1. Criar o script para as tabelas com base no seguinte modelo de dados (utilizar autoincremento quando possível e padronizar nomenclatura):

• Tb\_Editora (cod\_editora (pk), descricao, endereço)

CREATE TABLE tb\_editora (

cod\_editora INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

descricao VARCHAR(50),

endereco VARCHAR(50)

);

SELECT \* FROM tb\_editora;

• Tb\_Livro (cod\_livro (pk), ISBN, Título, num\_edicao, Preco, cod\_editora (fk))

CREATE TABLE tb\_livro (

cod\_livro INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

ISBN VARCHAR(20),

num\_edicao INT,

preco FLOAT,

cod\_editora INT UNSIGNED,

FOREIGN KEY (cod\_editora) REFERENCES tb\_editora (cod\_editora)

);

SELECT \* FROM tb\_livro;

• TB\_Autor (Cod\_autor (pk), nome, sexo, data\_nascimento)

CREATE TABLE tb\_autor(

cod\_autor INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(45),

sexo CHARACTER,

data\_nascimento DATETIME

);

SELECT \* FROM tb\_autor;

• TB\_Livro\_Autor (cód\_livro (pk)(fk), cód\_autor (pk) (fk))

CREATE TABLE tb\_livro\_autor(

cód\_livro INT UNSIGNED,

cód\_autor INT UNSIGNED,

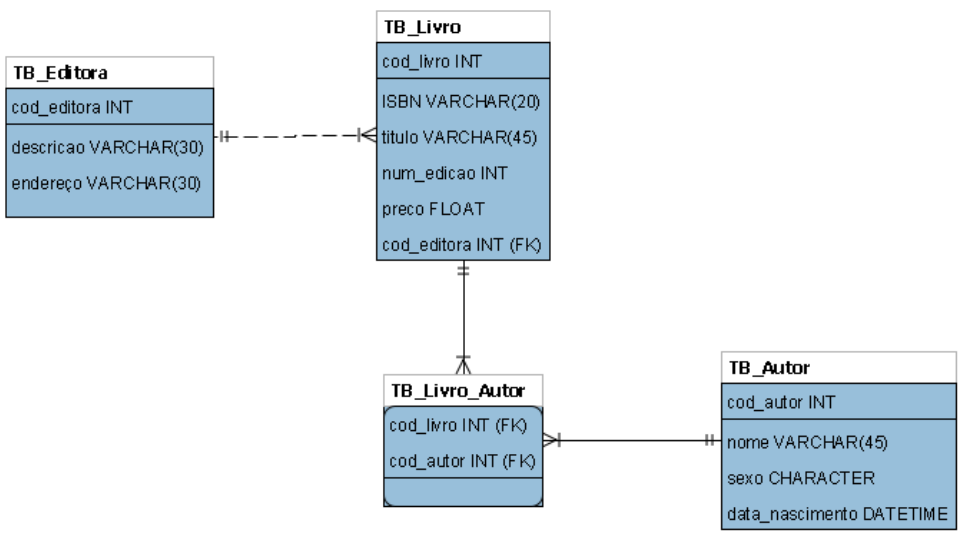
PRIMARY KEY (cód\_livro, cód\_autor),

FOREIGN KEY (cód\_livro) REFERENCES tb\_livro(cod\_livro),

FOREIGN KEY (cód\_autor) REFERENCES tb\_autor(cod\_autor)

);

SELECT \* FROM tb\_livro\_autor;



2. Criar script de povoamento (inserção de dados) do banco de dados:

1. TB\_Editora :

a. ‘Campus’, ‘Rua do Timbó’

INSERT INTO tb\_editora (descricao, endereco) VALUES ('Campus', 'Rua do Timbó');

SELECT \* FROM tb\_editora;

b. ‘Abril’

INSERT INTO tb\_editora(descricao) VALUES (‘Abril’');

SELECT \* FROM tb\_editora;

c. ‘Editora Teste’

INSERT INTO tb\_editora(descricao) VALUES ('Editora Teste');

SELECT \* FROM tb\_editora;

2. TB\_Livro :

a. ‘12345’,’Banco de Dados’,3, 70.00, 1

INSERT INTO tb\_livro (ISBN, titulo, num\_edicao, preco, cod\_editora)

VALUES ('12345', 'Banco de Dados', 3, 70.00, 1);

SELECT \* FROM tb\_livro;

b. ‘35790’,’SGBD’,1, 85.00, 2

INSERT INTO tb\_livro (ISBN, titulo, num\_edicao, preco, cod\_editora)

VALUES ('35790', 'SGBD', 1, 85.00, 2);

SELECT \* FROM tb\_livro;

c. ‘98765’,’Redes de Computadores’,2, 80.00,2

INSERT INTO tb\_livro (ISBN, titulo, num\_edicao, preco, cod\_editora)

VALUES ('98765', 'Redes de Computadores', 2, 80.00, 2);

SELECT \* FROM tb\_livro;

3. TB\_Autor :

a. ‘João’,’M’,’01.01.1970’

INSERT INTO tb\_autor (nome, sexo, data\_nascimento) VALUES ('João', 'M', '1970-01-01');

SELECT \* FROM tb\_autor;

b. ‘Maria, ‘F’, ’17.05.1974’.

