



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
TECNOLOGIA EM SISTEMAS PARA INTERNET**

MARCELA BARRETO DE OLIVEIRA KRAMER
MATHEUS PEREIRA DE SOUSA
PABLO DE LIMA MARQUES ESTRELA
RICARDO LUIZ DUARTE PRADO

**PROJETO PARA BANCO DE DADOS RELACIONAL DE BIBLIOTECA
UNIVERSITÁRIA**

JOÃO PESSOA - PB
2022

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO PROJETO

O projeto em questão aborda o funcionamento de uma biblioteca universitária através de uma modelagem conceitual. O esquema é modelado na ferramenta brModelo e as informações são representadas com intuito de oferecer o máximo de detalhes possível.

1.2 MINIMUNDO

Uma biblioteca universitária precisa melhorar o seu sistema de gerenciamento próprio. Para isso, decide reformular a organização dos seus dados da seguinte forma: um usuário é cadastrado com matrícula, nome, data de nascimento, telefones, e-mails, endereço (rua, UF, CEP, cidade, bairro e número) e gênero (opcional). Além disso, é mantido um registro do crachá dos usuários, que, por sua vez, é registrado com um QR code, data de validade e a foto do usuário. A biblioteca também armazena as informações dos livros que possui, especificando o ISBN, o título, a editora, o número da edição e a sua classificação por assunto, assim como mantém o registro dos autores desses livros, detalhando o código do autor, seu nome e sua data de nascimento. Ademais, os funcionários que trabalham na biblioteca também são registrados, nesse caso, com CPF, nome, endereço (rua, UF, CEP, cidade, bairro e número), telefone, data de nascimento e e-mail. Por fim, seguem as demais regras de negócio:

- Um usuário possui apenas um crachá, bem como um crachá só identifica um único usuário.
- Um usuário pode ou não pegar emprestado vários exemplares de livros, assim como um mesmo exemplar de livro pode ou não ser emprestado a vários usuários. Esse empréstimo — do qual são registradas as datas de empréstimo e devolução — pode ser feito online ou por intermédio de um funcionário. Logo, o empréstimo pode ser

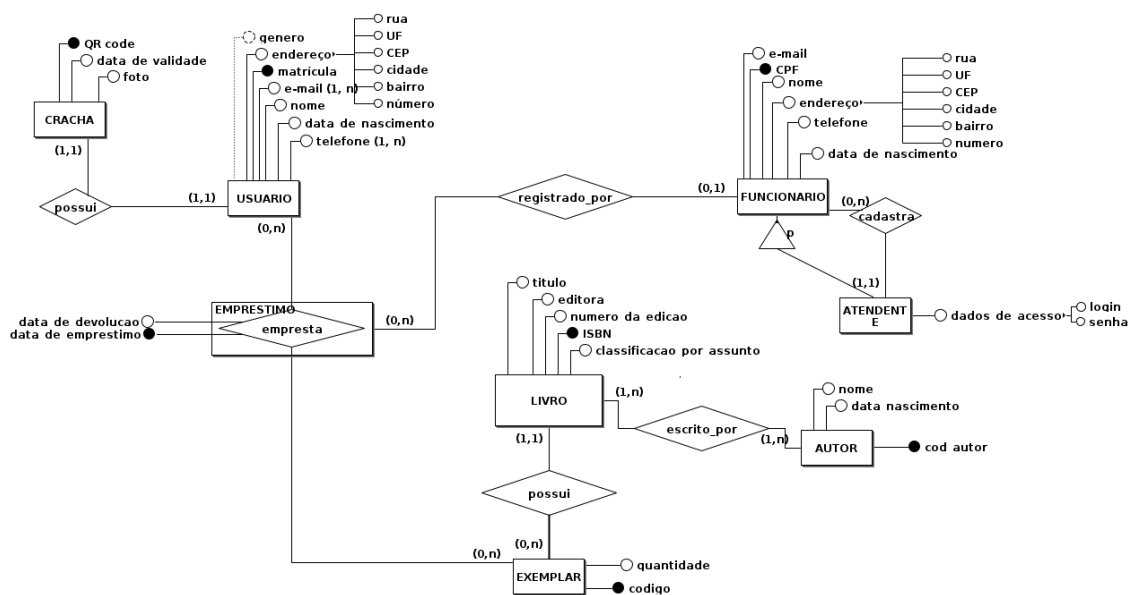
registrado ou não por um funcionário, bem como um funcionário pode registrar ou não vários empréstimos.

