

Sistema de Bibliotecas

Bancos de dados

Prof.^a Iana Daya Cavalcante Fagundo Passos

Joab Silva

Ayrton Medeiros

Caique Oliveira



Instituto Federal da Paraíba

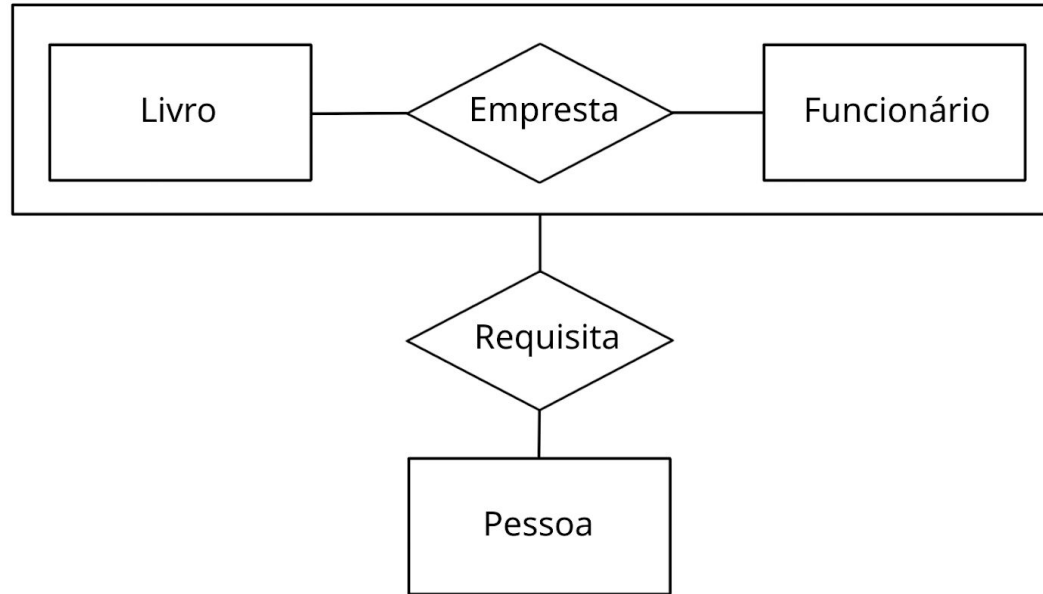
Diretrizes do Sistema de Bibliotecas:

Construir um Sistema de Bibliotecas. O acervo de uma biblioteca é composto por exemplares de livros. Cada livro é caracterizado por um ou mais autores, um título, uma editora, local de edição, um código ISBN e um conjunto de palavras-chave. A biblioteca possui pelo menos um exemplar de cada livro, numerados sequencialmente (exemplares 1, 2, 3, etc). Os associados da biblioteca podem retirar exemplares dos livros. Cada associado pode ter emprestados no máximo três exemplares. Para cada empréstimo, é registrada a data em que este foi realizado. Cada associado possui um código, nome e endereço.

As funcionalidades básicas são:

