Especificação Funcional – Projeto Pocco

Controle de versões

Autor	Versão	Data	Descrição
Priscila Ribeiro	1.0	10-06-2021	Criação do documento

Introdução

Este documento visa detalhar as necessidades do projeto migração de dados da empresa Pocco do ponto de vista técnico, bem como listar as possíveis soluções, suas premissas e atividades executadas durante o projeto.

Solicitação

O cliente deseja ter uma visão relacional dos dados gerados pelas vendas da Pocco e armazenados em Blob Storage, de forma simples e recorrente, através de processos de ETL e da criação de Data Warehouse em banco de dados MySQL, para possibilitar a construção de relatórios dinâmicos por meio do PowerBI.

Premissas da Solução

O fluxo de migração dos dados está dividido por camadas, com a finalidade de manter uma cópia, como são na origem e viabilizar formas performáticas de atualização.

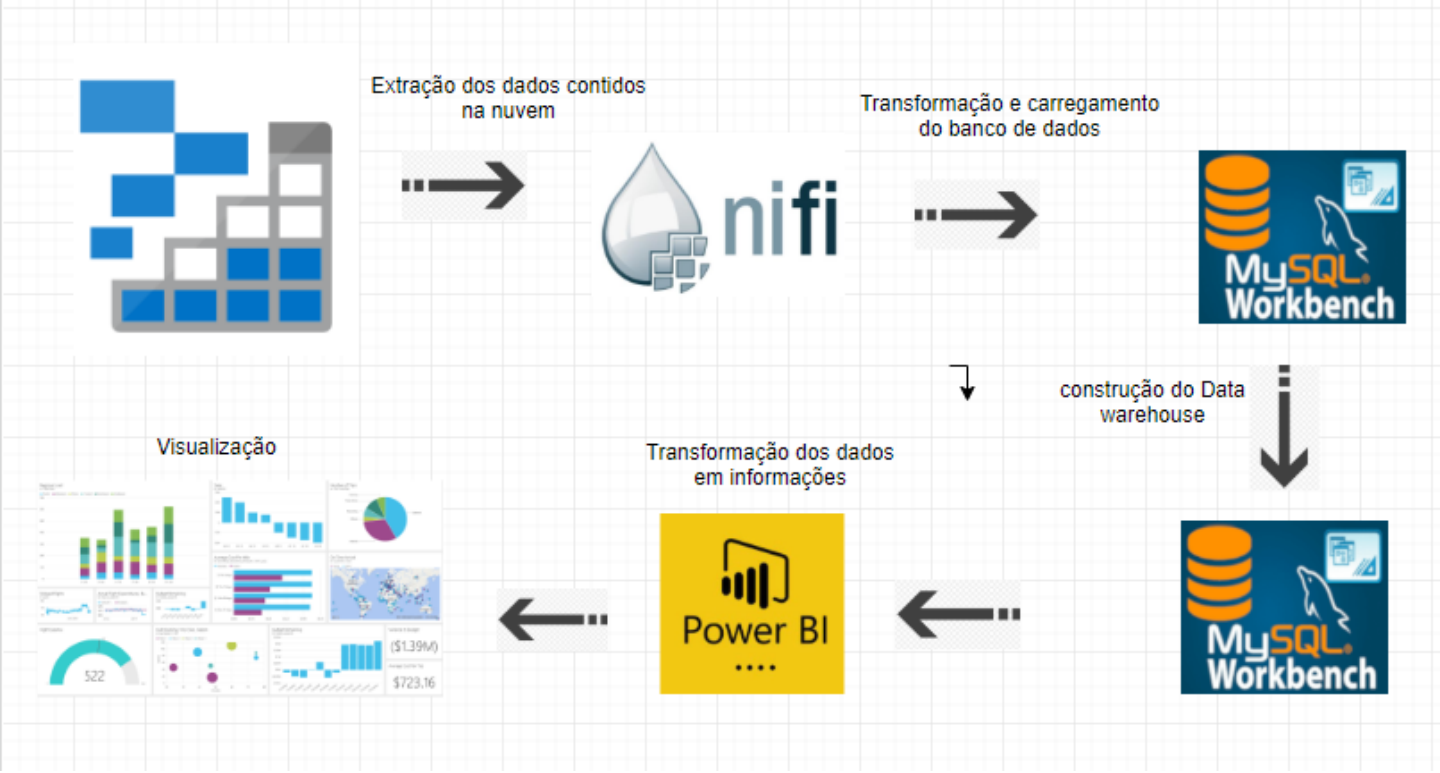
Azure Blob Storage: Local de origem/nuvem onde os dados da empresa Pocco estão disponibilizados, em arquivo formato CSV. O mesmo está localizado no container “nifi001”. Neste container ficará salvo o arquivo original, o qual será utilizado apenas para extração dos dados para manipulações futuras sem alterar o arquivo base.

