Cesar

Instructor’s Name

25 January 2021

Online Learning Platform

CSCI 210 Database Final Project

1. **Overview:**

There are many online learning platforms which provide massive open online courses, also called as MOOCs, which train students in various skills, help them prepare for exams, learn new subjects etc. They usually use various paid/open source platforms like OpenEdX, Moodle, Coursera… etc [1], which are also called as Learning Management System. But for the small businesses, the main problem with these LMS is that they are very expensive to maintain, host, and they are also expensive to customise [2]. For small businesses which only need few features it will be beneficial to have a lightweight platform, and this is exactly the business that we are targeting.

We will be building a database design for a small business which provides online courses to learners, where instructors can register and create new courses which will be sold to students through the online website/application.

2.. **Business Rules:**

Few important business rules that will have major impact on the database designs are as follows:

1. The database will involve tables to store information about the instructors, administrators, learners, courses being offered… etc.
2. Each course can have multiple instructors. These instructor will have write-permissions on the course content, and can modify it as well.
3. Each course will have multiple learning components, and some of them can be free which should be accessible to all the users. There has to be table which will store the information on the component of the courses that are free.
4. The learning component can also have date/time on which they will become accessible to a given user/learner. For example, instructor can set the date on which a given component will be visible/accessible to the student (depending upon their start date for the course).
5. Only registered and paid student can access the courses completely.
6. The access to the course should be limited to 3 login per user, i.e., at a time, for each login id we will only allow one login so that the course material cannot be exploited by multiple person with just a single registration/payment.
7. Each update to the course material should be recorded with the name of the instructor/admin who made the change. This could be stored in a separate table with the date and time of the update.
8. The database will also have to store information about the current progress of the student on each component of the course that they are registered on. This will help the application to resume exactly from where the user ended their last session.

The few important entities in the database would be:

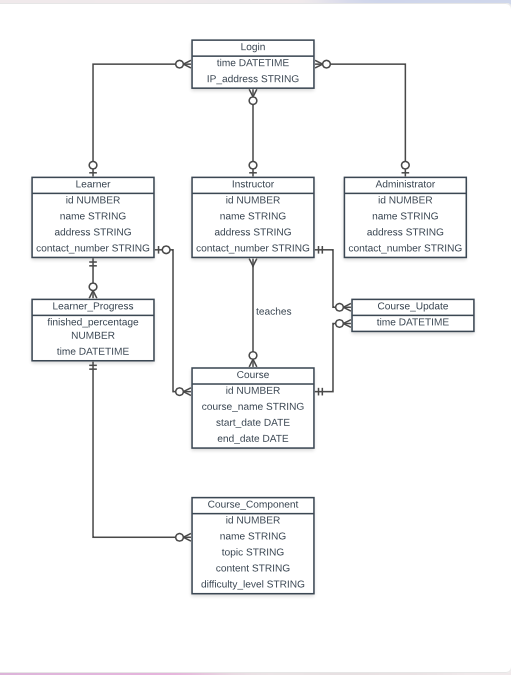
1. Instructor
2. Learner
3. Administrator
4. Course
5. Learner\_Course (learner and course pairs to store the information about registrations)
6. Instructor\_Course (instructor and course pair to store the information about the instructor which have write-permission on the course)
7. Login\_ID\_IP (login id and ip address pair to store information about the logins to limit the number of allowed logins at a time)
8. Course\_Learning\_Component (to store the learning components that constitute given course with the proper ordering information)
9. Learner\_Progress (the table will store the information about the progress of a learner on learning components of the course they are registered on)
10. Course\_Update (the table to store the course update logs)

Attributes for entities are given below (we might add more attributes in the future):

