Banco de Dados Olimpíadas

Parte 1

```
-- Create table pais(
--     id serial primary key,
--     nome varchar(50) not null
-- );
-- create table categoria(
--     id serial primary key,
```

```
--     nome varchar(50) not null,
--     sexo char not null
-- );
-- create table modalidade(
--     id serial primary key,
--     nome varchar(50) not null,
--     id_categoria int not null,
--     foreign key (id_categoria) references categoria(id)
-- );
-- create table premiacao(
--     id serial primary key,
--     id_modalidade int not null,
--     id_pais int not null,
--     ouro int,
--   prata int,
--     bronze int,
--     foreign key (id_modalidade) references modalidade(id),
--     foreign key (id_pais) references pais(id),
-- );
```

Parte 2

```
package geral;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Vector;

import javax.swing.ButtonGroup;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JCheckBoxMenuItem;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.ListSelectionModel;

public class Tela extends JFrame {
    public JComboBox cbPais = new JComboBox();
    public JComboBox cbModalidade = new JComboBox();
    public JTable tableMedalha;

    public Tela() throws SQLException {
        super("Tela de Cadastro de Medalhas");
        setSize(485, 680);
        setLayout(null);
        getContentPane().setBackground(new Color(240, 230, 140));
        setResizable(false);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        JLabel jLtitulo = new JLabel("Olimpíadas 2024 - Quadro de Medalhas");
        jLtitulo.setBounds(5, 0, 850, 60);
        jLtitulo.setFont(new Font ("Arial", Font.BOLD, 25));
```

```
        add(jLtitulo);
        JLabel jlNome = new JLabel("País:");
        jlNome.setBounds(10, 70, 850, 60);
        add(jlNome);
        BDUtil bdPais = new BDUtil();
        bdPais.executaBDSelect("Select * from pais","nome");
        cbPais.setModel(new DefaultComboBoxModel(new Vector(bdPais.getLista())));
        cbPais.setBounds(70, 90, 200, 20);
        add(cbPais);
        JLabel jLModal = new JLabel("Modalidade:");
        jLModal.setBounds(10, 110, 850, 60);
        add(jLModal);
        BDUtil bdModal = new BDUtil();
        bdModal.executaBDSelect("select modalidade.id, modalidade.nome, categoria.nome, categoria.sexo,
concat(modalidade.nome, ' - ', categoria.nome, ' - ', categoria.sexo) as modalidade from modalidade, categoria
where categoria.id = modalidade.id_categoria", "modalidade");
        cbModalidade.setModel(new DefaultComboBoxModel(new Vector(bdModal.getLista())));
        cbModalidade.setBounds(80, 130, 250, 20);
        add(cbModalidade);
        JLabel jlMedalhas = new JLabel("Medalhas:");
        jlMedalhas.setBounds(10, 150, 850, 60);
        add(jlMedalhas);
        JRadioButton jRadioButton1 = new JRadioButton();
        JRadioButton jRadioButton2 = new JRadioButton();
        JRadioButton jRadioButton3 = new JRadioButton();
        ButtonGroup g1 = new ButtonGroup();
        jRadioButton1.setText("OURO");
        jRadioButton1.setBackground(new Color(255, 215, 0));
        jRadioButton2.setText("PRATA");
        jRadioButton2.setBackground(new Color(192, 192, 192));
        jRadioButton3.setText("BRONZE");
        jRadioButton3.setForeground(Color.white);
        jRadioButton3.setBackground(new Color(210, 105, 30));
        jRadioButton1.setBounds(200, 170, 70, 20);
        jRadioButton2.setBounds(120, 190, 70, 20);
        jRadioButton3.setBounds(280, 200, 80, 20);
        add(jRadioButton1);
        add(jRadioButton2);
        add(jRadioButton3);
        g1.add(jRadioButton1);
        g1.add(jRadioButton2);
        g1.add(jRadioButton3);
        JLabel jlAros = new JLabel();
        jlAros.setVisible(true);
        jlAros.setBounds(295, 40, 300, 100);
        jlAros.setIcon(new ImageIcon(getClass().getResource("aros.png")));
        add(jlAros);
        JLabel jlPodio = new JLabel();
        jlPodio.setVisible(true);
        jlPodio.setBounds(100, 180, 300, 100);
        jlPodio.setIcon(new ImageIcon(getClass().getResource("podium.jpg")));
        add(jlPodio);
        JButton jbCadastro = new JButton("Cadastrar");
        jbCadastro.setBounds(190, 300, 100, 20);
        jbCadastro.setVisible(true);
        add(jbCadastro);
        setVisible(true);

        Object rowDataMedalhas[][] = new Object[25][25];
        for(int i = 0; i < 5; i++) {
           for(int j = 0; j < rowDataMedalhas[i].length; j++) {
              rowDataMedalhas[i][j] = null;
           }
        }
```

```
Object columnNameMedalhas[] = {"POS", "PAÍS", "OURO", "PRATA", "BRONZE", "TOTAL"};

tableMedalha = new JTable(rowDataMedalhas, columnNameMedalhas);
JScrollPane scrollMedalha = new JScrollPane(tableMedalha);
scrollMedalha.setBounds(10, 340, 450, 300);
tableMedalha.getColumnModel().getColumn(0).setPreferredWidth(50);
