

Bases de Dados

Gestão de Aeroportos

Grupo 506 – 2MIEIC05

(15 de abril de 2018)

José Rodrigues	up201708806@fe.up.pt
Luís Borges	up201605859@fe.up.pt
Sandro Campos	up201605947@fe.up.pt

Descrição

Este projeto baseia-se na gestão de um aeroporto e das suas partes integrantes principais inerentes ao seu funcionamento.

A base de dados armazena, sobre uma instância de **Aeroporto**, a informação que o representa e identifica, como a *sigla*, o seu *nome* e a *cidade* em que este se encontra.

Na relação com um aeroporto encontram-se presentes os seus **Funcionários**, as **Companhias Aéreas**, e os **Voos** que são operados por estas e que têm ligação com o respetivo aeroporto – isto é, os voos tanto podem descolar como aterrar nele.

Cada **Funcionário** tem um *cargo* e um *salário*, além de ser caracterizado por um *nome*, *NIF*, *data de nascimento*, *nacionalidade*, *morada*, *código postal* e ainda número de *telefone*, associada ao facto de este ser uma **Pessoa**.

Uma pessoa que não faça parte dos funcionários será um **Passageiro** que passa pelo aeroporto, tem um **Bilhete** com a informação sobre o *lugar* que ocupa no avião e o *tipo* de bilhete que possui para a viagem, que pode ser de classe económica, executiva ou primeira classe. Cada viajante tem a possibilidade de trazer consigo **Bagagens**, de *peso* e *dimensões* bem definidos.

Do voo interessa saber qual o seu *número*, as suas *horas de partida e de chegada*, a *lotação* a um determinado momento e a *data* na qual se realiza. Este é manobrado por 2 **Pilotos** (Piloto e Copiloto), entre os aeroportos de origem e destino.

Por fim e retratando a realidade, cada **Aeronave** é única, identificada pela sua *matrícula*, um **Modelo** e uma **Marca** que o produz. De forma a evitar inconsistências na informação, certifica-se que a *lotação* da aeronave terá que obrigatoriamente não exceder a *capacidade* da mesma.

Atributos

Pessoa

- Nome
- *NIF*
- *Data de Nascimento*
- Nacionalidade
- Morada
- Código Postal
- Telefone

Funcionário

- Cargo
- Salário

Passageiro

- (...)

Piloto

- (...)

