Criando o banco da empresa:

CREATE TABLE departamentos(

codigoDepartamento INT NOT NULL IDENTITY (1,1) PRIMARY KEY,

nome VARCHAR(50),

localizacao VARCHAR(100)

);

CREATE TABLE funcionarios(

codigoFuncionario INT NOT NULL IDENTITY (1,1) PRIMARY KEY,

PrimeiroNome VARCHAR(30),

segundoNome VARCHAR(30),

ultimoNome VARCHAR(30),

DataNasci DATETIME,

CPF VARCHAR (30),

RG VARCHAR(30),

endereco VARCHAR(100),

cep VARCHAR(30),

cidade VARCHAR(30),

fone VARCHAR(30),

codigoDepartamento INT REFERENCES departamentos (codigoDepartamento),

funcao VARCHAR(100),

salario DECIMAL(8,2)

);

ALTER TABLE departamentos

ADD codigoFuncionarioGerente INT REFERENCES funcionarios(codigoFuncionario);

Lista 01:

/\*QUESTÃO 01 - LISTA 01\*/

SELECT primeironome, segundonome

FROM funcionarios

ORDER BY segundoNome;

CREATE PROCEDURE P\_SEL\_LISTAR\_ORDENADO\_SOBRENOME

AS

SELECT primeironome, segundonome

FROM funcionarios

ORDER BY segundoNome;

/\*QUESTÃO 02 - LISTA 01\*/

SELECT \*

FROM funcionarios

ORDER BY cidade ASC;

/\*QUESTÃO 03 - LISTA 01\*/

SELECT \*

FROM funcionarios

WHERE salario > 1000

ORDER BY primeiroNome ASC, SEGUNDONOme ASC, ULTIMONOME ASC;

CREATE PROCEDURE P\_SEL\_MAIOR\_SALARIO\_QUE

@salario DECIMAL(8,2)

AS

SELECT \*

FROM funcionarios

WHERE salario > @salario

ORDER BY primeiroNome ASC, SEGUNDONOme ASC, ULTIMONOME ASC;

/\*QUESTÃO 04 - LISTA 01\*/

SELECT datanasci, primeironome

FROM funcionarios

ORDER BY datanasci ASC;

/\*QUESTÃO 05 - LISTA 01\*/

SELECT \*

FROM funcionarios

ORDER BY primeironome ASC, segundonome ASC;

/\*QUESTÃO 06 - LISTA 01\*/

SELECT SUM(SALARIO)

FROM funcionarios;

CREATE PROCEDURE P\_SEL\_TOTAL\_FOLHA

AS

SELECT SUM(salario)

FROM funcionarios;

/\*QUESTÃO 07 - LISTA 01\*/

SELECT f.primeironome, d.nome, f.funcao

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d. codigodepartamento;

/\*QUESTÃO 08 - LISTA 01\*/

SELECT d.nome, f.primeiroNome

FROM funcionarios f

JOIN departamentos d

ON f.codigofuncionario = d. codigofuncionariogerente;

CREATE PROCEDURE P\_SEL\_DEPARTAMENTOS\_GERENTES

AS

SELECT d.nome, f.primeiroNome

FROM funcionarios f

JOIN departamentos d

ON f.codigofuncionario = d. codigofuncionariogerente;

/\*QUESTÃO 09 - LISTA 01\*/

SELECT SUM(f.salario) Total, d.nome Departamento

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d. codigodepartamento

GROUP BY d.nome;

CREATE PROCEDURE P\_SEL\_FOLHA\_DEPARTAMENTO

AS

SELECT SUM(f.salario) Total, d.nome Departamento

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d.codigodepartamento

GROUP BY d.nome;

/\*QUESTÃO 10 - LISTA 01\*/

SELECT d.nome, f.primeiroNome

FROM funcionarios f

JOIN departamentos d

ON f.codigofuncionario = d. codigofuncionariogerente

WHERE f.funcao = 'supervisor';

CREATE PROCEDURE P\_SEL\_DEPARTAMENTO\_FUNCIONARIOS\_QUE

@funcao VARCHAR(30)

AS

SELECT d.nome, f.primeiroNome

FROM funcionarios f

JOIN departamentos d

ON f.codigofuncionario = d. codigofuncionariogerente

WHERE f.funcao = @funcao;

