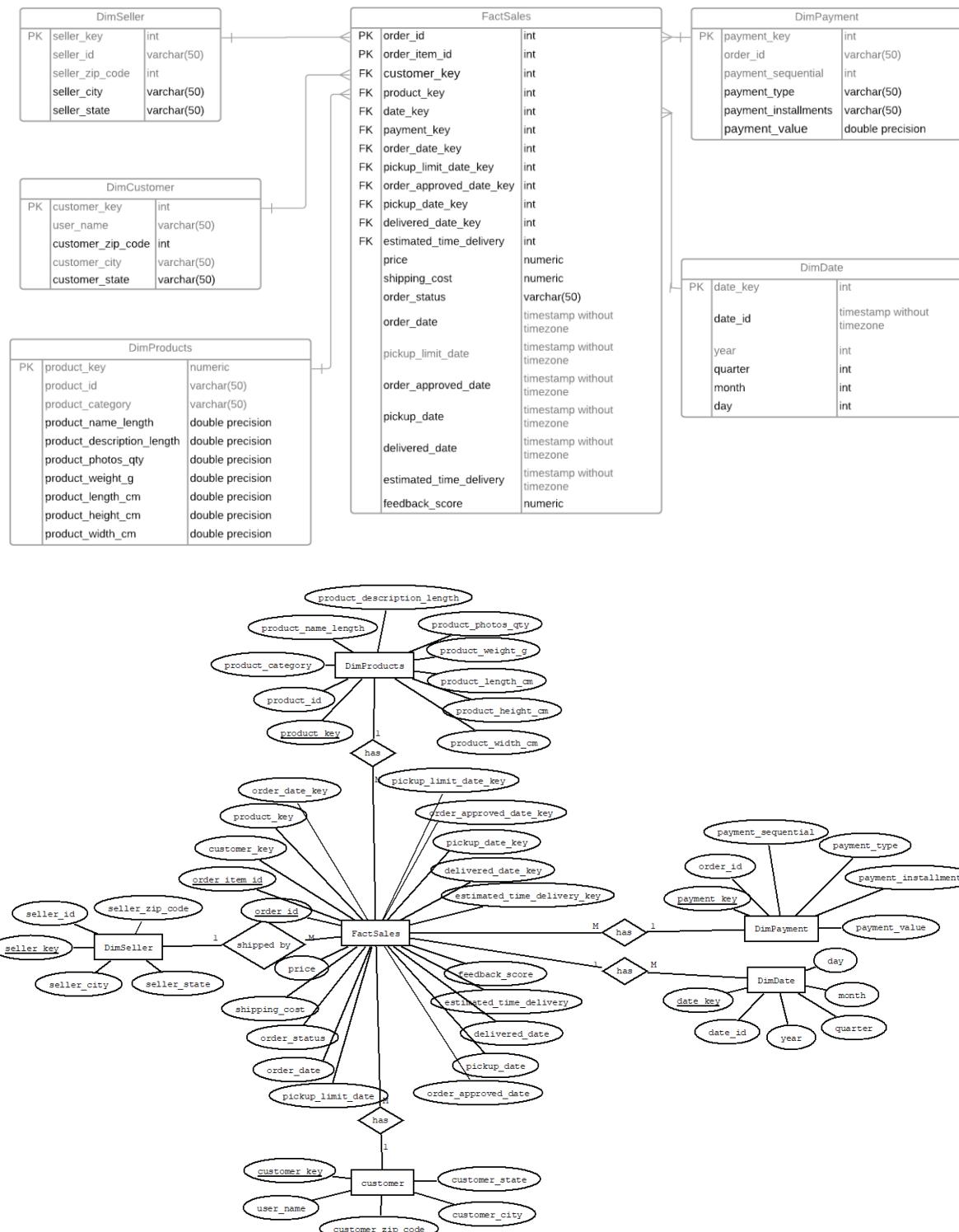


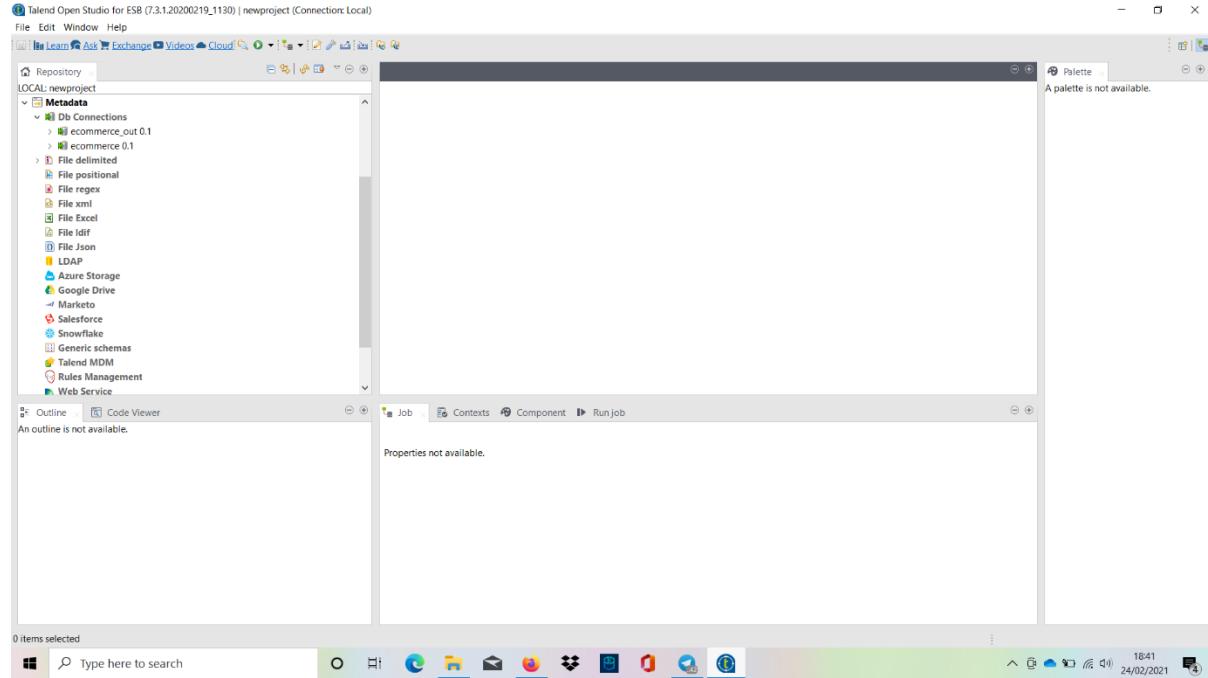
Progress Pembuatan Dimensi (20-23 Februari 2021)

