



CHRIST
(DEEMED TO BE UNIVERSITY)
B E N G A L U R U • I N D I A

DIABETES PREDICTION DATABASE

Project Report Submitted in partial fulfilment of the requirement for the award of degree of

Master of Science (Data Science)



Submitted By

NAME: ANGELINE A

REG NO: 2348409

Submitted To

Dr. Gobinath R

DEPARTMENT OF COMPUTER SCIENCE

**CHRIST (DEEMED TO BE UNIVERSITY), Bangalore Yeshwanthpur
Campus, Karnataka-560073**

INTRODUCTION:

In this project, we are going to create a database with data related to Diabetes which apparently helps us to make some predictions on the Dataset and we can check whether the person is having diabetes or not. And we are importing the data into MongoDB to analyze and work on the dataset.

PROBLEM STATEMENT:

Checking whether the Patient has diabetes or not based on their Lab Results, Vitals and Medical History.

METHODS:

The methods that were employed in this project are,

1. We used MySQL to create the Database
2. The created Database is exported in csv format
3. The csv file is imported into MongoDB
4. The analyzation of the data is being done with the help of Mongosh

PROCESS/DISCUSSION:

➤ SQL CODE FOR TABLE CREATION

In the SQL workbench, we created a new database named MongoDB and created table named DiabetesPrediction with columns PredictionID, PatientID, Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome, PredictionDate.

```
1 • create database mongodb;
2 • use mongodb;
3 • CREATE TABLE Patients (
4     PatientID INT AUTO_INCREMENT PRIMARY KEY,
5     FirstName VARCHAR(50),
6     LastName VARCHAR(50),
7     Gender ENUM('Male', 'Female'),
8     Birthdate DATE,
9     ContactNumber VARCHAR(15),
10    Email VARCHAR(100),
11    Address VARCHAR(255)
12 );
13 • CREATE TABLE MedicalHistory (
14     RecordID INT AUTO_INCREMENT PRIMARY KEY,
15     PatientID INT,
16     DateOfVisit DATE,
17     Diagnosis VARCHAR(255),
18     Medications VARCHAR(255),
19     CONSTRAINT fk_patient FOREIGN KEY (PatientID) REFERENCES Patients(PatientID) ON DELETE CASCADE
20 );
21 • CREATE TABLE Vitals (
22     VitalsID INT AUTO_INCREMENT PRIMARY KEY,
23     PatientID INT,
24     CheckupDate DATE,
25     BloodPressure INT,
26     HeartRate INT,
27     Temperature DECIMAL(4,1),
28     CONSTRAINT fk_patient_vitals FOREIGN KEY (PatientID) REFERENCES Patients(PatientID) ON DELETE CASCADE
29 );
30 • CREATE TABLE LabResults (
31     LabResultID INT AUTO_INCREMENT PRIMARY KEY,
```

```

32     PatientID INT,
33     TestName VARCHAR(100),
34     Result DECIMAL(8,2),
35     Units VARCHAR(20),
36     TestDate DATE,
37     CONSTRAINT fk_patient_lab FOREIGN KEY (PatientID) REFERENCES Patients(PatientID) ON DELETE CASCADE
38 );
39 • CREATE TABLE DiabetesPrediction (
40     PredictionID INT AUTO_INCREMENT PRIMARY KEY,
41     PatientID INT,
42     Pregnancies INT,
43     Glucose INT,
44     BloodPressure INT,
45     SkinThickness INT,
46     Insulin INT,
47     BMI DOUBLE,
48     DiabetesPedigreeFunction DOUBLE,
49     Age INT,
50     Outcome INT,
51     PredictionDate DATE,
52     CONSTRAINT fk_patient_prediction FOREIGN KEY (PatientID) REFERENCES Patients(PatientID) ON DELETE CASCADE
53 );

