



## **Desenvolvimento Data Mart**

1100560 – Carlos Martinez

1100644 – Nuno Cardoso

2014 / 2015

***ARPAD***

**Mestrado Engenharia Informática**



## Índice

---

Índice .....	2
Modelo de Kimball.....	6
1. Área de Negócio .....	6
2. Granularidade.....	6
3. Dimensões .....	6
4. Medidas/Factos.....	6
Modelo Dimensional.....	7
Processo ETL .....	8
Scripts SQL .....	12
Scripts Criação de Bases de dados .....	12
Script Criação de partições .....	12
Scripts Criação de tabelas Staging Area .....	13
Tabela Armazéns.....	13
Tabela Categorias .....	13
Tabela ClientesArriving .....	14
Tabela ClientesDuplicatesBetweenSource .....	15
Tabela DuplicatesInsideSource .....	16
Tabela ClientesExisting.....	17
Tabela ClientesNew.....	18
Tabela Documentos Faturação.....	19
Tabela Documentos Faturação Linhas .....	20
Tabela Famílias.....	20
Tabela Fornecedores.....	21
Tabela Funcionários.....	22
Tabela Produtos .....	22
Tabela Vendedores.....	23

Tabela Zona .....	23
Scripts Loading Dados Para Staging Area .....	23
Scripts Criação Dimensões e Tabelas de Factos .....	23
Dimensão Cliente .....	24
Dimensão Data.....	25
Dimensão DataMes.....	25
Dimensão Fornecedor.....	26
Dimensão Funcionário.....	27
Dimensão Moeda.....	28
Dimensão Produto .....	28
Dimensão Tempo .....	29
Dimensão Zona .....	29
Dimensão “Junk” Faturação.....	29
Tabela de Factos Taxa de Câmbios .....	30
Tabela de Factos Faturação Linhas.....	30
Tabela de Factos Faturação Mês .....	32
Scripts para a criação de Índices.....	32
Índices da dimensão Data.....	33
Índices da dimensão Tempo.....	33
Índices da dimensão Funcionário .....	33
Índices da dimensão Fornecedor .....	33
Índices da dimensão Cliente.....	33
Índices da dimensão Produto.....	33
Índices da tabela de factos Faturação Linhas.....	34
Índices da tabela de factos Faturação Linhas Produto Mês .....	34
Índices da tabela de factos Faturação Linhas Cliente Mês .....	35
Scripts para a criação de chaves primárias e estrangeiras .....	35

Chaves da dimensão Funcionário .....	35
Chaves da dimensão Fornecedor .....	36
Chaves da dimensão Cliente .....	37
Chaves da dimensão Produto .....	38
Chaves da tabela de factos Faturação Linhas.....	39
Chaves da tabela de factos Faturação Linhas Produto Mês .....	39
Análise 1.....	41
Análise 2.....	42
Análise 3.....	42
Análise 4.....	43
Análise 5.....	43
Análise 6.....	44
Análise 7.....	44
Análise 8.....	45
Análise 9.....	46
Análise 10.....	46
Análise 11.....	47
Análise 12.....	47
Análise 13.....	48
Análise 14.....	48
Análise 15.....	49
Análise 16.....	49
Análise 17.....	50
Análise 18.....	50
Análise 19.....	51
Análise 20.....	51
Opções .....	52

Novos campos.....	52
Taxas de câmbios.....	52
Surrogate Keys .....	52
Dimensão “lixo” .....	52
Junção de tabelas.....	53
Outras opções .....	53
Melhoramentos possíveis.....	54

## Modelo de Kimball

---

### 1. Área de Negócio

Vendas efectuadas;

### 2. Granularidade

Vendas efectuadas por dia, por Cliente, por Funcionário, por Vendedor, por Produto, por Zona e por Moeda;

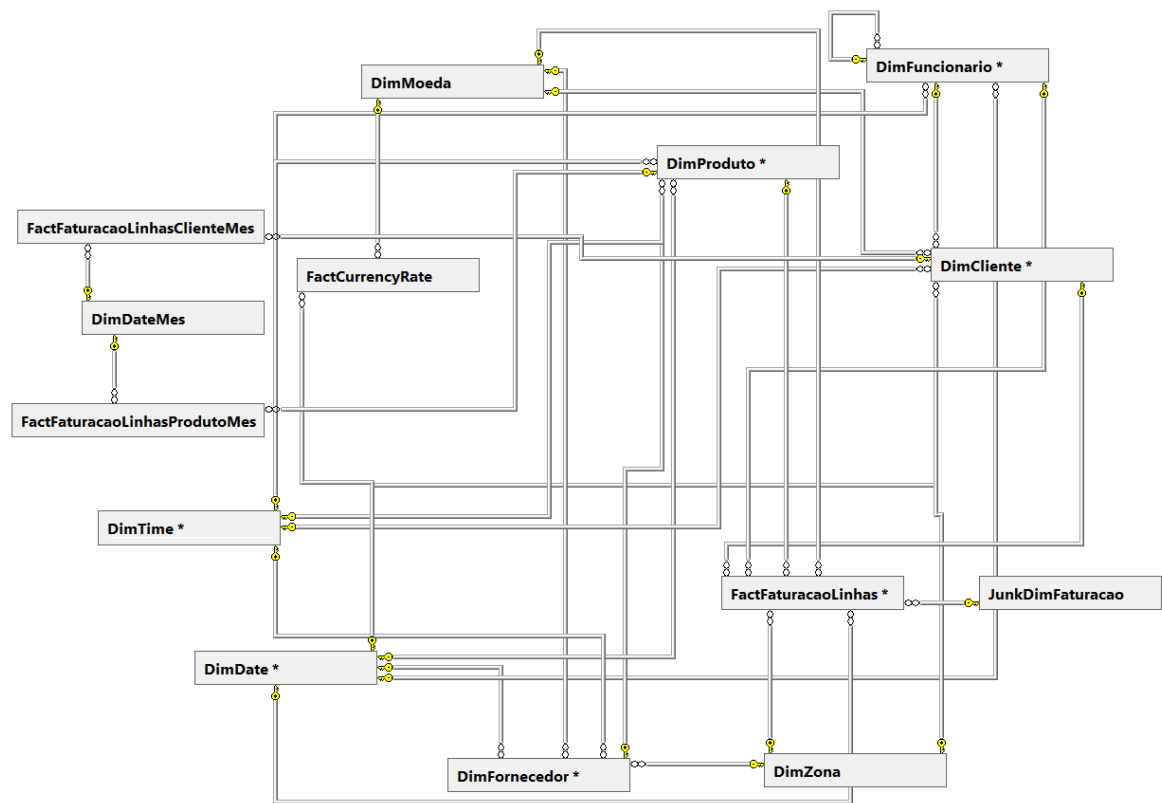
### 3. Dimensões

Time, Date, DateMes, Produto, Fornecedor, Zona, Cliente, Funcionario, JunkFaturacao, Moeda;

### 4. Medidas/Factos

1. FactFaturacaoLinhas, FactFaturacaoProdutoMes, FactFaturacaoClienteMes:  
Quantidade, TaxaIVA, PrecoUnidade, FreteUnidade, ValorIlíquido, ValorDesconto,  
PrecoCusto, PrecoCustoPonderado, TotalIlíquidoProdutos, TotalIVA, TotalFrete,  
TotalFinal, TotalCusto, Lucro
2. FactCurrencyRate: FatorCambio

# Modelo Dimensional



## Processo ETL

---

### *Transformação/Limpeza*

#### **Uniformizar ('Standardizing')**

Foram aplicadas regras de uniformização em vários atributos para um formato consistente e preferível:

- Campo "CentroCusto" – Continha valores abreviados que passaram, respetivamente, de "COM", "ADM" e "DIR" para "Comercial", "Administrativo" e "Direção".
- Campo "Telefone", "Telemovel" e "Fax" – Continham campos que tinham espaços entre números e outros não, os espaços foram removidos; os indicativos também foram removidos para tornar o campo totalmente constante.
- Campo "NumeroContribuinte" – Foi criado um script para validar o número de contribuinte consoante os requisitos necessários para um contribuinte aceitável. No entanto, como todos os números no sistema operacional são fictícios e portanto, inválidos, o script que tratava dessa tarefa foi removido e todos os números são aceites.
- Campos referentes a quantidades e stocks – Foram "truncados" para não serem representados com casas decimais porque todos os dados no sistema operacional referentes a quantidades e stocks todas as casas decimais a zero e tanto quantidades como stocks devem ser representados por inteiros.
- Campo "Quantidade" – Na tabela "DocumentosFaturacaoLinhas", todos os registos no campo aparentam estar divididos por 10, por isso no processo de limpeza foram multiplicados por 10 para representarem o valor corretamente.
- Campo "CategoriaFamilia", "ClassificacaoFamilia", "Morada", "CodigoPostal", "EmailEmpresa", "Localidade", "Tipo", "FuncaoContato", "Email", "Nacionalidade", "Distrito", "Classe" – Estes campos estavam ou a "NULL", ou vazios, ou apenas com um espaço, em todos os casos, foram substituídos pela palavra "Unknown".

#### **Deteção duplicados ('Matching')**



Como os dados carregados para o *Data Mart* vêm de várias fontes, o processo ETL criado contém um fluxo de dados que faz a deteção de duplicados na tabela clientes. No limite os dados podem-se encontrar duplicados no mesmo sistema operacional. O fluxo de dados criado utiliza a tabela ArrivingClientes como fonte de dados para detetar duplicados exatos, duplicados entre fontes e duplicados na mesma fonte e guarda na respetiva tabela da *Staging Area* para o efeito. Os clientes que não são duplicados são guardados na tabela ClientesNew para posterior *loading* na dimensão Clientes.

### *Carregamento dos dados*

Antes de os dados serem carregados da Staging Area para o Data Mart, todos os dados que apresentam problemas que não podem ser corrigidos, são enviados para tabelas especiais que são denominadas de DQP. Estas tabelas são iguais às suas tabelas correspondentes na Staging Area com a diferença de que têm mais um campo “varchar” chamado DQP que guarda a razão de certo registo não ter entrado no Data Mart. A nível de tabelas de dimensão, todos os campos que apresentam problemas foram corrigidos e portanto, apenas quando não é possível encontrar dados referentes a chaves estrangeiras é que o registo é passado para a respetiva tabela de DQP; A nível de tabelas de factos, todos os problemas com chaves estrangeiras também são passados para as tabelas DQP e valores incorretos encontrados em medidas são postos nas tabelas de DQP (valores a 0).

### *Optimizações*

#### **Particionamento**

Foi feito particionamento à tabela de factos “FactFaturacaoLinhas” uma vez que esta tem um número muito grande de dados. Para melhorar a performance das suas queries foram criadas partições horizontais segmentadas por *range* de anos. Mais precisamente foram criadas as partições com os seguintes *ranges*:

- 1900-2000;
- 2001-2005;
- 2006-2010;
- 2011-2013;
- 2014-2015.

```
CREATE PARTITION FUNCTION AnoRangePFN(int) AS  
RANGE LEFT FOR VALUES  
(2000, 2005, 2010, 2013, 2015)
```

O script de criação das partições é feito num script em C# e guardado numa variável do projeto para ser executado.

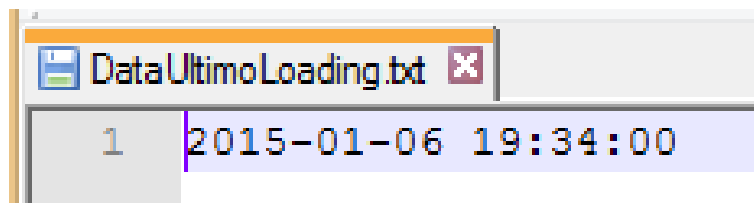
## Indexes

Os *indexes* melhoram o tempo de resposta das queries mas reduzem em muito a performance durante a inserção e edição dos dados. Desta forma, para a performance durante os processos ETL não ser prejudicada os indexes são removidos temporariamente. Logo após o seu loading para a Dim correspondente os indexes são criados novamente.

