(✓) Curso Técnico

Quem Investe no futuro faz QI

Informática para Internet

Desenvolvimento de Sistemas

Web III

Unidade VI

1. Banco de dados relacional

Há muito tempo que os bancos de dados são extremamente importantes em vários segmentos da sociedade. Pensar em sistema educacional que permite o registro das notas, um sistema financeiro, um sistema bancário ou sistema de pedidos via web, sem um banco de dados é uma tarefa pouquíssimo fácil nos dias atuais.

Um banco de dados relacional é um recurso amplamente conhecido pelos profissionais de tecnologia da informação, pois é um tipo de banco de dados que armazena e permite o acesso a pontos de dados relacionados entre si, através de uma chave primária – PK(Primary Key) e uma chave estrangeira – FK(Foreign Key). Um banco de dados relacional pode ser visto como uma ou um conjunto de tabelas em cada linha de cada tabela é um registro com uma ID exclusiva chamada chave primária – PK(Primary Key). As colunas de cada tabela contém atributos ou informações dos dados.

Então pode-se dizer que um banco de dados relacional é composto de tabelas ou relações, valendo ressaltar que a palavra tabela é mais comum ou popular nos espaços menos técnicos, pois a palavra relação é utilizada na literatura original sobre a abordagem relacional.

As tabelas são formadas por colunas(atributos) e via de regra uma dessas colunas é definida como chave primária, conforme exemplifica-se:

tbpessoa		
id	nome	cpf
1	MARIA	88888
2	JOSÉ	77777
3	LAURA	66666

A coluna id do exemplo acima é a chave primária da tabela tbpessoa, portanto não poderá ter valores repetidos nas linhas, ou seja, os valores da coluna id são únicos e não poderão ser repetidos nesta tabela, como exemplifica-se:

tbpessoa		
id	nome	cpf
1	MARIA	88888

2	JOSÉ	77777
3	LAURA	66666
2	MARIA	55555

No exemplo acima o valor 2 é repetido coluna id e o valor MARIA é repetido na coluna nome. Observa-se que a repetição da coluna id é vedada, pois é a coluna que identifica de forma exclusiva e inequívoca os dados do registro(linha ou tupla). Já a repetição na coluna nome pode ser realizada.

Na hipótese de haver duas ou mais tabelas cujos dados se relacionam, a chave estrangeira ou foreign key será imprescindível para a eficiência do relacionamento dos dados das tabelas, uma vez que a chave estrangeira é uma coluna ou combinação de colunas, cujo valores aparecem necessariamente na chave primária de uma outra tabela, ou seja, é um mecanismo que permite a implementação de relacionamentos em um banco de dados relacional, impondo restrições com o objetivo de garantir a execução de operações de alteração no banco de dados de forma adequada e consistente, como exemplifica-se:

tbpessoa		
id	nome	cpf
1	MARIA	88888
2	JOSÉ	77777
3	LAURA	66666

tbveiculo					
id	placa	marca	modelo	ano	idProp
1	AAA	GM	ONIX	2020	1
2	BBB	VW	GOL	2010	1
3	CCC	GM	CORSA	2005	2
4	DDD	FORD	KA	2009	1

Observa-se que as duas tabelas acima detêm a coluna ou atributo id, sendo este a chave primária de cada uma das tabelas. Observa-se ainda que a tabela tbpessoa detém três colunas(id, nome e cpf) e três registros ou tuplas, já a tabela tbveiculo detém seis colunas(id, nome, marca, modelo, ano e idProd) e quatro registros ou tuplas.

A tabela tbveiculo detém a coluna idProp que é uma chave estrangeira, consideradas as seguintes razões:

a) Como regra de negócio é interessante considerar a vida real para estabelecer que uma pessoa poderá ter vários veículos, mas um veículo não poderá ser, no sistema de registro que temos hoje, de várias pessoas. Assim, a tabela tbveiculo terá uma chave estrangeira que permita identificar o proprietário do veículo.

b) Ao cadastrar uma pessoa na tabela tbpessoa os dados poderão ser acessados por meio da chave primaria, que poderá ser facilmente inserida na coluna idProp da tabela tbveiculo, criando um ponto de conexão ou relacionamento entres as tabelas, ou seja, chave primária identificada na chave estrangeira.

c) A coluna idProp do exemplo acima detém quatro linhas, mas apenas dois dados distintos, isto é, a pessoa de id 1 na tabela tbpessoa é a proprietária de três veículos, conforme demonstra a chave estrangeira idProp da tabela tbveiculo.

Observa-se que para as situações em que há necessidade de relacionar registros de uma tabela, várias vezes com registros de outra tabela e vice-versa, surge a necessidade de criação de uma nova tabela, pois essa situação é intitulada como relacionamento N para N, conforme exemplifica-se:

tbpessoa		
id	nome	cpf
1	MARIA	88888
2	JOSÉ	77777
3	LAURA	66666

tbproduto					
id	nome	marca	modelo	descricao	valor
1	televisão	LG	AB55	55 poleg	4000
2	computador	DELL	optplex	I5	6000
3	console	Sony	PS5	Blu-Ray	4300
4	celular	Apple	iphone14	128GB	6500

Ao analisar as duas tabelas/entidades/relações acima nota-se que a inserção de uma chave estrangeira em uma das tabelas causaria uma anomalia de dados, uma vez que, a pessoa de id 1, pode adquirir/comprar o produto de id 1, mas a pessoa de id 2, também poderá adquirir/comprar o mesmo produto. Neste cenário surge a necessidade de criação de uma nova tabela/entidade para relacionar as tabelas tbpessoa e tbproduto, conforme demostra-se:

tbvenda			
id	idPessoa	idProduto	data
1	2	4	01/02/2022
2	2	2	12/02/2022
3	1	4	01/03/2022
4	3	4	15/03/2022
5	1	1	15/03/2022
6	2	3	16/03/2022
7	2	2	22/03/2022

