

BEST PRACTICES OF CREATING RELATIONAL DATABASE DESIGN FOR E-LEARNING PLATFORM

Badalova Lobar Burhonovna

Master's student of National University of Uzbekistan

Contacts: +998944040855 lobar.fb@gmail.com

Abstract: *This article presents best practices for creating a relational database design for an e-learning platform aimed at improving educational outcomes for children in Uzbekistan. The current school program is outdated, and there is a need for new tools and approaches to support the learning and development of young children. To address this need, we are developing a website that will provide video materials and lessons to help children learn new skills. Our research suggests that careful planning, normalization, and the use of appropriate data types and constraints are essential for creating a robust and scalable database design. While the website is not yet ready, we have completed the database design and are making progress towards our goal. This article provides valuable insights and practical guidance for developers and designers working on e-learning platforms.*

Keywords: *Database design, relational database, e-learning, online education, database schema, normalization, data types, constraints, children's education*


Introduction

Education is a fundamental right and a key driver of economic and social development. In Uzbekistan, however, many schools face challenges in providing high-quality education to young children. Outdated curricula, inadequate resources, and difficulties in monitoring student performance are among the obstacles that must be overcome to improve educational outcomes.

One promising approach to addressing these challenges is the use of e-learning platforms. By providing access to high-quality educational materials and tools for tracking student progress, e-learning platforms have the potential to revolutionize the way children learn and develop.

In this article, we focus on the use of e-learning platforms to teach coding skills to children in Uzbekistan. Many schools in Uzbekistan still use outdated programming languages like Pascal, which are no longer relevant in today's rapidly changing technological landscape. Our goal is to create an e-learning platform that





provides children with access to up-to-date coding courses and resources, and that allows teachers to create and sell high-quality educational content at an affordable price.

A key component of any e-learning platform is its database. The database must be carefully designed to support the efficient storage and retrieval of data, and to ensure the integrity and consistency of the information it contains. In this article, we present best practices for creating a relational database design for an e-learning platform aimed at improving coding education for children in Uzbekistan.

Our research is based on a review of the literature and an analysis of case studies. We identify key principles and strategies for designing an effective and efficient database, and provide practical guidance for developers and designers working on e-learning platforms. While our focus is on the specific challenges faced by schools in Uzbekistan, our findings have broader relevance for anyone interested in using technology to improve education.

Main Part

When creating a relational database design for an e-learning platform, it is important to follow best practices to ensure that the database is effective and efficient. In this section, we discuss several key best practices and provide practical examples of how they can be applied in the development of an e-learning platform.

Planning

The first step in creating a relational database design is careful planning. This involves identifying the data that will be stored in the database and organizing it into tables and columns. Each table should represent a distinct entity or concept, and each column should represent an attribute or characteristic of that entity.

For example, in our e-learning platform, we have tables for courses, students, teachers, enrollments, payment, assignments and more. The *courses* table contains information about the courses offered on the platform, such as the course title and description. The *students* table contains information about the students who use the platform, such as their name and contact information. The *teachers* table contains information about the teachers who create and deliver courses on the platform. The *enrollments* table contains information about which students are enrolled in which courses. And the *assignments* table contains information about the assignments that students submit as part of their coursework.

When planning our database design, we carefully considered the relationships between these different entities. For instance, we recognized that each course can have multiple students enrolled in it, and each student can be enrolled in multiple courses. This is an example of a many-to-many relationship.



We also recognized that each teacher can have multiple courses associated with it, and each course can belong to only one teacher. This is an example of a one-to-many relationship.

Normalization

The second best practice for creating a relational database design is normalization. Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. This involves dividing data into multiple related tables and establishing relationships between them using foreign keys.

