CSE 414 Database Systems

Section 1: Introduction, SQLite, Homework 1

TA: Daseul Lee (dslee@cs)
Introduction

• Your TAs:
  – Daseul Lee: 5th MS/BS student in CS
  – Joseph Xu: PhD student in Robotics, CS

• GoPost for asking questions

• Staff mailing list
  – cse414-staff@cs.washington.edu

• Office hours: TBD
  – Doodle survey for office hour preference
Introduce yourself

• Name
• Major
• Where are you from?
• Why taking this course? Any previous experience with database?
• Interesting facts about you
Section Plan

• During section each week, we will:
  – Highlight the important concepts from lectures
  – Work on some sample problems
  – Demo or brief introduction to homework
  – Any other suggestions?
SQLite: What is it

• SQLite is a C library that implements a relational database management system (DBMS).
  – Simple, lightweight: good for embedded software
  – But does not provide all of the functionalities that other DBMSs do

• Queries are case-insensitive

• We will use sqlite3

References:
  - http://www.sqlite.org/lang.html (SQL Syntax)
  - http://www.sqlite.org/datatype3.html (SQL Data type)
  - http://www.w3schools.com/sql/default.asp (w3school SQL tutorial)
On the Linux machines, or a recent Mac:

- Open a terminal, then run the command:

  sqlite3 database

  where "database" is the name of the database file you want to use.

- WARNING: If you don't specify a database file, sqlite3 won't complain, but your data will be lost!
SQLite: How to Run it (2/2)

• On the Windows machines:
  – Open a Cygwin terminal, then proceed as if you were on Linux.
  – If that doesn't work, you may need to install the "sqlite3" Cygwin package from Cygwin Setup.
  – If *that* doesn't work, try downloading sqlite yourself.

• Download it yourself:
  – Get the "sqlite-shell" binary for your OS from: http://www.sqlite.org/download.html
  – Extract "sqlite3" or "sqlite3.exe" from the archive and run it from a command line.
SQLite: Demo
SQLite: Commands (Not SQL)

- `.help` - lists other . commands
- `.header(s) ON/OFF` - show/hide column headers in query results
- `.mode [mode type]` - how to separate the columns in each row/tuple (for better formatting)
- `.read [file name]` - read and execute SQL code from the given file
- `.separator [string]` - changes the separator for output mode or importing files, i.e. `.separator`
- `.nullvalue [string]` - print the given string in place of NULL values
- `.import [file name] [table name]` - loads the file to the table
  - be careful to set the separator correctly!
- `.show` - see how we have set our parameters
- `.exit` - exit from sqlite3
SQLite: Basic SQL statements

- **CREATE** – creates a new table
  
  
  ex) `CREATE TABLE [table] ( ... );`

- **INSERT INTO** - inserts new data into a table
  
  ex) `INSERT INTO [table] VALUES ([value1], [value2], ...);`

- **SELECT** - extracts data from a table
  
  ex) `SELECT [column(s)] FROM [table_name];`

- **UPDATE** - updates data in a table
  
  ex) `UPDATE FROM [table] SET ... WHERE ...;`

- **DELETE** - deletes data from a table
  
  ex) `DELETE FROM [table] WHERE ...;`
SQLite: SQL keyword, operator, etc

• WHERE clause - filter records
• AND, OR operator - filter records based on more than one condition
• LIKE operator - used in a WHERE clause to search for a specified pattern in a column
• AS - give an alias name to a table or a column
• Relational operators: =, >, >=, <, <=
• Special functions: DATE(...), LENGTH(string), SUBSTR(string, start index, end index),
### SQLite: Example (1/4)

#### Class

<table>
<thead>
<tr>
<th>dept</th>
<th>number</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSE</td>
<td>378</td>
<td>Machine Organization and Assembly Language</td>
</tr>
<tr>
<td>CSE</td>
<td>451</td>
<td>Introduction to Operating Systems</td>
</tr>
<tr>
<td>CSE</td>
<td>461</td>
<td>Introduction to Computer Communication Networks</td>
</tr>
</tbody>
</table>

#### Teaches

<table>
<thead>
<tr>
<th>username</th>
<th>dept</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>zahorjan</td>
<td>cse</td>
<td>378</td>
</tr>
<tr>
<td>tom</td>
<td>cse</td>
<td>451</td>
</tr>
<tr>
<td>tom</td>
<td>cse</td>
<td>461</td>
</tr>
<tr>
<td>zahorjan</td>
<td>cse</td>
<td>451</td>
</tr>
<tr>
<td>zahorjan</td>
<td>cse</td>
<td>461</td>
</tr>
<tr>
<td>djw</td>
<td>cse</td>
<td>461</td>
</tr>
<tr>
<td>levy</td>
<td>cse</td>
<td>451</td>
</tr>
</tbody>
</table>

#### Instructor

<table>
<thead>
<tr>
<th>username</th>
<th>fname</th>
<th>Iname</th>
<th>started_on</th>
</tr>
</thead>
<tbody>
<tr>
<td>zahorjan</td>
<td>John</td>
<td>Zahorjan</td>
<td>1985-01-01</td>
</tr>
<tr>
<td>djw</td>
<td>David</td>
<td>Wetherall</td>
<td>1999-07-01</td>
</tr>
<tr>
<td>tom</td>
<td>Tom</td>
<td>Anderson</td>
<td>1997-10-01</td>
</tr>
<tr>
<td>levy</td>
<td>Hank</td>
<td>Levy</td>
<td>1988-04-01</td>
</tr>
</tbody>
</table>
SQLite: Example (2/4)

• Simple example queries
  – What courses are offered?
  – What's the first name of the instructor with login 'zahorjan'?
  – What 400-level CSE classes are offered?
  – What classes have titles starting with Introduction?
    • If we misspell Introduction as IMtroduction, how can we catch that?
SQLite: Example (3/4)

• Fun with strings
  – Show the class titles and their lengths
  – Truncate all class titles to 12 characters

• Date and time representations
  – Which instructors started before 1990?
  – Which instructors started before now?
  – Which instructors started on or after January 1, 15 years ago?
SQLite: Example (4/4)

• Example queries involving joins
  – Who teaches CSE 451?
  – What courses does John teach?
  – Which courses do both Hank and John teach?
  – Which courses do neither Hank nor David teach?
SQLite: things to watch out for

• SQLite allows a key to be null
• Older versions of sqlite do not enforce FOREIGN KEY constraints.
  – Newer versions are opt-in at both compile time and runtime (with PRAGMA FOREIGN_KEYS = ON)
• SQLite ignores string length maximums or fixed string lengths: N in VARCHAR(N) or CHAR(N)
• SQLite does not have a separate data type for dates, times, or combined date and time.
  – Instead, these are represented as specially formatted strings; dates are represented as yyyy-mm-dd
• And many more as you will discover!
Homework 1

• Create a table in sqlite3 and issue queries
• What to turn in: sql file containing sql commands that answer each question and relevant comments
  – Do not turn in input/output
  – Make sure to add a semicolon at the end of each sql command
  – You can add comments to sql file (for numbering each question)
    /* comment */ or -- comment
Upcoming deadlines

• Webquiz 1: 11pm Sun. April 7
• Homework 1: 11pm Wed. April 10
• Remember to fill out the doodle for office hours as soon as possible
Questions?