



DIABETES PREDICTION ANALYSIS

By-BHANUPRIYA



Retrieve the Patient_id and ages of all patients.

```
SELECT Patient_id, age FROM diabetes_predictions;
```

The screenshot displays a SQL query execution environment. The query entered is `select Patient_id, age from diabetes_prediction;`. The results are shown in a table with the following data:

Patient_id	age
PT101	80
PT102	54
PT103	28
PT104	36
PT105	76
PT106	20
PT107	44

Below the table, the output log shows the execution details:

#	Time	Action	Message
1	22:30:37	select Patient_id, age from diabetes_prediction	89887 row(s) returned

Select all female patients who are older than 40.

```
SELECT * FROM diabetics_prediction WHERE gender= "female" AND  
age> 40;
```

```
1  
2 • select * from diabetes_prediction where gender = "female" and age>40;  
3  
4  
5  
6
```

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

diabetes_prediction 8

Output

Action Output

#	Time	Action	Message
7	22:53:10	select * from diabetes_prediction where gender = "female" and age>40	27929 row(s) returned

Calculate the average BMI of patients.

```
SELECT AVG(bmi) FROM diabetes_prediction;
```

```
1 • select avg(bmi) from diabetes_prediction;
```


```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

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

	avg(bmi)
▶	27.322404018376705

Find patients who have hypertension and diabetes.

```
SELECT * FROM diabetes_prediction WHERE hypertension =1 AND diabetes=1;
```

```
1 • select * from diabetes_prediction where hypertension=1 and diabetes=1;
2
```

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

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_g
▶	JONES WONG	PT139	Male	50	1	0	current	27.32	5.7	260
	PATRIC STEELE	PT205	Female	80	1	0	Ex-smoker	27.32	6.8	280
	CHAD LAW	PT355	Male	63	1	0	Ex-smoker	35.06	5.8	200
	CATHERINE JAMES	PT451	Female	52	1	0	Ex-smoker	50.3	6.6	155
	JOHN HART	PT565	Male	48	1	0	current	36.12	6.8	140
	JOHN BARKER	PT567	Female	79	1	0	Ex-smoker	27.32	6.5	159
	ROBERT BONNET	PT632	Female	49	1	0	not current	36.93	8.8	155
	VITANI BENJAMIN	PT727	Male	43	1	0	not current	40.86	6.6	159

diabetes_prediction 6 x

Output

Action Output

#	Time	Action	Message
✓ 45	14:28:33	select * from diabetes_prediction where hypertension=1 and diabetes=1	1557 row(s) returned

List patients in descending order of blood glucose levels.