Rancangan Star Schema yang akan dibuat



Note

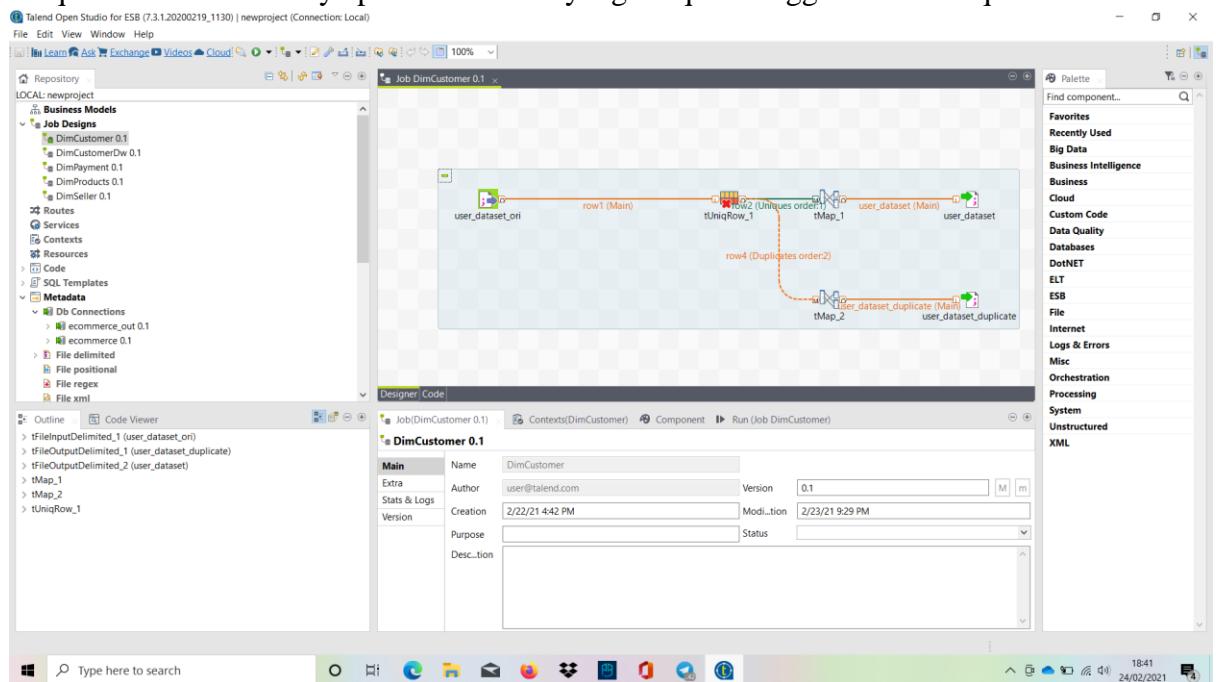
Dalam melakukan ETL, saya menggunakan aplikasi Talend versi 7.3.1 dengan JDK versi 8. Untuk input dan output saya koneksi langsung dengan SQL Server. Namun agar jobnya dapat di run oleh kakak-kakak mentor, saya akan memberikan file talend dimana input dan outputnya berbentuk file. Namun untuk penjelasan, saya menggunakan job dimana outputnya langsung ke database agar lebih cepat prosesnya.



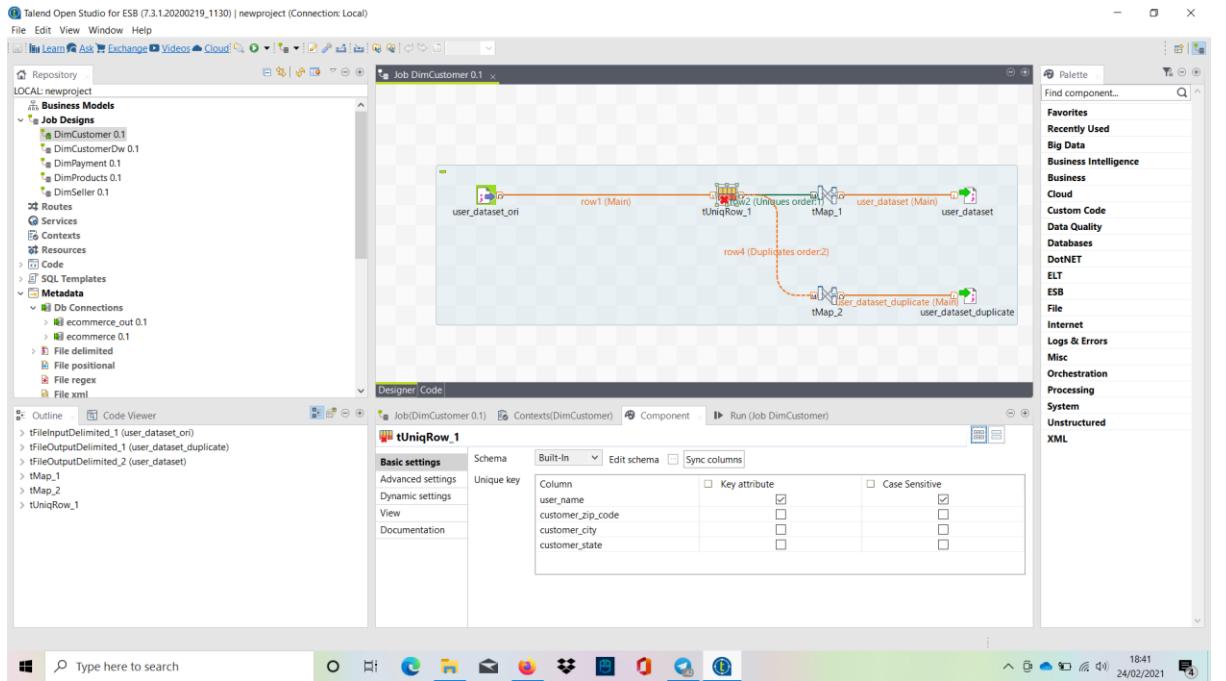
Pembuatan Dimensi Menggunakan Talend

1. DimCustomer

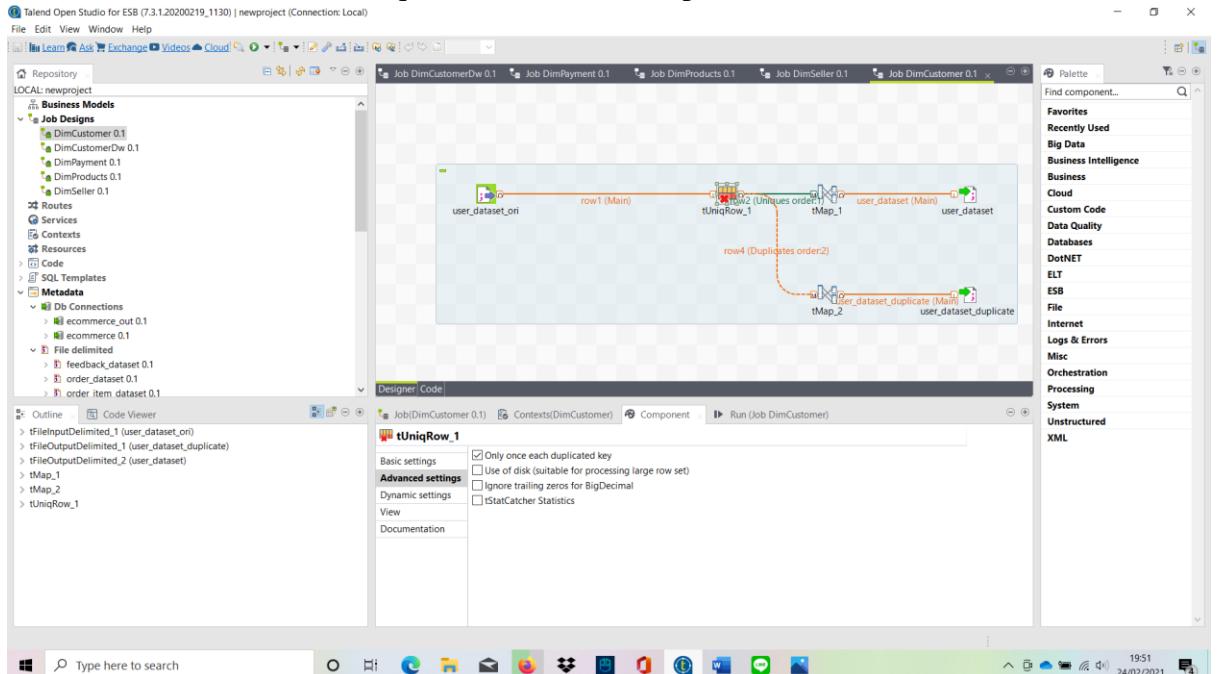
Pada pembuatan DimCustomer, dimasukkan file input “user_dataset.csv”. Pada file tersebut terdapat beberapa user_name yang duplikat, padahal seharusnya id tersebut unique. Maka dari itu saya pisahkan antara yang unique menggunakan tUniqRow.



Di tUniqRow, saya mencentang bagian key attribute pada user_name karena bagian ini yang saya inginkan untuk unique. Kemudian saya memilih case sensitive juga untuk membedakan huruf besar dan kecil.



Kemudian pada bagian Advanced Settings di tUniqrow, saya memilih only one each duplicated key. Ini menandakan bahwa saya hanya ingin user_name ini menjadi key, maka user_name ini harus unique dan tidak boleh duplikat.



