

Bootcamp IGTI: Engenheiro de Dados

Desafio do módulo

Módulo 2	Armazenamento de Dados
-----------------	-------------------------------

Objetivos

O objetivo desse exercício é fazer um processo simplificado de ETL no Pentaho a partir de arquivos do tipo **csv**.

Atividades

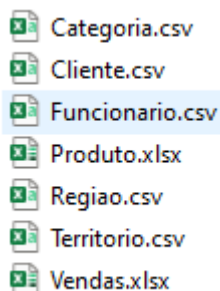
Os alunos deverão desempenhar as seguintes atividades:

1. Utilizar o banco de dados MySQL e o Pentaho. É possível utilizar outro gerenciador de banco de dados relacional de sua preferência.
2. Executar todo o processo de ETL no Pentaho conforme orientações.

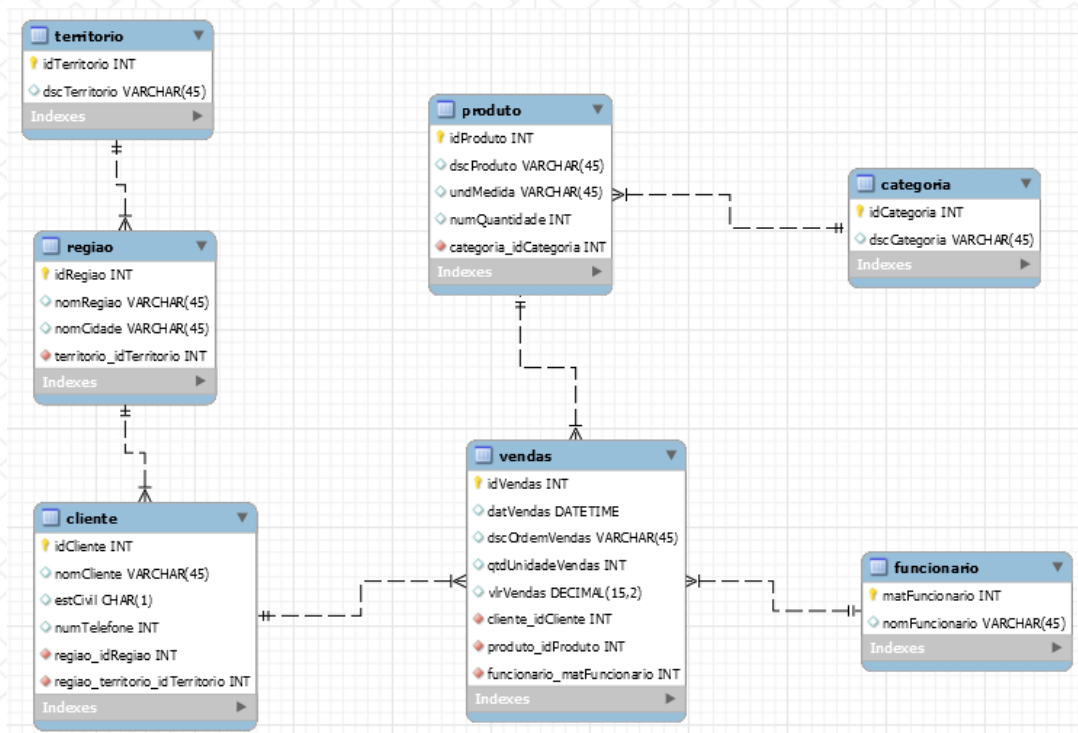
Enunciado

A partir de arquivos de dados (csv) que nos servirão como dados da origem – e que estão disponíveis no link abaixo – vamos modelar um DW.

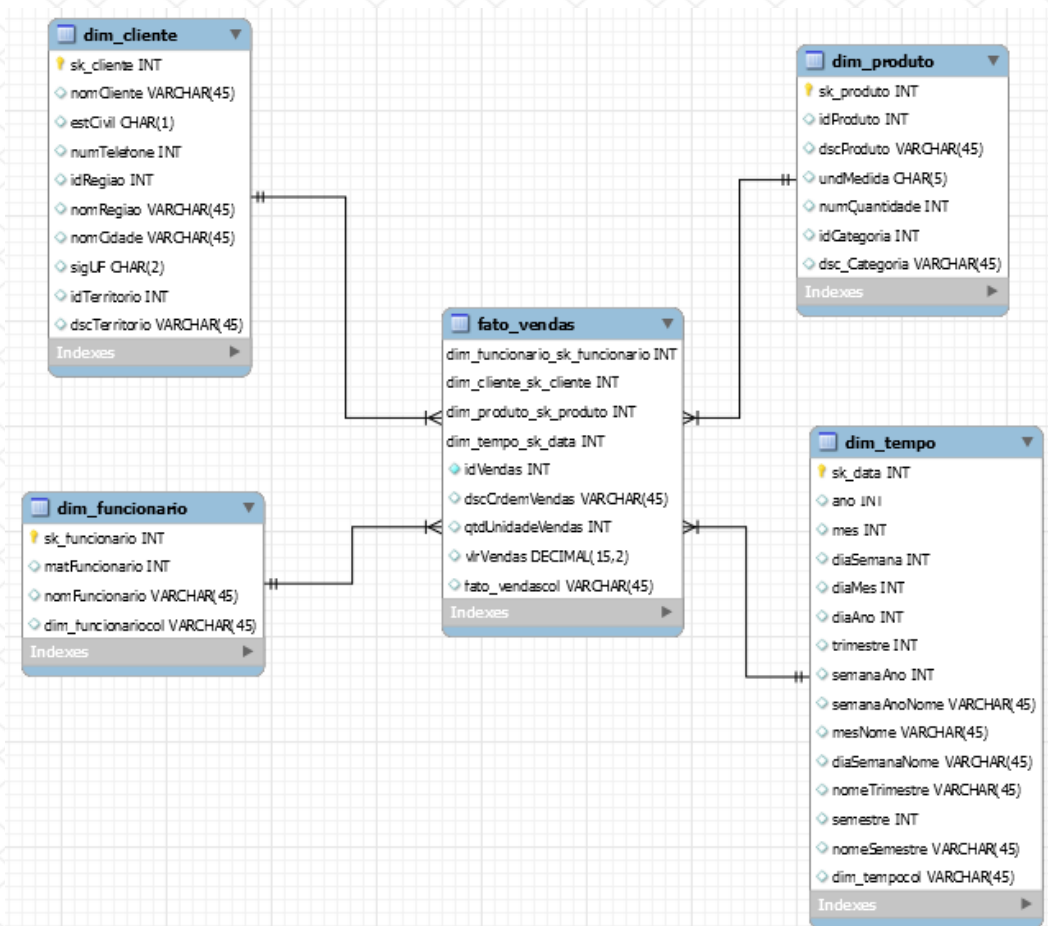
- https://drive.google.com/drive/folders/1sqqGG2eNmrr_39pRPUKY095-A10_cnnQ?usp=sharing



