



## **Base de Dados**

Fãs de Policiais



Trabalho realizado por:  
Filipe Alfaiate I43315  
José Santos I43017

Professor:  
Irene Rodrigues



## **Introdução e Objetivos**

Pretende-se desenvolver uma base de dados para guardar a informação sobre uma Fãs de Policiais.

Para gerir a companhia é necessário representar os dados sobre: os membros, livros policiais, géneros de livros, amigos e gostos.

Sobre os membros pretende-se registar o nome, um identificador único, o país onde nasceu, a cidade onde vive e a data de nascimento.

Os membros têm amigos, e é na relação amigos que se representa a amizade entre dois membros.

Para cada membro representam-se os livros que leram e gostaram na relação gosta.

A informação sobre os livros da rede é representada na relação livro que representa um livro com o título e o ISBN.

Na relação género representam-se os géneros de cada livro.

A relação autoria tem a informação sobre a autoria das obras.

A relação autor representa a informação sobre cada autor, contendo este, o nome, um código único e o país onde nasceu.

- membro(Nome, IdMemb, Pais, Cidade, DataNasc)
- amigo(IdMemb, IdMemb)
- gosta(IdMemb, ISBN)
- livro(ISBN, Título)
- genero(ISBN, Genero)
- autoria(ISBN, Coda)
- autor(Coda, Nome, Pais)



## **Chaves primarias, Chaves candidatas e Chaves estrangeiras**

**membro(Nome, IdMemb, Pais, Cidade, DataNasc)**

Chave primaria: {IdMemb}

Chave candidata: {IdMemb},

Chave estrangeira: {}

**amigo(IdMemb1, IdMemb2)**

Chave primaria: {IdMemb1, IdMemb2}

Chave candidata: {IdMemb1, IdMemb2}

Chave estrangeira: {IdMembr1, IdMemb2}

**gosta(IdMemb, ISBN)**

Chave primaria: {IdMemb, ISBN}

Chave candidata: {IdMemb, ISBN}

Chave estrangeira: {IdMemb}

**livro(ISBN, Titulo)**

Chave primaria: {ISBN}

Chave candidata: {ISBN}

Chave estrangeira: {}



**genero(ISBN, Genero)**

Chave primaria: {ISBN, Genero}

Chave candidata: {ISBN, Genero}

Chave estrangeira: {ISBN}

**autoria(ISBN, CodA)**

Chave primaria: {ISBN, CodA}

Chave candidata: {ISBN, CodA}

Chave estrangeira: {ISBN, CodA}

**autor(CodA, Nome, Pais)**

Chave primaria: {CodA}

Chave candidata: {CodA}

Chave estrangeira: {}



## **Queries, Inserções e Tabelas**

No ficheiro BD.txt está presente a criação das tabelas com as suas devidas relações entre si. No ficheiro queries.txt estão devidamente identificados os exemplos que são pedidos no exercício 3 para serem introduzidos na Base de Dados. Por fim, também no ficheiro queries.txt estão presentes todas as expressões em SQL que são pedidas no exercício 4.



## Álgebra Relacional e SQL

a) Qual é o nome dos autores de obras do género drama?

**Álgebra Relacional:**

a)

$$\pi_{\text{Nome}} \left( \sigma_{(\text{autor} \bowtie \text{autoria} \bowtie \text{genero})} \text{genero} = \text{'Drama'} \right)$$

**SQL:**

```
select distinct Nome
from autor natural inner join autoria natural inner join genero
where Genero like 'Drama';
```

b) Qual o nome dos membros que gostam de livros da Agatha Christie?

**Álgebra Relacional:**

b)

$$\pi_{\text{membro.nome}} \left( \sigma_{(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor})} \text{autor.nome} = \text{'Agatha Christie'} \right)$$

**SQL:**

```
select distinct membro.Nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Agatha Christie';
```



- c) Qual o nome dos membros que gostam de um livro de um autor que nasceu no seu país?

**Álgebra Relacional:**

$$c) \pi_{\text{membro.nome}} \left( G(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor}) \right) \\ \text{membro.pais} = \text{autor.pais}$$

**SQL:**

```
select distinct membro.nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where membro.pais = autor.pais;
```

- d) Quais os membros que não gostam de algum livro da Agatha Christie?

