Data Integration using Pentaho Data Integration

This tutorial will give you basic idea of Pentaho Data Integration Tool and how to use it to create transformations with step by step instructions. We will use two data sources to transform it and then load data into an Oracle table. we will perform transformations to parse date strings, combine fields, and perform validation checks. The two data sources provide new data for the *SSSales* table of the Store Sales data warehouse example. Thus, we need to create the Store Sales tables and sequences and load the sample data.

The instructions in the exercise demonstrate connection to my account on the Oracle database. Make sure that java is installed in your machine before you start. You can obtain a free copy of java from <http://www.java.com>.

You might want to download the input files (Excel file and Access database file) attached with the repository. You will use these input files in the beginning steps of the two job designs that you will create.

For this tutorial, we are going to use an open source version of Pentaho called community edition (CE). I wrote the tutorial for stable version is (5.0.1), available in <http://www.sourceforge.com/> website. Figure 1 shows the launch page of the stable version.

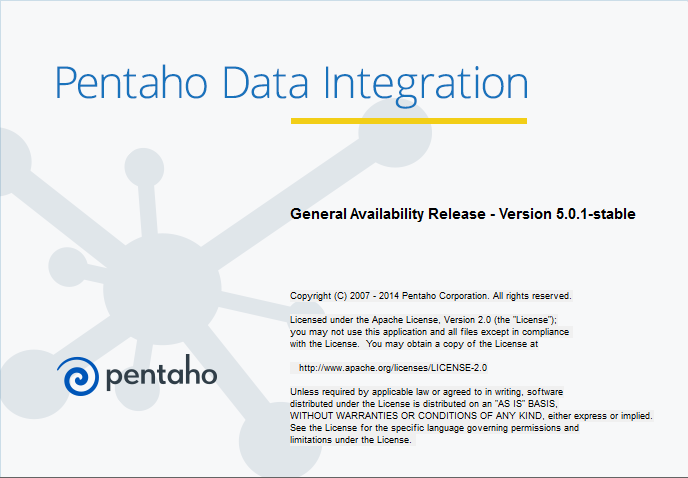


Figure 1: Pentaho Data Integration Welcome Window

To install Pentaho, you should follow the steps below. It is highly recommended that you use the community edition from SourceForge as the instructions in this document follow the community edition interface.

* I wrote the tutorial for stable version 5.0.1 although the latest version on the community edition page (<http://community.pentaho.com/projects/data-integration/>) is 7.0. You should be able to use either version to complete the tutorial and assignment.
* You can use this link to go directly to the Pentaho Data Integration download page. You can download the version 5.0.1 or the latest version version (7.0) using the appropriate folder.
  + https://sourceforge.net/projects/pentaho/files/Data%20Integration/
* Download the **pdi-ce-5.0.1A-stable.zip** or the latest version. You need to click on the file name to download it. If you use the Download button, you will download the latest version.
* Unzip the downloaded zip file to any folder.
* Windows users should copy the folder data-integration to the folder C:\Program Files\Pentaho. Mac and Linux users may move the file to any folder.

To ensure that the installation worked, you should launch Pentaho Data Integration.

* Run the file Spoon.bat by double clicking on it. You may want to create a shortcut to the spoon.bat file so starting data integration is easier. If you get a permission error or cannot execute the bat file, you should right click and select “Run as Administrator”. For Mac and Linux users, run the Spoon.sh from terminal (./spoon.sh).
* I have encountered a problem with executing the spoon.bat file on one of my computers. The spoon.bat file executes, but the data integration console does not load because of an error or incompatibility in the spoon.bat file. This problem may only involve the latest version (7.0) and Java 1.8. This page provides a fix that worked for me. You need to change the Xmx parameter value in the spoon.bat file from “Xmx2048m” to “Xmx1024m”. After you search for Xmx in the spoon.bat file and change the value, you should save the spoon.bat file.
  + http://stackoverflow.com/questions/34238611/not-able-to-start-pdi-gui-from-spoon-bat-in-pdi-community-edition
* After you launch Pentaho Data Integration, you will see the Welcome window (Figure 1) and then the Spoon designer (Figure 2).
* Exit Spoon before installing the database driver file in the next part of the instructions.

After you launch Pentaho Data Integration, the Spoon designer is launched at the same time (Figure 2). *Spoon* provides a graphical interface that supports creation of transformations (data flows) and jobs (execution sequences) as well as execution and testing of Pentaho Data Integration processes. Spoon builds jobs and transformations and can save them as database repository and files.

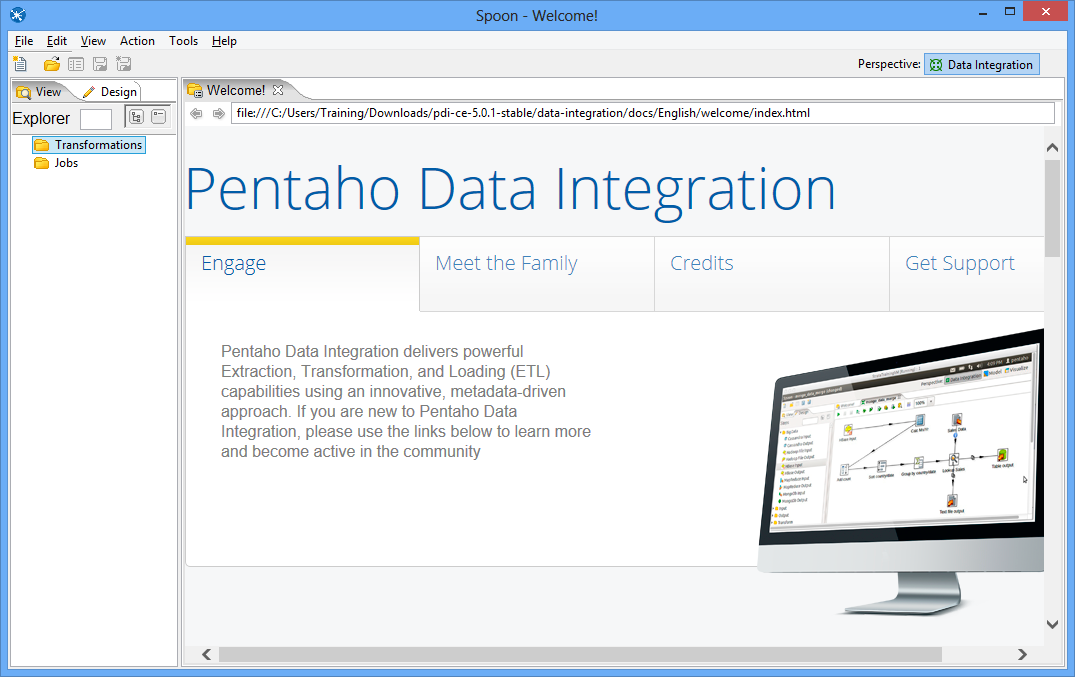


Figure 2: Spoon Opening Window

# 1. Managing Database Connections

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 the Database Connectionstep in the explorer View in Spoon.

There are several ways to define a new database connection:

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

You will configure the database connection later in this tutorial.

**Adding a JDBC Driver**

Before you can connect to a data source in any Pentaho server or client tool, you must first install the appropriate database driver. After you have installed the driver, follow the instructions below to copy it to the driver directories for all of the Business Analytics components that need to connect to this data source.

Various JDBC drivers can be downloaded from: <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>

The Business School uses Oracle 12c and its relevant JDBC Driver is “ojdbc7.jar”. It can be downloaded from the link on this page:

* <http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html>.
* Copy the driver JAR file to the following directory: C:\Program Files\Pentaho\data-integration\lib
* After copying the JDBC file to the specified folder, you should restart Pentaho Data Integration.

# 2. Creating your first transformation

The Data Integration component of Spoon allows you to create two basic document types: transformations and jobs. Transformations are used to describe data flows such as reading from a source, transforming data and loading it into a target location. Jobs are used to coordinate data integration activities such as defining the flow and dependencies for what order transformations should be run, or prepare for execution by checking 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 1) and the Spoon window (Figure 2).

