



REDE DE LIVRARIAS

SQL

RECIFE

2022

ARTHUR THIERRE DOS SANTOS SILVA <ATSS>

JEFERSON SEVERINO DE ARAÚJO <JSA2>

LUCAS NASCIMENTO BRANDÃO <LNB>

MARCUS VINÍCIUS DE FARIA SANTOS <MVFS>

VICTOR GABRIEL DE CARVALHO <VGC3>

REDE DE LIVRARIAS

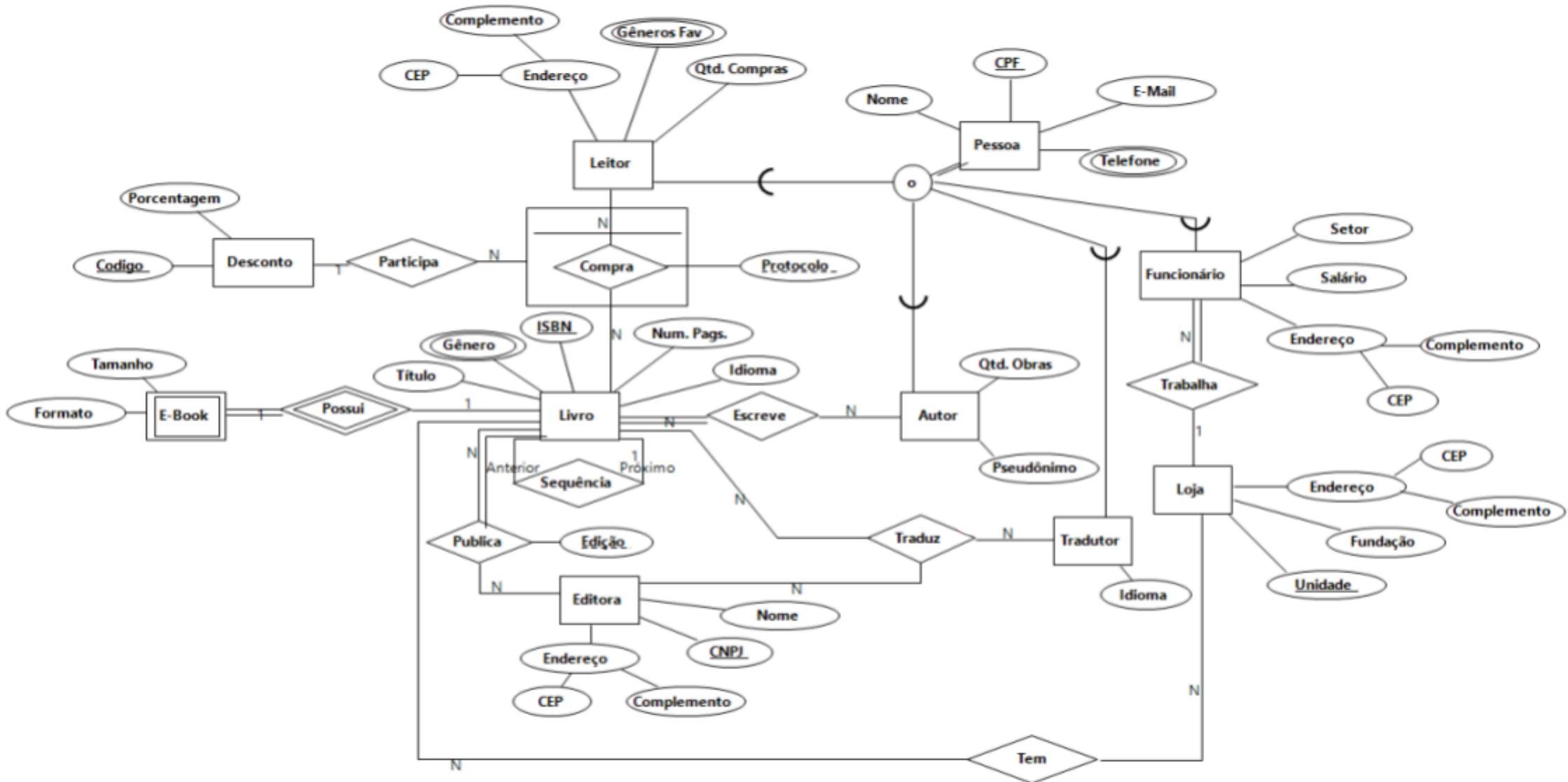
SQL

Trabalho apresentado ao Curso de
Gerenciamento de Dados e Informação da
Universidade Federal de Pernambuco.

