Data Mining: Guidelines for the Computer Labs

Contents

Introduction

Background Information

The **Data Mining course** is about OLAP (OnLine Analytical Processing), Data Warehouse and Data Mining technologies.

All the computer labs are held by means of **Personal Virtual Computer (PVC)** system. PVC installation instructions are available at PVC homepage: https://pvc.susu.ru/.

Computer labs are aimed to learning OLAP features of relational database management system (DBMS) and data mining tool. You will use the following free software within PVC:

- Oracle XE (eXpress Edition) DBMS as a database server and Oracle SQL Developer as a client program;
- KNIME Data Mining tool.

Doing a lab, **ask instructor to help** in case of any technical problem. Having done a lab, **ask instructor to verify results** of your lab.

Useful links

You may use these URLs to see or download content for self-study:

- Oracle DBMS
  - Documentation: Oracle XE Docs, Oracle SQL Developer Docs (PDF)
  - Downloads: Oracle XE, Download SQL Developer
- KNIME Data Mining tool
  - Product website: http://www.knime.org/
  - Documentation
    - Getting Started Guide
    - Demos and Tours
  - Downloads

Lab 1. Data Warehousing

OBJECTIVE. In this lab you will create a toy data warehouse and learn how to use SQL aggregation functions for OLAP purposes.

Activity 1. Create a Data Warehouse

1. Using SQL Developer create a database with the following structure:

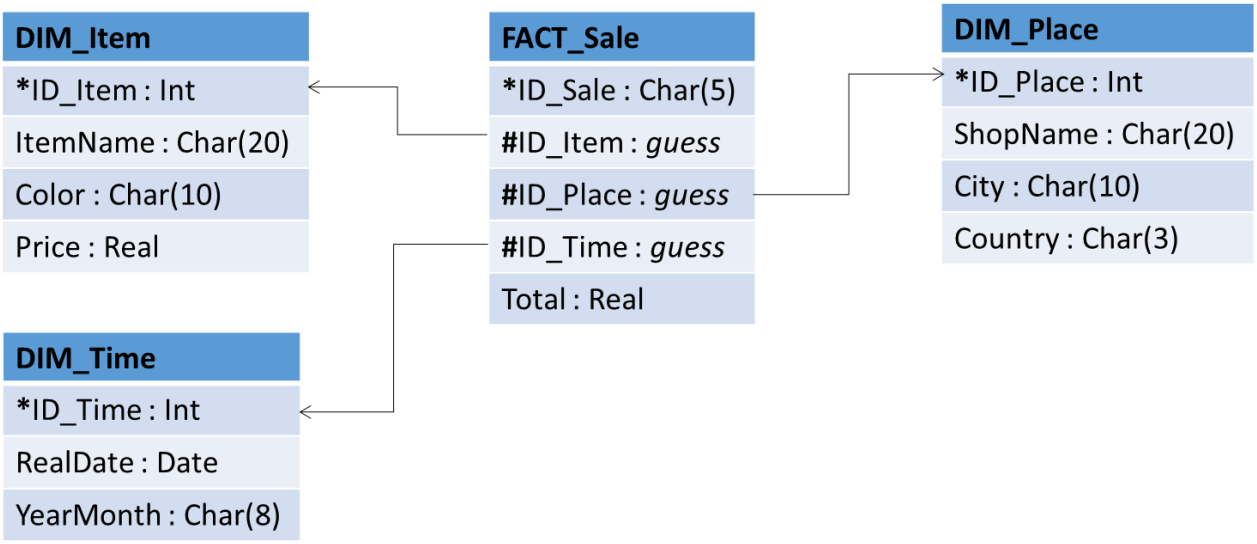

Fig. 1. Data Warehouse structure
(here sign * denotes primary key and sign # denotes foreign key)

2. Fill in the tables above with some real-like data (at least 5 records in each dimension table, at least 20 records in fact table). Fill the YearMonth field like '2014-FEB' for the respective value '01/02/2014' of the RealDate field.

ASK INSTRUCTOR to verify the results of your activity.

Activity 2. Aggregation with ROLLUP

ROLLUP enables a SELECT statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total.

The action of ROLLUP is straightforward: it creates subtotals that roll up from the most detailed level to a grand total, following a grouping list specified in the ROLLUP clause. ROLLUP takes as its argument an ordered list of grouping columns. First, it calculates the standard aggregate values specified in the GROUP BY clause. Then, it creates progressively higher-level subtotals, moving from right to left through the list of grouping columns. Finally, it creates a grand total. ROLLUP creates subtotals at n+1 levels, where n is the number of grouping columns.

An example:

```
SELECT Dim1, Dim2, SUM(Measure)
FROM FactTab
GROUP BY ROLLUP (Dim1, Dim2)
```

```
ORDER BY Dim1, Dim2
```

1. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the FACT_Sale.ID_Item and FACT_Sale.ID_Place fields. An example of the required result:

```
ID_ITEM   ID_PLACE     TOTAL
      1           1     4363.55
      1           2     4794.76
      1           3     4718.25
      1           4     5387.45
      1           5     5027.34
      1                24291.35
      2           1     5652.84
      2           2     4583.02
      2           3     5555.77
      2           4     5936.67
      2           5     4508.74
      2                26237.04
                       50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

2. Modify the query above to calculate subtotals across the Dim_Item.ItemName and Dim_Place.ShopName fields. An example of the required result:

```
ITEMNAME    SHOPNAME    TOTAL
   Bolt     MainShop    4363.55
   Bolt     Details     4794.76
   Bolt     Repair      4718.25
   Bolt     Tools4U     5387.45
   Bolt     HomeMaster  5027.34
   Bolt                24291.35
   Screw     MainShop   5652.84
   Screw     Details    4583.02
   Screw     Repair     5555.77
   Screw     Tools4U    5936.67
   Screw     HomeMaster 4508.74
   Screw                26237.04
                        50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

3. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.ItemName, DIM_Place.City and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

4. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.Color, DIM_Place.Country and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

Activity 3. Aggregation with CUBE

CUBE takes a specified set of grouping columns and creates subtotals for all of their possible combinations. In terms of multidimensional analysis, CUBE generates all the subtotals that could be calculated for a data cube with the specified dimensions. If n columns are specified for a CUBE, there will be 2 to the n combinations of subtotals returned.

