Universidade Estadual de Londrina

Centro de Ciências Exatas
Departamento de Computação
Curso de Ciência da Computação
Tópicos em Computação

Relatório de construção do

Data Warehouse

Fernando Morgado Pires Neto
Gabriel Ângelo Perez Gasparini Sabaudo
Gabriela Tieko Hirashima
Guilherme Henrique Gonçalves Silva
Sophie Nascimento

Prof. Dr. Vitor Valério de Souza Campos

2022

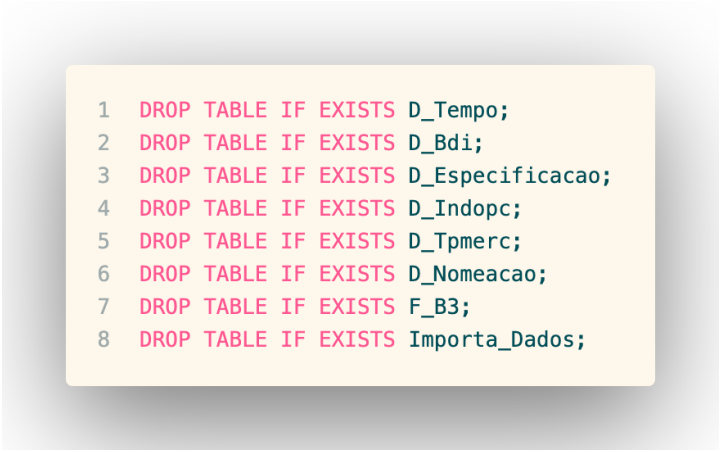
Introdução

O presente documento busca apresentar de forma simples as funções dos scripts executados para os trabalhos requisitados no segundo bimestre da matéria de Tópicos em Computação. Neste trabalho é feita a criação de um Data Warehouse, bem como as transformações no software Pentaho Data Integration. Nos Capítulos 1 e 2 são apresentados todos os scripts necessários para a criação da base de dados do Data Stage e Data Warehouse, respectivamente. No Capítulo 3, são explicadas todas as transformações referentes a todas as Dimensões, Fatos e demais. Para cada uma delas é dita sua função de forma simples e direta, seguido da sequência de todos os passos, descritos em detalhe. No Capítulo 4, é definida a criação do Data Warehouse, colocando de forma geral, como cada parte foi executada e o porquê de sua existência. Por fim, as dúvidas e dificuldades que foram levantadas durante a realização do trabalho estão presentes no penúltimo capítulo. O documento termina na conclusão que faz um apanhado do processo e descreve os frutos do trabalho.

Esquema do Banco de dados do DS

Abaixo estão os trechos de código em SQL relacionados com a construção do Data Storage em PostgreSQL.

Limpeza Inicial do Banco de Dados

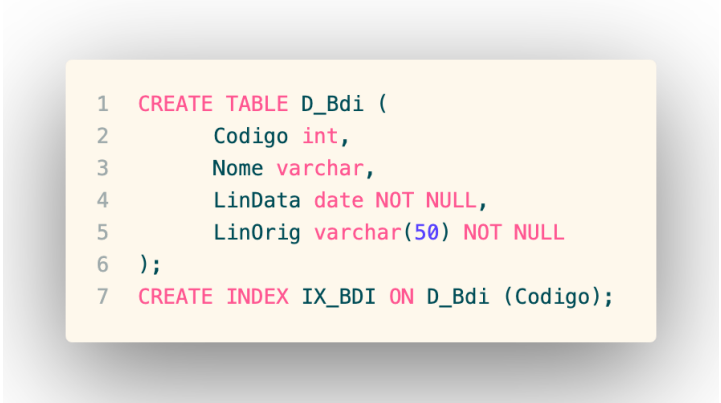


```sql
DROP TABLE IF EXISTS D_Tempo;
DROP TABLE IF EXISTS D_Bdi;
DROP TABLE IF EXISTS D_Especificacao;
DROP TABLE IF EXISTS D_Indopc;
DROP TABLE IF EXISTS D_Tpmerc;
DROP TABLE IF EXISTS D_Nomeacao;
DROP TABLE IF EXISTS F_B3;
DROP TABLE IF EXISTS Importa_Dados;
```

