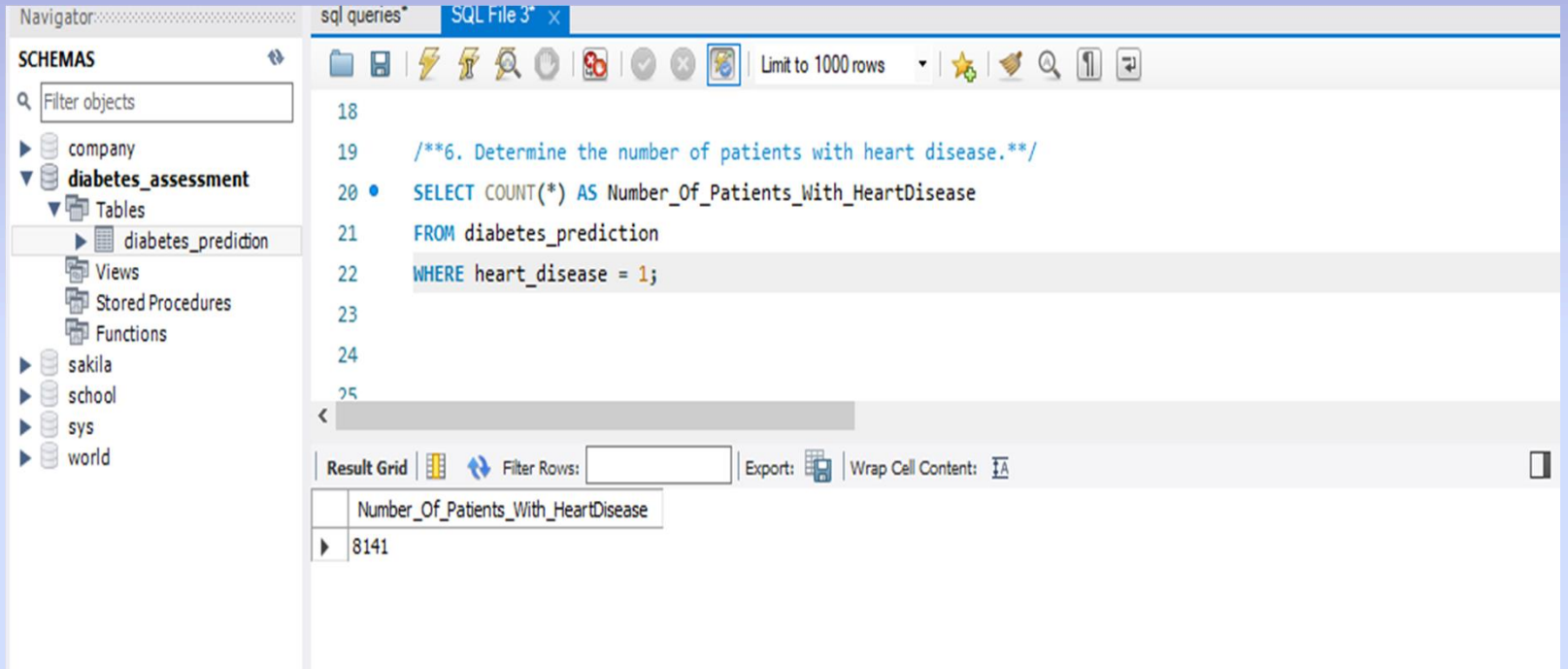
4. List patients in descending order of blood glucose levels.



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 non-smokers there are.

sql queries* | SQL File 3*

Limit to 1000 rows

```sql
/**7. Group patients by smoking history and count how many smokers and non-smokers there are.**/
select smoking_history , count(*) as count_of_smokers from diabetes_prediction group by smoking_history;
```

