

Data Integration Exercise with PDI and PostgreSQL

In the data integration exercise, you will use Pentaho Data Integration (PDI) to transform two data sources and load data into a PostgreSQL table. You 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. Before starting the exercise, you should install PDI and create the Store Sales tables and sequences and populate tables with sample rows.

This document involves a Table Output step to insert rows into the *SSSales* table. The Table Output step is conceptually preferred when only needing to insert rows into a fact table.

You can use the Store Sales tables on a local database with PostgreSQL installed on your computer. The instructions in the exercise demonstrate connection to PostgreSQL on your local computer.

You also need to download the input files (Excel file and Access database file) available in the class website. You will use these input files in the beginning steps of the two job designs that you will create.

This tutorial uses the community edition (CE) of PDI. The latest version (9.3) was installed in June 2022 using SourceForge (sourceforge.net/projects/pentaho/). The SourceForge page for Pentaho also contains installation guides. Figure 1 shows the launch page of the latest version.

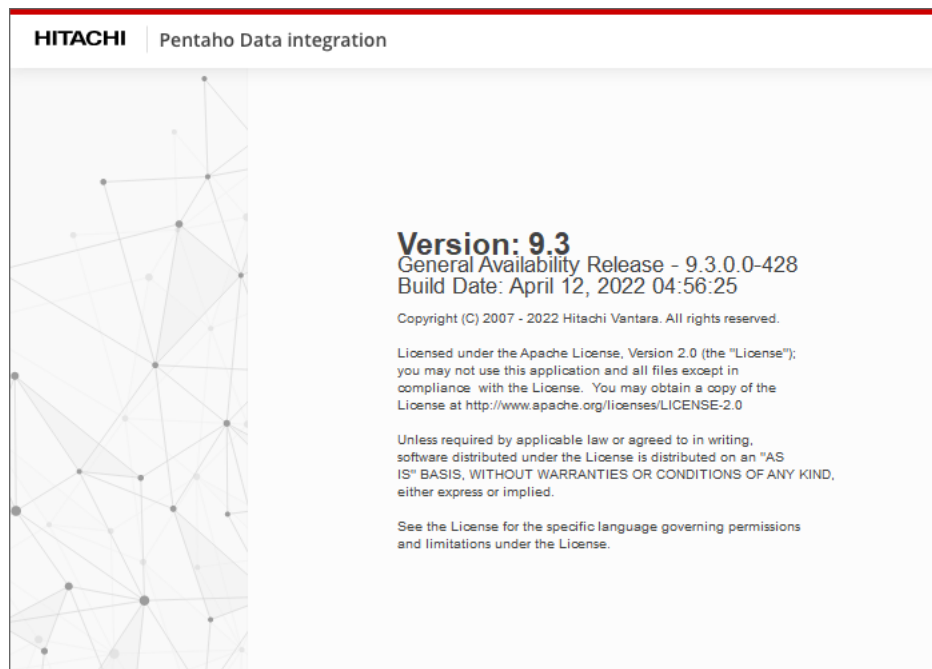


Figure 1: Pentaho Data Integration Welcome Window

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.

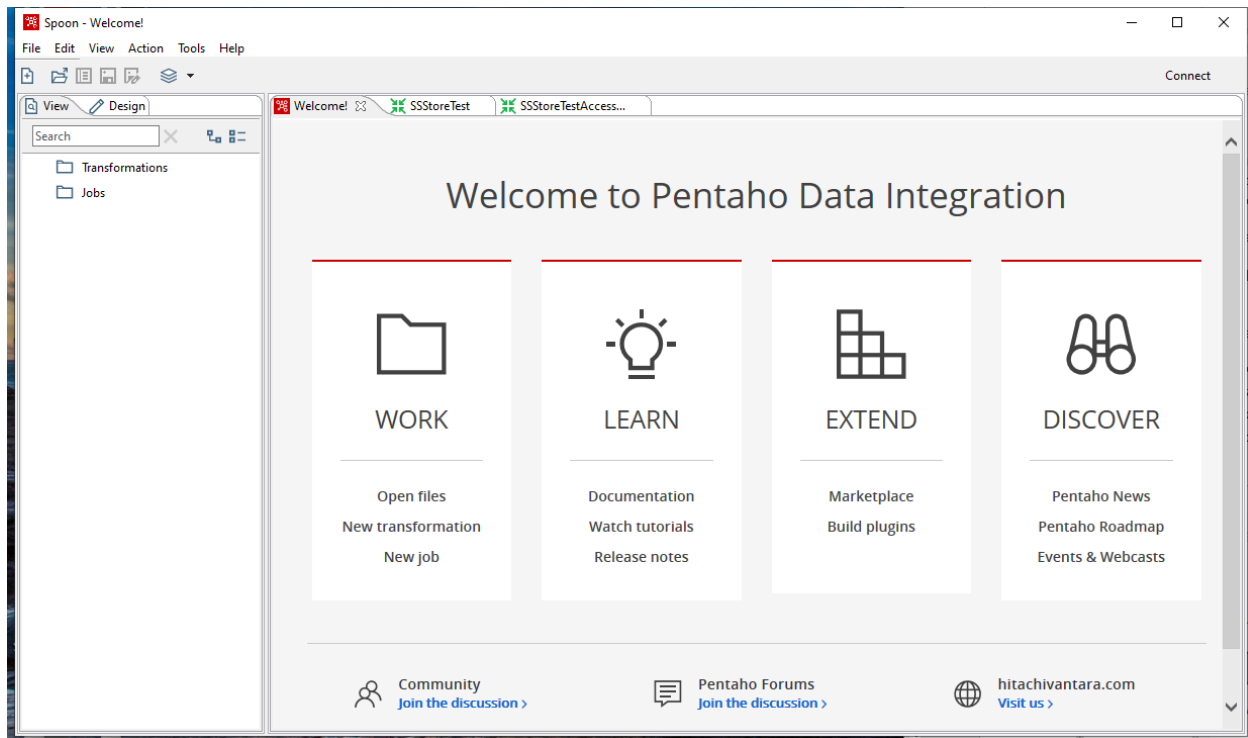


Figure 2: Spoon Opening Window

1. Managing Database Connections

Pentaho Data Integration allows you to define connections to multiple databases provided by multiple database vendors (MySQL, Oracle, PostgreSQL, and many more). Pentaho Data Integration installs 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 will not 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 Connection step in the explorer View in Spoon.

There are several ways to define a new database connection. You will configure the database connection later in this tutorial.

- In Spoon, under View in the navigation tab, right click and choose New.
- In Spoon, under View in the navigation tab, right click Database connections and choose New Connection Wizard (Figure 3).

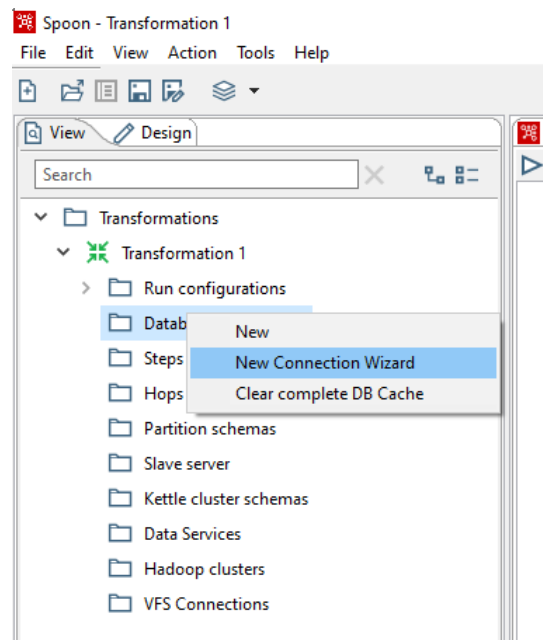


Figure 3: New Connection Wizard

Adding a JDBC Driver

Pentaho Data Integration uses database drivers to provide connections and other operations with databases. Installation of Pentaho Data Integration provides a standard set of database drivers. For PostgreSQL, the installation of Pentaho Data Integration includes a driver for PostgreSQL so you should not need to install a driver. This tutorial used PostgreSQL 14.1 without a need to install an additional database driver. The transformations in the tutorial were also executed using PostgreSQL 13 using the same default database driver.

I have not tested PDI with PostgreSQL with the Mac OS X. I doubt that a database driver download is necessary. If you cannot make a connection to PostgreSQL, you can try installing the jdbc driver for postgresql.

- https://help.pentaho.com/Documentation/9.1/Setup/JDBC_drivers_reference

- Copy the driver JAR file to the data-integration/lib folder where the data-integration folder resides inside the folder containing the Pentaho startup files (spoon.sh).
- From the window wizard (Figure 4) Enter a name of the database connection, choose PostgreSQL from the list, and Native (JDBC). Click Next.
- Enter the host name, port, and database name. Note: the default host name and port for PostgreSQL is showing in (Figure 5), make sure you have the same values in your PostgreSQL settings. Please refer to Section 5 in the tutorial document for PostgreSQL and pgAdmin for getting details about creating a connection to PostgreSQL Server.
- Enter the username and password that was during the installation of PostgreSQL. Click Test database connection.
- If you received a message Connection to the database was successful, then click Finish.

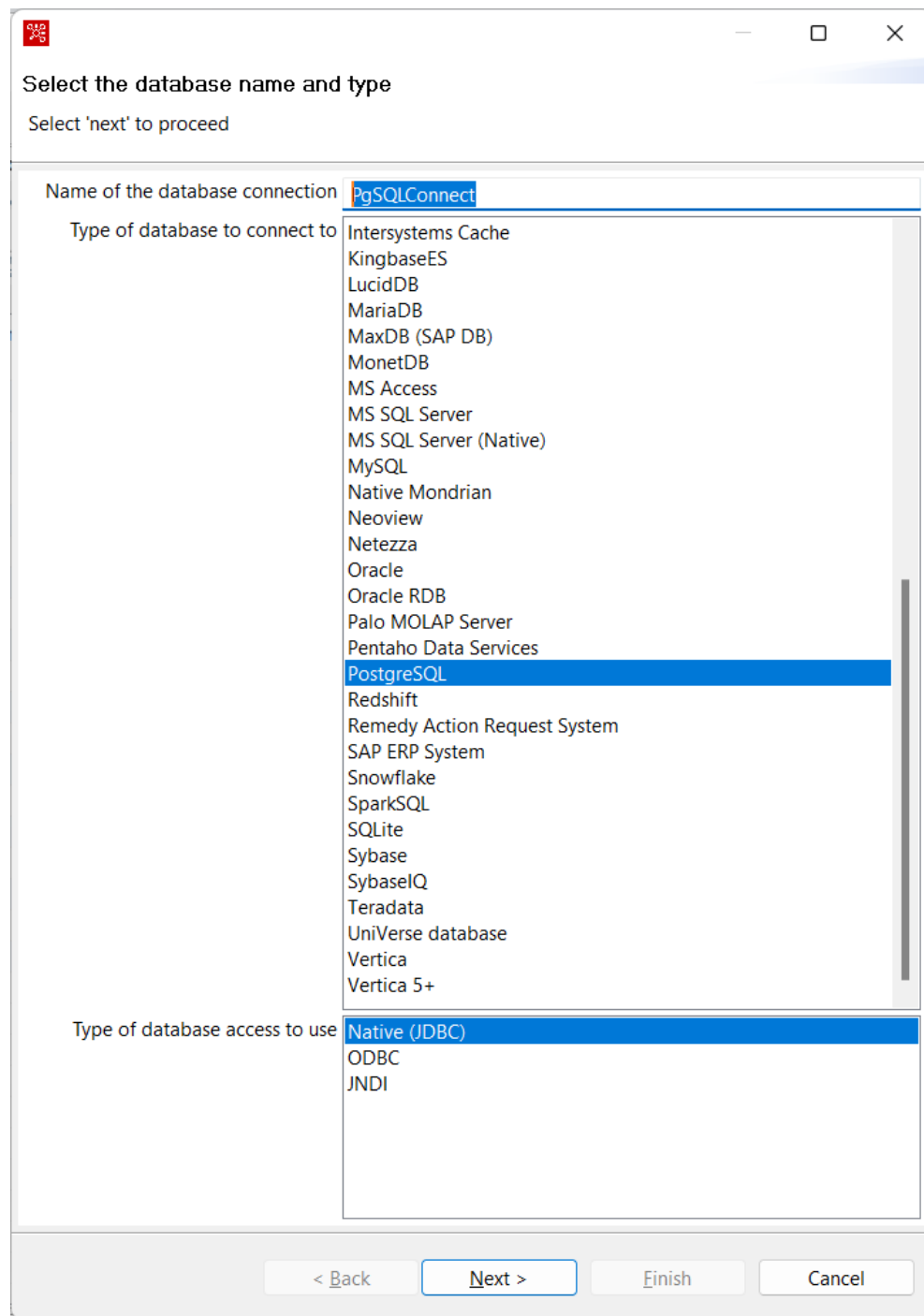


Figure 4: Window Wizard for a Database Connection

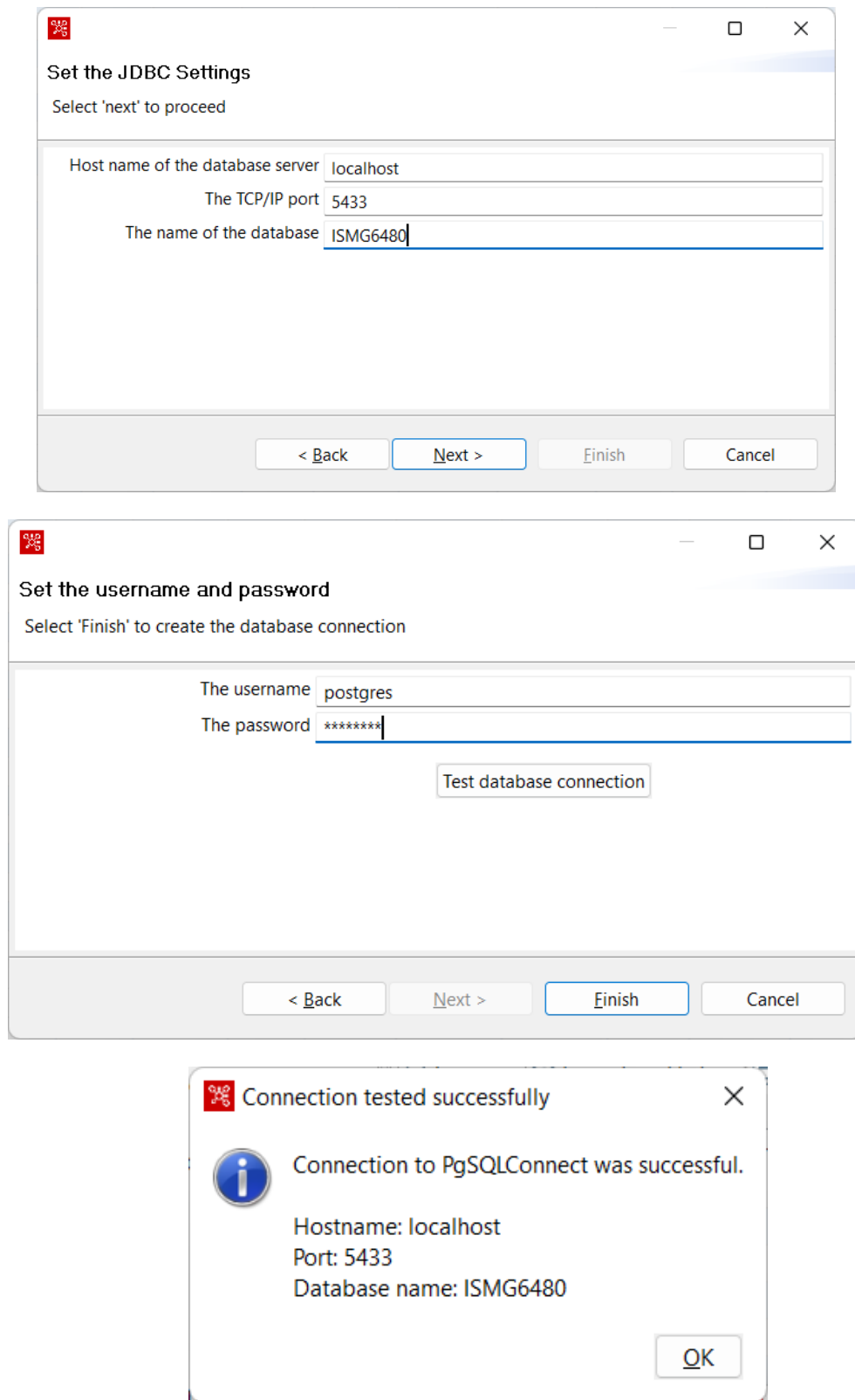


Figure 5: PostgreSQL Connection Details

2. Overview of Transformations in the Exercise

Spoon is the desktop client component of PDI supporting creation of transformations and jobs. Transformations describe data flows such as reading from a source, transforming data, and loading it into a target database table. Jobs 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?”

A Pentaho transformation executes as a data pipeline with steps connected by directed hops. The output of a prior step flows into the next step as indicated by the hop connecting the steps. Pipeline processing is a well-established processing model amenable to optimization and parallel processing depending on hardware configuration. Steps can execute in parallel, operating on different input records.

