Introduction to Data Management **CSE 344**

Lecture 6: Nested Queries in SQL

CSE 344 - Winter 2016

Announcements

- · Webquiz 2 is due on Sunday
- · Homework 2 is due on Tuesday

CSE 344 - Winter 2016

Lecture Goals

- · Today we will learn how to write (even) more powerful SQL queries
- · Reading: Ch. 6.3

CSE 344 - Winter 2016

Subqueries

- A subquery is a SQL query nested inside a larger query
- · Such inner-outer queries are called nested queries
- · A subquery may occur in:
 - A SELECT clause
 - A FROM clause
 - A WHERE clause
- Rule of thumb: avoid writing nested queries when possible; keep in mind that sometimes it's impossible

CSE 344 - Winter 2016

Subqueries...

- Can return a single constant and this constant can be compared with another value in a WHERE clause
- Can return relations that can be used in various ways in WHERE clauses
- Can appear in FROM clauses, followed by a tuple variable that represents the tuples in the result of the subquery
- · Can appear as computed values in a SELECT clause

CSE 344 - Winter 2016

1. Subqueries in SELECT

Product (pname, price, cid) Company(cid, cname, city)

For each product return the city where it is manufactured

CSE 344 - Winter 2016

1. Subqueries in SELECT

Product (<u>pname</u>, price, cid) Company(<u>cid</u>, cname, city)

For each product return the city where it is manufactured

SELECT X.pname, (SELECT Y.city FROM Company Y WHERE Y.cid=X.cid) as City FROM Product X

CSE 344 - Winter 2016

1. Subqueries in SELECT

Product (<u>pname</u>, price, cid) Company(<u>cid</u>, cname, city)

For each product return the city where it is manufactured

SELECT X.pname, (SELECT Y.city FROM Company Y WHERE Y.cid=X.cid) as City FROM Product X

What happens if the subquery returns more than one city? We get a runtime error

(SQLite simply ignores the extra values)

CSE 344 - Winter 2016

1. Subqueries in SELECT

Product (<u>pname</u>, price, cid) Company(<u>cid</u>, cname, city)

For each product return the city where it is manufactured

SELECT X.pname, (SELECT Y.city
FROM Company Y
WHERE Y.cid=X.cid) as City
FROM Product X

What happens if the subquery returns more than one city? We get a runtime error

(SQLite simply ignores the extra values)

CSE 344 - Winter 2016

1. Subqueries in SELECT

Product (<u>pname</u>, price, cid) Company(<u>cid</u>, cname, city)

For each product return the city where it is manufactured

"Correlated"

"Correlated"

SELECT X.pname, (SELECT Y.city
FROM Company Y
WHERE Y.cid=X.cid) as City

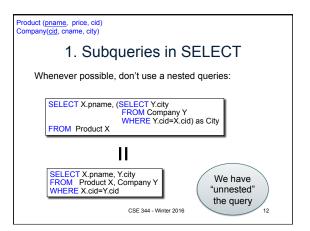
FROM Product X

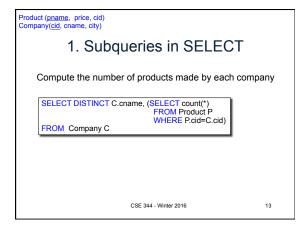
What happens if the subquery returns more than one city? We get a runtime error

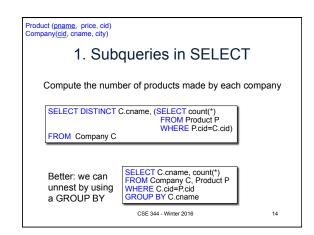
(SQLite simply ignores the extra values)

CSE 344 - Winter 2016 10

Product (pname, price, cid) Company(cid. cname, city) 1. Subqueries in SELECT Whenever possible, don't use a nested queries: SELECT X.pname, (SELECT Y.city FROM Company Y WHERE Y.cid=X.cid) as City FROM Product X SELECT X.pname, Y.city FROM Product X. Company Y WHERE X.cid=Y.cid







Product (pname, price, cid)
Company(cid, cname, city)

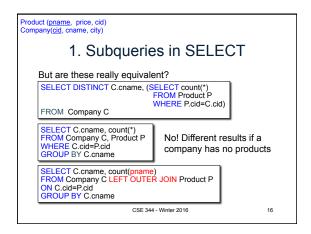
1. Subqueries in SELECT

But are these really equivalent?

