



Programação JAVA e JDBC (Java Database Connectivity)

André Flores dos Santos,
Lucas Alberto Schlestein.

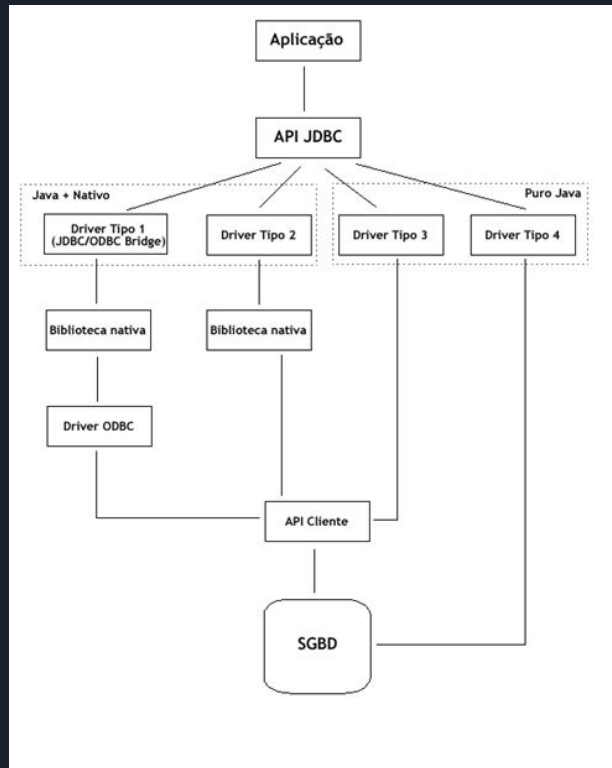


O Que é JDBC

JDBC é similar ao ODBC, e há alguns anos utilizava justamente ODBC para se conectar ao banco de dados. Uma porção de código nativo do Java, era capaz de “conversar” com qualquer banco de dados que tivesse um driver ODBC., assim, popularizando o JDBC no mercado.

O JDBC utiliza drivers, os quais são responsáveis pela conexão com o banco, e pela execução das instruções SQL.

Tipos de JDBC





Drivers JDBC

Os drivers JDBC são responsáveis pela conexão e execução das consultas ao mesmo.

No dia-a-dia encontraremos os mesmos em forma de arquivos JAR (Java ARchive) pelo fabricante do banco de dados ou terceiros.

Ex. de Drivers:

MySQL: <https://www.mysql.com/products/connector/>

Firebird: <https://firebirdsql.org/en/jdbc-driver/>

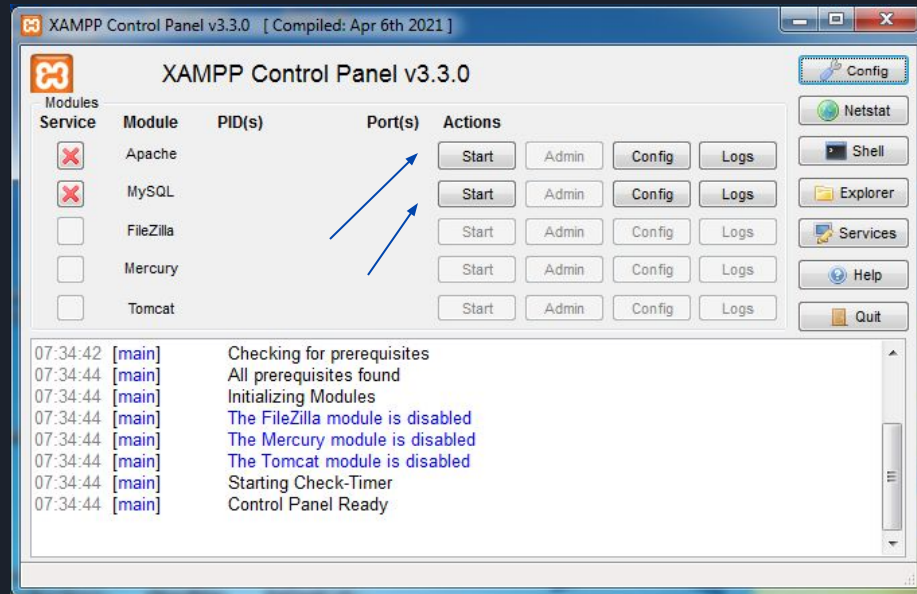
Microsoft SQL Server:

<https://docs.microsoft.com/pt-br/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>

PostgreSQL:

<https://jdbc.postgresql.org/>

Iniciando o MySQL e Criando uma Database





Criando a Database e Inserindo Dados

No navegador acessar: <http://localhost/phpmyadmin/> ou outro console (MySQL Workbench) por exemplo

