

**OPENINBIGDATA**

---

Treinamento Hitachi Vantara, Pentaho Data Integration

# Pentaho Data Integration

---

## Índice

1 – Criando e editando parâmetros de conexões .....	1
2 – Criando o repositório de metadados.....	3
3 – Criando e editando variáveis .....	7
4 – Criando um processo de carga na Staging 1.....	8
5 – Criando um processo de carga na Staging 2.....	16
6 – Criando o processo de carga da Dimensão Cliente.....	17
7 – Criando o processo de carga da dimensão Tempo .....	23
8 – Criando o processo de carga da dimensão Vendedor .....	24
9 – Criando o processo de carga da Fato Vendas .....	25
10 – Orquestrando a execução dos processos da Staging.....	34
11 – Orquestrando a execução dos processos Dimensões.....	39
12 – Orquestrando a execução do processo Fato .....	40
13 – Orquestrando todos os processos do DW .....	41
14 – Executando um Job através de linha de comando .....	47



## Exercícios do laboratório do módulo Pentaho Data Integration aka Kettle

*Os exercícios do laboratório do treinamento Pentaho Data Integration (aka Kettle) promovem o aprendizado para a criação dos processos de ETL que alimentarão o nosso modelo dimensional (Data Mart/ Data Warehouse).*

O conjunto de exercícios na sequência fixarão as principais técnicas na elaboração de processos de ETL utilizando a ferramenta Pentaho Data Integration.

**ATENÇÃO:** Para uma melhor visualização das imagens que servirão de instruções para os exercícios, **AUMENTE O ZOOM DO SEU PDF**

**ATENÇÃO:** Sempre após imagens com códigos escritos, haverá uma cópia do mesmo para você copiar

### 1 – Criando e editando parâmetros de conexões

**Objetivo:** O objetivo deste exercício é apresentar ao aluno a possibilidade de trabalhar com parâmetros de conexões externos a ferramenta, permitindo uma grande facilidade nas mudanças de configurações relacionadas aos ambientes de bancos de dados, como Desenvolvimento, Homologação e Produção.

Tempo médio para a construção do exercício: **10 minutos**

Complexidade para a construção do exercício: **baixa**

1. Edite o arquivo jdbc.properties que está localizado em `..\treinamento\design-tools\data-integration\simple-jndi`;
2. Adicione o seguinte bloco de texto abaixo, ao final do texto no arquivo:

```
des_star/type=javax.sql.DataSource  
des_star/driver=com.mysql.jdbc.Driver  
des_star/url=jdbc:mysql://localhost:3306/motor-inc-star  
des_star/user=root  
des_star/password=root
```

**ATENÇÃO:** Depois de colado o texto acima no arquivo jdbc.properties, confira se não há espaços a direita e também não deixe espaços entre as linhas do bloco de texto.

**DICA:** Acesse o blog [planetapentaho.com.br](http://planetapentaho.com.br), lá tem um post sobre como conectar outros bancos. ([openin.com.br/drivers-jdbc-pentaho-vantara](http://openin.com.br/drivers-jdbc-pentaho-vantara))

3. Sempre que editar este arquivo, se o spoon estiver aberto, será necessário reiniciá-lo para reconhecer as alterações; e
4. Salvar e fechar o arquivo.

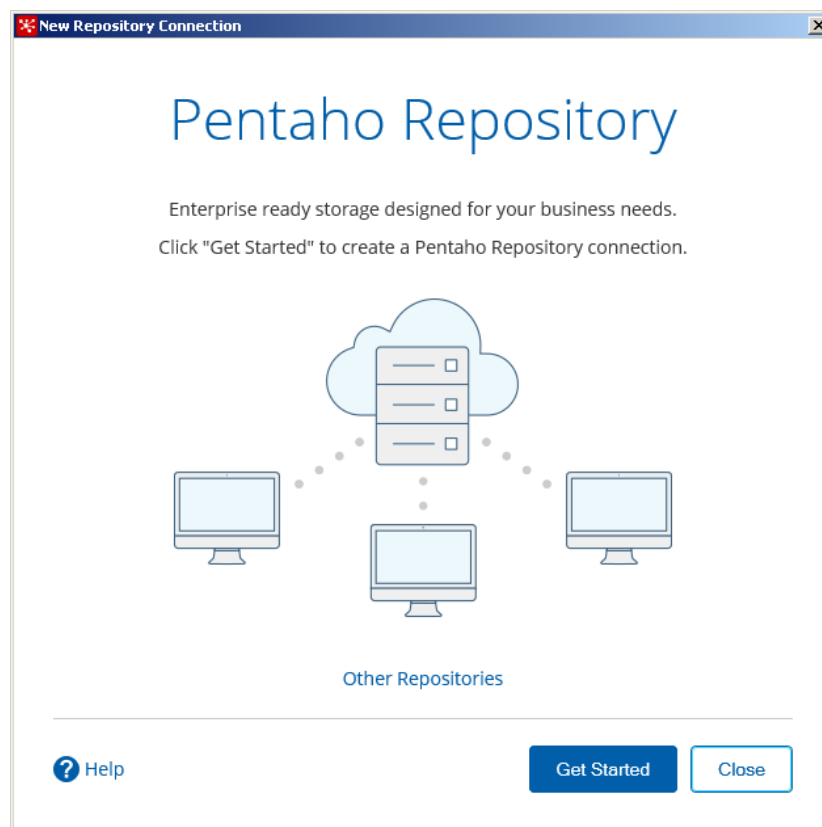
## 2 – Criando o repositório de metadados

**Objetivo:** O objetivo deste exercício é apresentar ao aluno como fazer para criar um repositório de metadados baseado em arquivos para ser utilizado pelas ferramentas da solução Pentaho Data Integration.

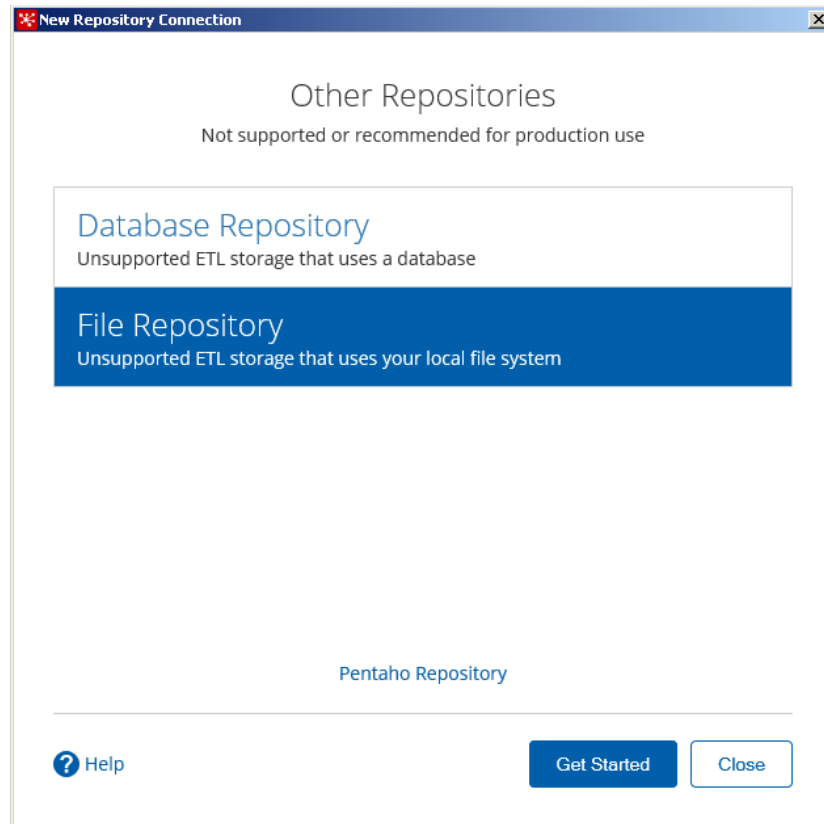
Tempo médio para a construção do exercício: **10 minutos**

Complexidade para a construção do exercício: **média**

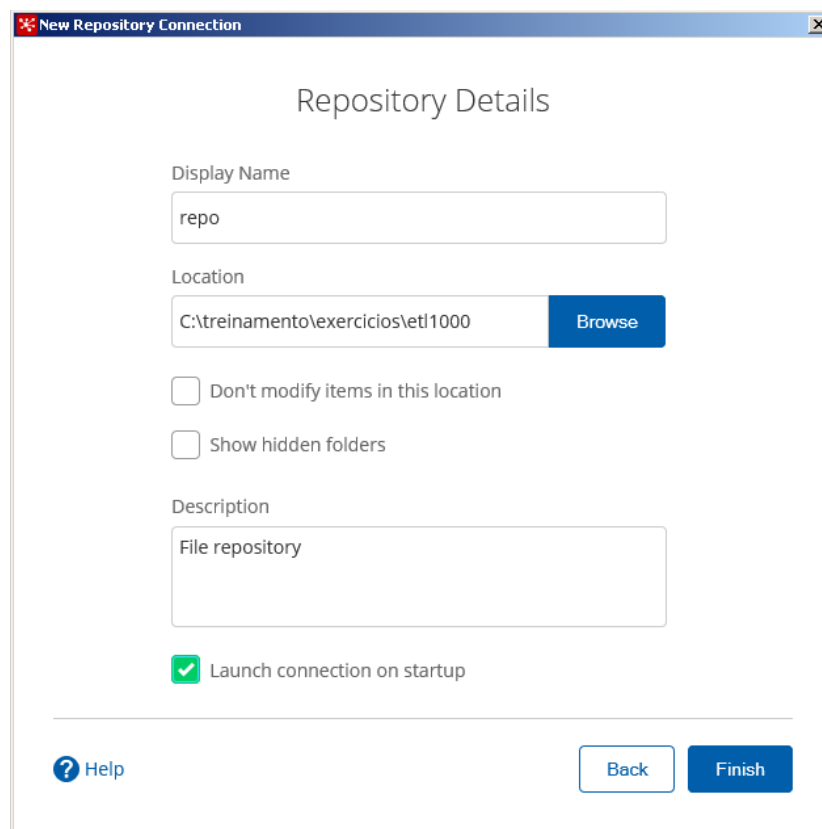
1. No diretório `..\treinamento\design-tools\data-integration` executar o arquivo `Spoon.bat`;
2. Logo após a tela de splash, no canto direito superior, clique no botão *Connect*;
3. Na janela *New Repository Connection*, clique no link *Other Repositories*;



4. Na próxima tela, selecione a opção File Repository para criar o seu repositório baseado em arquivos de sistema e clique em Get Started;



