



# UNIVERSIDADE DE ÉVORA

## **1º Trabalho de Bases de Dados**

2023/2024

João Santos, 51966  
Rafaela Abade, 52246  
Catarina Andrade, 55406

Docente Irene Pimenta

## **Exercício 1**

- **membro (Nome, IdMembro, País, Cidade, DataNasc)**

Super chaves:

{IdMemb};  
{IdMemb, Nome};  
{IdMemb, País};  
{IdMemb, Cidade};  
{IdMemb, DataNasc};  
{Nome, IdMemb, País};  
{Nome, IdMemb, Cidade};  
{Nome, IdMemb, DataNasc};  
{IdMemb, País, Cidade};  
{IdMemb, País, DataNasc};  
{IdMemb, Cidade, DataNasc};  
{Nome, IdMemb, País, Cidade};  
{Nome, IdMemb, País, DataNasc};  
{Nome, IdMemb, Cidade, DataNasc};  
{IdMemb, País, Cidade, DataNasc};  
{Nome, IdMemb, País, Cidade, DataNasc}.

Chaves Candidatas:

{IdMemb}

Chave Primária:

{IdMemb}

Chaves Estrangeiras: Não há

- **amigo (IdMemb1, IdMemb2)**

Super chaves:

{IdMemb1, IdMemb2}

Chaves candidatas:

{IdMemb1, IdMemb2}

Chave Primária:

{IdMemb1, IdMemb2}

Chave Estrangeiras:

IdMemb1 da relação membro

IdMemb2 da relação membro

- **gosta (IdMemb, ISBN)**

Super chaves:

{IdMemb, ISBN}

Chaves candidatas:

{IdMemb, ISBN}

Chave Primária:

{IdMemb, ISBN}

Chaves Estrangeiras:

ISBN da relação livro

IdMemb da relação membro

- **livro (ISBN,Título)**

Super chaves:

{ISBN}

{ISBN, Título}

Chaves Candidatas:

{ISBN}

Chave Primária:

{ISBN}

Chaves Estrangeiras: Não há

- **genero (ISBN, Genero)**

Super chaves:

{ISBN, Genero}

Chaves Candidatas:

{ISBN, Genero}

Chave Primária:

{ISBN Genero}

Chaves Estrangeiras:

ISBN da relação livro

- **autoria (ISBN,Coda)**

Super chaves:

{ISBN, Coda}

Chaves Candidatas:

{ISBN, Coda}

Chave Primária:

{ISBN, Coda}

Chaves Estrangeiras:

ISBN da relação livro

Coda da relação autor

- **autor (Coda, Nome, País)**

Super chaves:

{Coda};

{Coda, Nome};

{Coda, País};

{Coda, Nome, País}

Chaves Candidatas:

{Coda}

Chave Primária:

{Coda}

Chaves Estrangeiras: Não há

## **Exercício 2**

```
CREATE TABLE membro(  
  Nome VARCHAR(40),  
  IdMemb VARCHAR(20),  
  Pais VARCHAR(20),  
  Cidade VARCHAR(40),  
  DataNasc DATE,  
  PRIMARY KEY (IdMemb)  
);
```

```
CREATE TABLE amigo(  
  IdMemb1 VARCHAR(20),  
  IdMemb2 VARCHAR(20),  
  PRIMARY Key (IdMemb1, IdMemb2),  
  FOREIGN KEY (IdMemb1) REFERENCES membro(IdMemb),  
  FOREIGN KEY (IdMemb2) REFERENCES membro(IdMemb)  
);
```

```
CREATE TABLE livro(  
  ISBN INT NOT NULL,  
  título VARCHAR(40),  
  PRIMARY KEY (ISBN)  
);
```

```
CREATE Table gosta(  
  IdMemb VARCHAR(20),  
  ISBN INT Not NULL,  
  PRIMARY KEY(IdMemb, ISBN),  
  FOREIGN KEY (ISBN) REFERENCES livro(ISBN),  
  FOREIGN KEY (IdMemb) REFERENCES membro(IdMemb)  
);
```

```
CREATE TABLE genero(  
  ISBN INT NOT NULL,  
  genero VARCHAR(40),  
  PRIMARY KEY (ISBN, genero),  
  FOREIGN KEY (ISBN) REFERENCES livro(ISBN)  
);
```

```
CREATE TABLE autor(  
  Coda INT NOT NULL,  
  Nome VARCHAR(40),  
  Pais VARCHAR(20),  
  PRIMARY KEY(Coda)  
);
```

```
CREATE TABLE autoria(  
  ISBN INT NOT NULL,  
  Coda INT NOT NULL,  
  Primary KEY (ISBN, Coda),  
  FOREIGN KEY (ISBN) REFERENCES livro(ISBN),  
  FOREIGN KEY (Coda) REFERENCES autor(Coda)  
);
```

