

## Batch 3: Module 8 - Project Day

# Quiz App Database Design

## Introduction

The purpose of this document is to outline the database design for a quiz application. The database is designed to manage users (students and teachers), quizzes, questions, options for answers, and student responses efficiently. The following specification details the tables, their columns, relationships, and the overall schema for the application.

---

## Functional Requirements

- User Management:**
    - Allow teachers to create and manage quizzes.
    - Allow students to participate in quizzes.
  - Quiz Management:**
    - Each quiz is associated with a teacher.
    - Each quiz can have multiple questions.
  - Question Management:**
    - Each question belongs to a specific quiz.
    - Each question can have multiple answer options, with at least one correct answer.
  - Option Management:**
    - Define options as possible answers for a question, each linked to its respective question.
    - Specify whether each option is correct or not.
  - Answer Tracking:**
    - Students can attempt a quiz only once.
    - Track student answers to each question in a quiz.
- 

## Database Design

### Tables:

## Users

- **Columns:**
  - `id` (PK): Unique identifier for each user.
  - `name`: Name of the user.
  - `role`: User role, either 'student' or 'teacher'.
  - `created_at`: Timestamp when the user was created.

## Quizzes

- **Columns:**
  - `id` (PK): Unique identifier for each quiz.
  - `title`: Title of the quiz.
  - `teacher_id` (FK from `Users.id`): The ID of the teacher who created the quiz.
  - `created_at`: Timestamp when the quiz was created.

## Questions

- **Columns:**
  - `id` (PK): Unique identifier for each question.
  - `quiz_id` (FK from `Quizzes.id`): The ID of the quiz to which the question belongs.
  - `text`: The text of the question.

## Options

- **Columns:**
  - `id` (PK): Unique identifier for each option.
  - `question_id` (FK from `Questions.id`): The ID of the question to which the option belongs.
  - `text`: The text of the option.
  - `is_correct`: Boolean indicating if the option is correct.

## Student\_Answers

- **Columns:**
  - `id` (PK): Unique identifier for each student's answer.
  - `student_id` (FK from `Users.id`): The ID of the student who answered the question.
  - `quiz_id` (FK from `Quizzes.id`): The ID of the quiz being answered.
  - `question_id` (FK from `Questions.id`): The ID of the question being answered.
  - `option_id` (FK from `Options.id`): The ID of the option chosen by the student.

## Relationships:

1. **Users ↔ Quizzes:** One-to-Many
  - A teacher can create many quizzes.
  - Each quiz is associated with one teacher.
2. **Quizzes ↔ Questions:** One-to-Many
  - A quiz can have many questions.
  - Each question belongs to one quiz.
3. **Questions ↔ Options:** One-to-Many
  - A question can have many options.
  - Each option belongs to one question.
4. **Users ↔ Student\_Answers:** One-to-Many
  - A student can answer many questions.
  - Each answer is associated with one student.
5. **Quizzes ↔ Student\_Answers:** One-to-Many
  - A quiz can have many student answers.
  - Each answer is associated with one quiz.
6. **Questions ↔ Student\_Answers:** One-to-Many
  - A question can appear in multiple student answers.
  - Each answer is associated with one question.
7. **Options ↔ Student\_Answers:** One-to-Many
  - A single option can be chosen by many student answers.
  - Each student answer corresponds to one specific option.

## Quiz App Database Design (ER Diagram)

