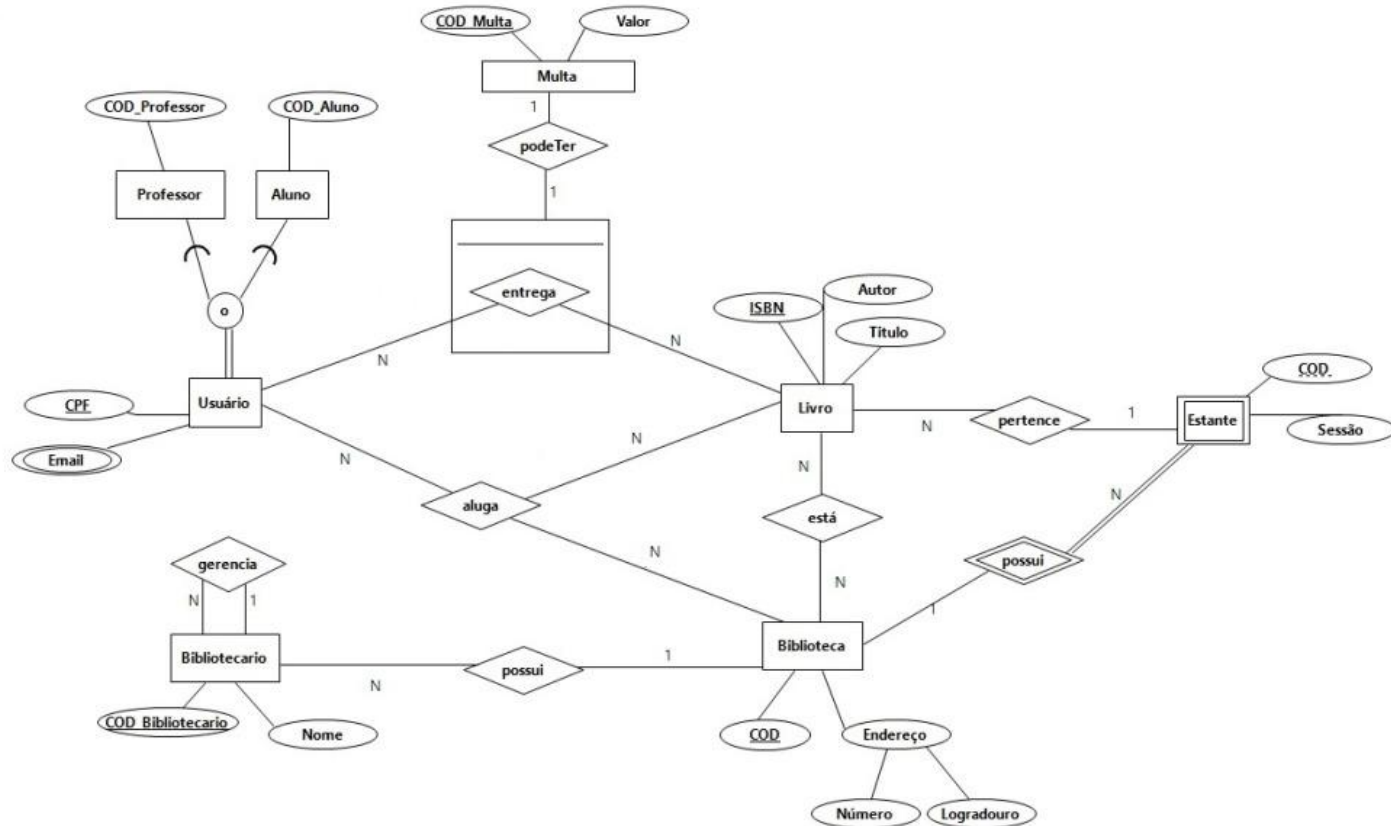


# Biblioteca / Livraria

Projeto de Banco de Dados

# Modelo Conceitual



# Modelo Conceitual

## Entidades

- Usuário
- Professor -> Herança\*
- Aluno -> Herança\*
- Multa -> Derivado de um relacionamento associativo
- Livro
- Biblioteca
- Estante -> Entidade Fraca\*
- Bibliotecário

