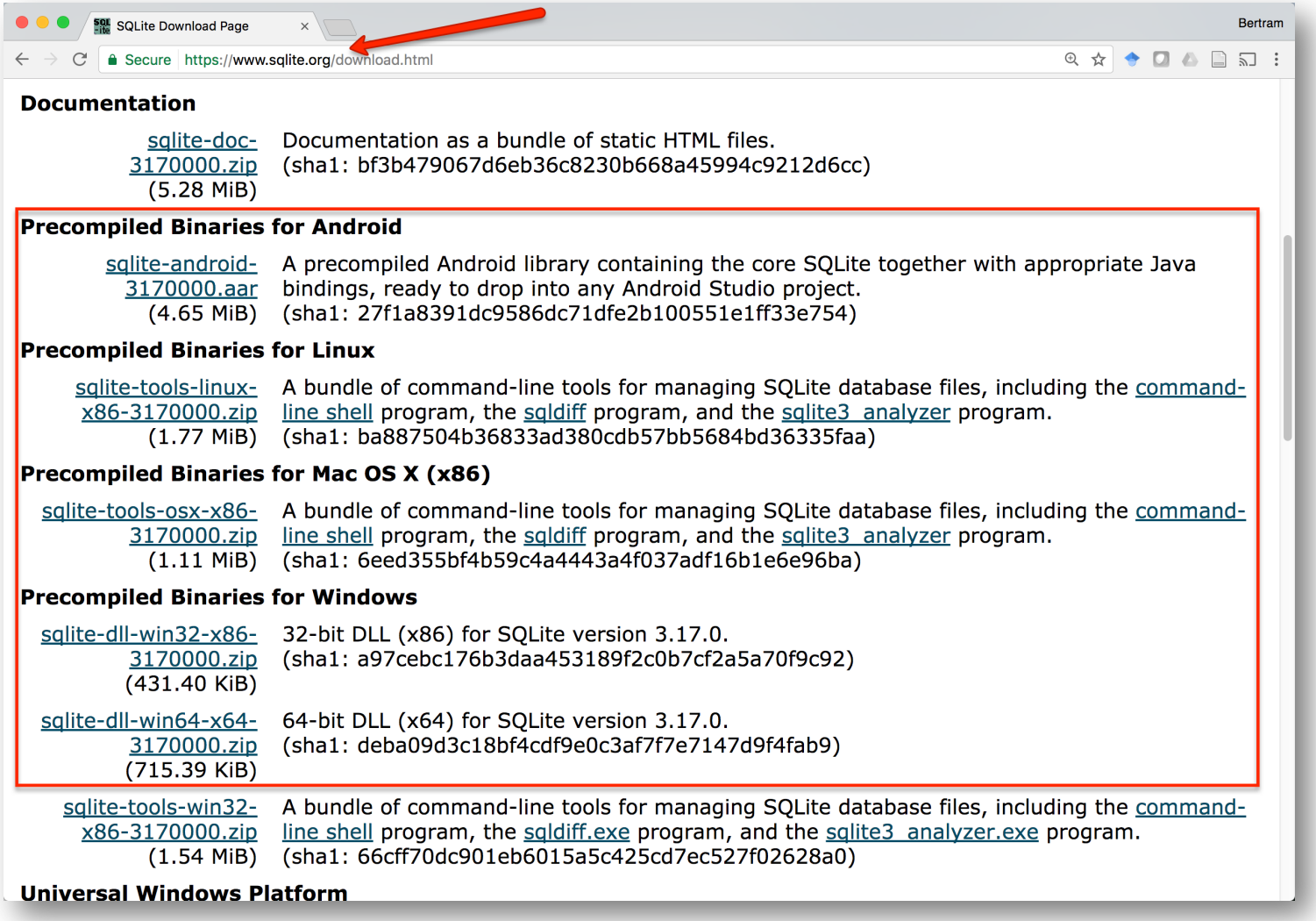
Theory & Practice of Data Cleaning

Installing the SQLite System



Documentation

sqlite-doc-3170000.zip (5.28 MiB) — Documentation as a bundle of static HTML files. (sha1: bf3b479067d6eb36c8230b668a45994c9212d6cc)

Precompiled Binaries for Android

sqlite-android-3170000.aar (4.65 MiB) — A precompiled Android library containing the core SQLite together with appropriate Java bindings, ready to drop into any Android Studio project. (sha1: 27f1a8391dc9586dc71dfe2b100551e1ff33e754)

