IST/DEI 2024/2025

Data Analysis and Integration

1st semester

Lab 3: Introduction to ETL tools

Running a simple query

1. Open a terminal and connect to the local MySQL server: **mysql -u aid -p**
   Password: **aid**

2. On the MySQL prompt, execute the following command to connect to the database:
   use employees

3. Take a moment to inspect the contents of this view:
   select * **from** curr_salaries **limit** 10;

4. Execute the following query:
   select emp_no, salary
   from curr_salaries
   where salary > 80000
   limit 10;

5. Leave the terminal open so that you can check these results later on.

Creating a new transformation

6. Open a new terminal and navigate to the folder: **~/Pentaho/data-integration**

7. Start Pentaho Data Integration (PDI) with: **./spoon.sh**

8. In the **File** menu, select **New > Transformation**.

Creating a database connection

9. In the left pane, switch from the **Design** to the **View** tab.

10. Right-click **Database connections** and select **New**.

11. In the **Database Connection** dialog, specify the following:

Connection Name:	**employees**
Connection Type:	**MySQL**
Access:	**Native (JDBC)**
Host Name:	**localhost**
Database Name:	**employees**
Port Number:	**3306**
User Name:	**aid**
Password:	**aid**

12. Press **Test** to test the database connection. A new dialog should say that the connection is OK.

13. Close the **Database Connection** dialog with **OK**.

14. In the **View** tab, expand **Database Connections**, right-click **employees** and select **Share**.
    Note: This will make the database connection available to other transformations as well.

Adding a table input step

15. In the left pane, switch to the **Design** tab.

16. Expand **Input**, and drag a **Table input** step to the canvas.
    Note: You can also find the step by searching for it in the text box at the top of the Design tab.

17. Double-click the **Table input** to configure it.

18. In **Connection**, choose the **employees** database connection.

19. Press the **Get SQL select statement** button.

20. In the **Database Explorer**, expand **employees**, **Tables** and **Views**.

21. Select the **curr_salaries** view, and press **OK**.

22. In the question dialog **Do you want to include the field-names in the SQL?** answer **Yes**.

23. Check if the SQL statement is correct and close the **Table input** configuration with **OK**.

24. Right-click the **Table input** step and select **Preview**.

25. In the **Transformation debug dialog**, press **Quick Launch**.

26. The **Examine preview data** window will appear with the output from the **Table input** step.

27. Check that the results agree with what you have obtained earlier when querying the database.

28. **Close** the window, and **Close** the **Select the preview step** window.

Adding a filter rows step

29. In the **Design** tab, expand **Flow**.

30. Drag a **Filter rows** step to the canvas.

31. Hold the **shift** key, and drag from the **Table input** to the **Filter rows** to create a hop.

32. Double click the **Filter rows** step to configure it.

33. Specify **The condition** as follows:
    - Click on the leftmost **<field>**, and select **salary**.
    - Click the equal sign (**=**) in the middle, and replace it with the **>** sign.
    - Click on the rightmost **<value>**, and write **80000** in **Value**.

34. Press **OK** to close the **Filter rows** configuration.

35. Right-click the **Filter rows** step and select **Preview**.

36. In the **Transformation debug dialog**, press **Quick Launch**.

37. The **Examine preview data** window will appear with the output from the **Filter rows** step.

38. Check that the results agree with what you have obtained earlier when querying the database.

39. **Close** the window, and **Close** the **Select the preview step** window.

Adding a text file output step

40. In the **Design** tab, expand **Output**.

41. Drag a **Text file output** step to the canvas.

42. Hold the **shift** key, and drag from the **Filter rows** to the **Text file output** to create a hop.

43. When the popup menu appears, select **Result is TRUE**.

44. Double click the **Text file output** step to configure it.

45. In the **File** tab, do the following:
    - In **Filename**, write **/home/aid/Downloads/salaries** (if you are on the VM)
    - Uncheck **Create Parent folder**

- Change the **Extension** from **txt** to **csv**
- Press the button **Show filenames** to check the full path to the file that will be created.

46. In the **Content** tab:
    - Check that the **Separator** is a semicolon (**;**)
    - Make sure that the option **Header** is checked.

47. In the **Fields** tab:
    - Press the **Get Fields** button.
    - Then press the **Minimal width** button.

48. Close the **Text file output** configuration with **OK**.

Saving and running the transformation

49. In the **File** menu, select **Save As…**

50. Navigate to **/home/aid/Downloads** and save the transformation as **salaries.ktr**

51. In the **Action** menu, select **Run** (or press the **Run** button in the toolbar).

52. In the **Run Options** dialog, press **Run**.

53. In the **Step Metrics** tab at the bottom, check that the **Text file output** has produced 83 rows as output. (Why 83 and not 82?)

54. Go to the folder where the **salaries.csv** file is located (/home/aid/Downloads).

55. Open the **salaries.csv** file in a text editor, and check its contents.

56. Open the **salaries.csv** file with **LibreOffice Calc**.

57. Indicate that the **separator** is a **Semicolon** (as specified earlier in the **Text file output** step configuration).