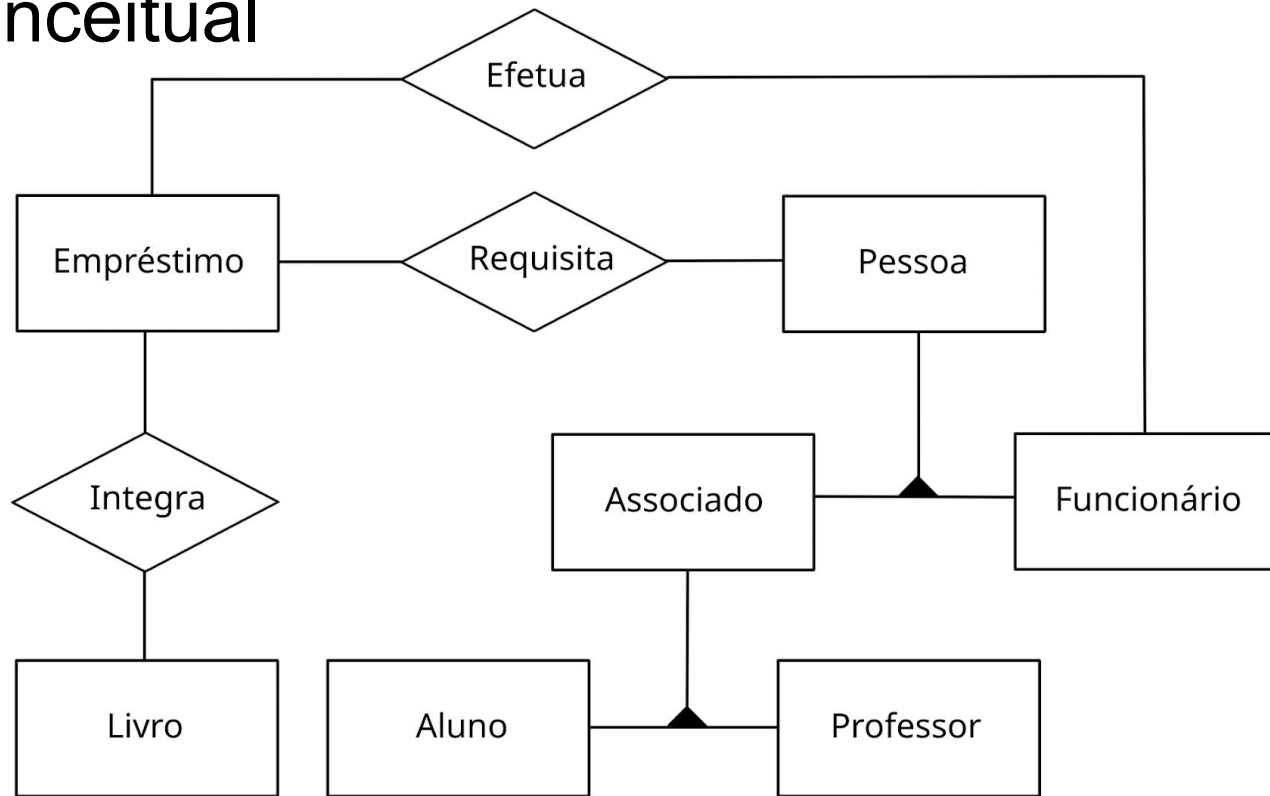
1. Cadastro dos livros com autores, título, editora, local da edição, código ISBN, conjunto de palavras-chave;
2. Cadastro dos associados com código (único), nome e endereço (rua, número, sala, cidade, cep, UF);
3. Cadastro dos empréstimos com número do empréstimo, data em que foi realizado;