Orientador: Professor Robson Fidalgo

RECIFE

06 DE SETEMBRO DE 2022



MAPEAMENTO RELACIONAL

Livro (ISBN, titulo, idioma, num_pags, [ISBN_ant])

ISBN_ant -> **Livro** (ISBN)

Genero (ISBN, genero)

ISBN -> **Livro** (ISBN)

E-Book (ISBN, tamanho, formato)

ISBN -> **Livro** (ISBN)

Editora (CNPJ, nome, end_CEP, end_complemento)

Loja (unidade, fundação, end_CEP, end_complemento)

Compra (ISBN, CPF, protocolo, codigo)

ISBN -> **Livro** (ISBN)

CPF -> **Leitor** (CPF)

codigo -> **Desconto** (codigo)

Desconto (codigo, porcentagem)

Leitor (CPF, nome, e-mail, tel1, tel2, qtd_compras, end_CEP, end_complemento)

Gens_fav (CPF, genero)

CPF -> **Leitor**(CPF)

Autor (CPF, nome, e-mail, tel1, tel2, pseudonimo, qtd_obras)

Funcionario (CPF, nome, e-mail, tel1, tel2, setor, salario, end_CEP, end_complemento, unidade!)

unidade -> **Loja** (unidade)

Tradutor (CPF, nome, e-mail, tel1, tel2, idioma)

Publica (ISBN, CNPJ, Edicao)

ISBN -> **Livro** (ISBN)

CNPJ -> **Editora** (CNPJ)

Escreve (ISBN, CPF)

ISBN -> **Livro** (ISBN)

CPF -> **Autor** (CPF)

Tem (ISBN, unidade)

ISBN -> **Livro** (ISBN)

unidade -> **Loja** (unidade)

Traduz (ISBN, CNPJ, CPF)

ISBN -> **Livro** (ISBN)

CNPJ -> **Editora** (CNPJ)

CPF -> **Tradutor** (CPF)

SQL Oracle

```
CREATE TABLE Livro(  
    livro_isbn VARCHAR(15),  
    livro_titulo VARCHAR(50),  
    livro_idioma VARCHAR(15),  
    livro_num_pags NUMBER(4),  
    livro_isbn_ant VARCHAR(15),  
    CONSTRAINT PK_livro PRIMARY KEY (livro_isbn),  
    CONSTRAINT FK_livro_ant FOREIGN KEY (livro_isbn_ant) REFERENCES Livro(livro_isbn)  
);
```

```
CREATE TABLE Genero(  
    genero_isbn VARCHAR(15),  
    genero VARCHAR(30),  
    CONSTRAINT PK_genero PRIMARY KEY (genero_isbn, genero),  
    CONSTRAINT FK_genero FOREIGN KEY (genero_isbn) REFERENCES Livro(livro_isbn)  
);
```

```
CREATE TABLE Ebook(  
    ebook_isbn VARCHAR(15),  
    ebook_tamanho NUMBER(4),  
    ebook_formato VARCHAR(10),  
    CONSTRAINT PK_ebook PRIMARY KEY (ebook_isbn),  
    CONSTRAINT FK_ebook FOREIGN KEY (ebook_isbn) REFERENCES Livro(livro_isbn)  
);
```

```
CREATE TABLE Editora(  
    editora_cnpj VARCHAR(14),  
    editora_nome VARCHAR(80),  
    editora_end_cep VARCHAR(20),  
    editora_end_compl VARCHAR(40) ,  
    CONSTRAINT PK_editora PRIMARY KEY (editora_cnpj)  
);
```

```
CREATE TABLE Loja(  
    loja_unidade VARCHAR(10),  
    loja_fundacao NUMBER(4),  
    loja_end_cep VARCHAR(20),  
    loja_end_compl VARCHAR(40),  
    CONSTRAINT PK_loja PRIMARY KEY (loja_unidade)  
);
```

```
CREATE TABLE Leitor(  
    leitor_cpf VARCHAR(15),  
    leitor_nome VARCHAR(20),
```

```

    leitor_email VARCHAR(20),
    leitor_tel1 VARCHAR(20) ,
    leitor_tel2 VARCHAR(20),
    leitor_qtd_compras NUMBER(4),
    leitor_end_cep VARCHAR(20) ,
    leitor_end_compl VARCHAR(20),
    CONSTRAINT PK_leitor PRIMARY KEY (leitor_cpf)
);

