## 

## 

## 

## 







## **PreLab**



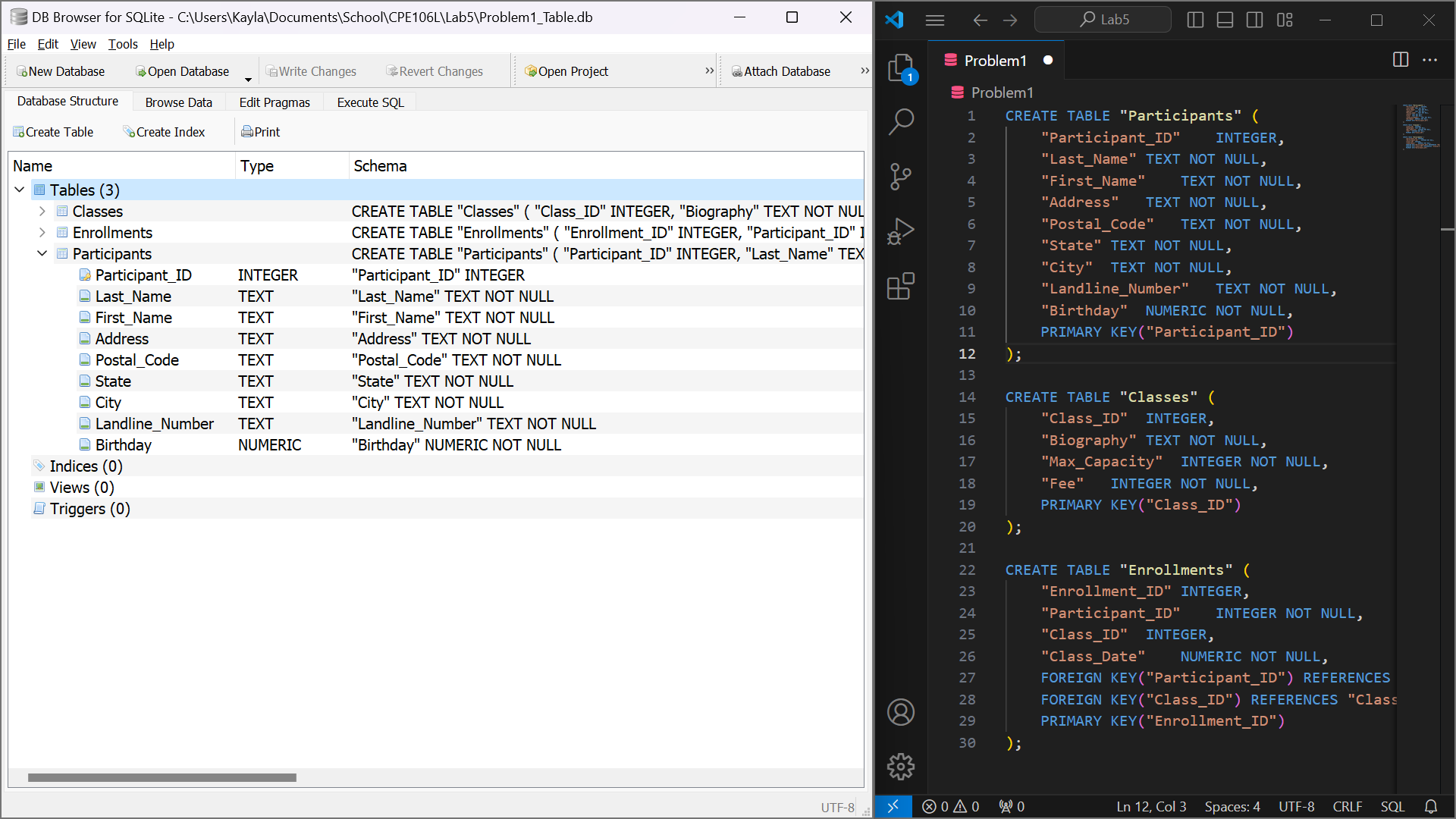
| **Readings, Insights, and Reflection**  **A Guide to SQL 9780357419830**  **Chapter 1:**  [Yapcengco]: *In this chapter, we explore the practical applications of databases in real-world scenarios.By examining specific databases, we gain insights into their structures and relationships, enabling efficient organization, viewing, and updating of information. Businesses across various industries utilize data management techniques based on these structures.*  [Zalameda]: *I gained knowledge about the basic idea of a database in this chapter. I learned that a database is an ordered set of structured data that has been arranged for effective storing, retrieval, and manipulation. I also learned more about the many kinds of databases, including NoSQL databases, which grow horizontally and provide flexibility in handling unstructured data, and relational databases, which manage data using tables and SQL. Building reliable applications and systems that need effective data management and retrieval requires an understanding of databases.*  **Chapter 2:**  [Yapcengco]: *In this chapter, we gained familiarity with fundamental database concepts and explored their interrelationships. Specifically, we delved into relational databases, entities, attributes, and relationships. Additionally, we discussed the concept of normalization, which helps us identify and address issues like redundancy and data duplication.*  [Zalameda]: *This chapter introduced me to the idea of normalization as well as other crucial database topics. I found that developing effective databases depends heavily on database principles like entities, properties, relationships, and keys. I also learned more about normalization, which is the act of arranging data to reduce dependencies and redundancies, enhancing data integrity and lowering anomalies.*  **Python Projects**  **9781118908891**  **Chapter 3:**  [Yapcengco]: *In this chapter, we explored the process of storing data in databases and using Python to interact with those databases through SQL queries and scripts. We gained a thorough understanding of relational databases and practiced executing read queries on them.*  [Zalameda]: *This chapter taught me how to use Python for data analysis and SQL for data management. I learned how well SQL can manage operations like building databases, specifying tables, and utilizing DML and DDL statements to manipulate data. I also learned how Python's data analysis packages, such as pandas and NumPy, enhance SQL's data management skills by offering strong tools for processing, visualizing, and extracting insights from big datasets.*    **Answers to Questions**   1. DML (Data Manipulation Language) commands are used in Structured Query Language (SQL) to control data inside database objects. Examples include *DELETE*, *UPDATE*, and *SELECT*. However, the structure of database objects is defined using DDL (Data Definition Language) statements. Examples include *ALTER*, *DROP*, and *CREATE*. DDL statements are used to define the structure and layout of tables and other database objects, whereas DML commands are used to manipulate the data contained in tables. 2. Scalar, Aggregate, and Window functions are the three primary categories into which SQLite functions fall. Each time they are called, scalar functions act on discrete values and deliver a single result. After operating on a set of values, aggregate functions produce a single output that summarizes the set. Last but not least, Window Functions compute a value over a range of table rows that are connected to the current row. 3. Use the command "sqlite3 --version" in the terminal to see if SQLite is installed on your Linux machine. If SQLite is installed on your machine, running sqlite3 --version in the terminal will either reveal the version number of SQLite or an error message stating that the command could not be found.  **InLab**    * **Objectives**  1. Familiarize with creating tables in DB Browser. 2. Familiarize the ‘SELECT’ functions on the ‘cassel’ databases in DB Browser.  * **Tools Used**   + Visual Studio Code   + SQLite   + DB Browser * **Procedure**   In Figure l.1, The DB Browser table creation process is shown in the screenshot. The interface displays a table creation dialog box where you may define the columns of the table and configure the name, data type, and any constraints (primary key, not null, etc.) for the table. The screenshot also shows how to add a row of data into the newly made table and how to enter values into the designated columns straight from the DB Browser graphical user interface.      ***Figure I.1*** *Creating a table in DB Browser and inserting data.*  ***Figure I.2.1*** *Getting to know the DB Browser's "SELECT" function with constraints.*  ***Figure I.2****.****2*** *Using the DB Browser, try the "SELECT" function in "cassel" both with and without constraints.*  In Figures I.2.1 and l.2.2, The screenshot shows how to use the SELECT function in DB Browser on the "cassel" database. It presents two cases: one in which SELECT is used without restrictions, showing all of the data in a given table or set of columns, and another in which SELECT is used with restrictions, such as WHERE clauses, filtering the results according to predefined criteria like matching particular values or ranges. |
| --- |

