Introduction to Data Management CSE 344

Lecture 2: Data Models

Announcements

- WQ1 and HW1 are out
  – Use your **CSE** ids to access the HW docs

- Use Piazza to post questions

- OHs are up on website

Ground Rules

- This is a huge room
  - Please sit in the front
  - I promise I don't bite!
- Let us know if you have suggestions for the class
  - Don't wait till course evals!
- Slides will be available before lecture
  - We can bring hardcopies if needed
- Some slides have details missing
  - Please pay attention in class and take notes!

Using Electronics in Class

- Opened laptops create disturbances to your neighbors
- Please sit in the back if you use your laptop to take notes
- OK if you use surfaces

- And please don't check your email / sms / youtube / fb / etc during class
  - If people are doing this we will have to ban all laptops ☹

Class Overview

- Relational Data Model
  - SQL, Relational Algebra, Relational Calculus, datalog
  - Query processing and optimization
- Semistructured Data Model
  - JSon (NoSQL)
- Conceptual design
  - E/R diagrams, Views, and Database normalization
- Transactions and their implementations
- Parallel databases
  - MapReduce, and Spark

Today

- Data models
- Relational data model
- SQL

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

- Suppose we have book data: author, title, publisher, pub date, price, etc
  - How should we organize such data in files?

Data model: a general, conceptual way of structuring 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
- We will study the first three in 344

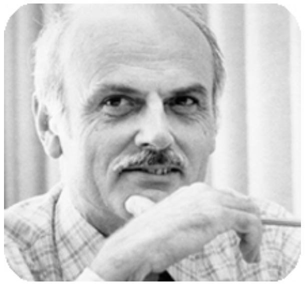
3 Elements of Data Models

- Instance
  - The actual data
- Schema
  - Describe what data is being stored
- Query language
  - How data can be retrieved and manipulated

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

The Relational Data Model

- Instance
  - Organized as “table” or “relation”
  - Consists of
    - “column” aka “attribute” aka “field”
    - “row” aka “tuple” aka “record”
- Schema
  - “table name” aka “relation name”
  - “column name” aka “attribute name”
  - Each attribute has a “type” aka “domain” aka “data type”

The Relational Data Model

- “degree” or “arity” of a relation
  - Number of attributes
- Example types:
  - Strings: CHAR(20), VARCHAR(50), TEXT
  - Numbers: INT, SMALLINT, FLOAT
  - MONEY, DATETIME, …
  - Usually vendor specific
  - Statically and strictly enforced

Keys

- An attribute that uniquely identifies a record
  - Example?
- A key can consist of multiple attributes
  - What does that mean?

Keys

- A relation can have many keys
  - But only one of them can be chosen to be the *primary key*
- Foreign key:
  - An attribute(s) that is a key for other relations

Relation Model: Example

- Instance

cname	country	no_employees	for_profit
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

- Schema

```
Company(cname, country, no_employees, for_profit)

Company(cname: varchar(30), country: char(20),
          no_employees: int, for_profit: char(1))
```

Relational Model: Example

```
Company(cname, country, no_employees, for_profit)

Country(name, population)
```