Tabela da dimensão Tempo

```sql
CREATE TABLE D_Tempo (
      Data date PRIMARY KEY NOT NULL,
      Dia char(2) NOT NULL,
      Mes char(2) NOT NULL,
      Ano char(4) NOT NULL
);
```

Tabela da dimensão BDI



```sql
CREATE TABLE D_Bdi (
      Codigo int,
      Nome varchar,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_BDI ON D_Bdi (Codigo);
```

Tabela da dimensão Especificação

```sql
CREATE TABLE D_Especificacao (
      Sigla varchar NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Especificacao ON D_Especificacao (Sigla);
```

Tabela da dimensão INDOPC

```sql
CREATE TABLE D_Indopc (
      Codigo varchar NOT NULL,
      Sigla varchar NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Indopc ON D_Indopc (Codigo);
```

Tabela da dimensão Tipo Mercado

```sql
CREATE TABLE D_Tpmerc (
      Id_Tpmerc int PRIMARY KEY NOT NULL DEFAULT 0,
      Codigo int NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Tpmerc ON D_Tpmerc (Codigo);
```

Tabela da dimensão Nomeação

```sql
CREATE TABLE D_Nomeacao  (
      Codneg varchar NOT NULL,
      Nomres varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Nomeacao ON D_Nomeacao (Codneg);
```

Tabela do fato B3

```sql
CREATE TABLE F_B3  (
      Data date NOT NULL,
      Id_Bdi int NOT NULL,
      Id_Especificacao int NOT NULL,
      Id_Indopc int NOT NULL,
      Id_Tpmerc int NOT NULL,
      Id_Nomeacao int NOT NULL,
      tipreg int NOT NULL,
      prazot int,
      modref varchar NOT NULL,
      preabe float NOT NULL,
      premax float NOT NULL,
      premin float NOT NULL,
      premed float NOT NULL,
      preult float NOT NULL,
      preofc float NOT NULL,
      preofv float NOT NULL,
      totneg bigint NOT NULL,
      quatot bigint NOT NULL,
      voltot float NOT NULL,
      preexe float NOT NULL,
      datven date NOT NULL,
      fatcot bigint NOT NULL,
      ptoexe float NOT NULL,
      codisi varchar NOT NULL,
      dismes int NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL,
      CONSTRAINT PK_F_B3 PRIMARY KEY(
            Data,
            Id_Bdi,
            Id_Especificacao,
            Id_Indopc,
            Id_Tpmerc,
            Id_Nomeacao,
            tipreg
      )
);
```

Tabela para importação dos dados

```
CREATE TABLE importa_dados (
      tipreg int NOT NULL,
      dataPregao varchar NOT NULL,
      codbdi int NOT NULL,
      codneg varchar NOT NULL,
      tpmerc int NOT NULL,
      nomres varchar NOT NULL,
      especi varchar NOT NULL,
      prazot int,
      modref varchar NOT NULL,
      preabe float NOT NULL,
      premax float NOT NULL,
      premin float NOT NULL,
      premed float NOT NULL,
      preult float NOT NULL,
      preofc float NOT NULL,
      preofv float NOT NULL,
      totneg bigint NOT NULL,
      quatot bigint NOT NULL,
      voltot float NOT NULL,
      preexe float NOT NULL,
      indopc bigint NOT NULL,
      datven date NOT NULL,
      fatcot bigint NOT NULL,
      ptoexe float NOT NULL,
      codisi varchar NOT NULL,
      dismes int NOT NULL
);
```

Esquema do Banco de dados do DW

Abaixo estão os trechos de código em SQL relacionados com a construção do Data Warehouse em PostgreSQL.