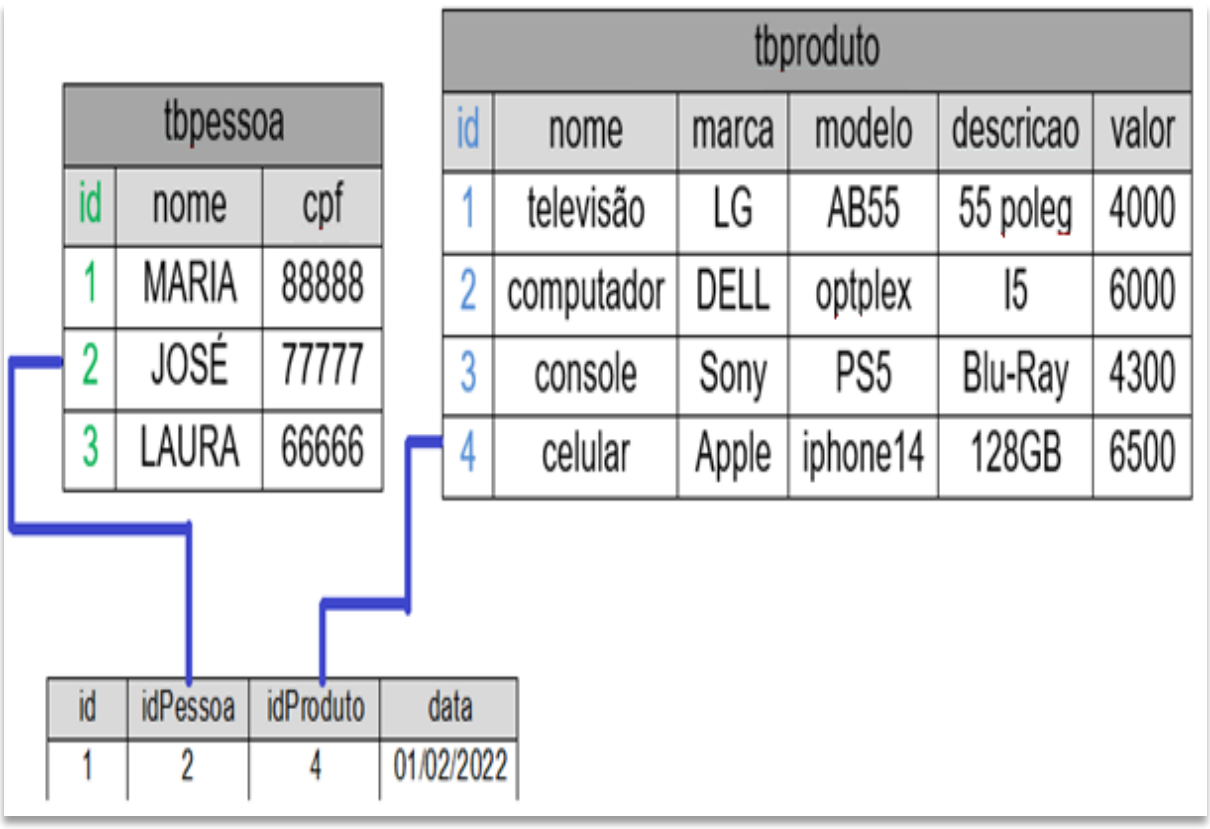
A nova tabela tbvenda pode ser considerada uma entidade associativa, uma vez que, permite o relacionamento entre a tabela tbpessoa, por meio da chave estrangeira idPessoa e a tabela tbproduto, por meio da chave estrangeira idProduto.

Vale salientar que a tabela tbvenda analisada com as tabelas tbpessoa e tbproduto fornece informações como:

- Quantidade de produtos que foram vendidos
- Quantos produtos cada pessoa comprou
- Quais os produtos foram vendidos

Permite entender que:

- Pode ser feito o registro da compra de mais de um produto pela mesma pessoa
- Pode ser feito o registro da compra do “mesmo” produto por pessoas diferentes
- Na imagem abaixo percebe-se que a pessoa de id 2 está associada ao produto de id 4, ou seja, JOSÉ comprou um celular:



2. Codificando o banco de dados

Os exemplos abordados neste material poderão ser codificados por meio da tecnologia/linguagem denominada SQL - Structured Query Language, ou Linguagem de Consulta Estruturada, uma espécie de padrão adotado por um SGBD - Sistema Gerenciador de Banco de Dados, que é um sistema que oferece meios de criação e definição de dados, bem como gerenciamento e manipulação desses dados armazenados. Vale registrar que a linguagem SQL foi desenvolvida nos anos 1970 nos laboratórios da IBM, dentro do projeto System R, sendo que o nome original da tecnologia/linguagem era SEQUEL, acrônimo para Structured English Query Language (Linguagem de Consulta Estruturada em Inglês). Embora a linguagem SQL tenha sido criada pela IBM, rapidamente surgiram algumas "derivações" produzidas por outros desenvolvedores, gerando a necessidade de criar e até mesmo adaptar um padrão para a linguagem. A American National Standards Institute (ANSI), em 1986, e a International Organization for Standardization (ISO), em 1987, criaram o padrão, sendo que em 2003 foi feita uma revisão, gerando a versão SQL:2003.

