

SQL

Joins

Product (pname, price, category, manufacturer)

Company (cname, stockPrice, country)

Find all products under \$200 manufactured in Japan;
return their names and prices.

```
SELECT PName, Price  
FROM Product, Company  
WHERE Manufacturer=CName AND Country='Japan'  
AND Price <= 200
```



Join
between Product
and Company

Joins

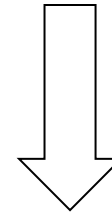
Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

Cname	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT PName, Price
FROM Product, Company
WHERE Manufacturer=CName AND Country='Japan'
AND Price <= 200
```



PName	Price
SingleTouch	\$149.99

More Joins

Product (pname, price, category, manufacturer)

Company (cname, stockPrice, country)

Find all Chinese companies that manufacture products both in the 'electronic' and 'toy' categories

```
SELECT  cname
```

```
FROM
```

```
WHERE
```

A Subtlety about Joins

Product (pname, price, category, manufacturer)

Company (cname, stockPrice, country)

Find all countries that manufacture some product in the 'Gadgets' category.

```
SELECT Country
FROM Product, Company
WHERE Manufacturer=CName AND Category='Gadgets'
```

Unexpected duplicates

A Subtlety about Joins

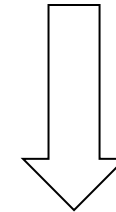
Product

<u>Name</u>	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

Company

<u>Cname</u>	StockPrice	Country
GizmoWorks	25	USA
Canon	65	Japan
Hitachi	15	Japan

```
SELECT Country
FROM Product, Company
WHERE Manufacturer=CName AND Category='Gadgets'
```



Country
??
??

What is
the problem ?
What's the
solution ?

Tuple Variables

Person(pname, address, worksfor)

Company(cname, address)

```
SELECT DISTINCT pname, address
FROM      Person, Company
WHERE     worksfor = cname
```

Which
address ?



```
SELECT DISTINCT Person.pname, Company.address
FROM      Person, Company
WHERE     Person.worksfor = Company.cname
```



```
SELECT DISTINCT x.pname, y.address
FROM      Person AS x, Company AS y
WHERE     x.worksfor = y.cname
```

Meaning (Semantics) of SQL Queries

```
SELECT  $a_1, a_2, \dots, a_k$   
FROM  $R_1$  AS  $x_1, R_2$  AS  $x_2, \dots, R_n$  AS  $x_n$   
WHERE Conditions
```

```
Answer = { }  
for  $x_1$  in  $R_1$  do  
    for  $x_2$  in  $R_2$  do  
        .....  
        for  $x_n$  in  $R_n$  do  
            if Conditions  
                then Answer = Answer  $\cup \{(a_1, \dots, a_k)\}$   
return Answer
```


An Unintuitive Query

```
SELECT DISTINCT R.A  
FROM R, S, T  
WHERE R.A=S.A OR R.A=T.A
```

What does it compute ?

Computes $R \cap (S \cup T)$

But what if $S = \phi$?

Subqueries Returning Relations

Company(name, city)

Product(pname, maker)

Purchase(id, product, buyer)

Return cities where one can find companies that manufacture products bought by Joe Blow

```
SELECT Company.city
FROM Company
WHERE Company.name IN
    (SELECT Product.maker
     FROM Purchase , Product
     WHERE Product.pname=Purchase.product
     AND Purchase .buyer = 'Joe Blow');
```

Subqueries Returning Relations

Is it equivalent to this ?

```
SELECT Company.city  
FROM    Company, Product, Purchase  
WHERE   Company.name= Product.maker  
        AND Product.pname = Purchase.product  
        AND Purchase.buyer = 'Joe Blow'
```

Beware of duplicates !

Removing Duplicates

```
SELECT DISTINCT Company.city
FROM    Company
WHERE   Company.name IN
        (SELECT Product.maker
         FROM   Purchase , Product
         WHERE  Product.pname=Purchase.product
              AND Purchase .buyer = 'Joe Blow');
```

```
SELECT DISTINCT Company.city
FROM    Company, Product, Purchase
WHERE   Company.name= Product.maker
        AND Product.pname = Purchase.product
        AND Purchase.buyer = 'Joe Blow'
```

Now
they are
equivalent