

Empresa de transporte ferroviário

Projeto BDAD - Entrega 2

Turma 3, Grupo 5

João Afonso Martins Domingues Andrade (up201905589)

Rodrigo Tuna de Andrade (up201904967)

Sérgio Miguel Rosa Estêvão (up201905680)

Abril 2021

Conteúdo

1	Contexto	3
1.1	Introdução	3
1.2	Descrição das Relações	3
2	Diagrama UML	4
3	Diagrama UML Revisto	5
4	Conversão para Esquema Relacional	6
4.1	Esquema Relacional	6
4.2	Considerações	6
5	Análise de Dependências Funcionais e Formas Normais	6
5.1	Listagem	6
5.2	Justificação e Conclusões	8
6	Lista de Restrições	10
7	Lista de Interrogações	18
8	Lista dos Triggers	18

1 Contexto

1.1 Introdução

No nosso trabalho iremos procurar desenvolver uma base de dados de uma empresa de transporte ferroviário. Foram introduzidos novos atributos em relação à descrição submetida de forma a utilizar diferentes conceitos de UML.

1.2 Descrição das Relações

Deseja-se armazenar um conjunto de **rotas**, que possuem um *título*. Cada **rota** passa por determinadas **estações**, pelo menos 2, que são caracterizadas pela sua *morada*, *código postal* e *localidade*, pretendendo-se guardar o tempo passado até que a **estação** é atingida durante determinada **rota**.

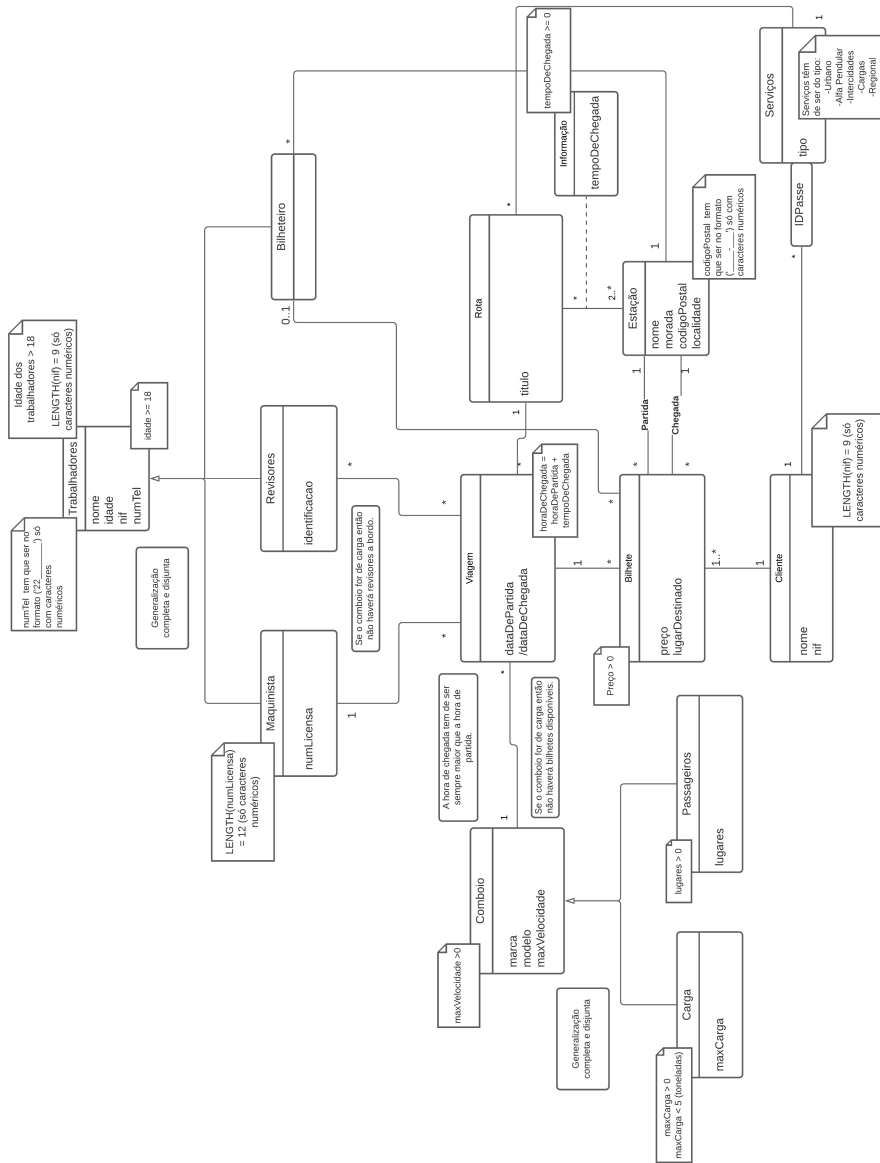
Pretende-se guardar informação acerca das **viagens** realizadas pela empresa, armazenando a sua *data*, *hora de partida* e *hora de chegada*, sendo que cada **viagem** realiza determinada **rota**.

