

Relatório Final



Instituto Politécnico de Coimbra

Instituto Superior de Engenharia de Coimbra

Departamento de Engenharia Informática e de Sistemas

Engenharia Informática

Bases de Dados

2021/2022

Cap 1 - Tema

Base de Dados - Armazém

Cap 2 - Identificação dos Autores

Autores:

Nº Aluno	Prática	Nome	Email
2016012191	P5	João Carlos Carvalho Rodrigues	a21260151@isec.pt
2014013189	P8	Rafael Filipe Martins Alves	a21240485@isec.pt
2019149526	P5	Afonso da Costa Brigas	a2019149526@isec.pt

Distribuição do esforço

Tarefa	Rafael	Alves	João	Rodrigues	Afonso	Brigas
Análise da situação atual	33%		33%		33%	
Meta 2	33%		33%		33%	
Meta 3	33%		33%		33%	
Meta 4	33%		33%		33%	

Cap 3 - Introdução

- O objetivo deste projeto é implementar um sistema que permita efetuar, de forma automatizada, a gestão e controlo de um Armazém.
- O armazém "BD21" foi fundado por Rafael Alves, Afonso Brigas e João Rodrigues, com o objetivo de, numa forma automatizada, não só criar condições de trabalho agradáveis mas focar no ponto principal que é a organização e armazenamento dos dados de todos os produtos que entram e saem do armazém.
- Esta base de dados permite-nos ter total controlo do produto em stock e sua secção pertencente. Dados relativos aos funcionários, secção e entrada de produtos no armazém pela qual são responsáveis. Permite-nos ainda guardar as informações relativas aos fornecedores de cada uma das entradas de produto e os dados relativos aos clientes e efetuadas pelos próprios.
- Cada entrada e venda de produto segue apenas um produto em específico.
- De uma forma episódica vamos tentar automatizar ao máximo o funcionamento do armazém.

No capítulo 4, faz-se um enquadramento da relevância da proposta do projeto, descreve-se a situação actual, os principais problemas encontrados e, por fim, descreve-se a proposta que irá permitir solucionar esses problemas e atingir o objectivo desejado. No capítulo 5, são descritas as funcionalidades da aplicação que irá interagir com a base de dados, apresentando-se os seus módulos e detalhando as suas restrições de funcionamento. No capítulo 6, apresenta-se a análise de dados necessária para a solução proposta, descrevendo-se todas as entidades de dados e os relacionamentos existentes entre elas. No capítulo 7, apresenta-se o modelo físico (ou diagrama de tabelas) da base de dados que vai servir de suporte ao sistema de informação proposto, assim como o respectivo *script* de criação da base de dados. No capítulo 8, são apresentadas consultas (código SQL) das principais funcionalidades de pesquisas especificadas na solução proposta para a livraria. Deve ser indicado para cada uma das consultas o módulo a que está associada e o seu objectivo. Por fim, no capítulo 8 são apresentadas conclusões do trabalho desenvolvido.

Cap 4 - Enquadramento da Proposta

Neste capítulo faz-se um enquadramento da proposta de trabalho no negócio de venda de livros da livraria VENDELIVROS. Na secção 4.1 descreve-se o funcionamento actual do negócio de venda de livros da livraria, apresentando-se na secção 4.2 o conjunto de problemas existentes actualmente na livraria e que se pretende solucionar. A secção 4.3 descreve a proposta de solução que permitirá solucionar os problemas existentes e satisfazer as necessidades encontradas.

Sec 4.1 - Diagnóstico da Situação Atual

- Actualmente a organização do armazém é controlada com registos em papel guardados em dossiers organizados alfabeticamente e com recurso do Software Excel não tendo uma gestão de dados muito eficiente.

Sempre que existem entradas ou saídas de stock, estas são registadas manualmente por um trabalhador.

- Devido a uma grande afluência de trabalho e movimento por vezes são observadas incongruências na documentação consequente do erro humano. Tal problema é inaceitável num negócio desta escala e pode causar grandes perdas.

Sec 4.2 - Problemas encontrados

- Um dos problemas encontrados, dada uma situação de um armazém com alguma dimensão, é a difícil gestão de todos os diferentes produtos, bem como as áreas do armazém onde se encontra cada produto, e o stock dos mesmos. Com isto achamos essencial dividir o armazém em secções para os diferentes tipos de produtos.
- Outro problema foi que devido ao elevado número de produtos diferentes foi difícil agrupar dentro de uma mesma entrada ou venda vários tipos de produtos diferentes.

Sec 4.3 - Descrição da Solução Proposta

- A solução a implementar deve ser baseada numa arquitetura cliente/servidor constituída por uns protótipos que representam a parte cliente e que se destina ao responsável do armazém ou ao gerente do armazém (com mais restrições de acessos em relação do responsável do armazém) e a parte servidor diz respeito à base de dados a implementar para guardar todos os dados que vão existir no armazém.
- A base de dados a criar deve ser capaz de listar diferentes secções, produtos pertencentes a essas secções, vendas pedidas por clientes e entrada de produtos geridas por funcionários e fornecidas por fornecedores.
- Para trabalhar sobre a base de dados implementada, devem ser criadas aplicações capazes de listar, adicionar, editar e remover dados das tabelas (do ponto de vista do gestor).
- Devido à grande abundância de produtos de várias áreas consideramos necessário criar vários setores dentro do armazém que apresente por categoria os vários tipos de produtos (construção, decoração de interiores, decoração exteriores, textil, etc).
- Cada operação de entrada ou venda de produtos do armazém, é realizada de forma separada para cada produto diferente.

Cada entrada ou venda pode apenas conter um produto do mesmo tipo.

