Introduction to Database Systems

Lecture 3: SQL Joins

(including most of Ch. 6.1-6.2)

SQL Joins

- Can use data from multiple tables:

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

- This is a selection and projection of the "join" of the Product and Company relations.

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks

('Canon', 'Japan', 'SingleTouch', 149.99, 'Canon')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks

('Canon', 'Japan', 'Gizmo', 19.99, 'GizmoWorks')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks

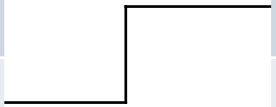
('Canon', 'Japan', 'PowerGizmo', 29.99, 'GizmoWorks')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks



('GizmoWorks', 'USA', 'SingleTouch', 149.99, 'Canon')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks

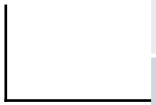
('GizmoWorks', 'USA', 'Gizmo', 19.99, 'GizmoWorks')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks



('GizmoWorks', 'USA', 'PowerGizmo', 29.99, 'GizmoWorks')

Interpreting Joins

- A JOIN B produces one row for every pair of rows
  - one row from A and one row from B

Cname	Country
Canon	Japan
GizmoWorks	USA

JOIN

Pname	Price	Manufacturer
SingleTouch	149.99	Canon
Gizmo	19.99	GizmoWorks
PowerGizmo	29.99	GizmoWorks

- This join produces 6 different rows
  - in general, # rows in join is (# rows in A) * (# rows in B)
  - number of rows often **much smaller** after selection…
  - DBMS will do everything in its power to not compute A JOIN B

Interpreting Joins (2)

- Can think of a join in terms of code:

```
for every row C in Company {
  for every row P in Product {
    if (P.manufacturer = C.cname and
        C.country = 'Japan' and
        P.price < 150.00)
      output (C.cname, C.country,
          P.pname, P.price, P.category,
          P.manufacturer);
  }
}
```

Types of Joins

- We usually think of the selection as part of the join
  - e.g., manufacturer = cname and country = ‘Japan’ and …
  - called the “join predicate”
- Join without a predicate is cross product / cross join
- Special names depending on predicate
  - natural join if “=“ between pairs of columns with same name
  - with well chosen col names, many joins become natural
- These are “inner” joins. We will discuss outer later…

(Inner) Joins

```
SELECT a1, a2, …, an
FROM   R1, R2, …, Rm
WHERE  Cond
```

(Nested loop semantics)

```
for t1 in R1:
  for t2 in R2:
    ...
      for tm in Rm:
        if Cond(t1.a1, t1.a2, …):
          output(t1.a1, t1.a2, …, tm.an)
```

(Inner) joins

Company(cname, country)
Product(pname, price, category, manufacturer)
– manufacturer is foreign key

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

pname	category	manufacturer	cname	country
Gizmo	gadget	GizmoWorks	GizmoWorks	USA

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because country != 'USA'
(also cname != manufacturer)

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because country != 'USA'

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because category != 'gadget' (and …)

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because category != 'gadget'

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because category != 'gadget'

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

Not output because category != 'gadget' (with any Company)

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
Camera	Photo	Hitachi
OneClick	Photo	Hitachi

restrict to category = 'gadget'

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product (where category = 'gadget')

pname	category	manufacturer
Gizmo	gadget	GizmoWorks

Company

cname	country
GizmoWorks	USA
Canon	Japan
Hitachi	Japan

restrict to country = 'USA'

(Inner) joins

```
SELECT DISTINCT cname
FROM    Product, Company
WHERE   country = 'USA' AND category = 'gadget' AND
        manufacturer = cname
```

Product (where category = 'gadget')

pname	category	manufacturer
Gizmo	gadget	GizmoWorks

Company (where country = 'USA')

cname	country
GizmoWorks	USA

Now only one combination to consider

(Query optimizers do this too.)

(Inner) joins

```
SELECT DISTINCT cname
FROM     Product, Company
WHERE    country = 'USA' AND category = 'gadget' AND
         manufacturer = cname
```

Alternative syntax:

```
SELECT DISTINCT cname
FROM     Product JOIN Company ON
     country = 'USA' AND category = 'gadget' AND
     manufacturer = cname
```

Emphasizes that the predicate is part of the join.