**Álgebra Relacional:**

$$d) \pi_{\text{Nome}} (G(\text{membro})) - \pi_{\text{membro.nome}} \left( G(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor}) \right) \\ \text{autor.nome} = \text{'Agatha Christie'}$$

**SQL:**

```
select Nome
from membro
except
select distinct membro.Nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Agatha Christie';
```



e) Quais os membros que não são amigos do membro com o IdMemb "oleitor"?

**Álgebra Relacional:**

$$e) \pi_{Nome} (G(membro)) - \pi_{Nome} (G(membro \bowtie_{IdMemb = 'oleitor'} amigo))$$

**SQL:**

```
select Nome
from membro
except
(select Nome
from membro, amigo
where idMemb=idMemb1 and amigo.idMemb2 like 'oleitor'
UNION
select Nome
from membro, amigo
where idMemb=idMemb2 and amigo.idMemb1 like 'oleitor'
UNION
select Nome
from membro
where idmemb like 'oleitor');
```





- f) Qual o nome dos amigos do “oleitor” que são mais jovens que ele?

**Álgebra Relacional:**

$$\begin{aligned} f) \quad r &\leftarrow \pi_{\text{Nome}, \text{IdMemb}} \left( \sigma_{\text{membro} \bowtie \text{amigo}} \left( \sigma_{\text{membro.IdMemb} = \text{'oleitor'}} \right) \right) \\ q &\leftarrow \pi_{*} \left( \sigma_{\text{IdMemb} = \text{'oleitor'}} \left( \sigma_{\text{DataNasc}} \right) \right) \\ \pi_{\text{Nome}} &\left( \sigma_{r \bowtie q} \left( \sigma_{\text{membro.IdMemb} = \text{amigo.IdMemb} \wedge \text{membro.DataNasc} < \text{amigo.DataNasc}} \right) \right) \end{aligned}$$

**SQL:**

```
select Nome from
(select Nome, DataNasc
from membro, amigo
where idMemb=idMemb1 and amigo.idMemb2 like 'oleitor'
UNION
select Nome, DataNasc
from membro, amigo
where idMemb=idMemb2 and amigo.idMemb1 like 'oleitor') as r
where(select DataNasc
from membro
where idMemb like 'oleitor')<r.Datanasc;
```

- g) Qual o nome dos membros que gostam de livros da Agatha Christie e do Francisco José Viegas?

**Álgebra Relacional:**

$$g) \quad \pi_{\text{membro.Nome}} \left( \sigma_{\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor}} \left( \sigma_{\text{autor.nome} = \text{'Agatha Christie'} \wedge \text{autor.nome} = \text{'Francisco Jose Viegas'}} \right) \right)$$



**SQL:**

```
select membro.Nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Francisco Jose Viegas'
INTERSECT
select membro.nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Agatha Christie';
```

**h)** Qual o nome dos membros que gostam de livros da Agatha Christie ou do Francisco José Viegas?

**Álgebra Relacional:**

h)  $\pi_{\text{membro.nome}} \left( \sigma_{\text{autor.nome} = \text{'Agatha Christie'} \vee \text{autor.nome} = \text{'Francisco José Viegas'}} \left( \text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor} \right) \right)$

**SQL:**

```
select membro.Nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Francisco Jose Viegas'
UNION
select membro.nome
from (membro natural inner join gosta) inner join (autoria natural
inner join autor) using (ISBN)
where autor.nome like 'Agatha Christie';
```



i) Quantos amigos tem o membro "oleitor"?

**Álgebra Relacional:**

$$i) \quad \sigma_{\text{count}(\text{idMemb})} \left( \sigma_{\text{membro} \bowtie \text{amigo}} \right)_{\text{membro.idMemb} = 'oleitor'}$$

**SQL:**

```
select count(*) from
(select Nome
from membro, amigo
where idMemb=idMemb1 and amigo.idMemb2 like 'oleitor'
UNION
select Nome
from membro, amigo
where idMemb=idMemb2 and amigo.idMemb1 like 'oleitor') as
r;
```

j) Qual é o membro que tem mais amigos?

