IE6700 Data Management for Analytics
STUDENT MENTAL HEALTH

Project Report

Group 20

Student 1: Vinyas Naidu Karri

Student 2: Sri Sai Tarun Vemu

857-506-4664 (Tel of Vinyas)

857-6935779 (Tel of Tarun)

karri.vi@northeastern.edu

vemu.s@northeastern.edu

Percentage of Effort Contributed by Student 1: 50%

Percentage of Effort Contributed by Student 2: 50%

Signature of Student 1: Vinyas Naidu Karri

Signature of Student 2: Tarun Vemu

Submission Date: 21-April-2024

Title: Student Mental Health

Vinyas Naidu & Tarun Vemu

I. Introduction:

Project Background :

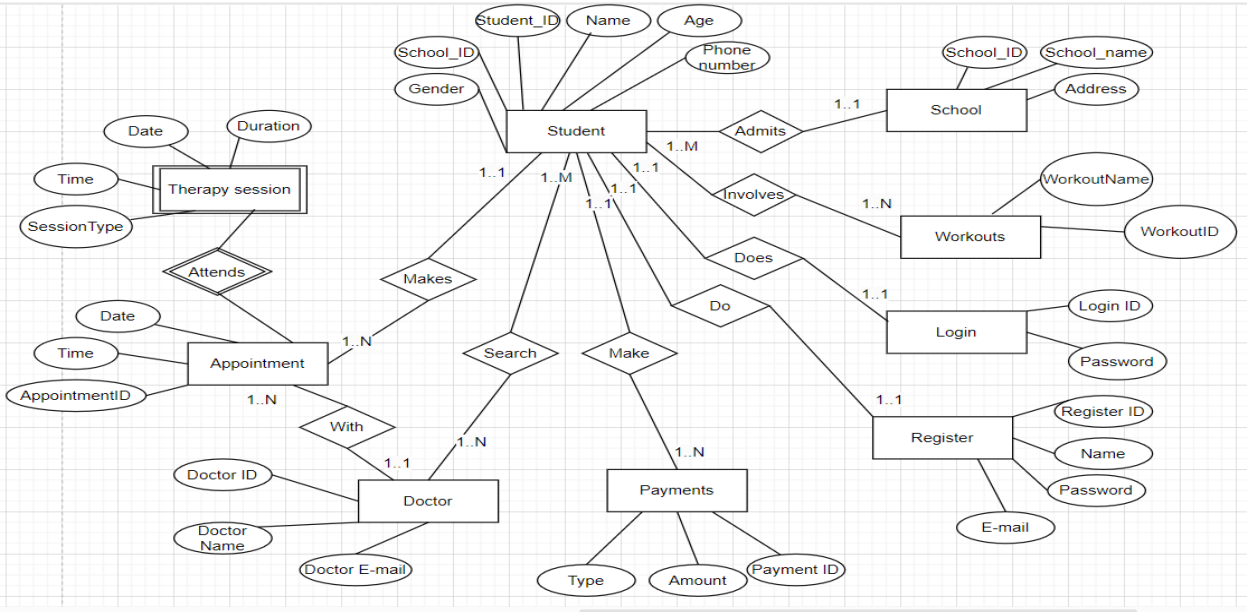
In many regions of the world, there is a serious and growing worry about students' mental health. Many of students are impacted by problems like depression, anxiety, and other mental health conditions, and society places a high importance on addressing these issues. The decision to engage on mental health-related projects is motivated by a desire to address a crucial social issue, lessen suffering, and enhance the wellbeing of people and communities.

Project Goals :