As **viagens** são realizadas por um **comboio** do qual se pretende guardar a *marca*, o *modelo* e a *velocidade máxima*, dividindo-se em **comboios de carga**, onde se pretende guardar a sua *capacidade de carga*, e **comboios de passageiros**, armazenando o *número de lugares*.

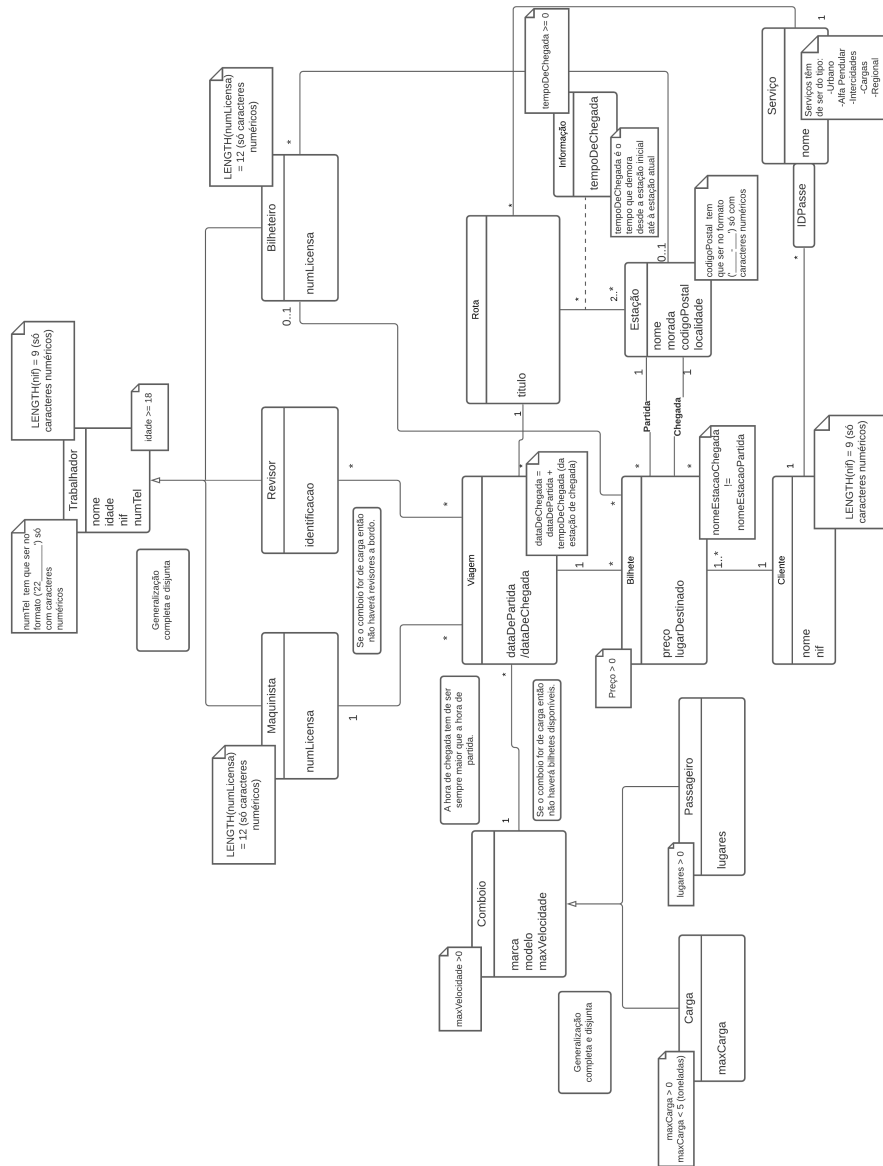
Existem também **serviços**, caracterizados pelo *nome*, dos quais fazem parte diversas **rotas**.

