Project Report: Exploratory Data Analysis on Diabetes Dataset

DATA UNDERSTANDING

The dataset contains information about patients with diabetes, including various features such as age, BMI, glucose levels, insulin levels, skin thickness, blood pressure, and diabetes status. Each row represents a unique patient, and there are several numeric and categorical attributes associated with each patient.

PROJECT OBJECTIVE

The main objective of this project is to perform exploratory data analysis (EDA) on the diabetes dataset using SQL for data manipulation. The goal is to gain insights into the relationships between different variables and their impact on diabetes diagnosis. Additionally, we aim to identify patterns, trends, and potential factors contributing to diabetes among patients.

TOOLS

- Microsoft SQL Server Management Studio(SSMS) - data manipulation and Analysis.

DATA SOURCES

The primary dataset employed for this analysis, extracted from Meriskill "Diabetes.csv" documents the cases of diabetes patients.

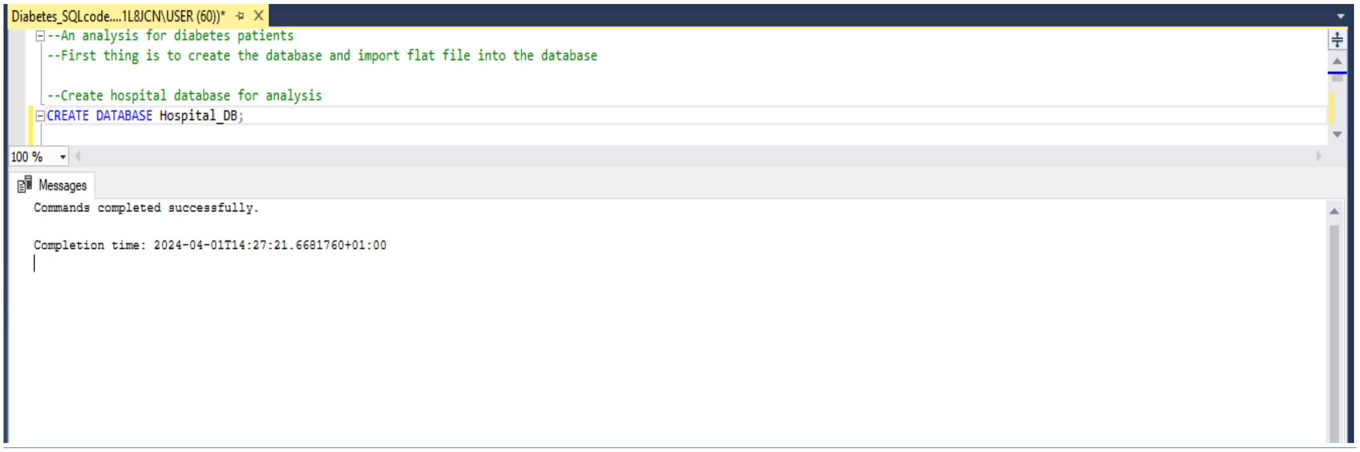
DATA REPROCESSING

- Checked for missing values.
- Checked for duplicates which none was found.
- Checked for inconsistent data or characters and none was found

PART 1: DATA IMPORT STEPS

DATABASE CREATION:

Created a new database named Hospital_DB in SQL Server Management Studio (SSMS)



```sql
--An analysis for diabetes patients
--First thing is to create the database and import flat file into the database

--Create hospital database for analysis
CREATE DATABASE Hospital_DB;
```

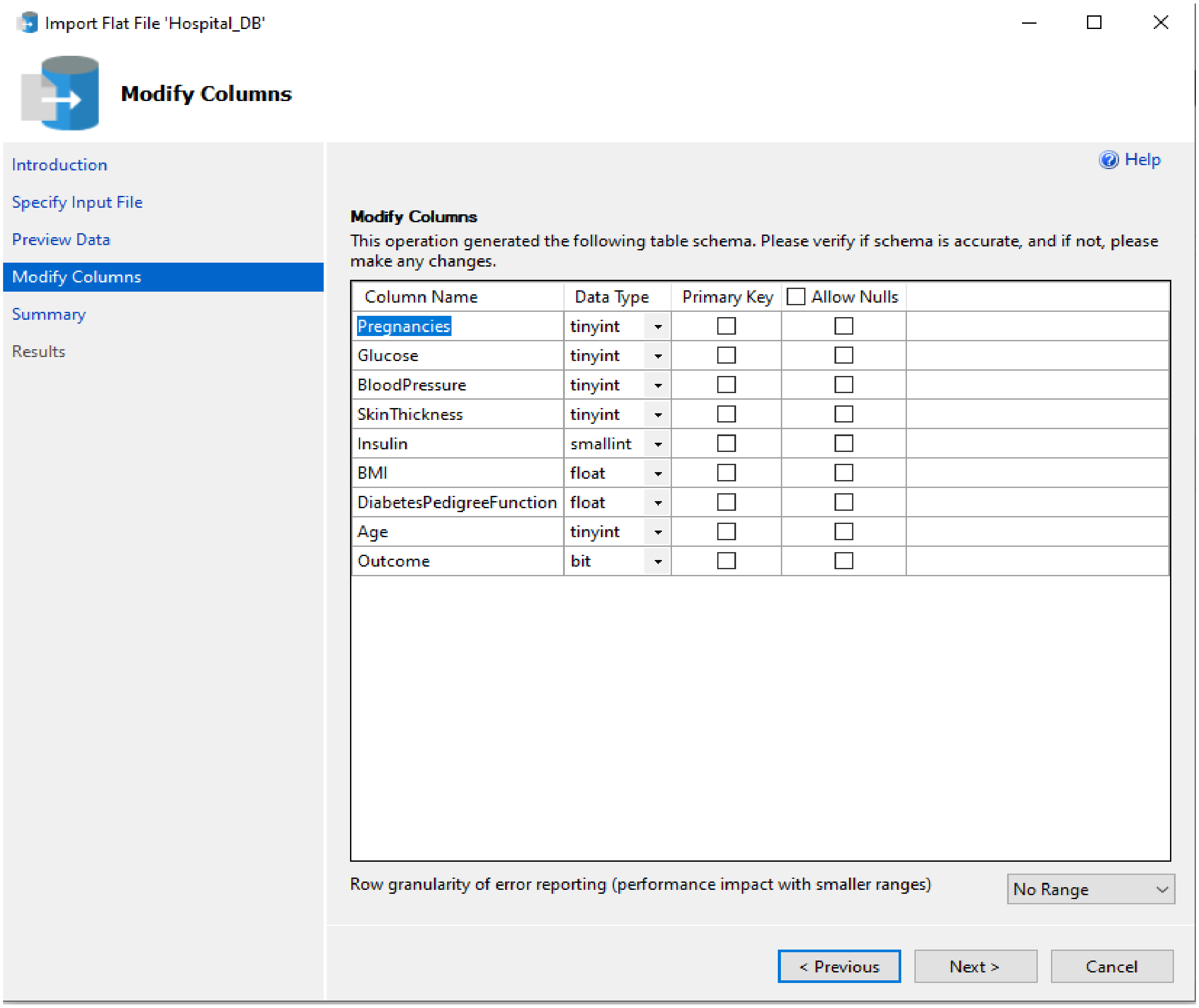
```
Commands completed successfully.

Completion time: 2024-04-01T14:27:21.6681760+01:00
```

DATA IMPORT USING IMPORT FLAT FILE:

The data import process involved employing the "Import Flat File" wizard in SSMS to seamlessly transfer data from the provided CSV files into their respective tables within the Hospital database.

