

BDAD 19/20

Entrega nº 2 – Grupo 607

Grupo 607

Gil Martins Andrade
Jorge Carlos Baptista Duarte
Nuno Filipe Amaral Oliveira

Tema:

Objetivo é criação de uma base de dados para gestão de uma loja que se foca em artigos eletrónicos e artigos culturais e de lazer (i.e.: Fnac, Radio Popular, etc.), tendo esta várias localizações (tendo cada uma destas o seu próprio stock). Teremos uma superclasse pessoa com dados pessoais, e subclasses para funcionários e clientes. Cada um terá um código único, os clientes podem ser Premium (para receberem descontos nas compras), e funcionários terão uma ou várias áreas de especialidade. Os artigos serão definidos, cada um, por um identificador único, pela loja em que se encontram, e alguns outros atributos dependendo do tipo de artigo. Música, livros, filmes, jogos, serão definidos por estilos e artistas. Eletrodomésticos e outros, por marca e modelo de forma a ajudar clientes e funcionários a encontrarem o que pretendem. As compras serão identificadas pelo seu próprio número, data de compra, preço, loja em que foram realizadas, e por um identificador que refere que pessoa fez a compra.

Diagrama de Classes:

Para esta base de dados, a importância maior recai sobre duas super-classes, pessoa e artigo, que guiarão o funcionamento da base de dados.

- Super-Classe “Pessoa”: tem associadas as subclasses Cliente e Funcionário, sendo que um funcionário também pode ser cliente, em dado momento. Havendo, também, atributos comuns entre Clientes e Funcionários, como: o código individual, o seu número de telefone, o seu e-mail e o seu número de identificação fiscal (NIF). Os clientes terão como atributo um identificador relativamente a se são, ou não, clientes Premium.

Já os funcionários, terão a informação de salário recebido, as suas especialidades (poderá ter apenas uma, ou várias), e, como será o seu horário num determinado dia (representado através da classe Dia de Trabalho, com o atributo Dia da Semana e qualificadores, Hora de Entrada e de Saída), sendo que vários funcionários poderão ter vários tipos de horário. No entanto, cada funcionário estará ligado a apenas uma loja, com os seus próprios atributos.

- Super-Classe “Artigo”: tem associadas cinco subclasses referentes a cada tipo de artigo disponível, para compra, estando caracterizadas pelos seguintes atributos: o seu código de barras, o seu nome, o seu preço, a loja em que se encontram e o seu ano de produção.

Cada artigo terá um identificador único e a loja em que se encontram. Se um artigo não se encontrar em nenhuma loja, significa que se encontra num armazém comum a todas as lojas.

Cada um destes produtos tem também os seus próprios atributos, organizados da seguinte forma:

- Eletrodomésticos – a sua marca e a sua função (i.e.: aspirador, frigorífico, etc.);
- Jogos – o seu género, a sua editora e a plataforma para o qual se destina (i.e.: PC, Xbox, Playstation, etc.);
- Filmes/Séries – o seu género, a distribuidora, o seu realizador e formato (i.e.: Blu-Ray, DVD, etc.);
- Livros – o seu género, o seu editor e o seu autor;
- Música – o seu artista, o seu editor e o seu estilo.

Todas estas características utilizadas ajudarão a uma melhor organização e a facilitar a procura e venda de determinados artigos, adequados ao que cada cliente procura.

- Classe “Loja”: A classe “Loja” é a classe com mais ligações. Esta classe está definida pelos atributos: ID, número de telefone, e-mail, número de identificação fiscal (NIF) e a sua morada/localização. Está também ligada a uma classe chamada “Horário Loja”, com os atributos Hora de abertura e de fecho.

- Classe “Compra”: A classe que guardará o registo de todas as transações comerciais. Cada transação será identificada pelo o seu próprio número, data de compra, preço, loja em que foi realizada, e por um identificador que refere que pessoa fez a compra.

Efetuando-se uma compra, o stock será automaticamente atualizado para os valores físicos existentes.