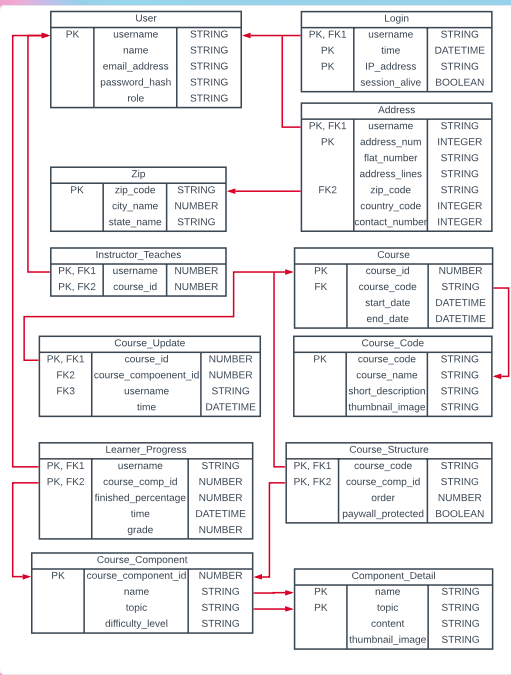
1. **Instructor**a). Id NUMBER  
   b). address STRING  
   c). name STRING   
   d). contact\_number STRING
2. **Learner**a). Id NUMBERb). address STRING  
   c). name STRING   
   d). contact\_number STRING
3. **Administrator**a). Id NUMBERb). address STRING  
   c). name STRING   
   d). contact\_number STRING
4. **Course**a). Id NUMBER  
   b). course\_name STRING  
   c). start\_date DATE  
   d). end\_date DATE
5. ***Login***a). time DATETIME  
   b). IP\_address STRING
6. **Course\_Component**a). name STRING  
   b). topic STRING  
   c). content STRING  
   d). difficulty\_level STRING
7. ***Course\_Update***a). time DATETIME
8. ***Learner\_Progress***a). finished\_percentage NUMBER  
   b). Time DATETIMe

\* The entities with name in *italics* are dependent entities.

**3. ERD:**

****

**4. Relational Diagram:**

****

**5. Keys:**

In this document we identify the candidate keys, primary key, and the foreign keys for each entity.

**In the following table, we list the Candidate Key(s) and Primary Key for each entity.**

|  |  |  |
| --- | --- | --- |
| Entity Name | Candidate Keys | Primary Key |
| User | username  email\_address  contact\_number | username |
| Course | course\_id  (course\_code, start\_date) | course\_id |
| Instructor\_Teaches | (username, course\_id) | (username, course\_id) |
| Course\_Componen | course\_component\_id | course\_component\_id |
| Course\_Structure | (course\_id, course\_comp\_id) | (course\_id, course\_comp\_id) |
| Course\_Update | (course\_component\_id, username, time) | (course\_component\_id, username, time) |
| Course\_Enrolled | (course\_id, username) | (course\_id, username) |
| Learner\_Progress | (username, course\_comp\_id) | (username, course\_comp\_id) |
| Login | (username, time, IP\_address) | (username, time, IP\_address) |

**In the following table, we list the foreign key(s) for each entity (if any) with the name of the entity and its attribute it refers to.**

|  |  |  |
| --- | --- | --- |
| Entity Name | Foreign Key | References |
| User | **None** | |
| Course | **None** | |
| Instructor\_Teaches | username | User(username) |
| course\_id | Course(course\_id) |
| Course\_Component | **None** | |
| Course\_Structure | course\_id | Course(course\_id) |
| course\_comp\_id | Course\_Component(course\_component\_id) |
| Course\_Update | course\_id | Course(course\_id) |
| course\_comp\_id | Course\_Component(course\_component\_id) |
| username | User(username) |
| Course\_Enrolled | course\_id | Course(course\_id) |
| username | User(username) |
| Learner\_Progress | username | User(username) |

**6.**

The integrity rules require the primary key to be not null and unique for each row of the table. We have already listed the primary key for each table in bullet #5. No additional integrity rules are there for this database.

**7.**

**8.**

The DB file is uploaded with the name **“online\_learning\_cesar.db”.**

All the tables are created and few dummy records are inserted too.