tableMedalha.getColumnModel().getColumn(1).setPreferredWidth(130);
tableMedalha.getColumnModel().getColumn(2).setPreferredWidth(60);
tableMedalha.getColumnModel().getColumn(3).setPreferredWidth(60);
tableMedalha.getColumnModel().getColumn(4).setPreferredWidth(60);
tableMedalha.getColumnModel().getColumn(5).setPreferredWidth(70);

for(int i = 0; i < 5; i++) {
   tableMedalha.getColumnModel().getColumn(i).setResizable(false);
}
tableMedalha.getTableHeader().setReorderingAllowed(false);
tableMedalha.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
tableMedalha.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
tableMedalha.setEnabled(true);
add(scrollMedalha);

exibeClassificacao();

BDUtil bd = new BDUtil();

jbCadastro.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent arg0) {
      int paisCombo = 0;
      int idModal = 0;
      if(jRadioButton1.isSelected()) {
         paisCombo = cbPais.getSelectedIndex();
         idModal = cbModalidade.getSelectedIndex();
         bd.sql = "INSERT INTO premiacao(id_modalidade,"
               + "id_pais, ouro, prata, bronze)"
               + "values(?, ?, ?, ?, ?)";
         try {
            bd.preparaBD();
            bd.preparedStatement.setInt(1, idModal);
            bd.preparedStatement.setInt(2, paisCombo);
            bd.preparedStatement.setInt(3, 1);
            bd.preparedStatement.setInt(4, 0);
            bd.preparedStatement.setInt(5, 0);
            bd.preparedStatement.execute();
            JOptionPane.showMessageDialog(null, "Dados inseridos com Sucesso!");
         }catch(SQLException e) {
            e.printStackTrace();
         }
      }else if(jRadioButton2.isSelected()) {
         paisCombo = cbPais.getSelectedIndex();
         idModal = cbModalidade.getSelectedIndex();
         bd.sql = "INSERT INTO premiacao(id_modalidade,"
               + "id_pais, ouro, prata, bronze)"
               + "values(?, ?, ?, ?, ?)";

         try {
            bd.preparaBD();
            bd.preparedStatement.setInt(1, idModal);
            bd.preparedStatement.setInt(2, paisCombo);
            bd.preparedStatement.setInt(3, 0);
            bd.preparedStatement.setInt(4, 1);
            bd.preparedStatement.setInt(5, 0);
            bd.preparedStatement.execute();
            JOptionPane.showMessageDialog(null, "Dados inseridos com Sucesso!");
         }catch(SQLException e) {
```

```
                e.printStackTrace();
            }

        }else if(jRadioButton3.isSelected()) {
            paisCombo = cbPais.getSelectedIndex();
            idModal = cbModalidade.getSelectedIndex();
            bd.sql = "INSERT INTO premiacao(id_modalidade,"
                  + "id_pais, ouro, prata, bronze)"
                  + "values(?, ?, ?, ?, ?)";

            try {
               bd.preparaBD();
               bd.preparedStatement.setInt(1, idModal);
               bd.preparedStatement.setInt(2, paisCombo);
               bd.preparedStatement.setInt(3, 0);
               bd.preparedStatement.setInt(4, 0);
               bd.preparedStatement.setInt(5, 1);
               bd.preparedStatement.execute();
               JOptionPane.showMessageDialog(null, "Dados inseridos com Sucesso!");
            }catch(SQLException e) {
               e.printStackTrace();
            }
         }
         cbPais.setSelectedItem("");
         cbModalidade.setSelectedItem("");
         exibeClassificacao();
      }
   });
  }
  public void exibeClassificacao() {
     Connection connection = null;
     PreparedStatement preparedStatementExibe = null;
     try {
        connection = DriverManager.getConnection(BDUtil.getUrlBd(),
              BDUtil.getUserBd(), BDUtil.getPassBd());

        String selectSqlPremiacao = "SELECT pais.nome AS pais,"
              + "SUM (premiacao.ouro) AS ouro,"
              + "SUM (premiacao.prata) AS prata,"
              + "SUM (premiacao.bronze) AS bronze"
              + "FROM premiacao, pais"
              + "WHERE pais.id = premiacao.id_pais"
              + "GROUP BY pais.nome"
              + "ORDER BY ouro DESC, prata DESC, bronze DESC, pais";

     }catch (Exception e) {

     }

  }
}
```

Aula Trigger

```
CREATE TABLE DEPARTAMENTO(
   COD_DEPARTAMENTO SERIAL PRIMARY KEY,
   NOME_DEPARTAMENTO VARCHAR(30) NOT NULL
);
CREATE TABLE SETOR(
   COD_SETOR SERIAL PRIMARY KEY,
   NOME_SETOR VARCHAR(30) NOT NULL,
   COD_DPTO INT NOT NULL,
   FOREIGN KEY (COD_DPTO) REFERENCES DEPARTAMENTO(COD_DEPARTAMENTO)
);
```

```
CREATE SEQUENCE SEQ_FUNC INCREMENT 1 START 1500;
CREATE TABLE FUNCIONARIO(
   MATRICULA INT PRIMARY KEY DEFAULT NEXTVAL('SEQ_FUNC') NOT NULL,
   NOME VARCHAR(100) NOT NULL,
   COD_SETOR INT NOT NULL,
   SALARIO FLOAT NOT NULL,
   FOREIGN KEY (COD_SETOR) REFERENCES SETOR(COD_SETOR)
);

ALTER TABLE FUNCIONARIO ALTER COLUMN SALARIO SET DEFAULT 0.0;