INSERT INTO tb\_autor (nome, sexo, data\_nascimento) VALUES ('Maria', 'F', '1974-05-17');

SELECT \* FROM tb\_autor;

c. ‘José’, ‘M’, ’10.10.1977’.

INSERT INTO tb\_autor (nome, sexo, data\_nascimento) VALUES ('José', 'M', '1977-10-10');

SELECT \* FROM tb\_autor;

d. ‘Carla’, ‘F’, 08.12.1964’

INSERT INTO tb\_autor (nome, sexo, data\_nascimento) VALUES ('Carla', 'F', '1964-12-08');

SELECT \* FROM tb\_autor;

4. TB\_Livro\_Autor:

a. Banco de Dados – Autores: João e Maria

SELECT \* FROM tb\_livro WHERE titulo = 'Banco de Dados';

SELECT \* FROM tb\_autor WHERE nome IN ('João','Maria');

INSERT INTO tb\_livro\_autor (cód\_livro, cód\_autor) VALUES ('1', '1'), ('1', '2');

SELECT \* FROM tb\_livro\_autor;

b. SGBD – Autores: Maria e Carla

SELECT \* FROM tb\_livro WHERE titulo = 'SGBD';

SELECT \* FROM tb\_autor WHERE nome IN ('Maria', 'Carla');

INSERT INTO tb\_livro\_autor (cód\_livro, cód\_autor) VALUES ('2', '2'), ('2', '4');

SELECT \* FROM tb\_livro\_autor;

c. Redes de Computadores – Autor: José

SELECT \* FROM tb\_livro WHERE titulo = 'Redes de Computadores';

SELECT \* FROM tb\_autor WHERE nome = 'José';

INSERT INTO tb\_livro\_autor (cód\_livro, cód\_autor) VALUES ('3','3');

SELECT \* FROM tb\_livro\_autor;

3. Criar script com comandos DML para as seguintes solicitações:

1. Excluir a ‘Editora Teste’

SELECT \* FROM tb\_editora;

DELETE FROM tb\_editora WHERE cod\_editora = '3';

1. Apresentar o nome e data de nascimento de todos os autores

SELECT nome, data\_nascimento FROM tb\_autor;

1. Apresentar o nome e a data de nascimento dos autores por ordem de nome.

SELECT nome, data\_nascimento FROM tb\_autor ORDER BY nome;

1. Apresentar o nome e a data de nascimento dos autores do sexo feminino ordenados pelo nome.

SELECT nome, data\_nascimento FROM tb\_autor WHERE sexo = 'F' ORDER BY nome;

1. Apresentar o nome das editoras que não tem o endereço cadastrado.

SELECT descricao FROM tb\_editora WHERE endereco IS NULL;

1. Apresentar o título do livro e o nome da sua editora

SELECT tb\_livro.titulo, tb\_editora.descricao FROM tb\_livro JOIN tb\_editora ON tb\_livro.cod\_editora = tb\_editora.cod\_editora;

1. .Apresentar o título do livro e o nome da sua editora. Caso haja alguma editora sem livro publicado, informar os dados da editora com valores nulos para os livros.

SELECT tb\_livro.titulo, tb\_editora.descricao FROM tb\_editora LEFT JOIN tb\_livro ON tb\_editora.cod\_editora = tb\_livro.cod\_editora;

1. Apresentar o título do livro e o nome dos seus autores

SELECT tb\_livro.titulo, GROUP\_CONCAT(tb\_autor.nome) AS autores FROM tb\_livro JOIN tb\_livro\_autor ON tb\_livro.cod\_livro = tb\_livro\_autor.cód\_livro

JOIN tb\_autor ON tb\_autor.cod\_autor = tb\_livro\_autor.cód\_autor GROUP BY tb\_livro.titulo;

1. Apresentar o nome da editora e o nome dos autores que já publicaram algum livro na editora.

SELECT tb\_editora.descricao, GROUP\_CONCAT(tb\_autor.nome) AS autores FROM tb\_editora JOIN tb\_livro ON tb\_livro.cod\_editora = tb\_editora.cod\_editora

JOIN tb\_autor ON tb\_autor.cod\_autor = tb\_livro.cod\_livro GROUP BY tb\_editora.descricao;

1. Apresentar o título dos livros que começam a string ‘Banco’.

SELECT titulo FROM tb\_livro WHERE titulo LIKE 'Banco%';