Bilhete

- Tipo
- Lugar

Voo

- Número do Voo
- Hora de Partida
- Hora de Chegada
- Lotação
- Data

Bagagem

- Peso
- Dimensões

Aeroporto

- Sigla
- Nome
- Cidade

Companhia Aérea

- IATA
- Nome
- País

Aeronave

- Matrícula

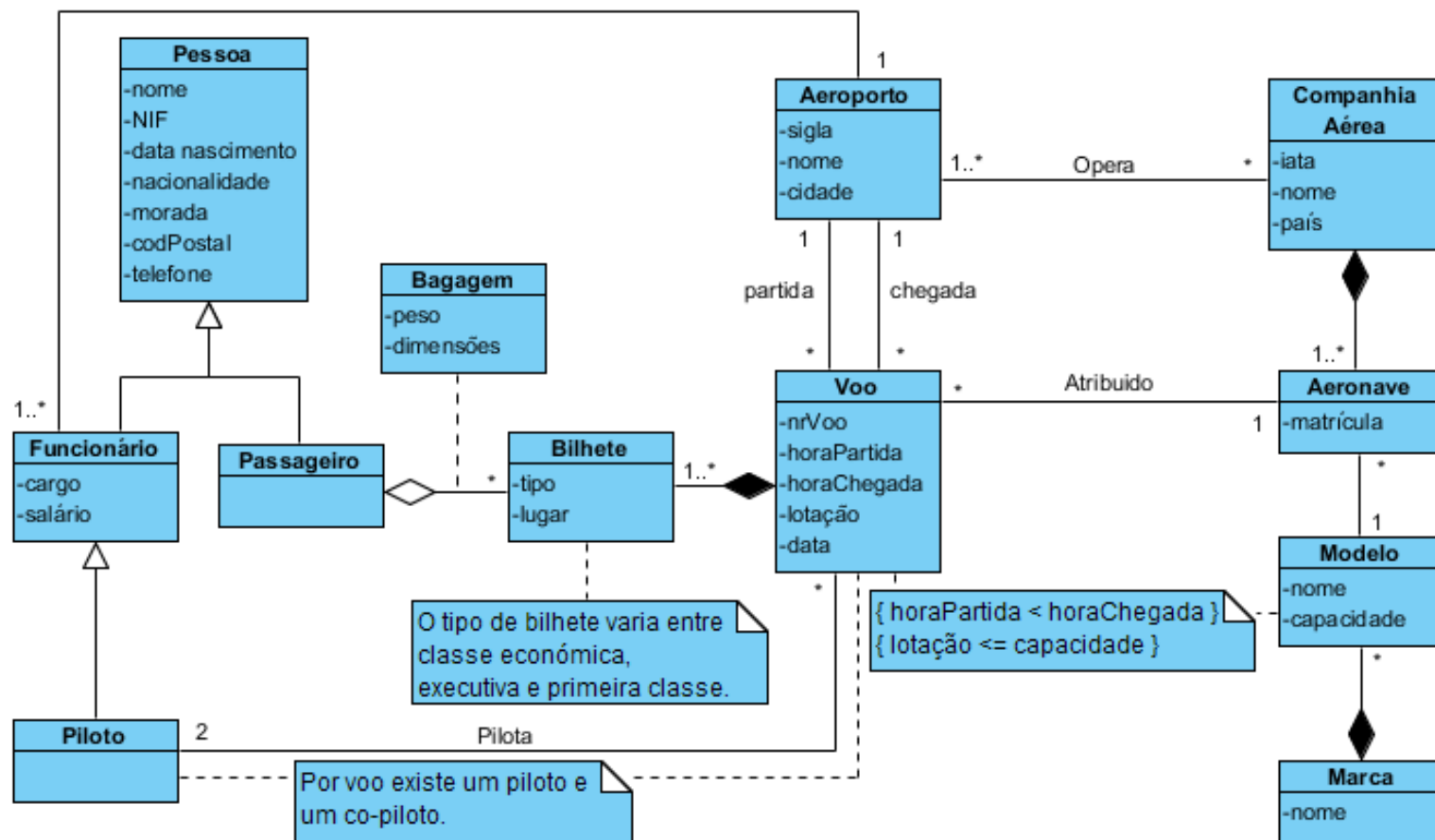
Modelo

- Nome
- Capacidade

Marca

- Nome

Diagrama de Classes



Esquema Relacional e Dependências Funcionais

Aeroporto (sigla, nome, cidade)

- sigla -> {nome, cidade}

- {nome, cidade} -> sigla

. Assumindo que a sigla é um identificador único do aeroporto e que, numa determinada cidade, não podem existir aeroportos com nomes em comum.

CompanhiaAerea (iata, nome, país)

- iata -> {nome, país}

- nome -> {iata, país}

. Considerando que cada companhia tem apenas um determinado país de origem e não existem companhias aéreas com nomes em comum.

Pessoa (nome, nif, dataNascimento, nacionalidade, morada, codPostal, telefone)

- nif -> {nome, dataNascimento, nacionalidade, morada, codPostal, telefone}

- telefone -> nif

. Considerando que cada pessoa, além de um nif, possui também um número de telefone único.

Funcionário (nif->Pessoa, cargo, salário, aeroporto->Aeroporto)

- {nif->Pessoa} -> {cargo, salário, aeroporto->Aeroporto}

Piloto (nif->Pessoa)

Passageiro (nif->Pessoa)

Voo (nrVoo, horaPartida, horaChegada, lotação, data)

- nrVoo -> {horaPartida, horaChegada, lotação, data}

Bilhete (tipo, lugar, nif->Pessoa, nrVoo->Voo)

- {lugar, nrVoo->Voo} -> {tipo, nif->Pessoa}

- {nif->Pessoa, nrVoo->Voo} -> {tipo, lugar}

. Considerando que a cada tipo de bilhete é reservado um conjunto de assentos no avião, separado dos outros restantes tipos.