- This entailed meticulously mapping columns from the CSV files to their corresponding counterparts in the database tables.
- Special attention was paid to preserving column names throughout the import process to facilitate future usability and maintenance.



Modify Columns

This operation generated the following table schema. Please verify if schema is accurate, and if not, please make any changes.

Column Name	Data Type	Primary Key	□ Allow Nulls
Pregnancies	tinyint	□	□
Glucose	tinyint	□	□
BloodPressure	tinyint	□	□
SkinThickness	tinyint	□	□
Insulin	smallint	□	□
BMI	float	□	□
DiabetesPedigreeFunction	float	□	□
Age	tinyint	□	□
Outcome	bit	□	□

Row granularity of error reporting (performance impact with smaller ranges) No Range

PART 2: DATA MANIPULATION

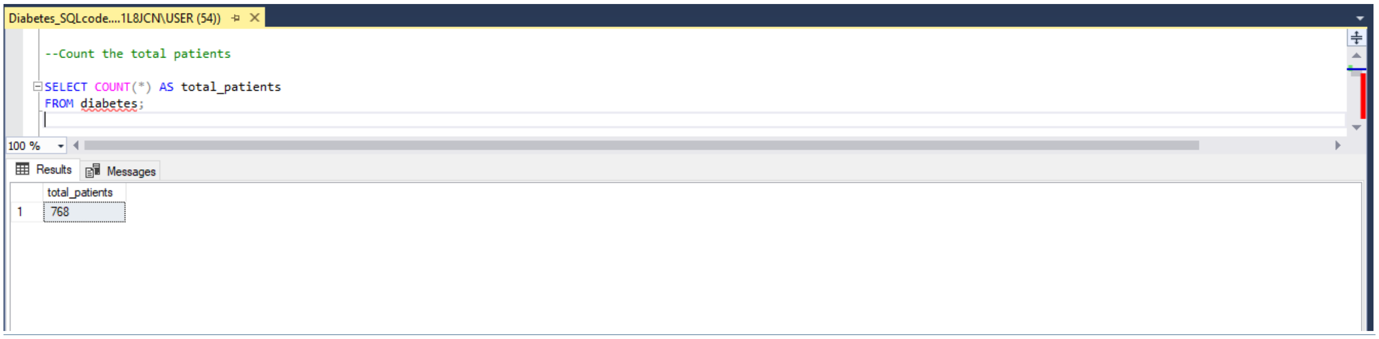
1. Count all patients

CODE

```
--Count the total patients

SELECT COUNT(*) AS total_patients
FROM diabetes;
```

RESULT



```
--Count the total patients

SELECT COUNT(*) AS total_patients
FROM diabetes;
```

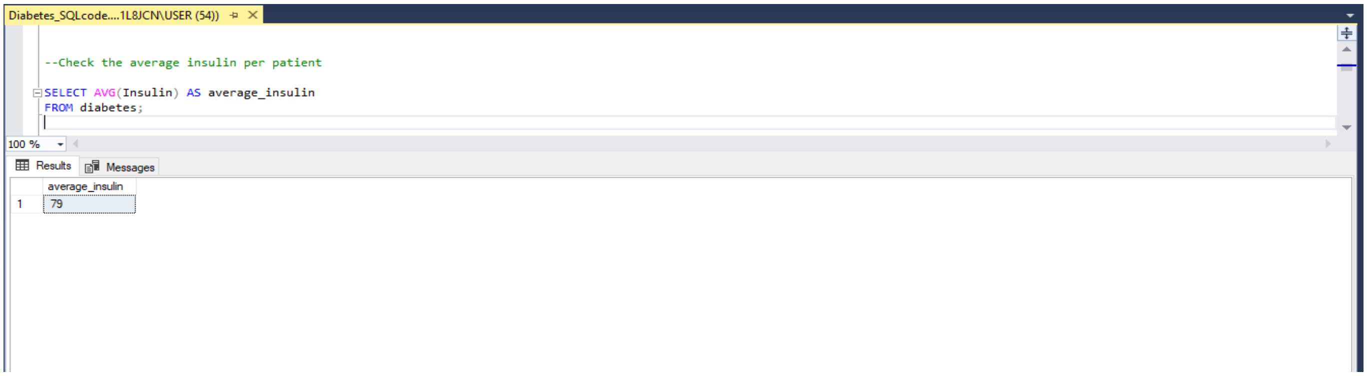
```
   total_patients
1  768
```

2. Average Insulin per patient

CODE

```
--Check the average insulin per patient
SELECT AVG(Insulin) AS average_insulin
FROM diabetes;
```

RESULT



```
--Check the average insulin per patient

SELECT AVG(Insulin) AS average_insulin
FROM diabetes;
```

```
   average_insulin
1  79
```

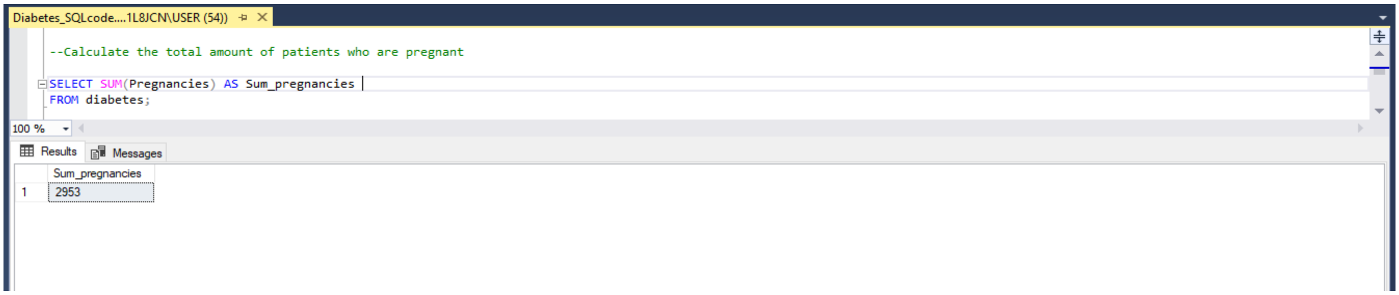
3. Total Amount of Patients who are pregnant

