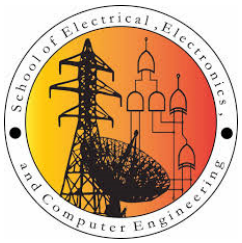
Expt 3.

Monday, February 24, 2020 5:12 PM

Mapua University School Electrical,
Electronics, and Computer Engineering



Experiment 3: Data Modeling and Database Systems

Presented By:

CPE106L-B2-GROUP09

CENTENO, Jarl Kayne Jon
CHUA, Richard Vincent
DOCTO, Jeloux
TAPAGANAO, Fil Janssen

Presented To:

Professor Dionis Padilla

Prelab

A. Readings, Insights, and Reflection

Chapter 1:

1) What is a Database?
    - A database is a systematic collection of data, support storage, retrieval, modification and manipulation of data. Since it is an organized collection of data, it can be easily accessed and managed. Most databases contain multiple tables and each of these tables have a different field that are relevant to the information stored in the table. It can be very useful for a company to have a database for their products on what they sell, tables for the employees or other financial records. It is recommendable to have a database in which the information can be stored and can be easily accessed without any hassle, it will save time and effort with having an organized database.
2) Database requirements of TAL Distributors, Colonial Adventure Tours, and Solmaris Condominium Group
    - Since TAL Distributors is a wholesaler company of finely wooden toys, games and puzzles, and their recent growth cannot be handled properly maintained and organized without a database. With the use of a database, the customers, order and inventory will be surely up-to-date and more accurate than the manual system.
    - Colonial Adventure Tours is a travel agency that requires a database with organized guides, trips, customers, and reservations. The table should contain a specific data of the customer such as, first and last name, address, city, state, postal code, phone number and the additional information for the guide such as the guide number and hire date.
    - This database requires very specific and unique information about the customer, guides, trips, fees, additional fees, reservations etc. to have an organized data storage where every transaction can be easily accessed, managed and manipulated to save time and effort.

- Solmaris Condominium Group manages two condominium complexes, Solmaris Ocean and Solmaris Bayside, which are located in Florida. They also offer maintenance with different amounts of fees, and in order to track and manage each and every operation, a database is required containing the tables with the location number, location name, address, city, state and postal code. It also requires the information about a specific unit such as the owner number, name, address, city, state and postal code. Each database has specific information about its unit, service category and requests.

Chapter 2:

1) Database Concepts
    - Familiarizing some important database concepts is a must before learning how to design a database. A relational database is a collection of data based on the relations of each data to each other. The three fundamentals terms are entity, attribute and relationship. It is important to identify the entity whether it is a person, place, thing or event. Then the attribute is a property for the entity, may include the entity of a customer such as name, address, city and so on. While the relationship is the association between entities.
2) Database Design Fundamentals
    - It is important to determine the purpose of the needed database then find and organize the information required. After getting the needed information, divide it to different tables based on the category and turn that information into columns. Specifying the primary key is also important and sets up the table relationships, applying the normalization rules after refining the design.
3) Normalization
    - Normalization is a database design technique that organizes tables in a manner that reduces redundancy and dependency of data. Normalization divides larger tables into smaller tables and links them using relationships. The purpose of Normalization is to eliminate redundant (useless) data and ensure data is stored

logically. It has an advantage of having a greater overall database organization, reduction of redundant data, data consistency within the database, much more flexible database design, a better handle on database security.

Chapter 24:

1) Types of Databases Used with Python
    - PostgreSQL and MySQL are two of the most common open source databases for storing Python web applications' data. SQLite is a database that is stored in a single file on disk. SQLite is built into Python but is only built for access by a single connection at a time. Therefore it is highly recommended to not run a production web application with SQLite.
    - PostgreSQL is the recommended relational database for working with Python web applications. PostgreSQL's feature set, active development and stability contribute to its usage as the backend for millions of applications live on the Web today. MySQL is another viable open source database implementation for Python applications. MySQL has a slightly easier initial learning curve than PostgreSQL but is not as feature rich.
2) Using MySQL from Python
    - MySQLdb module is needed to work with MySQL in a Python Program. Using mySQL from Python has the advantage of having data security since it is globally renowned for being the most secure and reliable database management system used in popular web applications like WordPress, Drupal, Joomla, Facebook and Twitter. Also, it is high performance with a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for flawless performance.

Chapter 3:

1) Relational Database Concepts

    - In a relational database, all data is held in tables, which are made up of rows and columns. Each table has one or more columns, and each column is assigned a specific data type, such as an integer number, a sequence of characters (for text), or a date. It has data accuracy, easy access to data, data integrity, flexibility, normalization, high security and feasible for future modification.

2) Structured Query Language. DML and DDL SQL commands

    - The difference between DDL and DML is that DDL doesn't have any further classification and has basic commands present such as CREATE, DROP, RENAME, ALTER etc. While DML is further classified into Procedural and Non-Procedural DML and its basic commands are: UPDATE, INSERT, MERGE ETC.

B. ANSWERS TO QUESTIONS

1) What are DML and DDL statements in Structured Query Language?

    - DML (Data Manipulation Language): Update, insert and merge. DDL (Data Defined Language): Create, Alter and Drop.

2) What are the categories of SQLite functions?

    - Numeric/Math Functions
    - Date/Time Functions
    - Advanced Functions

3) How do you check if you have SQLite installed in a system using the Linux Terminal?

- After installing and machine executing the commands, open a terminal and execute “sqlite3”, you will see the following lines with prompt.

```
$ sqlite3
SQLite version 3.8.2 2013-12-06 14:53:30
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
```

InLab

A. Objectives

- Learn to understand and make an UML diagram.
- Learn to modify programs.
- Learn the concept of inheritance.
- Learn the concept of classes.
- Learn the attributes and behavior of a class object.

B. Steps Performed with screenshots of tools used:

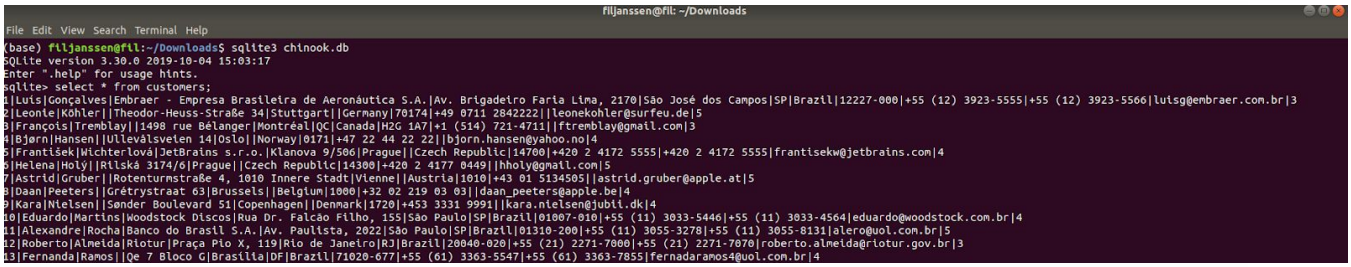


Figure 1 Opening chinook.db

This terminal shows how to open a .db file. It also shows running some syntax such as **select * from customers;**

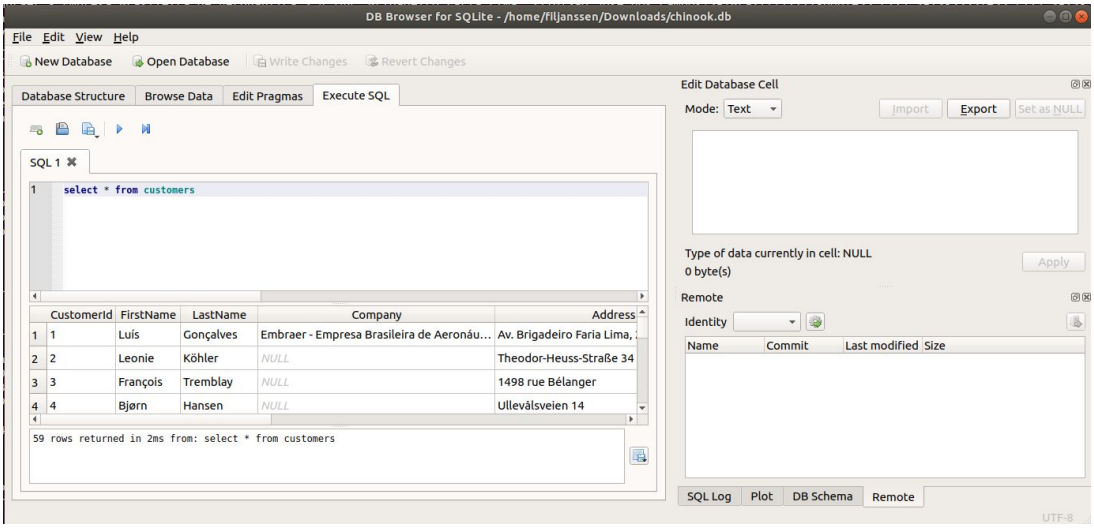


Figure 1.2 Opening chinook.db using DB Browser

This shows the same syntax that was typed in the terminal to show the same results.

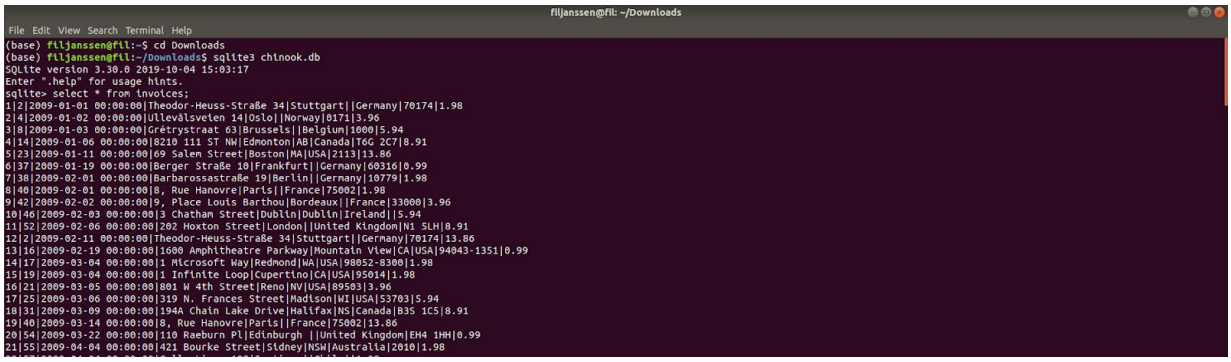


Figure 1.3 Putting other syntax

It shows here the other information inside the chinook.db.

