Database Design: Online Course System

A guide for a database design for a sample online course system

Ben Brumm

www.databasestar.com

Database Design: Online Courses

This guide is a companion to my YouTube video on designing a database for an Online Course website or Learning Management System (LMS).

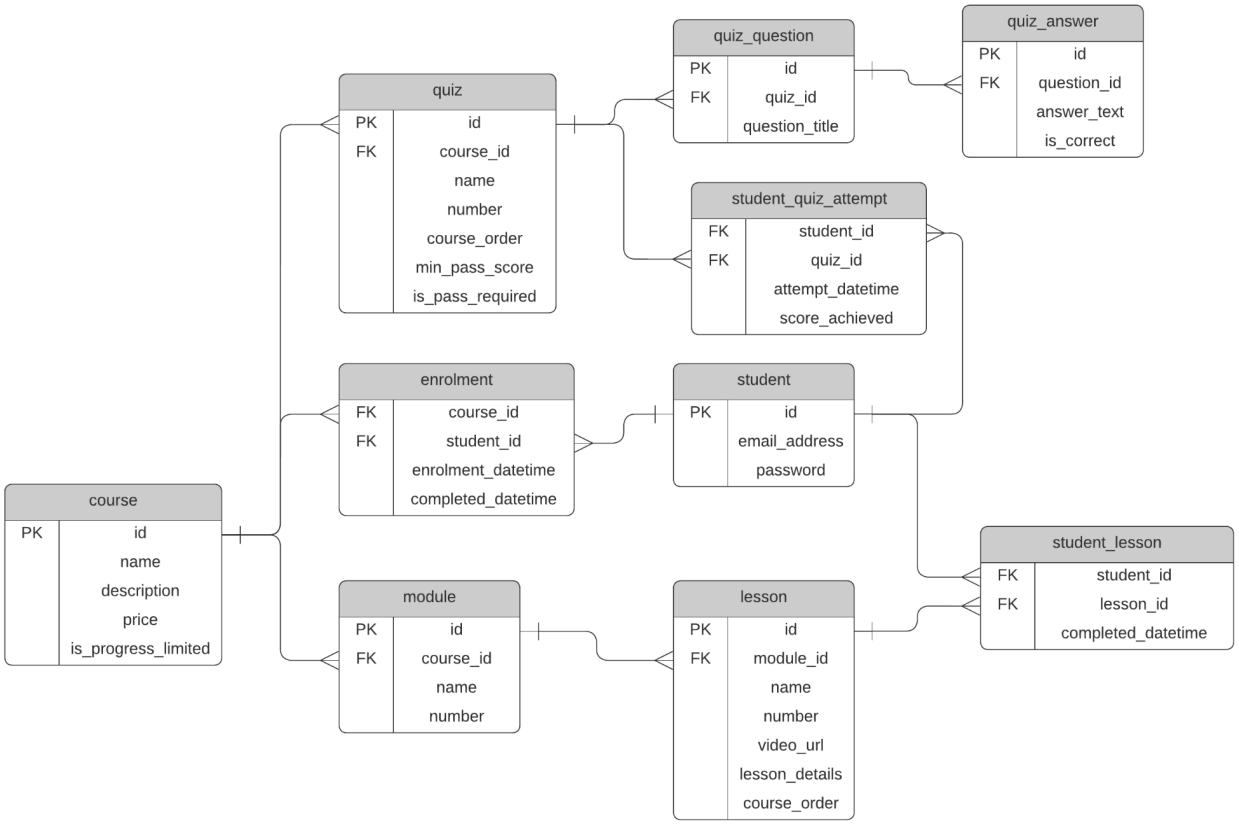
In this guide, you'll see:

- An Entity Relationship Diagram for an Online Course website, from my YouTube video.
- An explanation of the purpose of each table and field, with sample data
- SQL scripts to create each of these tables with some sample data

Let's get into it.

