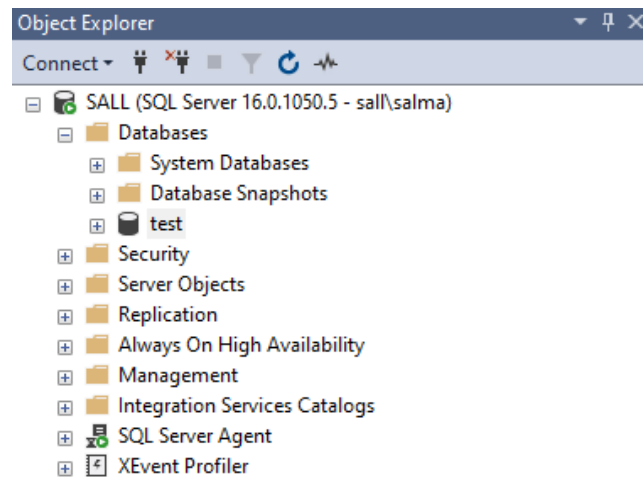
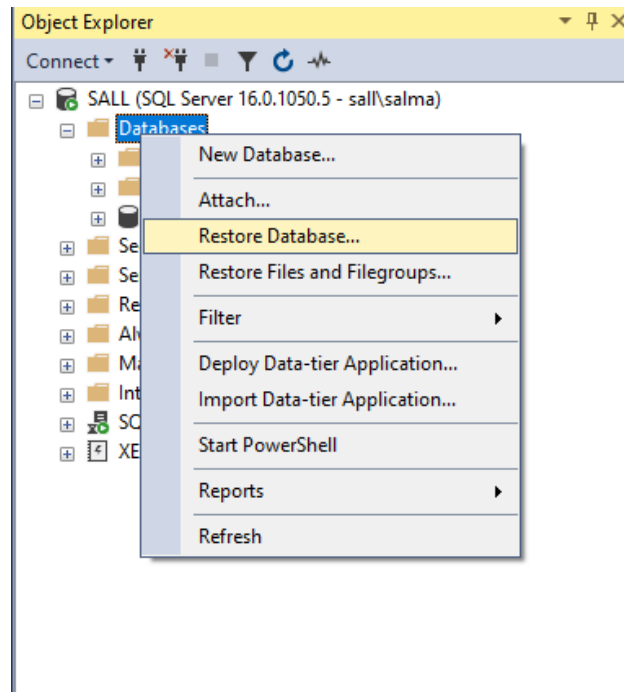


1. Melakukan Import/Restore Database Staging.

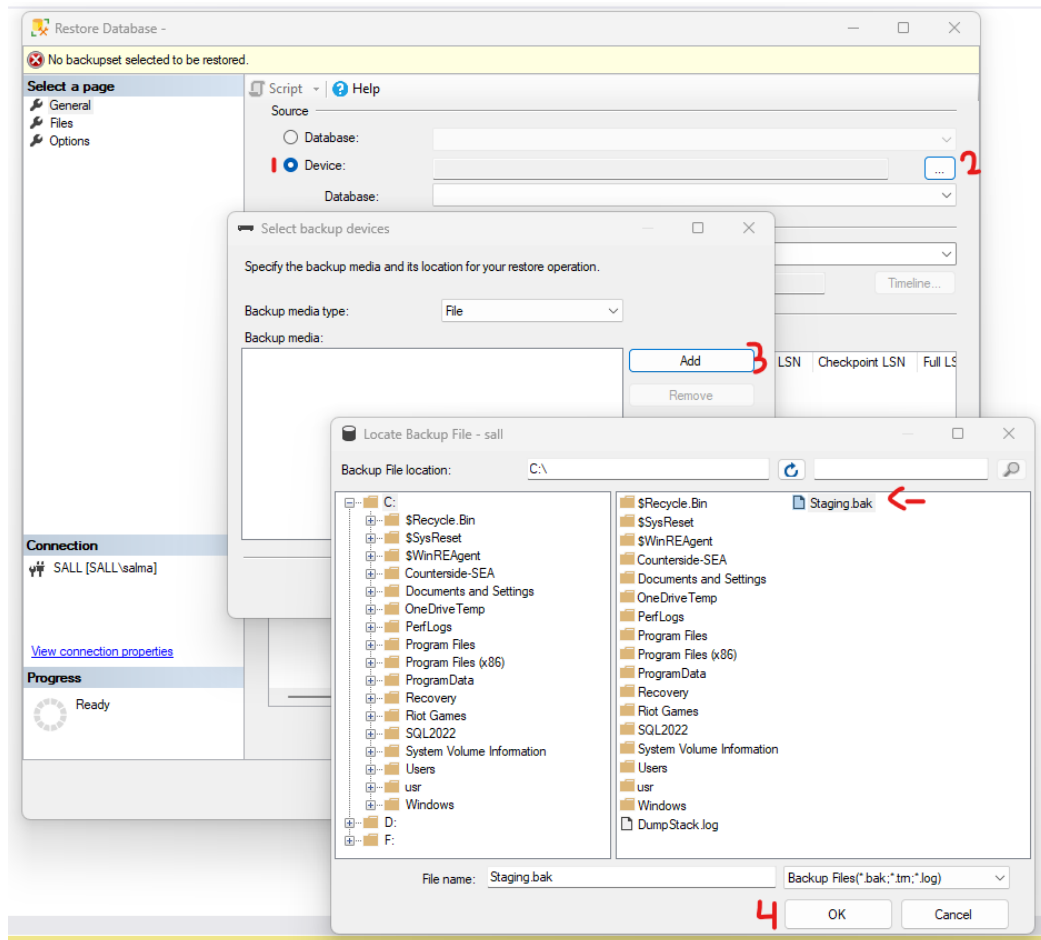
pada saat membuka SSMS pada databases tidak ada database Staging, untuk menambahkan database staging perlu dilakukan restore database yang dimana file tersebut sudah tersedia pada challenge ini.



