Projeto Pocco

Aprovadores:

Este documento requer as assinaturas dos seguintes aprovadores:

Nome	**Cargo/Função**	**Assinatura**

Histórico de revisões

Versão	**Data**	**Autor**	**Descrição de mudança**
1.0	**25/10/2019**	**DENISE PROENÇA**	

Visão geral do projeto.

Este projeto tem por objetivo proporcionar uma visão relacional dos dados gerados pelas vendas da empresa Pocco, de forma simples e recorrente, através de processos de ETL e da criação de um Data Warehouse possibilitar a fácil construção de relatórios OLAP e uma visualização das vendas de maneira mais dinâmica por meio do PowerBI.

Processamento de dados.

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

- 1) Azure Blob Storage: Local de origem/nuvem onde nós precisaremos que sejam disponibilizados os dados da empresa Pocco em um arquivo com formato csv, o qual ficará contido no container nifi001. Aqui ficará a salvo o verdadeiro arquivo e será manipulado para as análises apenas cópias do mesmo.

- 2) Apache Nifi: É o software que será utilizado para fazer a extração dos dados da nuvem, o tratamento, a conexão e o carregamento desses dados no banco de dados.

- 3) Mysql Workbech: SGBD que será 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 serão criadas duas bases de dados: uma para receber todos os dados (nifi001 – tabela orders) e outra para a construção do Data

warehouse(dw_orders). Nesta última serão armazenadas as tabelas fato e dimensões.

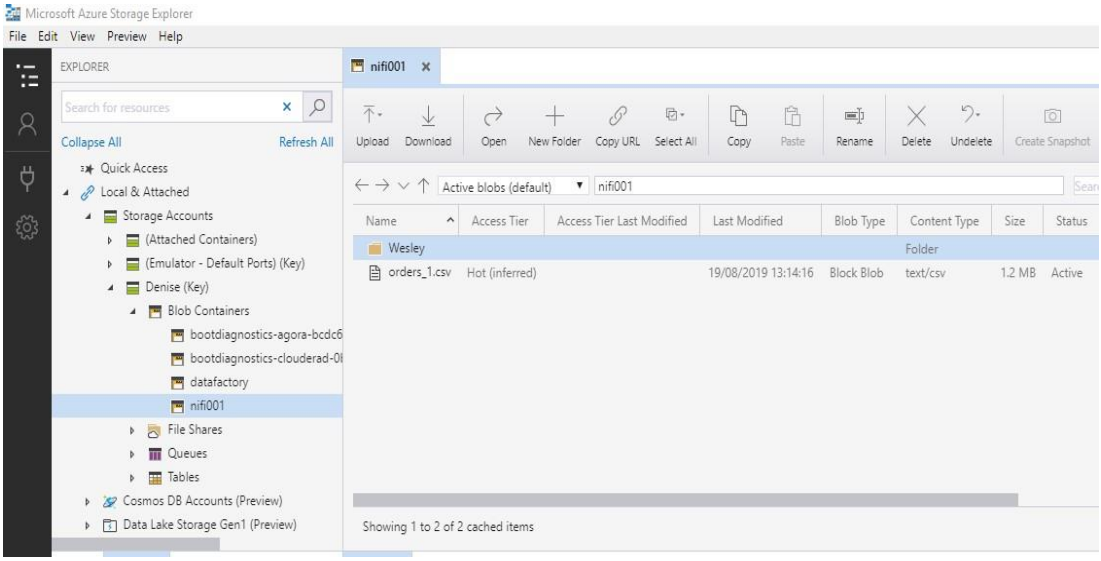
- 4)PowerBi: Através dele relacionaremos os dados de maneira que seja possível a visualização das informações em forma de gráficos.

