

A task de ETL para consolidar e modelar dados de vendas B2B foi **concluída com sucesso**, resultando na criação de um **Modelo Dimensional Estrela** pronto para BI no MySQL (schema bd\_smv).

## 1. PROBLEMA (Contexto e Desafios Iniciais)

O desafio da task foi consolidar e transformar dados de vendas B2B (Sellout e Registro de Oportunidades) provenientes de duas fontes CSV em um Modelo Dimensional Estrela no MYSQL

### Dificuldades Encontradas Durante o Desenvolvimento:

- **Formato de Data Misto:** As datas nas bases de origem apresentavam-se em múltiplos formatos (serial numérico do Excel e string (DD/MM/YYYY), exigindo uma lógica robusta de conversão (parse\_date} na função transform\_data para unificar para {YYYY-MM-DD}.
- **Inconsistência de Nomenclatura de Colunas:** A coluna Branch uma das bases precisou ser explicitamente mapeada para Tier2\_PartnerName na etapa (Transsform) para manter a coerência do modelo.
- elo.
- **Manutenção de Integridade de Chaves:** Foi necessário criar chaves de negócios compostas (chave\_negocio) para garantir o **UPSERT** na Tabela Fato e evitar duplicação em execuções sucessivas do pipeline.

## 2. AÇÃO (Processo de ETL Desenvolvido)

Foi desenvolvido e executado um script Python modelar modular, focado nas fases de ETL e na aderência ao Modelo Estrela.

Fase	Ação Chave no Script	Detalhes Técnicos

<b>EXTRAÇÃO</b>	<code>extract_data()</code>	Consolidação de <code>FILE_MON</code> e <code>FILE_MOB</code> via <code>pd.concat()</code> . Inclusão de coluna <code>LFD</code> para alinhamento de esquemas.
<b>TRANSFORMAÇÃO</b>	<code>transform_data()</code>	Restrição a <code>SalesType</code> 'Sellout'/'Seltru' e <code>Type</code> em ['MON', 'LFD', 'MOBILE', 'TABLET', 'NOTE PC']. <b>Padronização:</b> CNPJs limpos e preenchidos com <code>zfill</code> (14). <b>Modelagem:</b> Criação de 5 <code>DataFrames</code> dimensionais ( <code>dim_data</code> , <code>drevenda</code> , etc.) e da Tabela Fato. <b>Chave de Negócio:</b> Criação de <code>chave_negocio</code> (PK para o <code>UPSERT</code> ).
<b>CARGA</b> (\$\text{LOAD}\$)	<code>load_data()</code>	Utilização de <code>sqlalchemy</code> e <code>mysql-connector-python</code> . Carregamento das dimensões (com <code>if_exists='append'</code> ). Execução de <b>lógica UPSERT</b> na Tabela Fato ( <code>fsellout</code> ) via Tabela Staging e comando <code>INSERT... ON DUPLICATE KEY UPDATE</code> com <code>text()</code> .

### 3. STATUS DA TASK (Atualização de Progresso)

Fase da Task	Progresso (%)	Observações
Extração, Limpeza e Padronização	100%	Dados unificados, CNPJs padronizados e datas corrigidas.

<b>Modelagem Dimensional (SQL+PYTHON)</b>	<b>100%</b>	Todas as 5 dimensões e a Tabela Fato (fsellout) foram criadas e as chaves primárias/estrangeiras definidas.
<b>Carregamento e Persistência (MYSQL+AZURE)</b>	<b>100%</b>	O processo de LOAD (incluindo o UPSERT para garantir idempotência) foi concluído com sucesso.
<b>Status Geral da Task</b>	<b>100%</b>	<b>TASK CONCLUÍDA.</b> A base está pronta para consumo em BI.

#### 4. Consultas Essenciais para o MySQL Workbench

O modelo Estrela permite análises rápidas no MySQL (bd\_smv)). Abaixo, três consultas essenciais:

##### A. Acompanhamento de Vendas (Sellout) por Mês e Produto

*Métrica:* QTY Total. *Dimensões:* Data e Produto.

SELECT

DD.Ano,

MONTH(DD.SalesDate) AS Mes\_Num,

DP.Type AS Tipo\_Produto,

SUM(FV.QTY) AS Quantidade\_Vendida

FROM

fsellout FV

JOIN

dim\_data DD ON FV.data\_key = DD.data\_key

JOIN

dim\_produto DP ON FV.produto\_key = DP.produto\_key

JOIN

dim\_tipo\_venda DTV ON FV.tipo\_venda\_key = DTV.tipo\_venda\_key

WHERE

DTV.SalesType = 'Sellout'

GROUP BY

DD.Ano, Mes\_Num, Tipo\_Produto

ORDER BY

DD.Ano, Mes\_Num;

## **B. Top 10 CNPJs Consolidados (drevenda) por Volume de Oportunidade**

*SELECT*

DR.cnpj\_revenda,

DR.tier2\_partner\_name, -- Nome do Parceiro Tier 2 ou do Cliente Final

SUM(FV.QTY) AS Volume\_Total\_Oportunidade

FROM

fsellout FV

JOIN

drevenda DR ON FV.id\_revenda = DR.drevenda\_key -- Assumindo que id\_revenda será preenchido na próxima fase

JOIN

dim\_tipo\_venda DTV ON FV.tipo\_venda\_key = DTV.tipo\_venda\_key

WHERE

DTV.SalesType = 'Seltru'

GROUP BY

```
        DR.cnpj_revenda, DR.tier2_partner_name

ORDER BY

        Volume_Total_Oportunidade DESC

LIMIT 10;
```

### C. Detalhamento de Transações (Itens de Fato) por Projeto

```
SET sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
```

-- Cole o código SQL corrigido aqui:

```
SELECT

        DD.Ano,

        MONTH(DD.SalesDate) AS Mes_Num,

        DP.Type AS Tipo_Produto,

        SUM(FV.QTY) AS Quantidade_Vendida

FROM

        fsellout FV

JOIN

        dim_data DD ON FV.data_key = DD.data_key

JOIN

        dim_produto DP ON FV.produto_key = DP.produto_key

JOIN

        dim_tipo_venda DTV ON FV.tipo_venda_key = DTV.tipo_venda_key

WHERE
```

DTV.SalesType = 'Sellout'

GROUP BY

DD.Ano, MONTH(DD.SalesDate), DP.Type

ORDER BY

DD.Ano ASC, Mes\_Num ASC;