

## Bikes & Bikes Data Mart

João Figueiredo, 1230194

João Araújo, 1200584

Curso: Mestrado em Engenharia Informática

Disciplina: Armazém de Dados (ARMDD)

Professor: Paulo Jorge Machado Oliveira

Data: Dezenbro, 2023

Ano académico: 2023/2024

# Contents

<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	Contexto: . . . . .	4
1.2	Motivação: . . . . .	4
1.3	Objetivos do Projeto: . . . . .	4
1.4	Estrutura do documento . . . . .	5
1.4.1	Capítulo 1: Introdução . . . . .	5
1.4.2	Capítulo 2: Modelo Relacional . . . . .	5
1.4.3	Capítulo 3: Arquitetura do DataMart . . . . .	5
1.4.4	Capítulo 4: Estruturas de Dados na Staging Area . . . . .	5
1.4.5	Capítulo 5: Modelo Dimensional . . . . .	5
1.4.6	Capítulo 6: Mapeamento Lógico . . . . .	5
<b>2</b>	<b>Modelo relacional</b>	<b>6</b>
2.1	Breve Explicação das Tabelas da base de Dados . . . . .	7
<b>3</b>	<b>Arquitetura do DataMart</b>	<b>11</b>
<b>4</b>	<b>Estruturas de dados a criar na Staging Area</b>	<b>13</b>
<b>5</b>	<b>Modelo Dimensional</b>	<b>14</b>
5.1	Explicação das dimensões e tabelas de factos. Enumeração dos atributos/tipos . . . . .	15
<b>6</b>	<b>Mapeamento Lógico</b>	<b>19</b>

# List of Figures

2.1	Modelo Relacional . . . . .	6
3.1	Arquitetura do DataMart . . . . .	12
5.1	Modelo Dimensional . . . . .	14

# List of Tables

# Chapter 1

## Introdução

### 1.1 Contexto:

O projeto proposto visa o desenvolvimento de um armazém de dados para a empresa Bikes & Bikes, uma empresa nacional que atua no ramo de produção e venda de bicicletas e acessórios. O sistema operacional atual da empresa possui limitações na análise de dados de vendas, especialmente para análises históricas. Para superar essas limitações e permitir análises mais robustas, propõe-se a criação de um armazém de dados eficiente.

O armazém de dados terá como base os dados de vendas a clientes, contemplando informações como produtos, clientes, vendas, entre outros. O objetivo é criar um ambiente que facilite análises dimensionais e ofereça flexibilidade para consultas no nível de granularidade mais elementar. Para isso, seguirá a metodologia de Kimball, uma abordagem reconhecida no desenvolvimento de data warehouses.

### 1.2 Motivação:

A motivação para a criação do armazém de dados reside na necessidade de melhorar as capacidades analíticas da empresa em relação às vendas. O sistema operacional atual não atende totalmente às necessidades de análises detalhadas e históricas. Com o novo armazém de dados, a Bikes & Bikes poderá realizar análises mais aprofundadas, identificar tendências, padrões de comportamento dos clientes e otimizar estratégias de vendas.

A empresa, ao lidar com clientes de diferentes países que realizam pedidos online, necessita de uma visão abrangente e integrada de suas operações. Além disso, as questões relacionadas à gestão de moedas, problemas de qualidade nos dados e a necessidade de eficiência no processo ETL (Extração, Transformação e Carregamento) justificam a implementação desse projeto.

### 1.3 Objetivos do Projeto:

1. Desenvolver um processo de análise dimensional seguindo a metodologia de Kimball para criar um esquema conceitual eficiente para o armazém de dados.
2. Realizar a extração, transformação, limpeza, integração e carregamento dos dados no armazém de dados, utilizando o Integration Services Project do Visual Studio.
3. Garantir que o processo de ETL seja eficiente, considerando a confidencialidade dos dados e o grande volume de informações.
4. Realizar um processo de Data Profiling para identificar problemas de qualidade nos dados, filtrando e armazenando registros afetados em uma área de preparação (Staging Area).
5. Eliminar redundâncias nos dados durante a fase de limpeza, especialmente registros duplicados ou aproximadamente iguais.
6. Projetar o carregamento dos dados de forma incremental para possibilitar atualizações eficientes no armazém de dados.