Fonte dos dados

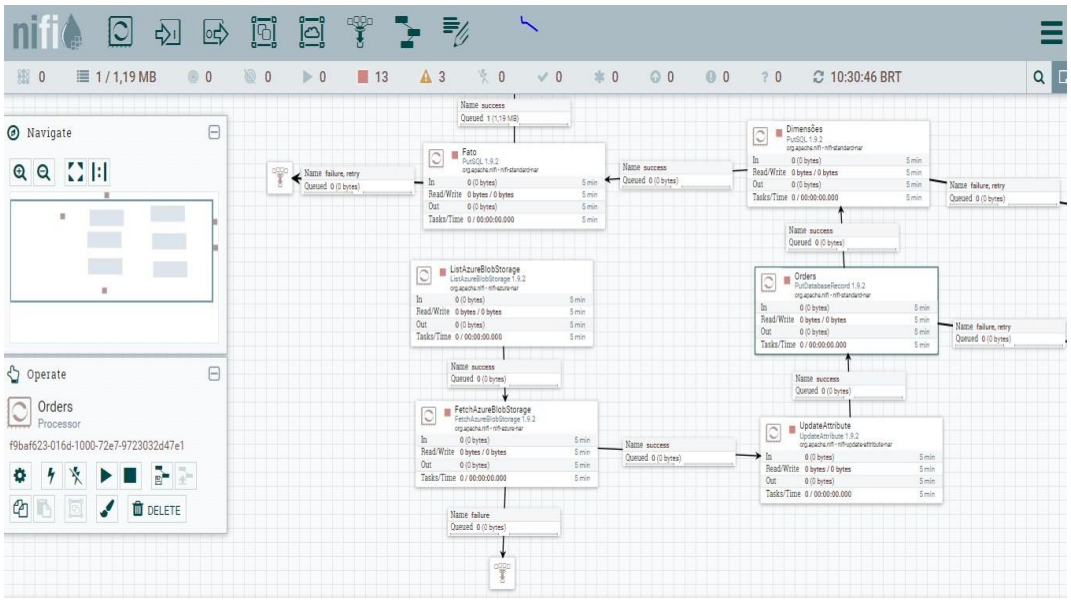
A camada de dados inicial estará localizada no Azure onde será definido um container para o processo. Esse container receberá os dados origens sem manipulação.

Container: nifi001

File: orders_1.csv



O Processo ETL



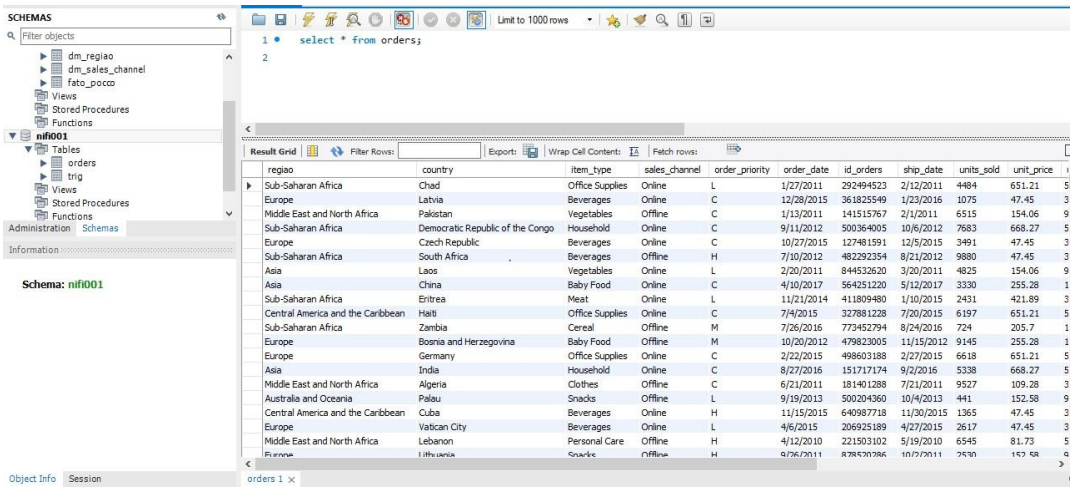
- ListAzureBlobStorage + FetchAzureBlobStorage são os processors que, juntos, fazem a extração dos dados da nuvem.
- UpdateAtribute será usado para que os dados sejam filtrados e atualizados antes de serem convertidos para sql.
- PutDataBaseRecord é o processor que converterá csv para SQL e carregará a tebela orders com esses dados “brutos”.

  Uma vez que a tabela orders estiver carregada, criaremos uma outra tabela que receberá uma mensagem de sucesso, em caso de sucesso, que acionará trigegrs e preencherá automaticamente as tabelas dimensões. O nome da tabela que receberá essa mensagem será trig e o processor responsável por essa automatização será do tipo PutSql.
- Após populadas as tabelas dimensões, utilizaremos no Nifi outro processor para preencher de maneira automática a tabela fato.

Onde serão relacionados de forma quantitativa os dados das vendas.

Bases de dados

- A tabela orders será construída, no schema nifi001, com os mesmos campos contidos no arquivo fonte, em forma de colunas e na mesma ordem.