2. Click  (New) in the upper left corner of the Spoon window.

3. Select **Transformation** from the list of components (Figure 3) displayed after selecting the **New** button.

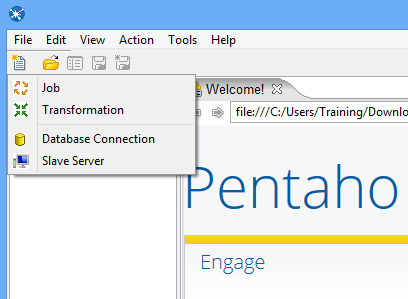


Figure 3: Spoon Transformation List

# 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 4).

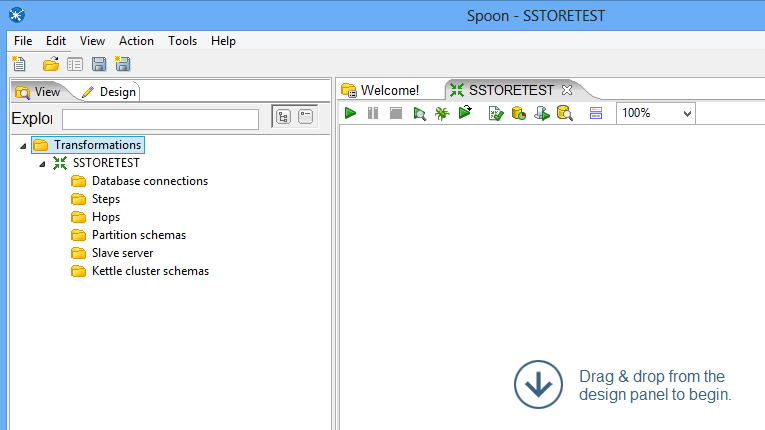


Figure 4: Empty Transformation Window

Step 4 – Create the Excel Input step:

* Under the Design tab, expand the Input step (Figure 5).

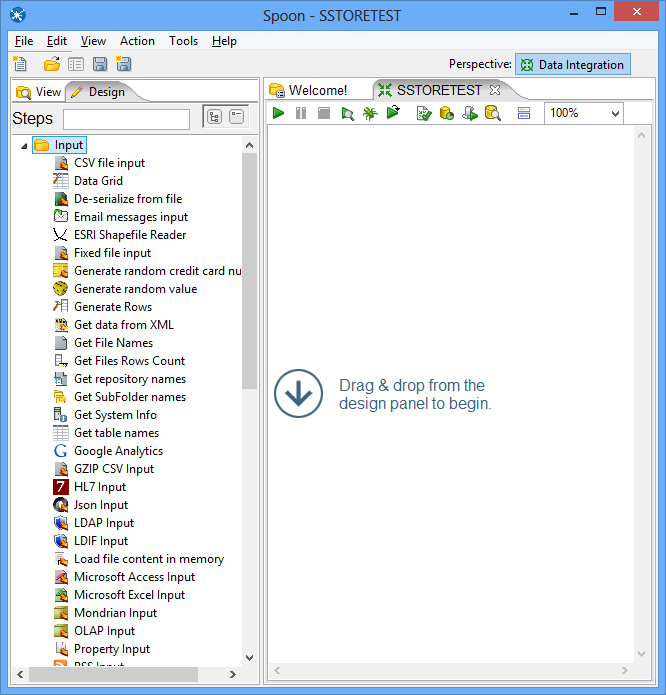


Figure 5: New Microsoft Excel Input Step

* 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 6) associated with the **Microsoft Excel Input** step appears. In this dialog box, you specify the properties related to a particular step.

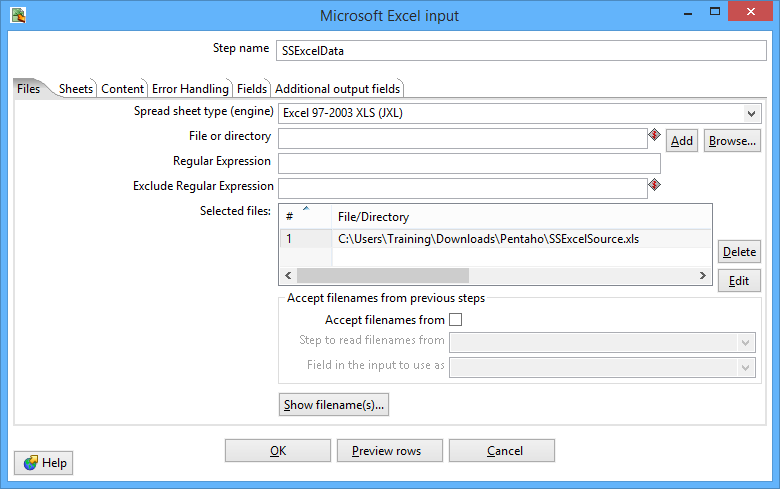


Figure 6: Files Window for Microsoft Excel Input Property Editing

* Set name for the Excel Input as **SSExcelData** 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 7) 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 8.

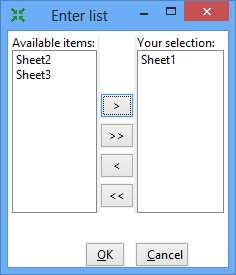


Figure 7: Sheet Specification Window

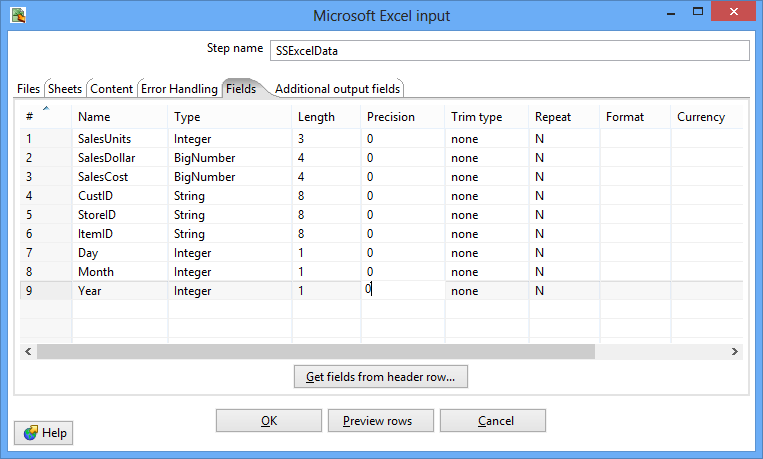


Figure 8: Fields Window for Microsoft Excel Input Property Editing

* Click **OK** at the bottom of the window. The input icon will change to the icon displayed in Figure 9.

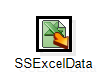


Figure 9: SSExcelData Icon

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).

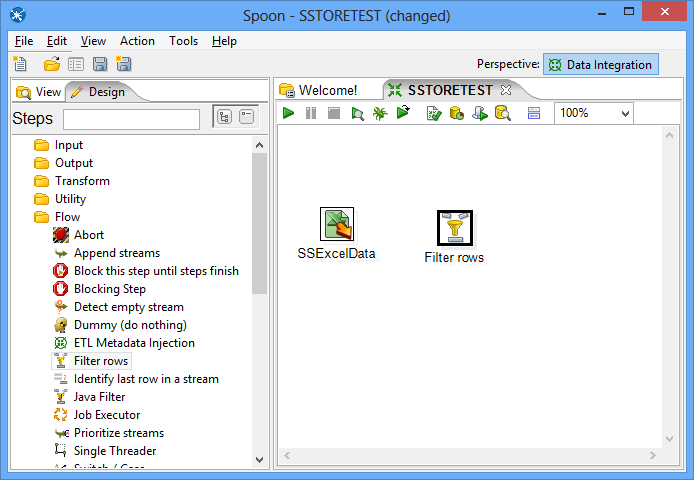


Figure 10: Excel Input Step and Filter Step in Spoon

* Create a “hop” between the **SSExcelSource** (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 **SSExcel Source** (Excel file input) step, then press the <**SHIFT**> key down and draw a line to the Filter Rows step (Figure 11).

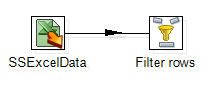


Figure 11: Hop connecting an Excel Input Step Connected to a Filter Step

* 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.

