GUVI ZEN CLASS DB DESIGN QUERIES

1.	**students Table**	`CREATE table students (`
		`student_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,`
		`name VARCHAR(25) NOT NULL,`
		`email VARCHAR(25) NOT NULL,`
		`phone_number BIGINT NOT NULL,`
		`course_id int NOT NULL,`
		`batch_id int NOT NULL,`
		`FOREIGN KEY (course_id) REFERENCES`
		`courses(course_id),`
		`FOREIGN KEY (batch_id) REFERENCES batches(batch_id)`
		`);`

2.	**courses Table**	`CREATE TABLE courses (`
		`course_id INT PRIMARY KEY NOT NULL`
		`AUTO_INCREMENT,`
		`course_name VARCHAR(100) NOT NULL,`
		`course_description VARCHAR(250) NOT NULL,`
		`enrolement_id INT NOT NULL,`
		`student_id INT NOT NULL,`
		`FOREIGN KEY (enrolement_id) REFERENCES`
		`enrolements(enrolement_id),`
		`FOREIGN KEY (student_id) REFERENCES`
		`students(student_id)`
		`);`

3.	**enrolements Table**	CREATE TABLE enrolements(enrolement_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, course_id INT NOT NULL, student_id INT NOT NULL, batch_mode_id INT NOT NULL, batch_language_id INT NOT NULL, FOREIGN KEY (course_id) REFERENCES courses(course_id), FOREIGN KEY (student_id) REFERENCES students(student_id), FOREIGN KEY (batch_mode_id) REFERENCES batch_modes(batch_mode_id), FOREIGN KEY (batch_language_id) REFERENCES languages(batch_language_id));
4.	**batches Table**	CREATE TABLE batches (batch_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, batch_name VARCHAR(100) NOT NULL, batch_description VARCHAR(250) NOT NULL, batch_language_id INT NOT NULL, batch_mode_id INT NOT NULL, student_id INT NOT NULL, mentor_id INT NOT NULL, FOREIGN KEY (student_id) REFERENCES students(student_id), FOREIGN KEY (mentor_id) REFERENCES mentors(mentor_id), FOREIGN KEY (batch_mode_id) REFERENCES batch_modes(batch_mode_id), FOREIGN KEY (batch_language_id) REFERENCES languages(batch_language_id));

5.	**languages Table**	`CREATE TABLE languages (`
		`  language_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,`
		`  language_name VARCHAR(30)`
		`);`

6.	**batch_modes Table**	`CREATE TABLE batch_modes (`
		`  batch_mode_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,`
		`  batch_mode VARCHAR(30)`
		`);`

7.	**session Table**	`CREATE TABLE sessions (`
		`  session_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,`
		`  title VARCHAR(30) NOT NULL,`
		`  description VARCHAR(250) NOT NULL,`
		`  pre_read VARCHAR(250),`
		`  task_id INT NOT NULL,`
		`  feedback_id INT NOT NULL,`
		`  session_link_id INT NOT NULL,`
		`  FOREIGN KEY (session_link_id) REFERENCES sessions_links(session_link_id),`
		`  FOREIGN KEY (task_id) REFERENCES tasks(task_id),`
		`  FOREIGN KEY (feedback_id) REFERENCES feedbacks(feedback_id)`
		`);`

8.	**session_links table**	`CREATE TABLE session_links (`
		`  session_link_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,`
		`  session_link VARCHAR(300) NOT NULL,`
		`  session_recording VARCHAR(300),`
		`  batch_id INT NOT NULL,`
		`  session_id INT NOT NULL,`
		`  FOREIGN KEY (batch_id) REFERENCES batches(batch_id),`
		`  FOREIGN KEY (session_id) REFERENCES sessions(session_id)`
		`);`

9.	**tasks table**	CREATE TABLE tasks (task_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, task_link_id INT NOT NULL, submission_id INT NOT NULL, comments VARCHAR(300), FOREIGN KEY (task_link_id) REFERENCES task_links(task_link_id), FOREIGN KEY (submission_id) REFERENCES submissions(submission_id));
10.	**task_submissions table**	CREATE TABLE task_submissions (submission_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, frontend_source_code VARCHAR(300), frontend_deployed_url VARCHAR(300), backend_source_code VARCHAR(300), backend_deployed_url VARCHAR(300), session_id INT, student_id INT, FOREIGN KEY (session_id) REFERENCES sessions(session_id), FOREIGN KEY (student_id) REFERENCES students(student_id));
11.	**task_links table**	CREATE TABLE task_links (task_link_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, task_link VARCHAR(300) NOT NULL);

12.	**feedbacks table**	CREATE TABLE feedbacks (feedback_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, session_rating INT NOT NULL, session_feedback VARCHAR(250) NOT NULL, mentor_rating INT NOT NULL, mentor_feedback VARCHAR(250) NOT NULL, key_takeaways VARCHAR(250) NOT NULL, session_id INT, student_id INT, FOREIGN KEY (session_id) REFERENCES sessions(session_id), FOREIGN KEY (student_id) REFERENCES students(student_id));
13.	**queries table**	CREATE TABLE queries (query_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, category VARCHAR(50) NOT NULL, prefered_communication_lanugage INT, query_title VARCHAR(100) NOT NULL, query_description VARCHAR(250) NOT NULL, available_from TIMESTAMP, available_to TIMESTAMP, student_id INT NOT NULL, FOREIGN KEY (prefered_communication_language) REFERENCES languages(language_id), FOREIGN KEY (student_id) REFERENCES students(student_id));

14.	**portfolio table**	CREATE TABLE portfolio (portfolio_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, github_url VARCHAR(50) NOT NULL, portfolio_url VARCHAR(50) NOT NULL, resume_url VARCHAR(100) NOT NULL, portfolio_status_id INT NOT NULL, batch_id INT NOT NULL, comment VARCHAR(250), student_id INT NOT NULL, FOREIGN KEY (portfolio_status_id) REFERENCES portfolio_statuses(portfolio_status_id), FOREIGN KEY (student_id) REFERENCES students(student_id));
15.	**code_kata table**	CREATE TABLE code_kata (code_kata_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, total_points BIGINT NOT NULL, student_id INT NOT NULL, FOREIGN KEY (student_id) REFERENCES students(student_id));

16.	**web_kata table**	CREATE TABLE web_kata (web_kata_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, total_points BIGINT NOT NULL, student_id INT NOT NULL, FOREIGN KEY (student_id) REFERENCES students(student_id));
17.	**leave_applications table**	CREATE TABLE leave_applications (leave_application_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, days INT NOT NULL, leave_date TIMESTAMP NOT NULL, reason VARCHAR(250), student_id INT NOT NULL, FOREIGN KEY (student_id) REFERENCES students(student_id));

18.	**mock_interviews table**	CREATE TABLE mock_interviews (mock_interview_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, interview_round VARCHAR(250) NOT NULL, score INT NOT NULL, mock_interview_recording_id INT NOT NULL, comments VARCHAR(250), FOREIGN KEY (mock_interview_recording_id) REFERENCES mock_interview_recording_urls(recording_url_id));
19.	**mock_interviews_recordings table**	CREATE TABLE mock_interviews_recording_urls (recording_url_id INT PRIMARY KEY NOT NULL AUTO_INCREMENT, recording_url VARCHAR(300) NOT NULL, mock_interview_id INT NOT NULL, student_id INT NOT NULL, FOREIGN KEY (mock_interview_id) REFERENCES mock_interviews(mock_interview_id), FOREIGN KEY (student_id) REFERENCES students(student_id))