Esses dados seguem a seguinte modelagem:

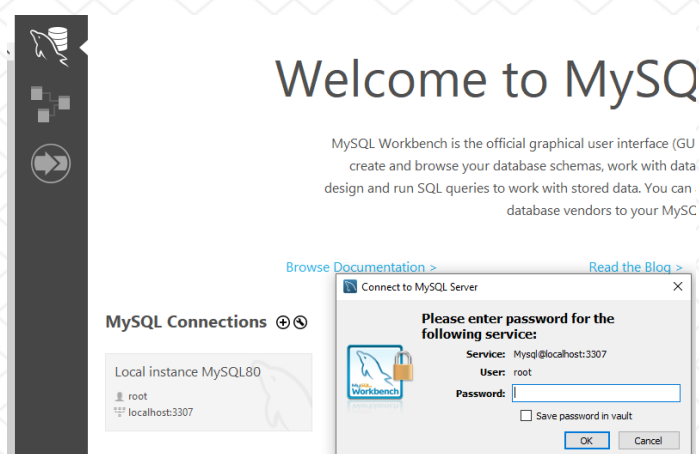


A partir desses dados de origem, o objetivo é modelar um DW (star schema) como esquema abaixo:

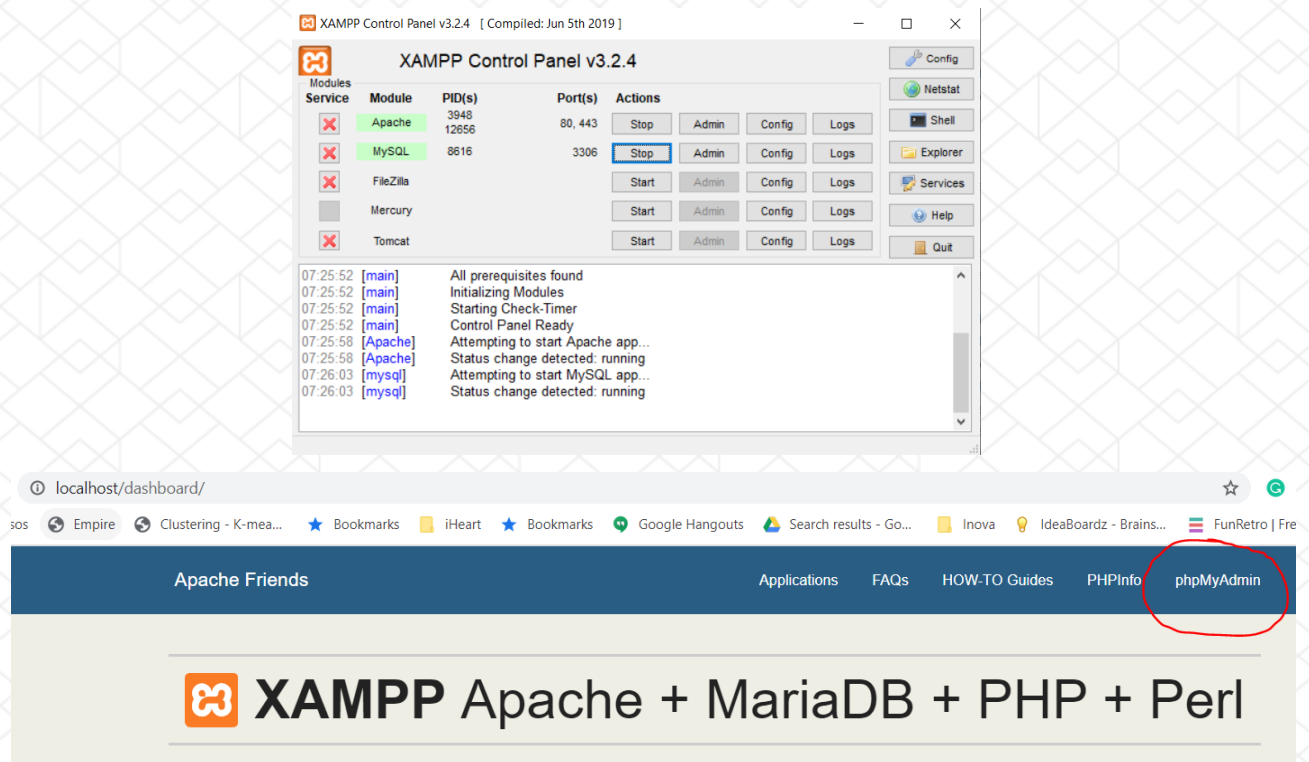


Sugestão para uso do MySql, origem (staging) e DW:

Utilizar o MySql via Workbench como o repositório para a origem.



