Introduction to Databases
CSE 414

Lecture 2: Data Models

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

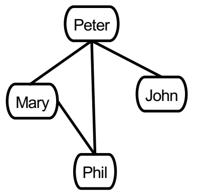
Unit 2

- Semi-structured (JSon)
  - Data represented as trees
- Key-value pairs
  - Used by NoSQL systems

Unit 3

- Graph
- Object-oriented

Example: storing FB friends



Or

Person1	Person2	is_friend
Peter	John	1
John	Mary	0
Mary	Phil	1
Phil	Peter	1
…	…	…

As a graph

As a relation

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

columns / attributes / fields

- Data is a collection of relations / tables:

rows / tuples / records

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False

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

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

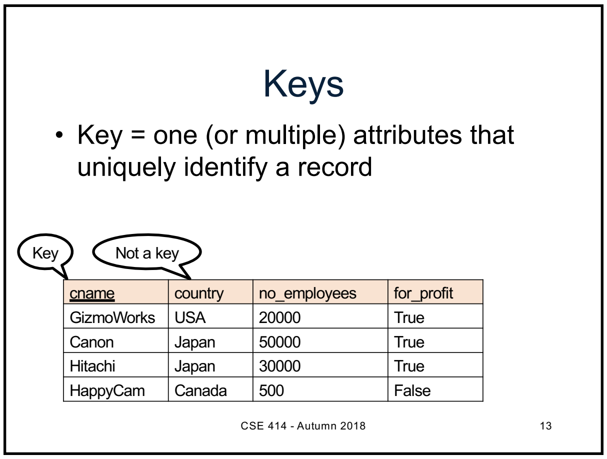
- Key = one (or multiple) attributes that uniquely identify a record

Keys

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

Key

<u>cname</u>	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False



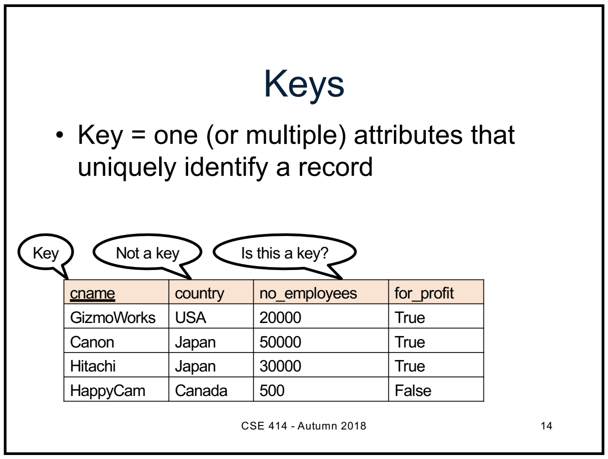
Keys

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

Key

Not a key

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False



Keys

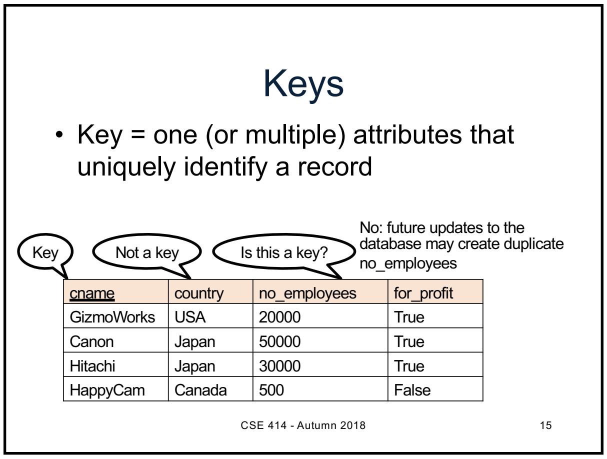
- Key = one (or multiple) attributes that uniquely identify a record

Key

Not a key

Is this a key?

