TASK 3 :- PROVIDING INSIGHTS ON DIABETES PREDICTION DATA USING SQL

NAME : SUPRIYA BHAT V
Data Analysis Project using SQL
Email ID: supriyabhatv@gmail.com

Dataset Overview: Diabetes Records for 100,000 Patients

1. **Employee Name :** The name of the employee or identifier associated with the patient record.
2. **Patient ID :** A unique identifier assigned to each patient.
3. **Smoking History :** Indicates the patient's smoking history, often as a binary variable
4. **Age :** Represents the age of the patient.
5. **Hypertension :** Binary variable indicating the presence or absence of hypertension.
6. **Heart Disease :** Binary variable indicating the presence or absence of heart disease.
7. **BMI :** Quantitative measure representing the body mass index of the patient.
8. **HbA1c Level :** Quantitative measure of the patient's HbA1c (glycated hemoglobin) levels.
9. **Blood Glucose Level :** Quantitative measure of the patient's blood glucose levels.
10. **Diabetes :** Binary variable indicating the presence or absence of diabetes.

1. Retrieve the Patient_id and ages of all patients.

MYSQL CODE

```
select Patient_id,age from Diabetics_Predictions;
```

OUTPUT

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

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

MYSQL CODE

```
select * from Diabetics_Predictions  where gender='Female' and age>40;
```

OUTPUT

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

3. Calculate the average BMI of patients.

MYSQL CODE

```
select round(avg(bmi),2) as average_bmi  from Diabetics_Predictions;
```

OUTPUT

average_bmi
27.32

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

MYSQL CODE

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

OUTPUT

Employee_Name	Patient_Ic	gender	age	hypertensior	heart_disease	smoking_history	bmi	HbA1c_le	blood_glucose_level	diabetes
Adrian G Mendez	PT98419	Male	57	0	0	Ex-smoker	27.32	6.5	300	1
Lenora G Banks	PT98454	Female	76	1	0	Ex-smoker	38.59	6.6	300	1
Dante Rogayan	PT98461	Male	72	0	0	Ex-smoker	27.72	6.6	300	1
Tinisha C Bishop	PT98500	Male	73	0	0	Ex-smoker	27.32	8.8	300	1
Tualatai Auimat...	PT98538	Female	17	0	0	never	26.52	8.2	300	1
Michelle D McGee	PT98852	Male	79	0	0	Ex-smoker	27.32	7.5	300	1
Lawrence Shum	PT98855	Male	43	0	0	former	48.56	6.8	300	1
Seth I Rubenstein	PT98911	Female	60	0	0	Ex-smoker	40.18	9	300	1
Philip Tran	PT99008	Male	69	0	0	Ex-smoker	31.56	7	300	1
Gilbert J Fragoso	PT99638	Female	67	1	0	Ex-smoker	34.3	5.7	300	1
Amado A Lumas Jr	PT99663	Male	56	1	0	Ex-smoker	28.47	6.1	300	1

5. Find patients who have hypertension and diabetes.

MYSQL CODE

```
select * from Diabetics_Predictions where hypertension =1 and diabetes=1;
```

OUTPUT

Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
PT139	Male	50	1	0	current	27.32	5.7	260	1
PT205	Female	80	1	0	never	27.32	6.8	280	1
PT343	Male	57	1	1	not current	27.77	6.6	160	1
PT355	Male	63	1	0	ever	35.06	5.8	200	1
PT451	Female	52	1	0	never	50.3	6.6	155	1
PT565	Male	48	1	0	current	36.12	6.8	140	1
PT567	Female	79	1	0	former	27.32	6.5	159	1
PT632	Female	49	1	0	not current	36.93	8.8	155	1
PT727	Male	43	1	0	not current	40.86	6.6	159	1
PT828	Female	38	1	0	not current	27.32	6.1	160	1
PT852	Female	28	1	0	never	20.09	6.6	200	1
PT861	Male	59	1	0	ever	25.94	9	140	1

6. Determine the number of patients with heart disease.

MYSQL CODE

```
select count(*) as num_patients_with_heart_disease from Diabetics_Predictions
where heart_disease=1;
```

OUTPUT

num_patients_with_heart_disease
7884

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

MYSQL CODE 1

```
select smoking_history, count(*) as num_patients from  Diabetics_Predictions group
by smoking_history;
```