```

We are inserting 15 values into the table.

```
54  -- Insert 15 entries into the Patients table
55  • INSERT INTO Patients (FirstName, LastName, Gender, Birthdate, ContactNumber, Email, Address)
56  VALUES
57      ('John', 'Mathew', 'Male', '1990-01-15', '9876543210', 'john.mathew@example.com', '123 Main St, City'),
58      ('Anna', 'George', 'Female', '1985-03-20', '8765432109', 'anna.george@example.com', '456 Oak St, Town'),
59      ('Joseph', 'Thomas', 'Male', '1988-07-12', '7890123456', 'joseph.thomas@example.com', '789 Pine St, Village'),
60      ('Mary', 'Philip', 'Female', '1992-04-30', '8901234567', 'mary.philip@example.com', '101 Maple St, Town'),
61      ('Thomas', 'George', 'Male', '1995-09-18', '9012345678', 'thomas.george@example.com', '222 Cedar St, City'),
62      ('Susan', 'John', 'Female', '1987-12-05', '1234567890', 'susan.john@example.com', '333 Elm St, Village'),
63      ('Daniel', 'Joseph', 'Male', '1980-06-25', '2345678901', 'daniel.joseph@example.com', '444 Oak St, City'),
64      ('Elizabeth', 'Samuel', 'Female', '1998-11-08', '3456789012', 'elizabeth.samuel@example.com', '555 Pine St, Town'),
65      ('Philip', 'Daniel', 'Male', '1983-02-14', '4567890123', 'philip.daniel@example.com', '666 Maple St, Village'),
66      ('Catherine', 'Jacob', 'Female', '1993-08-22', '5678901234', 'catherine.jacob@example.com', '777 Cedar St, City'),
67      ('Michael', 'Mathew', 'Male', '1982-05-10', '6789012345', 'michael.mathew@example.com', '888 Elm St, Town'),
68      ('Sara', 'George', 'Female', '1989-10-02', '7890123456', 'sara.george@example.com', '999 Pine St, Village'),
69      ('David', 'Joseph', 'Male', '1996-03-17', '8901234567', 'david.joseph@example.com', '111 Oak St, City'),
70      ('Rachel', 'Philip', 'Female', '1986-09-28', '9012345678', 'rachel.philip@example.com', '222 Maple St, Town'),
71      ('George', 'Thomas', 'Male', '1991-04-14', '1234567890', 'george.thomas@example.com', '333 Cedar St, Village');
72  • Select * from patients;

73  -- Insert 15 entries into the MedicalHistory table
74  • INSERT INTO MedicalHistory (PatientID, DateOfVisit, Diagnosis, Medications)
75  VALUES
76      (1, '2023-01-05', 'Hypertension', 'Medication A'),
77      (2, '2022-11-10', 'Diabetes', 'Insulin'),
78      (3, '2021-08-15', 'Migraine', 'Painkillers'),
79      (4, '2020-12-02', 'High Cholesterol', 'Statins'),
80      (5, '2022-02-28', 'Asthma', 'Inhaler'),
81      (6, '2023-03-20', 'Allergies', 'Antihistamines'),
82      (7, '2022-07-08', 'Arthritis', 'Anti-inflammatory'),
83      (8, '2021-05-12', 'Depression', 'Antidepressants'),
84      (9, '2020-11-01', 'Gastritis', 'Antacids'),
85      (10, '2022-09-25', 'Thyroid Disorder', 'Thyroid Medication'),
86      (11, '2020-04-15', 'Chronic Pain', 'Pain Management'),
87      (12, '2023-10-10', 'Anxiety', 'Anxiolytics'),
88      (13, '2021-06-30', 'Insomnia', 'Sleeping Pills'),
89      (14, '2020-03-17', 'COPD', 'Bronchodilators'),
90      (15, '2022-12-05', 'Kidney Stones', 'Pain Relief');
```

```

91      -- Insert 15 entries into the LabResults table
92 •   INSERT INTO LabResults (PatientID, TestName, Result, Units, TestDate)
93     VALUES
94         (1, 'Blood Sugar', 110, 'mg/dL', '2023-01-05'),
95         (2, 'Cholesterol', 180, 'mg/dL', '2022-11-10'),
96         (3, 'Hemoglobin', 14.5, 'g/dL', '2021-08-15'),
97         (4, 'Triglycerides', 120, 'mg/dL', '2020-12-02'),
98         (5, 'Lung Function', 85, '%', '2022-02-28'),
99         (6, 'IgE Level', 120, 'IU/mL', '2023-03-20'),
100        (7, 'Erythrocyte Sedimentation Rate', 15, 'mm/hr', '2022-07-08'),
101        (8, 'Serotonin Level', 35, 'ng/mL', '2021-05-12'),
102        (9, 'Stomach pH', 4.0, '', '2020-11-01'),
103        (10, 'Thyroid Stimulating Hormone', 2.5, 'uIU/mL', '2022-09-25'),
104        (11, 'Pain Biomarkers', 25, 'ng/mL', '2020-04-15'),
105        (12, 'Cortisol Level', 18, 'ug/dL', '2023-10-10'),
106        (13, 'Melatonin Level', 5, 'ng/mL', '2021-06-30'),
107        (14, 'Lung Capacity', 3.2, 'L', '2020-03-17'),
108        (15, 'Calcium Oxalate', 8.0, 'mg/dL', '2022-12-05');

91      -- Insert 15 entries into the Vitals table
92 •   INSERT INTO Vitals (PatientID, CheckupDate, BloodPressure, HeartRate, Temperature)
93     VALUES
94         (1, '2023-01-05', '120/80', 75, 98.6),
95         (2, '2022-11-10', '130/85', 80, 99.2),
96         (3, '2021-08-15', '110/70', 72, 98.0),
97         (4, '2020-12-02', '125/85', 78, 98.4),
98         (5, '2022-02-28', '118/75', 70, 98.8),
99         (6, '2023-03-20', '122/78', 68, 98.7),
100        (7, '2022-07-08', '128/82', 75, 99.0),
101        (8, '2021-05-12', '115/70', 80, 98.2),
102        (9, '2020-11-01', '135/90', 85, 99.5),
103        (10, '2022-09-25', '120/80', 72, 98.9),
104        (11, '2020-04-15', '130/85', 76, 98.3),
105        (12, '2023-10-10', '112/72', 70, 98.1),
106        (13, '2021-06-30', '126/84', 78, 99.1),
107        (14, '2020-03-17', '115/75', 74, 98.5),
108        (15, '2022-12-05', '128/80', 77, 98.7);

