



## **BDAD Cinemas**

**Projeto BDAD - 2018/19 - MIEIC**

**Turma 1 Grupo 2**

**Professor das Aulas Laboratoriais:** João Pedro Carvalho Leal Mendes Moreira

### **Autores**

Gustavo Torres, up201706473 (up201706473@fe.up.pt)

Leonardo Moura, up201706907 (up201706907@fe.up.pt)

Mário Mesquita, up201705723 (up201705723@fe.up.pt)

Porto, 26 de maio de 2019

# Índice

<b>Contexto</b>	<b>3</b>
<b>UML</b>	<b>4</b>
<b>Esquema relacional</b>	<b>5</b>
<b>Análise de Dependências Funcionais e Formas Normais</b>	<b>6</b>
<b>Restrições consideradas</b>	<b>11</b>
<b>Interrogação da Base de Dados</b>	<b>16</b>
Interrogação 1 - Informação adicional relevante sobre Filmes	16
Interrogação 2 - Listar filmes disponíveis de acordo com preferências de cliente	16
Interrogação 3 - Rendimento total dos filmes durante período de tempo	16
Interrogação 4 - Permanência nos cinemas dos filmes de determinada franquia	17
Interrogação 5 - Categorias mais/menos populares por cinema	17
Interrogação 6 - Períodos de tempo com mais adesão por cinema	17
Interrogação 7 - Clientes mais fiéis	17
Interrogação 8 - Produtos mais vendidos por categoria de filme	18
Interrogação 9 - Estatísticas monetárias sobre cada tipo de adesão	18
Interrogação 10 - Lugar preferido por sala de cinema	18
<b>Gatilhos</b>	<b>19</b>
Gatilho 1 - Atualização da base de dados na compra de um bilhete	19
Gatilho 2 - Definir o desconto associado a determinado membro	19
Gatilho 3 - Verificar a coerência do preço efetivo de cada pedido	20
Gatilhos relevantes adicionais	21

## Contexto

Pretende-se modelar a base de dados relativa a uma cadeia de cinemas, que intitulamos “BDAD cinemas”.

Sobre a empresa interessa guardar informação relativa aos locais físicos, os filmes disponibilizados, os funcionários, clientes e serviços.

Cada cinema possui as suas salas, que são caracterizadas pelo ecrã (normal, IMAX, ...), pelo sistema de som e pelo número de lugares. Os assentos devem ter associado uma fila e um número, podendo ser normais ou VIP. Os cinemas devem ainda guardar informação básica sobre si (como a localização e contacto), registar os seus funcionários, postos de venda e filmes.

Os postos de venda têm funcionários associados, responsáveis pelos pedidos efetuados. Podem ainda possuir ou não produtos para vender (merchandising, comida, ...).

Um pedido tem um ID, um preço original, um preço efetivo (preço após a aplicação do desconto) e a data do pagamento. Cada pedido está associado a um cliente (que efetuou a compra), ao funcionário que efetuou a transação, a um posto de vendas, aos bilhetes comprados e aos produtos adquiridos. Sobre estes últimos interessa saber a quantidade exata para efeitos de stock. Cada bilhete deve registar qual a sessão e qual o lugar associado.

