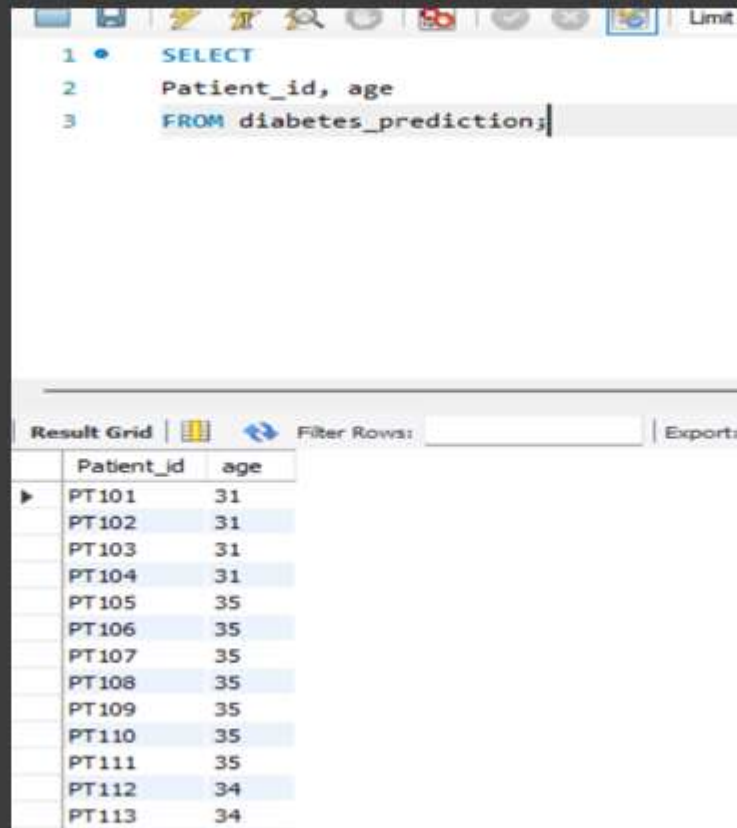




Data Analyst Internship

**TASK 3: DIABETES PREDICTION ANALYSIS
BY
AMAN SHAIKH**

Q1. Retrieve the Patient_id and ages of all patients.



The screenshot shows a SQL query editor window with a toolbar at the top. The query text is as follows:

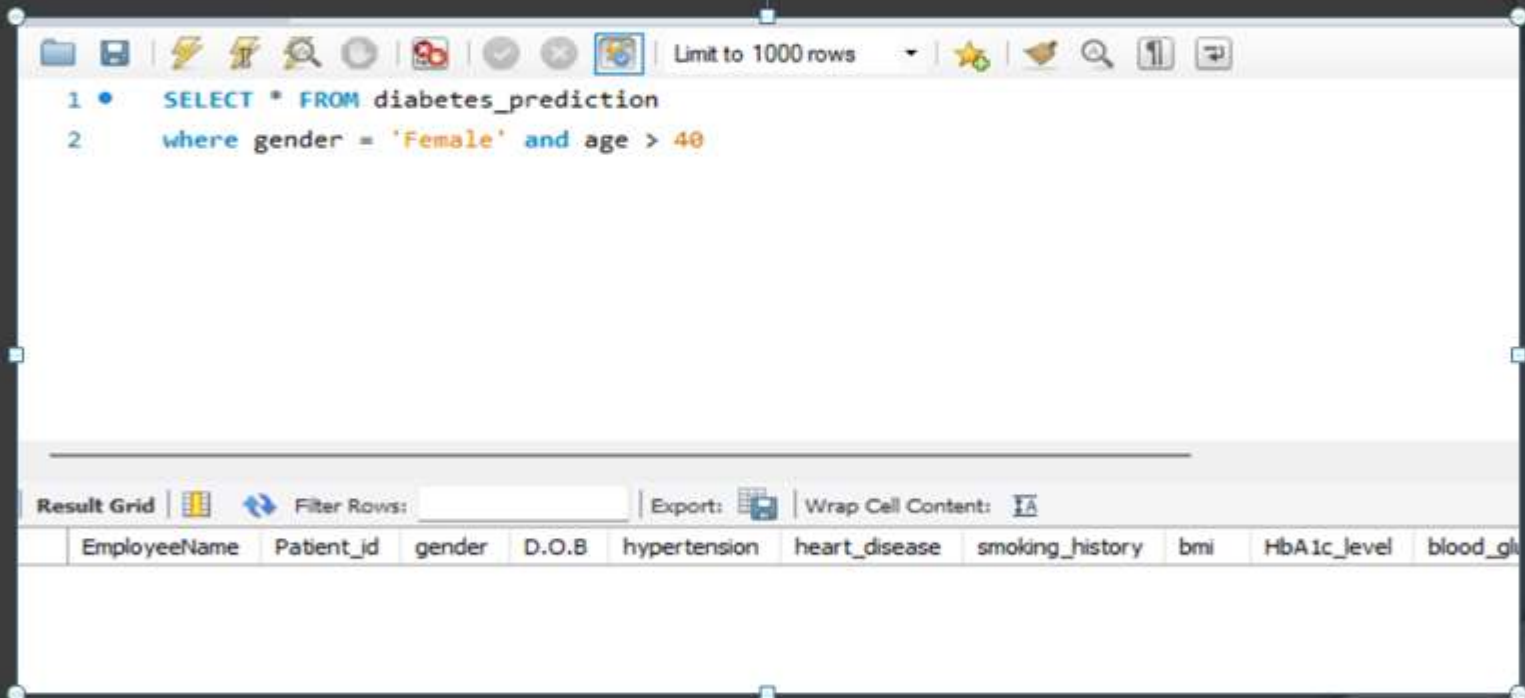
```
1 • SELECT
2 Patient_id, age
3 FROM diabetes_prediction;
```

Below the query editor is the 'Result Grid' tab, which displays the results of the query. It includes a 'Filter Rows' input field and an 'Export' button. The results are shown in a table with two columns: 'Patient_id' and 'age'.

	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	34
	PT113	34

Q2. Select all female patients who are older than 40.

In our patient population, there are no female individuals exceeding the age of 40.



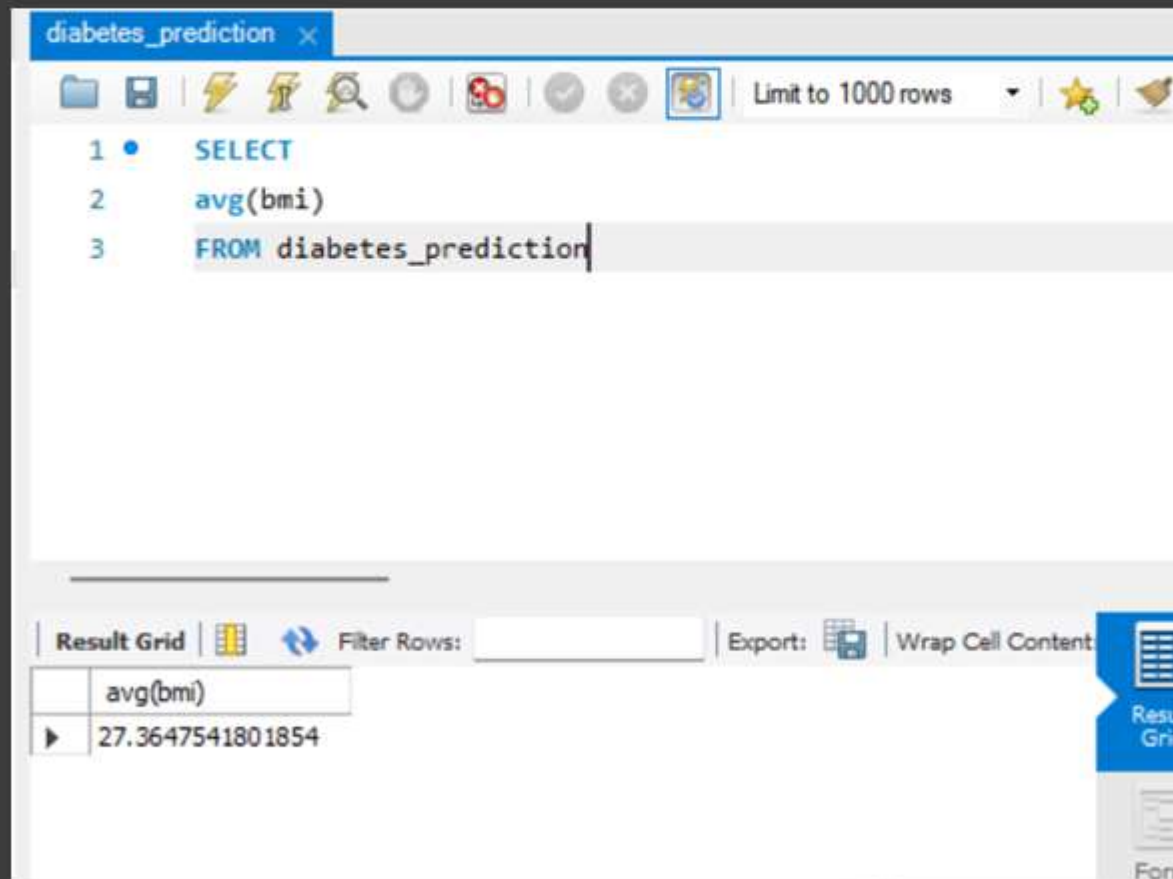
The screenshot shows a SQL query editor window. The query is as follows:

```
1 • SELECT * FROM diabetes_prediction
2   where gender = 'Female' and age > 40
```

