

Relational Databases

Charles Severance



open.michigan

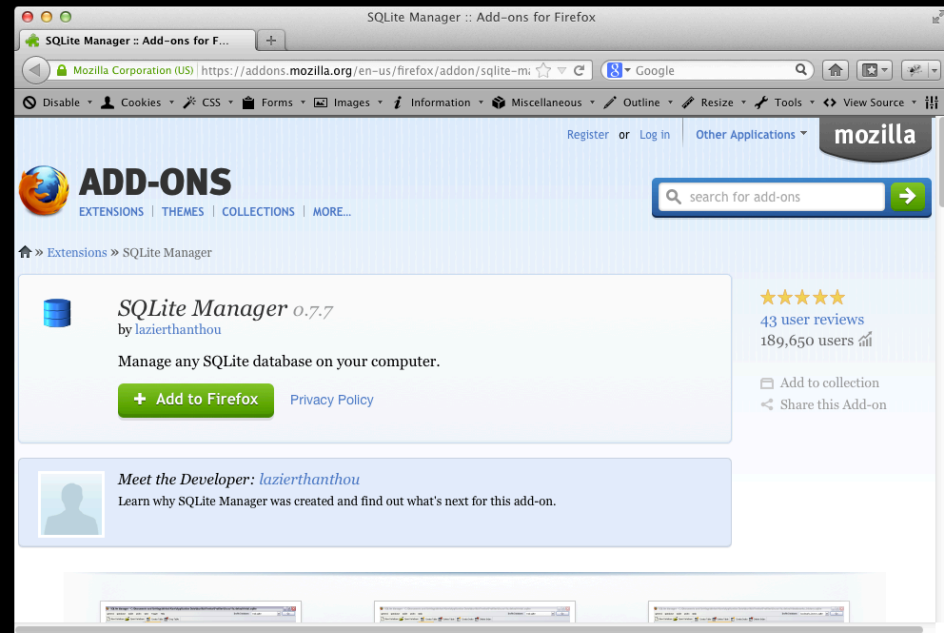
Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution 3.0 License.

<http://creativecommons.org/licenses/by/3.0/>.

Copyright 2009- Charles Severance



SQLite Manager Plugin



<https://addons.mozilla.org/en-us/firefox/addon/sqlite-manager/>

Terminology

- **Database** - Contains many tables
- **Relation (or table)** - contains tuples and attributes
- **Tuple (or row)** - is a set of fields it generally represents an “object” like a person or a music track
- **Attribute (also column or field)** - One of possibly many elements of data corresponding to the object represented by the row

SI502 - Database

New Open Save Print Import Copy Paste Format Undo Redo AutoSum Sort A-Z Sort Z-A Gallery Toolbox

Sheets Charts SmartArt Graphics WordArt

A B C D

1

Columns / Attributes

TITLE	RATING	LEN	
About to Rock	3	354	Rows /
Who Made Who	4	252	Tuples

Tables / Relations

Tracks Albums Artists Genres +

SQL

- **Structured Query Language** is the language we use to issue commands to the database
 - Create a table
 - Retrieve some data
 - Insert data
 - Delete data

<http://en.wikipedia.org/wiki/SQL>

Common Database Systems

- Three Major Database Management Systems in wide use
 - **Oracle** - Large, commercial, enterprise-scale, very very tweakable
 - **MySQL** - Simpler but very fast and scalable - commercial open source
 - **SqlServer** - Very nice - from Microsoft (also Access)
- Many other smaller projects, free and open source
 - HSQL, **SQLite**, Postgress, ...

SQLite Database Browser

- SQLite is a very popular database - it is free and fast and small
- We have a program to manipulate SQLite databases
 - <https://addons.mozilla.org/en-us/firefox/addon/sqlite-manager/>
- SQLite is embedded in Python and a number of other languages

SQLite is in lots of software...

