

Translated Relational Schema

CREATE TABLE User (

UserId VARCHAR(50) PRIMARY KEY,

FirstName VARCHAR(20) NOT NULL,

LastName VARCHAR(20),

Gender VARCHAR(20),

Age REAL,

Email VARCHAR(50) NOT NULL,

Password VARCHAR(100) NOT NULL,

Phone VARCHAR(20),

Location VARCHAR(200),

Type ENUM('S', 'T')

);

CREATE TABLE Student (

StudentId VARCHAR(50) REFERENCES User ON DELETE CASCADE,

Major VARCHAR(20),

PRIMARY KEY (StudentId)

);

CREATE TABLE Teacher (

TeacherId VARCHAR(50) REFERENCES User ON DELETE CASCADE,

Rating REAL,

YearofTeaching REAL NOT NULL,

PRIMARY KEY (TeacherId)

);

CREATE TABLE Sport(

SportId VARCHAR(50) PRIMARY KEY,

SportName VARCHAR(50) NOT NULL,

SportIntroduction VARCHAR(1000) NOT NULL,

SportImage VARCHAR(500),

);

CREATE TABLE Course (

CourseId VARCHAR(50) PRIMARY KEY,

Name VARCHAR(20) NOT NULL,

Location VARCHAR(200),

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

CourseType ENUM('Online', ‘In-person'),

SportId VARCHAR(50) REFERENCES Sport ON DELETE CASCADE,

TeacherId VARCHAR(50) REFERENCES Teacher ON DELETE CASCADE

);

CREATE TABLE Message(

MessageId VARCHAR(50),

LaunchTime DATETIME,

Title VARCHAR(50) NOT NULL,

Content VARCHAR(2000) NOT NULL,

StudentId VARCHAR(50)  REFERENCES Student ON DELETE CASCADE,

PRIMARY KEY (MessageId, StudentId)

);

CREATE TABLE Court(

CourtId VARCHAR(50) PRIMARY KEY,

Name VARCHAR(50),

Location VARCHAR(200) NOT NULL,

SportId VARCHAR(50) REFERENCES Sport ON DELETE CASCADE

);

CREATE TABLE Reservation(

ReservationId VARCHAR(50) PRIMARY KEY,

CourtId VARCHAR(50) REFERENCES Court ON DELETE CASCADE,

UserId VARCHAR(50) REFERENCES Student ON DELETE CASCADE,

BeginTime DATETIME NOT NULL,

EndTime DATETIME NOT NULL

);

CREATE TABLE Appointment (

AppointmentId VARCHAR(50) PRIMARY KEY,

StudentId VARCHAR(50) REFERENCES Student ON DELETE CASCADE,

TeacherId VARCHAR(50) REFERENCES Teacher ON DELETE CASCADE,

Link VARCHAR(200),

ReservationId VARCHAR(50) REFERENCES Reservation ON DELETE CASCADE,

AppointmentType ENUM('Online', 'In-person'),

Comment VARCHAR(2000)

);

CREATE TABLE Enrollment (

StudentId VARCHAR(50) REFERENCES Student ON DELETE CASCADE,

CourseId VARCHAR(50) REFERENCES Course ON DELETE CASCADE,

PRIMARY KEY (StudentId, CourseId)

);

CREATE TABLE Rate (

StudentId VARCHAR(50) REFERENCES Student ON DELETE CASCADE,

CourseId VARCHAR(50) REFERENCES Course ON DELETE CASCADE,

Rating REAL,

Review VARCHAR(2000),

PRIMARY KEY (StudentId, CourseId)

);

CREATE TABLE Hobby (

StudentId VARCHAR(50) REFERENCES Student ON DELETE CASCADE,

SportId VARCHAR(50) REFERENCES Sport ON DELETE CASCADE,

Year REAL,

PRIMARY KEY (StudentId, SportId)

);

Assumptions and Description of the ER/UML diagram

**User**

* A user login entity. It has attributes such as UserId, FirstName and LastName of a user, Gender, Age, Email, Phone, Location, and Type which identifies a user as a student or a teacher. Among them, UserId is the primary key.
* Assumption:
  + FirstName: null values are not allowed because the user's first name is used to reference a person.
  + Email: null values are not allowed because it’s a user’s primary contact information.
  + Type: only student and teacher users are allowed.

**Appointment (one - many relationship between student and teacher)**

* a schema that stores appointment information made between teachers and students. It has attributes such as AppointmentId, StudentId, TeacherId, AppointmentType which identifies it is online or in-person practices, Link used for online practices, ReservationId for in-person practices, and Comment for this appointment. Among them, AppointmentId is the primary key.
* Assumption:
  + StudentId, TeacherId: ON DELETE CASCADE, If a student or teacher user is removed from Student or Teacher entity, appointments involving such user should be canceled and any rows containing such id should also be removed from Appointment table.
  + ReservationId : If a student wants to get in-person practices, he or she should reserve a court and provide ReservationId when making an appointment.
  + AppointmentType: only online and in-person practices are allowed.
  + One-Many relationship: A student can make an appointment with at most one teacher, while a teacher can have appointments with multiple students.

**Teacher**

* An entity contains the information of a teacher. It has attributes such as TeacherId, Rating, and YearofTeaching  that indicates the number of years of teaching. Among them, TeacherId is the primary key.
* Assumption:
  + YearofTeaching: null values are not allowed because it provides information for students who prefer experienced teachers.
  + TeacherId: ON DELETE CASCADE, If a teacher user is removed from User table, appointments involving him or her should be canceled and any rows containing such id should also be removed from Teacher table.