The editor has a toolbar at the top with various icons and a "Limit to 1000 rows" dropdown. At the bottom, there is a "Result Grid" section with a table header containing the following columns: EmployeeName, Patient_id, gender, D.O.B, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, and blood_glucose.

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose
--------------	------------	--------	-------	--------------	---------------	-----------------	-----	-------------	---------------

Q3. Calculate the average BMI of patients.



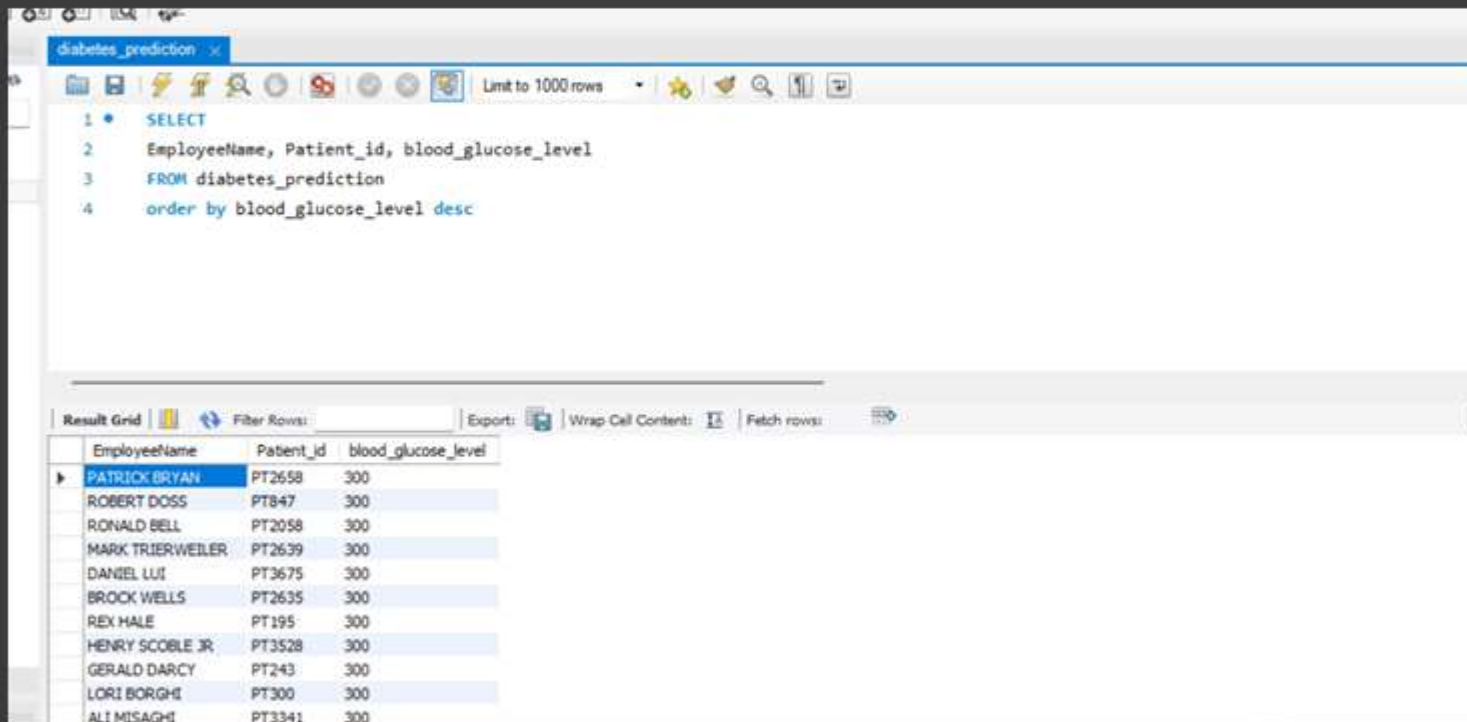
The screenshot shows a SQL query editor window titled "diabetes_prediction". The query is as follows:

```
1 SELECT
2 avg(bmi)
3 FROM diabetes_prediction
```

The result grid at the bottom shows the following data:

	avg(bmi)
▶	27.3647541801854

Q4. List patients in descending order of blood glucose levels.



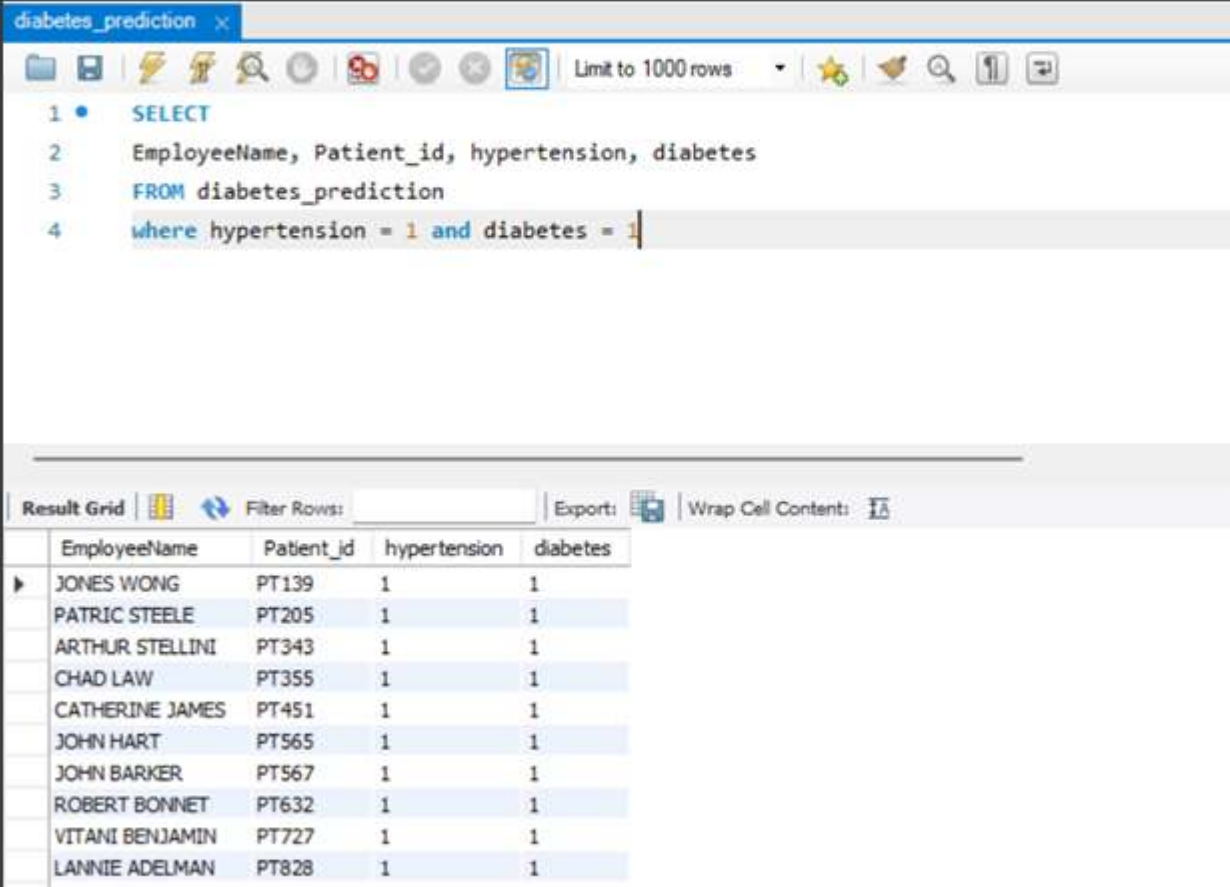
The screenshot shows a database query editor window titled "diabetes_prediction". The query is as follows:

```
1 SELECT
2 EmployeeName, Patient_id, blood_glucose_level
3 FROM diabetes_prediction
4 order by blood_glucose_level desc
```

Below the query editor, the "Result Grid" displays the results of the query. The results are sorted by blood glucose level in descending order. The first row, "PATRICK BRYAN", is highlighted. The table has three columns: EmployeeName, Patient_id, and blood_glucose_level.