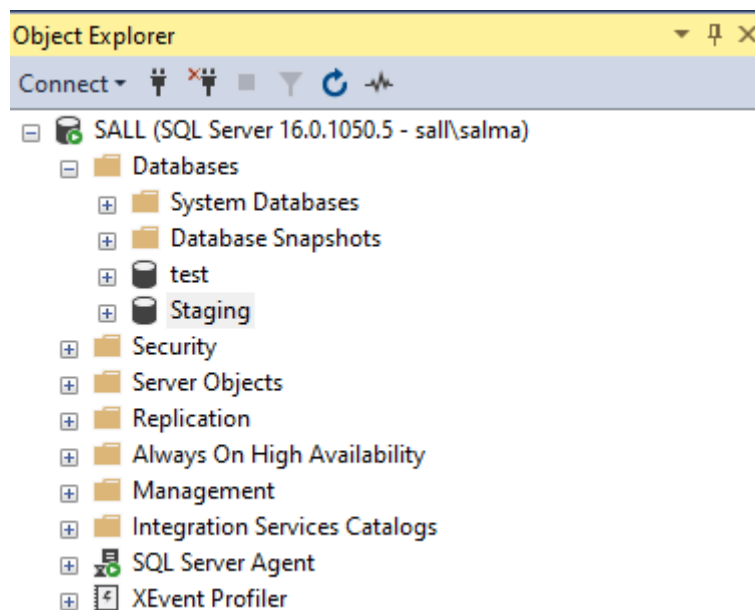
- Untuk restore database saya melakukannya dengan cara click kanan pada folder databases dan memilih restores database



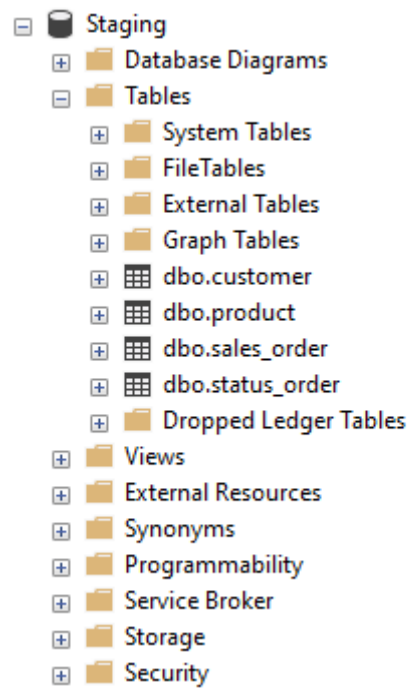
- Kemudian akan muncul tampilan seperti dibawah ini yang pertama dilakukan adalah klik Device seperti nomor 1 pada gambar, lalu klik button ... seperti yang terlihat pada nomor 2 di gambar setelah itu akan muncul tampilan Select Backup Device dan klik tombol Add seperti pada nomor 3 di gambar ini, setelah itu pilih Staging.bak dan tekan OK



- Setelah klik OK maka akan muncul database yang direstore tadi dengan nama Staging

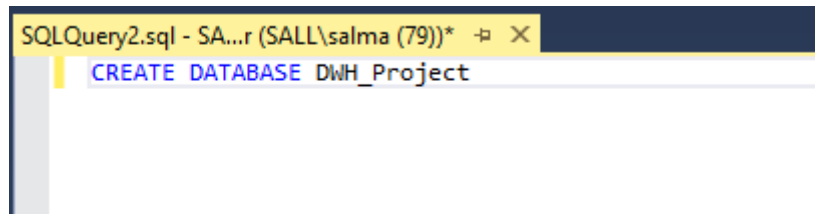


- Berikut adalah isi dari dataabse Staging, databases ini memiliki 4 table yaitu customer, product, sales_order dan status_order

