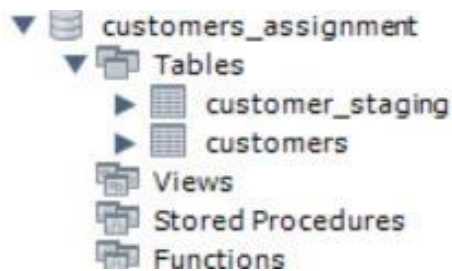


ETL (Talend Assignment)

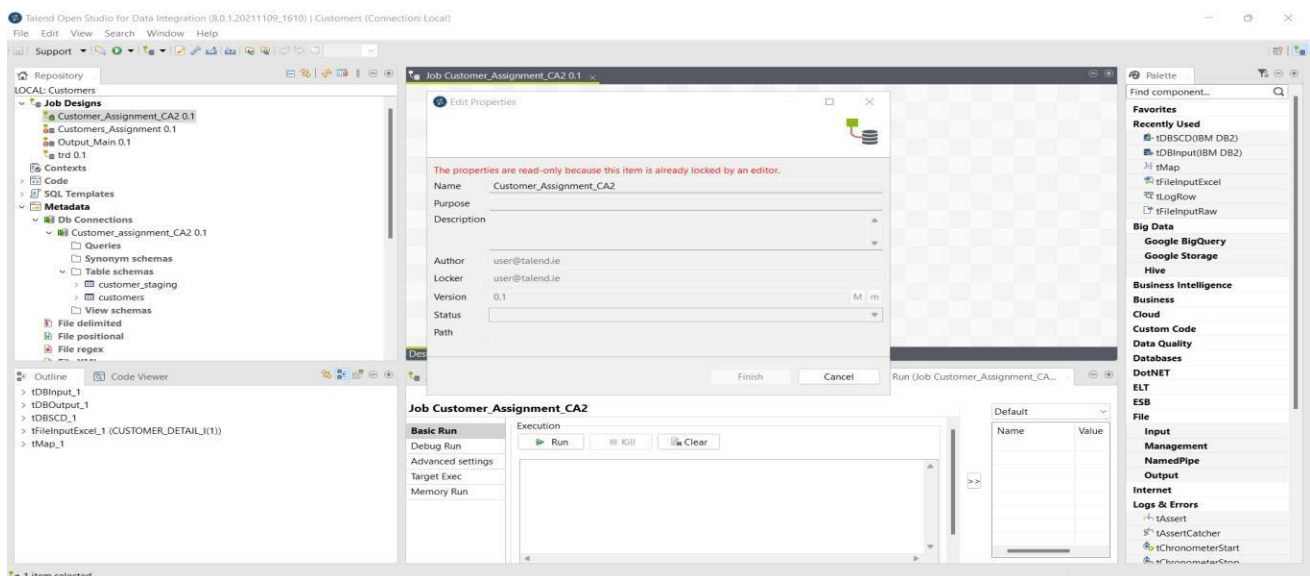
Description: The main goal of using Talend is to cleanse the data, i.e., special characters and spaces, from the raw Excel file and to transform the data and row values so that they can be understood properly. Using SCD (slowly changing dimension) so that it maintains all the historical and new records in the database.

Aim: There are 3 different Excel files that need to be cleaned, transformed, and loaded into a staging database named "customer staging", and then all the 3 staging outputs would be merged into one output to a dimensional database named "customers" using a slowly changing dimension (SCD).

I have created a MySQL schema called "customers_assignment" that contains two tables called "customers" and "customer_staging."



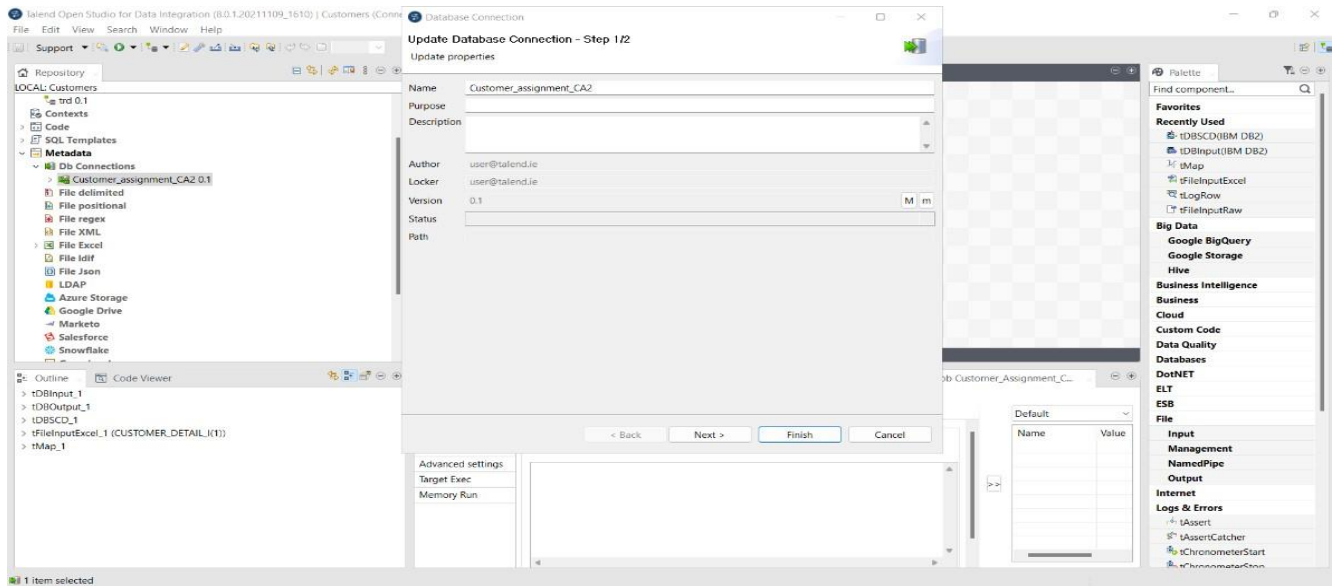
Step – 1: Creating a Job Design



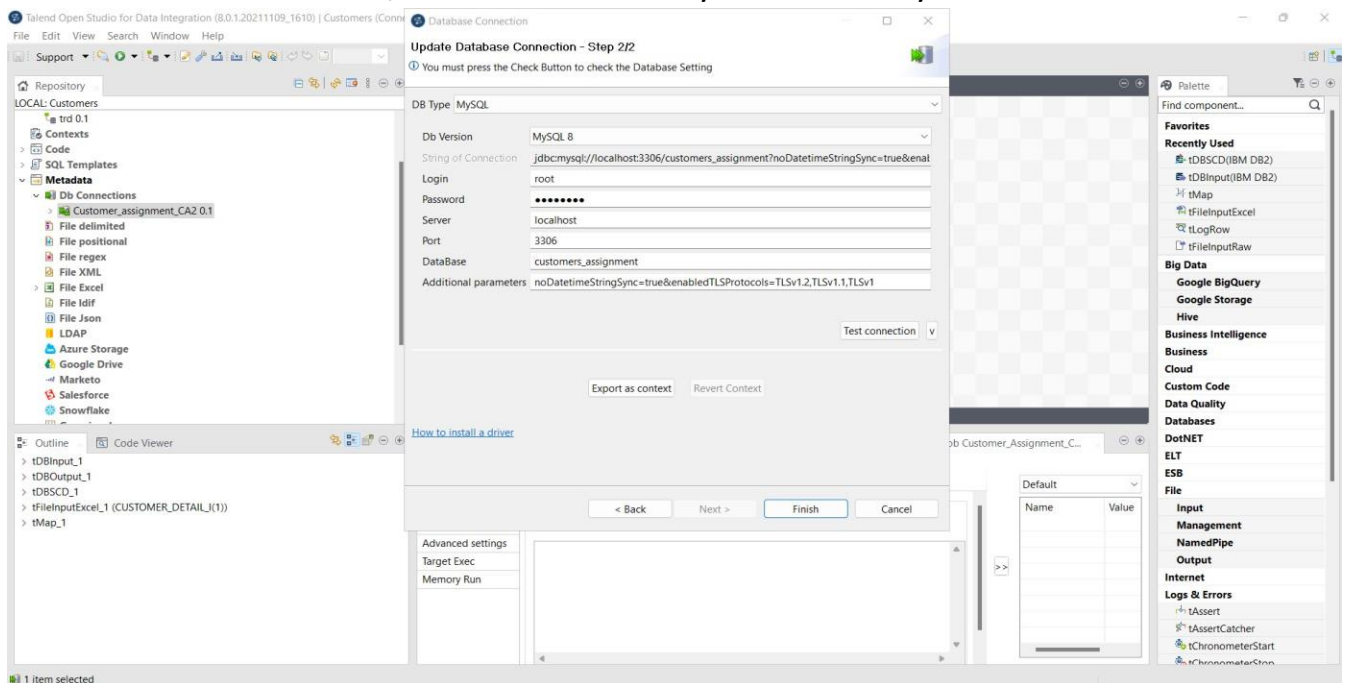
I have created a job design called "Customer_Assignment_CA2".

Step – 2: Creating and Adding Database connection in Metadata Repository.

In repository, selected Metadata → Db connections and then created a new database connection called as “Customer_assignment_CA2”.

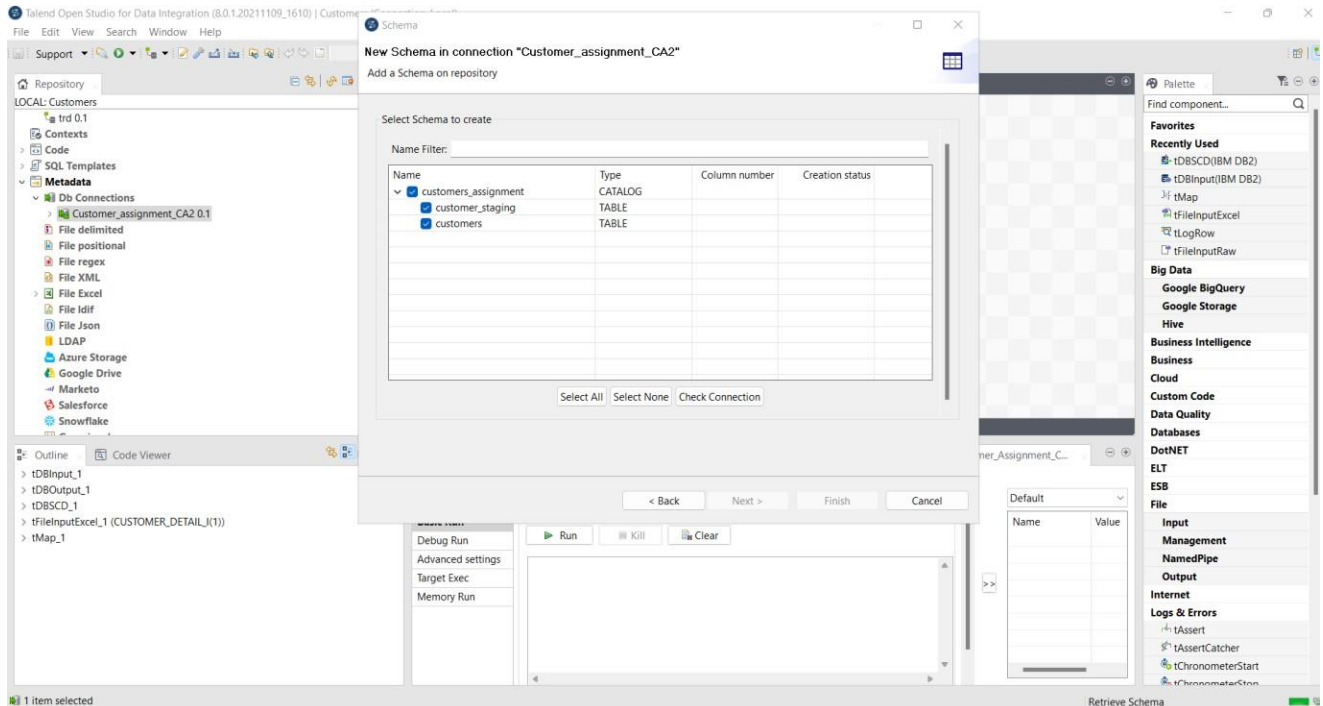


I updated the database connection with my MySQL credentials, such that it would connect to the databases, which I have already created in MySQL and its tables.