```

```

CREATE TABLE Desconto(
    desconto_codigo VARCHAR(30),
    desconto_porcentagem NUMBER(3),
    CONSTRAINT PK_desconto PRIMARY KEY (desconto_codigo)
);

```

```

CREATE TABLE Compra(
    compra_isbn VARCHAR(15),
    compra_cpf VARCHAR(20),
    compra_codigo VARCHAR(30),
    compra_protocolo VARCHAR(30),
    CONSTRAINT PK_compra PRIMARY KEY (compra_isbn, compra_cpf, compra_protocolo),
    CONSTRAINT FK_compra_livro FOREIGN KEY (compra_isbn) REFERENCES
Livro(livro_isbn),
    CONSTRAINT FK_compra_leitor FOREIGN KEY (compra_cpf) REFERENCES
Leitor(leitor_cpf),
    CONSTRAINT FK_compra_desconto FOREIGN KEY (compra_codigo) REFERENCES
Desconto(desconto_codigo)
);

```

```

CREATE TABLE Gens_fav(
    gf_genero VARCHAR(30),
    gf_cpf VARCHAR(15),
    CONSTRAINT PK_gens_fav PRIMARY KEY (gf_cpf, gf_genero),
    CONSTRAINT FK_gens_fav FOREIGN KEY (gf_cpf) REFERENCES Leitor(leitor_cpf)
);

```

```

CREATE TABLE Autor(
    autor_cpf VARCHAR(15),
    autor_nome VARCHAR(40),
    autor_tel1 VARCHAR(20),
    autor_tel2 VARCHAR(20),
    autor_pseudonimo VARCHAR(20),
    autor_qtd_obras NUMBER(3),
    CONSTRAINT PK_autor PRIMARY KEY (autor_cpf)
);

