Here’s a detailed description of the UniLearn database structure, covering all attributes and relationships:

In UniLearn, **universities** serve as the central entities. Each university has a unique id to identify it, along with attributes such as name, email, address, and phone for basic contact information. The subscription\_plan field indicates the university's subscription type, and the created\_at and updated\_at timestamps track when the university was added to the system and when it was last updated. A university can have multiple departments, university admins, and teachers associated with it.

**UniversityAdmins** are administrators within a university, each identified by a unique id. They belong to a specific university, indicated by the university\_id field. The name, email, and password fields store their personal and login information, while the role attribute specifies their position, such as Main Admin or Sub Admin. The created\_at and updated\_at fields track their record creation and update times.

**Departments** within a university are identified by their unique id. Each department is linked to a university through the university\_id field. Departments have a name, and the created\_at and updated\_at fields keep track of when the department was added and last modified. Departments can have multiple batches, department admins, and teachers.

**DepartmentAdmins** manage departmental activities. Each department admin is identified by a unique id and is linked to a specific department via the department\_id field. The name, email, and password fields store their personal and login details, and the role field indicates their position, such as Department Head or Coordinator. Their record creation and update times are stored in the created\_at and updated\_at fields.

**Teachers** play a crucial role in UniLearn. Each teacher has a unique id and can be associated with a university (university\_id) and a department (department\_id). The name, email, and password fields contain the teacher's personal and login information. The supervisor boolean attribute indicates whether the teacher also serves as a batch supervisor. The created\_at and updated\_at fields record when the teacher was added and last updated. Teachers can manage multiple courses, participate in course-specific chats, and supervise batches.

**Batches** represent groups of students within a department. Each batch has a unique id and is linked to a department through the department\_id field. The supervisor\_id field points to the teacher who supervises the batch. Batches are identified by their name and year. The created\_at and updated\_at fields track when the batch was created and last modified. Batches can include multiple courses, students, and group chats.

**Courses** are academic subjects taught within batches. Each course has a unique id and is linked to a batch via the batch\_id field. Courses are identified by their name, code, and description. The created\_at and updated\_at fields record when the course was added and last updated. A course can have multiple students and teachers (through the CourseTeacher entity), and it can have its own course-specific chats.

**CourseTeacher** is an entity that links teachers to the courses they teach. Each entry has a unique id and is connected to a specific course (course\_id) and a teacher (teacher\_id). The assigned\_at field records when the teacher was assigned to the course.

**Students** are the learners in UniLearn. Each student has a unique id and is associated with a batch via the batch\_id field. The name, email, and password fields store their personal and login details. The created\_at and updated\_at fields track when the student was added and last updated. Students can enroll in multiple courses (through the CourseStudent entity) and participate in course-specific chats. Each student also has a profile that stores additional information.

**CourseStudent** is an entity that connects students to the courses they are enrolled in. Each entry has a unique id and is linked to a specific course (course\_id) and student (student\_id). The enrolled\_at field records when the student enrolled in the course.

**GroupChats** facilitate communication within batches or courses. Each chat has a unique id and is associated with either a batch (batch\_id) or a course (course\_id). The created\_at and updated\_at fields track when the chat was created and last updated. Group chats can contain multiple messages and participants, including both teachers and students.

**Messages** are the individual communications within group or private chats. Each message has a unique id and is linked to a specific chat (chat\_id). The sender\_id field identifies the sender, who can be either a student or a teacher. The message\_text field contains the content of the message, and the sent\_at field records the time the message was sent.

**PrivateChats** enable one-on-one communication between students and teachers. Each chat has a unique id and involves a sender (sender\_id) and a receiver (receiver\_id), who can be either students or teachers. The created\_at and updated\_at fields record when the chat was created and last updated. Private chats can contain multiple messages.

**Profiles** store additional personal information for both students and teachers. Each profile has a unique id and is linked to a user (user\_id), which can be either a student or a teacher. The profile\_picture field stores the user's image, and the bio field contains a brief description of the user. The created\_at and updated\_at fields track when the profile was created and last updated.

**Overall Structure**

Universities manage their departments, teachers, and students through their main dashboard. Departments oversee batches and have communication with batch supervisors, who are also teachers. Batch supervisors manage courses, students, and other teachers within their batches. Both students and teachers have their own dashboards for managing courses and profiles. The system supports group and private messaging between different roles, ensuring effective communication throughout the platform.