```



```

109      -- Insert 15 entries into the LabResults table
110 •   INSERT INTO LabResults (PatientID, TestName, Result, Units, TestDate)
111      VALUES
112          (1, 'Blood Sugar', 110, 'mg/dL', '2023-01-05'),
113          (2, 'Cholesterol', 180, 'mg/dL', '2022-11-10'),
114          (3, 'Hemoglobin', 14.5, 'g/dL', '2021-08-15'),
115          (4, 'Triglycerides', 120, 'mg/dL', '2020-12-02'),
116          (5, 'Lung Function', 85, '%', '2022-02-28'),
117          (6, 'IgE Level', 120, 'IU/mL', '2023-03-20'),
118          (7, 'Erythrocyte Sedimentation Rate', 15, 'mm/hr', '2022-07-08'),
119          (8, 'Serotonin Level', 35, 'ng/mL', '2021-05-12'),
120          (9, 'Stomach pH', 4.0, '', '2020-11-01'),
121          (10, 'Thyroid Stimulating Hormone', 2.5, 'uIU/mL', '2022-09-25'),
122          (11, 'Pain Biomarkers', 25, 'ng/mL', '2020-04-15'),
123          (12, 'Cortisol Level', 18, 'ug/dL', '2023-10-10'),
124          (13, 'Melatonin Level', 5, 'ng/mL', '2021-06-30'),
125          (14, 'Lung Capacity', 3.2, 'L', '2020-03-17'),
126          (15, 'Calcium Oxalate', 8.0, 'mg/dL', '2022-12-05');
127
128 •   -- Insert 15 entries into the DiabetesPrediction table
129      INSERT INTO DiabetesPrediction (PatientID, Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, Di
130      VALUES
131          (1, 2, 120, 70, 23, 80, 25.5, 0.45, 32, 0, '2023-01-05'),
132          (2, 3, 150, 80, 25, 100, 28.0, 0.60, 40, 1, '2022-11-10'),
133          (3, 1, 110, 65, 20, 75, 24.0, 0.35, 28, 0, '2021-08-15'),
134          (4, 4, 140, 75, 22, 90, 26.2, 0.50, 35, 1, '2020-12-02'),
135          (5, 2, 130, 72, 21, 85, 25.0, 0.40, 30, 0, '2022-02-28'),
136          (6, 3, 125, 68, 20, 78, 23.5, 0.38, 27, 0, '2023-03-20'),
137          (7, 5, 135, 78, 24, 95, 27.5, 0.55, 38, 1, '2022-07-08'),
138          (8, 2, 115, 70, 22, 80, 24.8, 0.42, 33, 0, '2021-05-12'),
139          (9, 1, 105, 65, 19, 70, 22.0, 0.30, 25, 0, '2020-11-01'),
140          (10, 3, 140, 75, 23, 92, 26.8, 0.48, 36, 1, '2022-09-25'),
141          (11, 4, 125, 72, 20, 88, 25.5, 0.44, 31, 0, '2020-04-15'),
142          (12, 2, 130, 68, 21, 85, 24.9, 0.41, 29, 1, '2023-10-10'),
143          (13, 1, 120, 74, 19, 78, 23.2, 0.37, 26, 0, '2021-06-30'),
144          (14, 5, 138, 80, 25, 98, 28.3, 0.58, 39, 1, '2020-03-17'),
145          (15, 3, 142, 76, 22, 94, 27.0, 0.52, 34, 0, '2022-12-05');

