JOBSHEET 2
ETL untuk SLOWLY CHANGING DIMENSION

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu membuat paket ETL yang mengakomodasi slowly changing dimension type 1, 2, dan 3.

2. Praktikum

Tabel dimensi sering diasumsikan independen terhadap waktu, tidak seperti tabel fakta yang selalu berubah. Sebenarnya, meskipun tabel dimensi relatif statis, nilai atributnya bisa saja lambat laun berubah. Slowly Changing Dimension adalah konsep dalam *data warehouse* yang digunakan untuk menangani perubahan data pada tabel dimensi seiring waktu.

Untuk setiap atribut pada tabel dimensi, Anda harus menentukan strategi yang perlu dilakukan saat terdapat perubahan nilai atribut tersebut pada OLTP database. SCD Type 1, 2, dan 3 merupakan strategi yang paling sering digunakan. Tipe dipilih berdasarkan kebutuhan user yang diperoleh saat requirement gathering.

Paket ETL yang akan dibuat pada praktikum ini akan menggunakan database legendvehicle sebagai data source. Database ini merupakan OLTP database dari suatu aplikasi e-commerce.Data destination dari paket ETL adalah dw_legendvehicle yaitu suatu data warehouse yang akan digunakan untuk analisa penjualan.

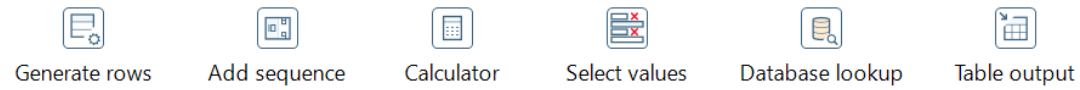
Tugas 1

Import database legendvehicle dan dw_legendvehicle. Buat database diagram dengan designer (yang disediakan phpMyAdmin) untuk kedua database tersebut.

2.1 Generate dan Load Rows untuk DimDate

Umumnya semua kolom pada DimDate dianggap sebagai SCD Type 0, artinya nilai kolom dalam dimension tidak pernah berubah. Data dianggap immutable (tetap) dan tidak diupdate. Sebagai contoh jika DimDate memiliki kolom Hari, Triwulan, dan Semester maka 9 Oktober 2020 akan selalu hari Jum'at, triwulan 4, dan semester 2. Hari, triwulan, dan semesternya tidak akan pernah berubah. Baris-baris pada DimDate biasanya disiapkan di awal, misalnya mencakup seluruh tanggal pada tahun 2000-2010. Jika perlu menambahkan tahun-tahun baru, ini bukan karena perubahan data historis tetapi karena kalender berjalan maju sehingga tidak dianggap sebagai "Slowly Changing" dalam konteks SCD.

1. Buat Transformation baru **File → New → Transformation.**
2. Drag and Drop beberapa objek berikut



Generate Rows: membuat baris data baru.

Add Sequence: membuat sequence

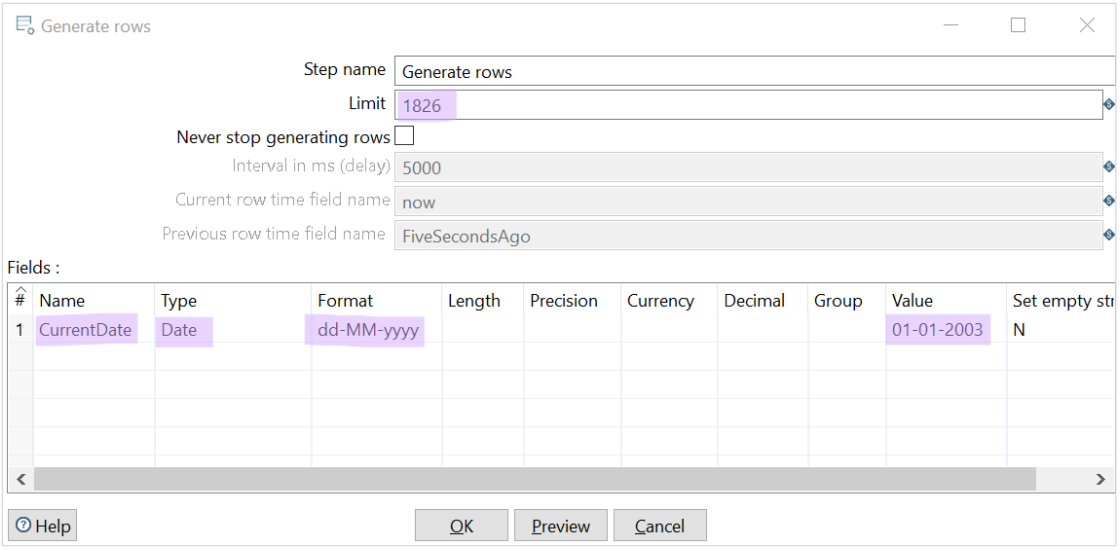
Calculator: melakukan kalkulasi

Select Values: memilih field yang ingin diteruskan ke step berikutnya

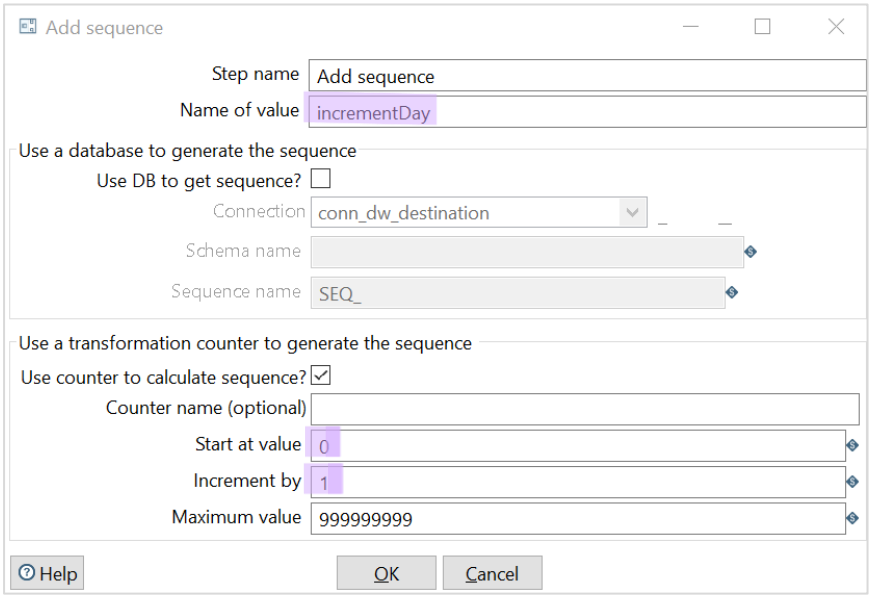
Database Lookup: mengecek data di database berdasarkan kondisi tertentu (seperti konsep join)

Table Output: menyimpan data ke tabel

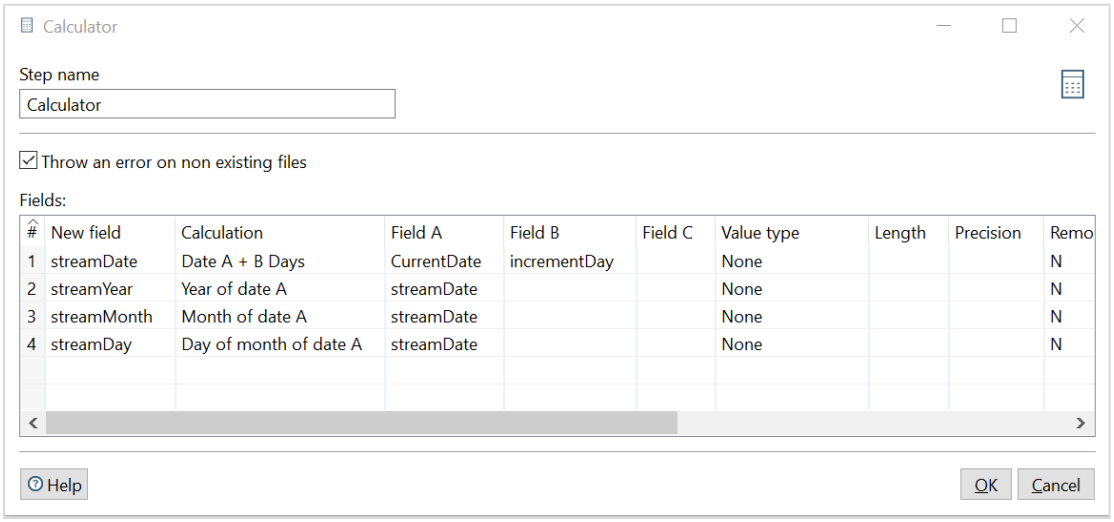
3. Konfigurasi **Generate Rows** dengan set limit 1826 karena akan dibuat 1826 baris pada DimDate. 1826 merupakan jumlah hari dalam 5 tahun (365 hari x 5 tahun + 1 hari pada tahun kabisat).
4. Tambahkan field bernama CurrentDate dengan type Date, format dd-MM-yyyy, dan value awal 01-01-2003.



