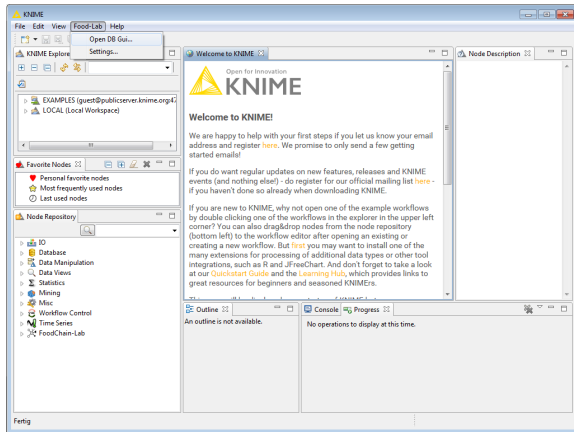
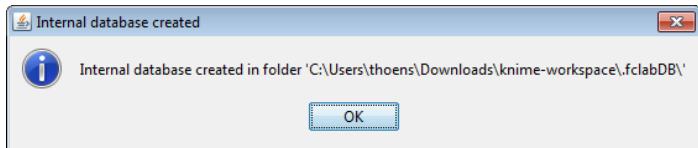


# Creating a Workflow in FoodChain-Lab 1

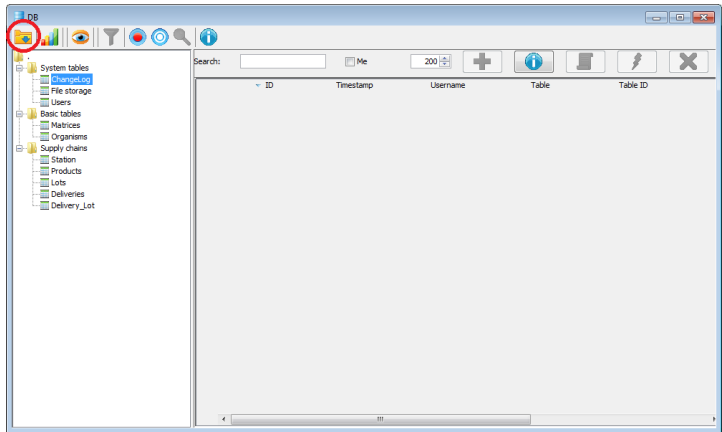
- Import the Example XLS template to FoodChain-Lab.
- You can download it from here: [https://github.com/SiLeBAT/BfROpenLabResources/raw/master/GitHubPages/documents/fake\\_data.xls](https://github.com/SiLeBAT/BfROpenLabResources/raw/master/GitHubPages/documents/fake_data.xls)
- Via the **Tracing** node assign weights of "1" to the supermarkets in Hamburg, Ingolstadt and Münster to mark them as outbreak locations.
- Use the **Tracing View** to look at the delivery network.



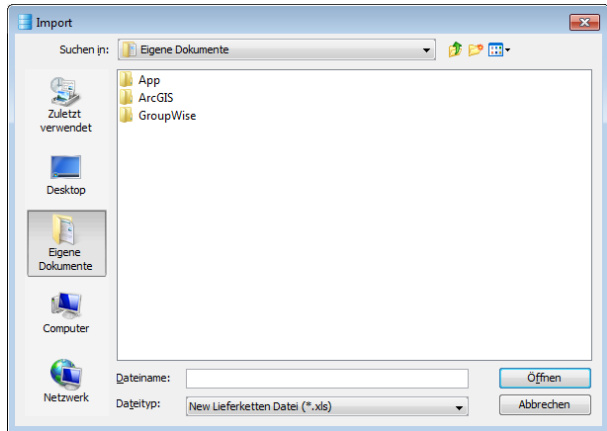
- Select **Food-Lab > Open DB Gui...** in the menu bar to open the database dialog.