Cap 5 - Funcionalidades da Aplicação

Sec 5.1 - Estrutura da Aplicação

- Gestão de vendas
- Gestão de clientes
- Gestão de produtos
- Gestão de secções
- Gestão de funcionários
- Gestão de entrada
- Gestão de fornecedores

Cap 6 - Análise de Dados

Neste capítulo pretende-se descrever detalhadamente algumas das entidades envolvidas na solução proposta. Na subsecção 6.1 são descritas as entidades (respectivos atributos e restrições), na secção 6.2 são descritas todos os relacionamentos existentes entre as diversas entidades, apresentando-se por fim, na secção 6.3, o modelo de Entidade / Relacionamento completo, incluindo todas as entidades e relações relevantes.

Sec 6.1 - Entidades

Nesta secção vão ser descritas algumas das Entidades relevantes para o negócio de venda de produtos no armazém. Após uma análise aprofundada do modelo de negócio, constatou-se a necessidade das seguintes Entidades:

- Vendas
- Produto
- Cliente
- Secções
- Fornecedores

Sec 6.1.1 - Entidade Vendas

A entidade vendas representa a informação relativa as vendas que existem no armazem. Esta entidade representa todas as vendas que ocorreram no armazem. É inserido um novo registo nesta entidade sempre que o armazem vende um produto.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
codigo_venda	Numérico (6 dígitos)	Código interno único atributo a cada venda. É um número de 6 dígitos gerado sequencialmente. Ex. 92233002
Preço unitário	Numérico (6 dígitos, com duas casas decimais)	Preço de cada produto individualmente. Ex. 18.99 €
data_venda	Data	Dia/mês/ano Ex. 06/11/2001
Quantidade_vendida	Numérico (4 dígitos)	Número de produtos. Ex. 4

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações

codigo_venda	N	S	Identificador (chave primária), não admite nulos. Não existem duas vendas com o mesmo código.
Preço_unitário	N	N	Não admite nulos, mas podem existir produtos com o mesmo preço.
data_venda	N	N	Não admite nulos. Podem existir duas vendas com a mesma data.
Quantidade_vendida	N	N	Não admite valor nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Tem	N : 1	cliente	Cliente
Inclui	N : 1	Produto	Produto

Sec 6.1.2 - Entidade Produtos

A entidade produto representa a informação relativa aos produtos pertencentes ao armazém tais como nomes dos produtos, quantidades em stock e preços.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
codigo_produto	Numérico (6 dígitos)	Código interno único atribuído a cada produto. Constituído por 6 dígitos e gerado sequencialmente. Ex. 922334

quant_stock	Numérico (3 dígitos)	Número existente em stock de determinado produto. Ex. 250
preço_tabela	Numérico (8 dígitos, com 2 casas decimais)	Preço de determinado produto. Ex. 25,45€
nome	30 Caracteres	Nome de determinado produto. Ex. Teclado X

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
codigo_produto	N	S	Identificador (chave primária), não admite nulos. Não existem dois produtos com o mesmo código.
quant_stock	N	N	Não admite valores nulos.
preço_tabela	N	N	Não admite valores nulos.
nome	N	S	Não admite valores nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
foi incluído em	1 : N	Vendas	Produtos
foi contido em	1 : N	Entrada	Entrada

pertencem a	N : 1	Secções	Produtos
-------------	-------	---------	----------

Sec 6.1.3 - Entidade Cliente

A entidade cliente representa a informação relativa aos clientes que realizam compras no armazem. É inserido um novo registo nesta entidade sempre que um cliente realiza uma compra no armazem.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
codigo_cliente	Numérico (6 dígitos)	Código interno único atributo a cada cliente. É um número de 6 dígitos gerado sequencialmente. Ex. 92233002
morada	50 caracteres	morada relativa a cada cliente. Ex. Rua Batalhoz, nº95 r/c, Cartaxo
Telefone	Numérico (9 dígitos)	Numero de telefone referente ao cliente. Ex. 966842008
Nome Cliente	50 carateres	Nome Cliente Ex. João Carlos

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
codigo_cliente	N	S	Identificador (chave primária), não admite nulos. Não existem dois cliente com o mesmo código.
morada	N	N	Não admite nulos, mas podem existir moradas iguais.
Telefone	N	S	Não admite nulos. Não podem existir dois numeros de telefone iguais.
Nome Cliente	N	N	Não admite valor nulos, mas podem existir nomes iguais.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
Efetua	1 : N	vendas	Cliente

Sec 6.1.4 - Entidade Fornecedores

A entidade cliente representa a informação relativa aos clientes que realizam compras no armazem. É inserido um novo registo nesta entidade sempre que um cliente realiza uma compra no armazem.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
------------------	---------------	-----------

codigo_fornecedor	Numérico (6 dígitos)	Código interno único atributo a cada fornecedor. É um número de 6 dígitos gerado sequencialmente. Ex. 92233002
nome_fornecedor	50 caracteres	Nome relativa a cada cliente. Ex. Nestle
Telefone	Numérico (9 dígitos)	Numero de telefone referente ao fornecedor. Ex. 243716444
Contribuinte	Numérico (9 dígitos)	Numero de Contribuinte. Ex. 244533243

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
codigo_fornecedor	N	S	Identificador (chave primária), não admite nulos. Não existem dois Fornecedores com o mesmo código.
nome_fornecedor	N	S	Não admite nulos, não mas podem existir nomes iguais.
Telefone	N	S	Não admite nulos. Não podem existir dois numeros de telefone iguais.
Contribuinte	N	S	Não admite valor nulos, mas não podem existir nomes iguais.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
fornece	1 : N	Entradas	Fornecedor

Sec 6.1.5 - Entidade Secções

A entidade secções representa a informação relativa às diferentes secções existentes no armazém.

Atributos relevantes da Entidade:

Nome do atributo	Tipo de Dados	Descrição
codigo_secçao	Numérico (6 dígitos)	Código interno único atribuído a cada secção. Constituído por 6 dígitos e gerado sequencialmente. Ex. 000256
secçao	10 Carateres	Nome de determinada secção. Ex. Informática

Restrições dos atributos da Entidade:

Nome do atributo	Aceita Nulos?	Valores Únicos?	Observações
codigo_secçao	N	S	Identificador (chave primária), não admite nulos. Não existem duas secções com o mesmo código.
secçao	N	S	Não admite valores nulos.

