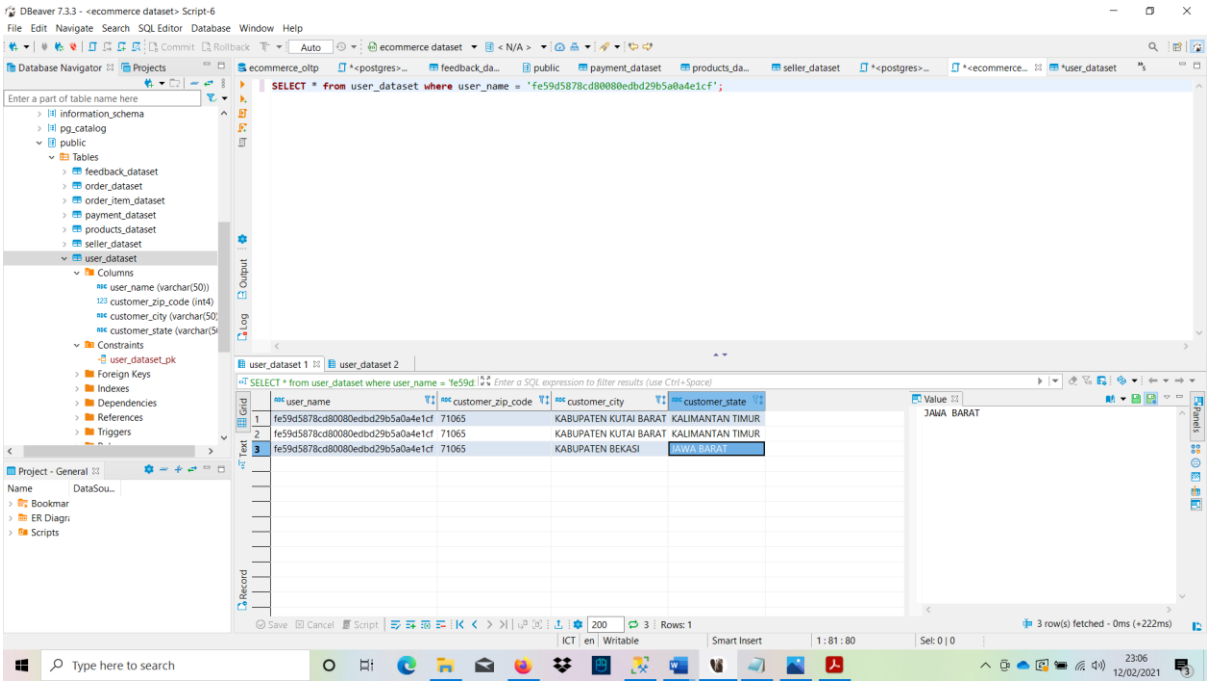
PROGRESS ETL

1. Cleaning user dataset

   Sebelumnya saya menemukan suatu hal yang agak ganjil pada user dataset.



