



Cadeia de lojas de informática

Base de Dados



2021/2021 – Licenciatura em Engenharia Informática e computação:

Grupo 302:

Docentes:

Ahmad Eddin - aeddin@fc.up.pt

Carla Lopes - ctl@fe.up.pt

Michel Ferreira - mpferrei@fc.up.pt

Estudantes & Autores:

João Araújo up202004293@up.pt

Rafael Cerqueira up201910200@up.pt

António Correia up201804832@up.pt

Índice

Contexto	3
Diagrama UML.....	4
Diagrama UML Revisto.....	5
Esquema Relacional	6
Análise das Dependências Funcionais e Formas Normais	7
Lista e forma de implementação das restrições	11

Contexto

Pretende-se armazenar informação relativa a uma cadeia de lojas físicas de venda de material informático.

Cada um dos estabelecimentos inerentes à cadeia é identificado por um ID, nº de telefone e morada. Em cada estabelecimento estão associados vários funcionários (entre eles um único gerente), identificados pelo NIF, nome, idade, nº de telefone e salário. Cada funcionário só pode trabalhar num único estabelecimento. A cada estabelecimento e funcionário estão associados um horário (data e hora), sendo que enquanto a loja está em funcionamento, tem de haver pelo menos um funcionário a trabalhar e do mesmo modo, um funcionário só pode trabalhar enquanto a loja estiver aberta.

Cada estabelecimento tem o registo do stock de todos os produtos. Cada produto tem um código identificador, nome, descrição, categoria, preço e fabricante.

De cada cliente interessa armazenar o NIF, nome, idade, nº de telefone, email e morada. Caso pretenda-se comprar em nome de uma empresa, é importante armazenar acrescentadamente o NIF e o nome da mesma.

Cada compra efetuada é identificada por um código único, informação do cliente, data, hora, ID do estabelecimento, informação do funcionário que esteve envolvido, produtos adquiridos, quantidade e preço. Caso seja necessário efetuar envio para a morada do cliente, este é realizado por um funcionário do estabelecimento.

O gerente de cada loja encomenda mais produtos mediante o seu nível de stock.