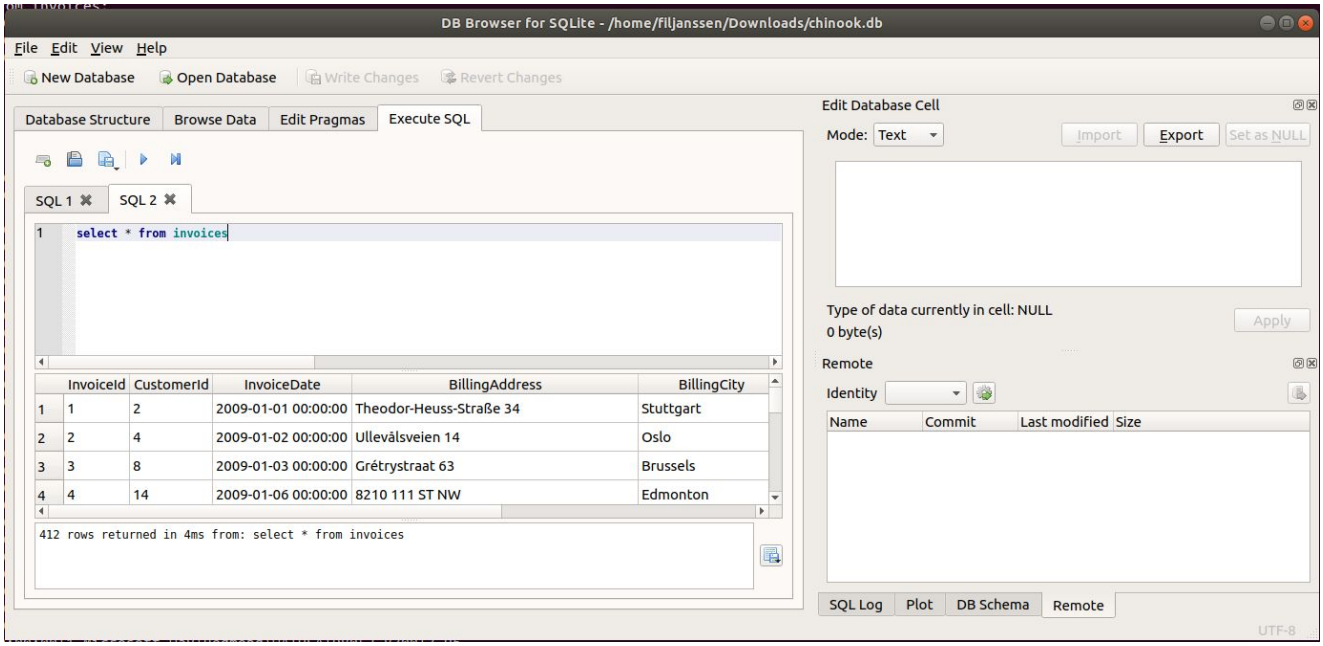
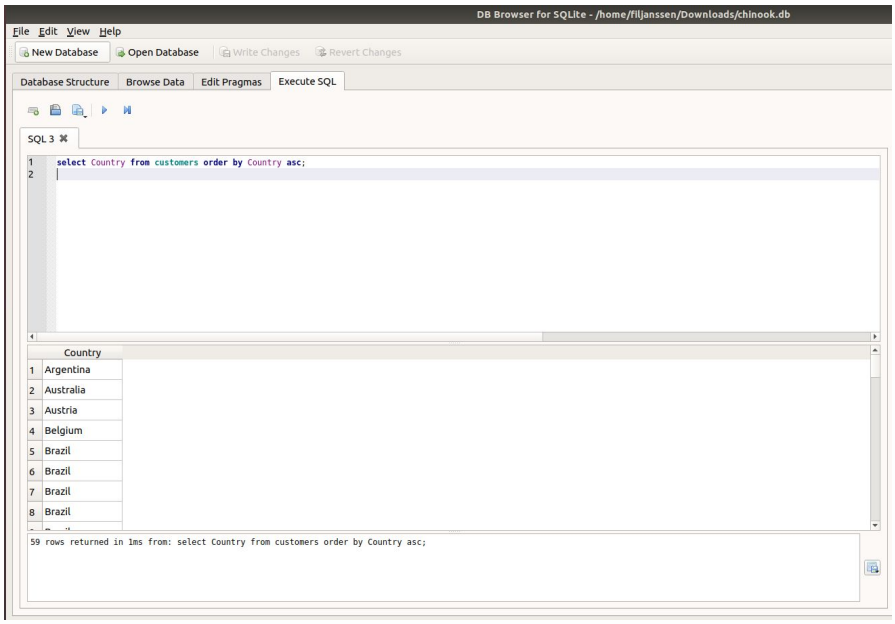


Figure 1.4 Putting the same syntax like in figure 1.3

Here, it is just the same syntax written but on the DB browser.

```
File  Edit  View  Search  Terminal  Help
(base) filjanssen@fil:~$ cd Downloads
(base) filjanssen@fil:~/Downloads$ sqlite3 chinook.db
SQLite version 3.30.0 2019-10-04 15:03:17
Enter ".help" for usage hints.
sqlite> select Country from customers order by Country asc;
Argentina
Australia
Austria
Belgium
Brazil
Brazil
Brazil
Brazil
Brazil
Canada
Canada
Canada
Canada
Canada
Canada
Canada
Canada
Canada
Chile
Czech Republic
Czech Republic
Denmark
Finland
France
France
France
France
France
Germany
Germany
Germany
Germany
```


DB Browser for SQLite - /home/filjanssen/Downloads/chinook.db
File Edit View Help
New Database
Open Database
Write Changes
Revert Changes
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 3
1 select Country from customers order by Country asc;
2
Country
1 Argentina
2 Australia
3 Austria
4 Belgium
5 Brazil
6 Brazil
7 Brazil
8 Brazil
59 rows returned in 1ms from: select Country from customers order by Country asc;

C. Python SQLite Database Connection sample run with **DISCUSSIONS**. Download the chinook sample database from the website mentioned in the PreLab section. Using the chinook database, run the SQLite SELECT commands in the Linux terminal and on the DB browser. Do the same using the source codes and SQL files of Cassel (Blackboard Course Materials). See below screengrabs.

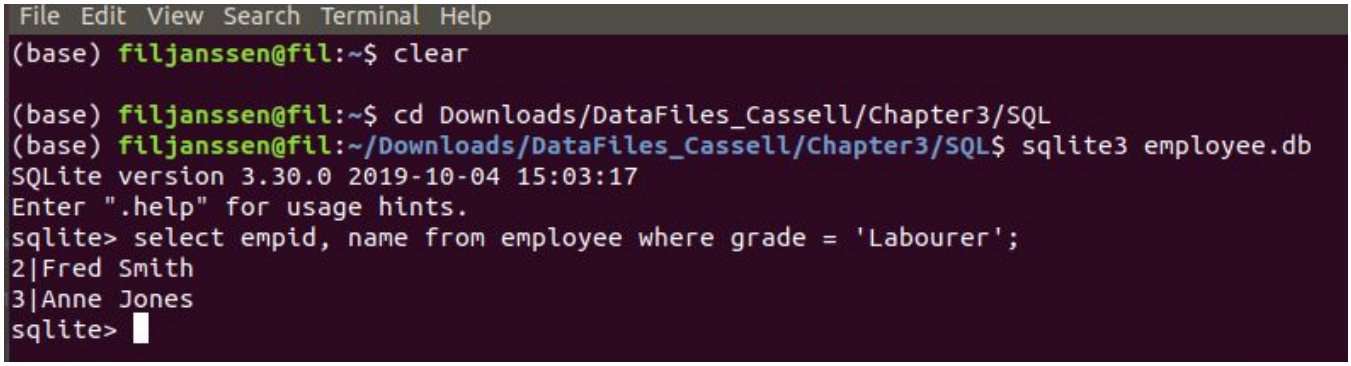


Figure 2 Opening employee.db

It just shows here how to open employee.db using terminal.

