# Data Integration Exercise with PDI and Oracle Cloud

In the data integration exercise, you will use Pentaho Data Integration (PDI) to transform two data sources and load data into an Oracle 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 tables. Before starting the exercise, you should 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.

This document provides instructions for using an Oracle cloud database with a Wallet connection. You should have installed PDI and completed the steps in the document "Configuring Pentaho Data Integration to use the Oracle Wallet" before starting on this exercise. Make sure that you have tested the Oracle Wallet connection in PDI using the Stores Sales tables.

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

This tutorial uses the community edition (CE) of PDI. The latest version (9.3) was installed in June 2022 using SourceForge <a href="https://sourceforge.net/projects/pentaho/">https://sourceforge.net/projects/pentaho/</a>. Figure 1 shows the launch page of the latest version.

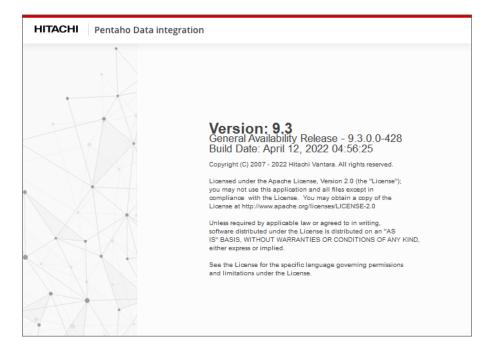


Figure 1: Pentaho Data Integration Welcome Window

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.

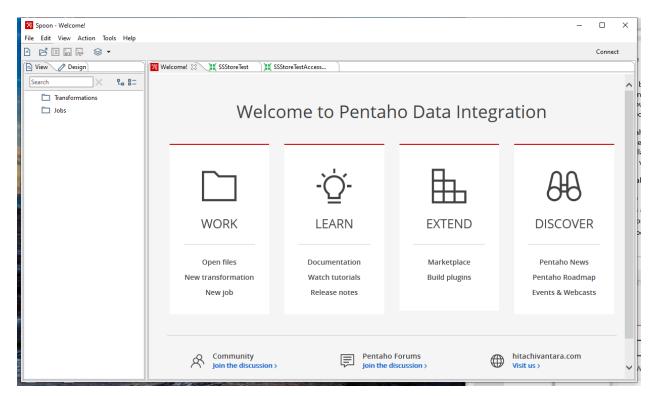


Figure 2: Spoon Opening Window

# 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 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.

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 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.

## Adding a JDBC Driver

Pentaho Data Integration uses database drivers to connect to databases. Installation of Pentaho Data Integration provides a standard set of database drivers. For Oracle Cloud, the standard set of database drivers do not include the driver to connect to the Oracle Cloud Database Server. Thus, you need to download and install the database drive. Please refer to course document "Configuring Pentaho Data Integration to use the Oracle Wallet" to install JDBC and connection to the database on Oracle Cloud.

## 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? You can fine more details about PDI in the course notes.

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 3 and 4. 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 step (Merge Join, Merge Join 2, Merge Join 3, and Merge Join 4 in Figures 3 and 4) 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 3, 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 create a primary key value.

The Access transformation (Figure 4) 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 parsed for the Merge join step. The Merge join step matches on the year, month, and day components of a date value.

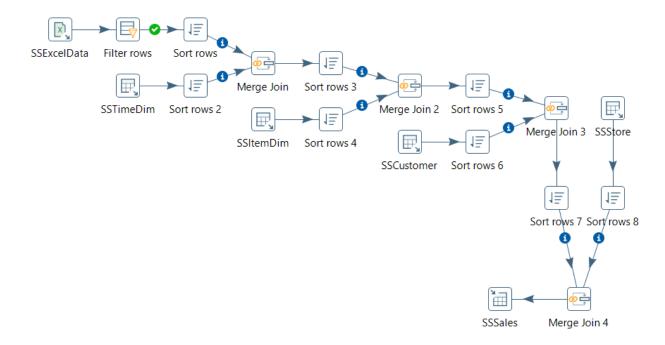


Figure 3: Transformation using Microsoft Excel Input

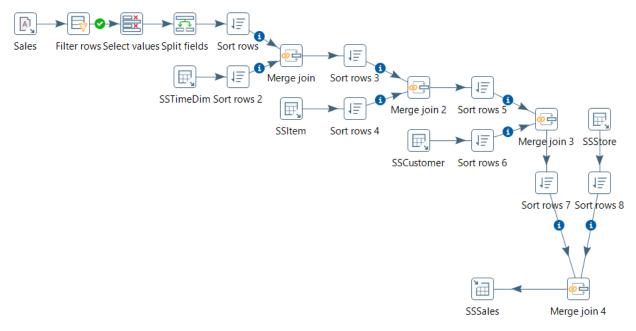


Figure 4: Transformation using Microsoft Access Input

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 5) for examining the input (Input Fields ...) and output (Output fields ...) fields on the stream for that step. Selecting Input Fields ... for the Sort rows step of the Excel transformation (Figure 3) provides details of the stream input to the step as shown in Figure 6. All fields originate (Step origin) in the SSExcelData step.

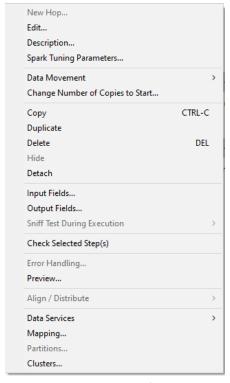


Figure 5: Menu Choices for Each Step

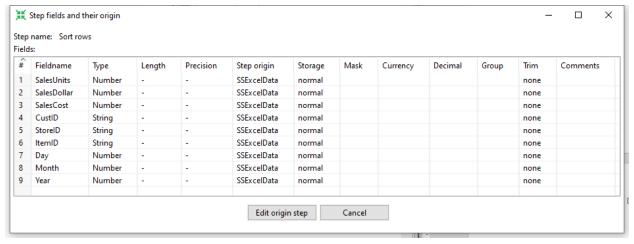


Figure 6: Stream Fields for the Sort rows Step

# 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 7) displayed after selecting the **New File** button.

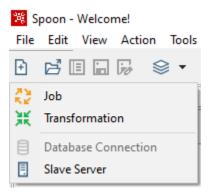


Figure 7: Spoon New File List

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**  $\rightarrow$  **Save.** You will see the empty transformation window in the Spoon (Figure 8).