```

➤ EXPORTING THE TABLES FROM SQL (in csv format)

Table Data Export

Select data for export

Select source table for export:

Select columns you'd like to export

Export	Column name
<input checked="" type="checkbox"/>	PatientID
<input checked="" type="checkbox"/>	FirstName
<input checked="" type="checkbox"/>	LastName
<input checked="" type="checkbox"/>	Gender
<input checked="" type="checkbox"/>	Birthdate
<input checked="" type="checkbox"/>	ContactNumber
<input checked="" type="checkbox"/>	Email
<input checked="" type="checkbox"/>	Address

☒ Select / Deselect all entries

Row Offset: Count:

Advanced >>

< Back Next > Cancel

Save As

Organize New folder

File name:

Save as type:

Save Cancel

Name	Date modified	Type	Size
DiabetesPrediction	1/21/2024 11:45 AM	Microsoft Excel C...	1
LabResults	1/21/2024 11:46 AM	Microsoft Excel C...	1
MedicalHistory	1/21/2024 11:46 AM	Microsoft Excel C...	1

Table Data Export

Select output file location

Table Data Export allows you to easily export data into CSV, JSON datafiles.

File Path: D:\Databases technologies_MDS271\MongoDB CAT2\Patients.csv

Browse...

☒ csv

☐ json

Options:

Field Separator

;

Line Separator

LF

Enclose Strings in

"

null and NULL word as SQL keyword

YES

☒ Export to local machine

If checked, rows will be exported on the location that started Workbench.
If not checked, rows will be exported on the server.
If server and computer that started Workbench are different machines, import of that file can be done manual way only.

< Back

Next >

Cancel

Table Data Export

Export Data

The following tasks will now be performed. Please monitor the execution.

