**

# **Comboios de Portugal**

Bases de Dados

2018-2019

17 de março de 2019

Eduardo Ferreira Campos, [**up201604920**@fe.up.pt](mailto:up201604920@fe.up.pt)

Miguel Rodrigues Gomes, [**up201605908**@fe.up.pt](mailto:up201605908@fe.up.pt)

**Contextualização do tema**

De modo a flexibilizar o sistema de controlo da circulação de comboios, a CP pretende começar a armazenar diversas informações.

Por todo o Portugal, os comboios circulam em linhas definidas. Estas linhas têm um nome e uma cor identificativa, bem como o conjunto de paragens que pertencem àquela linha.

Uma paragem é caracterizada pelo seu nome, a sua localização e também as linhas às quais pertence, visto que uma dada paragem pode pertencer a mais que uma linha.

De forma a simplificar, considera-se para o contexto deste trabalho que os comboios circulam apenas numa linha. Os comboios têm um número de identificação e uma lotação máxima. Está também indicado a que linha pertencem, bem como as viagens que realizam e a sua tripulação, constituída por um maquinista e diversos revisores.

Uma viagem consiste na data em que se realiza bem como a sua hora de início e de fim e a lotação da viagem, isto é, o número de passageiros que circularam naquela viagem. Guarda-se também os passageiros que circulam no comboio.

Uma dada pessoa é caracterizada pelo seu nome, telefone, número de identificação fiscal, género e morada. As pessoas podem ser passageiros, dos quais também se guarda o número do bilhete, ou funcionários. Dos funcionários armazena-se a data de início de trabalho na CP, o seu salário e o seu cargo, podendo este ser “maquinista” ou “revisor”.

Este tema foi baseado num tema proposto para o trabalho realizado na unidade curricular de Algoritmos e Estruturas de Dados do ano letivo de 2018/2019 (tema 3 da lista de temas propostos), tendo sofrido as alterações necessárias de modo a não ficar demasiadamente semelhante e se enquadrar nos objetivos da unidade curricular de Bases de Dados e do projeto em questão.

**Modelo conceptual**

De seguida apresenta-se o diagrama de classes UML para a base de dados:

**Uma imagem com captura de ecrã

Descrição gerada automaticamente**

**Esquema Relacional**

Segue o esquema relacional retirado do UML:

Person(PersonID, name, address, phone\_number, nif, gender)

Passenger(PassengerID->Person, ticket\_number)

Employee(EmployeeID -> Person, hiring\_date, wage)

TrainDriver(TrainDriverID -> Employee, position)

Conductor(ConductorID -> Employee, position)

Trip(TripID, current\_capacity, beginning\_hour, ending\_hour, trip\_date, TrainID->Train)

Train(TrainID, capacity, lineID->Line, trainDriverID->TrainDriver)

Line(LineID, name, color)

Stop(StopID, location)

Inspects(inspectorID -> Conductor, TrainID -> Train)

Belongs(LineID->Line, StopID->Stop)

Travels(PassengerID -> Passenger, TripID -> Trip)

**Dependências funcionais e Formas Normais**

Segue a tabela indicando as Dependências Funcionais e chave primária de cada relação:

|  |  |  |  |
| --- | --- | --- | --- |
| **Número da relação** | **Nome da relação** | **Chave**  **primária** | **Dependências Funcionais (FD)** |
| **#01** | Person | PersonID | PersonID-> name, address, phone\_number, nif, gender |
| **#02** | Passenger | PassengerID | PassengerID->ticket\_number |
| **#03** | Employee | EmployeeID | EmployeeID->hiring\_date, wage |
| **#04** | TrainDriver | TrainDriverID | TrainDriverID->position |
| **#05** | Conductor | ConductorID | ConductorID->position |
| **#06** | Trip | TripID | TripID->current\_capacity, beginning\_hour, ending\_hour, trip\_date, TrainID |
| **#07** | Train | TrainID | TrainID-> capacity, LineID, TrainDriverID |
| **#08** | Line | LineID | LineID-> name, color |
| **#09** | Stop | StopID | StopID-> location |
| **#10** | Inspects | {InspectorID, TrainID} | ---  --- |
| **#11** | Belongs | {LineID, TrainID} | --- |
| **#12** | Travels | {PassengerID, TripID} | ---  --- |

De modo a construir uma base de dados imune às diversas anomalias, i.e. inserção, remoção ou modificação, é necessário proceder à normalização de todas as suas relações. As formas normais mais comuns usadas para este fim são a Terceira Forma Normal (3FN) e a Forma Normal de Boyce-Codd (FNBC).

Consultando a tabela suprajacente, em especial a coluna das dependências funcionais, nota-se que todos os atributos são simples e indivisíveis, cumprindo assim a

definição de atributo atômico, logo pode-se dizer que as relações estão na Primeira Forma Normal. Similarmente não se encontram instâncias de dependências parciais, pelo que a Segunda Forma Normal também é seguida. É importante fazer este esclarecimento pois o primeiro requisito para que relações estejam na Terceira Forma Normal, como pretendido, é que cumpra a Segunda e Primeira Formas Normais.