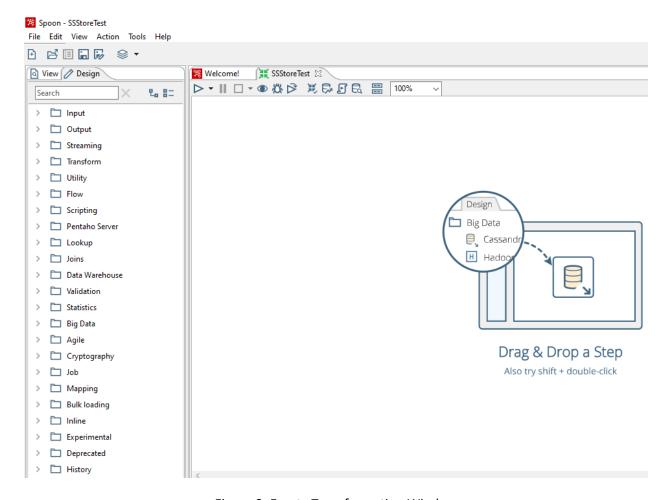


Figure 8: Empty Transformation Window

### Step 4 – Create the Excel Input step:

o Under the Design tab, expand the Input step (Figure 9).

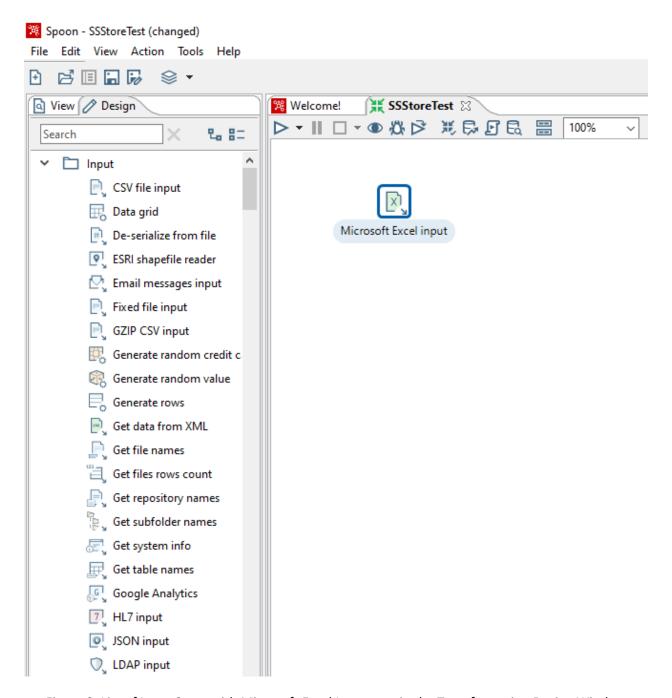


Figure 9: List of Input Steps with Microsoft Excel Input step in the Transformation Design Window

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

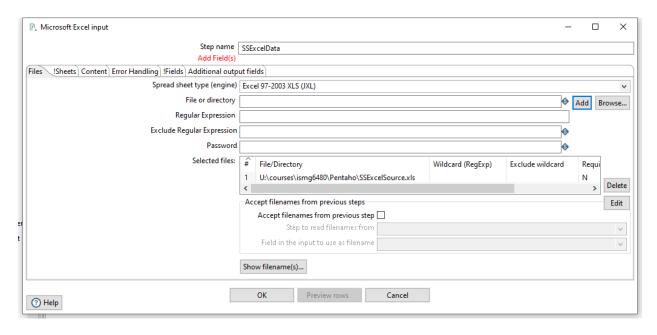


Figure 10: Files Window for Microsoft Excel Input Property Editing

- Set name for the Excel Input as SSExcelData and specify the Excel data source path in the Files tab.
- o 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 11) 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 12.

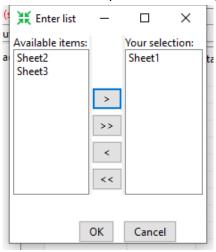


Figure 11: Sheet Specification Window

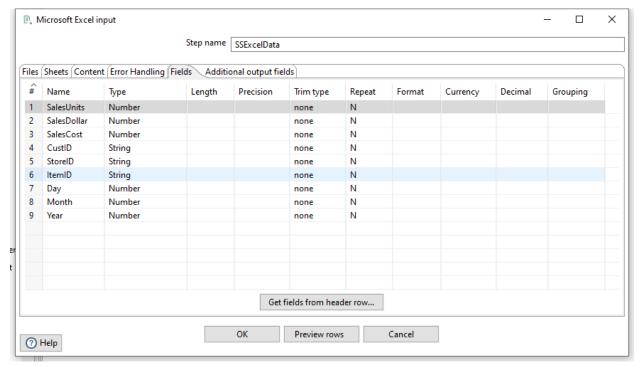


Figure 12: Fields Window for Microsoft Excel Input Property Editing

- To ensure that the Excel input can read rows from the associated worksheet, select the Preview rows button. Figure 13 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 14.

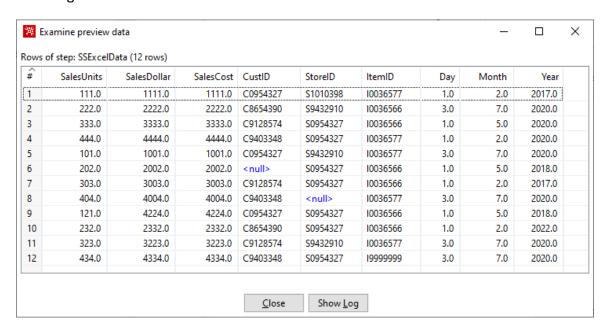


Figure 13: Preview Rows Window



Figure 14: SSExcelData 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.

- $\circ$  Add a Filter Rows step to your transformation. Under the **Design** table, go to **Flow**  $\rightarrow$  **Filter Rows**.
- 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 15).



Figure 15: 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 16) appears. Drag the hop painter icon from the source step to your target step.



Figure 16: Hover Menu

 Double-click the Filter Rows step. The Filter Rows edit properties dialog box appears (Figure 17).

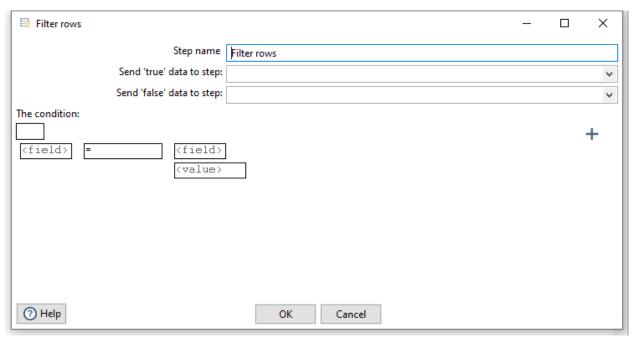


Figure 17: Property Edit Window of Filter Step

- 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.
- o In the Fields: dialog box (Figure 18) select SalesUnits and click OK.
- Click on the comparison operator (Figure 19) (set to = by default) and select the IS NOT NULL function and click OK.
- $\circ$  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"
- o 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 20.
- Save your transformation.