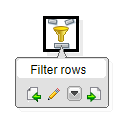


Figure 12: Hover Menu

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

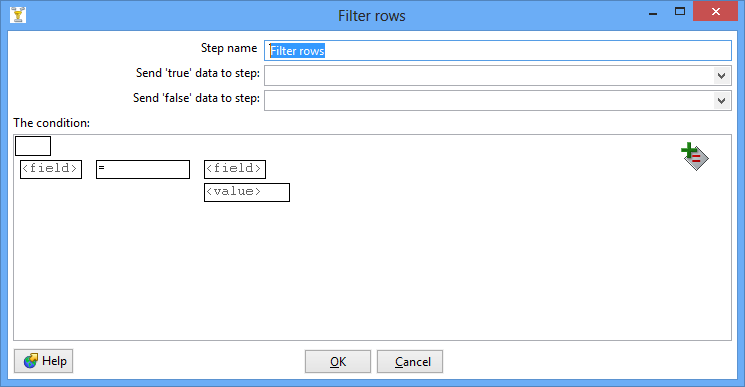


Figure 13: Property Edit Window of Filter Step

* 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**.

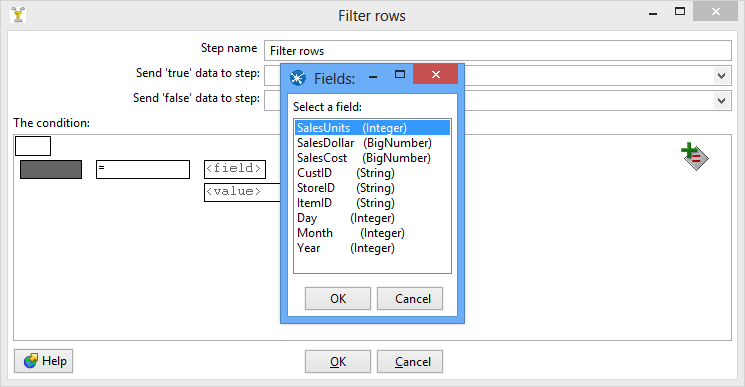


Figure 14: Condition Fields Selection Window

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

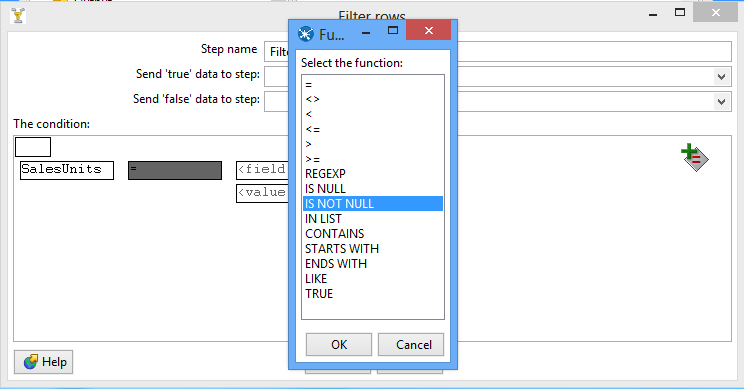


Figure 15: Comparison Operator List

* 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.

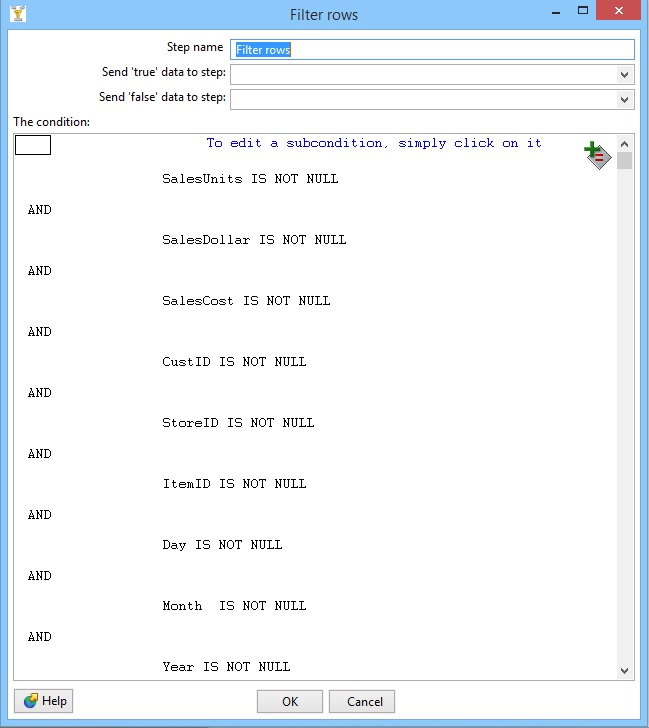


Figure 16: Filter Conditions Window

* 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** step.
* 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).

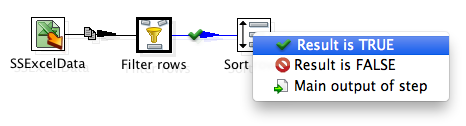


Figure 17: Filter Results Selection List

* 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.

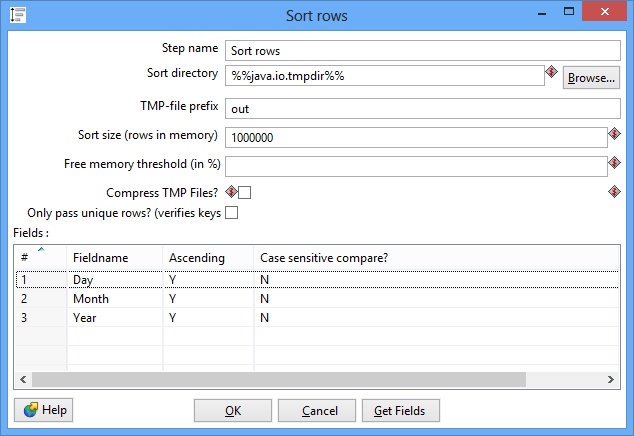


Figure 18: Property Edit Window of Sort Rows Step

# 4. Lookup Columns from the Oracles tables

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 Oracle tables to ensure loaded data does not contain invalid foreign keys.

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

* Under the **Design** tab, expand the contents of the **Input** step.
* 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*.

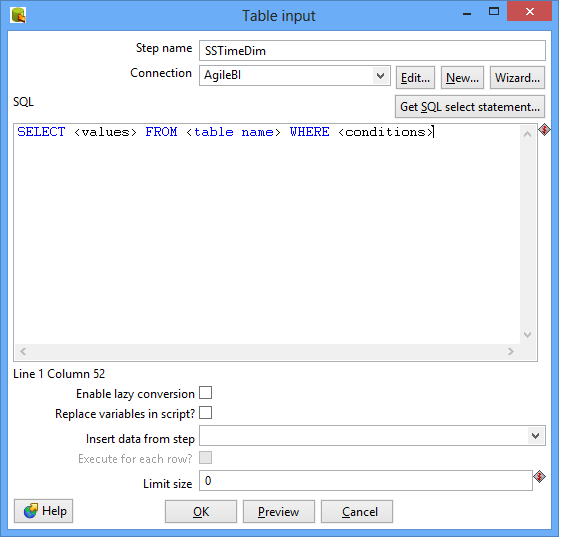


Figure 19: Property Edit Window of Table Input Step

* 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.

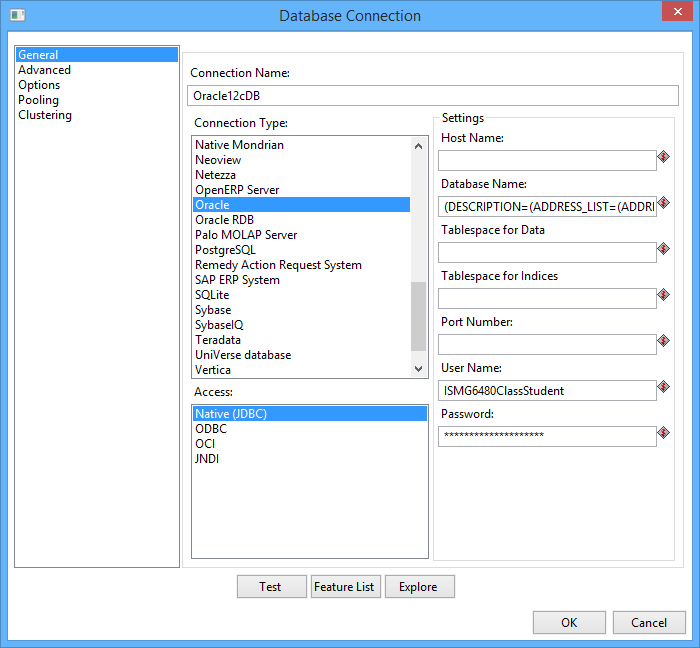


Figure 20: Database Connection Window

* Click “Test” to test the connection. Then success test result is shown by Figure 21.