Precompiled Binaries for Linux

sqlite-tools-linux-x86-3170000.zip (1.77 MiB) — A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (sha1: ba887504b36833ad380cdb57bb5684bd36335faa)

Precompiled Binaries for Mac OS X (x86)

sqlite-tools-osx-x86-3170000.zip (1.11 MiB) — A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (sha1: 6eed355bf4b59c4a4443a4f037adf16b1e6e96ba)

Precompiled Binaries for Windows

sqlite-dll-win32-x86-3170000.zip (431.40 KiB) — 32-bit DLL (x86) for SQLite version 3.17.0. (sha1: a97cebc176b3daa453189f2c0b7cf2a5a70f9c92)

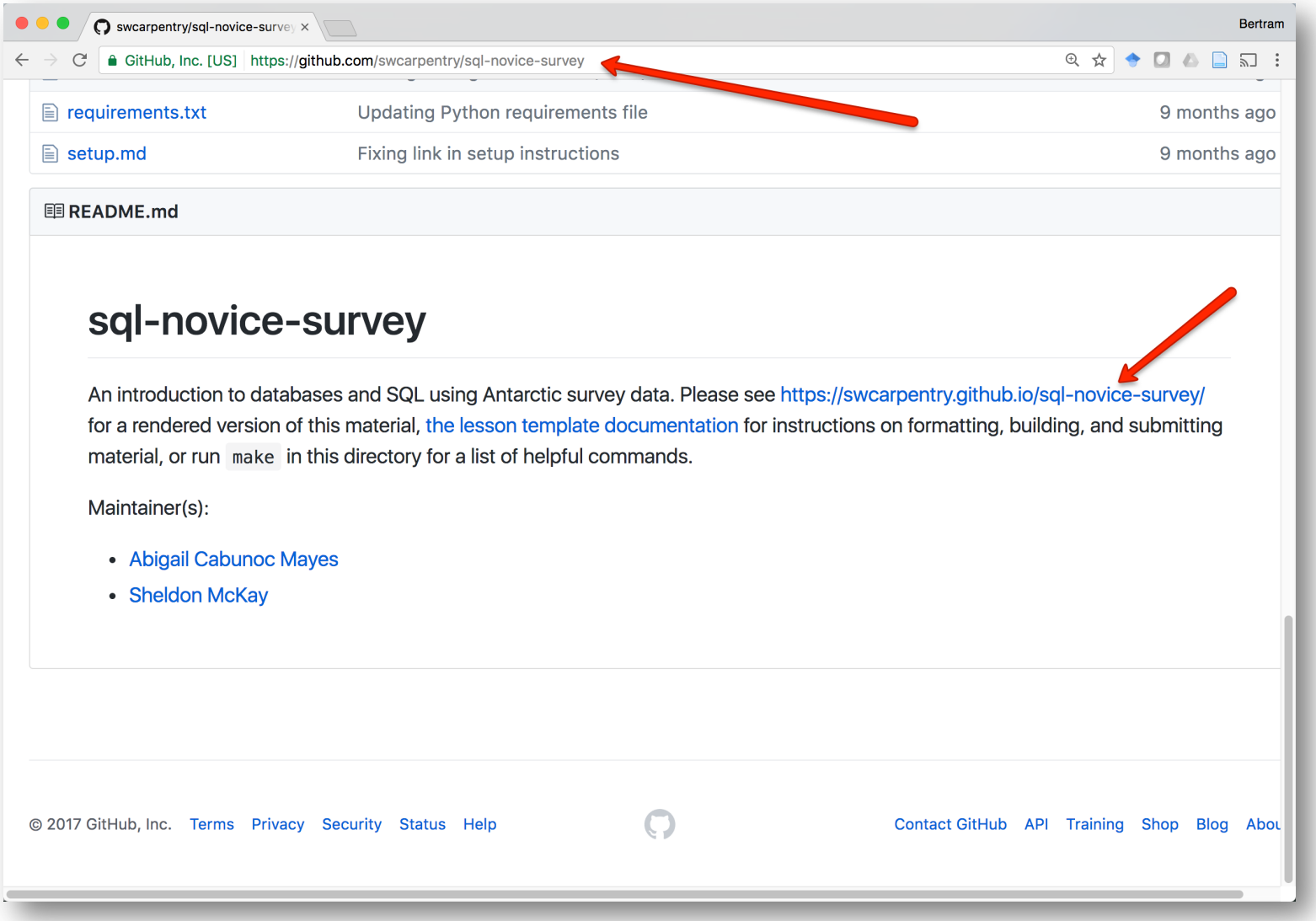
sqlite-dll-win64-x64-3170000.zip (715.39 KiB) — 64-bit DLL (x64) for SQLite version 3.17.0. (sha1: deba09d3c18bf4cdf9e0c3af7f7e7147d9f4fab9)