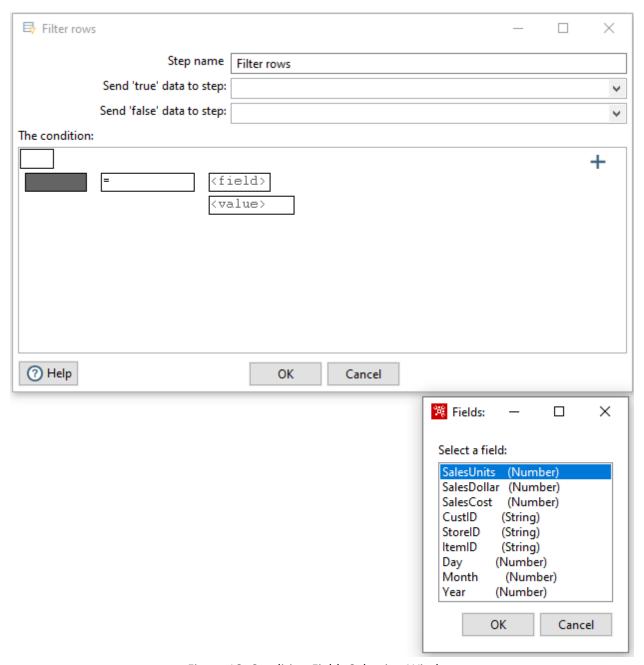


Figure 18: Condition Fields Selection Window

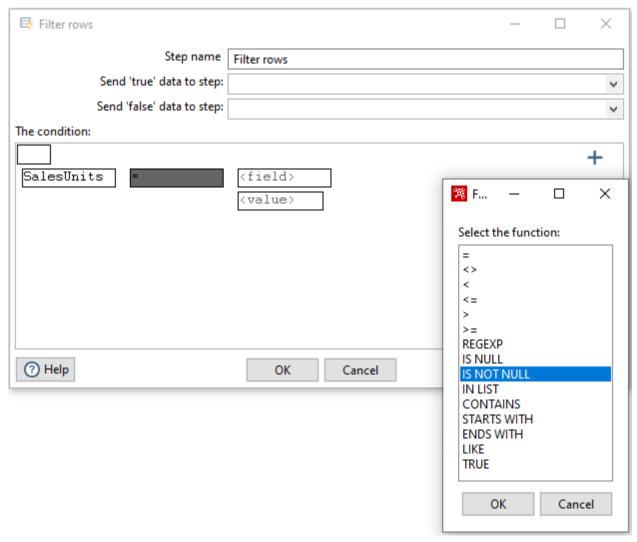


Figure 19: Comparison Operator List

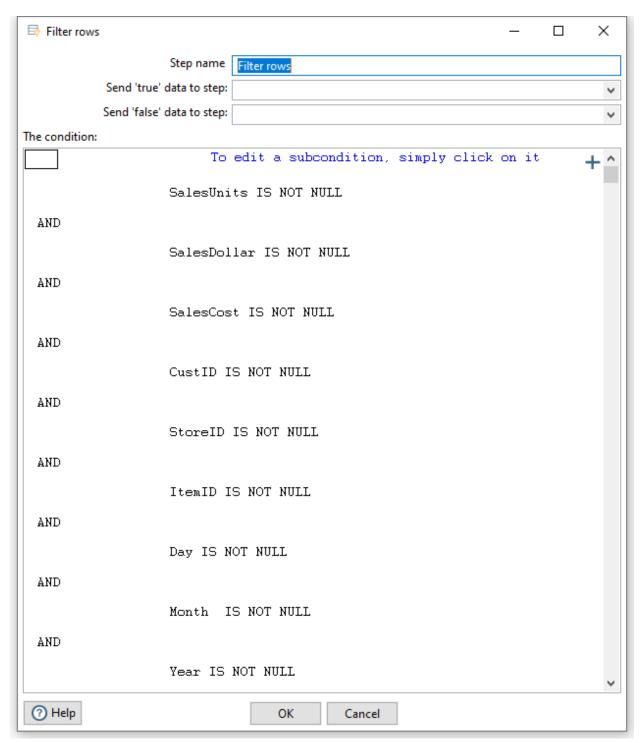


Figure 20: Filter Conditions Window

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

o 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 21).

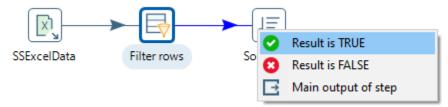


Figure 21: Filter Results Selection List

 Double-click the Sort Rows step to open its edit properties dialog box (Figure 22). Click "Get Fields" to obtain the fields. Delete other fields except the Day, Month and Year fields. Then click Ok.

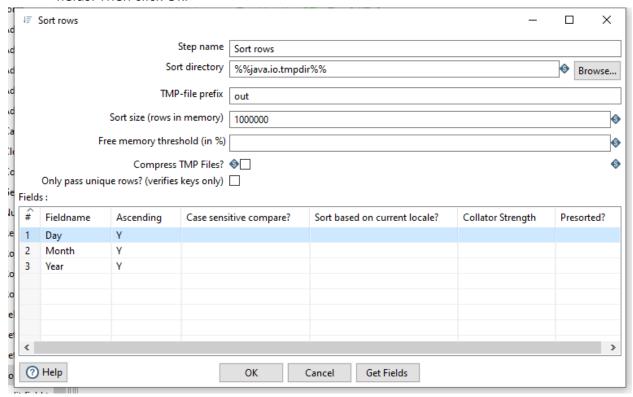


Figure 22: Property Edit Window of Sort Rows Step

# 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** category.
- Click and drag a Table Input step into your transformation.
- Double-click the Table Input step to open its edit properties dialog box (Figure 23).
- o Rename your Table Input step to SSTimeDim.

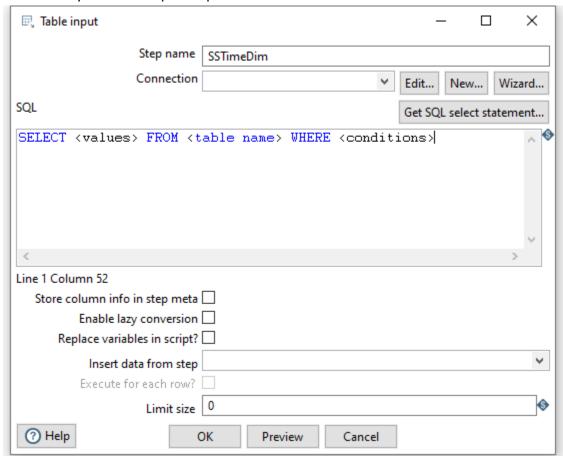


Figure 23: Property Edit Window of Table Input Step

- 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 24. You can choose any name for the connection.
- IMPORTANT: Before setting the connection information, you should first configure the JDBC driver according to the instructions provided in the course document entitled "Configuring Pentaho Data Integration to use the Oracle Wallet". Note that host name and port number are blank. The Database Name is only partially shown in Figure 24. The full value for database name is shown below.

