Descriptive Analysis and Visualisation
CA2 Lab Report

RICHARD OGUJAWA

February 3, 2024

Description

This report details one of the ways in which Talend OS can be used (alongside MySQL Workbench) to implement a slowly changing dimension (SCD). The report highlights the process of creating a staging and dimensional table in MYSQL. Data from Excel files were subsequently extracted and cleaned before being loaded into the staging table using Talend. Then the data from the staging table was extracted and added to the dimensional table using the Type 2 SCD method in Talend.

Aims and Objectives

- Set up the Staging and Dimensional Table in MySQL
- Create Talend Project
- Create MetaData Files
- Create Talend Jobs that Clean and Load the Data
- Run Talend Jobs to Insert and Update data in Database

Methods

Set up Staging and Dimensional Table in MySQL

After opening MYSQL Workbench, an SQL script was written to generate a staging table which was purposed to act as a a reservoir for the source data prior to it being integrated into the Dimensional Table. The script written in order to achieve this, did the following:

1. **XYZ_User:** Created a schema/user called `xyz_user`, named after the business that the tables are being created for. Prior to this, the xyz_user is dropped in case it already exists in the database.

```
-- Drop the schema if it already exists
DROP SCHEMA xyz_user;

-- Create the xyz_user schema
CREATE SCHEMA xyz_user;
```

Figure 1.1: SQL Command Used to Create `xyz_user` Schema

2. **Staging Table:** Created a customer staging table called `stg_customer_detail` with attributes that reflected the columns that were present in the original data:

CUSTOMER_ID	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_ADDRESS	PINCODE	DATE	CUSTOMER_DOB
12345	FIRNAME	A!!	#123,1ST MAIN, 2ND CROSS#	520083	01/01/2010	04/04/1977
678910	FNAME	B!!	#342,2ND MAIN, 15TH CROSS#	520084	02/01/2011	01/01/1971
11121314	FIRSAME	C!!	#32, 1ST MAIN, 1ST CROSS#	520085	03/01/2010	01/01/1979
15161718	FAME	D!!	#151, 15TH MAIN, 39TH CROSS#	520086	04/01/2013	01/01/1978
19202122	FIRSTAME	E!!	#155, 2ND CROSS,11TH AVENUE#	520087	04/05/2012	01/01/1984

Figure 2.1: The Original Source Data files

Alongside these fields an additional attribute (`customer_sk`) was created to store the values of the surrogate keys that Talend would generate to ensure the uniqueness of each row in the staging table.

```
-- Create the tables
CREATE TABLE IF NOT EXISTS stg_customer_detail(
  customer_sk INT NOT NULL AUTO_INCREMENT,
  customer_bk INT NOT NULL,
  customer_first_name VARCHAR(45) NULL,
  customer_last_name VARCHAR(45) NULL,
  customer_address VARCHAR(45) NULL,
  pincode INT NULL,
  date DATETIME NULL,
  customer_dob DATETIME NULL,
  PRIMARY KEY (customer_sk));
```

Figure 2.2: SQL Command to Create the Staging Table

3. **Dimensional Table:** Created a dimensional table called `customer_detail_dim` which would store the data for the SCD. The attributes mirror the ones found in the corresponding staging table, however, additional columns were added to take into consideration the fact that it would be storing slowly changing dimensions:

   - `scd_start` which stores the date that the data was added to the source file, this column is used in place of the 'date' column in `stg_customer_detail`.
   - `scd_end` is used to store the end date, i.e. the date when at least one value in that row, was no longer valid from.
   - `scd_active` is used to store a 1 or 0 depending on whether the record is the most up-to-date record.
   - `scd_version` is used to track which version the record is.

```
CREATE TABLE IF NOT EXISTS customer_detail_dim(
  customer_dim_sk INT NOT NULL AUTO_INCREMENT,
  customer_bk INT NOT NULL,
  customer_first_name VARCHAR(45) NULL,
  customer_last_name VARCHAR(45) NULL,
  customer_address VARCHAR(45) NULL,
  pincode INT NULL,
  customer_dob DATETIME NULL,
  scd_start DATETIME NULL, -- This refers to the date that
  -- the all the information in this row started being
  -- considered as valid/the most up to date data
  scd_end DATETIME NULL, -- This refers to the date
  -- when the information in this row stopped being valid,
  -- if not entirely then at least partially
  scd_active TINYINT NULL, -- 0 or 1 depending on whether
  -- or not this is the most up to date information
  -- about the customer
  scd_version INT NULL, -- this increments each time the
  -- given data point is updated
  PRIMARY KEY (customer_dim_SK));
```

