Learning Management System (LMS)

# Database Design

Version 1.0.0 02/11/2024

Project report for CSC422 by Group 3: Nathan Mollica, Peter Momsen, John Padworski, Brandon Perez-Lee

Table of Contents

[Database Design 1](#_Toc159786553)

[1. *Introduction* 4](#_Toc159786554)

[**Features and Functionalities:** 5](#_Toc159786555)

[**Unique Aspects:** 6](#_Toc159786556)

[2. Relationships & Cardinality 7](#_Toc159786557)

[**Relationships** 7](#_Toc159786558)

[**Cardinality and Participation Constraints** 7](#_Toc159786559)

[**Entity Relationship Diagram (ERD)** 8](#_Toc159786560)

[3. Project Normalization to 3NF 9](#_Toc159786561)

[Appendices 12](#_Toc159786562)

## Introduction

In the current era of digital education, Learning Management Systems (LMS) play a crucial role in facilitating educational and professional development across the globe. Platforms like Udemy have demonstrated the immense potential of LMS to offer accessible and diverse learning opportunities to a broad audience. This project centers on designing a database for an LMS that efficiently serves the dual audience of students seeking knowledge and instructors sharing their expertise. We aim to create a database structure that supports robust content delivery, enhances user engagement, and ensures scalability to accommodate growth. By focusing on these aims, we contribute to advancing digital learning, addressing the challenges learners and educators face in a rapidly evolving educational landscape.

The Learning Management System (LMS) we propose to develop is designed with the core functionalities that are essential for a comprehensive educational experience. At its heart, the LMS will facilitate course creation, allowing instructors to design and publish courses tailored to various subjects. However, students can enroll in these courses, engaging with content spanning various disciplines and skill levels. We envision content delivery within our LMS as dynamic and interactive, incorporating multimedia materials to enhance learning. Moreover, the system will support various assessment forms, including quizzes and assignments, to measure student progress and achievement effectively.

Our LMS will cater to two primary user groups: students and instructors. Students seeking to broaden their knowledge or acquire new skills will find the LMS a valuable resource for their educational journey. They can browse and enroll in courses, access lessons at their own pace, participate in quizzes, and track their achievements through a personalized dashboard. From academic to industry professionals, instructors can share their expertise by creating and managing courses. This interaction facilitates knowledge dissemination and fosters a community of learners and educators within the platform.

Key features of the LMS include an intuitive course browsing and enrollment system, ensuring that students can easily find courses that match their interests and goals. We design lesson access to be seamless, supporting various content formats to cater to different learning preferences. The quiz module will offer a robust tool for instructors to create engaging and challenging assessments. At the same time, the achievement tracking feature will motivate students by visibly marking their progress and accomplishments. Together, these features aim to create an enriching learning environment that supports the educational needs of our diverse user base.

### Features and Functionalities:

1. User Management:
   * The LMS app database application facilitates the management of user accounts, including students, instructors, and administrators.
   * It allows for user authentication, registration, and profile management, enabling users to access appropriate resources based on their roles.
   * User data storage and security are paramount, ensuring the confidentiality and integrity of personal information.
2. Course Management:
   * Course management features enable administrators and instructors to create, organize, and deliver educational content effectively.
   * The application supports the creation of courses, modules, lessons, assignments, quizzes, and other learning materials.
   * It allows for customization of course structures, including prerequisites, deadlines, and grading criteria.
3. Content Management:
   * The LMS app database application stores and manages various types of educational content, such as documents, presentations, videos, and interactive multimedia.
   * It provides version control and access permissions to ensure that users can only view or modify content as authorized.
4. Assessment and Grading:
   * Assessment functionalities enable instructors to create and administer quizzes, exams, and assignments within the LMS.
   * The application supports automated grading, feedback provision, and gradebook management, streamlining the assessment process.
5. Communication and Collaboration:
   * The LMS facilitates communication and collaboration among users through features like discussion forums, messaging, and video conferencing.
   * It fosters a sense of community and engagement, allowing students and instructors to interact effectively beyond traditional classroom settings.
6. Reporting and Analytics:
   * The application generates reports and analytics to track user progress, course performance, and system usage.
   * It offers insights into learning outcomes, student engagement, and areas for improvement, empowering administrators, and instructors to make data-driven decisions.