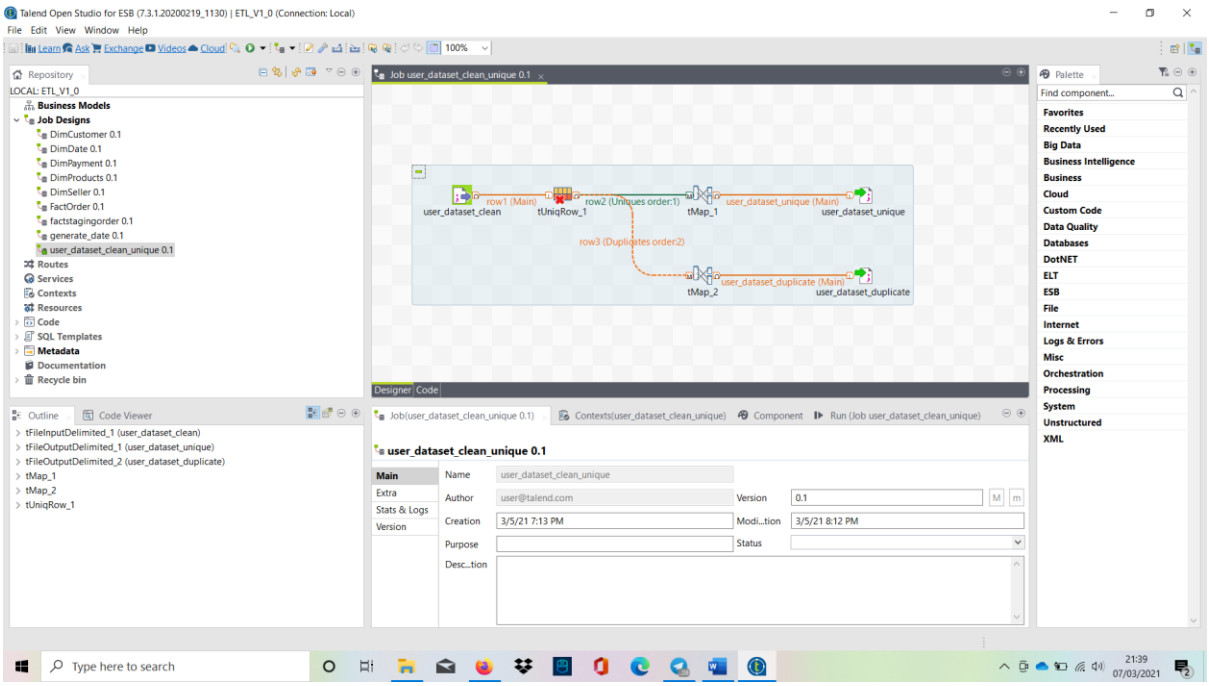
   Dimana 1 zip code, memiliki 2 wilayah yang sangat jauh berbeda. Tentu ini akan berpengaruh Ketika akan memisahkan antara data yang duplicate dengan unique. Bisa jadi nantinya tanpa sengaja akan terpilih data yang kurang tepat. Maka dari itu dilakukan cleansing terhadap data dimana satu zip code, spesifik terletak pada satu wilayah tertentu. Jika terdapat zip_code yang memiliki wilayah yang berbeda, saya pilih wilayah yang paling banyak. Missal terdapat zip_code dengan kode 71065. Terdapat 6 daerah yang menggunakan zip_code tersebut, 5 ada di Bekasi, Jawa Barat, 1 nya ada di Surabaya, Jawa Timur. Maka saya akan memilih Bekasi, Jawa Barat. Query untuk melakukan cleansing terdapat pada "**05_03_2021_cleaning query for user_dataset**". Disana ditemukan pula penamaan kota yang kurang valid seperti "KOTA B A T A M", yang seharusnya adalah "KOTA BATAM". Maka dari itu cleansing juga meliputi penyempurnaan nama kota.

2. Staging Data

   Untuk staging data dilakukan pada 2 aplikasi yang berbeda, yaitu SQL Server dan Talend. Untuk staging saya menggunakan SQL Server karena di laptop saya, run dan load datanya lebih cepat.

   a. user_dataset

      Diketahui bahwa user_dataset masih terdapat user_name yang duplicate, meskipun seharusnya itu unique. Maka dari itu, butuh bantuan dari aplikasi Talend untuk memisahkan antara user_name yang duplicate dengan yang unique. Berangkat dari dataset yang sudah dibersihkan sebelumnya, dengan menggunakan tUniqrow, saya memisahkan kedua data tersebut. Dengan cara seperti ini maka user_dataset baik yang duplicate maupun unique, tidak ada yang terhapus.



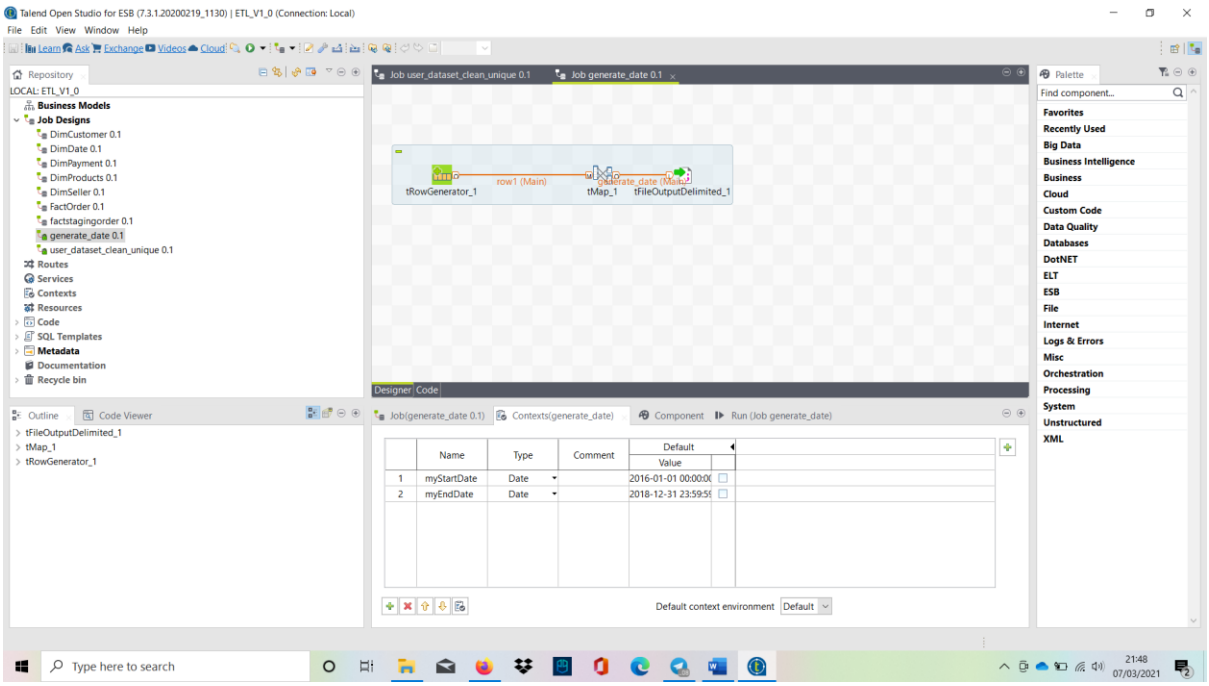
b. payment_dataset
Ketika saya melakukan pengecekan menggunakan SQL, saya menemukan bahwa kombinasi order_id dan payment_sequential adalah unique, sehingga tidak perlu memerlukan tools tambahan. Namun Ketika akan memasuki data warehouse, saya masih menggunakan tUniqrow untuk memastikan kualitas data.