5. Configure conforme imagem abaixo e clique em *Finish*;



New Repository Connection

### Repository Details

Display Name  
repo

Location  
C:\treinamento\exercicios\etl1000 Browse

☐ Don't modify items in this location

☐ Show hidden folders

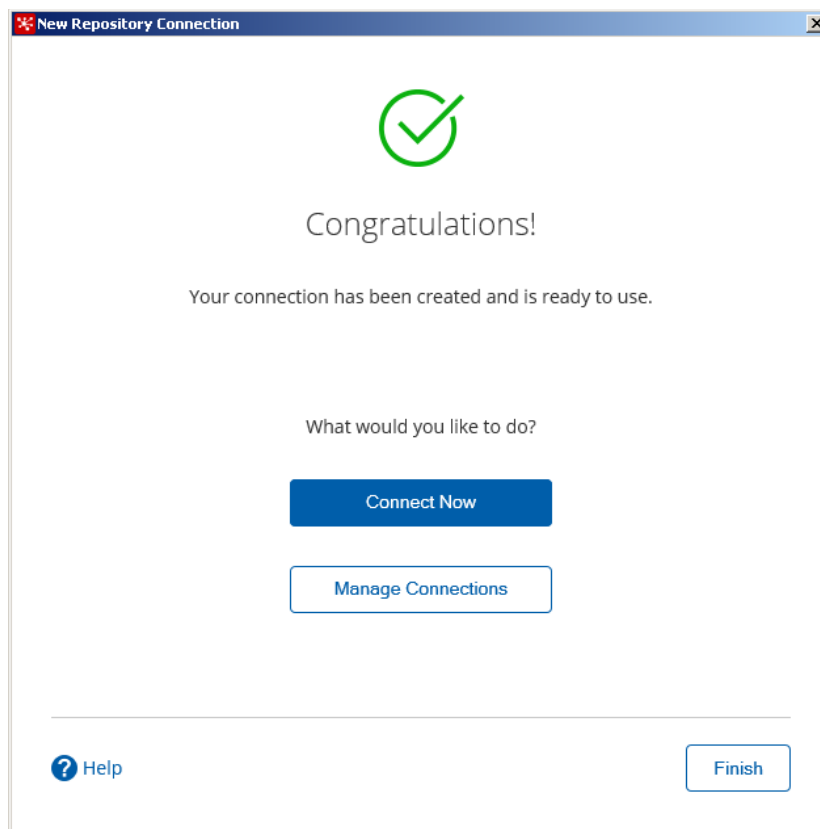
Description  
File repository

☒ Launch connection on startup

? Help Back Finish

**ATENÇÃO:** O valor da propriedade *Location* deve condizer com o local na sua estrutura de diretórios.

6. Para finalizar o processo, clique em *Finish*;



7. Feche o Spoon para fazer o exercício seguinte.



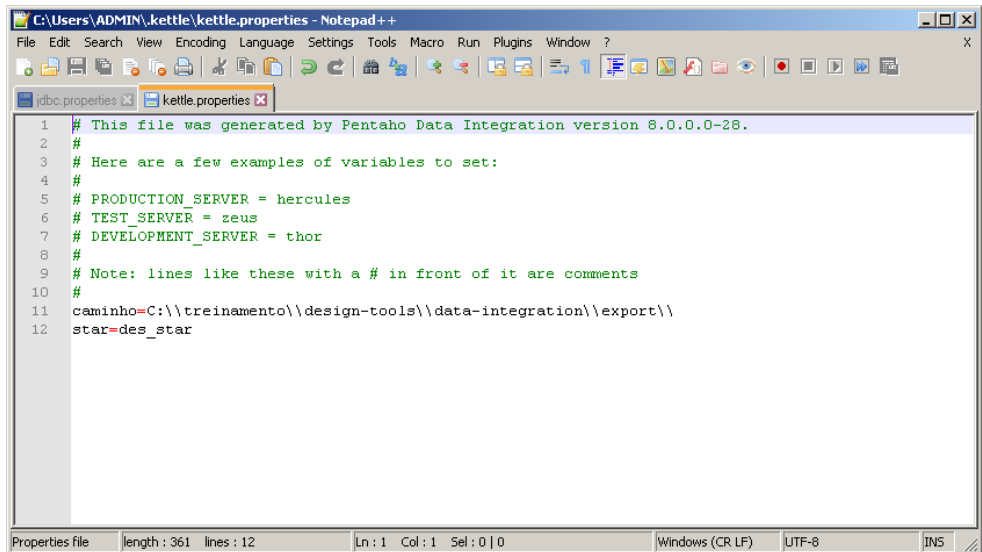
### 3 – Criando e editando variáveis

**Objetivo:** O objetivo deste exercício é apresentar ao aluno como fazer para utilizar o mecanismo de variáveis. Este recurso é muito importante, pois permite grande facilidade no uso de configurações como caminhos de diretórios em ambientes diferentes e nome de conexões em ambientes de desenvolvimento, homologação e produção.

Tempo médio para a construção do exercício: **10 minutos**

Complexidade para a construção do exercício: **baixa**

1. Edite o arquivo kettle.properties que está localizado no diretório .kettle dentro do diretório Profile do usuário conectado;
2. Crie as seguintes variáveis conforme imagem abaixo; e



```
1 # This file was generated by Pentaho Data Integration version 8.0.0.0-28.
2 #
3 # Here are a few examples of variables to set:
4 #
5 # PRODUCTION_SERVER = hercules
6 # TEST_SERVER = zeus
7 # DEVELOPMENT_SERVER = thor
8 #
9 # Note: lines like these with a # in front of it are comments
10 #
11 caminho=C:\\treinamento\\design-tools\\data-integration\\export\\
12 star=des_star
```

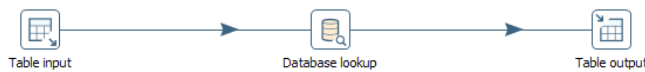
caminho=C:\\treinamento\\design-tools\\data-integration\\export\\  
star=des\_star

**ATENÇÃO:** Dentro do diretório data-integration, crie o diretório export  
**ATENÇÃO:** O conteúdo da variável caminho, deve condizer com o local na sua estrutura de diretórios e cuidado com os espaços em branco a direita de cada valor.

3. Salvar e fechar o arquivo.

## 4 – Criando um processo de carga na Staging 1

Exemplo de como será a *transformation*

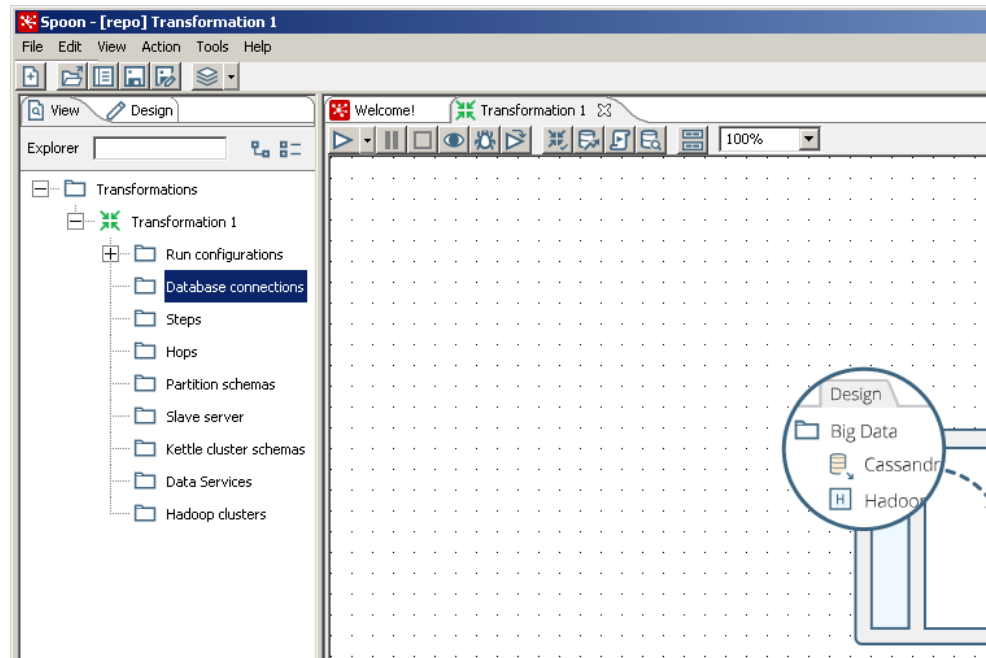


**Objetivo:** O objetivo desta *transformation* é apresentar ao aluno um caso de uso da área de estagiamento (*Staging area*). Nesta *transformation* iremos juntar informações de empregado com território, mas extraíndo esses dados em momentos distintos. Os dados carregados por esta *transformation* servirá de apoio ao próximo exercício.

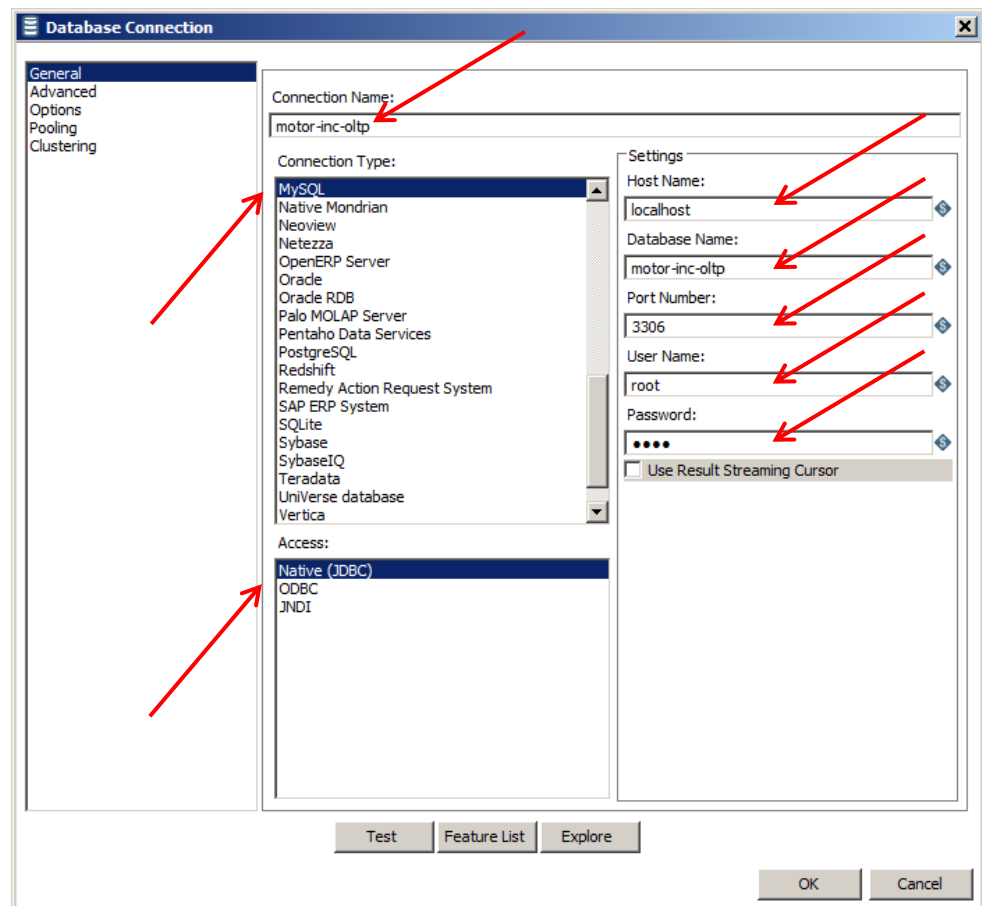
Tempo médio para a construção do exercício: **15 minutos**

Complexidade para a construção do exercício: **baixa-média**

1. Antes de tudo coloque o idioma da ferramenta em inglês, se não estiver;
2. Selecione o menu Tools e em seguida o menu Opções...;
3. Clique na aba Lock & Feel e em Preferred Language, selecione English (US);
4. Clique em OK e reinicialize o Spoon;
5. Na volta, crie uma nova *transformation*; (menu File > New > Transformation)
6. Crie uma nova conexão, selecionando *Database connections* na aba *View*, conforme imagem abaixo;



