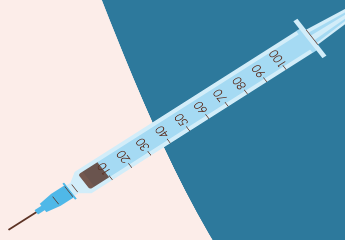
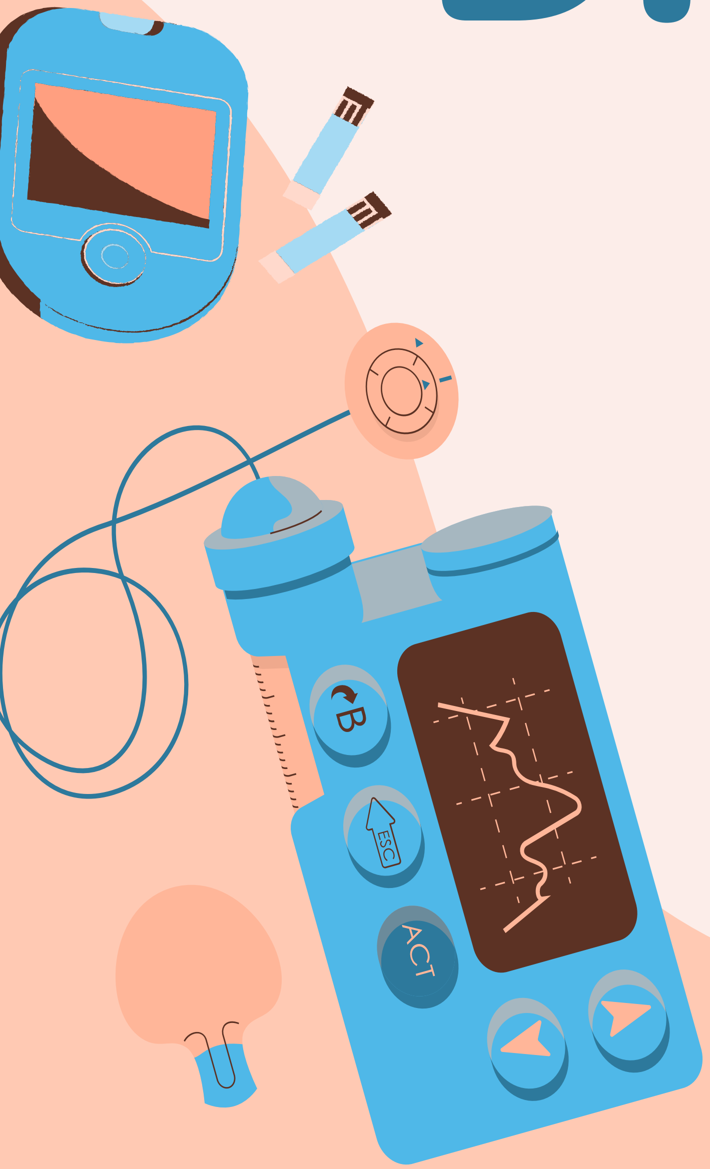
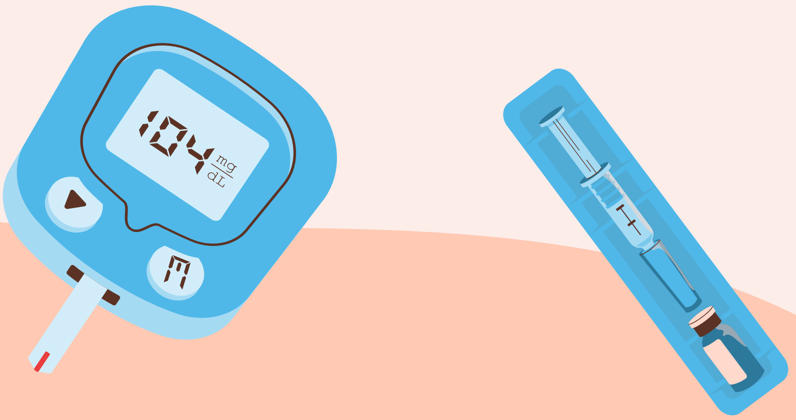


DIABETES PREDICTION ANALYSIS



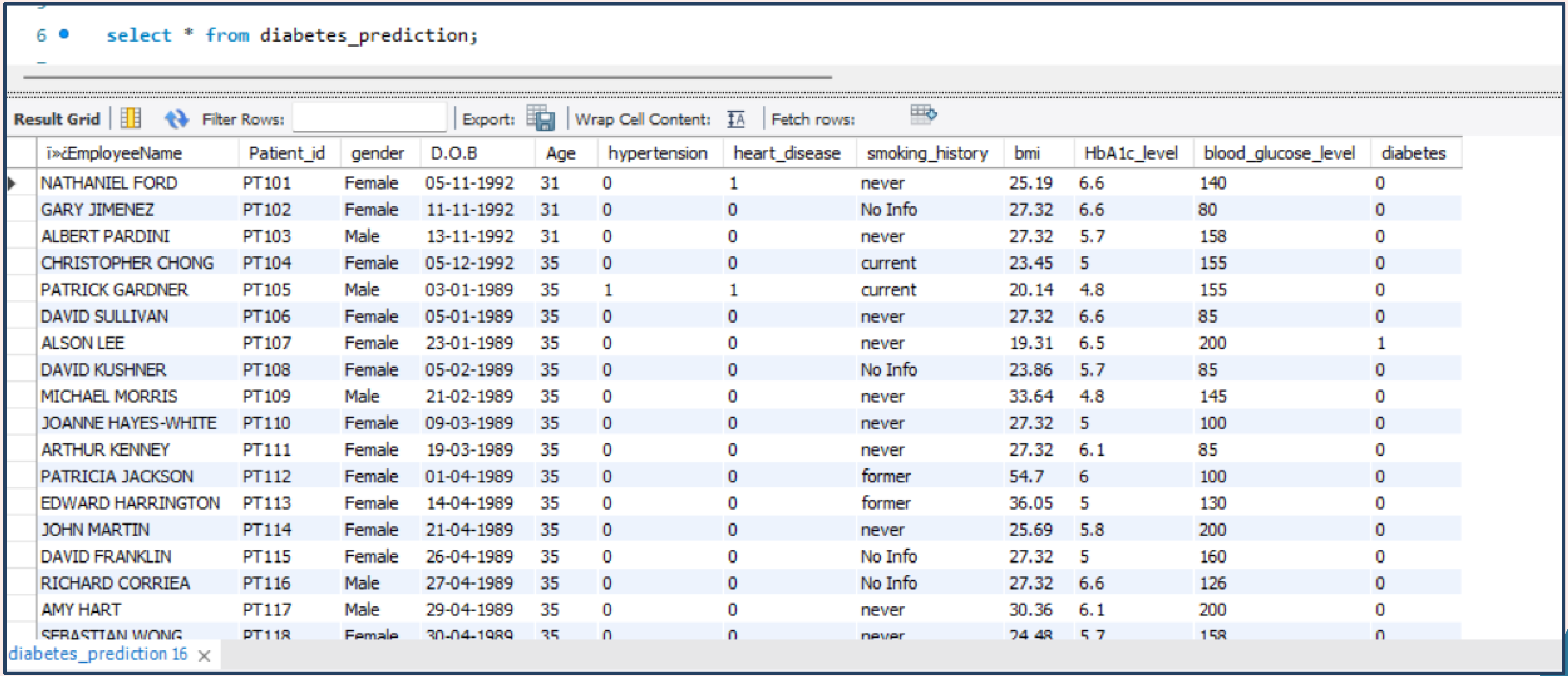


Presented by: Nitesh Kumar Prajapati

OBJECTIVE

The objective of this project is to analyze a dataset containing various demographic, clinical, and lifestyle factors of individuals, to predict the likelihood of diabetes onset. Through comprehensive data analysis and predictive modeling, the project seeks to identify key factors that contribute to diabetes risk and develop an accurate predictive model. The ultimate goal is to provide valuable insights for early detection, intervention, and management of diabetes, thereby contributing to improved healthcare outcomes and quality of life for individuals at risk of diabetes.