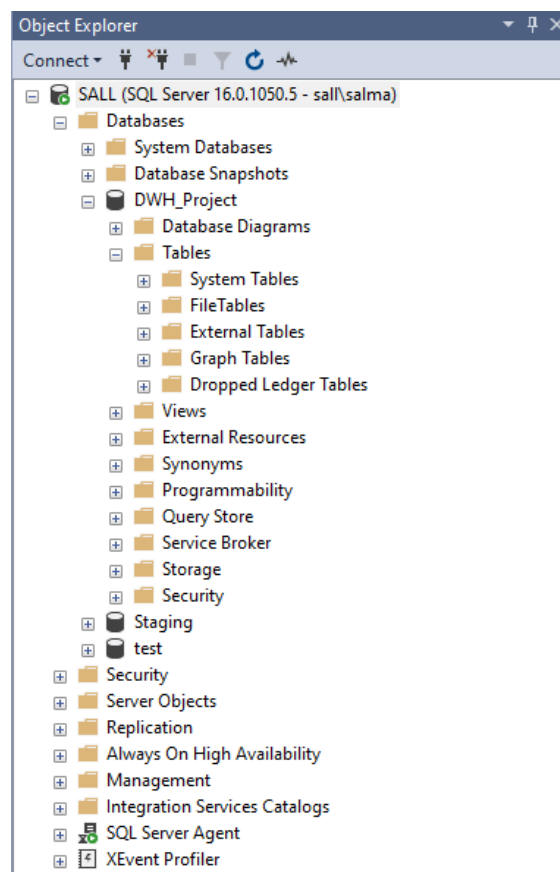


2. Membuat sebuah Database bernama DWH_Project, serta membuat Tabel Fact dan Dimension dari tabel yang ada di database Staging.

- Langkah pertama adalah membuat query untuk membuat database dengan nama "DWH_Project"



- Databases DWH_Project telah dibuat terlihat pada gambar dibawah ini, namun database tersebut belum memiliki table dan Langkah selanjutnya saya akan membuat table untuk database DWH_Project ini.



Pada bagian ini saya membuat 4 table dengan nama table DimCustomer, DimProduct, DimStatusOrder dan table FactSalesOrder.

- Gambar dibawah adalah query untuk membuat table dengan nama DimCustomer tabel ini memiliki 6 attribute dengan 2 attribute bertipe data integer dan 4 attribute bertipe data varchar dengan masing masing maksimal panjang character berjumlah 50 dan seluruh attribute tidak boleh null atau tidak boleh kosong, pada baris terakhir attribute CustomerID di set menjadi primary key.

```
CREATE TABLE DimCustomer(  
    CustomerID int NOT NULL,  
    CustomerName varchar(50) NOT NULL,  
    Age int NOT NULL,  
    Gender varchar(50) NOT NULL,  
    City varchar(50) NOT NULL,  
    NoHP varchar(50) NOT NULL,  
    CONSTRAINT PK_Customer PRIMARY KEY (CustomerID)  
)
```

- Selanjutnya membuat table dengan nama DimStatusOrder, table ini memiliki 4 attribute dengan 2 attribute bertipe data integer dan 2 attribute lainnya bertipe data varchar dimana attribute bertipe data varchar memiliki Panjang maksimal 50 karakter, semua attribute ini tidak boleh null atau tidak boleh kosong, pada baris terakhir attribute StatusID di set menjadi primary key.

```
CREATE TABLE DimStatusOrder(  
    StatusID int NOT NULL,  
    StatusOrder varchar(50) NOT NULL,  
    StatusOrderDesc varchar(50) NOT NULL,  
    CONSTRAINT PK_StatusOrder PRIMARY KEY (StatusID)  
)
```

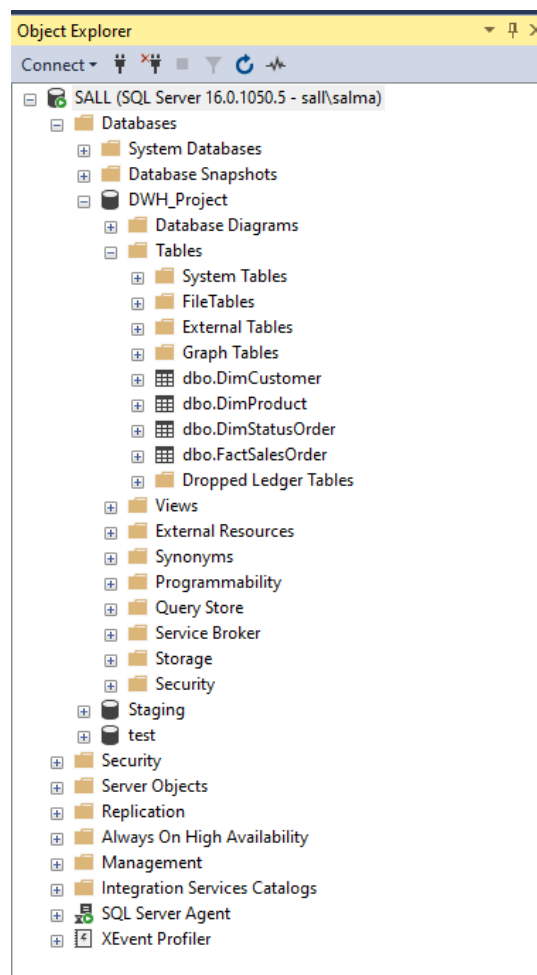
- Kemudian membuat tabel dengan nama DimProduct, tabel ini memiliki 4 attribut dengan 2 attribute bertipe data integer dan 2 attribute bertipe data varchar dengan panjang maksimal 255, semua attribute ini tidak boleh null atau kosong, pada baris terakhir attribute ProductID di set menjadi primary key.

```
CREATE TABLE DimProduct(  
    ProductID int NOT NULL,  
    ProductName varchar(255) NOT NULL,  
    ProductCategory varchar(255) NOT NULL,  
    ProductUnitPrice int NOT NULL,  
    CONSTRAINT PK_Product PRIMARY KEY (ProductID)  
)
```

