

---

## Base de dados

---

Diogo Fernandes (55466)

Tomás Mateus (55891)

**2024**

## Contents

1	Enunciado	2
2	Exercício 1/ Resolução	3
3	Exercício 2 / Resolução	6
4	Exercício 3 / Resolução	8
5	Exercício 4 / Resolução	15

# 1 Enunciado

**Pretende-se desenvolver uma base de dados** para gerir a informação de uma rede social de Fãs de Literatura. Para gerir a rede é necessário representar os dados dos membros, dos livros, dos géneros de livros, dos amigos e dos gostos. Para os membros deve-se registar: o nome, um identificador único para cada membro, 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, esta relação não deve conter tuplos simétricos, se o membro oleitor é amigo do membro odetective na relação amigos ou existe o tuplo (oleitor,odetective) ou o tuplo (odetective,oleitor ). Na relação 'gosta' registam-se os livros que os membros leram e gostaram. A informação sobre os livros da rede é representada na relação livro que representa um livro com o título e o isbn. Na relação genero representam-se os generos (policial, drama, romance, ...) de cada livro. Todos os livros têm um ou mais géneros. A relação autoria tem a informação sobre a autoria das obras, um obra (livro) tem um ou mais autores). A relação autor representa a informação sobre cada autor: o nome, um código único para cada autor e o país onde nasceu.

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

amigo(IdMemb, IdMemb)

gosta(IdMemb,ISBN)

livro(ISBN,Titulo)

genero(ISBN,Genero)

autoria(ISBN,Coda)

autor(Coda,Nome,Pais)

## 2 Exercício 1/ Resolução

**Enunciado:** Indique as chaves primárias, candidatas e estrangeiras de cada relação.

**Resolução:**

**Relação Membro:**

**Super chave:**  $\{\text{IdMmeb}\}$ ,  $\{\text{IdMmemb}, \text{Nome}\}$ ,  $\{\text{IdMmeb}, \text{Pais}\}$ ;  $\{\text{IdMemb}, \text{Cidade}\}$ ,  $\{\text{IdMemb}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{Pais}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{Cidade}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Pais}, \text{Cidade}\}$ ,  $\{\text{IdMemb}, \text{Pais}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Cidade}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{Pais}, \text{Cidade}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{Pais}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Pais}, \text{Cidade}, \text{DataNasc}\}$ ,  $\{\text{IdMemb}, \text{Nome}, \text{Pais}, \text{Cidade}, \text{DataNasc}\}$ ;

**Chave Primária:**  $\{\text{IdMemb}\}$ ;

**Chave Candidata:**  $\{\text{IdMemb}\}$ ;

**Chave Estrangeira:** Nenhuma;

**Relação Amigo:**

**Super chave:**  $\{\text{IdMemb1}, \text{IdMemb2}\}$ ;

**Chave Primária:**  $\{\text{IdMemb1}, \text{IdMemb2}\}$ ;

**Chave Candidata:** {IdMemb1, IdMemb2};

**Chave Estrangeira:** {IdMemb1, IdMemb2} com referência a relação membro(IdMemb);

**Relação Gosta:**

**Super Chave:** {IdMemb, ISBN};

**Chave Primária:** {IdMemb, ISBN};

**Chave Candidata:** {IdMemb, ISBN};

**Chave Estrangeira:** {IdMemb} com referência a relação membro(IdMemb), ISBN com referência a relação livro(ISBN);

**Relação Livro:**

**Super Chave:** {ISBN}, {ISBN,Titulo};

**Chave Primária:** {ISBN};

**Chave Candidata:** {ISBN};

**Chave Estrangeira:** Nenhuma;

**Relação Genero:**

**Super Chave:** {ISBN}, {ISBN,Genero};

**Chave Primária:** {ISBN};

**Chave Candidata:** {ISBN};

**Chave Estrangeira:** {ISBN} com referência a relação Livro(ISBN);

**Relação Autoria:**

**Super Chave:** {ISBN}, {ISBN, CodA};

**Chave Primária:** {ISBN};

**Chave Candidata:** {ISBN};

**Chave Estrangeira:** {ISBN} com referência a relação Livro(ISBN);

**Relação Autor:**

**Super Chave:** {CodA};

**Chave Primária:** {CodA};

**Chave Candidata:** {CodA};

**Chave Estrangeira:** {CodA} com referência a Autoria(CodA);

### 3 Exercício 2 / Resolução

**Enunciado:** Indique os comandos SQL para a criação das tabelas que constituem esta base de dados. E construa esta base de dados no PostGres.

