

## Guided Tutorial for Pentaho Data Integration using MySQL

In the data integration exercise, you will use the Pentaho Data Integration tool to transform two data sources and load data into a MySQL fact table. You will perform transformations to parse date strings, combine fields, and perform validation checks. Before starting this tutorial, you need to install necessary software, download data sources, and create tables used in the tutorial.

### 1. Tutorial Prerequisites

Before starting this tutorial, you should download and install the server and client for MySQL. You can find details in Module 1 about MySQL installation. If you have access to a remote MySQL server (perhaps through your employer), you do not need to install the server software on your own machine.

You also need to install Pentaho Data Integration before starting this tutorial. After installing Pentaho Data Integration, you need to install the Java Database Connectivity (JDBC) driver for MySQL. Module 1 contains installation instructions about Pentaho Data Integration and JDBC drivers. This tutorial demonstrates the community edition of the most recent stable version (5.0.1) of Pentaho Data Integration.

After installing Pentaho Data Integration, you need to obtain the data sources used in the tutorial from the class website.

- Excel file used in part 1 of the tutorial
- Access database used in part 2 of the tutorial

The tutorial uses the Store Sales data warehouse as depicted in Figure 1. Sales is the fact entity type surrounded by 1-M relationships with dimension entity types, Item, Customer, Store, and TimeDim. The schema design has a snowflake for the 1-M relationship from Division to Store. In the table design, table names have been preceded with the prefix “SS” to avoid conflicts with other tables. Thus, the fact table is SSSales, not Sales as shown in the ERD of Figure 1.

The class website contains documents for Oracle and MySQL. You need to create and populate the tables using one of these documents. The Oracle document also contains a statement to create a sequence object for the *SSSales* table.

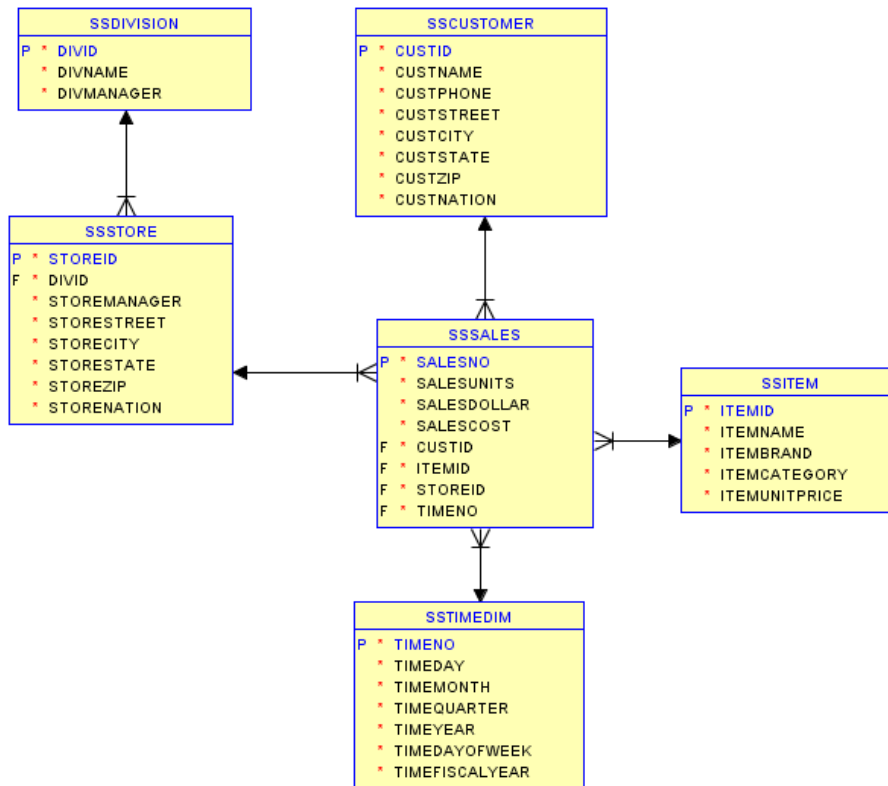


Figure 1: Oracle Snowflake Schema for the Store Sales Data Warehouse

## 2. Creating your First Transformation

The Data Integration component of Spoon allows you to create transformations and jobs. Transformations involve data flows such as reading from a source, transforming data and loading it into a target location. Jobs coordinate transformations such as defining dependencies among transformations and execution conditions such as, "Is my source file available?" or "Does a table exist in my database?"

This exercise will step you through building your first transformation with Pentaho Data Integration introducing common concepts along the way. Follow the instructions below to create a new transformation.


1. After starting Pentaho Data Integration, you will see the opening window (Figure 2) and the Spoon window (Figure 3).
2. Click  (New) in the upper left corner of the Spoon window.
3. Select **Transformation** from the list of components (Figure 4) displayed after selecting the **New** button.



Figure 2: Pentaho Data Integration Welcome Window

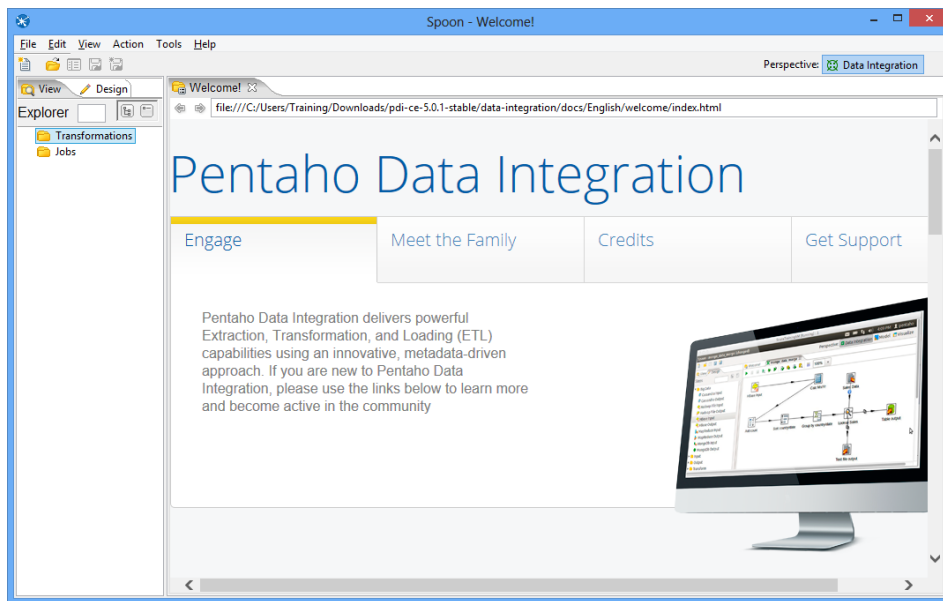


Figure 3: Spoon Opening Window

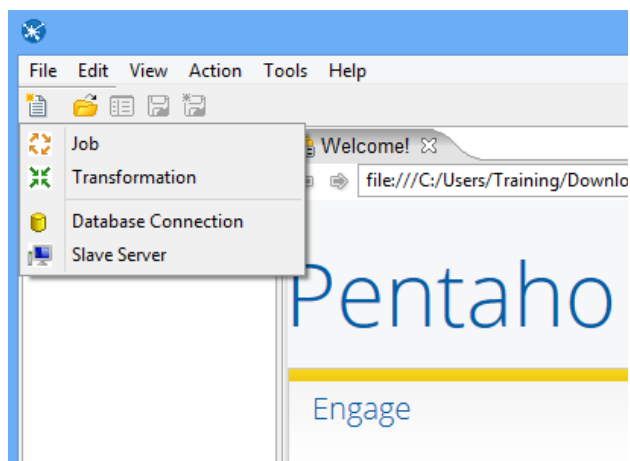


Figure 4: Spoon Transformation List

### 3. Load the first data source from Excel

Make sure that you have downloaded the Excel input file from the class website. You need to know the location of this file in Step 4 below.

Step 1 – In the View tab, right click the new transformation 1 and select “settings...”

Step 2 – Set the Transformation name for the new transformation as: SSTORETEST and click OK.

Step 3 – Save the transformation following **File** → **Save**. You will see the empty transformation window in the Spoon (Figure 5).

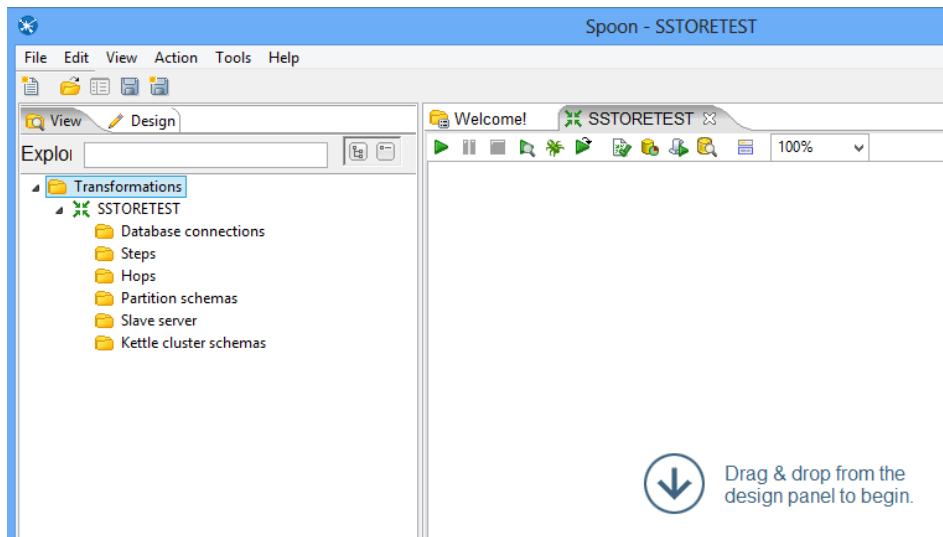


Figure 5: Empty Transformation Window

Step 4 – Create the Excel Input step:

- Under the Design tab, expand the Input node (Figure 6).

