

Relatório do Banco de Dados

purpleBoard

Goiânia

2019

Date Staging Area

A Date Staging Area foi formada por 5 fontes de dados distintas, sendo elas:

1. Fonte de dados em txt – Essa fonte de dados é a primeira a ser implementada, ela foi disponibilizada pelo professor André Luiz e desenvolvida em sala com o auxílio do mesmo.
2. Fonte de dados em PostgreSQL – Essa fonte de dados foi desenvolvida pelo aluno Carlos Eduardo Sousa Lopes, sendo previamente sorteada para o mesmo.
3. Fonte de dados em Access- Essa fonte de dados foi desenvolvida pelo aluno Marcelo Rangel, sendo previamente sorteada para o mesmo.
4. Fonte de dados em Access- Essa fonte de dados foi desenvolvida pelo aluno Maidno Lima, sendo previamente sorteada para o mesmo.
5. Fonte de dados em PostgreSQL – Essa fonte de dados tem origem em um projeto do período anterior, que basicamente é utilizado para realizar vendas. Ela foi utilizada com intuito de enriquecer os dados, para que nas consultas e gerações das tabelas e gráficos tenhamos dados mais variados por região, loja e tempo.

Tratamentos

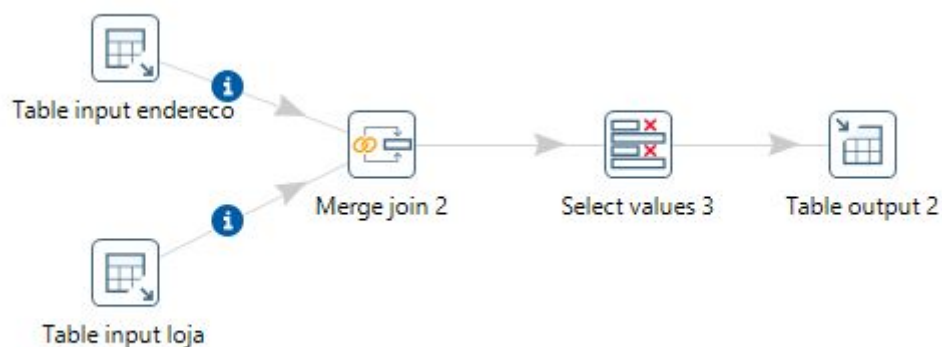
A maior parte dos tratamentos realizados nas fontes de dados é a renomeação dos campos de acordo com a Date Staging Area, aonde a maioria dessas renomeações é feita durante as transformações de ETL. Somente em alguns casos foi necessário realizar o tratamento, como:

- Fonte de dados em Access (Maidno Lima) – O número da nota fiscal estava como String , sendo composto pelas iniciais “nf” + uma sequência de números. O número foi alterado sendo removido as iniciais “nf” e mantendo somente a sequência de números.

