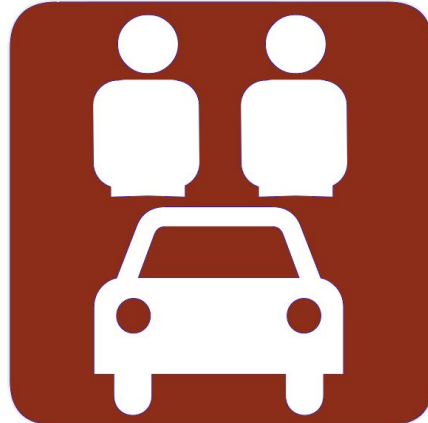


Base de Dados FEUP Sharing

2MIEIC01 - Grupo 105



| | |
|--------------------|--------------------|
| Diogo Yaguas | up201606165 |
| Duarte Frazão | up201605658 |
| Henrique Gonçalves | up201608320 |

11 de março de 2018

Descrição

Este projeto consiste numa plataforma de partilha de viagens de carro, de forma a diminuir o número de carros que viajam todos os dias para a faculdade dentro da comunidade FEUP.

A aplicação vai disponibilizar a um **Utilizador** da plataforma duas opções: no caso de também se registar como um **Condutor**, a **Partilha** de uma ou mais **Viagens**, ou então associar-se a uma já existente na sua localidade, mostrando interesse ou efetuar uma viagem.

A base de dados armazena uma grande quantidade de dados relativamente ao **Utilizador**, as suas informações pessoais como o nome, número up, número de telemóvel e a sua data de nascimento, mas também a sua pontuação enquanto passageiro.

Para um **Utilizador** se registar como **Condutor** (Classe derivada), terá de especificar o número e validade da sua carta de condução, tal como informações relativas ao **Carro** com que irá conduzir, como a sua matrícula, **Modelo** e **Marca**. O **Condutor** também possui uma pontuação.

O **Condutor** introduz uma **Partilha** na plataforma, que representa a(s) **Viagens** com características iguais que irá realizar nas semanas seguintes, indicando a hora de começo da viagem, dia da semana, preço total, lugares disponíveis, direção (origem/destino na FEUP), e o intervalo de datas em que estas **Viagens** se vão realizar.

O **Utilizador** pode demonstrar interesse numa **Viagem** indicando as **Zonas** por qual terá de passar de forma a que o condutor o possa buscar/levar.

No final de uma **Viagem**, caso esta se efetue, o **Condutor** avalia os passageiros e os passageiros o condutor, ambos de 1-5. Este par de avaliações fica guardada na classe **Avaliação**, que depois irão ser guardados na pontuação do condutor e do passageiro respetivamente através de uma média.

Também é disponibilizado um serviço de troca de mensagens entre **Utilizadores**, para facilitar a comunicação e algum ajuste que seja preciso fazer.

Atributos

Utilizador

- Nome
- Data de nascimento
- Número UP
- Número de Telemóvel
- Pontuação do Passageiro

Condutor

- Número de carta
- Data de expiração
- Pontuação do Condutor

Partilha

- flag to_feup(valor booleano)
- Preço total
- Número de lugares livres
- Hora de início
- Data de início
- Data de fim
- Dia da semana

Viagem

- Data

Avaliação

- Avaliação do Condutor
- Avaliação do Utilizador

Zona

- Nome do local

LocalParagem

- Morada
- Coordenadas GPS

Mensagem

- Texto da mensagem
- Data

Carro

- Matrícula

Modelo

- Nome do modelo

Marca

- Nome da Marca

Diagrama UML

Esquema Relacional, Dependências Funcionais e Formas Normais

Utilizador(nome, data_nascimento, **número_up**, número_telemóvel, pontuação_passageiro)

2. {número_up} → {id, nome, data_nascimento, número_telemóvel, pontuação_passageiro}

3. {número_telemóvel} → {id, nome, data_nascimento, número_up, pontuação_passageiro}

Nesta classe **numero_up** e **numero_telemovel** são chaves logo esta relação não viola nem a 3NF nem a BCNF.

- **Condutor** (id → Utilizador, numero_carta, data_validade_carta, pontuação_condutor, idCarro → Carro)

1. {id → Utilizador} → {número_carta, data_validade_carta, pontuação_condutor, idCarro}

2. {número_carta} → {id → Utilizador, data_validade_carta, pontuação_condutor, idCarro}

Uma vez que tanto o id como o numero da carta são chaves esta não viola nem a BCNF nem a 3NF

- **Mensagem** (id, texto, data, Emissor → Utilizador, Recetor → Utilizador)

1. {id} → {texto, data, Emissor, Recetor}

Esta FD não viola nem a BCNF nem a 3NF, uma vez que **id** é a chave desta relação

- **Carro** (id, matricula, idModelo → Modelo)

1. {id} → {matricula}

2. {matricula} → {id, idModelo}

Como tanto a matricula como o id são chaves, esta não viola nem a 3NF nem a BCNF

- **Modelo** (id, nome, idMarca → Marca)

1. {id} → {nome, idMarca}

2. {nome, idMarca} → {id}

Como id e (nome, idMarca) são chaves esta relação não viola a 3NF nem a BCNF

- **Marca** (id, nome)

1. {id} → {nome}

2. {nome} → {id}

Como tanto id como nome são chaves, esta relação não viola a BCNF nem a 3NF

- **Partilha** (id, para_feup, preço_total, lugares_disponíveis, hora_começo, data_inicio, data_fim, dia_da_semana, idCondutor → Condutor)

1. {id} → {para_feup, preço_total, lugares_disponíveis, hora_começo, data_inicio, data_fim, dia_da_semana, idCondutor}

A FD da relação tem **id** como uma chave da relação então pode-se concluir que não é violada nenhuma das formas normais a ser analisadas.