c. products_dataset

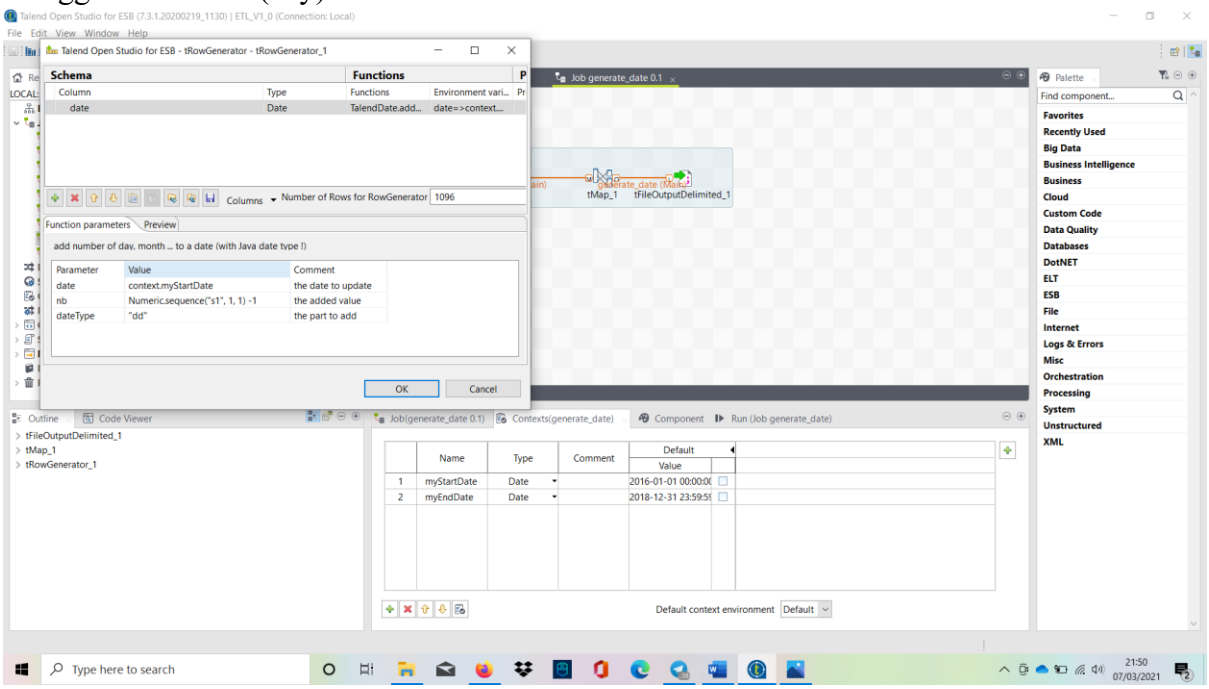
d. Ketika saya melakukan pengecekan menggunakan SQL, saya menemukan bahwa product_id adalah unique, sehingga tidak perlu memerlukan tools tambahan. Namun Ketika akan memasuki data warehouse, saya masih menggunakan tUniqrow untuk memastikan kualitas data.

e. seller_dataset
Ketika saya melakukan pengecekan menggunakan SQL, saya menemukan bahwa seller_id adalah unique, sehingga tidak perlu memerlukan tools tambahan. Namun Ketika akan memasuki data warehouse, saya masih menggunakan tUniqrow untuk memastikan kualitas data.

f. generate_date
Untuk membuat DimDate, maka perlu menggenerate data, maka saya memerlukan tools tambahan yaitu Talend. Diketahui bahwa order paling awal ada pada tahun 2016 dan paling akhir adalah 2018. Maka dari itu saya menentukan context myStartDate atau tanggal awal adalah 1 Januari 2016 dan myEndDate atau tanggal akhir pada 31 Desember 2018.

