

# 2º Trabalho de Base de Dados

## Relatório

### Licenciatura em Engenharia Informática

Ano Letivo 2024/2025

Trabalho realizado por:  
André Gonçalves – 58392  
André Zhan - 58762

## ÍNDICE

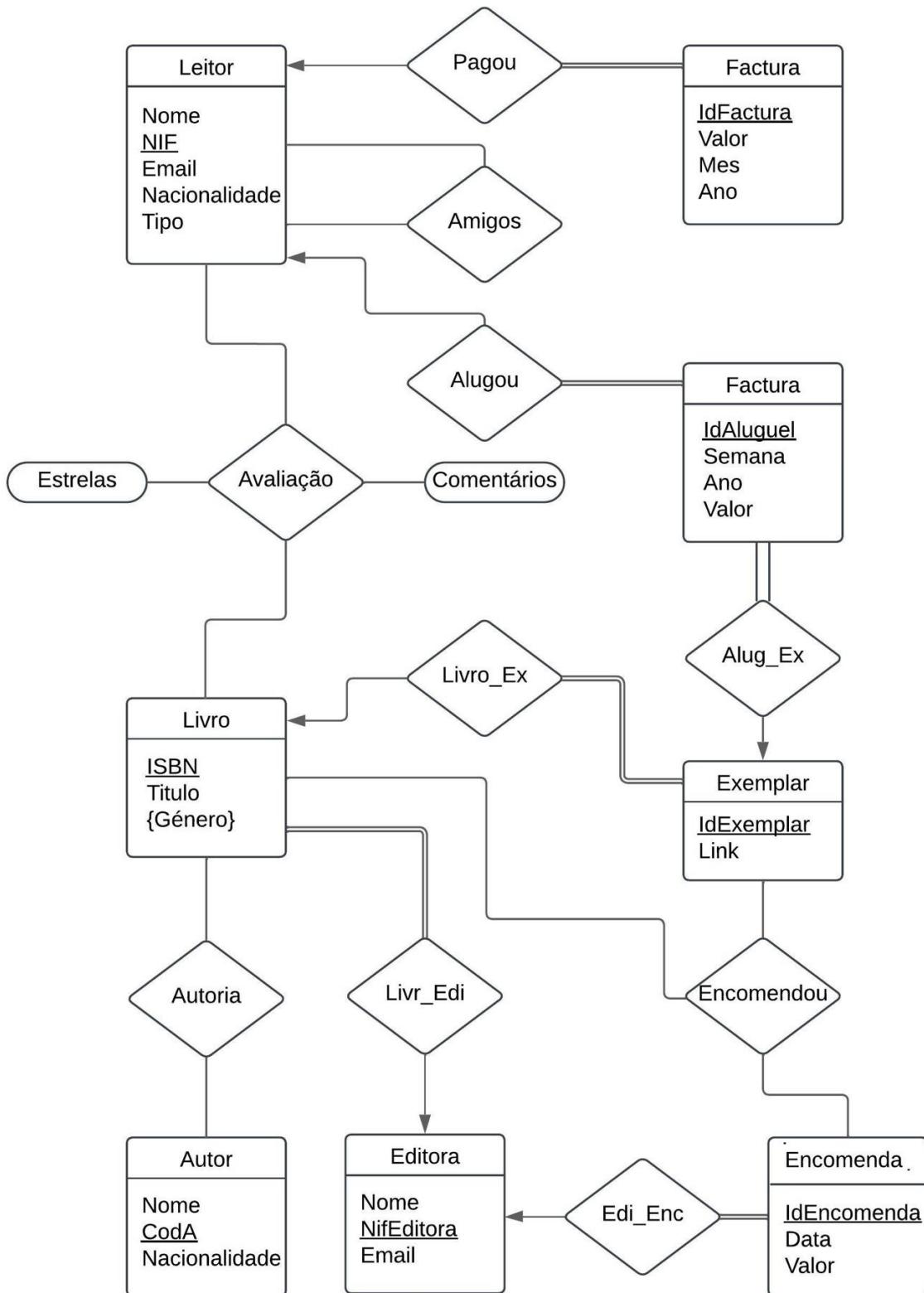
<i>Objetivos e Resolução .....</i>	3
<i>1 – Construção do diagrama Entidade-Relação .....</i>	4
<i>2 – Relações da Base de Dados .....</i>	5
<i>3 – Conjunto de Dependências Funcionais .....</i>	6
<i>Dependências funcionais .....</i>	7
<i>4 – Cobertura Canónica .....</i>	8
<i>1 – Juntar todos os atributos á direita que tenham a esquerda igual .....</i>	8
<i>2 – Verificar atributos extra á Esquerda e á Direita: .....</i>	9
<i>5 – Base de Dados na forma normal de Boyce Codd .....</i>	10
<i>Base de dados em BCNF .....</i>	13
<i>7 – Chaves Candidatas, Primárias e Estrangeiras .....</i>	15
<i>8 – Comandos SQL Das tabelas da criação da base de dados .....</i>	15
<i>9 – Expressões SQL Para Inserção de Dados .....</i>	20
<i>10 – Expressões em Álgebra Relacional e SQL .....</i>	36

## OBJETIVOS E RESOLUÇÃO

Pretende-se desenvolver uma base de dados para gerir os alugueres e as encomendas de exemplares de livros de uma empresa.

Quanto à nossa resolução, seguimos a ordem dos exercícios propostos e, no exercício 10., começamos por escrever as expressões de cada alínea em Álgebra Relacional e, só depois, traduzimos para SQL.

# 1 – CONSTRUÇÃO DO DIAGRAMA ENTIDADE-RELAÇÃO



## 2 – RELAÇÕES DA BASE DE DADOS

Leitor(Nome , NIF, Email, Nacionalidade, Tipo)

Exemplar(IdExemplar, ISBN, Link)

Livro(Titulo, ISBN, NifEditora)

Livro\_Gen( ISBN, Género)

Autor( Nome, CodA, Nacionalidade)

Autoria( ISBN, CodA)

Editora( nome, NifEditora, Email)

Amigos( NIF1, NIF2)

Avaliação( NIF, ISBN, Estrelas, Comentários)

Encomenda(IdEncomenda, Data, Valor, NifEditora)

Encomendou(IdExemplar, ISBN, IdEncomenda)

Factura(IdFatura, Nif, Valor, mês, Ano)

Aluguer(IdAluguel, Semana, Ano, Valor, Nif, IdExemplar)

### 3 – CONJUNTO DE DEPENDÊNCIAS FUNCIONAIS

1 - Cada NIF de leitor identifica um único leitor, incluindo o seu nome, email, data de nacionalidade e tipo de leitor (regular ou frequente).

**NIF → Nome, Email, Nacionalidade, Tipo**

2 - Cada ID de exemplar identifica um único exemplar de livro, indicando o ISBN do livro a que pertence e o link para download (se aplicável).

**IdExemplar → ISBN, Link**

3 - Cada ISBN identifica um único livro, indicando o seu título e o NIF da editora que o publicou.

**ISBN → Título, NifEditora**

4 - Cada CodA (código de autor) identifica um único autor, indicando o seu nome e data de nacionalidade.

**CodA → Nome, Nacionalidade**

5 - Cada NifEditora de editora identifica uma única editora, indicando o seu nome e email."

**NifEditora → Nome, Email**

6 - Cada combinação de NIF de leitor e ISBN de livro identifica uma única avaliação, incluindo o número de estrelas e o comentário dado pelo leitor sobre o livro.

**NIF, ISBN → Estrelas, Comentários**

7 – Cada IdAluguel identifica uma semana, ano e valor de um aluguer

**IdAluguel → Semana, Ano, Valor, NifIdExemplar**

8 - "Cada ID de fatura identifica uma única fatura, indicando o NIF do leitor, o valor total, o mês e o ano da fatura."