5. Hubungkan output dari Generate Rows menuju Add Sequence.
6. Konfigurasi **Add Sequences**. Set Name of value incrementDay (nama field yang akan menyimpan sequence) dengan start value bernilai 0 dan increment by bernilai 1



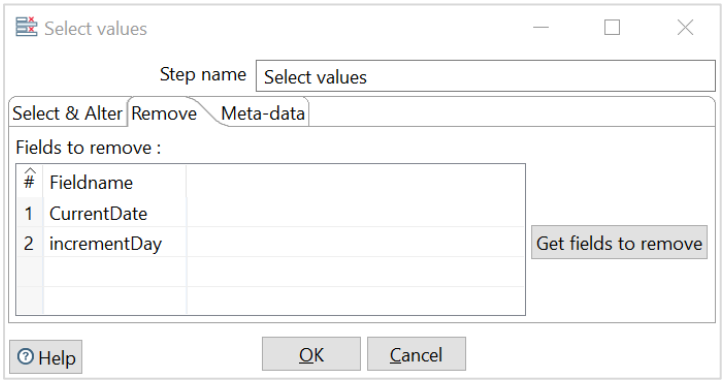
7. Hubungkan output dari Add Sequences menuju Calculator.
8. Konfigurasi pada **Calculator** dengan membuat fields baru sebagai berikut
   - streamDate merupakan kalkulasi dari CurrentDate + incrementDay
   - streamYear merupakan Year dari streamDate
   - streamMonth merupakan Month dari streamDate
   - streamDay merupakan Day of month dari streamDate



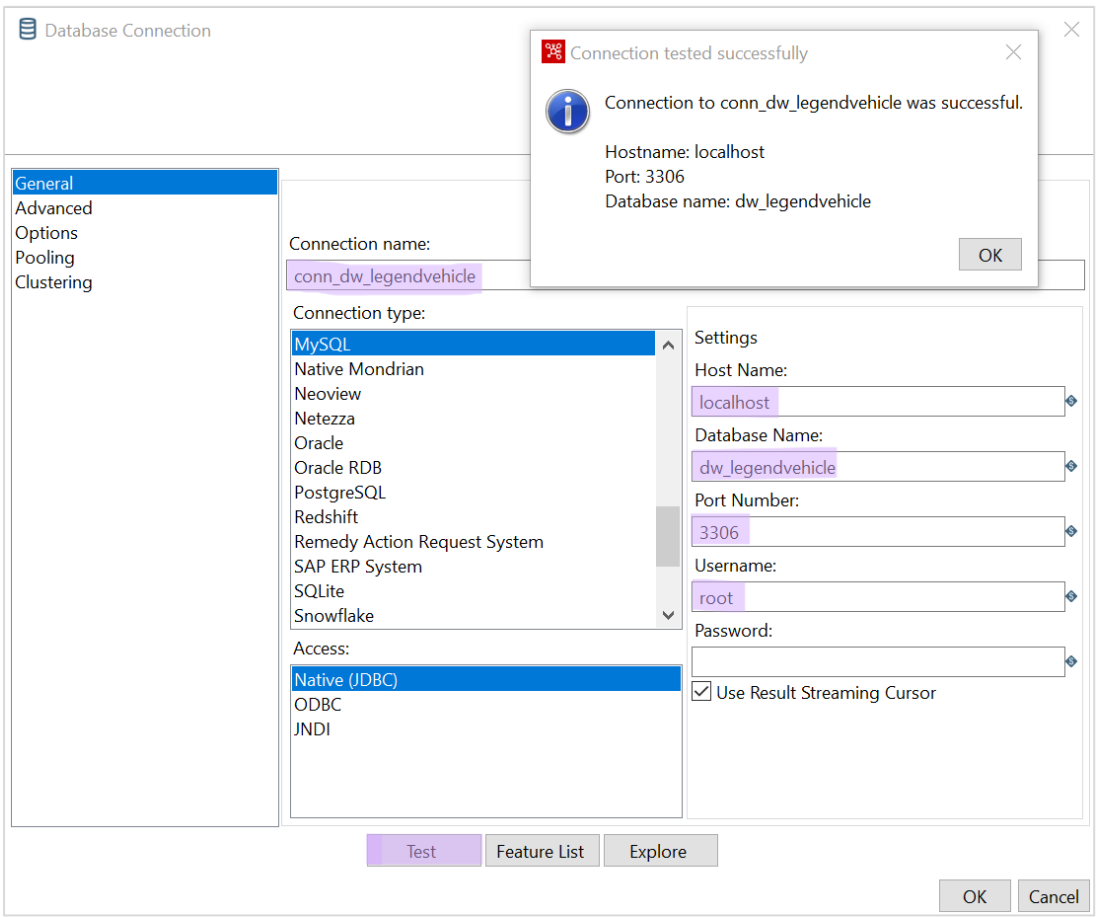
#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remo
1	streamDate	Date A + B Days	CurrentDate	incrementDay		None			N
2	streamYear	Year of date A	streamDate			None			N
3	streamMonth	Month of date A	streamDate			None			N
4	streamDay	Day of month of date A	streamDate			None			N

9. Hubungkan output dari Calculator menuju Select values
10. **Select values** digunakan untuk menentukan field yang akan diteruskan pada step selanjutnya. Anda bisa memilih untuk menggunakan tab **Select & Alter** atau tab **Remove**, bergantung pada banyaknya field yang akan dipilih atau dihapus. Gunakan button **Get fields to select** atau **Get fields to remove** jika diperlukan.

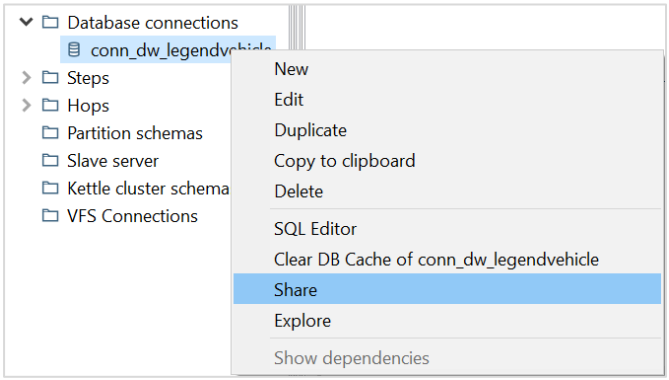
    Kali ini, gunakan tab Remove, kemudian pilih field CurrentDate dan incrementDay untuk dihapus karena keduanya tidak diperlukan lagi.