```

```
CREATE TABLE Funcionario(  
    func_cpf VARCHAR(15),  
    func_nome VARCHAR(20),  
    func_tel1 VARCHAR(20),  
    func_tel2 VARCHAR(20),  
    func_setor VARCHAR(20),  
    func_salario NUMBER(4),  
    func_end_cep VARCHAR(20),  
    func_end_compl VARCHAR(20),  
    CONSTRAINT PK_funcionario PRIMARY KEY (func_cpf)  
);
```

```
CREATE TABLE Trabalha(  
    trabalha_cpf VARCHAR(20),  
    trabalha_unidade VARCHAR(10),  
    data DATE,  
    CONSTRAINT PK_Trabalha PRIMARY KEY (trabalha_cpf, trabalha_unidade, data),  
    CONSTRAINT FK_trabalha_func FOREIGN KEY (trabalha_cpf) REFERENCES  
Funcionario(func_cpf),  
    CONSTRAINT FK_trabalha_loja FOREIGN KEY (trabalha_unidade) REFERENCES  
Loja(loja_unidade)  
);
```

```
CREATE TABLE Tradutor(  
    trad_cpf VARCHAR(15),  
    trad_nome VARCHAR(20),  
    trad_email VARCHAR(20),  
    trad_tel1 VARCHAR(20),  
    trad_tel2 VARCHAR(20),  
    trad_idioma VARCHAR(20),  
    CONSTRAINT PK_tradutor PRIMARY KEY (trad_cpf)  
);
```

```
CREATE TABLE Publica(  
    pub_isbn VARCHAR(15),  
    pub_cnpj VARCHAR(20),  
    pub_edicao NUMBER(2),  
    CONSTRAINT PK_publica PRIMARY KEY (pub_isbn, pub_cnpj, pub_edicao),  
    CONSTRAINT FK_publica_livro FOREIGN KEY (pub_isbn) REFERENCES Livro(livro_isbn),  
    CONSTRAINT FK_publica_editora FOREIGN KEY (pub_cnpj) REFERENCES  
Editora(editora_cnpj)  
);
```

```
CREATE TABLE Escreve(  
    escreve_isbn VARCHAR(15),  
    escreve_cpf VARCHAR(20),
```



```
CONSTRAINT PK_escreve PRIMARY KEY (escreve_isbn, escreve_cpf),
CONSTRAINT FK_escreve_livro FOREIGN KEY (escreve_isbn) REFERENCES
Livro(livro_isbn),
CONSTRAINT FK_escreve_autor FOREIGN KEY (escreve_cpf) REFERENCES
Autor(autor_cpf)
);
```

```
CREATE TABLE Tem(
    tem_isbn VARCHAR(15),
    tem_unidade VARCHAR(20),
    CONSTRAINT PK_tem PRIMARY KEY (tem_isbn, tem_unidade),
    CONSTRAINT FK_tem_livro FOREIGN KEY (tem_isbn) REFERENCES Livro(livro_isbn),
    CONSTRAINT FK_tem_loja FOREIGN KEY (tem_unidade) REFERENCES Loja(loja_unidade)
);
```

```
CREATE TABLE Traduz(
    traduz_isbn VARCHAR(15),
    traduz_cnpj VARCHAR(20),
    traduz_cpf VARCHAR(20),
    CONSTRAINT PK_traduz PRIMARY KEY (traduz_isbn, traduz_cnpj, traduz_cpf),
    CONSTRAINT FK_traduz_livro FOREIGN KEY (traduz_isbn) REFERENCES Livro(livro_isbn),
    CONSTRAINT FK_traduz_editora FOREIGN KEY (traduz_cnpj) REFERENCES
Editora(editora_cnpj),
    CONSTRAINT FK_taduz_tradutor FOREIGN KEY (traduz_cpf) REFERENCES
Tradutor(trad_cpf)
);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags)
VALUES ('0', 'Harry Potter e a Pedra Filosofal', 'Inglês', 250);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('1', 'Harry Potter e a Câmara Secreta', 'Inglês', 250, '0');
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('2', 'Harry Potter e o Prisioneiro de Azkaban', 'Inglês', 300, '1');
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('3', 'Harry Potter e o Calice de Fogo', 'Inglês', 500, '2');
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('4', 'Harry Potter e a Ordem da Fênix', 'Inglês', 650, '3');
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('5', 'Harry Potter e o Enigma do Príncipe', 'Inglês', 450, '4');
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('6', 'Harry Potter e as Relíquias da Morte', 'Inglês', 600, '5');
```

```
INSERT INTO livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('7', 'The Screwtape Letters', 'Inglês', '172', NULL);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('7593', 'The Worlds Last Night and Others Essays', 'Inglês', '130', NULL);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('72', 'Till We Have Faces', 'Inglês', '311', NULL);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('657', 'On Stories: and other essays on literature ', 'Inglês', '252', NULL);
```

```
INSERT INTO Livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('7579', 'God in the Dock: Essays in Theology and Ethics ', 'Inglês', '413', NULL);
```

```
INSERT INTO livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('947', 'De um pai para seus filhos ', 'Português ', '135', NULL);
```

```
INSERT INTO livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('646', 'Crime e castigo', 'Russo', '590', NULL);
```

```
INSERT INTO livro (livro_isbn, livro_titulo, livro_idioma, livro_num_pags, livro_isbn_ant)
VALUES ('267', 'Clean Code', 'Inglês', '410', NULL);
```

```
INSERT INTO editora (editora_cnpj, editora_nome, editora_end_cep, editora_end_compl)
VALUES ('0', 'Rocco', NULL, NULL);
```

```
INSERT INTO editora (editora_cnpj, editora_nome, editora_end_cep, editora_end_compl)
VALUES ('1790', 'Harper Collins', NULL, NULL);
```

```
INSERT INTO editora (editora_cnpj, editora_nome, editora_end_cep, editora_end_compl)
VALUES ('1798', 'Thomas Nelson', NULL, NULL);
```

```
INSERT INTO editora (editora_cnpj, editora_nome, editora_end_cep, editora_end_compl)
VALUES ('34', 'Editora 34', NULL, NULL);
```

```
INSERT INTO editora (editora_cnpj, editora_nome, editora_end_cep, editora_end_compl)
VALUES ('2011', 'Alta Books', NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('123', 'J. K. Rowling', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('777', 'C. S. Lewis', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('999', 'Asaph Borba', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('888', 'Fiódor Dostoiévski', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('232', 'Robert C. Martin', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('242', 'Michael Ottinger', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('252', 'Jeffrey Schuchert', NULL, NULL, NULL, NULL);