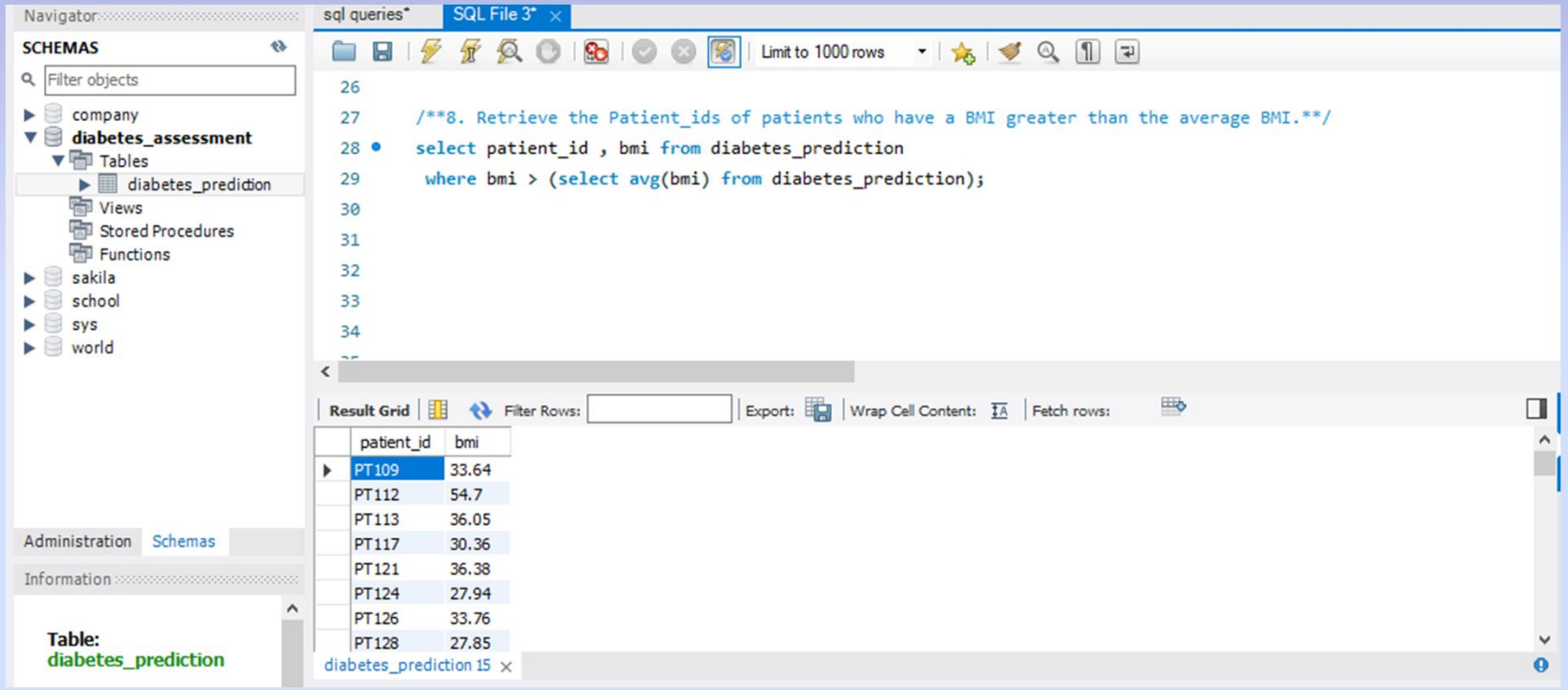
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

smoking_history	count_of_smokers
never	72523
No Info	74022
current	19183
former	19333
ever	8263
not current	13299

Result Grid | Form Editor

Result 14 | Read Only

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.

```
/**9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.**/
select EmployeeName, Patient_id, HbA1c_level,
case
when HbA1c_level=(select max(HbA1c_level) from diabetes_prediction)
then 'Highest'
when HbA1c_level=(select min(HbA1c_level) from diabetes_prediction)
then 'Highest'
end as HbA1c_level_stat
from diabetes_prediction
where HbA1c_level=(select max(HbA1c_level) from diabetes_prediction) or
HbA1c_level=(select min(HbA1c_level) from diabetes_prediction);
```