Modelo Conceitual



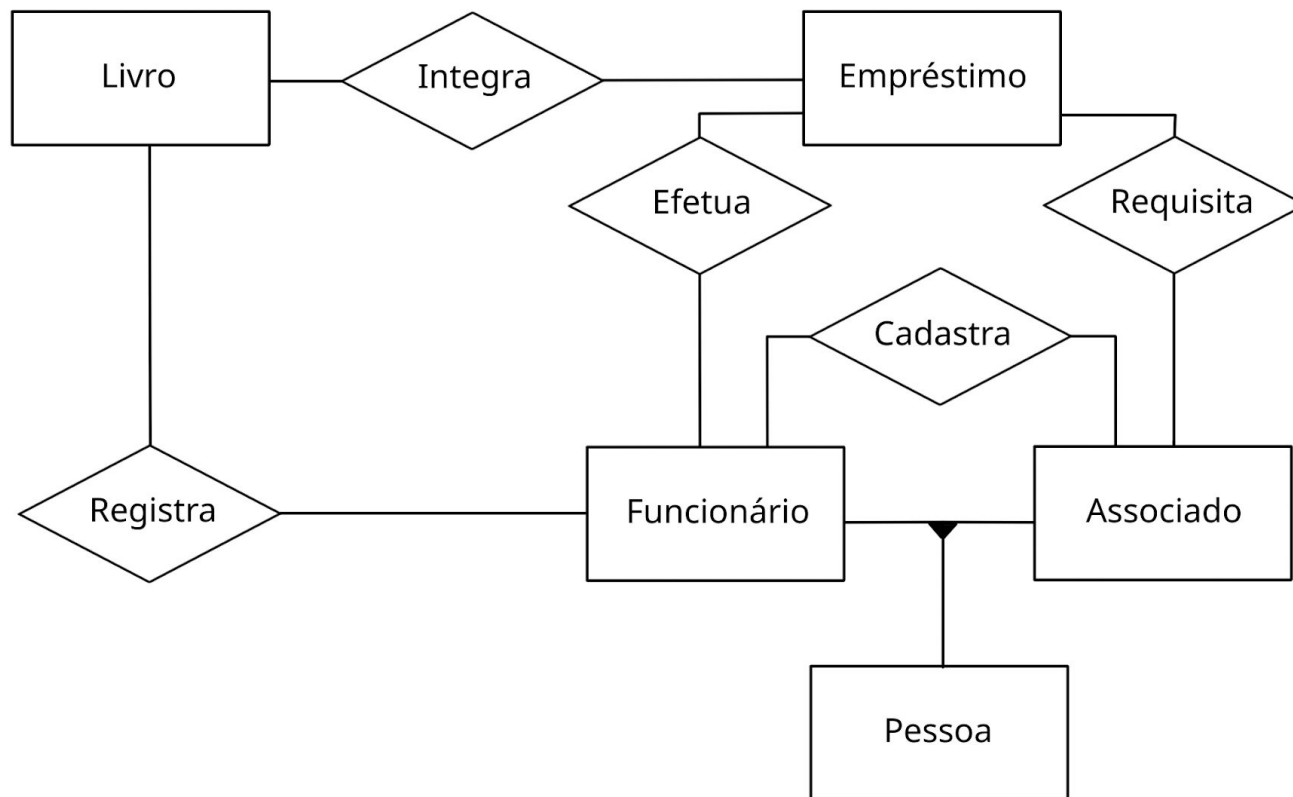
ver. 1

Modelo Conceitual

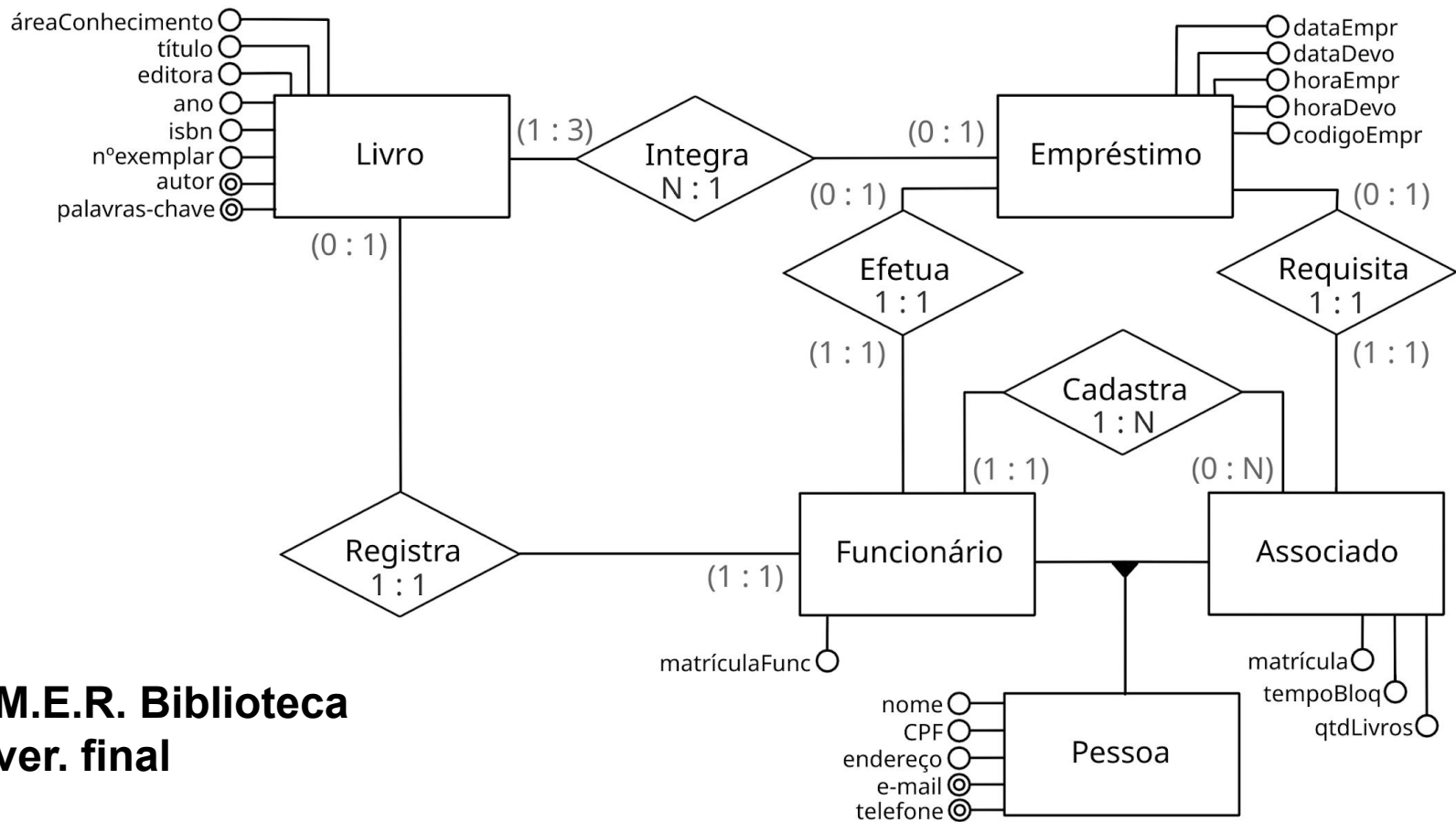


ver. 2

Modelo Conceitual



ver. 3



M.E.R. Biblioteca
ver. final

Modelo Relacional

LIVRO(titulo, nexemplar, isbn, areaConhecimento, editora, matricula_funcionario, id_emprestimo)

AUTOR_LIVRO(isbn_livro, nome_autor)

PALAVRA_CHAVE_LIVRO(isbn_livro, palavras_chave)

EMPRESTIMO(codigo_empr, data_e, hora_e, data_d, hora_d, quant_livros, matricula_funcionario, matricula_associado)

EMAIL_ASSOCIADO(matricula_associado, email)

EMAIL_FUNCIONARIO(matricula_funcionario, email)

TELEFONE_PESSOA(matricula_associado, telefone)

TELEFONE_FUNCIONARIO(matricula_funcionario, telefone)

FUNCIONARIO(nome, cpf, endereço, matricula)

ENDERECO(cep, rua, numero, bairro, cidade)