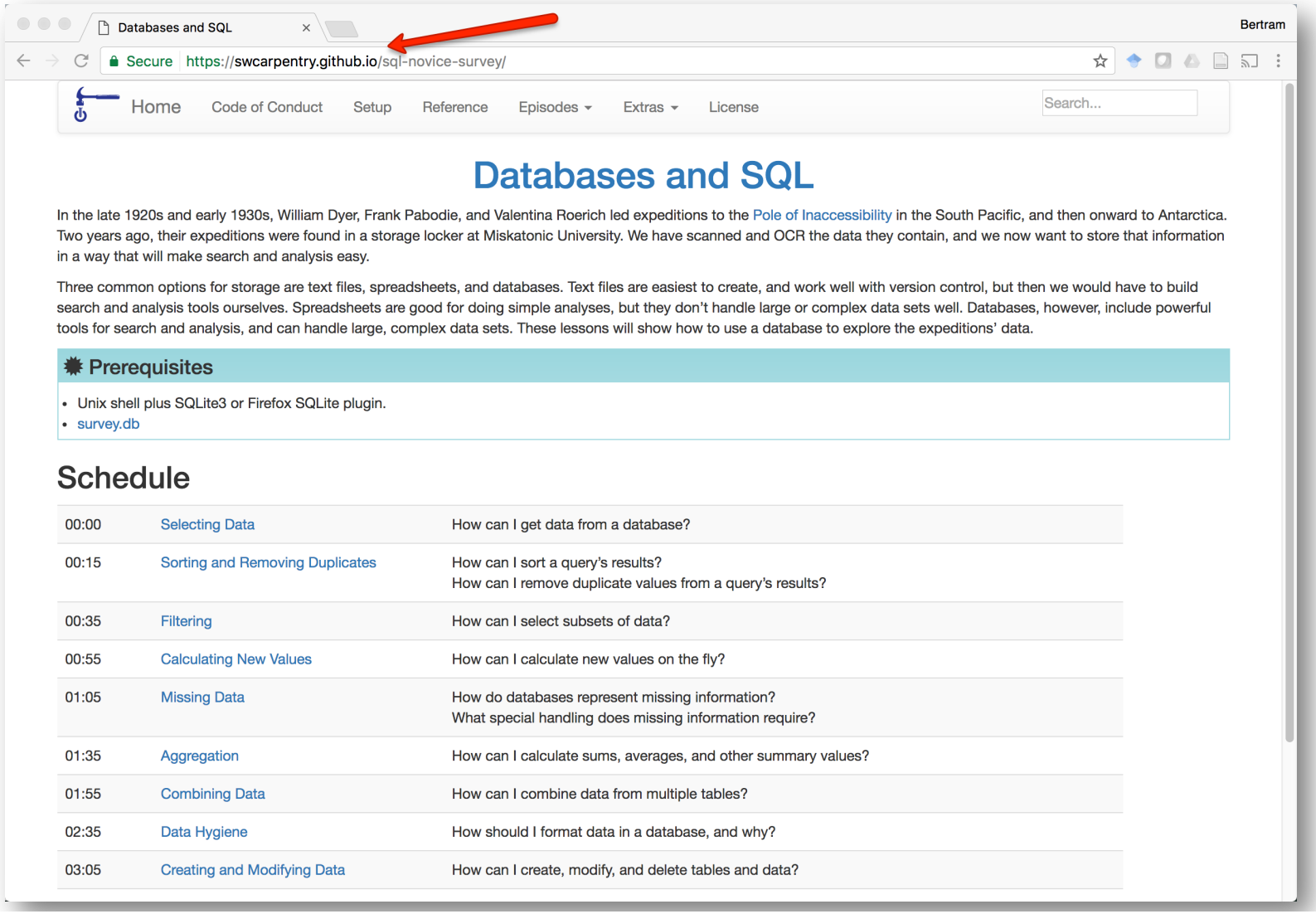
sqlite-tools-win32-x86-3170000.zip (1.54 MiB) — A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program. (sha1: 66cff70dc901eb6015a5c425cd7ec527f02628a0)