CODE

```
--Calculate the total amount of patients who are pregnant

SELECT SUM(Pregnancies) AS Sum_pregnancies
FROM diabetes;
```

RESULT



```
--Calculate the total amount of patients who are pregnant

SELECT SUM(Pregnancies) AS Sum_pregnancies
FROM diabetes;
```

```
   Sum_pregnancies
1  2953
```

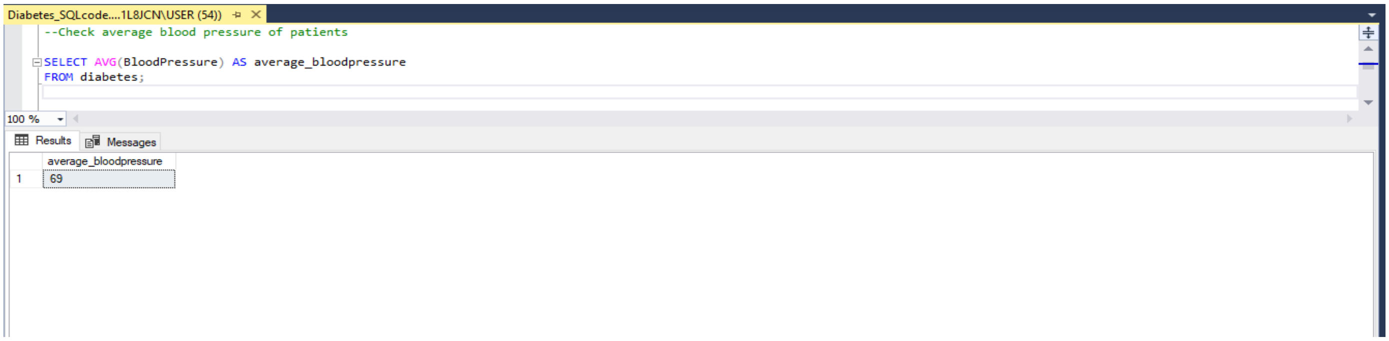
4. The Average blood pressure of patients

CODE

```sql
--Check the average blood pressure of patients

SELECT AVG(BloodPressure) AS average_bloodpressure
FROM diabetes;
```

RESULT



```sql
--Check average blood pressure of patients

SELECT AVG(BloodPressure) AS average_bloodpressure
FROM diabetes;
```

	average_bloodpressure
1	69

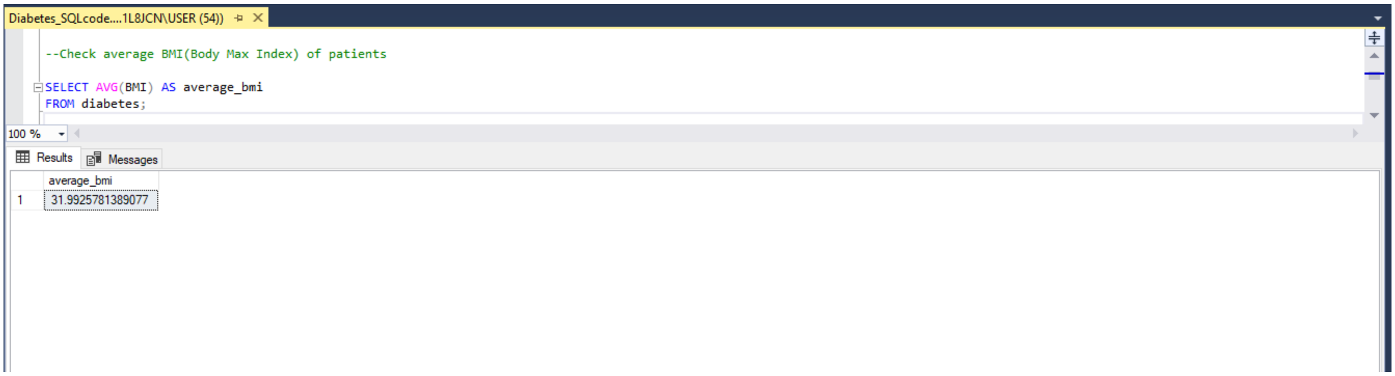
5. The Average Body Max Index (BMI) of Patients

CODE

```sql
--Check average BMI (Body Max Index) of patients

SELECT AVG(BMI) AS average_bmi
FROM diabetes;
```

RESULT



```
--Check average BMI(Body Max Index) of patients

SELECT AVG(BMI) AS average_bmi
FROM diabetes;
```

```
  average_bmi
1 31.9925781389077
```

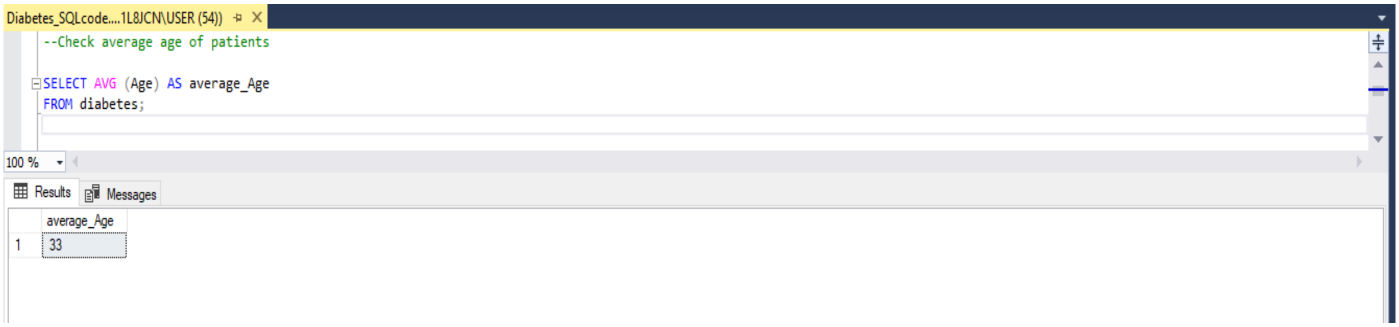
6. The Average Age of Patients

CODE

```
--Check average age of patients

SELECT AVG (Age) AS average_Age
FROM diabetes;
```

RESULT



```
--Check average age of patients

SELECT AVG (Age) AS average_Age
FROM diabetes;
```

```
  average_Age
1 33
```