- **Viagem** (id, data, partilha_associada → Partilha)

1. {id} → {data, idPartilha}

Existe uma única dependência funcional da relação. Como **id** é chave da relação, esta FD não viola a Forma Normal Boyce-Codd e, portanto, também não viola a 3ª Forma Normal.

- **Zona** (id, nome)

1. {id} → {nome}

2. {nome} → {id}

Como ambos os lados esquerdos das DP são chaves, podemos concluir que não viola a BCNF nem a 3NF.

- **LocalidadesPossiveis**(idZona → Zona, idViagem → Viagem)

A relação não possui qualquer dependência funcional, pelo que não violam a BCNF e, consequentemente, a 3NF.

- **Avaliação**(idUtilizador → Utilizador, idViagem → Viagem, avaliação_condutor, avaliação_passageiro)
1. {idUtilizador, idViagem} → {avaliação_condutor, avaliação_passageiro}

Tanto **idUtilizador** como **idViagem** são chave da relação **Avaliação** logo encontra-se na BCNF e, consequentemente, encontra-se também na 3NF.

- **LocalParagem** (idUtilizador → Utilizador, idViagem → Viagem, morada, coordenadas_GPS, idZona → Zona)
1. {idUtilizador, idViagem} → {morada, coordenadas_GPS}
2. {morada} → {idZona}
3. {coordenadas_GPS} → {idZona}

Como na segunda DP o lado esquerdo não é uma chave ou super chave a BCNF é violada, como o lado direito não é nenhum primo a 3NF é violada.

Restrições e respetiva implementação

Os seguintes atributos (identificados com sublinhado) possuem a restrição **PRIMARY KEY**, na sua relação (identificada a **negrito**), dado que identificam um tuplo na tabela respetiva de forma única:

Utilizador: id - não pode haver dois utilizadores com o mesmo id.

Mensagem: id - não pode haver duas mensagens com o mesmo id.

Carro: id - não pode haver dois carros com o mesmo id.

Modelo: id - não pode haver dois modelos com o mesmo id.

Marca: id - não pode haver duas marcas com o mesmo id.

Partilha: id - não pode haver duas partilhas com o mesmo id.

Viagem: id - não pode haver duas viagens com o mesmo id.

Zona: id - não pode haver duas zonas com o mesmo id.

Os seguintes atributos (identificados com sublinhado) possuem a restrição **FOREIGN KEY**, na sua relação (identificada a **negrito**), dado que representam entidades pertencentes a outras tabelas da base de dados, tendo de existir na sua relação:

Condutor: id(chave estrangeira para Utilizador) e idCarro(chave estrangeira para Carro)

Mensagem: Emissor(chave estrangeira para Utilizador) e Recetor(chave estrangeira para Utilizador)

Carro: idModelo(chave estrangeira para Modelo)

Modelo: idMarca(chave estrangeira para Marca)

Partilha: idCondutor(chave estrangeira para Condutor)

Viagem: partilha_associada(chave estrangeira para Partilha)

LocalidadesPossiveis: idZona (chave estrangeira para Zona) e idViagem (chave estrangeira para Viagem).

Avaliação: idUtilizador (chave estrangeira para Utilizador) e idViagem (chave estrangeira para Viagem).

LocalParagem: idUtilizador (chave estrangeira para Utilizador) e idViagem (chave estrangeira para Viagem).

Os únicos atributos (identificados com sublinhado) que não possuem a restrição **NOT NULL**, na sua relação (identificada a **negrito**), são os seguintes (todos os outros a possuem):

Utilizador: pontuação_passageiro - inicialmente a sua pontuação está a NULL e assim poderá continuar

Condutor: pontuação_condutor - inicialmente a sua pontuação está a NULL

Avaliação: avaliação_condutor e avaliação_passageiro podem ser NULL visto que poderão não receber uma avaliação.

As seguintes restrições são implementadas na respetiva relação através de um **CHECK** para assegurar que qualquer combinação dos atributos envolvidos nas instanciações tornem verdade as seguintes condições:

Utilizador: o número_telemóvel deve ter 9 dígitos e pontuação_passageiro tem de ser positiva

Condutor: data_validade_carta deve ser posterior à data atual e pontuação_condutor tem de ser positiva

Carro: matricula tem de obedecer ao método atual que se incute nas matrículas, isto é, ter 4 números e 2 letras

Partilha: para_feup tem de ser 0 ou 1, preço_total tem de ser positivo, lugares_disponiveis tem de ser 0 ou positivo e data_fim não pode ser anterior à data_inicio

Avaliação: avaliação_condutor e avaliação_passageiro têm de ser positivos.

Os seguintes atributos (identificados com sublinhado) possuem a restrição **UNIQUE** na sua relação (identificada a **negrito**), dado que representam entidades identificadores de uma certa classe, embora não operem como chaves primárias:

Utilizador: número_up (não pode haver dois utilizadores com o mesmo número up) e número_telemóvel (não pode haver dois utilizadores com o mesmo número de telemóvel)

Condutor: número_carta (não pode haver dois condutores com o mesmo número da carta)

Carro: matricula (não pode haver dois carros com a mesma matrícula)

Modelo: nome (não pode haver dois modelos com o mesmo nome)

Zona: nome (não pode haver duas zonas com o mesmo nome)

Criação e Povoamento da Base de Dados

As instruções SQL para a criação (com as restrições possíveis de implementar) e povoamento da base dados encontram-se nos ficheiro `criar.sql` e `povoar.sql`, respetivamente.