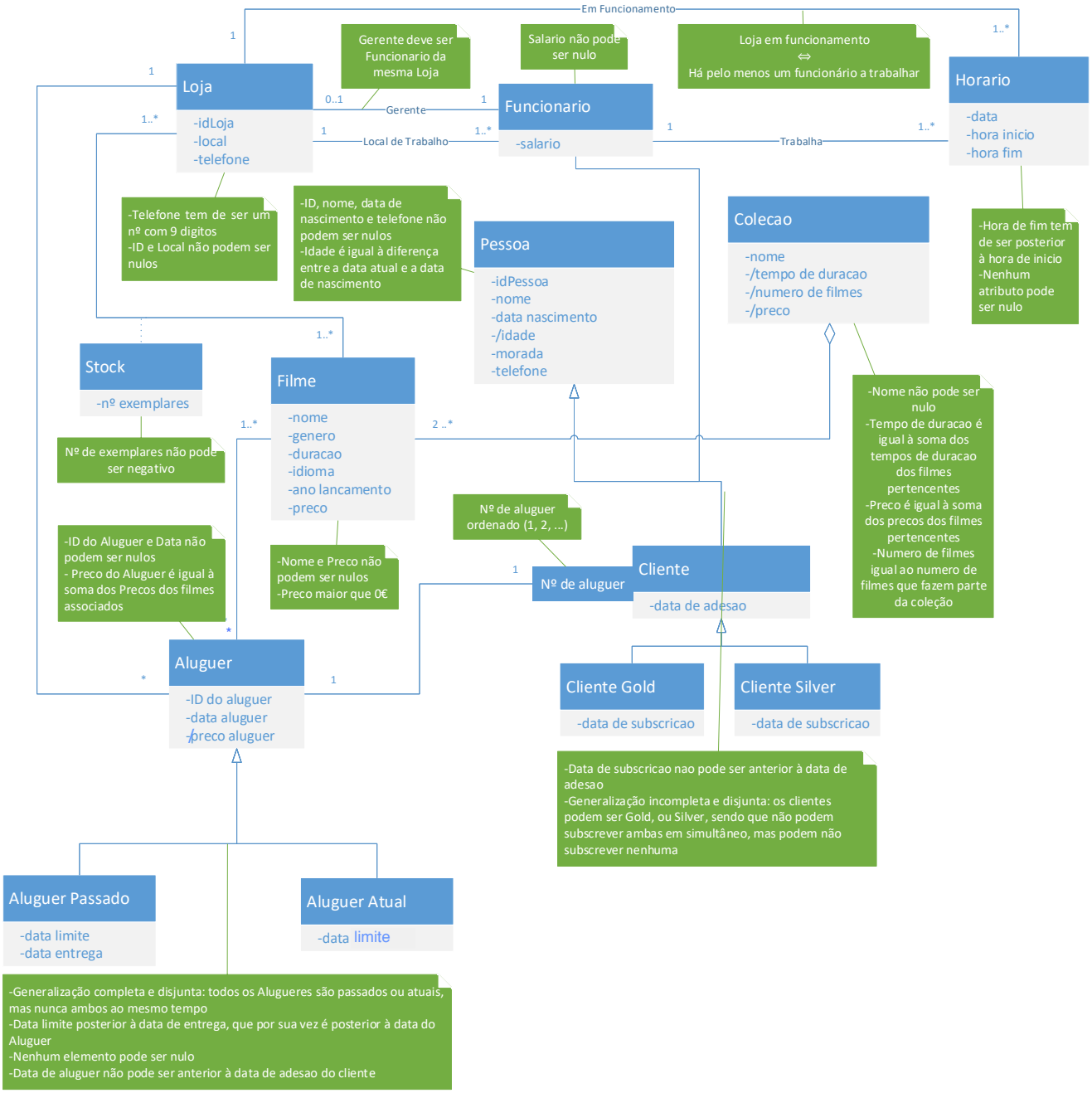
Entrega nº 1 (Diagrama UML e Contexto) - Grupo 711 - BDAD 18/19

Cadeia de lojas de aluguer de filmes

Diagrama UML



Contexto da Base de Dados

Introdução

A fim de se obter mais relações no modelo relacional da nossa base de dados, e também de modo a serem utilizadas várias técnicas de representação UML, como por exemplo a generalização e a agregação, foram acrescentadas algumas classes que não foram referidas na descrição inicial do tema do trabalho, e que irão ser cobertas neste relatório. Houve também alguns elementos que decidimos retirar da base de dados.

