

Departamento de Informática

licenciatura em Engenharia Informática

- A GARE -

(Cadeira de Base de Dados)

2ºAno



Eduardo Simão Ramos Almeida Costa Martins, 29035

Daniel Gonçalo Jesus Ramos, 29423

Évora, 2012

marcaAut	
Superchaves	{Modelo}, {Modelo, Marca}, {Modelo, NmaxP}, {Modelo, Marca, NmaxP}
Chaves primárias	{Modelo}
Chaves Candidatas	{Modelo}
Chaves estrangeiras	Não existe

autocarro	
Superchaves	{Matricula}, {Matricula, Modelo}, {Matricula, AnoMatricula}, {Matricula, Modelo, AnoMatricula}
Chaves primárias	{Matricula}
Chaves Candidatas	{Matricula}
Chaves estrangeiras	{Modelo}

motorista	
Superchaves	{NBi}, {NBi, Nome}, {NBi, Morada}, {NBi, Nome, Morada}
Chaves primárias	{NBi}
Chaves Candidatas	{NBi}
Chaves estrangeiras	Não existe

percurso	
Superchaves	{CodigoP}, {CodigoP, NomeP}, {CodigoP, NMaxP}
Chaves primárias	{CodigoP}
Chaves Candidatas	{CodigoP}
Chaves estrangeiras	Não existe

passoPercurso	
Superchaves	{CodigoP, Minutos, Paragem}
Chaves primárias	{CodigoP, Minutos, Paragem}
Chaves Candidatas	{CodigoP, Minutos, Paragem}

Chaves estrangeiras	{CodigoP},{Paragem}
----------------------------	---------------------

fezPercurso

Superchaves	{DataInicio, NBi} , {DataInicio, Matricula}, {DataInicio, CodigoP, NBi}, {DataInicio, CodigoP, Matricula}, {DataInicio, NBi, Matricula}, {DataInicio, DataFim, NBi}, {DataInicio, DataFim, Matricula}, {DataFim, NBi}, {DataFim, Matricula}, {DataFim, CodigoP, NBi} , {DataFim, CodigoP, Matricula}, {DataInicio, CodigoC, NBi, Matricula}, {DataInicio, NBi, Matricula, DataFim}, {DataInicio, DataFim, CodigoP, NBi}, {DataInicio, DataFim, CodigoP, NBi}, {DataInicio, NBi, Matricula, DataInicio, DataFim}
Chaves primárias	{DataInicio, NBi}
Chaves Candidatas	{DataInicio, NBi}, {DataInicio, Matricula}
Chaves estrangeiras	{CodigoP}, {NBi}, {Matricula}

Bilhete

Superchaves	{NBi, DataPass}, {NBi, DataPass, Paragem}, {NBi, DataInicio, DataPass, Paragem}
Chaves primárias	{NBi, DataPass}
Chaves Candidatas	{NBi, DataPass}
Chaves estrangeiras	{NBi}, {Paragem}, {DataInicio}

Paragem

Superchaves	{Paragem}
Chaves primárias	{Paragem}
Chaves Candidatas	{Paragem}
Chaves estrangeiras	Não existe

a) Marca, Modelo e Matricula dos autocarros que podem fazer a carreira 23

Expressão em Álgebra Relacional

π Marca, Modelo, Matricula(σ percurso.CodigoP = 23 \wedge marcaAut.NmaxP \geq percurso.NmaxP \wedge autocarro.Modelo = marcaAut.Modelo(marcaAut \times percurso \times autocarro))

Expressão em SQL

```
SELECT marcaAut.Marca, marcaAut.Modelo, autocarro.Matricula FROM percurso, autocarro, marcaAut WHERE (percurso.CodigoP=23 AND autocarro.Modelo = marcaAut.Modelo AND marcaAut.NmaxP >= percurso.NmaxP);
```

Output

marca	modelo	matricula
MAN	13-220	FH-76-MG
MAN	13-220	SS-98-KL
Mercedes	The Citaro	JT-65-BE
IVECO	Eurorider 43	EW-45-QQ
IVECO	Eurorider 43	PL-21-PO

b) Marca e Matricula dos autocarros que têm mais de 10 anos

Expressão em Álgebra Relacional

π Marca, Matricula(σ (2012 – AnoMatricula) > 10(marcaAut \bowtie autocarro))

Expressão em SQL

```
SELECT Marca, Matricula FROM (SELECT Marca, Matricula, (2012-anoMatricula) as Idade FROM marcaAut NATURAL INNER JOIN autocarro) as temp WHERE (Idade > 10);
```