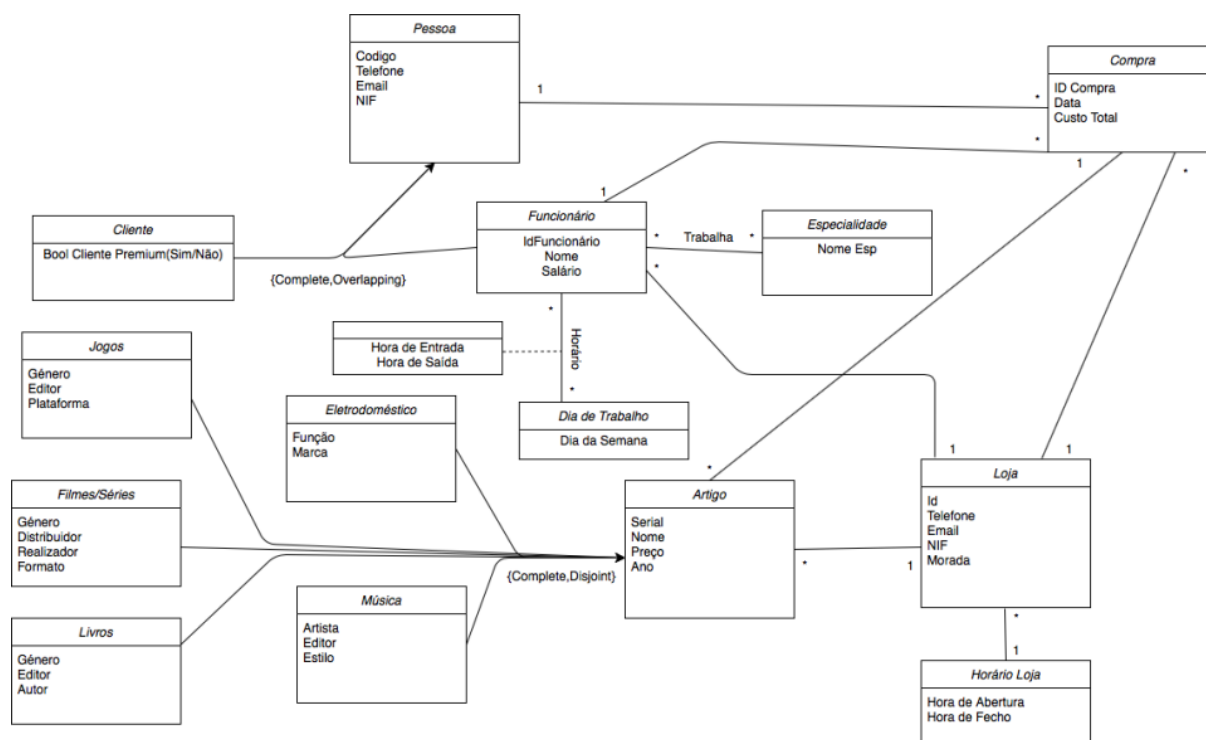


Fig. 1 – UML Revisto

Esquema Relacional:

- Loja(Id, Telefone, Email, NIF, Morada, IdHorário->HorárioLoja)
HorárioLoja(IdHorário, HoraDeAbertura, HoraDeFecho)
- Pessoa(Código, Telefone, Email, NIF)
- Funcionário(Código->Pessoa, IdFuncionario, Nome, Salário, Id->Loja)
DiaDeTrabalho(DiaDaSemana)
- Horário(IdFuncionario->Funcionario, DiaDaSemana->DiadeTrabalho,
HoraDeEntrada, HoraDeSaida)
- Especialidade(NomeEsp)
- Trabalha(IdFuncionario->Funcionario, NomeEsp->Especialidade)
- Cliente(Código->Pessoa, BoolPremium(S/N))
- Artigo(Serial, Nome, Preço, Ano, Id->Loja, IdCompra->Compra)
- Electrodoméstico(Serial->Artigo, Função, Marca)
- Jogos(Serial->Artigo, Género, Editor, Plataforma)
- Filmes/Séries(Serial->Artigo, Género, Distribuidor, Realizador, Formato)
- Livros(Serial->Artigo, Género, Editor, Autor)
- Música(Serial->Artigo, Artista, Editor, Estilo)
- Compra(IDcompra, Data, CustoTotal, id->Loja, Codigo->Pessoa,)

Restrições:

Loja

Id: PRIMARY KEY
Identificador da loja, chave primária e única.
Telefone: UNIQUE NOT NULL
Número de telefone de cada uma das lojas que deverá ser único para cada loja.
Email: UNIQUE NOT NULL
Endereço de e-mail, único para cada uma das lojas.
NIF: UNIQUE NOT NULL
Número de contribuinte da loja, único para cada localidade.
Morada: NOT NULL
Morada física da loja, não pode ser um atributo vazio.
IdHorario: FOREIGN KEY NOT NULL, loja_hora_fk REFERENCES
horarioLoja(idHorario)
Chave estrangeira que liga ao horário de loja.

HorarioLoja

IdHorario: PRIMARY KEY
Identificador da classe Horario de Loja.
HoraDeAbertura: CHECK (abertura>0700 AND abertura <2359)
Atributo que define a hora de abertura da loja, que terá de ser obrigatoriamente entre 07h00 e 23h59.
HoraDeFecho: CHECK (fecho>abertura AND fecho <2359)
Atributo que define a hora de fecho da loja, que terá de ser obrigatoriamente entre a hora de abertura e 23h59.