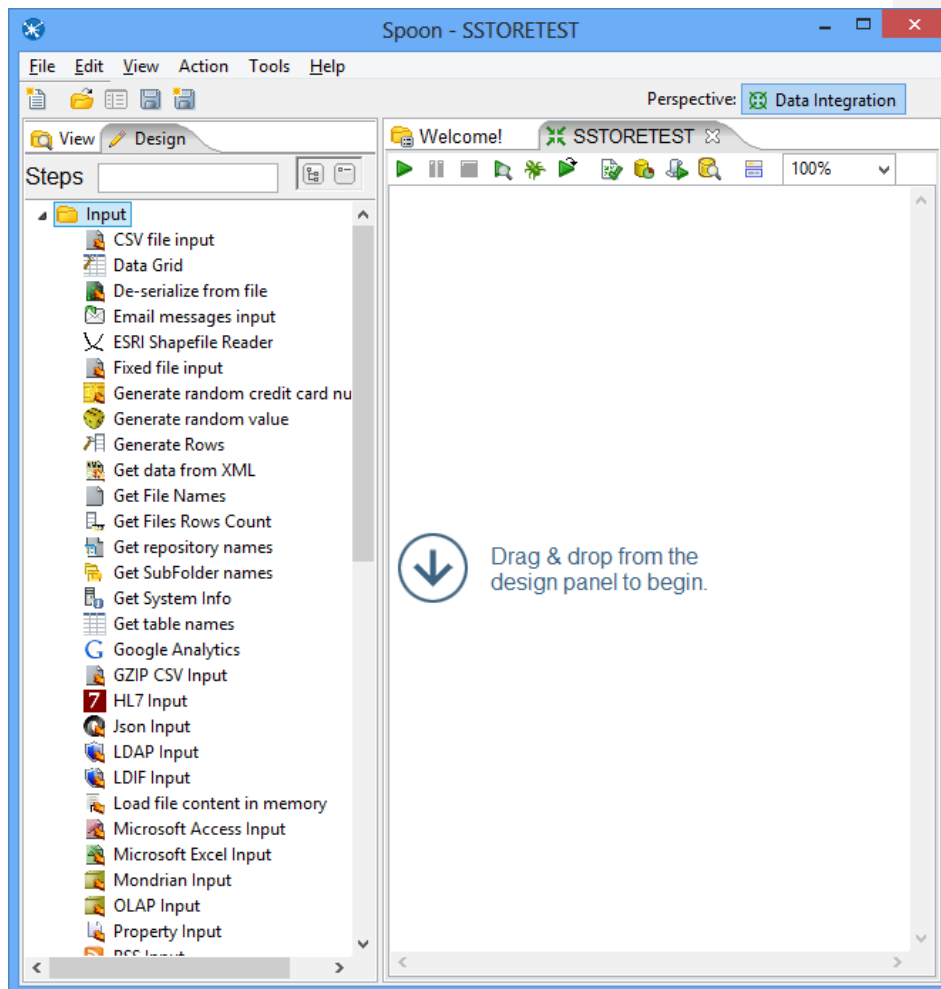


Figure 6: New Microsoft Excel Input Node

- Select and drag a **Microsoft Excel Input** step into the canvas on the right.
- Double Click on the **Microsoft Excel Input** step. The edit properties dialog box (Figure 7) associated with the **Microsoft Excel Input** step appears. In this dialog box, you specify the properties related to a particular step.

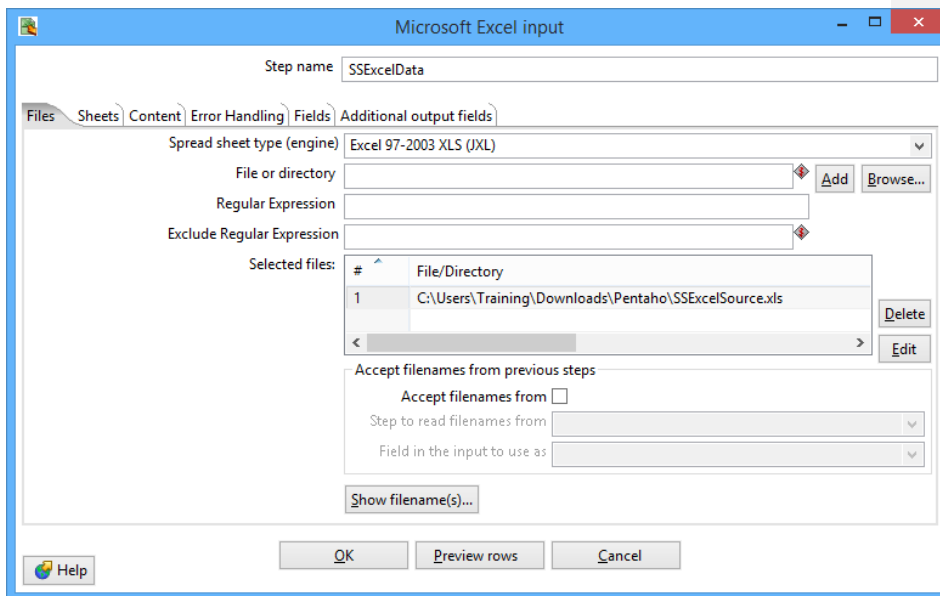


Figure 7: Files Window for Microsoft Excel Input Property Editing

- Set name for the Excel Input as **SSEExcelData** and specify the Excel data source path in the **Files** tab.
- In the tab named **Files**, click the button “Browse...” and locate the Excel file that you downloaded from the class website. Then, Click “Add” to add the file to the selected files area.
- In the tab named **Sheets**, click the button “Get sheetname(s)...”. There will appear an **Enter List** (Figure 8) to choose sheets. Select **Sheet 1**, press “>” to move it into the right area. Click **OK**.
- In the tab names **Fields**, click on “Get fields from header row...” You need to change the data types, length, and precision as the specification in Figure 9.

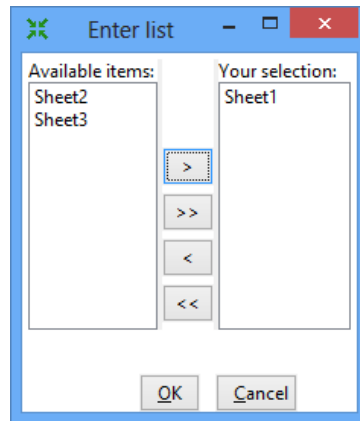


Figure 8: Sheet Specification Window

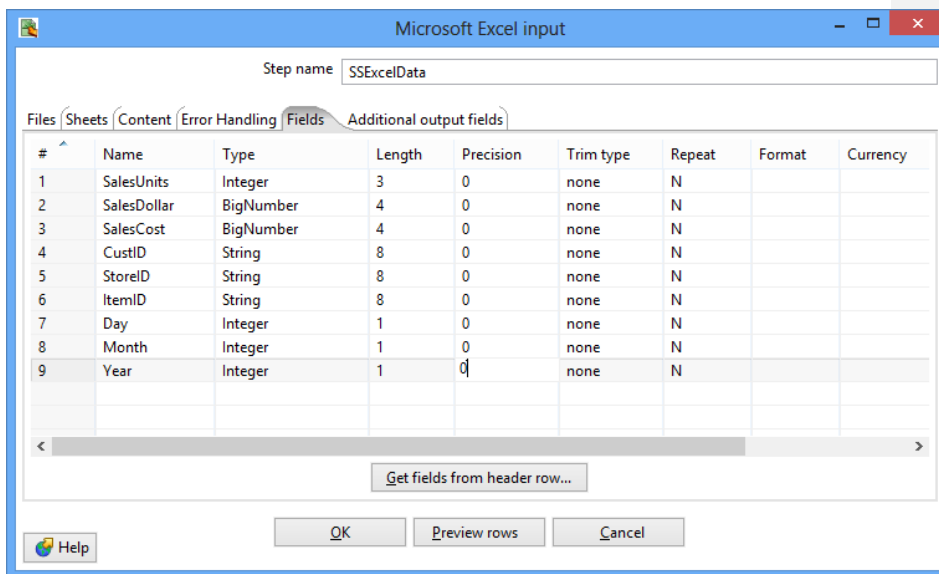


Figure 9: Fields Window for Microsoft Excel Input Property Editing

- Click **OK** at the bottom of the window. The input icon will change to the SSEExcel icon displayed in Figure 10.

Step 5 – In this part of the tutorial, you will add constraint checking for null values and appropriate data types for the Excel data source.



- Add a Filter Rows step to your transformation. Under the **Design** table, go to **Flow** → **Filter Rows** (Figure 10).

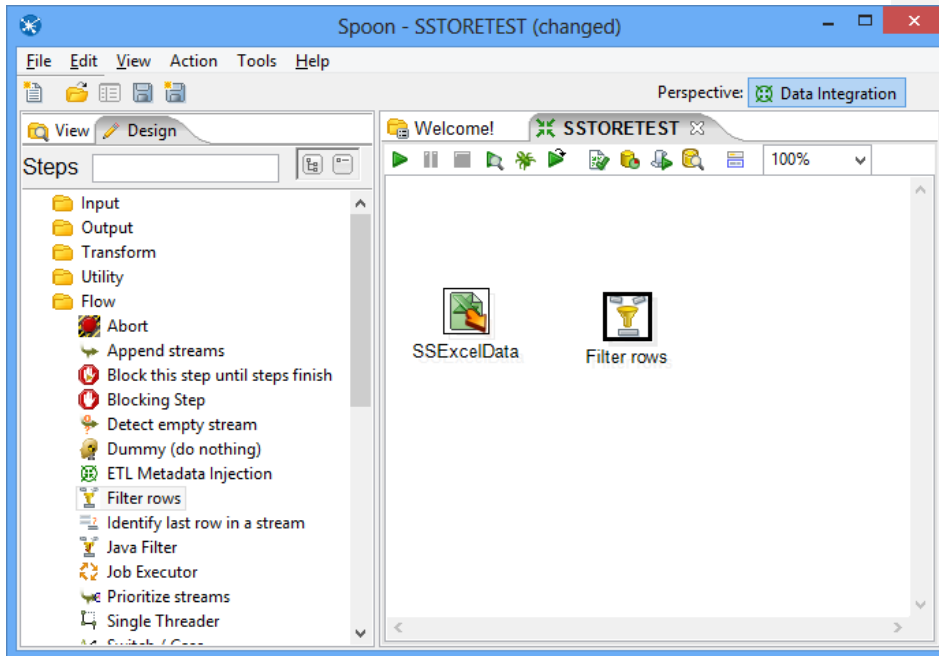


Figure 10: Excel Input Node and Filter Node in Spoon

- Create a “hop” between the **SSEExcelSource** (Excel file input) step and the **Filter Rows** step. Hops are used to describe the flow of data in your transformation. To create the hop, click the **SSEExcel Source** (Excel file input) step, then press the <SHIFT> key down and draw a line to the Filter Rows step (Figure 11).



Figure 11: Hop connecting an Excel Input Node Connected to a Filter Node

- Alternatively, you can draw hops by hovering over a step until the hover menu (Figure 12) appears. Drag the hop painter icon from the source step to your target step.

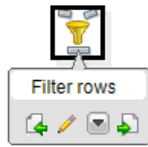


Figure 12: Hover Menu

- Double-click the **Filter Rows** step. The **Filter Rows** edit properties dialog box appears (Figure 13).

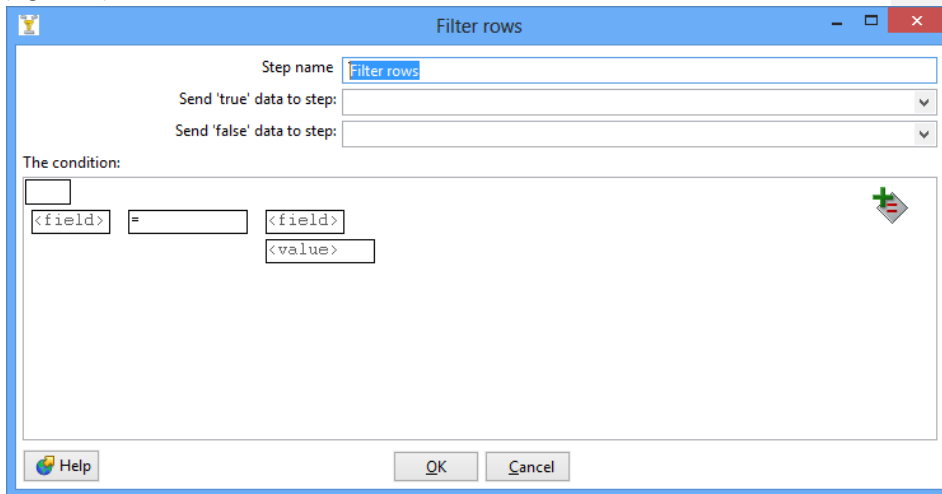


Figure 13: Property Edit Window of Filter Node

- The **Step Name** field is **Filter rows by default**.
- Under **The condition**, click <field>. A dialog box that contains the fields you can use to create your condition appears.
- In the **Fields** dialog box (Figure 14) select **SalesUnits** and click **OK**.