**Álgebra Relacional:**

$$d) \quad r \leftarrow \pi_{\text{Nome}} \left( \sigma_{\text{count}(\text{idMemb})} \left( \sigma_{\text{membro} \bowtie \text{amigo}} \right) \right) \text{ as } ca$$
$$\pi_{\text{Nome}} \left( \sigma_{\text{max}(ca)} \text{ as } ca(r) \bowtie r \right)$$

**SQL:**

```
with x as(
select Nome, count(idMemb2) as a
from membro, amigo
where idMemb=idMemb1 group by Nome),
```



```
y as (  
  select Nome, count(idMemb1) as b  
  from membro, amigo  
  where idMemb=idMemb2 group by Nome),  
  
z as(  
  select x.Nome, x.a  
  from x  
  where x.a = (select max(x.a) from x)),  
  
w as(  
  select y.Nome, y.b  
  from y  
  where y.b = (select max(y.b)from y)),  
  
k as(select Nome ,(x.a+y.b) as soma  
  from y inner join x using (nome))  
  
select * from(  
  select z.Nome  
  from z,w,k  
  where z.a>=w.b and z.a>=k.soma  
  UNION  
  select w.nome  
  from z,w,k  
  where w.b>=z.a and w.b>=k.soma  
  UNION  
  select k.nome  
  from z,w,k  
  where k.soma>=z.a and k.soma>=w.b) as r;
```



k) Qual o nome dos membros que são amigos do membro que gosta de mais livros?

**Álgebra Relacional:**

k)  $r \leftarrow \text{IdMemb} \Join_{\text{count(ISBN) as ml (gosta)}}$

$s \leftarrow \text{IdMemb} \Join_{\text{max(ml) (r)}} r$

$\pi_{\text{Nome}} \left( \sigma_{\substack{\text{(membro} \times s \times \text{amigo)} \\ s.\text{IdMemb} = \text{amigo}.\text{IdMemb}}} \right)$

**SQL:**

```
with y as(
select Nome, idMemb2
from membro, amigo
where idMemb=idMemb1),
```

```
x as (
select Nome, idMemb1
from membro, amigo
where idMemb=idMemb2),
```

```
z as (
select IdMemb, count(ISBN) as maisLivros
from gosta group by IdMemb),
```



```
aux as (select z.IdMemb  
from z  
where z.maislivros = (select max(z.maislivros) from z))
```

```
select distinct y.nome  
from y, aux  
where y.idMemb2=aux.idmemb  
UNION  
select distinct x.nome  
from x, aux  
where x.idMemb1=aux.idmemb;
```

I) Para cada livro indique o número de géneros.

**Álgebra Relacional:**

2)  $\text{Titulo } \bowtie \text{ count(gen\u00e9ros) (livro } \bowtie \text{ g\u00e9nero)}$

**SQL:**

```
select Titulo, count(g\u00e9nero)  
from livro natural inner join g\u00e9nero  
group by Titulo;
```



m) Para cada livro indique o número de género e o número de gostos.

**Álgebra Relacional:**

$$m) \quad r \leftarrow \pi_{\text{Titulo}} \sigma_{\text{count}(\text{generos})} \left( \text{livro} \bowtie \text{generos} \right)$$

$$s \leftarrow \pi_{\text{Titulo}} \sigma_{\text{count}(\text{IdMemb})} \left( \text{livro} \bowtie \text{gosta} \right)$$

$$\pi_{\text{Titulo}} (r \cup s)$$

**SQL:**

```
select * from(
(select Titulo, count(genero) as ngeneros
from livro natural inner join genero
group by Titulo) as a natural inner join
(select Titulo, count(IdMemb) as nmembros
from livro natural inner join gosta
group by Titulo) as b);
```



n) Para cada autor indique o número de livros, o número de géneros e o número de gostos.

**Álgebra Relacional:**

n)

$$r \leftarrow \text{Nome } \rho_{\text{count(ISBN)}} \left( \text{autor} \bowtie \text{autoria} \right)$$

$$s \leftarrow \text{Nome } \rho_{\text{count(generos)}} \left( \text{autor} \bowtie \text{autoria} \bowtie \text{genero} \right)$$

$$p \leftarrow \text{Nome } \rho_{\text{count(IdMemb)}} \left( \text{autor} \bowtie \text{autoria} \bowtie \text{gosta} \right)$$

$$\pi_{\text{Nome}} \left( r \bowtie s \bowtie p \right)$$

$r.\text{Nome} = s.\text{nome} \wedge r.\text{nome} = p.\text{nome}$

**SQL:**

```
select * from(
(select Nome, count(ISBN) as nlivros
from autor natural inner join autoria
group by Nome) as a natural inner join
(select Nome, count(genero) as ngeneros
from autor natural inner join autoria natural inner join genero
group by Nome) as b natural inner join
(select Nome, count(IdMemb) as ngostos
from autor natural inner join autoria natural inner join gosta
group by Nome) as c);
```





- o) Para cada membro, nome, indique o número de amigos e o número de livros de que gosta.