Running another query

58. Go back to the terminal where you have the **mysql** command prompt.

59. Execute the following query to obtain the number of employees by department, but only for departments with at least 40 employees:

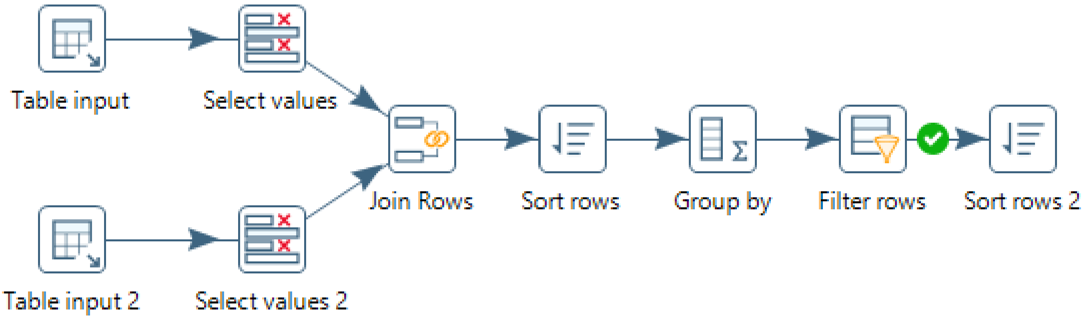
```
select b.dept_no, b.dept_name, count(emp_no) as count_emp_no
from curr_dept_emp as a, departments as b
where a.dept_no = b.dept_no
```

```
group by b.dept_no, b.dept_name
having count_emp_no >= 40
order by count_emp_no desc;
```

60. Leave the terminal open so that you can check these results later on.

Implementing the query as a transformation

The query above will be implemented as a transformation that looks like the figure below. The following steps will guide you through building this transformation.



61. In Pentaho Data Integration, create a new transformation.

62. Add a **Table input** step, and configure it to read the **curr_dept_emp** view.

63. Add a **Table input 2** step, and configure it to read the **departments** table.

64. **Preview** both steps to make sure that they are working correctly.

65. In the **Design** pane, expand **Transform** and drag two **Select values** steps to the canvas.

66. Connect **Table input** to **Select values**, and **Table input 2** to **Select values 2**.

67. Configure **Select values** as follows:
    - In the **Select & Alter** tab, press **Get fields to select**
    - Next to **dept_no**, write **dept_no_1** in the second column (**Rename to**)

68. Configure **Select values 2** as follows:
    - In the **Select & Alter** tab, press **Get fields to select**
    - Next to **dept_no**, write **dept_no_2** in the second column (**Rename to**)

69. **Preview** both steps to make sure that the **dept_no** fields are being renamed as intended.

70. In the **Design** pane, expand **Joins** and drag a **Join Rows (cartesian product)** step to the canvas.

71. Connect the **Select values** step to the **Join Rows** step. When a popup menu appears, choose **Main output of step**.

72. Connect the **Select values 2** step to the same **Join Rows** step. Again, choose **Main output of step**.

73. Configure the **Join Rows** step as follows:
    - Change its name to simply **Join Rows** without (cartesian product)
    - Specify **The condition** as follows:
        - Click on the leftmost **<field>**, and select **dept_no_1**.
        - Leave the equal sign (**=**) in the middle.
        - Click on the rightmost **<field>**, and select **dept_no_2**.

74. **Preview** the **Join Rows** step to make sure that it is working as intended.

75. In the **Design** pane, expand **Transform** and drag a **Sort rows** step to the canvas.

76. Connect the **Join Rows** step to the **Sort rows**.

77. Configure the **Sort rows** as follows:
    - In the first line of **Fields**, select as **Fieldname**: **dept_no_1**
    - In the second line of **Fields**, select as **Fieldname**: **dept_name**

74. **Preview** the **Sort rows** step to make sure that it is sorting the rows as intended.

78. In the **Design** pane, expand **Statistics** and drag a **Group by** step to the canvas.

79. Connect the **Sort rows** step to the **Group by** step.

80. Configure the **Group by** step as follows:
    - In **Group fields**, select **dept_no_1** in the first line and **dept_name** in the second line
    - In **Aggregates**, use only the first line:
        - Name: **count_emp_no**
        - Subject: **emp_no**
        - Type: **Number of Values (N)**

81. A **Notice** dialog will appear with the message: *If the incoming data is not sorted on the specified keys, the output results may not be correct. We recommend sorting the incoming data within the transformation.* (This is why we included a **Sort rows** step before the **Group by** step.)

82. **Preview** the **Group by** step to make sure that it is working as intended.

83. In the **Design** pane, expand **Flow** and drag a **Filter rows** step to the canvas.

84. Connect the **Group by** step to the **Filter rows** step.

85. Configure **The condition** of the **Filter rows** step as follows:
    - Click on the leftmost **<field>**, and select **count_emp_no**
    - Click the equal sign (**=**) in the middle, and replace it with the **>=** sign
    - Click on the rightmost **<value>**, and write **40** in **Value**.

86. **Preview** the **Filter rows** step to make sure that it is filtering the rows as intended.

87. In the **Design** pane, expand **Transform** and drag a **Sort rows 2** step to the canvas.

88. Connect the **Filter rows** step to the **Sort rows 2** step. When a popup menu appears, choose **Result is TRUE**.

89. Configure the **Sort rows 2** step as follows:
    - In the first line of **Fields**, select as **Fieldname**: **count_emp_no**
    - In the second column (**Ascending**), select **N**

90. **Preview** the **Sort rows 2** step to make sure that it is working correctly.

91. Compare the results with what you had obtained earlier when running the query on **mysql**.

Exercise

92. The following query obtains the sum of salaries by department:

```
select b.dept_no, c.dept_name, sum(a.salary) as sum_salary
from curr_salaries as a, curr_dept_emp as b, departments as c
where a.emp_no = b.emp_no and b.dept_no = c.dept_no
group by b.dept_no, c.dept_name
order by sum_salary desc;
```

📷 Implement this query as a transformation in Pentaho Data Integration.