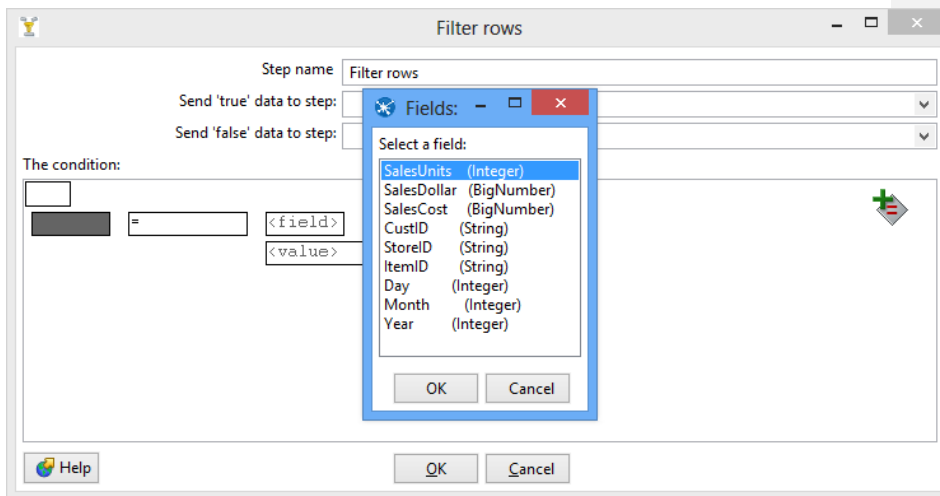


Figure 14: Condition Fields Selection Window

- Click on the comparison operator (Figure 15) (set to = by default) and select the **IS NOT NULL** function and click **OK**.

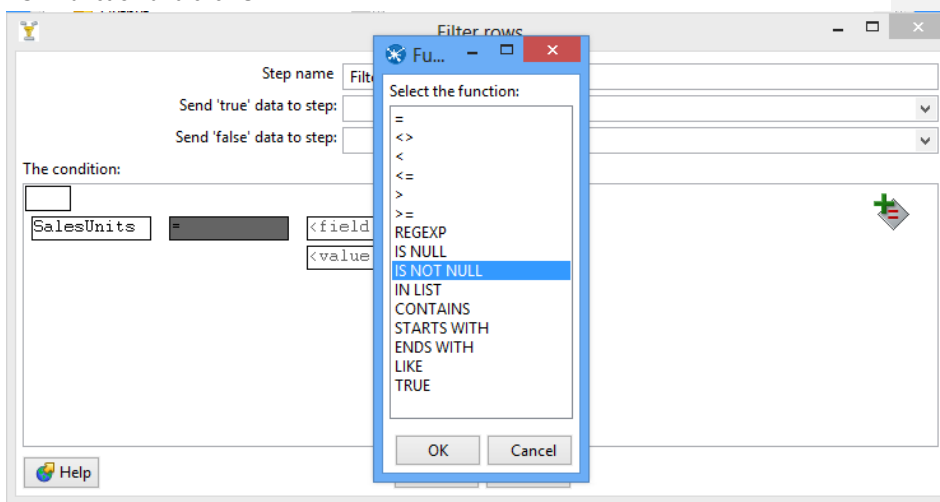




Figure 15: Comparison Operator List

- Click the button . A new condition row appears with **null = [ ]** as a default.
- Click on the expression and add constraints for the next column similarly to what you did for "SalesUnits"

- Click on **UP**. This will allow you to see both conditions joint by AND
- Click the button  again. Another new condition row appears with **null = [ ]** as a default.
- Keeping repeating these steps for all fields.
- The final view of filter conditions is shown by Figure 16.

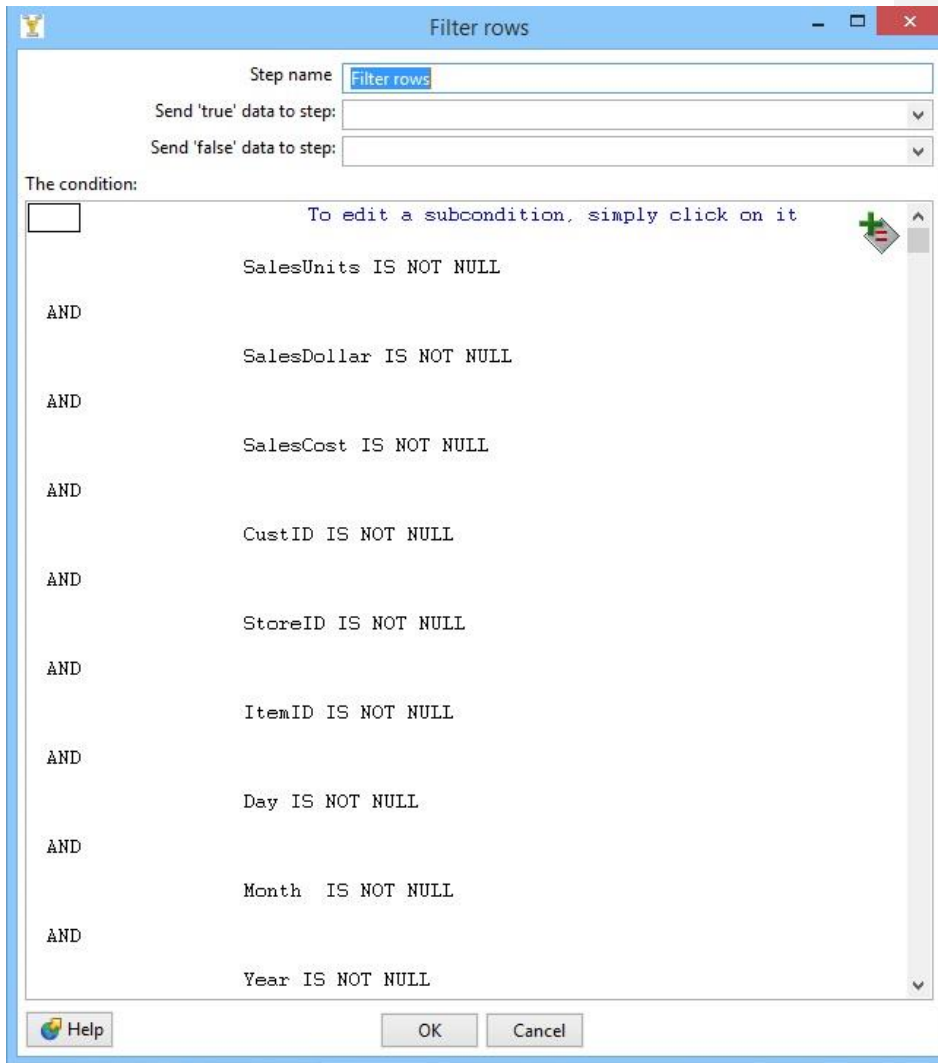


Figure 16: Filter Conditions Window

- Save your transformation.

Step 6 – Create a step to sort the result of the Filter Rows step.

- Under the **Design** tab, expand the contents of the **Transform** node.
- Click and drag a **Sort Rows** step into your transformation; create a hop between the **Filter rows** and Sort Rows steps. Select **Result is TRUE** in the filter results selection list (Figure 17).



Figure 17: Filter Results Selection List

- Double-click the **Sort Rows** step to open its edit properties dialog box (Figure 18). Click **Get Fields** to obtain the fields. Delete other fields except the Day, Month and Year fields. Then click Ok.

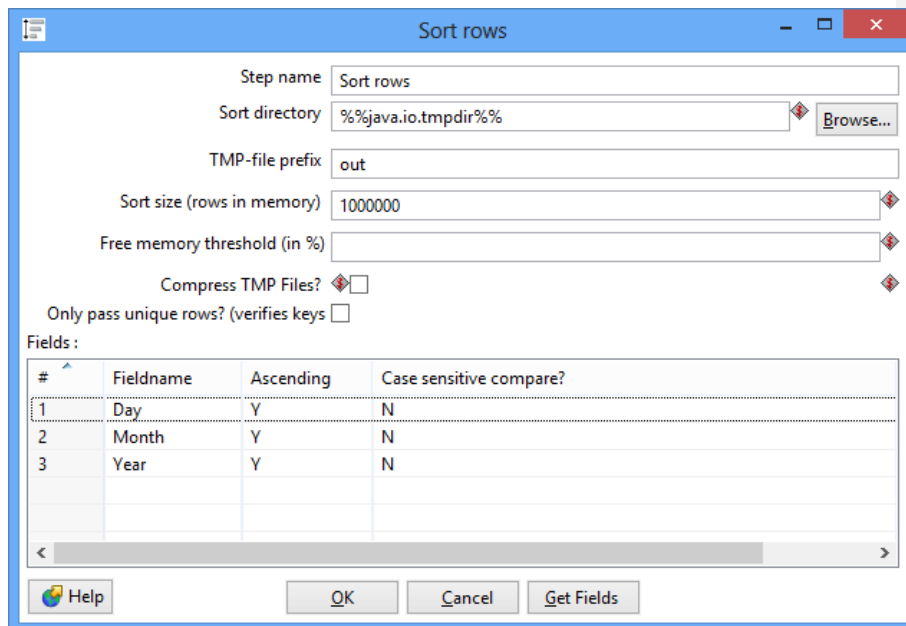


Figure 18: Property Edit Window of Sort Rows Node

#### 4. Using a Database Connection to Lookup Columns from MySQL tables

Pentaho Data Integration allows you to define connections to multiple databases provided by multiple database vendors (MySQL, Oracle, Postgres, and many more). Pentaho Data Integration ships with the most suitable JDBC drivers for supported databases and its primary interface to databases is through JDBC. Vendors write a driver that matches the JDBC specification and Pentaho Data Integration uses the driver. Unless you require extensive debugging or have other needs, you won't ever need to write your own database driver.

When you define a database connection, the connection information (username, password, port number, and so on) is stored in the Pentaho Enterprise Repository and is available to other users when they connect to the repository. If you are not using the Pentaho Enterprise Repository, the database connection information is stored in the XML file associated with a transformation or job.

Connections that are available for use with a transformation or job are listed under Database **Connection** node in the explorer **View** in Spoon.

There are several ways to define a new database connection:

- In Spoon, under View in the navigation tap, right click Database connections and choose New.
- In Spoon, under View in the navigation tap, right click Database connections and choose New Connection Wizard.
- In the Table input configuration box, click on New.

This part of the tutorial involves looking up the date from the *SSTimeDim* table to check the validity of dates in the Excel data source. In addition, you will lookup primary key columns from other MySQL tables to ensure loaded data does not contain invalid foreign keys.

Step 1 – Access the *SSTimeDim* table from MySQL database.

- Under the **Design** tab, expand the contents of the **Input** node.
- Click and drag a **Table Input** step into your transformation.
- Double-click the Table Input step to open its edit properties dialog box (Figure 19).
- Rename your Table Input step to *SSTimeDim*.