```
SELECT * FROM diabetes_prediction ORDER BY blood_glucose_level  
DESC;
```

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

2
3
4
5
6

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	
Aida C Henry	PT88238	Female	57	0	0	never	27.32	7	300	1	
Ernesto Fortu	PT88333	Male	53	0	0	never	36.08	6.5	300	1	
Janice Torbet	PT88440	Male	72	0	0	No Info	30.98	7	300	1	
Jacqueline M Phillips	PT89191	Female	51	0	0	ever	27.46	7.5	300	1	
Brenda G Velasquez	PT89459	Female	28	0	0	No Info	37.1	6.1	300	1	
Victoria Gonzalez	PT89505	Male	68	0	0	ever	29.13	7.5	300	1	

diabetes_prediction 10 x

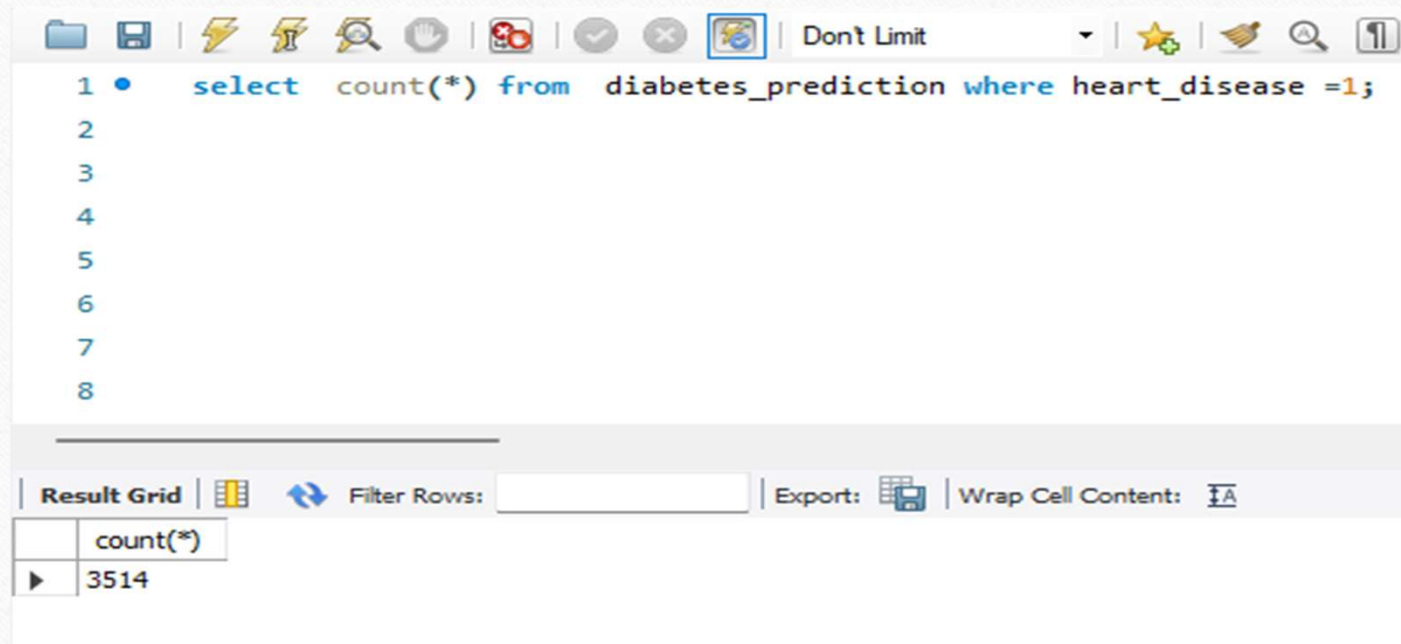
Output

Action Output

#	Time	Action	Message
9	22:53:20	select * from diabetes_prediction where gender = "female" and age>40	27929 row(s) returned

Determine the number of patients with heart disease.

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



The screenshot shows a SQL query editor interface. The query entered is: `select count(*) from diabetes_prediction where heart_disease =1;`. The query is executed, and the results are displayed in a table below the query editor. The table has two columns: `count(*)` and the value `3514`.

count(*)
3514

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

```
SELECT smoking_history, COUNT(*) AS number_of_Pateints FROM diabetes_prediction  
GROUP BY smoking_history;
```

```
1 • SELECT smoking_history, COUNT(*) AS number_of_Pateints FROM diabetes_prediction G  
2
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:		Result Grid
smoking_history	number_of_Pateints					
▶ never	31608					
No Info	32128					
current	8343					
former	8391					
ever	3614					
not current	5803					

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);
```

```
1 • select Patient_id from diabetes_prediction where bmi > (select avg(bmi) from
2 diabetes_prediction);
3
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:			
Wrap Cell Content:			
Fetch rows:			
Patient_id			
PT109			
PT112			
PT113			
PT117			
PT121			
PT124			
PT126			
PT128			
PT131			
PT140			

diabetes_prediction 4 x

Output

Action Output

#	Time	Action	Message
3	12:53:27	select Patient_id from diabetes_prediction where bmi > (select avg(bmi) from diabetes_predict...	30302 row(s) returned

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

```
SELECT MIN(HbA1c_level), MAX(HbA1c_level) FROM diabetes_prediction;
```

```
1 • select min(HbA1c_level), max(HbA1c_level) from diabetes_prediction;  
2
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	min(HbA1c_level)	max(HbA1c_level)			
▶	3.5	9			

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