Loja

Primeiramente, podemos observar a classe Loja, que representa cada uma das lojas da cadeia. Cada loja está presente num determinado local e possuí um número de telefone e um ID específico e único, que a identifica. Tem duas associações com a classe Funcionário, uma que indica se um funcionário trabalha nessa loja, e outra indicando qual é o funcionário que é gerente de uma determinada loja. É de salientar que: um funcionário trabalha apenas numa loja; uma loja tem um e um só gerente; e esse gerente é funcionário dessa loja.

Funcionário

Relativamente à classe Funcionário, vemos que ela, assim como a classe Cliente, são subclasses da classe Pessoa. Tal foi feito de modo a evitar a repetição de atributos que iriam ser comuns a funcionários e clientes, tais como o nome, data de nascimento, telefone, etc. Esta classe representa, como o nome indica, cada funcionário/trabalhador de cada loja.

Horário

Achamos relevante adicionar uma classe Horário, que retrata os dias e as horas em que cada loja está aberta, bem como o horário de trabalho de cada funcionário. Há que existir uma conciliação entre os horários dos funcionários de uma loja e essa mesma loja, de modo a que sempre que a loja esteja aberta tenha pelo menos um funcionário a trabalhar, e também de modo a que os funcionários não “trabalhem” enquanto a loja estiver fechada. Isto é indicado como uma restrição, no gráfico UML.

Cliente

De modo a introduzir mais classes/relações à base de dados, considerou-se que um cliente poderá aderir a um de dois planos “premium” da cadeia de lojas: o plano Silver e o plano Gold (pode haver clientes normais!). Cada um destes planos tem certos benefícios; os clientes Gold e Silver são indicados como subclasses da classe Cliente, em

que cada uma tem como atributo a data de início de subscrição do plano. É também guardado para cada cliente a data a que este aderiu aos serviços da cadeia de lojas, passando a ser registado na base de dados. Cada cliente tem associado a si a sua lista de alugueres (passados e atuais), que estão “ordenados” por um qualificador na associação de Cliente a Aluguer: o nº de aluguer refere-se ao n-ésimo aluguer que esse cliente fez (ex: aluguer nº 1, aluguer nº 2, etc).