- Selanjutnya membuat fact table dengan nama FactSalesOrder, table ini memiliki 7 attribute dengan 6 attribute bertipe data integer dan 1 attribute bertipe data date, seluruh attribute tidak boleh null atau kosong, attribute OrderID di set menjadi primary key dan attribute CustomerID, ProductID dan StatusID di set menjadi foreign key.

```
CREATE TABLE FactSalesOrder(  
    OrderID int NOT NULL,  
    CustomerID int NOT NULL,  
    ProductID int NOT NULL,  
    Quantity int NOT NULL,  
    Amount int NOT NULL,  
    StatusID int NOT NULL,  
    OrderDate date NOT NULL  
    CONSTRAINT PK_SalesOrder PRIMARY KEY (OrderID),  
    CONSTRAINT FK_Customer FOREIGN KEY (CustomerID)  
    REFERENCES DimCustomer (CustomerID),  
    CONSTRAINT FK_Product FOREIGN KEY (ProductID)  
    REFERENCES DimProduct (ProductID),  
    CONSTRAINT FK_StatusOrder FOREIGN KEY (StatusID)  
    REFERENCES DimStatusOrder (StatusID)  
)
```

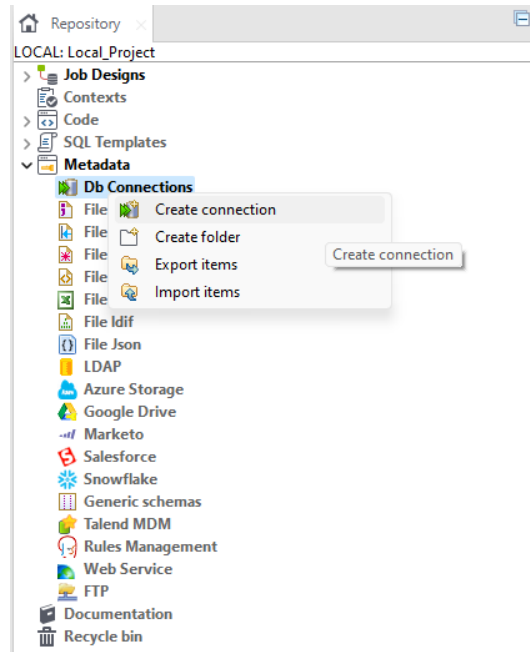
- Pada gambar dibawah sudah terlihat bahwa table sudah berhasil dibuat.



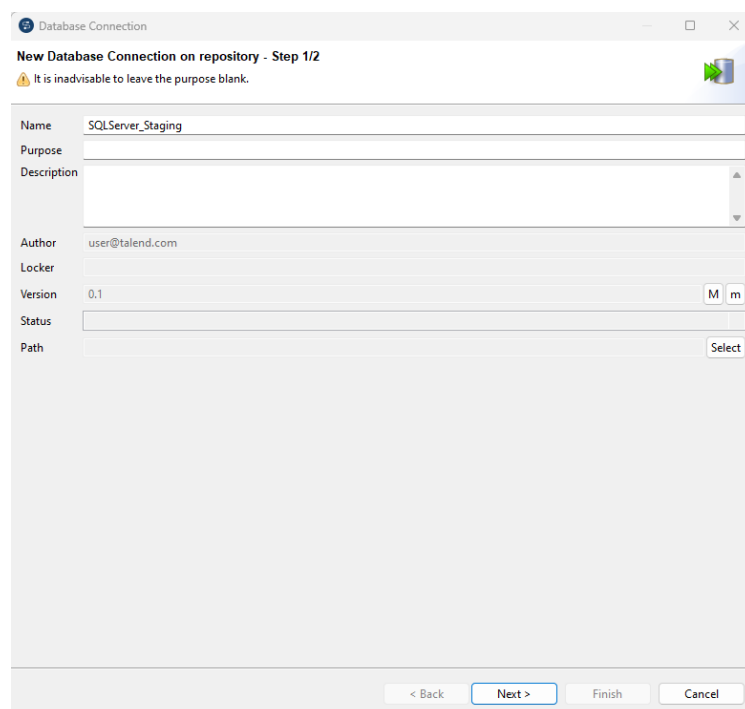
3. Membuat Job ETL di aplikasi talend untuk memindahkan data dari Staging ke Data Warehouse.

Khusus untuk Tabel DimCustomer, lakukan transformasi data dengan merubah data dari kolom FirstName dan LastName menjadi huruf kapital semua, lalu gabungkan kedua kolom tersebut menjadi satu kolom yang bernama CustomerName.