Figure 2.3: SQL Command to Create `customer_detail_dim`

At this point the tables looked like this:

```
52 •  SELECT * FROM xyz_user.stg_customer_detail; -- !!Note!!: Because we set 'Action on Table' in
53    -- Talend equal to Truncate, the old data will be dumped before updating the staging table.
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	date	customer_dob
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 2.4: Initial Staging Table `stg_customer_detail`

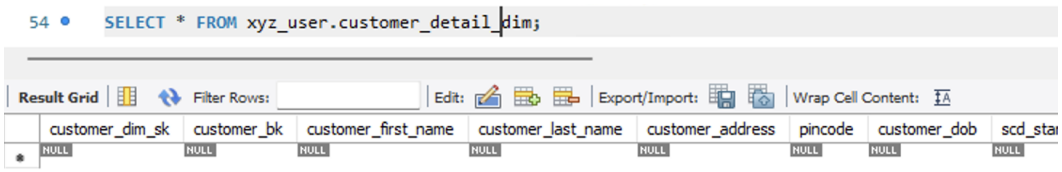


Figure 2.5: Initial Dimensional Table `customer_detail_dim`

Set up the jobs in Talend

- **Create a project:** Created a project in Talend called `Lab_Report`

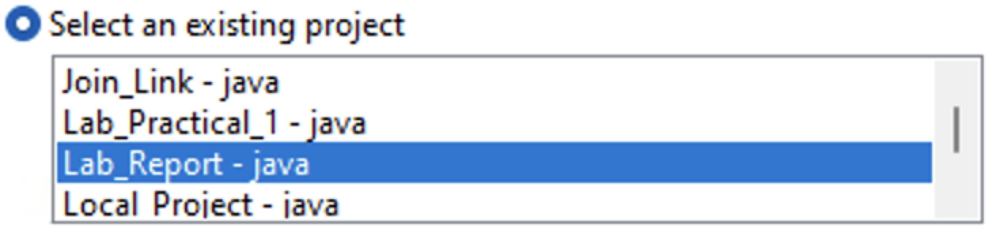


Figure 3.1: Selecting the Newly Created 'Lab _Report' project

- **File Excel:** In the Metadata section of the Repository, three File Excels were created. These three files stored not only the file path to the source data which held the data required for this project, but it also stored the metadata for each file most notably the schema. The reason for doing this is that it speeds up the job creation process: it stores in a repository everything required for a Talend component to use each file, and as such it doesn't require manually entering the file path and setting up the schema each time the file is required.

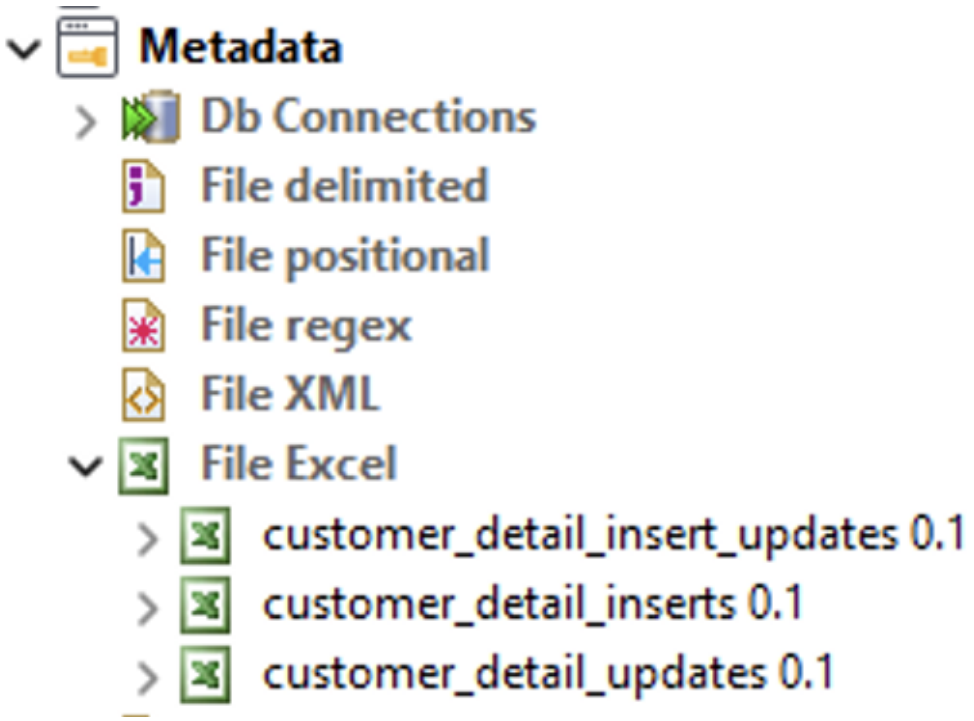


Figure 4.1: Metadata for File Excel Used in this Project

The process for creating each File Excel was the same:

1. 'File Excel' was right-clicked and 'Create File Excel' was chosen.
2. The properties for the file excel were defined such as name (the name of the File Excel), purpose (what the File Excel was being created for) and description (what the File Excel was). After this the `Next` button was clicked on to move on to the next step.

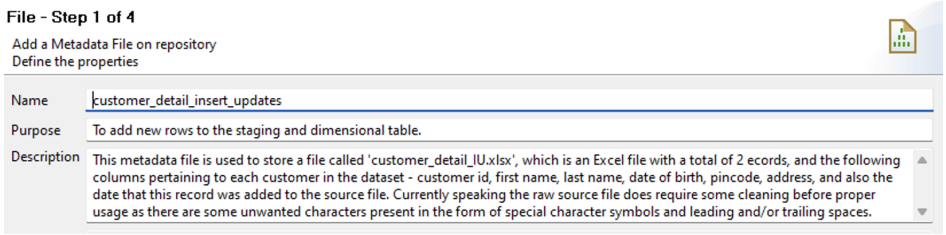


Figure 4.2: File Excel Properties

3. In Step 2, the path of the source file was defined and the desired sheet was also specified to let Talend know where the data was to be pulled from. In this case the sheet in the source file which stored the required data was called

'CUSTOMER'.

4. After these fields were populated the `Next` button was pressed to move on to the next step.

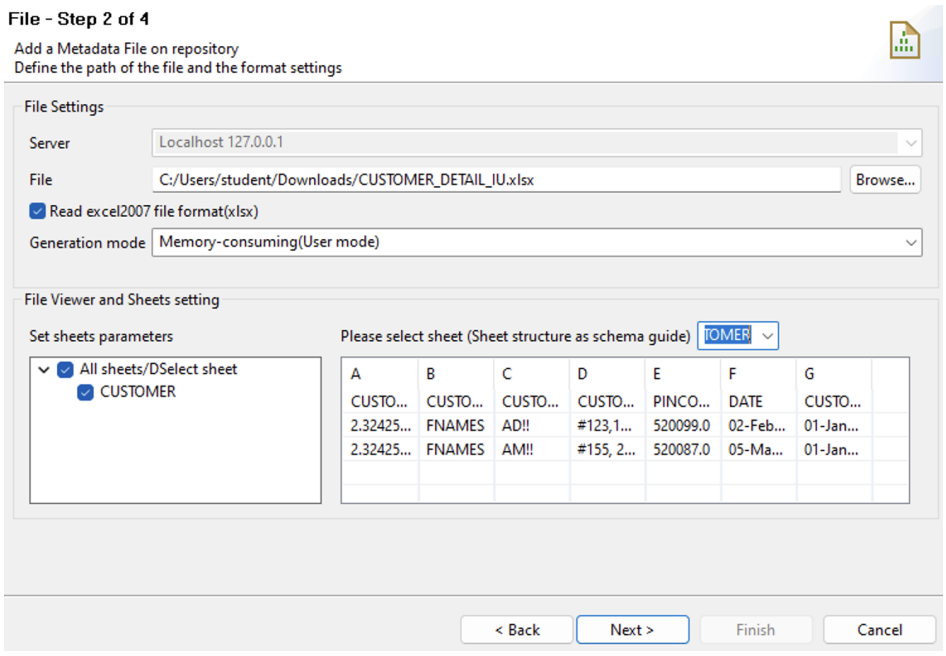


Figure 4.3: Setting File Path and Format Settings

5. The third step required defining the setting of the parse job. This is where the correct delimiter needed to be set (if it wasn't default setting didn't suffice). The number of rows to be regarded as heading rows were established, and the 'Set heading row as column names' checkbox was checked to ensure that the row set to be the heading row was used to determine the column names in the parsed dataset. Refresh Preview was then pressed to preview the resulting dataset and ensure that the settings were correct.

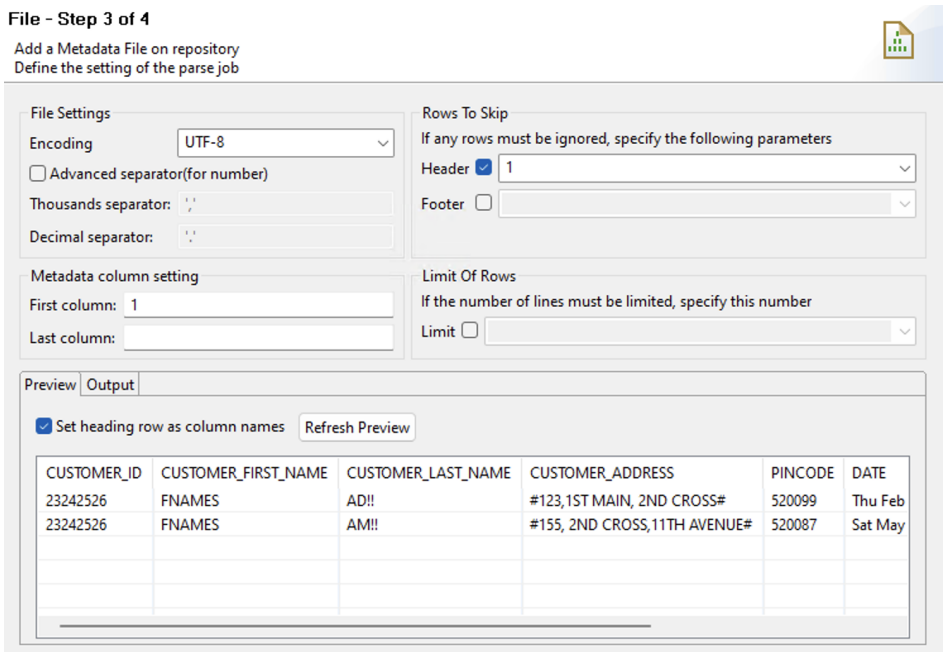


Figure 4.4: Defining the Setting of the Parsing Data

6. Then the `Next` button was pressed to go to the final step.
