

## DA 6223 Exercise 3

Upload your project under Exercise 3 and answer the questions on Canvas.

### Problem 1

#### Creating a New Table in the Filter and Sort Task with a Basic Filter

Use the Filter and Sort task to create a new table with San Diego employees sorted in postal code order.

- Create a new project. Add a new process flow and rename it as Problem 1, 2, 3 and 4. Assign ORION Library (use a task or a program).
- Open the **employee\_addresses** data.
- Using the Filter and Sort task, create a new table for the San Diego office manager. This table should include **Employee\_ID, Employee\_Name, Street\_Number, Street\_Name, and Postal\_Code**.
  - Create a filter to include only employees from San Diego in the output table.
  - Order the output table in ascending postal code order.
  - Name the task and output table **SanDiegoEmployees**.
  - Submit the task to create the new table.
  - Check the results.
- How many employees live in San Diego?

### Problem 2

#### Creating a New Table in the Query Builder with a Basic Filter

Use the Query Builder to create a new table that includes all employees with the word Sales in their job titles.

- Add the **employee\_organization** data set.
- Use the Query Builder to create a query named **Sales Employees Query** and a table named **sales\_emps**. Include all employees that contain the word Sales as part of their job titles (Hint: When you create the filter, remember that Sales is case sensitive).
  - Which one of the following operators is suitable for writing this filter? **Equal to, Contains, Between or In a list.**
- Include all columns and sort the resulting table by **Department**.
- Run the query and verify the results.

### Problem 3

#### Using a Compound Filter in the Query Builder to Create a Table

Use the Query Builder to create a table that includes all employees with the word Chief or Manager in their job titles. Use the **employee\_master** dataset.

- a. Use the Query Builder to create a query named Offsite Meeting Query and a table named **meeting\_emps**.
  - Include these columns: **Employee\_ID, Employee\_Name, Department, and Job\_Title**.
  - Filter the data to keep rows where the job title contains the word **Chief** or **Manager**. Note: When entering values, remember that Chief and Manager are case sensitive.
  - Order the output table by ascending Department and then Employee\_ID.
- b. Run the query and answer the following question:
- c. How many rows are in the new **meeting\_emps** table?

END OF EXERCISE 3!

\*\*\*\*\*

THE REST IS OPTIONAL BECAUSE WE DID NOT COVER THE MATERIAL IN WEEK #4. COME BACK TO THESE QUESTIONS LATER AND SOLVE THEM.

## Problem 4 - OPTIONAL

### Assigning Values Conditionally

Use PROC SQL to create a report that will print the marital status of each employee in the Employee\_Payroll table. Use the CASE expression to recode the Marital\_Status column conditionally.

- a. Add a new program and write a PROC SQL statement as follows:
- b. Select all columns except Marital\_Status.
- c. Create a new column named Marital\_Status\_new. If the value of Marital\_Status is "S", Marital\_Status\_new column should take the value of "Single", if Marital\_Status is "M", then Marital\_Status\_new should be "Married". For all other values, Marital\_Status\_new should be "Other".
- d. Run the query and view the results.

## Problem 5 - OPTIONAL

### Calculating the Elapsed Time Intervals with the INTCK Function

Use a PROC SQL step to analyze shipping methods. Orion wants to calculate the number of weekdays between the order date and the delivery date for each order. The INTCK function can be used to count the number of various time intervals between a start date and an end date.

- a. Create a new Process Flow and rename it Problem 4. Add the **order\_fact** table.
- b. Add the Customer\_ID, Order\_Date, Delivery\_Date, and Product\_ID columns.
- c. Use the INTCK function to create a new column named **WeekdaysToDeliver**. The new column values are the number of weekdays between Order\_Date and Delivery\_Date.
- d. Exclude all retail purchases (Order\_Type = 1) because the items that are purchases in the store do not require shipping.
- e. Order the results so that the orders with the longest delivery times are at the top of the data set.
- f. Run the code, verify the results.

What was the longest number of weekdays between Order\_Date and Delivery\_Date?