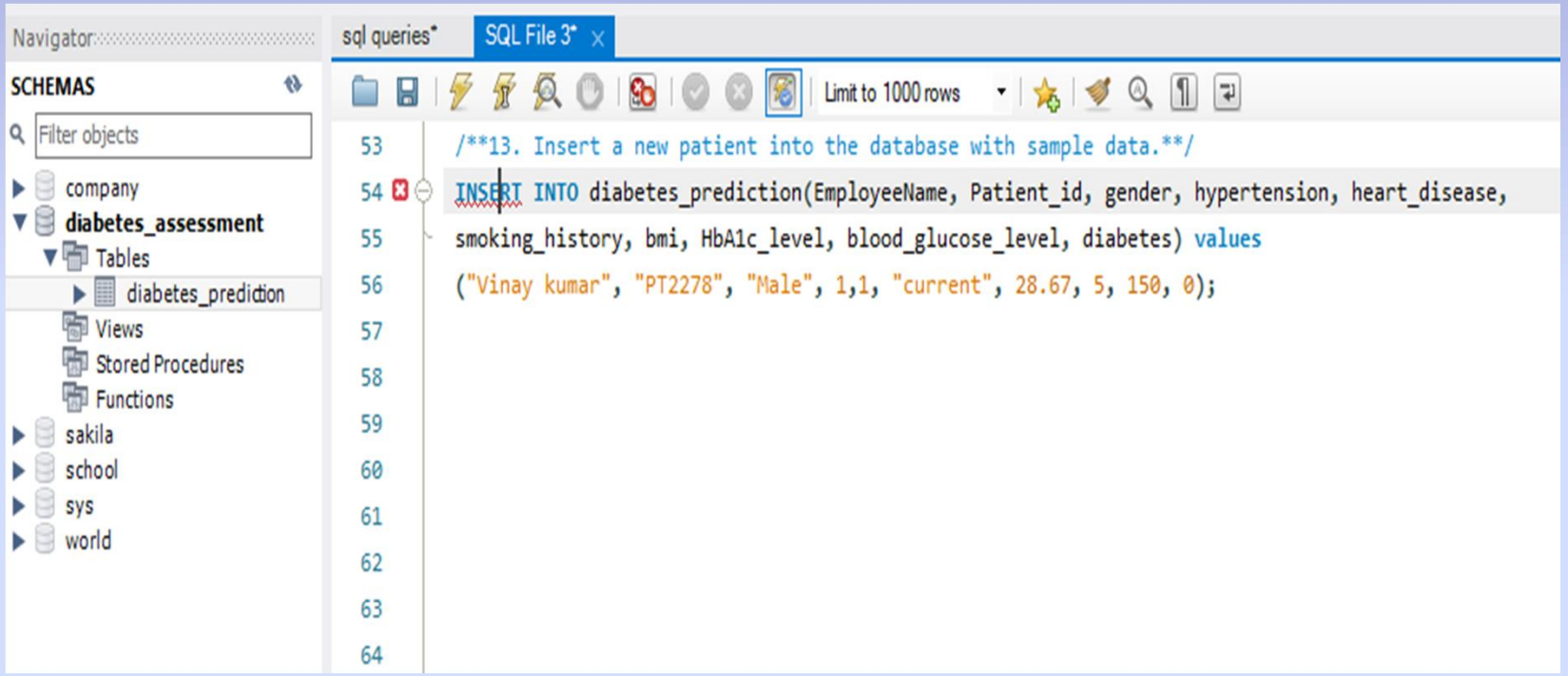
EmployeeName	Patient_id	HbA1c_level	HbA1c_level_stat
ELLEN MOFFATT	PT120	3.5	Highest
JOHN TURSI	PT134	3.5	Highest
MICHAEL THOMPSON	PT141	9	Highest
SHARON MCCOLE WICHER	PT145	3.5	Highest
KEVIN CASHMAN	PT156	9	Highest
MARK KEARNEY	PT158	3.5	Highest
MONIQUE MOYER	PT174	3.5	Highest

