VE472 — Methods and tools for big data

Lab 3

Manuel — UM-JI (Summer 2022)

Goals of the lab

- install and configure a sqlite3 database
- import data into a sqlite3 database
- Interaction with sqlite3 in Java / Python

1 Introduction

In this lab we will provide you a overview of sql language and how to interact it with other popular languages like Java and Python.

Your friend Reapor Rich finally established the cinema in Singapore and it was a big success with the help of the seat management system designed by you. However, after a new survey of the customers, it is shown that the poor selections of the movies may be a bottleneck of the cinema popularity. Now Reapor is finding you again to do some data analysis of the movies and choose some excellent ones for the *Reapor Movie's Week*.

Before starting the analysis, one of the most essential things to do is to build a database of all the movies. Reapor found that IMDb (Internet Movie Database) Datasets can provide sufficient data for the analysis, but the data are all in `tsv` format, which could not be indexed and queried quickly by the algorithms.

In order to solve this problem, you decided to use SQLite as your sql database and build a relational database with the data provided. Then you would like to write some sql queries to verify the dataset.

2 SQLite Environment Setup

2.1 What is SQLite?

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

2.2 Installation

2.2.1 Windows

Windows is **not recommended** in this course. If you really want to use SQLite on Windows, download sqlite-tools-win32-xxx (`https://www.sqlite.org/2019/sqlite-tools-win32-x86-3290000.zip`) from `https://www.sqlite.org/download.html`.

2.2.2 Linux

- Debian and derived distribution, e.g. Ubuntu, Linux Mint.

```sh
sh $ sudo apt update
sh $ sudo apt install sqlite3
```

- Arch Linux and derived distributions, e.g. Manjaro.

```sh
sh $ sudo pacman -S sqlite
```

  Check ArchWiki for more information.

- Other: use a proper search engine and share your findings such that we can update this guide.

2.2.3 Mac OS

With Homebrew, you can simply install SQLite by

```sh
sh $ brew install sqlite3
```

2.3 SQLite Database Creation

We first make a directory `var`, which is commonly used to store data, and initialize a SQLite database called `imdb` in it, and finally exit the database with the command `.quit`.

```sh
sh $ mkdir var
sh $ sqlite3 var/imdb.sqlite3
sh $ .quit
```

2.4 SQLite Table Creation and Data Import

Take the dataset file `name.basics.tsv` as an example, we can find these information on the details page:

- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)—name by which the person is most often credited
- birthYear —in YYYY format
- deathYear —in YYYY format if applicable, else `'\N'`
- primaryProfession (array of strings)—the top-3 professions of the person
- knownForTitles (array of tconsts) —titles the person is known for

Then we can create a table in the SQLite shell executed by

```sh
sh $ sqlite3 var/imdb.sqlite3
```

```
create table name
(
    nconst varchar(10) not null,
    primaryName text not null,
    birthYear varchar(4) not null,
```

```
    deathYear varchar(4) not null,
    primaryProfession text not null,
    knownForTitles text not null,
    primary key(nconst)
);
```

Finally, import the `tsv` file into the created table `name`, you need to remove the first line (header) of the file and set the separator as tab in SQLite.

```
sh $ sed -i '1d' name.basics.tsv
```

```
.separator "\t"
.import name.basics.tsv name
```

The process may take one or two minutes since the dataset is about several hundred MBs large. You will find some warnings about unescaped " character, and you may also fail in one line because there is a bug in the dataset there. You can ignore such warnings as most of data are in the correct format defined and ready to be used.

Similarly, create tables and import the rest `tsv` files in `IMDb`, and the database is then initialized successfully.

3 Verifying the Data

Once the database is up, you need to make some simple queries to verify whether the `IMDb` dataset was imported correctly.

Using basic queries with the database to find out the results of the following:

- The oldest movie
- The the longest movie in 2009
- The year with the most movies
- The name of the person who contains in the most movies
- The principal crew of the movie with highest average ratings and more than 500 votes
- (Advanced) The count of each Pair<BirthYear, DeathYear> of the people

4 Interaction with SQLite in Java / Python

After the data are verified, Reapor found that some of the data columns are in the format of comma separated string, which can not be easily queried and joined with some SQL statements.

After searching on the Internet, he found it possible to split the values in a pure SQL way, but it's beyond his (and maybe your) knowledge of basic SQL, so he decided to use some upper level programming languages like `Java` and `Python` to deal with this.

For example, for the `primaryProfession` column in the table `name`, you can split it into a new table `name_profession`, which can be created by

```
CREATE TABLE name_profession
(
    nconst varchar(10) not null,
    profession text not null,
    foreign key(nconst) references name(nconst)
)
```

Then use either `Java` or `Python` to select all rows from the table `name`, process the `primaryProfession` column, and write the results to the table `name_profession`.

4.1 Java

Oracle JDBC is wdiely used in `Java` to interact with various kinds of databases. For this lab, you can use the SQLite JDBC library.

Here is some sample code ported from `https://github.com/xerial/sqlite-jdbc#usage`.

sample.java

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Sample
{
  public static void main(String[] args)
  {
    Connection connection = null;
    try
    {
      // create a database connection
      connection = DriverManager.getConnection("jdbc:sqlite:sample.db");
      Statement statement = connection.createStatement();
      statement.setQueryTimeout(30);  // set timeout to 30 sec.

      statement.executeUpdate("drop table if exists person");
      statement.executeUpdate("create table person (id integer, name string)");
      statement.executeUpdate("insert into person values(1, 'leo')");
      statement.executeUpdate("insert into person values(2, 'yui')");
      ResultSet rs = statement.executeQuery("select * from person");
      while(rs.next())
      {
        // read the result set
        System.out.println("name = " + rs.getString("name"));
        System.out.println("id = " + rs.getInt("id"));
      }
    }
    catch(SQLException e)
    {
```

```
          // if the error message is "out of memory",
          // it probably means no database file is found
          System.err.println(e.getMessage());
        }
        finally
        {
          try
          {
            if(connection != null)
              connection.close();
          }
          catch(SQLException e)
          {
            // connection close failed.
            System.err.println(e.getMessage());
          }
        }
      }
    }
```

You can also take a look at the JDBC Tutorials and Oracle JDBC Documentation.

4.2 Python

In `Python`, there is a built-in module `sqlite3` which provide a simple interface to the SQLite database. The documentation of the module can be found in `https://docs.python.org/3/library/sqlite3.html`.

Here is some sample code ported from the documentation above.

sample.py

```
import sqlite3
conn = sqlite3.connect('example.db')

c = conn.cursor()

# Create table
c.execute('''CREATE TABLE stocks
             (date text, trans text, symbol text, qty real, price real)''')

# Insert a row of data
c.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

# Save (commit) the changes
conn.commit()

# We can also close the connection if we are done with it.
# Just be sure any changes have been committed or they will be lost.
conn.close()
```

5 Advanced Analysis with the new Tables

With the new tables generated by Java / Python above (you may need to generate more tables than the example one), it is possible for Reapor to do some more advanced analysis on the IMDb dataset. He asked you to write some sql statements to query:

- The top 3 most common professions among these people and also the average life span of these three professions
- The top 3 most popular (received most votes) genres
- The average time span (endYear - startYear) of the titles for each person