Relacionamentos da Entidade:

Nome do relacionamento	Cardinalidade	Entidade Relacionada	Participação Obrigatória
incluem os	1 : N	Produto	Produto
é gerida por	N : N	Funcionario	Secção

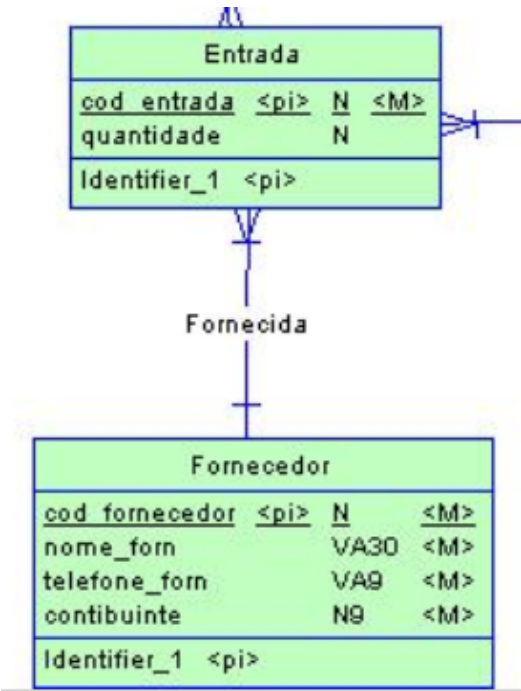
Sec 6.2 - Relacionamentos

Nesta secção são descritos todos os relacionamentos existentes entre as várias entidades. Após uma análise aprofundada do negócio de venda de produtos num armazem, constatou-se a necessidade dos seguintes relacionamentos:

- fornecida
- inclui
- gerida
- controlada

Sec 6.2.1 - Relacionamento: fornecido

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Fornecedores e entradas, ou seja, relativamente ao reabastecimento de stock do armazem. O objectivo é expressar quais os fornecedores que fornecem os produtos e vice-versa.



Após uma análise do funcionamento da armazem, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

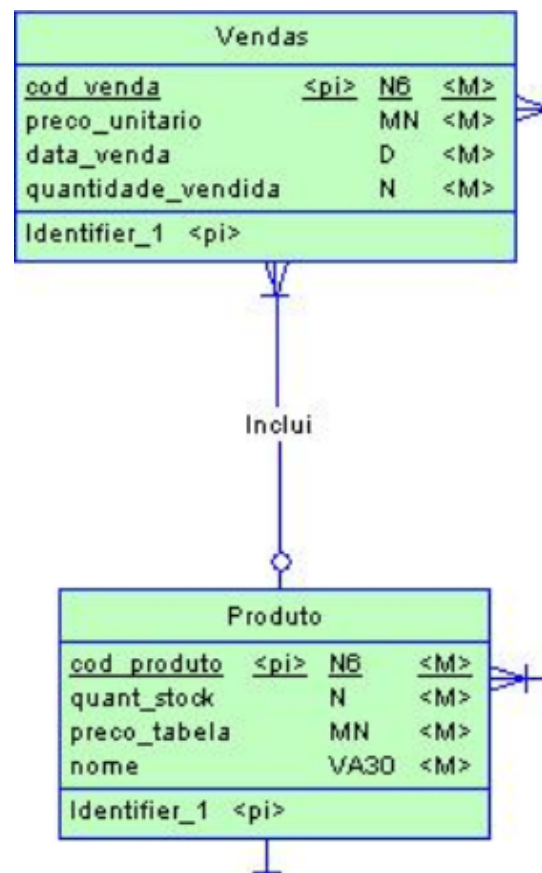
- Um produto pode ser fornecido por mais que um fornecedor,
- Um fornecedor pode fornecer vários produtos,
- Um produto necessariamente foi entregue por um fornecedor, caso contrário não é registado,
- Não é requerido que existência prévia no armazem, de um produto de um fornecedor, antes da sua inserção na base de dados.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
Entrada	NÃO	1 : N	SIM	Fornecedor
Observações				
<ul style="list-style-type: none">• um produto registado no armazem, obrigatoriamente foi entregue por um fornecedor,• um fornecedor pode ter entregue vários produtos que estão registados no armazem.				

Sec 6.2.2 - Relacionamento: inclui

Este relacionamento pretende expressar o relacionamento existente entre as Entidades vendas e produtos, de forma a controlar a saída de produtos do armazem. O objectivo é expressar quais os clientes que realizaram compras no armazem e os produtos que os mesmo compraram.



Após uma análise do funcionamento da armazem, e as orientações definidas pelo cliente, definiu-se as seguintes condições:

- Um produto pode estar incluído numa única venda,
- Uma venda pode incluir vários produtos,
- Uma venda necessariamente foi efetuada, caso contrário não é registada.

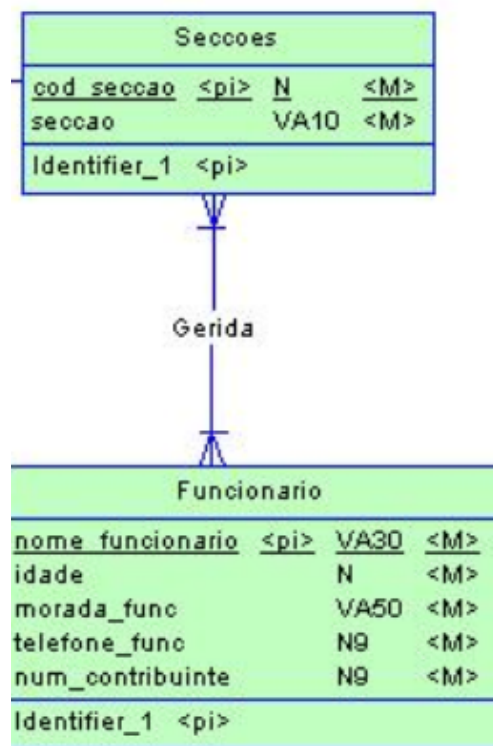
Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
vendas	NÃO	N : 1	SIM	produto
Observações				
<ul style="list-style-type: none">• uma venda registado no armazem, obrigatoriamente foi efetuada,				

Sec 6.2.3 - Relacionamento: gerida

Este relacionamento pretende expressar o relacionamento existente entre as Entidades secções e funcionarios, para que haja uma melhor gerência das secções do armazem. O objectivo é expressar quais os funcionários que gerem secções específicas do estabelecimento.

