



TRABALHO PRÁTICO 2

Bases de Dados e Armazém de Dados

Licenciatura em Engenharia de Telecomunicações e Informática
Instituto Superior de Engenharia do Porto
Janeiro de 2025

Luís Paulo Teixeira Oliveira, 1231002
Miguel Fundevila Pinto Da Silva, 1231058
Tiago Dos Santos Oliveira, 1230947

Introdução

O presente relatório pretende descrever a elaboração do segundo trabalho prático realizado no âmbito da unidade curricular de Bases de Dados e Armazém de Dados, para o qual foi proposto o desenvolvimento de um *data mart* a partir de dados respetivos a encomendas de produtos, feitas por clientes de uma dada empresa que comercializa bens alimentares.

Numa primeira fase, analisou-se o modelo de dados relativo ao sistema operacional fornecido e recorreu-se à construção de um modelo dimensional com o objetivo de definir um esquema concetual para o *data mart*.



Arquitetura do data mart

Definição de Data Mart

Um *data mart* é um pequeno armazém de dados focado em apenas um conjunto de factos ligados a uma única área do negócio em questão, neste caso, as encomendas de produtos.

Modelação dimensional

A modelação dimensional de um *data mart* passa, essencialmente, pela criação de um modelo em estrela baseado em tabelas de factos e tabelas dimensionais (denominadas dimensões ao longo deste relatório). Para tal, seguiu-se a metodologia de modelação dimensional de Kimball, que envolve identificar quatro pontos chave: área de negócio, nível de detalhe, dimensões e factos.

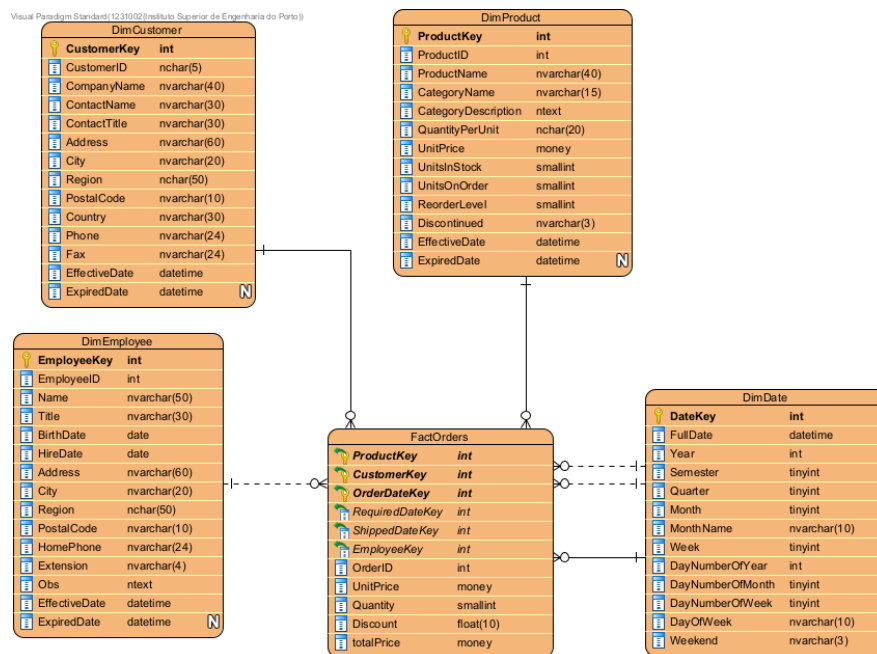
1. Área de negócio: gestão de encomendas e vendas;
2. Nível de detalhe: encomendas por cliente por produto por data;
3. Dimensões: customer, employee, product, date;
4. Factos: UnitPrice, quantity, discount, total price.

Seguindo esta metodologia, foi possível identificar que seriam criadas quatro dimensões: a DimCustomer, a DimEmployee, a DimProduct e a DimDate.

Foi também possível definir os factos que constituem a tabela de factos: preço unitário, quantidade, desconto e preço total.