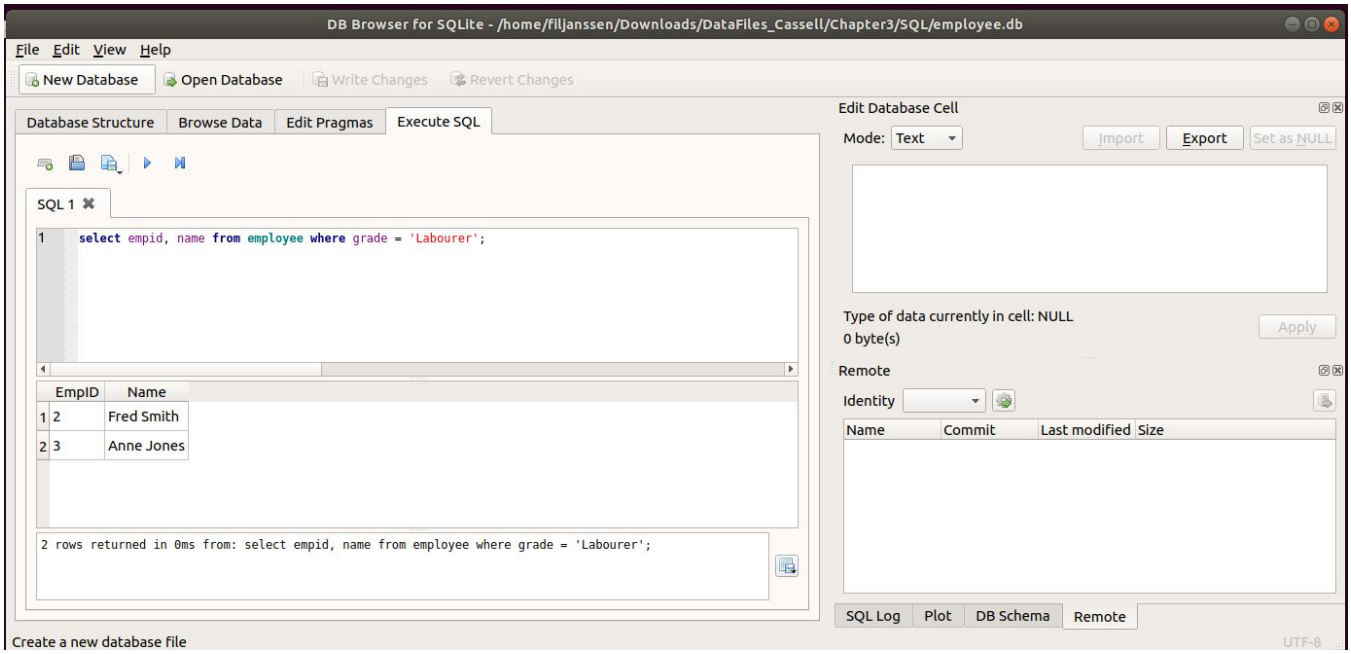
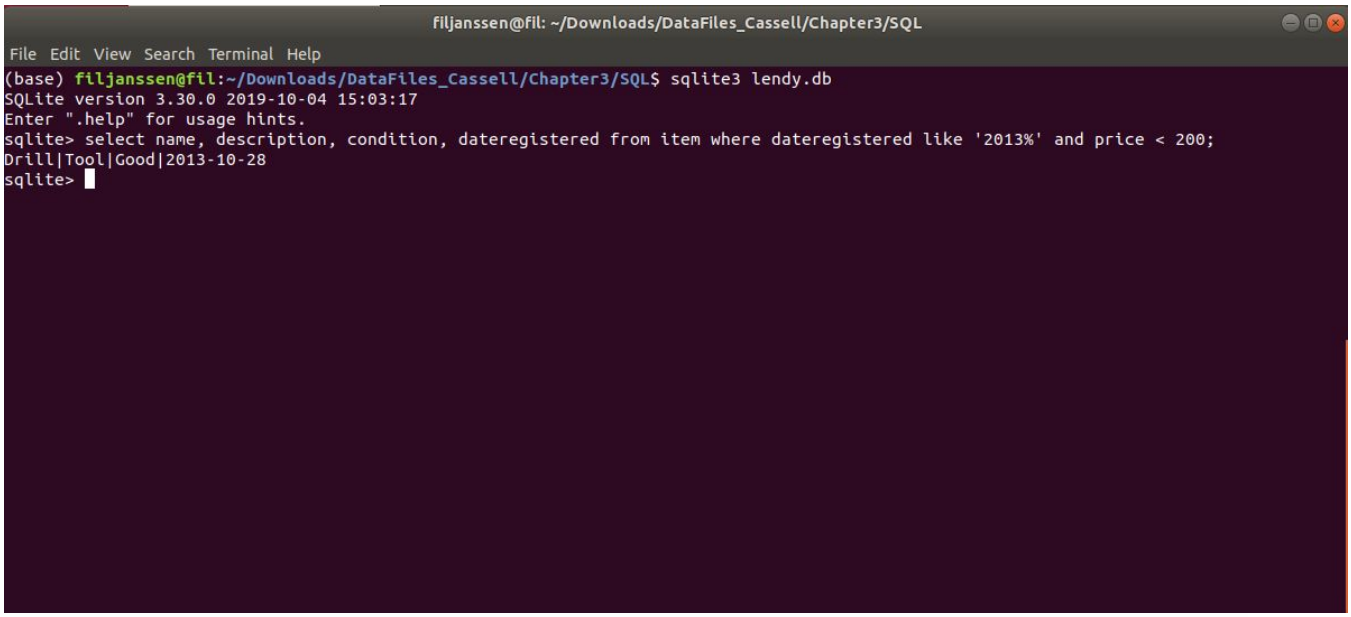
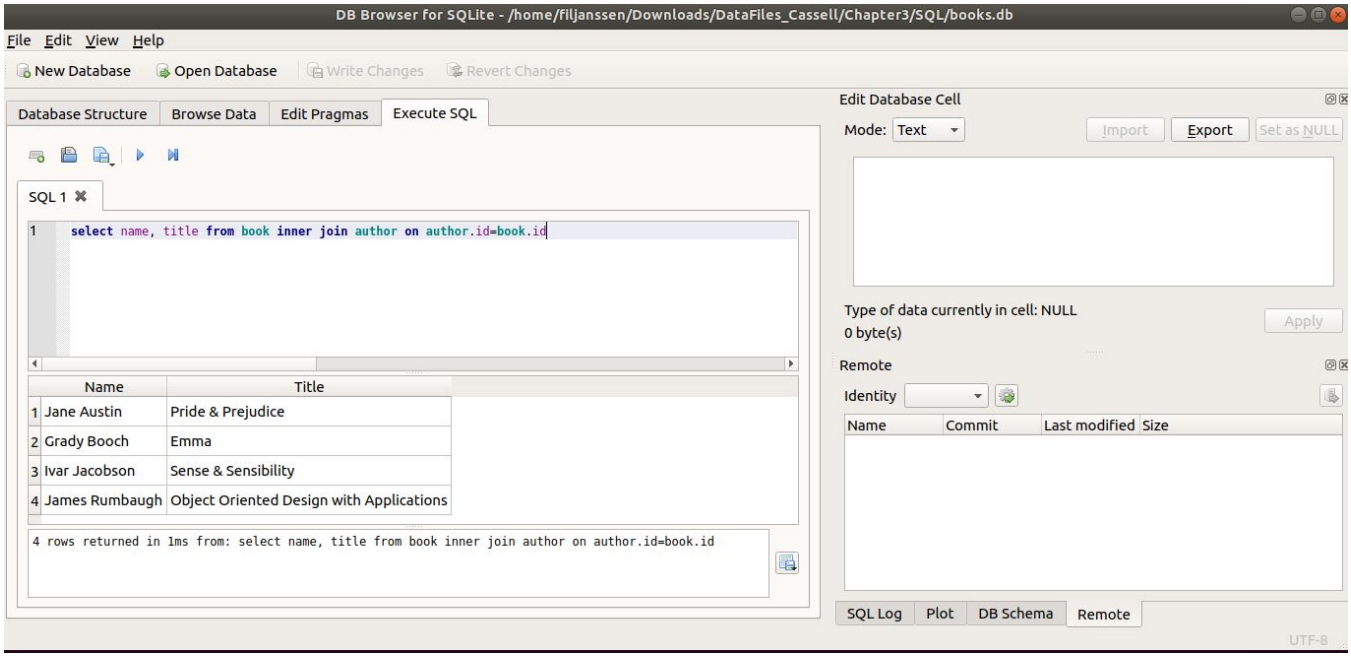
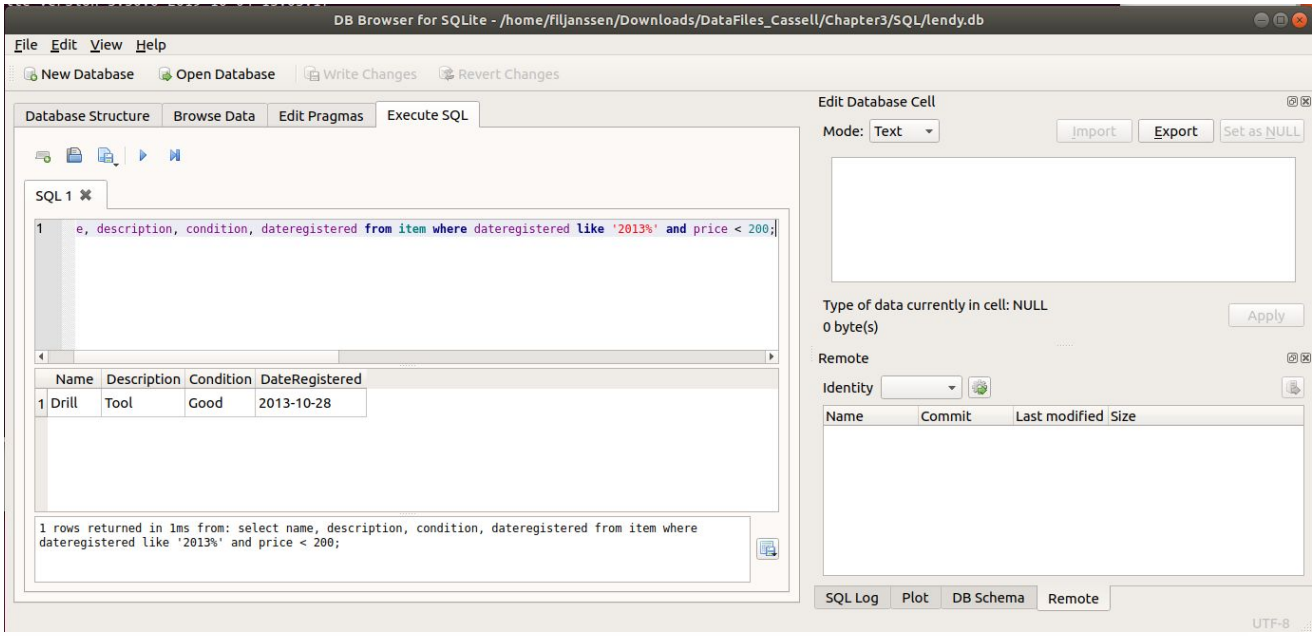