Kemudian saya memetakan tmap yang tidak memiliki user_name yang duplikat. Untuk user_dataset yang memiliki user_name yang unique, tidak boleh ada yang null. Dan untuk tipe data serta Panjang data sesuai dengan apa yang tercantum pada gambar di

bawah

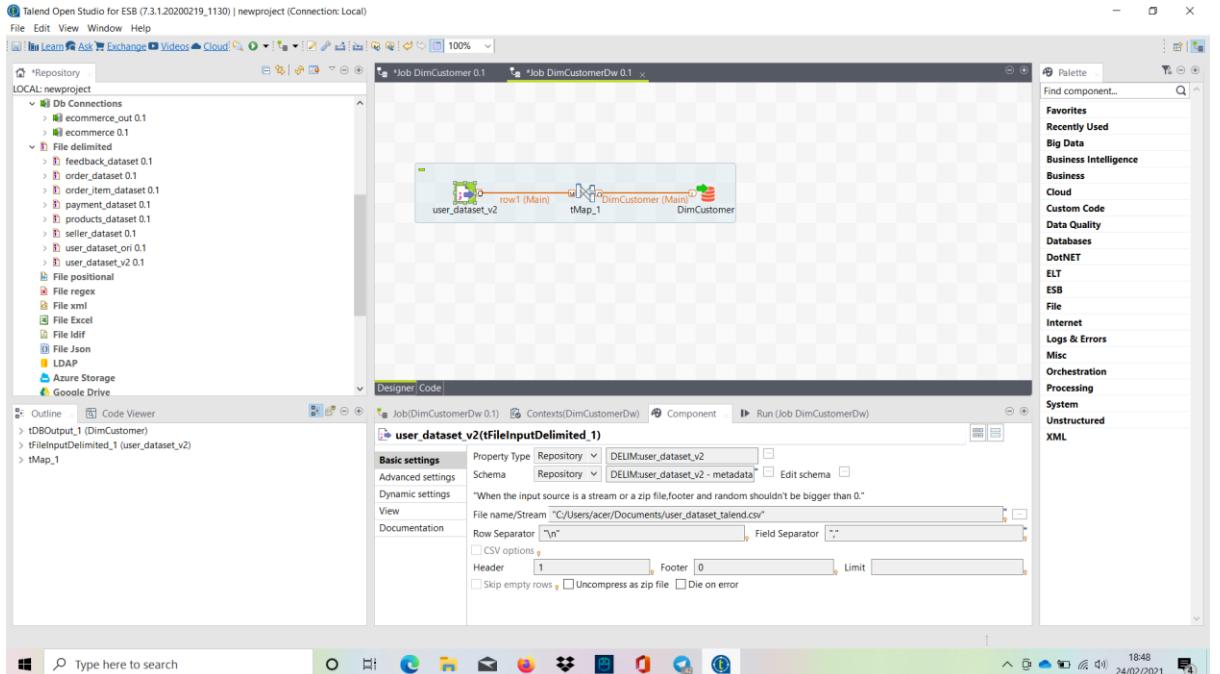
ini.

The screenshot shows the Talend Open Studio interface for a tMap component named 'tMap_1'. It consists of three main panels: a left panel for 'row2' with columns 'user_name', 'customer_zip_code', 'customer_city', and 'customer_state'; a central panel for 'Var' (Variables) which is currently empty; and a right panel for 'user_dataset' with the same four columns. Below these panels is a 'Schema editor' section containing two tables: 'row2' and 'user_dataset'. Both tables have columns 'user_name', 'customer_zip_code', 'customer_city', and 'customer_state'. The 'row2' table has 'Type' columns (String, Integer, String, String) and 'Length' values (32, 5, 28, 17). The 'user_dataset' table has 'Type' columns (String, int, String, String) and 'Length' values (50, 10, 50, 50). At the bottom of the interface is a Windows taskbar with various icons and a system tray showing the date and time (24/02/2021, 18:42).

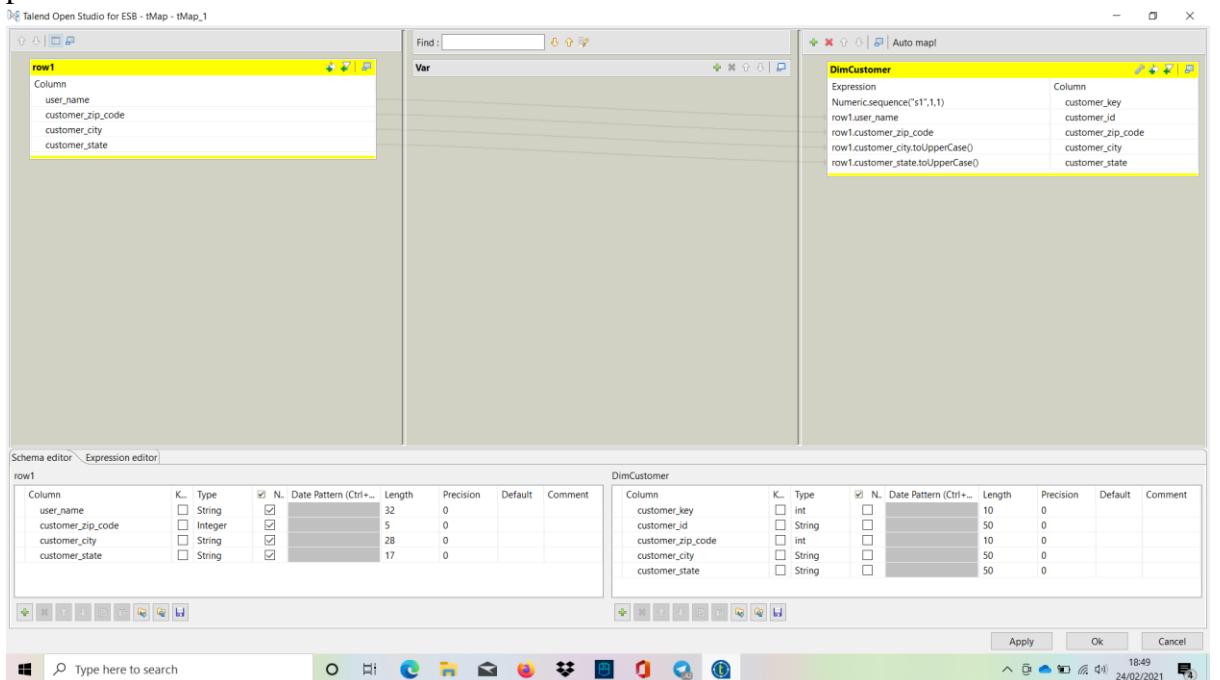
Kemudian untuk data yang duplikat, saya setting nullable.

The screenshot shows the Talend Open Studio interface for a tMap component named 'tMap_2'. It follows the same structure as the first screenshot, with 'row4' on the left, an empty 'Var' panel in the center, and 'user_dataset_duplicate' on the right. Below is a 'Schema editor' section with two tables: 'row4' and 'user_dataset_duplicate'. Both tables have the same four columns: 'user_name', 'customer_zip_code', 'customer_city', and 'customer_state'. The 'row4' table has 'Type' columns (String, Integer, String, String) and 'Length' values (32, 5, 28, 17). The 'user_dataset_duplicate' table has 'Type' columns (String, Integer, String, String) and 'Length' values (50, 10, 50, 50). The 'nullable' checkbox is checked for all columns in both tables. The bottom of the interface shows a Windows taskbar with icons and a system tray showing the date and time (24/02/2021, 18:42).