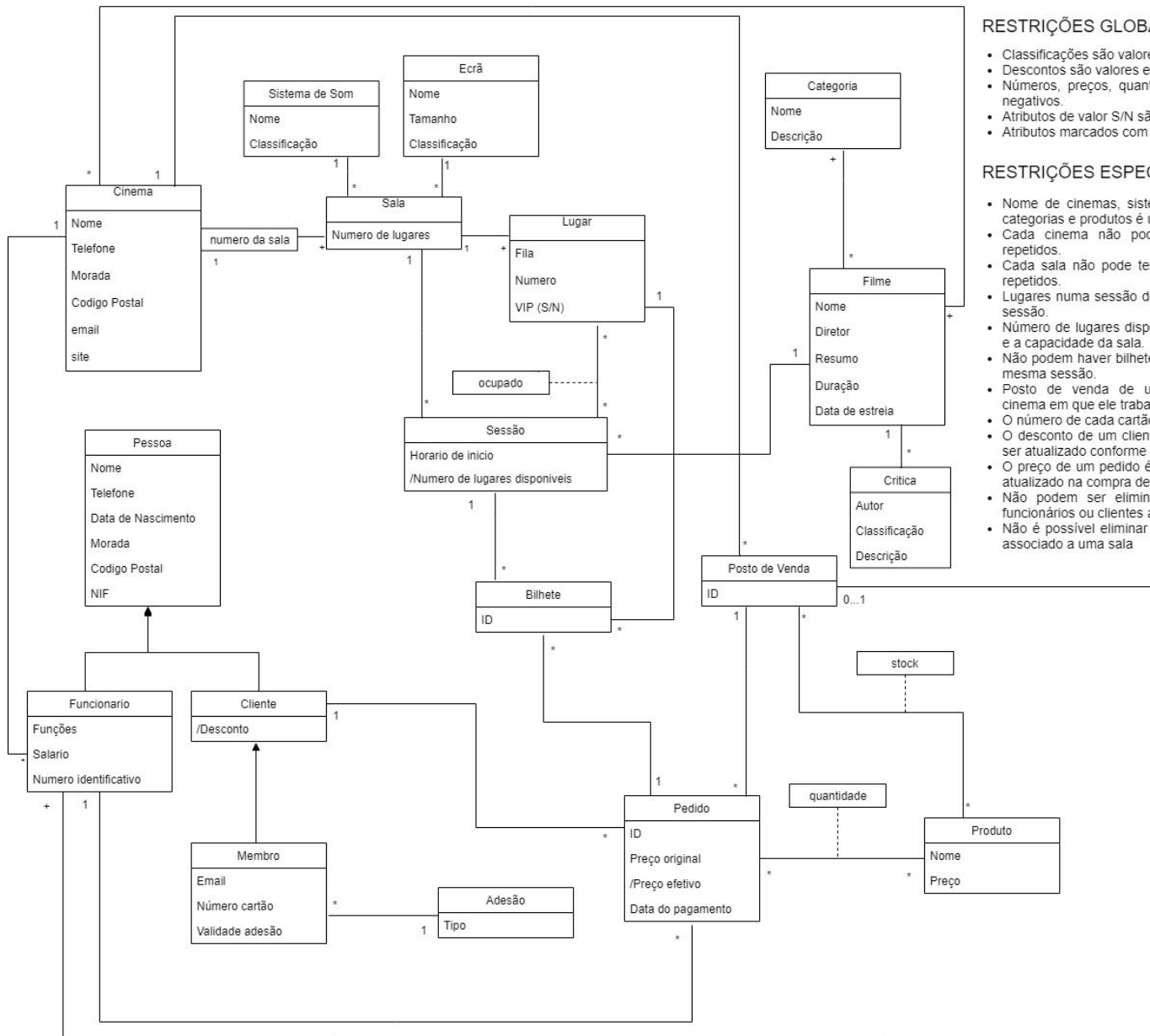
Para controlar quais os filmes disponíveis em cada data nos vários estabelecimentos, são registadas todas as sessões. Estas são caracterizadas por um filme, uma sala, um horário de início e o número de lugares disponíveis. Devem ainda relacionar-se com os lugares da sala respetiva, a fim de gerir quais os assentos disponíveis a cada altura.

A empresa guarda sobre os filmes o seu nome, categorias, diretor, resumo, duração, data de estreia e críticas. Estas últimas devem conter uma classificação, uma descrição e estar associadas a um autor. Sobre as categorias interessa guardar o seu nome e uma breve descrição.

Clientes e funcionários são pessoas com nome, NIF, morada, telefone, etc. Sobre os clientes salienta-se que têm um valor de desconto associado (por omissão igual a 0). Destaca-se sobre os funcionários que devem ser guardadas as suas funções e salário, assim como estar associados a um determinado cinema e, se o caso, a um posto de venda.

Clientes podem também ser Membros, tendo um tipo de Adesão associado e, consequentemente, um valor diferente de desconto. Estes possuem um cartão único com uma determinada validade.