Apache NiFi: Software utilizado para fazer todo o processo de ETL dos dados (extração dos dados da nuvem, tratamento e carregamento desses dados no Banco de dados MySQL).

MySQL Workbench: SGBD utilizado para armazenar e organizar os dados em formato de tabelas fato e dimensões, que servirão para relacionar esses dados de maneira que se transformem em informações referentes às vendas. Neste banco será criada a base de dados **`orders`** e a tabela `stage` para receber todos os dados originais conforme arquivo no Blob, também serão criadas as tabelas dimensões **`dim_country`**, `**dim_region**`, `**dim_saleschannel**` e tabela fato `**fato_vendas**` que através da Store Procedure `**return_orders**` serão carregadas com os dados já transformados para a construção desta Data Warehouse.

PowerBI: Através dele criaremos o dashboard com 4 relatórios, onde será possível visualizar de forma recorrente as informações de vendas solicitadas pela empresa.

Desenho do Fluxo de migração

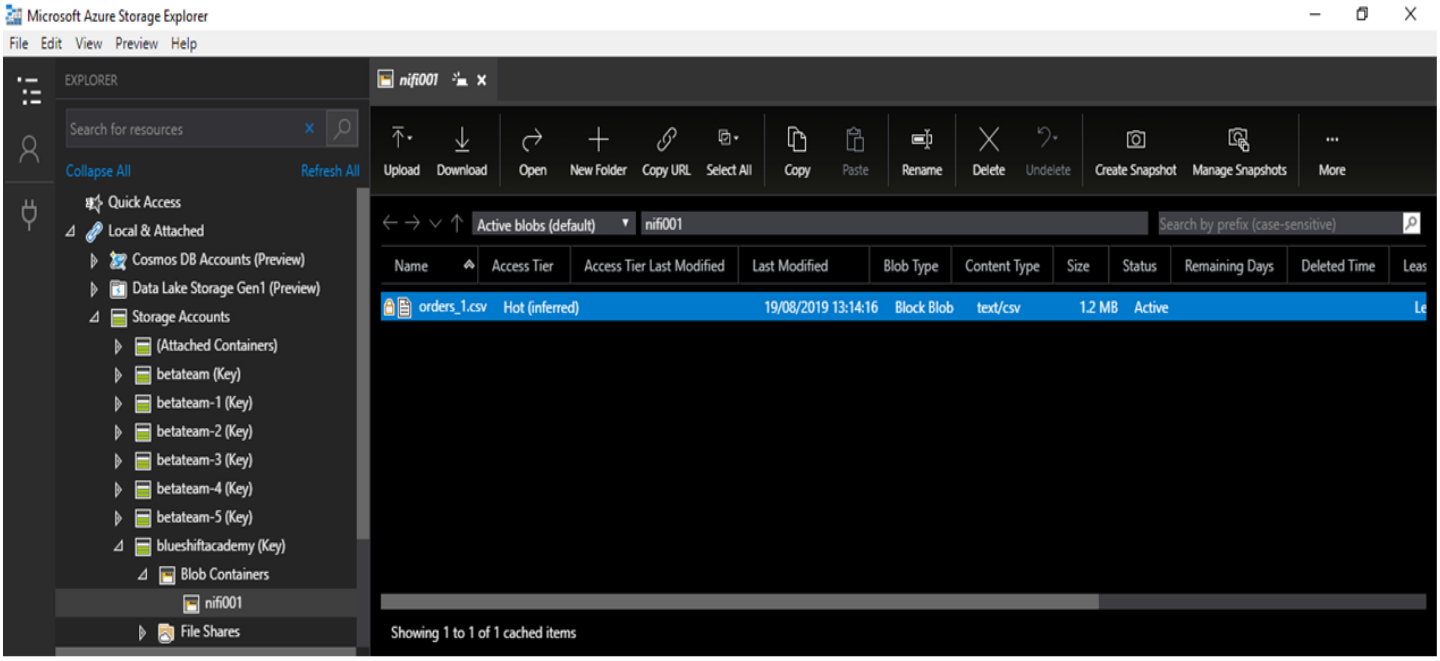


Fonte dos Dados

A camada de dados inicial estará localizada no Azure Blob Storage, onde será definido um container para o processo. Este container receberá os dados originais sem manipulação.