7. The final step required setting up the schema. Talend pulls in the column names and automatically generates data types for each one, so, although the schema didn't have to be created from scratch it still did need to be tweaked. Customer_id was set as the table's primary key, the lengths of the columns that were of datatype 'String' were adjusted from their original lengths to 45 characters, and the columns that stored dates were set to the appropriate datatype of Date as opposed their original datatype - String.
8. The `Finish` button was pressed to complete the creation of the File Excel.

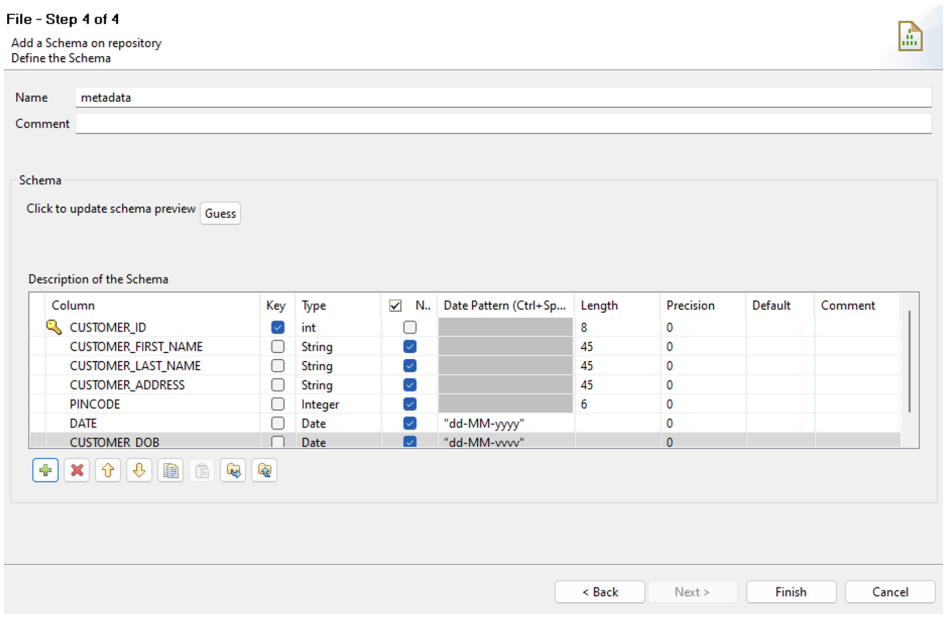


Figure 4.5: Defining File Excel Schema

As aforementioned this process was largely the same for all the excel files in this project, as such the same process was repeated for each one.

- **Db Connections:** The other metadata item which was required for this project was a Db connection. As the name suggests, this item stores a database connection and once again is extremely beneficial when it comes to the speed at which jobs are created due to similar reasons as was expressed for the purpose and benefit of creating File Excels.

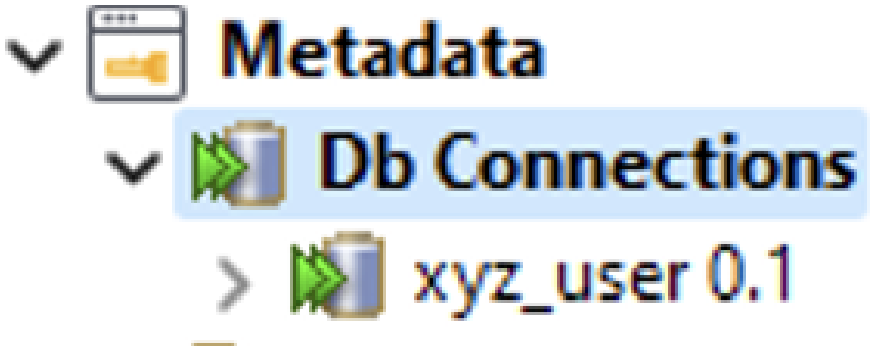


Figure 4.6: Db Connections

The Db Connection 'xyz_user' was created to store a connection to the xyz_user schema which was created in MySQL. The following steps were performed in order

to do this:

1. The first step was quite similar to the creation of a File Excel. After right-clicking on Db Connections and choosing the appropriate item from the menu, the name, purpose and description were set.

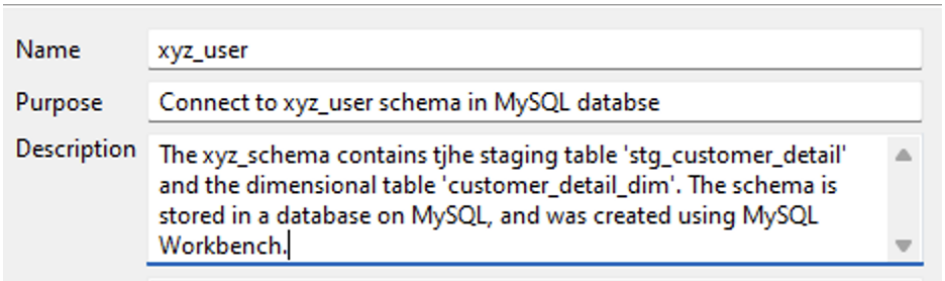


Figure 5.1: DB Connection Properties

2. The second (and final) step required to create the database connection required inputting the following data: 'Db Type' (the type of the database being connected to), 'Login' (the mysql username), 'Password' (the mysql password, which was simply 'password'), 'Server' (the server the database was being run on), and 'Database' (the database that was being connected to, which was the xyz_user schema created prior) given that "in MySQL, a schema is a database." (MySQL, 6)

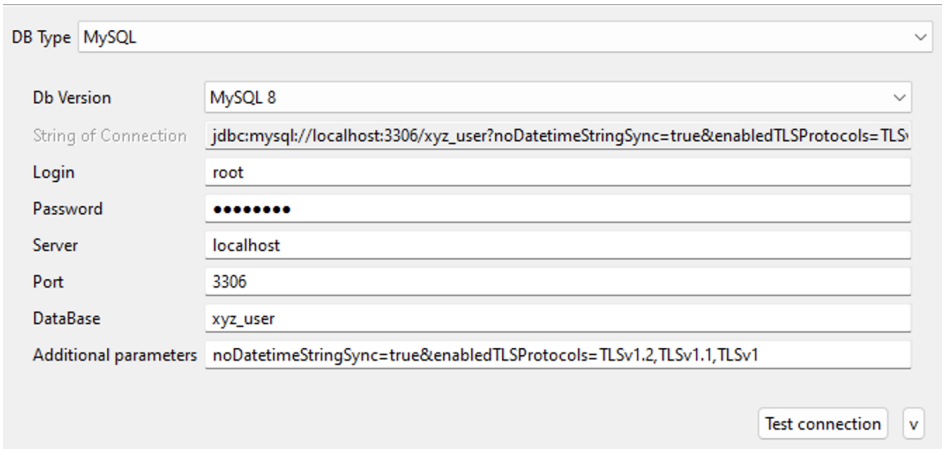


Figure 5.2: Db Connection Settings

3. After these details had been specified, the connection was tested using the `Test connection` button.

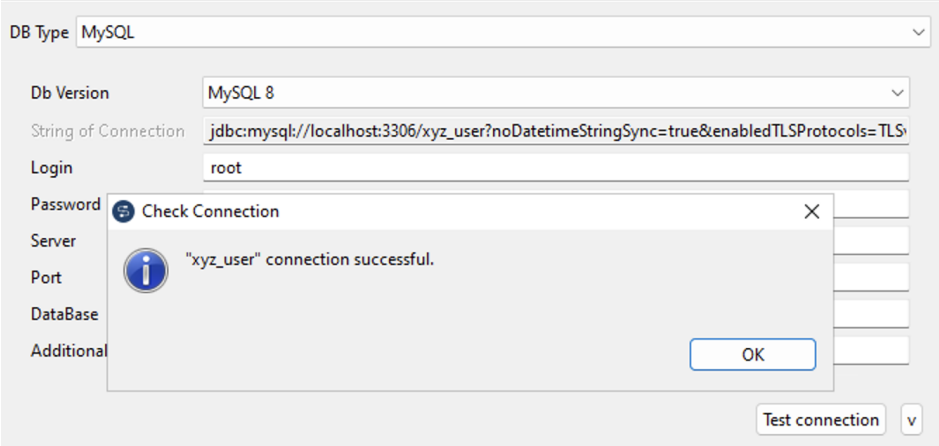


Figure 5.3: Db Connection Success Message

4. Upon receiving the "connection successful" message, the `Finish` button was pressed to create the DB Connection.

- **Setting up Jobs:** The Talend project comprises only three jobs, which were practically identical to each other. This report details one of them, as the other jobs were essentially duplicates of the first job with a minor change which will be mentioned later in this report.

  'Job Design' was right-clicked and 'Create job' was clicked on to create a new job. Properties were defined for the job, such as it's name, description and purpose and the `Finish` button was pressed to create the job.

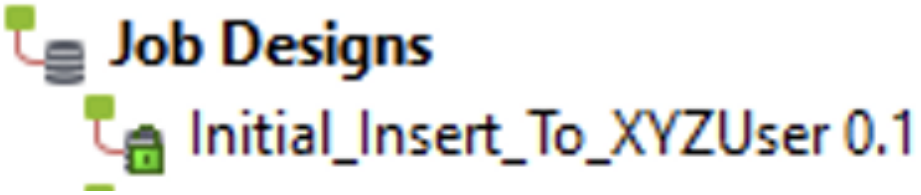


Figure 6.1: First Job

In order to achieve the desired result, the job (in summary) starts in the blue sub-job, where the data is taken from the File Excel, it is cleaned and passed into the staging table. The successful completion of this sub-job triggers the yellow sub-job which pulls data from the staging table and loads it into the dimensional table in the database. Finally, in the green coloured sub-jobs, a query is made to