Código Sql

```
CREATE TABLE FUNCIONARIO(  
matricula INT,  
nome VARCHAR(100) NOT NULL,  
cpf INT NOT NULL,  
cep INT NOT NULL,  
rua VARCHAR(100) NOT NULL,  
numero INT NOT NULL,  
bairro VARCHAR(100) NOT NULL,  
cidade VARCHAR (100) NOT NULL,  
PRIMARY KEY (matricula)  
);
```

```
CREATE TABLE ASSOCIADO(  
matricula int,  
nome VARCHAR(100) not null,  
cpf int unique not null,  
tempo_bloqueio INT,  
qtd_livro INT,  
cep INT NOT NULL,  
rua VARCHAR(100) NOT NULL,  
numero INT NOT NULL,  
bairro VARCHAR(100) NOT NULL,  
cidade VARCHAR (100) NOT NULL,  
matricula_funcionario INT NOT NULL,  
FOREIGN KEY (matricula_funcionario) REFERENCES FUNCIONARIO (matricula),  
PRIMARY KEY (matricula)  
);
```

```
CREATE TABLE EMAIL_ASSOCIADO(  
matricula_associado int,  
email varchar(100),  
FOREIGN KEY (matricula_associado) REFERENCES ASOCIADO (matricula),  
PRIMARY KEY (matricula_associado, email)  
);
```

```
CREATE TABLE EMAIL_FUNCIONARIO(  
matricula_funcionario int,  
email varchar(100),  
FOREIGN KEY (matricula_funcionario) REFERENCES FUNCIONARIO (matricula),  
PRIMARY KEY (matricula_funcionario, email)  
);
```

```
CREATE TABLE TELEFONE_ASSOCIADO(  
matricula_associado INT,  
telefone INT,  
FOREIGN KEY (matricula_associado) REFERENCES ASSOCIADO (matricula),  
PRIMARY KEY (matricula_associado, telefone)  
);
```

```
CREATE TABLE TELEFONE_FUNCIONARIO(  
matricula_funcionario INT,  
telefone INT,  
FOREIGN KEY (matricula_funcionario) REFERENCES FUNCIONARIO (matricula),  
PRIMARY KEY (matricula_funcionario, telefone)  
);
```

```
CREATE TABLE EMPRESTIMO(  
  codigo_empr INT,  
  data_e DATE NOT NULL,  
  hora_e TIME (0) NOT NULL,  
  data_d DATE NOT NULL,  
  hora_d TIME (0) NOT NULL,  
  quant_livros INT NOT NULL,  
  matricula_funcionario INT NOT NULL,  
  matricula_associado INT NOT NULL,  
  
  FOREIGN KEY (matricula_funcionario) REFERENCES FUNCIONARIO (matricula),  
  FOREIGN KEY (matricula_associado) REFERENCES ASSOCIADO (matricula),  
  PRIMARY KEY (codigo_empr)  
);
```

```
CREATE TABLE LIVRO (  
    isbn INT,  
    nExemplar INT ,  
    titulo varchar(100) NOT NULL,  
    areaConhecimento VARCHAR(100),  
    editora VARCHAR (100) NOT NULL,  
    matricula_funcionario INT,  
    id_emprestimo INT,  
  
    PRIMARY KEY (isbn, nExemplar),  
    FOREIGN KEY (matricula_funcionario) REFERENCES FUNCIONARIO (matricula),  
    FOREIGN KEY (id_emprestimo) REFERENCES EMPRESTIMO (codigo_empr)  
);
```

```
CREATE TABLE AUTOR_LIVRO(  
    isbn_livro INT,  
    nome_autor VARCHAR(100),  
  
    FOREIGN KEY (isbn_livro) REFERENCES LIVRO (isbn),  
    PRIMARY KEY (isbn_livro, nome_autor)  
);
```

```
CREATE TABLE PALAVRA_CHAVE_LIVRO(  
    isbn_livro INT,  
    palavra_chave varchar(100),  
  
    FOREIGN KEY (isbn_livro) REFERENCES LIVRO (isbn),  
    PRIMARY KEY (isbn_livro, palavra_chave)  
);
```

