

1) Depois de entender como funciona as **Slowly Changing Dimension**, você irá determinar no *Data Mart* da presidência que:

- As dimensões tempo e fábrica usarão auto incremento no banco de dados, sendo *Slowly Changing Dimension Tipo 1*
- As dimensões organizacional e cliente irão ser *Slowly Changing Dimension Tipo 2*
- A dimensão produto será *Slowly Changing Dimension Tipo 1*, mas com controle de ID feito pelo **Pentaho Data Integration**

2) Acesse o MySQL através do **HeidiSQL** e modifique o *Data Mart* da presidência. Apague o banco de dados **dmpresidencia**, crie-o novamente, e execute o seguinte script:

```
USE DMPRESIDENCIA;
```

```
CREATE TABLE Dim_Organizacional (  
    ID_Vendedor INT NOT NULL,  
    Cod_Vendedor NVARCHAR(50) NOT NULL,  
    Desc_Vendedor NVARCHAR(250) NOT NULL,  
    Cod_Gerente NVARCHAR(50) NOT NULL,  
    Desc_Gerente NVARCHAR(250) NOT NULL,  
    Cod_Diretor NVARCHAR(50) NOT NULL,  
    Desc_Diretor NVARCHAR(250) NOT NULL,  
    Versao INT NOT NULL,  
  
    DataInicial DATE NOT NULL,  
    DataFinal DATE NOT NULL,  
    PRIMARY KEY (ID_Vendedor)  
);
```

```
CREATE TABLE Dim_Produto (  
    ID_Produto INT NOT NULL,  
    Cod_Produto NVARCHAR(50) NOT NULL,  
    Desc_Produto NVARCHAR(250) NOT NULL,  
    Cod_Marca NVARCHAR(50) NOT NULL,  
    Atr_Tamanho NVARCHAR(250) NOT NULL,  
    Atr_Sabor NVARCHAR(250) NOT NULL,  
    Cod_Categoria NVARCHAR(50) NOT NULL
```

```
    Desc_Categoria NVARCHAR(250) NOT NULL,  
    PRIMARY KEY (ID_Produto)  
);
```

```
CREATE TABLE Dim_Cliente (  
    ID_Cliente INT NOT NULL,  
    Cod_Cliente NVARCHAR(50) NOT NULL,  
    Desc_Cliente NVARCHAR(250) NOT NULL,  
    Cod_Cidade NVARCHAR(50) NOT NULL,  
    Desc_Cidade NVARCHAR(250) NOT NULL,  
    Cod_Estado NVARCHAR(50) NOT NULL,  
    Desc_Estado NVARCHAR(250) NOT NULL,  
    Cod_Regiao NVARCHAR(50) NOT NULL,  
    Desc_Regiao NVARCHAR(250) NOT NULL,  
    Cod_Segmento NVARCHAR(50) NOT NULL,  
    Desc_Segmento NVARCHAR(250) NOT NULL,  
    Versao INT NOT NULL,  
    DataInicial DATE NOT NULL,  
    DataFinal DATE NOT NULL,  
    PRIMARY KEY (ID_Cliente)  
);
```

```
ALTER TABLE Dim_Cliente COMMENT 'Tabela da dimensÃ£o cliente';
```

```
CREATE TABLE Dim_Tempo (  
    ID_Tempo INT AUTO_INCREMENT NOT NULL,  
    Cod_Tempo NVARCHAR(50) NOT NULL,  
    Data DATE NOT NULL,  
    Numero_Dia_Semana NVARCHAR(50) NOT NULL,  
    Numero_Mes NVARCHAR(50) NOT NULL,  
    Numero_Ano NVARCHAR(50) NOT NULL,  
    Nome_Mes NVARCHAR(250) NOT NULL,  
    Numero_Trimestre NVARCHAR(50) NOT NULL,  
    Nome_Trimestre NVARCHAR(250) NOT NULL,  
    Numero_Semestre NVARCHAR(50) NOT NULL,  
    Nome_Semestre NVARCHAR(250) NOT NULL,  
    PRIMARY KEY (ID_Tempo)
```

```
);
```

```
CREATE TABLE Dim_Fabrica (  
    ID_Fabrica INT AUTO_INCREMENT NOT NULL,  
    Cod_Fabrica NVARCHAR(50) NOT NULL,  
    Desc_Fabrica NVARCHAR(250) NOT NULL,  
    PRIMARY KEY (ID_Fabrica)  
);
```

```
CREATE TABLE Fato_Presidencia (  
    ID_Tempo INT NOT NULL,  
    ID_Fabrica INT NOT NULL,  
    ID_Vendedor INT NOT NULL,  
    ID_Cliente INT NOT NULL,  
    ID_Produto INT NOT NULL,  
    Faturamento DOUBLE PRECISION NOT NULL,  
    Unidade_Vendida DOUBLE PRECISION NOT NULL,  
    Quantidade_Vendida DOUBLE PRECISION NOT NULL,  
    Imposto DOUBLE PRECISION NOT NULL,  
    Custo_Variavel DOUBLE PRECISION NOT NULL,  
    Custo_Frete DOUBLE PRECISION NOT NULL,  
    Custo_Fixo DOUBLE PRECISION NOT NULL,  
    Meta_Faturamento DOUBLE PRECISION NOT NULL,  
    Meta_Custo DOUBLE PRECISION NOT NULL,  
    PRIMARY KEY (ID_Tempo, ID_Fabrica, ID_Vendedor, ID_Cliente, ID_Produto)  
);
```

```
ALTER TABLE Fato_Presidencia ADD CONSTRAINT dim_organizacional_fato_  
FOREIGN KEY (ID_Vendedor)  
REFERENCES Dim_Organizacional (ID_Vendedor)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