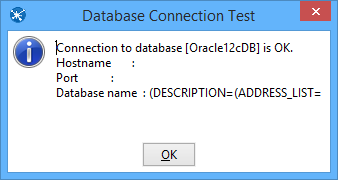


Figure 21: Database Connection Test

* 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.

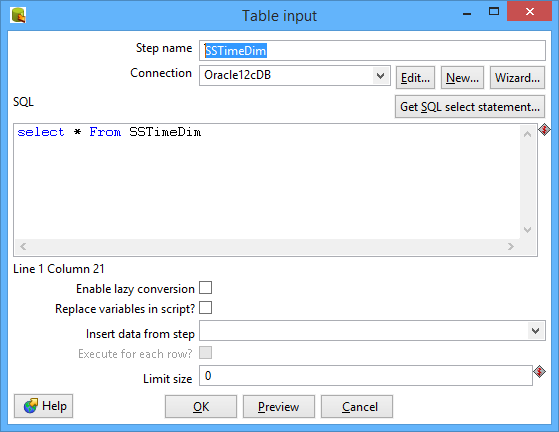


Figure 22: SQL Edit Section in Property Window of Table Input Step

* 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, TIMEMOHTH, TIMEYEAR fields.

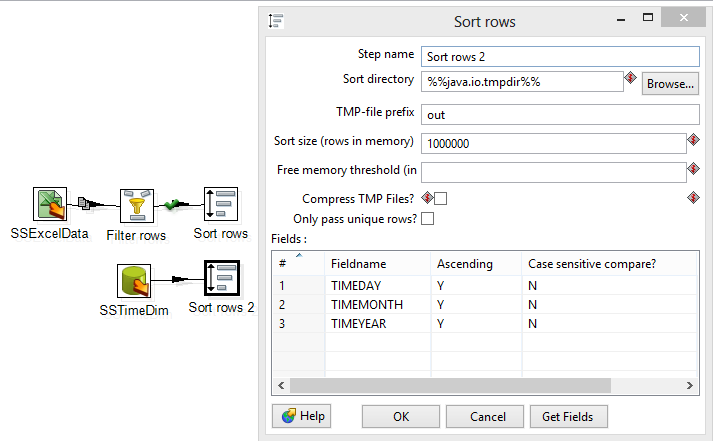


Figure 23: Property Edit Window of Sort Rows 2 Step

* Under the **Design** tab, expand the contents of the **Joins** step.
* 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).

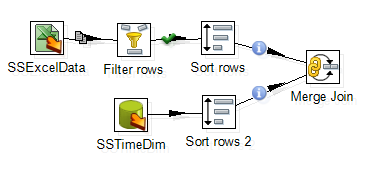


Figure 24: Two Sort Rows Steps Connected to Merge Join Step

* 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.

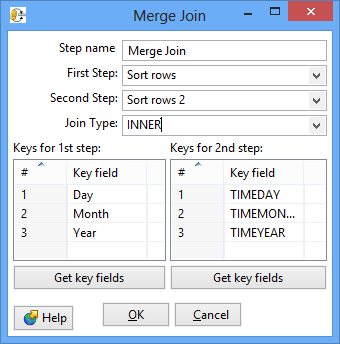


Figure 25: Property Edit Window of Merge Join Step

* 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 Oracle 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 steps and connections after Step 2 is shown by Figure 26.

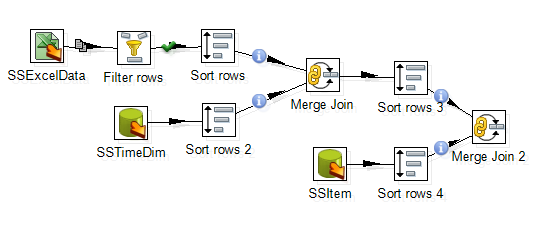


Figure 26: Global View of All Steps and Connections after Step 2

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 steps and connections after Step 3 is shown by Figure 27.

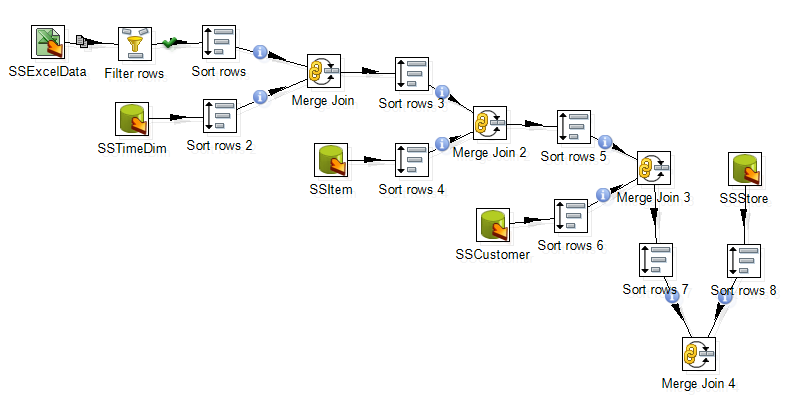


Figure 27: Global View of All Steps and Connections after Step 3

Step 4 – Create and connect an Add Sequence step to generate values for the *SalesNo* column.

* Under the **Design** tab, expand the contents of the **Transform** step.
* Click and drag an **Add sequence** step into your transformation; create a hop between the **Merge Join 4** and **Add Sequence** steps (Figure 28). To create the hop, click the **Merge Join 4** step, then hold the <**SHIFT**> key down and draw a line to the **Add Sequence** step.
* Double click on the newly created component to open its Basic Settings pane.
* Set **SalesNo** as the name of value. Check the box for use DB to get sequence. Select the **connection** as **Oracle12cDB.** Set **SSSalesNoSeq** as sequence name (Figure 29).

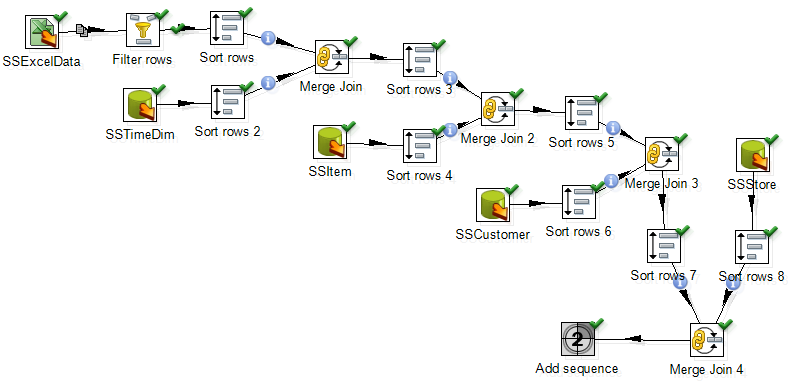


Figure 28: Global View of All Steps and Connections after Step 4

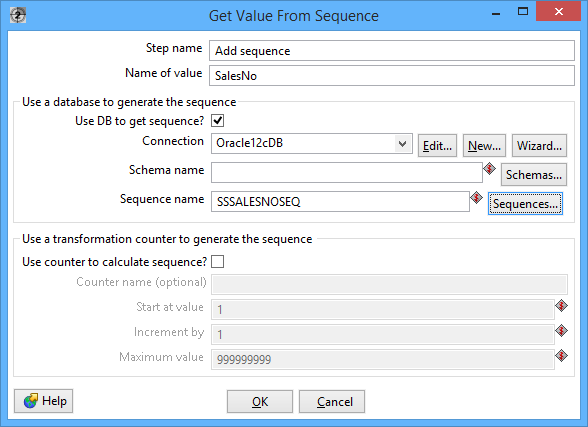


Figure 29: Property Edit Window of Add sequence Step

# 5. Insert data into the SSSales table

* Under the **Design** tab, expand the contents of the **Output** step.
* Click and drag an **Insert/Update** step into your transformation; create a hop between the **Add sequence** and **Insert/Update** steps. Figure 30 shows the Insert/Update step (**SSSales**) connected to Add sequence step.

