

Faculdade de Engenharia da Universidade do Porto



BASE DE DADOS DE UM CINEMA

PROJETO DE BASE DE DADOS



Bases de Dados - Grupo 1103

Elementos do Grupo

Inês Sá Pereira Estêvão Gaspar - 202007210 - up202007210@edu.fe.up.pt

Pedro Manuel Costa Aguiar Botelho Gomes - 202006086 - up202006086@edu.fe.up.pt

Pedro Nuno Ferreira Moura de Macedo - 202007531 - up202007531@edu.fe.up.pt

Novembro 2021

Índice

Contexto do Problema	2
Modelo Conceptual/Diagrama UML (Versão Inicial)	3
Modelo Conceptual/Diagrama UML (Revisto após 1ª Entrega)	4
Esquema Relacional	5
Análise de Dependências Funcionais e Formas Normais	6
Lista e Forma de Implementação das Restrições	8

Contexto do Problema

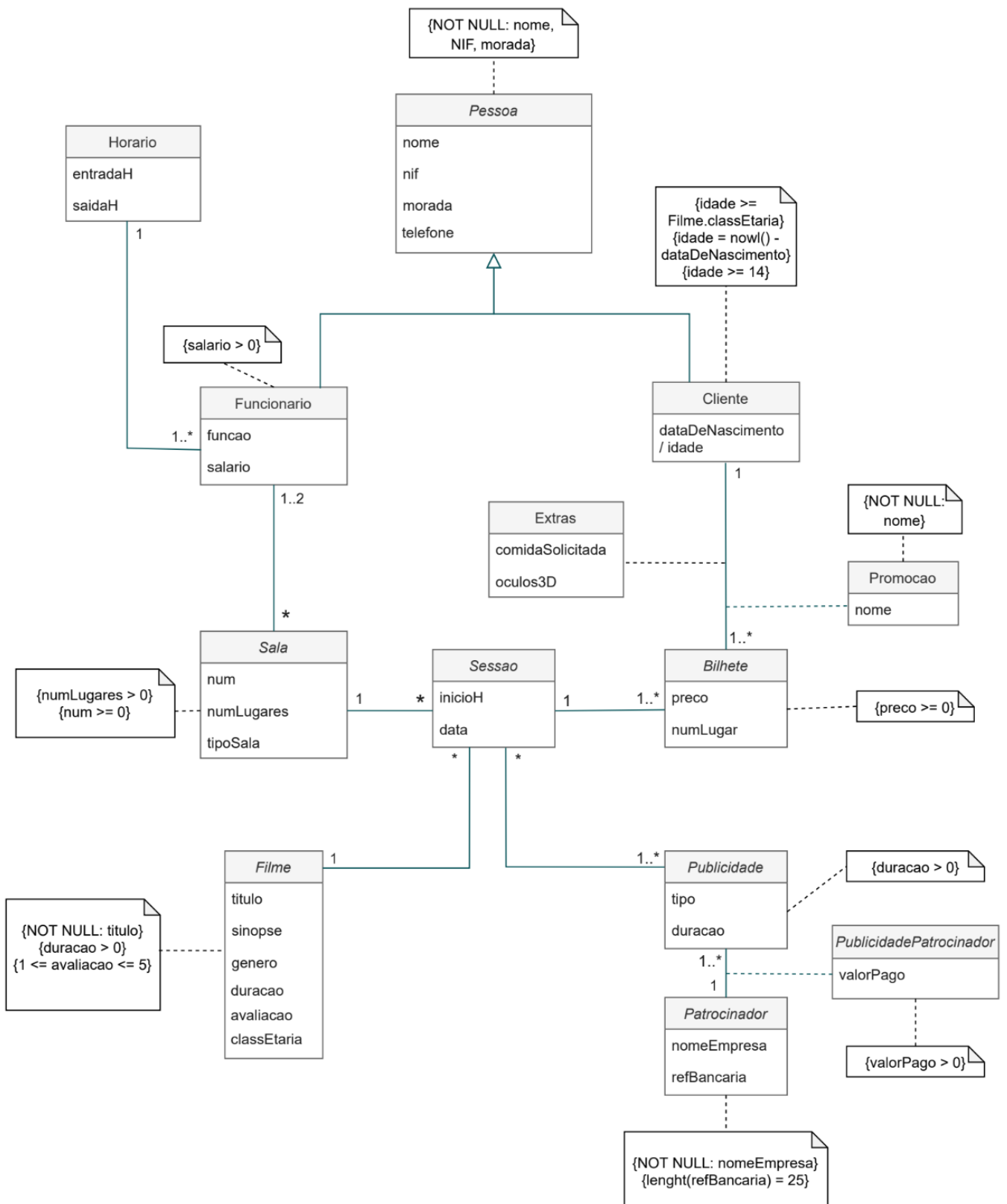
A base de dados desenvolvida tem como objetivo auxiliar um gerente de um cinema na gestão do mesmo. Inicialmente, interessa ao gerente registar informação sobre os clientes do cinema, importando registar os seus dados de faturação (nome, NIF, morada e telefone) e a sua idade ou data de nascimento (necessária para verificação em filmes com limitações de idade). Considera-se como cliente apenas o comprador dos bilhetes.