### **Exercício 3**

a)

```
INSERT INTO membro VALUES('Joao', 'oleitor', 'Portugal', 'Almeirim',
TO_DATE('21-01-2003', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Andre', 'ocorredor', 'Portugal', 'Lisboa',
TO_DATE('01-02-2005', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Diogo', 'otorto', 'Portugal', 'Lisboa', TO_DATE('25-01-2001',
'DD-MM-YYYY'));
INSERT INTO membro VALUES('Ricardo', 'orico', 'Portugal', 'Lisboa',
TO_DATE('12-01-2003', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Miguel', 'opobre', 'Portugal', 'Santarem',
TO_DATE('14-01-1999', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Carlos', 'ointeligente', 'Portugal', 'Beja',
TO_DATE('17-01-1999', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Joana', 'obenfiquista', 'Portugal', 'Beja',
TO_DATE('21-01-1995', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Diana', 'ogato', 'Portugal', 'Portimao',
TO_DATE('18-01-2006', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Andreia', 'oescritor', 'Portugal', 'Monte_Gordo',
TO_DATE('22-01-1993', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Catarina', 'opassaro', 'Portugal', 'Porto',
TO_DATE('26-01-1987', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Bernardo', 'oleao', 'Portugal', 'Porto',
TO_DATE('11-01-1993', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Filipa', 'ocrocodilo', 'Portugal', 'Porto',
TO_DATE('08-01-1994', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Vanessa', 'obucha', 'Portugal', 'Porto',
TO_DATE('09-01-1986', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Joao', 'oestica', 'Portugal', 'Coimbra',
TO_DATE('02-01-1985', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Georgina', 'oslime', 'Portugal', 'Coimbra',
TO_DATE('05-01-1982', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Mariana', 'oreligioso', 'Portugal', 'Evora',
TO_DATE('22-01-2007', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Victor', 'obolt', 'Portugal', 'Evora', TO_DATE('16-01-2008',
'DD-MM-YYYY'));
INSERT INTO membro VALUES('Carlos', 'odetetive', 'Portugal', 'Evora',
TO_DATE('17-01-2002', 'DD-MM-YYYY'));
INSERT INTO membro VALUES('Erica', 'orei', 'Portugal', 'Barreiro', TO_DATE('02-01-2003',
'DD-MM-YYYY'));
INSERT INTO membro VALUES('Antonio', 'oarthur', 'Portugal', 'Alcochete',
TO_DATE('22-01-2001', 'DD-MM-YYYY'));
```



b)

```
INSERT INTO livro VALUES('123', 'Memorial do Convento');
INSERT INTO livro VALUES('124', 'O Principezinho');
INSERT INTO livro VALUES('125', 'O Assassinato no Expresso Oriente');
INSERT INTO livro VALUES('126', 'A luz de Pequim');
INSERT INTO livro VALUES('127', 'Pai Rico, Pai Pobre');
INSERT INTO livro VALUES('128', 'A Caverna');
INSERT INTO livro VALUES('129', 'A noite das bruxas');
INSERT INTO livro VALUES('130', 'Os cinco porquinhos');
INSERT INTO livro VALUES('131', 'O colecionador de erva');
INSERT INTO livro VALUES('132', 'O mar em Casablanca');
```

```
INSERT INTO autor VALUES('1', 'Jose Saramago', 'Portugal');
INSERT INTO autor VALUES('2', 'Antoine de Saint-Exupery', 'França');
INSERT INTO autor VALUES('3', 'Robert Kiyosaki', 'EUA');
INSERT INTO autor VALUES('4', 'Agatha Christie', 'Reino Unido');
INSERT INTO autor VALUES('5', 'Francisco Jose Viegas', 'Portugal');
```