Remoção das chaves estrangeiras das tabelas, caso existam

```sql
ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Tempo;

ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Bdi;

ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Especificacao;

ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Indopc;

ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Tpmerc;

ALTER TABLE IF EXISTS F_B3
      DROP CONSTRAINT
      IF EXISTS FK_F_B3_D_Nomeacao;
```

Limpeza inicial do banco de dados

```sql
DROP TABLE IF EXISTS D_Tempo;
DROP TABLE IF EXISTS D_Bdi;
DROP TABLE IF EXISTS D_Especificacao;
DROP TABLE IF EXISTS D_Indopc;
DROP TABLE IF EXISTS D_Tpmerc;
DROP TABLE IF EXISTS D_Nomeacao;
DROP TABLE IF EXISTS F_B3;
```

Tabela da dimensão Tempo

```sql
CREATE TABLE D_Tempo (
      Id_Data int NOT NULL DEFAULT 0,
      Data date PRIMARY KEY NOT NULL,
      Dia char(2) NOT NULL,
      Mes char(2) NOT NULL,
      Ano char(4) NOT NULL
);
```

Tabela da dimensão BDI

```sql
CREATE TABLE D_Bdi (
      Id_Bdi int PRIMARY KEY NOT NULL DEFAULT 0,
      Codigo int,
      Nome varchar,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_BDI ON D_Bdi (Codigo);
```

Tabela da dimensão Especificação

```sql
CREATE TABLE D_Especificacao (
      Id_Especificacao int PRIMARY KEY NOT NULL DEFAULT 0,
      Sigla varchar NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Especificacao ON D_Especificacao (Sigla);
```

Tabela da dimensão INDOPC

```
CREATE TABLE D_Indopc (
      Id_Indopc int PRIMARY KEY NOT NULL DEFAULT 0,
      Codigo varchar NOT NULL,
      Sigla varchar NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Indopc ON D_Indopc (Codigo);
```

Tabela da dimensão Tipo Mercado

```
CREATE TABLE D_Tpmerc (
      Id_Tpmerc int PRIMARY KEY NOT NULL DEFAULT 0,
      Codigo int NOT NULL,
      Nome varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Tpmerc ON D_Tpmerc (Codigo);
```

Tabela da dimensão Nomeação

```
CREATE TABLE D_Nomeacao  (
      Id_Nomeacao int PRIMARY KEY NOT NULL DEFAULT 0,
      Codneg varchar NOT NULL,
      Nomres varchar NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL
);
CREATE INDEX IX_Nomeacao ON D_Nomeacao (codneg);
```

‘

Tabela do fato B3

```sql
CREATE TABLE F_B3  (
      Id_Fato int NOT NULL DEFAULT 0,
      Data date NOT NULL,
      Id_Bdi int NOT NULL,
      Id_Especificacao int NOT NULL,
      Id_Indopc int NOT NULL,
      Id_Tpmerc int NOT NULL,
      Id_Nomeacao int NOT NULL,
      tipreg int NOT NULL,
      prazot int,
      modref varchar NOT NULL,
      preabe float NOT NULL,
      premax float NOT NULL,
      premin float NOT NULL,
      premed float NOT NULL,
      preult float NOT NULL,
      preofc float NOT NULL,
      preofv float NOT NULL,
      totneg float NOT NULL,
      quatot bigint NOT NULL,
      voltot float NOT NULL,
      preexe float NOT NULL,
      datven date NOT NULL,
      fatcot bigint NOT NULL,
      ptoexe float NOT NULL,
      codisi varchar NOT NULL,
      dismes int NOT NULL,
      LinData date NOT NULL,
      LinOrig varchar(50) NOT NULL,
      CONSTRAINT PK_F_B3 PRIMARY KEY(
            Data,
            Id_Bdi,
            Id_Especificacao,
            Id_Indopc,
            Id_Tpmerc,
            Id_Nomeacao,
            tipreg
      ),
      CONSTRAINT FK_F_B3_D_Tempo FOREIGN KEY (Data)
            REFERENCES D_Tempo (Data) ON DELETE CASCADE,
      CONSTRAINT FK_F_B3_D_Bdi FOREIGN KEY(Id_Bdi)
            REFERENCES D_Bdi (Id_Bdi) ON DELETE CASCADE,
      CONSTRAINT FK_F_B3_D_Especificacao FOREIGN KEY (Id_Especificacao)
            REFERENCES D_Especificacao (Id_Especificacao) ON DELETE CASCADE,
      CONSTRAINT FK_F_B3_D_Indopc FOREIGN KEY (Id_Indopc)
            REFERENCES D_Indopc (Id_Indopc) ON DELETE CASCADE,
      CONSTRAINT FK_F_B3_D_Tpmerc FOREIGN KEY (Id_Tpmerc)
            REFERENCES D_Tpmerc (Id_Tpmerc) ON DELETE CASCADE,
      CONSTRAINT FK_F_B3_D_Nomeacao FOREIGN KEY (Id_Nomeacao)
);
```