**Resolução:**

```
CREATE TABLE Membro (  
  IdMemb VARCHAR(50) PRIMARY KEY,  
  Nome VARCHAR(30) NOT NULL,  
  Pais VARCHAR(50),  
  Cidade VARCHAR(50),  
  DataNasc DATE);
```

```
CREATE TABLE Amigo (  
  IdMemb1 VARCHAR(50),  
  IdMemb2 VARCHAR(50),  
  PRIMARY KEY (IdMemb1, IdMemb2),
```

```
FOREIGN KEY (IdMemb1) REFERENCES Membro(IdMemb),  
FOREIGN KEY (IdMemb2) REFERENCES Membro(IdMemb));
```

```
CREATE TABLE Livro (  
ISBN VARCHAR(3) PRIMARY KEY,  
Titulo VARCHAR(50) NOT NULL);
```

```
CREATE TABLE Gosta (  
IdMemb VARCHAR(50),  
ISBN VARCHAR(3),  
PRIMARY KEY (IdMemb, ISBN),  
FOREIGN KEY (IdMemb) REFERENCES Membro(IdMemb),  
FOREIGN KEY (ISBN) REFERENCES Livro(ISBN));
```

```
CREATE TABLE Genero (  
ISBN VARCHAR(3),  
Genero VARCHAR(25),  
PRIMARY KEY (ISBN, Genero),  
FOREIGN KEY (ISBN) REFERENCES Livro(ISBN));
```

```
CREATE TABLE Autoria (  
ISBN VARCHAR(3),
```



```
CodA VARCHAR(5),  
PRIMARY KEY (ISBN, CodA));
```

```
CREATE TABLE Autor (  
CodA VARCHAR(5) PRIMARY KEY,  
Nome VARCHAR(100) NOT NULL,  
Pais VARCHAR(50));
```

## 4 Exercício 3 / Resolução

**Enunciado:** Indique as expressões em SQL para inserir a seguinte informação na sua base de dados.

- (a) 20 membros (coloque valores adequados nos atributos)
- (b) 10 livros e 5 autores. Cada autor deve ter pelo menos 2 livros. Um dos livros deve ter pelo menos 3 autores e outro 2 autores. Todos os livros devem ter um ou mais g´eneros (drama, romance, ...).
- (c) 1 membro deve ser amigo de todos os outros membros. 5 membros devem ter pelo menos 3 amigos.
- (d) todos os membro devem gostar de pelo menos 1 livro. Pelo menos 1 membro deve gostar de todos os livros de um autor.
- (e) Em todas as perguntas abaixo o resultado deve ter no m´inimo um tuplo. Para isso, dever´a inserir mais valores nas suas tabelas de forma a garantir que a resposta nˆao ´e vazia.

