**SVKM’s NMIMS**

**School of Technology Management & Engineering, Chandigarh**

A.Y. 2023 - 24

**Course: Database Management Systems**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | Btech CE | |
| Semester | Semester 4 | |
| Name of the Project: | Mental Health | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| 1 | A066 | Gaganchandra Reddy Gollapalli |
| 1 | A071 | Parth Chikodi |
| 1 | A072 | Siddesh Lohkare |
| Date of Submission: 02-04-2024 | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No. | Name: | Contribution |
| A066 | Gaganchandra Reddy Gollapalli | E-R Diagram, SQL Queries |
| A071 | Parth Chikodi | Normalization, SQL Queries |
| A072 | Siddesh Lohkare | Relational Model, SQL Queries |

**Github link of your project:** [**https://github.com/Siddesh3108/DBMS-Project**](https://github.com/Siddesh3108/DBMS-Project)

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Project Report**

**Mental Health**

**by**

**Gaganchandra Reddy, Roll number: A066**

**Parth Chikodi, Roll number: A071**

**Siddesh Lohkare, Roll number: A072**

**Course: DBMS**

**AY: 2023-24**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline | 4 |
| **2** | Components of Database Design | 4-7 |
| **3** | Entity Relationship Diagram | 8 |
| **4** | Relational Model | 9 |
| **5** | Normalization | 10 |
| **6** | SQL Queries | 11-21 |
| **7** | Learning from the Project | 21 |
| **8** | Project Demonstration | 22-23 |
| **9** | Self-learning beyond classroom | 23 |
| **10** | Learning from the project | 23 |
| **8** | Challenges faced | 24 |
| **9** | Conclusion | 24 |

1. **Storyline**

Mental health is a crucial aspect of overall well-being, encompassing emotional, psychological, and social well-being. It influences how individuals think, feel, and act, impacting their ability to cope with stress, interact with others, and make decisions. Good mental health is essential for functioning effectively in daily life, maintaining relationships, and achieving personal goals. Conversely, mental health disorders can significantly impair these abilities, leading to distress and dysfunction. Understanding mental health is vital for promoting resilience, seeking appropriate support, and fostering a supportive environment for individuals to thrive.

Mental health score calculation plays a role in assessing and monitoring mental well-being. It involves evaluating various factors such as emotional state, stress levels, coping mechanisms, and social support. By assigning numerical values to these factors and analyzing the overall score, professionals can gauge an individual's mental health status and tailor interventions accordingly. This approach enhances early detection of issues, facilitates targeted interventions, and promotes holistic mental health care. Integrating mental health score calculation into broader discussions on mental health promotes awareness, encourages proactive self-care, and contributes to improved mental health outcomes.

**II. Components of Database Design**

**User:**

UserID (Primary Key)

Username

Email

Password

DateOfBirth

Gender

**Assessment:**

AssessmentID (Primary Key)

UserID (Foreign Key)

Assessment\_date

**Answers:**

AnswerID (Primary Key)

UserID (Foreign Key)

QuestionID (Foreign Key)

Answer\_text

**Score:**

ScoreID (Primary Key)

UserID (Foreign Key)

AssessmentID (Foreign Key)

ScoreValue

Date

**AssessmentResponse:**

AssessmentResponseID (Primary Key)

AssessmentID (Foreign Key)

QuestionID (Foreign Key)

AnswerID (Foreign Key)

**Threshold:**

ThresholdID (Primary Key)

Score\_id (Foreign key)

Description\_text

**ThresholdRange:**

RangeID (Primary Key)

ThresholdID (Foreign Key)

MinValue

MaxValue

**UserGroup:**

GroupID (Primary Key)

GroupName

**GroupMembership:**

MembershipID (Primary Key)

UserID (Foreign Key)

GroupID (Foreign Key)

**Form:**

Form\_ID(Primary key)

Form\_title

Form\_Des

**Questions:**

Questions\_ID(primary key)

form\_ID (Foreign Key)

Question\_text

**ActivityLog:**

LogID (Primary Key)

UserID (Foreign Key)

ActivityType

Timestamp

Duration

Notes

**Reminder:**

ReminderID (Primary Key)

UserID (Foreign Key)

