Lecture 2: Data Models & SQL
(Ch. 2.1-2.3)
Announcements

• Office Hours are listed on the calendar
  – one every Monday, Tuesday, Wednesday, and Friday
Data Models

• language / notation for talking about data

• models we will use:
  – relational: data is a collection of tables
  – semi-structured: data is a tree

• other models:
  – key-value pairs: used by NoSQL systems
  – graph data model: used by RDF (semi-structured can also do)
  – object oriented: often layered on relational, J2EE
Relational Model

• Data is a collection of relations / tables:

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
<th>Employees</th>
<th>For_Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>GizmoWorks</td>
<td>USA</td>
<td>20000</td>
<td>True</td>
</tr>
<tr>
<td>Canon</td>
<td>Japan</td>
<td>50000</td>
<td>True</td>
</tr>
<tr>
<td>Hitachi</td>
<td>Japan</td>
<td>30000</td>
<td>True</td>
</tr>
<tr>
<td>HappyCam</td>
<td>Canada</td>
<td>500</td>
<td>False</td>
</tr>
</tbody>
</table>

• mathematically, relation is a set of tuples
  – each tuple appears 0 or 1 times in the table
  – order of the rows is unspecified

rows / tuples / records

columns / attributes / fields
Relational Schema

• Each column has a “domain” (or type)
  – SQL has Java-like types for numbers, strings, etc.
  – domain is a constraint on the data allowed in the table

• Names and types part of the “schema” of the table:

  `Company(Name: string, Country: string,
            Employees: int, For_Profit: boolean)`

• Particular data is an “instance” of that relation
  – data changes over time
  – DBMS usually just stores the current instance
Keys

• Key = subset of columns that uniquely identifies tuple
• Another constraint on the table
  – no two tuples can have the same values for those columns
• Examples:
  – Movie(title, year, length, genre): key is (title, year)
  – what is a good key for Company?
• Part of the schema (book notation is underline):

  Company(Name: string, Country: string,
           Employees: int, For_Profit: boolean)
Keys (cont.)

• Can have multiple keys for a table

• Only one of those keys may be “primary”
  – DBMS often makes searches by primary key fastest
  – other keys are called “secondary”

• “Foreign key” is a column (or columns) whose value is a key of another table
  – i.e., a reference to another row in another table
SQL ("sequel")

- Standard query language for relational data
  - used for databases in many different contexts
  - inspires query languages for non-relational (e.g. SQL++)
- Everything not in quotes (‘…’) is case insensitive
- Provides standard types. Examples:
  - numbers: INT, FLOAT, DECIMAL(p,s)
    - DECIMAL(p,s): Exact numerical, precision p, scale s. Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal
  - strings: CHAR(n), VARCHAR(n)
    - CHAR(n): Fixed-length n
    - VARCHAR(n): Variable length. Maximum length n
• Provides standard types. Examples:
  – BOOLEAN
  – DATE, TIME, TIMESTAMP
    • DATE: Stores year, month, and day values
    • TIME: Stores hour, minute, and second values
    • TIMESTAMP: Stores year, month, day, hour, minute, and second values

• Additional types differ by vendor:
  – SQLite: http://www.sqlite.org/datatype3.html
SQL statements

• create table …
• drop table …
• alter table … add/remove …
• insert into … values …
• delete from … where …
• update … set … where …
CREATE TABLE Company(
    name VARCHAR(20) PRIMARY KEY,
    country VARCHAR(20),
    employees INT,
    for_profit CHAR(1))
drop table ...

DROP TABLE Company;
alter table ... add/remove ...

ALTER TABLE Company
ADD CEO VARCHAR(20);
insert into ... values ...

INSERT INTO Company VALUES
('GizmoWorks', 'USA', 20000, 'y');
delete from ... where ...

DELETE FROM Company
where name = 'GizmoWorks';
update ... set ... where ...

UPDATE Company
SET employees = employees + 120
where name = 'GizmoWorks';
Demo on Sqlite

• E.g., type sqlite3 in Cygwin
• .exit - exit from sqlite3