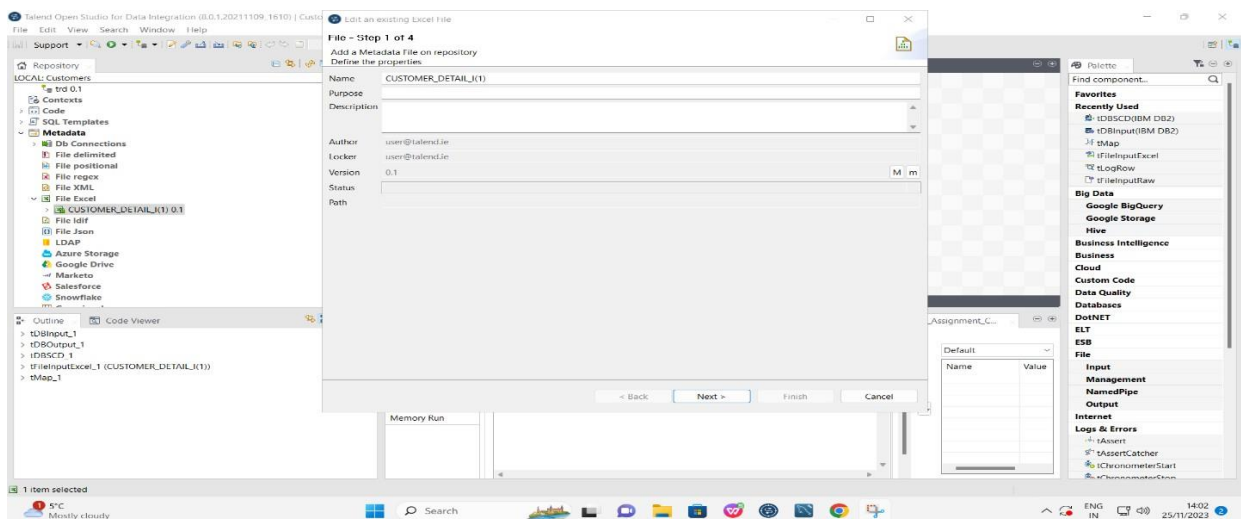
Test Connection was successful.

Step-3: Retrieving the Schema tables to Customer_Assignment_CA2



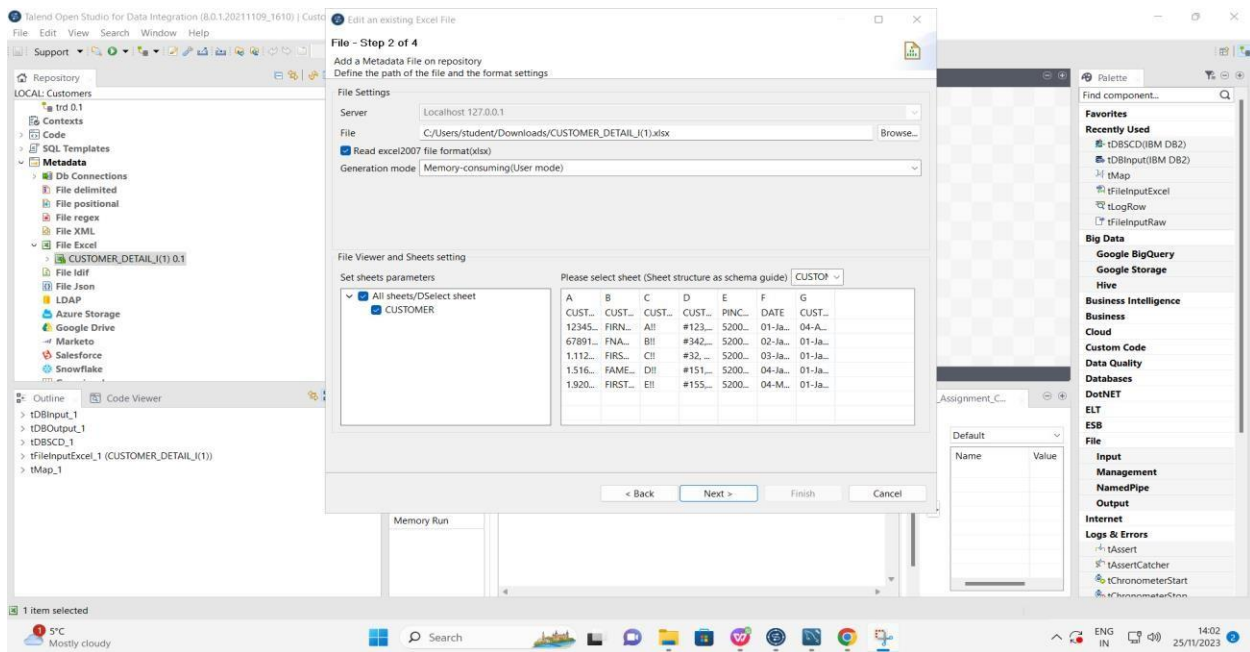
Above, I have added two schema tables “customer_staging” and “customers” to Customer_assignment_CA2 database connection.

Step – 1: Adding 1st source Excel file to the Metadata.

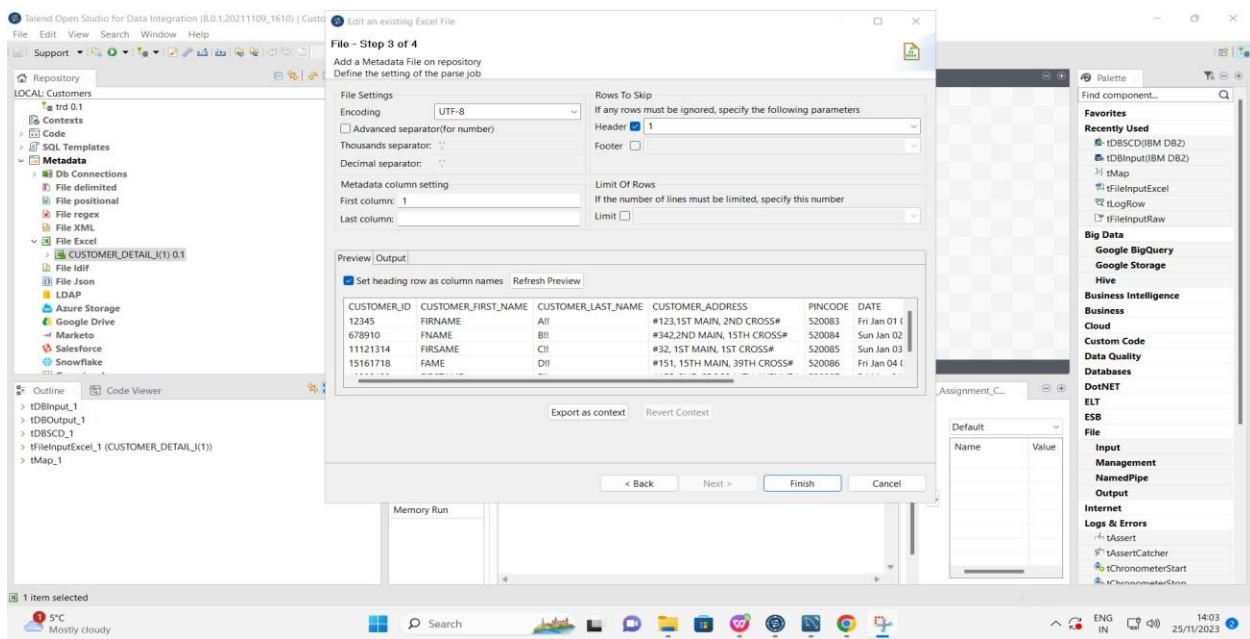


Adding Excel file & named the Excel file in repository as “CUSTOMER_DETAIL_I(1)”.

And then with the help of browse in File option added the path of the Excel file & selected the sheet in “Set sheets parameters” option.



Now, setting the headers so that we have set the correct customer headers and values, I clicked on the button “Refreshed Preview” to validate the headers and clicked the “Finish” button, like in the below picture.

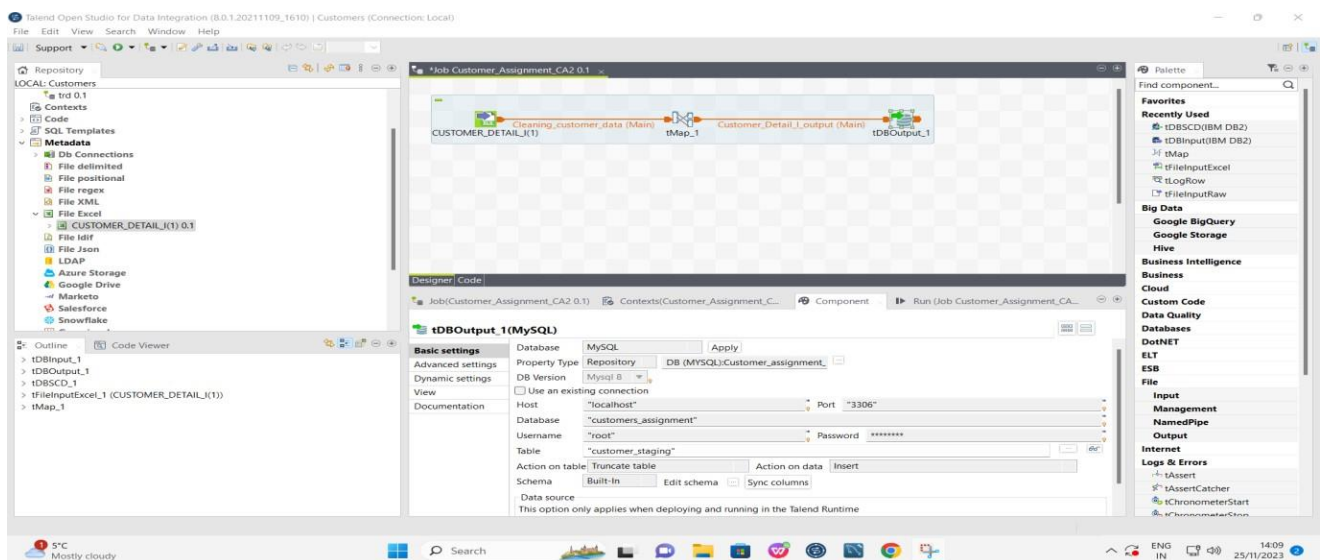


Step-2: Adding Components to Job Design.