CREATE TABLE ATUALIZACAO_SALARIAL(
   LOG_ALTERACAO SERIAL PRIMARY KEY,
   MATRICULA_FUNC INT NOT NULL,
   DATA_ALTERACAO TIMESTAMP NOT NULL,
   SAL_ANTERIOR FLOAT NOT NULL,
   SAL_ATUAL FLOAT NOT NULL,
   FOREIGN KEY (MATRICULA_FUNC) REFERENCES FUNCIONARIO(MATRICULA)
);

INSERT INTO DEPARTAMENTO(NOME_DEPARTAMENTO)
VALUES ('ADMINSTRATIVO');
INSERT INTO DEPARTAMENTO(NOME_DEPARTAMENTO)
VALUES ('TECNOLOGIA');
INSERT INTO DEPARTAMENTO(NOME_DEPARTAMENTO)
VALUES ('FINANCEIRO');
INSERT INTO DEPARTAMENTO(NOME_DEPARTAMENTO)
VALUES ('DIRETORIA');
INSERT INTO DEPARTAMENTO(NOME_DEPARTAMENTO)
VALUES ('OPERACIONAL');
INSERT INTO SETOR(NOME_SETOR, COD_DPTO)
VALUES('FROTA', 1);
INSERT INTO SETOR(NOME_SETOR, COD_DPTO)
VALUES('SECRETARIA', 1);
INSERT INTO SETOR(NOME_SETOR, COD_DPTO)
VALUES('RECEPCAO', 1);
INSERT INTO SETOR(NOME_SETOR, COD_DPTO)
VALUES('CONTROLE', 1);
INSERT INTO SETOR(NOME_SETOR, COD_DPTO)
VALUES('SUPORTE TECNICO', 2);
INSERT INTO FUNCIONARIO(NOME, COD_SETOR, SALARIO)
VALUES('HELENA', 1, 2740.25);
INSERT INTO FUNCIONARIO(NOME, COD_SETOR, SALARIO)
VALUES('GILMAR', 1, 2500.00);
INSERT INTO FUNCIONARIO(NOME, COD_SETOR, SALARIO)
VALUES('FELIPE', 2, 4250.00);
SELECT * FROM FUNCIONARIO;

CREATE OR REPLACE FUNCTION ATUALIZA_SALARIO()
RETURNS TRIGGER AS
$$
BEGIN
   INSERT INTO ATUALIZACAO_SALARIAL(MATRICULA_FUNC, DATA_ALTERACAO,
   SAL_ANTERIOR, SAL_ATUAL)
   VALUES(NEW.MATRICULA, NOW(), OLD.SALARIO, NEW.SALARIO);
   RETURN NULL;
END;
$$ LANGUAGE PLPGSQL;

CREATE TRIGGER TG_ATUALIZA_SALARIO
AFTER INSERT OR UPDATE
ON FUNCIONARIO
FOR EACH ROW
```

```
EXECUTE PROCEDURE ATUALIZA_SALARIO();


SELECT * FROM ATUALIZACAO_SALARIAL;
UPDATE FUNCIONARIO SET SALARIO = 8000.00
WHERE MATRICULA = 1501;
```

Portugol

```
//ARQUIVO 1
//programa
//{

//Tipos de variáveis e visibilidade
//inteiro int = 8456
//real rl = 2115.65
//caracter car = 't'//char
//logico log = verdadeiro //bolean
//cadeia cad = "Texto longo"//varchar

//Função para iniciar o programa
//funcao inicio()
//{
//escreva("Olá Mundo")
//escreva(teste()) // aceita apenas a function, pois "result" é uma variável local e não global

//}
//funcao inteiro teste(){
//inteiro result = 10 * int
//retorne result

//}


//}

//ARQUIVO 3
programa{

funcao inicio(){
inteiro int = 1
enquanto(int <= 100){
se(int % 3 == 0){
escreva(int, " ") // lembrar do espaço
}
int = int + 1 // incremento fora do SE
}
}
}

//ARQUIVO 4
programa{

funcao inicio(){
para(inteiro int = 1; int <= 100; int++){

se(int % 3 == 0){
```

```
escreva(int + "\n") // lembrar do espaço
}

}
}
}

//ARQUIVO 5
programa{

funcao inicio(){
inteiro int = 1
enquanto(int <18){
escreva("Digite sua idade: ")
leia(int)
}
escreva("Maior de Idade ") // lembrar do espaço
}
}

//ARQUIVO 6
programa{
inteiro soma = 0
funcao inicio(){
para( inteiro i = 10; i <= 20; i++){
soma += i //soma = soma + i
}
escreva(soma)
}

}

//ARQUIVO 8
programa{
cadeia alunos[8]
funcao inicio(){
/*alunos[0] = "Leonardo"
alunos[1] = "Nicoly"
alunos[2] = "Vitor"
alunos[3] = "Vinicius"
alunos[4] = "João"
alunos[5] = "Vitória"
alunos[6] = "Barbara"
alunos[7] = "Jonas"*/
para (inteiro i = 0; i<8; i++){
escreva("Digite um nome: ")
leia(alunos[i])
}
para (inteiro j = 0; j<8; j++){
escreva(alunos[j] + "\n")
}
}

}
```

```
//ARQUIVO 9
programa{
    cadeia clientes[] = {"Carlos", "Maria", "Ana", "Paulo","Bruno"}
    inteiro idades[] = {22, 19, 11, 16, 18}
    funcao inicio(){
        para(inteiro i = 0; i < 5; i++){
            se (idades[i] >= 18){
                escreva(clientes[i] + "\n")
            }
        }
    }

}
```

Notas Soltas

BD1

```
select curso.nome, disciplina.nome from curso, disciplina, curso_disciplina
where curso.id = curso_disciplina.id_curso and disciplina.id = curso_disciplina .id_disciplina

select * from curso_disciplina order by id_curso desc, id_disciplina desc;
-- delete from curso_disciplina where id_curso = 3 and id_disciplina = 2
```

BD2

```
select curso.nome as curso, disciplina.nome as disciplina, professor.nome as professor
from curso, disciplina, professor, curso_disciplina, professor_disciplina
where curso.id = curso_disciplina.id_curso
and professor.matricula = professor_disciplina.matricula_prof
and disciplina.id = professor_disciplina.id_disciplina
and disciplina.id = curso_disciplina.id_disciplina

order by curso.nome, disciplina.nome, professor.nome
```

BD3

```
select curso.nome as curso, disciplina.nome as disciplina, professor.nome as professor
from curso, disciplina, professor, curso_disciplina, professor_disciplina
where curso.id = curso_disciplina.id_curso
and professor.matricula = professor_disciplina.matricula_prof
and disciplina.id = professor_disciplina.id_disciplina
and disciplina.id = curso_disciplina.id_disciplina

order by curso.nome, disciplina.nome, professor.nome
```

BD4

```
-- alter table usuario