### Resolução

a)

```
INSERT INTO Membro VALUES('oleitor', 'Ant´onio', 'Portugal', '´Evora', '1985-01-01');
```

```
INSERT INTO Membro VALUES('odetative', 'Barão', 'Portugal', 'Évora', '1983-10-05');
INSERT INTO Membro VALUES('paginasemfolha', 'Charlotte', 'Reino Unido', 'Londres', '1996-01-01');
INSERT INTO Membro VALUES('bibliofilo', 'Diogo', 'Portugal', 'Amora', '1997-09-18');
INSERT INTO Membro VALUES('capitulomaniaco', 'Enoque', 'Angola', 'Luanda', '2001-04-25');
INSERT INTO Membro VALUES('resenhistaapaixonado', 'Francisco', 'Portugal', 'Tomar', '1976-01-21');
INSERT INTO Membro VALUES('estantecheia', 'Gustavo', 'Brasil', 'Porto Alegre', '1996-11-09');
INSERT INTO Membro VALUES('amantedahistoria', 'Irene', 'Portugal', 'Elvas', '1968-11-01');
INSERT INTO Membro VALUES('versolivre', 'Joaquim', 'Portugal', 'Chaves', '2002-06-21');
INSERT INTO Membro VALUES('poesiaemocional', 'Lígia', 'Portugal', 'Évora', '1972-11-25');
INSERT INTO Membro VALUES('dicionariodoconhecimento', 'Manuel', 'Portugal', 'Évora', '1955-10-11');
INSERT INTO Membro VALUES('contodeencantar', 'Nádia', 'Cabo Verde', 'Praia', '1992-11-16');
INSERT INTO Membro VALUES('encadernacaoartistica', 'Oscar', 'Portugal', 'Leiria', '1986-03-22');
INSERT INTO Membro VALUES('ilustracoescoloridas', 'Paulo', 'Portugal', 'Évora', '2010-11-22');
INSERT INTO Membro VALUES('revisaoatenta', 'Ricardo', 'Portugal', 'Évora', '1985-05-16');
INSERT INTO Membro VALUES('coleccionadordepalavras', 'Saul', 'Portugal', 'Évora', '1967-07-16');
INSERT INTO Membro VALUES('literaturaeterna', 'Tony', 'Portugal', 'Évora', '1996-01-18');
INSERT INTO Membro VALUES('volumenoinfinito', 'Vasco', 'Portugal', 'Évora', '1968-04-16');
INSERT INTO Membro VALUES('prateleiraimpecavel', 'Wesley', 'Brasil', 'Florianópolis', '1979-
```

07-20');

INSERT INTO Membro VALUES('aficionadodaleitura', 'Zanfir', 'Roménia', 'Craiova', '1997-08-10');

b)

INSERT INTO livro VALUES('000', 'Dom Quixote');

INSERT INTO livro VALUES('001', '1984');

INSERT INTO livro VALUES('002', 'Terceira Guerra Mundial');

INSERT INTO livro VALUES('003', 'Diário de um Estudante');

INSERT INTO livro VALUES('004', 'Moby Dick');

INSERT INTO livro VALUES('005', 'Aventuras de Tintin');

INSERT INTO livro VALUES('006', 'Serviços Secretos');

INSERT INTO livro VALUES('007', 'Romeu e Julieta ');

INSERT INTO livro VALUES('008', 'Vida Selvagem');

INSERT INTO livro VALUES('009', 'O Íncio do Fim');

INSERT INTO autor VALUES('12345', 'Agatha Christie', 'Reino Unido');

INSERT INTO autor VALUES('12346', 'Francisco José Viegas', 'Portugal');

INSERT INTO autor VALUES('12347', 'George Orwell', 'Reino Unido');

INSERT INTO autor VALUES('12348', 'Fernando Pessoa', 'Portugal');

INSERT INTO autor VALUES('12349', 'Herman Melville', 'Estados Unidos da América');

INSERT INTO autoria VALUES('000', '12345');

INSERT INTO autoria VALUES('000', '12346');

INSERT INTO autoria VALUES('000', '12348');

INSERT INTO autoria VALUES('001', '12345');

```
INSERT INTO autoria VALUES('001', '12349');
INSERT INTO autoria VALUES('002', '12346');
INSERT INTO autoria VALUES('003', '12346');
INSERT INTO autoria VALUES('004', '12347');
INSERT INTO autoria VALUES('005', '12347');
INSERT INTO autoria VALUES('006', '12348');
INSERT INTO autoria VALUES('007', '12348');
INSERT INTO autoria VALUES('008', '12349');
INSERT INTO autoria VALUES('009', '12349');
```

```
INSERT INTO genero VALUES('000', 'Romance');
INSERT INTO genero VALUES('001', 'Ficção Científica');
INSERT INTO genero VALUES('002', 'Guerra');
INSERT INTO genero VALUES('003', 'Biografia');
INSERT INTO genero VALUES('004', 'Romance');
INSERT INTO genero VALUES('005', 'Aventura');
INSERT INTO genero VALUES('006', 'Ficção Científica');
INSERT INTO genero VALUES('007', 'Tragédia');
INSERT INTO genero VALUES('007', 'Drama');
INSERT INTO genero VALUES('008', 'Ficção');
INSERT INTO genero VALUES('009', 'Ficção');
INSERT INTO genero VALUES('009', 'Drama');
```

