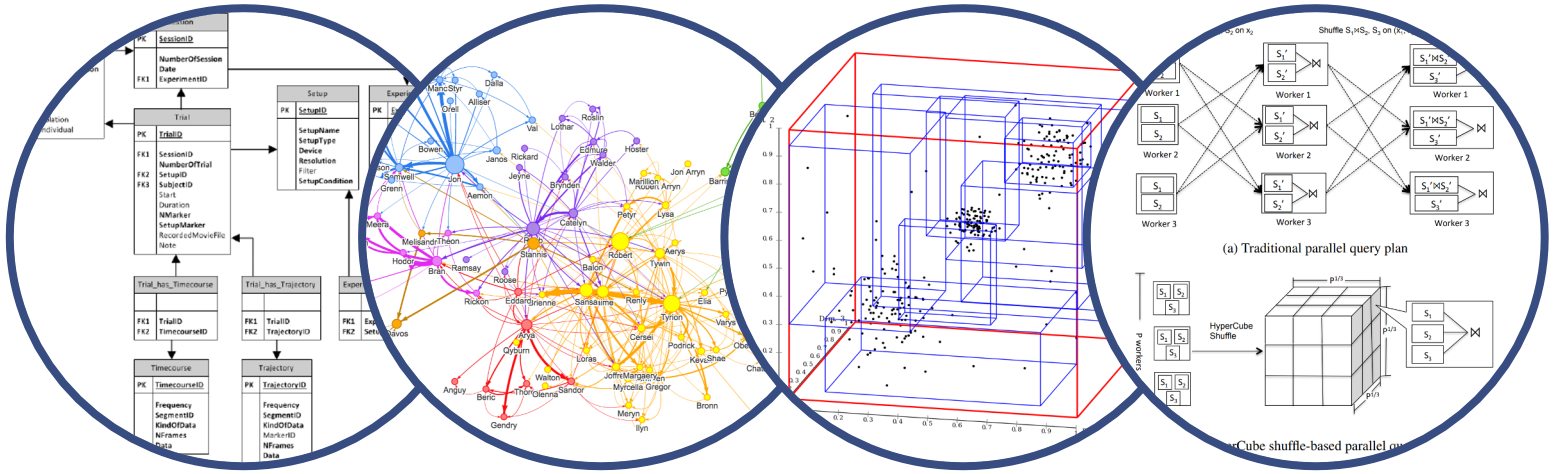

Introduction to Data Management

SQL Basics and Joins

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Review

- Relational data model
  - Schema+instance+query language
- Query language: SQL
  - Create tables
  - Retrieve records from tables
  - Declare keys and foreign keys

Keys

Key

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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

A **Key** is one or more attributes that, in aggregate, uniquely identify a row.

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
Canon	Japan	50000	Y
Hitachi	Japan	30000	Y

Country

cname	population
USA	320M
Japan	127M

Foreign Key

Foreign key to Country.cname

Company

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

Country

cname	population
USA	320M
Japan	127M

Foreign Key

Foreign key to Country.cname

Company

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

Country

cname	population
USA	320M
Japan	127M

Foreign Key

A **Foreign Key** is one or more attributes that, in aggregate, uniquely identify a row in *another table*.

Review

- Tables are NOT ordered
  - they are sets or multisets (bags)

Review

- Tables are NOT ordered
  - they are sets or multisets (bags)

- Tables are FLAT
  - No nested attributes
  - We say that the table is in First Normal Form

Review

- Tables are NOT ordered
  - they are sets or multisets (bags)
- Tables are FLAT
  - No nested attributes
  - We say that the table is in First Normal Form
- Tables DO NOT prescribe how they are implemented / stored on disk

Review

- Tables are NOT ordered
  - they are sets or multisets (bags)
- Tables are FLAT
  - No nested attributes
  - We say that the table is in First Normal Form
- Tables DO NOT prescribe how they are implemented / stored on disk
- Physical data independence

SQL

SQL

- **S**tructured **Q**uery **L**anguage
- Most widely used language to query relational data
- One of the many languages for querying relational data

- A **declarative** programming language

Note

SQL is a huge language. We cover in class the *important* concepts. You may have to google for some simple command that we don't cover in class.

SQL

Simplest variant: three clauses:

- SELECT: which attributes we want
- FROM: which table(s) we query
- WHERE: predicate saying which rows we want

URLs

ID	URL	Title	NumVisits
5	cs.washington.edu	Welcome to Paul G.	13
6	google.com	Google	2
34	washington.zoom.us	Video Conf...	4
37	canvas.uw.edu	Dashboard	6

SQL

URLs

ID	URL	Title	NumVisits
5	cs.washington.edu	Welcome to Paul G.	13
6	google.com	Google	2
34	washington.zoom.us	Video Conf...	4
37	canvas.uw.edu	Dashboard	6

```
SELECT URL, Title
  FROM URLS
 WHERE NumVisits > 5;
```

SQL

URLs

ID	URL	Title	NumVisits
5	cs.washington.edu	Welcome to Paul G.	13
6	google.com	Google	2
34	washington.zoom.us	Video Conf...	4
37	canvas.uw.edu	Dashboard	6

```
SELECT URL, Title
  FROM URLS
 WHERE NumVisits > 5;
```

Answer:

URL	Title
cs.washington.edu	Welcome to Paul G.
canvas.uw.edu	Dashboard

Demo

Demo will show the following:

- CREATE TABLE
- INSERT, DELETE
- NULL
- More SELECT:
  - SELECT *
  - SELECT DISTINCT
  - Discuss the notions of filter(selection), projection
  - LIKE
  - ORDER BY

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

cname is a key in Company
pname is a key in Product
manufacturer is a foreign to Company.cname

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

```
SELECT pname
  FROM Product, Company
```

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

```
SELECT pname
  FROM Product, Company
 WHERE country = 'Japan'
   AND price < 150
```

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

```
SELECT pname
  FROM Product, Company
 WHERE country = 'Japan'
   AND price < 150
   AND manufacturer = cname
```

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

```
SELECT pname
  FROM Product, Company
 WHERE country = 'Japan'
   AND price < 150
```

What does this return?

Joins in SQL

Product

pname	price	category	manufacturer
MultiTouch	199.99	gadget	Canon
SingleTouch	49.99	photography	Canon
Gizom	50	gadget	GizmoWorks
SuperGizmo	250.00	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Retrieve all Japanese products that cost < $150

```
SELECT pname
  FROM Product, Company
 WHERE country = 'Japan'
   AND price < 150
```

What does this return?

ALL products < $150

Demo

Demo will show

- foreign keys
- Joins
- cartesian producct

Discussion

To join two tables:

- Write those table in the FROM clause
- Write *explicitly* the join condition
- Records that do not join with anything in the other table are not included