Identificadas as dimensões e os factos, partiu-se para a construção do modelo dimensional. Como é possível constatar, os atributos das dimensões mantêm-se os mesmos das tabelas da base de dados operacional, apenas tendo sido adicionadas *surrogate keys* nas quais se iram basear os joins entre as dimensões e a tabela de factos, em vez de se basearem nas *primary keys* da base de dados operacional, evitando inconsistências no tipo de dados e protegendo o *data mart* de alterações feitas na base de dados operacional.

No caso da tabela de factos, foram seleccionadas apenas três das *foreign keys* para serem *primary keys*. Um cliente encomenda um produto num dia a uma certa hora, e se o mesmo produto for encomendado pelo mesmo cliente nesse mesmo dia a hora será diferente (Na tabela do sistema operacional o atributo é apenas date, o que se torna limitativo. Foi assumido como erro).



Estruturas de dados criadas

Script de criação da DimCustomer:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'DimCustomer')
BEGIN
    CREATE TABLE [dbo].[DimCustomer](
        [CustomerKey] [int] IDENTITY(1,1) NOT NULL,
        [CustomerID] [nchar](5) NOT NULL,
        [CompanyName] [nvarchar](40) NOT NULL,
        [ContactName] [nvarchar](30) NOT NULL,
        [ContactTitle] [nvarchar](30) NOT NULL,
        [Address] [nvarchar](60) NOT NULL,
        [City] [nvarchar](20) NOT NULL,
        [Region] [nchar](50) NOT NULL,
        [PostalCode] [nvarchar](10) NOT NULL,
        [Country] [nvarchar](30) NOT NULL,
        [Phone] [nvarchar](24) NOT NULL,
        [Fax] [nvarchar](24) NOT NULL,
        [EffectiveDate] [datetime] NOT NULL,
        [ExpiredDate] [datetime] NULL
    CONSTRAINT [PK_DimCustomer] PRIMARY KEY CLUSTERED
    (
        [CustomerKey] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

    CREATE NONCLUSTERED INDEX [NonClusteredIndex-CustomerID] ON [dbo].[DimCustomer]
    (
        [CustomerID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
END
```

Script de criação da DimDate:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'DimDate')
BEGIN
    CREATE TABLE [dbo].[DimDate](
        [DateKey] [int] IDENTITY(1,1) NOT NULL,
        [FullDate] [datetime] NOT NULL,
        [Year] [int] NOT NULL,
        [Semester] [tinyint] NOT NULL,
        [Quarter] [tinyint] NOT NULL,
        [Month] [tinyint] NOT NULL,
        [MonthName] [nvarchar](10) NOT NULL,
        [Week] [tinyint] NOT NULL,
        [DayNumberOfYear] [int] NOT NULL,
        [DayNumberOfMonth] [tinyint] NOT NULL,
        [DayNumberOfWeek] [tinyint] NOT NULL,
        [DayOfWeek] [nvarchar](10) NOT NULL,
        [Weekend] [nvarchar](3) NOT NULL
    CONSTRAINT [PK_DimDate] PRIMARY KEY CLUSTERED
    (
        [DateKey] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]
END
```

Script de criação da DimEmployee:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'DimEmployee')
BEGIN
    CREATE TABLE [dbo].[DimEmployee](
        [EmployeeKey] [int] IDENTITY(1,1) NOT NULL,
        [EmployeeID] [int] NOT NULL,
        [Name] [nvarchar](50) NOT NULL,
        [Title] [nvarchar](30) NOT NULL,
        [BirthDate] [date] NOT NULL,
        [HireDate] [date] NOT NULL,
        [Address] [nvarchar](60) NOT NULL,
        [City] [nvarchar](20) NOT NULL,
        [Region] [nchar](50) NOT NULL,
        [PostalCode] [nvarchar](10) NOT NULL,
        [HomePhone] [nvarchar](24) NOT NULL,
        [Extension] [nvarchar](4) NOT NULL,
        [Obs] [ntext] NOT NULL,
        [EffectiveDate] [datetime] NOT NULL,
        [ExpiredDate] [datetime] NULL
    CONSTRAINT [PK_DimEmployee] PRIMARY KEY CLUSTERED
    (
        [EmployeeKey] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

