PSYLIQ

DIABETES PREDICTION – SQL

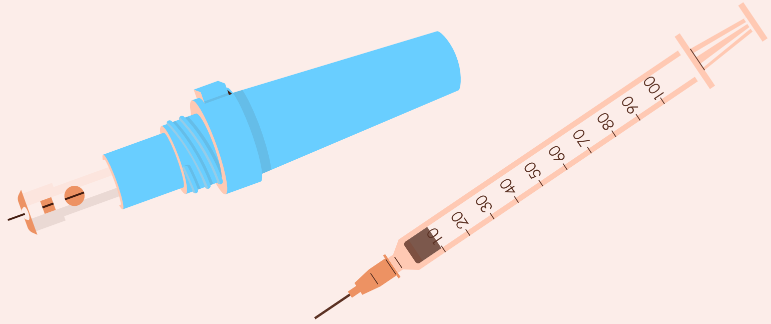
PSYLIQ Internship Project

Created by: Divya Pardeshi

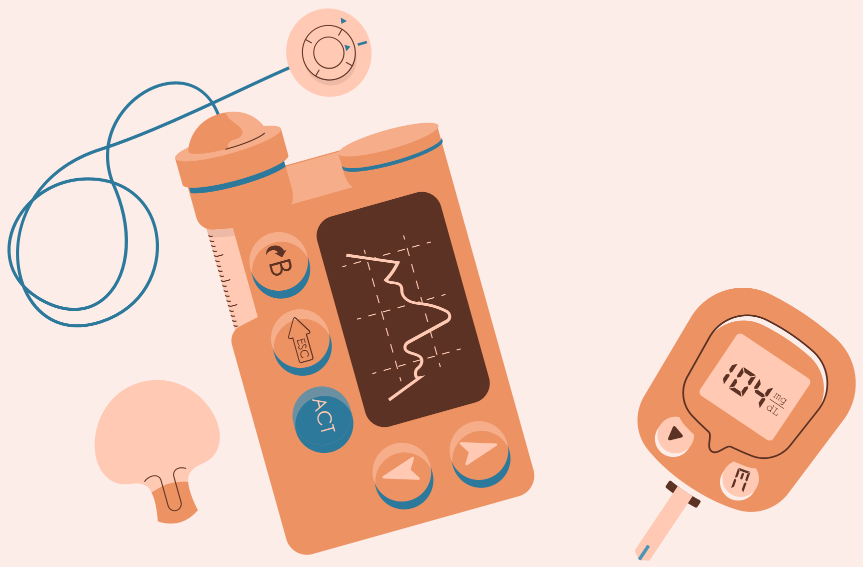
CONTENT

- Introduction
- Project Questions & SQL Queries
- Summary

INTRODUCTION



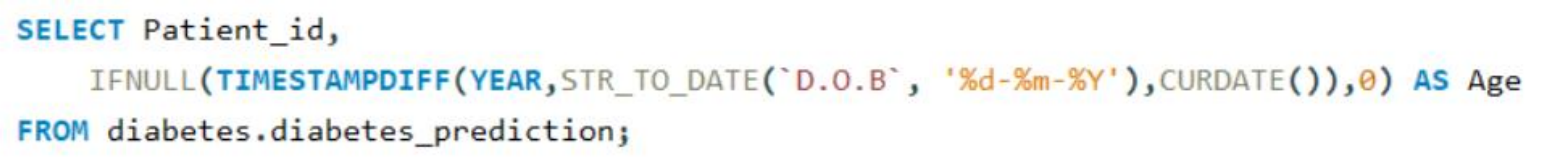
In healthcare, integrating data-driven methodologies is crucial for deriving valuable insights and enhancing patient outcomes. This project undertakes a comprehensive analysis of the "Diabetes Prediction" dataset provided by PSYLIQ, which includes medical records for over 100,000+ patients. The dataset covers key attributes such as hypertension, heart disease, smoking history, BMI (Body Mass Index), diabetes, and other essential factors. By harnessing the power of MySQL, this project aims to reveal significant patterns and trends that can drive better prevention, early detection, and management of diabetes.



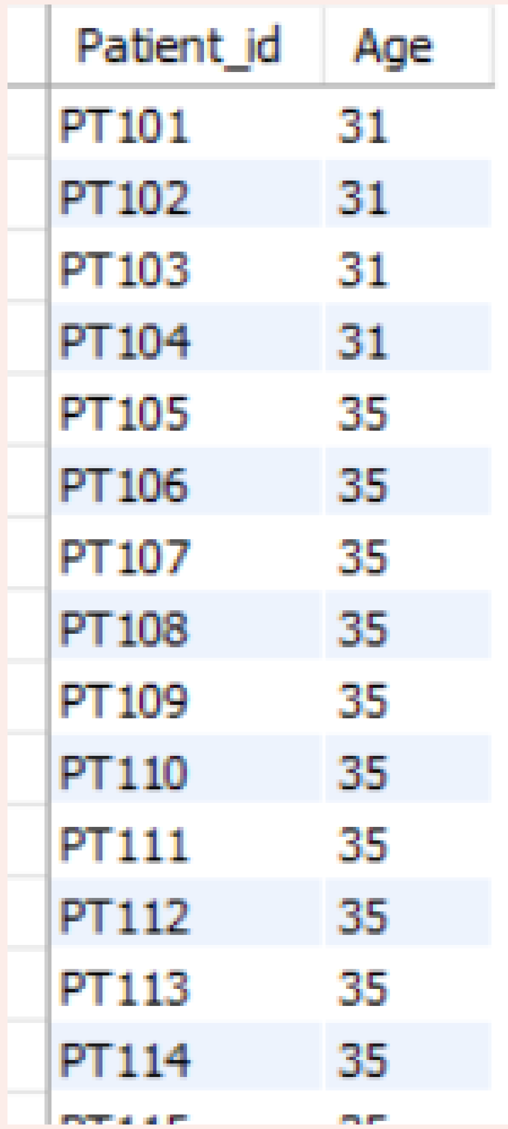
Project Questions

And SQL Queries

1. RETRIEVE THE PATIENT_ID AND AGES OF ALL PATIENTS.

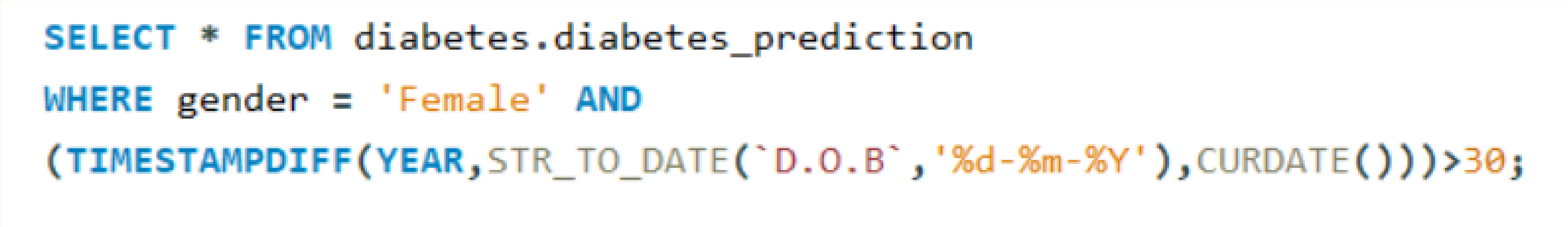


```
SELECT Patient_id,
    IFNULL(TIMESTAMPDIFF(YEAR,STR_TO_DATE(`D.O.B`, '%d-%m-%Y'),CURDATE()),0) AS Age
FROM diabetes.diabetes_prediction;
```

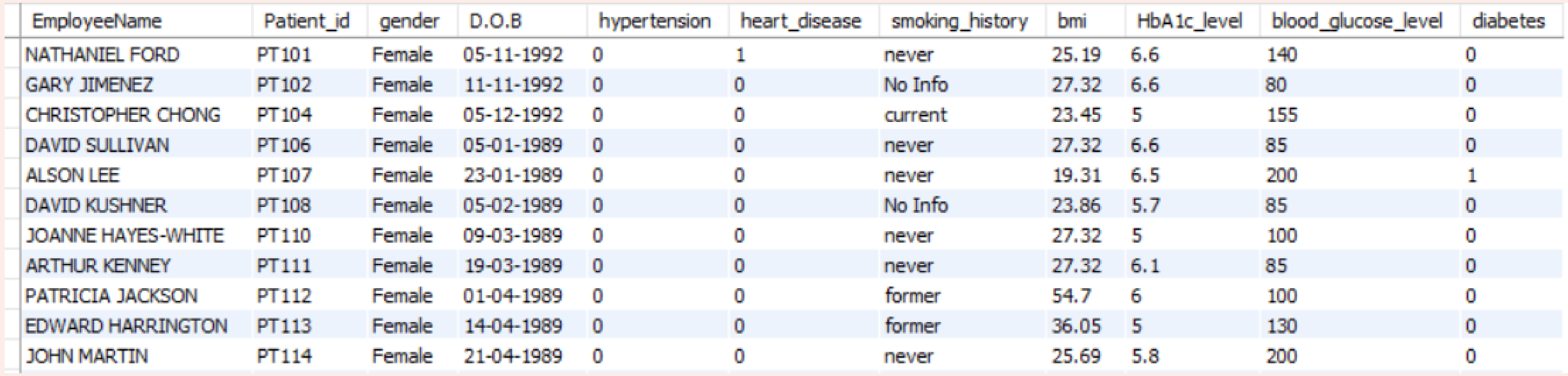


Patient_id	Age
PT101	31
PT102	31
PT103	31
PT104	31
PT105	35
PT106	35
PT107	35
PT108	35
PT109	35
PT110	35
PT111	35
PT112	35
PT113	35
PT114	35

2. SELECT ALL FEMALE PATIENTS WHO ARE OLDER THAN 30.

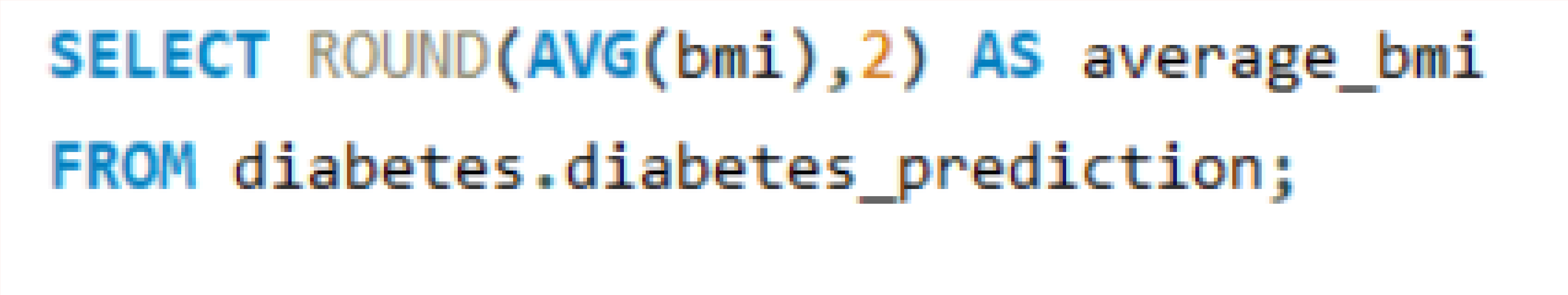


```sql
SELECT * FROM diabetes.diabetes_prediction
WHERE gender = 'Female' AND
(TIMESTAMPDIFF(YEAR,STR_TO_DATE(`D.O.B`,'%d-%m-%Y'),CURDATE()))>30;
```

