Introduction to Database Systems CSE 414

Lecture 4: SQL Joins and Aggregates

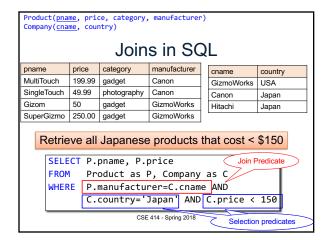
CSE 414 - Spring 2018

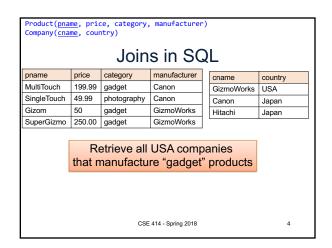
Announcements

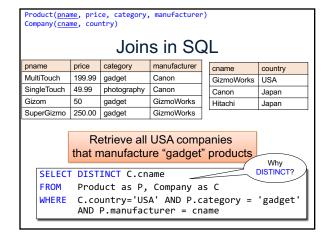
- · WQ1 due on Tuesday
 - No late days allowed
- · HW1 due on Tuesday
 - Submit using script in repo
- · WQ2 and HW2 will be out tomorrow
 - Both are 1 week assignments
- · Check website for up to date OHs

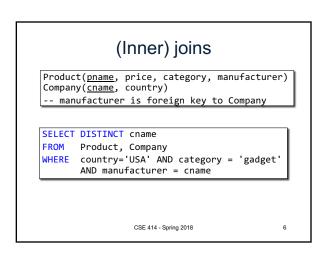
CSE 414 - Spring 2018

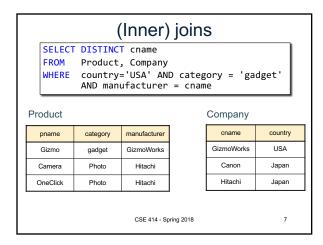
2

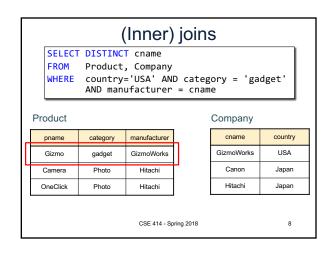


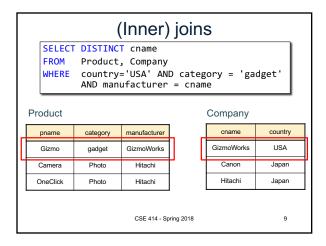


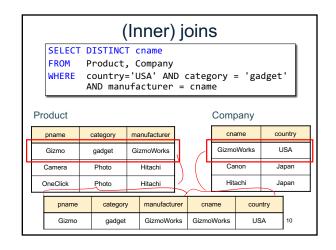


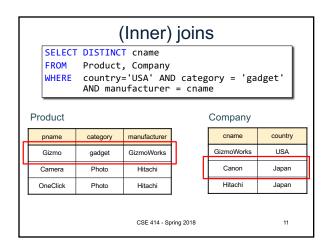


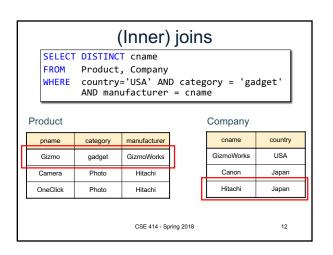












```
(Inner) joins

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

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

```
(Inner) Joins

SELECT x1.a1, x2.a2, ... xm.am
FROM R1 as x1, R2 as x2, ... Rm as xm
WHERE Cond

topls in R1

for x1 in R1:

for x2 in R2:

...

for xm in Rm:

if Cond(x1, x2...):

output(x1.a1, x2.a2, ... xm.am)

This is called nested loop semantics since we are interpreting what a join means using a nested loop
```

```
Another example

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

Find US companies that manufacture both 'gadgets' and 'photo' products

CSE 414-Spring 2018 15
```

```
Another example

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

Find US companies that manufacture both 'gadgets' and 'photo' products

SELECT DISTINCT z.cname
FROM Product x. Company z
WHERE z.country = 'USA'
AND x.manufacturer = z.cname
AND x.category = 'gadget'
AND x.category = 'gadget'
AND x.category = 'photography;

CSE 414-Spring 2018

16
```

```
Another example

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

Find US companies that manufacture both
'gadgets' and 'photo' products

SELECT DISTINCT z.cname
FROM Product x, Company z
WHERE z.country = 'USA'
AND x.manufacturer = z.cname
AND (x.category = 'gadget'
OR x.category = 'photography);