/\*QUESTÃO 11 - LISTA 01\*/

SELECT COUNT(codigofuncionario)

FROM funcionarios;

CREATE PROCEDURE P\_SEL\_TOTAL\_FUNCIONARIOS

AS

SELECT COUNT(codigofuncionario)

FROM funcionarios;

/\*QUESTÃO 12 - LISTA 01\*/

SELECT AVG(salario)

FROM funcionarios;

CREATE PROCEDURE P\_SEL\_MEDIA\_SALARIOS

AS

SELECT AVG(salario)

FROM funcionarios;

/\*QUESTÃO 13 - LISTA 01\*/

SELECT MIN(salario) Menor, d.nome Departamento

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d. codigodepartamento

GROUP BY d.nome;

CREATE PROCEDURE P\_SEL\_MENOR\_SALARIO\_POR\_DEPARTAMENTO

AS

SELECT MIN(salario) Menor, d.nome Departamento

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d. codigodepartamento

GROUP BY d.nome;

/\*QUESTÃO 14 - LISTA 01\*/

SELECT primeironome,segundonome,ultimonome

FROM funcionarios

WHERE segundonome IS NULL;

/\*QUESTÃO 15 - LISTA 01\*/

SELECT D.NOME Departamento, f.primeironome

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d. codigodepartamento

ORDER BY d.nome, f.primeironome;

/\*QUESTÃO 16 - LISTA 01\*/

SELECT primeironome, segundonome

FROM funcionarios

WHERE cidade = 'recife' AND funcao = 'Telefonista';

CREATE PROCEDURE P\_SEL\_CIDAED\_FUNCAO

@cidade VARCHAR(30),

@funcao VARCHAR(30)

AS

SELECT primeironome, segundonome

FROM funcionarios

WHERE cidade = @cidade AND funcao = @funcao;

/\*QUESTÃO 17 - LISTA 01\*/

SELECT f.primeironome, f.segundonome

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d.codigodepartamento

WHERE d.nome = 'Pessoal';

CREATE PROCEDURE P\_SEL\_FUNCIONARIOS\_DEPARTAMENTO

@departamento VARCHAR(30)

AS

SELECT f.primeironome, f.segundonome

FROM funcionarios f

JOIN departamentos d

ON f.codigodepartamento = d.codigodepartamento

WHERE d.nome = @departamento;

/\*QUESTÃO 18 - LISTA 01\*/

SELECT primeironome, segundonome

FROM funcionarios

WHERE salario > (SELECT MIN(f.salario)

FROM funcionarios f

JOIN departamentos d

ON f.codigofuncionario = d. codigofuncionariogerente

);

Criando o banco da livraria

CREATE TABLE genero(

idgenero INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

descricao NVARCHAR(100)

);

CREATE TABLE autor(

idautor INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

nome NVARCHAR(100),

email NVARCHAR(100)

);

CREATE TABLE livro(

idlivro INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

titulo NVARCHAR(100),

preco FLOAT,

estoque INT,

idgenero INT,

CONSTRAINT fk\_genero\_livro FOREIGN KEY (idgenero)REFERENCES genero(idgenero)

);

CREATE TABLE escreve(

idlivro INT REFERENCES livro(idlivro),

idautor INT REFERENCES autor(idautor)

);

CREATE TABLE cliente(

idcliente INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

nome NVARCHAR(100),

telefone NVARCHAR(50)

);

CREATE TABLE venda(

idvenda INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

data DATETIME,

total float,

idcliente INT REFERENCES cliente(idcliente)

);

CREATE TABLE itens\_da\_venda(

idvenda INT REFERENCES venda(idvenda),

idlivro INT REFERENCES livro(idlivro),

qtd INT,

subtotal NVARCHAR(45)

);

/\*INSERSÕES\*/

INSERT INTO cliente(nome, telefone)

VALUES

('Elaine Luciana', '987456345'),

('Tiago', '645745745'),

('Daniela', '75346345474'),

('Asdubral', '75647657'),

('Daniel', '54756854'),

('Anastácia', '4635464654')

;

INSERT INTO genero(descricao)

VALUES

('Medicina'),

('Comédia'),

('Infantil'),

('Suspense'),

('Drama'),

('Bibliografia')