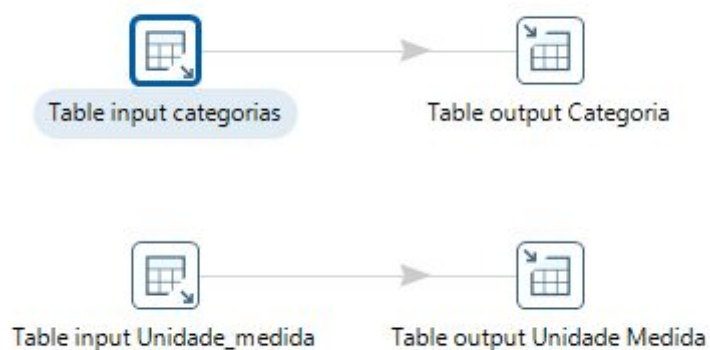
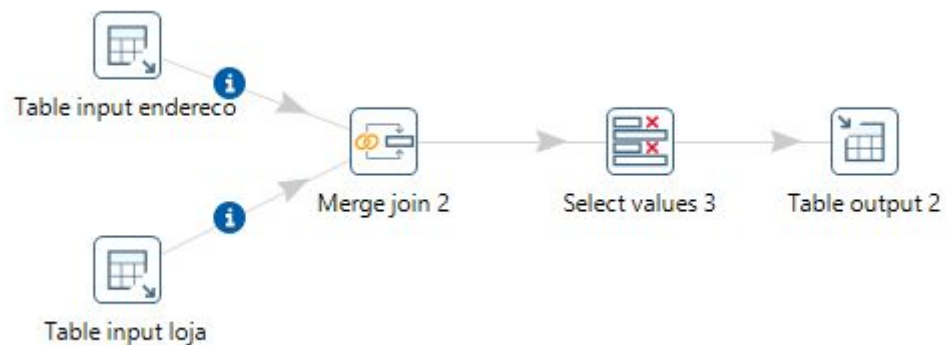
numero_nf ▾		numero_nf ▾
nf2000		2000
nf2001		2001
nf2002		2002
nf2003		2003
nf2004		2004
nf2005		2005
nf2006		2006
nf2007		2007
nf2008		2008
nf2009		2009
nf2010		2010
nf2011		2011
nf2012		2012
nf2013		2013
nf2014		2014

Transformações

1. Fonte de dados em PostgreSQL (Carlos Eduardo) para Date Staging Area
 - a) Transformações de Cliente: Na fonte de dados é utilizado uma tabela para armazenar os dados sobre cliente e uma tabela diferente para os dados de endereço do cliente. Inicia-se realizando um table input pegando os dados das duas tabelas e um merge join para unir as tabelas a partir do id_endereco . Depois finaliza com um select values selecionando os campos necessários e um table output carregando os dados na date staging área.

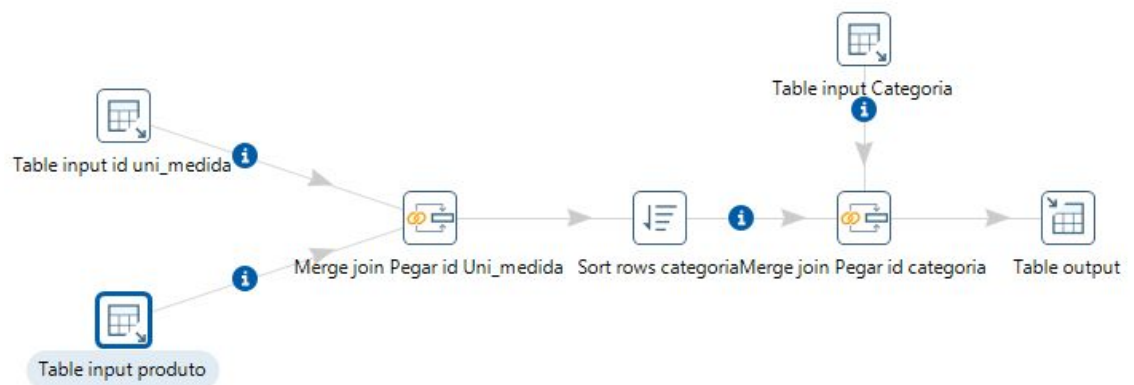


- b) Transformações de Loja: Na fonte de dados é utilizado uma tabela para armazenar os dados sobre loja e uma tabela diferente para os dados de endereço da loja. Inicia-se realizando um table input pegando os dados das duas tabelas e um merge join para unir as tabelas a partir do id_endereco . Depois finaliza com um select values selecionando os campos necessários e um table output carregando os dados na date staging área.