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False



Keys

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

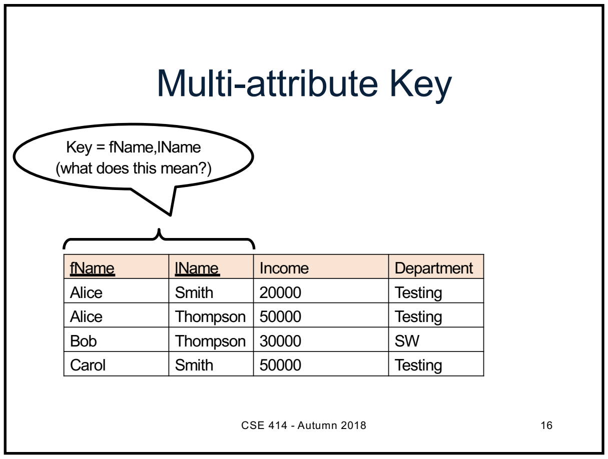
Key

Not a key

Is this a key?

No: future updates to the database may create duplicate no_employees

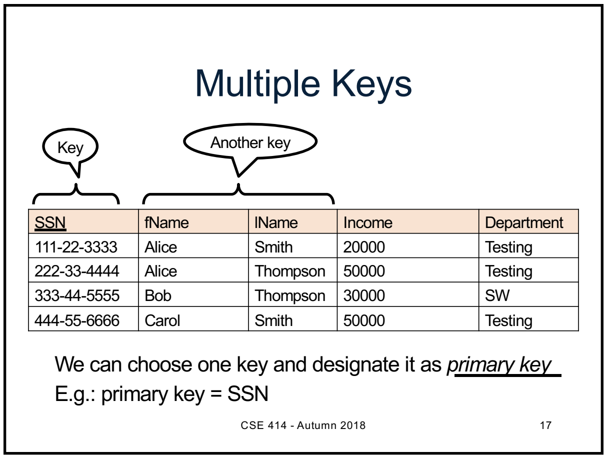
cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False



Multi-attribute Key

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

fName	lName	Income	Department
Alice	Smith	20000	Testing
Alice	Thompson	50000	Testing
Bob	Thompson	30000	SW
Carol	Smith	50000	Testing



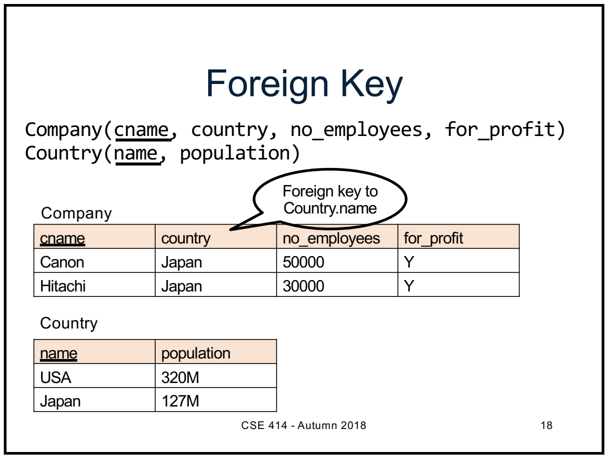
Multiple Keys

Key

Another key

SSN	fName	lName	Income	Department
111-22-3333	Alice	Smith	20000	Testing
222-33-4444	Alice	Thompson	50000	Testing
333-44-5555	Bob	Thompson	30000	SW
444-55-6666	Carol	Smith	50000	Testing

We can choose one key and designate it as *primary key*
E.g.: primary key = SSN



Foreign Key

```
Company(cname, country, no_employees, for_profit)
Country(name, population)
```

Foreign key to Country.name

Company

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

Country

name	population
USA	320M
Japan	127M

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
  - **S**tructured **Q**uery **L**anguage
  - 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**

Table Implementation

- How would you implement this?

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False

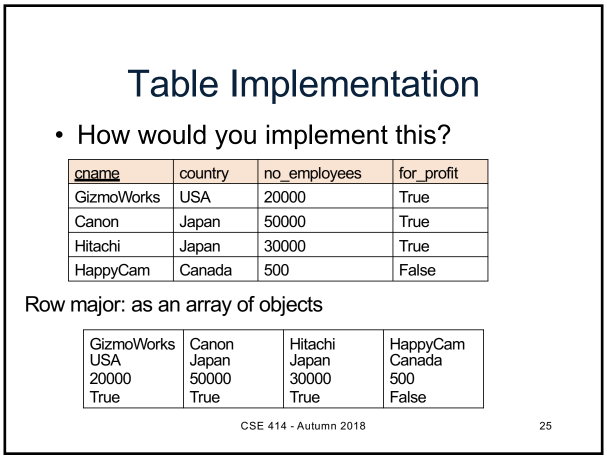


Table Implementation

- How would you implement this?