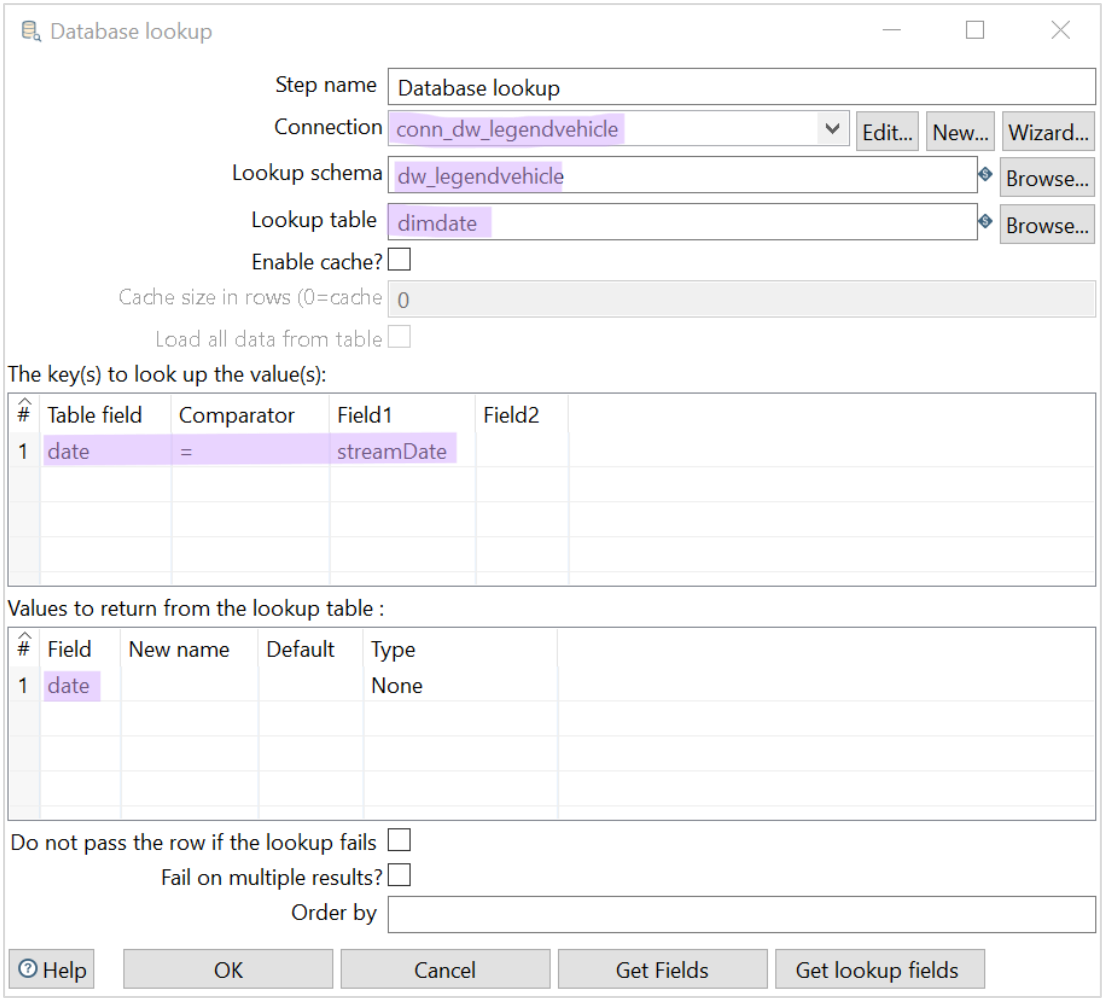
11. Hubungkan output Select values menuju Database lookup.
12. Sebelum melakukan konfigurasi pada database lookup, buatlah koneksi terlebih dahulu pada database melalui File → New → **Database Connection**. Gunakan Connection type MySQL dan access Native (JDBC) dengan host name, database name, port number, username dan password sesuai konfigurasi MySQL pada device masing-masing. Beri nama connection tersebut dengan nama **conn_dw_legendvechicle**. Test connection untuk memastikan settings sudah tepat.



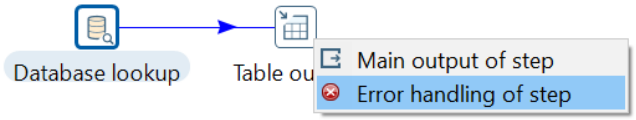
13. Right click pada connection yang baru dibuat, pilih **Share** agar dapat digunakan pada transformations lainnya



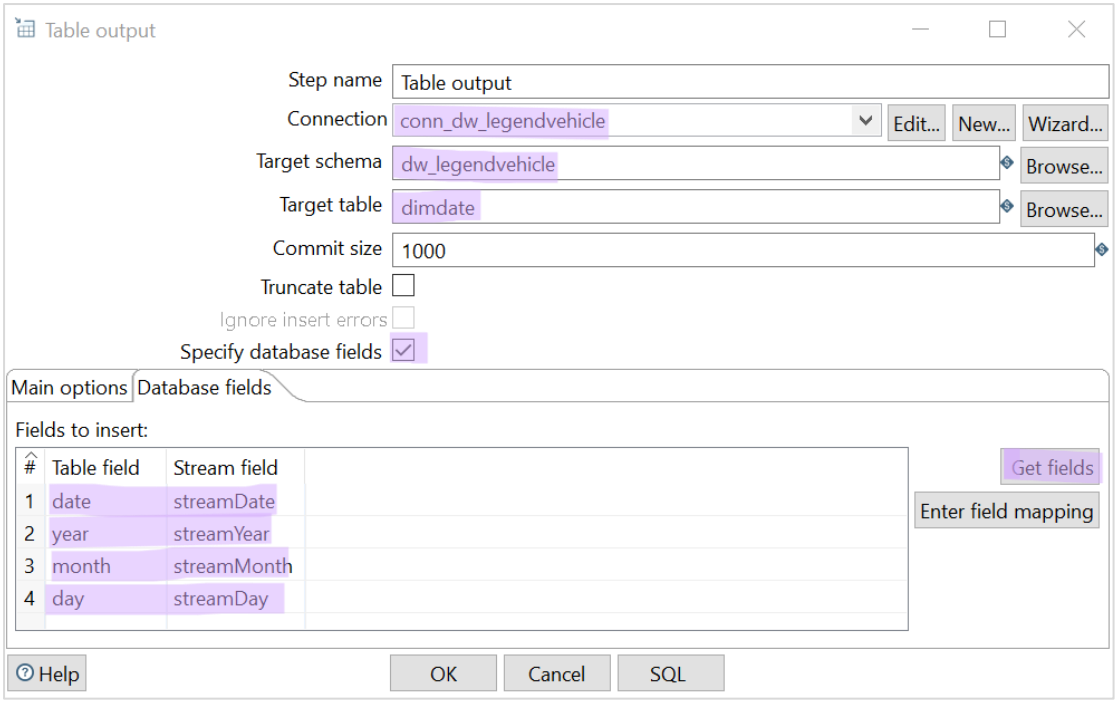
14. Konfigurasi pada **Database lookup**, set connection, lookup schema (nama database), dan lookup table (tabel yang akan dilookup). Step ini digunakan untuk mengecek apakah suatu tanggal sudah tersimpan di table DimDate dengan membandingkan field streamDate dari step sebelumnya dan field date dari table DimDate.



15. Hubungkan output dari Database lookup dengan Table Output, pilih **Error handling of step**. Artinya, baris yang akan di-insert adalah baris yang tidak berhasil di-lookup pada step Database lookup, yaitu tanggal yang belum tersedia pada table DimDate.



16. Konfigurasi **Table output** dengan set connection, target schema, dan target table. Aktifkan **Specify database fields**. Pada tab Database fields, mapping field streamDate, streamYear, streamMonth dan streamDay dengan fields yang ada pada DimDate (untuk membantu, klik button **Get fields** kemudian edit).



17. Save transformation kemudian Run
18. Cek isi table DimDate pada database. Transormation yang dibuat sukses jika DimDate terisi 1826 data.

Tugas 2

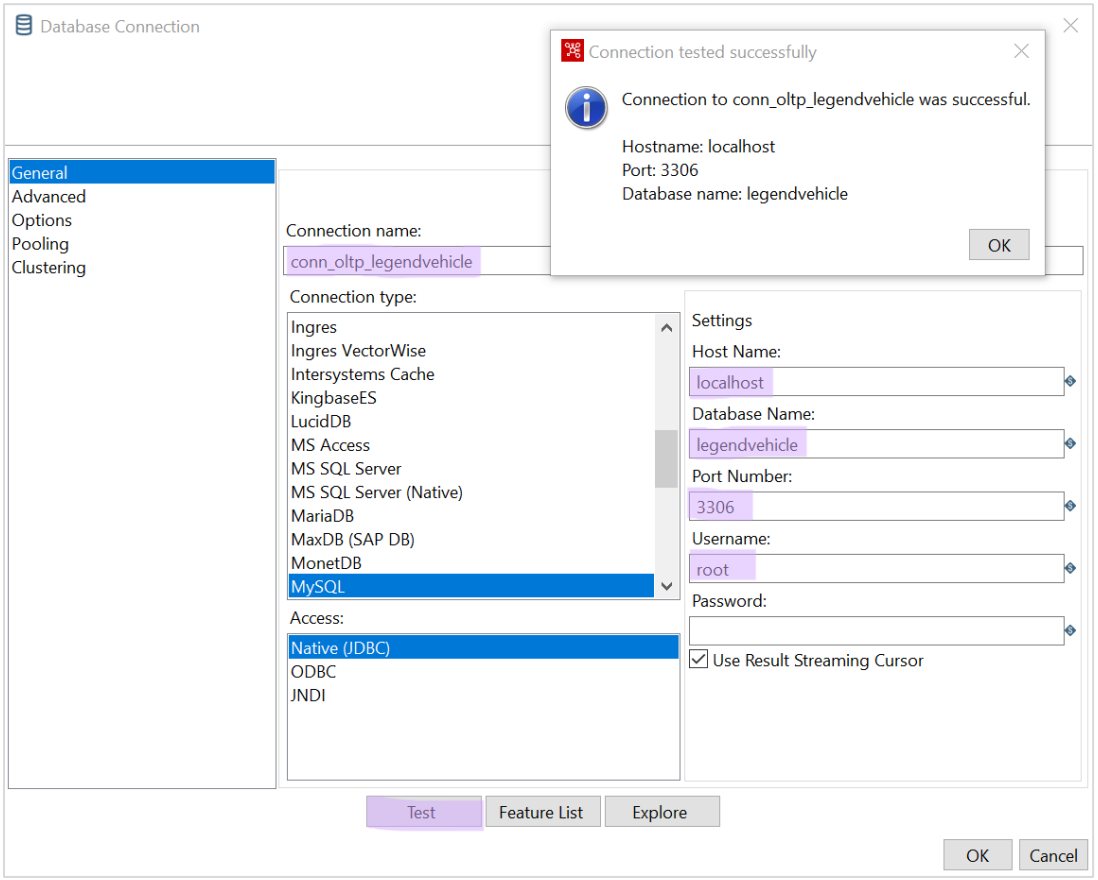
Buka tab Preview Data pada Execution Results. Screenhot Preview Data dan jelaskan apa yang dilakukan pada setiap step berikut:

1. Generate rows
2. Add Sequences
3. Calculator
4. Select values
5. Database lookup
6. Table Output

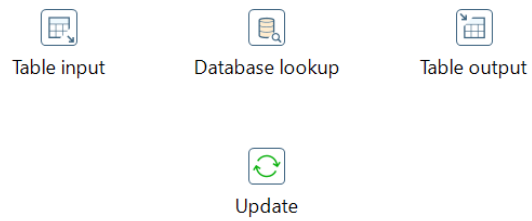
2.2 SCD Type 1 – Overwrite

SCD Type 1 umumnya digunakan untuk perubahan data yang merupakan proses koreksi atau jika riwayat perubahan nilai tidak perlu disimpan. Pada praktikum ini, semua field pada table DimProduct (selain keys) *diasumsikan* menggunakan SCD Type 1.

1. Buat Transformation baru **File → New → Transformation.**
2. Buat connection baru ke database legendvehicle sebagai data source untuk proses ekstraksi. Beri nama connection tersebut dengan nama **conn_oltp_legendvechicle**. Test connection untuk memastikan settings sudah tepat.

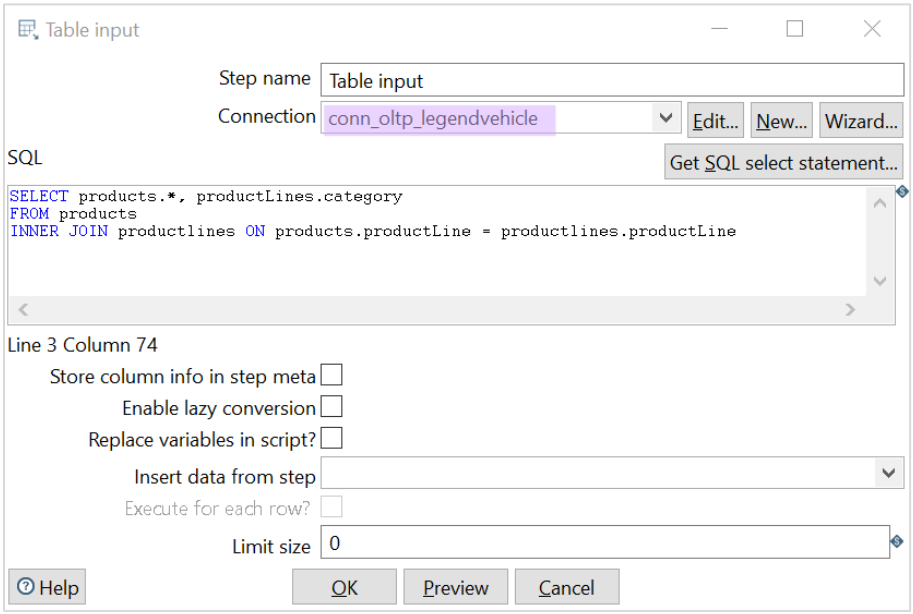


3. **Share conn_oltp_legendvehicle** dengan cara yang sama seperti sebelumnya agar dapat dipakai untuk transformations lain
4. Drag and drop beberapa object berikut:



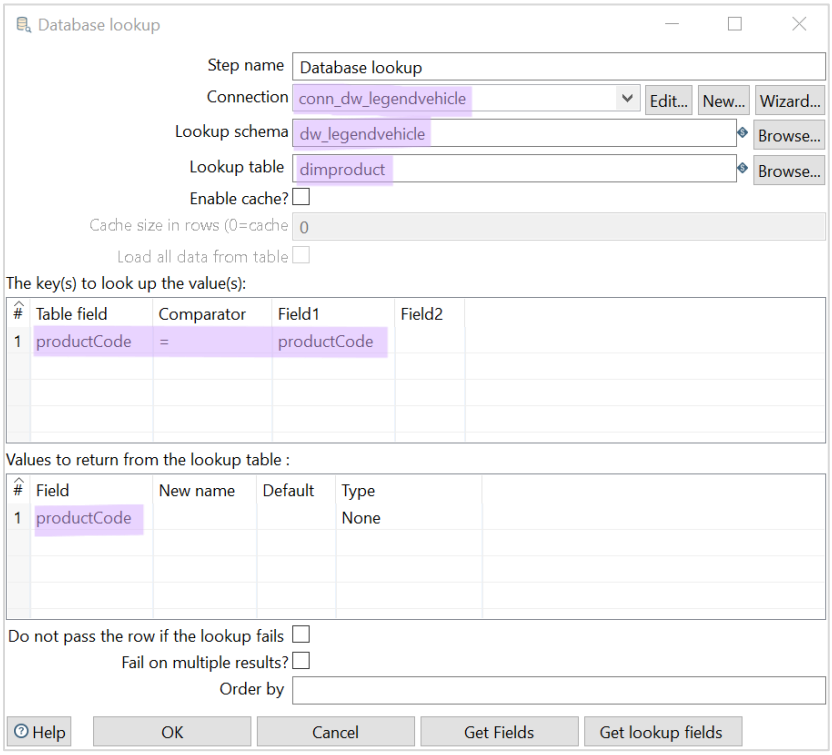
- **Table input**: melakukan ekstraksi data
- **Select values**: memilih dan merename field yang akan digunakan untuk proses transform dan load
- **Database lookup**: mengecek apakah baris data product sudah tersedia pada tabel DimProduct
- **Table output**: insert data baru yang belum tersedia di DimProduct
- **Update**: update jika data produk sudah tersedia pada table DimProduct

5. Konfigurasi step **Table Input**, set connection ke oltp dan gunakan query seperti pada gambar berikut untuk **join table products dan productLines** karena skema yang digunakan pada data warehouse adalah star schema.

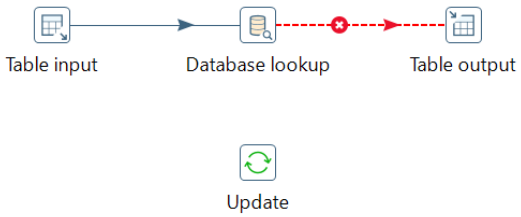


6. Hubungkan ouput step Table Input ke step Database Lookup
7. Konfigurasi **Database lookup**, set connection, lookup schema, dan lookup table. Kita akan mengecek apakah suatu data product di OLTP database sudah tersedia pada dimProduct berdasarkan primary key nya, yaitu productCode.

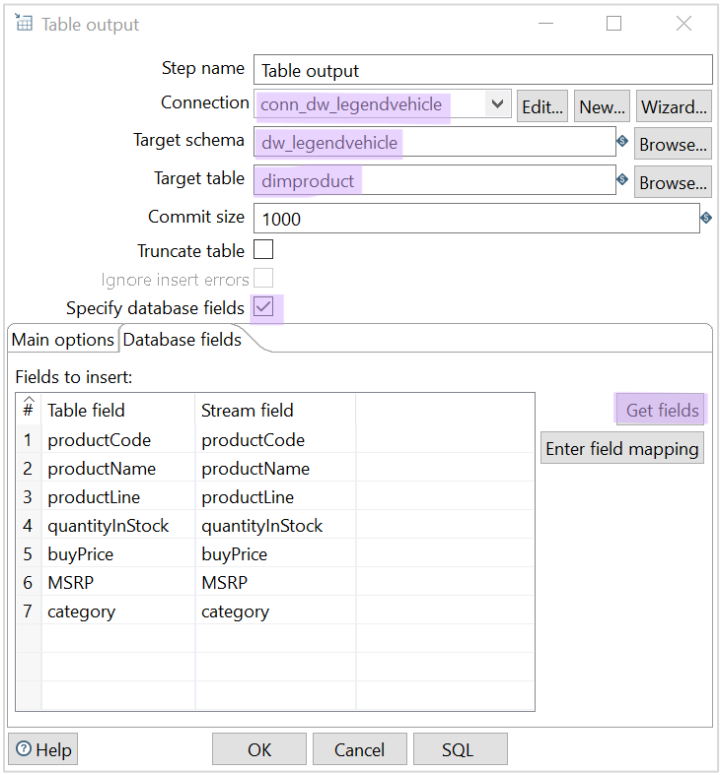
   Field yang akan dibandingkan adalah **productCode dari table dimProduct dengan productCode yang berasal step sebelumnya (hasil ekstraksi dari OLTP).**



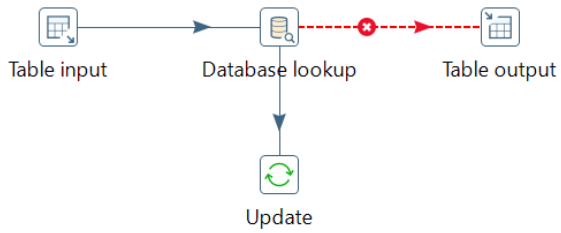
8. Hubungkan output step Database Lookup (**error**) ke step Table Output. Untuk setiap baris data yang productCode nya tidak berhasil di-lookup, maka akan ditambahkan baris baru pada tabel dimProduct.



