MIS 531: Homework SQL-2 (Part-2)

This assignment consists of a series of SQL questions. For each question, please include your:

- Your final SQL query
- No explanations are needed

Submission Instructions

1) By submitting each query: ***you affirm that the work submitted is your own and that you did not receive unauthorized assistance from anyone outside your two-person team nor did you provide unauthorized assistance to anyone.***

This is a two-person group assignment. Please do not solve the problem for others as you will not be helping them learn. The homework does not require a perfect score to do well (i.e., has built in bonus points). Please note, the maximum score (across both parts of the homework) is 100. In other words, if you score over 100, your score is reduced to 100 (*so there should be no incentive to cheat, instead please focus on learning*).

Please do not share your work or solutions in part or whole. **Policies for academic integrity** are specified in the syllabus. You can approach the instructor with questions or clarifications.

2) Use the D2L Turnitin dropbox to submit a soft copy of EACH question in your assignment separately. Only the final query for each question is required. Please keep the final query at the top of your file so the graders can find it easily if you include intermediate queries (to show how you solved each step).

3) Your queries must be executable. In other words: to grade them, we **must be able to copy-paste from your submitted file and run them in SQL Developer**. Please test this (e.g., on your machine, after preparing your document) to ensure you do not lose points. Please submit a text (ASCII) file (i.e., not a word or pdf file).

4) Please start early. SQL requires a good amount of practice and learning. Also, by leaving work for the last minute, you incur many risks including a potential server outage affecting your work.

5) The submission process takes time (and requires confirmation). Please do NOT wait till close to the deadline to submit your individual queries on D2L (as you may cross over into the late period and automatically incur the 10% penalty). To be fair to all students, we do NOT selectively waive the late penalty.

6) Keep the following instructions in mind while solving the queries.
   - The sub-parts of a query are <u>**CUMULATIVE**</u> (i.e., build on previous parts).
   - Please solve the queries in stages, i.e., <u>write SQL to answer each part to reduce the likelihood of making errors / finding errors easily</u>. Common ways to structure your query-writing include:
     - Start with a single table (e.g., employee data), add additional tables in stages
     - Add aggregation and/or filtering requirements (WHERE / HAVING clauses) in stages
     - Add formatting and sorting requirements in stages

Reminders

- Submit one file (with your final query) for each question
- All column Headings are case sensitive (for the entire homework)
- The sub-parts of each query are CUMULATIVE (requirement b) builds on requirement a), and so on)

Question-4

Notes (this query):

- Consider products with price > 26 (in the products table) as *high-priced* products and all other products (priced 26 or below) as *low-priced*.
- Ignore discounts when calculating sales revenue

Do not assume the product line name will always be unique (even though in the current dataset it is).

a) For each product line, display the product line name (heading: Line Name)
b) Show the number of low-priced products (heading: Low Price) and the sales revenue earned from low priced products (heading: Low Price Revenue, format $999,999.99)
c) Show the number of high-priced products (heading: High Price) and the sales revenue earned from high priced products (heading: High Price Revenue, format $999,999.99).
d) Also show the percentage of total revenue coming from sales of high-priced products (heading: Percent High-price Revenue). Format as a percentage with two decimals, e.g., 55.25%
e) Show the Year to date (YTD) revenue (heading: YTD Revenue, format $999,999.99). YTD revenue is revenue from orders placed (use the orderdate) in the beginning of the current year till today.

Hint: Use WITH (or equivalent). Review sysdate in the tutorial and class examples.

Question-5

a) For each employee, display the employee number (heading: Employee Num), employee last name and first name together (heading: Employee Name) in the following format: "Wildcat, Wilbur". Sort by employee last name.
b) Display a loyalty category for each employee based on the number of years the employee has served with CORP (heading: Employee Loyalty). Use the following table to determine loyalty.

No. of Years	Employee Loyalty Category
Fewer than 3	None
At least 3 but less than 5	Silver
At least 5 but less than 10	Gold
At least 10	Platinum

c) Display the total number of orders an employee has managed (heading: Total Orders). If the employee has no orders, display 0.
d) For each employee, display the total sales revenue in dollars (heading: Total Sales in Dollars, format: $9,999,990.99) across all orders. Consider discount in this calculation. If the employee has no orders, display $0.00. If you are unsure how discount affects sales revenue, please review Question 2 (in Part-1 of the homework).
e) Display the employee rank based on the total number of orders the employee has managed (highest value should be ranked 1). Use a heading: Ranking based on Orders.

f) Display the employee rank based on Total Sales (i.e., what was calculated in step d)), highest value should be rank 1. Use a heading: Ranking based on Sales.
g) Exclude employees who have been with CORP for 20 years or more.

Hint: Use WITH or equivalent. Use RANK().