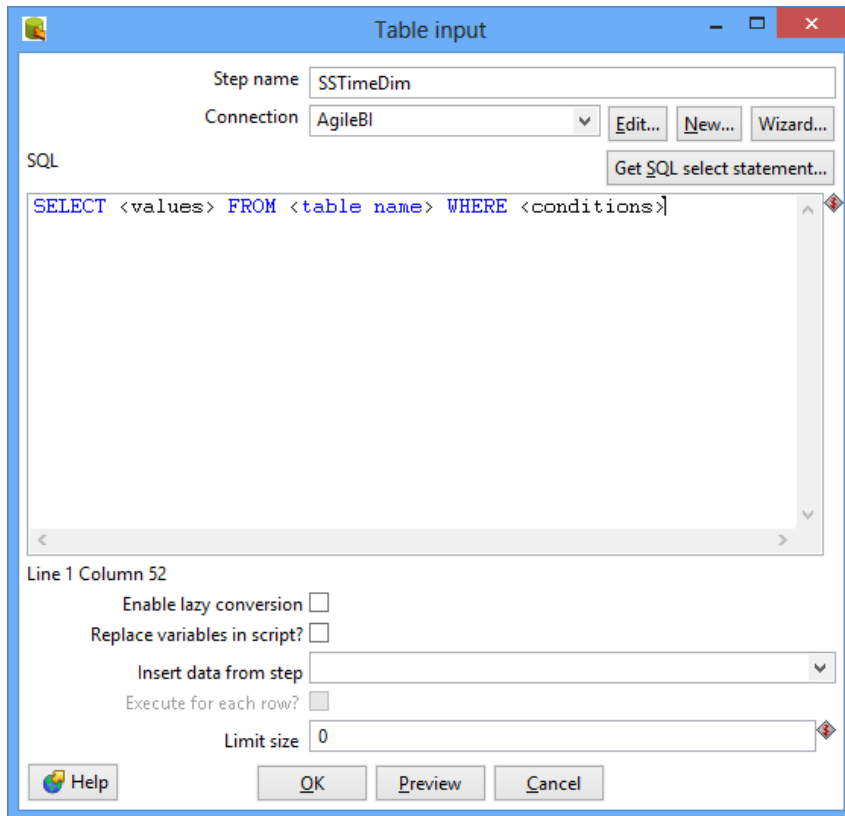


Figure 19: Property Edit Window of Table Input Node

- Click “New...” next to the connection field. You must create a connection to the database. The Database connection dialog box appears.
- Provide the settings for connecting to the database as shown in Figure 20.
- **IMPORTANT:** Before setting the connection information, you should first configure the JDBC driver according to the instructions described in Section 1. Also, if you are using a remote database make sure you are connected through the VPN prior to testing the connection. Figure 20 shows the details to connect to the MySQL server.



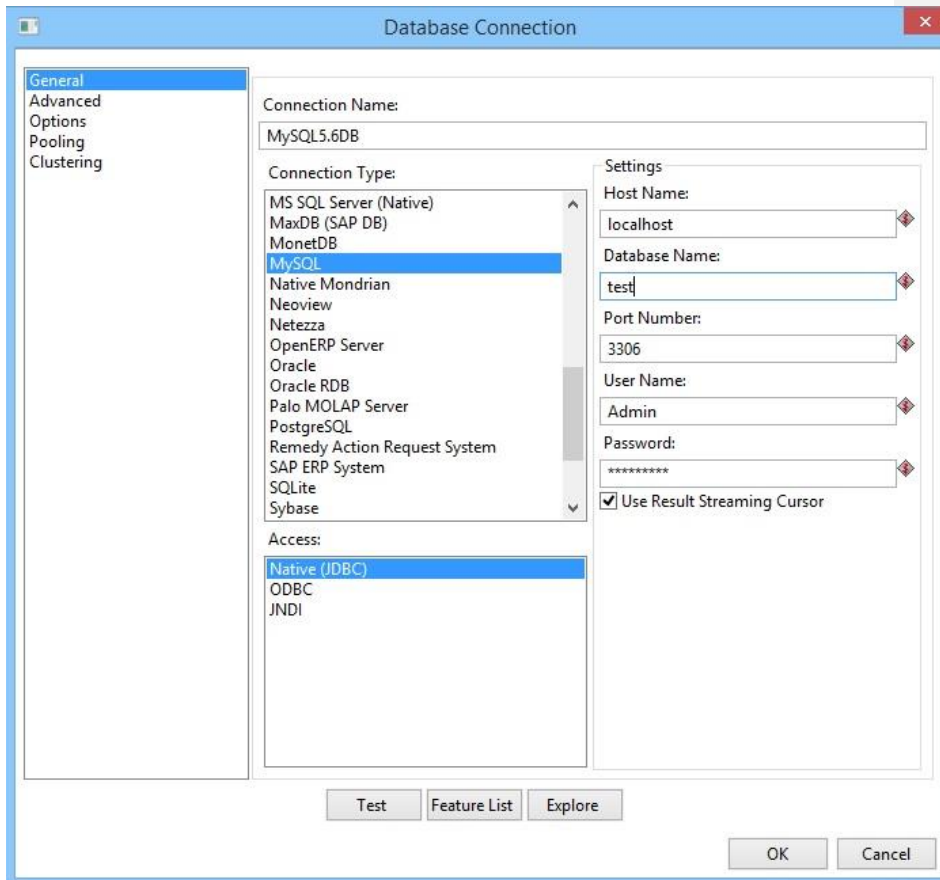


Figure 20: Database Connection Window

- Connection Name: MySQL5.6DB  
Connection Type: MySQL  
Access: Native (JDBC)  
Host Name: localhost  
Database Name: (This should be your database name)  
Port Number: 3360  
User name: (This should be your user name)  
Password: (This should be your password)
- Click "Test" to test the connection. Then success test result is shown by Figure 21.



Figure 21: Database Connection Test

- Type in "SELECT \* FROM SSTimeDim" in the SQL section (Figure 22). You can click the **Preview** button to view the database. Click Ok, to exit the Database Connection dialog box.

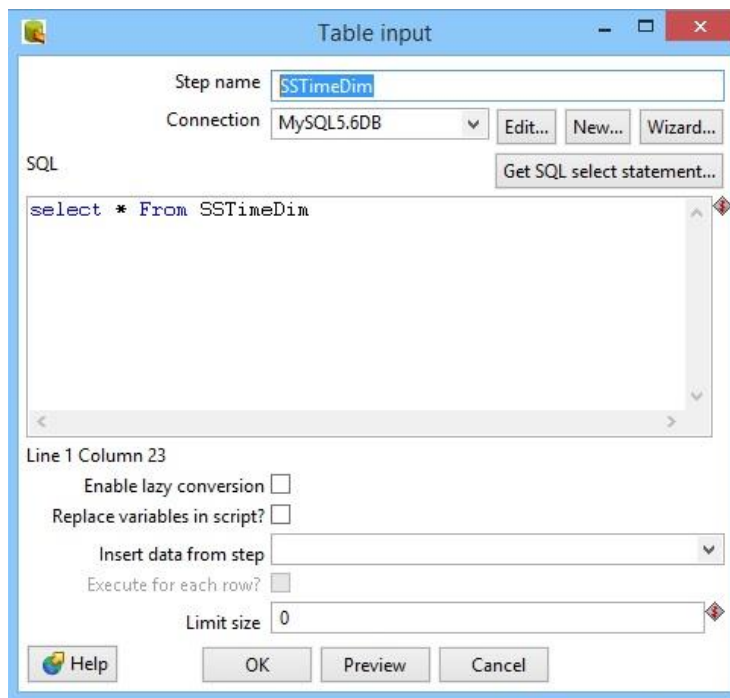


Figure 22: SQL Edit Section in Property Window of Table Input Node

- Add another sort rows component **Sort rows 2**, and a hop connecting the *SSTimeDim* step. In the field specification (Figure 23), delete other fields except TIMEDAY, TIMEMONTH, TIMEYEAR fields.

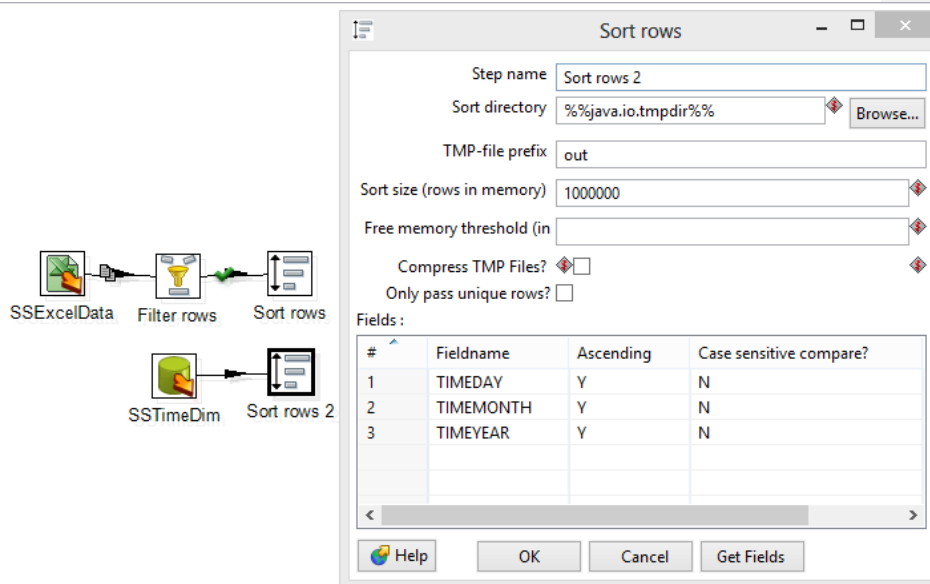


Figure 23: Property Edit Window of Sort Rows 2 Node

- Under the **Design** tab, expand the contents of the **Joins** node.
- Click and drag a **Merge Join** step into your transformation; create a hop between the **Sort rows**, **Sort rows 2** and **Merge Join** steps (Figure 24).

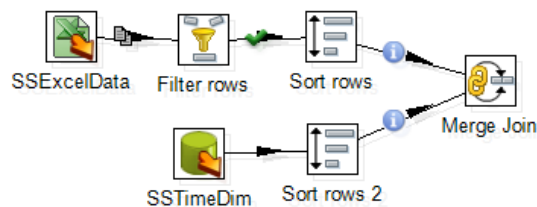


Figure 24: Two Sort Rows Nodes Connected to Merge Join Node

- Double-click the Merge Join step to specify its properties (Figure 25). Set **First step** as **Sort rows**, **Second step** as **Sort rows 2**, and **Join Type** as **INNER**. Click both of the “**Get key fields**” at left and right to get the possible fields to join. In the left table, delete other fields except Day, Month and Year fields. In the right table, delete other fields except *TIMEDAY*, *TIMEMONTH*, and *TIMEYEAR* fields. Then click OK.

