Group 22

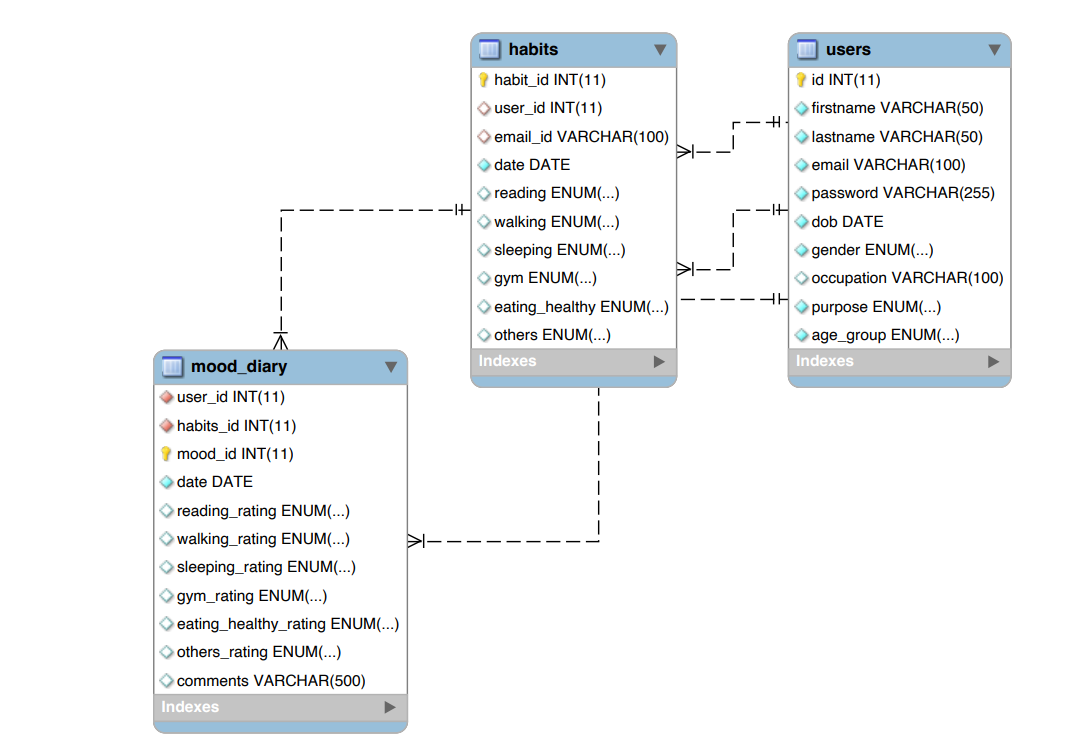
Ritika Shrivastava (Team Lead)

Rutuja Jangle

Naveen Addanki

**ADT Final Project Part 2 – Database Design**

# **1. Create Conceptual Diagram/Schema for database:**



The database consists of three tables: **users**, **habits**, and **mood\_diary**.

### **users Table/entity**

* id: The primary key for the users table, a unique identifier for each user record.
* firstname: The given name of the user.
* lastname: The family or surname of the user.
* email: The user's email address, used for identification and communication.
* password: The encrypted password for the user's account.
* dob: The date of birth of the user.
* gender: The gender of the user; limited to 'male', 'female', or 'other'.
* occupation: The user's job or profession.
* purpose: The user's intended use of the app, such as tracking weekly progress, improving productivity, ensuring accountability, or building new habits.
* age**\_**group: The user's age category, which can be 'under 18', '18-25', '26-40', or 'above 40'.

### **habits Table/entity**

* habit\_id: The primary key for the habits table, a unique identifier for each habit record.
* user\_id: A foreign key that references the id from the users table to associate a habit record with a user.
* email\_id: A foreign key that references the email from the ‘users’ table; its usage alongside user\_id is atypical and might be redundant or for a specific purpose not clear in the context.
* date: The date when the habit was tracked.
* reading: A boolean-like value indicating whether the user has read.
* walking: A boolean-like value indicating whether the user has walked.
* sleeping: A boolean-like value indicating the quality of sleep or if the user has slept well.
* gym: A boolean-like value indicating whether the user has gone to the gym.
* eating\_healthy: A boolean-like value indicating whether the user has eaten healthy.
* others: A boolean-like value for tracking any other habits not explicitly mentioned.

