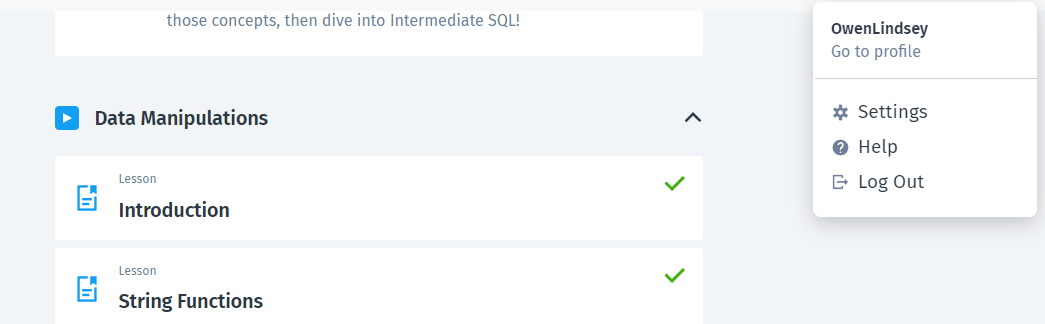
Owen Lindsey

CST-345

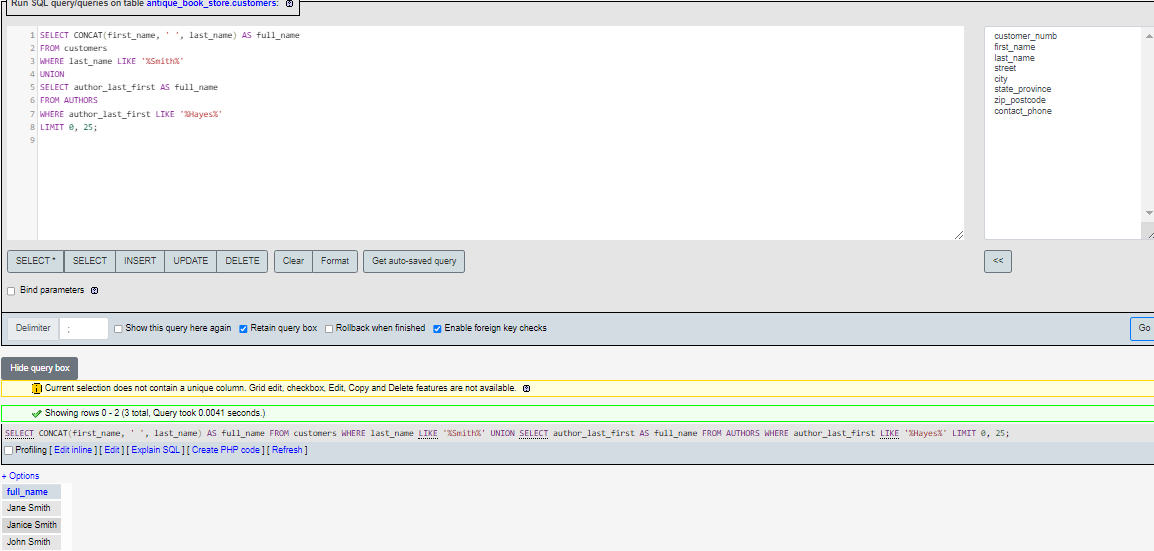
Professor Sluiter

Activity 3

**Part 1: Sololearn deliverables.**

****

**Part 2: Part 2 – Database Design Concepts**

1. **Union statement that uses two sets of user data and joins them in a single table.**
2. **an SQL statement to select all columns from the customers table. Exclude all customers who conducted a sale after August 1, 2021**