## Relacionamentos

- Entrega (Usuario - Livro)
- Pode ter\* (Entrega (Usuario, Livro) - Multa)
- Aluga (Usuario - Livro - Biblioteca)
- Pertence (Livro - Estante)
- Esta (Livro - Biblioteca)
- Possui (Biblioteca - Estante)
- Possui (Biblioteca - Bibliotecário)
- Gerencia (Bibliotecario - Bibliotecario)

# Modelo Lógico

**Multa**( COD\_Multa, Valor)

**Bibliotecário**( COD\_Bibliotecário, nome )

**Biblioteca** ( COD, end\_logradouro, end\_numero)

**Estante** ( COD\_Biblioteca, COD, Sessão)

COD\_Biblioteca =====> **Biblioteca** (COD)

**Usuário** ( CPF )

**Professor** (CPF, COD\_Professor)

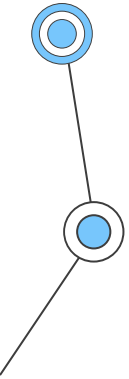
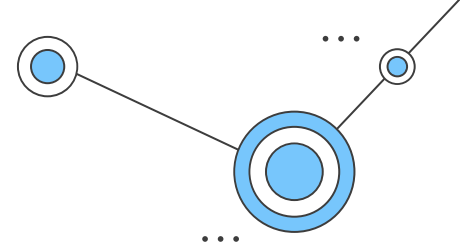
CPF =====> **Usuário**(CPF)

**Aluno** ( CPF, COD\_Aluno)

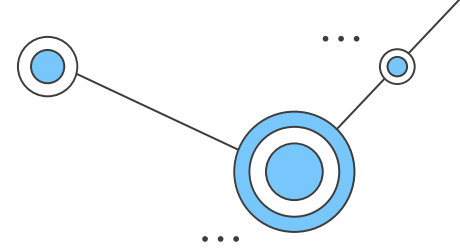
CPF =====> **Usuário**(CPF)

**Email** (CPF, email)

CPF =====> Usuário(CPF)



# Modelo Lógico



**Entrega** ( CPF, ISBN, [COD\_Multa] )

CPF =====> **Usuário**(CPF)

ISBN =====> **Livro**(ISBN)

COD\_Multa =====> **Multa**(COD\_Multa)

**Livro** ( ISBN, Autor, Título, Tradução, COD\_Biblioteca, COD )

COD\_Biblioteca, COD =====> **Estante**(COD\_Biblioteca, COD)

**Bibliotecario** ( COD\_Bibliotecário, nome, COD\_Biblioteca )

COD\_Biblioteca =====> **Biblioteca**(COD)

Gerenciador =====> **Bibliotecario**(COD\_Bibliotecário)

**Está** ( ISBN, COD\_Biblioteca )

ISBN =====> **Livro**(ISBN)

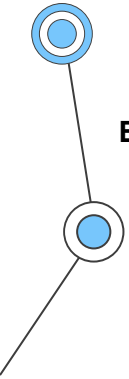
COD\_Biblioteca =====> **Biblioteca**(COD)

**Aluga** ( CPF, ISBN, COD\_Biblioteca )

CPF =====> **Usuário**(CPF)

ISBN =====> **Livro**(ISBN)

COD\_Biblioteca =====> **Biblioteca**(COD)



