Diabetes Prediction

Creating table after connecting to database

Query | Query History

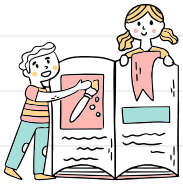
```
1  CREATE TABLE Diabetes (
2      EmployeeName varchar(255),
3      Patient_id varchar(255),
4      gender varchar(255),
5      age int,
6      hypertension int,
7      heart_disease int,
8      smoking_history varchar(20),
9      bmi double precision,
10     HbA1c_level double precision,
11     blood_glucose_level int,
12     diabetes int
13
14 );
15
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 482 msec.
```

FIRST VIEW OF DATASET



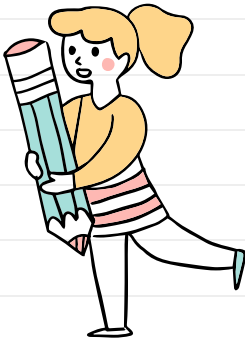
```
SELECT * FROM diabetes;
```

	employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying
1	NATHANIEL FORD	PT101	Female	80	0	1	never
2	GARY JIMENEZ	PT102	Female	54	0	0	No Info
3	ALBERT PARDINI	PT103	Male	28	0	0	never
4	CHRISTOPHER CHONG	PT104	Female	36	0	0	current
5	PATRICK GARDNER	PT105	Male	76	1	1	current
6	DAVID SULLIVAN	PT106	Female	20	0	0	never
7	ALSON LEE	PT107	Female	44	0	0	never
8	DAVID KUSHNER	PT108	Female	79	0	0	No Info
9	MICHAEL MORRIS	PT109	Male	42	0	0	never
10	JOANNE HAYES-WHITE	PT110	Female	32	0	0	never
11	ARTHUR KENNEY	PT111	Female	53	0	0	never
12	PATRICIA JACKSON	PT112	Female	54	0	0	former
13	EDWARD HARRINGTON	PT113	Female	78	0	0	former

1. Retrieve the Patient_id and ages of all patients.

```
SELECT Patient_id , age FROM Diabetes;
```

	patient_id character varying (255)	age integer
1	PT101	80
2	PT102	54
3	PT103	28
4	PT104	36
5	PT105	76
6	PT106	20
7	PT107	44
8	PT108	79
9	PT109	42
10	PT110	32
11	PT111	53
12	PT112	54
13	PT113	78
14	PT114	67
15	PT115	76



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

```
SELECT * FROM Diabetes
WHERE gender = 'Female' AND age > 40;
```

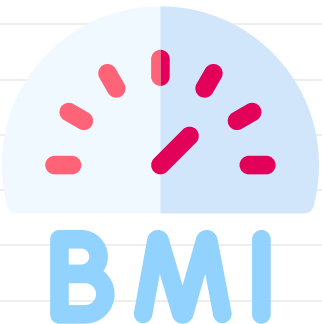
	employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba
1	NATHANIEL FORD	PT101	Female	80	0	1	never	25.19	
2	GARY JIMENEZ	PT102	Female	54	0	0	No Info	27.32	
3	ALSON LEE	PT107	Female	44	0	0	never	19.31	
4	DAVID KUSHNER	PT108	Female	79	0	0	No Info	23.86	
5	ARTHUR KENNEY	PT111	Female	53	0	0	never	27.32	
6	PATRICIA JACKSON	PT112	Female	54	0	0	former	54.7	
7	EDWARD HARRINGTON	PT113	Female	78	0	0	former	36.05	
8	JOHN MARTIN	PT114	Female	67	0	0	never	25.69	
9	DAVID FRANKLIN	PT115	Female	76	0	0	No Info	27.32	
10	SEBASTIAN WONG	PT118	Female	42	0	0	never	24.48	
11	MARTY ROSS	PT119	Female	42	0	0	No Info	27.32	
12	GEORGE GARCIA	PT123	Female	69	0	0	never	21.24	
13	VICTOR WYRSCH	PT124	Female	72	0	1	former	27.94	
14	HARLAN KELLY-JR	PT131	Female	53	0	0	No Info	31.75	
15	GARY AMELIO	PT133	Female	41	0	0	current	22.01	