An example:

```
SELECT Dim1, Dim2, SUM(Measure)
FROM FactTab
GROUP BY CUBE(Dim1, Dim2)
ORDER BY Dim1, Dim2
```

1. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the FACT_Sale.ID_Item and FACT_Sale.ID_Place fields. An example of the required result:

ID_ITEM	ID_PLACE	TOTAL
1	1	4363.55
1	2	4794.76
1	3	4718.25
1	4	5387.45
1	5	5027.34
1		24291.35
2	1	5652.84
2	2	4583.02
2	3	5555.77
2	4	5936.67
2	5	4508.74
2		26237.04
	1	10016.39
	2	9377.78
	3	10274.02
	4	11324.12
	5	9536.08
		50528.39

ASK INSTRUCTOR to verify the results of your activity.

2. Modify the query above to calculate subtotals across the Dim_Item.ItemName and Dim_Place.ShopName fields. An example of the required result:

ITEMNAME	SHOPNAME	TOTAL
Bolt	MainShop	4363.55
Bolt	Details	4794.76
Bolt	Repair	4718.25
Bolt	Tools4U	5387.45
Bolt	HomeMaster	5027.34
Bolt		24291.35
Screw	MainShop	5652.84
Screw	Details	4583.02
Screw	Repair	5555.77

```
Screw     Tools4U    5936.67
Screw     HomeMaster 4508.74
Screw                26237.04
          MainShop   10016.39
          Details     9377.78
          Repair     10274.02
          Tools4U    11324.12
          HomeMaster 9536.08
                     50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

5. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.ItemName, DIM_Place.City and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

6. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.Color, DIM_Place.Country and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

Lab 2. Data Mining with KNIME

OBJECTIVE. In this lab you will learn KNIME, an open-source user-friendly graphical workbench for the data analysis process (data access, data transformation, predictive analytics, visualization and reporting).

Activity 4. KNIME Basics Learning and First Run

1. Read KNIME's Workbench User Guide.
2. Test yourself if you really understand the following KNIME's basic terms: workflow, workspace, node, node status, connecting and configuring nodes.
3. Run KNIME using PVC and specify some folder in your profile to store your projects, like it is depicted at Fig. 2.

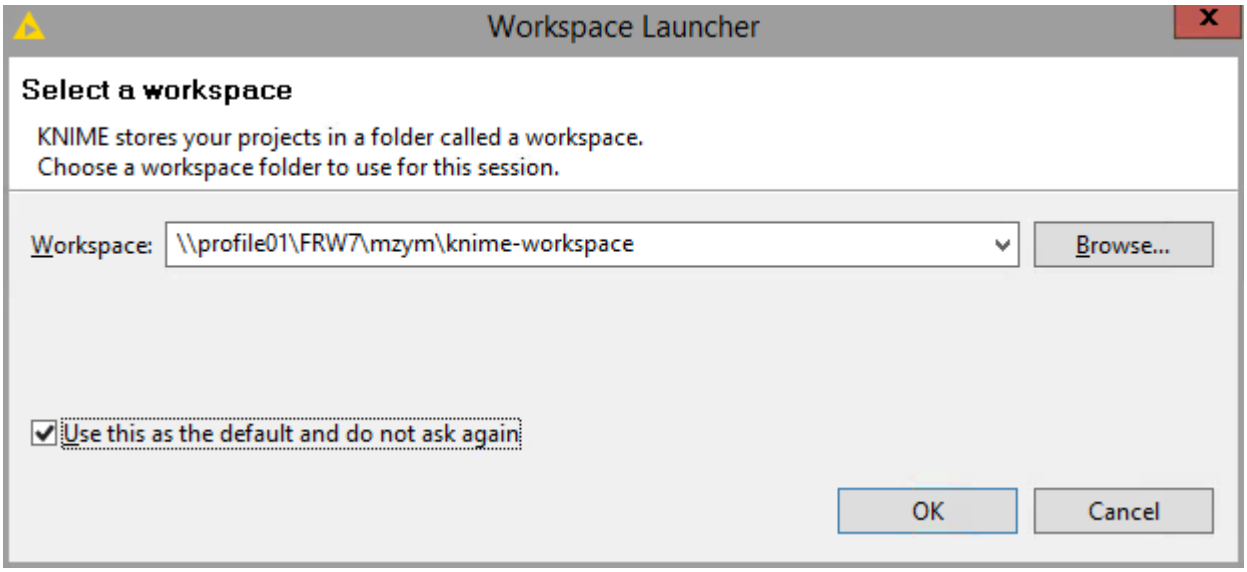

Fig. 2. Specifying a path to KNIME's workspace

