

ISYS2095 Database Concepts 2020

Relational Database Model

The objectives of this activity are:

- Learn about the relational database model;
- Learn how to create tables in SQLite Studio
- Learn how to do simple CRUD (create, read, update, and delete) data in tables;
- Learn about integrity constraints;
- Learn how to apply integrity constraints in SQLite Studio to make tables to act like relations;

Part 1

STEP 1:

By now, you should have SQLite installed on your laptop.

If not, there are instructional documents for downloading and using SQLite Studio on the course Canvas module Sample Databases and Tools.

If you continue to have trouble, please contact your tutor for assistance.

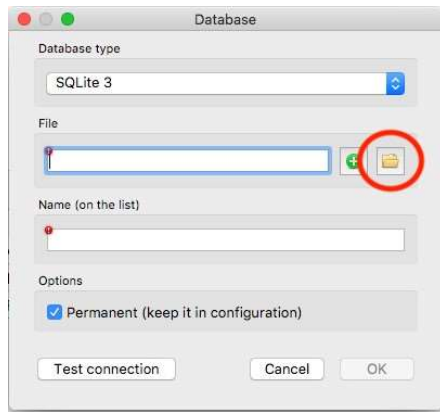
STEP 2:

Test your database installation with a sample database. Download the Movies.db file from the Sample Databases and Tools module.

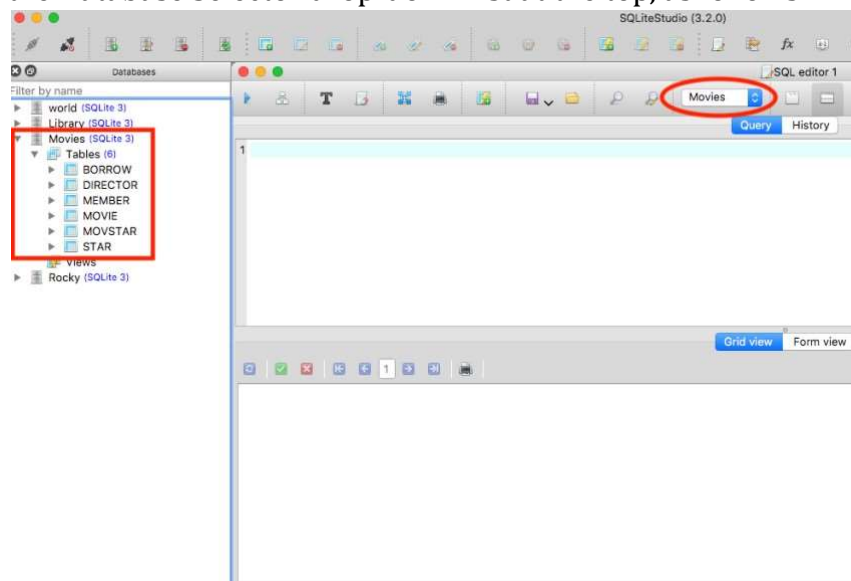
Step 3:

Connect.

Click on Database ➤ Add Database and on the dialog box, click “Browse for Existing Database Files” icon and “OK”.



This will build a new database and it should appear on the left-hand pane AND on the Database Selector drop-down list at the top, as follows:



(If it appears on left-hand pane, but not on the Database Selector list, that means, the database is available, but is not connected. It can be connected by double clicking database name on the left-hand pane.)

If you are successful, you should have presented a query editor where you can enter and run SQL commands and a Grid View underneath it to view query results.

PART 2:

Create a directory (say, Week-2-Lab) and download following three files from the module Sample Databases and Tools:

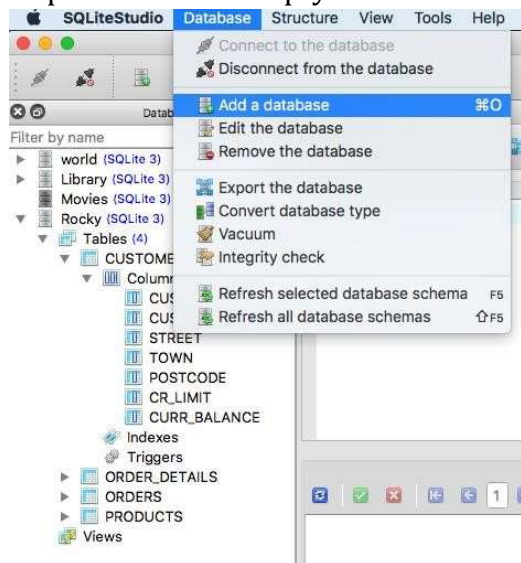
- Rocky-basics-No-Constraints.sql
- Rocky-basics-With-Constraints.sql
- Rocky-concrete-data.sql