## Carregamento incremental

Para a extração existem duas opções, extração estática ou extração incremental, a primeira faz a extração de todos os dados existentes no sistema operacional num dado momento para a staging area. A extração incremental apenas extrai os dados que foram alterados desde o último processo ETL, caso seja a primeira execução extrai todos os dados sistema operacional. Para retirar apenas os dados desde o último carregamento é possível usar os logs da DB ou então usar “*Audit columns*” que por norma são atualizados pela aplicação front-end ou por triggers aquando de atualizações e inserções e contém a data de última alteração.

Foi utilizado o carregamento incremental com a utilização das “*Audit columns*” disponibilizadas pelo sistema operacional para melhorar em muito a performance do processo ETL. Em apenas algumas tabelas pequenas foi utilizada a extração estática, tal como nas tabelas zona, categoria e armazens.



O ficheiro “DataUltimoLoading.txt” é utilizado para guardar a data e hora do último loading dos dados do sistema operacional para a *staging area*. Esta data é posteriormente usada numa variável para extrair apenas os dados actualizados após o último loading.


## Agregação

Tendo em conta o elevado número de dados e o facto de serem feitas várias análises ao nível do mês foram feitas as seguintes agregações à tabela de factos “FactFaturacaoLinhas”:

1. FactFaturacaoLinhasProdutoMes - Linhas de Faturação por mês e por produto.

2. FactFaturacaoLinhasClienteMes - Linhas de Faturação por mês e por cliente.

Para a criação destas tabelas agregadas foi necessário adicionar uma nova dimensão date (DimDateMes) agregada ao mês.

DimDateMes	
	DateKey
	Year
	Month
	MonthName
	Semester
	Trimester
	Quarter

## Scripts SQL

---

### Scripts Criação de Bases de dados

O valor do parâmetro utilizado corresponde à variável que contém o nome da BD.

```
EXEC('IF NOT EXISTS (SELECT name FROM sys.databases where name = '' +  
@DBName + '') CREATE DATABASE ' + @DBName)
```

### Script Criação de partições

Criado em script C#

```
string DB = this.Variables.DataMartDBName;  
string path = this.Variables.PackagePath;  
string SQLCommand = "ALTER DATABASE " + DB + " ADD FILEGROUP  
partition_1900_2000 ALTER DATABASE " + DB + " ADD FILEGROUP  
partition_2001_2005 ALTER DATABASE " + DB + " "  
+ " ADD FILEGROUP partition_2006_2010 ALTER DATABASE " + DB + " ADD  
FILEGROUP partition_2011_2013 ALTER DATABASE " + DB + " ADD FILEGROUP  
partition_2014_2015"  
+ " ALTER DATABASE " + DB + " ADD FILE (NAME = data_1900_2000, FILENAME  
= '' + path + "\\Files\\Partition_1900_2000.ndf', SIZE = 50MB, MAXSIZE =  
5000MB, FILEGROWTH = 200MB) TO FILEGROUP partition_1900_2000;"  
+ " ALTER DATABASE " + DB + " ADD FILE (NAME = data_2001_2005, FILENAME  
= '' + path + "\\Files\\Partition_2001_2005.ndf', SIZE = 50MB, MAXSIZE =  
3000MB, FILEGROWTH = 100MB) TO FILEGROUP partition_2001_2005;"  
+ " ALTER DATABASE " + DB + " ADD FILE (NAME = data_2006_2010, FILENAME  
= '' + path + "\\Files\\Partition_2006_2010.ndf', SIZE = 50MB, MAXSIZE =  
3000MB, FILEGROWTH = 100MB) TO FILEGROUP partition_2006_2010;"  
+ " ALTER DATABASE " + DB + " ADD FILE (NAME = data_2011_2013, FILENAME  
= '' + path + "\\Files\\Partition_2011_2013.ndf', SIZE = 50MB, MAXSIZE =  
2000MB, FILEGROWTH = 100MB) TO FILEGROUP partition_2011_2013;"  
+ " ALTER DATABASE " + DB + " ADD FILE (NAME = data_2014_2015, FILENAME  
= '' + path + "\\Files\\Partition_2014_2015.ndf', SIZE = 50MB, MAXSIZE =  
2000MB, FILEGROWTH = 100MB) TO FILEGROUP partition_2014_2015;"  
+ " CREATE PARTITION FUNCTION AnoRangePFN(int) AS RANGE LEFT FOR VALUES  
(2000, 2005, 2010, 2013, 2015) CREATE PARTITION SCHEME RangeDateScheme AS  
PARTITION AnoRangePFN TO "  
+ "  
([partition_1900_2000],[partition_2001_2005],[partition_2006_2010],[partition  
_2011_2013],[partition_2014_2015], [PRIMARY] )";  
  
this.Variables.PartitionCommand = SQLCommand;
```

## Scripts Criação de tabelas Staging Area

Nesta secção estão todos os scripts para a criação de todas as tabelas necessárias para a Staging Area. Para além dos scripts que vão ser apresentados, existem ainda os scripts de DQP que são usados para guardar registos com diversos problemas, as tabelas de DQP são apenas uma cópia da tabela original mais o campo (varchar) "DQP". As tabelas para as quais foram criadas tabelas de DQP criadas são as seguintes: Clientes, Faturação, Faturação Linhas, Faturação Linhas Mês, Fornecedores, Funcionários e Produtos.

### Tabela Armazéns

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Armazens')
CREATE TABLE [dbo].[Armazens](
    [codigo] [numeric](5, 0) NOT NULL,
    [armazem] [varchar](50) NULL)
ELSE
    TRUNCATE TABLE Armazens
```

### Tabela Categorias

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Categorias')
CREATE TABLE [dbo].[Categorias](
    [Codigo] [int] NOT NULL,
    [Categoria] [varchar](35) NULL)
ELSE
    TRUNCATE TABLE Categorias
```

## Tabela ClientesArriving

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'ClientesArriving')
CREATE TABLE [dbo].[ClientesArriving](
    [nome] [varchar](55) NOT NULL,
    [numero] [numeric](10, 0) NOT NULL,
    [ncontribuinte] [varchar](20) NOT NULL,
    [saldo] [numeric](19, 6) NOT NULL,
    [moeda] [varchar](11) NOT NULL,
    [fax] [varchar](60) NOT NULL,
    [telefone] [varchar](60) NOT NULL,
    [contacto] [varchar](30) NOT NULL,
    [morada] [varchar](55) NOT NULL,
    [localidade] [varchar](43) NOT NULL,
    [codpostal] [varchar](45) NOT NULL,
    [zona] [numeric](5, 0) NULL,
    [tipo] [varchar](20) NOT NULL,
    [vendedor] [numeric](4, 0) NOT NULL,
    [vencimento] [numeric](3, 0) NOT NULL,
    [plafond] [numeric](19, 6) NOT NULL,
    [nib] [varchar](28) NOT NULL,
    [segmento] [varchar](25) NOT NULL,
    [email] [varchar](45) NOT NULL,
    [centrocusto] [varchar](20) NOT NULL,
    [tlmv1] [varchar](45) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL)
ELSE
    TRUNCATE TABLE ClientesArriving
```

## Tabela ClientesDuplicatesBetweenSource

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'ClientesDuplicatesBetweenSources')
CREATE TABLE [dbo].[ClientesDuplicatesBetweenSources](
[nome (Input)] [varchar](55) NOT NULL, [ncontribuinte (Input)] [varchar](20) NOT NULL,
[saldo (Input)] [numeric](19, 6) NOT NULL, [moeda (Input)] [varchar](11) NOT NULL,
[fax (Input)] [varchar](60) NOT NULL, [telefone (Input)] [varchar](60) NOT NULL,
[contacto (Input)] [varchar](30) NOT NULL, [morada (Input)] [varchar](55) NOT NULL,
[localidade (Input)] [varchar](43) NOT NULL, [codpostal (Input)] [varchar](45) NOT NULL,
[zona (Input)] [numeric](5, 0) NULL, [tipo (Input)] [varchar](20) NOT NULL,
[vendedor (Input)] [numeric](4, 0) NOT NULL,
[vencimento (Input)] [numeric](3, 0) NOT NULL, [plafond (Input)] [numeric](19, 6) NOT NULL,
[nib (Input)] [varchar](28) NOT NULL, [segmento (Input)] [varchar](25) NOT NULL,
[email (Input)] [varchar](45) NOT NULL, [centrocusto (Input)] [varchar](20) NOT NULL,
[tlmvl (Input)] [varchar](45) NOT NULL, [data_criacao (Input)] [datetime] NOT NULL,
[hora_criacao (Input)] [varchar](8) NOT NULL,
[data_ult_gravacao (Input)] [datetime] NOT NULL,
[hora_ult_gravacao (Input)] [varchar](8) NOT NULL,

[nome (Matched)] [varchar](55) NOT NULL,
[ncontribuinte (Matched)] [varchar](20) NOT NULL,
[saldo (Matched)] [numeric](19, 6) NOT NULL,
[moeda (Matched)] [varchar](11) NOT NULL,
[fax (Matched)] [varchar](60) NOT NULL,
[telefone (Matched)] [varchar](60) NOT NULL,
[contacto (Matched)] [varchar](30) NOT NULL,
[morada (Matched)] [varchar](55) NOT NULL,
[localidade (Matched)] [varchar](43) NOT NULL,
[codpostal (Matched)] [varchar](45) NOT NULL,
[zona (Matched)] [numeric](5, 0) NULL,
[tipo (Matched)] [varchar](20) NOT NULL,
[vendedor (Matched)] [numeric](4, 0) NOT NULL,
[vencimento (Matched)] [numeric](3, 0) NOT NULL,
[plafond (Matched)] [numeric](19, 6) NOT NULL,
[nib (Matched)] [varchar](28) NOT NULL,
[segmento (Matched)] [varchar](25) NOT NULL,
[email (Matched)] [varchar](45) NOT NULL,
[centrocusto (Matched)] [varchar](20) NOT NULL,
[tlmvl (Matched)] [varchar](45) NOT NULL,
[data_criacao (Matched)] [datetime] NOT NULL,
[hora_criacao (Matched)] [varchar](8) NOT NULL,
[data_ult_gravacao (Matched)] [datetime] NOT NULL,
[hora_ult_gravacao (Matched)] [varchar](8) NOT NULL,

[_Similarity][real] NULL, [_Similarity_Nome][real] NULL,
[_Similarity_NContribuinte][real] NULL, [_Similarity_Telefone][real] NULL,
[_Similarity_Contacto][real] NULL, [_Similarity_Morada][real] NULL,
[_Similarity_Localidade][real] NULL, [_Similarity_CodPostal][real] NULL,
[_Similarity_Zona][real] NULL, [_Similarity_Tipo][real] NULL,
[_Similarity_Vendedor][real] NULL, [_Similarity_NIB][real] NULL,
[_Similarity_Segmento][real] NULL, [_Similarity_Email][real] NULL,
[_Similarity_CentroCusto][real] NULL, [_Similarity_Telemovel][real] NULL,
[_Similarity_Fax][real] NULL, [_Similarity_Moeda][real] NULL)
ELSE
TRUNCATE TABLE ClientesDuplicatesBetweenSources
```

## Tabela DuplicatesInsideSource

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name =
'ClientesDuplicatesInsideSources')
CREATE TABLE [dbo].ClientesDuplicatesInsideSources(
[_key_in] [int] NULL,      [_key_out] [int] NULL,
[nome] [varchar](55) NOT NULL,      [ncontribuinte] [varchar](20) NOT NULL,
[saldo] [numeric](19, 6) NOT NULL,      [moeda] [varchar](11) NOT NULL,
[fax] [varchar](60) NOT NULL,      [telefone] [varchar](60) NOT NULL,
[contacto] [varchar](30) NOT NULL,      [morada] [varchar](55) NOT NULL,
[localidade] [varchar](43) NOT NULL,      [codpostal] [varchar](45) NOT NULL,
[zona] [numeric](5, 0) NULL,      [tipo] [varchar](20) NOT NULL,
[vendedor] [numeric](4, 0) NOT NULL,      [vencimento] [numeric](3, 0) NOT NULL,
[plafond] [numeric](19, 6) NOT NULL,      [nib] [varchar](28) NOT NULL,
[segmento] [varchar](25) NOT NULL,      [email] [varchar](45) NOT NULL,
[centrocusto] [varchar](20) NOT NULL,      [tlmvl] [varchar](45) NOT NULL,
[data_criacao] [datetime] NOT NULL,      [hora_criacao] [varchar](8) NOT NULL,
[data_ult_gravacao] [datetime] NOT NULL, [hora_ult_gravacao] [varchar](8) NOT NULL,

