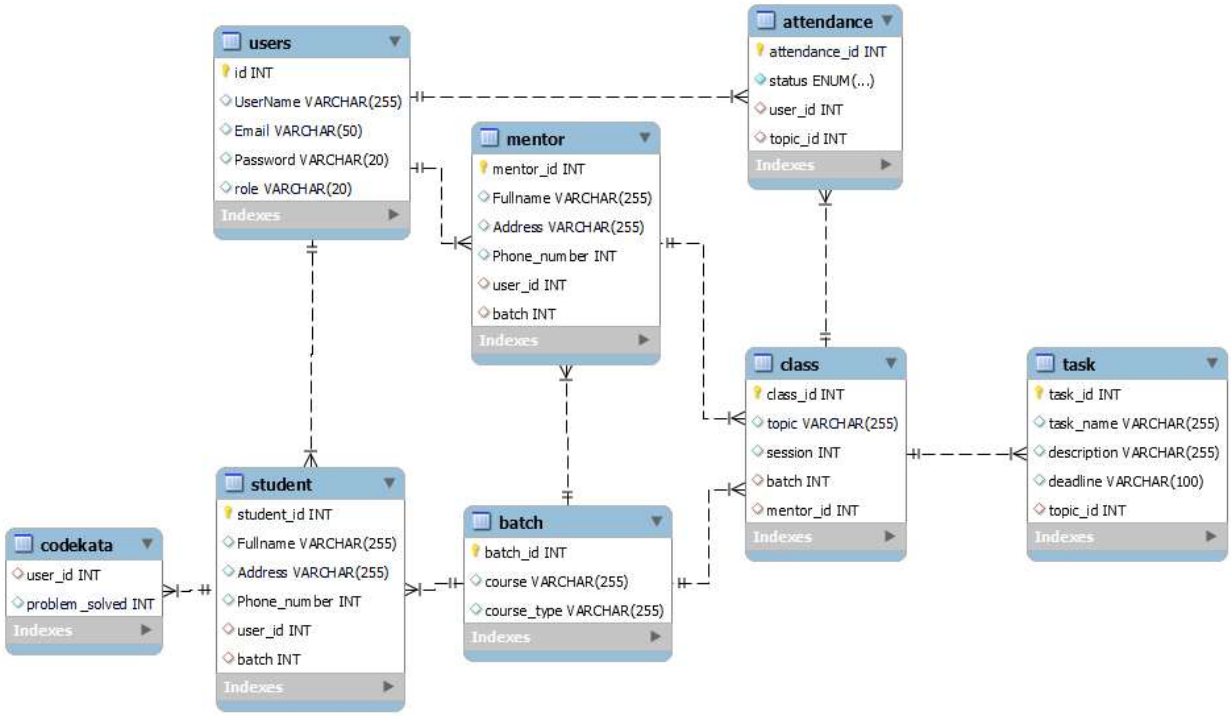
Database Model for Guvi Zen Class



Zen class Database Entities:

1. **Users**: Holds the information of registered user, including their username, email, password, and role. Depending on the role, it connects to the respective table (either "student" or "mentor").
2. **Student/Mentor (inherit Users)**: Table contains details of students/mentors, such as their full name, address, and phone number.
3. **Batch**: Stores information about the course assigned to a student (based on their choice) and the mentor associated with that batch.
4. **Class**: Table defines sessions for a course. It includes details such as the topic, session, the batch assigned to the student, and the mentor responsible for each topic in the class.
5. **Task**: Handles assignments or tasks assigned to students within a particular topic. Its columns include the task name, description, and deadline.
6. **Attendance**: Mark attendance for users during class sessions.
7. **Codekata**: Tracks the number of problems has been solved by students.

SQL COMMANDS

TO CREATE A DATABASE

```
CREATE DATABASE IF NOT EXIST Zen_db

USE zen_db
```

CREATE TABLES INSIDE THE DATABASE

Users

```
CREATE TABLE users (
  id int NOT NULL,
  UserName varchar(255) NOT NULL,
  Email varchar(50) NOT NULL,
  Password varchar(50) NOT NULL,
  role varchar(50) NOT NULL,
  PRIMARY KEY (id)
);
```

Student

```
CREATE TABLE student (
  student_id int NOT NULL,
  Fullname varchar(255) NOT NULL,
  Address varchar(255) NOT NULL,
  Phone_number int NOT NULL,
```

```
  user_id int NOT NULL,
  batch int NOT NULL,
  PRIMARY KEY (student_id),
 FOREIGN KEY (user_id) REFERENCES users (id),
 FOREIGN KEY (batch) REFERENCES batch (batch_id)
);
```

Mentor:

```
CREATE TABLE mentor (
  mentor_id int NOT NULL,
  Fullname varchar(255) NOT NULL,
  Address varchar(255) NOT NULL,
  Phone_number int NOT NULL,
  user_id int NOT NULL,
  batch int NOT NULL,
  PRIMARY KEY (mentor_id),
  FOREIGN KEY (user_id) REFERENCES users (id),
  FOREIGN KEY (batch) REFERENCES batch (batch_id)
) ;
```

Batch:

```
CREATE TABLE batch (
  batch_id int NOT NULL,
  course varchar(255) NOT NULL,
```

```
  course_type varchar(255) NOT NULL,
  PRIMARY KEY (batch_id)
) ;
```

Class:

```
CREATE TABLE class (
  class_id int NOT NULL,
  topic varchar(255) NOT NULL,
  session int NOT NULL,
  batch int NOT NULL,
  mentor_id int NOT NULL,
  PRIMARY KEY (class_id),
FOREIGN KEY (batch) REFERENCES batch (batch_id),
FOREIGN KEY (mentor_id) REFERENCES mentor (mentor_id)
);
```

Task:

```
CREATE TABLE task (
  task_id int NOT NULL,
  task_name varchar(255) NOT NULL,
  description varchar(255) NOT NULL,
  deadline varchar(100) NOT NULL,
  topic_id int NOT NULL,
  PRIMARY KEY (task_id),
```

```
  FOREIGN KEY (topic_id) REFERENCES class (class_id)
);
```

Codekata:

```
CREATE TABLE codekata (
  user_id int NOT NULL,
  problem_solved int NOT NULL,
  PRIMARY KEY (user_id),
  FOREIGN KEY (user_id) REFERENCES student (student_id)
);
```

Attendance:

```
CREATE TABLE attendance (
  attendance_id int NOT NULL,
  status enum('present','absent') NOT NULL,
  user_id int NOT NULL,
  topic_id int NOT NULL,
  PRIMARY KEY (attendance_id),
  FOREIGN KEY (user_id) REFERENCES users (id),
  FOREIGN KEY (topic_id) REFERENCES class (class_id)
);
```