Output

marca	matricula
MAN	FH-76-MG
MAN	SS-98-KL
Mercedes	JT-65-BE

c) Código e nome dos percursos que passam no “Lg Luís de Camões”

Expressão em Álgebra Relacional

$\pi_{CodigoP, NomeP}(\sigma_{Paragem = \text{“Lg Luis de Camoes”}}(passoPercurso \bowtie percurso))$

Expressão em SQL

SELECT DISTINCT CodigoP, NomeP **FROM** passoPercurso **NATURAL INNER JOIN** percurso
WHERE(paragem = 'Lg Luis de Camoes');

Output

codigop	nomep
21	Louredo-Lg Luis de Camoes
23	25 de Abril-Malagueira

d) Data-horas em que passou a carreira 21 em Louredo

Expressão em Álgebra Relacional

$\pi_{Minutos}(\sigma_{Paragem = \text{“Louredo”} \wedge CodigoP = 21}(passoPercurso))$

Expressão em SQL

SELECT Minutos **FROM** passoPercurso **WHERE**(CodigoP=21 **AND** Paragem='Louredo');

Output

minutos

06:26:00

07:26:00

07:39:00

11:51:00

13:51:00

15:51:00

18:39:00

19:39:00

08:27:00

09:17:00

10:17:00

12:39:00

14:39:00

15:49:00

16:39:00

19:27:00

20:27:00

e) Código e nome dos percursos que passam pela Malagueira e pela Vista Alegre

Expressão em Álgebra Relacional

$$\pi_{percurso.CodigoP, NomeP}(\sigma_{percurso.CodigoP = passoPercurso.CodigoP \wedge passoPercurso.CodigoP = passoPercursoB.CodigoP \wedge passoPercurso.Paragem = "Malagueira" \wedge passoPercursoB.Paragem = "Vista Alegre"}(passoPercurso \times percurso \times ppassoPercursoB(passoPercurso)))$$

Expressão em SQL

```
SELECT DISTINCT percurso.CodigoP, NomeP FROM passoPercurso, percurso, passoPercurso
as passoPercursoB WHERE ( percurso.CodigoP = passoPercurso.CodigoP AND
passoPercurso.CodigoP = passoPercursoB.CodigoP AND passoPercurso.Paragem =
'Malagueira' AND passoPercursoB.Paragem = 'Vista Alegre');
```

Output

codigop	nomep
23	25 de Abril-Malagueira

f)Nome dos motoristas da carreira 21 no mês de Novembro de 2009

Expressão em Álgebra Relacional

$\pi_{Nome}(\sigma_{CodigoP = 21 \wedge DataInicio \geq "2009-11-01\ 00:00:00" \wedge DataInicio \leq "2009-11-30\ 23:59:59"}(fezPercurso \bowtie motorista))$

Expressão em SQL

SELECT Nome **FROM** fezPercurso **NATURAL INNER JOIN** motorista **WHERE**(CodigoP = 21 **AND** DataInicio **between** '2009-11-01 00:00:00' **AND** '2009-11-30 23:59:59');

Output

nome
Gomes

g)Número de motoristas que fizeram o percurso 23

Expressão em Álgebra Relacional

$Gcount(NBi)(\sigma_{CodigoP = 23}(fezPercurso))$

Expressão em SQL

SELECT DISTINCT count(NBi) FROM fezPercurso **WHERE** (CodigoP = 23);

Output

count
1

h) Número de passageiros que entraram na paragem “Granito” no percurso 21 que se iniciou às 6.20 de 12/11/09

Expressão em Álgebra Relacional

Gcount(*)($\sigma_{\text{CodigoP} = 21 \wedge \text{Paragem} = \text{“Granito”} \wedge \text{DataInicio} = \text{“2009-11-12 06:20:00”}}$ (bilhete \bowtie fezPercurso))

Expressão em SQL

SELECT count(*) FROM bilhete NATURAL INNER JOIN fezPercurso WHERE (Paragem = 'Granito' AND CodigoP = 21 AND DataInicio = '2009-11-12 06:20:00');

Output

count
4

i) Carreira que teve mais passageiros no mês de Novembro de 2009

Expressão em Álgebra Relacional

R \leftarrow CodigoP, DataInicio **Gcount(*)** as total ($\sigma_{\text{DataInicio} \geq \text{“2009-11-01 00:00:00”} \wedge \text{DataInicio} \leq \text{“2009-11-30 23:59:59”}}$ (bilhete \bowtie fezPercurso))