OUTPUT

smoking_history	num_patients
never	70190
No Info	71632
current	18572
former	18704
ever	8008
not current	12894

MYSQL CODE 2

```
SELECT
SUM(CASE WHEN smoking_history IN ('Former', 'Current', 'Ever', 'Not Current')
THEN 1 ELSE 0 END) AS
smokers_count
SUM(CASE WHEN smoking_history = 'Never' THEN 1 ELSE 0 END) AS
non_smokers_count
FROM Diabetics_Predictions;
```

OUTPUT

smokers_count	non_smokers_count
58178	70190

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

MYSQL CODE

```
select Patient_id, EmployeeName, bmi  from Diabetics_Predictions
where bmi>(select avg(bmi) from Diabetics_Predictions);
```

OUTPUT

Patient_id	EmployeeName	bmi
PT109	MICHAEL MORRIS	33.64
PT112	PATRICIA JACKSON	54.7
PT113	EDWARD HARRINGTON	36.05
PT117	AMY HART	30.36
PT121	VENUS AZAR	36.38
PT124	VICTOR WYRSCH	27.94
PT126	GREGORY SUHR	33.76
PT128	RAYMOND GUZMAN	27.85
PT131	HARLAN KELLY-JR	31.75
PT140	BRENDAN WARD	56.43
PT143	THOMAS SIRAGUSA	32.02
PT144	MICHAEL THOMPSON	29.3
PT149	JAMES DUDLEY	28.27

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

MYSQL CODE 1

```
-- Patient with the highest HbA1c level
SELECT * FROM Diabetics_Predictions
WHERE HbA1c_level = (SELECT MAX(HbA1c_level  FROM  Diabetics_Predictions);
```

OUTPUT

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level
MICHAEL THOMPSON	PT141	Male	73	0	0	former	25.91	9
KEVIN CASHMAN	PT156	Male	50	0	0	former	37.16	9
MARK CASTAGNOLA	PT236	Male	80	0	0	never	22.06	9
WILLIAM SCOTT	PT270	Female	61	0	0	not current	39.36	9
JOANNE HOEPER	PT400	Female	42	0	0	never	24.81	9
VINCENT PAMPANIN	PT519	Female	52	0	0	No Info	27.32	9
FRANK KOSTA	PT673	Female	80	0	0	never	36.74	9
VINCENT NOLAN	PT710	Female	69	0	0	former	31.17	9
KAREN KUBICK	PT861	Male	59	1	0	ever	25.94	9
MANOUCHEHR BOOZARPOUR	PT907	Male	58	0	0	ever	19.46	9
VICTOR WONG	PT1242	Female	54	1	0	never	22.48	9
DANIEL DECOSSIO	PT1319	Male	65	1	0	former	22.06	9

MYSQL CODE 2

```
-- Patient with the lowest HbA1c level
SELECT * FROM Diabetics_Predictions
WHERE HbA1c_level = (SELECT MIN(HbA1c_level) FROM Diabetics_Predictions);
```

OUTPUT

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level
ELLEN MOFFATT	PT120	Male	37	0	0	ever	25.72	3.5
JOHN TURSI	PT134	Female	20	0	0	never	22.19	3.5
SHARON MCCOLE WICHER	PT145	Female	67	0	0	No Info	27.32	3.5
MARK KEARNEY	PT158	Female	19	0	0	never	23.35	3.5
MONIQUE MOYER	PT174	Male	43	0	0	not current	27.32	3.5
JOHN HALEY JR	PT213	Male	37	0	0	No Info	27.14	3.5
KHAIRUL ALI	PT219	Female	12	0	0	No Info	20.9	3.5
MICHAEL CASTAGNOLA	PT221	Female	36	0	0	No Info	27.32	3.5
JOHN RAHAIM	PT233	Female	79	0	0	No Info	27.32	3.5
PATRICIA CARR	PT250	Female	29	0	0	No Info	33.22	3.5
OSCAR CABRERA	PT265	Female	21	0	0	No Info	27.7	3.5
AMPARO RODRIGUEZ	PT269	Female	46	0	0	former	35.44	3.5

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

MYSQL CODE

```
SELECT EmployeeName,Patient_id,age, (YEAR(CURDATE()) - age) AS birth_year
FROM Diabetics_Predictions;
```

