Database Systems CSE 414

Lecture 6: Nested Queries in SQL

CSE 414 - Spring 2017

Announcements

- WQ2 is due on Sunday 11pm
 - no late days
- · HW2 is due on Tuesday 11pm

CSE 414 - Spring 2017

2017

Lecture Goals

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

CSE 414 - Spring 2017

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 nested queries when possible; keep in mind that sometimes it's impossible
 - (though use in FROM is often not as bad)

CSE 414 - Spring 2017

4

6

2

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 414 - Spring 2017

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

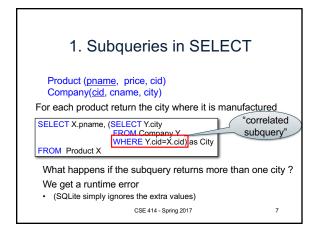
SELECT X.pname, (SELECT Y.city
FROM Company Y
WHERE Y.cid=X.cid) as City
FROM Product X

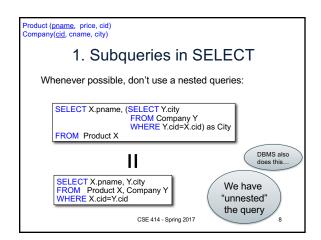
What happens if the subquery returns more than one city? We get a runtime error

(SQLite simply ignores the extra values)

CSE 414 - Spring 2017

1





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

1. Subqueries in SELECT

Compute the number of products made by each company

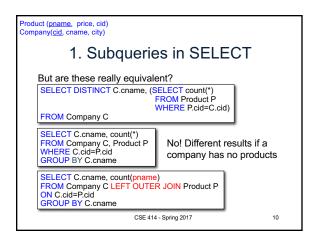
SELECT DISTINCT C.cname, (SELECT count(*)
FROM Product P
WHERE P.cid=C.cid)

Better: we can
unnest by using
a GROUP BY

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

CSE 414 - Spring 2017

9



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

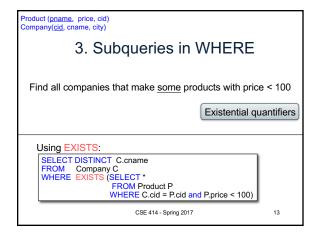
Unnest this query!

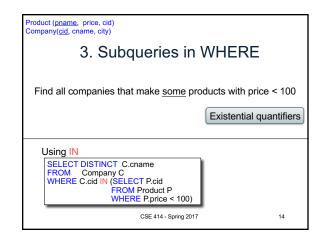
SELECT pname
FROM Product
WHERE price > 20 AND price < 500

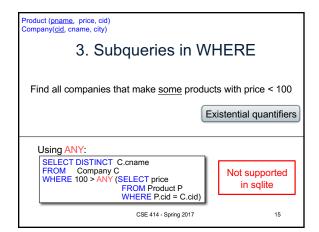
CSE 414 - Spring 2017

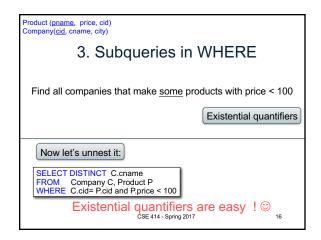
11

2. Subqueries in FROM • We will see that sometimes we really need a subquery – will see most compelling examples next lecture – in that case, we can put it in the FROM clause









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

3. Subqueries in WHERE

Find all companies where all their products have price < 100

same as:
Find all companies that make only products with price < 100

Universal quantifiers

Universal quantifiers are hard!

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

3. Subqueries in WHERE

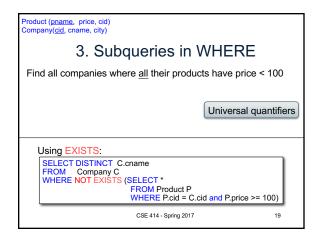
Find all companies where all their products have price < 100

1. Find the other companies: i.e. with some product >= 100

SELECT DISTINCT C.cname
FROM Company C
WHERE C.cid IN (SELECT P.cid
FROM Product P
WHERE P.price >= 100)

2. Find all companies where all their products have price < 100

SELECT DISTINCT C.cname
FROM Company C
WHERE C.cid NOT IN (SELECT P.cid
FROM Product P
WHERE P.price >= 100)



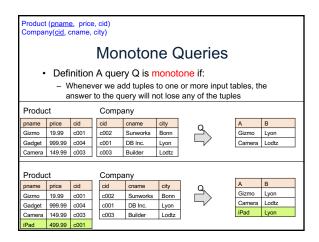


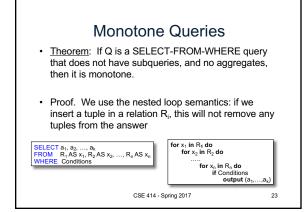
Question for Database Fans and their Friends

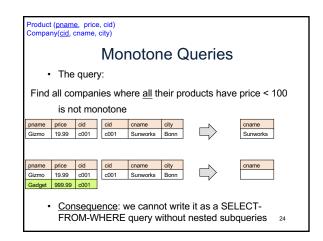
• Can we unnest the universal quantifier query?

CSE 414 - Spring 2017

21







Queries that must be nested (that is, cannot be SFW queries)

- Queries with universal quantifiers or negation
- Queries that use aggregates in certain ways
 - Note: sum(..) etc. are NOT monotone
 - select count(*) from R is not monotone!

CSE 414 - Spring 2017

25