### **mood\_diary Table/entity**

* user\_id: A foreign key that references the id from the users table, linking the mood diary entry to a user.
* habits\_id: A foreign key that references the habit\_id from the habits table, suggesting a relationship between habits and mood.
* mood: An identifier linking to another table or a set of values that represent the user's mood.
* date: The date of the mood diary entry.
* reading\_rating, walking\_rating, sleeping\_rating, gym\_rating, eating\_healthy\_rating, others\_rating: These fields are to store a form of evaluation or sentiment about each habit in relation to the user's mood.
* comments: A text field that allows users to add any additional notes or comments about their mood or habits for the day.

Since we have created our own database for this project, our tables are normalized (they satisfy 1NF, 2NF and 3NF).

**Purpose:**

* users Entity: This is the central entity that holds information about the users of the habit tracker WebApp. Key attributes include id, firstname, lastname, email, etc. The id attribute is the primary key.
* habits Entity: This entity contains data on the habits that users track, like reading, walking, sleeping, etc. It is related to the users entity by user\_id which is a foreign key referencing users(id). It also contains email\_id, which suggests another relation to the users entity, specifically referencing users(email).
* mood\_diary Entity: This holds entries about the user's mood and how it correlates with their habits. It is linked to habits by habits\_id, which is a foreign key referencing habits(habit\_id), and to users by user\_id.

**Relationships:**

* users to habits: A one-to-many relationship where a single user can have many habit records.
* habits to mood\_diary: A one-to-many relationship where a single habits record can be associated with multiple mood diary entries.
* users to mood\_diary: An indirect one-to-many relationship through the habits table, suggesting that a user's mood entries are associated with their habits.

**2. Database:**

**Database Constraints**:

* Primary Keys: The primary keys used in our schema are id for users, habit\_id for habits and mood\_id for mood\_diary tables respectively.
* Foreign Keys: The foreign key references used in our schema are as follows:

user\_id and email\_id in habits tables references id and email in users table; the user\_id in mood\_diary table references id in users table and the habits\_id in mood\_diary table references habit\_id in habits table respectively.

* NOT NULL constraints on critical attributes like firstname, lastname, id, password and more are applied to ensure data integrity.
* UNIQUE constraint on email in users to ensure each user has a unique email.
* ENUM data types are used for attributes with a limited set of valid values, such as gender, purpose, and age\_group.

The **NOT NULL** constraint in SQL is a rule applied to a column in a database table that prevents null values (or empty values) from being inserted into that column. This is important for several reasons:

* Data Completeness: Certain fields are critical for the records to be useful or valid. For example, in our user profile, the email or password fields are essential for account creation and user identification. Allowing null values in these fields would create incomplete records that could not be used for their intended purposes.
* Data Integrity: It helps maintain the integrity of the data. For instance, if the date field in the habits table were allowed to be null, it would be unclear when the recorded habit took place, which defeats the purpose of tracking habits over time.
* Business Rules: Many times, non-null constraints enforce business rules. For example, if the application's logic requires that all users must provide their date of birth (dob), then the dob field should have a NOT NULL constraint to ensure this rule is always followed.
* Referential Integrity: For foreign keys, NOT NULL ensures that every record in the child table always refers to an existing record in the parent table, preventing orphan records.

Enforcing **NOT NULL** constraints on these attributes ensures that every record in the database has meaningful and complete data, enabling the application to function correctly and reliably.

**3. Write code to create a database and build queries :**

For writing this code, we took help from MySQL.com to check for a few constraints and the correct syntax. We also took help from Professor Da Yan’s notes (ADC) for creating our schema properly and to understand the relationships that we could build between the different entities in our database.