EmployeeName	Patient_id	blood_glucose_level
PATRICK BRYAN	PT2658	300
ROBERT DOSS	PT847	300
RONALD BELL	PT2058	300
MARK TRIERWEILER	PT2639	300
DANIEL LUI	PT3675	300
BROCK WELLS	PT2635	300
REX HALE	PT195	300
HENRY SCOBLE JR	PT3528	300
GERALD DARCY	PT243	300
LORI BORGHI	PT300	300
ALI MISAGHI	PT3341	300

Q5. Find patients who have hypertension and diabetes.



The screenshot shows a database query tool interface. The top pane displays a SQL query to select patients with both hypertension and diabetes from the 'diabetes_prediction' table. The bottom pane shows the resulting data grid with 10 rows of patient information.

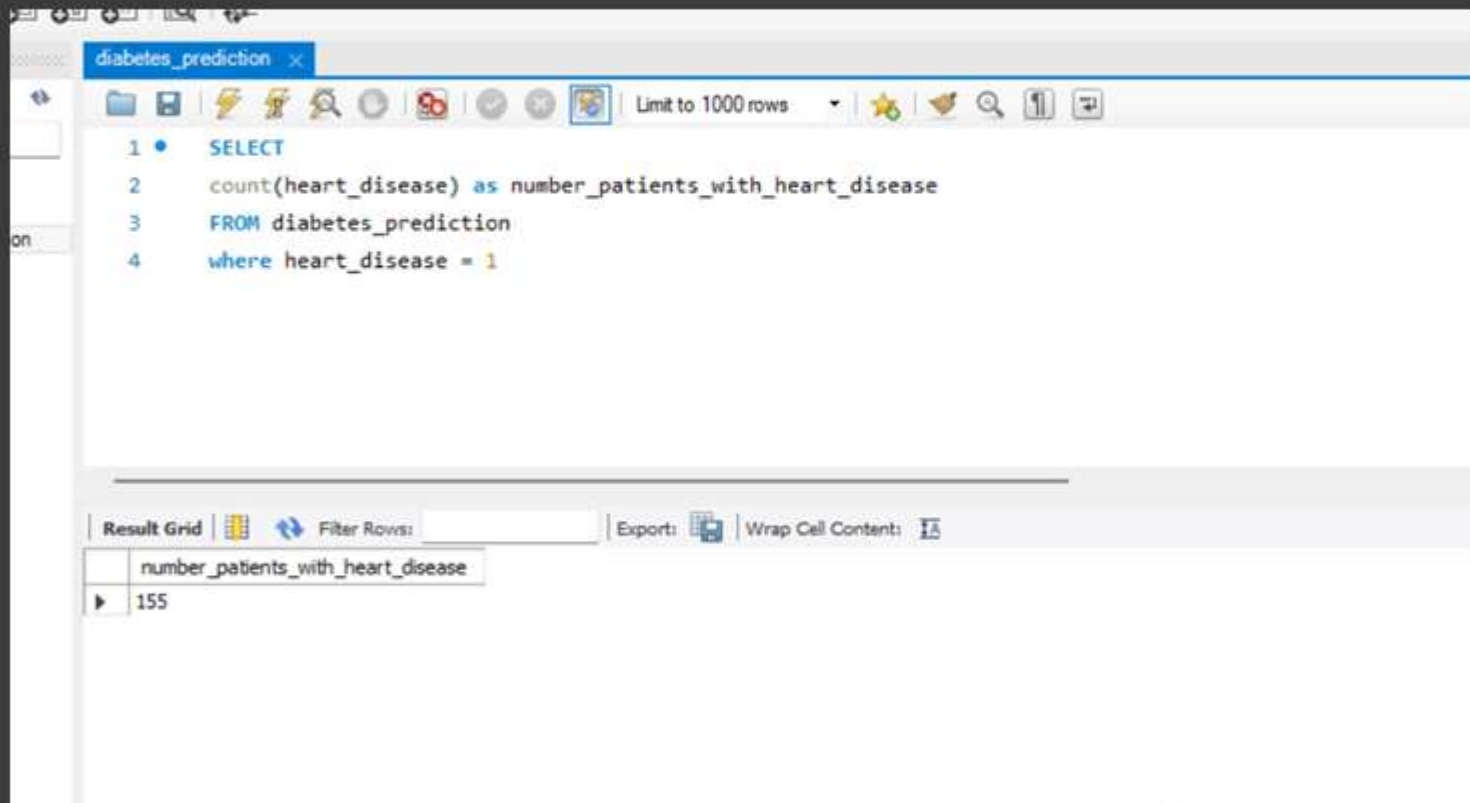
Query:

```
1 • SELECT
2   EmployeeName, Patient_id, hypertension, diabetes
3 FROM diabetes_prediction
4 where hypertension = 1 and diabetes = 1
```

Result Grid:

	EmployeeName	Patient_id	hypertension	diabetes
▶	JONES WONG	PT139	1	1
	PATRIC STEELE	PT205	1	1
	ARTHUR STELLINI	PT343	1	1
	CHAD LAW	PT355	1	1
	CATHERINE JAMES	PT451	1	1
	JOHN HART	PT565	1	1
	JOHN BARKER	PT567	1	1
	ROBERT BONNET	PT632	1	1
	VITANI BENJAMIN	PT727	1	1
	LANNIE ADELMAN	PT828	1	1

Q6. Determine the number of patients with heart disease.