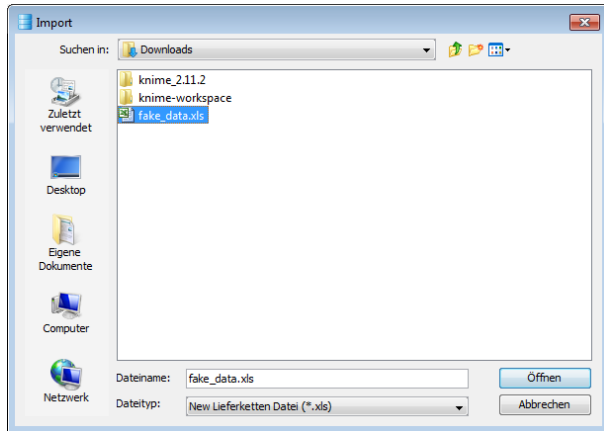
- If you get a message saying the internal database has been created, click **OK**.



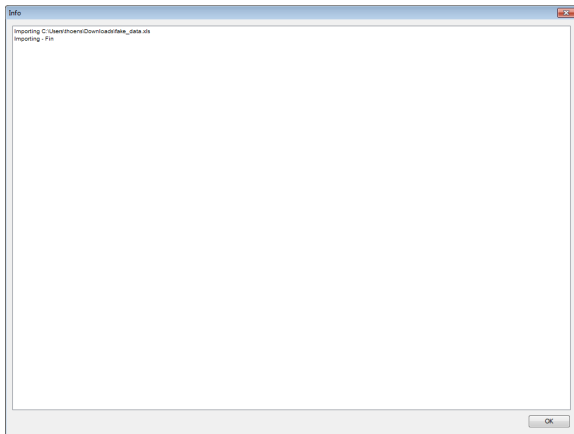
- In the database dialog click the **Table import** button in the upper left corner.



- Now a file dialog will pop up.
- \*.xls files in FoodChain-Lab format can be selected here.

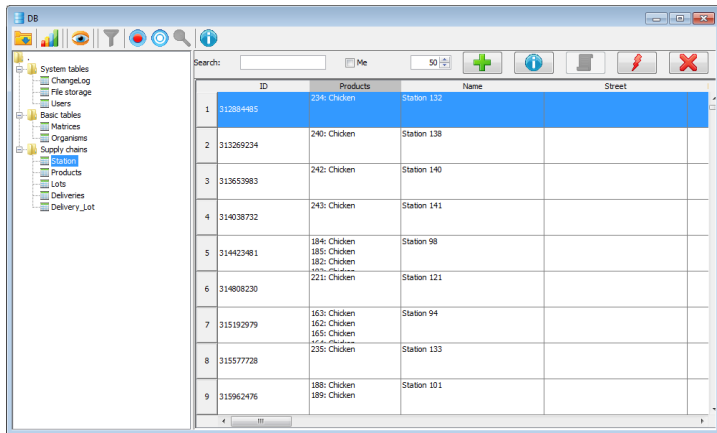


- Download the example file from [https://github.com/SiLeBAT/BfROpenLabResources/raw/master/GitHubPages/documents/fake\\_data.xls](https://github.com/SiLeBAT/BfROpenLabResources/raw/master/GitHubPages/documents/fake_data.xls).
- Select the file in the dialog and press **Open**.



- When the importing is finished you see a dialog with errors/warnings, that occurred in the import process.
- No errors occurred, so just press **OK**.





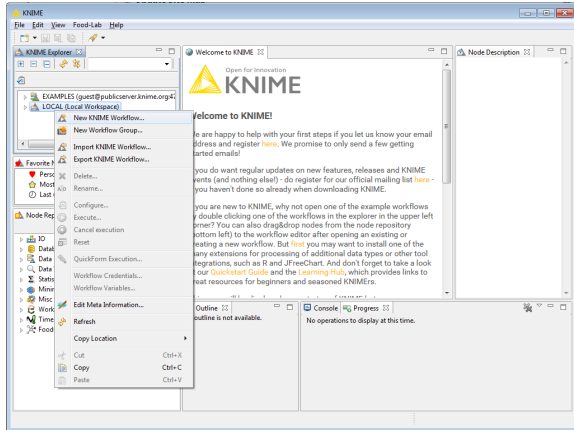
The screenshot shows a database application window titled 'DB'. On the left is a tree view of the database structure, including 'System tables', 'Basic tables', and 'Supply chains'. The 'Station' table is selected. The main area displays a table with 9 rows of data. The columns are 'ID', 'Products', 'Name', and 'Street'. The first row is highlighted in blue.

