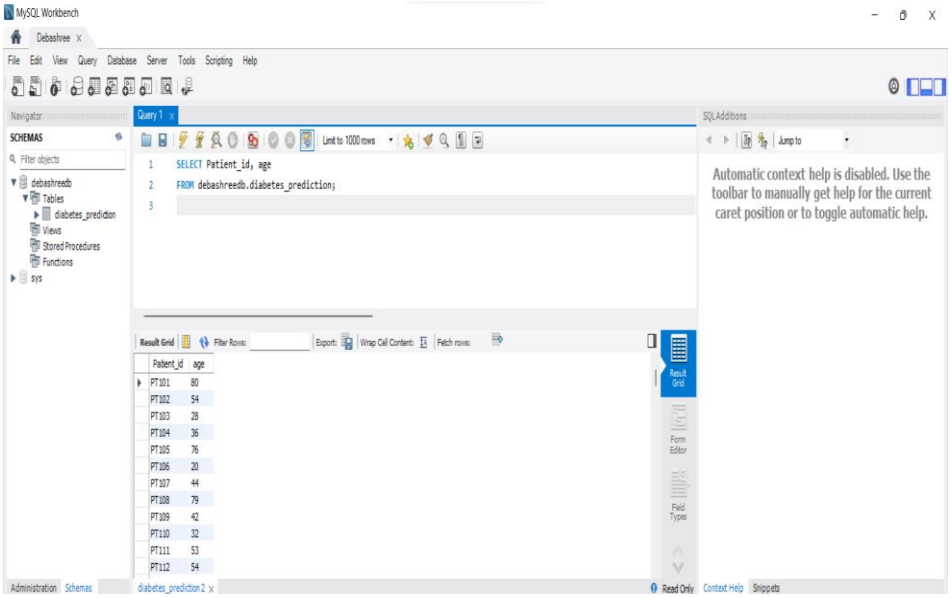
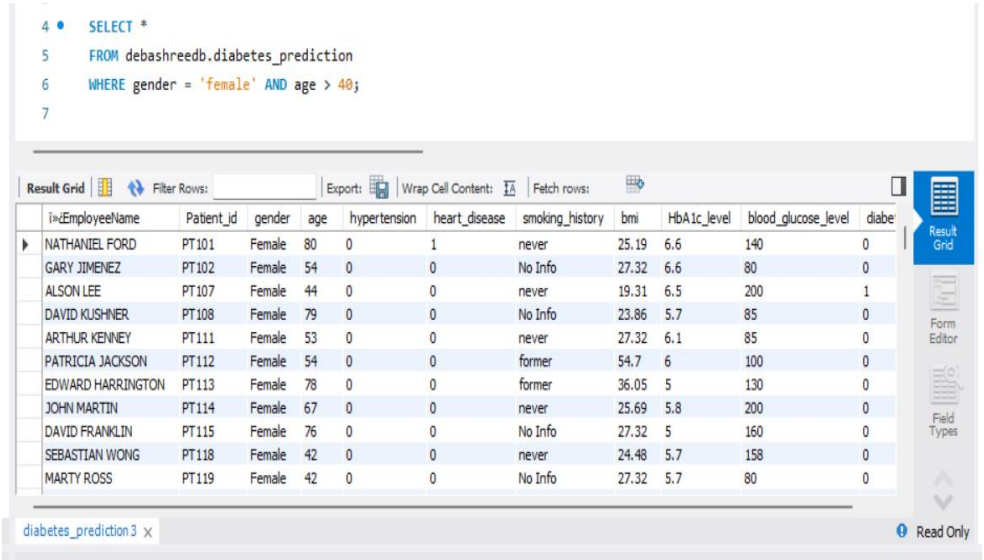
DIABETES PREDICATION ASSESSMENT

NAME- DEBASHREE PRIYA SAHOO

Q1. Retrieve the Patient_id and ages of all patients.



