

## Lab 1. Aggregation and Simple Joins

### Objectives:

- Practice performing single- and multi- table queries that incorporate aggregation
- Practice performing single- and multi- table queries that include a subquery
- Practice performing single- and multi- table queries that use standard Oracle-provided functions

### Questions

Check-off questions are marked in green.

Q1-Q23: 4 pts each. Q24: 6 pts, Q25: 2pts.

Order of Execution	
1	<p>What's the order of execution of the query below?</p> <p>Please include the answer using SQL comments in your submission. Use the same format as below but please update the numbers to reflect the execution order.</p> <ol style="list-style-type: none"> <li>1. SELECT NAME, COUNT(*), AVG(RATING)</li> <li>2. FROM BOOKSHELF</li> <li>3. WHERE RATING&gt;1</li> <li>4. GROUP BY CATEGORY_NAME</li> <li>5. HAVING CATEGORY_NAME LIKE 'A%'</li> <li>6. ORDER BY COUNT(*)</li> </ol>
Aggregation and Subqueries	
2	Identify <i>distinctly</i> all telephone <u>area codes</u> (e.g., the first 3 digits of the phone number) of Colorado (state is 'CO') customers.
3	Identify all telephone <u>area codes</u> of Colorado customers, as well as the <u>number of customers</u> in each area. Sort the results by the number of customers in <i>descending</i> order.
4	Identify the specific telephone <u>area code</u> containing the <i>largest</i> number of Colorado customers. Display the area code only. <b>(DO NOT hardcode the area code)</b>
5	Identify all customers living in the most popular Colorado area code. Display their name in <u>last name, (comma) first name</u> format (e.g. Simpson, Lisa), the <u>city</u> and <u>state</u> in which they live, and their <u>telephone number</u> . (Hint: the code from the previous question is a good starting point for this question)
6	Briefly explain why might we want to know the results of question 4? What business implications does it have? Please include the answer using SQL comments in your submission.
7	Identify the <u>customer ID</u> and <u>number of orders</u> placed by each customer during August 2010 (OrderDate is August 2010).

	Sort the results by the number of orders in <i>descending</i> order.
8	Identify the <u>maximum number of orders</u> placed by a customer during August 2010. Display only the maximum number of orders.
9	Identify the <u>customer ID</u> of the customer who placed the <u>largest number of orders</u> during August 2010, as well as the number of orders placed. (Hint: the code from the previous questions is a good starting point for this question)
10	Identify all customers who placed <i>greater</i> than the average number of orders during August 2010. Display the <u>customer ID</u> and the <u>number of orders</u> each of these customers placed. Sort the results by number of orders in <i>descending</i> order.
11	Identify all customers who placed <i>fewer</i> than the average number of orders during August 2010. Display the <u>customer ID</u> and the <u>number of orders</u> each of these customers placed. Sort the results by number of orders in <i>ascending</i> order.
12	Briefly explain why a business might want to know the results of questions 10 & 11? Please include the answer using SQL comments in your submission.
<b>Inner Join</b>	
13	Display the <u>customer ID</u> , <u>company name</u> , contact name in <u>last name, (comma) first name</u> format, (e.g. Simpson, Lisa), and <u>order date</u> in MM.DD.YYYY format (e.g. 12.30.2010) for all Indiana customers who placed orders in 2010. Sort the results by order date from <i>the oldest to the most recent</i> .
14	Display the <u>company name</u> , contact name in <u>title first Initial (dot) last name</u> format e.g. Ms. L. Simpson), <u>order date</u> , and <u>required date</u> for all orders placed by customer with ID C-300001. Sort the results by order date from <i>the oldest to the most recent</i> .
15	For all orders containing 'BOARD GAMES' software, display the <u>order ID</u> , <u>part number</u> , <u>part description</u> , <u>unit price</u> , <u>order quantity</u> , and <u>category name</u> . Sort the results by order quantity in <i>descending</i> order.
16	For all items ordered by customer ID C-300001 on July 14t, 2010, display the <u>order ID</u> , <u>part number</u> , <u>part description</u> , <u>unit price</u> , and <u>order quantity</u> . Sort the results by order quantity from <i>largest to smallest</i> .
17	For all items ordered by 'Bankruptcy Help' (company name) during 2011, display the <u>order date</u> in MM.DD.YYYY format (e.g. 12.30.2010), <u>order ID</u> , <u>part number</u> , <u>part description</u> , <u>unit price</u> , and <u>order quantity</u> . Sort the results first by order date, with the <i>most recent displayed first</i> . Then within a given date, sort by quantity, with the <i>greatest displayed first</i> .
18	For all items ordered by 'Bankruptcy Help' (company name) during 2011, display the <u>order date</u> in MM.DD.YYYY format (e.g. 12.30.2010), <u>order ID</u> , <u>part number</u> , and <u>part</u>

	<p><u>description</u>. In addition, calculate and display the <u>line item total</u> for each item. To calculate the line item total, multiply the unit price by the number of units ordered. Sort the results first by order date, with the <i>most recent displayed first</i>. Then within a given date, sort by quantity, with the <i>greatest displayed first</i>.</p> <p>(Hint: modify your code from question 17)</p>
<b>Inner Join with Aggregation</b>	
<b>19</b>	<p>Display the <u>customer ID</u>, <u>company name</u>, <u>contact name in last name</u>, (comma) <u>first name</u> format (e.g. Simpson, Lisa), and <u>number of orders placed</u> (NOT order quantity) for all Indiana customers who placed orders in January of 2011.</p> <p>Sort the results by numbers of orders placed in <i>ascending</i> order.</p>
<b>20</b>	<p>Display the <u>category name</u> and the <u>average stock level</u> of each category. Display up to 2 decimal places for the average stock level.</p> <p>Sort the results by average stock level in <i>ascending</i> order.</p>
<b>21</b>	<p>Display the <u>category detail</u> and the <u>number of part types</u> in each category (NOT stock level). A category detail consists of category name and category description, and it is formatted as category name: (colon) description (e.g. Software: Games, maps).</p> <p>Sort the results by number of part types in <i>ascending</i> order.</p>
<b>Putting it all together (Inner Join with Aggregation and Subqueries)</b>	
<b>22</b>	<p>Display the <u>weight of the heaviest part</u> in the Software category (CategoryName is 'Software').</p>
<b>23</b>	<p>For each of the Power, Software, and Storage category, display the <u>category name</u> and the <u>weight of the heaviest part</u> in the category.</p> <p>Sort the results by category name in <i>ascending</i> order.</p> <p>(Hint: the code from the previous question is a good starting point for this question)</p>
<b>24</b>	<p>For each of the Power, Software, and Storage category, display the <u>category name</u>, the <u>weight of the heaviest part</u> in the category, and the <u>corresponding part description(s)</u> of the heaviest part(s).</p> <p>Sort the results by category name in <i>ascending</i> order.</p> <p>(Hint: the code from the previous question is a good starting point for this question)</p>
<b>25</b>	<p>Is there anything that can be changed to make it run faster? If no, why? If yes, how? Please include the answer using SQL comments in your submission.</p>