### Unique Aspects:

1. Scalability:
   * One of the unique challenges of an LMS app database application is scalability, as it must accommodate a potentially large number of users, courses, and content items.
   * The database architecture should be designed to handle increasing loads and user concurrency efficiently, ensuring optimal performance under varying usage scenarios.
   * Scalability also extends to supporting multiple deployment models, such as cloud-based, on-premises, or hybrid, to meet the diverse needs of educational institutions.
2. Customization:
   * An effective LMS app database application allows for customization to meet the unique needs and preferences of different educational institutions, departments, or individual users.
   * Customization options may include branding, user interface configurations, integrations with external systems, and support for diverse instructional methodologies.
   * Flexibility in customization empowers educators to tailor the LMS to align with their pedagogical approaches and organizational requirements, enhancing user satisfaction and adoption.
3. Security:
   * Security is a critical consideration for an LMS app database application, given the sensitivity of educational data and the potential risks of unauthorized access or data breaches.
   * The application must implement robust security measures, including encryption, access controls, and regular security audits, to safeguard user privacy and confidentiality.
   * Compliance with data protection regulations, such as GDPR or FERPA, is essential to ensure legal and ethical handling of user data, reinforcing trust and confidence in the LMS platform.
4. Accessibility:
   * Accessibility is another significant aspect of an LMS app database application, as it should be designed to accommodate users with diverse needs and disabilities.
   * The application should adhere to accessibility standards, such as WCAG (Web Content Accessibility Guidelines), to ensure equitable access to educational resources and functionalities for all users.
   * Features like alternative text for images, keyboard navigation, and screen reader compatibility contribute to a more inclusive learning environment, promoting equal opportunities for all learners.

In summary, a LMS app database application plays a vital role in modern education by providing a centralized platform for managing courses, users, content, and assessments. Its features and functionalities empower educators to deliver engaging and interactive learning experiences while facilitating communication, collaboration, and data-driven decision-making. Despite the challenges of scalability, customization, security, and accessibility, a well-designed LMS app database application can significantly enhance the effectiveness and efficiency of online education, fostering a culture of continuous learning and innovation in educational institutions.

## Relationships & Cardinality

### **Relationships**

* **User to Course:** Many-to-Many (Instructors teach many courses; students enroll in many courses)
* **Course to Module:** One-to-Many (A course contains many modules)
* **Module to Lesson:** One-to-Many (A module contains many lessons)
* **Course to Assignment\_Quiz:** One-to-Many (A course has many assignments/quizzes)
* **User to Submission:** One-to-Many (A user can make many submissions, but each submission is made by one user)
* **Lesson to Content:** One-to-Many (A lesson can have multiple contents)
* **User to Communication:** Many-to-Many (Users can send messages to many users, and receive from many)

### **Cardinality and Participation Constraints**

* **User to Course:** (N:M)
  + A user can be associated with multiple courses either as a student, instructor, or both.
  + A course must have at least one instructor but can have many students.
  + (**Instructor**) - Mandatory
  + (**Student**) - Optional
* **Course to Module, Module to Lesson, Lesson to Content:** (1:N)
  + A parent entity must have at least one child entity but can have many.
  + (**Course-to-Module**) - Mandatory
  + (**Module-to-Lesson**) - Mandatory
  + (**Lesson-to-Content**) - Mandatory
* **Course to Assignment\_Quiz:** (1:N)
  + Each course must have at least one assignment or quiz.
  + Mandatory
* **User to Submission:** (1:N) '
  + A submission must be made by exactly one user, but a user can make multiple submissions.
  + Optional

### **Entity Relationship Diagram (ERD)**

A screenshot of a computer

Description automatically generated

Figure 1: Entity Relationship Diagram (ERD)

## Project Normalization to 3NF

*Prenormalized SQL Code -* [*Appendix 1: Original SQL Code*](#_Appendices)

**First Normal Form (1NF):** Each table has a primary key, and all columns contain atomic (indivisible) values. There are no repeating groups.

