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

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

5. Mentor class timings Details: Class Timing ID, Course, Timing, Date

6. Student mentors 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 schema Guvi;

use Guvi;

create table students(

student\_id int auto\_increment primary key,

name varchar(255),

age int,

course varchar(255),

email varchar(255),

date\_of\_joining date,

duration varchar(20)

);

insert into students (name, age, course, email, date\_of\_joining, duration) values

('Karthikeyan', 25, 'MERN stack development', 'karthi@guvi.com', '2024-04-21', '6 months'),

('Gokul', 25, 'Business Intelligence', 'gokul@guvi.com', '2024-02-21', '3 months'),

('Issac', 25, 'MERN stack development', 'issac@guvi.com', '2023-11-18', '3 months'),

('Abidh', 25, 'Python developer', 'abidh@guvi.com', '2024-05-10', '6 months'),

('Divyadharshini', 22, 'UI/UX Designer', 'divi@guvi.com', '2024-08-21', '5 months'),

('Kavya', 22, 'Data analytics', 'kavya@guvi.com', '2024-07-20', '3 months'),

('Sabari', 25, 'Data Science', 'sabari@guvi.com', '2024-05-05', '4 months'),

('selsia', 25, 'PHP developer', 'selsia@guvi.com', '2024-03-08', '6 months'),

('Dharshini', 24, 'Java developer', 'dharshini@guvi.com', '2023-10-28', '8 months'),

('Vinitha', 25, 'Angular developer', 'vinitha@guvi.com', '2024-06-15', '6 months');

create table tasks(

task\_id int auto\_increment primary key,

student\_id int,

course varchar(50),

task\_description varchar(100),

foreign key (student\_id) references Students(student\_id)

);

insert into tasks (student\_id, course, task\_description) values

(1, 'MERN stack development', 'Design a DB model for Guvi Zen'),

(2, 'Business Intelligence', 'Market Basket Analysis'),

(3, 'MERN stack development', 'Design a DB model for Guvi Zen'),

(4, 'Python developer', 'Password Generator'),

(5, 'UI/UX Designer', 'Design a UI for ecommerce'),

(6, 'Data analytics', 'Sales Forecasting Analysis'),

(7, 'Data Science', 'Hospital Treatment Pricing Prediction'),

(8, 'PHP developer', 'Leave Management System in PHP'),

(9, 'Java developer', 'Currency Converter'),

(10, 'Angular developer', 'Blog App');

create table queries(

query\_id int auto\_increment primary key,

student\_id int,

query varchar(255),

status varchar(255),

date\_of\_query\_raised DATE,

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 in sql', '2024-05-21', 'Solved', 'Vikram'),

(2, 'Issue with data visualization', '2024-03-21', 'Pending', 'Amar'),

(3, 'Doubt in sql', '2023-12-18', 'Solved', 'Bijoy'),

(4, 'Issue password generating', '2024-06-10', 'Pending', 'Santhanam'),

(5, 'Designing issue in Adobe', '2024-08-20', 'Solved', 'Dilli'),

(6, 'Issue with data visualization', '2024-07-28', 'Pending', 'Parthiban'),

(7, 'Issue in pricing prediction', '2024-05-10', 'Solved', 'Vikram'),

(8, 'Doubt in leave management system data', '2024-04-21', 'Pending', 'Amar'),

(9, 'Api fetching issue', '2023-11-21', 'solved', 'Santhanam'),

(10, 'Doubt on creating a blog app', '2024-07-28', 'Pending', 'Bijoy');

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),

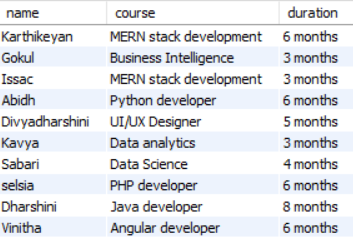
(8, 81.00),

(9, 70.00),

(10, 95.00);

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

select name, course, duration from Students;

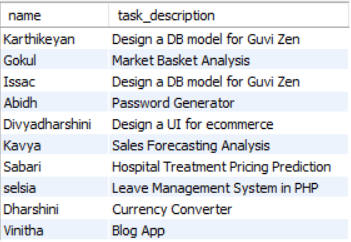


Query 2: Get tasks assigned to each student

SELECT s.name, t.task\_description

FROM Students s

JOIN Tasks t ON s.student\_id = t.student\_id;

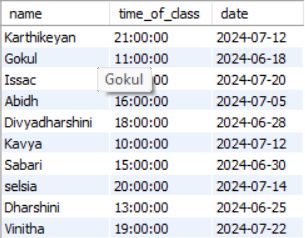


Query 3: Get class timings for each student

SELECT s.name, ct.time\_of\_class, ct.date

FROM Students s

JOIN ClassTimings ct ON s.student\_id = ct.student\_id;



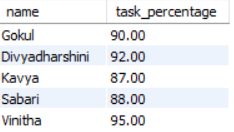
Query 4: Get students who have completed their tasks with a percentage above 85

SELECT s.name, pr.task\_percentage

FROM Students s

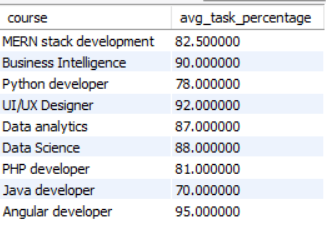
JOIN PerformanceReports pr ON s.student\_id = pr.student\_id