During execution of a transformation, the Pentaho processing engine manages a data structure known as the stream. Execution of a step modifies the stream such as by adding new fields in a stream record, sorting records on the stream, or deleting stream records. For example, the Filter step deletes rows on the stream, while the Sort rows step orders records on the stream.

This exercise involves development of two similar transformations shown in Figures 6 and 7. Both transformations process an input file containing rows to insert into the *SSSales* table of the Store Sales data warehouse. The initial input step (Microsoft Excel worksheet or Microsoft Access table) creates stream records with fields corresponding to columns in the *SSSales* table. Most steps in the transformations perform validations to ensure insertion of valid rows in the *SSSales* table. The Filter rows step deletes records with a null value in any field. The Merge join steps (Merge Join, Merge Join 2, Merge Join 3, and Merge Join 4 in Figures 6 and 7) combines two streams on a join condition to ensure valid foreign key values. Records not matching on the join condition are deleted from the stream. Merge joins require sorting of records in the same order. In Figure 6, the Merge Join step combines the streams starting with the *SSExcelData* step (a Microsoft Excel input step) and the *SSTimeDim* step (Table input step). The Merge Join step connects to the Table Output step to add rows to the *SSSales* table. The *SSSales* table uses an identifier column as its primary key so that the transformation does not need to create a primary key value.

The Access transformation (Figure 7) uses two additional steps (Select values and Split fields) to parse date components. The Access table step has a column with date values that must be parsed for the Merge join step. The Merge join step matches on the year, month, and day components of a date value.

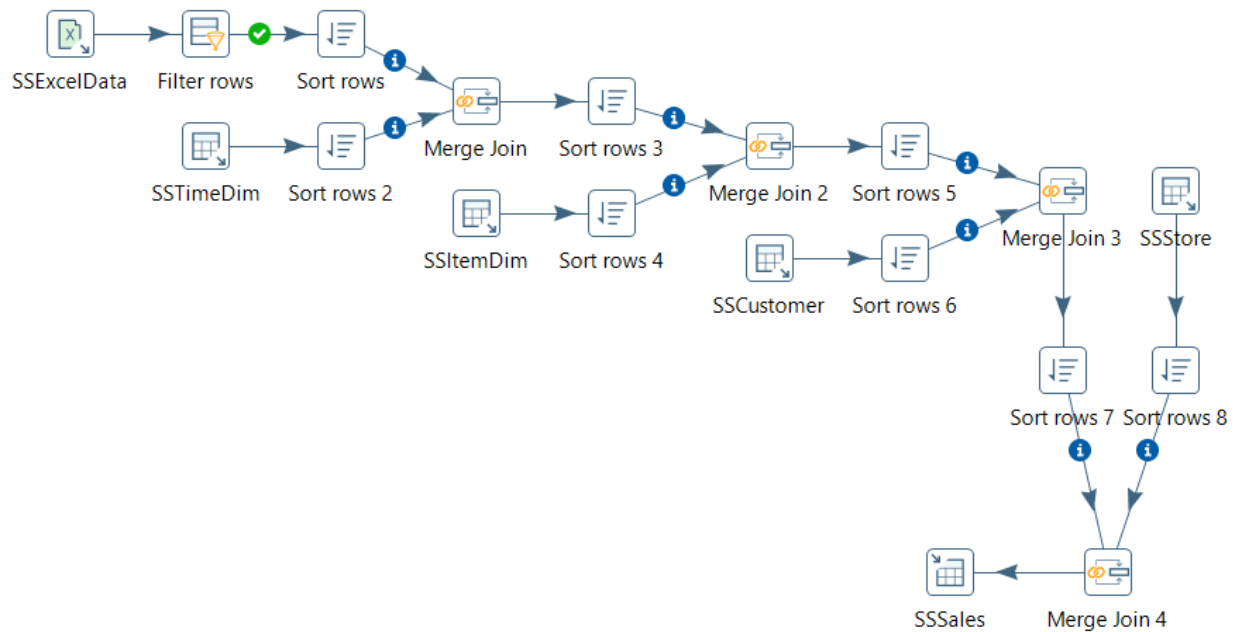


Figure 6: Transformation using Microsoft Excel Input

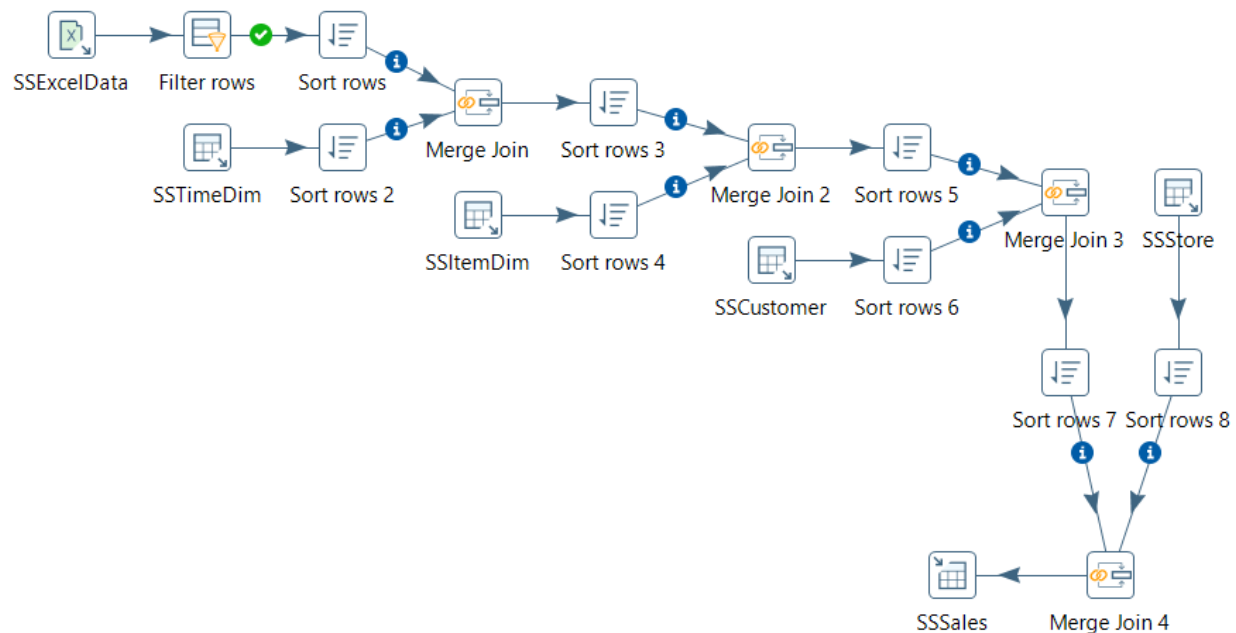


Figure 7: Transformation using Microsoft Access Input

To provide guidance about fields in stream records, Pentaho provides menu choices to examine the input and output fields for each step in a transformation. A right click on a step provides a menu with items (Figure 8) for examining the input (Input Fields ...) and output (Output fields ...) fields on the stream for that step. Selecting Input Fields ... for the Sort rows of the Excel

transformation (Figure 6) provides details of the stream input to the step as shown in Figure 9. All fields originate (Step origin) in the SSEXcelData step.

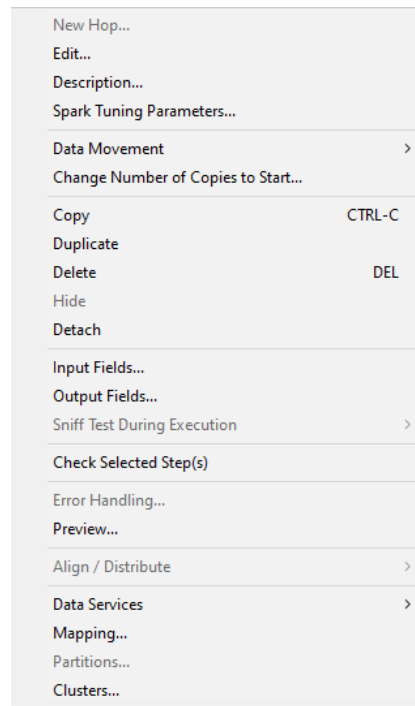


Figure 8: Menu Choices for Each Step

Step fields and their origin

Step name: Sort rows

Fields:

#	Fieldname	Type	Length	Precision	Step origin	Storage	Mask	Currency	Decimal	Group	Trim	Comments
1	SalesUnits	Number	-	-	SSEXcelData	normal					none	
2	SalesDollar	Number	-	-	SSEXcelData	normal					none	
3	SalesCost	Number	-	-	SSEXcelData	normal					none	
4	CustID	String	-	-	SSEXcelData	normal					none	
5	StoreID	String	-	-	SSEXcelData	normal					none	
6	ItemID	String	-	-	SSEXcelData	normal					none	
7	Day	Number	-	-	SSEXcelData	normal					none	
8	Month	Number	-	-	SSEXcelData	normal					none	
9	Year	Number	-	-	SSEXcelData	normal					none	

Edit origin step Cancel


Figure 9: Stream Fields for the Sort rows Step

3. Creating your first transformation and loading Excel worksheet

This exercise will step you through building your first transformation with the Spoon client of 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 File) in the upper left corner of the Spoon window.

3. Select **Transformation** from the list of components (Figure 10) displayed after selecting the **New File** button.

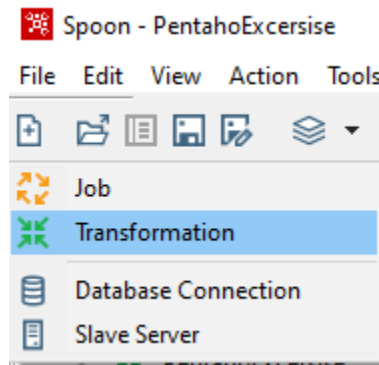


Figure 10: Spoon New File List

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 11).

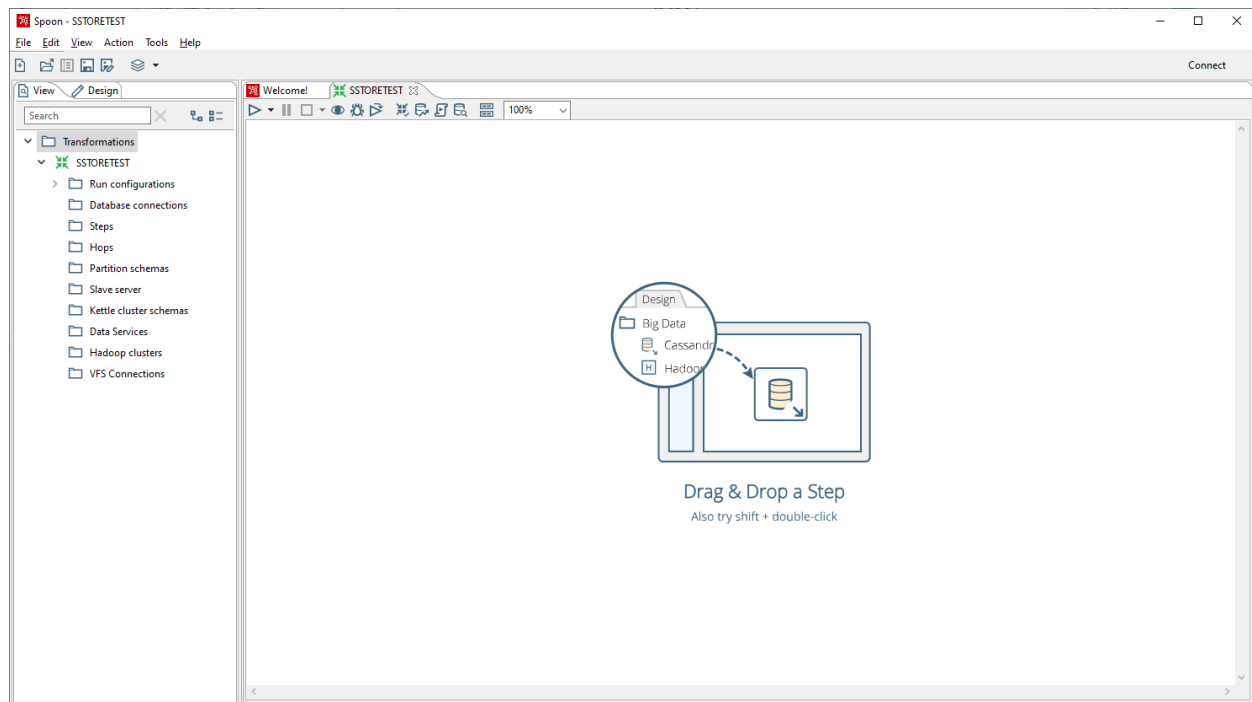


Figure 11: Empty Transformation Window

Step 4 – Create the Excel Input step:

- Under the Design tab, expand the Input step (Figure 12).

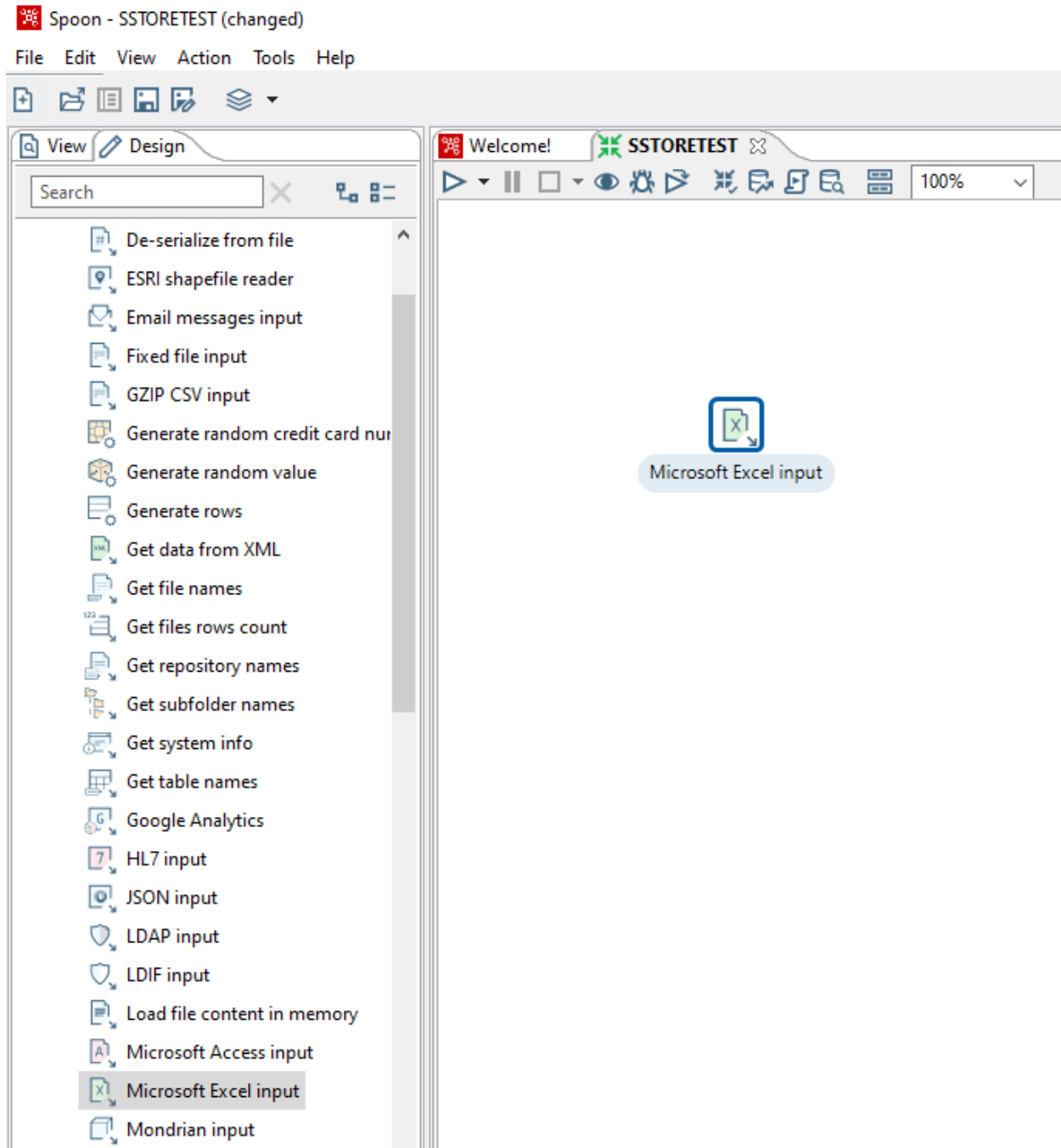


Figure 12: List of Input Steps with Microsoft Excel Input step in the Transformation Design Window

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

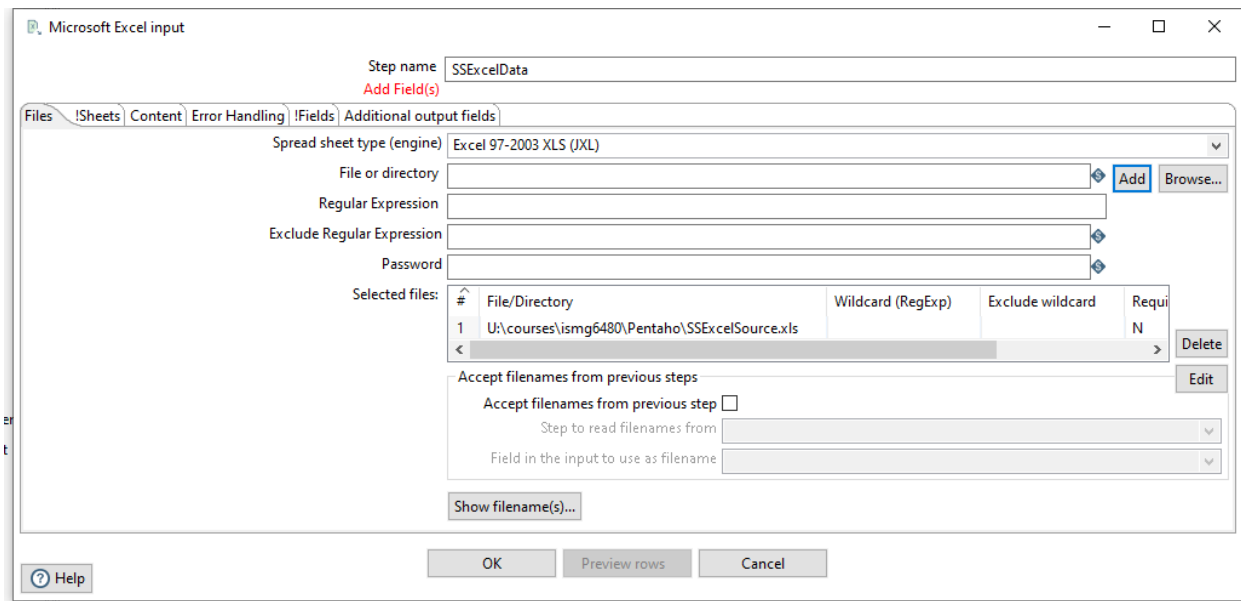


Figure 13: Files Window for Microsoft Excel Input Property Editing

- 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 14) 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 should see the data types, length, and precision as the specification in Figure 15.