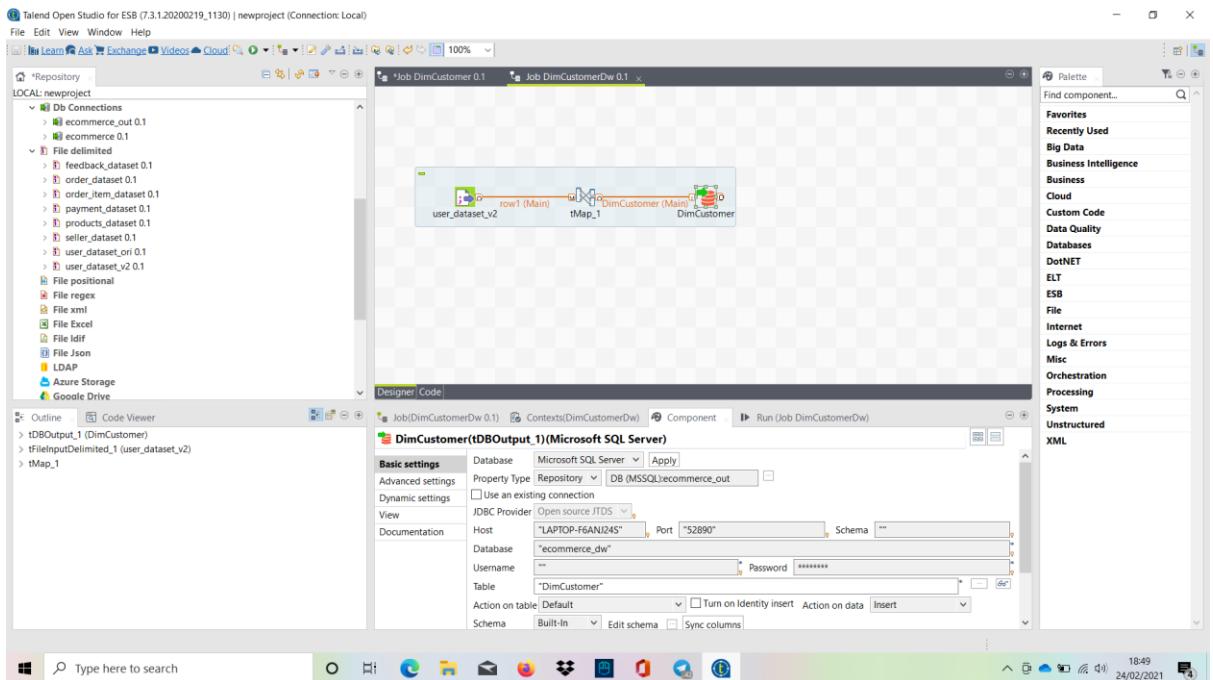
Kemudian untuk proses selanjutnya, saya akan langsung masukkan ke database SQL Server. Saya masukkan input berupa file delimiter, kemudian saya lakukan tmap.



Untuk tmap, saya memberikan surrogate key pada table DimCustomer dengan memberikan angka dari 1 hingga sebanyak jumlah record yang ada pada DimCustomer. Kemudian pada city dan state, saya membuat semua record dalam table tersebut menjadi upper case untuk mempermudah dalam pencarian, karena hanya memerhatikan penulisan huruf besar.

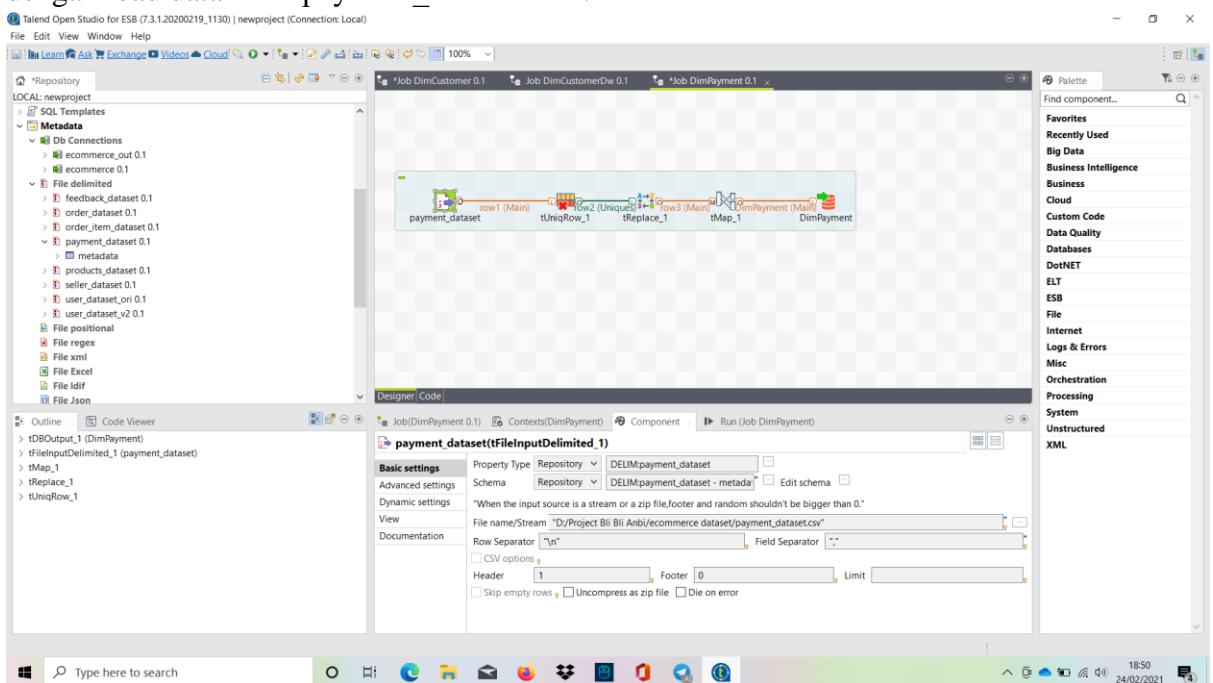


Kemudian hasilnya saya masukkan ke dalam database di SQL Server. Untuk pengumpulan saya akan ganti menjadi file input agar tidak error saat run job. Untuk saat ini saya hubungkan ke database langsung agar lebih cepat.