Um cliente pode comprar um ou vários bilhetes (para si e para acompanhantes), dos quais interessa saber o preço e o número do lugar associado. O cinema apenas aceita compradores com 14 anos ou mais. Além disso, apenas clientes com idade superior à classificação etária do filme podem comprar bilhetes para o mesmo. Podem, no entanto, comprar bilhetes para acompanhantes abaixo da classificação etária. Sempre que o cliente compra bilhetes, escolhe de entre vários tipos de promoções disponíveis para a compra dos mesmos, existindo promoções em que alguns bilhetes possam ser de graça. A promoção deve ser sempre especificada no ato de registo da compra para avaliação das promoções mais lucrativas para o estabelecimento. Cada bilhete dá acesso a uma sessão específica do cinema, da qual importa saber a hora de início e a data para facilitar a organização dos funcionários. A cada sessão está associada uma sala do cinema, da qual se regista o número que identifica a sala, o número de lugares dela e o tipo de sala (tipo de imagem e sistema de som disponíveis nessa sala), de forma a registar as salas preferidas pelo público. Quando um cliente compra bilhetes, pode ainda comprar um conjunto de produtos extras (comida e óculos 3D), registado para facilitar a gestão dos recursos do cinema.

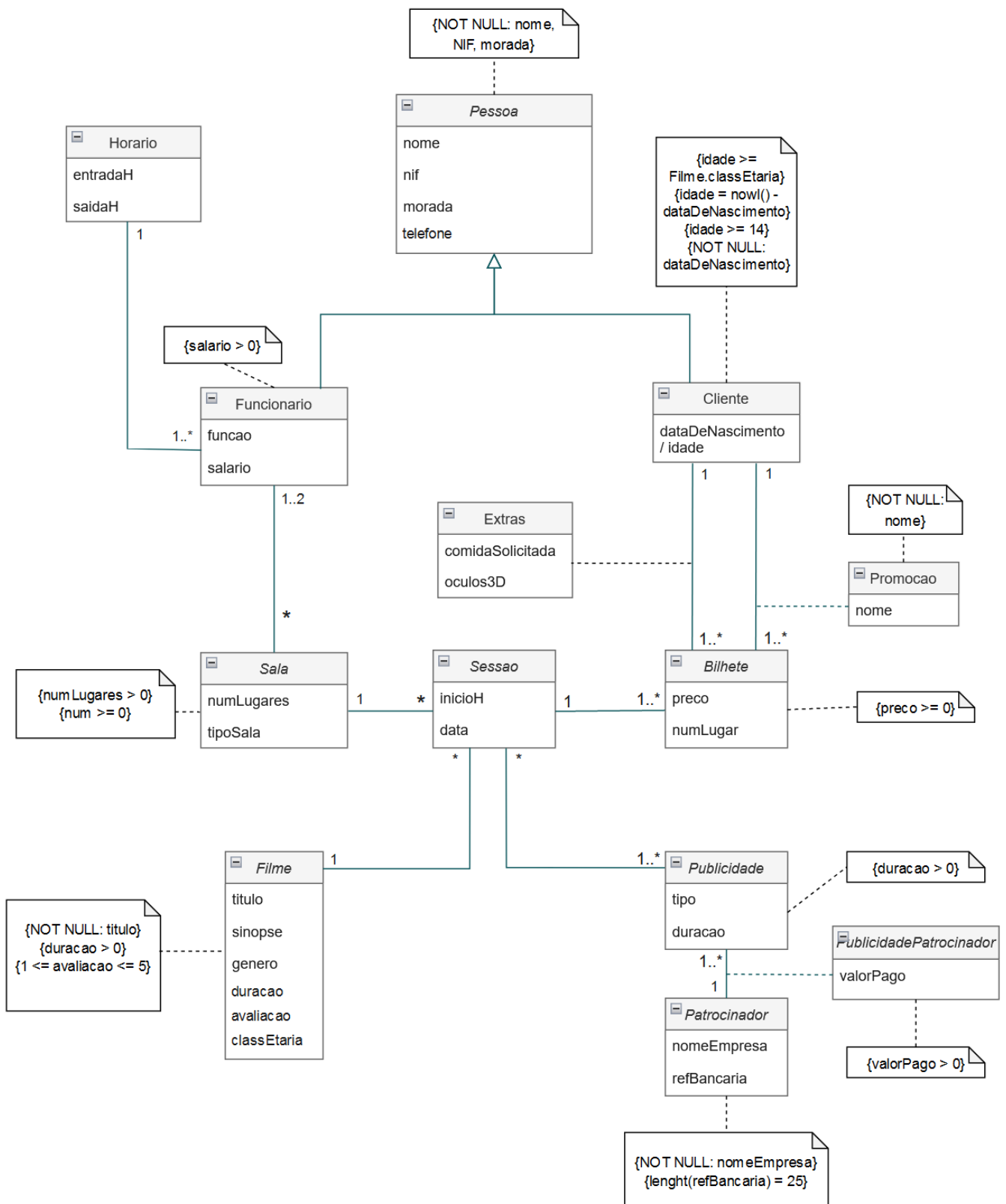
Em cada sessão de cinema passa um filme. É do interesse do gerente manter registadas as informações básicas dos filmes transmitidos (título, sinopse, género, duração, avaliação e classificação etária) para análise dos tipos de filme que geram mais lucro. A avaliação de um filme pode ser de entre 1 e 5 estrelas. A cada início de uma sessão passa também pelo menos um anúncio publicitário, do qual a base de dados deve registar o tipo de produto anunciado e a duração da publicidade. A cada anúncio está associado um patrocinador, do qual nos interessa o nome e a referência bancária, assim como o valor pago para o anúncio em questão ser exibido pelo cinema (dados estritamente necessários para a publicidade ser aceite e transmitida pelo cinema).

