CSE 344 Introduction to Data Management

Section 1: Introduction to SQLite
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)

• On the Linux machines, or Mac:
  – Open a terminal, then run the command:
    \[ \text{sqlite3 [database]} \]
    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: 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*
SQLlite: 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
Foreign Keys

• A column (or columns) whose value is a key of another table (Must be unique!)
  – i.e., a reference to another row in another table
Foreign Keys

• A column (or columns) whose value is a key of another table (Must be unique!)

How would you add a list of products to each company?

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_employees</th>
<th>for_profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canon</td>
<td>Japan</td>
<td>50000</td>
<td>Y</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Japan</td>
<td>30000</td>
<td>Y</td>
</tr>
</tbody>
</table>

Product(pname, price, category, mname, mcountry, FOREIGN KEY (mname, mcountry) REFERENCES Company(cname, country))
Foreign Keys

Company(cname, country, no_employees, for_profit)

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_employees</th>
<th>for_profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canon</td>
<td>Japan</td>
<td>50000</td>
<td>Y</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Japan</td>
<td>30000</td>
<td>Y</td>
</tr>
</tbody>
</table>

Product( pname, price, category, cname, ccountry, FOREIGN KEY (cname,ccountry) REFERENCES Company(cname,country) )

<table>
<thead>
<tr>
<th>pname</th>
<th>price</th>
<th>category</th>
<th>cname</th>
<th>country</th>
</tr>
</thead>
<tbody>
<tr>
<td>SingleTouch</td>
<td>149.99</td>
<td>photography</td>
<td>Canon</td>
<td>Japan</td>
</tr>
<tr>
<td>AC</td>
<td>300</td>
<td>Appliance</td>
<td>Hitachi</td>
<td>Japan</td>
</tr>
</tbody>
</table>
# SQLite: Example

<table>
<thead>
<tr>
<th>Class</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dept</td>
<td>number</td>
<td>title</td>
<td></td>
<td></td>
<td>username</td>
<td>dept</td>
<td>number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSE</td>
<td>378</td>
<td>Machine Organization and Assembly Language</td>
<td></td>
<td></td>
<td>zahorjan</td>
<td>cse</td>
<td>378</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSE</td>
<td>451</td>
<td>Introduction to Operating Systems</td>
<td></td>
<td></td>
<td>tom</td>
<td>cse</td>
<td>451</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSE</td>
<td>461</td>
<td>Introduction to Computer Communication Networks</td>
<td></td>
<td></td>
<td>zahorjan</td>
<td>cse</td>
<td>461</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>zahorjan</td>
<td>cse</td>
<td>461</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>djw</td>
<td>cse</td>
<td>461</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>levy</td>
<td>cse</td>
<td>451</td>
<td></td>
<td></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: 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!