```
ALTER TABLE Fato_Presidencia ADD CONSTRAINT produto_fato_presidencia_  
FOREIGN KEY (ID_Produto)  
REFERENCES Dim_Produto (ID_Produto)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

```
ALTER TABLE Fato_Presidencia ADD CONSTRAINT cliente_fato_presidencia_
FOREIGN KEY (ID_Cliente)
REFERENCES Dim_Cliente (ID_Cliente)
ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

```
ALTER TABLE Fato_Presidencia ADD CONSTRAINT tempo_fato_presidencia_fl
FOREIGN KEY (ID_Tempo)
REFERENCES Dim_Tempo (ID_Tempo)
ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

```
ALTER TABLE Fato_Presidencia ADD CONSTRAINT dim_fabrica_fato_presider
FOREIGN KEY (ID_Fabrica)
REFERENCES Dim_Fabrica (ID_Fabrica)
ON DELETE NO ACTION
ON UPDATE NO ACTION;
```

[COPIAR CÓDIGO](#)

3) Em **C:\treinamento\designer-tools\data-integration\simple-jndi**, abra o arquivo **jdbc.properties** e inclua os comandos de conexão:

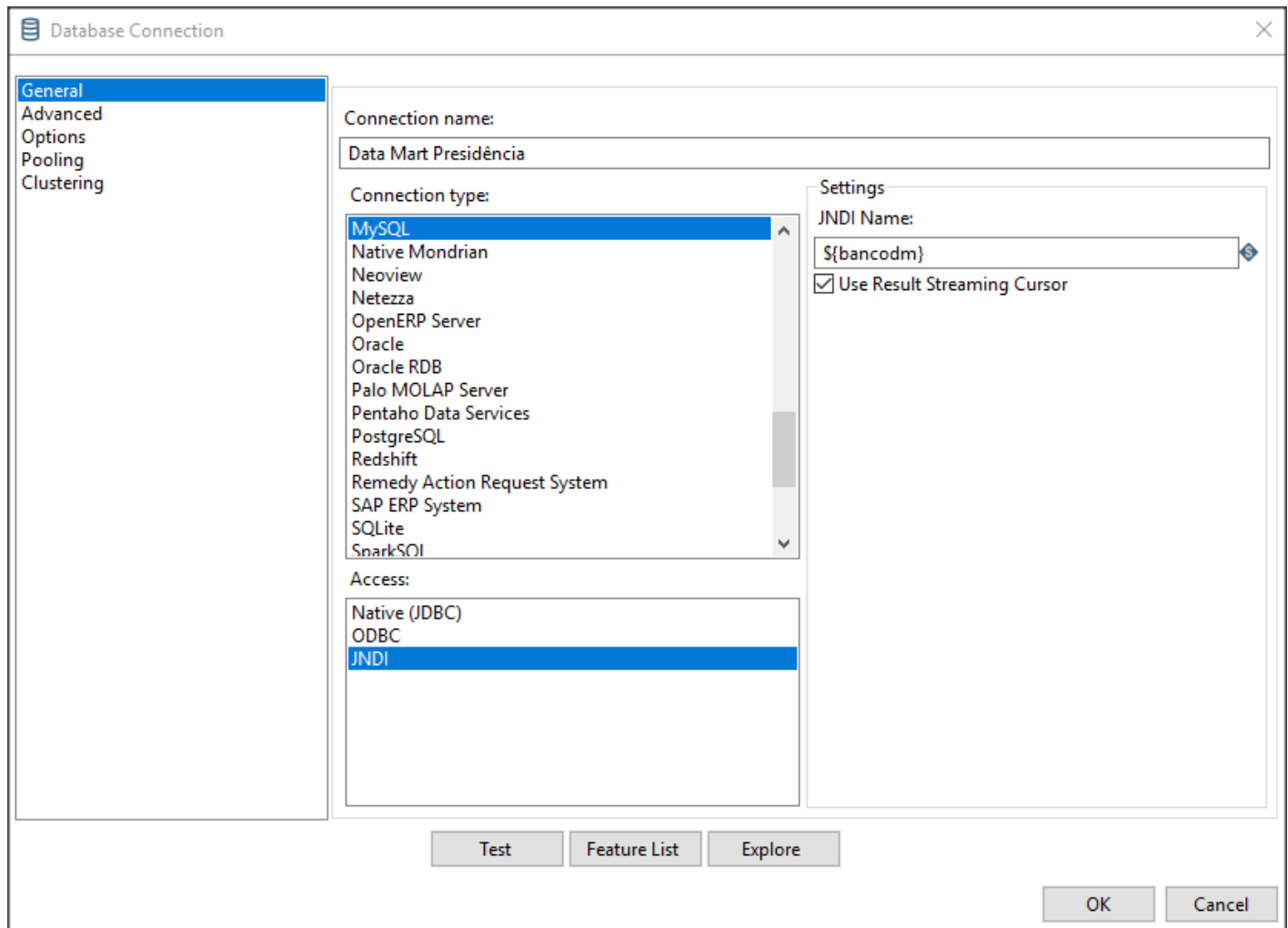
```
dmpresidencia/type=javax.sql.DataSource
dmpresidencia/driver=com.mysql.jdbc.Driver
dmpresidencia/url=jdbc:mysql://localhost:3306/dmpresidencia?useTimezo
dmpresidencia/user=root
dmpresidencia/password=root
```

