DOCUMENTAÇÃO:

Empresa: Super Store

- Sistema ETL :
  - Extract (Extrair)
  - Transform (Transformar)
  - Load (Carregar)

Com **tabelas de fatos** e **dimensões.**

1. Ferramentas, linguagens e insumos

a. Ferramentas e/ou plataformas:
   i. Google BigQuery
   ii. Planilhas Google ou Python e Google Colab

b. Linguagens:
   i. SQL no BigQuery e
   ii. Python no Google Colab (se você escolher usá-lo neste projeto).

c. Insumos:
   i. Tabela: superstore.csv

d. Criar:
   i. Projeto
   ii. Dataset
   iii. Importar tabela para o BigQuery

2. Marco obrigatório

a. 🟦 2.1 Processar e preparar base de dados

i. 2.1.1 🔵 Identificar e tratar valores nulos:

1. Nesse primeiro momento apenas identificar;
2.

```
# APENAS IDENTIFICAR VALORES NULOS

SELECT
    COUNTIF(category IS NULL) AS nulos_category,
    COUNTIF(city IS NULL) AS nulos_city,
    COUNTIF(country IS NULL) AS nulos_country,
    COUNTIF(customer_id IS NULL) AS nulos_customer_id,
```

```
    COUNTIF(customer_name IS NULL) AS
nulos_customer_name,
    COUNTIF(discount IS NULL) AS nulos_discount,
    COUNTIF(market IS NULL) AS nulos_market,
    COUNTIF(unknown IS NULL) AS nulos_unknown,
    COUNTIF(order_date IS NULL) AS nulos_order_date,
    COUNTIF(order_id IS NULL) AS nulos_order_id,
    COUNTIF(order_priority IS NULL) AS
nulos_order_priority,
    COUNTIF(product_id IS NULL) AS nulos_product_id,
    COUNTIF(product_name IS NULL) AS
nulos_product_name,
    COUNTIF(profit IS NULL) AS nulos_profit,
    COUNTIF(quantity IS NULL) AS nulos_quantity,
    COUNTIF(region IS NULL) AS nulos_region,
    COUNTIF(row_id IS NULL) AS nulos_row_id,
    COUNTIF(sales IS NULL) AS nulos_sales,
    COUNTIF(segment IS NULL) AS nulos_segment,
    COUNTIF(ship_date IS NULL) AS nulos_ship_date,
    COUNTIF(ship_mode IS NULL) AS nulos_ship_mode,
    COUNTIF(shipping_cost IS NULL) AS
nulos_shipping_cost,
    COUNTIF(state IS NULL) AS nulos_state,
    COUNTIF(sub_category IS NULL) AS
nulos_sub_category,
    COUNTIF(year IS NULL) AS nulos_year,
    COUNTIF(market2 IS NULL) AS nulos_market2,
    COUNTIF(weeknum IS NULL) AS nulos_weeknum
FROM
    `rota-super-store.superstore.superstore`;
```

ii. 2.1.2 🔵 Identificar e tratar valores duplicados

1. Nesse primeiro momento apenas identificar;
2.

```
# CENÁRIO 3: Visualizando os Registros Duplicados
Completos

SELECT
    *
FROM
    `rota-super-store.superstore.superstore`
QUALIFY ROW_NUMBER() OVER (PARTITION BY order_id,
```

```
product_id ORDER BY row_id) > 1;
```

3.