## **PostLab**



**Programming Problems**

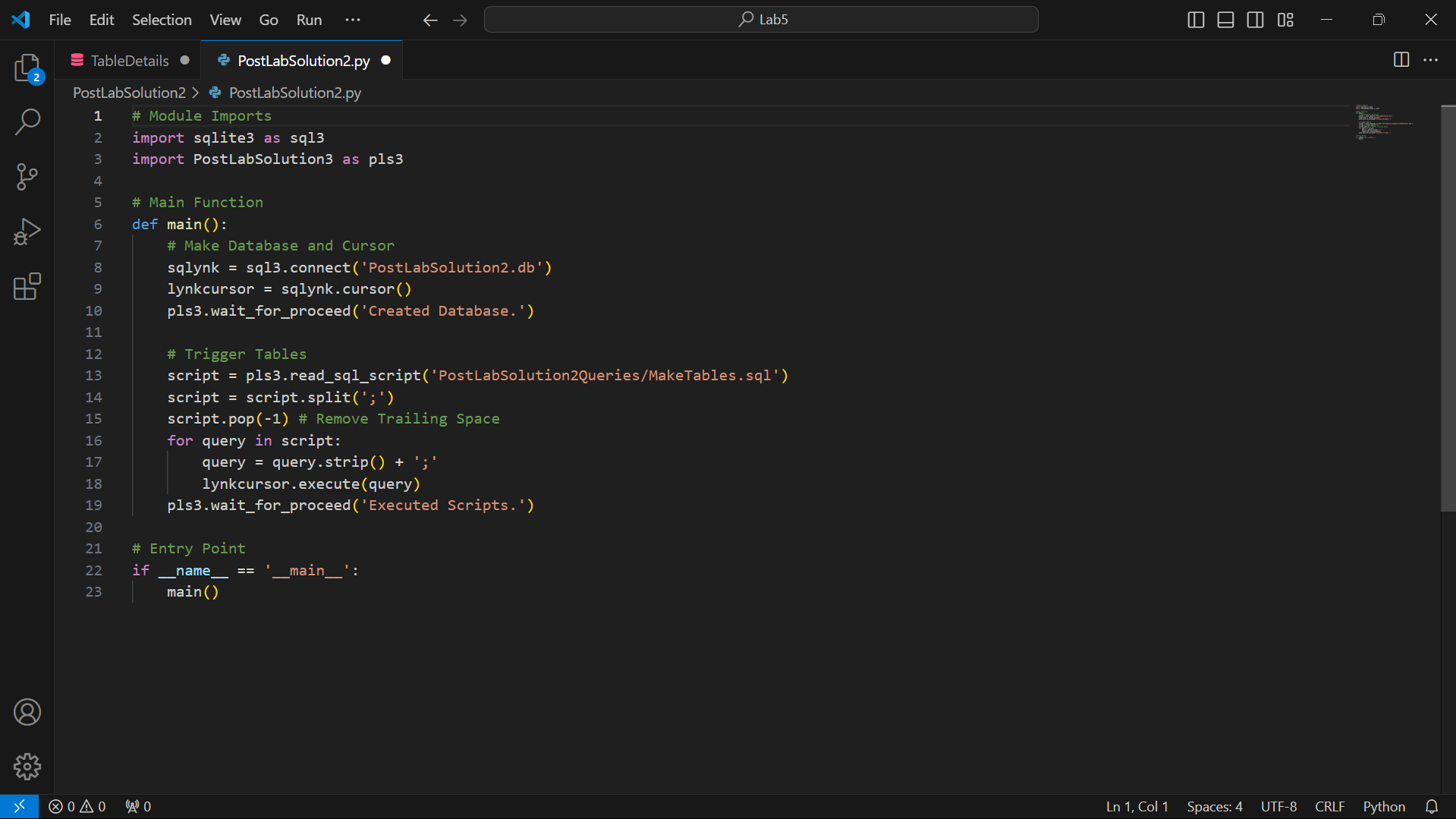
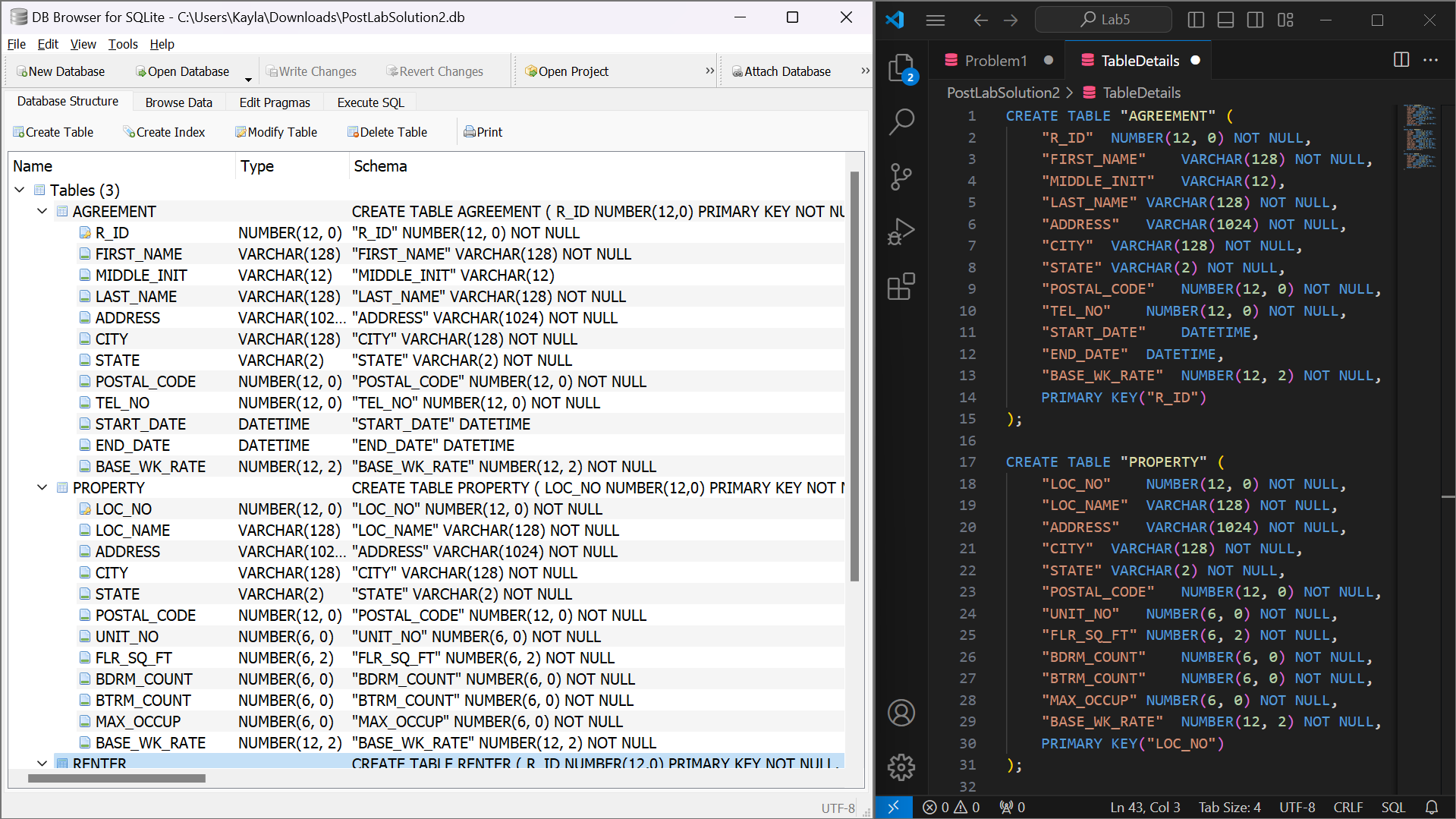
1. Colonial Adventure Tours is considering offering outdoor adventure classes to prepare people to participate in hiking, biking, and paddling adventures. Only one class is taught on any given day. Participants can enroll in one or more classes. Classes are taught by the guides that Colonial Adventure employs. Participants do not know who the instructor for a particular class will be until the day of the class. Colonial Adventure Tours needs your help with the database design for this new venture. In each step, represent your answer using the shorthand representation and a diagram. Use crow’s foot notation for the diagram. Follow the sample SQLite chinook database ERD (Download it from Blackboard Course Materials)
   1. For each participant, list his or her number, last name, first name, address, city, state, postal code, telephone number, and date of birth.
   2. For each adventure class, list the class number, class description, maximum number of people in the class, and class fee.
   3. For each participant, list his or her number, last name, first name, and the class number, class description, and date of the class for each class in which the participant is enrolled.
   4. For each class, list the class date, class number, and class description; and the number, last name, and first name of each participant in the class.