Filme

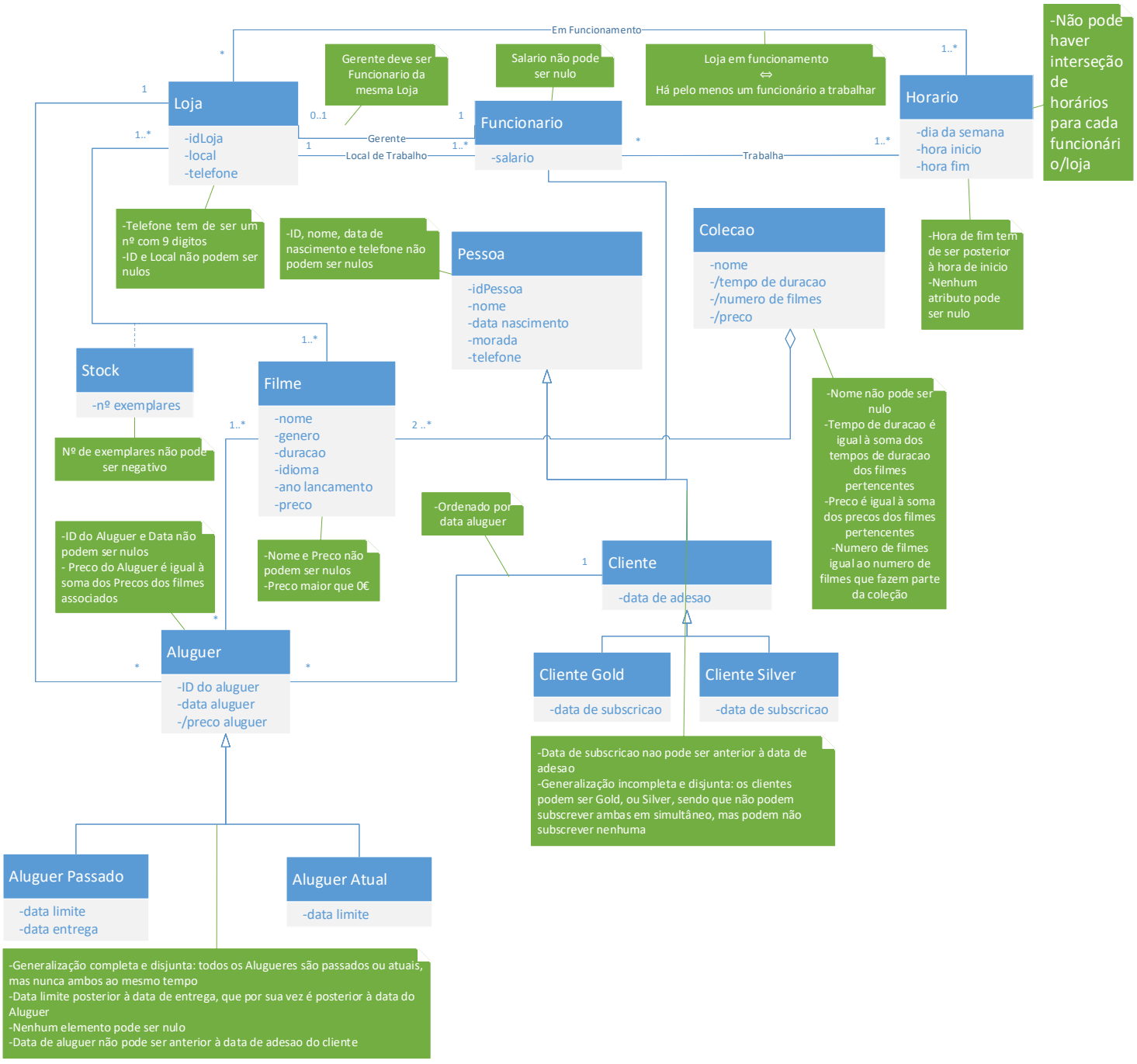
A classe Filme está ligada à classe Loja, de modo a registar o nº de exemplares de cada filme que cada loja possuí (pode não possuir nenhum). Em cada aluguer são registados todos os filmes que fazem parte dele. Alguns filmes encontram-se agregados em coleções, como retrata a classe Coleção, que possui o nome da mesma, a soma da duração e do preço de todos os filmes, e o nº de filmes que a constitui. Em relação a esta última classe, embora não acrescente muito em termos de informação, achamos que a sua adição seria relevante no contexto do tema.

Aluguer

Finalmente, temos a classe Aluguer, ligada a Cliente, Filme, e Loja. Cada cliente pode efetuar um aluguer de vários filmes ao mesmo tempo, sendo que a loja onde se deu esse aluguer fica também registada. Como já foi dito, decidimos utilizar uma associação qualificada entre Aluguer e Cliente de modo a “ordenar” os alugueres feitos por um dado cliente. Para cada aluguer é registada a data em que ocorreu, assim como o seu preço (soma do preço dos filmes), e um ID específico e único. Os alugueres estão divididos em alugueres passados, em que os filmes foram entregues antes da, ou na data limite, e em alugueres atuais, em que os filmes ainda não foram devolvidos.

Entrega nº 2 - Grupo 711 - BDAD 18/19

Diagrama UML revisto/modificado



NOTA: para além das modificações sugeridas pelo professor, de modo a tornar o esquema relacional melhor e menos complicado, decidimos fazer mais algumas alterações ao diagrama UML, como pode observar na imagem.

Por exemplo, nos horários, agora consideramos que há apenas uma instância de horário diferente para cada dia da semana, hora de entrada e hora de saída (como é explicado mais à frente nas restrições). Qualquer loja/funcionário que tenha esse horário deve estar associado a ele (daí a mudança das multiplicidades das associações, pois agora um horário pode estar ligado a vários funcionários/lojas).