Activity 1: Creating Database Tables in SQLite Studio

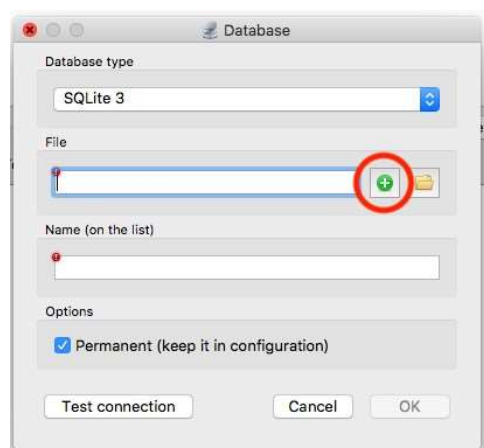
In this activity, you familiarize yourself with running a SQL DDL script in SQLite Studio to create following four tables in a database for a fictitious Rocky Concrete Store.

- Customers
- Products
- Orders
- Order_Details

Step 1: Create an empty database on SQLite Studio.



Click on Database ⑦ Add Database and on the dialog box, click “Create a new Database File” icon  and “OK”.

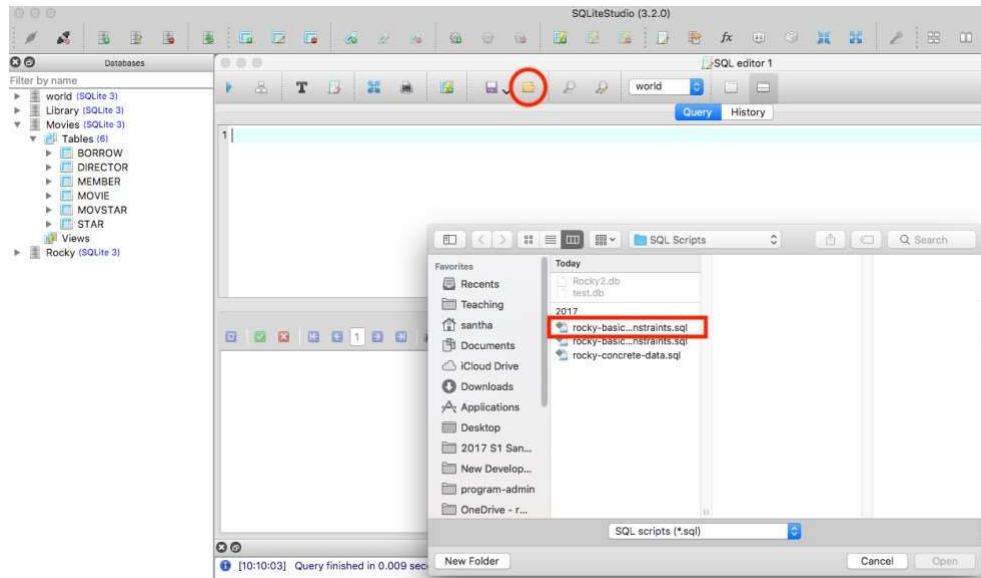


Call it “Rocky” and save it on Week-2-lab directory. At this point, you have a database, however, it is empty (no tables/ relations).

Step 2: Create tables.

You have been provided a SQL script to create four tables. (It first checks if these tables exist and if so, it deletes old copies and then create new versions).