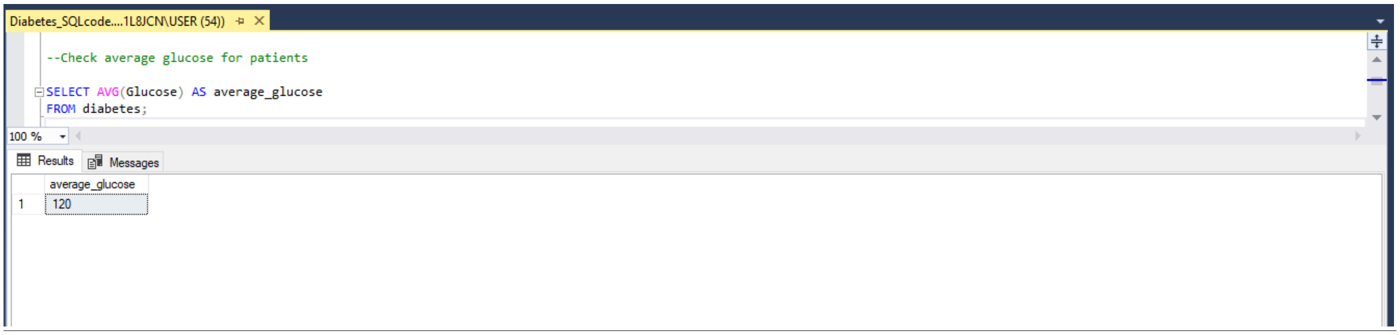
7. The Average Glucose for Patients

CODE

```
--Check average glucose for patients

SELECT AVG(Glucose) AS average_glucose
FROM diabetes;
```

RESULT



```
--Check average glucose for patients

SELECT AVG(Glucose) AS average_glucose
FROM diabetes;
```

```
   average_glucose
1  120
```

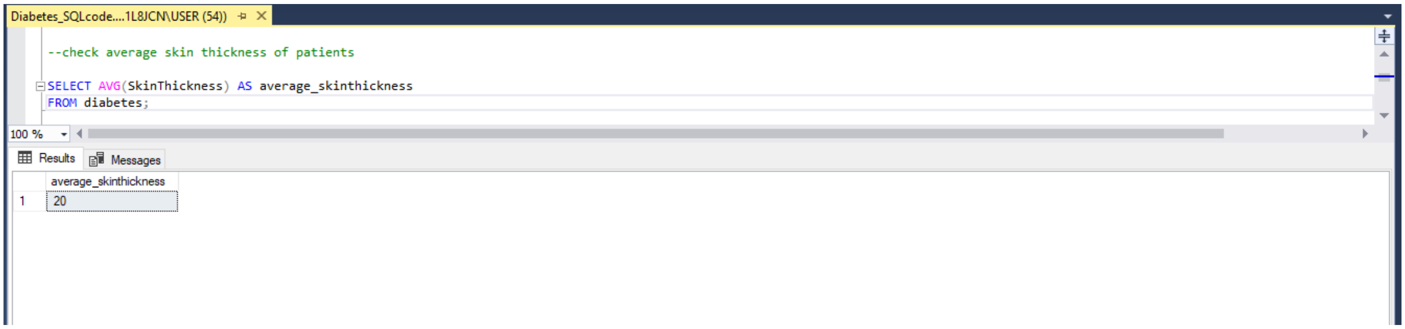
8. The Average Skin Thickness of Patients

CODE

```
--check average skin thickness of patients

SELECT AVG(SkinThickness) AS average_skinthickness
FROM diabetes;
```

RESULT



```
--check average skin thickness of patients

SELECT AVG(SkinThickness) AS average_skinthickness
FROM diabetes;
```

```
   average_skinthickness
1  20
```

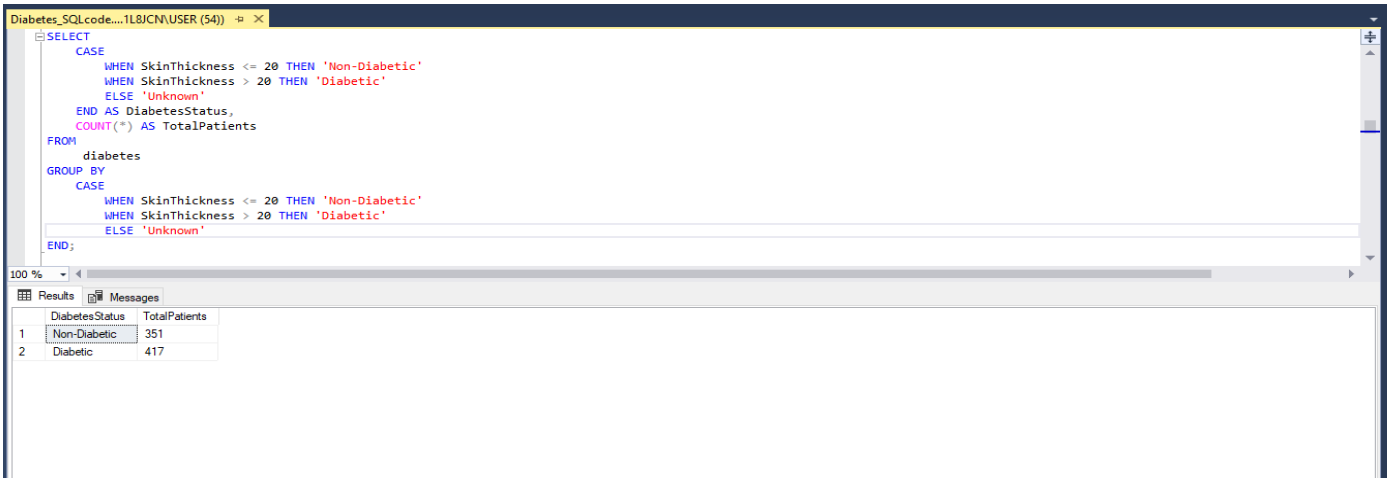
9. Diabetes status based on skin thickness

CODE

```
--Check diabetes status on skin thickness

SELECT
    CASE
        WHEN SkinThickness <= 20 THEN 'Non-Diabetic'
        WHEN SkinThickness > 20 THEN 'Diabetic'
        ELSE 'Unknown'
    END AS DiabetesStatus,
    COUNT(*) AS TotalPatients
FROM
     diabetes
GROUP BY
    CASE
        WHEN SkinThickness <= 20 THEN 'Non-Diabetic'
        WHEN SkinThickness > 20 THEN 'Diabetic'
        ELSE 'Unknown'
END;
```

RESULT



```
SELECT
    CASE
        WHEN SkinThickness <= 20 THEN 'Non-Diabetic'
        WHEN SkinThickness > 20 THEN 'Diabetic'
        ELSE 'Unknown'
    END AS DiabetesStatus,
    COUNT(*) AS TotalPatients
FROM
     diabetes
GROUP BY
    CASE
        WHEN SkinThickness <= 20 THEN 'Non-Diabetic'
        WHEN SkinThickness > 20 THEN 'Diabetic'
        ELSE 'Unknown'
END;
```