☐ Prepare Export

☐ Export data to file

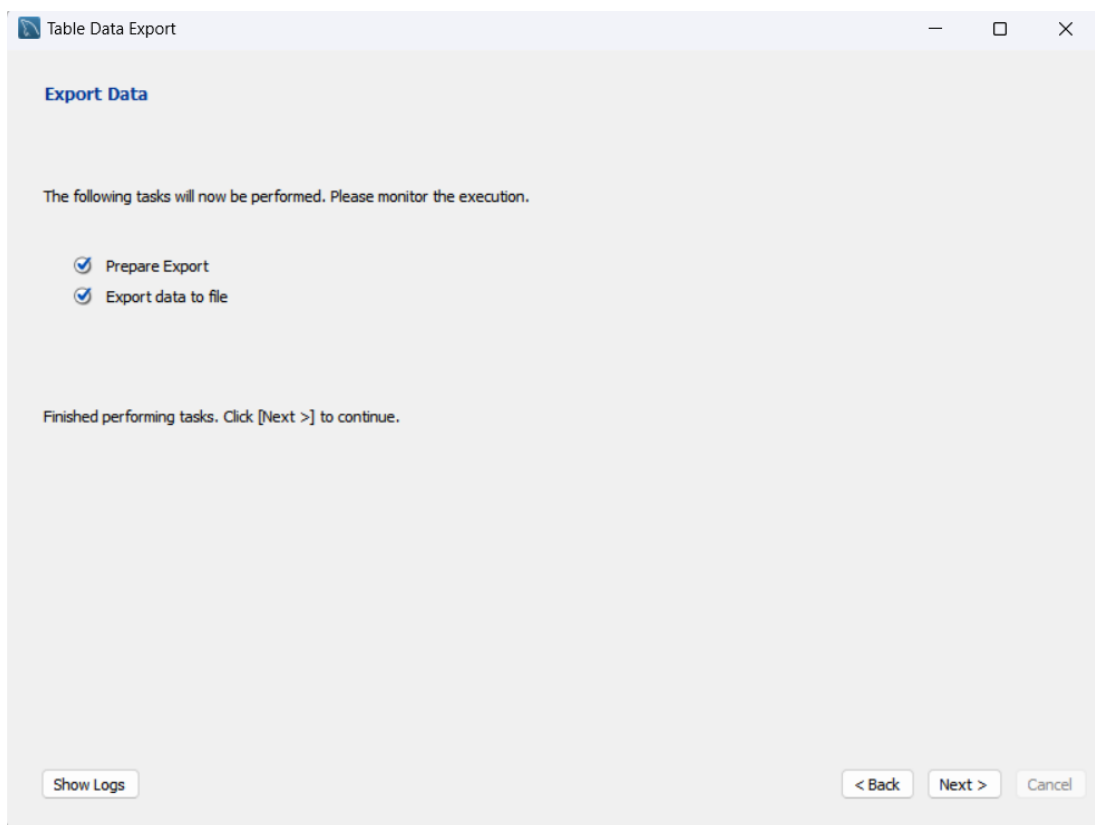
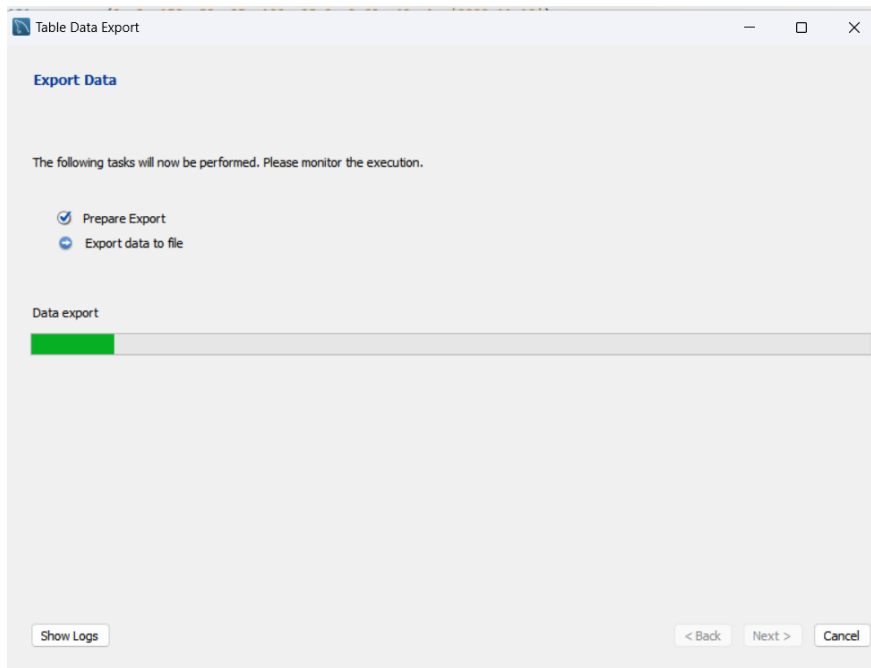
Click [Next >] to execute.

