Introduction to Databases
CSE 414

Lecture 2: Data Models
Announcements

• HW1 and WQ1 released
  – Both due next Tuesday

• Office hours start this week

• Sections tomorrow

• Make sure you sign up on piazza

• Please ask questions!
  – Both online and offline
Staff

- Instructor: Alvin Cheung
  - Office hour on Wednesdays, 1-2pm

From ACM Spring BBQ 15
Using Electronics in Class

In the lectures:
• Opened laptops may disturb neighbors
• Please sit in the back if you take notes on laptop; pads / surfaces are OK
• Please don’t check your email / youtube / fb

In the sections:
• Always bring your laptop (starting Thursday)
Class Overview

- Unit 1: Intro
- Unit 2: Relational Data Models and Query Languages
  - Data models, SQL, Relational Algebra, Datalog
- Unit 3: Non-relational data
- Unit 4: RDMBS internals and query optimization
- Unit 5: Parallel query processing
- Unit 6: DBMS usability, conceptual design
- Unit 7: Transactions
Review

• What is a database?
  – A collection of files storing related data

• What is a DBMS?
  – An application program that allows us to manage efficiently the collection of data files
Data Models

- Recall our example: want to design a database of books:
  - author, title, publisher, pub date, price, etc
  - How should we describe this data?
- **Data model** = mathematical formalism (or conceptual way) for describing the data
Data Models

- **Relational**
  - Data represented as relations

- **Semi-structured (JSON)**
  - Data represented as trees

- **Key-value pairs**
  - Used by NoSQL systems

- **Graph**

- **Object-oriented**
Example: storing FB friends

As a graph

Peter
  /   \\     \
 /       \\   \\
Mary    John
       /   \\
      /     \
     Phil

Or

<table>
<thead>
<tr>
<th>Person1</th>
<th>Person2</th>
<th>is_friend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>John</td>
<td>1</td>
</tr>
<tr>
<td>John</td>
<td>Mary</td>
<td>0</td>
</tr>
<tr>
<td>Mary</td>
<td>Phil</td>
<td>1</td>
</tr>
<tr>
<td>Phil</td>
<td>Peter</td>
<td>1</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

We will learn the tradeoffs of different data models later this quarter
3 Elements of Data Models

• Instance
  – The actual data

• Schema
  – Describe what data is being stored

• Query language
  – How to retrieve and manipulate data
Turing Awards in Data Management

Charles Bachman, 1973
IDS and CODASYL

Ted Codd, 1981
Relational model

Jim Gray, 1998
Transaction processing

Michael Stonebraker, 2014
INGRES and Postgres
Relational Model

- Data is a collection of relations / tables:
  - mathematically, relation is a set of tuples
    - each tuple appears 0 or 1 times in the table
    - order of the rows is unspecified

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_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>
The Relational Data Model

• Degree (arity) of a relation = #attributes
• Each attribute has a type.
  – Examples types:
    • Strings: CHAR(20), VARCHAR(50), TEXT
    • Numbers: INT, SMALLINT, FLOAT
    • MONEY, DATETIME, …
    • Few more that are vendor specific
  – Statically and strictly enforced
Keys

• Key = one (or multiple) attributes that uniquely identify a record
Keys

- Key = one (or multiple) attributes that uniquely identify a record

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_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>
Keys

- Key = one (or multiple) attributes that uniquely identify a record

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>no_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>
Keys

- Key = one (or multiple) attributes that uniquely identify a record

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_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>
Keys

- Key = one (or multiple) attributes that uniquely identify a record

<table>
<thead>
<tr>
<th>cname</th>
<th>country</th>
<th>no_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>

Key

Not a key

Is this a key?

No: future updates to the database may create duplicate no_employees
Multi-attribute Key

Key = fName,lName
(what does this mean?)

<table>
<thead>
<tr>
<th>fName</th>
<th>lName</th>
<th>Income</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Smith</td>
<td>20000</td>
<td>Testing</td>
</tr>
<tr>
<td>Alice</td>
<td>Thompson</td>
<td>50000</td>
<td>Testing</td>
</tr>
<tr>
<td>Bob</td>
<td>Thompson</td>
<td>30000</td>
<td>SW</td>
</tr>
<tr>
<td>Carol</td>
<td>Smith</td>
<td>50000</td>
<td>Testing</td>
</tr>
</tbody>
</table>
### Multiple Keys

We can choose one key and designate it as *primary key*

E.g.: primary key = SSN

<table>
<thead>
<tr>
<th>SSN</th>
<th>fName</th>
<th>lName</th>
<th>Income</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>111-22-3333</td>
<td>Alice</td>
<td>Smith</td>
<td>20000</td>
<td>Testing</td>
</tr>
<tr>
<td>222-33-4444</td>
<td>Alice</td>
<td>Thompson</td>
<td>50000</td>
<td>Testing</td>
</tr>
<tr>
<td>333-44-5555</td>
<td>Bob</td>
<td>Thompson</td>
<td>30000</td>
<td>SW</td>
</tr>
<tr>
<td>444-55-6666</td>
<td>Carol</td>
<td>Smith</td>
<td>50000</td>
<td>Testing</td>
</tr>
</tbody>
</table>
Foreign Key

Company:\text{(cname, country, no\_employees, for\_profit)}
Country:\text{(name, population)}

\begin{tabular}{|l|l|l|l|l|}
\hline
\text{cname} & \text{country} & \text{no\_employees} & \text{for\_profit} \\
\hline
Canon & Japan & 50000 & Y \\
Hitachi & Japan & 30000 & Y \\
\hline
\end{tabular}

\begin{tabular}{|l|l|}
\hline
\text{name} & \text{population} \\
\hline
USA & 320M \\
Japan & 127M \\
\hline
\end{tabular}
Keys: Summary

• Key = columns that uniquely identify tuple
  – Usually we underline
  – A relation can have many keys, but only one can be chosen as primary key

• Foreign key:
  – Attribute(s) whose value is a key of a record in some other relation
  – Foreign keys are sometimes called semantic pointer
Query Language

• SQL
  – **Structured Query Language**
  – Developed by IBM in the 70s
  – Most widely used language to query relational data

• Other relational query languages
  – Datalog, relational algebra
Our First DBMS

• SQL Lite
• Will switch to SQL Server later in the quarter
Demo 1
Discussion

• Tables are NOT ordered
  – they are sets or multisets (bags)

• Tables are FLAT
  – No nested attributes

• Tables DO NOT prescribe how they are implemented / stored on disk
  – This is called physical data independence