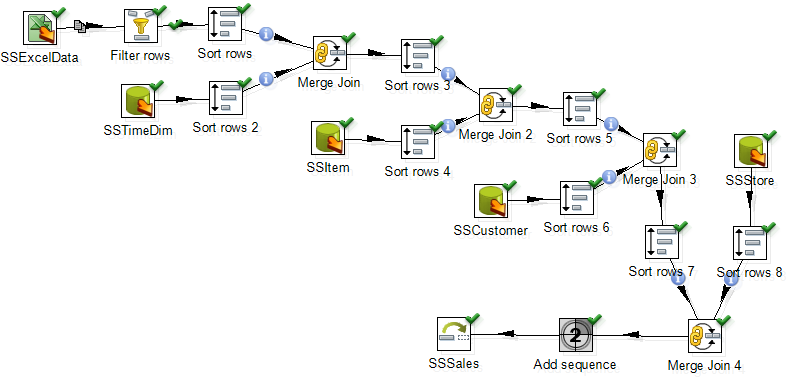


Figure 30: Connect Insert/Update Step to Last Merge Join Step

* Double click the **Insert/Update** component, to specify its properties (Figure 31). Set the **step name** as **SSSales**. Select the **connection** as **Oracle12cDB**. Type in the **Target table** as **SSSales**. Do not click the “**Get fields**” button. Instead, select SalesNo from the two sources and set the comparator to **=**. The final window should look like Figure 31.

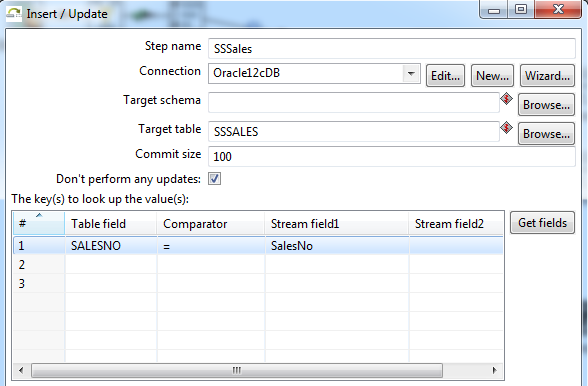


Figure 31: Property Edit Window of Insert/Update Step

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

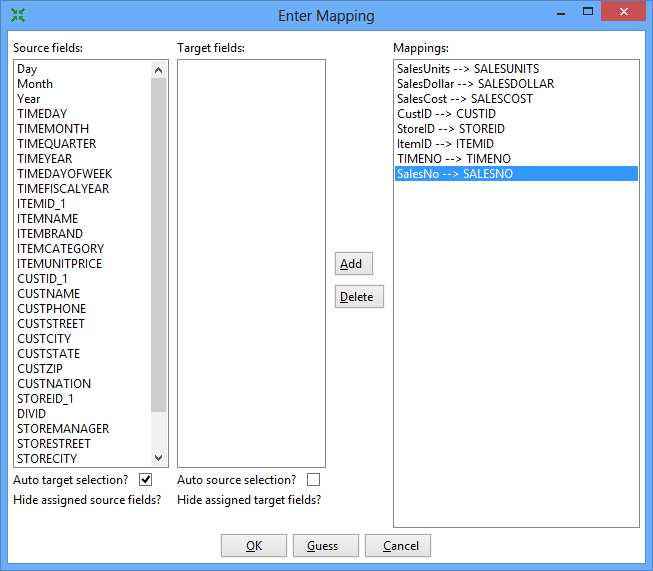


Figure 32: Mapping Edit Window

* The final view of the **SSSales** step will look like Figure 33.

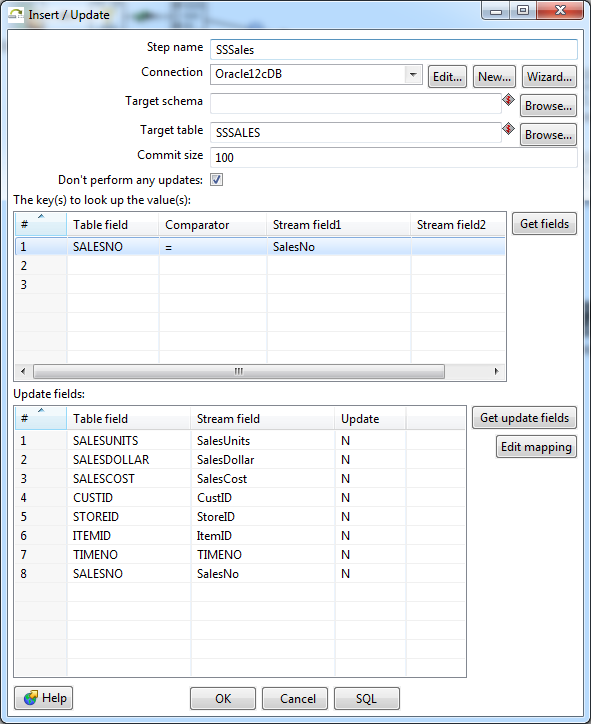


Figure 33: Final view of the SSSales step

* Select the **SSSales** step and run a preview by clicking on Preview.png. In the transformation debug dialog click on **Quick Launch** (Figure 34).

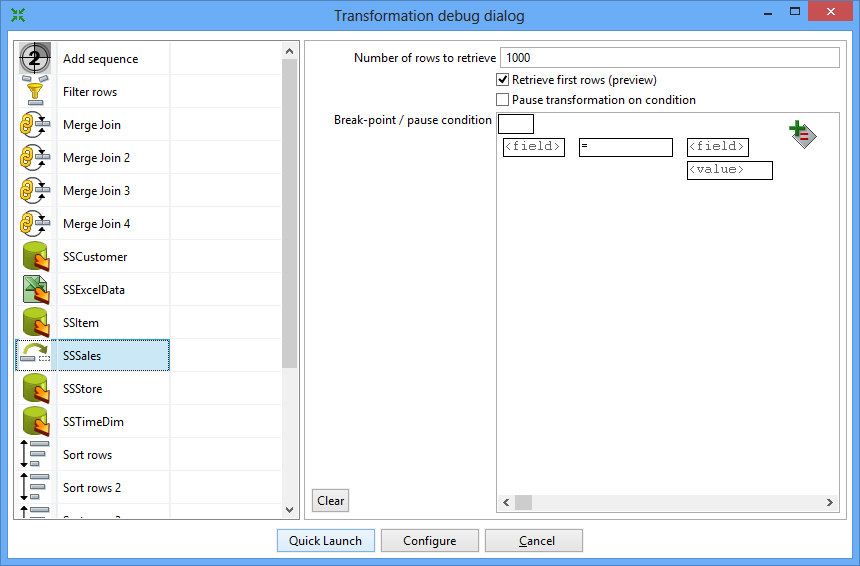


Figure 34: Transformation Debug Dialog

* The Examine preview data window is displayed by Figure 35.

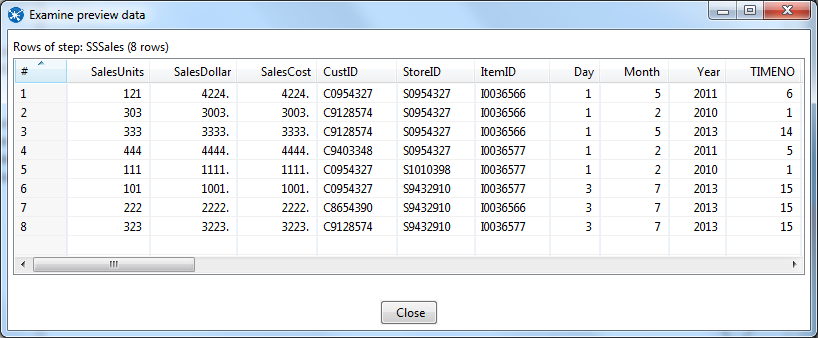


Figure 35: Examine Preview Data Window

* To examine the details of each step, you should examine the Execution Results window below the design pane. The Step Metrics tab (Figure 36) shows details about the execution of each step. You should verify that the **SSSales** step has 8 rows output.