- Um livro pode ter ou não vários exemplares, sendo cada um desses registrado com um código identificador. No entanto, um exemplar só pode ser de um único livro e só existe caso haja esse livro correspondente.
- Um livro é escrito por um ou vários autores e um autor pode escrever um ou mais livros.
- Um funcionário pode ser especializado como atendente, possuindo, assim, dados de acesso ao sistema da biblioteca, nesse caso, login e senha. O atendente, por sua vez, é o responsável por cadastrar nenhum ou vários funcionários de maneira geral, mas um funcionário é cadastrado por um único atendente.

2 PROJETO CONCEITUAL

2.1 DIAGRAMA E-R

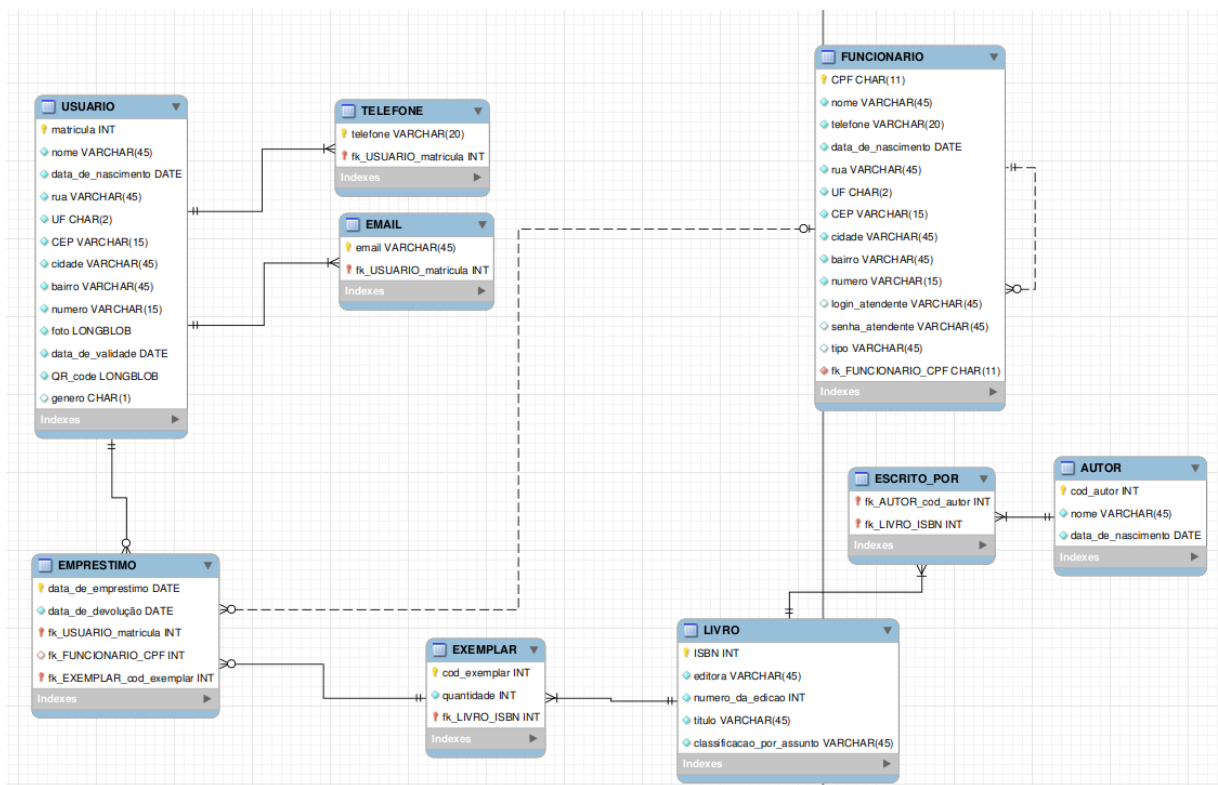
Figura 1 - Modelo conceitual da Biblioteca Universitária (brModelo)



3 PROJETO LÓGICO

3.1 DIAGRAMA RELACIONAL

Figura 2 - Modelo lógico da Biblioteca Universitária (MySQL Workbench)