Bagagem (lugar, nrVoo->Bilhete, nif->Pessoa, peso, dimensões)

- {lugar, nrVoo->Bilhete} -> {nif->Pessoa, peso, dimensões}

Aeronave (matricula, iata->CompanhiaAerea, modelo->Modelo)

- matrícula -> {iata->CompanhiaAerea, modelo->Modelo}

Modelo (nome, capacidade, marca->Marca)

- nome -> {capacidade, marca->Marca}

Marca (nome)

Opera (iata->CompanhiaAerea, aeroporto->Aeroporto)

Partida (nrVoo->Voo, sigla->Aeroporto)

- {nrVoo->Voo} -> {sigla->Aeroporto}

Chegada (nrVoo->Voo, sigla->Aeroporto)

- {nrVoo->Voo} -> {sigla->Aeroporto}

Pilota (nrVoo->Voo, nif->Pessoa)

- {nif->Pessoa} -> {nrVoo->Voo}

Atribuido (nrVoo->Voo, matricula->Aeronave)

- {nrVoo->Voo} -> {matricula->Aeronave}

Formas Normais

O modelo relacional apresentado cumpre a **BCNF** (Forma Normal de Boyce Codd) uma vez que todas as dependências funcionais não triviais acima possuem uma (super)chave à sua esquerda.

Acontece que, no entanto, não se encontra na **3NF** devido à transitividade existente entre dependências funcionais presentes na relação *Pessoa*. Observa-se que, ao se assumir um *telefone* único para cada pessoa, é possível obter o seu *nif* que por sua vez nos permite saber todos os outros atributos a ela referente. Este facto viola, por isso, o princípio necessário ao cumprimento desta regra.

Restrições

De forma a reforçar a consistência da base de dados e a assegurar uma maior correção dos dados nela inseridos, implementámos algumas restrições que considerámos relevantes.

Entre elas podemos encontrar restrições do tipo **PRIMARY KEY**, **FOREIGN KEY** e **REFERENCES**, que permitem declarar chaves primárias e estrangeiras, respetivamente, e **NOT NULL**, **UNIQUE** e **CHECK**, representando restrições de integridade referencial. Destas últimas, a primeira impossibilita que o atributo a ela associado seja nulo, ou seja inexistente, a segunda assegura que o atributo é único na relação, ainda que este não seja chave, e a última permite-nos verificar uma determinada condição entre um ou mais atributos.

A seguir apresentam-se, para cada relação, as suas utilizações.

Relação Pessoa:

- . O *nome*, a *dataNascimento* e a *nacionalidade* não podem ser nulos - Restrição **NOT NULL**
- . Não pode haver duas pessoas com o mesmo *nif*, nem este pode ser nulo - Restrição **PRIMARY KEY**
- . Não pode haver duas pessoas com o mesmo *telefone* - Restrição **UNIQUE**

Relação Funcionário:

- . Não pode haver dois funcionários com o mesmo *nif*, nem este pode ser nulo - Restrições **PRIMARY KEY REFERENCES Pessoa** e **NOT NULL**
- . O *cargo* e o *salário* não podem ser nulos - Restrição **NOT NULL**
- . Um funcionário trabalha num e num só aeroporto – Restrições **REFERENCES Aeroporto** e **NOT NULL**

Relação Passageiro:

- . Não pode haver dois passageiros com o mesmo *nif*, nem este pode ser nulo - Restrições **PRIMARY KEY REFERENCES Pessoa** e **NOT NULL**

Relação Bilhete:

. O *tipo* varia entre "1" (primeira classe), "2" (classe executiva) ou "3" (classe económica) e não pode ser nulo - Restrições **CHECK (*tipo* = 1 or *tipo* = 2 or *tipo* = 3)** e **NOT NULL**

. Não pode haver dois voos com o mesmo *nrVoo*, nem este pode ser nulo - Restrição **REFERENCES Voo**

. Não pode haver dois passageiros com o mesmo *nif*, nem este pode ser nulo - Restrições **REFERENCES Passageiro** e **NOT NULL**

. Não pode haver dois bilhetes num mesmo voo com o mesmo *lugar* - Restrição **PRIMARY KEY(lugar, nrVoo)**