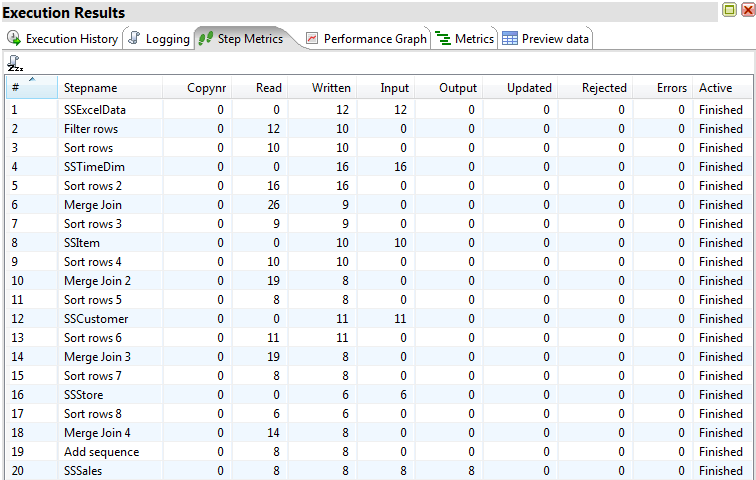


Figure 36: Step Metrics in the Execution Result Window

* Connect to your Oracle account (on your PC or Business School server) so you can verify the number of rows in the *SSSales* table. You should see 200 rows with 8 new rows added to the 192 rows in the samples rows.
* If you do not see the extra rows, the Oracle output component had a failure. To see the error, check the Logging and Step Metrics tabs of the **Execution Results** window.

# 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 step. Figure 37 shows the Design table and input step.

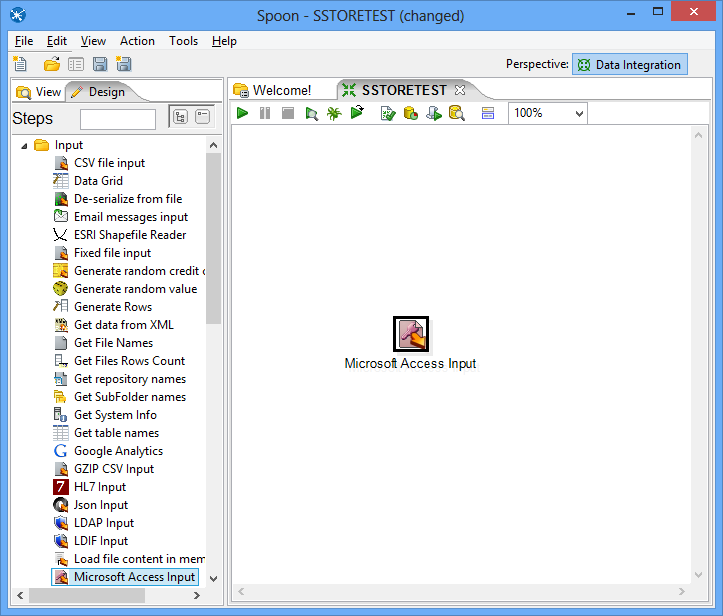


Figure 37: New Microsoft Access Input Step

* 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 38). In this dialog box, you specify the properties related to a particular step.

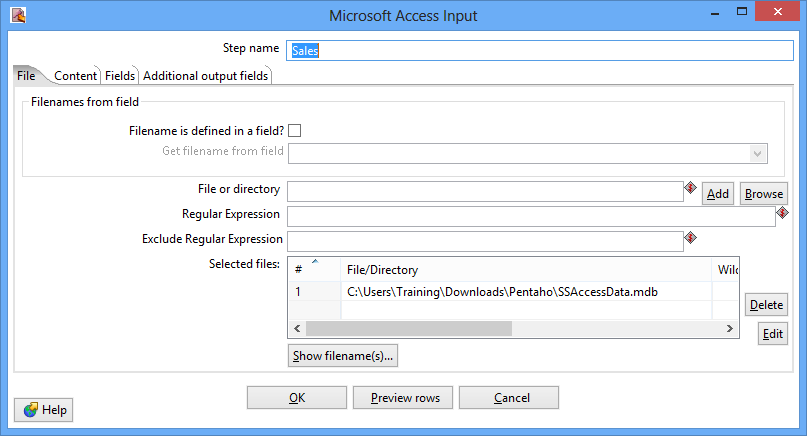


Figure 38: Property Edit Window of Microsoft Access Input Step

* 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 39). Select **Sales** as the table name, click **OK**.

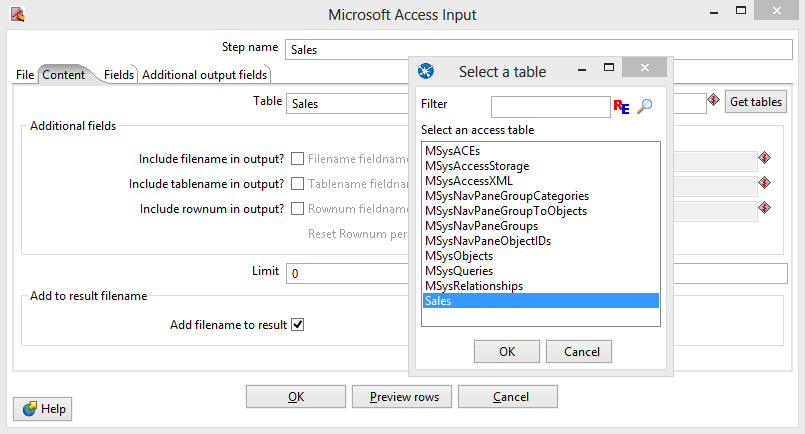


Figure 39: Table Selection Window

* In the tab named **Fields**, click the button “**Get fields**”. There will appear a list (Figure 40) showing the fields in the table named **Sales**.

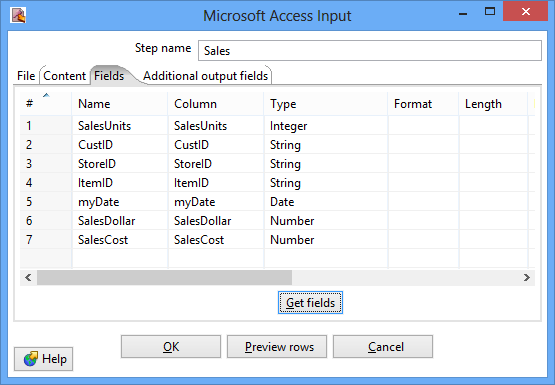


Figure 40: Fields Window for Microsoft Access Input Property Editing

* Click the button “**Preview rows**” to preview the database (Figure 41). When asked for the number of rows type 12 and click OK.

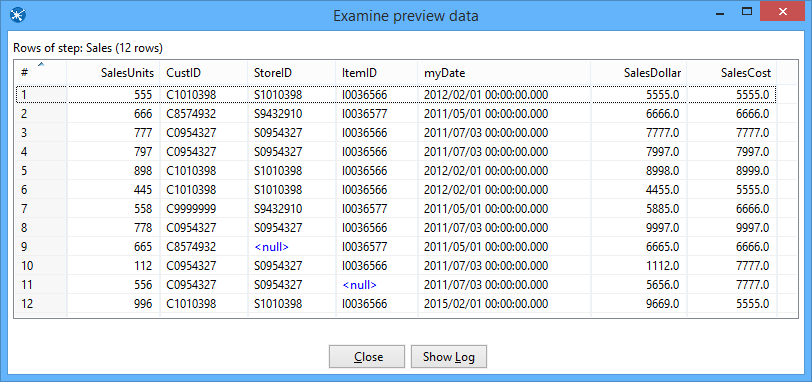


Figure 41: Examine Preview Data Window

* Click **OK** at the bottom of the window. The input icon will change to the shape shown by Figure 42.

Figure 39 Sales Node Icon.png

Figure 42: Sales Step 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 43).

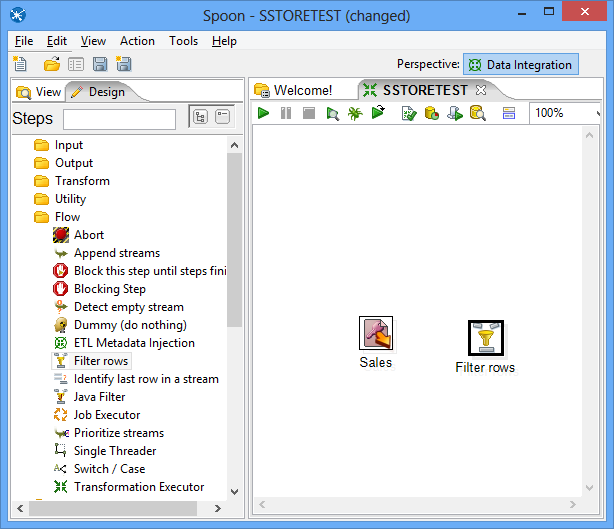


Figure 43: Access Input Step and Filter Step in Spoon

* 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 44.

