



PSYLIQ

DIABETES PREDICTION

ARFA AFSHEEN

Retrieve the Patient_id and ages of all patients.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' and 'psyliq_project' expanded. Under 'psyliq_project', the 'diabetes_prediction_csv' table is selected. The main editor window shows a SQL query in 'SQL File 3*':

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

Below the query editor, the 'Result Grid' tab is active, displaying the query results in a table format. The table has two columns: 'Patient_id' and 'age'. The results are as follows:

Patient_id	age
PT101	80
PT102	54
PT103	28
PT104	36
PT105	76
PT106	20
PT107	44
PT108	79
PT109	42
PT110	32
PT111	53
PT112	54

The bottom status bar indicates 'diabetes_prediction_csv 2 x' and 'Read Only'.

Select all female patients who are older than 40

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the Schemas tree with a search filter. The 'psysiq_project' database is selected, and the 'diabetes_prediction_csv' table is highlighted. The main editor displays a SQL query: `SELECT * FROM diabetes_prediction_csv WHERE gender='Female' and age>40;`. The bottom panel shows the Result Grid with 12 columns: EmployeeName, Patient_id, gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level, and diabetes. The results list 12 female patients aged 42 to 80.

Local instance MySQL82 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 3*

Limit to 1000 rows

```
1 • SELECT *
2 FROM diabetes_prediction_csv
3 WHERE gender='Female' and age>40;
```

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

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
NATHANIEL FORD	PT101	Female	80	0	1	never	25.19	6.6	140	0
GARY JIMENEZ	PT102	Female	54	0	0	No Info	27.32	6.6	80	0
ALSON LEE	PT107	Female	44	0	0	never	19.31	6.5	200	1
DAVID KUSHNER	PT108	Female	79	0	0	No Info	23.86	5.7	85	0
ARTHUR KENNEY	PT111	Female	53	0	0	never	27.32	6.1	85	0
PATRICIA JACKSON	PT112	Female	54	0	0	former	54.7	6	100	0
EDWARD HARRINGTON	PT113	Female	78	0	0	former	36.05	5	130	0
JOHN MARTIN	PT114	Female	67	0	0	never	25.69	5.8	200	0
DAVID FRANKLIN	PT115	Female	76	0	0	No Info	27.32	5	160	0
SEBASTIAN WONG	PT118	Female	42	0	0	never	24.48	5.7	158	0
MARTY ROSS	PT119	Female	42	0	0	No Info	27.32	5.7	80	0
GEORGE GARCIA	PT123	Female	69	0	0	never	21.24	4.8	85	0

Table: diabetes_prediction_csv

Columns: diabetes_prediction_csv 3

Calculate the average BMI of patients

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with the 'diabetes_prediction_csv' table selected under the 'psyliq_project' database. The main query editor contains the following SQL code:

```
1 • SELECT AVG(bmi) as Average_bmi
2 FROM diabetes_prediction_csv;
```

The 'Result Grid' at the bottom shows the execution results:

Average_bmi
27.310304509944558

At the bottom left, the 'Table: diabetes_prediction_csv' is listed.

List patients in descending order of blood glucose levels

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' and 'psylq_project' databases. Under 'psylq_project', the 'diabetes_prediction_csv' table is selected. The main editor shows a SQL query: `SELECT * FROM diabetes_prediction_csv ORDER BY blood_glucose_level DESC;`. The bottom panel shows the 'Result Grid' with 12 columns: EmployeeName, Patient_id, gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level, and diabetes. The results are sorted by 'blood_glucose_level' in descending order.

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
EGON STEIN	PT11119	Female	75	0	1	former	39.84	6.2	300	1
WILLIE CRAWFORD	PT9019	Male	80	1	1	never	23.24	6.6	300	1
ALBERT MAI	PT9011	Male	77	0	0	former	23.14	9	300	1
JAMES CERENIO	PT8250	Female	62	0	0	never	34.24	8.2	300	1
DAVID CHAN	PT9280	Male	41	0	0	No Info	27.32	7	300	1
ARTURO FARO	PT9741	Female	17	0	0	No Info	31.68	6.8	300	1
HOWARD CONROY	PT8894	Male	42	0	0	never	49.2	9	300	1
STEPHEN BARRETT	PT8782	Male	72	0	0	current	35.08	5.7	300	1
LESLIE LEVITAS-MARTIN	PT8481	Female	53	0	0	current	27.32	6.1	300	1
BORIS DELEPINE	PT8497	Male	72	0	0	No Info	26.31	6.5	300	1
TERRENCE HONG	PT8557	Female	80	0	0	former	32.05	9	300	1
GREGORY BAILEY	PT9398	Male	60	0	0	former	33.04	5.8	300	1