Ambiciona-se guardar informação relativa aos **trabalhadores**, dos quais se deve reter o *nome*, a *idade* e o *nº de telefone*. Os trabalhadores podem ser **maquinistas**, registando-se o *número de licença*, ou **revisores**, registando-se o *número de identificação*; estando ambos presentes em **viagens** de **comboios de passageiros** e só os **maquinistas** nas realizadas por **comboios de carga**. Os **trabalhadores** dividem-se ainda em **bilheteiros** que trabalham numa **estação**.

Um **cliente**, caracterizado por um *nome* e um *nif*, pode adquirir um **bilhete**, sendo que interessa guardar a informação relativa ao *preço* que pagou, o *lugar ocupado durante a viagem*, de que **estação** partiu e a que **estação** chegou, assim como que **bilheteiro** vendeu esse **bilhete**. Um **cliente** pode ainda adquirir um passe que lhe permitirá viajar em qualquer **rota** de um certo **serviço**, guardando o *id do passe*.



3 Diagrama UML Revisto



4 Conversão para Esquema Relacional

4.1 Esquema Relacional

- Cliente(nif, nome)
- Serviço(tipo)
- Assinatura(nif → Cliente, nomeServiço → Serviço, IDPasse)
- Estação(nome, morada, codigoPostal, localidade)
- Rota(id, título, nomeServiço → Serviço)
- Informação(nomeEstação → Estação, idRota → Rota, tempoDeChegada)
- Maquinista(nif , nome, idade, numTel, numLicensa)
- Revisor(nif , nome, idade, numTel, identificacao)
- Bilheteiro(nif , nome, idade, numTel, numLicensa, nomeEstacao → Estação)
- Comboio(marca, modelo, maxVelocidade, maxCarga, lugares);
- Viagem(id, dataDePartida, dataDeChegada, idComboioCarga→ComboioCarga, idComboioPassageiros→ComboioPassageiros, idRota→ Rota, nifMaquinista→Maquinista)
- RevisorViagem(idViagem→Viagem, idRevisor→Revisor)
- Bilhete(id, preço, lugarDestinado, nifCliente→Cliente, nomeEstaçãoPartida→Estação, nomeEstaçãoChegada→Estação, nifBilheteiro→Bilheteiro, idViagem→Viagem)

4.2 Considerações

O esquema relacional foi obtido aplicando as regras de conversão; para as generalizações, inicialmente foi utilizada a estratégia *Object Oriented*, dado as classes são disjuntas e completas, porém acabou por ser aplicada a estratégia *Use Nulls*, visto que a superclasse "Comboio" têm associações a outras classes e o uso desta estratégia tornava o esquema relacional mais conciso.

