



Universidade do Minho

Mestrado Integrado em Engenharia Informática
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Lectivo de 2020/2021

SupplyChain

Comércio e Distribuição de Produtos

João Duarte a81761,
Simão Brito a89482,
Luís Magalhães a89528,
Patrícia Pereira a89578,
Jaime Abreu a89598.

Dezembro 2020

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

SupplyChain

Comércio e Distribuição de Produtos

João Duarte a81761,
Simão Brito a89482,
Luís Magalhães a89528,
Patrícia Pereira a89578,
Jaime Abreu a89598.

Dezembro 2020

Resumo

Este documento, no âmbito da Unidade Curricular de Base de Dados, retrata de forma detalhada a realização de uma base de dados relacional para um sistema de cadeias de fornecimento para uma empresa, SupplyChain.

Este relatório encontra-se dividido em cinco partes relacionadas com o desenvolvimento do trabalho.

Primeiramente, analisamos a viabilidade do tipo de empresa Supply Chain. Iremos fazer uma contextualização e fundamentação da base de dados relacional, seguido de um levantamento e uma análise de requisitos de descrição, de exploração e de controlo. Posteriormente definiremos o modelo conceptual através da ferramenta brModelo, tendo em conta os requisitos levantados. Daqui partiremos para a conceção de um modelo lógico e por fim concluímos com a implementação física da base de dados onde passamos o modelo lógico para um modelo físico e traduzimos as questões de utilizador e as transações em código SQL.

Área de Aplicação: Desenho e arquitetura de Sistemas de Bases de Dados no âmbito do estudo e desenvolvimento de um sistema de cadeias de fornecimento.

Palavras-Chave: Base de Dados, Base de Dados Relacionais, Levantamento de Requisitos, Análise de Requisitos, Entidades, Atributos, Relacionamentos, Modelo Conceptual, Modelo Lógico, Modelo Físico, BrModelo, SQL.

Índice

Resumo	1
Índice	1
Índice de Figuras	3
Índice de Tabelas	4
1) Introdução - Definição do Sistema	1
1.1) Contextualização da aplicação	1
1.2) Apresentação do caso de estudo	1
1.3) Motivação e Objectivos	2
1.4) Estrutura do relatório	2
2) Levantamento e Análise de Requisitos	3
2.1) Método de levantamento e de análise de requisitos adotado	3
2.2) Requisitos levantados	3
2.2.1) Requisitos de descrição	3
2.2.2) Requisitos de exploração	4
2.2.3) Requisitos de controlo	5
2.3) Análise e validação geral dos requisitos	5
3) Modelação Conceptual	5
3.1) Apresentação da abordagem de modelação realizada	6
3.2) Identificação e caracterização das entidades	6
3.3) Identificação e caracterização dos relacionamentos	7
3.4) Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	7
3.5) Detalhe ou generalização de entidades	10
3.6) Apresentação e explicação do diagrama ER	10
3.7) Validação do modelo de dados produzido	11

4) Modelação Lógica	11
4.1) Construção e validação do modelo de dados lógico	11
4.2) Desenho do modelo lógico	14
4.3) Validação do modelo com interrogações do utilizador	14
4.4) Reavaliação e revisão do modelo lógico produzido	15
5) Implementação Física	15
5.1) Seleção do sistema de gestão de bases de dados	16
5.2) Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL	16
5.3) Tradução das interrogações do utilizador para SQL (alguns exemplos)	22
5.4) Escolha, definição e caracterização de índices em SQL	25
5.5) Escolha, definição e caracterização de índices em SQL	26
5.6) Estimativa do espaço em disco da base de dados e taxa de crescimento anual	26
5.7) Definição e caracterização das vistas de utilização em SQL (alguns exemplos)	28
5.8) Revisão do sistema implementado	28
6) Conclusões e Trabalho Futuro	29
Lista de Siglas e Acrónimos	29
Anexos	30

Anexos

I. **Erro! Indicador não definido.**

Índice de Figuras

Figura 1: Diagrama ER.....	10
Figura 2: Modelo Lógico.....	14
Figura 3: Código SQL para Criação de Tabela Produto_Tipo.....	17
Figura 4:Código SQL para Criação de Tabela Produto_Fisico.....	18
Figura 5: Código SQL para Criação de Tabela Componentes	19
Figura 6: Código SQL para Criação de Tabela Elementos	20
Figura 7: Código SQL para Criação de Tabela Empresa	21
Figura 8: Código SQL para Criação de Tabela Lote	22
Figura 9: Código SQL para consultar Lista de todos os produtos	23
Figura 10: Código SQL para consultar informações de todos os Lotes	23
Figura 11: Código SQL para consultar Lista de todas as empresas	23
Figura 12: Código SQL para consultar Lista de todos os fornecedores	23
Figura 13: Código SQL para consultar Lista de todos os armazenistas.....	23
Figura 14: Código SQL para consultar Lista de todas as produtoras	23
Figura 15: Código SQL para consultar Lista de todos os retalhos	23
Figura 16: Código SQL para consultar Lista da validade de produtos físicos	23
Figura 17: Código SQL para consultar Componentes dos produtos	24
Figura 18: Código SQL para consultar Todos os produtos fisicos.....	24
Figura 19: Código SQL para consultar top empresas com mais lotes recebidos	24
Figura 20: Código SQL para calcular preço de um lote.....	24
Figura 21: Código SQL para registrar empresa.....	25
Figura 22: Código SQL para registo de um produto.....	25
Figura 23: Código SQL para criação de um lote	25
Figura 24: Código SQL para criação de um tipo de produto	25
Figura 25: Código SQL para a criação dos Índices sobre as colunas Data_Validade e Data	26
Figura 26: Código SQL para lista de todos os tipos de produtos	28
Figura 27: Código SQL para lista de todos os produtos.....	28
Figura 28: Código SQL para lista de todas as empresas	28
Figura 29: Código SQL para lista de todos os lotes	28

Índice de Tabelas

Tabela 1: Entidades	6
Tabela 2: Relacionamentos.....	7
Tabela 3: Atributos	7
Tabela 4: Tamanho de cada entrada na respetiva Tabela	26

1) Introdução - Definição do Sistema

A empresa SupplyChain, um sistema de cadeias de fornecimento necessita da criação de uma base de dados que suporte a gestão de encomendas. A base de dados servirá para o armazenamento e correlação da informação sobre os produtos existentes, de modo a que em qualquer altura se possa consultar o estado destes. Com esta finalidade, o sistema deve conter um conjunto de métodos/funcionalidades que permitam obter certas informações, como a quantidade de encomendas a ser processadas, o valor faturado, entre outras.

O desenvolvimento deste sistema passará, pelas seguintes fases: uma definição e contextualização do sistema, levantamento e análise de requisitos, modelação conceptual, modelação lógica, implementação física.