11. Rank patients by blood glucose level within each gender group.

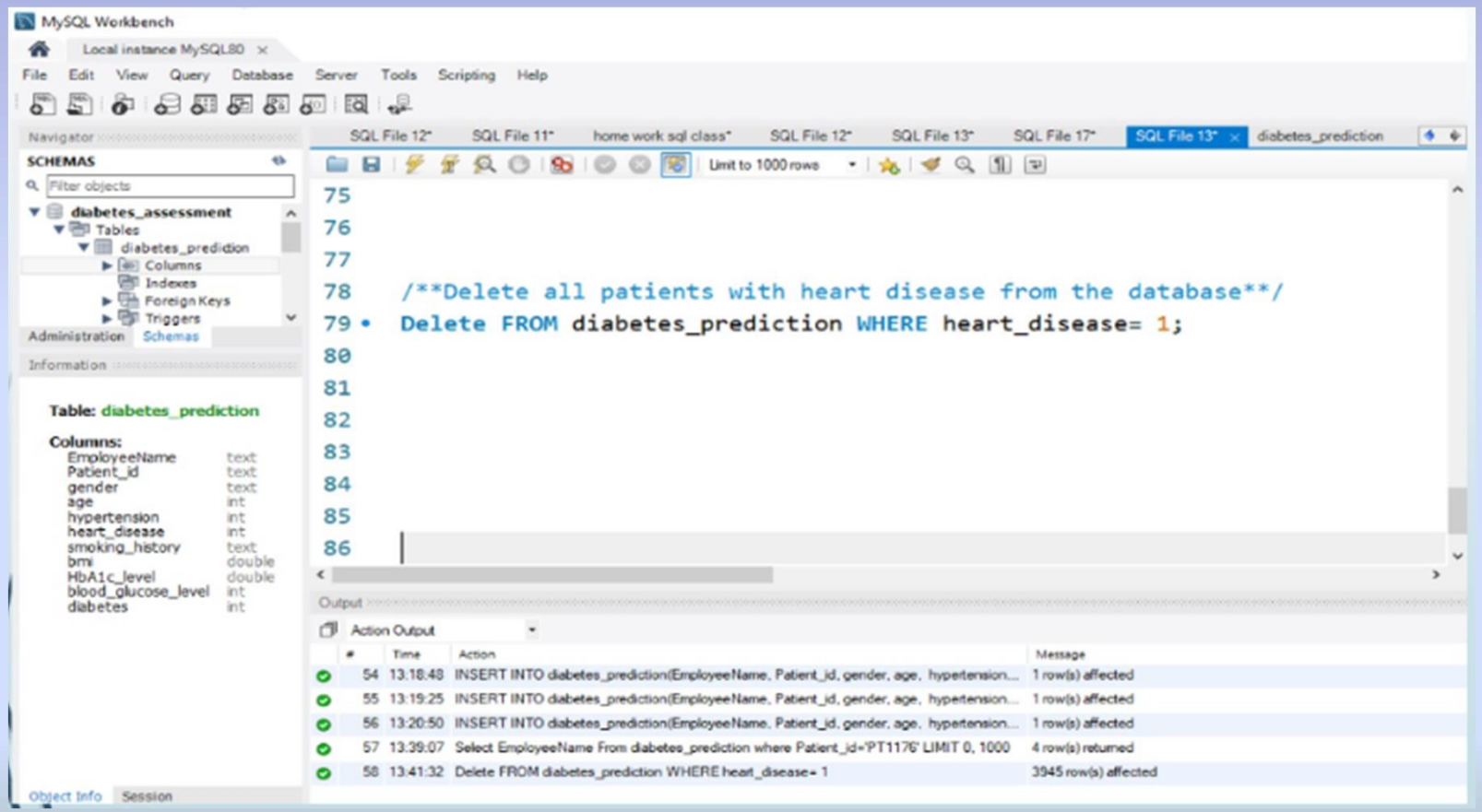
```
/**11. Rank patients by blood glucose level within each gender group.**/
select Patient_id, gender, blood_glucose_level, Rank()
over(partition by gender order by blood_glucose_level) as
Glucose_Level_Rank from diabetes_prediction;
```

Patient_id	gender	blood_glucose_level	Glucose_Level_Rank
PT78487	Female	80	1
PT80920	Female	80	1
PT78492	Female	80	1
PT79735	Female	80	1
PT77410	Female	80	1
PT79191	Female	80	1
PT79182	Female	80	1