7. Configurar o Integration Services Project de forma flexível, evitando caminhos absolutos e permitindo a especificação de parâmetros, como nomes de servidores e bases de dados.

Ao atingir esses objetivos, a empresa Bikes & Bikes estará mais bem equipada para realizar análises detalhadas e extrair insights valiosos de suas operações de vendas. O novo armazém de dados proporcionará uma base sólida para futuras análises e melhorias estratégicas.

## **1.4 Estrutura do documento**

Este documento está organizado em diversos capítulos, cada um abordando uma etapa específica do desenvolvimento do armazém de dados para a empresa Bikes & Bikes. A seguir, apresentamos uma breve descrição do conteúdo de cada capítulo:

### **1.4.1 Capítulo 1: Introdução**

Neste capítulo introdutório, são apresentados os objetivos do projeto e a motivação para o desenvolvimento do armazém de dados. Explora-se a importância de analisar os dados de vendas da empresa Bikes & Bikes para aprimorar as capacidades analíticas.

### **1.4.2 Capítulo 2: Modelo Relacional**

Aqui, será detalhado o modelo relacional dos dados existentes no sistema operacional da empresa. Examina-se a estrutura das tabelas, suas relações e características fundamentais para compreender a fonte dos dados.

### **1.4.3 Capítulo 3: Arquitetura do DataMart**

Exploramos a arquitetura do DataMart, destacando seu papel no contexto do sistema de informações. Este capítulo fornecerá uma visão geral sobre o DataMart.

### **1.4.4 Capítulo 4: Estruturas de Dados na Staging Area**

Focando na fase de extração, transformação, limpeza, este capítulo discute as estruturas necessárias na Staging Area para lidar com a preparação dos dados antes do carregamento para o armazém de dados.

### **1.4.5 Capítulo 5: Modelo Dimensional**

Desenvolvimento do processo de análise dimensional, seguindo a metodologia de Kimball. Apresentamos o modelo dimensional do armazém de dados.

### **1.4.6 Capítulo 6: Mapeamento Lógico**

Este capítulo aborda o mapeamento lógico do armazém de dados, mostrando como o modelo relacional é traduzido para o modelo dimensional.

## Chapter 2

# Modelo relacional

Neste capítulo, abordamos o modelo relacional da nossa base de dados, Bikes&Bikes através da geração de scripts SQL no SQL Server, os quais definem o esquema da base de dados. Com a execução destes scripts, construímos uma representação visual do modelo relacional por meio de um diagrama abrangente. Essa abordagem prática não apenas traduz o design conceitual em uma forma executável, mas também proporciona uma compreensão clara da estrutura da base de dados e das relações entre as entidades. Este capítulo serve como base para explorarmos os atributos a manter que dimensões criar no modelo dimensional.