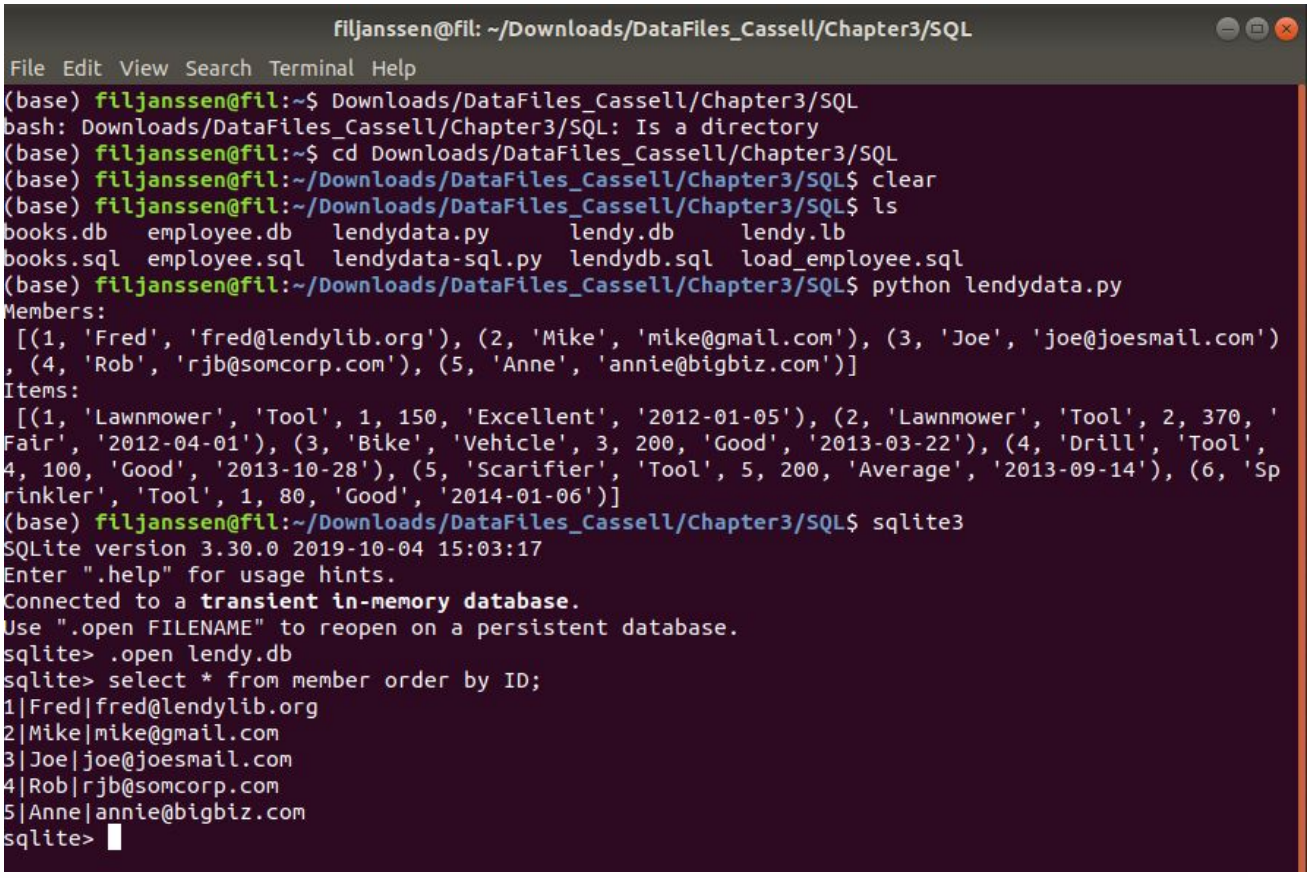
Figure 2.1 Opening the same command in DB Browser

It just shows here that DB Browser and using Linux terminal would come up with the same output.






DB Browser for SQLite - /home/filjanssen/Downloads/DataFiles_Cassell/Chapter3/SQL/lendy.db
File Edit View Help
New Database
Open Database
Write Changes
Revert Changes
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
1 e, description, condition, dateregistered from item where dateregistered like '2013%' and price < 200;
Name Description Condition DateRegistered
1 Drill Tool Good 2013-10-28
1 rows returned in 1ms from: select name, description, condition, dateregistered from item where dateregistered like '2013%' and price < 200;
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name Commit Last modified Size
SQL Log
Plot
DB Schema
Remote
UTF-8


filjanssen@fil: ~/Downloads/DataFiles_Cassell/Chapter3/SQL
File Edit View Search Terminal Help
(base) filjanssen@fil:~$ Downloads/DataFiles_Cassell/Chapter3/SQL
bash: Downloads/DataFiles_Cassell/Chapter3/SQL: Is a directory
(base) filjanssen@fil:~$ cd Downloads/DataFiles_Cassell/Chapter3/SQL
(base) filjanssen@fil:~/Downloads/DataFiles_Cassell/Chapter3/SQL$ clear
(base) filjanssen@fil:~/Downloads/DataFiles_Cassell/Chapter3/SQL$ ls
books.db employee.db lendydata.py lendy.db lendy.lb
books.sql employee.sql lendydata-sql.py lendydb.sql load_employee.sql
(base) filjanssen@fil:~/Downloads/DataFiles_Cassell/Chapter3/SQL$ python lendydata.py
Members:
[(1, 'Fred', 'fred@lendylib.org'), (2, 'Mike', 'mike@gmail.com'), (3, 'Joe', 'joe@joesmail.com'), (4, 'Rob', 'rjb@somcorp.com'), (5, 'Anne', 'annie@bigbiz.com')]
Items:
[(1, 'Lawnmower', 'Tool', 1, 150, 'Excellent', '2012-01-05'), (2, 'Lawnmower', 'Tool', 2, 370, 'Fair', '2012-04-01'), (3, 'Bike', 'Vehicle', 3, 200, 'Good', '2013-03-22'), (4, 'Drill', 'Tool', 4, 100, 'Good', '2013-10-28'), (5, 'Scarifier', 'Tool', 5, 200, 'Average', '2013-09-14'), (6, 'Sprinkler', 'Tool', 1, 80, 'Good', '2014-01-06')]
(base) filjanssen@fil:~/Downloads/DataFiles_Cassell/Chapter3/SQL$ sqlite3
SQLite version 3.30.0 2019-10-04 15:03:17
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open lendy.db
sqlite> select * from member order by ID;
1|Fred|fred@lendylib.org
2|Mike|mike@gmail.com
3|Joe|joe@joesmail.com
4|Rob|rjb@somcorp.com
5|Anne|annie@bigbiz.com
sqlite>

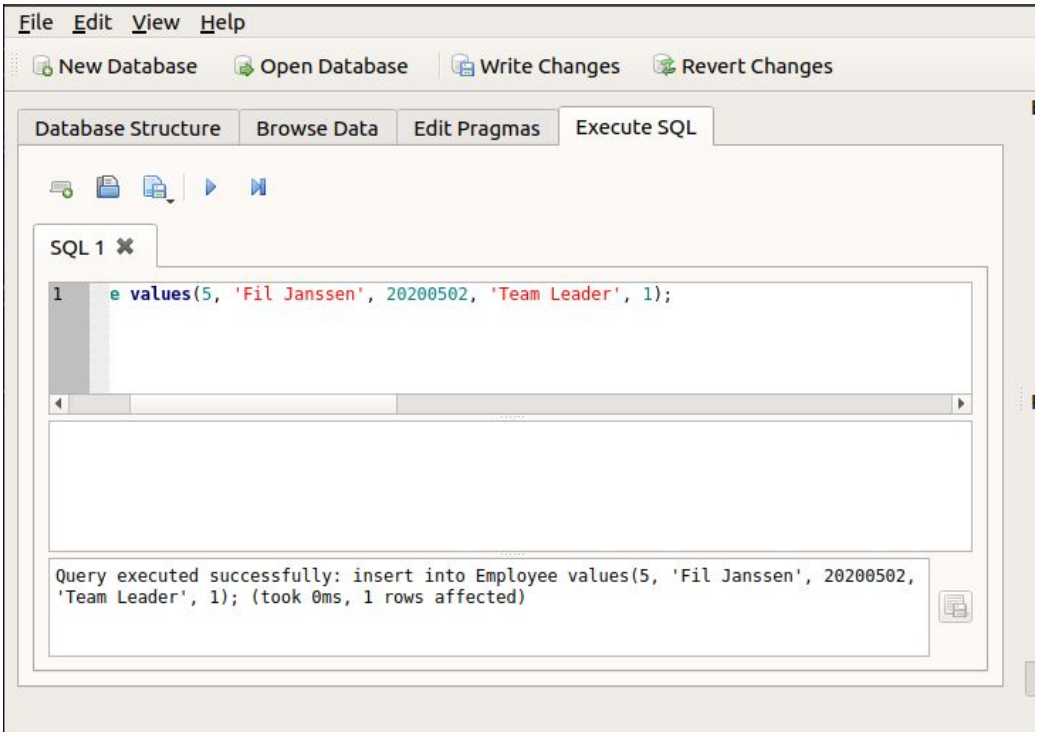


Figure 3 Adding preferred information using the given database

It just shows here that we can input some information in the database using the Execute SQL tab.

EmpID	Name	HireDate	Grade	ManagerID
Filter	Filter	Filter	Filter	Filter
1	John Brown	20030623	Foreman	*NULL*
2	Fred Smith	20040302	Labourer	1
3	Anne Jones	19991125	Labourer	1
5	Fil Janssen	20200502	Team Leader	1

Figure 3.1 Table wherein it shows the information that was input

It shows there what we did earlier.

PostLab

A. Machine 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.

   a) For each participant, list his or her number, last name, first name, address, city, state, postal code, telephone number, and date of birth.

      - For developing a report which presents the details of participants, create a participants table. The DDBL of Participants table is as Follow:
        - ➢ Participants (ParticipantNum, LastName, FirstName, Address, City, State, PostalCode, PhoneNum, DateOfBirth)
        - ➢ AK PhoneNum
        - ➢ Sk LastName
        - ➢ FK None
        - ➢ AK, SK and FK stand for Alternate Key, Secondary Key and Foreign Key respectively. The underlined attribute ParticipantNum represents the primary key.

   b) For each adventure class, list the class number, class description, maximum number of people in the class, and class fee

- For developing a report which presents the details of class, create a Class table. The DBDL of Class table is as follow:
    - ➢ Class(ClassNum, ClassDescription, MaxNumOfPersons, ClassFee)
    - ➢ AK None
    - ➢ Sk ClassDescription
    - ➢ Fk None
    - ➢ AK, SK, and FK stand for Alternate Key, Secondary Key, and Foreign Key respectively. The underlined attribute ClassNum represents the primary key.

c) 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.

- For developing a report which presents the details of class and participants both, create a ClassParticipant table. The DDBL of ClassParticipant table is as Follow:
    - ➢ ClassParticipant (ClassNum, ParticipantNum,Date) AK (ClassNum, ParticipantNum): composite key
    - ➢ Sk Date
    - ➢ FK ClassNum -> Class, ParticipantNum -> Participants
    - ➢ The above ClassParticipant connects the two tables, Class and Participants andActualParticipant field in the ClassParticipant table to add details about the actual participants.

d) 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.

- The new DBDL of ClassParticipant table is as Follow:
    - ➢ ClassParticipant (ClassName, ParticipantNum, Date, ActualParticipant)

- ➢ AK (ClassNum, ParticipantNum): composite key
- ➢ SK Date
- ➢ FK ClassNum -> Class, ParticipantNum -> Participants
- ➢ AK, SK and FK stands for Alternate Key, Secondary Key and Foreign Key respectively. The data about class and participants are fetched with the help of this table.