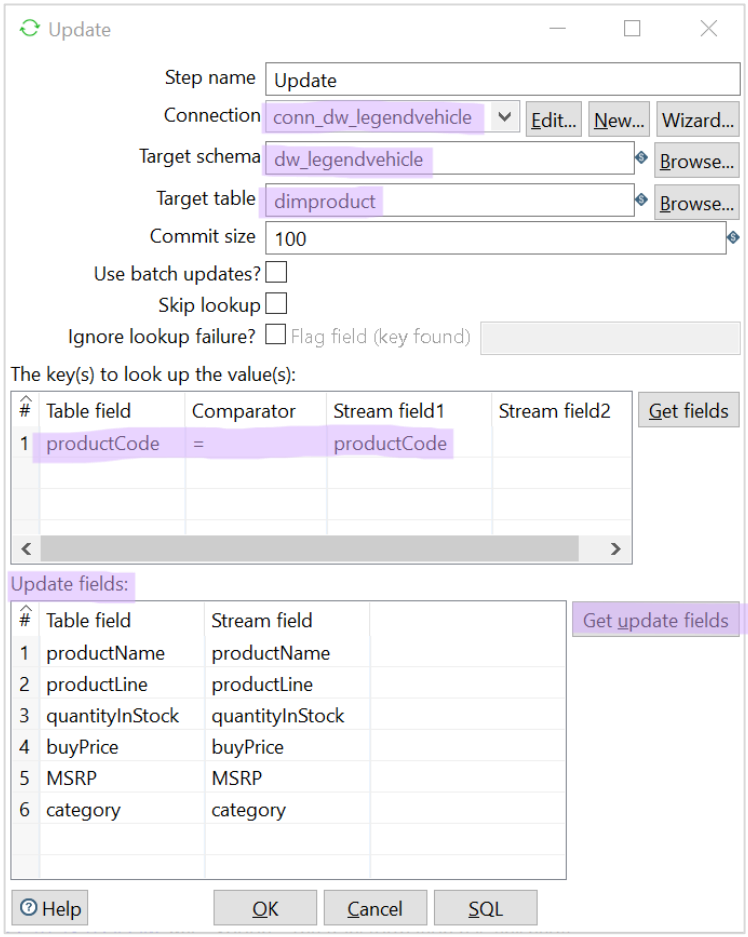
9. Pada step **Table Output**, set connection, target schema, dan target table. Aktifkan **Specify database fields** dan lakukan mapping dengan bantuan **Get fields**



10. Hubungkan output step Database Lookup (**main**) ke step Update. Set connection, target schema, target table, dan keys.



Data diupdate berdasarkan kesesuaian productCode (set di bagian **The keys to look up the values**) dan tentukan field yang akan diupdate di bagaian **Update fields**. Gunakan bantuan button **Get update fields** jika perlu.



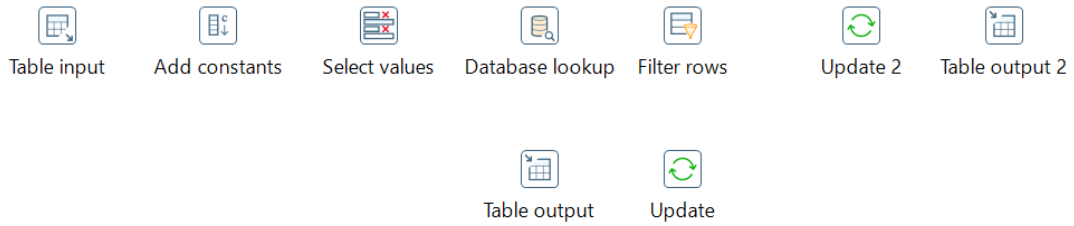
Tugas 3

1. Save dan run transformation. Screenshot Step Metrics pada Execution Results.
2. Periksa apakah data produk sudah tersimpan di DimProduct
3. Insert 1 produk baru di table products (OLTP) dan update 1 data produk (misalnya ubah productName atau category). Run transformation kembali dan screenshot Step Metrics pada Execution Result.
4. Apakah ada duplikasi data produk? Mengapa?
5. Modifikasi step **Update** pada langkah 10 agar field **buyPrice** menggunakan SCD Type 0. Setelah modifikasi dilakukan, coba lakukan perubahan di OLTP, run transformation, dan screenshot Step Metrics. Pastikan field buyPrice di DW tidak berubah meskipun ada perubahan di OLTP. Screenshot dan jelaskan modifikasi yang dilakukan.

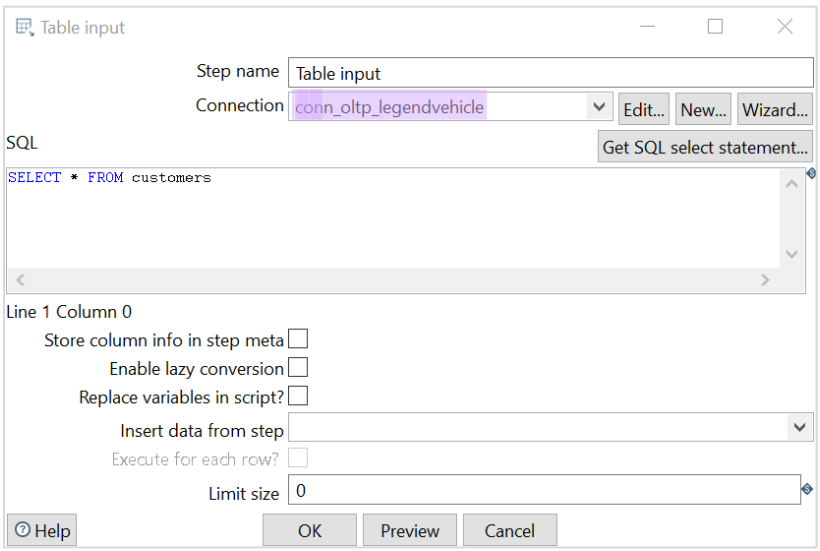
2.2 SCD Type 2 – Add New Row

SCD Type 2 digunakan jika riwayat suatu atribut penting untuk disimpan. Pada contoh kasus ini, **asumsikan** bahwa untuk keperluan analisa, pada table DimCustomer diberlakukan **SCD Type 1** untuk field **customerName** dan **phone** serta **SCD Type 2** untuk field **city**

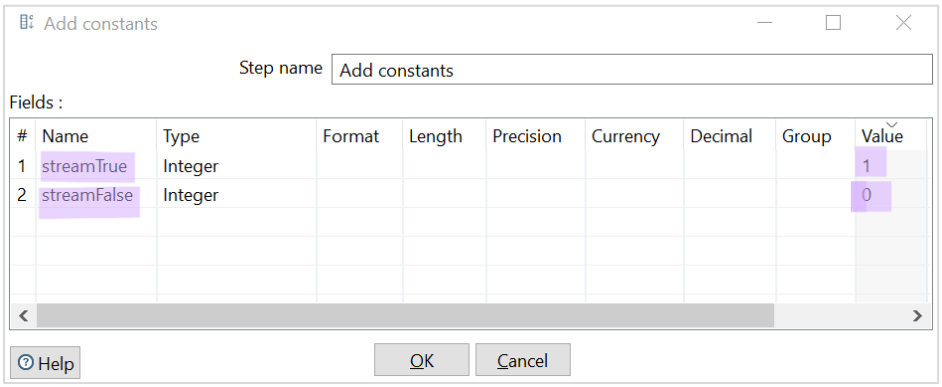
1. Buat Transformation baru **File → New → Transformation.**
2. Drag and drop step berikut