**-> Cada autor deve ter pelo menos 2 livros**

```
INSERT INTO autoria VALUES('123', '1');
INSERT INTO autoria VALUES('124', '2');
INSERT INTO autoria VALUES('125', '3');
INSERT INTO autoria VALUES('126', '4');
INSERT INTO autoria VALUES('127', '5');
```

```
INSERT INTO autoria VALUES('128', '1');
INSERT INTO autoria VALUES('129', '2');
INSERT INTO autoria VALUES('130', '3');
INSERT INTO autoria VALUES('131', '4');
INSERT INTO autoria VALUES('132', '5');
```

**-> Um dos livros deve ter pelo menos 3 autores e outro 2 autores**

```
INSERT INTO autoria VALUES('123', '2');
INSERT INTO autoria VALUES('123', '3');
```

```
INSERT INTO autoria VALUES('124', '4');
```

**-> Todos os livros devem ter um ou mais gêneros (drama, romance, ...).**

```
INSERT INTO genero VALUES ('123', 'Romance') ;
INSERT INTO genero VALUES ('124', 'Romance');
INSERT INTO genero Values ('125', 'Misterio');
INSERT INTO genero VALUES ('126', 'Ficção' );
INSERT INTO genero VALUES ('126', 'Misterio' );
INSERT INTO genero VALUES ('127', 'Finanças Pessoais');
INSERT INTO genero VALUES ('128', 'Romance');
INSERT INTO genero VALUES ('128', 'Ficção');
INSERT INTO genero VALUES ('129', 'Romance');
INSERT INTO genero VALUES ('129', 'Misterio');
INSERT INTO genero VALUES ('130', 'Romance');
INSERT INTO genero VALUES ('130', 'Misterio');
INSERT INTO genero VALUES ('131', 'Romance');
INSERT INTO genero VALUES ('132', 'Drama');
```

c)

**-> 1 membro deve ser amigo de todos os outros membros.**

```
INSERT INTO amigo VALUES ('ogato', 'ocorredor');
INSERT INTO amigo VALUES ('ogato', 'otorto');
INSERT INTO amigo VALUES ('ogato', 'orico');
INSERT INTO amigo VALUES ('ogato', 'opobre');
INSERT INTO amigo VALUES ('ogato', 'ointeligente');
INSERT INTO amigo VALUES ('ogato', 'obenfiquista');
INSERT INTO amigo VALUES ('ogato', 'oleitor');
INSERT INTO amigo VALUES ('ogato', 'oescritor');
INSERT INTO amigo VALUES ('ogato', 'opassaro');
INSERT INTO amigo VALUES ('ogato', 'oleao');
INSERT INTO amigo VALUES ('ogato', 'ocrocodilo');
INSERT INTO amigo VALUES ('ogato', 'obucha');
INSERT INTO amigo VALUES ('ogato', 'oestica');
INSERT INTO amigo VALUES ('ogato', 'oslime');
INSERT INTO amigo VALUES ('ogato', 'oreligioso');
INSERT INTO amigo VALUES ('ogato', 'obolt');
INSERT INTO amigo VALUES ('ogato', 'odetetive');
INSERT INTO amigo VALUES ('ogato', 'orei');
INSERT INTO amigo VALUES ('ogato', 'oarthur');
```

**-> 5 membros devem ter pelo menos 3 amigos.**

```
INSERT INTO amigo VALUES ('ocorredor', 'otorto');  
INSERT INTO amigo VALUES ('ocorredor', 'orico');  
INSERT INTO amigo VALUES ('ocorredor', 'opobre');
```

```
INSERT INTO amigo VALUES ('otorto', 'ointeligente');  
INSERT INTO amigo VALUES ('otorto', 'obenfiquista');  
INSERT INTO amigo VALUES ('otorto', 'oarthur');
```

```
INSERT INTO amigo VALUES ('orico', 'oescritor');  
INSERT INTO amigo VALUES ('orico', 'opassaro');  
INSERT INTO amigo VALUES ('orico', 'oleao');
```

```
INSERT INTO amigo VALUES ('opobre', 'ocrocodilo');  
INSERT INTO amigo VALUES ('opobre', 'obucha');  
INSERT INTO amigo VALUES ('opobre', 'oestica');
```

d)