c)

```
INSERT INTO amigo VALUES('odetative', 'oleitor');
INSERT INTO amigo VALUES('odetative', 'paginasemfolha');
INSERT INTO amigo VALUES('odetative', 'bibliofilo');
```

```

INSERT INTO amigo VALUES('odetative', 'capitulomaniaco');
INSERT INTO amigo VALUES('odetative', 'resenhista paixonado');
INSERT INTO amigo VALUES('odetative', 'estantecheia');
INSERT INTO amigo VALUES('odetative', 'amante da historia');
INSERT INTO amigo VALUES('odetative', 'versolivre');
INSERT INTO amigo VALUES('odetative', 'poesia emocional');
INSERT INTO amigo VALUES('odetative', 'dicionário do conhecimento');
INSERT INTO amigo VALUES('odetative', 'contos de encantar');
INSERT INTO amigo VALUES('odetative', 'encadernação artística');
INSERT INTO amigo VALUES('odetative', 'ilustrações coloridas');
INSERT INTO amigo VALUES('odetative', 'revisão atenta');
INSERT INTO amigo VALUES('odetative', 'coleccionador de palavras');
INSERT INTO amigo VALUES('odetative', 'literatura eterna');
INSERT INTO amigo VALUES('odetative', 'volumen infinito');
INSERT INTO amigo VALUES('odetative', 'prateleira impecável');
INSERT INTO amigo VALUES('odetative', 'aficionado da leitura');

```

— todos os amigos do leitor

```

INSERT INTO amigo VALUES('leitor', 'páginas em folha');
INSERT INTO amigo VALUES('leitor', 'bibliófilo');
INSERT INTO amigo VALUES('leitor', 'capitulomaniaco');
INSERT INTO amigo VALUES('leitor', 'versolivre');
INSERT INTO amigo VALUES('leitor', 'poesia emocional');
INSERT INTO amigo VALUES('leitor', 'dicionário do conhecimento');
INSERT INTO amigo VALUES('leitor', 'contos de encantar');
INSERT INTO amigo VALUES('leitor', 'literatura eterna');
INSERT INTO amigo VALUES('leitor', 'volumen infinito');
INSERT INTO amigo VALUES('leitor', 'prateleira impecável');

```

INSERT INTO amigo VALUES('oleitor', 'aficcionadodaleitura');

INSERT INTO amigo VALUES('versolivre', 'paginasemfolha');

INSERT INTO amigo VALUES('versolivre', 'bibliofilo');

INSERT INTO amigo VALUES('versolivre', 'capitulomaniaco');

INSERT INTO amigo VALUES('versolivre', 'resenhistaapaixonado');

INSERT INTO amigo VALUES('versolivre', 'estantecheia');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'ilustracoescoloridas');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'revisaoatenta');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'coleccionadordepalavras');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'poesiaemocional');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'literaturaeterna');

INSERT INTO amigo VALUES('resenhistaapaixonado', 'volumenoinfinito');

INSERT INTO amigo VALUES('bibliofilo', 'capitulomaniaco');

INSERT INTO amigo VALUES('bibliofilo', 'resenhistaapaixonado');

INSERT INTO amigo VALUES('bibliofilo', 'estantecheia');

INSERT INTO amigo VALUES('bibliofilo', 'amantedahistoria');

d)