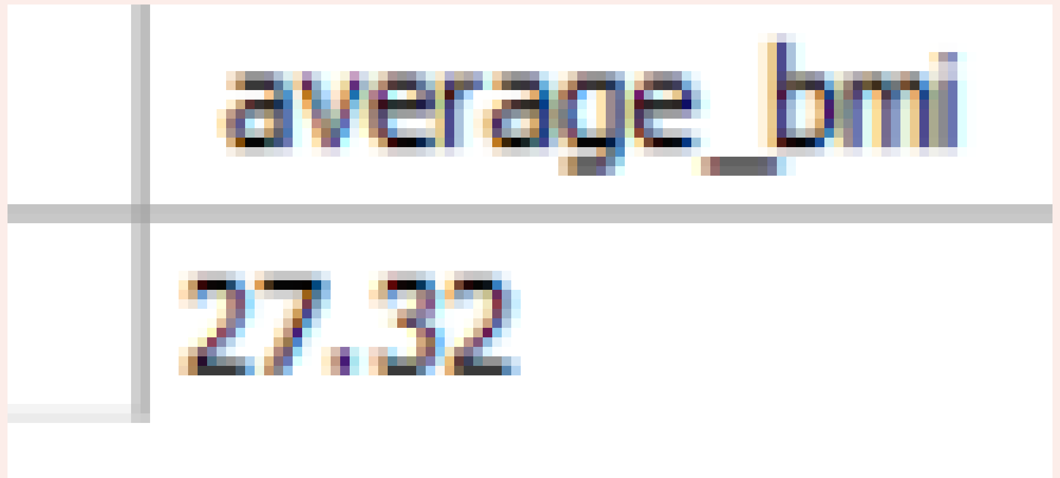


EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
NATHANIEL FORD	PT101	Female	05-11-1992	0	1	never	25.19	6.6	140	0
GARY JIMENEZ	PT102	Female	11-11-1992	0	0	No Info	27.32	6.6	80	0
CHRISTOPHER CHONG	PT104	Female	05-12-1992	0	0	current	23.45	5	155	0
DAVID SULLIVAN	PT106	Female	05-01-1989	0	0	never	27.32	6.6	85	0
ALSON LEE	PT107	Female	23-01-1989	0	0	never	19.31	6.5	200	1
DAVID KUSHNER	PT108	Female	05-02-1989	0	0	No Info	23.86	5.7	85	0
JOANNE HAYES-WHITE	PT110	Female	09-03-1989	0	0	never	27.32	5	100	0
ARTHUR KENNEY	PT111	Female	19-03-1989	0	0	never	27.32	6.1	85	0
PATRICIA JACKSON	PT112	Female	01-04-1989	0	0	former	54.7	6	100	0
EDWARD HARRINGTON	PT113	Female	14-04-1989	0	0	former	36.05	5	130	0
JOHN MARTIN	PT114	Female	21-04-1989	0	0	never	25.69	5.8	200	0

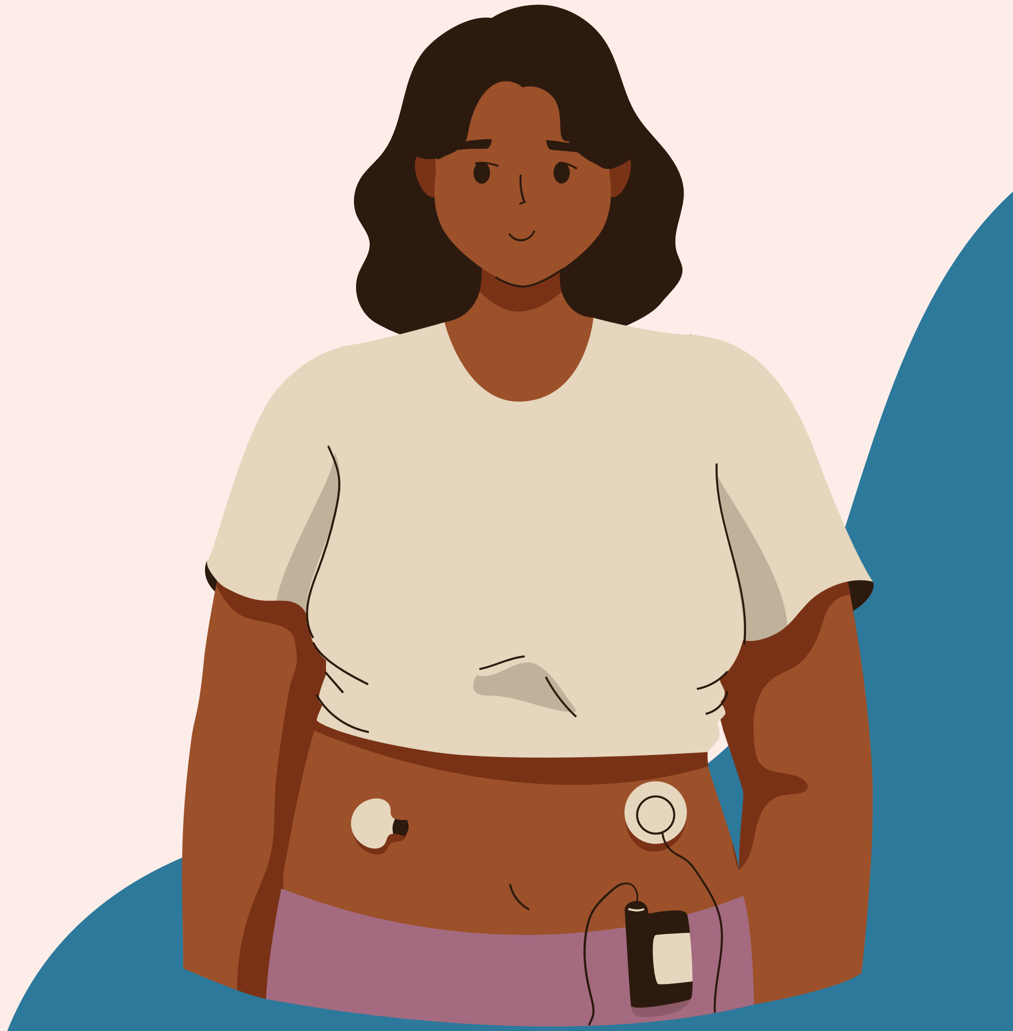
3. CALCULATE THE AVERAGE BMI OF PATIENTS.



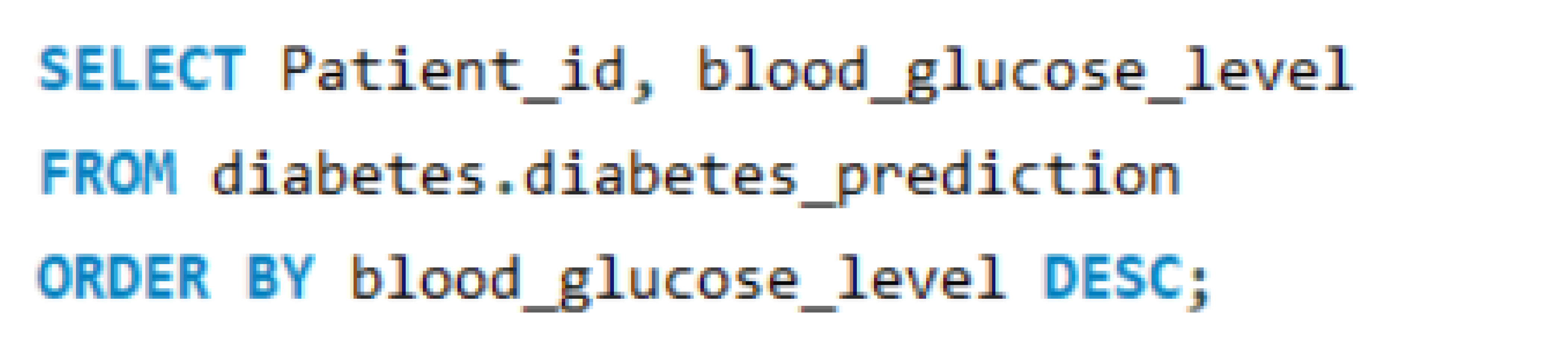
```sql
SELECT ROUND(AVG(bmi),2) AS average_bmi
FROM diabetes.diabetes_prediction;
```



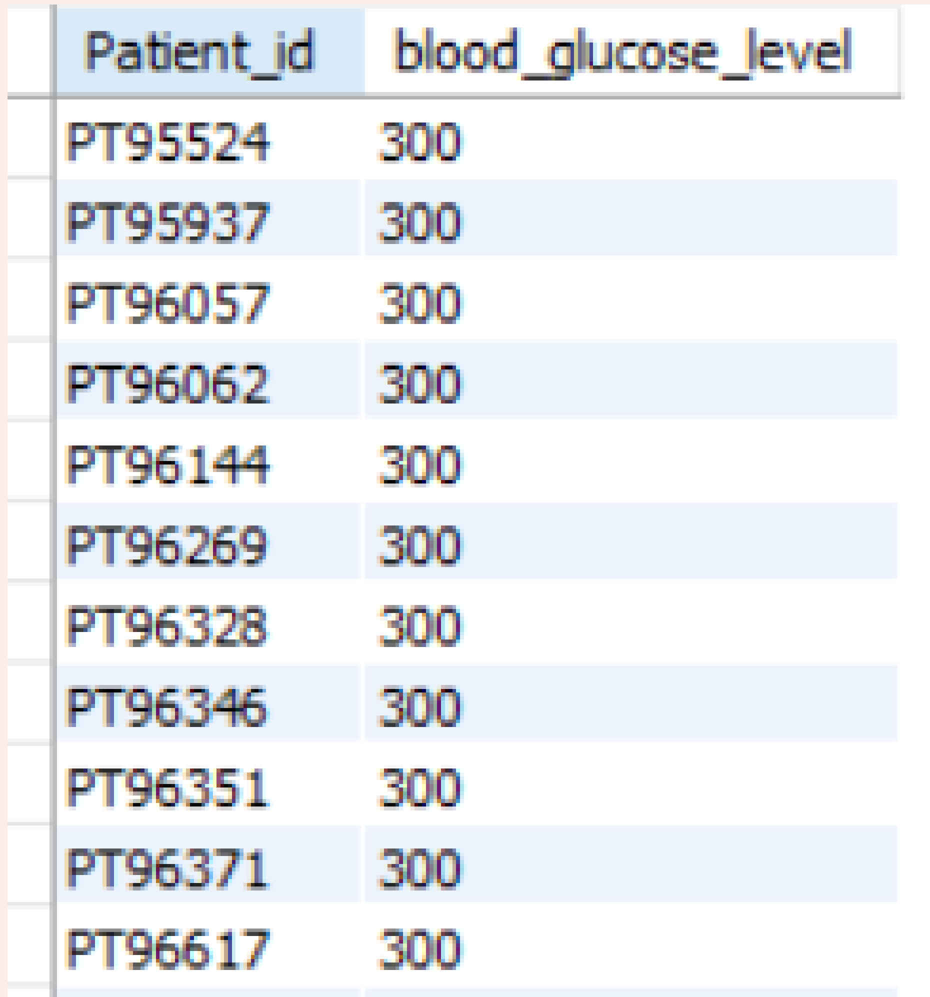
average_bmi
27.32



4. LIST PATIENTS IN DESCENDING ORDER OF BLOOD GLUCOSE LEVELS.

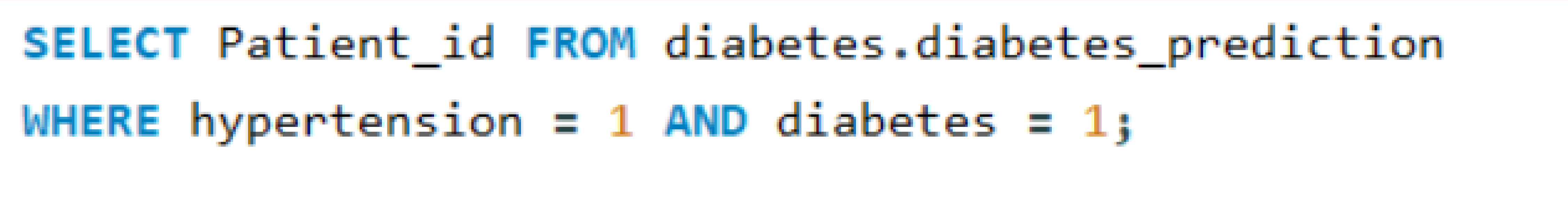


```
SELECT Patient_id, blood_glucose_level
FROM diabetes.diabetes_prediction
ORDER BY blood_glucose_level DESC;
```