the database to select all the data in the staging table (in the upper green sub-job) and all the data in the dimensional table (in the lower green sub-job). They are then printed out to the console to ensure that the results are satisfactory.

From a design stand-point, all the default names for components, links and sub-jobs were given more meaningful names to make the job more readable and easily navigable.

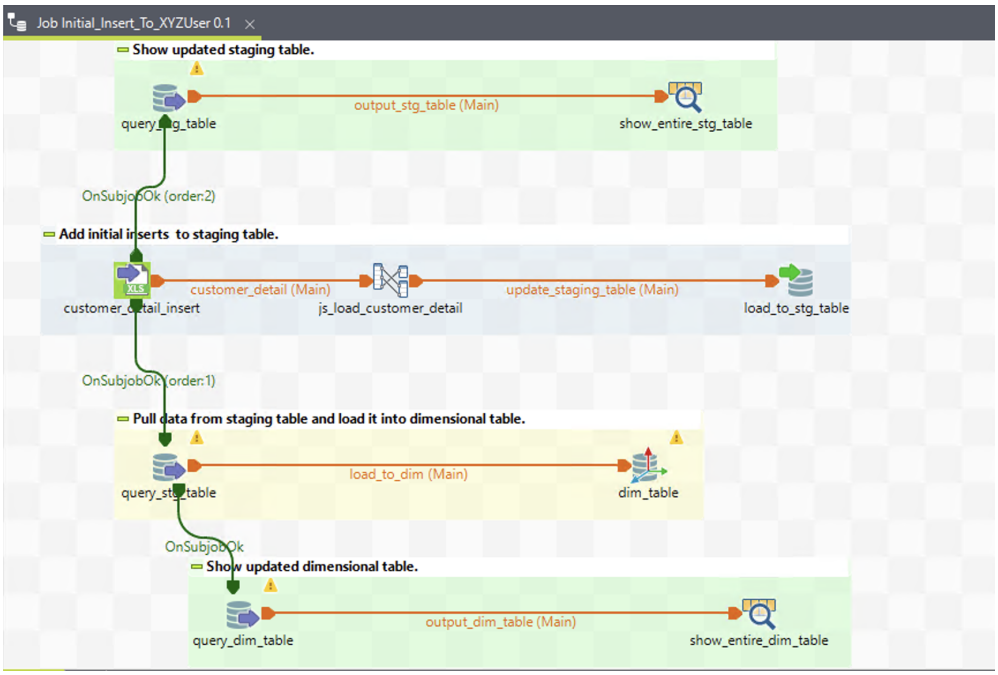


Figure 6.2: Job Design Overview

Blue sub-job: The blue sub-job starts with a **tFileInputExcel** component, which was set up to include the File Excel created earlier. This is the component that differentiates each job because it is the specific File Excel included in this component that determines what data is being inserted into the table. The settings for all the other components in each sub-job are identical across all three jobs.

'Property type' was set to 'Repository' so that the appropriate File Excel could be selected, and the 'schema' was also set to Repository so that the schema for the File Excel could also be chosen.

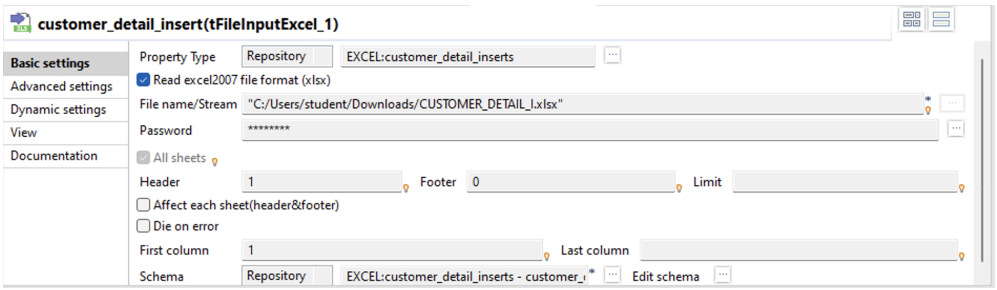


Figure 6.3: tFileInputExcel Set Up

This component was then connected to a **tMap** component to clean the data. Double-clicking on the tMap component opened up the map editor which was then configured to look like this:

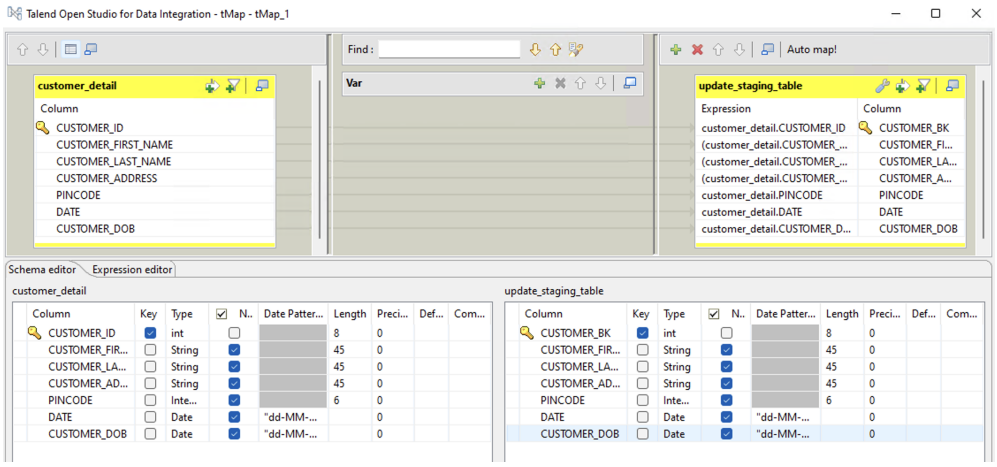


Figure 6.4: tMap Component for Cleaning Data

The customer_detail table is the input from the File Excel, and the update_staging_table is the output from tMap component with the cleaned data. Initially the update_staging_table table wasn't present so it had to be created by pressing the green plus icon on the top right, and then the table was populated with attributes by simply dragging and dropping each field from the column_detail table over to the update_staging_table table. The Expression column refers to what the values stored in each column should be. For example, the CUSTOMER_FIRST_NAME column had unwanted leading and trailing spaces; to deal with this the column had an Expression value of `(customer_detail.CUSTOMER_FIRST_NAME).strip()` which takes all the values in the CUSTOMER_FIRST_NAME column

and returns a value without the leading and trailing spaces using the .strip() method from Java. Another method used was .replaceAll() which replaces all occurrences of a sub-string with another string. This method was used to remove hash symbols from each customer's address and exclamation marks from their last names.

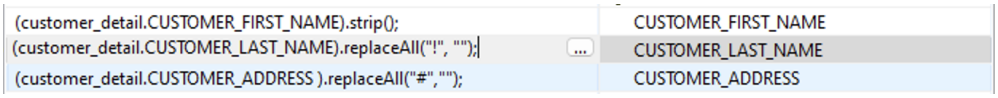


Figure 6.5: Expression Editor

The changes were applied by pressing the `Apply` button and then the `Ok` button. The output of that component was linked to a **tDBOuput** component, which allowed for the cleaned data to be inserted into the staging table. In the tDBOutput component the 'Database' was set to MySQL and then applied. The 'Property type' was set to the DB Connection created earlier. The 'Table' was set to the table that was to be manipulated in MySQL using this component (the 'stg_customer_detail' table). 'Action on table' was set to 'Truncate' which meant that the data was dumped from the table before new data was added. The final step was to set up the schema to match the schema of the table in the database.