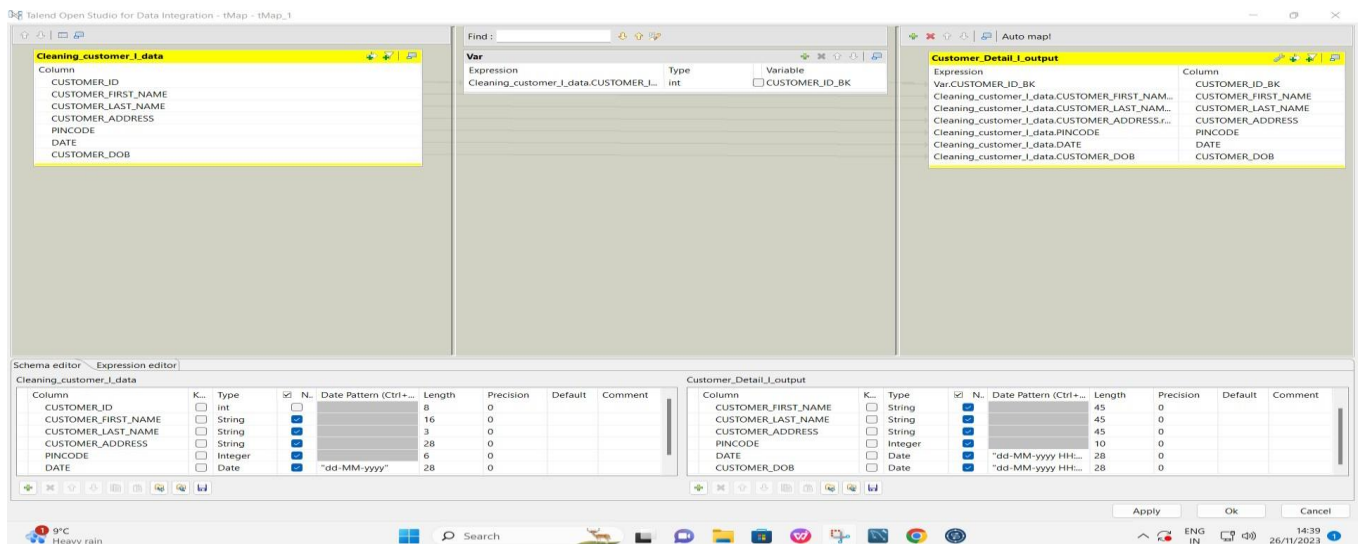
I have added Components (tFileInputExcel, tMap, tDBOutput), from Palette to the designer.

Double clicked on tDBOutput and in component added Database = MySQL, Property type = Repository and select the database(customers_assignment), Action on table = Truncate table, Schema = Repository and select table as customer_staging.

Now, created a link between the components with Row > Main.



Double clicked tMap and in tMap editor, I have cleaned data (CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, CUSTOMER_ADDRESS)



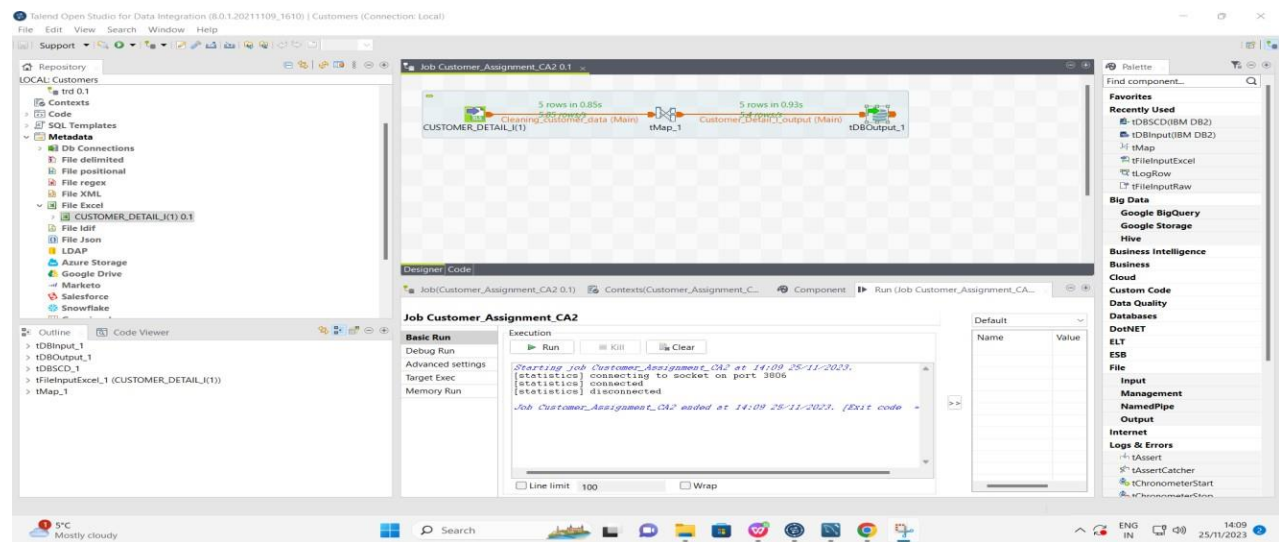
CUSTOMER_FIRST_NAME with extra spaces

CUSTOMER_LAST_NAME with !

CUSTOMER_ADDRESS with #

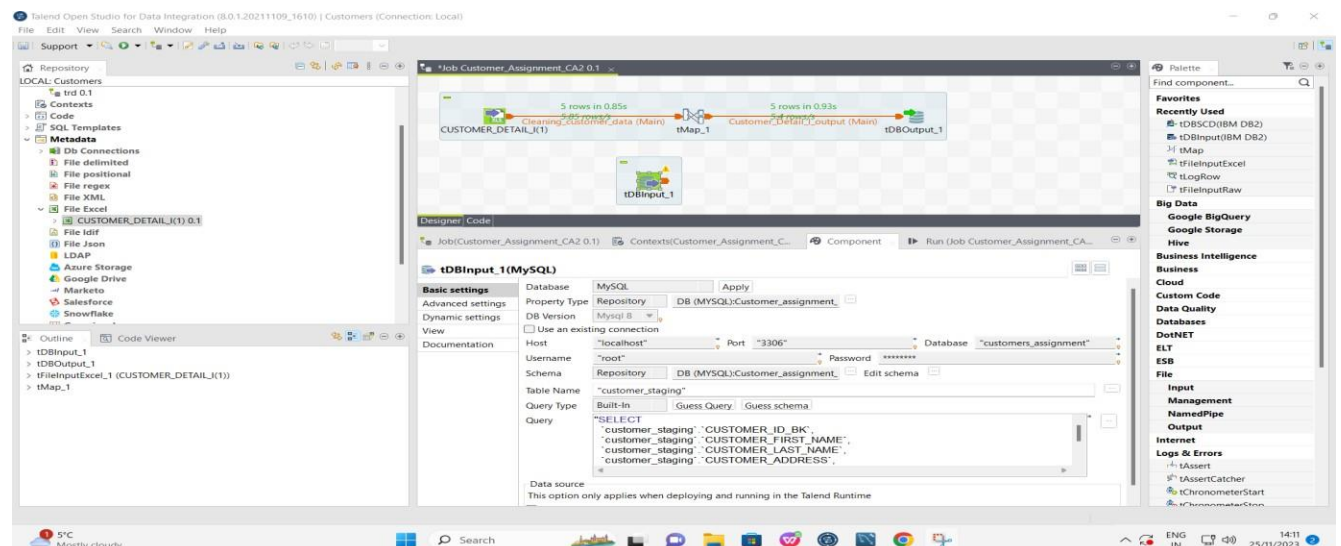
Step- 3: Run the Job

I have run the job and got 5 rows as output.

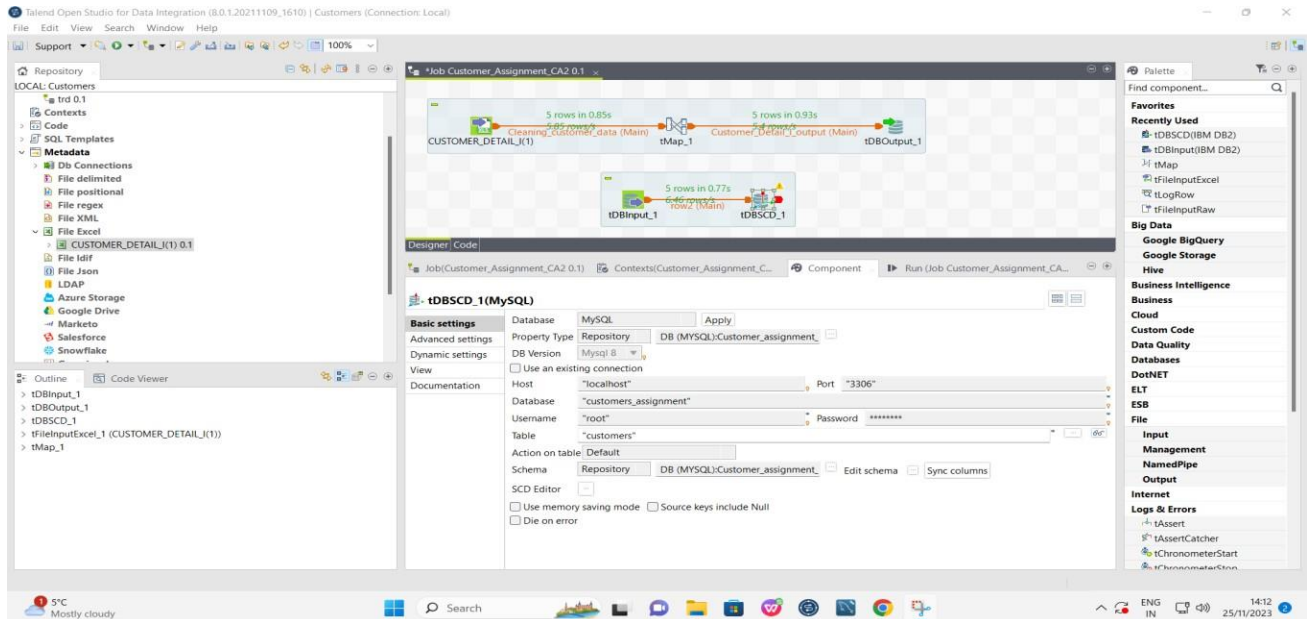


Step-4: Creating a Connection from Staging to Dimension.

Configuring Component (tDBInput) to job designer, and double clicked on tDBInput and in component added Database = MySQL, Property type = Repository and in that selecting the database(customers_assignment), Schema = Repository and in that selecting customer_staging.



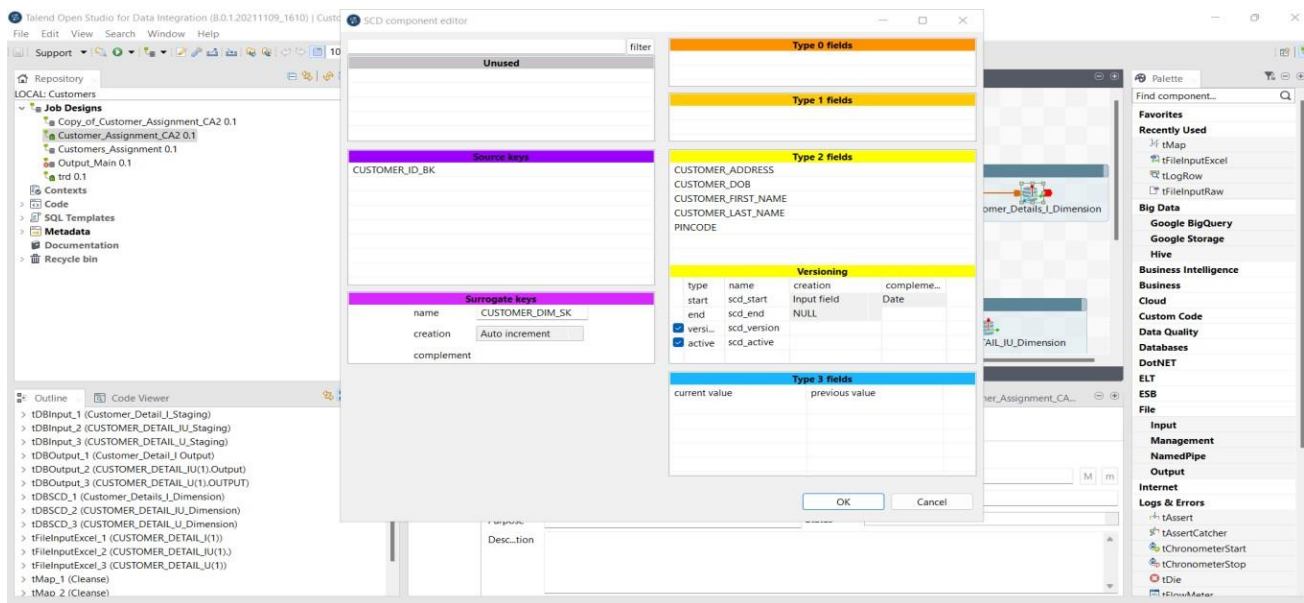
I have Added Component (tDBSCD_1) to job designer and configuring (tDBInput) in component added Database = MySQL, Property type = Repository and in that selecting the database (customers_assignment), Schema = Repository and in that selecting customer.