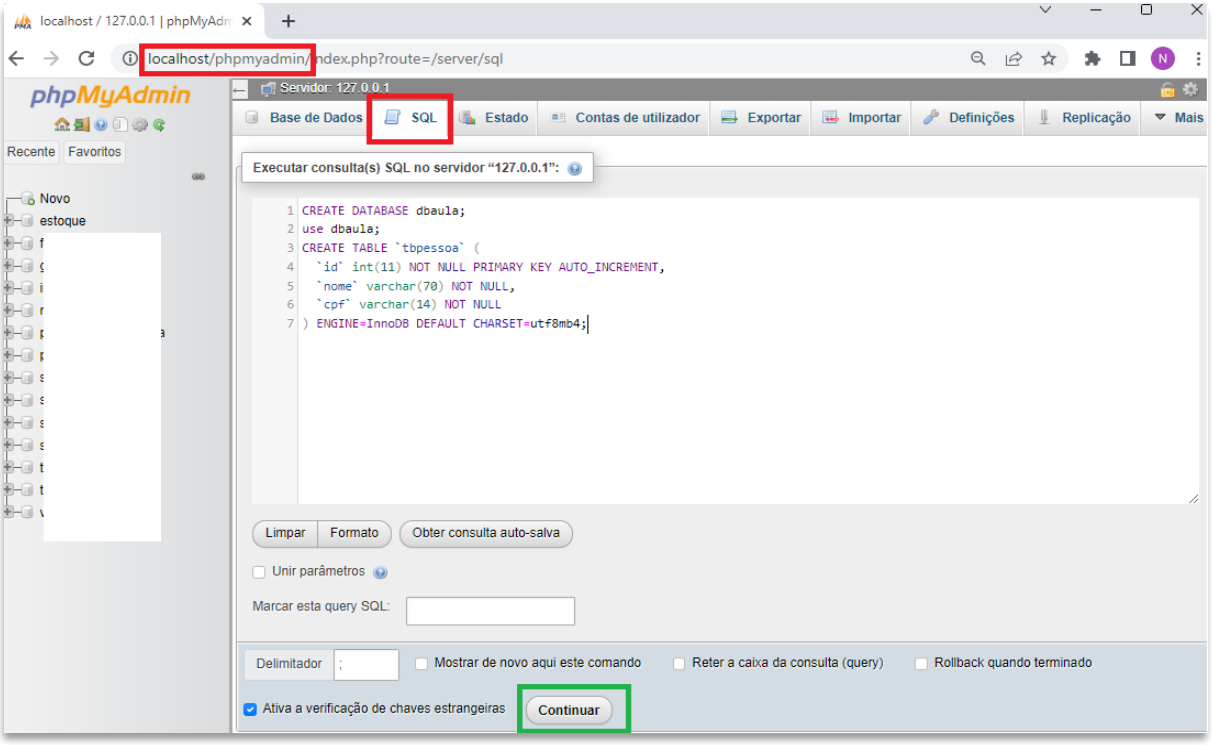
Para codificar os exemplos mencionados neste material, será utilizado o SQL, com o SGBD MariaDB (fork do MySQL) e a interface PHPMyAdmin.

O código SQL para criação do banco de dados chamado dbaula e a primeira tabela chamada tbpessoa é o escrito abaixo:

```
CREATE DATABASE dbaula;
use dbaula;
CREATE TABLE `tbpessoa` (
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `nome` varchar(70) NOT NULL,
  `cpf` varchar(14) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Vale salientar que para executar o código escrito acima, utilizando as tecnologias descritas, é interessante instalar o XAMPP para facilitar o processo, pois o XAMPP é uma distribuição Apache, contendo MySQL, PHP e Perl. Após a instalação do XAMPP é necessário abrir um navegador e digitar o endereço http://localhost/phpmyadmin e selecionar a opção SQL e digitar o código escrito acima e na sequência clicar no botão continuar para que o banco de dados

chamado dbaula com uma tabela chamada tbpessoa sejam criados, conforme exemplo:



O mesmo procedimento será adotado para a criação da tabela tbproduto, com o uso do seguinte código:

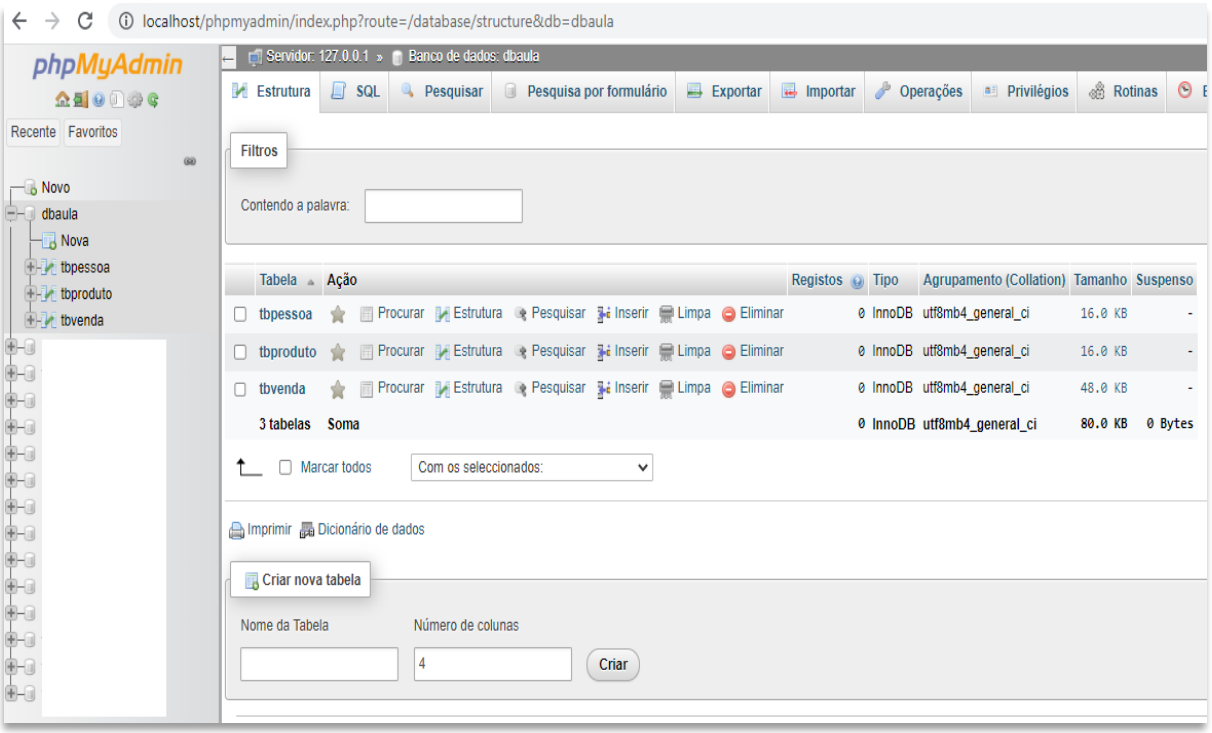
```
use dbaula;
CREATE TABLE `tbproduto` (
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `nome` varchar(80) NOT NULL,
  `marca` varchar(30) NOT NULL,
  `modelo` varchar(50) NOT NULL,
  `descricao` varchar(300) NOT NULL,
  `valor` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Para completar a estrutura do banco dbaula a tabela tbvenda deve ser criada utilizando o código abaixo e o mesmo procedimento das tabelas anteriores:

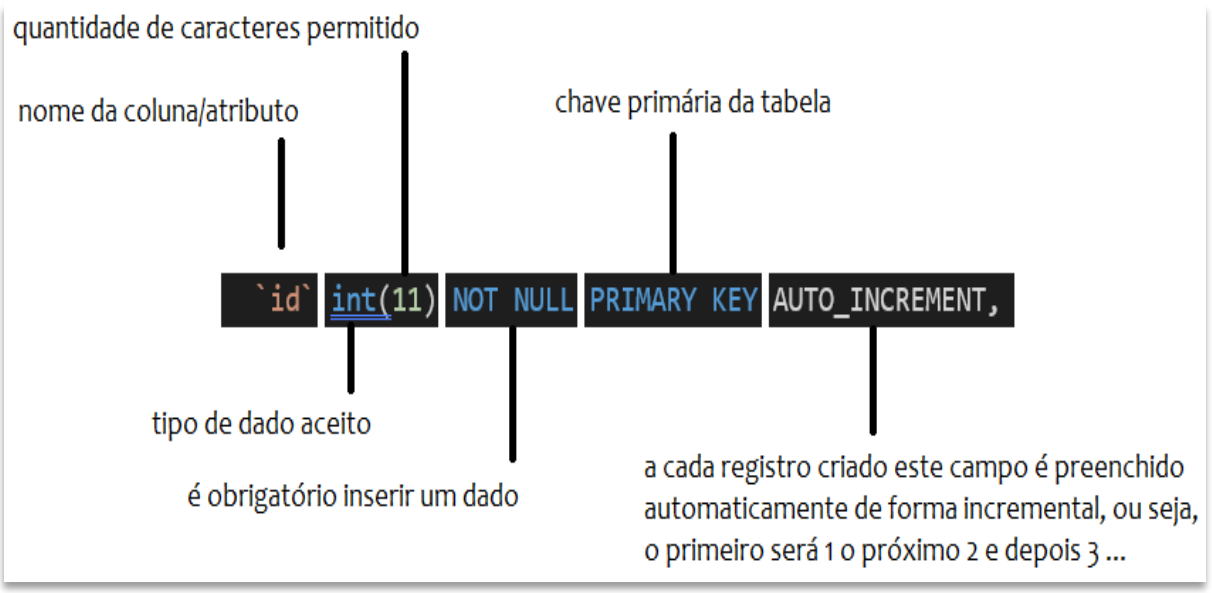
```
use dbaula;
CREATE TABLE `tbvenda` (
  `id` int(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
  `idPessoa` int(11) NOT NULL,
  `idProduto` int(11) NOT NULL,
  `data` timestamp NOT NULL DEFAULT current_timestamp(),
```

```
  CONSTRAINT `fkpessoa` FOREIGN KEY (`idPessoa`) REFERENCES `tbpessoa` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fkproduto` FOREIGN KEY (`idProduto`) REFERENCES `tbproduto` (`id`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

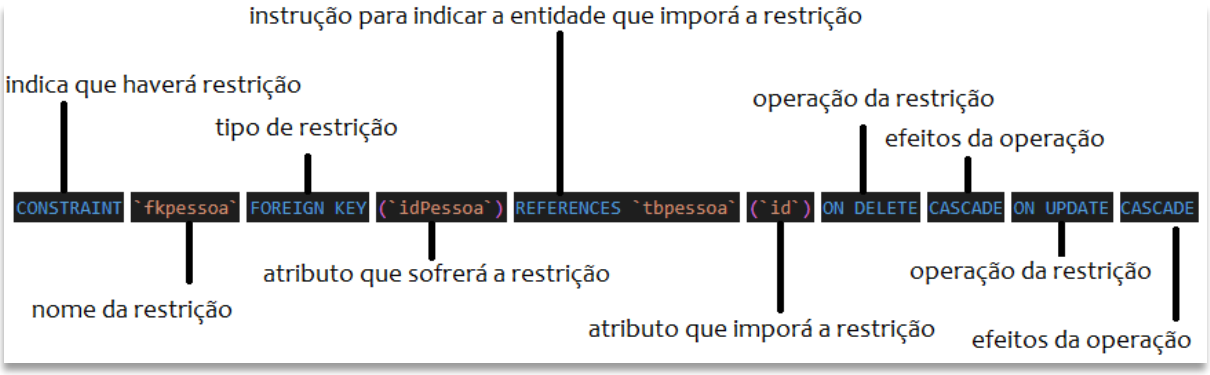
Após a conclusão da codificação do banco, a interface phpMyAdmin apresentará a seguinte estrutura:



Importante explicar a estrutura de criação dos atributos das tabelas acima apresentadas:



Vale ainda explicar a estrutura de criação de uma constrição/restrição no atributo idPessoa da tabela tbvenda acima apresentadas:



3. Conexão do banco de dados com o Java Web

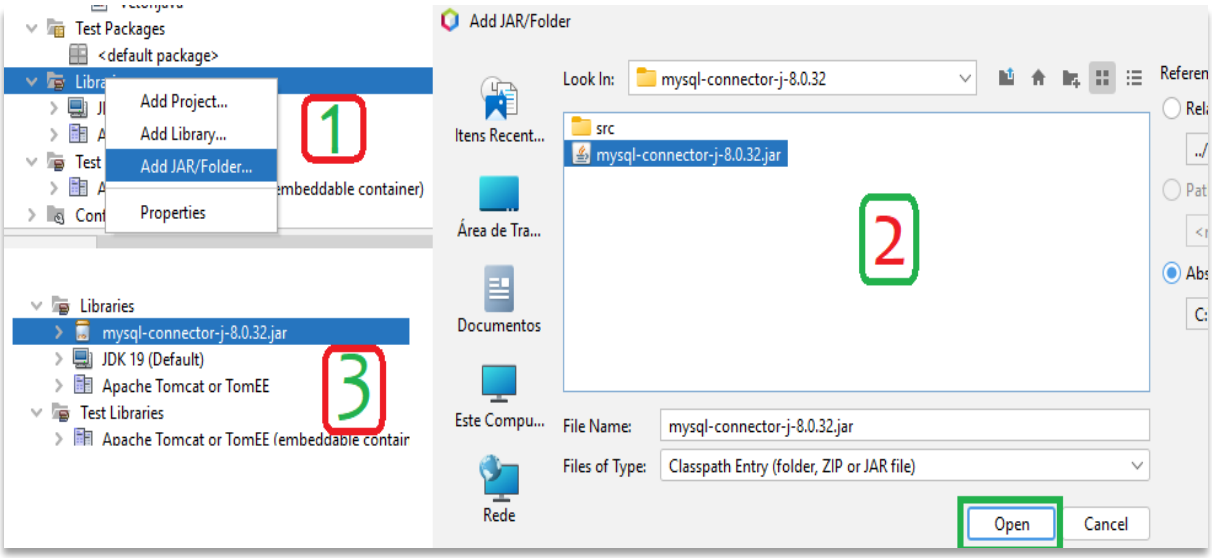
Após a criação do banco de dados é possível realizar a conexão com a linguagem Java, sendo necessário realizar alguns procedimentos e codificação.

O primeiro passo é criar um arquivo chamado testaconexao.jsp para verificar se já há um driver de conexão JDBC no seu projeto, conforme descrito:

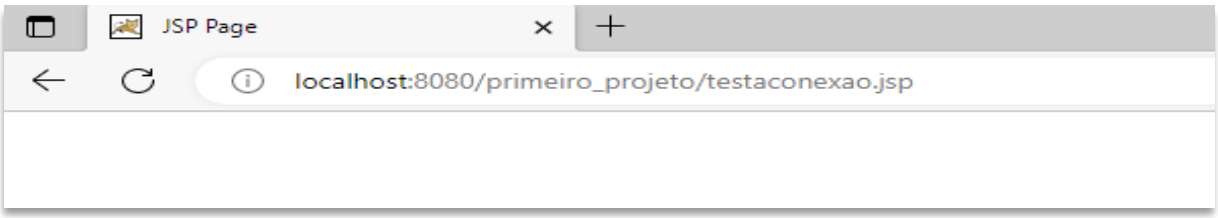
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <%
            try{
               Class.forName("com.mysql.jdbc.Driver");
            }catch(ClassNotFoundException ex){
                out.println("Driver não encontrado.");
            }
        %>
    </body>
</html>
```