5 Análise de Dependências Funcionais e Formas Normais

5.1 Listagem

- Cliente
 - FD's: nif→nome
 - BCNF: Sim
 - 3NF: Sim

- Serviço
 - FD's: -
 - BCNF: Sim
 - 3NF: Sim
- Assinatura
 - FD's: $nif, nomeServiço \rightarrow IDPasse$
 $nomeServiço, IDPasse \rightarrow nif$
 - BCNF: Sim
 - 3NF: Sim
- Estação
 - FD's: $nome \rightarrow morada, codigoPostal, localidade$
 $morada, codigoPostal \rightarrow nome, localidade$
 - BCNF: Sim
 - 3NF: Sim
- Rota
 - FD's: $id \rightarrow titulo, nomeServiço$
 - BCNF: Sim
 - 3NF: Sim
- Informação
 - FD's: $nomeEstação, idRota \rightarrow tempoDeChegada$
 $idRota, tempoDeChegada \rightarrow nomeEstação$
 - BCNF: Sim
 - 3NF: Sim
- Maquinista
 - FD's: $nif \rightarrow nome, idade, numTel, numLicensa$
 $numLicensa \rightarrow nome, idade, numTel, nif$
 - BCNF: Sim
 - 3NF: Sim
- Revisor
 - FD's: $nif \rightarrow nome, idade, numTel, identificação$
 $identificação \rightarrow nome, idade, numTel, nif$
 - BCNF: Sim
 - 3NF: Sim

- Bilheteiro
 - FD's: $nif \rightarrow nome, idade, numTel, numLicensa, nomeEstacao$
 - BCNF: Sim
 - 3NF: Sim
- Comboio
 - FD's: $id \rightarrow marca, modelo, maxVelocidade, maxCarga, lugares$
 $marca, modelo \rightarrow maxVelocidade, maxCarga, lugares$
 - BCNF: Não
 - 3NF: Não
- Viagem
 - FD's: $id \rightarrow dataDePartida, dataDeChegada, idComboio, idRota, nifMaquinista$
 $dataDePartida, idRota \rightarrow id, dataDeChegada, idComboio, nifMaquinista$
 - BCNF: sim
 - 3NF: Sim
- RevisorViagem
 - FD's: -
 - BCNF: Sim
 - 3NF: Sim
- Bilhete
 - FD's: $id \rightarrow preco, lugarDestinado, nifCliente, nomeEstaçãoPartida, nomeEstaçãoChegada, nifBilheteiro, idViagem$
 $idViagem, nomeEstaçãoChegada, nomeEstaçãoPartida \rightarrow preco$
 - BCNF: Não
 - 3NF: Não

5.2 Justificação e Conclusões

Nas relações **Cliente**, **Serviço**, **Rota**, **Bilheteiro**, **RevisorViagem** ou não existe nenhuma FD ou existe apenas uma que faz corresponder a chave primária aos restantes atributos, pelo que trivialmente estão na BCNF, logo também estão na 3NF, visto que a BCNF é uma extensão da 3NF.

Nas relações **Assinatura**, **Estação**, **Informação**, **Maquinista**, **Revisor**, **Assinatura** existe mais uma FD para além da chave primária, no entanto os atributos do lado esquerdo da FD formam eles próprios uma super chave pelo que as relações estão na forma BCNF, logo também estão na 3NF.

Nas relações **Comboio** e **Bilhete** existe uma FD que viola a BCNF pois os atributos do lado esquerdo não formam uma super chave e violam a 3NF pois os atributos do lado direito não fazem todos parte de uma chave canidata. Pelo que teremos de realizar as seguintes decomposições:

- Comboio: Comboio(id, marcaC→ComboioCaracteristicas, modeloC→ComboioCaracteristicas)
ComboioCaracteristicas(marca, modelo, maxVelocidade, maxCarga, lugares)
- Bilhete : Bilhete(id, lugarDestinado, nifCliente→Cliente, nomeEstaçãoPartida→Estação, nomeEstaçãoChegada→Estação, nifBilheteiro→Bilheteiro, idViagem→Viagem)
BilhetePreço(nomeEstaçãoPartida→Estação, nomeEstaçãoChegada→Estação, idViagem→Viagem, preço)

6 Lista de Restrições

- Cliente

- Não podem haver duas instâncias de Cliente com o mesmo nif.

```
1 nif PRIMARY KEY
```

- O atributo nif deve ser uma string de 9 caracteres numéricos

```
1 CHECK(LENGTH(nif) == 9 AND nif NOT GLOB '*[^\0-9]*')
```

- O atributo nif tem que estar definido.

```
1 nif NOT NULL
```

- Serviço

- Não podem haver duas instâncias de Servico com o mesmo nome.

```
1 nome PRIMARY KEY
```