Por fim, interessa-nos ainda registar dados de faturação dos funcionários, assim como a sua função e o salário mensal. Alguns funcionários estão associados a salas específicas do cinema (limpeza, manutenção, ...) e outros estão associados à caixa do cinema, por exemplo. Cada sala do cinema tem entre 1 e 2 funcionários associados a ela. Os funcionários possuem um horário, do qual a base de dados mantém informação sobre a hora de entrada e a hora de saída, horário esse a que o gerente deseja ter um acesso facilitado. Vários funcionários podem ter o mesmo horário.

Modelo Conceptual/Diagrama UML (Versão Inicial)



Modelo Conceptual/Diagrama UML (Revisto após 1ª Entrega)



Esquema Relacional

PessoaFuncionario(idPessoaF, nome, nif, morada, telefone, funcao, salario, idHorario->Horario)¹

PessoaCliente(idPessoaC, nome, nif, morada, telefone, dataDeNascimento, idade)

Horario(idHorario, entradaH, saidaH)

Sala(idSala, numLugares, tipoSala, idPessoaF ->PessoaFuncionario)

Sessao(idSessao, inicioH, data, idSala -> Sala, idFilme->Filme)

Filme(idFilme, titulo, sinopse, genero, duracao, avaliacao, classEtaria)

Publicidade (idPublicidade, tipo, duracao)

SessaoPublicidade(idSessao->Sessao, idPublicidade->Publicidade)

Patrocinador(idPatrocinador, nomeEmpresa, refBancaria)

PublicidadePatrocinador(idPublicidade->Publicidade, idPatrocinador ->Patrocinador, valorPago)

Bilhete(idBilhete, preco, numLugar, idSessao ->Sessao)

Extras(idPessoaC->PessoaCliente, idBilhete->Bilhete, comidaSolicitada, oculos3D)

Promocao(idPessoaC->PessoaCliente, idBilhete->Bilhete, nome)

¹ A Generalização Pessoa é disjunta e completa, ou seja, a pessoa ou é um cliente ou é um funcionário. Assim, optamos pela abordagem orientada a objetos para generalizações disjuntas. Isso significa que a relação Pessoa(nome, nif, morada, telefone) pode ser omitida.