O resto das mudanças foram pequenas, e não necessitam de explicação.

Esquema relacional

- Loja (idLoja, local, telefone, idGerente —> Funcionário)
- Pessoa (idPessoa, nome, dataNascimento, morada, telefone)
- Horário (idHorário, diaSemana, horaInicio, horaFim)
- HorárioFunc (idFuncionário —> Funcionário, idHorário —> Horário)
- HorárioLoja (idLoja —> Loja, idHorário —> Horário)
- Filme (idFilme, nome, género, duração, idioma, anoLançamento, preço, idColeção —> Coleção)
- Stock (idLoja —> Loja, idFilme —> Filme, numExemplares)
- Coleção (idColeção, nome, tempoDuração, numFilmes, preço)
- Aluguer (idAluguer, dataAluguer, dataLimite, dataEntrega, preçoAluguer, idLoja —> Loja, idCliente —> Cliente)
- AlugFilme (idAluguer —> Aluguer, idFilme —> Filme)
- Funcionário (idPessoa —> Pessoa, salário, dataContratação, idLoja —> Loja)
- Cliente (idPessoa —> Pessoa, dataAdesão)
- ClienteGold (idCliente —> Cliente, dataSubscrição)
- ClienteSilver (idCliente —> Cliente, dataSubscrição)

NOTA: em Aluguer, se dataEntrega for NULL, então é porque o aluguer é atual; caso contrário é aluguer passado.

Análise de Dependências Funcionais e Formas Normais

- Loja (idLoja, local, telefone, idGerente —> Funcionário)

  FDs: idLoja —> local, telefone, idGerente
  idGerente —> idLoja, local, telefone
  local —> idLoja, telefone, idGerente
  telefone —> idLoja, local, idGerente

  Formas: BCNF: sim
  3NF: sim

- Pessoa (idPessoa, nome, dataNascimento, morada, telefone)

  FDs: idPessoa —> nome, dataNascimento, morada, telefone

  Formas: BCNF: sim
  3NF: sim

- Horário (idHorário, diaSemana, horaInicio, horaFim)

  FDs: idHorário —> diaSemana, horaInicio, horaFim
  diaSemana, horaInicio, horaFim —> idHorário

  Formas: BCNF: sim
  3NF: sim

- HorárioFunc (idFuncionário —> Funcionário, idHorário —> Horário)

  FDs: —

  Formas: BCNF: sim
  3NF: sim

- HorárioLoja (idLoja —> Loja, idHorário —> Horário)

  FDs: —

  Formas: BCNF: sim
  3NF: sim

- Filme (idFilme, nome, género, duração, idioma, anoLançamento, preço, idColeção —> Coleção)

  FDs: idFilme —> nome, género, duração, idioma, anoLançamento, preço, idColeção

  Formas: BCNF: sim
  3NF: sim

- Stock (idLoja —> Loja, idFilme —> Filme, numExemplares)

  FDs: idLoja, idFilme —> numExemplares

  Formas: BCNF: sim
  3NF: sim

- Coleção (idColeção, nome, tempoDuração, numFilmes, preço)

  FDs: idColeção —> nome, tempoDuração, numFilmes, preço

  Formas: BCNF: sim
  3NF: sim

- Aluguer (idAluguer, dataAluguer, dataLimite, dataEntrega, preçoAluguer, idLoja —> Loja, idCliente —> Cliente)

  FDs: idAluguer —> dataAluguer, dataLimite, dataEntrega, preçoAluguer, idLoja, idCliente

  Formas: BCNF: sim
  3NF: sim

- AlugFilme (idAluguer —> Aluguer, idFilme —> Filme)

  FDs: —

  Formas: BCNF: sim
  3NF: sim

- Funcionário (idPessoa —> Pessoa, salário, dataContratação, idLoja —> Loja)

  FDs: idPessoa —> salário, dataContratação, idLoja

  Formas: BCNF: sim
  3NF: sim

- Cliente (idPessoa —> Pessoa, dataAdesão)

    FDs: idPessoa —> dataAdesão

    Formas: BCNF: sim
    3NF: sim

- ClienteGold (idCliente —> Cliente, dataSubscrição)