O primeiro capítulo deste relatório, é relativo à definição do sistema, sendo exposto o contexto da aplicação assim como a fundamentação para a implementação. No final é apresentada uma análise de viabilidade do processo.

1.1) Contextualização da aplicação

A empresa SupllyChain, está desde 1996 no mercado. É o distribuidor oficial de marcas como a LEVIS, Coca-cola e JOM. Comercializa uma vasta gama de produtos.

A empresa procura uma maior satisfação dos seus clientes, oferecendo-lhes um fornecimento mais rápido e sem falhas. Para isto necessita de uma aplicação que permita aos trabalhadores ter acesso ao estado das suas encomendas, a hora e dia a que esta será entregue, o destino e a origem e o conteúdo da encomenda. Para suportar esta funcionalidade, a empresa necessita de implementar um sistema de base de dados.

1.2) Apresentação do caso de estudo

Com o aumento do consumo e procura gastronómica no país, várias pequenas e médias empresas como cafés, bares, restaurantes, etc, têm procurado empresas como a SupplyChain para lhes fornecer os produtos que necessitam para vender, fazendo com que este tipo de empresa tenha uma procura elevada.

Para conseguir gerir melhor as encomendas que lhe chegam, e conseguir uma maior confiança dos seus clientes a empresa pretende que os seus colaboradores consigam saber o estado da sua encomenda de forma online.

1.3) Motivação e Objectivos

Os pais da Patrícia têm um restaurante. Como se verifica na maior parte dos casos, este também depende de cadeias de fornecimento para garantir o bom funcionamento e assegurar uma maior variedade de escolha de produtos aos seus clientes. Ora a Patrícia explicou-nos que nem todas as cadeias de fornecimento funcionam baseadas em sistemas informáticos, sendo algumas retrógradas ao ponto de ainda manter todos os seus registos num formato físico, ocupando gavetas e gavetas com todos os dados das transações que foram ocorrendo ao longo de muitos anos.

Assim, concordando que não faz sentido as cadeias de fornecimento descartarem a possibilidade de aderir a uma base de dados digital, o nosso objetivo é desenvolver uma simples mas eficaz base de dados que traduza o processo de uma das muitas cadeias de fornecimento que existem no nosso país.

1.4) Estrutura do relatório

No seguinte capítulo deste relatório procedermos ao levantamento, descrição e análise de requisitos, ao método pela qual decidimos optar.

Na 3ª parte apresentaremos também o nosso modelo conceptual e a abordagem de modelação que realizamos e identificamos as nossas entidades bem como os seus relacionamentos e identificamos também as associações dos atributos com as entidades e relacionamentos.

A fase seguinte diz respeito à modelação lógica, como procedemos para obter o nosso modelo, e à validação do modelo.

A parte final do projeto e que se encontra no capítulo 5, é a implementação física, onde faremos uma descrição detalhada de cada um dos processos anteriores.

2) Levantamento e Análise de Requisitos

No Capítulo 1 foi definido o contexto em que a base de dados iria ser implementada.

É necessário agora identificarmos a informação que o sistema terá de ser capaz de armazenar. Iremos, então, proceder ao levantamento de requisitos, recolhendo e analisando informação sobre o *modus operandi* da empresa e a sua interação com o cliente.

2.1) Método de levantamento e de análise de requisitos adotado

Nesta fase recorreremos à observação direta e testemunho de clientes e funcionários da empresa.

Recolhemos alguma informação sobre mudanças que os clientes gostariam de ver no contacto com a empresa. Falámos também com as pessoas responsáveis pelo transporte e com os responsáveis pelas encomendas de forma a tentar perceber o funcionamento da empresa e a forma como abordam os clientes.

2.2) Requisitos levantados

Analisando a informação que recolhemos, prosseguimos à definição dos requisitos que a base de dados deverá suportar. Procuramos atribuir à informação uma maior estruturação, de forma a obter uma diferenciação entre as perspectivas dos vários tipos de funcionários, uma vez que estes têm diferentes objetivos no que diz respeito à base de dados.

2.2.1) Requisitos de descrição

Requisitos necessários para o funcionamento da nossa cadeia de fornecimento:

Produto

Representa uma unidade de um produto e tem associado as seguintes especificações:

- Registado através de um identificador
- Tem um preço
- Dispõe de um peso
- Possui uma quantidade
- Apresenta um registo das dimensões deste
- Retém ainda uma data de validade no caso de ser perecível

Tipo de produto

Representa o tipo de um produto e tem associado as seguintes especificações:

- Registado através de um identificador
- Tem um preço
- Dispõe de um peso em função da sua área (kg/m²)

Estes dois últimos dados serão importantes para a faturação

Lote

Representa um conjunto de produtos pronto a expedir para uma empresa e tem associado as seguintes especificações:

- Registado através de um identificador
- Tem registada a data em que é expedido para a empresa

Empresa

Representa o destino de uma expedição de um lote e tem associada as seguintes especificações:

- Registada através do nome que vai ser o seu identificador
- Possui um contacto telefónico associado
- Dispõe de uma morada associada, sendo esta constituída por país, distrito, código postal, localidade e rua
- Conta com ainda um tipo de empresa relativa a ela própria

2.2.2) Requisitos de exploração

Requisitos necessários para a manipulação da base de dados:

- Lista de todos os produtos
- Informações de todos os Lotes
- Lista de todas as empresas
- Lista de todos os fornecedores
- Lista de todos os armazenistas
- Lista de todas as produtoras
- Lista de todos os retalhos
- Lista da validade de produtos físicos
- Componentes dos produtos
- Todos os Produtos Físicos
- Top Empresas com mais lotes recebidos
- Calcular preço de um lote

2.2.3) Requisitos de controlo

Sendo o objetivo de uma base de dados o armazenamento eficaz de informação num sistema informático, é importante que os seus utilizadores tenham acesso a todos os dados que lhes dizem respeito. No entanto, é preciso também garantir que não tenham acesso a informações que não necessitam.

As empresas podem:

- Criar uma empresa
- Criar um produto
- Criar um lote

Um administrador pode:

- Criar um tipo de produto

2.3) Análise e validação geral dos requisitos

Os requisitos de exploração levantados são descritivos do modo de funcionalidade da nossa empresa. Estes englobam e descreve, as componentes que a constituem e são referentes às interações entre estes.

Os requisitos de exploração ajudam-nos a identificar o papel do utilizador da base de dados, tendo em conta que os requisitos identificam os dados que serão guardados e o que se pretende do sistema.

3) Modelação Conceptual

Concluída a definição dos requisitos e tendo já realizado uma análise destes, nesta fase vamos é a modelação conceptual.