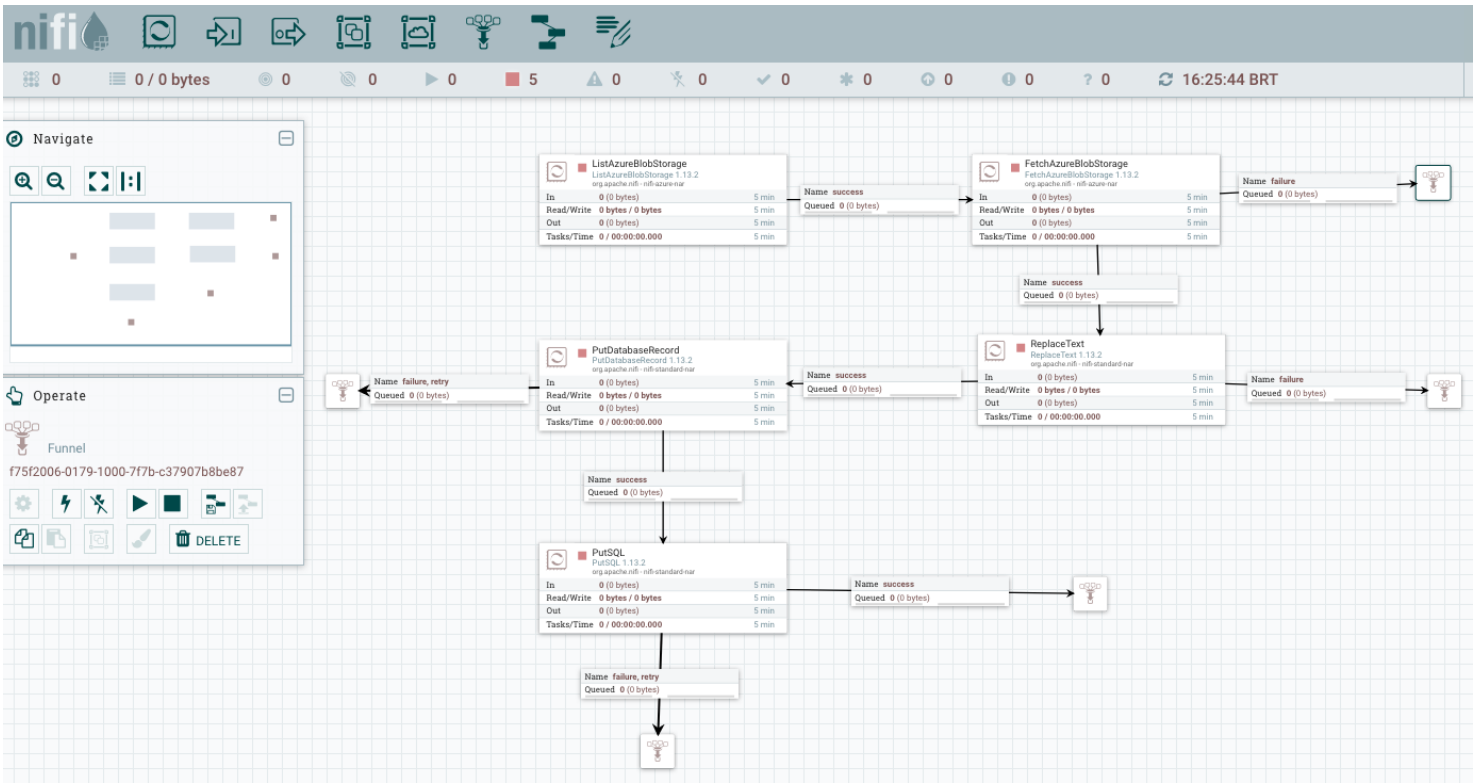
Container: nifi001

File: orders_1.csv



Processo ETL

Todo o processo de extração, transformação e carregamento dos dados será feito pelo Apache NiFi, conforme o fluxo de processors da imagem abaixo. Na sequência, teremos o detalhamento de cada um deles.



Detalhamento dos processors

ListAzureBlobStorage + FetchAzureBlobStorage: são os processors responsáveis por criar a conexão com o Blob Storage, e juntos fazem a extração dos dados contidos no arquivo CSV.