Find patients who have hypertension and diabetes

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' expanded, showing 'diabetes_prediction_csv' under 'Tables'. The main editor window shows a SQL query in 'SQL File 3*':

```
1 • SELECT EmployeeName,patient_id,gender,age,hypertension,diabetes
2 FROM diabetes_prediction_csv
3 WHERE diabetes=1 and hypertension=1;
```

Below the query editor, the 'Result Grid' shows the results of the query. The table has 7 columns: EmployeeName, patient_id, gender, age, hypertension, and diabetes. The results are as follows:

EmployeeName	patient_id	gender	age	hypertension	diabetes
JONES WONG	PT139	Male	50	1	1
PATRIC STEELE	PT205	Female	80	1	1
ARTHUR STELLINI	PT343	Male	57	1	1
CHAD LAW	PT355	Male	63	1	1
CATHERINE JAMES	PT451	Female	52	1	1
JOHN HART	PT565	Male	48	1	1
JOHN BARKER	PT567	Female	79	1	1
ROBERT BONNET	PT632	Female	49	1	1
VITANI BENJAMIN	PT727	Male	43	1	1
LANNIE ADELMAN	PT828	Female	38	1	1
JOEL DELIZONNA	PT852	Female	28	1	1
KAREN KUBICK	PT861	Male	59	1	1

Determine the number of patients with heart disease

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' and 'psyllq_project' (containing 'diabetes_prediction_csv'). The main editor shows a SQL query in 'SQL File 3*' that counts patients with heart disease. The 'Result Grid' at the bottom shows the query result.

Local instance MySQL82 - W...

File Edit View Query Database Server Tools Scripting Help

Navigator: SQL File 3*

Limit to 1000 rows

```
1 • SELECT COUNT(patient_id) as number_of_patients_with_heart_disease
2 FROM diabetes_prediction_csv
3 WHERE heart_disease=1;
```

Result Grid

number_of_patients_with_heart_disease
417

Administration Schemas

Information

diabetes_prediction_csv

Columns: EmployeeName text

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

SQL File 3* x

Limit to 1000 rows

```
1 • SELECT smoking_history,COUNT(Patient_id) as count_of_patient
2 FROM diabetes_prediction_csv
3 GROUP BY smoking_history
```

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

	smoking_history	count_of_patient
▶	never	3994
	No Info	3983
	current	1045
	former	1077
	ever	456
	not current	709

Result 12 x

Read Only

Result Grid
Form Editor
Field Types

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' and 'psyliq_project' expanded. Under 'psyliq_project', the 'diabetes_prediction_csv' table is selected. The main editor window shows a SQL query in 'SQL File 3*':

```
1 • SELECT patient_id,bmi
2 FROM diabetes_prediction_csv
3 WHERE bmi >
4      (SELECT AVG(bmi) FROM diabetes_prediction_csv);
```

The 'Result Grid' at the bottom shows the results of the query, displaying columns 'patient_id' and 'bmi'. The results are as follows:

patient_id	bmi
PT102	27.32
PT103	27.32
PT106	27.32
PT109	33.64
PT110	27.32
PT111	27.32
PT112	54.7
PT113	36.05
PT115	27.32
PT116	27.32
PT117	30.36
PT119	27.32

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'my_db' and 'psyliq_project' expanded. Under 'psyliq_project', the 'Tables' folder is expanded, showing 'diabetes_prediction_csv'. The main editor window, titled 'SQL File 3*', contains the following SQL query:

```
1 • SELECT MAX(HbA1c_level) AS max_HbA1c_level, MIN(HbA1c_level) as min_HbA1c_level
2 FROM diabetes_prediction_csv;
```

Below the query editor, the 'Result Grid' tab is active, displaying the query results in a table:

	max_HbA1c_level	min_HbA1c_level
▶ 9		3.5

The interface also includes a menu bar (File, Edit, View, Query, Database, Server, Tools, Scripting, Help), a toolbar with various icons, and a status bar at the bottom with options like 'Filter Rows', 'Export', and 'Wrap Cell Content'.