INSERT INTO gosta (IdMemb, ISBN) VALUES

('oleitor', '000'),

('oleitor', '001'),

('oleitor', '002'),

('oleitor', '003'),

('oleitor', '004'),

('oleitor', '005'),

('oleitor', '006'),

('oleitor', '007'),

('oleitor', '008'),

('oleitor', '009'),

('versolivre', '000'),

('versolivre', '001'),

('versolivre', '002'),

('versolivre', '003'),

('versolivre', '004'),

('versolivre', '005'),

('versolivre', '006'),

('versolivre', '007'),

('versolivre', '008'),

('versolivre', '009'),

–restantes

('odetative', '005'),

('paginasemfolha', '003'),

('bibliofilo', '002'),

('capitulomaniaco', '000'),

('resenhista paixonado', '009'),

('estantecheia', '008'),

('amantedahistoria', '001'),

('poesiaemocional', '007'),

('dicionariodoconhecimento', '008'),

('contodeencantar', '009'),

('encadernacaoartistica', '006'),

('ilustracoes coloridas', '003'),

('revisaoatenta', '004'),  
( 'coleccionadordepalavras', '006'),  
( 'literaturaeterna', '007'),  
( 'volumenoinfinito', '006'),  
( 'prateleiraimpecavel', '007'),  
( 'aficcionadodaleitura', '004'),  
( 'dicionariodoconhecimento', '005'),  
( 'contodeencantar', '002'),  
( 'encadernacaoartistica', '001'),  
( 'ilustracoescoloridas', '000'),  
( 'revisaoatenta', '006'),  
( 'coleccionadordepalavras', '009'),  
'literaturaeterna', '004'),  
( 'volumenoinfinito', '003'),  
( 'prateleiraimpecavel', '002'),  
( 'aficcionadodaleitura', '001');

## 5 Exercício 4 / Resolução

**Enunciado:** Indique a expressão em SQL e em Algebra Relacional para responder às seguintes perguntas:

a)

Qual é o nome dos autores de obras do género drama? (se não inseriu o género drama, insira-o e insira também este género em pelo menos uma obra)

**Álgebra Relacional:**

$\pi_{\text{autor.Nome}} (\sigma_{\text{genero.Genero}=\text{Drama} \wedge \text{autor.CodA}=\text{autoria.CodA} \wedge \text{genero.ISBN}=\text{autoria.ISBN}} (\text{genero} \times \text{autoria} \times \text{autor}))$



**SQL:**

```
SELECT DISTINCT autor.Nome
FROM genero, autoria, autor
WHERE genero.Genero = 'Drama' AND
autor.CodA = autoria.CodA AND
genero.ISBN = autoria.ISBN;
```

b)

Qual o nome dos membros que gostam de livros da Agatha Christie? (se não inseriu a autora insira e insira também pelo menos um gosto de um membro numa obra da autora)

**Álgebra Relacional:**

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{nome.autor} = \text{Agatha Christie} \wedge \\ \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ \text{autor.coda} = \text{autoria.coda} \end{array}} (\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor}) \right)$$

**SQL:**

```
SELECT DISTINCT Membro.nome FROM Membro, Gosta, Autoria, Autor
WHERE Autor.nome = 'Agatha Christie'
AND Gosta.IdMemb = Membro.IdMemb AND
Gosta.ISBN = Autoria.ISBN AND
```

Autor.codA = Autoria.codA

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

**Álgebra Relacional:**

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{membro.pais} = \text{autor.pais} \wedge \\ \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ \text{autor.codA} = \text{autoria.codA} \end{array}} (\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor}) \right)$$

**SQL:**

```
SELECT DISTINCT Membro.nome FROM Gosta, Membro, Autoria, Autor
WHERE Membro.pais = Autor.pais AND
Gosta.IdMemb = Membro.IdMemb AND
Gosta.ISBN = Autoria.ISBN
AND Autoria.codA = Autor.codA
```

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

**Álgebra Relacional:**

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

-

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{autor.nome} = \text{Agatha Christie} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ \text{autor.coda} = \text{autoria.coda} \end{array}} (\text{gosta} \times \text{membro} \times \text{autor} \times \text{autoria}) \right)$$

**SQL:**

```
SELECT DISTINCT Membro.nome FROM Gosta, Membro
WHERE Gosta.IdMemb = Membro.IdMemb
EXCEPT
SELECT DISTINCT Membro.nome FROM Gosta, Membro, Autor, Autoria
WHERE Gosta.IdMemb = Membro.IdMemb AND
Autor.nome = 'Agatha Christie' AND Gosta.ISBN = Autoria.ISBN
AND Autoria.coda = Autor.coda
```

e) Quais os membros que não são amigos do membro com o idMemb oleitor? (se o membro não existe insira-o e coloque-lhe 2 amigos)

**Álgebra Relacional:**

$$\pi_{\text{membro.Nome}} \left( \sigma_{\text{membro.IdMemb} \neq \text{oleitor}} (\text{membro}) \right)$$