ASK INSTRUCTOR to verify the results of your activity.

Activity 5. Building and Running a Workflow Step-by-step

The aim of this activity is to take you step-by-step through the process of building a small, simple workflow. This workflow reads data from a text file, assigns color to it, clusters the data and display the data in a table and a scatter plot.

1. Run KNIME, ensure that it starts with an empty workflow.
2. Add a "Read" node to the workflow as follows. In the Node Repository expand the "IO" and the contained "Read" category as depicted at Fig. 3 (left picture) and drag&drop the "File Reader" icon into the Workflow Editor window.

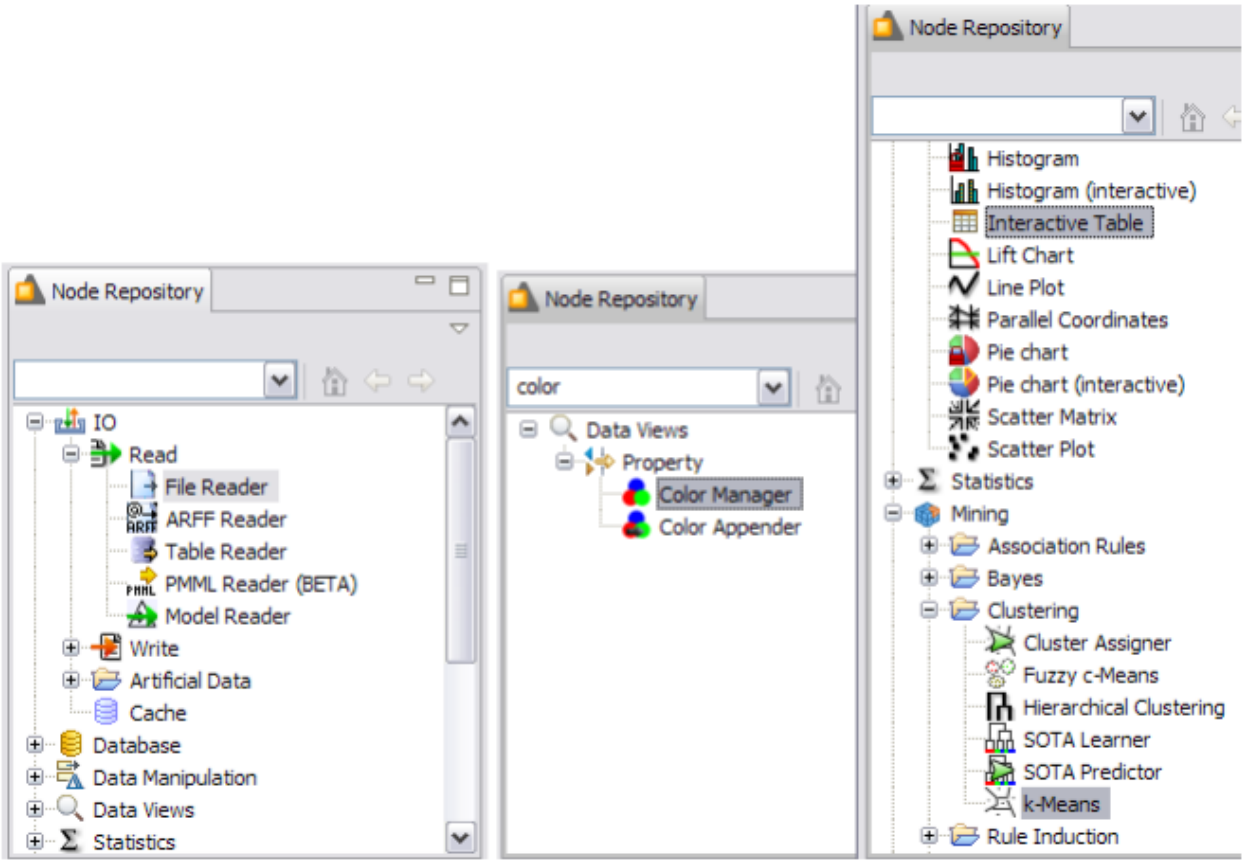

Fig. 3. Nodes of KNIME's Node Repository to construct a simple workflow

3. Proceeding similarly and using Fig. 3, add "k-Means", "Interactive Table" and "Color Manager" nodes to the workflow as depicted at Fig. 4.

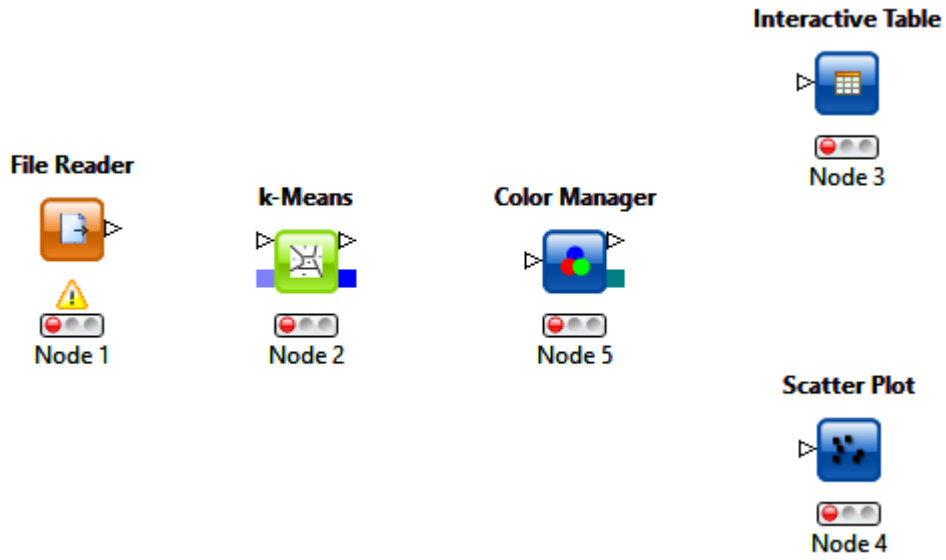

Fig. 4. Simple workflow with non-connected nodes

4. Click an output port and drag the connection to an appropriate input port as depicted at Fig. 5. Note that your nodes will not show a green status, as long as they are not configured and executed.

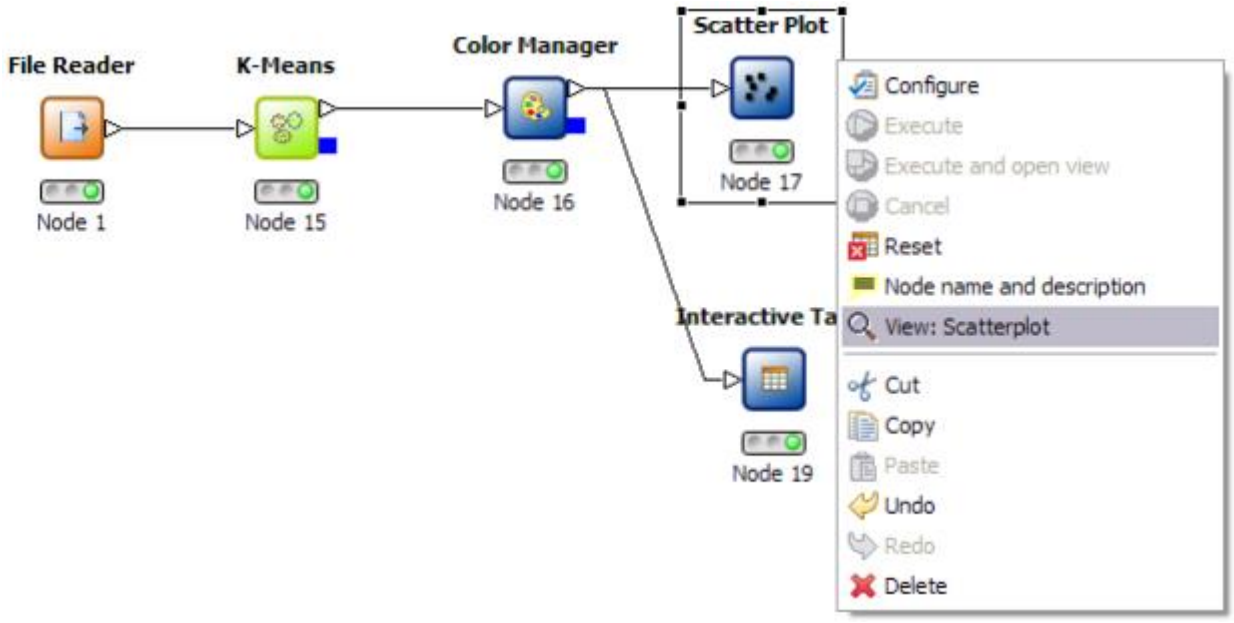

Fig. 5. Simple workflow with connected nodes

5. Right-click the “File Reader” node and select “Configure” from the menu. Navigate to the “IrisDataSet” directory located in the KNIMEDIR. Select the data.all file from this location. The File Reader's preview table shows a sample of the data.
6. Press OK to close the dialog of the “File Reader” node. Once the node has been configured correctly, it switches to yellow (meaning ready for execution). After that, the “K-Means” node will immediately turn yellow, since its default settings will be applied. To be sure, that the default settings fit your needs, open the dialog and inspect the default settings.
7. In order to configure the “Color Manager” node you must first execute the “K-Means” node. After execution all nominal values and ranges of all attributes are known: this meta information is propagated to the successor nodes. The Color Manager needs this data before it can be configured. Once the “K-Means” node is executed, open the configuration dialog of the “Color Manger” node (see Fig. 6).