O MySql via XAMPP/PHPAdmin servirá como o repositório para o DW.



Se preferir, pode utilizar somente uma instalação do MySql.

Precisamos de dois schemas diferentes, um para a Staging e outro para o DW. Crie os schemas no(s) MySql manualmente **ou** via comando sql:

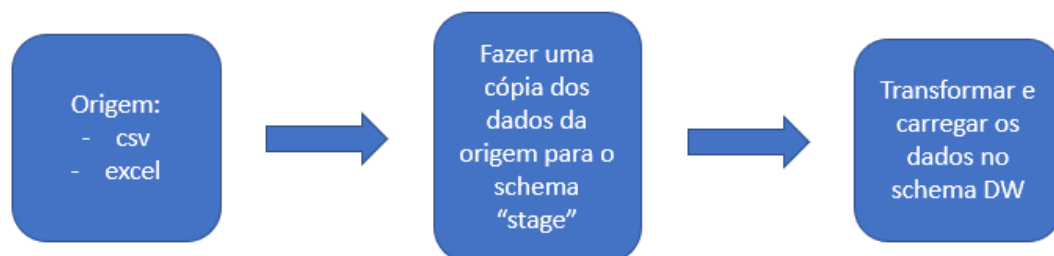
Exemplo:

No Workbench

```
CREATE SCHEMA IF NOT EXISTS `stage` DEFAULT CHARACTER SET utf8 ;
```

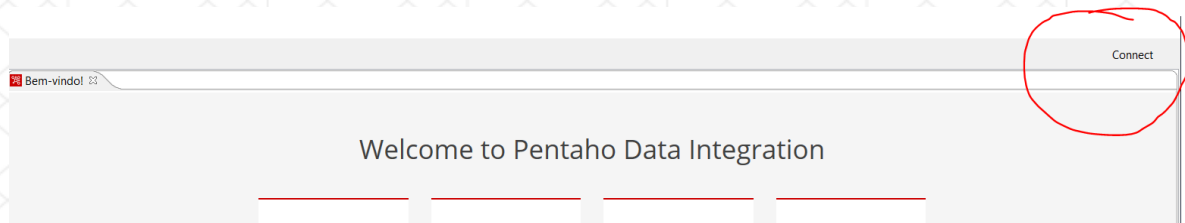
No XAMPP Apache (ele chama Schema de Database, é a mesma coisa)

```
CREATE SCHEMA IF NOT EXISTS `dw` DEFAULT CHARACTER SET utf8 ;
```

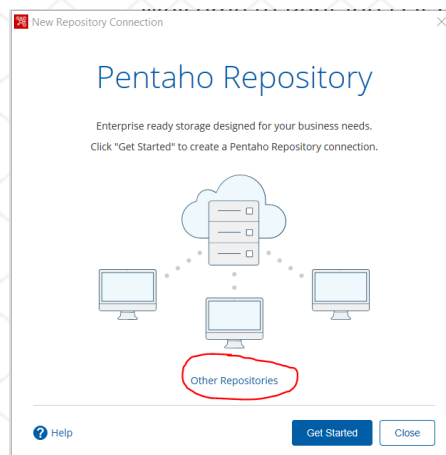


Vamos começar criando um repositório para o projeto do desafio.

Clique no botão conect.



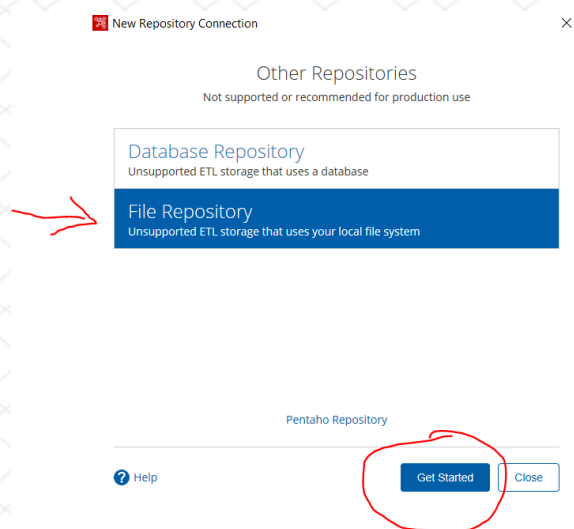
Clique em “Other Repositories”



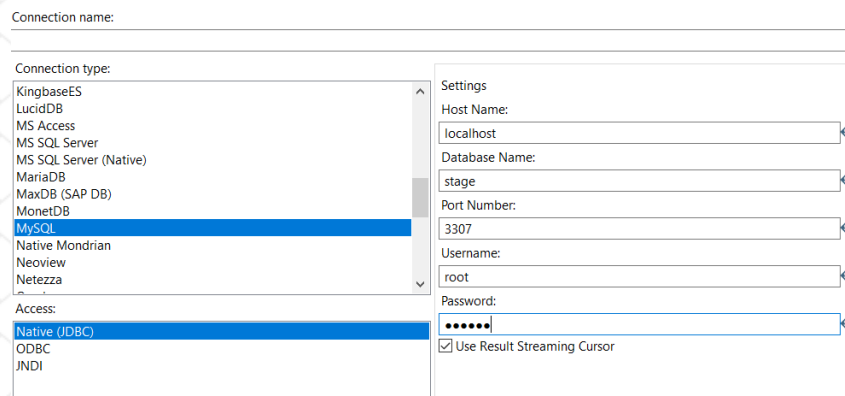
Selecione “File Repository” e clique em “Get Started”. Configure com o diretório de sua preferência para mandar o projeto salvo.