Comando SQL para criação do banco e das tabelas.

```
create database reuniao;  
use reuniao;  
CREATE TABLE Pessoa (  
    ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(40),  
    email VARCHAR(40),  
    cargo VARCHAR(30)  
);  
  
INSERT INTO pessoa values ('NULL', 'João Silva', 'joao@gmail.com', 'Aluno');  
INSERT INTO pessoa values ('NULL', 'Paulo Vargas', 'pauloVargas@ufn.edu.br', 'Professor');  
SELECT * FROM pessoa;
```

Criando a Database e Inserindo Dados

The screenshot shows the phpMyAdmin web interface in a browser. The address bar shows the URL: `localhost/phpmyadmin/index.php?route=/table/sql&db=reuniao2&table=pessoa`. The left sidebar shows the database structure, with 'reuniao2' selected and 'pessoa' table highlighted. The main panel displays the 'Executar consulta(s) SQL na tabela reuniao2.pessoa:' section. The SQL editor contains the following code:

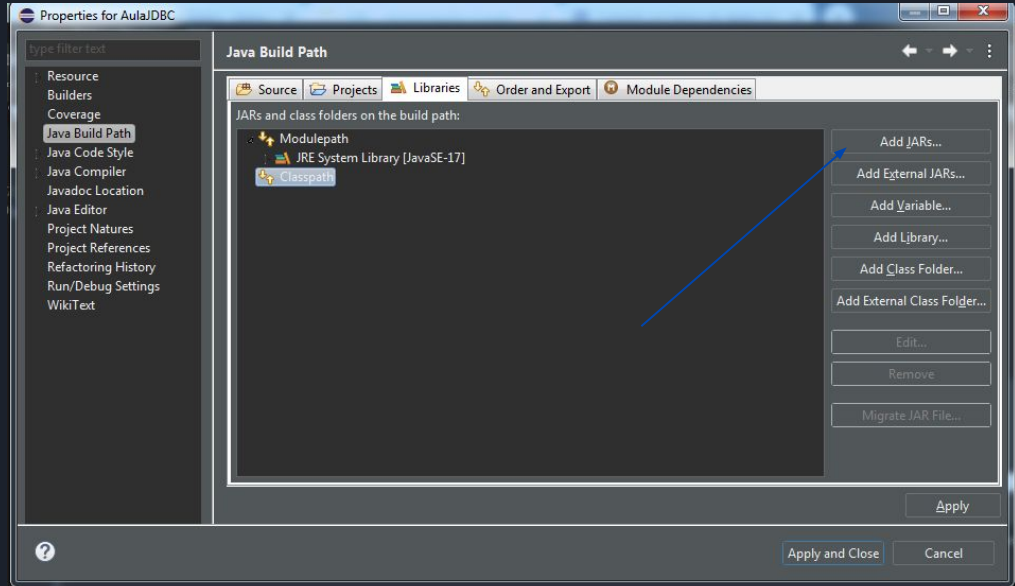
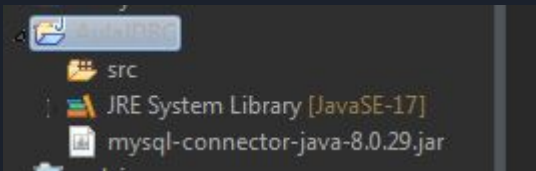
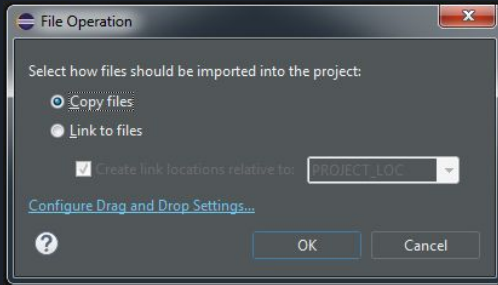
```
1 create database reuniao;
2 use reuniao;
3 CREATE TABLE Pessoa (
4     ID int NOT NULL AUTO_INCREMENT PRIMARY KEY,
5     nome VARCHAR(40),
6     email VARCHAR(40),
7     cargo VARCHAR(30)
8 );
9 INSERT INTO pessoa values ('NULL', 'João Silva', 'joao@gmail.com', 'Aluno');
10 INSERT INTO pessoa values ('NULL', 'Paulo Vargas', 'pauloVargas@ufn.edu.br', 'Professor');
11 SELECT * FROM pessoa;
```

Below the editor, there are buttons for 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', 'Limpar', 'Formato', and 'Obter consulta auto-salva'. A checkbox 'Unir parâmetros' is also present. The 'Marcar esta query SQL:' field is empty. At the bottom, there are checkboxes for 'Delimitador', 'Mostrar de novo aqui este comando', 'Reter a caixa da consulta (query)' (checked), and 'Rollback quando terminado'. A 'Continuar' button is at the bottom right. A blue arrow points to this button.