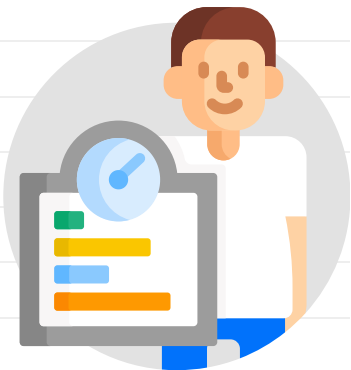
Total rows: 1000 of 31155 Query complete 00:00:00.641 Ln 21, Col 38

3. Calculate the average BMI of patients.

```
SELECT AVG(bmi) AS Average_BMI FROM Diabetes;
```

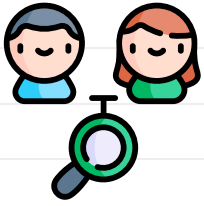
	average_bmi double precision
1	27.32076709999422



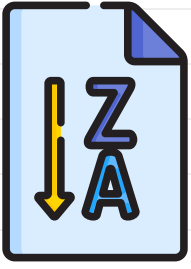


4. List patients in descending order of blood glucose levels.

```
SELECT employeename , patient_id , age , blood_glucose_level
FROM Diabetes
order by blood_glucose_level DESC;
```



	employeename character varying (255)	patient_id character varying (255)	age integer	blood_glucose_level integer
1	LAUREN GREEN	PT6046	61	300
2	KIRK EDISON JR	PT1461	66	300
3	THOMAS HULL	PT3865	39	300
4	RODERICK SHEHEE	PT19528	80	300
5	GREGORY KNIGHT	PT26430	65	300
6	ANDREW MOLINA	PT16657	58	300
7	BRENT BARNES	PT12682	72	300
8	RICHARD JONES	PT1466	77	300
9	ELLEN CHEN	PT25349	33	300
10	SAMANTHA OBRIEN	PT34314	54	300
11	Denise Martinez	PT37183	54	300
12	DANIEL DECOSSIO	PT1319	65	300
13	JOHN GUO	PT31310	71	300
14	WILLIAM GARCIA	PT1321	30	300
15	CECILIA RIOS	PT17379	80	300
16	KHANH CHAU	PT22898	66	300



5. Find patients who have hypertension and diabetes.

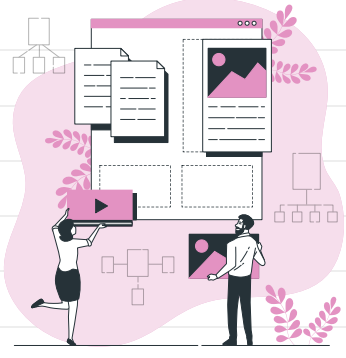
```
SELECT * FROM diabetes
WHERE heart_disease = 1 AND diabetes = 1;
```

	employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba1c_l double p
1	JOHN HANLEY	PT127	Male	67	0	1	not current	27.32	
2	ARTHUR STELLINI	PT343	Male	57	1	1	not current	27.77	
3	GHODSI DAVARY	PT462	Male	80	0	1	former	24.36	
4	JACK CHOW	PT667	Male	75	0	1	not current	28.12	
5	BRIAN PHILPOTT	PT691	Male	69	0	1	former	24.1	
6	ROBERT PORTER	PT719	Female	59	0	1	never	60.26	
7	TERRENCE YUEN	PT904	Male	80	0	1	former	32.95	
8	EDWARD LEE	PT1123	Female	62	1	1	former	44.23	
9	ELLEN BRIN	PT1236	Female	62	1	1	never	43.16	
10	KATHLEEN MURPHY	PT1375	Female	76	0	1	former	25.68	
11	KENNETH ROUX	PT1378	Female	67	1	1	current	28.52	
12	NIKKI GRIFFEY	PT1511	Male	80	0	1	never	28.66	
13	PATRICIA GREEN	PT1611	Male	76	0	1	never	20.96	
14	CHRISTINE MARTIN	PT1626	Male	67	0	1	No Info	35.48	
15	NARDA GILLESPIE	PT1699	Female	80	0	1	never	26	