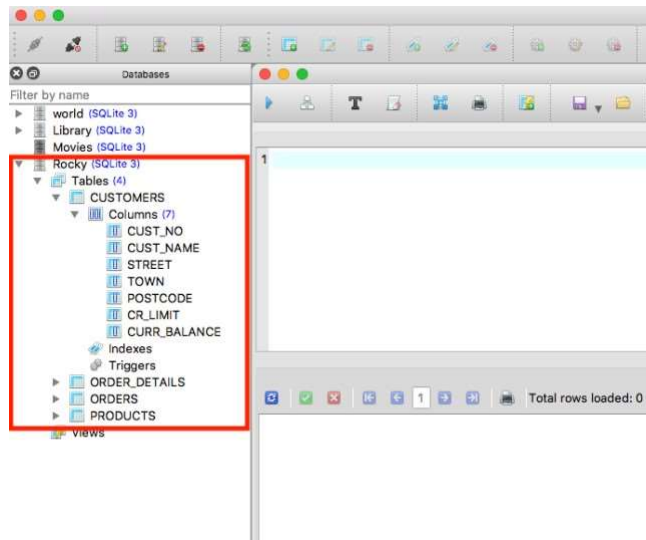
Click on “Load SQL from file” button as shown on the following screenshot, and choose the “rocky-concrete-No-Constraints.sql” file.



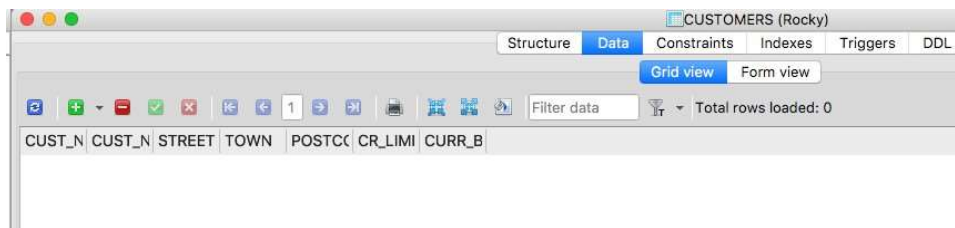
Once the script appears on your SQL Editor, highlight the whole script and hit the “Execute Query” button at the top. It is important to highlight the whole script, otherwise it will only run the topmost SQL Query.

Step 3: Get familiar with the database

1. To list all the tables in the database, double click on the database name (Rocky) on the left-hand pane.

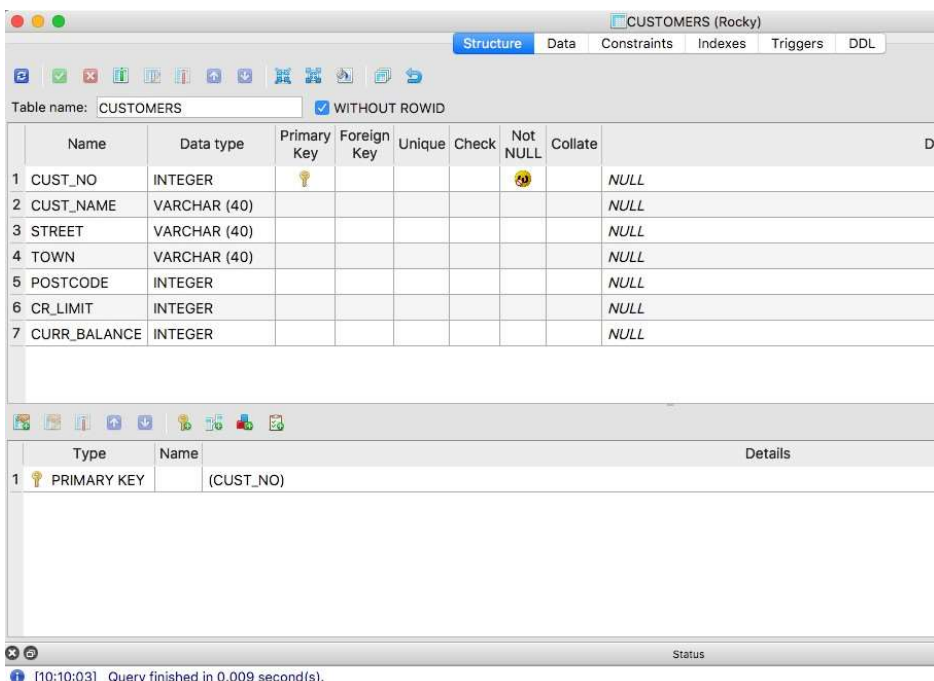


2. To view full listing of Customers table in, double click on the table name and click on “Data” tab.



Since you haven’t populated the tables yet, you won’t see any rows.