Universal Windows Platform

Theory & Practice of Data Cleaning

SQL(ite) by Example


swcarpentry/sql-novice-survey
Bertram
GitHub, Inc. [US] https://github.com/swcarpentry/sql-novice-survey
requirements.txt
Updating Python requirements file
9 months ago
setup.md
Fixing link in setup instructions
9 months ago
README.md
sql-novice-survey
An introduction to databases and SQL using Antarctic survey data. Please see https://swcarpentry.github.io/sql-novice-survey/ for a rendered version of this material, the lesson template documentation for instructions on formatting, building, and submitting material, or run make in this directory for a list of helpful commands.
Maintainer(s):
Abigail Cabunoc Mayes
Sheldon McKay
© 2017 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub API Training Shop Blog Abou


Databases and SQL
Bertram
Secure https://swcarpentry.github.io/sql-novice-survey/
Home Code of Conduct Setup Reference Episodes Extras License
Search...
Databases and SQL
In the late 1920s and early 1930s, William Dyer, Frank Pabodie, and Valentina Roerich led expeditions to the Pole of Inaccessibility in the South Pacific, and then onward to Antarctica. Two years ago, their expeditions were found in a storage locker at Miskatonic University. We have scanned and OCR the data they contain, and we now want to store that information in a way that will make search and analysis easy.
Three common options for storage are text files, spreadsheets, and databases. Text files are easiest to create, and work well with version control, but then we would have to build search and analysis tools ourselves. Spreadsheets are good for doing simple analyses, but they don't handle large or complex data sets well. Databases, however, include powerful tools for search and analysis, and can handle large, complex data sets. These lessons will show how to use a database to explore the expeditions' data.
Prerequisites
Unix shell plus SQLite3 or Firefox SQLite plugin.
survey.db
Schedule
00:00 Selecting Data How can I get data from a database?
00:15 Sorting and Removing Duplicates How can I sort a query's results? How can I remove duplicate values from a query's results?
00:35 Filtering How can I select subsets of data?
00:55 Calculating New Values How can I calculate new values on the fly?
01:05 Missing Data How do databases represent missing information? What special handling does missing information require?
01:35 Aggregation How can I calculate sums, averages, and other summary values?
01:55 Combining Data How can I combine data from multiple tables?
02:35 Data Hygiene How should I format data in a database, and why?
03:05 Creating and Modifying Data How can I create, modify, and delete tables and data?

SQL: Structured Query Language

- Based on **relational algebra** and tuple **relational calculus**.
- Consists of a data definition language, data manipulation language, and data control language.
- Scope of SQL includes data **insert**, **query**, **update** and **delete**, schema creation and modification, and data access control.
- SQL is a **declarative** language (4GL), but also includes procedural elements.

SQL: Structured Query Language

- A typical SQL query has the form

  select $A_1, A_2, \ldots, A_n$
  from $r_1, r_2, \ldots, r_k$
  where P

  - A_is represent attributes
  - r_is represent relations
  - P is a predicate

- This query is equivalent to the relational algebra expression

  $\pi_{A_1,A_2,\ldots,A_n}(\sigma_P(r_1 \times r_2 \times \ldots \times r_k))$

- The result of an SQL query is a relation (set of tuples) with a schema defined through the attributes A_is.

- The **select** clause corresponds to the projection operation of the relational algebra; it is used to list the attributes to be output in a query result.