Total rows: 1000 of 1267 | Query complete 00:00:00.518 | Ln 31, Col 2

6. Determine the number of patients with heart disease.

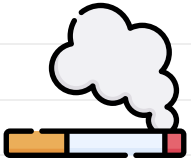
```
select count(patient_id) as Heart_Patients
from diabetes
where heart_disease = 1;
```



	heart_patients bigint
1	3942

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

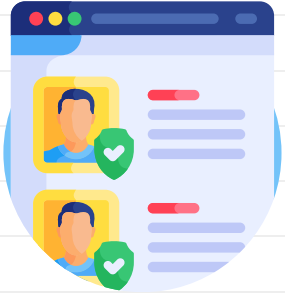
```
SELECT smoking_history , COUNT(*) AS Patient_count
FROM diabetes
WHERE smoking_history IN ('current','never')
GROUP BY 1;
```



	smoking_history character varying (20)	patient_count bigint
1	never	35095
2	current	9286

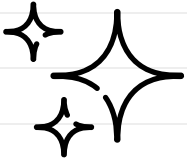
8. Retrieve the Patient_IDs of patients who have a BMI greater than the average BMI.

```
SELECT patient_id FROM diabetes
WHERE bmi > (SELECT AVG(bmi) FROM diabetes);
```



	patient_id character varying (255)
1	PT109
2	PT112
3	PT113
4	PT117
5	PT121
6	PT124
7	PT126
8	PT128
9	PT131
10	PT140
11	PT143
12	PT144
13	PT149
14	PT153
15	PT156

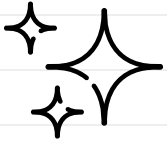
9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1c level.



```sql
--patient with the highest HbA1c level
SELECT *
FROM diabetes
ORDER BY hba1c_level DESC
LIMIT 1;
```

employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba1c_le double p
MICHAEL THOMPSON	PT141	Male	73	0	0	former	25.91	

```sql
--patient with the lowest HbA1c level
SELECT * FROM diabetes
ORDER BY hba1c_level ASC
LIMIT 1;
```

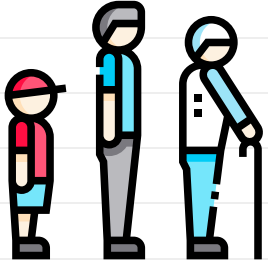


	employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba1c_level double precisio
1	ELLEN MOFFATT	PT120	Male	37	0	0	ever	25.72	

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

```
SELECT patient_id, age, EXTRACT(YEAR FROM CURRENT_DATE) - age AS Birth_Year
FROM diabetes;
```

	patient_id character varying (255)	age integer	birth_year numeric
1	PT101	80	1943
2	PT102	54	1969
3	PT103	28	1995
4	PT104	36	1987
5	PT105	76	1947
6	PT106	20	2003
7	PT107	44	1979
8	PT108	79	1944
9	PT109	42	1981
10	PT110	32	1991
11	PT111	53	1970
12	PT112	54	1969
13	PT113	78	1945
14	PT114	67	1956
15	PT115	76	1947
16	PT116	78	1945

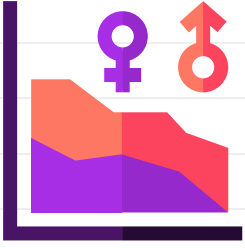


11. Rank patients by blood glucose level within each gender group.

```
SELECT employeename as Patient_name, patient_id , gender , blood_glucose_level ,
dense_rank () over (partition by gender order by blood_glucose_level desc) as Glucose_rank
FROM Diabetes;
```