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False

Row major: as an array of objects

GizmoWorks USA 20000 True	Canon Japan 50000 True	Hitachi Japan 30000 True	HappyCam Canada 500 False

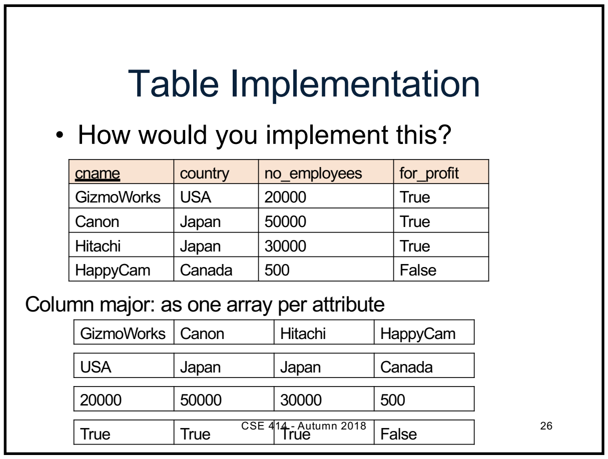


Table Implementation

- How would you implement this?

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False

Column major: as one array per attribute

GizmoWorks	Canon	Hitachi	HappyCam

USA	Japan	Japan	Canada

20000	50000	30000	500

True	True	True	False

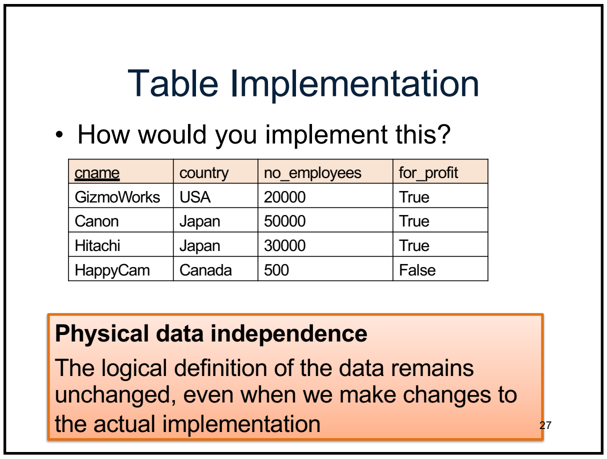


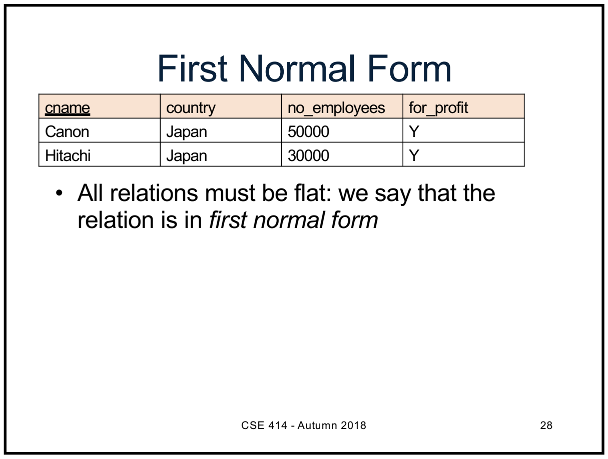
Table Implementation

- How would you implement this?

cname	country	no_employees	for_profit
GizmoWorks	USA	20000	True
Canon	Japan	50000	True
Hitachi	Japan	30000	True
HappyCam	Canada	500	False