symbian

 python™



skype™



Microsoft®

McAfee®



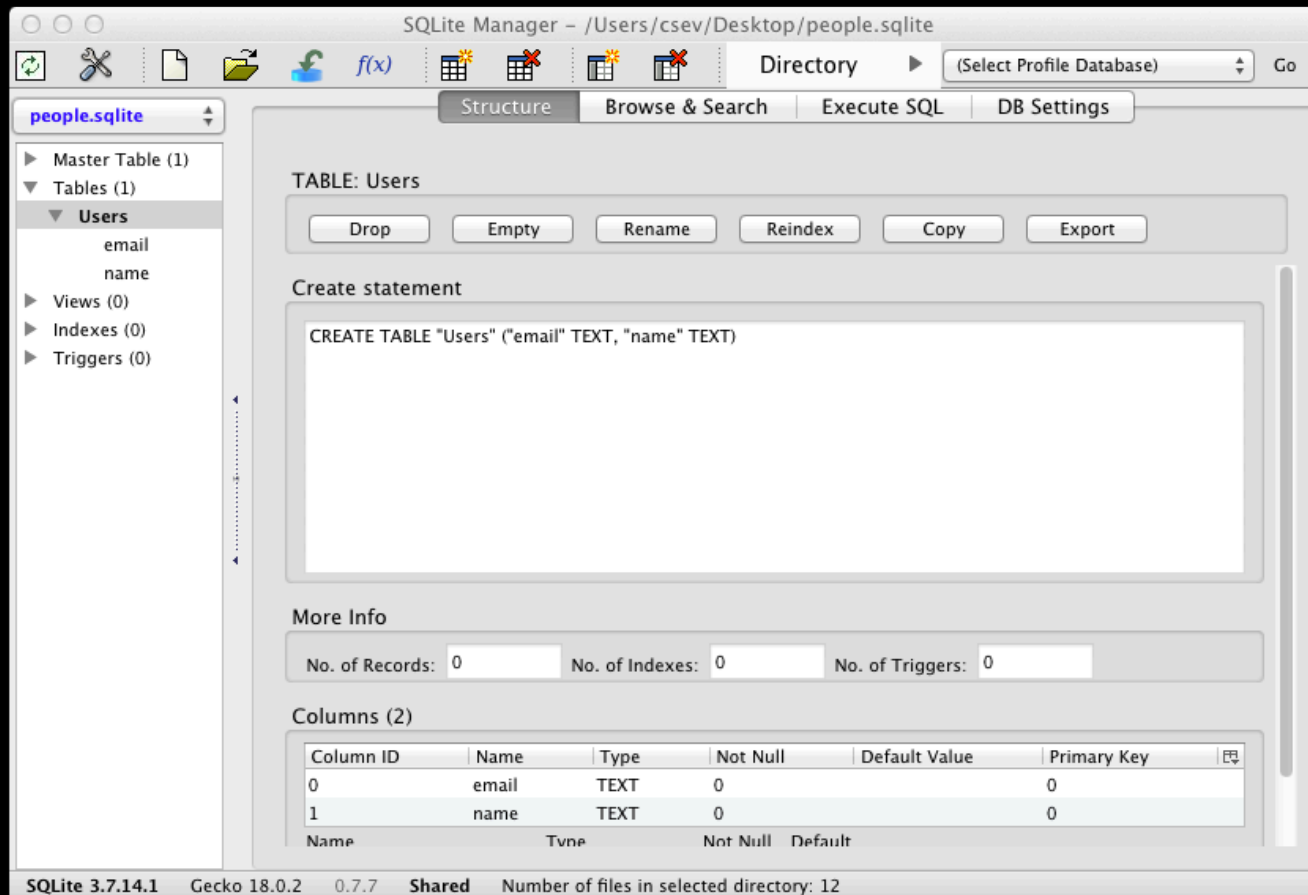
php

Google™

TOSHIBA

 **Sun**
microsystems

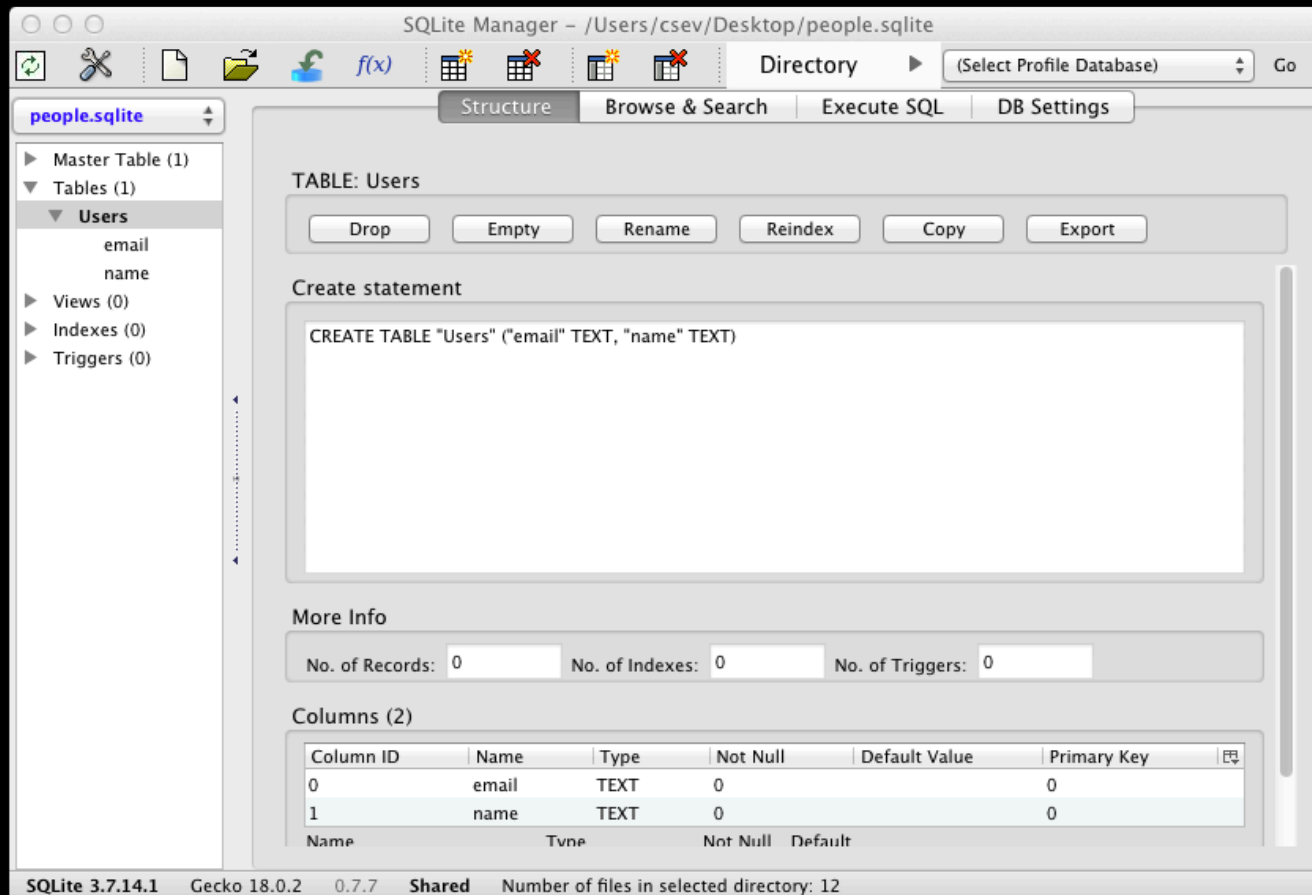
<http://www.sqlite.org/famous.html>



<https://addons.mozilla.org/en-us/firefox/addon/sqlite-manager/>

Start Simple - A Single Table

- Lets make a table of People - with a Name and an E-Mail



CREATE TABLE Users ("email" TEXT, "name" TEXT)

SQLite Manager - /Users/csev/Desktop/people.sqlite

Directory (Select Profile Database) Go

Structure Browse & Search Execute SQL DB Settings

people.sqlite

- Master Table (1)
- Tables (1)
 - Users
 - email
 - name
 - Views (0)
 - Indexes (0)
 - Triggers (0)

TABLE Users Search Show All Add Duplicate Edit Delete

rowid	email	name
1	csev@umich.edu	Chuck
2	botimer@umich.edu	Noah
3	cholma@umich.edu	Caitlin
4	font@umich.edu	Kristen

1 to 4 of 4

SQLite 3.7.14.1 Gecko 18.0.2 0.7.7 Shared Number of files in selected directory: 12 ET: 0 ms