    FDs: idCliente —> dataSubscrição

    Formas: BCNF: sim
    3NF: sim

- ClienteSilver (idCliente —> Cliente, dataSubscrição)

    FDs: idCliente —> dataSubscrição

    Formas: BCNF: sim
    3NF: sim

Como se pode observar, todas as relações da base de dados seguem tanto a Forma Normal de Boyce-Codd (BCNF) como a 3ª Forma Normal. Analisando as definições de cada uma destas formas, concluímos que:

Uma relação está em BCNF se, para todo A —> B não trivial, A é uma superkey/key;

Uma relação está em 3NF se, para todo A —> B não trivial, A é uma superkey/key OU B consiste apenas em atributos primos (atributos que são membros de pelo menos uma chave da relação).

Para todas as relações e todas as FDs da base de dados, a partir da parte esquerda de cada FD (conjunto de atributos A) conseguimos conhecer todos os atributos da relação, o que implica que A é (super)key. Assim, justifica-se o facto de se afirmar que todas as relações estão em BCNF e 3NF.

Restrições

Loja

- Não pode haver duas lojas com o mesmo ID
  - idLoja PRIMARY KEY

- Não podem existir nem locais nem telefones nem gerentes repetidos, nas várias lojas
  - local UNIQUE
  - telefone UNIQUE
  - idGerente UNIQUE

- Todas as lojas devem ter um local e um gerente atribuído a elas
  - local NOT NULL
  - idGerente NOT NULL

- O ID do gerente da loja deverá corresponder ao ID de um funcionário
  - idGerente REFERENCES Funcionário(idPessoa)

Pessoa

- Não pode haver duas pessoas com o mesmo ID
  - idPessoa PRIMARY KEY

- Todas as pessoas devem ter um nome, data de nascimento e telefone
  - nome NOT NULL
  - dataNascimento NOT NULL
  - telefone NOT NULL

Horário

- Não pode haver dois horários com o mesmo ID
  - idHorário PRIMARY KEY

- Deve existir apenas um horário para cada combinação diferente de dia da semana, hora de início e hora de fim
  - UNIQUE(diaSemana, horaInício, horaFim)

- Nenhum dos atributos pode ser nulo
  - diaSemana NOT NULL
  - horaInicio NOT NULL
  - horaFim NOT NULL

- O dia da semana deve ser um dia da semana válido
  - diaSemana 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’)

- A hora de fim tem de ser posterior à hora de início
  - CHECK (horaFim > horaInicio)

<u>Filme</u>

- Não pode haver dois filmes com o mesmo ID
  - idFilme PRIMARY KEY

- Cada filme deve ter obrigatoriamente um nome e um preço
  - nome NOT NULL
  - preço NOT NULL

- O preço tem de ser maior que 0
  - preço CHECK (preço > 0)

- O ID da colecção deve corresponder a um ID de uma instância da tabela Colecção
  - idColeção REFERENCES Coleção (idColeção)

- O género do filme deve ser: ação, comédia, romance, terror, documentário
  - género CHECK (género = ‘ACAO’ OR
    género = ‘COMEDIA’ OR
    género = ‘ROMANCE’ OR
    género = ‘TERROR’ OR
    género = ‘DOCUMENTARIO’)

- O idioma do filme deve ser: português, francês, inglês, espanhol ou japonês
  - idioma CHECK (idioma = ‘PT’ OR
    idioma = ‘FR’ OR
    idioma = ‘EN’ OR
    idioma = ‘ES’ OR
    idioma = ‘JP’)

<u>Stock</u>

- O ID da loja e do filme devem corresponder a um ID de uma instância das tabelas Loja e Filme, respetivamente
  - idLoja REFERENCES Loja (idLoja)
  - idFilme REFERENCES Filme (idFilme)

- Não deve haver duas instâncias com o mesmo par (idLoja, idFilme)
  - PRIMARY KEY (idLoja, idFilme)

- O número de exemplares deve ser maior do que 0
  - numExemplares CHECK (numExemplares > 0)