-- add column endereco varchar(100)

-- CREATE TABLE usuario(
-- id_user serial primary key,
-- nome_completo varchar(200),
-- login varchar(50),
-- email varchar(100),
-- cpf varchar (11)
-- );
-- CREATE TABLE senha(
-- id_senha serial primary key,
```

```
-- id_user int,
-- senha varchar(15),
-- foreign key(id_user) references usuario(id_user)
-- );
-- drop table senha
-- select * from usuario

BD5
-- update aluno set nome = 'MARIAH'
-- where matricula = 1  -- não esquecer do WHERE senão todos os nomes são trocados

-- select * from aluno
-- where nome like 'HUMBERTO%'

-- select * from aluno
-- order by matricula desc

-- delete from aluno
-- where nome = 'HUMBERTO'

BD6
-- CREATE TABLE categoria(
-- id_categoria serial primary key,
-- nome_categoria varchar(50),
-- genero char
-- );

-- CREATE TABLE modalidade(
-- id_modalidade serial primary key,
-- nome_modalidade varchar(50),
-- id_categoria int,
-- foreign key(id_categoria) references categoria(id_categoria)
-- );

-- CREATE TABLE pais(
-- id_pais serial primary key,
-- nome_pais varchar(50)
-- );

-- CREATE TABLE premiacao(
-- id_premiacao serial primary key,
-- id_modalidade int,
-- id_pais int,
-- ouro int,
-- prata int,
-- bronze int,
-- foreign key(id_modalidade) references modalidade(id_modalidade),
-- foreign key(id_pais) references pais(id_pais)
-- );

BD7
package geral;

import java.awt.Color;
import java.awt.Font;
```

```
import java.sql.SQLException;
import java.util.Vector;

import javax.swing.ButtonGroup;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JRadioButton;
import javax.swing.JTable;

public class Tela extends JFrame{
public JComboBox cbPais = new JComboBox();
public JComboBox cbModalidade = new JComboBox();
public JTable tableMedalha;

public Tela() throws SQLException {
super("Tela de Cadastro de Medalha");
setSize(485, 680);
setLayout(null);
getContentPane().setBackground(new Color(240, 230, 140));
setResizable(false);
setDefaultCloseOperation(EXIT_ON_CLOSE);
setLocationRelativeTo(null);
JLabel jLtitulo = new JLabel("Olimpíadas 2024 - Quadro de Medalhas");
jLtitulo.setBounds(5, 0, 850, 60);
jLtitulo.setFont( new Font ("Arial", Font.BOLD, 25));
add(jLtitulo);
JLabel jlNome = new JLabel("País:");
jlNome.setBounds(10, 70, 850, 60);
add(jlNome);
BDUtil bdPais = new BDUtil();
bdPais.executaBDSelect("Select * from pais", "nome_pais");
cbPais.setModel(new DefaultComboBoxModel(new Vector(bdPais.getLista())));
cbPais.setBounds(70, 90, 200, 20);
add(cbPais);
JLabel jLModal = new JLabel("Modalidade:");
jLModal.setBounds(10, 110, 850, 60);
add(jLModal);
BDUtil bdModal = new BDUtil();
bdModal.executaBDSelect(
"select modalidade, "
+ "modalidade.id_modalidade, "
+ "modalidade.nome_modalidade, "
+ "categoria.nome_categoria, "
+ "categoria.genero, "
+ "concat(modalidade.nome_modalidade, '-', "
+ "categoria.nome_categoria, '-', "
+ "categoria.genero) "
+ "as modalidade from modalidade, "
+ "categoria where "
+ "categoria.id_categoria = "
+ "modalidade.id_categoria",
```

```
"modalidade");

cbModalidade.setModel(new DefaultComboBoxModel(new Vector(bdModal.getLista())));
cbModalidade.setBounds(10, 150, 250, 20);
add(cbModalidade);
JLabel jlMedalhas = new JLabel("Medalhas");
jlMedalhas.setBounds(10, 150, 850, 60);
add(jlMedalhas);
JRadioButton jRadioButton1 = new JRadioButton();
JRadioButton jRadioButton2 = new JRadioButton();
JRadioButton jRadioButton3 = new JRadioButton();
ButtonGroup g1 = new ButtonGroup();
jRadioButton1.setText("OURO");
jRadioButton1.setBackground(new Color(255, 215, 0));
jRadioButton2.setText("PRATA");
jRadioButton2.setBackground(new Color(192, 192, 192));
jRadioButton3.setText("BRONZE");
jRadioButton3.setForeground(Color.white);
jRadioButton3.setBackground(new Color(210, 105, 30));
jRadioButton1.setBounds(200, 170, 70, 20);
jRadioButton2.setBounds(120, 190, 70, 20);
jRadioButton3.setBounds(280, 200, 80, 20);
add(jRadioButton1);
add(jRadioButton2);
add(jRadioButton3);
g1.add(jRadioButton1);
g1.add(jRadioButton2);
g1.add(jRadioButton3);
JLabel jlAros = new JLabel();
jlAros.setVisible(true);
jlAros.setBounds(295, 40, 300, 100);
jlAros.setIcon(new ImageIcon(getClass().getResource("aros.png")));
add(jlAros);
JLabel jlPodio = new JLabel();
jlPodio.setBounds(100, 180, 300, 100);
jlPodio.setIcon(new ImageIcon(getClass().getResource("podium.jpg")));
add(jlPodio);
JButton jbCadastro = new JButton("Cadastrar");
jbCadastro.setBounds(190, 300, 100, 20);
jbCadastro.setVisible(true);
add(jbCadastro);
setVisible(true);
}
}

BD8
-- CREATE SEQUENCE SEQ_MAT INCREMENT 5 START 1500;
-- CREATE TABLE funcionario(
-- MATRICULA INT PRIMARY KEY DEFAULT NEXTVAL('SEQ_MAT'),
-- NOME VARCHAR(1000)

-- )
SELECT * FROM FUNCIONARIO
INSERT INTO FUNCIONARIO (NOME) VALUES ('MARIA');
INSERT INTO FUNCIONARIO (NOME) VALUES ('PEDRO');
```

```
INSERT INTO FUNCIONARIO (NOME) VALUES ('ISA');