	ID	Products	Name	Street
1	312884485	234: Chicken	Station 132	
2	313269234	240: Chicken	Station 138	
3	313653983	242: Chicken	Station 140	
4	314038732	243: Chicken	Station 141	
5	314423481	184: Chicken 185: Chicken 182: Chicken 183: Chicken 221: Chicken	Station 98	
6	314808230		Station 121	
7	315192979	163: Chicken 162: Chicken 165: Chicken 164: Chicken	Station 94	
8	315577728	235: Chicken	Station 133	
9	315962476	188: Chicken 189: Chicken	Station 101	

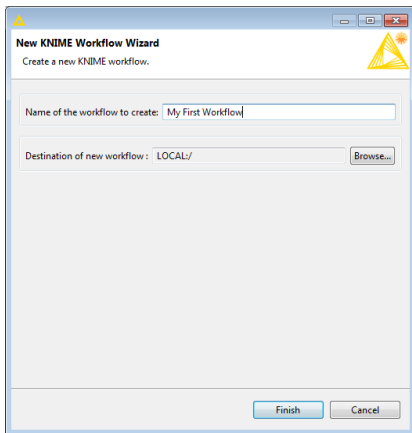
- In the database dialog, you can now look at the imported data and check the data for duplicates.
- Close the dialog.

Task

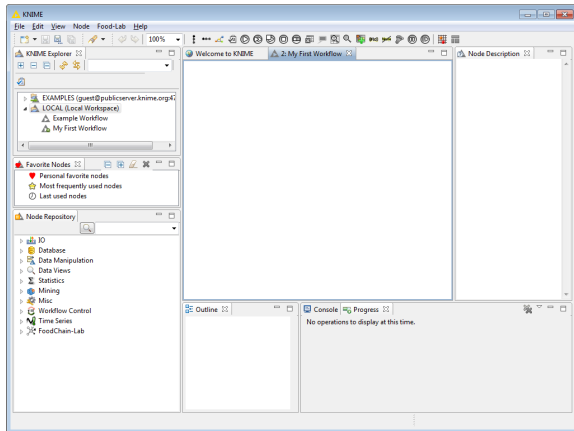
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23



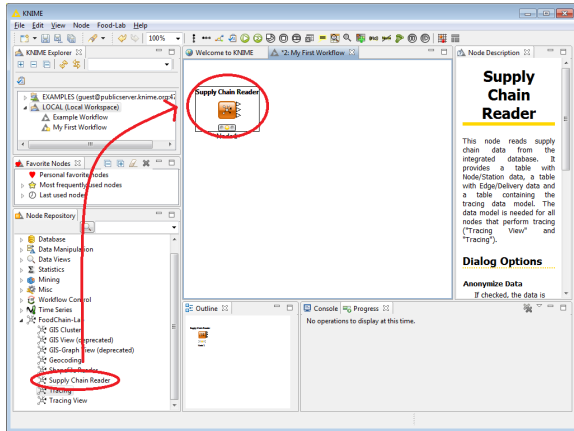
- Now we want to create a workflow, that uses the imported data.
- Right click on **LOCAL** in the **KNIME Explorer** and select **New KNIME Workflow...**



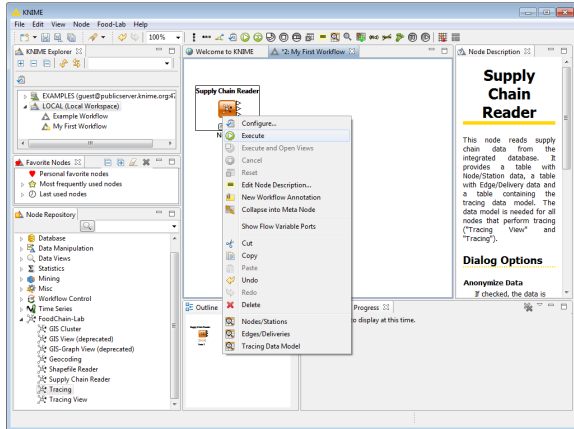
- In the dialog set the name of the workflow to "My First Workflow" and click **Finish**.