Inserção de Valores

```
INSERT INTO FUNCIONARIO VALUES(101001, "Ayrton", 212112, 1010, "1 de janeiro", 350, "Centro", "Campina Grande");
INSERT INTO FUNCIONARIO VALUES(101002, "Caique", 323223, 1313, "1 de abril", 40, "Santa Rosa", "Campina Grande");
INSERT INTO FUNCIONARIO VALUES(101003, "Joab", 434334, 6868, "24 de fevereiro", 2, "Dinamerica", "Campina Grande");
INSERT INTO ASSOCIADO VALUES(123001, "Angelo", 123987, 40, 0, 9898, "Frei Caneca", 12, "Liberdade", "Campina Grande",
101001);
INSERT INTO ASSOCIADO VALUES(123002, "Bianca", 369741, 0, 2, 7777, "Tiradentes", 55, "Centenario", "Campina Grande",
101001);
INSERT INTO ASSOCIADO VALUES(123003, "Caio", 456951, 0, 1, 1314, "1 de abril", 41, "Santa Rosa", "Campina Grande", 101002);
INSERT INTO ASSOCIADO VALUES(123004, "Debora", 753258, 2, 0, 3434, "Av. Floriano Peixoto", 32, "Centenario", "Campina
Grande", 101003);
INSERT INTO ASSOCIADO VALUES(123005, "Edgar", 789951, 0, 1, 9090, "Juscelino Kubitschek", 24, "Prata", "Campina Grande",
101001);
INSERT INTO ASSOCIADO VALUES(123006, "Fernanda", 201937, 0, 3, 3939, "Dom Pedro I", 16, "Centro", "Campina Grande",
101002);
INSERT INTO EMAIL_ASSOCIADO VALUES(123001, "Bianca@gmail.com");
INSERT INTO EMAIL_ASSOCIADO VALUES(123002, "Caio@gmail.com");
INSERT INTO EMAIL_ASSOCIADO VALUES(123003, "Debora@gmail.com");
INSERT INTO EMAIL_ASSOCIADO VALUES(123004, "Edgar@gmail.com");
```



```
INSERT INTO EMAIL_ASSOCIADO VALUES(123005, "Fernanda@gmail.com");
INSERT INTO EMAIL_FUNCIONARIO VALUES(101001, "ayrton@gmail.com");
INSERT INTO EMAIL_FUNCIONARIO VALUES(101002, "caique@gmail.com");
INSERT INTO EMAIL_FUNCIONARIO VALUES(101003, "joab@gmail.com");
INSERT INTO EMPRESTIMO VALUES(40400, "2019-03-05", "09:47:00", "2019-06-12", "09:47:00", 1, 101003, 123001);
INSERT INTO EMPRESTIMO VALUES(40401, "2019-11-27", "15:37:13", "2019-12-03", "15:37:13", 1, 101002, 123002);
INSERT INTO EMPRESTIMO VALUES(40402, "2019-10-31", "12:12:12", "2019-12-09", "12:12:12", 1, 101002, 123002);
INSERT INTO EMPRESTIMO VALUES(40403, "2019-11-15", "13:10:12", "2019-12-09", "13:10:12", 1, 101002, 123003);
INSERT INTO EMPRESTIMO VALUES(40404, "2019-11-26", "19:40:52", "2019-12-05", "19:40:52", 2, 101003, 123004);
INSERT INTO EMPRESTIMO VALUES(40405, "2019-09-30", "12:20:52", "2019-10-12", "12:20:52", 1, 101001, 123005);
INSERT INTO EMPRESTIMO VALUES(40406, "2019-10-30", "20:05:58", "2019-12-09", "20:05:58", 1, 101001, 123005);
INSERT INTO EMPRESTIMO VALUES(40407, "2019-11-27", "17:34:03", "2019-12-12", "17:34:03", 3, 101001, 123006);
INSERT INTO LIVRO VALUES(40028920, 10, "Capitães da Areia", "Literatura", "Companhia de Bolso", 101002, 40400);
INSERT INTO LIVRO VALUES(40028921, 11, "Capitães da Areia", "Literatura", "Companhia de Bolso", 101002, 40401);
INSERT INTO LIVRO VALUES(40028922, 20, "Calculo", "Matematica", "Trilha", 101003, 40402);
INSERT INTO LIVRO VALUES(40028923, 21, "Calculo", "Matematica", "Trilha", 101003, 40403);
INSERT INTO LIVRO VALUES(40028924, 22, "Calculo", "Matematica", "Trilha", 101003, 40404);
INSERT INTO LIVRO VALUES(40028925, 30, "Sistemas Operacionais", "Computacao", "Pearson", 101001, 40404);
INSERT INTO LIVRO VALUES(40028926, 31, "Sistemas Operacionais", "Computacao", "Pearson", 101001, 40405);
INSERT INTO LIVRO VALUES(40028927, 32, "Sistemas Operacionais", "Computacao", "Pearson", 101002, 40407);
INSERT INTO LIVRO VALUES(40028928, 33, "Sistemas Operacionais", "Computacao", "Pearson", 101002, 40406);
INSERT INTO LIVRO VALUES(40028929, 41, "Metodologia de Pesquisa Cientifica para Ciencia da
Computação", "Computacao", "Campus", 101001, 40407);
INSERT INTO LIVRO VALUES(40028930, 42, "Metodologia de Pesquisa Cientifica para Ciencia da
Computação", "Computacao", "Campus", 101001, 40407);
```

