Database Design

Create database

All data is stored in a Database named Assignment2.

Two Tables: users&tasks to store user information and task information.

```
CREATE DATABASE assignment2;
GRANT USAGE ON *.* TO root@localhost IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON root.* TO assignment2@localhost;

FLUSH PRIVILEGES;
```

constraint:

The `userId` column in the `tasks` table references the `userId` column in the `users` table(`FOREIGN KEY (userId) REFERENCES users(userId)`) defines a foreign key constraint that ensures establishing a relationship between tasks and users.

The users table:

```
USE assignment2;
DROP TABLE IF EXISTS users;

CREATE TABLE users (
    userId INT AUTO_INCREMENT PRIMARY KEY,
    userName VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE
);

```

The `users` table is designed to store information about users. It contains columns representing various attributes of a user, such as their unique identifier, name, and password.

Attributes:

1. userId (INT AUTO_INCREMENT PRIMARY KEY):

   The `userId` column serves as a unique identifier for each user in the table. The value of this column is automatically incremented with each new user added to the table.

2. userName (VARCHAR(255) NOT NULL UNIQUE):

   The `userName` column stores the name of each user. The data type is VARCHAR(255) (variable-length character string with a maximum length of 255 characters). This column must contain a value for every user, and each name must be unique within the table and also used to identify and differentiate the user's account.

3. password (VARCHAR(255) NOT NULL):

   The `password` column stores the password of each user. The data type is VARCHAR(255) (variable-length character string with a maximum length of 255 characters). This column must contain a value for every user.(NOT NULL). It stores the password of each user, Used for authenticating users during login processes.

4. email()

   The email column stores user’ s email address. The data type is VARCHAR(255) (variable-length character string with a maximum length of 255 characters).This column is also set to be as NOT NULL to facilitate communications with users.

The tasks table:

```
1
2  USE assignment2;
3
4  DROP TABLE IF EXISTS tasks;
5
6  CREATE TABLE tasks (
7      taskId INT AUTO_INCREMENT PRIMARY KEY,
8      userId INT,
9      taskName VARCHAR(255) NOT NULL,
10     description TEXT,
11     priority ENUM('High', 'Medium', 'Low') NOT NULL,
12     dueDate DATE,
13     createdDate TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
14     status ENUM('In Progress', 'Over Due', 'Completed') NOT NULL DEFAULT 'In
   Progress',
15     FOREIGN KEY (userId) REFERENCES users(userId)
16 );
```

The `tasks` table is designed to store information about tasks. It contains columns representing various attributes of a task, such as its name, description, priority, due date, and the timestamp of its creation.

Attributes:

1. taskId (INT AUTO_INCREMENT PRIMARY KEY):
    - The `taskId` column is of type `INT` (integer) and is automatically incremented for each new task added to the table. It uniquely identifies each task on the table and is designated as the primary key, meaning it uniquely identifies each row in the table and is used for efficient data retrieval and indexing.
2. userId(INT FOREIGNKEY):
    - An integer field representing the user ID associated with the task. It is a foreign key referencing the `userId` column in the `users` table.
3. taskName (VARCHAR(255) NOT NULL):
    - The `taskName` column is of type `VARCHAR(255)` (variable-length character string with a maximum length of 255 characters) and is designated as `NOT NULL`, meaning it must contain a value for every task. It stores the name or title of each task.
4. description (TEXT):
    - The `description` column is of type `TEXT`, which allows it to store larger amounts of text data. It is optional and can contain additional details or a description of each task.
5. priority (ENUM('High', 'Medium', 'Low') NOT NULL):
    - The `priority` column is of type `ENUM`, which means it can only contain one of the specified values: 'High', 'Medium', or 'Low'. It is designated as `NOT NULL`, meaning it must contain a value for every task. It represents the priority level of each task, indicating its importance or urgency.
6. dueDate (DATE):
    - The `dueDate` column is of type `DATE`, which stores dates in the format 'YYYY-MM-DD'. It stores the due date of each task, indicating when the task is expected to be completed.
7. createdDate (TIMESTAMP DEFAULT CURRENT_TIMESTAMP):
    - The `createdDate` column is of type `TIMESTAMP` and has a default value of `CURRENT_TIMESTAMP`, which means it automatically records the timestamp of when each task record is inserted into the table. It stores the timestamp of when each task was created, providing a record of when the task was added to the system.