Database Systems
CSE 303

Lecture 04

2016

Queries Over Multiple Table

Problems with single table

Add company info like country, stock price

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

Problems with single table

Product

PName	Price	Category	Manufacturer	Country	StockPrice
Gizmo	$19.99	Gadgets	GizmoWorks	USA	25
Powergizmo	$29.99	Gadgets	GizmoWorks	USA	25
SingleTouch	$149.99	Photography	Canon	Japan	65
MultiTouch	$203.99	Household	Hitachi	Japan	15

Solution to the problem: Split

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

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

Solution to the problem: Split

Foreign key

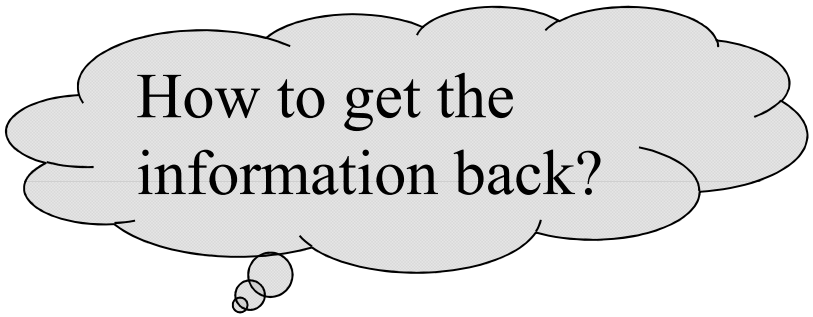
Primary Key

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

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


How to get the
information back?

Cross Product of two tables

Product A × B

Part of the SELECT statement - list more than one table after keyword FROM.

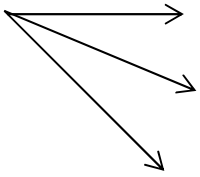
```
SELECT A, R.B, S.B, C
FROM R, S;
```

R

A	B
1	x
2	y

S

B	C
x	5
y	6
z	7

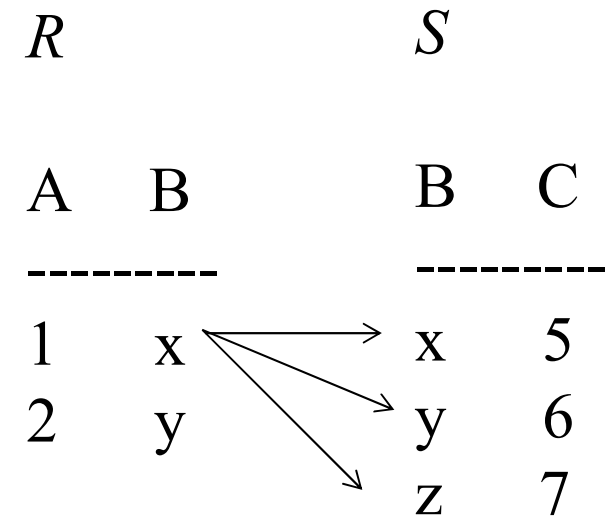


R × S

A	R.B	S.B	C
1	x	x	5
1	x	y	6
1	x	z	7
2	y	x	5
2	y	y	6
2	y	z	7

Join

```
SELECT *
FROM R, S
WHERE R.B=S.B;
```



$R \times S$

A	R.B	S.B	C
1	x	x	5
1	x	y	6
1	x	z	7
2	y	x	5
2	y	y	6
2	y	z	7

Join

```
SELECT *
FROM R, S
WHERE R.B=S.B;
```

R

A	B
1	x
2	y

S

B	C
x	5
y	6
z	7

A	R.B	S.B	C
1	x	x	5
2	y	y	6

Join

R natural join S

```
SELECT A, R.B, C
FROM R, S
WHERE R.B=S.B;
```

SELECT *
FROM R NATURAL JOIN S

R

A	B
1	x
2	y

S

B	C
x	5
y	6
z	7

R natural join S

A	R.B	C
1	x	5
2	y	6

Meaning (Semantics) of SQL Queries Using Cross Product

SELECT $a_1, a_2, \ldots, a_k$
FROM $R_1, R_2, \ldots, R_n$
WHERE Conditions

```
Answer = {}
for x1 in R1 do
    for x2 in R2 do
        …..
            for xn in Rn do
                if Conditions
                    then Answer = Answer ∪ {(a1,…,ak)}
return Answer
```

Joins

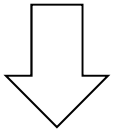
Product

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  *
FROM    Product, Company
WHERE   manufacturer=cname
```



Product join Campany

PName	Price	Category	Manufacturer	Cname	StockPrice	Country
Gizmo	$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	$29.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
SingleTouch	$149.99	Photography	Canon	Canon	65	Japan
MultiTouch	$203.99	Household	Hitachi	Hitachi	15	Japan

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

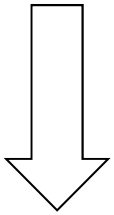
Product

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 electronic products or toy.

```
SELECT   cname
FROM     Product, Company
WHERE    manufacturer = cname
AND      country = 'China'
AND      (category = 'electronic' OR category = 'toy')
```

A Subtlety about Joins

Product (pname, price, category, manufacturer)
Company (cname, stockPrice, country)

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

A Subtlety about Joins

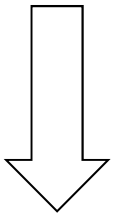
Product

Name	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   country
FROM     Product, Company
WHERE    manufacturer=cname
AND      category='Gadgets'
```



Country
??
??