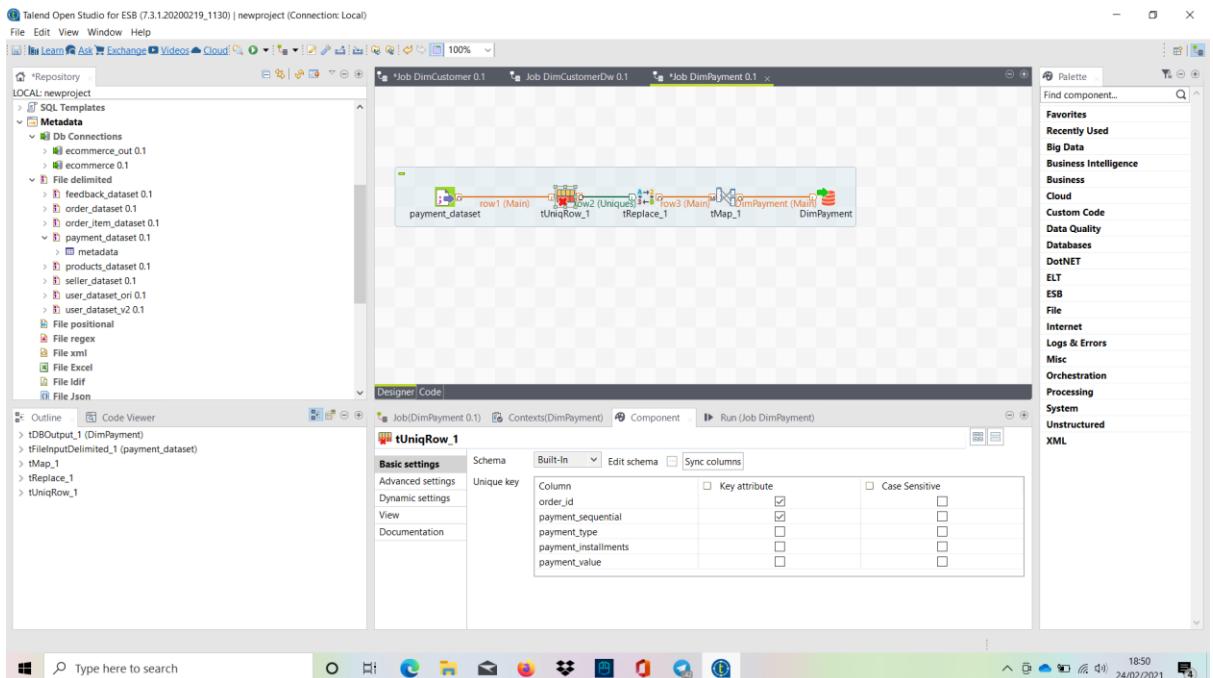


2. DimPayment

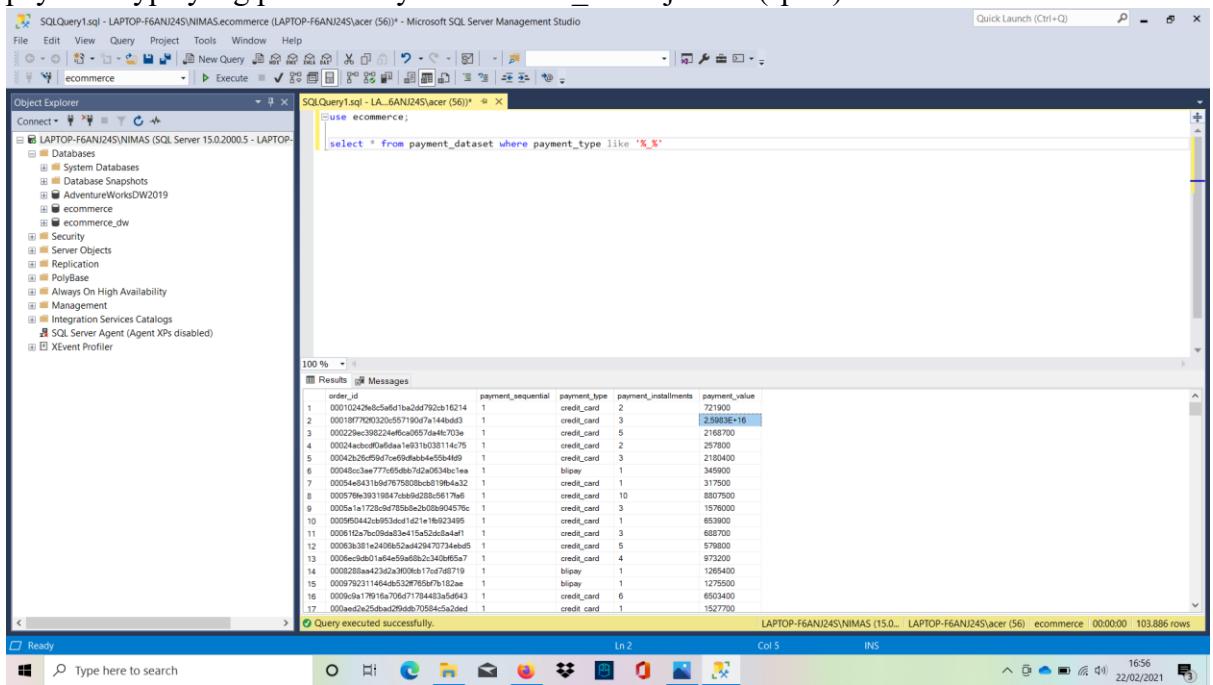
Untuk dimension berikutnya yaitu dimension payment. Caranya hampir sama yaitu dengan load data dari “payment_dataset.csv”.

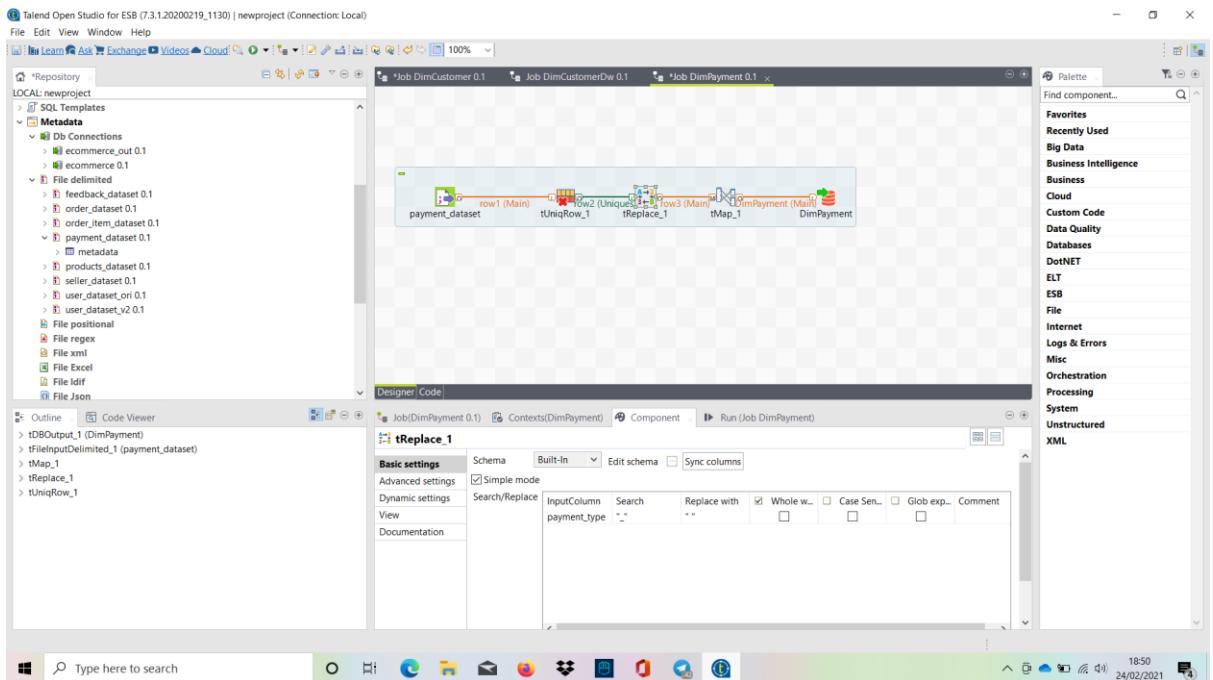


Kemudian masuk ke tUniqRow untuk memastikan bahwa baik order_id dan payment_sequential adalah unique, tanpa memperdulikan case sensitive.

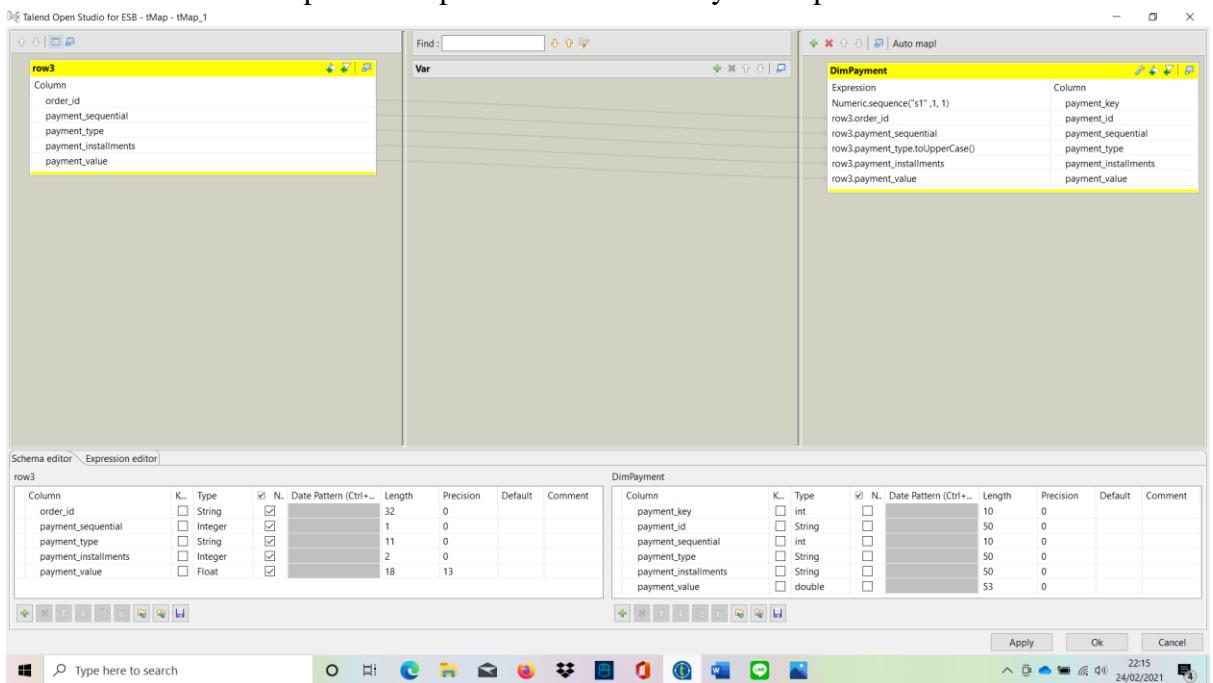


Kemudian selanjutnya masuk dalam tahap tReplace. Disini saya ingin mereplace payment type yang penulisannya memiliki “_” menjadi ““(spasi).