3.2 DICIONÁRIO DE DADOS

TABELA AUTOR

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
cod_autor	INT	✓			✓		Código identificador do autor.
nome	VARCHAR (45)				✓		Nome do autor.

data_de_nascimento	DATE				✓		Data de nascimento do autor. Não pode assumir valores futuros.
--------------------	------	--	--	--	---	--	--

TABELA EMAIL

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
email	VARCHAR (45)	✓			✓		Endereço de e-mail usuário da biblioteca. Garantir a unicidade da chave primária(email, fk_USUARIO_matricula).
fk_USUARIO_matricula	INT	✓	✓		✓		Referencia a tabela USUARIO. Garantir a unicidade da chave primária(email, fk_USUARIO_matricula).

TABELA EMPRÉSTIMO

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
data_de_emprestimo	DATE	✓			✓	TODAY	Data de empréstimo realizado. Não pode assumir valor futuro. Garante unicidade da chave primária(data_de_emprestimo, fk_USUARIO_matricula, fk_EXEMPLAR_cod_exemplar).
data_de_devolucao	DATE				✓		Data de devolução de empréstimo realizado. Não pode assumir valores passados.
fk_USUARIO_matricula	INT	✓	✓		✓		Referencia a tabela USUARIO. Garante unicidade da chave primária(data_de_emprestimo, fk_USUARIO_matricula, fk_EXEMPLAR_cod_exemplar).
fk_FUNCIONARIO_CPF	INT		✓				Referencia a tabela FUNCIONARIO.
fk_EXEMPLAR_cod_exemplar	INT	✓	✓		✓		Referencia a tabela EXEMPLAR. Garante unicidade da chave primária(data_de_emprestimo, fk_USUARIO_matricula, fk_EXEMPLAR_cod_exemplar).

TABELA ESCRITO POR

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
fk_AUTOR_cod_autor	INT	✓	✓		✓		Referencia a tabela AUTOR. Garantir a unicidade da chave primária(fk_AUTOR_cod_autor, fk_LIVRO_ISBN).
fk_LIVRO_ISBN	INT	✓	✓		✓		Referência a tabela LIVRO. Garantir a unicidade da chave primária(fk_AUTOR_cod_autor, fk_LIVRO_ISBN).

TABELA EXEMPLAR

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
cod_exemplar	INT	✓			✓		Código de exemplar de livro. Garantir a unicidade da chave primária (cod_exemplar, fk_LIVRO_ISBN).
quantidade	INT				✓	1	Quantidade de exemplares disponíveis. Só pode assumir valores maiores que zero.
fk_LIVRO_ISBN	INT	✓	✓		✓		Referencia a tabela LIVRO. Garantir a unicidade da chave primária (cod_exemplar, fk_LIVRO_ISBN).

TABELA FUNCIONÁRIO

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
CPF	CHAR(11)	✓			✓		Cadastro de Pessoa Física do funcionário da biblioteca.
nome	VARCHAR(45)				✓		Nome do funcionário da biblioteca.
telefone	VARCHAR(20)				✓		Número de telefone do funcionário da biblioteca.
data_de_nascimento	DATE				✓		Data de nascimento do funcionário da biblioteca. Não pode assumir valores futuros.

rua	VARCHAR (45)				✓		Logradouro do funcionário da biblioteca.
UF	CHAR(2)				✓		Unidade Federativa do funcionário da biblioteca. Pode assumir apenas os valores: AM, AP, PA, PE, PB, PI, MA, BA, SE, CE, RJ, SP, MG, ES, RN, RS, SC, PR, TO, RR, RO, AC, DF.
CEP	VARCHAR (15)				✓		Código postal do funcionário da biblioteca.
cidade	VARCHAR (45)				✓		Cidade do funcionário da biblioteca.
bairro	VARCHAR (45)				✓		Bairro do funcionário da biblioteca.
numero	VARCHAR (15)				✓		Número do endereço do funcionário da biblioteca.
login	VARCHAR (45)			✓			Login do atendente da biblioteca. Se o tipo do funcionário for 'atendente', não pode assumir valor nulo.
senha	VARCHAR (45)						Senha do atendente da biblioteca. Se o tipo do funcionário for 'atendente', não pode assumir valor nulo.
tipo	VARCHAR (45)						Tipo do funcionário da biblioteca. Quando login e senha forem valores não nulos (o usuário é atendente), o tipo possui valor 'atendente'.
fk_FUNCIONARIO_CPF	INT		✓		✓		Referencia a tabela FUNCIONARIO.