Copie as transformações anexadas ao desafio para o diretório que você criou e que vai servir como repositório:

- Origem.ktr
- Dimensao_Data.ktr
- Dimensao_Produto.ktr



Para iniciar os trabalhos, vamos criar as conexões com os bancos MySql.



Vamos utilizar o arquivo disponível no desafio chamado “origem.ktr” para trabalhar com as transformações, trazendo os dados da origem para a stage. Abra esse arquivo no Pentaho.

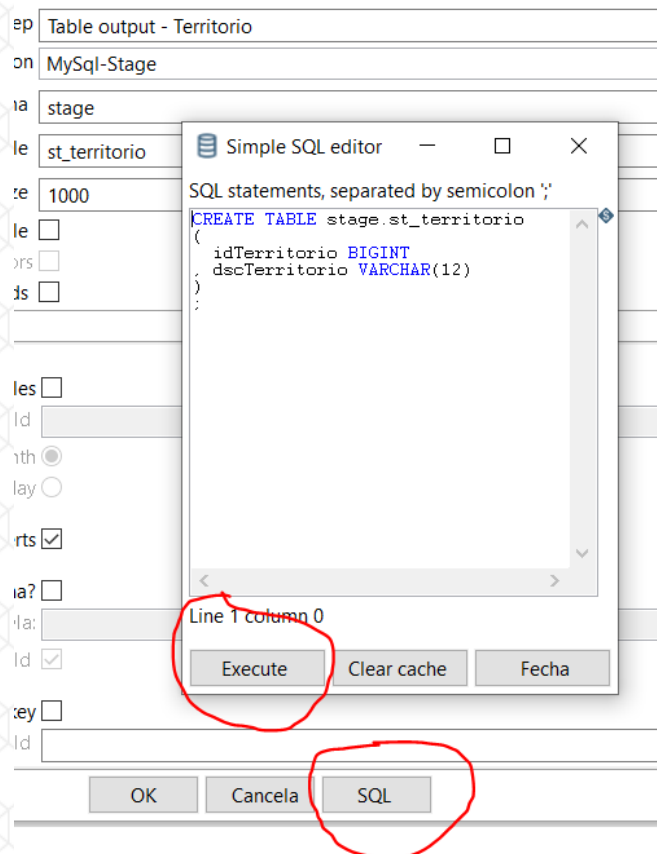
Você precisará fazer as seguintes alterações:

1. Alterar os steps CSV file input para que apontem para os arquivos da origem.

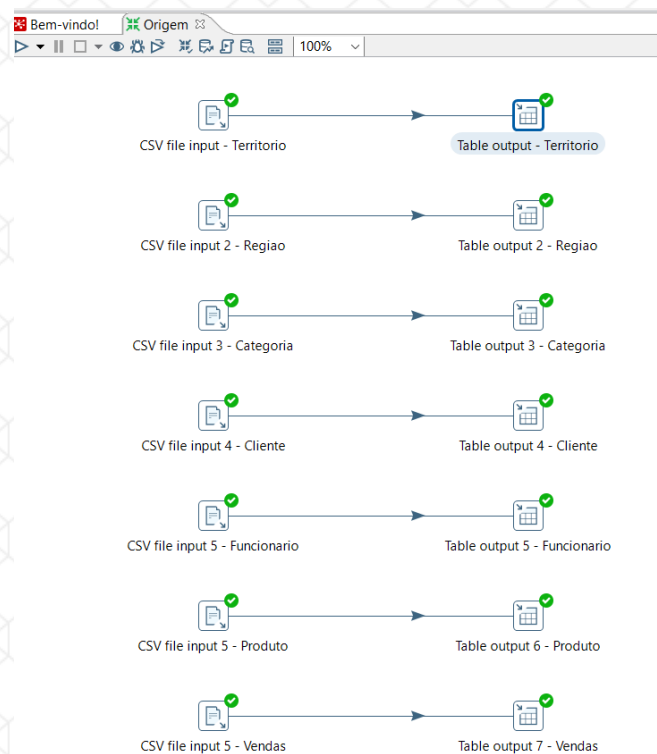
2. Alterar o steps Table output para que:

- Ajuste a conexão de forma que funcione conectada ao MySQL Workbench, por exemplo.
 - Aponte para o schema (Database) que receberá as tabelas da Staging.
3. O nome da tabela a ser criada no banco de dados é o nome que está no campo Target table. Portanto, é só clicar no botão SQL que será possível criar a tabela.

Reforçando, não é preciso criar as tabelas da stage e do DW nos bancos MySQL de forma manual. Você consegue fazer isso, por exemplo, pelo componente table output.