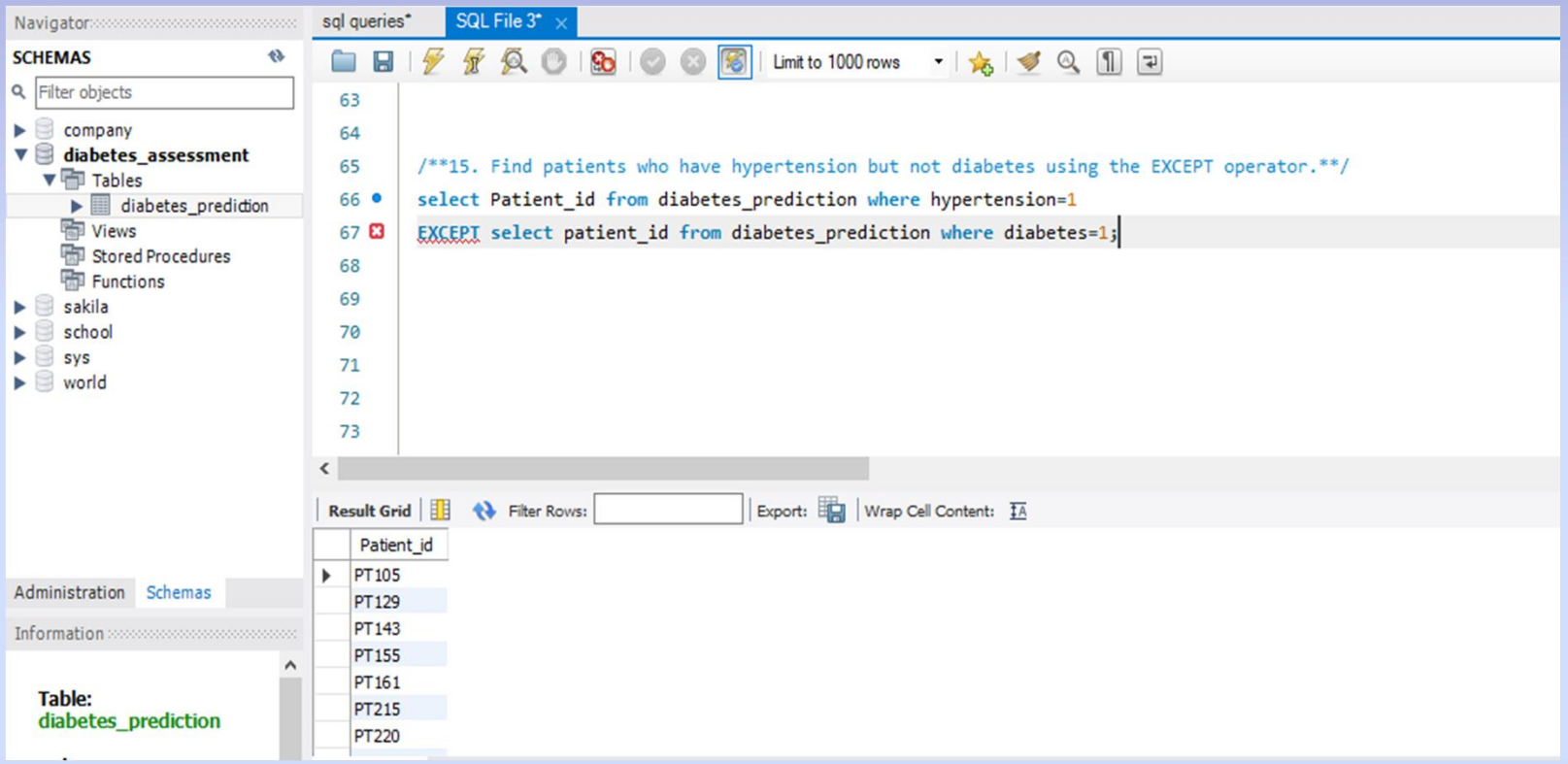
13. Insert a new patient into the database with sample data.