iii. 2.1.3 🔵 Identificar e tratar dados discrepantes em variáveis categóricas

1. identificar:
2.

```
# Identificando Inconsistências em Variáveis
Categóricas

# OK
SELECT DISTINCT category FROM
`rota-super-store.superstore.superstore` ORDER BY
category;

# RETORNOU MUITOS RESULTADOS, mas aparentemente OK
SELECT DISTINCT city FROM
`rota-super-store.superstore.superstore` ORDER BY
city;

# RETORNOU MUITOS RESULTADOS, mas aparentemente OK
SELECT DISTINCT country FROM
`rota-super-store.superstore.superstore` ORDER BY
country;

# (Geralmente IDs são consistentes, mas vale a pena
checar por formatação)
SELECT DISTINCT customer_id FROM
`rota-super-store.superstore.superstore` ORDER BY
customer_id;

# RETORNOU MUITOS RESULTADOS, mas aparentemente OK
SELECT DISTINCT customer_name FROM
`rota-super-store.superstore.superstore` ORDER BY
customer_name;

# ATENÇÃO
SELECT DISTINCT market FROM
`rota-super-store.superstore.superstore` ORDER BY
market;

#(Geralmente IDs são consistentes, mas vale a pena
```

```
checar por formatação)
SELECT DISTINCT order_id FROM
`rota-super-store.superstore.superstore` ORDER BY
order_id;

#OK
SELECT DISTINCT order_priority FROM
`rota-super-store.superstore.superstore` ORDER BY
order_priority;

#(Geralmente IDs são consistentes, mas vale a pena
checar por formatação)
SELECT DISTINCT product_id FROM
`rota-super-store.superstore.superstore` ORDER BY
product_id;

#ATENÇÃO
SELECT DISTINCT product_name FROM
`rota-super-store.superstore.superstore` ORDER BY
product_name;

#ATENÇÃO
SELECT DISTINCT region FROM
`rota-super-store.superstore.superstore` ORDER BY
region;

#OK
SELECT DISTINCT segment FROM
`rota-super-store.superstore.superstore` ORDER BY
segment;

#OK
SELECT DISTINCT ship_mode FROM
`rota-super-store.superstore.superstore` ORDER BY
ship_mode;

# APARENTEMENTE OK
SELECT DISTINCT state FROM
`rota-super-store.superstore.superstore` ORDER BY
state;

#OK
SELECT DISTINCT sub_category FROM
```

```
`rota-super-store.superstore.superstore` ORDER BY
sub_category;

#ATENÇÃO
SELECT DISTINCT market2 FROM
`rota-super-store.superstore.superstore` ORDER BY
market2;
```

3. Único tópico “tratado”, por ser recomendado para facilitar na criação das tabelas de dimensões e fatos.

4.

```
SELECT
    TRIM(UPPER(category)) AS category_cleaned,
    TRIM(UPPER(city)) AS city_cleaned,
    TRIM(UPPER(country)) AS country_cleaned,
    -- ... e assim por diante para todas as suas
colunas categóricas
    TRIM(UPPER(market)) AS market_cleaned,
    TRIM(UPPER(region)) AS region_cleaned,
    TRIM(UPPER(segment)) AS segment_cleaned,
    TRIM(UPPER(ship_mode)) AS ship_mode_cleaned,
    TRIM(UPPER(state)) AS state_cleaned,
    TRIM(UPPER(sub_category)) AS sub_category_cleaned,
    TRIM(UPPER(market2)) AS market2_cleaned
FROM
    `rota-super-store.superstore.superstore`;
```

iv. 2.1.4 🔵 Identificar e tratar dados discrepantes em variáveis numéricas

1. Foi feito a conferência e os dados que são números correspondem ao tipo correto, assim como os textos e datas.

2.