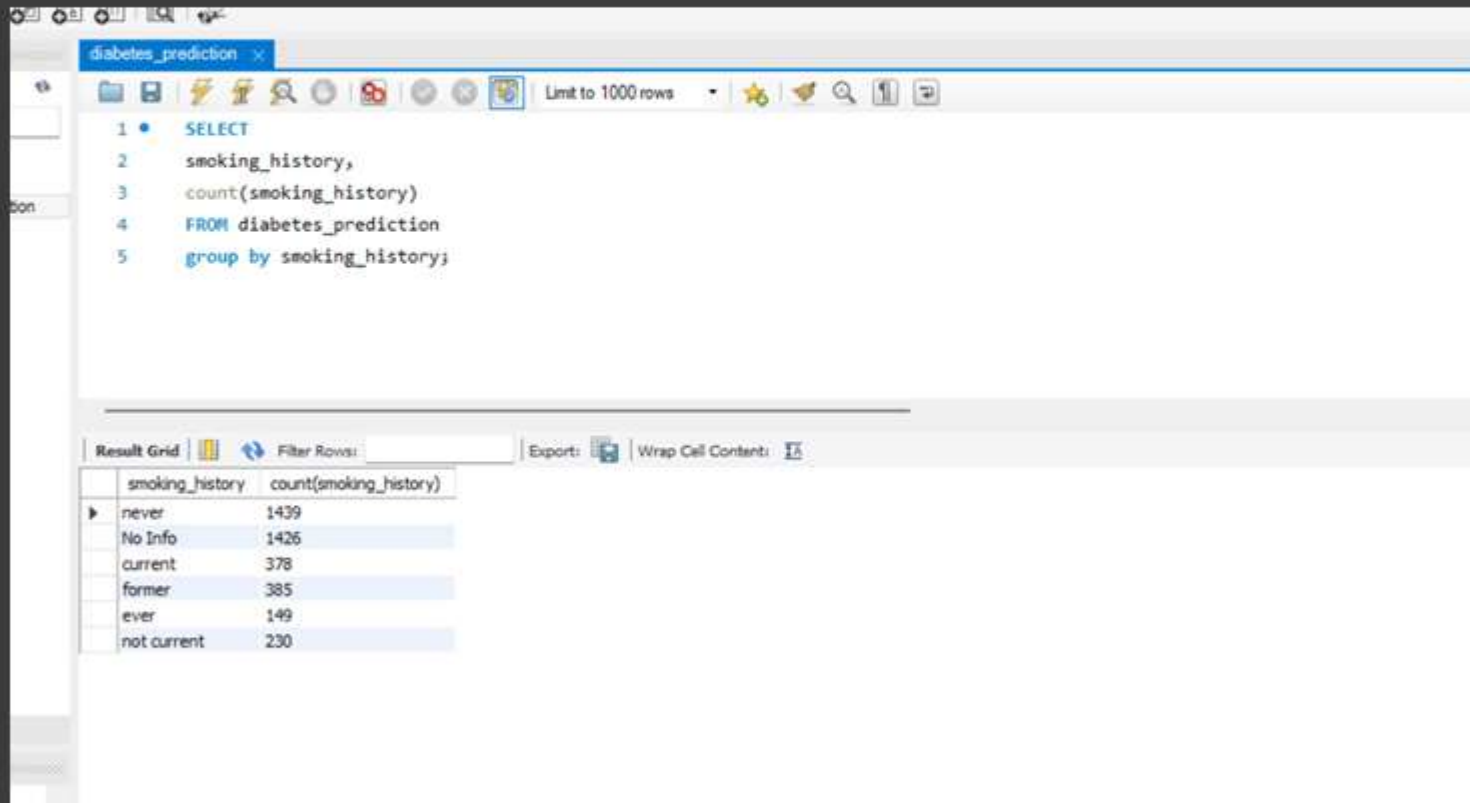
The screenshot shows a SQL query editor window titled "diabetes_prediction". The query is as follows:

```
1 • SELECT
2   count(heart_disease) as number_patients_with_heart_disease
3 FROM diabetes_prediction
4 where heart_disease = 1
```

Below the query editor, the "Result Grid" is displayed. It shows a single column named "number_patients_with_heart_disease" and a single row with the value 155.

number_patients_with_heart_disease
155

Q7. Group patients by smoking history and count how many smokers and nonsmokers there are.



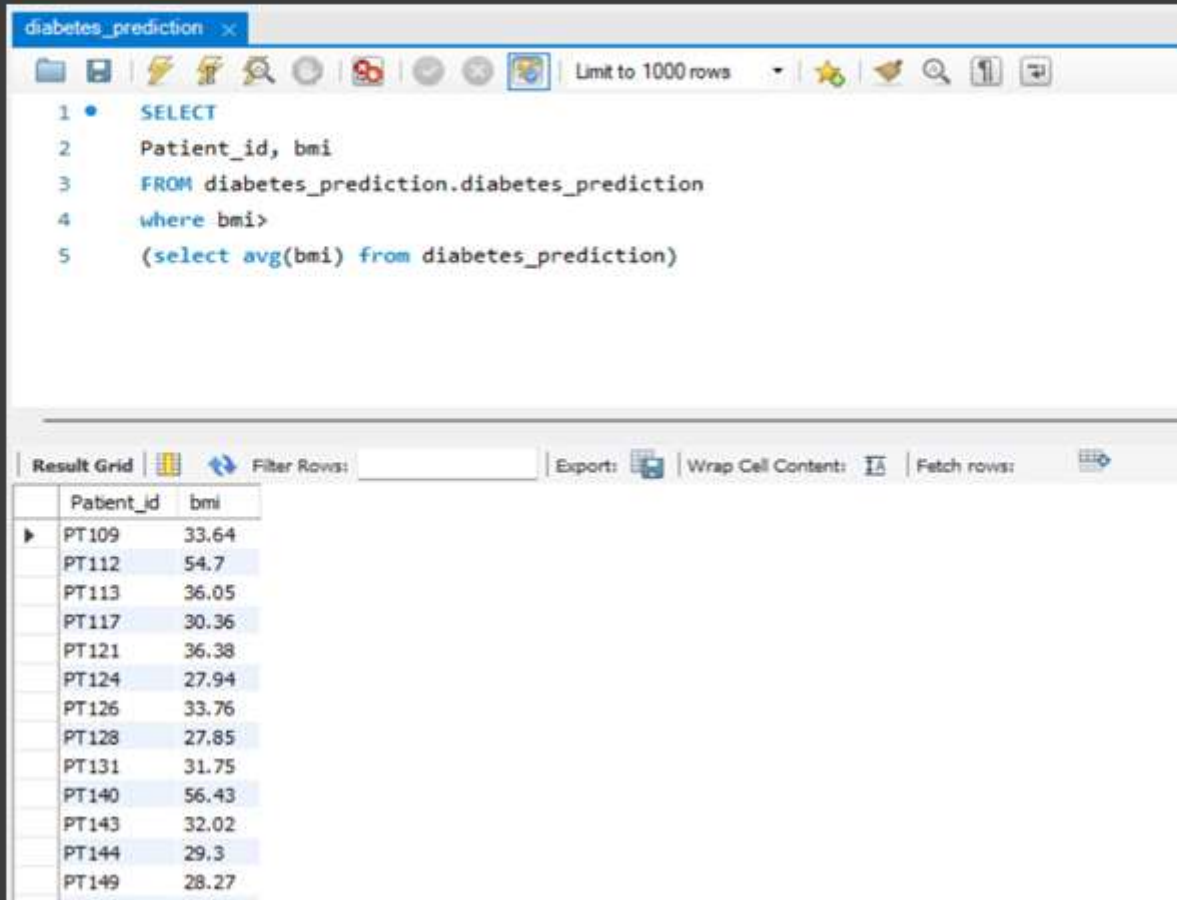
The screenshot shows a SQL query editor window titled "diabetes_prediction". The query is as follows:

```
1 SELECT
2   smoking_history,
3   count(smoking_history)
4 FROM diabetes_prediction
5 GROUP BY smoking_history;
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The results are as follows:

smoking_history	count(smoking_history)
never	1439
No Info	1426
current	378
former	385
ever	149
not current	230

Q8. Retrieve the Patient_ids of patients who have a BMI greater than the average BMI.



The screenshot shows a database query tool interface. The top section displays a SQL query in a text editor. The query is as follows:

```
1 • SELECT
2   Patient_id, bmi
3 FROM diabetes_prediction.diabetes_prediction
4 where bmi >
5   (select avg(bmi) from diabetes_prediction)
```

The bottom section shows the results of the query in a table format. The table has two columns: Patient_id and bmi. The results are as follows:

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
PT140	56.43
PT143	32.02
PT144	29.3
PT149	28.27

Q9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1c level.

● Highest

● Lowest

diabetes_prediction

