

Professor:  
Fabiano Tavares



**CARREGAR DADOS - LAB IV**

ANO: 2019

# **LAB IV – Carregar Dados**

**Criar Tabelas e Índices, Load, Carga, Fatos e Dimensões**

# Pré-requisitos

- ❖ JAVA;
- ❖ Instalar Pentaho Data Integration;
- ❖ Possuir o diretório de “Bases” de teste;
- ❖ MySQL Server instalado

# Exercício 1

Criando as tabelas de um banco de dados

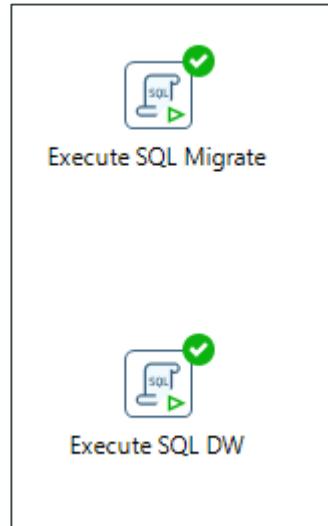
# Exercício 1 – Criando as tabelas de um banco de dados

**Exercício:** Criando as tabelas de um banco de dados para os exercícios de Migração e Carga em DW.

**Objetivo:** Criar estruturas físicas para receber cargas.

Depende: Exercício 3 do Lab I

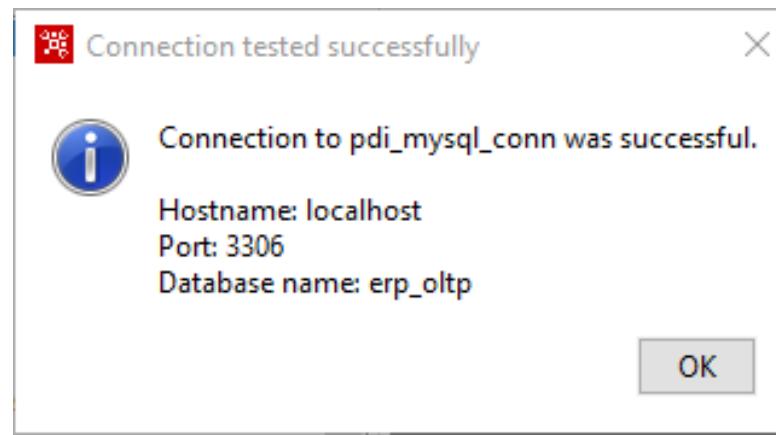
**Resultado:**



# Exercício 1 – Criando as tabelas de um banco de dados

1 – Para esse exercício, vamos utilizar os arquivo *scripts-products.sql* e *datawarehouse\_olap.sql*, localizados na pasta script do material do curso (portal do aluno).

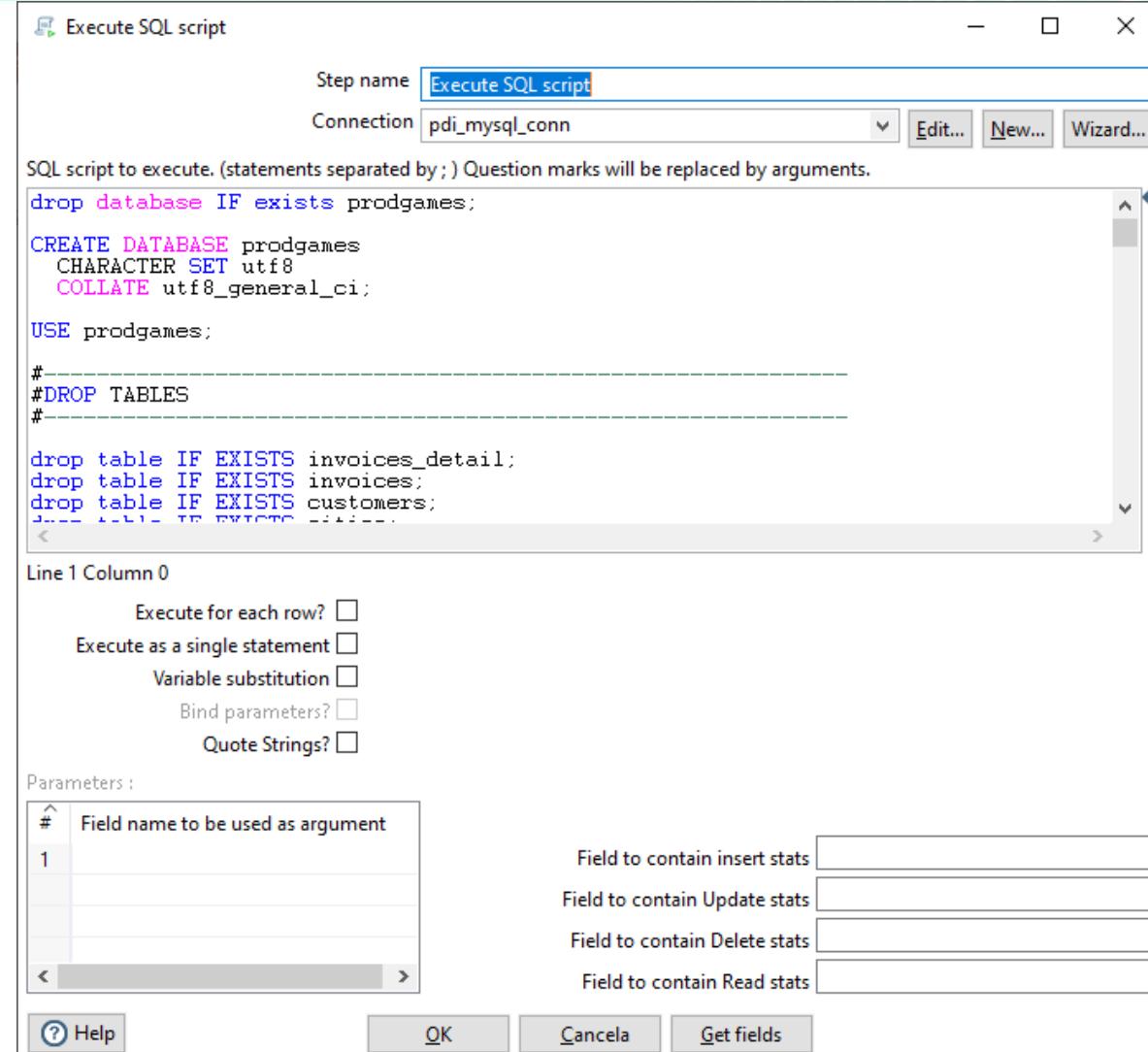
- *scripts-products.sql*: criam as tabelas que armazenam os dados de uma loja online de vendas de jogos.
- *datawarehouse\_olap.sql*: criam tabelas que armazenam os dados do DW de vendas.
- Antes de iniciar, verifique se a conexão *pdi\_mysql\_con* (criada no LAB I) está disponível.
- Acesse a aba *View* e expanda a árvore de conexões. Edite a conexão e pressione o botão *Test*.



# Exercício 1 – Criando as tabelas de um banco de dados

2 – Crie uma nova transformação e adicione um step do tipo *Execute SQL script*, da categoria *Scripting*.

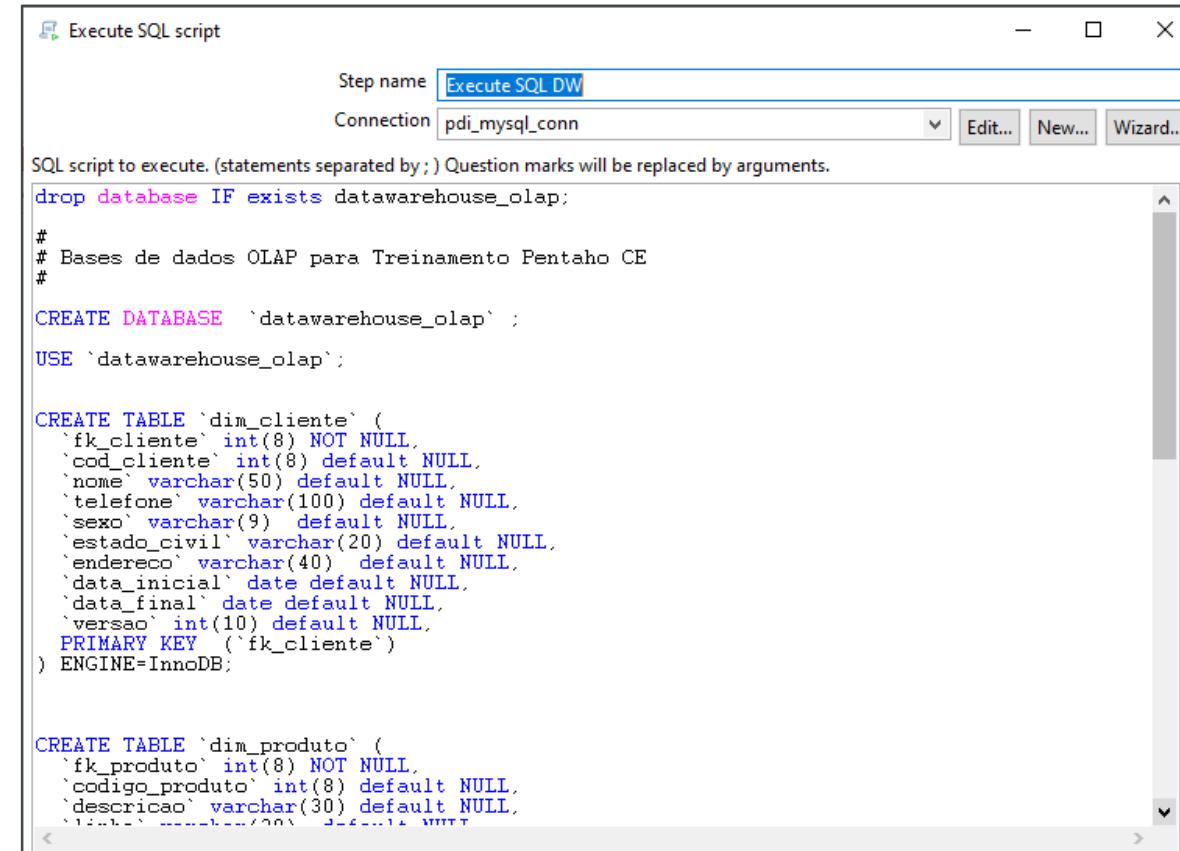
- Edite o step, escolhendo a conexão *pdi\_mysql\_conn* no campo *Connection*.
- Em seguida, abra o arquivo *scripts-products.sql* em um editor de texto, copie e cole o seu conteúdo no campo *SQLscript to execute*.



# Exercício 1 – Criando as tabelas de um banco de dados

3 – Adicione um segundo step do tipo *Execute SQL script*, da categoria *Scripting*.

- Edite o step, escolhendo a conexão *pdi\_mysql\_conn* no campo *Connection*.
  - Em seguida, abra o arquivo *datawarehouse.olap.sql* em um editor de texto, copie e cole o seu conteúdo no campo *SQLscript to execute*.
- ❖ Agora execute a transformação.



The screenshot shows the 'Execute SQL script' step configuration window. The 'Step name' is set to 'Execute SQL DW' and the 'Connection' is set to 'pdi\_mysql\_conn'. The 'SQL script to execute' field contains the following SQL code:

```
drop database IF exists datawarehouse.olap;
#
# Bases de dados OLAP para Treinamento Pentaho CE
#
CREATE DATABASE `datawarehouse.olap` ;
USE `datawarehouse.olap`;

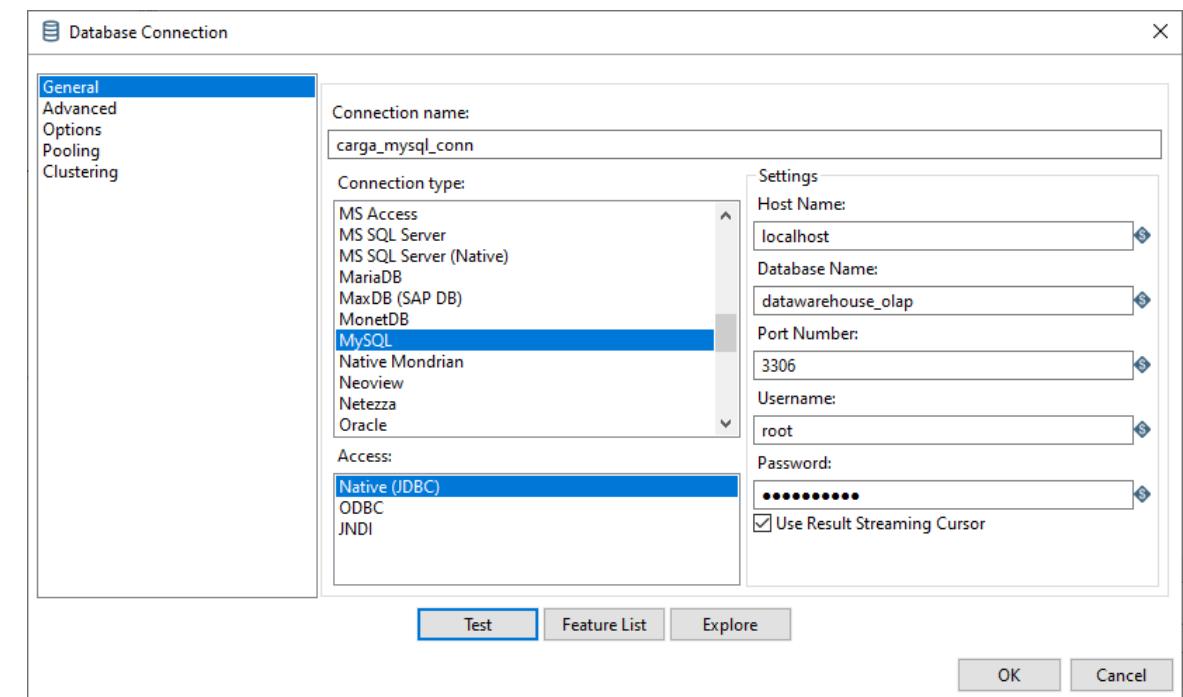
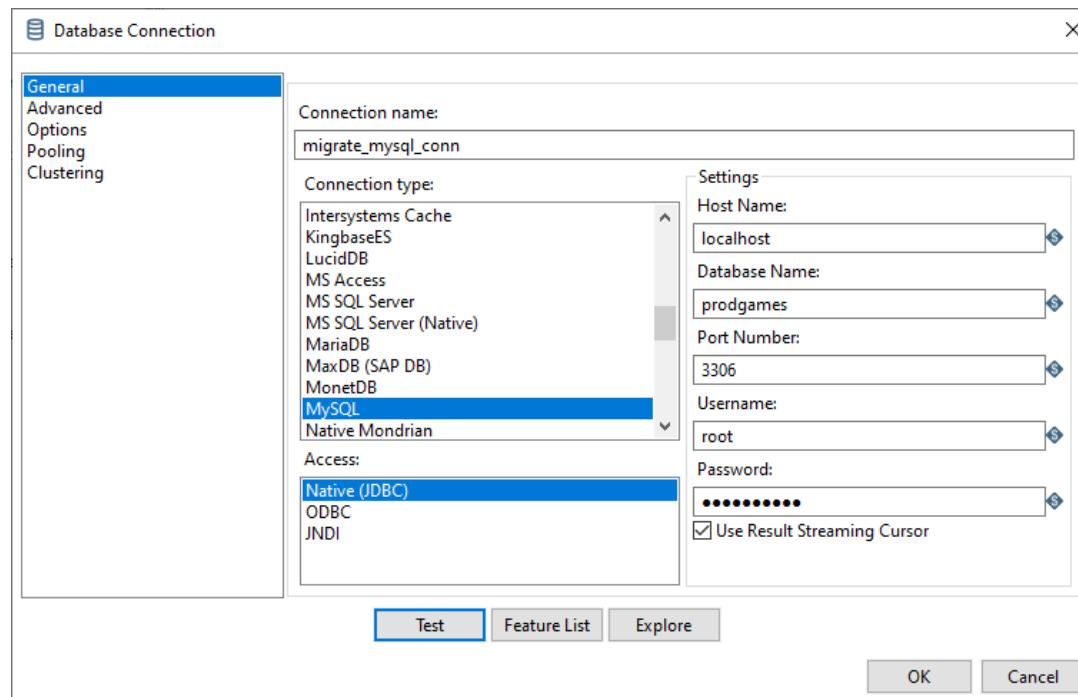
CREATE TABLE `dim_cliente` (
`fk_cliente` int(8) NOT NULL,
`cod_cliente` int(8) default NULL,
`nome` varchar(50) default NULL,
`telefone` varchar(100) default NULL,
`sexo` varchar(9) default NULL,
`estado_civil` varchar(20) default NULL,
`endereco` varchar(40) default NULL,
`data_inicial` date default NULL,
`data_final` date default NULL,
`versao` int(10) default NULL,
PRIMARY KEY (`fk_cliente`)
) ENGINE=InnoDB;

CREATE TABLE `dim_produto` (
`fk_produto` int(8) NOT NULL,
`codigo_produto` int(8) default NULL,
`descricao` varchar(30) default NULL,
`unidade` varchar(20) default NULL,
```

# Exercício 1 – Criando as tabelas de um banco de dados

3 – Agora crie mais duas conexões no pdi.

