

# Database Systems CSE 303

## Lecture 04

2016  
Queries Over Multiple Table

## Problems with single table

Product

Add company info like country, stock price

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

Company

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

## Solution to the problem: Split

Product

Foreign key

Company

Primary Key

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

CSE 303:Ashikur Rahman

5

How to get the information back?

CSE 303:Ashikur Rahman

6

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

$R \times S$

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

CSE 303:Ashikur Rahman

7

## Join

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

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

$R \times S$

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

CSE 303:Ashikur Rahman

8

## Join

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

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

CSE 303:Ashikur Rahman

9

## 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 natural join S*

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

CSE 303:Ashikur Rahman

10

10

## Meaning (Semantics) of SQL Queries Using Cross Product

```
SELECT a1, a2, ..., ak
FROM R1, R2, ..., Rn
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
```

CSE 303:Ashikur Rahman

11

## Joins

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

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

Product join Company

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

CSE 303:Ashikur Rahman

12

## 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

CSE 303:Ashikur Rahman

13

## Joins

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
Multi Touch	\$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

CSE 303:Ashikur Rahman

14

## 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')
```

CSE 303:Ashikur Rahman

15

## A Subtlety about Joins

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

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

CSE 303:Ashikur Rahman

16

## A Subtlety about Joins

Product				Company		
Name	Price	Category	Manufacturer	Cname	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
Single Touch	\$149.99	Photography	Canon	Hitachi	15	Japan
Multi Touch	\$203.99	Household	Hitachi			

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



Country
??
??

What is the problem ?

CSE 303:Ashikur Rahman

17

## A Subtlety about Joins

Product				Company		
Name	Price	Category	Manufacturer	Cname	StockPrice	Country
Gizmo	\$19.99	Gadgets	GizmoWorks	GizmoWorks	25	USA
Powergizmo	\$29.99	Gadgets	GizmoWorks	Canon	65	Japan
Single Touch	\$149.99	Photography	Canon	Hitachi	15	Japan
Multi Touch	\$203.99	Household	Hitachi			

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



Country
USA
USA

Duplicates !  
What's the solution ?

Use 'distinct' keyword

CSE 303:Ashikur Rahman

18

## 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
```

CSE 303:Ashikur Rahman

19

## 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

CSE 303:Ashikur Rahman

20

## Union

### Union $A \cup B$

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'

CSE 303:Ashikur Rahman

21

```
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'
```

CSE 303:Ashikur Rahman

22

## Intersection

### Intersection $A \cap B$

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

CSE 303:Ashikur Rahman

23

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

INTERSECT

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

CSE 303:Ashikur Rahman

24

## 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, 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.

CSE 303:Ashikur Rahman

29

## 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  $\text{MIN}(a,b)$  times in the final set when INTERSECT ALL is used.

CSE 303:Ashikur Rahman

30

## 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  $\text{MAX}(0,a-b)$  times in the final set when EXCEPT ALL is used.

CSE 303:Ashikur Rahman

31

## For example

ORACLE doesn't support INTERSECT ALL and MINUS ALL  
ONLY UNION ALL is supported.

CSE 303:Ashikur Rahman

32