[nome_clean] [varchar](55) NOT NULL,
[ncontribuinte_clean] [varchar](20) NOT NULL,
[moeda_clean] [varchar](11) NOT NULL,
[fax_clean] [varchar](60) NOT NULL,
[telefone_clean] [varchar](60) NOT NULL,
[contacto_clean] [varchar](30) NOT NULL,
[morada_clean] [varchar](55) NOT NULL,
[localidade_clean] [varchar](43) NOT NULL,
[codpostal_clean] [varchar](45) NOT NULL,
[tipo_clean] [varchar](20) NOT NULL,
[nib_clean] [varchar](28) NOT NULL,
[segmento_clean] [varchar](25) NOT NULL,
[email_clean] [varchar](45) NOT NULL,
[centrocusto_clean] [varchar](20) NOT NULL,
[tlmvl_clean] [varchar](45) NOT NULL,

[_Similarity][real] NULL,
[_Similarity_Nome][real] NULL,
[_Similarity_NContribuinte][real] NULL,
[_Similarity_Telefone][real] NULL,
[_Similarity_Contacto][real] NULL,
[_Similarity_Morada][real] NULL,
[_Similarity_Localidade][real] NULL,
[_Similarity_CodPostal][real] NULL,
[_Similarity_Tipo][real] NULL,
[_Similarity_NIB][real] NULL,
_Similarity_Segmento [real] NULL,
_Similarity_Email[real] NULL,
_Similarity_CentroCusto[real] NULL,
_Similarity_Telemovel[real] NULL,
_Similarity_Fax[real] NULL,
_Similarity_Moeda[real] NULL
)
ELSE
    TRUNCATE TABLE ClientesDuplicatesInsideSources
```



## Tabela ClientesExisting

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'ClientesExisting')
CREATE TABLE [dbo].[ClientesExisting](
    [nome] [varchar](55) NOT NULL,
    [numero] [numeric](10, 0) NOT NULL,
    [ncontribuinte] [varchar](20) NOT NULL,
    [saldo] [numeric](19, 6) NOT NULL,
    [moeda] [varchar](11) NOT NULL,
    [fax] [varchar](60) NOT NULL,
    [telefone] [varchar](60) NOT NULL,
    [contacto] [varchar](30) NOT NULL,
    [morada] [varchar](55) NOT NULL,
    [localidade] [varchar](43) NOT NULL,
    [codpostal] [varchar](45) NOT NULL,
    [zona] [numeric](5, 0) NULL,
    [tipo] [varchar](20) NOT NULL,
    [vendedor] [numeric](4, 0) NOT NULL,
    [vencimento] [numeric](3, 0) NOT NULL,
    [plafond] [numeric](19, 6) NOT NULL,
    [nib] [varchar](28) NOT NULL,
    [segmento] [varchar](25) NOT NULL,
    [email] [varchar](45) NOT NULL,
    [centrocusto] [varchar](20) NOT NULL,
    [tlmvl] [varchar](45) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL
)
```

## Tabela ClientesNew

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'ClientesNew')
CREATE TABLE [dbo].[ClientesNew](
    [nome] [varchar](55) NOT NULL,
    [numero] [numeric](10, 0) NOT NULL,
    [ncontribuinte] [varchar](20) NOT NULL,
    [saldo] [numeric](19, 6) NOT NULL,
    [moeda] [varchar](11) NOT NULL,
    [fax] [varchar](60) NOT NULL,
    [telefone] [varchar](60) NOT NULL,
    [contacto] [varchar](30) NOT NULL,
    [morada] [varchar](55) NOT NULL,
    [localidade] [varchar](43) NOT NULL,
    [codpostal] [varchar](45) NOT NULL,
    [zona] [numeric](5, 0) NULL,
    [tipo] [varchar](20) NOT NULL,
    [vendedor] [numeric](4, 0) NOT NULL,
    [vencimento] [numeric](3, 0) NOT NULL,
    [plafond] [numeric](19, 6) NOT NULL,
    [nib] [varchar](28) NOT NULL,
    [segmento] [varchar](25) NOT NULL,
    [email] [varchar](45) NOT NULL,
    [centrocusto] [varchar](20) NOT NULL,
    [tlmv1] [varchar](45) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL
)
ELSE
    TRUNCATE TABLE ClientesNew
```

## Tabela Documentos Faturação

```
IF NOT EXISTS (SELECT name FROM sys.tables where name =
'DocumentosFaturacao')
CREATE TABLE [dbo].[DocumentosFaturacao](
    [ftstamp] [char](25) NOT NULL,
    [nomedocumento] [varchar](20) NOT NULL,
    [anodocumento] [numeric](4, 0) NOT NULL,
    [numerodocumento] [numeric](10, 0) NOT NULL,
    [datadocumento] [datetime] NOT NULL,
    [cliente] [numeric](10, 0) NOT NULL,
    [funcionario] [numeric](6, 0) NOT NULL,
    [zona] [numeric](5, 0) NOT NULL,
    [vendedor] [numeric](4, 0) NOT NULL,
    [datapagamento] [datetime] NOT NULL,
    [moeda] [varchar](11) NOT NULL,
    [centrocosto] [varchar](20) NOT NULL,
    [segmento] [varchar](25) NOT NULL,
    [quantidadetotal] [numeric](15, 3) NOT NULL,
    [tipo] [varchar](20) NOT NULL,
    [totaliliquidoprods] [numeric](19, 6) NOT NULL,
    [totaldesconto] [numeric](19, 6) NOT NULL,
    [frete] [numeric](19, 6) NULL,
    [totalbaseincidencia] [numeric](19, 6) NOT NULL,
    [totaliva] [numeric](19, 6) NOT NULL,
    [totalfinal] [numeric](19, 6) NOT NULL,
    [totalcusto] [numeric](19, 6) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL
)
ELSE
TRUNCATE TABLE DocumentosFaturacao
```

## Tabela Documentos Faturação Linhas

```
IF NOT EXISTS (SELECT name FROM sys.tables where name =
'DocumentosFaturacaoLinhas')
CREATE TABLE [dbo].[DocumentosFaturacaoLinhas](
    [flstamp] [char](25) NOT NULL,
    [ftstamp] [char](25) NOT NULL,
    [nomedocumento] [varchar](20) NOT NULL,
    [anodocumento] [numeric](4, 0) NOT NULL,
    [numerodocumento] [varchar](20) NOT NULL,
    [datadocumento] [datetime] NOT NULL,
    [lordem] [numeric](10, 0) NOT NULL,
    [armazem] [numeric](5, 0) NOT NULL,
    [centrocosto] [varchar](20) NOT NULL,
    [produto] [char](18) NULL,
    [quantidade] [numeric](14, 4) NOT NULL,
    [iva] [numeric](19, 2) NOT NULL,
    [precounitario] [numeric](19, 6) NOT NULL,
    [valoriliquido] [numeric](19, 6) NOT NULL,
    [valordesconto] [numeric](19, 6) NOT NULL,
    [valortotal] [numeric](19, 6) NOT NULL,
    [precocusto] [numeric](19, 6) NOT NULL,
    [pcustoponderado] [numeric](19, 6) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL
)
ELSE
    TRUNCATE TABLE DocumentosFaturacaoLinhas
```

## Tabela Famílias

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Familias')
CREATE TABLE [dbo].[Familias](
    [referencia] [varchar](18) NOT NULL,
    [nome] [varchar](60) NOT NULL,
    [margem1] [numeric](16, 3) NOT NULL,
    [margem2] [numeric](16, 3) NOT NULL,
    [margem3] [numeric](16, 3) NOT NULL,
    [margem4] [numeric](16, 3) NOT NULL,
    [margem5] [numeric](16, 3) NOT NULL,
    [categoria] [varchar](35) NOT NULL,
    [classificacao] [varchar](20) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL,
    [taxaiva] [numeric](2, 0) NULL
)
ELSE
    TRUNCATE TABLE Familias
```

## Tabela Fornecedores

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Fornecedores')
CREATE TABLE [dbo].[Fornecedores](
    [nome] [varchar](55) NOT NULL,
    [numero] [numeric](10, 0) NOT NULL,
    [moeda] [varchar](11) NOT NULL,
    [fax] [varchar](60) NOT NULL,
    [telefone] [varchar](60) NOT NULL,
    [contacto] [varchar](30) NOT NULL,
    [morada] [varchar](55) NOT NULL,
    [localidade] [varchar](43) NOT NULL,
    [codpostal] [varchar](45) NOT NULL,
    [ncontribuinte] [varchar](20) NOT NULL,
    [zona] [numeric](5, 0) NULL,
    [tipo] [varchar](20) NOT NULL,
    [desconto] [numeric](5, 2) NOT NULL,
    [vencimento] [numeric](3, 0) NOT NULL,
    [descontopp] [numeric](5, 2) NOT NULL,
    [taxaiva] [numeric](2, 0) NOT NULL,
    [funcaocontacto] [varchar](15) NOT NULL,
    [email] [varchar](45) NOT NULL,
    [nacionalidade] [varchar](20) NOT NULL,
    [telemovel] [varchar](45) NOT NULL,
    [saldo] [numeric](19, 6) NOT NULL,
    [plafond] [numeric](19, 6) NOT NULL,
    [centrocosto] [varchar](20) NOT NULL,
    [classe] [varchar](30) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL,
)
ELSE
TRUNCATE TABLE Fornecedores
```

## Tabela Funcionários

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Funcionarios')
BEGIN
CREATE TABLE [dbo].[Funcionarios](
    [FuncionarioKey] [int] IDENTITY(1,1) NOT NULL,
    [iniciais] [varchar](3) NOT NULL,
    [codigo] [varchar](20) NOT NULL,
    [nome] [varchar](30) NOT NULL,
    [numero] [numeric](6, 0) NOT NULL,
    [grupo] [varchar](20) NOT NULL,
    [departamento] [varchar](20) NOT NULL,
    [email] [varchar](100) NOT NULL,
    [chefe] [numeric](6, 0) NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL
)
INSERT INTO [dbo].[Funcionarios]
([iniciais],[codigo],[nome],[numero],[grupo],[departamento],[email],[chefe],[
data_criacao],[hora_criacao],[data_ult_gravacao],[hora_ult_gravacao])
VALUES ('UNK','Unknown','Sem chefe/Não
Vendedor',0,'Unknown','Unknown','Unknown',NULL,'1900-01-01 00:00:00.000',
'08:00:01','1900-01-01 00:00:00.000','08:00:01')
END
ELSE
    TRUNCATE TABLE Funcionarios
```

## Tabela Produtos

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Produtos')
CREATE TABLE [dbo].[Produtos](
    [referencia] [char](18) NOT NULL,
    [designacao] [char](60) NOT NULL,
    [familia] [varchar](18) NOT NULL,
    [stockinicial] [numeric](13, 3) NOT NULL,
    [precovenda] [numeric](19, 6) NOT NULL,
    [pontoencomenda] [numeric](10, 3) NOT NULL,
    [fornecedor] [numeric](10, 0) NULL,
    [stockmaximo] [numeric](13, 3) NOT NULL,
    [stockminimo] [numeric](13, 3) NOT NULL,
    [dataabertura] [datetime] NOT NULL,
    [categoria] [int] NOT NULL,
    [classificacao] [varchar](20) NOT NULL,
    [precocusto] [numeric](19, 6) NOT NULL,
    [pcustoponderado] [numeric](19, 6) NOT NULL,
    [ultimoprecocusto] [numeric](19, 6) NOT NULL,
    [precomedio venda] [numeric](19, 6) NOT NULL,
    [ultimoprecovenda] [numeric](19, 6) NOT NULL,
    [quantidadecativa] [numeric](13, 3) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL,
    [taxaiva] [numeric](2, 0) NOT NULL
)
ELSE
    TRUNCATE TABLE Produtos
```

## Tabela Vendedores

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Vendedores')
CREATE TABLE [dbo].[Vendedores](
    [vendedor] [varchar](20) NOT NULL,
    [codigo] [numeric](4, 0) NOT NULL,
    [nome] [varchar](40) NOT NULL,
    [morada] [varchar](40) NOT NULL,
    [codpost] [varchar](28) NOT NULL,
    [local] [varchar](43) NOT NULL,
    [telefone] [varchar](60) NOT NULL,
    [email] [varchar](45) NOT NULL,
    [data_criacao] [datetime] NOT NULL,
    [hora_criacao] [varchar](8) NOT NULL,
    [data_ult_gravacao] [datetime] NOT NULL,
    [hora_ult_gravacao] [varchar](8) NOT NULL,
)
ELSE
    TRUNCATE TABLE Vendedores
```

## Tabela Zona

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'Zonas')
CREATE TABLE [dbo].[Zonas](
    [codigo] [numeric](5, 0) NOT NULL,
    [zona] [varchar](50) NULL,
)
ELSE
    TRUNCATE TABLE Zonas
```

## Scripts Loading Dados Para Staging Area

O parâmetro enviado neste script é a data do último loading efetuado. O script de loading incremental das outras tabelas é igual, mudando apenas o nome da tabela.