Show Logs

< Back

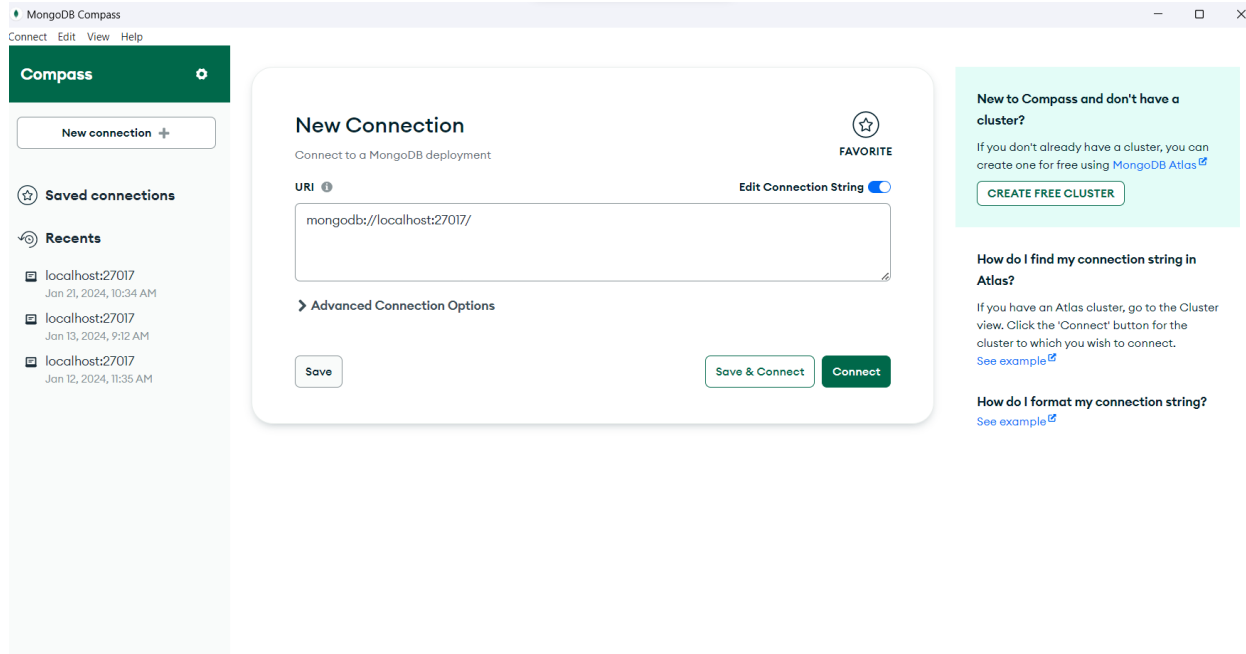
Next >

Cancel



➤ CREATING A DATABASE IN MONGODB AND IMPORTING THE CREATED DATABASE

The connection is made, the database is created and the dataset that we created in being imported into the MongoDB.



Create Database

Database Name

CAT2

Collection Name

DiabetesPredictionData

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database

ADD DATA EXPORT DATA					
1 - 15 of 15					
★ DiabetesPredictionData					
	_id ObjectId	PredictionID Int32	PatientID Int32	Pregnancies Int32	Glucose Int32
1	ObjectId('65acbf27aba4000...	1	1	2	120
2	ObjectId('65acbf27aba4000...	2	2	3	150
3	ObjectId('65acbf27aba4000...	3	3	1	110
4	ObjectId('65acbf27aba4000...	4	4	4	140
5	ObjectId('65acbf27aba4000...	5	5	2	130
6	ObjectId('65acbf27aba4000...	6	6	3	125
7	ObjectId('65acbf27aba4000...	7	7	5	135
8	ObjectId('65acbf27aba4000...	8	8	2	115
9	ObjectId('65acbf27aba4000...	9	9	1	105
10	ObjectId('65acbf27aba4000...	10	10	3	140
11	ObjectId('65acbf27aba4000...	11	11	4	125
12	ObjectId('65acbf27aba4000...	12	12	2	130
13	ObjectId('65acbf27aba4000...	13	13	1	120

➤ ANALYZATION OF THE DATASET AND FINDING INSIGHTS (in mongosh)