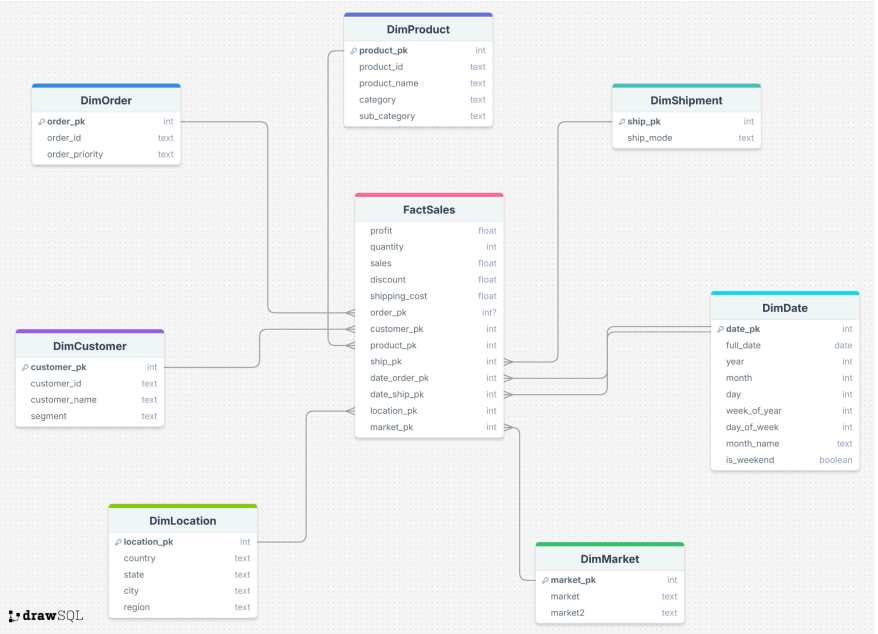
VARIÁVEL	TIPO
category	STRING
city	STRING
country	STRING
customer_id	STRING

customer_name	STRING
discount	FLOAT
market	STRING
unknown	INTEGER
order_date	TIMESTAMP
order_id	STRING
order_priority	STRING
product_id	STRING
product_name	STRING
profit	FLOAT
quantity	INTEGER
region	STRING
row_id	INTEGER
sales	INTEGER
segment	STRING
ship_date	TIMESTAMP
ship_mode	STRING
shipping_cost	FLOAT
state	STRING
sub_category	STRING
year	INTEGER
market2	STRING
weeknum	INTEGER

v. **2.1.5 🔵 Pesquisar dados de outras fontes:**

1. Foi usado o IMPORTHTML e Google Sheets;
2. https://docs.google.com/spreadsheets/d/1k7x10e43RqE3RSsFWFeFT7KzZ37kWOUPaadIrxI_Soc/edit?usp=sharing

vi. 2.1.6 🔵 **Projetar** estrutura de base de dados (tabelas de fatos e dimensões):

1. 

2. Link para o DrawSQL: https://drawsql.app/teams/eu-121/diagrams/super-store

3.

vii. 2.1.7 🔵 **Criar estrutura** de base de dados (tabelas de fatos e dimensões):

1. As novas tabelas estão listadas no dataset “superstore” no BigQuery. Elas estarão vazias neste momento, pois só foi criada a estrutura. O próximo passo seria o "L" (Load) do ETL, para preencher essas tabelas com os dados transformados.