O modelo conceptual é independente aos detalhes de implementação e representa o modelo de informação usado pela empresa. De forma a simplificar a definição e interpolação do modelo, este é acompanhado de documentação, composta de um diagrama ER e dados sobre as entidades, relações e atributos identificados.

3.1) Apresentação da abordagem de modelação realizada

A abordagem de modelação que utilizamos na realização do modelo adaptada à metodologia Connolly & Begg (2004), que visa orientar a modelação através de alguns dos seguintes passos:

- 1) Identificar os tipos de entidades que existem
- 2) Identificar os vários tipos de relacionamento
- 3) Identificar e associar os atributos para cada tipo de entidade ou relacionamento
- 4) Determinar o domínio de atributos
- 5) Determinar chaves candidatas, primárias e estrangeiras
- 6) Considerar a generalização de entidades o uso de detalhe
- 7) Verificar que o modelo conceptual suporta as transações necessárias
- 8) Validação do modelo de dados com o utilizador

Assim sendo, foi possível definir o modelo conceptual da maneira que vamos apresentar nas secções seguintes.

3.2) Identificação e caracterização das entidades

Primeiramente definimos as entidades existentes que identificamos pela análise de requisitos. desta forma chegamos às seguintes entidades:

Tabela 1: Entidades

Entidade	Descrição	Ocorrência
Produto_Tipo	Termo que descreve o tipo do produto em causa	Um Tipo de um Produto servirá para categorizar e agregar os vários produtos em stock
Produto_Fisico	Termo que representa uma unidade física de um determinado produtos	Um Produto Fisico estará associado à criação de Lotes, pode ser usado como Componente para a criação de outros Produtos e também é caracterizado por um certo Tipo de Produto.
Lote	Agregação de um determinado tipo de produto para ser exportado	Um Lote está associado aos Produto_Fisicos
Empresa	Empresa em causa	Uma Empresa estará associada aos Lotes

Componentes	Produtos usados para a criação de um outro produto	Competes serão associados a Produtos_Fisicos
-------------	--	--

3.3) Identificação e caracterização dos relacionamentos

Tabela 2: Relacionamentos

Entidade	Cardinalidade	Relacionamento	Cardinalidade	Entidade
Produto_Tipo	(1,1)	Pertence	(1,n)	Produto_Fisico
Produto_Fisico	(0,1)	Componentes	(0,n)	Produto_Fisico
	(0,n)	Tem	(1,n)	Lote
Lote	(0,n)	Destino	(1,1)	Empresa
	(0,n)	Origem	(1,1)	Empresa

3.4) Identificação e caracterização da associação dos atributos com as entidades e relacionamentos

Tabela 3: Atributos

Entidade	Atributo	Descrição	Data Type	Null	Multivalorado	Derivado
Produto_Tipo	idProduto_Tipo	Identifica um único tipo de produto	INT	Não	Não	Não
	Preço	Valor genérico por unidade	FLOAT	Não	Não	Não
	Peso_m2	Valor da divisão entre o valor do	FLOAT	Não	Não	Não

		peso desse Produto com a área que ele ocupa				
	Nome	Nome do um específico produto	VARCHAR(45)	Não	Não	Não
Produto_Fisico	idProduto_Fisico	Identifica uma unidade de um determinado produto	INT	Não	Não	Não
	Preço	Preço absoluto do produto	FLOAT	Não	Não	Sim
	Quantidade	Número de unidades do Produto no conjunto em causa	INT	Não	Não	Não
	Dimensões	Volume do conjunto total	FLOAT	Não	Não	Não
	Peso	Peso total do conjunto	FLOAT	Não	Não	Não
	Data_Validade	Data limite até o conjunto ficar fora de validade	DATE	Não	Não	Não
	Composto	Condição que indica se o produto é composto por outros produtos	VARCHAR	Não	Não	Não
Lote	ID	Número identificador de um respetivo lote	INT	Não	Não	Não
	DATA	Data da sua transferência	DATE	Não	Não	Não
Empresa	idEmpresa	Identifica uma única empresa	INT	Não	Não	Não
	Nome	Nome da empresa	VARCHAR(45)	Não	Não	Não
	Contacto	Número telefónico para contactar a respectiva empresa	FLOAT	Não	Não	Não

	Morada	País	Local onde situa a empresa	VARCHAR(45)	Não	Não	Não
		Rua		VARCHAR(45)	Não	Não	Não
		Código_Postal		VARCHAR(45)	Não	Não	Não
	Tipo		Tipo caraterístico da empresa	VARCHAR(45)	Não	Não	Não

