Principles of Software Engineering and Data Bases

Davide Yi Xian Hu
Email: davideyi.hu@polimi.it

Date: 26 November 2024

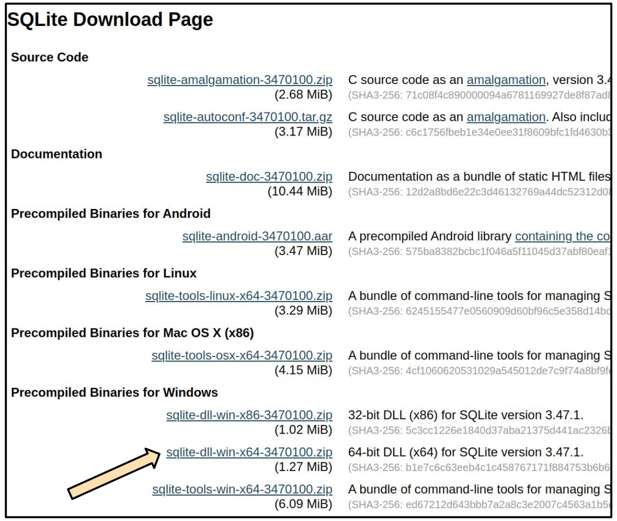
Exercise Lecture: 05 - SQLite and FireBase

POLITECNICO
MILANO 1863

Setup - Windows 10

Download SQLite from the SQLite Download Page
https://www.sqlite.org/download.html

Extract the downloaded files
and add the directory to your system's PATH
for easy command-line access.



SQLite Download Page

Source Code	
sqlite-amalgamation-3470100.zip (2.68 MiB)	C source code as an amalgamation, version 3. (SHA3-256: 71c08f4c890000094a6781169927de8f87ad
sqlite-autoconf-3470100.tar.gz (3.17 MiB)	C source code as an amalgamation. Also inclu (SHA3-256: c6c1756fbeb1e34e0ee31f8609bfc1fd4630b
Documentation	
sqlite-doc-3470100.zip (10.44 MiB)	Documentation as a bundle of static HTML files (SHA3-256: 12d2a8bd6e22c3d46132769a44dc52312d0
Precompiled Binaries for Android	
sqlite-android-3470100.aar (3.47 MiB)	A precompiled Android library containing the co (SHA3-256: 575ba8382bcbc1f046a5f11045d37abf80eaf
Precompiled Binaries for Linux	
sqlite-tools-linux-x64-3470100.zip (3.29 MiB)	A bundle of command-line tools for managing S (SHA3-256: 6245155477e0560909d60bf96c5e358d14bc
Precompiled Binaries for Mac OS X (x86)	
sqlite-tools-osx-x64-3470100.zip (4.15 MiB)	A bundle of command-line tools for managing S (SHA3-256: 4cf1060620531029a545012de7c9f74a8bf9f
Precompiled Binaries for Windows	
sqlite-dll-win-x86-3470100.zip (1.02 MiB)	32-bit DLL (x86) for SQLite version 3.47.1. (SHA3-256: 5c3cc1226e1840d37aba21375d441ac2326
sqlite-dll-win-x64-3470100.zip (1.27 MiB)	64-bit DLL (x64) for SQLite version 3.47.1. (SHA3-256: b1e7c6c63eeb4c1c458767171f884753b6b6
sqlite-tools-win-x64-3470100.zip (6.09 MiB)	A bundle of command-line tools for managing S (SHA3-256: ed67212d643bbb7a2a8c3e2007c4563a1b5

Setup - Linux Ubuntu

```
# Install SQLite
sudo apt update
sudo apt install sqlite3

# Install Python and PIP
sudo apt install python3 python3-pip

# Install SQLite
pip3 install sqlalchemy
```

Setup - First Steps

```python
import sqlite3

# Creates an in-memory database
conn = sqlite3.connect(":memory:")

cursor = conn.cursor()
cursor.execute("SELECT sqlite_version();")

print("SQLite version:", cursor.fetchone())

conn.close()
```

Setup - First Steps

```python
import sqlite3

# Creates a file-based database
conn = sqlite3.connect("database.db")

cursor = conn.cursor()
cursor.execute("SELECT sqlite_version();")

print("SQLite version:", cursor.fetchone())

conn.close()
```

Exercise 1 - Library Management System

Build a Database for a Library

Create a database with two tables:

Authors: Contains information about book authors.

- author_id (Primary Key, Auto Increment)
- name (Author's name)
- email (Optional)

Exercise 1 - Library Management System

Build a Database for a Library

Create a database with two tables:

Books: Contains information about books in the library.

- book_id (Primary Key, Auto Increment)
- title (Book title)
- author_id (Foreign Key referencing Authors.author_id)
- published_year (Year the book was published)

Exercise 1 - Library Management System

Populate the Database

Insert initial data into the database:

- Add at least two authors
  (e.g., J.K. Rowling, George R.R. Martin)

- Add at least four books
  (e.g., Harry Potter and the Philosopher's Stone,
  A Game of Thrones)

Exercise 1 - Library Management System

Build a Screen to View This Information

Displays the list of books in the database:

- Title
- Author's name
- Published year

Adds a Refresh button to reload the book list from the database.

Exercise 1 - Library Management System

Build a Screen to Add a Book

Open a new window with fields to input:

- Title of the book, Author's name, Published year.

Check that the author exists in the Authors table:

- If the author exists:
    - Add the new book to the Books table.
- If the author does not exist:
    - Show an error message.

Exercise 2 - IRIS Dataset

https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data

The Iris dataset contains the following columns:

- **sepal_length** (float): Length of the sepal in cm.
- **sepal_width** (float): Width of the sepal in cm.
- **petal_length** (float): Length of the petal in cm.
- **petal_width** (float): Width of the petal in cm.
- **species** (string): Name of the iris flower species (Iris-setosa, Iris-versicolor, Iris-virginica).

Exercise 2 - IRIS Dataset

- Load the CSV dataset using Pandas.
- Convert the Pandas dataframe into SQLite db.
- Find the average petal length for each species (in SQL).
- Find the average petal length for each species (in Pandas).
- Find all records where the sepal length is greater than 7.0 (in SQL and Pandas).
- Data Cleaning
- Visualization

Exercise 3 - NewYork Flights 13

https://raw.githubusercontent.com/vaibhavwalvekar/NYC-Flights-2013-Dataset-Analysis/refs/heads/master/flights.csv

- Total Rows: 336,776 (one row per flight)
- Total Columns: 19
- Airports Covered:
  - JFK (John F. Kennedy International Airport)
  - LGA (LaGuardia Airport)
  - EWR (Newark Liberty International Airport)

Exercise 3 - NewYork Flights 13

- Load the CSV dataset using Pandas.
- Convert the Pandas dataframe into SQLite db.
- Find (with SQL and Pandas):
  - Find the Top 5 Airlines with the Most Delayed Flights
  - Find the Airport with the Longest Average Arrival Delay
- Find and visualize the following information:
  - Analyze Monthly Average Arrival Delays Over the Year