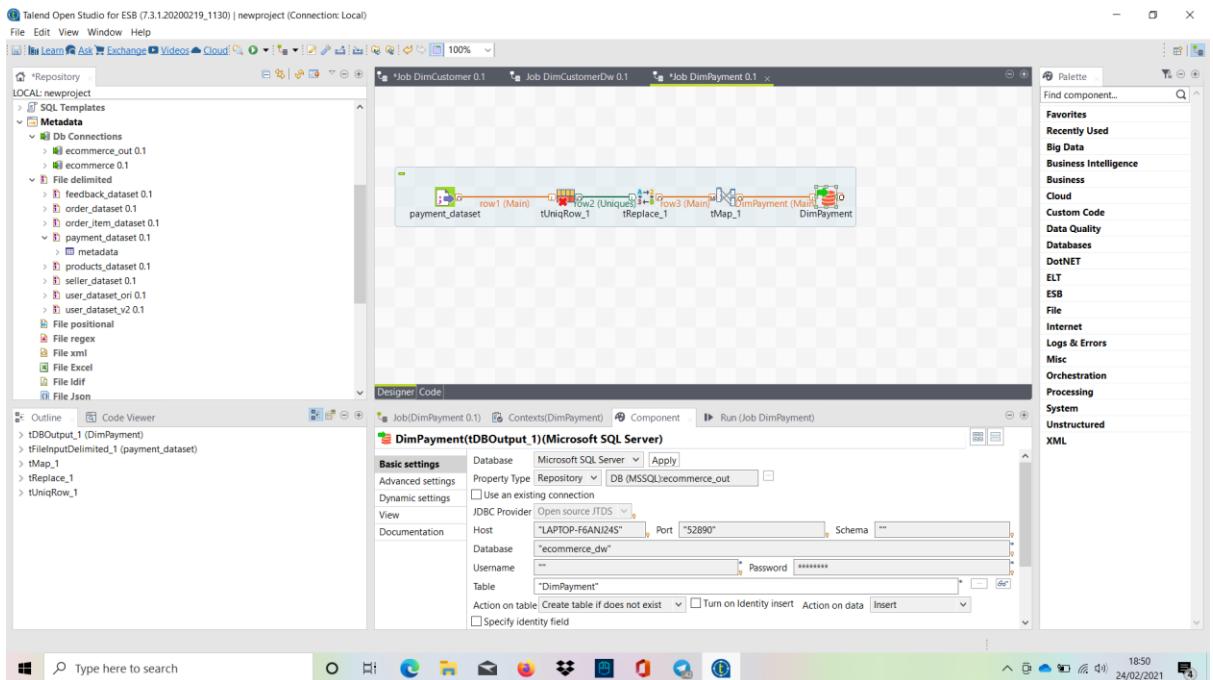




Kemudian di tmap saya membuat surrogate key seperti pada DimCustomer. Untuk tipe datanya saya sesuaikan dengan yang di csv. Kemudian untuk payment_type saya upper case semua untuk mempermudah pencarian karena hanya memperhatikan huruf besar.

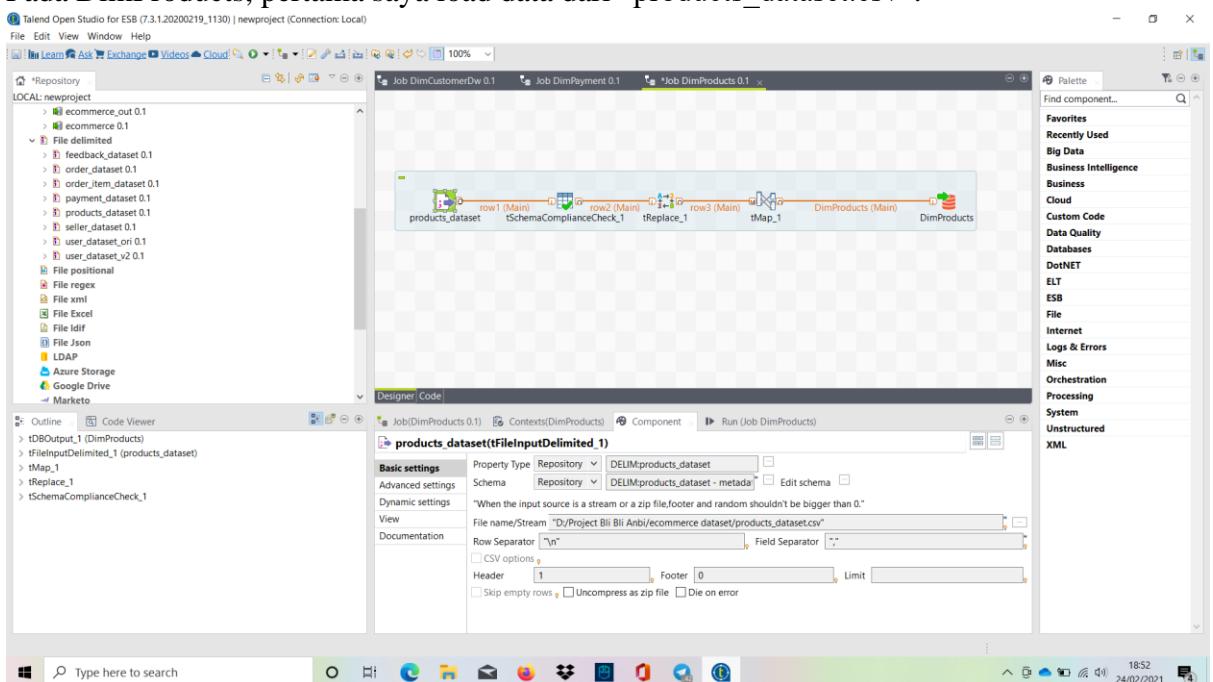


Yang terakhir adalah saya koneksi ke database SQL Server. Agar job bisa di run di semua tempat, untuk pengumpulan akan saya ubah jadi file delimited agar bisa di run di semua tempat. Namun disini saya akan tetap menggunakan koneksi sql server agar mempercepat pekerjaan.