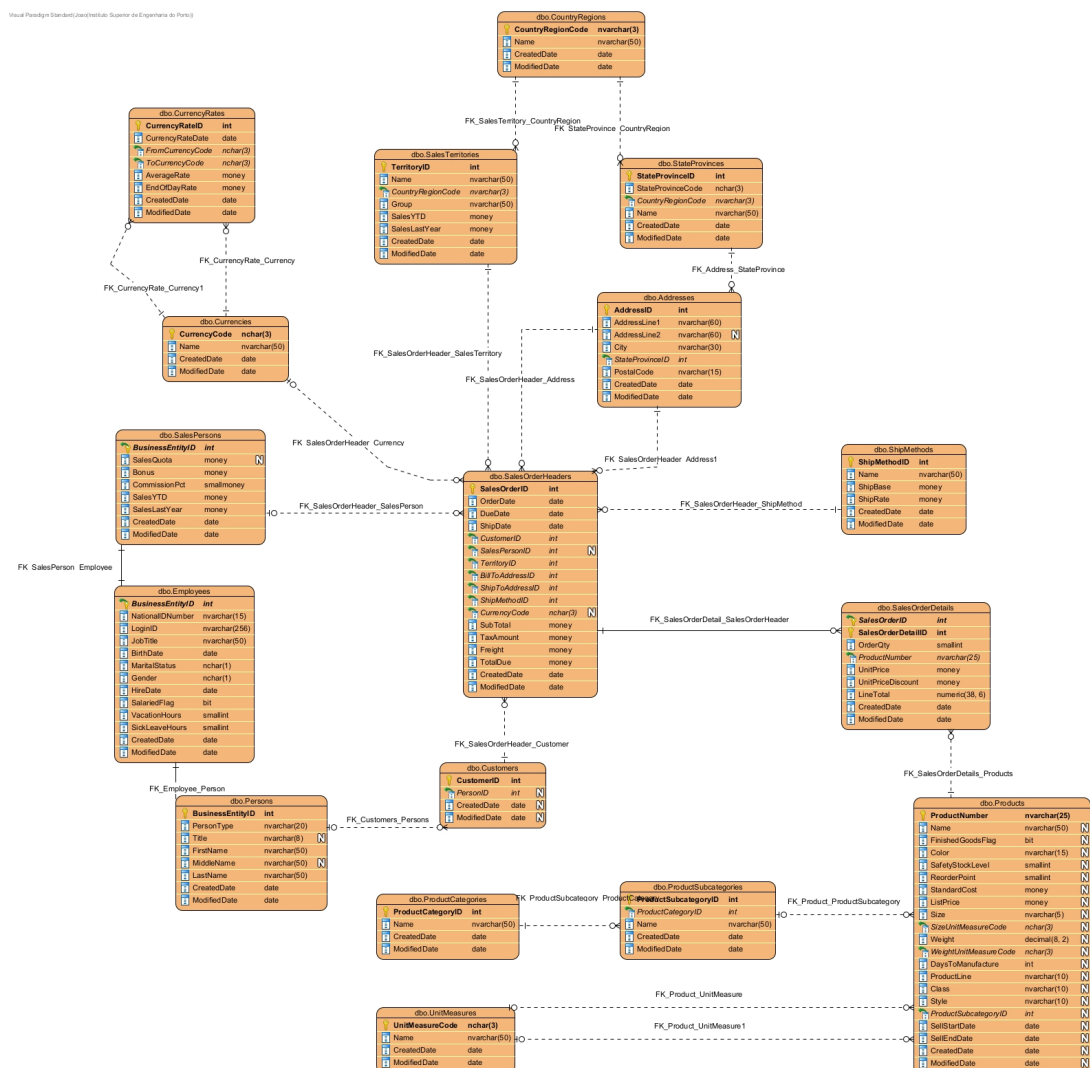


Figure 2.1: Modelo Relacional

## 2.1 Breve Explicação das Tabelas da base de Dados

### 1. Addresses:

- **AddressID:** Identificador único do endereço.
- **AddressLine1:** Linha de endereço 1.
- **AddressLine2:** Linha de endereço 2.
- **City:** Cidade.
- **StateProvinceID:** Identificador único da província/estado.
- **PostalCode:** Código postal.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

### 2. CountryRegions:

- **CountryRegionCode:** Código do país/região.
- **Name:** Nome do país/região.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

### 3. SalesOrderDetails:

- **SalesOrderID:** ID único do pedido de venda associado.
- **SalesOrderDetailID:** Identificador único do detalhe do pedido de venda.
- **OrderQty:** Quantidade do produto no pedido.
- **ProductNumber:** Número único do produto associado.
- **UnitPrice:** Preço unitário do produto.
- **UnitPriceDiscount:** Desconto no preço unitário.
- **LineTotal:** Total do pedido.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

### 4. SalesOrderHeaders:

- **SalesOrderID:** ID único do pedido de venda.
- **OrderDate:** Data do pedido de venda.
- **DueDate:** Data de vencimento do pedido.
- **ShipDate:** Data de envio do pedido.
- **CustomerID:** ID único do cliente associado ao pedido.
- **SalesPersonID:** ID único do representante de vendas associado ao pedido (pode ser nulo).
- **TerritoryID:** ID único do território associado ao pedido.
- **BillToAddressID:** ID único do endereço de cobrança associado ao pedido.
- **ShipToAddressID:** ID único do endereço de entrega associado ao pedido.
- **ShipMethodID:** ID único do método de envio associado ao pedido.
- **CurrencyCode:** Código da moeda utilizada no pedido (pode ser nulo).
- **SubTotal:** Subtotal do pedido.
- **TaxAmount:** Valor do imposto.
- **Freight:** Valor do frete.
- **TotalDue:** Total devido.
- **CreatedDate:** Data de criação do registo.