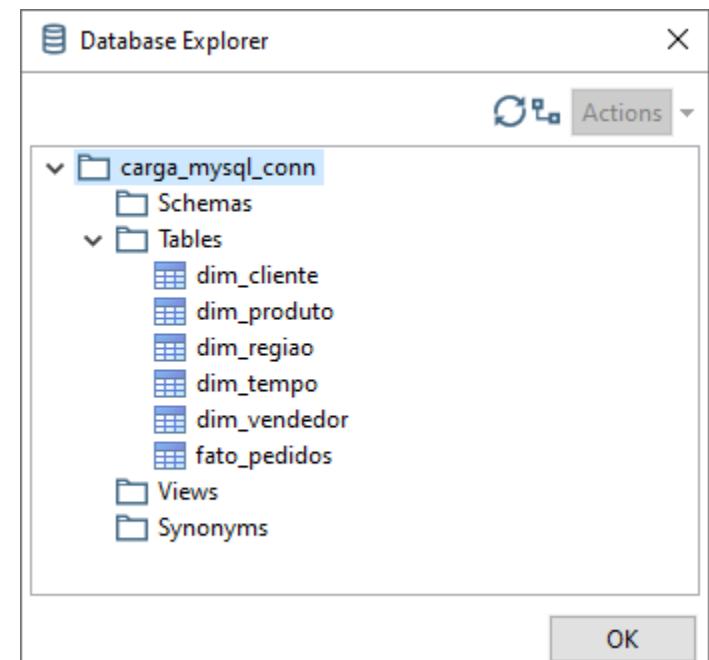
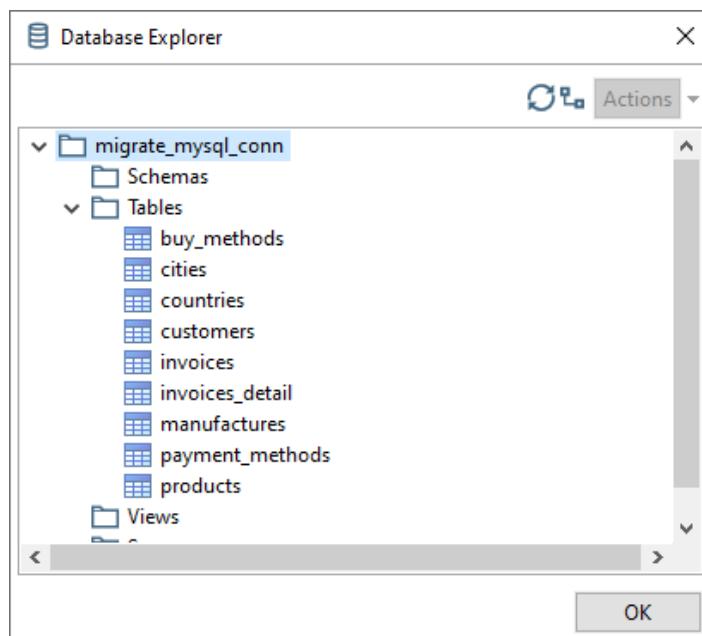
- No menu View do lado esquerdo do Spoon clique com botão direito na pasta conexões e vamos adicionar mais duas. Depois de adicionar compartilhe a conexão na opção *share* na pasta de conexões. Conforme figura abaixo:



# Exercício 1 – Criando as tabelas de um banco de dados

4 – Vamos confirmar se as tabelas foram realmente criadas.

- Na opção *View*, expanda a árvore de conexões e clique com o botão direito em cima da conexão *pdi\_labs\_con*, escolhendo a opção *Explorar*.
- Caso as tabelas tenham sido criadas corretamente, você poderá ver a lista expandindo a opção *Tables*.



# Migração de Bases

# Exercício 2

Carregando as tabelas através de transformações

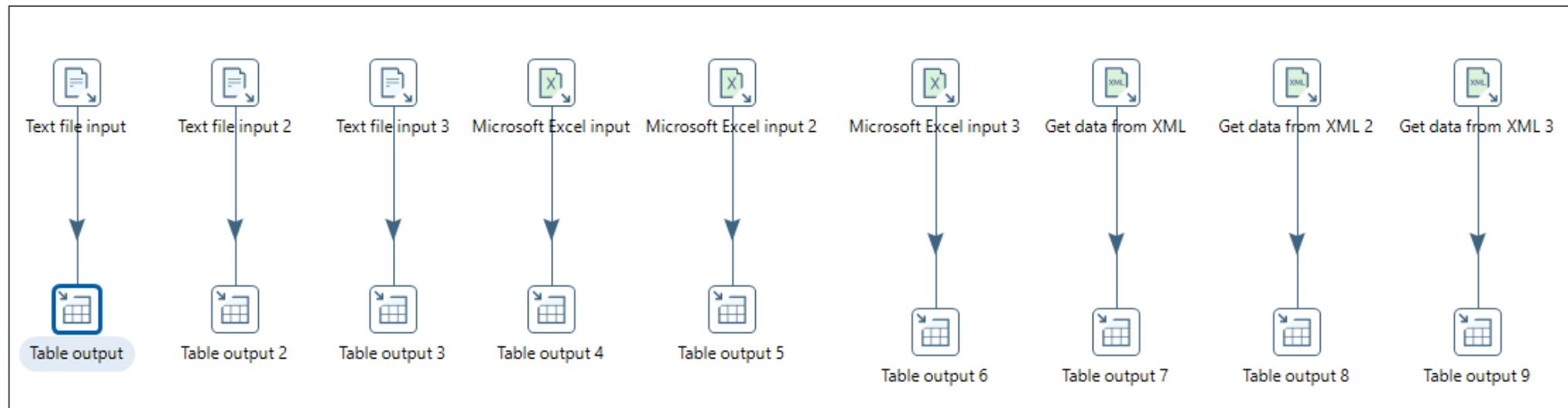
# Exercício 2 – Carregando as tabelas através de transformações

**Exercício:** Carregando as tabelas através de transformações.

**Objetivo:** Carga de Dados.

Depende: do exercício anterior (Ex. 1 LAB IV)

**Resultado:**



## Exercício 2 – Carregando as tabelas através de transformações

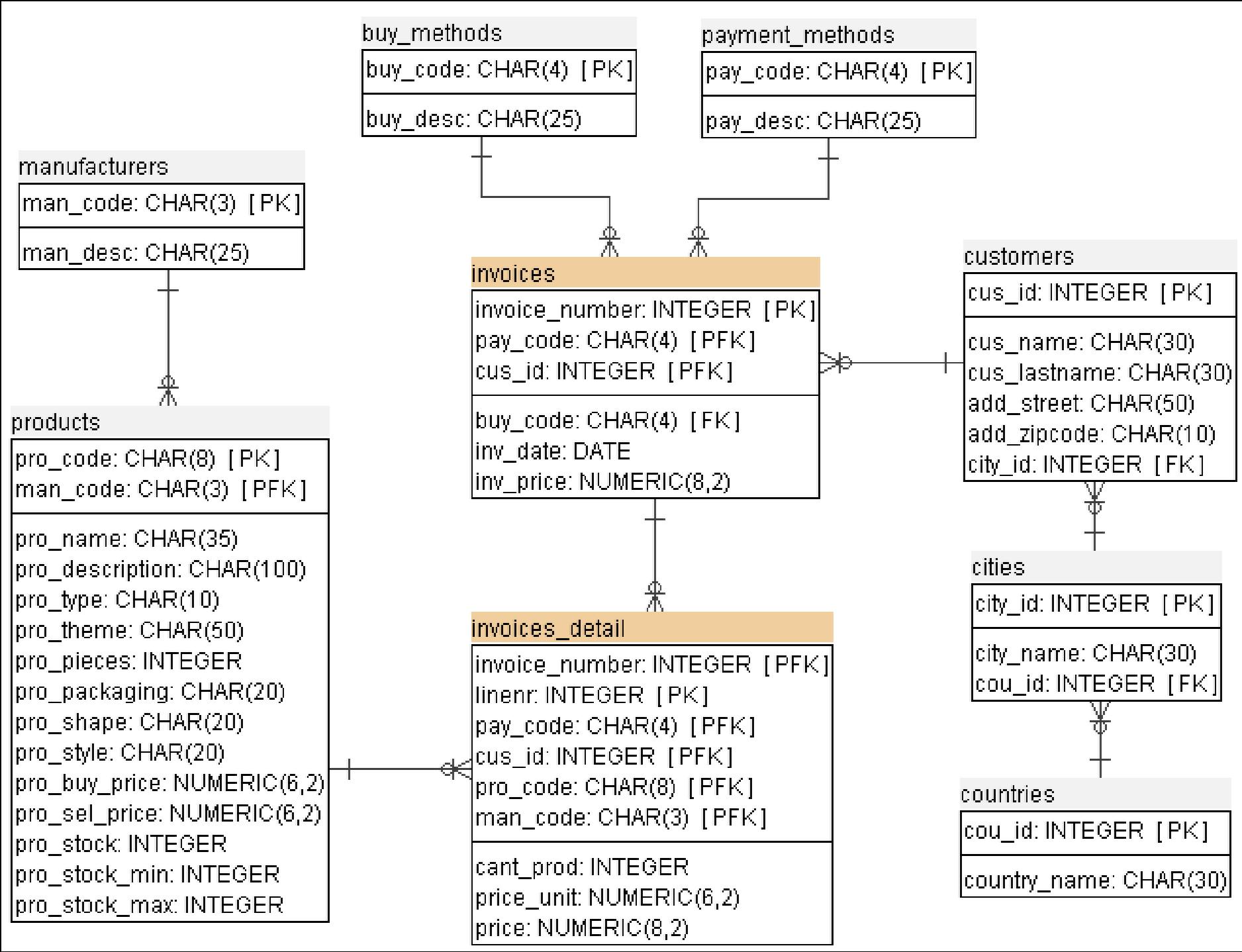
1 – Vamos agora carregar os dados nas tabelas do banco a partir de um conjunto de arquivos localizados na pasta /bases/banco do material distribuído. Os arquivos que você deverá usar estão listados ao lado.

1. ***BUY\_METHODS.csv***
2. ***CITIES.csv***
3. ***COUNTRIES.csv***
4. ***CUSTOMERS.xls***
5. ***INVOICES.xls***
6. ***INVOICES\_DETAIL.xls***
7. ***MANUFACTURERS.xml***
8. ***PAYMENT\_METHODS.xml***
9. ***PRODUCTS.xml***

# Exercício 1 – Criando as tabelas de um banco de dados

- ❖ A base de dados do exemplo possui 9 tabelas, de acordo com uma breve descrição de suas características abaixo:

Tabela	Descrição
<b>manufacturers</b>	Informações sobre os fabricantes dos produtos
<b>products</b>	Produtos que estão à venda na loja, como jogos, quebra-cabeça, acessórios, etc.
<b>buy_methods</b>	Contém informações sobre as modalidades de compra (física, por telefone, pela internet, etc.)
<b>payment_methods</b>	Contém informações sobre as modalidades de pagamento (dinheiro, cartão, cheque)
<b>countries</b>	Uma lista de países
<b>cities</b>	Uma lista de cidades
<b>customers</b>	Os clientes cadastrados na loja
<b>Invoices/invoices_detail</b>	Cabeçalho e detalhe das informações de faturas



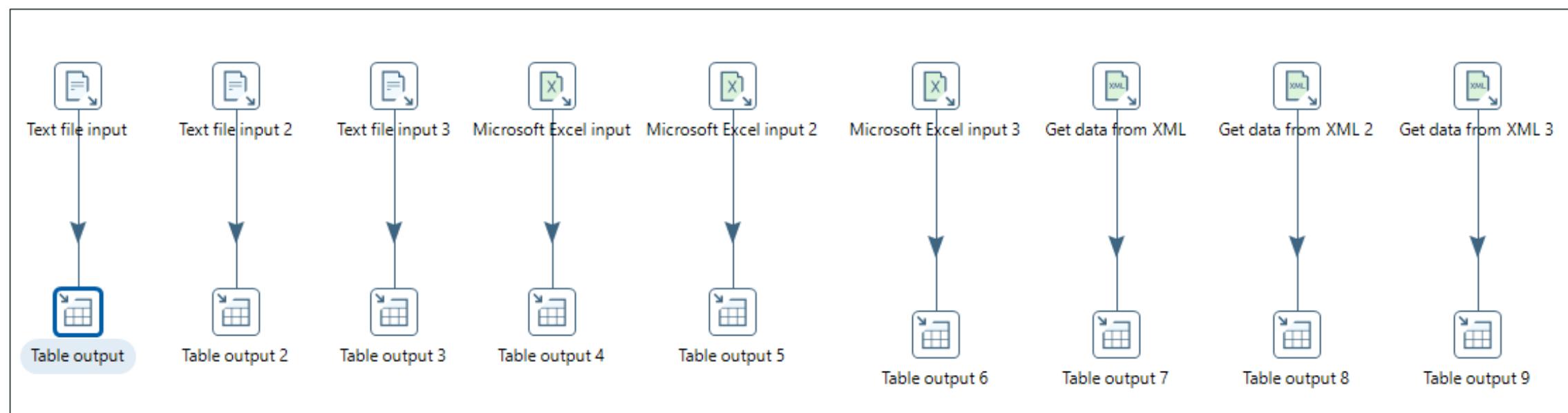
## Exercício 2 – Carregando as tabelas através de transformações

2 – Crie uma nova transformação e adicione vários steps para ler cada arquivo de entrada.

- Você terá que criar 3 *Text File input*, 3 *Excel input* e 3 *Get data from XML*.
- Conforme mostrado nos exercícios anteriores, para cada step de entrada, localize o arquivo e adicione-o à lista de arquivos selecionados.
- Configure o conteúdo na aba *Content* e obtenha os campos na aba *Fields*.

## Exercício 2 – Carregando as tabelas através de transformações

3 – Em seguida, crie 9 steps do tipo *Table output*, que irão armazenar os dados nas tabelas criadas no exercício anterior.

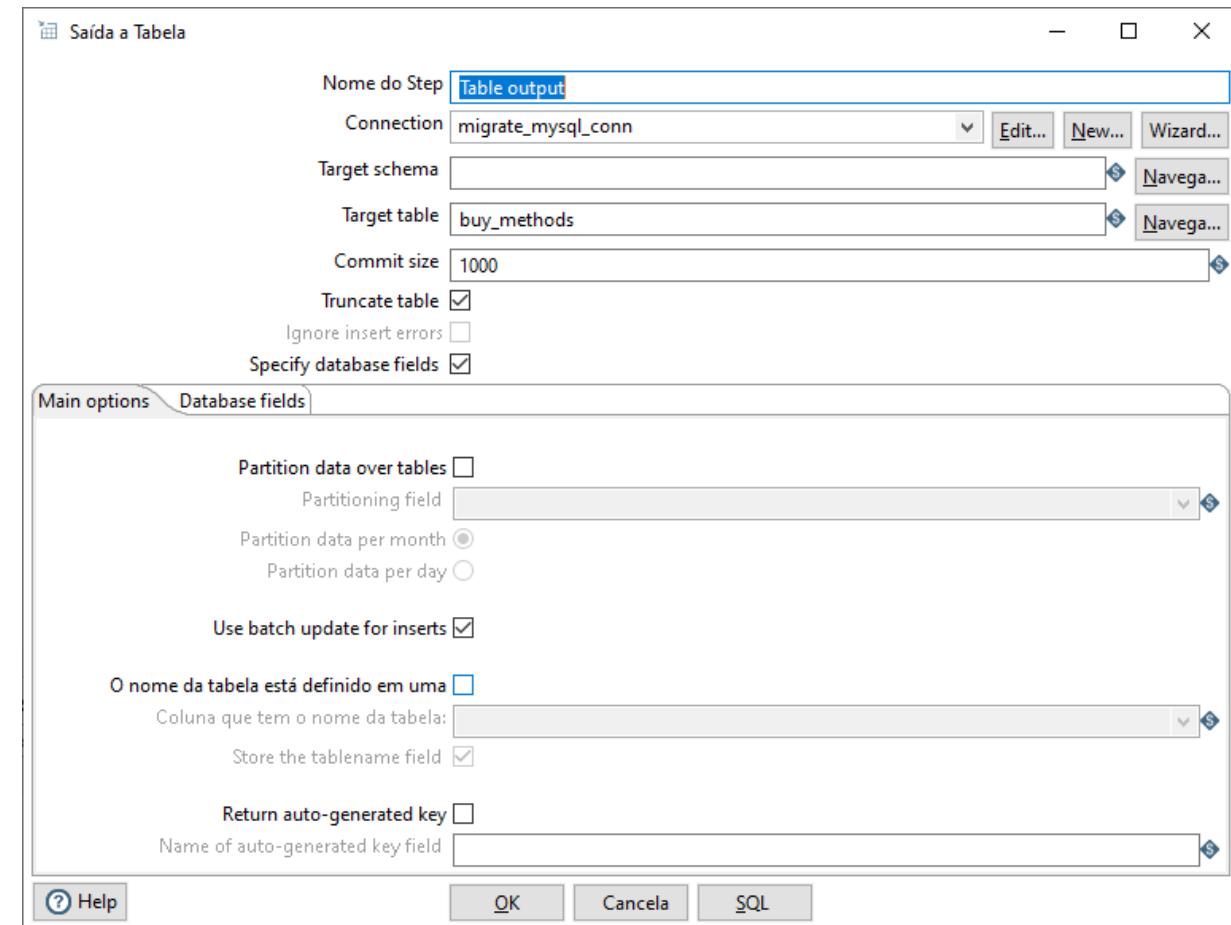


## Exercício 2 – Carregando as tabelas através de transformações

4 – Edite cada *Table output* da seguinte forma:

- *Connection*: escolha *migrate\_mysql\_conn*.
- *Target table*: uma das tabelas criadas no exercício anterior. Clique no botão Navegar para escolher uma tabela da conexão. (ex.: *BUY\_METHODS*)
- *Aba Database Fields*:
  - Clique no botão *Enter Field mapping* e escolha os mapeamentos adequados

Salve a transformação e repita esse passo para cada step *Table output*.