Selecting the SCD editor and configuring it by following below settings.

- CUSTOMER_ID_BK to the Source Key panel to use it as the key to ensure the uniqueness of the input data.
- CUSTOMER_DIM_SK to the Surrogate Key to links a record in the source to a record in the dimension table.
- Using Type 2 on CUSTOMER_ADDRESS, CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, PINCODE, so that it keeps a new version of the records when changes occur and maintains historical data.
- In versioning → start changed creation = input field and added Date so that it keeps track using date.
- Selected the version check box to hold the version numbers for the historical and current records in the SCD table and I also selected the active check box to add the column that will hold the true value for the current active record or the false value for the historical records in the SCD table.

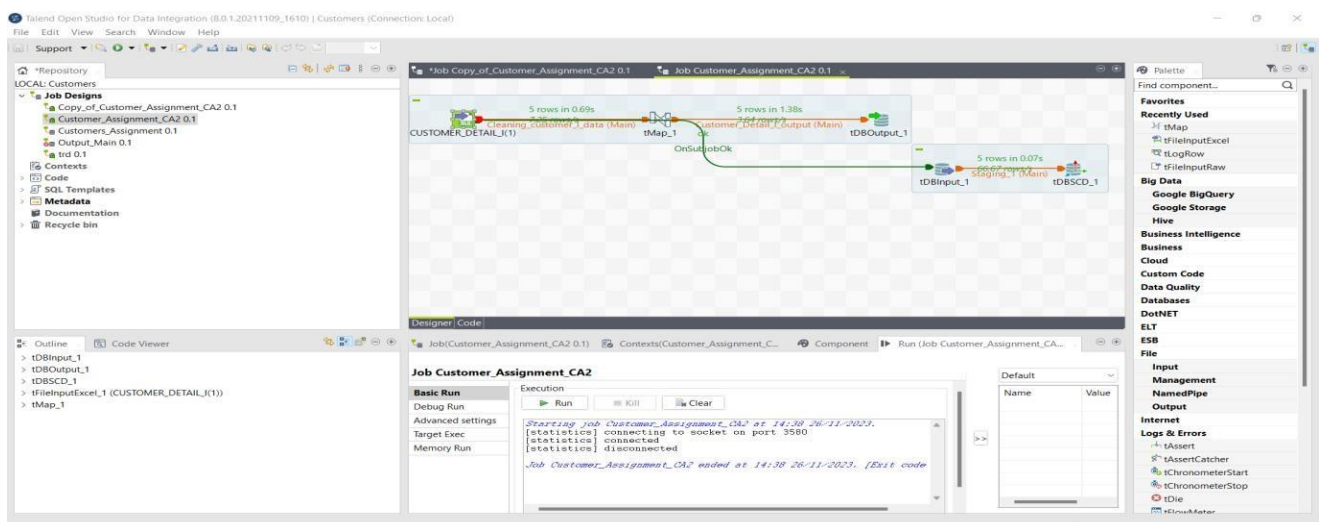
As below Screenshot.



Step – 5: Executing the Job

Joining Cleansed data to Dimension using OnSubjobOk.

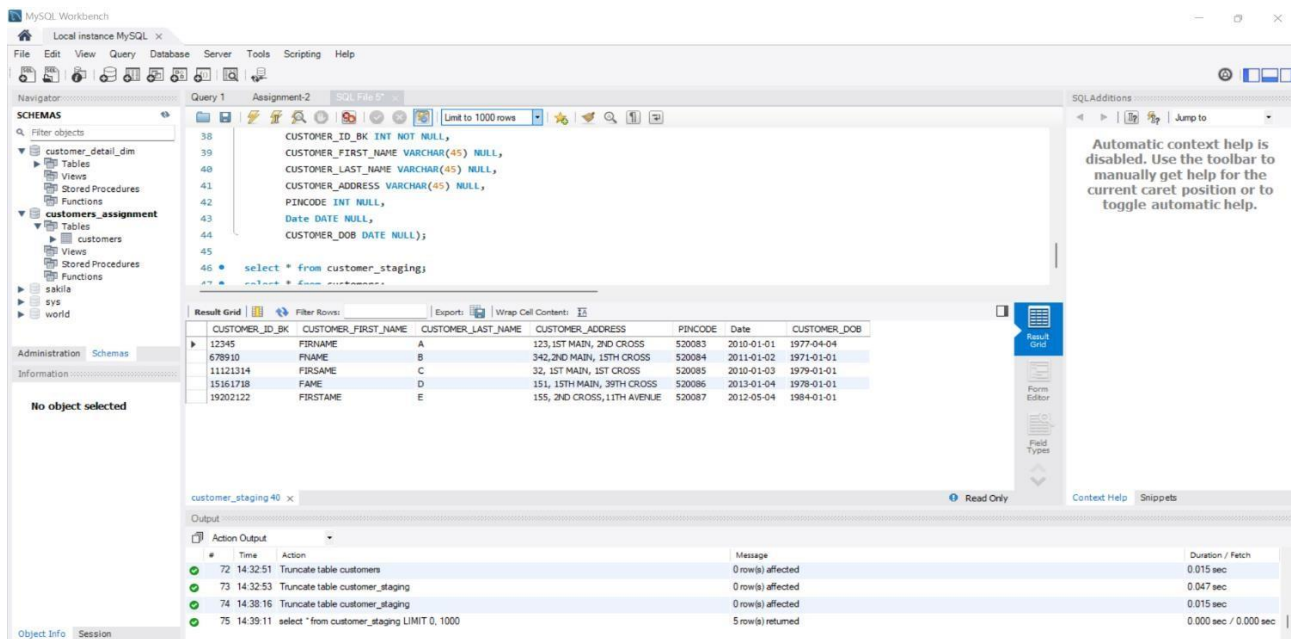
Running the Connection and could see output as 5 rows in Talend.



Step - 6: Executing query in MySQL

Now running the customer_staging table to check if the data is cleaned and getting all the row values correctly in CUSTOMER_ID_BK, CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, CUSTOMER_ADDRESS, PINCODE, DATE, CUSTOMER_DOB.

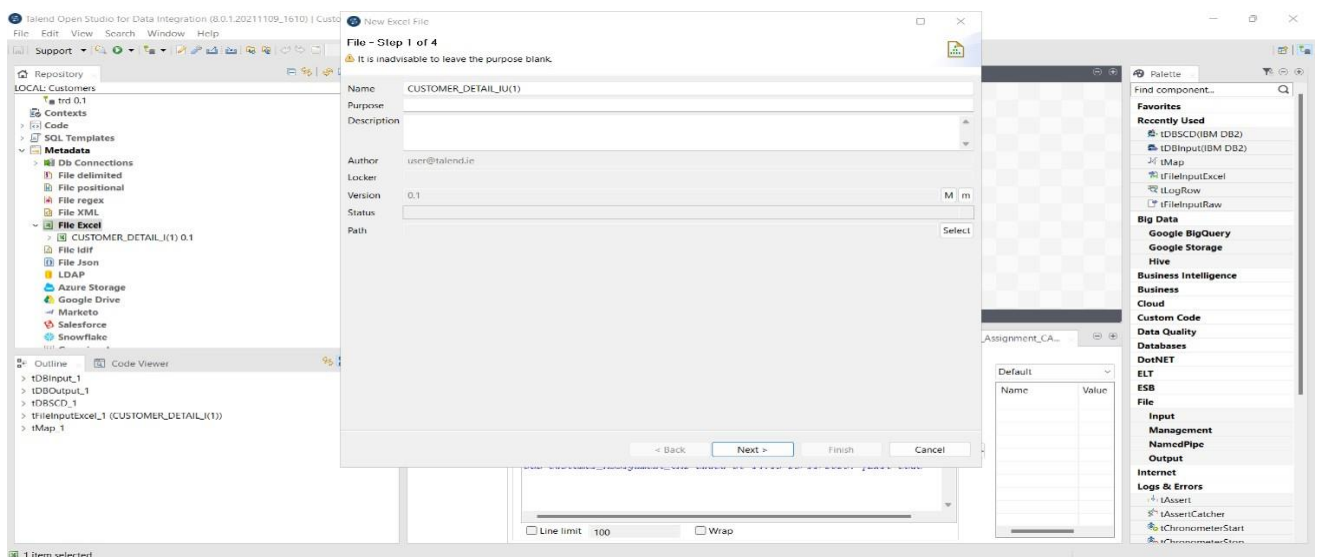
OUTPUT- 1

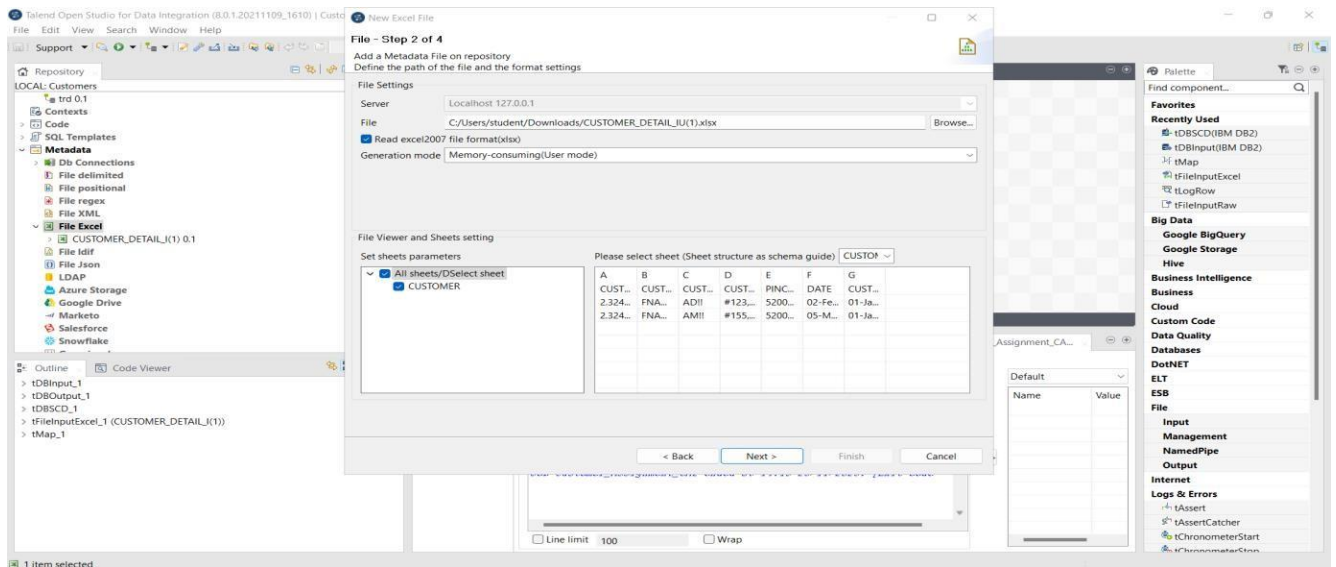


Completed cleaning and staging the 1st Excel File to customer_staging table in MySQL.

Step- 1: Adding 2nd source Excel file to the Metadata (Reference same as 1st Excel)

As I have done the first Excel file above, I need to do the same steps, adding components and their values to (tFileInputExcel, tMAP, tDBInput, tDBOutput, tDBDSCD).





Added the Excel file to the Repository.