OUTPUT

EmployeeName	Patient_id	age	birth_year
NATHANIEL FORD	PT101	80	1944
GARY JIMENEZ	PT102	54	1970
ALBERT PARDINI	PT103	28	1996
CHRISTOPHER CHONG	PT104	36	1988
PATRICK GARDNER	PT105	76	1948
DAVID SULLIVAN	PT106	20	2004
ALSON LEE	PT107	44	1980
DAVID KUSHNER	PT108	79	1945
MICHAEL MORRIS	PT109	42	1982
JOANNE HAYES-WHITE	PT110	32	1992
ARTHUR KENNEY	PT111	53	1971
PATRICIA JACKSON	PT112	54	1970
EDWARD HARRINGTON	PT113	78	1946

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

MYSQL CODE

```
SELECT
*,
ROW_NUMBER() OVER (PARTITION BY gender ORDER BY
Blood_Glucose_Level) AS Row_Number_Ranking,
DENSE_RANK() OVER (PARTITION BY gender ORDER BY
Blood_Glucose_Level) AS Dense_Ranking
FROM Diabetics_Predictions;
```

OUTPUT

Ranking order of Female

EmployeeName	Patient_id	gender	age	hype	heart_dis	smoking	bmi	HbA1c_level	blood_gl	diabetes	Row_Number_Ranking
Laverne Evans	PT99154	Female	45	0	0	never	49.17	4.5	80	0	1
Simon K Ng	PT99450	Female	29	0	0	never	27.32	6.5	80	0	2
Erik E Jaszewski	PT99449	Female	80	0	0	No Info	27.32	5.8	80	0	3
Francis C Cheung	PT99751	Female	55	0	0	never	52.97	5.7	80	0	4
David Gogna	PT99670	Female	49	0	0	current	30.05	6.5	80	0	5
Marcia Ortiz	PT99576	Female	27	0	0	current	37.72	5.7	80	0	6
Kira L Barrera	PT99243	Female	59	0	0	never	23.9	6.1	80	0	7
Margaret M Sie...	PT99579	Female	18	0	0	never	21.97	6.2	80	0	8
Jacquelyn J Va...	PT99650	Female	51	0	0	No Info	26.44	4.8	80	0	9
Jacqueline C R...	PT99894	Female	72	1	0	never	50.85	3.5	80	0	10
Judi Soto	PT99994	Female	20	0	0	current	28.06	6	80	0	11
Matthew R Davis	PT99038	Female	62	0	0	never	21.88	6.5	80	0	12

Ranking order of Male

EmployeeName	Patient_id	gender	age	hype	heart_dis	smoking	bmi	HbA1c_level	blood_gl	diabetes	Row_Number_Ranking
Loretta Meng	PT95800	Male	21	0	0	No Info	27.32	5.8	80	0	1
Carlos Mims	PT95062	Male	23	0	0	never	21.7	5.8	80	0	2
Selena Y Lee	PT95540	Male	68	0	0	No Info	27.32	6.5	80	0	3
Jon E Reiter	PT93813	Male	0.56	0	0	No Info	17.58	6.2	80	0	4
Isaiah J Hurtado	PT95063	Male	22	0	0	never	20.8	5.7	80	0	5
Cheryl Y Jones	PT96149	Male	16	0	0	never	17.85	6.5	80	0	6
Erika H Kiefer	PT94835	Male	47	1	0	never	31.11	3.5	80	0	7
Julio C Guillen	PT93731	Male	37	0	0	never	36.97	6.1	80	0	8
Clarisa C Zamo...	PT94278	Male	73	0	0	No Info	25.06	5	80	0	9
Juan Ortiz	PT96160	Male	80	0	0	No Info	27.32	5	80	0	10
Gladys M Carey	PT93761	Male	15	0	0	former	19.05	6.1	80	0	11
Ronald Turner	PT93858	Male	16	0	0	No Info	25.15	6	80	0	13

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

MYSQL CODE

```
UPDATE Diabetics_Predictions
SET smoking_history = 'Ex-smoker WHERE age > 50;

select * from Diabetics_Predictions;
```