Caso a mensagem seja “Driver não encontrado”, no site https://dev.mysql.com/downloads/connector/j/ terá o link para download do connector/J, sendo necessário selecionar a opção Platform Independent e clicar em Download na opção ***Platform Independent (Architecture Independent), ZIP Archive***, conforme demonstra-se:

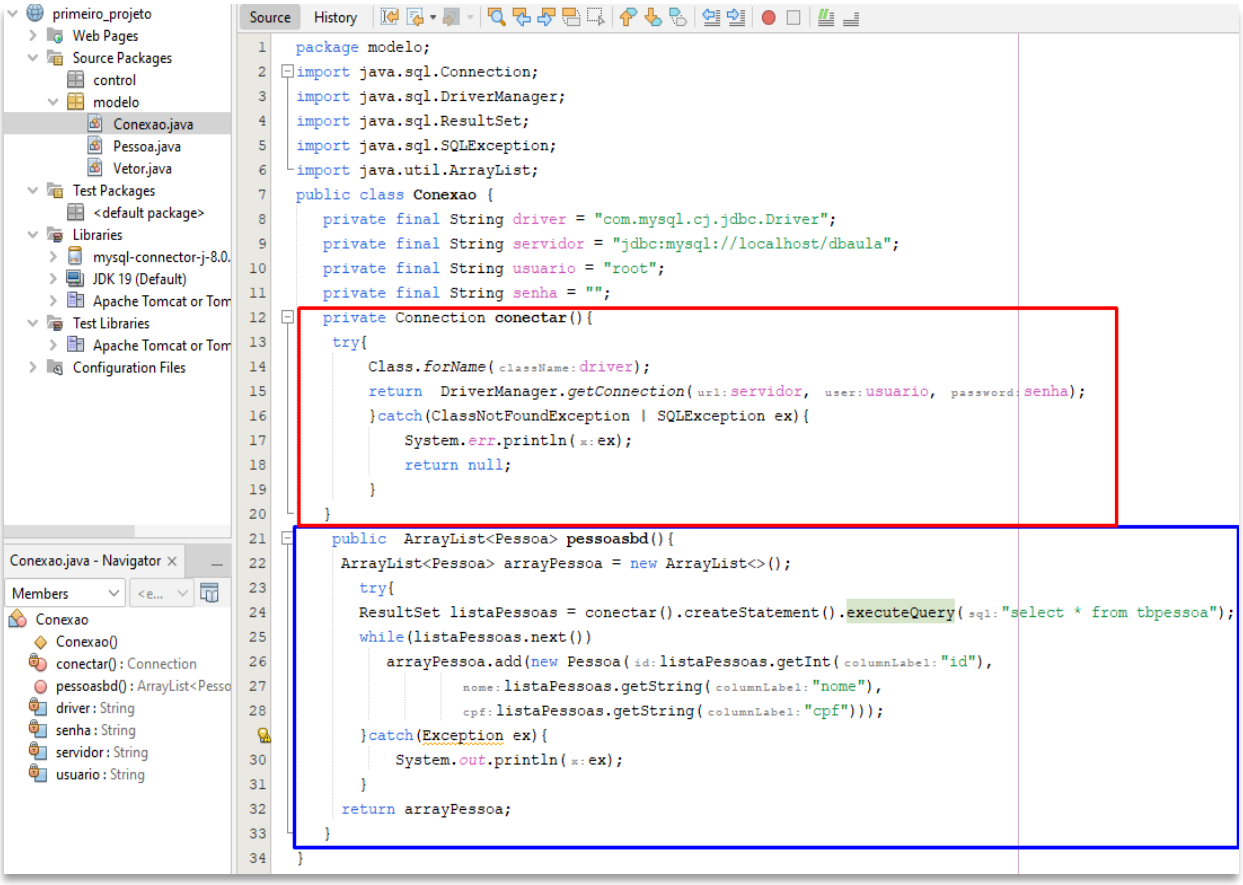
Após o download basta descompactar o arquivo e adicioná-lo no diretório Libraries:



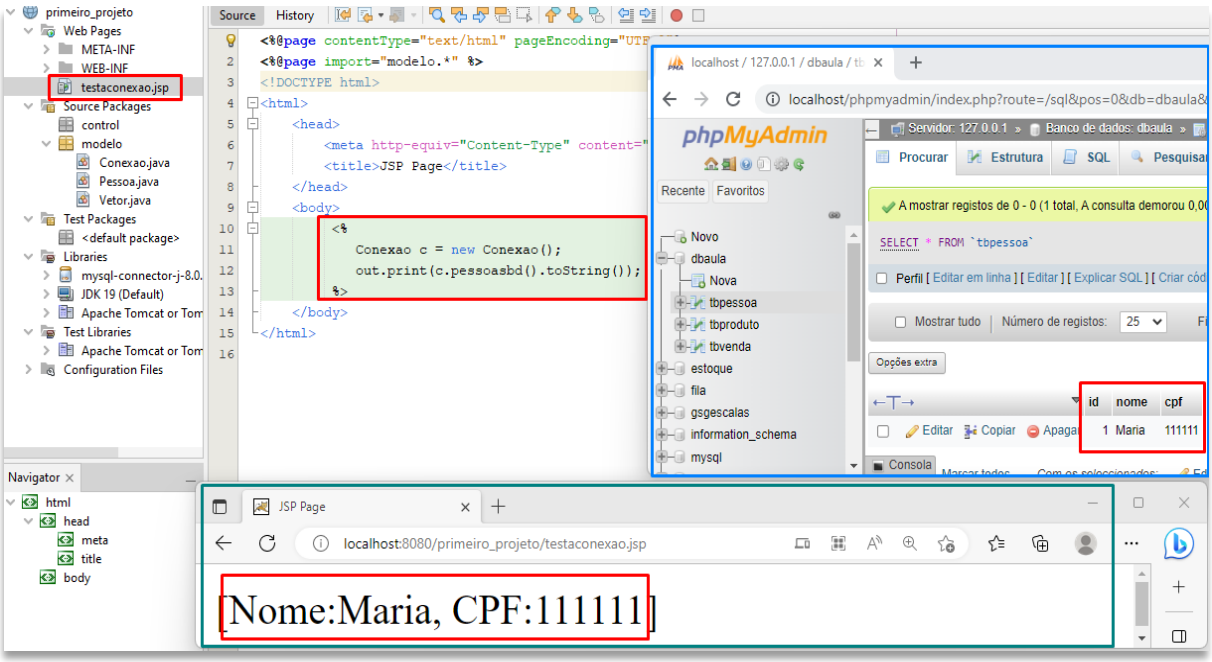
Realizada a execução dos passos acima, ao executar a página testaconexao.jsp não aparecerá mensagem:



Agora o projeto Java está pronto para estabelecer a conexão com o banco de dados dbaula, sendo que um arquivo chamado Conexao.java é utilizado para inserção do código de conexão, bem como o código de consulta aos dados existes na tabela tbpessoa, observando que o mesmo código poderá ser ajustado para a tabela tbproduto e tbvenda.