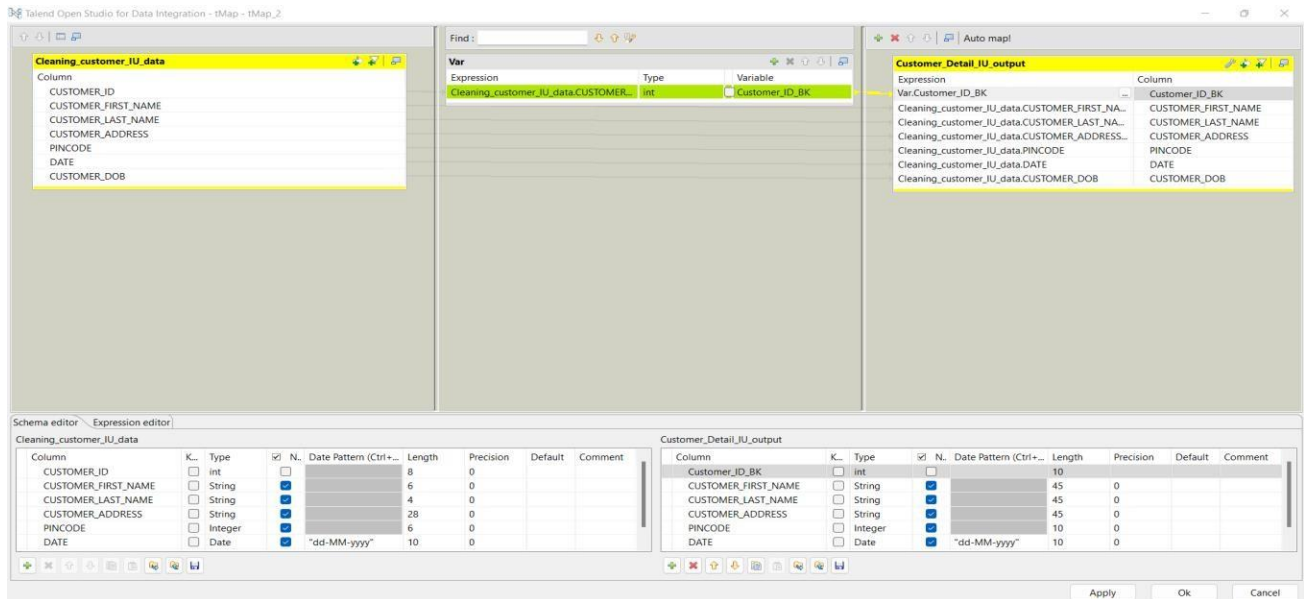
Step-2: Adding Components to Job Design.

Double clicked tMap and in tMap editor, I have cleaned data (CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, CUSTOMER_ADDRESS)

CUSTOMER_FIRST_NAME with extra spaces

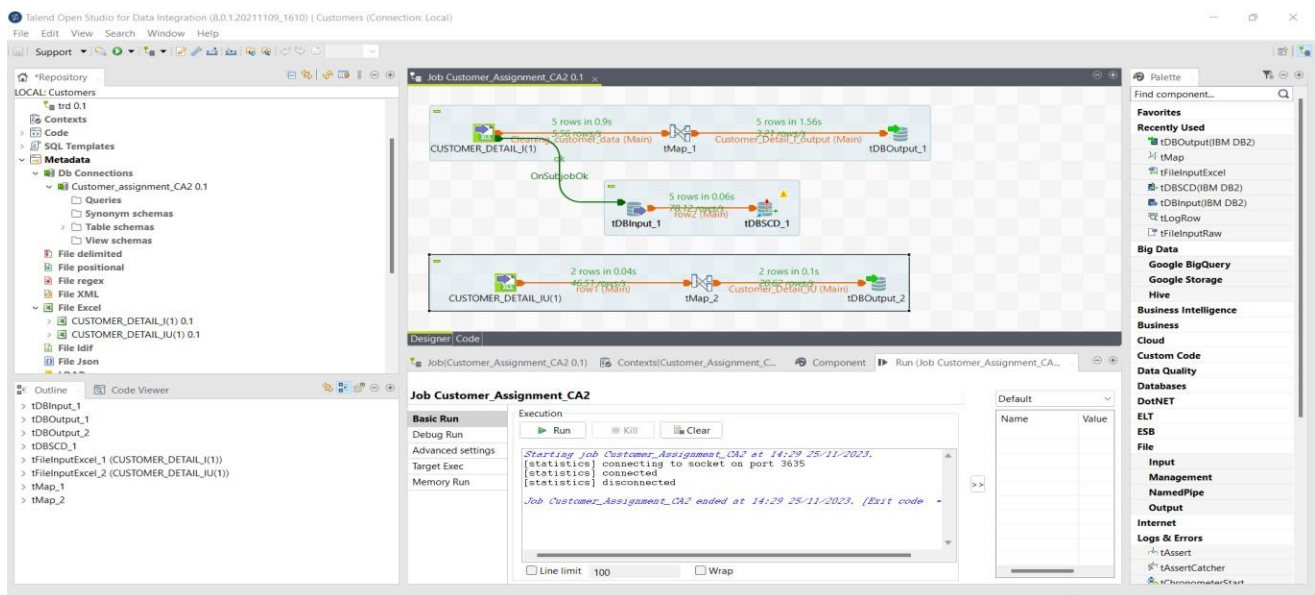
CUSTOMER_LAST_NAME with !

CUSTOMER_ADDRESS with #



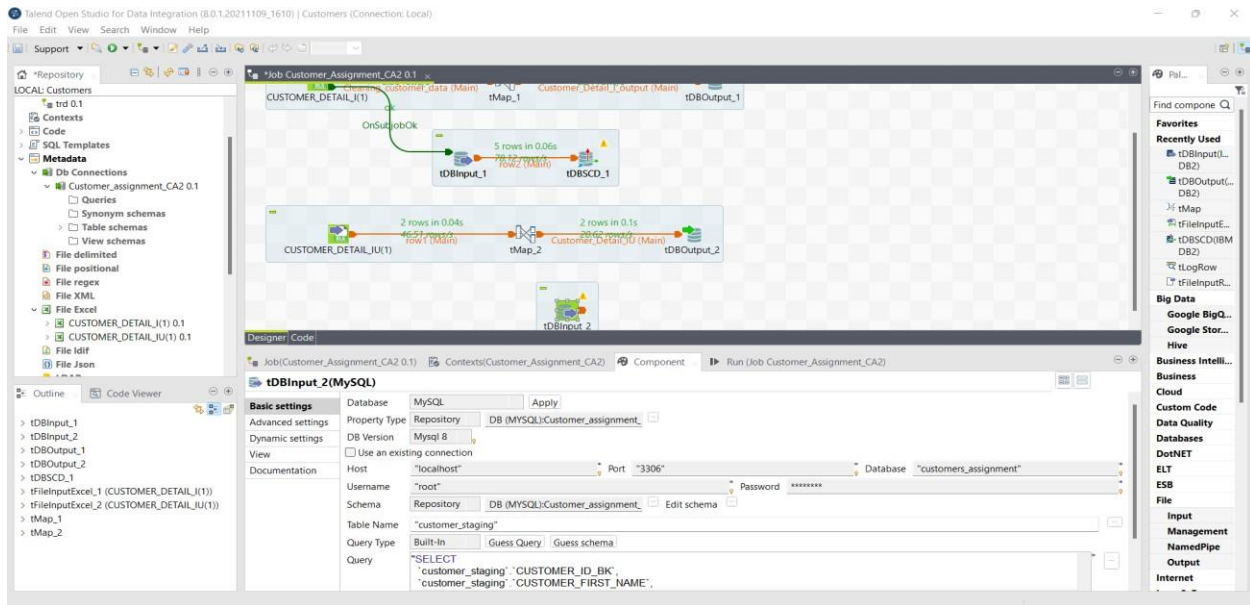
Step- 3: Run the Job

I have run the job and got 2 rows as output.



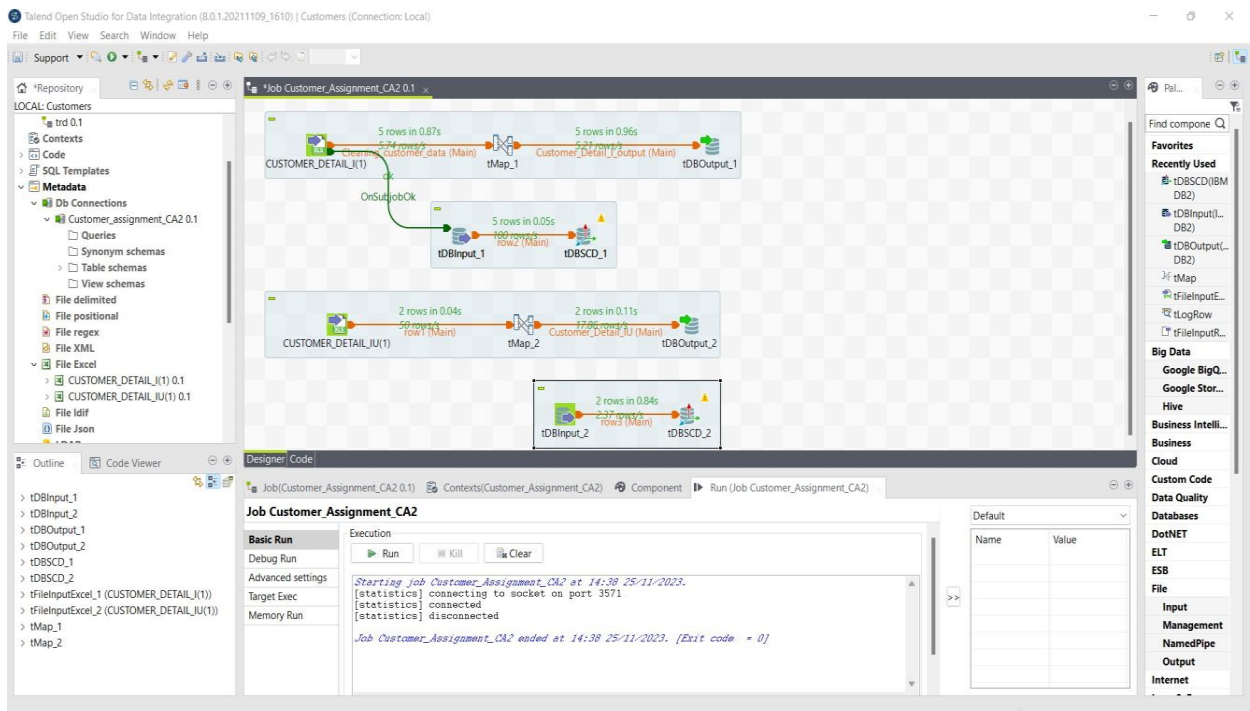
Step-4: Creating a Connection from Staging to Dimension.

Adding & Configuring Component (tDBInput) to job designer. Adding the same component values as above Excel File.

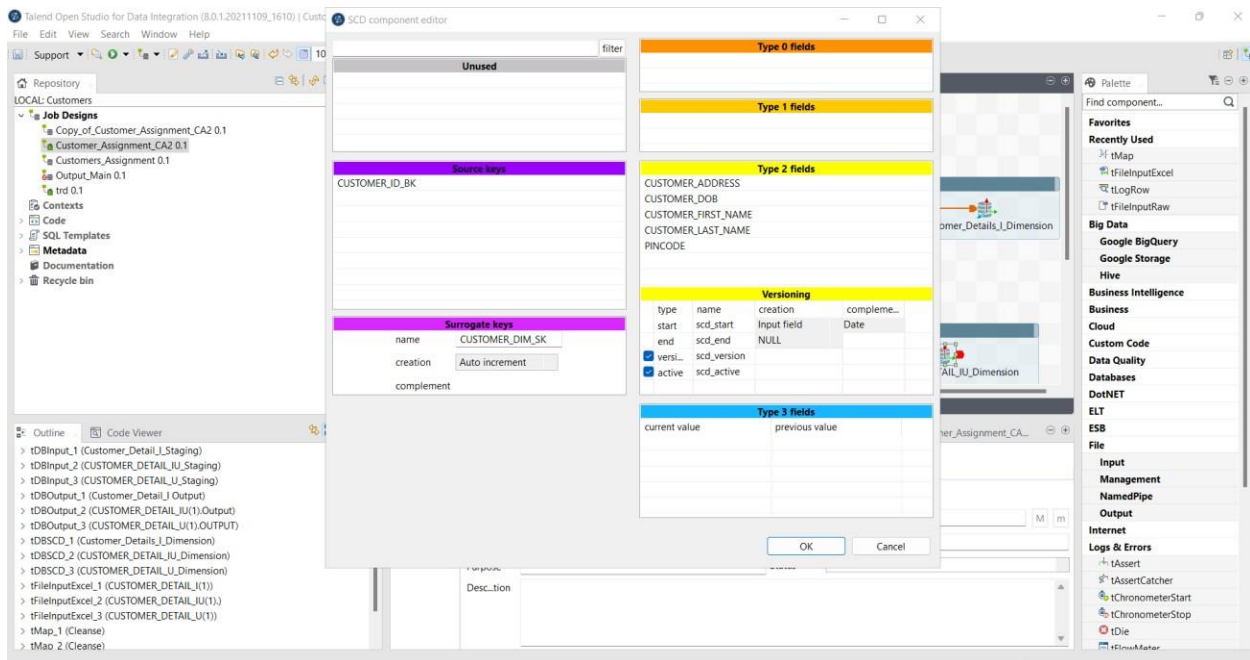


I have Added Component (tDBSCD_1) to job designer and configuring (tDBInput) in component added Database = MySQL, Property type = Repository and in that selecting the database (customers_assignment), Schema = Repository and in that selecting customer.