**SQL statement:**

**SELECT \***

**FROM customers c**

**WHERE NOT EXISTS (**

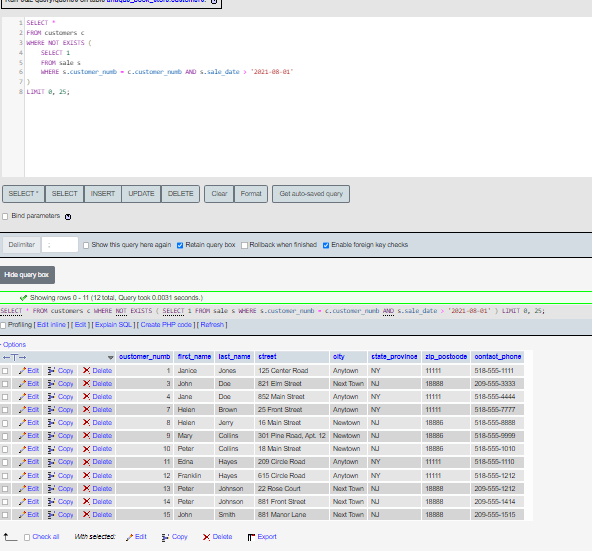
**SELECT 1**

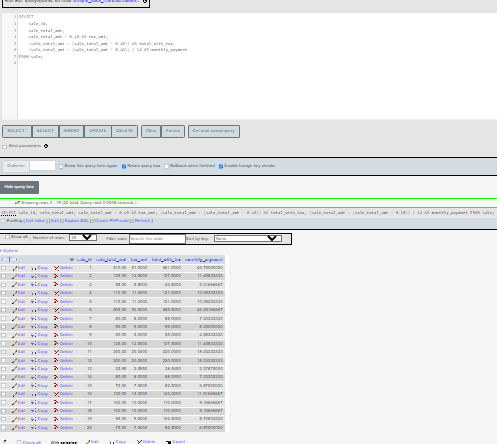
**FROM sale s**

**WHERE s.customer\_numb = c.customer\_numb AND s.sale\_date > '2021-08-01'**

**)**

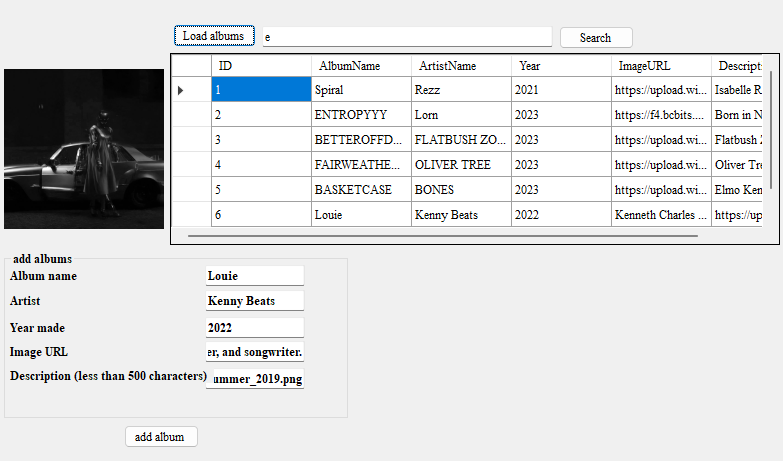
**LIMIT 0, 25**

**Output:**

**C.) Write a SQL statement to select sale\_id and sale\_total\_amt from the sale table. Calculate a 10% tax on each sale.**

**Part 3: Prepare SQL Statements for the Music App**

1. **This screenshot contains all the requirements for the deliverables section of part 3.**

****

**B.) Summary of music app**

**In this lesson, we delved into creating a music application that interfaces with a database, emphasizing integrating a relational database management system using MySQL with a C# application. Utilizing the `MySql.Data` library, we crafted a structured data access layer through the `AlbumDAO` class, enabling functionalities such as retrieving all albums, adding new ones, and performing specific searches based on album titles. The application also demonstrated how to bind data to UI components, specifically the `DataGridView`, for a user-friendly presentation of the data. Additionally, best practices like parameterized queries were employed to prevent SQL injection attacks. Essential error-handling techniques, such as ensuring safe database connections and managing data insertion outcomes, were encapsulated in the code, ensuring robustness. Through this exercise, learners gained a holistic understanding of connecting front-end interfaces with backend databases in the context of a real-world application.**