TABELA LIVRO

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
ISBN	INT	✓			✓		Código identificador internacional do livro.

editora	VARCHAR (45)				✓		Nome da editora responsável pela publicação do livro.
numero_da_edicao	INT				✓	1	Número da edição do livro. Só pode assumir valores maiores que zero.
titulo	VARCHAR (45)				✓		Título do livro.
classificacao_por_assunto	VARCHAR (45)				✓		Classificação por assunto do livro.

TABELA TELEFONE

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
telefone	VARCHAR (20)	✓			✓		Número de telefone do usuário da biblioteca. Garantir a unicidade da chave primária(telefone, fk_USUARIO_matricula).
fk_USUARIO_matricula	INT	✓	✓		✓		Referencia a tabela USUARIO. Garantir a unicidade da chave primária(telefone, fk_USUARIO_matricula).

TABELA USUÁRIO

Coluna	Tipo	PK	FK	Único	NN	Default	Comentário
matricula	INT	✓			✓		Matrícula do usuário da biblioteca.
nome	VARCHAR (45)				✓		Nome do usuário da biblioteca.
data_de_nascimento	DATE				✓		Data de nascimento do usuário da biblioteca. Não pode assumir um valor futuro.
rua	VARCHAR (45)				✓		Logradouro do usuário da biblioteca.
UF	CHAR(2)				✓		Unidade Federativa do usuário da biblioteca. Pode assumir apenas os valores:

							AM, AP, PA, PE, PB, PI, MA, BA, SE, CE, RJ, SP, MG, ES, RN, RS, SC, PR, TO, RR, RO, AC, DF.
CEP	VARCHAR (15)				✓		Código postal do usuário da biblioteca.
cidade	VARCHAR (45)				✓		Cidade do usuário da biblioteca.
bairro	VARCHAR (45)				✓		Bairro do usuário da biblioteca.
numero	VARCHAR (15)				✓		Número do endereço do usuário da biblioteca.
foto	LONGBLOB			✓	✓		BLOB(Binary Large Object) da foto do crachá do usuário da biblioteca. Garantir unicidade da chave candidata.
data_de_validade	DATE				✓		Data de validade do crachá do usuário da biblioteca. Não pode assumir valores passados.
QR_code	LONGBLOB			✓	✓		LONGBLOB(Long Binary Large Object) do código de barras bidimensional do crachá do usuário da biblioteca. É gerado a partir da matrícula e nome do usuário. Garantir unicidade da chave candidata.
genero	CHAR(1)						Gênero do usuário da biblioteca. Pode assumir apenas os valores: 'M' ou 'F'.