Connecting tDBInput_2 and tDBSCD_2



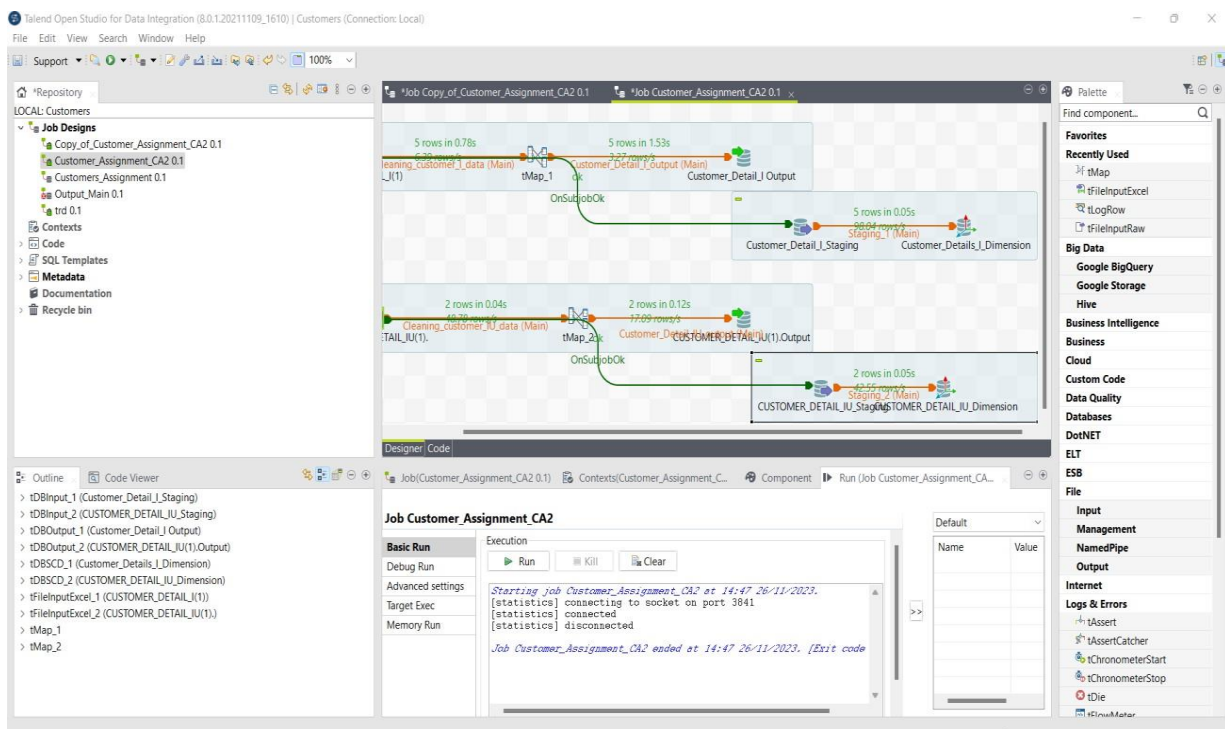
Selecting the SCD editor and configuring it same as above settings.



Step – 5: Executing the Job

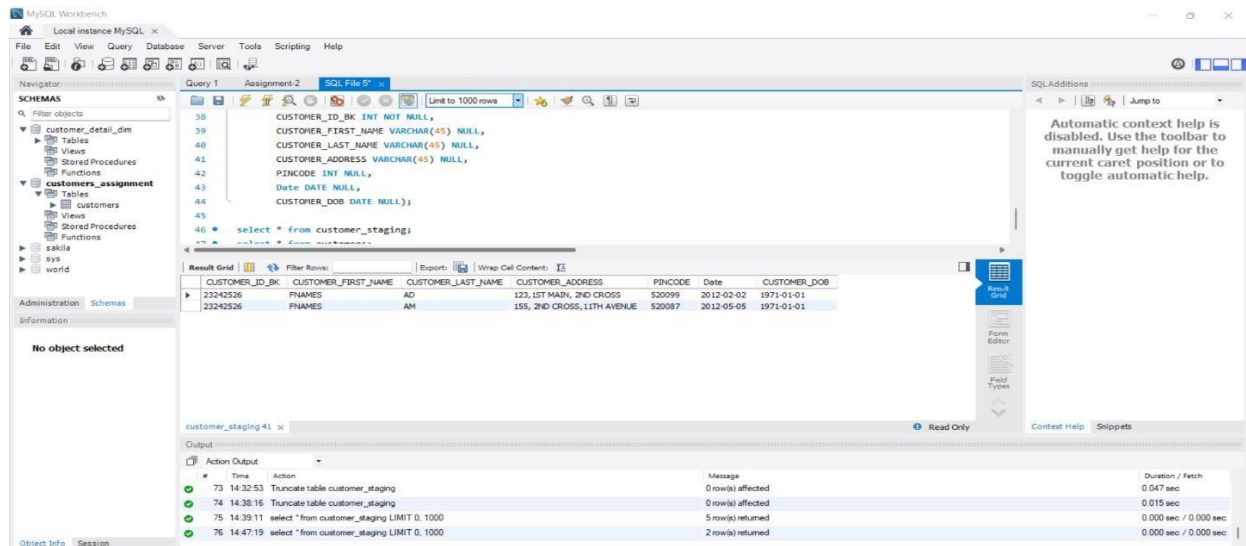
Joining Cleansed data to Dimension using OnSubjobOk.

Running the Connection and could see output as 2 rows in Talend.



Step - 6: Executing query in MySQL

OUTPUT- 2

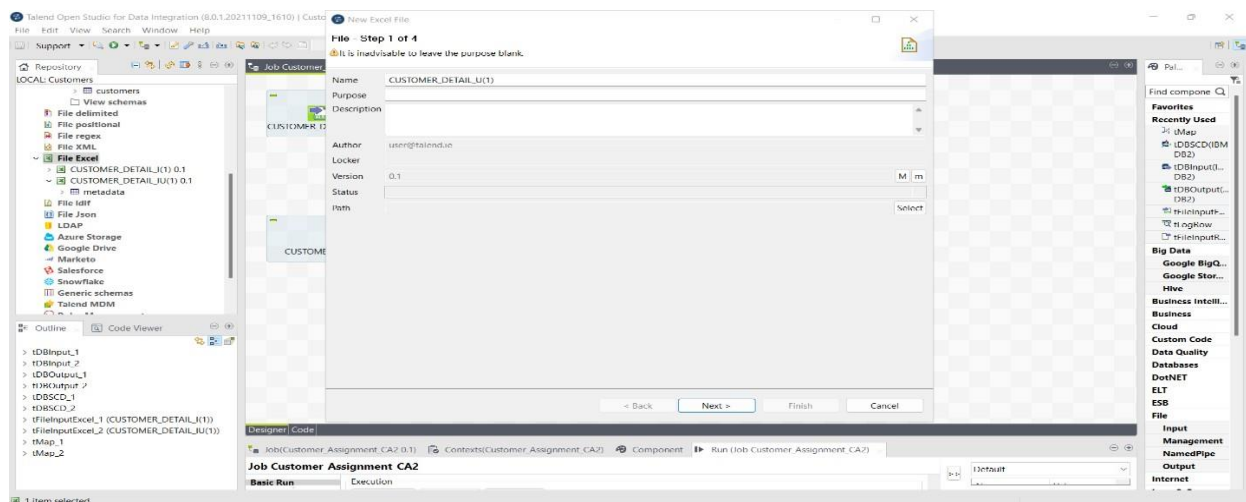


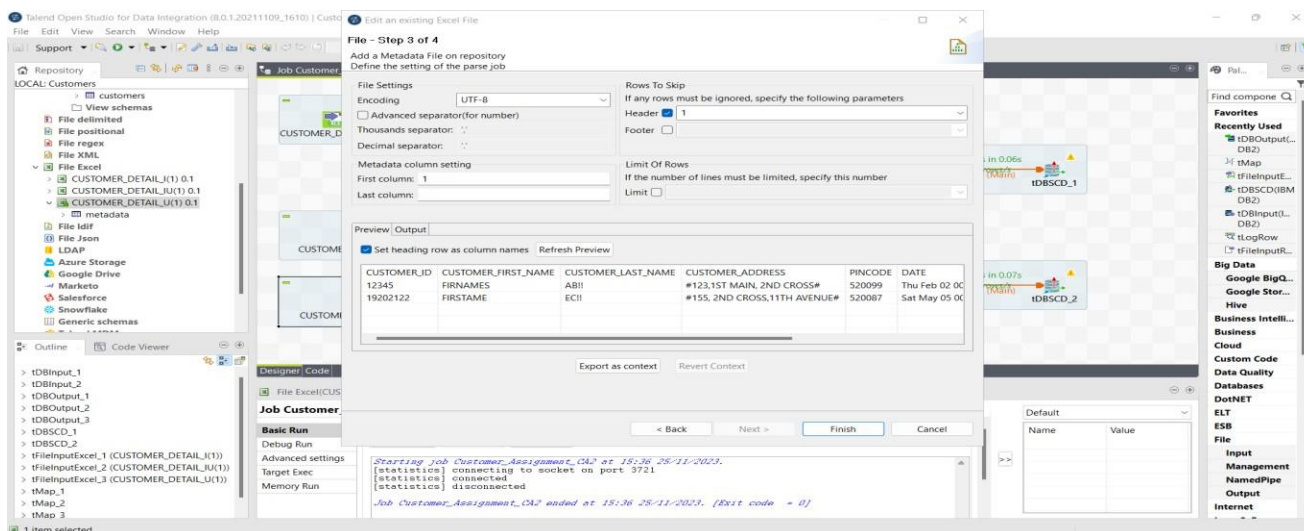
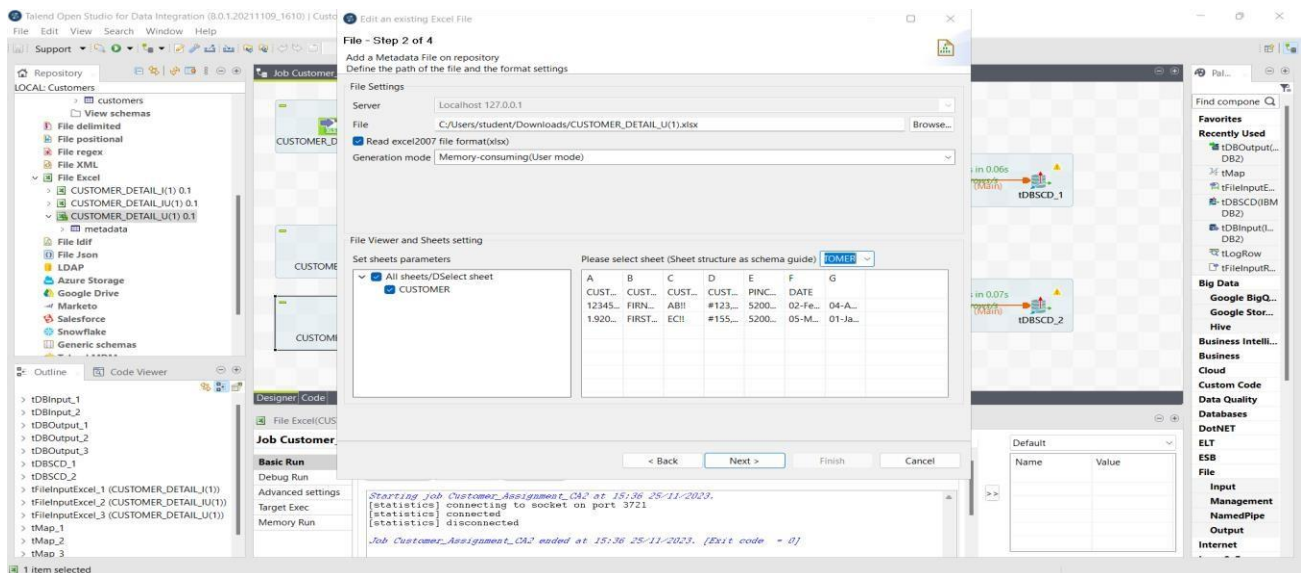
Completed cleaning and staging the 2nd Excel File to customer_staging table in MySQL.