8. Execute the “Scatter Plot” node, and the KNIME workbench will execute all predecessor nodes. In a larger, more complex flow you could select multiple nodes and trigger execution for all of them. The workflow manager will execute the nodes as needed, if possible in parallel.
9. Open the “K-Means”, “Interactive Table” and “Scatter Plot” nodes’ views using their context menus.

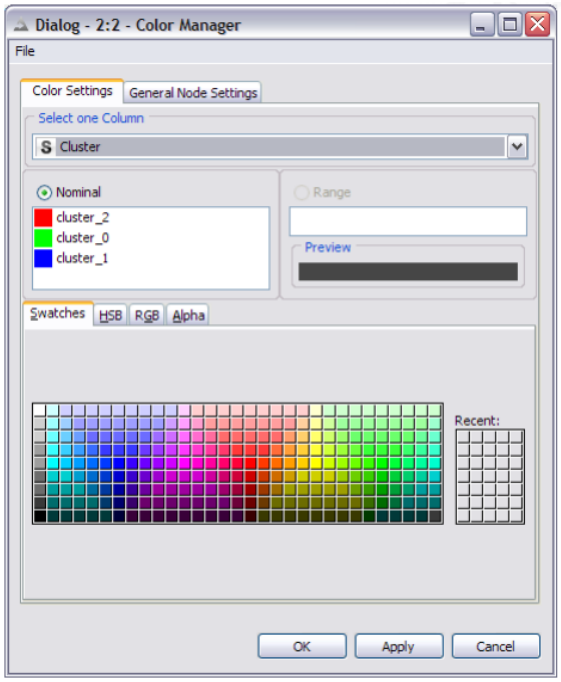

Fig. 6. "Color Manager" node configuring

10. Select some points in the scatter plot and choose "Hilite Selected" from the "Hilite" menu. The hilited points are marked with an orange border. You will also see the hilited points in the table view. The propagation of the hilite status works for all views in all branches of the flow displaying the same data.

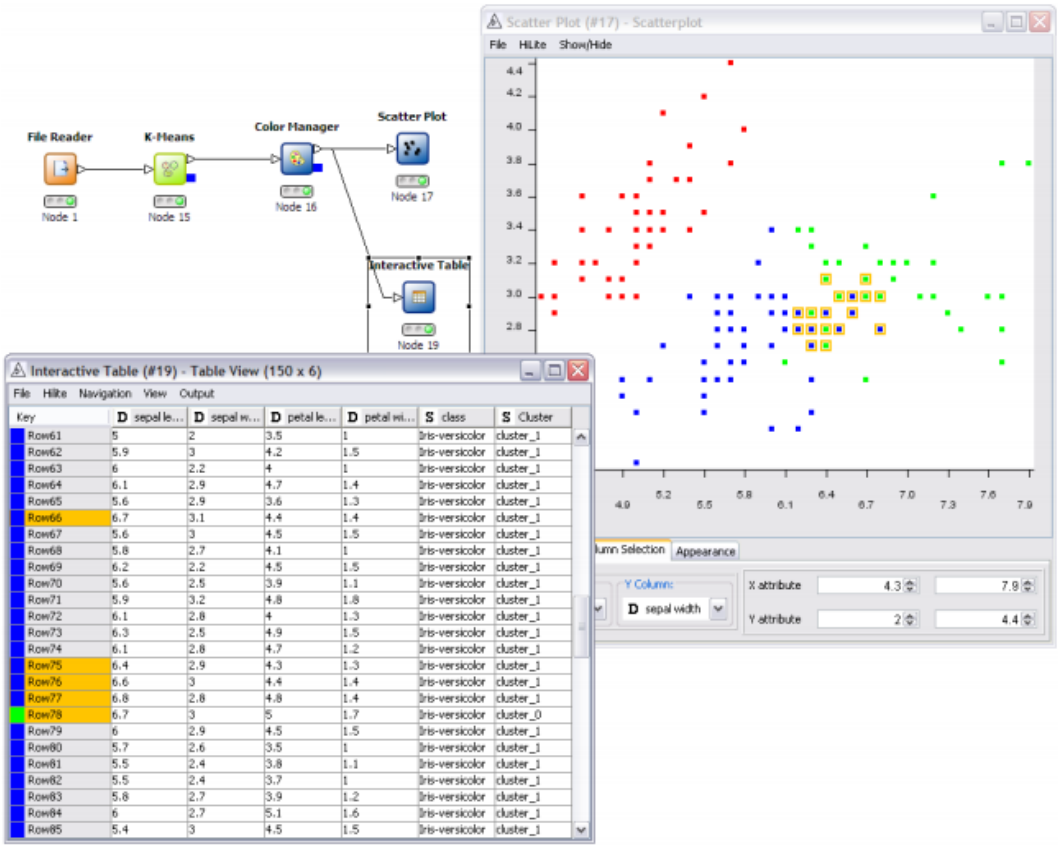

Fig. 7. Hiliting of selected data

ASK INSTRUCTOR to verify the results of your activity.

Activity 6. Building an Association Rule Mining Workflow

The aim of this activity is to build a simple workflow concerning market basket analysis problem and association rule mining. This workflow reads data from a text file with market basket data, converts it into a special collection data type, finds frequent itemsets and association rules and displays the results.

1. Download the baskets.csv file (zip-archive). This file contains the anonymized retail market basket data from an anonymous Belgian retail store. Each row represents one basket and contains the comma-separated IDs of the purchased items. There are 88162 rows in total, 16469 distinct items and 30 is maximum itemset's length.
2. Run KNIME and create a new workflow named MarketBasket. You are to create the workflow depicted at the Fig. 8.

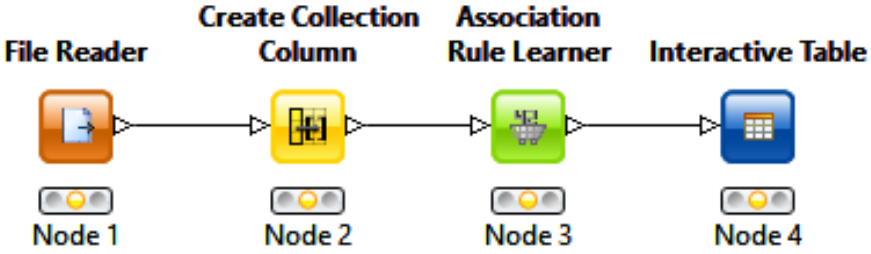

Fig. 8. Simple workflow concerning market basket analysis problem

3. Add a File Reader node and configure it in the following way: check the "Read column headers" option, set a comma as a column delimiter and allow short (non-completed) lines ("Advanced" button, "Short lines" tab).