**IdFatura → Nif, Valor, mês, Ano**

9 - Amizades são simétricas

Se um leitor é amigo de outro, a relação deve ser recíproca.

**NIF1, NIF2 → (NIF2, NIF1)**

10 – Cada IdEncomenda identifica a data, valor e o Nif da editora de uma encomenda.

**IdEncomenda, → Data, Valor,      NifEditora**

11 – Cada IdExemplar tem um só ISBN

**IdExemplar → ISBN**

12 – Cada Encomenda é feita a uma só editora

**IdEncomenda → NifEditora**

## DEPENDÊNCIAS FUNCIONAIS

NIF → Nome, Email, Nacionalidade, Tipo

IdExemplar → ISBN, Link

ISBN → Título, NifEditora

CodA → Nome, Nacionalidade

NifEditora → Nome, Email

NIF, ISBN → Estrelas, Comentários

IdAluguel → Semana, Ano, Valor, NifIdExemplar

IdFatura → Nif, Valor, mês, Ano

NIF1, NIF2 → (NIF2, NIF1)

IdEncomenda, → Data, Valor,      NifEditora

IdExemplar → ISBN

IdEncomenda → NifEditora

## 4 – COBERTURA CANÓNICA

### 1 – JUNTAR TODOS OS ATRIBUTOS Á DIREITA QUE TENHAM A ESQUERDA IGUAL

IdEncomenda, → Data, Valor, NifEditora

IdEncomenda → NifEditora

### União dos dois fica:

IdEncomenda, → Data, Valor, NifEditora

IdExemplar → ISBN, Link

IdExemplar → ISBN

### União dos dois fica:

IdExemplar → ISBN, Link

### Depois do passo 1 o conjunto de relações fica:

NIF → Nome, Email, Nacionalidade, Tipo

IdExemplar → ISBN, Link

ISBN → Título, NifEditora

CodA → Nome, Nacionalidade

NifEditora → Nome, Email

NIF, ISBN → Estrelas, Comentários

IdAluguel → Semana, Ano, Valor, NifIdExemplar

IdFatura → Nif, Valor, mês, Ano

NIF1, NIF2 → (NIF2, NIF1)

IdEncomenda, → Data, Valor, NifEditora

## 2 – VERIFICAR ATRIBUTOS EXTRA Á ESQUERDA E Á DIREITA:

Uma vez que não existe atributos extra á esquerda nem á direita a cobertura canónica fica:

NIF → Nome, Email, Nacionalidade, Tipo

IdExemplar → ISBN, Link

ISBN → Título, NifEditora

CodA → Nome, Nacionalidade

NifEditora → Nome, Email

NIF, ISBN → Estrelas, Comentários

IdAluguel → Semana, Ano, Valor, NifIdExemplar

IdFatura → Nif, Valor, mês, Ano

NIF1, NIF2 → (NIF2, NIF1)

IdEncomenda, → Data, Valor, NifEditora

## 5 – BASE DE DADOS NA FORMA NORMAL DE BOYCE CODD

**Tendo em conta as seguintes Relações da Base de Dados**

Leitor(Nome , NIF, Email, Nacionalidade, Tipo)

Exemplar(IdExemplar, ISBN, Link)

Livro(Título, ISBN, NifEditora)

Livro\_Gen( ISBN, Género)

Autor( Nome, CodA, Nacionalidade)

Autoria( ISBN, CodA)

Editora( nome, NifEditora, Email)

Amigos( NIF1, NIF2)

Avaliação( NIF, ISBN, Estrelas, Comentários)

Encomenda(IdEncomenda, Data, Valor, NifEditora)

Encomendou(IdExemplar, ISBN, IdEncomenda)

Factura(IdFatura, Nif, Valor, mês, Ano)

Aluguer(IdAluguel, Semana, Ano, Valor, Nif, IdExemplar)

**E as seguintes dependências funcionais (Cobertura Canónica)**

NIF → Nome, Email, Nacionalidade, Tipo

IdExemplar → ISBN, Link

ISBN → Título, NifEditora

CodA → Nome, Nacionalidade

NifEditora → Nome, Email

NIF, ISBN → Estrelas, Comentários

IdAluguel → Semana, Ano, Valor, Nif, IdExemplar

IdFatura → Nif, Valor, mês, Ano

NIF1, NIF2 → (NIF2, NIF1)

IdEncomenda, → Data, Valor, NifEditora

**Vamos verificar as dependências funcionais válidas para cada uma das relações e verificar se estão na BCNF**

#### **1 - Leitor(NIF, Nome , Email, Nacionalidade, Tipo)**

NIF → Nome, Email, Nacionalidade, Tipo

NIF é Superchave da relação Leitor, logo está em BCNF

#### **2 - Exemplar(IdExemplar, ISBN, Link)**

IdExemplar → ISBN, Link

IdExemplar é Superchave da relação Exemplar, logo está em BCNF

#### **3 - Livro(Titulo, ISBN, NifEditora)**

ISBN → Título, NifEditora

ISBN é Superchave da relação Livro, logo está em BCNF

#### **4 - Livro\_Gen(ISBN, Género)**

Não se verifica nenhuma dependência funcional logo já está em BCNF

#### **5 - Autor( Nome, CodA, Nacionalidade)**

CodA → Nome, Nacionalidade

CodA é Superchave da relação Autor, logo está em BCNF

#### **6 - Autoria( ISBN, CodA)**

Não se verifica nenhuma dependência funcional logo já está em BCNF

## 7 - Editora( nome, NifEditora, Email)

NifEditora → Nome, Email

NifEditora é Superchave da relação Editora, logo está em BCNF

## 8 - Amigos( NIF1, NIF2)

NIF1, NIF2 → (NIF2, NIF1)

NIF1, NIF2 é uma dependência funcional Trivial, logo está em BCNF

## 9 - Avaliaçao( NIF, ISBN, Estrelas, Comentários)

NIF, ISBN → Estrelas, Comentários

NIF, ISBN é Superchave da relação Avaliaçao, logo está em BCNF

## 10 - Encomenda(IdEncomenda, Data, Valor, NifEditora)

IdEncomenda, → Data, Valor, NifEditora

IdEncomenda é Superchave da relação Encomenda, logo está em BCNF

## 11 - Encomendou(IdExemplar, ISBN, IdEncomenda)

Não se verifica nenhuma dependência funcional logo já está em BCNF

## 12 - Factura(IdFatura, Nif, Valor, mês, Ano)

IdFatura → Nif, Valor, mês, Ano

IdFatura é Superchave da relação Factura, logo está em BCNF

## 13 - Aluguer(IdAluguel, Semana, Ano, Valor, Nif, IdExemplar)

IdAluguel → Semana, Ano, Valor, IdExemplar

IdAluguel é Superchave da relação Aluguer, logo está em BCNF

**Logo A base de dados em BCNF fica igual uma vez que todas as relações já estavam em BCNF**

## BASE DE DADOS EM BCNF

Leitor(Nome , NIF, Email, Nacionalidade, Tipo)

Exemplar(IdExemplar, ISBN, Link)

Livro(Titulo, ISBN, NifEditora)

Livro\_Gen( ISBN, Género)

Autor( Nome, CodA, Nacionalidade)

Autoria( ISBN, CodA)

Editora( nome, NifEditora, Email)

Amigos( NIF1, NIF2)

Avaliação( NIF, ISBN, Estrelas, Comentários)

Encomenda(IdEncomenda, Data, Valor, NifEditora)

Encomendou(IdExemplar, ISBN, IdEncomenda)

Factura(IdFatura, Nif, Valor, mês, Ano)

Aluguer(IdAluguel, Semana, Ano, Valor, Nif, IdExemplar)

## 6 – Preserva Dependências Funcionais / 3<sup>a</sup> Forma Normal

Vamos verificar para cada dependência relacional se é preservada na base de dados

**NIF → Nome, Email, Nacionalidade, Tipo**

Sim, na relação Leitor.