Após uma análise do funcionamento da armazem, e as orientações definidas pelo cliente, definiu-se as seguintes condições:



- Um produto pode estar incluído numa única venda,
- Uma venda pode incluir vários produtos,
- Uma venda necessariamente foi efetuada, caso contrário não é registada.

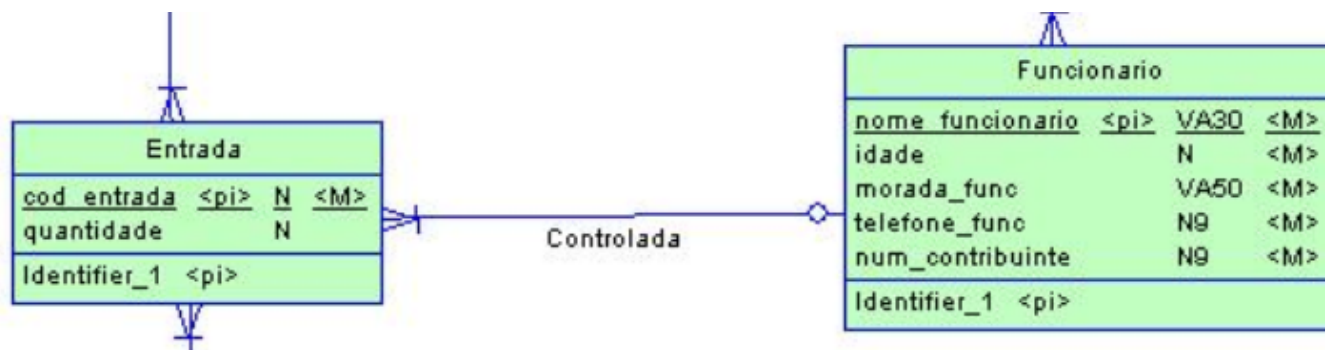
Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
secções	SIM	N : N	SIM	funcionário
Observações				
<ul style="list-style-type: none"> • um funcionário tem obrigatoriamente que gerir pelo menos uma secção, • uma secção tem de ser gerida por pelo menos um funcionário, 				

Sec 6.2.4 -Relacionamento: controlada

Este relacionamento pretende expressar o relacionamento existente entre as Entidades Funcionário e entradas, para que haja uma melhor gerência das entradas de produtos noo armazem. O objectivo é expressar quais os produtos que entram no armazem, cujo controlo e realizado por um funcionário.

Após uma análise do funcionamento da armazem, e as orientações definidas pelo cliente, definiu-se as seguintes condições:



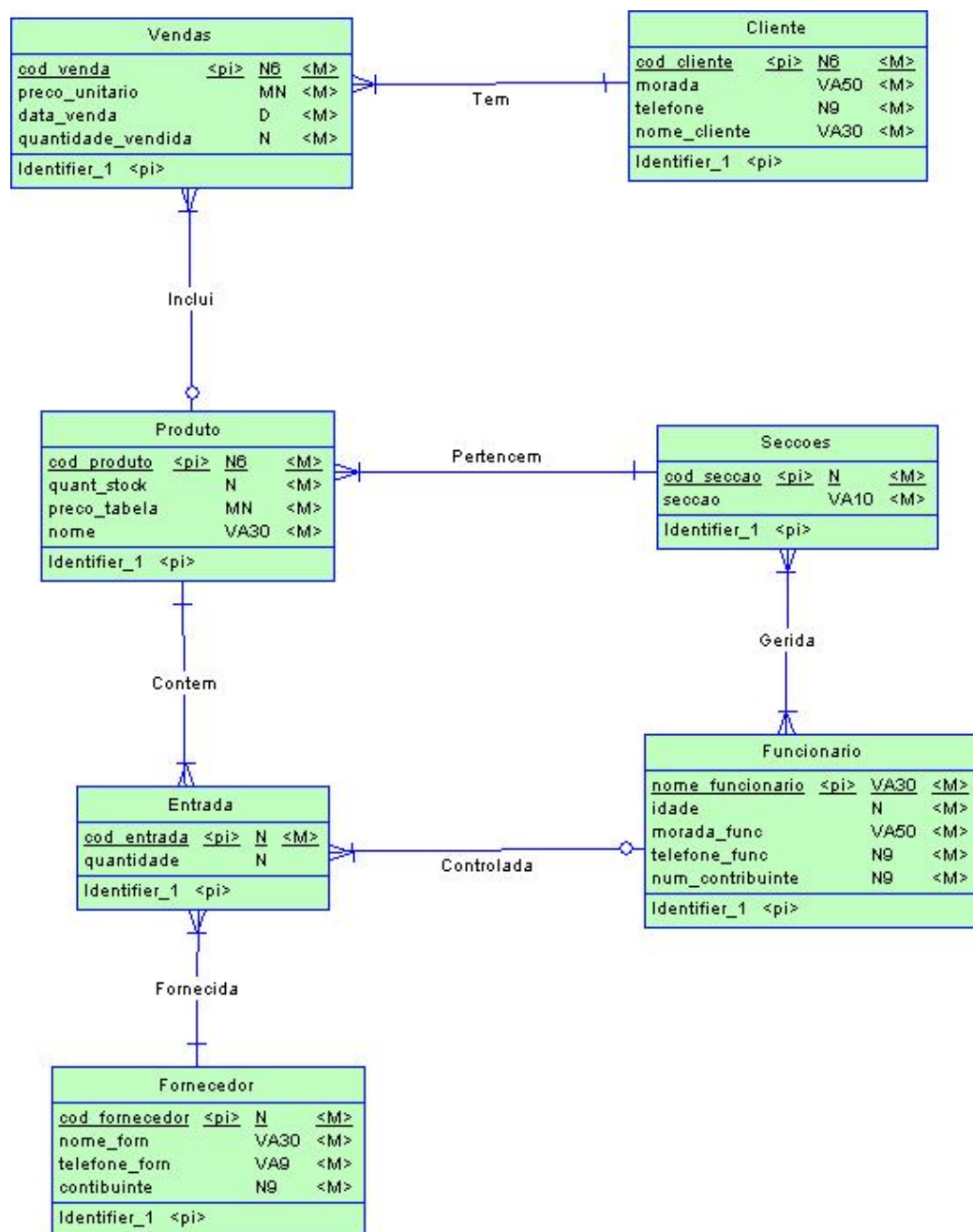
- Uma entrada precisa apenas de um funcionario,
- Um funcionário pode controlar mais do que uma entrada,
- Uma entrada necessariamente foi controlada por um funcionario, caso contrário não é registada.