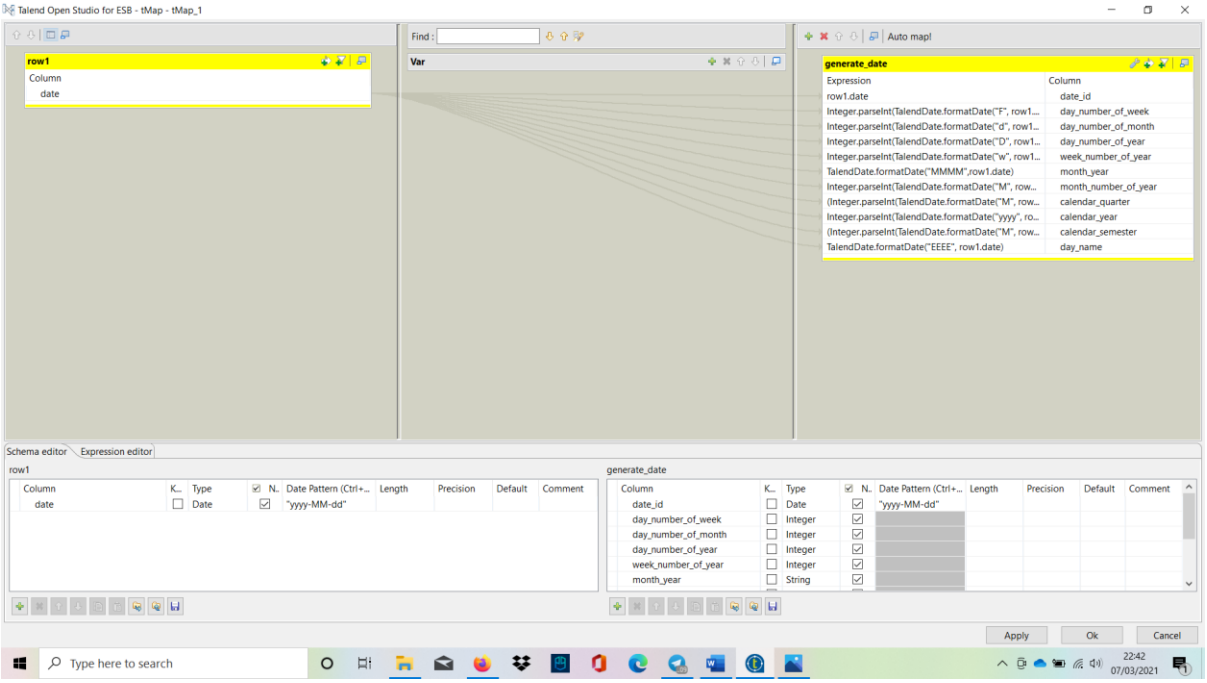


Kemudian terdapat pengaturan pada tRowGenerator. Yang pertama adalah di parameter date, menggunakan context.myStartDate yang berarti dimulai dari tanggal yang ada pada myStartDate yang terdapat pada context. Kemudian untuk urutannya menggunakan Numeric.sequence("s1",1,1) -1. -1 ditambahkan di belakang agar dimulai dari 1 Januari 2016. Kemudian untuk dateType menggunakan "dd" (day).



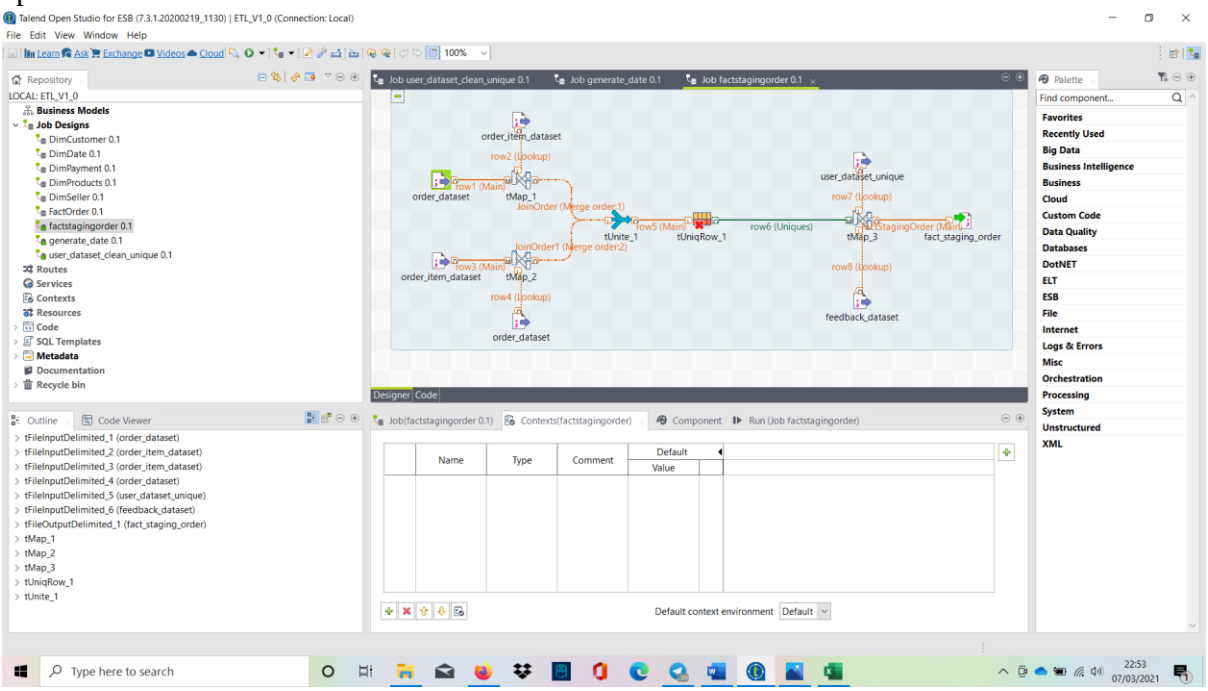
Kemudian saya membuat berbagai macam kolom. Yang pertama ada day_number_of_week, yang merupakan minggu ke berapa pada tanggal tersebut (Week 1-4). Kemudian ada day_number_of_month, yang merupakan hari ke berapa pada suatu tanggal di bulan tersebut(1-31). Kemudian ada day_number_of_year, yang merupakan hari ke berapa pada suatu tanggal di tahun tersebut(1-366). Month_year merupakan bulan dalam bentuk string tanggal tersebut(January –

