

Linguagem SQL

Guia Prático de Aprendizagem

Luciana Ferreira Baptista

Respostas dos Exercícios

Editora Érica Ltda.

Capítulo 1

1.

```
CREATE DATABASE Concessionaria
```

2.

```
USE Concessionaria
```

3.

```
CREATE TABLE Veiculo (  
    chassi CHAR(17) PRIMARY KEY,  
    marca VARCHAR(10) ,  
    modelo VARCHAR(20) ,  
    anoFabricacao INT,  
    anoModelo INT,  
    combustivel CHAR(1)  
)
```

4.

```
ALTER TABLE Veiculo  
ADD valor money, motor VARCHAR(20)
```

5.

```
ALTER TABLE Veiculo  
DROP COLUMN motor
```

6.

```
CREATE INDEX VeiculoMarcaModelo  
ON Veiculo (marca, modelo)
```

7.

```
CREATE INDEX VeiculoAnoFabricacao  
ON Veiculo (anoFabricacao DESC)
```

8.

```
DROP INDEX VeiculoMarcaModelo  
ON Veiculo
```

9.

```
DROP TABLE Veiculo
```

10.

```
USE master  
DROP DATABASE Concessionaria
```

Capítulo 2

1.

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(5, 'Carlos Dias', 'Av. Lapa, 121', 'Itu', 'SP', 'carlao@gmail.com',
'1990-03-31');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(6, 'Ana Maria da Cunha', 'Av. São Paulo, 388', 'Itu', 'SP',
'aninhacunha@gmail.com', '1988-04-12');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(7, 'Cláudia Regina Martins', 'Rua Holanda, 89', 'Campinas', 'SP',
'cregina@gmail.com', '1988-12-04');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(8, 'Marcela Tatho', 'Rua Bélgica, 43', 'Campinas', 'SP',
'marctatho@hotmail.com', '1987-11-09');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(9, 'Jorge Luis Rodrigues', 'Av. da Saudade, 1989', 'São Paulo',
'SP', 'jorgeluis@yahoo.com.br', '1990-05-05');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(10, 'Ana Paulo Camargo', 'Rua Costa e Silva', 'Jundiaí', 'SP',
'apcamargo@gmail.com', '1991-06-30');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(11, 'Ivo Cunha', 'Av. Raio de Luz, 100', 'Campinas', 'SP', 'ivo@
bol.com.br', '1987-04-11');
```

```
INSERT INTO Funcionario
(idFuncionario, nome, endereco, cidade, estado, email, dataNascto)
VALUES
(12, 'Carlos Luis de Souza', 'Rua Nicolau Coelho, 22', 'São Paulo',
'SP', 'cls@bol.com.br', '1988-04-30');
```

2.

```
UPDATE Funcionario SET
cidade = 'Valinhos'
WHERE cidade = 'Itu'
```

3.

```
UPDATE Funcionario SET
cargo = 'AI', salario = 1100 -- auxiliar de informática
WHERE cidade = 'Valinhos'
```

```
UPDATE Funcionario SET
cargo = 'PC', salario = 1700 -- programador de computador
WHERE cidade = 'Campinas'
```

```
UPDATE Funcionario SET
cargo = 'TI', salario = 750 -- Técnico de informática
WHERE cidade = 'Jundiaí'
```

4.

```
SELECT nome, cargo
FROM Funcionario
```

5.

```
SELECT idFuncionario, email
FROM Funcionario
WHERE estado='SP'
```

6.

```
DELETE FROM Funcionario
WHERE idFuncionario = 5
```

7.

```
SELECT DISTINCT cidade, estado
FROM Funcionario
WHERE cargo='PC'
```

Capítulo 3

1.

```
SELECT nome, salario*1.30
FROM Funcionario
```

2.

```
SELECT nome, salario, salario*0.80
FROM Funcionario
WHERE cidade = 'Campinas'
```

3.

```
SELECT nome, salario
FROM Funcionario
WHERE salario > 1500
```

4.

```
SELECT nome, cidade
FROM Funcionario
WHERE NOT cidade='Valinhos'
```

5.

```
SELECT idFuncionario, cidade
FROM Funcionario
WHERE cidade='Valinhos' OR cidade='Campinas'
```

6.

```
SELECT idFuncionario, cargo
FROM Funcionario
WHERE NOT cidade='São Paulo' AND salario >= 1000
```

7.

```
SELECT nome
FROM Funcionario
WHERE cargo IS NOT NULL
```

8.

```
SELECT nome, salario
FROM Funcionario
WHERE salario BETWEEN 500 AND 1500
```

9.

```
SELECT nome, email
FROM Funcionario
WHERE email LIKE '%hotmail%'
```

10.

```
SELECT nome, email
FROM Funcionario
WHERE email LIKE '%.br'
ORDER BY nome
```

11.

```
SELECT nome, email
FROM Funcionario
WHERE email NOT LIKE '%.com'
ORDER BY nome
```

12.

```
SELECT nome, email
FROM Funcionario
WHERE email LIKE '__r%'
```

Capítulo 4

1.

```
SELECT nome, DAY(dataNascto) dia, MONTH(dataNascto) mes,  
YEAR(dataNascto) ano  
FROM Funcionario
```

2.

```
SELECT DISTINCT DATENAME(MONTH,dataNascto) AS nome_mes  
FROM Funcionario  
ORDER BY nome_mes
```

3.

```
SELECT idFuncionario, nome  
FROM Funcionario  
WHERE YEAR(dataNascto)=1987
```

4.

```
SELECT nome, DAY(dataNascto)  
FROM Funcionario  
WHERE MONTH(dataNascto)=4 AND YEAR(dataNascto)=1988
```

5.

```
SELECT nome, DATEADD(MONTH, 2, dataNascto)  
FROM Funcionario
```

6.

```
SELECT nome, DATEDIFF(YEAR, dataNascto, GETDATE())  
FROM Funcionario
```

7.

```
SELECT idFuncionario, nome, YEAR(dataNascto)  
FROM Funcionario  
WHERE (MONTH(dataNascto) BETWEEN 3 AND 5) AND  
YEAR(dataNascto)=1990
```

8.

```
SELECT nome, YEAR(dataNascto)  
FROM Funcionario  
WHERE estado='SP'
```

9.

```
SELECT nome  
FROM Funcionario  
WHERE YEAR(dataNascto) < 1990
```

10.

```
SELECT DISTINCT cidade, estado  
FROM Funcionario  
WHERE YEAR(dataNascto) > 1989
```

11.

```
SELECT *  
FROM Funcionario  
WHERE YEAR(dataNascto) IN (1988, 1990)
```

12.

```
SELECT nome  
FROM Funcionario  
WHERE DAY(dataNascto) = 30
```

Capítulo 5

1.

```
SELECT nome, salario+PI()  
FROM Funcionario
```

2.

```
SELECT SQRT(DAY(dataNascto))  
FROM Funcionario  
WHERE cidade='Valinhos'
```

3.

```
SELECT LOG(MONTH(dataNascto))  
FROM Funcionario  
WHERE YEAR(dataNascto)=1990
```

4.

```
SELECT nome, DAY(dataNascto)  
FROM Funcionario  
WHERE POWER(DAY(dataNascto),3) >= 1000
```

5.

```
SELECT ROUND((salario * 1.156),0)  
FROM Funcionario  
WHERE salario > 1000
```

6.

```
SELECT ABS(1500 - salario)  
FROM Funcionario
```

7.

```
SELECT idFuncionario, SQRT(idFuncionario)  
FROM Funcionario  
WHERE dataNascto < '1989-04-01'
```

8.

```
SELECT nome, ROUND((salario * 0.65),1)  
FROM Funcionario
```

9.

```
SELECT LOG(idFuncionario)
FROM Funcionario
```

10.

```
SELECT SQRT(idFuncionario)
FROM Funcionario
```

11.