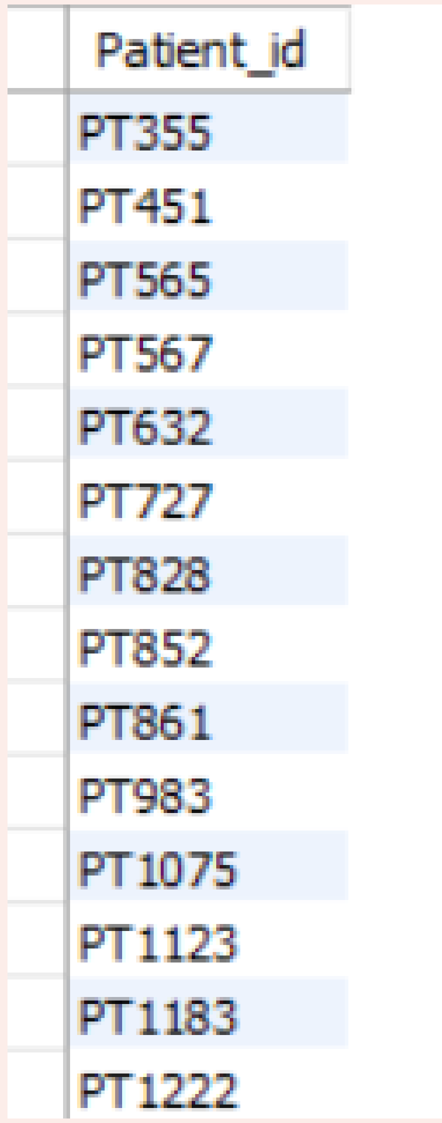


Patient_id	blood_glucose_level
PT95524	300
PT95937	300
PT96057	300
PT96062	300
PT96144	300
PT96269	300
PT96328	300
PT96346	300
PT96351	300
PT96371	300
PT96617	300

5. FIND PATIENTS WHO HAVE HYPERTENSION AND DIABETES.

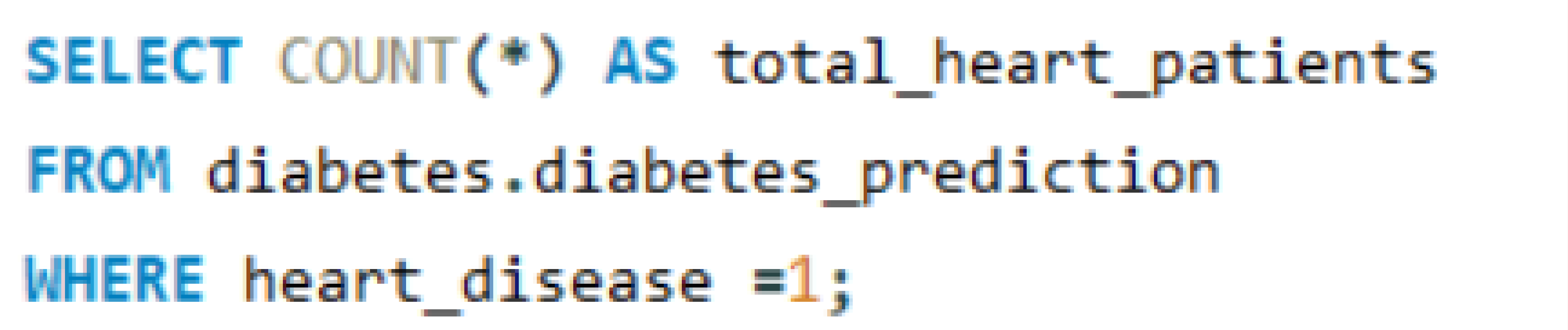


```sql
SELECT Patient_id FROM diabetes.diabetes_prediction
WHERE hypertension = 1 AND diabetes = 1;
```

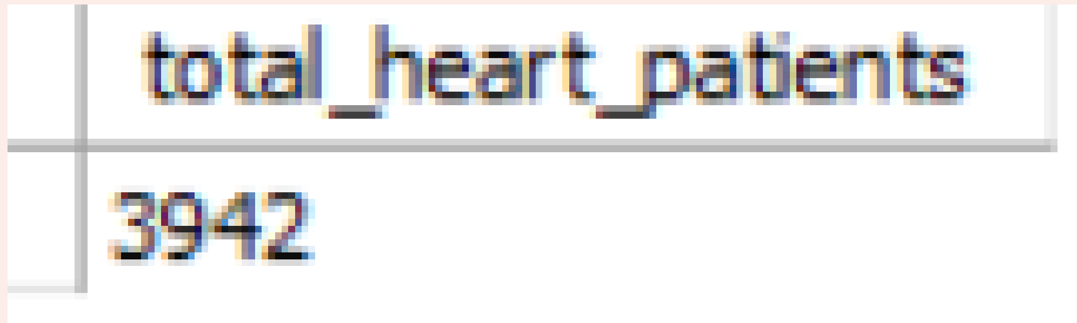


Patient_id
PT355
PT451
PT565
PT567
PT632
PT727
PT828
PT852
PT861
PT983
PT1075
PT1123
PT1183
PT1222

6. DETERMINE THE NUMBER OF PATIENTS WITH HEART DISEASE.

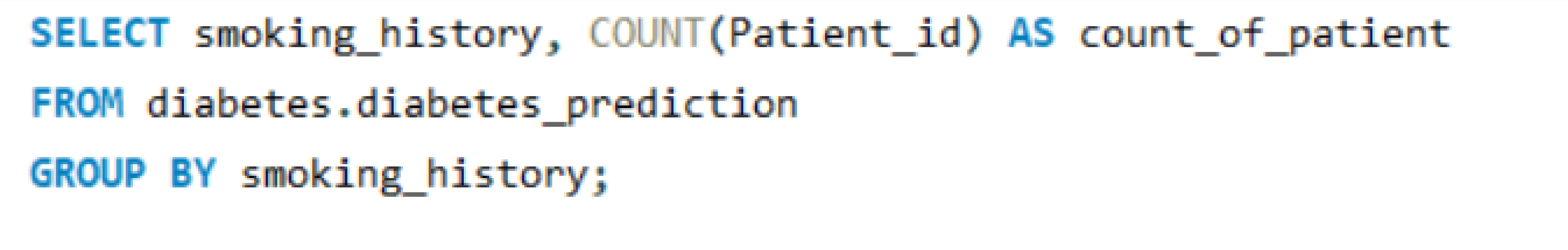


```sql
SELECT COUNT(*) AS total_heart_patients
FROM diabetes.diabetes_prediction
WHERE heart_disease =1;
```