In our e-learning platform, we used normalization to organize our data into multiple related tables. For example, instead of storing all information about courses and students in a single table, we divided this data into separate *courses* and *students* tables. We then used foreign keys to link these tables together and establish relationships between them. (image 1)

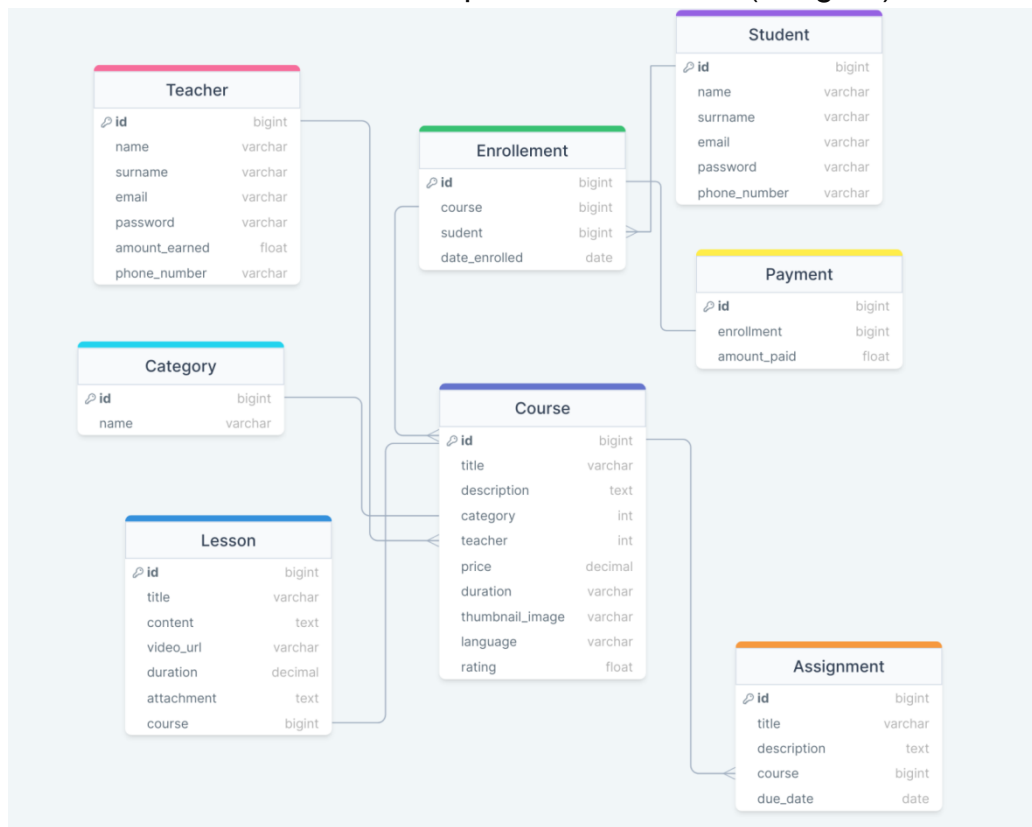



Image 1.

To represent the many-to-many relationship between courses and students, we created an additional *enrollments* table. This table has two foreign keys: one





that references the *students* table to identify the student who is enrolled in the course, and another that references the *courses* table to identify the course in which the student is enrolled.

To represent the one-to-many relationship between courses and teacher, we added a foreign key to the *courses* table that references the *teacher* table. This foreign key identifies the teacher to whom each course belongs.(image 1)

Setting Constraints

The third best practice for creating a relational database design is setting constraints. Constraints are rules that define what data can be stored in a database and how it can be manipulated. They help to ensure data integrity by preventing invalid or inconsistent data from being entered into the database.

In our e-learning platform, we used several types of constraints to ensure data integrity. For example, we used *primary key* constraints to uniquely identify each row in a table. We also used *foreign key* constraints to ensure that data in related tables was consistent. And we used check constraints to ensure that data entered into the database met certain criteria.

For instance, we might use a check constraint on the *enrollments* table to ensure that students can only enroll in courses that are currently being offered on the platform. This could be achieved by checking that the course referenced by each row in the *enrollments* table exists in the *courses* table.

```
CREATE TABLE enrollments (  
    student_id INTEGER REFERENCES students(id),  
    course_id INTEGER REFERENCES courses(id),  
    UNIQUE(student_id, course_id),  
    CHECK (course_id IN (SELECT id FROM courses))  
);
```

We also used unique constraints to ensure that each *student* could only enroll in a given course once. This could be achieved by creating a unique constraint on the combination of foreign keys in the *enrollments* table.