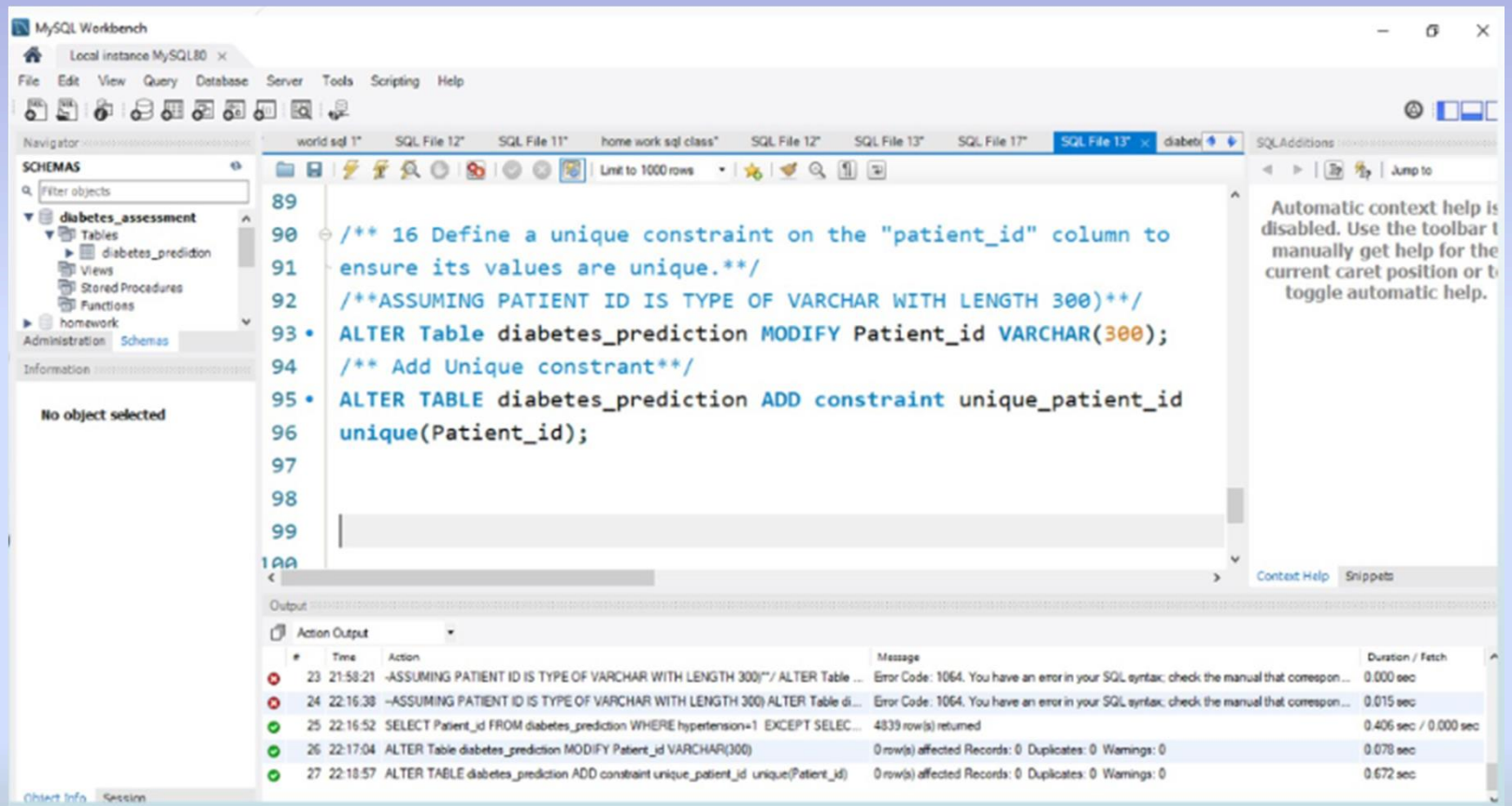
14. Delete all patients with heart disease from the database.



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.

sql queries* | SQL File 3*

Limit to 1000 rows

```
/**17. Create a view that displays the Patient_ids, ages, and BMI of patients.**/
create view PatientBMI as select patient_id,
 bmi from diabetes_prediction;
 select * from PatientBMI;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

patient_id	bmi
PT101	25.19
PT102	27.32
PT103	27.32
PT104	23.45
PT105	20.14

PatientBMI 20 | Read Only

Output

Action Output

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 improvements:
- **Normalization:** Normalize the database schema to reduce data redundancy and ensure data integrity. This involves organizing the data into separate tables based on functional dependencies and avoiding repeating groups in a single table.
- **Use of Foreign Keys:** Implement foreign keys to establish relationships between tables. This ensures referential integrity, where data in one table (child table) that references data in another table (parent table) is consistent and accurate.
- **Regular Data Quality Checks:** Implement processes for regular data quality checks and data cleansing.
- **Data Validation Rules:** Define data validation rules to ensure that data entered into the database meets specified criteria.

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

- **Enhance query efficiency by abstaining from the use of 'SELECT' and choose columns that really need for better performance.**
- **Use efficiently the WHERE clause to work with only the necessary records.**
- **Prefer the use of joins instead of subqueries whenever it's possible.**
- **use windows to work with the data of interest.**
- **Use Aggregate functions like Count, Average, to reduce processing time.**