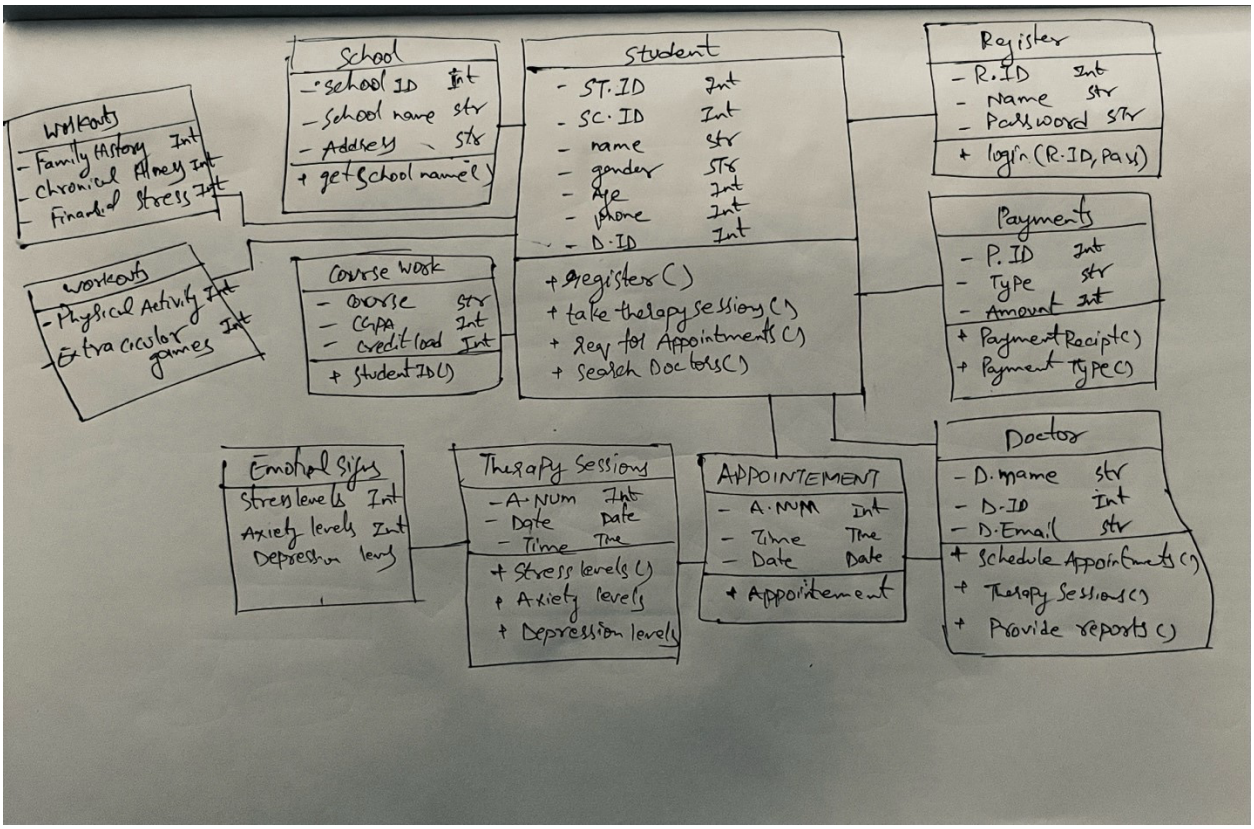
The main objective is to enhance/provide insights on mental health of people and communities, which can lead to stigma reduction, which involves making society more understanding and supportive so that people feel comfortable seeking treatment when necessary. Further, avoiding mental health problems before they escalate and providing early care to people who are at risk are crucial objectives. Moreover, ensuring the sustainability of mental health initiatives is crucial. Coming to a business perspective, there is a web application where all the student information is stored. We will be having school id, student id, name, age, gender, the course which they are enrolled, cgpa, credit load, their anxiety stress depression levels, where do they stay (on campus or off campus), counseling sessions, substance use, sleep and diet cycle, relationship status, family history, physical activity and extra-curricular activity, financial stress, the doctors information that they are consulting, information about appointments, therapy sessions, payment information etc.. The web application enables stakeholders to gain valuable insights into the mental health of students. Through data analysis and visualization, trends and patterns related to mental health can be identified, allowing for targeted interventions and support strategies. Moreover, the application facilitates collaboration among mental health professionals, educators, and support staff to deliver personalized care and promote overall well-being within the student community

I. Conceptual Data Modelling

Extended Entity Relation Model



Unified Modelling Language



II. Mapping Conceptual Model to Relational Model

STUDENT: StudentID (PK), SchoolID_FK (FK to SCHOOL), UserID_FK (FK to LOGIN), Name, Age, PhoneNumber, Gender, Course, CreditLoad, GPA.

SCHOOL: SchoolID (PK), Name, Address.

WORKOUT: WorkoutID (PK), WorkoutName.

STUDENT_WORKOUTS: StudentID_FK (FK to STUDENT), WorkoutID_FK (FK to WORKOUT).

LOGIN: UserID (PK), PasswordHash.

REGISTER_C: RegisterID (PK), StudentID_FK (FK to STUDENT), Name, Email.

PAYMENTS: PaymentID (PK), StudentID_FK (FK to STUDENT), Amount, Type.

DOCTORS: DoctorID (PK), DoctorName, DoctorEmail.

STUDENT_DOCTOR: StudentID_FK (FK to STUDENT), DoctorID_FK (FK to DOCTORS).

APPOINTMENTS: AppointmentID (PK), StudentID_FK (FK to STUDENT), DoctorID_FK (FK to DOCTORS) , Date, Time.

THERAPY_SESSIONS: AppointmentID_FK (FK to APPOINTMENTS), Date, Duration, Time, SessionType.

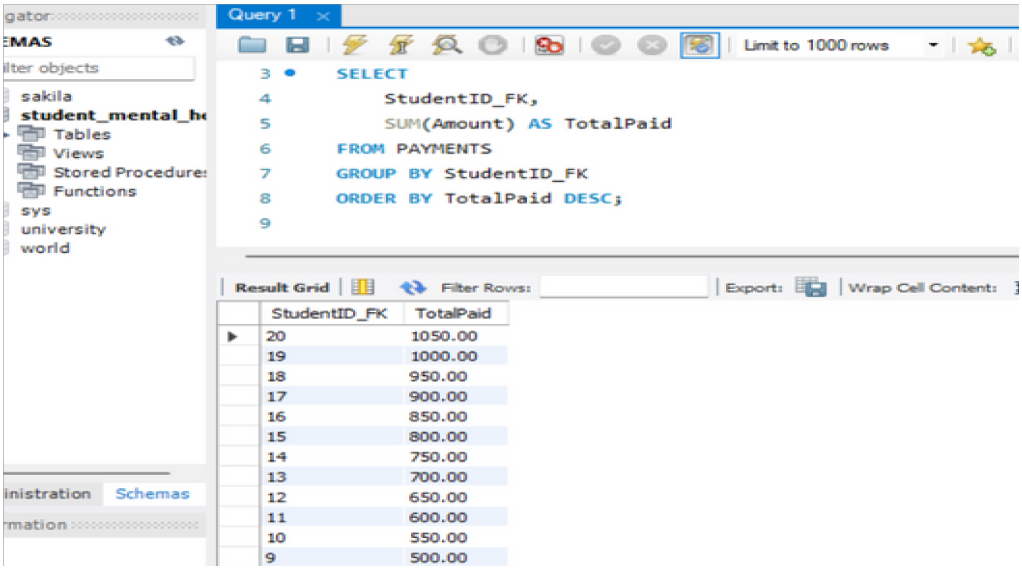
III .Implementation of Relational Model: MySQL