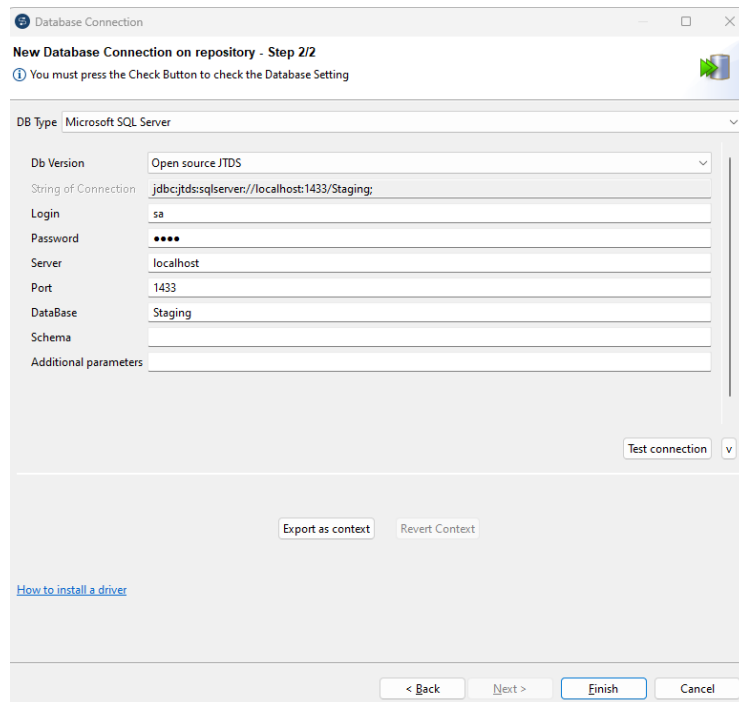
- Langkah pertama membuat koneksi database untuk menghubungkan talend dengan sql server, dengan cara klik metadata lalu klik kanan pada DB connections dan pilih Create connection



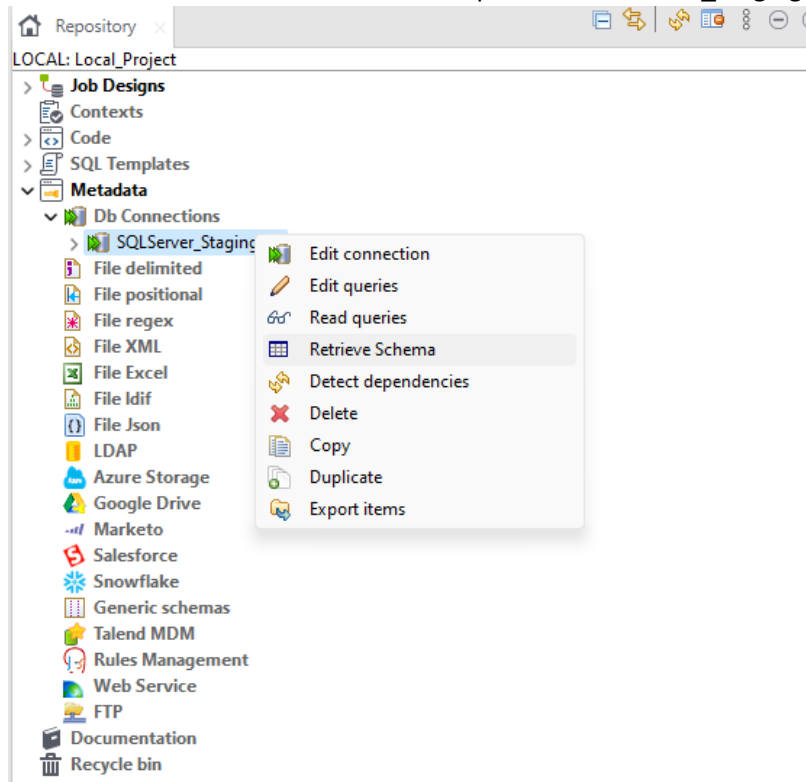
- Kemudian akan muncul tampilan seperti dibawah ini, disini saya memberi nama connection SQLServer_Staging setelah itu klik next.



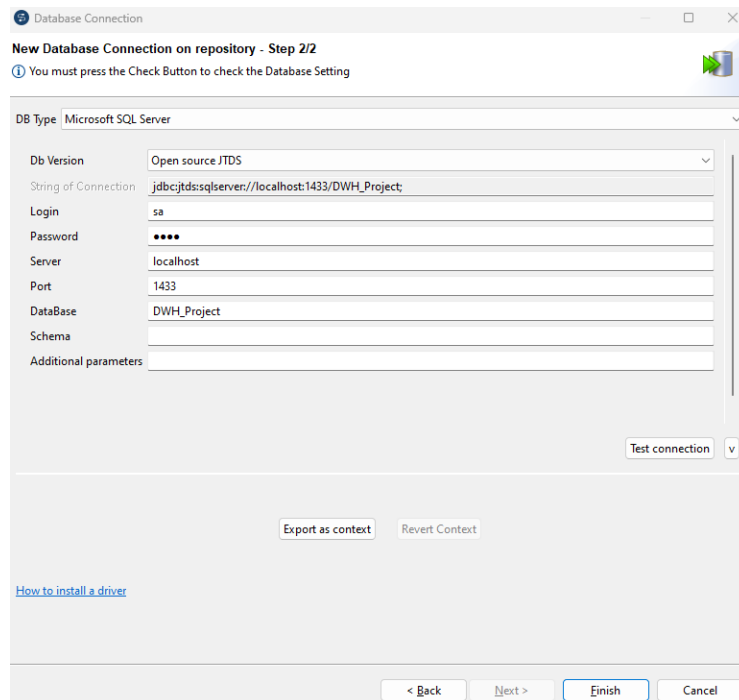
- Kemudian melakukan pengisian pada gambar dibawah ini, saya memilih DB type Microsoft SQL Server dan DB Version Open source JTDS dan database Staging karena database Staging akan dijadikan sebagai source



