# Introduction to Data Management CSE 344

Lectures 4: Aggregates in SQL

CSE 344 - Winter 2016

### **Announcements**

- · Did you remember the web guiz yesterday?
- HW 1 is due Tuesday (tomorrow), 11pm
- · Next web quiz is out due next Sunday
- · HW2 coming out on Wednesday

CSE 344 - Winter 2016

## Outline

- Inner joins (6.2, review)
- Outer joins (6.3.8)
- Aggregations (6.4.3 6.4.6)
- Examples, examples, examples...

CSE 344 - Winter 2016

# (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)

> > CSE 344 - Winter 2016

Nested loop semantics

# (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: semantics if Cond(x1, x2...):
outp ut(x1.a1, x2.a2, ... xm.am)

CSE 344 - Winter 2016

Nested loop

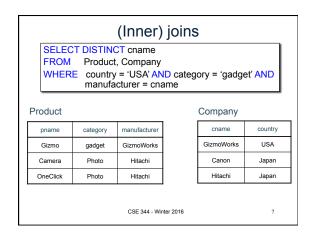
# (Inner) joins

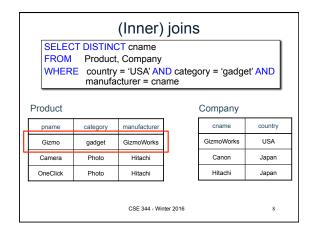
Company(<u>cname</u>, country)
Product(<u>pname</u>, price, category, manufacturer)
– manufacturer is foreign key

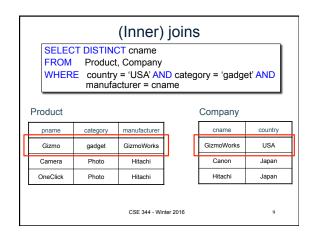
SELECT DISTINCT cname FROM Product, Company

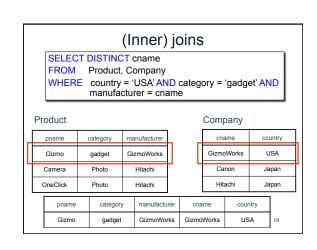
WHERE country = 'USA' AND category = 'gadget' AND manufacturer = cname