```

1 SELECT * FROM diabetes_prediction
2 where HbA1c_level = (select max(HbA1c_level)
3                     from diabetes_prediction);
4
5 select * from diabetes_prediction
6 where HbA1c_level = (select min(HbA1c_level)
7                     from diabetes_prediction);

```

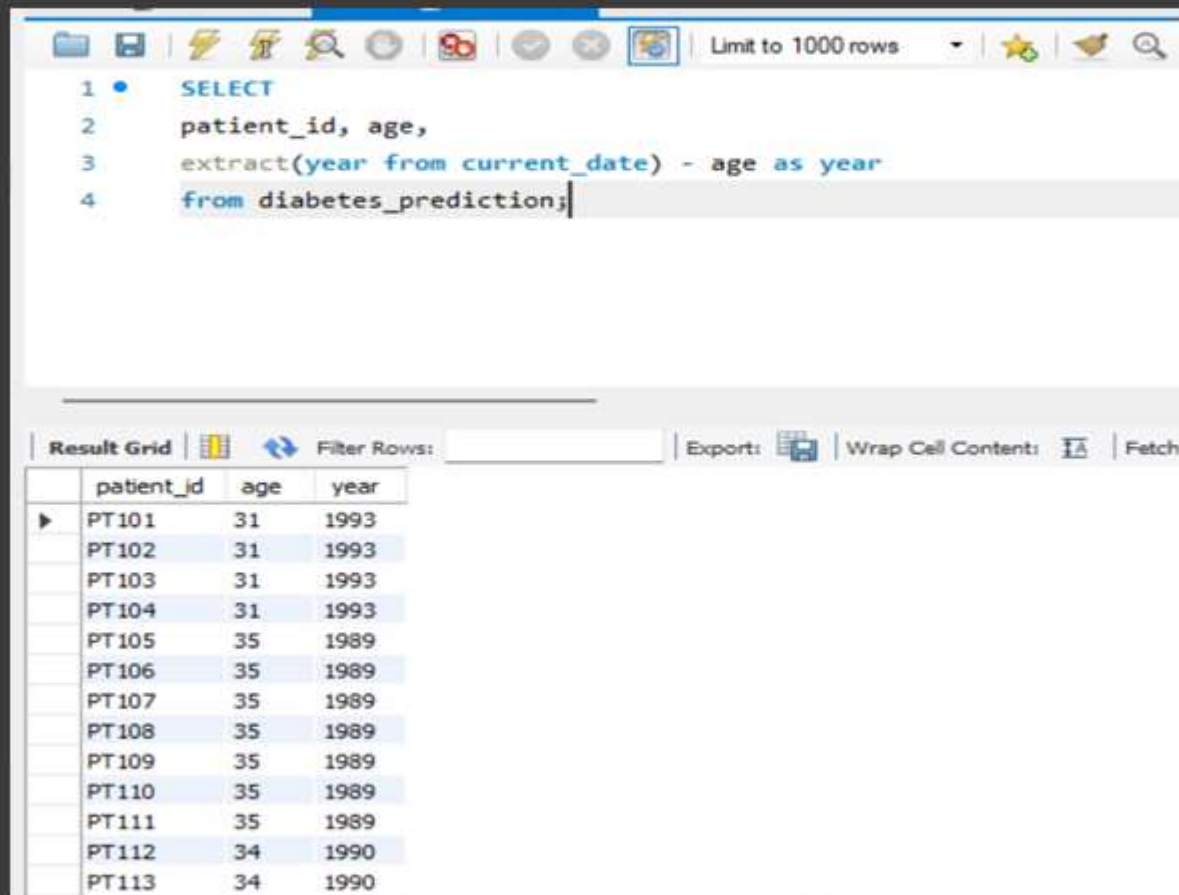
Result Grid

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age
MICHAEL THOMPSON	PT141	Male	8/27/1989	0	0	former	25.51	9	160	1	34
KEVIN CASHMAN	PT136	Male	1/12/1997	0	0	former	37.56	9	159	1	27
MARK CASTAGNOLA	PT236	Male	7/7/1997	0	0	never	22.06	9	155	1	26
WILLIAM SCOTT	PT270	Female	8/4/1997	0	0	not current	39.36	9	140	1	26
JOANNE HOEPE	PT400	Female	10/2/1997	0	0	never	24.81	9	159	1	26
VINCENT PAMPINON	PT519	Female	11/2/1997	0	0	No Info	27.32	9	140	1	26
FRANK KOSTA	PT973	Female	12/13/1997	0	0	never	36.74	9	130	1	26
VINCENT NOLAN	PT710	Female	12/21/1997	0	0	former	31.17	9	260	1	26
KAREN KLEICK	KAREN KLEICK	Male	1/26/1999	1	0	ever	25.94	9	140	1	25
NAVIOUCHER BOGDARPOUR	PT907	Male	1/22/1999	0	0	ever	19.46	9	130	1	25
VICTOR WONG	PT1242	Female	2/3/1999	1	0	never	22.48	9	126	1	25
DANIEL DECOSSIO	PT1319	Male	3/8/1999	1	0	former	22.06	9	200	1	25
MARKUS BONILLA JR	PT1332	Male	5/9/1999	1	0	never	45.7	9	159	1	25
VICTORIA W. DOW	PT1375	Female	5/11/1999	0	1	former	26.69	8	260	1	25

Result Grid

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age
ELLEN MOFFATT	PT120	Male	5/10/1989	0	0	ever	25.72	3.5	159	0	34
JOHN TURSI	PT134	Female	7/24/1989	0	0	never	22.19	3.5	100	0	34
SHARON MCCOLE WIDGER	PT145	Female	10/25/1989	0	0	No Info	27.32	3.5	160	0	34
MARK KEARNEY	PT158	Female	2/1/1997	0	0	never	23.35	3.5	155	0	27
MONIQUE MOYER	PT174	Male	4/22/1997	0	0	not current	27.32	3.5	126	0	26
JOHN HALEY JR.	PT213	Male	6/7/1997	0	0	No Info	27.14	3.5	90	0	26
KHAIRUL ALI	PT219	Female	6/9/1997	0	0	No Info	20.9	3.5	158	0	26
MICHAEL CASTAGNOLA	PT221	Female	6/20/1997	0	0	No Info	27.32	3.5	160	0	26
JOHN RAHAM	PT233	Female	7/4/1997	0	0	No Info	27.32	3.5	200	0	26
PATROCIA CARR	PT250	Female	8/9/1997	0	0	No Info	33.22	3.5	159	0	26
OSCAR CABRERA	PT265	Female	8/2/1997	0	0	No Info	27.7	3.5	159	0	26
ANIPARO RODRIGUEZ	PT269	Female	8/3/1997	0	0	former	35.44	3.5	140	0	26
LEON WHITE	PT279	Male	8/9/1997	0	0	never	25.84	3.5	158	0	26
GRAD GREEN	PT292	Male	8/16/1997	0	0	not current	30.37	3.5	145	0	26
ROSELVIN JEQUINITO	PT304	Male	8/21/1997	0	0	never	23.59	3.5	130	0	26
TROY JOLLIFF	PT305	Female	8/22/1997	0	0	No Info	23.56	3.5	90	0	26

Q10. Calculate the age of patients in years (assuming the current date as of now).



The screenshot shows a SQL query editor interface. The query is as follows:

```
1 • SELECT
2   patient_id, age,
3   extract(year from current_date) - age as year
4   from diabetes_prediction;
```

Below the query editor, the 'Result Grid' is displayed, showing the output of the query. The grid has three columns: 'patient_id', 'age', and 'year'. The data is as follows:

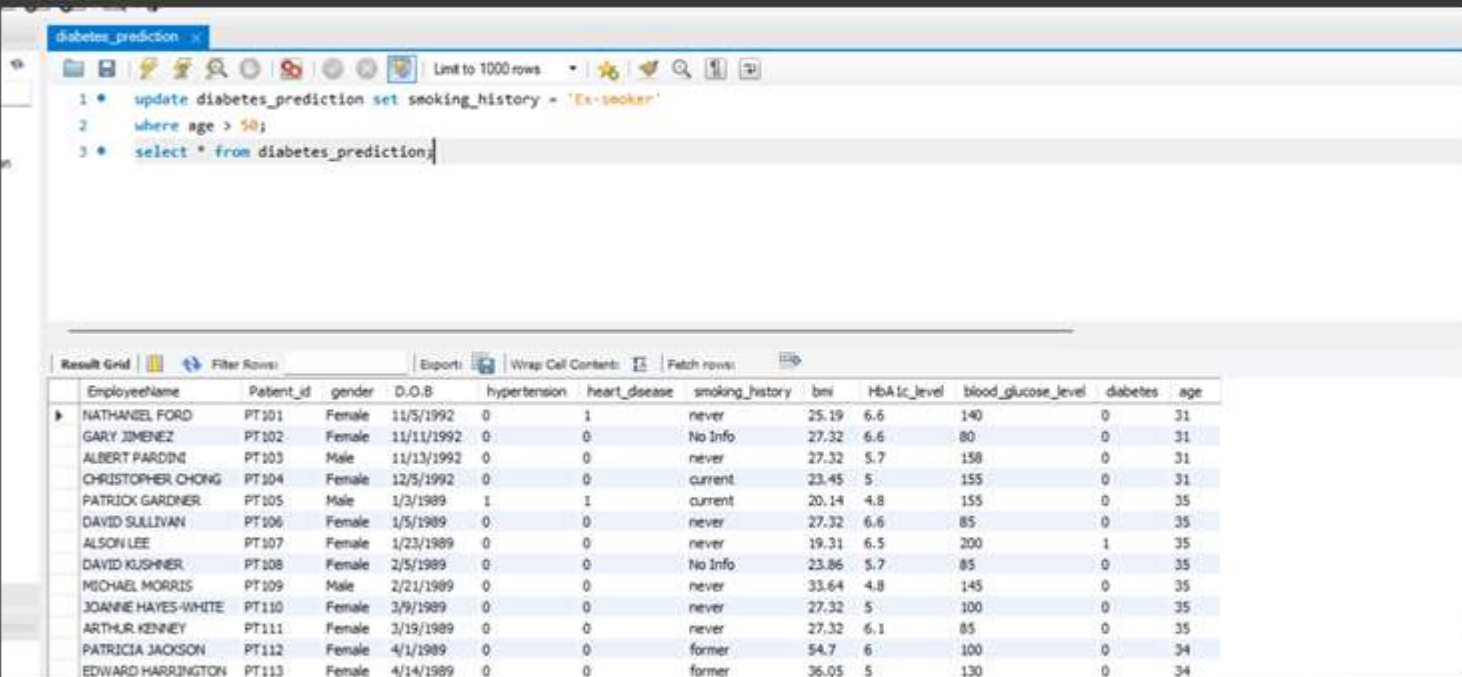
patient_id	age	year
PT101	31	1993
PT102	31	1993
PT103	31	1993
PT104	31	1993
PT105	35	1989
PT106	35	1989
PT107	35	1989
PT108	35	1989
PT109	35	1989
PT110	35	1989
PT111	35	1989
PT112	34	1990
PT113	34	1990

Q11. Rank patients by blood glucose level within each gender group.