```
SELECT POWER(idFuncionario,2)
FROM Funcionario
```

12.

```
SELECT ABS(idFuncionario - 10) AS valor_abs
FROM Funcionario
ORDER BY valor_abs DESC
```

Capítulo 6

1.

```
SELECT UPPER(nome)
FROM Funcionario
```

2.

```
SELECT DISTINCT DATENAME(MONTH,dataNascto),
LEN(DATENAME(MONTH,dataNascto))
FROM Funcionario
```

3.

```
SELECT REPLACE(nome,' ','-')
FROM Funcionario
```

4.

```
SELECT LEFT(nome,3), RIGHT(nome,3)
FROM Funcionario
```

5.

```
SELECT SQRT(LEN(nome))
FROM Funcionario
```

6.

```
SELECT DISTINCT SUBSTRING(cidade,3,5)
FROM Funcionario
```

7.

```
SELECT DISTINCT SUBSTRING(cidade,3,5)
FROM Funcionario
```


8.

```
SELECT CHAR(idFuncionario)
FROM Funcionario
WHERE cidade='Campinas'
```

9.

```
SELECT ASCII(nome)
FROM Funcionario
WHERE DAY(dataNascto) > 20
```

10.

```
SELECT RTRIM(LEFT(cidade,4))
FROM Funcionario
```

11.

```
SELECT LTRIM(RIGHT(cidade,6))
FROM Funcionario
```

12.

```
SELECT DISTINCT LOWER(cidade)
FROM Funcionario
```

Capítulo 7

1.

```
SELECT MAX(salario), MIN(salario)
FROM Funcionario
WHERE estado='SP'
```

2.

```
SELECT SUM(salario)
FROM Funcionario
WHERE nome LIKE '%Cunha'
```

3.

```
SELECT AVG(salario)
FROM Funcionario
WHERE email LIKE '%yahoo%'
```

4.

```
SELECT COUNT(*)
FROM Funcionario
WHERE email LIKE '%br'
```

5.

```
SELECT MIN(dataNascto)
FROM Funcionario
```

6.

```
SELECT MAX(dataNascto) AS Maior_Nascimento
FROM Funcionario
```

7.

```
SELECT COUNT(*) AS Quantidade_de_Valinhos
FROM Funcionario
WHERE cidade='Valinhos'
```

8.

```
SELECT SUM(salario)
FROM Funcionario
WHERE cidade='Campinas'
```

9.

```
SELECT AVG(salario)
FROM Funcionario
WHERE cidade='São Paulo'
```

10.

```
SELECT SUM(salario)
FROM Funcionario
WHERE nome LIKE 'Ana%'
```

11.

```
SELECT COUNT(*)
FROM Funcionario
WHERE nome LIKE '%Luis%'
```

12.

```
SELECT MIN(salario), MAX(salario)
FROM Funcionario
WHERE endereco LIKE 'Av. São Paulo%'
```

Capítulo 8

1.

```
SELECT cargo, COUNT(*) AS quantidade
FROM Funcionario
GROUP BY cargo
ORDER BY quantidade
```

2.

```
SELECT cargo, COUNT(*)
FROM Funcionario
WHERE NOT cargo IS NULL
GROUP BY cargo
```

3.

```
SELECT cargo, AVG(salario) AS Media_Salarios_Cargo
FROM Funcionario
GROUP BY cargo
```

4.

```
SELECT cargo, SUM(salario)
FROM Funcionario
GROUP BY cargo
HAVING SUM(salario) > 3000
```

5.

```
SELECT cargo, SUM(salario)
FROM Funcionario
WHERE estado='SP'
GROUP BY cargo
```

6.

```
UPDATE Funcionario SET
ativo=1
WHERE (cidade='Jundiaí') OR (cidade='São Paulo')
```

7.

```
UPDATE Funcionario SET
ativo=0
WHERE NOT ((cidade='Jundiaí') OR (cidade='São Paulo'))
```

8.

```
SELECT ativo, COUNT(*)
FROM Funcionario
GROUP BY ativo
```

9.

```
SELECT cidade, SUM(salario)
FROM Funcionario
GROUP BY cidade
```

10.

```
SELECT cidade, AVG(salario)
FROM Funcionario
GROUP BY cidade
HAVING NOT AVG(salario) IS NULL
```

11.

```
SELECT cargo, SUM(salario), AVG(salario)
FROM Funcionario
GROUP BY cargo
HAVING SUM(salario) < 5000
```

12.

```
SELECT cidade, cargo, SUM(salario), AVG(salario)
FROM Funcionario
GROUP BY cidade, cargo
```

Capítulo 9

1.

```
SELECT TOP 4 nome  
FROM Funcionario
```

2.

```
SELECT TOP 2 *  
FROM Funcionario  
WHERE cidade='Valinhos'
```

3.

```
SELECT TOP 1 nome, dataNascto  
FROM Funcionario  
ORDER BY dataNascto ASC
```

4.

```
SELECT TOP 2 cidade, COUNT(*)  
FROM Funcionario  
GROUP BY cidade
```

5.

```
SELECT TOP 2 cargo, COUNT(*)  
FROM Funcionario  
GROUP BY cargo
```

6.

```
SELECT TOP 30 PERCENT *  
FROM Funcionario
```

7.

```
SELECT TOP 6 nome, email  
FROM Funcionario
```

8.

```
SELECT TOP 70 PERCENT idFuncionario, cargo, ativo  
FROM Funcionario
```

9.

```
SELECT TOP 1 idFuncionario, salario  
FROM Funcionario  
WHERE NOT salario IS NULL  
ORDER BY salario ASC
```

10.

```
SELECT TOP 1 nome, salario  
FROM Funcionario  
ORDER BY salario DESC
```

11.

```
SELECT TOP 1 nome, endereco
FROM Funcionario
```

12.

```
SELECT TOP 90 PERCENT *
FROM Funcionario
```

13.

```
SELECT TOP 1 *
FROM Funcionario
WHERE cidade='São Paulo'
```

14.

```
SELECT TOP 20 PERCENT nome, endereco, cidade, estado
FROM Funcionario
```

15.

```
SELECT TOP 2 *
FROM Funcionario
WHERE YEAR(dataNascto) = 1988
```

Capítulo 10

1.

```
CREATE DATABASE Compras
```

2.

```
USE Compras
```

3.

```
CREATE TABLE Cliente(
    IdCliente int identity primary key,
    Nome varchar(50) NOT NULL,
    Endereco varchar(50) NOT NULL,
    Cidade varchar(50) NOT NULL,
    Estado char(2) NOT NULL
)
```

```
CREATE TABLE Produto(
    IdProduto INT IDENTITY PRIMARY KEY,
    Descricao VARCHAR(50) NOT NULL,
    Preço DECIMAL(5,2) NOT NULL,
    Qtde INT NOT NULL
)
```

```
CREATE TABLE Compram (
    IdCompra INT IDENTITY(1000,2),
    IdCliente INT,
    IdProduto INT,
```

```
Data DATETIME NOT NULL,
Qtde INT,
Valor DECIMAL(5,2),
PRIMARY KEY(IdCompra,IdCliente,IdProduto)
)
```

4.

```
ALTER TABLE Cliente
ADD sexo CHAR(1) NOT NULL
```

5.