;

INSERT INTO livro(titulo, preco, estoque, idgenero)

VALUES

('Bichos Monstruosos', 22.87, 15, 3),

('A medicina', 869.75, 7, 1),

('A medicina II', 756.43, 3, 1),

('Comedia em pé', 67.8, 0, 2),

('A volta dos que não foram', 124.22, 13, 5),

('Italo Hiago Lopes', 386.44, 11, 6)

;

INSERT INTO autor(nome, email)

VALUES

('Robson Pinheiro', 'Robson@escritor.com.br'),

('Hilary Duff', 'hilarymedicina@email.com'),

('Maria José Valero', 'emailpessoa@maria.com'),

('Italo Hiago', 'italohiagolopes@outlook.com'),

('Evandro Luz', 'evandro@hotmail.com'),

('Asdrubalstacio ', 'asd\_89@outlook.com.br')

;

INSERT INTO escreve(idlivro, idautor)

VALUES

(1,5),

(1,6),

(4,1),

(5, 1),

(3,1),

(4,6),

(5,2),

(2,2),

(2,3)

;

INSERT INTO venda (data, total, idcliente)

VALUES

('2017-02-01 12:32:00', 235.43, 3),

('2017-03-01 15:22:23', 547.12, 1),

('2017-07-08 22:43:12', 4575.32, 2),

('2017-08-08 14:34:32', 123.34, 3),

('2017-08-03 09:43:42', 333.54, 4),

('2017-11-06 08:32:41', 455.45, 5),

('2017-12-06 14:11:30', 435.56, 6),

('2017-06-12 16:42:07', 586.76, 7),

('2016-12-12 22:32:06', 577.76, 8),

('2017-05-06 23:12:12', 655.76, 2),

('2017-04-04 09:10:32', 123.67, 1),

('2017-02-12 11:11:09', 432.87, 7),

('2017-12-09 16:05:16', 444.78, 8)

;

INSERT itens\_da\_venda(idvenda, idlivro, qtd, subtotal)

VALUES

(1, 6, 2, 543.12),

(1, 4, 3, 345.12),

(2, 1, 1, 546.24),

(3, 1, 6, 465.43),

(5, 2, 4, 645.08),

(6, 3, 2, 678.68),

(7, 4, 3, 123.56),

(4, 5, 1, 321.76),

(8, 6, 1, 213.34),

(9, 3, 4, 132.65),

(10, 1, 3, 231.98),

(11, 2, 1, 123.16),

(12, 5, 1, 345.65),

(13, 4, 2, 123.26)

;

DROP TABLE itens\_da\_venda, venda, cliente, escreve, autor, livro, genero;

Lista 02

/\*QUESTÃO 01 LISTA 02\*/

SELECT a.nome

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON l.idlivro = e.idlivro

WHERE l.estoque = 0;

CREATE PROCEDURE P\_SEL\_AUTOR\_ESTOQUE\_ZERADO

AS

SELECT a.nome

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON l.idlivro = e.idlivro

WHERE l.estoque = 0;

/\*QUESTÃO 02 LISTA 02\*/

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

WHERE l.titulo = 'Bichos Monstruosos'

GROUP BY c.nome;

CREATE PROCEDURE P\_SEL\_QUEM\_COMPROU\_O\_LIVRO

@livro VARCHAR(30)

AS

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

WHERE l.titulo = @livro

GROUP BY c.nome;

/\*QUESTÃO 03 LISTA 02\*/

SELECT g.descricao

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON l.idlivro = e.idlivro

JOIN genero g

ON l.idgenero = g.idgenero

WHERE a.nome = 'Robson Pinheiro'

CREATE PROCEDURE P\_SEL\_GENERO\_AUTOR\_ESCREVE

@autor VARCHAR(30)

AS

SELECT g.descricao

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON l.idlivro = e.idlivro

JOIN genero g

ON l.idgenero = g.idgenero

WHERE a.nome = @autor

/\*QUESTÃO 04 LISTA 02\*/

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE a.nome = 'Hilary Duff'

GROUP BY c.nome;

CREATE PROCEDURE P\_SEL\_CLIENTES\_QUE\_COMPRARAM\_DE

@autor VARCHAR(30)

AS

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE a.nome = @autor

GROUP BY c.nome;