- O atributo nome tem de ser um dos indicados na restrição.

```
1 CHECK(( nome == 'Alfa-Pendular' OR
2         nome == 'Intercidades' OR
3         nome == 'Regional' OR
4         nome == 'Urbano' OR
5         nome == 'Cargas' ))
```

- O atributo nome tem que estar definido.

```
1 nome NOT NULL
```

- Assinatura

- Não podem haver duas instâncias de Assinatura com o mesmo nif e nomeServico.

```
1 PRIMARY KEY(nif, nomeServico)
```

- Os atributos nif e nomeServico devem corresponder ao nif do cliente e nome do serviço, respetivamente. Se forem apagadas instâncias das tabelas referenciadas (Cliente e Servico), então devem ser também apagadas as instâncias da tabela Assinatura correspondentes.

```
1 nif REFERENCES Cliente ON DELETE CASCADE ON UPDATE CASCADE
2 nomeServico REFERENCES Servico ON DELETE CASCADE ON UPDATE CASCADE
```

- Os atributos nif, nomeServiço e IDPasse têm que estar definidos.

```
1 nif NOT NULL
2 nomeServico NOT NULL
3 IDPasse NOT NULL
```

- IDPasse e NomeServico são atributos únicos.

```
1 UNIQUE(nomeServico, IDPasse)
```

- Estação

- Não podem haver duas instâncias de Estacao com o mesmo nome.

```
1 nome PRIMARY KEY
```

- Os atributos nome, morada, codigoPostal e localidade têm que estar definidos.

```
1 nome NOT NULL
2 morada NOT NULL
3 codigoPostal NOT NULL
4 localidade NOT NULL
```

- O atributo codigoPostal deve seguir o formato ('----') e deve ter apenas caracteres numéricos.

```
1 CHECK(codigoPostal GLOB '[0-9][0-9][0-9][0-9] - [0-9][0-9][0-9]')
```

- Rota

- Não podem haver duas instâncias de Rota com o mesmo id.

```
1 id PRIMARY KEY
```

- O atributo nomeServico deve corresponder ao nome do serviço. Se forem apagadas instâncias das tabelas referenciadas (Servico), então devem ser também apagadas as instâncias da tabela Rota correspondentes.

```
1 nomeServico REFERENCES Servico ON DELETE CASCADE ON UPDATE CASCADE
```

- Os atributos id e nomeServico têm que estar definidos.

```
1 id NOT NULL
2 nomeServico NOT NULL
```

- Informação

- Não podem haver duas instâncias da tabela Informacao com o mesmo nomeEstacao e idRota.

```
1 PRIMARY KEY(nomeEstacao, idRota)
```

- Os atributos nomeEstacao e idRota devem corresponder ao nome da estação e id da rota, respetivamente. Se forem apagadas instâncias das tabelas referenciadas (Estacao e Rota), então devem ser também apagadas as instâncias da tabela Informacao correspondentes.

```
1 nomeEstacao REFERENCES Estacao ON DELETE CASCADE ON UPDATE
  CASCADE
2 idRota REFERENCES Rota ON DELETE CASCADE ON UPDATE CASCADE
```

- Os atributos nomeEstacao, idRota e tempoDeChegada têm que estar definidos.

```
1 nomeEstacao NOT NULL
2 idRota NOT NULL
3 tempoDeChegada NOT NULL
```

- O atributo tempoDeChegada tem de ser maior ou igual a zero.

```
1 CHECK(tempoDeChegada >= 0)
```

- idRota e tempoDeChegada são atributos únicos.

```
1 UNIQUE(idRota, tempoDeChegada)
```

- Os atributos nomeEstacao e idRota devem corresponder ao nome da estação e id da rota, respetivamente. Se forem apagadas instâncias das tabelas referenciadas (Estacao e Rota), então devem ser também apagadas as instâncias da tabela Informacao correspondentes.

```
1 nomeEstacao REFERENCES Estacao ON DELETE CASCADE ON UPDATE
  CASCADE
2 idRota REFERENCES Rota ON DELETE CASCADE ON UPDATE CASCADE
```

- Maquinista

- Não podem haver duas instâncias de Maquinista com o mesmo nif.

```
1 nif PRIMARY KEY
```