OUTPUT

EmployeeName	Patient_id	gender	age	hype	heart_dis	smoking_history	bmi	HbA1c_level	blood_gl	diabetes
NATHANIEL FORD	PT101	Female	80	0	1	Ex-smoker	25.19	6.6	140	0
GARY JIMENEZ	PT102	Female	54	0	0	Ex-smoker	27.32	6.6	80	0
ALBERT PARDINI	PT103	Male	28	0	0	never	27.32	5.7	158	0
CHRISTOPHER CHONG	PT104	Female	36	0	0	current	23.45	5	155	0
PATRICK GARDNER	PT105	Male	76	1	1	Ex-smoker	20.14	4.8	155	0
DAVID SULLIVAN	PT106	Female	20	0	0	never	27.32	6.6	85	0
ALSON LEE	PT107	Female	44	0	0	never	19.31	6.5	200	1

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

```
INSERT INTO Diabetics_Predictions (EmployeeName, Patient_id, gender,
age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level,
blood_glucose_level, diabetes)
VALUES ('Sample Employee', 'PT001', 'Male', 35, 0, 0, 'current', 25.5, 5.6,
100, 0);

SELECT * FROM Diabetics_Predictions WHERE Patient_id = 'PT001'
```

EmployeeName	Patient_id	gender	age	hype	heart_dis	smoking_history	bmi	HbA1c_level	blood_gl	diabetes
Sample Employee	PT001	Male	35	0	0	current	25.5	5.6	100	0

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

MYSQL CODE

```
DELETE FROM Diabetics_Predictions WHERE heart_disease = 1;

# After deleting

select * from Diabetics_Predictions where heart_disease=1;
```

OUTPUT

EmployeeName	Patient_id	gender	age	hype	heart_dis	smoking_history	bmi	HbA1c_level	blood_gl	diabetes

15. Find patients who have hypertension but not diabetes.

MYSQL CODE

```
SELECT * FROM Diabetics_Predictions WHERE  hypertension = 1 AND
diabetes = 'No';
```

OUTPUT

EmployeeName	Patient_id	gender	age	hyperten	heart_dis	smoking_history	bmi	HbA1c_level	blood_gl	diabetes
DENISE SCHMITT	PT129	Male	45	1	0	never	26.47	4	158	0
RAY CRAWFORD	PT155	Female	45	1	0	never	23.05	4.8	130	0
KENNETH SMITH	PT161	Male	44	1	0	current	27.86	6.6	145	0
CHARLES SCOTT	PT215	Female	55	1	0	Ex-smoker	34.2	5.7	140	0
SHANNON SAKOWSKI	PT227	Male	79	1	0	Ex-smoker	28.73	6.6	160	0
MARISA MORET	PT241	Female	80	1	0	Ex-smoker	44.06	6.5	160	0
STEPHEN TACCHINI	PT326	Female	48	1	0	never	36.73	6.6	126	0

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

MYSQL CODE

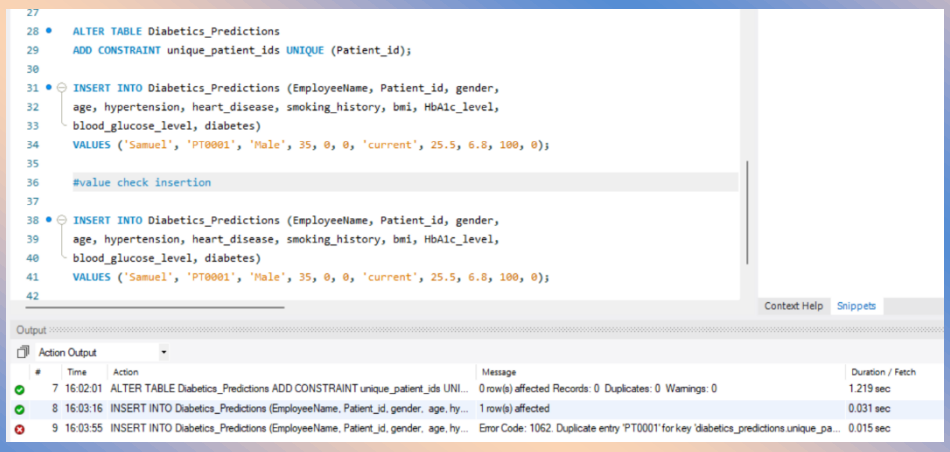
```
ALTER TABLE Diabetics_Predictions
ADD CONSTRAINT unique_patient_ids UNIQUE (Patient_id);

INSERT INTO Diabetics_Predictions (EmployeeName, Patient_id, gender,
age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level,
blood_glucose_level, diabetes)
VALUES ('Samuel', 'PT0001', 'Male', 35, 0, 0, 'current', 25.5, 6.8, 100, 0);

#value check insertion

INSERT INTO Diabetics_Predictions (EmployeeName, Patient_id, gender,
age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level,
blood_glucose_level, diabetes)
VALUES ('Samuel', 'PT0001', 'Male', 35, 0, 0, 'current', 25.5, 6.8, 100, 0);
```