R2 \leftarrow **Gmax**(total) as total (**R**)

$\pi_{\text{CodigoP}, \text{DataInicio}}$ (R \bowtie R2)

Expressão em SQL

```
SELECT Codigo, DataInicio FROM (
  (SELECT max(c) as c FROM
    (SELECT CodigoP, DataInicio, count(*) as c FROM bilhete NATURAL INNER JOIN
    fezPercurso WHERE ( DataInicio BETWEEN '2009-11-01 00:00:00' AND '2009-11-30
    23:59:59') group by CodigoP, DataInicio) as r
  ) as r3
  NATURAL INNER JOIN
  (SELECT CodigoP, DataInicio, count(*) as c FROM bilhete NATURAL INNER JOIN
  fezPercurso WHERE ( DataInicio BETWEEN '2009-11-01 00:00:00' AND '2009-11-30
  23:59:59') group by CodigoP, DataInicio) as r2);
```

Output

codigop	datainicio
23	2009-11-12 11:20:00

j)Autocarros que nunca fizeram o percurso 21

Expressão em Álgebra Relacional

$$\pi_{\text{Matricula}}(\text{autocarro}) - \pi_{\text{Matricula}}(\sigma_{\text{CodigoP} = 21}(\text{autocarro} \bowtie \text{fezPercurso}))$$

Expressão em SQL

```
(SELECT Matricula FROM autocarro) EXCEPT (SELECT Matricula FROM autocarro NATURAL
INNER JOIN fezPercurso WHERE (CodigoP = 21));
```

Output

matricula
PL-21-PO
EW-45-QQ
JT-65-BE
SS-98-KL

k) Motorista que transportou mais passageiros no mês de Novembro de 2009

Expressão em Álgebra Relacional

$R \leftarrow \pi_{NBi}(\sigma_{DataInicio \geq "2009-11-01 00:00:00" \wedge DataInicio \leq "2009-11-30 23:59:59"}(bilhete))$

$R1 \leftarrow \rho_{max}(total) \text{ as total } (R)$

$\pi_{NBi}(R \bowtie R1)$

Expressão em SQL

```
SELECT Nbi FROM ((SELECT max(total) as total FROM (SELECT NBi, count(NBi) as total FROM bilhete WHERE ( DataInicio BETWEEN '2009-11-01 00:00:00' AND '2009-11-30 23:59:59') group by NBi) as r1 ) as r2 NATURAL INNER JOIN (SELECT NBi, count(NBi) as total FROM bilhete WHERE ( DataInicio BETWEEN '2009-11-01 00:00:00' AND '2009-11-30 23:59:59') group by NBi) as r3);
```

Output

nbi
78906543

l) Motorista que fez todos os percursos

Expressão em Álgebra Relacional

$\pi_{NBi, CodigoP}(fezPercurso) \div \pi_{CodigoP}(percurso)$

Expressão em SQL

```
SELECT r.NBi FROM fezPercurso as r WHERE NOT EXISTS ((SELECT CodigoP FROM percurso) EXCEPT (SELECT t.CodigoP FROM fezPercurso as t WHERE (r.NBi=t.NBi)));
```

Output(sem output)

m) Percurso com mais paragens

Expressão em Álgebra Relacional

```
r ← CodigoP Gcount(CodigoP) as p (passoPercurso)  
π CodigoP (r ∞ Gmax(p)(r))
```

Expressão em SQL

```
SELECT CodigoP FROM((SELECT CodigoP,count(CodigoP) as p FROM passoPercurso group  
by CodigoP) as r1 NATURAL INNER JOIN (SELECT max(p) as p FROM (SELECT  
CodigoP,count(CodigoP) as p FROM passoPercurso group by CodigoP) as r3) as r2);
```

Output

codigop
21

n) Quanto a empresa facturou no mês de Novembro se cada passageiro pagasse 50 cêntimos.

Expressão em Álgebra Relacional

```
π passageiros*50 (Gcount(*) as passageiros (σDataInicio ≥ "2009-11-01 00:00:00" ∧  
DataFim ≤ "2009-11-30 23:59:59" (bilhete ∞ fezPercurso)))
```

Expressão em SQL

```
SELECT (passageiros*50) as dinheiro FROM (SELECT count (*) as passageiros FROM (bilhete  
NATURAL INNER JOIN fezPercurso) WHERE (datainicio >= '2009-11-01 00:00:00' AND  
datafim <= '2009-11-30 23:59:59')) as c;
```

Output

dinheiro
2450