

# EXERCÍCIOS DE BANCO DE DADOS PARA ENTREGA – BIBLIOTECA:

**Aluno:** Flavio Henrique Rosa Tatit Júnior

**Prof.:** Herryson R. Figueiredo

Data: 06/05/2022

### Exercícios de Banco de Dados para entrega – Biblioteca:

Uma biblioteca quer manter informações sobre seus livros. Deseja-se armazenar para os livros as seguintes características: ISBN, título, ano, editora e autor do livro. Para os autores, deseja-se manter: nome e nacionalidade. Cada livro da biblioteca pertence a uma categoria, então, a biblioteca deseja manter um cadastro de todas as categorias existentes, com informações como: código da categoria e descrição.

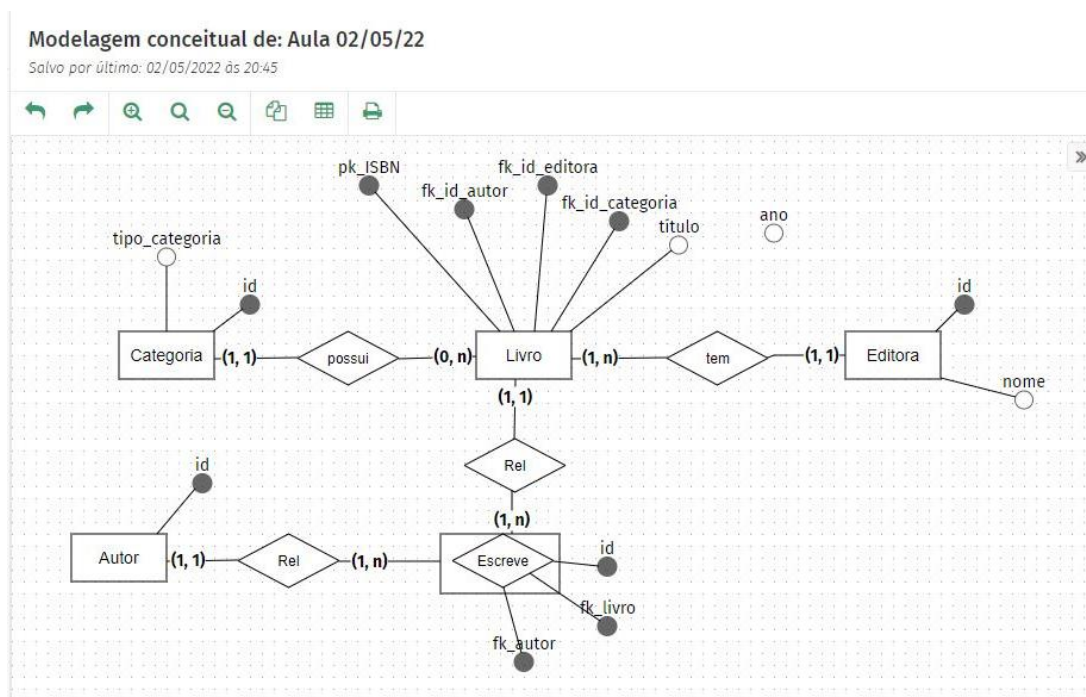
Inicialmente a biblioteca conta com os seguintes livros:

ISBN	Título	Ano	Editora	Autor / Nacionalidade	Categoria
8532511015	Harry Potter e a Pedra Filosofal	2000	Rocco	J. K. Rowling (Inglaterra)	Literatura Juvenil
9788578270698	As Crônicas de Nárnia	2009	Wmf Martins Fontes	Clive Staples Lewis (Inglaterra)	Literatura Juvenil
9788577343348	O Espadachim de Carvão	2013	Casa da Palavra	Affonso Solano (Brasil)	Ficção Científica
9788581742458	O Papai é Pop	2015	Belas Letras	Marcos Piangers (Brasil)	Humor
9788582302026	Pior que tá não Fica	2015	Matrix	Ciro Botelho - Tiririca (Brasil)	Humor
9788577345670	Garota Desdobrável	2015	Casa da Palavra	Bianca Mól (Brasil)	Literatura Juvenil
8532512062	Harry Potter e o Prisioneiro de Azkaban	2000	Rocco	J. K. Rowling (Inglaterra)	Literatura Juvenil