I. Finding the Average BMI

```
CAT2> var BMI = db.CAT2.DiabetesPrediction.aggregate([{$group: {_id: "$Outcome", count: {$sum: 1}, avgBMI: {$avg: "$BMI"}}}])
```

```
CAT2> BMI.forEach(function(doc){print("Outcome: " + doc._id); print("Count: " + doc.count); print("Average BMI: " + doc.avgBMI); print("====="); });
```

II. Deleting the Values

```
CAT2> db.CAT2.DiabetesPrediction.deleteMany({"Age": {$lt: 30}})
{ acknowledged: true, deletedCount: 0 }
```

III. Updating the values

```
CAT2> db.CAT2.DiabetesPrediction.updateOne({"PatientID": 1}, {$set: {"Glucose": 130}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

IV. Finding/Retrieving the Data (Patients having Diabetes)

```
CAT2> db.CAT2.DiabetesPrediction.find()
CAT2> db.CAT2.DiabetesPrediction.find({"Outcome": 1})
```

V. Inserting One Value

```
CAT2> db.CAT2.DiabetesPrediction.insertOne({PatientID:1,Pregnancies:5,Glucose:120,BloodPressure:80,SkinThickness:25,Insulin:40,BMI:30.5,DiabetesPedigreeFunction:0.6,Age:35,Outcome:0,PredictionDate:new Date()})
{
  acknowledged: true,
  insertedId: ObjectId('65accf6bb9474e1d3f747cd2')
}
```

VI. Updating Many values

```
CAT2> db.CAT2.DiabetesPrediction.updateMany({"Outcome":1},{ $set: {"BMI":32.5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