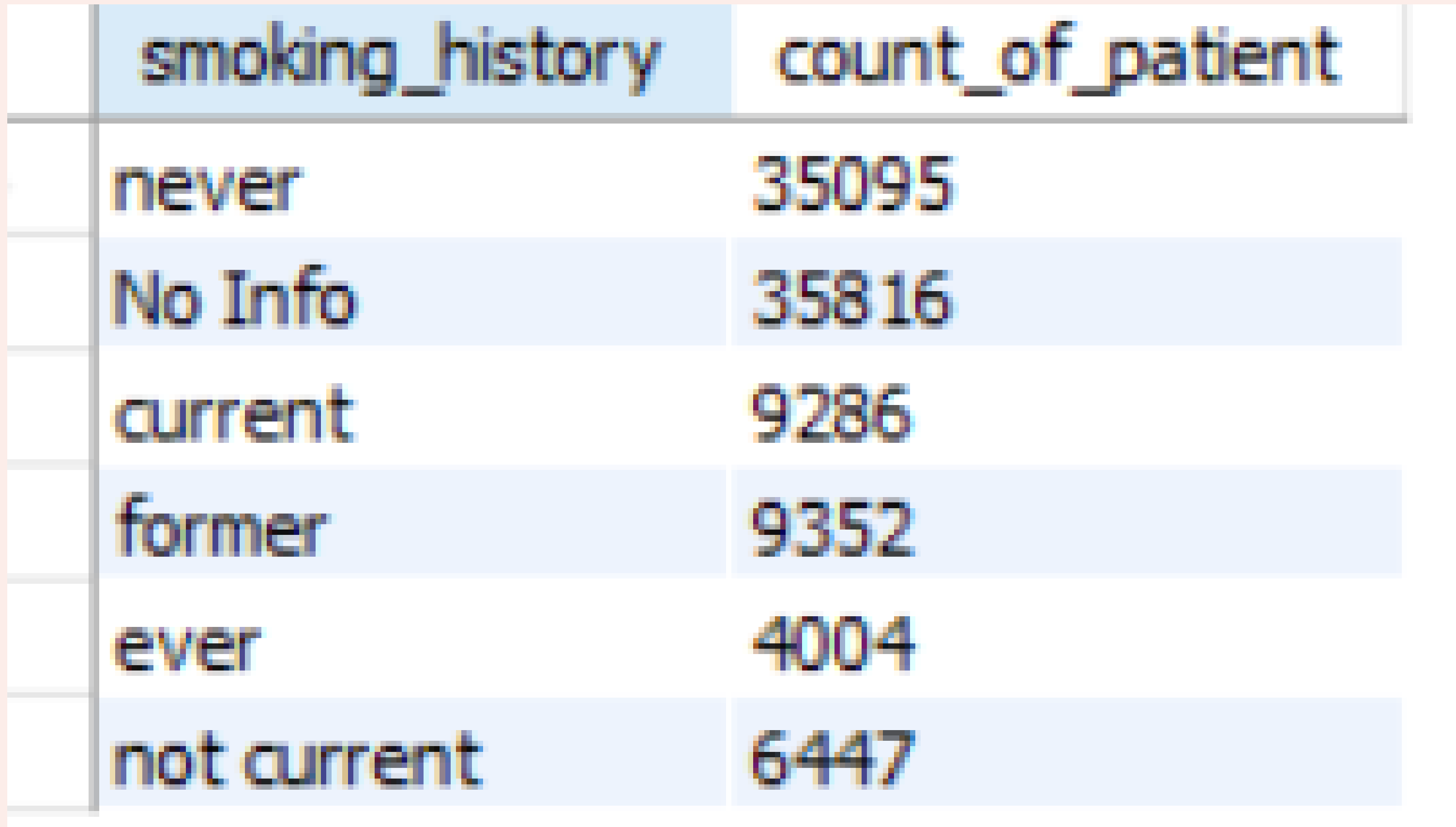


total_heart_patients
3942

7. GROUP PATIENTS BY SMOKING HISTORY AND COUNT HOW MANY SMOKERS AND NON?SMOKERS THERE ARE.

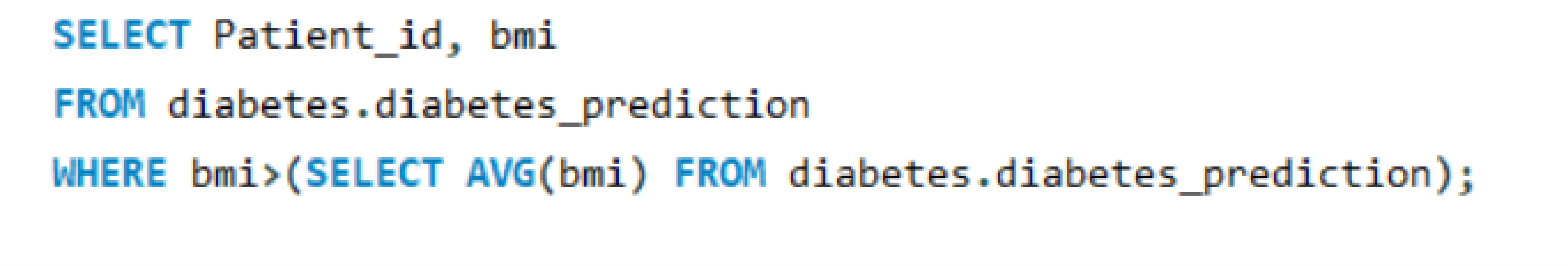


```
SELECT smoking_history, COUNT(Patient_id) AS count_of_patient
FROM diabetes.diabetes_prediction
GROUP BY smoking_history;
```

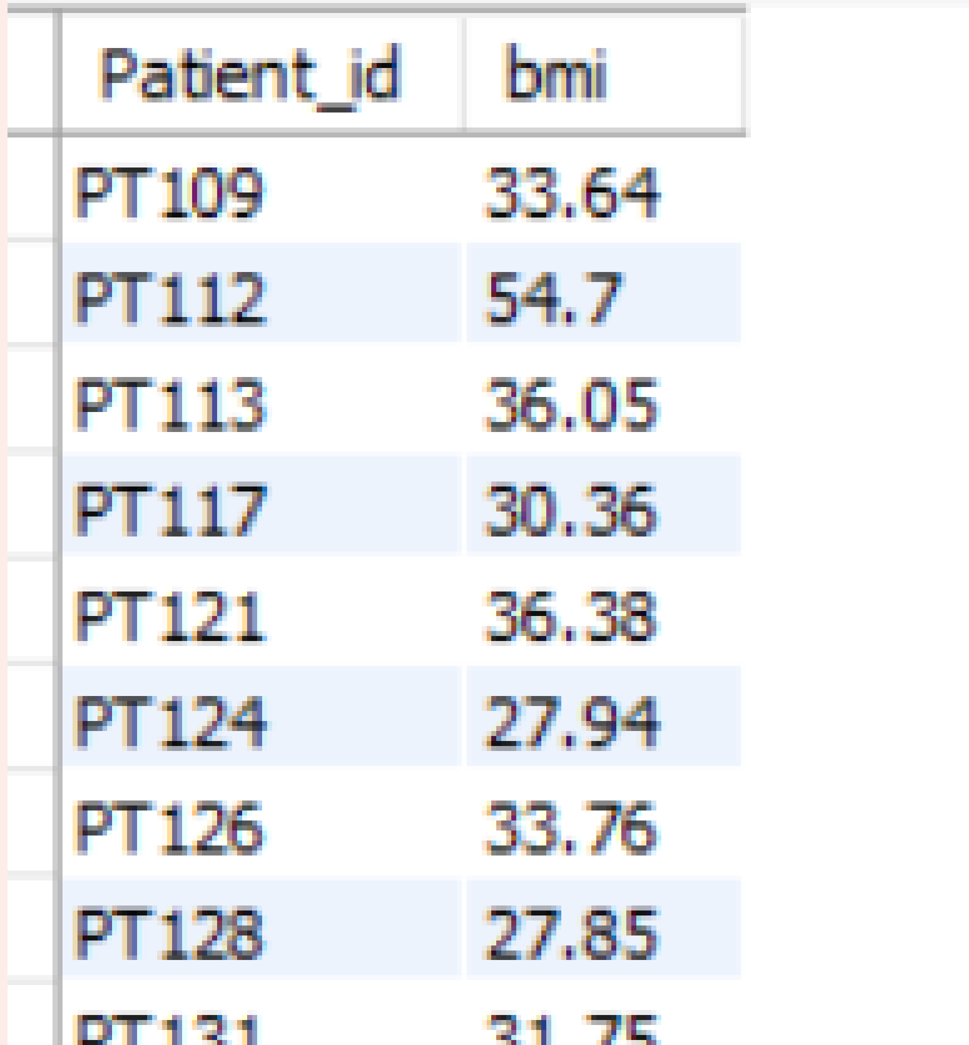


smoking_history	count_of_patient
never	35095
No Info	35816
current	9286
former	9352
ever	4004
not current	6447

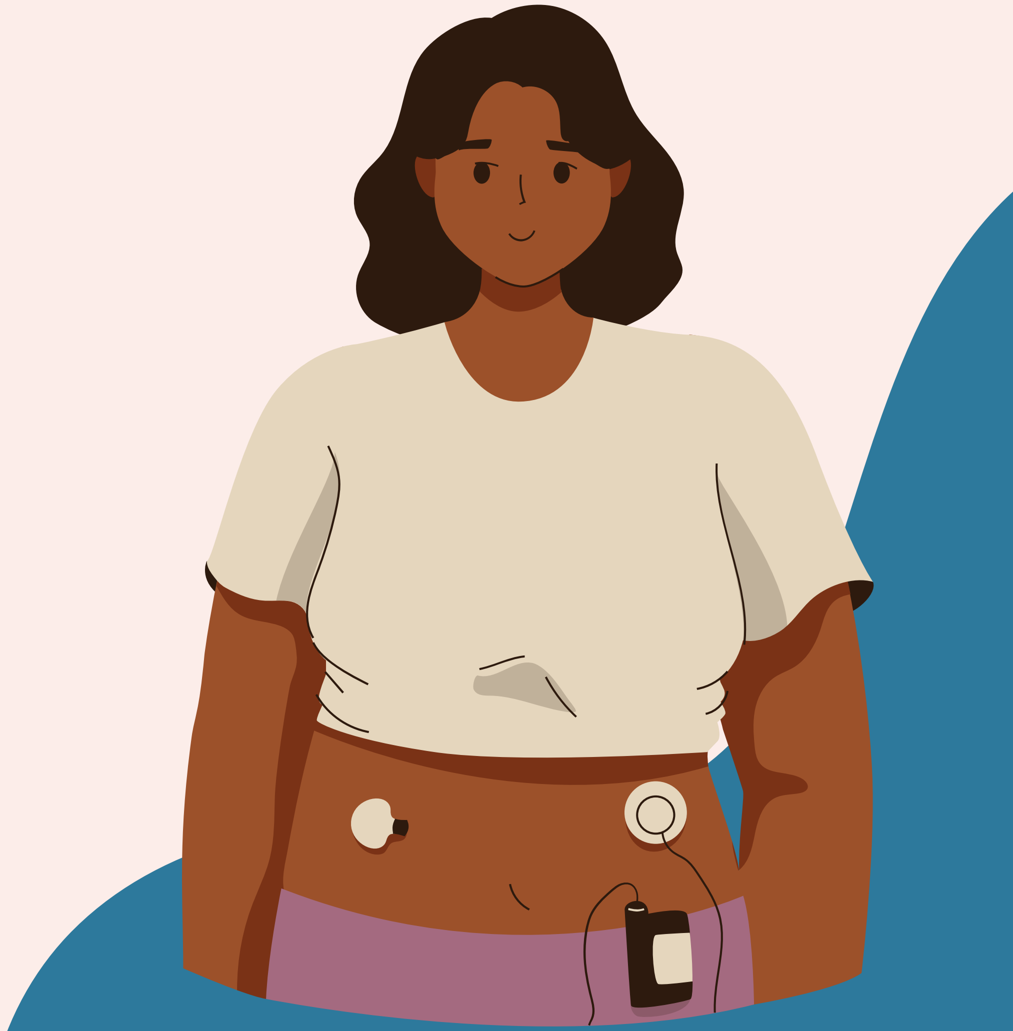
8. RETRIEVE THE PATIENT_ID OF PATIENTS WHO HAVE A BMI GREATER THAN THE AVERAGE BMI.



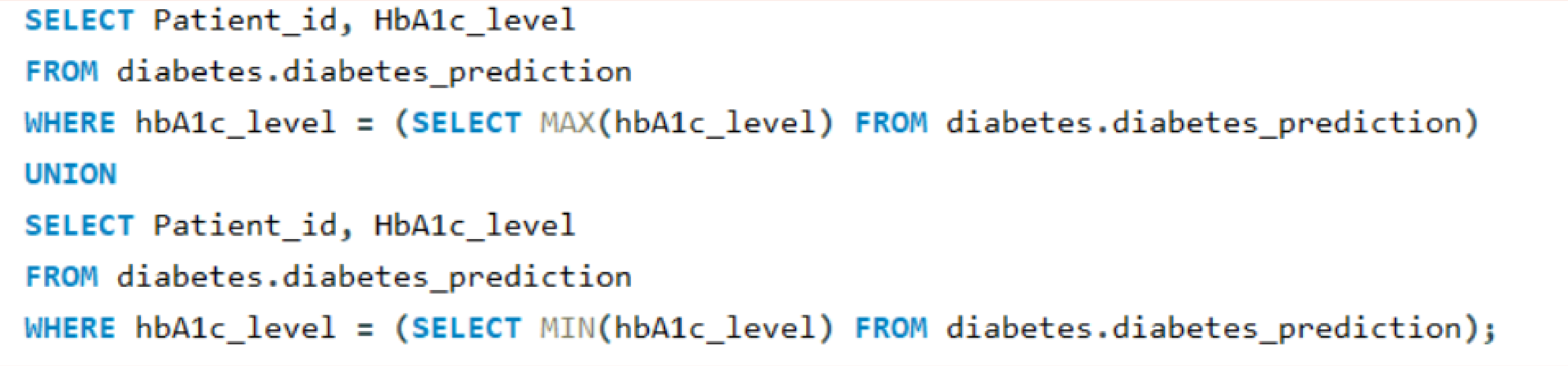
```sql
SELECT Patient_id, bmi
FROM diabetes.diabetes_prediction
WHERE bmi>(SELECT AVG(bmi) FROM diabetes.diabetes_prediction);
```