```

```
INSERT INTO autor (autor_cpf, autor_nome, autor_tel1, autor_tel2, autor_pseudonimo,
autor_qtd_obras)
VALUES ('262', 'James Wampler', NULL, NULL, NULL, NULL);
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('0', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('1', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('2', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('3', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('4', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('5', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('6', '123');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('7', '777');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('72', '777');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('657', '777');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('7579', '777');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('7593', '777');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('947', '999');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('646', '888');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('267', '232');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('267', '242');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('267', '252');
```

```
INSERT INTO escreve (escreve_isbn, escreve_cpf)
VALUES ('267', '262');
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)
VALUES ('0', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)
VALUES ('0', '34', '1');
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)
VALUES ('1', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)
```

```
VALUES ('2', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('3', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('4', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('5', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('6', '0', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('7', '1790', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('657', '1790', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('72', '1790', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('7579', '1790', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('7593', '1790', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('947', '1798', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('646', '34', 1);
```

```
INSERT INTO publica (pub_isbn, pub_cnpj, pub_edicao)  
VALUES ('267', '2011', '1');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)  
VALUES ('7676', 'Giuliana Niedhardt', NULL, NULL, NULL, 'Português');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)  
VALUES ('7777', 'Francisco Nunes', NULL, NULL, NULL, 'Português');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)  
VALUES ('7878', 'Jorge Camargo', NULL, NULL, NULL, 'Português');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)
VALUES('7979', 'Ana Paula Spolon', NULL, NULL, NULL, 'Português');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)
VALUES ('3434', 'Paulo Bezerra', NULL, NULL, NULL, 'Português');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)
VALUES ('0909', 'Leandro Chu', NULL, NULL, NULL, 'Português ');
```

```
INSERT INTO tradutor (trad_cpf, trad_nome, trad_email, trad_tel1, trad_tel2, trad_idioma)
VALUES ('0880', 'Michele Aguiar', NULL, NULL, NULL, 'Português ');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('7593', '1798', '7777');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('657', '1798', '7777');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('7579', '1798', '7676');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('72', '1798', '7878');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('72', '1798', '7979');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('267', '2011', '0909');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('267', '2011', '0880');
```

```
INSERT INTO traduz (traduz_isbn, traduz_cnpj, traduz_cpf)
VALUES ('646', '34', '3434');
```

```
INSERT INTO leitor (leitor_cpf, leitor_nome, leitor_email, leitor_tel1, leitor_tel2,
leitor_qtd_compras, leitor_end_cep, leitor_end_compl)
VALUES ('1', 'Jeferson', 'jeferson@email', '12345-1234', '12354-1234', 2, '55825', 'liberdade');
```

```
INSERT INTO leitor (leitor_cpf, leitor_nome, leitor_email, leitor_tel1, leitor_tel2,
leitor_qtd_compras, leitor_end_cep, leitor_end_compl)
VALUES ('2', 'Victor', 'victor@email', '12345-1231', '12354-876', 2, '55825', 'rua da oficina');
```

```
INSERT INTO leitor (leitor_cpf, leitor_nome, leitor_email, leitor_tel1, leitor_tel2,
leitor_qtd_compras, leitor_end_cep, leitor_end_compl)
VALUES ('3', 'Marcus', 'marcus@email', '12345-1211', '12354-765', 2, '55825', 'rua da
faculdade');
```

```
INSERT INTO leitor (leitor_cpf, leitor_nome, leitor_email, leitor_tel1, leitor_tel2,
leitor_qtd_compras, leitor_end_cep, leitor_end_compl)
VALUES ('4', 'Arthur', 'arthur@email', '12345-4543', '12354-5675', 2, '55825', 'rua do
consultorio');
```

```
INSERT INTO leitor (leitor_cpf, leitor_nome, leitor_email, leitor_tel1, leitor_tel2,
leitor_qtd_compras, leitor_end_cep, leitor_end_compl)
VALUES ('5', 'Lucas', 'lucas@email', '12345-111', '12354-5675', 2, '55825', 'rua do shopping');
```

```
INSERT INTO Loja(loja_unidade, loja_fundacao, loja_end_cep, loja_end_compl)
VALUES ('001', 2014, '5555-55', NULL);
```

```
INSERT INTO Loja(loja_unidade, loja_fundacao, loja_end_cep, loja_end_compl)