Coleção

- Não deve haver duas coleções com o mesmo ID
  - idColeção PRIMARY KEY

- Todas as coleções devem ter um nome
  - nome NOT NULL

HorárioFunc

- Não deve haver duas instâncias com o mesmo par (idFuncionário, idHorário)
  - PRIMARY KEY (idFuncionário, idHorário)

- O ID de funcionário deve corresponder a um ID da tabela Funcionário; a mesma coisa para o ID do horário, mas para a tabela Horário
  - idFuncionário REFERENCES Funcionário (idFuncionário)
  - idHorário REFERENCES Horário (idHorário)

HorárioLoja

- Não deve haver duas instâncias com o mesmo par (idLoja, idHorário)
  - PRIMARY KEY (idLoja, idHorário)

- O ID de loja deve corresponder a um ID da tabela Loja; a mesma coisa para o ID do horário, mas para a tabela Horário
  - idLoja REFERENCES Loja (idLoja)
  - idHorário REFERENCES Horário (idHorário)

Aluguer

- Não pode haver dois alugueres com o mesmo ID
  - idAluguer PRIMARY KEY

- Todos os alugueres deverão ter a data em que este se deu, e a data limite
  - dataAluguer NOT NULL
  - dataLimite NOT NULL

- O ID de loja deve corresponder a um ID da tabela Loja; a mesma coisa para o ID do cliente, mas para a tabela Cliente
  - idLoja REFERENCES Loja (idLoja)
  - idCliente REFERENCES Cliente (idCliente)

- A data limite tem de ser posterior ou igual à data de entrega, e esta tem de ser posterior ou igual à data do aluguer
  - CHECK ((dataLimite >= dataEntrega AND dataEntrega >= dataAluguer AND dataLimite >= dataAluguer) OR (dataLimite >= dataAluguer AND dataEntrega IS NULL))

AlugFilme

- Não deve haver duas instâncias com o mesmo par (idAluguer, idFilme)
  - PRIMARY KEY (idAluguer, idFilme)

- O ID de aluguer deve corresponder a um ID da tabela Aluguer; a mesma coisa para o ID do filme, mas para a tabela Filme
  - idAluguer REFERENCES Aluguer (idAluguer)
  - idFilme REFERENCES Filme (idFilme)

Funcionário

- Não pode haver dois funcionários com o mesmo ID; este ID deve corresponder a um ID de uma pessoa na tabela Pessoa
  - PRIMARY KEY idPessoa REFERENCES Pessoa (idPessoa)

- Todos os funcionários devem ter um salário associado, bem como a data em que foram contratados
  - salário NOT NULL
  - dataContratação NOT NULL

- O ID da loja deve corresponder ao ID de uma loja da tabela Loja
  - idLoja REFERENCES Loja (idLoja)

Cliente

- Não pode haver dois clientes com o mesmo ID; este ID deve corresponder a um ID de uma pessoa na tabela Pessoa
  - PRIMARY KEY idPessoa REFERENCES Pessoa (idPessoa)

- Todos os clientes devem ter associados a eles a sua data de adesão
  - dataAdesão NOT NULL

ClienteGold

- Não pode haver dois clientes Gold com o mesmo ID; este ID deve corresponder a um ID de um cliente na tabela Cliente
  - PRIMARY KEY idCliente REFERENCES Cliente (idPessoa)

- Todos os clientes Gold devem ter associados a eles a sua data de subscrição ao plano
  - dataSubscrição NOT NULL

ClienteSilver

- Não pode haver dois clientes Silver com o mesmo ID; este ID deve corresponder a um ID de um cliente na tabela Cliente
  - PRIMARY KEY idCliente REFERENCES Cliente (idPessoa)

- Todos os clientes Silver devem ter associados a eles a data de subscrição ao plano
  - dataSubscrição NOT NULL

NOTA: as outras restrições que foram anunciadas anteriormente e que não constam nesta parte do relatório serão implementadas com triggers, na 3ª entrega, devido à impossibilidade de o fazer sem o uso deles. Por exemplo, restrições que comparam atributos de diferentes relações, etc.

Para além disso, na base de dados em SQL foram utilizadas restrições do género ON UPATE e ON DELETE, para manter a integridade dos dados armazenados.