    CREATE NONCLUSTERED INDEX [NonClusteredIndex-EmployeeID] ON [dbo].[DimEmployee]
    (
        [EmployeeID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
END
```

Script de criação da DimProduct:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'DimProduct')
BEGIN
    CREATE TABLE [dbo].[DimProduct](
        [ProductKey] [int] IDENTITY(1,1) NOT NULL,
        [ProductID] [int] NOT NULL,
        [ProductName] [nvarchar](40) NOT NULL,
        [CategoryName] [nvarchar](15) NOT NULL,
        [CategoryDescription] [ntext] NOT NULL,
        [QuantityPerUnit] [nchar](20) NOT NULL,
        [UnitPrice] [money] NOT NULL,
        [UnitsInStock] [smallint] NOT NULL,
        [UnitsOnOrder] [smallint] NOT NULL,
        [ReorderLevel] [smallint] NOT NULL,
        [Discontinued] [nvarchar](3) NOT NULL,
        [EffectiveDate] [datetime] NOT NULL,
        [ExpiredDate] [datetime] NULL
    CONSTRAINT [PK_DimProduct] PRIMARY KEY CLUSTERED
    (
        [ProductKey] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

    CREATE NONCLUSTERED INDEX [NonClusteredIndex-ProductID] ON [dbo].[DimProduct]
    (
        [ProductID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
END
```

Script de criação da tabela de factos, FactOrders:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'FactOrders')
CREATE TABLE [dbo].[FactOrders](
    [ProductKey] [int] NOT NULL,
    [CustomerKey] [int] NOT NULL,
    [OrderDateKey] [int] NOT NULL,
    [RequiredDateKey] [int] NOT NULL,
    [ShippedDateKey] [int] NOT NULL,
    [EmployeeKey] [int] NOT NULL,
    [OrderID] [int] NOT NULL,
    [UnitPrice][money] NOT NULL,
    [Quantity][smallint] NOT NULL,
    [Discount][float] NOT NULL,
    [totalPrice][money] NOT NULL
CONSTRAINT [PK_FactOrders] PRIMARY KEY CLUSTERED
(
    [ProductKey] ASC,
    [CustomerKey] ASC,
    [OrderDateKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Staging area

A staging area serve de intermediário entre a base de dados operacional e o data mart, constituindo um ponto onde são aplicados processos de extração, transformação e carregamento que asseguram que os dados carregados contêm o menor número de erros possível.

Estruturas de dados criadas

Script de criação da tabela Categories:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Categories')
CREATE TABLE [dbo].[Categories]
(
    [CategoryID] int,
    [CategoryName] [nvarchar](15),
    [Description] [ntext]
)
ELSE
TRUNCATE TABLE [Categories]
```

Script de criação da tabela Customers:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Customers')
CREATE TABLE [dbo].[Customers]
(
    [CustomerID] [nchar](5),
    [CompanyName] [nvarchar](40),
    [ContactName] [nvarchar](30),
    [ContactTitle][nvarchar](30),
    [Address][nvarchar](60),
    [City][nvarchar](20),
    [RegionID] [int],
    [PostalCode][nvarchar](10),
    [Country][nvarchar](30),
    [Phone][nvarchar](24),
    [Fax][nvarchar](24)
)
ELSE
TRUNCATE TABLE [Customers]
```