```
SELECT * FROM produtos
WHERE (data_ult_gravacao >= CONVERT(varchar(20), ?, 20))
```

## Scripts Criação Dimensões e Tabelas de Factos

Nesta secção estão todos os scripts para a criação das tabelas de dimensão e tabelas de factos do Data Mart.

## Dimensão Cliente

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimCliente')
CREATE TABLE [dbo].[DimCliente](
    [ClienteKey] [int] IDENTITY(1,1) NOT NULL,
    [ClienteId] [numeric](10, 0) NOT NULL,
    [Nome] [varchar](55) NOT NULL,
    [Contacto] [varchar](30) NOT NULL,
    [Morada] [varchar](55) NOT NULL,
    [Localidade] [varchar](45) NOT NULL,
    [CodigoPostal] [varchar](45) NOT NULL,
    [ZonaKey] [int] NOT NULL,
    [Email] [varchar](45) NOT NULL,
    [Telefone] [varchar](20) NOT NULL,
    [Telemovel] [varchar](20) NOT NULL,
    [Fax] [varchar](20) NOT NULL,
    [NumeroContribuinte] [varchar](20) NOT NULL,
    [CentroCusto] [varchar](20) NOT NULL,
    [MoedaKey] [int] NOT NULL,
    [VendedorKey] [int] NOT NULL,
    [NIB] [varchar](30) NOT NULL,
    [Segmento] [varchar](10) NOT NULL,
    [TipoCliente] [varchar](20) NOT NULL,
    [Saldo] [numeric](19, 6) NOT NULL,
    [Vencimento] [numeric](3, 0) NOT NULL,
    [Plafond] [numeric](19, 6) NOT NULL,
    [DataCriacaoKey] [int] NOT NULL,
    [HoraCriacaoKey] [int] NOT NULL,
    [EffectiveDateKey] [int] NOT NULL,
    [EffectiveTimeKey] [int] NOT NULL,
    [ExpiredDateKey] [int] NULL,
    [ExpiredTimeKey] [int] NULL,
    [IsCurrent] [nvarchar](3) NOT NULL
    CONSTRAINT [PK_DimCliente] PRIMARY KEY CLUSTERED( [ClienteKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-ClienteId')
    DROP INDEX [NonClusteredIndex-ClienteId] ON [dbo].[DimCliente]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimCliente_DimZona')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimZona]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimFuncionario')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimFuncionario]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimCliente_DimMoeda')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimMoeda]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimDate_Criacao')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimDate_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimDate_Effective')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimDate_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimDate_Expired')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimDate_Expired]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimTime_Criacao')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimTime_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimTime_Effective')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimTime_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimCliente_DimTime_Expired')
ALTER TABLE [dbo].[DimCliente] DROP CONSTRAINT [FK_DimCliente_DimTime_Expired]
```



## Dimensão Data

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimDate')
CREATE TABLE [dbo].[DimDate](
    [DateKey] [int] IDENTITY(1,1) NOT NULL,
    [FullDate] [datetime] NOT NULL,
    [Year] [int] NOT NULL,
    [Semester] [tinyint] NOT NULL,
    [Trimester] [tinyint] NOT NULL,
    [Quarter] [tinyint] NOT NULL,
    [Month] [tinyint] NOT NULL,
    [MonthName] [nvarchar](10) NOT NULL,
    [Week] [tinyint] NOT NULL,
    [Season] [nvarchar](10) NOT NULL,
    [DayNumberOfYear] [int] NOT NULL,
    [DayNumberOfMonth] [tinyint] NOT NULL,
    [DayNumberOfWeek] [tinyint] NOT NULL,
    [DayOfWeek] [nvarchar](10) NOT NULL,
    [LastDayMonth] [nvarchar](4) 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]
```

## Dimensão DataMes

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimDateMes')
CREATE TABLE [dbo].[DimDateMes](
    [DateKey] [int] IDENTITY(1,1) NOT NULL,
    [Year] [int] NOT NULL,
    [Month] [tinyint] NOT NULL,
    [MonthName] [nvarchar](10) NOT NULL,
    [Semester] [tinyint] NOT NULL,
    [Trimester] [tinyint] NOT NULL,
    [Quarter] [tinyint] NOT NULL,
    CONSTRAINT [PK_DimDateMes] 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]
```

## Dimensão Fornecedor

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimFornecedor')
CREATE TABLE [dbo].[DimFornecedor](
    [FornecedorKey] [int] IDENTITY(1,1) NOT NULL,
    [FornecedorId] [numeric](10, 0) NOT NULL,
    [Nome] [varchar](55) NOT NULL,    [Contacto] [varchar](30) NOT NULL,
    [FuncaoContacto] [varchar](15) NOT NULL,
    [Classe] [varchar](30) NOT NULL,
    [Nacionalidade] [varchar](20) NOT NULL,
    [Morada] [varchar](55) NOT NULL,
    [Localidade] [varchar](45) NOT NULL,
    [CodigoPostal] [varchar](45) NOT NULL,
    [ZonaKey] [int] NOT NULL,
    [Email] [varchar](45) NOT NULL,
    [Telefone] [varchar](20) NOT NULL,
    [Telemovel] [varchar](20) NOT NULL,
    [Fax] [varchar](20) NOT NULL,
    [NumeroContribuinte] [varchar](20) NOT NULL,
    [CentroCusto] [varchar](20) NOT NULL,
    [Saldo] [numeric](19, 6) NOT NULL,
    [Vencimento] [numeric](10) NOT NULL,
    [Plafond] [numeric](19, 6) NOT NULL,
    [MoedaKey] [int] NOT NULL,
    [Desconto] [numeric](5, 2) NOT NULL,
    [DescontoProntoPagamento] [numeric](5, 2) NOT NULL,
    [TaxaIva] [numeric](2, 0) NOT NULL,
    [Tipo] [varchar](20) NOT NULL,
    [DataCriacaoKey] [int] NOT NULL,
    [HoraCriacaoKey] [int] NOT NULL,
    [EffectiveDateKey] [int] NOT NULL,
    [EffectiveTimeKey] [int] NOT NULL,
    [ExpiredDateKey] [int] NULL,
    [ExpiredTimeKey] [int] NULL,
    [IsCurrent] [nvarchar](3) NOT NULL
    CONSTRAINT [PK_DimFornecedor] PRIMARY KEY CLUSTERED([FornecedorKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY] ) ON [PRIMARY]
ELSE BEGIN
IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-FornecedorId')
DROP INDEX [NonClusteredIndex-FornecedorId] ON [dbo].[DimFornecedor]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimFornecedor_DimZona')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimZona]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimFornecedor_DimMoeda')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimMoeda]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimDate_Criacao')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimDate_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimDate_Effective')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimDate_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimDate_Expired')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimDate_Expired]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimTime_Criacao')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimTime_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimTime_Effective')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimTime_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFornecedor_DimTime_Expired')
ALTER TABLE [dbo].[DimFornecedor] DROP CONSTRAINT [FK_DimFornecedor_DimTime_Expired]
END
```

## Dimensão Funcionário

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimFuncionario')
BEGIN
CREATE TABLE [dbo].[DimFuncionario](
    [FuncionarioKey] [int] NOT NULL,
    [FuncionarioId] [numeric](6, 0) NOT NULL,
    [VendedorId] [numeric](4, 0) NOT NULL,
    [Nome] [varchar](30) NOT NULL,
    [Iniciais] [varchar](3) NOT NULL,
    [Codigo] [varchar](20) NOT NULL,
    [Tipo] [varchar](20) NOT NULL,
    [Departamento] [varchar](20) NOT NULL,
    [ChefeKey] [int] NULL,
    [EmailEmpresa] [varchar](45) NOT NULL,
    [EmailVendedor] [varchar](45) NOT NULL,
    [Telefone] [varchar](20) NOT NULL,
    [Morada] [varchar](55) NOT NULL,
    [CodigoPostal] [varchar](45) NOT NULL,
    [Localidade] [varchar](45) NOT NULL,
    [DataCriacaoKey] [int] NOT NULL,
    [HoraCriacaoKey] [int] NOT NULL,
    [EffectiveDateKey] [int] NOT NULL,
    [EffectiveTimeKey] [int] NOT NULL,
    [ExpiredDateKey] [int] NULL,
    [ExpiredTimeKey] [int] NULL,
    [IsCurrent] [nvarchar](3) NOT NULL
    CONSTRAINT [PK_DimFuncionario] PRIMARY KEY CLUSTERED([FuncionarioKey] 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
ELSE
BEGIN
IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-FuncionarioId')
DROP INDEX [NonClusteredIndex-FuncionarioId] ON [dbo].[DimFuncionario]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimFuncionario_Chefe')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT
[FK_DimFuncionario_DimFuncionario_Chefe]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimDate_Criacao')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT [FK_DimFuncionario_DimDate_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimDate_Effective')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT
[FK_DimFuncionario_DimDate_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimDate_Expired')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT [FK_DimFuncionario_DimDate_Expired]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimTime_Criacao')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT [FK_DimFuncionario_DimTime_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimTime_Effective')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT
[FK_DimFuncionario_DimTime_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimFuncionario_DimTime_Expired')
ALTER TABLE [dbo].[DimFuncionario] DROP CONSTRAINT [FK_DimFuncionario_DimTime_Expired]
END
```

## Dimensão Moeda

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimMoeda')
CREATE TABLE [dbo].[DimMoeda](
    [MoedaKey] [int] IDENTITY(1,1) NOT NULL,
    [Descricao] [nvarchar](50) NULL,
    [Abreviatura] [nvarchar](5) NULL
    CONSTRAINT [PK_DimMoeda] PRIMARY KEY CLUSTERED([MoedaKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
```

## Dimensão Produto

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimProduto')
CREATE TABLE [dbo].[DimProduto](
    [ProdutoKey] [int] IDENTITY(1,1) NOT NULL, [ProdutoId] [char](18) NOT NULL,
    [FamiliaId] [varchar](18) NOT NULL, [DescricaoProduto] [char](60) NOT NULL,
    [DescricaoFamilia] [varchar](60) NOT NULL, [CategoriaProduto] [varchar](35) NOT NULL,
    [CategoriaFamilia] [varchar](35) NOT NULL, [ClassificacaoProduto] [varchar](20) NOT NULL,
    [ClassificacaoFamilia] [varchar](20) NOT NULL, [StockInicial] [numeric](13) NOT NULL,
    [StockMaximo] [numeric](13) NOT NULL, [StockMinimo] [numeric](13) NOT NULL,
    [Margem1] [numeric](16, 3) NOT NULL, [Margem2] [numeric](16, 3) NOT NULL,
    [Margem3] [numeric](16, 3) NOT NULL, [Margem4] [numeric](16, 3) NOT NULL,
    [Margem5] [numeric](16, 3) NOT NULL, [PrecoVenda] [numeric](19, 6) NOT NULL,
    [PrecoCusto] [numeric](19, 6) NOT NULL, [PrecoCustoPonderado] [numeric](19, 6) NOT NULL,
    [UltimoPrecoCusto] [numeric](19, 6) NOT NULL, [UltimoPrecoVenda] [numeric](19, 6) NOT NULL,
    [QuantidadeCativa] [numeric](13) NOT NULL, [TaxaIVA] [numeric](2) NOT NULL,
    [PontoEncomenda] [numeric](10, 3) NOT NULL, [FornecedorKey] [int] NOT NULL,
    [DataAberturaKey] [int] NOT NULL, [DataCriacaoKey] [int] NOT NULL,
    [HoraCriacaoKey] [int] NOT NULL, [EffectiveDateKey] [int] NOT NULL,
    [EffectiveTimeKey] [int] NOT NULL, [ExpiredDateKey] [int] NULL, [ExpiredTimeKey] [int] NULL,
    [IsCurrent] [nvarchar](3) NOT NULL
    CONSTRAINT [PK_DimProduto] PRIMARY KEY CLUSTERED( [ProdutoKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
ELSE BEGIN
IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-ProdutoId')
DROP INDEX [NonClusteredIndex-ProdutoId] ON [dbo].[DimProduto]
IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-FamiliaId')
DROP INDEX [NonClusteredIndex-FamiliaId] ON [dbo].[DimProduto]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimProduto_DimFornecedor_Fornecedor')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimFornecedor_Fornecedor]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_DimProduto_DimDate_DataAbertura')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimDate_DataAbertura]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimDate_Criacao')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimDate_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimDate_Effective')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimDate_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimDate_Expired')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimDate_Expired]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimTime_Criacao')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimTime_Criacao]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimTime_Effective')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimTime_Effective]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_DimProduto_DimTime_Expired')
ALTER TABLE [dbo].[DimProduto] DROP CONSTRAINT [FK_DimProduto_DimTime_Expired]
END
```