- Os atributos nif, idade e numLicensa têm que estar definidos.

```
1 nif NOT NULL
2 idade NOT NULL
3 numLicensa NOT NULL
```

- O atributo nif deve ser uma string de 9 caracteres numéricos.

```
1 CHECK(LENGTH(nif) == 9 AND nif NOT GLOB '*[^0-9]*')
```

- O maquinista deve ter uma idade maior ou igual a 18.

```
1 CHECK(idade >= 18)
```

- O atributo numTelefone deve ser uma string de 9 caracteres numéricos começando por 22.

```
1 CHECK(numTelefone LIKE '22_____' AND numTelefone NOT
  GLOB '*[^0-9]*')
```

- O atributo numLicensa deve ser uma string de 12 caracteres numéricos.

```
1 CHECK(LENGTH(numLicensa) == 12 AND numLicensa NOT GLOB '
  *[^0-9]*')
```

- numLicença é um atributo único.

```
1 UNIQUE
```

- Bilheteiro

- Não podem haver duas instâncias de Bilheteiro com o mesmo nif.

```
1 nif PRIMARY KEY
```

- O atributo estacao deve corresponder ao nome da estação na relação Estacao. Se forem apagadas instâncias da tabela referenciadas (Estacao), então devem ser colocadas a NULL todas as instâncias da tabela Bilheteiro correspondentes.

```
1 nomeEstacao REFERENCES Estacao ON DELETE SET NULL ON  
UPDATE CASCADE
```

- Os atributos nif, idade e nomeEstacao têm que estar definidos.

```
1 nif NOT NULL  
2 idade NOT NULL  
3 nomeEstacao NOT NULL
```

- O bilheteiro deve ter uma idade maior ou igual a 18.

```
1 CHECK(idade >= 18)
```

- O atributo numTelefone deve ser uma string de 9 caracteres numéricos começando por 22.

```
1 CHECK(numTelefone LIKE '22_____' AND numTelefone NOT  
GLOB '*[0-9]*')
```

- O atributo nif deve ser uma string de 9 caracteres numéricos.

```
1 CHECK(LENGTH(nif) == 9 AND nif NOT GLOB '*[0-9]*')
```

- Revisor

- Não podem haver duas instâncias da relação Revisor com o mesmo nif.

```
1 nif PRIMARY KEY
```

- Os atributos nif, idade e identificacao têm que estar definidos.

```
1 nif NOT NULL  
2 idade NOT NULL  
3 identificacao NOT NULL
```

- O revisor deve ter uma idade maior ou igual a 18.

```
1 CHECK(idade >= 18)
```

- O atributo numTelefone deve ser uma string de 9 caracteres numéricos começando por 22.

```
1 CHECK(numTelefone LIKE '22_____' AND numTelefone NOT  
   GLOB '*[0-9]*')
```

- Os atributos nif e identificacao devem ser uma string de 9 caracteres numéricos.

```
1 CHECK(LENGTH(nif) == 9 AND nif NOT GLOB '*[0-9]*')  
2 CHECK(LENGTH(identificacao) == 9 AND identificacao NOT  
   GLOB '*[0-9]*')
```

- Comboio

- Não podem haver duas instâncias de Comboio com o mesmo id.

```
1 id PRIMARY KEY
```

- Os atributos marcaC e modeloC devem corresponder à marca e modelo do Comboio, respetivamente. Se forem apagadas instâncias da tabela ComboioCaracteristicas, então devem ser também apagadas as instâncias da tabela Comboio correspondentes.

```
1 FOREIGN KEY(marcaC, modeloC) REFERENCES  
   ComboioCaracteristicas(marca, modelo) ON DELETE  
   CASCADE ON UPDATE CASCADE
```

- Os atributos id, marca e modelo têm que estar definidos.

```
1 id NOT NULL  
2 marca NOT NULL  
3 modelo NOT NULL
```

- ComboioCaracteristicas

- Não podem haver duas instâncias de ComboioCaracteristicas com a mesma marca e modelo.

```
1 PRIMARY KEY(marca, modelo)
```