3. To view Customers table schema, double click on the table name and click on the “Structure” tab.



This will show the list of attributes in the table with data types and (if any) constraints.

Step 4: Populate the tables.

Open a SQL Editor window (if you haven't got one already opened).

Tools  Open SQL Editor

1. Start to populate the Customer Table using the following SQL statement. (If you wish to copy-and-paste the command, do not use the following (as it was generated using Word. Use the SQL statements in the "rocky-concretedata.sql" you downloaded from the Blackboard.)

```
INSERT into Customers  
VALUES (2001, 'Glads Gladdies', 'Childs Road', 'Mill Park', 3082, 500, 0)
```

Try Step 2 above (in step 3) again, this time using a SQL command.

```
SELECT * FROM customers
```

5. Let's try another row. This time, customer's name is "Nev's Nursery". It contains an apostrophe (single quotation mark). A single quotation mark (') within the literal must be preceded by an escape character. To represent one single quotation mark within a literal, enter two single quotation marks.

```
INSERT into Customers  
VALUES (1066, 'Nev's Nursery', 'White Hart Lane', 'Bundoora', 3083, 500, 450);
```

Note that some text editors change two adjacent single quotation marks into one double quotation mark. If you do a copy-and-paste from a text editor, make sure that they stay as two single quotation marks.

7. Let's try another row. This time try re-inserting the first row again.

```
INSERT into Customers  
VALUES (2001, 'Glads Gladdies', 'Childs Road', 'Mill Park', 3082, 500, 0)
```

Were you able to enter this new row?

What constraint have you violated? Discuss.

Now, try to insert a new customer, who hasn't got a customer number. When you enter only a few values of a row, you must specify which attribute values you are going to enter, as follows:

```
INSERT into Customers (cust_name, street, town, postcode)
VALUES ('Di Hunter', 'Thornton Farm', 'Whittlesea', 3757);
```

Were you able to enter this new row?

What constraint have you violated? Discuss.

7. Continue populating other tables with the following data.

```
INSERT INTO Products VALUES ('MOO', 'Medium Cattle Trough', 'A', 150, 6, 3, 5);
INSERT INTO Orders VALUES (1, '01-07-1993', 13144);
INSERT INTO Order_details VALUES(1, 'STAND', 10, 45);
```

Were you able to enter these new rows?

What constraint have you violated? Discuss.

You have noticed that the tables you created and populated in the Rocky Concrete database do not comply with additional requirements to be treated as **relations**. They are just plain tables.

In the next activity, we rebuild them with required additional constraints to make them into proper **relations**.

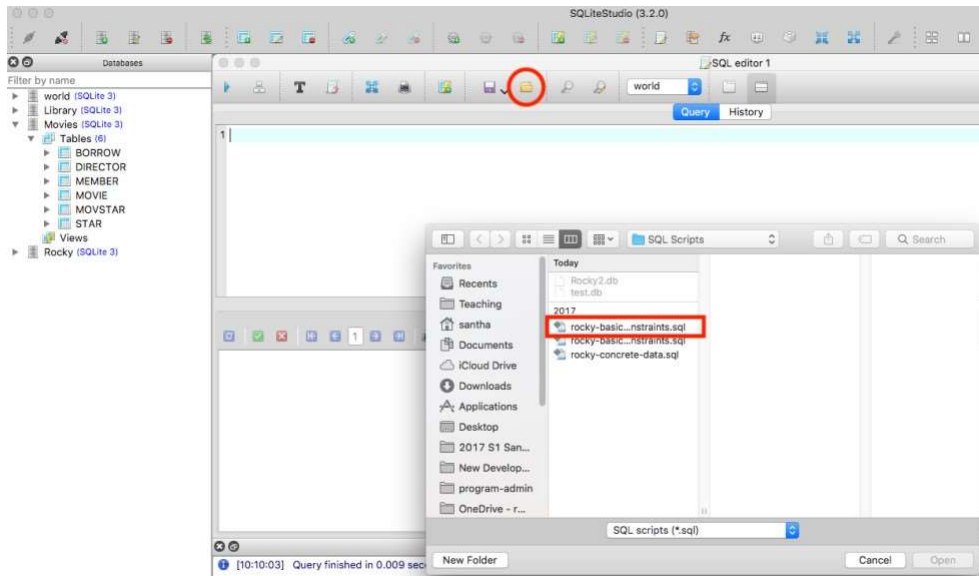
PART 2

Creating proper “Relations” in SQLite Studio

The file you downloaded at the beginning – “rocky-concrete-With-Constraints.sql” – has key, entity integrity and referential integrity constraints embedded in the corresponding CREATE TABLE SQL commands.