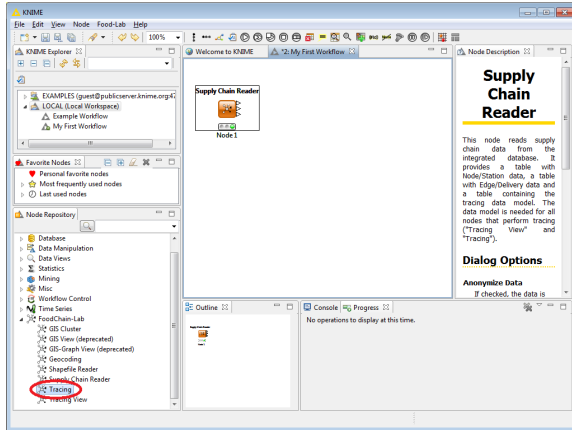
- The created workflow will show up in the editor in the center.



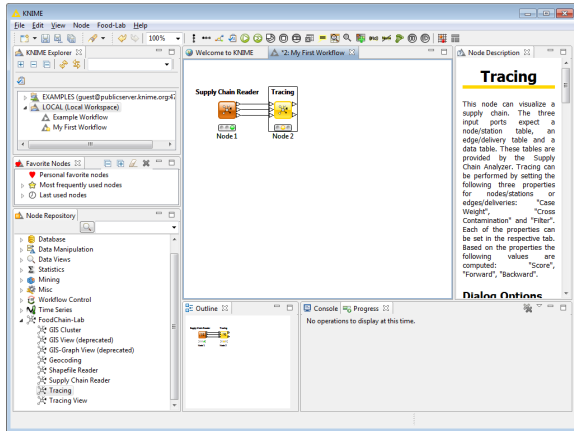
- Drag the **Supply Chain Reader** from the **Node Repository** to the workflow.



- We do not need to configure the **Supply Chain Reader**.
- Right click on it and select **Execute**.

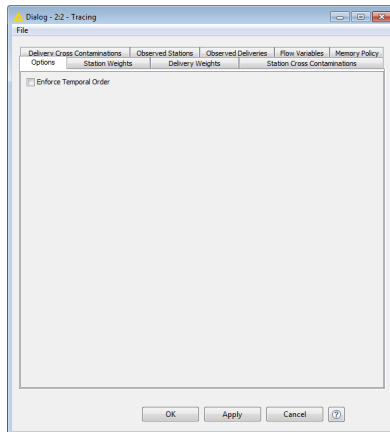


- The **Supply Chain Reader** has now read all data from the internal database.
- Select the **Supply Chain Reader** in the workflow (so that a rect is drawn around it) and double click on the **Tracing** node in the **Node Repository**.

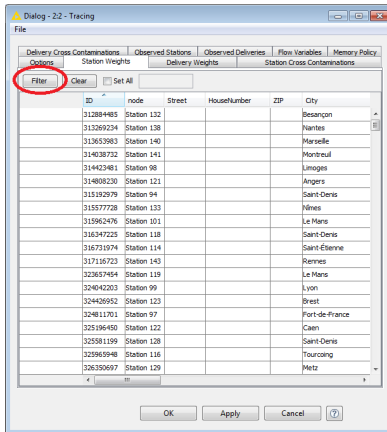


- The **Tracing** node should show up in the workflow and its three input ports should be automatically connected to the **Supply Chain Reader**.
- Double click on the **Tracing** node to configure it.





- You will notice several tabs for different parameters.
- "Weight" and "Cross Contamination" can be set for stations/deliveries. Based on these attributes a "Score" is computed for each station/delivery.
- Additionally you can set "Observed" stations/deliveries.
- Select the **Station Weights** tab.



- A table with all available stations will pop up.
- The weight can be set in the left column.
- Since scrolling through all stations is very inefficient, we can filter out all desired stations.
- Click on **Filter**.

Highlight Condition

Type: Logical Condition

Property	Operation	Value
ID	==	

Add Remove

Add

OK Cancel

- In this dialog you can specify which stations should appear in the table.