- The E-R diagram of the above tables is as follow:

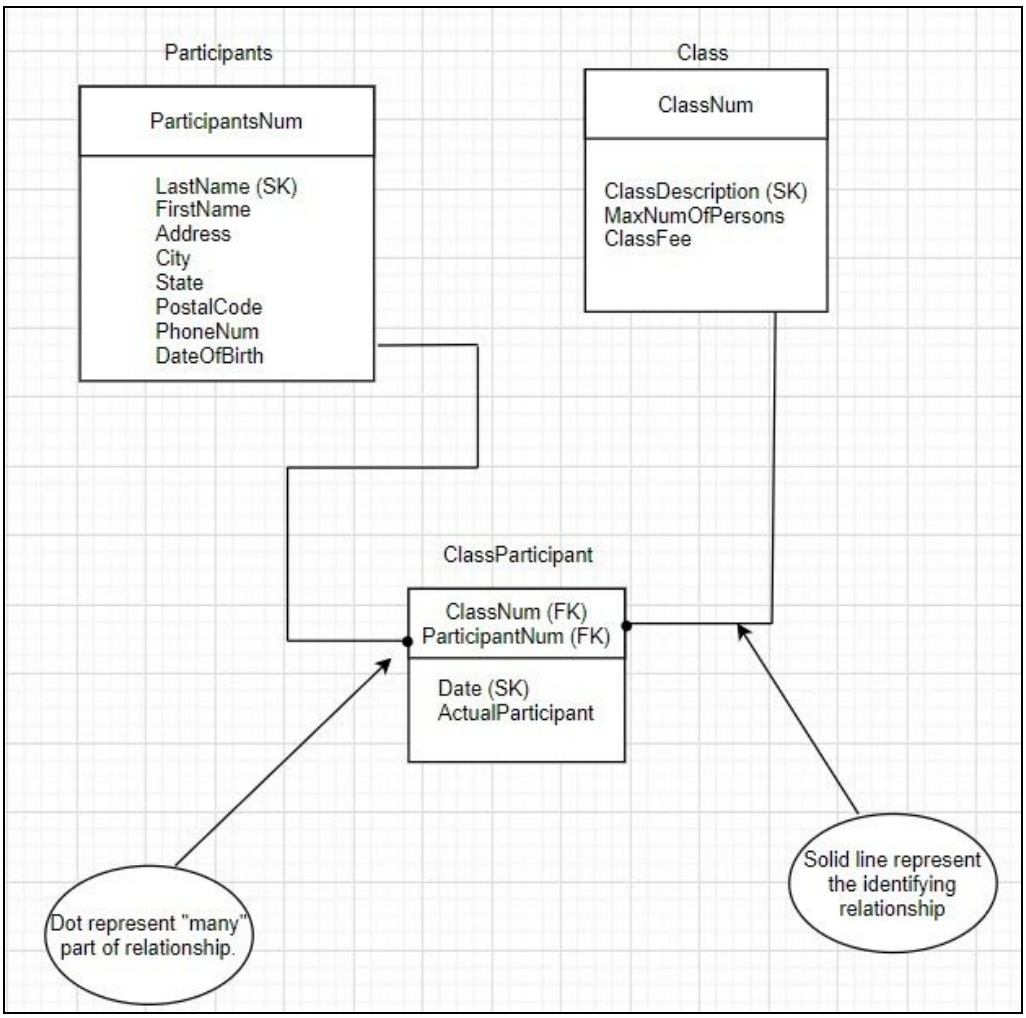


Figure 4E-Diagram for Problem 1

2. Solmaris Condominium Group has many condos that are available as weekly vacation rentals. Design a database to meet the following requirements:
    - This database is managing 2 condominium complexes, Solmaris Ocena and Solmaris Bayside, that both are located in Florida. The database has stored the data that are relevant to the managed operation and area of the condominium. The data consists of the following:
        - ➢ OWNER
        - ➢ LOCATION
        - ➢ SERVICE_REQUEST
        - ➢ CONDO_UNIT
    - We will determine the functional dependencies for the given table:
        - ➢ We assume that the weekly rate is always the same then the rate would be stored only in the CONDO_UNIT table. The design also assumes that both the columns LOCATION_NUM and the CONDO_UNIT_NUM uniquely identified a given condo number.
        - ➢ Functional Dependency:

          RENTER_NUM – FIRST_NAME, MID_INITIAL, LAST_NAME, ADDRESS, CITY, STATE, ZIP_CODE, PHONE_NUM, EMAIL
          LOCATION_NUM – LOCATION_NAME, ADDRESS, CITY, STATE, ZIP_CODE
          LOCATION_NUM, CONDO_UNIT_NUM – SQR_FT, BEDRMS, BATHS, MAX_PERSONS, WEEKLY_RATE
          RENTER_NUM, LOCATION_NUM, CONDO_UNIT_NUM – START_DATE, END_DATE, RENTAL_RATE

    - **Third normal form of normalization:**

a) For each renter, list his or her number, first name, middle initial, last name, address, city, state, postal code, telephone number, and email address.

- ➢ RENTER (RENTER_NUM, FIRST_NAME, MID_INITIAL, LAST_NAME, ADDRESS, CITY, STATE, ZIP_CODE, PHONE_NUM, EMAIL)

- **Location (LOCATION_NUM, LOCATION_NAME, ADDRESS, CITY, STATE, POSTAL_CODE):**

b) 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.

- ➢ CONDO_UNIT (CONDO_UNIT_NUM, LOCATION_NUM, SQR_FT, BEDRMS, BATHS, MAS_PERSONS, WEEKLY_RATE).

c) 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.

- ➢ RENTAL_AGREEMENT (RENTER_NUM, LOCATION_NUM, CONDO_UNIT_NUM, START_DATE, EDN_DATE, RENTAL_RATE).

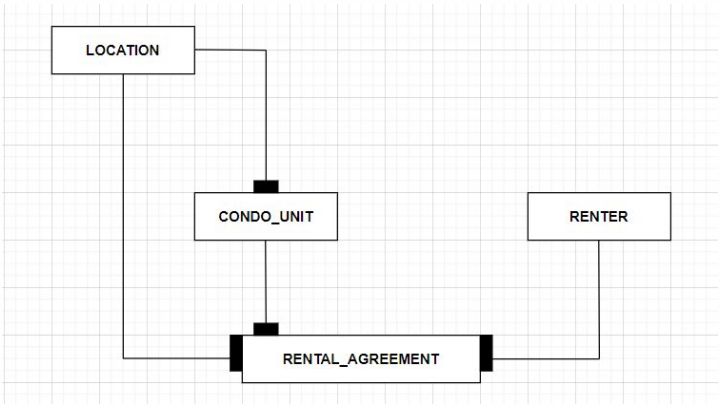


Figure 5 EDR Diagram

3. Use SQLite commands to complete the following exercises.
    a) 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.

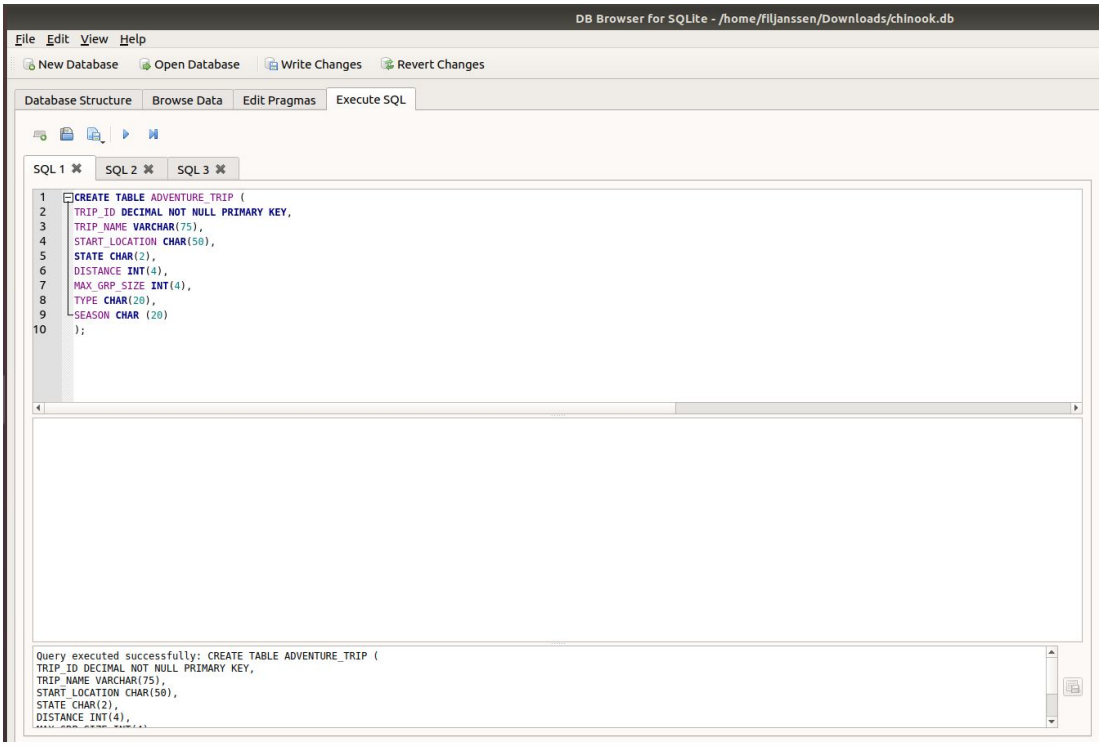


Figure 6 Creating the Table

It just shows here the syntax of creating the table in DB Browser. As you can see at the bottom, it was said to be successfully done.

b) 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.