INSERT INTO FUNCIONARIO (NOME) VALUES ('JORGE');

BD9
-- create function formata_moeda(valor float)
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return concat('R$', round(cast(valor as numeric),2));
-- end;
-- $$;

BD10
-- create table vendas(
-- codigo serial primary key,
-- cod_prod int not null,
-- cod_cli int not null,
-- data_venda date not null default current_date ----padronizar com a data atual
-- );

-- insert into vendas(cod_prod, cod_cli) values(15, 63);
-- select * from vendas;

-- insert into vendas(cod_prod, cod_cli, data_venda)
-- values(10, 44, '29/10/2024'); ----prioridade para a data manual, mesmo com a
padronização
-- select * from vendas;

-- alter table vendas add column valor float not null default 0;

-- update  vendas set valor = 50.0 where  cod_prod = 15;
-- select * from vendas;

-- update  vendas set valor = 35.0 where  cod_prod = 10;
-- select * from vendas;

-- create function formata_moeda(valor float)
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return concat('R$', round(cast(valor as numeric),2));
-- end;
-- $$;

-- select cod_cli, cod_prod, formata_moeda(valor) as PREÇO from vendas;

-- alter table vendas add column qtd int default 0;
-- select * from vendas;

-- update vendas set qtd = 10 where codigo = 4;
-- select * from vendas;

-- alter table vendas add column total int;

-- create function multiplicar(valor int)
```

```
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return sum(qtd*valor);
-- end;
-- $$;
-- select * from vendas;

-- create function exibe_total(val float, qtde int)
-- returns float language PLPGSQL as
-- $$
-- begin
-- return sum(val * qtde);
-- end;
-- $$;

select cod_cli, cod_prod, valor, qtd,
exibe_total(valor, qtd) as totais from vendas;
-- select * from vendas

BD11
-- create table vendas(
-- codigo serial primary key,
-- cod_prod int not null,
-- cod_cli int not null,
-- data_venda date not null default current_date ----padronizar com a data atual
-- );

-- insert into vendas(cod_prod, cod_cli) values(15, 63);
-- select * from vendas;

-- insert into vendas(cod_prod, cod_cli, data_venda)
-- values(10, 44, '29/10/2024'); ----prioridade para a data manual, mesmo com a
padronização
-- select * from vendas;

-- alter table vendas add column valor float not null default 0;

-- update  vendas set valor = 50.0 where  cod_prod = 15;
-- select * from vendas;

-- update  vendas set valor = 35.0 where  cod_prod = 10;
-- select * from vendas;

-- create function formata_moeda(valor float)
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return concat('R$', round(cast(valor as numeric),2));
-- end;
-- $$;

-- select cod_cli, cod_prod, formata_moeda(valor) as PREÇO from vendas;

-- alter table vendas add column qtd int default 0;
```

```
-- select * from vendas;

-- update vendas set qtd = 10 where codigo = 4;
-- select * from vendas;

-- alter table vendas add column total int;

-- create function multiplicar(valor int)
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return sum(qtd*valor);
-- end;
-- $$;
-- select * from vendas;

-- create function exibe_total(val float, qtde int)
-- returns float language PLPGSQL as
-- $$
-- begin
-- return sum(val * qtde);
-- end;
-- $$;

-- select cod_cli, cod_prod, valor, qtd,
-- exibe_total(valor, qtd) as totais from vendas;
-- select * from vendas

-- create function formata_total(valor float, qtd int)
-- returns varchar(20) language PLPGSQL as
-- $$
-- begin
-- return concat('R$', round(cast(sum(valor * qtd)as numeric),2));
-- end;
-- $$;

select cod_cli, cod_prod, valor, qtd,
formata_total(valor, qtd) as totais from vendas;
select * from vendas

BD12
-- create table Departamento(
-- cod_departamento serial primary key,
-- nome_departamento varchar(200)
-- );

--  insert into departamento(nome_departamento)
--  values ('administrativo');
--  insert into departamento(nome_departamento)
--  values ('tecnologia');
--  insert into departamento(nome_departamento)
--  values ('financeiro');
--  insert into departamento(nome_departamento)
--  values ('inovação');
--  insert into departamento(nome_departamento)
```

```
--  values ('RH');
--  select * from departamento;

-- create table Setor(
-- cod_setor serial primary key,
-- nome_setor varchar(200),
-- cod_departamento int,
-- foreign key(cod_departamento) references Departamento(cod_departamento)
-- );

--  insert into Setor(nome_setor, cod_departamento )
--  values ('logística', '1');
--  insert into Setor(nome_setor, cod_departamento )
--  values ('suporte', '2');
--  insert into Setor(nome_setor, cod_departamento )
--  values ('patrocínio', '3');
--  insert into Setor(nome_setor, cod_departamento )
--  values ('marketing', '4');
--  insert into Setor(nome_setor, cod_departamento )
--  values ('gerenciamento', '5');
--  select * from setor;