Diagrama UML

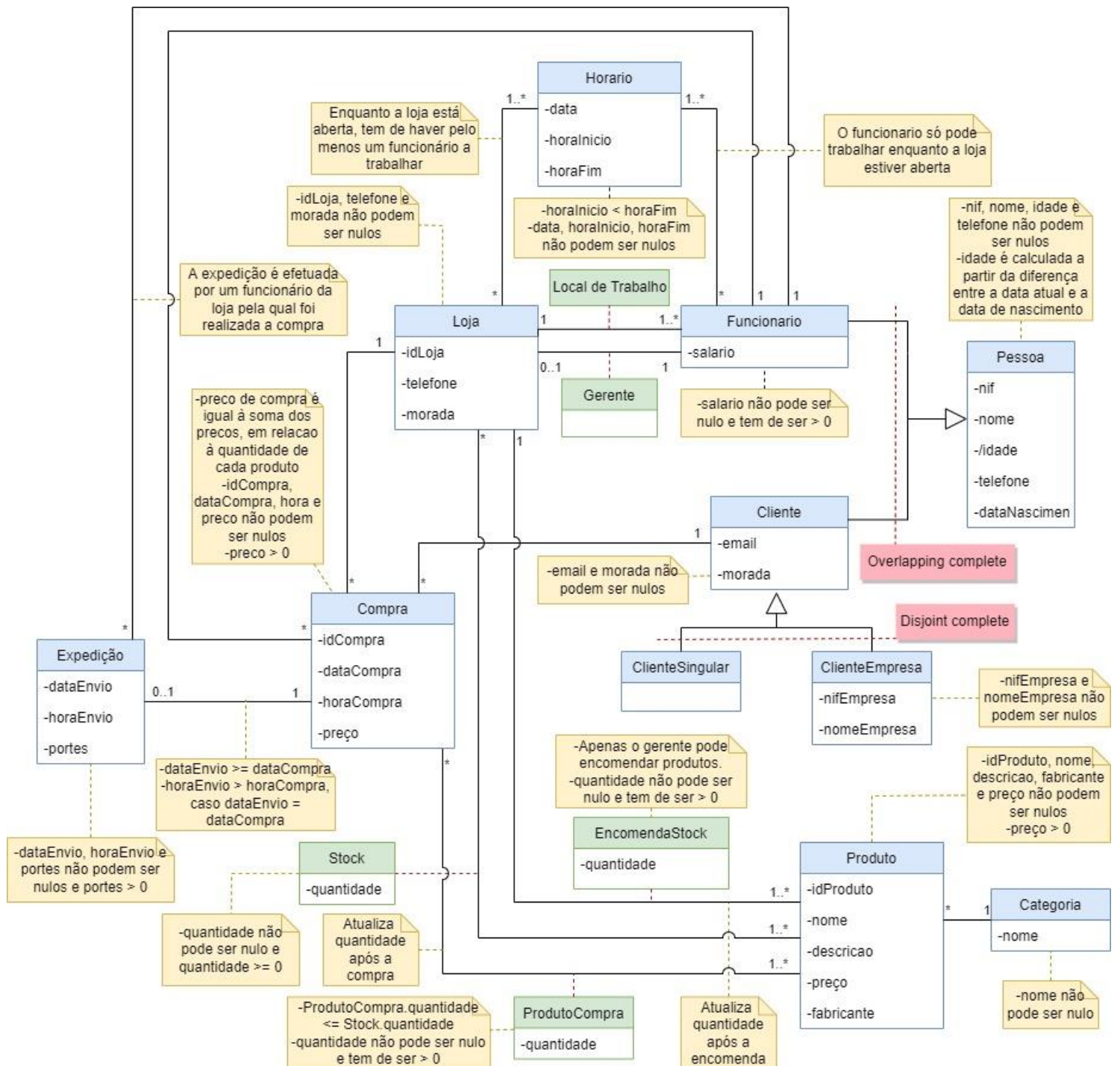
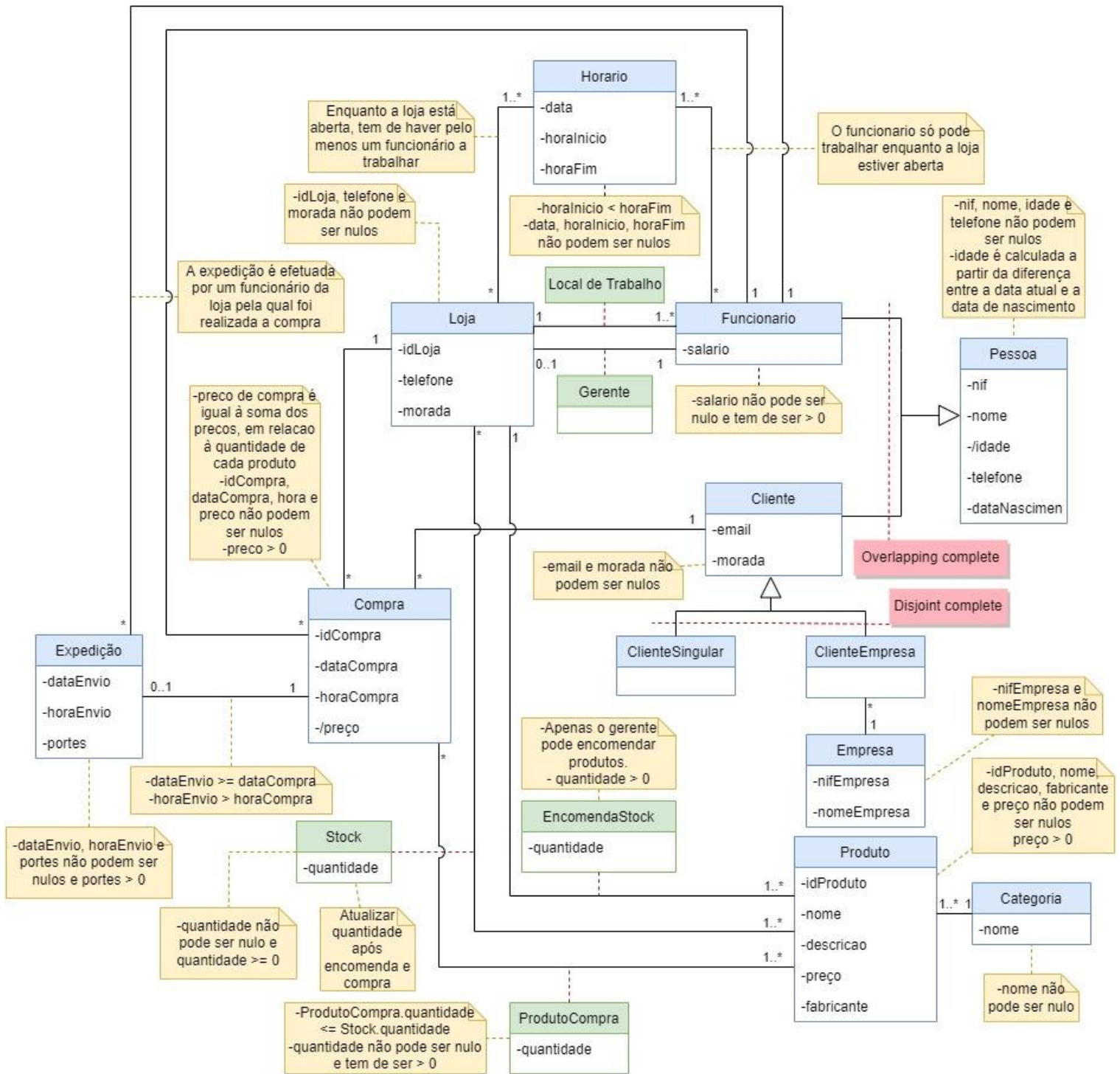