- **ModifiedDate:** Data da última modificação do registo.

#### 5. SalesPersons:

- **BusinessEntityID:** ID único da entidade de negócios (chave estrangeira para Employees).
- **SalesQuota:** Cota de vendas.
- **Bonus:** Bónus concedido.
- **CommissionPct:** Percentual de comissão.
- **SalesYTD:** Vendas acumuladas do ano até a data.
- **SalesLast Year:** Vendas do ano anterior.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 6. SalesTerritories:

- **TerritoryID:** ID único do território.
- **Name:** Nome do território.
- **CountryRegionCode:** Código do país/região associado ao território.
- **Group:** Grupo de territórios.
- **SalesYTD:** Vendas acumuladas do ano até a data.
- **SalesLast Year:** Vendas do ano anterior.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 7. ShipMethods:

- **ShipMethodID:** ID único do método de envio.
- **Name:** Nome do método de envio.
- **ShipBase:** Valor base do envio.
- **ShipRate:** Taxa de envio.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 8. StateProvinces:

- **StateProvinceID:** ID único da província/estado.
- **StateProvinceCode:** Código da província/estado.
- **CountryRegionCode:** Código do país/região associado à província/estado.
- **Name:** Nome da província/estado.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 9. UnitMeasures:

- **UnitMeasureCode:** Código único da unidade de medida.
- **Name:** Nome da unidade de medida.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 10. Persons:

- **BusinessEntityID:** ID único da entidade de negócios.
- **PersonType:** Tipo de pessoa.
- **Title:** Título.

- **FirstName:** Primeiro nome.
- **MiddleName:** Nome do meio (pode ser nulo).
- **LastName:** Sobrenome.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 11. Employees:

- **BusinessEntityID:** ID único da entidade de negócios (chave estrangeira para Persons).
- **NationalIDNumber:** Número de identificação nacional.
- **LoginID:** ID de login.
- **JobTitle:** Cargo.
- **BirthDate:** Data de nascimento.
- **MaritalStatus:** Estado civil.
- **Gender:** Género.
- **HireDate:** Data de contratação.
- **SalariedFlag:** Indicador de salário (bit).
- **VacationHours:** Horas de férias.
- **SickLeaveHours:** Horas de licença médica.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 12. Customers:

- **CustomerID:** ID único do cliente.
- **PersonID:** ID único da pessoa associada ao cliente (pode ser nulo).
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 13. CurrencyRates:

- **CurrencyRateID:** ID único da taxa de câmbio.
- **CurrencyRateDate:** Data da taxa de câmbio.
- **FromCurrencyCode:** Código da moeda de origem.
- **ToCurrencyCode:** Código da moeda de destino.
- **AverageRate:** Taxa média de câmbio.
- **EndOfDayRate:** Taxa de câmbio do final do dia.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 14. Currencies:

- **CurrencyCode:** Código único da moeda.
- **Name:** Nome da moeda.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

#### 15. CountryRegions:

- **CountryRegionCode:** Código único do país/região.
- **Name:** Nome do país/região.
- **CreatedDate:** Data de criação do registo.

- **ModifiedDate:** Data da última modificação do registo.

16. **ProductCategories:**

- **ProductCategoryID:** ID único da categoria de produto.
- **Name:** Nome da categoria de produto.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

17. **ProductSubcategories:**

- **ProductSubcategoryID:** ID único da subcategoria de produto.
- **ProductCategoryID:** ID único da categoria de produto à qual a subcategoria pertence.
- **Name:** Nome da subcategoria de produto.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

18. **StateProvinces:**

- **StateProvinceID:** ID único da província/estado.
- **StateProvinceCode:** Código único da província/estado.
- **CountryRegionCode:** Código único do país/região associado à província/estado.
- **Name:** Nome da província/estado.
- **CreatedDate:** Data de criação do registo.
- **ModifiedDate:** Data da última modificação do registo.