## Exercício 2 – Carregando as tabelas através de transformações

5 – Para verificar se as tabelas foram realmente carregadas, clique na opção *View*, expanda a árvore de conexões e clique com o botão direito em cima da conexão *migrate\_mysql\_conn*, escolhendo a opção explorar. Escolha uma tabela e clique na opção *Preview First 100 rows of <nome\_da\_tabela>*.

# Carregamento do DW

# Atividades

## ❖ Carregamento de Dimensões:

- Exercício 3 – Carga da tabela dimensional de **Tempo**
- Exercício 4 – Carga da tabela dimensional de **Cliente**
- Exercício 5 – Carga da tabela dimensional **Vendedor**
- Exercício 6 – Carga da tabela **Produtos**
- Exercício 7 – Carga da tabela dimensional **Regiões**

## ❖ Carregamento da **Fato**:

- Exercício 8 – Carga da tabela de medida (**Fato**)

# Exercício 3

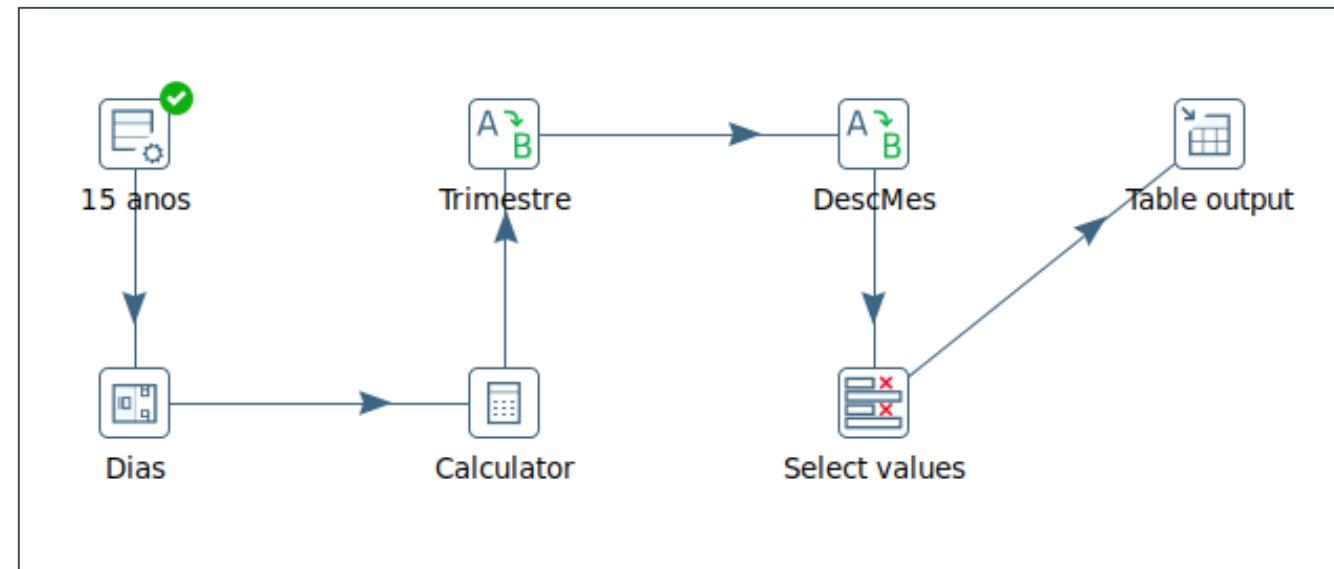
Carga da tabela dimensional de Tempo

## Exercício 3 – Carga da tabela dimensional de Tempo

**Exercício:** Carregar a tabela dimensional **dim\_tempo** com todos os registros de tempo, serão carregados 15 anos de registros (1 para cada dia)

**Objetivo:** Carga da Dimensão de Tempo.

**Resultado:**



## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 1 – O resultado final a tabela deve ficar carregada com dados iguais ao exemplo abaixo:

fk_tempo	mes	trimestre	ano	data	desc_mes
1	1	1	2010	01/01/2010	JAN
2	1	1	2010	02/01/2010	JAN
3	1	1	2010	03/01/2010	JAN
4	1	1	2010	04/01/2010	JAN
5	1	1	2010	05/01/2010	JAN
...	...	...	...	...	....
...	...	...	...	...	....
5474	12	4	2025	29/12/2015	DEZ
5475	12	4	2025	31/12/2015	DEZ

## Exercício 3 – Carga da tabela dimensional de Tempo

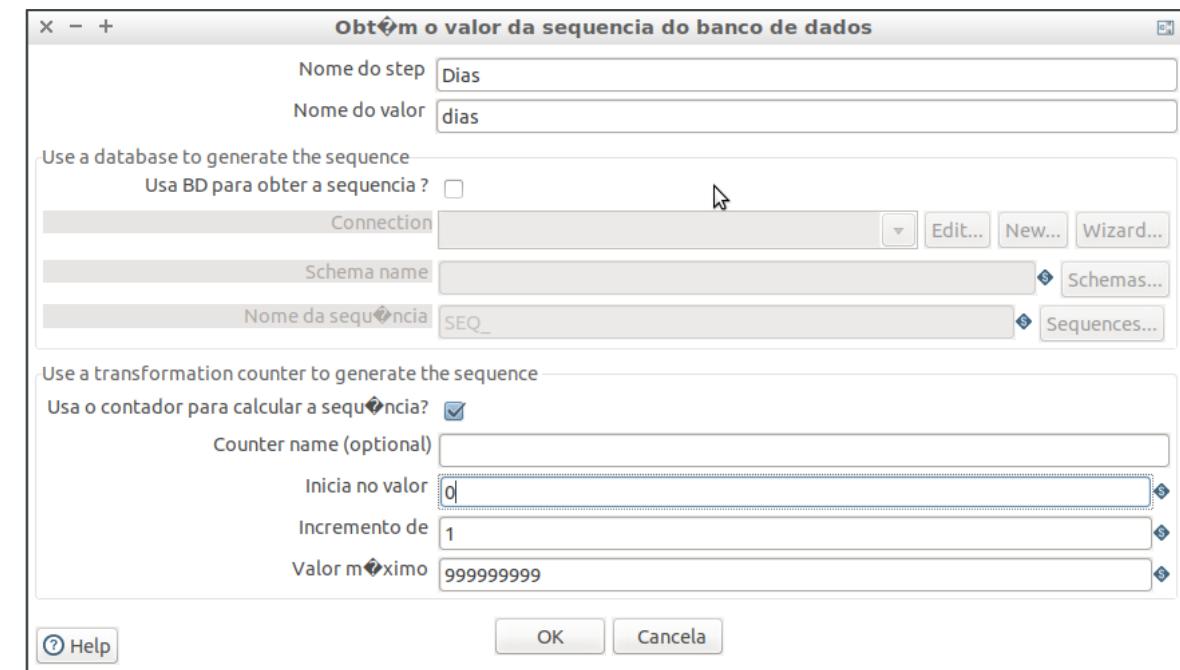
- ❖ 2 - Clique em File / Novo / Transformação para criar uma nova transformação no PDI.
    - Clique em *Design* e no botão + do *Input*
    - Clique e arraste o step *Generate Rows* para a área da transformação
    - Selecione as propriedades do step
    - Configure o nome do step com: 15 anos
    - No campo *limit* informe 5475
    - Na lista de *fields* informe uma variável com o nome "inicio" com o tipo *Date* e o formato como *yyyyMMdd* e no valor informe 20100101

**Generate rows**

Nome do Step	15 anos								
Limit	5475								
Never stop generating rows	<input type="checkbox"/>								
Interval in ms (delay)	5000								
Current row time field name	now								
Previous row time field name	FiveSecondsAgo								
<b>Fields :</b>									
Nome	Tipo	Formato	Tamanho	Precisão	Moeda	Decimal	Grupo	Valor	Set empty string?
1 inicio	Date	yyyyMMdd						20100101	N

## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 3 - Insira um step *Add Sequences* (Transform) na transformação e ligue o mesmo com o *Step* anterior.
- Na propriedade nome *step* coloque como "Dias" e no nome do valor informe "dias"
  - Na propriedade Inicial do valor informe 0
  - Na propriedade incremento de valor informe 1
  - Na propriedade *Valor máximo* informe 10000000



# Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 5 - Insira um *step Calculator* ( *Transform* ) e insira um hop de ligação com o *step* anterior.
  - Configure o *step calculator* conforme imagem abaixo:

**Calculadora**

Nome do Step

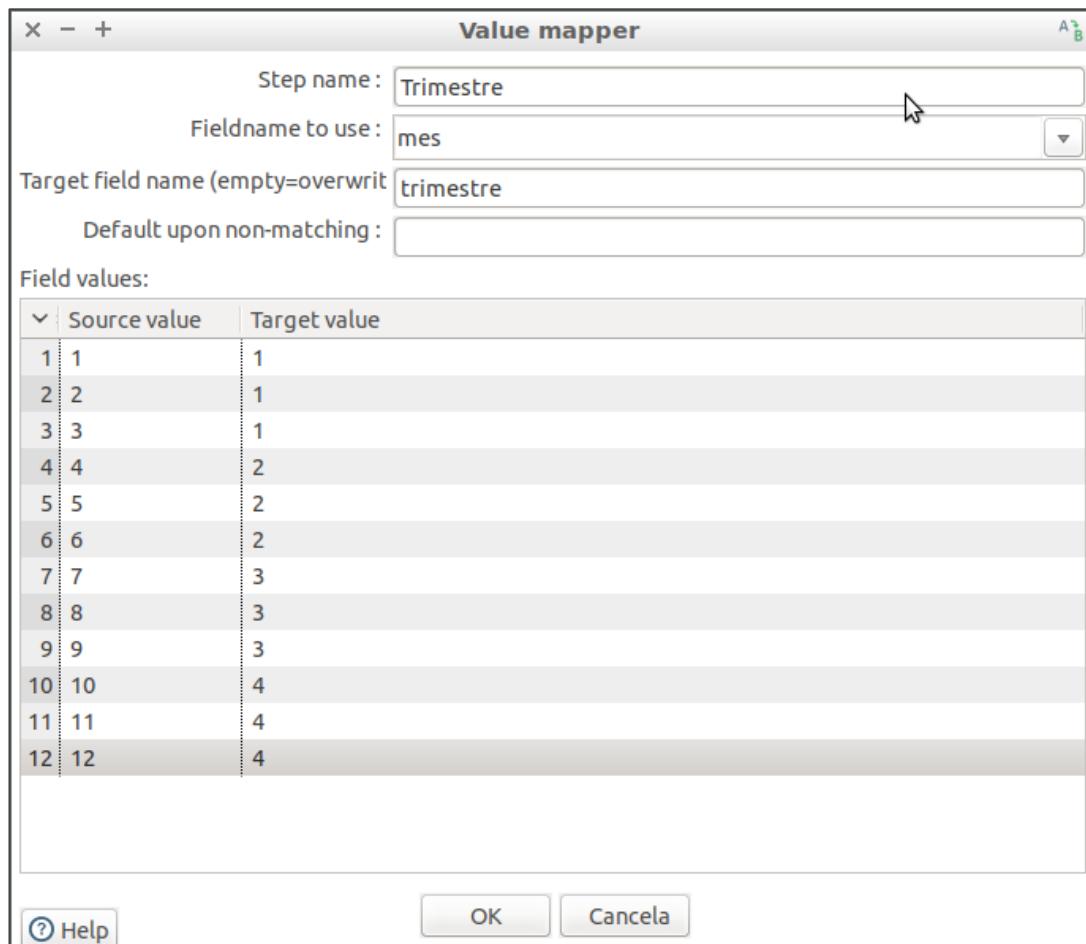
Throw an error on non existing files

Campos:

	Novo campo	Calculo	Campo A	Campo B	Campo C	Tipo do valor	Tamanho	Precisão	Remove	Coluna
1	um	Define o valor constante A para o campo	1			Integer	1	0	S	
2	data	Data A + B Dias	inicio	dias		Date			N	
3	ano	Ano da data A	data			Integer	4	0	N	
4	mes	Mês da data A	data			Integer	2	0	N	
5	fk_tempo	A + B	dias	um		Integer		0	N	

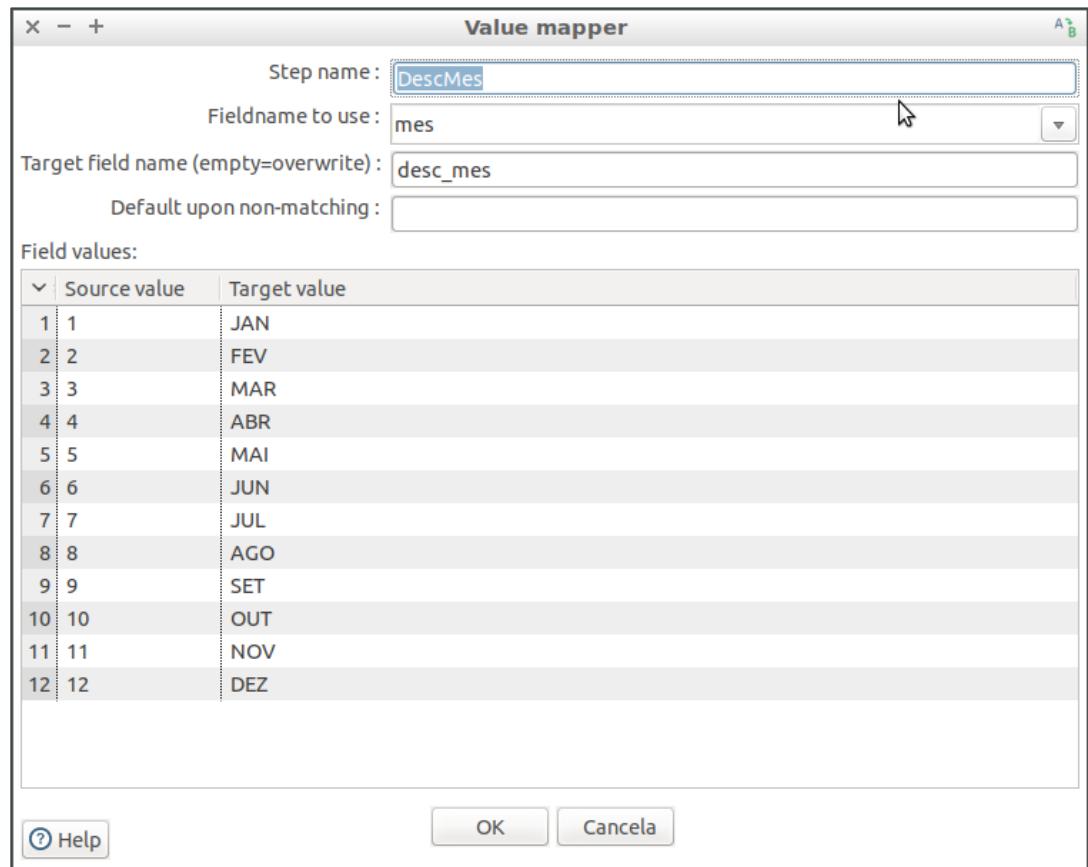
## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 6 - Insira um *step Value Mapper Transform* ligue ao *step* anterior com um *hop* e configure conforme abaixo:



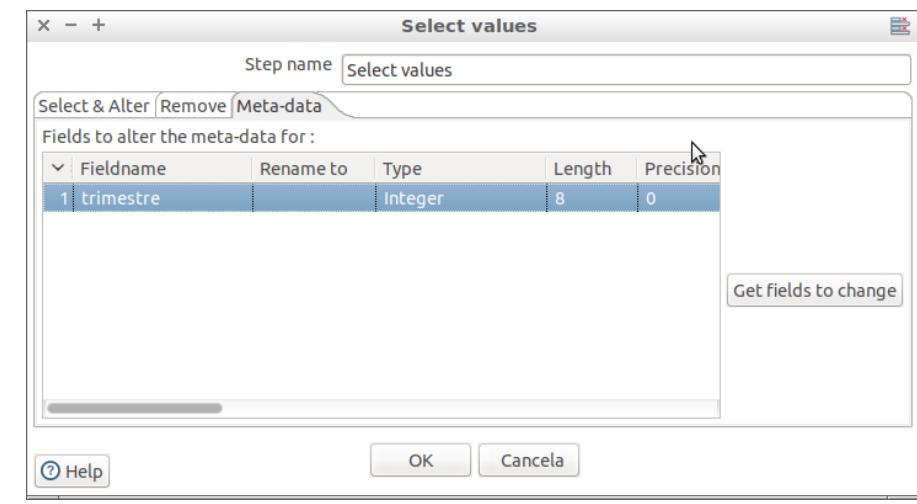
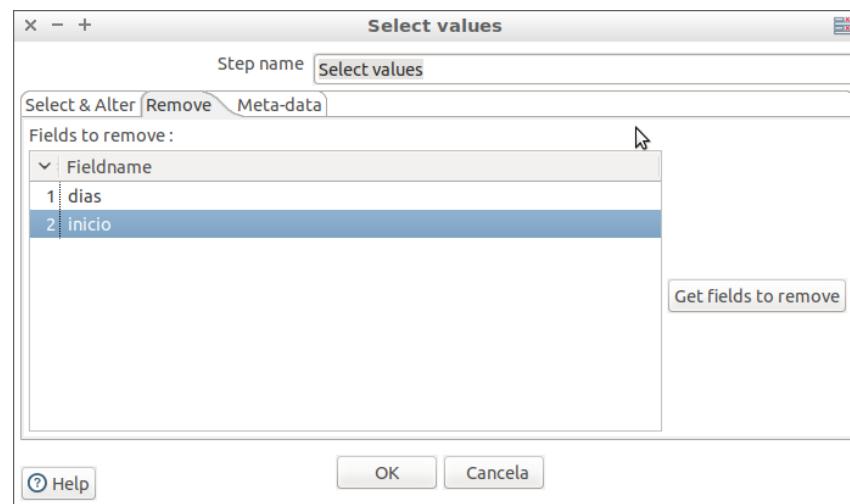
## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 7 - Insira mais um step Value Mapper e ligue com um hop ao step anterior e configure conforme imagem abaixo:



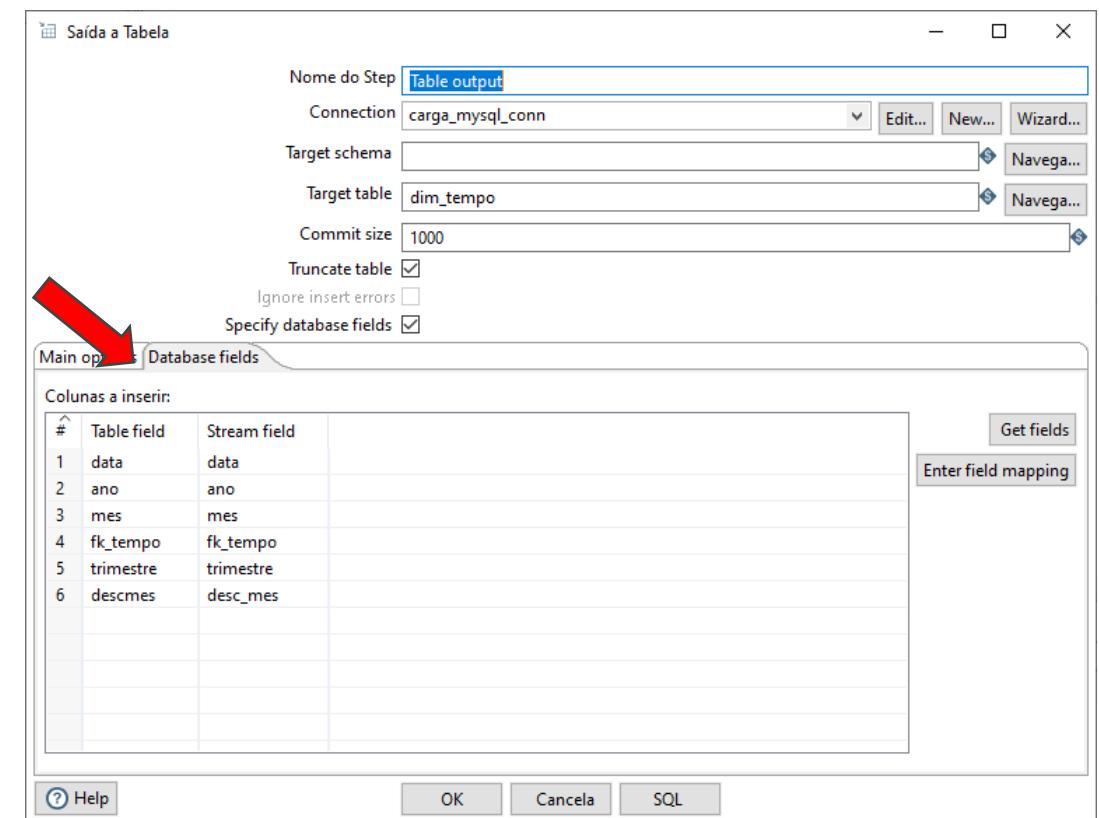
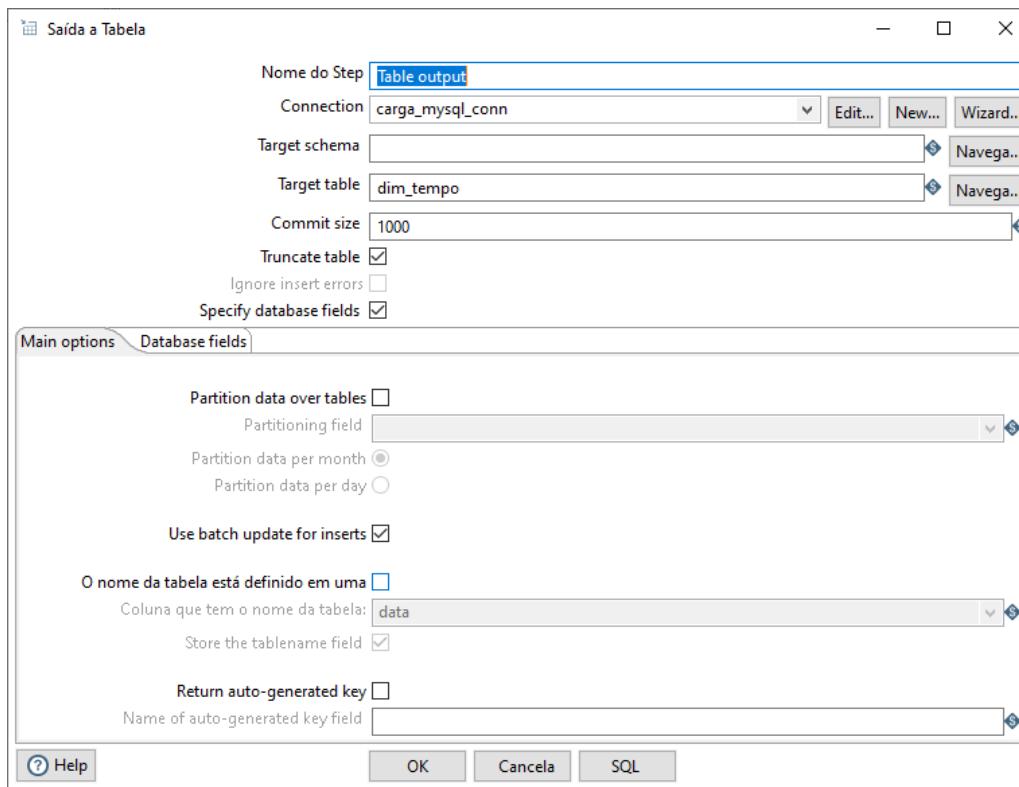
## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 8 - Insira um *step Select / rename values* , adicione um hop entre os steps, e configure a aba remove para eliminar as variáveis que não serão mais utilizadas , conforme imagem abaixo.
- ❖ 9 - Selecione a Aba *Meta-Data* para alterar o campo Trimestre de *String* para *Inteiro* (*Integer*)



# Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 10 - Insira na transformação o step Table Output (categoria output), o mesmo será responsável pela gravação dos dados na base *datawarehouse\_olap*, tabela *dim\_tempo* (dimensão de tempo), crie o hop de ligação e configure o step conforme imagem abaixo



## Exercício 3 – Carga da tabela dimensional de Tempo

- ❖ 11 - Salve e execute a transformação.

Examine preview data

Rows of step: dim\_tempo (100 rows)

v	fk_tempo	mes	trimestre	ano	data	descmes
1	1	1	1	2010	2010/01/01 00:00:00.000	JAN
2	2	1	1	2010	2010/01/02 00:00:00.000	JAN
3	3	1	1	2010	2010/01/03 00:00:00.000	JAN
4	4	1	1	2010	2010/01/04 00:00:00.000	JAN
5	5	1	1	2010	2010/01/05 00:00:00.000	JAN
6	6	1	1	2010	2010/01/06 00:00:00.000	JAN
7	7	1	1	2010	2010/01/07 00:00:00.000	JAN
8	8	1	1	2010	2010/01/08 00:00:00.000	JAN
9	9	1	1	2010	2010/01/09 00:00:00.000	JAN
10	10	1	1	2010	2010/01/10 00:00:00.000	JAN
11	11	1	1	2010	2010/01/11 00:00:00.000	JAN
12	12	1	1	2010	2010/01/12 00:00:00.000	JAN
13	13	1	1	2010	2010/01/13 00:00:00.000	JAN
14	14	1	1	2010	2010/01/14 00:00:00.000	JAN
15	15	1	1	2010	2010/01/15 00:00:00.000	JAN
16	16	1	1	2010	2010/01/16 00:00:00.000	JAN
17	17	1	1	2010	2010/01/17 00:00:00.000	JAN
18	18	1	1	2010	2010/01/18 00:00:00.000	JAN
19	19	1	1	2010	2010/01/19 00:00:00.000	JAN
20	20	1	1	2010	2010/01/20 00:00:00.000	JAN

Fecha

# Exercício 4

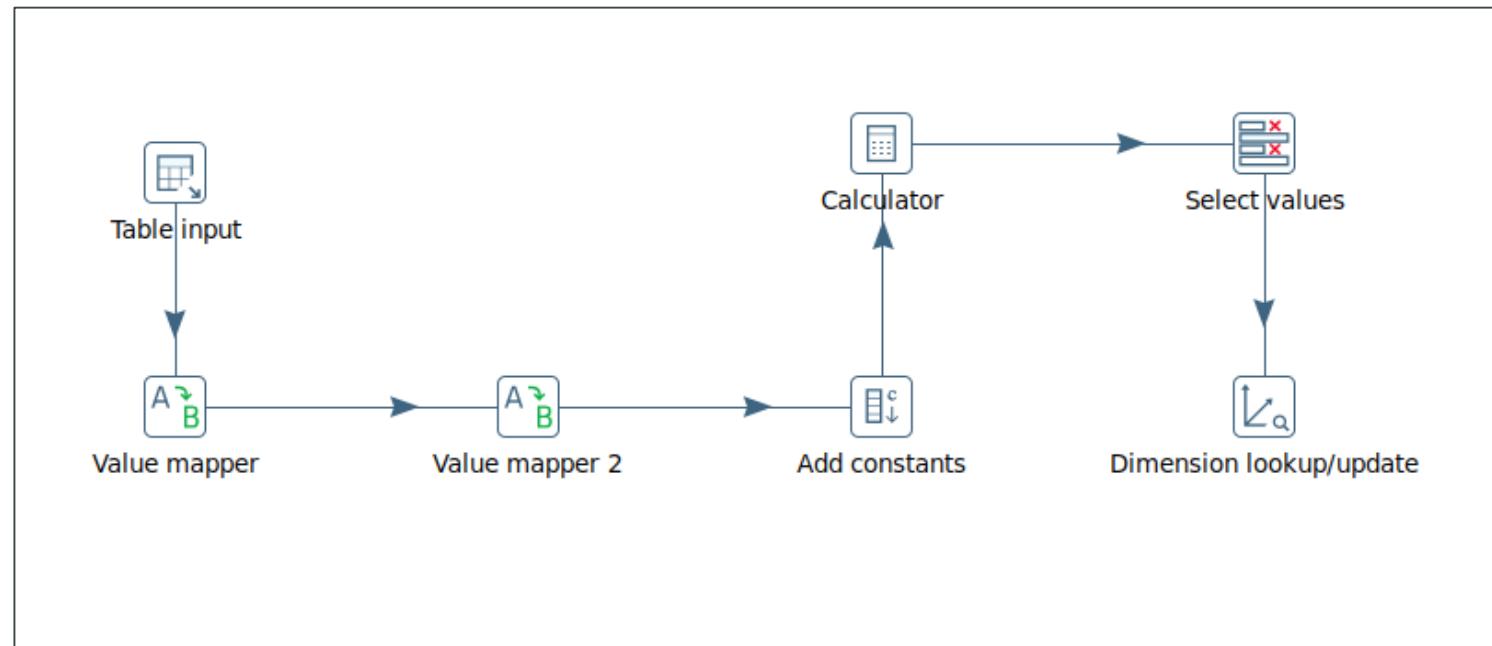
Carga da tabela dimensional de Cliente

## Exercício 4 – Carga da tabela dimensional de Cliente

**Exercício:** Popular a tabela dimensão de clientes (`dim_cliente`), esta tabela terá versionamento de registros caso ocorram alterações do mesmo.

**Objetivo:** Carga da Dimensão Cliente

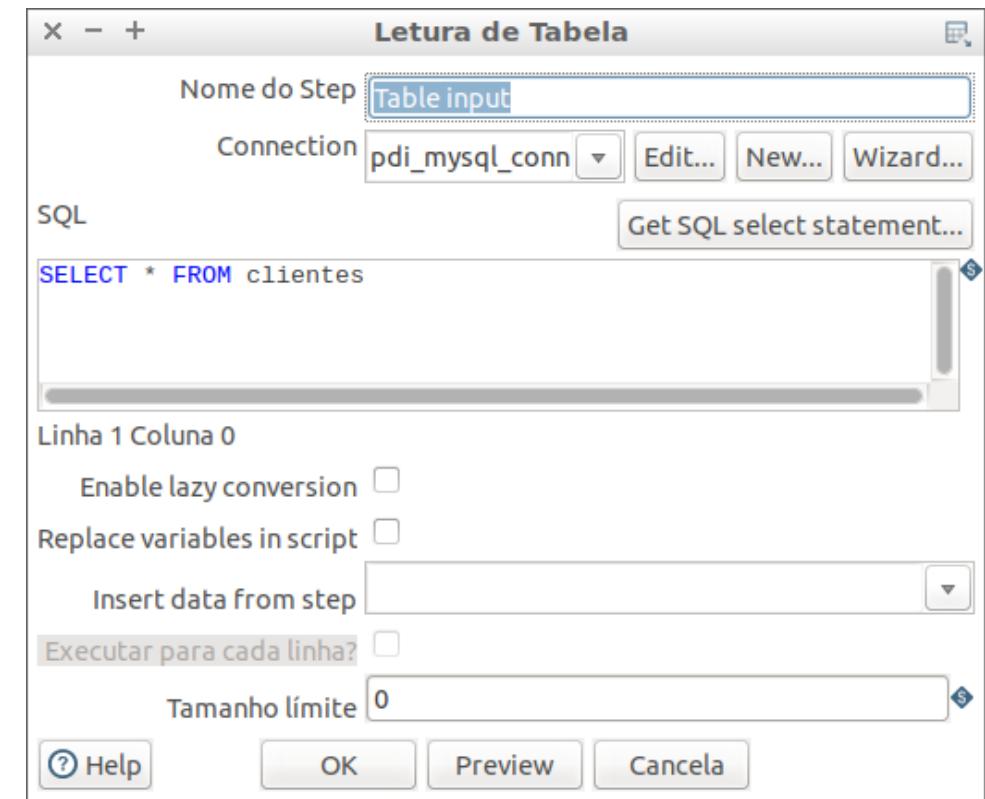
**Resultado:**



## Exercício 4 – Carga da tabela dimensional de Cliente

❖ 1 - Clique em File / Novo / Transformação para criar uma nova transformação no PDI.

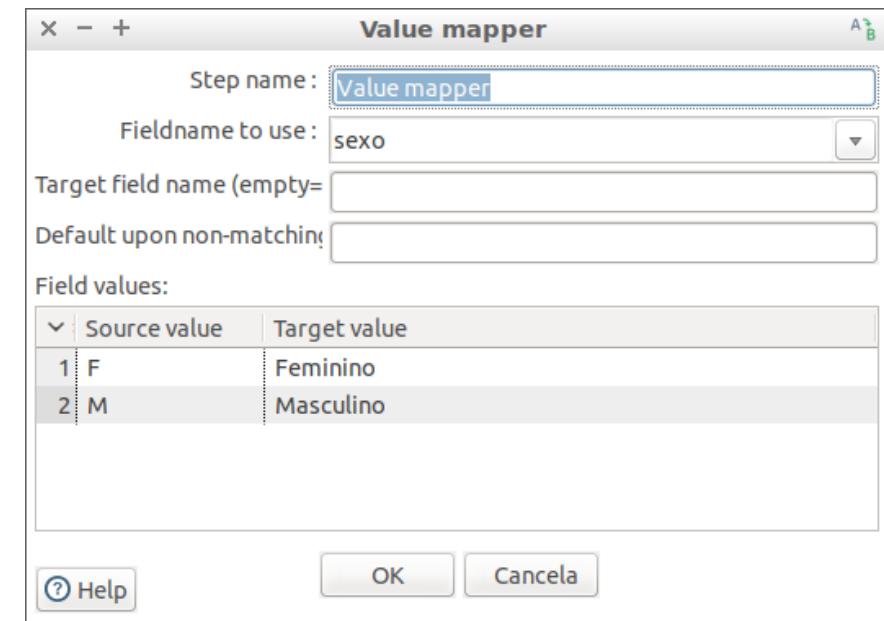
- Clique em Design e no botão + do Input
- Clique e arraste o step *Table input* para a área da transformação.
- Nas propriedades do step selecione a conexão *pdi\_mysql\_conn* (Ex. 3 LAB I).
- Selecione a tabela clientes, com todos os campos.
  - "SELECT \* FROM clientes"
- Confira se as propriedades do *step* ficaram conforme imagem a lado.