Diagrama UML Revisto



Esquema relacional

- Loja (idLoja, telefone, morada, idGerente->Funcionario)
- Horario (idHorario, dataHorario, horaInicio, horaFim)
- HorarioFuncionario (idHorario->Horario, idFuncionario->Funcionario)
- HorarioLoja (idHorario->Horario, idLoja->Loja)
- Funcionario (idFuncionario->Pessoa, salario)
- Cliente (idCliente->Pessoa, email, morada)
- Pessoa (nif, nome, idade, telefone, dataNascimento)
- LocalDeTrabalho (idFuncionario->Funcionario, idLoja->Loja)
- ClienteEmpresa (idCliente-> Cliente, nifEmpresa-> Empresa)
- Empresa (nifEmpresa, nomeEmpresa)
- ClienteSingular (idCliente-> Cliente)
- Compra (idCompra, dataCompra, horaCompra, preco, idFuncionario->Funcionario, idLoja->Loja, idCliente->Cliente)
- Expedicao (idExpedicao, dataEnvio, horaEnvio, portes, idCompra->Compra, idFuncionario->Funcionario, dataCompra->Compra, horaCompra->Compra)
- Produto (idProduto, nome, descricao, preco, fabricante, idCategoria->Categoria)
- Categoria (idCategoria, nome)
- Stock (idProduto->Produto, idLoja->Loja, quantidade)
- ProdutoCompra (idProduto->Produto, idCompra->Compra, quantidade)
- EncomendaProduto (idEncomenda, idProduto->Produto, idLoja->Loja, quantidade)

Análise das dependências funcionais e formas normais

- Loja (idLoja, telefone, morada, idGerente->Funcionario)
FDs: idLoja->telefone, morada, idGerente
telefone->idLoja, morada, idGerente
morada->idLoja, telefone, idGerente
idGerente->idLoja, telefone, morada
BCNF: Sim
3NF: Sim
- Horario (idHorario, dataHorario, horaInicio, horaFim)
FDs: idHorario->dataHorario, horaInicio, horaFim
data, horaInicio, horaFim->idHorario
BCNF: Sim
3NF: Sim
- HorarioFuncionario (idHorario->Horario, idFuncionario->Funcionario)
FDs: ____
BCNF: Sim
3NF: Sim
- HorarioLoja (idHorario->Horario, idLoja->Loja)
FDs: ____
BCNF: Sim
3NF: Sim
- Funcionario (idFuncionario->Pessoa, salario)
FDs: idFuncionario -> salario
BCNF: Sim
3NF: Sim
- Cliente (idCliente->Pessoa, email, morada)
FDs: idCliente -> email, morada
BCNF: Sim
3NF: Sim

- Pessoa (nif, nome, idade, telefone, dataNascimento)
FDs: nif -> nome, idade, telefone, dataNascimento
BCNF: Sim
3NF: Sim
- LocalDeTrabalho (idFuncionario->Funcionario, idLoja->Loja)
FDs: idFuncionario->idLoja
BCNF: Sim
3NF: Sim
- ClienteEmpresa (idCliente-> Cliente, nifEmpresa, nomeEmpresa)
FDs: idCliente-> nifEmpresa
BCNF: Sim
3NF: Sim
- Empresa (nifEmpresa, nomeEmpresa)
FDs: nifEmpresa-> nomeEmpresa
BCNF: Sim
3NF: Sim
- ClienteSingular (idCliente-> Cliente)
FDs: ____
BCNF: Sim
3NF: Sim
- Compra (idCompra, dataCompra, horaCompra, preco, idFuncionario->Funcionario, idLoja->Loja, idCliente->Cliente)
FDs: idCompra -> dataCompra, horaCompra, preco, idFuncionario, idLoja, idCliente
BCNF: Sim
3NF: Sim
- Expedicao (idExpedicao, dataEnvio, horaEnvio, portes, idCompra->Compra, idFuncionario->Funcionario, dataCompra->Compra, horaCompra->Compra)
FDs: idExpedicao -> dataEnvio, horaEnvio, portes, idCompra, idFuncionario
idCompra -> dataCompra, horaCompra
BCNF: Não