Schema Name used: Student_Mental_Health

Total No. Of tables after Normalization: 11

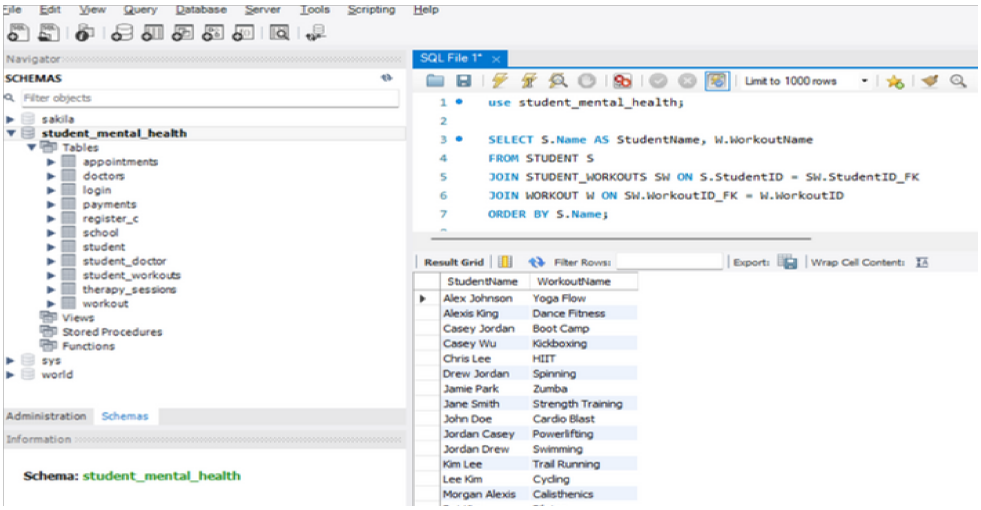
Query 1 - how much total amount each student has paid, and order the students by the total amount paid in descending order

```
SELECT
    StudentID_FK,
    SUM(Amount) AS TotalPaid
FROM
    PAYMENTS
GROUP BY
    StudentID_FK
ORDER BY
    TotalPaid DESC;
```



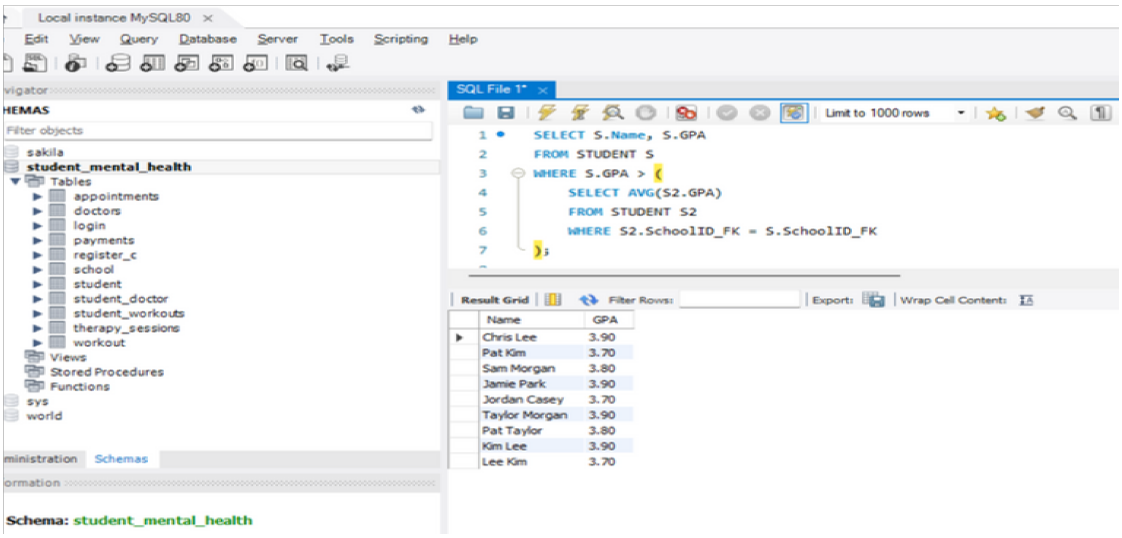
Query 2 - List all students with their workouts

```
SELECT S.Name AS StudentName, W.WorkoutName FROM STUDENT S JOIN STUDENT_WORKOUTS SW ON S.StudentID = SW.StudentID_FK JOIN WORKOUT W ON SW.WorkoutID_FK = W.WorkoutID ORDER BY S.Name;
```



