Guvi zen class DB design

Introduction

This database manages students and mentors, tracking their details, tasks, queries, and class schedules.

Entities :

1. **Students**

   Details: ID, Name, Age, Course, Email, Joining Date, Duration

2. **Mentors**

   Details: ID, Name, Age, Email, Joining Date, Subject Taught

3. **TaskSubmissions**

   Details: Submission ID, Student Name, Course, Task, Marks, Submission Date

4. **MentorQueries**

   Details: Query ID, Student Name, Query, Date Raised, Status, Mentor Assigned

5. **MentorClassTimings**

   Details: Class Timing ID, Course, Timing, Date

6. **StudentMentors**

   Details: Student ID, Mentor ID (links students to their mentors based on course)

Relationships :

- **Students and Mentors:** Students are matched with mentors based on their course and the mentor's subject.
- **Tasks:** Each student can have multiple tasks recorded.
- **Queries:** Each query is linked to a student and handled by a mentor.
- **Class Timings:** Each mentor's classes are scheduled with timings and dates.

```
create database guvizen;
use guvizen;

CREATE TABLE Students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    course VARCHAR(50),
    email VARCHAR(100),
    date_of_joining DATE,
    duration VARCHAR(20)
);

INSERT INTO Students (name, age, course, email, date_of_joining, duration)
VALUES
('John Doe', 21, 'Full Stack Development', 'johndoe@example.com', '2023-04-12', '6 months'),
('Jane Smith', 22, 'Data Science', 'janesmith@example.com', '2022-11-10', '12 months'),
('Mike Brown', 20, 'Cyber Security', 'mikebrown@example.com', '2023-01-15', '8 months'),
('Lisa White', 24, 'Cloud Computing', 'lisawhite@example.com', '2023-05-20', '9 months'),
```

('Dave Green', 23, 'Artificial Intelligence', 'davegreen@example.com', '2022-12-01', '10 months'),

('Sarah Blue', 19, 'Web Development', 'sarahblue@example.com', '2023-06-01', '6 months'),

('Chris Black', 25, 'Machine Learning', 'chrisblack@example.com', '2023-02-18', '7 months'),

('Anna Grey', 22, 'Mobile App Development', 'annagrey@example.com', '2023-07-10', '6 months'),

('Paul Brown', 23, 'Blockchain Technology', 'paulbrown@example.com', '2023-03-12', '5 months'),