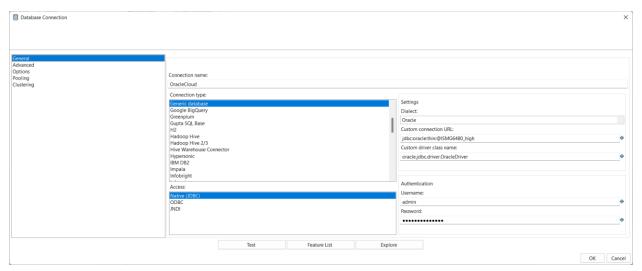


Figure 24: Database Connection Window

o Connection Name: OracleCloud (You can use another name if you want)

Connection Type: Generic Database

Access: Native (JDBC)

Dialect: Oracle

Custom connection URL: jdbc:oracle:thin:@ismg6480\_high (replace ismg6480 with your

database name)

Customer driver class name:

You need to provide user name and password.

Click "Test" to test the connection. Then success test result is shown by Figure 25.

 If you installed the driver file in the lib folder, but you get an error, close PDI and restart it.



Figure 25: Database Connection Test

O Type in "SELECT \* FROM SSTIMEDIM" in the SQL section (Figure 26). You can click the **Preview** button to view the database. Click Ok, to exit the Database Connection dialog box.

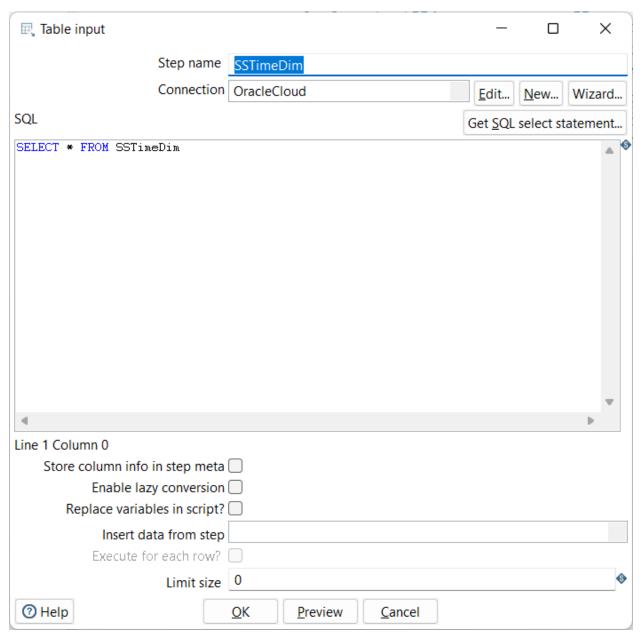


Figure 26: SQL Edit Section in Property Window of Table Input Step

 Add another sort rows component Sort rows 2, and a hop connecting the SSTimeDim step. In the field specification (Figure 27), delete other fields except TIMEDAY, TIMEMOHTH, TIMEYEAR fields.

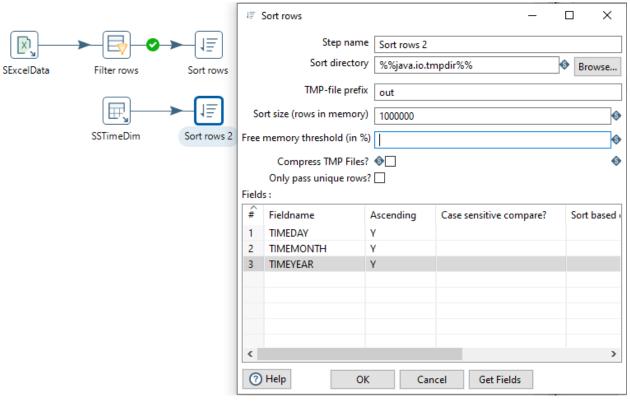


Figure 27: Property Edit Window of Sort Rows 2 Step

- Under the **Design** tab, expand the contents of the **Joins** category.
- 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 28).

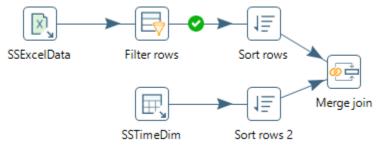


Figure 28: Two Sort Rows Steps Connected to Merge Join Step

O Double-click the Merge Join step to specify its properties (Figure 29). 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.

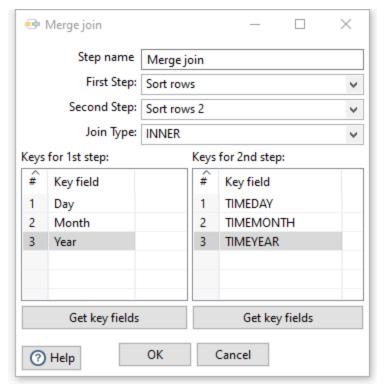


Figure 29: Property Edit Window of Merge Join Step

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

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

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 Oracle schema before these steps.

- o 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.
- o 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 30 shows all steps and hops to the Merge join 2 step.

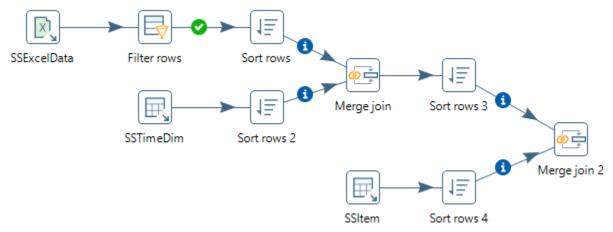


Figure 30: Transformation Design Showing Steps and Hops to the Merge Join 2 Step

#### Step 3 – Inner join the tables.

- o Inner join the tables named *SSCustomer* and *SSStore* in your transformation using the same method described previously.
- o 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.
- o Figure 31 shows all steps and hops after the Merge join 4 step.

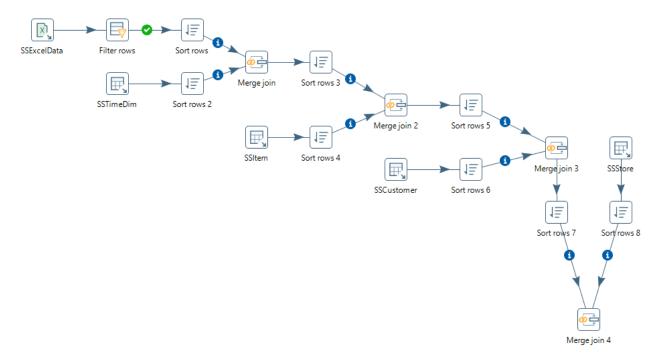


Figure 31: Transformation Design Showing Steps and Hops after the Merge Join 4 Step

#### 5. Insert data into the SSSales table

- o 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 Merge join 4 and Table Output steps. Figure 32 shows the Table Output step (SSSales) connected to the Merge join 4 step.