Descrição das Execuções

- **Transformação 01 - Importar dados para o DS**

    - **O que faz:** Carrega os arquivos de entrada, renomeia os campos e os insere na tabela importa_dados do DS.

    - **Passos:**
        - **(1) Text file input:** Seleciona os arquivos contendo os dados desejados. Como exemplo, temos os arquivos 2016.csv, 2017.csv, 2018.csv e 2019.csv.
        - **(2) Select values:** Seleciona os campos, renomeando-os e informando seu tipo e tamanho.
        - **(3) Table output:** Realiza a conexão com a tabela importa_dados do banco de dados DS, inserindo os dados selecionados.

    - Número de registros inseridos: 2315233 (dois milhões trezentos e quinze mil duzentos e trinta e três).

- **Transformação 02 – Dimensão Tempo para o DS:**

    O que faz: Seleciona os dados pertinentes à tabela D_Tempo da tabela importa_vendas, separa a data em campos separados e formata a data completa. Em seguida, verifica se existem valores duplicados, campos nulos e por fim envia os objetos do fluxo para a tabela D_Tempo no DS.

    - **Passos:**
        - **(1) Table input:** Realiza a conexão com o banco de dados DS para buscar os campos desejados da tabela importa_dados, por meio de uma instrução SELECT de SQL.
        - **(2) Strings cut:** Separa a data completa em ano, mês e dia.
        - **(3) Replace in string:** No campo datapregao, substitui “.” por “-”.
        - **(4) Select values BI:** Seleciona os campos datapregao, ano, mês e dia.
        - **(5) If field value is null:** Verifica se os campos datapregao, ano, mes e dia. Em caso positivo, insere os valores 1900/01/01, 1900, 01, 01 respectivamente.
        - **(6) Sort rows:** Ordena os objetos de maneira ascendente de acordo com os campos dataPregao, ano, mês e dia.
        - **(7) Unique rows:** Elimina objetos duplicados.
        - **(8) Table input:** Realiza a conexão com o banco de dados, para carregar os atributos que vêm do fluxo à tabela D_Tempo do DS.

    - Número de registros inseridos: 988 (novecentos e oitenta e oito).

- **Transformação 03 – Dimensão Tempo para o DW:**

- **O que faz:** Seleciona os objetos da tabela D_Tempo no DS para inseri-los no DW, adicionando um ID incremental único para cada objeto.

- **Passos:**
  - **(1) Table input:** Conecta a tabela D_Tempo no DS para permitir a seleção dos valores.
  - **(2) Select values:** Seleciona os campos data, dia, mes e ano.
  - **(3) Combination lookup/update:** Realiza a conexão com a tabela D_Tempo no DW para inserir os campos que vem do fluxo nos campos Data, Dia, Mes e Ano. Além disso, para cada objeto insere um campo Id_Data incremental.

- Número de registros inseridos: 988 (novecentos e oitenta e oito).

- **Transformação 04 – Dimensão BDI para o DS:**

  - **O que faz:** Seleciona os dados do arquivo Boletim_diario_do_mercado.xlsx, renomeia o nome dos campos, utiliza a data do sistema para preencher o campo LinData, preenche o campo LinOrig e, por fim, insere os objetos na tabela D_BDI do DS.