1. Relacionar as entidades e seus respectivos atributos para este banco de dados;

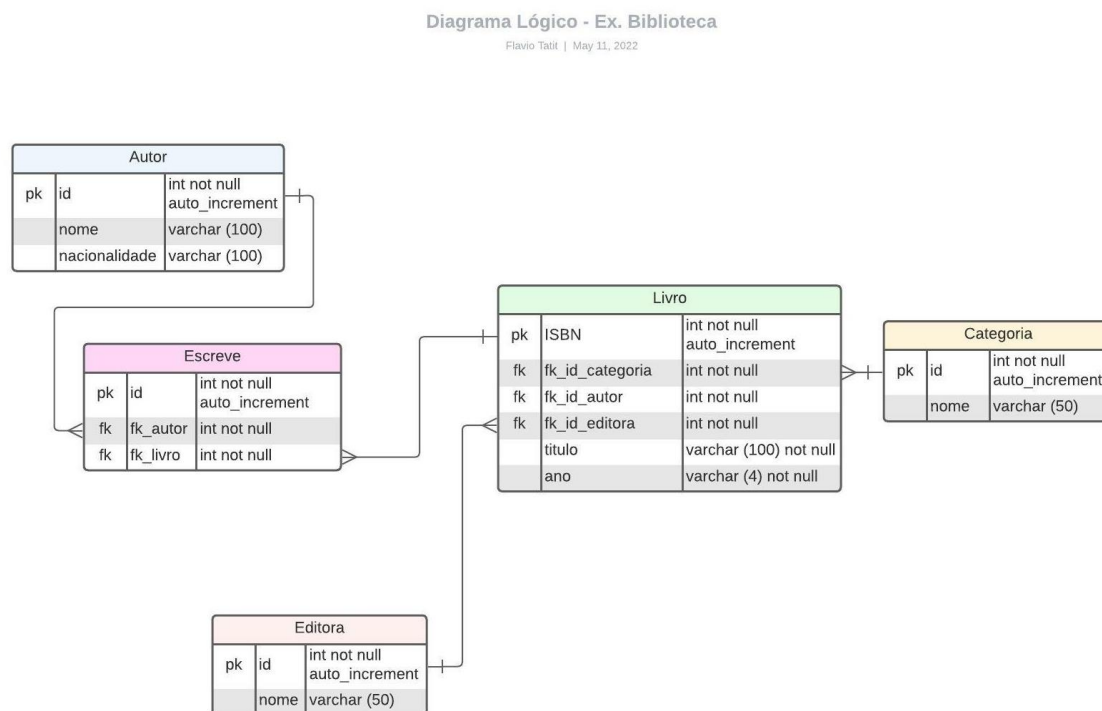
Entidades	Atributos
Livro	titulo / ano
Categoria	tipo_categoria
Editora	nome
Escreve	-
Autor	nome / nacionalidade

2. Crie o modelo conceitual desde banco de dados;



Criado através da ferramenta web: **BR Modelo Web**  
(<https://app.brmodeloweb.com/#!/>)

### 3. Crie o modelo lógico deste banco de dados;



Criado através da ferramenta web: **Lucid**  
(<https://lucid.app/>)

Após a finalização do modelo lógico, através do Lucid, podemos exportar os dados do diagrama para alguns formatos, conforme a seguir:



Exportando para MySQL:

Para utilizarmos o Microsoft SQL Server Management Studio 18, precisamos alterar o comando “**auto\_increment**” por “**identity**”, conforme a seguir:

```
CREATE TABLE Categoria (
    id int not null identity,
    nome varchar (50),
    PRIMARY KEY ( id )
);

CREATE TABLE Editora (
    id int not null identity,
    nome varchar (50),
    PRIMARY KEY ( id )
);

CREATE TABLE Livro (
    ISBN int not null identity,
    fk_id_categoria int not null,
    fk_id_autor int not null,
    fk_id_editora int not null,
    titulo varchar (100) not null,
    ano varchar (4) not null,
    FOREIGN KEY ( fk_id_categoria ) REFERENCES Categoria ( id ),
    FOREIGN KEY ( fk_id_editora ) REFERENCES Editora ( id ),
    PRIMARY KEY ( ISBN ),
);

CREATE TABLE Autor (
    id int not null identity,
    nome varchar (100),
    nacionalidade varchar (100),
    PRIMARY KEY ( id )
);

CREATE TABLE Escreve (
    id int not null identity,
    fk_autor int not null,
    fk_livro int not null,
    FOREIGN KEY ( fk_autor ) REFERENCES Autor ( id ),
    FOREIGN KEY ( fk_livro ) REFERENCES Livro ( ISBN ),
    PRIMARY KEY ( id ),
);
```

**4. Crie o comando SQL para a criação do bando de dados BIBLIOTECA;**

```
/*Criação Banco de dados "Biblioteca": */  
CREATE DATABASE Biblioteca;
```

**5. Crie o comando SQL para a criação das tabelas solicitadas;**

```
/*Criação das tabelas: */  
CREATE TABLE Categoria (  
    id int not null identity,  
    nome varchar (50),  
    PRIMARY KEY ( id )  
);  
  
CREATE TABLE Editora (  
    id int not null identity,  
    nome varchar (50),  
    PRIMARY KEY ( id )  
);  
  
CREATE TABLE Livro (  
    ISBN int not null identity,  
    fk_id_categoria int not null,  
    fk_id_autor int not null,  
    fk_id_editora int not null,  
    titulo varchar (100) not null,  
    ano varchar (4) not null,  
    FOREIGN KEY ( fk_id_categoria ) REFERENCES Categoria ( id ),  
    FOREIGN KEY ( fk_id_editora ) REFERENCES Editora ( id ),  
    PRIMARY KEY ( ISBN ),  
  
);  
  
CREATE TABLE Autor (  
    id int not null identity,  
    nome varchar (100),  
    nacionalidade varchar (100),  
    PRIMARY KEY ( id )  
);  
  
CREATE TABLE Escreve (  

```

```

        id int not null identity,
        fk_autor int not null,
        fk_livro int not null,
        FOREIGN KEY ( fk_autor ) REFERENCES Autor ( id ),
        FOREIGN KEY ( fk_livro ) REFERENCES Livro ( ISBN ),
        PRIMARY KEY ( id ),

);

```

## 6. Crie o comando SQL para inserção dos referidos dados em todas as tabelas;

```

/*Inserindo os dados na tabela "Categoria": */
INSERT into Categoria (nome) values ('Literatura Juvenil');
INSERT into Categoria (nome) values ('Ficção Científica');
INSERT into Categoria (nome) values ('Humor');

select * from Categoria;
/*- - - - - */

/*Inserindo os dados na tabela "Editora": */
INSERT into Editora (nome) values ('Rocco');
INSERT into Editora (nome) values ('Wmf Martins Fontes');
INSERT into Editora (nome) values ('Casa da Palavra');
INSERT into Editora (nome) values ('Belas Letras');
INSERT into Editora (nome) values ('Matrix');

select * from Editora;
/*- - - - - */

/*Inserindo os dados na tabela "Autor/Nacionalidade": */
INSERT into Autor (nome, nacionalidade) values ('J. K. Rowling',
'Inglaterra');
INSERT into Autor (nome, nacionalidade) values ('Clive Staples
Lewis', 'Inglaterra');
INSERT into Autor (nome, nacionalidade) values ('Affonso Solano',
'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Marcos Piangers',
'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Ciro Botelho -
Tiririca', 'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Bianca Mól',
'Brasil');

select * from Autor;
/*- - - - - */

/*Inserindo os dados na tabela "Livro": */