***Figure P.2****. Proposed Database design from the first problem requirements.*

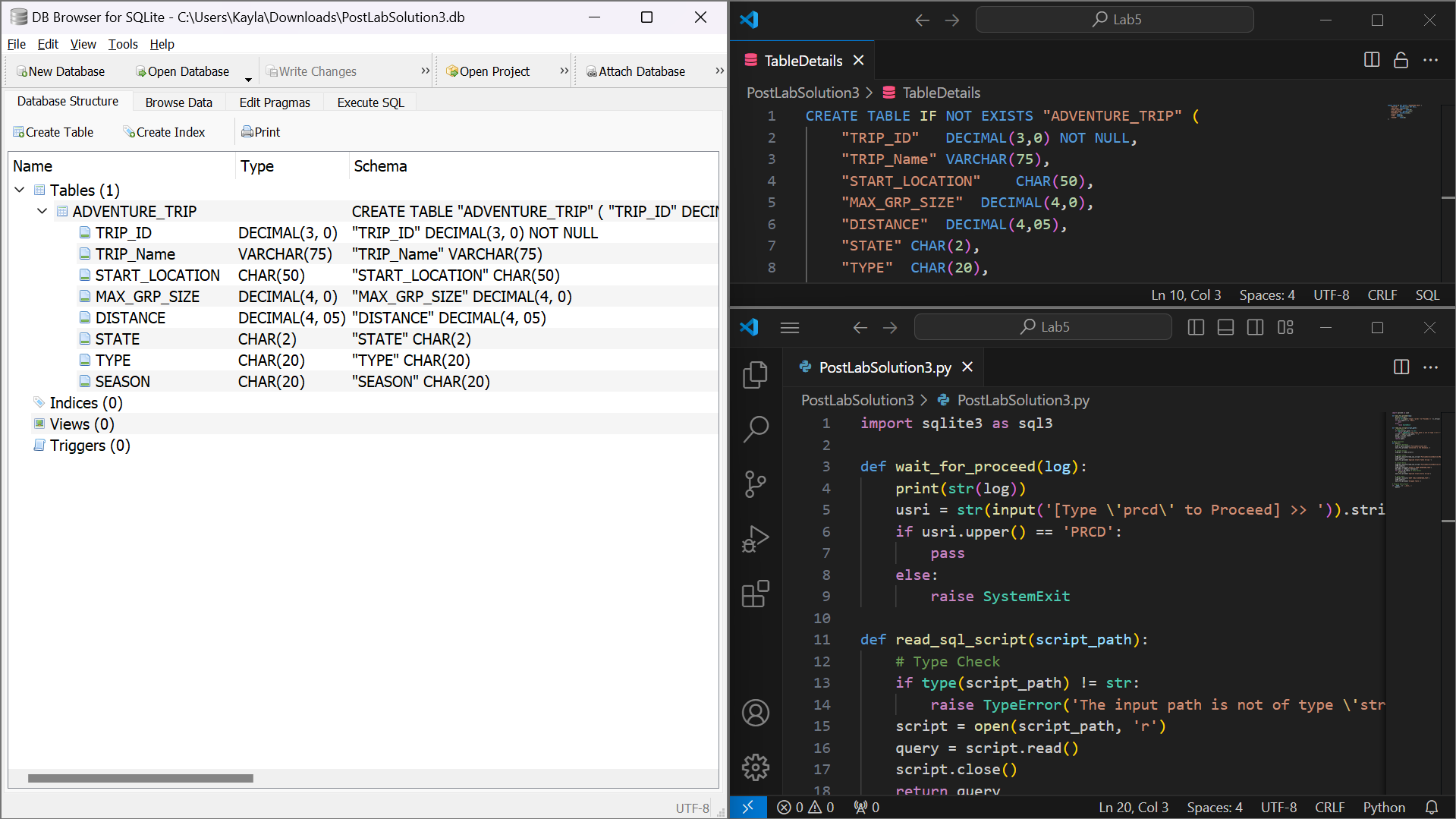
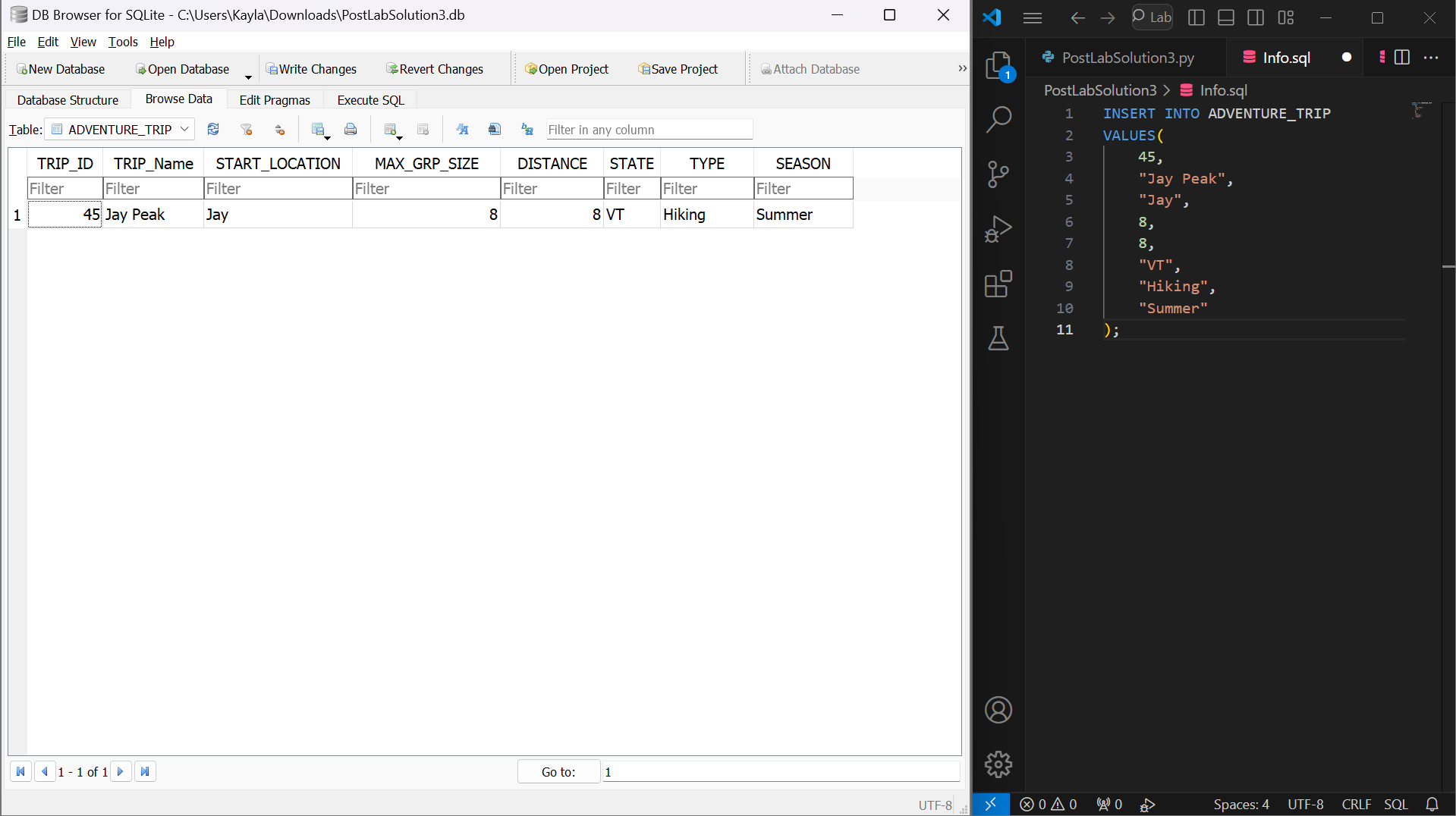
In figure P.2, A relational database system's request for detailed information is shown in the screenshot, which shows a database query being run in DB Browser for SQLite.

1. Solmaris Condominium Group has many condos that are available as weekly vacation rentals. Design a database to meet the following requirements:
   1. For each renter, list his or her number, first name, middle initial, last name, address, city, state, postal code, telephone number, and email address.
   2. For each property, list the condo location number, condo location name, address, city, state, postal code, condo unit number, square footage, number of bedrooms, number of bathrooms, maximum number of persons that can sleep in the unit, and the base weekly rate.
   3. For each rental agreement, list the renter number, first name, middle initial, last name, address, city, state, postal code, telephone number, start date of the rental, end date of the rental, and the weekly rental amount. The rental period is one or more weeks.