**IdExemplar → ISBN, Link**

Sim, na relação Exemplar.

**ISBN → Título, NifEditora**

Sim, na relação Livro.

**CodA → Nome, Nacionalidade**

Sim, na relação autor.

**NifEditora → Nome, Email**

Sim, na relação Editora.

**NIF, ISBN → Estrelas, Comentários**

Sim, na relação Avaliação.

**IdAluguel → Semana, Ano, Valor, Nif, IdExemplar**

Sim, na relação Aluguer.

**IdFatura → Nif, Valor, mês, Ano**

Sim, na relação Factura.

**NIF1, NIF2 → (NIF2, NIF1)**

Sim, na relação Amigos.

**IdEncomenda → Data, Valor, NifEditora**

Sim, na relação Encomenda

Logo podemos concluir que a base de dados preserva todas as dependências funcionais e não é necessário colocar na 3<sup>a</sup> Forma Normal.

## 7 – CHAVES CANDIDATAS, PRIMÁRIAS E ESTRANGEIRAS

<b>Relações/Chaves</b>	<b>Candidatas</b>	<b>Primária</b>	<b>Estrangeiras</b>
<b>Leitor</b>	NIF, Email	NIF	-
<b>Exemplar</b>	IdExemplar	IdExemplar	ISBN
<b>Livro</b>	ISBN	ISBN	NifEditora
<b>Livro_Gen</b>	(ISBN, Género)	(ISBN, Género)	ISBN
<b>Autor</b>	CodA	CodA	-
<b>Autoria</b>	(ISBN, CodA)	(ISBN, CodA)	ISBN, CodA
<b>Editora</b>	NifEditora, Email	NifEditora	-
<b>Amigos</b>	(NIF1, NIF2)	(NIF1, NIF2)	NIF
<b>Avaliação</b>	(NIF, ISBN)	(NIF, ISBN)	NIF, ISBN
<b>Encomenda</b>	IdEncomenda	IdEncomenda	NifEditora
<b>Encomendou</b>	(IdExemplar, ISBN, IdEncomenda)	(IdExemplar, ISBN, IdEncomenda)	IdExemplar, ISBN, IdEncomenda
<b>Factura</b>	IdFactura	IdFactura	Nif
<b>Aluguer</b>	IdAluguer	IdAluguer	Nif, IdExemplar

## 8 – COMANDOS SQL DAS TABELAS DA CRIAÇÃO DA BASE DE DADOS

### -- Tabela leitor

```
CREATE TABLE leitor (
  Nome VARCHAR(50) NOT NULL,
  Nif INT primary key,
  Email VARCHAR(50) NOT NULL,
  Nacionalidade VARCHAR(50) NOT NULL,
```

```
Tipo VARCHAR(20) NOT NULL  
);
```

### -- Tabela autor

```
CREATE TABLE autor (  
    Nome VARCHAR(50) NOT NULL,  
    CodA INT PRIMARY KEY,  
    Nacionalidade VARCHAR(50) NOT NULL  
);
```

### -- Tabela editora

```
CREATE TABLE editora (  
    Nome VARCHAR(50) NOT NULL,  
    NifEditora INT PRIMARY KEY,  
    Email VARCHAR(100) NOT NULL  
);
```

### -- Tabela livro

```
CREATE TABLE livro (  
    Titulo VARCHAR(50) NOT NULL,  
    ISBN INT PRIMARY KEY,  
    NifEditora INT NOT NULL,  
    FOREIGN KEY (NifEditora) REFERENCES editora(NifEditora)  
);
```

### -- Tabela exemplar

```
CREATE TABLE exemplar (
```

```
IdExemplar INT PRIMARY KEY,  
ISBN INT NOT NULL,  
LinkDownload VARCHAR(100) NOT NULL,  
FOREIGN KEY (ISBN) REFERENCES livro(ISBN)  
);
```

### -- Tabela livro\_gen

```
CREATE TABLE livro_gen (  
ISBN INT NOT NULL,  
Genero VARCHAR(50) NOT NULL,  
PRIMARY KEY (ISBN, Genero),  
FOREIGN KEY (ISBN) REFERENCES livro(ISBN)  
);
```

### -- Tabela autoria

```
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)  
);
```

### -- Tabela amigos

```
CREATE TABLE amigos (  
Nif1 INT NOT NULL,  
Nif2 INT NOT NULL,
```

```
PRIMARY key (Nif1, Nif2),  
FOREIGN KEY (Nif1) REFERENCES leitor(Nif),  
FOREIGN KEY (Nif2) REFERENCES leitor(Nif),  
CHECK (Nif1 <> Nif2)  
);
```

### -- Tabela avaliacao

```
CREATE TABLE avaliacao (  
    Nif INT NOT NULL,  
    ISBN INT NOT NULL,  
    Estrelas INT CHECK (Estrelas BETWEEN 0 AND 5) NOT NULL,  
    Comentario VARCHAR(100) NOT NULL,  
    PRIMARY KEY (Nif, ISBN),  
    FOREIGN key (Nif) REFERENCES leitor(Nif),  
    FOREIGN key (ISBN) REFERENCES livro(ISBN)  
);
```

### -- Tabela encomenda

```
CREATE TABLE encomenda (  
    IdEncomenda INT primary key,  
    DataEnc DATE NOT NULL,  
    Valor INT NOT NULL,  
    NifEditora INT NOT NULL,  
    FOREIGN key (NifEditora) REFERENCES editora(NifEditora)  
);
```

**-- Tabela encomendou**

```
CREATE TABLE encomendou (
    IdExemplar INT NOT NULL,
    ISBN INT NOT NULL,
    IdEncomenda INT NOT NULL,
    PRIMARY key (IdExemplar, ISBN, IdEncomenda),
    FOREIGN key (IdExemplar) REFERENCES exemplar(IdExemplar),
    FOREIGN key (ISBN) REFERENCES livro(ISBN),
    FOREIGN key (IdEncomenda) REFERENCES encomenda(IdEncomenda)
);
```

**-- Tabela factura**

```
CREATE TABLE factura (
    IdFactura INT PRIMARY KEY,
    Nif INT NOT NULL,
    Valor INT NOT NULL,
    Mes INT CHECK (Mes BETWEEN 1 AND 12) NOT NULL,
    Ano INT NOT NULL,
    FOREIGN key (Nif) REFERENCES leitor(Nif)
);
```

**-- Tabela aluguer**

```
CREATE TABLE aluguer (
    IdAluguer INT PRIMARY KEY,
    Semana INT CHECK (Semana BETWEEN 1 AND 52) NOT NULL,
    Ano INT NOT NULL,
    Valor INT NOT NULL,
```

```
Nif INT NOT NULL,  
IdExemplar INT NOT NULL,  
FOREIGN key (Nif) REFERENCES leitor(Nif),  
FOREIGN key (IdExemplar) REFERENCES exemplar(IdExemplar)  
);
```

## 9 – EXPRESSES SQL PARA INSERÇÃO DE DADOS

### -- Inserção de 10 leitores, 5 regulares e 5 frequentes

```
INSERT INTO leitor (Nome, NIF, Email, Nacionalidade, Tipo) VALUES  
('João Maria', 1001, 'joao@mail.com', 'Português', 'regular'),  
('Maria Beatriz', 1002, 'maria@mail.com', 'Português', 'regular'),  
('Filipe Santos', 1003, 'filipe@mail.com', 'Português', 'regular'),  
('Ana Filipa', 1004, 'ana@mail.com', 'Português', 'regular'),  
('Pedro Matos', 1005, 'pedro@mail.com', 'Português', 'regular'),  
('Lucas Marques', 1006, 'lucas@mail.com', 'Português', 'frequente'),  
('Rita Costa', 1007, 'rita@mail.com', 'Português', 'frequente'),  
('Beatriz Almeida', 1008, 'beatriz@mail.com', 'Português', 'frequente'),  
('Francisco Gonçalves', 1009, 'francisco@mail.com', 'Português', 'frequente'),  
('Sofia Trindade', 1010, 'sofia@mail.com', 'Português', 'frequente');
```

