Data Analysis and Integration

Lab 7: Data profiling

Creating the steelwheels database

1. Download the file **steelwheels.sql**.

2. Take a moment to inspect the contents of the **steelwheels.sql** script.
   - Locate the CREATE DATABASE statement.
   - Locate all CREATE TABLE statements.
   - Check the columns and data types for each table.
   - Check the primary and foreign keys for each table.
   - Locate all INSERT instructions to load data into these tables.

3. Open a terminal and navigate to the folder where the **steelwheels.sql** script is located.

4. Execute the following command to login to the local MySQL server: **mysql -u aid -p**
   Password: **aid**

5. On the MySQL prompt, execute the following command to create the database:
   source steelwheels.sql

6. Execute the following command to change to the steelwheels database:
   use steelwheels

Inspecting the steelwheels database

7. Open **MySQL Workbench**.

8. Next to **MySQL Connections**, press the plus (+) button.

9. In the **Setup New Connection** dialog, configure the new connection as follows:
   - Connection Name: **steelwheels**
   - Hostname: **localhost**
   - Port: **3306**
   - Username: **aid**
   - Password: (Store in Keychain) **aid**
   - Default Schema: **steelwheels**

10. Press **Test Connection** to test the new connection. Then close both windows.

11. Click on the **steelwheels** database connection to open it.

12. In the left pane, check that the **steelwheels** database is selected (in bold).
    Note: If it is not in bold, double-click to select it.

13. Expand **steelwheels > Tables** to show the tables in the database.

14. Right-click the **offices** table and choose **Select Rows**.
    - Check the information that is being stored about each office.

15. Right-click the **employees** table and choose **Select Rows**.
    - Check the information that is being stored about each employee.
    - Note that **OFFICECODE** and **REPORTSTO** are foreign keys.

16. Right-click the **products** table and choose **Select Rows**.
    - What kind of products does this company sell?
    - Also, check the columns **QUANTITYINSTOCK**, **BUYPRICE** and **MSRP**.

    Note: MSRP is the manufacturer's suggested retail price.

17. Right-click the **customers** table and choose **Select Rows**.
    - Check the information that is being stored about each customer.
    - In particular, check the columns with **NULL** values.
    - Note that **SALESREPEMPLOYEENUMBER** is a foreign key.

18. Right-click the **orders** table and choose **Select Rows**.
    - Check the information that is being stored about each order.
    - Note that **CUSTOMERNUMBER** is a foreign key.

19. Right-click the **orderdetails** table and choose **Select Rows**.
    - This table contains the products that are included in each order.
    - **ORDERLINENUMBER** indicates the position in which the product appears in the order.
    - Note that **ORDERNUMBER** and **PRODUCTCODE** are foreign keys.
    - Note that **PRICEEACH** is the unit price (it must be multiplied by **QUANTITYORDERED**).

Data profiling with DataCleaner

DataCleaner is a tool for data profiling and data quality analysis which includes features to check the completeness of data, to perform data cleaning and standardization, and to detect and eliminate approximate duplicates (this feature is available in the commercial version only). In this lab we will use DataCleaner (community version) to gain a better understanding of the steelwheels database and the data contained therein.

20. Open a new terminal and navigate to the folder: **~/Pentaho/DataCleaner**

21. Start DataCleaner with: **./datacleaner.sh**

22. Once DataCleaner starts, click on **Manage datastores**. The **Datastore Management** window will open.

23. At the bottom of the window, click on the **MySQL** icon to register a new datastore.

24. In the **Database connection** dialog, configure the new datastore as follows:

• Datastore name:	**steelwheels**
• Hostname:	**localhost**
• Port:	**3306**
• Database:	**steelwheels**
• Username:	**aid**
• Password:	**aid**