# Modelo Físico

```
-- CRIA TABELA DE USUÁRIO
CREATE TABLE USUARIO(
CPF VARCHAR(4) NOT NULL,
CONSTRAINT PK_USUARIO PRIMARY KEY(CPF));

-- CRIA TABELA DE EMAIL
CREATE TABLE EMAIL(
CPF VARCHAR(4) NOT NULL,
EMAIL VARCHAR(90),
CONSTRAINT PK_EMAIL PRIMARY KEY(CPF, EMAIL),
CONSTRAINT FK_EMAIL_USUARIO FOREIGN KEY(CPF) REFERENCES USUARIO (CPF));

-- CRIA TABELA PROFESSOR
CREATE TABLE PROFESSOR(
CPF VARCHAR(4) NOT NULL,
COD_PROFESSOR VARCHAR(11),
CONSTRAINT PK_PROFESSOR PRIMARY KEY(CPF),
CONSTRAINT FK_PROFESSOR_USUARIO FOREIGN KEY(CPF) REFERENCES USUARIO (CPF));

-- CRIA TABELA ALUNO
CREATE TABLE ALUNO(
CPF VARCHAR(4),
COD_PROFESSOR VARCHAR(11),
CONSTRAINT PK_ALUNO PRIMARY KEY(CPF),
CONSTRAINT FK_ALUNO_USUARIO FOREIGN KEY(CPF) REFERENCES USUARIO (CPF));

-- CRIA TABELA MULTA
CREATE TABLE MULTA (
COD_MULTA VARCHAR(4),
VALOR NUMBER (4),
CONSTRAINT PK_MULTA PRIMARY KEY (COD_MULTA));

-- CRIA TABELA LIVRO
CREATE TABLE LIVRO (
ISBN VARCHAR(13),
AUTOR VARCHAR(80) CONSTRAINT NM_AUTOR_LIVRO NOT NULL,
TITULO VARCHAR(80) NOT NULL,
COD_ESTANTE VARCHAR(4),
COD_BIBLIOTECA VARCHAR(4),
CONSTRAINT PK_LIVRO PRIMARY KEY(ISBN),
CONSTRAINT FK_LIVRO_ESTANTE FOREIGN KEY(COD_ESTANTE, COD_BIBLIOTECA) REFERENCES Estante(COD_ESTANTE, COD_BIBLIOTECA));

-- CRIA TABELA BIBLIOTECA
CREATE TABLE BIBLIOTECA(
COD VARCHAR(4),
END_NUMERO NUMBER(5),
END_LOGRADOURO VARCHAR(30),
CONSTRAINT PK_BIBLIOTECA PRIMARY KEY (COD));
```

```
-- CRIA TABELA Estante
CREATE TABLE Estante (
COD_Estante VARCHAR(4),
COD_BIBLIOTECA VARCHAR(4),
SESSAO VARCHAR(10),
CONSTRAINT PK_Estante PRIMARY KEY(COD_Estante, COD_BIBLIOTECA),
CONSTRAINT FK_Estante_BIBLIOTECA FOREIGN KEY(COD_BIBLIOTECA) REFERENCES BIBLIOTECA(COD));

-- CRIA TABELA BIBLIOTECARIO
CREATE TABLE BIBLIOTECARIO(
COD_BIBLIOTECARIO VARCHAR(3),
NOME VARCHAR(80),
COD_BIBLIOTECA VARCHAR(4),
GERENCIADOR VARCHAR(3),
CONSTRAINT PK_BIBLIOTECARIO PRIMARY KEY(COD_BIBLIOTECARIO),
CONSTRAINT FK_BIBLIOTECARIO_BIBLIOTECA FOREIGN KEY(COD_BIBLIOTECA) REFERENCES BIBLIOTECA(COD),
CONSTRAINT FK_BIBLIOTECARIO_BIBLIOTECARIO FOREIGN KEY(GERENCIADOR) REFERENCES BIBLIOTECARIO(COD_BIBLIOTECARIO));

----- RELACIONAMENTOS -----
-- CRIA TABELA ENTREGA

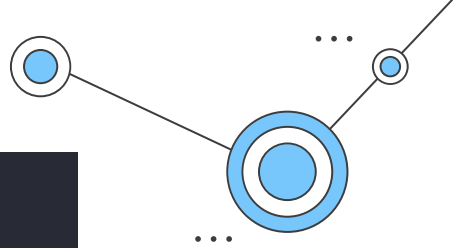
CREATE TABLE ENTREGA (
CPF VARCHAR(4) NOT NULL,
ISBN VARCHAR(13) NOT NULL,
COD_MULTA VARCHAR(4) UNIQUE,
CONSTRAINT PK_ENTREGA PRIMARY KEY(CPF, ISBN),
CONSTRAINT FK_ENTREGA_USUARIO FOREIGN KEY(CPF) REFERENCE USUARIO (CPF),
CONSTRAINT FK_ENTREGA_LIVRO FOREIGN KEY(ISBN) REFERENCE LIVRO(ISBN),
CONSTRAINT FK_ENTREGA_MULTA FOREIGN KEY(COD_MULTA) REFERENCE MULTA(COD_MULTA));

-- CRIA TABELA ALUGA

CREATE TABLE ALUGA(
CPF VARCHAR(4) NOT NULL,
ISBN VARCHAR(13) NOT NULL,
COD_BIBLIOTECA VARCHAR(4) NOT NULL,
CONSTRAINT PK_ENTREGA_ALUGA PRIMARY KEY(CPF, ISBN),
CONSTRAINT FK_ENTREGA_USUARIO_ALUGA FOREIGN KEY(CPF) REFERENCE USUARIO (CPF),
CONSTRAINT FK_ENTREGA_LIVRO_ALUGA FOREIGN KEY(ISBN) REFERENCE LIVRO(ISBN),
CONSTRAINT FK_ENTREGA_BIBLIOTECA_ALUGA FOREIGN KEY(COD_BIBLIOTECA) REFERENCE BIBLIOTECA(COD));
);

-- CRIA TABELA ESTA
CREATE TABLE ESTA(
ISBN VARCHAR(13) NOT NULL,
COD_BIBLIOTECA VARCHAR(4),
CONSTRAINT PK_ESTA PRIMARY KEY(ISBN, COD_BIBLIOTECA),
CONSTRAINT FK_ESTA_LIVRO FOREIGN KEY(ISBN) REFERENCES LIVRO(ISBN),
CONSTRAINT FK_ESTA_BIBLIOTECA FOREIGN KEY(COD_BIBLIOTECA) REFERENCES BIBLIOTECA(COD));
```