```
INSERT INTO Cliente
(Nome,Endereco,Cidade,Estado,Sexo)
VALUES
('José de Oliveira','Av. Jatobá,34','Jundiaí','SP','F')
```

```
INSERT INTO Cliente
(Nome,Endereco,Cidade,Estado,Sexo)
VALUES
('Maria da Silva','Av. Presidente,12','Itatiba','MG','F')
```

```
INSERT INTO Cliente
(Nome,Endereco,Cidade,Estado,Sexo)
VALUES
('Antonio Carlos','R. Florença,5','Jundiaí','SP','M')
```

```
INSERT INTO Cliente
(Nome,Endereco,Cidade,Estado,Sexo)
VALUES
('Luisa de Souza','Av. Jatobá,45','Jundiaí','MG','F')
```

```
INSERT INTO Cliente
(Nome,Endereco,Cidade,Estado,Sexo)
VALUES
('Calos de Souza','Av. Jatobá,45','Jundiaí','SP','M')
```

6.

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Lápis',1.50,20)
```

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Borracha',1.00,15)
```

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Caneta',1.75,35)
```

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Compasso',5.20,10)
```

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Régua',0.75,16)
```

```
INSERT INTO Produto
(Descricao,Preco,Qtde)
VALUES
('Papel Sulfite',10.50,5)
```

7.

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(1,1,'2010-12-01',2,1.50)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(2,1,'2010-12-03',5,1.50)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(1,3,'2011-01-05',13,1.75)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(1,4,'2011-01-11',1,5.20)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(3,2,'2011-03-16',7,1.00)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(4,5,'2011-05-21',10,0.75)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(2,6,'2011-06-07',2,10.50)
```

```
INSERT INTO Compras
(IdCliente,IdProduto,Data,Qtde,Valor)
VALUES
(5,3,'2011-06-07',2,1.75)
```

8.

```
UPDATE Cliente  
SET Estado = 'SP'
```

9.

```
SELECT Nome, Estado  
FROM Cliente
```

10.

```
UPDATE Cliente  
SET Sexo = 'M'  
WHERE Nome = 'José de Oliveira'
```

11.

```
SELECT Descricao, Preço  
FROM Produto
```

12.

```
DELETE FROM Produto  
WHERE Descricao = 'Papel Sulfite'
```

13.

```
UPDATE Produto  
SET Qtde = 15  
WHERE Descricao = 'Lápis'
```

14.

```
SELECT TOP 2 LOWER(Descricao)  
FROM Produto
```

15.

```
SELECT SUM(Valor)  
FROM Compram  
WHERE IdProduto = 1
```

16.

```
SELECT AVG(valor)  
FROM Compram  
WHERE IdCliente = 1
```

17.

```
SELECT Nome  
FROM Cliente  
WHERE Cidade = 'Jundiaí'
```

18.

```
SELECT IdCliente, UPPER(Nome)  
FROM Cliente  
WHERE Nome LIKE '%Carlos%'
```


19.

```
SELECT Descricao, Preco, Qtde  
FROM Produto  
WHERE Preco > 1 AND Qtde >= 10
```

20.

```
SELECT *  
FROM Cliente  
ORDER BY Nome
```

21.

```
SELECT DISTINCT cidade, COUNT(*)  
FROM Cliente  
GROUP BY Cidade  
ORDER BY COUNT(*)
```

22.

```
SELECT SUM(Preco) AS SomaPreco, AVG(Preco) AS MediaPreco  
FROM Produto
```

23.

```
SELECT MAX(Preco) AS PrecoMaisCaro, MIN(Preco) AS PrecoMaisBarato  
FROM Produto
```

24.

```
SELECT SUM(Valor)  
FROM Compram  
WHERE YEAR(Data) = '2010'
```

25.

```
SELECT TOP 1 Valor  
FROM Compram  
WHERE YEAR(Data) = '2011'  
ORDER BY Data
```

26.

```
SELECT Nome  
FROM Cliente  
WHERE Sexo = 'F'
```

27.

```
SELECT *  
FROM Compram  
WHERE DAY(Data) IN ('1','11')
```

28.

```
SELECT Descricao, Preco, (Preco + (Preco*0.1)) AS  
PrecoAcrescido10porCento  
FROM Produto
```

29.

```
SELECT IdCliente, COUNT(*) AS QuantidadeCompra
FROM Compras
GROUP BY IdCliente
```

30.

```
UPDATE Produto
SET Preço = (Preço - (Preço*0.1))
WHERE Qtde < 15
```

31.

```
SELECT IdProduto, DAY(Data)
FROM Compras
```

32.

```
SELECT DISTINCT Sexo, COUNT(*)
FROM Cliente
GROUP BY Sexo
```

33.

```
DELETE FROM Compras
WHERE IdCompra = 1000
```

34.

```
SELECT Descricao, POWER(Qtde,2) AS QtdeAoQuadrado
FROM Produto
WHERE Qtde > 15 AND Qtde < 25
```

35.

```
SELECT SQRT(Qtde) AS RaizDaQuantidade
FROM Produto
WHERE Descricao LIKE 'C%'
```

36.

```
SELECT Nome
FROM Cliente
WHERE Endereço LIKE 'Av. Jatobá%'
```

37.

```
SELECT Nome, LEN(Nome) AS QuantidadeDeCaractere
FROM Cliente
```

38.

```
SELECT IdCompra, Valor, (Valor-(Valor*0.2)) AS
Valor20PorCentoDesconto
FROM Compras
WHERE IdCliente = 2
```

39.

```
SELECT YEAR(Data), COUNT(*)
FROM Compras
GROUP BY YEAR(Data)
```

40.

```
SELECT IdCompra, DAY(Data) AS DiaDaCompra, DATENAME(MONTH,Data)
AS MesDaCompra, YEAR(Data) AS AnoDasCompras
FROM Compram
```

41.

```
SELECT IdProduto, SUM(Valor*Qtde)
FROM Compram
GROUP BY IdProduto
HAVING SUM(Valor*Qtde) > 7
```

42.

```
DELETE FROM Compram
WHERE IdCliente BETWEEN 3 AND 5
```

43.

```
DROP TABLE Produto
```

44.

```
USE MASTER
DROP DATABASE Compras
```

Capítulo 11



O leitor poderá executar o arquivo **empresa.sql** disponível no site da Editora Érica, que criará as tabelas e colocará dados nelas.

1.

```
CREATE DATABASE Empresa
```

2.

```
USE Empresa
```

3.

```
CREATE TABLE Fornecedores (
    CodFor INT IDENTITY NOT NULL,
    Empresa VARCHAR(40),
    Contato VARCHAR(30),
    Cargo VARCHAR(30),
    Endereco VARCHAR(60),
    Cidade VARCHAR(15),
    CEP VARCHAR(10),
    Pais VARCHAR(15),
    PRIMARY KEY (CodFor)
)

CREATE TABLE Categorias (
    CodCategoria INT IDENTITY NOT NULL,
    Descr VARCHAR(15),
    PRIMARY KEY (CodCategoria)
)
```

```

CREATE TABLE Clientes (
    CodCli CHAR(5) NOT NULL,
    Nome VARCHAR(40) NOT NULL,
    Contato VARCHAR(30) NOT NULL,
    Cargo VARCHAR(30) NOT NULL,
    Endereco VARCHAR(60) NOT NULL,
    Cidade VARCHAR(15) NOT NULL,
    Regiao VARCHAR(15) NOT NULL,
    CEP VARCHAR(10) NOT NULL,
    Pais VARCHAR(15) NOT NULL,
    Telefone VARCHAR(24) NOT NULL,
    Fax VARCHAR(24) NOT NULL,
    PRIMARY KEY(CodCli)
)

CREATE TABLE Funcionarios(
    CodFun INT IDENTITY NOT NULL,
    Sobrenome VARCHAR(20),
    Nome VARCHAR(10),
    Cargo VARCHAR(30),
    DataNasc DATE,
    Endereco VARCHAR(60),
    Cidade VARCHAR(15),
    CEP VARCHAR(10),
    Pais VARCHAR(15),
    Fone VARCHAR(24),
    Salario MONEY DEFAULT 0.0,
    PRIMARY KEY (CodFun)
)