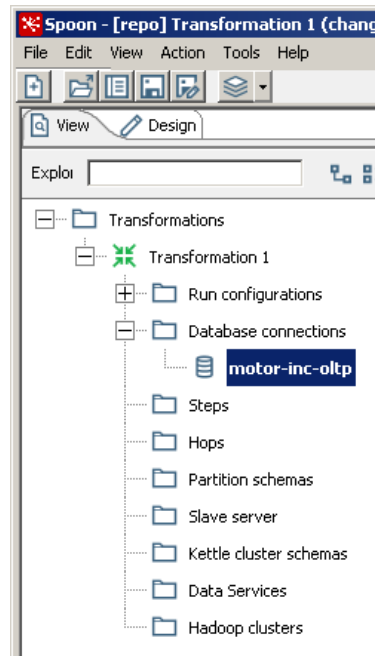
7. Clique com o botão direito do mouse em *Database connections* e clique em *New* e parametrize conforme imagem abaixo, lembrando que a senha do usuário root é root;



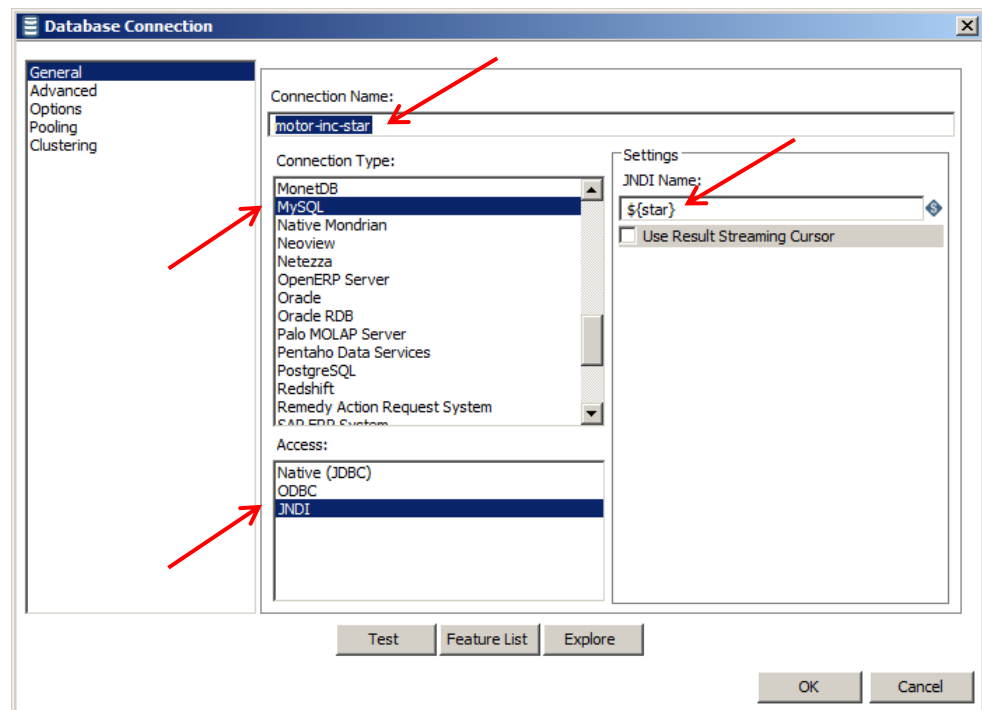
AJUDA:

<http://wiki.pentaho.com/display/BAD/Create+RDBMS+Connection>

8. Uma vez criado a conexão motor-inc-oltp, vamos compartilhá-la para uso futuro em outras *transformations*. Para isso clique com o botão direito do mouse sob a conexão motor-inc-oltp e selecione *Share*. Verifique que o nome da conexão ficará em negrito, conforme imagem abaixo;



9. Agora, vamos criar uma nova conexão, mas ao invés de usarmos um acesso nativo JDBC, vamos usar um acesso JNDI e baseado na variável **des\_star** criada no exercício anterior;
10. Crie uma nova conexão, selecionando New em *Database connections* na aba *View*, conforme imagem abaixo;

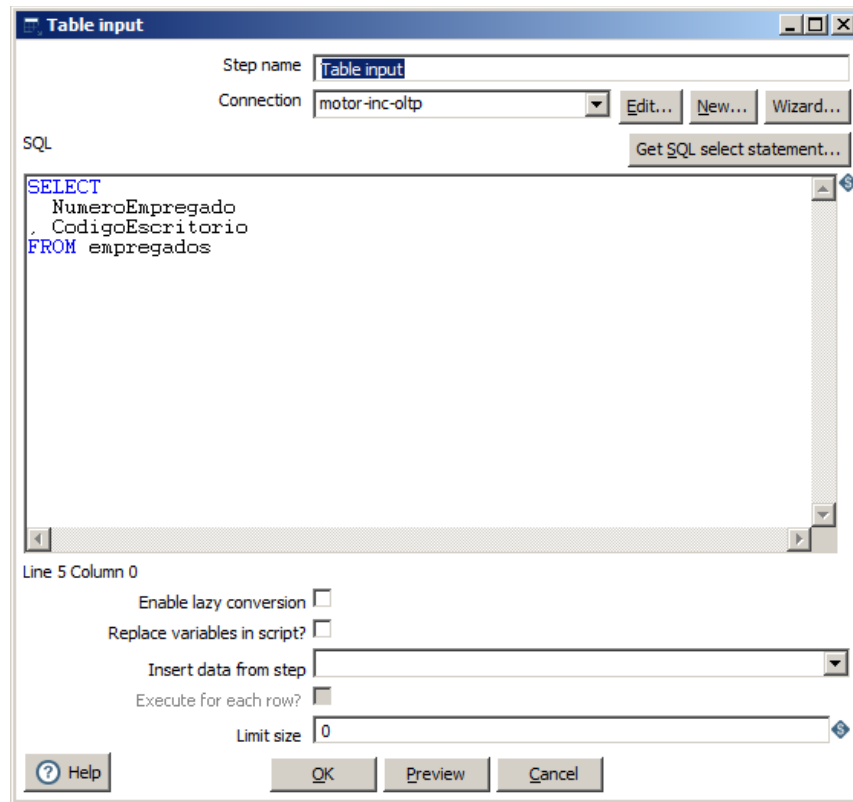


11. Faça também o compartilhamento da conexão motor-inc-star;
12. Agora, vamos iniciar o desenvolvimento do fluxo de dados, arraste para a área de trabalho o seguinte step:

a. *Table input* (localizado na categoria *Input* na aba *Design*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Table+Input>

Parametrize conforme imagem abaixo.



13. Query abaixo;

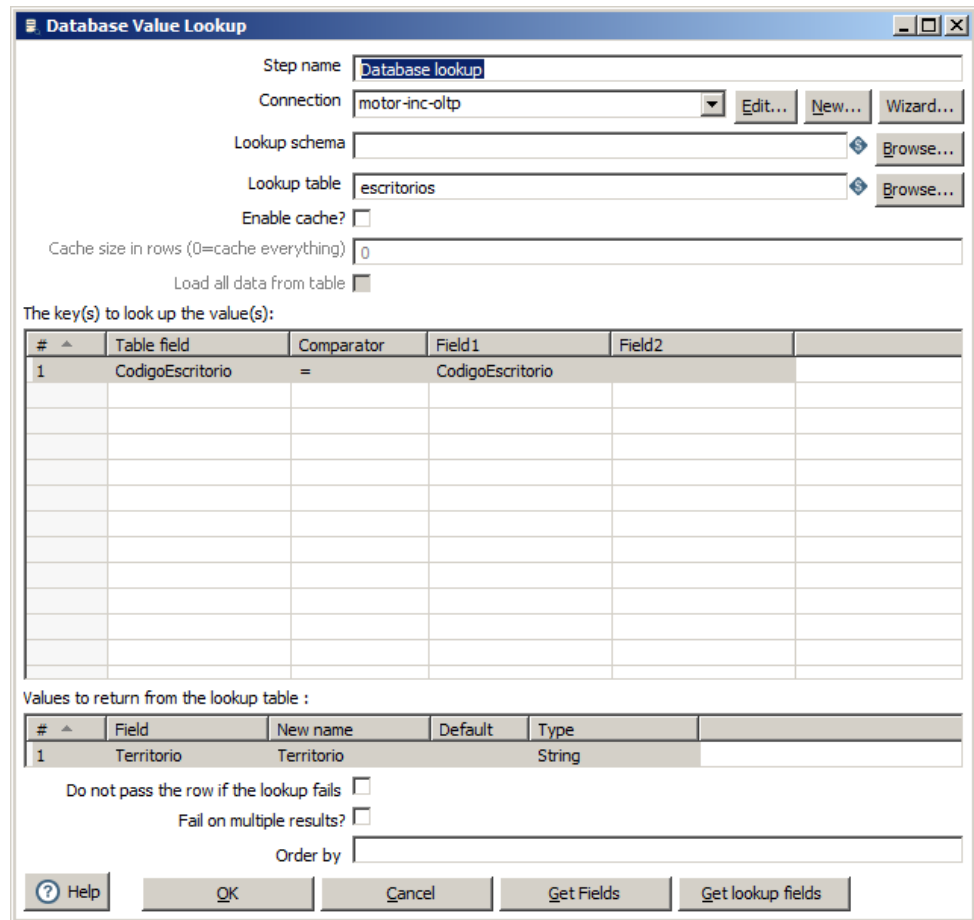
```
SELECT
    NumeroEmpregado
, CodigoEscritorio
FROM empregados
```

14. Na sequencia do fluxo de dados apresentado no início deste exercício, utilize o seguinte step:

b. *Database lookup* (Localizado na categoria *Lookup* na aba *Design*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Database+lookup>

Parametrize conforme imagem abaixo.



Step name: Database lookup

Connection: motor-inc-oltp

Lookup schema:

Lookup table: escritorios

Enable cache? ☐

Cache size in rows (0=cache everything): 0

Load all data from table ☐

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

#	Table field	Comparator	Field1	Field2	
1	CodigoEscritorio	=	CodigoEscritorio		

Values to return from the lookup table :

#	Field	New name	Default	Type	
1	Territorio	Territorio		String	

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

? Help OK Cancel Get Fields Get lookup fields

DICA: Não digite os nomes dos campos, clique nos botões de Get Fields e Get lookup fields. Se a hop estiver estabelecida entre os steps, os Get carregam para você os nomes dos campos.

15. Na sequencia do fluxo de dados apresentado no início deste exercício, utilize o seguinte step:

c. *Table output* (Localizado na categoria *Output* na aba *Design*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Table+Output>



Parametrize conforme imagem abaixo.

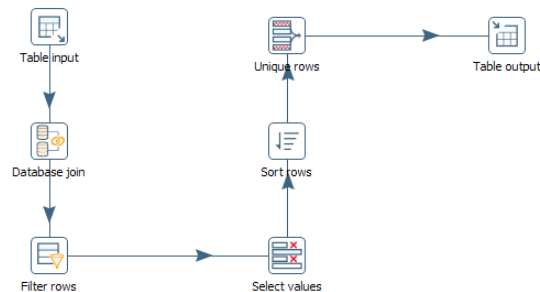
**ATENÇÃO:** Quando for criar a ligação (*Hop*) entre os *steps Database lookup* e *Table output* e aparecer um *popup* com as opções *Main output of step* e *Error handling of step*, sempre selecionem a primeira opção.

Faça isso até o instrutor explicar o motivo no exercício 9.

16. Salve a *transformation* como **4\_carga\_stg\_territorios**, na raiz do repositório; e
17. Execute a *transformation* (menu Action > Run), clique em Run e observe os dados carregados no banco de dados através da ferramenta HeidiSQL, fazendo um select na tabela carregada.

## 5 – Criando um processo de carga na Staging 2

Exemplo de como será a *transformation*



**Objetivo:** O objetivo desta *transformation* é apresentar ao aluno um outro caso de uso da área de estagiamento (*Staging area*). Nesta *transformation* iremos juntar informações de clientes como País e complementar com informações vindas do vendedor, como Território, pois a tabela de clientes não possui a informação de Território. O resultado desta *transformation* será utilizado para compor com os outros dados de clientes no processo de carga da dimensão cliente.