-- Habit tracker project part 2

-- Team Members: Ritika Shrivastav, Rutuja Jangle, Naveen Addanki

-- Users table was created by Ritika including all the constraints which were discussed among all members of the team

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

firstname VARCHAR(50) NOT NULL,

lastname VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL, -- Assuming encryption is handled outside of the database

dob DATE NOT NULL,

gender ENUM('male', 'female', 'other') NOT NULL,

occupation VARCHAR(100), -- Can be NULL

purpose ENUM('weekly progress tracking', 'productivity', 'accountability', 'habit building') NOT NULL,

age\_group ENUM('under 18', '18-25', '26-40', 'above 40') NOT NULL

);

-- Habits table was created by Naveen which included column category discussions from all members

CREATE TABLE habits (

habit\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

email\_id VARCHAR(100),

date DATE NOT NULL,

reading ENUM('True', 'False') DEFAULT NULL,

walking ENUM('True', 'False') DEFAULT NULL,

sleeping ENUM('True', 'False') DEFAULT NULL,

gym ENUM('True', 'False') DEFAULT NULL,

eating\_healthy ENUM('True', 'False') DEFAULT NULL,

others ENUM('True', 'False') DEFAULT NULL,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (email\_id) REFERENCES users(email)

);

-- Each one of us inserted 5 entries in the users table making it a total of 15 entries initially,

-- starting with Ritika for the first 5,

-- followed by Rutuja for the next 5,

-- and Naveen inserted the last 5 entries

INSERT INTO users (firstname, lastname, email, password, dob, gender, occupation, purpose, age\_group)

VALUES

('John', 'Doe', CONCAT('john', 'doe', '@yahoo.com'), 'password1', '1990-05-15', 'male', 'Engineer', 'weekly progress tracking', '26-40'),

('Jane', 'Smith', CONCAT('jane', 'smith', '@gmail.com'), 'password2', '1985-08-22', 'female', 'Teacher', 'productivity', '26-40'),

('Michael', 'Johnson', CONCAT('michael', 'johnson', '@outlook.com'), 'password3', '1995-02-10', 'male', 'Doctor', 'accountability', '26-40'),

('Emily', 'Williams', CONCAT('emily', 'williams', '@edu.com'), 'password4', '2000-11-30', 'female', 'Student', 'habit building', '18-25'),

('David', 'Brown', CONCAT('david', 'brown', '@hotmail.com'), 'password5', '1978-07-03', 'male', 'Manager', 'weekly progress tracking', 'above 40'),

('Sarah', 'Miller', CONCAT('sarah', 'miller', '@yahoo.com'), 'password6', '1992-04-18', 'female', 'Software Developer', 'productivity', '26-40'),

('Ryan', 'Davis', CONCAT('ryan', 'davis', '@gmail.com'), 'password7', '1987-09-25', 'male', 'Artist', 'accountability', '26-40'),

('Olivia', 'Wilson', CONCAT('olivia', 'wilson', '@outlook.com'), 'password8', '1983-12-08', 'female', 'Writer', 'habit building', '26-40'),

('William', 'Moore', CONCAT('william', 'moore', '@edu.com'), 'password9', '2002-01-20', 'male', NULL, 'weekly progress tracking', 'under 18'),

('Emma', 'Taylor', CONCAT('emma', 'taylor', '@hotmail.com'), 'password10', '1975-06-12', 'female', 'Entrepreneur', 'productivity', 'above 40'),

('James', 'Anderson', CONCAT('james', 'anderson', '@yahoo.com'), 'password11', '1998-03-05', 'male', 'Consultant', 'accountability', '18-25'),

('Ava', 'Thomas', CONCAT('ava', 'thomas', '@gmail.com'), 'password12', '1993-10-14', 'female', 'Lawyer', 'habit building', '26-40'),

('Benjamin', 'Jackson', CONCAT('benjamin', 'jackson', '@outlook.com'), 'password13', '1980-09-02', 'male', 'Accountant', 'weekly progress tracking', 'above 40'),