# UML



## RESTRIÇÕES GLOBAIS

- Classificações são valores entre 0 e 5.
- Descontos são valores entre 0 e 100.
- Números, preços, quantidades e durações são não negativos.
- Atributos de valor S/N são booleanos, ou 0 ou 1.
- Atributos marcados com '/' são derivados.

## RESTRIÇÕES ESPECÍFICAS

- Nome de cinemas, sistemas de som, ecrãs, filmes, categorias e produtos é único.
- Cada cinema não pode ter salas com números repetidos.
- Cada sala não pode ter lugares com número e fila repetidos.
- Lugares numa sessão devem pertencer à sala dessa sessão.
- Número de lugares disponíveis numa sessão entre 0 e a capacidade da sala.
- Não podem haver bilhetes para o mesmo lugar numa mesma sessão.
- Posto de venda de um funcionário pertence ao cinema em que ele trabalha.
- O número de cada cartão é único.
- O desconto de um cliente é 0 por omissão, devendo ser atualizado conforme a sua Adesão.
- O preço de um pedido é 0 por omissão, devendo ser atualizado na compra de bilhetes e produtos.
- Não podem ser eliminados os pontos de venda, funcionários ou clientes associados a um pedido.
- Não é possível eliminar um sistema de som ou ecrã associado a uma sala.

## Esquema relacional

- Cinema(cinemaID, nome, telefone, morada, codigoPostal, email, site)
- Sala(salaID, numero, cinema->Cinema, numLugares, som->SistemaSom, ecra->Ecra)
- SistemaSom(sistemaSomID, nome, classificacao)
- Ecra(ecraID, nome, tamanho, classificacao)
- Lugar(lugarID, sala->Sala, fila, numero, vip)
- LugarOcupado(sessaoID->Sessao, lugarID->Lugar)
- Filme(filmeID, nome, diretor, resumo, duracao, dataEstreia)
- Categoria(categorialID, nome, descricao)
- Critica(criticalID, autor, filme->Filme, classificacao, descricao)
- FilmeTemCategoria(filme->Filme, categoria->Categoria)
- CinemaTemFilme(cinema->Cinema, filme->Filme)
- Sessao(sessaoID, horaInicio, lugaresDisponiveis, sala->Sala, filme->Filme)
- Bilhete(bilheteID, sessao->Sessao, lugar->Lugar, pedido->Pedido)
- Pedido(pedidoID, precoOriginal, precoEfetivo, dataPagamento, postoVenda->PostoVenda, funcionario->Funcionario, cliente->Cliente)
- PostoVenda(postoVendaID, cinema->Cinema)
- Produto(produtoID, nome, preco)
- ProdutoDisponivel(produto->Produto, postoVenda->PostoVenda, stock)
- ProdutoAdquirido(produto->Produto, pedido->Pedido, quantidade)
- Pessoa(pessoalID, nome, NIF, telefone, dataNascimento, morada, codigoPostal)
- Funcionario(pessoa->Pessoa, funcoes, salario, cinema->Cinema, postoVenda->PostoVenda, numeroTrabalhador)
- Cliente(pessoa->Pessoa, desconto)
- Membro(pessoa->Cliente, email, numeroCartao, validadeAdesao, adesao->Adesao)
- Adesao(adesaoID, tipo)

# Análise de Dependências Funcionais e Formas Normais

## Cinema

- cinemaID -> nome, telefone, morada, codigoPostal, email, site
- nome -> cinemaID, telefone, morada, codigoPostal, email, site

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Sala

- salaID -> numero, cinema, numLugares, som, ecra
- numero, cinema -> salaID, numLugares, som, ecra

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## SistemaSom

- sistemaSomID -> nome, classificacao
- nome -> sistemaSomID, classificacao

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Ecra

- ecraID -> nome, tamanho, classificacao
- nome -> ecraID, tamanho, classificacao

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Lugar

- lugarID -> sala, fila, numero, vip
- sala, fila, numero -> lugarID, vip

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## LugarOcupado

Esta relação não possui dependências funcionais não-triviais. Por este motivo está na BCNF e, uma vez que a 3NF é um superset da BCNF, também na 3NF.

