Print

Practice Challenge: Joining Data Using a Non-Equijoin Condition

In this practice, you use the Query Builder to join data from two sources where the join condition involves a non-equijoin condition.

Two columns in the **bonus_schedule** table, **Employed_After** and **Employed_Before**, define the time range for a given bonus percentage. To find the correct bonus percentage for each employee:

- Join the **Hire_Date** column from the **employee_master** table with the **Employed_After** column using the greater-than-or-equal-to operator.
- Set a second join with the **Hire_Date** and **Employed_Before** columns with the less-than-or-equal-to operator.

1. If necessary, in the **Lesson4** project, add the **employee_master** and **bonus_schedule** tables to the Practices process flow.

    - Select **File > Open** and navigate to the course data location.
    - Select **employee_master > Open**. The data appears on a new tab in the work area.
    - Select **File > Open** and navigate to the course data location.
    - Select **bonus_schedule > Open**. The data appears on a new tab in the work area.

2. Because there are no matching columns between these two tables, a manual join is required. Join the tables so that **Hire_Date** is on or after the **Employed_After** column and is on or before the **Employed_Before** column.

    - In the process flow, hold down the Ctrl key and select the **employee_master** and **bonus_schedule** tables.
        - Right-click one of the tables and select **Query builder**. A message appears and indicates that Enterprise Guide cannot automatically join the tables by common columns.
        - Click **OK**.
        - In the Tables and Joins window, select **Hire_Date** in the **employee_master** table.
        - Select **Hire_Date** a second time, and drag the cursor to connect with **Employed_After** in the **bonus_schedule** table.
        - The Join Properties window appears automatically. Change the join condition to **t1.Hire_Date >= t2.Employed_After**.
        - Click **OK**.
        - Select **Hire_Date** again in the **employee_master** table.
        - Select **Hire_Date** a second time and drag the cursor to connect with **Employed_Before** in the **bonus_schedule** table.
        - The Join Properties window appears automatically. Change the join condition to **t1.Hire_Date <= t2.Employed_Before**.
        - Click **OK**.
        - Click **Close**.

3. Name the query **Employees Bonuses Query** and the table **bonuses**. Include the **Employee_ID**, **Employee_Name**, **Hire_Date**, **Salary**, and **Bonus_Percent** columns.

    - Enter **Employees Bonuses Query** in the Query name field.
    - Click **Change** and enter **bonuses** in the File name field.
    - Click **Save**.
    - On the Select Data tab, double-click the following columns to select them: **Employee_ID**, **Employee_Name**, **Hire_Date**, **Salary**, and **Bonus_Percent**.

4. Create a new column named **Bonus_Amount** that multiplies **Salary** by the percent value in the **Bonus_Percent** column. Display **Bonus_Amount** with a dollar sign, comma, and two decimal

places.

- Click the **Add A New Computed Column** icon on the Select Data tab
    - In Step 1, select **Advanced expression > Next**.
    - In Step 2, type or click to create the following expression:
      ```
      t1.Salary * t2.Bonus_Percent
      ```
    - Click **Next**.
    - In Step 3, enter **Bonus_Amount** in the Column Name field.
    - Click **Change** next to the Format field.
    - Select **Currency** from the Categories pane and **DOLLARw.d** from the Formats pane.
    - Change the overall width to **12** and the decimal places to **2**.
    - Click **OK**.
    - Click **Next > Finish**.

5. Run the query. What are the **Bonus_Percent** and **Bonus_Amount** values for Riu Horsey?

   Click **Run** to execute the query. Riu's **Bonus_Percent** is **2.20%** and his **Bonus_Amount** is *$738.93*.

6. Close all of the tabs except for the process flow and save the **Lesson4** project.

Hide Solution