	patient_name character varying (255)	patient_id character varying (255)	gender character varying (255)	blood_glucose_level integer	glucose_rank bigint
1	ADELIA CARANDANG	PT21073	Female	300	1
2	SACHI MANALISAY	PT27338	Female	300	1
3	WINSON SETO	PT6227	Female	300	1
4	Marcus Dobrowolski	PT41248	Female	300	1
5	JOSEFINA JULATON	PT18493	Female	300	1
6	Marilyn Melgarejo	PT40222	Female	300	1
7	LILIAN REYNA	PT23667	Female	300	1
8	JOANNA CHAN	PT24450	Female	300	1
9	ARSENIA DAY	PT21339	Female	300	1
10	SHUK HA SIU	PT34301	Female	300	1
11	LAURA GRGICH	PT23312	Female	300	1
12	BETH KUHNS	PT6500	Female	300	1
13	BROCK WELLS	PT2635	Female	300	1
14	ALI MISAGHI	PT3341	Female	300	1
15	DEE DEE TYSON	PT21704	Female	300	1
16	BILAL LOUBIE	PT26831	Female	300	1

Total rows: 1000 of 100000 Query complete 00:00:01.336



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

```sql
-- Start a transaction
BEGIN;

-- Update smoking history for patients older than 50 to "Ex Smoker"
UPDATE diabetes
SET smoking_history = 'Ex Smoker'
WHERE age > 50;

-- Commit the transaction
COMMIT;

select Patient_id , smoking_history , age from diabetes where age > 50;
```

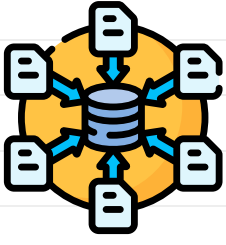
	patient_id character varying (255)	smoking_history character varying (20)	age integer
1	PT101	Ex Smoker	80
2	PT102	Ex Smoker	54
3	PT105	Ex Smoker	76
4	PT108	Ex Smoker	79
5	PT111	Ex Smoker	53
6	PT112	Ex Smoker	54
7	PT113	Ex Smoker	78
8	PT114	Ex Smoker	67
9	PT115	Ex Smoker	76
10	PT116	Ex Smoker	78
11	PT123	Ex Smoker	69
12	PT124	Ex Smoker	72
13	PT127	Ex Smoker	67
14	PT131	Ex Smoker	53
15	PT135	Ex Smoker	76
16	PT141	Ex Smoker	73

Total rows: 1000 of 38463 Query complete 00:00:00.360

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

```
INSERT INTO Diabetes
(Employeename,Patient_id,gender,age,hypertension,heart_disease,smoking_history,bmi,HbA1c_level,blood_glucose_level,diabetes)
VALUES ('Andrew Simon','PT0154235','Male',54,1,1,'Ex Smoker',26.12,6.2,187,1);
SELECT * FROM diabetes
WHERE Employeename = 'Andrew Simon';
```

	employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba1c_level double precisio
1	Andrew Simon	PT0154235	Male	54	1	1	Ex Smoker	26.12	



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

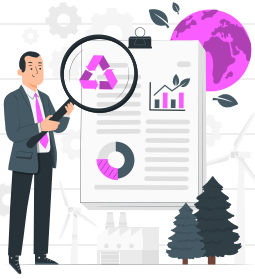
```
DELETE FROM Diabetes
WHERE heart_disease = 1;
SELECT * FROM Diabetes WHERE heart_disease = 1;
```

employeename character varying (255)	patient_id character varying (255)	gender character varying (255)	age integer	hypertension integer	heart_disease integer	smoking_history character varying (20)	bmi double precision	hba1c_level double precisio

EMPTY RESULT SHOWS SUCCESSFULL QUERY

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

```
SELECT Employeename, Patient_id , hypertension , diabetes FROM diabetes WHERE hypertension = 1
EXCEPT
SELECT Employeename, Patient_id , hypertension , diabetes FROM diabetes WHERE diabetes = 1;
```



	employeename character varying (255)	patient_id character varying (255)	hypertension integer	diabetes integer
1	Aaron Fischer	PT78453	1	0
2	AARON DEL TREDICI	PT4079	1	0
3	AARON HOLLISTER	PT18270	1	0
4	Aaron I Maxwell	PT99335	1	0
5	Aaron W Wu	PT91573	1	0
6	ABDIWAHAB HASHI	PT16085	1	0
7	Abdul Lateef	PT92308	1	0
8	ABELARDO GOMEZ	PT22079	1	0
9	Abraham Hagos	PT53834	1	0
10	ADA ARANDA	PT13683	1	0
11	Ada C Aranda	PT84656	1	0
12	Ada Lien	PT70785	1	0
13	Ada Wong	PT55270	1	0
14	Adam B Lobsinger	PT89749	1	0
15	ADAM EATIA	PT4526	1	0
16	Adam Kujath	PT42292	1	0