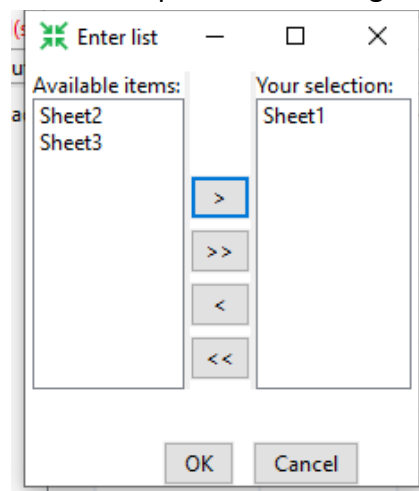


Figure 14: Sheet Specification Window

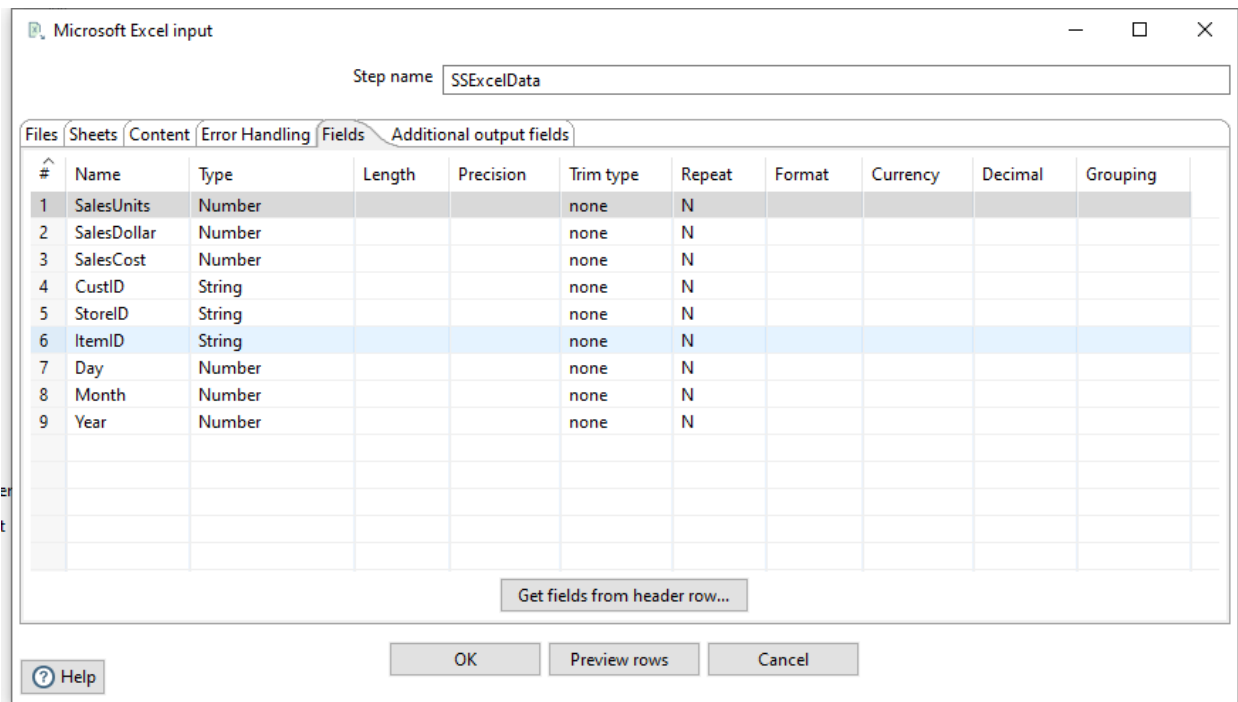


Figure 15: Fields Window for Microsoft Excel Input Property Editing

- To ensure that the Excel input can read rows from the associated worksheet, select the Preview rows button. Figure 16 shows the preview of rows. Close the window when finished previewing rows.
- Click **OK** at the bottom of the window. The input icon will change to the icon displayed in Figure 17.

Examine preview data

Rows of step: SSEcelData (12 rows)

#	SalesUnits	SalesDollar	SalesCost	CustID	StoreID	ItemID	Day	Month	Year
1	111.0	1111.0	1111.0	C0954327	S1010398	I0036577	1.0	2.0	2018.0
2	222.0	2222.0	2222.0	C8654390	S9432910	I0036566	3.0	7.0	2021.0
3	333.0	3333.0	3333.0	C9128574	S0954327	I0036566	1.0	5.0	2021.0
4	444.0	4444.0	4444.0	C9403348	S0954327	I0036577	1.0	2.0	2021.0
5	101.0	1001.0	1001.0	C0954327	S9432910	I0036577	3.0	7.0	2021.0
6	202.0	2002.0	2002.0	<null>	S0954327	I0036566	1.0	5.0	2019.0
7	303.0	3003.0	3003.0	C9128574	S0954327	I0036566	1.0	2.0	2018.0
8	404.0	4004.0	4004.0	C9403348	<null>	I0036577	3.0	7.0	2021.0
9	121.0	4224.0	4224.0	C0954327	S0954327	I0036566	1.0	5.0	2019.0
10	232.0	2332.0	2332.0	C8654390	S0954327	I0036566	1.0	2.0	2023.0
11	323.0	3223.0	3223.0	C9128574	S9432910	I0036577	3.0	7.0	2021.0
12	434.0	4334.0	4334.0	C9403348	S0954327	I9999999	3.0	7.0	2021.0

Buttons: Close, Show Log

Figure 16: Preview Rows Window

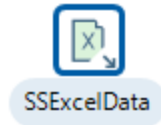


Figure 17: SSEExcelData Icon

Step 5 – In this part of the tutorial, you will add constraint checking for null values in stream records obtained from the Excel data source. The rows from the Excel data source were added to the stream (data maintained in a Pentaho transformation) in the output of the first step.

- Add a Filter Rows step to your transformation. Under the **Design** table, go to **Flow** → **Filter Rows**.
- 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 18).



Figure 18: Hop connecting an Excel Input Step Connected to a Filter Rows Step

- Alternatively, you can draw hops by hovering over a step until the hover menu (Figure 19) appears. Drag the hop painter icon from the source step to your target step.

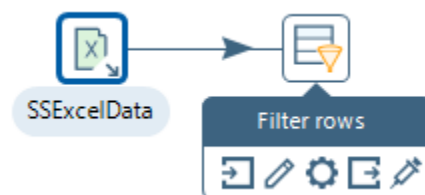


Figure 19: Hover Menu

- Double-click the **Filter Rows** step. The **Filter Rows** edit properties dialog box appears (Figure 20).

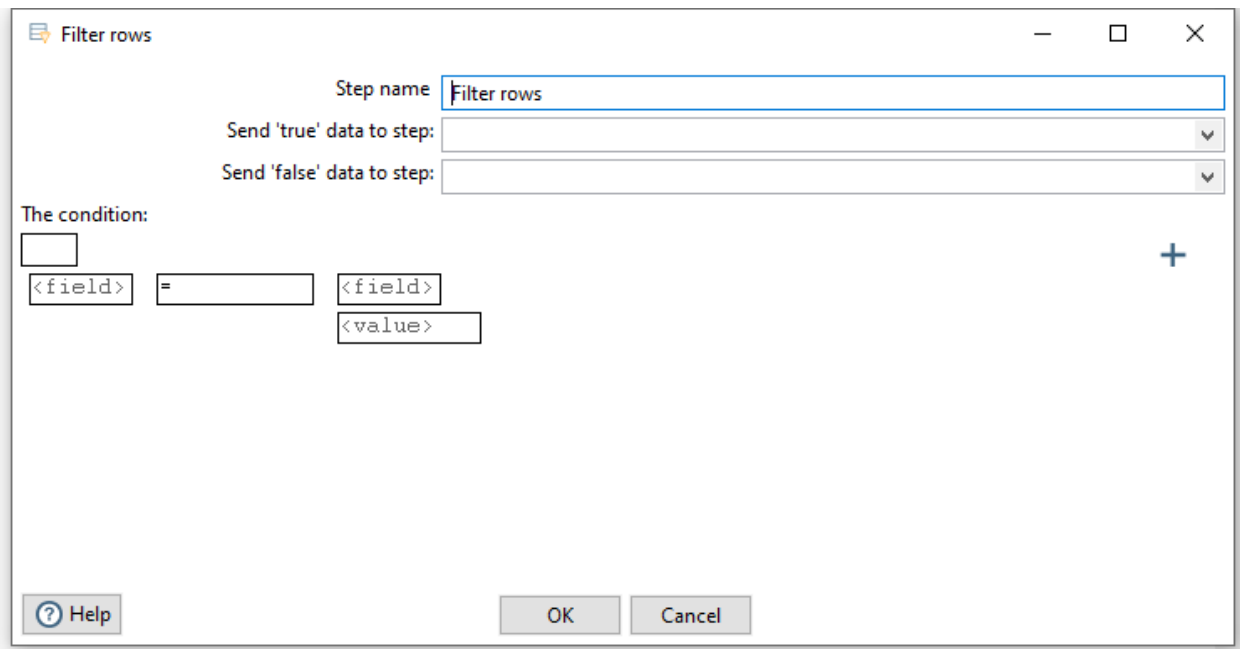


Figure 20: Property Edit Window of Filter Step

- The **Step Name** field is **Filter rows**.
- 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 21) select **SalesUnits** and click **OK**.
- Click on the comparison operator (Figure 22) (set to = by default) and select the **IS NOT NULL** function and click **OK**.
- 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 connected 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 23.
- Save your transformation.

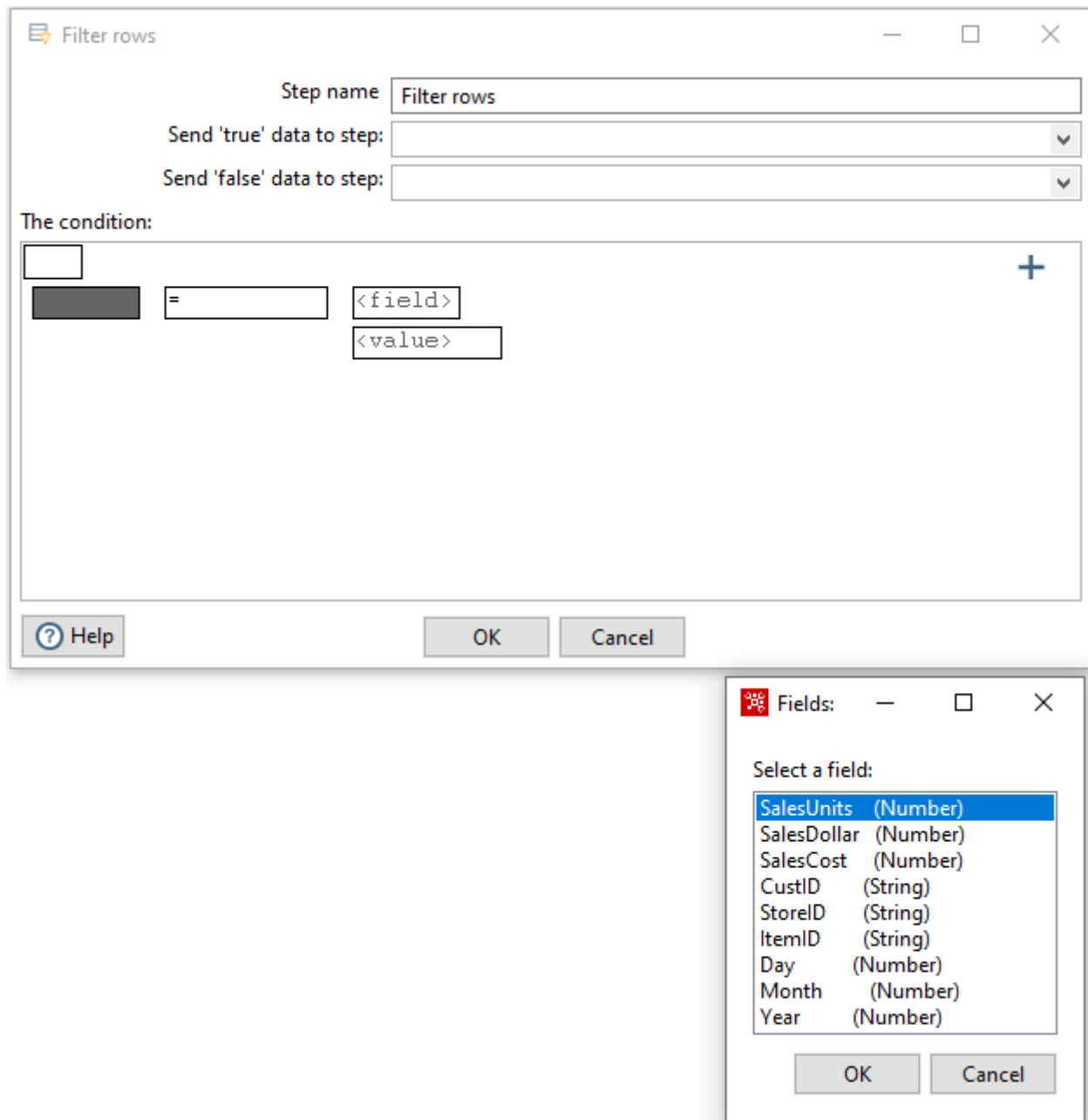


Figure 21: Condition Fields Selection Window

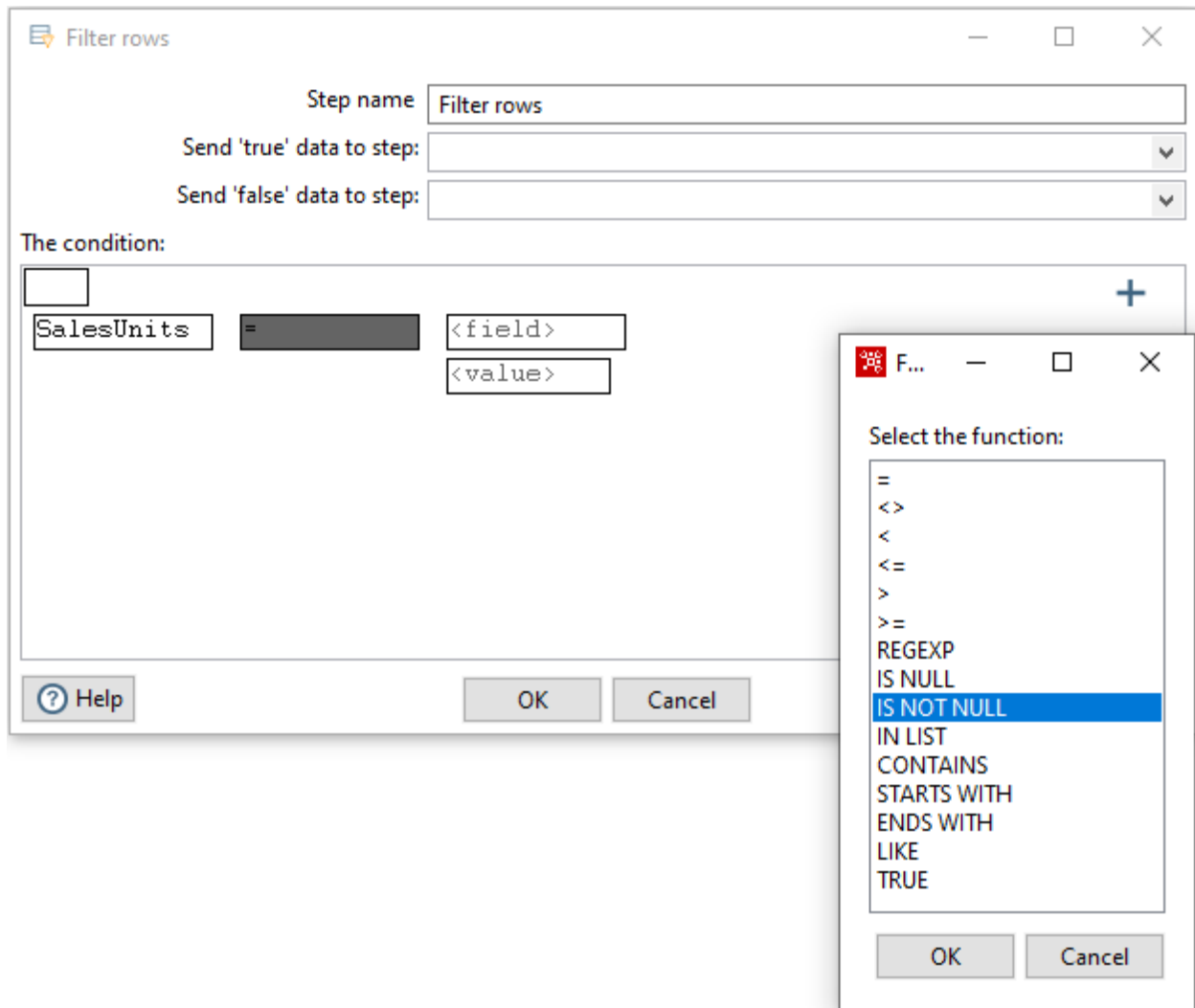


Figure 22: Comparison Operator List

Filter rows

Step name:

Send 'true' data to step:

Send 'false' data to step:

The condition:

<input type="checkbox"/>	^^ UP ^^	Level 2, Select UP to go up one level	+
		SalesUnits IS NOT NULL	
AND		SalesDollar IS NOT NULL	
AND		SalesCost IS NOT NULL	
AND		CustID IS NOT NULL	
AND		StoreID IS NOT NULL	
AND		ItemID IS NOT NULL	
AND		Day IS NOT NULL	
AND		Month IS NOT NULL	
AND		Year IS NOT NULL	

Help OK Cancel

Figure 23: Filter Conditions Window

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

- Under the **Design** tab, expand the contents of the **Transform** category.
- 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 24).

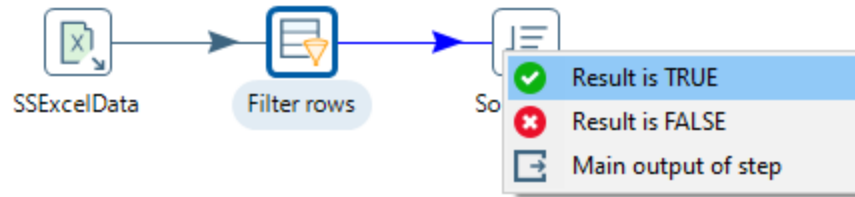


Figure 24: Filter Results Selection List

- Double-click the **Sort Rows** step to open its edit properties dialog box (Figure 25). Click “**Get Fields**” to obtain the fields. Delete other fields except the Day, Month and Year fields. Then click Ok.

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Co
1	Day	Y	N	N	0
2	Month	Y	N	N	0
3	Year	Y	N	N	0

This option prevents duplicate rows from being written
This option only verifies uniqueness of the specified key

Figure 25: Property Edit Window of Sort Rows Step

4. Lookup Columns from the PostgreSQL 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 PostgreSQL tables to ensure loaded data does not contain invalid foreign keys.

Step 1 – Access the *SSTimeDim* table from PostgreSQL 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 26).
- Rename your Table Input step to *SSTimeDim*.

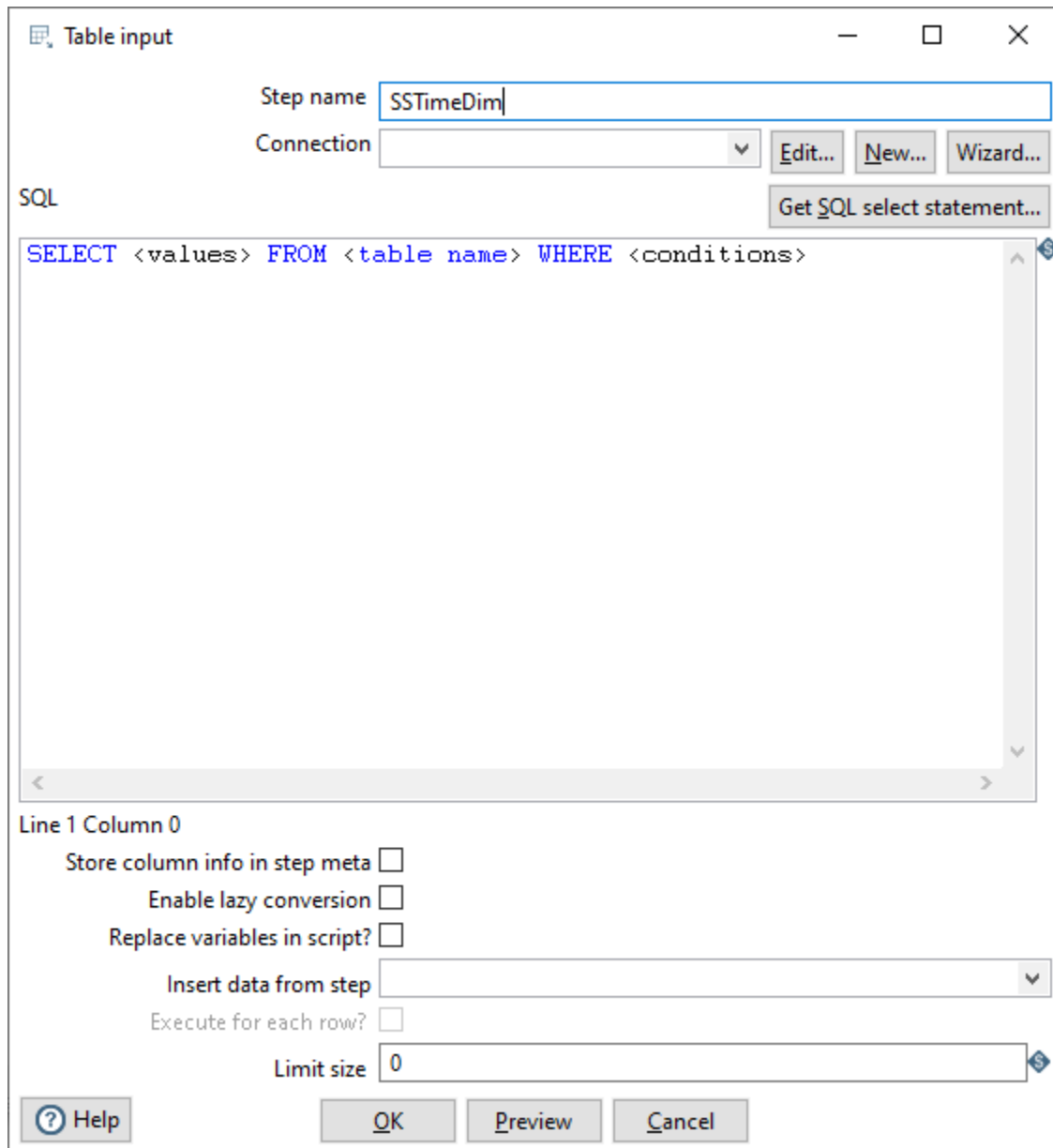


Figure 26: Property Edit Window of Table Input Step

- 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 27. You can choose any name for the connection.
- **IMPORTANT:** Before setting the connection information, you should ensure that the database is created in PostgreSQL along with the tables and the records inserted into those tables. Here are the details to connect to the PostgreSQL 14.1. The Database Name and the Username are the one is the one that you create in PostgreSQL. The full value for database connection is given in the remainder of step 1.

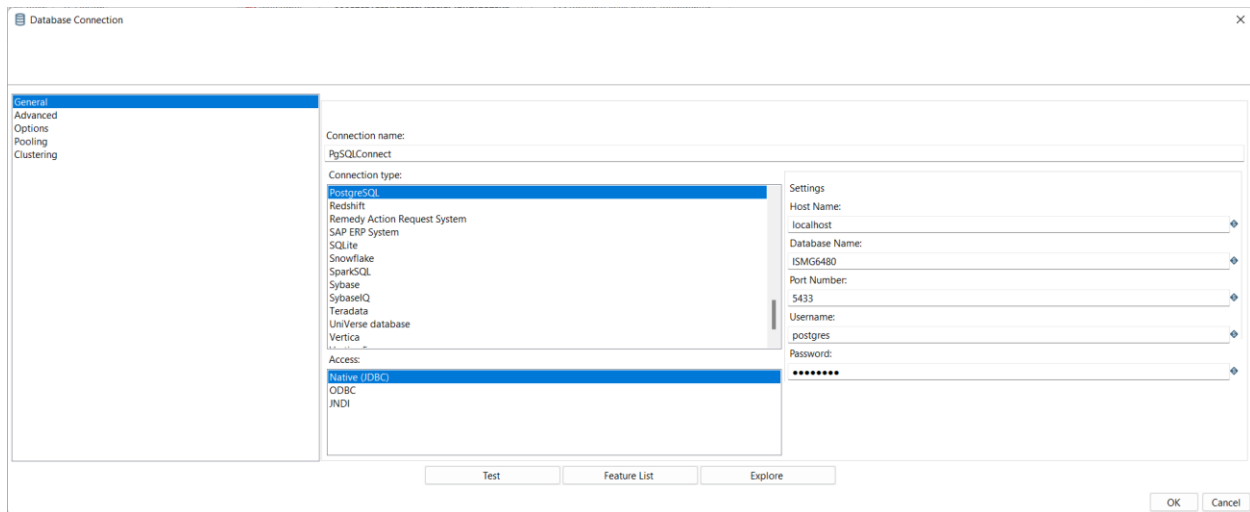


Figure 27: Database Connection Window

- Connection Name: PgSQLConnect (You can use another name if you want)
 Connection Type: PostgreSQL
 Host Name: localhost
 Database Name: ISMG6480 (Use the database name that you used.)
 Port Number: 5433 is the default port number. To check the port used by your installed PostgreSQL database server, right click on the server in pgAdmin. Then click **Properties...** You can find the port in the Connection tab. You will not be able to connect to your PostgreSQL database server if you use an incorrect port number.
 Username: postgres (default administrative user name)
 Password: <blank> (if password was not specified or use password specified.)
 Access: Native (JDBC)
 Note: The username “postgres” is the default administrative user after installation of PostgreSQL. For the password, you need to use the password that you specified if any. In this example, the password for the postgres user was omitted during PostgreSQL installation. The Host Name and port number are the default values for Windows 10 installation of PostgreSQL version 14.1.
- Click “Test” to test the connection. Then success test result is shown by Figure 28.

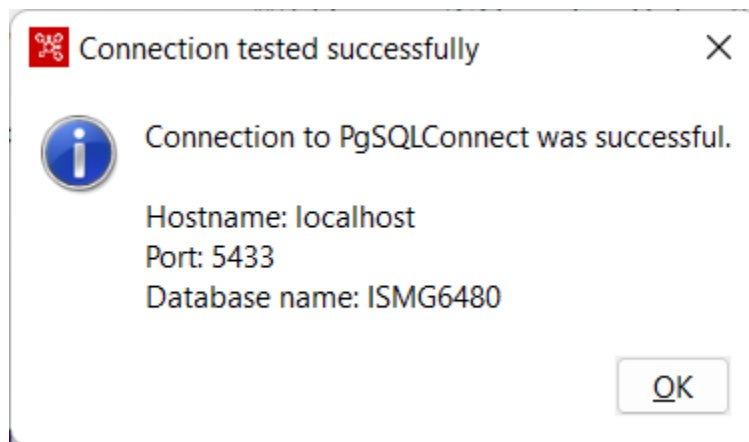


Figure 28: Database Connection Test

- Type in “`SELECT * FROM SSTimeDim`” in the SQL section (Figure 29). You can click the **Preview** button to view the database. Click Ok, to exit the Database Connection dialog box.
- Add another sort rows component **Sort rows 2**, and a hop connecting the *SSTimeDim* step. In the field specification (Figure 30), delete other fields except TIMEDAY, TIMEMONTH, TIMEYEAR fields.

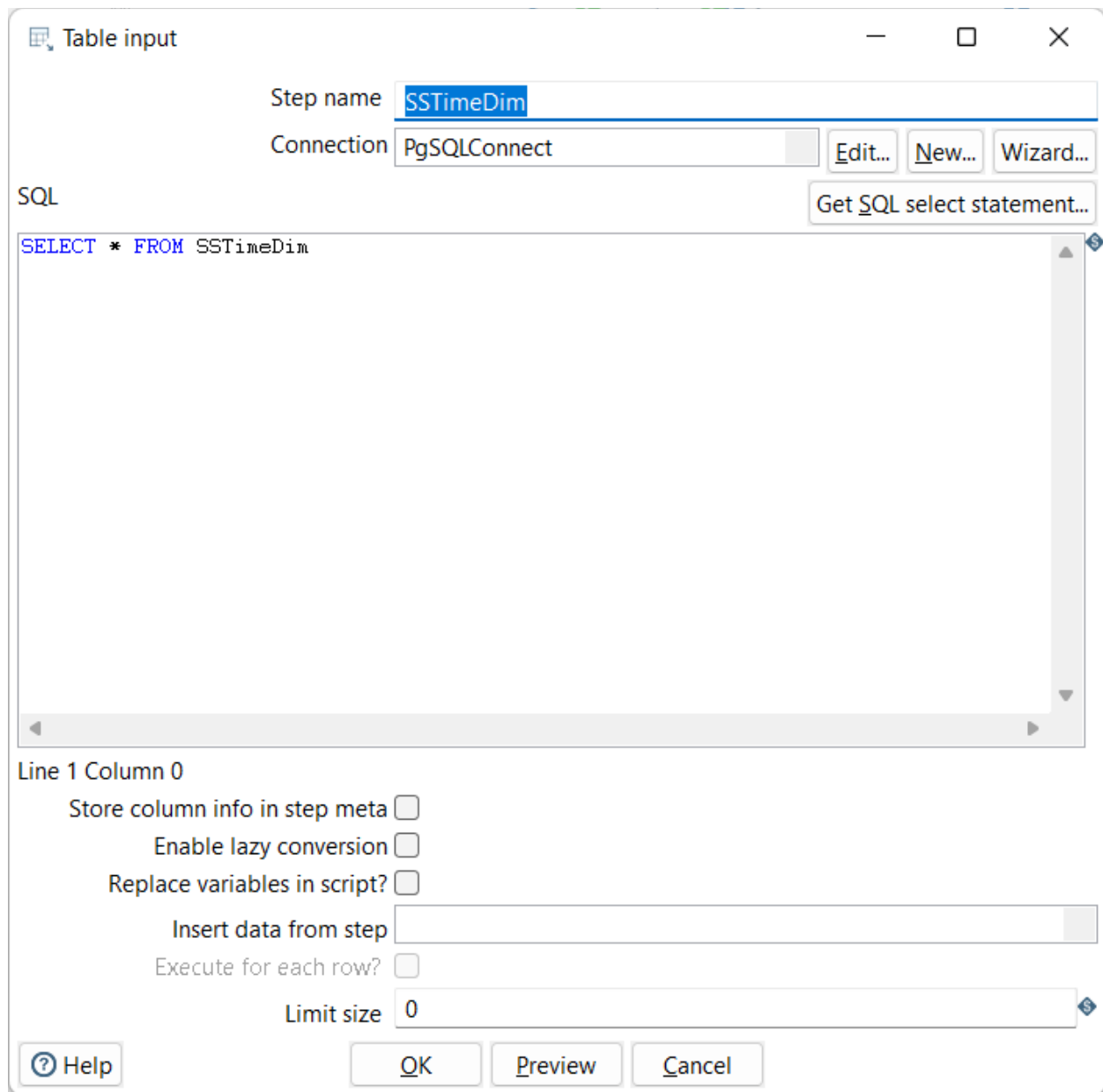


Figure 29: SQL Edit Section in Property Window of Table Input Step

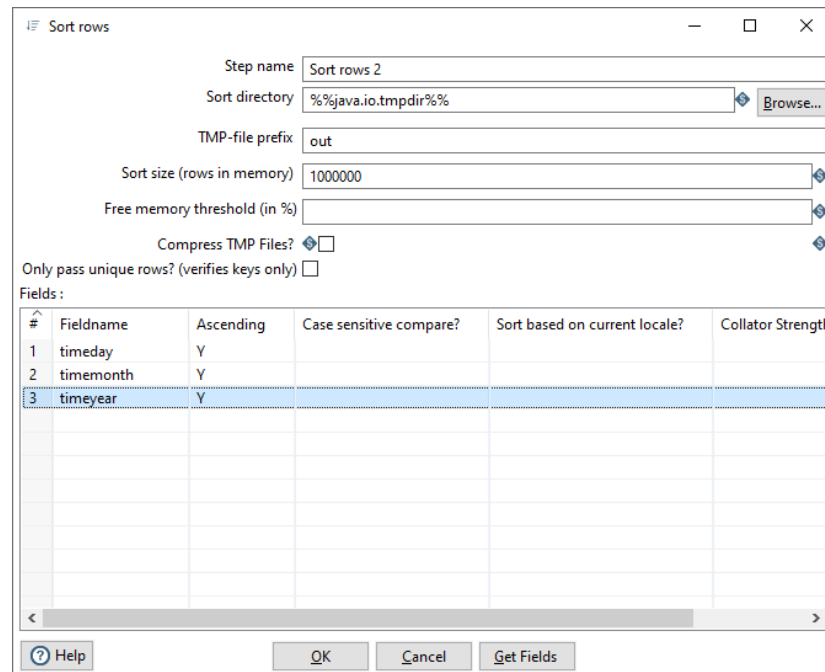


Figure 30: Property Edit Window of Sort Rows 2 Step

- 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 31).