Physical data independence

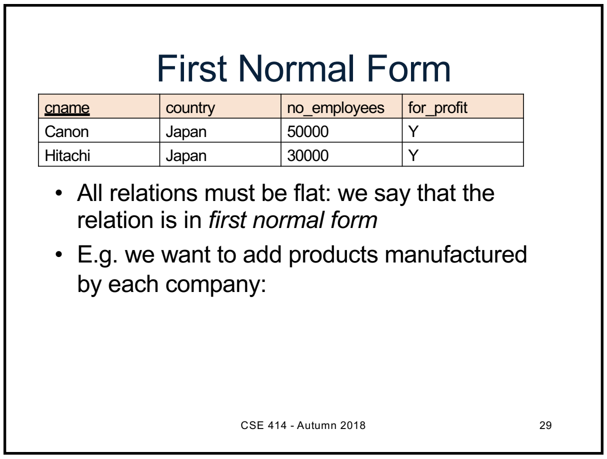
The logical definition of the data remains unchanged, even when we make changes to the actual implementation



First Normal Form

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

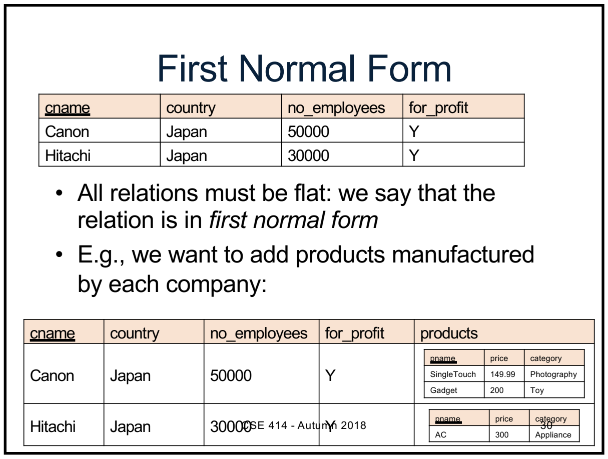
- All relations must be flat: we say that the relation is in *first normal form*



First Normal Form

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

- All relations must be flat: we say that the relation is in *first normal form*
- E.g. we want to add products manufactured by each company:



First Normal Form

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

- All relations must be flat: we say that the relation is in *first normal form*
- E.g., we want to add products manufactured by each company:

<table>
<tr><th>cname</th><th>country</th><th>no_employees</th><th>for_profit</th><th>products</th></tr>
<tr><td>Canon</td><td>Japan</td><td>50000</td><td>Y</td><td><table><tr><th>pname</th><th>price</th><th>category</th></tr><tr><td>SingleTouch</td><td>149.99</td><td>Photography</td></tr><tr><td>Gadget</td><td>200</td><td>Toy</td></tr></table></td></tr>
<tr><td>Hitachi</td><td>Japan</td><td>30000</td><td>Y</td><td><table><tr><th>pname</th><th>price</th><th>category</th></tr><tr><td>AC</td><td>300</td><td>Appliance</td></tr></table></td></tr>
</table>

First Normal Form

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

- All relations must be flat: we say that the relation is in *first normal form*
- E.g., we want to add products manufactured by each company:

Non-1NF!

<table>
<tr><th>cname</th><th>country</th><th>no_employees</th><th>for_profit</th><th>products</th></tr>
<tr><td>Canon</td><td>Japan</td><td>50000</td><td>Y</td><td><table><tr><th>pname</th><th>price</th><th>category</th></tr><tr><td>SingleTouch</td><td>149.99</td><td>Photography</td></tr><tr><td>Gadget</td><td>200</td><td>Toy</td></tr></table></td></tr>
<tr><td>Hitachi</td><td>Japan</td><td>30000</td><td>Y</td><td><table><tr><th>pname</th><th>price</th><th>category</th></tr><tr><td>AC</td><td>300</td><td>Appliance</td></tr></table></td></tr>
</table>

First Normal Form

Now it's in 1NF

Company

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

Products

pname	price	category	manufacturer
SingleTouch	149.99	Photography	Canon
AC	300	Appliance	Hitachi
Gadget	200	Toy	Canon

Demo 1 (cont'd)

Data Models: Summary

- Schema + Instance + Query language
- Relational model:
  - – Database = collection of tables
  - – Each table is flat: "first normal form"
  - – Key: may consists of multiple attributes
  - – Foreign key: "semantic pointer"
  - – Physical data independence