Database Architecture Document for EduLearn LMS

Shivaramakrishna Aravapally, Surya Prakash Reddy Pachika

September 2024

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the database architecture of the Learning Management System (LMS) application. This includes the design of tables, relationships between entities, and how data flows within the system. The goal is to ensure that the database supports the required functionality of the LMS application.

1.2 Scope

This document covers the database structure for the LMS application, including all tables related to users, classes, assignments, materials, videos, sessions, and other key entities. It is intended for developers, database administrators, and other stakeholders involved in the design, implementation, and maintenance of the LMS database.

2 Database Design Overview

The LMS application uses a relational database to store and manage data. The primary entities in the database include Teachers, Students, Classes, Assignments, Materials, Videos, Submissions and Sessions. These entities are represented by tables in the database, with relationships established between them to ensure data integrity and efficient querying.

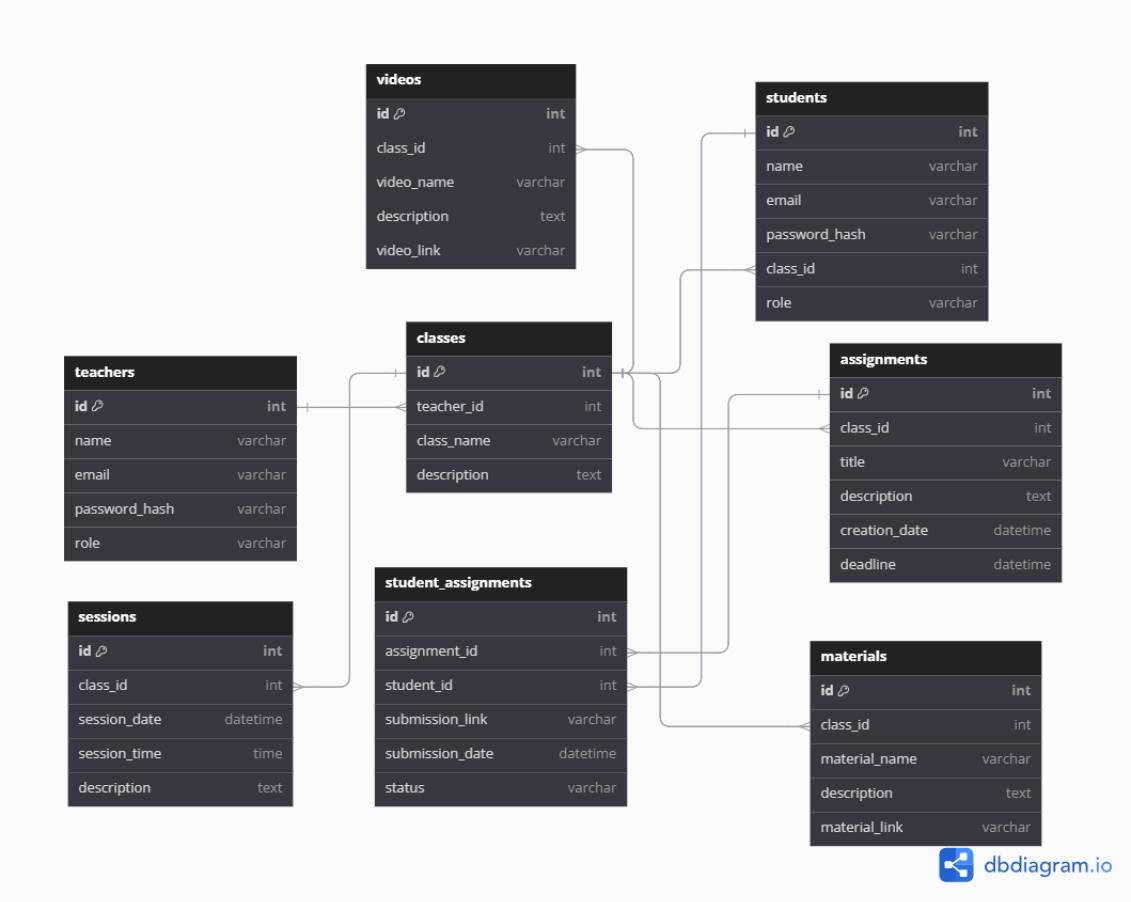

Figure 1: Database Schema

3 Database Tables and Relationships

3.1 Teachers Table

The **Teachers** table stores information specific to teachers, including their name, email, and the classes they manage.

- **Columns**:
  - *id* (Primary Key)
  - *name*
  - *email*
  - *password*
  - *role* (Foreign Key, references Users table)

3.2 Students Table

The **Students** table stores information specific to students, including their name, email, and the classes they are enrolled in.

- **Columns**:
  - *id* (Primary Key)
  - *name*
  - *email*
  - *password*
  - *class_id* (Foreign Key, references Classes table)

3.3 Classes Table

The **Classes** table stores information about the classes managed by teachers and attended by students.

- **Columns**:
  - *id* (Primary Key)
  - *name*
  - *description*
  - *teacher_id* (Foreign Key, references Teachers table)

3.4 Assignments Table

The **Assignments** table stores details about assignments given to students in a class.

- **Columns**:
  - *id* (Primary Key)
  - *title*
  - *description*
  - *file_link*
  - *deadline*
  - *class_id* (Foreign Key, references Classes table)

3.5 Materials Table

The **Materials** table stores educational materials uploaded by teachers for their classes.

- **Columns**:
  - *id* (Primary Key)
  - *name*
  - *description*
  - *link*
  - *class_id* (Foreign Key, references Classes table)

3.6 Videos Table

The **Videos** table stores video content uploaded by teachers for their classes.

- **Columns**:
  - *id* (Primary Key)
  - *title*
  - *link*
  - *description*
  - *class_id* (Foreign Key, references Classes table)

3.7 Sessions Table

The **Sessions** table stores information about class sessions, including the date, time, and session link.

- **Columns**:
  - *id* (Primary Key)
  - *name*
  - *class_id* (Foreign Key, references Classes table)
  - *session_date*
  - *session_time*
  - *session_link*
  - *description*

3.8 Assignment Submissions Table

The **Assignment Submissions** table tracks student submissions for assignments.

- **Columns**:
  - *id* (Primary Key)
  - *student_id* (Foreign Key, references Students table)
  - *assignment_id* (Foreign Key, references Assignments table)
  - *submission_link*
  - *submitted_at*

4 Data Flow

Data flows through the LMS database as users interact with the system. For instance, when a teacher creates an assignment, the Assignment Service inserts the assignment details into the Assignments table, linked to the appropriate class. When students submit their assignments, their submissions are recorded in the Assignment Submissions table, linked to both the student and the assignment.

Similarly, when a teacher uploads materials or videos, the Material Service and Video Service store the respective content in the Materials and Videos tables. Students can then access this content through the Student Service, which retrieves the data based on their enrolled classes.

Data consistency and integrity are maintained through foreign key relationships, ensuring that all records are properly linked and no orphaned records exist in the database.

5 Conclusion

This document outlines the database architecture for the LMS application, detailing the design of tables, relationships, and data flow. The architecture is designed to support the scalability, reliability, and performance needs of the LMS, ensuring a robust and efficient system for managing digital learning.