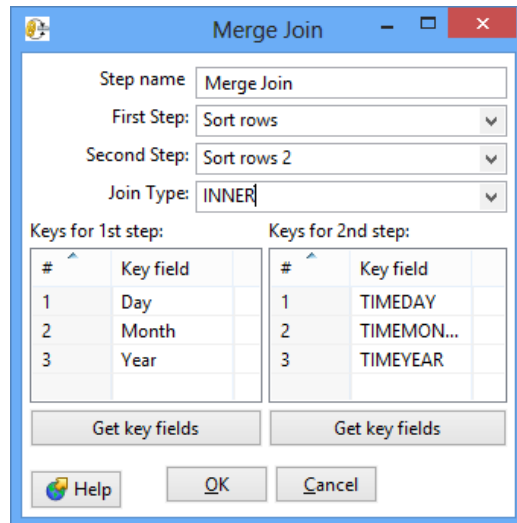


Figure 25: Property Edit Window of Merge Join Node

- Now, we have finished inner join between Excel input and *SSTimeDim* table.

Step 2 – Inner join the *SSItem*, *SSCustomer*, and *SSStore* tables.

Similar to getting data from the *SSTimeDim* table in the previous section, inner joining these tables requires **Table Input** components. First, we set the connection and query properties for the *SSItem* table. Note that these tables should exist in your MySQL schema before these steps.

- Drag and drop the **Table Input 2** into the design pane.
- Double click on the newly created component to open its Basic Settings pane. Specify the connection as shown in previous figure.
- Use “SSItem” as the Table Name value and “SELECT \* FROM SSItem” as the Query value.
- Create two **sort rows** components: **Sort rows 3** and **Sort rows 4**, connecting **Merge Join** and **SSItem** respectively. See the field to be sorted as: **ItemID** and **ITEMID** respectively.
- Drag and drop the **Merge Join 2** into the design pane. Connect **Sort rows 3** and **Sort rows 4** to **Merge Join 2**. Set the field to be joined as **Item ID** and **ITEMID**.
- The global view of all nodes and connections after Step 2 is shown by Figure 26.

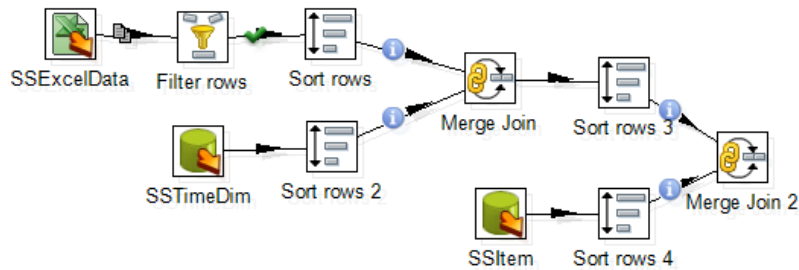


Figure 26: Global View of All Nodes and Connections after Step 2

### Step 3 – Inner join the tables.

- Inner join the tables named *SSCustomer* and *SSStore* in your transformation using the same method described previously.
- For the *SSCustomer* step, connect the *CustID* (from Excel file) and CUSTID (from Database) fields.
- For the *SSStore* step, connect the *StoreID* (from Excel file) and STOREID (from Database) fields.
- The global view of all nodes and connections after Step 3 is shown by Figure 27.

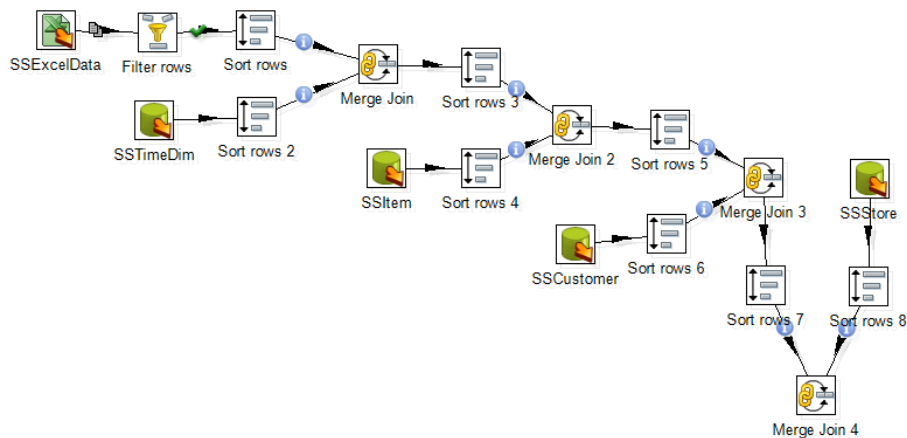


Figure 27: Global View of All Nodes and Connections after Step 3

## 5. Insert data into the SSSales table

- Under the **Design** tab, expand the contents of the **Output** node.
- Click and drag an **Insert/Update** step into your transformation; create a hop between the **Merge Join 4** and **Insert/Update** steps (Figure 28).

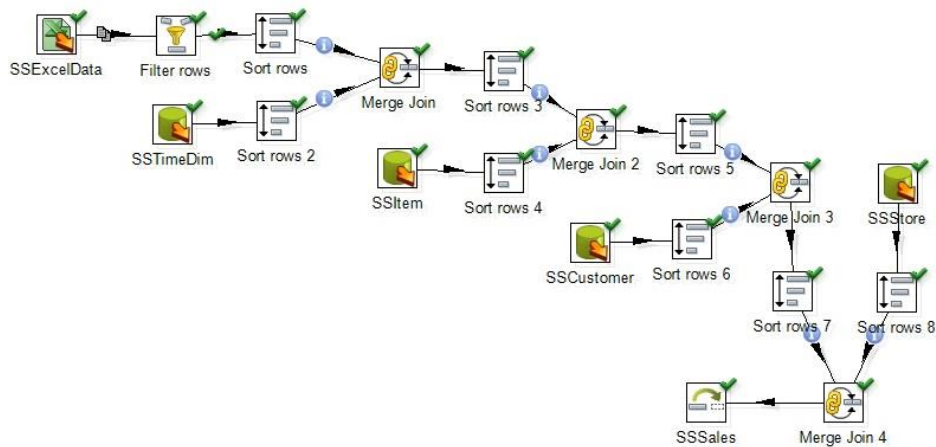


Figure 28: Connect Insert/Update Node to Last Merge Join Node

- Double click the **Insert/Update** component, to specify its properties (Figure 29). Set the **step name** as **SSSales**. Select the **connection** as **MySQL5.6DB**. Type in the **Target table** as **SSSales**. **DON'T** click the button “**Get fields**”. Instead, select the names from the two table fields and set the comparator between them to “**=**”. The final window should look like Figure 29.

Step name: SSSales

Connection: MySQL5.6DB [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: SSSales [Browse...]

Commit size: 100

Don't perform any updates: ☒

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	SalesUnits	=	SalesUnits	
2	SalesDollar	=	SalesDollar	
3	SalesCost	=	SalesCost	
4	CustID	=	CustID	
5	StoreID	=	StoreID	
6	ItemID	=	ItemID	
7	TIMENO	=	TIMENO	

[Get fields]

Update fields:

#	Table field	Stream field	Update
1			

[Get update fields]

[Edit mapping]

[Help] [OK] [Cancel] [SQL]

Figure 29: Property Edit Window of Insert/Update Node

- Click the button “**Get Updated fields**” and then click on “**Edit mapping**” button to edit mapping. The mapping edit window is shown by Figure 30. Select the fields named **SalesUnits**, **SalesDollar**, **SaleCost**, **CustID**, **StoreID**, **ItemID** and **TIMENO** into the **mappings** field. Pentaho will automatically match the corresponding name in the Target field. Then click **OK**.

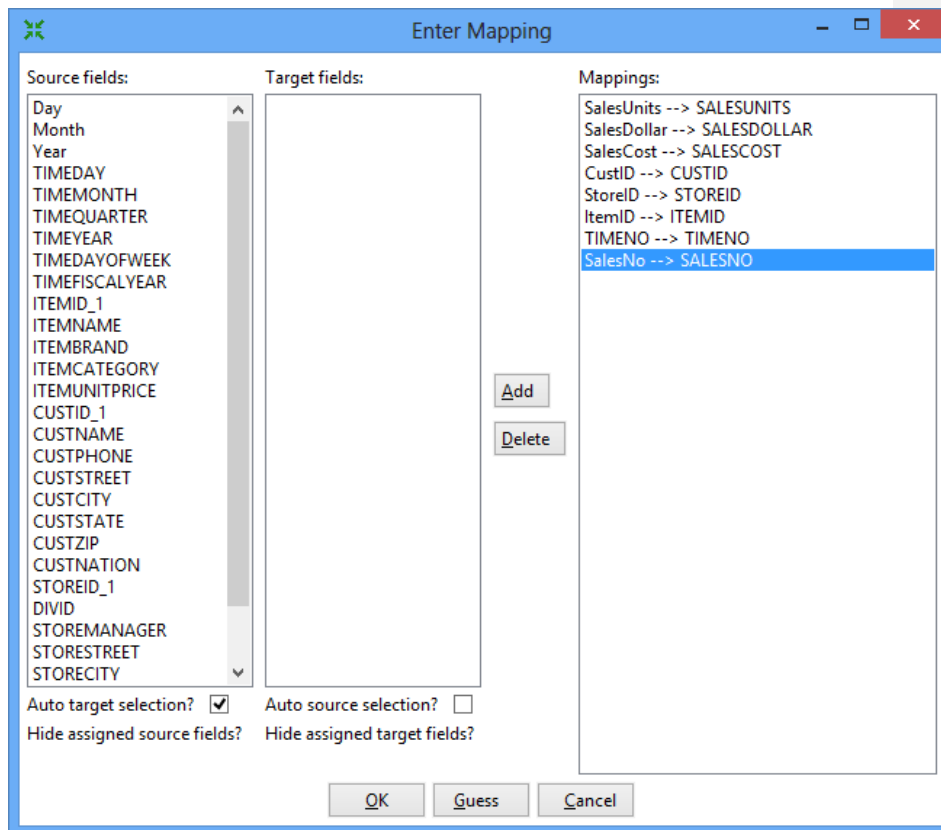


Figure 30: Mapping Edit Window

- The final view of the **SSSales** step will look like Figure 31



Step name: SSSales

Connection: MySQL5.6DB

Target schema:

Target table: SSSales

Commit size: 100

Don't perform any updates: ☒

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	SalesUnits	=	SalesUnits	
2	SalesDollar	=	SalesDollar	
3	SalesCost	=	SalesCost	
4	CustID	=	CustID	
5	StoreID	=	StoreID	
6	ItemID	=	ItemID	
7	TIMENO	=	TIMENO	

Update fields:

#	Table field	Stream field	Update
1	SALESUNITS	SalesUnits	N
2	SALES DOLLAR	SalesDollar	N
3	SALES COST	SalesCost	N
4	CUSTID	CustID	N
5	STOREID	StoreID	N
6	ITEMID	ItemID	N
7	TIMENO	TIMENO	N

Buttons: Help, OK, Cancel, SQL, Get fields, Get update fields, Edit mapping

Figure 31: Final view of the SSSales step

- Select the **SSSales** step and run a preview by clicking on . In the transformation debug dialog click on **Quick Launch** (Figure 32).