### ****Entities and Relationships****

1. **Universities**
   * **Fields:**
     + id: Primary Key
     + name
     + email
     + address
     + phone
     + subscription\_plan
     + created\_at
     + updated\_at
   * **Relationships:**
     + Has many Departments
     + Has many UniversityAdmins
     + Has many Teachers
2. **UniversityAdmins**
   * **Fields:**
     + id: Primary Key
     + university\_id: Foreign Key
     + name
     + email
     + password
     + role: (Main Admin, Sub Admin, etc.)
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to University
3. **Departments**
   * **Fields:**
     + id: Primary Key
     + university\_id: Foreign Key
     + name
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to University
     + Has many Batches
     + Has many DepartmentAdmins
     + Has many Teachers
4. **DepartmentAdmins**
   * **Fields:**
     + id: Primary Key
     + department\_id: Foreign Key
     + name
     + email
     + password
     + role: (Department Head, Coordinator, etc.)
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to Department
5. **Teachers**
   * **Fields:**
     + id: Primary Key
     + university\_id: Foreign Key
     + department\_id: Foreign Key
     + name
     + email
     + password
     + supervisor: Boolean (true if the teacher is also a batch supervisor)
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to University
     + Belongs to Department
     + Has many Courses (via CourseTeacher)
     + Participates in CourseChats
     + Has one Profile
     + Can supervise Batches
6. **Batches**
   * **Fields:**
     + id: Primary Key
     + department\_id: Foreign Key
     + supervisor\_id: Foreign Key (points to Teachers.id)
     + name
     + year
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to Department
     + Belongs to Supervisor (Teacher)
     + Has many Courses
     + Has many Students
     + Has many GroupChats
7. **Courses**
   * **Fields:**
     + id: Primary Key
     + batch\_id: Foreign Key
     + name
     + code
     + description
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to Batch
     + Has many Students (via CourseStudent)
     + Has many Teachers (via CourseTeacher)
     + Has many CourseChats
8. **CourseTeacher**
   * **Fields:**
     + id: Primary Key
     + course\_id: Foreign Key
     + teacher\_id: Foreign Key
     + assigned\_at
   * **Relationships:**
     + Belongs to Course
     + Belongs to Teacher
9. **Students**
   * **Fields:**
     + id: Primary Key
     + batch\_id: Foreign Key
     + name
     + email
     + password
     + created\_at
     + updated\_at
   * **Relationships:**
     + Belongs to Batch
     + Has many Courses (via CourseStudent)
     + Participates in CourseChats
     + Has one Profile
10. **CourseStudent**
    * **Fields:**
      + id: Primary Key
      + course\_id: Foreign Key
      + student\_id: Foreign Key
      + enrolled\_at
    * **Relationships:**
      + Belongs to Course
      + Belongs to Student
11. **GroupChats**
    * **Fields:**
      + id: Primary Key
      + batch\_id: Foreign Key (for general batch chat)
      + course\_id: Foreign Key (for course-specific chat)
      + created\_at
      + updated\_at
    * **Relationships:**
      + Belongs to Batch or Course
      + Has many Messages
      + Has many Participants (Teachers and Students)
12. **Messages**
    * **Fields:**
      + id: Primary Key
      + chat\_id: Foreign Key (GroupChat or PrivateChat)
      + sender\_id: Foreign Key (Student or Teacher)
      + message\_text
      + sent\_at
    * **Relationships:**
      + Belongs to GroupChat or PrivateChat
      + Belongs to Sender (Student or Teacher)
13. **PrivateChats**
    * **Fields:**
      + id: Primary Key
      + sender\_id: Foreign Key (Student or Teacher)
      + receiver\_id: Foreign Key (Student or Teacher)
      + created\_at
      + updated\_at
    * **Relationships:**
      + Has many Messages
14. **Profiles**
    * **Fields:**
      + id: Primary Key
      + user\_id: Foreign Key (Teacher or Student)
      + profile\_picture
      + bio
      + created\_at
      + updated\_at
    * **Relationships:**
      + Belongs to User (Teacher or Student)

### ****Overall Structure****

* **Universities** manage their departments, teachers, and students through their main dashboard.
* **Departments** manage batches, teachers, and have communication with batch supervisors.
* **BatchSupervisors** (who are also teachers) manage courses, students, and other teachers within their batches.
* **Students and Teachers** have their own dashboards for managing courses and profiles.
* **Chatting System** supports group and private messages between different roles (admins, teachers, students).