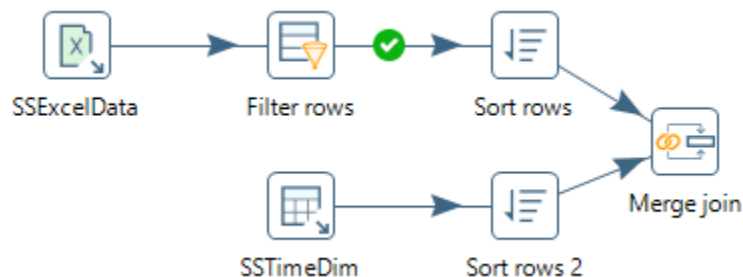


Figure 31: Two Sort Rows Steps Connected to Merge Join Step

- Double-click the Merge Join step to specify its properties (Figure 32). 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.

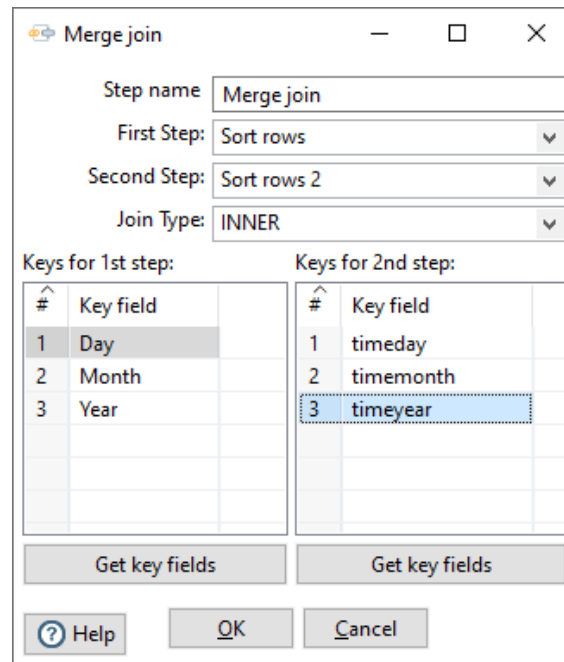


Figure 32: Property Edit Window of Merge Join Step

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

Step 2 – Inner join the *SSItem*, *SSCustomer*, and *SSStore* tables. Note that the Merge Join step requires sorted inputs. Merge Join will produce incorrect results if the inputs are not sorted or sorted inconsistently. For example, the Merge Join step in Figure 31 will generate incorrect results if one Sort Rows step sorts by day, month, and year but the other Sort Rows step sorts by year, month, and day. With multiple columns, a Sort Rows step depends on the order of fields. Consistent Sort Row steps use the same field order.

Like getting data from the *SSTimeDim* table in the previous section, inner joining these tables requires **Table Input** components. First, you should set the connection and SELECT statement for the *SSItem* table. Note that these tables should exist in your PostgreSQL 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 **ItemID** and **ITEMID**.
- Figure 33 shows all steps and hops to the Merge join 2 step.

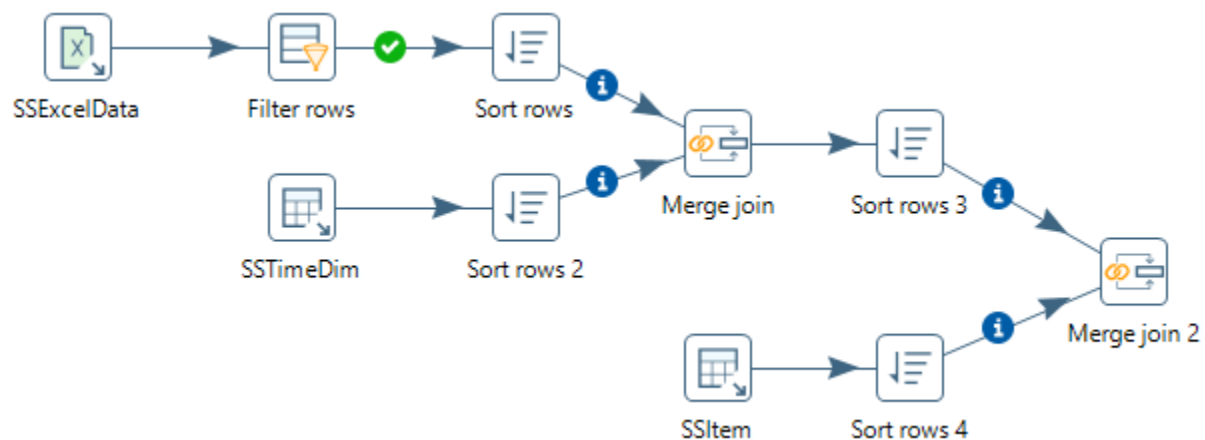


Figure 33: Transformation Design Showing Steps and Hops to the Merge Join 2 Step

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.
- Figure 34 shows all steps and hops after the Merge join 4 step.

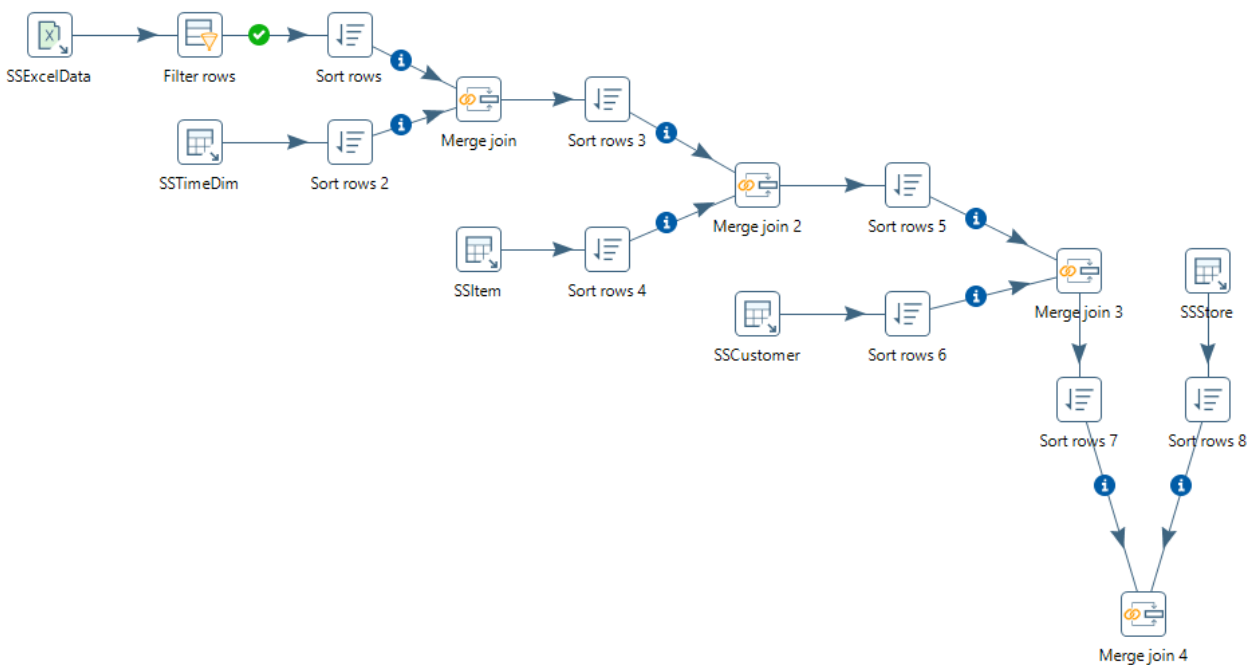


Figure 34: Transformation Design Showing Steps and Hops after the Merge Join 4 Step

5. Insert data into the SSSales table

- Under the **Design** tab, expand the contents of the **Output** step.
- Click and drag a **Table Output** step into your transformation; create a hop between the **Add sequence** and **Table Output** steps. Figure 35 shows the Table Output step (SSSales) connected to Add sequence step.

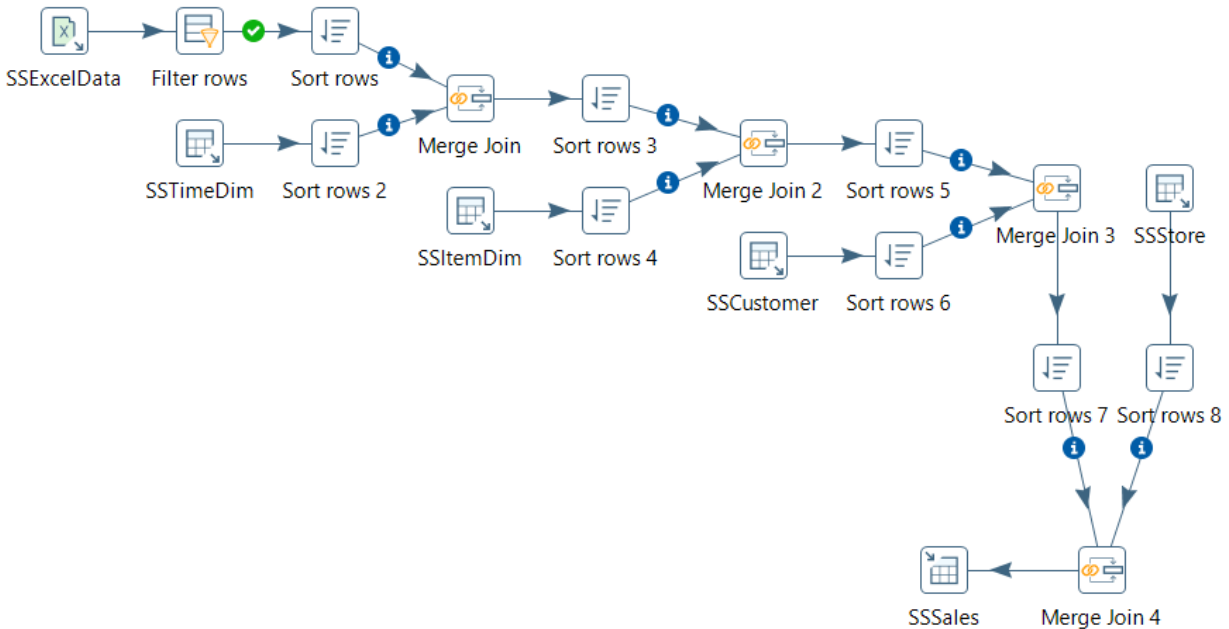


Figure 35: Connect Insert/Update Step to the Table Output Step

- Double click the **Table Output** step, to specify its properties (Figure 36). Set the **step name** as **SSSales**. Select the **connection** as **PgSQLConnect**. Type in the **Target table** as **SSSales** or click the **Browse** button and select the table from the list. Check the box for “Specify database fields”. The window should look like Figure 36.
- Click the tab “Database fields”. Then click on **Enter Field mapping** button to edit mapping. Click the **Guess** button to show default mappings. If you see extra mappings, delete them so your mappings match Figure 37. You can uncheck the “Hide assigned target fields” option to display the target fields in the Table Output step. Click **OK** after specifying the mappings.

Table output

Step name: SSSales

Connection: PgSQLConnect [Edit...] [New...] [Wizard...]

Target schema: [Browse...]

Target table: SSSales [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | **Database fields**

Partition data over tables: ☐

Partitioning field: []

Partition data per month: ☒

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field? ☐

Field that contains name of table: []

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: []

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

Figure 36: Property Edit Window of Table Output Step

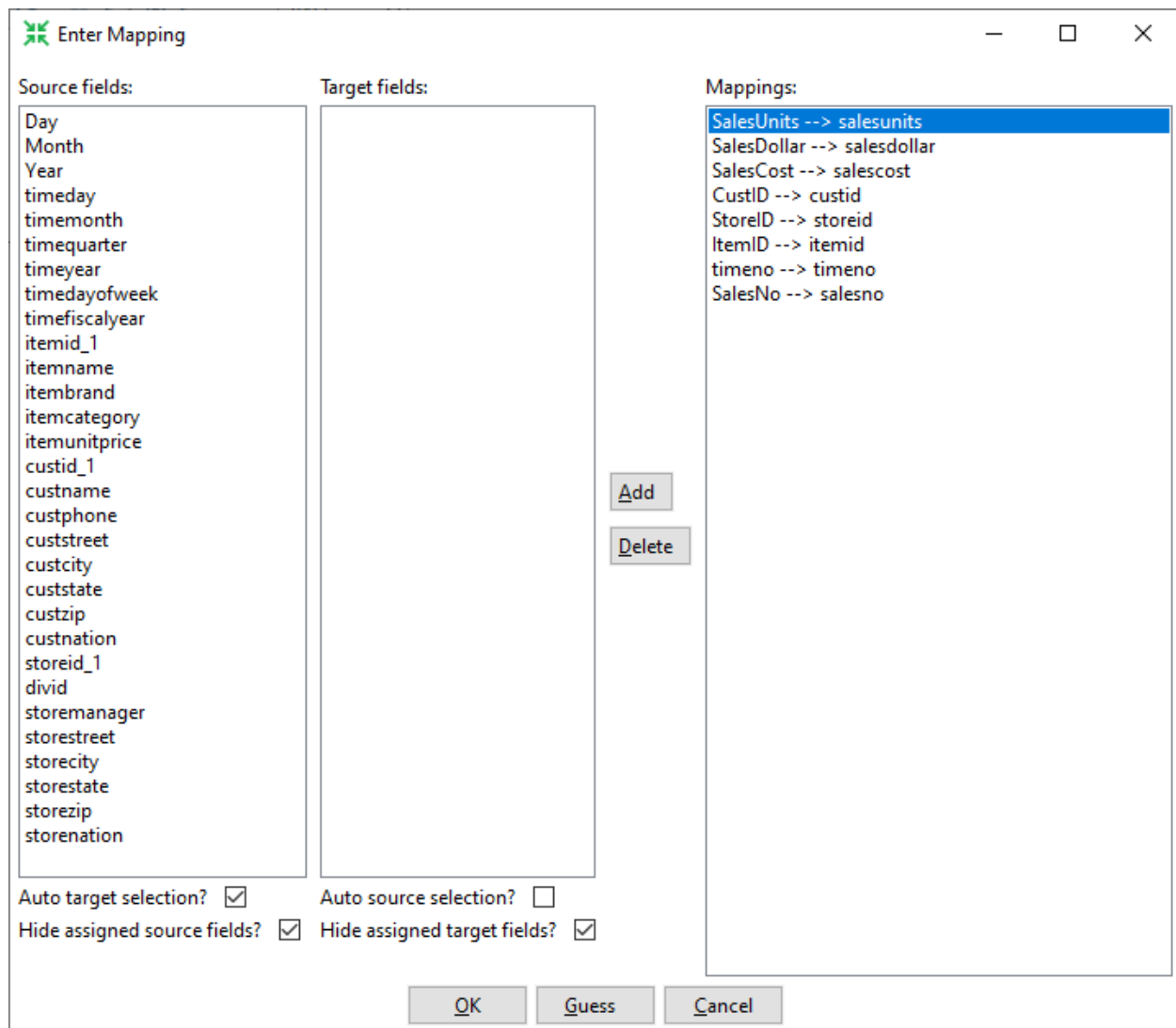


Figure 37: Mapping Edit Window

- Check the button "Don't perform any updates:". The final view of the **SSSales** step will look like Figure 38.

The screenshot shows the 'Table output' dialog box. At the top, there are fields for 'Step name' (SSSales), 'Connection' (PgSQLConnect), 'Target schema' (empty), 'Target table' (SSSales), and 'Commit size' (1000). There are also checkboxes for 'Truncate table' and 'Ignore insert errors', both of which are unchecked. A checkbox labeled 'Specify database fields' is checked. Below these options are two tabs: 'Main options' and 'Database fields'. The 'Database fields' tab is selected, showing a table titled 'Fields to insert:' with columns '#', 'Table field', and 'Stream field'. The table contains seven rows of data mapping table fields to stream fields. To the right of the table are buttons for 'Get fields' and 'Enter field mapping'. At the bottom of the dialog are buttons for '? Help', 'OK', 'Cancel', and 'SQL'.

