TÉCNICO
LISBOA

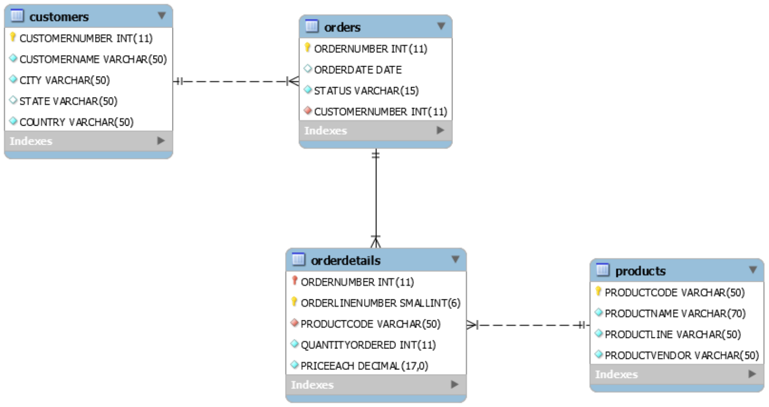
Data Analysis and Integration

ETL Processes Data Warehousing

Introduction

- How to build a data warehouse
  - ETL process
    - **Extract** data from original database
    - **Transform** data to fit star schema
    - **Load** data onto data warehouse

database

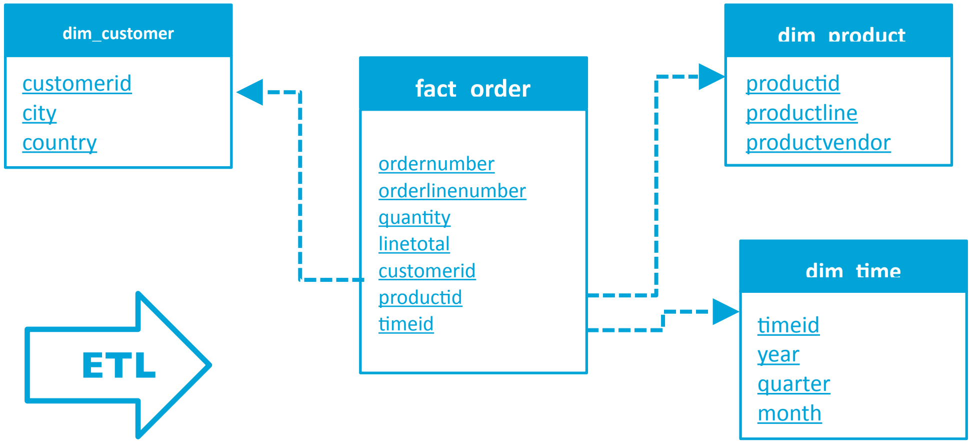


customers
- CUSTOMERNUMBER INT(11)
- CUSTOMERNAME VARCHAR(50)
- CITY VARCHAR(50)
- STATE VARCHAR(50)
- COUNTRY VARCHAR(50)

orders
- ORDERNUMBER INT(11)
- ORDERDATE DATE
- STATUS VARCHAR(15)
- CUSTOMERNUMBER INT(11)

orderdetails
- ORDERNUMBER INT(11)
- ORDERLINENUMBER SMALLINT(6)
- PRODUCTCODE VARCHAR(50)
- QUANTITYORDERED INT(11)
- PRICEEACH DECIMAL(17,0)

products
- PRODUCTCODE VARCHAR(50)
- PRODUCTNAME VARCHAR(70)
- PRODUCTLINE VARCHAR(50)
- PRODUCTVENDOR VARCHAR(50)



dim_customer
- customerid
- city
- country

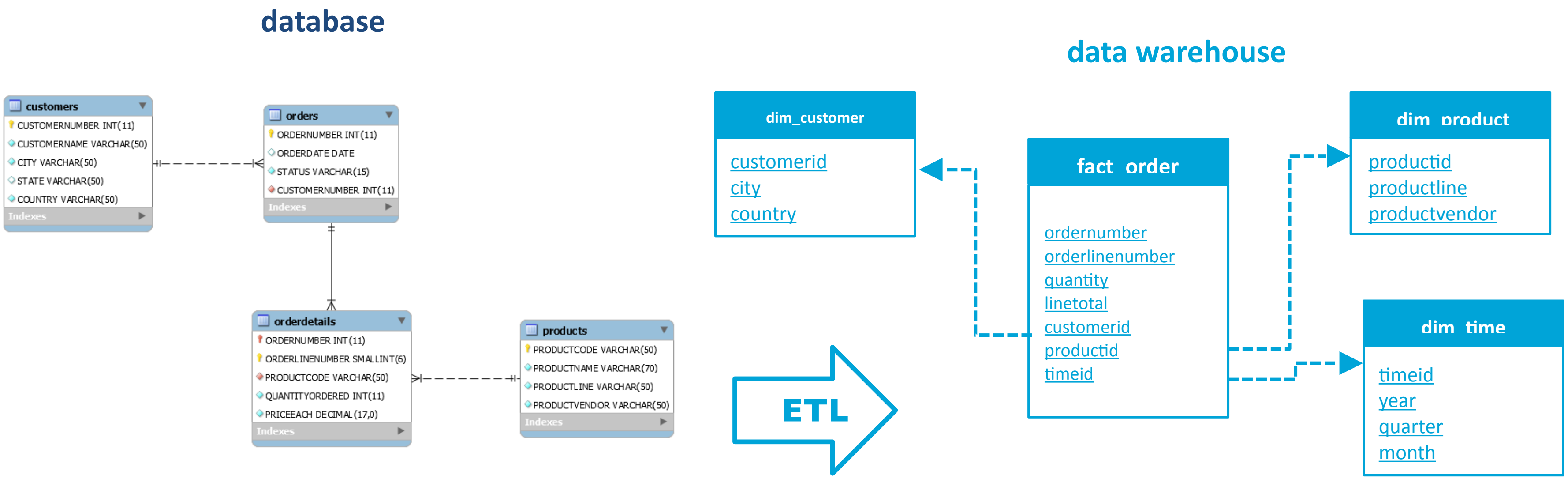
fact order
- ordernumber
- orderlinenumber
- quantity
- linetotal
- customerid
- productid
- timeid

dim product
- productid
- productline
- productvendor

dim time
- timeid
- year
- quarter
- month

Introduction

- How to build a data warehouse
  - this usually involves
    - one transformation for each dimension table
    - one transformation for the fact table
    - a job that runs all transformations in the correct sequence



database

customers
CUSTOMERNUMBER INT(11)
CUSTOMERNAME VARCHAR(50)
CITY VARCHAR(50)
STATE VARCHAR(50)
COUNTRY VARCHAR(50)
Indexes

orders
ORDERNUMBER INT(11)
ORDERDATE DATE
STATUS VARCHAR(15)
CUSTOMERNUMBER INT(11)
Indexes

orderdetails
ORDERNUMBER INT(11)
ORDERLINENUMBER SMALLINT(6)
PRODUCTCODE VARCHAR(50)
QUANTITYORDERED INT(11)
PRICEEACH DECIMAL(17,0)
Indexes

products
PRODUCTCODE VARCHAR(50)
PRODUCTNAME VARCHAR(70)
PRODUCTLINE VARCHAR(50)
PRODUCTVENDOR VARCHAR(50)
Indexes

ETL

data warehouse

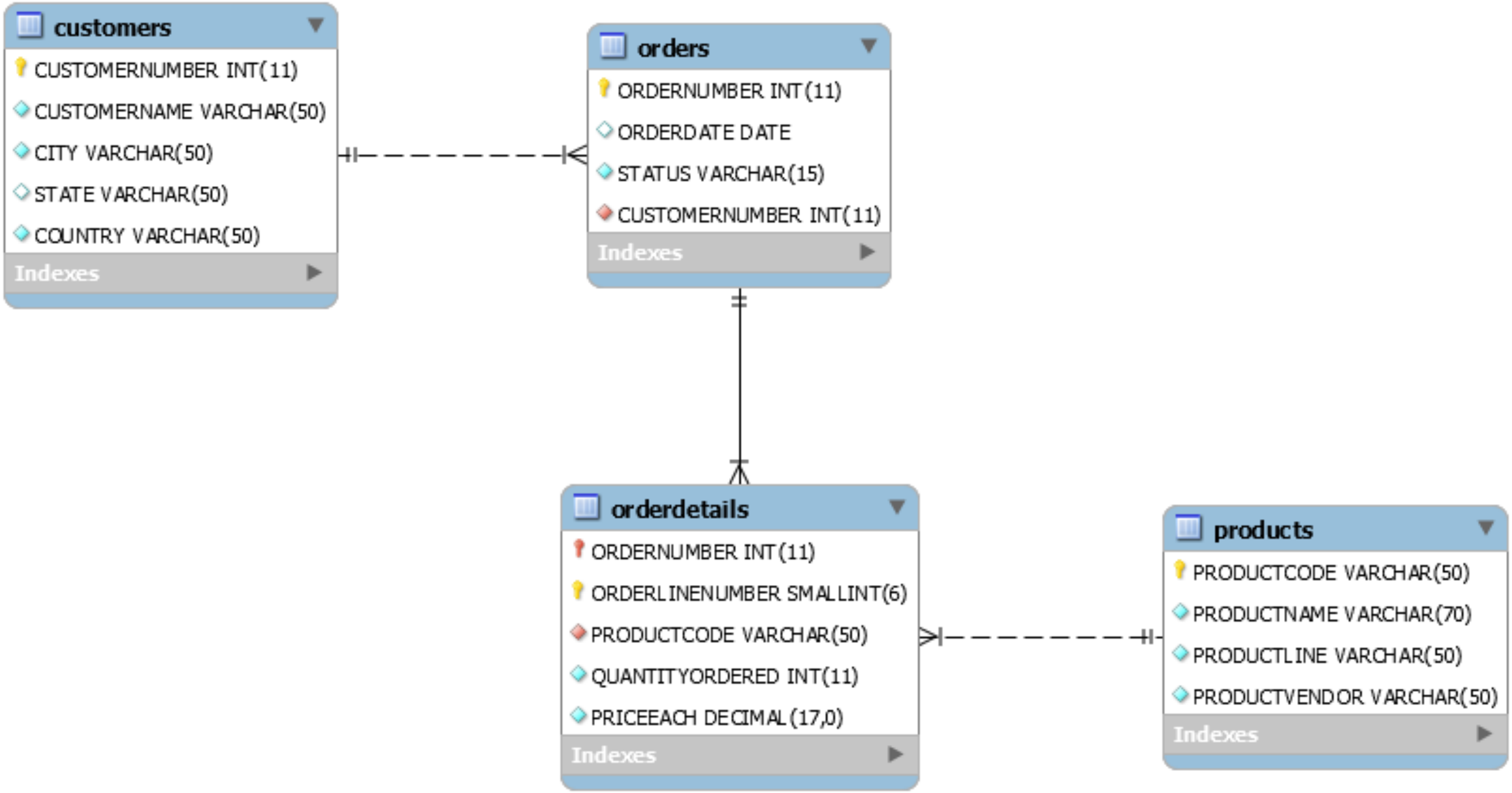
dim_customer
customerid
city
country

fact order
ordernumber
orderlinenumber
quantity
linetotal
customerid
productid
timeid

dim product
productid
productline
productvendor

dim time
timeid
year
quarter
month

Database



customers

CUSTOMERNUMBER INT(11)

CUSTOMERNAME VARCHAR(50)

CITY VARCHAR(50)

STATE VARCHAR(50)

COUNTRY VARCHAR(50)

orders

ORDERNUMBER INT(11)

ORDERDATE DATE

STATUS VARCHAR(15)

CUSTOMERNUMBER INT(11)

orderdetails

ORDERNUMBER INT(11)

ORDERLINENUMBER SMALLINT(6)

PRODUCTCODE VARCHAR(50)

QUANTITYORDERED INT(11)

PRICEEACH DECIMAL(17,0)

products

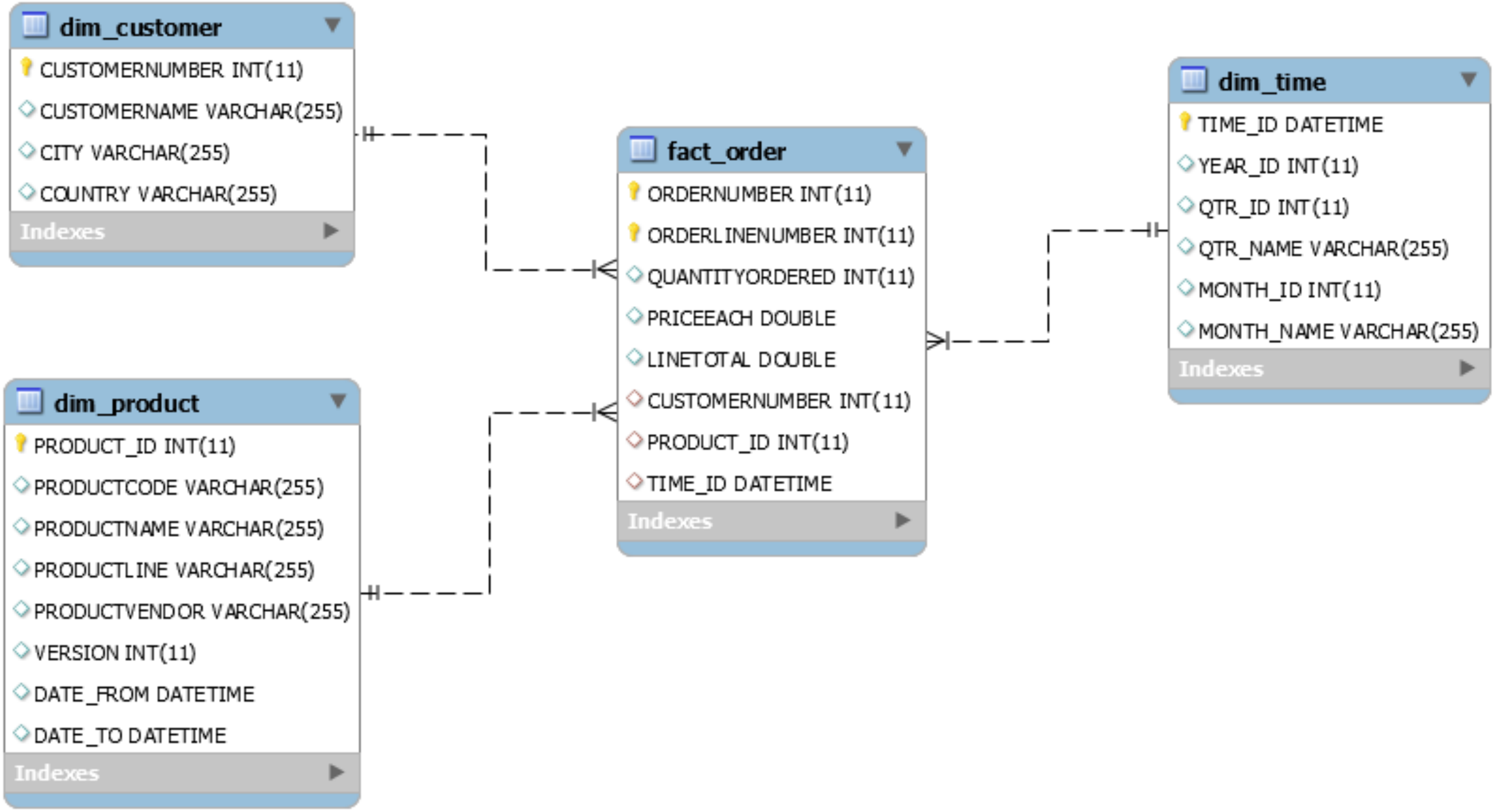
PRODUCTCODE VARCHAR(50)

PRODUCTNAME VARCHAR(70)

PRODUCTLINE VARCHAR(50)

PRODUCTVENDOR VARCHAR(50)

Data warehouse



dim_customer

CUSTOMERNUMBER INT(11)
CUSTOMERNAME VARCHAR(255)
CITY VARCHAR(255)
COUNTRY VARCHAR(255)

dim_product

PRODUCT_ID INT(11)
PRODUCTCODE VARCHAR(255)
PRODUCTNAME VARCHAR(255)
PRODUCTLINE VARCHAR(255)
PRODUCTVENDOR VARCHAR(255)
VERSION INT(11)
DATE_FROM DATETIME
DATE_TO DATETIME

fact_order

ORDERNUMBER INT(11)
ORDERLINENUMBER INT(11)
QUANTITYORDERED INT(11)
PRICEEACH DOUBLE
LINETOTAL DOUBLE
CUSTOMERNUMBER INT(11)
PRODUCT_ID INT(11)
TIME_ID DATETIME

dim_time

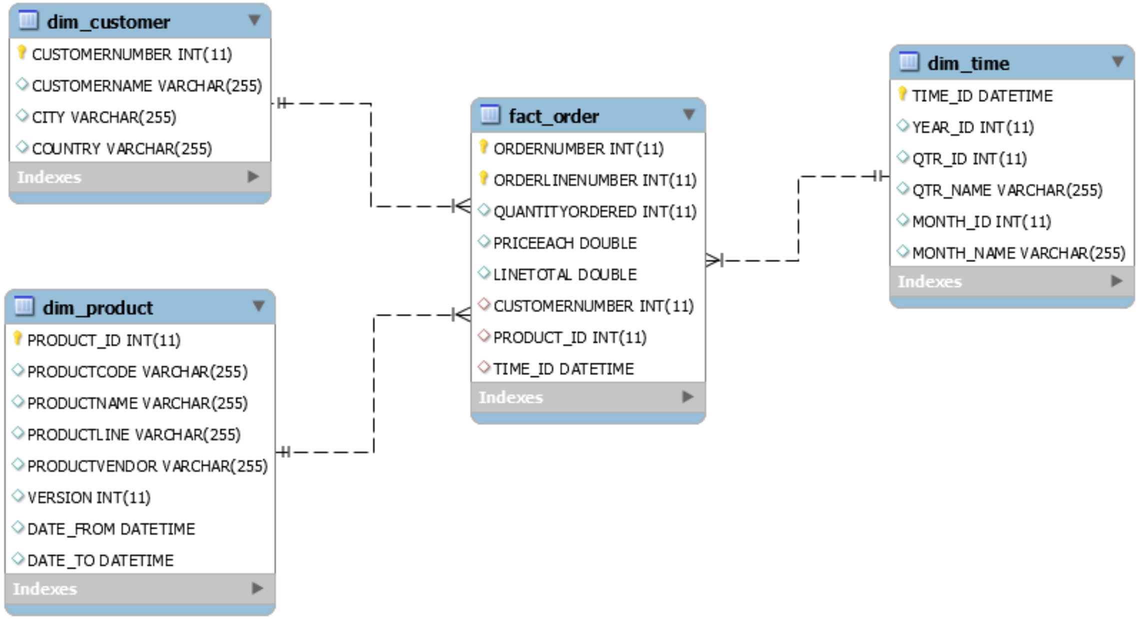
TIME_ID DATETIME
YEAR_ID INT(11)
QTR_ID INT(11)
QTR_NAME VARCHAR(255)
MONTH_ID INT(11)
MONTH_NAME VARCHAR(255)

Data warehouse

- Some notes on this example
  - ⚠ **dim_customer** is simplified, it should have a surrogate key
    - e.g. CUSTOMER_ID of type INT
    - in this case, it was simplified by reusing the natural key
  - ⚠ **dim_time is simplified**, it should have a proper surrogate key
    - e.g. TIME_ID of type INT
    - in this case, it was simplified by using ORDERDATE as key
  - ✓ **dim_product** is a slowly-changing dimension of Type 2
    - surrogate key is PRODUCT_ID of type INT
    - includes version field and validity interval (date fields)

Fact and dimension tables

- Building the data warehouse
  - Star schema with fact table and dimension tables
  - Fact table has FKs to dimension tables
    - Therefore, dimension tables must be populated first



dim_customer
- CUSTOMERNUMBER INT(11)
- CUSTOMERNAME VARCHAR(255)
- CITY VARCHAR(255)
- COUNTRY VARCHAR(255)
- Indexes

fact_order
- ORDERNUMBER INT(11)
- ORDERLINENUMBER INT(11)
- QUANTITYORDERED INT(11)
- PRICEEACH DOUBLE
- LINETOTAL DOUBLE
- CUSTOMERNUMBER INT(11)
- PRODUCT_ID INT(11)
- TIME_ID DATETIME
- Indexes