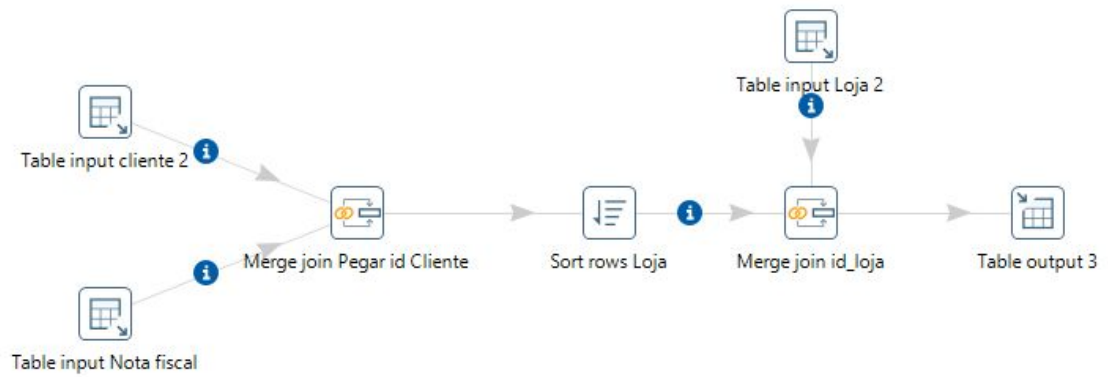
- c) Transformação Unidade_Medida e Categoria : Ambas transformações são bem simples, necessitando apenas renomear os campos conforme a date staging área. Inicia-se com um table input que utiliza no seu select a função “as” para renomear os campos e finaliza com um table output carregando os dados na date staging área.

d) Transformação de Produtos: Inicia-se com um table input que utiliza no seu select a função “as” para renomear os campos e a função “inner join” para pegar os campos necessários. Depois faz uso de dois merges join para buscar o respectivo id_unidade_medida e depois o

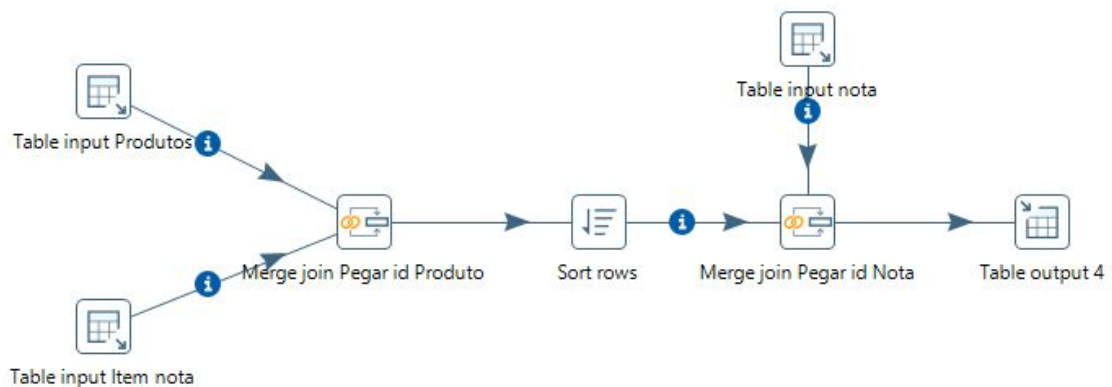


id_categoria e finaliza com um table output carregando os dados na date staging área.

e) Transformação de Nota Fiscal : Inicia-se com um table input que utiliza no seu select a função “as” para renomear os campos e a função “inner join” para pegar os campos necessários e a função order by para ordenar os valores para poder fazer o merger join. Depois faz uso de dois merges join para buscar o respectivo id_Cliente e depois o id_Loja , e finaliza com um table output carregando os dados na date staging área.



- f) Transformação Itens da Nota: Inicia-se com um table input que utiliza no seu select a função “as” para renomear os campos e a função “inner join” para pegar os campos necessários e a função order by para ordenar os valores para poder fazer o merger join. Depois faz uso de dois merges join para buscar o respectivo id_produto e depois o



id_nota_fiscal , e finaliza com um table output carregando os dados na date staging área.