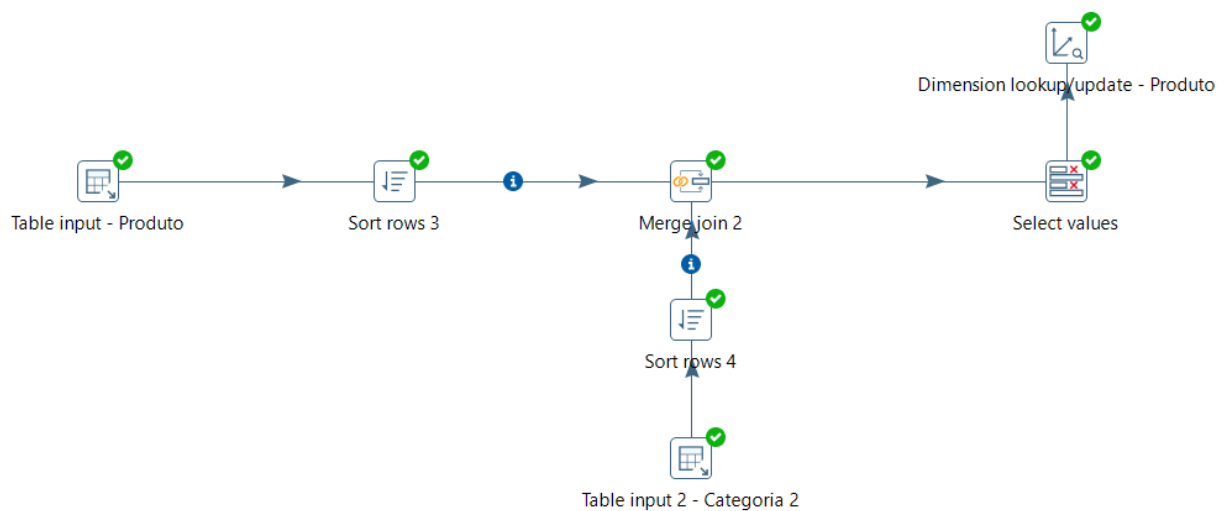


Portanto, a transformação ficará assim:



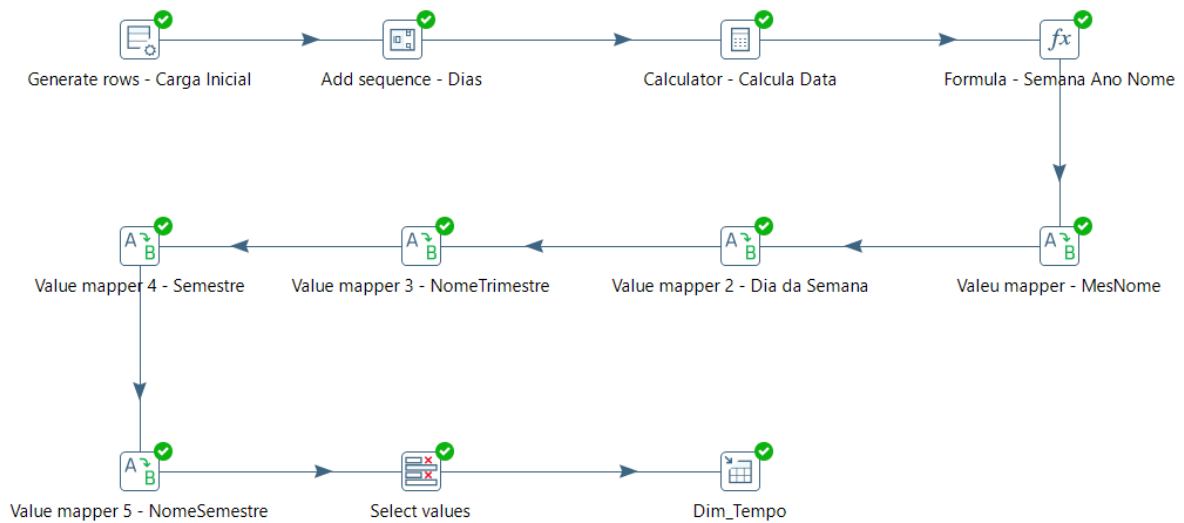
Carga para Dimensão Produto: abra no Pentaho o arquivo fornecido – Dimensao_Produto.ktr.

- Altere as conexões dos steps Table input para Produto e para Categoria.
- Avalie as configurações de cada step aqui presente.
- Rode a transformação.

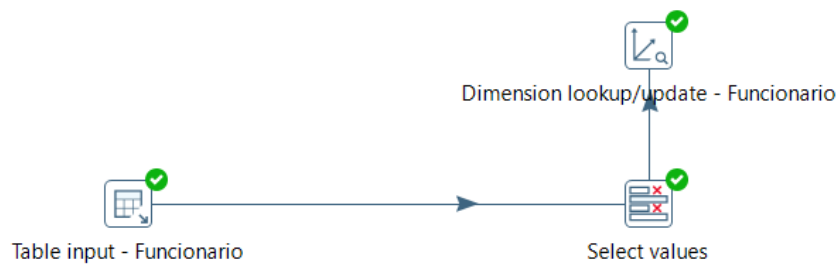


Carga para Dimensão Tempo: abra no Pentaho o arquivo fornecido – Dim_Tempo.ktr.