Script de criação da tabela CustomersDQP:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'CustomersDQP')
CREATE TABLE [dbo].[CustomersDQP]
(
    [CustomerID] [nchar](5),
    [CompanyName] [nvarchar](40),
    [ContactName] [nvarchar](30),
    [ContactTitle][nvarchar](30),
    [Address][nvarchar](60),
    [City][nvarchar](20),
    [Region][nchar](50),
    [PostalCode][nvarchar](10),
    [Country][nvarchar](30),
    [Phone][nvarchar](24),
    [Fax][nvarchar](24),
    DQP nvarchar(100)
)
ELSE
TRUNCATE TABLE CustomersDQP
```

Script de criação da tabela Employees:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Employees')
    CREATE TABLE [dbo].[Employees]
    (
        [EmployeeID] int,
        [Name] [nvarchar](50),
        [Title] [nvarchar](30),
        [BirthDate][date],
        [HireDate][date],
        [Address][nvarchar](60),
        [City] [nvarchar](20),
        [RegionID] int,
        [PostalCode][nvarchar](10),
        [HomePhone][nvarchar](24),
        [Extension][nvarchar](4),
        [Obs][nvarchar](MAX)
    )
ELSE
    TRUNCATE TABLE [Employees]
```

Script de criação da tabela OrderDetails:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'OrderDetails')
    CREATE TABLE [dbo].[OrderDetails]
    (
        [OrderID] int,
        [ProductID] int,
        [UnitPrice] [money],
        [Quantity] smallint,
        [Discount][float]
    )
ELSE
    TRUNCATE TABLE [OrderDetails]
```

Script de criação da tabela Orders:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Orders')
    CREATE TABLE [dbo].[Orders]
    (
        [OrderID] int,
        [CustomerID][nchar](5),
        [EmployeeID] int,
        [OrderDate][datetime],
        [RequiredDate][datetime],
        [ShippedDate][datetime]
    )
ELSE
    TRUNCATE TABLE [Orders]
```

Script de criação da tabela Products:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Products')
    CREATE TABLE [dbo].[Products]
    (
        [ProductID] int,
        [ProductName][nvarchar](40),
        [CategoryID] int,
        [QuantityPerUnit][nchar](20),
        [UnitPrice][money],
        [UnitsInStock]smallint,
        [UnitsOnOrder] smallint,
        [ReorderLevel] smallint,
        [Discontinued][bit]
    )
ELSE
    TRUNCATE TABLE [Products]
```

Script de criação da tabela Regions:

```
IF NOT EXISTS (SELECT name FROM sys.tables WHERE name = 'Regions')
    CREATE TABLE [dbo].[Regions]
    (
        [RegionID] int,
        [RegionDescription] [nchar](50)
    )
ELSE
    TRUNCATE TABLE [Regions]
```


Processos de extração, transformação e carregamento

| Target | | | | | Source | | | | Transformation |
|--------------|----------------------|--------------|------------|---------|---|--------------------------|--------------------|--------------|--|
| Table Name | Column Name | Data Type | Table Type | CD Type | Database Name | Table Name | Column Name | Data Type | |
| DIM_CUSTOMER | CUSTOMER_KEY | NUMBER | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | | NUMBER | Surrogate key |
| DIM_CUSTOMER | CUSTOMER_ID | NCHAR(5) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | CUSTOMER_ID | NCHAR(5) | |
| DIM_CUSTOMER | COMPANY_NAME | NVARCHAR(40) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | COMPANY_NAME | NVARCHAR(30) | |
| DIM_CUSTOMER | CONTACT_NAME | NVARCHAR(30) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | CONTACT_NAME | NVARCHAR(30) | |
| DIM_CUSTOMER | CONTACT_TITLE | NVARCHAR(30) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | CONTACT_TITLE | NVARCHAR(30) | |
| DIM_CUSTOMER | ADDRESS | NVARCHAR(60) | Dimension | 2 | OPERATIONAL_DB | CUSTOMERS | ADDRESS | NVARCHAR(60) | |
| DIM_CUSTOMER | CITY | NVARCHAR(20) | Dimension | 2 | OPERATIONAL_DB | CUSTOMERS | CITY | NVARCHAR(20) | SELECT dbo.Regions.RegionDescription FROM dbo.Regions WHERE dbo.Customers.RegionID = dbo.Regions.RegionID |
| | | | | 1 | | | | | |
| DIM_CUSTOMER | REGION | NCHAR(50) | Dimension | | OPERATIONAL_DB | REGIONS | REGION_DESCRIPTION | NCHAR(50) | |
| DIM_CUSTOMER | POSTAL_CODE | NVARCHAR(10) | Dimension | 2 | OPERATIONAL_DB | CUSTOMERS | POSTAL_CODE | NVARCHAR(10) | |
| DIM_CUSTOMER | COUNTRY | NVARCHAR(30) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | COUNTRY | NVARCHAR(30) | |
| DIM_CUSTOMER | PHONE | NVARCHAR(24) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | PHONE | NVARCHAR(24) | |
| DIM_CUSTOMER | FAX | NVARCHAR(24) | Dimension | 1 | OPERATIONAL_DB | CUSTOMERS | FAX | NVARCHAR(24) | Surrogate key |
| DIM_EMPLOYEE | EMPLOYEE_KEY | NUMBER | Dimension | 1 | | | | NUMBER | |
| DIM_EMPLOYEE | EMPLOYEE_ID | NUMBER | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | EMPLOYEE_ID | NUMBER | |
| DIM_EMPLOYEE | NAME | NVARCHAR(50) | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | NAME | NVARCHAR(50) | |
| DIM_EMPLOYEE | TITLE | NVARCHAR(30) | Dimension | 2 | OPERATIONAL_DB | EMPLOYEES | TITLE | NVARCHAR(30) | |
| DIM_EMPLOYEE | BIRTH_DATE | DATE | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | BIRTH_DATE | DATE | |
| DIM_EMPLOYEE | HIRE_DATE | DATE | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | HIRE_DATE | DATE | SELECT dbo.Regions.RegionDescription FROM dbo.Regions WHERE dbo.Employees.RegionID = dbo.Regions.RegionID |
| DIM_EMPLOYEE | ADDRESS | NVARCHAR(60) | Dimension | 2 | OPERATIONAL_DB | EMPLOYEES | ADDRESS | NVARCHAR(60) | |
| DIM_EMPLOYEE | CITY | NVARCHAR(20) | Dimension | 2 | OPERATIONAL_DB | EMPLOYEES | CITY | NVARCHAR(20) | |
| | | | | 2 | | | | | |
| DIM_EMPLOYEE | REGION | NCHAR(50) | Dimension | | OPERATIONAL_DB | REGIONS | REGION_DESCRIPTION | NCHAR(50) | |
| DIM_EMPLOYEE | POSTAL_CODE | NVARCHAR(10) | Dimension | 2 | OPERATIONAL_DB | EMPLOYEES | POSTAL_CODE | NVARCHAR(10) | |
| DIM_EMPLOYEE | HOME_PHONE | NVARCHAR(24) | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | HOME_PHONE | NVARCHAR(24) | Surrogate key |