Tomando estas condições em consideração, definiram-se as seguintes características

Entidade	Obrigatório	Cardinalidade	Obrigatório	Entidade
funcionário	SIM	1 : N	NÃO	entradas
Observações				
<ul style="list-style-type: none"> • uma entrada tem de ser obrigatoriamente controlada por um funcionario, 				

Sec 6.3 - Diagrama do Modelo Conceptual

O modelo de Entidade Relacionamento completo é o seguinte.



Cap 7 - Modelo Físico

Neste capítulo documenta-se detalhadamente todas as tabelas geradas para o Sistema de Gestão de Bases de Dados Relacional (SGBDR) Oracle, a partir do modelo Relacional definido no capítulo 6. Na subsecção 7.1, é descrita cada uma das tabelas (respectivos atributos e restrições de integridade), apresentando-se na secção 7.2 o modelo Físico (ou modelo de tabelas) completo, composto por todas as tabelas e restrições de integridade referencial. A secção 7.3 apresenta o código SQL necessário para criar as tabelas e as restrições definidas.

Sec 7.1 - Tabelas

Nesta secção são descritas as tabelas necessárias para implementar o negócio de venda de livros da livraria. Estas tabelas foram extraídas, tomando em consideração o modelo Entidade / Relacionamento definido no capítulo 6. As tabelas são as seguintes:

- Vendas
- Cliente
- Produto
- Secções
- Entrada
- Funcionário
- Fornecedor

Sec 7.1.1 - Tabela Vendas

A tabela vendas vai conter a informação das vendas efetuadas.

Restrições dos atributos da tabela: Vendas

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_venda	NUMBER(6)	pk_cod_venda	PRIMARY KEY
cod_produto	NUMBER(6)	fk_cod_produto	FOREIGN KEY REFERENCES Produto (cod_produto)
cod_cliente	NUMBER(6)	fk_cod_cliente	FOREIGN KEY REFERENCES Cliente (cod_cliente)
preco_unitario	NUMBER(8,2)	preco_unitario	UNIQUE
data_venda	DATE	data_venda	UNIQUE

quantidade_vendida	NUMBER	quantidade_vendida	UNIQUE
--------------------	--------	--------------------	--------

Sec 7.1.2 - Tabela Clientes

A tabela clientes vai conter a informação dos clientes que efetuaram compras.

Restrições dos atributos da tabela: Clientes

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_cliente	NUMBER(6)	pk_cod_cliente	PRIMARY KEY
morada	VARCHAR2(50)	morada	UNIQUE
telefone	NUMBER(9)	telefone	UNIQUE
nome_cliente	VARCHAR2(30)	nome_cliente	UNIQUE

Sec 7.1.3 - Tabela Produto

A tabela produto vai conter a informação relativa aos produtos existentes no armazém.

Restrições dos atributos da tabela: Produto

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_produto	NUMBER(6)	pk_cod_produto	PRIMARY KEY
cod_seccao	NUMBER	fk_cod_seccao	FOREIGN KEY REFERENCES Seccoes (cod_seccao)
quant_stock	NUMBER	quant_stock	UNIQUE
preco_tabela	NUMBER(8,2)	preco_tabela	UNIQUE
nome	VARCHAR2(30)	nome	UNIQUE

Sec 7.1.4 - Tabela Secções

A tabela Seccoes vai conter a informação relativa às diferentes secções presentes no armazém.

Restrições dos atributos da tabela: Seccoes

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_seccao	NUMBER	pk_cod_produto	PRIMARY KEY
seccao	VARCHAR2(10)	seccao	UNIQUE

Sec 7.1.5 - Tabela Entrada

A tabela Entrada vai conter a informação relativa à entrada de produtos no armazém.

Restrições dos atributos da tabela: Entrada

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_entrada	NUMBER	pk_cod_entrada	PRIMARY KEY
nome_funcionario	VARCHAR2(30)	fk_nome_funcionario	FOREIGN KEY REFERENCES Funcionario (nome_funcionario)
cod_produto	NUMBER(6)	fk_cod_produto	FOREIGN KEY REFERENCES Produto (cod_produto)
cod_fornecedor	NUMBER	fk_cod_fornecedor	FOREIGN KEY REFERENCES Fornecedor (cod_fornecedor)
quantidade	NUMBER	quantidade	UNIQUE

Sec 7.1.6 - Tabela Funcionário

A tabela Funcionario vai conter a informação relativa aos funcionários empregados no armazém.