## Chapter 3

# Arquitetura do DataMart

- **Sistemas de dados de origem**

Os sistemas de dados de origem referem-se aos sistemas e aplicações de onde se originam os dados brutos de uma organização. Esses sistemas são o ponto de partida dos dados, e os dados normalmente são recolhidos e armazenados nesses sistemas de origem antes de serem processados e integrados num Data Warehouse ou noutros ambientes analíticos.

- **ETL (Extract, Transform, Load)**

ETL significa Extrair, Transformar, Carregar e refere-se ao processo de extrair dados de sistemas de origem, transformá-los num formato utilizável e carregá-los num sistema de destino.

**Extrair:** Na fase de extração, os dados são recolhidos de base de dados, arquivos simples, etc. O processo de extração envolve a leitura dos dados de origem e a inserção de informações relevantes no sistema ETL.

**Transformar:** A fase de transformação envolve limpeza, validação e estruturação dos dados extraídos para atender aos requisitos do sistema de destino. As tarefas de transformação de dados podem incluir limpeza, validação e enriquecimento de dados e também a aplicação de regras de negócios.

**Carregar:** Na fase de carregamento, os dados transformados são carregados no sistema de destino. O carregamento pode envolver a inserção de novos registos e a atualização dos existentes. O objetivo é garantir que os dados no sistema de destino sejam precisos, consistentes e prontos para análise.

- **Staging Area**

A Staging Area é uma área de armazenamento intermediária onde os dados brutos dos sistemas de origem são inicialmente carregados antes de serem transformados e carregados no Data Mart. A Staging Area é um espaço de armazenamento temporário onde os dados passam por tarefas de pré-processamento, como limpeza, validação e transformação. Isso garante que os dados estejam num formato consistente e utilizável antes de serem carregados no Data Mart.

- **Data Mart**

Um Data Mart é um subconjunto de um Data Warehouse, normalmente focado numa linha de negócios de uma organização. Os Data Mart's são projetados para atender às necessidades específicas de um determinado grupo ou unidade de negócio. Contêm um subconjunto de dados do Data Warehouse, adaptado para atender aos requisitos de um grupo específico de utilizadores.

- **Ferramentas de apresentação ao utilizador final**

As ferramentas de apresentação ao utilizador final são aplicações ou plataformas projetadas para acessar e interagir com dados para fins de relatórios, análises e tomada de decisões. Essas ferramentas são componentes essenciais dos ecossistemas de Business Intelligence (BI) e de análise, o que permite que os utilizadores não técnicos explorem e visualizem os dados sem a necessidade de conhecimento técnico aprofundado. Essas ferramentas normalmente ligam-se a fontes de dados, como Data Warehouse's, bancos de dados ou outros repositórios de dados, para recuperar e apresentar informações de maneira significativa.

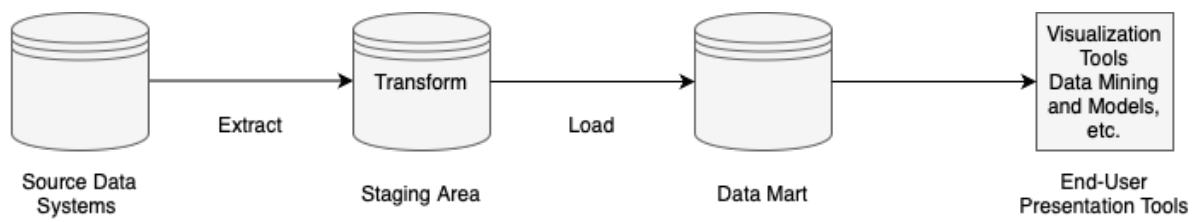


Figure 3.1: Arquitetura do DataMart

## Chapter 4

# Estruturas de dados a criar na Staging Area

No contexto de uma arquitetura de Data Warehouse, a Staging Area é um espaço de armazenamento intermédio onde os dados brutos dos sistemas de origem são armazenados temporariamente antes de serem transformados e carregados no Data Warehouse ou Data Mart's. Os dados na área de preparação estão na forma bruta e não processada e passam por diversas transformações antes de serem integrados ao repositório de dados principal. A criação de estruturas de dados na Staging Area é essencial para processar dados de forma eficiente.