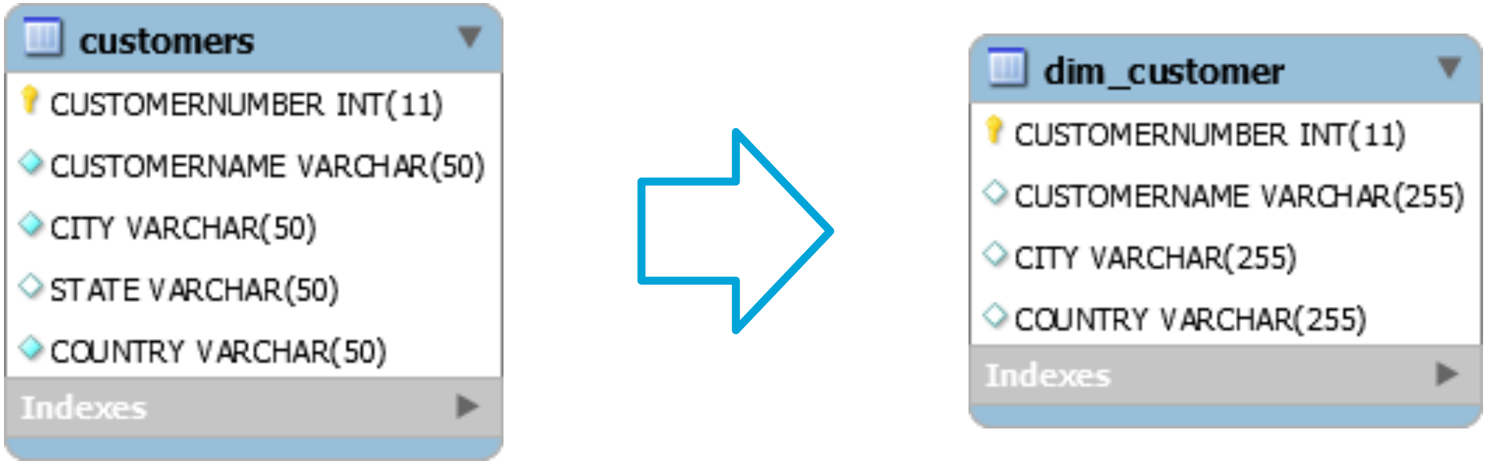
dim_time
- TIME_ID DATETIME
- YEAR_ID INT(11)
- QTR_ID INT(11)
- QTR_NAME VARCHAR(255)
- MONTH_ID INT(11)
- MONTH_NAME VARCHAR(255)
- Indexes

dim_product
- PRODUCT_ID INT(11)
- PRODUCTCODE VARCHAR(255)
- PRODUCTNAME VARCHAR(255)
- PRODUCTLINE VARCHAR(255)
- PRODUCTVENDOR VARCHAR(255)
- VERSION INT(11)
- DATE_FROM DATETIME
- DATE_TO DATETIME
- Indexes

Loading Dimension Tables

Dimension tables

- The customer dimension
  - customer name, city and country (no state)
  - data comes from customers table
  - same key as customers table (natural key)



customers

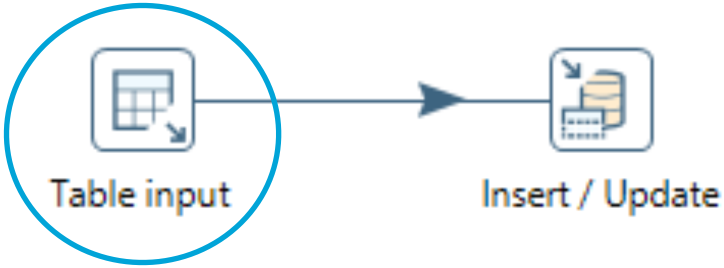
CUSTOMERNUMBER INT(11)
CUSTOMERNAME VARCHAR(50)
CITY VARCHAR(50)
STATE VARCHAR(50)
COUNTRY VARCHAR(50)

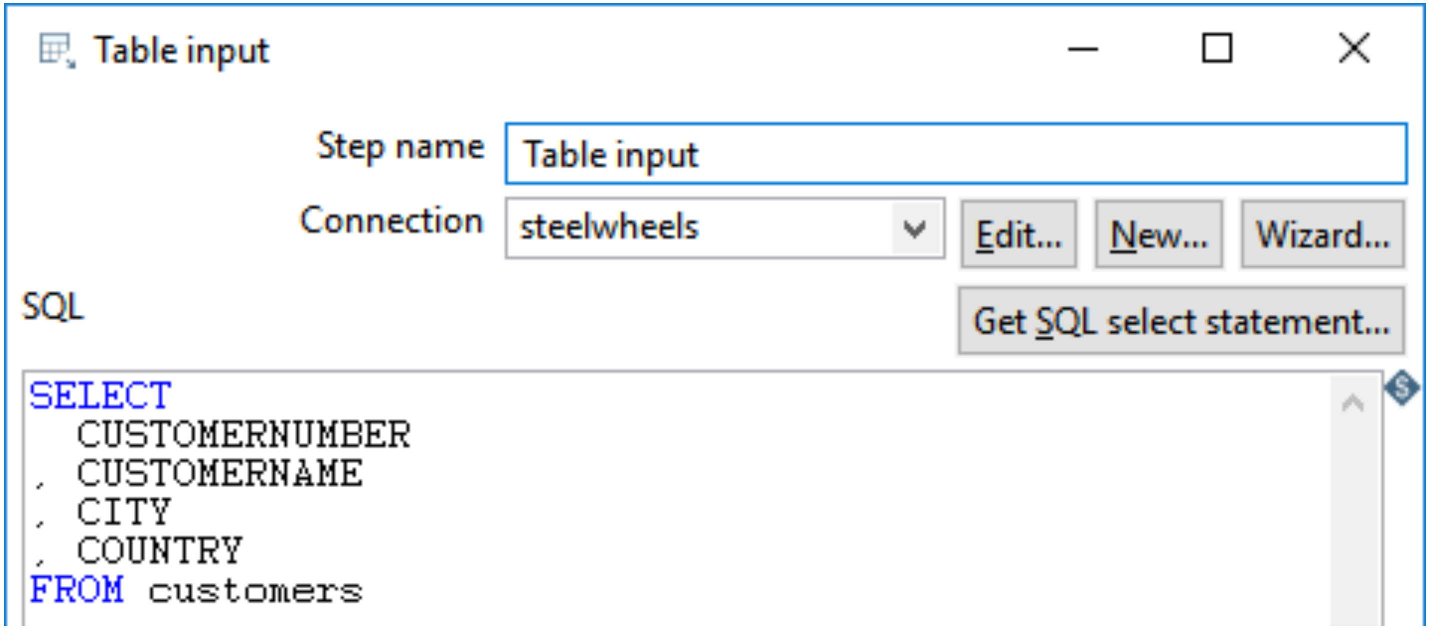
dim_customer

CUSTOMERNUMBER INT(11)
CUSTOMERNAME VARCHAR(255)
CITY VARCHAR(255)
COUNTRY VARCHAR(255)

Dimension tables

- The customer dimension





Step name: Table input

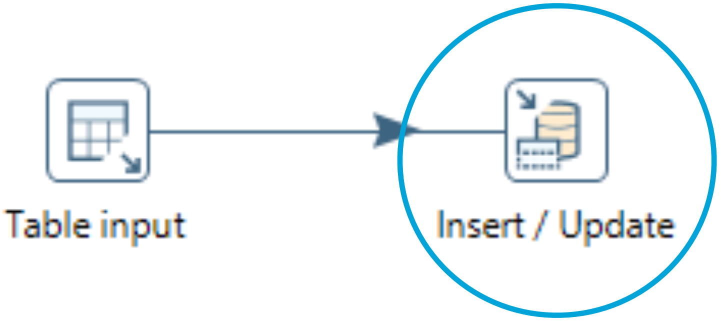
Connection: steelwheels

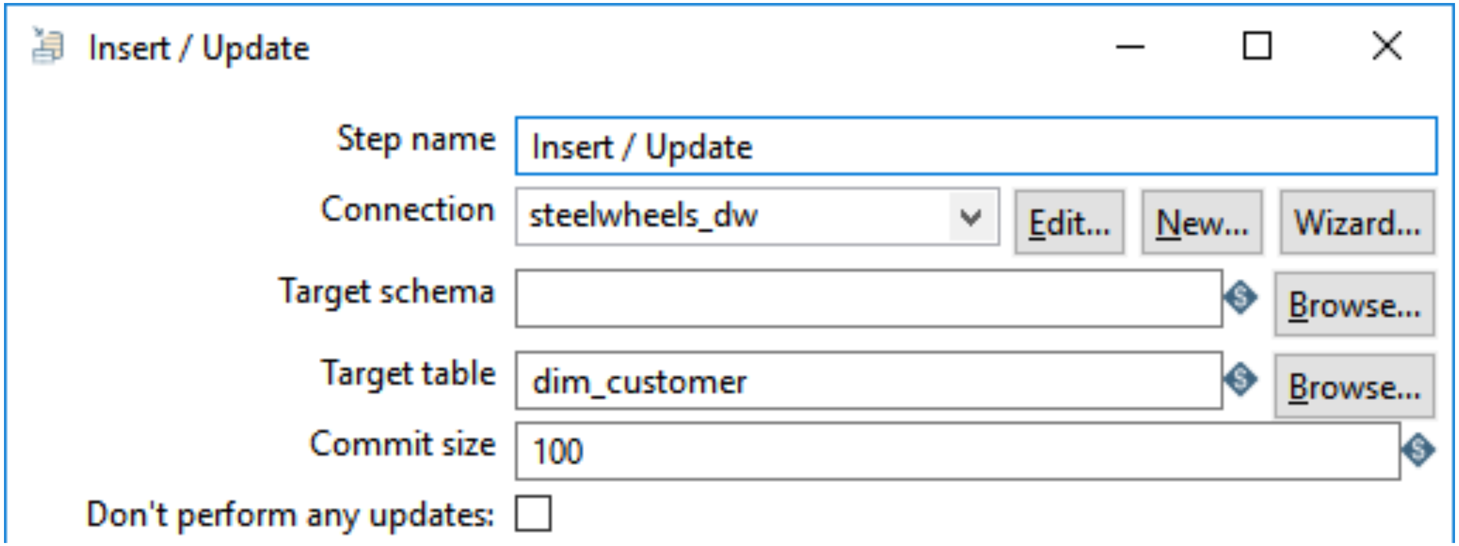
SQL

```
SELECT
  CUSTOMERNUMBER
, CUSTOMERNAME
, CITY
, COUNTRY
FROM customers
```

Dimension tables

- The customer dimension





Step name: Insert / Update

Connection: steelwheels_dw

Target schema:

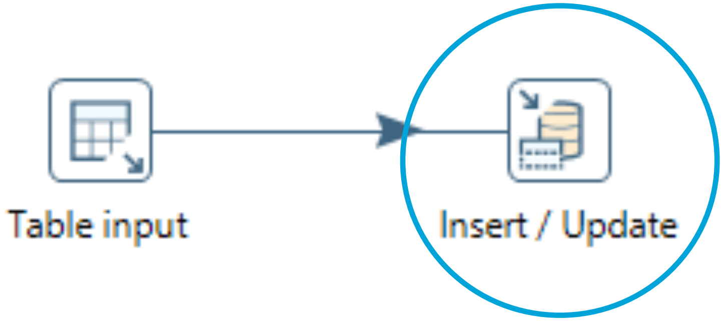
Target table: dim_customer

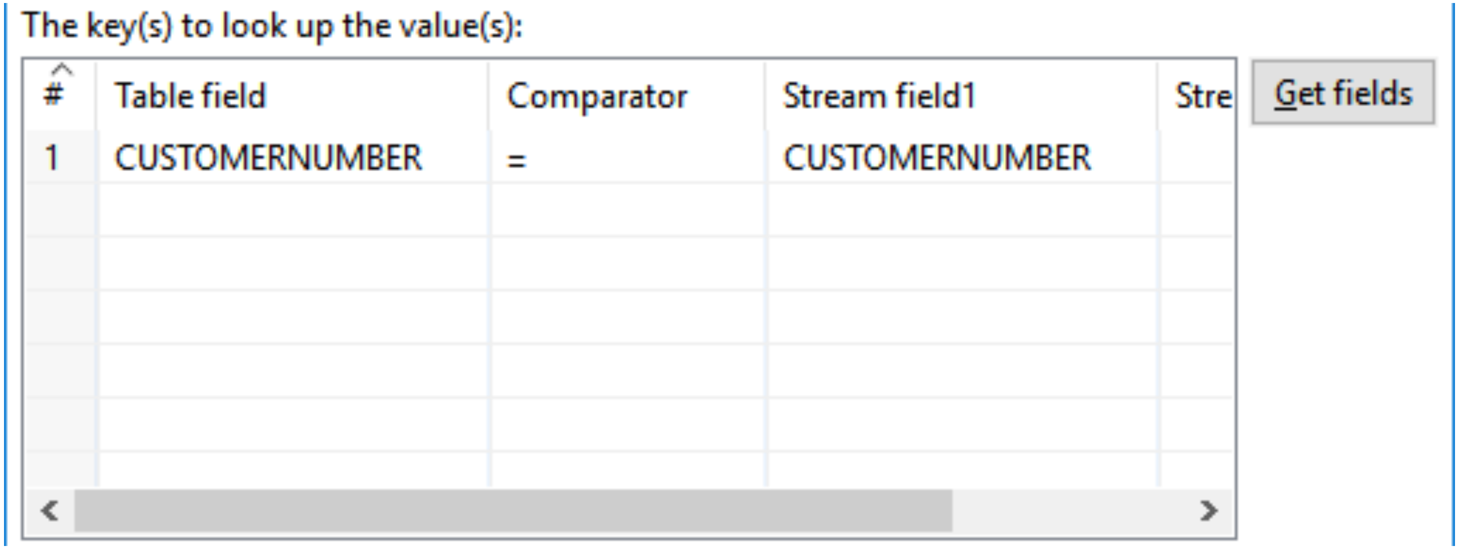
Commit size: 100

Don't perform any updates: □

Dimension tables

- The customer dimension

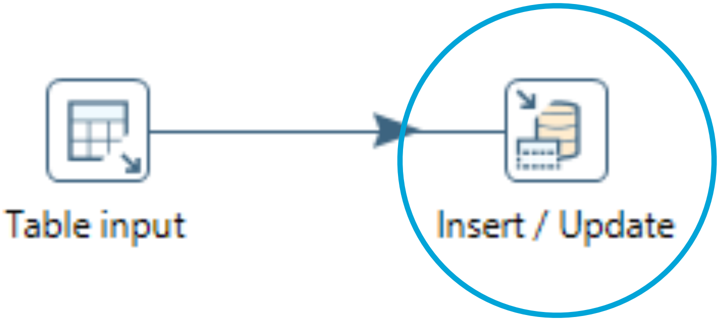


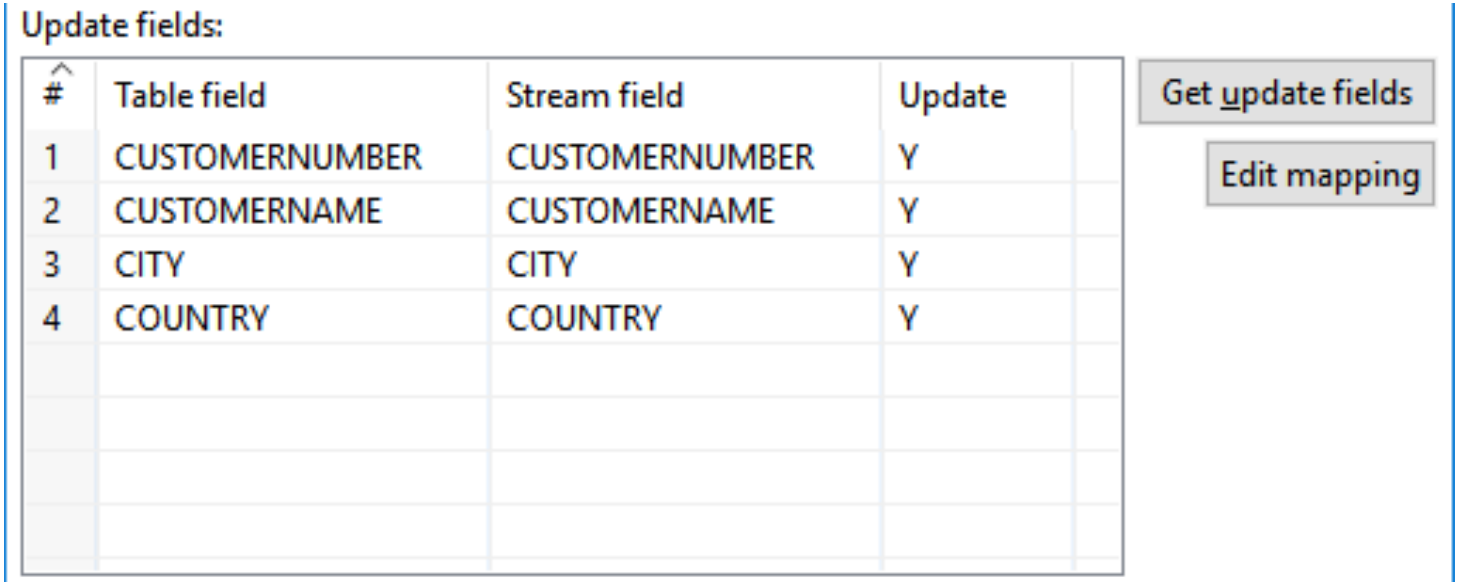

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stre
1	CUSTOMERNUMBER	=	CUSTOMERNUMBER	

Dimension tables

- The customer dimension

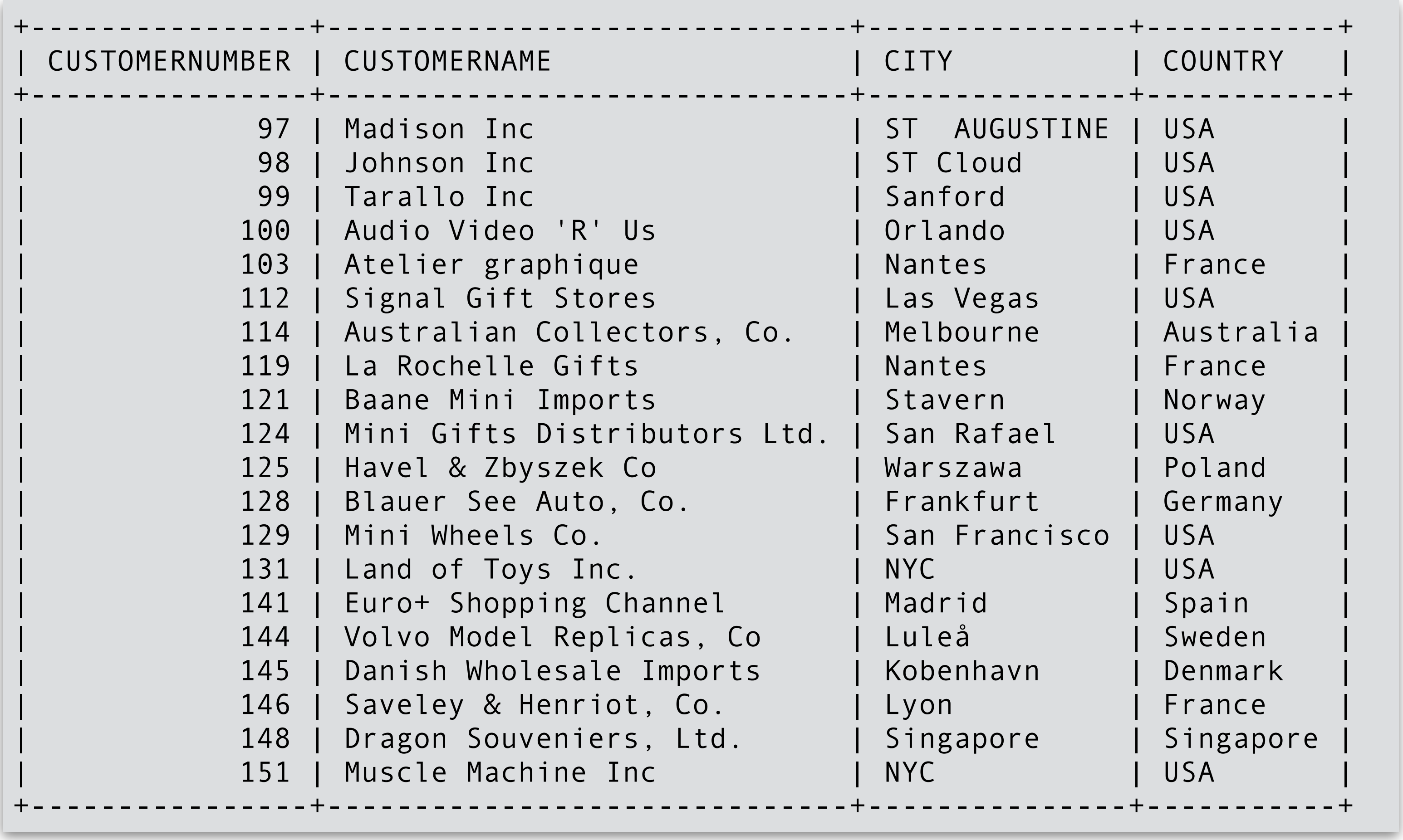



Update fields:

#	Table field	Stream field	Update
1	CUSTOMERNUMBER	CUSTOMERNUMBER	Y
2	CUSTOMERNAME	CUSTOMERNAME	Y
3	CITY	CITY	Y
4	COUNTRY	COUNTRY	Y

Dimension tables

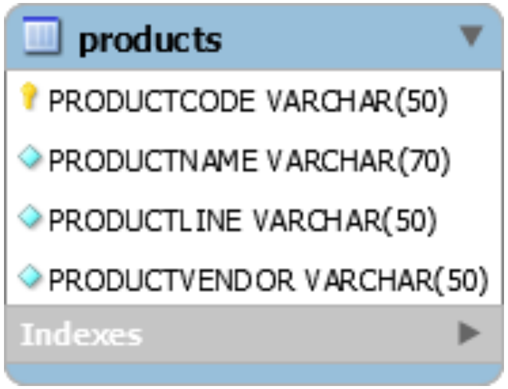
- The customer dimension



```
+----------------+------------------------------+---------------+-----------+
| CUSTOMERNUMBER | CUSTOMERNAME                 | CITY          | COUNTRY   |
+----------------+------------------------------+---------------+-----------+
|             97 | Madison Inc                  | ST  AUGUSTINE | USA       |
|             98 | Johnson Inc                  | ST Cloud      | USA       |
|             99 | Tarallo Inc                  | Sanford       | USA       |
|            100 | Audio Video 'R' Us           | Orlando       | USA       |
|            103 | Atelier graphique            | Nantes        | France    |
|            112 | Signal Gift Stores           | Las Vegas     | USA       |
|            114 | Australian Collectors, Co.   | Melbourne     | Australia |
|            119 | La Rochelle Gifts            | Nantes        | France    |
|            121 | Baane Mini Imports           | Stavern       | Norway    |
|            124 | Mini Gifts Distributors Ltd. | San Rafael    | USA       |
|            125 | Havel & Zbyszek Co           | Warszawa      | Poland    |
|            128 | Blauer See Auto, Co.         | Frankfurt     | Germany   |
|            129 | Mini Wheels Co.              | San Francisco | USA       |
|            131 | Land of Toys Inc.            | NYC           | USA       |
|            141 | Euro+ Shopping Channel       | Madrid        | Spain     |
|            144 | Volvo Model Replicas, Co     | Luleå         | Sweden    |
|            145 | Danish Wholesale Imports     | Kobenhavn     | Denmark   |
|            146 | Saveley & Henriot, Co.       | Lyon          | France    |
|            148 | Dragon Souveniers, Ltd.      | Singapore     | Singapore |
|            151 | Muscle Machine Inc           | NYC           | USA       |
+----------------+------------------------------+---------------+-----------+
```

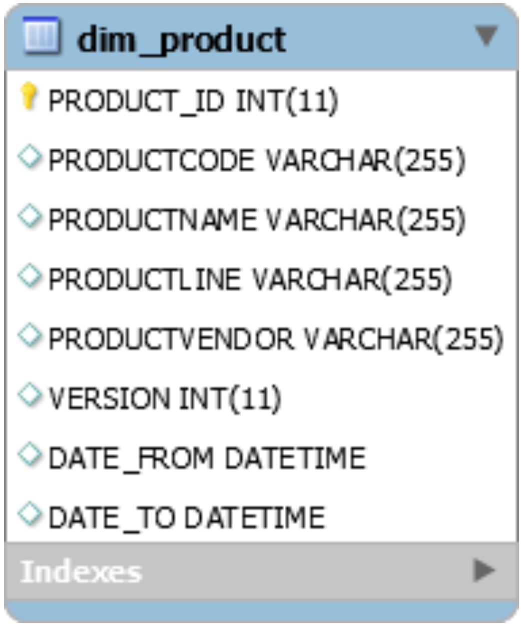
Dimension tables

- The product dimension
  - product name, line, vendor
  - data comes from products table
  - key is not product code but product id (surrogate key)
  - slowly-changing dimension (Type 2)



products

- PRODUCTCODE VARCHAR(50)
- PRODUCTNAME VARCHAR(70)
- PRODUCTLINE VARCHAR(50)
- PRODUCTVENDOR VARCHAR(50)

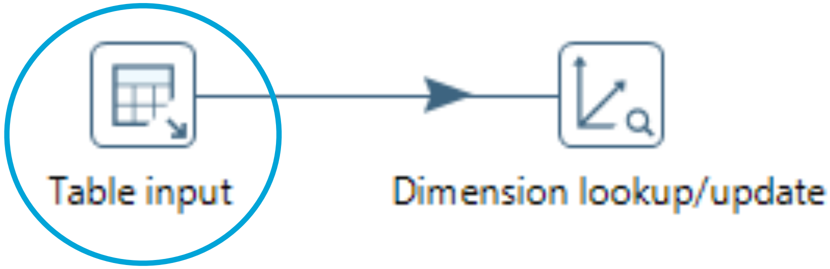


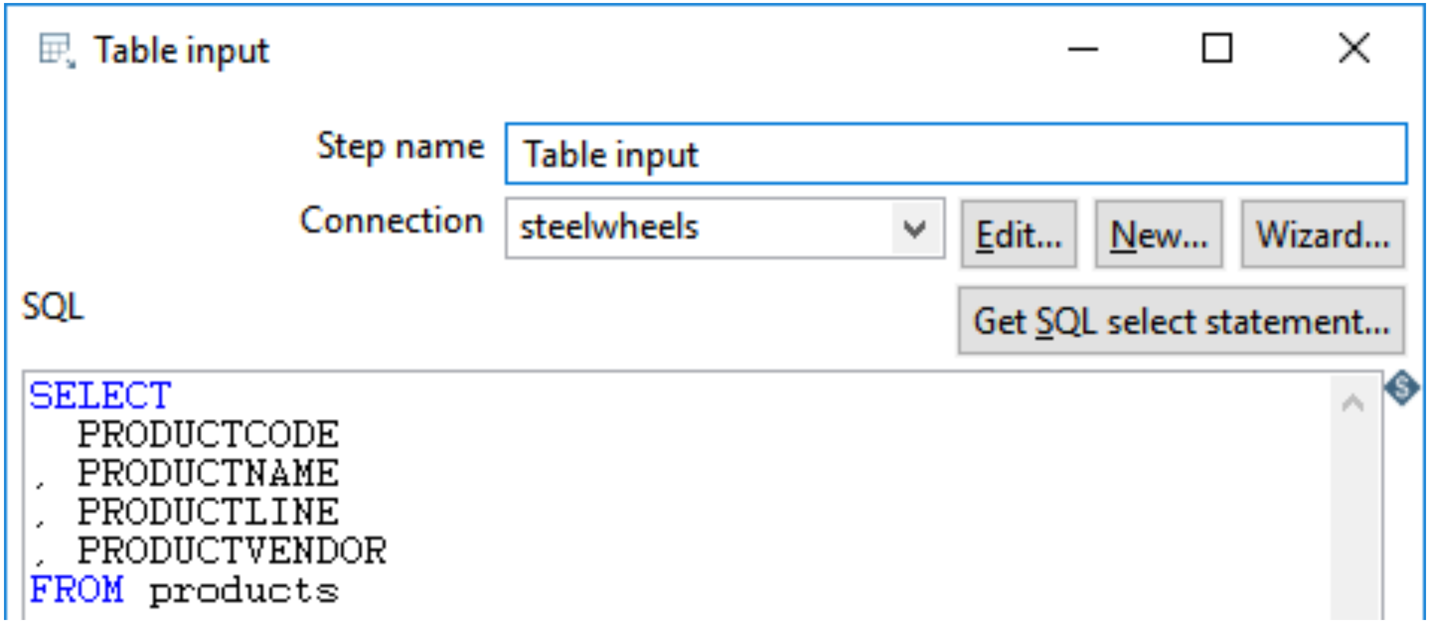
dim_product

- PRODUCT_ID INT(11)
- PRODUCTCODE VARCHAR(255)
- PRODUCTNAME VARCHAR(255)
- PRODUCTLINE VARCHAR(255)
- PRODUCTVENDOR VARCHAR(255)
- VERSION INT(11)
- DATE_FROM DATETIME
- DATE_TO DATETIME

Dimension tables

- The product dimension





Step name: Table input

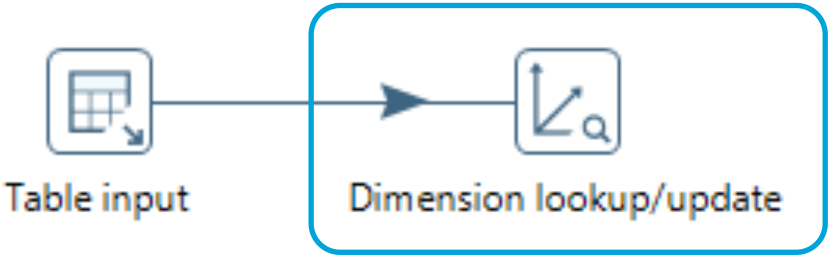
Connection: steelwheels

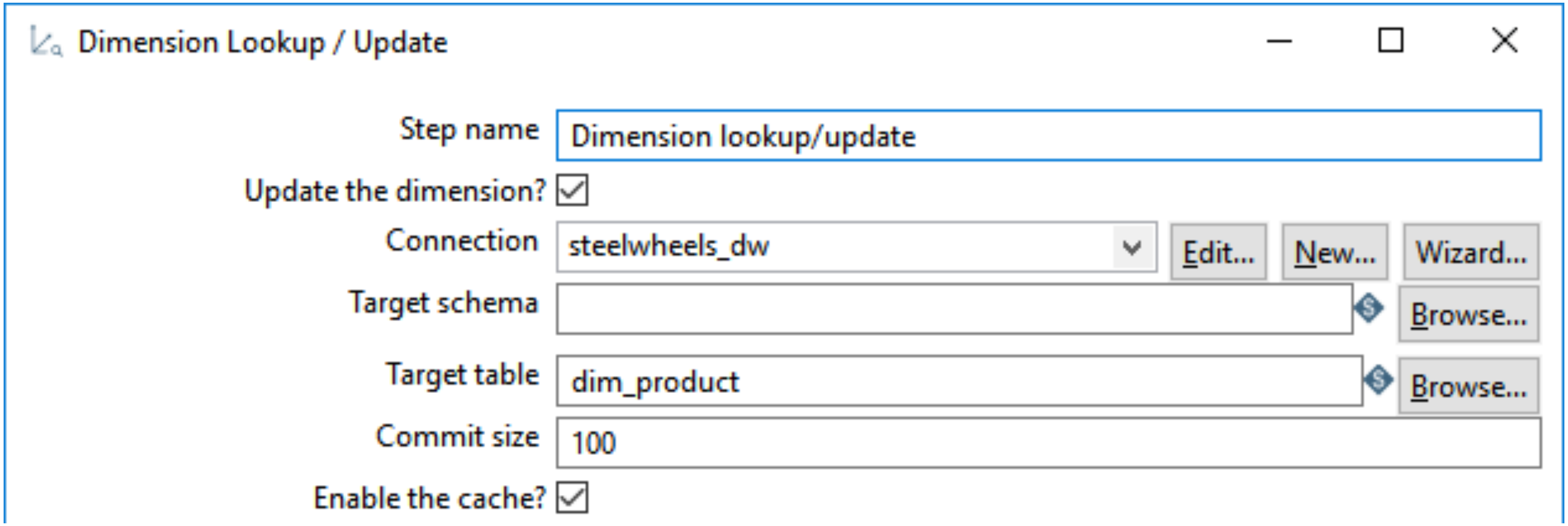
SQL

```
SELECT
   PRODUCTCODE
,  PRODUCTNAME
,  PRODUCTLINE
,  PRODUCTVENDOR
FROM products
```

Dimension tables

- The product dimension





Step name: Dimension lookup/update

Update the dimension? ☑

Connection: steelwheels_dw

Target schema:

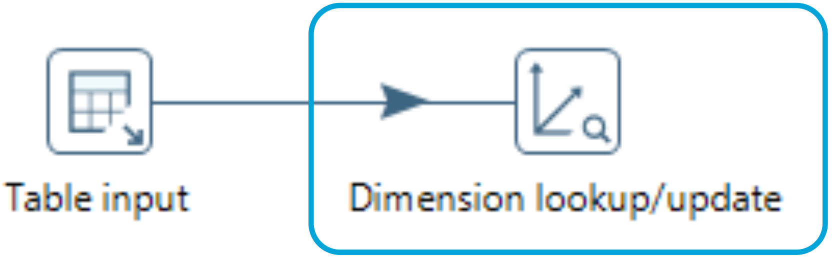
Target table: dim_product

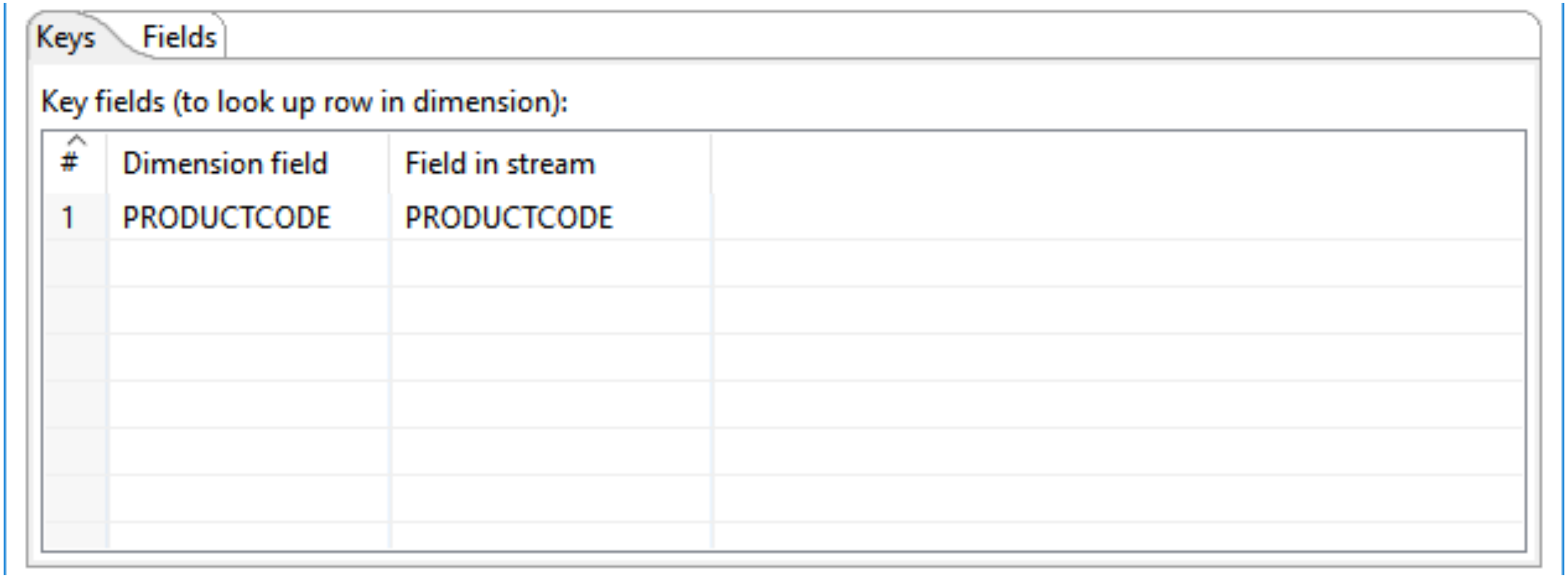
Commit size: 100

Enable the cache? ☑

Dimension tables

- The product dimension



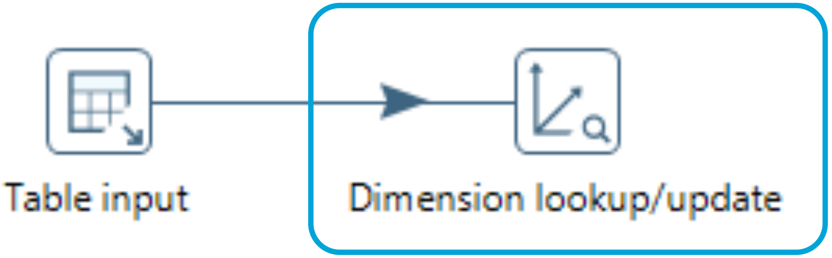


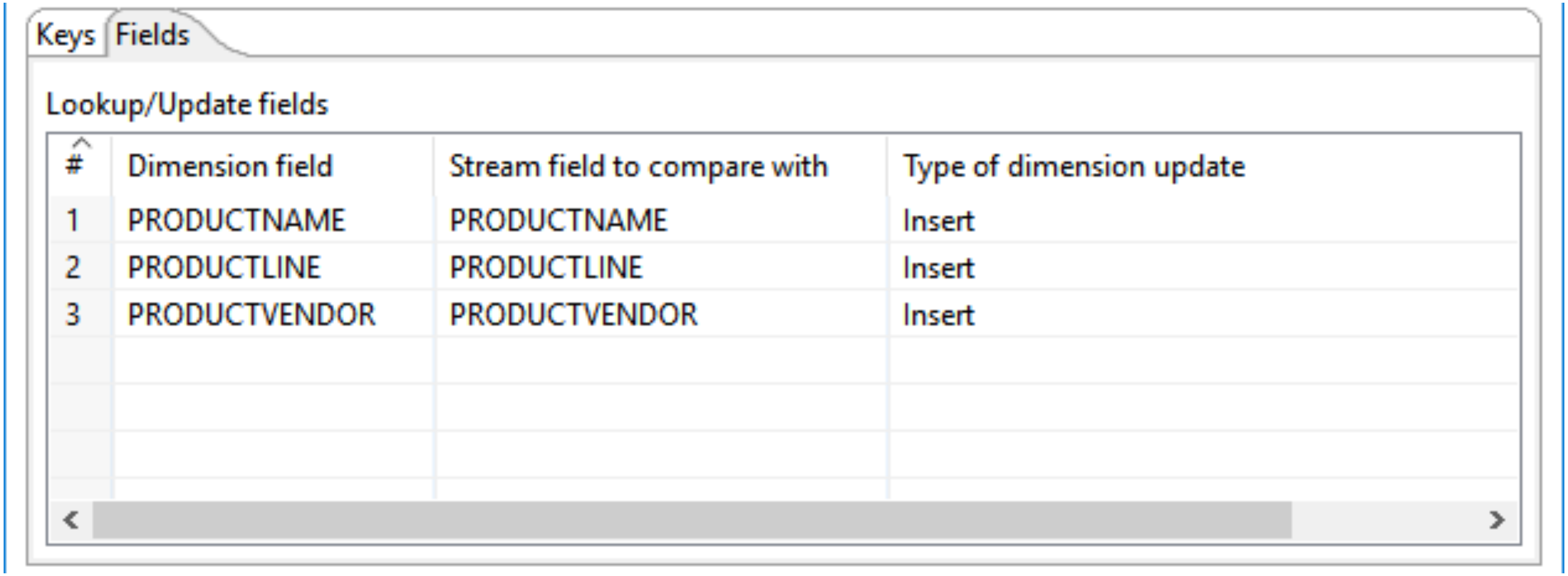
Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	PRODUCTCODE	PRODUCTCODE