Restrições dos atributos da tabela: Funcionario

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
nome_funcionario	VARCHAR2(30)	pk_nome_funcionario	PRIMARY KEY
idade	NUMBER	idade	UNIQUE
morada_func	VARCHAR2(50)	morada_func	UNIQUE
telefone_func	NUMBER(9)	telefone_func	UNIQUE
num_contribuinte	NUMBER(9)	num_contribuinte	UNIQUE

Sec 7.1.7 - Tabela Fornecedor

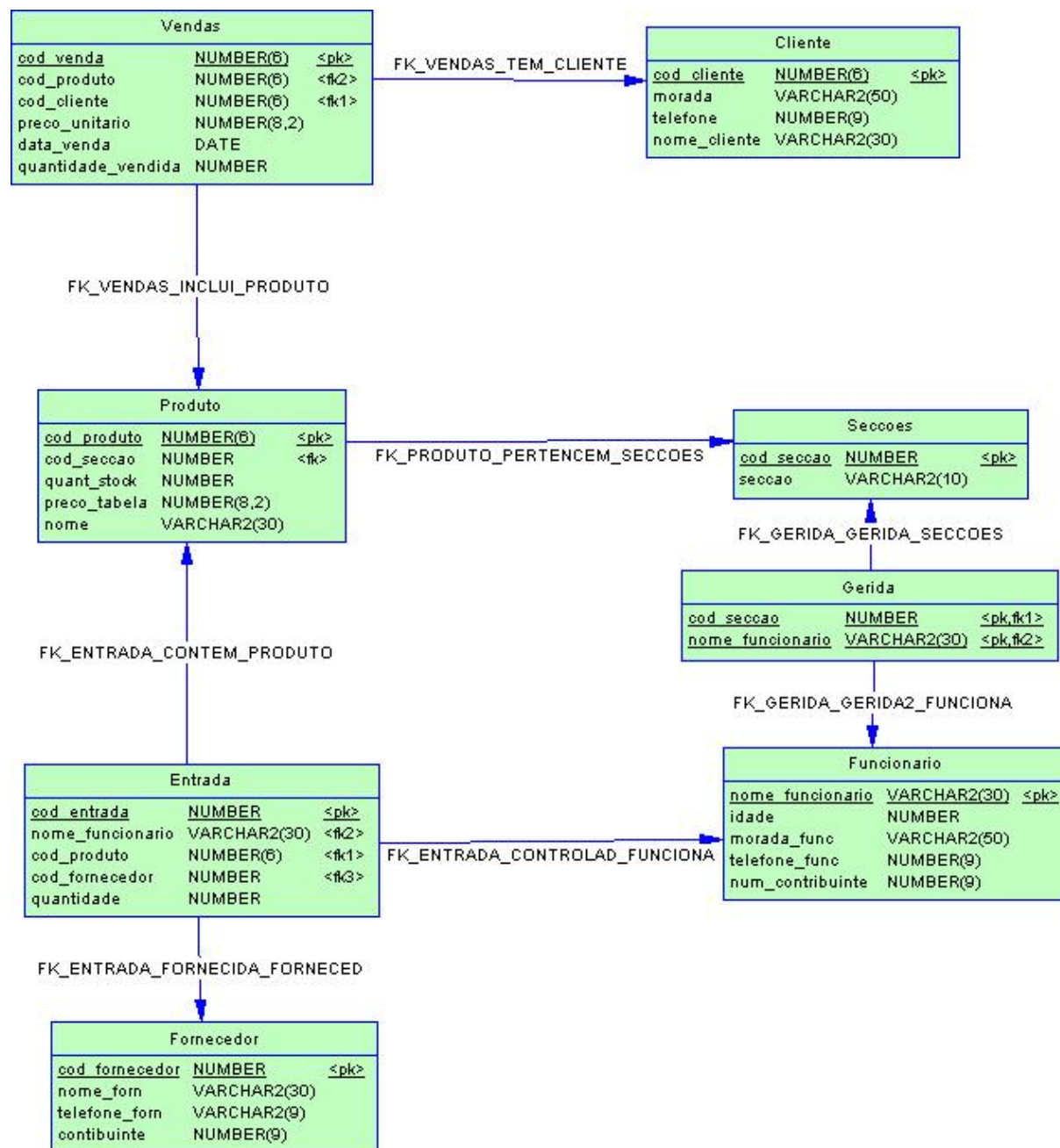
A tabela Fornecedor vai conter a informação relativa aos fornecedores de produtos do armazém.

Restrições dos atributos da tabela: Fornecedor

Atributo(s)	Tipo de Dados	Nome da Restrição	Restrições de Integridade Referencial
cod_fornecedor	NUMBER	pk_cod_fornecedor	PRIMARY KEY
nome_forn	VARCHAR2(30)	nome_forn	UNIQUE
telefone_forn	NUMBER(6)	telefone_forn	UNIQUE
contribuinte	NUMBER	contribuinte	UNIQUE

Sec 7.2 - Diagrama do Modelo Físico

O modelo Físico (ou diagrama de tabelas) completo com todas as tabelas é o seguinte.