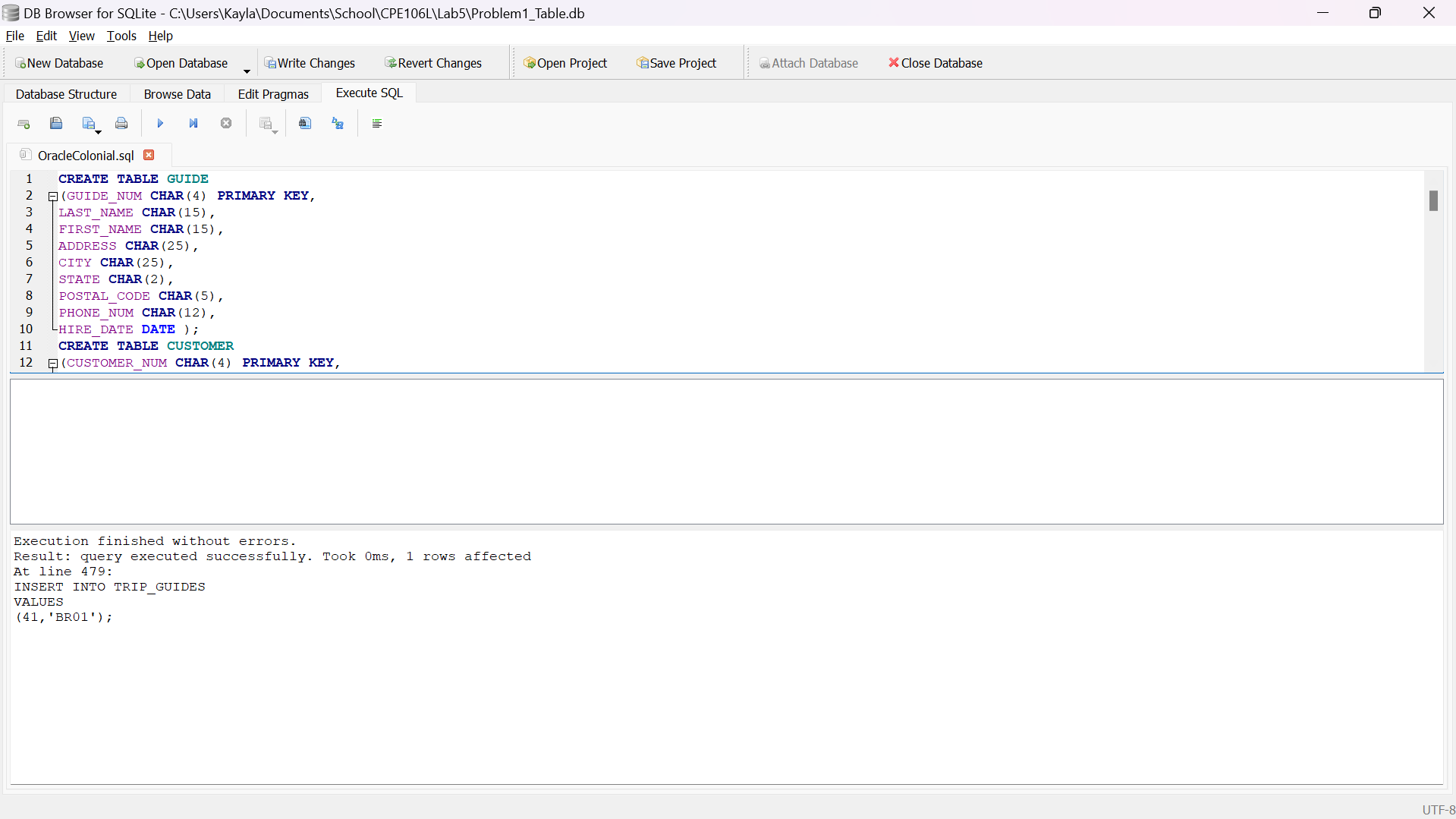
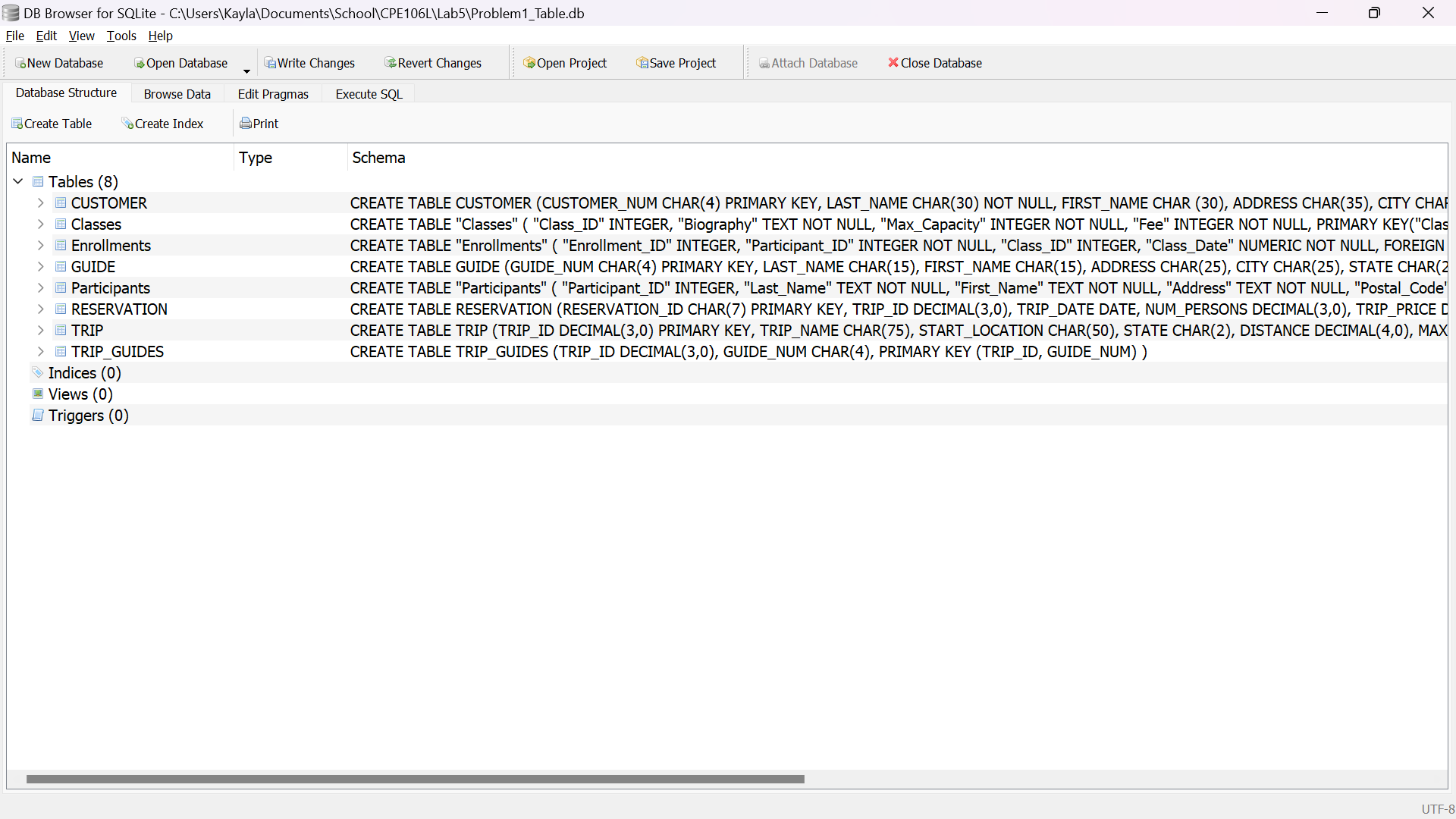
  