## Dimensão Tempo

---

```
IF NOT EXISTS(SELECT name FROM sys.tables WHERE name = 'DimTime')
CREATE TABLE [dbo].[DimTime](
    [TimeKey] [int] IDENTITY(1,1) NOT NULL,
    [FullTime] [time] NOT NULL,
    [FullTime_int] [int] NOT NULL,
    [FullTime_varchar] [varchar](8) NOT NULL,
    [Hour] [tinyint] NOT NULL,
    [Minute] [tinyint] NOT NULL,
    [Second] [tinyint] NOT NULL,
    [Period] [nvarchar](10) NOT NULL,
    [LunchTime] [nvarchar](3) NOT NULL,
    [DinnerTime] [nvarchar](3) NOT NULL,
    CONSTRAINT [PK_DimTime] PRIMARY KEY CLUSTERED ([TimeKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
```

## Dimensão Zona

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'DimZona')
BEGIN
CREATE TABLE [dbo].[DimZona](
    [ZonaKey] [int] IDENTITY(1,1) NOT NULL,
    [ZonaId] [numeric](5, 0) NOT NULL,
    [Zona] [varchar](50) NULL
    CONSTRAINT [PK_DimZona] PRIMARY KEY CLUSTERED ([ZonaKey] 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-ZonaId] ON
[dbo].[DimZona]([ZonaId] 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)
END
```

## Dimensão “Junk” Faturação

---

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'JunkDimFaturacao')
BEGIN
CREATE TABLE [dbo].[JunkDimFaturacao](
    [JunkFaturacaoKey] [int] IDENTITY(1,1) NOT NULL,
    [NumeroDocumento] [varchar](20) NOT NULL,
    [NomeDocumento] [varchar](20) NOT NULL,
    [Armazem] [varchar](50) NOT NULL,
    [LOrdem] [numeric](10, 0) NOT NULL,
    [CentroCusto] [varchar](20) NOT NULL,
    [Segmento] [varchar](25) NOT NULL,
    [Tipo] [varchar](20) NOT NULL
    CONSTRAINT [PK_JunkDimFaturacao] PRIMARY KEY CLUSTERED([JunkFaturacaoKey] 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
```

## Tabela de Factos Taxa de Câmbios

```
IF NOT EXISTS (SELECT name FROM sys.tables where name = 'FactCurrencyRate')
CREATE TABLE [dbo].[FactCurrencyRate](
    [DateKey] [int] NOT NULL,
    [MoedaOrigemKey] [int] NOT NULL,
    [MoedaDestinoKey] [int] NOT NULL,
    [FatorCambio] [numeric](20, 10) NOT NULL
CONSTRAINT [PK_FactCurrencyRate] PRIMARY KEY CLUSTERED(
    [DateKey] ASC,
    [MoedaOrigemKey] ASC,
    [MoedaDestinoKey] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]) ON [PRIMARY]
ELSE
BEGIN
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactCurrencyRate_DimDate')
ALTER TABLE [dbo].[FactCurrencyRate] DROP CONSTRAINT [FK_FactCurrencyRate_DimDate]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_FactCurrencyRate_DimMoedaOrigem')
ALTER TABLE [dbo].[FactCurrencyRate] DROP CONSTRAINT
[FK_FactCurrencyRate_DimMoedaOrigem]
IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
'FK_FactCurrencyRate_DimMoedaDestino')
ALTER TABLE [dbo].[FactCurrencyRate] DROP CONSTRAINT
[FK_FactCurrencyRate_DimMoedaDestino]
END
```

## Tabela de Factos Faturação Linhas

---

```

IF NOT EXISTS (SELECT name FROM sys.tables where name = 'FactFaturacaoLinhas')
BEGIN
    CREATE TABLE [dbo].[FactFaturacaoLinhas](
        [AnoDocumento] int NOT NULL,
        [DateKey] [int] NOT NULL,
        [ClienteKey] [int] NOT NULL,
        [FuncionarioKey] [int] NOT NULL,
        [VendedorKey] [int] NOT NULL,
        [ProdutoKey] [int] NOT NULL,
        [ZonaKey] [int] NOT NULL,
        [MoedaKey] [int] NOT NULL,
        [JunkFaturacaoKey] [int] NOT NULL,
        [FTStamp] [char](25) NOT NULL,
        [FLStamp] [char](25) NOT NULL,
        [Armazem] [varchar](50) NOT NULL,
        [Quantidade] [numeric](14, 0) NOT NULL,
        [TaxaIVA] [numeric](4, 2) NOT NULL,
        [PrecoUnidade] [numeric](19, 6) NOT NULL,
        [FreteUnidade] [numeric](19, 6) NOT NULL,
        [ValorIliquido] [numeric](19, 6) NOT NULL,
        [ValorDesconto] [numeric](19, 6) NOT NULL,
        [PrecoCusto] [numeric](19, 6) NOT NULL,
        [PrecoCustoPonderado] [numeric](19, 6) NOT NULL,
        [TotalIliquidoProdutos] [numeric](19, 6) NOT NULL,
        [TotalIVA] [numeric](19, 6) NOT NULL,
        [TotalFrete] [numeric](19, 6) NOT NULL,
        [TotalFinal] [numeric](19, 6) NOT NULL,
        [TotalCusto] [numeric](19, 6) NOT NULL,
        [Lucro] [numeric](19, 6) NOT NULL) ON RangeDateScheme ([AnoDocumento])
    ALTER TABLE [dbo].[FactFaturacaoLinhas] ADD CONSTRAINT [PK_FactFaturacaoLinhas] PRIMARY KEY
    CLUSTERED (
        [DateKey] ASC,
        [ClienteKey] ASC,
        [FuncionarioKey] ASC,
        [VendedorKey] ASC,
        [ProdutoKey] ASC,
        [ZonaKey] ASC,
        [FLStamp] ASC ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
END
ELSE
BEGIN
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-DateKey')
    DROP INDEX [NonClusteredIndex-DateKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-ClienteKey')
    DROP INDEX [NonClusteredIndex-ClienteKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-FuncionarioKey')
    DROP INDEX [NonClusteredIndex-FuncionarioKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-VendedorKey')
    DROP INDEX [NonClusteredIndex-VendedorKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-ProdutoKey')
    DROP INDEX [NonClusteredIndex-ProdutoKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-ZonaKey')
    DROP INDEX [NonClusteredIndex-ZonaKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-MoedaKey')
    DROP INDEX [NonClusteredIndex-MoedaKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-JunkFaturacaoKey')
    DROP INDEX [NonClusteredIndex-JunkFaturacaoKey] ON [dbo].[FactFaturacaoLinhas]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimDate')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimDate]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimCliente')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimCliente]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimFuncionario')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimFuncionario]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimVendedor')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimVendedor]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimProduto')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimProduto]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimZona')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimZona]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name = 'FK_FactFaturacaoLinhas_DimMoeda')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimMoeda]
    IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
    'FK_FactFaturacaoLinhas_DimJunkFaturacao')
    ALTER TABLE [dbo].[FactFaturacaoLinhas] DROP CONSTRAINT [FK_FactFaturacaoLinhas_DimJunkFaturacao]
END

```



## Tabela de Factos Faturação Mês

```
IF NOT EXISTS (SELECT name FROM sys.tables where name =
'FactFaturacaoLinhasMes')
BEGIN
    CREATE TABLE [dbo].[FactFaturacaoLinhasMes](
        [DateMesKey] [int] NOT NULL,
        [ProdutoKey] [int] NOT NULL,
        [Quantidade] [numeric](14, 0) NOT NULL,
        [PrecoUnidade] [numeric](19, 6) NOT NULL,
        [FreteUnidade] [numeric](19, 6) NOT NULL,
        [ValorIliquido] [numeric](19, 6) NOT NULL,
        [ValorDesconto] [numeric](19, 6) NOT NULL,
        [PrecoCusto] [numeric](19, 6) NOT NULL,
        [PrecoCustoPonderado] [numeric](19, 6) NOT NULL,
        [TotalIliquidoProdutos] [numeric](19, 6) NOT NULL,
        [TotalIVA] [numeric](19, 6) NOT NULL,
        [TotalFrete] [numeric](19, 6) NOT NULL,
        [TotalFinal] [numeric](19, 6) NOT NULL,
        [TotalCusto] [numeric](19, 6) NOT NULL,
        [Lucro] [numeric](19, 6) NOT NULL
        CONSTRAINT [PK_FactFaturacaoLinhasMes] PRIMARY KEY CLUSTERED (
            [DateMesKey] ASC,
            [ProdutoKey] 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
ELSE
    BEGIN
        TRUNCATE TABLE [FactFaturacaoLinhasMes]
        IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-
        DateMesKey')
        DROP INDEX [NonClusteredIndex-DateMesKey] ON [dbo].[FactFaturacaoLinhasMes]
        IF EXISTS (select * from sys.indexes where name = 'NonClusteredIndex-
        ProdutoKey')
        DROP INDEX [NonClusteredIndex-ProdutoKey] ON [dbo].[FactFaturacaoLinhasMes]
        IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
        'FK_FactFaturacaoLinhasMes_DimDateMes')
        ALTER TABLE [dbo].[FactFaturacaoLinhasMes] DROP CONSTRAINT
        [FK_FactFaturacaoLinhasMes_DimDateMes]
        IF EXISTS (SELECT name FROM sys.foreign_keys WHERE name =
        'FK_FactFaturacaoLinhasMes_DimProduto')
        ALTER TABLE [dbo].[FactFaturacaoLinhasMes] DROP CONSTRAINT
        [FK_FactFaturacaoLinhasMes_DimProduto]
    END
```

## Scripts para a criação de Índices

Nesta secção estão todos os scripts que foram usados para criar os índices como medida de otimização.