ReplaceText: Responsável por atualizar e substituir os valores de acordo com o conteúdo contido no arquivo fonte antes dos dados serem carregados no banco de dados.

PutDatabaseRecord: Este processor fará a conexão com o banco de dados e converterá o arquivo CSV para SQL, então fará o carregamento (INSERT) dos dados brutos na tabela **stage**.

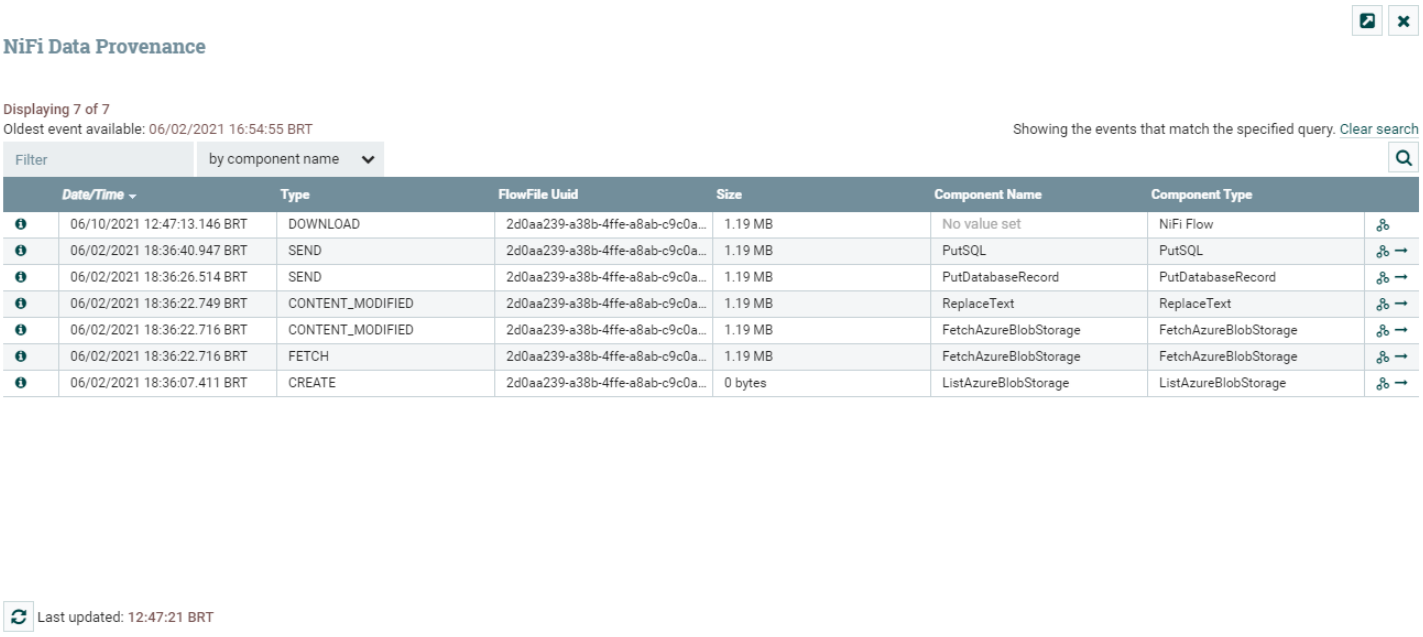
PutSQL: Este processor fará a chamada (CALL) da Store Procedure, que preencherá automaticamente as tabelas dimensões e fato, onde teremos os dados com as tipagens corretas para manipulação e criação dos relatórios de vendas.

