CSE 344 Introduction to Data Management

Section 1: Introduction to SQLite
Review: Database and DBMS

• What is a database?
  – Collection of files storing related data
  – Organized collection of data with some data model

• What is a DBMS?
  – Program that allows for efficient management of large database and allows it to persist over time
Review: Database and DBMS

• What kind of features and operations would you want from DBMS?
  – Persistence
  – Efficiency
  – Failure recovery
  – Modularity
  – Reliability
  – Integrity
  – Security
  – ...

SQL (Structural Query Language)

• A special-purpose programming language designed for managing data held in a relational database management system (RDBMS)

• It is a declarative query language:
  – Describes what the program should accomplish, not how to go about accomplishing it

• What can it do?
  – Data insert, query, update and delete, schema creation and modification, and data access control
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

• sqlite3: a standalone program that can run queries and manage an SQLite database

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)
SQLite: How to Run it (1/2)

• Linux or Mac:
  – Open a terminal, then run the command:
    ```
    sqlite3 [database]
    
    where "database" is the name of the database file you want to use.
    
    • Older Mac versions may need to download sqlite
  – 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](http://www.sqlite.org/download.html)
  – Extract "sqlite3" or "sqlite3.exe" from the archive and run it from a command line.

• Make sure your solutions run on the CSE Linux machines
SQLite: . Commands (Not SQL)

- .help - list other . commands
- .header(s) ON/OFF - show/hide column headers in query results
- .mode [mode type] - change 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] - change 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] - load 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 ...;

*Note: Queries are case-insensitive in SQLite*
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), etc
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!
SQLite: Formatting output

• .headers on
• .mode column
• Use .help for more

```sql
sqlite> .headers on
sqlite> .mode column
sqlite> select * from class;
depth   number       title
CSE      378          Machine Organization and Assembly Language
CSE      451          Introduction to Operating Systems
CSE      461          Introduction to Computer Communication Net
```
### 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 (cont.)

• 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?
SQLite: Example (cont.)

• The LIKE Operator
  – What classes have title starting with Introduction?
  – If we misspell Introduction as IMtroduction, let’s catch that by matching any second character, and find the classes that have titles starting with the proper and the misspelling of Introduction
  – Use ‘_’ to match any single character
  – Use ‘%’ to match an arbitrary number of characters
SQLite: Example (cont.)

• 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?
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 files
  - Don’t forget 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 – Due Tuesday 10/4, 11 pm
• Homework 1 – Due Wednesday 10/5, 11 pm