4. Add a Create Collection Column node and read its description. Configure this node in the following way: include all the input file columns into the output collection and check the "Create collection of type set", "Ignore missing values", "Remove aggregated columns from table" options.
5. Connect the File Reader and Create Collection Column nodes and execute them. Ensure that result is of collection data type.
6. Add an Association Rule Learner node and read its description. Add an Interactive Table node and connect existing nodes in an appropriate way.
7. Configure the Association Rule Learner node with different minimum support and maximal itemset length values and various itemset type (free/closed/maximal), then execute whole workflow. Explain the results (how do the parameters mentioned above affect the resulting frequent itemsets?).

ASK INSTRUCTOR to verify the results of your activity.

8. Configure the Association Rule Learner node to output association rules with different minimum confidence and various itemset type (free/closed/maximal), then execute whole workflow. Explain the results (how do the parameters mentioned above affect the resulting association rules?).

ASK INSTRUCTOR to verify the results of your activity.

9. Build and execute an association rule mining workflow to find frequent itemsets and association rules from the following market basket data:

ID	Itemsets
1	water, cola, bread, chips, nuts
2	water, chips
3	bread, cola, chips
4	water, nuts
5	cola, chips
6	water
7	nuts, cola, bread, chips
8	cola, bread, chips
9	cola, chips
10	nuts, cola, bread, chips

ASK INSTRUCTOR to verify the results of your activity.

10. Solve the market basket problem mentioned above using Oracle DBMS and SQL.
    - Connect to Oracle XE DBMS using Oracle SQL Developer and create Basket table with the following fields: ID (i.e. identity of the basket) and Item (i.e. one of the item of this basket's itemset).
    - Fill in the Basket table with the data mentioned above as follows:

ID	Item
1	water
1	cola
1	bread
1	chips
1	nuts
2	water,
2	chips
…	…

    - Create table Cand with the same structure as Basket table. Generate candidates to frequent itemsets using the following query (change *minsup* into 5):

