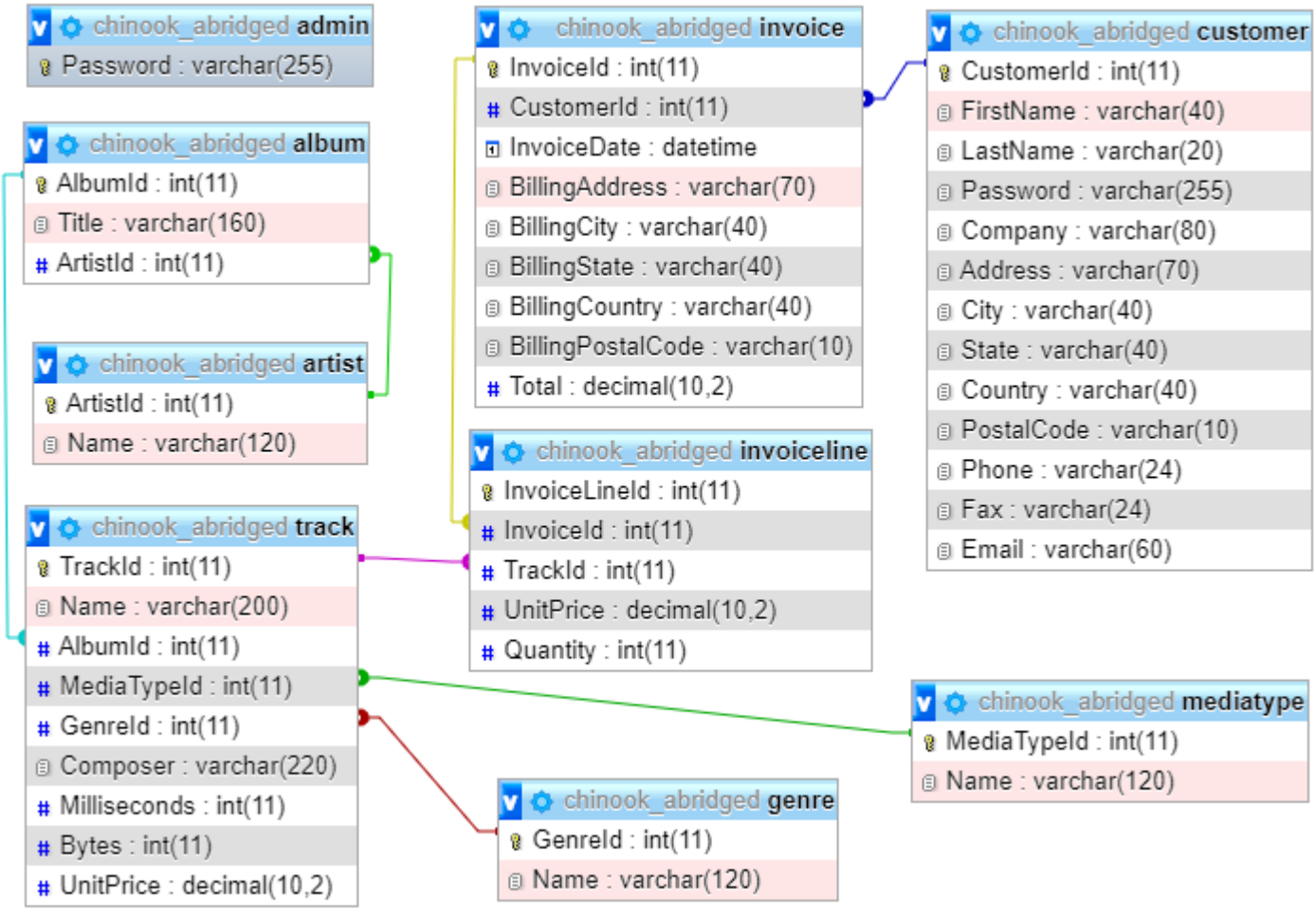
Second Mandatory Assignment

Cloud-Deployed Full-Stack Secure (Responsive) Web Application with Session Authentication and a RESTful API

Description and Tasks

Create a CRUD web application to manage an abridged adaptation of the *Chinook* database, which contains information about online sales of music (the original *Chinook* is at https://github.com/lerocha/chinook-database):



The following tables will be treated as lookup tables without maintenance in the application:

- genre
- mediatype

The application will implement two different logins:

- An administrative login whose hashed password is in the ‘admin’ table (it is ‘admin’). The administrator will be able to do the following:
    - Create, read, update and delete artists, albums, and tracks
- A user login based on the customer table. The hashed password of all existing customers is ‘customer’. Users will be able to do the following:
    - Browse tracks, albums and artists
    - Buy tracks, with the possibility of selecting several tracks into a cart for later purchase.
        - When the user decides to make the purchase effective, an invoice is automatically generated with the non-editable total monetary amount automatically calculated
        - The customer can introduce a billing address that differs from the customer address, but the customer address will be offered as default billing address
    - Sign up, so that they are created as a new customer
    - Edit their own customer data, including changing their password

Please note that, due to referential integrity constraints in the database, it is not possible to delete customers with current purchases, or tracks that have been purchased, or artists with albums. Make sure that your application deals with these situations properly.

The application’s architecture will be divided into a backend that will serve database information in a RESTful API and a frontend that will consume it via Ajax. PUT requests can be managed via POST, if the student so wishes.

The application will implement security measures against SQL injection, XSS and CSRF.

The application and the database will be deployed to the student’s AWS Educate account.

Tools allowed:

- HTML5, CSS3, JavaScript, jQuery, Ajax, PHP, MySQL/MariaDB
- The Bootstrap CSS/JS framework is allowed. However, since Bootstrap enforces the application responsiveness and simplifies enormously the use of modals, if the student decides to use Bootstrap, the application must be fully responsive
- To communicate with the database, the student can decide whether to use PDO or MySQLi

Delivery and Exam

The assignment is individual.

The deliverable will be a zip file containing:

- All the code for the application, including external libraries (e.g., jQuery, Bootstrap)
- The API's documentation
- A text file including the link to the running AWS application

The zip file will be delivered to Fronter's assignment folder **Resources\- Final Mandatory Assignment\Please upload here your Final Mandatory Assignment** no later than 18 December 2020, 12:00h (noon).

The student may continue working on the application after the delivery date, if s/he so wishes.

Each student will use a maximum of 10 minutes to present his/her final application at the subject's exam. The presentation will include:

1. Showing that the cloud-deployed application is online and functional
2. A walkthrough of the application (a local version of the application can be used)
3. A structural explanation of the code

The rest of the exam will be led by the examiners, and will consist of the following:

1. Questions about the application
2. Live coding within the application
3. Questions about the subject's contents
4. Live coding from scratch

The student will leave the exam's room for the final 5 minutes, which will be used for grading.

The application will not be graded per se. **The grade will be solely assigned based on the student's performance at the exam**. However, if the application is incomplete, the student will be likely asked to complete it at the exam.

This information is not definitive, therefore subject to change