## Índices da dimensão Data

```
CREATE UNIQUE NONCLUSTERED INDEX [NonClusteredIndex-FullDate] ON
[dbo].[DimDate] ( [FullDate] ASC ) WITH (PAD_INDEX = OFF,
STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
```

## Índices da dimensão Tempo

```
CREATE UNIQUE NONCLUSTERED INDEX [NonClusteredIndex-FullTime_int] ON
[dbo].[DimTime]( [FullTime_int] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON)
```

## Índices da dimensão Funcionário

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-FuncionarioId] ON
[dbo].[DimFuncionario] ( [FuncionarioId] 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)
```

## Índices da dimensão Fornecedor

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-FornecedorId] ON
[dbo].[DimFornecedor]( [FornecedorId] 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)
```

## Índices da dimensão Cliente

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ClienteId] ON
[dbo].[DimCliente]( [ClienteId] 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)
```

## Índices da dimensão Produto

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ProdutoId] ON [dbo].[DimProduto]
([ProdutoId] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-FamiliaId] ON
[dbo].[DimProduto] ([FamiliaId] 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)
```

## Índices da tabela de factos Faturação Linhas

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-DateKey] ON
[dbo].[FactFaturacaoLinhas]( [DateKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ClienteKey] ON
[dbo].[FactFaturacaoLinhas]( [ClienteKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-FuncionarioKey] ON
[dbo].[FactFaturacaoLinhas]( [FuncionarioKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-VendedorKey] ON
[dbo].[FactFaturacaoLinhas]( [VendedorKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ProdutoKey] ON
[dbo].[FactFaturacaoLinhas]( [ProdutoKey] 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) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ZonaKey] ON
[dbo].[FactFaturacaoLinhas]( [ZonaKey] 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) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [NonClusteredIndex-MoedaKey] ON
[dbo].[FactFaturacaoLinhas]( [MoedaKey] 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) ON [PRIMARY]
CREATE NONCLUSTERED INDEX [NonClusteredIndex-JunkFaturacaoKey] ON
[dbo].[FactFaturacaoLinhas]( [JunkFaturacaoKey] 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) ON [PRIMARY]
```

## Índices da tabela de factos Faturação Linhas Produto Mês

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-DateMesKey] ON
[dbo].[FactFaturacaoLinhasProdutoMes]( [DateMesKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ProdutoKey] ON
[dbo].[FactFaturacaoLinhasProdutoMes]( [ProdutoKey] 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) ON [PRIMARY]
```

## Índices da tabela de factos Faturação Linhas Cliente Mês

```
CREATE NONCLUSTERED INDEX [NonClusteredIndex-DateMesKey] ON
[dbo].[FactFaturacaoLinhasClienteMes] ( [DateMesKey] 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)
CREATE NONCLUSTERED INDEX [NonClusteredIndex-ClienteKey] ON
[dbo].[FactFaturacaoLinhasClienteMes] ( [ClienteKey] 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) ON [PRIMARY]
```

## Scripts para a criação de chaves primárias e estrangeiras

Nesta secção estão todos os scripts que foram usados para criar as chaves primárias e as chaves estrangeiras que unem o Data Mart.

### Chaves da dimensão Funcionário

```
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimFuncionario_Chefe] FOREIGN KEY([ChefeKey])
REFERENCES [dbo].[DimFuncionario] ([FuncionarioKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimFuncionario_Chefe]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimDate_Criacao] FOREIGN KEY([DataCriacaoKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimDate_Criacao]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimDate_Effective] FOREIGN KEY([EffectiveDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimDate_Effective]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimDate_Expired] FOREIGN KEY([ExpiredDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimDate_Expired]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimTime_Criacao] FOREIGN KEY([HoraCriacaoKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimTime_Criacao]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimTime_Effective] FOREIGN KEY([EffectiveTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimTime_Effective]
ALTER TABLE [dbo].[DimFuncionario] WITH CHECK ADD CONSTRAINT
[FK_DimFuncionario_DimTime_Expired] FOREIGN KEY([ExpiredTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFuncionario] CHECK CONSTRAINT
[FK_DimFuncionario_DimTime_Expired]
```

## Chaves da dimensão Fornecedor

```
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimZona] FOREIGN KEY([ZonaKey])
REFERENCES [dbo].[DimZona] ([ZonaKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT [FK_DimFornecedor_DimZona]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimMoeda] FOREIGN KEY(MoedaKey)
REFERENCES [dbo].[DimMoeda] (MoedaKey)
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimMoeda]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimDate_Criacao] FOREIGN KEY([DataCriacaoKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimDate_Criacao]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimDate_Effective] FOREIGN KEY([EffectiveDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimDate_Effective]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimDate_Expired] FOREIGN KEY([ExpiredDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimDate_Expired]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimTime_Criacao] FOREIGN KEY([HoraCriacaoKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimTime_Criacao]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimTime_Effective] FOREIGN KEY([EffectiveTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimTime_Effective]
ALTER TABLE [dbo].[DimFornecedor] WITH CHECK ADD CONSTRAINT
[FK_DimFornecedor_DimTime_Expired] FOREIGN KEY([ExpiredTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimFornecedor] CHECK CONSTRAINT
[FK_DimFornecedor_DimTime_Expired]
```

## Chaves da dimensão Cliente

---

```
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimZona] FOREIGN KEY([ZonaKey])
REFERENCES [dbo].[DimZona] ([ZonaKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT [FK_DimCliente_DimZona]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimFuncionario] FOREIGN KEY([VendedorKey])
REFERENCES [dbo].[DimFuncionario] ([FuncionarioKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimFuncionario]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimMoeda] FOREIGN KEY([MoedaKey])
REFERENCES [dbo].[DimMoeda] ([MoedaKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT [FK_DimCliente_DimMoeda]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimDate_Criacao] FOREIGN KEY([DataCriacaoKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimDate_Criacao]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimDate_Effective] FOREIGN KEY([EffectiveDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimDate_Effective]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimDate_Expired] FOREIGN KEY([ExpiredDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimDate_Expired]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimTime_Criacao] FOREIGN KEY([HoraCriacaoKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimTime_Criacao]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimTime_Effective] FOREIGN KEY([EffectiveTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimTime_Effective]
ALTER TABLE [dbo].[DimCliente] WITH CHECK ADD CONSTRAINT
[FK_DimCliente_DimTime_Expired] FOREIGN KEY([ExpiredTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimCliente] CHECK CONSTRAINT
[FK_DimCliente_DimTime_Expired]
```

## Chaves da dimensão Produto

```
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimFornecedor_Fornecedor] FOREIGN KEY([FornecedorKey])
REFERENCES [dbo].[DimFornecedor] ([FornecedorKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimFornecedor_Fornecedor]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimDate_DataAbertura] FOREIGN KEY([DataAberturaKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimDate_DataAbertura]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimDate_Criacao] FOREIGN KEY([DataCriacaoKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimDate_Criacao]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimDate_Effective] FOREIGN KEY([EffectiveDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimDate_Effective]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimDate_Expired] FOREIGN KEY([ExpiredDateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimDate_Expired]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimTime_Criacao] FOREIGN KEY([HoraCriacaoKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimTime_Criacao]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimTime_Effective] FOREIGN KEY([EffectiveTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimTime_Effective]
ALTER TABLE [dbo].[DimProduto] WITH CHECK ADD CONSTRAINT
[FK_DimProduto_DimTime_Expired] FOREIGN KEY([ExpiredTimeKey])
REFERENCES [dbo].[DimTime] ([TimeKey])
ALTER TABLE [dbo].[DimProduto] CHECK CONSTRAINT
[FK_DimProduto_DimTime_Expired]
```

## Chaves da tabela de factos Faturação Linhas

```
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimDate] FOREIGN KEY([DateKey])
REFERENCES [dbo].[DimDate] ([DateKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimDate]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimCliente] FOREIGN KEY([ClienteKey])
REFERENCES [dbo].[DimCliente] ([ClienteKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimCliente]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimFuncionario] FOREIGN KEY([FuncionarioKey])
REFERENCES [dbo].[DimFuncionario] ([FuncionarioKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimFuncionario]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimVendedor] FOREIGN KEY([VendedorKey])
REFERENCES [dbo].[DimFuncionario] ([FuncionarioKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimVendedor]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimProduto] FOREIGN KEY([ProdutoKey])
REFERENCES [dbo].[DimProduto] ([ProdutoKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimProduto]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimZona] FOREIGN KEY([ZonaKey])
REFERENCES [dbo].[DimZona] ([ZonaKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimZona]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimMoeda] FOREIGN KEY([MoedaKey])
REFERENCES [dbo].[DimMoeda] ([MoedaKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimMoeda]
ALTER TABLE [dbo].[FactFaturacaoLinhas] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhas_DimJunkFaturacao] FOREIGN KEY([JunkFaturacaoKey])
REFERENCES [dbo].[JunkDimFaturacao] ([JunkFaturacaoKey])
ALTER TABLE [dbo].[FactFaturacaoLinhas] CHECK CONSTRAINT
[FK_FactFaturacaoLinhas_DimJunkFaturacao]
```

## Chaves da tabela de factos Faturação Linhas Produto Mês

```
ALTER TABLE [dbo].[FactFaturacaoLinhasProdutoMes] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhasProdutoMes_DimDateMes] FOREIGN KEY([DateMesKey])
REFERENCES [dbo].[DimDateMes] ([DateKey])
ALTER TABLE [dbo].[FactFaturacaoLinhasProdutoMes] CHECK CONSTRAINT
[FK_FactFaturacaoLinhasProdutoMes_DimDateMes]
ALTER TABLE [dbo].[FactFaturacaoLinhasProdutoMes] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhasProdutoMes_DimProduto] FOREIGN KEY([ProdutoKey])
REFERENCES [dbo].[DimProduto] ([ProdutoKey])
ALTER TABLE [dbo].[FactFaturacaoLinhasProdutoMes] CHECK CONSTRAINT
[FK_FactFaturacaoLinhasProdutoMes_DimProduto]
```

## Chaves da tabela de factos Faturação Linhas Cliente Mês

```
ALTER TABLE [dbo].[FactFaturacaoLinhasClienteMes] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhasClienteMes_DimDateMes] FOREIGN KEY([DateMesKey])
REFERENCES [dbo].[DimDateMes] ([DateKey])
ALTER TABLE [dbo].[FactFaturacaoLinhasClienteMes] CHECK CONSTRAINT
[FK_FactFaturacaoLinhasClienteMes_DimDateMes]
ALTER TABLE [dbo].[FactFaturacaoLinhasClienteMes] WITH CHECK ADD CONSTRAINT
[FK_FactFaturacaoLinhasClienteMes_DimCliente] FOREIGN KEY([ClienteKey])
REFERENCES [dbo].[DimCliente] ([ClienteKey])
ALTER TABLE [dbo].[FactFaturacaoLinhasClienteMes] CHECK CONSTRAINT
[FK_FactFaturacaoLinhasClienteMes_DimCliente]
```



## Análises Dimensionais

### Análise 1

Valores totais das vendas efetuadas no primeiro semestre de 2012, detalhados por zona geográfica e por categoria de produto.

Date.Semester	1	
Date.Year	2012	
Rótulos de Linha	Total Final - Faturacao Linhas	
<b>Centro</b>	<b>18,719.37 €</b>	
Ar	7,744.08 €	
Cases	68.88 €	
Cordas	6,780.99 €	
Corpo do Instrumento	1,006.14 €	
Interface	113.16 €	
Leitor de Cd	125.46 €	
Leitor de Mp3	178.35 €	
Membranas	2,678.94 €	
Pilhas	23.37 €	
<b>Espanha</b>	<b>4,900.00 €</b>	
Ar	1,845.00 €	
Cordas	2,633.00 €	
Corpo do Instrumento	320.00 €	
Leitor de Cd	102.00 €	
<b>Norte</b>	<b>14,860.86 €</b>	
Ar	2,733.06 €	
Cordas	6,231.18 €	
Corpo do Instrumento	398.52 €	
Leitor de Mp3	2,675.25 €	
Membranas	2,638.35 €	
Peças	159.90 €	
Pilhas	24.60 €	
<b>Sul</b>	<b>4,256.94 €</b>	
Cases	68.88 €	
Cordas	1,937.06 €	
Corpo do Instrumento	570.48 €	
Interface	113.16 €	
Leitor de Cd	125.46 €	
Leitor de Mp3	250.92 €	
Membranas	1,108.23 €	
Pilhas	82.75 €	
<b>Total geral</b>	<b>42,737.17 €</b>	

## Análise 2

Valores totais dos fretes despendidos no transporte dos produtos durante o mês de Fevereiro de 2012, detalhados por armazém e por família de produto.

A	B
Date.Year	2012
Date.Month Name	February
Valores totais dos fretes despendidos no	
Rótulos de Linha	Total Frete
Algarve	85.043748
Ar	30.45625
Cordas	21.533332
Corpo do Instrumento	33.054166
Aveiro	63.262916
Ar	18.965
Membranas	37.297916
Pilhas	7
Coimbra	48.277083
Ar	15.2875
Cordas	20.03125
Membranas	12.958333
Lisboa	75.015833
Cordas	25.965
Corpo do Instrumento	20.915
Membranas	25.223333
Pilhas	2.9125
Porto	71.525416
Cordas	31.039583
Corpo do Instrumento	40.485833
Total geral	343.124996

## Análise 3

Quantidades dos produtos vendidos no primeiro trimestre de 2012, detalhados por fornecedor e por vendedor.