```

```

INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (1,1,1,'Harry Potter e a Pedra Filosofal',
2000);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (1,2,2,'As Crônicas de Nárnia', 2009);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (2,3,3,'O Espadachim de Carvão', 2013);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (3,4,4,'O Papai é Pop', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (3,5,5,'Pior que tá não Fica', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (1,6,1,'Garota Desdobrável', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora,
titulo, ano) values (1,1,1,'Harry Potter e o Prisioneiro de
Azkaban', 2000);

select * from Livro;
/*- - - - - */

```

7. Crie uma consulta para relacionar todos os dados disponíveis de todos os livros existentes na biblioteca em ordem alfabética de título;

```

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Ordenando a tabela em ordem alfabética de nome: */

ORDER BY Livro.titulo;

```

8. Crie uma consulta para relacionar todos os dados disponíveis de todos os livros existentes na biblioteca em ordem alfabética de autor;

```

SELECT Livro.ISBN AS 'ISBN',

```



```

Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Ordenando a tabela em ordem alfabética de autor: */

ORDER BY Autor.nome;

```

9. Crie uma consulta para relacionar todos os dados disponíveis dos livros da categoria de literatura Juvenil em ordem de ano.

```

SELECT Grupo.nome AS 'Grupo',
Dinossauro.nome AS 'Dinossauro',
Dinossauro.ano_descoberta AS 'Ano Descoberta',
Dinossauro.toneladas AS 'Toneladas',
Descobridor.nome AS 'Descobridor',
Era.id AS 'Era',
Era.Ano_de_fim AS 'Início (milhões)',
Era.Ano_de_fim AS 'Fim (milhões)',
Pais.id AS 'País'

FROM Dinossauro, Grupo, Pais, Descobridor, Era
WHERE Dinossauro.id_grupo = Grupo.id
AND Dinossauro.id_pais = Pais.id
AND Dinossauro.id_descobridor = Descobridor.id
AND Dinossauro.id_era = Era.id

/* Filtrando para o grupo de dinossauros "Anquilossauros": */
AND Grupo.nome = 'Anquilossauros'

/* Ordenando a tabela em relação a ordem de descoberta: */
ORDER BY Dinossauro.ano_descoberta

```

10. Crie uma consulta para relacionar todos os dados disponíveis dos livros de humor ou ficção científica com ano entre 2000 e 2010.

```

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Filtrando os livros de humor e ficção científica: */

AND Categoria.nome in('Humor' , 'Ficção Científica')

/* Filtrando para os dados entre os anos de 2000 e 2010: */

AND Livro.ano between 2000 AND 2010

ORDER BY Livro.titulo;