Automação do processo

Através do NiFi, serão utilizadas estratégias de automação de preenchimento de tabelas do banco de dados, de maneira que o sistema apresente o melhor e mais rápido desempenho.

Ao configurar cada processor do fluxo, faremos com que haja insert a cada 10 segundos caso o “play” esteja acionado. Embutida na configuração desses processors estão os comandos SQL, em forma de script, que irão executar os insert’s.

Para alimentar as tabelas dimensões e fato, será utilizada a chamada da procedure no processor PutSQL, que permitirá otimizar o processo de atualização do fluxo, garantindo que o processo de popular as tabelas só se dê após os registros já existentes sejam deletados pelo script do TRUNCATE TABLE, garantindo que não haja informações duplicadas a cada inserção de dados na tabela stage.



NiFi Data Provenance

Displaying 7 of 7
Oldest event available: 06/02/2021 16:54:55 BRT

Showing the events that match the specified query. Clear search

Filter | by component name

Date/Time	Type	FlowFile Uuid	Size	Component Name	Component Type
06/10/2021 12:47:13.146 BRT	DOWNLOAD	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	No value set	NiFi Flow
06/02/2021 18:36:40.947 BRT	SEND	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	PutSQL	PutSQL
06/02/2021 18:36:26.514 BRT	SEND	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	PutDatabaseRecord	PutDatabaseRecord
06/02/2021 18:36:22.749 BRT	CONTENT_MODIFIED	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	ReplaceText	ReplaceText
06/02/2021 18:36:22.716 BRT	CONTENT_MODIFIED	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	FetchAzureBlobStorage	FetchAzureBlobStorage
06/02/2021 18:36:22.716 BRT	FETCH	2d0aa239-a38b-4ffe-a8ab-c9c0a...	1.19 MB	FetchAzureBlobStorage	FetchAzureBlobStorage
06/02/2021 18:36:07.411 BRT	CREATE	2d0aa239-a38b-4ffe-a8ab-c9c0a...	0 bytes	ListAzureBlobStorage	ListAzureBlobStorage

Last updated: 12:47:21 BRT

Base de dados

A tabela **stage** será construída no schema **orders,** com os mesmos campos contidos no arquivo fonte, seguindo a mesma ordem de colunas:

Region	varchar(40)
Country	varchar(40)
Item_Type	varchar(40)
Sales_Channel	varchar(40)
Order_Priority	varchar(40)
Order_Date	varchar(40)
Order_ID	varchar(40)
Ship_Date	varchar(40)
Units_Sold	varchar(40)
Unit_Price	varchar(40)
Unit_Cost	varchar(40)
Total_Revenue	varchar(40)
Total_Cost	varchar(40)
Total_Profit	varchar(40)

Na sequência, criaremos a Data Warehouse com as tabelas fato e dimensões.