ReminderText

ReminderDate

IsCompleted

Notes

**Contact:**

ContactID (Primary Key)

UserID (Foreign Key)

Name

Email

Phone

Relationship

Notes

**Expense:**

ExpenseID (Primary Key)

UserID (Foreign Key)

Date

Amount

Category

Description

Notes

**User:**

**Relationships:**

One-to-One with Assessment

One-to-Many with GroupMembership

One-to-Many with ActivityLog

One-to-Many with Reminder

One-to-Many with Contact

One-to-Many with Expense

Diamond Shape Label: "Has"

**Assessment:**

**Relationships:**

One-to-One with User

One-to-Many with Score

One-to-Many with AssessmentResponse

Diamond Shape Label: "Belongs To"

**Answer:**

**Relationships:**

One-to-Many with User

One-to-Many with Questions

Diamond Shape Label: "Chosen By"

**Score:**

**Relationships:**

One-to-Many with User

One-to-One with Assessment

Diamond Shape Label: "Assigned For"

**AssessmentResponse:**

**Relationships:**

One-to-Many with Assessment

One-to-Many with Questions

One-to-Many with Answer

Diamond Shape Label: "Contains"

**Threshold:**

**Relationships:**

One-to-Many with ThresholdRange

Diamond Shape Label: "Has"

**ThresholdRange:**

**Relationships:**

Many-to-One with Threshold

Diamond Shape Label: "Part Of"

**UserGroup:**

**Relationships:**

Many-to-One with User (through GroupMembership)

Diamond Shape Label: "Includes"

**GroupMembership:**

**Relationships:**

One-to-Many with User

One-to-Many with UserGroup

Diamond Shape Label: "Belongs To"

**Form:**

**Relationships:**

One-to-Many with Questions

Diamond Shape Label: "Contains"

**Questions:**

**Relationships:**

One-to-Many with Form

One-to-Many with AssessmentResponse

One-to-Many with Answer

Diamond Shape Label: "Part Of"

**ActivityLog:**

**Relationships:**

One-to-Many with User

Diamond Shape Label: "Records"

**Reminder:**

**Relationships:**

One-to-Many with User

Diamond Shape Label: "Is For"

**Contact:**

**Relationships:**

One-to-Many with User

Diamond Shape Label: "Related To"

**Expense:**

**Relationships:**

One-to-Many with User

Diamond Shape Label: "Incurred By"

**III. Entity Relationship Diagram**

A diagram of a network

Description automatically generated

**IV. Relational Model**

A white board with text on it

Description automatically generated

**V. Normalization**

First Normal Form (1NF):

In 1NF, each table should have a primary key, and each column should contain atomic (indivisible) values. There should be no repeating groups or arrays within a column.

Example: In the User table, each attribute (User\_ID, User\_name, User\_email, User\_password, User\_dob, User\_gender) contains atomic values, and there are no repeating groups.

Second Normal Form (2NF):

For a table to be in 2NF, it must first be in 1NF. Additionally, every non-prime attribute should be fully functionally dependent on the primary key.

Example: In the Score table, Score\_Value and Date\_Score depend on both User\_ID and Assessment\_ID, which form the composite primary key. There are no partial dependencies.

Third Normal Form (3NF):

A table is in 3NF if it is in 2NF and there are no transitive dependencies. In other words, non-prime attributes should not depend on other non-prime attributes.

Example: In the Threshold table, Description\_text and Title\_text depend only on the Assessment\_ID, which is the primary key. There are no transitive dependencies.

Boyce-Codd Normal Form (BCNF):

This in BCNF.