**-> Todos os membros devem gostar de pelo menos 1 livro.**

```
INSERT INTO gosta VALUES ('oleitor', '123') ;  
INSERT INTO gosta VALUES ('ocorredor', '124') ;  
INSERT INTO gosta VALUES ('otorto', '125') ;  
INSERT INTO gosta VALUES ('orico', '126') ;  
INSERT INTO gosta VALUES ('opobre', '127') ;
```

```
INSERT INTO gosta VALUES ('ointeligente', '128') ;  
INSERT INTO gosta VALUES ('obenfiquista', '129') ;  
INSERT INTO gosta VALUES ('ogato', '130') ;  
INSERT INTO gosta VALUES ('oescritor', '131') ;  
INSERT INTO gosta VALUES ('opassaro', '132') ;
```

```
INSERT INTO gosta VALUES ('oleao', '123') ;  
INSERT INTO gosta VALUES ('ocrocodilo', '124') ;  
INSERT INTO gosta VALUES ('obucha', '125') ;  
INSERT INTO gosta VALUES ('oestica', '126') ;  
INSERT INTO gosta VALUES ('oslime', '127') ;
```

```
INSERT INTO gosta VALUES ('oreligioso', '128') ;  
INSERT INTO gosta VALUES ('obolt', '129') ;  
INSERT INTO gosta VALUES ('odetative', '130') ;  
INSERT INTO gosta VALUES ('orei', '131') ;  
INSERT INTO gosta VALUES ('oarthur', '132') ;
```

**-> Pelo menos 1 membro deve gostar de todos os livros de um autor.**

```
INSERT INTO gosta VALUES ('ocorredor', '127') ;  
INSERT INTO gosta VALUES ('ocorredor', '132') ;
```

## **Exercício 4**

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

Expressão em SQL:

```
SELECT nome FROM autor, autoria, genero
WHERE autor.coda = autoria.coda
AND autoria.isbn = genero.isbn
AND genero = 'Drama' ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{nome}}(\sigma_{(\text{autor.coda} = \text{autoria.coda} \wedge \text{autoria.isbn} = \text{genero.isbn} \wedge \text{genero} = \text{'Drama'})}(\text{autor} \times \text{autoria} \times \text{genero}))$$

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

Expressão em SQL:

```
SELECT DISTINCT membro.nome FROM membro, gosta, autoria, autor
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = autoria.isbn
AND autoria.coda = autor.coda
AND autor.nome = 'Agatha Christie' ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{membro.nome}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{autoria.isbn} \wedge \text{autoria.coda} = \text{autor.coda} \wedge \text{autor.nome} = \text{'Agatha Christie'})}(\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor}))$$

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

Expressão em SQL:

```
SELECT DISTINCT membro.nome FROM membro, gosta, autoria, autor
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = autoria.isbn
AND autoria.coda = autor.coda
AND autor.pais = membro.pais ;
```

Expressão em Álgebra Relacional:

$\pi_{\text{membro.nome}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{autoria.isbn} \wedge \text{autoria.coda} = \text{autor.coda} \wedge \text{autor.pais} = \text{membro.pais})}(\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor}))$

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

Expressão em SQL:

```
SELECT membro.idmemb FROM membro, gosta, autoria, autor
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = autoria.isbn
AND autoria.coda = autor.coda
```

EXCEPT

```
SELECT membro.idmemb FROM membro, gosta, autoria, autor
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = autoria.isbn
AND autoria.coda = autor.coda
AND autor.nome = 'Agatha Christie' ;
```

Expressão em Álgebra Relacional:

$\pi_{\text{membro.idmemb}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{autoria.isbn} \wedge \text{autoria.coda} = \text{autor.coda})}(\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor})) - \pi_{\text{membro.idmemb}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{autoria.isbn} \wedge \text{autoria.coda} = \text{autor.coda} \wedge \text{autor.nome} = \text{'Agatha Christie'})}(\text{membro} \times \text{gosta} \times \text{autoria} \times \text{autor}))$