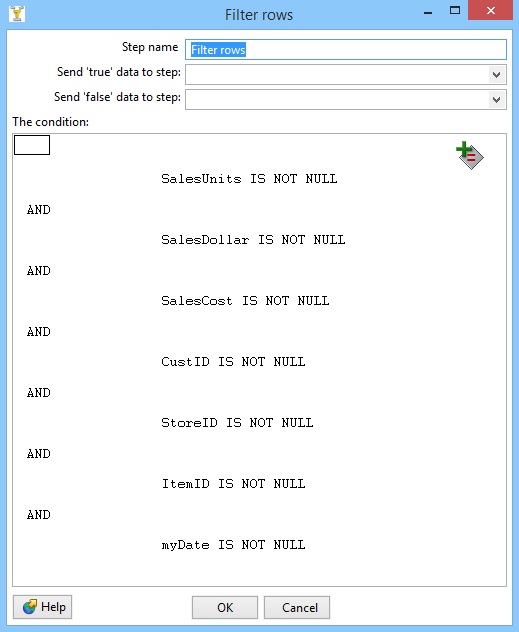


Figure 44: Filter Conditions Window

* 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** step.
* Click and drag a **Select values** step into your transformation.
* Create a “hop” between the **Filter rows** step and the **Select values** step (Figure 45). Select **Result is TRUE** in the filter results selection list

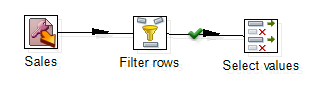


Figure 45: True Filter Results Connected to Select Values Step

* 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 46. Change the **Type** of field **myDate** as **String**, change its **Format** as dd-MM-yyyy. Click **OK**.

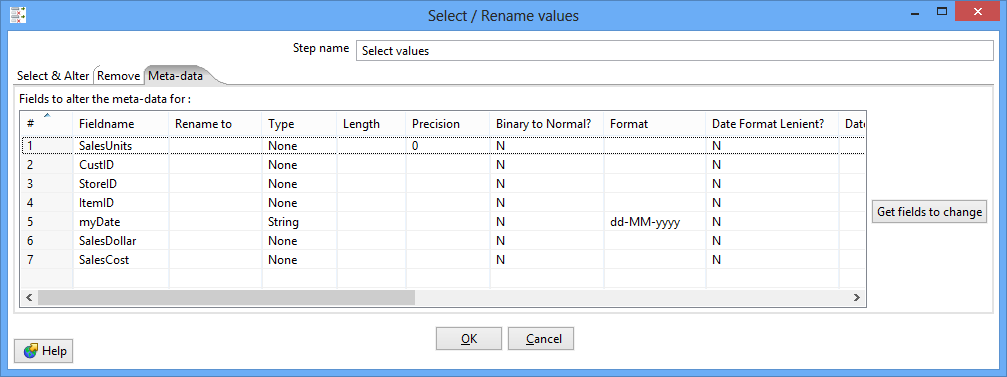


Figure 46: Meta-data Tab of Select Values Property Edit Window

* Under the **Design** tab, expand the contents of the **Transform** step.
* Click and drag a **Split fields** step into your transformation (Figure 47).

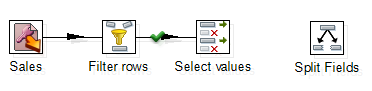


Figure 47: Create Split Fields in Spoon

* 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 48).
* 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**.

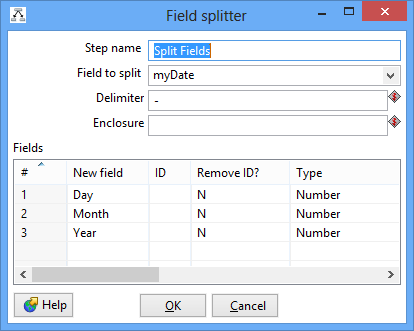


Figure 48: Property Edit Window of Field Splitter Step

* Click OK.
* Click Icon_2_15.bmp , to preview this transform (Figure 49). Make sure that Split Fields step is selected from the left side panel of the transformation debug dialog and click on “**Quick Launch**” button.

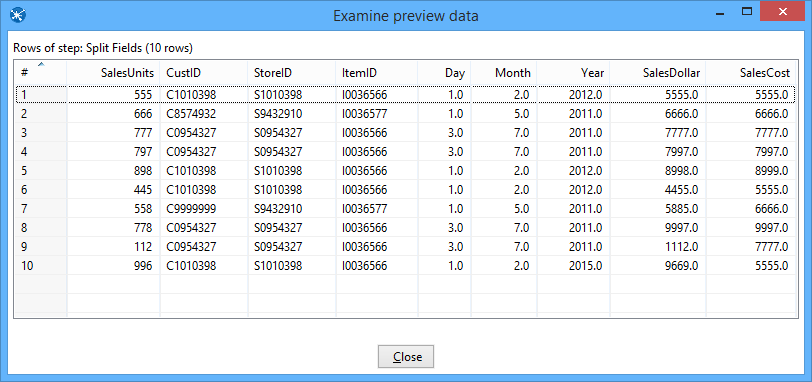


Figure 49: Examine Preview Data Window

# 8. Lookup Columns from the Oracle 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 Oracle database.

* Under the **Design** tab, expand the contents of the **Input** step.
* 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: Oracle12cDB

Connection Type: Oracle

Host Name:

Database Name: (DESCRIPTION=(ADDRESS\_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=132.194.167.74)(PORT=1521)))(CONNECT\_DATA=(SERVICE\_NAME=portdb2.ucdenver.pvt)))

Port Number:

Access: Native (JDBC)

You need to use your assigned user name and password. Do not use ISMG6480ClassStudent.

* 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** step.
* 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** step.
* 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 50 shows the global view of all steps and connections after Step 1.

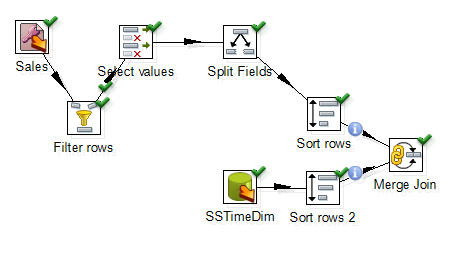


Figure 50: Global View of All Steps and Connections after Step 1

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 51 shows the global view of all steps and connections after Step 2.

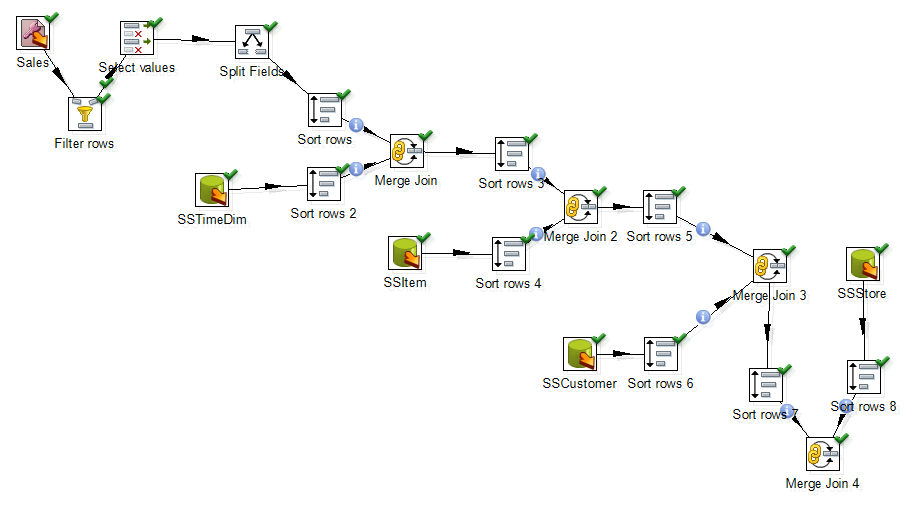


Figure 51: Global View of All Steps and Connections after Step 2

Step 3 – Add SalesNo column.