### -- Inserção de 3 autores

```
INSERT INTO autor (Nome, CodA, Nacionalidade) VALUES  
('José Saramago', 01, 'Português'),  
('Fernando Pessoa', 02, 'Português'),
```

('Eça de Queirós', 03, 'Português');

### -- Inserção de 2 editoras

```
INSERT INTO editora (Nome, NifEditora, Email) VALUES
('Porto Editora', 101, 'portoeditora@gmail.com'),
('LeYa', 102, 'leyaeditora@gmail.com');
```

### -- Inserção de 10 obras

```
INSERT INTO livro (Titulo, ISBN, NifEditora) VALUES
('Memorial do Convento', 10001, 101),
('Ensaio sobre a Cegueira', 10002, 101),
('O Homem Duplicado', 10003, 101),
('Mensagem', 10004, 101),
('Poemas de Alberto Caeiro', 10005, 101),
('O Guardador de Rebanhos', 10006, 102),
('Os Maias', 10007, 102),
('A Cidade e as Serras', 10008, 102),
('A Relíquia', 10009, 102),
('Caim', 10010, 102);
```

### -- 10 obras de 3 autores diferentes, 2 obras têm dois autores (10005 e 10010)

```
INSERT INTO autoria (ISBN, CodA) VALUES
(10001, 01),
(10002, 01),
(10003, 01),
(10004, 02),
```

(10005, 02),  
(10005, 03),  
(10006, 02),  
(10007, 03),  
(10008, 03),  
(10009, 03),  
(10010, 01),  
(10010, 02);

### -- 3 Exemplares para cada obra

```
INSERT INTO exemplar (IdExemplar, ISBN, LinkDownload) VALUES  
(01, 10001, 'http://example.com/memorial01'),  
(02, 10001, 'http://example.com/memorial02'),  
(03, 10001, 'http://example.com/memorial03'),  
(04, 10002, 'http://example.com/ensaio01'),  
(05, 10002, 'http://example.com/ensaio02'),  
(06, 10002, 'http://example.com/ensaio03'),  
(07, 10003, 'http://example.com/homemduplicado01'),  
(08, 10003, 'http://example.com/homemduplicado02'),  
(09, 10003, 'http://example.com/homemduplicado03'),  
(10, 10004, 'http://example.com/mensagem01'),  
(11, 10004, 'http://example.com/mensagem02'),  
(12, 10004, 'http://example.com/mensagem03'),  
(13, 10005, 'http://example.com/poemasalbertocaeiro01'),  
(14, 10005, 'http://example.com/poemasalbertocaeiro02'),  
(15, 10005, 'http://example.com/poemasalbertocaeiro03'),  
(16, 10006, 'http://example.com/guardadorrebanhos01'),
```

(17, 10006, 'http://example.com/guardadorrebanhos02'),  
(18, 10006, 'http://example.com/guardadorrebanhos03'),  
(19, 10007, 'http://example.com/maias01'),  
(20, 10007, 'http://example.com/maias02'),  
(21, 10007, 'http://example.com/maias03'),  
(22, 10008, 'http://example.com/cidadeserras01'),  
(23, 10008, 'http://example.com/cidadeserras02'),  
(24, 10008, 'http://example.com/cidadeserras03'),  
(25, 10009, 'http://example.com/reliquia01'),  
(26, 10009, 'http://example.com/reliquia02'),  
(27, 10009, 'http://example.com/reliquia03'),  
(28, 10010, 'http://example.com/caim01'),  
(29, 10010, 'http://example.com/caim02'),  
(30, 10010, 'http://example.com/caim03');

**-- Cada obra tem 3 géneros (há 6 géneros diferentes: Romance, Fantasia, Mistério, Aventura, História e Biografia)**

INSERT INTO livro\_gen (ISBN, Genero) VALUES

(10001, 'Romance'),  
(10001, 'História'),  
(10001, 'Mistério'),  
(10002, 'Fantasia'),  
(10002, 'Aventura'),  
(10002, 'Mistério'),  
(10003, 'Biografia'),  
(10003, 'História'),  
(10003, 'Fantasia'),

(10004, 'Romance'),  
(10004, 'Aventura'),  
(10004, 'História'),  
(10005, 'Mistério'),  
(10005, 'Romance'),  
(10005, 'Aventura'),  
(10006, 'Fantasia'),  
(10006, 'Aventura'),  
(10006, 'Biografia'),  
(10007, 'História'),  
(10007, 'Biografia'),  
(10007, 'Romance'),  
(10008, 'Mistério'),  
(10008, 'Fantasia'),  
(10008, 'Aventura'),  
(10009, 'Fantasia'),  
(10009, 'História'),  
(10009, 'Aventura'),  
(10010, 'Biografia'),  
(10010, 'Mistério'),  
(10010, 'Aventura');

## -- Inserção dos alugueres nas primeiras 8 semanas (o leitor 1001 não tem nenhum aluguer nessas semanas)

```
INSERT INTO aluguer (IdAluguer, Semana, Ano, Valor, Nif, IdExemplar) VALUES
```