  - **Passos:**
    - **(1) Microsoft Excel input:** Seleciona os campos pertinentes através de uma conexão com o arquivo Boletim_diario_do_mercado.xlsx.
    - **(2) Select values:** Utiliza a data do sistema para preencher o campo LinData.
    - **(3) Get system info:** Utiliza a data do sistema para preencher o campo LinData.
    - **(4) Add constants:** Adiciona o valor “Arquivo BDI” no campo LinOrig.
    - **(5) Table Output:** Realiza a conexão com o banco de dados, para carregar os atributos que vêm do fluxo à tabela D_BDI do DS.

  - Número de registros inseridos: 44 (quarenta e quatro).

- **Transformação 05 – Dimensão BDI para o DW:**

- **O que faz:** Seleciona os objetos da tabela D_BDI no DS para inseri-los no DW, adicionando um ID incremental único para cada objeto.

- **Passos:**
  - **(1) Table input:** Realiza a conexão da tabela D_BDI no DS para permitir a seleção dos valores.
  - **(2) Combination lookup/update:** Se conecta a tabela D_BDI no DW para inserir os campos que vem do fluxo nos campos Codigo, Nome, LinData e LinOrig. Além disso, para cada objeto insere um campo Id_BDI incremental.

- Número de registros inseridos: 44 (quarenta e quatro).

● **Transformação 06 – Dimensão INDOPC para o DS:**

- **O que faz:** Seleciona os dados do arquivo Tabela_de_IDOPC.xlsx, renomeia o nome dos campos, utiliza a data do sistema para preencher o campo LinData, preenche o campo LinOrig e insere os objetos na tabela D_INDOPC do DS.

- **Passos:**

  ■ **(1) Microsoft Excel Input:** Seleciona os campos pertinentes através de uma conexão com o arquivo Tabela_de_IDOPC.xlsx.
  ■ **(2) Select values:** Seleciona os campos Codigo_INDOPC, Sigla_INDOPC e Nome_INDOPC, renomeando-os para codigo, sigla e nome.
  ■ **(3) Get system info:** Utiliza a data do sistema para preencher o campo LinData.
  ■ **(4) Add constants:** Adiciona o valor “Arquivo INDOPC” no campo LinOrig.
  ■ **(5) Table output:** Realiza a conexão com o banco de dados, para carregar os atributos que vêm do fluxo à tabela D_INDOPC do DS.
- Número de registros inseridos: 4 (quatro).

● **Transformação 07 – Dimensão INDOPC para o DW:**

- **O que faz:** Seleciona os objetos da tabela D_INDOPC no DS para inseri-los no DW, adicionando um ID incremental único para cada objeto.

- **Passos:**

  ■ **(1) Table input:** Se conecta a tabela D_INDOPC no DS para permitir a seleção dos valores.
  ■ **(2) Combination lookup/update:** Se conecta a tabela D_Especificacao no DW para inserir os campos que vem do fluxo nos campos Codigo, Sigla, Nome, LinData e LinOrig. Além disso, para cada objeto insere um campo Id_INDOPC incremental.
- Número de registros inseridos: 4 (quatro).

● **Transformação 08 – Dimensão Especificação para o DS:**

- **O que faz:** Seleciona os dados do arquivo Tabela_de_especificacao.xlsx, renomeia o nome dos campos, utiliza a data do sistema para preencher o campo LinData, preenche o campo LinOrig e insere os objetos na tabela D_Especificacao do DS.

- **Passos:**

  ■ **(1) Microsoft Excel Input:** Seleciona os campos pertinentes através de uma conexão com o arquivo Tabela_de_especificacao.xlsx.
  ■ **(2) Select values:** Seleciona os campos Sigla_especificacao e Nome_especificacao, renomeando-os para sigla e nome.
  ■ **(3) Get system info:** Utiliza a data do sistema para preencher o campo LinData.
  ■ **(4) Add constants:** Adiciona o valor "Arquivo especificação" no campo LinOrig.
  ■ **(5) Table output:** Realiza a conexão com o banco de dados, para carregar os atributos que vêm do fluxo à tabela D_Especificacao do DS.
- Número de registros inseridos: 189 (cento e oitenta e nove).

● **Transformação 09 – Dimensão Especificação para o DW:**

- **O que faz:** Seleciona os objetos da tabela D_Especificacao no DS para inseri-los no DW, adicionando um ID incremental único para cada objeto.