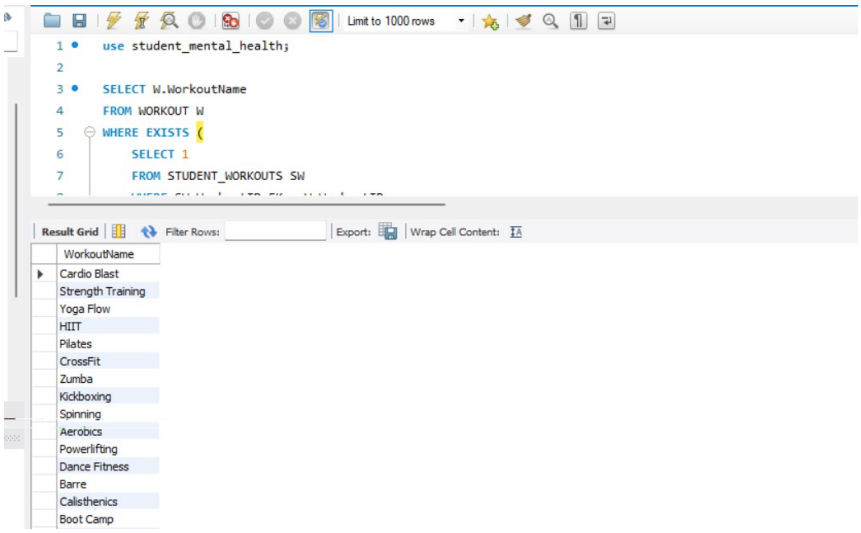
Query 3 List students and their GPAs if they are above the average GPA of their school

```
SELECT S.Name, S.GPA
FROM STUDENT S
WHERE S.GPA > (
SELECT AVG(S2.GPA)
FROM STUDENT S2
WHERE S2.SchoolID_FK = S.SchoolID_FK
);
```



Query 4 Find workouts that have been attended by at least one student

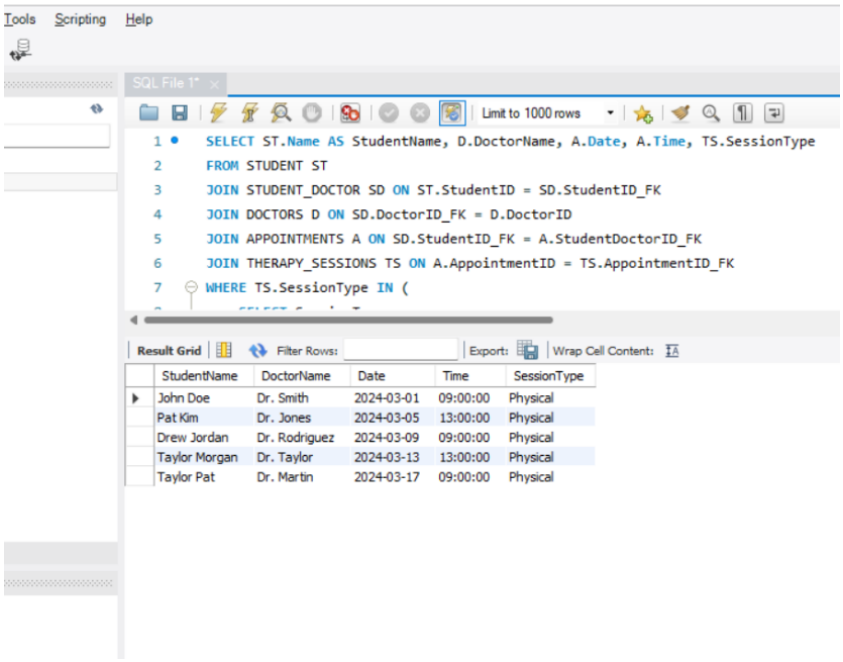
```
SELECT W.WorkoutName
FROM WORKOUT W
WHERE EXISTS (
   SELECT 1
   FROM STUDENT_WORKOUTS SW
   WHERE SW.WorkoutID_FK = W.WorkoutID
);
```



Query 5 List students, their doctors, and appointment details for a specific type of therapy session

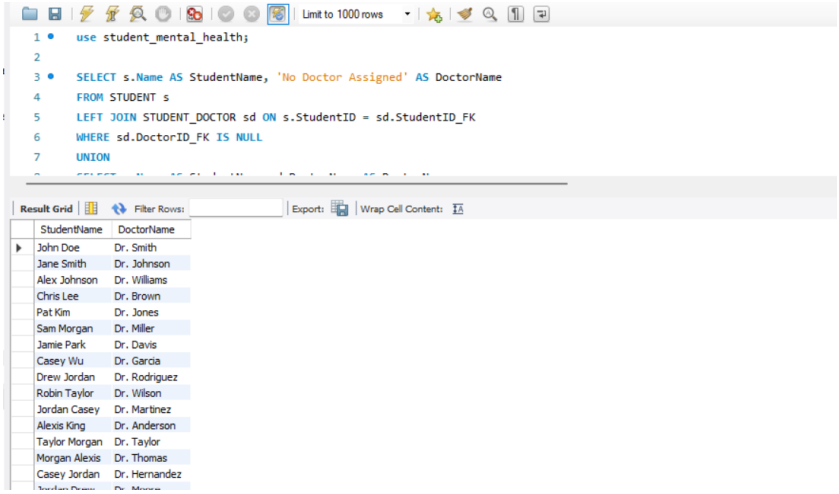
```
SELECT ST.Name AS StudentName, D.DoctorName, A.Date, A.Time,
TS.SessionType
FROM STUDENT ST
JOIN STUDENT_DOCTOR SD ON ST.StudentID = SD.StudentID_FK
JOIN DOCTORS D ON SD.DoctorID_FK = D.DoctorID
JOIN APPOINTMENTS A ON SD.StudentID_FK =
A.StudentDoctorID_FK
JOIN THERAPY_SESSIONS TS ON A.AppointmentID =
```

```
TS.AppointmentID_FK
WHERE TS.SessionType IN (
SELECT SessionType
FROM THERAPY_SESSIONS
WHERE SessionType = 'Physical' )
ORDER BY A.Date, A.Time;
```