25. Click **Test connections** to check that the connection is correctly configured.

26. Close the dialog by pressing **Register datastore**.

27. Select the **steelwheels** datastore and press **Build job**.

Analyzing the customers table

28. On the left pane, expand **steelwheels** to get the list of tables.

29. Drag the **customers** table to the right pane.

30. On the left pane, expand **Library > Analyze** and drag **Completeness analyzer** to the right pane.

31. In the right pane, right-click on **customers** and select **Link to**. Then click on **Completeness analyzer**.

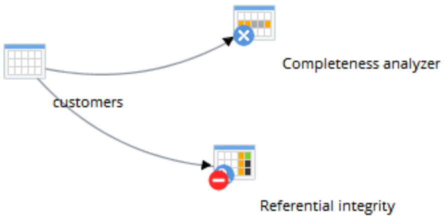
32. The two steps will be connected by an arrow, and the **Completeness analyzer** dialog will open.

33. We want to check the completeness of every field, so leave everything checked and press **Close**.

34. At the top-right corner of the main window, click **Execute**.

35. The **Analysis results** window will appear with the results of **Completeness analyzer**. From these results, we can conclude that:
    - Most records do not use the **ADDRESSLINE2** field.
    - Many records do not use the **STATE** field.
    - Some records have no **SALESREPEMPLOYEENUMBER**.

36. Close the **Analysis results** window.



37. From the left pane, drag **Referential integrity** to the right pane.

38. Right-click the **customers** table and select **Link to**. Then click on **Referential integrity**.

39. The two steps will be connected by an arrow, and the **Referential integrity** dialog will open.

40. We want to check the referential integrity of **SALESREPEMPLOYEENUMBER**, so select that column.

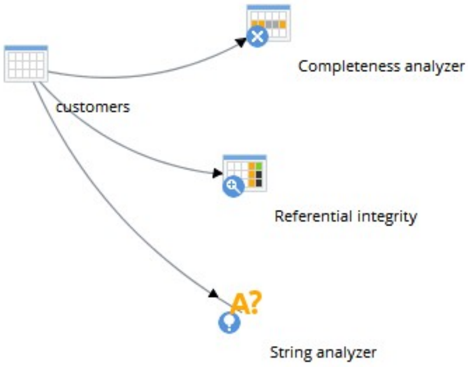
41. In the **Required properties** below, select:
    - Datastore: **steelwheels**
    - Schema name: **steelwheels**
    - Table name: **employees**
    - Column name: **EMPLOYEENUMBER**

42. Close the **Referential integrity** dialog and **Execute** the job.

43. When the **Analysis results** window opens, select **Referential Integrity** on the left pane.

44. The result should be: **Records with unresolved foreign key values (0)**. From this we can conclude that there are no unresolved values, i.e. every **SALESREPEMPLOYEENUMBER** in the customers table exists as an **EMPLOYEENUMBER** in the employees table.

45. Close the **Analysis results** window.



46. From the left pane, drag **String analyzer** to the right pane.

47. Right-click the **customers** table and select **Link to**. Then click on **String analyzer**.

48. When the **String analyzer** window opens, leave everything selected, and click **Close**.

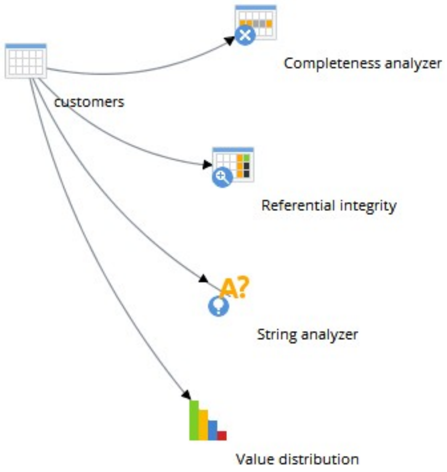
49. **Execute** the job.

50. When the **Analysis results** window opens, select **String analyzer** on the left side.

51. Each row in the results gives you some useful information about the values in each column of the customers table. For example, try to answer the following questions:

- Which column has the longest string?
- Which columns have strings with multiple words rather than a single word?
- Is there any column whose values are written entirely in uppercase?
- Which columns have mostly digits when comparing to the total char count?