3.5) Detalhe ou generalização de entidades

3.6) Apresentação e explicação do diagrama ER

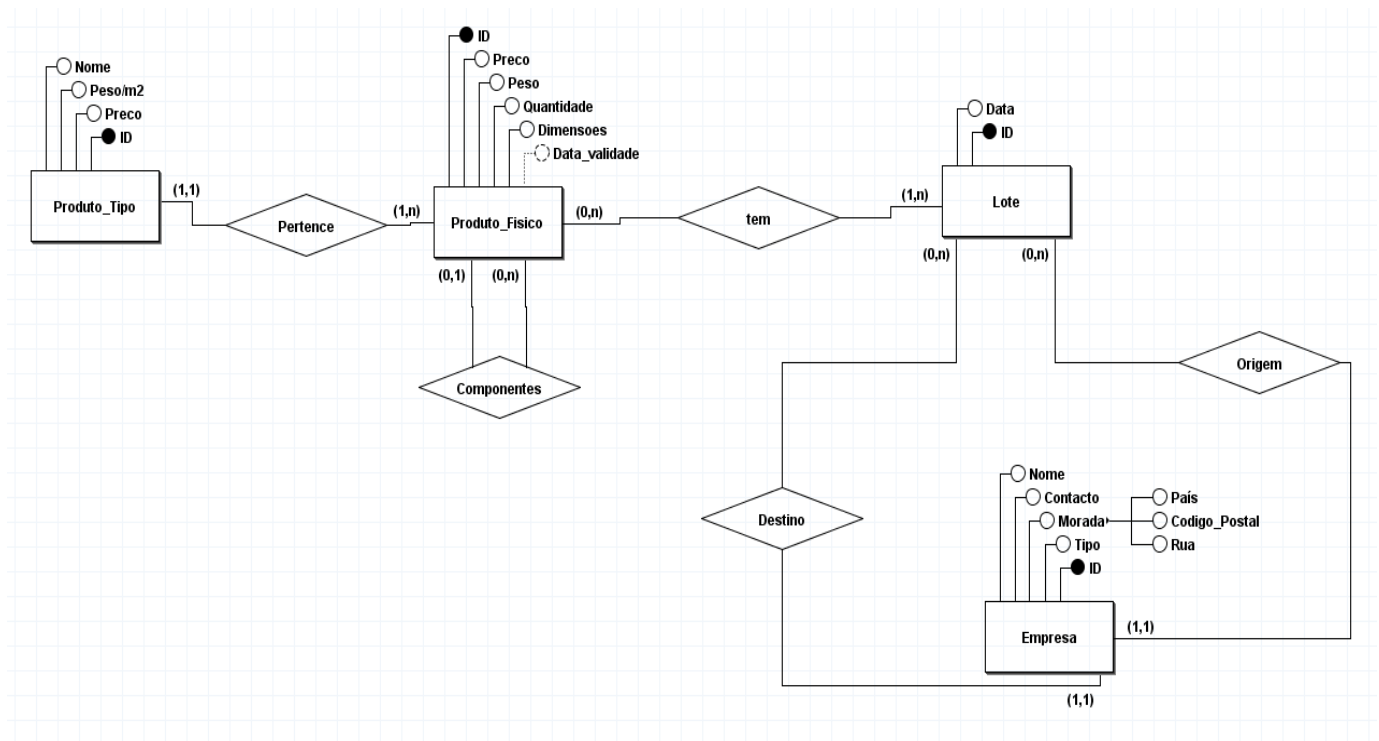


Figura 1: Diagrama ER

Pela análise do diagrama acima apresentado, verificamos que as entidades e atributos vão ao encontro dos requisitos de exploração definidos:

Na entidade "Produto_Tipo" verificamos a existência de 3 chaves identificativas: "Tipo", "Peso/m2" e "Preço". Esta entidade é identificada pelo seu tipo de produto, ou seja, se é por exemplo, uma coca-cola de 1L garrafa de plástico, ou uma cola-cola de 0.35 garrafas de plástico. associados a este produto está também o preço e o peso/m2 que são importantes a nível de faturação.

A entidade "Produto_Fisico" é composta pelas chaves que achamos necessárias na identificação do produto, sendo o seu "ID" a chave primária, assim como a entidade "Encomenda"

A entidade "Encomenda" tem vários atributos necessários para a identificação da empresa sendo o seu "Nome" chaves candidatas, sendo esta a chave primária uma vez que tem um carácter permanente.

3.7) Validação do modelo de dados produzido

Para concluir a fase final do modelodados, procedeu-se à revisão deste. Após confirmarmos que este representa o pretendido e não necessitamos de fazer alterações ou acrescentos, dêmo-lo como validado e podemos então passar à fase seguinte do projecto.

4) Modelação Lógica

Procedemos agora para a construção do modelo lógico foi orientada pelo modelo de dados relacional.

Nos seguintes tópicos procedemos à descrição do processo de construção deste modelo. Explicando as validações com as interrogações do utilizador. No fim é feita a revisão do modelo.

4.1) Construção e validação do modelo de dados lógico

O modelo lógico deve ser capaz de identificar as entidades, relacionamento e atributos, tendo de estabelecer relações necessárias para tal.

Em cada etapa da análise do modelo concetual identificamos ocorrências dos certos elementos. Derivando a sua representação no modelo, a ordem foi a seguinte:

- 1) Entidades forte
 - **Produto_tipo**(idProduto_Tipo, Peso_m2,Preco)
Chave Primária - idProduto_Tipo
 - **Produto_Fisico**(idProduto,Preco,Peso,Quantidade,Dimensoes,Data_Validade, Composto)

- Chave Primária** - idProduto
 - **Lote**(idLote,data)
 - Chave Primária** - idLote
 - **Empresa**(idEmpresa,Nome,Contacto,Pais,Rua,Codigo_Postal,Tipo)
 - Chave Primária** - nome
- 2) Entidades fracas
 - **Componentes**(Produto_origem,componentes)
 - Chave Estrangeira** - Produto_origem
 - Chave Estrangeira** - componentes
 - **Elementos**(fk_idLote,fk_Produto_Fisico)
 - Chave Estrangeira** - fk_idLote
 - Chave Estrangeira** - fk_idProduto_Fisico
- 3) Relacionamentos binários de um para muitos
 - **Produto_Fisico**(idProduto,Preco,Peso,Quantidade,Dimensoes,Data_Validade,Composto,fk_idProdutoTipo)
 - Chave Primária** - idProduto
 - Chave Estrangeira** - fk_idProdutoTipo tipo
 - **Lote**(idLote,data,fk_idEmpresa_Origem, fk_idEmpresa_Destino)
 - Chave Primária** - idLote
 - Chave Estrangeira** - fk_idEmpresa_Origem **referencia** Empresa(idEmpresa,Nome,Contacto,Pais,Rua,Codigo_Postal,Tipo)
 - Chave Estrangeira** - fk_idEmpresa_Destino **referencia** Empresa(idEmpresa,Nome,Contacto,Pais,Rua,Codigo_Postal,Tipo)
 - **Componentes**(Produto_origem,componentes)
 - Chave Estrangeira** - Produto_origem **referencia** Produto_Fisico(idProduto,Preco,Peso,Quantidade,Dimensoes,Data_Validade,Composto,fk_idProdutoTipo)
 - Chave Estrangeira** - componentes **referencia** Produto_Fisico(idProduto,Preco,Peso,Quantidade,Dimensoes,Data_Validade,Composto,fk_idProdutoTipo)
 - **Elementos**(fk_idLote,fk_Produto_Fisico)
 - Chave Estrangeira** - fk_idLote **referencia** Lote(idLote,data)
 - Chave Estrangeira** - fk_idProduto_Fisico **referencia** Produto_Fisico(idProduto,Preco,Peso,Quantidade,Dimensoes,Data_Validade,Composto,fk_idProdutoTipo)
- 4) Relacionamentos de um para um
- 5) Relacionamentos recursivos de um para um
- 6) Relacionamentos do tipo superclasse/subclasse
- 7) Relacionamentos de muito para muitos

- 8) Relacionamentos complexos
- 9) Atributos multivalorados

Concluimos que as entidades que representamos no modelo são fortes. Foram derivadas novas relações, que deram origem às entidades "Componentes" e "Elementos".

Identificamos sete relacionamentos de um para muitos.

Não foi derivada mais nenhuma relação dos outros enumerados, nem nenhum atributo multivalorados.