Our table with four rows

SQL

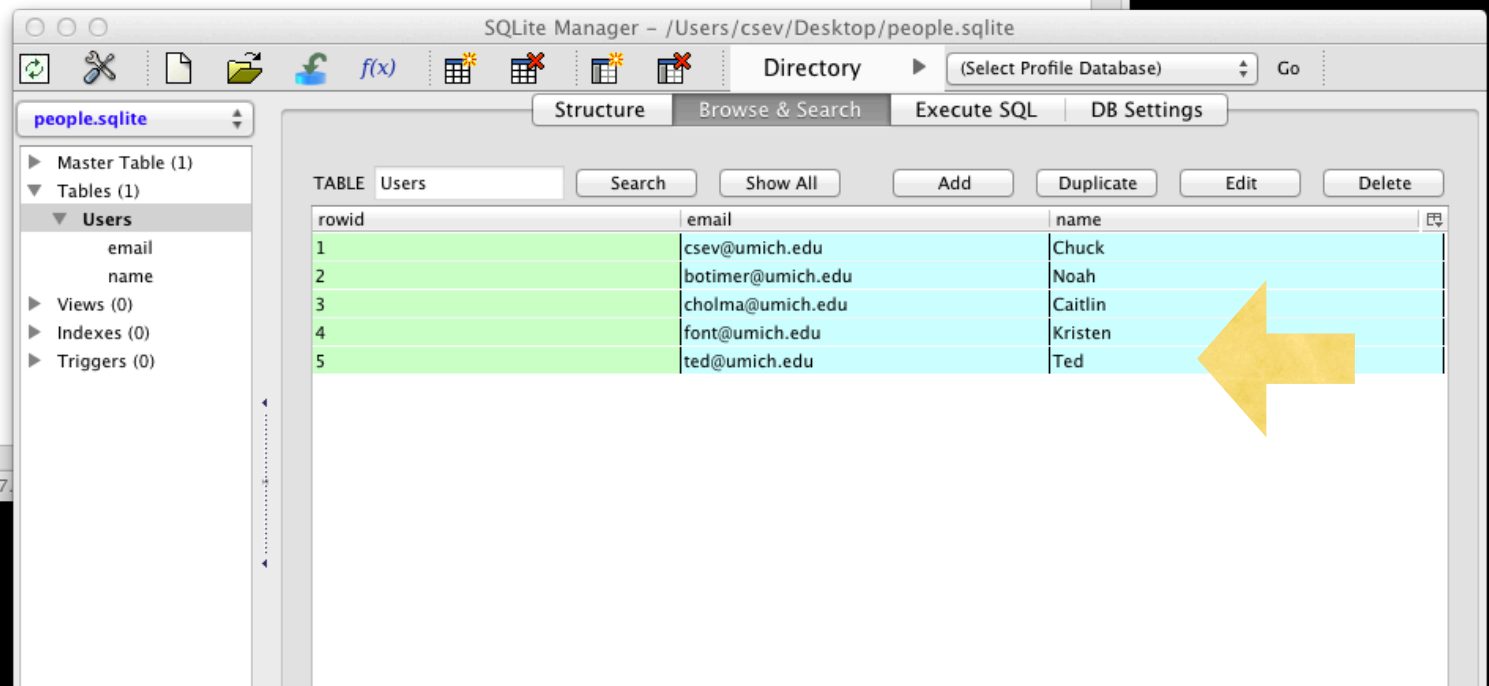
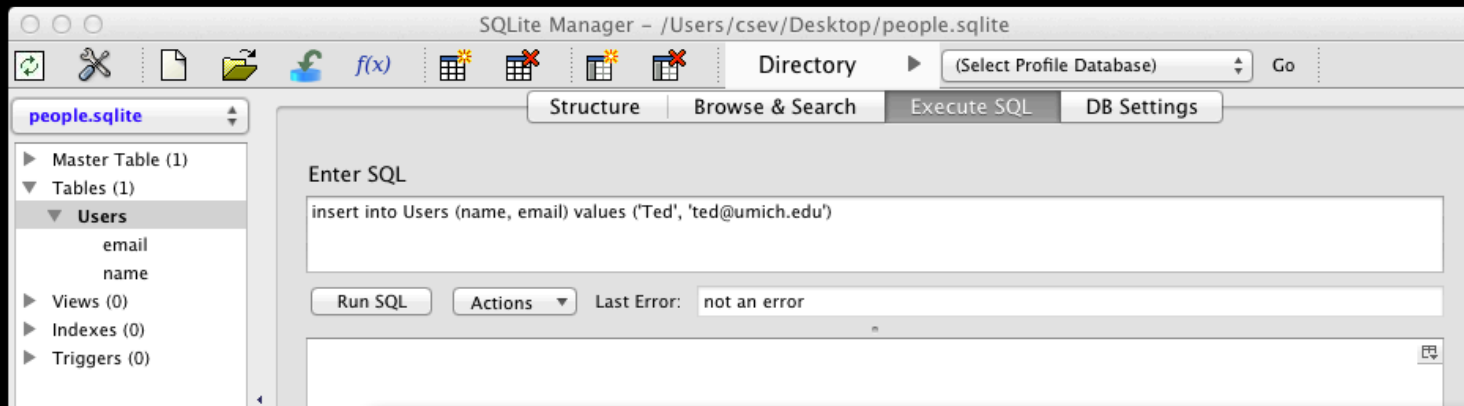
- **Structured Query Language** is the language we use to issue commands to the database
 - Create a table
 - Retrieve some data
 - Insert data
 - Delete data

