Introduction to Database Systems **CSE 414**

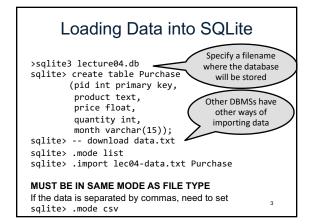
Lecture 4: SQL Joins and Aggregates

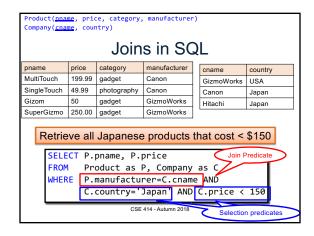
CSE 414 - Autumn 2018

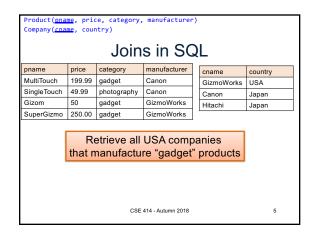
Announcements

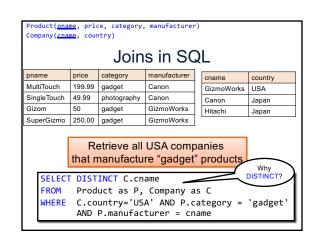
- · Homework 2 out now
 - git pull upstream master to get the starter code
 - Due Tuesday Oct. 9 at midnight
- · Web quiz 1 due Friday midnight
- Section tomorrow important for HW2

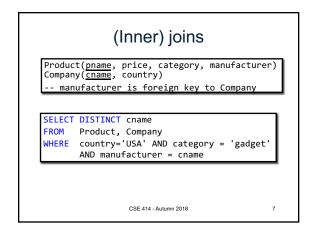
CSE 414 - Autumn 2018

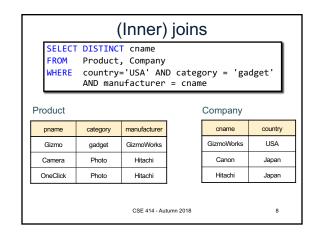


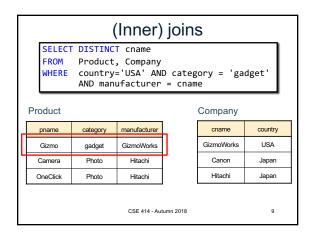


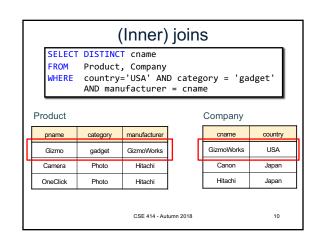


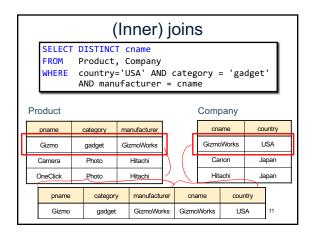


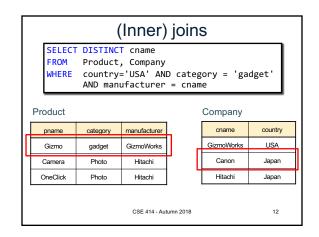


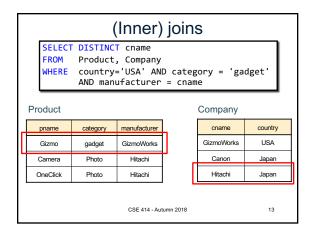


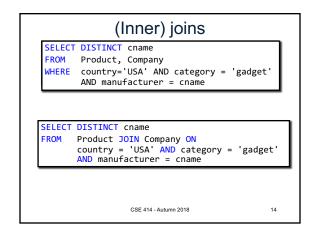










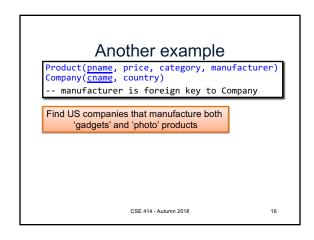


```
(Inner) Joins

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

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
```



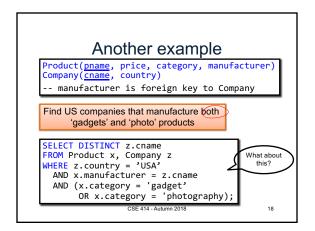
```
Another example

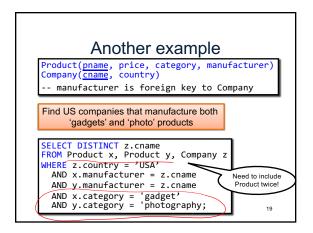
Product(pname, price, category, manufacturer)
Company(cname, country)
-- manufacturer is foreign key to Company

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

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 = 'photography;

CSE 414-Autumn 2018
```