## Filme

- filmeID -> nome, diretor, resumo, duracao, dataEstreia
- nome -> filmeID, diretor, resumo, duracao, dataEstreia

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Categoria

- categorialID -> nome, descricao
- nome -> categorialID, descricao

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Critica

- criticalID -> autor, filme, classificacao, descricao

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## FilmeTemCategoria

Esta relação não possui dependências funcionais não-triviais. Por este motivo está na BCNF e, uma vez que a 3NF é um superset da BCNF, também na 3NF.

## CinemaTemFilme

Esta relação não possui dependências funcionais não-triviais. Por este motivo está na BCNF e, uma vez que a 3NF é um superset da BCNF, também na 3NF.

## Sessao

- sessaoID -> horaInicio, lugaresDisponiveis, numeroSala, filme

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Bilhete

- bilheteID -> sessaoID, lugar, pedidoID
- sessaoID, lugar -> bilheteID, pedidoID

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Pedido

- pedidoID -> precoOriginal, precoEfetivo, dataPagamento, postoVendaID, funcionario, cliente
- precoOriginal, cliente -> precoEfetivo
- funcionario -> postoVendaID

Esta relação viola a BCNF, uma vez que na dependência funcional não trivial **funcionario>postoVendaID**, funcionario não é uma (super)key. O mesmo se verifica na dependência funcional **precoOriginal, cliente->precoEfetivo**.

Esta relação também viola a 3NF, devido ao facto de, na dependência funcional não trivial **funcionario>postoVendaID**, funcionario não ser uma (super)key e postoVendaID não ser um atributo primo. O mesmo se verifica na dependência funcional **precoOriginal, cliente->precoEfetivo**.



## PostoVenda

- postoVendaID -> numero, cinema
- numero, cinema -> postoVendaID

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Produto

- produtoID -> nome, preco
- nome -> produtoID, preco

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## ProdutoDisponivel

- produto, postoVendaID -> stock

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## ProdutoAdquirido

- produto, pedidoID -> quantidade

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Pessoa

- pessoaID -> nome, NIF, telefone, dataNascimento, morada, codigoPostal
- NIF -> pessoaID, nome, telefone, dataNascimento, morada, codigoPostal

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Funcionario

- pessoalID -> funcoes, salario, cinema, postoVenda, numeroTrabalhador
- cinema -> postoVenda

Esta relação viola a BCNF, uma vez que na dependência funcional não trivial **cinema->postoVenda**, cinema não é uma (super)key.

Esta relação também viola a 3NF, devido ao facto de, na dependência funcional não trivial **cinema->postoVenda**, cinema não ser uma (super)key e postoVenda não ser um atributo primo.

## Cliente

- pessoalID -> desconto

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Membro

- pessoalID -> email, numeroCartao, validadeAdesao, adesao
- numeroCartao -> pessoa, email, validadeAdesao, adesao

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Adesao

- adesaoID -> tipo

Esta relação não viola nem a BCNF nem a 3NF. Isto porque, no caso da BCNF, para cada dependência funcional não trivial (neste caso todas as dependências funcionais), A -> B, A é uma (super)key. Como esta relação não viola a BCNF, também não viola a 3NF, uma vez que a 3NF é um superset da BCNF.

## Restrições consideradas

Em todas as tabelas (exceto as de associação) é utilizado um ID do tipo INTEGER como PRIMARY KEY, com AUTOINCREMENT, prioritariamente pelos seguintes motivos:

- acesso às tabelas mais rápido.
- separar a informação contida nas tabelas do “contexto real”, evitando situações em que seja necessário alterar a chave primária.

Estas restrições PRIMARY KEY estão nomeadas.

Relativamente às chaves estrangeiras foram também consideradas as restrições ON DELETE e ON UPDATE, de forma a manter a integridade referencial das tabelas.

### Cinema

- cada cinema deve ter um ID único (PRIMARY KEY).
- não pode haver dois cinemas com o mesmo nome (UNIQUE).
- todos os cinemas devem ter pelo menos um ID, nome, telefone, morada, código postal e email (NOT NULL).