```
SELECT Patient_id, ABS((age- year(now())) AS year_of_birth from  
diabetes_prediction;
```

```
1 • select patient_id, abs((age- year(now())) as year_of_birth from diabetes_prediction;|  
2
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	patient_id	year_of_birth			
▶	PT101	1943			
	PT102	1969			
	PT103	1995			
	PT104	1987			
	PT105	1947			
	PT106	2003			
	PT107	1979			
	PT108	1944			
	PT109	1981			
	PT110	1991			

Result 7 ×

Output

Action Output

#	Time	Action	Message
6	13:08:47	select patient_id, abs((age- year(now())) as year_of_birth from diabetes_prediction	89887 row(s) returned

Rank patients by blood glucose level within each gender group.

```
SELECT Patient_id, gender, row_number() OVER(ORDER BY blood_glucose_level DESC ,  
gender DESC) AS PT_Rank FROM diabetes_prediction;
```

```
1 • select Patient_id, gender, row_number() over(order by blood_glucose_level desc, gender desc)  
2 from diabetes_prediction;  
3
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
Patient_id	gender	PT_Rank					
PT87825	Male	1					
PT89757	Male	2					
PT89505	Male	3					
PT88440	Male	4					
PT88333	Male	5					
PT75103	Male	6					
PT75767	Male	7					
PT75109	Male	8					
PT78736	Male	9					
PT77283	Male	10					

Result 8 x

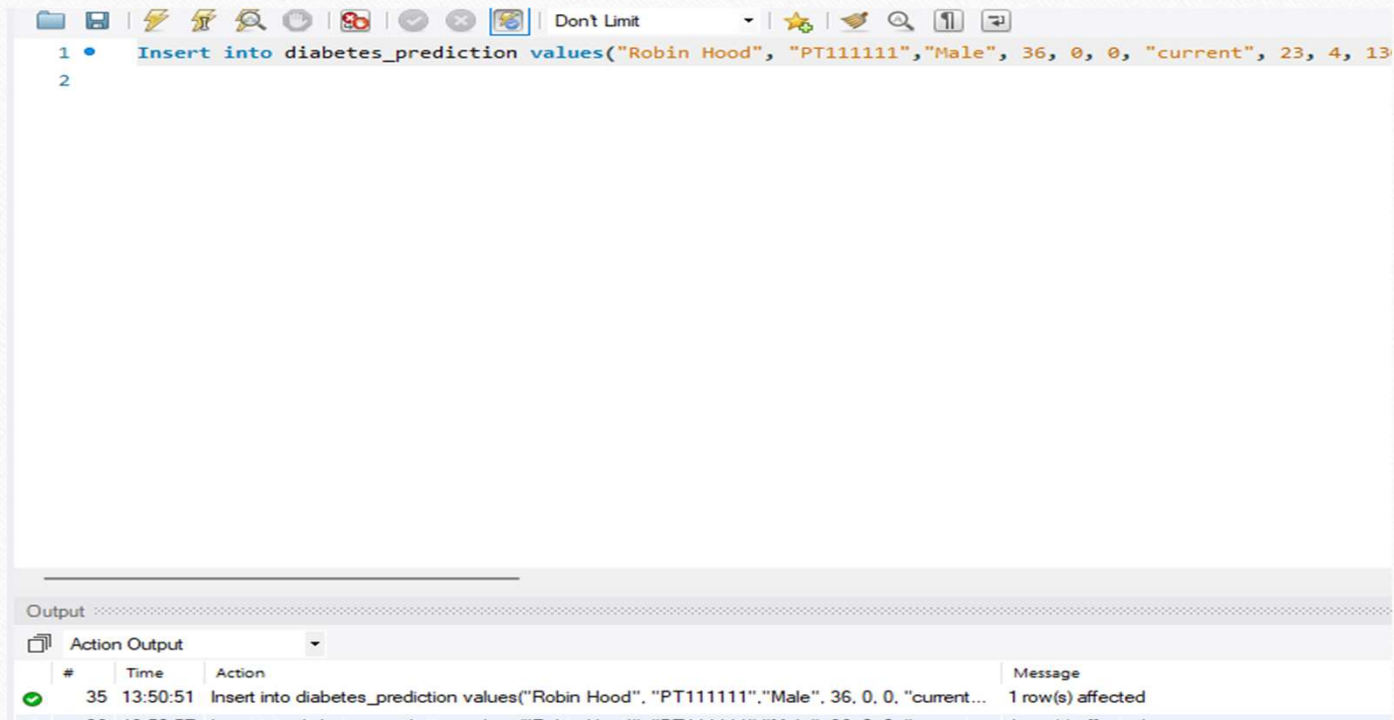
Output

Action Output

#	Time	Action	Message
7	13:08:56	select patient_id, abs((age- year(now())))) as year_of_birth from diabetes_prediction	89887 row(s) returned

. Insert a new patient into the database with sample data