Query 6 List All Students and Their Doctors

```
SELECT s.Name AS StudentName, 'No Doctor Assigned' AS DoctorName
FROM STUDENT s
LEFT JOIN STUDENT_DOCTOR sd ON s.StudentID = sd.StudentID_FK
WHERE sd.DoctorID_FK IS NULL
UNION
SELECT s.Name AS StudentName, d.DoctorName AS DoctorName
FROM STUDENT s
JOIN STUDENT_DOCTOR sd ON s.StudentID = sd.StudentID_FK
JOIN DOCTORS d ON sd.DoctorID_FK = d.DoctorID;
```



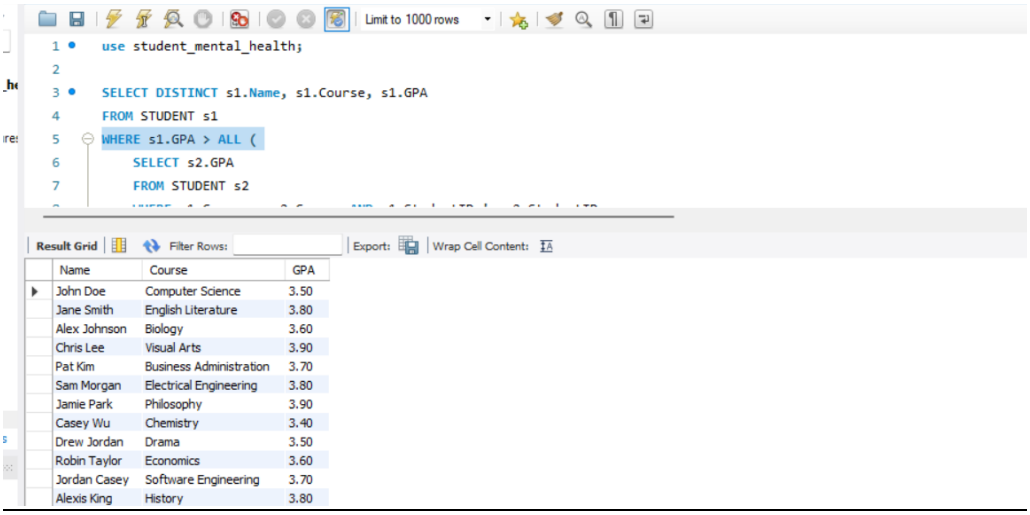
```
use student_mental_health;

SELECT s.Name AS StudentName, 'No Doctor Assigned' AS DoctorName
FROM STUDENT s
LEFT JOIN STUDENT_DOCTOR sd ON s.StudentID = sd.StudentID_FK
WHERE sd.DoctorID_FK IS NULL
UNION
```

StudentName	DoctorName
John Doe	Dr. Smith
Jane Smith	Dr. Johnson
Alex Johnson	Dr. Williams
Chris Lee	Dr. Brown
Pat Kim	Dr. Jones
Sam Morgan	Dr. Miller
Jamie Park	Dr. Davis
Casey Wu	Dr. Garcia
Drew Jordan	Dr. Rodriguez
Robin Taylor	Dr. Wilson
Jordan Casey	Dr. Martinez
Alexis King	Dr. Anderson
Taylor Morgan	Dr. Taylor
Morgan Alexis	Dr. Thomas
Casey Jordan	Dr. Hernandez

Query 7 find students who have a GPA that is higher than all other students in the same course

```
SELECT DISTINCT s1.Name, s1.Course, s1.GPA
FROM STUDENT s1
WHERE s1.GPA > ALL (
    SELECT s2.GPA
    FROM STUDENT s2
    WHERE s1.Course = s2.Course AND s1.StudentID != s2.StudentID
);
```



```
use student_mental_health;

SELECT DISTINCT s1.Name, s1.Course, s1.GPA
FROM STUDENT s1
WHERE s1.GPA > ALL (
    SELECT s2.GPA
    FROM STUDENT s2
```

Name	Course	GPA
John Doe	Computer Science	3.50
Jane Smith	English Literature	3.80
Alex Johnson	Biology	3.60
Chris Lee	Visual Arts	3.90
Pat Kim	Business Administration	3.70
Sam Morgan	Electrical Engineering	3.80
Jamie Park	Philosophy	3.90
Casey Wu	Chemistry	3.40
Drew Jordan	Drama	3.50
Robin Taylor	Economics	3.60
Jordan Casey	Software Engineering	3.70
Alexis King	History	3.80