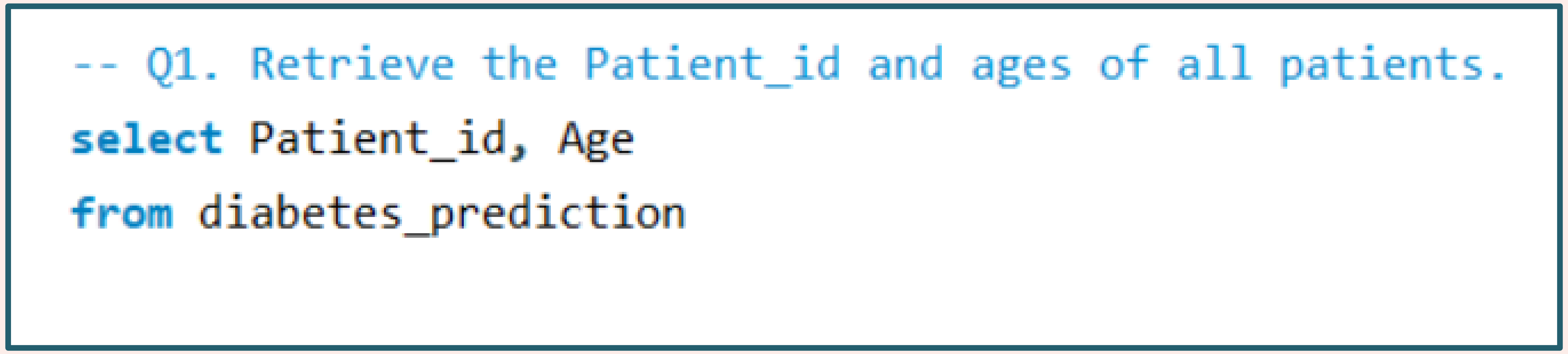
This is the Data Table Preview



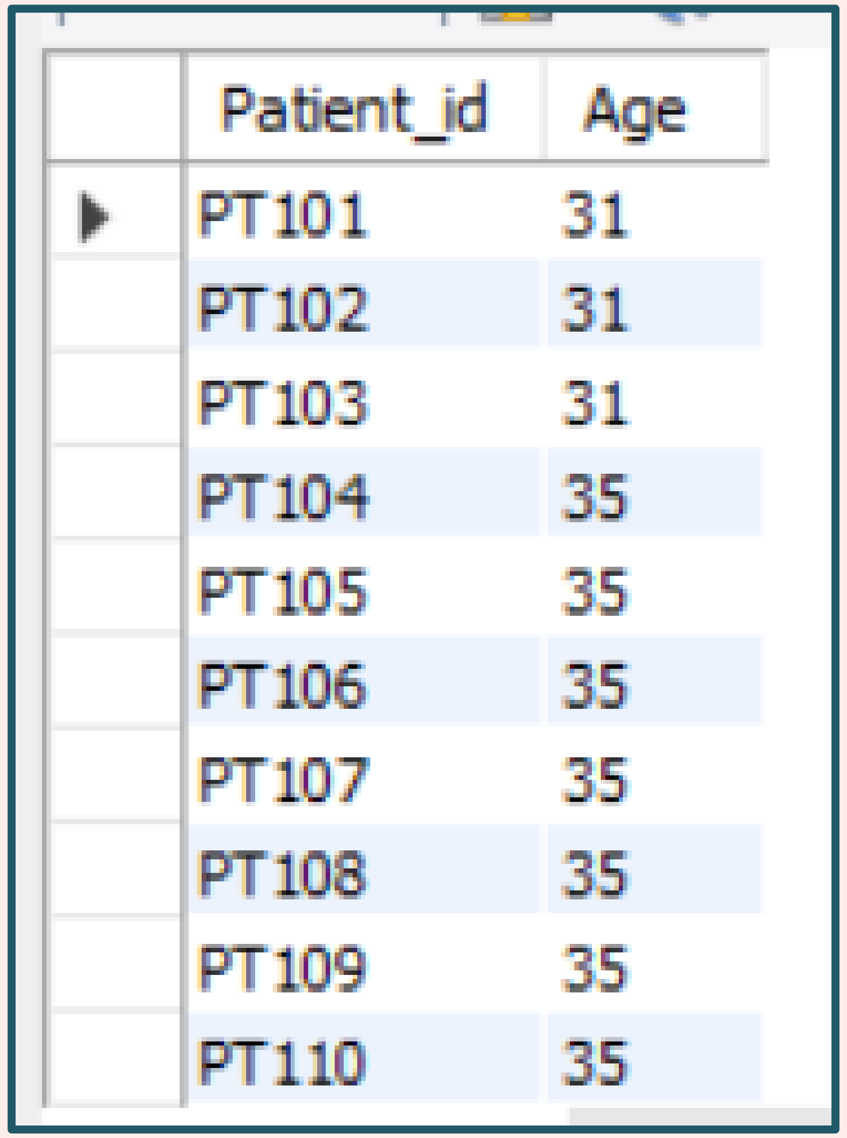
```sql
select * from diabetes_prediction;
```

ï»¿EmployeeName	Patient_id	gender	D.O.B	Age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
NATHANIEL FORD	PT101	Female	05-11-1992	31	0	1	never	25.19	6.6	140	0
GARY JIMENEZ	PT102	Female	11-11-1992	31	0	0	No Info	27.32	6.6	80	0
ALBERT PARDINI	PT103	Male	13-11-1992	31	0	0	never	27.32	5.7	158	0
CHRISTOPHER CHONG	PT104	Female	05-12-1992	35	0	0	current	23.45	5	155	0
PATRICK GARDNER	PT105	Male	03-01-1989	35	1	1	current	20.14	4.8	155	0
DAVID SULLIVAN	PT106	Female	05-01-1989	35	0	0	never	27.32	6.6	85	0
ALSON LEE	PT107	Female	23-01-1989	35	0	0	never	19.31	6.5	200	1
DAVID KUSHNER	PT108	Female	05-02-1989	35	0	0	No Info	23.86	5.7	85	0
MICHAEL MORRIS	PT109	Male	21-02-1989	35	0	0	never	33.64	4.8	145	0
JOANNE HAYES-WHITE	PT110	Female	09-03-1989	35	0	0	never	27.32	5	100	0
ARTHUR KENNEY	PT111	Female	19-03-1989	35	0	0	never	27.32	6.1	85	0
PATRICIA JACKSON	PT112	Female	01-04-1989	35	0	0	former	54.7	6	100	0
EDWARD HARRINGTON	PT113	Female	14-04-1989	35	0	0	former	36.05	5	130	0
JOHN MARTIN	PT114	Female	21-04-1989	35	0	0	never	25.69	5.8	200	0
DAVID FRANKLIN	PT115	Female	26-04-1989	35	0	0	No Info	27.32	5	160	0
RICHARD CORRIEA	PT116	Male	27-04-1989	35	0	0	No Info	27.32	6.6	126	0
AMY HART	PT117	Male	29-04-1989	35	0	0	never	30.36	6.1	200	0
SEBASTIAN WONG	PT118	Female	30-04-1989	35	0	0	never	24.48	5.7	158	0

1. Retrieve the Patient_id and ages of all patients.

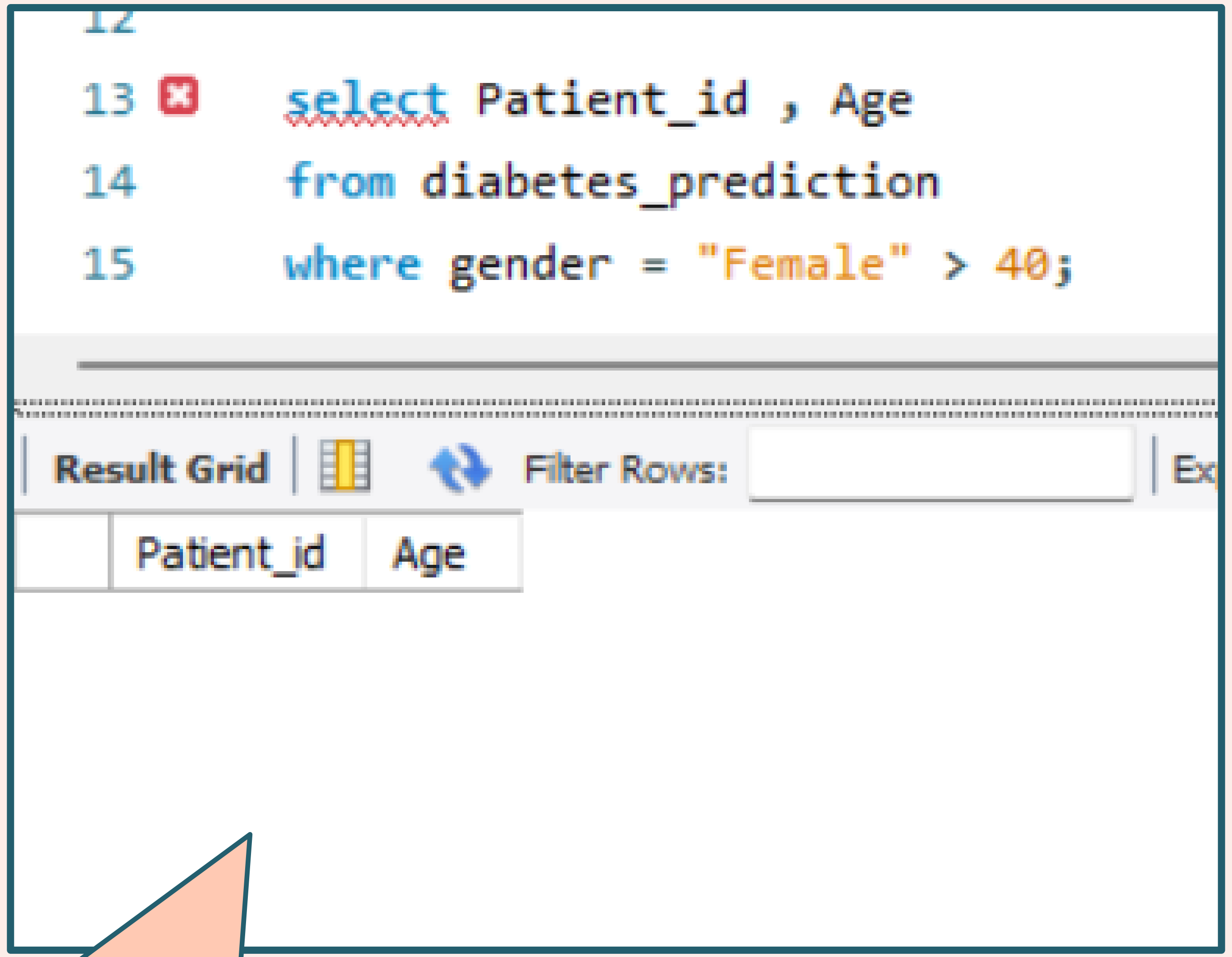


```sql
-- Q1. Retrieve the Patient_id and ages of all patients.
select Patient_id, Age
from diabetes_prediction
```



Patient_id	Age
PT101	31
PT102	31
PT103	31
PT104	35
PT105	35
PT106	35
PT107	35
PT108	35
PT109	35
PT110	35

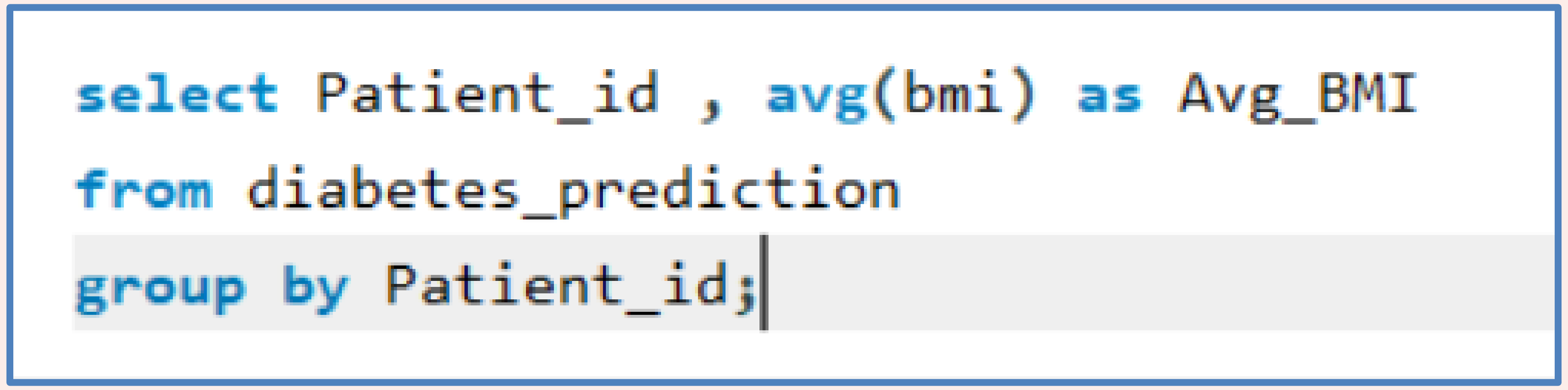
2. Select all female patients who are older than 40.