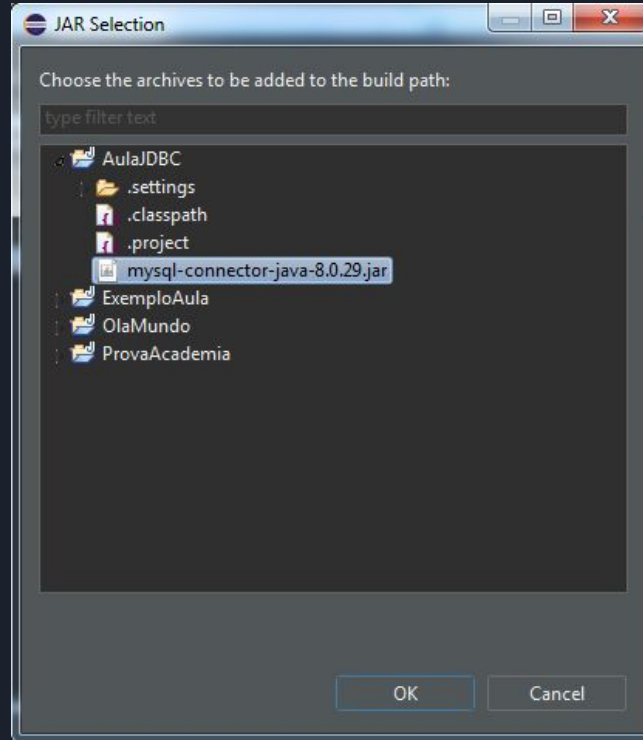
The bottom status bar shows the system clock as 08:20 on 10/05/2022.

Adicionando o Driver ao Projeto

Arrastar ou copiar o .JAR para o Projeto



Adicionando o Driver ao Projeto mysql-connector-java-8.0.30.jar



Conectando Uma Aplicação Java ao BD Banco MySQL

Responsável pela “Tradução” JAVA x BD

```
1 import java.sql.Connection;
8
9 public class Principal {
10
11     public static void main(String[] args) {
12         final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados
13         final String db_query = "SELECT * FROM pessoa";// query a ser executada pelo banco de dados
14         final String db_user = "root";//usuário do SGBD
15         final String db_password = "";//senha do SGBD
16         Pessoa[] p;
17         int resultSetRows = 0;
18         System.out.println("Iniciando conexão ao DB");
19         //utilizando try com recursos para conexão e tratamento de exceções
20         try (Connection c = DriverManager.getConnection(db_url, db_user, db_password);
21             Statement statement = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY); //con
22             ResultSet resultSet = statement.executeQuery(db_query)) { //executa a query configurada
23             System.out.println("Conectado ao MySql");
```

Endereço do BD

Query a ser executada

Retorna as linhas conforme
query

Conectando Uma Aplicação Java ao BD

RDBMS	Formato de URL de banco de dados
MySQL	<code>jdbc:mysql:// nomeDoHost: númeroDePorta /nomeDoBancoDeDados</code>
ORACLE	<code>jdbc:oracle:thin:@ nomeDoHost: númeroDePorta :nomeDoBancoDeDados</code>
DB2	<code>jdbc:db2: nomeDoHost: númeroDePorta /nomeDoBancoDeDados</code>
PostgreSQL	<code>jdbc:postgresql:// nomeDoHost: númeroDePorta /nomeDoBancoDeDados</code>
Java DB/Apache Derby	<code>jdbc:derby:nomeDoBancoDados</code> (incorporado) <code>jdbc:derby://nomeDoHost: númeroDaPorta /nomeDoBancoDeDados</code> (rede)
Microsoft SQL Server	<code>jdbc:sqlserver://nomeDoHost: númeroDaPorta ;databaseName=nomeDoBancoDeDados</code>
Sybase	<code>jdbc:sybase:Tds: nomeDoHost: númeroDePorta /nomeDoBancoDeDados</code>

Conectando Uma Aplicação Java ao BD

db_url= endereço + porta + nome do banco

jdbc: é o protocolo;
mysql, é o subprotocolo;

//localhost é o endereço do
servidor (IP ou nome);