-- create table Funcionario(
--  matricula_func serial primary key,
-- nome varchar(200),
-- cod_setor int,
-- salario float,
-- foreign key(cod_setor) references Setor(cod_setor)
-- );

--  insert into Funcionario(nome, cod_setor, salario)
--  Values('Creusa', 1, 5000.00);
--  insert into Funcionario(nome, cod_setor, salario)
--  Values('Raimundo', 2, 2500.00);
--  insert into Funcionario(nome, cod_setor, salario)
--  Values('Ferdinando', 3, 4000.00);
--  select * from funcionario;

-- create table atualizacao_salarial(
-- log_alteracao serial primary key,
-- matricula_func int not null,
-- data_alteracao timestamp not null,
-- sal_anterior float not null,
-- sal_atual float not null,
-- foreign key(matricula_func) references Funcionario(matricula_func)
-- );

-- create or replace function atualiza_salario()
-- returns trigger  as
-- $$
-- begin
-- insert into atualizacao_salarial(matricula_func, data_alteracao, sal_anterior, sal_atual)
-- values (new.matricula_func, now(), old.salario, new.salario);
-- return null;
```

```
-- end;
-- $$ language PLPGSQL;

-- create trigger tg_atualiza_salario
-- after insert or update
-- on funcionario
-- for each row
-- execute procedure atualiza_salario();

select * from atualizacao_salarial;
update funcionario set salario = 6000.00
where matricula_func = 1;

BD13
-- create table Produto(
-- codigo serial primary key,
-- descricao varchar(50)not null unique,
-- preco float not null
-- );

-- create table Prod_hist(
-- cod_hist serial primary key,
-- cod_prod int,
-- preco_ant float,
-- preco_atual float,
-- data_alt date
-- );

-- insert into produto values (1, 'caneta bic azul ', 1.50);
-- insert into produto values (2, 'caneta bic vermelha ', 1.50);
-- insert into produto values (3, 'caneta bic preta ', 1.50);
-- insert into produto values (4, 'lapis faber castel', 1.50);
-- insert into produto values (5, 'caderno surfista', 10.50);

-- select * from produto;

-- create or replace function atualiza_preco()
-- returns trigger  as
-- $$
-- begin
-- if new.preco >= old.preco + 1 then
-- insert into Prod_hist
-- (cod_prod, preco_ant, preco_atual, data_alt)
-- values
-- (old.codigo, old.preco, new.preco, now());
-- end if;
-- return new;
-- end;
-- $$ language PLPGSQL;

-- create or replace function atualiza_preco()
-- returns trigger  as
-- $$
-- begin
-- insert into Prod_hist
```

```
-- (cod_prod, preco_ant, preco_atual, data_alt)
-- values
-- (old.codigo, old.preco, new.preco, now());
-- return new;
-- end;
-- $$ language PLPGSQL;

-- create or replace trigger tg_prod_hist
-- after update on Produto
-- for each row
-- execute procedure atualiza_preco();

-- update produto set preco = 2.99 where codigo = 2;
-- select * from prod_hist;

BD14
-- create table Cliente(
-- id_cliente serial primary key,
-- nome varchar(200)not null unique,
-- numero varchar(200)
-- );

-- create table auditar(
-- id_auditar serial primary key,
-- id_cliente int,
-- numero_antes varchar(200),
-- numero_atual varchar(200),
-- data_alt date
-- );

create table excluidos(
id_exclusao serial primary key,
id_cliente int,
numero varchar(200),
data_exc date
);

-- ALTER TABLE auditar
-- ALTER COLUMN numero_antes TYPE varchar(200),
-- ALTER COLUMN numero_atual TYPE varchar(200);

-- --Pessoas inseridas para o trigger update
-- insert into Cliente values (1, 'Fernanda', '3333-4605');
-- insert into Cliente values (2, 'Abgail', '3223-4605');
-- insert into Cliente values (3, 'Rogério', '1102-0231');
-- insert into Cliente values (4, 'Rafael', '1212-3839');
-- insert into Cliente values (5, 'Isabelle', '3233-3039');

--pessoas inseridas para o trigger delete
insert into Cliente values (6, 'Matheus', '3333-4605');
insert into Cliente values (7, 'Carol', '3223-4605');
insert into Cliente values (8, 'Camila', '1102-0231');
insert into Cliente values (9, 'Rosana', '1212-3839');
insert into Cliente values (10, 'Matilda', '3233-3039');
```

```
select * from Cliente;

-- create or replace function tg_cad_clientes()
-- returns trigger  as
-- $$
-- begin
-- insert into auditar
-- (id_cliente, numero_antes, numero_atual, data_alt)
-- values
-- (old.id_cliente, old.numero, new.numero, now());
-- return new;
-- end;
-- $$ language PLPGSQL;

-- create or replace trigger tg_auditar
-- after update on Cliente
-- for each row
-- execute procedure tg_cad_clientes();

-- update cliente set numero = '1323-2132' where id_cliente = 1;
-- update cliente set numero = '1323-2322' where id_cliente = 2;
-- update cliente set numero = '6767-8877' where id_cliente = 3;
-- update cliente set numero = '8788-0788' where id_cliente = 4;
-- update cliente set numero = '2323-3233' where id_cliente = 5;
-- select * from auditar;

create or replace function antigos_clientes()
returns trigger  as
$$
begin
insert into excluidos
(id_cliente, numero, data_exc)