December). Terdapat month_number_of_year, yaitu bulan ke berapa pada tanggal tersebut (1-12). Calendar_quarter merupakan kuarter ke berapa tanggal tersebut (1-4). Calendar_year merupakan tahun berapa pada tanggal tersebut (2016-2018). Calendar_semester merupakan semester ke berapa pada tanggal tersebut (1-2). Day name merupakan nama hari pada tanggal tersebut (Monday-Sunday).

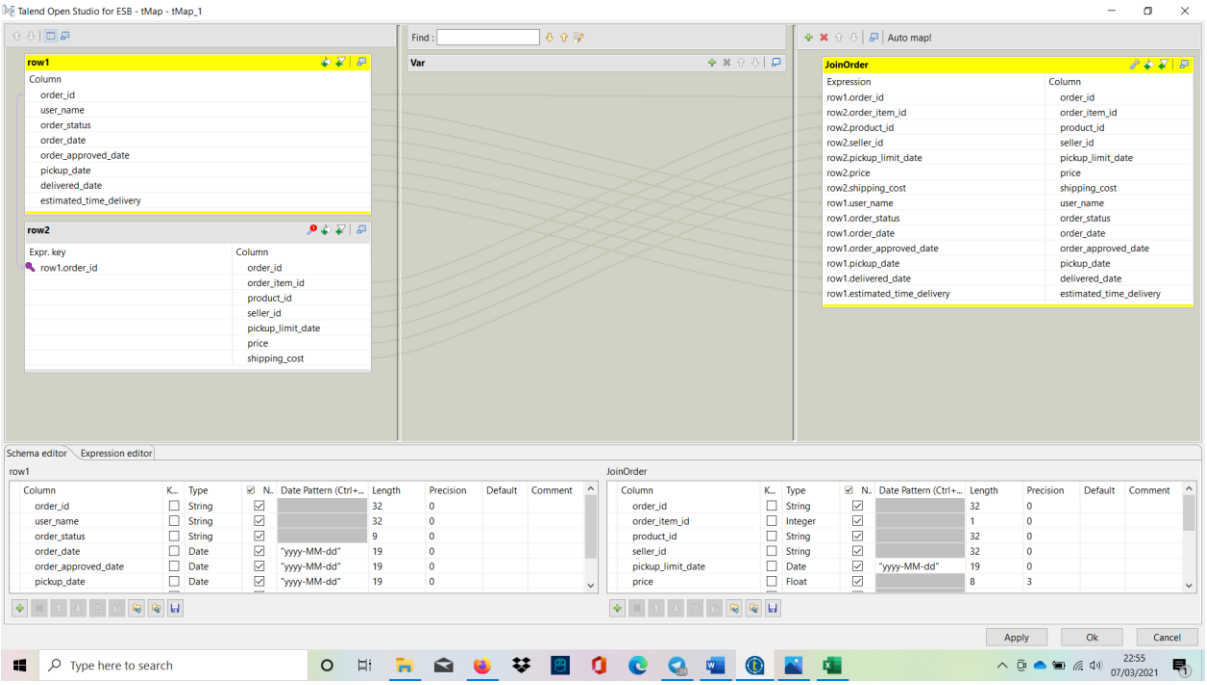


g. fact_staging_order

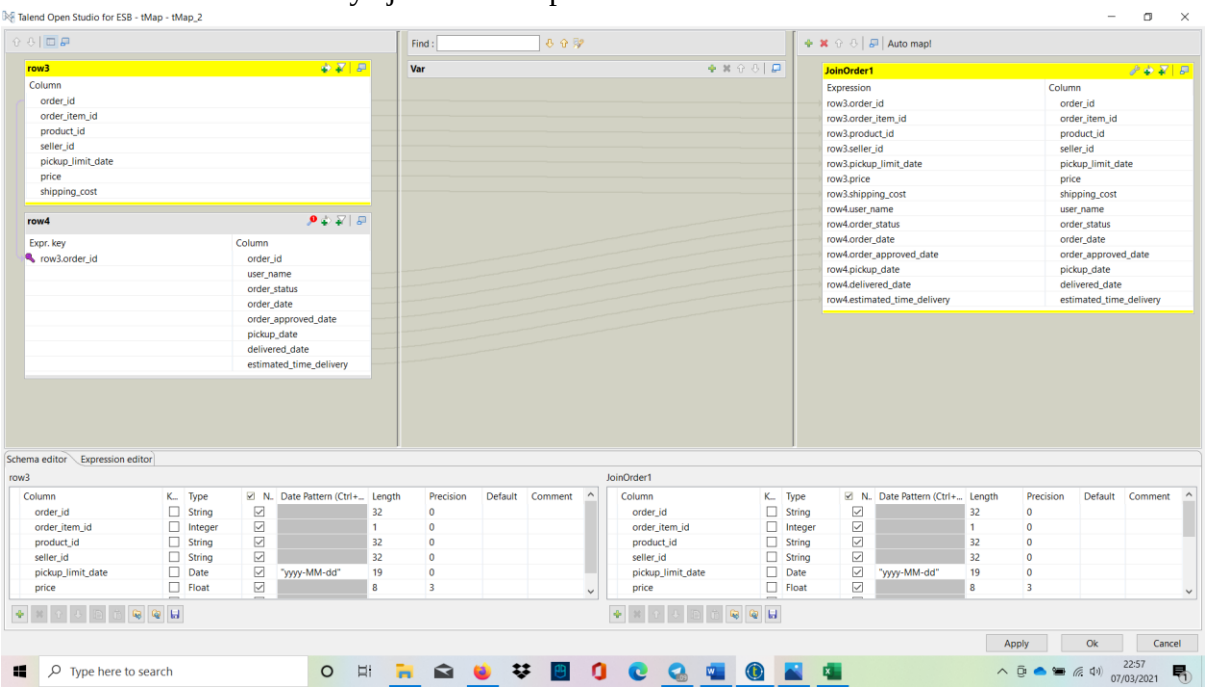
Untuk fact_staging_order, saya melakukan outer join pada order_dataset dan order_item_dataset. Hal ini saya lakukan agar tidak ada data yang terhapus karena menurut saya atribut yang terdapat pada kedua dataset tersebut sangat penting. Untuk mempermudah pembuatan fact Order, maka saya membutuhkan bantuan dari aplikasi Talend.