**e) Quais os membros que não são amigos do membro com o idMemb oleitor?**

Expressão em SQL:

```
SELECT idmemb FROM membro
```

EXCEPT

```
(SELECT idmemb FROM membro, amigo
WHERE amigo.idmemb1 = 'oleitor' AND amigo.idmemb2 = idmemb
```

UNION

```
SELECT idmemb FROM membro, amigo
WHERE amigo.idmemb2 = 'oleitor' AND amigo.idmemb1 = idmemb) ;
```

Expressão em Álgebra Relacional:

$\pi_{\text{idmemb}}(\text{membro}) - (\pi_{\text{idmemb}}(\sigma_{(\text{amigo.idmemb1} = \text{'oleitor'} \wedge \text{amigo.idmemb2} = \text{idmemb})}(\text{membro} \times \text{amigo}) \cup (\pi_{\text{idmemb}}(\sigma_{(\text{amigo.idmemb2} = \text{'oleitor'} \wedge \text{amigo.idmemb1} = \text{idmemb})}(\text{membro} \times \text{amigo})))$

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

Expressão em SQL:

```
WITH amigos AS (  
SELECT DISTINCT membro.idmemb, membro.datanasc  
FROM membro INNER JOIN amigo ON idmemb = idmemb2  
WHERE idmemb1 = 'oleitor'
```

UNION

```
SELECT DISTINCT membro.idmemb, membro.datanasc  
FROM membro INNER JOIN amigo ON idmemb = idmemb1  
WHERE idmemb2 = 'oleitor')  
SELECT membro.nome  
FROM (membro NATURAL INNER JOIN amigos), (SELECT idmemb, datanasc  
FROM membro  
WHERE IdMemb = 'oleitor') AS leitor  
WHERE amigos.datanasc > leitor.datanasc ;
```

Expressão em Álgebra Relacional:

$\text{amigos} \leftarrow (\pi_{\text{membro.idmemb, membro.datanasc}}(\sigma_{(\text{idmemb} = \text{idmemb2} \wedge \text{idmemb1} = \text{'oleitor'})}(\text{membro} \bowtie \text{amigo}) \cup \pi_{\text{membro.idmemb, membro.datanasc}}(\sigma_{(\text{idmemb} = \text{idmemb1} \wedge \text{idmemb2} = \text{'oleitor'})}(\text{membro} \bowtie \text{amigo})))$   
 $\text{leitor} \leftarrow \pi_{\text{idmemb, datanasc}}(\sigma_{(\text{idmemb} = \text{'oleitor'})}(\text{membro}))$   
 $\pi_{\text{membro.nome}}(\sigma_{(\text{amigo.datanasc} > \text{leitor.datanasc})}((\text{membro} \bowtie \text{amigo}) \times \text{leitor}))$

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

Expressão em SQL:

```
SELECT DISTINCT membro.nome FROM membro
INNER JOIN gosta ON gosta.idmemb = membro.idmemb
INNER JOIN autoria ON autoria.isbn = gosta.isbn
INNER JOIN autor ON autor.coda = autoria.coda
WHERE autor.nome = 'Agatha Christie'
```

INTERSECT

```
SELECT DISTINCT membro.nome FROM membro
INNER JOIN gosta ON membro.idmemb = gosta.idmemb
INNER JOIN autoria ON autoria.isbn = gosta.isbn
INNER JOIN autor ON autor.coda = autoria.coda
WHERE autor.nome = 'Francisco Jose Viegas' ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{membro.nome}}(\sigma_{(\text{gosta.idmemb} = \text{membro.idmemb} \wedge \text{autoria.isbn} = \text{gosta.isbn} \wedge \text{autor.coda} = \text{autoria.coda} \wedge \text{autor.nome} = \text{'Agatha Christie'})})(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor}) - \pi_{\text{membro.nome}}(\sigma_{(\text{gosta.idmemb} = \text{membro.idmemb} \wedge \text{autoria.isbn} = \text{gosta.isbn} \wedge \text{autor.coda} = \text{autoria.coda} \wedge \text{autor.nome} = \text{'Francisco Jose Viegas'})})(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor})$$

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

Expressão em SQL:

```
SELECT DISTINCT membro.nome FROM membro
INNER JOIN gosta ON membro.idmemb = gosta.idmemb
INNER JOIN autoria ON autoria.isbn = gosta.isbn
INNER JOIN autor ON autor.coda = autoria.coda
WHERE autor.nome = 'Agatha Christie' OR autor.nome = 'Francisco Jose Viegas' ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{membro.nome}}(\sigma_{(\text{membro.idmemb} = \text{gosta.idmemb} \wedge \text{autoria.isbn} = \text{gosta.isbn} \wedge \text{autor.coda} = \text{autoria.coda} \wedge \text{autor.nome} = \text{'Agatha Christie'} \vee \text{autor.nome} = \text{'Francisco Jose Viegas'})})(\text{membro} \bowtie \text{gosta} \bowtie \text{autoria} \bowtie \text{autor})$$

**i) Quantos amigos tem o membro oleitor?**

Expressão em SQL:

```
SELECT COUNT(*) FROM(
(SELECT DISTINCT membro.nome
FROM membro INNER JOIN amigo ON idmemb = idmemb2
WHERE idmemb1 = 'oleitor'
```

UNION

```
SELECT DISTINCT membro.nome
FROM membro INNER JOIN amigo ON idmemb = idmemb1
WHERE idmemb2 = 'oleitor'))
AS amigos ;
```

Expressão em Álgebra Relacional:

$$\text{amigos} \leftarrow (\pi_{\text{membro.nome}}(\sigma_{(\text{idmemb} = \text{idmemb2} \wedge \text{idmemb1} = \text{'oleitor'})}(\text{membro} \bowtie \text{amigo})) \cup \\ (\pi_{\text{membro.nome}}(\sigma_{(\text{idmemb} = \text{idmemb1} \wedge \text{idmemb2} = \text{'oleitor'})}(\text{membro} \bowtie \text{amigo}))) \\ \mathcal{G}_{\text{count}(*), \text{as } n}(\text{amigos})$$

**j) Qual é o membro que tem mais amigos?**

Expressão em SQL:

```
SELECT membro.idmemb, COUNT (amigo.idmemb1)
FROM membro
INNER JOIN amigo ON (amigo.idmemb1 = membro.idmemb OR amigo.idmemb2 =
membro.idmemb)
GROUP by membro.idmemb
ORDER BY COUNT (amigo.idmemb1) DESC
LIMIT 1 ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{membro.idmemb}} \mathcal{G}_{\text{count}(\text{amigo.idmemb1})}(\sigma_{(\text{amigo.idmemb1} = \text{membro.idmemb} \vee \text{amigo.idmemb2} = \text{membro.idmemb})}(\text{membro} \bowtie \text{amigo}))$$



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

Expressão em SQL:

```
WITH leitor AS (  
WITH expr AS (  
SELECT DISTINCT membro.idmemb, COUNT(isbn) AS Livros  
FROM gosta NATURAL INNER JOIN membro  
GROUP BY membro.idmemb)
```

```
SELECT DISTINCT expr.idmemb  
FROM expr NATURAL INNER JOIN (  
SELECT MAX(expr.Livros) AS fim  
FROM expr)  
AS tabela  
WHERE tabela.fim = expr.Livros)
```

```
SELECT membro.nome  
FROM (  
SELECT DISTINCT amigos.Id  
FROM (  
SELECT DISTINCT idmemb1 AS Id  
FROM amigo, leitor  
WHERE idmemb2 = leitor.idmemb
```

UNION

```
SELECT DISTINCT idmemb2 AS Id  
FROM amigo, leitor  
WHERE idmemb1 = leitor.idmemb) AS amigos)  
AS amigos INNER JOIN membro ON amigos.Id = membro.idmemb ;
```

Expressão em Álgebra Relacional:

```
expr  $\leftarrow$  ( $\pi_{\text{membro.idmemb}}$   $\mathcal{G}_{\text{count (isbn) as n}}$ (Livros)(gosta  $\bowtie$  membro)  
tabela  $\leftarrow$   $\mathcal{G}_{\text{max (expr.Livros) as n}}$ (fim)(expr)  
leitor  $\leftarrow$  ( $\pi_{\text{expr.idmemb}}$  ( $\sigma_{\text{(tabela.fim = expr.Livros)}}$ )(expr  $\bowtie$  tabela)  
amigos  $\leftarrow$  ( $\pi_{\text{idmemb1 as Id}}$  ( $\sigma_{\text{(idmemb2 = leitor.idmemb)}}$ )(amigo x leitor)  $\cup$  ( $\pi_{\text{idmemb2 as Id}}$   
( $\sigma_{\text{(idmemb1 = leitor.idmemb)}}$ )(amigo x leitor))  
 $\pi_{\text{membro.nome}}$  ( $\sigma_{\text{(amigos.Id = membro.Id)}}$ )(amigos  $\bowtie$  membro)
```

**l) Para cada livro indique o número de géneros.**

Expressão em SQL:

```
SELECT livro.titulo, COUNT (genero)
FROM livro
INNER JOIN genero ON livro.isbn = genero.isbn
GROUP BY livro.titulo ;
```

Expressão em Álgebra Relacional:

$\text{livro.titulo } \mathcal{G}_{\text{count (genero) as n}}(\text{livro} \bowtie \text{genero})$

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

Expressão em SQL:

```
SELECT gostos.titulo, gostos.ngostos, generos.ngeneros
from (
(SELECT DISTINCT titulo, isbn, COUNT(idmemb) AS ngostos
FROM livro NATURAL INNER JOIN gosta
GROUP BY isbn) AS gostos
INNER JOIN
(SELECT isbn, COUNT(genero) AS ngeneros
FROM livro NATURAL INNER JOIN genero
GROUP BY isbn) AS generos USING(isbn));
```

Expressão em Álgebra Relacional:

$\text{gostos} \leftarrow \pi_{\text{titulo, isbn}} \mathcal{G}_{\text{count (idmemb) as n}}(\text{ngostos})(\text{livro} \bowtie \text{gosta})$   
 $\text{gneros} \leftarrow \pi_{\text{isbn}} \mathcal{G}_{\text{count (genero) as n}}(\text{ngeneros})(\sigma_{(\text{livro.isbn} = \text{genero.isbn})}(\text{livro} \bowtie \text{genero}))$   
 $\pi_{\text{gostos.titulo, gostos.ngostos, generos.ngeneros}}(\text{gostos} \bowtie \text{generos})$

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

Expressão em SQL:

```
SELECT livros.nome, livros.nlivros, generos.ngeneros, gostos.ngostos
FROM (
  (SELECT autor.nome, COUNT(isbn) AS nlivros
   FROM autor NATURAL INNER JOIN autoria
   GROUP BY autor.nome) AS livros
 NATURAL INNER JOIN
  (SELECT autor.nome, COUNT(genero) AS ngeneros
   FROM autor NATURAL INNER JOIN autoria
   NATURAL INNER JOIN genero
   GROUP BY autor.nome) AS generos
 NATURAL INNER JOIN
  (SELECT autor.nome, COUNT(idmemb) AS ngostos
   FROM autor NATURAL INNER JOIN autoria
   NATURAL INNER JOIN gosta
   GROUP BY autor.nome) AS gostos);
```

Expressão em Álgebra Relacional:

```
livros ←  $\pi_{\text{autor.nome}} \mathcal{G}_{\text{count (isbn) as n}}(\text{nlivros})(\text{autor} \bowtie \text{autoria})$ 
generos ←  $\pi_{\text{autor.nome}} \mathcal{G}_{\text{count (genero) as n}}(\text{ngeneros})(\text{autor} \bowtie \text{autoria} \bowtie \text{genero})$ 
gostos ←  $\pi_{\text{autor.nome}} \mathcal{G}_{\text{count (dmemb) as n}}(\text{ngostos})(\text{autor} \bowtie \text{autoria} \bowtie \text{gosta})$ 
 $\pi_{\text{livros.nome, livros.nlivros, generos.nlivros, gostos.ngostos}}(\text{livros} \bowtie \text{generos} \bowtie \text{gostos})$ 
```

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

Expressão em SQL:

```
SELECT amigos.idmemb, livros.nome, livros.cont AS nlivros, amigos.cont AS namigos
FROM (
  (SELECT membro.idmemb, membro.nome, COUNT(gosta.isbn) AS cont
   FROM membro NATURAL INNER JOIN gosta
   GROUP BY membro.idmemb) AS livros
 INNER JOIN
  (SELECT membro.idmemb, membro.nome, COUNT(idmemb1) AS cont
   FROM membro INNER JOIN amigo ON idmemb2 = idmemb OR idmemb1 = idmemb
   GROUP BY membro.idmemb) AS amigos
 USING(idmemb));
```

Expressão em Álgebra Relacional:

$livros \leftarrow \pi_{membro.idmemb, membro.nome} \mathcal{G}_{count} (gosta.isbn) \text{ as } n(cont)(membro \bowtie gosta)$   
 $amigos \leftarrow \pi_{membro.idmemb, membro.nome} \mathcal{G}_{count} (idmemb1) \text{ as } n(cont)(\sigma_{(idmemb2 = idmemb \vee idmemb1 = idmemb)})(membro \bowtie amigo)$   
 $\pi_{amigos.idmemb, livros.nome, livros.cont \text{ as } n(livros, amigos.cont \text{ as } n(amigos) (livros \bowtie amigos)$

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

Expressão em SQL:

```
WITH dados AS (  
  SELECT DISTINCT amigos.nome, SUM(namigos)  
  FROM (  
    SELECT DISTINCT membro.nome, COUNT(idmemb) AS amigos  
    FROM membro INNER JOIN amigo ON idmemb = idmemb1  
    GROUP BY membro.nome  
  
    UNION  
  
    SELECT DISTINCT membro.nome, COUNT (idmemb) AS amigos  
    FROM membro INNER JOIN amigo ON idmemb = idmemb2  
    GROUP BY membro.nome) AS amigos  
  GROUP BY amigos.nome)  
  
SELECT dados.nome  
FROM (  
  SELECT COUNT (idmemb) - 1 AS cont  
  FROM membro) AS tabela  
NATURAL INNER JOIN dados  
WHERE dados.sum = tabela.cont ;
```

Expressão em Álgebra Relacional:

$amigos \leftarrow (\pi_{membro.nome} \mathcal{G}_{count} (idmemb) \text{ as } n(namigos)(\sigma_{(idmemb = idmemb1)})(membro \bowtie amigo) \cup (\pi_{membro.nome} \mathcal{G}_{count} (idmemb) \text{ as } n(namigos)(\sigma_{(idmemb = idmemb2)})(membro \bowtie amigo))$   
 $dados \leftarrow (\pi_{amigos.nome} \mathcal{G}_{sum} (namigos) \text{ as } n(amigos)$   
 $tabela \leftarrow \pi_{\mathcal{G}_{count} (idmemb) - 1 \text{ as } cont}(membro)$   
 $\pi_{dados.nome}(\sigma_{dados.sum = tabela.cont})(tabela \bowtie dados)$

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

Expressão em SQL:

```
SELECT DISTINCT livro.titulo
FROM membro, gosta, livro, amigo
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = livro.isbn
AND membro.idmemb = amigo.idmemb1
AND amigo.idmemb2 = 'leitor'
```

UNION

```
SELECT DISTINCT livro.titulo
FROM membro, gosta, livro, amigo
WHERE membro.idmemb = gosta.idmemb
AND gosta.isbn = livro.isbn
AND membro.idmemb = amigo.idmemb2
AND amigo.idmemb1 = 'leitor' ;
```

Expressão em Álgebra Relacional:

$$\pi_{\text{livro.titulo}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{livro.isbn} \wedge \text{membro.idmemb} = \text{amigo.idmemb1} \wedge \text{amigo.idmemb2} = \text{'leitor'})})(\text{membro} \times \text{gosta} \times \text{livro} \times \text{amigo}) \cup \pi_{\text{membro.idmemb}}(\sigma_{(\text{membro.idmembro} = \text{gosta.idmembro} \wedge \text{gosta.isbn} = \text{livro.isbn} \wedge \text{membro.idmemb} = \text{amigo.idmemb2} \wedge \text{amigo.idmemb1} = \text{'leitor'})})(\text{membro} \times \text{gosta} \times \text{livro} \times \text{amigo})$$