

```

-- Create the database

CREATE DATABASE guvi_zen_task;

use guvi_zen_task;

-- Create the user table

CREATE TABLE Users (
    user_id INT PRIMARY KEY,
    name VARCHAR(50),
    email VARCHAR(100),
    phone VARCHAR(15),
    joined_date DATE
);

-- Create the codekata table

CREATE TABLE CodeKata (
    user_id INT,
    problems_solved INT,
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the attendance table

CREATE TABLE Attendance (
    attendance_id INT PRIMARY KEY,
    user_id INT,
    date DATE,
    status ENUM('Present', 'Absent'),
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the topics table

CREATE TABLE Topics (
    topic_id INT PRIMARY KEY,
    name VARCHAR(100),
    date_covered DATE
);

-- Create the tasks table

```

```

CREATE TABLE Tasks (
    task_id INT PRIMARY KEY,
    topic_id INT,
    user_id INT,
    submission_date DATE,
    status ENUM('Submitted', 'Not Submitted'),
    FOREIGN KEY (topic_id) REFERENCES Topics(topic_id),
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Create the companydrives table

CREATE TABLE CompanyDrives (
    drive_id INT PRIMARY KEY,
    company_name VARCHAR(100),
    drive_date DATE
);

-- Create the mentors table

CREATE TABLE Mentors (
    mentor_id INT PRIMARY KEY,
    name VARCHAR(50)
);

-- Create the mentees table

CREATE TABLE Mentees (
    mentor_id INT,
    user_id INT,
    FOREIGN KEY (mentor_id) REFERENCES Mentors(mentor_id),
    FOREIGN KEY (user_id) REFERENCES Users(user_id)
);

-- Users
INSERT INTO Users (user_id, name, email, phone, joined_date)
VALUES (1, 'Alice', 'alice@example.com', '1234567890', '2020-08-01'),
       (2, 'Bob', 'bob@example.com', '1234567891', '2020-09-01'),
       (3, 'Charlie', 'charlie@example.com', '1234567892', '2020-09-15');

SELECT * FROM USERS;

```

```

-- CodeKata
INSERT INTO CodeKata (user_id, problems_solved)
VALUES (1, 120), (2, 85), (3, 150);

SELECT * FROM CODEKATA;


-- Attendance

INSERT INTO Attendance (attendance_id, user_id, date, status)
VALUES (1, 1, '2020-10-16', 'Present'),
       (2, 2, '2020-10-16', 'Absent'),
       (3, 3, '2020-10-20', 'Present'),
       (4, 1, '2020-10-20', 'Absent'),
       (5, 2, '2020-10-25', 'Absent');

SELECT * FROM ATTENDANCE;


-- Topics
INSERT INTO Topics (topic_id, name, date_covered)
VALUES (1, 'SQL Basics', '2020-10-05'),
       (2, 'HTML & CSS', '2020-10-10'),
       (3, 'JavaScript', '2020-10-15'),
       (4, 'Python Intro', '2020-10-20');

SELECT * FROM TOPICS;


-- Tasks
INSERT INTO Tasks (task_id, topic_id, user_id, submission_date, status)
VALUES (1, 1, 1, '2020-10-06', 'Submitted'),
       (2, 2, 1, '2020-10-11', 'Not Submitted'),
       (3, 3, 2, '2020-10-16', 'Submitted'),
       (4, 4, 3, '2020-10-21', 'Not Submitted');

SELECT * FROM TASKS;

```

```
-- Company Drives
INSERT INTO CompanyDrives (drive_id, company_name, drive_date)
VALUES (1, 'Google', '2020-10-16'),
       (2, 'Amazon', '2020-10-20'),
       (3, 'Facebook', '2020-10-25');
```

```
SELECT * FROM COMPANY_DRIVES;
```

```
-- Mentors
INSERT INTO Mentors (mentor_id, name)
VALUES (1, 'Emily Green'), (2, 'Mark Johnson');
```

```
SELECT * FROM MENTORS;
```

```
-- Mentees
INSERT INTO Mentees (mentor_id, user_id)
VALUES (1, 1), (1, 2), (2, 3);
```

```
SELECT * FROM MENTEES;
```

-- 1.Find all the topics and tasks which are thought in the month of October

```
SELECT Topics.name AS Topic, Tasks.task_id AS Task
FROM Topics
LEFT JOIN Tasks ON Topics.topic_id = Tasks.topic_id
WHERE MONTH(Topics.date_covered) = 10;
```

-- 2.Find all the company drives which appeared between 15 oct-2020 and 31-oct-2020

```
SELECT * FROM CompanyDrives
WHERE drive_date BETWEEN '2020-10-15' AND '2020-10-31';
```

-- 3.Find all the company drives and students who are appeared for the placement.

```
SELECT CompanyDrives.company_name, Users.name AS Student
FROM CompanyDrives
INNER JOIN Attendance ON Attendance.date = CompanyDrives.drive_date AND
Attendance.status = 'Present'
INNER JOIN Users ON Users.user_id = Attendance.user_id;
```

-- 4.Find the number of problems solved by the user in codekata

```
SELECT Users.name, CodeKata.problems_solved
FROM Users
INNER JOIN CodeKata ON Users.user_id = CodeKata.user_id;
```

-- 5.Find all the mentors with who has the mentee's count more than 15

```
SELECT Mentors.name, COUNT(Mentees.user_id) AS Mentee_Count
FROM Mentors
INNER JOIN Mentees ON Mentors.mentor_id = Mentees.mentor_id
GROUP BY Mentors.mentor_id
HAVING Mentee_Count > 15;
```

-- 6.Find the number of users who are absent and task is not submitted between 15 oct-2020 and 31-oct-2020

```
SELECT COUNT(DISTINCT Users.user_id) AS Absent_NoTask
FROM Users
INNER JOIN Attendance ON Users.user_id = Attendance.user_id
LEFT JOIN Tasks ON Users.user_id = Tasks.user_id
    AND Tasks.submission_date BETWEEN '2020-10-15' AND '2020-10-31'
WHERE Attendance.date BETWEEN '2020-10-15' AND '2020-10-31'
    AND Attendance.status = 'Absent'
    AND (Tasks.status IS NULL OR Tasks.status = 'Not Submitted');
```