Calculate the age of patients in years

The screenshot shows the MySQL Workbench interface. The 'Navigator' pane on the left displays the database structure, including the 'diabetes_prediction_csv' table. The 'SQL File 3*' editor in the center contains the following query:

```
1 • SELECT patient_id, age
2 FROM diabetes_prediction_csv;
3
4
5
```

The 'Result Grid' at the bottom displays the query results in a table format. The table has two columns: 'patient_id' and 'age'. The results are as follows:

patient_id	age
PT101	80
PT102	54
PT103	28
PT104	36
PT105	76
PT106	20
PT107	44
PT108	79
PT109	42
PT110	32
PT111	53
PT112	54

Rank patients by blood glucose level within each gender group

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'my_db' and 'psylliq_project'. Under 'psylliq_project', there is a table named 'diabetes_prediction_csv'. The main editor window displays a SQL query in 'SQL File 3*' that uses the `dense_rank()` function to rank patients by blood glucose level within each gender group. The query is as follows:

```
1 SELECT patient_id,gender,blood_glucose_level,  
2 dense_rank() OVER(PARTITION BY gender ORDER BY blood_glucose_level DESC) as rank_of_blood_glucose  
3 FROM diabetes_prediction_csv;  
4  
5  
6
```

Below the query editor, the 'Result Grid' shows the output of the query. The columns are 'patient_id', 'gender', 'blood_glucose_level', and 'rank_of_blood_glucose'. The results show 15 rows, all with a rank of 1, indicating that all patients in the dataset have the same blood glucose level of 300.

patient_id	gender	blood_glucose_level	rank_of_blood_glucose
PT9741	Female	300	1
PT11119	Female	300	1
PT2809	Female	300	1
PT1461	Female	300	1
PT2658	Female	300	1
PT1838	Female	300	1
PT2635	Female	300	1
PT4197	Female	300	1
PT2639	Female	300	1
PT243	Female	300	1
PT2417	Female	300	1
PT2662	Female	300	1

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

The screenshot shows the MySQL Workbench interface. The top menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The left sidebar shows the 'SCHEMAS' tree with a search filter 'Filter objects'. Under 'my_db', the 'psyliq_project' database is expanded, showing 'Tables' and 'diabetes_prediction_csv'. The 'diabetes_prediction_csv' table is selected, showing its structure (Columns, Indexes, Foreign Keys, Triggers, Views, Stored Procedures, Functions). The bottom left shows 'Administration' and 'Schemas' tabs, with 'Schemas' selected. The bottom left also shows 'Information' and 'Table: diabetes_prediction_csv'.

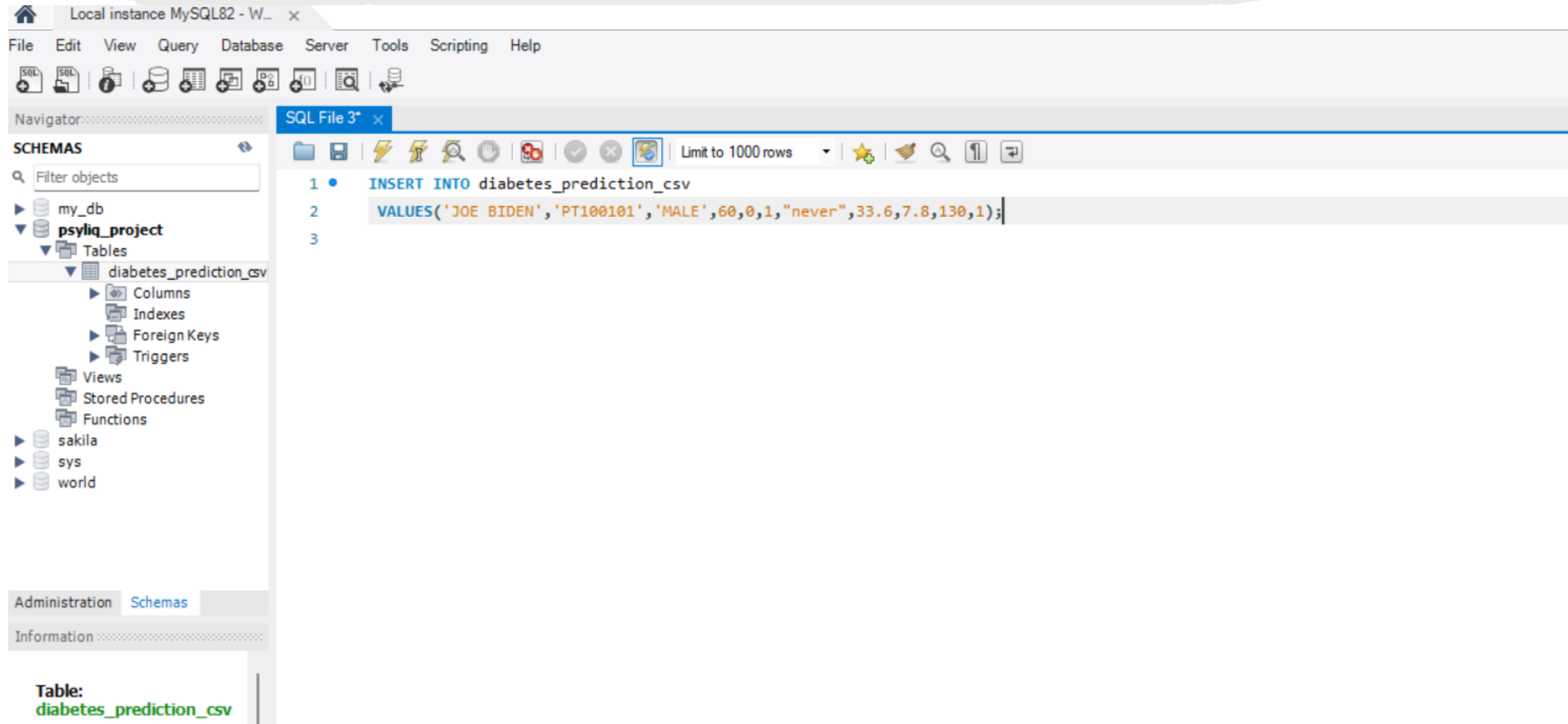
The main window displays a SQL query in the 'SQL File 3*' tab. The query is:

```
1 • UPDATE diabetes_prediction_csv
2   SET smoking_history="Ex-Smoker" WHERE age>50 LIMIT 100;
3
4 • SELECT patient_id,smoking_history,age
5   FROM diabetes_prediction_csv;
6
```