* Under the **Design** tab, expand the contents of the **Transform** step.
* Click and drag **Add sequence** step into your transformation; create a hop between the **Merge Join 4** and **Add Sequence** steps (Figure 52).
* Double click on the newly created component to open its Basic Settings pane.
* Set **SalesNo** as the name of value. Check the box for use DB to get sequence. Select the **connection** as **Oracle12cDB.** Set **SSSalesNoSeq** as sequence name (Figure 53)

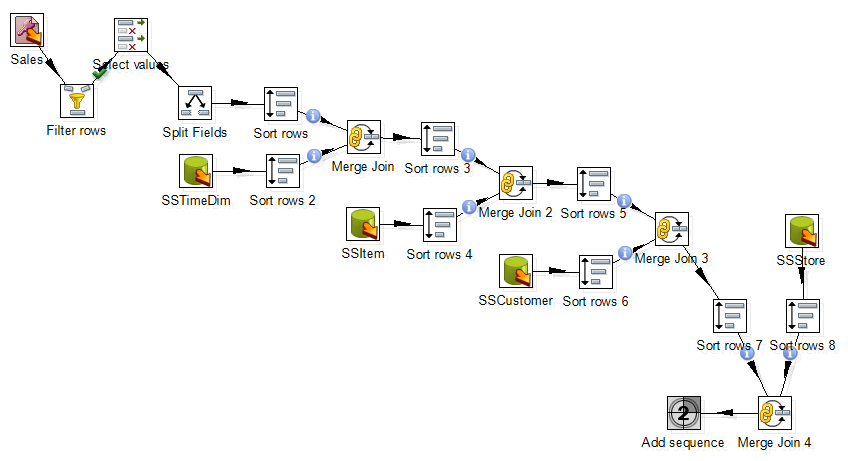


Figure 52: Global View of All Steps and Connections after Step 3

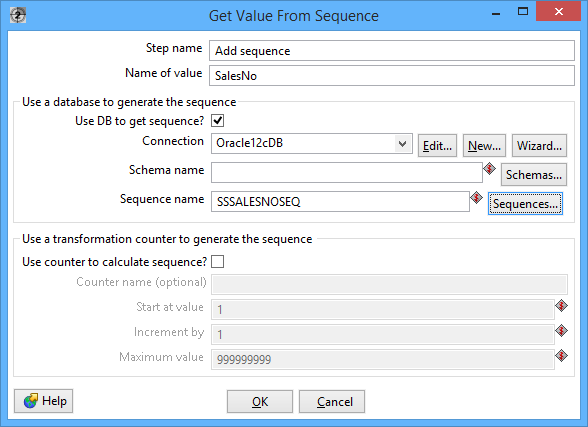


Figure 53: Property Edit Window of Add sequence step

# 9. Insert rows into the SSSales table

* Under the **Design** tab, expand the contents of the **Output** step.
* Click and drag an **Insert/Update** step into your transformation; create a hop between the **Add sequence** and **Insert/Update** steps. Figure 54 shows the connection in the transformation design pane.
* Double click the **Insert/Update** component, to specify its properties. Set the **step name** as **SSSales**. Select the **connection** as **Oracle12cDB**. Type in the **Target table** as **SSSales**. Do not click the “**Get fields**” button. Instead, select SalesNo from the two sources and set the comparator to **=**. Figure 55 shows the window with the lookup values in the middle part.
* Click the button “**Get Updated fields**” and then click on “**Edit mapping**” button to edit mapping. Select the fields named **SalesUnits**, **SalesDollar**, **SaleCost**, **CustID,** **StoreID**, **ItemID** **TIMENO** and **SalesNo** into the **mappings** field. Pentaho will automatically match the corresponding name in the Target field. Only **SalesNo** column must be manually matched with the **SALESNO** field. Then click **OK**. Figure 55 shows the final window.

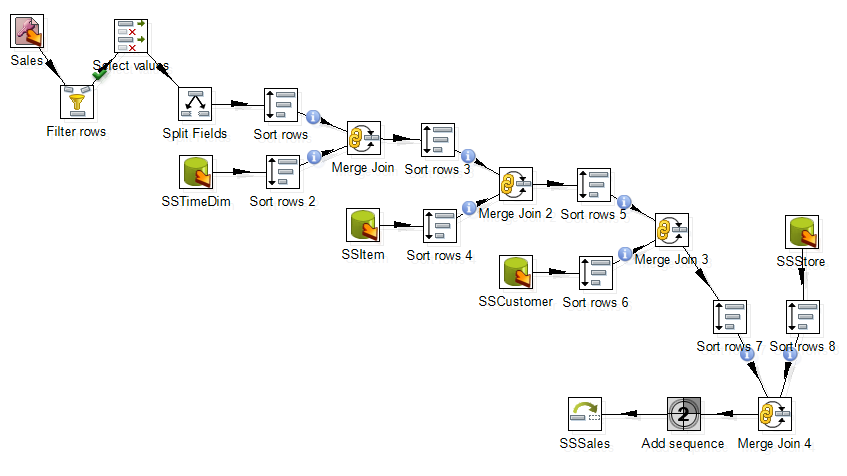


Figure 54: Connect Insert/Update Step to Last Merge Join Step

* Select the **SSSales** step and run a preview by clicking on Preview.png. In the transformation debug dialog click on **Quick Launch** button. After the execution, you should see the Execution Results window below the transformation design. The Step Metrics tab (Figure 56) shows that 8 rows were inserted into the *SSSales* table.
* Connect to your Oracle account (on your PC or Business School server) so you can verify the number of rows in the *SSSales* table. You should see 208 rows with 8 new rows added to the 200 rows in the sample rows (192 original rows and 8 rows from the Excel transformation).

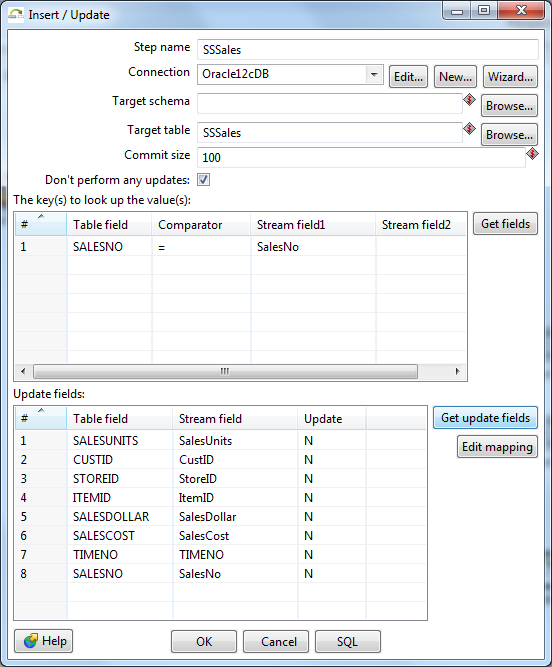


Figure 55: Insert/Update Step Window

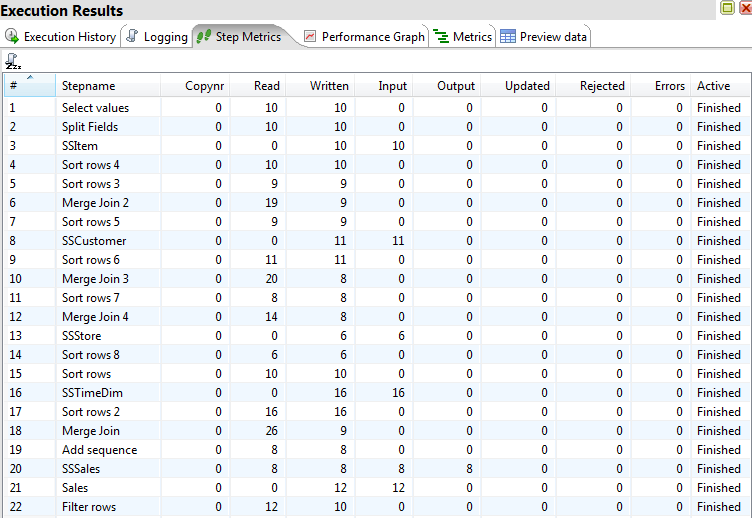


Figure 56: Step Metrics in the Execution Result Window for the Access Transformation