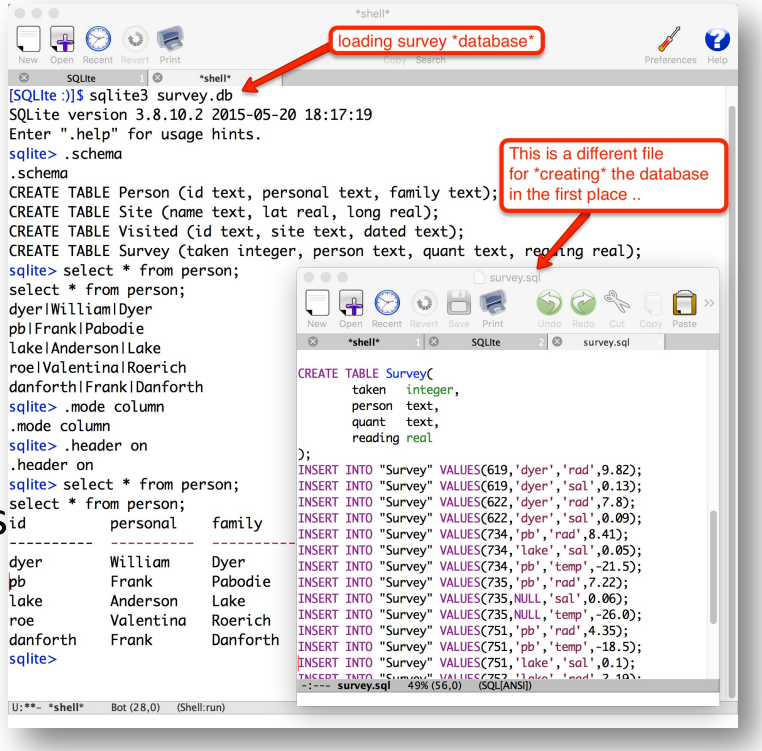
Theory & Practice of Data Cleaning

SQLite by Example (cont'd)

Queries and Simple Data Cleaning in SQLite

Software Carpentry: Databases & SQL

1. Selecting Data
2. Sorting and Removing Duplicates
3. Filtering
4. Calculating New Values
5. Missing Data
6. Aggregation
7. Combining Data
8. Data Hygiene
9. Creating and Modifying Data
10. Programming with Databases



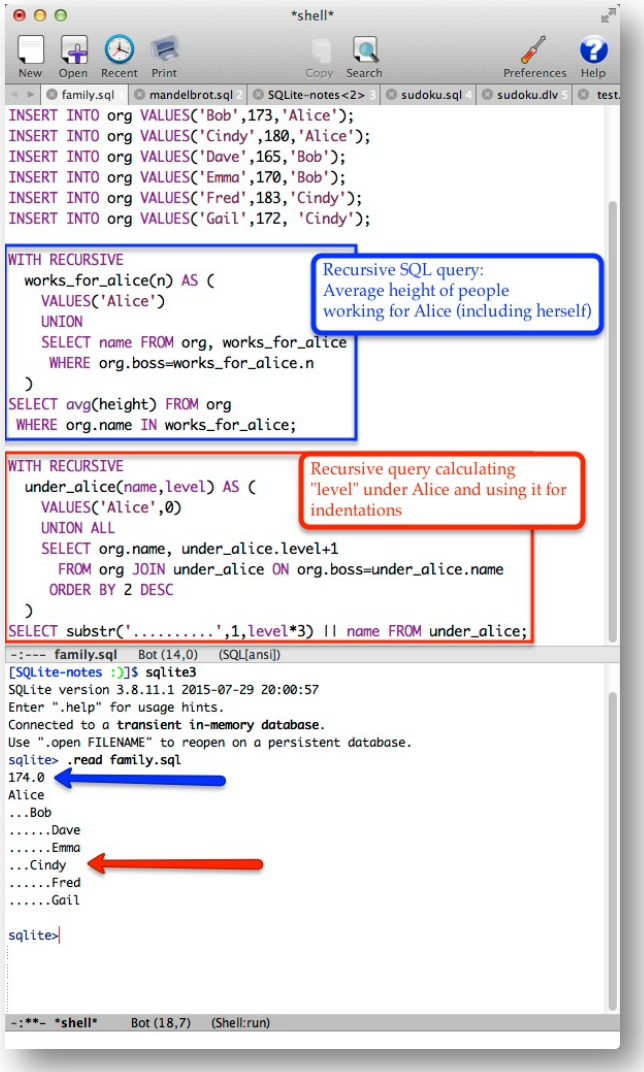
Theory & Practice of Data Cleaning

Recursion in SQL

Recursion in SQL

- Traditionally, SQL, First-order Predicate Logic, Relational Algebra, don't allow *recursion* (or iteration) => Datalog to the rescue!
- Since SQL:1999 standard recursion is allowed using CTEs (*common table expressions*):