Dimension tables

- The product dimension



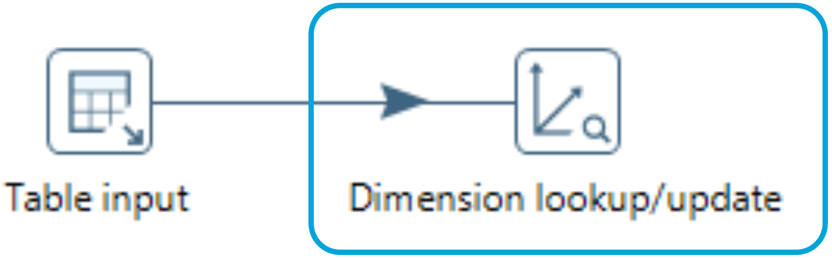


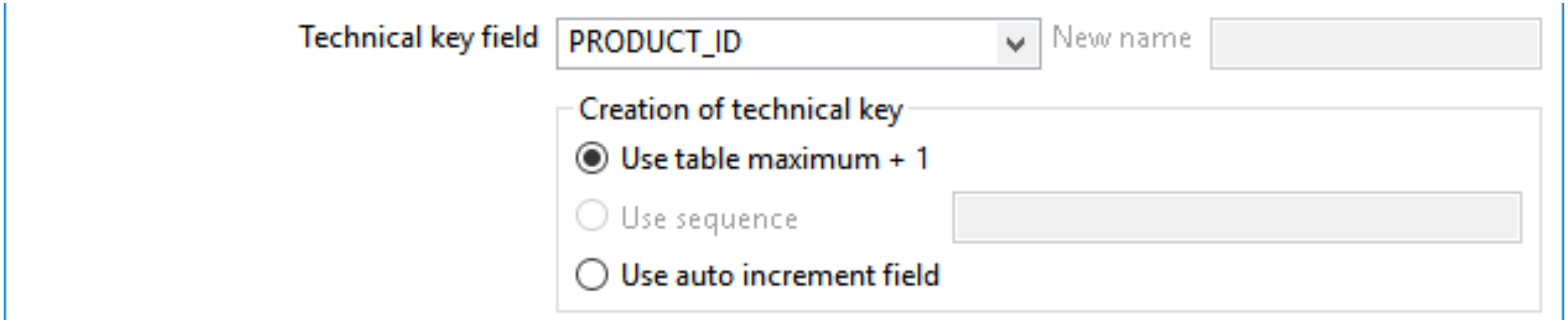
Lookup/Update fields

#	Dimension field	Stream field to compare with	Type of dimension update
1	PRODUCTNAME	PRODUCTNAME	Insert
2	PRODUCTLINE	PRODUCTLINE	Insert
3	PRODUCTVENDOR	PRODUCTVENDOR	Insert

Dimension tables

- The product dimension





Technical key field: PRODUCT_ID

New name:

Creation of technical key

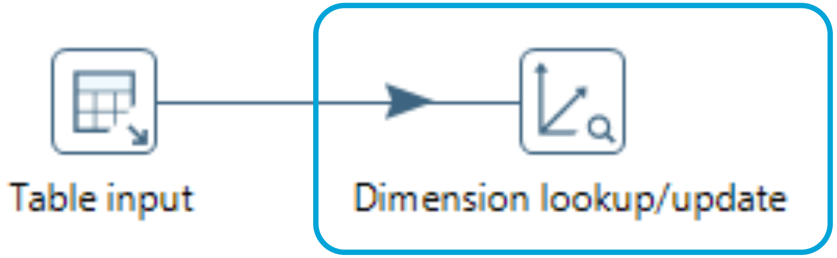
● Use table maximum + 1

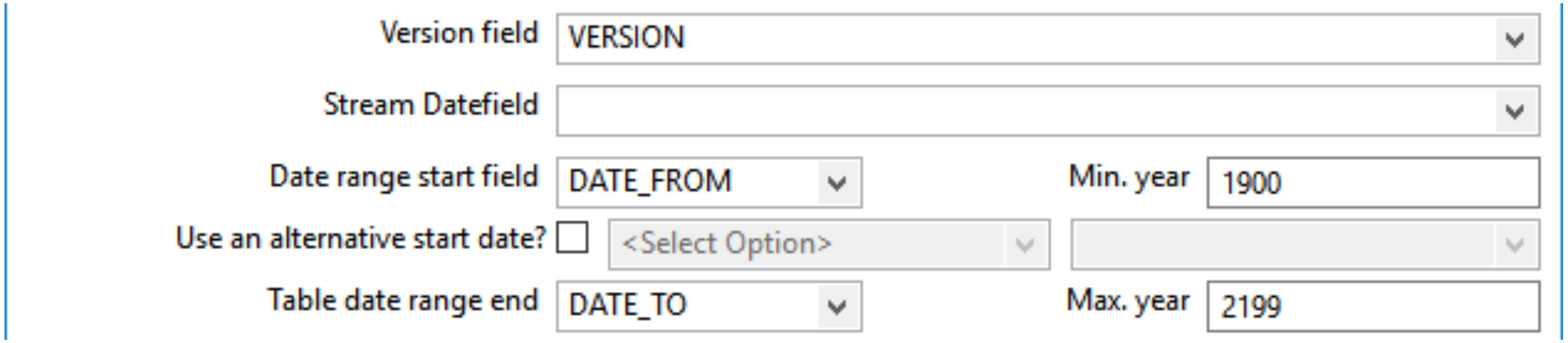
○ Use sequence

○ Use auto increment field

Dimension tables

- The product dimension

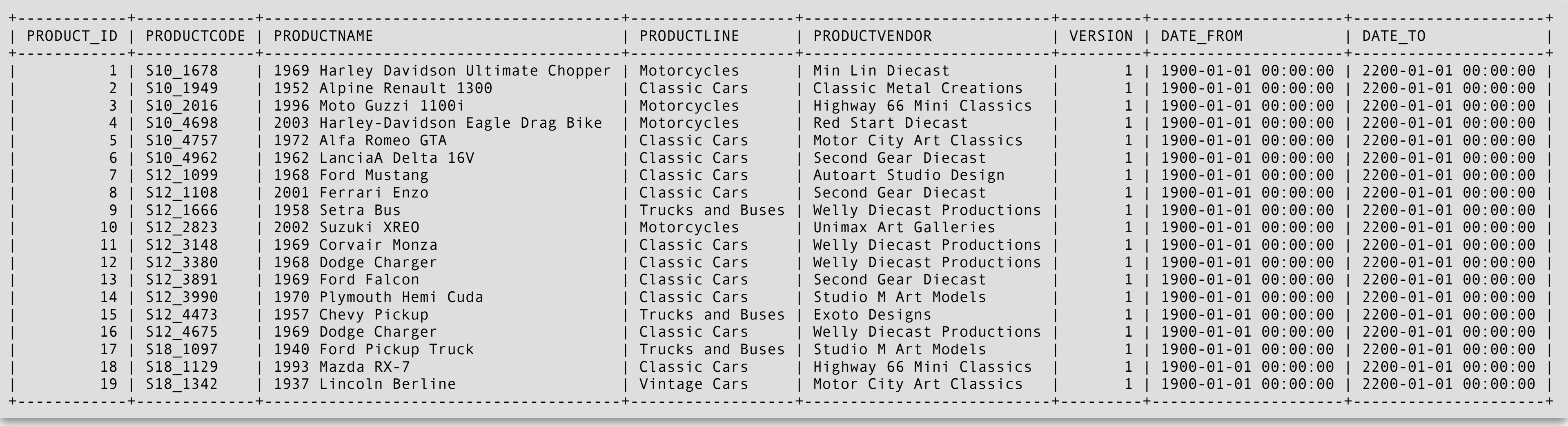




Version field	VERSION		
Stream Datefield			
Date range start field	DATE_FROM	Min. year	1900
Use an alternative start date? □	<Select Option>		
Table date range end	DATE_TO	Max. year	2199

Dimension tables

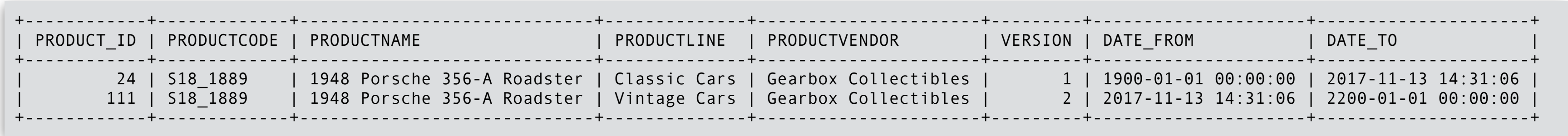
- The product dimension



```
+------------+-------------+---------------------------------------+------------------+---------------------------+---------+---------------------+---------------------+
| PRODUCT_ID | PRODUCTCODE | PRODUCTNAME                           | PRODUCTLINE      | PRODUCTVENDOR             | VERSION | DATE_FROM           | DATE_TO             |
+------------+-------------+---------------------------------------+------------------+---------------------------+---------+---------------------+---------------------+
|          1 | S10_1678    | 1969 Harley Davidson Ultimate Chopper | Motorcycles      | Min Lin Diecast           |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          2 | S10_1949    | 1952 Alpine Renault 1300              | Classic Cars     | Classic Metal Creations   |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          3 | S10_2016    | 1996 Moto Guzzi 1100i                 | Motorcycles      | Highway 66 Mini Classics  |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          4 | S10_4698    | 2003 Harley-Davidson Eagle Drag Bike  | Motorcycles      | Red Start Diecast         |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          5 | S10_4757    | 1972 Alfa Romeo GTA                   | Classic Cars     | Motor City Art Classics   |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          6 | S10_4962    | 1962 LanciaA Delta 16V                | Classic Cars     | Second Gear Diecast       |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          7 | S12_1099    | 1968 Ford Mustang                     | Classic Cars     | Autoart Studio Design     |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          8 | S12_1108    | 2001 Ferrari Enzo                     | Classic Cars     | Second Gear Diecast       |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|          9 | S12_1666    | 1958 Setra Bus                        | Trucks and Buses | Welly Diecast Productions |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         10 | S12_2823    | 2002 Suzuki XREO                      | Motorcycles      | Unimax Art Galleries      |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         11 | S12_3148    | 1969 Corvair Monza                    | Classic Cars     | Welly Diecast Productions |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         12 | S12_3380    | 1968 Dodge Charger                    | Classic Cars     | Welly Diecast Productions |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         13 | S12_3891    | 1969 Ford Falcon                      | Classic Cars     | Second Gear Diecast       |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         14 | S12_3990    | 1970 Plymouth Hemi Cuda               | Classic Cars     | Studio M Art Models       |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         15 | S12_4473    | 1957 Chevy Pickup                     | Trucks and Buses | Exoto Designs             |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         16 | S12_4675    | 1969 Dodge Charger                    | Classic Cars     | Welly Diecast Productions |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         17 | S18_1097    | 1940 Ford Pickup Truck                | Trucks and Buses | Studio M Art Models       |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         18 | S18_1129    | 1993 Mazda RX-7                       | Classic Cars     | Highway 66 Mini Classics  |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
|         19 | S18_1342    | 1937 Lincoln Berline                  | Vintage Cars     | Motor City Art Classics   |       1 | 1900-01-01 00:00:00 | 2200-01-01 00:00:00 |
+------------+-------------+---------------------------------------+------------------+---------------------------+---------+---------------------+---------------------+
```

Dimension tables

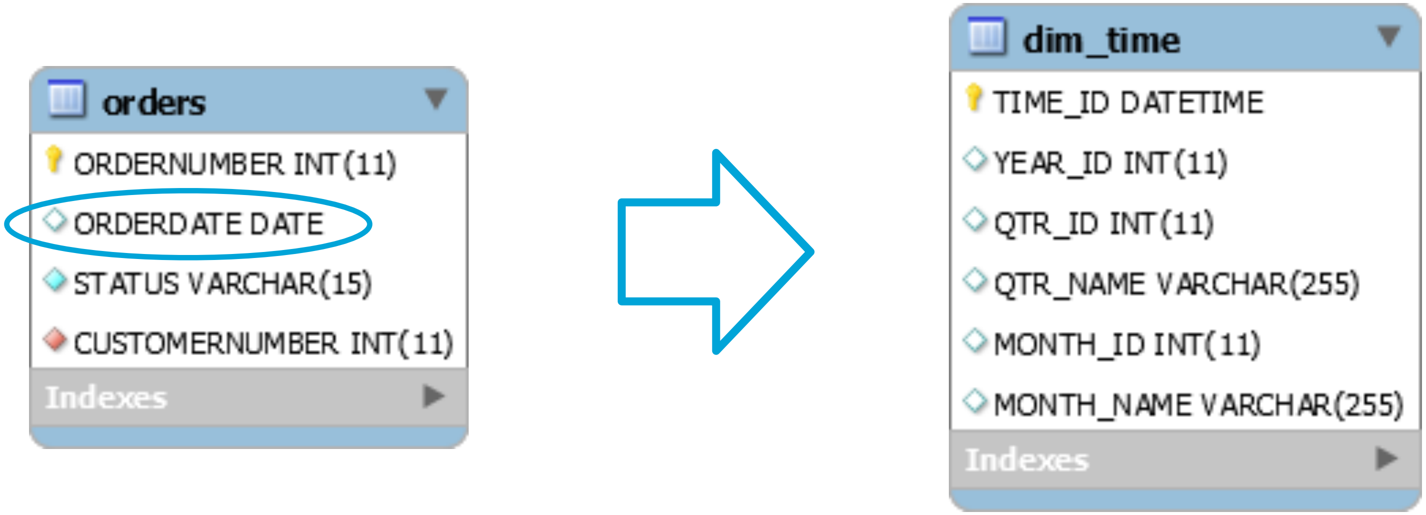
- The product dimension
  - **Testing the slowly changing dimension**
    - Change product line in the original database
    - Run transformation again
    - There should be now two rows for the same product



```
+------------+-------------+-----------------------------+--------------+----------------------+---------+---------------------+---------------------+
| PRODUCT_ID | PRODUCTCODE | PRODUCTNAME                 | PRODUCTLINE  | PRODUCTVENDOR        | VERSION | DATE_FROM           | DATE_TO             |
+------------+-------------+-----------------------------+--------------+----------------------+---------+---------------------+---------------------+
|         24 | S18_1889    | 1948 Porsche 356-A Roadster | Classic Cars | Gearbox Collectibles |       1 | 1900-01-01 00:00:00 | 2017-11-13 14:31:06 |
|        111 | S18_1889    | 1948 Porsche 356-A Roadster | Vintage Cars | Gearbox Collectibles |       2 | 2017-11-13 14:31:06 | 2200-01-01 00:00:00 |
+------------+-------------+-----------------------------+--------------+----------------------+---------+---------------------+---------------------+
```

Dimension tables

- The time dimension
  - Year, quarter, month (id and name for quarter and month)
  - Data comes from order date alone
  - Key is time_id



orders

ORDERNUMBER INT(11)

ORDERDATE DATE

STATUS VARCHAR(15)

CUSTOMERNUMBER INT(11)

Indexes

dim_time

TIME_ID DATETIME

YEAR_ID INT(11)

QTR_ID INT(11)

QTR_NAME VARCHAR(255)

MONTH_ID INT(11)

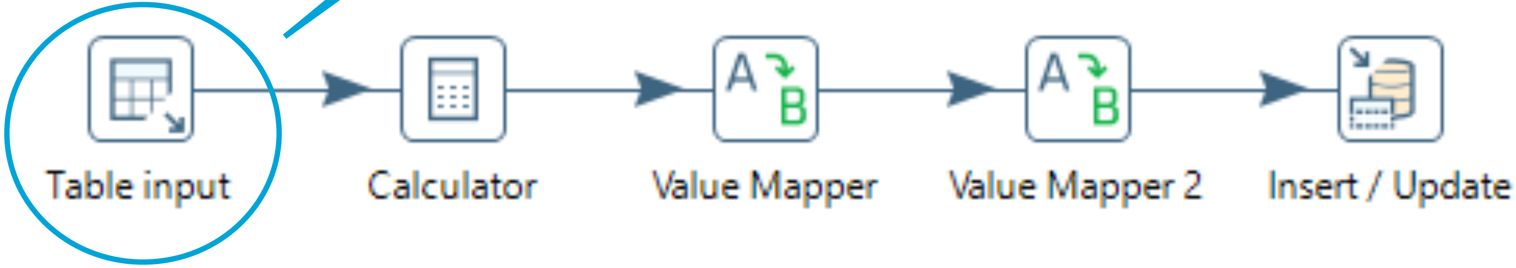
MONTH_NAME VARCHAR(255)

Indexes

Dimension tables

Not the most robust way to do it! WHY?