[COPIAR CÓDIGO](#)

Observe as diferenças dos comandos acima com o do seu ambiente, como estar ou não no horário de verão e o usuário e senha do MySQL.

4) Acesse o **Pentaho Data Integration** no repositório **reposucos** e crie uma transformação chamada **DimensoesDataMart**.

5) Crie uma conexão, chamada **Data Mart Presidência**, com as seguintes propriedades:



6) Nesta transformação, acrescente um *step* do tipo **Table input**. Chame-o de **Dimensão Fábrica DW** e inclua as seguintes propriedades:

The screenshot shows the 'Table input' dialog box in Pentaho Data Integration. The 'Step name' field is 'Dimensão Fábrica DW'. The 'Connection' dropdown is set to 'Datawarehouse Sucos'. The 'SQL' text area contains the query 'SELECT * from Dim_Fabrica'. The 'Limit size' field is set to 0. The 'Insert data from step' dropdown is empty. The 'Execute for each row?' checkbox is unchecked. The 'Enable lazy conversion' and 'Replace variables in script?' checkboxes are also unchecked. The 'Get SQL select statement...' button is visible next to the SQL text area. The 'OK', 'Preview', and 'Cancel' buttons are at the bottom right, and the 'Help' button is at the bottom left.

Table input

Step name: Dimensão Fábrica DW

Connection: Datawarehouse Sucos

SQL: SELECT * from Dim_Fabrica

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step: [Empty dropdown]

Execute for each row? ☐

Limit size: 0

Buttons: Help, OK, Preview, Cancel

7) Adicione um *step* do tipo **Insert / update**. Chame-o de **Carga Dimensão Fábrica DM**, ligue o *step* **Dimensão Fábrica DW** a este e nas propriedades deste novo *step*, inclua:

Insert / update

Step name:

Connection:

Target schema:

Target table:

Commit size:

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2	
1	Cod_Fabrica	=	Cod_Fabrica		

Update fields:

#	Table field	Stream field	Update	
1	Cod_Fabrica	Cod_Fabrica	N	
2	Desc_Fabrica	Desc_Fabrica	Y	

8) Acrescente um *step* do tipo **Table input**. Chame-o de **Dimensão Tempo DW** e inclua as seguintes propriedades:

The screenshot shows the 'Table input' dialog box in Pentaho Data Integration. The 'Step name' field is 'Dimensão Tempo DW', and the 'Connection' is 'Datawarehouse Sucos'. The SQL query is 'SELECT * from Dim_Tempo'. The 'Limit size' is set to 0. The dialog box has a 'Get SQL select statement...' button and a 'Help' button.