- **Arquivos simples:** Criação de arquivos de texto simples onde os dados são armazenados em formato tabular com linhas e colunas. Os arquivos CSV são fáceis de criar, ler e manipular.
- **Tabelas Temporárias:** Criação de tabelas temporárias para armazenar dados brutos para auxiliar o tratamento dos dados, essas tabelas devem ser muito semelhantes à estrutura dos dados de origem.
- **Tabelas Particionadas:** Ao lidar com grandes conjuntos de dados, considera-se o particionamento de tabelas com base em certos critérios. Isso pode melhorar o desempenho da consulta durante o processo de transformação.
- **Duplicados:** Podem existir registos que têm os mesmos valores em todos os campos. Às vezes, os duplicados podem ter valores muito semelhantes para isso existe técnicas de correspondência difusa que podem ser aplicadas para identificar esses tipos de duplicados. Erros de digitação, erros de entrada manual ou falhas no sistema podem levar à criação de registos duplicados. Ao combinar dados de diferentes fontes, duplicados podem surgir se houver inconsistências na representação de dados ou se os registos não forem correspondidos adequadamente.
- **Operação Look Up:** O Look Up é uma operação comum em processos ETL onde os dados são recuperados de um conjunto de dados com base em certos critérios. As pesquisas são usadas para enriquecer ou aumentar os dados, incorporando informações adicionais de um conjunto de dados de referência.
- **DQP (Data Quality Process):** O Processo de Qualidade de Dados é um conjunto de atividades e procedimentos que garante que os dados dentro de uma organização atendam a determinados padrões de qualidade. Dados de alta qualidade são fundamentais para relatórios, análises e tomadas de decisões precisas.

## Chapter 5

# Modelo Dimensional

Este capítulo explora o Modelo Dimensional, uma técnica vital em data warehousing e business intelligence. Abordamos tabelas essenciais, destacando dimensões, factos e relacionamentos. A estrutura dimensionada facilita análises eficientes, proporcionando insights cruciais para a tomada de decisões. Detalhamos as dimensões e factos, promovendo uma compreensão clara da organização dos dados e seu impacto nas análises.