CREATE TABLE Produtos(
    CodProd INT IDENTITY NOT NULL,
    Descr VARCHAR(40),
    CodFor INT,
    CodCategoria INT,
    Preco MONEY DEFAULT 0.0,
    Unidades SMALLINT DEFAULT 0,
    Descontinuado BIT,
    PRIMARY KEY (CodProd),
    FOREIGN KEY (CodCategoria) REFERENCES Categorias(CodCategoria)
ON DELETE CASCADE,
    FOREIGN KEY (CodFor) REFERENCES Fornecedores(CodFor) ON DELETE
CASCADE
)

CREATE TABLE Pedidos(
    NumPed INT NOT NULL,
    CodCli CHAR(5),
    CodFun INT DEFAULT 0,
    DataPed DATE,
    DataEntrega DATE,
    Frete MONEY DEFAULT 0.0,
    PRIMARY KEY (NumPed),
    FOREIGN KEY (CodCli) REFERENCES Clientes(CodCli) ON DELETE
CASCADE,

```

```

    FOREIGN KEY (CodFun) REFERENCES Funcionarios(CodFun) ON DELETE
    CASCADE
)

CREATE TABLE DetalhesPed(
    NumPed INT ,
    CodProd INT ,
    Preco MONEY,
    Qtde SMALLINT ,
    Desconto FLOAT,
    PRIMARY KEY (NumPed, CodProd),
    FOREIGN KEY (NumPed) REFERENCES Pedidos(NumPed) ON DELETE
    CASCADE,
    FOREIGN KEY (CodProd) REFERENCES Produtos(CodProd) ON DELETE
    CASCADE
)

```

Capítulo 12

1.

```

SELECT TOP 1 Descr, Preco
FROM Produtos
ORDER BY Preco DESC

```

2.

```

SELECT TOP 5 NumPed, DataPed
FROM Pedidos
ORDER BY Frete

```

3.

```

SELECT Nome, Cargo
FROM Clientes
UNION
SELECT Nome, Cargo
FROM Funcionarios
WHERE Pais = 'Reino Unido'

```

4.

```

SELECT TOP 3 Nome, Sobrenome, Cargo, Salario
FROM Funcionarios
ORDER BY Salario DESC

```

5.

```

SELECT TOP 1 Nome, Sobrenome
FROM Funcionarios
ORDER BY DataNasc DESC

```

6.

```

SELECT TOP 5 *
FROM Pedidos
ORDER BY DataPed

```

7.

```
SELECT TOP 6 *  
FROM Pedidos  
WHERE YEAR(DataPed) = 1996
```

8.

```
SELECT Nome, Cargo  
FROM Funcionarios  
WHERE Pais = 'EUA'  
UNION  
SELECT Contato, Cargo  
FROM Fornecedores  
WHERE Pais = 'EUA'
```

9.

```
SELECT Nome, Contato, Pais  
FROM Clientes  
WHERE Pais = 'Brasil'  
UNION  
SELECT Nome, Contato, Pais  
FROM Clientes  
WHERE Pais = 'Alemanha'
```

10.

```
SELECT Nome, Contato, Cidade  
FROM Clientes  
WHERE Cidade = 'Madrid'  
UNION  
SELECT Nome, Contato, Cidade  
FROM Clientes  
WHERE Cidade = 'Paris'
```

11.

```
SELECT Descr, Preco  
FROM Produtos  
WHERE CodCategoria = 2  
UNION  
SELECT Descr, Preco  
FROM Produtos  
WHERE CodCategoria = 4
```

12.

```
SELECT Nome, Cargo, Pais  
FROM Funcionarios  
WHERE Pais = 'Reino Unido'  
UNION  
SELECT Contato, Cargo, Pais  
FROM Fornecedores  
WHERE Pais = 'França'
```

Capítulo 13

1.

```
SELECT Pais, COUNT(*) AS QtdeClientes
FROM Clientes
GROUP BY Pais
```

2.

```
SELECT SUM(Preco) AS Soma, AVG(Preco) AS Media, MAX(Preco) AS
MaiorPreco, MIN(Preco) AS MenorPreco
FROM Produtos
```

3.

```
SELECT C.Pais, COUNT(P.NumPed) AS QtdePedidos
FROM Clientes C, Pedidos P
WHERE C.CodCli = P.CodCli
GROUP BY C.Pais
ORDER BY COUNT(P.NumPed) DESC
```

4.

```
SELECT Nome, Sobrenome, Cargo, Salario, (Salario*1.10) AS
Salario_Novo
FROM Funcionarios
```

5.

```
SELECT SUM(DP.Preco) AS SomadosPrecos
FROM DetalhesPed DP, Pedidos P
WHERE (DP.NumPed = P.NumPed) AND (YEAR(P.DataEntrega) = 1997) AND
(MONTH(P.DataEntrega) = 5)
```

6.

```
SELECT C.CodCli, C.Nome, C.Pais
FROM Clientes C, Pedidos P
WHERE (C.CodCli = P.CodCli) AND (YEAR(P.DataPed) = 1997) AND
(MONTH(P.DataPed) = 09)
ORDER BY (C.Pais)
```

7.

```
SELECT F.Nome, P.*
FROM Funcionarios F, Pedidos P
WHERE (F.CodFun = P.CodFun) AND (F.Nome LIKE 'A%')
```

8.

```
SELECT P.Descr, P.Unidades
FROM Produtos P, Fornecedores F
WHERE (P.CodFor = F.CodFor) AND (F.Empresa = 'Exotic Liquids')
```

9.

```
SELECT DISTINCT P.Descr
FROM Produtos P, DetalhesPed DP, Pedidos Pd
WHERE (P.CodProd = DP.CodProd) AND (DP.NumPed = Pd.NumPed)
AND (DP.Qtde >= 50) AND (YEAR(Pd.DataPed) = 1997)
ORDER BY P.Descr
```

10.

```
SELECT DISTINCT C.Descr, P.Descr
FROM Categorias C, Produtos P, DetalhesPed DP, Pedidos Pd
WHERE (C.CodCategoria = P.CodCategoria) AND (P.CodProd =
DP.CodProd) AND (DP.NumPed = Pd.NumPed) AND (DP.Qtde >= 50) AND
(YEAR(Pd.DataEntrega) = 1997)
ORDER BY C.Descr DESC
```

Capítulo 14

1.

```
SELECT C.*
FROM Clientes C INNER JOIN Pedidos P
ON C.Codcli = P.Codcli
AND YEAR(P.DataPed) = 1996
```

2.

```
SELECT F.Nome
from Funcionarios F INNER JOIN Pedidos P ON F.Codfun = P.Codfun
INNER JOIN Clientes C ON P.Codcli = C.Codcli
AND C.Nome = 'Around the horn'
```

3.

```
SELECT P.*
FROM Pedidos P INNER JOIN Clientes C ON P.Codcli = C.Codcli
AND C.Nome = 'Comércio Mineiro'
```

4.

```
SELECT F.*
FROM Funcionarios F INNER JOIN Pedidos P ON F.CodFun = P.CodFun
AND YEAR(P.DataPed) = 1996 AND MONTH(P.DataPed)= 9
```

5.

```
SELECT P.*, C.Descr
FROM Produtos P INNER JOIN categoria C ON P.CodCategoria =
C.CodCategoria
AND C.Descr = 'laticínios'
```

6.

```
SELECT P.*, Pd.NumPed AS NumeroPedido
FROM Produtos P INNER JOIN DetalhesPed DP ON P.CodProd =
DP.CodProd
INNER JOIN Pedidos Pd ON DP.NumPed = Pd.NumPed
AND Pd.DataPed = '1996-07-08'
```

7.

```
SELECT F.Nome, P.NumPed AS NumeroPedido
FROM Funcionarios F INNER JOIN Pedidos P ON F.CodFun = P.CodFun
AND P.DataPed = '1997-05-01'
```


8.

```
SELECT F.Nome, P.*
FROM Funcionarios F INNER JOIN Pedidos P ON F.CodFun = P.CodFun
AND F.Salario > 10000
```

9.