- **Passos:**

  ■ **(1) Table input:** Se conecta a tabela D_Especificacao no DS para permitir a seleção dos valores.
  ■ **(2) Combination lookup/update:** Se conecta a tabela D_Especificacao no DW para inserir os campos que vem do fluxo nos campos Sigla, Nome, LinData e LinOrig. Além disso, para cada objeto insere um campo Id_Especificacao incremental.
- Número de registros inseridos: 189 (cento e oitenta e nove).

● **Transformação 10 – Dimensão TipoMercado para o DS:**

- **O que faz:** Seleciona os dados do arquivo Tabela_de_TipoMercado.xlsx, renomeia o nome dos campos, utiliza a data do sistema para preencher o campo LinData, preenche o campo LinOrig e insere os objetos na tabela D_Tpmerc do DS.

- **Passos:**

  ■ **(1) Microsoft Excel Input:** Seleciona os campos pertinentes através de uma conexão com o arquivo Tabela_de_TipoMercado.xlsx.

■ **(2) Select values:** Seleciona os campos Codigo_tpmerc, Nome_tpmerc, renomeando-os para codigo e nome.
■ **(3) Get system info:** Utiliza a data do sistema para preencher o campo LinData.
■ **(4) Add constants:** Adiciona o valor “Arquivo Excel Tabela TipoMercado” no campo LinOrig.
■ **(5) Table output:** Realiza a conexão com o banco de dados, para carregar os atributos que vêm do fluxo à tabela D_Tpmerc do DS.

- Número de registros inseridos: 10 (dez).

● **Transformação 11 – Dimensão TipoMercado para o DW:**

○ **O que faz:** Seleciona objetos da tabela D_Tpmerc no DS, os colocando no DW. Também incrementa um ID único em cada objeto.

○ **Passos:**
■ (**1) Table input:** Faz a conexão a tabela D_Tpmerc no DS e seleciona os valores.
■ **(2) Combination lookup/update:** Conexão a tabela D_Tpmerc no DW, insere os campos que vem do fluxo nos campos Codigo, Nome, LinData e LinOrig. Para cada objeto, Id_Tpmerc incrementa.

○ Número de registros inseridos: 10 (dez).

● **Transformação 12 – Dimensão Nomeação para o DS:**

○ **O que faz:** Seleciona Codneg e Nomres da tabela importa_dados, utilizando a data do sistema para preencher LinData e LinOrig. Insere os objetos na tabela D_Nomeacao do DS.

○ **Passos:**
■ **(1) Table input:** Seleciona Codneg e Nomres da tabela do DS importa_dados.
■ **(2) Get system info:** Preenche LinData com a data do sistema.
■ **(3) Add constants:** “Arquivo TXT Histórico de Cotação” é adicionado ao campo LinOrig.
■ **(4) Table output:** Realiza a conexão com o banco de dados, carregando o fluxo à tabela D_Nomeacao do DS.

○ **Número de registros inseridos:** 2315233 (dois milhões trezentos e quinze mil trezentos e trinta e três)

● **Transformação 13 – Dimensão Nomeação para o DW:**

○ **O que faz:** Seleciona dados da tabela D_Nomeacao no DS para inseri-los no DW. Também incrementa um ID único em cada objeto.

○ **Passos:**

■ **(1) Table input:** Faz a conexão a tabela D_Nomeacao no DS para permitir a seleção dos valores.
■ **(2) Combination lookup/update:** Se conecta a tabela D_Nomeacao no DW, inserindo dados vindos de Codneg, Nomres, LinData e LinOrig. Também incrementa unicamente cada objeto inserido.

○ Número de registros inseridos: 2315233 (dois milhões trezentos e quinze mil trezentos e trinta e três)

● **Transformação 14 – Fato B3 para o DS:**

○ **O que faz:** Seleciona dados da tabela importa_dados no DS. Preenche os campos LinOrig, cem, e milhao com valores padrões. Após fazer isso, formata o campo especi em três operações distintas. Busca em DW os campos Id_Bdi, Id_Especificacao, Id_Indopc, Id_Nomeacao e Id_Tpmerc, preenchendo LinData. Para finalizar, verifica se existem nulos e insere os dados do fluxo na tabela F_B3 do DS.