-- Alugueres do leitor 1002

(1, 1, 2024, 2, 1002, 01),

(2, 2, 2024, 2, 1002, 02),

(3, 3, 2024, 2, 1002, 03),

(4, 4, 2024, 2, 1002, 04),

(5, 5, 2024, 2, 1002, 05),

(6, 6, 2024, 2, 1002, 06),

(7, 7, 2024, 2, 1002, 07),

(8, 8, 2024, 2, 1002, 08),

#### -- Alugueres do leitor 1003

(9, 1, 2024, 2, 1003, 09),

(10, 2, 2024, 2, 1003, 10),

(11, 3, 2024, 2, 1003, 11),

(12, 4, 2024, 2, 1003, 12),

(13, 5, 2024, 2, 1003, 13),

(14, 6, 2024, 2, 1003, 14),

(15, 7, 2024, 2, 1003, 15),

(16, 8, 2024, 2, 1003, 16),

#### -- Alugueres do leitor 1004

(17, 1, 2024, 2, 1004, 17),

(18, 2, 2024, 2, 1004, 18),

(19, 3, 2024, 2, 1004, 19),

(20, 4, 2024, 2, 1004, 20),

(21, 5, 2024, 2, 1004, 21),

(22, 6, 2024, 2, 1004, 22),

(23, 7, 2024, 2, 1004, 23),

(24, 8, 2024, 2, 1004, 24),

#### -- Alugueres do leitor 1005

(25, 1, 2024, 2, 1005, 25),

(26, 2, 2024, 2, 1005, 26),

(27, 3, 2024, 2, 1005, 27),

(28, 4, 2024, 2, 1005, 28),

(29, 5, 2024, 2, 1005, 29),

(30, 6, 2024, 2, 1005, 30),

(31, 7, 2024, 2, 1005, 01),

(32, 8, 2024, 2, 1005, 02),

#### -- Alugueres do leitor 1006

(33, 1, 2024, 1, 1006, 03),

(34, 2, 2024, 1, 1006, 04),

(35, 3, 2024, 1, 1006, 05),

(36, 4, 2024, 1, 1006, 06),

(37, 5, 2024, 1, 1006, 07),

(38, 6, 2024, 1, 1006, 08),

(39, 7, 2024, 1, 1006, 09),

(40, 8, 2024, 1, 1006, 10),

(41, 1, 2024, 1, 1006, 11),

(42, 2, 2024, 1, 1006, 12),

(43, 3, 2024, 1, 1006, 13),

(44, 4, 2024, 1, 1006, 14),

(45, 5, 2024, 1, 1006, 15),

(46, 6, 2024, 1, 1006, 16),

(47, 7, 2024, 1, 1006, 17),

(48, 8, 2024, 1, 1006, 18),

(49, 1, 2024, 1, 1006, 19),

(50, 2, 2024, 1, 1006, 20),

(51, 3, 2024, 1, 1006, 21),

(52, 4, 2024, 1, 1006, 22),

(53, 5, 2024, 1, 1006, 23),

(54, 6, 2024, 1, 1006, 24),

(55, 7, 2024, 1, 1006, 25),

(56, 8, 2024, 1, 1006, 26),

#### -- Alugueres do leitor 1007

(57, 1, 2024, 1, 1007, 27),

(58, 2, 2024, 1, 1007, 28),

(59, 3, 2024, 1, 1007, 29),

(60, 4, 2024, 1, 1007, 30),

(61, 5, 2024, 1, 1007, 01),

(62, 6, 2024, 1, 1007, 02),

(63, 7, 2024, 1, 1007, 03),

(64, 8, 2024, 1, 1007, 04),

(65, 1, 2024, 1, 1007, 05),

(66, 2, 2024, 1, 1007, 06),

(67, 3, 2024, 1, 1007, 07),

(68, 4, 2024, 1, 1007, 08),

(69, 5, 2024, 1, 1007, 09),

(70, 6, 2024, 1, 1007, 10),

(71, 7, 2024, 1, 1007, 11),

(72, 8, 2024, 1, 1007, 12),

(73, 1, 2024, 1, 1007, 13),

(74, 2, 2024, 1, 1007, 14),

(75, 3, 2024, 1, 1007, 15),

(76, 4, 2024, 1, 1007, 16),

(77, 5, 2024, 1, 1007, 17),

(78, 6, 2024, 1, 1007, 18),

(79, 7, 2024, 1, 1007, 19),

(80, 8, 2024, 1, 1007, 20),

#### -- Alugueres do leitor 1008

(81, 1, 2024, 1, 1008, 21),

(82, 2, 2024, 1, 1008, 22),

(83, 3, 2024, 1, 1008, 23),

(84, 4, 2024, 1, 1008, 24),

(85, 5, 2024, 1, 1008, 25),

(86, 6, 2024, 1, 1008, 26),

(87, 7, 2024, 1, 1008, 27),

(88, 8, 2024, 1, 1008, 28),

(89, 1, 2024, 1, 1008, 29),

(90, 2, 2024, 1, 1008, 30),

(91, 3, 2024, 1, 1008, 01),

(92, 4, 2024, 1, 1008, 02),

(93, 5, 2024, 1, 1008, 03),

(94, 6, 2024, 1, 1008, 04),

(95, 7, 2024, 1, 1008, 05),

(96, 8, 2024, 1, 1008, 06),

(97, 1, 2024, 1, 1008, 07),

(98, 2, 2024, 1, 1008, 08),

(99, 3, 2024, 1, 1008, 09),

(100, 4, 2024, 1, 1008, 10),

(101, 5, 2024, 1, 1008, 11),

(102, 6, 2024, 1, 1008, 12),

(103, 7, 2024, 1, 1008, 13),

(104, 8, 2024, 1, 1008, 14),

#### -- Alugueres do leitor 1009

(105, 1, 2024, 1, 1009, 15),

(106, 2, 2024, 1, 1009, 16),

(107, 3, 2024, 1, 1009, 17),

(108, 4, 2024, 1, 1009, 18),

(109, 5, 2024, 1, 1009, 19),

(110, 6, 2024, 1, 1009, 20),

(111, 7, 2024, 1, 1009, 21),

(112, 8, 2024, 1, 1009, 22),

(113, 1, 2024, 1, 1009, 23),

(114, 2, 2024, 1, 1009, 24),

(115, 3, 2024, 1, 1009, 25),

(116, 4, 2024, 1, 1009, 26),

(117, 5, 2024, 1, 1009, 27),

(118, 6, 2024, 1, 1009, 28),

(119, 7, 2024, 1, 1009, 29),

(120, 8, 2024, 1, 1009, 30),

(121, 1, 2024, 1, 1009, 01),

(122, 2, 2024, 1, 1009, 02),

(123, 3, 2024, 1, 1009, 03),

(124, 4, 2024, 1, 1009, 04),

(125, 5, 2024, 1, 1009, 05),

(126, 6, 2024, 1, 1009, 06),

(127, 7, 2024, 1, 1009, 07),

(128, 8, 2024, 1, 1009, 08),

-- Alugueres do leitor 1010

(129, 1, 2024, 1, 1010, 09),

(130, 2, 2024, 1, 1010, 10),

(131, 3, 2024, 1, 1010, 11),

(132, 4, 2024, 1, 1010, 12),

(133, 5, 2024, 1, 1010, 13),

(134, 6, 2024, 1, 1010, 14),

(135, 7, 2024, 1, 1010, 15),

(136, 8, 2024, 1, 1010, 16),

(137, 1, 2024, 1, 1010, 17),

(138, 2, 2024, 1, 1010, 18),

(139, 3, 2024, 1, 1010, 19),

(140, 4, 2024, 1, 1010, 20),

(141, 5, 2024, 1, 1010, 21),

(142, 6, 2024, 1, 1010, 22),

(143, 7, 2024, 1, 1010, 23),

(144, 8, 2024, 1, 1010, 24),

(145, 1, 2024, 1, 1010, 25),

(146, 2, 2024, 1, 1010, 26),

(147, 3, 2024, 1, 1010, 27),

(148, 4, 2024, 1, 1010, 28),

(149, 5, 2024, 1, 1010, 29),

(150, 6, 2024, 1, 1010, 30),

(151, 7, 2024, 1, 1010, 08),

(152, 8, 2024, 1, 1010, 07);

**-- Inserção dos pagamentos de 9 leitores corretos e 1 com um pagamento inferior ao devido (leitor 1010 fez o pagamento incorreto)**

```
INSERT INTO factura (IdFactura, Nif, Valor, Mes, Ano) VALUES
```

```
(1, 1001, 2, 1, 2024),  
(2, 1001, 2, 2, 2024),  
(3, 1002, 10, 1, 2024),  
(4, 1002, 10, 2, 2024),  
(5, 1003, 10, 1, 2024),  
(6, 1003, 10, 2, 2024),  
(7, 1004, 10, 1, 2024),  
(8, 1004, 10, 2, 2024),  
(9, 1005, 10, 1, 2024),  
(10, 1005, 10, 2, 2024),  
(11, 1006, 13, 1, 2024),  
(12, 1006, 13, 2, 2024),  
(13, 1007, 13, 1, 2024),  
(14, 1007, 13, 2, 2024),  
(15, 1008, 13, 1, 2024),  
(16, 1008, 13, 2, 2024),  
(17, 1009, 13, 1, 2024),  
(18, 1009, 13, 2, 2024),
```

-- O leitor 1010 devia ter pago 13€ em cada mes e apenas pagou 10€

```
(19, 1010, 10, 1, 2024),  
(20, 1010, 10, 2, 2024);
```

**-- Inserção de 3 avaliações por cada leitor de forma a que todos os livros tenham pelo menos uma avaliação (o livro 10010 não tem avaliações)**

```
INSERT INTO avaliacao (Nif, ISBN, Estrelas, Comentario) VALUES
```

**-- Avaliações do leitor 1001**

```
(1001, 10001, 4, 'Ótimo livro!'),  
(1001, 10002, 5, 'Incrível e emocionante.'),  
(1001, 10003, 3, 'Bastante interessante.'),
```

**-- Avaliações do leitor 1002**

```
(1002, 10004, 5, 'Fantástico, recomendo!'),  
(1002, 10005, 2, 'Pouco envolvente.'),  
(1002, 10006, 4, 'Uma leitura agradável.'),
```

**-- Avaliações do leitor 1003**

```
(1003, 10007, 3, 'Bom, mas não excepcional.'),  
(1003, 10008, 5, 'Adorei cada página!'),  
(1003, 10009, 4, 'Uma história fascinante.'),
```