```
SELECT P.NumPed, C.Nome
FROM Pedidos P INNER JOIN Clientes C ON P.CodCli = C.CodCli
AND MONTH(P.DataPed) = 5 and YEAR(P.DataPed) = 1997
```

10.

```
SELECT DISTINCT C.Descr, P.Descr
FROM Categorias C INNER JOIN Produtos P ON C.CodCategoria =
P.CodCategoria
INNER JOIN DetalhesPed DP ON P.CodProd = DP.CodProd
AND DP.Qtde <= 10
INNER JOIN Pedidos Pd ON DP.NumPed = Pd.NumPed
AND YEAR(Pd.DataPed) = 1998
ORDER BY C.Descr DESC
```

11.

```
SELECT DP.*
FROM DetalhesPed DP INNER JOIN Pedidos P ON DP.NumPed = P.NumPed
AND YEAR(P.DataEntrega) = 1997
```

12.

```
SELECT DISTINCT C.Descr, P.Descr
FROM Categorias C CROSS JOIN Produtos P
```

Capítulo 15

1.

```
SELECT *
FROM Pedidos
WHERE CodCli IN (SELECT CodCli
FROM Clientes
WHERE Pais = 'Alemanha')
```

2.

```
SELECT *
FROM Produtos
WHERE CodCategoria IN (SELECT CodCategoria
FROM Categorias
WHERE Descr = 'Condimentos')
```

3.

```
SELECT Descr
FROM Produtos
WHERE CodFor NOT IN (SELECT CodFor
FROM Fornecedores
WHERE Pais = 'EUA')
```

4.

```

SELECT Descr
FROM Produtos
WHERE CodProd IN (SELECT CodProd
FROM DetalhesPed
WHERE NumPed IN (SELECT NumPed
FROM Pedidos
WHERE YEAR(Dataped) <> 1997
AND MONTH(Dataped) <> 3 ))

```

5.

```

SELECT CodProd, Descr, Preco
FROM Produtos
WHERE Preco = (SELECT MIN(Preco)
FROM Produtos)

```

6.

```

SELECT Nome, Salario
FROM Funcionarios
WHERE Salario = (SELECT MAX(Salario)
FROM Funcionarios)

```

7.

```

SELECT Nome, Salario
FROM Funcionarios
WHERE Salario = (SELECT MAX(Salario)
FROM Funcionarios)
OR Salario = (SELECT MIN(Salario)
FROM Funcionarios)
ORDER BY Salario

```

8.

```

SELECT CodProd, Descr, Preco
FROM Produtos
WHERE Preco > (SELECT AVG(Preco)
FROM Produtos)

```

9.

```

SELECT Nome, Sobrenome, Cargo, Salario
FROM Funcionarios
WHERE Cargo = 'Representante de Vendas'
AND Salario < ALL (SELECT Salario
FROM Funcionarios
WHERE Cargo LIKE 'gerente%'
OR Cargo LIKE 'coordenador%')

```

10.

```

SELECT Nome, Sobrenome, Cargo, Salario
FROM Funcionarios
WHERE Cargo LIKE 'Coordenador%'
AND Salario > ANY (SELECT Salario
FROM Funcionarios
WHERE Cargo = 'Representante de Vendas')

```

11.

```
SELECT F.Nome, P.*
FROM Funcionarios F, Pedidos P
WHERE F.CodFun = P.CodFun AND P.Frete > (SELECT AVG(Frete)
FROM Pedidos)
```

12.

```
SELECT *
FROM Produtos
WHERE Preco < ALL (SELECT Preco
FROM Produtos
WHERE CodCategoria IN (SELECT CodCategoria
FROM Categorias
WHERE Descr = 'Confeitos'))
```

Capítulo 16

1.

```
CREATE VIEW Preco_Baixo AS
SELECT CodProd, Descr, Preco
FROM Produtos
WHERE Preco < (SELECT AVG(Preco)
FROM Produtos)
```

2.

```
SELECT *
FROM Preco_Baixo
WHERE Descr LIKE 'C%'
```

3.

```
CREATE VIEW Funcionarios_Cargo AS
SELECT Cargo, COUNT(*) AS FuncionariosPorCargo
FROM Funcionarios
GROUP BY Cargo
```

4.

```
SELECT Cargo
FROM Funcionarios_Cargo
WHERE FuncionariosPorCargo = (SELECT MAX(FuncionariosPorCargo)
FROM Funcionarios_Cargo)
```

5.

```
CREATE VIEW Produtos_Categoria AS
    SELECT P.Descr AS DescrProduto, C.Descr AS DescrCategoria
    FROM Produtos P, Categorias C
WHERE P.CodCategoria = C.CodCategoria
```

6.

```
SELECT DescrCategoria, COUNT(*) AS
QuantidadeDeProdutosPorCategoria
FROM Produtos_Categoria
GROUP BY DescrCategoria
```

7.

```
CREATE VIEW Clientes_Resumo AS
    SELECT CodCli, Nome, Contato, Cargo, Pais
FROM Clientes
```

8.

```
CREATE VIEW Pedidos_Resumo_abr97 AS
    SELECT NumPed, CodCli, DataEntrega
FROM Pedidos
WHERE YEAR(DataEntrega) = 1997 AND MONTH(DataEntrega) = 4
```

9.

```
SELECT C.*
FROM Clientes_Resumo C INNER JOIN Pedidos_Resumo_abr97 P
ON C.CodCli = P.CodCli
```

10.

```
CREATE VIEW Clientes_Resumo_W AS
    SELECT *
FROM Clientes_Resumo
WHERE Nome LIKE 'W%'
```

11.

```
DROP VIEW Preco_Baixo
DROP VIEW Funcionarios_Cargo
DROP VIEW Produtos_Categoria
DROP VIEW Clientes_Resumo
DROP VIEW Pedidos_Resumo_abr97
DROP VIEW Clientes_Resumo_W
```

Capítulo 17

1.

```
DECLARE @i INT;
SET @i = 100
WHILE @i >= 0
BEGIN
    PRINT @i;
    SET @i = @i - 2;
END;
```

2.

```
DECLARE @a INT, @b INT, @c INT;
SET @a = 1;
SET @b = 2;
SET @c = 3;
IF (@a > @b) AND (@b > @c)
BEGIN
    PRINT @a;
    PRINT @b;
    PRINT @c;
END
```

```

ELSE IF (@a>@c) AND (@c>@b)
BEGIN
    PRINT @a;
    PRINT @c;
    PRINT @b;
END
ELSE IF (@b>@a) AND (@a>@c)
BEGIN
    PRINT @b;
    PRINT @a;
    PRINT @c;
END
ELSE IF (@b>@c) AND (@c>@a)
BEGIN
    PRINT @b;
    PRINT @c;
    PRINT @a;
END
ELSE IF (@c>@a) AND (@a>@b)
BEGIN
    PRINT @c;
    PRINT @a;
    PRINT @b;
END
ELSE IF (@c>@b) AND (@b>@a)
BEGIN
    PRINT @c;
    PRINT @b;
    PRINT @a;
END
END

```

3.

```

DECLARE @num1 INT;
DECLARE @num2 INT;
SET @num1 = ROUND(10 * RAND(),0,1)
SET @num2 = ROUND(10 * RAND(),0,1)
IF ((@num1%2) = 0)
    PRINT CONVERT(CHAR(2),@num1) + 'O Numero 1 é Par!';
ELSE
    PRINT CONVERT(CHAR(2),@num1) + 'O Numero 1 é Impar!'
IF ((@num2%2) = 0)
    PRINT CONVERT(CHAR(2),@num2) + 'O Numero 2 é Par!';
ELSE
    PRINT CONVERT(CHAR(2),@num2) + 'O Numero 2 é Impar!'

```

4.