***Figure P.2****. Proposed Database design from the second problem requirements.*

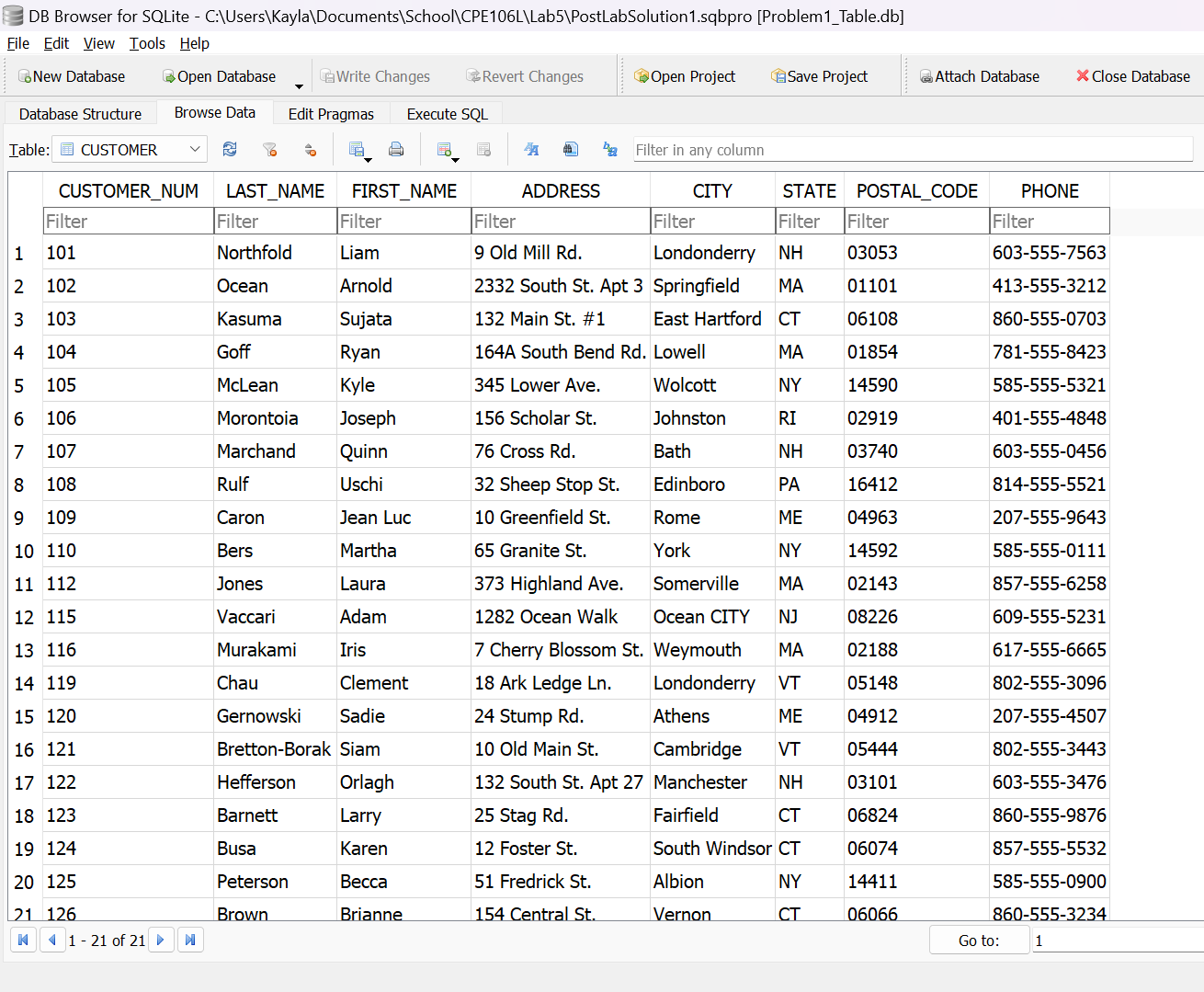
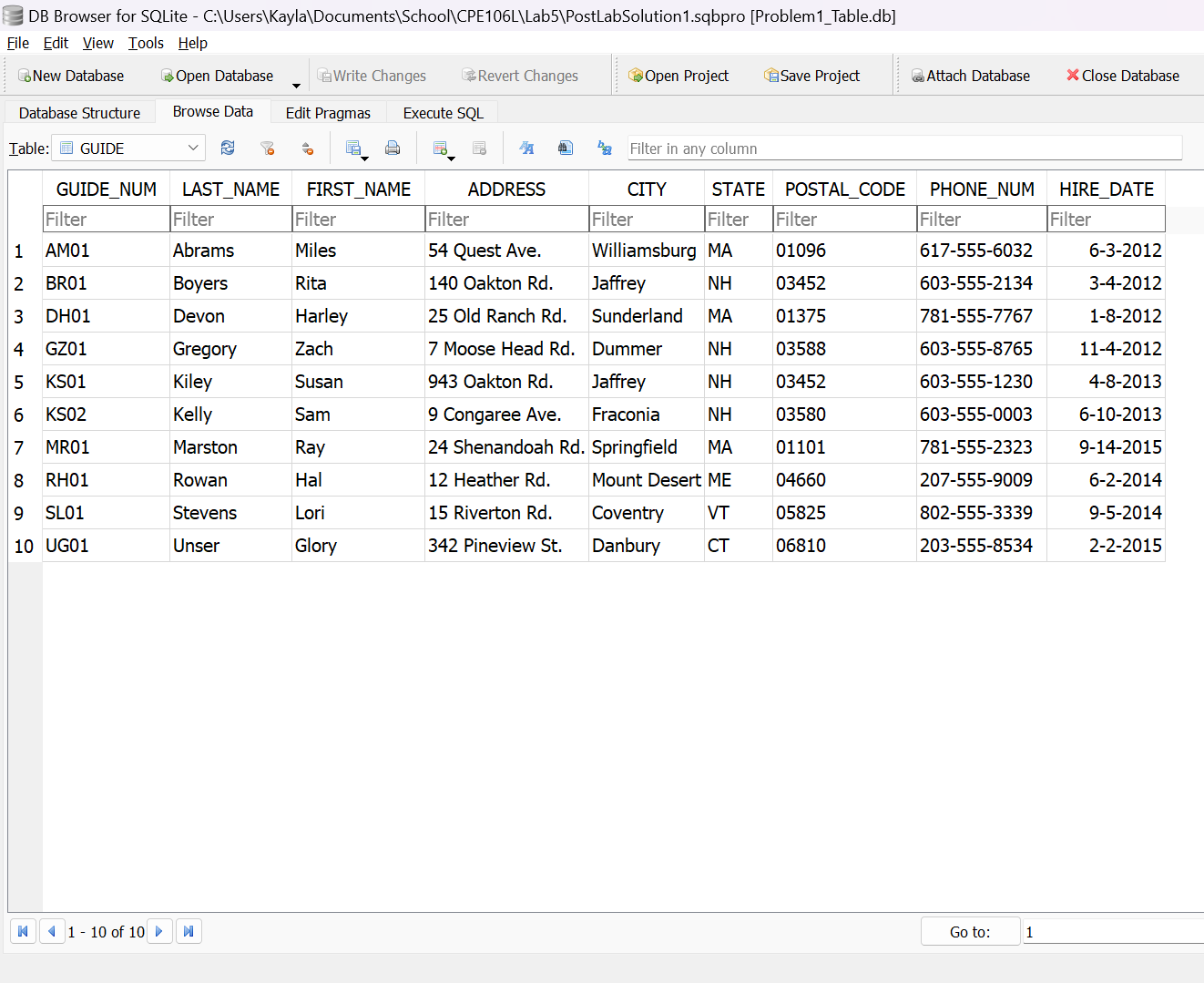