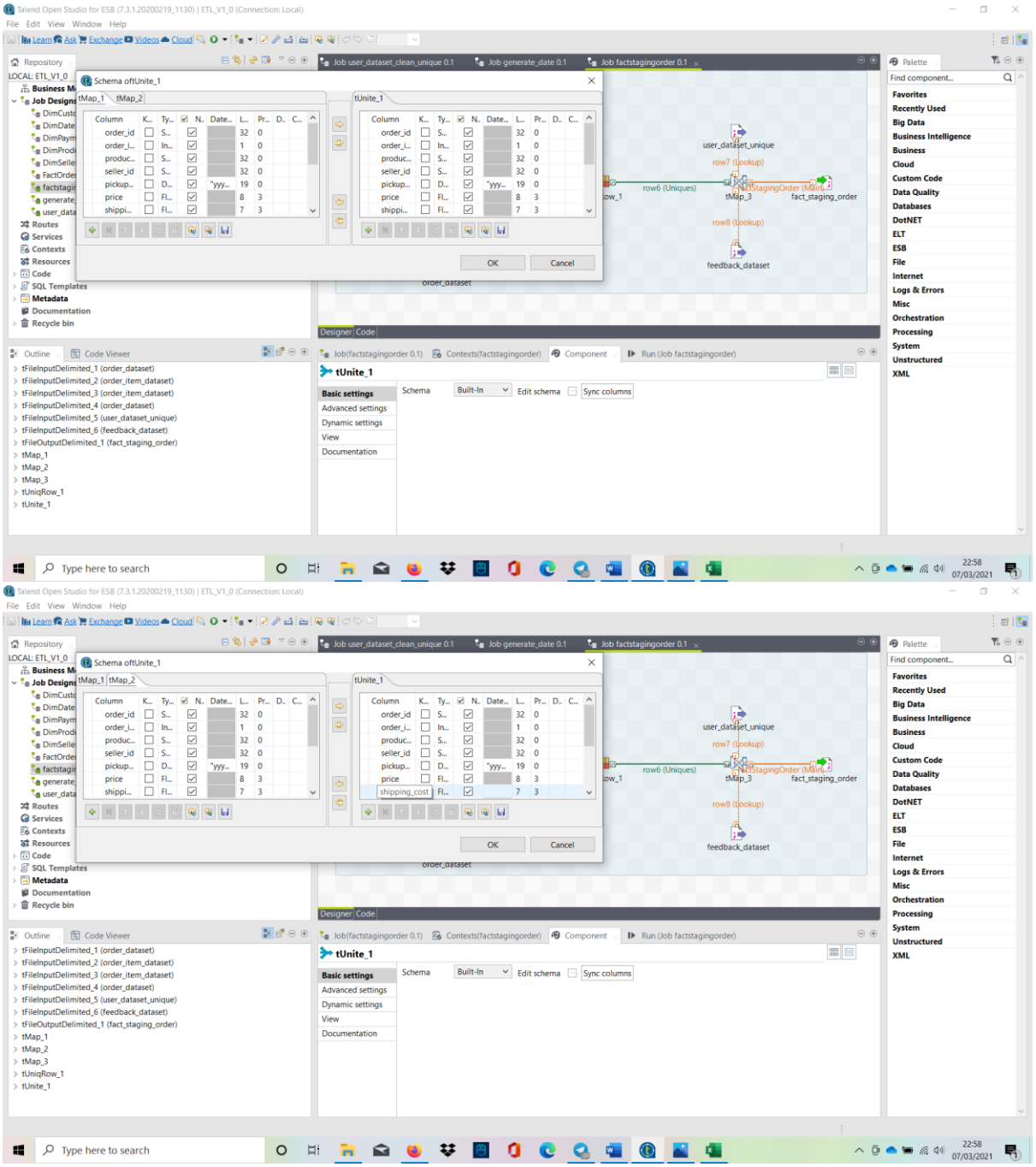
Pada tMap pertama, saya menggunakan order_dataset sebagai main dan order_item_dataset sebagai lookupnya. Dari order_id yang ada pada order_dataset, saya hubungkan dengan order_id yang ada pada order_item_dataset. Kemudian atribut dari kedua table tersebut saya jadikan satu pada JoinOrder.