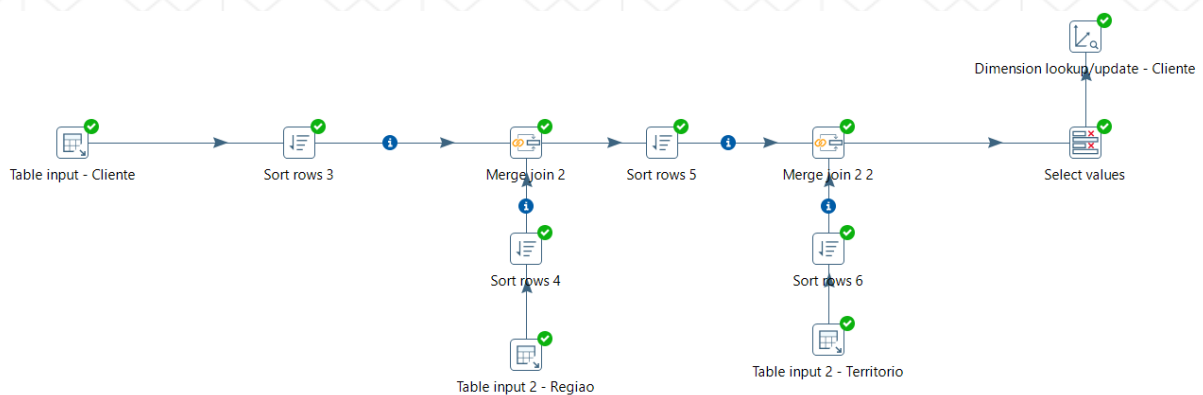
- Altere as conexões do step Table output chamado Dim_Tempo
- Avalie as configurações de cada step aqui presente.
- Rode a transformação.



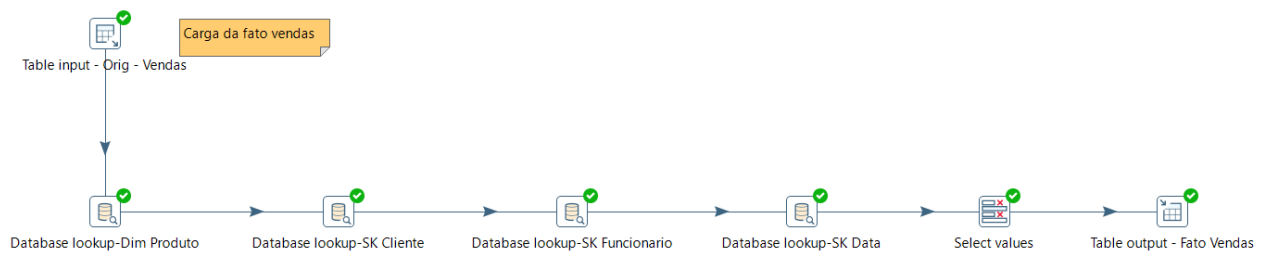
Carga para Dimensão Funcionário: faça essa transformação baseada no que já vimos até agora.



Carga para Dimensão Cliente: faça essa transformação baseada no que já vimos até agora.



Carga da tabela Fato Venda: tente configurar a transformação conforme imagem abaixo.



Os steps lookup devem ser configurados conforme exemplo do Database lookup-SK-Cliente:

Lookup de valor do banco de dados

Nome do Step: Database lookup_DimCliente

Connection: Xampp5

Lookup schema: dw1

Tabela Lookup: dim_cliente

Habilita cache? ☐

Tamanho do cache em linhas (0=cache total): 0

Load all data from table ☐

A chave(s) para examinar o valor(s):

#	Campo da tabela	Comparador	Campo1	Campo2
1	idClientes	=	Clientes_idClientes	

Valores a serem retornados da tabela lookup:

#	campo	Novo nome	Default	Tipo
1	sk_cliente			Integer

Não passa a linha se o lookup falhar ☐

Falha quando ocorrerem resultados múltiplos ☐

Ordem por:

Buttons: Help, OK, Cancela, Obtem Campos, Obtem campos lookup

Repare, pelo “preview”, que nas últimas linhas da tabela fato a sk_produto está nula. Isso não deu erro porque a tabela fato não está com as surrogate keys das dimensões setada como chave.

Vamos fazer um ajuste e ver o que acontece.

Rode o comando sql no banco de dados do schema onde está o DW.

```
ALTER TABLE `fato_vendas`
ADD PRIMARY KEY (`sk_produto`,`sk_cliente`,`sk_funcionario`,`sk_data`);
```




```
ALTER TABLE `fato_vendas` ADD PRIMARY KEY (`sk_produto`, `sk_cliente`, `sk_funcionario`, `sk_data`)
```

Warning: #1265 Data truncated for column 'sk_funcionario' at row 161

Warning: #1265 Data truncated for column 'sk_funcionario' at row 162

Warning: #1265 Data truncated for column 'sk_funcionario' at row 163

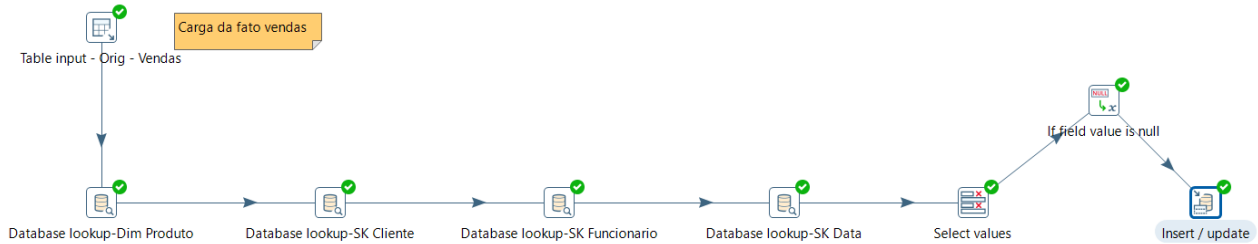
Warning: #1265 Data truncated for column 'sk_funcionario' at row 164