| DIM_EMPLOYEE | EXTENSION | NVARCHAR(4) | Dimension | 1 | OPERATIONAL_DB | EMPLOYEES | EXTENSION | NVARCHAR(4) | |
| DIM_EMPLOYEE | OBS | NTEXT | Dimension | N/A | OPERATIONAL_DB | EMPLOYEES | OBS | NTEXT | |
| DIM_PRODUCT | PRODUCT_KEY | NUMBER | Dimension | 1 | | | | NUMBER | |
| DIM_PRODUCT | PRODUCT_ID | NUMBER | Dimension | 1 | OPERATIONAL_DB | PRODUCTS | PRODUCT_ID | NUMBER | |
| DIM_PRODUCT | PRODUCT_NAME | NVARCHAR(40) | Dimension | 1 | OPERATIONAL_DB | PRODUCTS | PRODUCT_NAME | NVARCHAR(40) | Changed type of boolean from "BIT" to "yes/no" |
| DIM_PRODUCT | CATEGORY_NAME | NVARCHAR(15) | Dimension | 1 | OPERATIONAL_DB | CATEGORIES | CATEGORY_NAME | NVARCHAR(15) | |
| DIM_PRODUCT | CATEGORY_DESCRIPTION | NTEXT | Dimension | N/A | OPERATIONAL_DB | CATEGORIES | DESCRIPTION | NTEXT | |
| DIM_PRODUCT | QUANTITY_PER_UNIT | NCHAR(20) | Dimension | 2 | OPERATIONAL_DB | PRODUCTS | QUANTITY_PER_UNIT | NCHAR(20) | |
| DIM_PRODUCT | UNIT_PRICE | NUMBER | Dimension | 2 | OPERATIONAL_DB | PRODUCTS | UNIT_PRICE | NUMBER | |
| DIM_PRODUCT | UNITS_IN_STOCK | NUMBER | Dimension | 1 | OPERATIONAL_DB | PRODUCTS | UNITS_IN_STOCK | NUMBER | |
| DIM_PRODUCT | UNITS_ON_ORDER | NUMBER | Dimension | 1 | OPERATIONAL_DB | PRODUCTS | UNITS_ON_ORDER | | SELECT ProductKey, ProductID FROM DimProduct WHERE ExpiredDate IS NULL |
| DIM_PRODUCT | REORDER_LEVEL | NUMBER | Dimension | 1 | OPERATIONAL_DB | PRODUCTS | REORDER_LEVEL | | |
| DIM_PRODUCT | DISCONTINUED | BOOLEAN | Dimension | 2 | OPERATIONAL_DB | PRODUCTS | DISCONTINUED | BOOLEAN | |
| | | | | N/A | TRABPRAT_STAGINGAR EA | PRODUCTS | PRODUCT_ID | INT | |
| FACT_ORDERS | PRODUCT_KEY | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | CUSTOMERS | CUSTOMER_ID | NCHAR(5) | |
| FACT_ORDERS | CUSTOMER_KEY | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDERS | ORDER_DATE | DATE | Mapped OrderDate to FullDate on DimDate |
| FACT_ORDERS | ORDER_DATE_KEY | DATETIME | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDERS | REQUIRED_DATE | DATE | Mapped RequiredDate to FullDate on DimDate |
| FACT_ORDERS | REQUIRED_DATE_KEY | DATETIME | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDERS | SHIPPED_DATE | DATE | Mapped ShippedDate to FullDate on DimDate |
| FACT_ORDERS | SHIPPED_DATE_KEY | DATETIME | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDERS | EMPLOYEE_ID | INT | SELECT EmployeeKey, EmployeeID FROM DimEmployee WHERE ExpiredDate IS NULL |
| FACT_ORDERS | EMPLOYEE_KEY | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDER_DETAILS, ORDERS | ORDER_ID | NUMBER | |
| FACT_ORDERS | ORDER_ID | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDER_DETAILS | UNIT_PRICE | MONEY | |
| FACT_ORDERS | UNIT_PRICE | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDER_DETAILS | QUANTITY | NUMBER | |
| FACT_ORDERS | QUANTITY | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDER_DETAILS | DISCOUNT | FLOAT | |
| FACT_ORDERS | DISCOUNT | FLOAT | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | ORDER_DETAILS | TOTALPRICE | MONEY | SELECT (OrderDetails.Quantity*OrderDetails.UnitPrice) / (1-OrderDetails.Discount) AS totalPrice |
| FACT_ORDERS | TOTAL_PRICE | NUMBER | Fact | | OPERATIONAL_DB, TRABPRAT_STAGINGAR EA | | | | |

Conclusão

Todos os objetivos para o trabalho foram alcançados, tendo sido desenvolvido o data mart na sua totalidade e aplicados os processos de extração, transformação e carregamento.

O desenvolvimento deste trabalho permitiu a consolidação dos conhecimentos obtidos no decorrer das aulas teóricas e práticas, constituindo uma experiência que será certamente valiosa num momento de futuro contacto com a área.