4 SCRIPT SQL

```
/** Criação do banco de dados BibliotecaUniv */
CREATE DATABASE BibliotecaUniv;

/** Utilização do banco de dados BibliotecaUniv */
use BibliotecaUniv;

/** Criação da Tabela Autor */
create table Autor(
    cod_autor                int                not null ,
    nome                     varchar(45)        not null,
    data_de_nascimento       date               not null,
    constraint pk_autor      primary key (cod_autor),
    constraint ck_data_de_nascimento_autor check(data_de_nascimento < sysdate())
);

/** Criação da Tabela Livro */
create table Livro(
    ISBN                     int                not null,
    editora                  varchar(45)        not null,
    numero_da_edicao          int                default 1,
    titulo                   varchar(45)        not null,
    classificacao_por_assunto varchar(45)        not null,
    constraint pk_ISBN       primary key (ISBN),
    constraint ck_numero_da_edicao check(numero_da_edicao > 0),
    constraint ck_ISBN_livro check(ISBN > 0)
);

/** Criação da Tabela Exemplar */
create table Exemplar(
    cod_exemplar             int                not null,
    quantidade               int                not null,
    ISBN_livro               int                not null,
    constraint pk_exemplar   primary key (cod_exemplar,ISBN_livro),
    constraint fk_livro_exemplar foreign key (ISBN_livro) references Livro (ISBN)
);

/** Criação da Tabela Escrito_por */
create table Escrito_por(
    cod_autor                int                not null,
    ISBN_livro               int                not null,
    constraint fk_cod_autor  foreign key (cod_autor) references Autor
(cod_autor),
    constraint fk_livro_escrito_por foreign key (ISBN_livro) references Livro (ISBN),
    constraint pk_escrito_por primary key (cod_autor,ISBN_livro)
);

/** Criação da Tabela Usuário */
create table Usuario(
    matricula                int                not null,
```

```

nome                varchar(45)      not null,
data_de_nascimento  date          not null,
rua                 varchar(45)      not null,
UF                  char(2)         not null,
CEP                 varchar(15)     not null,
cidade              varchar(45)     not null,
bairro              varchar(45)     not null,
numero              varchar(15)     not null,
foto                longblob        not null,
data_de_validade    date            not null,
qr_code             longblob        not null,
genero              char(1)         null,
constraint pk_usuario primary key (matricula),
constraint ck_data_de_nascimento_usuario check(data_de_nascimento < sysdate()),
constraint ck_data_de_validade_usuario   check(data_de_validade > sysdate())
);

/** Criação da Tabela Email */
create table Email(
    email                varchar(45)      not null,
    matricula_usuario    int              not null,
    constraint fk_usuario_email foreign key (matricula_usuario) references Usuario
(matricula),
    constraint pk_email primary key (matricula_usuario,email)
);

/** Criação da Tabela Telefone */
create table Telefone(
    telefone              varchar(45)      not null,
    matricula_usuario     int              not null,
    constraint fk_usuario_telefone foreign key (matricula_usuario) references Usuario
(matricula),
    constraint pk_telefone primary key (matricula_usuario,telefone)
);

/** Criação da Tabela Funcionario */
create table Funcionario(
    cpf                  char(11)         not null,
    nome                 varchar(45)      not null,
    telefone             varchar(20)      not null,
    data_de_nascimento   date            not null,
    rua                  varchar(45)      not null,
    uf                   varchar(2)       not null,
    cep                  varchar(15)     not null,
    cidade               varchar(45)     not null,
    bairro               varchar(45)     not null,
    numero               varchar(15)     not null,
    login_atendente      varchar(45)     null,
    senha_atendente      varchar(45)     null,
    tipo                 varchar(45)     null,
    cpf_responsavel_cadastro char(11)    null,
    constraint pk_funcionario primary key (cpf),
    constraint fk_atendente_responsavel foreign key (cpf_responsavel_cadastro) references
Funcionario(cpf),

```

```

        constraint uk_login                unique (login_atendente),
        constraint ck_funcionario_cpf      check(char_length(cpf)=11),
        constraint ck_data_de_nascimento_funcionario check(data_de_nascimento < sysdate())
    );

    /** Criação da Tabela Empréstimo */
    create table Emprestimo(
        matricula_usuario                int                not null,
        cpf_funcionario                  char(11)           null,
        cod_exemplar                      int                not null,
        data_de_emprestimo                date               not null,
        data_de_devolucao                 date               not null,
        constraint fk_usuario_matricula    foreign key (matricula_usuario) references Usuario
        (matricula),
        constraint fk_cod_exemplar          foreign key (cod_exemplar) references Exemplar
        (cod_exemplar),
        constraint fk_cpf_funcionario       foreign key (cpf_funcionario) references
        Funcionario (cpf),
        constraint ck_data_de_emprestimo    check(data_de_emprestimo <= sysdate()),
        constraint ck_data_de_devolucao     check(data_de_devolucao > sysdate()),
        constraint pk_emprestimo            primary key
        (matricula_usuario,cod_exemplar,data_de_emprestimo)
    );

    /* Inserindo informações na tabela autor */
    insert into Autor(cod_autor,nome, data_de_nascimento)
    values(1,'Eva Heller', '1948-08-08'), /*psicologia das cores*/
    (2, 'Andrew Stuart Tanenbaum', '1944-03-16'),/* Sistemas Operativos Modernos*/
    (3, 'Charles Duhigg', '1974-06-06'), /*o poder do hábito*/
    (4, 'Stephen Hawking', '1942-01-08'), /*uma breve história do tempo*/
    (5, 'Karl Marx', '1818-05-05'),/* O capital*/
    (6, 'Candido', '1976-06-06') ;

    /* Inserindo informações na tabela livro*/
    insert into Livro(ISBN, editora, numero_da_edicao, titulo, classificacao_por_assunto)
    values(1, 'Pearson' , 4, 'Sistemas Operativos Modernos', 'tecnologia'),
    (2, 'olhares', 4, 'Psicologia das cores', 'designer' ),
    (3, 'objetiva', 2, 'O poder do hábito', 'psicologia'),
    (4, 'intrinsace', 13, 'Uma Breve História do Tempo', 'física'),
    (5, 'Veneta', null, 'O capital','sociologia' ),
    (6, 'Pearson', 1, 'O senhor dos aneis', 'Fantasia'),
    (7, 'Rocco', 1, 'Harry Potter e a preda filosonal', 'Fantasia'),
    (8, 'Veneta', 5 , 'Jogos Vorazes', 'Ficcao'),
    (9, 'Rocco', 4, 'O pequeno Principe', 'Fantasia'),
    (10, 'Presença', 1,'Cartas de amor aos mortos', 'Tecnologia');

    /* Inserindo informações na tabela Exemplar*/
    insert into Exemplar(cod_exemplar, quantidade, ISBN_livro)
    values(02946, 20, 1), /*Psicologia das cores*/
    (01298, 10, 1),/*Sistemas Operativos Modernos*/
    (68460, 40, 2),/*O poder do hábito*/
    (47420, 21, 3), /*Uma breve história do tempo*/
    (07213, 13, 4);/*O capital*/