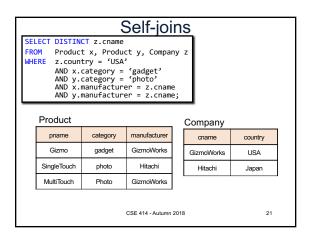
Self-Joins and Tuple Variables

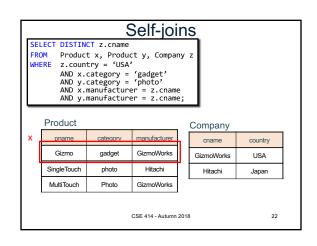
Find US companies that manufacture both 'gadgets' and 'photo' products

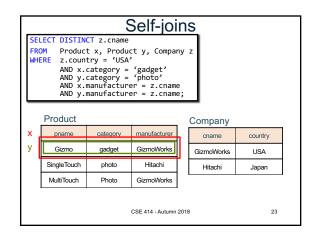
- Joining Product with Company is insufficient: need to join Product, with Product, and with Company
- When a relation occurs twice in the FROM clause we call it a self-join; in that case we must use tuple variables (aka table aliases) (why?)

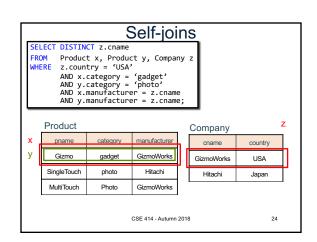
CSE 414 - Autumn 2018

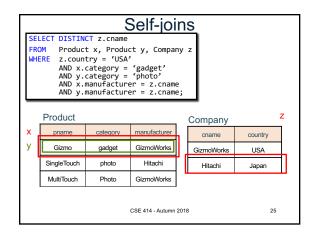
20

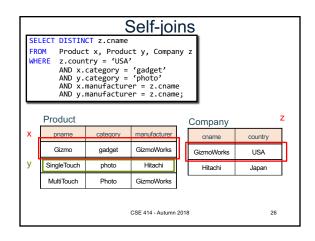


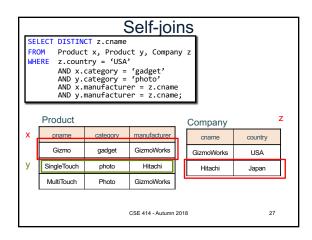


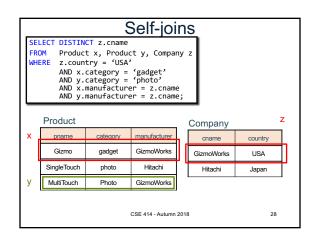


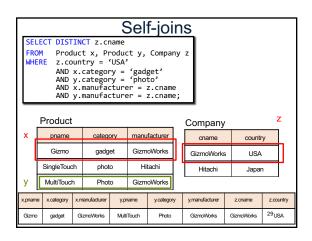


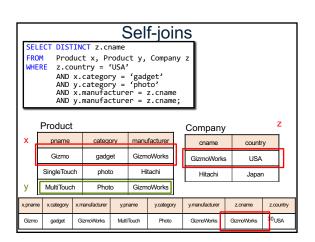












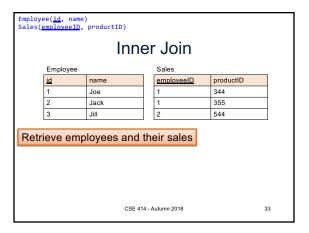


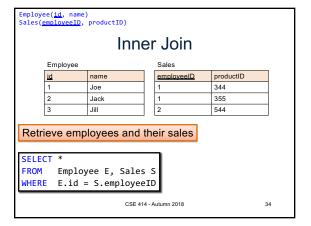
- The join we have just seen is sometimes called an inner join
 - Each row in the result must come from both tables in the join
- Sometimes we want to include rows from only one of the two table: outer join