- As tabelas dimensões serão compostas por: **dim_country**, **dim_region**, **dim_saleschannel.** Cada uma conterá o próprio tipo de dado e um identificador numérico por onde faremos seus relacionamentos.

dim_country (185 registros)

id_country	int(11)
Country	varchar(40)

dim_region (7 registros)

id_region	int(11)
Region	varchar(40)

dim_saleschannel (2 registros)

id_channel	int(11)
Sales_Channel	varchar(40)

- A Tabela **fato_vendas** será criada com base nos relacionamentos das tabelas dimensões através de seus identificadores, e com a tipagem correta de todos os dados. As colunas em vermelho representam as chaves estrangeiras.

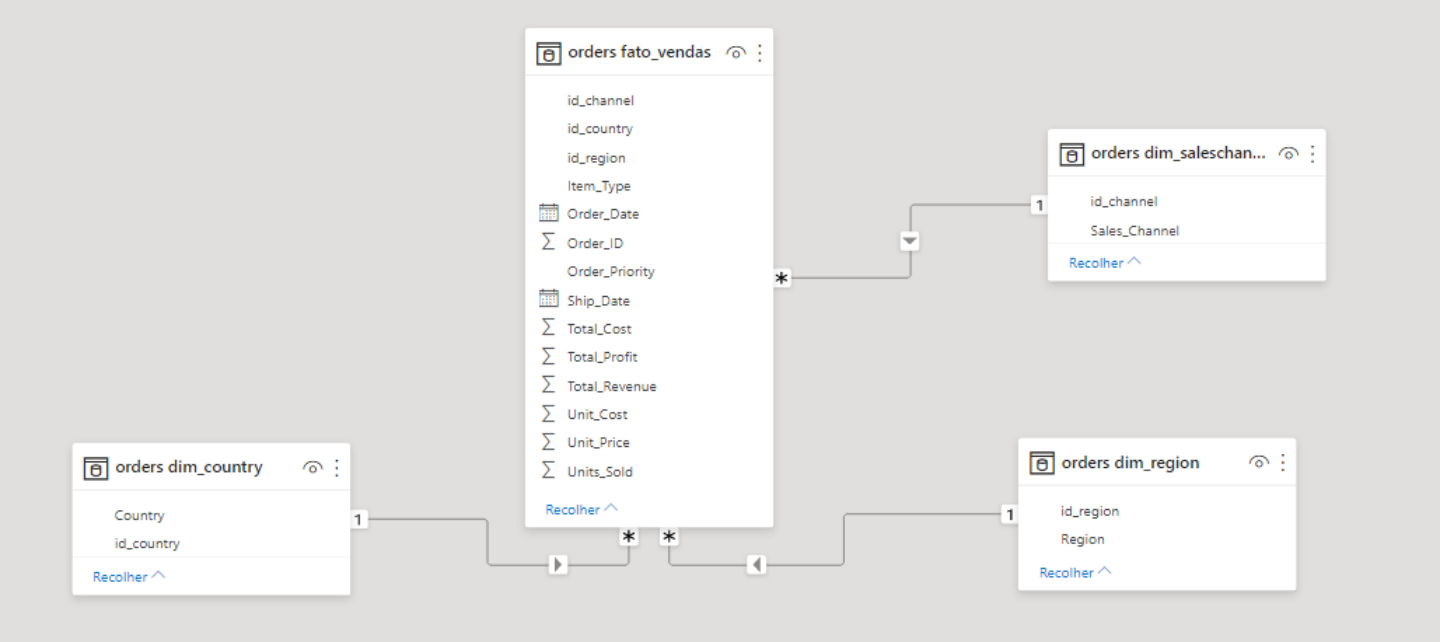
id_region	int(11)
id_country	int(11)
Item_Type	varchar(40)
id_ channel	int(11)
Order_Priority	varchar(40)
Order_Date	date
Order_ID	int(11)
Ship_Date	date
Units_Sold	decimal(10,2)
Unit_Price	decimal(10,2)
Unit_Cost	decimal(10,2)
Total_Revenue	decimal(10,2)
Total_Cost	decimal(10,2)
Total_Profit	decimal(10,2)