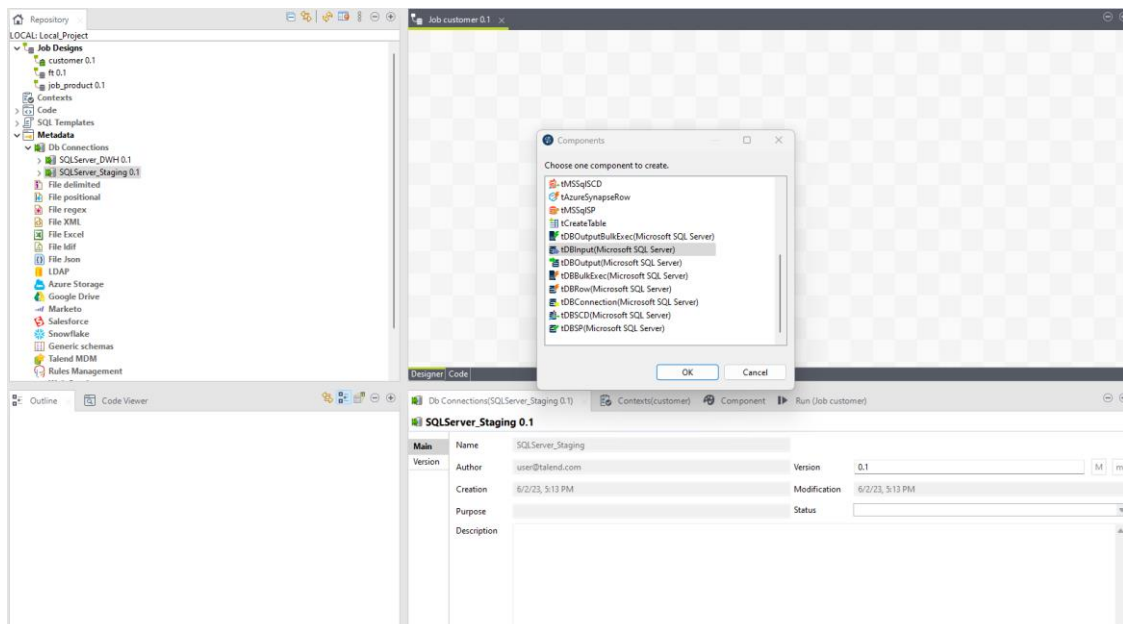
- Kemudian melakukan Retrieve Scheme pada db SQLServer_Staging



- lalu melakukan hal yang sama seperti membuat connection pada SQLServer_Staging namun untuk koneksi database DWH_Project, disini saya member nama connection SQLServer_DWH dan database yang di masukan adalah DWH_Project.

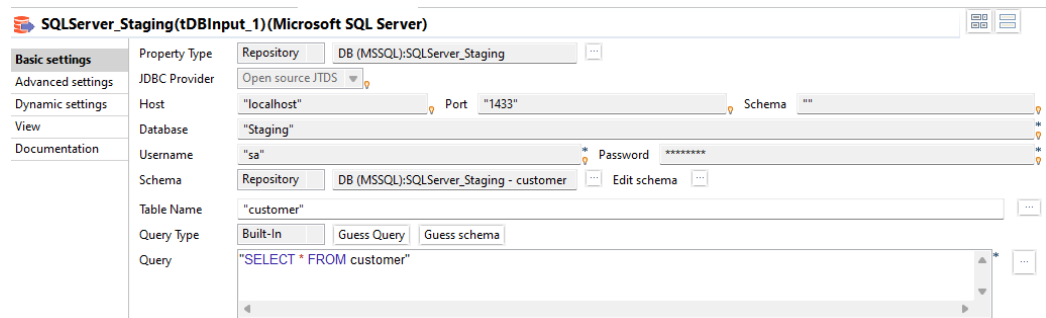


- Lalu memasukan component tDBInput untuk SQLServer_Staging karena database ini akan digunakan sebagai source.