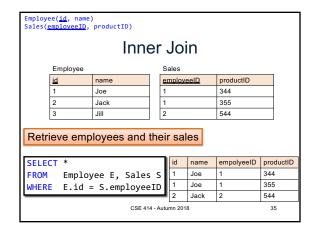
CSE 414 - Autumn 2018

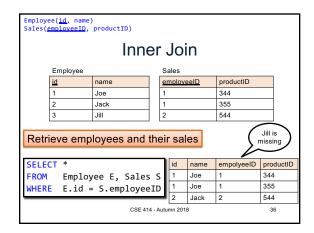
nn 2018

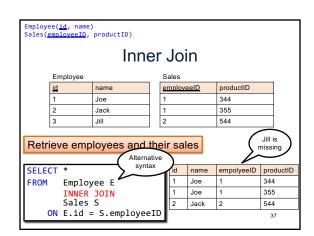
32

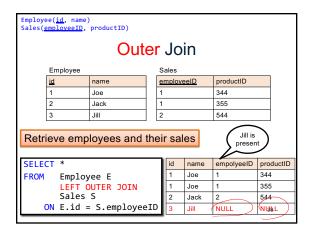


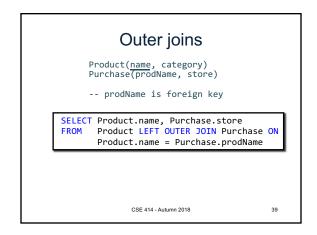


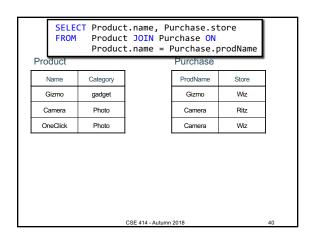


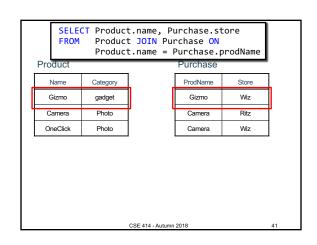


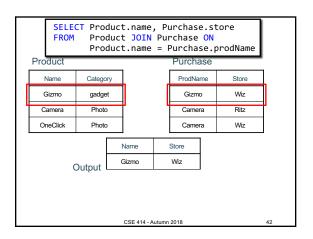


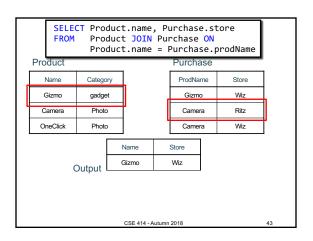


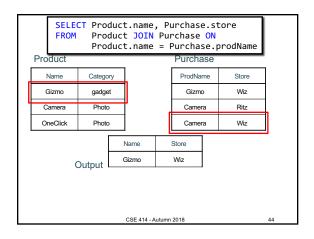


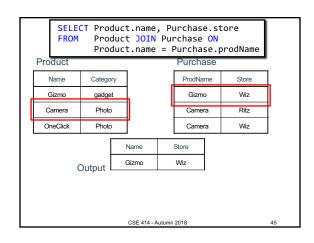


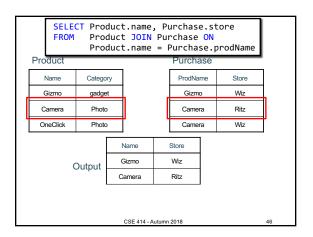


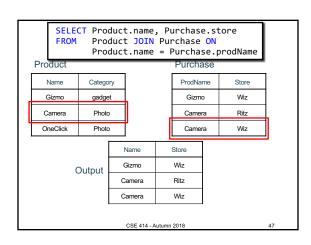


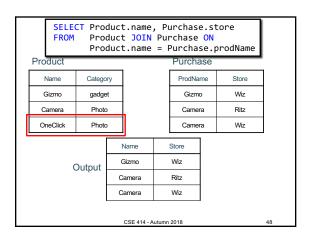


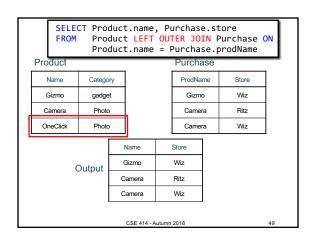


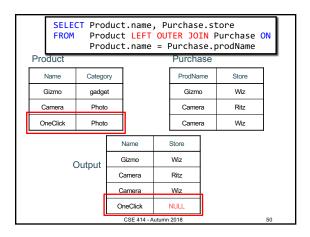


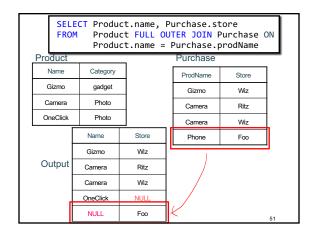












Outer Joins tableA (LEFT/RIGHT/FULL) OUTER JOIN tableB ON p • Left outer join: - Include tuples from tableA even if no match • Right outer join: - Include tuples from tableB even if no match • Full outer join:

Include tuples from both even if no match

In all cases:

- Patch tuples without matches using NULL

CSE 414 - Autumn 2018

52

Aggregates in SQL

Simple Aggregations

Five basic aggregate operations in SQL

```
select count(*) from Purchase
select sum(quantity) from Purchase
select avg(price) from Purchase
select max(quantity) from Purchase
select min(quantity) from Purchase
```

Except count, all aggregations apply to a single attribute

CSE 414 - Autumn 2018

55

Aggregates and NULL Values Null values are not used in aggregates

CSE 414 - Autumn 2018