Warning: #1265 Data truncated for column 'sk_funcionario' at row 165

Agora que ligamos as chaves primárias na fato, tente rodar a carga da fato novamente. Você vai obter o erro: “Column ‘sk_funcionario’ cannot be null”.

```
2020/08/08 10:26:47 - Database lookup SK Data.0 - finished processing (I=165, O=0, R=165, W=165, U=0, E=0)
2020/08/08 10:26:47 - Select values.0 - Finished processing (I=0, O=0, R=165, W=165, U=0, E=0)
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - ERROR (version 9.0.0.0-423, build 9.0.0.0-423 from 2020-01-31 04:53.04 by buildguy) : Unexpected batch
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - ERROR (version 9.0.0.0-423, build 9.0.0.0-423 from 2020-01-31 04:53.04 by buildguy) : org.pentaho.di.co
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - Error updating batch
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - Column 'sk_funcionario' cannot be null
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - 
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - at org.pentaho.di.core.database.Database.createKettleDatabaseBatchException(Database.java:1434)
2020/08/08 10:26:47 - Table output - Fato Vendas.0 - at org.pentaho.di.core.database.Database.commitAndCommit(Database.java:1433)
```

Vamos resolver isso trocando o step “table output” por um step “insert update”.



Vamos configurar quais são as chaves na tabela, isso vai permitir fazer um update em caso de insert sem sucesso.

Repare que as dimensões o primeiro registro tem a **Surrogate Key = 1 e demais registros vazios**. Ele serve para você utilizá-lo em caso de inconsistência. Portanto, vamos utilizar o atributo com a SK = 1 para o caso de erro na carga.

	sk_cliente	versao	date_from	date_to	idClientes	nomCliente	estCivil	numTelefone	idRegiao	nomRegia
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	1	2020-11-07 13:14:28	2199-12-31 23:59:59	1	Cliente XY-1	C	92979881234	1	Centro-Nor
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	1	2020-11-07 13:14:28	2199-12-31 23:59:59	2	Cliente XY-2	S	48979881235	2	Centro-Sul
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	28	1	2020-11-07 13:14:28	2199-12-31 23:59:59	3	Cliente XY-3	S	81979881236	3	Centro-Nordeste
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	35	1	2020-11-07 13:14:28	2199-12-31 23:59:59	4	Cliente XY-4	D	63979881237	4	Sul-CentroOes
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	42	1	2020-11-07 13:14:28	2199-12-31 23:59:59	5	Cliente XY-5	C	11979881238	5	Sul-Sudest

Teremos que inserir um step “if field is null”. Configure conforme figura abaixo.

No caso de termos valores nulos para as surrogate keys, vamos atribuir um valor = 1 para eles.

Lembrando que o Pentaho reserva o primeiro registro com a SK = 1. Os campos são nulos nessa linha e ele serve exatamente para tratar erros como acima.

If field value is null

Step name: If field value is null

Replace Null for all fields

Replace by value:

Set empty string? ☐

Mask (Date):

Select fields ☒

Select value type ☐

Value types

#	Type	Replace by value	Conversion mask (Date)	Set empty string?

Fields

#	Field	Replace by value	Conversion mask (Date)	Set empty string?
1	sk_produto	1		N
2	sk_cliente	1		N
3	sk_funcionario	1		N
4				

< >

Help OK Obtem campos Cancela

Configure o step “insert / update” conforme figura abaixo.

Insert / update

Step name: Insert / update

Connection: MySql-Dw Edit... New... Wizard...

Target schema: dw Navega...

Target table: fato_vendas Browse...

Commit size: 100

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream
1	sk_produto	=	sk_produto	
2	sk_cliente	=	sk_cliente	
3	sk_funcionario	=	sk_funcionario	
4	sk_data	=	sk_data	

< >






























Update fields:

#	Table field	Stream field	Update
1	idVendas	idVendas	Y
2	dscOrdemVendas	dscOrdemVendas	Y
3	qtdUnidadeVendas	qtdUnidadeVendas	Y
4	vlrVendas	vlrVendas	Y

Get update fields Edit mapping

Help OK Cancela SQL

Rode a carga e depois confira o resultado no MySQL. Verique que as últimas linhas tem a sk_funcionario com valores = 1. Ou seja, há dados na tabela fato referenciando dados nas dimensões, porém esse dados na dimensão estão ausentes.

				idVenc	dscOrdemVendas	qtdUnidadeVenc	vlrVendas	sk_prod	sk_cli	sk_funciona	sk_data
					84	Ordem 84		300	3000.0		
								10	5	5	2020-01-05 00:00:00
					126	Ordem 126		300	3000.0		
								24	64	17	2020-01-10 00:00:00
					135	Ordem 135		1200	12000.0		
								25	67	23	2020-01-10 00:00:00
					144	Ordem 144		1500	15000.0		
								26	34	23	2020-01-10 00:00:00
					13	Ordem 13		400	4000.0		
								27	4	14	2020-01-02 00:00:00
					89	Ordem 89		800	8000.0		
								27	63	10	2020-01-05 00:00:00
					97	Ordem 97		1600	16000.0		
								28	41	2	2020-01-05 00:00:00
					21	Ordem 21		1200	12000.0		
								28	51	6	2020-01-02 00:00:00
					165	Ordem 165		100	8000.0		
								28	59	1	2020-01-10 00:00:00
					161	Ordem 161		600	6000.0		
								28	83	1	2020-01-10 00:00:00
					162	Ordem 162		700	7000.0		
								28	84	1	2020-01-10 00:00:00

