# Database Systems
## CMPT 308

## *Lab 4: SQL Queries - The Subqueries Sequel - 20 points*

**Goals**

- Write some interesting SQL queries **using subqueries**. Please do not use joins; save them for the next assignment.
- Enjoy the beauty and accuracy of the relational model.
- Bask in the coolness of the beauty and accuracy of the relational model. (Bask, I say!)
- Work a little for some lab points.

**Before you begin**

Check that your instance of our beloved CAP3 database is **exactly** the same as mine in the script on our class web site.

**Instructions**

Use CAP3 to answer all of these questions. Remember that CAP3 is a snapshot in time and your queries must return the correct answer for all time. **Do not** use primary key values in the queries. E.g., you may not assume that customer "Basics" is 'c007'. Also, please do not use joins to answer these queries; use sub-queries instead.

1. Get the cities of agents booking an order for a customer whose cid is 'c002'.

2. Get the ids of products ordered through any agent who takes at least one order from a customer in Dallas, sorted by pid from highest to lowest. (This is not the same as asking for ids of products ordered by customers in Dallas.)

3. Get the ids and names of customers who did not place an order through agent a01.

4. Get the ids of customers who ordered both product p01 **and** p07.

5. Get the ids of products **not** ordered by any customers who placed any order through agent a07 in pid order from highest to lowest.

6. Get the name, discounts, and city for all customers who place orders through agents in London or New York.

7. Get all customers who have the same discount as that of any customers in Dallas or London

8. Tell me about check constraints: What are they? What are they good for? What's the advantage of putting that sort of thing inside the database? Make up some examples of good uses of check constraints and some examples of bad uses of check constraints. Explain the differences in your examples and argue your case.

**Advice**

Test, test, and test again. Then test some more. When you think you've tested enough, go back and keep testing. Then get someone else to test for you while you test theirs.

Push your work to your GitHub repository early and often. Be sure to write meaningful commit messages.

**Resources**

- Chapters 6.1 - 6.4 in our text, especially 6.3 and 6.4
- SQL tag at Stack Overflow - http://stackoverflow.com/questions/tagged/sql

**Submitting**

Submit your work as a text file with a *.sql* extension. (Put your answer to number 8 inside comments.) Push your work to your GitHub repository **before** the due date (see syllabus). Remember to include your name, the date, and the assignment in the (copious, meaningful, and accurate) comments in your code.