- The time dimension



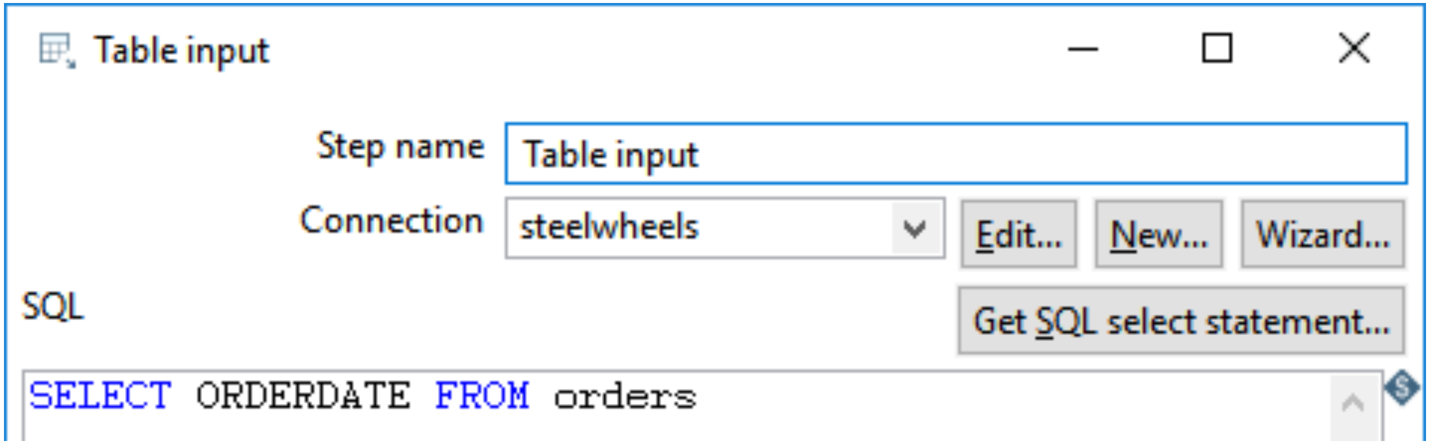


Table input

Step name: Table input

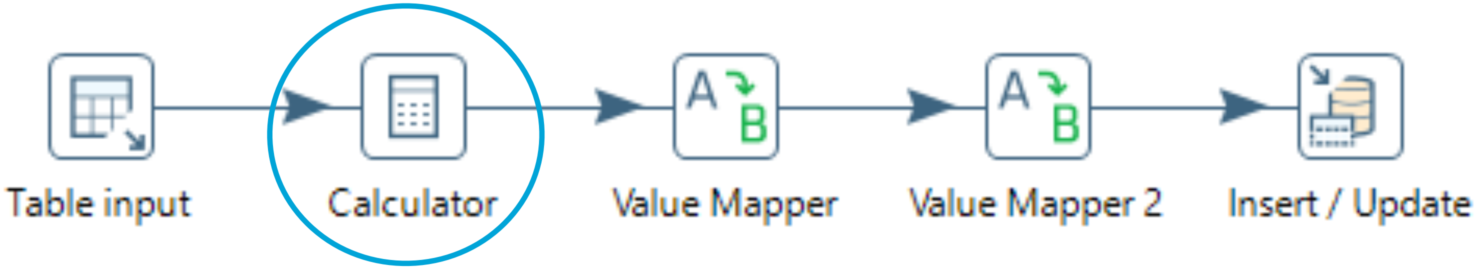
Connection: steelwheels

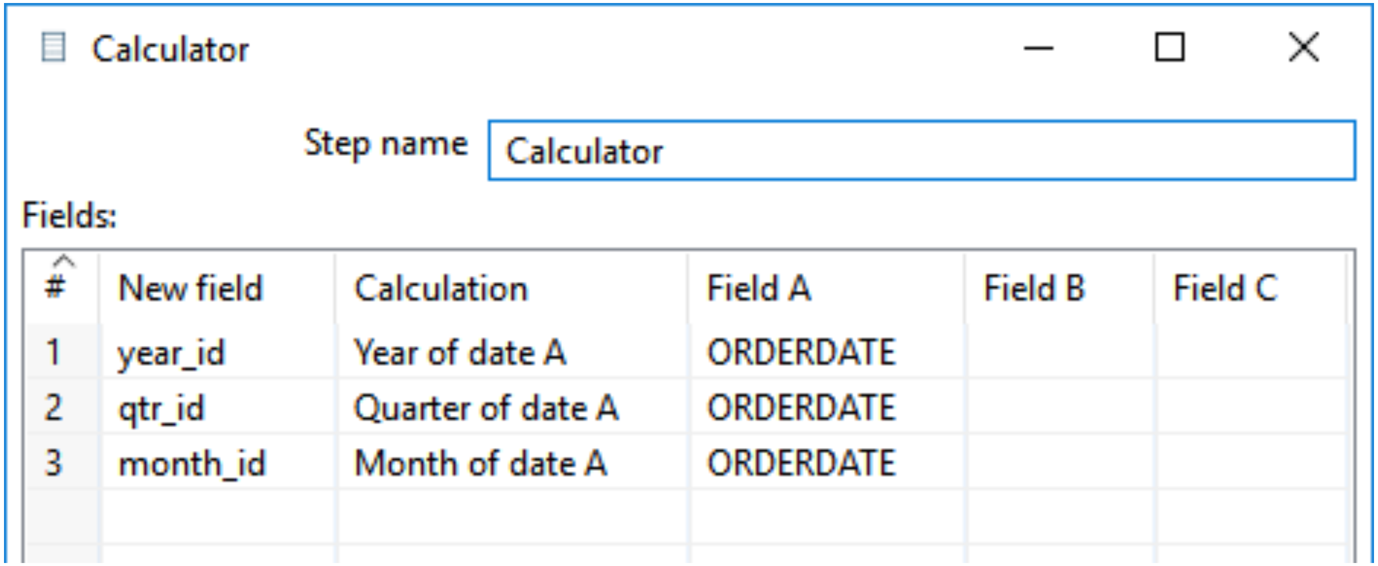
SQL

```
SELECT ORDERDATE FROM orders
```

Dimension tables

- The time dimension




Calculator

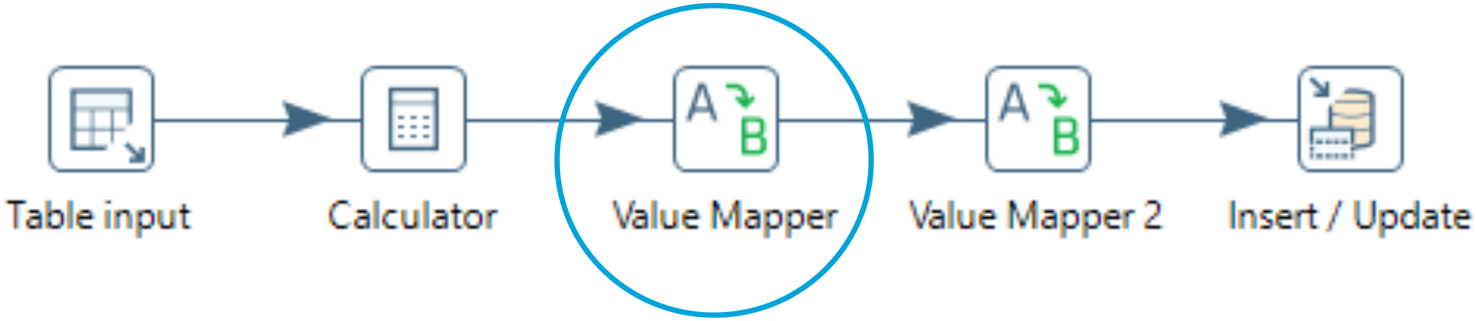
Step name: Calculator

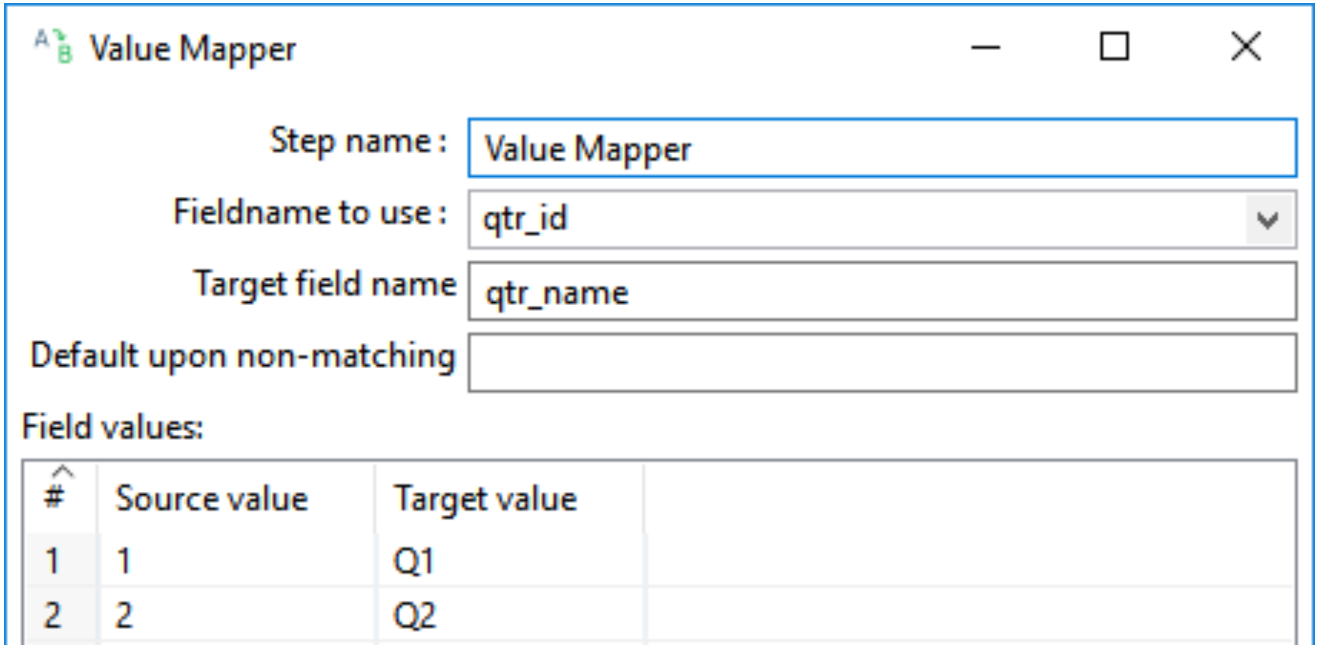
Fields:

#	New field	Calculation	Field A	Field B	Field C
1	year_id	Year of date A	ORDERDATE		
2	qtr_id	Quarter of date A	ORDERDATE		
3	month_id	Month of date A	ORDERDATE		

Dimension tables

- The time dimension





Step name : Value Mapper

Fieldname to use : qtr_id

Target field name qtr_name

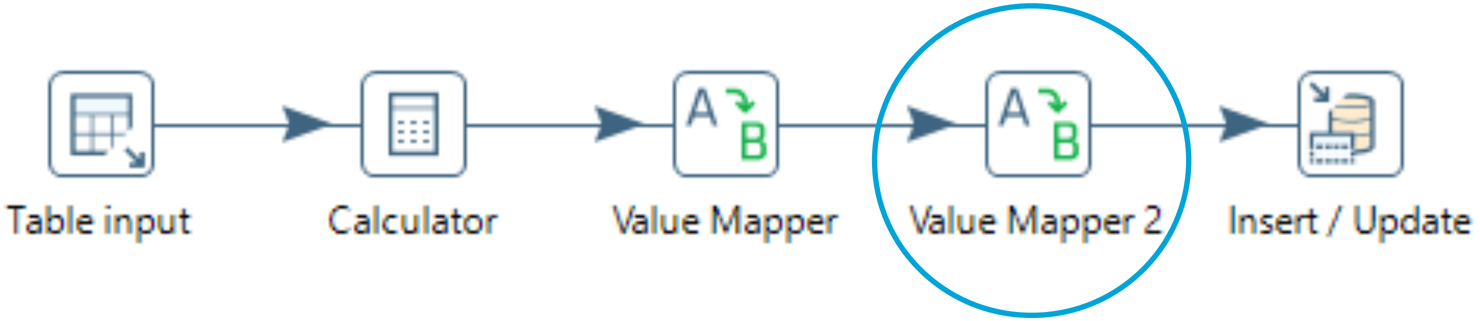
Default upon non-matching

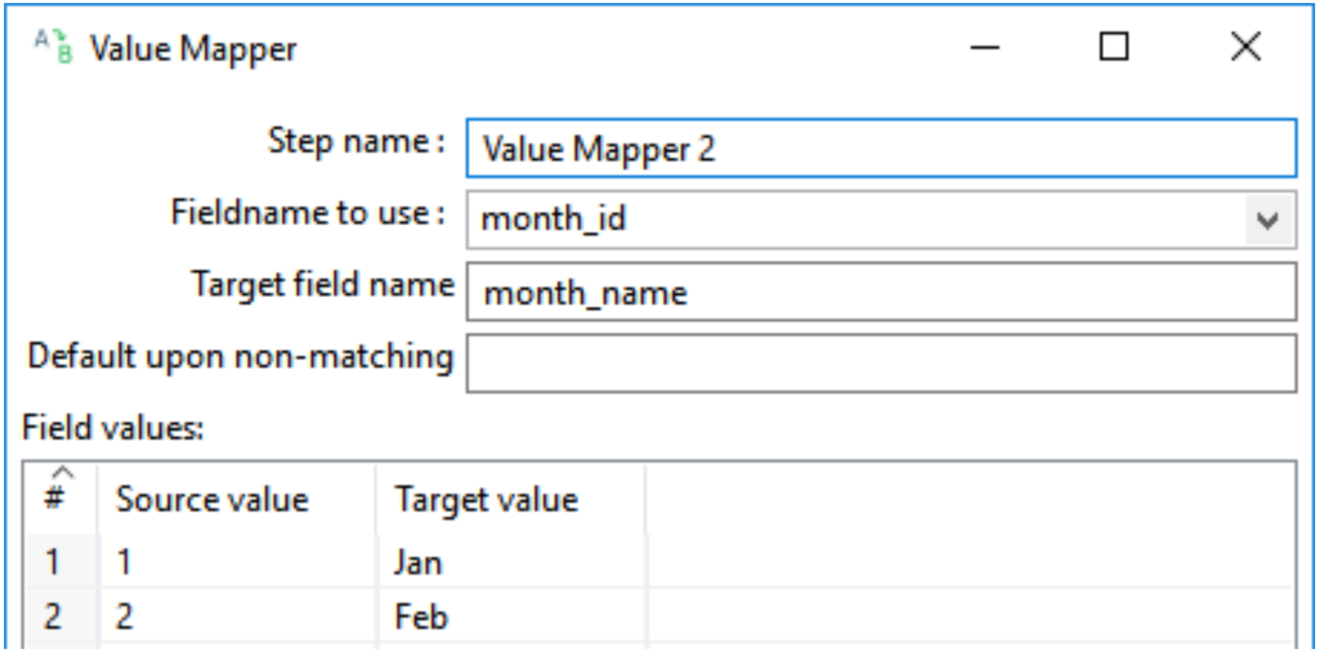
Field values:

#	Source value	Target value
1	1	Q1
2	2	Q2

Dimension tables

- The time dimension





Step name : Value Mapper 2

Fieldname to use : month_id

Target field name month_name

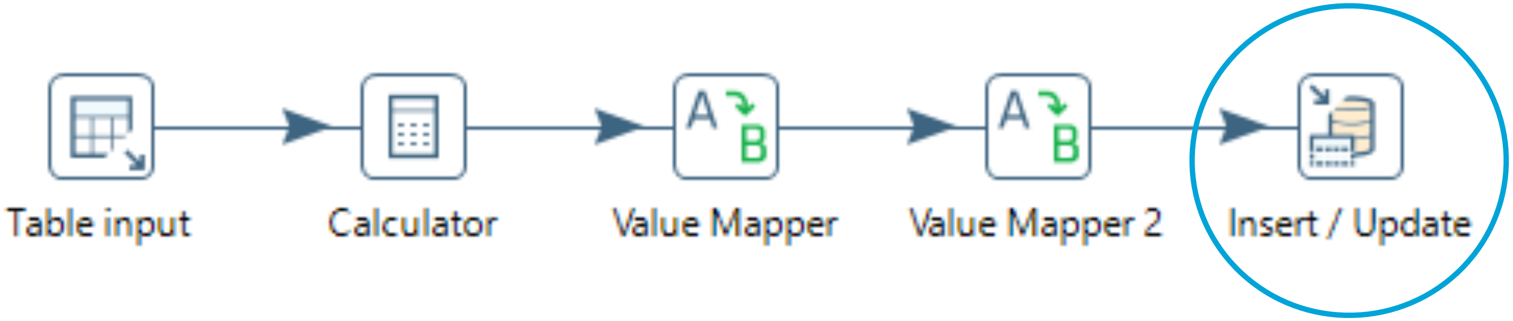
Default upon non-matching

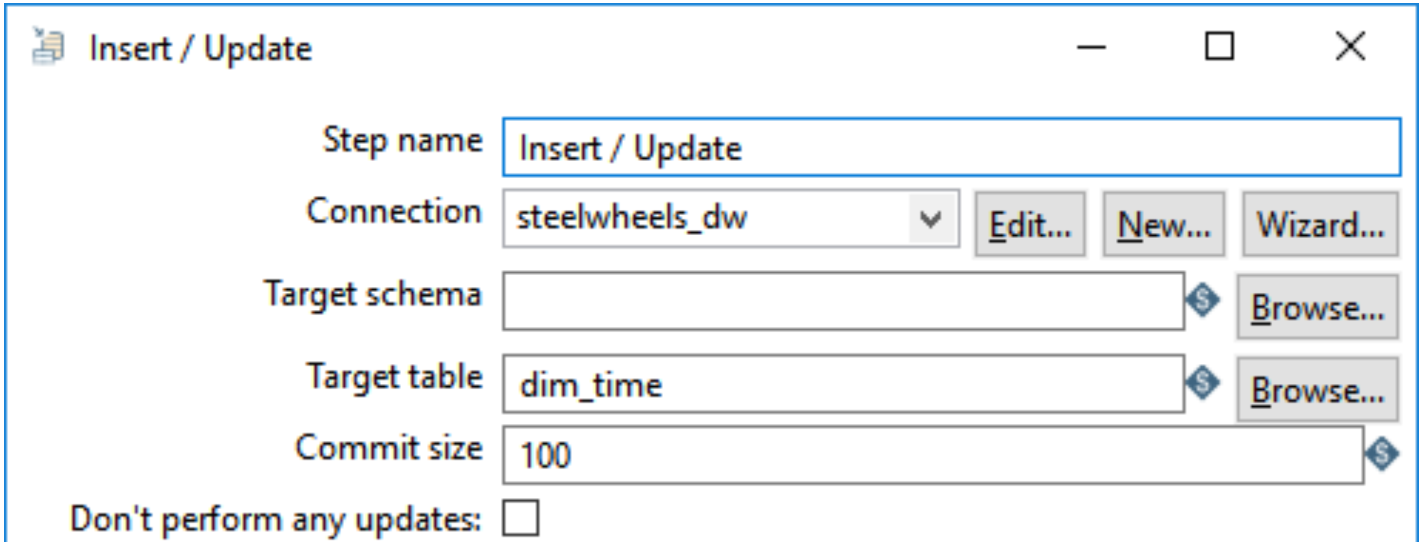
Field values:

#	Source value	Target value
1	1	Jan
2	2	Feb

Dimension tables

- The time dimension

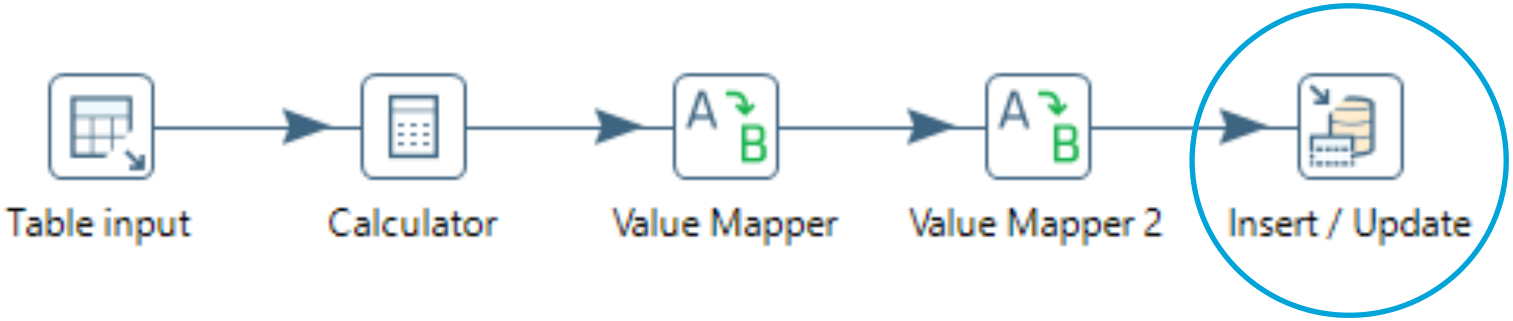


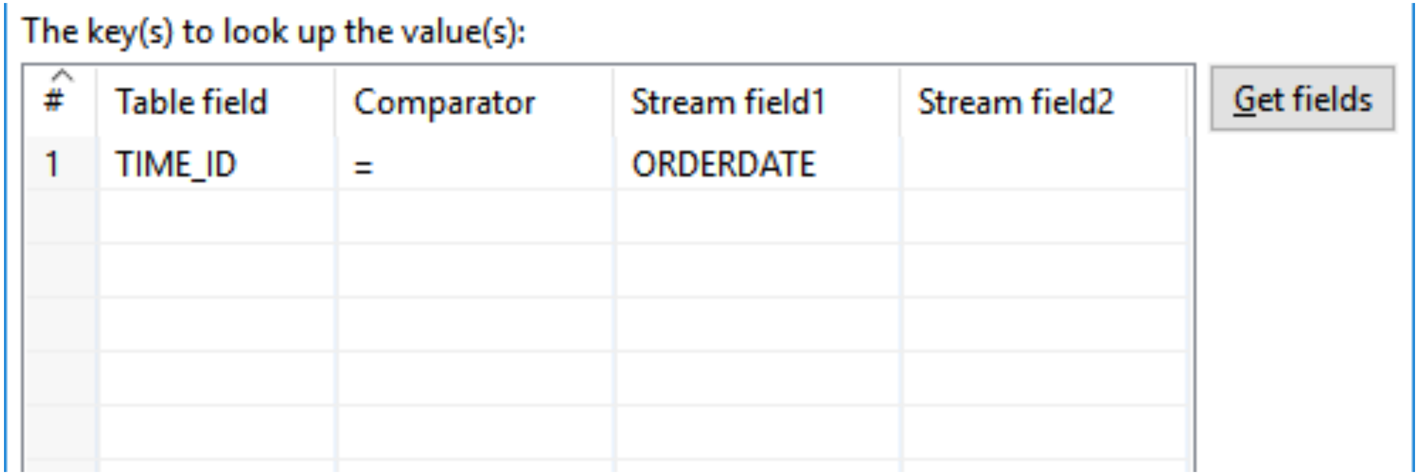


Step name	Insert / Update
Connection	steelwheels_dw
Target schema	
Target table	dim_time
Commit size	100
Don't perform any updates:	□