In addition to these general constraints, we also implemented specific constraints to support the functionality of our e-learning platform. For example, we might use check constraints on the *assignments* table to ensure that students can only submit assignments for courses in which they are enrolled.

```
CREATE TABLE assignments (  
    ...  
    student_id INTEGER REFERENCES students(id),  
    course_id INTEGER REFERENCES courses(id),  
    CHECK ((student_id, course_id) IN (SELECT student_id,  
    course_id FROM enrollments))  
);
```



These are just some examples of best practices for creating a relational database design for an e-learning platform. By following these principles and strategies and applying them in practical ways, developers and designers can create effective and efficient databases that support the goals of their e-learning platforms.

We developed a Django admin panel for our e-learning platform to allow administrators to manage the platform's data. Using the admin panel, administrators can create and manage records in the Course (image 3), Category (image 2), and Lesson(image 4) tables.

The Django admin panel provides a user-friendly interface for managing the platform's data. Administrators can use the panel to create new records, update existing records, and delete records as needed. This makes it easy for administrators to keep the platform's data up-to-date and ensure that students have access to the latest courses and lessons.

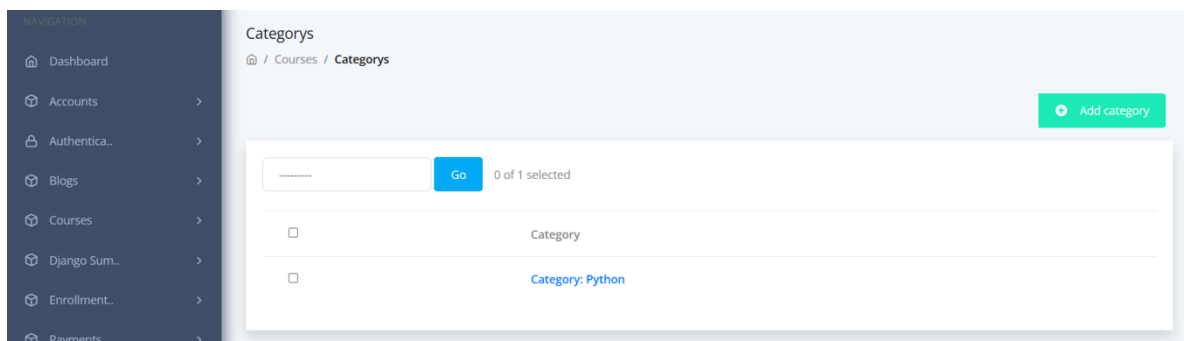


Image 2

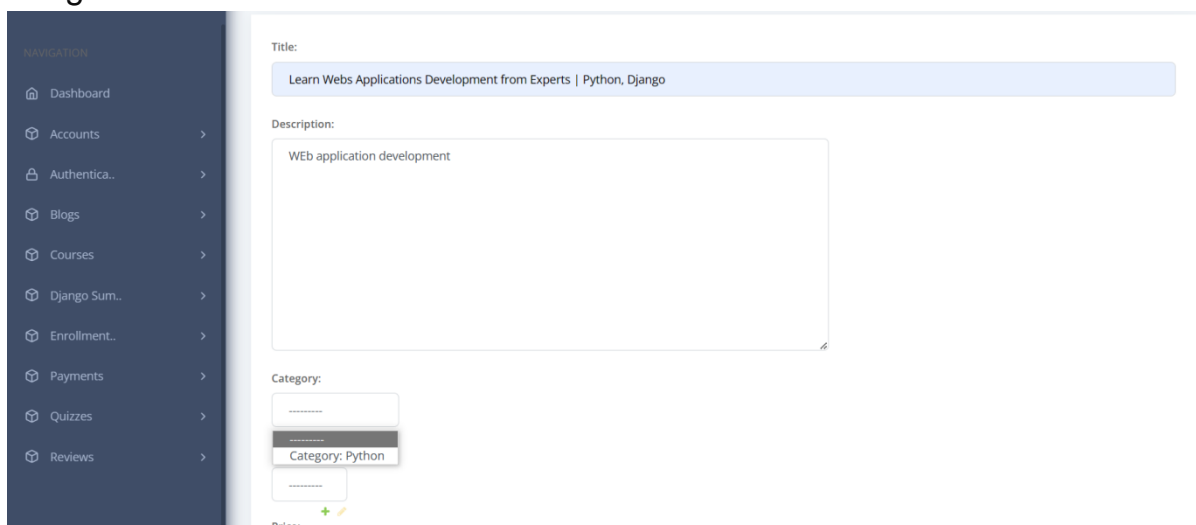


Image 3



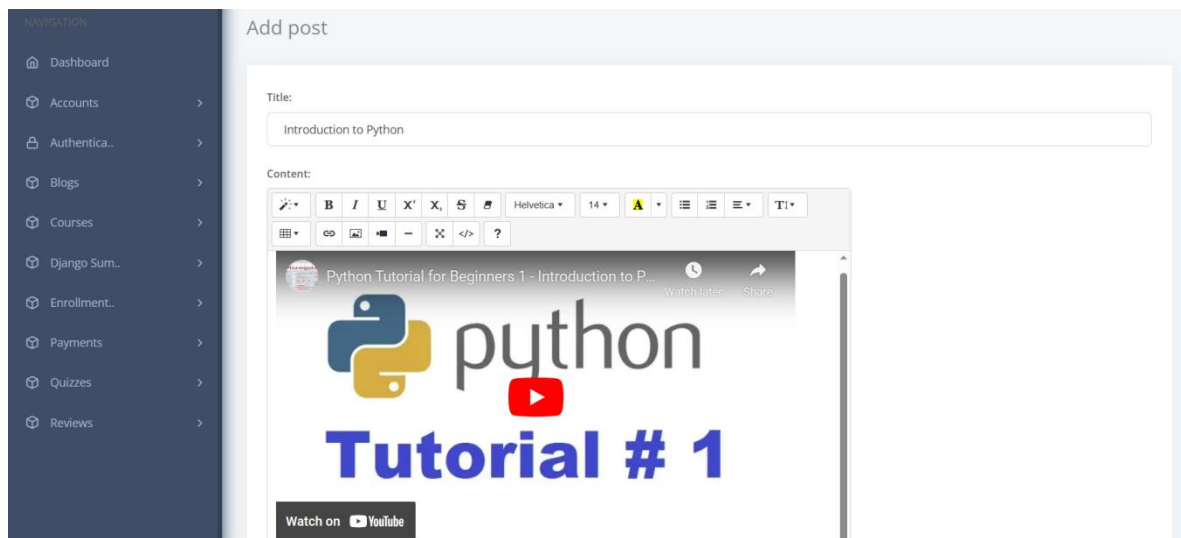


Image 4

Conclusion