Self-Joins and Tuple Variables

- **Ex**: find companies that manufacture both products in the ‘gadgets’ category and in the ‘photo’ category
- Just joining Company with Product is insufficient: need to join Company with Product with Product

```
FROM Company, Product, Product
```

- When a relation occurs twice in the FROM clause we call it a *self-join*; in that case every column name in Product is ambiguous (why?)
  - are you referring to the tuple in the 2nd or 3rd loop?

Name Conflicts

we used cname / pname to avoid this problem

- When a name is ambiguous, qualify it:

```
WHERE Company.name = Product.name AND …
```

- For self-join, we need to distinguish tables:

```
FROM Product x, Product y, Company
```

- These new names are called “tuple variables”
  - can think of as name for the variable of each loop
  - can also write “`Company AS C`” etc.
  - can make SQL query shorter: `C.name` vs. `Company.name`

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

cname	country
GizmoWorks	USA
Hitachi	Japan

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

X

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

cname	country
GizmoWorks	USA
Hitachi	Japan

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

x

y

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

cname	country
GizmoWorks	USA
Hitachi	Japan

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

x

y

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

z

cname	country
GizmoWorks	USA
Hitachi	Japan

restrict to country = 'USA'

Not output because y.category != 'photo'

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

x

y

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

z

cname	country
GizmoWorks	USA
Hitachi	Japan

Not output because y.manufacturer != cname

Self-joins

```
SELECT DISTINCT z.cname
FROM    Product x, Product y, Company z
WHERE   z.country = 'USA'
    AND   x.category = 'gadget'
    AND   y.category = 'photo'
    AND   x.manufacturer = cname
    AND   y.manufacturer = cname;
```

Product

x

y

pname	category	manufacturer
Gizmo	gadget	GizmoWorks
SingleTouch	photo	Hitachi
MultiTouch	photo	GizmoWorks

Company

z

cname	country
GizmoWorks	USA
Hitachi	Japan

x.pname	x.category	x.manufacturer	y.pname	y.category	y.manufacturer	z.cname	z.country
Gizmo	gadget	GizmoWorks	MultiTouch	Photo	GizmoWorks	GizmoWorks	USA

Outer joins

Product(name, category)
Purchase(prodName, store) -- prodName is foreign key

```
SELECT Product.name, …, Purchase.store
FROM   Product, Purchase
WHERE  Product.name = Purchase.prodName
```

Or equivalently:

```
SELECT Product.name, …, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

But some Products may not be not listed. Why?

Outer joins

Product(name, category)
Purchase(prodName, store) -- prodName is foreign key

If we want to include products that never sold,
then we need an “outer join”:

```
SELECT Product.name, …, Purchase.store
FROM     Product LEFT OUTER JOIN Purchase ON
                    Product.name = Purchase.prodName
```

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz
Camera	Ritz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product LEFT OUTER JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

```
SELECT Product.name, Purchase.store
FROM   Product LEFT OUTER JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
OneClick	NULL

```
SELECT Product.name, Purchase.store
FROM   Product RIGHT OUTER JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
Phone	Foo

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
NULL	Foo

```
SELECT Product.name, Purchase.store
FROM   Product FULL OUTER JOIN Purchase ON
       Product.name = Purchase.prodName
```

Product

Name	Category
Gizmo	gadget
Camera	Photo
OneClick	Photo

Purchase

ProdName	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
Phone	Foo

Name	Store
Gizmo	Wiz
Camera	Ritz
Camera	Wiz
OneClick	NULL
NULL	Foo

Outer Joins

- Left outer join:
  - Include the left tuple even if there's no match
- Right outer join:
  - Include the right tuple even if there's no match
- Full outer join:
  - Include both left and right tuples even if there's no match

- (Also something called a UNION JOIN, though it's rarely used.)
- (Actually, all of these are used much more rarely than inner joins.)

Join Examples

- See lec03-sql-joins.sql…