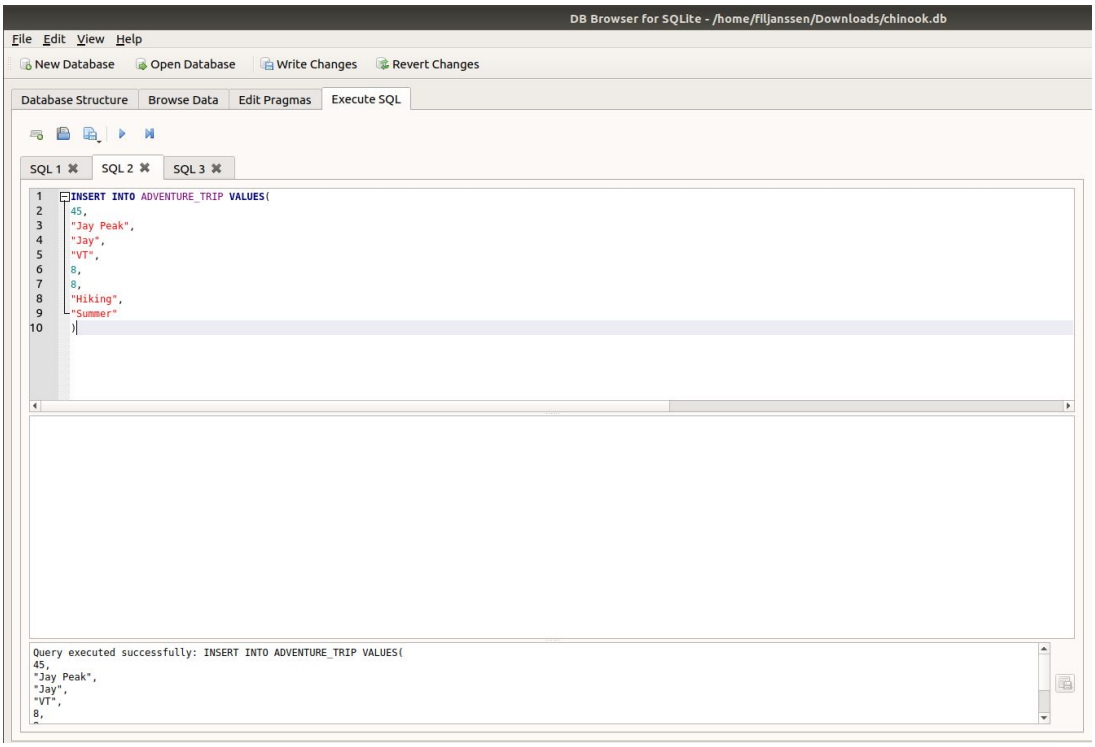


Figure 6.1 Inserting Values

It just shows here the syntax of having a value inside the table we created in figure 6,

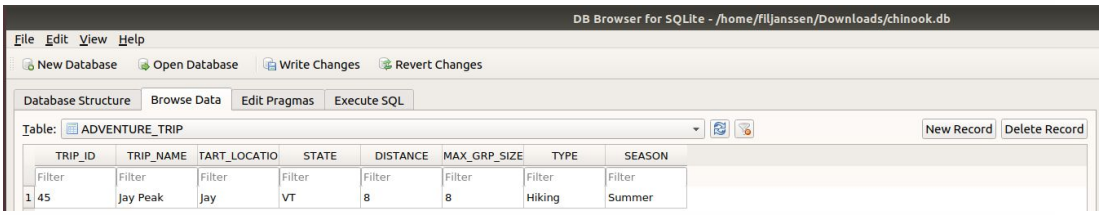


Figure 6.2 Final Table

It shows here that the information we input in figure 6.1 was added in the final table.

c) Delete the ADVENTURE_TRIP table.

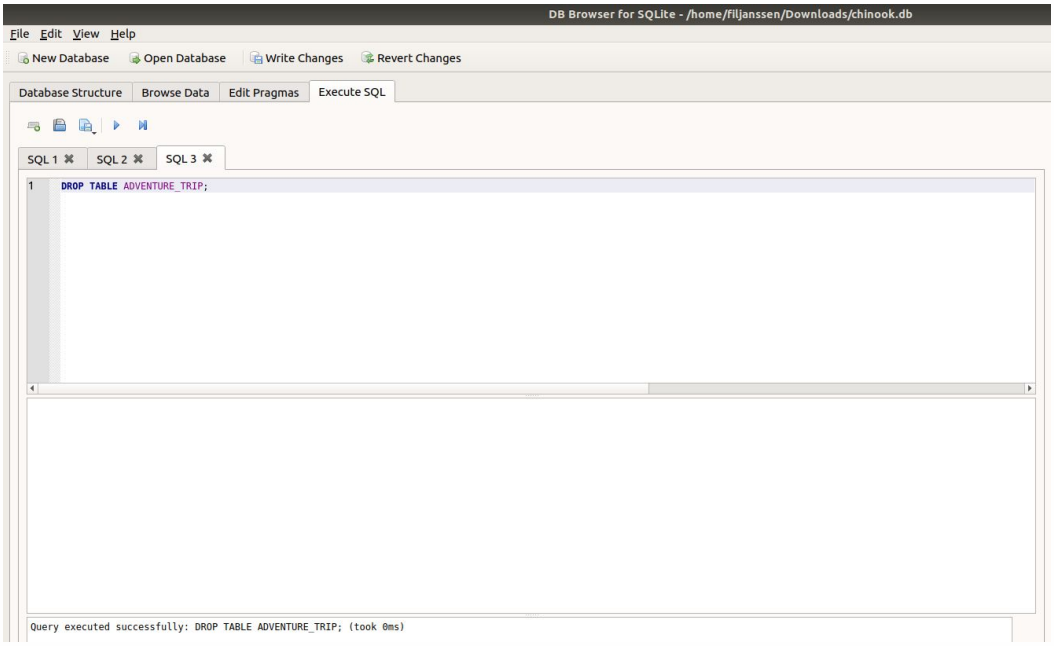


Figure 6.3 **Deleting table**

It shows here how to delete a specific table with a certain syntax.

d) 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.

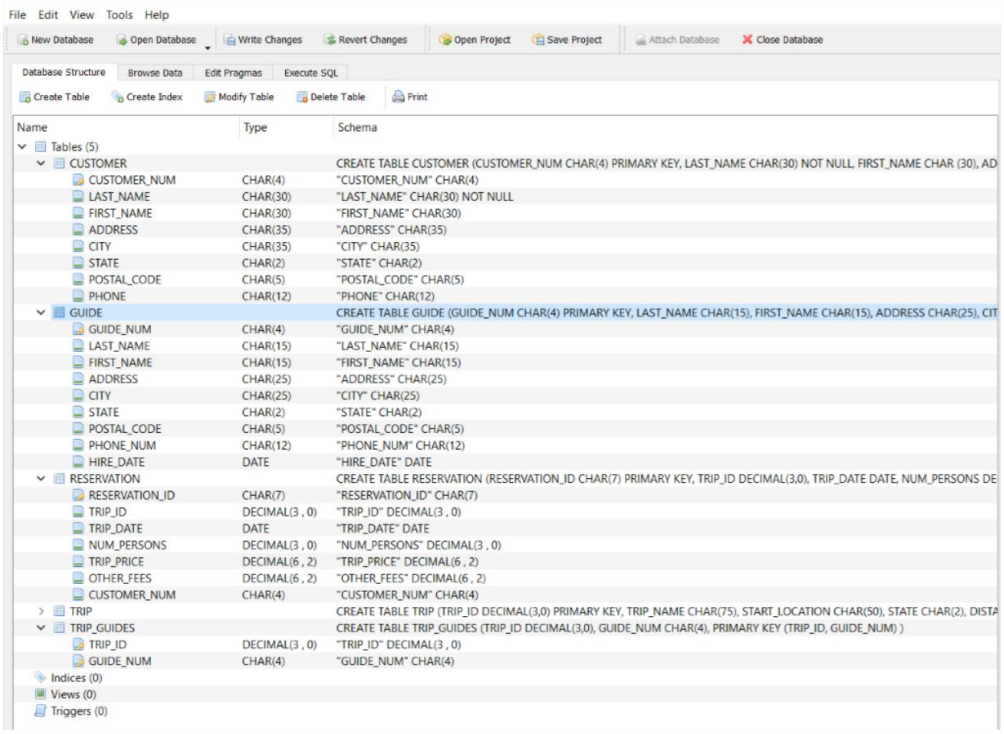
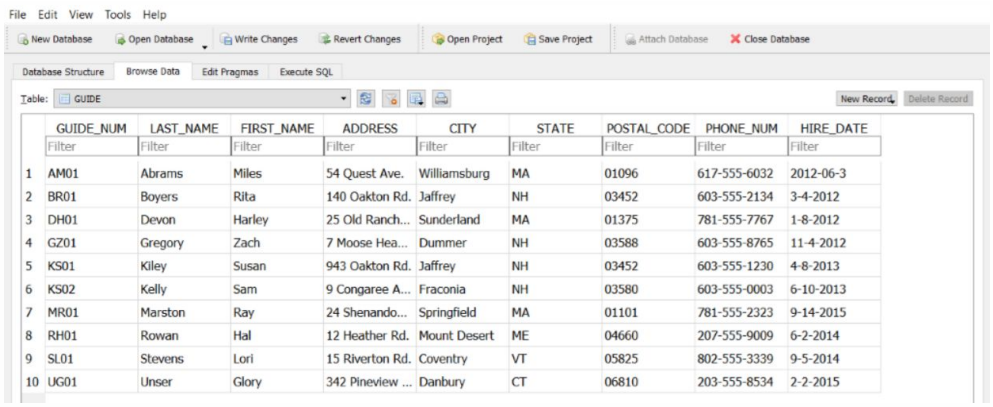


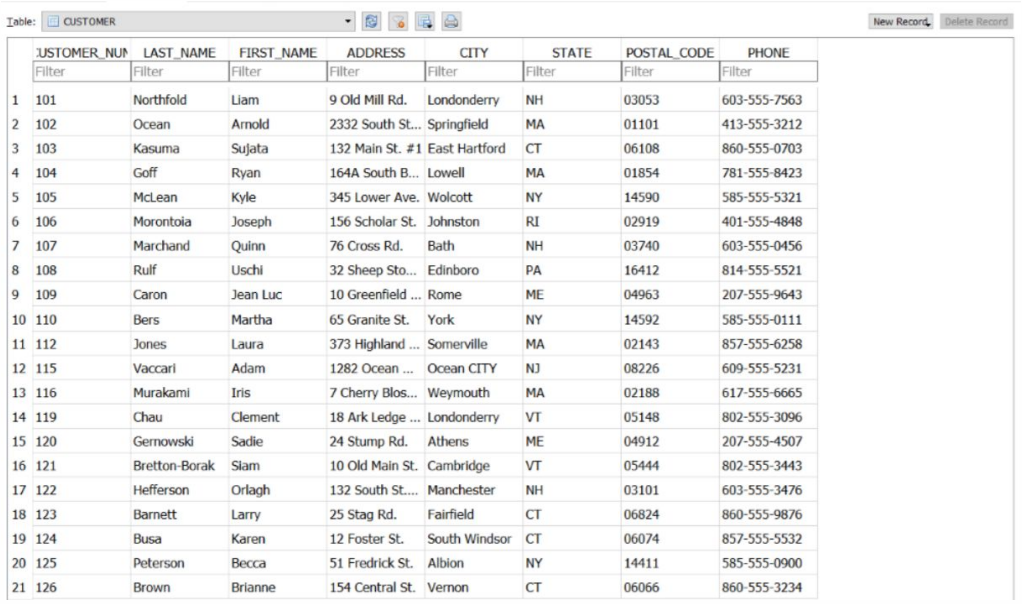
Figure 7 Executing the file in DB Browser

e) 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.