Pessoa

Codigo: PRIMARY KEY
Identificador único para cada pessoa, cliente e funcionário.
Telefone: UNIQUE
Número de telefone/telemóvel da pessoa, atributo único para cada uma.
NIF: UNIQUE
Número de contribuinte da pessoa, atributo único para cada uma.

Funcionario

Codigo: PRIMARY KEY FOREIGN KEY, func_cod_fk REFERENCES
pessoa(codigo)
Chave primária e estrangeira, código da Pessoa.
IdFuncionario: NOT NULL UNIQUE
Identificador do funcionário, único para cada um, não pode ser atributo vazio.
Nome: NOT NULL
Nome do funcionário, não pode ser um atributo vazio.
Salario: CHECK(salario>0.0)
Valor de remuneração do funcionário, tem de ser um valor maior que 0.

DiaDeTrabalho

DiadaSemana: PRIMARY KEY CHECK (diaSemana = "SEGUNDA-FEIRA" OR diaSemana = "TERCA-FEIRA" OR diaSemana = "QUARTA-FEIRA" OR diaSemana = "QUINTA-FEIRA" OR diaSemana = "SEXTA-FEIRA" OR diaSemana = "SABADO" OR diaSemana = "DOMINGO")

Atributo que define o dia da semana, entre Segunda-Feira e Domingo.

Horario

IdFuncionario: FOREIGN KEY, func_cod_fk REFERENCES funcionario(idFuncionario)

Chave estrangeira, identificador do funcionário.

DiaDaSemana: FOREIGN KEY, func_cod_fk REFERENCES diaTrabalho(diaSemana)

Chave estrangeira, indica o dia da semana.

Abertura: CHECK (abertura>0700 AND abertura <2359)

Horário de entrada do funcionário na loja, também entre 07h00 e 23h59.

Fecho: CHECK (fecho>abertura AND fecho <2359)

Horário de saída do funcionário na loja, valor deverá ser maior que a hora de entrada e no máximo igual à hora de fecho, 23h59.

CONSTRAINT horario_pk PRIMARY KEY (idFuncionario,diaSemana)

Relação estabelecida entre cada funcionário e o seu horário.

Especialidade

NomeEsp: PRIMARY KEY CHECK (nome="ELETRODOMESTICOS" OR nome="JOGOS" OR nome="TELEVISAO" OR nome="LIVROS" OR nome="MUSICA")

Nome da especialidade a que cada funcionário se dedica dentro da loja.

Trabalha

IdFuncionario: FOREIGN KEY, trabalha_id_fk REFERENCES funcionario(idFuncionario)

Nome: FOREIGN KEY, trabalha_nome_fk REFERENCES especialidade(nome)

CONSTRAINT trabalha_pk PRIMARY KEY(idFuncionario,nome)

Relação estabelecida entre funcionário e a sua especialidade na loja.

Cliente

Codigo: PRIMARY KEY FOREIGN KEY cliente_cod_fk REFERENCES pessoa(codigo)

Chave estrangeira, atributo referido em Pessoa.

BoolPremium: DEFAULT 0

Atributo que nos indica se o cliente tem um cartão de cliente premium ou não.

Artigo

SN: PRIMARY KEY

Serial number do artigo, um valor único e individual para todos os artigos, mesmo que sejam por exemplo dois eletrodomésticos com o mesmo nome, modelo e marca.

Nome: NOT NULL

Nome do artigo, não pode ser um atributo vazio.

Preço: DEFAULT 0.0

Preço do artigo, se nada for indicado, terá por defeito o valor 0.

Ano: CHECK (ANO>0 AND ANO<=2020)

Ano de lançamento do artigo, maior que zero e menor ou igual a 2020, ano atual.

IdCompra: FOREIGN KEY artigo_loja_fk REFERENCES compra(idCompra)

Chave estrangeira que estabelece ligação à classe de Compra.

Electrodomestico

SN: PRIMARY KEY FOREIGN KEY, CONSTRAINT eletro_sn_fk REFERENCES artigo(SN)

Serial Number estabelecido em Artigo.

Função: NOT NULL

Função do eletrodoméstico, indica se é por exemplo um aspirador, frigorífico, etc.

Não pode ser um atributo vazio.

Marca: NOT NULL

Marca do eletrodoméstico, não pode ser um atributo vazio.

Jogos

SN: PRIMARY KEY FOREIGN KEY, jogos_sn_fk REFERENCES artigo(SN)

Serial Number estabelecido em Artigo.

Genero: CHECK(genero="ACAO" OR genero="AVENTURA" OR genero="RPG" OR genero="SIMULACAO" OR genero="DESPORTO" OR genero="OUTRO")

Género em que o videojogo se insere, deverá ser um dos 6 indicados.