```
INSERT INTO Cand
  SELECT *
  FROM Basket
  WHERE Item in (
    SELECT Item
    FROM Basket
    GROUP BY Item
    HAVING COUNT(*) >= minsup)
```

    - Find frequent 2-itemsets using the following query (change *minsup* into 5):

```
SELECT A.Item, B.Item, COUNT(A.ID) AS Sup_Count
FROM Cand A, Cand B
WHERE A.ID=B.ID AND A.Item<B.Item
GROUP BY A.Item, B.Item
```

```
HAVING COUNT(A.ID) >= minsup;
```

Compare the frequent 2-itemsets found here with the frequent 2-itemsets found at the previous step using KNIME.

– Design and run queries to find frequent 3-itemsets and 4-itemsets. Compare the results found here with the results found at the previous step using KNIME.

ASK INSTRUCTOR to verify the results of your activity.

– All the previous queries to find frequent itemsets treated *minsup* as support *count*, i.e. *number* of baskets that contain an itemset. Rewrite these queries with respect to *minsup* treated as a *percentage* of baskets that contain an itemset. It means that *minsup*=5/10=0.5 instead of *minsup*=5 should be used, where 10 is number of baskets in the Basket table. Note that number of baskets is not equal number of records in the Basket table; number of baskets should be calculated as a number of groups of records with the same ID. Run queries to find frequent {1,2,3,4}-itemsets. Compare the results found here with the results found at the previous step.

ASK INSTRUCTOR to verify the results of your activity.

Activity 7. Building a Decision Tree Workflow

The aim of this activity is to build a simple workflow concerning classification problem by means of decision tree. This workflow reads data from a text file with data to be classified, builds a decision tree and displays the results.

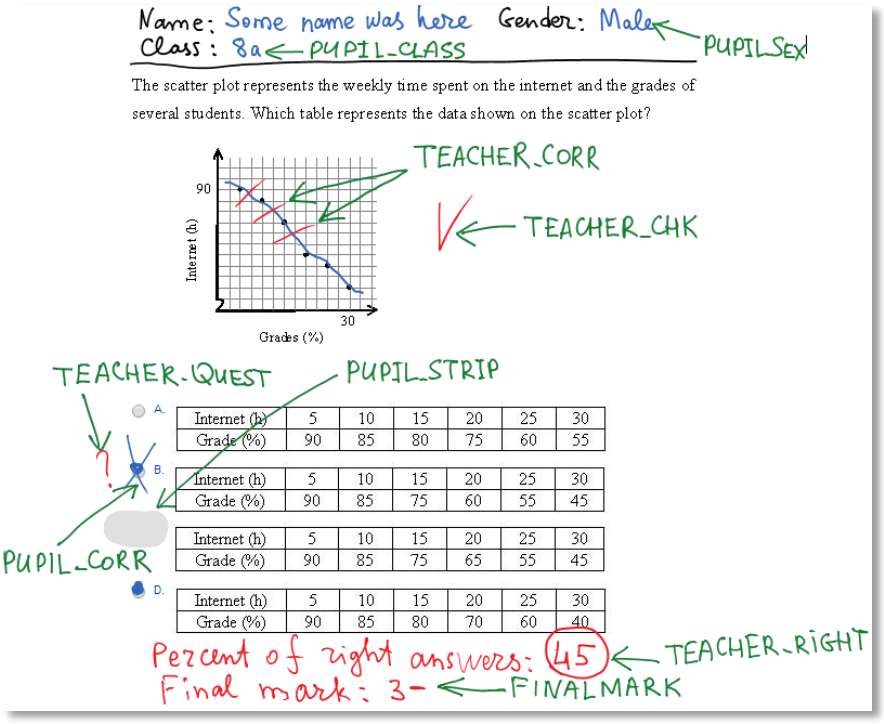

Fig. 9. An example of quiz with mark's attributes semantic

1. Download the marks.csv file. This file contains the anonymized data of pupils quiz marks. Mark's attributes depicted at the Fig. 9.
2. Run KNIME and create a new workflow named Classification. You are to create the workflow depicted at the Fig. 10.

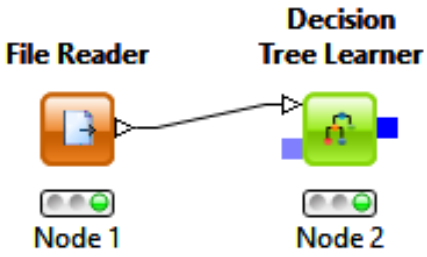

Fig. 10. Simple workflow concerning classification problem by means of decision tree

3. Add a File Reader node and configure it in the following way: check the "Read column headers" option, set a comma as a column delimiter.
4. Add a Decision Tree Learner node and read its description. Connect existing nodes in an appropriate way.
5. Configure the Decision Tree Learner node, setting FINALMARK as Class column. Execute whole workflow. Ensure that resulting decision tree depicts that the final mark is objective and depends on teacher's attributes only (e.g. TEACHER_RIGHT or TEACHER_CORR, etc.), not on pupil's attributes.
6. Try different values of the Decision Tree Learner node parameters, then execute whole workflow. Ensure that results still show that final mark depends on teacher's attributes only.

ASK INSTRUCTOR to verify the results of your activity.