○ **Passos:**
■ **(1) Table Input:** É realizada uma conexão com o banco de dados DS, buscando campos de importa_dados, por meio de uma instrução SELECT de SQL.
■ **(2) Select values:** Os campos voltot, tpmerc, totneg, tipreg, quatot, ptoexe, preult, preofv, preofc, premin, premed, premax, preexe, preabe, prazot, modref, indopc, fatcot, especi, dismes, datven, datapregao, codneg, codisi e codbi são selecionados. Renomeia o campo datapregao para data.
■ **(3) Add constants**: Preenche LinOrig, cem e milhao com os valores “Arquivo TXT de Cotação Histórica”, “100” e “1000000” respectivamente.
■ **(4) Calculator:** Divide preabe, premax, premin, premed, preult, preofc, preofv, totneg, voltot, preexe por 100, criando campos de mesma nomenclatura e um novo sufixo, no caso, “_1”, para receber o resultado e manter os dados. Também divide o campo ptoexe por 1000000, realizando uma operação de nomenclatura semelhante a já descrita, utilizando do sufixo “_1” com o valor do resultado operação.
■ **(5) Strings cut:** Seleciona os oito primeiros caracteres do campo especi.
■ **(6) Replace in string:** Remove espaços duplos no campo especi, substituindo-os por um espaço.
■ **(7) String operations:** Remove os valores nulos à esquerda e direita do campo especi.
■ **(8) Database lookup:** Utiliza codbdi para comparar com o campo codigo da tabela d_bdi do banco DW. Quando encontra um objeto com este campo equivalente, busca o campo Id_Bdi do mesmo.
■ **(9) Database lookup 2:** Utiliza o campo especi para realizar comparações com a sigla da tabela d_especificacao do banco DW. Quando encontra um objeto com este campo equivalente, busca o campo Id_Especificacao do mesmo.
■ **(10) Database lookup 3:** Utiliza o campo indopc para comparar com o campo codigo da tabela d_indopc do banco DW. Caso encontre um objeto com este campo equivalente, busca o campo Id_Indopc do mesmo.

■ **(11) Database lookup 4:** Utiliza o campo codneg para comparar com o campo codneg da tabela d_nomeacao do banco DW. Quando encontra um objeto com este campo equivalente, busca o campo Id_Nomeacao do mesmo.
■ **(12) Database lookup 5:** Utiliza o campo tpmerc para comparar com o campo codigo da tabela d_tpmerc do banco DW. Quando encontra um objeto com este campo equivalente, busca o campo Id_Tpmerc do mesmo.
■ **(13) Get system info:** Insere no campo LinData a data atual do sistema.
■ **(14) If field value is null:** Verifica se os campos LinOrig, prazot, id_especificacao e id_indopc são nulos. Se positivo, preenche com os valores “Registro padrão inserido manualmente”, 0, 1 e 1 respectivamente.
■ **(15) Insert/update fields:** Associa os atributos que vem do fluxo com os atributos da tabela F_B3 no DS que vão ser inseridos / atualizados.

○ Número de registros inseridos: 2315233 (dois milhões trezentos e quinze mil duzentos e trinta e três).

● **Transformação 15 – Fato B3 para o DW:**

○ **O que faz:** Seleciona dados da tabela D_B3 no DS, insere no DW e adiciona um ID incremental único para cada objeto.

○ **Passos:**
■ **(1) Table input:** Se conecta a tabela F_B3 no DS e seleciona os valores.
■ **(2) Combination lookup/update:** Se conecta a tabela F_B3 no DW para atualizar dados do fluxo para os campos data, id_bdi, id_especificacao, id_indopc, id_tpmerc, id_nomeacao, voltot, tpmerc, totneg, tipreg, quatot, ptoexe, preult, preofv, preofc, premin, premed, premax, preexe, preabe, prazot, modref, indopc, fatcot, especi, dismes, datven, data, codneg, codisi e codbi. Por fim, cria o campo Id_Fato de id incremental.