3NF: Não

- Produto (idProduto, nome, descricao, preco, fabricante, idCategoria->Categoria)
FDs: idProduto->nome, descricao, preco, fabricante, idCategoria
BCNF: Sim
3NF: Sim
- Categoria (idCategoria, nome)
FDs: idCategoria->nome
nome->idCategoria
BCNF: Sim
3NF: Sim
- Stock (idProduto->Produto, idLoja->Loja, quantidade)
FDs: idProduto, idLoja->quantidade
BCNF: Sim
3NF: Sim
- ProdutoCompra (idProduto->Produto, idCompra->Compra, quantidade)
FDs: __
BCNF: Sim
3NF: Sim
- EncomendaProduto (idEncomenda, idProduto->Produto, idLoja->Loja, quantidade)
FDs: idEncomenda -> idProduto, idLoja, quantidade
BCNF: Sim
3NF: Sim

Sabendo que:

- Uma relação está na Forma Normal de Boyce-Cood (BCNF) se, para toda a dependência funcional $A \rightarrow B$, $A \rightarrow B$ é trivial ou A é uma superkey/key;
- Uma relação está na 3ª Forma Normal (3NF) se, para toda a dependência funcional $A \rightarrow B$ não trivial, A é uma superkey/key ou B consiste somente de atributos primos.

Podemos constatar que somente a tabela Expedição viola as regras para estar em BCNF e 3NF, dado que através de idExpedição, não é possível obter dataCompra e horaCompra. Contudo, relativamente às restantes FDs nesta base de dados, utilizando acrescentadamente o que fora dito previamente, podemos afirmar que estas cumprem os requisitos para estar em BCNF e 3NF.

Lista e forma de implementação das restrições

Loja

Cada loja é identificada por um ID único.

- idLoja PRIMARY KEY

Cada loja tem somente um único número de telefone, uma única morada e um único gerente

- telefone UNIQUE
- morada UNIQUE
- idGerente UNIQUE

Cada loja tem de ter obrigatoriamente um telefone, uma morada e um gerente atribuído

- telefone NOT NULL
- morada NOT NULL
- idGerente NOT NULL

O ID do gerente corresponde ao ID de um funcionário

- idGerente REFERENCES Funcionario (idFuncionario)

Horario

Cada horário é identificado por um ID único

- idHorario PRIMARY KEY

Cada conjunto constituído pela data, hora inicial e hora final são únicos

- UNIQUE(dataHorario, horaInicio, horaFim)

Cada horário tem de ter obrigatoriamente uma data, uma hora inicial e uma hora final

- dataHorario NOT NULL
- horaInicio NOT NULL
- horaFim NOT NULL

A hora final tem de ser posterior à hora inicial

- CHECK (horaFim > horaInicio)

HorarioFuncionario

Cada conjunto composto pelo idHorario e idFuncionario deve ser único

- PRIMARY KEY (idFuncionario, idHorario)

O ID do horário corresponde a um idHorario da tabela Horario. O ID do funcionário corresponde a um idFuncionario da tabela funcionario.

- idFuncionario REFERENCES Funcionario (idFuncionario)
- idHorario REFERENCES Horario (idHorario)

HorarioLoja

Cada conjunto composto pelo idHorario e idLoja deve ser único

- PRIMARY KEY (idLoja, idHorario)

O ID do horário corresponde a um idHorario da tabela Horario. O ID da loja corresponde a um idLoja da tabela Loja.

- idLoja REFERENCES Loja (idLoja)
- idHorario REFERENCES Horario (idHorario)

Funcionario

Cada funcionário é identificado por um ID único e este é correspondente a um ID da tabela Pessoa

- idFuncionario PRIMARY KEY REFERENCES Pessoa (nif)

Cada funcionário tem um salário associado

- salario NOT NULL
- salario CHECK (salario > 0)

Cliente

Cada cliente é identificado por um ID único e este é correspondente a um ID da tabela Pessoa

- idCliente PRIMARY KEY REFERENCES Pessoa (nif)

Cada cliente deve ter associado uma morada e um email

- morada NOT NULL
- email NOT NULL

Pessoa

Cada pessoa é identificada por um nif único

- nif PRIMARY KEY

Todas as pessoas têm obrigatoriamente associadas um nome, telefone e data de nascimento

- nome NOT NULL
- telefone NOT NULL
- dataNascimento NOT NULL

LocalTrabalho

Cada conjunto composto pelo idFuncionario e idLoja deve ser único

- PRIMARY KEY (idFuncionario, idLoja)

O ID da loja corresponde a um idLoja da tabela Loja. O ID do funcionário corresponde a um idFuncionario da tabela funcionário.