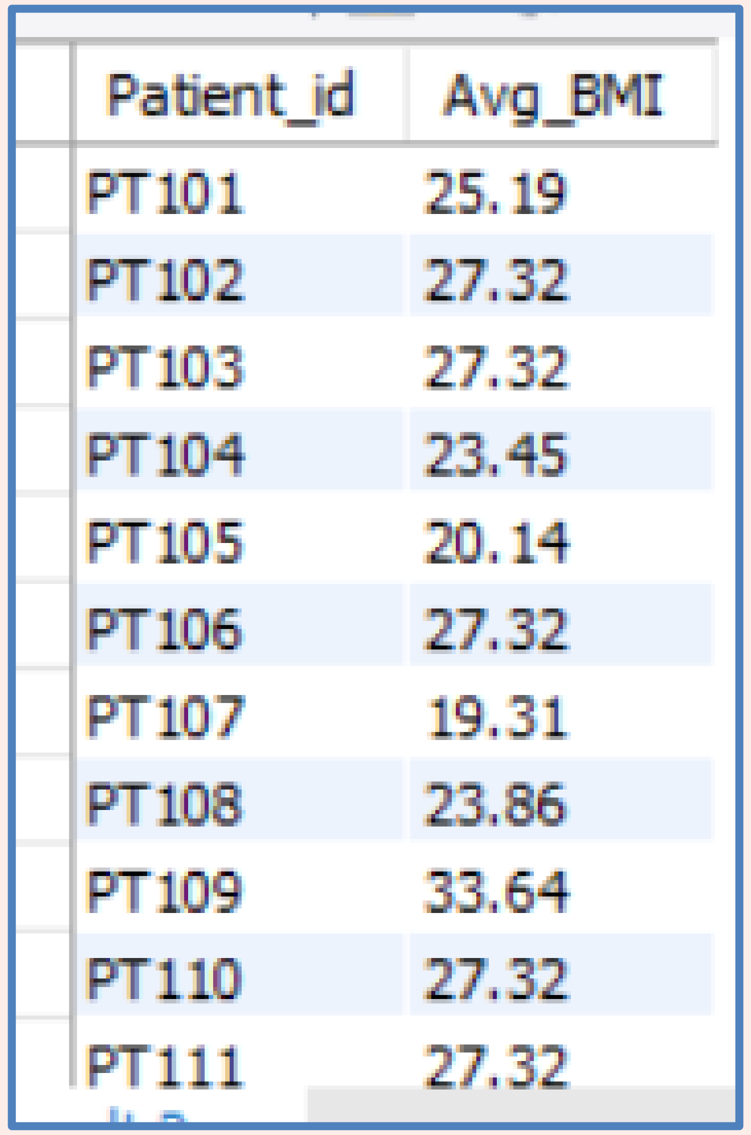
```
select Patient_id , Age
from diabetes_prediction
where gender = "Female" > 40;
```

There is no data on Female patients who are older than 40.

3. Calculate the average BMI of patients.

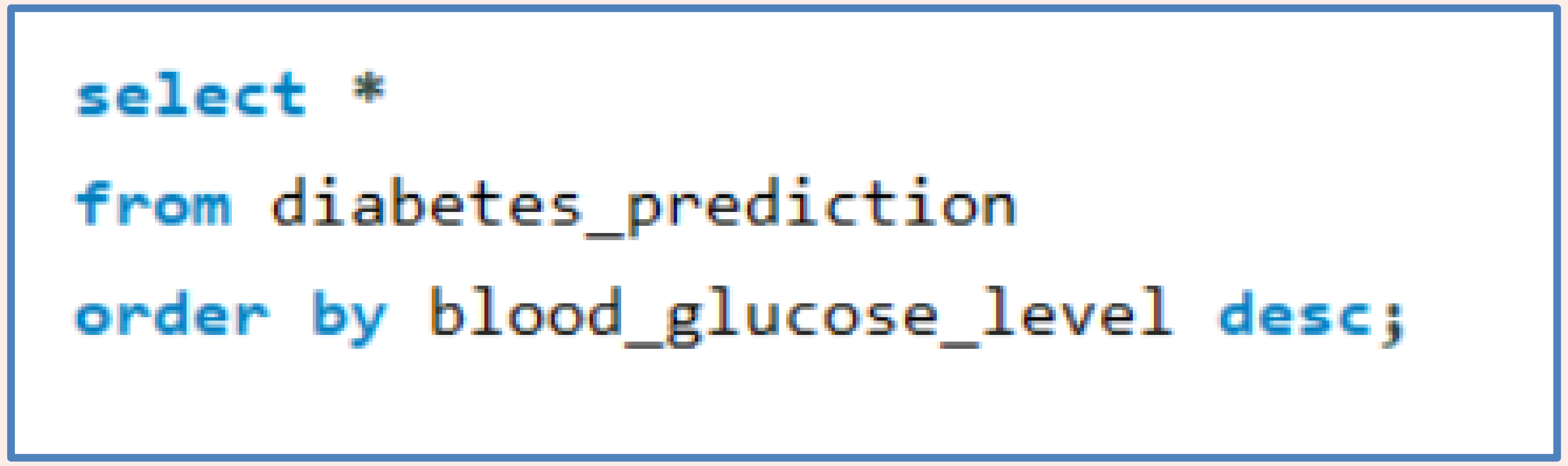


```
select Patient_id , avg(bmi) as Avg_BMI
from diabetes_prediction
group by Patient_id;
```

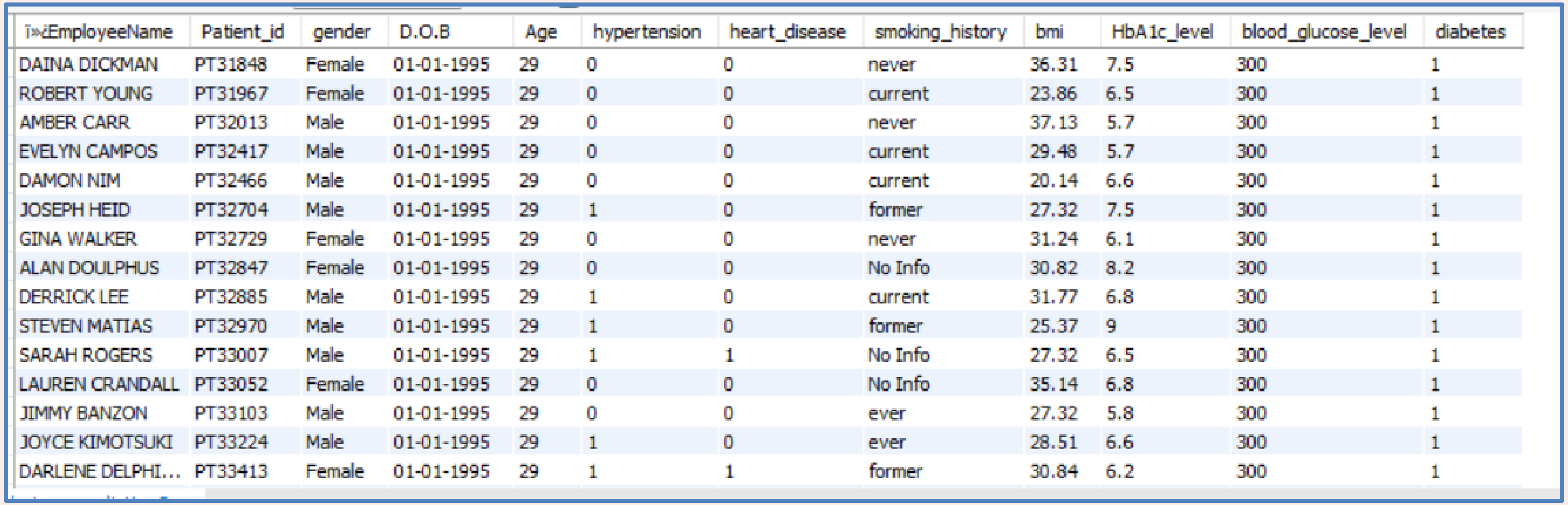


Patient_id	Avg_BMI
PT101	25.19
PT102	27.32
PT103	27.32
PT104	23.45
PT105	20.14
PT106	27.32
PT107	19.31
PT108	23.86
PT109	33.64
PT110	27.32
PT111	27.32

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

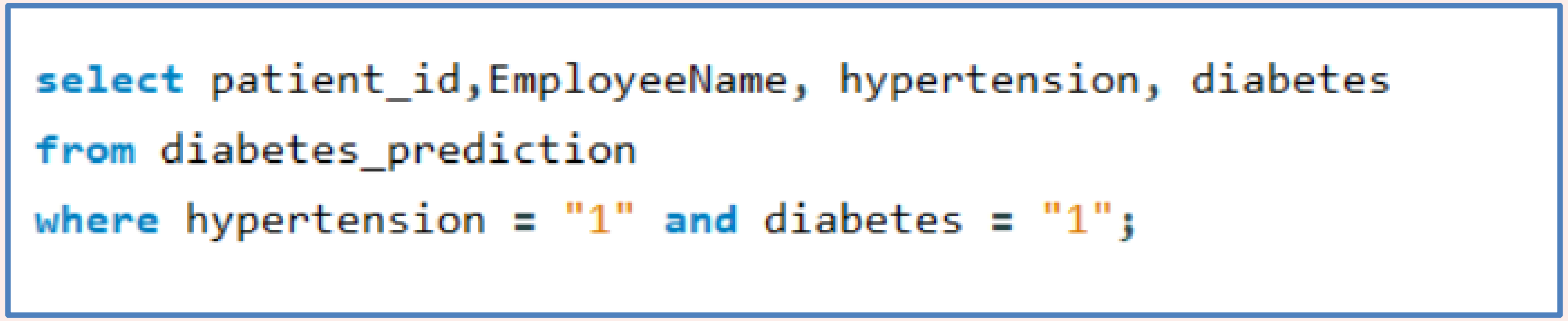


```
select *
from diabetes_prediction
order by blood_glucose_level desc;
```