Implementation of Python and Sql

Connected the sql workbench with python using root user and password.

```
In [3]: import mysql.connector
        from mysql.connector import Error

In [4]: connection = mysql.connector.connect(host=' localhost',
                                             database='student_mental_health',
                                             user='root',
                                             password='new_password',
                                             auth_plugin='mysql_native_password')

In [5]: connection

Out[5]: <mysql.connector.connection_cext.CMySQLConnection at 0x20d7b450970>

In [8]: if connection.is_connected():
            db_Info = connection.get_server_info()
            print("Connected to MySQL Server version ", db_Info)
            cursor = connection.cursor()
            cursor.execute("select database();")
            record = cursor.fetchone()
            print("Your connected to database: ", record)

Connected to MySQL Server version  8.0.23
Your connected to database:  ('student_mental_health',)
```

After successfully connecting the data base from sql workbench we have retrieved student table and converted it into a data frame to perform some visualizations in Jupiter notebook to get visual insights for a better understanding of the data base

```
def select_query():
        sql_select_Query = "select * from student"
        cursor = connection.cursor()
        cursor.execute(sql_select_Query)
        records = cursor.fetchall()
        print("Students  in the garace list and details will be :\n")
        for row in records:
            print('Students_table each row =', row, "\n")


select_query()
```

```
Students  in the garace list and details will be :

Students_table each row = (1, 1, 1, 'John Doe', 20, '555-0001', 'Male', '
```

```
query = "SELECT * FROM student"
cursor.execute(query)

# Fetch all rows from the result set
data = cursor.fetchall()

# Create a DataFrame from the fetched data
columns = [desc[0] for desc in cursor.description]
df = pd.DataFrame(data, columns=columns)

print(df)
```

```
<IPython.core.display.Javascript object>
```

	StudentID	SchoolID_FK	UserID_FK	Name
0	1	1	1	John Doe
1	2	2	2	Jane Smith
2	3	3	3	Alex Johnson
3	4	4	4	Chris Lee
4	5	5	5	Pat Kim
5	6	1	6	Sam Morgan
6	7	2	7	Jamie Park
7	8	3	8	Casey Wu
8	9	4	9	Drew Jordan
9	10	5	10	Robin Taylor
10	11	1	11	Jordan Casey

Visualization 1

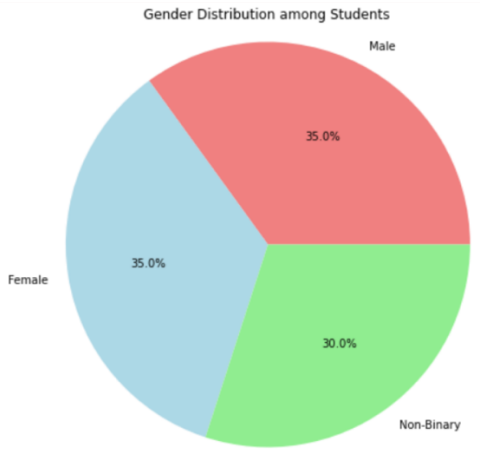
Gender distribution among students

1. **Diverse Gender Representation :** The chart depicts three categories of gender - Male, Female, and Non-Binary. This indicates that the data set includes a diverse representation of gender identities.

2. **Largest Group - Female Students :** The largest segment of the chart is occupied by Female students, who constitute 35% of the total student population. This suggests that female students make up the majority in the given dataset.

3. **Comparable Representation of Male and Non-Binary Students :** Male and Non-Binary students are almost equally represented, with Male students slightly outnumbering Non-Binary students by 5%. Male students make up 35% of the population, and Non-Binary students make up 30%.

```
# Pie Chart
gender_counts = df['Gender'].value_counts()
plt.figure(figsize=(8, 6))
plt.pie(gender_counts, labels=gender_counts.index,
        autopct='%1.1f%%',
        colors=['lightcoral', 'lightblue', 'lightgreen'])
plt.title('Gender Distribution among Students')
plt.axis('equal')
plt.tight_layout()
plt.show()
```



Visualization 2

Correlation heat map

The visualization is a correlation heatmap created using the seaborn library in Python, and it analyzes the relationships between three variables: Age, CreditLoad, and GPA.

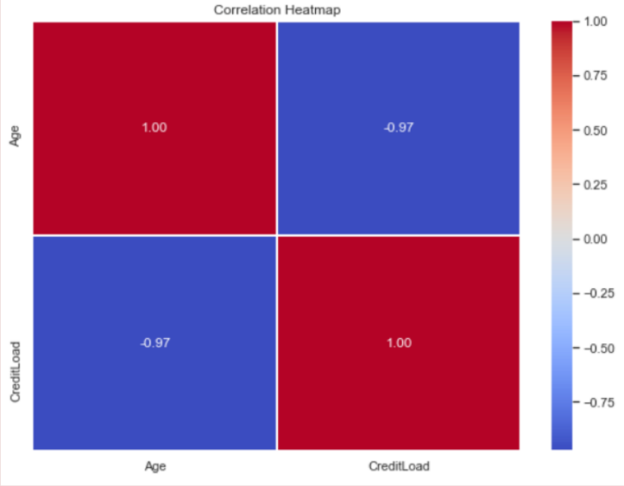
1. **Strong Negative Correlation :** There is a strong negative correlation (-0.97) between CreditLoad and GPA, which suggests that as the credit load increases, the GPA tends to decrease, or vice versa. This could indicate that a heavier credit load may have a detrimental impact on student academic performance.