	DiabetesStatus	TotalPatients
1	Non-Diabetic	351
2	Diabetic	417

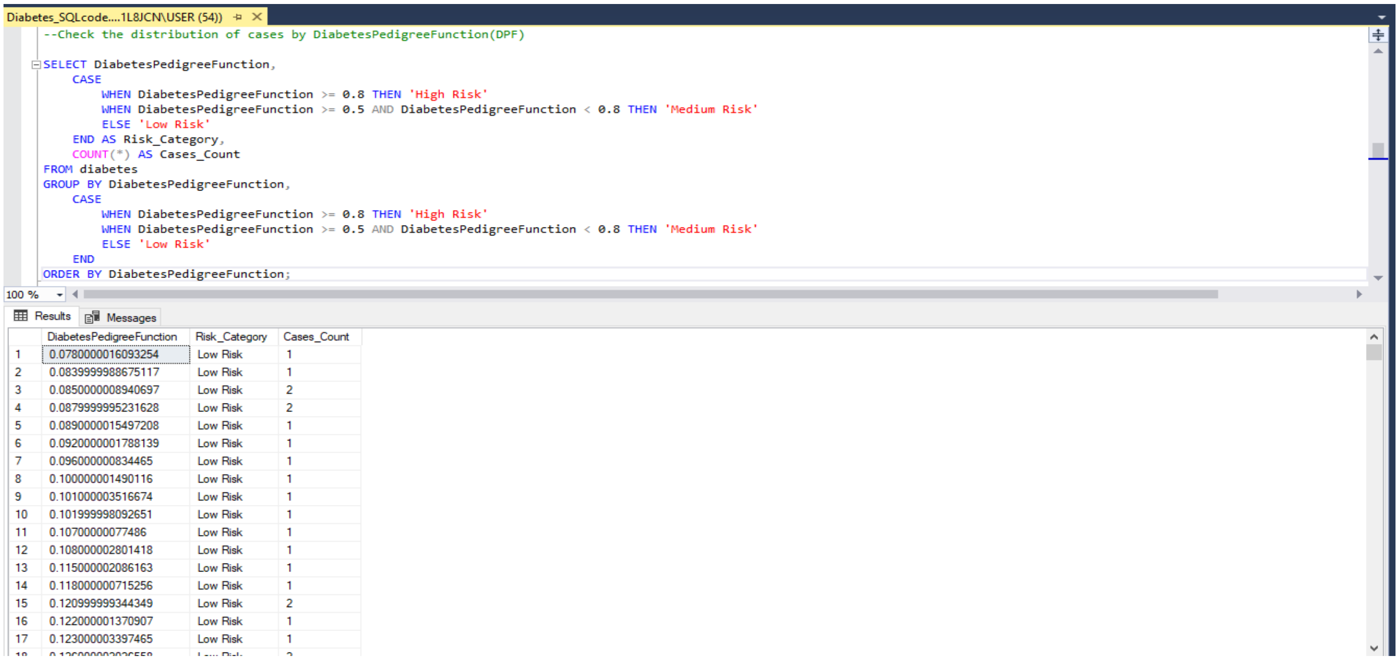
10. The distribution of cases by DiabetesPedigreeFunction(DPF)

CODE

```
--Check the distribution of cases by DiabetesPedigreeFunction(DPF)
```

```
SELECT DiabetesPedigreeFunction,
    CASE
        WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
        WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN
'Medium Risk'
        ELSE 'Low Risk'
    END AS Risk_Category,
    COUNT(*) AS Cases_Count
FROM diabetes
GROUP BY DiabetesPedigreeFunction,
    CASE
        WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
        WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN
'Medium Risk'
        ELSE 'Low Risk'
    END
ORDER BY DiabetesPedigreeFunction;
```

RESULT



```
--Check the distribution of cases by DiabetesPedigreeFunction(DPF)

SELECT DiabetesPedigreeFunction,
    CASE
        WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
        WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN 'Medium Risk'
        ELSE 'Low Risk'
    END AS Risk_Category,
    COUNT(*) AS Cases_Count
FROM diabetes
GROUP BY DiabetesPedigreeFunction,
    CASE
        WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
        WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN 'Medium Risk'
        ELSE 'Low Risk'
    END
ORDER BY DiabetesPedigreeFunction;
```

	DiabetesPedigreeFunction	Risk_Category	Cases_Count
1	0.0780000016093254	Low Risk	1
2	0.0839999988675117	Low Risk	1
3	0.0850000008940697	Low Risk	2
4	0.0879999995231628	Low Risk	2
5	0.0890000015497208	Low Risk	1
6	0.0920000001788139	Low Risk	1
7	0.096000000834465	Low Risk	1
8	0.100000001490116	Low Risk	1
9	0.101000003516674	Low Risk	1
10	0.101999998092651	Low Risk	1
11	0.10700000077486	Low Risk	1
12	0.108000002801418	Low Risk	1
13	0.115000002086163	Low Risk	1
14	0.118000000715256	Low Risk	1
15	0.120999999344349	Low Risk	2
16	0.122000001370907	Low Risk	1
17	0.123000003397465	Low Risk	1