Sugestão para consultas:

- As diversas formas de buscar um livro na biblioteca (ISBN, título, autor, ...)
- Identificação do número de livros por autor, por editora...
- Relação de livros emprestados;
- Relação de clientes com o número máximo de livros emprestados;
- Identificação do mês ou meses onde foi registrado o maior número de empréstimos;
- Identificação do(s) cliente(s) que retirou mais livros na biblioteca;
- Proponha novas consultas/funcionalidades não listadas anteriormente.

Sugestão para consultas:

- As diversas formas de buscar um livro na biblioteca (ISBN, título, autor, ...)

```
SELECT isbn  
FROM livro;
```

```
SELECT titulo  
FROM livro;
```

- Identificação do número de livros por autor, por editora ..

```
SELECT COUNT(*)  
FROM livro  
WHERE areaConhecimento = "Matematica";
```

- Relação de livros emprestados;

```
SELECT livro.titulo, livro.nExemplar, emprestimo.codigo_empr  
FROM emprestimo, livro;
```

- Relação de clientes com o número máximo de livros emprestados;

```
SELECT qtd_livro, nome  
FROM associado  
WHERE qtd_livro = 3;
```

- Identificação do mês ou meses onde foi registrado o maior número de empréstimos;

```
SELECT COUNT(*), MONTH(data_e)  
FROM emprestimo  
GROUP BY MONTH(data_e)
```

- Identificação do(s) cliente(s) que retirou mais livros na biblioteca;

```
SELECT ASSOCIADO.nome, sum(EMPRESTIMO.quant_livros)  
FROM ASSOCIADO, EMPRESTIMO  
WHERE (EMPRESTIMO.matricula_associado = ASSOCIADO.matricula)  
GROUP BY ASSOCIADO.nome
```

Funções Propostas

- Listar todos associados em ordem crescente de nomes

```
SELECT * FROM associado  
ORDER BY nome ASC
```

- Listar livros por área de conhecimento

```
SELECT *  
FROM livro  
WHERE areaConhecimento="Literatura"
```

Dicionário de Dados

Tabela		FUNCIONARIO	
Descrição		Armazena as informações dos funcionários	
Observações		Essa tabela possui como chave primária matricula	
Campos			
Atributos	Tipo de dado	Tamanho	Restrições de domínio
matricula	INT		PRIMARY KEY
nome	VARCHAR	100	NOT NULL
cpf	INT		NOT NULL
tempo_bloqueio	INT		NOT NULL
cep	INT		NOT NULL
rua	VARCHAR	100	NOT NULL
numero	INT		NOT NULL
bairro	VARCHAR	100	NOT NULL
cidade	VARCHAR	100	NOT NULL

Tabela	ASSOCIADO		
Descrição	Armazena as informações dos associados		
Observações	Essa tabela possui como chave primária matricula		
Campos			
Atributos	Tipo de dado	Tamanho	Restrições de domínio
matricula	INT		PRIMARY KEY
nome	VARCHAR	100	NOT NULL
cpf	INT		UNIQUE NOT NULL
tempo_bloqueio	INT		
qtd_livro	INT		
cep	INT		NOT NULL
rua	VARCHAR	100	NOT NULL
numero	INT		NOT NULL
bairro	VARCHAR	100	NOT NULL
cidade	VARCHAR	100	NOT NULL
matricula_funcionario	INT		FOREIGN KEY