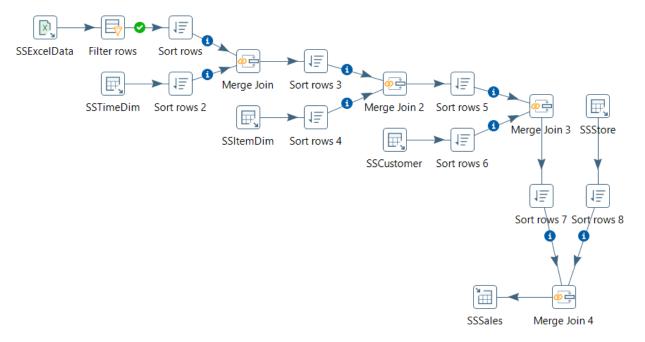


Figure 32: Connect Table Output Step to the Add Sequence Step

- Double click the Table Output step, to specify its properties (Figure 31). Set the step name as SSSales. Select the connection as OracleCloud. 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 33.
- O 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 34. 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.
- o The final view of the **SSSales** step will look like Figure 35.

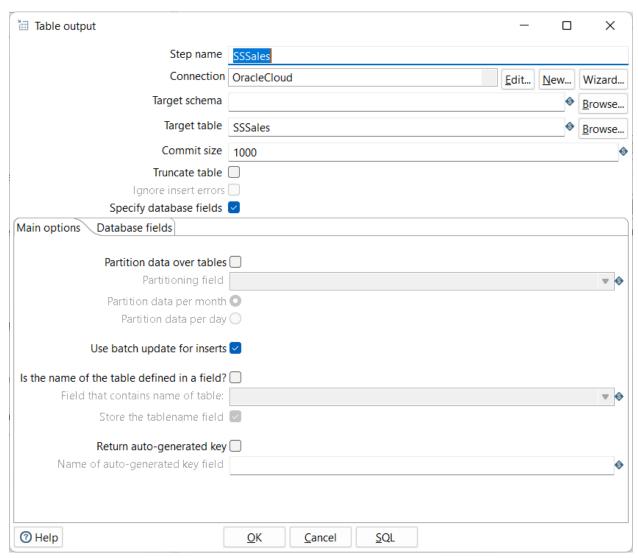


Figure 33: Property Edit Window of Table Output Step

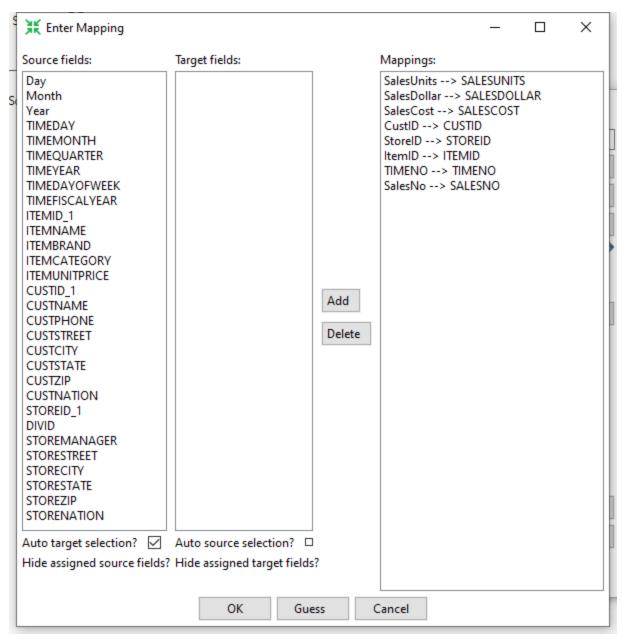


Figure 34: Enter Mapping Window

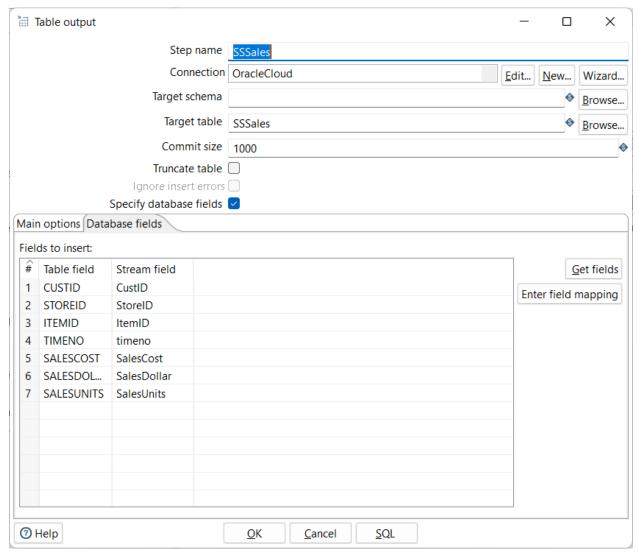


Figure 35: Final View of the SSSales Step

- Select the **SSSales** step and run a preview by clicking on . In the transformation debug dialog, click on **Quick Launch** (Figure 36).
- o Figure 37 shows the Examine preview data window.