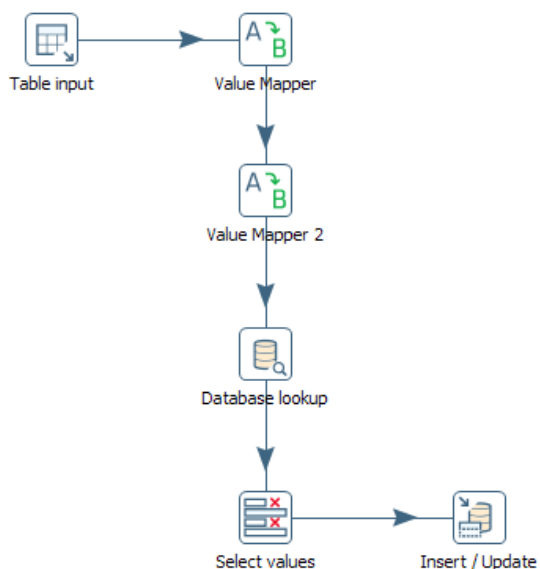
Tempo médio para a construção do exercício: **35 minutos**

Complexidade para a construção do exercício: **média**

1. Este exercício você recebeu por email e está pronto para usar; e
2. Importe a *transformation* **5\_carga\_stg\_clientes\_territorios** e acompanhe a explicação dada pelo instrutor.

## 6 – Criando o processo de carga da Dimensão Cliente

Exemplo de como será a *transformation*



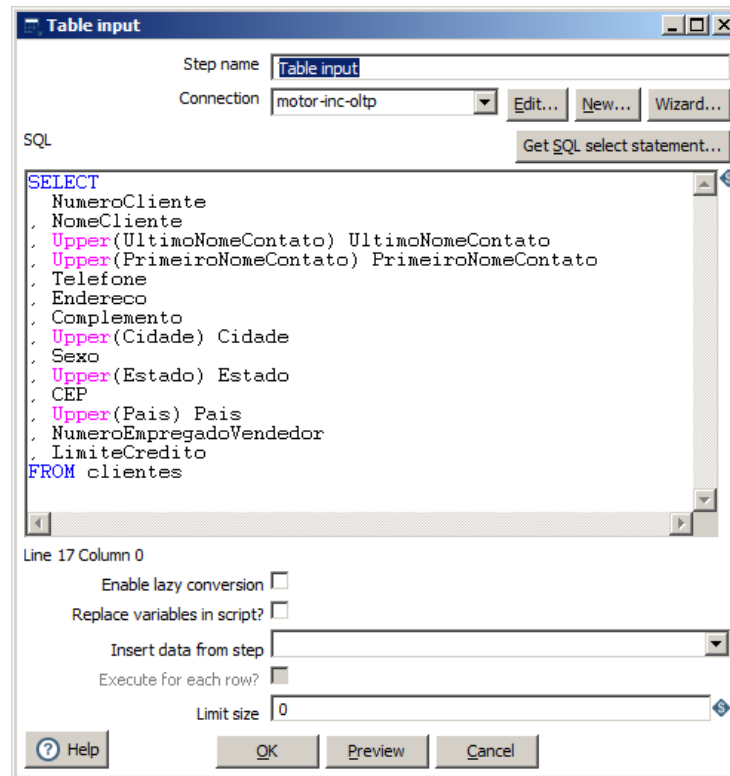
**Objetivo:** O objetivo desta *transformation* é apresentar ao aluno como utilizar uma informação gerada em processo anterior e que foi armazenada na área de estagiamento. Nesta *transformation* iremos complementar as informações de clientes com o território, pois grande parte dos clientes não possuem a informação de território em sua tabela origem.

Tempo médio para a construção do exercício: **30 minutos**

Complexidade para a construção do exercício: **média**

1. Crie uma nova *transformation*;
2. Arraste para a área de trabalho os seguintes steps:
  - a. *Table input* (Localizado na categoria *Input*)
  - b. *Value Mapper* (Localizado na categoria *Transform*)  
AJUDA: <http://wiki.pentaho.com/display/EAI/Unique+Rows>
  - c. *Value Mapper* (Localizado na categoria *Transform*)

Parametrize conforme imagem abaixo.



### 3. Query abaixo;

```
SELECT
    NumeroCliente
,   NomeCliente
,   Upper(UltimoNomeContato) UltimoNomeContato
,   Upper(PrimeiroNomeContato) PrimeiroNomeContato
,   Telefone
,   Endereco
,   Complemento
,   Upper(Cidade) Cidade
,   Sexo
,   Upper(Estado) Estado
,   CEP
,   Upper(Pais) Pais
,   NumeroEmpregadoVendedor
,   LimiteCredito
FROM clientes
```

**Value Mapper**

Step name : Value Mapper

Fieldname to use : Sexo

Target field name (empty=overwrite) :

Default upon non-matching :

Field values:

#	Source value	Target value
1	F	FEMININO
2	M	MASCULINO

Help OK Cancel

**Value Mapper**

Step name : Value Mapper 2

Fieldname to use : Pais

Target field name (empty=overwrite) :

Default upon non-matching :

Field values:

#	Source value	Target value
1	ESTADOS UNIDOS	EUA

Help OK Cancel

ATENÇÃO: O texto a ser buscado deve ser em maiúsculo, pois o conteúdo do campo País foi colocado em maiúsculo. Além disso estamos num ambiente Java e java é case sensitive, ou seja, uma letra maiúscula é diferente da mesma letra minúscula.

4. Na sequencia do lay-out apresentado no início do exercício utilize os seguintes steps:
  - a. *Database lookup* (Localizado na categoria *Lookup*)
  - b. *Select Values* (Localizado na categoria *Transform*)

Parametrize conforme imagens abaixo.

**Database Value Lookup**

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache everything):

Load all data from table ☐

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

#	Table field	Comparator	Field 1	Field 2
1	Pais	=	Pais	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	Territorio			String

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

**Select / Rename values**

Step name:

Select & Alter  Remove  Meta-data

Fields to remove :

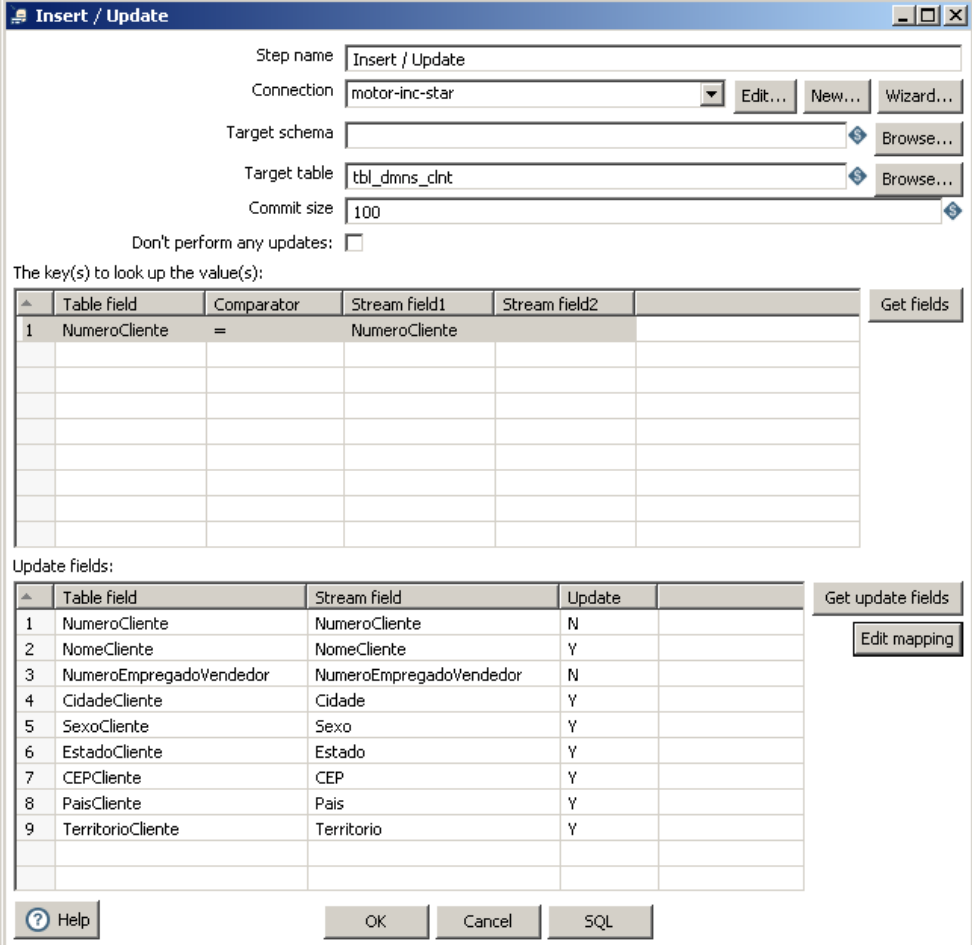
#	Fieldname
1	UltimoNomeContato
2	PrimeiroNomeContato
3	Telefone
4	Endereco
5	Complemento
6	LimiteCredito

5. Na sequência do lay-out apresentado no início do exercício utilize o seguinte step:

a. *Insert / Update* (Localizado na categoria *Output*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Insert+-+Update>

Parametrize conforme imagem abaixo.



Step name: Insert / Update

Connection: motor-inc-star

Target schema:

Target table: tbl\_dmns\_clnt

Commit size: 100

Don't perform any updates: ☐

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

	Table field	Comparator	Stream field1	Stream field2
1	NumeroCliente	=	NumeroCliente	

Update fields:

	Table field	Stream field	Update
1	NumeroCliente	NumeroCliente	N
2	NomeCliente	NomeCliente	Y
3	NumeroEmpregadoVendedor	NumeroEmpregadoVendedor	N
4	CidadeCliente	Cidade	Y
5	SexoCliente	Sexo	Y
6	EstadoCliente	Estado	Y
7	CEPCliente	CEP	Y
8	PaisCliente	Pais	Y
9	TerritorioCliente	Territorio	Y

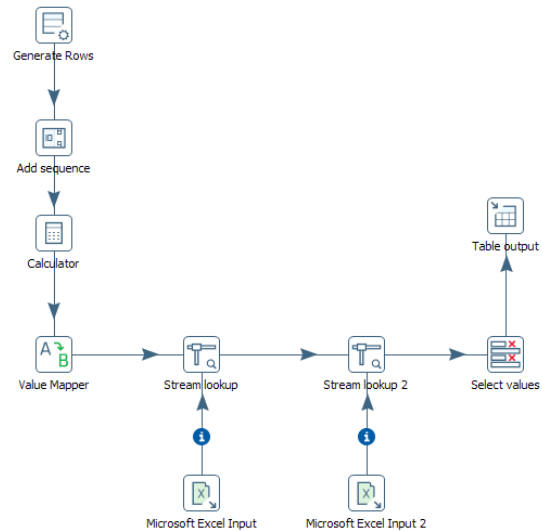
6. Salve a *transformation* como **6\_carga\_tbl\_dmns\_clnt**;
7. Execute a *transformation* (menu Action > Run), clique em Run e observe os dados carregados no banco de dados através da ferramenta HeidiSQL;

8. Como sugestão, altere o território de algum cliente na origem e rode a *transformation* novamente e observando a alteração efetuada; e
9. Fechar a *transformation*.