2. Fonte de dados em Access(Marcelo Rangel e Maidno Lima)

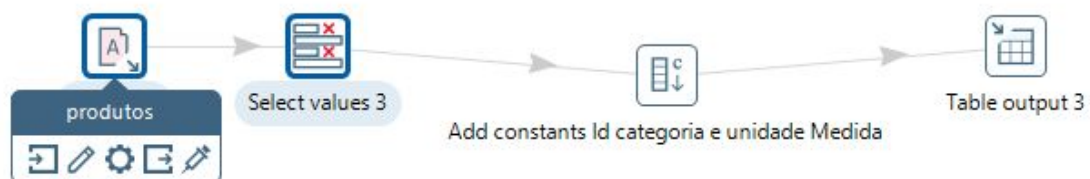
O processo de transformação em ambas fontes de dados é idêntica , precisando mudar somente o diretório de origem da fonte de dados.

- a) Transformação de Cliente e Loja : Transformação bastante simples , utiliza-se um access input para extrair os dados e depois um select values para selecionar e renomear os campos. Finaliza com um table output carregando os dados na data staging área.



- b) Transformação de Categoria e Unidade de Medida: Não foi necessário essas transformação pois em ambas fontes de dados só havia uma unidade de medida e uma categoria . Assim elas foram inseridas manualmente no banco de dados.

- c) Transformação de Produtos : Transformação bastante simples , utiliza-se um access input para extrair os dados e depois um select values para selecionar e renomear os campos. Depois utiliza-se um add



constant para adicionar os campos id_categoria e id_loja . Finaliza com um table output carregando os dados na data staging area.

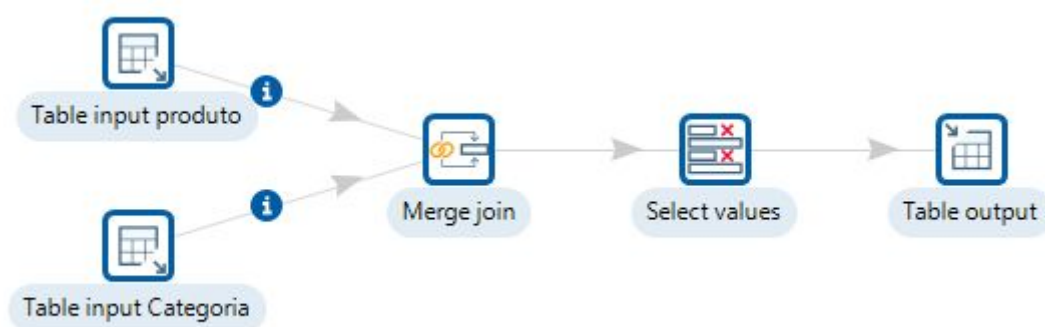
- d) Transformação de Nota Fiscal e Itens da Nota ; Ambas transformações foram complicadas principalmente por conta de falta de conhecimento

com a ferramenta pentaho e access. O grande problema estava em realizar um access input que pega-se os campos em tabelas diferentes através da chave estrangeira id. Como alternativa eu utilizei a própria função merge join do pentaho para fazer essa relação e unir as tabelas , depois foi necessário fazer novos merges join para relacionar os valores vindos do access input com seus respectivos valores já presentes no date staging área para obter os campos ids respectivos de cada tabela. Lembrando também que foi necessário realizar antes de cada merger join a função sort rows ou a order by no próprio table input para ordenar os valores.

Data Warehouse

O processo de transformação da maioria das tabelas do data warehouse foi bem tranquilo, levando em consideração que todos os dados já estavam prontos e tratados. Somente a tabela de fatos que demandou maior atenção.

1. Transformação de Cliente e Loja: Transformações simples necessitando somente do table input com os campos necessários e um table output para carregar os dados.
2. Transformação de Produtos: Nessa transformação foi necessário dois table input, um para pegar os dados de categoria e o outro para pegar os dados de produtos, sendo que nos selects foi utilizado order by para ordenar os campos. Depois foi utilizado um merge join para unir os dados e um select value para selecionar os dados e renomear lós conforme o data warehouse. E por fim um table input para carregar os dados no data warehouse.