4.2) Desenho do modelo lógico

Com as alterações derivadas do ponto anterior, surgiu o seguinte esquema para o modelo lógico:

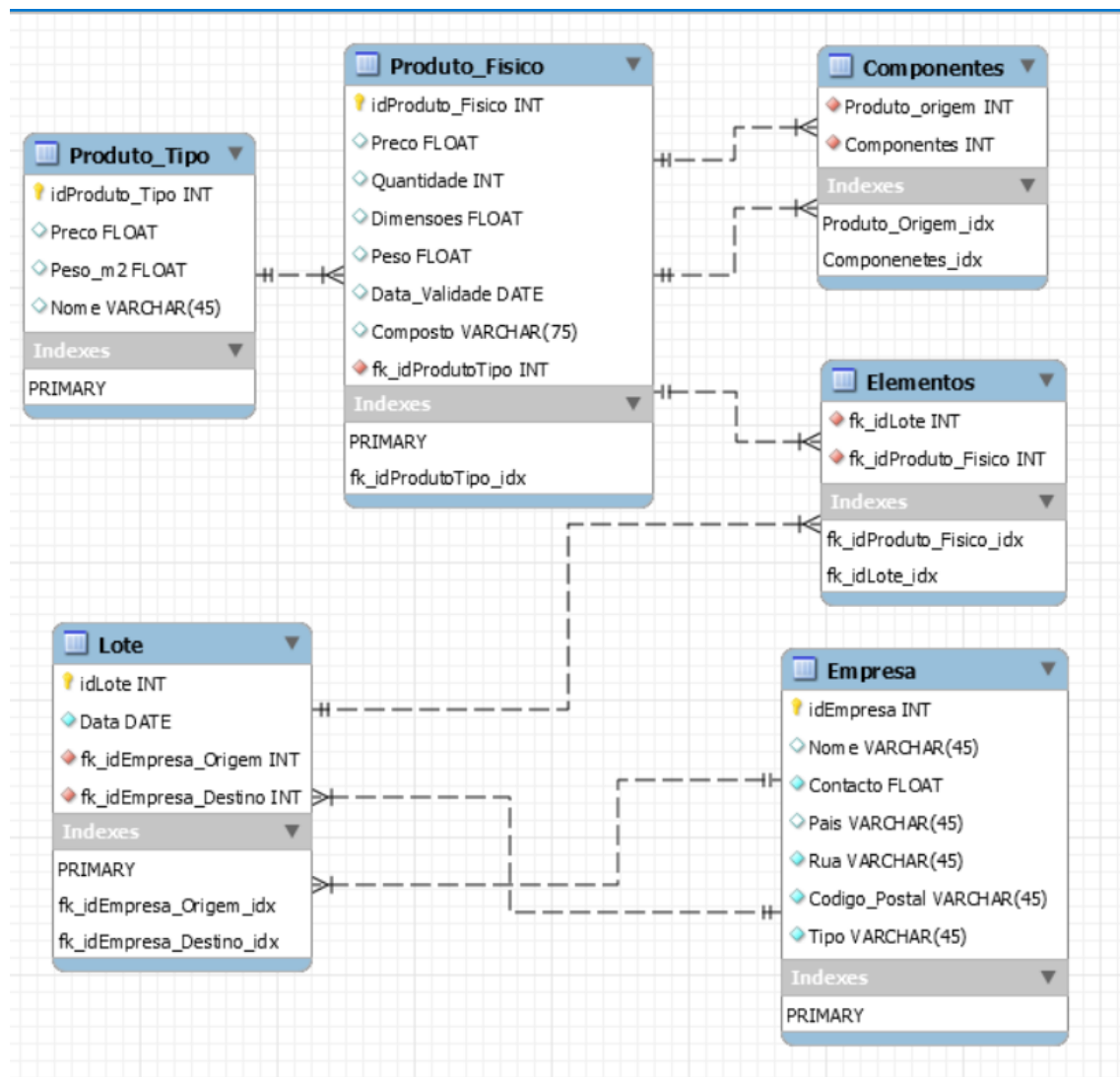


Figura 2: Modelo Lógico

4.3) Validação do modelo com interrogações do utilizador

Que tipo de empresas existem?

É necessário verificar o atributo Tipo na tabela das Empresas. Posteriormente, perceber-se-á que existem quatro tipos de empresas: fornecedora, armazenista, produtoras e de retalho.

Posso consultar o meu histórico de transações?

É necessário consultar todas as transições de lotes em que uma empresa esteve envolvida procurando pelo seu ID nas fk dos lotes.

Como posso saber o preço de um lote?

Primeiramente é preciso obter todos os produtos pertencentes ao lote selecionado, utilizando o ID do lote na tabela de elementos. Seguidamente, somamos o valor do atributo Preço de todos esses produtos.

Que tipo de listagens posso consultar?

É possível consultar a lista de empresas (por tipo ou todas), lotes, produtos e produtos tipo.

Posso procurar paletes fora de validade?

Para encontrar paletes fora de validade é necessário comparar o atributo da validade (se este não for null, dado que nem todos os produtos possuem uma data de validade) com a data atual.

4.4) Reavaliação e revisão do modelo lógico produzido

Para concluir a fase final do modelo lógico, procedeu-se à revisão deste. Assim, verificando que esta responde a todos os requisitos necessários de um sistema de cadeias de fornecimento, descartamos a necessidade de uma reavaliação e procedemos a considerar, então, o nosso modelo como válido.

5) Implementação Física

O processo de construção do sistema de base de dados passou pela sua implementação física, povoamento, exploração, e monitorização da informação armazenada.

Primeiramente traduzimos o esquema lógico do capítulo anterior para o SGBB escolhido.

Implementado o esquema físico, traduzimos as interrogações do utilizador e das transações para código SQL.

Neste capítulo fazemos uma descrição detalhada de cada processo, assim como apresentamos uma estimativa do espaço que a base de dados ocupará em disco.

5.1) Seleção do sistema de gestão de bases de dados

Com o objetivo de desenvolver o esquema físico da base de dados tivemos de escolher qual o SGBD a utilizar. Encontrando o domínio do modelo de dados relacional, usamos o *MySQL* uma vez que este disponibiliza mecanismos de controlo de concorrência, pois permite eliminar problemas que seriam graves no contexto da aplicação. O *MySQL* permite também a definição de perfis de utilização.

5.2) Tradução do esquema lógico para o sistema de gestão de bases de dados escolhido em SQL

No processo de construção físico identificamos as relações base, produzidas aquando da definição do esquema lógico.