```

# Código do Microsoft SQL Server Management Studio completo:

```
/*Criação Banco de dados "Biblioteca": */
CREATE DATABASE Biblioteca;

USE Biblioteca;

/*Criação das tabelas: */
CREATE TABLE Categoria (
    id int not null identity,
    nome varchar (50),
    PRIMARY KEY ( id )
);

CREATE TABLE Editora (
    id int not null identity,
    nome varchar (50),
    PRIMARY KEY ( id )
);

CREATE TABLE Livro (
    ISBN int not null identity,
    fk_id_categoria int not null,
    fk_id_autor int not null,
    fk_id_editora int not null,
    titulo varchar (100) not null,
    ano varchar (4) not null,
    FOREIGN KEY ( fk_id_categoria ) REFERENCES Categoria ( id ),
    FOREIGN KEY ( fk_id_editora ) REFERENCES Editora ( id ),
    PRIMARY KEY ( ISBN ),
);

CREATE TABLE Autor (
    id int not null identity,
    nome varchar (100),
    nacionalidade varchar (100),
    PRIMARY KEY ( id )
);

CREATE TABLE Escreve (
    id int not null identity,
    fk_autor int not null,
    fk_livro int not null,
    FOREIGN KEY ( fk_autor ) REFERENCES Autor ( id ),
    FOREIGN KEY ( fk_livro ) REFERENCES Livro ( ISBN ),
    PRIMARY KEY ( id ),
);

/*Inserindo os dados na tabela "Categoria": */
INSERT into Categoria (nome) values ('Literatura Juvenil');
INSERT into Categoria (nome) values ('Ficção Científica');
INSERT into Categoria (nome) values ('Humor');
```

```

select * from Categoria;
/*- - - - - */

/*Inserindo os dados na tabela "Editora": */
INSERT into Editora (nome) values ('Rocco');
INSERT into Editora (nome) values ('Wmf Martins Fontes');
INSERT into Editora (nome) values ('Casa da Palavra');
INSERT into Editora (nome) values ('Belas Letras');
INSERT into Editora (nome) values ('Matrix');

select * from Editora;
/*- - - - - */

/*Inserindo os dados na tabela "Autor/Nacionalidade": */
INSERT into Autor (nome, nacionalidade) values ('J. K. Rowling', 'Inglaterra');
INSERT into Autor (nome, nacionalidade) values ('Clive Staples Lewis', 'Inglaterra');
INSERT into Autor (nome, nacionalidade) values ('Affonso Solano', 'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Marcos Piangers', 'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Ciro Botelho - Tiririca', 'Brasil');
INSERT into Autor (nome, nacionalidade) values ('Bianca Mól', 'Brasil');

select * from Autor;
/*- - - - - */

/*Inserindo os dados na tabela "Livro": */
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(1,1,1,'Harry Potter e a Pedra Filosofal', 2000);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(1,2,2,'As Crônicas de Nárnia', 2009);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(2,3,3,'O Espadachim de Carvão', 2013);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(3,4,4,'O Papai é Pop', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(3,5,5,'Pior que tá não Fica', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(1,6,1,'Garota Desdobraível', 2015);
INSERT into Livro (fk_id_categoria, fk_id_autor, fk_id_editora, titulo, ano) values
(1,1,1,'Harry Potter e o Prisioneiro de Azkaban', 2000);

select * from Livro;
/*- - - - - */

/*
7. Crie uma consulta para relacionar todos os dados disponíveis de todos
os livros existentes na biblioteca em ordem alfabética de título; */

```

```

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

```

```

/* Ordenando a tabela em ordem alfabética de nome: */

ORDER BY Livro.titulo;

/*- - - - - */

/*
8. Crie uma consulta para relacionar todos os dados disponíveis de todos
os livros existentes na biblioteca em ordem alfabética de autor; */

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Ordenando a tabela em ordem alfabética de autor: */

ORDER BY Autor.nome;

/*- - - - - */

/*
9. Crie uma consulta para relacionar todos os dados disponíveis
dos livros da categoria de literatura Juvenil em ordem de ano:*/

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Filtrando os livros de humor e ficção científica: */

AND Categoria.nome = 'Literatura Juvenil'

/* Ordenando a tabela em ordem do ano: */

ORDER BY Livro.ano;

```

```

/*- - - - - */

/*
10. Crie uma consulta para relacionar todos os dados disponíveis
dos livros de humor ou ficção científica com ano entre 2000 e 2010:*/

SELECT Livro.ISBN AS 'ISBN',
Livro.titulo AS 'Título',
Livro.ano AS 'Ano',
Editora.nome AS 'Editora',
Autor.nome AS 'Autor',
Autor.nacionalidade AS 'Nacionalidade',
Categoria.nome AS 'Categoria'

FROM Livro, Categoria, Editora, Autor

WHERE Livro.fk_id_categoria = Categoria.id
AND Livro.fk_id_editora = Editora.id
AND Livro.fk_id_autor = Autor.id

/* Filtrando os livros da categoria "humor" e "ficção científica": */

AND Categoria.nome in('Humor' , 'Ficção Científica')

/* Filtrando para os dados entre os anos de 2000 e 2010: */

AND Livro.ano between 2000 AND 2020

ORDER BY Livro.titulo;

/*- - - - - */

```