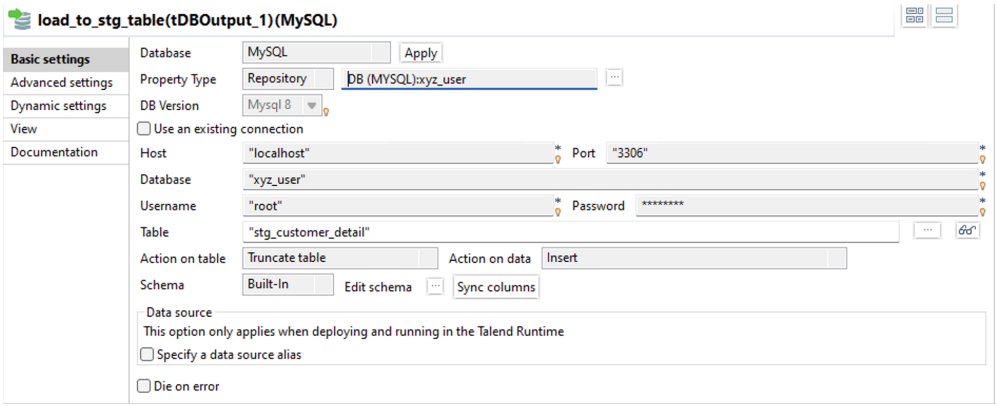


Figure 6.6: tDBOutput Component

On completion the blue sub-job triggers the yellow sub-job, which starts with a **tDBInput** which queries the staging table in the database. The settings for this component are largely similar to the ones for tDBOuput, the main difference

being the 'Query' section where the SQL command was written to pull all the data from the staging table by selecting all of the columns in the table.

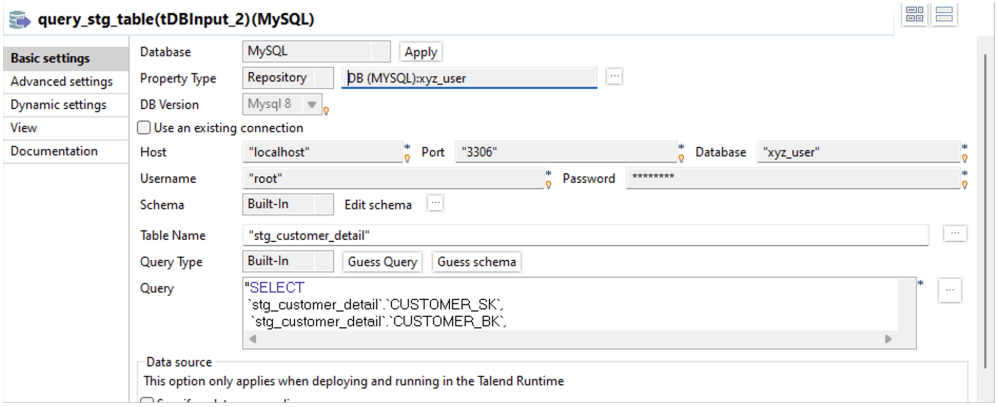


Figure 6.7: tDBInput Component

The tDBInput component was then connected to the **tDBSCD** component which inserted this data into the dimensional table in the database. The settings for this component were largely similar to the tDBOuput component, the only difference being that the 'Table' being targeted for the tDBSCD component was the dimensional table rather than the staging table.

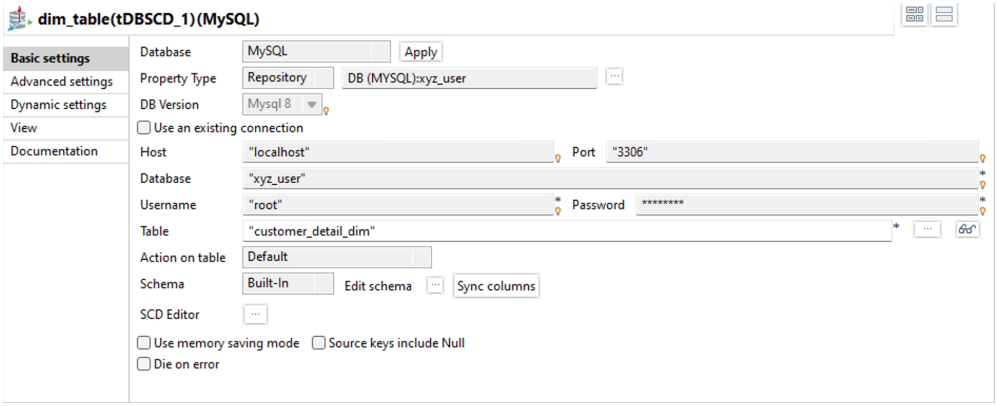


Figure 6.8: tDBCD Component

The SCD component editor was then used to edit the behaviour of the data being passed into the dimensional table. The CUSTOMER_BK was set as the source key. The surrogate key was set to be the CUSTOMER_SK. scd_start was set to be the date that the record was added to the source file. scd_version and scd_active were checked to generate columns which would track the versioning of each record

and the most up-to-date records respectively. Finally the SCD Type implemented for all of the remaining columns was Type 2 which created new records for each record update to ensure the maintenance of data history.

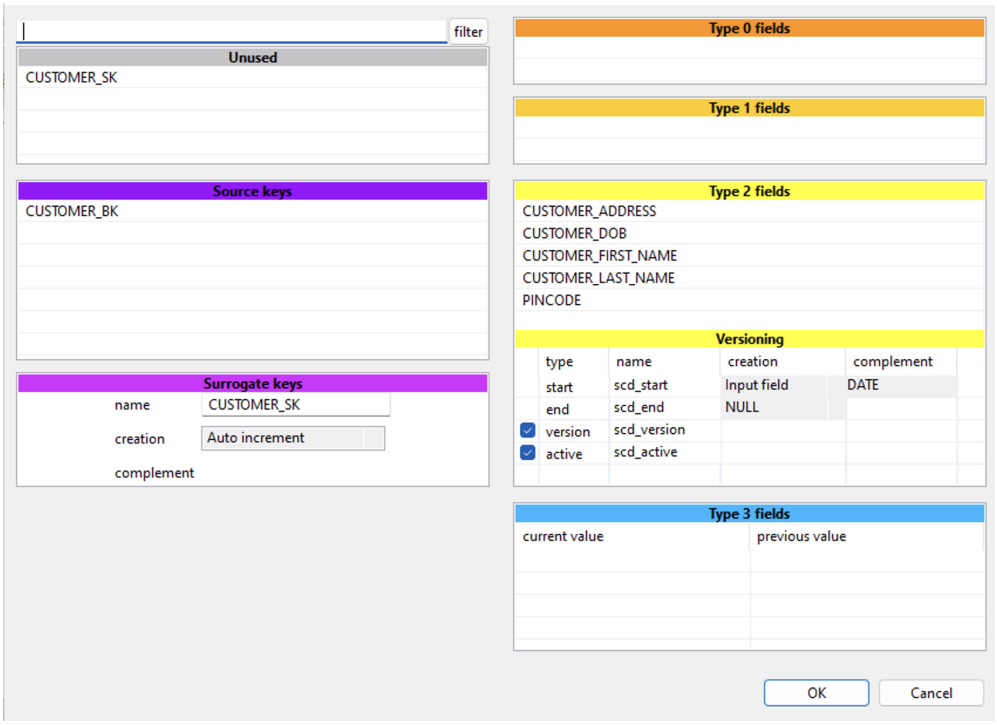


Figure 6.9: SCD Component Editor

On completion, this sub-job triggers the upper green sub-job, which is similar to the yellow sub-job, the only difference being that all the data selected from the dimensional table is logged out into the console rather than being inserted into the dimensional table in the database. To make the output look better the **tLogRow** component had its mode set to 'Table'.

After this sub-job was completed the upper green sub-job is triggered to do a similar task with the key difference being that rather than selecting all the data from the dimensional table, it selects and logs out all the data from the staging table.

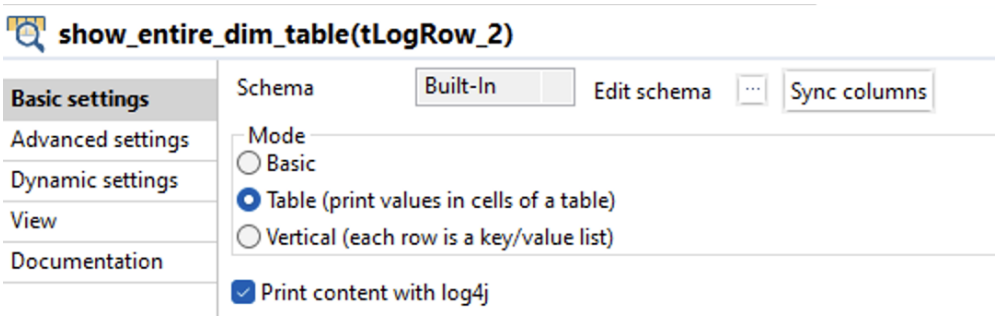
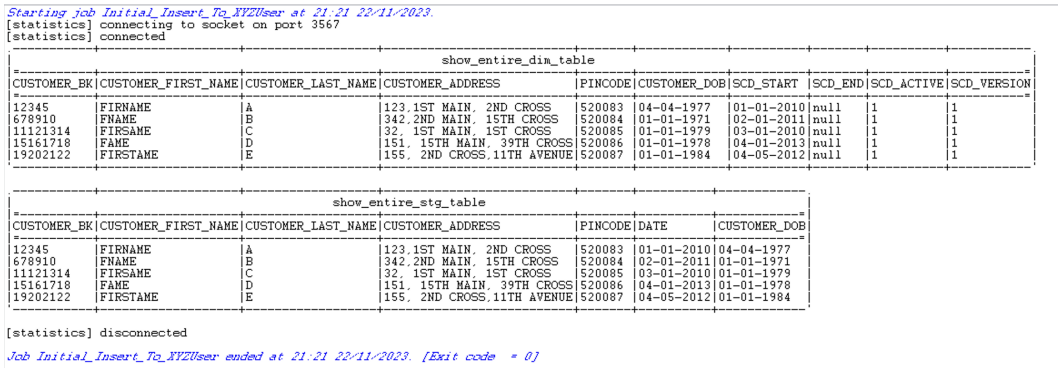


Figure 6.10: tLogRow Component

Results

As was aforementioned, each job was identical to the last except for the File Excel in the tFileInputExcel component. Once all the jobs had been created and configured accordingly (the appropriate File Excel was included in each job), the jobs were run, and the following outputs were achieved:

1. The first job's tFileInputExcel component had the File Excel with the source file set to CUSTOMER_DETAIL_I.xlsx. After running this job, the data was cleaned, and passed into the staging table. The data from that table was extracted and inserted into the dimensional table, and then the data from both tables were logged out to ensure that the appropriate updates had been made to the tables.