What is the problem ?

A Subtlety about Joins

Product

Name	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   country
FROM     Product, Company
WHERE    manufacturer=cname
AND      category='Gadgets'
```

Country
USA
USA

Duplicates ! What's the solution ?

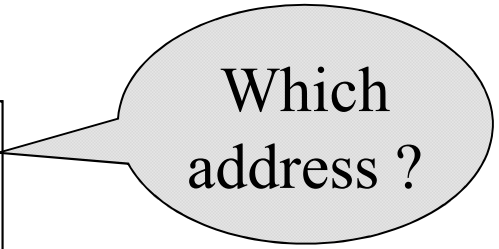
Use 'distinct' keyword

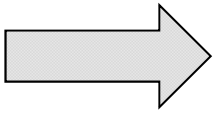
Tuple Variables

Person(pname, address, worksfor)

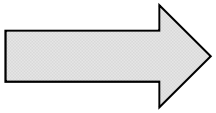
Company(cname, address)

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





```
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
```

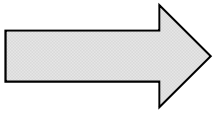
Tuple Variables

Person(pname, address, worksfor)

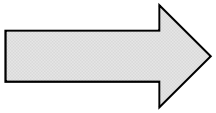
Company(cname, address)

```
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
```

No 'AS' in Oracle

Union

Union A ∪ B

Use SQL keyword `UNION`. Tables must be compatible ... have the same attributes (column headings).

Product (maker, model, type)
PC (model, speed, ram, hd, price)
Laptop (model, speed, ram, hd, screen, price)

Find the model number and price of all PCs and Laptops made by manufacturer 'B'

```
SELECT   PC.model, PC.price
FROM     Product, PC
WHERE    Product.model = PC.model
AND      Product.maker = 'B'

UNION

SELECT   Laptop.model, Laptop.price
FROM     Product, Laptop
WHERE    Product.model = Laptop.model
AND      Product.maker = 'B'
```

Intersection

Intersection A ∩ B

Use SQL keyword `INTERSECT`. Tables must be compatible.

Product (maker, model, type)
PC (model, speed, ram, hd, price)
Laptop (model, speed, ram, hd, screen, price)

Find those manufacturers that sell both PCs and Laptops

```
SELECT   maker
FROM     Product
WHERE    type = 'pc'

INTERSECT

SELECT   maker
FROM     Product
WHERE    type = 'laptop'
```

Difference

Use SQL keyword MINUS. Tables must be compatible.

Product (maker, model, type)
PC (model, speed, ram, hd, price)
Laptop (model, speed, ram, hd, screen, price)

Find those manufacturers that sell Laptops, but not PCs

```
SELECT   maker
FROM     Product
WHERE    type = 'laptop'

MINUS

SELECT   maker
FROM     Product
WHERE    type = 'pc'
```

Unlike SELECT operator,

UNION, INTERSECT, MINUS removes duplicates.

Thus they convert bags to sets first and then return

sets and not *Bags*

Example:

T1: 1, 2, 2, 2, 3, 4, 4

T2: 2, 3, 4, 4, 4, 5

SELECT * FROM T1

UNION (or INTERSECT or MINUS operator)

SELECT * FROM T2

T1 converted to set: 1, 2, 3, 4

T2 converted to set : 2, 3, 4, 5

T1 UNION T2: 1, 2, 3, 4, 5

T1 INTERSECT T2: 2, 3, 4

T1 MINUS T2: 1

If you want duplicates you must use the keyword ALL.

Example:

T1: 1, 2, 2, 2, 3, 4, 4

T2: 2, 3, 4, 4, 4, 5

SELECT * FROM T1

UNION ALL (or INTERSECT or MINUS)

SELECT * FROM T2

T1 UNION ALL T2: 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 5

T1 INTERSECT ALL T2: 2, 3, 4, 4

T1 EXCEPT ALL T2: 1, 2, 2

For example

```
SELECT   maker
FROM     Product
WHERE    type = 'pc'

UNION ALL

SELECT   maker
FROM     Product
WHERE    type = 'laptop'
```

If a maker appears *a* times in the first set and *b* times in the second set then it appears (*a*+*b*) times in the final set when UNION ALL is used.

For example

```
SELECT   maker
FROM     Product
WHERE    type = 'pc'

INTERSECT ALL

SELECT   maker
FROM     Product
WHERE    type = 'laptop'
```

If a maker appears *a* times in the first set and *b* times in the second set then it appears MIN(*a*,*b*) times in the final set when INTERSECT ALL is used.

For example

```
SELECT   maker
FROM     Product
WHERE    type = 'pc'

EXCEPT ALL

SELECT   maker
FROM     Product
WHERE    type = 'laptop'
```

If a maker appears *a* times in the first set and *b* times in the second set then it appears MAX(*0*,*a*-*b*) times in the final set when EXCEPT ALL is used.

For example

ORACLE doesn’t support INTERSECT ALL and MINUS ALL

ONLY UNION ALL is supported.