Tabela	EMAIL_ASSOCIADO		
Descrição	Armazena as informações dos Emails dos associados		
Observações	Essa tabela possui matricula associado como chave estrangeira matricula_associado		
Campos			
Atributos	Tipo de dado	Tamanho	Restrições de domínio
matricula_associado	INT		PRIMARY KEY, FOREIGN KEY
email	VARCHAR	100	PRIMARY KEY

Tabela	EMAIL_FUNCIONARIO		
Descrição	Armazena as informações dos Emails dos funcionários		
Observações	Essa tabela possui matricula funcionario como chave estrangeira matricula_funcionario		
Campos			
Atributos	Tipo de dado	Tamanho	Restrições de domínio
matricula_funcionario	INT		PRIMARY KEY, FOREIGN KEY
email	VARCHAR	100	PRYMARY KEY

Tabela	TELEFONE_ASSOCIADO		
Descrição	Armazena as informações dos telefones dos associados		
Observações	Essa tabela possui matricula associado como chave estrangeira		
matricula_associado			
Campos			
Atributos	Tipo de dado	Tamanho	Restrições de domínio
matricula_associado	INT		PRIMARY KEY, FOREIGN KEY
telefone	INT		PRYMARY KEY

Tabela	TELEFONE_FUNCIONARIO		
Descrição	Armazena as informações dos telefones dos funcionários		
Observações	Essa tabela possui matricula funcionario como chave estrangeira		
matricula_funcionario			
Campos			
Atributos	Tipo de dados	Tamanho	Restrições de domínio
matricula_funcionario	INT		PRIMARY KEY, FOREIGN KEY
telefone	INT		PRYMARY KEY

Tabela	EMPRESTIMO		
Descrição	Armazena as informações dos empréstimos de livros		
Observações	Essa tabela possui codigo_empr como chave primária		
Campos			
Atributos	Tipo de dados	Tamanho	Restrições de domínio
codigo_empr	INT		PRYMARY KEY
data_e	DATE		NOT NULL
hora_e	TIME(0)		NOT NULL
data_d	DATE		NOT NULL
hora_d	TIME(0)		NOT NULL
quant_livros	INT		NOT NULL
matricula_funcionario	INT		NOT NULL, FOREIGN KEY
matricula_associado	INT		NOT NULL, FOREIGN KEY

Tabela	LIVRO		
Descrições	Armazena as informações dos livro		
Observações	Essa tabela possui nExemplar como chave primária		
Campos			
Atributos	Tipo de dados	Tamanho	Restrições de domínio
Isbn	INT		PRIMARY KEY
nExemplar	INT		PRIMARY KEY
titulo	VARCHAR	100	NOT NULL
areaConhecimento	VARCHAR	100	
editora	VARCHAR	100	NOT NULL
matricula_funcionario	INT		FOREIGN KEY
Id_emprestimo	INT		FOREIGN KEY

Tabela	AUTOR_LIVRO		
Descrições	Armazena as informações dos autores e livros		
Observações	Essa tabela possui isbn_livro como chave estrangeira		
Campos			
Atributos	Tipo de dados	Tamanho	Restrições de domínio
isbn_livro	INT		PRYMARY KEY, FOREIGN KEY
nome_autor	VARCHAR	100	PRIMARY KEY

Tabela	PALAVRA_CHAVE_LIVRO		
Descrições	Armazena as informações das palavras chaves dos livros		
Observações	Essa tabela possui isbn_livro como chave estrangeira		
Campos			
Atributos	Tipos de dados	Tamanho	Restrições de domínio
isbn_livro	INT		PRYMARY KEY, FOREIGN KEY
palavra_chave	VARCHAR	100	PRYMARY KEY