CSE 414-Spring 2018

17
```

```
Another example

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

Find US companies that manufacture both
'gadgets' and 'photo' products

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

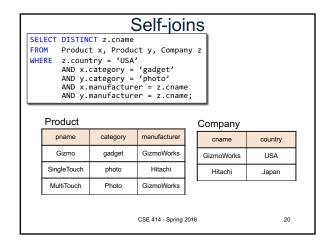
18
```

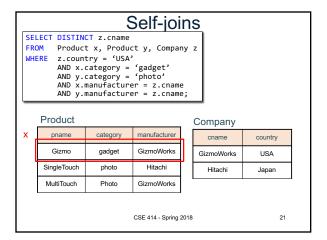
Self-Joins and Tuple Variables

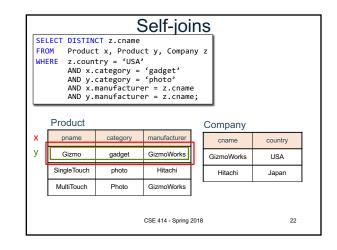
Find US companies that manufacture both 'gadgets' and 'photo' products

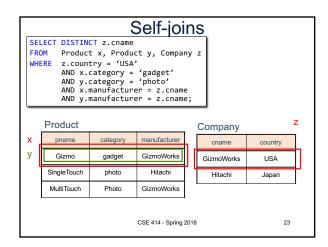
- Joining Product with Company is insufficient: need to join Product, with Product, and with Company
- When a relation occurs twice in the FROM clause we call it a self-join; in that case we must use tuple variables (aka table aliases) (why?)

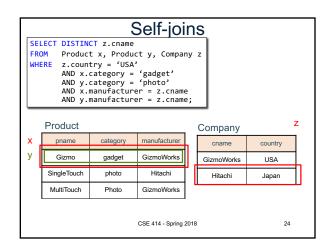
CSE 414 - Spring 2018

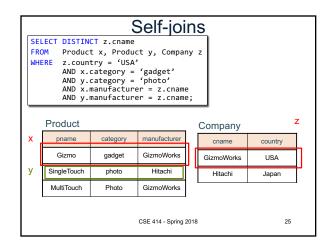


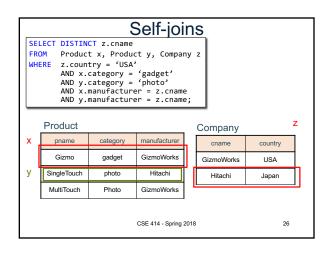


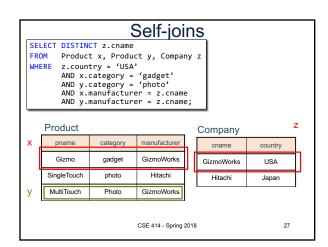


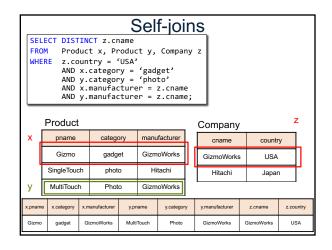


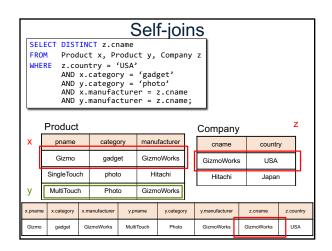




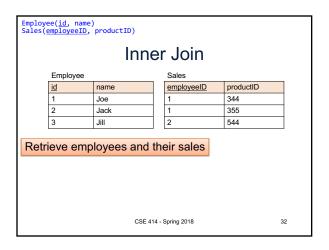


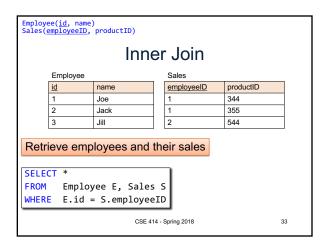


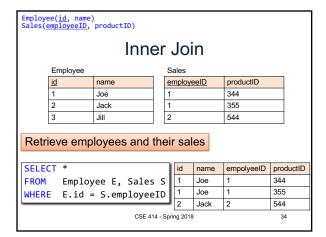


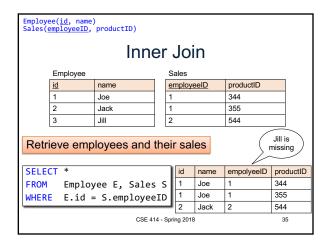