('Sophia', 'Harris', CONCAT('sophia', 'harris', '@yahoo.com'), 'password14', '1989-07-29', 'female', 'Doctor', 'productivity', '26-40'),

('Ethan', 'White', CONCAT('ethan', 'white', '@edu.com'), 'password15', '2005-08-09', 'male', NULL, 'accountability', 'under 18');

-- Similarly for the habits table, Assuming some random habit data for each user, each one of us inserted 5 entries in the same order as above,

-- Ritika (first 5) -> Rutuja (next 5) -> Naveen (last 5)

INSERT INTO habits (user\_id, email\_id, date, reading, walking, sleeping, gym, eating\_healthy, others)

VALUES

(1, 'johndoe@yahoo.com', '2024-04-07', 'True', 'True', 'True', 'False', 'True', 'False'),

(2, 'janesmith@gmail.com', '2024-04-07', 'True', 'False', 'True', 'True', 'False', 'False'),

(3, 'michaeljohnson@outlook.com', '2024-04-07', 'True', 'True', 'False', 'True', 'True', 'False'),

(4, 'emilywilliams@edu.com', '2024-04-07', 'False', 'True', 'True', 'False', 'False', 'True'),

(5, 'davidbrown@hotmail.com', '2024-04-07', 'True', 'True', 'True', 'True', 'True', 'True'),

(6, 'sarahmiller@yahoo.com', '2024-04-07', 'False', 'True', 'True', 'False', 'True', 'True'),

(7, 'ryandavis@gmail.com', '2024-04-07', 'True', 'True', 'True', 'True', 'False', 'False'),

(8, 'oliviawilson@outlook.com', '2024-04-07', 'False', 'False', 'True', 'False', 'True', 'True'),

(9, 'williammoore@edu.com', '2024-04-07', 'True', 'False', 'False', 'True', 'False', 'True'),

(10, 'emmataylor@hotmail.com', '2024-04-07', 'True', 'True', 'True', 'True', 'True', 'False'),

(11, 'jamesanderson@yahoo.com', '2024-04-07', 'False', 'True', 'False', 'False', 'True', 'True'),

(12, 'avathomas@gmail.com', '2024-04-07', 'True', 'True', 'True', 'True', 'True', 'True'),

(13, 'benjaminjackson@outlook.com', '2024-04-07', 'False', 'False', 'True', 'False', 'True', 'False'),

(14, 'sophiaharris@yahoo.com', '2024-04-07', 'True', 'True', 'True', 'False', 'False', 'True'),

(15, 'ethanwhite@edu.com', '2024-04-07', 'False', 'False', 'False', 'True', 'True', 'False');

-- For checking whether the entries are inserted properly we used the select statement, coded by Ritika

select \* from habits;

-- Mood\_diary table was created by Rutuja which included the contraints discussions from all members

CREATE TABLE mood\_diary (

user\_id INT NOT NULL,

habits\_id INT NOT NULL,

mood\_id INT AUTO\_INCREMENT PRIMARY KEY,

date DATE NOT NULL,

reading\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

walking\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

sleeping\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

gym\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

eating\_healthy\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

others\_rating ENUM('1','2','3','4','5') DEFAULT NULL,

comments VARCHAR(500),

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (habits\_id) REFERENCES habits(habit\_id)

);

-- Similarly for the mood\_diary table, Assuming some random data for each user, each one of us inserted 5 ntries in the same order as the first two tables,

-- Ritika (first 5) -> Rutuja (next 5) -> Naveen (last 5)

INSERT INTO mood\_diary (user\_id, habits\_id, date, reading\_rating, walking\_rating, sleeping\_rating, gym\_rating, eating\_healthy\_rating, others\_rating, comments)

VALUES

(1, 47, '2024-04-07', '5', '4', '5', '3', '5', '4', 'Feeling great today!'),

(2, 33, '2024-04-07', '4', '5', '4', '4', '3', '5', 'Had a productive day.'),

(3, 34, '2024-04-07', '5', '4', '3', '5', '5', '4', 'Feeling energetic.'),

(4, 35, '2024-04-07', '3', '5', '5', '2', '4', '5', 'Could use some improvement in reading habit.'),