-

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} (\text{membro.IdMemb} = \text{amigo.IdMemb2} \wedge \\ \text{amigo.IdMemb1} = \text{oleitor}) \vee \\ (\text{membro.IdMemb} = \text{amigo.IdMemb1} \wedge \\ \text{amigo.IdMemb2} = \text{oleitor}) \end{array}} (\text{membro} \times \text{amigo}) \right)$$

**SQL:**

```
SELECT Membro.nome FROM Membro
```

```
WHERE Membro.IdMemb != 'oleitor'
```

```
EXCEPT
```

```
SELECT Membro.nome FROM Membro, Amigo
```

```
WHERE Membro.IdMemb = Amigo.IdMemb2 AND
```

```
Amigo.IdMemb1 = 'oleitor' OR
```

```
Membro.IdMemb = Amigo.IdMemb1 AND
```

```
Amigo.IdMemb2 = 'oleitor';
```

f)

Qual o nome dos amigos do oleitor que são mais jovens que ele?

**Álgebra Relacional:**

$$\pi_{\text{Membro2.Nome}} \left( \sigma_{\text{Membro1.DataNasc} > \text{Membro2.DataNasc}} \left( \text{Amigo} \bowtie_{\substack{\text{Amigo.Membro1}=\text{Membro1.IdMemb} \\ \text{Amigo.Membro2}=\text{Membro2.IdMemb}}} \text{Membro} \times \text{Membro} \right) \right)$$

**SQL:**

```
SELECT membro.Nome AS NomeAmigo
FROM Membro
JOIN Amigo ON membro.IdMemb = amigo.IdMemb2
WHERE amigo.IdMemb1 = 'oleitor' AND
membro.DataNasc > (SELECT DataNasc FROM Membro WHERE IdMemb = 'oleitor');
```

g)

Qual o nome dos membros que gostam de livros da Agatha Christie e do Francisco José Viegas?  
(ajuste os dados na sua base de dados para que exista pelo menos um membro a gostar dos dois livros)

**Álgebra Relacional:**

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{autor.nome} = \text{Agatha Christie} \wedge \\ \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ \text{autor.codA} = \text{autoria.codA} \end{array}} (\text{membro} \times \text{gosta} \times \text{autor} \times \text{autoria}) \right)$$

 $\cap$

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{autor.nome} = \text{Francisco José Viegas} \wedge \\ \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ \text{autor.codA} = \text{autoria.codA} \end{array}} (\text{membro} \times \text{gosta} \times \text{autor} \times \text{autoria}) \right)$$

**SQL:**

```
SELECT Membro.nome FROM Membro, Gosta, Autor, Autoria
WHERE Autor.nome = 'Agatha Christie' AND Gosta.IdMemb = Membro.IdMemb AND
Gosta.ISBN = Autoria.ISBN AND Autor.CodA = Autoria.CodA
INTERSECT
SELECT Membro.nome FROM Membro, Gosta, Autor, Autoria
WHERE Autor.nome = 'Francisco José Viegas' AND
Gosta.IdMemb = Membro.IdMemb AND Gosta.ISBN = Autoria.ISBN AND
Autor.CodA = Autoria.CodA;
```

h)

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

$$\pi_{\text{membro.Nome}} \left( \sigma_{\begin{array}{l} \text{gosta.IdMemb} = \text{membro.IdMemb} \wedge \\ \text{autor.CodA} = \text{autoria.CodA} \wedge \\ \text{gosta.ISBN} = \text{autoria.ISBN} \wedge \\ (\text{autor.nome} = \text{'Agatha Christie'} \vee \\ \text{autor.nome} = \text{'Francisco José Viegas'}) \end{array}} (\text{gosta} \times \text{autor} \times \text{autoria} \times \text{membro}) \right)$$

**SQL:**

```
SELECT DISTINCT Membro.nome FROM Gosta, Autor, Autoria, Membro
WHERE Gosta.IdMemb = Membro.IdMemb AND
Autor.CodA = Autoria.CodA and
Gosta.ISBN = Autoria.ISBN AND
Autor.nome = 'Agatha Christie' OR Autor.nome = 'Francisco José Viegas';
```

i)

Quantos amigos tem o membro oleitor?