Análise de Dependências Funcionais e Formas Normais

PessoaFuncionario(idPessoaF,nif,nome,morada,telefone,funcao,salario,idHorario->Horario)

FD's: idPessoaF->{nome,nif,morada,telefone,funcao,salario,idHorario} **Key:** {idPessoaF}

Está na 3NF e na BCNF porque idPessoaF é uma superchave.

PessoaCliente(idPessoa,nif,nome,morada,telefone, dataDeNascimento, idade)

FD's: idPessoaC->{nome,nif,morada,telefone,dataDeNascimento,idade} **Key:** {idPessoaC}

Está na BCNF e na 3NF porque idPessoaC é uma superchave.

Horario(idHorario,entradaH,saidaH)

FD's: idHorario->{entradaH,saidaH} **Key:** {idHorario}

Está na BCNF e 3NF porque idHorario é uma superchave.

Sala(idSala,numLugares,tipoSala, idPessoaF ->PessoaFuncionario)

FD's: idSala->{numLugares, tipoSala, idPessoaF} **Key:** {idSala}

Está na 3NF e na BCNF porque idSala é superchave.

Sessao(idSessao, inicioH,data, num -> Sala, idFilme->Filme)

FD's: idSessao->{inicioH,data, num, idFilme} **Key:** {idSessao}

Está na 3NF e na BCNF (idSessao é superchave).

Filme(idFilme, titulo, sinopse, genero, duracao, avaliacao, classEtaria)

FD's: idFilme->{titulo,sinopse,genero,duracao,avaliacao,classEtaria} **Key:** {idFilme}

Está na 3NF e na BCNF (idFilme é superchave).

Publicidade (idPublicidade,tipo,duracao)

FD's: idPublicidade->{tipo,duracao} **Key:** {idPublicidade}

Está na 3NF e na BCNF (idPublicidade é superchave).

SessaoPublicidade(idSessao->Sessao,idPublicidade->Publicidade)

FD's: idSessao ->{idPublicidade} **Key:** {idSessao}

Está na 3NF e na BCNF (idSessao é superchave).

Patrocinador(idPatrocinador,nomeEmpresa,refBancaria)

FD's: idPatrocinador -> {nomeEmpresa, refBancaria} **Key:** {idPatrocinador}

Está na 3NF e na BCNF (idPatrocinador é superchave).

PublicidadePatrocinador(idPublicidade->Publicidade, idPatrocinador ->Patrocinador, valorPago)

FD's: idPublicidade->{valorPago, idPatrocinador} **Key:** {idPublicidade}

Está na 3NF e na BCNF (idPublicidade é superchave).

Bilhete(idBilhete, preco, numLugar, idSessao ->Sessao)

FD's: idBilhete->{preco,numLugar,idSessao} **Key:** {idBilhete}

Está na 3NF e na BCNF (idBilhete é superchave).

Extras(idPessoaC->PessoaCliente, idBilhete->Bilhete, comidaSolicitada, oculos3D)

FD's: idBilhete->{comidaSolicitada,oculos3D} **Key:** {idBilhete,idPessoaC}

Está na 3NF (comidaSolicitada e oculos3D são atributos primários) mas não está na BCNF (idBilhete não é superchave).

Promocao(idPessoaC->PessoaCliente, idBilhete->Bilhete,nome)

FD's: idBilhete->{nome,idPessoaC} **Key:** {idBilhete}

Está na 3NF e na BCNF (idBilhete é superchave).

Lista e Forma de Implementação das Restrições

PessoaFuncionario

idPessoaF identifica um único funcionário (PRIMARY KEY UNIQUE). Nome e morada devem sempre ser registados para efeito de faturação (NOT NULL). Duas pessoas diferentes não têm o mesmo número de telefone (UNIQUE). A cada id de horário corresponde um horário da tabela Horario (idHorario REFERENCES Horario). Se um horário for atualizado, todos os funcionários com esse horário têm o seu horário atualizado (ON UPDATE CASCADE). Um horário apenas pode ser apagado caso nenhum funcionário tenha esse horário atualmente (ON DELETE RESTRICT (definição padrão)).