('Emma King', 20, 'Game Development', 'emmaking@example.com', '2023-01-08', '8 months');

```
CREATE TABLE Tasks (
    task_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    course VARCHAR(50),
    task_description VARCHAR(255),
    FOREIGN KEY (student_id) REFERENCES Students(student_id)
);
```

```
INSERT INTO Tasks (student_id, course, task_description)
VALUES
(1, 'Full Stack Development', 'Design a DB model for Guvi Zen'),
(2, 'Data Science', 'Analyze sales data with Python'),
(3, 'Cyber Security', 'Create a network security plan'),
(4, 'Cloud Computing', 'Set up a virtual server on AWS'),
(5, 'Artificial Intelligence', 'Develop a chatbot with NLP'),
(6, 'Web Development', 'Build a responsive website'),
(7, 'Machine Learning', 'Implement a recommendation system'),
```

```
(8, 'Mobile App Development', 'Create a mobile app using Flutter'),
(9, 'Blockchain Technology', 'Develop a smart contract'),
(10, 'Game Development', 'Design a 2D game with Unity');

CREATE TABLE Queries (
    query_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    query VARCHAR(255),
    date_of_query_raised DATE,
    status VARCHAR(20),
    assigned_to VARCHAR(50),
    FOREIGN KEY (student_id) REFERENCES Students(student_id)
);

INSERT INTO Queries (student_id, query, date_of_query_raised, status, assigned_to)
VALUES
(1, 'Doubt about rendering data in MySQL', '2024-06-12', 'Solved', 'Sathish'),
(2, 'Issue with data visualization', '2024-05-20', 'Pending', 'Priya'),
(3, 'Clarification on encryption methods', '2024-05-18', 'Solved', 'Ravi'),
(4, 'Help needed for AWS configuration', '2024-06-05', 'Solved', 'Kumar'),
(5, 'NLP model accuracy issues', '2024-07-01', 'Pending', 'Anjali'),
(6, 'Problem with CSS layout', '2024-07-10', 'Solved', 'Deepak'),
(7, 'Questions on clustering algorithms', '2024-06-28', 'Pending', 'Raj'),
(8, 'Issue with app deployment', '2024-07-12', 'Solved', 'Sunita'),
(9, 'Trouble with blockchain integration', '2024-07-20', 'Pending', 'Manoj'),
(10, 'Error in Unity project build', '2024-07-15', 'Solved', 'Neha');
```

```
CREATE TABLE ClassTimings (
    class_timing_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    time_of_class TIME,
    date DATE,
    FOREIGN KEY (student_id) REFERENCES Students(student_id)
);

INSERT INTO ClassTimings (student_id, time_of_class, date)
VALUES
(1, '21:00:00', '2024-07-12'),
(2, '11:00:00', '2024-06-18'),
(3, '14:00:00', '2024-07-20'),
(4, '16:00:00', '2024-07-05'),
(5, '18:00:00', '2024-06-28'),
(6, '10:00:00', '2024-07-12'),
(7, '15:00:00', '2024-06-30'),
(8, '20:00:00', '2024-07-14'),
(9, '13:00:00', '2024-06-25'),
(10, '19:00:00', '2024-07-22');
```

```
CREATE TABLE Feedbacks (
    feedback_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    feedback TEXT,
    date DATE,
    FOREIGN KEY (student_id) REFERENCES Students(student_id)
);

INSERT INTO Feedbacks (student_id, feedback, date)
VALUES
(1, 'Excellent flow of covering topics', '2024-07-11'),
(2, 'Good hands-on practice', '2024-06-30'),
(3, 'More examples needed', '2024-07-05'),
(4, 'Great support from mentors', '2024-07-02'),
(5, 'Course pacing is fast', '2024-06-20'),
(6, 'Enjoyed the interactive sessions', '2024-07-09'),
(7, 'Detailed explanations appreciated', '2024-07-13'),
(8, 'Loved the practical projects', '2024-07-19'),
(9, 'More focus on advanced topics required', '2024-07-07'),
(10, 'Satisfied with the course content', '2024-07-15');
```

```
CREATE TABLE PerformanceReports (
    report_id INT AUTO_INCREMENT PRIMARY KEY,
    student_id INT,
    task_percentage DECIMAL(5,2),
    FOREIGN KEY (student_id) REFERENCES Students(student_id)
);

INSERT INTO PerformanceReports (student_id, task_percentage)
VALUES
(1, 80.00),
(2, 90.00),
(3, 85.00),
(4, 78.00),
(5, 92.00),
(6, 87.00),
(7, 88.00),
(8, 81.00),
(9, 70.00),
(10, 95.00);
```

Query 1: Get student details along with their course and duration,

```
SELECT s.student_id, s.name, s.course, s.duration

FROM Students s;
```

1	John Doe	Full Stack Development	6 months
2	Jane Smith	Data Science	12 months
3	Mike Brown	Cyber Security	8 months
4	Lisa White	Cloud Computing	9 months
5	Dave Green	Artificial Intelligence	10 months
6	Sarah Blue	Web Development	6 months
7	Chris Black	Machine Learning	7 months
8	Anna Grey	Mobile App Development	6 months
9	Paul Brown	Blockchain Technology	5 months
10	Emma King	Game Development	8 months

Query 2: Get tasks assigned to each student

```
SELECT s.name, t.task_description

FROM Students s

JOIN Tasks t ON s.student_id = t.student_id;
```

John Doe	Design a DB model for Guvi Zen
Jane Smith	Analyze sales data with Python
Mike Brown	Create a network security plan

Lisa White	Set up a virtual server on AWS
Dave Green	Develop a chatbot with NLP
Sarah Blue	Build a responsive website
Chris Black	Implement a recommendation system
Anna Grey	Create a mobile app using Flutter
Paul Brown	Develop a smart contract
Emma King	Design a 2D game with Unity

Query 3: Get class timings for each student

```
SELECT s.name, ct.time_of_class, ct.date

FROM Students s

JOIN ClassTimings ct ON s.student_id = ct.student_id;
```

John Doe	21:00:00	12-07-2024
Jane Smith	11:00:00	18-06-2024
Mike Brown	14:00:00	20-07-2024
Lisa White	16:00:00	05-07-2024
Dave Green	18:00:00	28-06-2024
Sarah Blue	10:00:00	12-07-2024
Chris Black	15:00:00	30-06-2024
Anna Grey	20:00:00	14-07-2024
Paul Brown	13:00:00	25-06-2024
Emma King	19:00:00	22-07-2024

Query 4: Get students who have completed their tasks with a percentage above 85

```
SELECT s.name, pr.task_percentage

FROM Students s
```

JOIN PerformanceReports pr ON s.student_id = pr.student_id

WHERE pr.task_percentage > 85;