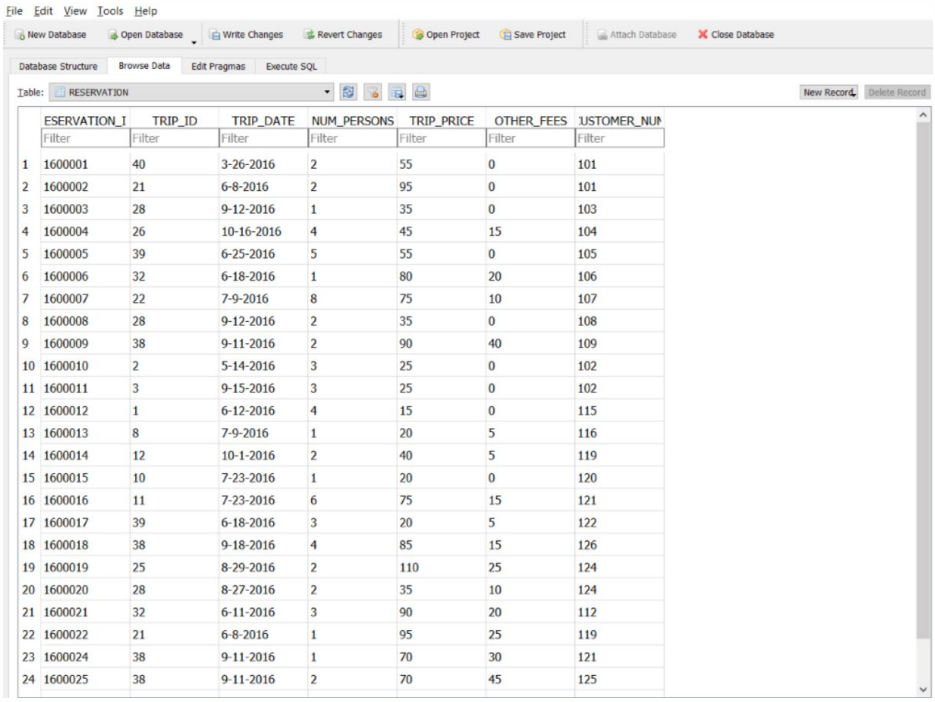
	GUIDE_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	POSTAL_CODE	PHONE_NUM	HIRE_DATE
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	AM01	Abrams	Miles	54 Quest Ave.	Williamsburg	MA	01096	617-555-6032	2012-06-3
2	BR01	Boyers	Rita	140 Oakton Rd.	Jaffrey	NH	03452	603-555-2134	3-4-2012
3	DH01	Devon	Harley	25 Old Ranch...	Sunderland	MA	01375	781-555-7767	1-8-2012
4	GZ01	Gregory	Zach	7 Moose Hea...	Dummer	NH	03588	603-555-8765	11-4-2012
5	KS01	Kiley	Susan	943 Oakton Rd.	Jaffrey	NH	03452	603-555-1230	4-8-2013
6	KS02	Kelly	Sam	9 Congaree A...	Fraconia	NH	03580	603-555-0003	6-10-2013
7	MR01	Marston	Ray	24 Shenando...	Springfield	MA	01101	781-555-2323	9-14-2015
8	RH01	Rowan	Hal	12 Heather Rd.	Mount Desert	ME	04660	207-555-9009	6-2-2014
9	SL01	Stevens	Lori	15 Riverton Rd.	Coventry	VT	05825	802-555-3339	9-5-2014
10	UG01	Unser	Glory	342 Pineview ...	Danbury	CT	06810	203-555-8534	2-2-2015

Figure 7.1 Colonial Adventure Tours Database GUIDE Table



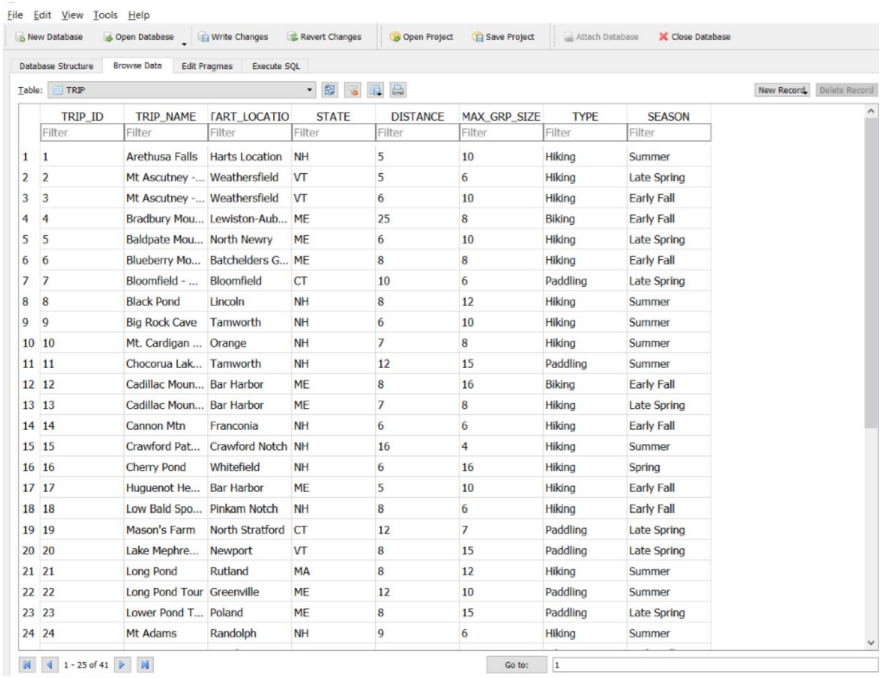
	USTOMER_NUM	LAST_NAME	FIRST_NAME	ADDRESS	CITY	STATE	POSTAL_CODE	PHONE
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	101	Northfold	Liam	9 Old Mill Rd.	Londonderry	NH	03053	603-555-7563
2	102	Ocean	Arnold	2332 South St...	Springfield	MA	01101	413-555-3212
3	103	Kasuma	Sujata	132 Main St. #1	East Hartford	CT	06108	860-555-0703
4	104	Goff	Ryan	164A South B...	Lowell	MA	01854	781-555-8423
5	105	McLean	Kyle	345 Lower Ave.	Wolcott	NY	14590	585-555-5321
6	106	Morontoia	Joseph	156 Scholar St.	Johnston	RI	02919	401-555-4848
7	107	Marchand	Quinn	76 Cross Rd.	Bath	NH	03740	603-555-0456
8	108	Rulf	Uschi	32 Sheep Sto...	Edinboro	PA	16412	814-555-5521
9	109	Caron	Jean Luc	10 Greenfield ...	Rome	ME	04963	207-555-9643
10	110	Bers	Martha	65 Granite St.	York	NY	14592	585-555-0111
11	112	Jones	Laura	373 Highland ...	Somerville	MA	02143	857-555-6258
12	115	Vaccari	Adam	1282 Ocean ...	Ocean CITY	NJ	08226	609-555-5231
13	116	Murakami	Iris	7 Cherry Blos...	Weymouth	MA	02188	617-555-6665
14	119	Chau	Clement	18 Ark Ledge ...	Londonderry	VT	05148	802-555-3096
15	120	Gernowski	Sadie	24 Stump Rd.	Athens	ME	04912	207-555-4507
16	121	Bretton-Borak	Slam	10 Old Main St.	Cambridge	VT	05444	802-555-3443
17	122	Hefferson	Orlagh	132 South St....	Manchester	NH	03101	603-555-3476
18	123	Barnett	Larry	25 Stag Rd.	Fairfield	CT	06824	860-555-9876
19	124	Busa	Karen	12 Foster St.	South Windsor	CT	06074	857-555-5532
20	125	Peterson	Becca	51 Fredrick St.	Albion	NY	14411	585-555-0900
21	126	Brown	Brianne	154 Central St.	Vernon	CT	06066	860-555-3234

Figure 7.2 Colonial Adventure Tours Database RESERVATION Table



	ESERVATION_I	TRIP_ID	TRIP_DATE	NUM_PERSONS	TRIP_PRICE	OTHER_FEES	USTOMER_NUI
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1600001	40	3-26-2016	2	55	0	101
2	1600002	21	6-8-2016	2	95	0	101
3	1600003	28	9-12-2016	1	35	0	103
4	1600004	26	10-16-2016	4	45	15	104
5	1600005	39	6-25-2016	5	55	0	105
6	1600006	32	6-18-2016	1	80	20	106
7	1600007	22	7-9-2016	8	75	10	107
8	1600008	28	9-12-2016	2	35	0	108
9	1600009	38	9-11-2016	2	90	40	109
10	1600010	2	5-14-2016	3	25	0	102
11	1600011	3	9-15-2016	3	25	0	102
12	1600012	1	6-12-2016	4	15	0	115
13	1600013	8	7-9-2016	1	20	5	116
14	1600014	12	10-1-2016	2	40	5	119
15	1600015	10	7-23-2016	1	20	0	120
16	1600016	11	7-23-2016	6	75	15	121
17	1600017	39	6-18-2016	3	20	5	122
18	1600018	38	9-18-2016	4	85	15	126
19	1600019	25	8-29-2016	2	110	25	124
20	1600020	28	8-27-2016	2	35	10	124
21	1600021	32	6-11-2016	3	90	20	112
22	1600022	21	6-8-2016	1	95	25	119
23	1600024	38	9-11-2016	1	70	30	121
24	1600025	38	9-11-2016	2	70	45	125