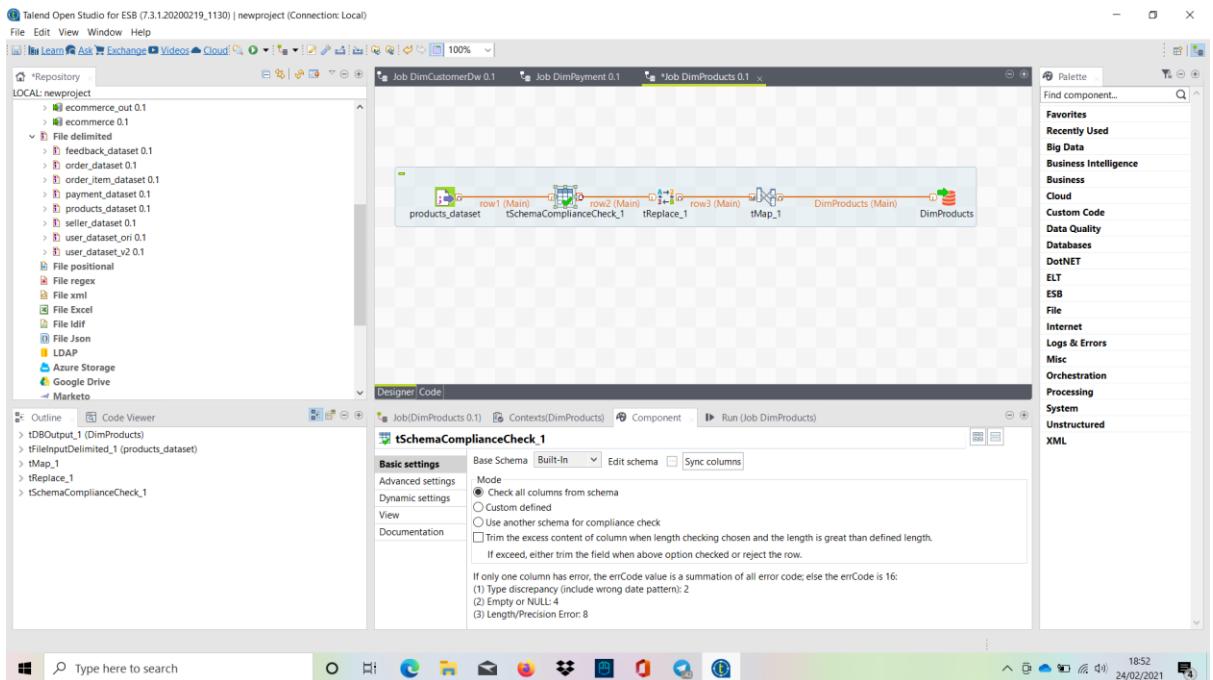


3. DimProducts

Pada DimProducts, pertama saya load data dari “products_dataset.csv”.



Kemudian saya melakukan tSchemaComplianceCheck, untuk mengecek kualitas data. Karena di products_dataset terdapat null value, jika tanpa menggunakan tools ini, terkadang bisa error Ketika job nya di Run.



Kemudian ada product_category, terdapat product yang penamaannya dipisahkan oleh “_”, bukan “ ” (spasi). Kemudian saya replace “_” dengan “ ”(spasi).

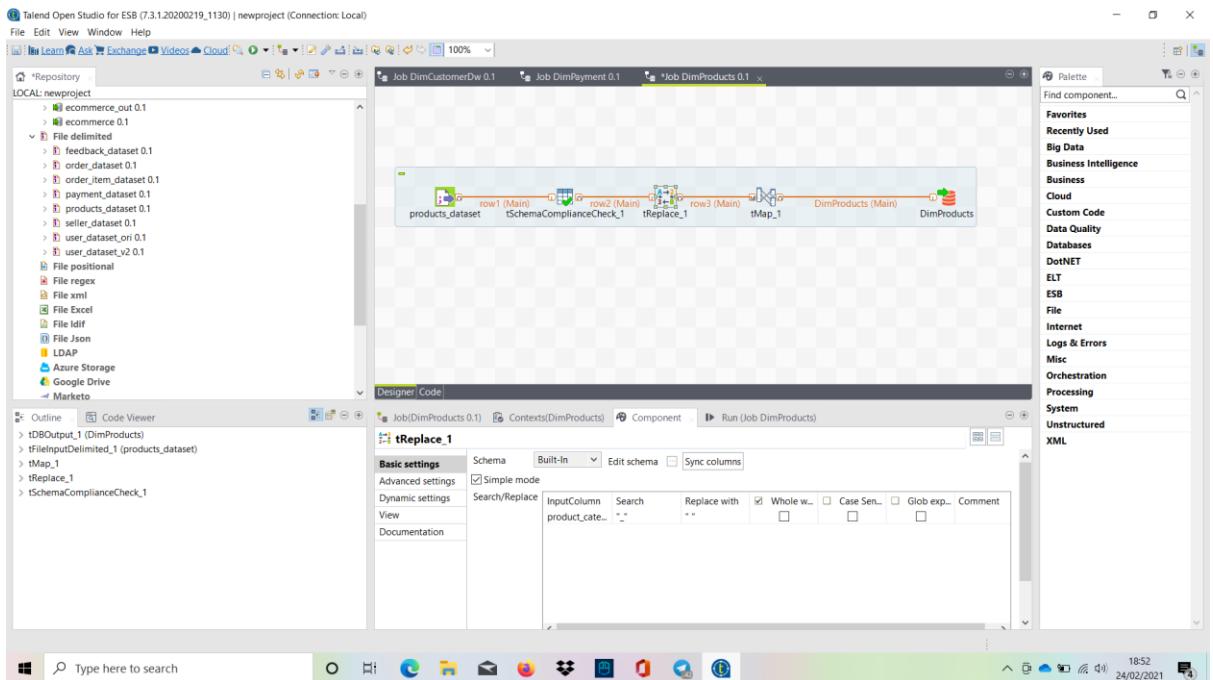
The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'ecommerce' database. The central pane displays the results of a query:

```
use ecommerce;
select * from products_dataset where product_category like '%_%'
```

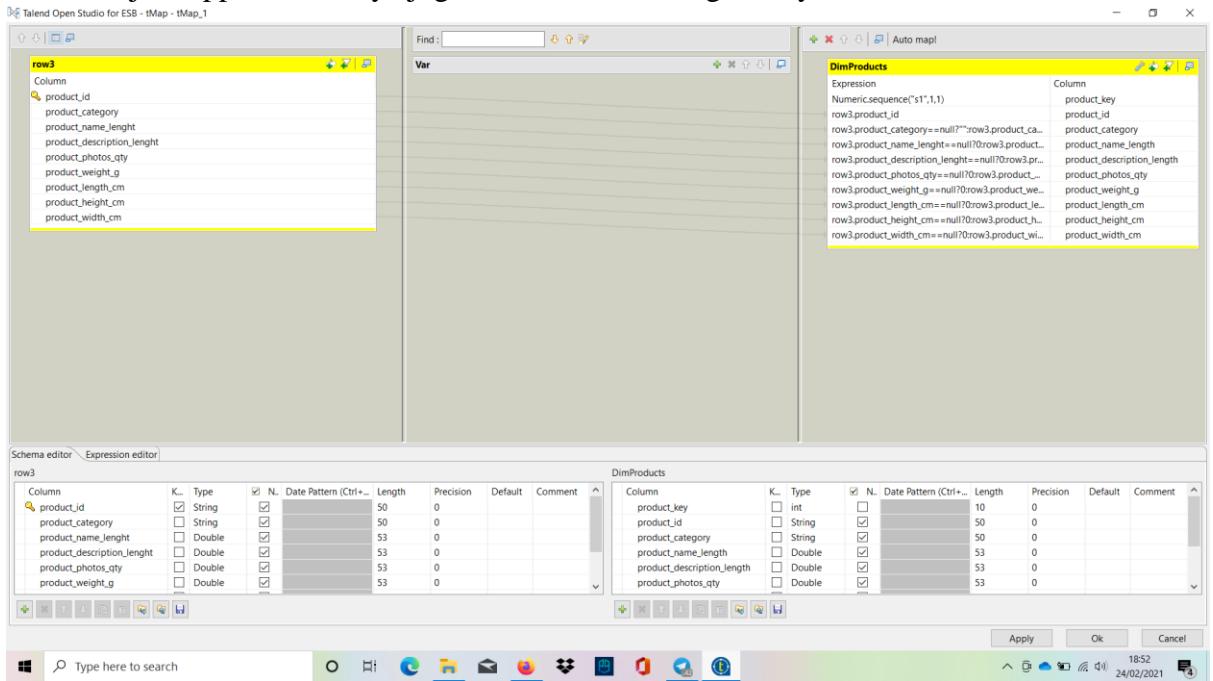
The results show a list of products with their category names containing underscores. The columns include product_id, product_category, product_name_length, product_description_length, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, and product_width_cm. The data is as follows:

product_id	product_category	product_name_length	product_description_length	product_photos_qty	product_weight_g	product_length_cm	product_height_cm	product_width_cm
1	perfume	530	60	300	200	150	150	260
2	perfume_decor	560	7820	40	1250	550	100	260
3	bed_bath_table	500	2660	20	3000	450	150	350
4	headphones	290	3640	30	5500	190	240	120
5	watches_gifts	480	6130	40	2500	220	110	150
6	auto	580	1770	10	1000	160	150	160
7	cool_stuff	420	24610	10	7000	250	50	150
8	consoles_games	530	2740	10	6000	300	200	200
9	bed_bath_table	420	2530	10	60000	400	40	300
10	furniture_decor	450	5200	30	6000	260	50	220
11	bed_bath_table	500	3810	10	10000	470	210	410
12	health_beauty	560	38030	60	2000	300	150	150
13	fashion_shoes	550	3440	30	18500	360	370	160
14	computer_accessories	580	9460	60	6500	250	100	150
15	housewares	530	6310	20	7500	300	50	350
16	cool_stuff	180	2630	10	3550	22	160	120
17	furniture_decor	540	17500	30	26000	1050	30	700