### Sala

- cada sala deve ter um ID único (PRIMARY KEY).
- cinema é chave estrangeira para Cinema, sistemaSom para SistemaSom e ecrã para Ecrã (REFERENCES).
- o número de lugares deve ser não negativo (CHECK).
- não podem haver salas no mesmo cinema com o mesmo número (UNIQUE).
- todas as salas devem ter pelo menos um ID, numero, cinema, número de lugares, sistema de som e ecrã (NOT NULL).
- não é possível eliminar um sistema de som ou ecrã associado a uma sala (ON DELETE RESTRICT).

### SistemaSom

- cada sistema de som deve ter um ID único (PRIMARY KEY).
- a classificação deve ser um número entre 0 e 5 (CHECK).
- não podem haver sistemas de som com o mesmo nome (UNIQUE).
- todos os sistemas de som devem ter pelo menos um ID e nome (NOT NULL).

### Ecrã

- cada ecrã deve ter um ID único (PRIMARY KEY).
- as polegadas deve ser um número não negativo (CHECK).
- a classificação deve ser um número entre 0 e 5 (CHECK).
- não podem haver sistemas de som com o mesmo nome (UNIQUE).
- todos os ecrãs devem ter pelo menos um ID e nome (NOT NULL).

## Lugar

- cada lugar deve ter um ID único (PRIMARY KEY).
- sala é chave estrangeira para Sala (REFERENCES).
- o atributo VIP apenas pode ser 0 ou 1 (CHECK).
- não podem haver lugares na mesma sala com a mesma fila e número (UNIQUE).
- todos os lugares devem ter pelo menos um ID, sala, fila, numero e VIP (NOT NULL).

## LugarOcupado

- cada par de sessão e lugar deve ser único (PRIMARY KEY).
- sessao é chave estrangeira para Sessao e lugar para Lugar (REFERENCES).
- todos os tuplos devem ter pelo menos uma sessao e um lugar (NOT NULL).
- o lugar deve pertencer à sala da sessão (TRIGGER).

## Filme

- cada filme deve ter um ID único (PRIMARY KEY).
- a duração deve ser maior que 0 (CHECK).
- todos os filmes devem ter pelo menos um ID e um nome (NOT NULL).

## Categoria

- cada categoria deve ter um ID único (PRIMARY KEY).
- todas as categorias devem ter pelo menos um ID e um nome (NOT NULL).

## Critica

- cada critica deve ter um ID único (PRIMARY KEY).
- a classificação deve ser um número entre 0 e 5 (CHECK).
- todas as criticas devem ter pelo menos um ID, autor, classificação e descrição (NOT NULL).

## FilmeTemCategoria

- cada par de filme e categoria deve ser único (PRIMARY KEY).
- filme é chave estrangeira para Filme e categoria para Categoria (REFERENCES).

## CinemaTemFilme

- cada par de cinema e filme deve ser único (PRIMARY KEY).
- filme é chave estrangeira para Filme e cinema para Cinema (REFERENCES).

## Sessao

- cada sessão deve ter um ID único (PRIMARY KEY).
- sala é chave estrangeira para Sala e filme para Filme (REFERENCES).
- o valor dos lugares disponíveis é 0 por omissão (DEFAULT).
- o valor dos lugares disponíveis deve ser não negativo (CHECK).
- todas as sessões devem ter pelo menos um ID, hora de início, número de lugares disponíveis, sala e filme (NOT NULL).
- o valor dos lugares disponíveis deve ser menor do que a capacidade da sala (TRIGGER).

## Bilhete

- cada bilhete deve ter um ID único (PRIMARY KEY).
- sessao é chave estrangeira para Sessao, lugar para Lugar e pedido para Pedido (REFERENCES).
- não podem existir bilhetes para uma sessão com o mesmo lugar (UNIQUE).
- todos os bilhetes devem ter pelo menos um ID, sessão, lugar e pedido (NOT NULL).
- não podem haver lugares repetidos para a mesma sessão (TRIGGER).