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



Q3. Calculate the average BMI of patients.

```
7
8 •  SELECT AVG(bmi) AS average_bmi
9    FROM debashreedb.diabetes_prediction;
10
11
```

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

average_bmi
27.32076709999422

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

```
11 •  SELECT *
12    FROM debashreedb.diabetes_prediction
13    ORDER BY blood_glucose_level DESC;
14
```

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
Michele A Flowers	PT93343	Female	47	1	0	current	47.23	6.5	300	1
Editha J Pascual	PT91144	Male	38	0	0	never	37.49	9	300	1
Anthony Bruce	PT93259	Male	28	0	0	ever	27.32	7.5	300	1
Nigel L Hicks	PT91250	Female	80	1	0	never	30.38	8.2	300	1
Mary Ann Moran	PT92871	Female	60	0	0	never	53.4	5.8	300	1
Silvia Woo	PT91896	Female	74	0	1	never	25.92	5.7	300	1
Joshua Kumli	PT75052	Male	41	1	0	never	33.08	8.8	300	1
Rita L Kearns	PT76623	Female	42	0	0	never	23.56	6	300	1
Karen K Liu	PT75767	Male	66	0	1	ever	35	6.8	300	1
Francis J Valiquette	PT76586	Female	61	0	0	not current	32.23	9	300	1
Feliks R Gasanyan	PT75103	Male	75	1	0	former	40	8.8	300	1
Adriano E Castro	PT75109	Male	50	0	0	current	30.67	6.6	300	1

Q5. Find patients who have hypertension and diabetes.

```
14
15 •  SELECT *
16    FROM debashreedb.diabetes_prediction
17    WHERE hypertension = 1 AND diabetes = 1;
18
```

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
JONES WONG	PT139	Male	50	1	0	current	27.32	5.7	260	1
PATRIC STEELE	PT205	Female	80	1	0	never	27.32	6.8	280	1
ARTHUR STELLINI	PT343	Male	57	1	1	not current	27.77	6.6	160	1
CHAD LAW	PT355	Male	63	1	0	ever	35.06	5.8	200	1
CATHERINE JAMES	PT451	Female	52	1	0	never	50.3	6.6	155	1
JOHN HART	PT565	Male	48	1	0	current	36.12	6.8	140	1
JOHN BARKER	PT567	Female	79	1	0	former	27.32	6.5	159	1
ROBERT BONNET	PT632	Female	49	1	0	not current	36.93	8.8	155	1
VITANI BENJAMIN	PT727	Male	43	1	0	not current	40.86	6.6	159	1
LANNIE ADELMAN	PT828	Female	38	1	0	not current	27.32	6.1	160	1
JOEL DELIZONNA	PT852	Female	28	1	0	never	20.09	6.6	200	1

Q6. Determine the number of patients with heart disease.

```
19 •  SELECT COUNT(*) AS patient_count_with_heart_disease
20    FROM debashreedb.diabetes_prediction
21    WHERE heart_disease = 1;
```

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

patient_count_with_heart_disease
3942

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

```
23
24 •  SELECT smoking_history, COUNT(*) AS patient_count
25    FROM debashreedb.diabetes_prediction
26    GROUP BY smoking_history;
```

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

smoking_history	patient_count
never	35095
No Info	35816
current	9286
former	9352
ever	4004
not current	6447

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

```
28 •  SELECT Patient_id
29    FROM debashreedb.diabetes_prediction
30    WHERE bmi > (SELECT AVG(bmi) FROM debashreedb.diabetes_prediction);
31
```

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

Patient_id
PT109
PT112
PT113
PT117
PT121
PT124
PT126
PT128
PT131
PT140
PT143
PT144

diabetes_prediction 9 | Read Only | Context H

Q9. Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.

```
36    -- Patient with the highest HbA1c level
37 •  SELECT *
38    FROM debashreedb.diabetes_prediction
39    ORDER BY HbA1c_level DESC
```

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

ï»¿EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Seth I Rubenstein	PT98911	Female	60	0	0	current	40.18	9	300	1
Carla Y Ovando	PT99155	Female	53	0	0	never	27.32	9	160	1
Robert A Young	PT99175	Female	57	0	0	former	39.58	9	260	1
Guadalupe T Perez	PT99266	Female	68	0	0	never	28.6	9	280	1
Yolanda Broussard	PT99298	Female	61	0	0	never	22.95	9	240	1
Medrine A Baltazar	PT99442	Female	80	0	0	never	20.99	9	126	1
Miao Ling Huang	PT99613	Male	60	0	0	former	27.32	9	280	1
Angelica J Young	PT99764	Male	80	0	0	No Info	34	9	300	1
Sidney Green	PT99807	Male	54	1	0	current	23.06	9	126	1
George B Peng	PT99841	Male	80	0	1	never	25.76	9	220	1
Jonathan S Chiu	PT97400	Male	71	0	1	never	25	9	130	1

diabetes_prediction 14 | Read Only

```
36    -- Patient with the lowest HbA1c level
37 •  SELECT *
38    FROM debashreedb.diabetes_prediction
39    ORDER BY HbA1c_level ASC
40
```