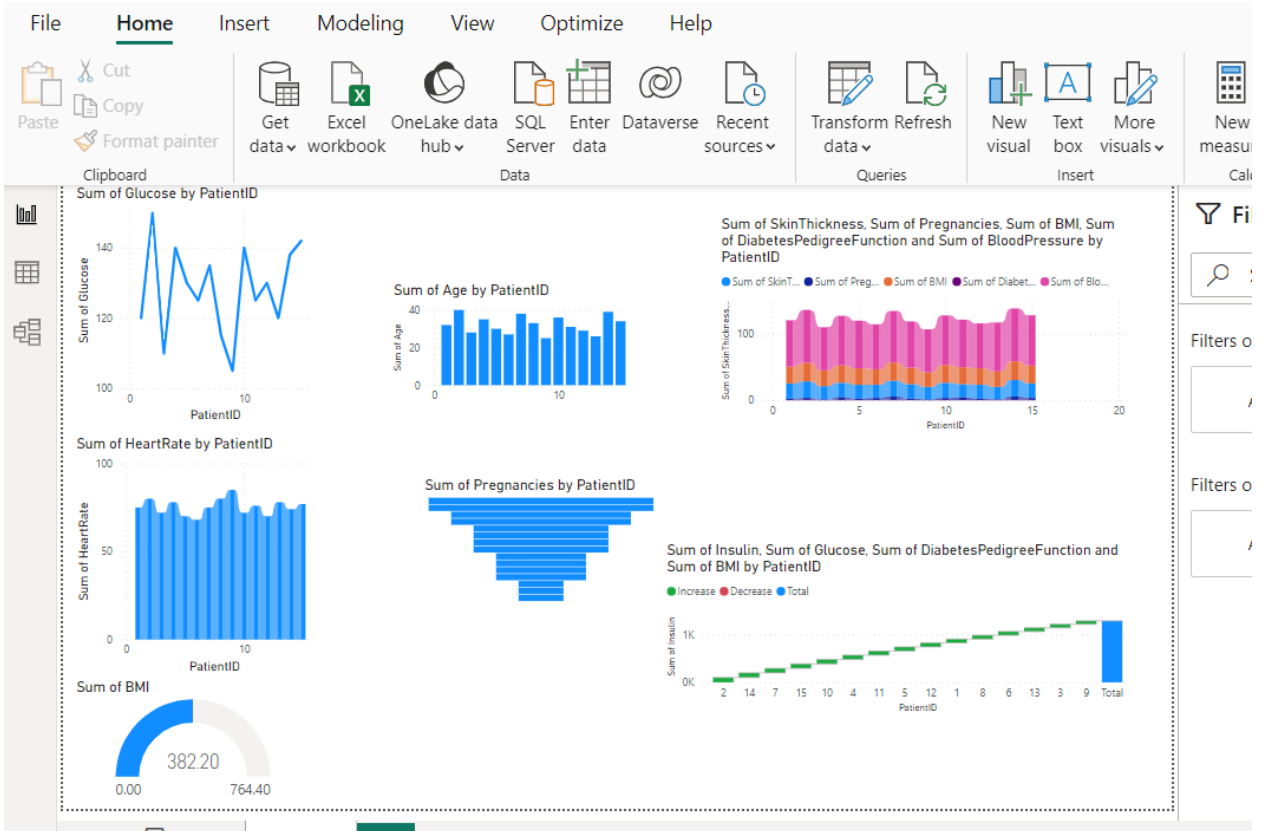
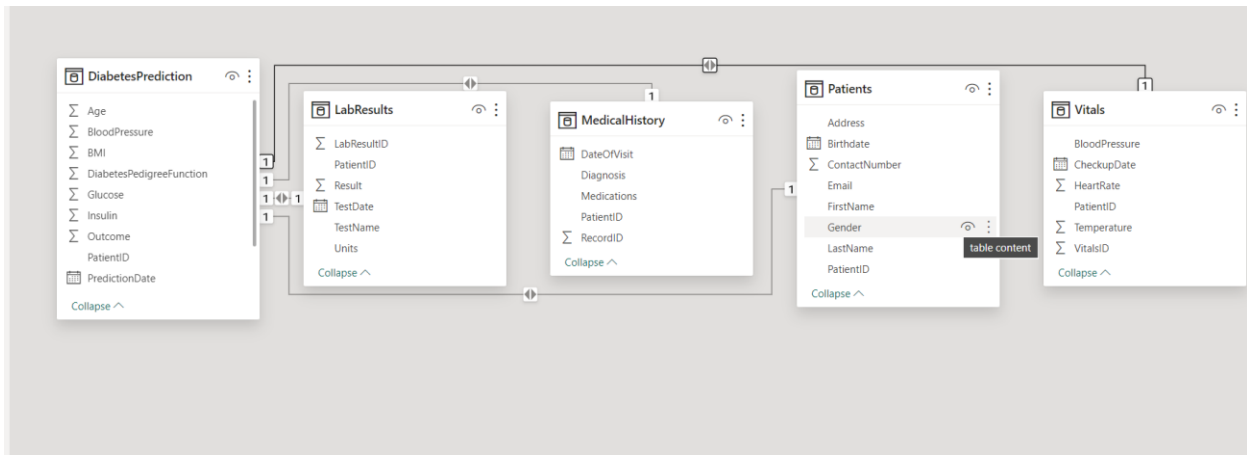
VII. Retrieving Specific fields

```
CAT2> db.CAT2.DiabetesPrediction.find({}, {"PatientID":1,"Glucose":1,"Outcome":1})
[
  {
    _id: ObjectId('65accf6bb9474e1d3f747cd2'),
    PatientID: 1,
    Glucose: 120,
    Outcome: 0
  }
]
```

VIII. Limit and Sort Results

```
CAT2> db.CAT2.DiabetesPrediction.find().limit(5).sort({"Age":-1})
[
  {
    _id: ObjectId('65accf6bb9474e1d3f747cd2'),
    PatientID: 1,
    Pregnancies: 5,
    Glucose: 120,
    BloodPressure: 80,
    SkinThickness: 25,
    Insulin: 40,
    BMI: 30.5,
    DiabetesPedigreeFunction: 0.6,
    Age: 35,
    Outcome: 0,
    PredictionDate: ISODate('2024-01-21T08:01:47.112Z')
  }
]
```

POWERBI CONNECTION



CONCLUSION:

We have done all the procedures and analyzed the data with the help of MongoDB. We performed CRUD (Create, Read, Update, Delete) in mangosh which helped us to get meaningful insights from the created database and we refreshed the data in MongoDB which gave us the new updated data after all operations (CRUD). The tables are imported into PowerBI and I visualized the changes.