**VI. SQL Queries**

CREATE DATABASE aq;

USE aq;

CREATE TABLE User (

User\_ID INT NOT NULL PRIMARY KEY,

User\_name VARCHAR(50) NOT NULL,

User\_password VARCHAR(50) NOT NULL,

User\_dob DATE NOT NULL,

User\_gender CHAR(1) NOT NULL

);

INSERT INTO User (User\_ID, User\_name, User\_password, User\_dob, User\_gender)

VALUES

(1, 'John Doe', 'password123', '1990-05-15', 'M'),

(2, 'Jane Smith', 'pass@123', '1985-08-25', 'F'),

(3, 'Alice Johnson', 'securepwd', '1995-03-10', 'F'),

(4, 'Bob Williams', 'bob@123', '1988-11-30', 'M'),

(5, 'Emily Brown', 'emilypass', '1992-09-20', 'F'),

(6, 'Michael Clark', 'mike@123', '1987-06-18', 'M'),

(7, 'Sarah Wilson', 'sarahpass', '1998-04-05', 'F'),

(8, 'David Martinez', 'david@123', '1993-12-12', 'M'),

(9, 'Jennifer Anderson', 'jenpass', '1996-10-30', 'F'),

(10, 'William Taylor', 'will@123', '1989-07-22', 'M');

CREATE TABLE User\_Group (

Group\_ID INT NOT NULL PRIMARY KEY,

Group\_name VARCHAR(50) NOT NULL

);

INSERT INTO User\_Group (Group\_ID, Group\_name)

VALUES

(1, 'Admins'),

(2, 'Users'),

(3, 'Managers'),

(4, 'Developers'),

(5, 'Testers'),

(6, 'Support'),

(7, 'Sales'),

(8, 'Marketing'),

(9, 'Finance'),

(10, 'HR');

CREATE TABLE Group\_Membership (

Membership\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Group\_ID INT NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID),

FOREIGN KEY (Group\_ID) REFERENCES User\_Group(Group\_ID)

);

INSERT INTO Group\_Membership (Membership\_ID, User\_ID, Group\_ID)

VALUES

(1, 1, 1),

(2, 2, 2),

(3, 3, 3),

(4, 4, 4),

(5, 5, 5),

(6, 6, 6),

(7, 7, 7),

(8, 8, 8),

(9, 9, 9),

(10, 10, 10);

CREATE TABLE Assessment (

Assessment\_id INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Assessment\_date DATE NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO Assessment (Assessment\_id, User\_ID, Assessment\_date)

VALUES

(1, 1, '2024-01-15'),

(2, 2, '2024-02-20'),

(3, 3, '2024-03-10'),

(4, 4, '2024-04-05'),

(5, 5, '2024-05-12'),

(6, 6, '2024-06-25'),

(7, 7, '2024-07-08'),

(8, 8, '2024-08-14'),

(9, 9, '2024-09-30'),

(10, 10, '2024-10-22');

CREATE TABLE Score (

Score\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Assessment\_ID INT NOT NULL,

Score\_Value INT NOT NULL,

Date\_Score DATE NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID),

FOREIGN KEY (Assessment\_ID) REFERENCES Assessment(Assessment\_id)

);

INSERT INTO Score (Score\_ID, User\_ID, Assessment\_ID, Score\_Value, Date\_Score)

VALUES

(1, 1, 1, 80, '2024-01-15'),

(2, 2, 2, 75, '2024-02-20'),

(3, 3, 3, 90, '2024-03-10'),

(4, 4, 4, 85, '2024-04-05'),

(5, 5, 5, 70, '2024-05-12'),

(6, 6, 6, 78, '2024-06-25'),

(7, 7, 7, 82, '2024-07-08'),

(8, 8, 8, 88, '2024-08-14'),

(9, 9, 9, 92, '2024-09-30'),

(10, 10, 10, 95, '2024-10-22');

CREATE TABLE Answers (

Answer\_id INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Question\_id INT NOT NULL,

Answer\_text VARCHAR(150) NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO Answers (Answer\_id, User\_ID, Question\_id, Answer\_text)

VALUES

(1, 1, 1, 'Yes'),

(2, 2, 1, 'No'),

(3, 3, 1, 'Sometimes'),

(4, 4, 1, 'Yes'),

(5, 5, 1, 'No'),

(6, 6, 1, 'No'),

(7, 7, 1, 'Yes'),

(8, 8, 1, 'Sometimes'),

(9, 9, 1, 'No'),

(10, 10, 1, 'Yes');

CREATE TABLE Threshold (

ID INT NOT NULL PRIMARY KEY,

Assessment\_ID INT NOT NULL,

Description\_text VARCHAR(250) NOT NULL,

Title\_text VARCHAR(150) NOT NULL,

FOREIGN KEY (Assessment\_ID) REFERENCES Assessment(Assessment\_id)