## 7 – Criando o processo de carga da dimensão Tempo

Exemplo de como será a *transformation*

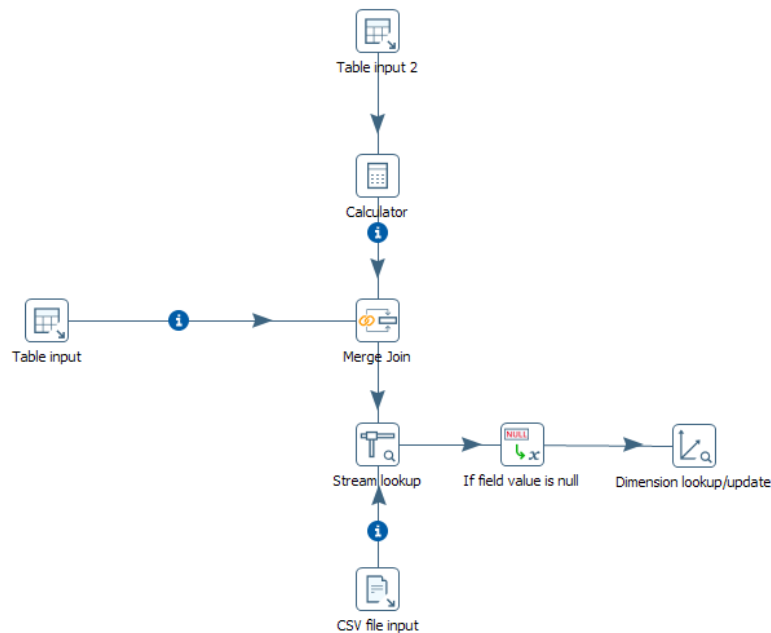


**Objetivo:** O objetivo desta *transformation* é apresentar ao aluno a construção de um processo que irá popular a dimensão tempo. Nesta *transformation* utilizaremos informações manuais e inseridas nos próprios *steps*, dando uma excelente ideia do potencial da ferramenta em termos de inserir informações não sistematizadas.

3. Este exercício você recebeu por email e está pronto para usar; e
4. Importe a *transformation* 7\_carga\_tbl\_dmns\_tmp e acompanhe a explicação dada pelo instrutor.

## 8 – Criando o processo de carga da dimensão Vendedor

Exemplo de como será a *transformation*



**Objetivo:** O objetivo desta transformação é apresentar ao aluno a possibilidade em se desnormalizar uma estrutura de dados e ainda obter informações de uma fonte em arquivo texto. Nesta *transformation* estamos desnormalizando a estrutura comercial da motor-inc, a fim de, criar uma estrutura dimensional dinâmica, permitindo a inserção de novos vendedores de forma automática. Aqui teremos também o controle de versionamento, SCD do tipo 2.

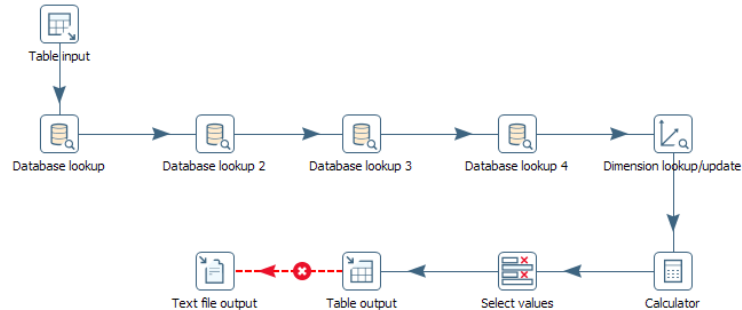
1. Este exercício você recebeu por email e está pronto para usar; e
2. Abra a *transformation* 8\_carga\_tbl\_dmns\_vndr e acompanhe a explicação dada pelo instrutor.

Bug conhecido

<http://jira.pentaho.com/browse/PDI-2292>

## 9 – Criando o processo de carga da Fato Vendas

Exemplo de como será a *transformation*



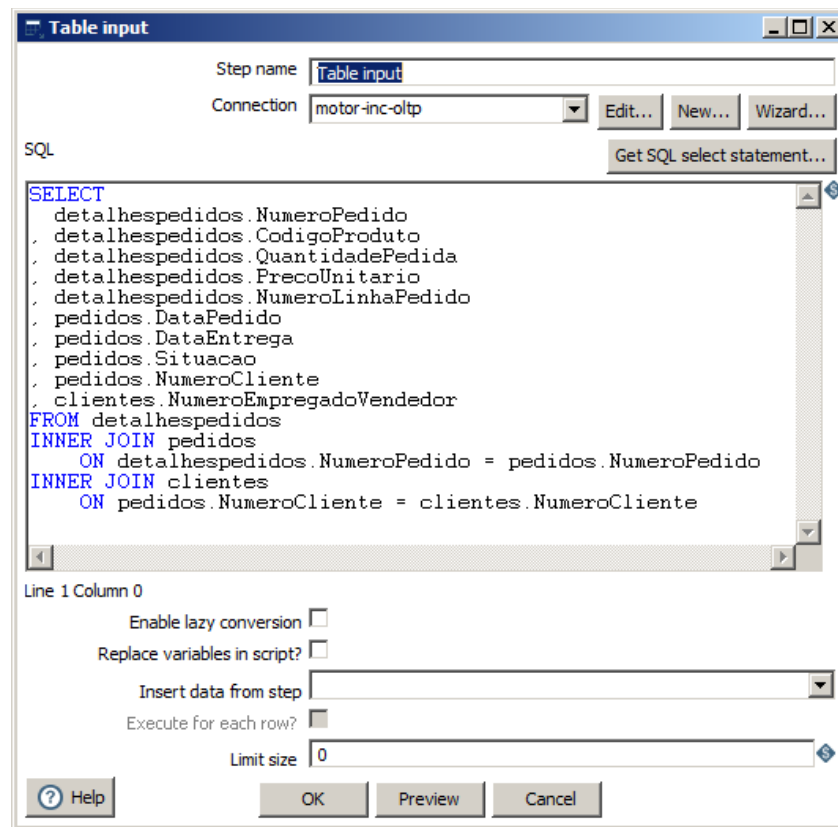
**Objetivo:** O objetivo desta *transformation* é apresentar ao aluno a criação do processo de carga da tabela fato Vendas. Nesta *transformation*, além dos dados extraídos da origem, existe um processo de troca (lookup) de dados entre as dimensões. A carga de dados é sempre inserção e caso haja algum erro de violação de constraint o registro será expurgado para um arquivo texto. Neste momento iremos também gerar uma informação nova, um procedimento de enriquecimento de dados.

Tempo médio para a construção do exercício: **40 minutos**

Complexidade para a construção do exercício: **média-alta**

1. Crie uma nova *transformation*;
2. Arraste para a área de trabalho o seguinte step:
  - a. *Table input* (Localizado na categoria *Input*)

Parametrize conforme imagem abaixo.



3. Query abaixo;

```

SELECT
    detalhespedidos.NumeroPedido
,   detalhespedidos.CodigoProduto
,   detalhespedidos.QuantidadePedida
,   detalhespedidos.PrecoUnitario
,   detalhespedidos.NumeroLinhaPedido
,   pedidos.DataPedido
,   pedidos.DataEntrega
,   pedidos.Situacao
,   pedidos.NumeroCliente
,   clientes.NumeroEmpregadoVendedor
FROM detalhespedidos
INNER JOIN pedidos
    ON detalhespedidos.NumeroPedido = pedidos.NumeroPedido
INNER JOIN clientes
    ON pedidos.NumeroCliente = clientes.NumeroCliente
  
```

4. Na sequencia do lay-out apresentado no início do exercício, utilize os seguintes steps:

- Database Lookup* (Localizado na categoria *Lookup*)
- Database Lookup* (Localizado na categoria *Lookup*)
- Database Lookup* (Localizado na categoria *Lookup*)
- Database Lookup* (Localizado na categoria *Lookup*)
- Dimension lookup/update* (Localizado na categoria Data Warehouse)

AJUDA: <http://wiki.pentaho.com/display/EAI/Dimension+Lookup+Update>

Parametrize conforme imagens abaixo.

Database Value Lookup

Step name: Database lookup

Connection: motor-inc-star

Lookup schema:

Lookup table: tbl\_dmns\_dnt

Enable cache? ☐

Cache size in rows (0=cache everything): 0

Load all data from table ☒

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

#	Table field	Comparator	Field1	Field2
1	NumeroCliente	=	NumeroCliente	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	ClienteId		-1	Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

**Database Value Lookup**

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache everything):

Load all data from table ☐

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

#	Table field	Comparator	Field1	Field2
1	CodigoProduto	=	CodigoProduto	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	ProdutoId		-1	Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

**Database Value Lookup**

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache everything):

Load all data from table ☐

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

#	Table field	Comparator	Field1	Field2
1	DataTempo	=	DataPedido	

Values to return from the lookup table :

#	Field	New name	Default	Type
1	TempoId	TempoIdPedido	-1	Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:

**Database Value Lookup**

Step name:

Connection:

Lookup schema:

Lookup table:

Enable cache? ☐

Cache size in rows (0=cache everything):

Load all data from table ☐

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

#	Table field	Comparator	Field1	Field2
1	DataTempo	=	DataEntrega	

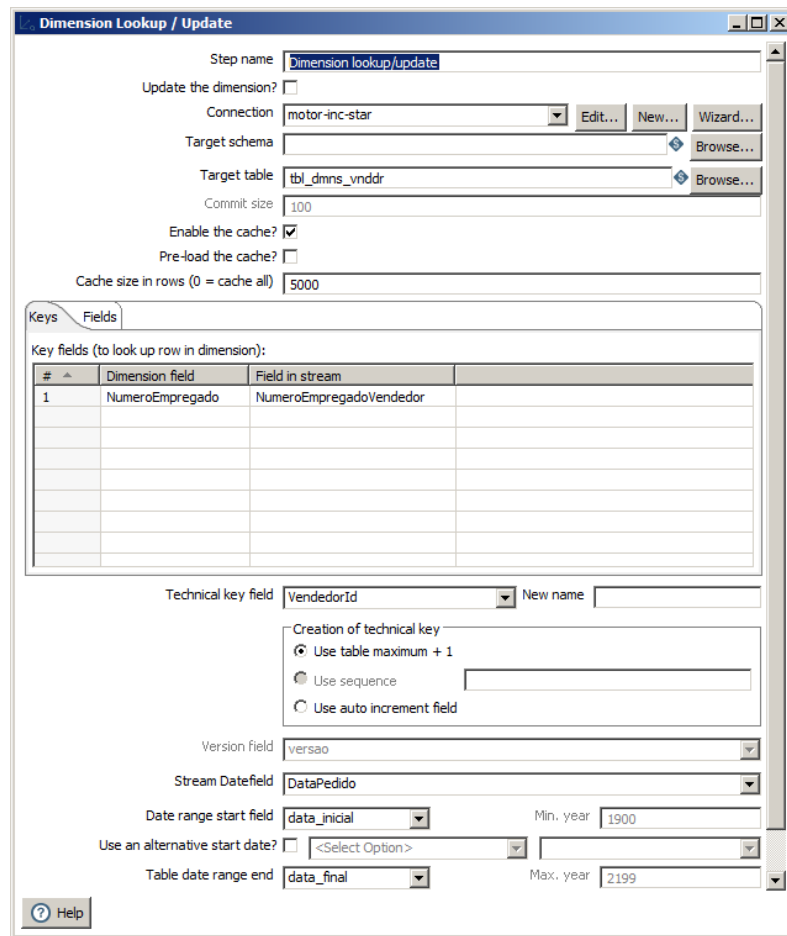
Values to return from the lookup table :

#	Field	New name	Default	Type
1	TempoId	TempoIdEntrega	-1	Integer

Do not pass the row if the lookup fails ☐

Fail on multiple results? ☐

Order by:



**Dimension Lookup / Update**

Step name:

Update the dimension? ☐

Connection:

Target schema:

Target table:

Commit size:

Enable the cache? ☒

Pre-load the cache? ☐

Cache size in rows (0 = cache all):

**Keys** **Fields**

Key fields (to look up row in dimension):

#	Dimension field	Field in stream
1	NumeroEmpregado	NumeroEmpregadoVendedor

Technical key field:  New name:

Creation of technical key:

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field:

Stream Datefield:

Date range start field:  Min. year:

Use an alternative start date? ☐

Table date range end:  Max. year:

5. Na sequencia do lay-out apresentado no início deste exercício, utilize os seguintes steps:

a. *Calculator* (Localizado na categoria *Transform*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Calculator>

b. *Select values* (Localizado na categoria *Transform*)

c. *Table output* (Localizado na categoria *Output*)

d. *Text file output* (Localizado na categoria *Output*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Text+File+Output>



Parametrize conforme imagens abaixo.