Date.Quarter	1
Date.Year	2012
Rótulos de Linha	Quantidade
Bruno Silva Matos	108
Nuno Gama Pinto	31
Sara Sofia Ramos	77
FarolInstrumentos	109
Nuno Gama Pinto	91
Sara Sofia Ramos	18
FarolInstrumentos II	18
Nuno Gama Pinto	9
Sara Sofia Ramos	9
MusiCarlos, Audio Ida	38
Nuno Gama Pinto	35
Sara Sofia Ramos	3
Total geral	273

## Análise 4

Valores líquidos das vendas efetuadas e respetivos valores de IVA pagos durante o ano de 2012, detalhados por cliente e funcionário.

Date.Year	2012	
	Valores	
Rótulos de Linha	Total Ilíquido Produtos - Faturacao Linhas	Total IVA - Faturacao Linhas
Carmina Sofia Costa das Neves	7,301.00 €	1,679.23 €
Bernardo Santiago	145.00 €	33.35 €
Fernando Martins	1,454.00 €	334.42 €
Gosto pela Música	2,893.00 €	665.39 €
Hora a Hora, Ida	1,835.00 €	422.05 €
Luz e Som, Ida	204.00 €	46.92 €
Orquestra de Vila Real	19.00 €	4.37 €
Ritmos Quentes, Ida	751.00 €	172.73 €
Elvira Cristina P. Andrade	10,374.00 €	1,913.37 €
Albertino Maciel	816.00 €	187.68 €
Bernardo Santiago	1,044.00 €	240.12 €
Gosto pela Música	280.00 €	64.40 €
La Cartuja Musical	2,055.00 €	0.00 €
Loja de Instrumentos, Ida	2,300.00 €	529.00 €
Rancho Folclórico da Beira Baixa	3,879.00 €	892.17 €
Francisco Miguel Gamito	3,865.00 €	888.95 €
Coro Nossa Senhora Assunção	3,793.00 €	872.39 €
Eira e Beira, Ida	72.00 €	16.56 €
Jorge Junqueira	3,800.00 €	680.80 €
Coro Nossa Senhora Assunção	662.00 €	152.26 €
La Musical Fashion	840.00 €	0.00 €
Ritmos Quentes, Ida	1,685.00 €	387.55 €
Sons e Ritmos	237.00 €	54.51 €
Toque a Musica, Lda.	376.00 €	86.48 €
Nuno Gama Pinto	4,570.00 €	396.75 €
Hora a Hora, Ida	234.00 €	53.82 €
La Cartuja Musical	990.00 €	0.00 €
La Musical Fashion	1,855.00 €	0.00 €
Sons e Ritmos	268.00 €	61.64 €
Supermercados Gota Doce, S.A.	1,223.00 €	281.29 €
Sara Sofia Ramos	5,909.00 €	1,359.07 €
Albertino Maciel	2,058.00 €	473.34 €
Coro Nossa Senhora Assunção	543.00 €	124.89 €
Orquestra de Vila Real	2,329.00 €	535.67 €
Rancho Folclórico da Beira Baixa	851.00 €	195.73 €
Supermercados Gota Doce, S.A.	128.00 €	29.44 €
Total geral	35,819.00 €	6,918.17 €

## Análise 5

Valores totais das vendas efetuadas no último dia de cada mês do ano de 2012, detalhados por categoria de produto, com possibilidade de análise até ao nível do produto.

Date.Last Day Month	Sim	
Date.Year	2012	
Rótulos de Linha	Total Final	
Ar	82.41 €	
Flauta	82.41 €	
Cordas	6,301.05 €	
Guitarra portuguesa	814.26 €	
Rabeca chuleira	1,626.06 €	
Viola amarantina	667.89 €	
Viola campaniça	495.00 €	
Viola de arame	2,055.78 €	
Viola toeira	642.06 €	
Corpo do Instrumento	89.79 €	
Ferrinhos	66.42 €	
Gaita de amolador	14.76 €	
Gaita de palhas	8.61 €	
Leitor de Cd	102.00 €	
Leitor de cd cdp 5300	102.00 €	
Pilhas	8.61 €	
Pilha alcalina lr6	4.92 €	
Pilha zinco-carbono r14	3.69 €	
Total geral	6,583.86 €	

## Análise 6

Valores totais do lucro na 6ª semana do ano de 2012, referentes às vendas ocorridas na zona Norte, detalhados por família de produto, com possibilidade de análise até ao nível do produto.

Date.Year	2012	
Date.Week	6	
Zona.Zona	Norte	
Rótulos de Linha	▼	Lucro
[-] Acessórios		183.30 €
Cordas para guitarras		183.30 €
[-] Instrumentos		2,420.52 €
Bombos		75.88 €
Caixa e tamboril		1,550.26 €
Cavaquinho		259.65 €
Gaita de foles		513.06 €
Gaita de palhas		6.22 €
Matracas		15.45 €
Total geral		2,603.82 €

## Análise 7

Valores líquidos e respetivos valores dos descontos referentes às vendas ocorridas no 2º quadrimestre de 2012, detalhados por tipo de cliente e por produto, com possibilidade de análise apenas ao nível do fornecedor (do produto).

A		B
Date.Year	2012	
Date.Trimester	2	
Total Ilíquido Produtos - Faturacao Linhas	Rótulos de Coluna	▼
Rótulos de Linha	▼	

## Análise 8

Valores totais e respetivos custos dos produtos vendidos durante a primavera e verão de 2012, detalhados por localidade do cliente e família dos produtos.

Date.Season	(Vários itens)		
Date.Year	2012		
Valores			
Rótulos de Linha	Preço Custo	Total Final - Faturacao Linhas	Total Custo - Faturacao Linhas
⊖ Abrantes	21.95 €	88.56 €	55.85 €
Instrumentos cujo elemento vibratório é o próprio corpo	21.95 €	88.56 €	55.85 €
⊖ A-dos-Bispos	161.26 €	1,003.68 €	249.35 €
Aparelhos Electrónicos	83.99 €	125.46 €	83.99 €
Artigos de Pilhas	1.05 €	9.84 €	4.20 €
Instrumento cujo elemento vibratório é uma membrana retesada	8.72 €	693.72 €	26.16 €
Instrumentos cujo elemento vibratório é o próprio corpo	67.50 €	174.66 €	135.00 €
⊖ Albufeira	139.25 €	329.64 €	228.50 €
Acessórios	127.25 €	295.20 €	204.50 €
Instrumentos cujo elemento vibratório é o próprio corpo	12.00 €	34.44 €	24.00 €
⊖ Almodena	15.49 €	23.37 €	15.49 €
Instrumentos cujo elemento vibratório é o próprio corpo	15.49 €	23.37 €	15.49 €
⊖ Ameijeira	179.99 €	250.92 €	179.99 €
Aparelhos Electrónicos	179.99 €	250.92 €	179.99 €
⊖ Bairro Agro-tuela	950.50 €	1,239.84 €	950.50 €
Instrumento cujo elemento vibratório são cordas	950.50 €	1,239.84 €	950.50 €
⊖ Bairro Além do Rio	505.00 €	667.89 €	505.00 €
Instrumento cujo elemento vibratório são cordas	505.00 €	667.89 €	505.00 €
⊖ Bairro Boa Esperança	819.00 €	1,046.73 €	819.00 €
Instrumento cujo elemento vibratório são cordas	651.00 €	813.03 €	651.00 €
Instrumentos cujo elemento vibratório é o próprio corpo	168.00 €	233.70 €	168.00 €
⊖ Campolide	84.00 €	157.44 €	108.80 €
Artigos de Pilhas	1.14 €	8.61 €	3.94 €
Instrumentos cujo elemento vibratório é o ar	60.86 €	82.41 €	60.86 €
Instrumentos cujo elemento vibratório é o próprio corpo	22.00 €	66.42 €	44.00 €
⊖ Damaia	105.75 €	462.48 €	346.50 €
Acessórios	38.25 €	113.16 €	76.50 €
Instrumentos cujo elemento vibratório é o próprio corpo	67.50 €	349.32 €	270.00 €
⊖ Lisboa	957.50 €	1,284.12 €	957.50 €
Instrumento cujo elemento vibratório são cordas	957.50 €	1,284.12 €	957.50 €
⊖ Quarteira	963.22 €	2,257.05 €	1,426.44 €
Instrumento cujo elemento vibratório é uma membrana retesada	508.72 €	1,108.23 €	517.44 €
Instrumento cujo elemento vibratório são cordas	387.00 €	974.16 €	774.00 €
Instrumentos cujo elemento vibratório é o próprio corpo	67.50 €	174.66 €	135.00 €
⊖ Unknown	1,632.53 €	1,855.00 €	1,632.53 €
Instrumento cujo elemento vibratório são cordas	522.53 €	680.00 €	522.53 €
Instrumentos cujo elemento vibratório é o ar	1,110.00 €	1,175.00 €	1,110.00 €
<b>Total geral</b>	<b>6,535.43 €</b>	<b>10,666.72 €</b>	<b>7,475.44 €</b>

## Análise 9

Valores totais das vendas por cliente, com possibilidade de análise apenas ao nível da zona (do cliente), e por mês do ano de 2012, com possibilidade de análise apenas ao nível do trimestre, apenas dos produtos que pertencem às categorias “Ar”, “Cordas” e “Membranas”.

[illegible]

## Análise 10

Valores totais das vendas referentes às vendas ocorridas ao fim-de-semana dos produtos que pertencem às famílias “Acessórios”, “Amplificadores” e “Teclados”, detalhados por fornecedor, com possibilidade de análise até ao nível do produto, e por zona do cliente, com possibilidade de análise até ao nível do cliente.

Date.Weekend	Sim		
Descricao Familia	Acessórios		
Total Final		Rótulos de Coluna	
	Centro	Sul	Total geral
Rótulos de Linha			
MusiCarlos, Audio Ida	339.48 €	295.20 €	634.68 €
Caixa para Guitarras		68.88 €	68.88 €
Cordas para guitarras	226.32 €	113.16 €	339.48 €
Formas para guitarras	113.16 €	113.16 €	226.32 €
Total geral	339.48 €	295.20 €	634.68 €

## Análise 11

Número de faturasções e valor total das vendas efetuadas ao fim de semana e à semana detalhadas por Moeda no ano de 2012.

Date.Year	2012	
Número de linhas de faturasções e valor total das vendas efetuadas ao fim de semana		
Valores		
Rótulos de Linh	Número de linhas de Faturação	Total Final - Faturacao Linhas
<input checked="" type="checkbox"/> No	67	37868.67
EURO	67	37868.67
<input checked="" type="checkbox"/> Yes	20	4868.5
EURO	20	4868.5
Total geral	87	42737.17

## Análise 12

Lucros detalhados por tipo de cliente e mês, com possibilidade de análise até à zona.

Lucro - Faturacao Linhas	Rótulos de Coluna	
Rótulos de Linha	Cliente Final	Revendedor Total geral
<input checked="" type="checkbox"/> April	1,531.15 €	1,002.68 € 2,533.83 €
Centro	1,308.68 €	1,308.68 €
Espanha	222.47 €	222.47 €
Sul		1,002.68 € 1,002.68 €
<input checked="" type="checkbox"/> February	1,650.03 €	3,566.34 € 5,216.37 €
Centro		1,349.59 € 1,349.59 €
Espanha	142.32 €	142.32 €
Norte	1,305.28 €	2,126.67 € 3,431.95 €
Sul	202.43 €	90.07 € 292.50 €
<input checked="" type="checkbox"/> January	3,561.20 €	85.51 € 3,646.71 €
Centro	1,981.33 €	1,981.33 €
Espanha	193.52 €	193.52 €
Norte	1,386.35 €	1,386.35 €
Sul		85.51 € 85.51 €
<input checked="" type="checkbox"/> March	1,033.73 €	526.92 € 1,560.65 €
Centro	862.96 €	192.68 € 1,055.64 €
Norte	170.77 €	334.24 € 505.01 €
Total geral	7,776.11 €	5,181.44 € 12,957.56 €

## Análise 13

Quantidades vendidas no ano 2012, detalhado por estação do ano e vendedor, com possibilidade de análise apenas ao funcionário.