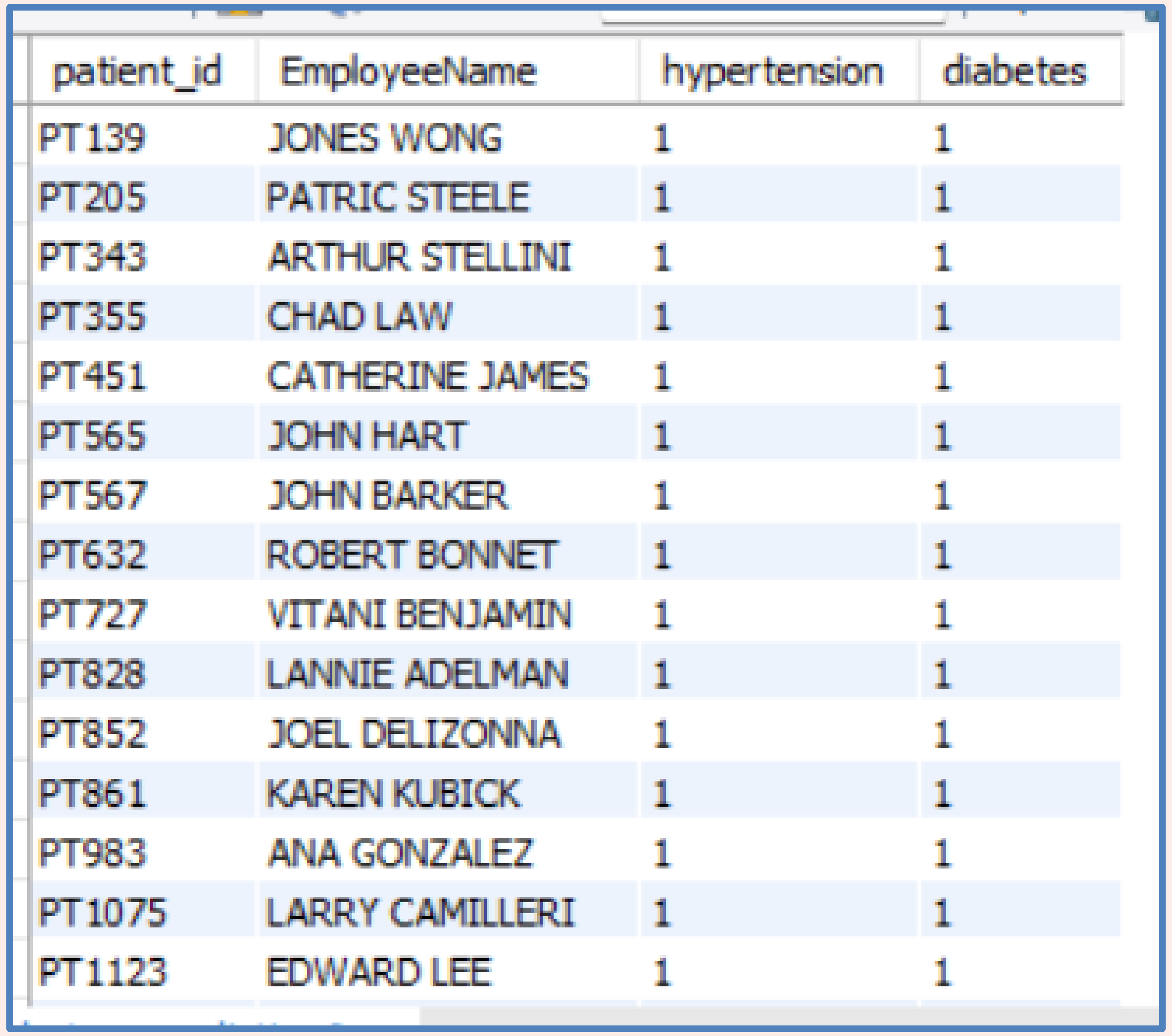


ï»¿EmployeeName	Patient_id	gender	D.O.B	Age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
DAINA DICKMAN	PT31848	Female	01-01-1995	29	0	0	never	36.31	7.5	300	1
ROBERT YOUNG	PT31967	Female	01-01-1995	29	0	0	current	23.86	6.5	300	1
AMBER CARR	PT32013	Male	01-01-1995	29	0	0	never	37.13	5.7	300	1
EVELYN CAMPOS	PT32417	Male	01-01-1995	29	0	0	current	29.48	5.7	300	1
DAMON NIM	PT32466	Male	01-01-1995	29	0	0	current	20.14	6.6	300	1
JOSEPH HEID	PT32704	Male	01-01-1995	29	1	0	former	27.32	7.5	300	1
GINA WALKER	PT32729	Female	01-01-1995	29	0	0	never	31.24	6.1	300	1
ALAN DOULPHUS	PT32847	Female	01-01-1995	29	0	0	No Info	30.82	8.2	300	1
DERRICK LEE	PT32885	Male	01-01-1995	29	1	0	current	31.77	6.8	300	1
STEVEN MATIAS	PT32970	Male	01-01-1995	29	1	0	former	25.37	9	300	1
SARAH ROGERS	PT33007	Male	01-01-1995	29	1	1	No Info	27.32	6.5	300	1
LAUREN CRANDALL	PT33052	Female	01-01-1995	29	0	0	No Info	35.14	6.8	300	1
JIMMY BANZON	PT33103	Male	01-01-1995	29	0	0	ever	27.32	5.8	300	1
JOYCE KIMOTSUKI	PT33224	Male	01-01-1995	29	1	0	ever	28.51	6.6	300	1
DARLENE DELPHI...	PT33413	Female	01-01-1995	29	1	1	former	30.84	6.2	300	1

5. Find patients who have hypertension and diabetes.

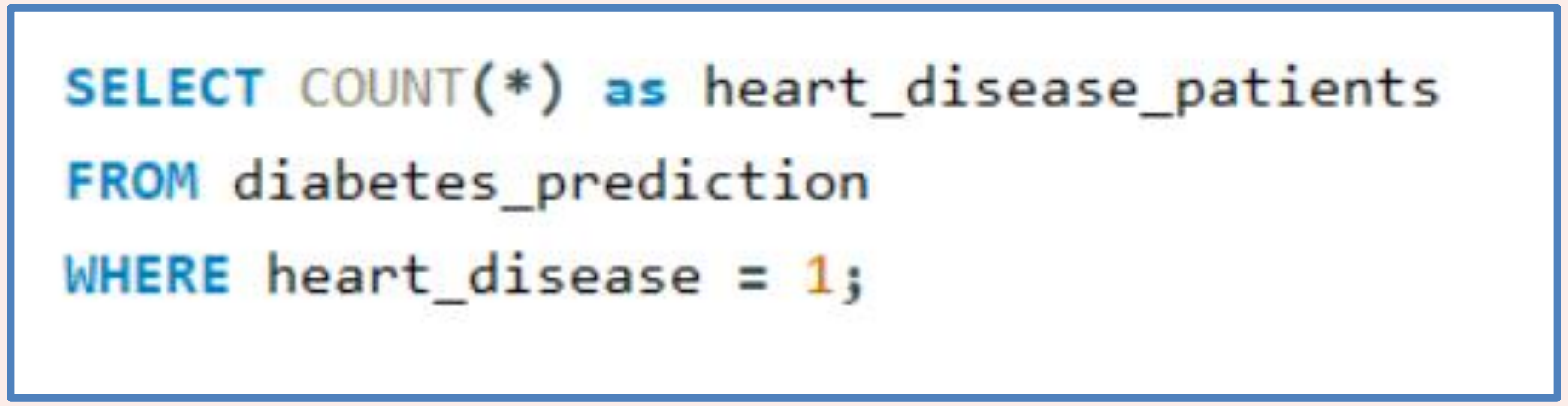


```
select patient_id,EmployeeName, hypertension, diabetes
from diabetes_prediction
where hypertension = "1" and diabetes = "1";
```

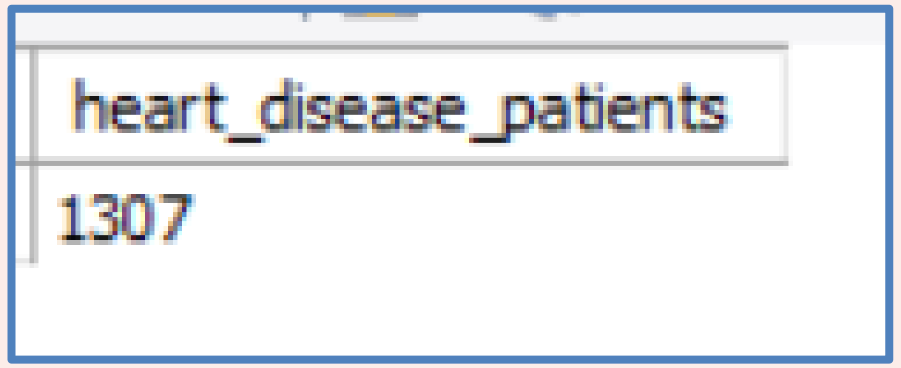


patient_id	EmployeeName	hypertension	diabetes
PT139	JONES WONG	1	1
PT205	PATRIC STEELE	1	1
PT343	ARTHUR STELLINI	1	1
PT355	CHAD LAW	1	1
PT451	CATHERINE JAMES	1	1
PT565	JOHN HART	1	1
PT567	JOHN BARKER	1	1
PT632	ROBERT BONNET	1	1
PT727	VITANI BENJAMIN	1	1
PT828	LANNIE ADELMAN	1	1
PT852	JOEL DELIZONNA	1	1
PT861	KAREN KUBICK	1	1
PT983	ANA GONZALEZ	1	1
PT1075	LARRY CAMILLERI	1	1
PT1123	EDWARD LEE	1	1

6. Determine the number of patients with heart disease.

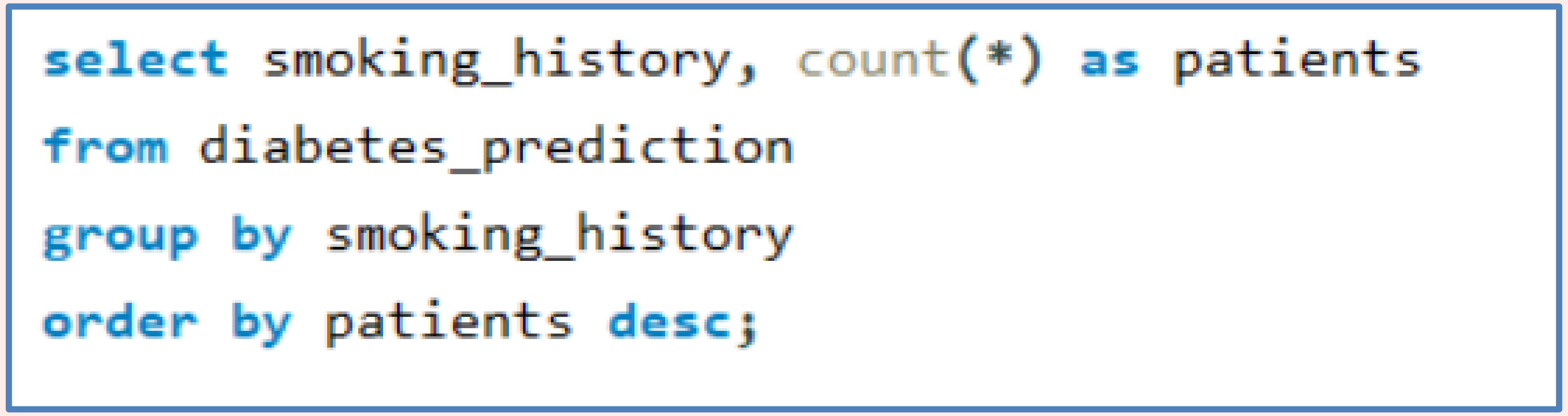


```sql
SELECT COUNT(*) as heart_disease_patients
FROM diabetes_prediction
WHERE heart_disease = 1;
```