Compare these commands with the ones appear on “rocky-concrete-No-Constraints.sql”. Identify the new clauses included in this file, which were absent in the first one.

Click on “Load SQL from file” button as shown on the following screenshot and choose “rocky-concrete-With-Constraints.sql” file.



Once the script appears on your SQL Editor, highlight the whole script and hit the “Execute Query” button at the top. It is important to highlight the whole script, otherwise it will only run the topmost SQL Query.

1. Start to populate the Customer Table using the following SQL statement. If you wish to copy-and-paste the command, **do not** use the following (as it was generated using Word and copy-pasting the rich text will likely introduce syntax errors). Instead, use the SQL statements in the “rockyconcrete-data.sql” you downloaded prior.

INSERT into Customers
VALUES (2001, 'Glads Gladdies', 'Childs Road', 'Mill Park', 3082, 500, 0)

2. Let’s try another row. This time try re-inserting the first row again.

INSERT into Customers
VALUES (2001, 'Glads Gladdies', 'Childs Road', 'Mill Park', 3082, 500, 0)

Were you able to enter this new row? Discuss.

3. Now, try to insert a new customer, who hasn’t got a customer number. When you enter only a few values of a row, you must specify which attribute values you are going to enter, as follows:


```
INSERT into Customers (cust_name, street, town, postcode)
VALUES ('Di Hunter', 'Thornton Farm', 'Whittlesea', 3757);
```

Were you able to enter this new row? Discuss.

7. Continue populating other tables with following data.

```
INSERT INTO Products VALUES ('MOO', 'Medium Cattle Trough', 'A', 150, 6, 3, 5);
INSERT INTO Orders VALUES (1, '01-07-1993', 13144);
INSERT INTO Order_details VALUES (1, 'STAND', 10, 45);
```

Were you able to enter these new rows? Discuss.

PART 3

REVIEW OF INTEGRITY CONSTRAINTS

This question has been adopted from Fundamentals of Database Systems, Elmasri and Navathe. (Question 5.11).

A relational database schema and an instance of this schema are given below. Each of the operations given in questions 1 to 8 are applied on the original database instance, disregarding the effects of the previous operations. Suppose that each of the update operations given below is applied directly to the database instance shown in the diagram. (That is, assume we rollback the database instance to its original state after each operation.)

Most of the attribute names are self-explanatory. Super_SSN refers to the corresponding employee's supervisor's SSN (Social Security Number). This example is based on US system, assume it is similar to the Australian Tax File Number.

Arrows indicate foreign keys and the corresponding attributes in the parent relation. In the case of Super_SSN, the parent relation is the Employee relation itself (self-referencing).

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

© 2007 Ramez Elmasri and Shamkant B. Navathe

List **all** integrity constraints violated by each operation. If the operation does not violate any constraints, indicate this as 'no violations'.

For each and every violation, explain why such violation occurred and actions you can take to make sure that the operation does not violate any integrity constraints.

1. Insert <'Robert', 'F', 'Scott', '987654321', '21-JUN-42', '2365 Newcastle Rd, Bellaire, TX', M, 58000, '888665555', 1 > into EMPLOYEE.
2. Insert <'ProductA', 3, 'Bellaire', 2> into PROJECT.
3. Insert <'Production', 4, '943775543', '01-OCT-88' > into DEPARTMENT.
4. Insert <'123454321', null, '40.0'> into WORKS_ON.
5. Insert <'453345453', 'John', M, '12-DEC-60', 'SPOUSE'> into DEPENDENT.
6. Delete the DEPENDENT tuples with dependent_name= 'Joy'.
7. Delete the EMPLOYEE tuple with SSN= '888665555'.
8. Update the SUPER_SSN of the EMPLOYEE tuple with SSN='123456789' to '666884444'.

Have your answers prepared and ready to discuss in the next webinar session.