As relações que temos são:

- **Relação Poduto_Tipo:**

Domínio Id Produto_Tipo:	inteiro
Domínio Preco:	valor entre 000.00-999.99
Domínio Peso por m2:	valor entre 000.00-999.99
Domínio Nome:	string de comprimento variável, comprimento 75

```
Produto_Tipo(  
    idProduto_Tipo      ID Produto Tipo      NOT NULL,  
    Preco               Preco               NOT NULL,  
    Peso               Peso               NOT NULL,  
    Nome               Nome               NOT NULL,  
    CHAVE PRIMARIA(IDpPRODUTO_TIPO));
```

```

-----
-- Table `SupplyChain`.`Produto_Tipo`
-----

CREATE TABLE IF NOT EXISTS `SupplyChain`.`Produto_Tipo` (
  `idProduto_Tipo` INT NOT NULL,
  `Preco` FLOAT NULL,
  `Peso_m2` FLOAT NULL,
  `Nome` VARCHAR(45) NULL,
  PRIMARY KEY (`idProduto_Tipo`))
ENGINE = InnoDB;

```

Figura 3: Código SQL para Criação de Tabela Produto_Tipo

- **Relação Produto_Fisico:**

Domínio id Produto_Fisico:	inteiro
Domínio Preço:	valor entre 000.00-999.99
Domínio Quantidade:	inteiro
Domínio Peso:	valor entre 000.00-999.99
Domínio Dimensões:	valor entre 000.00-999.99
Domínio Data de validade:	temporal
Domínio Composto:	string de comprimento variável, comprimento 75
Domínio id Produto Tipo	inteiro

```

Produto_Fisico(
  idProduto_Fisico    iD Produto Físico    NOT NULL,
  Preco               Preço                NOT NULL,
  Quantidade           Quantidade           NOT NULL,
  Dimensao            Dimensão             NOT NULL,
  Peso                Peso                 NOT NULL,
  Data_validade       Data de validade     NOT NULL,
  Composto            Composto             NOT NULL,
  CHAVE PRIMÁRIA(idProduto_Fisico)
  CHAVE ESTRANGEIRA(fk_idProduto_tipo));

```



```

-----
-- Table `SupplyChain`.`Produto_Fisico`
-----
CREATE TABLE IF NOT EXISTS `SupplyChain`.`Produto_Fisico` (
  `idProduto_Fisico` INT NOT NULL,
  `Preco` FLOAT NULL,
  `Quantidade` INT NULL,
  `Dimensoes` FLOAT NULL,
  `Peso` FLOAT NULL,
  `Data_Validade` DATE NULL,
  `Composto` VARCHAR(75) NULL,
  `fk_idProdutoTipo` INT NOT NULL,
  PRIMARY KEY (`idProduto_Fisico`),
  INDEX `fk_idProdutoTipo_idx` (`fk_idProdutoTipo` ASC) VISIBLE,
  CONSTRAINT `fk_idProdutoTipo`
    FOREIGN KEY (`fk_idProdutoTipo`)
    REFERENCES `SupplyChain`.`Produto_Tipo` (`idProduto_Tipo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 4: Código SQL para Criação de Tabela Produto_Fisico

- Relação Componetes:

Domínio Produto origem:	inteiro	
Domínio Componentes:	inteiro	
Componente(
Produto_origem	Produto origem	NOT NULL,
Componentes	Componentes	NOT NULL,
CHAVE ESTRANGEIRA(Produto_Origem)		
CHAVE ESTANGEIRA(Componentes));		

```

-----
-- Table `SupplyChain`.`Componentes`
-----

CREATE TABLE IF NOT EXISTS `SupplyChain`.`Componentes` (
  `Produto_origem` INT NOT NULL,
  `Componentes` INT NOT NULL,
  INDEX `Produto_Origem_idx` (`Produto_origem` ASC) VISIBLE,
  INDEX `Componentes_idx` (`Componentes` ASC) VISIBLE,
  CONSTRAINT `Produto_Origem`
    FOREIGN KEY (`Produto_origem`)
    REFERENCES `SupplyChain`.`Produto_Fisico` (`idProduto_Fisico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `Componentes`
    FOREIGN KEY (`Componentes`)
    REFERENCES `SupplyChain`.`Produto_Fisico` (`idProduto_Fisico`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 5: Código SQL para Criação de Tabela Componentes

- Relação Elementos;

Domínio id Lote: inteiro

Domínio id Produto Físico: inteiro

Elementos(

fk_idLote id Lote NOT NULL,

fk_idProduto id Produto NOT NULL,

CHAVE ESTRANGEIRA(fk_idLote)

CHAVE ESTRANGEIRA(fk_idProduto));

```
-- Table `SupplyChain`.`Elementos`
```

```
CREATE TABLE IF NOT EXISTS `SupplyChain`.`Elementos` (
  `fk_idLote` INT NOT NULL,
  `fk_idProduto_Fisico` INT NOT NULL,
  INDEX `fk_idProduto_Fisico_idx` (`fk_idProduto_Fisico` ASC) VISIBLE,
  INDEX `fk_idLote_idx` (`fk_idLote` ASC) VISIBLE,
  CONSTRAINT `fk_idLote`
    FOREIGN KEY (`fk_idLote`)
      REFERENCES `SupplyChain`.`Lote` (`idLote`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_idProduto_Fisico`
    FOREIGN KEY (`fk_idProduto_Fisico`)
      REFERENCES `SupplyChain`.`Produto_Fisico` (`idProduto_Fisico`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

Figura 6: Código SQL para Criação de Tabela Elementos

- Relação Empresa:

Domínio id Empresa: inteiro

Domínio Nome: string de comprimento variável, comprimento 45

Domínio Contacto: valor entre 000.00-999.99

Domínio Pais: string de comprimento variável, comprimento 45

Domínio Rua: string de comprimento variável, comprimento 45

Domínio Código postal: string de comprimento variável, comprimento 45

Domínio Tipo: string de comprimento variável, comprimento 45

```

Empresa(
    idEmpresa          id Empresa          NOT NULL,
    Nome                Nome                NOT NULL,
    Contacto            Contacto            NOT NULL,
    Pais                País                NOT NULL,
    Rua                 Rua                 NOT NULL,
    Codigo_Postal       Código Postal       NOT NULL,
    Tipo                Tipo                NOT NULL,
    CHAVE PRIMARIA(idEmpresa));

```

```

-----
-- Table `SupplyChain`.`Empresa`
-----

```

```

CREATE TABLE IF NOT EXISTS `SupplyChain`.`Empresa` (
  `idEmpresa` INT NOT NULL,
  `Nome` VARCHAR(45) NULL,
  `Contacto` FLOAT NOT NULL,
  `Pais` VARCHAR(45) NULL,
  `Rua` VARCHAR(45) NOT NULL,
  `Codigo_Postal` VARCHAR(45) NOT NULL,
  `Tipo` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idEmpresa`))
ENGINE = InnoDB;

```

Figura 7: Código SQL para Criação de Tabela Empresa

- Relação Lote:

Domínio id Lote:	inteiro
Domínio Data:	temporal
Domínio id Empresa Origem:	inteiro
Domínio id Empresa Destino:	inteiro

```

Lote(
    idLote              id Lote              NOT NULL,
    Data                Data                NOT NULL,
    fk_idEmpresa_Origem id Empresa Origem    NOT NULL,
    fk_idEmpresa_Destino id Empresa Destino  NOT NULL,
    CHAVE PRIMARIA(idLote)
    CHAVE ESTRANGEIRA(fk_idEmpresa_Origem)
    CHAVE ESTRANGEIRA(fk_idEmpresa_Destino));

```