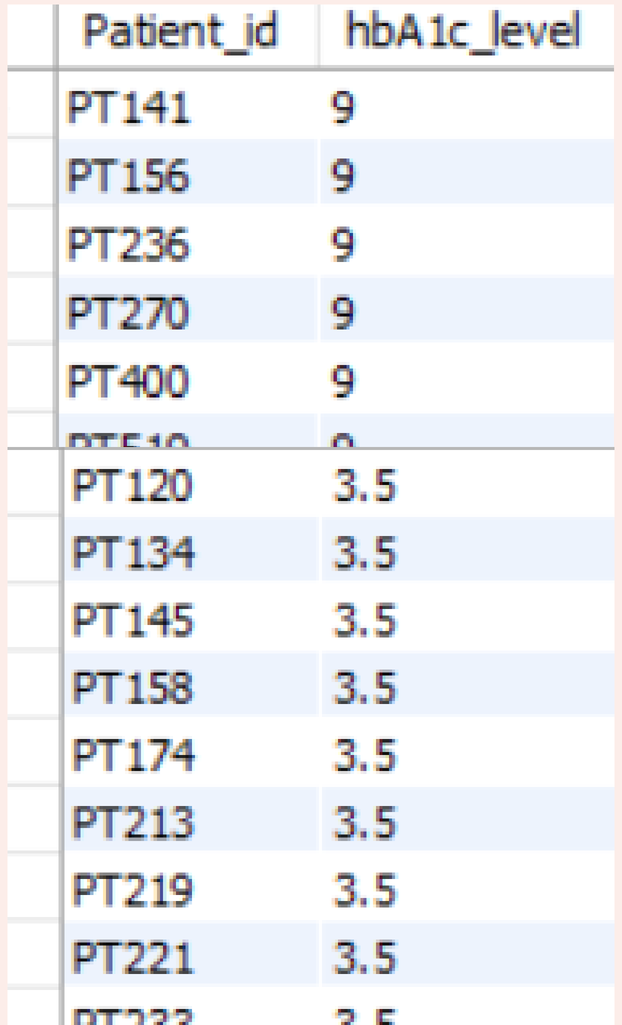
Patient_id	bmi
PT109	33.64
PT112	54.7
PT113	36.05
PT117	30.36
PT121	36.38
PT124	27.94
PT126	33.76
PT128	27.85
PT131	31.75



9. FIND THE PATIENT WITH THE HIGHEST HBA1C LEVEL AND THE PATIENT WITH THE LOWEST HBA1CLEVEL

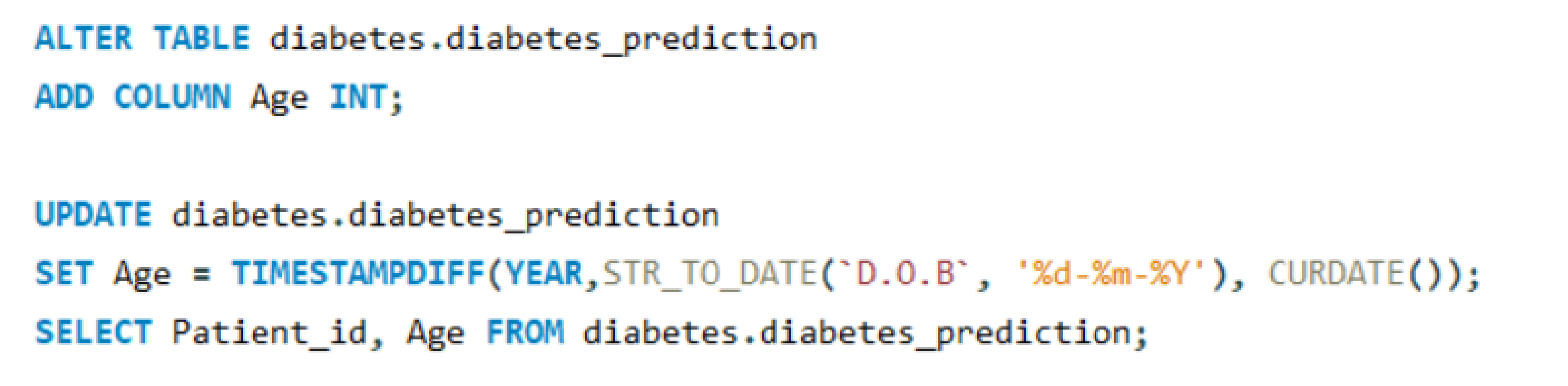


```
SELECT Patient_id, HbA1c_level
FROM diabetes.diabetes_prediction
WHERE hbA1c_level = (SELECT MAX(hbA1c_level) FROM diabetes.diabetes_prediction)
UNION
SELECT Patient_id, HbA1c_level
FROM diabetes.diabetes_prediction
WHERE hbA1c_level = (SELECT MIN(hbA1c_level) FROM diabetes.diabetes_prediction);
```

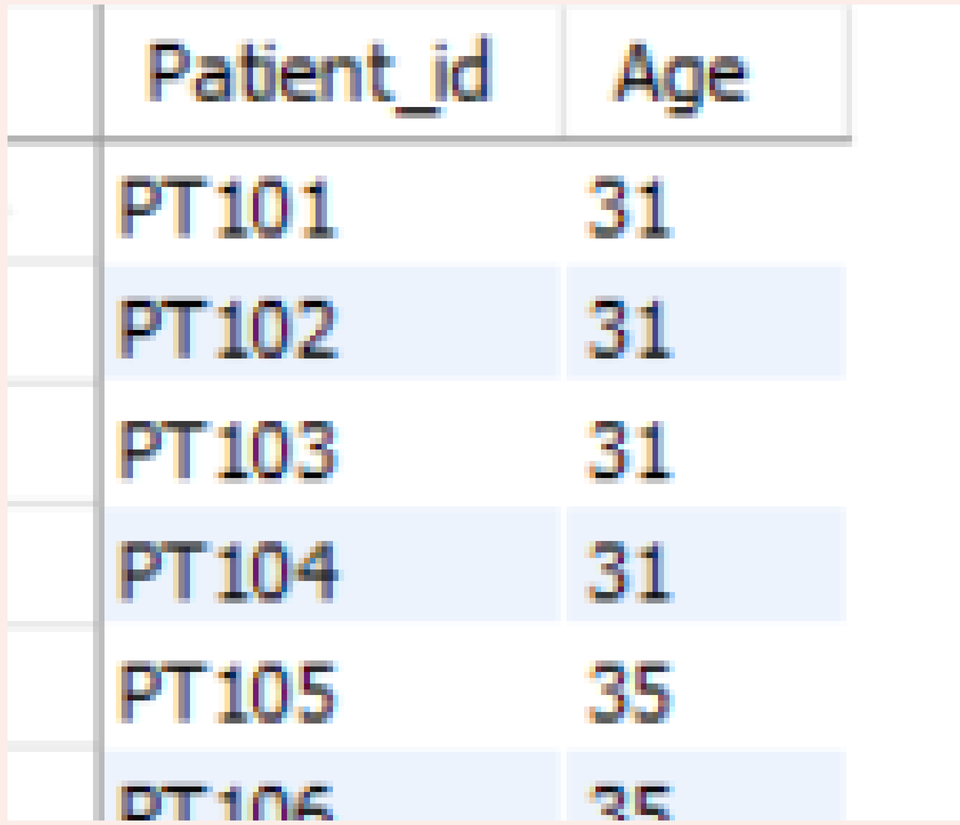


Patient_id	hbA1c_level
PT141	9
PT156	9
PT236	9
PT270	9
PT400	9
PT510	9
PT120	3.5
PT134	3.5
PT145	3.5
PT158	3.5
PT174	3.5
PT213	3.5
PT219	3.5
PT221	3.5
PT233	3.5

10. CALCULATE THE AGE OF PATIENTS IN YEARS (ASSUMING THE CURRENT DATE AS OF NOW).

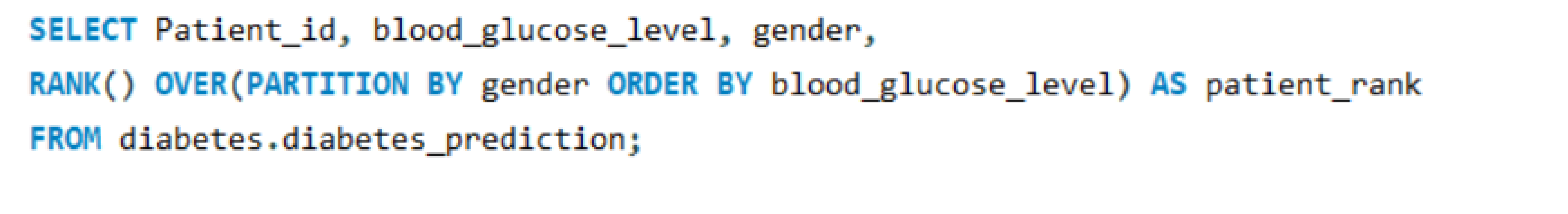


```
ALTER TABLE diabetes.diabetes_prediction
ADD COLUMN Age INT;

UPDATE diabetes.diabetes_prediction
SET Age = TIMESTAMPDIFF(YEAR,STR_TO_DATE(`D.O.B`, '%d-%m-%Y'), CURDATE());
SELECT Patient_id, Age FROM diabetes.diabetes_prediction;
```