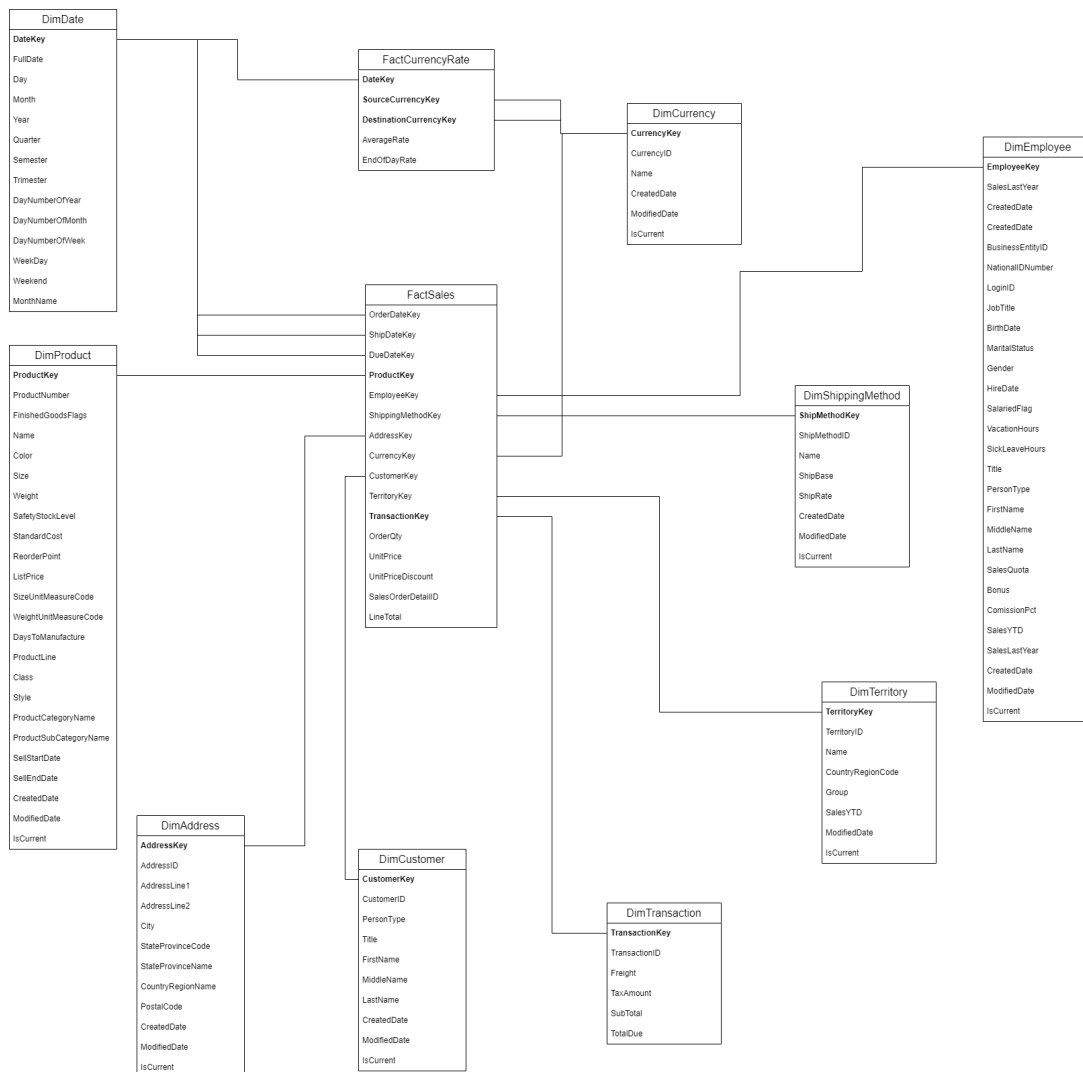


Figure 5.1: Modelo Dimensional

## 5.1 Explicação das dimensões e tabelas de factos. Enumeração dos atributos/tipos

- **DimDate**

- DateKey - Integer - Primary Key
- FullDate - Date
- Day - Integer
- Month - Integer
- Year - Integer
- Quarter - Integer
- Semester - Integer
- Trimester - Integer
- DayNumberOfYear - Integer
- DayNumberOfMonth - Integer
- DayNumberOfWeek - Integer
- WeekDay - nVarchar(20)
- Weekend - nVarchar(3)

- **DimProduct**

- ProductKey - Integer - Primary Key
- ProductNumber - Integer
- FinishedGoodsFlag - Bit
- Name - nvarchar(20)
- Color - nVarchar(15)
- Size - nVarchar(5)
- Weight - Decimal(8, 2)
- SafetyStockLevel - Smallint
- StandardCost - Money
- ReorderPoint - Smallint
- ListPrice - Money
- SizeUnitMeasureName - nchar(3)
- WeightUnitMeasureName - nchar(3)
- DaysToManufacture - Integer
- ProductLine - nVarchar(10)
- Class - nVarchar(10)
- Style - nVarchar(10)
- ProductCategoryName - nVarchar(50)
- ProductSubCategoryName - nVarchar(50)
- SellStartDate - Date
- SellEndDate - Date
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nVarchar(3)

- **DimCustomer**

- CustomerKey - Integer - Primary Key



- CustomerID - Integer
- PersonType - nvarchar(20)
- Title - nvarchar(8)
- FirstName - nvarchar(50)
- MiddleName - nvarchar(50)
- LastName - nvarchar(50)
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nvarchar(3)