Colunas da Orders:

regiao varchar(255)
country varchar(255)
item_type varchar(255)
sales_channel varchar(255)
order_priority varchar(255)
order_date varchar(255)
id_orders int(11)
ship_date varchar(255)
units_sold double
unit_price double
unit_cost double
total_revenue double
total_cost double
total_profit double
times_temp timestamp

Antes de criar a tabela trig, criaremos o Data Warehouse com as tabelas fato e dimensões.

- As tabelas dimensões serão: dm_country, dm_item_type, dm_order_priority, dm_regiao e dm_sales_channel. Cada uma conterá o próprio tipo de dado e um identificador numérico pelo qual será referenciado esse dado. Por exemplo: A tabela dm_regiao conterá uma coluna região que receberá o nome de regiões e uma outra coluna com um identificador numérico, assim, quando quisermos referenciar uma região na tabela fato podemos apenas usar esse número identificador.
- A tabela trig será composta por apenas uma coluna chamada msg, que receberá a mensagem de sucesso. Após criadas e preenchidas as tabelas dimensões, criaremos os triggers da tabela trig. Lembrando que são os triggers que tornam algum processo automático, no nosso caso ele fará o insert automático nas tabelas dimensões.
- Tabela fato: nessa tabela relacionaremos os dados de maneira numérica, sendo que cada dado não numérico terá o seu identificador como chave estrangeira de alguma tabela dimensão.

Suas colunas serão: as colunas em negrito representam as chaves estrangeiras e identificadores dos dados não numéricos.

id_regiao	int(11)
id_item_type	int(11)
id_sales_channel	int(11)
id_order_priority	int(11)
order_date	date
id_orders	int(11)
ship_date	date
units_sold	double
unit_price	double
unit_cost	double
total_revenue	double
total_cost	double
total_profit	double
id_country	int(11)
times_temp	timestamp

Transformando dados em informações

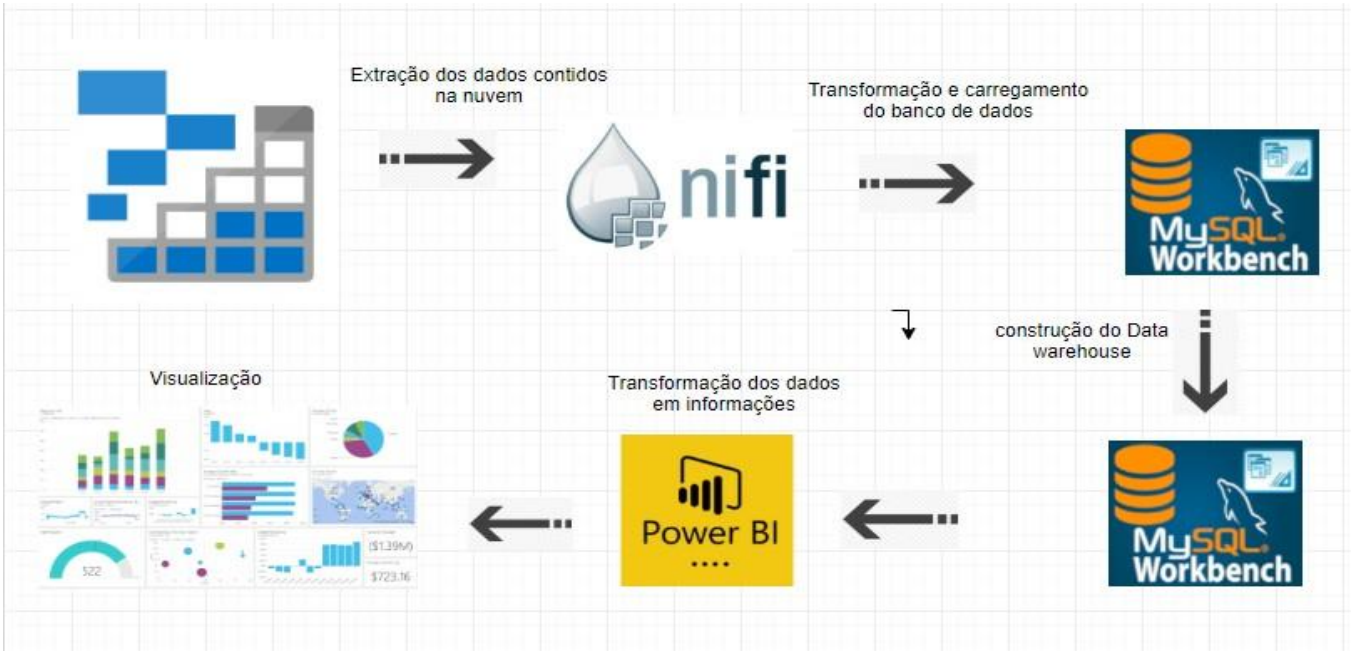
Nós utilizaremos o PowerBi para relacionar os dados das vendas e transformá-los nas informações que a empresa Pocco nos solicitou. Para isso, segue o relacionamento de das informações e as suas finalidades:

- Conseguiremos encontrar o acumulado de vendas por país e região no último ano (entre janeiro de 2017 e 28 de julho de 2017) se relacionarmos a coluna country da tabela dm_country, a coluna regiao da tabela dm_regiao e a coluna total_revenue da tabela fato. Nessa coluna total_revenue estão contidos os valores referentes à receita total.
- Para obter a quantidade de vendas nos últimos 10 dias (entre 18/07 a 28/07/2017) nós relacionaremos as colunas id_orders e order_date. Sendo que id_orders significa quantidade de vendas e order_date todas as datas referentes a cada venda.
- A quantidade e o acumulado de vendas no último mês (entre 28/6 e 28/7 de 2017) serão calculados através da relação entre id_orders, order_date e uma função do powerBi que calcula o total acumulado.

- O acumulado de vendas por canal e país será demonstrado relacionando a coluna country da dm_country, a coluna sales_channel da dm_sales_channel, a coluna total_revenue e a order_date que pertencem a tabela fato.

Todas essas relações entre colunas de dados gerarão gráficos que não necessariamente podem ter seus tipos modificados, porque existem certas relações que tem seu tipo de gráfico exato para uma visualização coerente.

Fluxo de migração.



Automação do processo

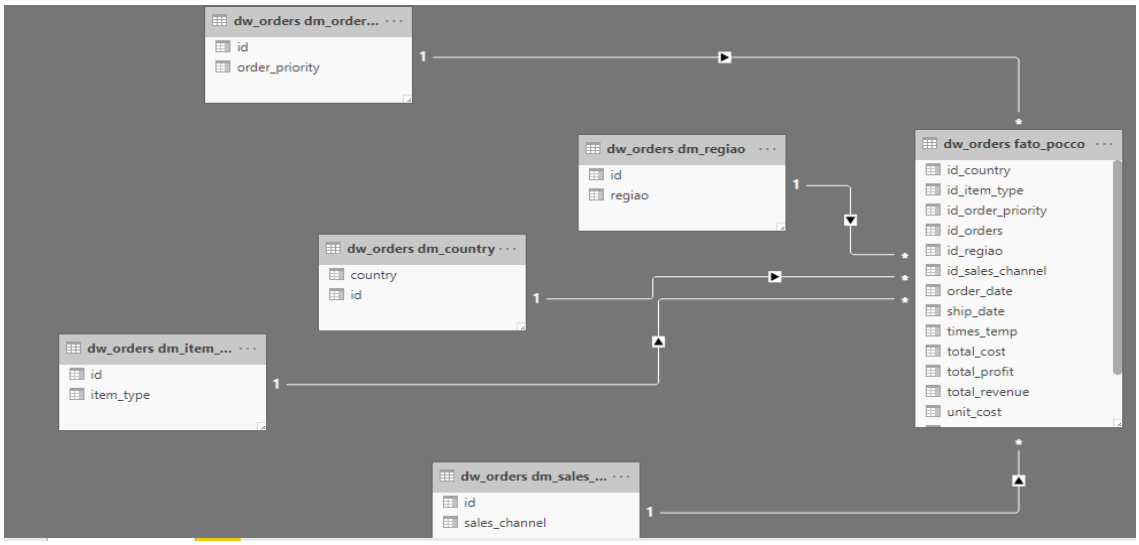
Através do Nifi serão utilizadas estratégias de automação de preenchimento de tabelas do banco de dados, de maneira a qual o sistema apresente o melhor e mais rápido desempenho.

Ao configurar cada processor do fluxo, faremos com que aja insert/update 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, serão utilizadas triggers que serão acionados em uma tabela chamada trig, após uma mensagem programada em caso de sucesso na população da tabela orders. Fazer dessa forma permitirá otimizar o processo de atualização do fluxo, visto que o uso de

triggers ao mesmo tempo da população da tabela orders deixará o sistema extremamente lento. Assim, com os triggers na tabela trig, garantiremos que o processo de popular as tabelas dimensões só se iniciará depois que a orders estiver populada.

Modelo entidade-relacionamento



Visualização: pbix