7. Add a Color Manager and a Decision Tree nodes and perform connections between nodes as depicted at the Fig. 11. Read descriptions of the nodes.

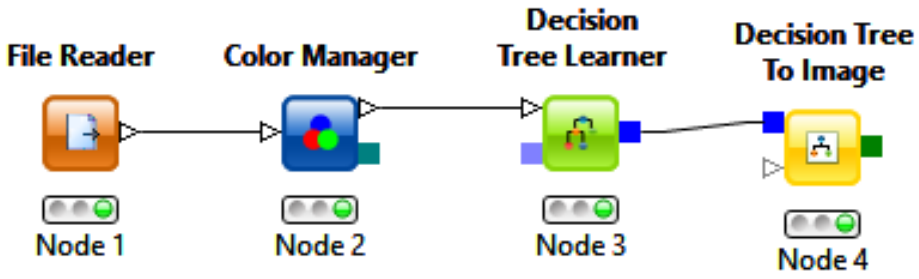

Fig. 11. Simple workflow concerning classification problem with additional nodes

8. Configure Color Manager node and select FINALMARK as a colored column. Execute whole workflow.
9. Using Decision Tree Learner node's context menu, try "Decision Tree View" and "Decision Tree View (simple)" commands and compare results.

10. Using Decision Tree to Image node's context menu, try "Decision Tree View" command. Try different values of node parameters, then execute whole workflow. Try to get lowest and highest decision tree images.

ASK INSTRUCTOR to verify the results of your activity.

Activity 8. Building a Clustering Workflow

The aim of this activity is to build a simple workflow concerning clustering problem. This workflow reads data from a text file with data to be clustered and displays the results.

1. Download the basketball.csv file. This file contains the anonymized data of basketball players (age, height, average time played in minutes, assists per minute and points per minute).
2. Run KNIME and create a new workflow named Clustering. Create the workflow depicted at the Fig. 12.

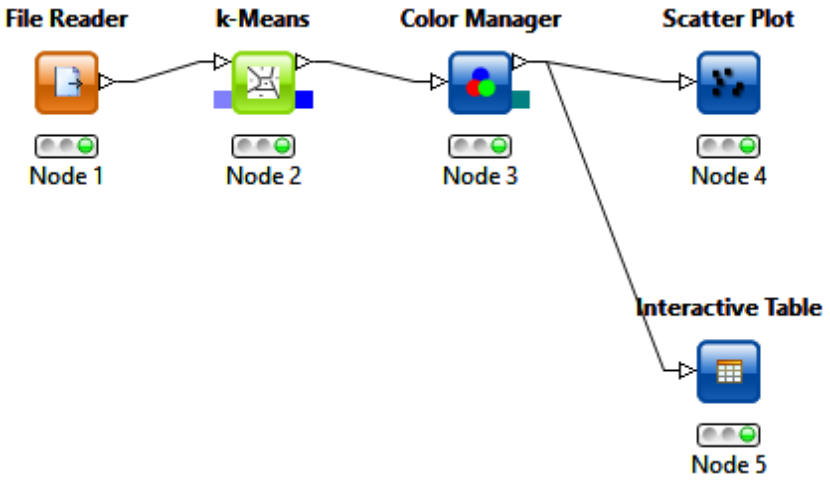

Fig. 12. Simple workflow concerning clustering problem

3. Execute the workflow while changing the following parameters: number of clusters, colors to depict clustered points. Repeat the same with different attributes of tuples (e.g. height, assists per minute and points per minute or average time played in minutes, assists per minute and points per minute, etc.).

ASK INSTRUCTOR to verify the results of your activity.

4. Add Hierarchical node and connect it with the File Reader node. Execute the workflow while changing the following parameters: number of clusters, colors to depict clustered points. Repeat the same with different attributes of tuples (e.g. height, assists per minute and points per minute or average time played in minutes, assists per minute and points per minute, etc.). Compare the results of clustering by k-Means algorithm and hierarchical clustering algorithm.

ASK INSTRUCTOR to verify the results of your activity.

Activity 9. Building a Data Preprocessing Workflow

The aim of this activity is to build a simple workflow to preprocess data. We will use anonymized data of 1994 USA census where each data tuple consists of the attributes listed below.

No	Attribute	Semantic
1.	AGE	The age of the individual.
2.	WORKCLASS	The type of employer the individual has (possible values are Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked).
3.	FNLWGT	The number of people the census takers believe that observation represents.
4.	EDUCATION	The highest level of education achieved for that individual (possible values are Bachelor, Some-college, Master, Doctorate, etc.).