```

SELECT QTDE,
    CASE
        WHEN QTDE < 10 THEN 'DESCONTO = 0'
        WHEN QTDE >= 10 AND QTDE < 30 THEN 'DESCONTO = 3'
        WHEN QTDE >= 30 AND QTDE < 50 THEN 'DESCONTO = 5'
        WHEN QTDE >= 50 AND QTDE < 70 THEN 'DESCONTO = 7'
        ELSE 'DESCONTO = 9'
    END AS DescontoPedido
FROM DetalhesPed

```

OU

```

SELECT QTDE,
       CASE
         WHEN QTDE < 10 THEN 0
         WHEN QTDE < 30 THEN 3
         WHEN QTDE < 50 THEN 5
         WHEN QTDE < 70 THEN 7
         ELSE 9
       END AS DescontoPedido
FROM DetalhesPed

```

5.

```

DECLARE @i INT;
DECLARE @lidos INT;
DECLARE @num INT;
SET @i = 0;
SET @lidos = ROUND(10 * RAND(),0,1);
PRINT 'Quantidade de números lidos: ' + CONVERT(CHAR(2),@lidos)
WHILE(@i < @lidos)
BEGIN
  SET @num = ROUND(100 * RAND(),0,1);
  IF (@num%2)=0
    PRINT(@num);
    SET @i += 1;
END;

```

6.

```

DECLARE @i INT;
SET @i = 0;
WHILE(@i <= 1000)
BEGIN
  IF ((@i%10)=0)
    PRINT(@i);
    SET @i = @i + 1;
END;

```

7.

```

SELECT NOME,PAIS,
       CASE (PAIS)
         WHEN 'Brasil' THEN 'Exportação'
         ELSE 'Importação'
       END AS Situacao
FROM CLIENTES

```

8.

```

DECLARE @NOME VARCHAR(50)
SET @NOME = 'NATHAN CIRILLO E SILVA'
PRINT(@NOME)
PRINT(LEN(@NOME))

```

Capítulo 18

1.

```
CREATE PROCEDURE Busca_Func @CodFun INT
AS
    SELECT Nome, Sobrenome, Cargo
FROM Funcionarios
WHERE CodFun = @CodFun

EXEC Busca_Func 2
```

2.

```
CREATE PROCEDURE Insere_Fornec
@Empresa VARCHAR(40),
@Contato VARCHAR(30),
@Cargo VARCHAR(30),
@Endereco VARCHAR(60),
@Cidade VARCHAR(15),
@Cep VARCHAR(10),
@Pais VARCHAR(15)
AS
INSERT INTO Fornecedores
VALUES
(@Empresa, @Contato, @Cargo, @Endereco, @Cidade, @Cep, @Pais)

EXEC Insere_Fornec 'CPS', 'José Silva', 'Operário', 'Rua 25 de
Março', 'São Paulo', '12345', 'Brasil'
```

3.

```
CREATE PROCEDURE Insere_Detalhes
@NumPed INT,
@CodProd INT,
@Preco MONEY,
@Qtde SMALLINT,
@Desconto FLOAT
AS
    DECLARE @ContagemNumPed INT
    DECLARE @ContagemCodProd INT
    DECLARE @VerificaNumPedECodProd INT

SELECT @ContagemNumPed = COUNT(*)
FROM Pedidos
WHERE NumPed = @NumPed

    SELECT @ContagemCodProd = COUNT(*)
FROM Produtos
WHERE CodProd = @CodProd

    SELECT @VerificaNumPedECodProd = COUNT(*)
FROM DetalhesPed
WHERE NumPed = @NumPed AND CodProd = @CodProd
```

```

IF (@ContagemCodProd = 0)
    PRINT 'Codigo do Produto não Cadastrado!';
ELSE IF (@ContagemNumPed = 0)
    PRINT 'Numero do Pedido não Cadastrado!';
ELSE
    BEGIN
        IF (@VerificaNumPedECodProd <> 0)
            PRINT 'Produto já cadastrado para esse pedido'
        ELSE
            BEGIN
                INSERT INTO DetalhesPed
                VALUES
                (@NumPed, @CodProd, @Preco, @Qtde, @Desconto)
                PRINT 'Dados Cadastrados com Sucesso!'
            END;
        END;
    END;

EXEC InseRe_Detalhes 10868, 199, 1, 1, 1 --Codigo do Produto não
Cadastrado!

EXEC InseRe_Detalhes 100, 11, 1, 1, 1 --Numero do Pedido não
Cadastrado!

EXEC InseRe_Detalhes 10867, 53, 1, 1, 1 --Produto já cadastrado
para esse pedido

EXEC InseRe_Detalhes 10867, 55, 1, 1, 1 --Dados Cadastrados com
Sucesso!

```

4.

```

CREATE PROCEDURE Aumenta_Preco
@CodProd INT,
@PercAum DECIMAL(5,2)
AS
    IF (@CodProd = 0)
        UPDATE Produtos
    SET Preco = Preco + (Preco * @PercAum);
    ELSE
        UPDATE Produtos
    SET Preco = Preco + (Preco * @PercAum)
    WHERE CodProd = @CodProd

EXEC Aumenta_Preco 43, 0.0 --um produtos

EXEC Aumenta_Preco 0, 0.0 --todos

```

5.

```

CREATE PROCEDURE Exclui_Produto
@CodProd INT
AS
    IF (NOT EXISTS (SELECT CodProd
FROM Produtos
WHERE CodProd = @CodProd))

```



```

    PRINT 'Não há produto cadastrado com esse código para ser
excluído!';
ELSE
BEGIN
    DELETE FROM Produtos
WHERE CodProd = @CodProd
    PRINT 'Dados Excluídos com Sucesso!'
END;

```

```

EXEC Exclui_Produto 444 --Não há produto cadastrado com esse
código para ser excluído!

```

```

EXEC Exclui_Produto 77 --Dados Excluídos com Sucesso!

```

6.

```

CREATE PROCEDURE Altera_Produto
@CodProd INT,
@DescrProd VARCHAR(40)
AS
    IF (NOT EXISTS (SELECT CodProd
FROM Produtos
WHERE CodProd = @CodProd))
        PRINT 'Não há produto cadastrado com esse código para ser
alterado!';
    ELSE
        BEGIN
            UPDATE Produtos
SET Descr=@DescrProd
WHERE CodProd = @CodProd
            PRINT 'Produto alterado com Sucesso!'
        END;

```

```

EXEC Altera_Produto 444, 'chocolate' --Não há produto cadastrado
com esse código para ser alterado!

```

```

EXEC Altera_Produto 76, 'chocolate'

```

7.

```

CREATE PROC Exclui_Pedido
@numPed INT
AS
    IF (NOT EXISTS (SELECT NumPed
FROM Pedidos
WHERE NumPed = @numPed))
        PRINT 'Não há pedido cadastrado com esse número!';
    ELSE
        BEGIN
            DELETE FROM DetalhesPed
WHERE NumPed = @numPed

DELETE FROM Pedidos
WHERE NumPed = @numPed

```

```
PRINT 'Pedido Excluído com Sucesso!'
END;
```

```
EXEC Exclui_Pedido 22 --Não há pedido cadastrado com esse número!
```

```
EXEC Exclui_Pedido 10867 --Pedido Excluído com Sucesso!
```

8.

```
CREATE PROC Funcionarios_Cargo
@cargo VARCHAR(30)
AS
    IF (NOT EXISTS (SELECT Cargo
FROM Funcionarios
WHERE Cargo = @cargo))
        PRINT 'Não há funcionários cadastrado com esse cargo!';
    ELSE
        BEGIN
            SELECT *
FROM Funcionarios
WHERE Cargo=@cargo
        END;

EXEC Funcionarios_Cargo 'secretária' --Não há funcionários
cadastrado com esse cargo

EXEC Funcionarios_Cargo 'representante de vendas'
```

9.