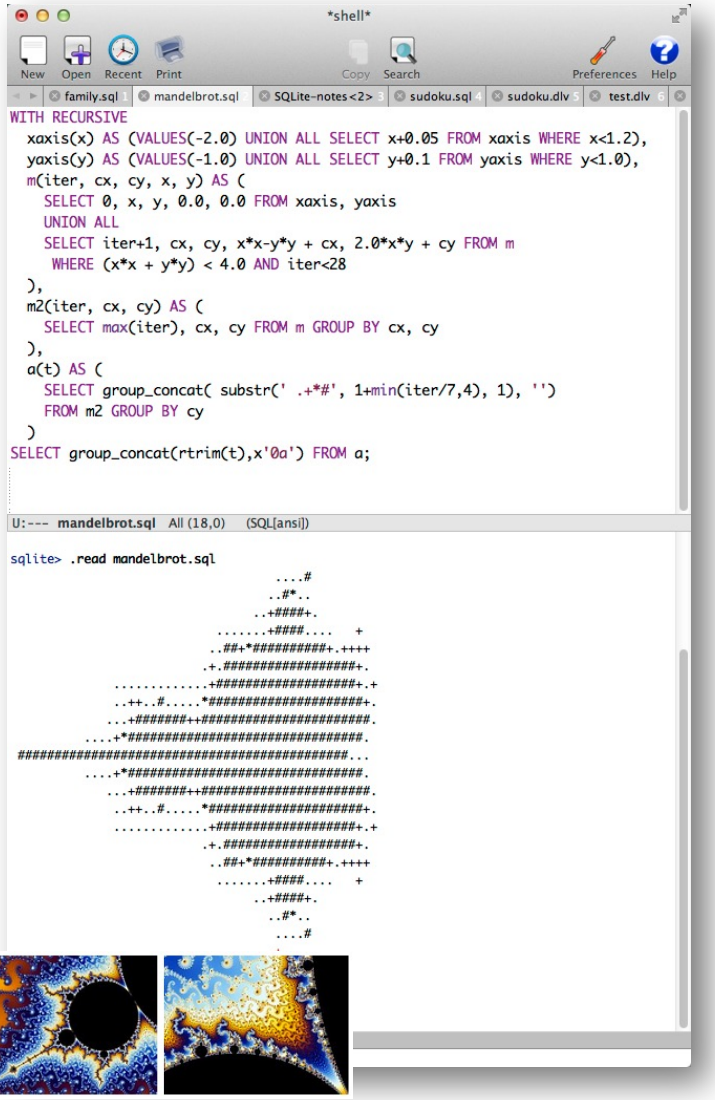
```
WITH RECURSIVE (…)
SELECT …
```

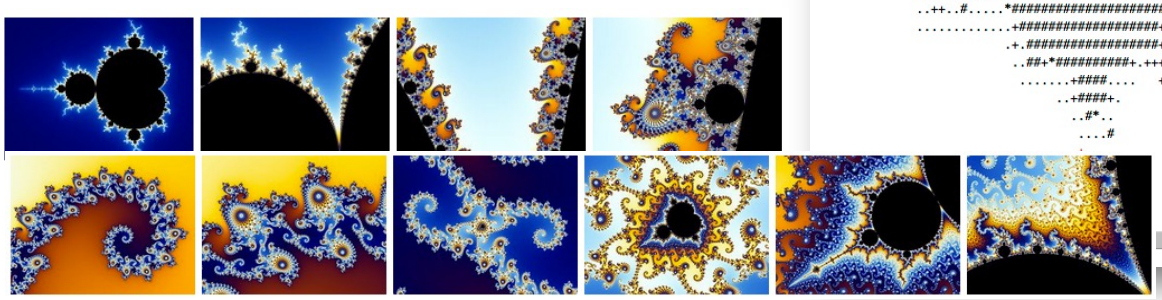


Recursion in SQL

- ASCII-art version of famous Mandelbrot fractal in SQL with recursion (right)
- Zooming into the Mandelbrot set (below)

  Source: [Wikipedia]





Crazy, Recursive SQL (cont'd)

- Sudoku in SQL!
- Basic idea:
  - For each blank cell, find a digit 1..9 that doesn't violate the complex IC (another "denial constraint"):

```
WHERE …
AND NOT EXISTS …
```