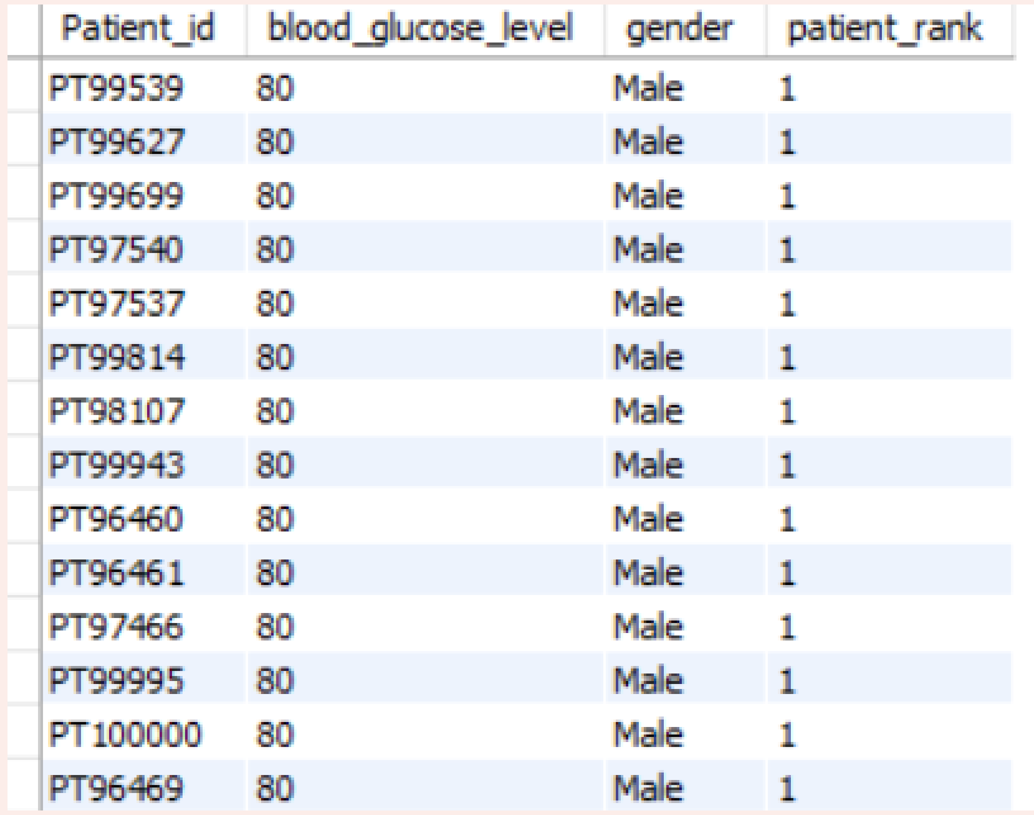


Patient_id	Age
PT101	31
PT102	31
PT103	31
PT104	31
PT105	35
PT106	35

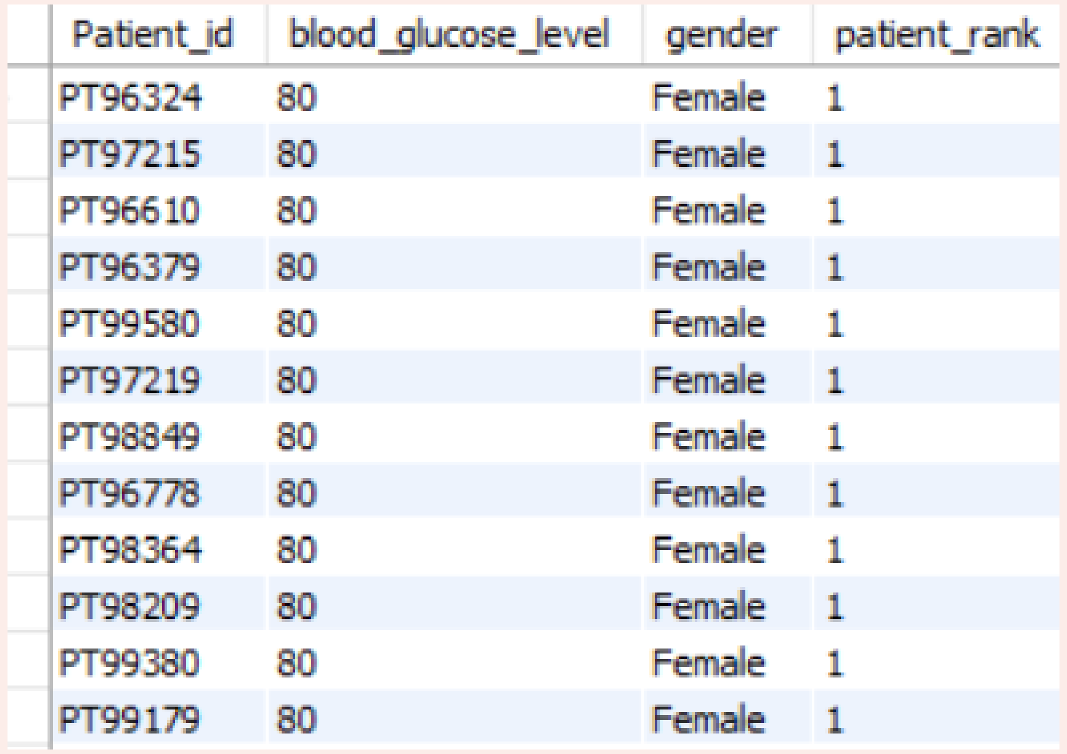
11. RANK PATIENTS BY BLOOD GLUCOSE LEVEL WITHIN EACH GENDER GROUP.



```
SELECT Patient_id, blood_glucose_level, gender,
RANK() OVER(PARTITION BY gender ORDER BY blood_glucose_level) AS patient_rank
FROM diabetes.diabetes_prediction;
```

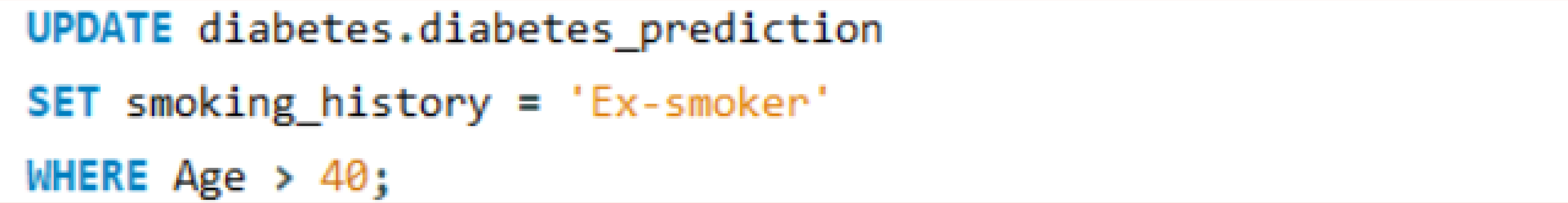


Patient_id	blood_glucose_level	gender	patient_rank
PT99539	80	Male	1
PT99627	80	Male	1
PT99699	80	Male	1
PT97540	80	Male	1
PT97537	80	Male	1
PT99814	80	Male	1
PT98107	80	Male	1
PT99943	80	Male	1
PT96460	80	Male	1
PT96461	80	Male	1
PT97466	80	Male	1
PT99995	80	Male	1
PT100000	80	Male	1
PT96469	80	Male	1

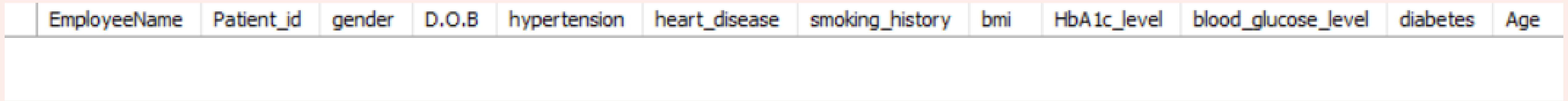


Patient_id	blood_glucose_level	gender	patient_rank
PT96324	80	Female	1
PT97215	80	Female	1
PT96610	80	Female	1
PT96379	80	Female	1
PT99580	80	Female	1
PT97219	80	Female	1
PT98849	80	Female	1
PT96778	80	Female	1
PT98364	80	Female	1
PT98209	80	Female	1
PT99380	80	Female	1
PT99179	80	Female	1

12. UPDATE THE SMOKING HISTORY OF PATIENTS WHO ARE OLDERTHAN 40 TO "EX-SMOKER."



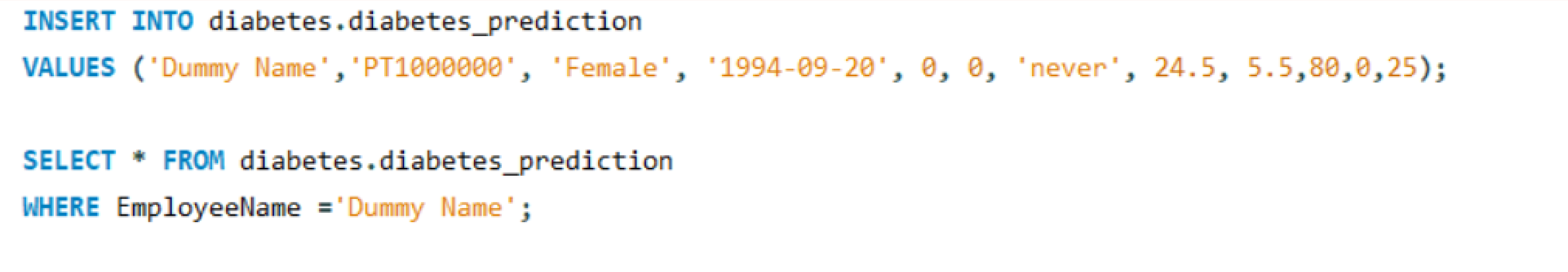
```sql
UPDATE diabetes.diabetes_prediction
SET smoking_history = 'Ex-smoker'
WHERE Age > 40;
```



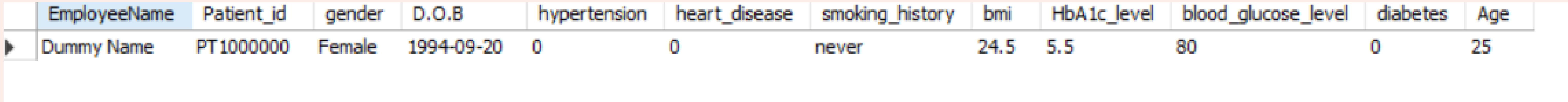
	EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	Age

There are no patients in the dataset where the age is greater than 40, hence the output is a blank table.

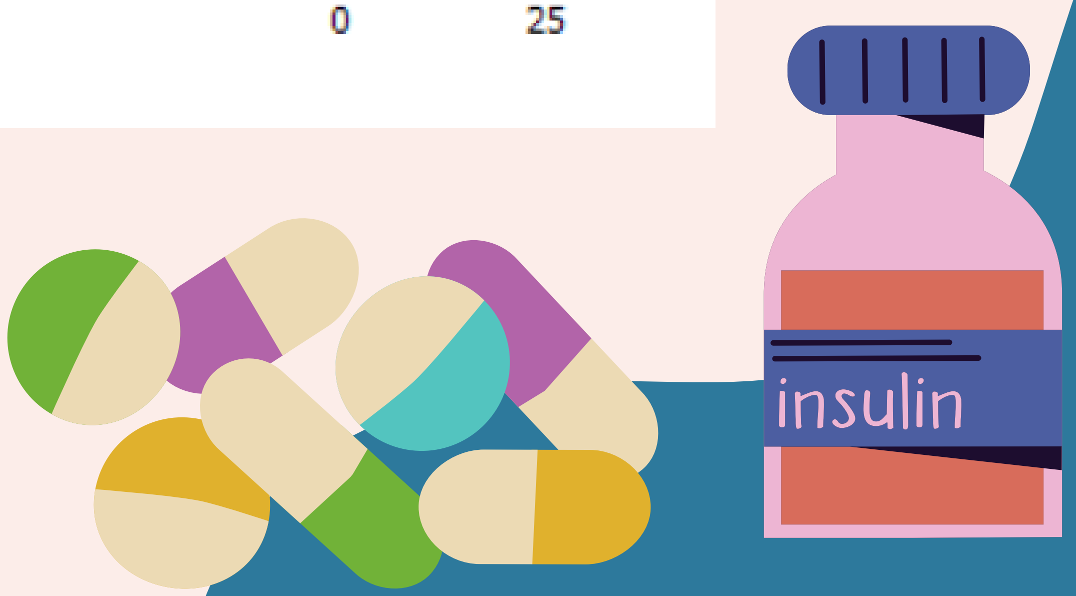
13. INSERT A NEW PATIENT INTO THE DATABASE WITH SAMPLE DATA.