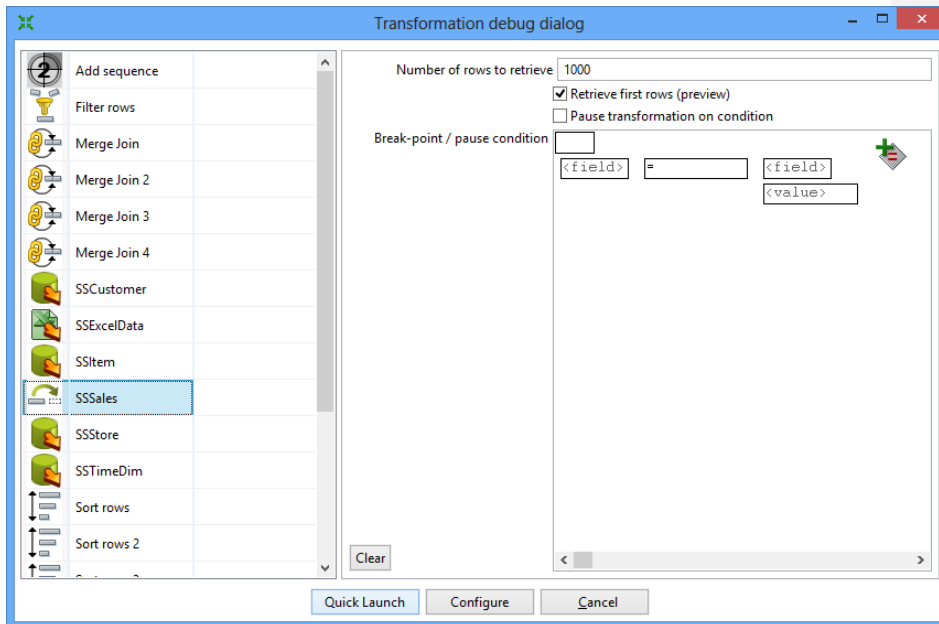


Figure 32: Transformation Debug Dialog

- The Examine preview data window is displayed by Figure 33.

#	SalesUnits	SalesDollar	SalesCost	CustID	StoreID	ItemID	Day	Month	Year
1	121	4224.	4224.	C0954327	S0954327	I0036566	1	5	2011
2	303	3003.	3003.	C9128574	S0954327	I0036566	1	2	2010
3	333	3333.	3333.	C9128574	S0954327	I0036566	1	5	2013
4	444	4444.	4444.	C9403348	S0954327	I0036577	1	2	2011
5	111	1111.	1111.	C0954327	S1010398	I0036577	1	2	2010
6	101	1001.	1001.	C0954327	S9432910	I0036577	3	7	2013
7	222	2222.	2222.	C8654390	S9432910	I0036566	3	7	2013
8	323	3223.	3223.	C9128574	S9432910	I0036577	3	7	2013

Figure 33: Execution Report Window

- Connect to your MySQL account so you can verify the number of rows in the *SSSales* table. You should see 104 rows with 8 new rows added to the 96 rows in the sample data (Figure 34).

	SALESNO	SALESUNITS	TIMENO	CUSTID	STOREID	ITEMID	SALES...	SALESC...
92	92	159	16	C9432910	S9432910	I0036577	6145	5387
93	93	105	13	C1010398	S1010398	I0036566	5455	4087
94	94	160	14	C0954327	S1010398	I0036566	6875	4996
95	95	90	15	C2388597	S1010398	I0036566	4805	4207
96	96	110	16	C8574932	S1010398	I0036566	5448	4188
97	97	121	6	C0954327	S0954327	I0036566	4224	4224
98	98	303	1	C9128574	S0954327	I0036566	3003	3003
99	99	333	14	C9128574	S0954327	I0036566	3333	3333
100	100	444	5	C9403348	S0954327	I0036577	4444	4444
101	101	111	1	C0954327	S1010398	I0036577	1111	1111
102	102	101	15	C0954327	S9432910	I0036577	1001	1001
103	103	222	15	C8654390	S9432910	I0036566	2222	2222
104	104	323	15	C9128574	S9432910	I0036577	3223	3223

Figure 34: Inserted Data in Oracle Database

- If you do not see the extra rows, the MySQL output component had a failure. To see the error, check the **Execution Results** section.

## 6. Load second data source from Access

The next part of the exercise involves creation of a new transformation to process the Access data source. Make sure that you have downloaded the Access database file from the class website and noted its location on your computer. You will begin by loading the data from a table in this database.

### Step 1- Add the Access Input Step

- Under the Design tab, expand the Input node. Figure 35 shows the Design table and input node.

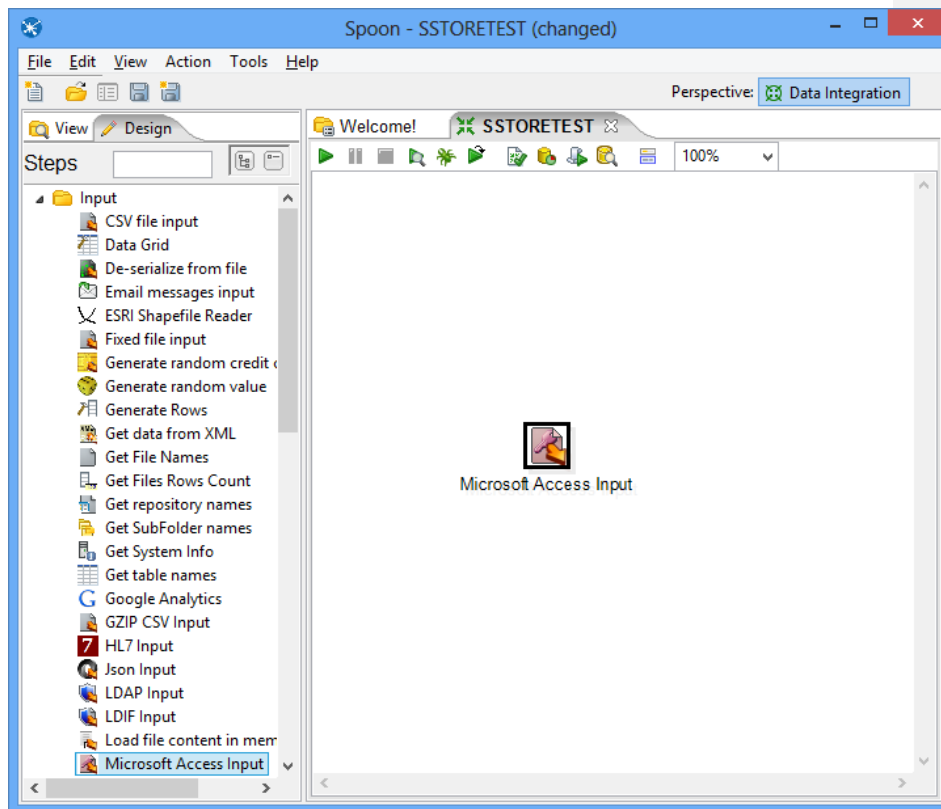


Figure 35: New Microsoft Access Input Node

- Select and drag a **Microsoft Access Input** step onto the canvas on the right;
- Double Click on the **Microsoft Access Input**. The edit properties dialog box associated with the **Microsoft Access Input** step appears (Figure 36). In this dialog box, you specify the properties related to a particular step.

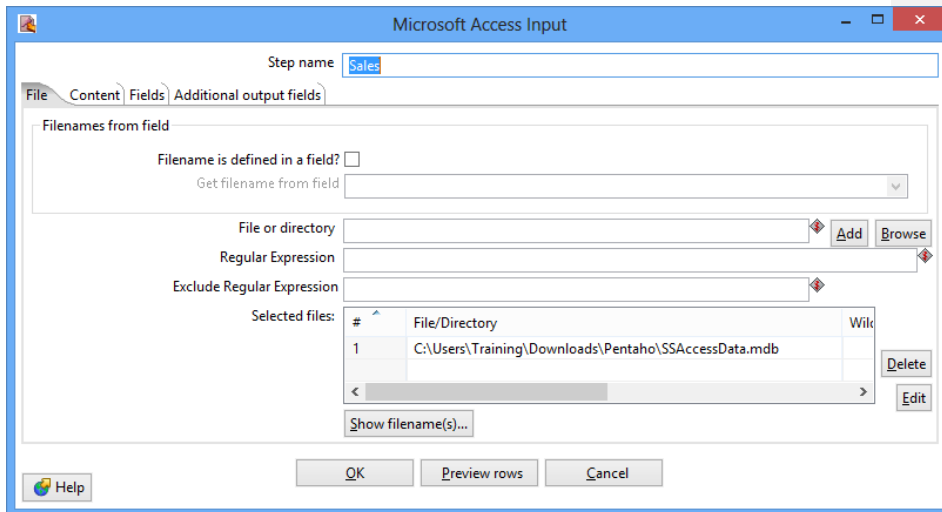


Figure 36: Property Edit Window of Microsoft Access Input Node

- Set name for the Access Input as **Sales** and specify the Excel data source path in the **Files** tab.
- In the tab named **Content**, click the button “**Get tables**” of **table** section. There will appear a window (Figure 37). Select **Sales** as the table name, click **OK**.

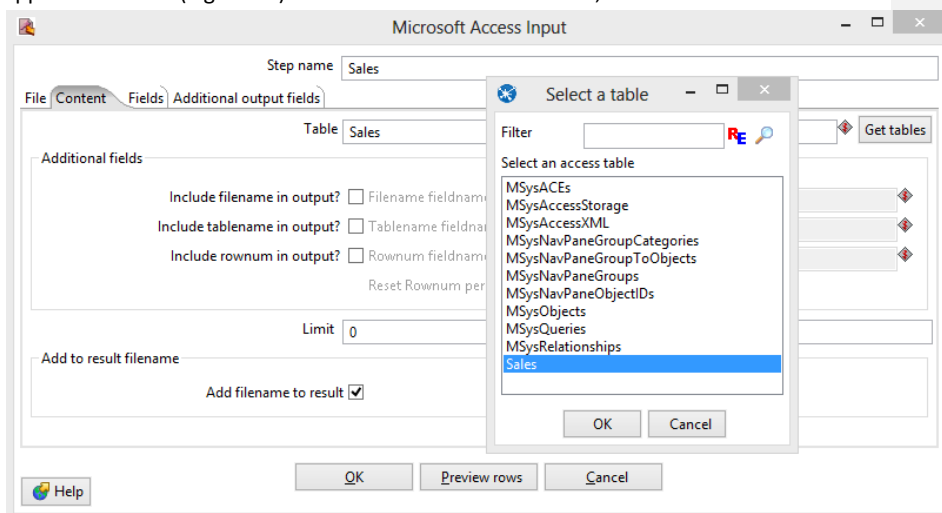


Figure 37: Table Selection Window

- In the tab named **Fields**, click the button “**Get fields**”. There will appear a list (Figure 38) showing the fields in the table named **Sales**.