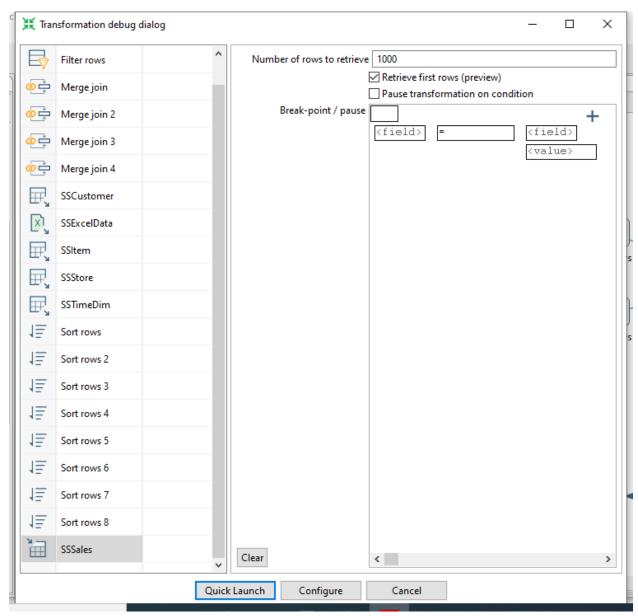


Figure 36: Transformation Debug Dialog

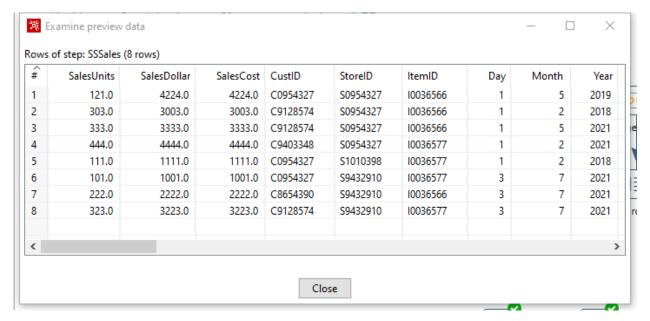


Figure 37: Examine Preview Data Window

 To examine the details of each step, you should examine the Execution Results window below the design pane. The Step Metrics tab (Figure 38) shows details about the execution of each step. You should verify that the SSSales step has 8 output rows.

□ Logging ② Execution History □ Step Metrics ✓ Performance Graph □ Metrics ● Preview data  • Performance Graph □ Metrics ● Preview data													
												# ^	Stepname
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 38: Step Metrics in the Execution Result Window

 Each step in the transformation should have a check mark indicating execution as shown in Figure 39.

- Connect to your Oracle Cloud account 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 Oracle SSSales table.
- If you execute the transformation again, an additional 8 rows will be inserted into the *SSSales* table.
- o 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.

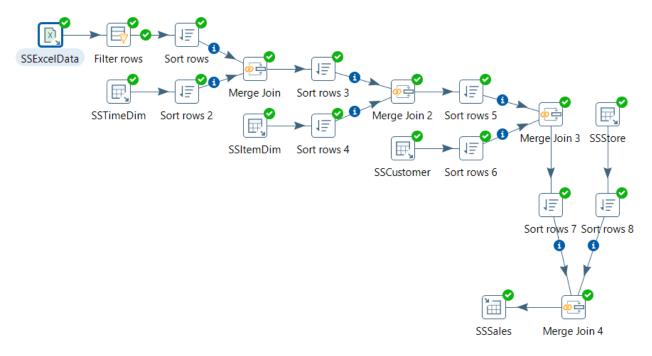


Figure 39: Transformation Design with Check Marks for Each Step

### 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  $File \rightarrow New \rightarrow Transformation$  with name "SSStoreTestAccess". Use  $File \rightarrow Save As ...$  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 40 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 41). In this dialog box, you specify the properties related to this step.

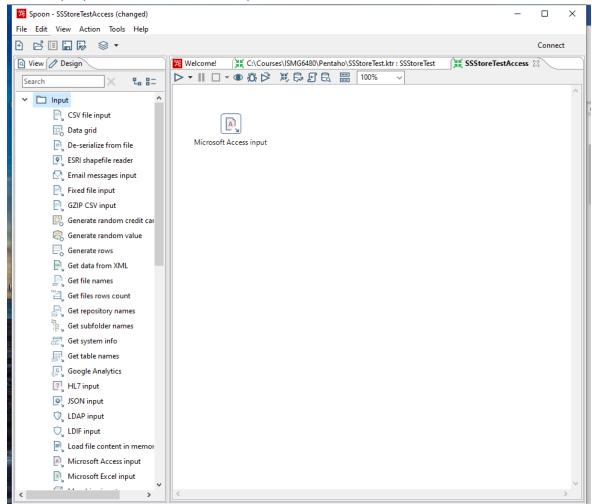


Figure 40: New Microsoft Access Input Step

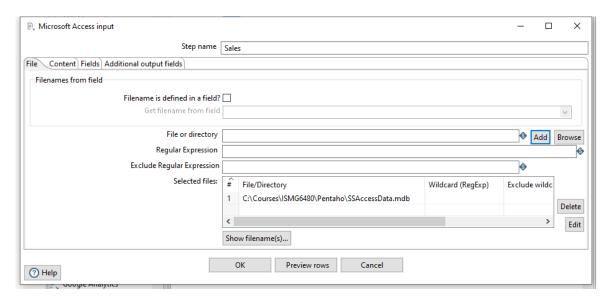


Figure 41: Property Edit Window of Microsoft Access Input Step

- Set name for the Access Input as Sales and specify the Excel data source path in the Files tab.
- o In the tab named **Content**, click the button "**Get tables**" of **table** section. There will appear a window (Figure 42). Select **Sales** as the table name, click **OK**.

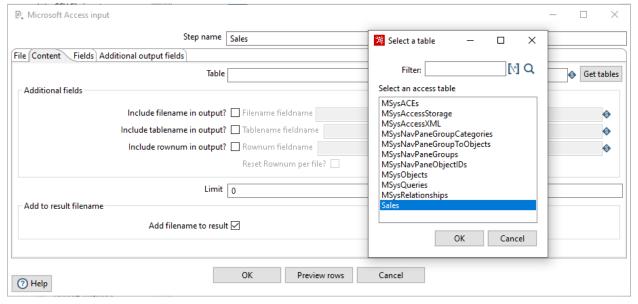


Figure 42: Table Selection Window

o In the tab named **Fields**, click the button "**Get fields**". There will appear a list (Figure 43) showing the fields in the table named **Sales**.

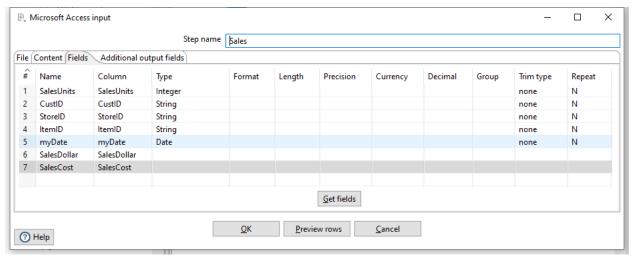


Figure 43: Fields Window for Microsoft Access Input Property Editing

 Click the button "Preview rows" to preview the database (Figure 44). When asked for the number of rows type 12 and click OK.

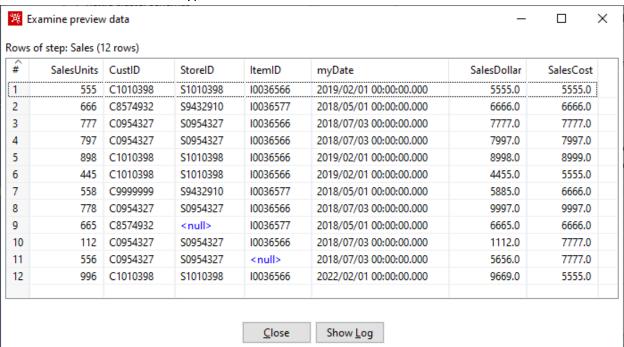


Figure 44: Examine Preview Data Window

 Click **OK** at the bottom of the window. The input icon will change to the shape shown by Figure 45.