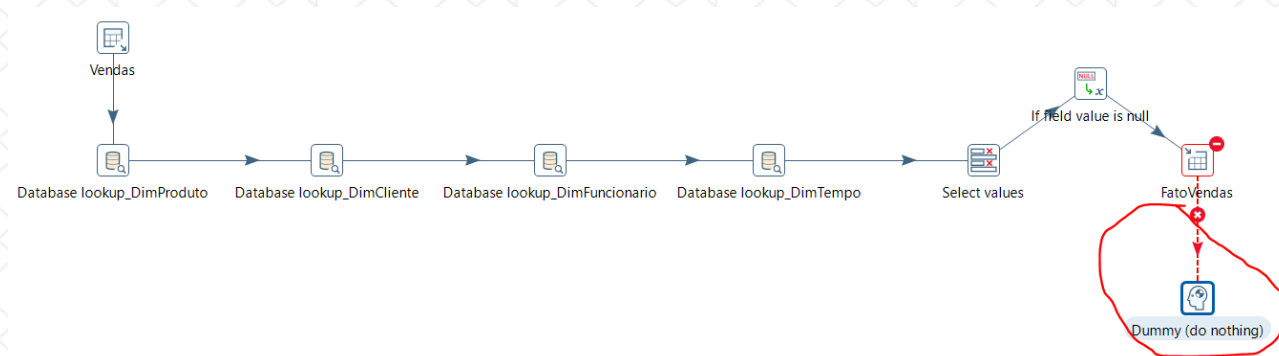
Se você tentar rodar novamente a carga da tabela fato, como ela faz um processo de insert, ocorrerá erro na carga por conta da chave primária.

```

2020/12/02 15:17:09 - FatoVendas.0 - ERROR (version 8.2.0.0-342, build 8.2.0.0-342 from 2018-11-14 10.30.55 by buildguy) : Unexpected batch
2020/12/02 15:17:09 - FatoVendas.0 - ERROR (version 8.2.0.0-342, build 8.2.0.0-342 from 2018-11-14 10.30.55 by buildguy) : org.pentaho.di.cor
2020/12/02 15:17:09 - FatoVendas.0 - Error updating batch
2020/12/02 15:17:09 - FatoVendas.0 - Duplicate entry '28-59-1-2020-01-10 00:00:00' for key 'PRIMARY'
2020/12/02 15:17:09 - FatoVendas.0 - 
2020/12/02 15:17:09 - FatoVendas.0 - at org.pentaho.di.core.database.Database.createKettleDatabaseBatchException(Database.java:1425)
2020/12/02 15:17:09 - FatoVendas.0 - at org.pentaho.di.core.database.Database.emptyAndCommit(Database.java:1414)
2020/12/02 15:17:09 - FatoVendas.0 - at org.pentaho.di.trans.steps.tableoutput.TableOutput.dispose(TableOutput.java:590)
2020/12/02 15:17:09 - FatoVendas.0 - at org.pentaho.di.trans.step.RunThread.run(RunThread.java:97)

```

Você pode resolver isso pelo uso do step Dummy (do nothing):



Ou você pode usar um step Insert/update ao invés do step Table output.

Insert / update

Step name: Insert / update

Connection: Xampp5

Target schema: dw1

Target table: Fato_Vendas

Commit size: 100

Don't perform any updates: ☐

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

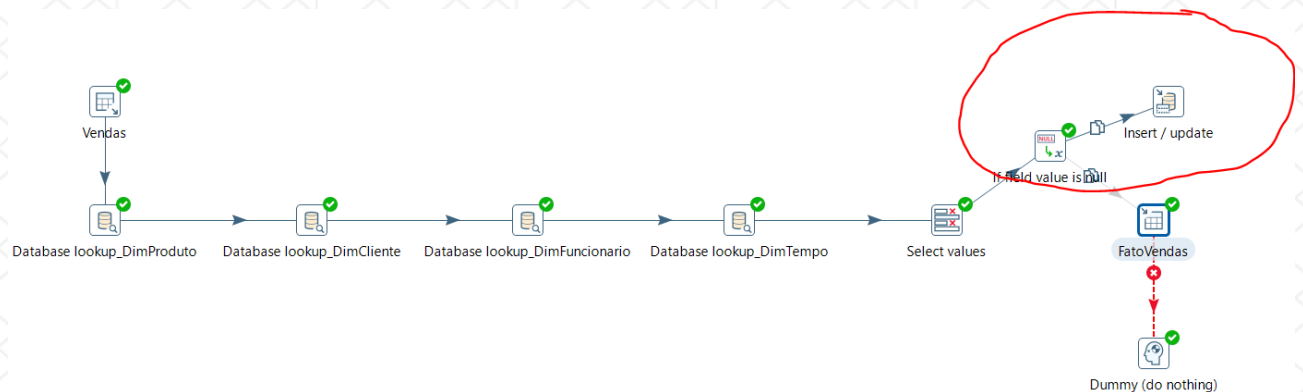
#	Table field	Comparator	Stream field1	Stream field2
1	sk_produto	=	sk_produto	
2	sk_cliente	=	sk_cliente	
3	sk_funcionario	=	sk_funcionario	
4	sk_data	=	sk_data	

Update fields:

#	Table field	Stream field	Update
1	qtdUnidadeVendas	qtdUnidadeVendas	Y
2	virVendas	virVendas	Y
3	datVendas	datVendas	Y

Buttons: Help, OK, Cancela, SQL

Desabilite o Hop que vai para o step Table output (FatoVendas) e rode duas vezes para ver se dá erro.



Chegamos ao final do nosso desafio que é fazer o processo de ETL no Pentaho.

Respostas Finais

Os alunos deverão desenvolver a prática e, depois, responder às questões objetivas.