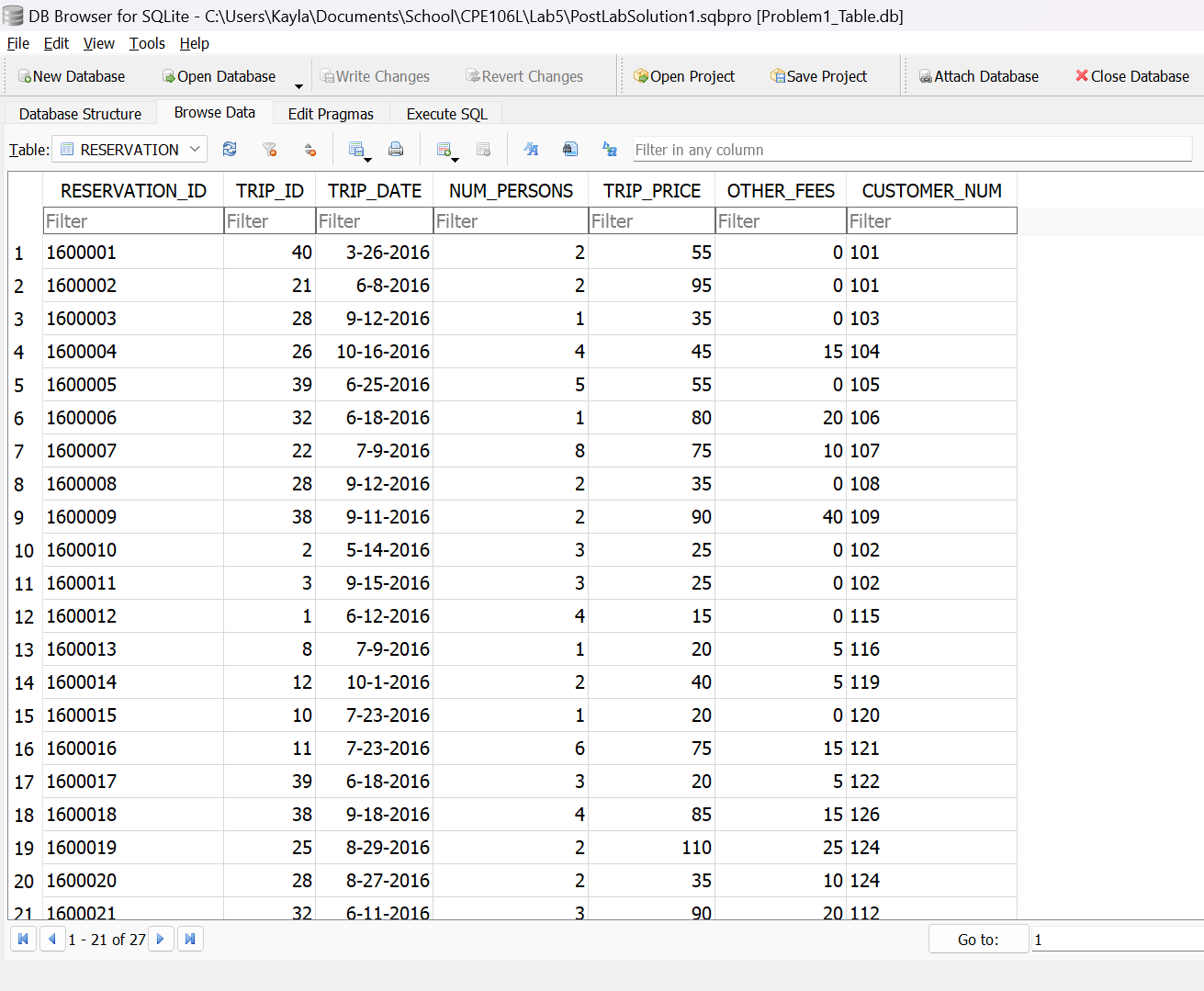
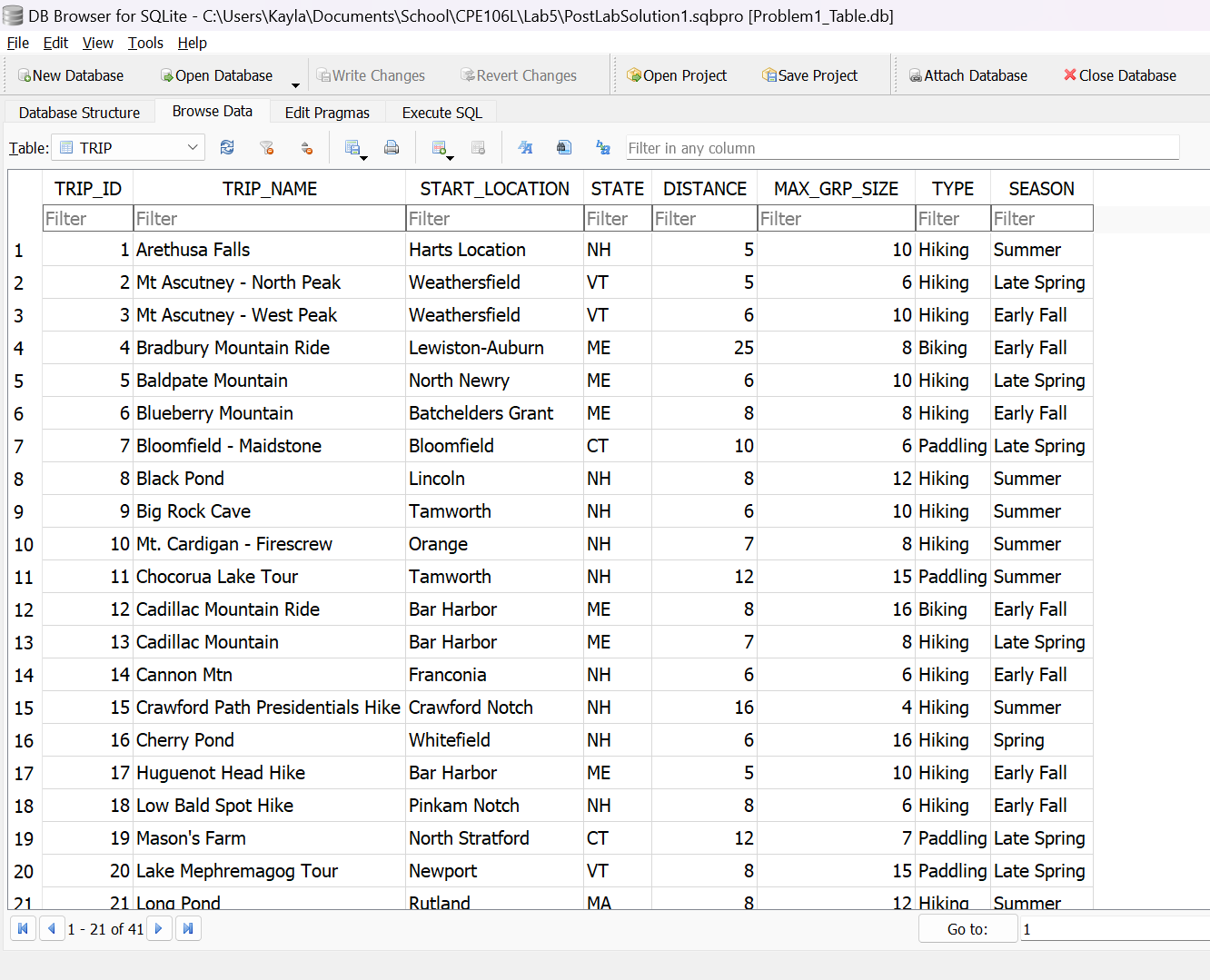
In Figure P.2., The vacation rental system of Solmaris Condominium Group is represented in the snapshot by the database architecture and query execution in DB Browser for SQLite. It displays the construction of tables with specific fields and relationships for renters, properties, and rental agreements.