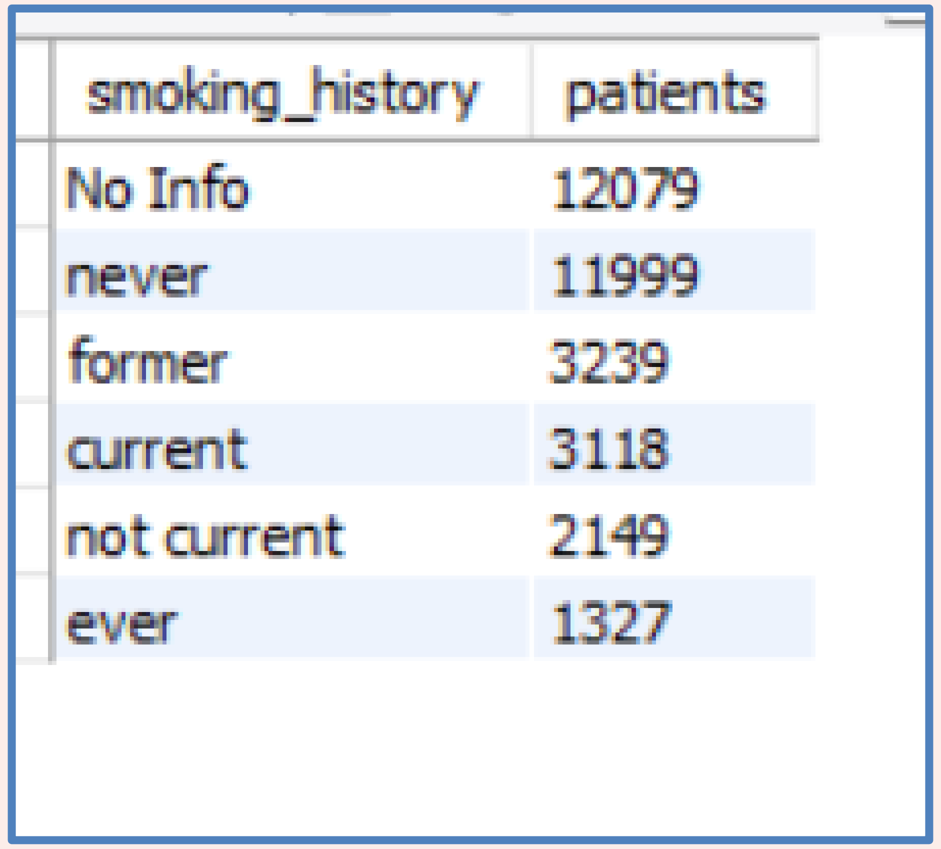


```
heart_disease_patients
1307
```

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

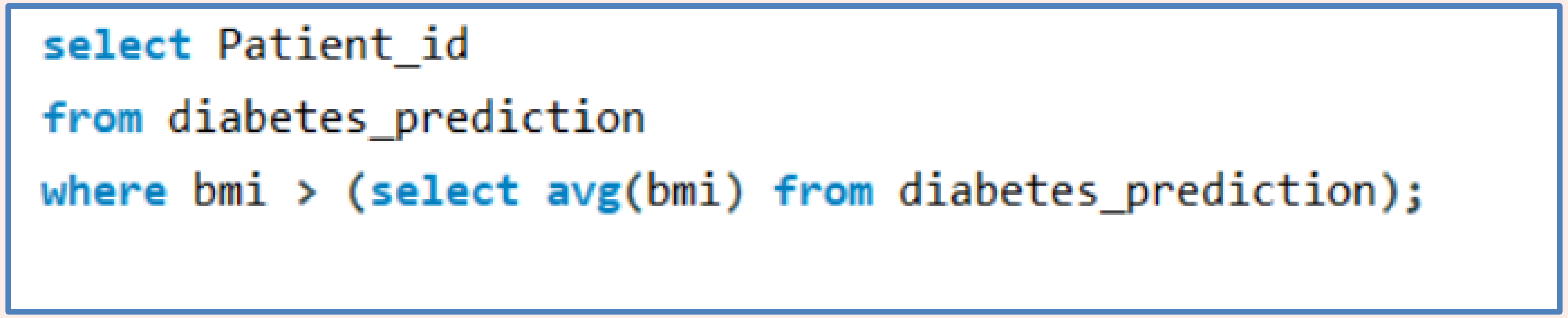


```
select smoking_history, count(*) as patients
from diabetes_prediction
group by smoking_history
order by patients desc;
```

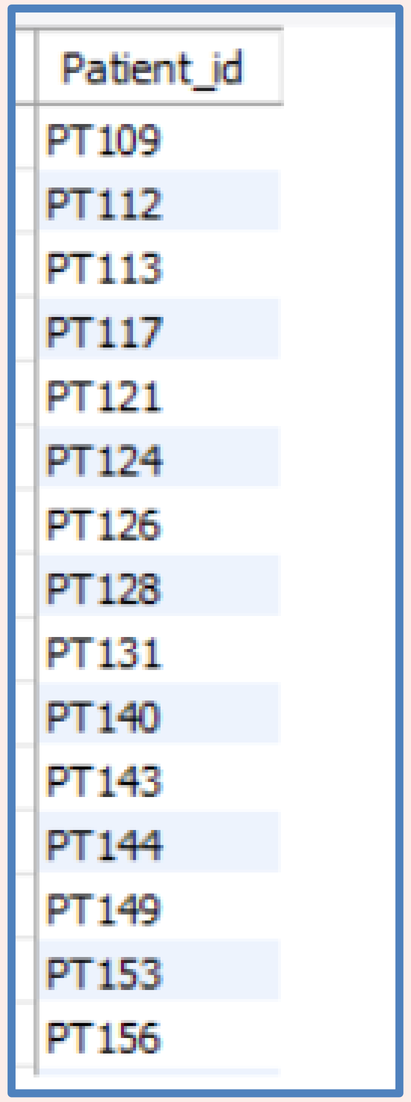


smoking_history	patients
No Info	12079
never	11999
former	3239
current	3118
not current	2149
ever	1327

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

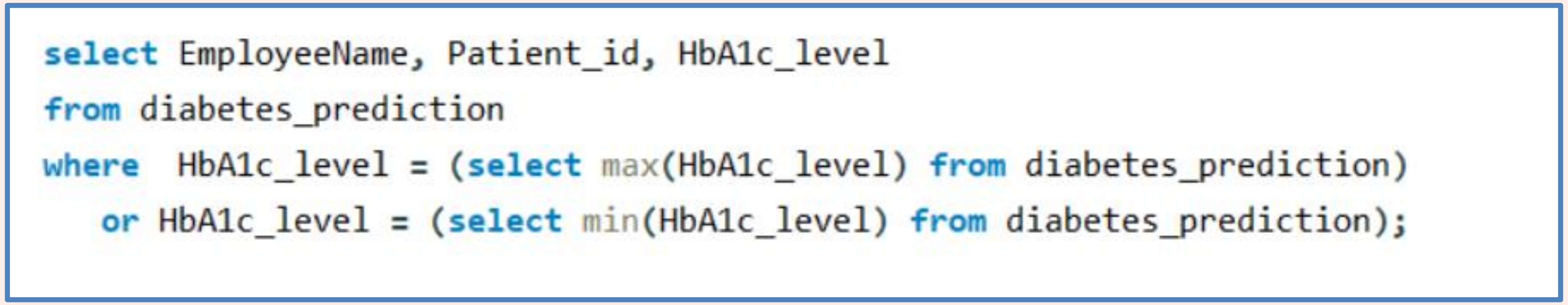


```
select Patient_id
from diabetes_prediction
where bmi > (select avg(bmi) from diabetes_prediction);
```

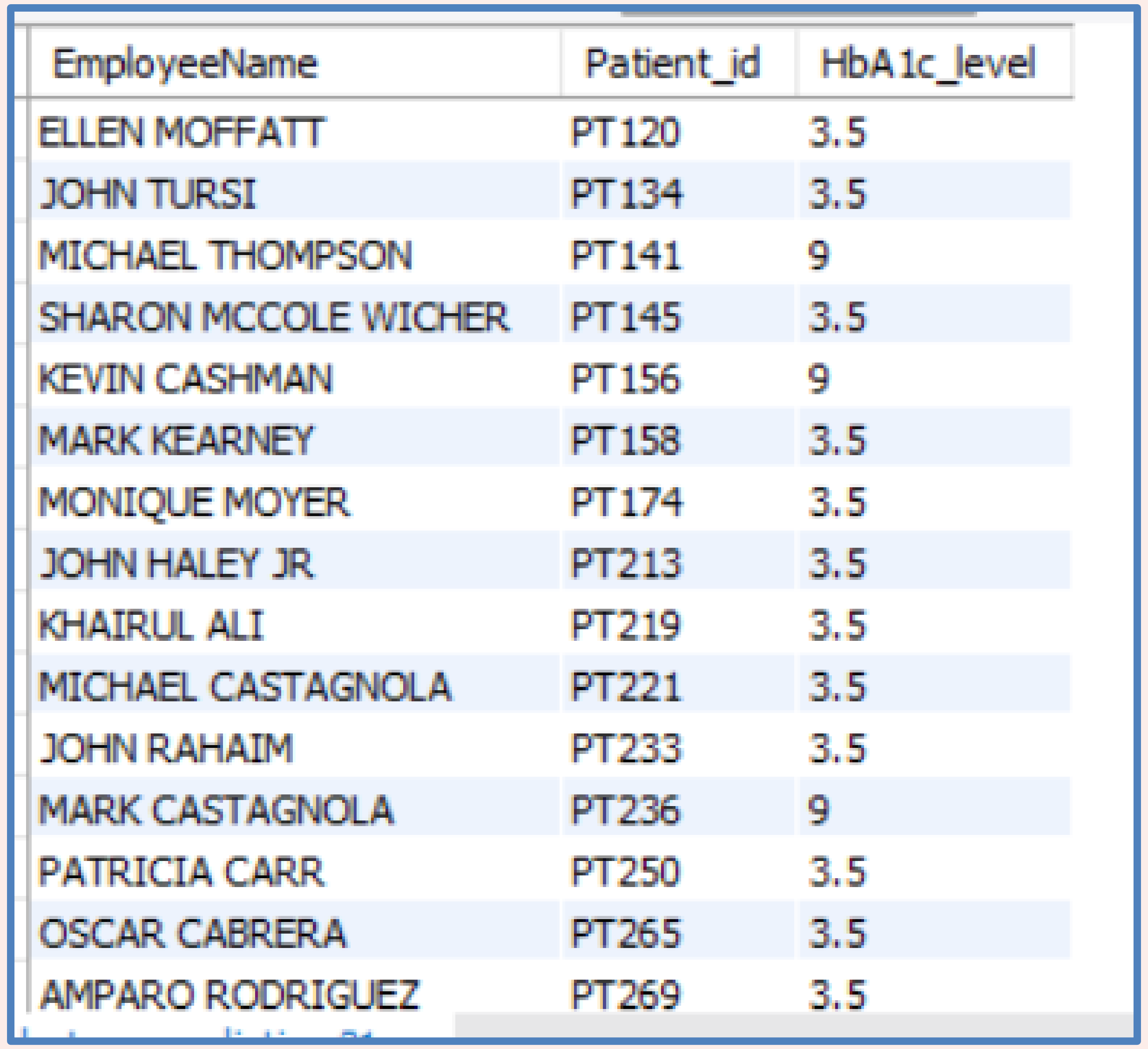


Patient_id
PT109
PT112
PT113
PT117
PT121
PT124
PT126
PT128
PT131
PT140
PT143
PT144
PT149
PT153
PT156