```

```

/* Inserindo informações na tabela Escrito_por*/
insert into Escrito_por(cod_autor, ISBN_livro)
values(1, 1), /*Psicologia das cores*/
(2, 2), /*Sistemas Operativos Modernos*/
(3, 3), /*O poder do hábito*/
(4, 4), /*Uma breve história do tempo*/
(5, 5), /*O capital*/
(5, 6),
(5, 7),
(2, 8),
(3, 9),
(1, 10);

/* O comando abaixo exemplifica o procedimento para a inserção de imagens
rodar esse comando: select @@GLOBAL.secure_file_priv;
substituir 'nome_do_arquivo' pelo resultado do comando anterior: select
load_file('/var/lib/mysql-files/{nome_do_arquivo}');
*/
/* Inserindo informações na tabela Usuario */
insert into Usuario(matricula, nome, data_de_nascimento, rua, UF, CEP, cidade, bairro,
numero, foto, data_de_validade, qr_code)
values(2022001, 'Vanessa Silva', '2000-10-31', 'Avenida Coremas','PB', '58013-430', 'João
Pessoa','Centro','561' , load_file('/var/lib/mysql-files/imagem.png'), '2024-12-31',
load_file('/var/lib/mysql-files/imagem.png')),
(2022002, 'Rita Clara', '2001-11-06', 'Rua Comercário Antônio Manoel de Sousa','PB',
'58071-585', 'João Pessoa','Cristo Redentor','20' ,
load_file('/var/lib/mysql-files/imagem.png'), '2022-12-31',
load_file('/var/lib/mysql-files/imagem.png')),
(2022003, 'Lais Epifanio Machado', '2002-10-10', 'Rua Osvaldo Travassos Campos','PB',
'58080-540', 'João Pessoa','Ernani Sátiro','16' ,
load_file('/var/lib/mysql-files/imagem.png'), '2023-12-31',
load_file('/var/lib/mysql-files/imagem.png')),
(2022004, 'Cleiton Bernadino', '2001-11-08', 'Conjunto Jacinto Medeiros','PB', '58026-080',
'João Pessoa','Treze de Maio','120' , load_file('/var/lib/mysql-files/imagem.png'),
'2022-12-31', load_file('/var/lib/mysql-files/imagem.png')),
(2022005, 'Raimundo de Moraes', '2004-03-23', 'Rua Guadalupe','PB', '58079-806', 'João
Pessoa','Grotão', '777', load_file('/var/lib/mysql-files/imagem.png'), '2024-12-31',
load_file('/var/lib/mysql-files/imagem.png'));

/*Inserindo informações na tabela email*/
insert into Email(email, matricula_usuario)
values('Silva.Vanessa@academico.ifpb.edu.br', 2022001),
('Silva.Rita@academico.ifpb.edu.br', 2022002),
('Lais.Machado@academico.ifpb.edu.br', 2022003),
('Cleiton.Bernadito@academico.ifpb.edu.br', 2022004),
('Mores.Santos@academico.ifpb.edu.br', 2022005);

/* Inserindo informações na tabela telefone*/
insert into Telefone(telefone, matricula_usuario)
values('98854-1234', 2022001),
('94002-8922', 2022002),
('92034-9933', 2022003),
('91234-5678', 2022004),