PessoaCliente

idPessoaC identifica um único cliente (PRIMARY KEY UNIQUE). O cliente tem que ter pelo menos 14 anos (CHECK (idade>=14)). A idade do cliente tem que ser igual ou superior à classificação etária do filme (CHECK (idade >= Filme.classEtaria)).

Sala

Não existem duas salas com o mesmo id (PRIMARY KEY). Uma sala tem pelo menos 30 lugares (CHECK (numLugares>=30)).

Sessao

Cada sessão possui um id próprio (PRIMARY KEY UNIQUE). Cada id de sala corresponde a um id da tabela Sala (num REFERENCES Sala). Se uma sala for alterada, as sessões nessa sala sofrem essa atualização. Uma sala apenas pode ser removida se não houver sessões para acontecer nela (ON UPDATE CASCADE ON DELETE RESTRICT). Cada id de filme corresponde a um id da tabela Filme (idFilme REFERENCES Filme). Se um filme for alterado, as sessões onde ele passa também são alteradas. Se um filme deixar de ser passado pelo cinema, as sessões dele são desmarcadas (ON UPDATE CASCADE ON DELETE CASCADE).

Filme

O filme possui um id próprio (PRIMARY KEY UNIQUE). Todos os filmes têm título, género e classificação etária (NOT NULL). A avaliação do filme é dada entre 1 e 5 estrelas (CHECK (avaliacao>0 and avaliacao<=5)).

Publicidade

Cada publicidade tem um id próprio (PRIMARY KEY UNIQUE). Duração máxima da publicidade é de 3 minutos (CHECK (duracao <= 3)).

SessaoPublicidade

Cada id de Sessão corresponde a um id da tabela Sessao (idSessao REFERENCES Sessao). Cada id de Publicidade corresponde a um id da tabela Publicidade (idPublicidade REFERENCES Publicidade). Sempre que uma Publicidade ou Patrocinador é alterada ou apagada, a relação deve ser igualmente alterada (ON UPDATE CASCADE ON DELETE CASCADE).

Patrocinador

Cada patrocinador tem um id próprio (PRIMARY KEY UNIQUE).

PublicidadePatrocinador

Cada id de Publicidade corresponde a um id da tabela Publicidade (idPublicidade REFERENCES Publicidade). Cada id de Patrocinador corresponde a um id da tabela Patrocinador (idPatrocinador REFERENCES Patrocinador). Relação é atualizada sempre que uma publicidade ou patrocinador é alterada (ON UPDATE CASCADE ON DELETE CASCADE). Valor Pago é superior a 0 (check (valorPago > 0)).

Bilhete

Cada bilhete é identificado por um id próprio (PRIMARY KEY UNIQUE). Cada id de sessão corresponde a um id da tabela Sessao (idSessao REFERENCES Sessao). Se uma sessão for cancelada, não há bilhetes para ela. Se for alterada, alteram-se os bilhetes (ON UPDATE CASCADE ON DELETE CASCADE).

Extras

Cada id de cliente corresponde a um id da tabela PessoaCliente (idPessoaC REFERENCES PessoaCliente). Quando um cliente for alterado ou apagado, essa alteração também se verifica no seu registo de extras comprados (ON UPDATE CASCADE ON DELETE CASCADE). Cada id de bilhete corresponde a um id da tabela Bilhete (idBilhete REFERENCES Bilhete). Quando as informações de um bilhete forem alteradas, elas também serão alteradas no registo dos extras. Quando um bilhete é apagado, o registo dos extras também se apaga. (ON UPDATE CASCADE ON DELETE CASCADE).

Promocao

O id do Bilhete define a promoção com que este foi comprado (PRIMARY KEY(idBilhete)). Cada id de cliente corresponde a um id da tabela PessoaCliente (idPessoaC REFERENCES PessoaCliente). Quando um cliente é alterado, essa alteração é feita no registo da Promoção. Quando um cliente é apagado, também as promoções de que usufruiu são apagadas (ON UPDATE CASCADE ON DELETE CASCADE). Cada id de bilhete corresponde a um id da tabela Bilhete (idBilhete REFERENCES Bilhete). Quando um bilhete tem as suas informações alteradas, essas alterações devem ser feitas no registo de promoção. Quando um bilhete é apagado, também as informações relativas à sua promoção devem ser (ON UPDATE CASCADE ON DELETE CASCADE).