generator client {

provider = "prisma-client-js"

}

datasource db {

provider = "mongodb"

url = env("DATABASE\_URL")

}

model University {

id String @id @default(auto()) @map("\_id") @db.ObjectId

name String

email String @unique

address String

phone String

subscriptionPlan String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

departments Department[]

admins UniversityAdmin[]

teachers Teacher[]

}

model UniversityAdmin {

id String @id @default(auto()) @map("\_id") @db.ObjectId

universityId String @db.ObjectId

university University @relation(fields: [universityId], references: [id])

name String

email String @unique

password String

role String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

}

model Department {

id String @id @default(auto()) @map("\_id") @db.ObjectId

universityId String @db.ObjectId

university University @relation(fields: [universityId], references: [id])

name String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

batches Batch[]

admins DepartmentAdmin[]

teachers Teacher[]

}

model DepartmentAdmin {

id String @id @default(auto()) @map("\_id") @db.ObjectId

departmentId String @db.ObjectId

department Department @relation(fields: [departmentId], references: [id])

name String

email String @unique

password String

role String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

}

model Teacher {

id String @id @default(auto()) @map("\_id") @db.ObjectId

universityId String @db.ObjectId

departmentId String @db.ObjectId

university University @relation(fields: [universityId], references: [id])

department Department @relation(fields: [departmentId], references: [id])

name String

email String @unique

password String

supervisor Boolean

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

courses CourseTeacher[]

profile Profile?

supervisedBatches Batch[]

}

model Batch {

id String @id @default(auto()) @map("\_id") @db.ObjectId

departmentId String @db.ObjectId

supervisorId String @db.ObjectId

department Department @relation(fields: [departmentId], references: [id])

supervisor Teacher @relation(fields: [supervisorId], references: [id])

name String

year Int

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

courses Course[]

students Student[]

groupChats GroupChat[]

}

model Course {

id String @id @default(auto()) @map("\_id") @db.ObjectId

batchId String @db.ObjectId

batch Batch @relation(fields: [batchId], references: [id])

name String

code String

description String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

students CourseStudent[]

teachers CourseTeacher[]

courseChats CourseChat[]

}

model CourseTeacher {

id String @id @default(auto()) @map("\_id") @db.ObjectId

courseId String @db.ObjectId

teacherId String @db.ObjectId

course Course @relation(fields: [courseId], references: [id])

teacher Teacher @relation(fields: [teacherId], references: [id])

assignedAt DateTime @default(now())

}

model Student {

id String @id @default(auto()) @map("\_id") @db.ObjectId

batchId String @db.ObjectId

batch Batch @relation(fields: [batchId], references: [id])

name String

email String @unique

password String

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

courses CourseStudent[]

profile Profile?

courseChats CourseChat[]

}

model CourseStudent {

id String @id @default(auto()) @map("\_id") @db.ObjectId

courseId String @db.ObjectId

studentId String @db.ObjectId

course Course @relation(fields: [courseId], references: [id])

student Student @relation(fields: [studentId], references: [id])

enrolledAt DateTime @default(now())

}

model GroupChat {

id String @id @default(auto()) @map("\_id") @db.ObjectId

batchId String? @db.ObjectId

courseId String? @db.ObjectId

batch Batch? @relation(fields: [batchId], references: [id])

course Course? @relation(fields: [courseId], references: [id])

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

messages Message[]

}

model Message {

id String @id @default(auto()) @map("\_id") @db.ObjectId

chatId String @db.ObjectId

senderId String @db.ObjectId

chat GroupChat? @relation(fields: [chatId], references: [id])

messageText String

sentAt DateTime @default(now())

sender User @relation(fields: [senderId], references: [id])

}

model PrivateChat {

id String @id @default(auto()) @map("\_id") @db.ObjectId

senderId String @db.ObjectId

receiverId String @db.ObjectId

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

messages Message[]

}

model Profile {

id String @id @default(auto()) @map("\_id") @db.ObjectId

userId String @db.ObjectId

profilePicture String?

bio String?

createdAt DateTime @default(now())

updatedAt DateTime @updatedAt

user User @relation(fields: [userId], references: [id])

}

model User {

id String @id @default(auto()) @map("\_id") @db.ObjectId

email String @unique

password String

}