Highlight Condition

Type: Logical Condition

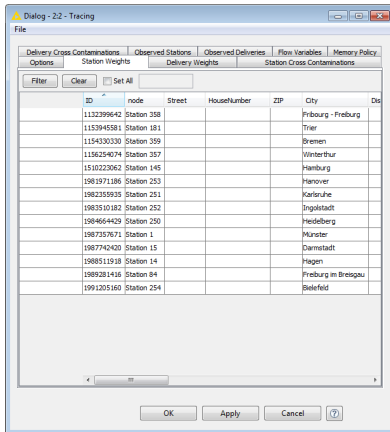
Property	Operation	Value
type of business	==	Supermarket

Buttons: Add, Remove, Add, OK, Cancel

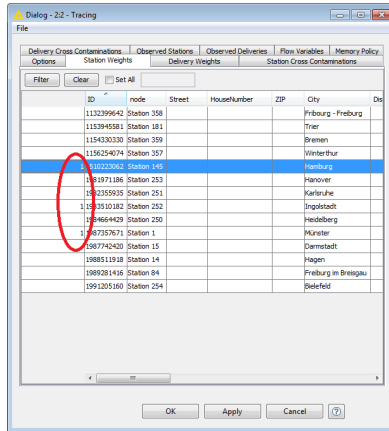
- We only want to specify weights for supermarkets, since that is where contaminated products were found.
- Set **Property** to "type of business" and **Value** to "Supermarket".
- Press **OK**.

Task

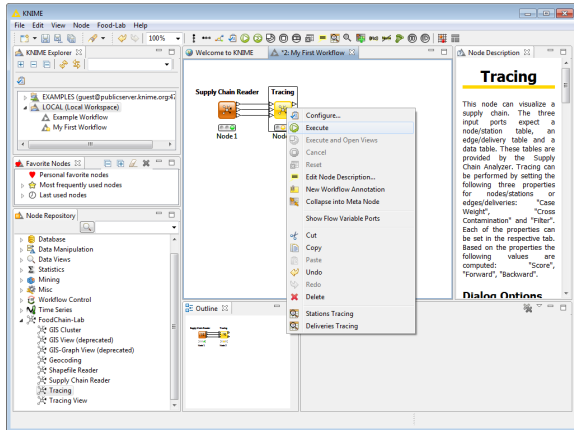
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23



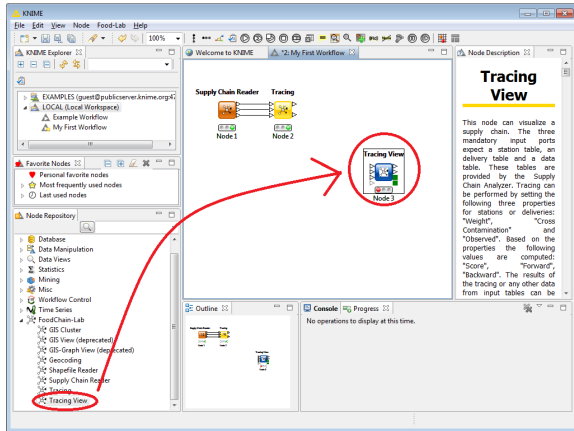
- Now you only see supermarkets in the dialog.



- Set a weight of "1" to the supermarkets in "Hamburg", "Ingolstadt" and "Münster" to indicate that contaminated products were found there.
- Click **OK** apply the settings and close the dialog.

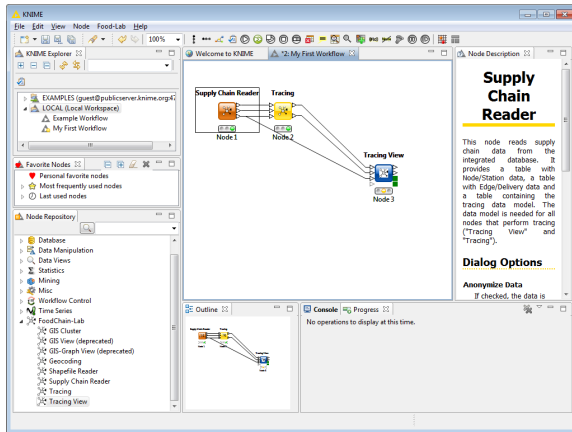


- Right click on the **Tracing** node and select **Execute** to execute the node.



- Drag the **Tracing View** from the **Node Repository** to the workflow.





- Connect the output ports of the **Tracing** node to the first two input ports of the **Tracing View**.
- Connect the third output port of the **Supply Chain Reader** to the third input port of the **Tracing View**.
- Now you open the **Tracing View** and analyze the data. This will be shown in the second part of this tutorial.