```
Starting job Initial_Insert_To_XYZUser at 21:21 22/11/2023.
[statistics] connecting to socket on port 3567
[statistics] connected
+-----------+-----------------+----------------+-------------------+-------+-------------+----------+----------+-----------+---------------+
|                                                  show_entire_dim_table                                                                  |
+===========+=================+================+===================+=======+=============+==========+==========+===========+===============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS  |PINCODE|CUSTOMER_DOB|SCD_START |SCD_END   |SCD_ACTIVE|SCD_VERSION    |
+-----------+-----------------+----------------+-------------------+-------+-------------+----------+----------+-----------+---------------+
|12345      |FIRNAME          |A               |123,1ST MAIN, 2ND CROSS|520083|04-04-1977  |01-01-2010|null      |1          |1              |
|678910     |FNAME            |B               |342,2ND MAIN, 15TH CROSS|520084|01-01-1971  |02-01-2011|null      |1          |1              |
|11121314   |FIRSAME          |C               |32, 1ST MAIN, 1ST CROSS|520085|01-01-1979  |03-01-2010|null      |1          |1              |
|15161718   |FAME             |D               |151, 15TH MAIN, 39TH CROSS|520086|01-01-1978|04-01-2013|null      |1          |1              |
|19202122   |FIRSTAME         |E               |155, 2ND CROSS,11TH AVENUE|520087|01-01-1984|04-05-2012|null      |1          |1              |
+-----------+-----------------+----------------+-------------------+-------+-------------+----------+----------+-----------+---------------+