The 'Result Grid' shows the results of the query. The columns are 'patient_id', 'smoking_history', and 'age'. The results are as follows:

patient_id	smoking_history	age
PT101	Ex-Smoker	80
PT102	Ex-Smoker	54
PT103	never	28
PT104	current	36
PT105	Ex-Smoker	76
PT106	never	20
PT107	never	44
PT108	Ex-Smoker	79
PT109	never	42
PT110	never	32
PT111	Ex-Smoker	53
PT112	Ex-Smoker	54

Insert a new patient into the database with sample data

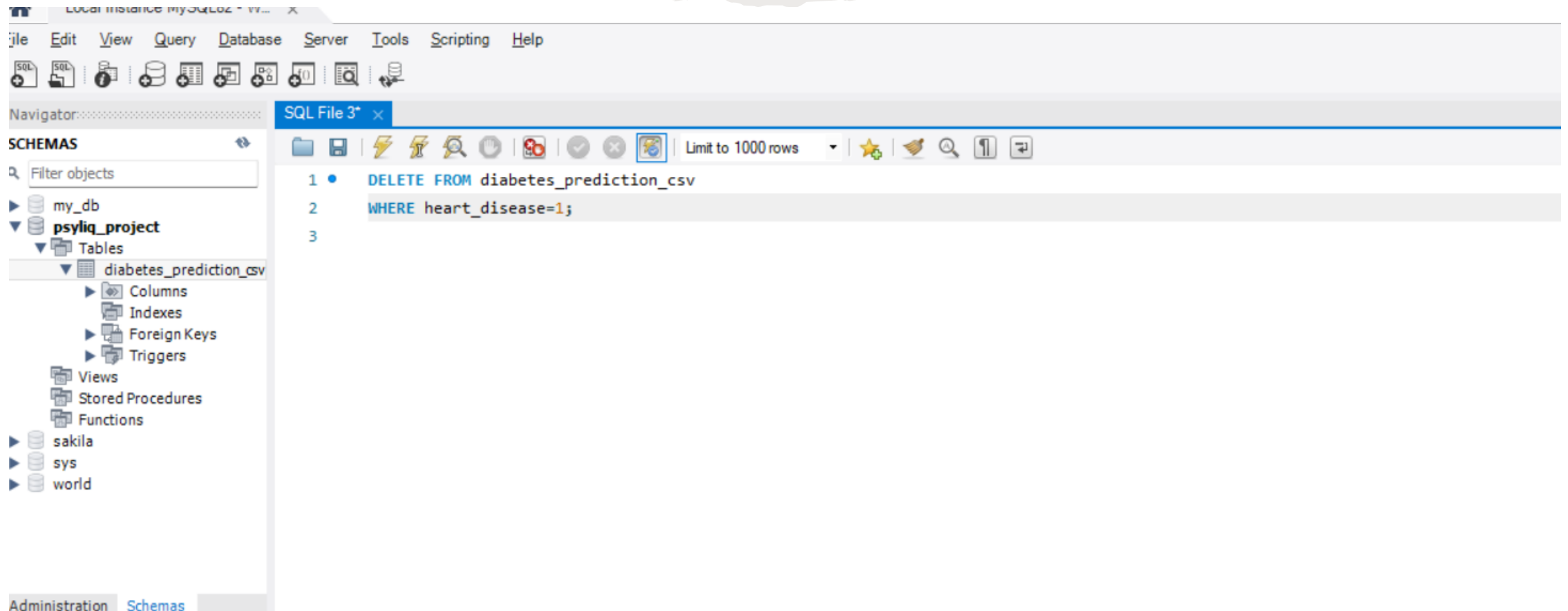


The screenshot shows the MySQL Workbench interface. The title bar indicates the connection is to a 'Local instance MySQL82 - W...'