Jane Smith	90
Dave Green	92
Sarah Blue	87
Chris Black	88
Emma King	95

Query 5: Get the average task percentage for each course

SELECT s.course, AVG(pr.task_percentage) as avg_task_percentage

FROM Students s

JOIN PerformanceReports pr ON s.student_id = pr.student_id

GROUP BY s.course;

Full Stack Development	80
Data Science	90
Cyber Security	85
Cloud Computing	78
Artificial Intelligence	92
Web Development	87
Machine Learning	88
Mobile App Development	81
Blockchain Technology	70
Game Development	95

Query 6: Get the count of students enrolled in each course

```
SELECT course, COUNT(*) as student_count
FROM Students
GROUP BY course;
```

Full Stack Development	1
Data Science	1
Cyber Security	1
Cloud Computing	1
Artificial Intelligence	1
Web Development	1
Machine Learning	1
Mobile App Development	1
Block chain Technology	1
Game Development	1

Query 7: Get Students with Pending Queries and Their Assigned Mentors

```
SELECT s.student_id, s.name, q.query, q.assigned_to
FROM Students s
JOIN Queries q ON s.student_id = q.student_id
WHERE q.status = 'Pending';
```

2	Jane Smith	Issue with data visualization	Priya
5	Dave Green	NLP model accuracy issues	Anjali
7	Chris Black	Questions on clustering algorithms	Raj
9	Paul Brown	Trouble with blockchain integration	Manoj

Mentors DB

```
CREATE TABLE Mentors (
    mentor_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    email VARCHAR(100),
    date_of_joining DATE,
    teaching VARCHAR(50)
);

INSERT INTO Mentors (name, age, email, date_of_joining, teaching)
VALUES
('Jane', 31, 'jane@example.com', '2022-04-12', 'Full Stack Development'),
('Alex', 35, 'alex@example.com', '2022-01-10', 'Data Science'),
('Ravi', 40, 'ravi@example.com', '2021-11-25', 'Cyber Security'),
('Anita', 29, 'anita@example.com', '2023-02-18', 'Cloud Computing'),
('Kumar', 33, 'kumar@example.com', '2022-07-15', 'Artificial Intelligence'),
('Priya', 37, 'priya@example.com', '2021-12-05', 'Web Development'),
('Raj', 38, 'raj@example.com', '2022-08-20', 'Machine Learning'),
('Sunita', 32, 'sunita@example.com', '2023-03-30', 'Mobile App Development'),
('Manoj', 34, 'manoj@example.com', '2022-05-18', 'Blockchain Technology'),
('Neha', 28, 'neha@example.com', '2023-01-14', 'Game Development');
```

```
CREATE TABLE TaskSubmissions (
    submission_id INT AUTO_INCREMENT PRIMARY KEY,
    student_name VARCHAR(50),
    course VARCHAR(50),
    task VARCHAR(255),
    marks DECIMAL(5,2),
    date_of_submission DATE
);

INSERT INTO TaskSubmissions (student_name, course, task, marks, date_of_submission)
VALUES
('John Doe', 'Full Stack Development', 'Design a DB model for Guvi Zen', 75, '2024-06-12'),
('Jane Smith', 'Data Science', 'Analyze sales data with Python', 85, '2024-06-15'),
('Mike Brown', 'Cyber Security', 'Create a network security plan', 90, '2024-06-20'),
('Lisa White', 'Cloud Computing', 'Set up a virtual server on AWS', 80, '2024-07-01'),
('Dave Green', 'Artificial Intelligence', 'Develop a chatbot with NLP', 88, '2024-07-03'),
('Sarah Blue', 'Web Development', 'Build a responsive website', 92, '2024-06-25'),
('Chris Black', 'Machine Learning', 'Implement a recommendation system', 84, '2024-07-05'),
('Anna Grey', 'Mobile App Development', 'Create a mobile app using Flutter', 87, '2024-06-30'),
('Paul Brown', 'Blockchain Technology', 'Develop a smart contract', 78, '2024-07-10'),
('Emma King', 'Game Development', 'Design a 2D game with Unity', 95, '2024-07-15');
```

```
CREATE TABLE MentorQueries (
    query_id INT AUTO_INCREMENT PRIMARY KEY,
    student_name VARCHAR(50),
    query VARCHAR(255),
    date_of_query_raised DATE,
    status VARCHAR(20),
    assigned_to VARCHAR(50)
);


INSERT INTO MentorQueries (student_name, query, date_of_query_raised, status, assigned_to)
VALUES
('John Doe', 'Doubt about rendering data in MySQL', '2024-06-12', 'Solved', 'Sathish'),
('Jane Smith', 'Issue with data visualization', '2024-05-20', 'Pending', 'Priya'),
('Mike Brown', 'Clarification on encryption methods', '2024-05-18', 'Solved', 'Ravi'),
('Lisa White', 'Help needed for AWS configuration', '2024-06-05', 'Solved', 'Kumar'),
('Dave Green', 'NLP model accuracy issues', '2024-07-01', 'Pending', 'Anjali'),
('Sarah Blue', 'Problem with CSS layout', '2024-07-10', 'Solved', 'Deepak'),
('Chris Black', 'Questions on clustering algorithms', '2024-06-28', 'Pending', 'Raj'),
('Anna Grey', 'Issue with app deployment', '2024-07-12', 'Solved', 'Sunita'),
('Paul Brown', 'Trouble with blockchain integration', '2024-07-20', 'Pending', 'Manoj'),
('Emma King', 'Error in Unity project build', '2024-07-15', 'Solved', 'Neha');
```

```
CREATE TABLE MentorClassTimings (
    class_timing_id INT AUTO_INCREMENT PRIMARY KEY,
    course VARCHAR(50),
    timing TIME,
    date DATE
);

INSERT INTO MentorClassTimings (course, timing, date)
VALUES
('Full Stack Development', '21:00:00', '2024-07-12'),
('Data Science', '11:00:00', '2024-06-18'),
('Cyber Security', '14:00:00', '2024-07-20'),
('Cloud Computing', '16:00:00', '2024-07-05'),
('Artificial Intelligence', '18:00:00', '2024-06-28'),
('Web Development', '10:00:00', '2024-07-12'),
('Machine Learning', '15:00:00', '2024-06-30'),
('Mobile App Development', '20:00:00', '2024-07-14'),
('Blockchain Technology', '13:00:00', '2024-06-25'),
('Game Development', '19:00:00', '2024-07-22');
```