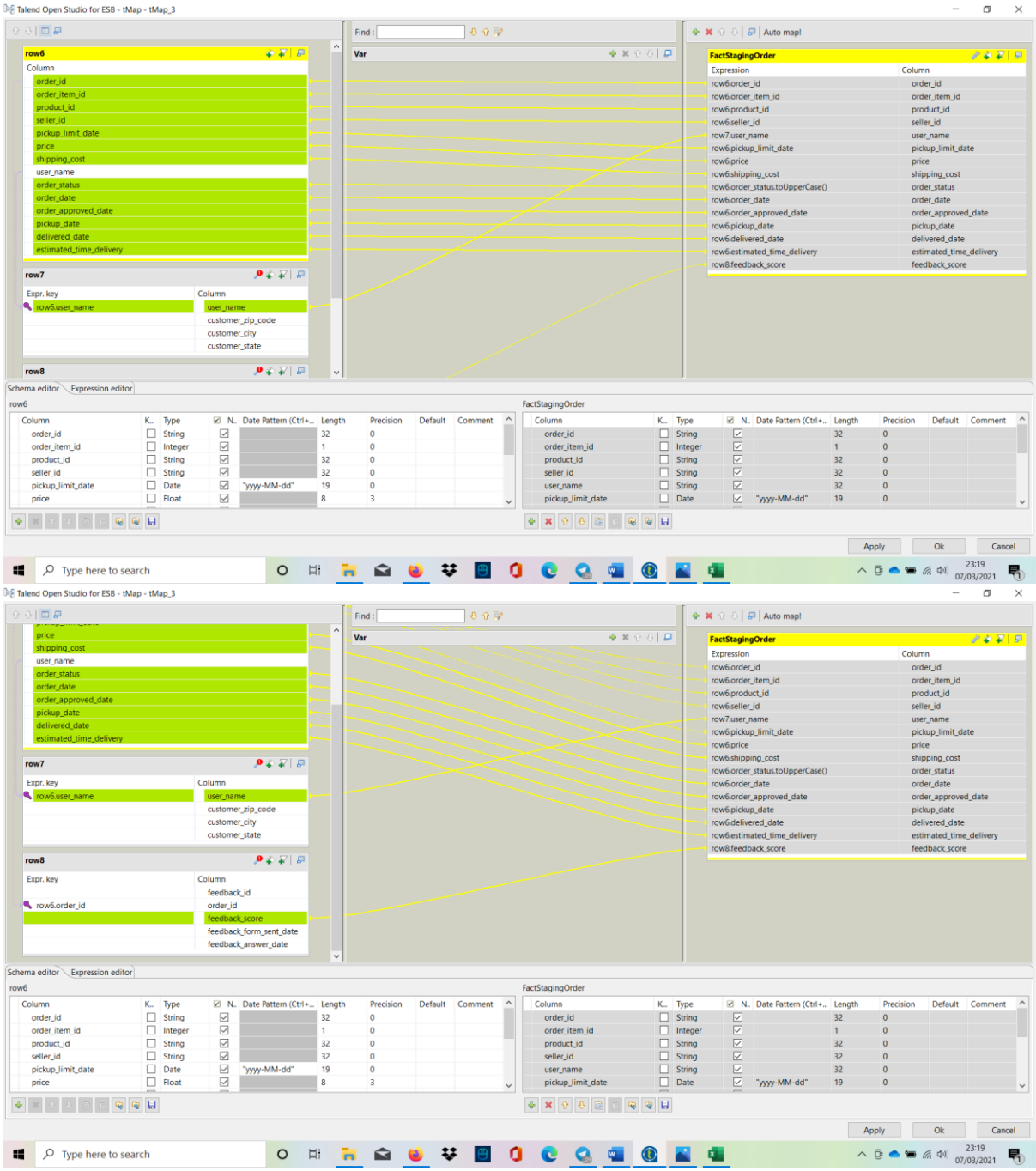
Pada tMap kedua, saya menggunakan order_item_dataset sebagai main dan order_dataset sebagai lookupnya. Dari order_id yang ada pada order_item_dataset, saya hubungkan dengan order_id yang ada pada order_dataset. Kemudian atribut dari kedua table tersebut saya jadikan satu pada JoinOrder1.