```

-----
-- Table `SupplyChain`.`Lote`
-----

CREATE TABLE IF NOT EXISTS `SupplyChain`.`Lote` (
  `idLote` INT NOT NULL,
  `Data` DATE NOT NULL,
  `fk_idEmpresa_Origem` INT NOT NULL,
  `fk_idEmpresa_Destino` INT NOT NULL,
  PRIMARY KEY (`idLote`),
  INDEX `fk_idEmpresa_Origem_idx` (`fk_idEmpresa_Origem` ASC) VISIBLE,
  INDEX `fk_idEmpresa_Destino_idx` (`fk_idEmpresa_Destino` ASC) VISIBLE,
  CONSTRAINT `fk_idEmpresa_Origem`
    FOREIGN KEY (`fk_idEmpresa_Origem`)
    REFERENCES `SupplyChain`.`Empresa` (`idEmpresa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_idEmpresa_Destino`
    FOREIGN KEY (`fk_idEmpresa_Destino`)
    REFERENCES `SupplyChain`.`Empresa` (`idEmpresa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Figura 8: Código SQL para Criação de Tabela Lote

5.3) Tradução das interrogações do utilizador para SQL (alguns exemplos)

Neste subcapítulo apresentamos alguns exemplos do código SQL que permitem obter resposta a perguntas do utilizador:

```
-- 1. Lista de todos os produtos
select Nome from produto_tipo order by Nome ASC;
```

Figura 9: Código SQL para consultar Lista de todos os produtos

```
-- 2. Informações de todos os Lote
select l.idLote, l.data, emp1.nome as EmpresaOrigem, emp2.nome as EmpresaDestino, e.fk_idProduto_Fisico as IdProdutoFisico
from lote l, elementos e, empresa emp1, empresa emp2
where l.idLote = e.fk_idLote
AND emp1.idEmpresa = l.fk_idEmpresa_Origem
AND emp2.idEmpresa = l.fk_idEmpresa_Destino;
```

Figura 10: Código SQL para consultar informações de todos os Lotes

```
-- 3. Lista de todas as empresas
select * from Empresa;
```

Figura 11: Código SQL para consultar Lista de todas as empresas

```
-- 4. Lista de todos os fornecedores
select * from Empresa where Tipo in ('Fornecedor');
```

Figura 12: Código SQL para consultar Lista de todos os fornecedores

```
-- 5. Lista de todos os armazenistas
select * from Empresa where Tipo in ('Armazenista');
```

Figura 13: Código SQL para consultar Lista de todos os armazenistas

```
-- 6. Lista de todas as produtoras
select * from Empresa where Tipo in ('Produtor');
```

Figura 14: Código SQL para consultar Lista de todas as produtoras

```
-- 7. Lista de todos os retalhos
select * from Empresa where Tipo in ('Retalho');
```

Figura 15: Código SQL para consultar Lista de todos os retalhos

```
-- 8. Lista da validade de produtos físicos
select pt.Nome, pf.idProduto_Fisico, pf.data_validade < CURRENT_TIMESTAMP as ForaValidade from produto_fisico pf, produto_tipo pt
where pf.fk_idProdutoTipo = pt.idProduto_Tipo AND pf.data_validade is not null order by pt.Nome ASC;
```

Figura 16: Código SQL para consultar Lista da validade de produtos físicos

```
-- 9. Componentes dos produtos
select c.produto_origem, pf.idProduto_Fisico as IdProdutoOrigem, pt.Nome as NomeProdutoOrigem,
c.componentes, pf1.idProduto_Fisico as IdProdutoComponentes, pt1.Nome as NomeProdutoComponentes
from produto_tipo pt, produto_fisico pf, componentes c, produto_tipo pt1, produto_fisico pf1
where c.Produto_origem = pf.idProduto_Fisico
AND pf.fk_idProdutoTipo = pt.idProduto_Tipo
AND c.componentes = pf1.idProduto_Fisico
AND pf1.fk_idProdutoTipo = pt1.idProduto_Tipo
order by c.Produto_origem;
```

Figura 17: Código SQL para consultar Componentes dos produtos

```
-- 10. Todos os Produtos Fisicos
select * from produto_fisico pf, produto_tipo pt where pf.fk_idProdutoTipo = pt.idProduto_Tipo order by pf.fk_idProdutoTipo;
```

Figura 18: Código SQL para consultar Todos os produtos fisicos

```
-- 11. Top Empresas com mais lotes recebidos
select e.idEmpresa, e.nome, e.pais from empresa e, lote l where e.idEmpresa = l.fk_idEmpresa_Destino order by e.nome asc;
```

Figura 19: Código SQL para consultar top empresas com mais lotes recebidos

```
-- 12. Calcular preco de um lote
DELIMITER //
create function precoLote (idLote int) returns float
BEGIN
declare price float;
select sum(p.preco) INTO price
from elementos e, produto_fisico p
where v.fk_idProdutoFisico = p.idProduto_Fisico
and v.fk_ifLote = idLote;
return price;
END//
DELIMITER ;
```

Figura 20: Código SQL para calcular preço de um lote

5.4) Escolha, definição e caracterização de índices em SQL

```
-- Criação de uma empresa
Create procedure adiciona_Empresa(IN idEmpresa int,
    IN nome VARCHAR(75),
    IN Contacto int,
    IN pais VARCHAR(75),
    IN Rua VARCHAR(75),
    IN codigo_Postal VARCHAR(75),
    IN tipo VARCHAR(75))

Begin
INSERT INTO Empresa(idEmpresa, Nome, Contacto, Pais, Rua, Codigo_Postal, Tipo)
    Values (idEmpresa, Nome, Contacto, Pais, Rua, Codigo_Postal, Tipo);
END //
```

Figura 21: Código SQL para registar empresa

```
-- Criação de um produto
Create procedure adiciona_produto(IN idProduto_Fisico int,
    IN Preco float,
    IN Quantidade int,
    IN Dimensoes float,
    IN Peso float,
    IN Data_validade Date,
    IN fk_idProdutoTipo INT,
    Composto VARCHAR(75))

Begin
INSERT INTO produto_fisico (idProduto_Fisico, Preco, Quantidade, Dimensoes, Peso, Data_validade, fk_idProdutoTipo, Composto)
    Values (idProduto_Fisico, Preco, Quantidade, Dimensoes, Peso, Data_validade, fk_idProdutoTipo, Composto);
END //
```

Figura 22: Código SQL para registo de um produto

```
-- Criacao de um lote
Create procedure adiciona_lote(IN idLote int, IN Data Date, IN fk_idEmpresa_Origem int, IN fk_idEmpresa_Destino INT)
Begin
INSERT INTO lote (idLote, Data, fk_idEmpresa_Origem, fk_idEmpresa_Destino)
    Values (idLote, Data, fk_idEmpresa_Origem, fk_idEmpresa_Destino);
END //
```