Calculator

Step name: Calculator

Fields:

#	New field	Calculation	Field A	Field B	Field C	Valu...	Length	Precision	Remove	Conversion mask
1	PrecoTotal	A * B	PrecoUnitario	QuantidadePedida		Number			N	0.00

Help OK Cancel

Select / Rename values

Step name: Select values

Select & Alter Remove Meta-data

Fields:

#	Fieldname	Rename to	Length	Precision
1	NumeroPedido			
2	QuantidadePedida			
3	NumeroLinhaPedido			
4	Situacao			
5	ClienteId			
6	ProdutoId			
7	TempoIdPedido			
8	TempoIdEntrega			
9	VendedorId			
10	PrecoTotal			

Get fields to select Edit Mapping

Include unspecified fields, ordered by name ☐

Help OK Cancel

**Table output**

Step name:

Connection:

Target schema:

Target table:

Commit size:

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

**Main options** **Database fields**

Partition data over tables: ☐

Partitioning field:

Partition data per month: ☐

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table:

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field:

**Text file output**

Step name:

**File** **Content** **Fields**

Filename:

Run this as command instead?: ☐

Pass output to servlet: ☐

Create Parent folder: ☒

Do not create file at start: ☐

Accept file name from field?: ☐

File name field:

Extension:

Include stepnr in filename?: ☐

Include partition nr in filename?: ☐

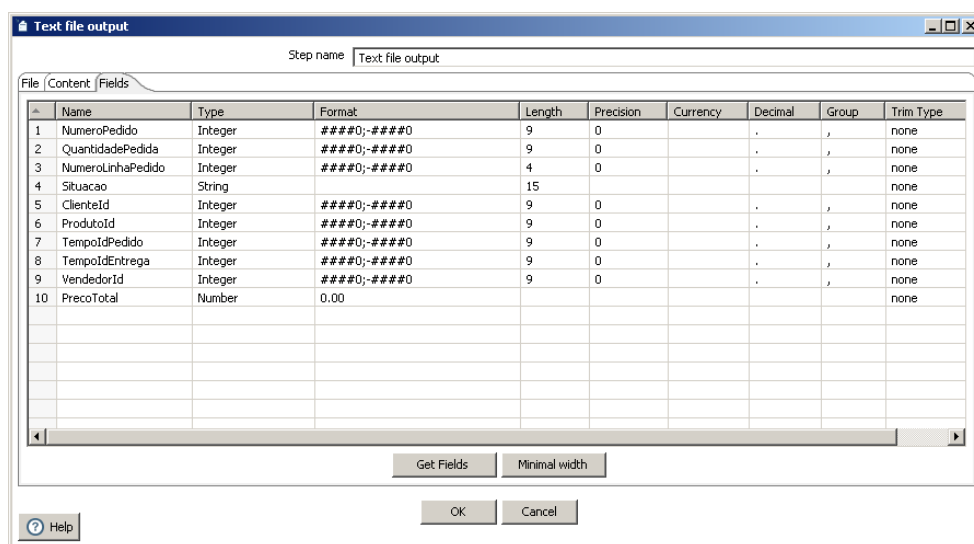
Include date in filename?: ☐

Include time in filename?: ☐

Specify Date time format:

Date time format:

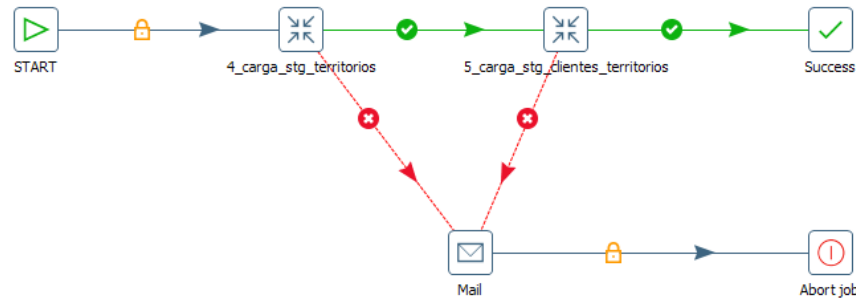
Add filenames to result: ☒



6. Salve a *transformation* como **9\_carga\_fato\_vendas**;
7. Execute a *transformation* (menu Action > Run) e observe os dados carregados no banco de dados através da ferramenta HeidiSQL; e
8. Fechar a *transformation*.

## 10 – Orquestrando a execução dos processos da Staging

Exemplo de como será o *job*



**Objetivo:** Neste Job, apresentaremos ao aluno, como encadear as chamadas, primeiramente pelos processos da área de estagiamento, resultando em sucesso ou em falha e neste caso, enviando e-mail de erro.

Tempo médio para a construção do exercício: **15 minutos**

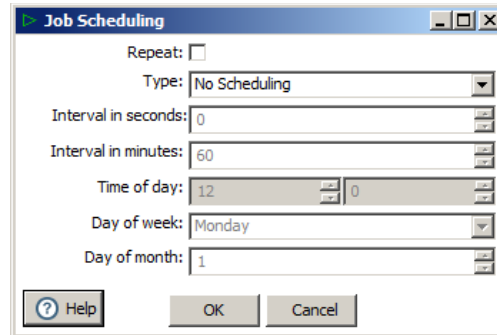
Complexidade para a construção do exercício: **baixa**

1. Crie um novo *job*;
2. Arraste para a área de trabalho os seguintes steps e desenhe o layout igual a imagem acima:
  - a. *Start* (Localizado na categoria *General*)  
AJUDA: NOT AVAILABLE
  - b. *Transformation* (Localizado na categoria *General*)  
AJUDA:  
<http://wiki.pentaho.com/display/EAI/Transformation+%28job+entry%29>
  - c. *Transformation* (Localizado na categoria *General*)
  - d. *Mail* (Localizado na categoria *Mail*)  
AJUDA: <http://wiki.pentaho.com/display/EAI/Mail>
  - e. *Success* (Localizado na categoria *General*)  
AJUDA: <http://wiki.pentaho.com/display/EAI/Success>

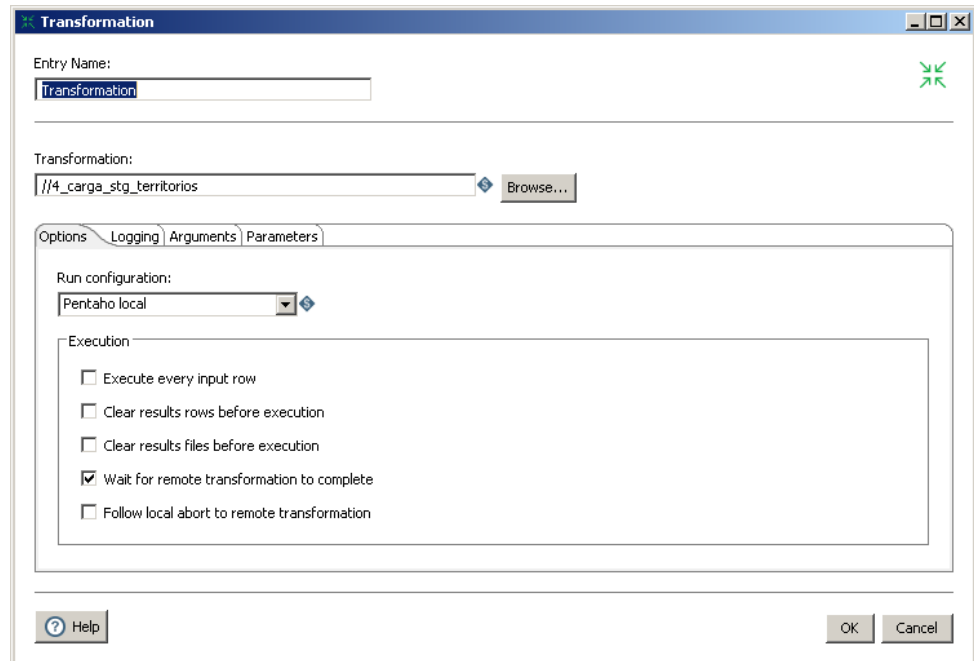
f. *Abort Job* (Localizado na categoria *Utility*)

AJUDA: <http://wiki.pentaho.com/display/EAI/Abort+Job>

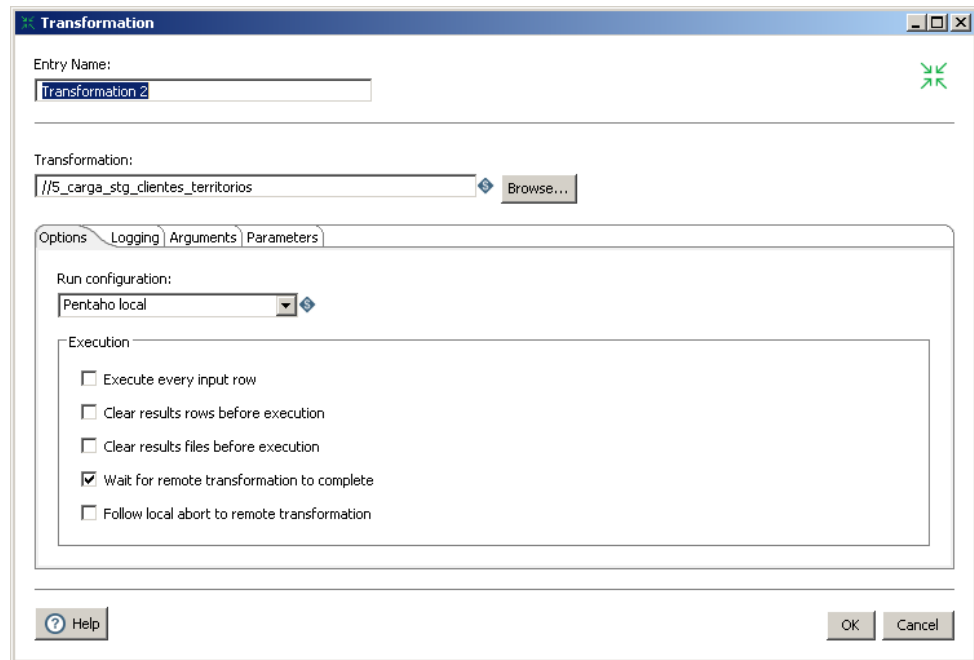
Parametrize conforme imagens abaixo.



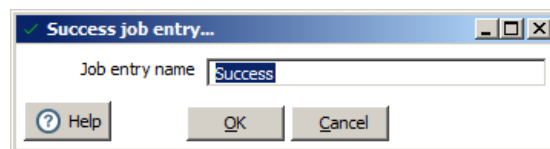
The 'Job Scheduling' dialog box is shown with the following settings: Repeat is unchecked; Type is 'No Scheduling'; Interval in seconds is 0; Interval in minutes is 60; Time of day is 12:00; Day of week is 'Monday'; and Day of month is 1. Buttons for Help, OK, and Cancel are at the bottom.