```
1 select
2   *, rank() over(partition by gender order by blood_glucose_level) as ranking
3 from diabetes_prediction
```

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age	ranking
CARLOS RECINOS	PT2286	Female	4/16/1999	0	0	No Info	16.3	4.8	80	0	24	1
SANDY CUADRA	PT3938	Female	5/30/1999	0	0	No Info	26.26	6	80	0	24	1
DERMOT DORGAN	PT2420	Female	4/20/1999	0	0	No Info	27.32	5.7	80	0	24	1
JOSEPH ROBLES	PT3909	Female	5/29/1999	0	0	never	22.61	6.6	80	0	24	1
SARAH WILNER	PT2402	Female	4/19/1999	0	0	No Info	16.7	5.7	80	0	24	1
LORI CADIGAN	PT2235	Female	4/15/1999	1	0	never	28.62	6.1	80	0	24	1
SANDAI NATH	PT2952	Female	5/3/1999	0	0	never	18.38	5.8	80	0	24	1
JERRY TIDWELL	PT3697	Female	5/24/1999	0	0	No Info	22.58	6	80	0	24	1
MICHELLE JEAN	PT2365	Female	4/18/1999	0	0	ever	35.83	6.6	80	0	24	1
STEPHANIE STUART	PT3432	Female	5/18/1999	0	0	former	27.74	5.7	80	0	24	1
TAIRA DE BERNARDI	PT3672	Female	5/23/1999	0	0	No Info	27.32	6.6	80	0	24	1

Q12. Update the smoking history of patients who are older than 50 to "Ex-smoker."



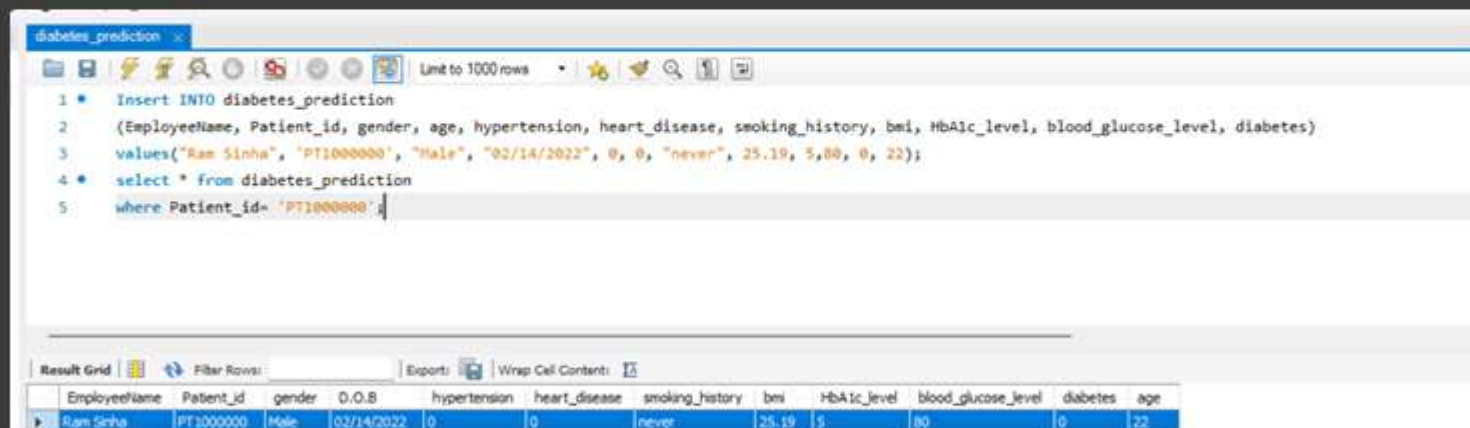
The screenshot shows a database management interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 • update diabetes_prediction set smoking_history = 'Ex-smoker';
2 • where age > 50;
3 • select * from diabetes_predictions;
```

The result grid displays the following data:

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age
NATHANIEL FORD	PT101	Female	11/5/1992	0	1	never	25.19	6.6	140	0	31
GARY JIMENEZ	PT102	Female	11/11/1992	0	0	No Info	27.32	6.6	80	0	31
ALBERT PARDINI	PT103	Male	11/13/1992	0	0	never	27.32	5.7	158	0	31
CHRISTOPHER CHONG	PT104	Female	12/5/1992	0	0	current	23.45	5	155	0	31
PATRICK GARDNER	PT105	Male	1/3/1989	1	1	current	20.14	4.8	155	0	35
DAVID SULLIVAN	PT106	Female	1/5/1989	0	0	never	27.32	6.6	85	0	35
ALSON LEE	PT107	Female	1/23/1989	0	0	never	19.31	6.5	200	1	35
DAVID KUSHNER	PT108	Female	2/5/1989	0	0	No Info	23.86	5.7	85	0	35
MICHAEL MORRIS	PT109	Male	2/21/1989	0	0	never	33.64	4.8	145	0	35
JOANNE HAYES-WHITE	PT110	Female	3/9/1989	0	0	never	27.32	5	100	0	35
ARTHUR KENNEY	PT111	Female	3/19/1989	0	0	never	27.32	6.1	85	0	35
PATRICIA JACKSON	PT112	Female	4/1/1989	0	0	former	54.7	6	100	0	34
EDWARD HARRINGTON	PT113	Female	4/14/1989	0	0	former	36.05	5	130	0	34

Q13. Insert a new patient into the database with sample data.



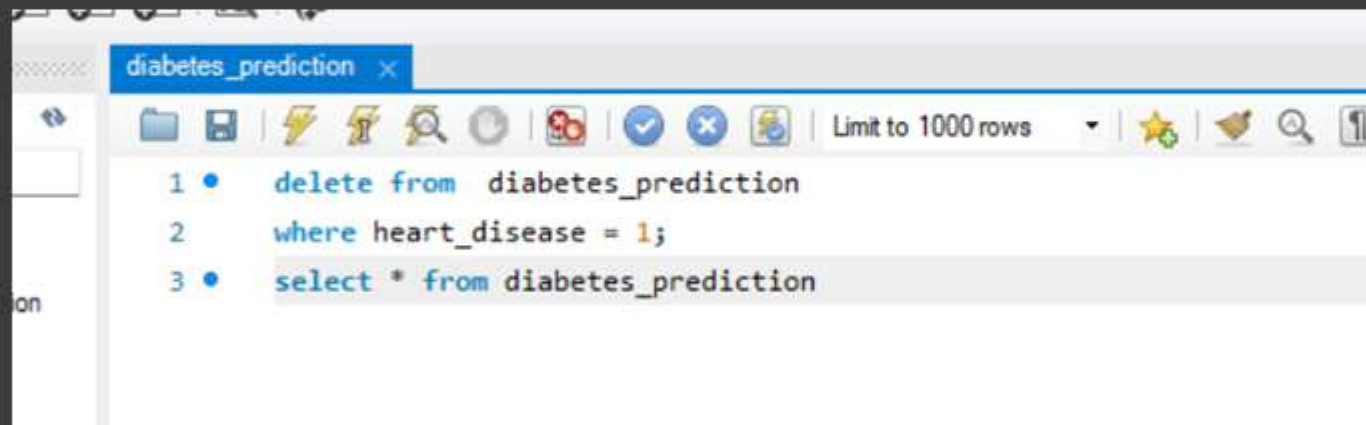
The screenshot shows a database management interface with a tab labeled "diabetes_prediction". The SQL editor contains the following code:

```
1 * Insert INTO diabetes_prediction
2 (EmployeeName, Patient_id, gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level, diabetes)
3 values('Ram Sinha', 'PT1000000', 'Male', '02/14/2022', 0, 0, 'never', 25.19, 5.80, 0, 22);
4 * select * from diabetes_prediction
5 where Patient_id= 'PT1000000';
```