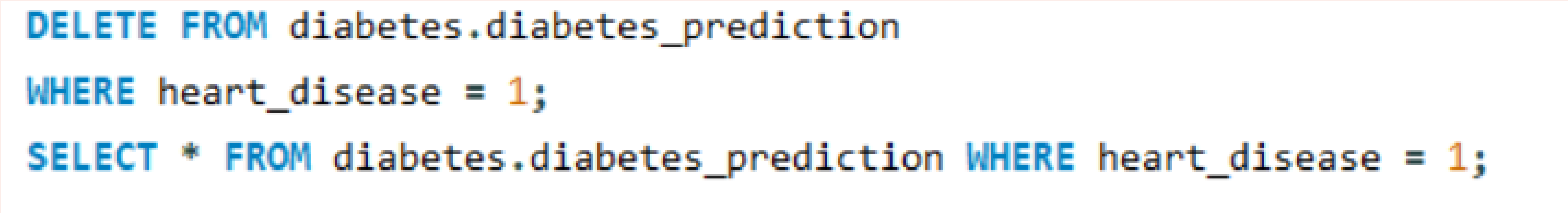
```sql
INSERT INTO diabetes.diabetes_prediction
VALUES ('Dummy Name','PT1000000', 'Female', '1994-09-20', 0, 0, 'never', 24.5, 5.5,80,0,25);

SELECT * FROM diabetes.diabetes_prediction
WHERE EmployeeName ='Dummy Name';
```



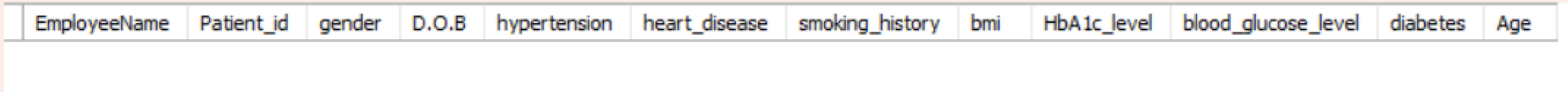
EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	Age
Dummy Name	PT1000000	Female	1994-09-20	0	0	never	24.5	5.5	80	0	25



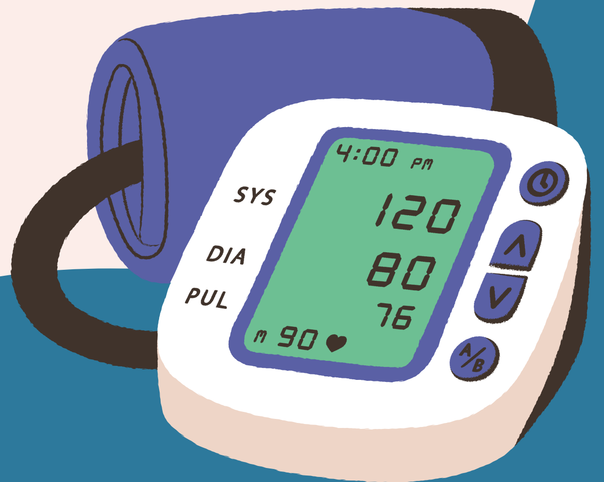
14. DELETE ALL PATIENTS WITH HEART DISEASE FROM THE DATABASE.



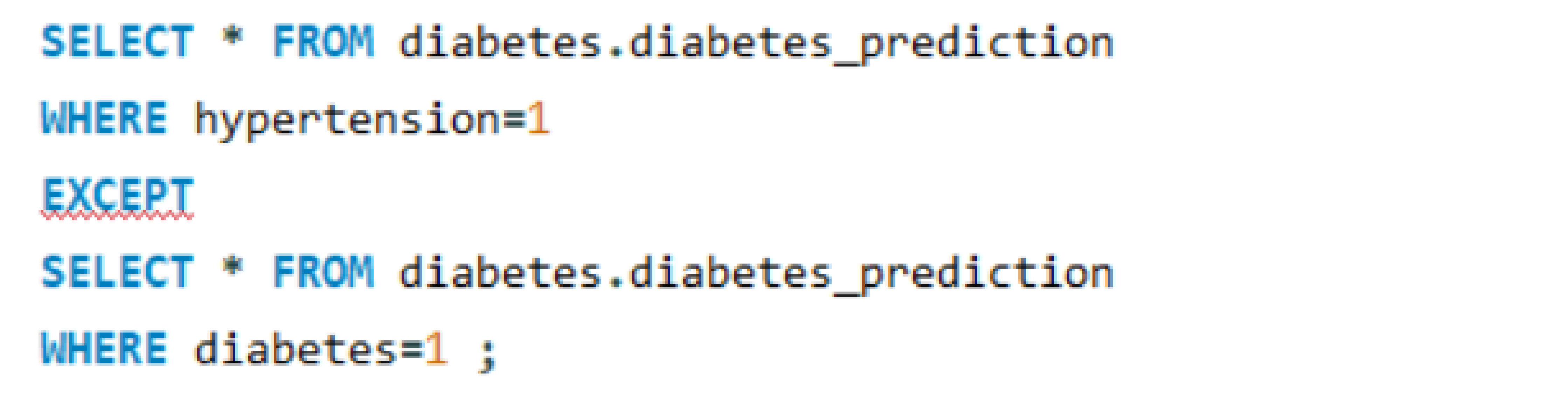
```sql
DELETE FROM diabetes.diabetes_prediction
WHERE heart_disease = 1;
SELECT * FROM diabetes.diabetes_prediction WHERE heart_disease = 1;
```



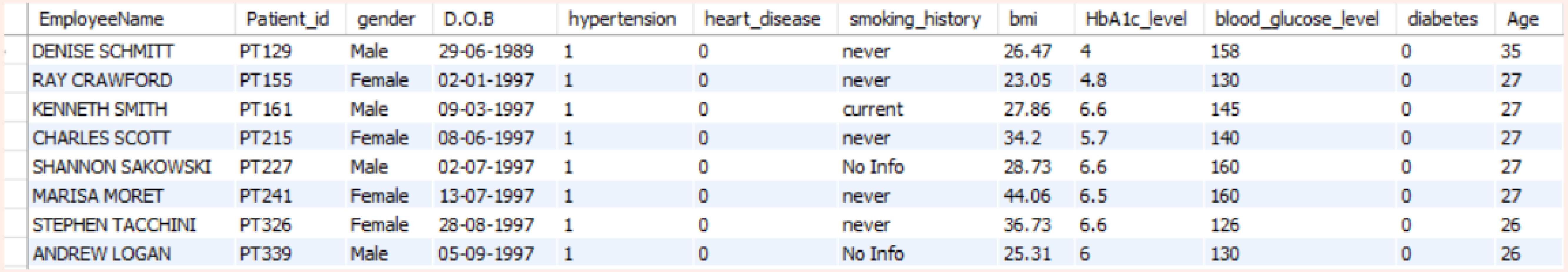
EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	Age



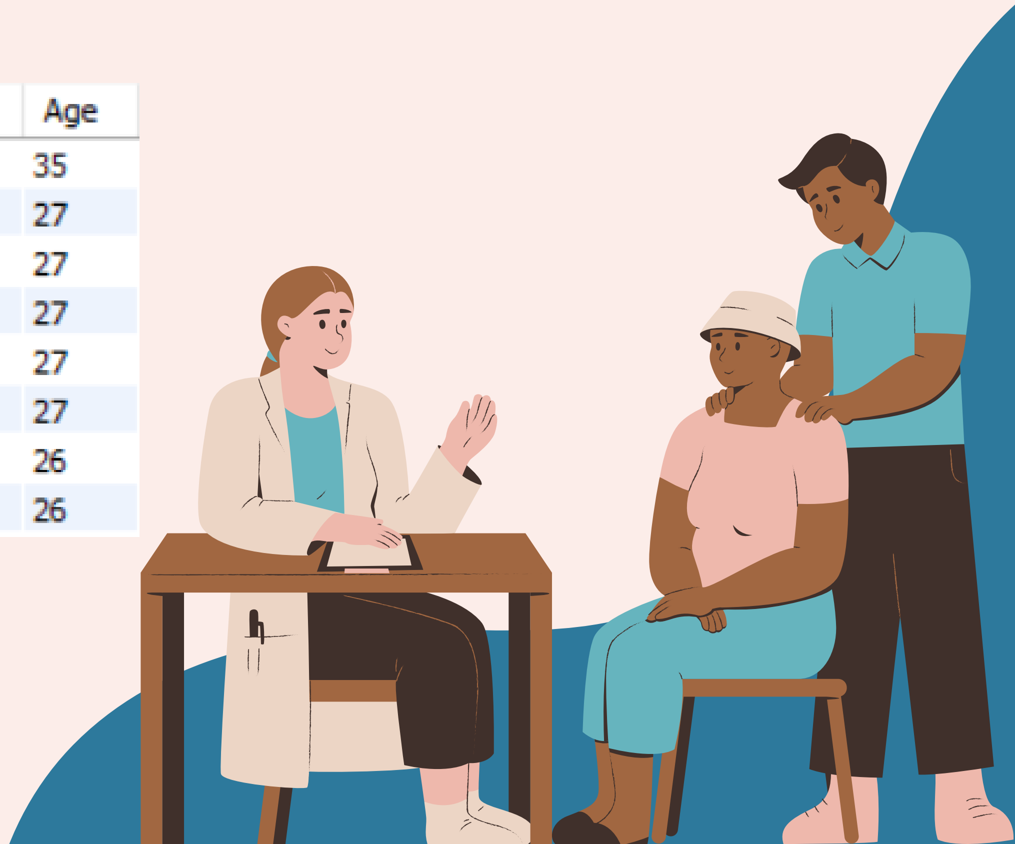
15. FIND PATIENTS WHO HAVE HYPERTENSION BUT NOT DIABETES USING THE EXCEPT OPERATOR.