Store Procedure

Ainda no MySQL, criaremos a procedure **return_orders**, que quando feita a chamada no Nifi irá popular as tabelas da Data Warehouse e dar truncate na tabela **stage** para não ocorrer duplicidade de dados.

```
CREATE PROCEDURE `return_orders`()
BEGIN

truncate table fato_vendas;
truncate table dim_region;
truncate table dim_country ;
truncate table dim_saleschannel;

insert into dim_region (Region)
select distinct Region from stage
where stage.Order_ID NOT IN(select stage.Order_ID from stage
inner join fato_vendas on stage.Order_ID = fato_vendas.Order_ID);

insert into dim_country (Country)
select distinct Country from stage
where stage.Order_ID not in (select stage.Order_ID from stage
inner join fato_vendas on stage.Order_ID = fato_vendas.Order_ID);

insert into dim_saleschannel (Sales_Channel)
select distinct Sales_Channel from stage
where stage.Order_ID not in (select stage.Order_ID from stage
inner join fato_vendas on stage.Order_ID = fato_vendas.Order_ID);

insert into fato_vendas
(id_region ,id_country, Item_Type, id_channel,Order_Priority, Order_Date ,Order_ID ,Ship_Date ,Units_Sold ,Unit_Price,
Unit_Cost,Total_Revenue,Total_Cost, Total_Profit)
select dim_region.id_region, dim_country.id_country, Item_Type, dim_saleschannel.id_channel, Order_Priority,
str_to_date(order_date,'%c/%e/%Y'), Order_ID,
str_to_date(Ship_Date,'%c/%e/%Y'),
cast(Units_Sold as decimal (10,2 )),
cast(Unit_Price as decimal (10,2)),
cast(Unit_Cost as decimal (10,2)),
cast(Total_Revenue as decimal (10,2)),
cast(Total_Cost as decimal (10,2)),
cast(Total_Profit as decimal (10,2))

from stage
join dim_region on dim_region.Region = stage.Region
join dim_country on dim_country.Country = stage.Country
join dim_saleschannel on dim_saleschannel.Sales_Channel = stage.Sales_Channel
where stage.Order_ID not in(select stage.Order_ID from stage join fato_vendas on stage.Order_ID =
fato_vendas.order_id);

truncate table stage;
END
```

Modelo entidade-relacionamento



Dashboards

Utilizaremos o PowerBI para relacionar os dados das vendas e transformá-los nas informações solicitadas pela Pocco.

Teremos 4 relatórios, abaixo seguem suas finalidades e o relacionamento das informações:

- **Total de vendas e quantidade acumulada por dia no último mês (entre 06/2017 e 07/2017):** representado por gráfico de linhas, onde são calculados através da relação entre as colunas order_date, Total_Revenue e Units_Sold da tabela fato_vendas, e aplicados os filtros de ano e mês.

- **Vendas por região e país no último ano**: representado em mapa, demonstrados a partir do relacionamento da coluna Country da tabela dim_country, a coluna Region da tabela dim_region e a coluna Total_Revenue da tabela fato_vendas, e aplicado filtro de ano (2017).

- **Acumulado de vendas por canal e país do último ano**: demonstrado em gráfico de barras, a parti do relacionamento da coluna Country da tabela dim_country, a coluna Sales_Channel da tabela dim_saleschannel, a coluna Total_Revenue e Order_Date da tabela fato_vendas, aplicando filtro de ano (2017).

- **Vendas nos últimos 10 dias:** visualizado em gráfico de barras, relacionando as colunas Order_Date e Units_Sold da tabela fato_vendas, aplicando filtro de dia/mês/ano para retornar apenas o período solicitado.

Como filtragem extra, é possível ver no dashboard de modo dinâmico o total de unidades vendidas, e o filtro por categoria dos itens, uma vez que selecionado ou clicando em cima de qualquer dado dos gráficos.

Visualização estática do dashboard