No arquivo testaconexao.jsp é inserido o código para exibir no navegador os dados constantes no banco de dados dbaula, especificamente na tabela tbpessoa:



```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="modelo.*" %>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <%
         Conexao c = new Conexao();
         out.print(c.pessoasbd().toString());
      %>
   </body>
</html>
```

Vale demonstrar a exibição dos dados em uma tabela HTML inserida no arquivo testaconexão.jsp, considerando mais inserções de registros no banco de dados dbaula.

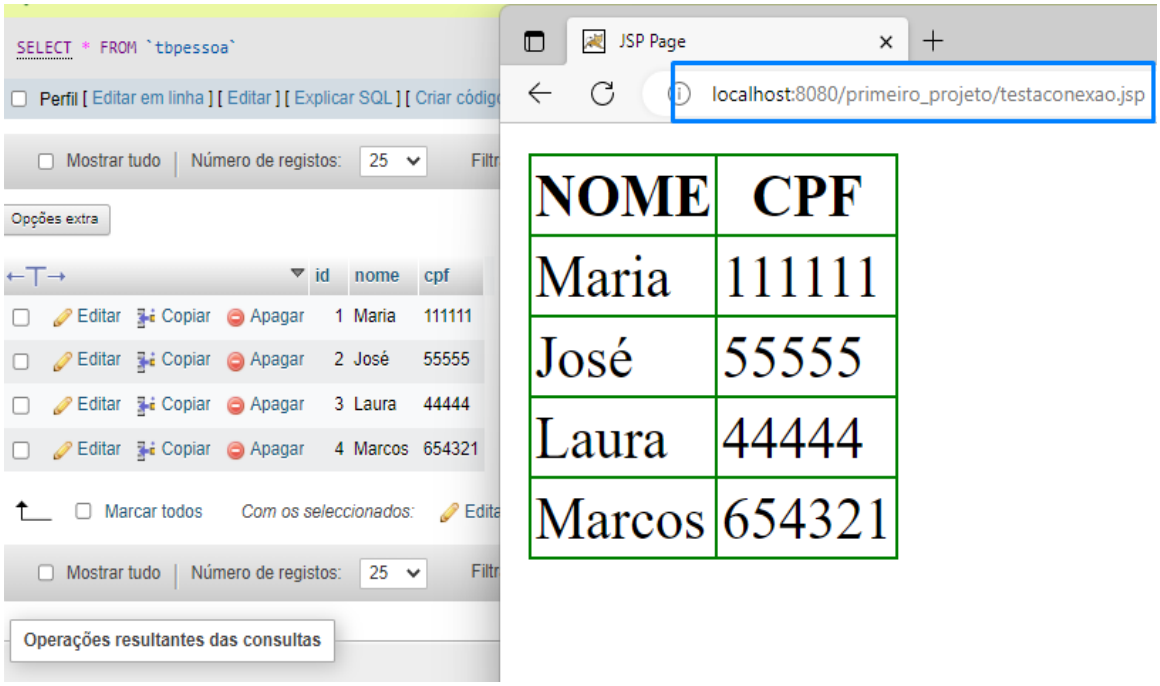
Para inserir dados uma tabela do MySQL a instrução insert é utilizada, com sintaxe genérica INSERT INTO nomeDaTabela (nomeDoAtributo1, nomeDoAtributo2) VALUES(valorDoAtributo1, valorDoAtributo2), que para a tabela tbpessoa utilizada neste material e considerando que o atributo id é autoincremento, uma instrução possível será: `INSERT INTO tbpessoa (`nome`, `cpf`) VALUES ('Marcos' ,'654321');`

Modificando o arquivo testaconexao.jsp para o código abaixo, será possível exibir os dados do banco num formato de tabela HTML:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="modelo.*" %>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
      <style>
         table,th,td {
            border: 1px solid green;
            border-collapse: collapse;
         }
      </style>
   </head>
   <body>
      <%
```

```
        Conexao c = new Conexao();
        int qtd = 0;
        out.print("<table>"
              + "<tr><th>NOME</th>"
              + "<th>CPF</th></tr>");
        while(qtd<c.pessoasbd().size()){
out.print("<tr><td>"+c.pessoasbd().get(qtd).getNome()+"</td><td>"+c.pessoasbd().get(qtd).getC
pf()+"</td></tr>");
            qtd++;
         }
        out.print("</table>");
      %>
  </body>
</html>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="modelo.*" %>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        <style>
            table,th,td {border: 1px solid green;border-collapse: collapse;}
        </style>
    </head>
    <body>
        <%
            Conexao c = new Conexao();
            int qtd = 0;
            out.print("<table><tr><th>NOME</th><th>CPF</th></tr>");
            while(qtd<c.pessoasbd().size()){
                out.print("<tr><td>"+c.pessoasbd().get(qtd).getNome()+
                "</td><td>"+c.pessoasbd().get(qtd).getCpf()+"</td></tr>");
                qtd++;
             }
            out.print("</table>");
        %>
    </body>
</html>
```



Como já foi mencionado, utiliza-se a instrução insert do SQL para inserir dados em uma tabela de um banco de dados. Na linguagem Java e considerado o contexto abordado neste material, pode-se incluir um método inserirpessoasbd na classe Conexao do arquivo Conexao.Java, deixando o código com o seguinte teor:

```
package modelo;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
public class Conexao {
   private final String driver = "com.mysql.cj.jdbc.Driver";
   private final String servidor = "jdbc:mysql://localhost/dbaula";
   private final String usuario = "root";
   private final String senha = "";
   private Connection conectar(){
    try{
        Class.forName(driver);
        return  DriverManager.getConnection(servidor, usuario, senha);
        }catch(ClassNotFoundException | SQLException ex){
            System.err.println(ex);
            return null;
        }
   }
    public  ArrayList<Pessoa> pessoasbd(){
     ArrayList<Pessoa> arrayPessoa = new ArrayList<>();
```