$(\text{Amigo} \times \text{Membro}) \rightarrow x$

$y \leftarrow \sigma ((\text{Membro.IdMemb} = \text{Amigo.IdMemb1} \wedge \text{Amigo.IdMemb2} = \text{'oleitor'}) \vee (\text{Membro.IdMemb} = \text{Amigo.IdMemb1} = \text{'oleitor'}(x)))$

$\mathcal{G}_{count}(y)$

**SQL:**

```
SELECT COUNT(Membro.nome) AS Amigosoleitor
```

```
FROM Amigo, Membro
WHERE (Membro.IdMemb = Amigo.IdMemb1 AND Amigo.IdMemb2 = 'oleitor') OR
(Membro.IdMemb = Amigo.IdMemb2 AND Amigo.IdMemb1 = 'oleitor');
```

j)

Qual é o membro que tem mais amigos?

**Álgebra Relacional:**

$$\pi_{\text{Membro.IdMemb1}} (\sigma_{\text{Membro.IdMemb1} = \text{Membro.IdMemb2}} (\mathcal{G}_{count(*)} (\text{Amigo}) \bowtie \text{Membro} \times \text{Membro}))$$

**SQL:**

```
SELECT membro.IdMemb, membro.Nome,
(SELECT COUNT(*)
FROM Amigo
WHERE amigo.IdMemb1 = membro.IdMemb) AS QuantidadeDeAmigos
FROM Membro
ORDER BY QuantidadeDeAmigos DESC;
```

k)

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

$$\pi_{\text{Membro2.Nome}} \left( \text{Amigo} \bowtie_{\substack{\text{Amigo.Membro1} = \text{Gosta.IdMemb} \\ \text{Gosta.ISBN} = \text{Livro.ISBN}}} \text{Gosta} \bowtie_{\text{Gosta.IdMemb} = \text{Membro1.IdMemb}} \text{Membro} \times \text{Membro} \right)$$

**SQL:**

```
SELECT DISTINCT M2.Nome AS NomeAmigo
FROM Amigo, Membro M1, Membro M2
WHERE amigo.IdMemb1 = M1.IdMemb
AND amigo.IdMemb2 = M2.IdMemb
AND amigo.IdMemb1 = (
SELECT IdMemb FROM gosta
```



```
GROUP BY IdMemb  
ORDER BY COUNT(DISTINCT ISBN) DESC  
LIMIT 1);
```

l)

Para cada livro indique o número de géneros.

### Álgebra Relacional:

$$x \leftarrow \pi_{\text{Genero.genero}}(\text{livro} \bowtie \text{genero})$$
$$\text{livro.ISBN, livro.Titulo } \mathcal{G}_{\text{count}}(x)$$

### SQL:

```
SELECT livro.ISBN AS ISBNLivro, livro.Titulo, COUNT(DISTINCT genero.Genero)  
AS NumeroDeGeneros  
FROM livro  
JOIN genero ON livro.ISBN = genero.ISBN  
GROUP BY livro.ISBN, livro.Titulo;
```

m)

Para cada livro indique o número de géneros e o número de gostos.

### Álgebra Relacional:

$$\pi_{\text{livro.ISBN, livro.Titulo, COUNT(DISTINCT Genero.genero) AS NumeroDeGeneros}}$$
$$\text{COUNT(DISTINCT gosta.IdMemb) AS NumeroDeGostos}(\text{livro} \bowtie \text{genero} \bowtie \text{gosta} \bowtie)$$

### SQL:

```
SELECT livro.ISBN AS ISBNLivro, livro.Titulo,  
COUNT(DISTINCT genero.Genero) AS NumeroDeGeneros,  
COUNT(DISTINCT gosta.IdMemb) AS NumeroDeGostos  
FROM livro  
JOIN genero ON livro.ISBN = genero.ISBN
```

```
JOIN gosta ON livro.ISBN = gosta.ISBN
GROUP BY livro.ISBN, livro.Titulo;
```

n)