<http://en.wikipedia.org/wiki/SQL>

SQL Insert

- The Insert statement inserts a row into a table

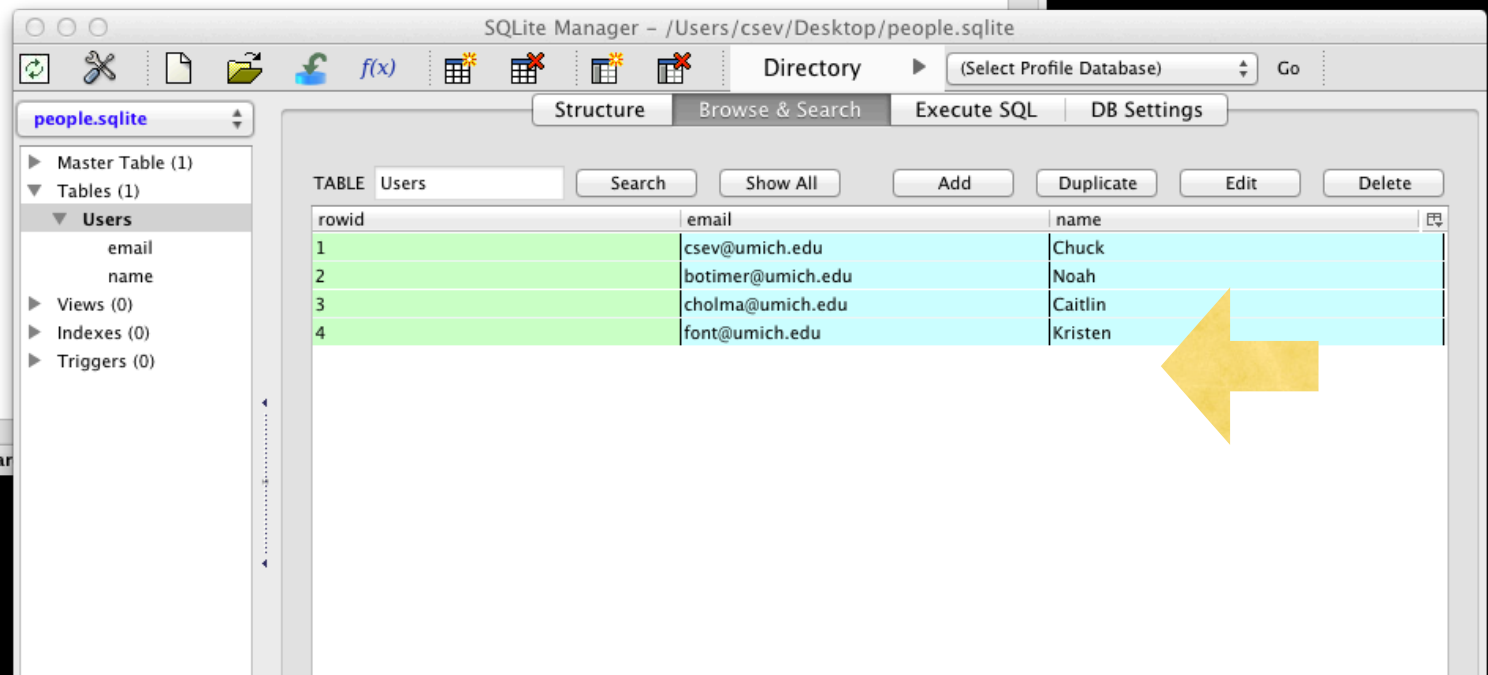