values
(old.id_cliente, old.numero);
return new;
end;
$$ language PLPGSQL;

create or replace trigger tg_excluir
after update on Cliente
for each row
execute procedure tg_cad_clientes();

DELETE FROM cliente WHERE id_cliente = 6;
DELETE FROM cliente WHERE id_cliente = 7;
DELETE FROM cliente WHERE id_cliente = 8;
DELETE FROM cliente WHERE id_cliente = 9;
DELETE FROM cliente WHERE id_cliente = 10;
select * from excluidos;

BD15
create table cliente(
nome varchar(300),
cpf varchar(11) primary key,
data_nasc date,
```

```
muncipio varchar(50),
genero varchar(01),
valor_produto numeric(10,2)
);

insert into cliente
(nome, cpf, data_nasc, muncipio, genero, valor_produto)
values
-- ('Maria', '12345678910', '07/10/2024', 'São José', 'F', 22.50);
-- ('Fernanda', '12345338310', '10/10/2024', 'Florianópolis', 'F', 40.50);
-- ('João', '22325678910', '20/10/2024', 'Monte Castelo', 'M', 82.50);

select * from cliente;
when 'F' then 'Feminino'
when 'M' then 'Masculino'
end as genero
from cliente;

select data_nasc, coalesce(data_nasc, 'Não Informado')
from cliente;

BD16
create domain nome_medio as varchar(50);
create domain data_only as date;
create domain moeda as numeric(6,2);

create table funcionario(
matricula serial primary key,
nome nome_medio not null,
data_admissao data_only,
salario moeda not null,
cargo nome_medio not null
);

insert into funcionario
-- values (4, 'Isabel', '15/07/2001', 3560.54, 'Analista')
-- values (5, 'Rafaella', '15/08/2001', 4000.54, 'Marketeira')
-- values (6, 'Matilda', '15/09/2001', 5000.54, 'Fiscal')
-- values (7, 'Fernanda', '20/07/2004', 6960.54, 'RH')
-- values (8, 'Cassandra', '22/07/2005', 4545.54, 'Gerente de Produção')
-- values (9, 'Samanta', '23/07/2003', 4355.54, 'Contadora')
-- values (10, 'Rafael', '22/10/2001', 8999.54, 'Administradora')
-- values (11, 'Michel', '12/07/2020', 7687.54, 'Atendente')
-- values (12, 'Katrina', '12/07/2020', 4788.54, 'Suporte Técnico')
-- values (13, 'Humberto', '12/07/2020', 8878.54, 'Porteiro')
select * from funcionario

select avg(salario) as Media_Salarial from funcionario
where salario > 5000;

select count (matricula) from funcionario
where nome like 'M%';

select max (salario) from funcionario;
```

```
select min (salario) from funcionario;

create table descontos(
cod_desc serial primary key,
mat_func smallint not null,
valor_desc moeda not null,
foreign key (mat_func) references funcionario(matricula)
);

select * from descontos;

insert into descontos
-- values (1, 11, 112.12)
-- values (2, 11, 132.12)
-- values (3, 12, 43.12)
-- values (4, 12, 23.12)
-- values (5, 13, 234.12)
-- values (6, 11, 2434.12)

select sum(valor_desc) from descontos;


select mat_func, sum(valor_desc) from descontos
group by mat_func;

select func.nome, func.salario, sum(dsc.valor_desc) as desconto,
(salario - dsc.valor_desc) as Salario_Liquido
from funcionario func, descontos dsc
where func.matricula = dsc.mat_func
group by func.nome, dsc.valor_desc, func.salario;

BD17
create table paul_got_back(
codigo_show serial primary key,
data_show date not null unique,
pais varchar(20) not null,
cidade varchar(50) not null,
valor_ingresso numeric(10,2) --até duas casas após a virgular
);

insert into paul_got_back
(data_show, pais, cidade, valor_ingresso)
values
-- ('25/01/2024', 'Australia', 'Melbourne', 95.50);
-- ('27/01/2024', 'Australia', 'Sidney', 95.50);
-- ('06/02/2024', 'Inglaterra', 'Londres', 125.50);
-- ('08/02/2024', 'Inglaterra', 'Londres', 125.50);
-- ('03/03/2024', 'EUA', 'Nova York', 55.50);
-- ('15/03/2024', 'Brasil', 'São Paulo', 850.50);
-- ('16/03/2024', 'Brasil', 'São Paulo', 850.50);
-- ('19/03/2024', 'Brasil', 'Florianópolis', 850.50);
select * from paul_got_back;

-- select cidade,
-- case extract(month from data_show)
```

```
-- when 1 then 'Janeiro'
-- when 2 then 'Fevereiro'
-- when 3 then 'Março'
-- when 4 then 'Abril'
-- when 5 then 'Maio'
-- when 6 then 'Junho'
-- when 7 then 'Julho'
-- when 8 then 'Agosto'
-- when 9 then 'Setembro'
-- when 10 then 'Outubro'
-- when 11 then 'Novembro'
-- when 12 then 'Dezembro'
-- end as mes
-- from paul_got_back;

-- select substring (pais from 1 for 3), cidade,
-- case extract(month from data_show)
-- when 1 then 'Janeiro'
-- when 2 then 'Fevereiro'
-- when 3 then 'Março'
-- when 4 then 'Abril'
-- when 5 then 'Maio'
-- when 6 then 'Junho'
-- when 7 then 'Julho'
-- when 8 then 'Agosto'
-- when 9 then 'Setembro'
-- when 10 then 'Outubro'
-- when 11 then 'Novembro'
-- when 12 then 'Dezembro'
-- end as mes
-- from paul_got_back;

create table cliente(
cod_cli serial primary key,
nome varchar(50) not null,
cpf varchar(11)
);

insert into cliente(nome, cpf)
values ('Maria', '12345678910');
select * from cliente;

select nome, coalesce(cpf, 'Não Informado')
from cliente;

BD18
-- create table cliente(
-- nome varchar(300),