**Reservation (one - many relationship between student and court)**

* This table is used to store students’ reservation information of courts. It has attributes ReservationId, StudentId, CourtId, BeginTime, EndTime. Among them, ReservationId is the primary key that uniquely identifies a reservation. And StudentId is a foreign key referencing a Student entity and CourtId is a foreign key referencing Court entity. To be more specific, BeginTime means when the reservation begins for the court and EndTime means when it ends.
* Assumption:
  + ReservationId: a student could reserve only one court at one time and he could not make any reservation for any court until this reservation ends. But two students could reserve a court for two different time.The time of reservation starts from BeginTime and ends at EndTime. There should be no time conflict for a specific court.
  + StudentId: ON DELETE CASCADE, If a student user is removed from Student entity, reservation made by him or her should be canceled and any rows containing such id should also be removed from Reservation table.
  + CourtId: ON DELETE CASCADE, If a court is removed from Court entity, all reservations of this court should be canceled and any rows containing such id should also be removed from Reservation table.

**Sport**

* One entity of sports type. The primary key is SportId. SportName means the type of the sport, such as baseball, tennis etc. SportIntroduction is a brief introduction of the sport.
* Assumption:
  + SportName: The NULL value of SportName is not allowed as every sport should have a name, and the Student and the Teacher entity should know its name.
  + SportIntroduction: The NULL value of SportIntroduction is not allowed. This text will be presented for the Student entity to introduce each sport.

**Court**

* One entity of the court. It has attributes CourtId, Name, Location, SportIdAmong them, CourtId is the primary key that uniquely identifies a Court. And SportId is a foreign key referencing the Sport tableTo be more specific, BeginTime means when the reservation begins for the court and EndTime means when it ends.
* Assumption:
  + SportId. We assume the student entity could only be one type of sport in the court entity.
  + Location. The NULL value of the location of court entity is not allowed as we will use the location to recommend the nearest court for students.

**Message**

* This table is used to store the invitation messages of students when initiating sports activities. It has attributes MessageId, LaunchTime, Title, Content, StudentId. As entity Message is a weak entity to table Student, StudentId of the sender would be part of its primary key as well as a foreign key referencing table Student. Other than that, attribute MessageId is used to identify the messages sent by the same student. Attribute LaunchTime means the time when invitation messages are sent. Title and Content show when and where the sports activities will take place.
* Assumption:
  + Title: NULL value is not allowed as it is a good way to let other students understand what's inside the message.
  + Content: NULL value is not allowed as we don't hope there are lots of meaningless posts.
  + ON DELETE CASCADE: When accounts of sender students are removed from the database, all the invitation messages they sent will also be deleted, so ON DELETE CASCADE is used in the Message table.

**Rate (many - many relationship between Student and Course)**

* Each student can rate multiple courses and one course can be rated by multiple students. This table is used to store course ratings. It has attributes StudentId, CourseId, Rating and Review. Among them, StudentId and CourseId are the primary keys that uniquely identify a rating.
* Assumption:
  + CourseId: When a course is removed, its rating and reviews received will also be removed. That's why we use ON DELETE CASCADE in the schema.
  + StudentId: When a student account is deleted from the database, all of him/her ratings and reviews will be deleted at the same time.

**Hobby (many - many relationship between Student and Sport)**

* Each student may have many hobbies and each sport can be loved by lots of students. This table is used to store students' hobby information. It has attributes StudentId, SportId, Year. StudentId and SportId are used to identify a hobby relationship. To be more specific, attribute Year is used to store how long one student has been loving the sport.
* Assumption:
  + SportId: When one kind of sport is removed from the database, hobbies of related students should be changed at the same time. That's why we use ON DELETE CASCADE in the Hobby relationship.

**Student**

* An entity contains the information of a student. It has attributes StudentId, Major. Among them, StudentId is the primary key that uniquely identifies a student. StudentId is also the foreign key that references the User entity.
* Assumption:
  + StudentId: If the Student User Information is moved from the User entity, the related Student entity should be moved.

**Course**

* An entity contains the information of a course. It has attributes CourseId, Name, Location, StartDate, EndDate, CourseType, SportId, TeacherId. Among them, CourseId is the primary key that uniquely identifies a course.

To be more specific, attribute Location means the location where this course will be taught. Attribute StartDate and EndDate mean when will the course begin and end. Attribute SportId means which kind of sport will be taught in this class.

* Assumption:
  + Name: The name of course should not be empty.
  + StartDate, EndDate: The start date and end date of the course should not be empty.
  + CourseType: Only two types of classes are allowed: Online or In-person.
  + SportId: The sport that teaches in one class should be found in the Sport entity and if one sport is deleted in Sport entity it should be deleted in Course entity. That's why we use ON DELETE CASCADE in the schema.

**Enrollment (Many to many relationship between student and course)**

* A relationship between student and course. It has attributes StudentId, CourseId. Both of them are the primary keys. And StudentId is a foreign key referencing the Student entity and CourseId is a foreign key referencing the Course entity.
* Assumption:
  + StudentId: When a student entity is deleted, the related enrollment entity should be deleted too. That's why we use ON DELETE CASCADE in the schema.
  + CourseId: When a course entity is deleted, the related enrollment entity should be deleted too. That's why we use ON DELETE CASCADE in the schema.