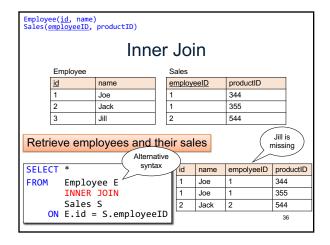
Joins in SQL • The join we have just seen is sometimes called an inner join - Each row in the result must come from both tables in the join • Sometimes we want to include rows from only one of the two table: outer join

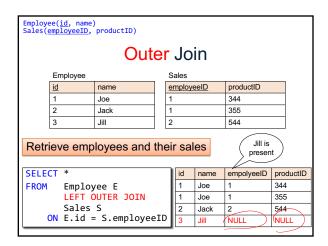


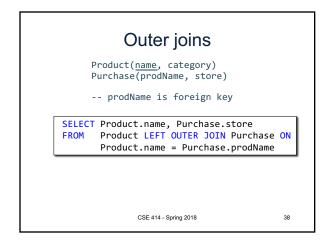


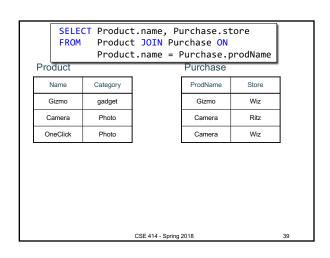


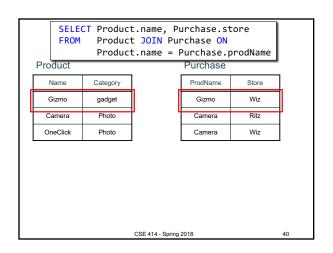


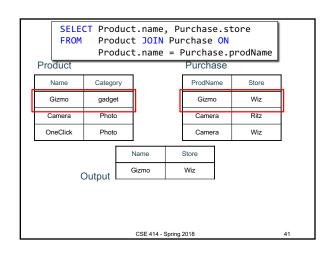


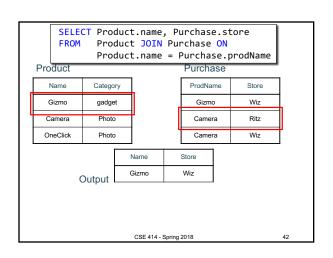


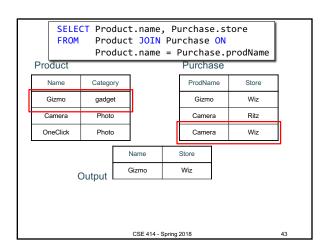


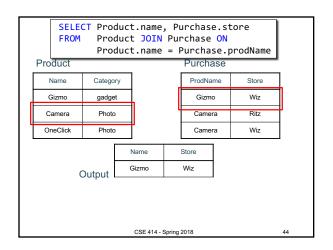


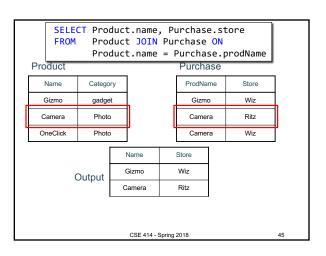


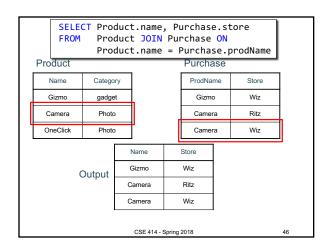


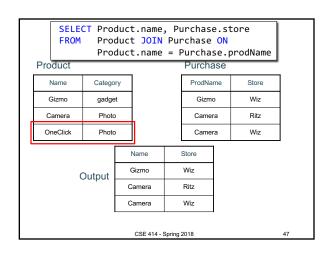


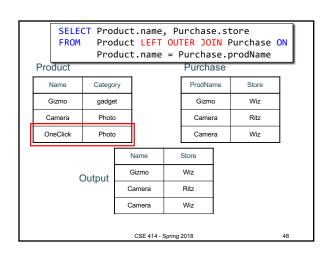


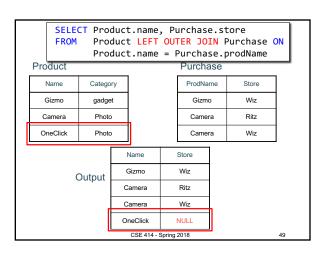


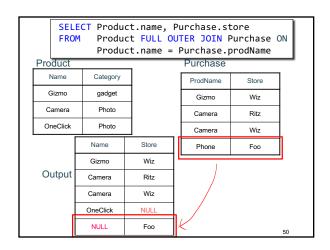












Outer Joins tableA (LEFT/RIGHT/FULL) OUTER JOIN tableB ON p • Left outer join: - Include tuples from tableA even if no match • Right outer join: - Include tuples from tableB even if no match • Full outer join: - Include tuples from both even if no match • In all cases: - Patch tuples without matches using NULL

51

CSE 414 - Spring 2018