- Os atributos marca, modelo e velMaxima têm que estar definidos.

```
1 marca NOT NULL  
2 modelo NOT NULL  
3 velMaxima NOT NULL
```

- O atributo velMaxima deve ser maior do que 0.

```
1 CHECK (velMaxima > 0)
```

- O atributo maxCarga deve ser maior do que 0 e inferior a 5 toneladas. Senão, deve ser igual a NULL.

```
1 CHECK ((maxCarga > 0 AND maxCarga < 5) OR maxCarga IS NULL)
```

- O atributo lugares deve ser maior do que 0. Senão, deve ser igual a NULL.

```
1 CHECK (lugares > 0 OR lugares IS NULL)
```

- Apenas um dos atributos maxCarga ou lugares pode ser diferente de NULL.

```
1 CHECK((maxCarga IS NULL) OR (lugares IS NULL))
```

- Viagem

- Não podem haver duas Viagens com o mesmo id.

```
1 id PRIMARY KEY
```

- Os atributos idComboio, nifMaquinista e idRota devem corresponder ao id de um Comboio, ao nif de um Maquinista e ao id de uma Rota, respetivamente. Se forem apagadas instâncias das tabelas referenciadas Comboio e Maquinista, então devem ser colocadas a NULL todas as instâncias da tabela Viagem correspondentes. Por outro lado, se forem apagadas instâncias da tabela Rota, então devem ser também apagadas as instâncias da tabela Viagem correspondentes.

```
1 idComboio REFERENCES Comboio ON DELETE SET NULL ON UPDATE CASCADE
2 idRota REFERENCES Rota ON DELETE CASCADE UPDATE CASCADE
3 nifMaquinista REFERENCES Maquinista ON DELETE SET NULL ON UPDATE CASCADE
```

- Os atributos id, dataDePartida, dataDeChegada, idRota e idMaquinista têm que estar definidos.

```
1 id NOT NULL
2 dataDePartida NOT NULL
3 dataDeChegada NOT NULL
4 idRota NOT NULL
5 idMaquinista NOT NULL
```

- A combinação dos atributos dataDePartida e idRota é única para cada instância de Viagem.

```
1 UNIQUE(dataDePartida, idRota)
```

- RevisorViagem

- Não podem haver duas instâncias de RevisorViagem com o mesmo nifRevisor e idViagem.

```
1 PRIMARY KEY(nifRevisor, idViagem)
```

- Os atributos nifRevisor e idViagem devem corresponder a um nif de um Revisor e ao id de uma Viagem, respetivamente. Se forem apagadas instâncias das tabelas referenciadas Revisor e Viagem, então devem ser também apagadas as instâncias da tabela RevisorViagem correspondentes.

```
1 nifRevisor REFERENCES Revisor ON DELETE CASCADE ON UPDATE
  CASCADE
2 idViagem REFERENCES Viagem ON DELETE CASCADE ON UPDATE
  CASCADE
```

- Os atributos nifRevisor e idViagem têm que estar definidos.

```
1 nifRevisor NOT NULL
2 idViagem NOT NULL
```

- Bilhete

- Não podem haver dois Bilhetes com o mesmo id.

```
1 id INTEGER PRIMARY KEY
```

- Os atributos nifCliente, idViagem, nifBilheteiro, nomeEstacaoPartida e nomeEstacaoChegada devem corresponder a um nif de um Cliente, um id de uma Viagem, um nif de um Bilheteiro e a nomes de Estacoes, respetivamente. Se forem apagadas instâncias das tabelas referenciadas Cliente, Estacao, Bilheteiro, então devem ser colocadas a NULL todas as instâncias da tabela Bilhete correspondentes. Por outro lado, se forem apagadas instâncias da tabela Viagem, então devem ser também apagadas as instâncias da tabela Bilhete correspondentes.

```
1 nifCliente REFERENCES Cliente ON DELETE SET NULL ON UPDATE
  CASCADE
2 idViagem REFERENCES Viagem ON DELETE CASCADE ON UPDATE
  CASCADE
3 nifBilheteiro REFERENCES Bilheteiro ON DELETE SET NULL ON
  UPDATE CASCADE
4 nomeEstacaoPartida REFERENCES Estacao ON DELETE SET NULL
  ON UPDATE CASCADE
5 nomeEstacaoChegada REFERENCES Estacao ON DELETE SET NULL
  ON UPDATE CASCADE
```