Dimension tables

- The time dimension

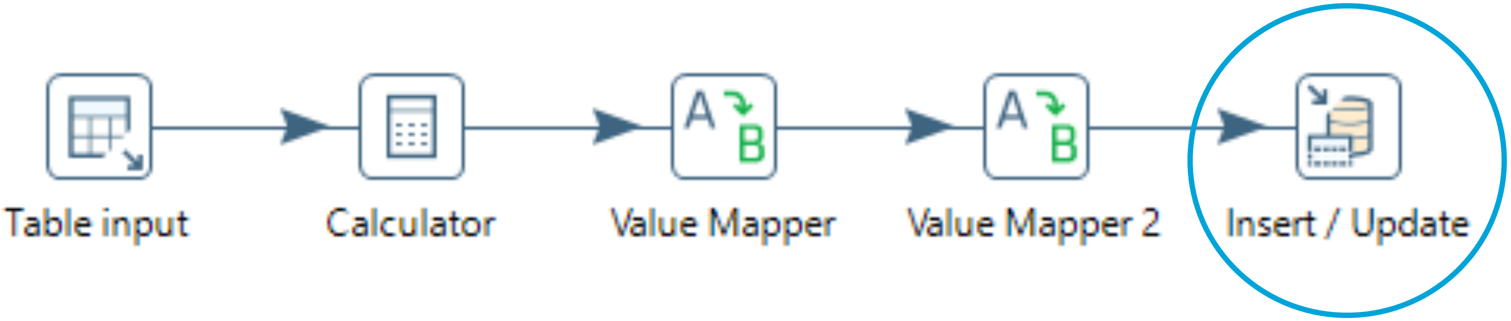


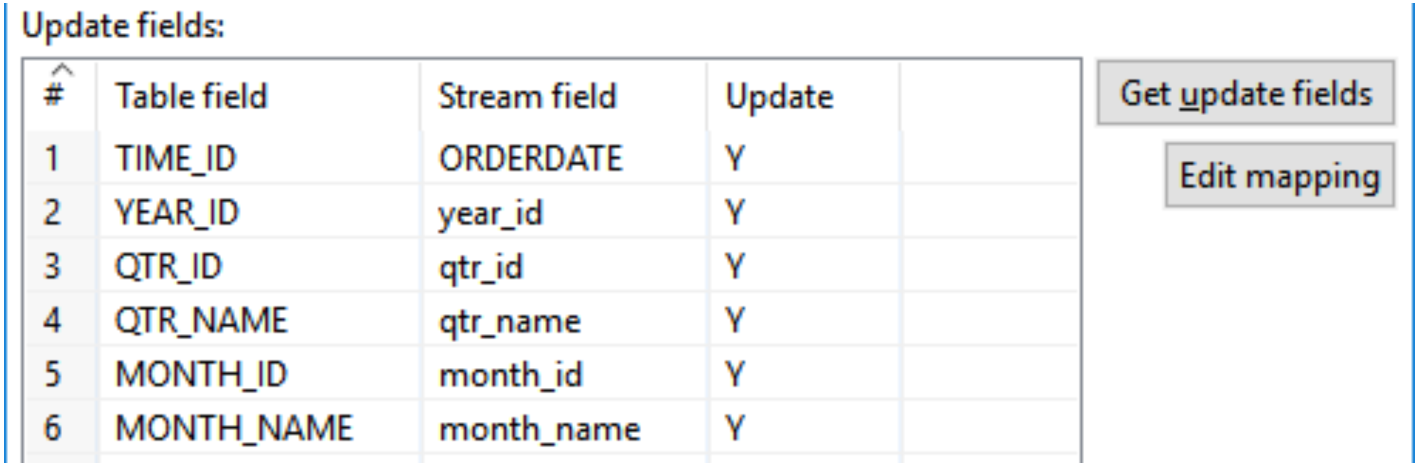

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	TIME_ID	=	ORDERDATE	

Dimension tables

- The time dimension

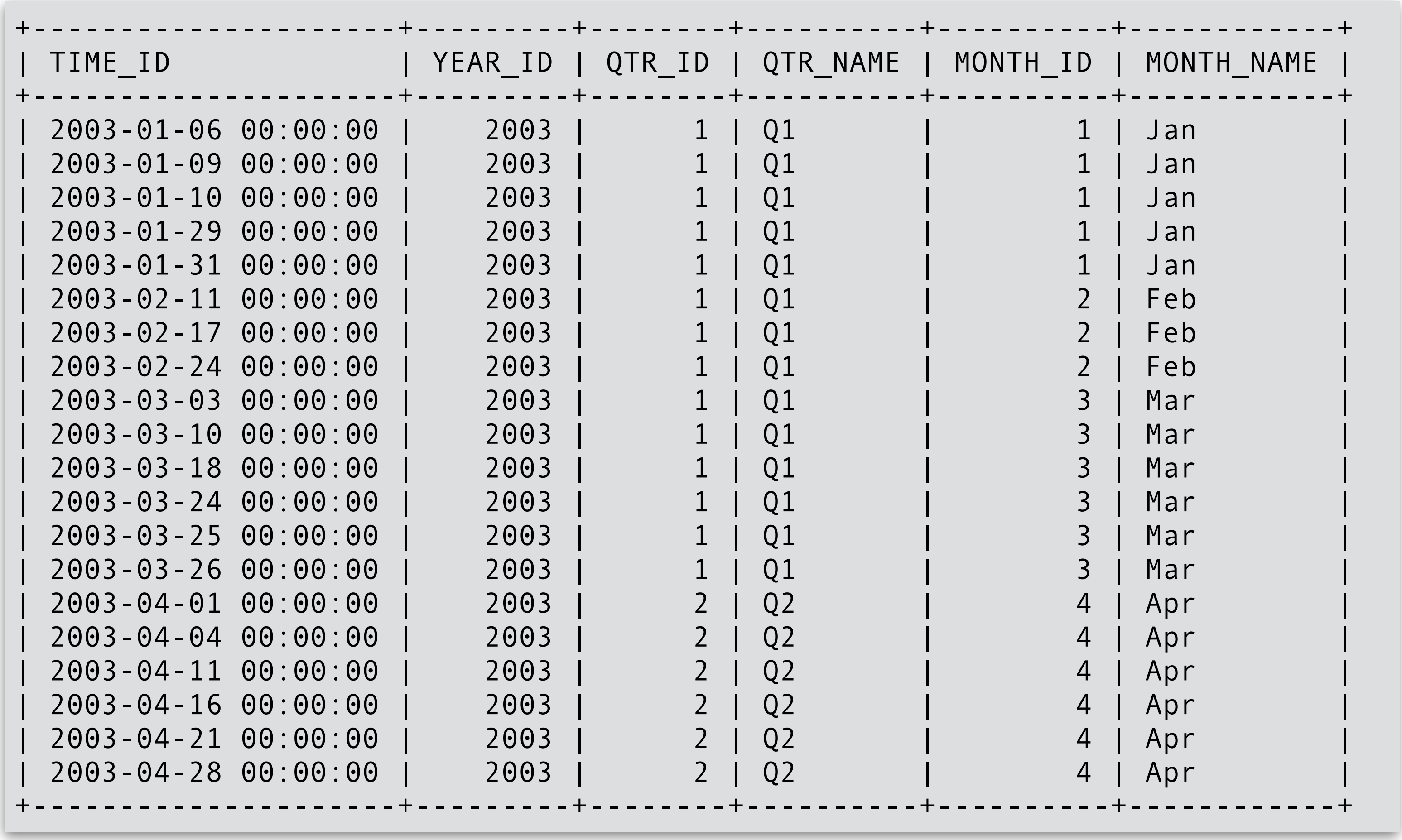



Update fields:

#	Table field	Stream field	Update
1	TIME_ID	ORDERDATE	Y
2	YEAR_ID	year_id	Y
3	QTR_ID	qtr_id	Y
4	QTR_NAME	qtr_name	Y
5	MONTH_ID	month_id	Y
6	MONTH_NAME	month_name	Y

Dimension tables

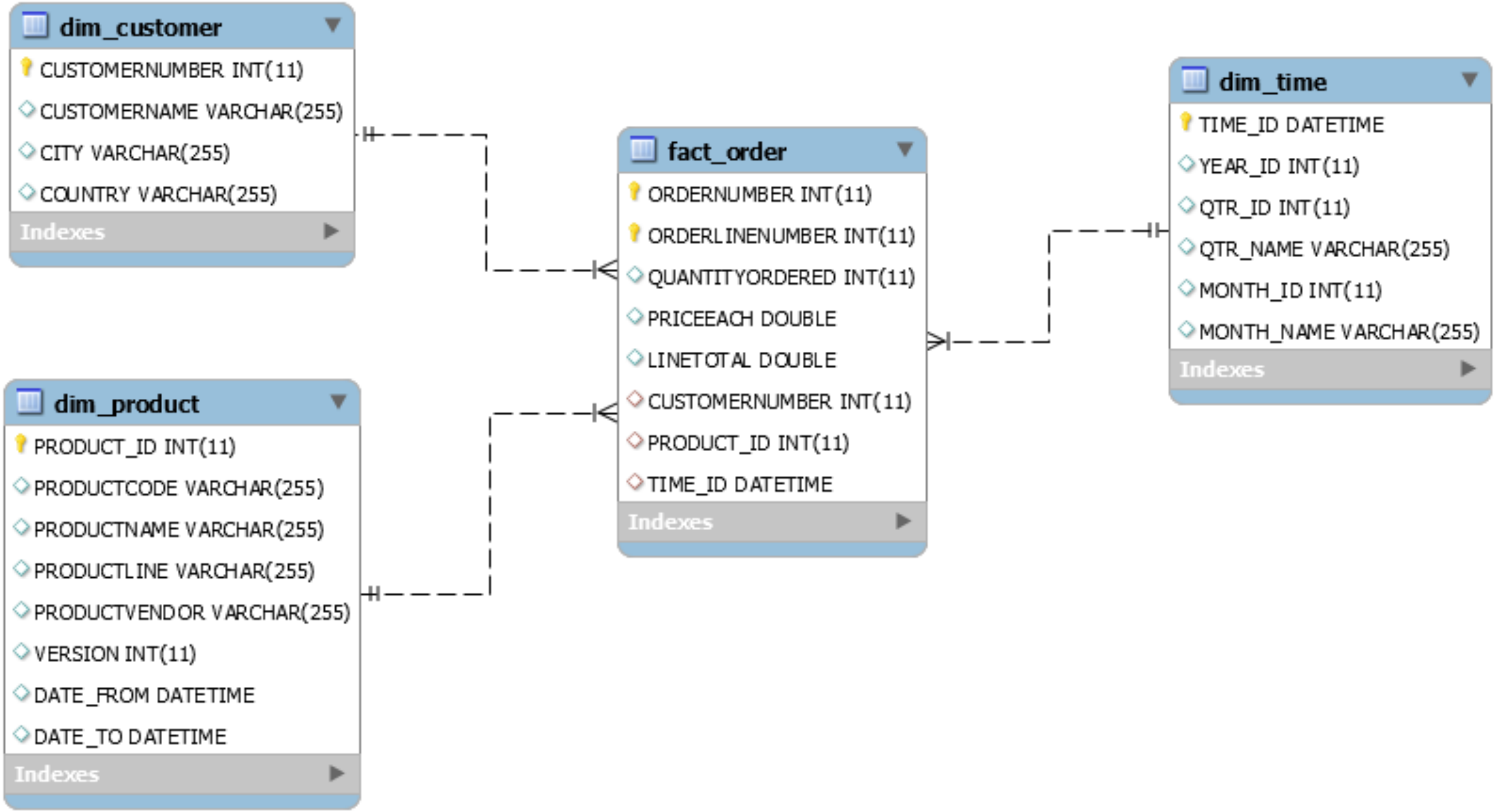
- The time dimension



```
+---------------------+---------+--------+----------+----------+------------+
| TIME_ID             | YEAR_ID | QTR_ID | QTR_NAME | MONTH_ID | MONTH_NAME |
+---------------------+---------+--------+----------+----------+------------+
| 2003-01-06 00:00:00 |    2003 |      1 | Q1       |        1 | Jan        |
| 2003-01-09 00:00:00 |    2003 |      1 | Q1       |        1 | Jan        |
| 2003-01-10 00:00:00 |    2003 |      1 | Q1       |        1 | Jan        |
| 2003-01-29 00:00:00 |    2003 |      1 | Q1       |        1 | Jan        |
| 2003-01-31 00:00:00 |    2003 |      1 | Q1       |        1 | Jan        |
| 2003-02-11 00:00:00 |    2003 |      1 | Q1       |        2 | Feb        |
| 2003-02-17 00:00:00 |    2003 |      1 | Q1       |        2 | Feb        |
| 2003-02-24 00:00:00 |    2003 |      1 | Q1       |        2 | Feb        |
| 2003-03-03 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-03-10 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-03-18 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-03-24 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-03-25 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-03-26 00:00:00 |    2003 |      1 | Q1       |        3 | Mar        |
| 2003-04-01 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
| 2003-04-04 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
| 2003-04-11 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
| 2003-04-16 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
| 2003-04-21 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
| 2003-04-28 00:00:00 |    2003 |      2 | Q2       |        4 | Apr        |
+---------------------+---------+--------+----------+----------+------------+
```

Loading the Facts table

Data warehouse



dim_customer
CUSTOMERNUMBER INT(11)
CUSTOMERNAME VARCHAR(255)
CITY VARCHAR(255)
COUNTRY VARCHAR(255)

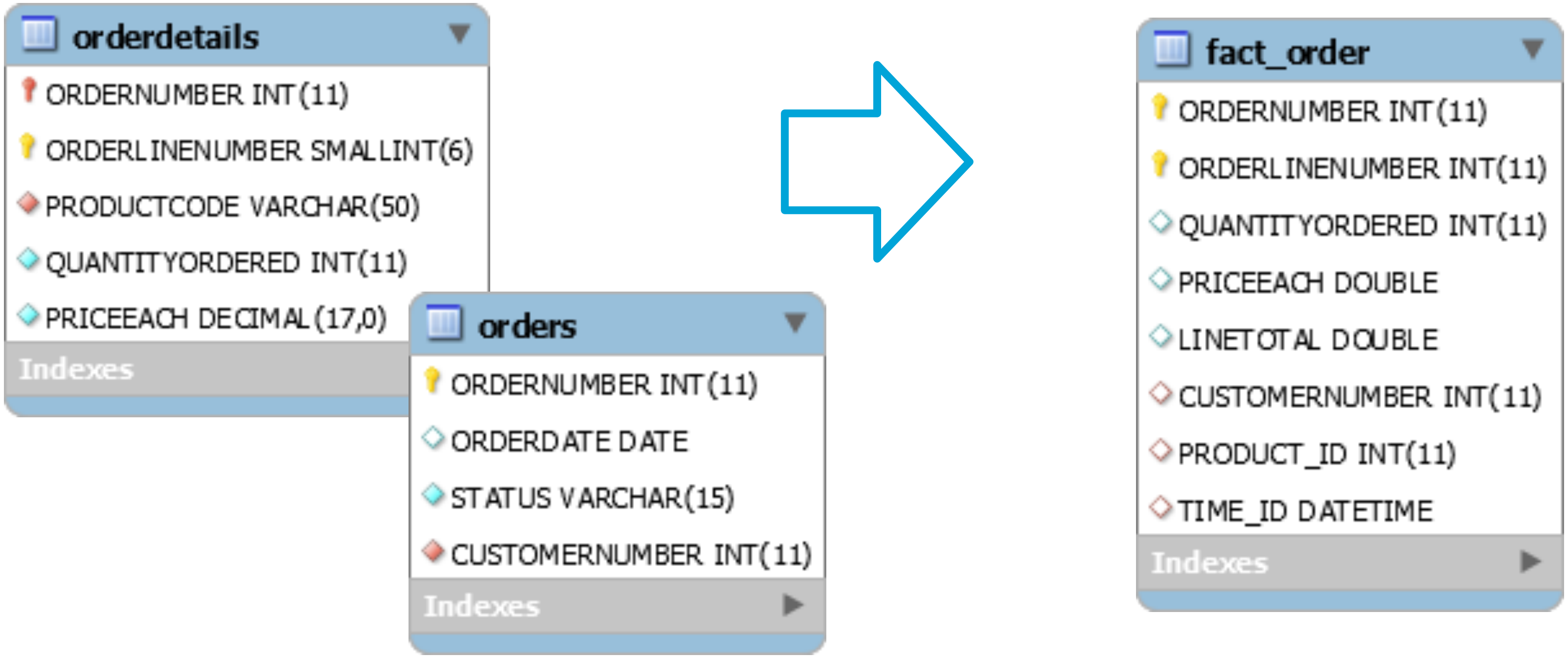
dim_product
PRODUCT_ID INT(11)
PRODUCTCODE VARCHAR(255)
PRODUCTNAME VARCHAR(255)
PRODUCTLINE VARCHAR(255)
PRODUCTVENDOR VARCHAR(255)
VERSION INT(11)
DATE_FROM DATETIME
DATE_TO DATETIME

fact_order
ORDERNUMBER INT(11)
ORDERLINENUMBER INT(11)
QUANTITYORDERED INT(11)
PRICEEACH DOUBLE
LINETOTAL DOUBLE
CUSTOMERNUMBER INT(11)
PRODUCT_ID INT(11)
TIME_ID DATETIME

dim_time
TIME_ID DATETIME
YEAR_ID INT(11)
QTR_ID INT(11)
QTR_NAME VARCHAR(255)
MONTH_ID INT(11)
MONTH_NAME VARCHAR(255)

Fact table

- The fact table
  - quantity, unit price, and line total (must be calculated)
  - data comes from orderdetails and orders
    - but product id must come from product dimension (!)



orderdetails

ORDERNUMBER INT(11)
ORDERLINENUMBER SMALLINT(6)
PRODUCTCODE VARCHAR(50)
QUANTITYORDERED INT(11)
PRICEEACH DECIMAL(17,0)
Indexes

orders

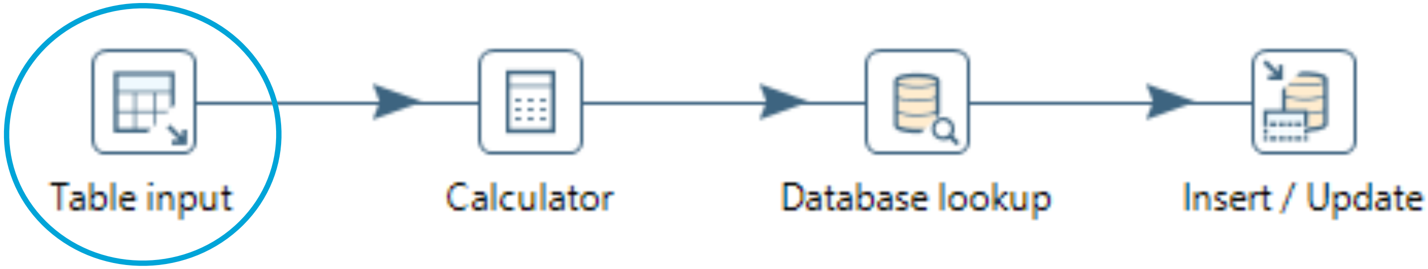
ORDERNUMBER INT(11)
ORDERDATE DATE
STATUS VARCHAR(15)
CUSTOMERNUMBER INT(11)
Indexes