52. Close the **Analysis results** window.



53. From the left pane, drag **Value distribution** to the right pane.

54. Right-click the **customers** table and select **Link to**. Then click on **Value distribution**.

55. When the **Value distribution** dialog opens, uncheck everything and select **COUNTRY** only.

56. Close the **Value distribution** dialog, and **Execute** the job.

57. When the **Analysis results** window opens, select **Value distribution** on the left side.

58. Which country do most customers live in? Is **Portugal** in this chart? How many customers come from **Portugal**?

59. Click the small green arrow next to **Portugal**. You should get the records of those Portuguese customers.

60. Close the **Analysis results** window.

61. Close the analysis job and you will be back to the main window of DataCleaner.

Analyzing the results of a transformation

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

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

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

65. In the left pane, switch from the **Design** to the **View** tab, and expand **Transformations > Transformation 1 > Database connections**.

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

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

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

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

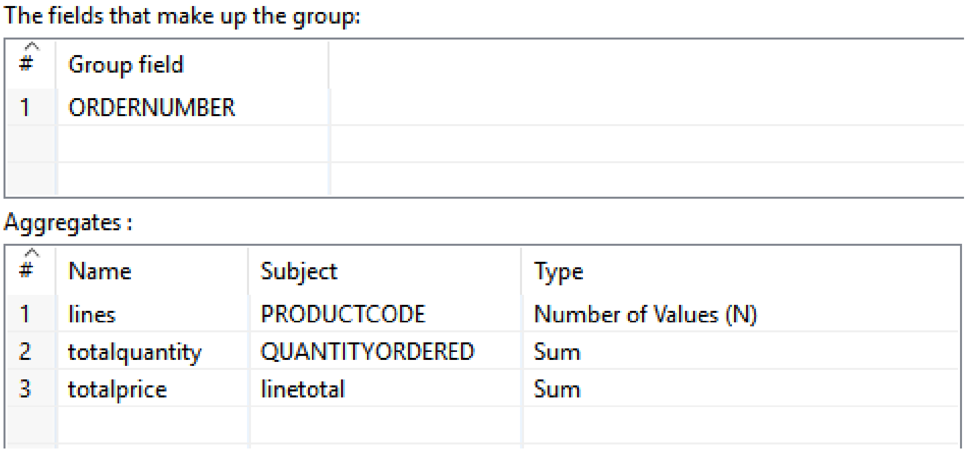
69. Close the **Database Connection** dialog with **OK**.

70. In the **View** tab, right-click the **steelwheels** database connection and select **Share**. This will make the database connection available to other transformations.

71. Use a **Table Input** step to read the **orderdetails** table.

72. Add a **Calculator** to compute a new field **totatotal** as **A*B** where **A** is **PRICEEACH** and **B** is **QUANTITYORDERED**.

73. Add a **Group by** step and configure it as in the following figure:



The fields that make up the group:

#	Group field
1	ORDERNUMBER

Aggregates :

#	Name	Subject	Type
1	lines	PRODUCTCODE	Number of Values (N)
2	totalquantity	QUANTITYORDERED	Sum
3	totalprice	linetotal	Sum

Note: In general, it is necessary to use a Sort rows step before a Group by step. However, in this case, the data is already sorted by ORDERNUMBER.

74. Do a **Preview** of the **Group by** step to check that these aggregates are being calculated correctly.

75. Add a **Text file output** step after the **Group By**, connect them, and configure it as follows:
File tab:

- In **Filename**, write **/home/aid/Downloads/ordertotals** (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.

Content tab:

- Check that the **Separator** is a semicolon (**;**)
- Make sure that the option **Header** is checked.

Fields tab:

- Press the **Get Fields** button.
- Then press the **Minimal width** button.