## Exercício 4 – Carga da tabela dimensional de Cliente

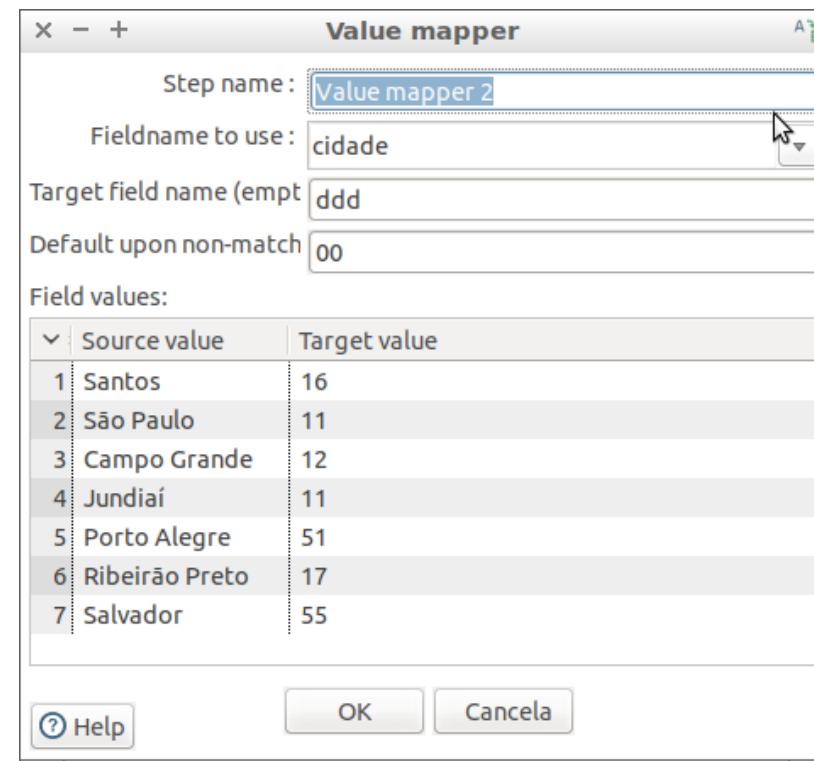
❖ 2 - Adicione um *step Value Mapper* (Transform), para fazermos o mapeamento do campo sexo , quem vem preenchido com F e M e deve ser alterado para Feminino e Masculino respectivamente.

- Ligue os steps com um hop.
- Configura as propriedades do step , informe o step name como SEXO e o *fieldname to use*, informe o campo que esta na base de dados com os objetos a serem mapeados, neste caso o campo sexo.
- Configure o Field values conforme ao lado.



## Exercício 4 – Carga da tabela dimensional de Cliente

- ❖ 3 - Adicione um step Value Mapper ( Transform ), para fazermos o mapeamento do campo ddd, e configure conforme imagem abaixo, nesta caso se a cidade não for encontrada na lista , o código de ddd padrão será “00” , informado na propriedade *Default*.



## Exercício 4 – Carga da tabela dimensional de Cliente

- ❖ 4 - Adicione um step Add constants , e nele configura conforme abaixo: Obs : No campo espaço\_nome adicione um espaço “ ” no atributo valor!

**Adiciona valores constantes**

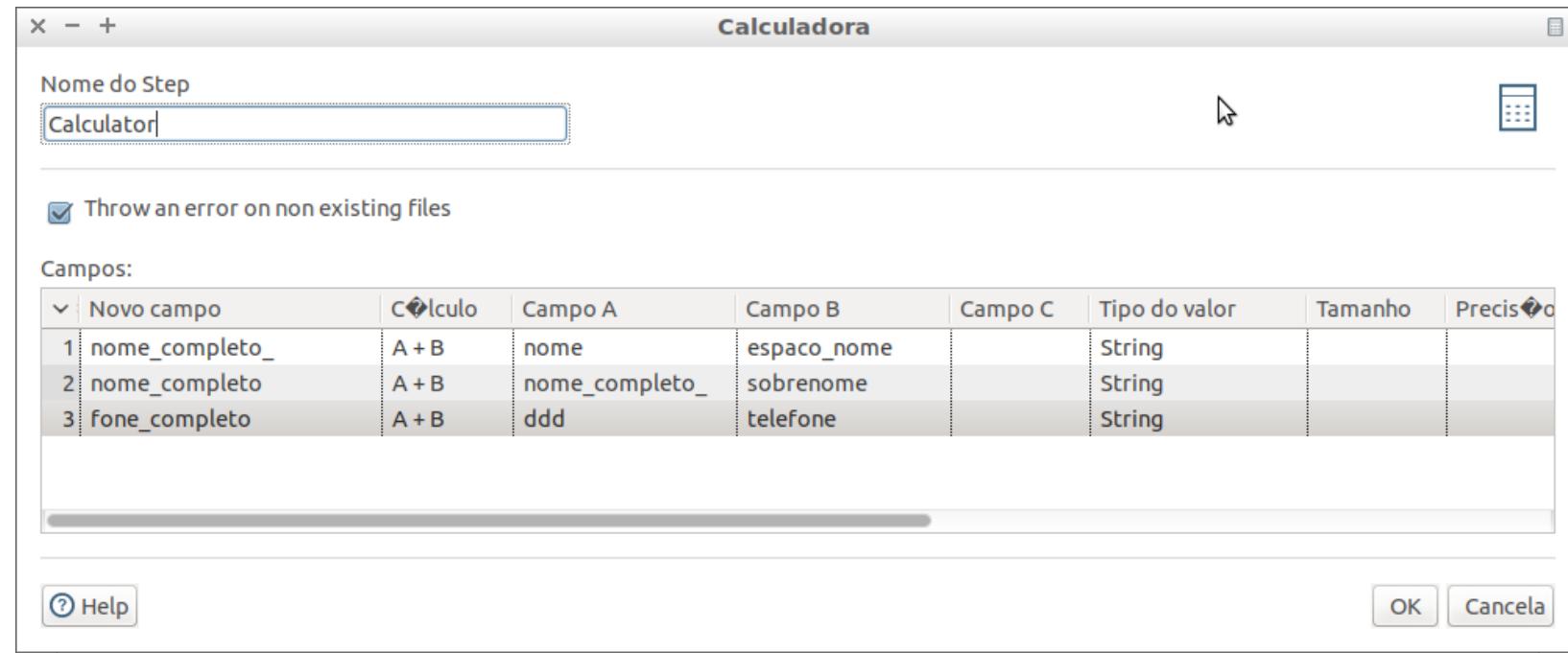
Nome do Step **Add constants**

Campos :	Nome	Tipo	Formato	Tamanho	Precisão	Moeda	Decimal	Grupo	Valor	Set empty string?
	espaço_nome	String		1						N
	versao	Integer							1	N
	data_inicial	Date	yyyy/MM/dd						2019/01/01	N
	data_final	Date	yyyy/MM/dd						2099/12/31	N

OK Cancela

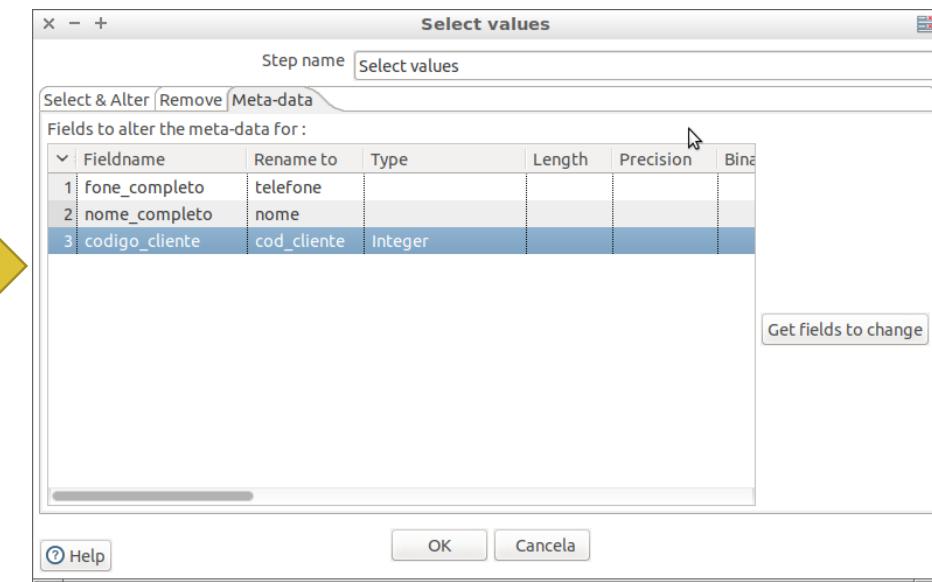
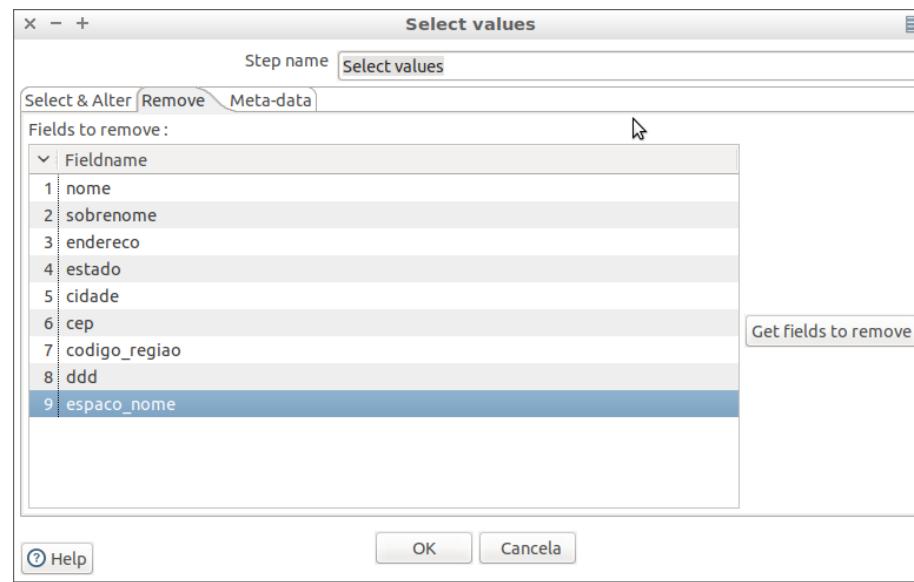
## Exercício 4 – Carga da tabela dimensional de Cliente

- ❖ 5 - Adicione um step calculator e configura conforme abaixo:



# Exercício 4 – Carga da tabela dimensional de Cliente

- ❖ 6 - Adicione um step Select / Values e configure as duas abas conforme imagens abaixo:



- ❖ 7 - Adicione um step Dimension lookup/update, e configura conforme parâmetros abaixo, configure as duas abas:

**Dimension lookup/update**

Step name: Dimension lookup/update

Update the dimension?

Connection: carga\_mysql\_conn

Target schema:

Target table: dim\_cliente

Commit size: 100

Enable the cache?

Pre-load the cache?

Cache size in rows (0 = cache all): 5000

**Keys** **Fields**

Key fields (to look up row in dimension):

Dimension field	Field in stream
1 cod_cliente	cod_cliente

Technical key field: fk\_cliente

Creation of technical key:

- Use table maximum + 1
- Use sequence
- Use auto increment field

Version field: versao

Stream Datefield:

Date range start field: data\_inicial  Min. year: 2019

Use an alternative start date?  <Select Option>

Table date range end: data\_final  Max. year: 2199

**Dimension lookup/update**

Step name: Dimension lookup/update

Update the dimension?

Connection: carga\_mysql\_conn

Target schema:

Target table: dim\_cliente

Commit size: 100

Enable the cache?

Pre-load the cache?

Cache size in rows (0 = cache all): 5000

**Keys** **Fields**

Lookup/Update fields

Dimension field	Stream field to compare with	Type of dimension update
1 telefone	telefone	Insert
2 sexo	sexo	Insert
3 estado_civil	estado_civil	Insert
4 nome	nome	Insert
5 telefone	telefone	Insert

Technical key field: fk\_cliente

Creation of technical key:

- Use table maximum + 1
- Use sequence
- Use auto increment field

Version field: versao

Stream Datefield:

Date range start field: data\_inicial  Min. year: 2019

Use an alternative start date?  <Select Option>

Table date range end: data\_final  Max. year: 2199

# Exercício 4 – Carga da tabela dimensional de Cliente

- ❖ 8 – Salve e execute a transformação.

v	cod_cliente	endereco	sexo	estado_civil	versao	data_inicial	data_final	nome_completo_	nome	telefone
1	51	5462 Ac Road	Masculino	Solteiro	1	2019/01/01	2099/12/31	Sydney	Sydney Vincent	001 55 547 7624-9315
2	52	850-4985 Ullamcorper, Avenue	Masculino	Casado	1	2019/01/01	2099/12/31	Judah	Judah Franco	001 55 652 2761-0777
3	53	P.O. Box 312, 2893 Metus Ave	Masculino	Casado	1	2019/01/01	2099/12/31	Inez	Inez Cruz	511 55 436 2072-7494
4	54	Ap #586-997 Luctus Rd.	Masculino	Viudo	1	2019/01/01	2099/12/31	Lee	Lee Harper	001 55 514 5962-4672
5	55	P.O. Box 457, 1977 Lectus Av.	Feminino	Viudo	1	2019/01/01	2099/12/31	Desiree	Desiree Landry	551 55 102 9643-2150
6	56	Ap #783-1492 Enim Av.	Masculino	Viudo	1	2019/01/01	2099/12/31	Raven	Raven Emerson	111 55 203 8561-7210
7	57	Ap #242-818 Sollicitudin Avenue	Feminino	Viudo	1	2019/01/01	2099/12/31	Dustin	Dustin Terry	001 55 796 3570-8804
8	58	Ap #459-9003 Et Road	Feminino	Viudo	1	2019/01/01	2099/12/31	Molly	Molly Horne	001 55 802 5722-6379
9	59	Ap #481-6790 Nisl. Road	Masculino	Viudo	1	2019/01/01	2099/12/31	Rajah	Rajah Chase	001 55 643 4760-0305
10	60	P.O. Box 255, 6409 Luctus St.	Masculino	Solteiro	1	2019/01/01	2099/12/31	Kiayada	Kiayada McCormick	001 55 154 7258-5534
11	61	Ap #160-1936 Neque St.	Masculino	Casado	1	2019/01/01	2099/12/31	Breanna	Breanna Hyde	171 55 169 3178-6135
12	62	2843 Nibh. Rd.	Masculino	Viudo	1	2019/01/01	2099/12/31	Vivian	Vivian Pacheco	161 55 160 4781-4059
13	63	Ap #728-4137 Facilisis Avenue	Masculino	Solteiro	1	2019/01/01	2099/12/31	Glenna	Glenna Wall	551 55 856 5489-4394
14	64	P.O. Box 137, 5886 Id Street	Feminino	Casado	1	2019/01/01	2099/12/31	Lani	Lani Huber	001 55 311 6010-3140
15	65	5537 Sed Ave	Masculino	Viudo	1	2019/01/01	2099/12/31	Kaye	Kaye Rivera	001 55 399 9910-3253
16	66	870-9660 Lectus Rd.	Feminino	Solteiro	1	2019/01/01	2099/12/31	Warren	Warren McLaughlin	001 55 397 3186-3390
17	67	Ap #929-2638 Ac, Avenue	Masculino	Casado	1	2019/01/01	2099/12/31	Jessamine	Jessamine Nicholson	001 55 848 9243-6336

# Exercício 5

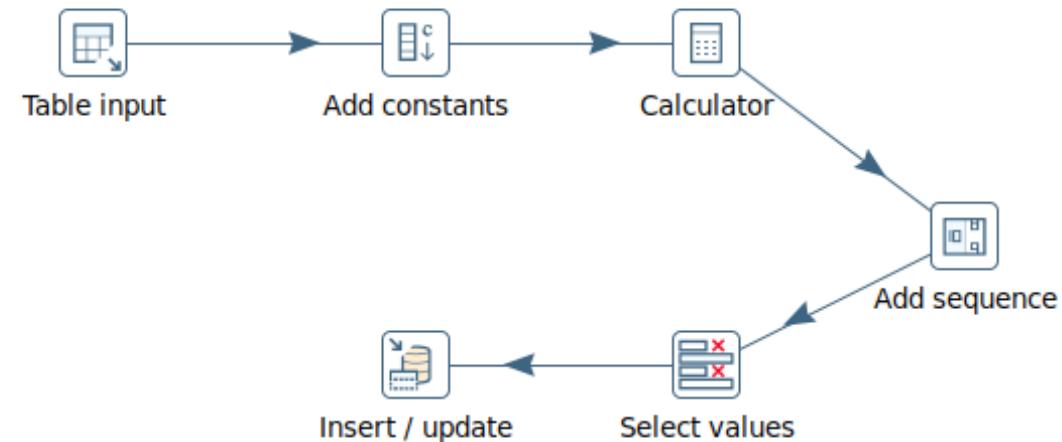
Carga da tabela dimensional Vendedor

## Exercício 5 – Carga da tabela dimensional Vendedor

**Exercício:** Popular a tabela dimensão de vendedor (**dim\_vendedor**), esta tabela **NÃO** terá versionamento de registros caso ocorram alterações do mesmo.