Question-6

Consider all of the orders that were placed in January 2020. Display the following:

a) The date (format: 'DD-MON-YYYY’, heading: OrderDate). **Include every day of January 2020** (even if there were no orders on that day)**.**
b) The total number of orders placed on that day (heading: Orders Today)
c) The number of orders placed exactly 7 days ago (heading: Week Ago). For example, if today is 10-JAN-2020, show the number of orders placed on 03-JAN-2020.
d) The number of orders placed the day before (heading: Previous Day). For Jan 1, display orders from Dec 31.
e) The number of orders placed the day after (heading: Next Day). For Jan 31, display orders from Feb 01.
f) The change in number of orders between 7 days ago and the current day. This would be the orders taken on the current day minus the number of orders placed 7 days ago (heading: Weekly Change).
g) The daily change between the day before and on that day. This is the number of orders placed on the current day minus the number of orders placed 1 day ago (heading: Daily Change).
h) Add a row that shows the word `Total` in the first column and the total of orders (for Orders Today) in the second column (and NULL values in other columns)
i) Sort the results by date (earliest first; the total row should be at the end).

Notes: If no orders exist for a day please show: 0 in that cell.

Example (order data may not match your output; this is purely for illustration):

OrderDate	Orders Today	Week Ago	Previous Day	Next Day	Weekly Change	Daily Change
01-JAN-2020	0	3	2	5	-3	-2
02-JAN-2020	5	2	0	6	3	5
…						
31-JAN -2020	1	2	1	0	-1	0
Total	25	NULL	NULL	NULL	NULL	NULL

Hint: Use LEAD() and LAG() functions.

Homework FAQ

Question: Where can I run and test homework queries?
Answer: The queries for your homework are to be executed using your Oracle accounts with SQL Developer

Question: Where can I find the dataset to answer the questions?
Answer: The queries are based on the ***CORP*** database (owner is CORP). It is recommended that you get familiar with both the structure and the contents of the database to help you better understand the query requirements and validate your answers. As a quick recap, the "describe <table-name>" command lets you see the structure of tables (this is also posted on the website). Doing a "SELECT * FROM <owner>.<table-name>;" will let you see all the data in the table.

Question: Can I get partial credit if the solution is not complete?
Answer: Yes, partial credit is available for solutions, so please build your SQL queries in stages.

Question: What is expected in the textual explanation of my answer?
Answer: The textual explanation of your queries consists of a natural language (i.e., English, not SQL) description of how you solved the query; including an explanation of how you chose to write the WHERE / GROUP BY / HAVING clauses. A clearly specified approach will be considered when awarding points. If you aren't able to explain your query, it will cost you points since an understanding of the query was not demonstrated. See the sample homework for some examples.

Question: Can I create temporary tables to help me answer queries?
Answer: You can use the WITH clause or inline SELECTs. However, you should NOT run DDL statements to create tables / views to answer your queries (in other words, your solution should be in the form of a single query statement that is terminated with a semi-colon).

Question: Do I lose points if I do not have the most efficient solution?
Answer: We care about the correct logic used (not just the correct answer).

Question: Can I compute some results and hard code them into the answers?
Answer: Do not hard code any computed values into your query.

Question: Can I add extra formatting for readability?
Answer: You may add extra formatting to your output for readability (but do include formatting we ask for). You may need to add extra formatting to get the output in an easy to read format (please try and ensure that a single row of output fits in a single line of the screen).

Question: What submission file formats are acceptable?
Answer: Please use an ASCII text file (.txt) only. Do not use other formats like Word or PDF.
Reason: we need to copy and paste and run your code (and word/pdf sometimes adds special characters for line breaks, smart-quotes, formatting, etc. that prevent your code from being executed without errors)

Question: Can I help someone with their homework, e.g., show them my code or see their code?
Answer: No.
However, you can talk to the instructor or a TA (you can show us your code to get help)

Question: Can I help someone understand concepts?
Answer: Yes, if someone is having trouble with subqueries (for example), you can use examples we covered in class or in the practice problems to help them understand. Please do NOT use the homework problem as an example. Do NOT show anyone your solution. Please do not see someone else's solution either (e.g., to debug a problem). Solutions may be shown to the TAs or the instructor. However, we will avoid telling you "is this correct" or "wrong". Instead, you will need to be specific in terms of "here's a problem I have, e.g., this code provides the following error when run."

Question: Can I suggest that someone use a specific pattern (e.g., "try a subquery for that question")
Answer: Yes, if all you say is, "Try a GROUP BY for that query." (or: HAVING / OUTER JOIN / subquery / WITH clause, etc.). Please do NOT share any details of the pattern (e.g., what elements or code should be included in the subquery, clause, etc.).

If you have any questions, please email for clarifications (please do not assume).