**- Avaliações do leitor 1004**

```
(1004, 10001, 2, 'Não foi do meu agrado.'),  
(1004, 10003, 4, 'Muito bem escrito.'),  
(1004, 10004, 5, 'Uma obra-prima!'),
```

**- Avaliações do leitor 1005**

```
(1005, 10002, 3, 'Interessante, mas um pouco longo.'),  
(1005, 10005, 4, 'Gostei bastante.'),  
(1005, 10006, 5, 'Simplesmente maravilhoso!'),
```

**- Avaliações do leitor 1006**

```
(1006, 10007, 2, 'Esperava mais.'),  
(1006, 10008, 3, 'Boa leitura.'),
```

(1006, 10009, 5, 'Muito envolvente.'),

- Avaliações do leitor 1007

(1007, 10001, 5, 'Excelente leitura!'),

(1007, 10002, 4, 'Bem interessante.'),

(1007, 10003, 3, 'Achei mediano.'),

- Avaliações do leitor 1008

(1008, 10004, 5, 'Uma obra incrível!'),

(1008, 10005, 3, 'Uma leitura ok.'),

(1008, 10006, 4, 'Gostei da narrativa.'),

- Avaliações do leitor 1009

(1009, 10007, 5, 'Um dos melhores que já li!'),

(1009, 10008, 4, 'Muito bom!'),

(1009, 10009, 3, 'Esperava algo mais profundo.'),

- Avaliações do leitor 1010

(1010, 10001, 4, 'Um clássico excelente.'),

(1010, 10002, 2, 'Não me agradou muito.'),

(1010, 10003, 5, 'Simplesmente perfeito!');

**-- Amizades: o leitor 1001 é amigo de todos. O leitor 1010 não tem nenhum amigo. Todos os outros teem 2 ou mais amigos**

INSERT INTO amigos (NIF1, NIF2) VALUES

-- Leitor 1001 é amigo de todos os outros

(1001, 1002),

(1001, 1003),

(1001, 1004),

(1001, 1005),

(1001, 1006),

(1001, 1007),

(1001, 1008),

(1001, 1009),

-- Relações simétricas para 1001

(1002, 1001),

(1003, 1001),

(1004, 1001),

(1005, 1001),

(1006, 1001),

(1007, 1001),

(1008, 1001),

(1009, 1001),

-- Amizades dos leitores 1002 a 1009 (com pelo menos 2 amigos cada)

-- Leitor 1002

(1002, 1003),

(1002, 1004),

(1003, 1002),

(1004, 1002),

- Leitor 1003

(1003, 1005),

(1003, 1006),

(1005, 1003),

(1006, 1003),

- Leitor 1004

(1004, 1006),

(1004, 1007),

(1006, 1004),

(1007, 1004),

- Leitor 1005

(1005, 1007),

(1005, 1008),

(1007, 1005),

(1008, 1005),

- Leitor 1006

(1006, 1008),

(1006, 1009),

(1008, 1006),

(1009, 1006),

- Leitor 1007

(1007, 1008),

(1007, 1009),

(1008, 1007),

(1009, 1007),

- Leitor 1008

(1008, 1-009),

(1009, 1008);

**-- Encomendas: Inserção de 2 encomendas a duas editoras: uma com 2 obras e 2 exemplares de uma e 3 de outra. Outra com uma obra e um exemplar.**

```
INSERT INTO encomenda (IdEncomenda, DataEnc, Valor, NifEditora) VALUES
```

```
(1, '2024-01-3', 250, 101),
```

```
(2, '2024-01-5', 50, 102);
```

```
INSERT INTO encomendou (IdExemplar, ISBN, IdEncomenda) VALUES
(01, 10001, 1),
(03, 10001, 1),
(10, 10004, 1),
(11, 10004, 1),
(12, 10004, 1),
(16, 10006, 2);
```

**-- Inserção extra de uma encomenda para ter resultado na alínea e)**

```
INSERT INTO encomenda (IdEncomenda, DataEnc, Valor, NifEditora) VALUES
```

```
(3, '2024-01-17', 50, 102);
```

```
INSERT INTO encomendou (IdExemplar, ISBN, IdEncomenda) VALUES
```

```
(19, 10007, 3);
```

## 10 – EXPRESSÕES EM ÁLGEBRA RELACIONAL E SQL

**a) Quais são os leitores regulares?**

**SQL:**

```
SELECT Nome
```

```
FROM leitor
```

```
WHERE Tipo = 'regular';
```

**Álgebra Relacional:**

$$\pi_{\text{Nome}}(\sigma_{\text{Tipo} = \text{'Regular'}}(\text{leitor}))$$

## b) Para cada género indique o título e autores dos livros

**SQL:**

```
WITH l_gen as (
    SELECT Titulo, livro.ISBN, Genero
    FROM livro natural inner join livro_gen),
autores as (
    SELECT ISBN, Nome
    FROM autor natural inner join autoria)
SELECT Genero, Titulo, Nome
FROM l_gen natural inner join autores;
```

**Álgebra Relacional:**

$$\begin{aligned} L\_Gen &\leftarrow \pi_{\text{Titolo}, \text{ISBN}, \text{Genero}} (\text{Livro} \bowtie \text{Livro\_Gen}) \\ \text{Autores} &\leftarrow \pi_{\text{ISBN}, \text{Nome}} (\text{autor} \bowtie \text{autoria}) \\ \pi_{\text{Genero}, \text{Titolo}, \text{Nome}}(L\_Gen \bowtie \text{autores}) \end{aligned}$$

## c) Para cada género indique o número de amigos

**SQL:**

```
SELECT Genero, COUNT(ISBN)
FROM livro_gen
GROUP by Genero;
```

**Álgebra Relacional:**

$$\text{Genero} \text{ } gCount(\text{ISBN}) \text{ (livro\_gen)}$$

## d) Para cada leitor frequente indique o número de amigos

### SQL:

```
WITH l_amigos as (
    SELECT NIF, NIF2
    FROM leitor, amigos
    WHERE Tipo = 'frequente' and NIF = NIF1)
SELECT NIF, COUNT(NIF2)
FROM l_amigos
GROUP by NIF;
```

### Álgebra Relacional:

$$\begin{aligned} L_{\text{amigos}} &\leftarrow \pi_{NIF, NIF2} (\sigma_{\text{Tipo} = \text{'Frequente'}} \wedge NIF=NIF2 (\text{leitor} \times \text{amigos})) \\ NIF \text{ } gCount(NIF2) \text{ } (L_{\text{amigos}}) \end{aligned}$$

## e) Para cada leitor indique todos os alugueres que fez: título do livro, semana, ano e valor a pagar

### SQL:

```
SELECT e.ISBN, COUNT(en.IdExemplar) AS NumeroDeExemplares
FROM encomenda enc
JOIN encomendou en ON enc.IdEncomenda = en.IdEncomenda
JOIN exemplar e ON en.IdExemplar = e.IdExemplar
WHERE EXTRACT(WEEK FROM enc.DataEnc) = 3 AND EXTRACT(YEAR FROM enc.DataEnc) = 2024
GROUP BY e.ISBN;
```

### Álgebra Relacional:

$$\begin{aligned} EncomendasSemana3 &\leftarrow \pi_{e.ISBN, em.IdExemplar} (\sigma_{EXTRACT(\text{WEEK FROM DtaEnd}) = 3 \wedge EXTRACT(\text{YEAR FROM DataEnc}) = 2024} \wedge en.IdExemplar = e.IdExemplar (\text{encomenda as enc} \times \text{encomendou as en} \times \text{exemplar as e})) \end{aligned}$$

ISBN  $\text{gCount}(\text{IdExemplar})$  (EncomendasSemana3)