);

INSERT INTO Threshold (ID, Assessment\_ID, Description\_text, Title\_text)

VALUES

(1, 1, 'Mild Concerns', 'Mental Health Score'),

(2, 2, 'Moderate Concerns', 'Mental Health Score'),

(3, 3, 'Severe Concerns', 'Mental Health Score'),

(4, 4, 'Mild Concerns', 'Mental Health Score'),

(5, 5, 'Moderate Concerns', 'Mental Health Score'),

(6, 6, 'Mild Concerns', 'Mental Health Score'),

(7, 7, 'Moderate Concerns', 'Mental Health Score'),

(8, 8, 'Severe Concerns', 'Mental Health Score'),

(9, 9, 'Mild Concerns', 'Mental Health Score'),

(10, 10, 'Moderate Concerns', 'Mental Health Score');

CREATE TABLE Form (

Form\_ID INT NOT NULL PRIMARY KEY,

Form\_title VARCHAR(50) NOT NULL,

Form\_Des VARCHAR(130) NOT NULL

);

INSERT INTO Form (Form\_ID, Form\_title, Form\_Des)

VALUES

(1, 'Mental Health Assessment Form', 'Assessment form for mental health evaluation'),

(2, 'Feedback Form', 'Form for collecting user feedback'),

(3, 'Employee Satisfaction Survey', 'Survey to gauge employee satisfaction'),

(4, 'Training Feedback Form', 'Form to gather feedback on training programs'),

(5, 'Customer Satisfaction Survey', 'Survey to measure customer satisfaction');

CREATE TABLE Reminder (

Log\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Reminder\_Text VARCHAR(250) NOT NULL,

Reminder\_Date DATE NOT NULL,

Notes\_Text VARCHAR(250) NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO Reminder (Log\_ID, User\_ID, Reminder\_Text, Reminder\_Date, Notes\_Text)

VALUES

(1, 1, 'Appointment with therapist', '2024-04-10', 'Discuss recent stress triggers'),

(2, 2, 'Complete feedback form', '2024-03-25', 'Provide detailed feedback on services'),

(3, 3, 'Follow up on assessment results', '2024-03-15', 'Review suggestions for improvement'),

(4, 4, 'Attend team meeting', '2024-04-01', 'Discuss project progress'),

(5, 5, 'Submit expense report', '2024-05-20', 'Include all receipts and details'),

(6, 6, 'Review performance goals', '2024-06-30', 'Prepare for quarterly review'),

(7, 7, 'Submit sales report', '2024-07-15', 'Include sales figures and analysis'),

(8, 8, 'Marketing campaign brainstorming', '2024-08-05', 'Discuss ideas for upcoming campaign'),

(9, 9, 'Financial audit preparation', '2024-09-25', 'Gather financial documents for audit'),

(10, 10, 'HR policy review', '2024-10-15', 'Review and update HR policies');

CREATE TABLE Contact (

Contact\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Contact\_Name TEXT NOT NULL,

Phone INT NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO Contact (Contact\_ID, User\_ID, Contact\_Name, Phone)

VALUES

(1, 1, 'Therapist Office', 0000000001),

(2, 2, 'Customer Support', 0000000002),

(3, 3, 'Manager', 0000000003),

(4, 4, 'Team Lead', 0000000004),

(5, 5, 'Finance Department', 0000000005),

(6, 6, 'Support Desk', 000000006),

(7, 7, 'Sales Manager', 0000000007),

(8, 8, 'Marketing Team', 0000000008),

(9, 9, 'Finance Director', 0000000009),

(10, 10, 'HR Manager', 0000000010);

CREATE TABLE Expense (

Expense\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Date\_text DATE NOT NULL,

Amount\_Int INT NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO Expense (Expense\_ID, User\_ID, Date\_text, Amount\_Int)

VALUES

(1, 1, '2024-03-10', 100),

(2, 2, '2024-02-15', 75),

(3, 3, '2024-01-20', 150),

(4, 4, '2024-04-05', 200),

(5, 5, '2024-05-12', 50),

(6, 6, '2024-06-18', 120),

(7, 7, '2024-07-30', 180),

(8, 8, '2024-08-10', 90),

(9, 9, '2024-09-05', 250),

(10, 10, '2024-10-01', 150);