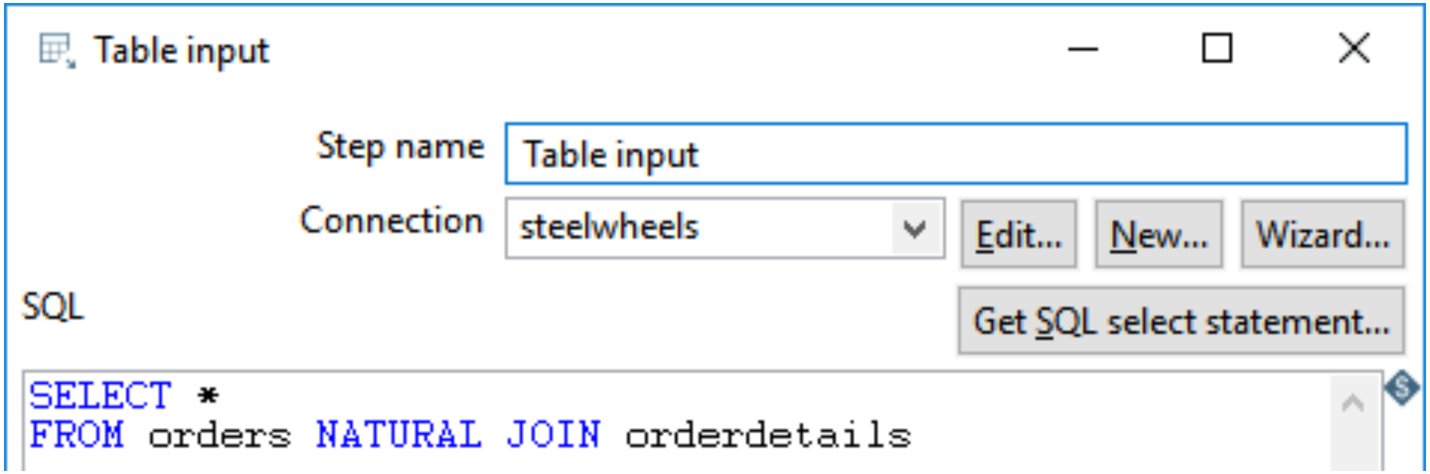
fact_order

ORDERNUMBER INT(11)
ORDERLINENUMBER INT(11)
QUANTITYORDERED INT(11)
PRICEEACH DOUBLE
LINETOTAL DOUBLE
CUSTOMERNUMBER INT(11)
PRODUCT_ID INT(11)
TIME_ID DATETIME
Indexes

Fact table

- The fact table





Step name: Table input

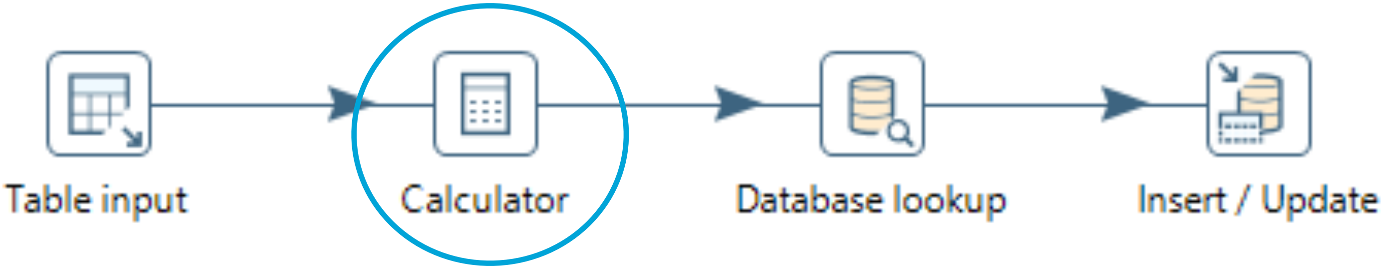
Connection: steelwheels

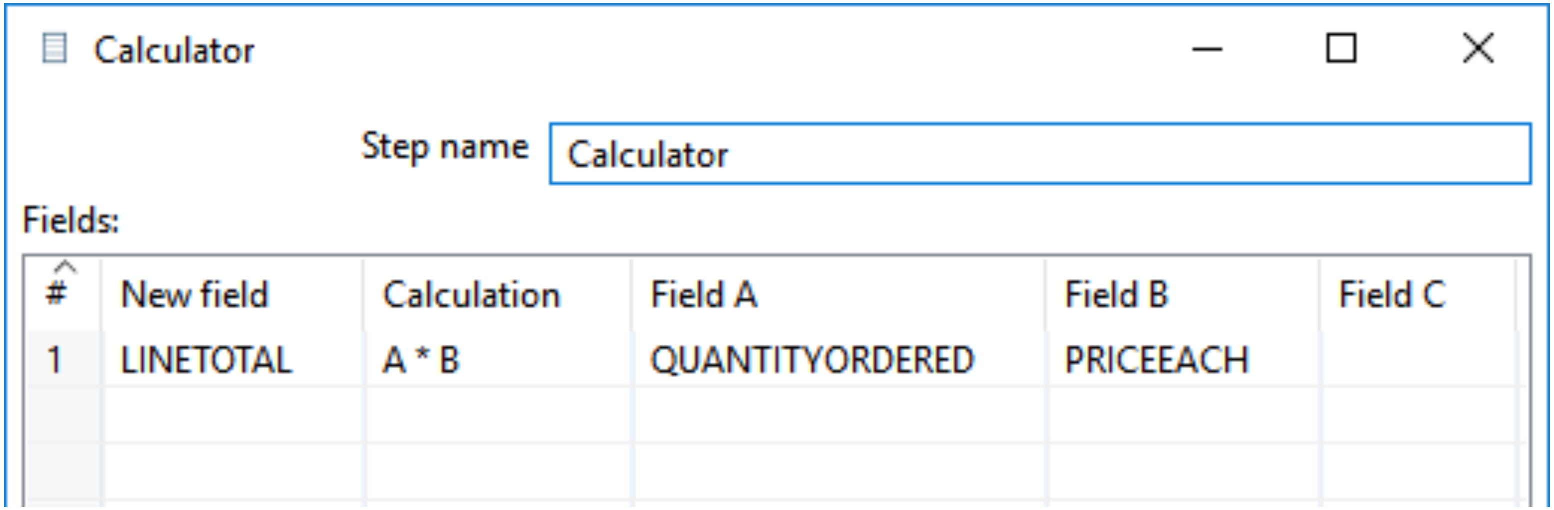
SQL

```
SELECT *
FROM orders NATURAL JOIN orderdetails
```

Fact table

- The fact table





Calculator

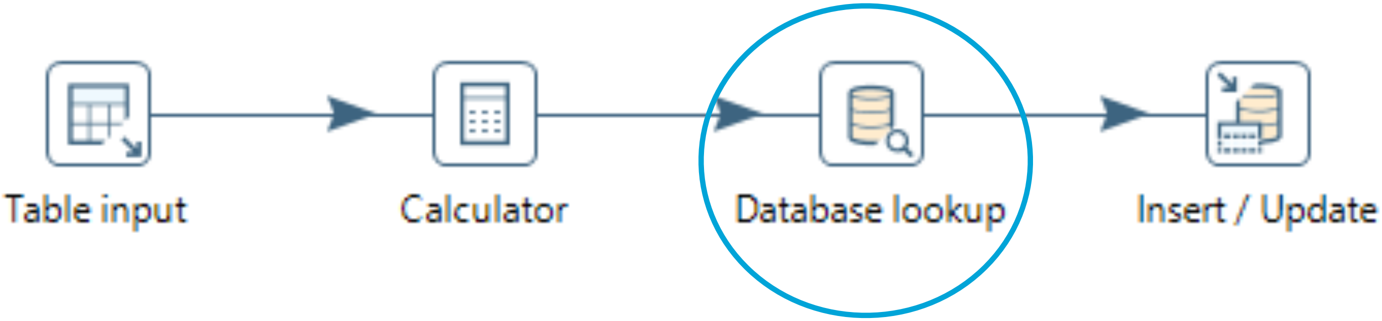
Step name: Calculator

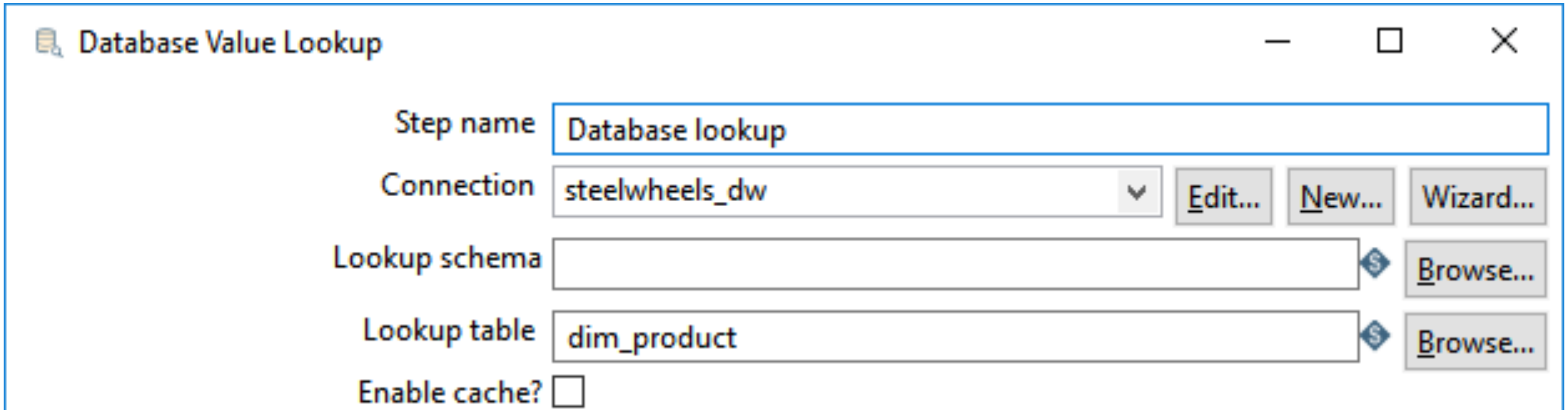
Fields:

#	New field	Calculation	Field A	Field B	Field C
1	LINETOTAL	A * B	QUANTITYORDERED	PRICEEACH	

Fact table

- The fact table

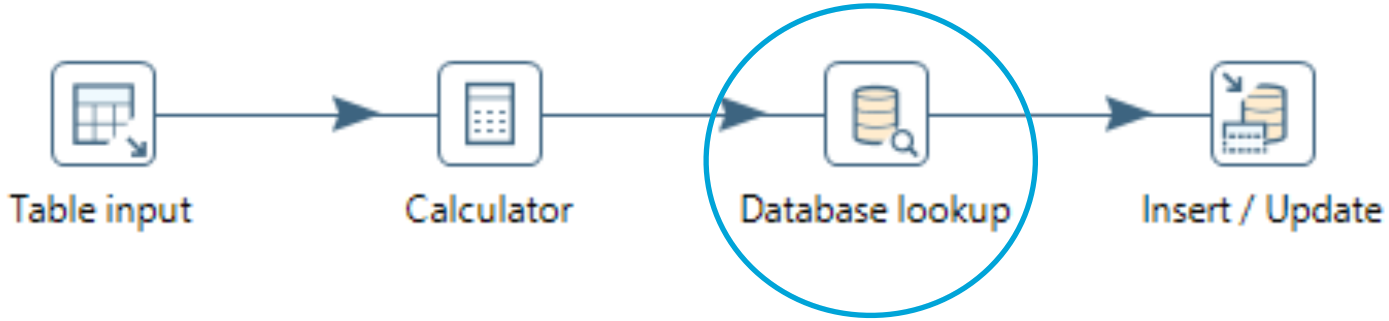


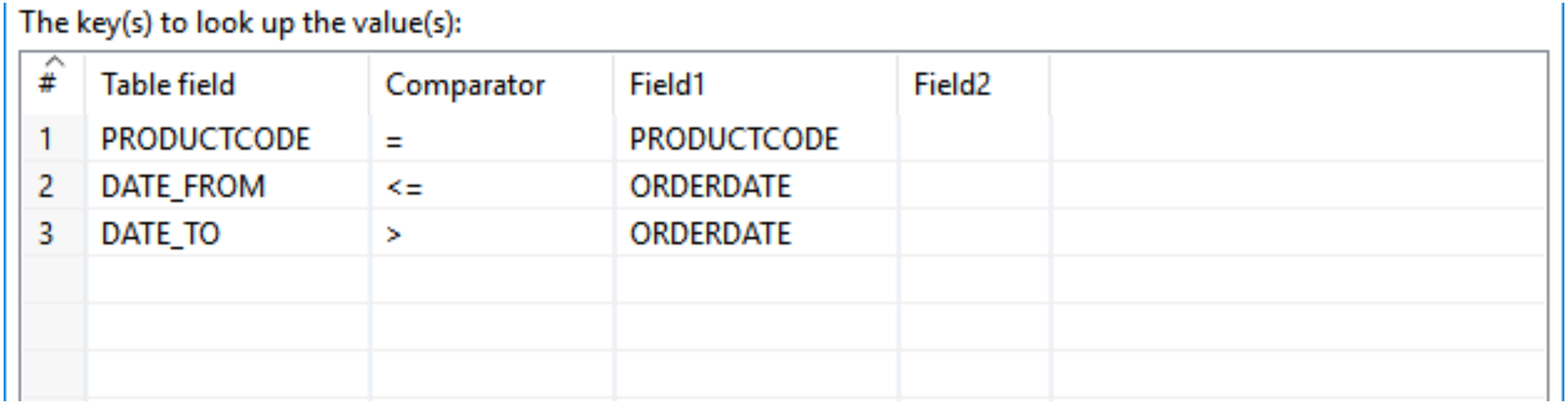


Step name	Database lookup
Connection	steelwheels_dw
Lookup schema	
Lookup table	dim_product
Enable cache?	□

Fact table

- The fact table



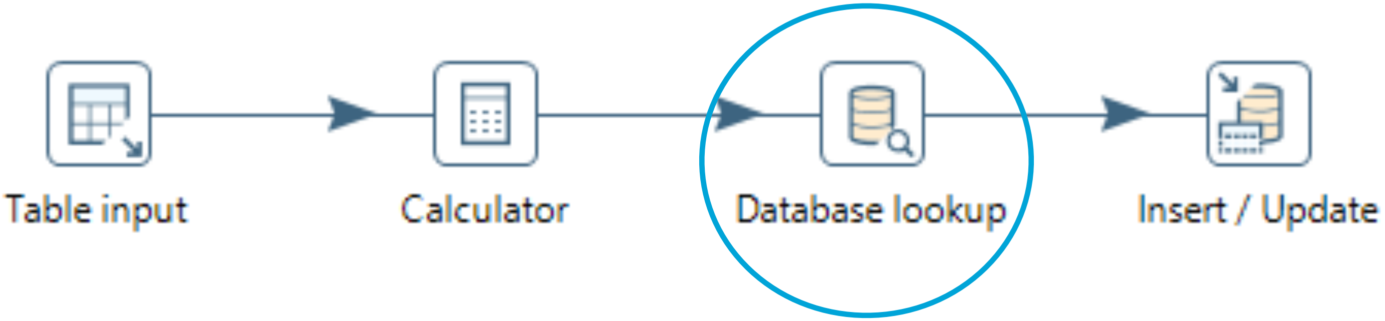


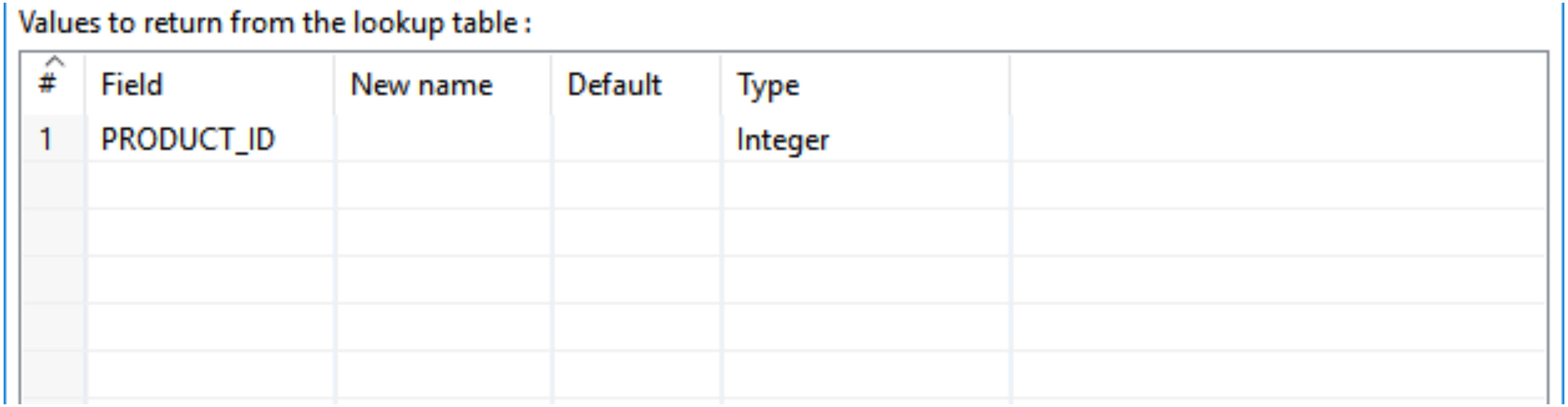
The key(s) to look up the value(s):

#	Table field	Comparator	Field1	Field2
1	PRODUCTCODE	=	PRODUCTCODE	
2	DATE_FROM	<=	ORDERDATE	
3	DATE_TO	>	ORDERDATE	

Fact table

- The fact table



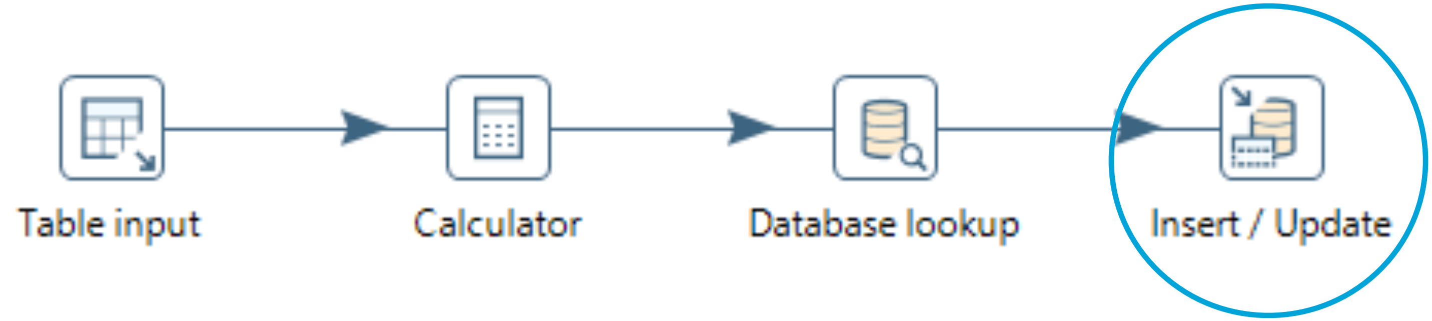


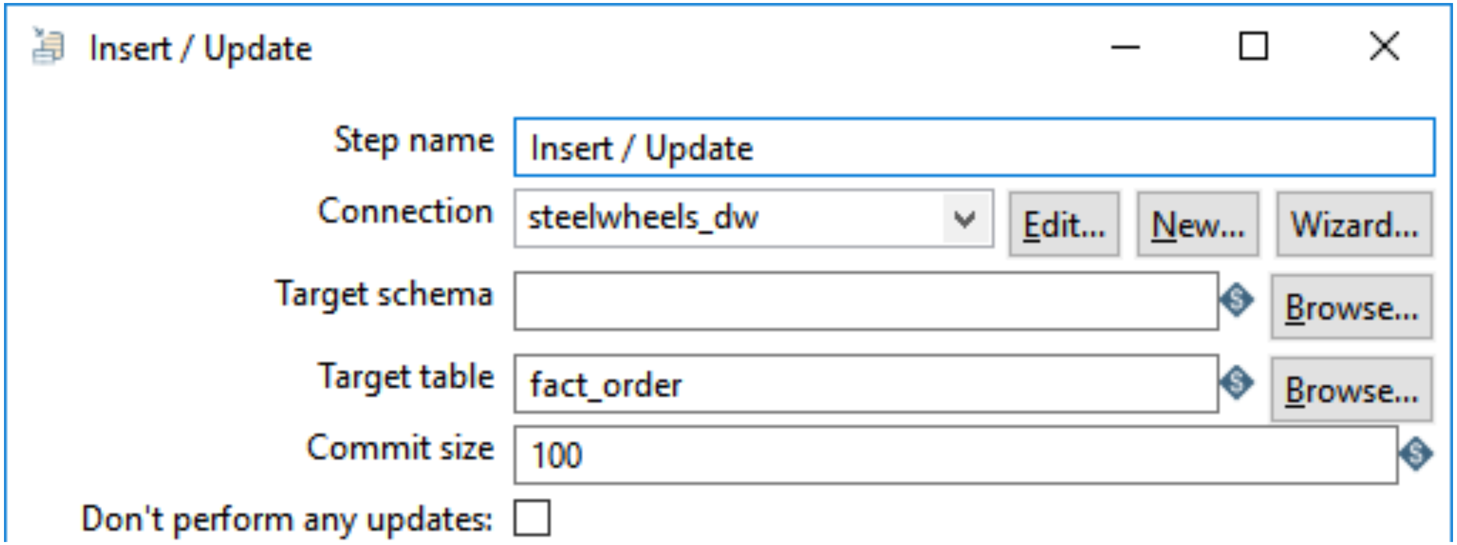
Values to return from the lookup table :