Figure 45: Sales Step Icon

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

- $\circ$  Add a Filter Rows step to your transformation. Under the **Design** table, go to **Flow**  $\rightarrow$  **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 46 shows the transformation window after adding the new step and hop.

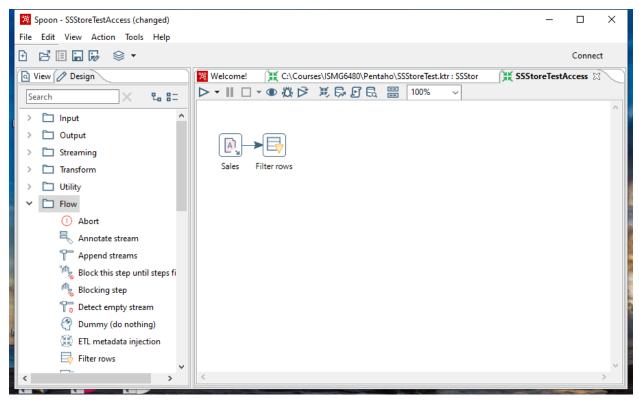


Figure 46: Access Input Step and Filter Step in Spoon

- 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.
- o 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 47. Save the transformation before adding new steps.

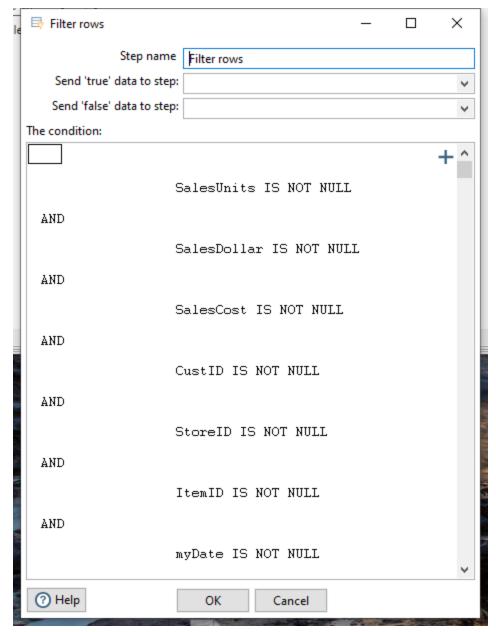


Figure 47: Filter Conditions Window

# 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 48).
   Select Result is TRUE in the filter results selection list.



Figure 48: True Filter Results Connected to Select Values Step

- o Double-click the Select values step to open its edit properties dialog box.
- o In the tab named Meta-data, click the button "**Get fields to change**", to get the fields to change, which is shown by Figure 49. Change the **Type** of field **myDate** as **String**, change its **Format** to dd-MM-yyyy. Click **OK**.

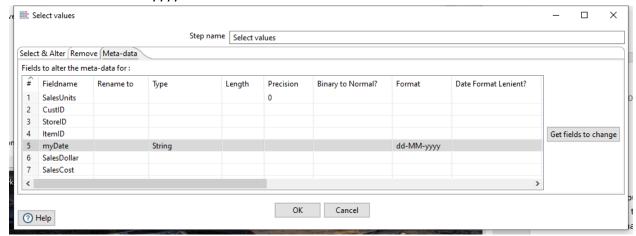


Figure 49: Meta-data Tab of Select Values Property Edit Window

- Under the **Design** tab, expand the contents of the **Transform** step.
- o Click and drag a **Split fields** step into your transformation (Figure 50).
- 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 50: Create Split Fields with Hop between Steps

- o Double-click the **Split fields** step to open its edit properties dialog box (Figure 51).
- 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 Integer. Click Ok when finished.

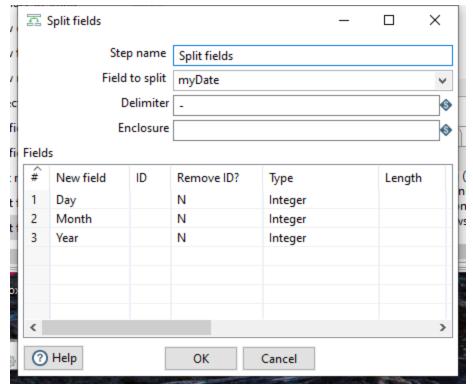


Figure 51: Property Edit Window of Field Splitter Step

Select the Split fields step in the canvas and click , to preview this transform (Figure 52). Make sure that Split Fields step is selected from the left side panel of the transformation debug dialog and click on "Quick Launch" button.

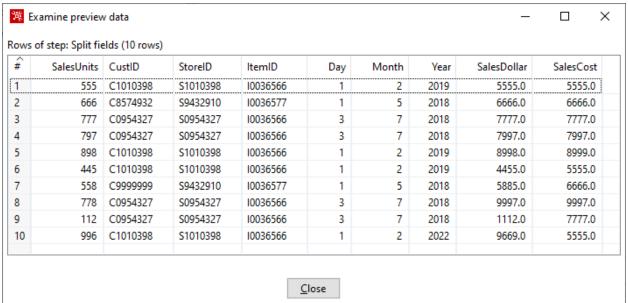


Figure 52: Examine Preview Data Window

### 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 resembles the details in Section 4.

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.
- o Double-click the Table Input step to open its edit properties dialog box.
- o Rename your Table Input step to SSTimeDim.
- For the Connection field, select OracleCloud 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 24.
- Connection Name: OracleCloud (You can use another name if you want)

Connection Type: Generic Database

Access: Native (JDBC)

Dialect: Oracle

Custom connection URL: jdbc:oracle:thin:@ismg6480\_high (replace ismg6480 with your database name)

Customer driver class name:

You need to provide user name and password.

- Click "Test", to test the connection.
- O 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.
- o 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.
- o Now, we have finished inner join between the Access table and *SSTimeDim* table.
- Figure 53 shows the transformation design with all steps and hops to the Merge join step.

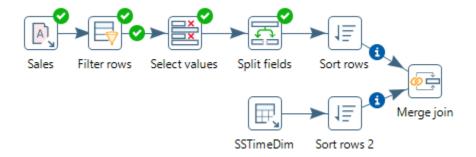


Figure 53: Transformation Design with Steps and Hops to the Merge Join Step

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.
- o For SSItem step, connect ItemID (from Excel file) and ITEMID (from Database) fields.
- o For SSCustomer step, connect CustID (from Excel file) and CUSTID (from Database) fields.
- o For SSStore step, connect StoreID (from Excel file) and STOREID (from Database) fields.
- o Figure 54 shows the transformation design for steps and hops to the Merge join 4 step.