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

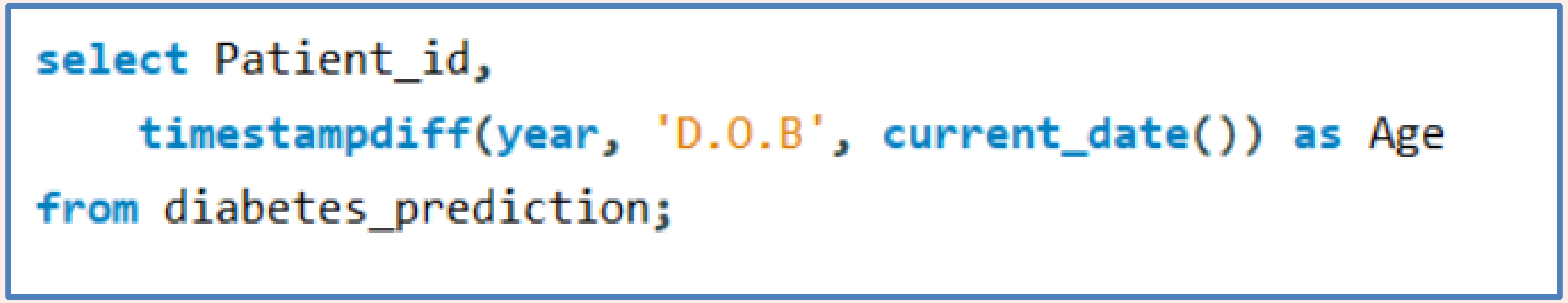


```
select EmployeeName, Patient_id, HbA1c_level
from diabetes_prediction
where  HbA1c_level = (select max(HbA1c_level) from diabetes_prediction)
   or HbA1c_level = (select min(HbA1c_level) from diabetes_prediction);
```

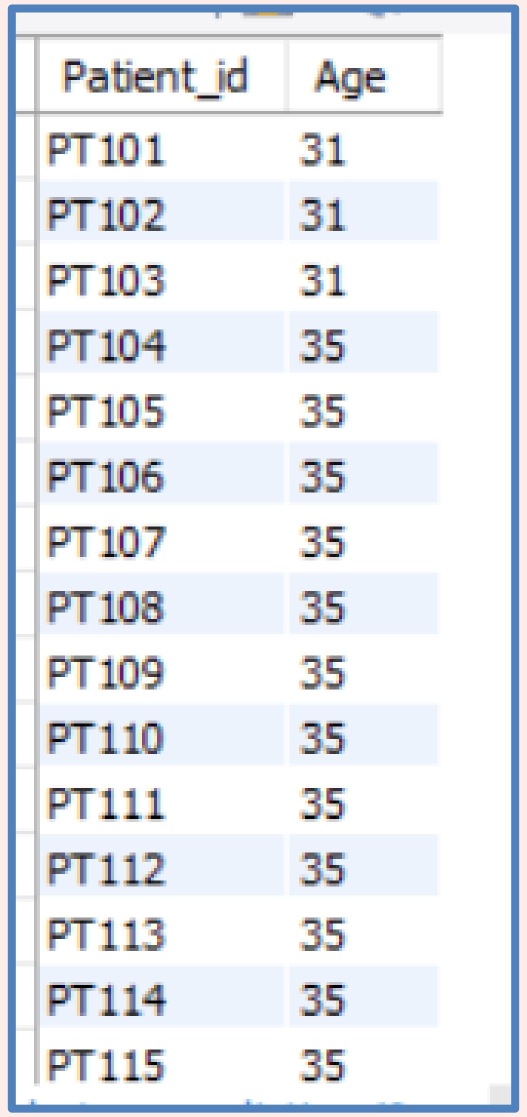


EmployeeName	Patient_id	HbA1c_level
ELLEN MOFFATT	PT120	3.5
JOHN TURSI	PT134	3.5
MICHAEL THOMPSON	PT141	9
SHARON MCCOLE WICHER	PT145	3.5
KEVIN CASHMAN	PT156	9
MARK KEARNEY	PT158	3.5
MONIQUE MOYER	PT174	3.5
JOHN HALEY JR	PT213	3.5
KHAIRUL ALI	PT219	3.5
MICHAEL CASTAGNOLA	PT221	3.5
JOHN RAHAIM	PT233	3.5
MARK CASTAGNOLA	PT236	9
PATRICIA CARR	PT250	3.5
OSCAR CABRERA	PT265	3.5
AMPARO RODRIGUEZ	PT269	3.5

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

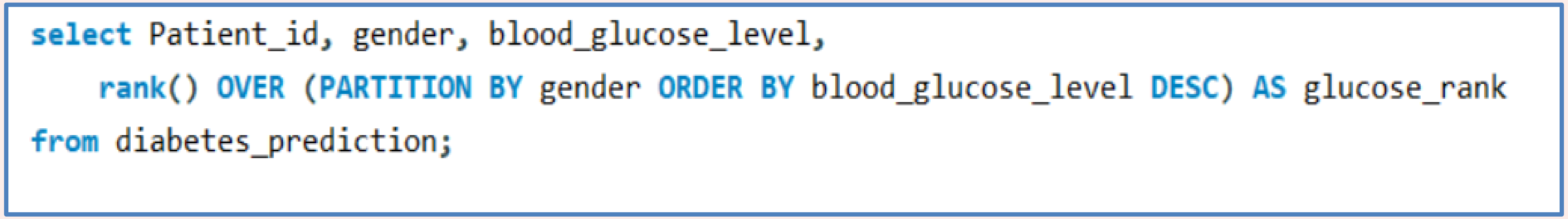


```sql
select Patient_id,
    timestampdiff(year, 'D.O.B', current_date()) as Age
from diabetes_prediction;
```

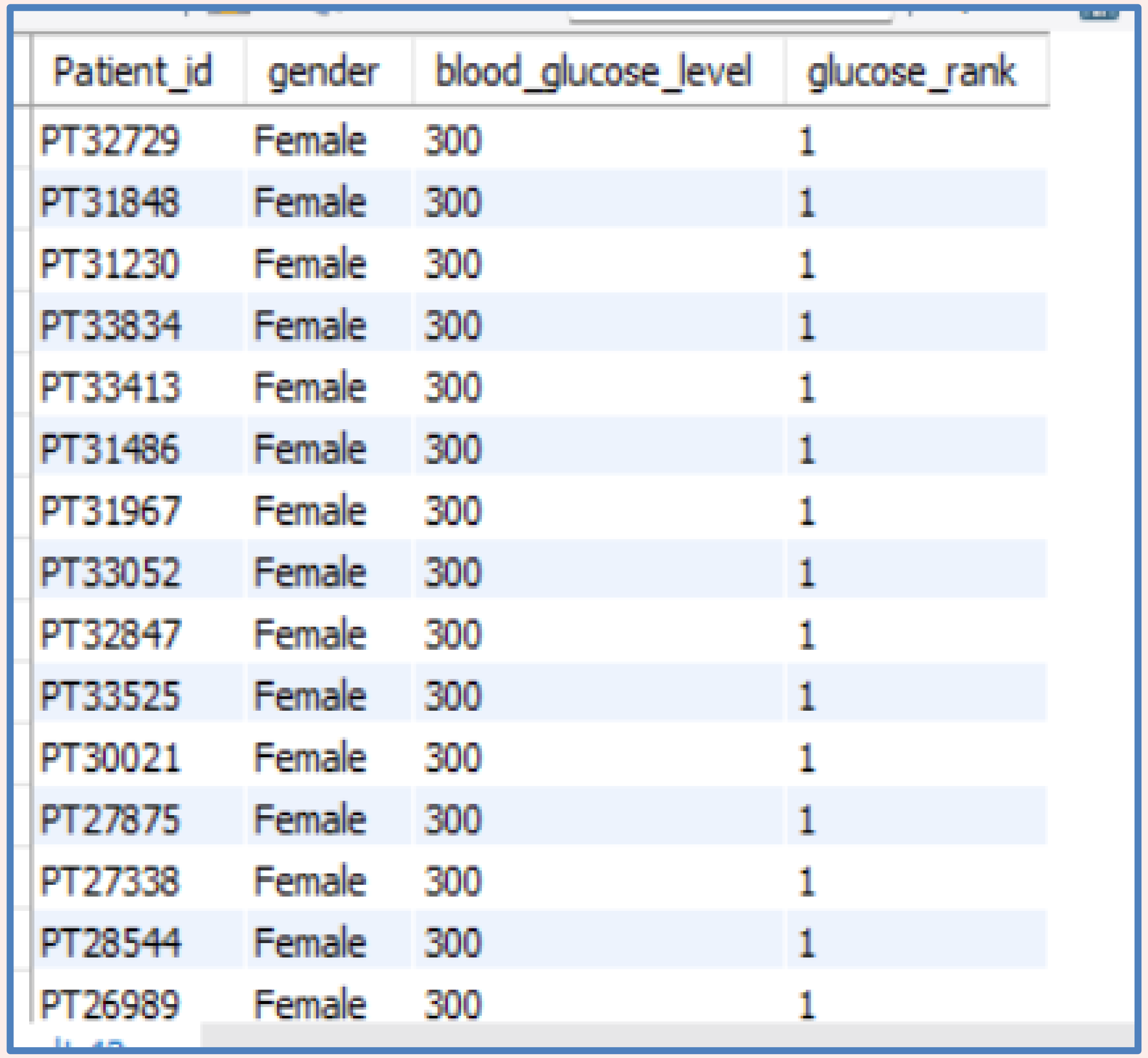


Patient_id	Age
PT101	31
PT102	31
PT103	31
PT104	35
PT105	35
PT106	35
PT107	35
PT108	35
PT109	35
PT110	35
PT111	35
PT112	35
PT113	35
PT114	35
PT115	35