**Second Normal Form (2NF):** It is in 1NF, and all non-key attributes are fully functional and dependent on the primary key.

**Third Normal Form (3NF):** It is in 2NF, and all the attributes are only dependent on the primary key, not on any other non-prime attribute (i.e., no transitive dependencies).

**LMS Schema Review**

**Users Table**

**1NF:** Achieved through a primary key (UserID) and atomic columns.

**2NF:** Achieved since all information is directly related to the UserID.

**3NF:** Achieved, as there are no transitive dependencies; all attributes directly depend on UserID.

**Courses Table**

**1NF:** Primary key (CourseID) ensures uniqueness, and all attributes are atomic.

**2NF:** Achieved, but introduces a potential issue with InstructorID pointing to Users. The direct dependency is on CourseID, but InstructorID relies on the existence of a user. This setup is logical for this context, as it represents a real-world relationship.

**3NF:** The table is in 3NF if we consider InstructorID as a foreign key that directly depends on the CourseID for its context within the Courses table. It's not a transitive dependency because it represents a direct relationship, not an attribute of another non-prime attribute.

**Modules, Lessons, Assignment\_Quiz, Submissions, Content, and Communication Tables**

**1NF:** Each table has a primary key and atomic attribute.

**2NF:** All non-key attributes are fully dependent on their respective primary keys, with no partial dependencies.

**3NF:** There are no transitive dependencies; non-key attributes depend only on the primary key.

**Considerations for Normalization to 3NF:**

**Eliminate Redundant Data:** Ensure that user information is not duplicated across tables. For example, InstructorID in Courses table should not lead to redundant user information elsewhere.

**Foreign Key References:** Maintain integrity through foreign keys without causing unnecessary duplication. For instance, InstructorID, UserID, and ModuleID serve as foreign keys that link tables based on logical relationships, not duplicating information.

**Avoid Transitive Dependencies:** If any table appears to have attributes that depend on non-key attributes, consider restructuring or splitting the table. However, in the provided schema, such dependencies do not seem to be present.

**Specific Adjustments:**

**Courses Table:** If instructors can teach multiple courses and if there's significant instructor-specific data relevant to courses (e.g., teaching style, feedback ratings), we might consider an InstructorDetails table linked to Users to keep user data and instructor-specific data separate.

This is more about enhancing design than a requirement for 3NF.

**Normalization and Real-world Applications:** While the schema is normalized to 3NF based on the provided structure, it's essential to balance normalization with practical considerations like query performance, readability, and the complexity of JOIN operations in a real-world application.

**Key Adjustments:**

**Instructors Table:**

A new table Instructors is introduced to separate instructor-specific data from the general user data. This table includes attributes like Bio and Qualifications which are specific to instructors.

It's linked to the Users table via a foreign key (UserID), ensuring that each instructor record is directly associated with a user record.

This separation maintains the 3NF by eliminating transitive dependencies within the Users table and allows for more detailed information to be stored about instructors without cluttering the Users table.



**Courses Table:**

The foreign key in the Courses table now references Instructors (InstructorID) instead of Users (UserID), directly linking courses to the instructors table.

This change ensures that course-instructor relationships are clearly defined and manageable, especially when a user has multiple roles or when detailed instructor information needs to be accessed or updated.



These adjustments enhance the database schema's adherence to 3NF by ensuring all attributes depend on the primary key, reducing redundancy, and improving the clarity and efficiency of data management within the LMS.

# Appendices



Appendix 1: Original SQL Code