Date.Year	2012	
Quantidades vendidas no ano 2012, detalhado por estação do ano e vendedor, com po		
Quantidade - Faturacao Linhas	Rótulos de Coluna	
Rótulos de Linha	Inverno	Primavera Total geral
<b>Nuno Gama Pinto</b>	<b>147</b>	<b>36 183</b>
Carminda Sofia Costa das Neves	8	2 10
Elvira Cristina P. Andrade	15	10 25
Francisco Miguel Gamito	11	11
Jorge Junqueira	33	6 39
Nuno Gama Pinto	34	7 41
Sara Sofia Ramos	46	11 57
<b>Sara Sofia Ramos</b>	<b>100</b>	<b>22 122</b>
Carminda Sofia Costa das Neves	7	9 16
Elvira Cristina P. Andrade	7	4 11
Francisco Miguel Gamito		5 5
Jorge Junqueira	57	57
Nuno Gama Pinto	29	2 31
Sara Sofia Ramos		2 2
<b>Total geral</b>	<b>247</b>	<b>58 305</b>

## Análise 14

Totais custos no mês de Janeiro detalhados categoria do produto.

Date.Month Name	January	
Totais custos no mês de Janeiro detalhados por departar		
Rótulos de Linha	Total Custo - Produto Mês	
Ar	9,739.43 €	
Cases	100.00 €	
Cordas	13,459.15 €	
Corpo do Instrumento	1,662.00 €	
Interface	156.00 €	
Leitor de Cd	251.97 €	
Leitor de Mp3	1,872.00 €	
Membranas	2,358.90 €	
Peças	115.00 €	
Pilhas	65.16 €	
<b>Total geral</b>	<b>29,779.61 €</b>	



## Análise 15

Total IVA e Total Frete filtrados por zona Norte, cujo fornecedor pertence há localidade Alagoeiros e o Vendedor há localidade Olivais.

Localidade Fornecedor	Alagoeiros
Zona.Zona	Norte
Vendedor.Localidade	Olivais
Total IVA e Total Frete filtrados por zo	
<b>Valores</b>	
<b>Total IVA</b>	<b>Total Frete</b>
12.42 €	10.77 €

## Análise 16

Número de linhas de faturasões e quantidades vendidas no primeiro trimestre detalhadas por localidade do vendedor e por segmento do cliente com possibilidade de análise até à localidade do cliente.

Date.Trimester	1
Número de linhas de faturasões e quantidades vendidas no primeiro trimestre detalhadas por localidade do vendedor e por segm	
<b>Valores</b>	
<b>Rótulos de Linha</b>	<b>Quantidade - Faturacao Linhas</b>
<b>Número de linhas de Faturação</b>	
<b>Amadora</b>	<b>183</b>
Alto	78
Bairro Além do Rio	29
Campolide	14
Unknown	35
Baixo	85
Albufeira	13
Almodena	44
Bairro Além do Rio	13
Santarém	15
Médio	20
A-dos-Bispos	14
Damaia	6
Olivais	122
Alto	44
Abrantes	5
Lisboa	3
Quarteira	36
Baixo	6
Santarém	6
Médio	72
Alfragide	57
Ameljeira	2
Bairro Agro-tuela	6
Bairro Boa Esperança	5
Unknown	2
<b>Total geral</b>	<b>305</b>

## Análise 17

Total Fretes filtrado por segmento médio, detalhado por família com possibilidade de análise até ao produto.

Segmento	Médio
Total Fretes filtrado por segmento médio, detalhado por família com possibilidade de análise até ao produto	
Rótulos de Linha	Total Frete - Faturacao Linhas
[-] Acessórios	21.85 €
Caixa para Guitarras	17.15 €
Cordas para guitarras	4.70 €
[-] Aparelhos Electrónicos	10.20 €
Leitor de cd cdp 5300	7.65 €
Leitor de mp3 gmini xs 100	2.55 €
[-] Artigos de Pilhas	12.10 €
Pilha alcalina lr6	7.00 €
Pilha zinco-carbono r20	5.10 €
[-] Instrumento cujo elemento vibratório é uma membrana retesada	5.10 €
Caixa e tamboril	5.10 €
[-] Instrumento cujo elemento vibratório são cordas	121.86 €
Guitarra portuguesa	58.10 €
Rabeca chuleira	27.79 €
Viola amarantina	10.77 €
Viola de arame	12.60 €
Viola toeira	12.60 €
[-] Instrumentos cujo elemento vibratório é o ar	120.16 €
Acordeon gaita sanfona minuano 120 baixos	48.49 €
Acordeon gaita sanfona minuano 40 baixos	48.49 €
Gaita de foles	23.19 €
[-] Instrumentos cujo elemento vibratório é o próprio corpo	38.20 €
Búzios	9.80 €
Ferrinhos	17.77 €
Reque reque	10.64 €
<b>Total geral</b>	<b>329.47 €</b>

## Análise 18

Totais ilíquidos e totais finais do funcionário "Nuno Gama Pinto" detalhados por mês, com possibilidade de análise até ao dia da semana.

Funcionario.Nome	Nuno Gama Pinto	
Totais ilíquidos e totais finais do funcionário "Nuno Gama Pinto" detalhados por mês, com possibilic		
Valores		
Rótulos de Linha	Total Final - Faturacao Linhas	Total Ilíquido Produtos - Faturacao Linhas
[-] April	2,184.64 €	2,123.00 €
Saturday	2,184.64 €	2,123.00 €
[-] February	2,782.11 €	2,447.00 €
Friday	1,504.29 €	1,223.00 €
Monday	287.82 €	234.00 €
Tuesday	990.00 €	990.00 €
Total geral	4,966.75 €	4,570.00 €

Análise 19

Os lucros detalhados por localidade do cliente e com possibilidade de análise apenas por trimestre.

Lucro - Faturacao Linhas		Rótulos de Coluna				
Rótulos de Linha	1	2	3	4	Total geral	
Abrantes	32.71 €	32.71 €	32.71 €	32.71 €	32.71 €	
A-dos-Bispos	1,411.14 €	1,411.14 €	1,411.14 €	1,411.14 €	1,411.14 €	
Albufeira	186.65 €	186.65 €	186.65 €	186.65 €	186.65 €	
Alfragide	202.43 €	202.43 €	202.43 €	202.43 €	202.43 €	
Almodena	1,181.37 €	1,181.37 €	1,181.37 €	1,181.37 €	1,181.37 €	
Ameijeira	70.93 €	70.93 €	70.93 €	70.93 €	70.93 €	
Bairro Agro-tuela	801.34 €	801.34 €	801.34 €	801.34 €	801.34 €	
Bairro Além do Rio	3,340.60 €	3,340.60 €	3,340.60 €	3,340.60 €	3,340.60 €	
Bairro Boa Esperança	1,478.90 €	1,478.90 €	1,478.90 €	1,478.90 €	1,478.90 €	
Campolide	407.07 €	407.07 €	407.07 €	407.07 €	407.07 €	
Damaia	115.98 €	115.98 €	115.98 €	115.98 €	115.98 €	
Lisboa	399.97 €	399.97 €	399.97 €	399.97 €	399.97 €	
Quarteira	920.68 €	920.68 €	920.68 €	920.68 €	920.68 €	
Santarém	1,849.47 €	1,849.47 €	1,849.47 €	1,849.47 €	1,849.47 €	
Unknown	558.31 €	558.31 €	558.31 €	558.31 €	558.31 €	
Total geral	12,957.56 €	12,957.56 €	12,957.56 €	12,957.56 €	12,957.56 €	

Análise 20

Os lucros em euros e dollars obtidos pela família "Pilhas".

Moeda Origem.Abreviatura	EUR		
Moeda Destino.Abreviatura	USD		
Classificacao Familia	Pilhas		
Os lucros em euros e dollars obtidos pela família "Pilhas"			
Valores			
Rótulos de Linha	Fator Cambio	Lucro - Faturacao Linhas	Lucro (USD)
Pilha alcalina lr6	1.215875	30.64 €	\$37.25
Pilha zinco-carbono r14	1.215875	29.28 €	\$35.60
Pilha zinco-carbono r20	1.215875	5.64 €	\$6.86
Total geral	1.215875	65.56 €	\$79.71

## Opções

---

Nesta secção encontra-se algumas das opções mais relevantes que foram tomadas no desenvolvimento deste projeto.

### Novos campos

Na criação da tabela de factos “Faturação Linhas” notou-se que dados referentes ao lucro e ao frete não estavam presentes e como tal, para tornar a tabela mais completa, os campos “Lucro”, “FreteUnidade” e “TotalFrete” foram calculados e adicionados. O campo “Lucro” consiste numa simples subtração entre o campo “TotalCusto” e “TotalFinal”; o campo “TotalFrete” é calculado através do campo “TotalFrete” existente na tabela da Staging Area “Documentos Faturação”, o cálculo consiste em dividir o total dos fretes pelo número de linhas na faturação e pela quantidade; o campo “FreteUnidade” é igual ao campo “TotalFrete” mas, no cálculo, não é dividido pela quantidade.

### Taxas de câmbios

Para suportar vários tipos de moedas, foi criada uma tabela de factos chamada “Currency Rate” que contém as taxas de câmbio para 6 tipos de moedas diferentes, esta tabela é composta por duas chaves estrangeiras (também fazem parte da chave primária) que referenciam a dimensão “Moeda” e guardam o nome da moeda original e o nome da moeda para a qual se quer converter o valor, existe ainda o campo “Fator Câmbio” que contém a taxa de câmbio usada no cálculo da conversão. O preenchimento da tabela é feita a partir de um ficheiro Excel que contém as abreviaturas da moeda origem e da moeda destino mais o fator câmbio entre elas.

### Surrogate Keys

As *surrogate keys* das dimensões são geradas por *auto-number* aquando da criação da linha na dimensão. A surrogate key da Dimensão Funcionario por outro lado é criada no processo ETL na tabela intermédia da Staging Area com *auto-number* e enviada depois para a dimensão. Isto porque é necessário fazer a correspondência entre o funcionário e o seu respetivo chefe tanto nos funcionários já existentes na “DimFuncionarios” como nos funcionários que estão a ser carregados nesse momento.

### Dimensão “lixo”

Foi criada uma dimensão especial que guarda valores da faturação que não são importantes na tabela de factos e não são relevantes o suficiente para terem um lugar numa tabela de dimensão normal. A dimensão de lixo criada é a “Junk Faturação” e contém os campos referentes ao documento da faturação, ao tipo de cliente e o armazém. Esta dimensão é preenchida antes do carregamento das tabelas de factos com todas as combinações existentes nas tabelas de faturação das Staging Area.

## Junção de tabelas

O sistema operacional contém tabelas que para o Data Mart não faz sentido serem convertidas para dimensões e podem ser agregadas com outras tabelas. Neste caso, as seguintes tabelas foram agregadas:

- A tabela “Armazéns” foi juntada à tabela “lixo” uma vez que era apenas utilizado na tabela de factos “Faturação Linhas”;
- As tabelas “Categorias” e “Famílias” foram juntadas com a tabela “Produtos” criando assim apenas uma dimensão que guarda todos os dados referentes ao produto, categoria e família de produtos;
- A tabela “Vendedor” foi juntada à tabela “Funcionário” uma vez que os dados existentes dos vendedores encontravam-se repetidos na tabela de funcionários.

## Outras opções

- Algumas tabelas do Sistema Operacional não foram utilizadas neste processo ETL porque não são referentes à área de negócio a que este Data Mart se destina.
- No projeto SSIS, foi utilizado o componente “Component Script” para a tarefa “Get & Set DataUltimoLoading” e no contentor “Create Partitions”.
- Sempre que o processo ETL executa a tarefa de recarregamento da tabela de factos agregada ao mês, os dados são sempre recalculados.

## Melhoramentos possíveis

---

Nesta secção estão alguns dos melhoramentos que podem ser feitos para tornar o Data Mart mais completo/rápido.

- Passar o campo “Armazém” da tabela “lixo” para uma dimensão à parte é um melhoramento que podia ser realizado para o caso de haver outras tabelas que fossem criadas no futuro e quisessem fazer uso desse campo;
- Adicionar mais partições nas dimensões mais “pesadas” como o produto e o cliente, a adição de mais partições permitem obter um ganho de performance com o custo de ser necessário mais espaço em disco;
- Adicionar diferentes tipos de agregações que abrangem um variado tipo de queries. Neste caso, as agregações estão feitas a nível de produto e mês ou cliente e mês, ainda há espaço para agregações a nível só do mês ou das zonas, por exemplo;