Table input

Step name: Dimensão Tempo DW

Connection: Datawarehouse Sucos

SQL: SELECT * from Dim_Tempo

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size: 0

Buttons: Help, OK, Preview, Cancel

9) Adicione um *step* do tipo **Insert / update**. Chame-o de **Carga Dimensão Tempo DM**, ligue o *step* **Dimensão Tempo DW** a este e nas propriedades deste novo *step*, inclua:

Insert / update

Step name:

Connection:

Target schema:

Target table:

Commit size:

Don't perform any updates: ☐

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

#	Table field	Comparator	Stream field1	Stream field2
1	Cod_Tempo	=	Cod_Tempo	

Update fields:

#	Table field	Stream field	Update
1	Cod_Tempo	Cod_Tempo	N
2	Data	Data	Y
3	Nome_Mes	Nome_Mes	Y
4	Nome_Semestre	Nome_Semestre	Y
5	Nome_Trimestre	Nome_Trimestre	Y
6	Numero_Ano	Numero_Ano	Y
7	Numero_Dia_Semana	Numero_Dia_Semana	Y
8	Numero_Mes	Numero_Mes	Y
9	Numero_Semestre	Numero_Semestre	Y
10	Numero_Trimestre	Numero_Trimestre	Y

10) Acrescente um *step* do tipo **Table input**. Chame-o de **Dimensão Produto DW** e inclua as seguintes propriedades:

Table input

Step name:

Connection:

SQL

```
SELECT DIM_CATEGORIA.COD_CATEGORIA, DIM_CATEGORIA.DESC_CATEGORIA,
DIM_MARCA.COD_MARCA, DIM_MARCA.DESC_MARCA,
DIM_PRODUTO.COD_PRODUTO, DIM_PRODUTO.DESC_PRODUTO, DIM_PRODUTO.ATR_SABOR,
DIM_PRODUTO.ATR_TAMANHO FROM
DIM_PRODUTO
INNER JOIN DIM_MARCA ON DIM_PRODUTO.COD_MARCA = DIM_MARCA.COD_MARCA
INNER JOIN DIM_CATEGORIA ON DIM_PRODUTO.COD_CATEGORIA = DIM_CATEGORIA.COD_CATEGORIA;
```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size

11) Adicione um *step* do tipo **Combination lookup/update**. Chame-o de **Carga Dimensão Produto DM**, ligue o *step* **Dimensão Produto DW** a este e nas propriedades deste novo *step*, inclua:

Combination lookup/update

Step name: **Carga Dimensão Produto DM**

Connection: Data Mart Presidência [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: dim_produto [Browse...]

Commit size: 100 Cache size: 9999

Pre-load the cache? ☐

Key fields (to look up row in table):

#	Dimension field	Field in stream
1	Atr_Sabor	ATR_SABOR
2	Atr_Tamanho	ATR_TAMANHO
3	Cod_Categoria	COD_CATEGORIA
4	Cod_Marca	COD_MARCA
5	Cod_Produto	COD_PRODUTO
6	Desc_Categoria	DESC_CATEGORIA
7	Desc_Marca	DESC_MARCA
8	Desc_Produto	DESC_PRODUTO

Technical key field: ID_Produto

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence []

☐ Use auto increment field

Remove lookup fields? ☐

Use hashcode? ☐

Hashcode field in table

Date of last update field (optional) []

[?] Help OK Cancel Get Fields SQL

12) Acrescente um *step* do tipo **Table input**. Chame-o de **Dimensão Cliente DWe** inclua as seguintes propriedades:

Table input

Step name:

Connection:

SQL

```
SELECT * FROM DIM_CLIENTE
```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size

13) Adicione um *step* do tipo **Dimension lookup/update**. Chame-o de **Carga Dimensão Cliente DM**, ligue o *step* **Dimensão Cliente DW** a este e nas propriedades deste novo *step*, inclua:

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	Cod_Cliente	Cod_Cliente

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:

14) Acrescente um *step* do tipo **Table input**. Chame-o de **Dimensão Vendedor DW** e inclua as seguintes propriedades:

Table input

Step name:

Connection:

SQL

```
SELECT A.COD_ORGANIZACIONAL AS COD_VENDEDOR, A.DESC_ORGANIZACIONAL AS DESC_VENDEDOR,
B.COD_ORGANIZACIONAL AS COD_GERENTE, B.DESC_ORGANIZACIONAL AS DESC_GERENTE,
C.COD_ORGANIZACIONAL AS COD_DIRETOR, C.DESC_ORGANIZACIONAL AS DESC_DIRETOR
FROM DIM_ORGANIZACIONAL A
INNER JOIN DIM_ORGANIZACIONAL B ON A.COD_PAI = B.COD_ORGANIZACIONAL
INNER JOIN DIM_ORGANIZACIONAL C ON B.COD_PAI = C.COD_ORGANIZACIONAL
WHERE A.DIREITA - A.ESQUERDA = 1
UNION
SELECT A.COD_ORGANIZACIONAL AS COD_VENDEDOR, A.DESC_ORGANIZACIONAL AS DESC_VENDEDOR,
B.COD_ORGANIZACIONAL AS COD_GERENTE, B.DESC_ORGANIZACIONAL AS DESC_GERENTE,
B.COD_ORGANIZACIONAL AS COD_DIRETOR, B.DESC_ORGANIZACIONAL AS DESC_DIRETOR
FROM DIM_ORGANIZACIONAL A
INNER JOIN DIM_ORGANIZACIONAL B ON A.COD_PAI = B.COD_ORGANIZACIONAL
WHERE A.DIREITA - A.ESQUERDA = 1 AND B.COD_PAI = ''
```

Line 1 Column 0

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size

15) Adicione um *step* do tipo **Dimension lookup/update**. Chame-o de **Carga Dimensão Vendedor DM**, ligue o *step* **Dimensão Vendedor DW** a este e nas propriedades deste novo *step*, inclua:

Dimension lookup/update

Step name: Carga Dimensão Vendedor DM

Update the dimension? ☒

Connection: Data Mart Presidência Edit... New... Wizard...

Target schema: Browse...

Target table: dim_organizacional Browse...

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_Vendedor	COD_VENDEDOR

Technical key field: ID_Vendedor New name

Creation of technical key

☒ Use table maximum + 1

☐ Use sequence

☐ Use auto increment field

Version field: Versao

Stream Datefield:

Date range start field: DataInicial Min. year: 1900

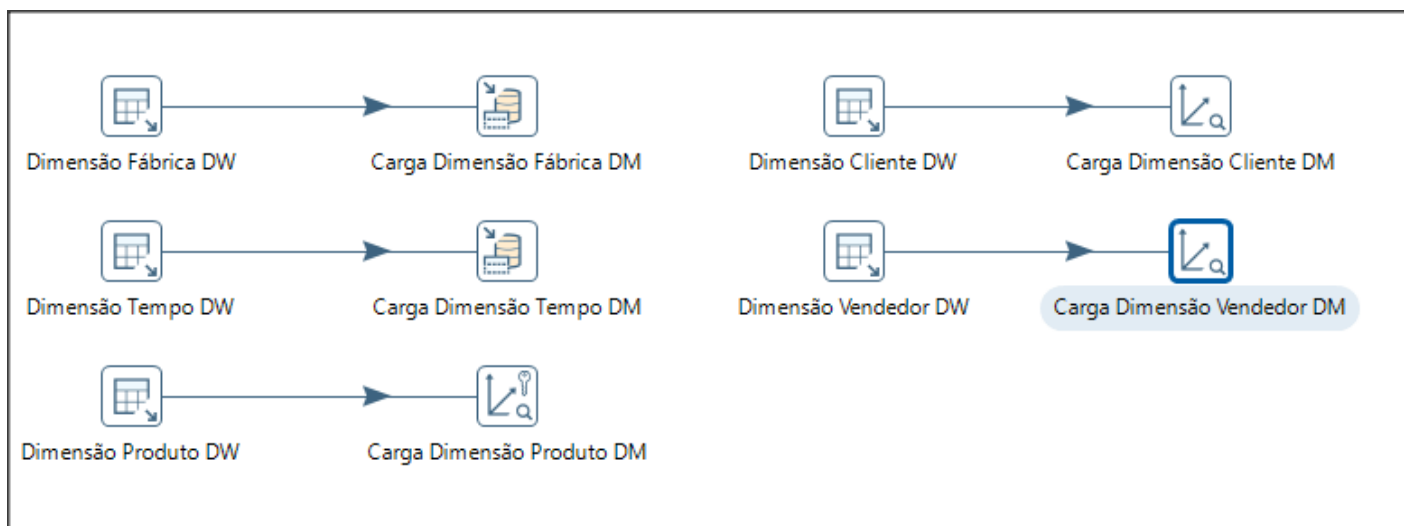
Use an alternative start date? ☐ <Select Option>

Table date range end: DataFinal Max. year: 2199

OK Cancel Get Fields SQL

Help

16) Você terá:



17) Salve e execute a transformação. Verifique no MySQL se as tabelas de dimensão foram carregadas.