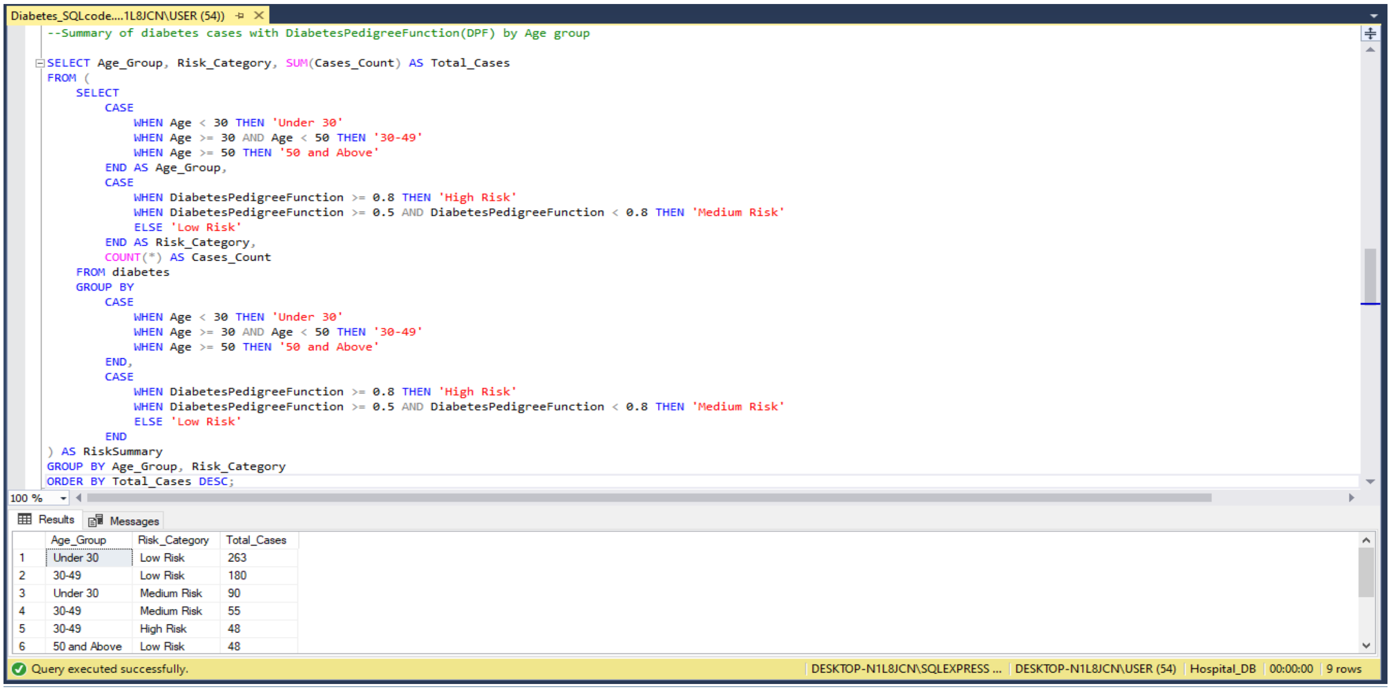
11. Summary of diabetes cases with DiabetesPedigreeFunction(DPF) by Age group

CODE

```
--Summary of diabetes cases with DiabetesPedigreeFunction(DPF) by Age group

SELECT Age_Group, Risk_Category, SUM(Cases_Count) AS Total_Cases
FROM (
    SELECT
        CASE
            WHEN Age < 30 THEN 'Under 30'
            WHEN Age >= 30 AND Age < 50 THEN '30-49'
            WHEN Age >= 50 THEN '50 and Above'
        END AS Age_Group,
        CASE
            WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
            WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN
'Medium Risk'
            ELSE 'Low Risk'
        END AS Risk_Category,
        COUNT(*) AS Cases_Count
    FROM diabetes
    GROUP BY
        CASE
            WHEN Age < 30 THEN 'Under 30'
            WHEN Age >= 30 AND Age < 50 THEN '30-49'
            WHEN Age >= 50 THEN '50 and Above'
        END,
        CASE
            WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
            WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN
'Medium Risk'
            ELSE 'Low Risk'
        END
) AS RiskSummary
GROUP BY Age_Group, Risk_Category
ORDER BY Total_Cases DESC;
```

RESULT



```
--Summary of diabetes cases with DiabetesPedigreeFunction(DPF) by Age group

SELECT Age_Group, Risk_Category, SUM(Cases_Count) AS Total_Cases
FROM (
    SELECT
        CASE
            WHEN Age < 30 THEN 'Under 30'
            WHEN Age >= 30 AND Age < 50 THEN '30-49'
            WHEN Age >= 50 THEN '50 and Above'
        END AS Age_Group,
        CASE
            WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
            WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN 'Medium Risk'
            ELSE 'Low Risk'
        END AS Risk_Category,
        COUNT(*) AS Cases_Count
    FROM diabetes
    GROUP BY
        CASE
            WHEN Age < 30 THEN 'Under 30'
            WHEN Age >= 30 AND Age < 50 THEN '30-49'
            WHEN Age >= 50 THEN '50 and Above'
        END,
        CASE
            WHEN DiabetesPedigreeFunction >= 0.8 THEN 'High Risk'
            WHEN DiabetesPedigreeFunction >= 0.5 AND DiabetesPedigreeFunction < 0.8 THEN 'Medium Risk'
            ELSE 'Low Risk'
        END
) AS RiskSummary
GROUP BY Age_Group, Risk_Category
ORDER BY Total_Cases DESC;
```

	Age_Group	Risk_Category	Total_Cases
1	Under 30	Low Risk	263
2	30-49	Low Risk	180
3	Under 30	Medium Risk	90
4	30-49	Medium Risk	55
5	30-49	High Risk	48
6	50 and Above	Low Risk	48

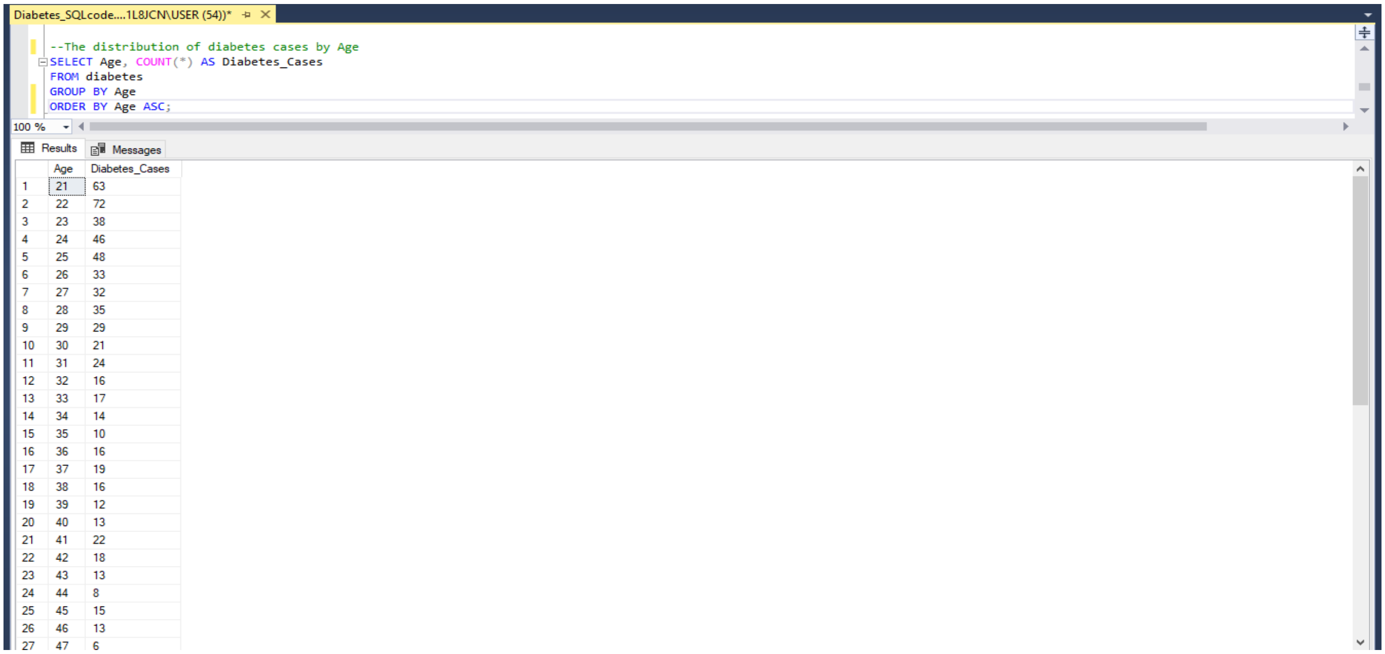
Query executed successfully. | DESKTOP-N1L8JCN\SQLEXPRESS ... | DESKTOP-N1L8JCN\USER (54) | Hospital_DB | 00:00:00 | 9 rows