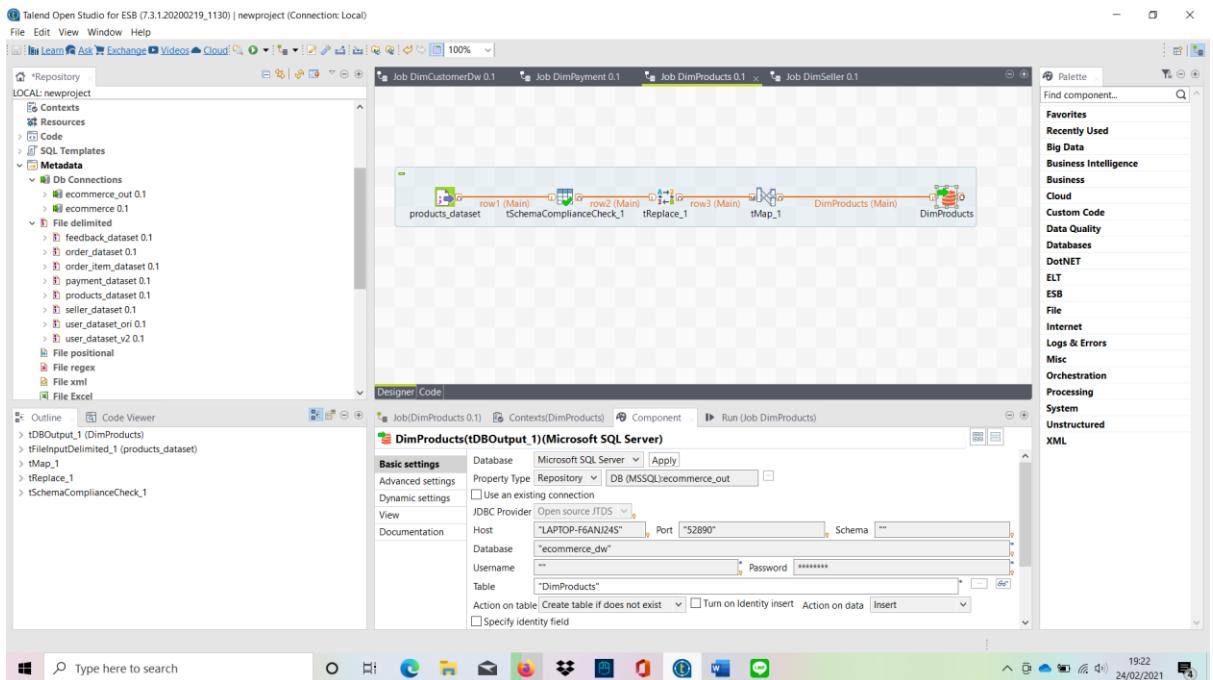
The status bar at the bottom indicates the query was executed successfully on 'LAPTOP-F6AN1Z4S\NIMAS (15.0...)' at '22/02/2021 9:29'.



Selanjutnya di tmap, saya awalnya menemukan banyak error di bagian row yang memiliki null value. Kemudian saya siasati dengan menangkap record yang null, kemudian jika string maka saya ubah nilainya menjadi “ ”(spasi), dan jika numeric maka saya ubah nilai null menjadi 0. Untuk product category, semua penamaan saya ubah menjadi upper case. Saya juga memberikan surrogate key untuk DimProducts.

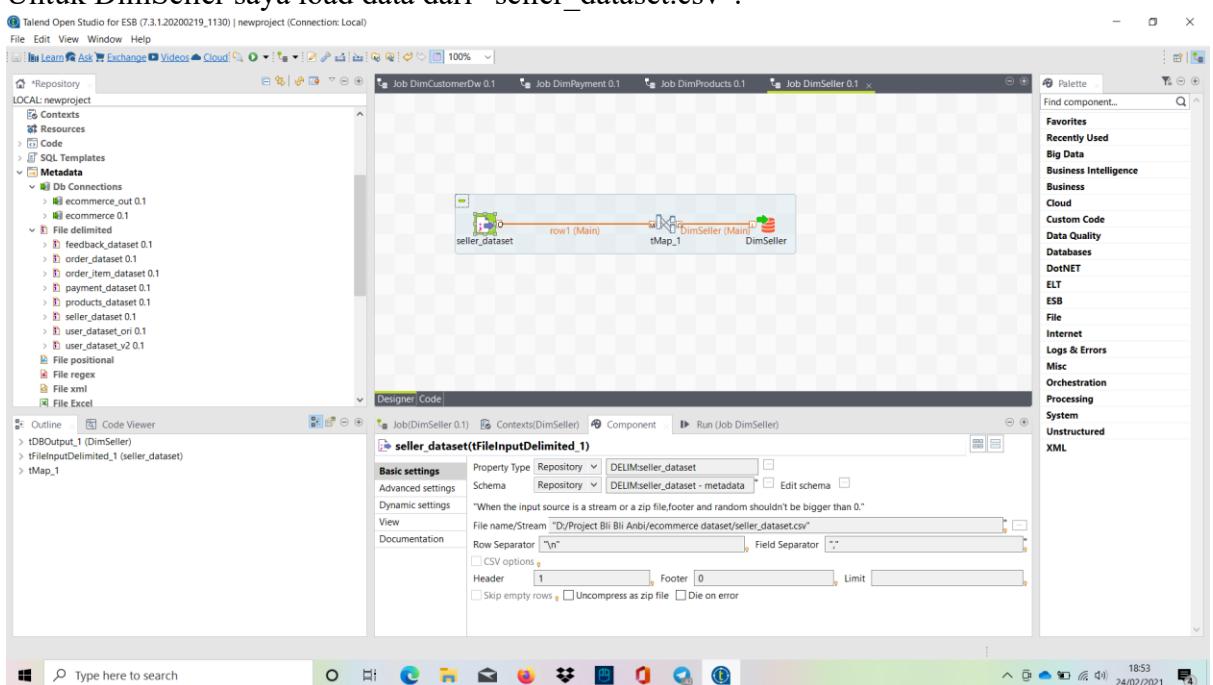


Untuk mempercepat pekerjaan, saya langsung sambung ke SQL Server. Namun untuk pengumpulan tugas saya akan ganti menjadi file delimited agar bisa dengan mudah di run.



4. DimSeller

Untuk DimSeller saya load data dari “seller_dataset.csv”.



Selanjutnya saya petakan menggunakan tmap. Saya menambahkan surrogate key pada DimSeller, kemudian melakukan upper case pada city dan state.

Column	K_...	Type	N...	Date Pattern (Ctrl+...)	Length	Precision	Default	Comment
seller_id	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		32	0		
seller_zip_code	<input type="checkbox"/>	Integer	<input checked="" type="checkbox"/>		5	0		
seller_city	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		28	0		
seller_state	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		18	0		

Column	K_...	Type	N...	Date Pattern (Ctrl+...)	Length	Precision	Default	Comment
seller_key	<input type="checkbox"/>	int	<input checked="" type="checkbox"/>		50	0		
seller_id	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		10	0		
seller_zip_code	<input type="checkbox"/>	int	<input checked="" type="checkbox"/>		50	0		
seller_city	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50	0		
seller_state	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		50	0		

Selanjutnya saya langsung transfer ke SQL Server. Untuk mempermudah pengoreksian, di akhir saya akan merubah outputnya menjadi file delimited, sehingga tidak perlu melakukan koneksi ke database.