2. **No Correlation with Age :** Age appears to have no correlation with GPA and CreditLoad, as indicated by the correlation coefficient of 1.00 between Age and itself and -0.97 with CreditLoad and GPA when compared with Age, which is likely a result of the diagonal of the matrix that always equals 1.00 (perfect correlation with itself).

```
import seaborn as sns

# Calculate correlation matrix
correlation_matrix = df[
    ['Age', 'CreditLoad', 'GPA']].corr()

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True,
            cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



Visualization 3

Time distribution

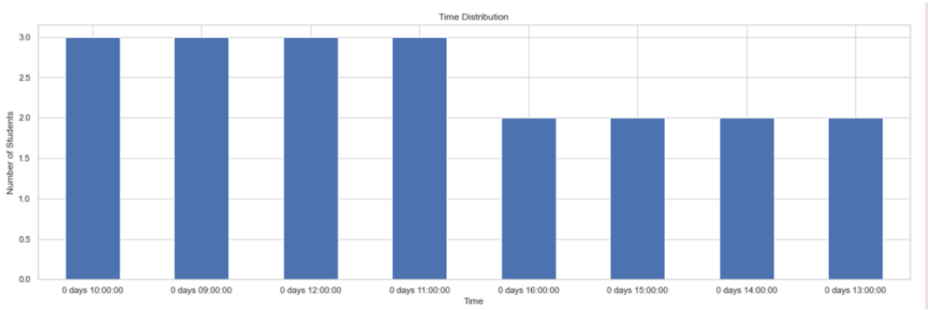
The visualization shows a bar chart titled "Time Distribution," which appears to display the distribution of a certain variable across different times of the day. The x-axis is labeled 'Time', and the y-axis is labeled 'Number of Students'.

Here are some observations and insights that can be drawn from the bar chart:

1. **Time Intervals :** The chart breaks down the distribution across various one-hour time intervals throughout the day, starting from 09:00 to 16:00.

2. **Peak Time :** The highest bar represents the time slot "0 days 10:00-10:59". This suggests that whatever activity or variable being measured reaches its peak during this hour.

3. **Uniformity in Other Time Slots :** The remaining time slots, from 09:00 to 16:00, with the exception of 10:00, show a nearly uniform distribution. This could indicate that the activity or measurement being tracked remains relatively steady throughout these times.



```
plt.figure(figsize=(20, 6))
gender_counts = df2['Time'].value_counts()
gender_counts.plot(kind='bar')
plt.title('Time Distribution')
plt.xlabel('Time')
plt.ylabel('Number of Students')
plt.xticks(rotation=0)
```

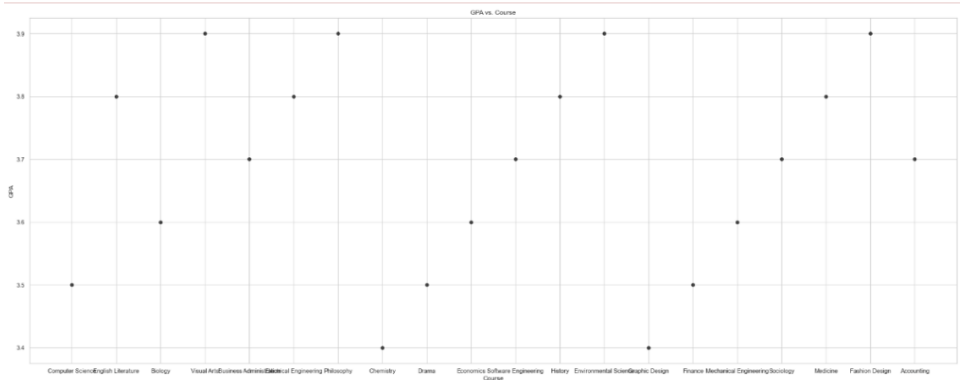
Visualization 4

GPA vs Course

The visualization is titled "GPA vs. Course" and appears to be a scatter plot, which is intended to display the relationship between students' GPAs and the courses they are enrolled in.

1. **Variation in GPA across Courses :** The plot likely aims to show how GPAs vary across different courses. The courses would be on the x-axis, and the corresponding GPA for students in each course would be on the y-axis.

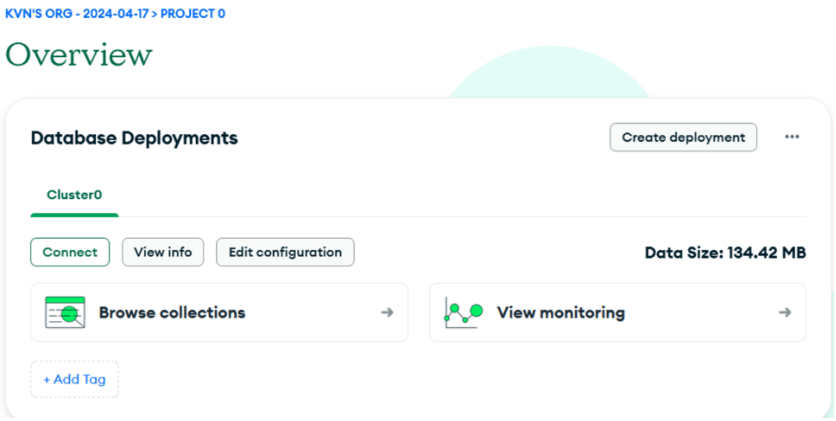
2. **Comparison Between Courses :** The chart could be used to compare the average GPA between different courses to see if some courses have higher or lower averages than others.

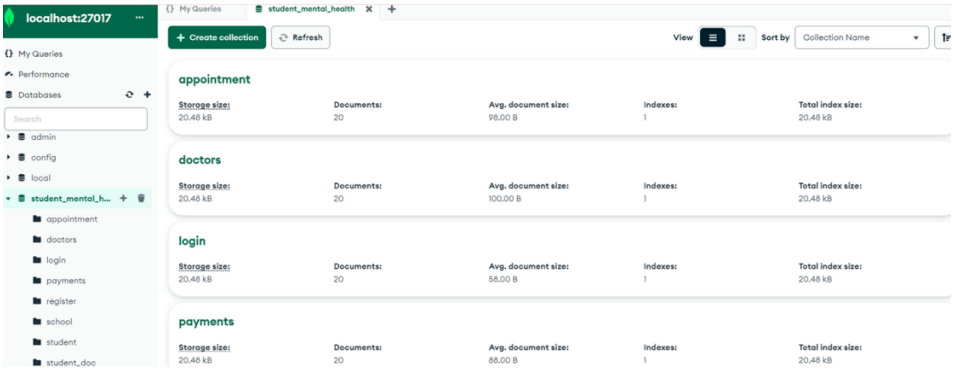