```
      try{
      ResultSet listaPessoas = conectar().createStatement().executeQuery("select *
from tbpessoa");
      while(listaPessoas.next())
        arrayPessoa.add(new Pessoa(listaPessoas.getInt("id"),
              listaPessoas.getString("nome"),
              listaPessoas.getString("cpf")));
      }catch(Exception ex){
          System.out.println(ex);
      }
    return arrayPessoa;
  }
```

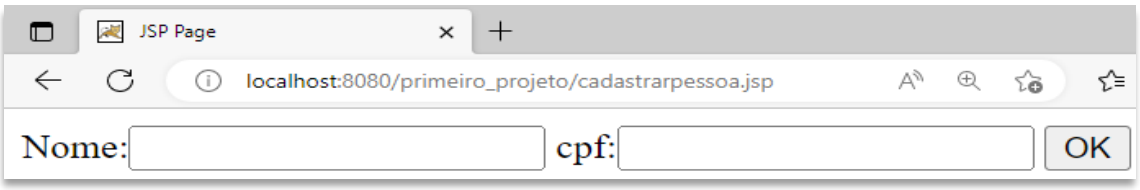
```
   public  ArrayList<Pessoa> inserirpessoasbd(String nome, String cpf){
    ArrayList<Pessoa> arrayPessoa = new ArrayList<>();
      try{
          PreparedStatement inserirPessoa = conectar().prepareStatement("insert into
tbpessoa(nome,cpf) values('"+nome+"','"+cpf+"')");
          inserirPessoa.execute();
          conectar().commit();
      }catch(Exception ex){
          System.out.println(ex);
      }
    return arrayPessoa;
  }
```

```
}
```

Após a criação do arquivo acima, um arquivo chamado cadastrarpessoa.jsp que será a tela de cadastro terá o seguinte código:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="modelo.*" %>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <form action="enviarbd.jsp" method="POST">
         Nome:<input type="text" name="n">
         CPF:<input type="text" name="c">
         <button type="submit" name="ok" value="1">OK</button>
      </form>
   </body>
</html>
```

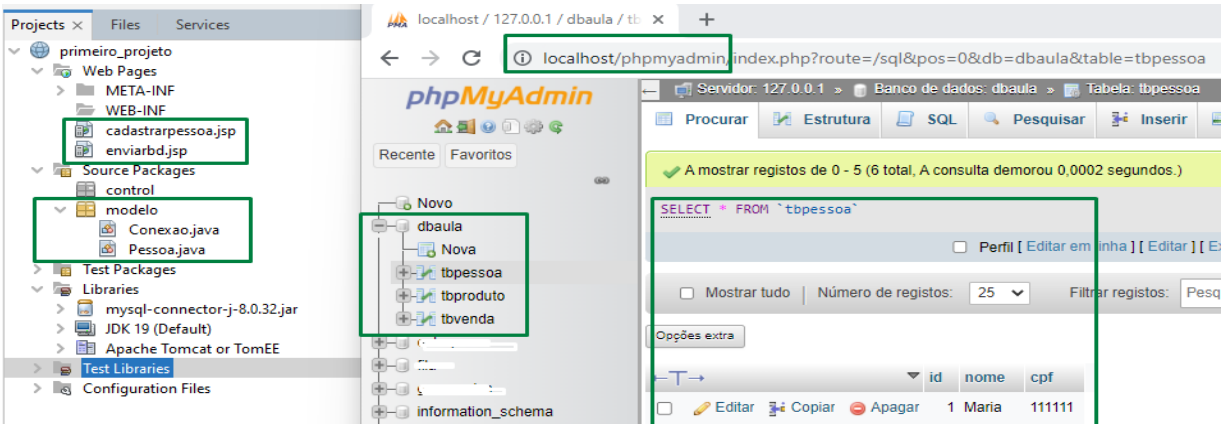
O código acima gera a seguinte tela:



Ao preencher os campos e clicar no botão OK, o arquivo enviarbd.jsp será executado, sendo que o conteúdo do arquivo é o seguinte:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="modelo.*" %>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <%
         Conexao c = new Conexao();
          if(request.getParameter("n")!=null && request.getParameter("c") !=null){
            c.inserirpessoasbd(request.getParameter("n"),request.getParameter("c"));
            out.print("<h2>Gravado no banco de dados.</h2>");
          }
          response.setHeader("Refresh","3;URL=cadastrarpessoa.jsp");
       %>
   </body>
</html>
```

Vale registrar que a estrutura de arquivos no NetBeans terá os arquivos cadastrarpessoa.jsp e enviarbd.jsp, no diretório Web Pages, bem como terá os arquivos Conexao.java e Pessoa.Java no pacote modelo que está no diretório Source Packages do projeto, conforme a seguinte imagem:



A imagem demostra a estrutura de diretórios e arquivos do Projeto no NetBeans, bem como o banco de dados na interface phpmyadmin que permitirá a manipulação e gestão do banco de dados dbaula, permitindo ainda confirmar se os registros foram realizados.

Referências

- https://www.oracle.com/br/database/what-is-a-relational-database/
- https://www.gsigma.ufsc.br/~popov/aulas/bd1/abordagem_relacional.html
- https://www.apachefriends.org/pt_br/index.html
- https://materialpublic.imd.ufrn.br/curso/disciplina/3/73/8/2
- https://dev.mysql.com/downloads/connector/j/
- https://www.apachefriends.org/pt_br/index.html
- https://www.oracle.com/java/technologies/jspt.html