. The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The Navigator pane on the left shows the 'SCHEMAS' tree with a search filter. Under the 'psyliq_project' schema, the 'diabetes_prediction_csv' table is selected. The main editor pane shows an SQL query in 'SQL File 3*' with the following code:

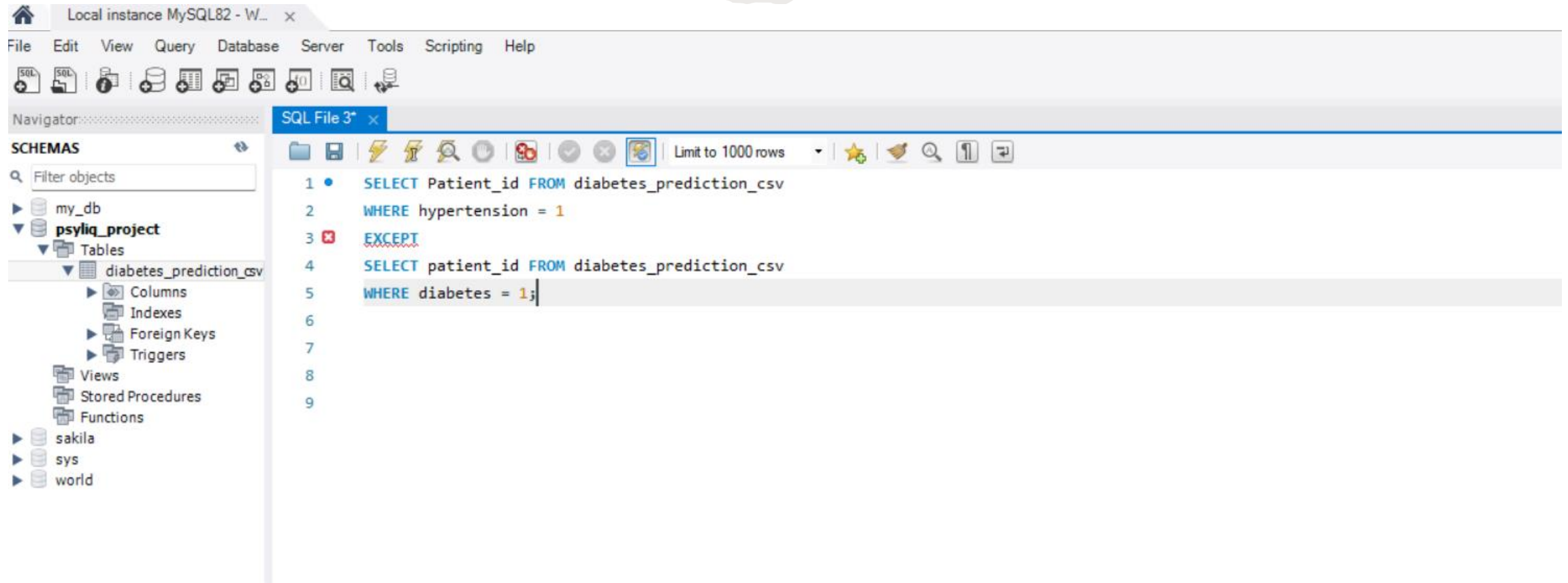
```
1 • INSERT INTO diabetes_prediction_csv
2   VALUES('JOE BIDEN','PT100101','MALE',60,0,1,"never",33.6,7.8,130,1);
3
```

At the bottom, the 'Administration' tab is active, showing the 'Schemas' section. The 'Table:' field displays 'diabetes_prediction_csv'.