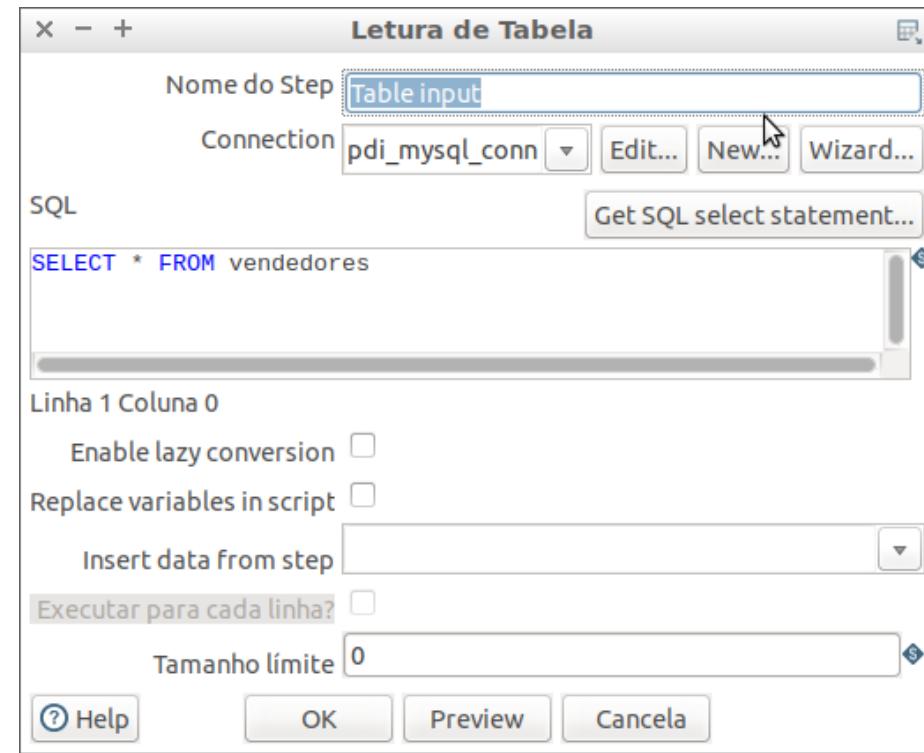
**Objetivo:** Carga da Dimensão Vendedor

**Resultado:**



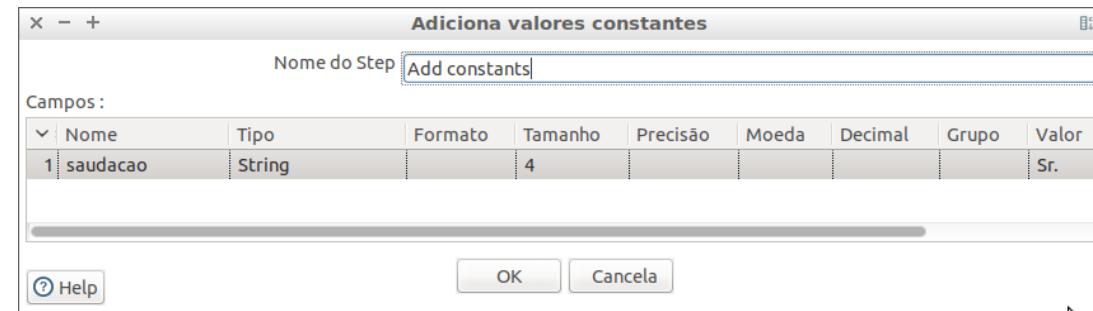
## Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 1 - Adicione um step *table input* e configure com a tabela de vendedores da base erp\_oltp, conforme imagem abaixo:

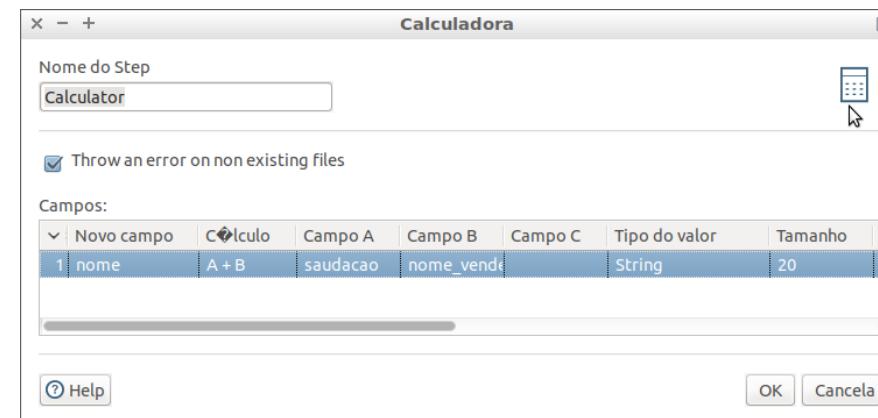


# Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 2 - Adicione um step add constants e configure conforme abaixo:

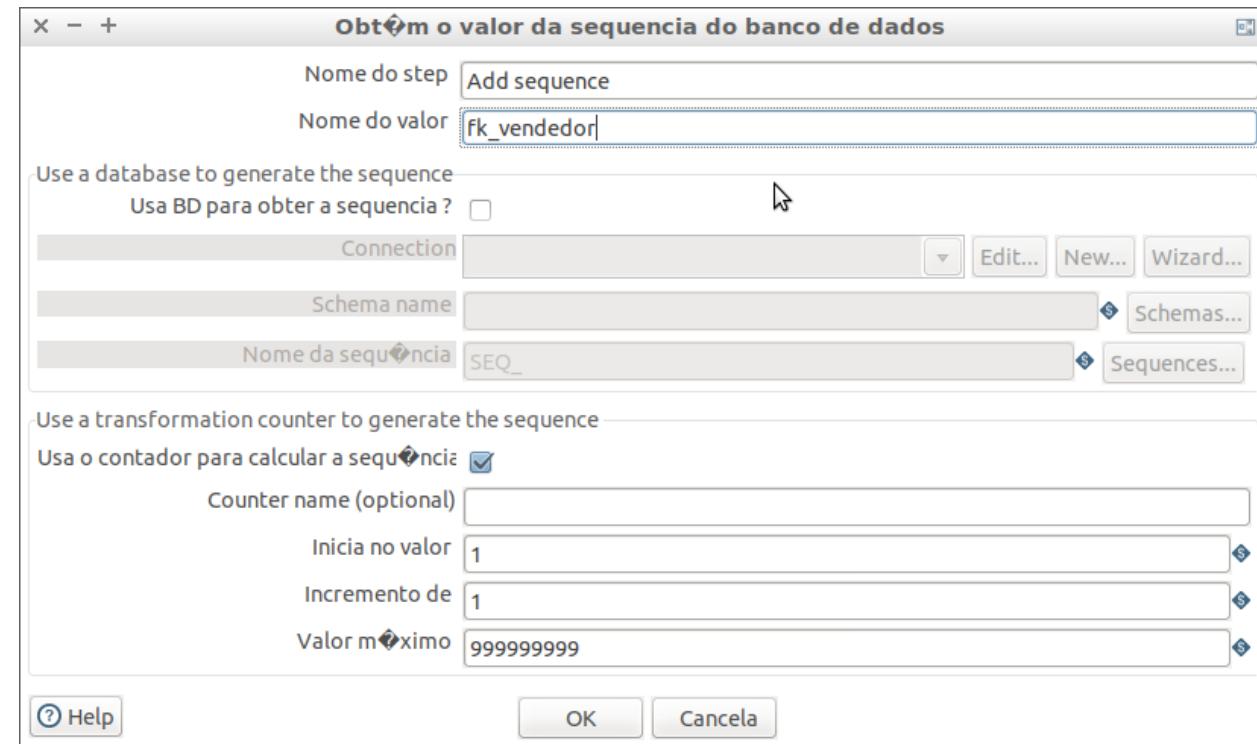


- ❖ 3 - Adicione um step calculator para somar a saudação ao nome do vendedor, configure conforme abaixo



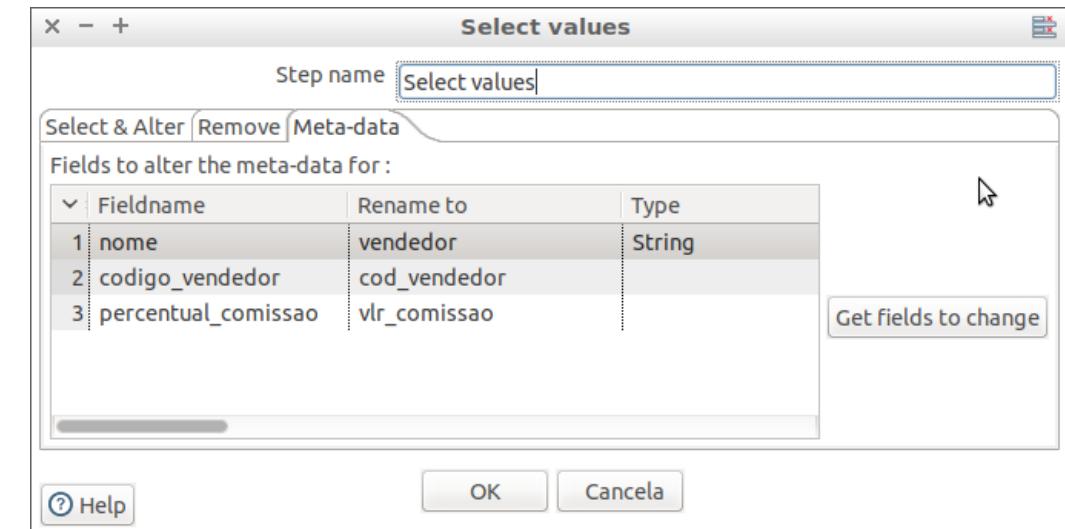
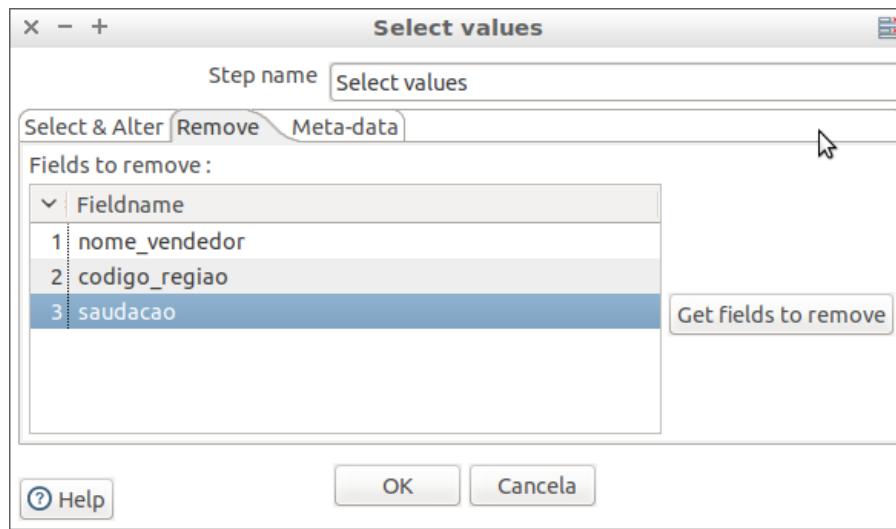
## Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 4 - Adicione um step add Sequences , para criar uma chave primaria independente na base de dados OLAP. Configure conforme abaixo:



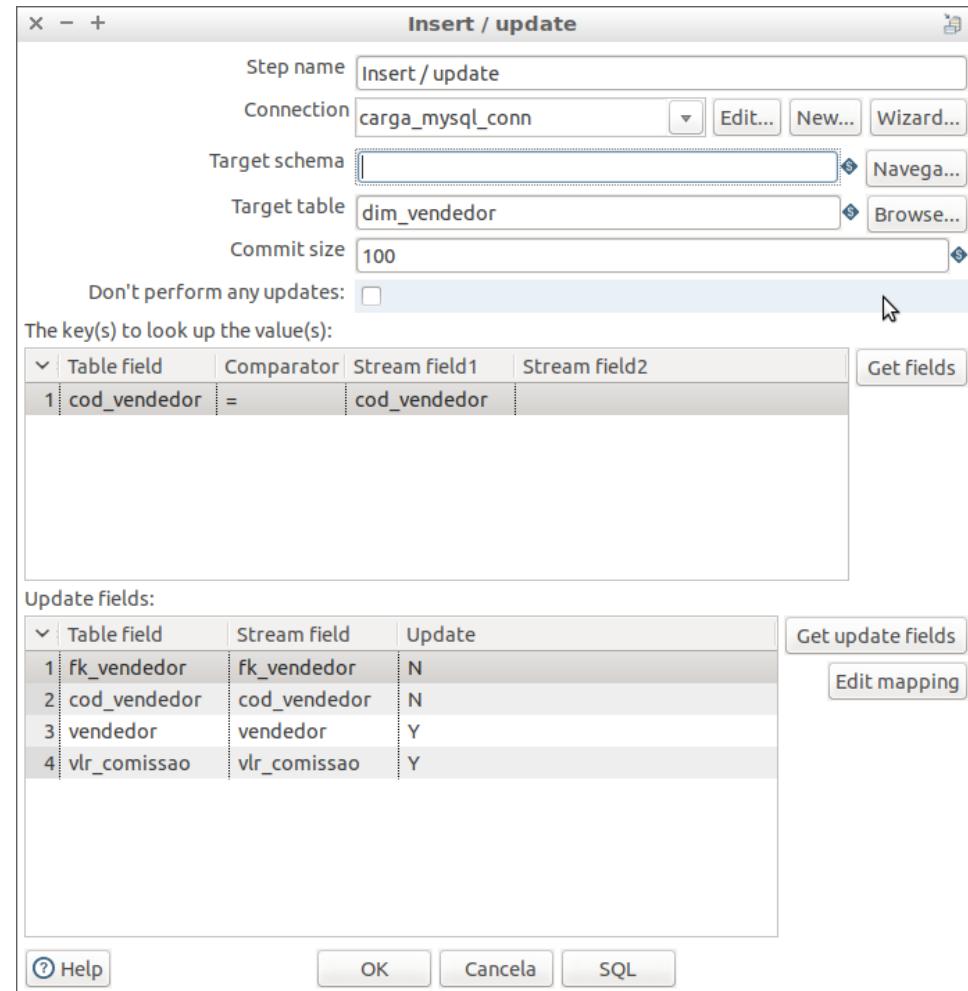
# Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 5 - Adicione um step *Select Values*, e configure conforme abaixo:



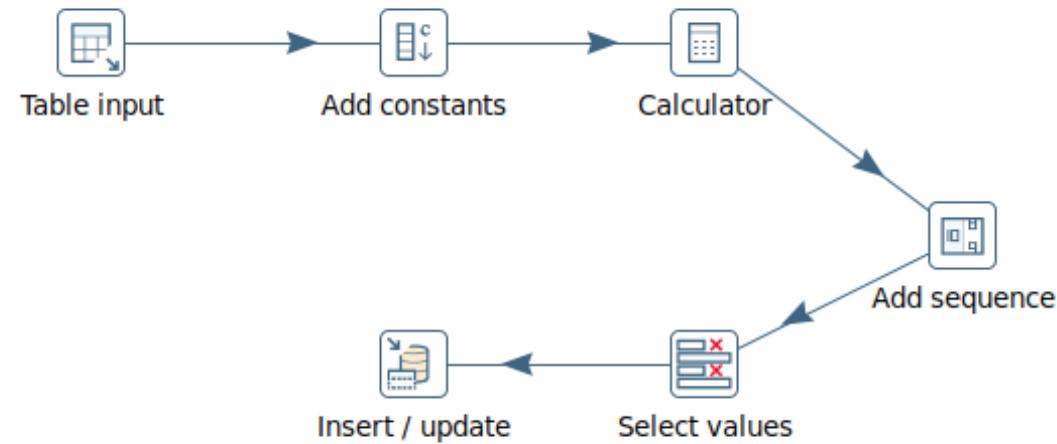
# Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 6 - Adicione um step *Insert/ Update* e configure conforme ao lado:
- ❖ Note que estamos sobrescrevendo algumas informações.