12. The distribution of diabetes cases by Age

CODE

```
--The distribution of diabetes cases by Age
SELECT Age, COUNT(*) AS Diabetes_Cases
FROM diabetes
GROUP BY Age;
```

RESULT



```
--The distribution of diabetes cases by Age
SELECT Age, COUNT(*) AS Diabetes_Cases
FROM diabetes
GROUP BY Age
ORDER BY Age ASC;
```

	Age	Diabetes_Cases
1	21	63
2	22	72
3	23	38
4	24	46
5	25	48
6	26	33
7	27	32
8	28	35
9	29	29
10	30	21
11	31	24
12	32	16
13	33	17
14	34	14
15	35	10
16	36	16
17	37	19
18	38	16
19	39	12
20	40	13
21	41	22
22	42	18
23	43	13
24	44	8
25	45	15
26	46	13
27	47	6

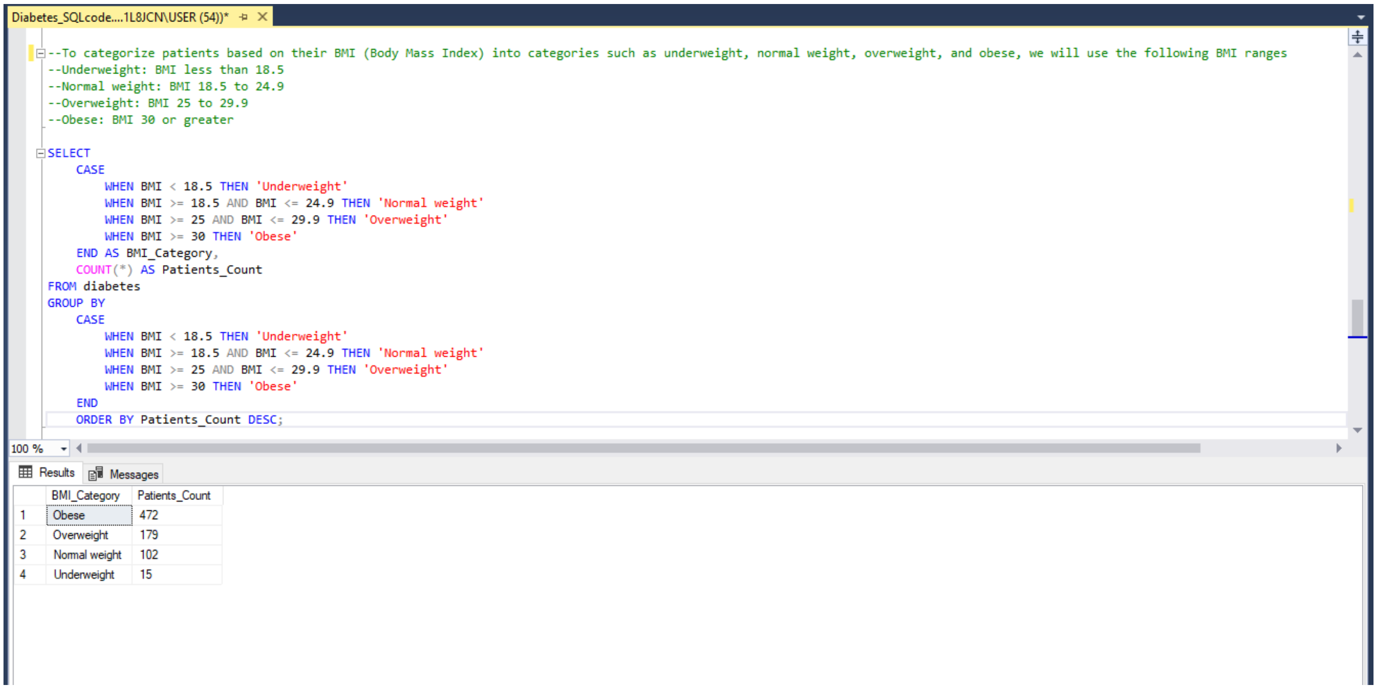
13. categorize patients based on their BMI (Body Mass Index)

CODE

```
--To categorize patients based on their BMI (Body Mass Index) into categories such as
underweight, normal weight, overweight, and obese, we will use the following BMI
ranges
--Underweight: BMI less than 18.5
--Normal weight: BMI 18.5 to 24.9
--Overweight: BMI 25 to 29.9
--Obese: BMI 30 or greater

SELECT
    CASE
        WHEN BMI < 18.5 THEN 'Underweight'
        WHEN BMI >= 18.5 AND BMI <= 24.9 THEN 'Normal weight'
        WHEN BMI >= 25 AND BMI <= 29.9 THEN 'Overweight'
        WHEN BMI >= 30 THEN 'Obese'
    END AS BMI_Category,
    COUNT(*) AS Patients_Count
FROM diabetes
GROUP BY
    CASE
        WHEN BMI < 18.5 THEN 'Underweight'
        WHEN BMI >= 18.5 AND BMI <= 24.9 THEN 'Normal weight'
        WHEN BMI >= 25 AND BMI <= 29.9 THEN 'Overweight'
        WHEN BMI >= 30 THEN 'Obese'
    END
       ORDER BY Patients_Count DESC;
```

RESULT



```
--To categorize patients based on their BMI (Body Mass Index) into categories such as underweight, normal weight, overweight, and obese, we will use the following BMI ranges
--Underweight: BMI less than 18.5
--Normal weight: BMI 18.5 to 24.9
--Overweight: BMI 25 to 29.9
--Obese: BMI 30 or greater

SELECT
    CASE
        WHEN BMI < 18.5 THEN 'Underweight'
        WHEN BMI >= 18.5 AND BMI <= 24.9 THEN 'Normal weight'
        WHEN BMI >= 25 AND BMI <= 29.9 THEN 'Overweight'
        WHEN BMI >= 30 THEN 'Obese'
    END AS BMI_Category,
    COUNT(*) AS Patients_Count
FROM diabetes
GROUP BY
    CASE
        WHEN BMI < 18.5 THEN 'Underweight'
        WHEN BMI >= 18.5 AND BMI <= 24.9 THEN 'Normal weight'
        WHEN BMI >= 25 AND BMI <= 29.9 THEN 'Overweight'
        WHEN BMI >= 30 THEN 'Obese'
    END
    ORDER BY Patients_Count DESC;
```