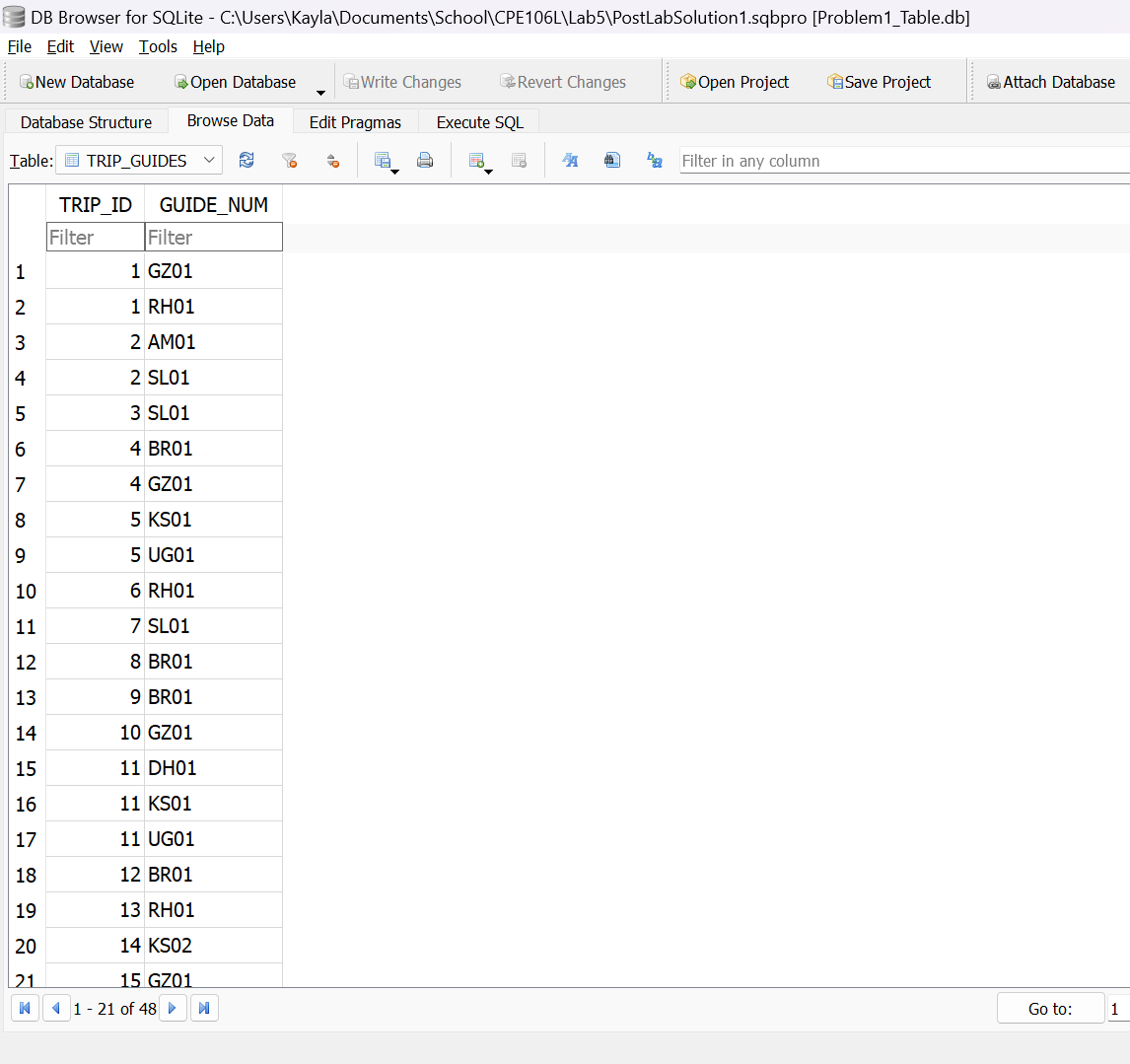
1. Use SQLite commands to complete the following exercises.
   1. Create a table named ADVENTURE\_TRIP. The table has the same structure as the TRIP table shown in Figure 3-2 below except the TRIP\_NAME column should use the VARCHAR data type and the DISTANCE and MAX\_GRP\_SIZE columns should use the NUMBER data type. Execute the command to describe the layout and characteristics of the ADVENTURE\_TRIP table.
   2. Add the following row to the ADVENTURE\_TRIP table: trip ID: 45; trip name: Jay Peak; start location: Jay; state: VT; distance: 8; maximum group size: 8; type: Hiking and sea- son: Summer. Display the contents of the ADVENTURE\_TRIP table.
   3. Delete the ADVENTURE\_TRIP table.
   4. Open the script file (SQLServerColonial.sql) to create the six tables and add records to the tables. Revise the script file so that it can be run in the DB Browser.
   5. Confirm that you have created the tables correctly by describing each table and comparing the results to the figures shown below. Confirm that you have added all data correctly by viewing the data in each table and comparing the results to Figures 1-4 through 1-8 shown below.

   
***Figure P.3.1****. Created a table ‘ADVENTURE\_TRIP’ and the inserted row entrey*

The screen grab, as shown in Figure P.3.1., shows SQL operations being executed in DB Browser for SQLite. It demonstrates the construction of a table called ADVENTURE\_TRIP with unique column data types, the addition of a new adventure trip record, and the use of a SELECT query to confirm the contents of the table. The screenshot also shows sample data being queried to view rental details including start and finish dates and weekly rental amounts associated with individual renters, as well as sample data being placed into these tables to populate renter and property information.

   
***Figure P.3.2*** *Executed “OracleColonial.sql” file and the updated tables.*

  
***Figure P.3.3****. Executed Tables from “OracleColonial.sql” file (Customer, Guide, Reservation, Trip, Trip Guides)*

For both Figure P.3.2. and P.3.3., it provides the output of loading and running pre-written.sql scripts through the DB Browser to create tables and entries in the database.