Delete all patients with heart disease from the database.



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

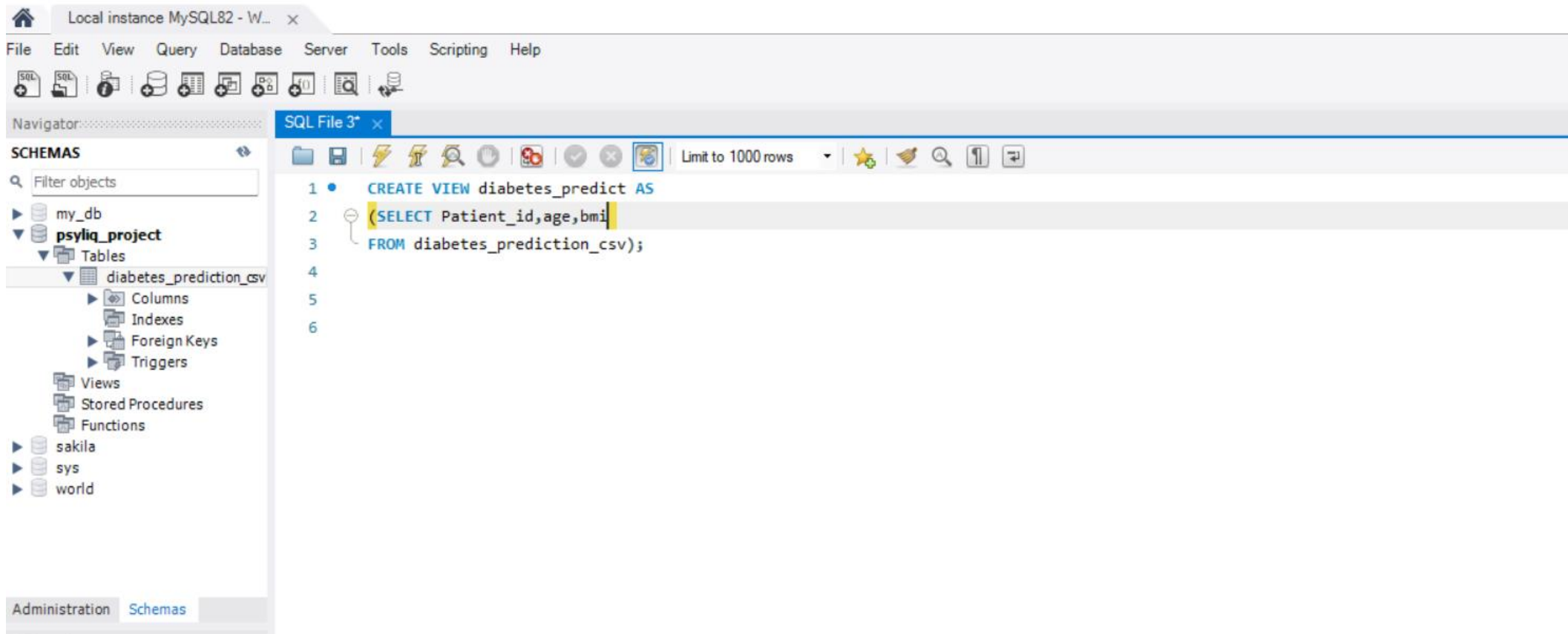


The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of databases. The 'psyliq_project' database is expanded, showing its tables, including 'diabetes_prediction_csv'. The main editor window, titled 'SQL File 3', contains the following SQL query:

```
1 • SELECT Patient_id FROM diabetes_prediction_csv
2 WHERE hypertension = 1
3 EXCEPT
4 SELECT patient_id FROM diabetes_prediction_csv
5 WHERE diabetes = 1;
```

The query is designed to find patients who have hypertension (hypertension = 1) but do not have diabetes (diabetes = 1) by using the EXCEPT operator to subtract the second set of results from the first.

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



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

- Moving employee's name to another table and replacing them by employee id.
- Adding a table for the patient's name, age, gender and other info, hence, removing them from data table.
- Using primary and foreign keys for patients, employees and general data.
- Setting up constraints such as unique, not null and check.

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

- Using correct data types.
- If there are frequently used columns, we can make use of indexing
- Writing simple queries and avoiding sub-queries.
- Avoiding the (SELECT *) queries and instead retrieving only necessary columns.