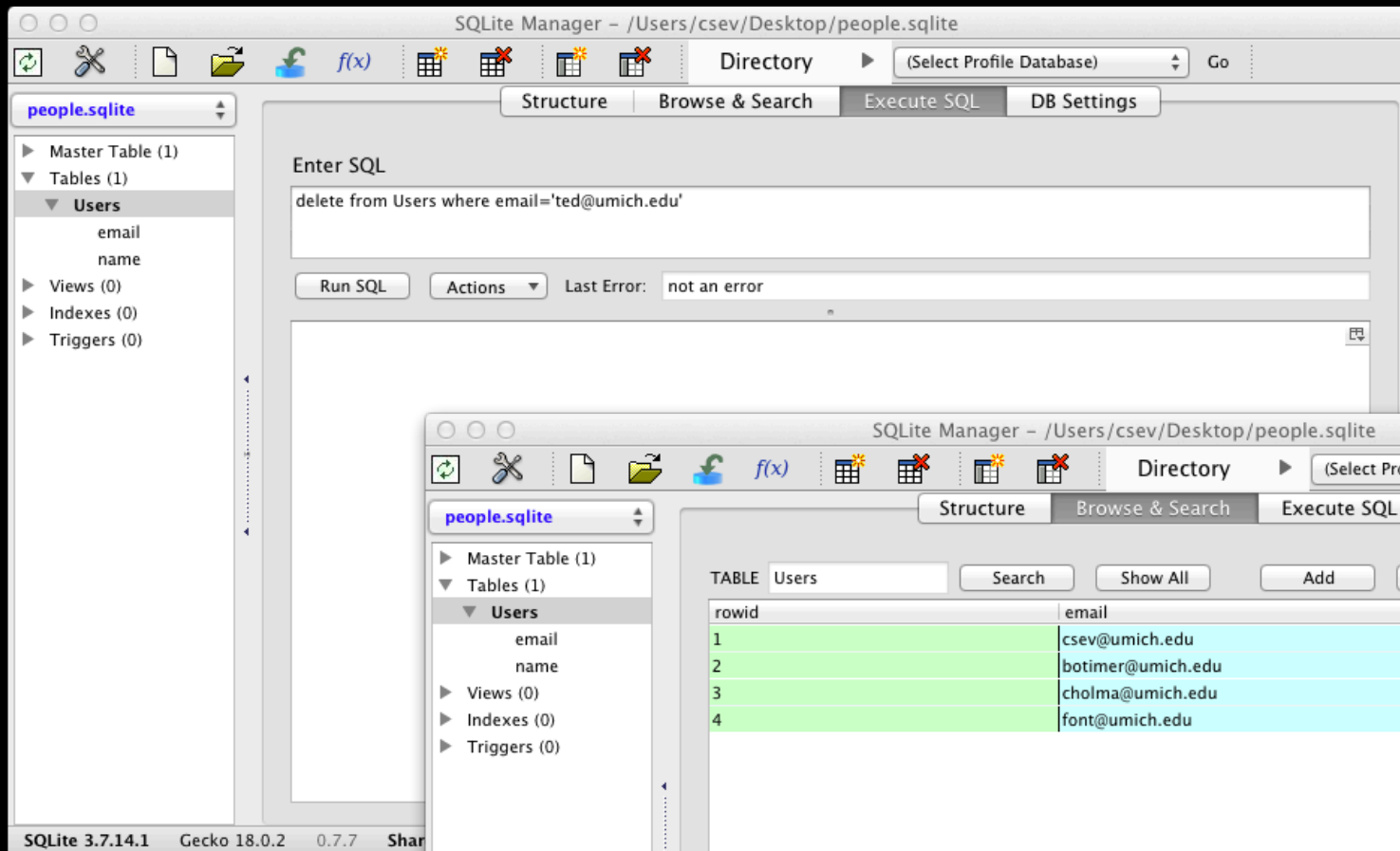
```
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
```



