Users :

* UserID (PK) AUTO\_INCREMENT
* Name
* Email (Unique)
* Password
* Gender
* Birth of date
* Profile Picture
* Biography
* Locations (FK)

UserPreferences : (Store user setting e.g. notification settings, preferred subjects)

* UserPreferenceID (PK) AUTO\_INCREMENT
* UserID(FK)
* PreferenceKey VARCHAR (50)

Teachers :

* TeacherID (PK) AUTO\_INCREMENT
* UserID (FK)
* Expertise
* Rate
* Review (FK)

Locations :

* LocationsID (PK) AUTO\_INCREMENT
* UserID(FK)
* Address TEXT

Course :

* CourseID (PK) AUTO\_INCREMENT
* TeacherID (FK)
* Subject
* Description
* Date
* Time
* Duration
* Price

Matching Criteria

* CriterialID (PK) AUTO\_INCREMENT
* UserID (FK)
* Subject
* Price
* Date
* Time

Matches :

* MatchesID(PK) AUTO\_INCREMENT
* CourseID(FK)
* MatchesDateTime
* Status ENUM(‘Active’, ’Completed’, ’Dropped’)

Record :

* RecordID(PK) AUTO\_INCREMENT
* UserID (FK)
* CourseID(FK)
* RecordDateTime
* Status ENUM(‘Active’, ’Completed’, ’Dropped’)

Review :

* ReviewID (PK) AUTO\_INCREMENT
* UserID(FK)
* CourseID(FK)
* Comment Text
* Rating

Relationships :

1. Users has a one-to-many relationship with UserPreferences, Teachers, Locations, MatchingCriteria, Record, and Review.
2. Teachers has a many-to-one relationship with Users.
3. Locations has a many-to-one relationship with Users.
4. Course has a many-to-one relationship with Teachers and a one-to-many relationship with Matches and Record.
5. MatchingCriteria has a many-to-one relationship with Users.
6. Matches has a many-to-one relationship with Course.
7. Record has a many-to-one relationship with both Users and Course.
8. Review has a many-to-one relationship with both Users and Course.

Code :

CREATE TABLE IF NOT EXISTS Users (

UserID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

Password VARCHAR(255) NOT NULL,

Gender ENUM('Male', 'Female', 'Other') NOT NULL,

BirthOfDate DATE NOT NULL,

ProfilePicture VARCHAR(255),

Biography TEXT

);

CREATE TABLE IF NOT EXISTS UserPreferences (

UserPreferenceID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

PreferenceKey VARCHAR(50),

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Teachers (

TeacherID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Expertise VARCHAR(100) NOT NULL,

Rate DECIMAL(5, 2) NOT NULL,

ReviewID INT,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Locations (

LocationsID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Address TEXT,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Course (

CourseID INT PRIMARY KEY AUTO\_INCREMENT,

TeacherID INT,

Subject VARCHAR(100) NOT NULL,

Description TEXT,

Date DATE NOT NULL,

Time TIME NOT NULL,

Duration INT NOT NULL, -- Duration in minutes

Price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (TeacherID) REFERENCES Teachers(TeacherID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS MatchingCriteria (

CriteriaID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

Subject VARCHAR(100),

Price DECIMAL(10, 2),

Date DATE,

Time TIME,

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Matches (

MatchesID INT PRIMARY KEY AUTO\_INCREMENT,

CourseID INT,

MatchesDateTime DATETIME DEFAULT CURRENT\_TIMESTAMP,

Status ENUM('Active', 'Completed', 'Dropped'),

FOREIGN KEY (CourseID) REFERENCES Course(CourseID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Record (

RecordID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

CourseID INT,

RecordDateTime DATETIME DEFAULT CURRENT\_TIMESTAMP,

Status ENUM('Active', 'Completed', 'Dropped'),

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,

FOREIGN KEY (CourseID) REFERENCES Course(CourseID) ON DELETE CASCADE

);

CREATE TABLE IF NOT EXISTS Review (

ReviewID INT PRIMARY KEY AUTO\_INCREMENT,

UserID INT,

CourseID INT,

Comment TEXT,

Rating INT CHECK (Rating BETWEEN 1 AND 5),

FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,

FOREIGN KEY (CourseID) REFERENCES Course(CourseID) ON DELETE CASCADE

);

Testing :

-- Insert Users

INSERT INTO Users (Name, Email, Password, Gender, BirthOfDate, ProfilePicture, Biography) VALUES

('Alice Johnson', 'alice@example.com', 'password123', 'Female', '1995-06-15', 'alice.jpg', 'Passionate tutor with 5 years of experience.'),

('Bob Smith', 'bob@example.com', 'password456', 'Male', '1988-03-22', 'bob.jpg', 'Expert in Mathematics and Physics.'),

('Charlie Brown', 'charlie@example.com', 'password789', 'Other', '1990-12-01', 'charlie.jpg', 'Lifelong learner and educator.');

-- Insert UserPreferences

INSERT INTO UserPreferences (UserID, PreferenceKey) VALUES

(1, 'Notification: Email'),

(1, 'Preferred Subject: Math'),

(2, 'Notification: SMS'),

(3, 'Preferred Subject: Science');

-- Insert Teachers

INSERT INTO Teachers (UserID, Expertise, Rate, ReviewID) VALUES

(1, 'Mathematics', 30.00, NULL),

(2, 'Physics', 40.00, NULL);

-- Insert Locations

INSERT INTO Locations (UserID, Address) VALUES

(1, '123 Maple St, Springfield, IL'),

(2, '456 Oak St, Springfield, IL'),

(3, '789 Pine St, Springfield, IL');

-- Insert Courses with Duration

INSERT INTO Course (TeacherID, Subject, Description, Date, Time, Duration, Price) VALUES

(1, 'Algebra 101', 'Introductory course on Algebra.', '2024-10-20', '10:00:00', 90, 50.00), -- 90 minutes

(2, 'Physics Fundamentals', 'Basic concepts in Physics.', '2024-10-21', '14:00:00', 120, 60.00); -- 120 minutes

-- Insert MatchingCriteria

INSERT INTO MatchingCriteria (UserID, Subject, Price, Date, Time) VALUES

(1, 'Algebra', 45.00, '2024-10-20', '10:00:00'),

(2, 'Physics', 55.00, '2024-10-21', '14:00:00'),

(3, 'Science', 50.00, '2024-10-22', '16:00:00');

-- Insert Matches

INSERT INTO Matches (CourseID, MatchesDateTime, Status) VALUES

(1, '2024-10-15 12:00:00', 'Active'),

(2, '2024-10-16 13:00:00', 'Completed');

-- Insert Record

INSERT INTO Record (UserID, CourseID, RecordDateTime, Status) VALUES

(1, 1, '2024-10-15 12:00:00', 'Active'),

(2, 2, '2024-10-16 13:00:00', 'Completed');

-- Insert Review

INSERT INTO Review (UserID, CourseID, Comment, Rating) VALUES

(1, 1, 'Great course! Very helpful.', 5),

(2, 2, 'The teacher was knowledgeable.', 4);