5.	EDUCATION-NUM	Highest level of education in numerical form.
6.	MARITAL-STATUS	Marital status of the individual (possible values are Married-civ-spouse, Divorced, Never-married, etc.).
7.	OCCUPATION	The occupation of the individual (possible values are Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Farming-fishing, Transport-moving, Armed-Forces, etc.).
8.	RELATIONSHIP	Family relationship of the individual (possible values are Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried).
9.	RACE	The individual's race (possible values are White, Asian-Pac-Islander, Amer-Indian-Eskimo, Black, Other).
10.	SEX	The individual's gender (possible values are Female, Male).
11.	CAPITAL-GAIN	The individual's capital gains recorded.
12.	CAPITAL-LOSS	The individual's capital losses recorded.
13.	HOURS-PER-WEEK	Hours worked by the individual per week.
14.	NATIVE-COUNTRY	Country of origin for the individual (possible values are United-States, Cambodia, England, Puerto-Rico, Canada, Germany, etc.).
15.	INCOME	A flag to show whether the individual makes over $50000 a year (possible values are >50K, <=50K).

You are to perform consequently the following preprocessing steps:

1) Split the input table vertically into two tables: the First, with nominal columns only (i.e. WORKCLASS, EDUCATION, MARITAL-STATUS, OCCUPATION, RELATIONSHIP, RACE, SEX, INCOME) and the Second, with numerical columns only (i.e. AGE, FNLWGT, EDUCATION-NUM, CAPITAL-GAIN, CAPITAL-LOSS, HOURS-PER-WEEK).
2) Keep all the rows of the First table where EDUCATION is Bachelor or Master, and remove others.
3) Keep all the rows of the Second table where AGE between 21 and 65, and remove others.
4) Join the First and the Second tables to enforce both criteria mentioned above.

5) Calculate various statistics (min, max, mean, etc.) of the resulting table from the previous step.
6) Keep AGE and HOURS-PER-WEEK columns of the Second table, and remove others.
7) Draw box plot for the resulting table from the previous step.

1. Download the adult.csv file (zip-archive) with census data.
2. Run KNIME and create a new workflow named Data preprocessing. Create the workflow depicted at the Fig. 13.

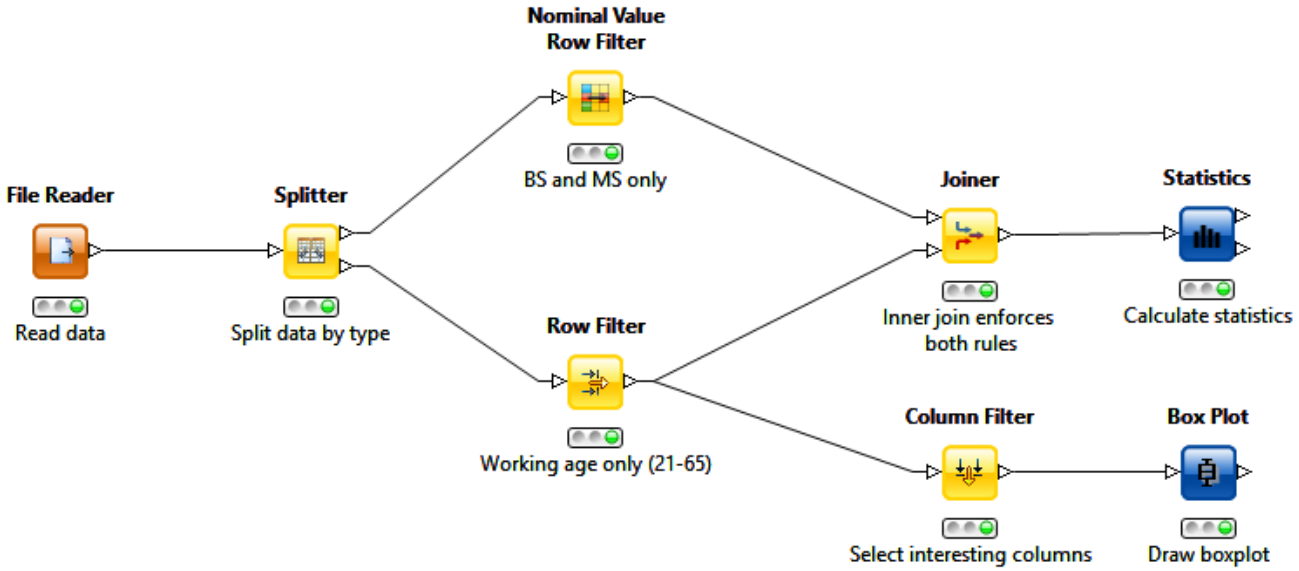

Fig. 13. Simple workflow for data preprocessing

Configure every node according to the preprocessing steps mentioned above. Having added a new node execute it and verify partial results.

ASK INSTRUCTOR to verify the results of your activity.

3. Execute the whole workflow and explain the results.

ASK INSTRUCTOR to verify the results of your activity.

4. Create a copy of the Data preprocessing workflow (e.g. named Data preprocessing-2). Change configuration of the nodes to perform modified preprocessing steps mentioned above at your option: e.g. WORKCLASS is Private, Without-pay or Never-worked at the step 2; HOURS-PER-WEEK is greater than 40 at the step 3, etc. Execute the whole workflow and explain the results.

ASK INSTRUCTOR to verify the results of your activity.