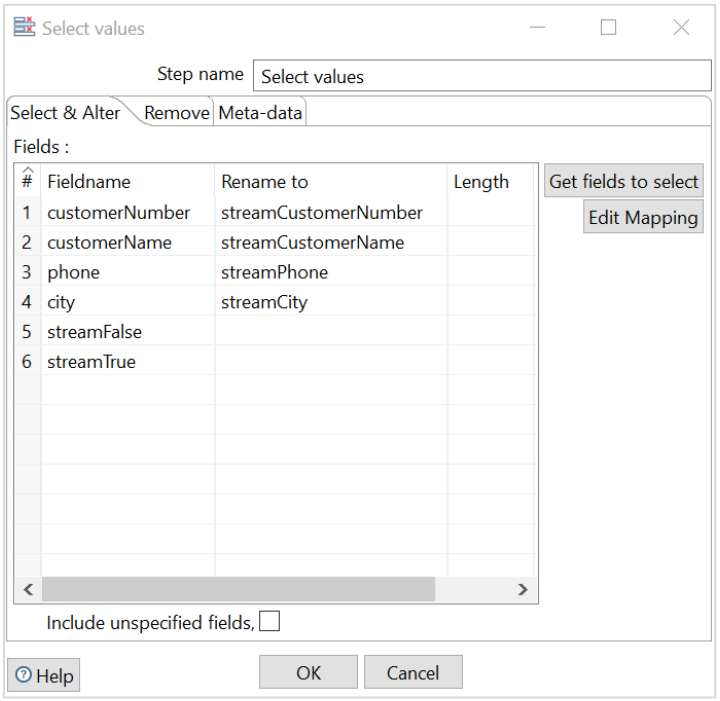
3. Konfigurasi **Table input**. Set connection dan lakukan ekstraksi dari tabel customers dengan query seperti pada gambar.



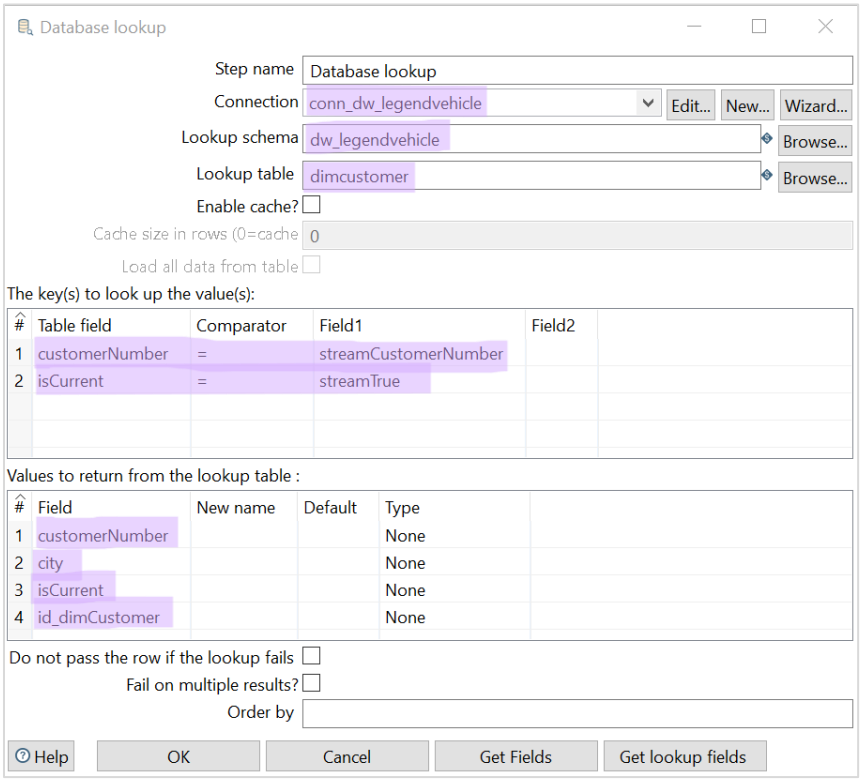
4. Hubungkan output step Table input ke Add constant.
5. Konfigurasi **Add constant**. Field streamTrue dan streamFalse nantinya akan digunakan untuk mengecek nilai field isCurrent.



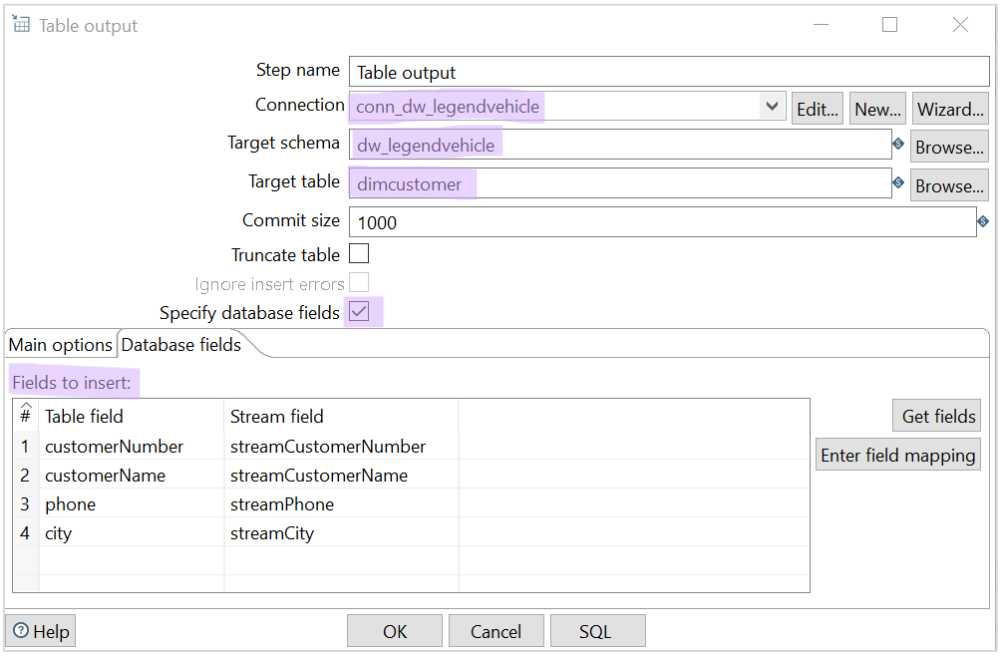
6. Hubungkan output Add constant ke Select values.
7. Konfigurasi **Select values**. Dalam kasus ini, step Select values lebih ditujukan untuk melakukan rename field (menambahkan prefix stream…) agar tidak ambigu pada step selanjutnya.



8. Hubungkan output Select values ke Database lookup
9. Konfigurasi **Database lookup**. Set connection, lookup schema, dan lookup table. Step ini digunakan untuk mengecek apakah data suatu customer sudah tersimpan dengan field isCurrent bernilai true. Selain field customerNumber dan isCurrent yang dipakai dalam lookup, tambahkan pula field city dan id_dimCustomer yang nanti diperlukan pada step selanjutnya.

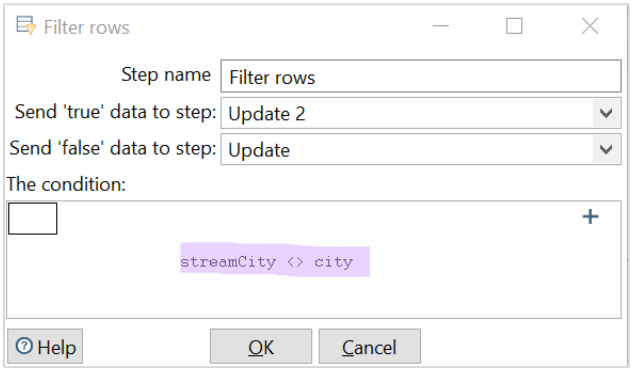


10. Hubungkan output dari Database lookup (**error**) ke Table output, artinya jika lookup gagal data customer baru akan ditambahkan.
11. Konfigurasi **Table output**, set connection, target schema, dan target table. Check Specify database fields kemudian mapping kolom pada bagian Fields to insert seperti pada gambar.