Editor: NOT NULL

Editora que distribui o jogo, não pode ser um atributo vazio.

Plataforma: CHECK (plataforma="PS4" OR plataforma="PS3" OR plataforma="XBOX ONE" OR plataforma="PC" OR plataforma="NINTENDO SWITCH" OR plataforma="PS VITA")

Plataforma para a qual se destina o jogo, deverá ser uma das 6 plataformas indicadas visto serem as únicas vendidas na loja.

Filmes/Series (Entretenimento para Televisão)

SN: PRIMARY KEY FOREIGN KEY, CONSTRAINT tv_sn_fk REFERENCES artigo(SN)

Serial Number estabelecido em Artigo.

Genero: CHECK (genero = "ACAO" OR genero = "COMEDIA" OR genero = "ROMANCE" OR genero = "TERROR" OR genero = " BIOGRAFIA" OR genero = "FANTASIA" OR genero = "DRAMA" OR genero = "SCI-FI")

Género em que o filme/série se insere, deverá ser um dos 8 indicados.

Formato: CHECK (formato = "DVD" OR formato = "BLU-RAY" OR formato = "DIGITAL")

Formato do filme/série que terá de ser DVD, Blu-Ray ou um voucher para formato digital.

Livros

SN: PRIMARY KEY FOREIGN KEY livros_sn_fk REFERENCES artigo(SN)

Serial Number estabelecido em Artigo.

Genero: CHECK(genero="ARTE" OR genero="COMICS" OR genero=" BIOGRAFIA" OR genero=" FANTASIA" OR genero="ROMANCE" OR genero="OUTRO")

Género em que o livro se insere, deverá ser um dos 6 indicados.

Autor: NOT NULL

Autor que escreveu o livro, deverá ser um atributo não vazio.

Musica

SN: PRIMARY KEY FOREIGN KEY, livros_sn_fk REFERENCES artigo(SN)

Serial Number estabelecido em Artigo.

Estilo: CHECK(estilo="POP" OR estilo="RAP" OR estilo="PORTUGUESA" OR estilo="CLASSICA" OR estilo="ROCK" OR estilo="METAL" OR estilo="JAZZ" OR estilo="OUTRO") Estilo musical, deverá ser um dos 8 indicados.

Artista: DEFAULT "ANONIMO"

Artista que criou o álbum musical, na ausência do nome, por defeito é Anónimo.

Compra

IdCompra: PRIMARY KEY

Identificador único de cada compra efetuada, chave primária.

Data: NOT NULL

Data em que foi efetuada a compra, atributo não vazio.

Preço: NOT NULL

Preço total da compra, atributo não vazio.

IdLoja: FOREIGN KEY, compra_idLoja_fk REFERENCES loja(id)

Chave estrangeira, indica em que loja particular foi efetuada determinada compra.

CodigoPessoa: FOREIGN KEY, compra_codP_fk REFERENCES pessoa(codigo)

Chave estrangeira, indica o Funcionário e/ou Cliente associado a determinada compra.

Em todas as subclasses:

Eletrrodomésticos, Filmes/Séries, Livros, Música e Jogos para Artigo; assim como Cliente, Funcionário para Pessoa; os códigos contêm uma instrução ON DELETE CASCADE e ON UPDATE CASCADE de forma a garantir que a dependência em relação à super-classe é respeitada e, as subclasses serão atualizadas/apagadas se o item na super-classe também for.

Análise de Dependências Funcionais e Formas Normais:

Para todas as relações da nossa base de dados, verifica-se que nenhuma está em violação da Forma Normal de Boyce-Codd ou Terceira Forma Normal.

Isto é comprovado pelo facto de que, para cada Dependência Funcional, considerando $A \rightarrow B$, conseguimos sempre obter todos os atributos da relação a partir de A, ou seja, A é uma Super Key.

Pela definição da Forma Normal de Boyce Codd e Terceira Forma Normal:

Forma Normal de Boyce-Codd (ou BCNF) requer que não exista nenhuma dependência funcional não trivial de atributos em algo mais do que um superconjunto de uma chave candidata. Neste estágio, todos os atributos são dependentes de uma chave, de uma chave inteira e de nada mais que uma chave (excluindo dependências triviais, como $A \rightarrow A$);

Terceira Forma Normal (ou 3FN) define que todos os atributos dessa tabela devem ser funcionalmente independentes uns dos outros, ao mesmo tempo que devem ser dependentes exclusivamente da chave primária da tabela;

Assim sendo, estas condições verificam-se e por isso todas as nossas relações estão na Forma Normal de Boyce-Codd ou Terceira Forma Normal.

(Definição de 3FN e BCNF baseado em:

https://pt.wikipedia.org/wiki/Normalização_de_dados)