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

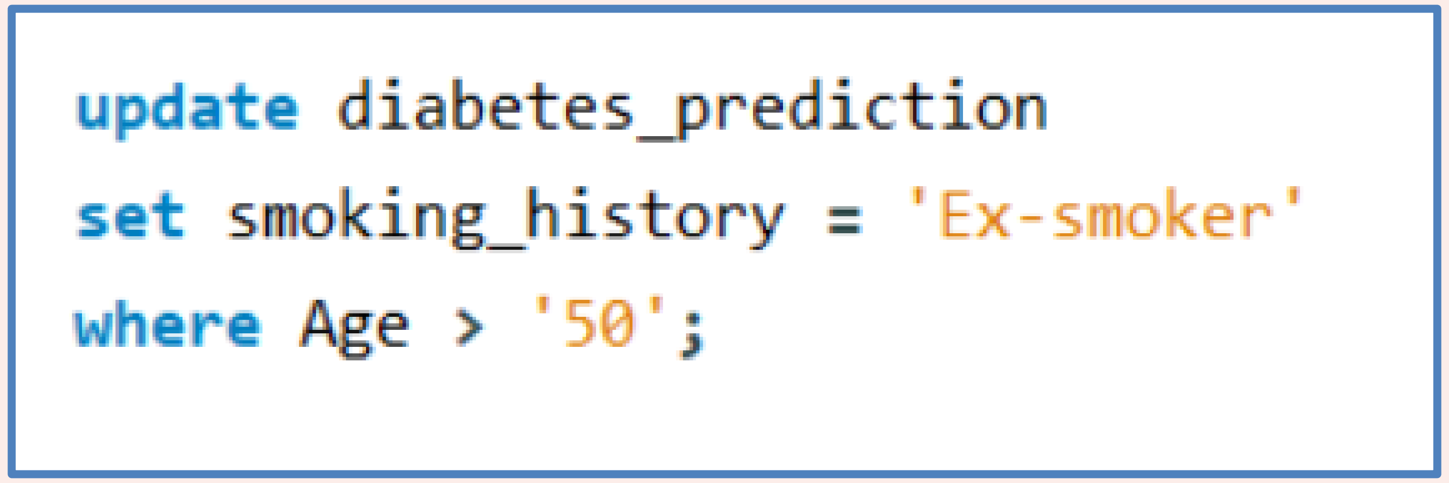


```sql
select Patient_id, gender, blood_glucose_level,
    rank() OVER (PARTITION BY gender ORDER BY blood_glucose_level DESC) AS glucose_rank
from diabetes_prediction;
```



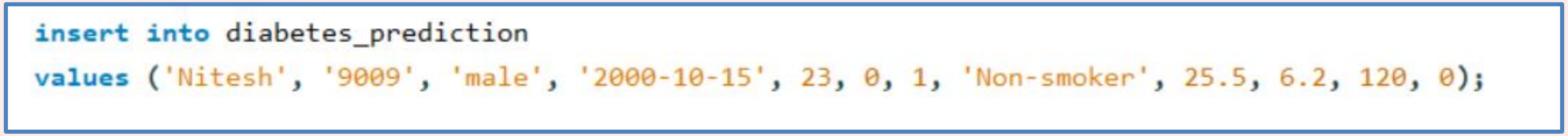
Patient_id	gender	blood_glucose_level	glucose_rank
PT32729	Female	300	1
PT31848	Female	300	1
PT31230	Female	300	1
PT33834	Female	300	1
PT33413	Female	300	1
PT31486	Female	300	1
PT31967	Female	300	1
PT33052	Female	300	1
PT32847	Female	300	1
PT33525	Female	300	1
PT30021	Female	300	1
PT27875	Female	300	1
PT27338	Female	300	1
PT28544	Female	300	1
PT26989	Female	300	1

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

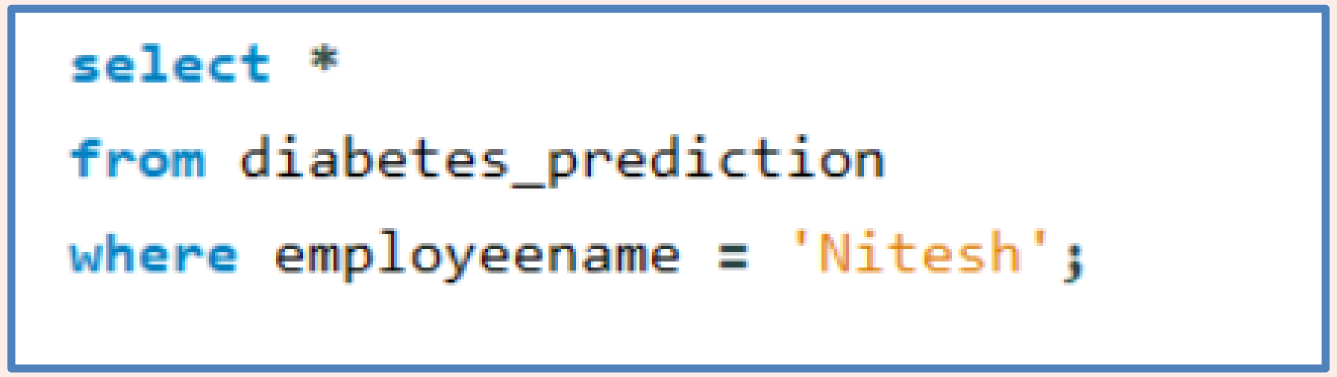


```sql
update diabetes_prediction
set smoking_history = 'Ex-smoker'
where Age > '50';
```

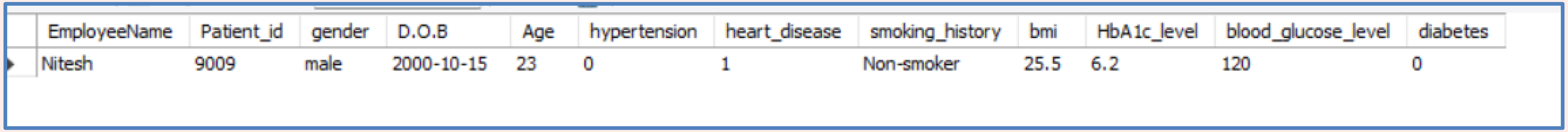
13. Insert a new patient into the database with sample data.



```
insert into diabetes_prediction
values ('Nitesh', '9009', 'male', '2000-10-15', 23, 0, 1, 'Non-smoker', 25.5, 6.2, 120, 0);
```

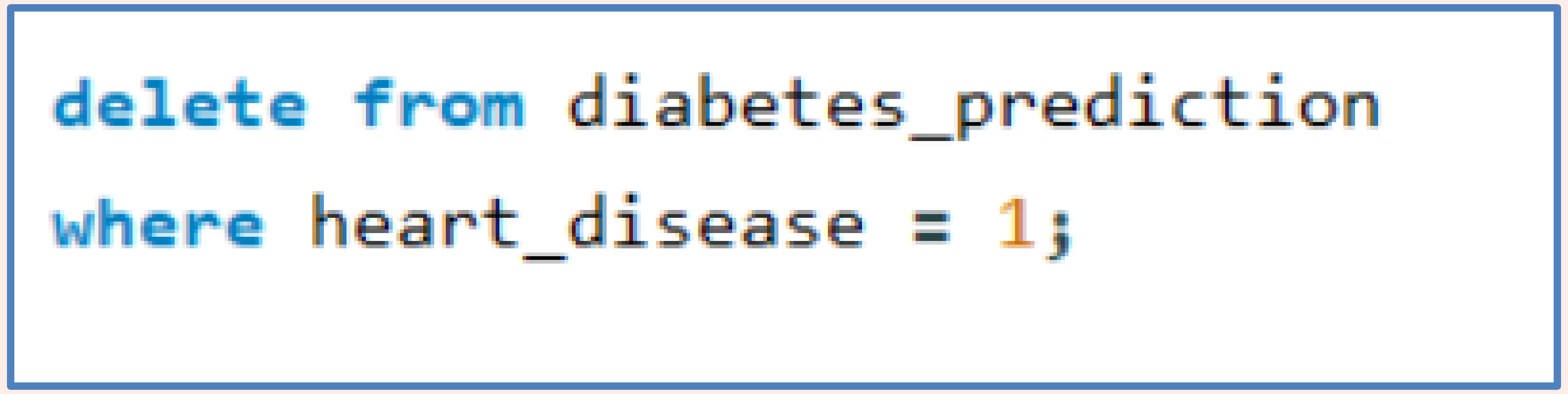


```
select *
from diabetes_prediction
where employeename = 'Nitesh';
```



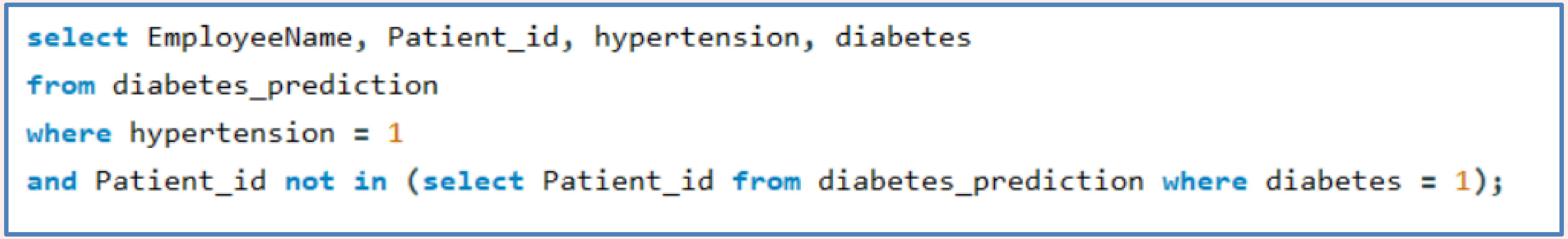
EmployeeName	Patient_id	gender	D.O.B	Age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Nitesh	9009	male	2000-10-15	23	0	1	Non-smoker	25.5	6.2	120	0