Entity Relationship Diagram

Here's the ERD for this database:



A PNG file of this ERD is available here:

https://dbshostedfiles.s3.us-west-2.amazonaws.com/dbs/erd_online_courses.png

Database Definition

This section explains each of these tables and fields.

course

A course that teaches a specific topic and can be purchased by students.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
name	The name of the course that is being offered	Introduction to Web Development Advanced Data Analysis Modern JavaScript
description	A description of the course, such as what it's about and what's included.	This course will teach you the basics of web development so you can…
price	The price of the course, in USD.	100 25 500
is_progress_limited	A boolean value to indicate if a student needs to complete one lesson before proceeding to the next lesson. False or 0 = not limited, student can watch lessons in any order True or 1 = limited, student must complete a lesson before watching the next lesson	False/0 True/1

module

A smaller part of a course, which contains lessons.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3

course_id	A foreign key to the course table, to indicate the course that the module belongs to.	2, 6, 10
name	The name of the module so the students know what's inside	Introduction Setup Variables
number	The number of the module, which could be displayed on the course page	1, 3, 7

lesson

An individual step within a course that may include a video and a description.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
module_id	A foreign key to the module table to indicate the module that the lesson belongs to.	5, 18, 43
name	The name of the lesson.	Set Up Your Environment Creating a Variable
number	The number of the lesson within the course.	1, 5, 12
video_url	A URL to the video that is shown for this lesson.	https://www.yourserver.com/videos/video123.mp4
lesson_details	A large text value that describes the lesson.	In this lesson you will…
course_order	A number to indicate where in the course that this lesson appears.	1, 50, 85

student

A person who can take courses and watch lessons.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
email_address	The email address of the student, which can be used to login.	johnsmith@abc.com
password	The student's password for logging in.	

enrolment

A combination of students and courses to indicate the courses that students have enrolled in.

Column	Description	Sample Data
course_id	A foreign key to the course table to indicate the related course.	1, 2, 3
student_id	A foreign key to the student table to indicate the related student.	2, 6, 10
enrolment_datetime	The date and time that the student enrolled in the course	2023-08-09 16:21:05
completed_datetime	The date and time that the student completed the course. This would be NULL if the student has not completed the course yet.	2023-09-15 08:45:10

student_lesson

A record of students and the lessons for the courses they have enrolled in, so we can track the completion percentage.

Column	Description	Sample Data
student_id	A foreign key to the student table, to indicate the student that has taken the lesson.	1, 2, 3
lesson_id	A foreign key to the lesson table, to	5, 18, 40

	indicate the lesson.	
completed_datetime	The date and time that this lesson was completed. It would be NULL if the lesson is not completed	2023-09-18 20:21:14

quiz

A record of a quiz within a course, that a student can take to test their knowledge.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
course_id	A foreign key to the course table, to indicate the course that a quiz is related to.	6, 13, 20
name	The name of the quiz	Variables Quiz Basic Java Quiz
number	The number of the quiz within a course.	1, 3, 4
course_order	A number to indicate where in the course that this quiz appears.	3, 10, 19
min_pass_score	The minimum score that a student needs to achieve on a quiz for them to pass	50, 80, 100
is_pass_required	A boolean value to indicate whether the student needs to pass the quiz to proceed.	False/0 True/1

student_quiz_attempt

A record of attempts that a student has made on a course, including their score on the quiz.

Column	Description	Sample Data
student_id	A foreign key to the student table, to indicate which student attempted the quiz.	1, 2, 3

quiz_id	A foreign key to the quiz table, to indicate which quiz the student has attempted.	6, 19, 24
attempt_datetime	The date and time the student attempted the quiz.	2023-09-21 16:12:05
score_achieved	The score that the student achieved.	0, 30, 80

quiz_question

An individual question on a quiz.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
quiz_id	A foreign key to the quiz table, to indicate which quiz the question is for.	1, 7, 18
question_title	The actual question being asked.	Which of the following is true?

quiz_answer

A record of answers that can be selected for a question.

Column	Description	Sample Data
id	Primary key. A unique identifier for the row.	1, 2, 3
question_id	A foreign key to the quiz_question table, to indicate which question this answer is for.	5, 20, 31
answer_text	The answer that is shown to the user	"A variable must be declared"
is_correct	A boolean value to indicate if this answer is correct or incorrect	False or 0 True or 1

SQL Scripts

Here is the SQL code to create the tables for this database.