(5, 36, '2024-04-07', '4', '4', '4', '4', '4', '4', 'Maintaining a good routine.'),

(6, 37, '2024-04-07', '5', '3', '5', '3', '4', '4', 'Feeling focused.'),

(7, 38, '2024-04-07', '4', '4', '4', '4', '3', '3', 'Average day overall.'),

(8, 39, '2024-04-07', '3', '3', '4', '2', '4', '3', 'Could improve sleeping habits.'),

(9, 40, '2024-04-07', '4', '2', '2', '4', '3', '4', 'Good progress in gym.'),

(10, 41, '2024-04-07', '5', '5', '5', '5', '5', '5', 'Excellent day overall.'),

(11, 42, '2024-04-07', '3', '4', '2', '3', '4', '3', 'Need to work on sleeping habits.'),

(12, 43, '2024-04-07', '5', '5', '5', '5', '5', '5', 'Feeling accomplished.'),

(13, 44, '2024-04-07', '4', '3', '4', '2', '3', '3', 'Could improve eating habits.'),

(14, 45, '2024-04-07', '5', '4', '5', '4', '5', '4', 'Great day overall.'),

(15, 46, '2024-04-07', '3', '2', '2', '4', '3', '3', 'Need to be more consistent.');

-- Similarly like the habits table, for checking whether the data is inserted properly or not in the mood\_diary table we used the select statement, coded by Rutuja

select \* from mood\_diary;

**Authorship**:

* The users table was attributed to Ritika.
* The mood\_diary table was created by Rutuja.
* The habits table creation is credited to Naveen.
* Data insertion in all these three tables was a collaborative effort: Ritika for the first 5 entries, followed by Rutuja for the next 5 and then Naveen for the last 5.

**4. Overall Contribution Summary:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Task** | **Contribution** | **Avg Time spent (hrs)** |
| Ritika Shrivastava | Conceptual Schema  Database  Code | Provided ideas for the final database schema which included a discussion with Rutuja wherein we both went through some websites which gave us a clear understanding of the schema. We built our own database from scratch and hence we followed all the constraint related rules mentioned on mysql.com. We both contributed in laying out a basic structure of how our schema will look using the Jamboard.  Created `users` table along with its columns, constraints on the attributes, and inserted the initial 5 entries in each of the 3 tables in our database.  Contributed to SQL code for creating the `users` table and inserting initial 5 data entries in each table and the select statement to check whether the habits table entries are inserted properly or not. | The average time spent was around 3 hours in the initial discussion and then 1 hour for the code and at the last 30 mins for the final report. |
| Rutuja Jangle | Conceptual Schema  Database  Code | Contributed ideas for the initial database schema and attributes in each table. This was done using a Jamboard which we used initially to design the layout of our schema in order to understand what attributes we require in each of the tables and what constraints could be added to each attribute. My ADC notes helped me and Ritika to understand the relationships between the different entities in our database. Discussion included the types of columns in each table, the total number of tables and the constraints applied to each attribute of the tables.  Created `mood\_diary` table along with its columns, constraints on the attributes, and even the constraints of the habits table discussed along with all members and inserted the next 5 entries in each of the 3 tables in our database.  Contributed to SQL code for creating the `mood\_diary` table and inserting the next 5 data entries in each table and the select statement to check whether the mood\_diary table entries are inserted properly or not. | The average time spent was around 3 hours in the initial discussion and then 45 mins for the code and at the last 1 hour mins for the final report. |
| Naveen Addanki | Conceptual Schema  Database  Code | Contributed to schema design and relationship mapping.  Created `habits` table and established foreign key relationships.  Added SQL code for `habits` table and inserted the final set of data entries in each table. | The average time spent was around 30 mins for the code and then 1 hour for preparing the final report. |

We have uploaded our SQL code file along with our schema pdf and the Jamboard diagram on our github repository.