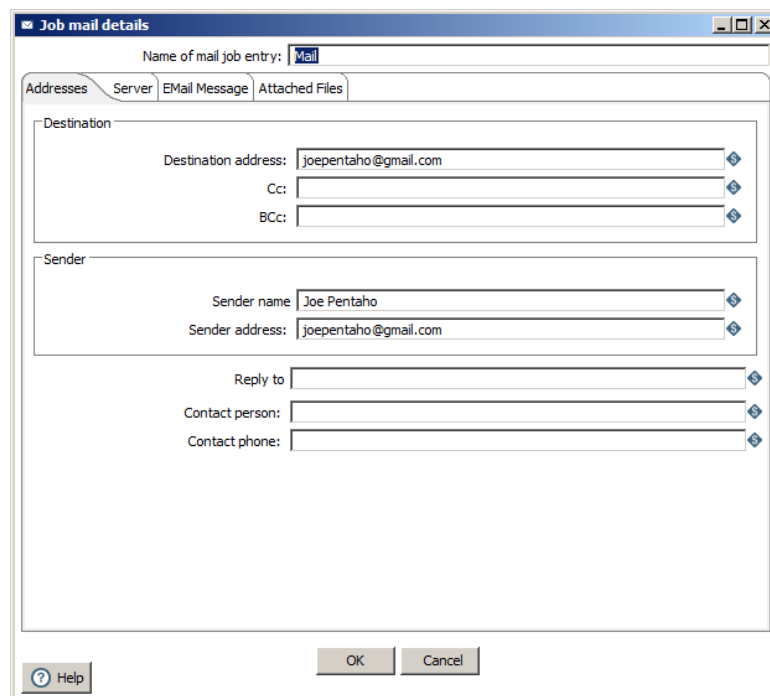
The 'Transformation' dialog box is shown with the following settings: Entry Name is 'Transformation'; Transformation is '//4\_carga\_stg\_territorios'; Run configuration is 'Pentaho local'. Under the 'Options' tab, the 'Execution' section has 'Wait for remote transformation to complete' checked. Buttons for Help, OK, and Cancel are at the bottom.



The 'Transformation' dialog box is shown with the 'Entry Name' field set to 'Transformation 2'. The 'Transformation' field contains the path '//5\_carga\_stg\_clientes\_territorios' and a 'Browse...' button. Below, the 'Options' tab is selected, showing 'Run configuration' set to 'Pentaho local'. The 'Execution' section contains several checkboxes: 'Execute every input row' (unchecked), 'Clear results rows before execution' (unchecked), 'Clear results files before execution' (unchecked), 'Wait for remote transformation to complete' (checked), and 'Follow local abort to remote transformation' (unchecked). At the bottom are 'Help', 'OK', and 'Cancel' buttons.



The 'Success job entry...' dialog box shows the 'Job entry name' field set to 'Success'. It includes 'Help', 'OK', and 'Cancel' buttons at the bottom.



The 'Job mail details' dialog box has the 'Name of mail job entry' field set to 'Mail'. It features four tabs: 'Addresses', 'Server', 'E-Mail Message', and 'Attached Files'. The 'Addresses' tab is active, showing fields for 'Destination' (with 'Destination address' set to 'joepentaho@gmail.com', and empty 'Cc' and 'Bcc' fields) and 'Sender' (with 'Sender name' and 'Sender address' both set to 'joepentaho@gmail.com', and empty 'Reply to', 'Contact person', and 'Contact phone' fields). 'Help', 'OK', and 'Cancel' buttons are at the bottom.

The screenshot shows the 'Job mail details' dialog box with the 'Server' tab selected. The 'Name of mail job entry' is 'Mail'. Under the 'SMTP Server' section, the 'SMTP Server' is 'smtp.gmail.com' and the 'Port' is '465'. Under the 'Authentication' section, 'Use authentication?' is checked, 'Authentication user' is 'joepentaho@gmail.com', 'Authentication password' is masked with dots, 'Use secure authentication?' is checked, and 'Secure connection type' is 'SSL'. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

3. A senha para a conexão SMTP é password0101;

The screenshot shows the 'Job mail details' dialog box with the 'E-Mail Message' tab selected. Under 'Message Settings', 'Include date in message?', 'Only send comment in mail body?', and 'Use HTML format in mail body?' are all unchecked. 'Encoding' is 'UTF-8', 'Manage priority' is unchecked, and 'Priority', 'Importance', and 'Sensitivity' are all set to 'Normal'. Under the 'Message' section, the 'Subject' is 'Job executado com falhas' and the 'Comment' is 'Job do treinamento Openin Big Data foi executado com falhas'. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

**Job mail details**

Name of mail job entry: Mail

Addresses Server EMail Message Attached Files

Files added in result filename

Attach file(s) to message? ☒

Select file type: General  
Log  
Error line  
Error  
Warning

Zip files to single archive? ☐

Name of zip archive:

Embedded images

Filename: Add Browse files ...

Content ID:

#	Image	Content ID
1		

Delete Edit

Help OK Cancel

**Abort job...**

Abort job: Abort job

Message:

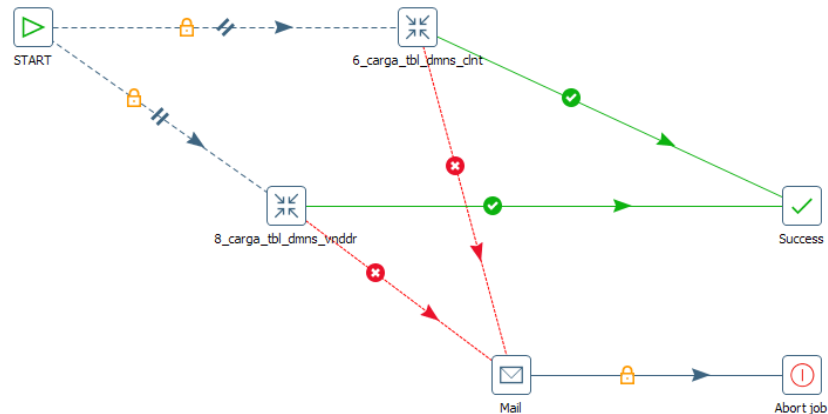
Help OK Cancel

4. Salvar o job como **10\_Carrega\_stage**; e
5. Fechar o *Job*.



## 11 – Orquestrando a execução dos processos Dimensões

Exemplo de como será o *job*



**Objetivo:** Neste Job, apresentaremos ao aluno como encadear as chamadas pelos processos que carregam as dimensões de Produto, Cliente, Tempo e Vendedor, resultando em sucesso ou em falha e neste caso, enviando e-mail de erro.

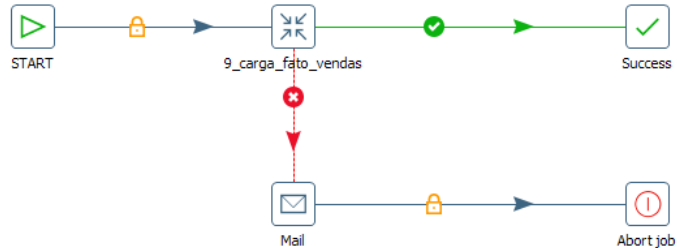
Tempo médio para a construção do exercício: **20 minutos**

Complexidade para a construção do exercício: **média**

1. Este exercício você recebeu por email e está pronto para usar; e
2. Importe o *job* **11\_Carrega\_dimensao\_DW.kjb** e acompanhe a explicação dada pelo instrutor.

## 12 – Orquestrando a execução do processo Fato

Exemplo de como será o *job*



**Objetivo:** Neste Job, apresentaremos ao aluno como encadear a chamada ao processo da tabela fato vendas, resultando em sucesso ou em falha e neste caso, enviando e-mail de erro.

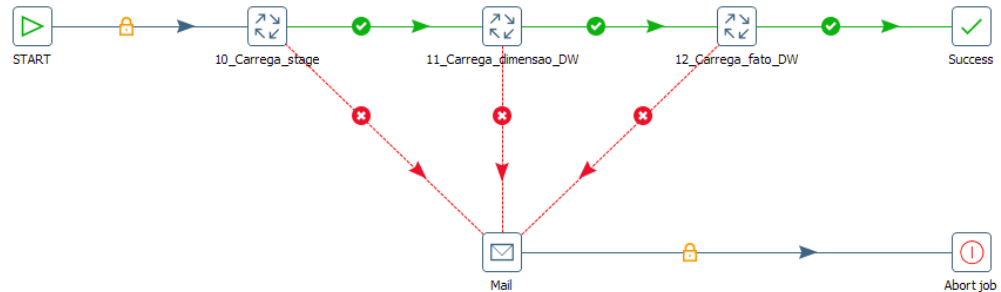
Tempo médio para a construção do exercício: **5 minutos**

Complexidade para a construção do exercício: **baixa**

1. Este exercício você recebeu por email e está pronto para usar; e
2. Importe o *job* **12\_Carrega\_fato\_DW.kjb** e acompanhe a explicação dada pelo instrutor.

## 13 – Orquestrando todos os processos do DW

Exemplo de como será o *job*



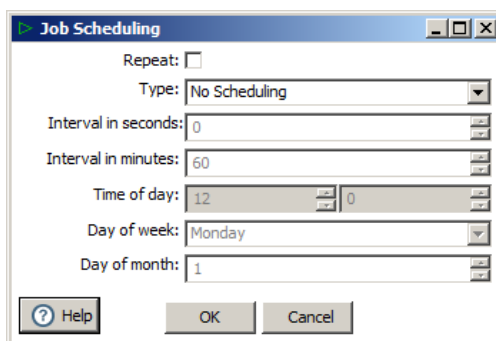
**Objetivo:** Neste Job, apresentaremos ao aluno como encadear as chamadas primeiramente da área de estagiamento, na sequência dos processos das dimensões e por último o processo da tabela fato vendas, resultando em sucesso ou em falha e neste caso, enviando e-mail de erro.

Tempo médio para a construção do exercício: **5 minutos**

Complexidade para a construção do exercício: **média**

1. Crie um novo *job*;
  2. Arraste para a área de trabalho os seguintes steps e desenhe o layout igual a imagem acima
    - a. *Start* (Localizado na categoria General)
    - b. *Job* (Localizado na categoria General)
    - c. *Job* (Localizado na categoria General)
    - d. *Job* (Localizado na categoria General)
    - e. *Mail* (Localizado na categoria Mail)
    - f. *Success* (Localizado na categoria General)
    - g. *Abort Job* (Localizado na categoria Utility)
- AJUDA: <http://wiki.pentaho.com/display/EAI/Job+%28Job+Entry%29>

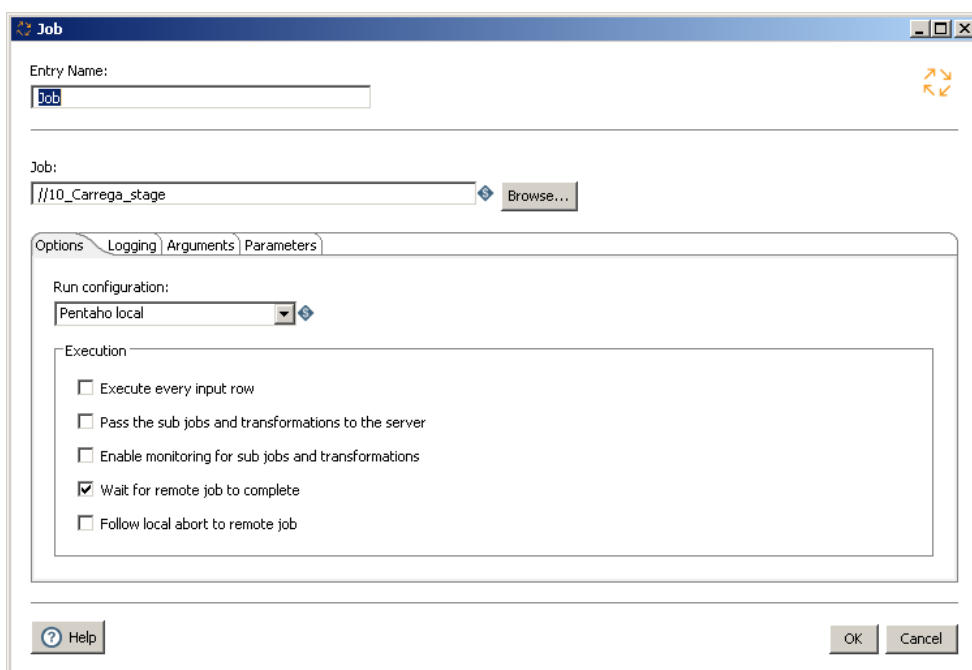
Parametrize conforme imagens abaixo.