```
CREATE PROCEDURE Aumenta_Salario
@CodFun INT,
@PercAum DECIMAL(5,2)
AS
    IF (@CodFun = 0)
        UPDATE Funcionarios
SET Salario = Salario + (Salario * @PercAum);
    ELSE
        UPDATE Funcionarios
SET Salario = Salario + (Salario * @PercAum)
WHERE CodFun = @CodFun

EXEC Aumenta_Salario 5, 0.0 --um funcionário

EXEC Aumenta_Salario 0, 0.0 --todos
```

10.

```
CREATE PROC Clientes_Cidade
@cidade VARCHAR(15)
AS
    IF (NOT EXISTS (SELECT Cidade
FROM Clientes
WHERE Cidade = @cidade))
        PRINT 'Não há clientes cadastrado para essa cidade!';
```

```

ELSE
    BEGIN
        SELECT *
    FROM Clientes
    WHERE Cidade=@cidade
    END;

EXEC Clientes_Cidade 'Jundiaí' --Não há clientes cadastrado para
essa cidade

EXEC Clientes_Cidade 'Madrid'

```

Capítulo 19

1.

```

CREATE FUNCTION RetornaDiaDaSemana(@data DATE)
RETURNS VARCHAR(10)
AS
BEGIN
    RETURN (DATENAME(WEEKDAY, @data));
END;

SELECT dbo.RetornaDiaDaSemana ('2009-05-01')

```

2.

```

ALTER FUNCTION RetornaDiaDaSemanaPortugues(@data DATE)
RETURNS VARCHAR(15)
AS
BEGIN
    DECLARE @diaSemana INT = DATEPART(WEEKDAY, @data);
    DECLARE @diaSemanaPortugues VARCHAR(15);
    SET @diaSemanaPortugues = CASE @diaSemana
                                WHEN 1 THEN 'Domingo'
                                WHEN 2 THEN 'Segunda-feira'
                                WHEN 3 THEN 'Terça-feira'
                                WHEN 4 THEN 'Quarta-feira'
                                WHEN 5 THEN 'Quinta-feira'
                                WHEN 6 THEN 'Sexta-feira'
                                WHEN 7 THEN 'Sábado'
                                END;
    RETURN (@diaSemanaPortugues);
END;

SELECT dbo.RetornaDiaDaSemanaPortugues ('2009-05-01')

```

3.

```

CREATE FUNCTION SomaIntervalo(@valorMinimo INT, @valorMaximo INT)
RETURNS INT
AS
BEGIN
    DECLARE @soma INT = 0;
    WHILE @valorMinimo <= @valorMaximo

```

```

BEGIN
    SET @soma = @soma + @valorMinimo;
    SET @valorMinimo = @valorMinimo + 1;
END;
RETURN @soma;
END

```

```
SELECT dbo.SomaIntervalo (1, 3)
```

```
SELECT dbo.SomaIntervalo (3, 1)
```

4.

```

CREATE FUNCTION DataFormatada(@Dia INT, @Mes INT, @Ano INT)
RETURNS CHAR(10)
AS
BEGIN
    DECLARE @data CHAR(10);
    SET @data = CONVERT(CHAR(2),@Dia) + '/' + CONVERT(CHAR(2),@Mes)
+ '/' + CONVERT(CHAR(4),@Ano);
    RETURN @data;
END

```

```
SELECT dbo.DataFormatada (01, 03, 2011)
```

5.

```

CREATE FUNCTION PrimeiroDiaData(@Data DATE)
RETURNS CHAR(10)
AS
BEGIN
    DECLARE @mes INT, @ano INT;
    DECLARE @primeiroDia CHAR(10);

    SET @mes = MONTH(@Data);
    SET @ano = YEAR(@Data);

    SET @primeiroDia = dbo.DataFormatada(1, @mes, @ano);
    RETURN @primeiroDia;
END

```

```
SELECT dbo.PrimeiroDiaData ('2010-11-23')
```

6.

```

CREATE FUNCTION MediaNotas(@nota1 FLOAT, @nota2 FLOAT, @nota3
FLOAT, @nota4 FLOAT)
RETURNS FLOAT
AS
BEGIN
    RETURN (@nota1 + @nota2 + @nota3 + @nota4) /4;
END

```

```
SELECT dbo.MediaNotas (1, 3, 4, 4)
```

7.

```
CREATE FUNCTION AreaQuadrado(@lado INT)
RETURNS INT
AS
BEGIN
    RETURN @lado * @lado;
END
```

```
SELECT dbo.AreaQuadrado (2)
```

8.

```
CREATE FUNCTION SomaPares(@valorMinimo INT, @valorMaximo INT)
RETURNS INT
AS
BEGIN
    DECLARE @soma INT = 0;
    WHILE @valorMinimo <= @valorMaximo
    BEGIN
        IF (@valorMinimo%2) = 0
            SET @soma = @soma + @valorMinimo;
        SET @valorMinimo = @valorMinimo + 1;
    END;
    RETURN @soma;
END
```

```
SELECT dbo.SomaPares (1, 5)
```

9.

```
ALTER FUNCTION SomaImpares()
RETURNS INT
AS
BEGIN
    DECLARE @i INT = 0, @soma INT = 0;
    WHILE @i <= 50
    BEGIN
        IF (@i%2) = 1
            SET @soma = @soma + @i;
        SET @i = @i + 1;
    END;
    RETURN @soma;
END
```

```
SELECT dbo.SomaImpares()
```

10.

```
CREATE FUNCTION Equacao2Grau(@a INT, @b INT, @c INT)
RETURNS VARCHAR(30)
AS
BEGIN
    DECLARE @resp VARCHAR(30);
    DECLARE @delta FLOAT, @valor1 FLOAT, @valor2 FLOAT;

    SET @delta = POWER(@b,2) - 4 * @a * @c;
    IF @delta < 0
```

```

    SET @resp = 'Não há resposta'
ELSE
    IF @delta = 0
        BEGIN
            SET @valor1 = (- @b + SQRT(@delta)) / 2 * @a;
            SET @resp = 'Resp1= ' + CONVERT (CHAR(5),@valor1)
        END
    ELSE
        BEGIN
            SET @valor1 = (- @b + SQRT(@delta)) / 2 * @a;
            SET @valor2 = (- @b - SQRT(@delta)) / 2 * @a;
            SET @resp = 'Resp1= ' + CONVERT (CHAR(5),@valor1) + ' e
Resp2= ' + CONVERT (CHAR(5),@valor2);
        END
    RETURN @resp;
END

SELECT dbo.Equacao2Grau(3, 1, 2) --Não há resposta

SELECT dbo.Equacao2Grau(-1, 4, -4) --Resp1=2

SELECT dbo.Equacao2Grau(1, -5, 6) --Resp1=3 e Resp2=2

```

Capítulo 20

1.

```

SELECT *
FROM Clientes
WHERE Cidade = 'Buenos Aires'
ORDER BY Nome

```

2.

```

SELECT *
FROM Fornecedores
WHERE Pais = 'Japão'
ORDER BY Cargo DESC

```

3.

```

SELECT Cargo, COUNT(*) AS QtdeFornecedorPorCargo
FROM Fornecedores
GROUP BY Cargo

```

4.

```

SELECT Contato, Cargo, 'Clientes' AS tipo
FROM Clientes
WHERE Pais = 'Alemanha'
UNION
SELECT Contato, Cargo, 'Fornecedores' AS tipo
FROM Fornecedores
WHERE Pais = 'Brasil'
ORDER BY tipo

```

5.

```
SELECT Contato, Cargo
FROM Clientes
WHERE Cargo = 'Agente de vendas'
UNION
SELECT Contato, Cargo
FROM Fornecedores
WHERE Cargo = 'Gerente de Marketing'
```

6.