• **DimEmployee**

- EmployeeKey - Integer - Primary Key
- BusinessEntityID - Integer
- NationalIDNumber - Integer
- LoginID - Integer
- JobTitle - nvarchar(50)
- BirthDate - Date
- MaritalStatus - nchar(1)
- Gender - nchar(1)
- HireDate - Date
- SalariedFlag - Bit
- VacationHours - Smallint
- SickLeaveHours - Smallint
- Title - nvarchar(8)
- PersonType - nvarchar(20)
- FirstName - nvarchar(50)
- MiddleName - nvarchar(50)
- LastName - nvarchar(50)
- SalesQuota - Money
- Bonus - Money
- CommissionPct - Smallmoney
- SalesYTD - Money
- SalesLastYear - Money
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nvarchar(3)

• **DimAddress**

- AddressKey - Integer - Primary Key
- AddressID - Integer
- AddressLine1 - nvarchar(60)
- AddressLine2 - nvarchar(60)
- City - nvarchar(30)
- StateProvinceCode - Integer
- StateProvinceName - nvarchar(50)
- CountryRegionName - nvarchar(50)

- PostalCode - nvarchar(15)
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nvarchar(3)

- **DimTerritory**

- TerritoryKey - Integer - Primary Key
- TerritoryID - Integer
- Name - nvarchar(50)
- CountryRegionName - nvarchar(50)
- Group - nvarchar(50)
- SalesYTD - Money
- SalesLastYear - Money
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nvarchar(3)

- **DimCurrency**

- CurrencyKey - Integer - Primary Key
- CurrencyID - Integer
- Name - nvarchar(50)
- CreatedDate - Date
- ModifiedDate - Date

- **DimShippingMethod**

- ShipMethodKey - Integer - Primary Key
- ShipMethodID - Integer
- Name - nvarchar(50)
- ShipBase - Money
- ShipRate - Money
- CreatedDate - Date
- ModifiedDate - Date
- IsCurrent - nvarchar(3)

- **DimTransaction**

- TransactionKey - Integer - Primary Key
- Freight - Money
- TaxAmount - Money
- SubTotal - Money
- TotalDue - Money
- SalesOrderId - Integer

- **FactSales**

- OrderDateKey - Integer - Primary Key
- ShipDateKey - Integer
- DueDateKey - Integer
- CustomerKey - Integer - Primary Key

- EmployeeKey - Integer
- ShippingMethodKey - Integer
- AddressKey - Integer
- ProductKey - Integer - Primary Key
- TransactionKey
- CurrencyKey - Integer
- TerritoryKey - Integer
- OrderQty - Smallint
- UnitPrice - Money
- UnitPriceDiscount - Money
- LineTotal - Numeric(38, 6)
- SalesOrderDetailID - Integer

- **FactCurrencyRate**

- DateKey - Integer - Primary Key
- SourceCurrencyKey - Integer - Primary Key
- DestinationCurrencyKey - Integer - Primary Key
- AverageRate - Money
- EndOfDayRate - Money

## Chapter 6

# Mapeamento Lógico

Neste capítulo, apresentamos o Mapeamento Lógico realizado como parte integrante do processo de design do banco de dados. O mapeamento proporciona uma visão abrangente das transformações e correspondências entre as fontes de dados e as tabelas de destino. Utilizando uma abordagem estruturada e eficiente, o mapeamento foi elaborado utilizando o Microsoft Excel, aproveitando suas capacidades de células, linhas e colunas para fornecer uma representação clara e detalhada para a inclusão no relatório.

O mapeamento lógico é uma etapa crucial no desenvolvimento do banco de dados, pois estabelece a base para a implementação física posterior. Cada coluna desempenha um papel significativo na definição da estrutura final do banco de dados. A seguir, são apresentadas as colunas utilizadas no mapeamento:

- **Target**

- Table Name
- Column Name
- Data Type
- Table Type
- SCD Type
- DataBase Name

- **Source**

- Table Name
- Column Name
- Data Type

- **Transformation**

- Incorpora as transformações específicas aplicadas durante o processo.

Este mapeamento proporciona uma visão detalhada da correspondência entre as fontes de dados e as tabelas de destino, facilitando a compreensão e implementação eficiente do banco de dados.