## Pedido

- cada pedido deve ter um ID único (PRIMARY KEY).
- postoVenda é chave estrangeira para postoVenda, funcionario para Funcionario e cliente para Cliente (REFERENCES).
- todos os pedidos devem ter pelo menos um ID, preço original, preço efetivo, data de pagamento, posto de venda, funcionário e cliente (NOT NULL).
- os preços não podem ser negativos (CHECK).
- não podem ser eliminados os pontos de venda, funcionários ou clientes associados a um pedido (ON DELETE RESTRICT).
- o preço original e efetivo são 0 por omissão (DEFAULT).
- o preço original deve ser construído com base nos bilhetes e produtos adquiridos (TRIGGER).
- o preço efetivo deve ser derivado do preço original e do desconto associado ao cliente (TRIGGER).

## PostoVenda

- cada posto de venda deve ter um ID único (PRIMARY KEY).
- cinema é chave estrangeira para Cinema (REFERENCES).
- todos os postos de venda devem ter um ID e um cinema (NOT NULL).

## Produto

- cada produto deve ter um ID único (PRIMARY KEY).
- todos os produtos devem ter um ID, nome e preço (NOT NULL).
- o preço deve ser não negativo (CHECK).

### ProdutoDisponivel

- cada par de produto e postoVenda deve ser único (PRIMARY KEY).
- produto é chave estrangeira para Produto e postoVenda para PostoVenda (REFERENCES).
- todos os tuplos devem ter um produto, posto de venda e stock (NOT NULL).
- o stock deve ser não negativo (CHECK).

### ProdutoAdquirido

- cada par de produto e pedido deve ser único (PRIMARY KEY).
- produto é chave estrangeira para Produto e pedido para Pedido (REFERENCES).
- todos os tuplos devem ter um produto, pedido e quantidade (NOT NULL).
- a quantidade deve ser igual ou superior a 1 (CHECK).

### Pessoa

- cada pessoa deve ter um ID único (PRIMARY KEY).
- todas as pessoas devem ter pelo menos um ID, nome e NIF (NOT NULL).
- não podem haver pessoas com o mesmo NIF (UNIQUE).

### Funcionario

- cada funcionário deve ter um ID único (PRIMARY KEY).
- pessoaID é uma chave estrangeira para Pessoa, cinema para Cinema e postoVenda para PostoVenda (REFERENCES).
- todos os funcionários devem ter pelo menos um ID, funções, salário e número de trabalhador (NOT NULL).
- o número de trabalhador deve ser não negativo e o NIF deve ter 9 algarismos (CHECK).
- funcionários não ativos possuem os campos de Cinema e postoVenda a NULL.
- o posto de venda deve pertencer ao cinema em que o funcionário trabalha (TRIGGER).

### Cliente

- cada cliente deve ter um ID único (PRIMARY KEY).
- pessoaID é uma chave estrangeira para Pessoa (REFERENCES).
- todos os clientes devem ter um ID e um desconto (NOT NULL).
- o desconto deve estar entre 0 e 100 (CHECK).
- o desconto por omissão de um novo cliente é 0 (DEFAULT).
- o desconto deve ser atualizado caso o Cliente seja um Membro e possua uma Adesão ativa, ou no caso de a sua Adesão expirar (TRIGGER).

## Membro

- cada membro deve ter um ID único (PRIMARY KEY).
- clienteID é uma chave estrangeira para Cliente e adesao para Adesao (REFERENCES).
- não podem existir cartões com o mesmo número (UNIQUE).
- cada membro deve ter um ID, email, número de cartão, adesão e validade de adesão (NOT NULL).

## Adesao

- cada adesão deve ter um ID único (PRIMARY KEY).
- cada adesão deve ter um ID e tipo (NOT NULL).

# Interrogação da Base de Dados

## Interrogação 1 - Informação adicional relevante sobre Filmes

Esta interrogação pretende listar, para cada filme, informação adicional considerada relevante, seja para o cliente ou para a empresa de cinemas.

Assim, além do nome de cada filme, apresenta-se a sua classificação média (com base nas críticas recebidas), número de críticas recebidas, número de sessões existentes e número de bilhetes vendidos.

## Interrogação 2 - Listar filmes disponíveis de acordo com preferências de cliente

Esta interrogação visa simular o ato de um cliente pesquisar por filmes disponíveis de acordo com as suas preferências.