VALUES ('002', 2017, '6666-66', NULL);
```

```
INSERT INTO Loja(loja_unidade, loja_fundacao, loja_end_cep, loja_end_compl)
VALUES ('003', 2022, '7777-77', NULL);
```

```
INSERT INTO desconto (desconto_codigo, desconto_porcentagem)
VALUES ('1', 10);
```

```
INSERT INTO desconto (desconto_codigo, desconto_porcentagem)
VALUES ('2', 10);
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('6', '1', null, '1');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('0', '1', '1', '2');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('6', '2', null, '3');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('4', '2', '1', '4');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('3', '3', null, '5');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('0', '3', null, '6');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('6', '4', null, '7');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('0', '4', null, '8');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('0', '5', '2', '9');
```

```
INSERT INTO compra (compra_isbn, compra_cpf, compra_codigo, compra_protocolo)
VALUES ('4', '5', null, '10');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('0', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('1', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('2', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('3', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('4', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('5', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('6', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('7', '001');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('72', '002');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('947', '002');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('0', '002');
```



```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('1', '002');
```

```
INSERT INTO Tem(tem_isbn, tem_unidade)
VALUES ('2', '002');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('646', '9', 'EPUB');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('947', '5', 'EPUB');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('7579', '7', 'PDF');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('72', '4', 'PDF');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('4', '21', 'PDF');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('7', '12', 'EPUB');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('2', '11', 'EPUB');
```

```
INSERT INTO ebook (ebook_isbn, ebook_tamanho, ebook_formato)
VALUES ('3', '7', 'EPUB');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,
func_salario, func_end_cep, func_end_compl)
VALUES ('11', 'antonio', '8763-1321', '6544-7565', 'vendas', 2500, '55501-091', 'rua do hospital');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,
func_salario, func_end_cep, func_end_compl)
VALUES ('12', 'otavio', '1232-1231', '5634-5656', 'gerente', 5000, '55501-091', 'rua do
restaurante');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,
func_salario, func_end_cep, func_end_compl)
VALUES ('13', 'maria', '6354-7353', '5436-3453', 'vendas', 2500, '55501-091', 'rua do senador');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,
func_salario, func_end_cep, func_end_compl)
```

```
VALUES ('14', 'ana', '3456-3456', '3453-4356', 'gerente', 5000, '55501-091', 'rua da republica');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,  
func_salario, func_end_cep, func_end_compl)  
VALUES ('15', 'pedro', '5433-5424', '3443-7745', 'vendas', 2500, '55501-091', 'rua do sargento');
```

```
INSERT INTO funcionario (func_cpf, func_nome, func_tel1, func_tel2, func_setor,  
func_salario, func_end_cep, func_end_compl)  
VALUES ('16', 'silva', '2345-1234', '1235-1211', 'gerente', 5000, '55501-091', 'rua dos  
computadores');
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('11', '001', TO_DATE('20/12/2019', 'DD/MM/YYYY'));
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('12', '001', TO_DATE('20/12/2020', 'DD/MM/YYYY'));
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('13', '002', TO_DATE('13/09/2019', 'DD/MM/YYYY'));
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('14', '002', TO_DATE('20/07/2021', 'DD/MM/YYYY'));
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('15', '003', TO_DATE('22/08/2022', 'DD/MM/YYYY'));
```

```
INSERT INTO trabalha (trabalha_cpf, trabalha_unidade, data)  
VALUES ('16', '003', TO_DATE('24/07/2022', 'DD/MM/YYYY'));
```

```
INSERT INTO genero(genero_isbn, genero)  
VALUES ('1', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)  
VALUES ('2', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)  
VALUES ('3', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)  
VALUES ('4', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)  
VALUES ('5', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('6', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('7', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('0', 'fantasia');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('267', 'horror');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('947', 'drama');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('7579', 'romance');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('7593', 'thriller');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('72', 'suspense');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('657', 'drama');
```

```
INSERT INTO genero(genero_isbn, genero)
VALUES ('646', 'drama');
```