3. Transformação de Tempo: Nessa transformação eu utilizei o select abaixo para obter os campos dia, mês e ano individualmente e a função when para gerar os campos bimestre, trimestre e semestre. Depois foi necessário um select values para passar o tipo e tamanho desses novos campos e um unique rows para o campo data_venda. E por fim um table output para carregar os dados no data warehouse.

```
select data_venda,
       EXTRACT(day from data_venda) as dia,
       EXTRACT(Month from data_venda) as mes,
       EXTRACT(year from data_venda) as ano,
       case when EXTRACT(Month from data_venda)=1 then 1
            when EXTRACT(Month from data_venda)=2 then 1
            when EXTRACT(Month from data_venda)=3 then 2
            when EXTRACT(Month from data_venda)=4 then 2
            when EXTRACT(Month from data_venda)=5 then 3
            when EXTRACT(Month from data_venda)=6 then 3
            when EXTRACT(Month from data_venda)=7 then 4
            when EXTRACT(Month from data_venda)=8 then 4
            when EXTRACT(Month from data_venda)=9 then 5
            when EXTRACT(Month from data_venda)=10 then 5
            when EXTRACT(Month from data_venda)=11 then 6
            when EXTRACT(Month from data_venda)=12 then 6
       end as bimestre,
       case when EXTRACT(Month from data_venda)=1 then 1
            when EXTRACT(Month from data_venda)=2 then 1
            when EXTRACT(Month from data_venda)=3 then 1
            when EXTRACT(Month from data_venda)=4 then 2
            when EXTRACT(Month from data_venda)=5 then 2
            when EXTRACT(Month from data_venda)=6 then 2
            when EXTRACT(Month from data_venda)=7 then 3
            when EXTRACT(Month from data_venda)=8 then 3
            when EXTRACT(Month from data_venda)=9 then 3
            when EXTRACT(Month from data_venda)=10 then 4
            when EXTRACT(Month from data_venda)=11 then 4
            when EXTRACT(Month from data_venda)=12 then 4
       end as trimestre,
       case when EXTRACT(Month from data_venda)=1 then 1
            when EXTRACT(Month from data_venda)=2 then 1
            when EXTRACT(Month from data_venda)=3 then 1
            when EXTRACT(Month from data_venda)=4 then 1
            when EXTRACT(Month from data_venda)=5 then 1
            when EXTRACT(Month from data_venda)=6 then 1
            when EXTRACT(Month from data_venda)=7 then 2
            when EXTRACT(Month from data_venda)=8 then 2
            when EXTRACT(Month from data_venda)=9 then 2
            when EXTRACT(Month from data_venda)=10 then 2
            when EXTRACT(Month from data_venda)=11 then 2
            when EXTRACT(Month from data_venda)=12 then 2
       end as semestre
from nota_fiscal order by data_venda
```

4. Transformação de Localidade : Essa transformação segue a mesma logica da anterior, foi utilizado no select do table input a função when para gerar o campo região. E depois é utilizado um unique rows para o campo cidade, finalizando com um table output para carregar os dados na data warehouse.

```
select cidade, estado,
       case when estado='AM' then 'Norte'
            when estado='RR' then 'Norte'
            when estado='AP' then 'Norte'
            when estado='PA' then 'Norte'
            when estado='TO' then 'Norte'
            when estado='RO' then 'Norte'
            when estado='AC' then 'Norte'

            when estado='MA' then 'Nordeste'
            when estado='PI' then 'Nordeste'
```

5. Transformação de Vendas(Tabela de Fatos) : Essa transformação tem início com table input aonde em seu select reúne dados de todas as outras tabelas através da função inner join, esse select também possui a função sum na quantidade e valor vendas para gerar os respectivos campos na tabela de fatos. Depois segue por diversos sort rows e merge joins para pegar todos os respectivos ids(id_produto, id_localidade, id_tempo, id_cliente, id_loja). Finaliza com table output carregando todos os dados na data stage área.