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



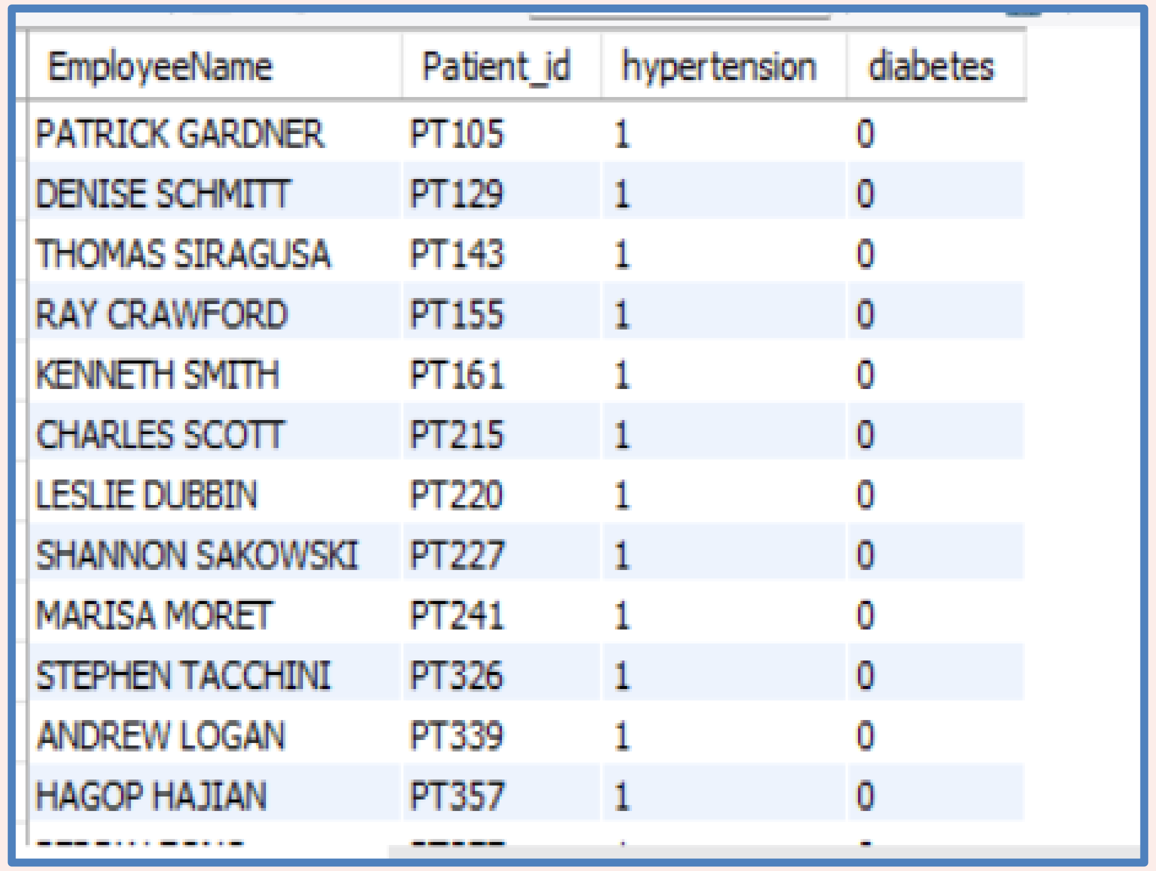
```
delete from diabetes_prediction
where heart_disease = 1;
```

15. Find patients who have hypertension but not diabetes using the EXCEPT operator. 2 of 2



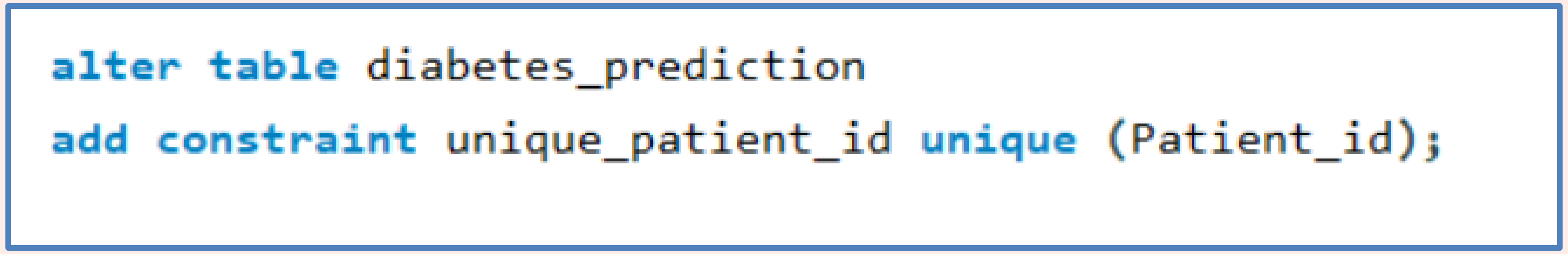
```
select EmployeeName, Patient_id, hypertension, diabetes
from diabetes_prediction
where hypertension = 1
and Patient_id not in (select Patient_id from diabetes_prediction where diabetes = 1);
```

The EXCEPT operator isn't directly available in MySQL.



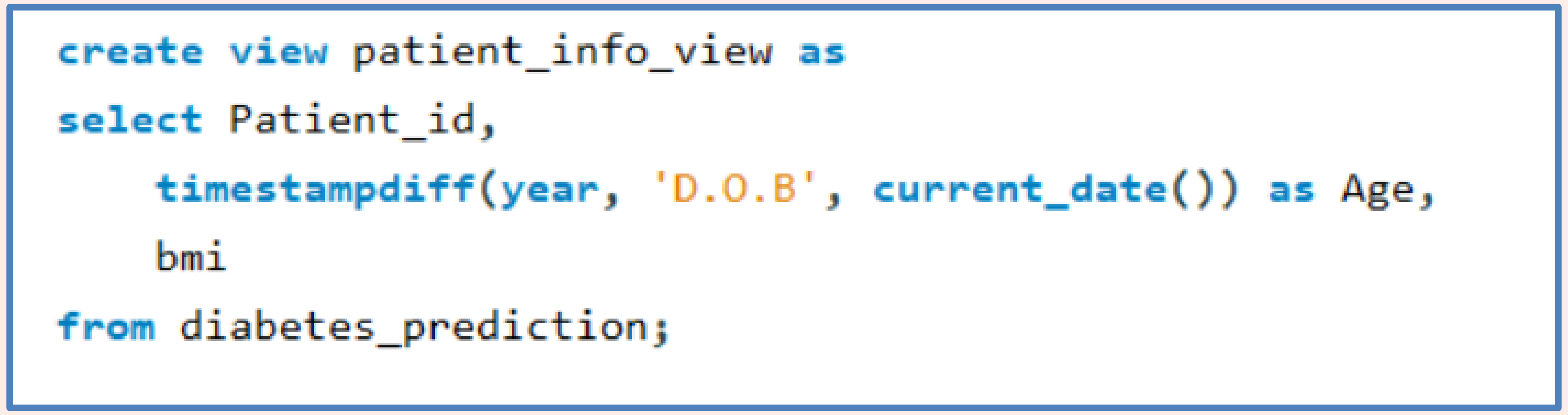
EmployeeName	Patient_id	hypertension	diabetes
PATRICK GARDNER	PT105	1	0
DENISE SCHMITT	PT129	1	0
THOMAS SIRAGUSA	PT143	1	0
RAY CRAWFORD	PT155	1	0
KENNETH SMITH	PT161	1	0
CHARLES SCOTT	PT215	1	0
LESLIE DUBBIN	PT220	1	0
SHANNON SAKOWSKI	PT227	1	0
MARISA MORET	PT241	1	0
STEPHEN TACCHINI	PT326	1	0
ANDREW LOGAN	PT339	1	0
HAGOP HAJIAN	PT357	1	0

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

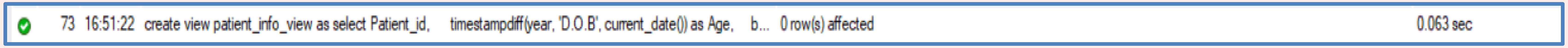


```
alter table diabetes_prediction
add constraint unique_patient_id unique (Patient_id);
```

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



```
create view patient_info_view as
select Patient_id,
    timestampdiff(year, 'D.O.B', current_date()) as Age,
    bmi
from diabetes_prediction;
```



```
73  16:51:22  create view patient_info_view as select Patient_id,     timestampdiff(year, 'D.O.B', current_date()) as Age,     b...  0 row(s) affected          0.063 sec
```

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

- **Normalization:** Break down data into separate tables based on entities. Use primary and foreign keys for relationships.
- **Entity-Relationship Diagram (ERD):** Visualize entity relationships to identify normalization opportunities. Define entities like Patients, Medical History, etc.
- **Attribute Separation:** Split columns into distinct attributes for clarity. For example, store medical conditions separately.
- **Data Types and Constraints:** Employ appropriate data types for efficient storage. Implement constraints for data integrity.
- **Normalization Levels:** Aim for 3NF or higher to minimize redundancy. Decompose tables to eliminate dependencies.
- **Indexing:** Create indexes on frequently used columns for better performance. Improve search and retrieval efficiency.
- **Data Validation and Cleaning:** Enforce validation rules to ensure data consistency. Regularly clean and validate data for accuracy.

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

To optimize the performance of SQL queries on your dataset, consider the following strategies:

1. **Indexing:** Identify columns frequently used in WHERE clauses or JOIN conditions and create indexes on those columns. This speeds up data retrieval by enabling the database to quickly locate relevant rows.
2. **Query Optimization:** Write efficient queries by avoiding unnecessary computations, using appropriate JOIN types, and limiting the number of rows returned. Use EXPLAIN to analyze query execution plans and identify areas for optimization.
3. **Data Partitioning:** If the dataset is large, consider partitioning tables based on certain criteria (e.g., date ranges) to distribute data across multiple storage devices. This can improve query performance by reducing the amount of data that needs to be scanned.
4. **Normalization:** Ensure that the database schema is properly normalized to minimize data redundancy and improve query efficiency. Use appropriate indexing and JOIN operations to retrieve data from related tables.
5. **Caching:** Implement caching mechanisms to store frequently accessed query results in memory. This reduces the need to recompute results for identical queries, improving overall performance.

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

6. **Hardware Optimization:** Ensure that the database server is properly configured with sufficient memory, CPU resources, and disk I/O capacity to handle query loads efficiently. Consider using solid-state drives (SSDs) for faster data access.
7. **Query Tuning:** Monitor query performance regularly and identify slow-performing queries using tools like EXPLAIN and query logs. Optimize these queries by rewriting them, adding or adjusting indexes, or restructuring the data model if necessary.
8. **Connection Pooling:** Use connection pooling to reduce the overhead of establishing and tearing down database connections for each query. This improves query response time by reusing existing connections.
9. **Query Caching:** Enable query caching if your database management system supports it. This caches the results of frequently executed queries, reducing the need for redundant computations.
10. **Database Maintenance:** Regularly perform database maintenance tasks such as vacuuming, reindexing, and updating statistics to ensure optimal performance and prevent performance degradation over time.

PSYLIQ

THANK YOU!

PRESENTED BY: NITESH KUMAR PRAJAPATI

www.linkedin.com/in/nitesh-kumar-prajapati