Total rows: 1000 of 4839 Query complete 00:00:00.669

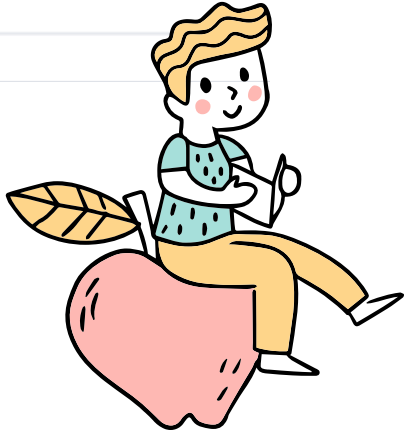
16. Define a unique constraint on the "patient_id" column to ensure its values are unique.

```
88  ALTER TABLE diabetes ADD CONSTRAINT Unique_Patient_id UNIQUE(Patient_id);
89
90
```

Data Output | Messages | Notifications

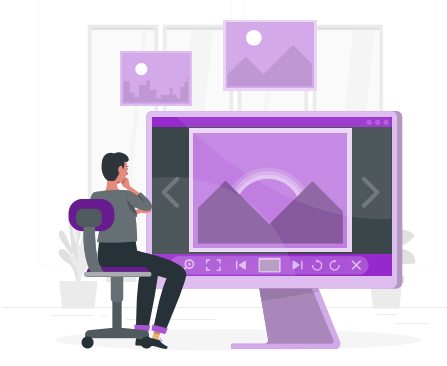
```
ALTER TABLE

Query returned successfully in 2 secs 626 msec.
```



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

```
CREATE VIEW Patient_details AS
SELECT Patient_id, age, bmi FROM diabetes;
SELECT * FROM Patient_details;
```



	patient_id character varying (255)	age integer	bmi double precision
1	PT103	28	27.32
2	PT104	36	23.45
3	PT106	20	27.32
4	PT107	44	19.31
5	PT109	42	33.64
6	PT110	32	27.32
7	PT117	15	30.36
8	PT118	42	24.48
9	PT119	42	27.32
10	PT120	37	25.72
11	PT121	40	36.38
12	PT122	5	18.8
13	PT125	4	13.99
14	PT126	30	33.76
15	PT128	40	27.85
16	PT129	45	26.47

Total rows: 1000 of 96058 Query complete 00:00:00.739

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

Here are suggestions for improving the database schema:

- **Normalization**: Ensure the database follows normalization principles to minimize data redundancy and dependencies.
- **Foreign Keys**: Use foreign keys to establish relationships, ensuring referential integrity and preventing orphaned records.
- **Indexes**: Create indexes on frequently used columns to improve query performance, but avoid excessive indexing.
- **Default Values and Constraints**: Employ default values and constraints to enforce data integrity rules, reducing the risk of invalid data.
- **Audit Trails**: Implement audit trails to track changes, providing a historical record and enhancing accountability.






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

Here are few points for optimizing SQL queries on this dataset:

- **Indexing:** Create indexes on columns frequently used in WHERE clauses or JOIN conditions to enhance query performance.
- **Limit SELECT Columns:** Select only the necessary columns in your queries to reduce data transfer and improve efficiency.
- **Optimize WHERE Clauses:** Ensure efficient WHERE clauses by avoiding functions on indexed columns and optimizing conditions.
- **Use JOINs Efficiently:** Optimize JOIN operations by selecting the appropriate type and ensuring efficient join conditions.
- **Update Statistics Regularly:** Keep table statistics up-to-date to assist the query planner in making informed execution plans.