Sec 7.3 - *Script* de criação da Base de Dados

Nesta secção apresentam-se as instruções SQL necessárias para criar as tabelas descritas anteriormente na secção 7.1 no SGBDR Oracle. O código SQL apresentado permite criar as tabelas, as restrições de integridade (chave primária, integridade referencial) suportadas pelo SGBD, assim como as validações de dados definidas e valores por omissão.

```
/*=====*/  
/* DBMS name:   ORACLE Version 10g           */  
/* Created on:   1/16/2022 6:17:01 PM         */  
/*=====*/
```

```
alter table ENTRADA
```

```
drop constraint FK_ENTRADA_CONTEM_PRODUTO;
```

```
alter table ENTRADA
```

```
drop constraint FK_ENTRADA_CONTROLAD_FUNCIONA;
```

```
alter table ENTRADA
```

```
drop constraint FK_ENTRADA_FORNECIDA_FORNECED;
```



```
alter table GERIDA
```

```
drop constraint FK_GERIDA_GERIDA_SECCOES;
```

```
alter table GERIDA
```

```
drop constraint FK_GERIDA_GERIDA2_FUNCIONA;
```

```
alter table PRODUTO
```

```
drop constraint FK_PRODUTO_PERTENCEM_SECCOES;
```

```
alter table VENDAS
```

```
drop constraint FK_VENDAS_INCLUI_PRODUTO;
```

```
alter table VENDAS
```

```
drop constraint FK_VENDAS_TEM_CLIENTE;
```

```
drop index CONTEM_FK;
```

```
drop index CONTROLADA_FK;
```

```
drop index FORNECIDA_FK;
```

```
drop index GERIDA2_FK;
```

```
drop index GERIDA_FK;
```

```
drop index PERTENCEM_FK;
```

```
drop index INCLUI_FK;
```

```
drop index TEM_FK;
```

```
drop table CLIENTE cascade constraints;
```

```
drop table ENTRADA cascade constraints;
```

```
drop table FORNECEDOR cascade constraints;
```

```
drop table FUNCIONARIO cascade constraints;
```

```
drop table GERIDA cascade constraints;
```

```
drop table PRODUTO cascade constraints;
```

```
drop table SECCOES cascade constraints;
```

```
drop table VENDAS cascade constraints;
```

```
/*=====*/
```

```
/* Table: CLIENTE */
```

```
/*=====*/
```

```
create table CLIENTE (
```

```
    COD_CLIENTE    NUMBER(6)          not null,
```

```
    MORADA         VARCHAR2(50)       not null,
```

```
    TELEFONE       NUMBER(9)          not null,
```

```
    NOME_CLIENTE   VARCHAR2(30)       not null,
```

```
    constraint PK_CLIENTE primary key (COD_CLIENTE)
```

```
);
```

```
/*=====*/
```

```
/* Table: ENTRADA */
```

```
/*=====*/
```

```
create table ENTRADA (
```

```
    COD_ENTRADA    NUMBER              not null,
```

```
    NOME_FUNCIONARIO VARCHAR2(30),
```

```
    COD_PRODUTO    NUMBER(6)          not null,
```

```
    COD_FORNECEDOR NUMBER              not null,
```

```
    QUANTIDADE      NUMBER,  
    constraint PK_ENTRADA primary key (COD_ENTRADA)  
);
```

```
/*=====*/
```

```
/* Index: CONTEM_FK                */
```

```
/*=====*/
```

```
create index CONTEM_FK on ENTRADA (  
    COD_PRODUTO ASC  
);
```

```
/*=====*/
```

```
/* Index: CONTROLADA_FK            */
```

```
/*=====*/
```

```
create index CONTROLADA_FK on ENTRADA (  
    NOME_FUNCIONARIO ASC  
);
```

```
/*=====*/
```

```
/* Index: FORNECIDA_FK             */
```

```
/*=====*/
```

```
create index FORNECIDA_FK on ENTRADA (  

```

COD_FORNECEDOR ASC

);

/*=====*/

/* Table: FORNECEDOR */

/*=====*/

create table FORNECEDOR (

COD_FORNECEDOR NUMBER not null,

NOME_FORN VARCHAR2(30) not null,

TELEFONE_FORN VARCHAR2(9) not null,

CONTIBUINTE NUMBER(9) not null,

constraint PK_FORNECEDOR primary key (COD_FORNECEDOR)

);