Below the editor, the "Result Grid" shows the output of the query. It includes a toolbar with "Filter Rows", "Export", and "Wrap Cell Contents" options. The result is a single row of data:

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age
Ram Sinha	PT1000000	Male	02/14/2022	0	0	never	25.19	5.80	0	0	22

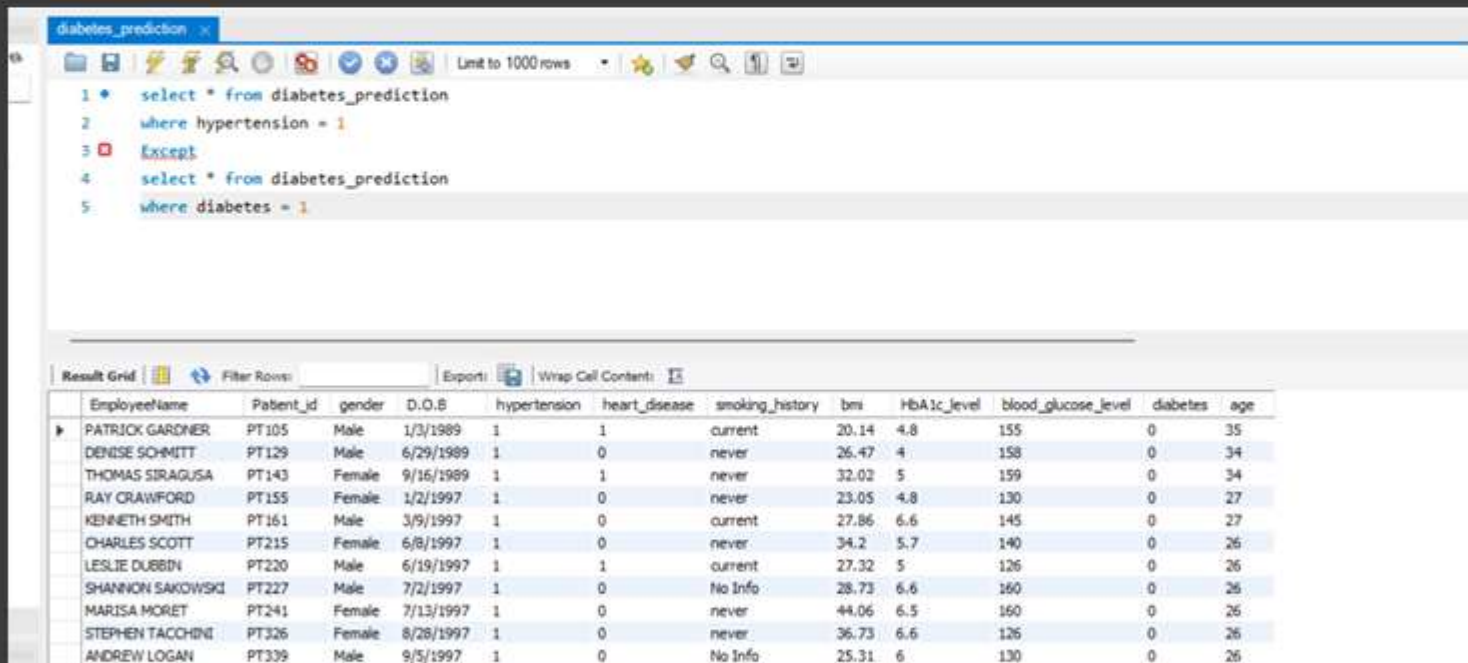
Q14. Delete all patients with heart disease from the database.



The screenshot shows a database management interface with a tab labeled "diabetes_prediction". The interface includes a toolbar with various icons for file operations, execution, and search. Below the toolbar, three SQL queries are listed, each preceded by a blue bullet point and a line number:

```
1 • delete from diabetes_prediction
2   where heart_disease = 1;
3 • select * from diabetes_prediction
```

Q15. Find patients who have hypertension but not diabetes using the EXCEPT operator.



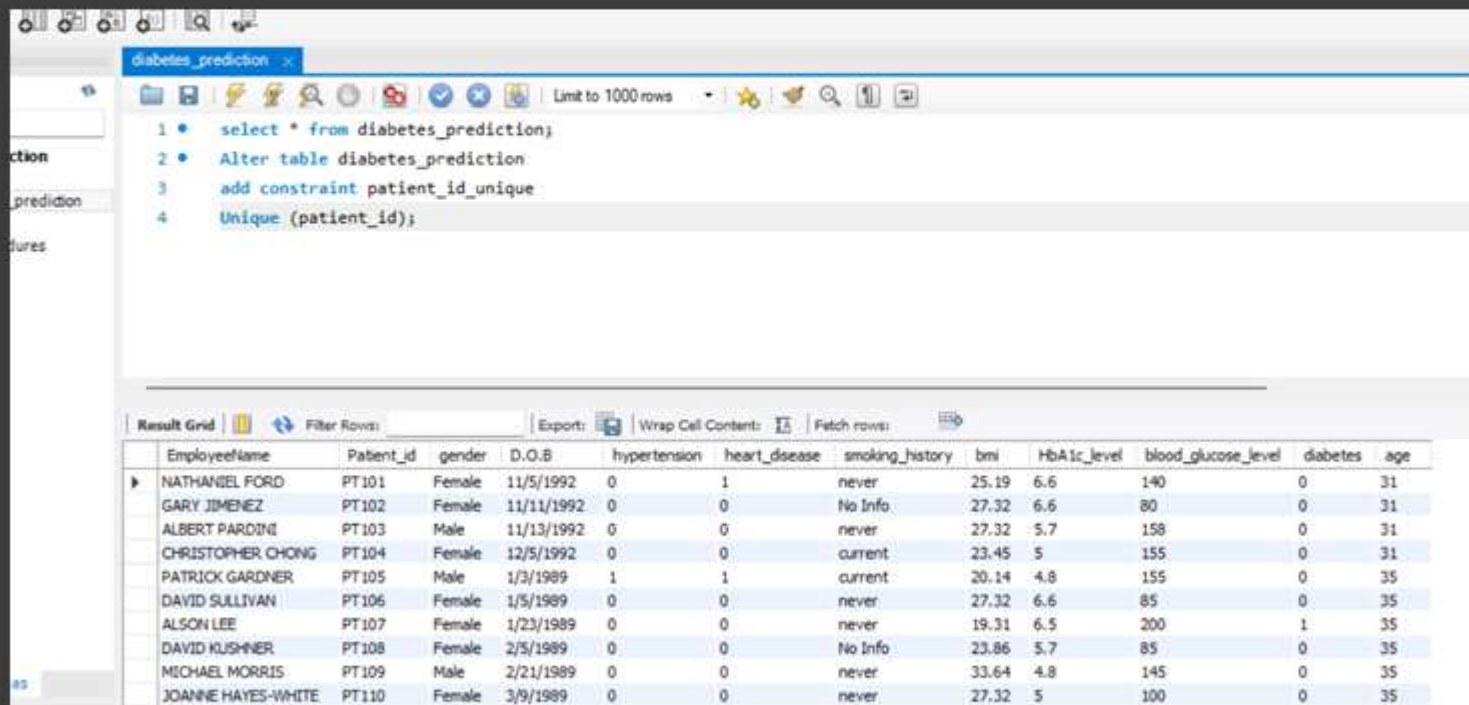
The screenshot shows a database query editor window titled "diabetes_prediction". The query is as follows:

```
1 select * from diabetes_prediction
2 where hypertension = 1
3 except
4 select * from diabetes_prediction
5 where diabetes = 1
```