Consultas SQL

Fator de popularidade das lojas em relação ao seu estoque de livros.

```
SELECT T.tem_unidade as Unidade, COUNT(*) as Vendas
FROM tem T INNER JOIN Compra C ON C.compra_isbn = T.tem_isbn
GROUP BY T.tem_unidade
ORDER BY Vendas DESC;
```

Nomes das editoras que publicaram mais de um livro

```
SELECT editora_nome, COUNT(*)
FROM Editora E INNER JOIN Publica P ON E.editora_cnpj = P.pub_cnpj
GROUP BY editora_nome
HAVING COUNT(*) > 1
```

Livros vendidos ordenados por vendas

```
SELECT livro_titulo, COUNT(livro_titulo) AS "qtd_vendidos"
FROM Compra, Livro
WHERE (compra_ISBN = livro_isbn)
GROUP BY livro_titulo
ORDER BY COUNT (livro_titulo)
```

Nome e cpf do leitor que comprou livro que está em estoque na loja com unidade 002.

```
SELECT L.leitor_nome, L.leitor_cpf
FROM Leitor L
WHERE EXISTS (SELECT *
              FROM Compra C, Tem T
              WHERE T.tem_isbn = C.compra_isbn and C.compra_cpf = L.leitor_cpf AND T.tem_unidade = '002');
```

Selecionar livros que não possuem tradução para o português.

```
SELECT *
FROM Livro L
WHERE NOT EXISTS (SELECT *
                  FROM Traduz T
                  WHERE T.traduz_isbn = L.livro_isbn and T.traduz_cpf in (SELECT T1.trad_cpf
                                                                           FROM tradutor T1
                                                                           WHERE T1.trad_idioma = 'Português'));
```

Retornar todos os leitores que não compraram nenhum livro.

```
SELECT leitor_nome
FROM Leitor L LEFT OUTER JOIN Compra C ON (L.leitor_cpf = C.compra_cpf)
WHERE C.compra_cpf IS NULL
```

Selecionar os nomes dos autores que escreveram livros em inglês.

```
SELECT A.autor_nome
FROM autor A WHERE A.autor_cpf IN (SELECT E.escreve_cpf
                                   FROM escreve E
                                   WHERE E.escreve_isbn IN (SELECT L.livro_isbn
                                                            FROM livro L
                                                            WHERE L.livro_idioma = 'Inglês' ));
```

Retorna as unidades que tem mais livros em estoque do que a unidade 002

```
SELECT T.tem_unidade
FROM tem T
GROUP BY T.tem_unidade
HAVING count(*) > (SELECT COUNT(*)
                  FROM tem T
                  WHERE T.tem_unidade = '002'
                  GROUP BY T.tem_unidade );
```

Mostra Título e Quantidade de livros vendidos que tem uma quantidade de venda menor do que três

Título dos livros publicados na mesma editora que o livro 'till we have faces'

Mostra os livros em inglês que foram publicados em mais de uma editora

[illegible]