- idFuncionario REFERENCES Funcionario (idFuncionario)
- idLoja REFERENCES Loja (idLoja)

ClienteEmpresa

Cada conjunto composto pelo idCliente e nifEmpresa deve ser único

- PRIMARY KEY (idCliente, nifEmpresa)

O ID da Pessoa corresponde a um idPessoa da tabela Pessoa. O NIF da Empresa corresponde a um nifEmpresa da tabela Empresa.

- idCliente REFERENCES Cliente (idCliente)
- nifEmpresa REFERENCES Empresa (nifEmpresa)

Empresa

Cada empresa é identificada por um nif único

- nifEmpresa PRIMARY KEY

A empresa tem de ter um nome

- nomeEmpresa NOT NULL

ClienteSingular

Cada cliente singular é identificado por um ID único e este é correspondente a um ID da tabela Pessoa

- idCliente PRIMARY KEY REFERENCES Cliente (idCliente)

Compra

Cada compra é identificada por um ID único

- idCompra PRIMARY KEY

Cada compra necessita de ter associado uma data e hora

- dataCompra NOT NULL
- horaCompra NOT NULL

O ID da loja corresponde a um idLoja da tabela Loja. O ID do funcionário corresponde a um idFuncionario da tabela Funcionario. O ID do cliente corresponde a idPessoa da tabela Cliente.

- idLoja REFERENCES Loja (idLoja)
- idFuncionario REFERENCES Funcionario (idFuncionario)
- idCliente REFERENCES Cliente (idCliente)

Expedicao

Cada expedicao é identificada por um ID único

- idExpedicao PRIMARY KEY

Cada compra necessita de ter associado uma data e hora

- dataEnvio NOT NULL
- horaEnvio NOT NULL

O ID do funcionário corresponde a um idFuncionario da tabela Funcionario. O data da compra corresponde a dataCompra da tabela Compra. A hora compra corresponde a dataCompra da tabela Compra.

- idFuncionario REFERENCES Funcionario (idFuncionario)
- dataCompra REFERENCES Compra (dataCompra)
- horaCompra REFERENCES Compra (horaCompra)

A data e hora de envio tem de ser posterior à data e hora de compra

CHECK ((dataEnvio > dataCompra) OR (dataEnvio = dataCompra AND horaEnvio > horaCompra))

Produto

Cada produto é identificado por um ID único

idProduto PRIMARY KEY

Cada produto necessita de ter associado um nome, descrição, preço e fabricante

- nome NOT NULL
- descricao NOT NULL
- preco NOT NULL
- fabricante NOT NULL

O preço tem de ser positivo

- preco CHECK (preco > 0)

O ID da categoria corresponde a um idCategoria da tabela Categoria.

- idCategoria REFERENCES Categoria (idCategoria)

Categoria

Cada categoria é identificada por um id único

- idCategoria PRIMARY KEY

Cada categoria necessita de ter associado um nome

- nome NOT NULL

Stock

O conjunto composto pelo idProduto e idLoja deve ser único

- PRIMARY KEY (idProduto, idLoja)

O ID do produto corresponde ao idProduto da tabela Produto. O ID da loja corresponde ao idLoja da tabela Loja.

- idProduto REFERENCES Produto (idProduto)
- idLoja REFERENCES Loja (idLoja)

O stock do produto da loja deve ser não negativo

- quantidade CHECK (quantidade ≥ 0)

ProdutoCompra

O conjunto composto pelo idProduto e idCompra deve ser único

- PRIMARY KEY (idProduto, idCompra)

O ID do produto corresponde ao idProduto da tabela Produto. O ID da compra corresponde ao idCompra da tabela Compra.

- idProduto REFERENCES Produto (idProduto)
- idCompra REFERENCES Compra (idCompra)

A quantidade do produto deve ser positiva

- quantidade CHECK (quantidade > 0)

EncomendaProduto

Cada encomenda de produtos deve ser identificada por um ID único

- idEncomenda PRIMARY KEY

O ID do produto corresponde ao idProduto da tabela Produto. O ID da compra corresponde ao idCompra da tabela Compra.

- idProduto REFERENCES Produto (idProduto)
- idLoja REFERENCES Loja (idLoja)

A quantidade do produto deve ser positiva

- quantidade CHECK (quantidade > 0)