Below the query editor is a "Result Grid" showing 12 rows of patient data. The columns are: EmployeeName, Patient_id, gender, D.O.B, hypertension, heart_disease, smoking_history, bms, HbA1c_level, blood_glucose_level, diabetes, and age. The data is as follows:

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bms	HbA1c_level	blood_glucose_level	diabetes	age
PATRICK GARDNER	PT105	Male	1/3/1989	1	1	current	20.14	4.8	155	0	35
DENISE SCHMITT	PT129	Male	6/29/1989	1	0	never	26.47	4	158	0	34
THOMAS SIRAGUSA	PT143	Female	9/16/1989	1	1	never	32.02	5	159	0	34
RAY CRAWFORD	PT155	Female	1/2/1997	1	0	never	23.05	4.8	130	0	27
KENNETH SMITH	PT161	Male	3/9/1997	1	0	current	27.86	6.6	145	0	27
CHARLES SCOTT	PT215	Female	6/8/1997	1	0	never	34.2	5.7	140	0	26
LESLIE DUBBIN	PT220	Male	6/19/1997	1	1	current	27.32	5	126	0	26
SHANNON SAKOWSKI	PT227	Male	7/2/1997	1	0	No Info	28.73	6.6	160	0	26
MARISA MORET	PT241	Female	7/13/1997	1	0	never	44.06	6.5	160	0	26
STEPHEN TACCHINI	PT326	Female	8/28/1997	1	0	never	36.73	6.6	126	0	26
ANDREW LOGAN	PT339	Male	9/5/1997	1	0	No Info	25.31	6	130	0	26

Q16. Define a unique constraint on the "patient_id" column to ensure its values are unique.



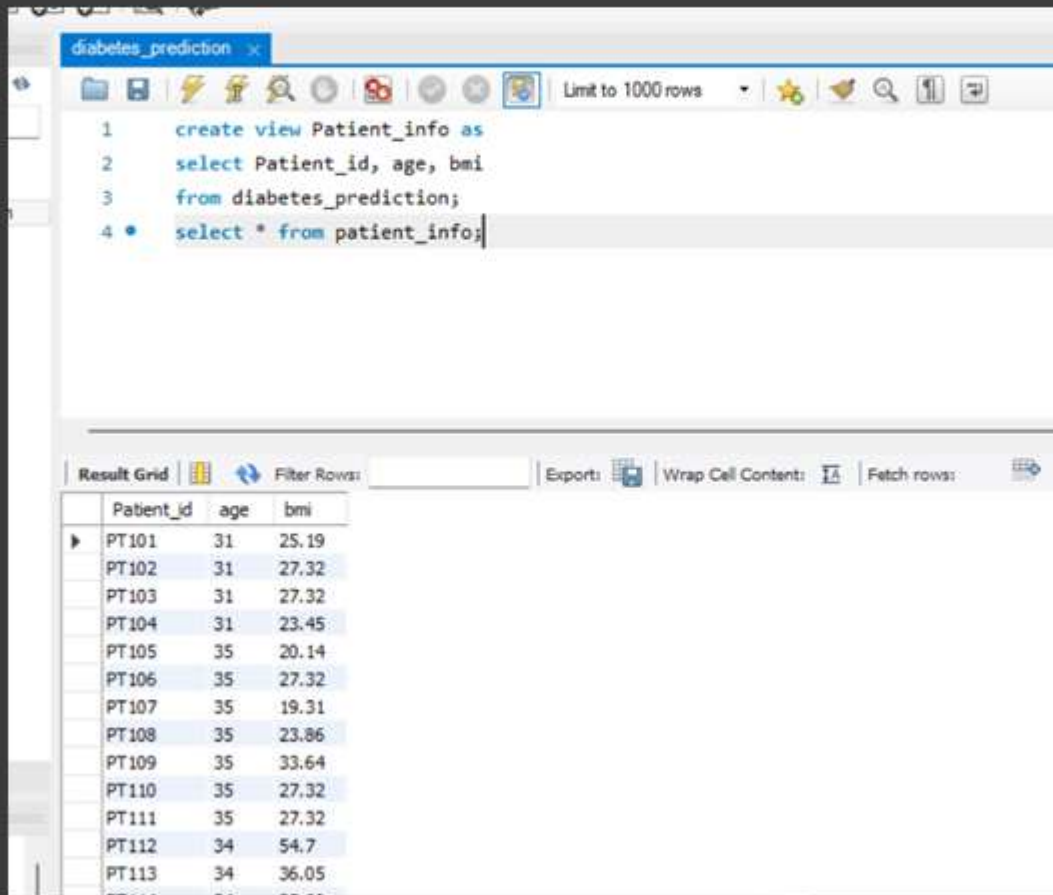
The screenshot shows a database management interface with a SQL editor and a result grid. The SQL editor contains the following commands:

```
1 • select * from diabetes_prediction;  
2 • Alter table diabetes_prediction  
3   add constraint patient_id_unique  
4   Unique (patient_id);
```

The result grid displays 12 columns: EmployeeName, Patient_id, gender, D.O.B, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level, diabetes, and age. It contains 11 rows of data:

EmployeeName	Patient_id	gender	D.O.B	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	age
NATHANIEL FORD	PT101	Female	11/5/1992	0	1	never	25.19	6.6	140	0	31
GARY JIMENEZ	PT102	Female	11/11/1992	0	0	No Info	27.32	6.6	80	0	31
ALBERT PARDINI	PT103	Male	11/13/1992	0	0	never	27.32	5.7	158	0	31
CHRISTOPHER CHONG	PT104	Female	12/5/1992	0	0	current	23.45	5	155	0	31
PATRICK GARDNER	PT105	Male	1/3/1989	1	1	current	20.14	4.8	155	0	35
DAVID SULLIVAN	PT106	Female	1/5/1989	0	0	never	27.32	6.6	85	0	35
ALSON LEE	PT107	Female	1/23/1989	0	0	never	19.31	6.5	200	1	35
DAVID KUSHNER	PT108	Female	2/5/1989	0	0	No Info	23.86	5.7	85	0	35
MICHAEL MORRIS	PT109	Male	2/21/1989	0	0	never	33.64	4.8	145	0	35
JOANNE HAYES-WHITE	PT110	Female	3/9/1989	0	0	never	27.32	5	100	0	35

Q17. Create a view that displays the Patient_ids, ages, and BMI of patients.



The screenshot shows a database management interface with a tab labeled "diabetes_prediction". The SQL editor contains the following code:

```
1 create view Patient_info as
2 select Patient_id, age, bmi
3 from diabetes_prediction;
4 select * from patient_info;
```

Below the editor is a "Result Grid" showing the output of the query. The grid has columns for Patient_id, age, and bmi. The data is as follows:

Patient_id	age	bmi
PT101	31	25.19
PT102	31	27.32
PT103	31	27.32
PT104	31	23.45
PT105	35	20.14
PT106	35	27.32
PT107	35	19.31
PT108	35	23.86
PT109	35	33.64
PT110	35	27.32
PT111	35	27.32
PT112	34	54.7
PT113	34	36.05

Q18. Suggest improvements in the database schema to reduce data redundancy and improve data integrity.

1. **Normalization (Up to 3NF):** Organize data into separate tables based on functional dependencies to eliminate redundancy and ensure data integrity.
2. **Primary Keys and Foreign Keys:** Use primary keys to uniquely identify rows in each table and foreign keys to establish relationships between tables, enforcing referential integrity.
3. **Avoid Multi-Valued Dependencies:** Identify and eliminate multi-valued dependencies by decomposing tables or creating separate tables for related attributes to maintain atomicity and data integrity.
4. **Denormalization for Performance:** Consider denormalizing tables or introducing calculated fields to improve query performance, but carefully balance performance gains with the risk of data inconsistency.

Q19. Explain how you can optimize the performance of SQL queries on this dataset.

1. **Indexing:** Use indexes on columns frequently used in WHERE clauses, JOIN conditions, and ORDER BY clauses to expedite row retrieval.
2. **Optimize Joins:** Prefer INNER JOIN over OUTER JOIN for efficiency. Prioritize smaller tables as driving tables in the join clause.
3. **Avoid SELECT :** Explicitly specify needed columns instead of using SELECT *, reducing data retrieval overhead.
4. **Limit Result Set:** Apply LIMIT clause to restrict the number of rows returned, minimizing data transfer and processing.