Find the Mentors with the Oldest Age

```
SELECT name, age, email
FROM Mentors
ORDER BY age DESC
LIMIT 3;
```

Ravi	40	ravi@example.com
Raj	38	raj@example.com
Priya	37	priya@example.com

List Mentors with Their Email and Teaching Field in Alphabetical Order,

```
SELECT name, email, teaching AS course
FROM Mentors
ORDER BY name ASC;
```

Alex	alex@example.com	Data Science
Anita	anita@example.com	Cloud Computing
Jane	jane@example.com	Full Stack Development
Kumar	kumar@example.com	Artificial Intelligence
Manoj	manoj@example.com	Blockchain Technology
Neha	neha@example.com	Game Development
Priya	priya@example.com	Web Development
Raj	raj@example.com	Machine Learning
Ravi	ravi@example.com	Cyber Security
Sunita	sunita@example.com	Mobile App Development

Pending Queries,

```
SELECT student_name, query, date_of_query_raised, assigned_to
FROM MentorQueries
WHERE status = 'Pending';
```

Jane Smith	Issue with data visualization	20-05-2024	Priya
Dave Green	NLP model accuracy issues	01-07-2024	Anjali
Chris Black	Questions on clustering algorithms	28-06-2024	Raj
Paul Brown	Trouble with blockchain integration	20-07-2024	Manoj

Queries Raised Before a Certain Date,

```
SELECT student_name, query, date_of_query_raised, status, assigned_to
FROM MentorQueries
WHERE date_of_query_raised < '2024-07-01';
```

John Doe	Doubt about rendering data in MySQL	12-06-2024	Solved	Sathish
Jane Smith	Issue with data visualization	20-05-2024	Pending	Priya
Mike Brown	Clarification on encryption methods	18-05-2024	Solved	Ravi
Lisa White	Help needed for AWS configuration	05-06-2024	Solved	Kumar
Chris Black	Questions on clustering algorithms	28-06-2024	Pending	Raj

Query to Show Students and Their Mentors

```
SELECT s.name AS student_name, m.name AS mentor_name, s.course
FROM StudentMentors sm
JOIN Students s ON sm.student_id = s.student_id
JOIN Mentors m ON sm.mentor_id = m.mentor_id;
```

student name	mentor name	course
John Doe	Jane	Full Stack Development
Jane Smith	Alex	Data Science
Mike Brown	Ravi	Cyber Security
Lisa White	Anita	Cloud Computing
Dave Green	Kumar	Artificial Intelligence
Sarah Blue	Priya	Web Development
Chris Black	Raj	Machine Learning
Anna Grey	Sunita	Mobile App Development
Paul Brown	Manoj	Blockchain Technology
Emma King	Neha	Game Development