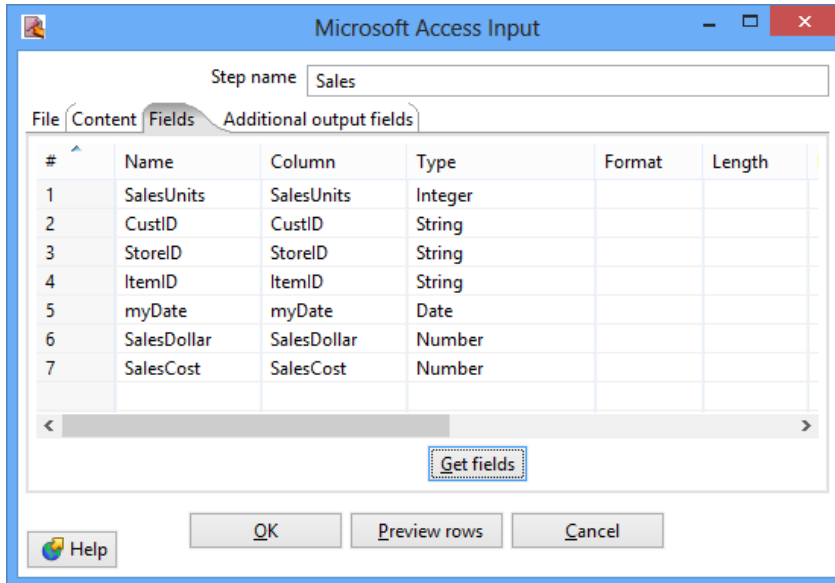


Figure 38: Fields Window for Microsoft Access Input Property Editing

- Click the button “**Preview rows**” to preview the database (Figure 39). When asked for the number of rows type 12 and click OK.

#	SalesUnits	CustID	StoreID	ItemID	myDate	SalesDollar	SalesCost
1	555	C1010398	S1010398	I0036566	2012/02/01 00:00:00.000	5555.0	5555.0
2	666	C8574932	S9432910	I0036577	2011/05/01 00:00:00.000	6666.0	6666.0
3	777	C0954327	S0954327	I0036566	2011/07/03 00:00:00.000	7777.0	7777.0
4	797	C0954327	S0954327	I0036566	2011/07/03 00:00:00.000	7997.0	7997.0
5	898	C1010398	S1010398	I0036566	2012/02/01 00:00:00.000	8998.0	8999.0
6	445	C1010398	S1010398	I0036566	2012/02/01 00:00:00.000	4455.0	5555.0
7	558	C9999999	S9432910	I0036577	2011/05/01 00:00:00.000	5885.0	6666.0
8	778	C0954327	S0954327	I0036566	2011/07/03 00:00:00.000	9997.0	9997.0
9	665	C8574932	<null>	I0036577	2011/05/01 00:00:00.000	6665.0	6666.0
10	112	C0954327	S0954327	I0036566	2011/07/03 00:00:00.000	1112.0	7777.0
11	556	C0954327	S0954327	<null>	2011/07/03 00:00:00.000	5656.0	7777.0
12	996	C1010398	S1010398	I0036566	2015/02/01 00:00:00.000	9669.0	5555.0

Figure 39: Examine Preview Data Window

- Click **OK** at the bottom of the window. The input icon will change to the shape shown by Figure 40.



Figure 40: Sales Node Icon

Step 2 –You will add constraint checking for null values using the Filter Rows step.

- Add a Filter Rows step to your transformation. Under the **Design** table, go to **Flow** → **Filter Rows** (Figure 41).

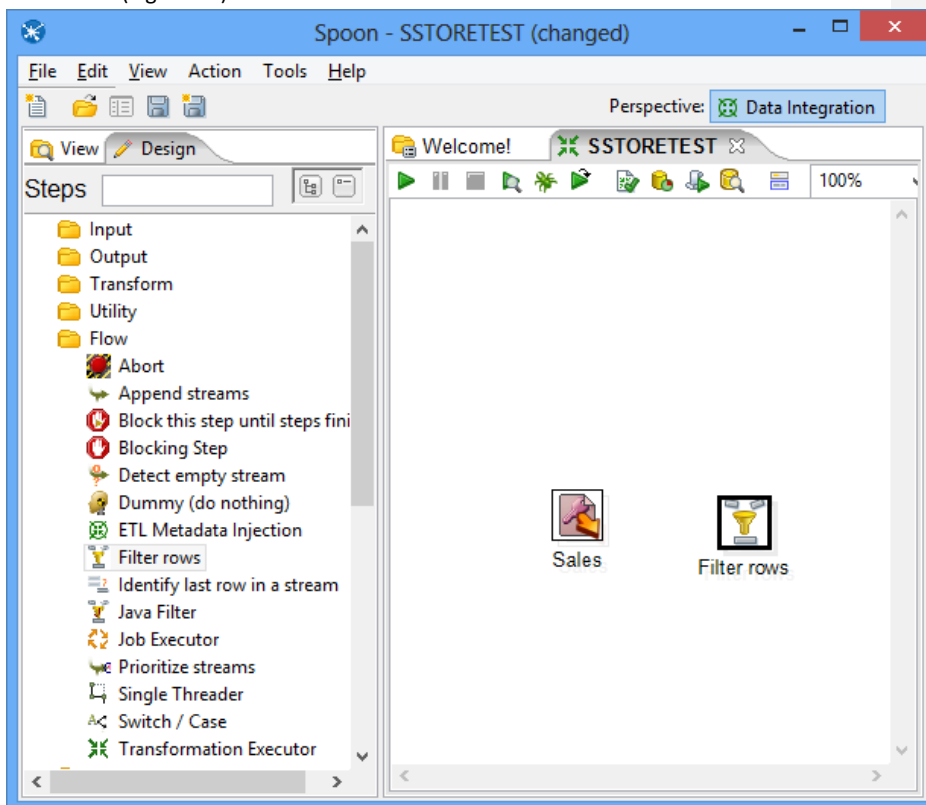


Figure 41: Access Input Node and Filter Node in Spoon

- Create a hop between the **Sales** (Access file input) step and the **Filter Rows** step. Hops are used to describe the flow of data in your transformation. To create the hop, click the

**Sales** (Access file input) step, then press the <SHIFT> key down and draw a line to the Filter Rows step.

- Alternatively, you can draw hops by hovering over a step until the hover menu appears. Drag the hop painter icon from the source step to your target step.
- Double-click the **Filter Rows** step. The **Filter Rows** edit properties dialog box appears.
- In the **Step Name** field type, **Filter rows**.
- The configuration of this step is similar to what you did in the previous excel transformation.
- The final view of filter conditions is shown by Figure 42.

**Commented [MM1]:** Need some more explanation and possibly a snapshot

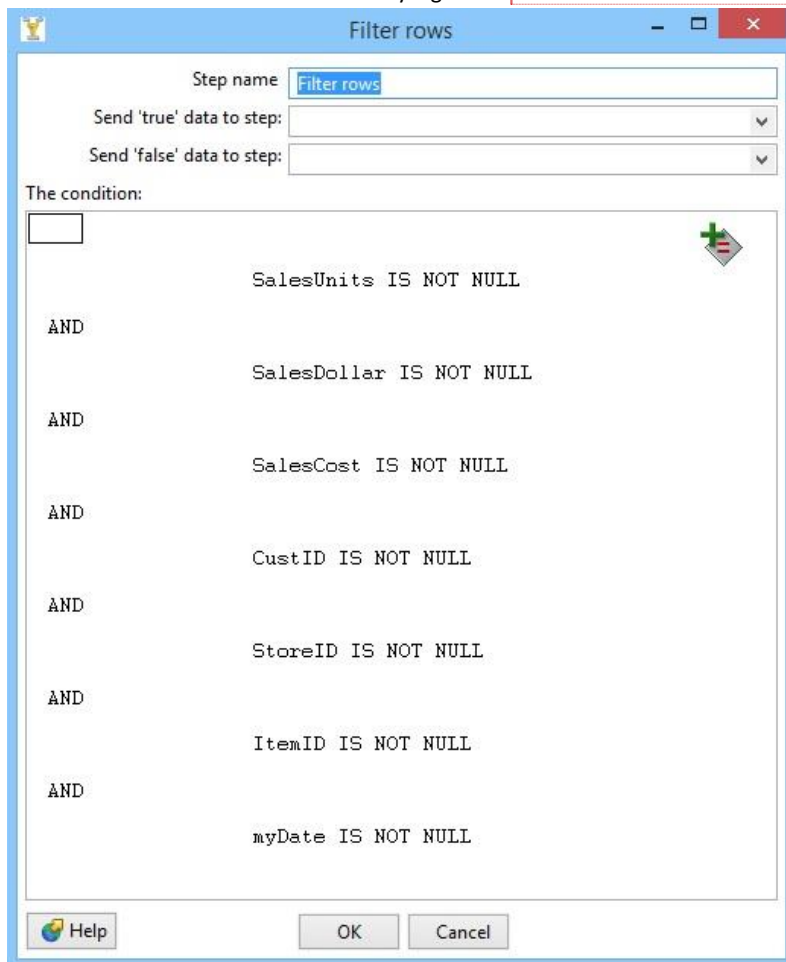


Figure 42: Filter Conditions Window



- Save your transformation.

## 7. Separate SalesDay fields into Day, Month, Year fields

In this part of the tutorial, you will use the Select Values step to change the format of the myDate field and the Split Fields step to parse the field into date components.

- Under the **Design** tab, expand the contents of the **Transform** node.
- Click and drag a **Select values** step into your transformation.
- Create a “hop” between the **Filter rows** step and the **Select values** step (Figure 43).  
Select **Result is TRUE** in the filter results selection list

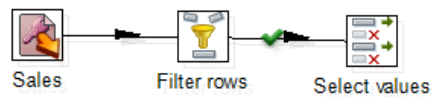


Figure 43: True Filter Results Connected to Select Values Node

- Double-click the Select values step to open its edit properties dialog box.
- In the tab named Metadata, click the button “**Get fields to change**”, to get the fields to change, which is shown by Figure 44. Change the **Type** of field **myDate** as **String**, change its **Format** as dd-MM-yyyy. Click **OK**.

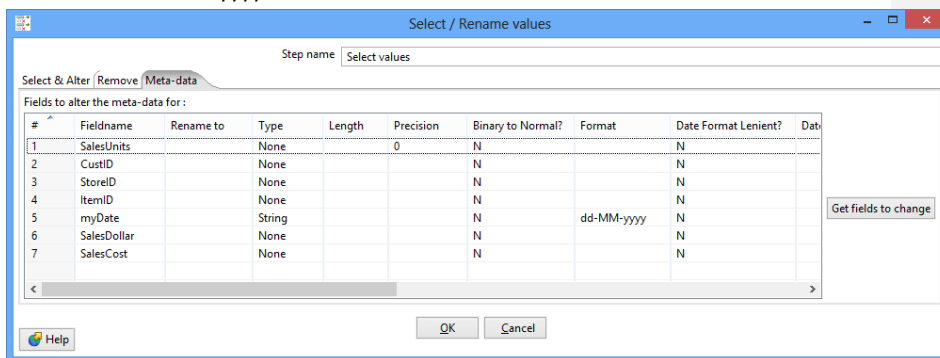


Figure 44: Meta-data Tab of Select Values Property Edit Window

- Under the **Design** tab, expand the contents of the **Transform** node.
- Click and drag a **Split fields** step into your transformation (Figure 45).




Figure 45: Create Split Fields in Spoon

- Create a “hop” between the **Select values** step and the **Split fields** step.
- Double-click the **Split fields** step to open its edit properties dialog box (Figure 46).
- Select **myDate** in the **Field to split**, type “-” as the **Delimiter**. Type in **Year, Month** and **Day** in the Column named **New field**, and set their **Type** as **Number**.

#	New field	ID	Remove ID?	Type
1	Day		N	Number
2	Month		N	Number
3	Year		N	Number

Figure 46: Property Edit Window of Field Splitter Node

- Click OK.
- Click  , to preview this transform (Figure 47). Make sure that Split Fields step is selected from the left side panel of the transformation debug dialog and click on “**Quick Launch**” button.