```
# Scatter Plot
plt.figure(figsize=(25, 10))
plt.scatter(df['Course'], df['GPA'],
            color='black', alpha=0.7)
plt.xlabel('Course')
plt.ylabel('GPA')
plt.title('GPA vs. Course')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Implementation of NoSql and Sql

Created a cluster on MongoDB for all the MySql tables as shown





We have performed some queries to retrieve database student_mental_health

Query 1 Finding all the male students in the database

```
> db.student.find({"Gender":"Male"});
< {
    _id: ObjectId("661ca5f3389e0c978178b79c"),
    StudentID: 1,
    SchoolID_FK: 1,
    UserID_FK: 1,
    Name: 'John Doe',
    Age: 20,
    PhoneNumber: '555-0001',
    Gender: 'Male',
    Course: 'Computer Science',
    CreditLoad: 15,
    GPA: 3.5
  }
  {
    _id: ObjectId("661ca5f3389e0c978178b79f"),
    StudentID: 4,
    SchoolID_FK: 4,
    UserID_FK: 4,
    Name: 'Chris Lee',
```

Query 2 Finding all the students whose age is above 20

```
> db.student.find({ "Age": { $gt:20 } });
< {
    _id: ObjectId("661ca5f3389e0c978178b79d"),
    StudentID: 2,
    SchoolID_FK: 2,
    UserID_FK: 2,
    Name: 'Jane Smith',
    Age: 22,
    PhoneNumber: '555-0002',
    Gender: 'Female',
    Course: 'English Literature',
    CreditLoad: 12,
    GPA: 3.8
  }
  {
    _id: ObjectId("661ca5f3389e0c978178b79f"),
    StudentID: 4,
    SchoolID_FK: 4,
    UserID_FK: 4,
    Name: 'Chris Lee',
    Age: 21,
    PhoneNumber: '555-0004',
    Gender: 'Male',
    Course: 'Visual Arts',
    CreditLoad: 14,
    GPA: 3.9
```

Query 3 Finding all the students whose GPA is greater than 3.5

```
> db.student.find({ "GPA": { $gt:3.5 } });
< {
    _id: ObjectId("661ca5f3389e0c978178b79d"),
    StudentID: 2,
    SchoolID_FK: 2,
    UserID_FK: 2,
    Name: 'Jane Smith',
    Age: 22,
    PhoneNumber: '555-0002',
    Gender: 'Female',
    Course: 'English Literature',
    CreditLoad: 12,
    GPA: 3.8
  }
  {
    _id: ObjectId("661ca5f3389e0c978178b79e"),
    StudentID: 3,
    SchoolID_FK: 3,
    UserID_FK: 3,
    Name: 'Alex Johnson',
    Age: 19,
    PhoneNumber: '555-0003',
    Gender: 'Non-Binary',
    Course: 'Biology',
    CreditLoad: 16,
    GPA: 3.6
```

Query 4 Counting number of students based on gender

```
> db.student.aggregate([ { $group: { _id: "$Gender", total: { $sum: 1 } } }] )
< {
    _id: 'Male',
    total: 7
  }
  {
    _id: 'Female',
    total: 7
  }
  {
    _id: 'Non-Binary',
    total: 6
  }
```

Query 5 Students whose name starts with A

```
> db.student.find({Name: /^A/ })
< {
    _id: ObjectId("661ca5f3389e0c978178b79e"),
    StudentID: 3,
    SchoolID_FK: 3,
    UserID_FK: 3,
    Name: 'Alex Johnson',
    Age: 19,
    PhoneNumber: '555-0003',
    Gender: 'Non-Binary',
    Course: 'Biology',
    CreditLoad: 16,
    GPA: 3.6
  }
  {
    _id: ObjectId("661ca5f3389e0c978178b7a7"),
    StudentID: 12,
    SchoolID_FK: 2,
    UserID_FK: 12,
    Name: 'Alexis King',
    Age: 22,
    PhoneNumber: '555-0012',
    Gender: 'Non-Binary',
    Course: 'History',
    CreditLoad: 12,
```