**f) Para cada leitor indique todos os alugueres que fez: título do livro, semana, ano e valor a pagar.**

**SQL:**

```
WITH alug_exemp as (
    SELECT Nif, ISBN, Semana, Ano, Valor
    FROM aluguer natural inner join exemplar)
SELECT Nif, Titulo, Semana, Ano, Valor
FROM alug_exemp NATURAL INNER JOIN livro;
```

**Álgebra Relacional:**

$\text{Alug\_exemp} \leftarrow \pi_{\text{Nif}, \text{ISBN}, \text{Semana}, \text{Ano}, \text{Valor}}(\text{aluguer} \bowtie \text{exemplar})$

$\pi_{\text{NIF}, \text{Titulo}, \text{Semana}, \text{Ano}, \text{Valor}}(\text{alug\_exemp} \bowtie \text{livro})$

**g) Para cada livro indique todos os alugueres, leitor, semana, ano, valor a pagar**

**SQL:**

```
SELECT ISBN, Nif, Semana, Ano, Valor
FROM aluguer NATURAL INNER join exemplar;
```

**Álgebra Relacional:**

$\pi_{\text{ISBN}, \text{NIF}, \text{Semana}, \text{Ano}, \text{Valor}}(\text{aluguer} \bowtie \text{exemplar})$

**h) Para cada leitor indique o valor em dívida e o valor pago no final da semana 4**

**SQL:**

```
WITH AlugueresAteSemana4 AS (
    SELECT Nif, SUM(Valor) AS ValorDivida
    FROM aluguer
    WHERE Semana <= 4
    GROUP BY Nif),
PagamentosAteSemana4 AS (
    SELECT Nif, SUM(Valor) AS ValorPago
    FROM factura
    WHERE (Ano = 2024 AND Mes <= 1)
    GROUP BY Nif)

SELECT
-- COALESCE para preencher valores nulos com 0
COALESCE(a.Nif, p.Nif) AS Nif,
COALESCE(ValorDivida, 0) AS ValorDivida,
COALESCE(ValorPago, 0) AS ValorPago
FROM
AlugueresAteSemana4 a
FULL OUTER JOIN PagamentosAteSemana4 p
ON a.Nif = p.Nif;
```

**Álgebra Relacional:**

```
AluguerSemana4 ← ( $\sigma_{\text{Semana} \leq 4}(\text{aluguer})$ )
ValorDivida ← Nif  $\text{gSum}(\text{Valor})$  as dividas (AluguerSemana4)
FacturasAteJan ← ( $\sigma_{\text{Ano} = 2024 \wedge \text{Mês} \leq 1}(\text{factura})$ )
ValorPago ← Nif  $\text{gSum}(\text{Valor})$  as pago (FacturasAteJan)
```

$\text{DividaPagamento} \leftarrow \text{ValorDivida} \bowtie \text{Nif} = \text{NifValorPago}$

$\text{ApenasDivida} \leftarrow \text{ValorDivida} - \pi_{\text{Nif}}(\text{DividaPagamento})$

$\text{ApenasPagamento} \leftarrow \text{ValorPago} - \pi_{\text{Nif}}(\text{Divida Pagamento})$

$\text{ResultadoFinal} \leftarrow \text{DividaPagamento} \cup \text{ApenasDivida} \cup \text{ApenasPagamento}$

$\pi_{\text{Nif}}$ , COALESCE (Dividou,0) as divida,

COALESCE (Pago,0) as pago ( ResultadoFinal)

### i) Indique todas as avaliações do Memorial do Convento, leitor, estrelas e comentários

**SQL:**

```
SELECT NIF, Estrelas, Comentario
FROM avaliacao natural inner join livro
WHERE Titulo = 'Memorial do Convento';
```

**Álgebra Relacional:**

$\pi_{\text{NIF}}, \text{Estrelas}, \text{Comentarios}(\sigma_{\text{Titulo} = \text{'Memorial do Convento'}}(\text{avaliação} \bowtie \text{livro}))$

### j) Que leitores são amigos de leitores que avaliaram livros do género mistério e fantasia

**SQL:**

```
WITH misterio AS (
    SELECT ISBN
    FROM avaliacao NATURAL JOIN livro_gen
    WHERE Genero = 'Mistério'),
fantasia AS (
    SELECT ISBN
```

```

FROM avaliacao NATURAL JOIN livro_gen
WHERE Genero = 'Fantasia'),
livros AS (
    SELECT ISBN
    FROM misterio
INTERSECT
    SELECT ISBN
    FROM fantasia),
leitores AS (
    SELECT NIF
    FROM avaliacao NATURAL JOIN livros)
SELECT a.NIF1
FROM amigos a
JOIN leitores l ON a.NIF2 = l.NIF;

```

### **Álgebra Relacional:**

misterio  $\leftarrow \pi_{ISBN} (\sigma_{Genero = 'mistério'} (avaliação \bowtie livro\_Gen))$

fantasia  $\leftarrow \pi_{ISBN} (\sigma_{Genero = 'fantasia'} (avaliação \bowtie livro\_Gen))$

livros  $\leftarrow \pi_{ISBN} (\text{misterio}) \wedge \pi_{ISBN} (\text{fantasia})$

leitores  $\leftarrow \pi_{NIF} (\text{avaliação} \bowtie \text{livros})$

$\pi_{NIF1} (\sigma_{NIF2 = NIF} (\text{amigos} \times \text{leitores}))$

### **k) Que leitores alugaram mais de 10 livros mas 4 primeiras semanas de 2024**

#### **SQL:**

```

WITH leitor_liv as (
    SELECT Nif, COUNT(IdExemplar) as n

```

```

FROM aluguer
GROUP by Nif)

SELECT Nif

FROM aluguer natural inner join leitor_liv

WHERE n > 10 and Semana < 5 and Ano = 2024;

```

### **Álgebra Relacional:**

$\text{Leitor\_liv} \leftarrow \text{NIF } \text{GCount}(\text{IdExemplar}) \text{ as } n \text{ (Aluguer)}$

$\pi_{\text{NIF}} (\sigma_{n > 10 \wedge \text{Semana} < 5 \wedge \text{Ano} = 2024} (\text{aluguer} \bowtie \text{Leitor\_liv}))$

### **I) Que livros foram avaliados com uma média de estrelas superior a 3?**

#### **SQL:**

```

WITH liv_est as (
    SELECT ISBN, avg(Estrelas) as n
    FROM avaliacao
    GROUP by ISBN)

```

```

SELECT ISBN
FROM liv_est
WHERE n > 3;

```

### **Álgebra Relacional:**

$\text{Liv\_est} \leftarrow \text{ISBN } \text{Gavg}(\text{Estrelas}) \text{ as } n \text{ (avaliacao)}$

$\pi_{\text{ISBN}} (\sigma_{n > 3}(\text{liv\_est}))$

## m) Que leitores frequentes nunca alugaram o “Memorial do Convento”?

**SQL:**

```
WITH leitores as (
    SELECT leitor.NIF
    FROM leitor, aluguer, exemplar, livro
    WHERE Tipo = 'frequente' and Titulo = 'Memorial do Convento' and
    leitor.NIF = aluguer.Nif and aluguer.idexemplar = exemplar.IdExemplar and
    exemplar.ISBN = livro.ISBN )
    (SELECT NIF
    FROM leitor
    WHERE Tipo = 'frequente')
EXCEPT
    (SELECT NIF
    FROM leitores);
```

## Álgebra Relacional:

$$\text{leitores} \leftarrow \pi_{\text{leitor.NIF}} (\sigma_{\text{Tipo} = \text{'frequente'}} \wedge \text{Titulo} = \text{"Memorial do Convento"} \wedge
\text{leitor.NIF} = \text{aluguer.Nif} \wedge
\text{aluguer.IdExemplar} = \text{exemplar.IdExemplar} \wedge
\text{exemplar.ISBN} = \text{livro.ISBN}
(\text{leitor} \times \text{aluguer} \times \text{exemplar} \times \text{livro})$$

$$\pi_{\text{NIF}} (\sigma_{\text{Tipo} = \text{'frequente'}} (\text{leitor})) - \pi_{\text{NIF}} (\text{leitores})$$