WHERE pr.task\_percentage > 85;



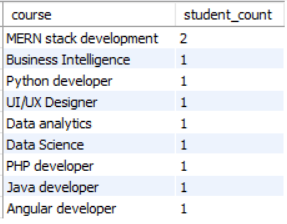
Query 5: Get the average task percentage for each course

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;



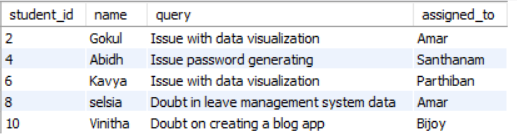
Query 6: Get the count of students enrolled in each course

SELECT course, COUNT(\*) as student\_count FROM Students GROUP BY course;



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';



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

('Vikram', 31, 'vikram@guvi.com', '2022-04-12', 'MERN stack development'),

('Amar', 35, 'amar@guvi.com', '2022-01-10', 'Business Intelligence'),

('Santhanam', 40, 'santhanam@guvi.com', '2021-11-25', 'Python developer'),

('Bijoy', 29, 'bijoy@guvi.com', '2023-02-18', 'MERN stack development'),

('Parthiban', 33, 'parthiban@guvi.com', '2022-07-15', 'Data analytics'),

('Dilli', 37, 'dilli@guvi.com', '2021-12-05', 'UI/UX designer'),

('Surya', 38, 'surya@guvi.com', '2022-08-20', 'Machine Learning'),

('Sunita', 32, 'sunita@guvi.com', '2023-03-30', 'Mobile App Development'),

('Manoj', 34, 'manoj@guvi.com', '2022-05-18', 'Blockchain Technology'),

('Neha', 28, 'neha@guvi.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

('Karthikeyan', 'MERN stack development', 'Design a DB model for Guvi Zen', 75, '2024-06-12'),

('Gokul', 'Business Intelligence', 'Market Basket Analysis', 85, '2024-06-15'),

('Issac', 'MERN stack development', 'Design a DB model for Guvi Zen', 90, '2024-06-20'),

('Abidh', 'Python developer', 'Password Generator', 80, '2024-07-01'),

('Divyadharshini', 'UI/UX Designer', 'Design a UI for ecommerce',88, '2024-07-03'),

('Kavya', 'Data analytics', 'Sales Forecasting Analysis',92, '2024-06-25'),

('Sabari', 'Data Science', 'Hospital Treatment Pricing Prediction',84, '2024-07-05'),

('selsia', 'PHP developer', 'Leave Management System in PHP', 87, '2024-06-30'),

('Dharshini', 'Java developer', 'Currency Converter', 78, '2024-07-10'),

('Vinitha', 'Angular developer', 'Blog App',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

(1, 'Doubt in sql', '2024-05-21', 'Solved', 'Vikram'),

(2, 'Issue with data visualization', '2024-03-21', 'Pending', 'Amar'),

(3, 'Doubt in sql', '2023-12-18', 'Solved', 'Bijoy'),

(4, 'Issue password generating', '2024-06-10', 'Pending', 'Santhanam'),

(5, 'Desigining issue in Adobe', '2024-08-20', 'Solved', 'Dilli'),

(6, 'Issue with data visualization', '2024-07-28', 'Pending', 'Parthiban'),

(7, 'Issue in pricing prediction', '2024-05-10', 'Solved', 'Vikram'),

(8, 'Doubt in leave management system data', '2024-04-21', 'Pending', 'Amar'),

(9, 'Api fetching issue', '2023-11-21', 'solved', 'Santhanam'),

(10, 'Doubt on creating a blog app', '2024-07-28', 'Pending', 'Bijoy');

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

('MERN stack development', '21:00:00', '2024-07-12'),

('Data Science', '11:00:00', '2024-06-18'),

('Data analytics', '14:00:00', '2024-07-20'),

('Cloud Computing', '16:00:00', '2024-07-05'),

('PHP developer', '18:00:00', '2024-06-28'),

('Business Intelligence', '10:00:00', '2024-07-12'),

('Python developer', '15:00:00', '2024-06-30'),

('UI/UX designer', '20:00:00', '2024-07-14'),

('Java developer', '13:00:00', '2024-06-25'),

('Angular developer', '19:00:00', '2024-07-22');

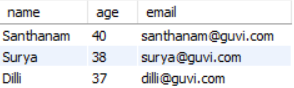
Query: - Find the Mentors with the Oldest Age

SELECT name, age, email

FROM Mentors

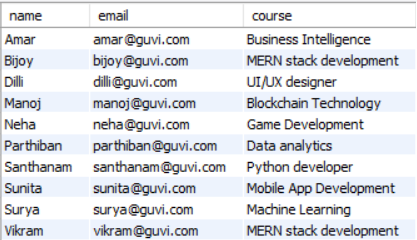
ORDER BY age DESC

LIMIT 3;



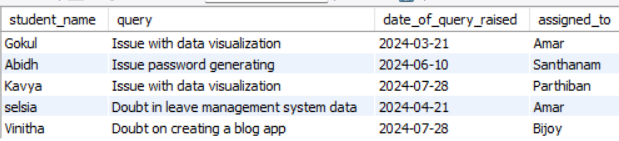
Query:-List Mentors with Their Email and Teaching Field in Alphabetical Order

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



Query:- Pending Queries

SELECT student\_name, query, date\_of\_query\_raised, assigned\_to FROM MentorQueries WHERE status = 'Pending';



Query:- 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';