2. Querys:

a.

```
#DimCustomer

CREATE TABLE
`rota-super-store.superstore.DimCustomer` (
    customer_pk INT64 OPTIONS(description="Chave primária da dimensão de cliente"),
    customer_id STRING OPTIONS(description="ID original do cliente"),
    customer_name STRING OPTIONS(description="Nome do cliente"),
    segment STRING OPTIONS(description="Segmento do cliente")
);
```

```
# DimDate

CREATE TABLE
`rota-super-store.superstore.DimDate` (
    date_pk INT64 OPTIONS(description="Chave primária da dimensão de data (formato YYYYMMDD)"),
    full_date DATE OPTIONS(description="Data completa"),
    year INT64 OPTIONS(description="Ano"),
    month INT64 OPTIONS(description="Número do mês (1-12)"),
    day INT64 OPTIONS(description="Número do dia do mês"),
    week_of_year INT64 OPTIONS(description="Número da semana no ano"),
    day_of_week INT64 OPTIONS(description="Dia da semana (1=Domingo, 7=Sábado)"),
    month_name STRING OPTIONS(description="Nome do mês"),
    quarter INT64 OPTIONS(description="Trimestre"),
    is_weekend BOOL OPTIONS(description="Indica se é fim de semana")
);
```

```
#DimLocation

CREATE TABLE
`rota-super-store.superstore.DimLocation` (
    location_pk INT64 OPTIONS(description="Chave primária da dimensão de localização"),
    country STRING OPTIONS(description="País"),
    state STRING OPTIONS(description="Estado"),
    city STRING OPTIONS(description="Cidade"),
    region STRING OPTIONS(description="Região")
);
```

```
# DimMarket

CREATE TABLE
`rota-super-store.superstore.DimMarket` (
    market_pk INT64 OPTIONS(description="Chave
```

```
primária da dimensão de mercado"),
    market STRING OPTIONS(description="Nome do
mercado"),
    market2 STRING
OPTIONS(description="Informação adicional de
mercado")
);
```

```
# DimOrder

CREATE TABLE
`rota-super-store.superstore.DimOrder` (
    order_pk INT64 OPTIONS(description="Chave
primária da dimensão de pedido"),
    order_id STRING OPTIONS(description="ID
original do pedido"),
    order_priority STRING
OPTIONS(description="Prioridade do pedido")
);
```

```
# DimProduct

CREATE TABLE
`rota-super-store.superstore.DimProduct` (
    product_pk INT64 OPTIONS(description="Chave
primária da dimensão de produto"),
    product_id STRING OPTIONS(description="ID
original do produto"),
    product_name STRING
OPTIONS(description="Nome do produto"),
    category STRING
OPTIONS(description="Categoria do produto"),
    sub_category STRING
OPTIONS(description="Subcategoria do produto")
);
```

```
#DimShipment

CREATE TABLE
`rota-super-store.superstore.DimShipment` (
    shipment_pk INT64 OPTIONS(description="Chave
```

```
primária da dimensão de envio"),
    ship_mode STRING OPTIONS(description="Modo
de envio")
);
```

```
# FactSales (FATOS)

CREATE TABLE
`rota-super-store.superstore.FactSales` (
    order_pk INT64 OPTIONS(description="Chave
estrangeira para DimOrder"),
    customer_pk INT64 OPTIONS(description="Chave
estrangeira para DimCustomer"),
    product_pk INT64 OPTIONS(description="Chave
estrangeira para DimProduct"),
    shipment_pk INT64 OPTIONS(description="Chave
estrangeira para DimShipment"),
    order_date_pk INT64
OPTIONS(description="Chave estrangeira para
DimDate (Data do Pedido)"),
    ship_date_pk INT64
OPTIONS(description="Chave estrangeira para
DimDate (Data de Envio)"),
    location_pk INT64 OPTIONS(description="Chave
estrangeira para DimLocation"),
    market_pk INT64 OPTIONS(description="Chave
estrangeira para DimMarket"),
    quantity INT64
OPTIONS(description="Quantidade de itens
vendidos"),
    sales FLOAT64 OPTIONS(description="Valor
total da venda"),
    profit FLOAT64 OPTIONS(description="Lucro
gerado"),
    discount FLOAT64
OPTIONS(description="Desconto aplicado"),
    shipping_cost FLOAT64
OPTIONS(description="Custo de envio")
);
```

viii. 2.1.8 🔵 Agendar atualizações de tabelas:

1. **Objetivo:** projetar o fluxo de atualização das tabelas criadas;

2.

Projetando o Fluxo de Atualização do Pipeline ETL (Ordem de Dependência):

🟣 O Fluxo Básico:

1. **Limpeza/Padronização da Fonte (Tabela Stage):** A primeira coisa a ser feita é preparar seus dados brutos.
2. **Carregamento das Dimensões:** Carregue os dados únicos para cada uma das suas tabelas de dimensão.
3. **Carregamento da Tabela de Fatos:** Por último, carregue os dados transacionais, referenciando as chaves primárias já existentes nas dimensões.

🟣Ordem Detalhada de Atualização:

Passo 1: Fonte de Dados Limpa e Padronizada (Staging)

- **Ação:** Crie ou atualize a tabela `rota-super-store.superstore.superstore_standardized`.
- **Dependências:** Nenhuma. Esta é a fonte inicial de dados para o seu Data Warehouse, onde você aplica as primeiras limpezas (como padronização de capitalização).
- **Comando Referência:**

```
CREATE OR REPLACE TABLE `rota-super-store.superstore.superstore_standardized` AS

SELECT

  TRIM(UPPER(category)) AS category,

  -- ... todas as outras colunas, com UPPER/TRIM onde aplicável

FROM

  `rota-super-store.superstore.superstore`;
```

Obs: Esta tabela serve como um "stage" (área de preparação) para o carregamento do seu DW.