SQL Delete

- Deletes a row in a table based on a selection criteria

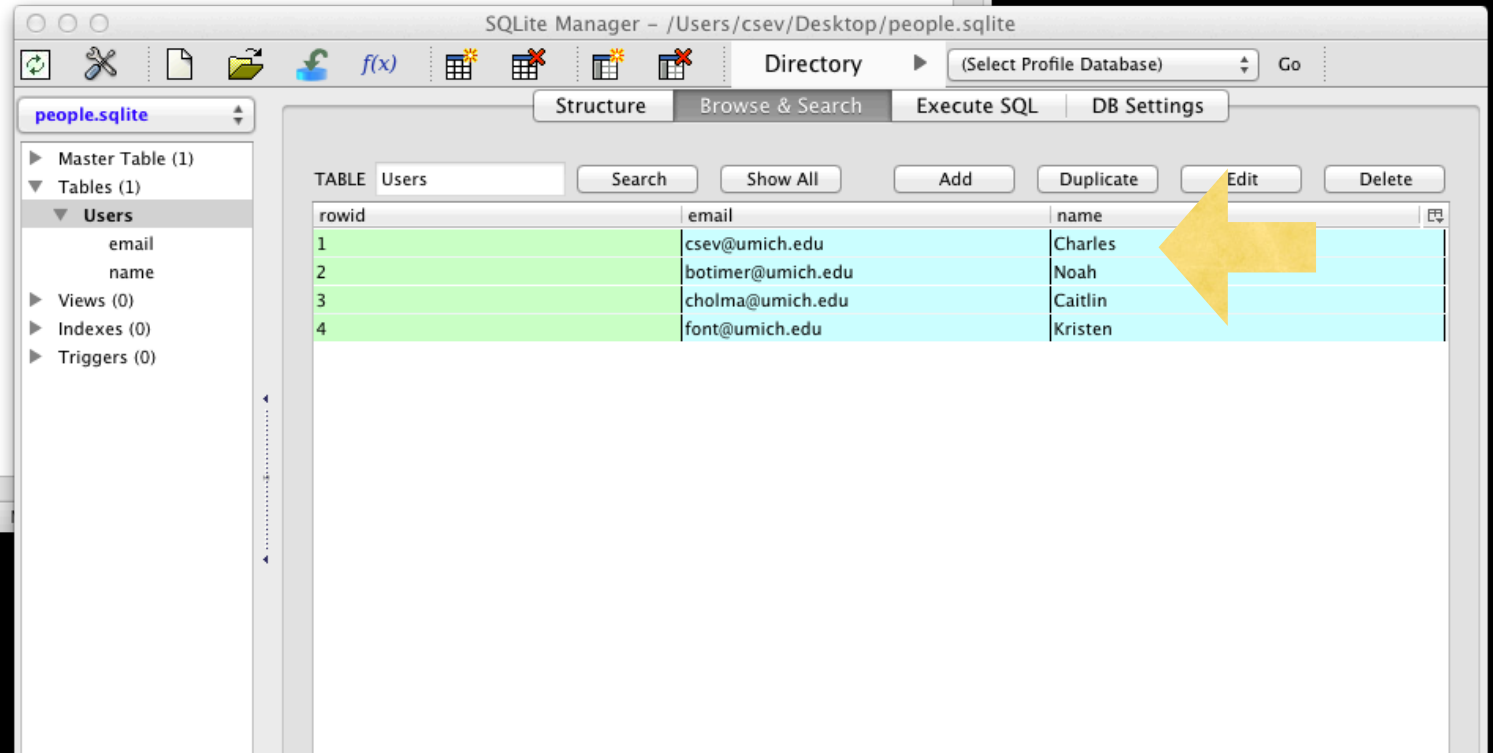
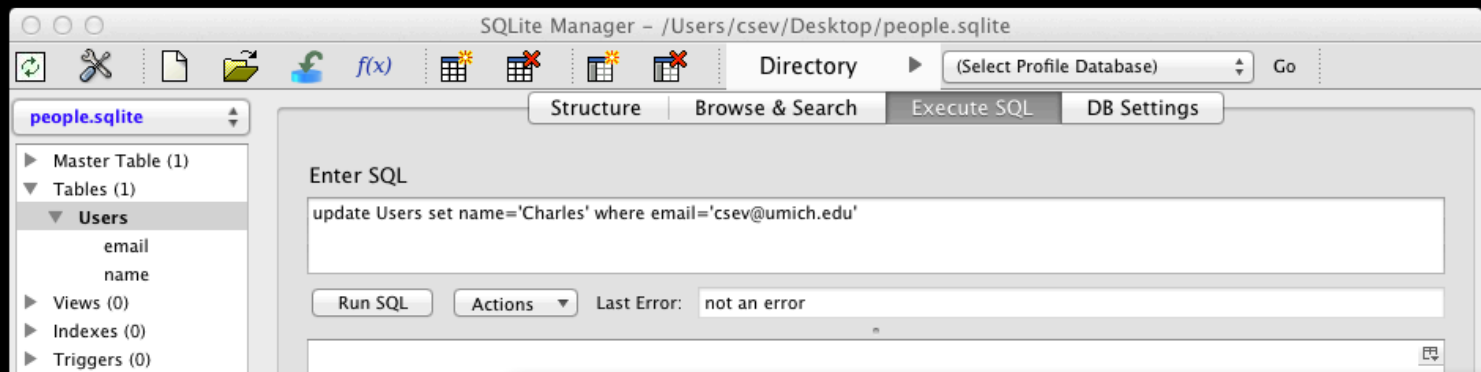
```
DELETE FROM Users WHERE email='ted@umich.edu'
```