```

```

('98765-4321', 2022005);

/* Insert de funcionario */
insert into Funcionario(cpf, nome, telefone, data_de_nascimento, rua, uf, cep, cidade,
bairro,numero, login_atendente, senha_atendente, tipo, cpf_responsavel_cadastro)
value('10274829351','Caetano Veloso', '98867-1235', '1975-10-05', 'Avenida Coremas','PB',
'58013-430', 'João Pessoa','Centro','565' , null, null, null, null),
('15624829361','Alcione', '99357-1251', '1955-11-13', 'Rua Guadalupe','PB', '58079-806',
'João Pessoa','Grotão', '777', null, null, null, '10274829351'),
('12924862965','Renata Costa', '98887-1235', '1975-03-05', 'Rua Comerciaro Antônio Manoel
de Sousa','PB', '58071-585', 'João Pessoa','Cristo Redentor','18', null, null, null,
'10274829351'),
('10252049367','Catarina Rios', '98868-3215', '1985-10-10', 'Conjunto Jacinto
Medeiros','PB', '58026-080', 'João Pessoa','Treze de Maio','122', null, null, null,
'10274829351'),
('41274829954','Felipe Neto', '98867-1235', '1984-01-11', 'Rua Osvaldo camara Travassos
Campos','PB', '58080-540', 'João Pessoa','Ernani Sátiro','19', null, null, null,
'10274829351'),
('41274829953','Eliana', '98867-1235', '1984-06-11', 'Rua da areia Travassos Campos','PB',
'58080-540', 'João Pessoa','Ernani Sátiro','193', null, null, null, '10274829351'),
('41274829951','Edemberg', '98867-1235', '1994-05-11', 'Rua Osvaldo Travassos Campos','PE',
'58080-510', 'Recife','Boa Viagem','21', 'edemberg', '123456', 'atendente', '10274829351');

/*Inserindo informações na tabela empréstimo*/
insert into Emprestimo(matricula_usuario, cpf_funcionario, cod_exemplar,data_de_emprestimo,
data_de_devolucao)
value(2022001, null,02946, '2022-11-20', '2022-12-21'), /* Vanessa fez um empréstimo livro
"Psicologia das cores"*/
(2022002, '15624829361' ,02946, '2022-11-30', '2022-12-30'), /* Rita fez um empréstimo
livro "Psicologia das cores"*/
(2022001, '15624829361' ,47420, '2022-11-06', '2022-12-25'), /* Laís fez um empréstimo
livro "Uma breve história do tempo"*/
(2022003, '10274829351' ,07213, '2022-11-20', '2022-12-26'), /* Cleiton fez um empréstimo
livro "O capital"*/
(2022004, '10274829351' ,01298, '2022-11-28', '2022-12-28'); /* Raimundo fez um empréstimo
livro "Sistemas Operativos Modernos"*/

/* selecionando todos os empréstimos que nao tiveram um funcionário envolvido */
select *
from Emprestimo
where cpf_funcionario is null;

/* selecionando todos os exemplares cujo a quantidade está entre 11 e 30 */
select *
from Exemplar
where quantidade between 11 and 30;

/* selecionando a quantidade de funcionarios por cep, agrupando-os */
select cep, count(*) as quantidade
from Funcionario
group by cep;

/* selecionando o Funcionario mais velho dentre os que possuem cep iniciando com 58080 */

```

```

select min(data_de_nascimento)
from Funcionario
where cep like '58080%';

/* selecionando todos os autores e ordenando do mais novo pro mais velho pela data de
nascimento */
select *
from Autor
order by data_de_nascimento desc;

/* selecionando os livros que possuem edição diferente de 1 e 4 */
select *
from Livro
where numero_da_edicao not in (1,4);

/* Selecionando os autores que possuem mais que um registro de livro
escrito*/
select A.nome as autor
from Autor A
join Escrito_por E
on A.cod_autor = E.cod_autor
group by A.nome
Having count(A.nome) > 1;

```