Relação Voo:

. Não pode haver dois voos com o mesmo *nrVoo* - Restrição **PRIMARY KEY**

. A *horaPartida* tem de ser anterior à *horaChegada*, e nenhuma destas pode ser nula - Restrições **CHECK (*horaChegada* < *horaPartida*)** e **NOT NULL**

. A *lotacao* tem que ser positiva – Restrições **CHECK(*lotacao* >= 0)** e **NOT NULL**

. A *data* não pode ser nula – Restrição **NOT NULL**

Relação Aeroporto:

. Não pode haver dois aeroportos com a mesma *sigla* - Restrição **PRIMARY KEY**

Relação Companhia Aérea:

. Não pode haver duas companhias aéreas com o mesmo *iata* - Restrição **PRIMARY KEY**

Relação Aeronave:

. Não pode haver duas aeronaves com a mesma *matricula* - Restrição **PRIMARY KEY**

. O *iata* da companhia a que a aeronave pertence não pode ser nulo – Restrições **REFERENCES CompanhiaAerea** e **NOT NULL**

. O *modelo* não pode ser nulo - Restrições **REFERENCES Modelo** e **NOT NULL**

Relação Modelo:

- . Não pode haver dois modelos com o mesmo *nome* - Restrição **PRIMARY KEY**
- . A *marca* de um modelo não pode ser nula – Restrições **REFERENCES Marca** e **NOT NULL**
- . A *capacidade* tem que ser positiva – Restrições **CHECK(capacidade >= 0)** e **NOT NULL**

Relação Marca:

- . Não pode haver duas marcas com o mesmo *nome* - Restrição **PRIMARY KEY**

Relação Piloto:

- . Não pode haver dois pilotos com o mesmo *nif*, nem este pode ser nulo - Restrições **REFERENCES Funcionario** e **NOT NULL**

Relação Partida:

- . Não pode haver dois voos com o mesmo *nrVoo*, nem este pode ser nulo – Restrições **PRIMARY KEY REFERENCES Voo** e **NOT NULL**
- . Não pode haver dois aeroportos com a mesma *sigla*, nem esta pode ser nula – Restrições **REFERENCES Aeroporto** e **NOT NULL**

Relação Chegada:

- . Não pode haver dois voos com o mesmo *nrVoo*, nem este pode ser nulo – Restrições **PRIMARY KEY REFERENCES Voo** e **NOT NULL**
- . Não pode haver dois aeroportos com a mesma *sigla*, nem esta pode ser nula – Restrições **REFERENCES Aeroporto** e **NOT NULL**

Relação Bagagem:

- . O *nrVoo* e o *lugar* não podem ser nulos – Restrição **NOT NULL**
- . O *peso* tem que ser positivo – Restrições **CHECK(peso >= 0)** e **NOT NULL**
- . Não pode haver duas bagagens com o mesmo *lugar* e *nrVoo*, ou seja, cada bagagem pertence a um e um só bilhete – Restrições **PRIMARY KEY(lugar, nrVoo)** e **FOREIGN KEY(lugar, nrVoo) REFERENCES Bilhete**

Relação Pilota:

. Não pode haver dois voos com o mesmo *nrVoo*, nem este pode ser nulo – Restrições

PRIMARY KEY REFERENCES Voo e NOT NULL

. Não pode haver dois pilotos com o mesmo *nif*, nem este pode ser nulo - Restrições

REFERENCES Piloto e NOT NULL

Relação Opera:

. Não pode haver duas companhias aéreas com o mesmo *iata*, nem este pode ser nulo -

Restrições **REFERENCES CompanhiaAerea e NOT NULL**

. Não pode haver dois aeroportos com a mesma *sigla*, nem esta pode ser nula -

Restrições **REFERENCES Aeroporto e NOT NULL**

Relação Atribuido:

. Não pode haver dois voos com o mesmo *nrVoo*, nem este pode ser nulo – Restrição

PRIMARY KEY REFERENCES Voo e NOT NULL

. Não pode haver duas aeronaves com a mesma *matrícula*, nem esta pode ser nula –

Restrições **REFERENCES Aeronave e NOT NULL**