Table output

Step name SSSales

Connection PgSQLConnect Edit... New... Wizard...

Target schema Browse...

Target table SSSales Browse...

Commit size 1000

Truncate table ☐

Ignore insert errors ☐

Specify database fields ☒

Main options Database fields

Fields to insert:


#	Table field	Stream field
1	CUSTID	CustID
2	STOREID	StoreID
3	ITEMID	ItemID
4	TIMENO	timeno
5	SALESCOST	SalesCost
6	SALESDOL...	SalesDollar
7	SALESUNITS	SalesUnits

Get fields

Enter field mapping

? Help OK Cancel SQL

Figure 38: 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 39).
- The Examine preview data window is displayed in Figure 40.

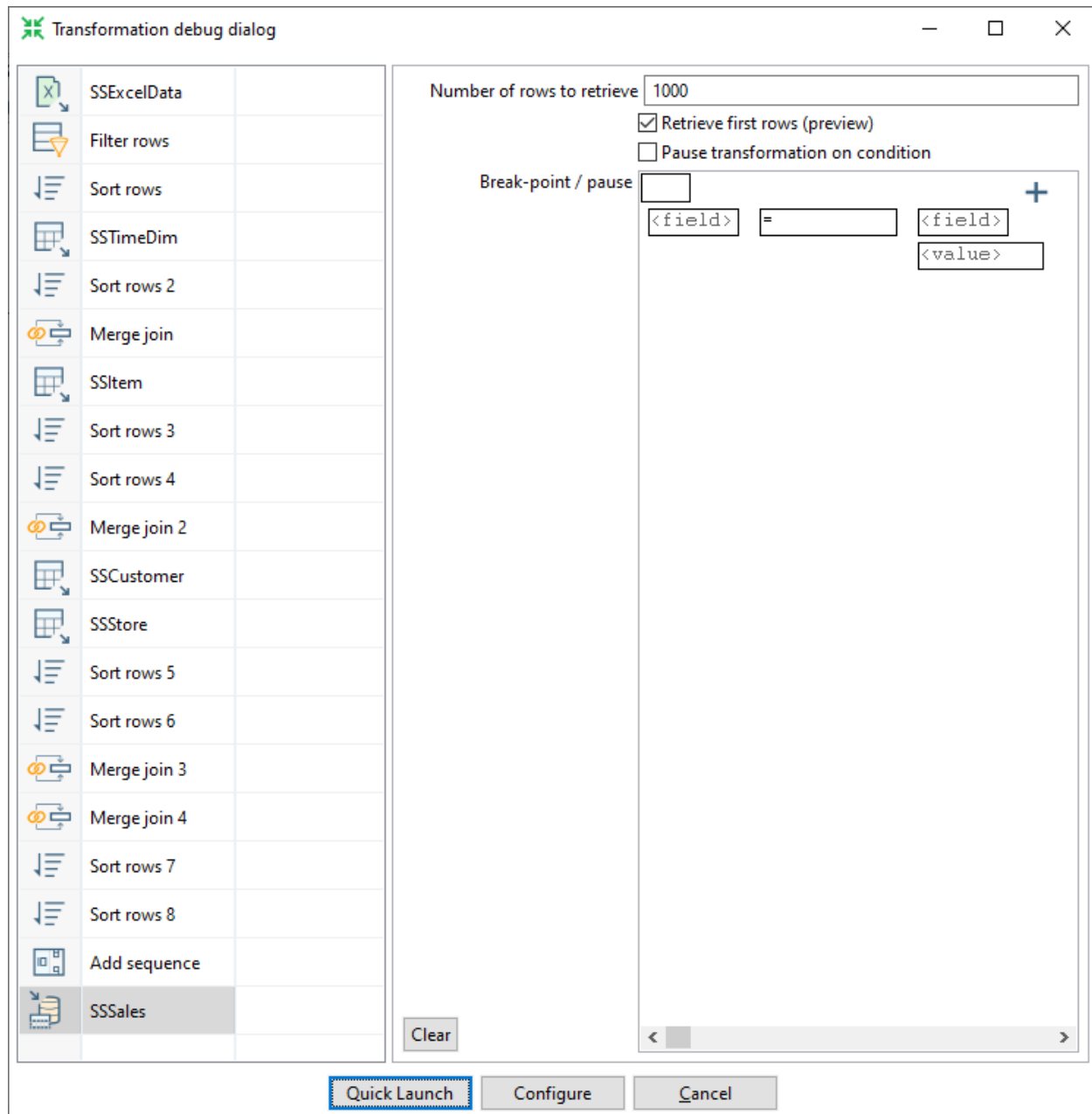
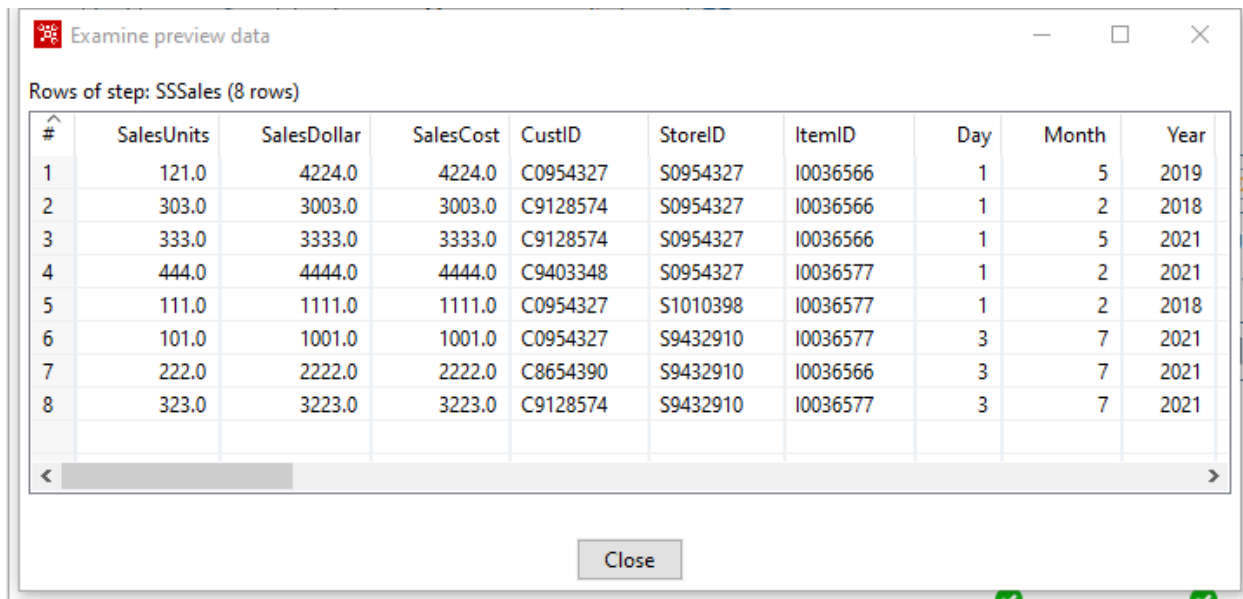


Figure 39: Transformation Debug Dialog



Examine preview data

Rows of step: SSSales (8 rows)

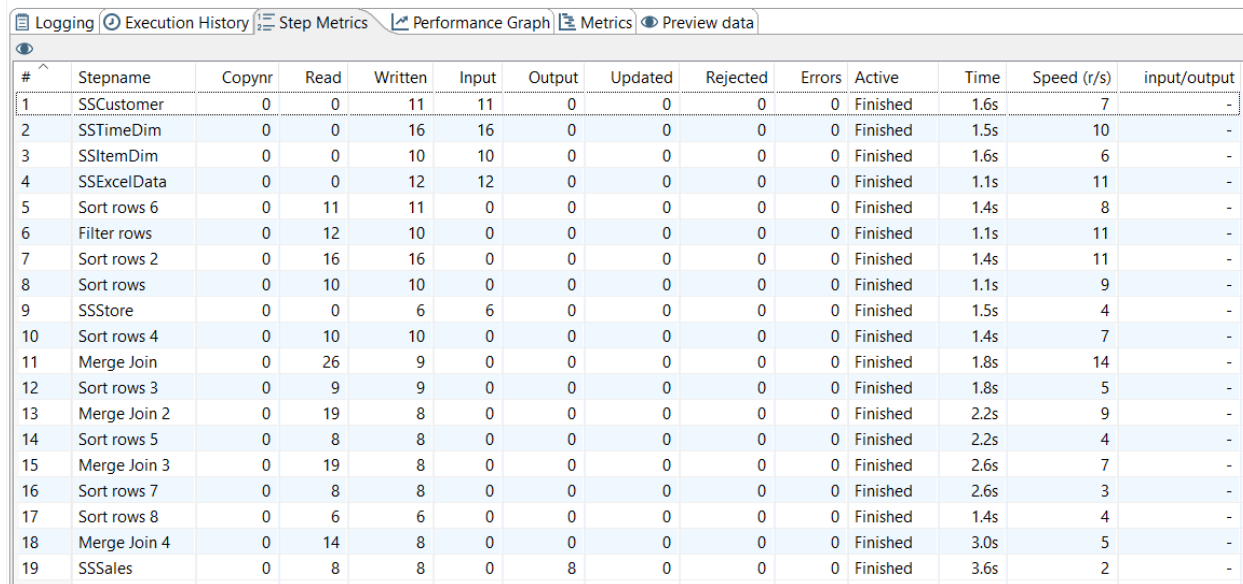
#	SalesUnits	SalesDollar	SalesCost	CustID	StoreID	ItemID	Day	Month	Year
1	121.0	4224.0	4224.0	C0954327	S0954327	I0036566	1	5	2019
2	303.0	3003.0	3003.0	C9128574	S0954327	I0036566	1	2	2018
3	333.0	3333.0	3333.0	C9128574	S0954327	I0036566	1	5	2021
4	444.0	4444.0	4444.0	C9403348	S0954327	I0036577	1	2	2021
5	111.0	1111.0	1111.0	C0954327	S1010398	I0036577	1	2	2018
6	101.0	1001.0	1001.0	C0954327	S9432910	I0036577	3	7	2021
7	222.0	2222.0	2222.0	C8654390	S9432910	I0036566	3	7	2021
8	323.0	3223.0	3223.0	C9128574	S9432910	I0036577	3	7	2021

Close

Figure 40: 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 41) shows details about the execution of each step. You should verify that the **SSSales** step has 8 output rows.

Execution Results



Logging Execution History Step Metrics Performance Graph Metrics Preview data

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	SSCustomer	0	0	11	11	0	0	0	0	Finished	1.6s	7	-
2	SSTimeDim	0	0	16	16	0	0	0	0	Finished	1.5s	10	-
3	SSItemDim	0	0	10	10	0	0	0	0	Finished	1.6s	6	-
4	SSExcelData	0	0	12	12	0	0	0	0	Finished	1.1s	11	-
5	Sort rows 6	0	11	11	0	0	0	0	0	Finished	1.4s	8	-
6	Filter rows	0	12	10	0	0	0	0	0	Finished	1.1s	11	-
7	Sort rows 2	0	16	16	0	0	0	0	0	Finished	1.4s	11	-
8	Sort rows	0	10	10	0	0	0	0	0	Finished	1.1s	9	-
9	SSStore	0	0	6	6	0	0	0	0	Finished	1.5s	4	-
10	Sort rows 4	0	10	10	0	0	0	0	0	Finished	1.4s	7	-
11	Merge Join	0	26	9	0	0	0	0	0	Finished	1.8s	14	-
12	Sort rows 3	0	9	9	0	0	0	0	0	Finished	1.8s	5	-
13	Merge Join 2	0	19	8	0	0	0	0	0	Finished	2.2s	9	-
14	Sort rows 5	0	8	8	0	0	0	0	0	Finished	2.2s	4	-
15	Merge Join 3	0	19	8	0	0	0	0	0	Finished	2.6s	7	-
16	Sort rows 7	0	8	8	0	0	0	0	0	Finished	2.6s	3	-
17	Sort rows 8	0	6	6	0	0	0	0	0	Finished	1.4s	4	-
18	Merge Join 4	0	14	8	0	0	0	0	0	Finished	3.0s	5	-
19	SSSales	0	8	8	0	8	0	0	0	Finished	3.6s	2	-

Figure 41: Step Metrics in the Execution Result Window

- Each step in the transformation should have a check mark indicating execution as shown in Figure 42.

- Connect to your PostgreSQL account (on your PC) 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 PostgreSQL *SSSales* table.
- If you do not see the extra rows, the PostgreSQL output component had a failure. To see the error, check the Logging and Step Metrics tabs of the **Execution Results** window.

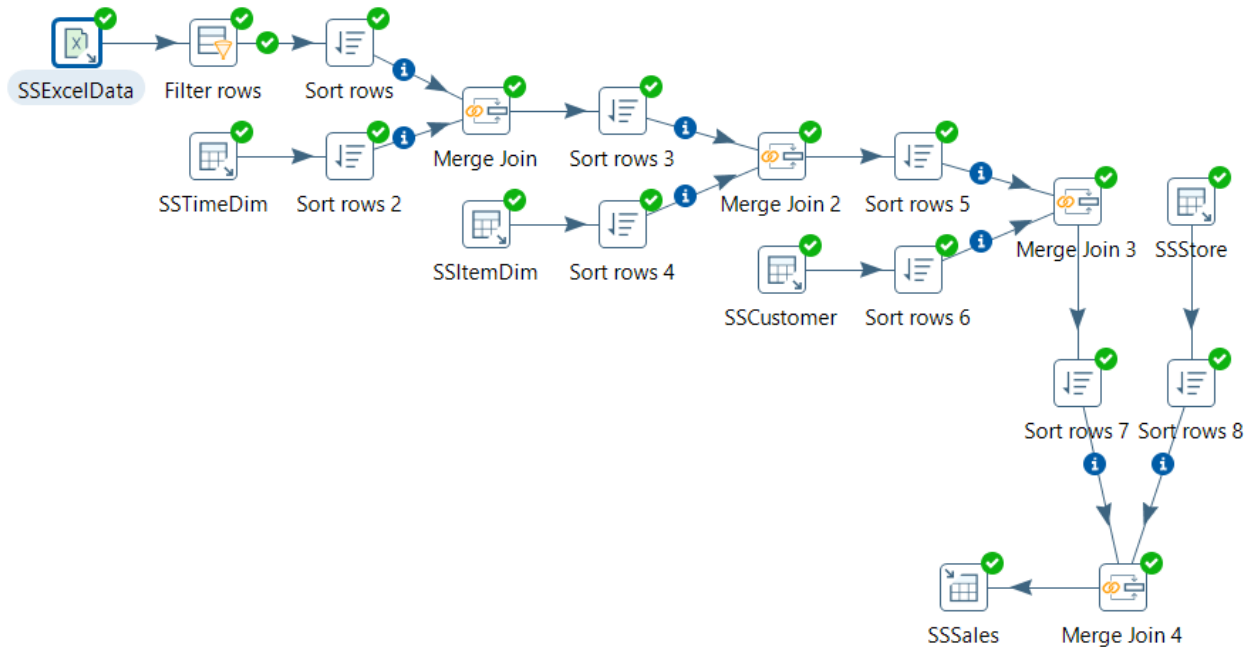


Figure 42: Transformation Design with Check Marks for Each Step

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. Create a new transformation using **File → New → Transformation** with name “SSStoreTestAccess”. Use **File → Save As ...** to save the transformation file as “SSStoreTestAccess” to a folder of your choice. Then, you will begin by loading the rows from a table in the Access database.

Step 1- Add the Access Input Step

- Under the Design tab, expand the Input step. Figure 43 shows the Design table and 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 44). In this dialog box, you specify the properties related to this step.