## n) Qual é o leitor que alugou mais livros na semana 4?

**SQL:**

```
WITH alugueres AS (
    SELECT Nif, IdExemplar
    FROM aluguer
```

```

WHERE Semana = 4),
r AS (
    SELECT Nif, COUNT(IdExemplar) AS n
    FROM alugueres
    GROUP BY Nif)
SELECT Nif, n
FROM r
WHERE n = (SELECT MAX(n) FROM r);

```

### **Álgebra Relacional:**

$\text{alugueres} \leftarrow \pi_{Nif, IdExemplar} (\sigma_{\text{Semana} = 4}(\text{aluguer}))$   
 $r \leftarrow \text{NIF } g_{\text{Count}}(\text{IdExemplar}) \text{ as } n \text{ (alugueres)}$   
 $\text{Max\_n} \leftarrow g_{\text{Max}}(n)(r)$   
 $\pi_{Nif} (\sigma_{r.n = \text{Max\_n}.n}(r \bowtie \text{Max\_n}))$

**o) Qual é o livro que foi mais alugado nas primeiras 8 semanas de 2024?**

### **SQL:**

```

WITH alugados AS (
    SELECT ISBN, aluguer.IdExemplar
    FROM aluguer
    NATURAL INNER JOIN exemplar
    WHERE Semana < 9 AND Ano = 2024),
r AS (
    SELECT ISBN, COUNT(IdExemplar) AS n
    FROM alugados
    GROUP BY ISBN)

```

```
SELECT ISBN, n
FROM r
WHERE n = (SELECT MAX(n) FROM r);
```

### **Álgebra Relacional:**

$\text{alugados} \leftarrow \pi_{\text{ISBN}, \text{aluguer.IdExemplar}} (\sigma_{\text{Semana} = 8 \wedge \text{Ano} = 2024} (\text{aluguer} \bowtie \text{exemplar}))$

$r \leftarrow \text{ISBN } gCount(\text{IdExemplar}) \text{ as } n \text{ (alugados)}$

$\text{Max\_n} \leftarrow gMax(n)(r)$

$\pi_{\text{ISBN}} (\sigma_{r.n = \text{Max\_n}.n} (r \bowtie \text{Max\_n}))$

**p) Que leitores são amigos de todos os leitores que alugaram o “Memorial do Convento”?**

### **SQL:**

```
SELECT L.NIF
FROM Leitor L
WHERE NOT EXISTS (
    SELECT 1
    FROM (
        -- Subconsulta que seleciona os leitores que alugaram o livro
        SELECT DISTINCT Aluguer.NIF
        FROM Aluguer
        JOIN Exemplar ON Aluguer.IdExemplar = Exemplar.IdExemplar
        WHERE Exemplar.ISBN = (SELECT ISBN FROM Livro WHERE Titulo =
        'Memorial do Convento')
    ) Alugadores
    WHERE NOT EXISTS (
        -- Verificar se o leitor L é amigo do leitor da subconsulta
        SELECT 1
```

```

FROM Amigos
WHERE (Amigos.NIF1 = L.NIF AND Amigos.NIF2 = Alugadores.NIF)
OR (Amigos.NIF1 = Alugadores.NIF AND Amigos.NIF2 = L.NIF)
)
);

```

### **Álgebra Relacional:**

leitores  $\leftarrow \pi_{NIF} \text{ as } NIF2 (\sigma_{\text{Titulo} = \text{'Memorial do convento'}} \wedge$   
 $\text{aluguer.IdExemplar} = \text{exemplar.idExemplar} \wedge$   
 $\text{exemplar.ISBN} = \text{livro.ISBN}$  (aluguer x exemplar x livro))

$\pi_{NIF1, NIF2}(\text{amigos}) \div \pi_{NIF2}(\text{leitores})$

### **q) Que leitores não tem amigos?**

#### **SQL:**

WITH leitores as (

```

SELECT NIF1
FROM amigos

```

(SELECT NIF

FROM leitor)

EXCEPT

(SELECT NIF1

FROM leitores);

### **Álgebra Relacional:**

leitores  $\leftarrow \pi_{NIF1}(\text{amigos})$

$\pi_{NIF}(\text{leitor}) - \pi_{NIF1}(\text{leitores})$

## r) Que leitores tem os pagamentos em dia?

**SQL:**

```
WITH DvidaPorLeitor AS (
    SELECT Nif, SUM(Valor) AS TotalDvida
    FROM aluguer
    GROUP BY Nif),
PagamentoPorLeitor AS (
    SELECT Nif, SUM(Valor) AS TotalPago
    FROM factura
    GROUP BY Nif),
Comparacao AS (
    SELECT
        -- COALESCE para preencher valores nulos com 0
        COALESCE(d.Nif, p.Nif) AS Nif,
        COALESCE(d.TotalDvida, 0) AS TotalDvida,
        COALESCE(p.TotalPago, 0) AS TotalPago
    FROM DvidaPorLeitor d
    FULL OUTER JOIN PagamentoPorLeitor p
    ON d.Nif = p.Nif
)
-- Leitores com pagamentos em dia
SELECT Nif
FROM Comparacao
WHERE TotalPago >= TotalDvida;
```

## Álgebra Relacional:

**TotalDivida**  $\leftarrow \text{Nif } g\text{Sum} (\text{Valor}) \text{ as divida (aluguer)}$

**TotalPago**  $\leftarrow \text{Nif } g\text{Sum} (\text{Valor}) \text{ as pago (aluguer)}$

**DividaPagamento**  $\leftarrow (\text{TotalDivida} \bowtie \text{Nif} = \text{Nif} \text{TotalPago}) \cup$

$(\text{TotalDivida} - \pi_{\text{Nif}} (\text{TotalDivida} \bowtie \text{Nif} = \text{Nif})) \cup$

$(\text{TotalPago} - \pi_{\text{Nif}} (\text{TotalDivida} \bowtie \text{Nif} = \text{Nif}))$

$\pi_{\text{Nif}} (\sigma_{\text{Pago}} \geq \text{divida} (\text{DividaPagamento}))$

**s) Para cada livro indique o número total de exemplares encomendados?**

**SQL:**

```
SELECT ISBN, count(IdExemplar)
```

```
FROM encomendou
```

```
GROUP by ISBN;
```

## Álgebra Relacional:

**ISBN**  $g\text{Count} (\text{IdExemplar})$  (**encomendou**)

**t) Que livros já tiveram todos os seus exemplares alugados?**

**SQL:**

```
WITH Total_Exemplares AS (
```

```
    SELECT Exemplar.ISBN, COUNT(Exemplar.IdExemplar) AS Total
```

```
    FROM Exemplar
```

```
GROUP BY Exemplar.ISBN),  
Exemplares_Alugados AS (  
    SELECT Exemplar.ISBN, COUNT(DISTINCT Aluguer.IdExemplar) AS  
    Alugado  
        FROM Aluguer  
        JOIN Exemplar ON Aluguer.IdExemplar = Exemplar.IdExemplar  
        GROUP BY Exemplar.ISBN)  
SELECT L.Titulo  
FROM Livro L  
JOIN Total_Exemplares TE ON L.ISBN = TE.ISBN  
JOIN Exemplares_Alugados EA ON L.ISBN = EA.ISBN  
WHERE TE.Total = EA.Alugados;
```

### Álgebra Relacional:

```
Total_Exemp ← ISBN gCount (IdExemplar) as n (exemplar)  
Total_Alug ← ISBN gCount (IdExemplar) as m (aluguer ↗ exemplar)  
Total ←  $\pi_{ISBN}(\sigma_{n = m} (Total_Exemp \bowtie Total_Alug))$ 
```