-- cpf varchar(11) primary key,
-- data_nasc date,
-- muncipio varchar(50),
-- genero varchar(01),
-- valor_produto numeric(10,2)
-- );
```

```
-- insert into cliente
-- (nome, cpf, data_nasc, muncipio, genero, valor_produto)
-- values
-- ('Maria', '12345678910', '07/10/2024', 'São José', 'F', 22.50),
-- ('Fernanda', '12345338310', '10/10/2024', 'Florianópolis', 'F', 40.50),
-- ('João', '22325678910', '20/10/2024', 'Monte Castelo', 'M', 82.50);
```

BD19

```
-- create table cliente(
-- nome varchar(300),
-- cpf varchar(11) primary key,
-- data_nasc date,
-- muncipio varchar(50),
-- genero varchar(01),
-- valor_produto numeric(10,2)
-- );

-- insert into cliente
-- (nome, cpf, data_nasc, muncipio, genero, valor_produto)
-- values
-- ('Maria', '12345678910', '07/10/2024', 'São José', 'F', 22.50),
-- ('Fernanda', '12345338310', '10/10/2024', 'Florianópolis', 'F', 40.50),
-- ('João', '22325678910', '20/10/2024', 'Monte Castelo', 'M', 82.50);
```

BD20

```
-- 1
-- SELECT nome FROM vendedor
-- ORDER BY nome, nome ASC

-- 2
-- SELECT nome, valor FROM produto
-- WHERE valor > 200
-- ORDER BY valor asc;

-- 3 -----------------
--  SELECT nome, valor FROM produto
-- where  valor = valor + ( (valor/100) * 0,1)
-- ORDER BY nome asc;

-- 4
-- select nome from municipio
-- where iduf = 5

-- 5
-- select data_pedido from pedido
-- where data_pedido between '2008-04-10' and '2008-04-10'
-- order by data_pedido desc;

-- 6
-- select valor from pedido
-- where valor between '1000' and '1500'
-- order by valor desc;

-- 7
-- select valor from pedido
```

```
-- where valor not between '100' and '500'
-- order by valor desc;

-- 8
-- select * from pedido
-- where idvendedor = 1
-- order by valor desc;

--  9
-- select * from pedido
-- where idcliente = 1
-- order by valor asc;

-- 10
-- select * from pedido
-- where idvendedor = 1 and idcliente = 15
-- order by valor desc;

-- 11
-- select * from pedido
-- where idtransportadora = 2;

-- 12
-- select * from pedido
-- where idvendedor = 5 or idvendedor = 7;

-- 13
-- select * from cliente
-- where idmunicipio = 1 or idmunicipio = 2;




-- 21 não é fornecedor e sim cliente

BD21
-- 1
-- select nome from vendedor
-- order by nome, nome ASC

-- 2
-- select nome, valor from produto
-- where valor > 200
-- order by valor asc;

-- 3
-- select nome, valor, valor * 1.1 as valor_reajustado
-- from produto order by produto asc;

-- 4
-- select nome from municipio
-- where iduf = 5
```

```
-- 5
-- select data_pedido from pedido
-- where data_pedido between '2008-04-10' and '2008-04-10'
-- order by data_pedido desc;

-- 6
-- select valor from pedido
-- where valor between '1000' and '1500'
-- order by valor desc;

-- 7
-- select valor from pedido
-- where valor not between '100' and '500'
-- order by valor desc;

-- 8
-- select * from pedido
-- where idvendedor = 1
-- order by valor desc;

--  9
-- select * from pedido
-- where idcliente = 1
-- order by valor asc;

-- 10
-- select * from pedido
-- where idvendedor = 1 and idcliente = 15
-- order by valor desc;

-- 11
-- select * from pedido
-- where idtransportadora = 2;

-- 12
-- select * from pedido
-- where idvendedor = 5 or idvendedor = 7;

-- 13
-- select * from cliente
-- where idmunicipio = 1 or idmunicipio = 2;

-- 14
-- select * from cliente
-- where idmunicipio NOT IN (1, 2);

-- 15
-- select * from cliente
-- where logradouro is null

-- 16
-- select * from cliente
-- where logradouro like 'Av%';

-- 17
```

```
-- select * from vendedor
-- where nome like 'S%';

-- 18
-- select * from vendedor
-- where nome like '%a';

-- 19
-- select * from vendedor
-- where nome not like 'A%';

-- 20
-- select * from municipio
-- where nome like 'P%' and iduf = 1;

-- 21 (fornecedor = cliente)
-- select * from cliente
-- where logradouro is not null
-- and idcomplemento is not null
-- and idbairro is not null
-- and idmunicipio is not null

-- 22
-- select Produto.nome, Pedido.idpedido from Produto, Pedido
-- where Pedido.idpedido = 1;

-- 23
-- select Produto.nome, Pedido.idpedido from Produto, Pedido
-- where Pedido.idpedido in (6, 10);




--Questões que eu tive que pesquisar como fazia -_- :
-- 3, 14, 15, 21, 22 e 23

BD22
-->1
-- select nome from cliente
-- where idmunicipio = (select idmunicipio
-- from cliente where nome = 'Manoel')
-- and nome not like 'Manoel';

-->2

-->3

-->4

-->5
select nome, idmunicipio from cliente
where idmunicipio in (select idmunicipio from transportadora
```

```
where idmunicipio = 9 or idmunicipio = 5);

-->6
select nome, idmunicipio from cliente
where idmunicipio in (select idmunicipio from transportadora
where idmunicipio = 9 or idmunicipio = 5);

-- select nome, valor, valor * 0.5 as valor_reajustado
-- from produto order by produto asc;



select * from cliente
select * from transportadora
select * from municipio
```