OUTPUT



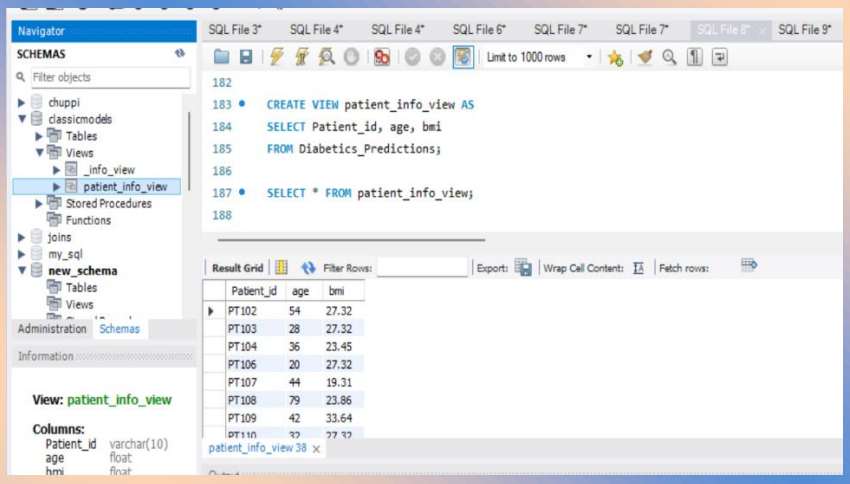
17. Create a view that displays the Patient_ids, ages, and BMI of patients.

MYSQL CODE

```
CREATE VIEW patient_info_view AS
SELECT Patient_id, age, bmi
FROM Diabetics_Predictions;

SELECT * FROM patient_info_view;
```

OUTPUT



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

1. **Normalization:** Apply normalization techniques to organize data efficiently and minimize redundancy. This involves breaking down large tables into smaller ones and establishing relationships between them.

2. **Primary Keys and Foreign Key:** Define appropriate primary keys for each table to uniquely identify rows. Use foreign keys to establish relationships between tables, ensuring referential integrity.

3. **Composite Keys:** Avoid using composite keys unless necessary. Instead, use a single, simple primary key whenever possible.

4. **Data Types:** Choose appropriate data types for each column to optimize storage and prevent data inconsistencies. Ensure that the length of VARCHAR fields is sufficient but not excessive.

5. **Default Values and Constraints:** Use default values for columns whenever applicable to reduce the need for unnecessary data entry.Apply constraints such as NOT NULL, UNIQUE, and CHECK constraints to enforce data integrity rules.

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

Indexing: Create appropriate indexes on columns frequently used in WHERE clauses, JOIN conditions, and ORDER BY clauses. Be mindful of the trade-offs between the number and size of indexes and the performance of data modification operations (INSERT, UPDATE, DELETE).

Query Optimization: Analyze and optimize your queries using tools like EXPLAIN or query execution plans provided by your database system. Ensure that your queries are written efficiently, avoiding unnecessary joins, subqueries, or complex expressions.

Limit the Result Set: Retrieve only the necessary columns and rows by using the SELECT statement judiciously.Consider using the LIMIT clause (or equivalent, depending on the database system) to restrict the number of rows returned, especially for paginated results.

***Avoid SELECT :** Explicitly list the columns you need instead of using SELECT * to avoid fetching unnecessary data.Fetching only the required columns reduce the amount of data transferred and can significantly improve performance.

Normalization and Denormalization: Strike a balance between normalization for data integrity and denormalization for performance.Denormalize data when necessary for read-heavy operations to avoid excessive JOIN operations.