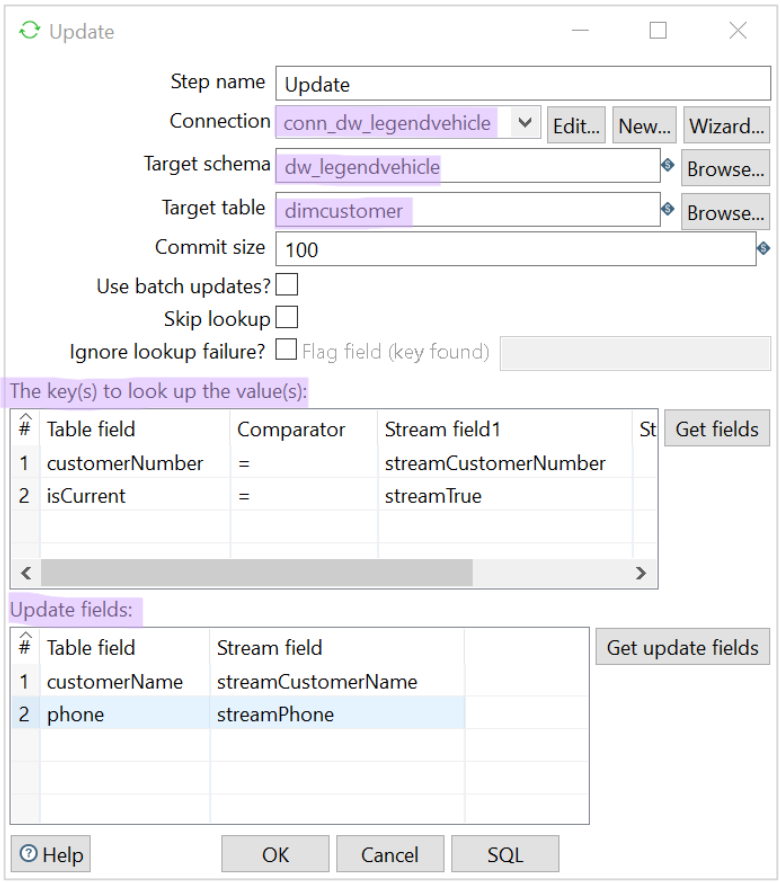


12. Hubungkan output dari Database lookup (**main**) ke Filter rows.

13. Konfigurasi **Filter rows**. Step ini digunakan untuk mengecek apakah terdapat perubahan nilai field city. Field streamCity diperoleh dari proses ekstraksi terhadap OLTP sedangkan field city merupakan field dari table DimCustomer.

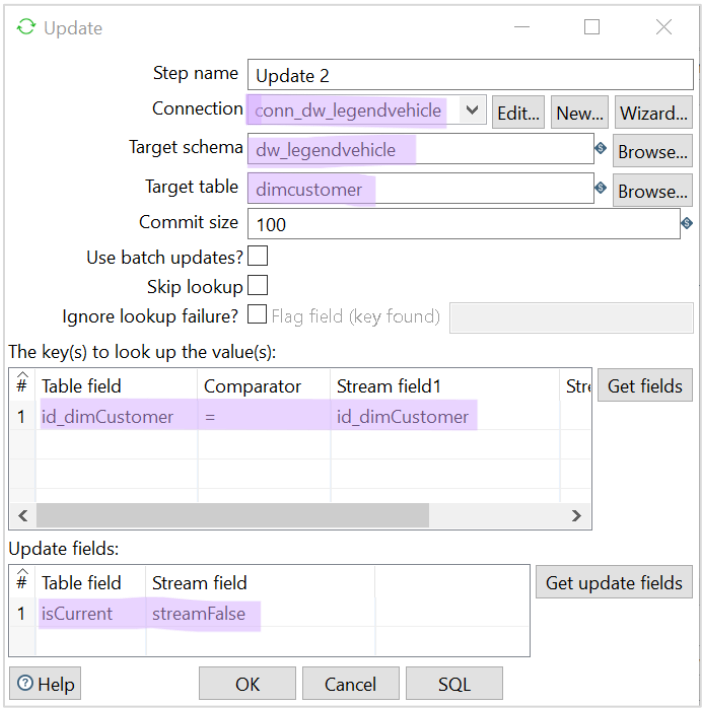


14. Hubungkan output Filter rows (**Result is false**) ke step Update.
15. Konfigurasi step **Update**. Pada langkah ini, dilakukan update jika tidak terdapat perubahan nilai field city. Set connection, target schema, dan target table. Set **Keys to look up the values** untuk memastikan data yang diupdate sesuai. **Update fields** customerName dan phone berdasarkan data stream yang diperoleh dari ekstraksi.

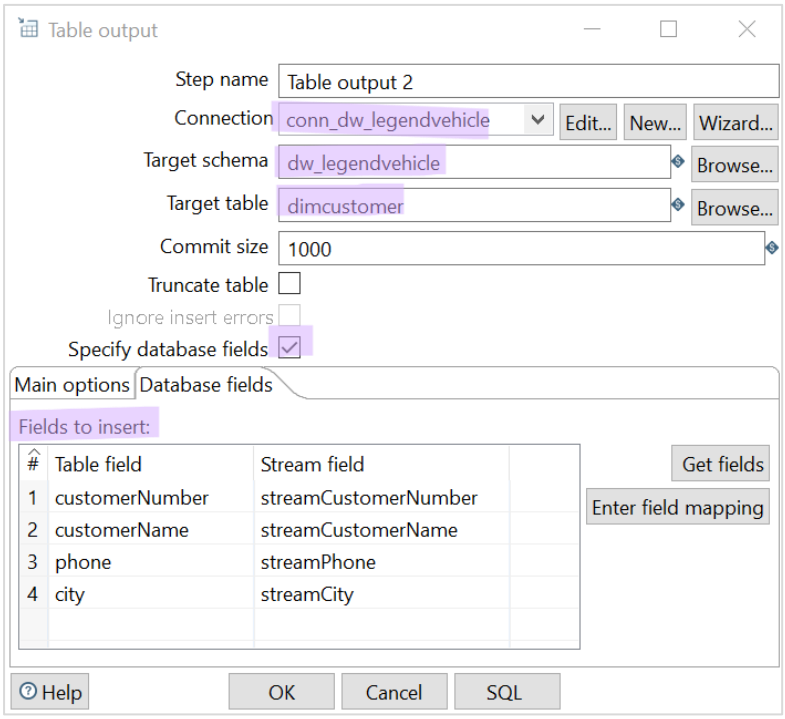


16. Hubungkan output Filter rows (**Result is true**) ke step Update 2. Result is true artinya terdapat perubahan nilai field city.

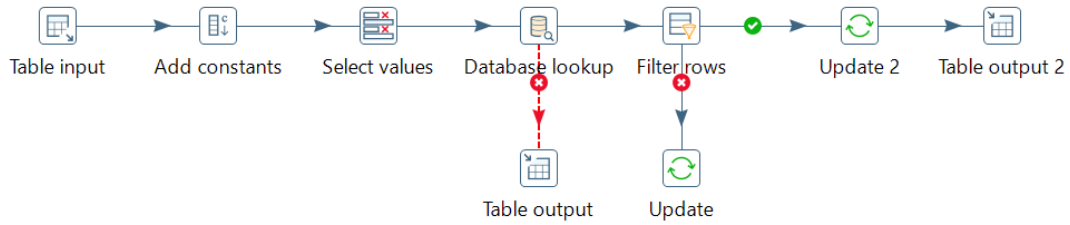
17. Step **Update 2** digunakan untuk mengupdate nilai field isCurrent menjadi false untuk data customer dengan nilai city yang lama. Set connection, target schema, dan target table. Pastikan perubahan dilakukan terhadap data yang benar dengan lookup value id_dimCustomer. Update nilai isCurrent menjadi false.



18. Hubungkan output Update 2 ke step Table output.
19. Konfigurasi **Table output** untuk menambahkan baris baru untuk customer yang sama jika terdapat perubahan nilai field city.



20. Transformation akhir terlihat seperti gambar berikut



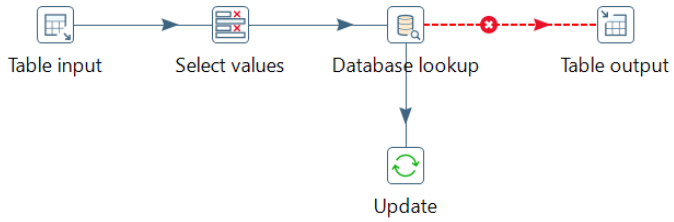
Tugas 4

1. Save dan run transformation. Screenshot Step Metrics pada Execution Results.
2. Insert 1 baris baru pada table customers di OLTP. Modifikasi pula field phone pada salah satu data customer dan field city pada salah satu customer lainnya.
3. Run transformation kembali dan screenshot Step Metrics pada Execution Result. Ada berapa baris baru yang ditambahkan pada table DimCustomer?
4. Lakukan percobaan, jika **The keys to look up the values** pada langkah 14 diubah seperti pada langkah 16, apakah paket ETL masih bekerja dengan benar? Lakukan percobaan kemudian simpulkan.