**9. SQL for table creation:**

**CREATE** **TABLE** **USER** (

username **VARCHAR**(30)**PRIMARY** **KEY**,

name **VARCHAR**(30),

email\_address **VARCHAR**(60),

password\_hash **VARCHAR**(300),

**ROLE** **VARCHAR**(30));

**CREATE** **TABLE** Course (

course\_id **VARCHAR**(30) **PRIMARY** **KEY**,

course\_code **VARCHAR**(30),

start\_date DATETIME,

end\_date DATETIME,

**FOREIGN** **KEY** (course\_code) **REFERENCES** CourseCode(course\_code)

);

**CREATE** **TABLE** CourseCode (

course\_code **VARCHAR**(30) **PRIMARY** **KEY**,

course\_name **VARCHAR**(30),

short\_description **VARCHAR**(200),

thumbnail\_image **VARCHAR**(200)

);

**CREATE** **TABLE** Login (

username **VARCHAR**(30),

**TIME** DATETIME,

IP\_address **VARCHAR**(30),

session\_alive **BOOLEAN**,

**PRIMARY** **KEY** (username, **TIME**),

**FOREIGN** **KEY** (username) **REFERENCES** **USER**(username)

);

**CREATE** **TABLE** ZipCode (

zip\_code **VARCHAR**(10) **PRIMARY** **KEY**,

city\_name **VARCHAR**(30),

state\_naem **VARCHAR**(30)

);

**CREATE** **TABLE** Address (

username **VARCHAR**(30),

address\_num **INTEGER**,

flat\_number **VARCHAR**(10),

address\_lines **VARCHAR**(30),

country\_code **INTEGER**,

contact\_number **VARCHAR**(10),

zip\_code **VARCHAR**(10),

**PRIMARY** **KEY** (username, address\_num),

**FOREIGN** **KEY** (username) **REFERENCES** **USER**(username),

**FOREIGN** **KEY** (zip\_code) **REFERENCES** ZipCode(zip\_code)

);

**CREATE** **TABLE** Learner\_Progress (

username **VARCHAR**(30),

course\_id **VARCHAR**(30),

finished\_percentage **INTEGER**,

**TIME** DATETIME,

grade **INTEGER**,

**PRIMARY** **KEY** (username, course\_id),

**FOREIGN** **KEY** (username) **REFERENCES** **USER**(username),

**FOREIGN** **KEY** (course\_id) **REFERENCES** Course(course\_id)

);

**CREATE** **TABLE** Course\_Update (

course\_id **VARCHAR**(30),

**TIME** DATETIME,

remark **VARCHAR**(100),

**PRIMARY** **KEY** (course\_id, **TIME**),

**FOREIGN** **KEY** (course\_id) **REFERENCES** Course(course\_id)

);

**CREATE** **TABLE** CourseComponent(

course\_comp\_id **VARCHAR**(30) **PRIMARY** **KEY**,

name **VARCHAR**(30),

topic **VARCHAR**(30),

difficutly\_level **VARCHAR**(30),

content TEXT,

isExternalTool **BOOLEAN**

);

**CREATE** **TABLE** Course\_Structure (

course\_code **VARCHAR**(30),

course\_comp\_id **VARCHAR**(30),

index\_number **INTEGER**,

paywall\_protected **BOOLEAN**,

**PRIMARY** **KEY** (course\_code, course\_comp\_id),

**FOREIGN** **KEY** (course\_code) **REFERENCES** CourseCode(course\_code),

**FOREIGN** **KEY** (course\_comp\_id) **REFERENCES** CourseComponent(course\_comp\_id)

);

**CREATE** **TABLE** LTI\_Tool(

lti\_tool\_id **VARCHAR**(30) **PRIMARY** **KEY**,

tool\_name **VARCHAR**(30),

lti\_username **VARCHAR**(30),

lti\_password **VARCHAR**(30),

url **VARCHAR**(200)

);

**CREATE** **TABLE** LTI\_Tool\_Component (

unique\_id **VARCHAR**(30) **PRIMARY** **KEY**,

window\_width **INTEGER**,

window\_height **INTEGER**

);

**10.** The SQL statements used to assign key to tables are highlighted in bullet #8 with yellow color.

1. The SQL statements used to establish foreign keys are highlighted in bullet #8 with green color.
2. The SQL statements for populating the tables are given below:

**INSERT** **INTO** **USER**

**VALUES**

("alex", "Alex", "alex@example.com", "123QWERTY", "Student"),

("bob", "Bob", "bob@example.com", "123ASDF", "Student"),

("cara", "Cara", "cara@example.com", "123YUIP", "Teacher"),

("marie", "Marie", "marie@example.com", "123QZCV", "Teacher"),

("john", "John", "john@example.com", "12XVXCX", "Student");

**INSERT** **INTO** CourseCode

**VALUES**

("CS101", "Computer Science Foundation", "This is a simple course.", "cs101.jpeg"),

("CS201", "Automata Theory", "This is a simple course.", "cs201.jpeg"),

("CS301", "Complexity Theory", "This is a simple course.", "cs301.jpeg"),

("CS401","Model Checking", "This is a simple course.", "cs401.jpeg"),

("CS501", "Timed Automata", "This is a simple course.", "cs501.jpeg");

**INSERT** **INTO** Course

**VALUES**

("CS101\_2021", "CS101", "01-08-2021", "30-12-2021"),

("CS201\_2021", "CS201", "01-08-2021", "30-12-2021"),

("CS301\_2021", "CS301", "01-08-2021", "30-12-2021"),

("CS401\_2021", "CS401", "01-08-2021", "30-12-2021"),

("CS501\_2021", "CS501", "01-08-2021", "30-12-2021");

**INSERT** **INTO** Login

**VALUES**

("alex", "29-03-2021", "201.196.108.1", 0),

("bob", "29-04-2021", "201.196.108.2", 1),

("cara", "29-05-2021", "201.196.108.3", 0),

("marie", "29-06-2021", "201.196.108.4", 1),

("john", "29-07-2021", "201.196.108.5", 0);

**INSERT** **INTO** ZipCode

**VALUES**

("35618", "Courtland", "Lawrence"),

("35650", "Moulton", "Lawrence"),

("35672", "Town Creek", "Lawrence"),

("35643", "Hillsboro", "Lawrence"),

("35651", "Mount Hope", "Lawrence");

**INSERT** **INTO** Address

**VALUES**

("alex", 1, "100", "City Hall", 1, "989545343", "35618"),

("alex", 2, "101", "Hotel Plaza", 1,"2239545343", "35618"),

("bob", 1, "298", "Mad House", 1,"9555545343", "35643"),

("marie", 1, "210", "Rehab", 1, "444545343", "35650"),

("john", 1, "5942", "Motel Bottle","3439545343", 1, "35651");

**INSERT** **INTO** CourseComponent

**VALUES**

("1", "Automata Definition","Automata", "1", "Automata is easy.", 0),

("2", "Product Automata","Automata", "1", "Product Automata is easy.", 0),

("3", "Timed Automata","Automata", "1", "Timed Automata is easy.", 0),

("4", "Cost Register Automata","Automata", "1", "Cost Register Automata is easy.", 0),

("5", "Tree Automata", "1","Automata", "Tree Automata is easy.", 0);

**INSERT** **INTO** Course\_Structure

**VALUES**

("CS101", "1", 1, 0),

("CS101", "2", 2, 0),

("CS101", "3", 3, 0),

("CS101", "4", 4, 0),

("CS101", "5", 5, 0);

**INSERT** **INTO** LTI\_Tool

**VALUES**

("1", "Quizzes", "hit", "me", "https://quizzeslti.com"),

("2", "Fill", "hit", "me", "https://fillslti.com"),

("3", "Matchings", "hit", "me", "https://matchingslti.com"),

("4", "Vocab", "hit", "me", "https://vocablti.com"),

("5", "MathTool", "hit", "me", "https://mathtoollti.com");

**INSERT** **INTO** LTI\_Tool\_Component

**VALUES**

("1", "1920", "1024"),

("2", "1920", "1024"),

("3", "1920", "1024"),

("4", "1920", "1024"),

("5", "1920", "1024");

**INSERT** **INTO** Learner\_Progress

**VALUES**

("alex", "CS101\_2021", 95, "21-09-2021", 80),

("alex", "CS201\_2021", 80, "21-09-2021", 60),

("alex", "CS301\_2021", 85, "21-09-2021", 95),

("alex", "CS401\_2021", 75, "21-09-2021", 67),

("alex", "CS501\_2021", 99, "21-09-2021", 98);