SQL: Update

- Allows the updating of a field with a where clause

```
UPDATE Users SET name='Charles' WHERE email='csev@umich.edu'
```



Retrieving Records: Select

- The select statement retrieves a group of records - you can either retrieve all the records or a subset of the records with a WHERE clause

```
SELECT * FROM Users
```

```
SELECT * FROM Users WHERE email='csev@umich.edu'
```

SQLite Manager - /Users/csev/Desktop/people.sqlite

Directory (Select Profile Database) Go

Structure Browse & Search Execute SQL DB Settings

people.sqlite

- Master Table (1)
- Tables (1)
 - Users
 - email
 - name
- Views (0)
- Indexes (0)
- Triggers (0)


Enter SQL

select * from Users

Run SQL Actions Last Error: not an error

email	name
csev@umich.edu	Charles
botimer@umich.edu	Noah
cholma@umich.edu	Caitlin
font@umich.edu	Kristen

SQLite 3.7.14.1 Gecko 18.0.2 0.7.7 Shared Number



SQLite Manager - /Users/csev/Desktop/people.sqlite

Directory (Select Profile Database) Go

Structure Browse & Search Execute SQL DB Settings

people.sqlite

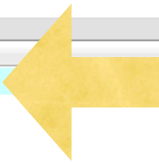
- Master Table (1)
- Tables (1)
 - Users
 - email
 - name
- Views (0)
- Indexes (0)
- Triggers (0)

Enter SQL

select * from Users where email='csev@umich.edu'

Run SQL Actions Last Error: not an error

email	name
csev@umich.edu	Charles



Sorting with ORDER BY

- You can add an **ORDER BY** clause to **SELECT** statements to get the results sorted in ascending or descending order

```
SELECT * FROM Users ORDER BY email
```

```
SELECT * FROM Users ORDER BY name
```

SQLite Manager - /Users/csev/Desktop/people.sqlite

people.sqlite

Master Table (1)

Tables (1)

Users

email

name

Views (0)

Indexes (0)

Triggers (0)

Directory

(Select Profile Database)

Go

StructureBrowse & SearchExecute SQLDB Settings

Enter SQL

select * from Users order by email

Run SQLActionsLast Error: not an error

email	name
botimer@umich.edu	Noah
cholma@umich.edu	Caitlin
csev@umich.edu	Charles
font@umich.edu	Kristen

SQLite 3.7.14.1Gecko 18.0.20.7.7Shared

SQLite Manager - /Users/csev/Desktop/people.sqlite

people.sqlite

Master Table (1)

Tables (1)

Users

email

name

Views (0)

Indexes (0)

Triggers (0)

Directory

(Select Profile Database)

Go

StructureBrowse & SearchExecute SQLDB Settings

Enter SQL

select * from Users order by name

Run SQLActionsLast Error: not an error

email	name
cholma@umich.edu	Caitlin
csev@umich.edu	Charles
font@umich.edu	Kristen
botimer@umich.edu	Noah

SQL Summary

```
CREATE TABLE Users ("email" TEXT, "name" TEXT)
```

```
INSERT INTO Users (name, email) VALUES ('Ted', 'ted@umich.edu')
```

```
DELETE FROM Users WHERE email='ted@umich.edu'
```

```
UPDATE Users SET name='Charles' WHERE email='csev@umich.edu'
```

```
SELECT * FROM Users
```

```
SELECT * FROM Users WHERE email='csev@umich.edu'
```

```
SELECT * FROM Users ORDER BY email
```

Auto Generated Columns

- Sometimes we want a "row identifier" or "key" so we can easily reference or point to a row in another table
- JOIN is the SQL notion of connecting tables

```
CREATE TABLE IF NOT EXISTS Pages  
(id INTEGER PRIMARY KEY, url TEXT UNIQUE, html TEXT,  
old_rank REAL, new_rank REAL)
```