## Exercício 5 – Carga da tabela dimensional Vendedor

- ❖ 7 - Salve e execute a transformação, a mesma deve ficar semelhante a esta imagem:



# Exercício 6

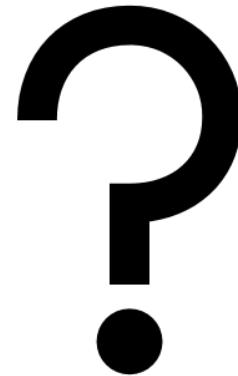
Carga da tabela Produtos

## Exercício 6 – Carga da tabela Produtos

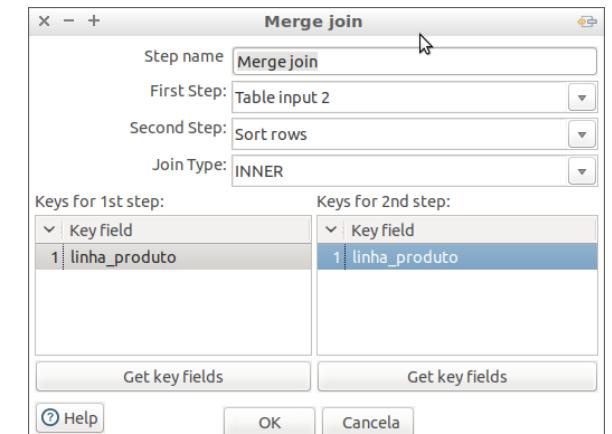
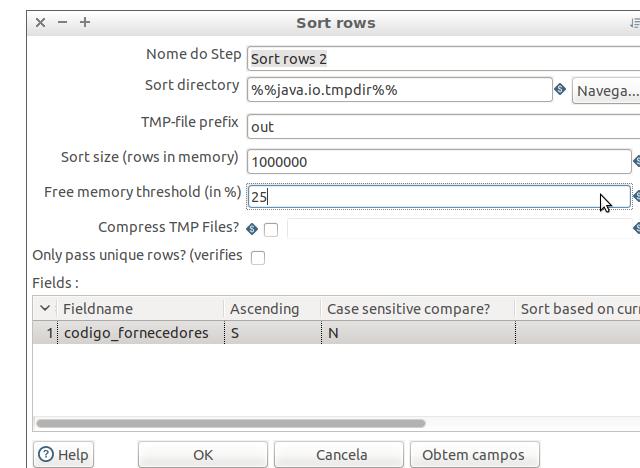
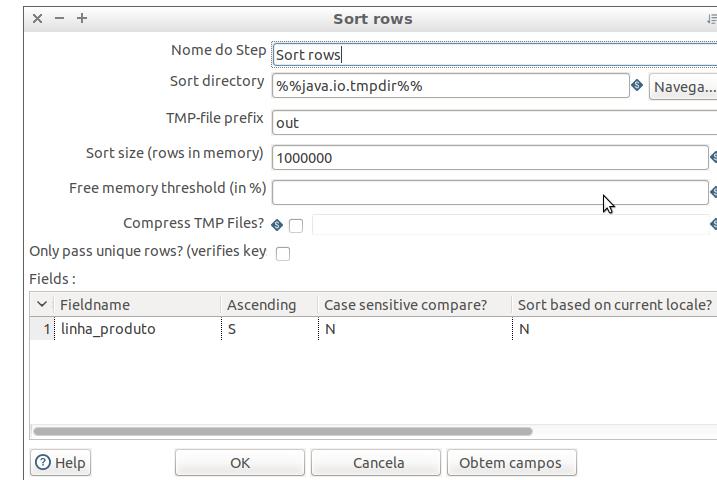
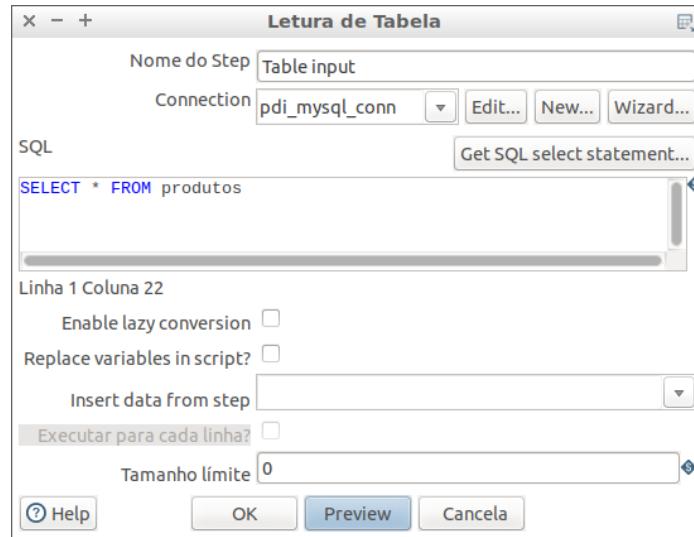
**Exercício:** Popular a tabela dimensão de produtos(**dim\_ produtos**).

**Objetivo:** Carga da Dimensão Produtos

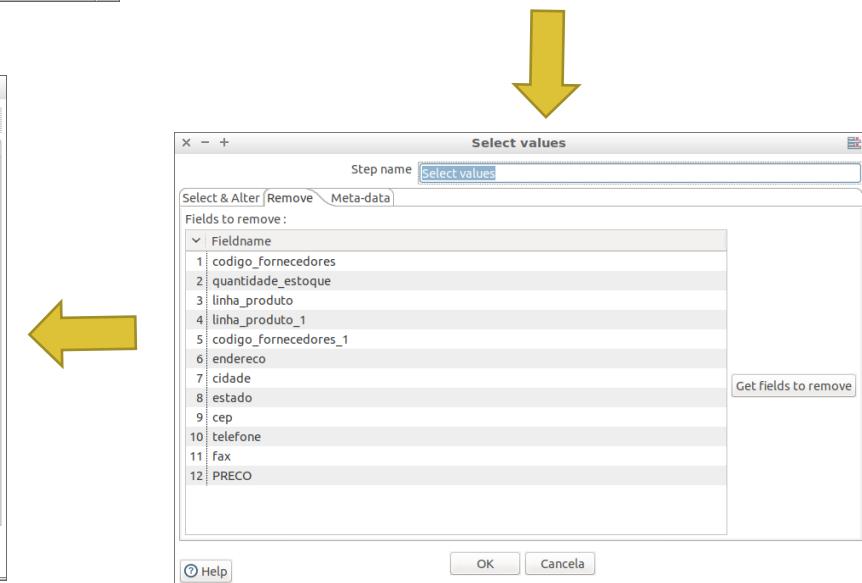
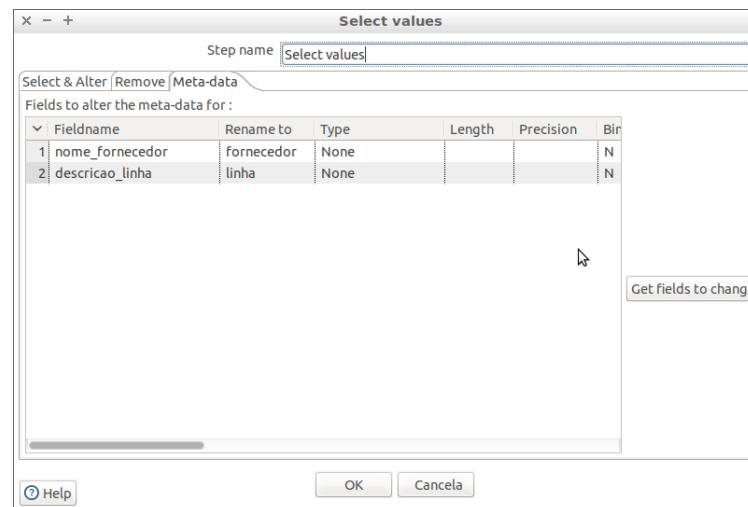
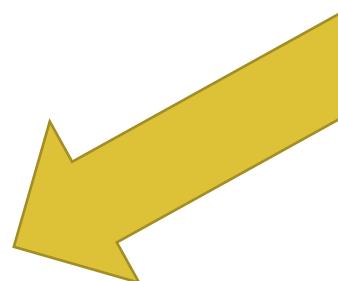
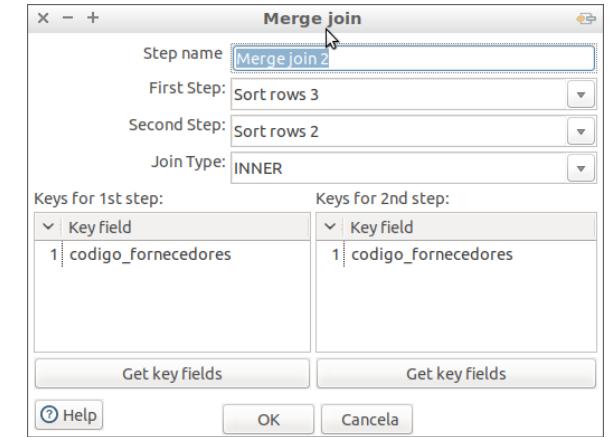
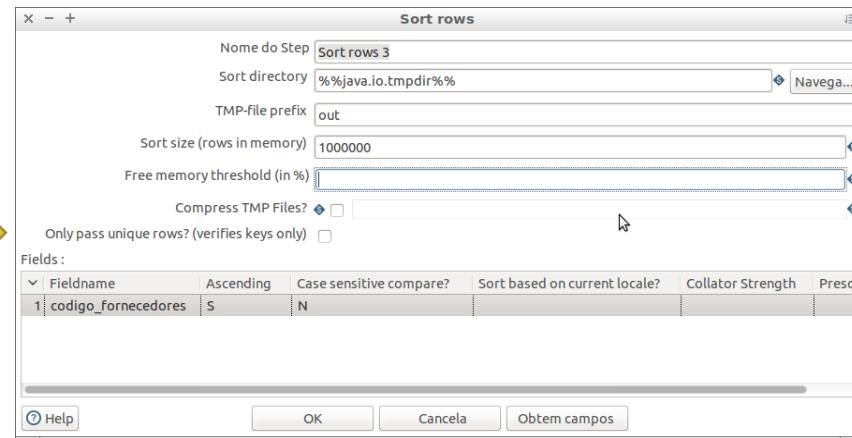
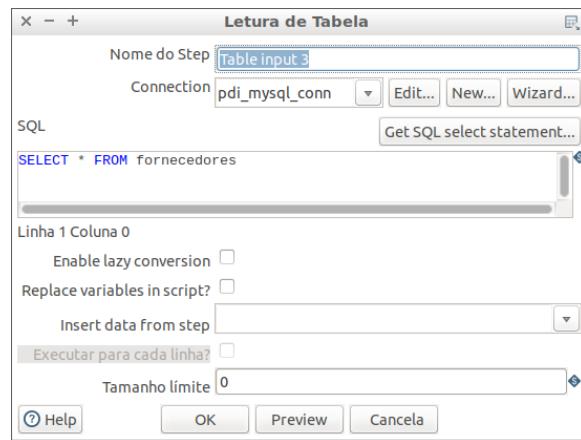
**Resultado:**



# Exercício 6 – Carga da tabela Produtos



# Exercício 6 – Carga da tabela Produtos



# Exercício 6 – Carga da tabela Produtos

**Obtém o valor da sequencia do banco de dados**

Nome do step: Add sequence  
Nome do valor: fk\_produto

Use a database to generate the sequence  
Usa BD para obter a sequencia?

Connection:     
Schema name:   
Nome da sequência: SEQ

Use a transformation counter to generate the sequence  
Usa o contador para calcular a sequencia?

Counter name (optional):  
Inicia no valor: 1  
Incremento de: 1  
Valor maximo: 999999999



**Insert / update**

Step name: Insert / update  
Connection: carga\_mysql\_conn     
Target schema:   
Target table: dim\_produto   
Commit size: 100  
Don't perform any updates:

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

Table field	Comparator	Stream field1	Stream field2	Get fields
1 codigo_produto	=		codigo_produto	<input type="button" value="Get fields"/>

Update fields:

Table field	Stream field	Update	Get update fields	Edit mapping
1 fornecedor	fornecedor	Y	<input type="button" value="Get update fields"/>	<input type="button" value="Edit mapping"/>
2 linha	linha	Y		
3 codigo_produto	codigo_produto	Y		
4 descricao	descricao	Y		
5 fk_produto	fk_produto	Y		

# Exercício 6 – Carga da tabela Produtos

x - + Examine preview data

Rows of step: dim\_produto (20 rows)

v	fk_produto	codigo_produto	descricao	linha	fornecedor
1	1	302	Fiat Linea	Passeio	FBUni
2	2	601	Chevrolet Corsa	Passeio	FBUni
3	3	201	Hyundai Tucson	Utilitários	FBUni
4	4	303	KIA Sportge	Utilitários	FBUni
5	5	402	Chevrolet Omega	Luxo	FBUni
6	6	509	Honda Civic	Luxo	FBUni
7	7	100	Ford Fiesta	Passeio	Apache
8	8	204	Chevrolet Vectra	Passeio	Apache
9	9	400	Volkswagen Jetta	Passeio	Apache
10	10	403	Volkswagen Gol	Passeio	Apache
11	11	304	Fiat Palio	Passeio	Pentaho
12	12	500	Ford Focus	Passeio	Pentaho
13	13	804	Jeep Gran Cherokee	Utilitários	Pentaho
14	14	102	Hyundai Azera	Luxo	Pentaho
15	15	301	Citroen C5	Luxo	Google
16	16	504	BMW X3	Luxo	OpenOffice.org
17	17	704	Volkswagen Passat	Luxo	OpenOffice.org
18	18	101	Toyota Hilux	Utilitários	Red Hat
19	19	401	Ford Edge	Utilitários	Red Hat
20	20	202	Toyota Corolla	Luxo	Red Hat

Fecha

# Exercício 7

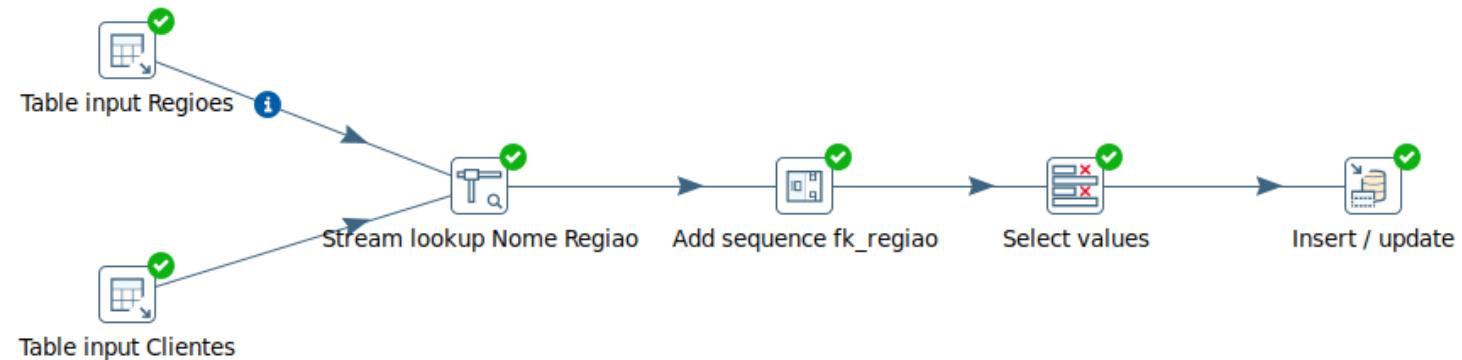
Carga da tabela dimensional Regiões

# Exercício 7 – Carga da tabela dimensional Regiões

**Exercício:** Popular a tabela dimensão de região (**dim\_regiao**).

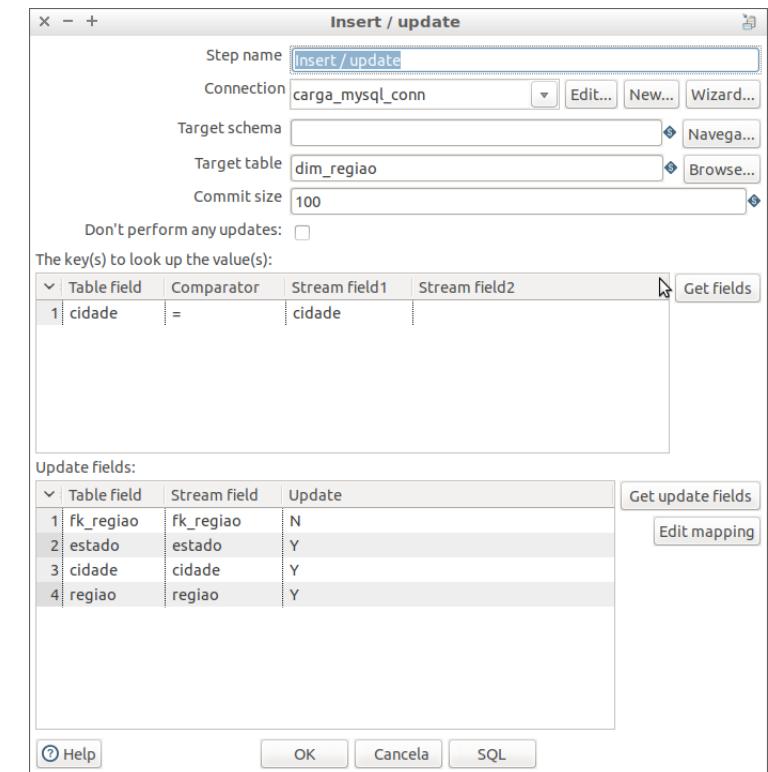
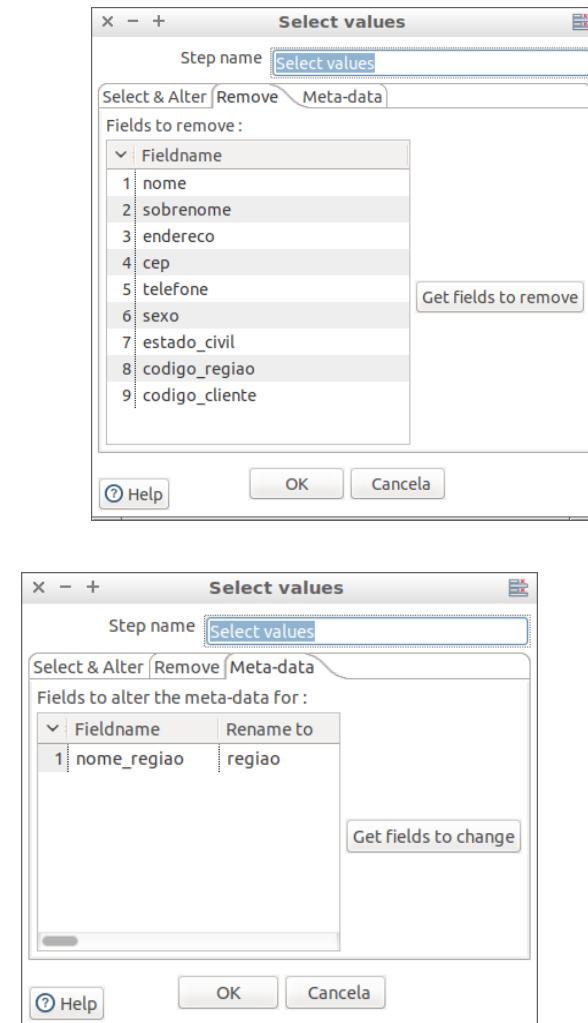
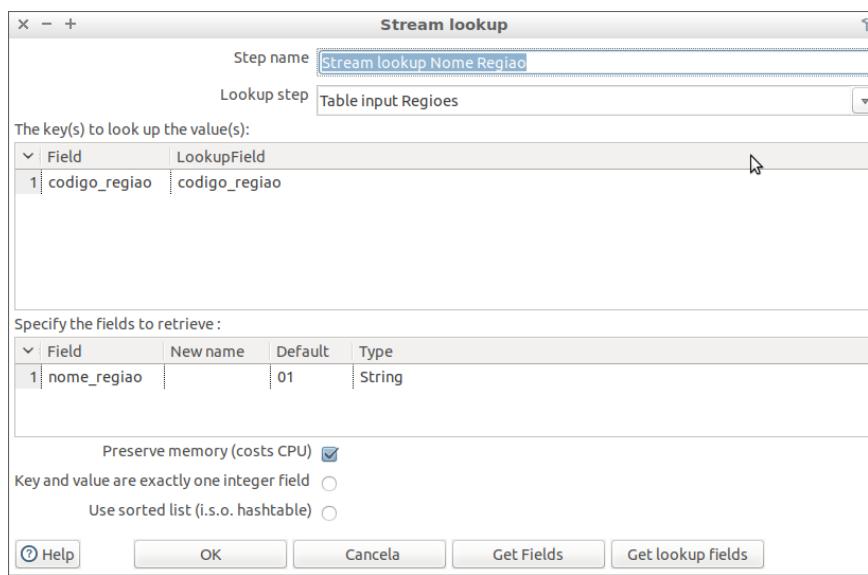
**Objetivo:** Carga da Dimensão Região