```
SELECT TOP 3 Descr, Preço
FROM Produtos
ORDER BY Preço DESC
```

7.

```
SELECT TOP 4 Descr, Preço
FROM Produtos
ORDER BY Preço
```

8.

```
SELECT SUM(Preço) AS Soma, AVG(Preço) AS Media, MAX(Preço) AS
MaiorPreço, MIN(Preço) AS MenorPreço
FROM Produtos
WHERE CodCategoria = (SELECT CodCategoria
FROM Categorias
WHERE Descr = 'Frutos do Mar')
```

9.

```
SELECT *, (Preço*1.10) AS PreçoAumentado
FROM Produtos
WHERE CodCategoria = (SELECT CodCategoria
FROM Categorias
WHERE Descr = 'Bebidas')
```

10.

```
SELECT *, (Preço - (Preço*0.2)) AS PreçoDiminuido
FROM Produtos
WHERE CodCategoria = (SELECT CodCategoria
FROM Categorias
WHERE Descr = 'Condimentos')
```

11.

```
SELECT p.*
FROM Pedidos p INNER JOIN Clientes c ON p.CodCli = c.CodCli
AND c.Cargo = 'Proprietário'
```

OU

```
SELECT *
FROM Pedidos
WHERE CodCli IN (SELECT CodCli
FROM Clientes
WHERE Cargo = 'Proprietário')
```

12.

```
SELECT DISTINCT c.*
FROM Clientes C INNER JOIN Pedidos Pd ON C.CodCli = Pd.CodCli
INNER JOIN DetalhesPed DP ON Pd.NumPed = DP.NumPed
INNER JOIN Produtos P ON DP.CodProd = P.CodProd
AND P.Descr = 'Guaraná Fantástica'
```

OU

```
SELECT *
FROM Clientes
WHERE CodCli IN (SELECT CodCli
FROM Pedidos
WHERE NumPed IN (SELECT NumPed
FROM DetalhesPed
WHERE CodProd IN (SELECT CodProd
FROM Produtos
WHERE Descr = 'Guaraná Fantástica'))))
```

13.

```
SELECT *
FROM Clientes
WHERE CodCli IN (SELECT CodCli
FROM Pedidos
WHERE YEAR(DataPed)=1997 AND MONTH(DataPed)=09)
```

14.

```
SELECT *
FROM Produtos
WHERE CodProd IN (SELECT CodProd
FROM DetalhesPed
WHERE NumPed NOT IN (SELECT NumPed
FROM Pedidos
WHERE YEAR(DataPed) = 1996 AND MONTH(DataPed) = 04))
```

15.

```
SELECT Nome
FROM Clientes
WHERE CodCli IN (SELECT CodCli
FROM Pedidos
WHERE NumPed IN (SELECT NumPed
FROM DetalhesPed
WHERE CodProd NOT IN (SELECT CodProd
FROM Produtos
WHERE Descr = 'chocolade')) )
```

16.

```
SELECT Descr
FROM Produtos
WHERE CodProd IN (SELECT CodProd
FROM DetalhesPed
```



```

WHERE NumPed IN (SELECT NumPed
FROM Pedidos
WHERE CodCli IN (SELECT CodCli
FROM Clientes
WHERE Nome LIKE 'A%'))))

```

17.

```

CREATE VIEW Produtos_carnes AS
SELECT P.*
FROM Produtos P JOIN Categorias C ON P.CodCategoria=C.
CodCategoria
AND C.Descr='carnes/aves'

```

18.

```

CREATE VIEW Pedidos_EUA AS
SELECT *
FROM Pedidos
WHERE CodCli IN (SELECT CodCli
FROM Clientes
WHERE Pais='EUA')

```

19.

```

CREATE VIEW Pedidos_entregues_berlin_1997 AS
SELECT P.*
FROM Pedidos P JOIN Clientes C ON P.CodCli=C.CodCli
AND C.Cidade='Berlin' AND YEAR(P.DataEntrega)=1997

```

20.

```

CREATE VIEW Pedidos_descontinuados AS
SELECT Pd.NumPed
FROM Pedidos Pd JOIN DetalhesPed DP ON Pd.NumPed=DP.NumPed
JOIN Produtos Pr ON DP.CodProd=Pr.CodProd
AND Pr.Descontinuado=1

```

21.

```

SELECT TOP 5 NumPed, DataEntrega
FROM Pedidos
ORDER BY Frete

```

22.

```

SELECT Nome, Sobrenome
FROM Funcionarios
WHERE DataNasc = (SELECT MAX(DataNasc)
FROM Funcionarios)

```

23.

```

CREATE VIEW Clientes_n96 AS
SELECT Nome, Contato, Cargo
FROM Clientes
WHERE CodCli NOT IN (SELECT CodCli
FROM Pedidos
WHERE YEAR(DataEntrega)=1996)

```

24.

```
SELECT Cargo, COUNT(Cargo)
FROM Clientes_n96
GROUP BY Cargo
```

25.

```
CREATE VIEW Valores_Pais AS
SELECT Pais, SUM(Preco*Qtde) AS valor
FROM Clientes C JOIN Pedidos P ON C.CodCli=P.CodCli
JOIN DetalhesPed DP ON DP.NumPed=P.NumPed
GROUP BY Pais
```

26.

```
SELECT Pais
FROM Valores_Pais
WHERE valor > (SELECT AVG(valor)
FROM Valores_Pais)
```

27.

```
CREATE PROCEDURE Diminui_Preco
@CodProd INT,
@Perc DECIMAL(5,2)
AS
    UPDATE Produtos
    SET Preco = Preco - (Preco * @Perc)
    WHERE CodProd = @CodProd

EXEC Diminui_Preco 43, 0.0 --um produtos

EXEC Diminui_Preco 90, 0.0 --não existe o produto
```

28.

```
CREATE PROCEDURE Fornecedores_Pais
@Pais VARCHAR(15)
AS
    SELECT *
    FROM Fornecedores
    WHERE Pais=@Pais

EXEC Fornecedores_Pais 'Brasil'
```

29.

```
CREATE PROCEDURE Conta_Categoria
@Categoria VARCHAR(15)
AS
    SELECT COUNT(CodProd)
    FROM Produtos
    WHERE CodCategoria IN (SELECT CodCategoria
    FROM Categorias
    WHERE Descr=@Categoria)

EXEC Conta_Categoria 'Confeitos'
```

30.

```

CREATE PROCEDURE Media_Frete
@DataInicial DATE,
@DataFinal DATE
AS
    SELECT AVG(Frete) AS media, SUM(Frete) AS soma
    FROM Pedidos
WHERE DataEntrega BETWEEN @DataInicial AND @DataFinal

EXEC Media_Frete '1996-01-01', '1996-12-31'

```

31.

```

CREATE FUNCTION SeuNomeParImpar(@SeuNome VARCHAR(10))
RETURNS VARCHAR(5)
AS
BEGIN
    DECLARE @resp VARCHAR(5);
    IF (LEN(@SeuNome) % 2) = 0
        SET @resp = 'Par'
    ELSE
        SET @resp = 'Impar';
    RETURN @resp;
END

```

```
SELECT dbo.SeuNomeParImpar('Luciana') --Impar
```

```
SELECT dbo.SeuNomeParImpar('Caio') --Par
```

32.

```

CREATE TRIGGER AlertaInsercaoFornecedor
ON Fornecedores
FOR INSERT AS
PRINT ('Nova inserção de fornecedor !!!')

INSERT INTO Fornecedores
VALUES ('Editora Erica', 'João', 'Editor', 'Rua de São Paulo',
'São Paulo', '', 'Brasil')

```

33.

```

CREATE TRIGGER Mensagem_Exclui_Pedido
ON Pedidos
FOR DELETE AS
PRINT ('*** Pedido Excluído ***')

DELETE FROM Pedidos where NumPed=0

```