○ Número de registros inseridos: 2225224 (dois milhões duzentos e vinte e cinco mil duzentos e vinte e quatro).

● **Job:**

○ **O que faz:** Organiza um job (1) a fim de rodar e executar todas as transformações (2 - 16) de maneira automática.

○ **Passos:**
■ **(1) Start:** Configura o job. Organiza agendamentos e repetições.
■ **(2) Transformation Importa Dados:** Executa a transformação de importação dos dados para o DS.
■ **(3) Transformation D_Tempo DS:** Executa a transformação da dimensão Tempo para o DS.
■ **(4) Transformation D_Tempo DW:** Executa a transformação da dimensão Tempo para o DW.

■ **(5) Transformation D_BDI DS:** Executa a transformação da dimensão BDI para o DS.
■ **(6) Transformation D_BDI DW:** Executa a transformação da dimensão BDI para o DW.
■ **(7) Transformation D_INDOPC DS:** Executa a transformação da dimensão INDOPC para o DS.
■ **(8) Transformation D_INDOPC DW:** Executa a transformação da dimensão INDOPC para o DW.
■ **(9) Transformation D_Especificacao DS:** Executa a transformação da dimensão Especificacao para o DS.
■ **(10) Transformation D_Especificacao DW:** Executa a transformação da dimensão Especificacao para o DW.
■ **(11) Transformation D_Tpmercado DS:** Executa a transformação da dimensão Tipo Mercado para o DS.
■ **(12) Transformation D_Tpmercado DW:** Executa a transformação da dimensão Tipo Mercado para o DW.
■ **(13) Transformation D_Nomeacao DS:** Executa a transformação da dimensão Nomeacao para o DS.
■ **(14) Transformation D_Nomeacao DW:** Executa a transformação da dimensão Nomeacao para o DW.
■ **(15) Transformation D_FatoB3 DS:** Executa a transformação da dimensão FatoB3 para o DS.
■ **(16) Transformation D_FatoB3 DW:** Executa a transformação da dimensão FatoB3 para o DW.

Construção do Data Warehouse

Para a construção do Data Warehouse, deve-se notar que a ordem em que as transformações são executadas são de extrema importância. Qualquer transformação realizada fora da sequência citada, pode afetar diretamente o resultado.

O início do processo de transformação envolve trabalhar a massa de dados dos arquivos CSV da Cotação Histórica da B3 dos anos de 2016, 2017, 2018 e 2019 em tabelas de importação, selecionando fatos e dimensões relevantes para a manipulação de dados.

Outro ponto a se notar, são os dados que não provém das tabelas de importação. Esses dados são providos por planilhas auxiliares, e são utilizadas em algumas das transformações, como por exemplo, Transformação 04 – Dimensão BDI para o DS.

As transformações são inseridas no DS ou no DW. Desta forma, as transformações DS devem ser validadas e tratadas antes de suas inserções, enquanto em DW, essas verificações não são necessárias (já que já foram feitas em DS).

O job, o passo final, é automatização de todos os passos propostos, de maneira a simplificar a execução das transformações.

Dificuldades e Dúvidas

Algumas dificuldades no trabalho envolveram o tratamento de dados em certos pontos do trabalho. Inconsistências e tipagem dos valores foram as maiores dificuldades encontradas, como por exemplo, valores decimais representados como números inteiros.

Conclusão

Conseguimos concluir com este trabalho, o quanto o impacto das transformações pode alterar o resultado entregue. A automatização de um processo como este é de extrema importância quando se trabalha com massas de dados como as exemplificados no trabalho, que caso fosse realizada de maneira manual e individualmente, levaria muito mais tempo do que o necessário.

A ferramenta Pentaho Data Integration, possibilita a automatização de maneira prática e eficiente, e se torna mais aliado para a nossa formação.

O trabalho possibilitou a visualização do uso do software de maneira mais visual e integrada, solidificando nosso conhecimento e aumentando nossos horizontes sobre como podemos aplicar o conhecimento adquirido ao longo da graduação em outras aplicações.