**Álgebra Relacional:**

o)

$$r \leftarrow \text{Nome} \underset{\text{as } ca}{\text{g count}} (\text{amigo.idMemb}) \left( \text{membro} \bowtie \text{amigo} \right)$$

$$s \leftarrow \text{Nome} \underset{\text{as } cl}{\text{g count}} (\text{ISBN}) \left( \text{membro} \bowtie \text{gosta} \right)$$

$$\pi_{\text{Nome}, ca, cl} \left( \begin{array}{l} \text{membro} \times r \times s \\ \text{membro.nome} = r.\text{nome} \wedge \text{membro.nome} = s.\text{nome} \end{array} \right)$$

**SQL:**

with tudo as (with x as(  
select Nome, count(idMemb2) as a  
from membro, amigo  
where idMemb=idMemb1 group by Nome),

y as (  
select Nome, count(idMemb1) as a  
from membro, amigo  
where idMemb=idMemb2 group by Nome),

k as(select Nome ,(x.a+y.a) as a  
from y join x using (nome)),

aux1 as(select Nome  
from x



except

select nome

from k),

aux2 as (select Nome

from y

except

select nome

from k)

select \* from(

select Nome,a

from x natural inner join aux1

UNION

select Nome,a

from y natural inner join aux2

UNION select Nome,a

from k) as r)

select Nome, tudo.a as nAmigos, count(ISBN) as nGostos

from tudo natural inner join membro natural inner join gosta  
group by (Nome, tudo.a);



p) Qual o nome dos membros que são amigos de todos os membros?

**Álgebra Relacional:**

$$P) \quad r \leftarrow \pi_{IdMemb} (membro)$$

$$\pi_{Nome} (membro \bowtie amigo) \div r$$

**SQL:**

with nMemb as(

select count(IdMemb)-1 as c

from membro),

tudo as (with x as(

select Nome, count(idMemb2) as a

from membro, amigo

where idMemb=idMemb1 group by Nome),

y as (

select Nome, count(idMemb1) as b

from membro, amigo

where idMemb=idMemb2 group by Nome),

z as(

select x.Nome, x.a



```
from x
where x.a = (select max(x.a) from x)),

w as(
select y.Nome, y.b
from y
where y.b = (select max(y.b)from y)),

k as(select Nome ,(x.a+y.b) as soma
from y inner join x using (nome))

select * from(
select z.Nome, z.a as ola
from z,w,k
where z.a>=w.b and z.a>=k.soma
UNION
select w.nome, w.b as ola
from z,w,k
where w.b>=z.a and w.b>=k.soma
UNION
select k.nome,k.soma as ola
from z,w,k
where k.soma>=w.b and k.soma>=z.a) as r)

select Nome
from tudo, Nmemb
where tudo.ola=Nmemb.c;
```



q) Quais os títulos dos livros de que todos os amigos do leitor gostam?

**Álgebra Relacional:**

$$9) \quad r \leftarrow \pi_{IdMem} \left( \begin{array}{l} \text{membro} \bowtie \text{amigo} \\ \text{membro.IdMembro} = \text{'oleitor'} \end{array} \right)$$

$$s \leftarrow \pi_{ISBN} (r \bowtie \text{gosta})$$

$$\pi_{Titulo} (\text{livro}) \div s$$

**SQL:**

```
with x as (  
  select idMemb  
  from membro, amigo  
  where idMemb=idMemb1 and amigo.idMemb2 like 'oleitor'  
  UNION  
  select idMemb  
  from membro, amigo  
  where idMemb=idMemb2 and amigo.idMemb1 like 'oleitor')  
  
select distinct Titulo  
from gosta natural inner join livro natural inner join x;
```