	BMI_Category	Patients_Count
1	Obese	472
2	Overweight	179
3	Normal weight	102
4	Underweight	15

14. Analyze the distribution of diabetes patients by Age and outcome

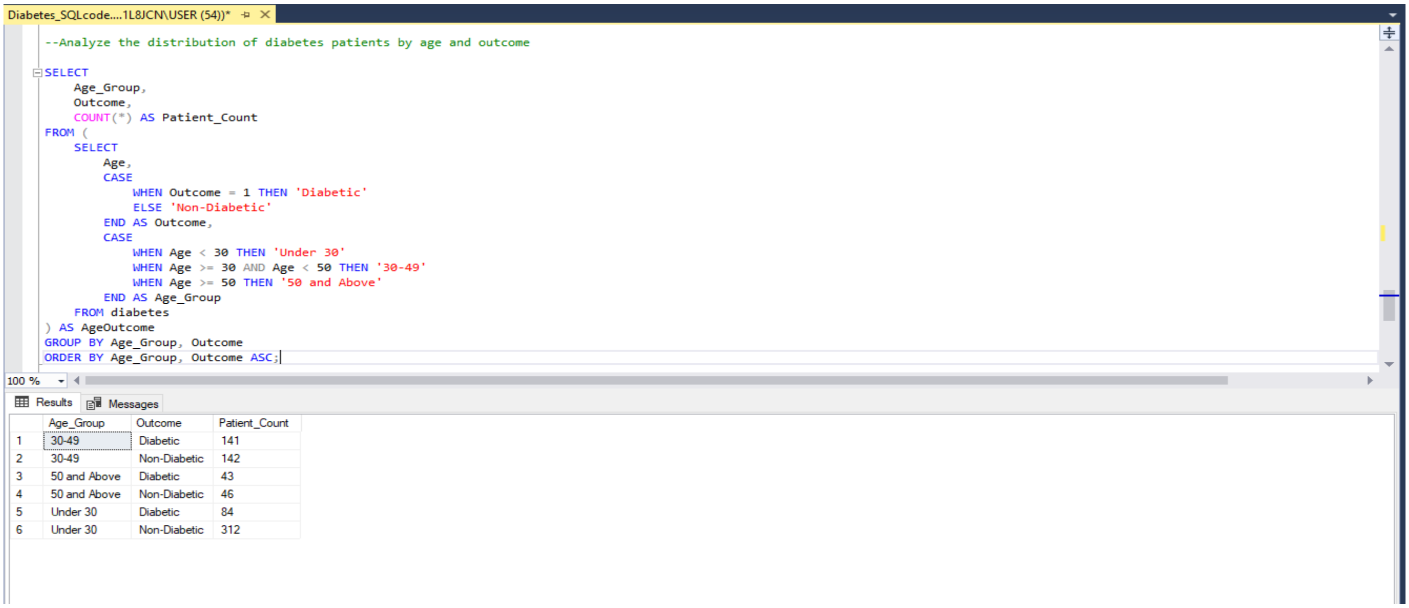
CODE

```
--Analyze the distribution of diabetes patients by age and outcome

SELECT
    Age_Group,
    Outcome,
    COUNT(*) AS Patient_Count
FROM (
    SELECT
        Age,
        CASE
            WHEN Outcome = 1 THEN 'Diabetic'
            ELSE 'Non-Diabetic'
        END AS Outcome,
```

```sql
            CASE
                WHEN Age < 30 THEN 'Under 30'
                WHEN Age >= 30 AND Age < 50 THEN '30-49'
                WHEN Age >= 50 THEN '50 and Above'
            END AS Age_Group
        FROM diabetes
) AS AgeOutcome
GROUP BY Age_Group, Outcome
ORDER BY Age_Group, Outcome ASC;
```

RESULT



```sql
--Analyze the distribution of diabetes patients by age and outcome

SELECT
    Age_Group,
    Outcome,
    COUNT(*) AS Patient_Count
FROM (
    SELECT
        Age,
        CASE
            WHEN Outcome = 1 THEN 'Diabetic'
            ELSE 'Non-Diabetic'
        END AS Outcome,
        CASE
            WHEN Age < 30 THEN 'Under 30'
            WHEN Age >= 30 AND Age < 50 THEN '30-49'
            WHEN Age >= 50 THEN '50 and Above'
        END AS Age_Group
    FROM diabetes
) AS AgeOutcome
GROUP BY Age_Group, Outcome
ORDER BY Age_Group, Outcome ASC;
```

	Age_Group	Outcome	Patient_Count
1	30-49	Diabetic	141
2	30-49	Non-Diabetic	142
3	50 and Above	Diabetic	43
4	50 and Above	Non-Diabetic	46
5	Under 30	Diabetic	84
6	Under 30	Non-Diabetic	312

CONCLUSION

In conclusion, exploratory data analysis on the diabetes dataset using SQL has provided valuable insights into the factors influencing diabetes diagnosis and prevalence among patients. By leveraging SQL for data manipulation and analysis, we have gained a deeper understanding of the relationships between various variables and their implications for diabetes risk assessment and management. Moving forward, these insights can inform targeted interventions, public health initiatives, and personalized healthcare strategies aimed at preventing and managing diabetes effectively.

RECOMMENDATIONS

Based on our exploratory data analysis, we provide the following recommendations:

- **Risk Factors Identification:** Identify the key risk factors associated with diabetes based on the analysis of the dataset. Factors such as high glucose levels, insulin resistance, elevated BMI, and abnormal blood pressure may indicate an increased risk of diabetes.

- **Early Detection Strategies:** Develop early detection strategies based on predictive modeling and machine learning techniques. Utilize the insights gained from the EDA to build predictive models that can accurately predict diabetes diagnosis and identify high-risk individuals for early intervention and preventive measures.

- **Lifestyle Interventions:** Promote lifestyle interventions such as a healthy diet, regular exercise, weight management, and stress reduction to mitigate the risk of diabetes among at-risk populations. Provide educational resources and support programs to empower individuals to adopt healthier lifestyles.

- **Regular Monitoring:** Emphasize the importance of regular health screenings and monitoring of key health indicators such as glucose levels, BMI, and blood pressure. Encourage proactive healthcare practices to detect and manage diabetes early, leading to better health outcomes and reduced complications.