

Database Systems

CMPT 308

- Lab 6: Interesting and Painful Queries - 20 points

Goals	Write some fun yet difficult SQL queries. Now you've got to earn the points.
Before you begin	Check that your instance of our beloved CAP database is exactly the same as mine.
Instructions	<p>Use CAP to answer all of these questions. For each question, first write the SQL query yourself and validate your answer against the CAP database as shown on our web site. Then use an AI to generate the SQL and run that. Compare your answer to the results from the AI-generated query and grade the AI's output.</p> <ol style="list-style-type: none">1. Display the cities that makes the most different kinds of products. Experiment with the <i>rank()</i> function.2. Display the names of products whose priceUSD is less than 1% of the average priceUSD, in alphabetical order. from A to Z.3. Display the customer last name, product id ordered, and the totalUSD for all orders made in March of any year, sorted by totalUSD from low to high.4. Display the last name of all customers (in reverse alphabetical order) and their total ordered by customer, and nothing more. Use <i>coalesce</i> to avoid showing NULL totals.5. Display the names of all customers who bought products from agents based in Regina 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 totalUSD 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. If there are any incorrect values, explain why they are wrong. Round to exactly two decimal places.7. Display the first and last name of all customers who are also agents.8. Create a VIEW of all Customer and People data called PeopleCustomers. Then another VIEW of all Agent and People data called PeopleAgents. Then <i>select *</i> from each of them to test them.9. Display the first and last name of all customers who are also agents, this time using the views you created.10. Compare your SQL in #7 (no views) and #9 (using views). The output is the same. How does that work? What is the database server doing internally when it processes the #9 query?11. [Bonus] 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 CAP database to make your points here.)
Advice	Test, test, and test again. Then test some more. You know the deal. Push your work to your GitHub repository early and often. Write meaningful commit messages.
Submitting	Submit your work as a text file with a <i>.sql</i> extension or Markdown (even better) before the due date (see syllabus).