+-----------+-----------------+----------------+-------------------+-------+----------+-------------+
|                                          show_entire_stg_table                                      |
+===========+=================+================+===================+=======+==========+=============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS  |PINCODE|DATE      |CUSTOMER_DOB|
+-----------+-----------------+----------------+-------------------+-------+----------+-------------+
|12345      |FIRNAME          |A               |123,1ST MAIN, 2ND CROSS|520083|01-01-2010|04-04-1977  |
|678910     |FNAME            |B               |342,2ND MAIN, 15TH CROSS|520084|02-01-2011|01-01-1971  |
|11121314   |FIRSAME          |C               |32, 1ST MAIN, 1ST CROSS|520085|03-01-2010|01-01-1979  |
|15161718   |FAME             |D               |151, 15TH MAIN, 39TH CROSS|520086|04-01-2013|01-01-1978|
|19202122   |FIRSTAME         |E               |155, 2ND CROSS,11TH AVENUE|520087|04-05-2012|01-01-1984|
+-----------+-----------------+----------------+-------------------+-------+----------+-------------+

[statistics] disconnected
Job Initial_Insert_To_XYZUser ended at 21:21 22/11/2023. [Exit code  = 0]
```

Figure 7.1: Console Output After Running First Job in Talend

customer_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	date	customer_dob
1	12345	FIRNAME	A	123,1ST MAIN, 2ND CROSS	520083	2010-01-01 00:00:00	1977-04-04 00:00:00
2	678910	FNAME	B	342,2ND MAIN, 15TH CROSS	520084	2011-01-02 00:00:00	1971-01-01 00:00:00
3	11121314	FIRSAME	C	32, 1ST MAIN, 1ST CROSS	520085	2010-01-03 00:00:00	1979-01-01 00:00:00
4	15161718	FAME	D	151, 15TH MAIN, 39TH CROSS	520086	2013-01-04 00:00:00	1978-01-01 00:00:00
5	19202122	FIRSTAME	E	155, 2ND CROSS,11TH AVENUE	520087	2012-05-04 00:00:00	1984-01-01 00:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.2: Staging Table After First Job

customer_dim_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	customer_dob	scd_start	scd_end	scd_active	scd_version
131	12345	FIRNAME	A	123,1ST MAIN, 2ND CROSS	520083	1977-04-04 00:00:00	2010-01-01 00:00:00	NULL	1	1
132	678910	FNAME	B	342,2ND MAIN, 15TH CROSS	520084	1971-01-01 00:00:00	2011-01-02 00:00:00	NULL	1	1
133	11121314	FIRSAME	C	32, 1ST MAIN, 1ST CROSS	520085	1979-01-01 00:00:00	2010-01-03 00:00:00	NULL	1	1
134	15161718	FAME	D	151, 15TH MAIN, 39TH CROSS	520086	1978-01-01 00:00:00	2013-01-04 00:00:00	NULL	1	1
135	19202122	FIRSTAME	E	155, 2ND CROSS,11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-04 00:00:00	NULL	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.3: Dimensional Table After First Job

2. The second job followed the same process as the first, however the tFileInputExcel component had the File Excel with the source file set to CUSTOMER_DETAIL_IU.xlsx.

```
Starting job Insert_Updates_To_XYZUser at 21:28 22/11/2023.
[statistics] connecting to socket on port 3947
[statistics] connected
+-----------+-----------------+----------------+------------------------------+---------+-------------+-----------+-----------+------------+-------------+
|                                                    show_entire_dim_table                                                                          |
+===========+=================+================+==============================+=========+=============+===========+===========+============+=============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS              |PINCODE  |CUSTOMER_DOB |SCD_START  |SCD_END    |SCD_ACTIVE  |SCD_VERSION  |
+-----------+-----------------+----------------+------------------------------+---------+-------------+-----------+-----------+------------+-------------+
|12345      |FIRNAME          |A               |123,1ST MAIN, 2ND CROSS       |520083   |04-04-1977   |01-01-2010 |null       |1           |1            |
|678910     |FNAME            |B               |342,2ND MAIN, 15TH CROSS      |520084   |01-01-1971   |02-01-2011 |null       |1           |1            |
|11121314   |FIRSAME          |C               |32, 1ST MAIN, 1ST CROSS       |520085   |01-01-1979   |03-01-2010 |null       |1           |1            |
|15161718   |FAME             |D               |151, 15TH MAIN, 39TH CROSS    |520086   |01-01-1978   |04-01-2013 |null       |1           |1            |
|19202122   |FIRSTAME         |E               |155, 2ND CROSS,11TH AVENUE    |520087   |01-01-1984   |04-05-2012 |null       |1           |1            |
|23242526   |FNAMES           |AD              |123,1ST MAIN, 2ND CROSS       |520099   |01-01-1971   |02-02-2012 |05-05-2012 |0           |1            |
|23242526   |FNAMES           |AM              |155, 2ND CROSS,11TH AVENUE    |520087   |01-01-1971   |05-05-2012 |null       |1           |2            |
+-----------+-----------------+----------------+------------------------------+---------+-------------+-----------+-----------+------------+-------------+

+-----------+-----------------+----------------+------------------------------+---------+-----------+-------------+
|                                              show_entire_stg_table                                                |
+===========+=================+================+==============================+=========+===========+=============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS              |PINCODE  |DATE       |CUSTOMER_DOB |
+-----------+-----------------+----------------+------------------------------+---------+-----------+-------------+
|23242526   |FNAMES           |AD              |123,1ST MAIN, 2ND CROSS       |520099   |02-02-2012 |01-01-1971   |
|23242526   |FNAMES           |AM              |155, 2ND CROSS,11TH AVENUE    |520087   |05-05-2012 |01-01-1971   |
+-----------+-----------------+----------------+------------------------------+---------+-----------+-------------+