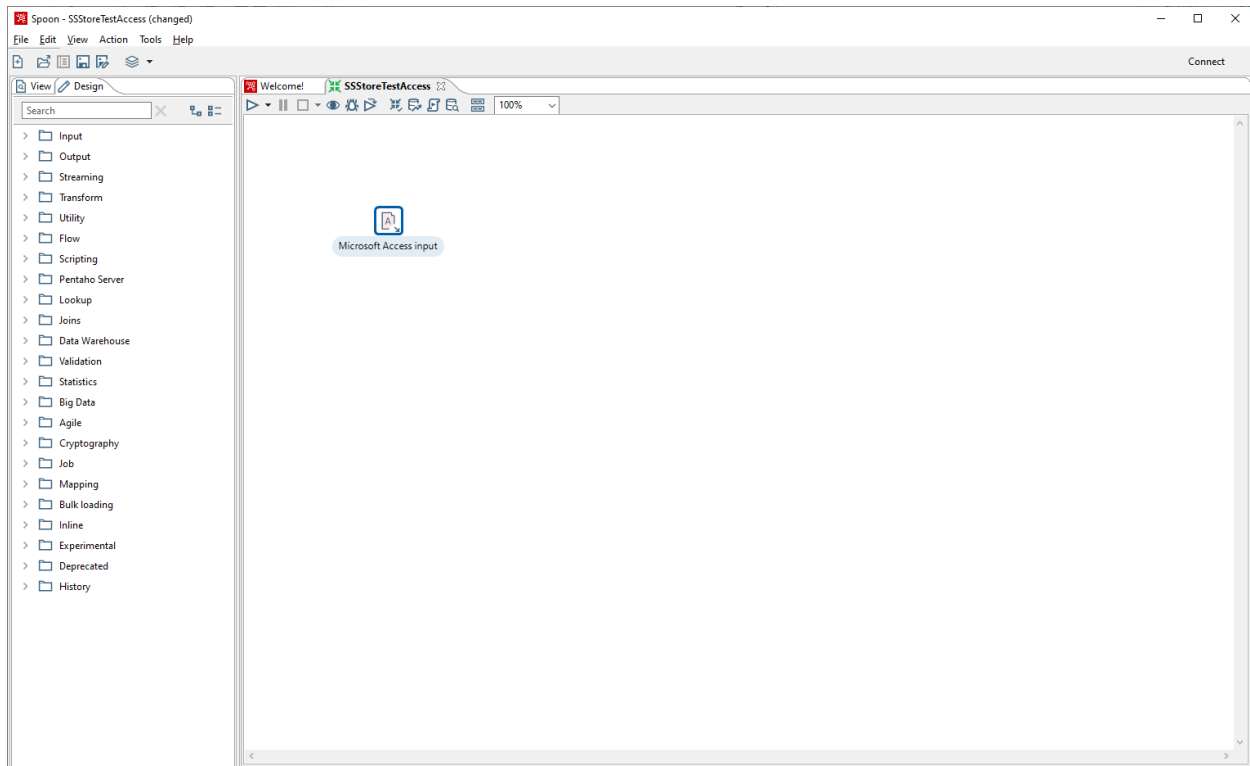


Figure 43: New Microsoft Access Input Step

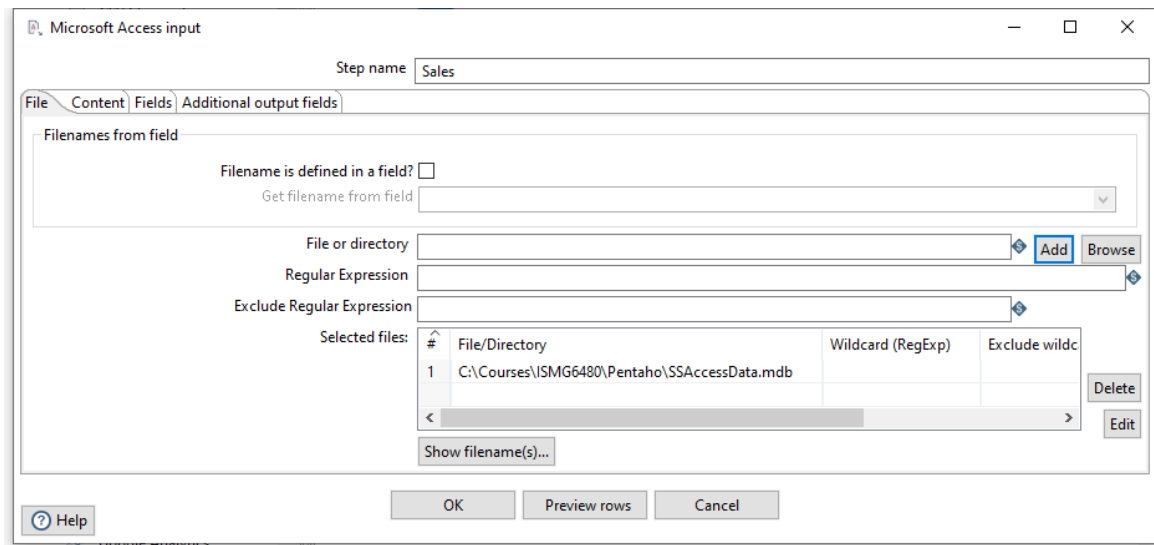


Figure 44: Property Edit Window of Microsoft Access Input Step

- Set name for the Access Input as **Sales** and specify the Access 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 45). Select **Sales** as the table name, click **OK**.

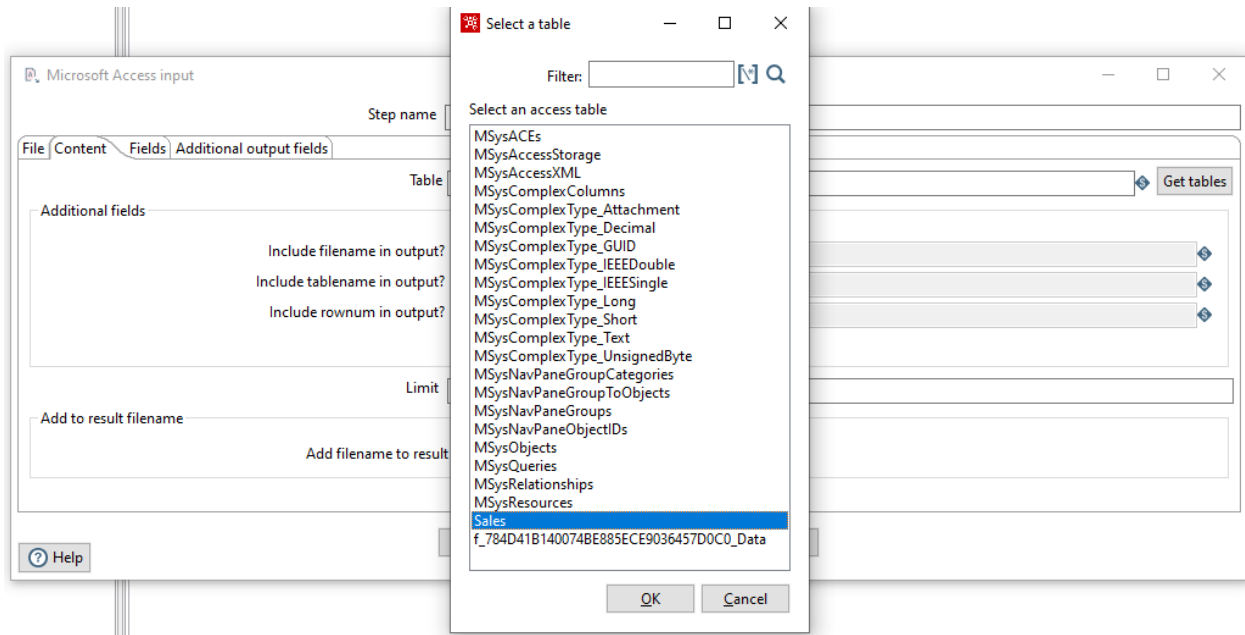


Figure 45: Table Selection Window

- In the tab named **Fields**, click the button “**Get fields**”. There will appear a list (Figure 46) showing the fields in the table named **Sales**.

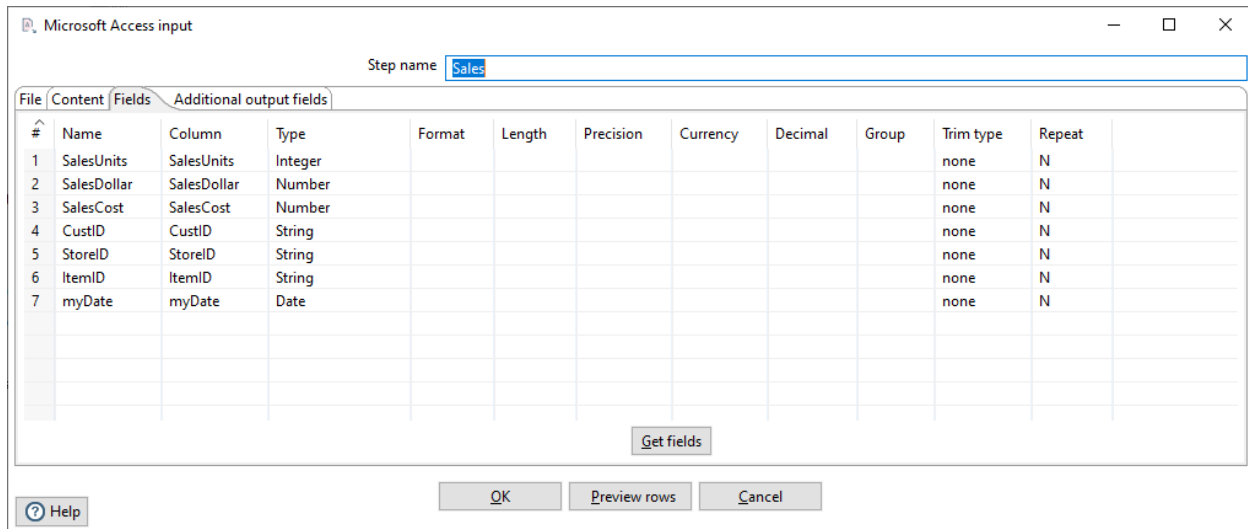


Figure 46: Fields Window for Microsoft Access Input Property Editing

- Click the button “**Preview rows**” to preview the database (Figure 47). When asked for the number of rows, type 12 and click OK.

Examine preview data

Rows of step: Sales (12 rows)

#	SalesUnits	CustID	StoreID	ItemID	myDate	SalesDollar	SalesCost
1	555	C1010398	S1010398	I0036566	2019/02/01 00:00:00.000	5555.0	5555.0
2	666	C8574932	S9432910	I0036577	2018/05/01 00:00:00.000	6666.0	6666.0
3	777	C0954327	S0954327	I0036566	2018/07/03 00:00:00.000	7777.0	7777.0
4	797	C0954327	S0954327	I0036566	2018/07/03 00:00:00.000	7997.0	7997.0
5	898	C1010398	S1010398	I0036566	2019/02/01 00:00:00.000	8998.0	8999.0
6	445	C1010398	S1010398	I0036566	2019/02/01 00:00:00.000	4455.0	5555.0
7	558	C9999999	S9432910	I0036577	2018/05/01 00:00:00.000	5885.0	6666.0
8	778	C0954327	S0954327	I0036566	2018/07/03 00:00:00.000	9997.0	9997.0
9	665	C8574932	<null>	I0036577	2018/05/01 00:00:00.000	6665.0	6666.0
10	112	C0954327	S0954327	I0036566	2018/07/03 00:00:00.000	1112.0	7777.0
11	556	C0954327	S0954327	<null>	2018/07/03 00:00:00.000	5656.0	7777.0
12	996	C1010398	S1010398	I0036566	2022/02/01 00:00:00.000	9669.0	5555.0

Close Show Log

Figure 47: Examine Preview Data Window

- Click **OK** at the bottom of the window. The input icon will change to the shape shown by Figure 48.



Figure 48: 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**.
- 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. The hop should be the main output of the **Sales** step. Figure 49 shows the transformation window after adding the new step and hop.

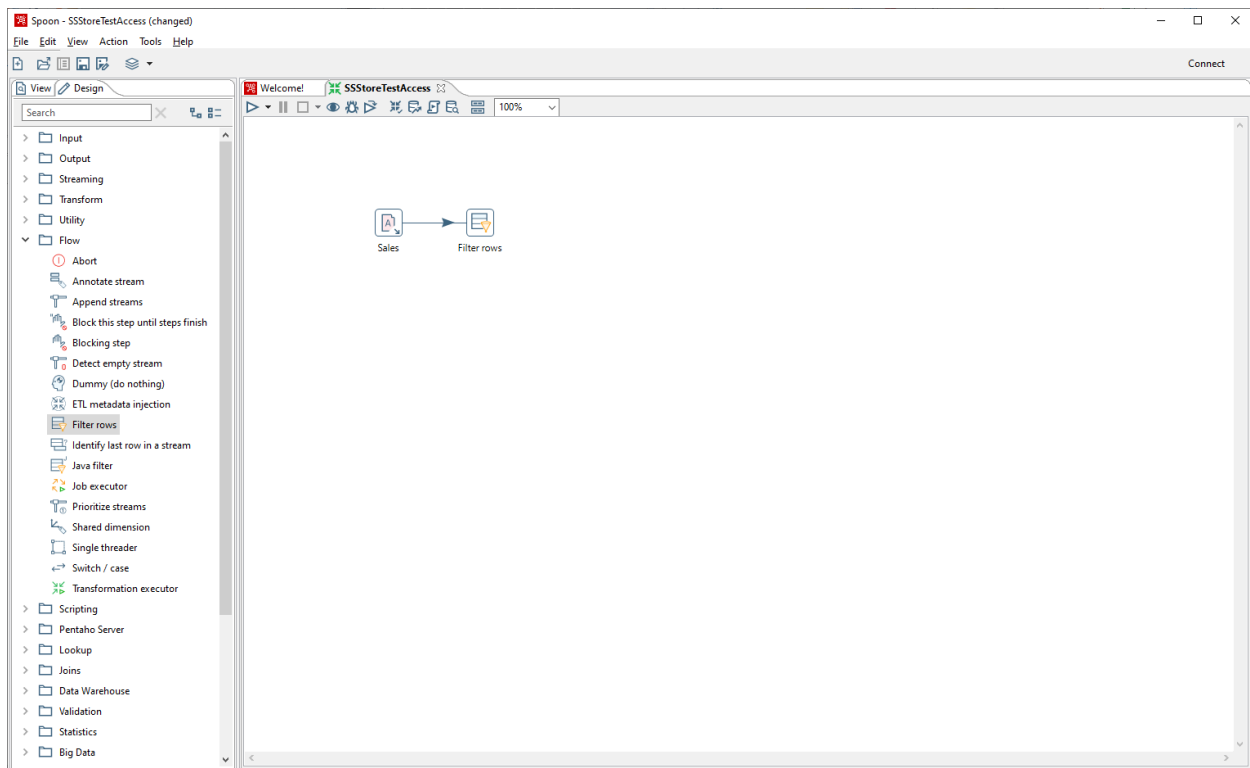


Figure 49: Access Input Step and Filter Step in Spoon

- 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 like the previous Excel transformation.
- The final view of filter conditions is shown in Figure 50. Save the transformation before adding new steps.

Filter rows

Step name: Filter Rows

Send 'true' data to step: [dropdown]

Send 'false' data to step: [dropdown]

The condition:

To edit a subcondition, simply click on it +

SalesUnits IS NOT NULL

AND

SalesDollar IS NOT NULL

AND

SalesCost IS NOT NULL

AND

CustID IS NOT NULL

AND

StoreID IS NOT NULL

AND

ItemID IS NOT NULL

AND

myDate IS NOT NULL

Help OK Cancel

Figure 50: Filter Conditions Window

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 51). Select **Result is TRUE** in the filter results selection list.



Figure 51: 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 Meta-data, click the button “**Get fields to change**”, to get the fields to change, which is shown by Figure 52. Change the **Type** of field **myDate** as **String**, change its **Format** to **dd-MM-yyyy**. Click **OK**.

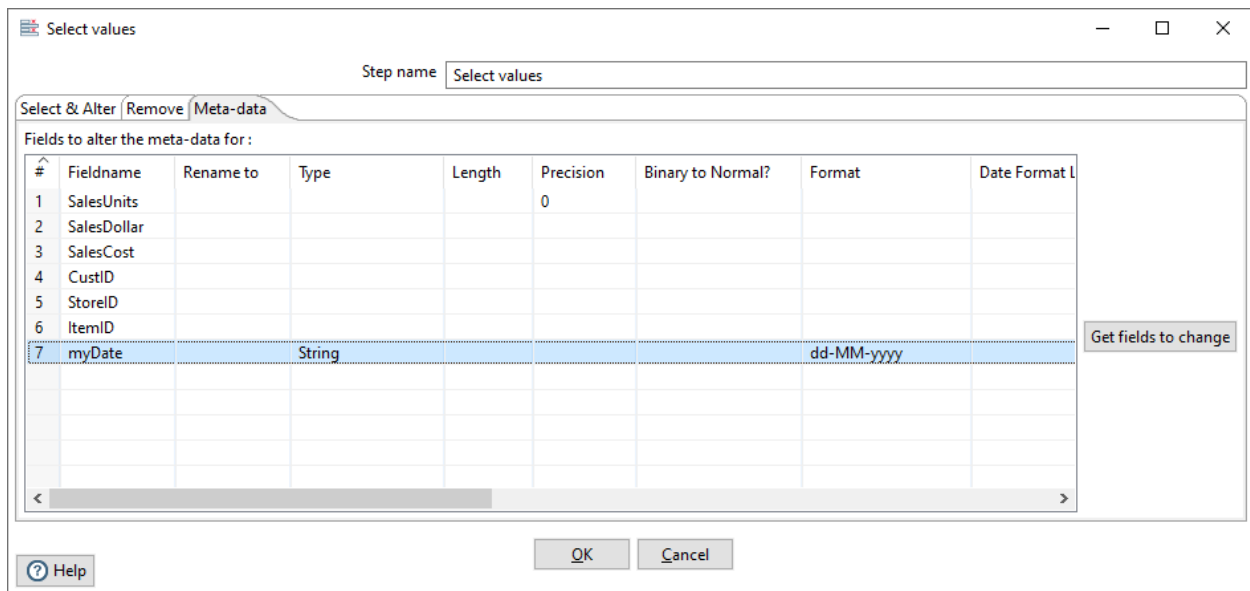


Figure 52: Meta-data Tab of Select Values Property Edit Window

- Under the **Design** tab, expand the contents of the **Transform** step.
- Click and drag a **Split fields** step into your transformation (Figure 53).
- Create a “hop” between the **Select values** step and the **Split fields** step. The hop should be the main output of the previous step.