Portanto, para que uma dada relação esteja na Terceira Forma Normal, para além de cumprir os requisitos das formas precedentes, deve garantir que todos os atributos não-chave são totalmente dependentes de atributos-chave, mantendo independência entre si, algo que, novamente por análise das Dependências Funcionais listadas, é cumprido.

Por último, a FNBC existe como uma restrição da Terceira Forma Normal, pelo que esta deve ser cumprida de modo a que a relação possa ser normalizada em FNBC. As relações encontram-se na FNBC pois todos os atributos dependem funcionalmente e exclusivamente da chave-primária, o que se revela como sendo apenas uma extensão dos requisitos da Terceira Forma Normal.

**Restrições à base de dados**

**Tabela ‘person’:**

* Restrição chave-primária: personID (deve ser único e não-nulo);
* Restrição Not Null (os seguintes parâmetros não podem ser nulos):
  + name;
  + address;
  + phone\_number;
  + nif;
  + gender;
* Restrição Unique (os seguintes parâmetros são únicos na tabela):
  + name;
  + nif;
* Restrição Check (os seguintes parâmetros têm que cumprir uma dada condição):
  + gender: deve ser igual a ‘M’ ou a ‘F’;

**Tabela ‘passenger’:**

* Restrição chave-primária: passengerID (deve ser único e não-nulo);
* Restrição chave-estrangeira: passengerID é uma referência a person<personID>;
* Restrição Not Null: ticket\_number: não deve ser nulo;
* Restrição Unique: ticket\_number: não se deve repetir na tabela;
* Restrição Check: ticket\_number: deve ser superior a zero;

**Tabela ‘employee’:**

* Restrição chave-primária: employeeID (deve ser único e não-nulo);
* Restrição chave-estrangeira: employeeID é uma referência a person<personID>;
* Restrição Not Null (os seguintes parâmetros não podem ser nulos):
  + hiring\_date;
  + wage;
* Restrição Check: wage deve ser superior a zero;

**Tabela ‘trainDriver’:**

* Restrição chave-primária: trainDriverID (deve ser único e não-nulo);
* Restrição chave-estrangeira: trainDriverID é uma referência a employee<employeeD>;
* Restrição Not Null: position não deve ser nula;
* Restrição Check: position deve ter valor igual a ‘Train Driver’;

**Tabela ‘conductor’:**

* Restrição chave-primária: conductorID (deve ser único e não-nulo);
* Restrição chave-estrangeira: conductorID é uma referência a employee<employeeD>;
* Restrição Not Null: position não deve ser nula;
* Restrição Check: position deve ter valor igual a ‘Conductor’;

**Tabela ‘line’:**

* Restrição chave-primária: lineID (deve ser único e não-nulo);
* Restrição Not Null (os seguintes parâmetros não podem ser nulos):
  + name;
  + color;
* Restrição Unique: name não se deve repetir na tabela;

**Tabela ‘train’:**

* Restrição chave-primária: trainID (deve ser único e não-nulo);
* Restrição chave-estrangeira:
  + lineID é uma referencia para line<lineID>;
  + trainDriverID é uma referência para trainDriver<trainDriverID>;
* Restrição Not Null: capacity não deve ser nula;
* Restrição Check: capacity deve ter valor superior a zero;

**Tabela ‘trip’:**

* Restrição chave-primária: tripID (deve ser único e não-nulo);
* Restrição chave-estrangeira:
  + trainID é uma referência para train<trainID>;
* Restrição Not Null (os seguintes parâmetros não podem ser nulos):
  + current\_capacity;
  + beginning\_hour;
  + ending\_hour;
  + trip\_date;
* Restrição Check: current\_capacity deve ter valor superior a zero;

**Tabela ‘stop’:**

* Restrição chave-primária: stopID (deve ser único e não-nulo);
* Restrição Not Null: location não deve ser nula;
* Restrição Unique: location não se deve repetir na tabela;

**Tabela ‘inspects’:**

* Restrição chave-primária: {inspectorID, trainID} (devem ser únicos e não-nulos);
* Restrição chave-estrangeira:
  + inspectorID é uma referencia para conductor<conductorID>;
  + trainID é uma referência para train<trainID>;

**Tabela ‘belongs’;**

* Restrição chave-primária: {lineID, stopID} (devem ser únicos e não-nulos);
* Restrição chave-estrangeira:
  + lineID é uma referencia para line<lineID>;
  + stopID é uma referência para stop<stopID>;

**Tabela ‘travels’:**

* Restrição chave-primária: {tripID, passengerID} (devem ser únicos e não-nulos);
* Restrição chave-estrangeira:
  + tripID é uma referencia para trip<tripID>;
  + passengerID é uma referência para passenger<passengerID>;