/*=====*/

/* Table: FUNCIONARIO */

/*=====*/

create table FUNCIONARIO (

NOME_FUNCIONARIO VARCHAR2(30) not null,

IDADE NUMBER not null,

MORADA_FUNC VARCHAR2(50) not null,

TELEFONE_FUNC NUMBER(9) not null,

```
NUM_CONTRIBUINTE    NUMBER(9)          not null,  
constraint PK_FUNCIONARIO primary key (NOME_FUNCIONARIO)  
);
```

```
/*=====*/
```

```
/* Table: GERIDA          */
```

```
/*=====*/
```

```
create table GERIDA (  
    COD_SECCAO        NUMBER          not null,  
    NOME_FUNCIONARIO  VARCHAR2(30)    not null,  
    constraint PK_GERIDA primary key (COD_SECCAO, NOME_FUNCIONARIO)  
);
```

```
/*=====*/
```

```
/* Index: GERIDA_FK          */
```

```
/*=====*/
```

```
create index GERIDA_FK on GERIDA (  
    COD_SECCAO ASC  
);
```

```
/*=====*/
```

```
/* Index: GERIDA2_FK        */
```

```
/*=====*/
```

```
create index GERIDA2_FK on GERIDA (  
    NOME_FUNCIONARIO ASC  
);
```

```
/*=====*/
```

```
/* Table: PRODUTO          */
```

```
/*=====*/
```

```
create table PRODUTO (  
    COD_PRODUTO      NUMBER(6)          not null,  
    COD_SECCAO       NUMBER              not null,  
    QUANT_STOCK      NUMBER              not null,  
    PRECO_TABELA     NUMBER(8,2)        not null,  
    NOME              VARCHAR2(30)       not null,  
    constraint PK_PRODUTO primary key (COD_PRODUTO)  
);
```

```
/*=====*/
```

```
/* Index: PERTENCEM_FK          */
```

```
/*=====*/
```

```
create index PERTENCEM_FK on PRODUTO (  
    COD_SECCAO ASC
```

```
);
```

```
/*=====*/
```

```
/* Table: SECCOES */
```

```
/*=====*/
```

```
create table SECCOES (
```

```
    COD_SECCAO      NUMBER          not null,
```

```
    SECCAO          VARCHAR2(10)    not null,
```

```
    constraint PK_SECCOES primary key (COD_SECCAO)
```

```
);
```

```
/*=====*/
```

```
/* Table: VENDAS */
```

```
/*=====*/
```

```
create table VENDAS (
```

```
    COD_VENDA       NUMBER(6)       not null,
```

```
    COD_PRODUTO     NUMBER(6),
```

```
    COD_CLIENTE     NUMBER(6)       not null,
```

```
    PRECO_UNITARIO  NUMBER(8,2)     not null,
```

```
    DATA_VENDA     DATE            not null,
```

```
    QUANTIDADE_VENDIDA NUMBER        not null,
```

```
    constraint PK_VENDAS primary key (COD_VENDA)
```



```
);
```

```
/*=====*/
```

```
/* Index: TEM_FK */
```

```
/*=====*/
```

```
create index TEM_FK on VENDAS (
```

```
    COD_CLIENTE ASC
```

```
);
```

```
/*=====*/
```

```
/* Index: INCLUI_FK */
```

```
/*=====*/
```

```
create index INCLUI_FK on VENDAS (
```

```
    COD_PRODUTO ASC
```

```
);
```

```
alter table ENTRADA
```

```
    add constraint FK_ENTRADA_CONTEM_PRODUTO foreign key (COD_PRODUTO)
```

```
        references PRODUTO (COD_PRODUTO);
```

```
alter table ENTRADA
```

```
    add constraint FK_ENTRADA_CONTROLAD_FUNCIONA foreign key (NOME_FUNCIONARIO)
```

```
references FUNCIONARIO (NOME_FUNCIONARIO);
```

```
alter table ENTRADA
```

```
add constraint FK_ENTRADA_FORNECIDA_FORNECED foreign key (COD_FORNECEDOR)  
references FORNECEDOR (COD_FORNECEDOR);
```

```
alter table GERIDA
```

```
add constraint FK_GERIDA_GERIDA_SECCOES foreign key (COD_SECCAO)  
references SECCOES (COD_SECCAO);
```

```
alter table GERIDA
```

```
add constraint FK_GERIDA_GERIDA2_FUNCIONA foreign key (NOME_FUNCIONARIO)  
references FUNCIONARIO (NOME_FUNCIONARIO);
```

```
alter table PRODUTO
```

```
add constraint FK_PRODUTO_PERTENCEM_SECCOES foreign key (COD_SECCAO)  
references SECCOES (COD_SECCAO);
```

```
alter table VENDAS
```

```
add constraint FK_VENDAS_INCLUI_PRODUTO foreign key (COD_PRODUTO)  
references PRODUTO (COD_PRODUTO);
```

```
alter table VENDAS
```

```
add constraint FK_VENDAS_TEM_CLIENTE foreign key (COD_CLIENTE)  
references CLIENTE (COD_CLIENTE);
```

Cap 8 - Pesquisas SQL

Q1 - Quantidade em stock de ratos logitech.

```
SELECT quant_stock
```

```
FROM produto
```

```
WHERE nome = 'Rato Logitech';
```

Q2 - Número de contribuinte dos funcionário responsáveis pela entrada de produtos fornecidos pela Nestle, ordenado de forma decrescente pela idade.

```
SELECT funcionario.num_contribuinte
```

```
FROM funcionario, entrada, fornecedor
```

```
WHERE funcionario.nome_funcionario = entrada.nome_funcionario
```

```
AND entrada.cod_fornecedor = fornecedor.cod_fornecedor
```

```
AND fornecedor.nome_forn = 'NESTLE'
```

```
ORDER BY funcionario.idade DESC;
```

Q3 - Nome dos produtos vendidos no mês de Janeiro de 2022 ordenados pelo código de produto.

```
SELECT produto.nome
```

```
FROM produto, vendas  
WHERE produto.cod_produto = vendas.cod_produto  
AND to_char(vendas.data_venda, 'DD-MM-YYY') LIKE '%01-2022'  
ORDER BY cod_produto ASC;
```

Q4 - Código do fornecedor do produto incluído na venda com código de venda 05.

```
SELECT fornecedor.cod_forn  
FROM fornecedor, produto, vendas  
WHERE produto.cod_produto = entrada.cod_produto  
AND vendas.cod_produto = vendas.cod_produto  
AND vendas.cod_venda = 05;
```

Q5 - Morada do cliente que comprou mais de 5 produtos pertencentes à secção de Informática.

```
SELECT cliente.morada  
FROM cliente, vendas, produto, seccoos  
WHERE vendas.cod_cliente = clientes.cod_cliente  
AND vendas.cod_produto = produto.cod_produto  
AND produto.cod_seccao = seccoos.cod_seccao  
AND seccoos.seccao = 'Informática'  
AND vendas.quantidade_vendida > 5;
```

Cap 9 - Conclusões

Com o desenvolvimento de uma base de dados para um armazém, é necessário destacar o quão complexo este tipo de negócio é, refletindo sobre isto tivemos de adaptar várias vezes o nosso trabalho para chegarmos a um ponto adequado, devido a todo grupo de variáveis. Primeiramente a distribuição e organização de produtos num armazém é muito facilitada com a implementação de uma base de dados, em segundo a gestão de todos os funcionários, bem como o que cada um deles gere e controla, torna-se mais fácil. Por último o controlo da entrada e saída de produtos que ocorre durante uma venda ou uma entrega feita por um fornecedor, que seria muito mais difícil sem a ajuda de um sistema como uma base de dados onde são registadas todas as operações ocorridas neste meio.

A realização deste trabalho também serve como um chamada de atenção para o facto de haver uma grande necessidade para uso deste tipo de ferramentas em muitos tipos de negócios e não apenas para um armazém. É notória a grande ajuda que este sistema oferece, havendo uma maior organização podendo consecutivamente melhorar os lucros do negócio.

Cap 10 - Referências Bibliográficas

Cap 11 - Anexos