CREATE TABLE ActivityLog (

Log\_ID INT NOT NULL PRIMARY KEY,

User\_ID INT NOT NULL,

Activity\_type\_text VARCHAR(250) NOT NULL,

Duration\_int INT NOT NULL,

FOREIGN KEY (User\_ID) REFERENCES User(User\_ID)

);

INSERT INTO ActivityLog (Log\_ID, User\_ID, Activity\_type\_text, Duration\_int)

VALUES

(1, 1, 'Exercise', 60),

(2, 2, 'Reading', 30),

(3, 3, 'Meeting', 120),

(4, 4, 'Coding', 180),

(5, 5, 'Walking', 45),

(6, 6, 'Training', 90),

(7, 7, 'Sales Call', 45),

(8, 8, 'Marketing Campaign', 240),

(9, 9, 'Financial Analysis', 150),

(10, 10, 'HR Training', 120);

**Questions:**

1. Retrieve the names of users who do not have any associated contacts.

SELECT \* FROM User WHERE User\_name = 'John Doe';

SELECT \* FROM User WHERE User\_name != 'John Doe';

SELECT \* FROM Score WHERE Score\_Value > 80;

SELECT \* FROM Score WHERE Score\_Value < 80;

SELECT \* FROM Score WHERE Score\_Value >= 80;

SELECT \* FROM Score WHERE Score\_Value <= 80;

SELECT \* FROM User WHERE User\_gender IN ('M', 'F');

SELECT \* FROM User WHERE User\_gender = 'M' AND User\_dob < '1990-01-01';

SELECT \* FROM User WHERE User\_gender = 'F' OR User\_dob >= '1990-01-01';

SELECT \* FROM User;

INSERT INTO User (User\_ID, User\_name, User\_password, User\_dob, User\_gender)

VALUES (11, 'Jessica Johnson', 'jess@123', '1991-04-20', 'F');

SELECT \* FROM Score ORDER BY Score\_Value DESC;

SELECT u.User\_name, g.Group\_name

FROM User u

JOIN Group\_Membership gm ON u.User\_ID = gm.User\_ID

JOIN User\_Group g ON gm.Group\_ID = g.Group\_ID;

SELECT u.User\_name, MIN(s.Score\_Value) AS Min\_Score, MAX(s.Score\_Value) AS Max\_Score

FROM User u

JOIN Score s ON u.User\_ID = s.User\_ID

GROUP BY u.User\_name;

SELECT User\_name FROM User

UNION

SELECT Contact\_Name FROM Contact;

SELECT User\_name FROM User

UNION ALL

SELECT Contact\_Name FROM Contact;

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

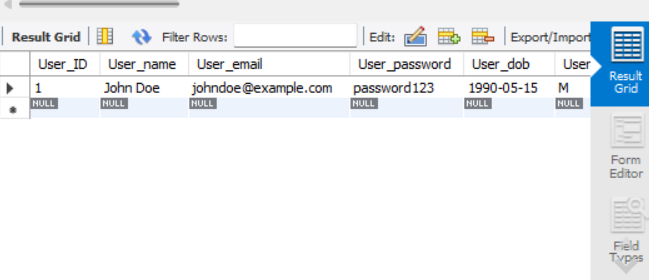
SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

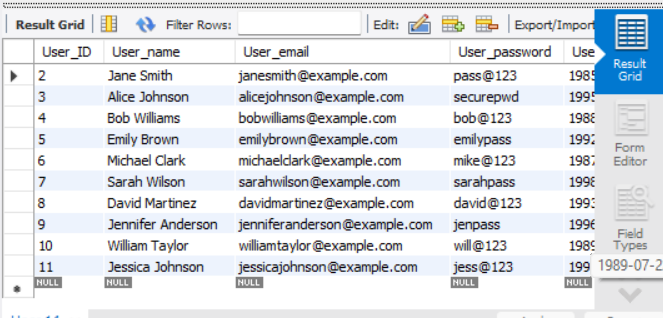
**Retrieve the user details for the user with the username 'John Doe'.**