Figura 23: Código SQL para criação de um lote

```
-- Criação de um tipo de produto
Create procedure adiciona_lote(IN idProduto_Tipo int, IN Preco float, IN Peso_m2 float, IN Nome VARCHAR(75))
Begin
INSERT INTO produto_tipo (idProduto_Tipo, Preco, Peso_m2, Nome)
    Values (idProduto_Tipo, Preco, Peso_m2, Nome);
END //
```

Figura 24: Código SQL para criação de um tipo de produto

5.5) Escolha, definição e caracterização de índices em SQL

```
1 • use supplychain;
2
3 • create index idx_data_validade on produto_fisico (data_validade);
4
5 • create index idx_data_lote on lote (data);
6
7
```

Figura 25: Código SQL para a criação dos Índices sobre as colunas Data_Validade e Data

5.6) Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Na tabela seguinte encontra-se o cálculo do espaço que ocupará a base de dados em disco. Assumimos que os atributos VARCHAR utilizam o tamanho máximo declarado.

Tabela 4: Tamanho de cada entrada na respetiva Tabela

Relação	Atributo	Data Type	Tamanho(B)	Total(B)
Produto_Tipo	idProduto_tipo	INT	4	65
	Preco	FLOAT	8	
	Preso_m2	FLOAT	8	
	Nome	VARCHAR(45)	45	
Produto_Fisico	idProduto	INT	4	114
	Preco	FLOAT	8	
	Quantidade	INT	4	
	Dimensoes	FLOAT	8	
	Peso	FLOAT	8	

	Data_Validade	DATE	3	
	Composto	VARCHAR(75)	75	
	fk_idProduto	INT	4	
Lote	idLote	INT	4	15
	Data	DATE	3	
	fk_idEmpresa_Origem	INT	4	
	fk_idEmpresa_Destino	INT	4	
Componentes	Produto_origem	INT	4	8
	Componentes	INT	4	
Elemento	fk_idLote	INT	4	8
	fk_idProduto_Fisico	INT	4	
Empresa	idEmpresa	INT	4	237
	Nome	VARCHAR(45)	45	
	Contacto	FLOAT	8	
	Pais	VARCHAR(45)	45	
	Rua	VARCHAR(45)	45	
	Codigo_Postal	VARCHAR(45)	45	

Consideremos que vamos ter temos a utilizar a nossa aplicação nas primeiras 10 empresas registadas ($237 \times 10 = 2370$). O conjunto de todas as empresas realizou 80 lotes ($80 \times 15 = 1200$). São registados 700 produtos tipo e 700 Produtos Físico ($700 \times 114 = 79800 + 700 \times 15 = 10500$), 100 ($100 \times 8 = 800$) são Componentes e 80 ($80 \times 8 = 160$) são elementos.

Ou seja ao todo prevemos $2370 + 1200 + 7980 + 10500 + 800 + 160 = 23010$ Bytes/ano.

5.7) Definição e caracterização das vistas de utilização em SQL (alguns exemplos)

Uma view é o resultado de uma ou mais operações sobre a base de dados. É uma tabela dinâmica resultante de várias operações entre tabelas e outras views.

```
-- Lista de todos os tipos de produtos
create view ListaProdutosTipos as
select Nome from produto_tipo order by Nome ASC;
```

Figura 26: Código SQL para lista de todos os tipos de produtos

```
-- Lista de todos os produtos
create view ListaProdutos as
select * from produto_fisico pf, produto_tipo pt where pf.fk_idProdutoTipo = pt.idProduto_Tipo order by pf.fk_idProdutoTipo;
```

Figura 27: Código SQL para lista de todos os produtos

```
-- Lista de todas as empresas
create view ListaEmpresas as
select * from Empresa;
```

Figura 28: Código SQL para lista de todas as empresas

```
-- Lista de todos os lotes
create view ListaLotes as
select l.idLote, l.data, emp1.nome as EmpresaOrigem, emp2.nome as EmpresaDestino, e.fk_idProduto_Fisico as IdProdutoFisico
from lote l, elementos e, empresa emp1, empresa emp2
where l.idLote = e.fk_idLote
AND emp1.idEmpresa = l.fk_idEmpresa_Origem
AND emp2.idEmpresa = l.fk_idEmpresa_Destino;
```

Figura 29: Código SQL para lista de todos os lotes

5.8) Revisão do sistema implementado

Concluída a última fase da construção do sistema como pretendíamos, reunimos com o corpo administrativo da empresa, onde testamos as várias funcionalidades desenvolvidas e explicando seu propósito exemplificando-os.

O produto final foi aceite pelo corpo administrativo, com satisfação das partes destes, sendo que este cumprimos com os requisitos pretendidos.

6) Conclusões e Trabalho Futuro

O desenvolvimento do sistema de base de dados que apresentamos foi faseado de forma a contribuir para uma maior estruturação e robustez da solução apresentada.

A existência de uma metodologia de trabalho revelou-se essencial para a diminuição da ocorrência de erros no processo de desenvolvimento, simplificando o processo de criação do sistema. Tentou-se realizar todas as cinco etapas deste trabalho da forma mais simples mas eficaz possível, de forma a que conseguíssemos responder da melhor maneira a todos os objetivos propostos nesta primeira fase.

No decorrer do trabalho, deparamo-nos com certas dificuldades, como escolher um diagrama ER mais adequado ao problema, aprofundar o estudo do comportamento da empresa e na criação de queries que achamos adequadas.

Pela análise do trabalho que desenvolvemos concluímos que as fases de definição do problema e desenvolvimento do modelo conceptual são das mais importantes e aquelas que mais nos permitiram avançar no projeto.

No futuro temos de melhorar o nosso povoamento, de maneira a ter um trabalho mais próximo da realidade.

Em suma, concluímos que os objetivos foram cumpridos. Sendo o resultado deste processo uma base de dados funcional e capaz de cumprir o propósito que levou à sua implementação.

Lista de Siglas e Acrónimos

BD	- Base de Dados
DW	- Data Warehouse
OLTP	- <i>On-Line Analytical Processing</i>
SGBD	- <i>Sistema de Gestão de Base de Dados</i>
SQL	- <i>Structured Query Language</i>

Anexos

<<Os anexos deverão ser utilizados para a inclusão de informação adicional necessária para uma melhor compreensão do relatório o para complementar tópicos, secções ou assuntos abordados. Os anexos criados deverão ser numerados e possuir uma designação. Estes dados permitirão complementar o Índice geral do relatório relativamente à enumeração e apresentação dos diversos anexos.>>

I. Anexo 1