# Modelo Físico



```
--Group by/Having
SELECT TRADUCAO, COUNT(*) AS QTD
  FROM LIVRO
 GROUP BY TRADUCAO HAVING TRADUCAO = 'EN-US';

--Junção interna
SELECT A.TITULO, B.COD, B.END_LOGRADOURO
FROM LIVRO A INNER JOIN BIBLIOTECA B ON (A.COD_BIBLIOTECA = B.COD);

--Junção externa
SELECT distinct * FROM USUARIO U
LEFT OUTER JOIN ALUGA A
ON (U.CPF = A.CPF);

--Semi junção
SELECT A.NOME FROM BIBLIOTECARIO A
WHERE EXISTS
(SELECT * FROM BIBLIOTECA B WHERE A.COD_BIBLIOTECA = B.COD);

--Anti-junção
SELECT L.TITULO FROM LIVRO L
WHERE NOT EXISTS
(SELECT * FROM ESTANTE E WHERE L.COD_ESTANTE = E.COD_ESTANTE);
```

```
--Subconsulta do tipo escalar
SELECT A.NOME
FROM BIBLIOTECARIO A
WHERE A.COD_BIBLIOTECARIO = (SELECT AVG(B.COD)
FROM BIBLIOTECA B);

--Subconsulta do tipo linha
SELECT L.TITULO, L.COD_ESTANTE, L.COD_BIBLIOTECA
FROM LIVRO L
WHERE (L.AUTOR, L.TITULO) = (SELECT L.AUTOR, L.TITULO
FROM LIVRO L
WHERE L.ISBN = '000005');

Subconsulta do tipo tabela
SELECT L.AUTOR, L.TITULO
FROM LIVRO L
WHERE L.ISBN = ANY (SELECT L.ISBN
FROM LIVRO L
WHERE L.COD_ESTANTE = '006' AND
L.COD_BIBLIOTECA = '002');

--Operação de conjunto
SELECT A.CPF
FROM ALUNO A
UNION
SELECT P.CPF
FROM PROFESSOR;
```

# Modelo Físico

```
CREATE OR REPLACE PROCEDURE ObterLivroPorISBN(livro_isbn
IN VARCHAR2)
IS
    RESULTADO Livro%ROWTYPE;
BEGIN
    SELECT * INTO RESULTADO FROM LIVRO
    WHERE ISBN = livro_isbn;

    DBMS_OUTPUT.PUT_LINE(RESULTADO.ISBN || ' - ' ||
RESULTADO.Titulo || ' - ' || RESULTADO.Autor);
END;
```