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

### Álgebra Relacional:

$$x \leftarrow \pi_{\text{autoria.ISBN}}(\text{autoria} \bowtie \text{autor} \bowtie \text{livro} \bowtie \text{genero} \bowtie \text{gosta})$$

autor.CodA, autor.nome  $\mathcal{G}_{\text{count}}(x)$  AS Numlivros

$$y \leftarrow \pi_{\text{Genero.genero}}(\text{autoria} \bowtie \text{autor} \bowtie \text{livro} \bowtie \text{genero} \bowtie \text{gosta})$$

autor.CodA, autor.nome  $\mathcal{G}_{\text{count}}(y)$  AS Numlivros

$$z \leftarrow \pi_{\text{gosta.ISBN}}(\text{autoria} \bowtie \text{autor} \bowtie \text{livro} \bowtie \text{genero} \bowtie \text{gosta})$$

autor.CodA, autor.nome  $\mathcal{G}_{\text{count}}(z)$  AS Numgosto

### SQL:

```
SELECT autor.Coda AS CodigoAutor, autor.Nome AS NomeAutor,
COUNT(DISTINCT autoria.ISBN) AS NumeroDeLivros,
COUNT(DISTINCT genero.Genero) AS NumeroDeGeneros,
COUNT(DISTINCT gosta.ISBN) AS NumeroDeGostos
FROM autor
JOIN autoria ON autor.Coda = autoria.Coda
JOIN livro ON autoria.ISBN = livro.ISBN
JOIN genero ON livro.ISBN = genero.ISBN
JOIN gosta ON livro.ISBN = gosta.ISBN
GROUP BY autor.Coda, autor.Nome;
```

o)

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

### Álgebra Relacional:

$$x \leftarrow \pi_{\text{membro.nome}}(\sigma(\text{IdMemb} = \text{IdMemb1} \text{ or } \text{IdMemb} = \text{IdMemb2}) (\text{membro} \bowtie \text{amigo} \bowtie$$

gosta))

membro.nome  $\mathcal{G}\text{count}(x)$  AS QT.Amigos

membro.nome  $\mathcal{G}\text{count}(y)$  AS QT.Gostos

$y \leftarrow \pi_{\text{DISTINCT}} \text{gosta.ISBN}(\sigma(\text{IdMemb} = \text{idMemb1} \text{ or } \text{IdMemb} = \text{IdMemb2} \text{ (membro} \bowtie \text{ amigo} \bowtie \text{gosta))})$

### SQL:

```
SELECT Membro.nome,
(SELECT COUNT(*) FROM Amigo
WHERE Amigo.idmemb1 = Membro.idmemb OR
Amigo.idmemb2 = Membro.idmemb) AS Qtd_Amigos,
(SELECT COUNT(DISTINCT Gosta.ISBN)
FROM Gosta WHERE Gosta.idmemb = Membro.idmemb) AS Qtd_Livros
FROM Membro;
```

p)

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

### Álgebra Relacional:

$$\pi_{\text{Membro1.IdMemb}} (\text{Membro} - (\text{Membro} \bowtie \text{Amigo}))$$

### SQL:

```
SELECT M1.Nome
FROM Membro M1 WHERE NOT EXISTS (
SELECT M2.IdMemb FROM Membro M2
WHERE NOT (M2.IdMemb = M1.IdMemb) AND NOT EXISTS (
SELECT * FROM Amigo A
WHERE A.IdMemb1 = M1.IdMemb
```

AND A.IdMemb2 = M2.IdMemb));

q)

Quais os titulos dos livros de que todos os amigos do leitor gostam?

**Álgebra Relacional:**

$$\begin{aligned} & \pi_{\text{Livro.Titulo}} (\sigma_{\text{Gosta.IdMemb} = \text{'leitor'}} (\text{Gosta}) \\ & - \left( \pi_{\text{Livro.Titulo}} \left( \text{Gosta} \bowtie_{\substack{\text{Gosta.IdMemb} = \text{Amigo.IdMemb2} \\ \text{Amigo.IdMemb1} = \text{'leitor'}}}} \text{Amigo} \right) \right. \\ & \left. \bowtie_{\text{Gosta.ISBN} = \text{Livro.ISBN}} \text{Livro} \right) \end{aligned}$$

**SQL:**

```
SELECT livro.titulo
FROM Livro
WHERE 'oleitor' NOT IN (
SELECT amigo.idmemb2
FROM amigo
WHERE amigo.IdMemb1 = 'oleitor'
EXCEPT
SELECT gosta.idmemb
FROM Gosta
WHERE gosta.isbn = livro.isbn);
```