#	Field	New name	Default	Type
1	PRODUCT_ID			Integer

Fact table

- The fact table





Step name: Insert / Update

Connection: steelwheels_dw

Target schema:

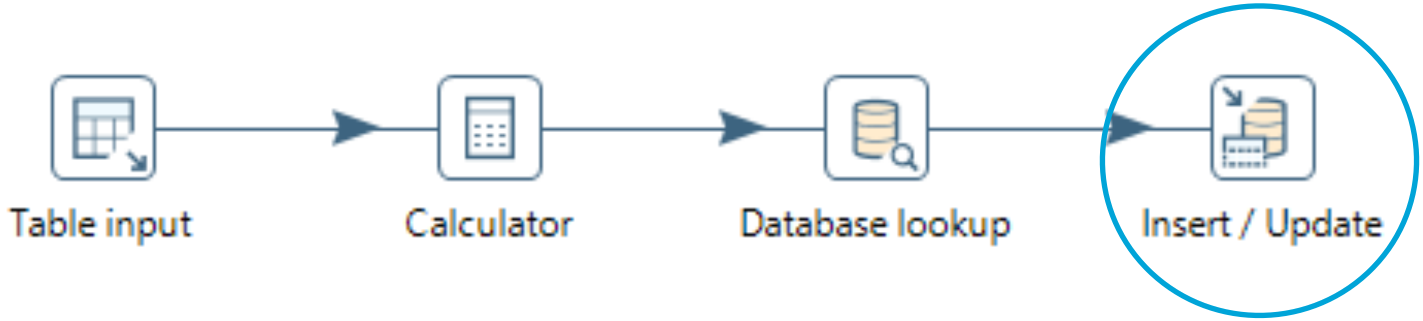
Target table: fact_order

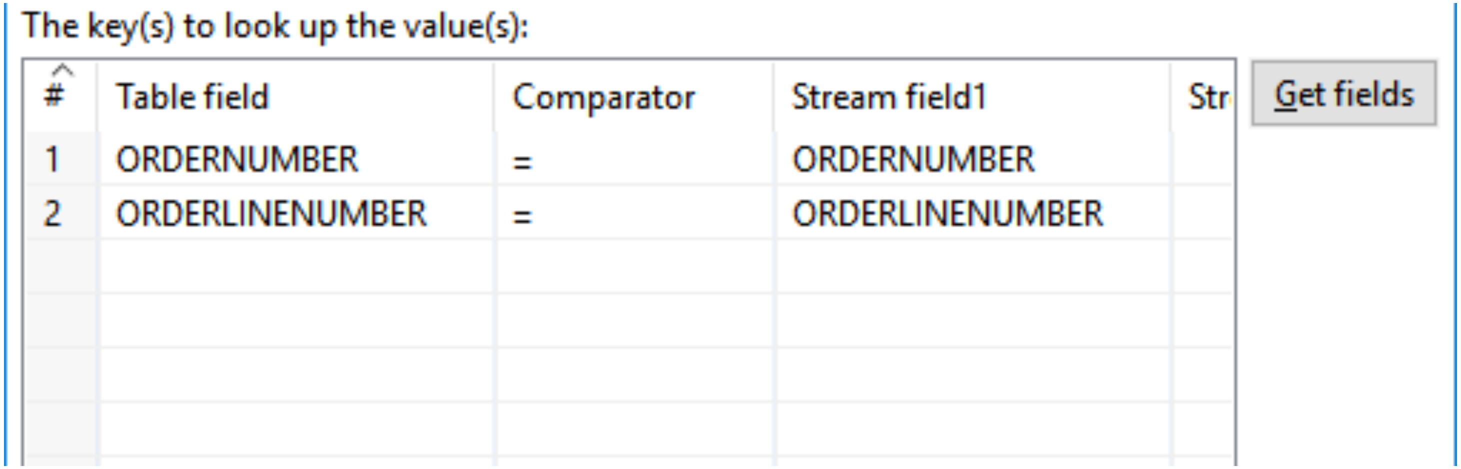
Commit size: 100

Don't perform any updates: □

Fact table

- The fact table

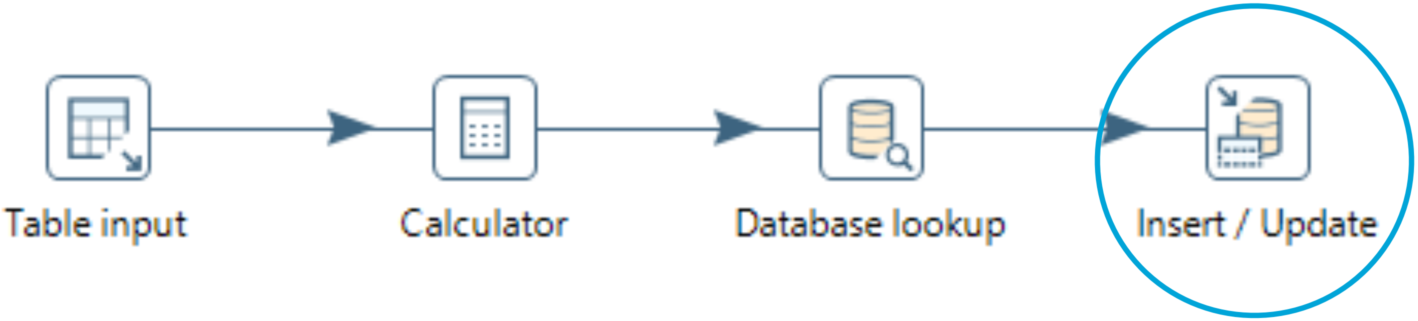


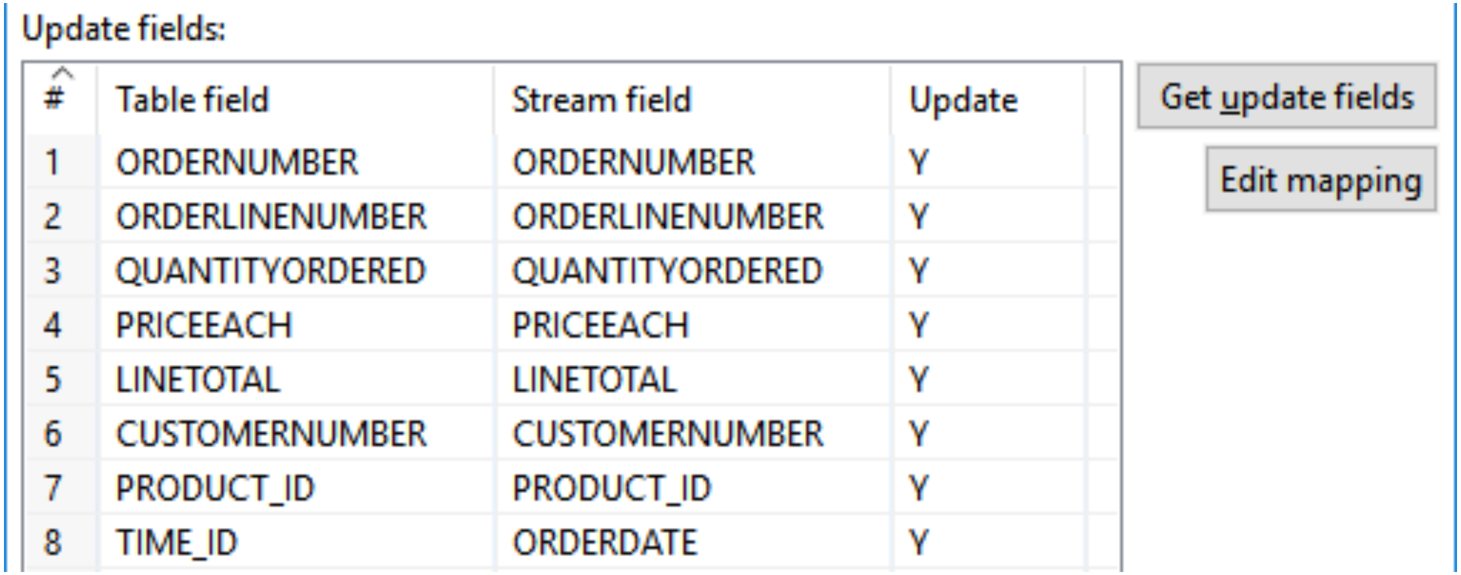

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Str
1	ORDERNUMBER	=	ORDERNUMBER	
2	ORDERLINENUMBER	=	ORDERLINENUMBER	

Fact table

- The fact table

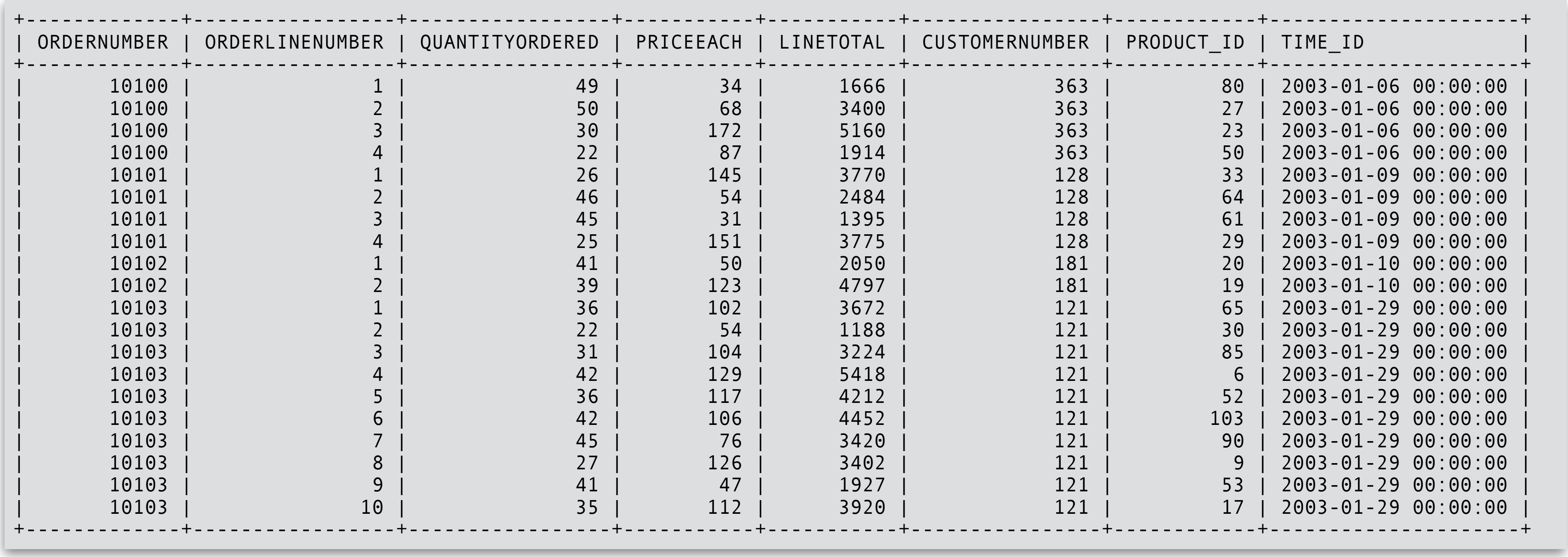



Update fields:

#	Table field	Stream field	Update
1	ORDERNUMBER	ORDERNUMBER	Y
2	ORDERLINENUMBER	ORDERLINENUMBER	Y
3	QUANTITYORDERED	QUANTITYORDERED	Y
4	PRICEEACH	PRICEEACH	Y
5	LINETOTAL	LINETOTAL	Y
6	CUSTOMERNUMBER	CUSTOMERNUMBER	Y
7	PRODUCT_ID	PRODUCT_ID	Y
8	TIME_ID	ORDERDATE	Y

Fact table

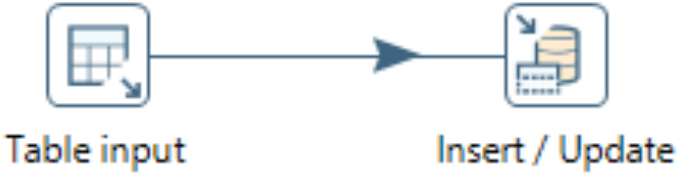
- The fact table

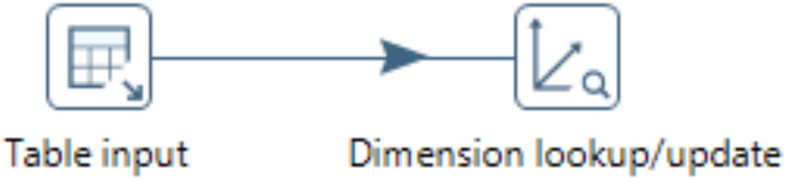


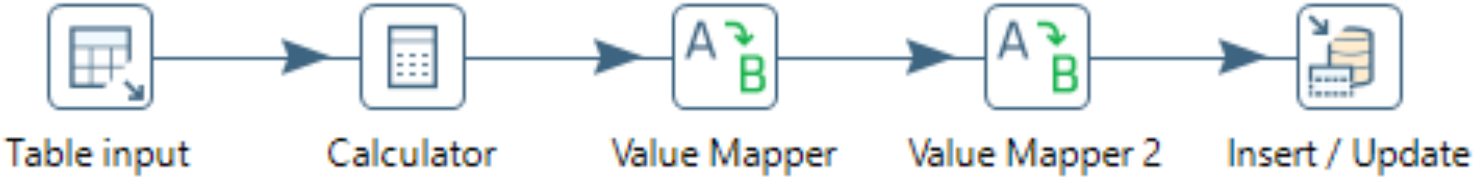
```
+-------------+-----------------+-----------------+-----------+-----------+----------------+------------+---------------------+
| ORDERNUMBER | ORDERLINENUMBER | QUANTITYORDERED | PRICEEACH | LINETOTAL | CUSTOMERNUMBER | PRODUCT_ID | TIME_ID             |
+-------------+-----------------+-----------------+-----------+-----------+----------------+------------+---------------------+
|       10100 |               1 |              49 |        34 |      1666 |            363 |         80 | 2003-01-06 00:00:00 |
|       10100 |               2 |              50 |        68 |      3400 |            363 |         27 | 2003-01-06 00:00:00 |
|       10100 |               3 |              30 |       172 |      5160 |            363 |         23 | 2003-01-06 00:00:00 |
|       10100 |               4 |              22 |        87 |      1914 |            363 |         50 | 2003-01-06 00:00:00 |
|       10101 |               1 |              26 |       145 |      3770 |            128 |         33 | 2003-01-09 00:00:00 |
|       10101 |               2 |              46 |        54 |      2484 |            128 |         64 | 2003-01-09 00:00:00 |
|       10101 |               3 |              45 |        31 |      1395 |            128 |         61 | 2003-01-09 00:00:00 |
|       10101 |               4 |              25 |       151 |      3775 |            128 |         29 | 2003-01-09 00:00:00 |
|       10102 |               1 |              41 |        50 |      2050 |            181 |         20 | 2003-01-10 00:00:00 |
|       10102 |               2 |              39 |       123 |      4797 |            181 |         19 | 2003-01-10 00:00:00 |
|       10103 |               1 |              36 |       102 |      3672 |            121 |         65 | 2003-01-29 00:00:00 |
|       10103 |               2 |              22 |        54 |      1188 |            121 |         30 | 2003-01-29 00:00:00 |
|       10103 |               3 |              31 |       104 |      3224 |            121 |         85 | 2003-01-29 00:00:00 |
|       10103 |               4 |              42 |       129 |      5418 |            121 |          6 | 2003-01-29 00:00:00 |
|       10103 |               5 |              36 |       117 |      4212 |            121 |         52 | 2003-01-29 00:00:00 |
|       10103 |               6 |              42 |       106 |      4452 |            121 |        103 | 2003-01-29 00:00:00 |
|       10103 |               7 |              45 |        76 |      3420 |            121 |         90 | 2003-01-29 00:00:00 |
|       10103 |               8 |              27 |       126 |      3402 |            121 |          9 | 2003-01-29 00:00:00 |
|       10103 |               9 |              41 |        47 |      1927 |            121 |         53 | 2003-01-29 00:00:00 |
|       10103 |              10 |              35 |       112 |      3920 |            121 |         17 | 2003-01-29 00:00:00 |
+-------------+-----------------+-----------------+-----------+-----------+----------------+------------+---------------------+
```

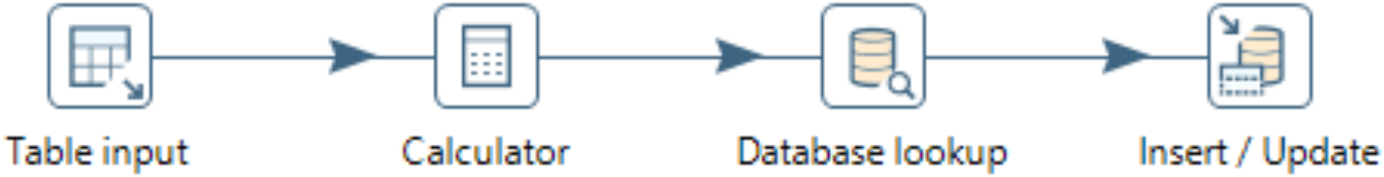
Complete ETL process

- Transformations
  - the customer dimension
  - the product dimension
  - the time dimension
  - the fact table



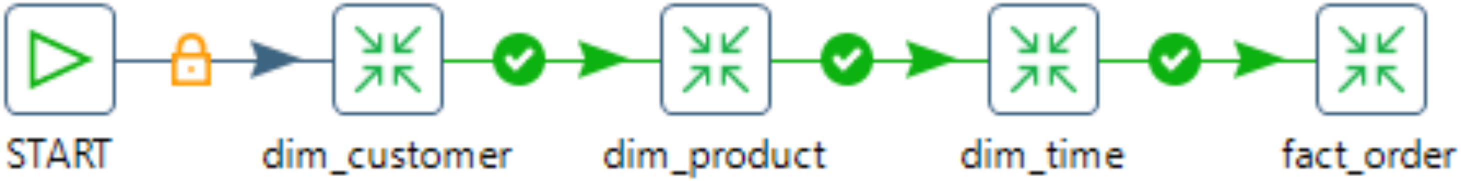






Complete ETL process

- Defining a job
  - A job is a sequence of transformations
    - each transformation runs only upon successful completion of the previous one
  - The complete ETL process for the data warehouse
    - can be run multiple times to update the data warehouse


START
dim_customer
dim_product
dim_time
fact_order