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

ï»¿EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Dante O Mendoza	PT98694	Male	54	0	0	never	24.34	3.5	159	0
Petra O Brady	PT98725	Female	49	0	0	never	30.29	3.5	130	0
Marisol A Pastran	PT98727	Male	46	0	0	former	33.31	3.5	85	0
Martin G Espinosa	PT98741	Female	67	0	0	never	23.48	3.5	85	0
Laura M Aguirre	PT98748	Male	31	0	0	not current	19.01	3.5	160	0
Kenneth B Buncum Jr	PT98750	Female	60	0	0	never	23.32	3.5	155	0
Mikaela Merchant	PT98783	Female	53	1	0	No Info	26.95	3.5	130	0
Connie W Kwan	PT98803	Female	75	0	0	former	18.91	3.5	90	0
Letitia Tsui	PT98823	Female	23	0	0	never	23	3.5	158	0
Franz Brustmeyer	PT98827	Female	51	0	0	current	49.11	3.5	100	0
Maria Lourdes R Car...	PT98840	Female	0	0	0	No Info	15.14	3.5	158	0

diabetes_prediction 15 | Read Only

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

```
38 •  SELECT Patient_id,
39          age,
40          DATE_SUB(NOW(), INTERVAL age YEAR)AS birthdate
41    FROM debashreedb.diabetes_prediction;
42
```

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

Patient_id	age	birthdate
PT101	80	1944-01-08 22:37:06
PT102	54	1970-01-08 22:37:06
PT103	28	1996-01-08 22:37:06
PT104	36	1988-01-08 22:37:06
PT105	76	1948-01-08 22:37:06
PT106	20	2004-01-08 22:37:06
PT107	44	1980-01-08 22:37:06
PT108	79	1945-01-08 22:37:06
PT109	42	1982-01-08 22:37:06
PT110	32	1992-01-08 22:37:06
PT111	53	1971-01-08 22:37:06
PT112	54	1970-01-08 22:37:06

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

```
43 •  SELECT
44        Patient_id,
45        gender,
46        blood_glucose_level,
47        RANK() OVER (PARTITION BY gender ORDER BY blood_glucose_level DESC) AS glucose_level_rank
48    FROM
49        debashreedb.diabetes_prediction;
50
```

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

Patient_id	gender	blood_glucose_level	glucose_level_rank
PT97622	Female	300	1
PT96814	Female	300	1
PT96815	Female	300	1
PT97708	Female	300	1
PT96902	Female	300	1
PT97955	Female	300	1
PT97141	Female	300	1
PT96371	Female	300	1
PT98911	Female	300	1
PT98454	Female	300	1
PT96346	Female	300	1
PT97671	Female	300	1
PT97820	Female	300	1

Result Grid | Form Editor | Field Types

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

```
57 •  UPDATE debashreedb.diabetes_prediction
58    SET smoking_history = 'Ex-smoker'
59    WHERE age > 50;
60
61
62 •  SELECT Patient_id, smoking_history
63    FROM
64        debashreedb.diabetes_prediction where age > 50;
65
```

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

Patient_id	smoking_history
PT101	Ex-smoker
PT102	Ex-smoker
PT105	Ex-smoker
PT108	Ex-smoker
PT111	Ex-smoker
PT112	Ex-smoker
PT113	Ex-smoker
PT114	Ex-smoker
PT115	Ex-smoker
PT116	Ex-smoker
PT123	Ex-smoker
PT124	Ex-smoker
PT127	Ex-smoker