**Resultado:**



# Exercício 7 – Carga da tabela dimensional Regiões

## ❖ Algumas configurações:



# Exercício 8

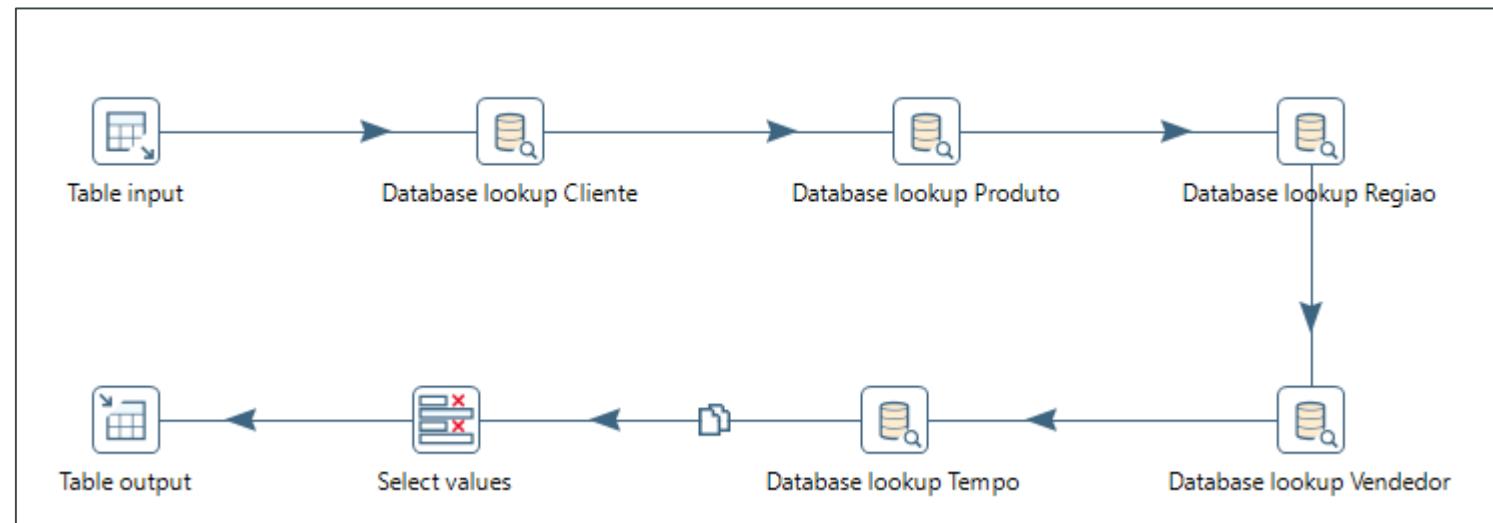
Carga da tabela de medida (Fato)

# Exercício 8 – Carga da tabela de medida (Fato)

**Exercício:** Popular a tabela fato com os dados atrelados as dimensões.

**Objetivo:** Carga da Tabela Fato

**Resultado:**

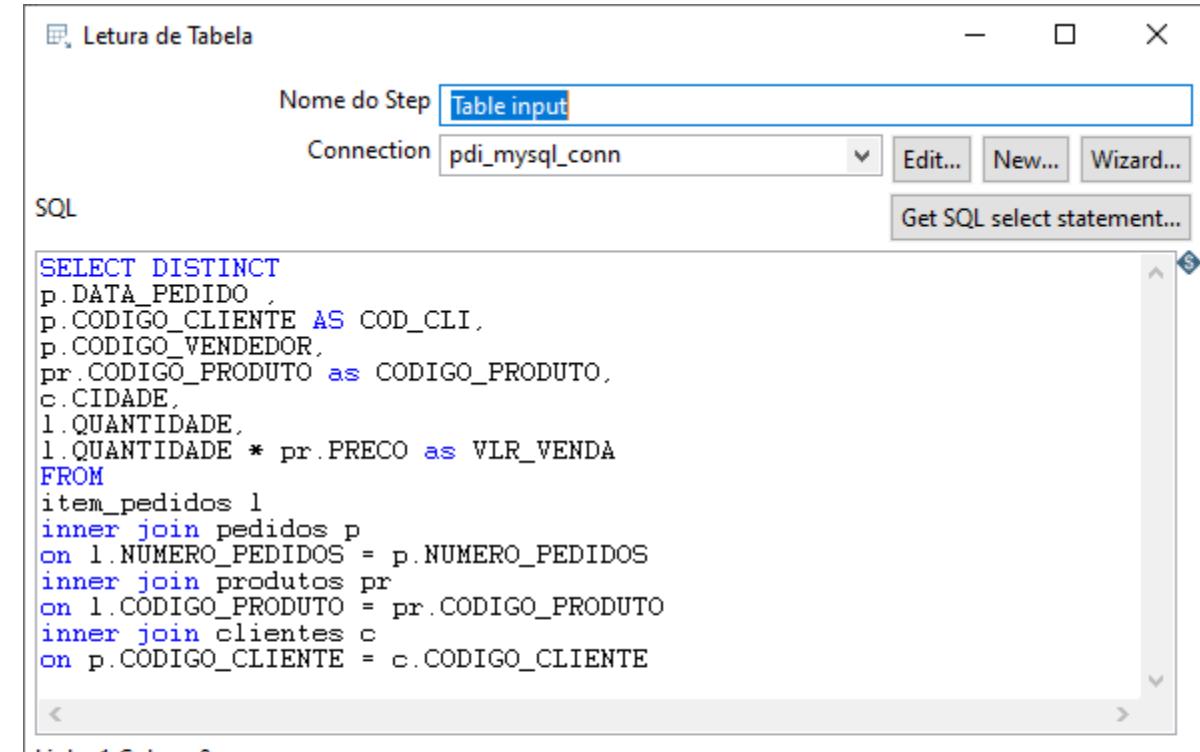


## Exercício 8 – Carga da tabela de medida (Fato)

- ❖ 1 - Clique em File / Novo / Transformação para criar uma nova transformação.
  - Adicione um *Step Table Input* e configure conforme abaixo:
  - Comando SQL par copiar:

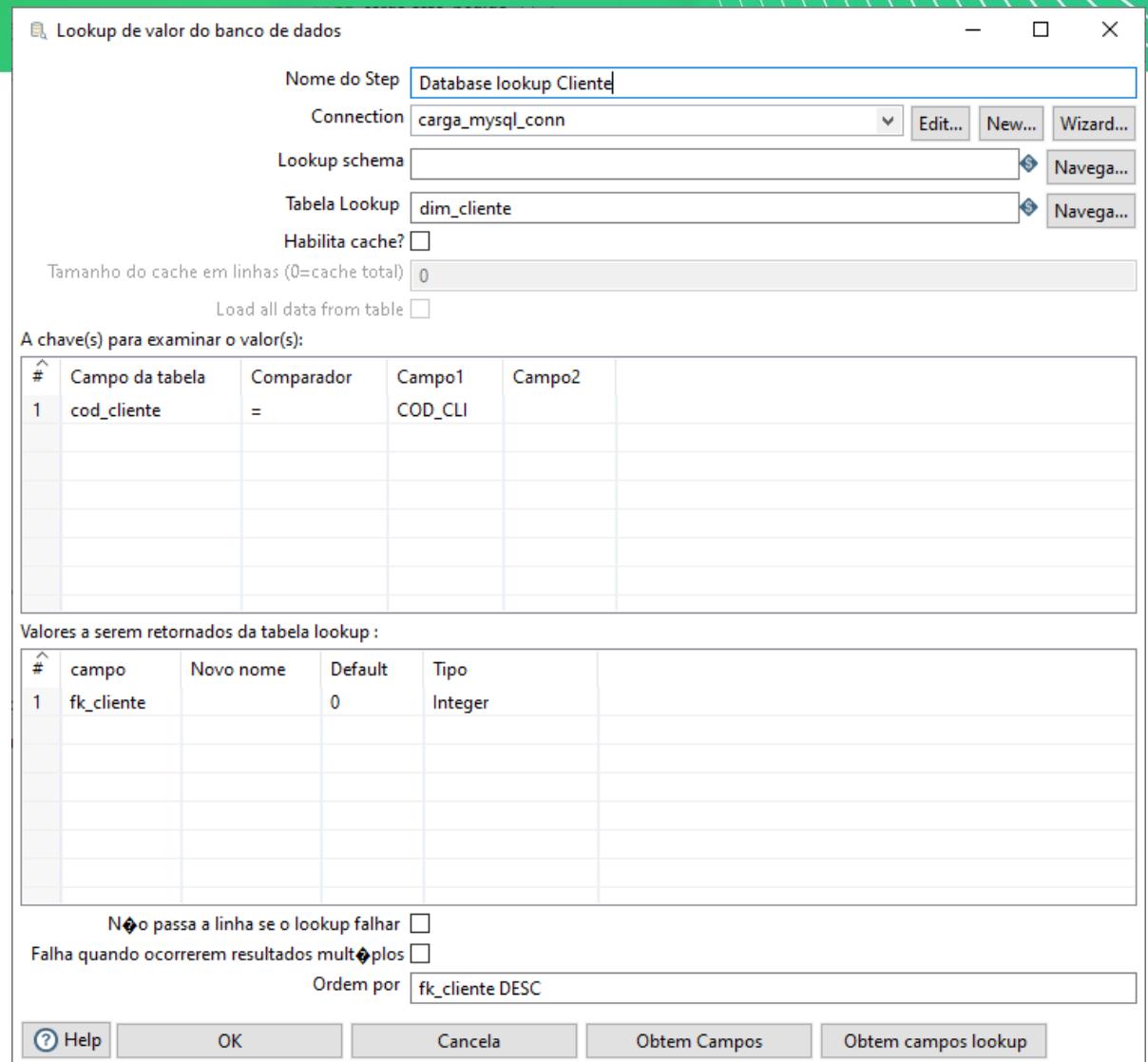
```

SELECT DISTINCT
    p.DATA_PEDIDO ,
    p.CODIGO_CLIENTE AS COD_CLI,
    p.CODIGO_VENDEDOR,
    pr.CODIGO_PRODUTO as CODIGO_PRODUTO,
    c.CIDADE,
    l.QUANTIDADE,
    l.QUANTIDADE * pr.PRECO as VLR_VENDA
FROM
    item_pedidos l
inner join pedidos p
    on l.NUMERO_PEDIDOS = p.NUMERO_PEDIDOS
inner join produtos pr
    on l.CODIGO_PRODUTO = pr.CODIGO_PRODUTO
inner join clientes c
on p.CODIGO_CLIENTE = c.CODIGO_CLIENTE
  
```



## Exercício 8 – Carga da tabela de medida (Fato)

- ❖ 2 - Adicione um *Step Database Lookup* e configure conforme ao lado:

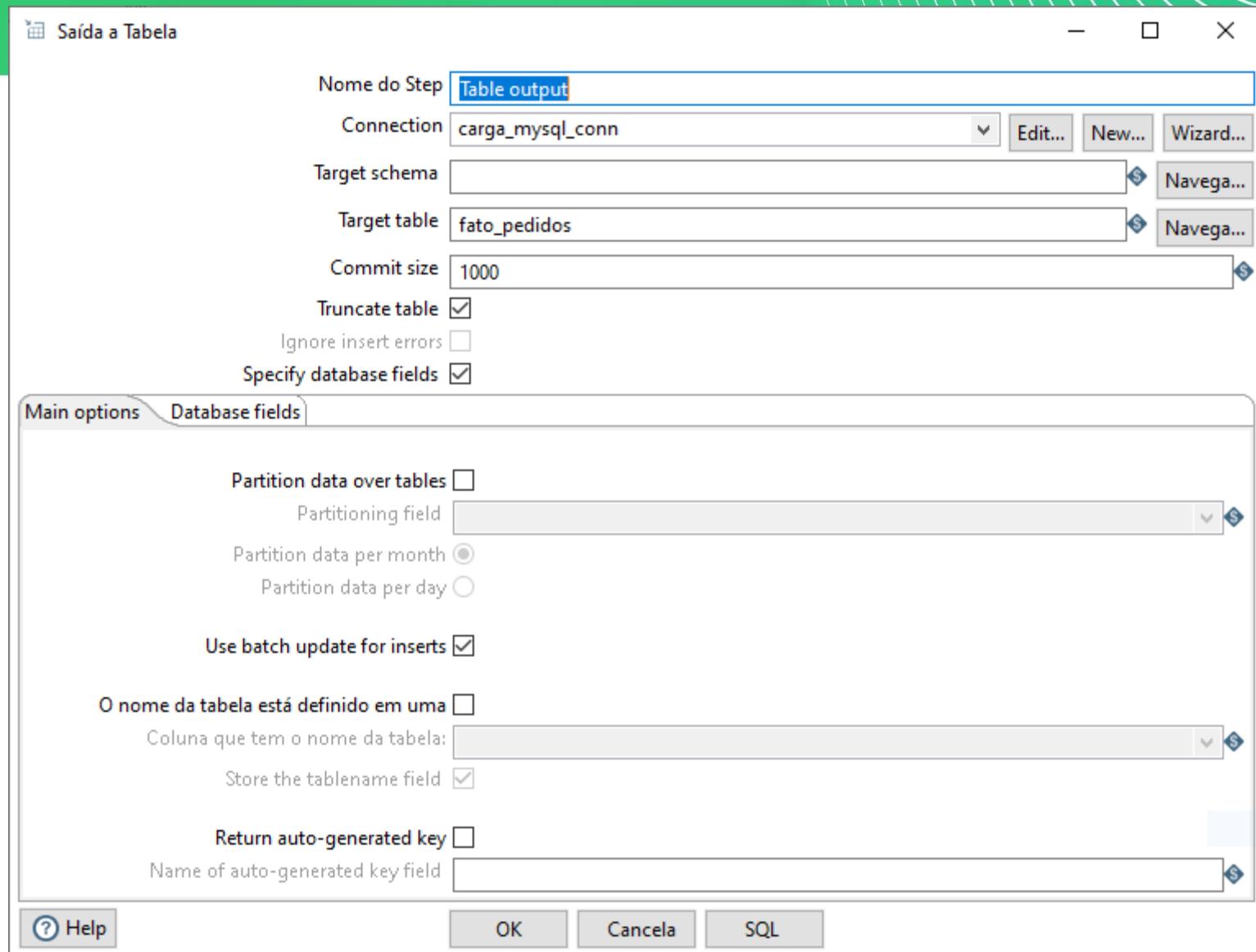


## Exercício 8 – Carga da tabela de medida (Fato)

- ❖ 3 - Siga a mesma lógica e adicione o *Step Database Lookup*, para todas as tabelas de dimensões do Datawarehouse;
  - **OBS:** No *Database Lookup* do **tempo** marque a opção: Não passa a linha se o lookup falhar
- ❖ 4 - Adicione um *Select Values*;

## Exercício 8 – Carga da tabela de medida (Fato)

- ❖ 5 - Por ultimo configure o step *Table Output* conforme imagem;
- ❖ 6 – Salve, execute e visualize o *preview*



## Exercício 8 – Carga da tabela de medida (Fato)

- ❖ 6 – Salve, execute e visualize o *preview*

#	fk_cliente	fk_produto	fk_regiao	fk_vendedor	fk_tempo	VLR_VENDA	QUANTIDADE
1	13	15	5	2	3347	380000,0	2
2	20	4	20	5	3503	83450,0	1
3	22	20	22	6	3474	156990,0	1
4	6	4	6	4	3595	83450,0	1
5	7	14	2	5	3600	356685,0	3
6	3	16	3	6	3631	129990,0	1
7	3	6	3	4	3319	140000,0	1
8	10	17	10	3	3428	119999,0	1
9	28	8	28	3	3389	223200,0	4
10	6	15	6	5	3651	190000,0	1
11	28	20	28	1	3299	313980,0	2
12	22	15	22	4	2228	570000,0	3
13	11	8	11	3	2410	111600,0	2
14	4	15	4	3	3480	190000,0	1
15	11	20	11	3	2410	313980,0	2
16	1	17	1	4	3358	119999,0	1
17	4	15	4	3	3480	190000,0	1

# Exercício 9

Criar o pipeline do ETL

# Exercício 9 – Criar o pipeline do ETL

**Exercício:** Criação de um Job com todo pipeline de atualização do ETL para DW.

**Objetivo:** Job com carga completa

**Resultado:**