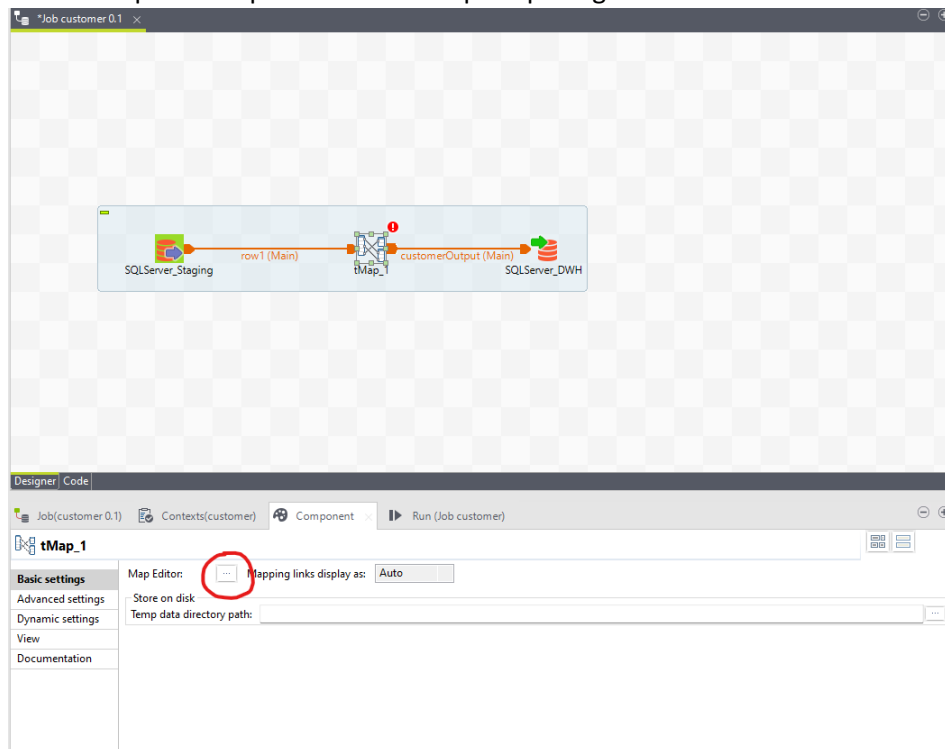


- Kemudian memasukan component tMap dan tDBOutput

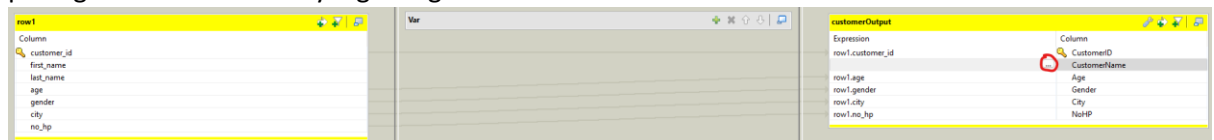
- Kemudian pada tDInput saya memilih schema repository dan mengambil tbale customer dan memasukan query untuk mengambil seluruh attribute pada table customer.



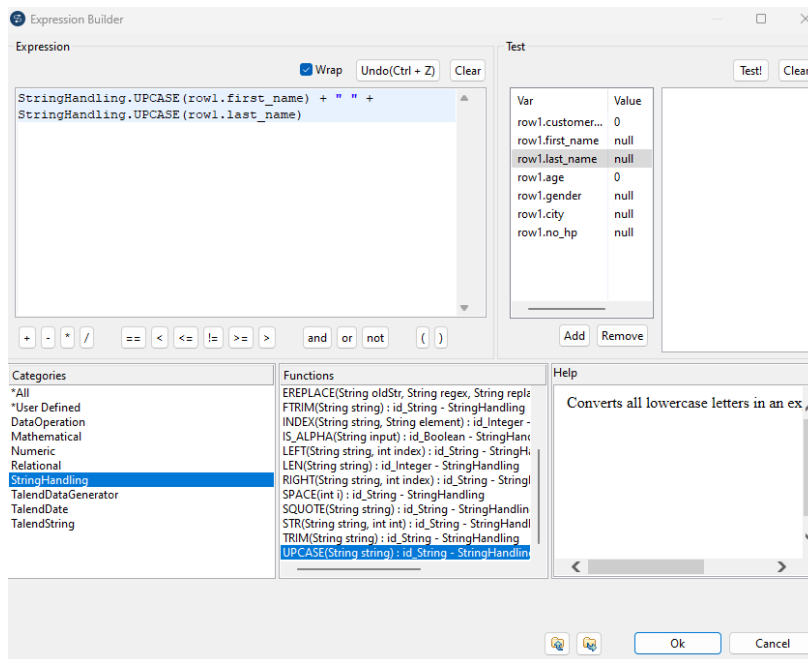
- Kemudian pada tMap klik tombol ... seperti pada gambar dibawah