Figure 53: Create Split Fields with Hop between Steps

- Double-click the **Split fields** step to open its edit properties dialog box (Figure 54).

- Select **myDate** in the **Field to split**, type “-” as the **Delimiter**. Type in **Day**, **Month** and **Year** in the Column named **New field**, and set their **Type** as **Integer**. Click Ok when finished.

[illegible]

Figure 54: Property Edit Window of Field Splitter Step

- Select the Split fields step in the canvas and click  , to preview this transform (Figure 55). Make sure that Split Fields step is selected from the left side panel of the transformation debug dialog and click on “**Quick Launch**” button.

Examine preview data

Rows of step: Split fields (10 rows)

#	SalesUnits	CustID	StoreID	ItemID	Day	Month	Year	SalesDollar	SalesCost
1	555	C1010398	S1010398	I0036566	1	2	2019	5555.0	5555.0
2	666	C8574932	S9432910	I0036577	1	5	2018	6666.0	6666.0
3	777	C0954327	S0954327	I0036566	3	7	2018	7777.0	7777.0
4	797	C0954327	S0954327	I0036566	3	7	2018	7997.0	7997.0
5	898	C1010398	S1010398	I0036566	1	2	2019	8998.0	8999.0
6	445	C1010398	S1010398	I0036566	1	2	2019	4455.0	5555.0
7	558	C9999999	S9432910	I0036577	1	5	2018	5885.0	6666.0
8	778	C0954327	S0954327	I0036566	3	7	2018	9997.0	9997.0
9	112	C0954327	S0954327	I0036566	3	7	2018	1112.0	7777.0
10	996	C1010398	S1010398	I0036566	1	2	2022	9669.0	5555.0

Close

Figure 55: Examine Preview Data Window

8. Lookup Columns from the PostgreSQL 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 PostgreSQL tables to ensure loaded data does not contain invalid foreign keys. This part of the exercise resembles details in Section 4.

Step 1 – Access the *SSTimeDim* table from PostgreSQL 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*.
- For the Connection field, select your existing connection to PostgreSQL 14.1 if it is available in the connection list. Otherwise, click “**New**” next to the connection field. Provide the settings for connecting to the database as shown in the Figure 27.
- Connection Name: PgSQLConnect (You can use another name if you want)
 Connection Type: PostgreSQL
 Host Name: localhost
 Database Name: StoreSales14 (unless you used a different name in PostgreSQL)
 Port Number: 5433
 Username: postgres (default administrative user for PostgreSQL)
 Password: <blank> (or the password used when you installed PostgreSQL)

Access: Native (JDBC)

- 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. Make sure that the steps are in the same order (day, month, year) in each step part. Then click OK.
- Now, we have finished inner join between the Access table and *SSTimeDim* table.
- Figure 56 shows the transformation design with all steps and hops to the Merge join step.

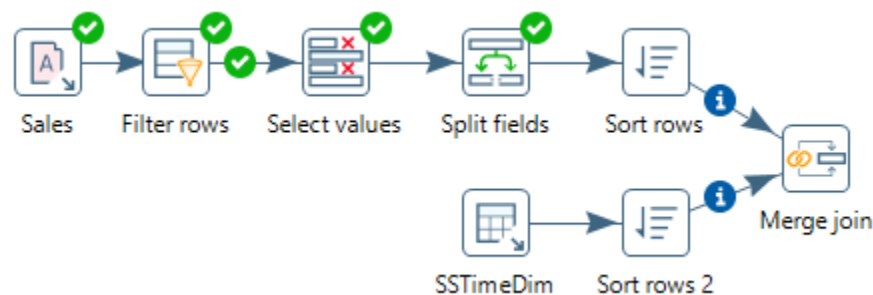


Figure 56: Transformation Design with Steps and Hops to the Merge Join Step

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 Access file) and *ITEMID* (from Database) fields.
- For *SSCustomer* step, connect *CustID* (from Access file) and *CUSTID* (from Database) fields.
- For *SSStore* step, connect *StoreID* (from Accessfile) and *STOREID* (from Database) fields.
- Figure 57 shows the transformation design for steps and hops to the Merge join 4 step.

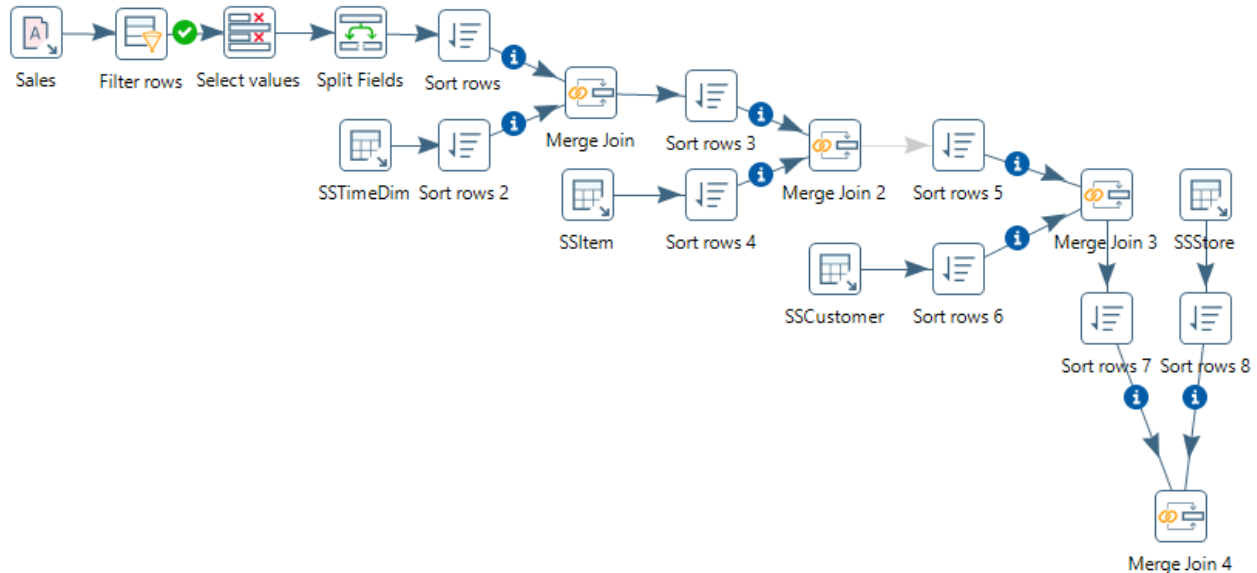


Figure 57: Transformation Design with Steps and Hops to the Merge Join 4 Step

9. Insert rows into the SSSales table

- Under the **Design** tab, expand the contents of the **Output** step.
- Click and drag and **Table Output** step into your transformation; create a hop between the **Merge Join 4** and **Table Output** steps. Figure 58 shows the connection in the transformation design pane.
- Double click the **Table Output** component, to specify its properties. Set the **step name** as **SSSales**. Select the **connection** as **PgSQLConnect**. Type in the **Target table** as **SSSales** as shown in Figure 59.
- Click the tab “Database fields”. Then click on **Enter Field mapping** button to edit mapping. In the Enter Mapping window, you should click the **Guess** button to show default mappings. Compare the field mapping to the final window showing the database fields in Figure 60. Then click **OK**.

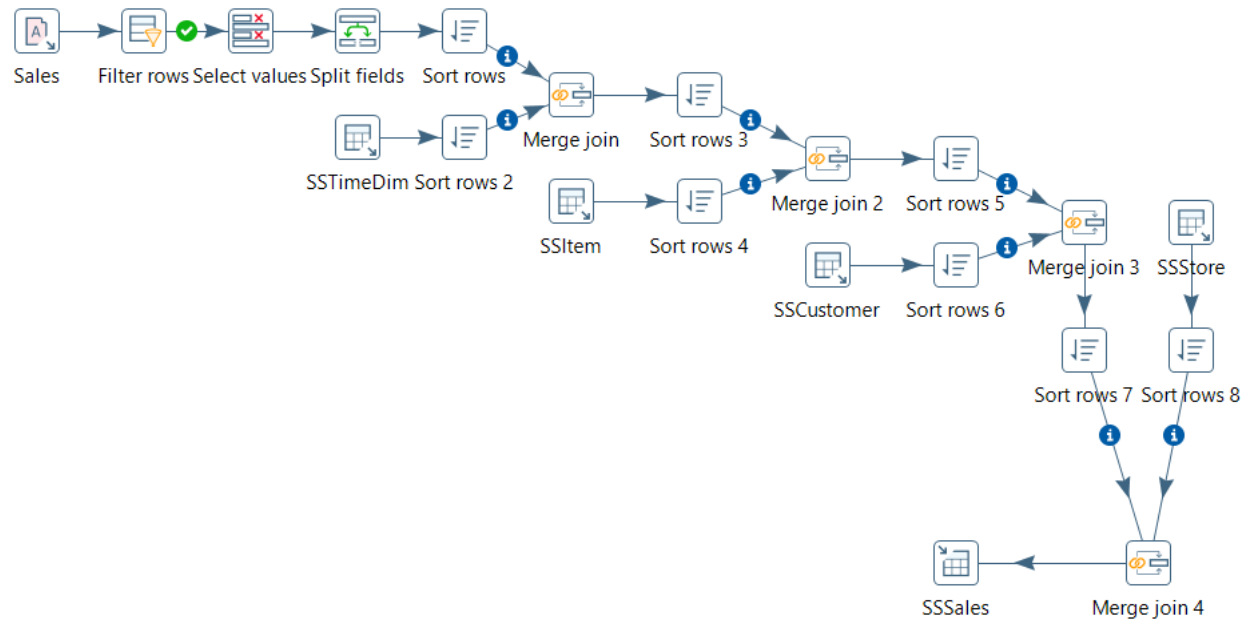


Figure 58: Connect Table Output Step to Merge join 4 Step

The screenshot shows the 'Table output' step configuration window. The 'Step name' is 'SSSales'. The 'Connection' is 'PgSQLConnect'. The 'Target schema' is empty, and the 'Target table' is 'SSSales'. The 'Commit size' is '1000'. The 'Truncate table' checkbox is unchecked. The 'Ignore insert errors' checkbox is unchecked. The 'Specify database fields' checkbox is checked. The 'Main options' tab is selected, showing the following settings:

- 'Partition data over tables' checkbox is unchecked.
- 'Partitioning field' dropdown is empty.
- 'Partition data per month' radio button is selected.
- 'Partition data per day' radio button is unselected.
- 'Use batch update for inserts' checkbox is checked.
- 'Is the name of the table defined in a field?' checkbox is unchecked.
- 'Field that contains name of table:' dropdown is empty.
- 'Store the tablename field' checkbox is checked.
- 'Return auto-generated key' checkbox is unchecked.
- 'Name of auto-generated key field' dropdown is empty.

At the bottom, there are buttons for 'Help', 'OK', 'Cancel', and 'SQL'.

Figure 59: Table Output Step Window showing Main Options

Table output

Step name: **SSSales**

Connection: PgSQLConnect [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: **SSSales** [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	CUSTID	CustID
2	STOREID	StoreID
3	ITEMID	ItemID
4	TIMENO	timeno
5	SALESCOST	SalesCost
6	SALESDOL...	SalesDollar
7	SALESUNITS	SalesUnits

[Get fields]

[Enter field mapping]

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

Figure 60: Completed Table Output Step Window showing Database Fields

- Select the **SSSales** step and run a preview by clicking on . In the transformation debug dialog click on **Quick Launch** button. Figure 61 shows the executed transformation. Figure 62 shows result rows added to the *SSSales* table after execution of the transformation. The Step Metrics tab (Figure 63) shows that 8 rows were inserted into the *SSSales* table in the Output column.
- Connect to your PostgreSQL account (on your PC) 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 existing after the Excel transformation execution (192 original rows and 8 rows from the Excel transformation).

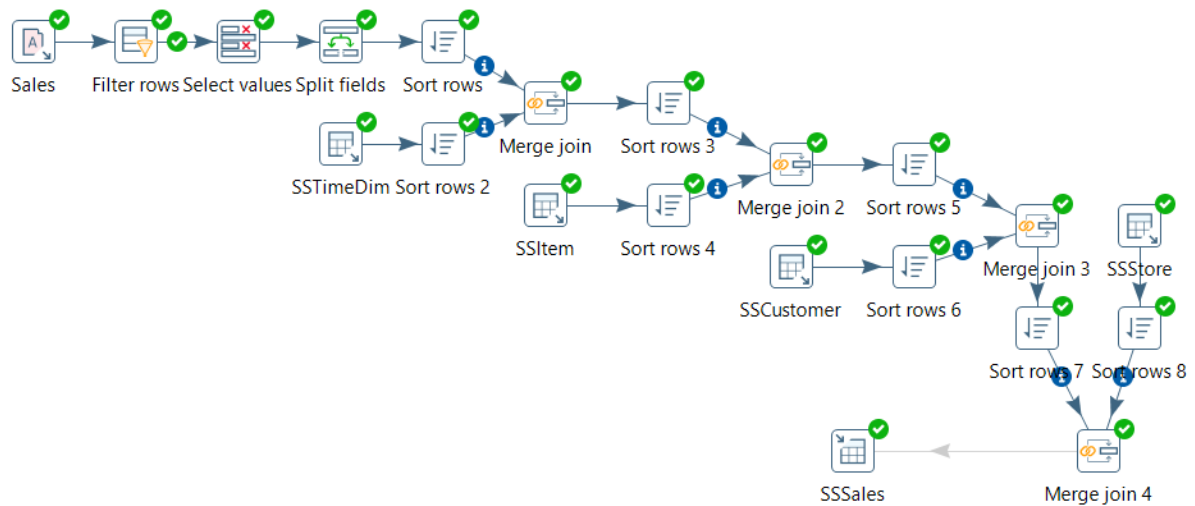


Figure 61: Executed Transformation

Examine preview data									
Rows of step: SSSales (8 rows)									
#	SalesUnits	CustID	StoreID	ItemID	Day	Month	Year	SalesDollar	SalesCost
1	777	C0954327	S0954327	I0036566	3	7	2018	7777.0	7777.0
2	797	C0954327	S0954327	I0036566	3	7	2018	7997.0	7997.0
3	778	C0954327	S0954327	I0036566	3	7	2018	9997.0	9997.0
4	112	C0954327	S0954327	I0036566	3	7	2018	1112.0	7777.0
5	555	C1010398	S1010398	I0036566	1	2	2019	5555.0	5555.0
6	898	C1010398	S1010398	I0036566	1	2	2019	8998.0	8999.0
7	445	C1010398	S1010398	I0036566	1	2	2019	4455.0	5555.0
8	666	C8574932	S9432910	I0036577	1	5	2018	6666.0	6666.0

Figure 62: Preview Data for the SSSales Step in the Access Transformation

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data													
#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	input/output
1	Sales	0	0	12	12	0	0	0	0	Finished	0.1s	164	-
2	SSTimeDim	0	0	16	16	0	0	0	0	Finished	0.0s	356	-
3	SSCustomer	0	0	11	11	0	0	0	0	Finished	0.0s	244	-
4	Filter rows	0	12	10	0	0	0	0	0	Finished	0.1s	113	-
5	Select values	0	10	10	0	0	0	0	0	Finished	0.1s	73	-
6	SSItem	0	0	10	10	0	0	0	0	Finished	0.0s	227	-
7	Sort rows 2	0	16	16	0	0	0	0	0	Finished	0.1s	271	-
8	Split fields	0	10	10	0	0	0	0	0	Finished	0.2s	60	-
9	Sort rows	0	10	10	0	0	0	0	0	Finished	0.2s	50	-
10	SSStore	0	0	6	6	0	0	0	0	Finished	0.0s	136	-
11	Sort rows 6	0	11	11	0	0	0	0	0	Finished	0.1s	190	-
12	Sort rows 4	0	10	10	0	0	0	0	0	Finished	0.1s	169	-
13	Merge join	0	26	9	0	0	0	0	0	Finished	0.6s	45	-
14	Sort rows 3	0	9	9	0	0	0	0	0	Finished	0.6s	15	-
15	Merge join 2	0	19	9	0	0	0	0	0	Finished	1.0s	19	-
16	Sort rows 5	0	9	9	0	0	0	0	0	Finished	1.0s	9	-
17	Merge join 3	0	20	8	0	0	0	0	0	Finished	1.4s	14	-
18	Sort rows 7	0	8	8	0	0	0	0	0	Finished	1.4s	6	-
19	Sort rows 8	0	6	6	0	0	0	0	0	Finished	0.1s	103	-
20	Merge join 4	0	14	8	0	0	0	0	0	Finished	1.8s	8	-
21	SSSales	0	8	8	0	8	0	0	0	Finished	1.8s	4	-

Figure 63: Step Metrics in the Execution Result Window for the Access Transformation