Passo 2: Carregamento das Tabelas de Dimensão

Estas tabelas podem ser carregadas em paralelo (se a

ferramenta de pipeline permitir) ou em qualquer ordem, pois não dependem umas das outras. No entanto, é importante que **todas as dimensões estejam prontas** antes da `FactSales`.

- **Ação:** Preencha as tabelas de dimensão com os dados únicos da `superstore_standardized`, gerando suas chaves primárias (`_pk`).
- **Dependências:** Todas as dimensões dependem da `superstore_standardized`.
    1. **`DimCustomer`**
    2. **`DimProduct`**
    3. **`DimOrder`**
    4. **`DimShipment`**
    5. **`DimLocation`**
    6. **`DimMarket`** (Se você decidir utilizá-la)
    7. **`DimDate`**: Esta é um caso especial. A `DimDate` é geralmente populada uma única vez, de forma programática (gerando todas as datas para um período futuro), e depois atualizada apenas para adicionar novas datas conforme o tempo avança. Ela não depende dos dados transacionais da `superstore_standardized` para ser criada, apenas para ser populada com os valores relevantes de data.
        - *Exemplo:* Você pode criar uma `DimDate` para os próximos 10 ou 20 anos de uma vez

Passo 3: Carregamento da Tabela de Fatos

Esta é a última tabela a ser carregada, pois ela depende das chaves primárias de todas as dimensões associadas.

- **Ação:** Preencha a tabela `FactSales` com as métricas e, crucialmente, as **chaves estrangeiras (FKs)** das dimensões. Para isso, você fará `JOINs` entre a `superstore_standardized` e cada `Dimensão` para buscar as PKs correspondentes e inseri-las como FKs na `FactSales`.
- **Dependências:** Depende da `superstore_standardized` E de **todas** as tabelas de dimensão carregadas.
    1. **`FactSales`**
        - **Exemplo (Lógica para carregar a `FactSales`):** Para cada linha da `superstore_standardized`, você vai:
            - Pegar o `customer_id` dela e

procurar o `customer_pk` correspondente na `DimCustomer`.

- Pegar o `product_id` dela e procurar o `product_pk` correspondente na `DimProduct`.
- Pegar o `order_id` dela e procurar o `order_pk` correspondente na `DimOrder`.
- Pegar o `ship_mode` dela e procurar o `shipment_pk` correspondente na `DimShipment`.
- Pegar a `order_date` dela e procurar o `date_pk` correspondente na `DimDate` (para a coluna `order_date_pk`).
- Pegar a `ship_date` dela e procurar o `date_pk` correspondente na `DimDate` (para a coluna `ship_date_pk`).
- Pegar o `country`, `state`, `city`, `region` dela e procurar o `location_pk` correspondente na `DimLocation`.
- Pegar o `market` e `market2` dela e procurar o `market_pk` correspondente na `DimMarket`.
- E então, inserir todas as métricas (`sales`, `quantity`, `profit`, etc.) e essas FKs na `FactSales`.

🟣 Representação Visual do Fluxo

```
graph TD
    A[Dados Brutos rota-super-store.superstore.superstore] --> B(Limpeza e Padronização);
    B --> C{Criação/Atualização: rota-super-store.superstore.superstore_standardized};
```

```
C --> D1(Carregar DimCustomer);
C --> D2(Carregar DimProduct);
C --> D3(Carregar DimOrder);
C --> D4(Carregar DimShipment);
C --> D5(Carregar DimLocation);
C --> D6(Carregar DimMarket);
SubGraph DimDate_Gen
   D7_1[Gerar / Carregar DimDate]
End
D7_1 --> F(Carregar FactSales);

D1 --> F;
D2 --> F;
D3 --> F;
D4 --> F;
D5 --> F;
D6 --> F;
```

3.