SELECT DISTINCT C.cname, (SELECT count(*) FROM Product P WHERE P.cid=C.cid)

FROM Company C

SELECT C.cname, count(*) FROM Company C, Product P WHERE C.cid=P.cid
GROUP BY C.cname



Product (pname, price, cid)
Company(cid, cname, city)

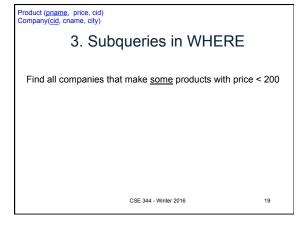
2. Subqueries in FROM

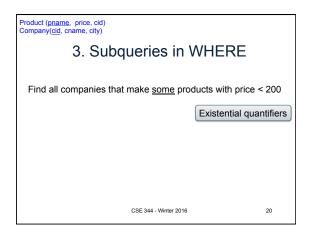
Find all products whose prices is > 20 and < 500

SELECT X.pname
FROM (SELECT * FROM Product AS Y WHERE price > 20) as X WHERE X.price < 500

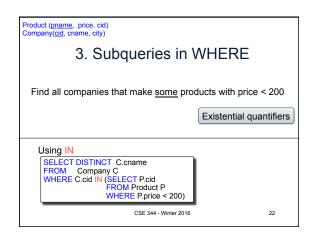
Unnest this query!

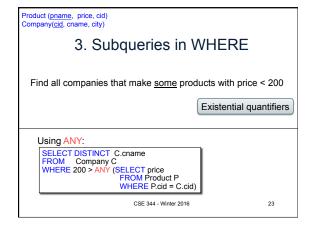
2. Subqueries in FROM At the end of the lecture we will see that sometimes we really need a subquery and one option will be to put it in the FROM clause.

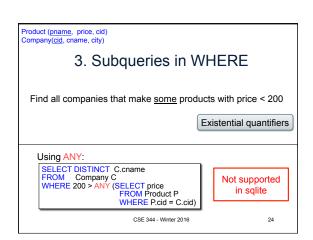


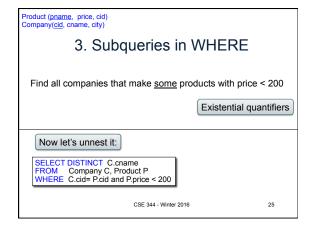


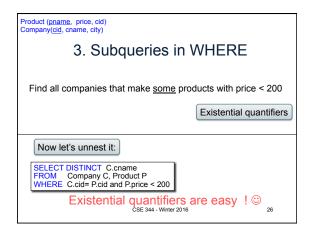












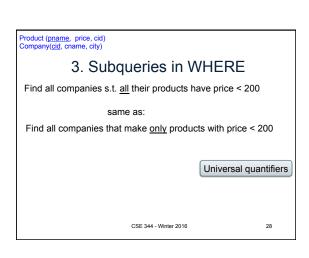
Product (pname, price, cid)
Company(cid, chame, city)

3. Subqueries in WHERE

Find all companies s.t. all their products have price < 200
same as:

Find all companies that make only products with price < 200

CSE 344 - Winter 2016 27



Product (pname, price, cid)
Company(cid. cname, city)

3. Subqueries in WHERE
Find all companies s.t. all their products have price < 200
same as:
Find all companies that make only products with price < 200

Universal quantifiers

Universal quantifiers are hard!