The script is written for MySQL, but it can easily be modified to work on your preferred database vendor by changing the data types and removing the IF EXISTS (if your database doesn't support it).

```
CREATE DATABASE online_course;

USE online_course;

DROP TABLE IF EXISTS student_quiz_attempt;
DROP TABLE IF EXISTS quiz_answer;
DROP TABLE IF EXISTS quiz_question;
DROP TABLE IF EXISTS quiz;
DROP TABLE IF EXISTS student_lesson;
DROP TABLE IF EXISTS enrolment;
DROP TABLE IF EXISTS student;
DROP TABLE IF EXISTS lesson;
DROP TABLE IF EXISTS module;
DROP TABLE IF EXISTS course;


CREATE TABLE course (
     id INT AUTO_INCREMENT,
    name VARCHAR(500),
    description VARCHAR(4000),
    price INT,
    is_progress_limited TINYINT,
    CONSTRAINT pk_course PRIMARY KEY (id)
);

CREATE TABLE module (
     id INT AUTO_INCREMENT,
    course_id INT,
    name VARCHAR(500),
    number INT,
    CONSTRAINT pk_module PRIMARY KEY (id),
     CONSTRAINT fk_module_course
          FOREIGN KEY (course_id) REFERENCES course (id)
);

CREATE TABLE lesson (
     id INT AUTO_INCREMENT,
    module_id INT,
```

```
    name VARCHAR(500),
    number INT,
    video_url VARCHAR(500),
    lesson_details VARCHAR(4000),
    course_order INT,
    CONSTRAINT pk_lesson PRIMARY KEY (id),
     CONSTRAINT fk_lesson_module
          FOREIGN KEY (module_id) REFERENCES module (id)
);

CREATE TABLE student (
     id INT AUTO_INCREMENT,
    email_address VARCHAR(350),
    password VARCHAR(500),
    CONSTRAINT pk_student PRIMARY KEY (id)
);

CREATE TABLE enrolment (
     course_id INT,
    student_id INT,
    enrolment_datetime DATETIME,
    completed_datetime DATETIME,
    CONSTRAINT fk_enrol_course
          FOREIGN KEY (course_id) REFERENCES course (id),
     CONSTRAINT fk_enrol_student
          FOREIGN KEY (student_id) REFERENCES student (id)
);

CREATE TABLE student_lesson (
    student_id INT,
    lesson_id INT,
    completed_datetime DATETIME,
    CONSTRAINT fk_stdles_student
          FOREIGN KEY (student_id) REFERENCES student (id),
     CONSTRAINT fk_stdles_lesson
          FOREIGN KEY (lesson_id) REFERENCES lesson (id)
);

CREATE TABLE quiz (
     id INT AUTO_INCREMENT,
    course_id INT,
    name VARCHAR(500),
    number INT,
    course_order INT,
    min_pass_score INT,
    is_pass_required TINYINT,
    CONSTRAINT pk_quiz PRIMARY KEY (id),
    CONSTRAINT fk_quiz_course
```

```
          FOREIGN KEY (course_id) REFERENCES course (id)
);

CREATE TABLE quiz_question (
     id INT AUTO_INCREMENT,
    quiz_id INT,
    question_title VARCHAR(500),
    CONSTRAINT pk_quizquest PRIMARY KEY (id),
    CONSTRAINT fk_quizquest_quiz
          FOREIGN KEY (quiz_id) REFERENCES quiz (id)
);

CREATE TABLE quiz_answer (
     id INT AUTO_INCREMENT,
    question_id INT,
    answer_text VARCHAR(500),
    is_correct TINYINT,
    CONSTRAINT pk_quizans PRIMARY KEY (id),
    CONSTRAINT fk_quizans_question
          FOREIGN KEY (question_id) REFERENCES quiz_question (id)
);

CREATE TABLE student_quiz_attempt (
     student_id INT,
    quiz_id INT,
    attempt_datetime DATETIME,
    score_achieved INT,
    CONSTRAINT fk_stdquiz_student
          FOREIGN KEY (student_id) REFERENCES student (id),
     CONSTRAINT fk_stdquiz_quiz
          FOREIGN KEY (quiz_id) REFERENCES quiz (id)
);
```

Conclusion

I hope you found this guide useful. If you have any questions or issues with it, let me know at ben@databasestar.com.

Thanks,

Ben Brumm

www.DatabaseStar.com