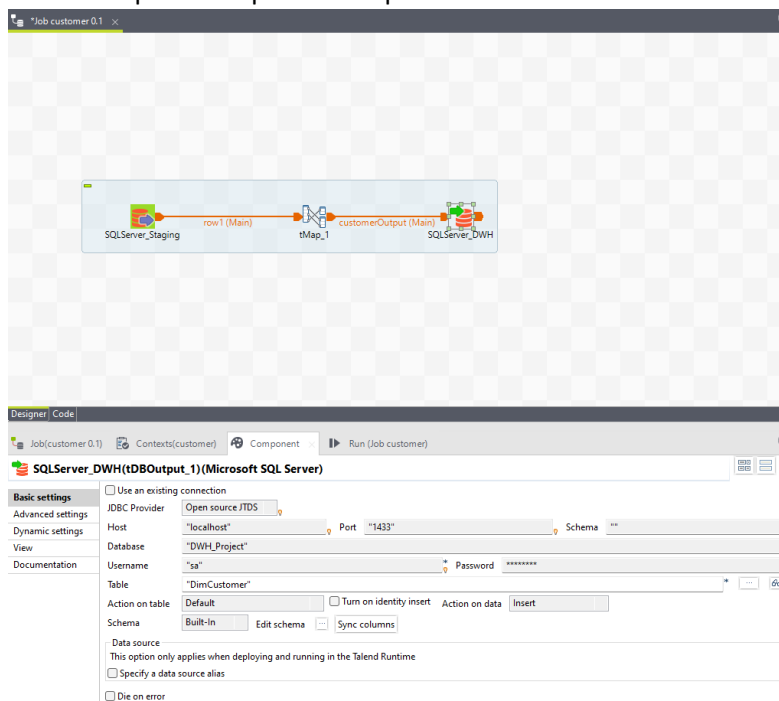
- Setelah itu menghubungkan attribute yang ada pada table customer di database Staging ke output untuk dimasukan kedalam table DimCustomer pada database DWH_project, setelah itu saya melakukan editing pada expression builder yang tertera pada gambar dibawah ini yang dilingkari merah.



- Pada bagian StringHandling saya memilih UPPERCASE untuk attribute first_name dan last_name untuk mengubah data menjadi huruf besar semua dan menambah space, setelah itu klik ok



- Kemudian pada component output memilih table DimCustomer



- Setelah itu saya melakukan step yang sama namun hanya berbeda pada table saja, seperti pada table product, sales_order, dan order status lalu run semua job yang sudah dibuat.

NOTE : perubahan pada attribute hanya dilakukan pada kolom DimCustomer karena menggabungkan kolom first_name dan last_name pada database Staging ke CustomerName pada database DWH_Project.

4. Membuat Store Procedure (SP) untuk menampilkan summary sales order berdasarkan status pengiriman.

```
CREATE PROCEDURE summary_order_status
    @StatusID int
AS
BEGIN
    SELECT
        a.OrderID,
        b.CustomerName,
        c.ProductName,
        a.Quantity,
        d.StatusOrder
    FROM FactSalesOrder a
    INNER JOIN DimCustomer b ON a.CustomerID = b.CustomerID
    INNER JOIN DimProduct c ON a.ProductID = c.ProductID
    INNER JOIN DimStatusOrder d ON a.StatusID = d.StatusID
    WHERE d.StatusID = @StatusID
END
```

Dalam store procedure ini, kita menggunakan beberapa JOIN antara tabel fact dan dimension (DimCustomer, DimProduct, DimStatusOrder) untuk mengambil kolom-kolom yang diperlukan (OrderID, CustomerName, ProductName, Quantity, StatusOrder). Selain itu, terdapat parameter @StatusID yang berfungsi sebagai filter untuk query, di mana hanya data dengan StatusID yang sesuai dengan nilai parameter yang akan ditampilkan.

Result :

```
EXEC summary_order_status @StatusID = 1;
```

Results		Messages			
	OrderID	CustomerName	ProductName	Quantity	StatusOrder
1	1303	BUDI SANTOSO	Macbook Air 2020 13 inch	1	Awaiting Payment
2	1310	LIA RAHMAWATI	Kipas Angin Cosmos	2	Awaiting Payment

```
EXEC summary_order_status @StatusID = 2;
```

Results		Messages			
	OrderID	CustomerName	ProductName	Quantity	StatusOrder
1	1301	LIA RAHMAWATI	Converse Cap Original	2	Awaiting Shipment
2	1304	AJENG SRIASIH	T-Shirt Polo Nevada	2	Awaiting Shipment
3	1307	RAHMA AMELIA	Pull & Bear T-Shirt	1	Awaiting Shipment

```
EXEC summary_order_status @StatusID = 3;
```

Results		Messages			
	OrderID	CustomerName	ProductName	Quantity	StatusOrder
1	1305	BAGUS PRAKOSO	Blender Philips 500 watt	3	Shipped
2	1308	BELA ADRILIA	Luciana Set Dress 2 in 1	2	Shipped

```
EXEC summary_order_status @StatusID = 4;
```

Results		Messages			
	OrderID	CustomerName	ProductName	Quantity	StatusOrder
1	1302	RIFKI MUHAMMAD	HP Elitebook 840 G4	1	Completed
2	1306	RIFKI MUHAMMAD	Asus Zenbook 800	1	Completed
3	1309	AJENG SRIASIH	Bagpack Navy Club	1	Completed

```
EXEC summary_order_status @StatusID = 5;
```

Results		Messages			
	OrderID	CustomerName	ProductName	Quantity	StatusOrder

Note:

StatusID	Keterangan
1	Awaiting Payment
2	Awaiting Shipment
3	Shipped
4	Completed
5	Cancelled

Pada database ini tidak ada order yang memiliki statusID 5 sehingga tidak ada value yang muncul pada saat statusID 5 dipanggil