The 'Job Scheduling' dialog box is shown with the following settings:

- Repeat: ☐
- Type: No Scheduling
- Interval in seconds: 0
- Interval in minutes: 60
- Time of day: 12:00
- Day of week: Monday
- Day of month: 1

Buttons: Help, OK, Cancel



The 'Job' configuration dialog box is shown with the following settings:

- Entry Name: Job
- Job: //10\_Carrega\_stage
- Run configuration: Pentaho local
- Execution options:
  - ☐ Execute every input row
  - ☐ Pass the sub jobs and transformations to the server
  - ☐ Enable monitoring for sub jobs and transformations
  - ☒ Wait for remote job to complete
  - ☐ Follow local abort to remote job

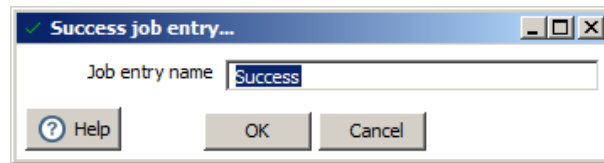
Buttons: Help, OK, Cancel

The screenshot shows the 'Job' configuration window. The 'Entry Name' field contains 'Job 2'. The 'Job' field contains '///11\_Carrega\_dimensao\_DW'. Below these fields are tabs for 'Options', 'Logging', 'Arguments', and 'Parameters'. The 'Options' tab is selected, showing 'Run configuration' set to 'Pentaho local'. Under the 'Execution' section, the following options are listed: 'Execute every input row' (unchecked), 'Pass the sub jobs and transformations to the server' (unchecked), 'Enable monitoring for sub jobs and transformations' (unchecked), 'Wait for remote job to complete' (checked), and 'Follow local abort to remote job' (unchecked). At the bottom, there are 'Help', 'OK', and 'Cancel' buttons.

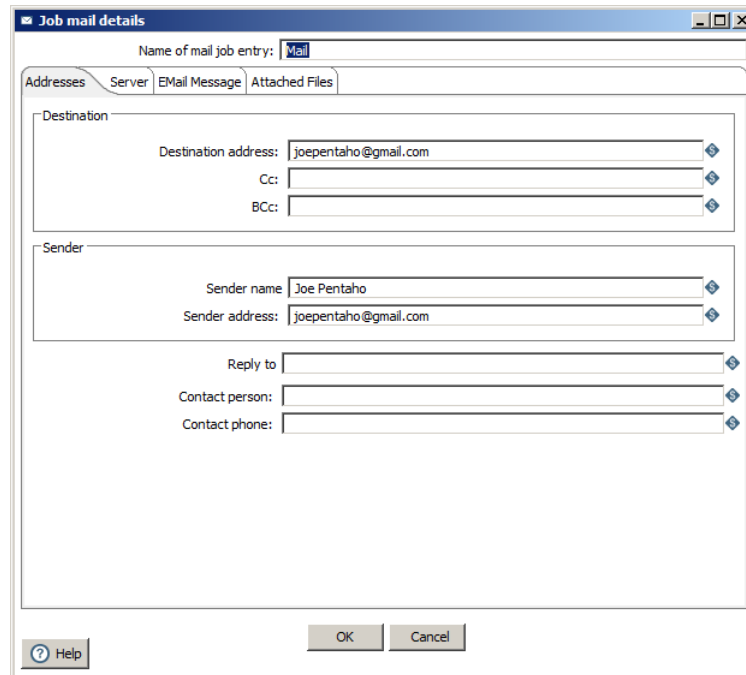
Job configuration window showing Entry Name: Job 2, Job: ///11\_Carrega\_dimensao\_DW, and Execution options (Wait for remote job to complete is checked).

The screenshot shows the 'Job' configuration window. The 'Entry Name' field contains 'Job 3'. The 'Job' field contains '///12\_Carrega\_fato\_DW'. Below these fields are tabs for 'Options', 'Logging', 'Arguments', and 'Parameters'. The 'Options' tab is selected, showing 'Run configuration' set to 'Pentaho local'. Under the 'Execution' section, the following options are listed: 'Execute every input row' (unchecked), 'Pass the sub jobs and transformations to the server' (unchecked), 'Enable monitoring for sub jobs and transformations' (unchecked), 'Wait for remote job to complete' (checked), and 'Follow local abort to remote job' (unchecked). At the bottom, there are 'Help', 'OK', and 'Cancel' buttons.

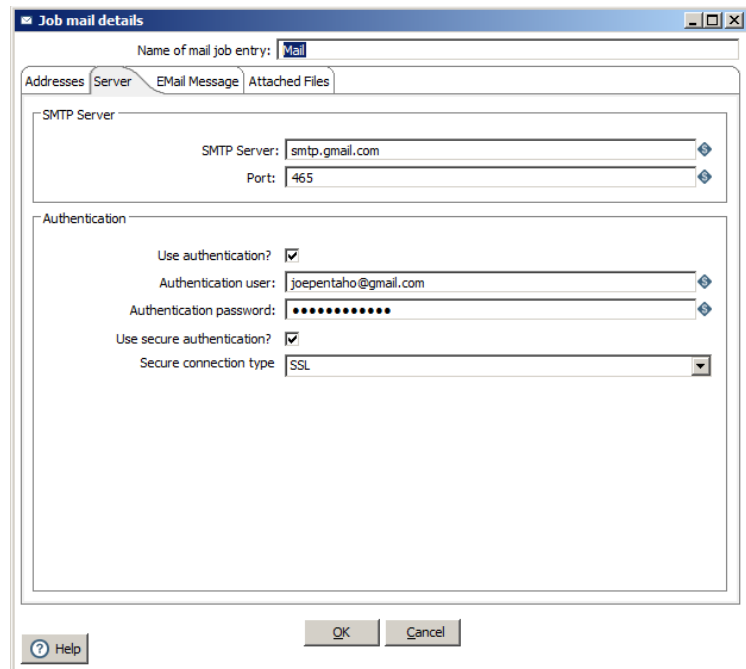
Job configuration window showing Entry Name: Job 3, Job: ///12\_Carrega\_fato\_DW, and Execution options (Wait for remote job to complete is checked).



A small dialog box titled "Success job entry..." with a green checkmark icon. It contains a text field labeled "Job entry name" with the value "Success". At the bottom are three buttons: "Help" (with a question mark icon), "OK", and "Cancel".



A dialog box titled "Job mail details" with tabs for "Addresses", "Server", "E-Mail Message", and "Attached Files". The "Addresses" tab is active. It shows fields for "Destination address" (joepentaho@gmail.com), "Cc", and "Bcc". Below these are fields for "Sender name" (Joe Pentaho), "Sender address" (joepentaho@gmail.com), "Reply to", "Contact person", and "Contact phone". At the bottom are "Help", "OK", and "Cancel" buttons.



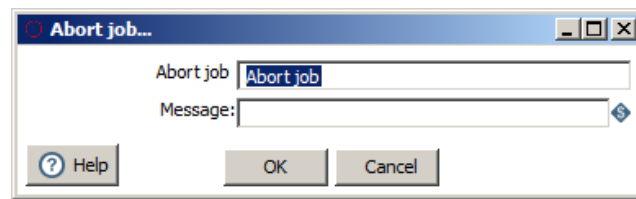
The same "Job mail details" dialog box, but with the "Server" tab active. It shows fields for "SMTP Server" (smtp.gmail.com) and "Port" (465). Below these are checkboxes for "Use authentication?" (checked) and "Use secure authentication?" (checked). The "Authentication user" field contains joepentaho@gmail.com, and the "Authentication password" field is masked with dots. A "Secure connection type" dropdown menu is set to "SSL". At the bottom are "Help", "OK", and "Cancel" buttons.

3. A senha para a conexão SMTP é password0101;

The screenshot shows the 'Job mail details' dialog box with the 'E-Mail Message' tab selected. The 'Name of mail job entry' field contains 'Mail'. The 'Message Settings' section includes checkboxes for 'Include date in message?', 'Only send comment in mail body?', and 'Use HTML format in mail body?'. The 'Encoding' dropdown is set to 'UTF-8'. The 'Manage priority' checkbox is unchecked. The 'Priority', 'Importance', and 'Sensitivity' dropdowns are all set to 'Normal'. The 'Message' section has a 'Subject' field with 'Job executado com falhas' and a 'Comment' field with 'Job do treinamento Openin Big Data foi executado com falhas'. The 'OK' and 'Cancel' buttons are at the bottom right, and a 'Help' button is at the bottom left.

The screenshot shows the 'Job mail details' dialog box with the 'Attached Files' tab selected. The 'Name of mail job entry' field contains 'Mail'. The 'Files added in result filename' section has the 'Attach file(s) to message?' checkbox checked. The 'Select file type' dropdown is open, showing options: 'General', 'Log', 'Error line', 'Error', and 'Warning'. The 'Zip files to single archive?' checkbox is unchecked. The 'Name of zip archive' field is empty. The 'Embedded images' section has a 'Filename' field, a 'Content ID' field, and 'Add' and 'Browse files ...' buttons. Below this is a table with columns '#', 'Image', and 'Content ID'. The table contains one row with the number '1' in the '#' column. To the right of the table are 'Delete' and 'Edit' buttons. The 'OK' and 'Cancel' buttons are at the bottom right, and a 'Help' button is at the bottom left.

#	Image	Content ID
1		



4. Salvar o job como **13\_Job\_Principal**; e
5. Fechar o *Job*.



## 14 – Executando um Job através de linha de comando

**Objetivo:** Neste exercício apresentaremos ao aluno como executar um Job através de linha comando, usando a ferramenta Kitchen.

Tempo médio para a construção do exercício: **20 minutos**

Complexidade para a construção do exercício: **média**

1. Crie um arquivo chamado executa\_job.bat com as seguintes instruções (use o notepad++); e

```
echo Iniciando Job
```

```
cd C:\treinamento\design-tools\data-integration
```

```
kitchen.bat /rep:repo /job:13_Job_Principal /dir: /level:Basic  
/log:"C:\treinamento\design-tools\data-integration\executa_job.log" >  
"C:\treinamento\design-tools\data-integration\executa_job_console.log"
```

**ATENÇÃO:** Os comandos acima para executem com sucesso, devem ficar cada um em uma única linha.

AJUDA: <http://wiki.pentaho.com/display/EAI/Kitchen+User+Documentation>

2. Agora é só colocar o arquivo em alguma ferramenta de agendamento e agendar a execução do mesmo.