SELECT \* FROM User WHERE User\_name = 'John Doe';



**Fetch all user details except for the user with the username 'John Doe'.**

SELECT \* FROM User WHERE User\_name != 'John Doe';



**Retrieve all scores where the score value is greater than 80.**

SELECT \* FROM Score WHERE Score\_Value > 80;

A screenshot of a computer

Description automatically generated

**Fetch all scores where the score value is less than 80.**

SELECT \* FROM Score WHERE Score\_Value < 80;

A screenshot of a computer

Description automatically generated

**Get all scores where the score value is greater than or equal to 80.**

SELECT \* FROM Score WHERE Score\_Value >= 80;

A screenshot of a computer

Description automatically generated

**Retrieve scores where the score value is less than or equal to 80.**

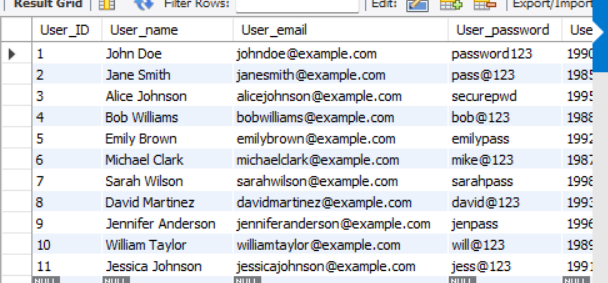
SELECT \* FROM Score WHERE Score\_Value <= 80;

A screenshot of a computer

Description automatically generated

**Retrieve user details for users with genders either Male or Female.**

SELECT \* FROM User WHERE User\_gender IN ('M', 'F');



**Fetch user details for male users born before January 1, 1990.**

SELECT \* FROM User WHERE User\_gender = 'M' AND User\_dob < '1990-01-01';

A screenshot of a computer

Description automatically generated

**Retrieve user details for female users or users born on or after January 1, 1990.**

SELECT \* FROM User WHERE User\_gender = 'F' OR User\_dob >= '1990-01-01';

A screenshot of a computer

Description automatically generated

**Fetch all user details from the database.**

SELECT \* FROM User;

A screenshot of a computer

Description automatically generated

**Insert a new user record for Jessica Johnson with her email, password, date of birth, and gender.**

INSERT INTO User (User\_ID, User\_name, User\_email, User\_password, User\_dob, User\_gender)

VALUES (11, 'Jessica Johnson', 'jessicajohnson@example.com', 'jess@123', '1991-04-20', 'F');

**Retrieve all scores from the database and sort them in descending order based on the score value.**

SELECT \* FROM Score ORDER BY Score\_Value DESC;

A screenshot of a computer

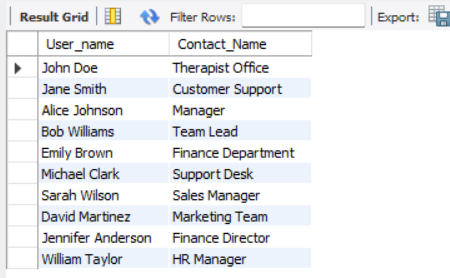
Description automatically generated

**Retrieve the names of users and their corresponding contact names.**

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;



**Fetch the user names and their associated contact names through an inner join between the User and Contact tables.**

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

A screenshot of a computer

Description automatically generated

**Retrieve the user names and their corresponding contact names by performing an inner join operation between the User and Contact tables.**

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

A screenshot of a computer

Description automatically generated

**Fetch the names of users and their associated contact names using an inner join between the User and Contact tables.**

SELECT u.User\_name, c.Contact\_Name

FROM User u

INNER JOIN Contact c ON u.User\_ID = c.User\_ID;

A screenshot of a computer

Description automatically generated

**VI. Project demonstration**

For our project demonstration on mental health score calculation, we utilized a combination of MySQL for writing queries and Draw.io application for creating an E-R (Entity-Relationship) Model and Relational Model. Here's an expanded overview of how we employed these tools:

1. **MySQL for Querying:** We leveraged MySQL, a widely-used relational database management system, to handle the storage and querying of data related to mental health assessments. MySQL provided us with a robust platform to store structured data efficiently and to execute complex queries to retrieve, manipulate, and analyze this data.
   * We designed a database schema tailored to our specific requirements, including tables for storing user information, assessment scores, responses, and other relevant data points.
   * Through MySQL, we were able to write SQL queries to calculate mental health scores based on various parameters and factors. These queries allowed us to aggregate, analyze, and derive meaningful insights from the collected data.
2. **Draw.io for E-R Model and Relational Model:** Draw.io served as a valuable tool for visualizing the structure and relationships within our database through the creation of both E-R and Relational Models.
   * **Entity-Relationship (E-R) Model:** Using Draw.io, we developed an E-R diagram to represent the entities involved in our mental health assessment system, along with their attributes and relationships. This provided a clear visualization of the data model and helped in understanding the conceptual framework of our database design.
   * **Relational Model:** In addition to the E-R diagram, Draw.io allowed us to construct a Relational Model that outlined the tables, attributes, and relationships in a more structured format. This model served as a blueprint for implementing the database schema in MySQL, ensuring consistency and coherence between the conceptual design and its practical implementation.

By combining the capabilities of MySQL for data management and querying with Draw.io for visual modelling, we were able to develop a comprehensive solution for mental health score calculation. This integrated approach facilitated efficient database design, data manipulation, and visualization, ultimately contributing to the success of our project demonstration.

**VII. Self -Learning beyond classroom**

Engaging with the mental health database this provided a learning experience in managing query and entries in the database. This activity has given me insights into principles of database design, such as organizing tables to accurately represent entities and their relationships. This provided an insight in how the database of mental health score is calculated and how the improvement steps are decided. This also provided an insight in understanding various concepts of mental health which will be taken into consideration in questions for score calculation.

**VIII. Learning from the Project**

This project has helped me gain the following concepts: -  
Database Design and Normalization: Through this project, you likely gained a deeper understanding of database design principles, including entity-relationship modeling, normalization (1NF, 2NF, 3NF, and BCNF), primary keys, foreign keys, and ensuring data integrity and efficiency.

SQL Queries: By implementing SQL queries for creating tables, inserting data, querying information, and performing joins, you likely improved your skills in structuring and retrieving data from databases.

Entity Relationships: Designing and managing entity relationships such as users, groups, assessments, scores, answers, forms, reminders, contacts, expenses, and activity logs would have provided you with hands-on experience in defining complex relationships in a database system.

Project Management: Coordinating various aspects of the project, including storyline development, character roles, database schema design, SQL query implementation, and normalization, would have honed your project management and organizational skills.

Domain Knowledge: Working on a project related to mental health services likely enhanced your understanding of mental health assessment processes, support services, data analysis in healthcare settings, and the importance of data accuracy and security in sensitive domains.

**IX. Challenges Faced**

Here are some challenges faced:-

1. **Data Privacy and Security:** Ensuring the confidentiality and security of sensitive client information posed a significant challenge. Implementing robust data encryption, access control measures, and compliance with data protection regulations required meticulous planning and execution.
2. **Database Complexity:** Managing a complex database structure with interrelated tables and dependencies presented challenges in terms of data integrity, normalization, and query optimization. Developing efficient database schemas and query optimization strategies were key areas of focus.
3. **Redundancy:** Managing redundancies if any .

**X. Conclusion**

The project on Mental Health Services (MHS) organization has been a comprehensive learning journey that has significantly enhanced my skills and knowledge in various domains. From designing a robust database schema to implementing SQL queries, managing entity relationships, and understanding the intricacies of mental health services, this project has provided invaluable insights and practical experience.

Through this project, I have gained proficiency in database design principles, normalization techniques, and SQL querying, which are foundational skills in the field of data management. Moreover, working with a domain-specific context like mental health services has deepened my understanding of healthcare processes, data privacy considerations, and the importance of data accuracy and integrity in sensitive domains.

Furthermore, the project has honed my project management skills by requiring careful planning, coordination, and execution of tasks such as defining the storyline, assigning character roles, and ensuring the functionality and security of the database system.

Overall, this project has not only equipped me with technical skills but also provided a valuable opportunity to apply these skills in a real-world scenario, thereby preparing me for future endeavors in database management, software development, and healthcare-related projects.