Procedimentos PL

```
-- Nomes das editoras em ordem decrescente de venda de um livro passado
por parametro.

create or replace procedure ordem_vendas_editora_livro (isbn varchar)
is

    cursor ordem_editoras is
        select editora_nome
        from Editora
        where editora_cnpj in (
            select pub_cnpj
            from Publica
            where pub_isbn in (
                select livro_isbn
                from Livro L
                where exists (
                    select compra_isbn
                    from Compra C
                    where C.compra_isbn = L.livro_isbn
                ) and livro_isbn = isbn
            )
        )
        group by editora_nome
        order by count(*) desc;

    nomeEditora Editora.editora_nome % type

begin
    open ordem_editoras ;

    dbms_output.put_line('As editoras que mais venderam esse livro por
ordem decrescente são: ');

    loop

        fetch ordem_editoras into nomeEditora;
        exit when ordem_editoras % NOTFOUND;

        dbms_output.put_line(nomeEditora);

    end loop;
```



```
        close ordem_editoras;  
end;
```

```
-- Procedimento que vai retornar todos os leitores que leram livros no  
idioma passado por parametro  
  
create or replace procedure leitores_idioma (idioma varchar) is  
  
    cursor nomes_leitores is  
        select leitor_nome  
        from Leitor  
        where leitor_cpf = (  
            select compra_cpf  
            from Compra  
            where compra_isbn = (  
                select livro_isbn  
                from Livro  
                where livro_idioma = idioma  
            )  
        )  
    );  
  
    nomeLeitor Leitor.leitor_nome % type;  
  
begin  
    open nomes_leitores;  
  
    loop  
        fetch nomes_leitores into nomeLeitor;  
        exit when nomes_leitores % NOTFOUND;  
  
        dbms_output.put_line(nomes_leitores || ' Leu pelo menos um livro  
nesse idioma');  
  
    end loop;  
  
    close nomes_leitores;  
  
end;
```

Triggers

Impedir a inserção de salários negativos

```
create or replace trigger salario_negativo
before insert or update of salario on Funcionario
referencing old as v new as n
for each row
begin
    if salario < 0 then
        raise_application_error(-20205, "Não pode cadastrar salário
negativo");
    else
        if updating then
            dbms_output.put_line('Salário atualizado com sucesso');
            dbms_output.put_line('Salário antigo: ' || :v.salario);
            dbms_output.put_line('Novo salário: ' || :n.salario);
        else
            dbms_output.put_line('Novo salário cadastrado com sucesso');
        end if;
    end if;
end;
```

Desconto Válido

```
create or replace trigger desconto_valido
before insert or update of porcentagem on Desconto
referencing old as v new as n
for each row
begin
    if porcentagem > 70 then
        raise_application_error(-20000, 'Cupom de desconto com valor
muito alto');

    elsif porcentagem < 0 then
        raise_application_error(-20001, 'Porcentagem de desconto não
pode ser negativa');
    else
        if updating then
            dbms_output.put_line('Novo desconto atualizado com sucesso');
            dbms_output.put_line('Desconto antigo: ' || :v.porcentagem);
            dbms_output.put_line('Novo desconto: ' || :n.porcentagem);
        else
            dbms_output.put_line('Novo cupom de desconto cadastrado com
desconto válido');
        end if;
    end if;
end;
```

Trigger para cadastro incorreto do livro

```
create or replace trigger cadastro_invalido_livro
before insert or update of livro_num_pags, livro_idioma on Livro
```

```

referencing old as v new as n

for each row

begin

    if livro_num_pags < 0 then
        raise_application_error(-20000, 'Negado. Não existe livro com
uma quantidade negativa de páginas');

    elsif livro_idioma != 'Portugues' and livro_idioma != 'Ingles' and
livro_idioma != 'Espanhol' then
        raise_application_error(-20001, 'Registrando um idioma
diferente dos disponíveis');

    else
        if updating then
            dbms_output.put_line('Novo livro atualizado com sucesso');
        else
            dbms_output.put_line('Novo livro atualizado com sucesso');
        end if;
    end if;

end;

```

Trigger para tradutor em um idioma não disponível

```

create or replace trigger cadastro_invalido_tradutor
before insert or update of trad_idioma on Tradutor
referencing old as v new as n

```

```
for each row
begin
    if trad_idioma != 'Portugues' and trad_idioma != 'Ingles' and
trad_idioma != 'Espanhol' then
        raise_application_error(-20003, 'O idioma inserido não faz
parte dos idiomas disponíveis');
    else
        if updating then
            dbms_output.put_line('O idioma do tradutor foi alterado com
sucesso');
            dbms_output.put_line('Idioma antigo: ' || :v.trad_idioma);
            dbms_output.put_line('Novo idioma: ' || :n.trad_idioma);
        else
            dbms_output.put_line('Novo tradutor cadastrado com
sucesso');
        end if;
    end if;
end;
```