Kemudian saya menggunakan tUnite. tUnite digunakan untuk melakukan outer join pada kedua map. Disini harus dipastikan bahwa input dan output sudah sesuai dengan yang diinginkan.



Kemudian saya menggunakan tUniqRow untuk memastikan bahwa order_id dan order_item_id adalah unique, kemudian dipetakan ke dalam tmap. Kemudian saya petakan beberapa hal seperti user_dataset_clean, dimana user_name dalam user_dataset_unique terhubung dengan order yang sudah di join, kemudian ambil user_name dari user_dataset_unique. Hal ini juga berlaku pada feedback_score dimana order_id pada feedback_score dihubungkan dengan order yang sudah di join. Kemudian hanya ambil feedback_score dari feedback_dataset.


Talend Open Studio for ESB - tMap - tMap_3
row6
Column
order_id
order_item_id
product_id
seller_id
pickup_limit_date
price
shipping_cost
user_name
order_status
order_date
order_approved_date
pickup_date
delivered_date
estimated_time_delivery
row7
Expr. key
Column
row6.user_name
user_name
customer_zip_code
customer_city
customer_state
row8
Find :
Var
Auto map!
FactStagingOrder
Expression
Column
row6.order_id
row6.order_item_id
row6.product_id
row6.seller_id
row7.user_name
row6.pickup_limit_date
row6.price
row6.shipping_cost
row6.order_status.toUpperCase()
row6.order_date
row6.order_approved_date
row6.pickup_date
row6.delivered_date
row6.estimated_time_delivery
row8.feedback_score
Schema editor
Expression editor
row6
FactStagingOrder
Column
K...
Type
N...
Date Pattern (Ctrl+...
Length
Precision
Default
Comment
String
Integer
Date
Float
"yyyy-MM-dd"
Apply
Ok
Cancel
Type here to search
23:19
07/03/2021

Talend Open Studio for ESB - tMap - tMap_3
price
shipping_cost
user_name
order_status
order_date
order_approved_date
pickup_date
delivered_date
estimated_time_delivery
row7
Expr. key
Column
row6.user_name
user_name
customer_zip_code
customer_city
customer_state
row8
Expr. key
Column
feedback_id
row6.order_id
order_id
feedback_score
feedback_form_sent_date
feedback_answer_date
Find :
Var
Auto map!
FactStagingOrder
Expression
Column
row6.order_id
row6.order_item_id
row6.product_id
row6.seller_id
row7.user_name
row6.pickup_limit_date
row6.price
row6.shipping_cost
row6.order_status.toUpperCase()
row6.order_date
row6.order_approved_date
row6.pickup_date
row6.delivered_date
row6.estimated_time_delivery
row8.feedback_score
Schema editor
Expression editor
row6
FactStagingOrder
Column
K...
Type
N...
Date Pattern (Ctrl+...
Length
Precision
Default
Comment
String
Integer
Date
Float
"yyyy-MM-dd"
Apply
Ok
Cancel
Type here to search
23:19
07/03/2021