Result Grid | Form Editor | Field Types

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

```
66 •      INSERT INTO debashreedb.diabetes_prediction VALUES ('John Doe', 'P123456', 'male', 55, 1, 0, 'Non-smoker', 25.5, 6
67
68 •  SELECT *
69    FROM debashreedb.diabetes_prediction
70    WHERE Patient_id = 'P123456';
71
```

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

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
John Doe	P123456	male	55	1	0	Non-smoker	25.5	6.2	120	0

Result Grid | Form Editor | Field Types

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

```
72 •  DELETE FROM debashreedb.diabetes_prediction
73    WHERE heart_disease = 1;
74
75 •  SELECT *
76    FROM debashreedb.diabetes_prediction;
77
```

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	diabet
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
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
DAVID KUSHNER	PT108	Female	79	0	0	Ex-smoker	23.86	5.7	85	0
MICHAEL MORRIS	PT109	Male	42	0	0	never	33.64	4.8	145	0
JOANNE HAYES-WHITE	PT110	Female	32	0	0	never	27.32	5	100	0
ARTHUR KENNEY	PT111	Female	53	0	0	Ex-smoker	27.32	6.1	85	0
PATRICIA JACKSON	PT112	Female	54	0	0	Ex-smoker	54.7	6	100	0
EDWARD HARRINGTON	PT113	Female	78	0	0	Ex-smoker	36.05	5	130	0
JOHN MARTIN	PT114	Female	67	0	0	Ex-smoker	25.69	5.8	200	0

Result Grid | Form Editor | Field Types

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

```
78    SELECT Patient_id, hypertension
79    FROM debashreedb.diabetes_prediction
80    WHERE hypertension = 1
81    EXCEPT
82    SELECT Patient_id, hypertension
83    FROM debashreedb.diabetes_prediction
84    WHERE diabetes = 1;
```

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

Patient_id	hypertension
PT129	1
PT155	1
PT161	1
PT215	1
PT227	1
PT241	1
PT326	1
PT339	1
PT357	1
PT377	1
PT379	1
PT446	1
PT474	1

Result 22 | Result Grid | Form Editor | Field Types | Read Only

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

```
86 •  ALTER TABLE debashreedb.diabetes_prediction
87    MODIFY COLUMN Patient_id VARCHAR(255);
88
89 •  ALTER TABLE debashreedb.diabetes_prediction
90    ADD CONSTRAINT unique_patient_id UNIQUE (Patient_id(255));
91
```

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

```
92 •  USE debashreedb;
93 •  CREATE VIEW PatientInfoView AS
94    SELECT Patient_id, age, bmi
95    FROM debashreedb.diabetes_prediction;
96 •  SELECT * FROM PatientInfoView;
```

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

Patient_id	age	bmi
PT102	54	27.32
PT103	28	27.32
PT104	36	23.45
PT106	20	27.32
PT107	44	19.31
PT108	79	23.86
PT109	42	33.64
PT110	32	27.32
PT111	53	27.32
PT112	54	54.7
PT113	78	36.05
PT114	67	25.69
PT115	76	27.32

Result Grid | Form Editor | Field Types | Read Only

PatientInfoView 23

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

- Apply normalization techniques (1NF, 2NF, 3NF, etc.) to eliminate redundancy and organize data efficiently. Break down tables into smaller, related tables to store data in a structured manner.
- Use appropriate primary keys for each table to ensure uniqueness. Avoid using columns with potentially duplicated values as primary keys.
- Establish foreign key relationships between tables to enforce referential integrity. Foreign keys help maintain consistency and prevent orphaned records.
- Review the schema for redundant columns and eliminate them. Redundant columns can lead to unnecessary data duplication and potential inconsistencies.
- Consider denormalization for performance optimization in situations where read performance is critical.

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

- Ensure that appropriate indexes are created on columns frequently used in WHERE clauses and JOIN conditions.

- Optimize JOIN conditions and use INNER JOINs where possible. Ensure that foreign key columns are indexed to speed up JOIN operations.
- Use the LIMIT or TOP clause to restrict the number of rows returned, especially during testing and development.
- Ensure that the data structure is normalized to minimize redundancy and improve query performance. Normalize tables to eliminate duplicate data.
- Avoid using functions in the WHERE clause, as this can hinder the use of indexes. Instead, try to compare columns directly.