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

77. Save the transformation as **/home/aid/Downloads/ordertotals.ktr**

78. Run the transformation, and check the output file **/home/aid/Downloads/ordertotals.csv** has been created.

79. Go back to DataCleaner, click on **New** (top left) and then **Manage datastores**.

80. At the bottom of the window, click on the **CSV file** icon to register a new datastore.

81. In the **CSV file datastore** window, configure the datastore as follows:
    - In **Datastore name**, write **ordertotals**
    - In **Source**, browse to **/home/aid/Downloads/ordertotals.csv**

82. Close the dialog by pressing **Register datastore**.

83. Select the **ordertotals** datastore and press **Build job**.

84. On the left pane, expand **Downloads** and drag **ordertotals.csv** to the right pane.

85. Expand **Library > Transform > Conversion** and drag a **Convert to number** to the right pane.

86. Link **ordertotals.csv** to **Convert to number**.

87. In the **Convert to number** dialog, leave everything selected and press **Close**.

88. Expand **Library > Analyze** and drag a **Number analyzer** to the right pane.

89. Link **Convert to number** to the **Number analyzer**.

90. In the **Number analyzer** dialog, select everything except **ORDERNUMBER**.

91. **Execute** the analysis job.

92. Answer the following questions:
    - On average, how many lines has an order?
    - Is the smallest order (in terms of quantity) the one that has also the lowest total price? (You will have to click some green arrows to find this.)
    - How much is the total sales for this company?

93. Close the **Analysis results** window.

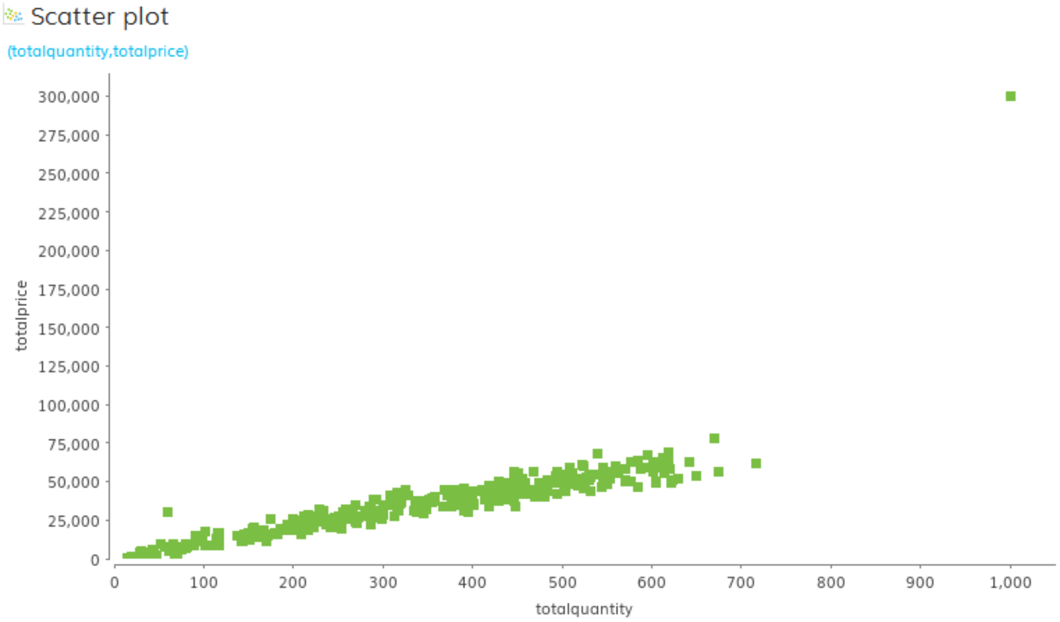
94. On the left pane, expand **Analyze > Visualization** and drag a **Scatter Plot** to the right pane.

95. Link the **Convert to number** to the **Scatter plot**, and:
    - For **Variable1**, select **totalquantity**
    - For **Variable2**, select **totalprice**

96. Close the **Scatter plot** dialog and **Execute** the analysis job.

97. In the **Analysis results** window, change to the **Scatter plot** results.

98. The scatter plot shows that, as expected, there is a correlation between the total quantity and the total price of an order.



99. However, there are at least two outliers that deviate from the trend. Can you spot them? Click on them to find their ORDERNUMBERs.