```
INSERT INTO diabetes_prediction VALUES("Robin Hood", PT11111", "Male", 36, 0, 0,  
"current",23, 4, 130 ,0);
```



Delete all patients with heart disease from the database.

DELETE FROM diabetes_prediction WHERE heart_disease;

```
1 • delete from diabetes_prediction where heart_disease=1;  
2
```

Output

Action Output

#	Time	Action	Message
1	13:56:44	delete from diabetes_prediction where heart_disease=1	3514 row(s) affected

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

```
UPDATE diabetes_prediction SET smoking_history = "Ex-Smoker" WHERE age > 50;
```

The screenshot displays a database management tool interface. At the top, a SQL query is entered in a text area: `1 • update diabetes_prediction set smoking_history= "Ex-smoker" where age>50;`. To the right of the query area, there is a vertical sidebar with the text "Automa the tool" and "curre". Below the query area, there is a section labeled "Output" with a dropdown menu set to "Action Output". Below this, a table shows the execution results:

#	Time	Action	Message
3	14:02:08	update diabetes_prediction set smoking_history= "Ex-smoker" where age>50	31239 row(s) affected Rows matched: 31239 Changed: 31239

At the bottom right of the interface, there is a link labeled "Context Help".

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

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

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

Result Grid										
Filter Rows:										
Export: Wrap Cell Content: Fetch rows:										
	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_sugar
▶	DENISE SCHMITT	PT129	Male	45	1	0	never	26.47	4	158
	RAY CRAWFORD	PT155	Female	45	1	0	never	23.05	4.8	130
	KENNETH SMITH	PT161	Male	44	1	0	current	27.86	6.6	145
	CHARLES SCOTT	PT215	Female	55	1	0	Ex-smoker	34.2	5.7	140
	SHANNON SAKOWSKI	PT227	Male	79	1	0	Ex-smoker	28.73	6.6	160
	MARISA MORET	PT241	Female	80	1	0	Ex-smoker	44.06	6.5	160
	STEPHEN TACCHINI	PT326	Female	48	1	0	never	36.73	6.6	126
	ANDREW LOGAN	PT339	Male	59	1	0	Ex-smoker	25.31	6	130

Result 3

Output

Action Output

#	Time	Action	Message
8	14:09:48	select * from diabetes_prediction where hypertension=1 except select * from diabetes_predic...	4340 row(s) returned

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

- ❖ Adding a table for the Patient's name, age, gender and other information , hence removing them from data table.
 - ❖ Using primary keys and foreign keys for Patients, general data.
 - ❖ Using views for simplifying queries.
- ❖ Setting up constraints such as unique , not null and check constraints.

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

- Ensuring the relevant columns used in WHERE clauses are indexed.
- Limit SELECT columns i.e. , retrieve only the needed columns rather than selecting all columns.
- Applying filter conditions as early as possible in the query to reduce the dataset size before performing complex operations.
 - Writing simple queries and avoiding sub-queries.



THANKYOU