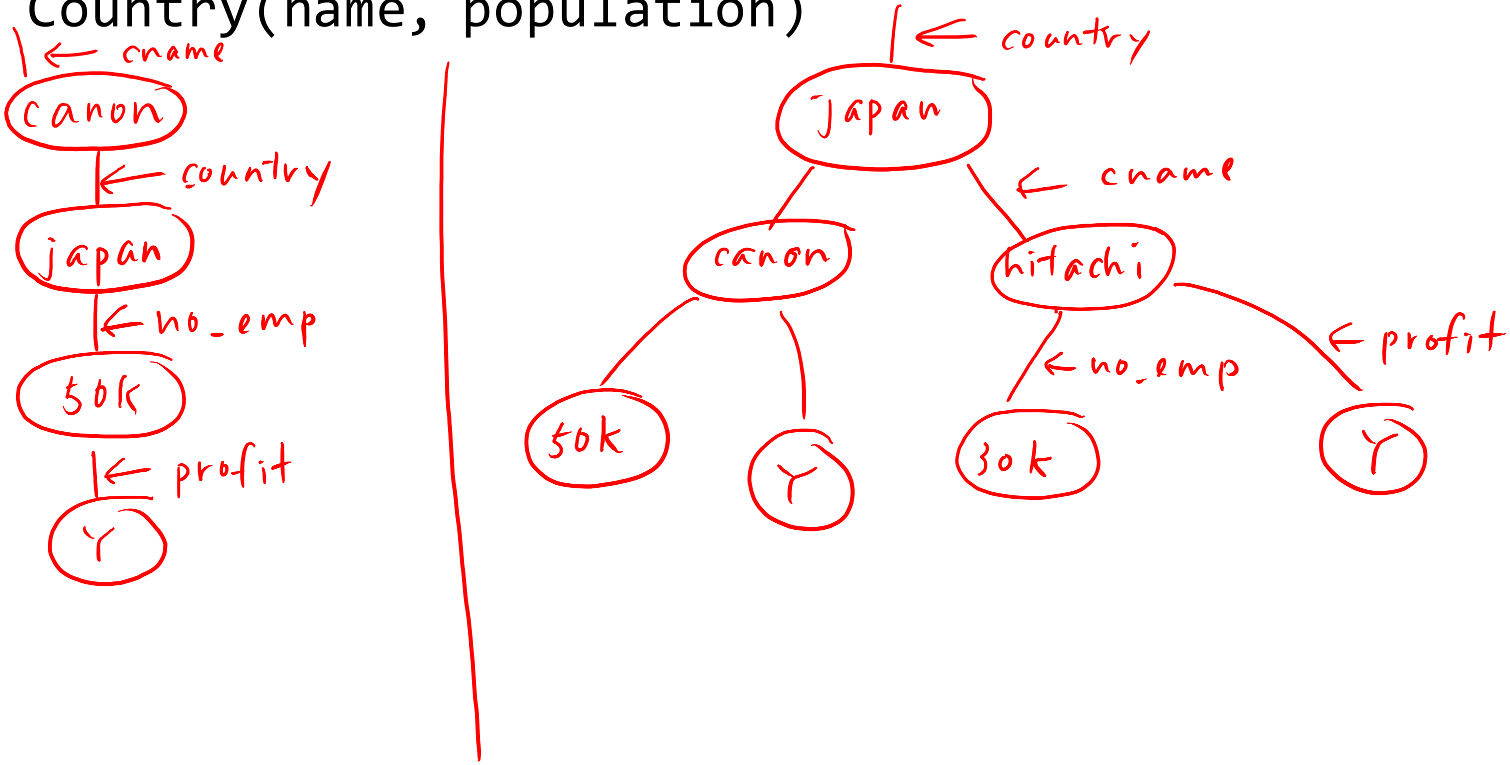
cname	country	no_employees	for_profit
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

name	population
USA	320M
Japan	127M

Aside: Semi-Structured Model: Example

```
Company(cname, country, no_employees, for_profit)

Country(name, population)
```



Query Language

- SQL
  - **S**tructured **Q**uery **L**anguage
  - Developed by IBM in the 70s
  - Most widely used language to query relational data
- We will see other languages for the relational model later on
  - Relational algebra, relational calculus, etc.

Our First DBMS

- SQL Lite
- Will switch to SQL Server later in the quarter

Demo

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**

Table Implementation

- How would you implement this?

cname	country	no_employees	for_profit
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

- What happens when you alter a table?

Physical data independence

The logical definition of the data remains unchanged, even when we make changes to the actual implementation

Adding Attributes

cname	country	no_employees	for_profit
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

- Let’s add a list of product that each company produces
  - How? Recall that tables are flat!

Adding Attributes

cname	country	no_employees	for_profit
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

```
Product(pname, price, category, manufacturer)
```

pname	price	category	manufacturer
SingleTouch	149.99	photography	Canon
AC	300	Appliance	Hitachi

Demo