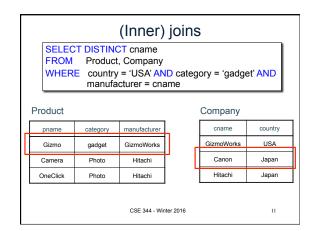
CSE 344 - Winter 2016

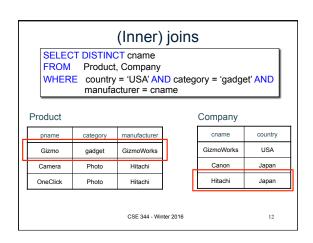


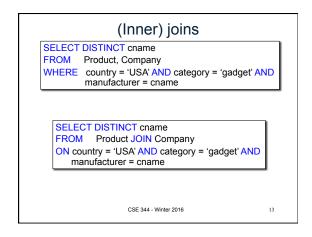








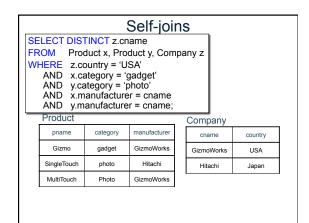


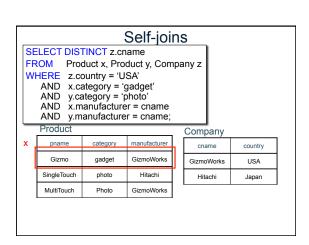


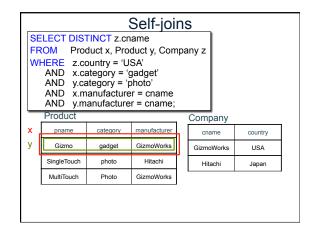
# Self-Joins and Tuple Variables

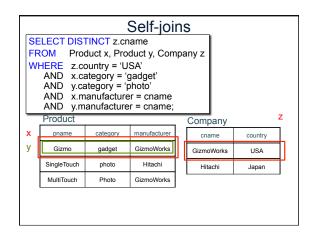
- Find all companies that manufacture both products in the 'gadgets' category and in the 'photo' category
- Just joining Product with Company is insufficient: instead need to join Product, with Product, with Company
- When a relation occurs twice in the FROM clause we call it a self-join; in that case we must use tuple variables (why?)

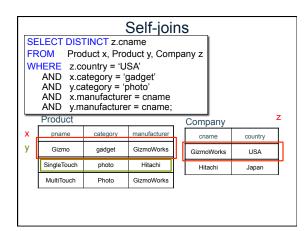
CSE 344 - Winter 2016

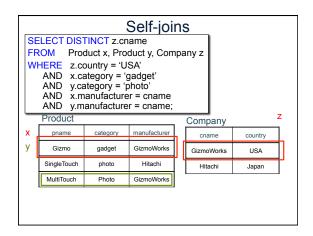


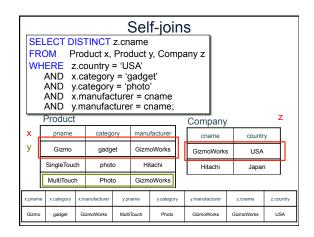


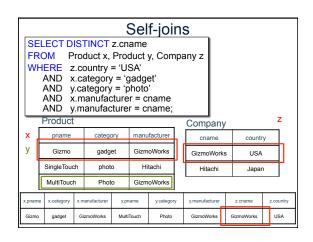






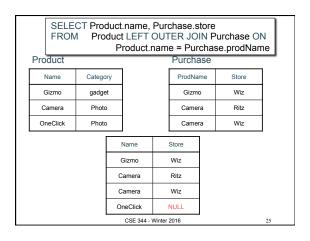






# Outer joins Product(name, category) Purchase(prodName, store) -- prodName is foreign key An "inner join": SELECT Product.name, Purchase.store FROM Product, Purchase WHERE Product.name = Purchase.prodName Same as: SELECT Product.name, Purchase.store FROM Product JOIN Purchase ON Product.name = Purchase.prodName But some Products are not listed! Why?





### **Outer Joins**

- · Left outer join:
  - Include the left tuple even if there's no match
- · Right outer join:
  - Include the right tuple even if there's no match
- · Full outer join:
  - Include both left and right tuples even if there's no match

CSE 344 - Winter 2016

```
Aggregation in SQL
                                   Specify a filename
>sqlite3 lecture04
                                   where the database
                                     will be stored
sqlite> create table Purchase
        (pid int primary key,
        product text,
                                   Other DBMSs have
         price float,
                                    other ways of
         quantity int,
                                    importing data
         month varchar(15));
sqlite> -- download data.txt
sqlite> .import lec04-data.txt Purchase
                   CSE 344 - Winter 2016
```

## Comment about SQLite

- One cannot load NULL values such that they are actually loaded as null values
- · So we need to use two steps:
  - Load null values using some type of special value
  - Update the special values to actual null values

# 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 344 - Winter 2016

## Aggregates and NULL Values

Null values are not used in aggregates

insert into Purchase
values(12, 'gadget', NULL, NULL, 'april')
Let's try the following
 select count(\*) from Purchase
 select count(quantity) from Purchase

select sum(quantity) from Purchase
select sum(quantity)
from Purchase

where quantity is not null;

# Aggregates and NULL Values Null values are not used in aggregates insert into Purchase values(12, 'gadget', NULL, NULL, 'april') Let's try the following select count(\*) from Purchase select count(quantity) from Purchase select sum(quantity) from Purchase select sum(quantity) from Purchase where quantity is not null;