Step- 1: Adding 3rd source Excel file to the Metadata (Reference same as 1st Excel)

As I have done the first Excel file above, I need to do the same steps, adding components and their values to (tFileInputExcel, tMAP, tDBInput, tDBOutput, tDBDSCD).

Adding Excel file & named it in repository as “CUSTOMER_DETAIL_U(1)”





Added the Excel file to the Repository.

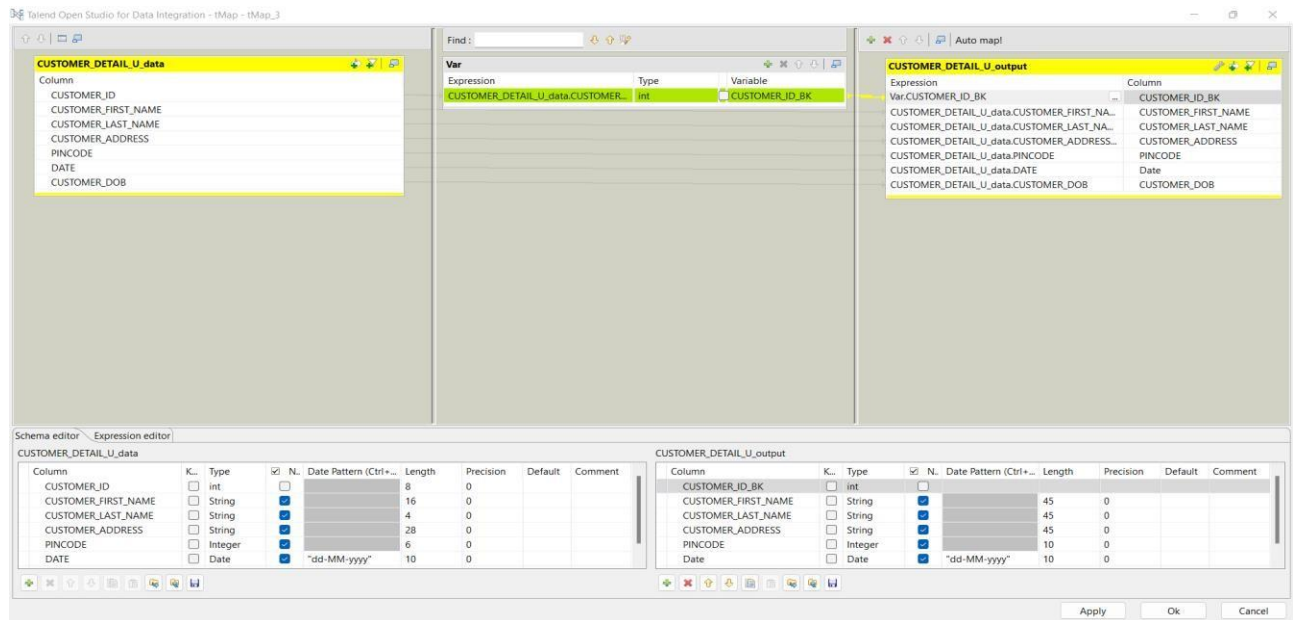
Step-2: Adding Components to Job Design.

Double clicked tMap and in tMap editor, I have cleaned data
(CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, CUSTOMER_ADDRESS)

CUSTOMER_FIRST_NAME with extra spaces

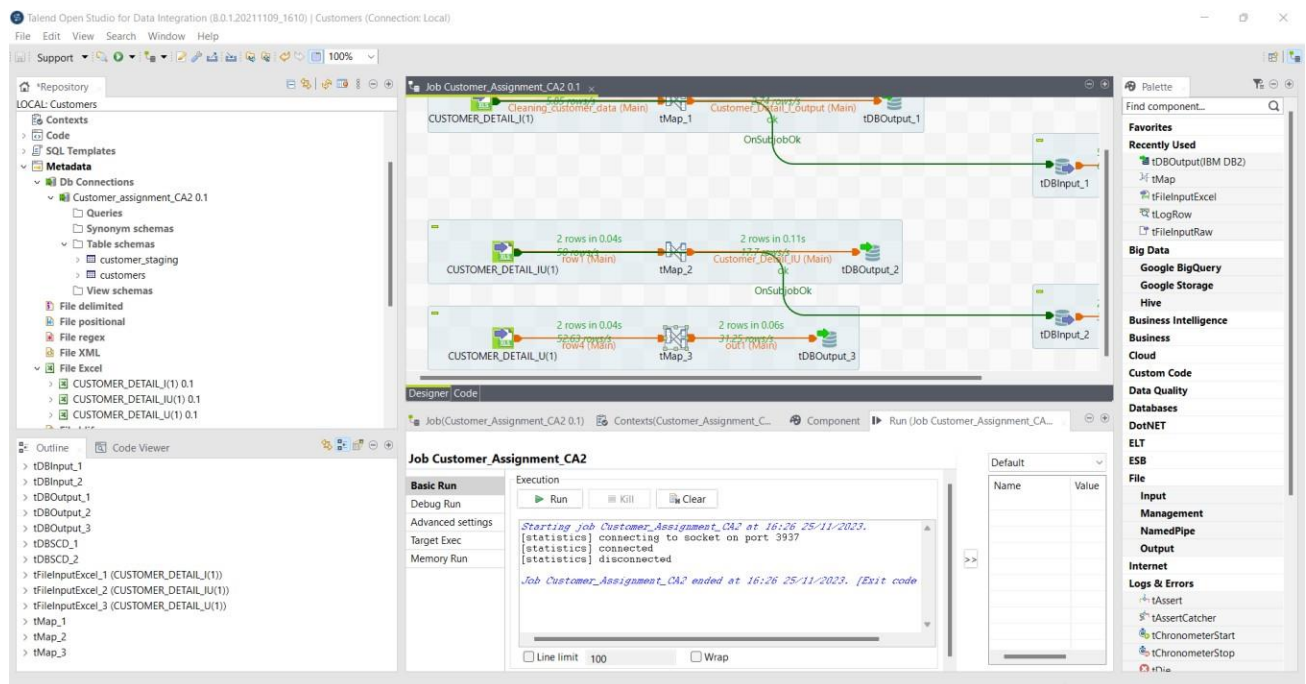
CUSTOMER_LAST_NAME with !

CUSTOMER_ADDRESS with #



Step- 3: Run the Job

I have run the job and got 2 rows as output for tDBOutput_3



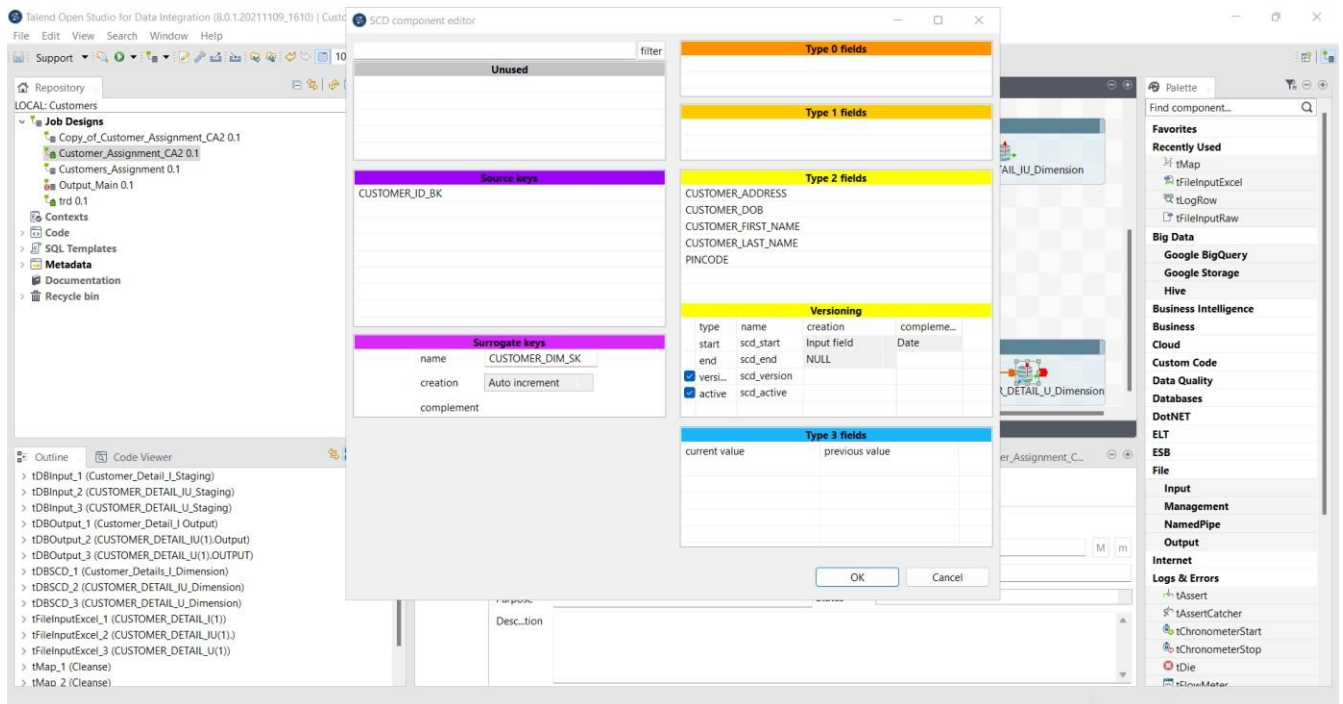
Step-4: Creating a Connection from Staging to Dimension.

Adding & Configuring Component (tDBInput) to job designer. Adding the same component values as above Excel File. I have Added Component (tDBSCD_1) to job designer and configuring (tDBInput) in component added Database = MySQL,

Property type = Repository and in that selecting the database
(customers_assignment), Schema = Repository and in that selecting customer.