In conclusion, creating a relational database design for an e-learning platform involves following several key best practices. These include careful planning to identify the data that will be stored in the database and organize it into tables and columns, normalization to reduce redundancy and improve data integrity, and setting constraints to ensure data integrity by preventing invalid or inconsistent data from being entered into the database.


By following these best practices and applying them in practical ways, developers and designers can create effective and efficient databases that support the goals of their e-learning platforms. This includes supporting specific functionality such as enrolling in courses, taking quizzes, and submitting assignments.

Overall, a well-designed relational database is a crucial component of any successful e-learning platform. By following the best practices outlined in this article, developers can create databases that are robust, reliable, and capable of supporting the needs of their users.

REFERENCES:

1. Hernandez, M. J. (2013). Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design (3rd ed.). Addison-Wesley Professional.
2. Davidson, L. (2021). Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance. Apress.



- 
3. Coursera. (n.d.). What Is a Relational Database and How Does It Work? Retrieved June 6, 2023, from <https://www.coursera.org/articles/relational-database>
 4. ResearchGate. (n.d.). Relational Database Design. Retrieved June 6, 2023, from <https://www.researchgate.net/topic/Relational-Database-Design/publications>
 5. IBM. (n.d.). What is a relational database? Retrieved June 6, 2023, from <https://www.ibm.com/topics/relational-databases>