[statistics] disconnected
Job Insert_Updates_To_XYZUser ended at 21:28 22/11/2023. [Exit code  = 0]
```

Figure 7.4: Console Output After Running Second Job in Talend

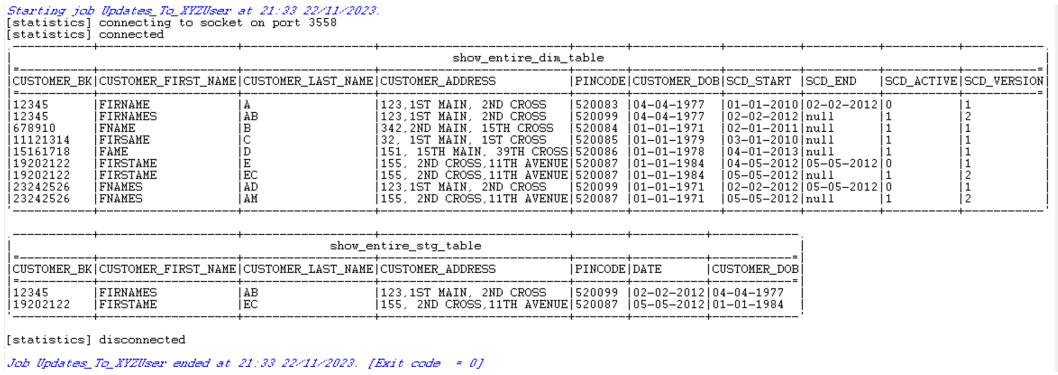
customer_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	date	customer_dob
1	23242526	FNAMES	AD	123,1ST MAIN, 2ND CROSS	520099	2012-02-02 00:00:00	1971-01-01 00:00:00
2	23242526	ES	AM	155, 2ND CROSS,11TH AVENUE	520087	2012-05-05 00:00:00	1971-01-01 00:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.5: Staging Table After Second Job

customer_dim_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	customer_dob	scd_start	scd_end	scd_active	scd_version
131	12345	FIRNAME	A	123,1ST MAIN, 2ND CROSS	520083	1977-04-04 00:00:00	2010-01-01 00:00:00	NULL	1	1
132	678910	FNAME	B	342,2ND MAIN, 15TH CROSS	520084	1971-01-01 00:00:00	2011-01-02 00:00:00	NULL	1	1
133	11121314	FIRSAME	C	32, 1ST MAIN, 1ST CROSS	520085	1979-01-01 00:00:00	2010-01-03 00:00:00	NULL	1	1
134	15161718	FAME	D	151, 15TH MAIN, 39TH CROSS	520086	1978-01-01 00:00:00	2013-01-04 00:00:00	NULL	1	1
135	19202122	FIRSTAME	E	155, 2ND CROSS,11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-04 00:00:00	NULL	1	1
136	23242526	FNAMES	AD	123,1ST MAIN, 2ND CROSS	520099	1971-01-01 00:00:00	2012-02-02 00:00:00	2012-05-05 00:00:00	0	1
137	23242526	FNAMES	AM	155, 2ND CROSS,11TH AVENUE	520087	1971-01-01 00:00:00	2012-05-05 00:00:00	NULL	1	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

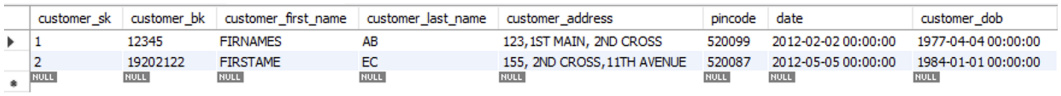
Figure 7.6: Dimensional Table After Second Job

3. The final job also followed the same process as the first two, however the tFileInputExcel component had the File Excel with the source file set to CUSTOMER_DETAIL_U.xlsx.



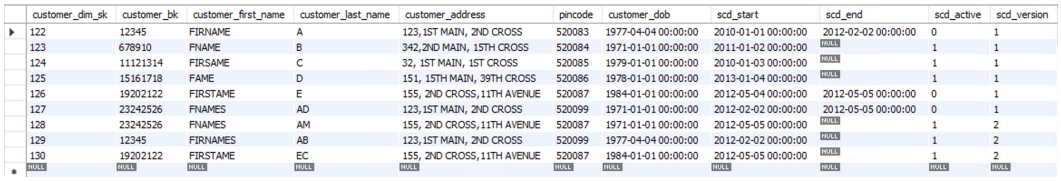
```
Starting job Updates_To_XYZUser at 21:33 22/11/2023.
[statistics] connecting to socket on port 3558
[statistics] connected
+-----------+-----------------+----------------+-----------------------+---------+-------------+------------+------------+-------------+
|                                                show_entire_dim_table                                                                  |
+===========+=================+================+=======================+=========+=============+============+============+=============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS      |PINCODE  |CUSTOMER_DOB |SCD_START   |SCD_END     |SCD_ACTIVE   |SCD_VERSION  |
+-----------+-----------------+----------------+-----------------------+---------+-------------+------------+------------+-------------+
|12345      |FIRNAME          |A               |123,1ST MAIN, 2ND CROSS|520083   |04-04-1977   |01-01-2010  |02-02-2012  |0            |1            |
|12345      |FIRNAMES         |AB              |123,1ST MAIN, 2ND CROSS|520099   |04-04-1977   |02-02-2012  |null        |1            |2            |
|678910     |FNAME            |B               |342,2ND MAIN, 15TH CROSS|520084  |01-01-1971   |02-01-2011  |null        |1            |1            |
|11121314   |FIRSAME          |C               |32, 1ST MAIN, 1ST CROSS|520085   |01-01-1979   |03-01-2010  |null        |1            |1            |
|15161718   |FAME             |D               |151, 15TH MAIN, 39TH CROSS|520086|01-01-1978   |04-01-2013  |null        |1            |1            |
|19202122   |FIRSTAME         |E               |155, 2ND CROSS,11TH AVENUE|520087|01-01-1984   |04-05-2012  |05-05-2012  |0            |1            |
|19202122   |FIRSTAME         |EC              |155, 2ND CROSS,11TH AVENUE|520087|01-01-1984   |05-05-2012  |null        |1            |2            |
|23242526   |FNAMES           |AD              |123,1ST MAIN, 2ND CROSS|520099   |01-01-1971   |02-02-2012  |05-05-2012  |0            |1            |
|23242526   |FNAMES           |AM              |155, 2ND CROSS,11TH AVENUE|520087|01-01-1971   |05-05-2012  |null        |1            |2            |
+-----------+-----------------+----------------+-----------------------+---------+-------------+------------+------------+-------------+

+-----------+-----------------+----------------+-----------------------+---------+----------+-------------+
|                                          show_entire_stg_table                                            |
+===========+=================+================+=======================+=========+==========+=============+
|CUSTOMER_BK|CUSTOMER_FIRST_NAME|CUSTOMER_LAST_NAME|CUSTOMER_ADDRESS      |PINCODE  |DATE      |CUSTOMER_DOB |
+-----------+-----------------+----------------+-----------------------+---------+----------+-------------+
|12345      |FIRNAMES         |AB              |123,1ST MAIN, 2ND CROSS|520099   |02-02-2012|04-04-1977   |
|19202122   |FIRSTAME         |EC              |155, 2ND CROSS,11TH AVENUE|520087|05-05-2012|01-01-1984   |
+-----------+-----------------+----------------+-----------------------+---------+----------+-------------+

[statistics] disconnected
Job Updates_To_XYZUser ended at 21:33 22/11/2023. [Exit code  = 0]
```

Figure 7.7: Console Ouptut After Running Third Job in Talend



customer_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	date	customer_dob
1	12345	FIRNAMES	AB	123,1ST MAIN, 2ND CROSS	520099	2012-02-02 00:00:00	1977-04-04 00:00:00
2	19202122	FIRSTAME	EC	155, 2ND CROSS,11TH AVENUE	520087	2012-05-05 00:00:00	1984-01-01 00:00:00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.8: Staging Table After Third Job



customer_dim_sk	customer_bk	customer_first_name	customer_last_name	customer_address	pincode	customer_dob	scd_start	scd_end	scd_active	scd_version
122	12345	FIRNAME	A	123,1ST MAIN, 2ND CROSS	520083	1977-04-04 00:00:00	2010-01-01 00:00:00	2012-02-02 00:00:00	0	1
123	678910	FNAME	B	342,2ND MAIN, 15TH CROSS	520084	1971-01-01 00:00:00	2011-01-02 00:00:00	NULL	1	1
124	11121314	FIRSAME	C	32, 1ST MAIN, 1ST CROSS	520085	1979-01-01 00:00:00	2010-01-03 00:00:00	NULL	1	1
125	15161718	FAME	D	151, 15TH MAIN, 39TH CROSS	520086	1978-01-01 00:00:00	2013-01-04 00:00:00	NULL	1	1
126	19202122	FIRSTAME	E	155, 2ND CROSS,11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-04 00:00:00	2012-05-05 00:00:00	0	1
127	23242526	FNAMES	AD	123,1ST MAIN, 2ND CROSS	520099	1971-01-01 00:00:00	2012-02-02 00:00:00	2012-05-05 00:00:00	0	1
128	23242526	FNAMES	AM	155, 2ND CROSS,11TH AVENUE	520087	1971-01-01 00:00:00	2012-05-05 00:00:00	NULL	1	2
129	12345	FIRNAMES	AB	123,1ST MAIN, 2ND CROSS	520099	1977-04-04 00:00:00	2012-02-02 00:00:00	NULL	1	2
130	19202122	FIRSTAME	EC	155, 2ND CROSS,11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-05 00:00:00	NULL	1	2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 7.9: Dimensional Table After Third Job

Conclusion

This report details one of the many ways in which an SCD can be implemented using Talend and MySQL. The staging table and dimensional table were set up in MySQL Workbench. Then in Talend a new project was created, metadata files were set up to store the file path to the source files along with their respective schemata, and a database connection metadata file was also created. Once again, the purpose of these files were to speed up the development process. Finally Talend jobs were created to clean and subsequently load the data into both tables in the database. Finally these jobs were run and logged out to the console at each step to ensure that the desired outcome was reached.

References

1. MySQL. 5.3 The mysql System Schema [online] Available at: https://dev.mysql.com/doc/refman/8.0/en/system-schema.html#: :text=schemata%20%3A%20Information%20about%20schemata.,table%20provides%20information%20about%20databases. [Accessed 24 Oct. 2023].