Database Systems

CMPT 308

Lab 6: Interesting and Painful Queries - 20 points

Goals

- Write some downright difficult SQL queries
- Work really hard for 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.

1. Display the name and city of customers who live in **any** city that makes the **most** different kinds of products. (There are two cities that make the most different products. Return the name and city of customers from **either one** of those.)
2. Display the names of products whose priceUSD is strictly above the average priceUSD, in reverse-alphabetical order.
3. Display the customer name, pid ordered, and the total for all orders, sorted by total from high to low.
4. Display all customer names (in alphabetical order) and their total ordered, and nothing more. Use coalesce to avoid showing NULLs.
5. Display the names of all customers who bought products from agents based in Tokyo along with the names of the products they ordered, and the names of the agents who sold it to them.
6. Write a query to check the accuracy of the dollars column in the Orders table. This means calculating Orders.totalUSD from data in other tables and comparing those values to the values in Orders.totalUSD. Display all rows in Orders where Orders.totalUSD is incorrect, if any.
7. What's the difference between a LEFT OUTER JOIN and a RIGHT OUTER JOIN? Give example queries in SQL to demonstrate. (Feel free to use the CAP2 database t make your points here.)

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

- Chapter 6 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 7 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.