#	SalesUnits	CustID	StoreID	ItemID	Day	Month	Year	SalesDollar	SalesCost
1	555	C1010398	S1010398	I0036566	1.0	2.0	2012.0	5555.0	5555.0
2	666	C8574932	S9432910	I0036577	1.0	5.0	2011.0	6666.0	6666.0
3	777	C0954327	S0954327	I0036566	3.0	7.0	2011.0	7777.0	7777.0
4	797	C0954327	S0954327	I0036566	3.0	7.0	2011.0	7997.0	7997.0
5	898	C1010398	S1010398	I0036566	1.0	2.0	2012.0	8998.0	8999.0
6	445	C1010398	S1010398	I0036566	1.0	2.0	2012.0	4455.0	5555.0
7	558	C9999999	S9432910	I0036577	1.0	5.0	2011.0	5885.0	6666.0
8	778	C0954327	S0954327	I0036566	3.0	7.0	2011.0	9997.0	9997.0
9	112	C0954327	S0954327	I0036566	3.0	7.0	2011.0	1112.0	7777.0
10	996	C1010398	S1010398	I0036566	1.0	2.0	2015.0	9669.0	5555.0

Figure 47: Examine Preview Data Window

## 8. Lookup Columns from the MySQL tables

This part of the exercise involves looking up the date from the *SSTimeDim* table to check the validity of dates in the Access data source. In addition, you will lookup primary key columns from other Oracle tables to ensure loaded data does not contain invalid foreign keys. This part of the exercise is similar to Section 3.

Step 1 – Access the *SSTimeDim* table from MySQL database.

- Under the **Design** tab, expand the contents of the **Input** node.
- Click and drag a **Table Input** step into your transformation.
- Double-click the Table Input step to open its edit properties dialog box.
- Rename your Table Input step to *SSTimeDim*.
- Click **"New"** next to the connection field. You must create a connection to the database. The Database connection dialog box appears.
- Provide the settings for connecting to the database as shown in the Figure 20.
- Connection Name: MySQL5.6DB  
 Connection Type: MySQL  
 Access: Native (JDBC)  
 Host Name: localhost  
 Database Name: (This should be your database name)  
 Port Number: 3360  
 User name: (This should be your user name)  
 Password: (This should be your password)

- Click **“Test”**, to test the connection.
- Type in **“SELECT \* FROM SSTimeDim”** in the SQL section. You can click the **Preview** button to view the database. Click **Ok**, to exit the Database Connection dialog box.
- Under the **Design** tab, expand the contents of the **Transform** node.
- Click and drag a **Sort Rows** step into your transformation; create a hop between the **Split fields** and **Sort Rows** steps.
- Double-click the **Sort Rows** step to open its edit properties dialog box. Click **“Get fields”** to obtain the fields. Delete other fields except the Day, Month and Year fields. Then click **Ok**.
- Add one more sort rows component **Sort rows 2**, and a hop connecting the *SSTimeDim* step. In the field specification, delete other fields except *TIMEDAY*, *TIMEMOHTH*, *TIMEYEAR* fields.
- Under the **Design** tab, expand the contents of the **Join** node.
- Click and drag a **Merge Join** step into your transformation; create a hop between the **Sort rows**, **Sort rows 2** and **Merge Join** steps.
- Double-click the Merge Join step to specify its properties. Set **First step** as **Sort rows**, **Second step** as **Sort rows 2**, and **Join Type** as **INNER**. Click both of the **“Get key fields”** at left and right to get the possible fields to join. In the left table, delete other fields except Day, Month and Year fields. In the right table, delete other fields except *TIMEDAY*, *TIMEMONTH*, and *TIMEYEAR* fields. Then click **OK**.
- Now, we have finished inner join between the Access table and *SSTimeDim* table.
- Figure 48 shows the global view of all nodes and connections after Step 1.

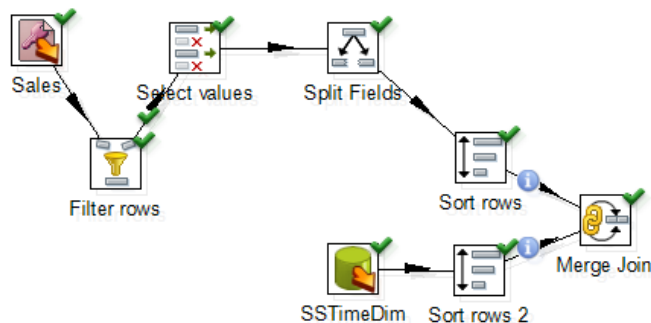


Figure 48: Global View of All Nodes and Connections after Step 1

Step 2 – Inner join *SSItem*, *SSCustomer*, and *SSStore* to Access table.

- Inner join the tables named *SSItem*, *SSCustomer*, and *SSStore* in your transformation using the same method described before.
- For *SSItem* step, connect *ItemID* (from Excel file) and *ITEMID* (from Database) fields.
- For *SSCustomer* step, connect *CustID* (from Excel file) and *CUSTID* (from Database) fields.
- For *SSStore* step, connect *StoreID* (from Excel file) and *STOREID* (from Database) fields.
- Figure 49 shows the global view of all nodes and connections after Step 2.

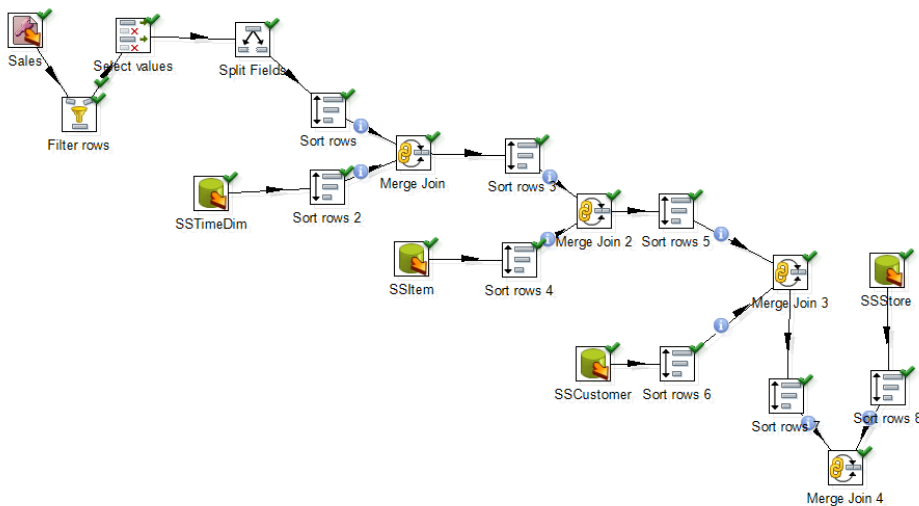


Figure 49: Global View of All Nodes and Connections after Step 2

## 9. Insert data into the SSSales table

- Under the **Design** tab, expand the contents of the **Output** node.
- Click and drag an **Insert/Update** step into your transformation; create a hop between the **Merge Join 4** and **Insert/Update** steps. Figure 50 shows the connection.
- Double click the **Insert/Update** component, to specify its properties. Set the **step name** as **SSSales**. Select the **connection** as **MySQL5.6DB**. Type in the **Target table** as **SSSales**. **DON'T** click the buttons "**Get fields**". Instead, select the names from the two table fields and set the comparator between them to "**=**". The final window should look like Figure 31.

- Click the button “**Get Updated fields**” and then click on “**Edit mapping**” button to edit mapping. The mapping edit window is shown by Figure 32. Select the fields named **SalesUnits**, **SalesDollar**, **SaleCost**, **CustID**, **StoreID**, **ItemID** and **TIMENO** into the **mappings** field. Pentaho will automatically match the corresponding name in the Target field. Then click **OK**.

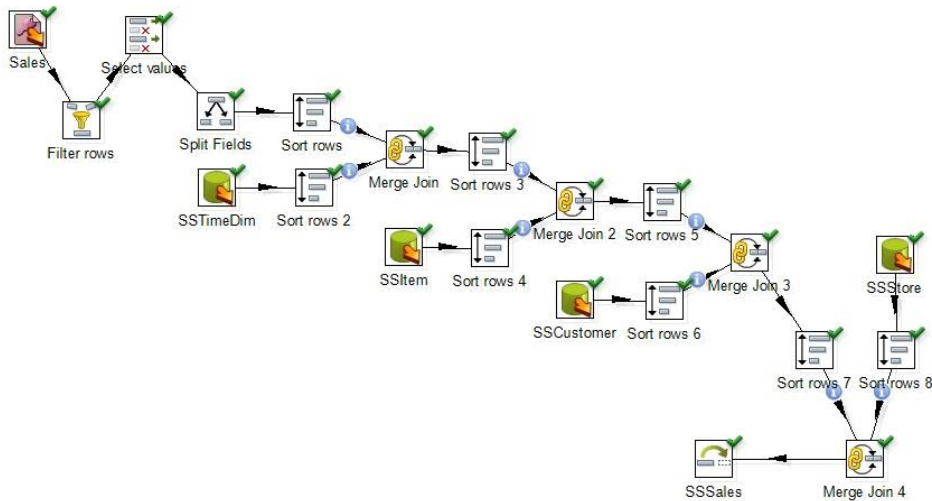



Figure 50: Connect Insert/Update Node to Last Merge Join Node

- Select the **SSSales** step and run a preview by clicking on . In the transformation debug dialog click on **Quick Launch** (Figure 32).
- The Examine preview data window is displayed like Figure 33.

Connect to your MySQL account so you can verify the number of rows in the *SSSales* table. You should see 112 rows with 8 new rows added to the 104 rows in the sample data (Figure 51).









	 SALESNO	 SALESUNITS	 TIMENO	 CUSTID	 STOREID	 ITEMID	 SALES...	 SALESC...
92	92	159	16	C9432910	S9432910	I0036577	6145	5387
93	93	105	13	C1010398	S1010398	I0036566	5455	4087
94	94	160	14	C0954327	S1010398	I0036566	6875	4996
95	95	90	15	C2388597	S1010398	I0036566	4805	4207
96	96	110	16	C8574932	S1010398	I0036566	5448	4188
97	97	121	6	C0954327	S0954327	I0036566	4224	4224
98	98	303	1	C9128574	S0954327	I0036566	3003	3003
99	99	333	14	C9128574	S0954327	I0036566	3333	3333
100	100	444	5	C9403348	S0954327	I0036577	4444	4444
101	101	111	1	C0954327	S1010398	I0036577	1111	1111
102	102	101	15	C0954327	S9432910	I0036577	1001	1001
103	103	222	15	C8654390	S9432910	I0036566	2222	2222
104	104	323	15	C9128574	S9432910	I0036577	3223	3223
105	105	777	7	C0954327	S0954327	I0036566	7777	7777
106	106	797	7	C0954327	S0954327	I0036566	7997	7997
107	107	778	7	C0954327	S0954327	I0036566	9997	9997
108	108	112	7	C0954327	S0954327	I0036566	1112	7777
109	109	555	9	C1010398	S1010398	I0036566	5555	5555
110	110	898	9	C1010398	S1010398	I0036566	8998	8999
111	111	445	9	C1010398	S1010398	I0036566	4455	5555
112	112	666	6	C8574932	S9432910	I0036577	6666	6666

Figure 51: Inserted Data in Oracle Database