1. Apresentar o título dos livros que tem a string ‘do’.

SELECT titulo FROM tb\_livro WHERE titulo LIKE '%do%';

1. Apresentar o nome de cada livro e seu preço reajustado em 5%

SELECT titulo, preco \* 1.05 AS preco FROM tb\_livro;

1. Apresentar o nome dos autores que nasceram no mês de outubro

SELECT nome, data\_nascimento FROM tb\_autor WHERE EXTRACT(MONTH FROM data\_nascimento) = 08;

1. Apresentar o número de livros do acervo

SELECT COUNT(\*) AS 'Livros do acervo' FROM tb\_livro;

1. Apresentar o número de autores do livro ‘Banco de Dados’

SELECT COUNT(DISTINCT tb\_autor.cod\_autor) AS 'Quantidade de autor(es)'

FROM tb\_autor

JOIN tb\_livro\_autor ON tb\_autor.cod\_autor = tb\_livro\_autor.cód\_autor

JOIN tb\_livro ON tb\_livro.cod\_livro = tb\_livro\_autor.cód\_livro

WHERE tb\_livro.titulo = 'Banco de Dados';

1. Apresentar o somatório dos preços dos livros do acervo

SELECT SUM(preco) AS 'Soma dos preços' FROM tb\_livro;

1. Apresentar a média de preços dos livros da editora Campus

SELECT AVG(preco) AS 'Media do valor do(s) livro(s) da editora Campus'

FROM tb\_livro

JOIN tb\_editora ON tb\_livro.cod\_editora = tb\_editora.cod\_editora

WHERE tb\_editora.descricao = 'Campus';

1. Apresentar o maior preço dentre todos os livros do acervo.

SELECT MAX(preco) AS 'Mais caro' FROM tb\_livro;

1. Apresentar a data de nascimento do autor mais velho

SELECT MIN(data\_nascimento) AS 'Autor mais Velho' FROM tb\_autor;

1. Apresentar o número de livros por editora

SELECT tb\_editora.descricao AS 'Editoras', COUNT(tb\_livro.cod\_livro) AS 'Numeros de Livros' FROM tb\_editora

JOIN tb\_livro ON tb\_livro.cod\_editora = tb\_editora.cod\_editora

GROUP BY tb\_editora.cod\_editora;

1. Apresentar o somatório e média de preço dos livros por editora

SELECT tb\_editora.descricao AS 'Editora', AVG(tb\_livro.preco) AS 'Media de preço dos livros', SUM(tb\_livro.preco) AS 'Soma de preço dos livros'

FROM tb\_editora JOIN tb\_livro ON tb\_livro.cod\_editora = tb\_editora.cod\_editora GROUP BY tb\_editora.cod\_editora;

1. Apresentar o número de autores por livro, mas apenas dos livros que possuem mais de 1 autor

SELECT tb\_livro.titulo AS 'Titulo', COUNT(tb\_livro\_autor.cód\_autor) AS 'Numero de autor(es)' FROM tb\_livro

JOIN tb\_livro\_autor ON tb\_livro\_autor.cód\_livro = tb\_livro.cod\_livro

GROUP BY tb\_livro.cod\_livro HAVING COUNT(tb\_livro\_autor.cód\_autor) > 1;

25. Apresentar a média de preços geral por editora, mas apenas as editoras que possuem média maior que R$ 80,00

SELECT tb\_editora.descricao AS 'Editora', AVG(tb\_livro.preco) AS 'Media' FROM tb\_editora

JOIN tb\_livro ON tb\_editora.cod\_editora = tb\_livro.cod\_editora

GROUP BY tb\_editora.cod\_editora HAVING AVG(tb\_livro.preco) > 80.00;

26. Apresentar o nome dos autores que não são autores do livro Banco de Dados

SELECT tb\_autor.nome AS 'Nome de autor(es)' FROM tb\_autor

JOIN tb\_livro\_autor ON tb\_autor.cod\_autor = tb\_livro\_autor.cód\_autor

JOIN tb\_livro ON tb\_livro.cod\_livro = tb\_livro\_autor.cód\_livro

WHERE tb\_livro.titulo != 'Banco de Dados';

27. Apresentar a quantidade de livros da editora Campus e Abril em colunas diferentes.

SELECT COUNT(CASE WHEN tb\_editora.descricao = 'Campus' THEN tb\_livro.cod\_livro END) AS 'Quantida Campus', COUNT(CASE WHEN tb\_editora.descricao = 'Abril' THEN tb\_livro.cod\_livro END) AS 'Quantida Abril' FROM tb\_editora JOIN tb\_livro ON tb\_editora.cod\_editora = tb\_livro.cod\_editora WHERE tb\_editora.descricao IN ('Campus', 'Abril');