No caso implementado, admite-se um cliente que pretenda pesquisar filmes de “Fantasia” disponíveis a 24/05/2019 cuja classificação média seja superior a 3,0. É listado o nome de todos os filmes relevantes e o número de sessões disponíveis na data mencionada.

## Interrogação 3 - Rendimento total dos filmes durante período de tempo

Esta interrogação providencia informação sobre o rendimento total de cada filme durante um determinado período de tempo, contabilizando o dinheiro proveniente de bilhetes e de produtos.

É apresentado o nome dos filmes junto do seu rendimento, sendo imposto um limite de apenas mostrar os 10 resultados de maior rendimento.



## Interrogação 4 - Permanência nos cinemas dos filmes de determinada franquia

Esta interrogação mostra informação sobre a permanência nos cinemas de cada filme de uma determinada franquia, isto é, o intervalo de dias desde que o filme estreou até à última sessão registada na base de dados.

No caso implementado, admite-se que a franquia é “Pokemon”, pelo que são apenas admitidos filmes que possuem esta palavra (ou derivados da palavra) na sua designação ou descrição.

## Interrogação 5 - Categorias mais/menos populares por cinema

Esta interrogação lista, sobre cada cinema, qual a categoria mais popular e qual a categoria menos popular. A popularidade é determinada pelo número de bilhetes vendidos para sessões relativas a cada filme.

É apresentado o nome dos cinemas, juntamente do nome das categorias mais e menos populares, por esta ordem.

## Interrogação 6 - Períodos de tempo com mais adesão por cinema

Interroga a base de dados, sobre cada cinema, qual o período de tempo em que houve uma maior afluência de espectadores às salas de cinemas, classificando em 7 momentos do dia : ‘Início da Manhã ‘ , ‘Fim da Manhã ‘ , ‘Início da Tarde‘ , ‘Meio da Tarde‘ , ‘Fim da Tarde‘ , ‘Noite’ e ‘Madrugada’. A afluência é determinada pelo número de bilhetes comprados para cada sessão em determinado horário.

É apresentado para cada cinema o seu nome, e períodos do dia com mais e menos espectadores. É também apresentado para cada período o número de bilhetes associado.

## Interrogação 7 - Clientes mais fiéis

Esta interrogação lista os dois clientes mais fiéis para cada cinema da empresa. O cálculo é feito consoante o número de vezes que um pedido foi efetuado em nome do cliente.

A informação apresentada segue o formato: Nome do cinema, Cliente mais fiel, Número de pedidos do cliente mais fiel, Segundo cliente mais fiel, Número de pedidos do segundo cliente mais fiel.

## Interrogação 8 - Produtos mais vendidos por categoria de filme

Esta interrogação procura determinar, para cada categoria de filme, quais os produtos mais e menos vendidos, a fim de ser possível inferir alguma estatística útil para a empresa de cinemas.

A tabela resultante lista o nome de cada categoria associado do produto mais vendido, o segundo produto mais vendido, o produto menos vendido e o respetivo número de vendas a cada categoria.

## Interrogação 9 - Estatísticas monetárias sobre cada tipo de adesão

Esta interrogação procura fornecer informações monetárias sobre os vários tipo de adesão, de modo a ser possível retirar conclusões acerca dos descontos que estão a ser aplicados.

A interrogação apresenta, para cada tipo de Adesão: o dinheiro total que seria gasto pelos membros dessa adesão se não fosse feito desconto; o dinheiro efetivo gasto pelos membros aplicando descontos; a percentagem de dinheiro que foi descontado.

Salienta-se que esta interrogação admite que, para cada adesão, os descontos podem ser variáveis (por vários motivos, como por exemplo campanhas específicas para Estudantes ou Crianças). Se os valores de desconto fossem fixos, a percentagem de dinheiro descontado na tabela também o seria.

## Interrogação 10 - Lugar preferido por sala de cinema

Esta interrogação procura determinar, para cada sala de cinema, qual o lugar favorito pelos clientes. O melhor lugar é considerado aquele que tem um maior número de bilhetes associados.

É apresentado na tabela o nome de cada cinema e número de sala, junto do lugar preferido (resultante da concatenação da fila do lugar com o número do lugar, no formato "Fila-Lugar") e o respetivo número de bilhetes associado.