Connecting tDBInput_2 and tDBSCD_2

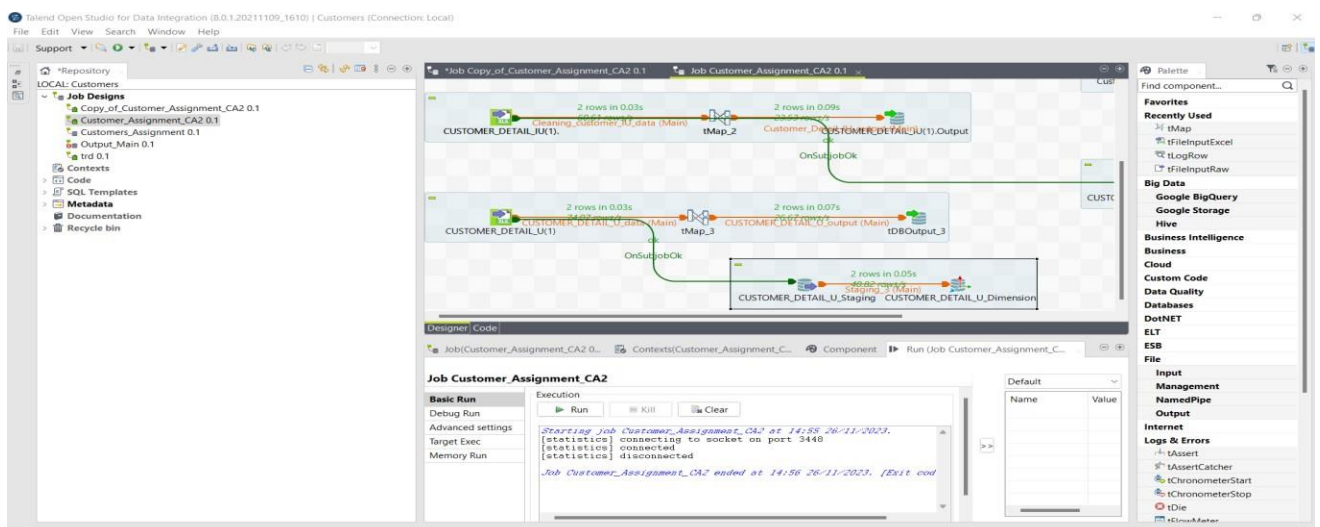
Selecting the SCD editor and configuring it same as above settings.



Step – 5: Executing the Job

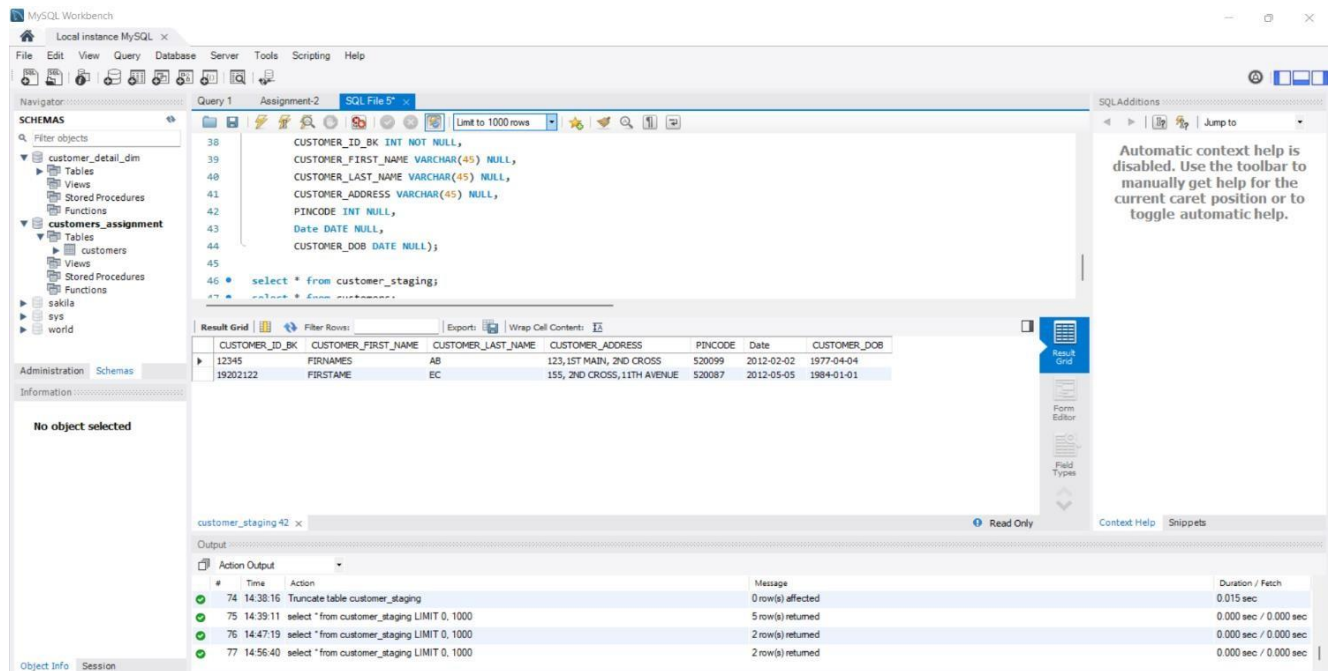
Joining Cleansed data to Dimension using OnSubjobOk.

Running the Connection and could see output as 2 rows in Talend.



Step - 6: Executing query in MySQL

OUTPUT- 3



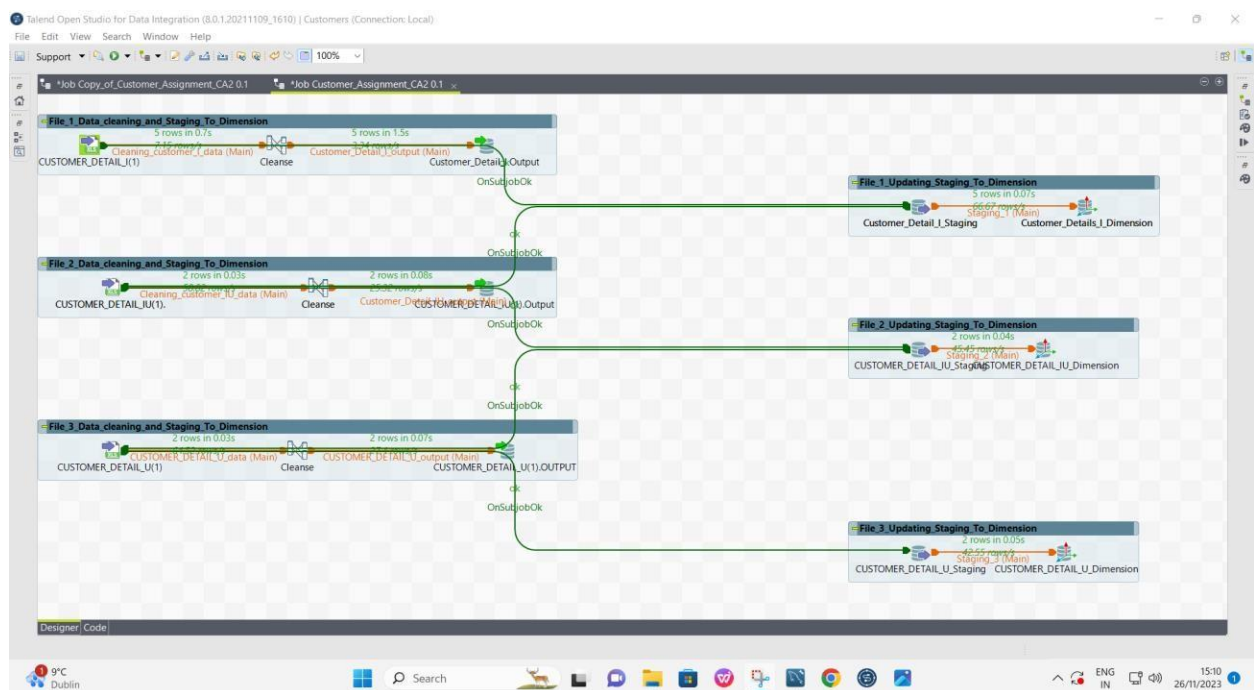
The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying a SQL query that truncates the 'customer_staging' table and then inserts data from 'customer_staging' into 'customer_details'. The query is as follows:

```
TRUNCATE TABLE customer_staging;
INSERT INTO customer_details (
  CUSTOMER_ID, FIRST_NAME, LAST_NAME, ADDRESS, ADDRESS2, CITY, STATE, COUNTRY,
  POSTAL_CODE, PHONE_NUMBER, EMAIL, GENDER, DOB, CREATED_DATE,
  UPDATED_DATE, STATUS
)
SELECT * FROM customer_staging;
```

The 'Result Grid' shows the output of the query, which is a table with 15 columns: CUSTOMER_ID, FIRST_NAME, LAST_NAME, ADDRESS, ADDRESS2, CITY, STATE, COUNTRY, POSTAL_CODE, PHONE_NUMBER, EMAIL, GENDER, DOB, CREATED_DATE, and UPDATED_DATE. The table contains 15 rows of data.

The 'Output' tab is also visible, showing the execution progress of the query. It indicates that the query was executed successfully and that 15 rows were returned.

Connecting all the Job designing using OnSubjobOk so that it runs and executes all the three files output one by one.



Overall Output (Merged)

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' tree with 'customer_detail_dim' and 'customers_assignment' expanded. The main editor shows a SQL query with line numbers 39 to 47. The query defines columns for a table and then selects data from 'customer_staging' and 'customers'. The 'Result Grid' shows a table with 9 rows and 8 columns: CUSTOMER_DIM_SK, CUSTOMER_ID_BK, CUSTOMER_FIRST_NAME, CUSTOMER_LAST_NAME, CUSTOMER_ADDRESS, PINCODE, CUSTOMER_DOB, and SCD_START. The bottom 'Output' tab shows the execution log with four entries, all successful.

```
39      CUSTOMER_FIRST_NAME VARCHAR(45) NULL,  
40      CUSTOMER_LAST_NAME VARCHAR(45) NULL,  
41      CUSTOMER_ADDRESS VARCHAR(45) NULL,  
42      PINCODE INT NULL,  
43      Date DATE NULL,  
44      CUSTOMER_DOB DATE NULL);  
45  
46 • select * from customer_staging;  
47 • select * from customers;
```

	CUSTOMER_DIM_SK	CUSTOMER_ID_BK	CUSTOMER_FIRST_NAME	CUSTOMER_LAST_NAME	CUSTOMER_ADDRESS	PINCODE	CUSTOMER_DOB	SCD_START
1	12345	FIRNAME	A	123, 1ST MAIN, 2ND CROSS	520083	1977-04-04 00:00:00	2010-01-01 00:00:00	
2	678910	FINAME	B	342, 2ND MAIN, 15TH CROSS	520084	1971-01-01 00:00:00	2011-01-02 00:00:00	
3	11121314	FIRNAME	C	32, 1ST MAIN, 1ST CROSS	520085	1979-01-01 00:00:00	2010-01-03 00:00:00	
4	15161718	FAME	D	151, 15TH MAIN, 39TH CROSS	520086	1978-01-01 00:00:00	2013-01-04 00:00:00	
5	19202122	FIRSTNAME	E	155, 2ND CROSS, 11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-04 00:00:00	
6	23242526	FINAMES	AD	123, 1ST MAIN, 2ND CROSS	520099	1971-01-01 00:00:00	2012-02-02 00:00:00	
7	23242526	FINAMES	AM	155, 2ND CROSS, 11TH AVENUE	520087	1971-01-01 00:00:00	2012-05-05 00:00:00	
8	12345	FIRNAMES	AB	123, 1ST MAIN, 2ND CROSS	520099	1977-04-04 00:00:00	2012-02-02 00:00:00	
9	19202122	FIRSTNAME	EC	155, 2ND CROSS, 11TH AVENUE	520087	1984-01-01 00:00:00	2012-05-05 00:00:00	
10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

#	Time	Action	Message	Duration / Fetch
77	14:56:40	select * from customer_staging LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
78	15:01:40	Truncate table customers	0 row(s) affected	0.032 sec
79	15:02:51	select * from customer_staging LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
80	15:02:56	select * from customers LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

Summary: I added all three Excel files to the job design one by one, and after cleaning the file using tMap, I have run all three jobs separately. Which got executed, and gave three different outputs (output-1, output-2, output-3) separately in MySQL using the staging table (customer_staging).

Then I merged all three jobs using SCD and executed the job to get the overall output (overall output- merged) in MySQL using the dimension table (customers).