Product (pname, price, cid)
Company(cid, cname, city)

3. Subqueries in WHERE

Find all companies s.t. all their products have price < 200

1. Find the other companies: i.e. s.t. some product ≥ 200

SELECT DISTINCT C.cname
FROM Company C
WHERE C.cid IN (SELECT P.cid
FROM Product P
WHERE P.price >= 200)

```
Product (pname. price, cid)
Company(cid. cname, city)

3. Subqueries in WHERE

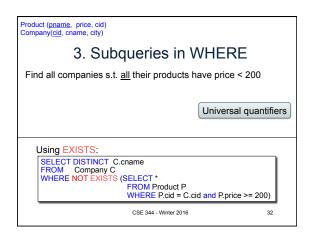
Find all companies s.t. all their products have price < 200

1. Find the other companies: i.e. s.t. some product ≥ 200

SELECT DISTINCT C.cname
FROM Company C
WHERE C.cid IN (SELECT P.cid
FROM Product P
WHERE P.price >= 200)

2. Find all companies s.t. all their products have price < 200

SELECT DISTINCT C.cname
FROM Company C
WHERE C.cid NOT IN (SELECT P.cid
FROM Product P
WHERE P.price >= 200)
```



```
Product (pname, price, cid)
Company(cid, cname, city)

3. Subqueries in WHERE

Find all companies s.t. all their products have price < 200

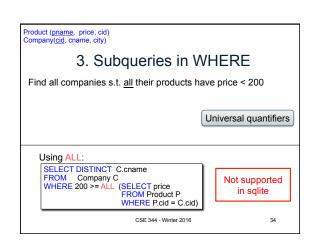
Universal quantifiers

Using ALL:

SELECT DISTINCT C.cname
FROM Company C
WHERE 200 >= ALL (SELECT price
FROM Product P
WHERE P.cid = C.cid)

CSE 344 - Winter 2016

33
```



Question for Database Fans and their Friends

• Can we unnest the universal quantifier query ?

CSE 344 - Winter 2016

35

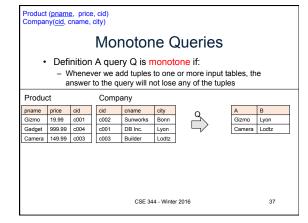
Product (pname, price, cid)
Company(cid, cname, city)

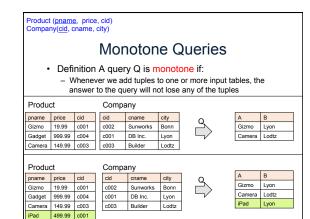
Monotone Queries

• Definition A query Q is monotone if:

- Whenever we add tuples to one or more input tables, the answer to the query will not lose any of the tuples

CSE 344 - Winter 2016 36



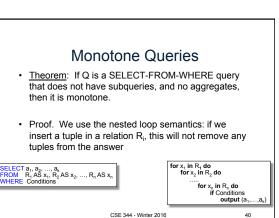


Monotone Queries

 Theorem: If Q is a SELECT-FROM-WHERE query that does not have subqueries, and no aggregates, then it is monotone.

CSE 344 - Winter 2016

39



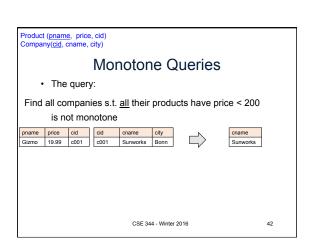
Product (pname, price, cid)
Company(cid, cname, city)

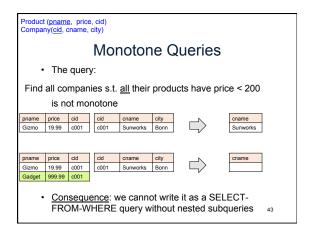
Monotone Queries

• The query:

Find all companies s.t. all their products have price < 200 is not monotone

CSE 344 - Winter 2016 41





Queries that must be nested

Queries with universal quantifiers or with negation

CSE 344 - Winter 2016

1 - Winter 2016

Queries that must be nested

- Queries with universal quantifiers or with negation
- Queries that use aggregates in certain ways
 - Note: sum(..) and count(*) are NOT monotone, because they do not satisfy set containment
 - select count(*) from R is not monotone!

CSE 344 - Winter 2016

45