/\*QUESTÃO 05 LISTA 02\*/

SELECT g.descricao

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN genero g

ON l.idgenero = g.idgenero

WHERE c.nome = 'Elaine Luciana'

GROUP BY g.descricao;

CREATE PROCEDURE P\_SEL\_GENERO\_CLIENTE\_GOSTA

@cliente VARCHAR(30)

AS

SELECT g.descricao Genero

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN genero g

ON l.idgenero = g.idgenero

WHERE c.nome = @cliente

GROUP BY g.descricao;

/\*QUESTÃO 06 LISTA 02\*/

SELECT l.titulo

FROM livro l

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE a.nome LIKE 'Maria José Valero'

group by L.TITULO;

CREATE PROCEDURE P\_SEL\_ESTOQUE\_AUTOR

@autor VARCHAR(30)

AS

SELECT l.titulo

FROM livro l

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE a.nome = @autor

group by L.TITULO;

/\*QUESTÃO 08 LISTA 02\*/

SELECT A.nome

FROM livro l

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

JOIN itens\_da\_venda i

ON l.idlivro = i.idlivro

WHERE i.qtd > 1

GROUP BY a.nome;

CREATE PROCEDURE P\_SEL\_AUTOR\_VENDEU\_MAIOR\_QUE

@quantidade INT

AS

SELECT A.nome

FROM livro l

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

JOIN itens\_da\_venda i

ON l.idlivro = i.idlivro

WHERE i.qtd >= @quantidade

GROUP BY a.nome;

/\*QUESTÃO 09 LISTA 02\*/

SELECT l.titulo

FROM livro l

JOIN itens\_da\_venda i

ON l.idlivro = i.idlivro

WHERE i.qtd = (SELECT MAX(qtd)

FROM itens\_da\_venda

)

CREATE PROCEDURE P\_SEL\_MAIS\_LIVROS\_VEZ

AS

SELECT l.titulo

FROM livro l

JOIN itens\_da\_venda i

ON l.idlivro = i.idlivro

WHERE i.qtd = (SELECT MAX(qtd)

FROM itens\_da\_venda

)

/\*QUESTÃO 10 LISTA 02\*/

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

WHERE l.preco = ( SELECT MAX(preco)

FROM livro

);

CREATE PROCEDURE P\_SEL\_QUEM\_COMPROU\_LIVRO\_MAIS\_CARO

AS

SELECT c.nome

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

WHERE l.preco = ( SELECT MAX(preco)

FROM livro

);

/\*QUESTÃO 11 LISTA 02\*/

SELECT a.nome

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON e.idlivro = l.idlivro

WHERE l.preco = ( SELECT MIN(preco)

FROM livro

);

CREATE PROCEDURE QUAL\_AUTOR\_TEM\_LIVRO\_MAIS\_BARATO

AS

SELECT a.nome

FROM autor a

JOIN escreve e

ON a.idautor = e.idautor

JOIN livro l

ON e.idlivro = l.idlivro

WHERE l.preco = ( SELECT MIN(preco)

FROM livro

);

/\*QUESTÃO 13 LISTA 02\*/

SELECT AVG(preco)

FROM livro

WHERE idgenero = ( SELECT idgenero

FROM genero

WHERE descricao = 'Medicina'

);

CREATE PROCEDURE MEDIA\_PRECO\_GENERO

@genero VARCHAR(30)

AS

SELECT AVG(preco) Media

FROM livro

WHERE idgenero = ( SELECT idgenero

FROM genero

WHERE descricao = @genero

);

/\*QUESTÃO 15 LISTA 02\*/

select c.nome cliente, l.titulo titulo, a.nome autor

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE v.data = (SELECT MIN(data) from venda)

CREATE PROCEDURE INFOS\_PRIMEIRA\_VENDA

AS

SELECT c.nome cliente, l.titulo titulo, a.nome autor

FROM cliente c

JOIN venda v

ON c.idcliente = v.idcliente

JOIN itens\_da\_venda i

ON v.idvenda = i.idvenda

JOIN livro l

ON i.idlivro = l.idlivro

JOIN escreve e

ON l.idlivro = e.idlivro

JOIN autor a

ON e.idautor = a.idautor

WHERE v.data = (SELECT MIN(data) from venda)