## Gatilhos

### Gatilho 1 - Atualização da base de dados na compra de um bilhete

Aquando da compra de um bilhete, deve ser atualizada a base de dados das seguintes formas:

- Adicionada uma relação LugarOcupado para indicar que o lugar do novo bilhete se encontra em uso para a sessão a que diz respeito;
- Atualizado o campo 'lugaresDisponíveis' da respectiva sessão, decrementando-o de 1 unidade.

Este gatilho está definido para agir após uma inserção na tabela Bilhete.

No ficheiro 'gatilho1\_verifica.sql' é criada inicialmente uma sessão sem bilhetes. São depois inseridos 2 bilhetes para essa sessão, sendo criadas 2 entradas na tabela LugarOcupado e passado o valor de 'lugaresDisponíveis' da sessão a 2 unidades abaixo do inicial.

### Gatilho 2 - Definir o desconto associado a determinado membro

Cada cliente do cinema tem um certo desconto associado que, por omissão, é igual a 0. Quando um cliente passa a ser também um Membro, este valor deve ser atualizado conforme o tipo de Adesão associado. Assim, este gatilho foi criado para, quando um membro é adicionado à base de dados, o valor de desconto do cliente correspondente ser atualizado.

Este valor dependerá do tipo de Adesão, que pode ser 'Estudante', 'Criança', 'Jovem', entre outras. Foi definido para agir após uma inserção na tabela Membro.

No ficheiro 'gatilho2\_verifica.sql' são criados inicialmente clientes sem adesão, cujo desconto é 0. Posteriormente, são identificados como Membros e o seu valor de desconto é atualizado.

## Gatilho 3 - Verificar a coerência do preço efetivo de cada pedido

Cada pedido no cinema é caracterizado por um 'precoOriginal', dependente do número de bilhetes comprados e da quantidade e tipo de produtos adquiridos. No entanto, o 'precoEfetivo' que é cobrado aos clientes poderá variar se estes possuírem uma adesão e, por consequência, um desconto associado.

Este gatilho foi implementado para garantir a coerência do valor 'precoEfetivo', calculando-o novamente sempre que uma atualização for feita na tabela Pedido que poderá conduzir a um valor errado ou desatualizado deste atributo. Assim, é chamado ON UPDATE dos valores 'precoOriginal', 'precoEfetivo' ou cliente (poderá alterar o desconto associado).

No ficheiro 'gatilho3\_verifica.sql', inicialmente são criados pedidos cujos valores de 'precoEfetivo' estão propositadamente errados (utilizando um trigger ON INSERT isto não devia ser possível, mas aqui não está a ser considerado). De seguida, é feito um Update na tabela Pedido que aumenta o preço original de cada pedido por 10. Observando os resultados da interrogação final, verificamos que realmente os novos valores de 'precoEfetivo' refletem o aumento e a aplicação do desconto de cada cliente.

## Gatilhos relevantes adicionais

Segue-se uma listagem dos gatilhos mais importantes que poderiam ser implementados numa perspetiva de manutenção e monitorização, além dos mencionados acima.

- Atualizar o preço dos pedidos sempre que uma compra relacionada com eles for adicionada, seja de bilhetes ou de produtos.
- Verificar se existe stock suficiente antes da compra de um produto.
- Atualizar o valor do stock dos produtos sempre que são comprados.
- Definir o preço efetivo dos pedidos aquando da inserção destes na tabela e não apenas aquando de uma atualização.
- Garantir que os tuplos de LugarOcupado referenciam lugares que pertencem à sala da sessão associada.
- Garantir que o valor dos lugares disponíveis em Sessão é menor ou igual à capacidade da sala associada.
- Verificar que o número de lugares na base de dados de uma determinada sala é igual ao valor de número de lugares dentro da tabela da sala.
- Garantir que todas as sessões para um determinado filme são realizadas após a sua data de estreia.
- Impedir que sejam adicionados bilhetes com lugares repetidos para a mesma sessão.
- Averiguar se o posto de venda de um funcionário pertence ao cinema ao qual este está associado.