

Projecto de Bases de Dados, Parte 2

Professor Gabriel Pestana

Turno: Quinta-feira, 11:00-12:30

Grupo 45

Inês Sequeira, nº 81719

Pedro Gomes, nº 81534

Rafael Belchior, nº 80970

Esforço (em horas)

81719 ----- 3 horas

81534 ----- 3 horas

80970 ----- 3 horas

Modelo Relacional:

Edificio(morada)

Alugavel_cont(codigo, morada, foto)
morada: FK Edificio(morada)

Posto(codigoPosto, morada, codigo)
codigoPosto, morada : FK Alugavel_cont(codigo, morada)
codigo: FK Espaco (codigo)

Espaco(codigo, morada)
codigo, morada : FK Alugavel_cont(codigo, morada)

User(NIF, telefone, nome)

Arrenda(codigo, morada, NIF)
NIF: FK User(NIF)
codigo, morada: FK Alugavel_cont(codigo, morada)

Fiscal(ID, empresa)

Fiscaliza(ID, codigo, morada, NIF)
ID: FK Fiscal(ID)
NIF: FK User(NIF)
codigo, morada: FK Alugavel_cont(codigo, morada)

Oferta_ofer(codigo, morada, data_inicio, data_fim, tarifa)
codigo, morada : FK Alugavel_cont(codigo, morada)

Reserva(numero)

Paga(numero, data, metodo)
numero : FK Reserva(numero)

Estado_log(timestamp, numero, estado)
numero : FK Reserva(numero)

Aluga(NIF, codigo, morada, data_inicio, numero)
codigo, morada, data_inicio : FK Oferta_ofer(codigo, morada, data_inicio)
NIF: FK User(NIF)
numero : FK Reserva(numero)

Restrições de Integridade:

RI1: Posto and Espaco covers Alugavel_cont.

RI2: O atributo "Estado" da relação "Reserva" pode conter os valores "Pendente", "Aceite", "Declinada" ou "Cancelada".

RI3: Uma Reserva só pode ser paga se estado actual for "Aceite".

RI4: No máximo, só pode existir uma Reserva aceite sobre cada oferta.

RI5: As ofertas para o mesmo Alugavel_cont não se podem sobrepor no tempo.

RI6: O atributo codigo da relação Alugavel_cont deve ser um número sequencial.

Álgebra relacional:

1.

$\rho_3 \leftarrow \text{numOfertas}(\text{morada}, \text{codigo} \text{ G}_{\text{count}(\text{data_inicio})}(\text{Oferta_ofer}))$

$\pi_{\text{morada}}(\sigma_{\text{numOfertas} > 1}(\text{Ofertas}))$

2.

$\rho_2 \leftarrow \text{maisRecente}(\text{numero} \text{ G}_{\text{max}(\text{timestamp})}(\text{Paga} \bowtie \text{Estado_log}) \bowtie \text{Estado_log})$

$\pi_{\text{numero}, \text{estado}}(\sigma_{\text{timestamp} = \text{maisRecente}}(\text{EstadoMaisRecente}))$

3.

$\rho_3 \leftarrow \text{numPostosAceites}(\text{codigo}, \text{morada} \text{ G}_{\text{count}(\text{codigoPosto})}(\sigma_{\text{estado} = \text{"aceite"}}(\text{Posto} \bowtie \text{Oferta_ofer} \bowtie \text{Aluga} \bowtie \text{Estado_log})))$

$\rho_3 \leftarrow \text{numPostos}(\text{codigo}, \text{morada} \text{ G}_{\text{count}(\text{codigoPosto})}(\text{Posto}))$

$\pi_{\text{codigo}, \text{morada}}(\sigma_{\text{numPostos} = \text{numPostosAceites}}(\text{PostosAceites} \bowtie \text{PostosTotal}))$

SQL:

1.

```
SELECT morada
FROM Oferta_ofer
GROUP BY morada, codigo
HAVING COUNT(data_inicio) > 1
```

2.

```
SELECT e1.numero, e1.estado
FROM Paga NATURAL JOIN Estado_log e1
WHERE e1.timestamp = (
    SELECT MAX(e2.timestamp)
    FROM Paga NATURAL JOIN Estado_log e2
    GROUP BY e2.numero
    WHERE e1.numero = e2.numero);
```