- Os atributos id, nifCliente, nomeEstacaoPartida, nomeEstacaoChegada e idViagem têm que estar definidos.

```
1 id NOT NULL
2 nifCliente NOT NULL
3 nomeEstacaoPartida NOT NULL
4 nomeEstacaoChegada NOT NULL
5 idViagem NOT NULL
```


- Os atributos nomeEstacaoChegada e nomeEstacaoPartida não podem ser idênticos.

```
1 CHECK(nomeEstacaoChegada != nomeEstacaoPartida)
```

- BilhetePreço

- Não podem haver duas instancias de BilhetePreço com o mesmo nomeEstacaoPartida, nomeEstacaoChegada e idViagem.

```
1 PRIMARY KEY(nomeEstacaoPartida, nomeEstacaoChegada, idViagem)
```

- Os atributos idViagem, nomeEstacaoPartida e nomeEstacaoChegados devem corresponder a um id de uma Viagem e a nomes de Estações. Se forem apagadas instâncias da tabela referenciada Estacao, então devem ser colocadas a NULL todas as instâncias da tabela BilhetePreço correspondentes. Por outro lado, se forem apagadas instâncias da tabela Viagem, então devem ser também apagadas as instâncias da tabela BilhetePreço correspondentes.

```
1 idViagem REFERENCES Viagem ON DELETE CASCADE ON UPDATE CASCADE
2 nomeEstacaoPartida REFERENCES Estacao ON DELETE SET NULL ON UPDATE CASCADE
3 nomeEstacaoChegada REFERENCES Estacao ON DELETE SET NULL ON UPDATE CASCADE
```

- Os atributos idViagem, nomeEstacaoPartida, nomeEstacaoChegados e preco têm que estar definidos.

```
1 idViagem NOT NULL
2 nomeEstacaoPartida NOT NULL
3 nomeEstacaoChegados NOT NULL
4 preco NOT NULL
```

- O atributo preco têm que ser positivo.

```
1 CHECK(preco > 0)
```

- Os atributos nomeEstacaoChegada e nomeEstacaoPartida não podem ser idênticos.

```
1 CHECK(nomeEstacaoChegada != nomeEstacaoPartida)
```

7 Lista de Interrogações

1. Todas as viagens que vão de Porto - Lisboa no dia 11/05
2. Tempo da viagem mais longa de cada cliente cuja viagem tenha duração superior a 1 hora, ordenado por ordem decrescente
3. Estações onde passam mais de 3 serviços diferentes
4. Todos os clientes que já tenham passado em todas as estações
5. As cinco estações por onde passaram mais clientes no mês de Maio de 2021.
6. Taxa de ocupação média de cada rota (média da percentagem de lugares ocupados nas viagens de passageiros de cada rota)
7. Todos os clientes que já estiveram em pelo menos uma viagem de 2 ou mais serviços
8. Tabela por ordem, das despesas totais de todos os clientes que não têm assinatura em nenhum serviço (do cliente que gastou mais para o menos)
9. Comboios de passageiros da marca 'Stadler' que têm mais de 100 lugares e que já fizeram pelo menos uma viagem que passou pela estação Coimbra-B
10. Nomes dos clientes que não têm assinatura em nenhum serviço mas que já compraram pelo menos 1 bilhete, tendo o bilheteiro que vendeu o bilhete mais de 30 anos

8 Lista dos Triggers

1. Legalidade de bilhetes:
 - Não é possível haver mais bilhetes de uma viagem do que lugares disponíveis no comboio associado a essa viagem
 - As estações apresentadas no bilhete têm de existir na rota e uma tem de ser a seguir à outra
 - Para haver bilhete então tem de ser uma viagem de passageiros
2. Calcular o tempo de chegada de uma viagem quando esta é inserida
3. Garantir que são atribuídos comboios de carga a viagens de carga e comboios de passageiros a viagem de passageiros.