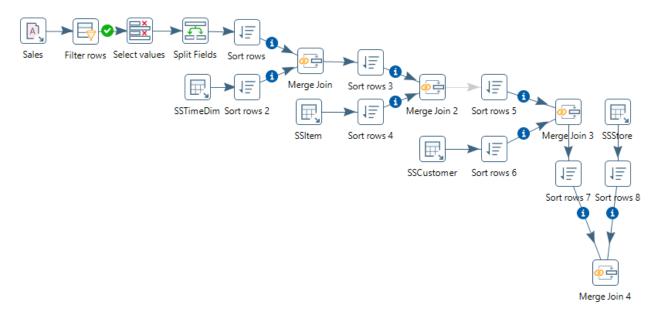


Figure 54: Transformation Design with Steps and Hops to the Merge Join 4 Step

#### 9. Insert rows into the SSSales table

- o 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 Merge Join 4 and Table Output steps. Figure 55 shows the connection in the transformation design pane.
- Double click the Table Output step, to specify its properties. Set the step name as SSSales. Select the connection as OracleCloud. Type in the Target table as SSSales.
- 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 56. Then click OK.

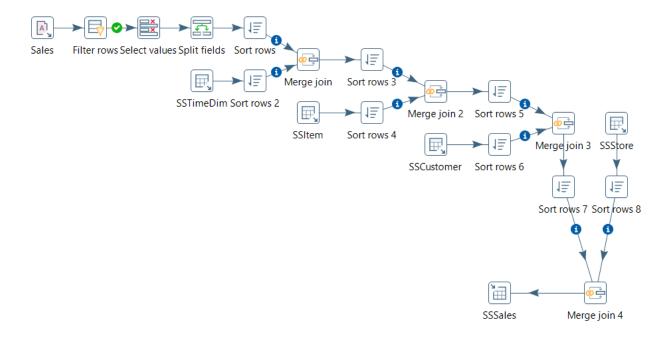


Figure 55: Connect Table Output Step to Merge Join 4

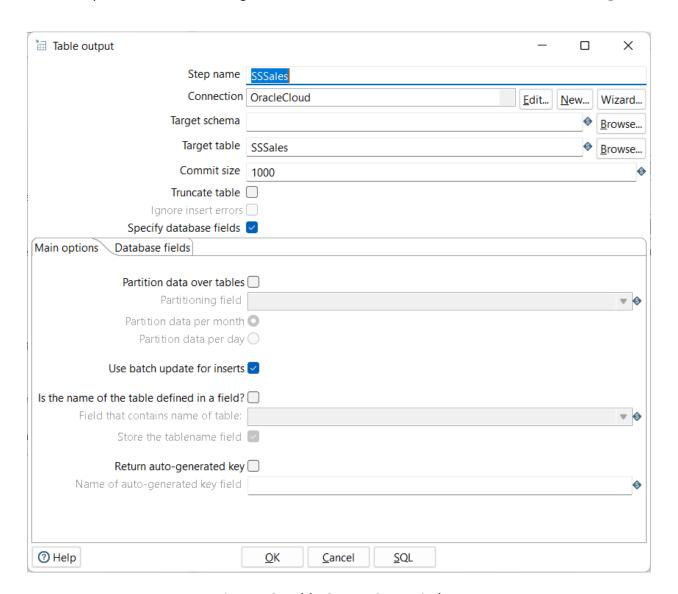


Figure 56: Table Output Step Window

- Select the SSSales step and run a preview by clicking on . In the transformation debug dialog click on Quick Launch button. Figure 57 shows the result rows added to the SSSales table after execution of the transformation. The Step Metrics tab (Figure 58) shows that 8 rows were inserted (Output column in Figure 59) into the SSSales table.
- Connect to your Oracle Cloud account 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).

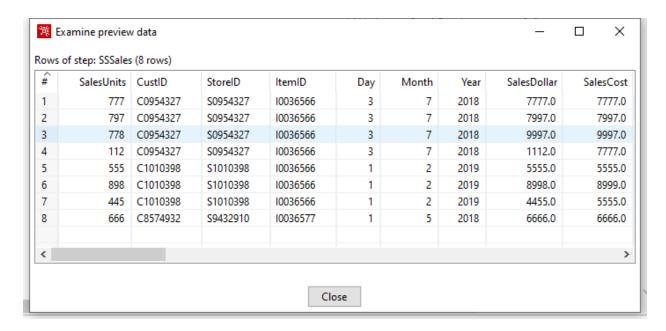


Figure 57: Preview Data for the SSSales Step in the Access Transformation

□ Logging ② Execution History ☐ Step Metrics ✓ Performance Graph ► Metrics ● Preview data  • Preview data													
												# ^	Stepname
1	Sales	0	0	12	12	0	0	0	0	Finished	0.1s	102	-
2	SSTimeDim	0	0	16	16	0	0	0	0	Finished	0.5s	30	-
3	SSCustomer	0	0	11	11	0	0	0	0	Finished	0.6s	18	-
4	Filter rows	0	12	10	0	0	0	0	0	Finished	0.1s	98	-
5	Select values	0	10	10	0	0	0	0	0	Finished	0.2s	67	-
6	SSItem	0	0	10	10	0	0	0	0	Finished	0.5s	18	-
7	Sort rows 2	0	16	16	0	0	0	0	0	Finished	0.3s	56	-
8	Split fields	0	10	10	0	0	0	0	0	Finished	0.2s	53	-
9	Sort rows	0	10	10	0	0	0	0	0	Finished	0.2s	47	-
10	SSStore	0	0	6	6	0	0	0	0	Finished	0.5s	13	-
11	Sort rows 6	0	11	11	0	0	0	0	0	Finished	0.4s	27	-
12	Sort rows 4	0	10	10	0	0	0	0	0	Finished	0.3s	29	-
13	Merge join	0	26	9	0	0	0	0	0	Finished	0.6s	40	-
14	Sort rows 3	0	9	9	0	0	0	0	0	Finished	0.7s	13	-
15	Merge join 2	0	19	9	0	0	0	0	0	Finished	1.1s	18	-
16	Sort rows 5	0	9	9	0	0	0	0	0	Finished	1.1s	8	-
17	Merge join 3	0	20	8	0	0	0	0	0	Finished	1.5s	14	-
18	Sort rows 7	0	8	8	0	0	0	0	0	Finished	1.5s	5	-
19	Sort rows 8	0	6	6	0	0	0	0	0	Finished	0.2s	31	-
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	2.4s	3	_

Figure 58: Step Metrics in the Execution Result Window for the Access Transformation