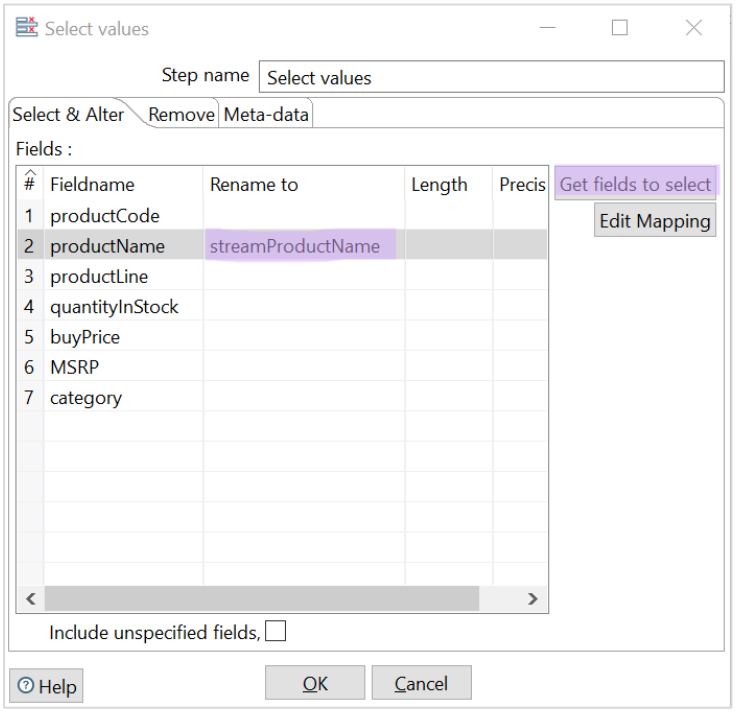
2.3 SCD Type 3 – Add New Column

SCD Type 3 dengan menambahkan kolom baru umumnya digunakan jika riwayat yang ingin disimpan jumlahnya terbatas, sedikit, dan sudah diketahui di awal pengembangan data warehouse. **Asumsikan** bahwa pada kasus ini, atribut productName perlu disimpan data saat ini dan satu data sebelumnya.

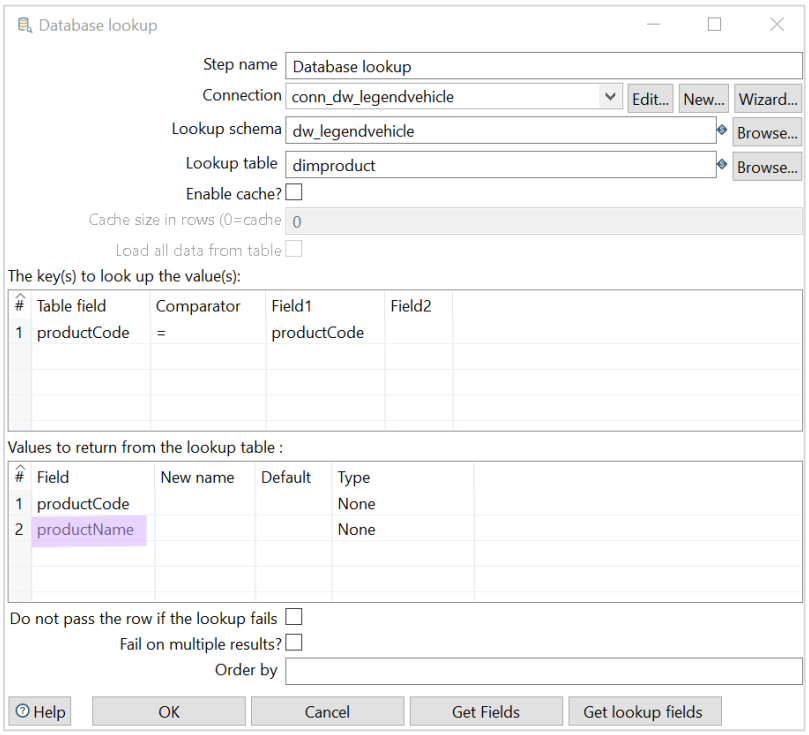
1. Pada table dimProduct di database dw_legendvehicle, tambahkan kolom priorProductName (allow NULL)
2. Buka transformasi yang dibuat pada praktikum 2.1 kemudian lakukan modifikasi berikut.
3. Agar field productName tidak ambigu pada step berikutnya, tambahkan step **Select values** kemudian tata kembali input output sehingga berubah seperti gambar berikut.



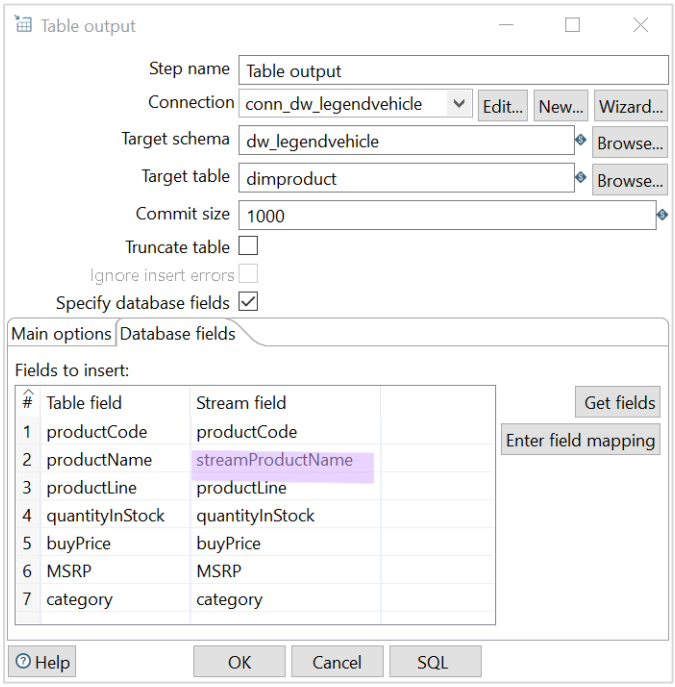
4. Konfigurasi **Select values**. Gunakan button **Get fields to select** untuk membantu. Rename field productName menjadi streamProductName.



5. Update step **Database lookup**. Tambahkan field productName dari table DimProduct karena nantinya akan digunakan untuk mengupdate nilai field priorProductName.

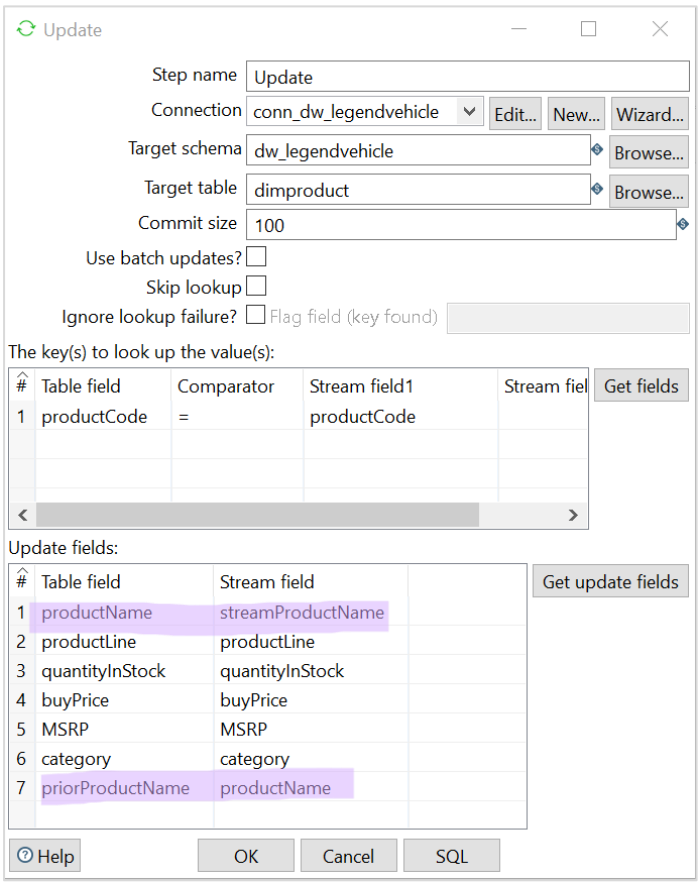


6. Update step **Table output**



7. Update step **Update**. Field productName di table DimProduct akan diupdate berdasarkan streamProductName yang berasal dari OLTP. Sedangkan field priorProductName diupdate berdasarkan productName sebelumnya.

   Catatan: rename field productName menjadi streamProductName sebelumnya dilakukan agar tidak terjadi ambiguitas pada langkah ini.



Tugas 5

1. Save dan run transformation. Screenshot Step Metrics pada Execution Results
2. Lakukan perubahan productName di OLTP. Run transformation dan screenshot Step Metrics kembali. Periksa apakah field productName dan priorProductName di DimProduct berhasil diupdate.
3. Tanpa melakukan perubahan data di OLTP, run kembali transformation. Perhatikan DimProduct. Mengapa priorProductName tetap diupdate meskipun tidak ada perubahan? Perbaiki kesalahan tersebut dengan menambahkan step **Filter rows** sebelum **Update** seperti pada contoh berikut**.** Screenshot konfigurasi step yang ditambahkan dan dimodifikasi.