Figure 7.3 Colonial Adventure Tours Database Reservation Table



	TRIP_ID	TRIP_NAME	TART_LOCATIO	STATE	DISTANCE	MAX_GRP_SIZE	TYPE	SEASON
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Arethusa Falls	Harts Location	NH	5	10	Hiking	Summer
2	2	Mt Ascutney -...	Weathersfield	VT	5	6	Hiking	Late Spring
3	3	Mt Ascutney -...	Weathersfield	VT	6	10	Hiking	Early Fall
4	4	Bradbury Mou...	Lewiston-Aub...	ME	25	8	Biking	Early Fall
5	5	Baldpate Mou...	North Newry	ME	6	10	Hiking	Late Spring
6	6	Blueberry Mo...	Batchelders G...	ME	8	8	Hiking	Early Fall
7	7	Bloomfield - ...	Bloomfield	CT	10	6	Paddling	Late Spring
8	8	Black Pond	Lincoln	NH	8	12	Hiking	Summer
9	9	Big Rock Cave	Tamworth	NH	6	10	Hiking	Summer
10	10	Mt. Cardigan ...	Orange	NH	7	8	Hiking	Summer
11	11	Chocorua Lak...	Tamworth	NH	12	15	Paddling	Summer
12	12	Cadillac Moun...	Bar Harbor	ME	8	16	Biking	Early Fall
13	13	Cadillac Moun...	Bar Harbor	ME	7	8	Hiking	Late Spring
14	14	Cannon Mtn	Franconia	NH	6	6	Hiking	Early Fall
15	15	Crawford Pat...	Crawford Notch	NH	16	4	Hiking	Summer
16	16	Cherry Pond	Whitefield	NH	6	16	Hiking	Spring
17	17	Huguenot He...	Bar Harbor	ME	5	10	Hiking	Early Fall
18	18	Low Bald Spo...	Pinkam Notch	NH	8	6	Hiking	Early Fall
19	19	Mason's Farm	North Stratford	CT	12	7	Paddling	Late Spring
20	20	Lake Mephre...	Newport	VT	8	15	Paddling	Late Spring
21	21	Long Pond	Rutland	MA	8	12	Hiking	Summer
22	22	Long Pond Tour	Greenville	ME	12	10	Paddling	Summer
23	23	Lower Pond T...	Poland	ME	8	15	Paddling	Late Spring
24	24	Mt Adams	Randolph	NH	9	6	Hiking	Summer

Figure 7.4 Colonial Adventure Tours Database TRIP Table

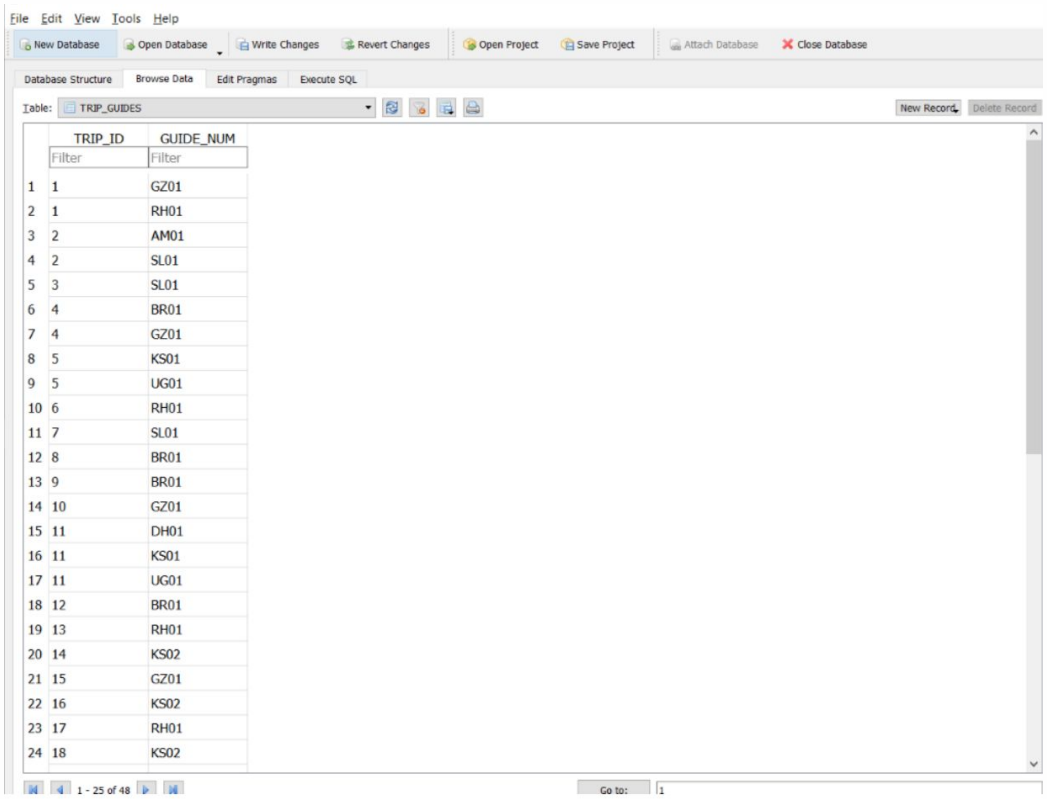


Figure 7.5 Colonial Adventure Tours Database TRIP_GUIDES Table

B. Debugging and Sample Run of Python program connection to your created SQLite database (with edited screengrabs and discussion)

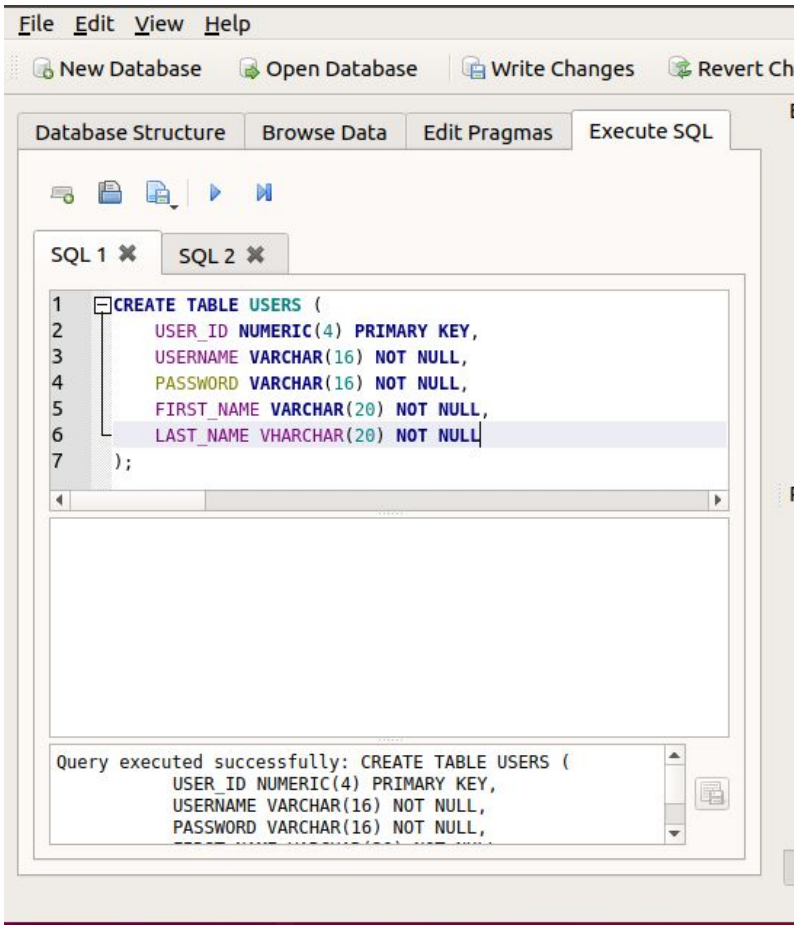


Figure 8 Creating Table

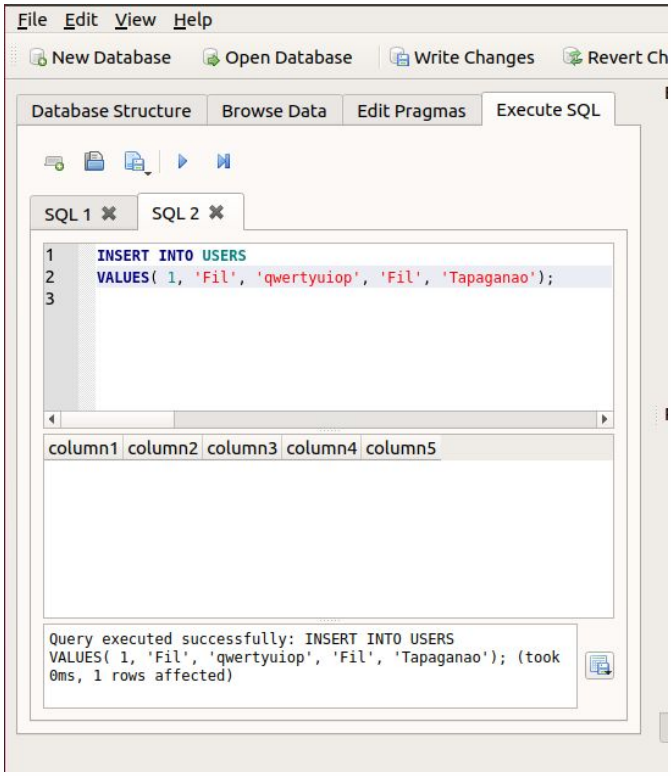


Figure 8.1 Inserting Values

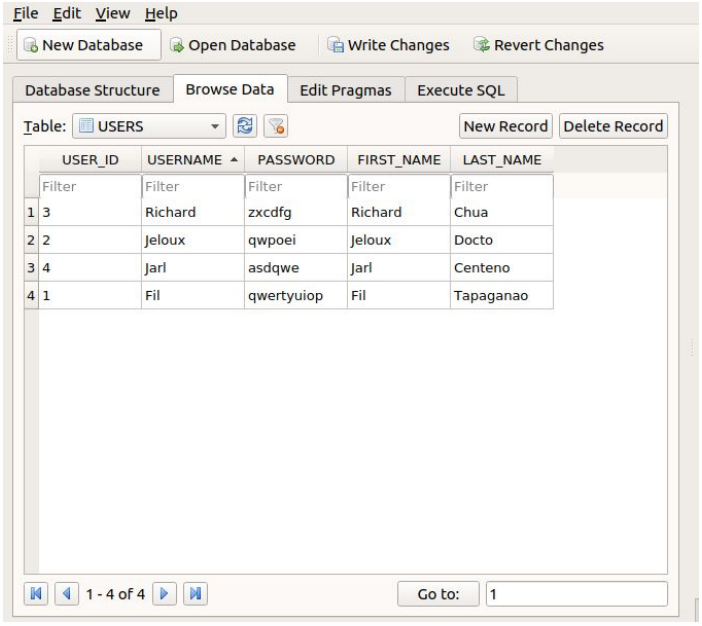


Figure 8.2 Table Created

B. Note:

- **Github:** https://bit.ly/2z2SVdn
- **OneDrive:** https://bit.ly/3d0HpOi
- **OneNote:** https://bit.ly/2zUkFkV