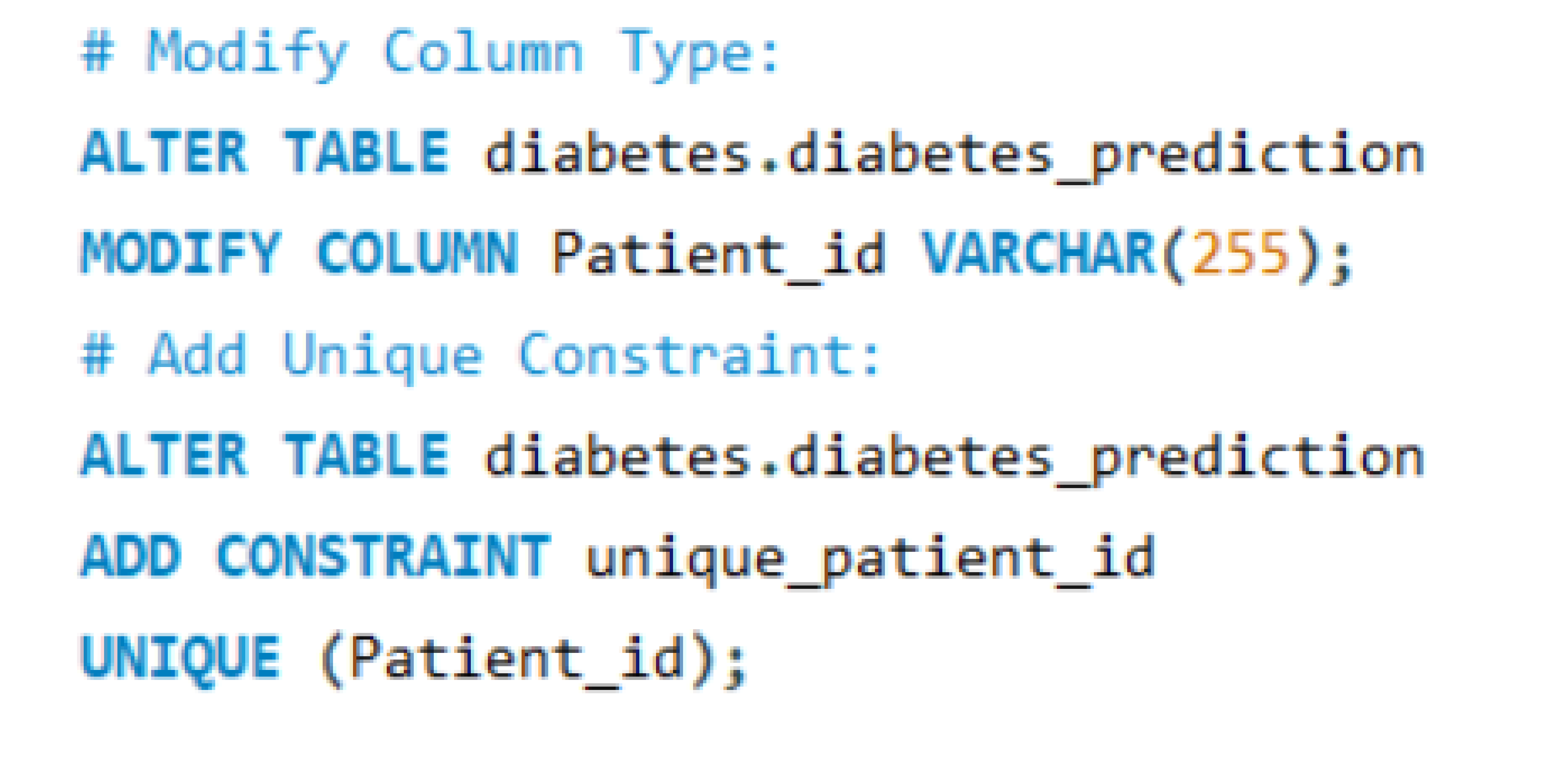
```
SELECT * FROM diabetes.diabetes_prediction
WHERE hypertension=1
EXCEPT
SELECT * FROM diabetes.diabetes_prediction
WHERE diabetes=1 ;
```



EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	Age
DENISE SCHMITT	PT129	Male	29-06-1989	1	0	never	26.47	4	158	0	35
RAY CRAWFORD	PT155	Female	02-01-1997	1	0	never	23.05	4.8	130	0	27
KENNETH SMITH	PT161	Male	09-03-1997	1	0	current	27.86	6.6	145	0	27
CHARLES SCOTT	PT215	Female	08-06-1997	1	0	never	34.2	5.7	140	0	27
SHANNON SAKOWSKI	PT227	Male	02-07-1997	1	0	No Info	28.73	6.6	160	0	27
MARISA MORET	PT241	Female	13-07-1997	1	0	never	44.06	6.5	160	0	27
STEPHEN TACCHINI	PT326	Female	28-08-1997	1	0	never	36.73	6.6	126	0	26
ANDREW LOGAN	PT339	Male	05-09-1997	1	0	No Info	25.31	6	130	0	26

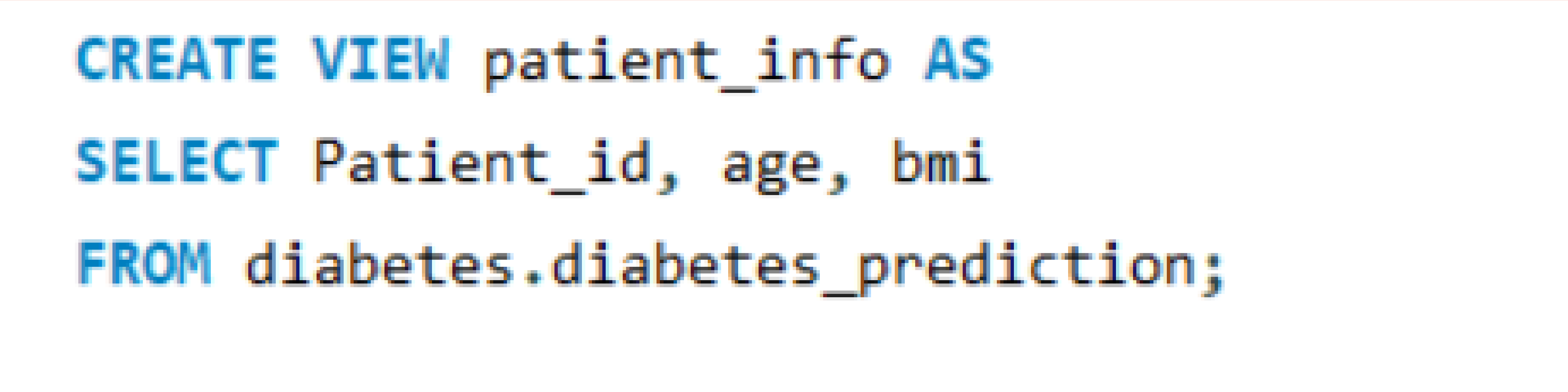


16. DEFINE A UNIQUE CONSTRAINT ON THE "PATIENT_ID" COLUMN TO ENSURE ITS VALUES ARE UNIQUE.

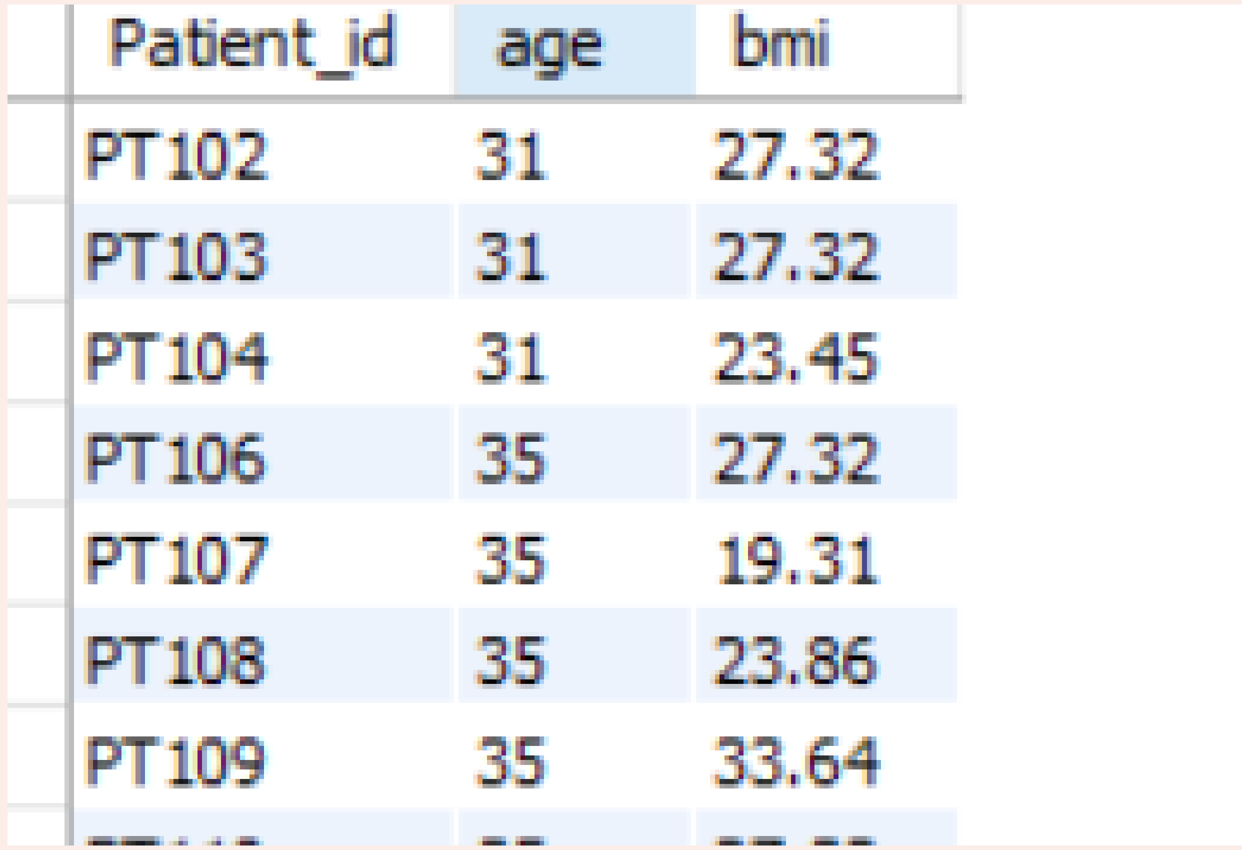


```
# Modify Column Type:
ALTER TABLE diabetes.diabetes_prediction
MODIFY COLUMN Patient_id VARCHAR(255);
# Add Unique Constraint:
ALTER TABLE diabetes.diabetes_prediction
ADD CONSTRAINT unique_patient_id
UNIQUE (Patient_id);
```

17. CREATE A VIEW THAT DISPLAYS THE PATIENT_IDS, AGES, AND BMI OF PATIENTS.



```sql
CREATE VIEW patient_info AS
SELECT Patient_id, age, bmi
FROM diabetes.diabetes_prediction;
```



Patient_id	age	bmi
PT102	31	27.32
PT103	31	27.32
PT104	31	23.45
PT106	35	27.32
PT107	35	19.31
PT108	35	23.86
PT109	35	33.64

18. SUGGEST IMPROVEMENTS IN THE DATABASE SCHEMA TO REDUCE DATA REDUNDANCY AND IMPROVE DATA INTEGRITY

1. **Normalize the Schema:** Create a separate table to Store patient-specific details such as Patient id, Employee Name, gender, and Date of Birth. This helps to centralize patient information and avoid duplication and a separate table to store health-related attributes such as hypertension, heart disease, smoking history, bmi, HbA1c level, blood glucose level, and diabetes. This separates health information from general patient information, making it easier to manage and update.
2. **Data Type Considerations:** Consider creating a fixed-length character field (e.g., CHAR(1)) to store gender, which ensures consistency and reduces storage requirements. Use the date data type field for the date of Birth column. Additionally, Use decimal types to store precise numerical values with a fixed number of decimal places For columns like (bmi,HbA1c_level).
3. **Data Integrity:** Ensure that each table has a unique identifier (e.g., Patient id as a primary key) to enforce uniqueness and enable accurate referencing between tables and Implement foreign key relationships to maintain data consistency and integrity across related tables, linking health details to specific patients.

19. EXPLAIN HOW YOU CAN OPTIMIZE THE PERFORMANCE OF SQL QUERIES ON THIS DATASET.

Database Schema Design: Normalize tables to minimize redundancy and use appropriate data types to ensure efficient storage and querying.

Indexing: Create indexes on columns frequently used in WHERE clauses, JOIN conditions, or ORDER BY clauses to speed up data retrieval.

Optimize Queries: Select only the necessary columns (SELECT specific fields) and filter data as early as possible in the query to reduce the amount of data processed. Apply the LIMIT clause to restrict the number of rows returned, which can enhance query performance and manage large result sets.

Query Execution Plan: Use tools like EXPLAIN to analyze query execution plans and identify bottlenecks or inefficiencies in your queries.

THANK YOU!

Divya Pardeshi

- Data Analyst Intern at Psyliq