3306 é a porta de acesso
reuniao nome da DB

db_user usuário, nesse caso
root

db_password senha, nesse
caso em branco;

```
1 import java.sql.Connection;
2
3 public class Principal {
4
5     public static void main(String[] args) {
6         final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados
7         final String db_query = "SELECT * FROM pessoa";// query a ser executada pelo banco de dados
8         final String db_user = "root";//usuário do SGBD
9         final String db_password = "";//senha do SGBD
10        Pessoa[] p;
11        int resultSetRows = 0;
12        System.out.println("Iniciando conexão ao DB");
13        //utilizando try com recursos para conexão e tratamento de exceções
14        try (Connection c = DriverManager.getConnection(db_url, db_user, db_password);
15             Statement statement = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_
16             ResultSet resultSet = statement.executeQuery(db_query))){//executa a query configurada
17            System.out.println("Conectado ao MySql");
18        }
```

```
Iniciando conexao ao DB
Conectado ao MySql
```

Criando uma *Consulta* ao Banco de Dados Após Conectar:

query: Consulta a ser executada no BD

Ex.: `SELECT * FROM pessoa`

Statement **statement;**
responsável por criar o objeto
que fará as consultas no banco

ResultSet **resultSet;** 'Tabela'
com resultados de nossa
pesquisa;

`executeQuery()` deve ser
utilizado somente para
consultas.

```
1 import java.sql.Connection;
2
3
4
5
6
7
8
9 public class Principal {
10
11     public static void main(String[] args) {
12         final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados
13         final String db_query = "SELECT * FROM pessoa";// query a ser executada pelo banco de dados
14         final String db_user = "root";//usuário do SGBD
15         final String db_password = "";//senha do SGBD
16         Pessoa[] p;
17         int resultSetRows = 0;
18         System.out.println("Iniciando conexão ao DB");
19         //utilizando try com recursos para conexão e tratamento de exceções
20         try (Connection c = DriverManager.getConnection(db_url, db_user, db_password);
21             Statement statement = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_
22             ResultSet resultSet = statement.executeQuery(db_query))){//executa a query configurada
23             System.out.println("Conectado ao MySql");
```



Criando uma *Consulta* ao Banco de Dados

`query`: `SELECT * FROM pessoa`

`resultSet.next()`: avança o cursor do `ResultSet` até percorrer todas suas linhas.

`resultSet.getString(1)`: busca o valor da coluna 1 do `resultSet`, *id* nesse exemplo

As colunas da tabela `pessoa` são: (id, nome, email e cargo).

```
while(resultSet.next()) {  
    System.out.println(resultSet.getString(1)+" "+resultSet.getString(2)+" "+resultSet.getString(3)+" "+resultSet.getString(4));  
}
```



Resultado da *Consulta* ao Banco de Dados

Consulta: `SELECT * FROM pessoa`

```
Iniciando conexao ao DB
Conectado ao MySql
1 Jo?o Silva joao@gmail.com Aluno
2 Paulo Vargas pauloVargas@ufn.edu.br Professor
3 Jo?o Batista joaoBatista@yahoo.com Professor
```

Executando Alterações no Banco de Dados

Inserir um Registro

query: Consulta a ser executada no BD


Ex.: "INSERT INTO pessoa (nome, email, cargo) values ('Alberto Santos', 'santos@yahoo.com', 'Professor')"

Statement **statement**; responsável por criar o objeto que fará as consultas (alterações) no banco

`executeUpdate()` deve ser utilizado para inserções, alterações (update) e remoção (delete),

```
public static void main(String[] args) {
    final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados
    final String db_query = "INSERT INTO pessoa (nome, email, cargo) values ('Alberto Santos', 'santos@yahoo.com', 'Professor')";// query
    final String db_user = "root";//usuário do SGBD
    final String db_password = "";//senha do SGBD

    System.out.println("Iniciando conexao ao DB");
    //utilizando try com recursos para conexão e tratamento de exceções
    try {
        Connection c = DriverManager.getConnection(db_url, db_user, db_password);
        System.out.println("Conectado ao MySql");
        Statement statement = c.createStatement();//configuração de acesso aos cursores
        int linhas = statement.executeUpdate(db_query);
        System.out.println("A query afetou "+linhas);
    }
}
```

Executando *Alterações* no Banco de Dados

Inserir um Registro

`query: "INSERT INTO pessoa (nome, email, cargo) values ('Alberto Santos', 'santos@yahoo.com', 'Professor')"`

Notem que o método retorna um inteiro *linhas*. Esse retorno indica quantas linhas foram afetadas pela nossa “query” de inserção.

Resultado da Execução:

```
Iniciando conexao ao DB
Conectado ao MySql
A query afetou 1
```

Executando Alterações no Banco de Dados

Atualizar um Registro

`query: "UPDATE pessoa SET nome='Paulo Henrique', email='paulo.henri@globo.com', cargo='Aluno' WHERE nome='Alberto Santos'";`

Notem que o método retorna um inteiro *linhas*. Esse retorno indica quantas linhas foram afetadas pela nossa “query” de Update.

```
public static void main(String[] args) {
    final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados
    final String db_query = "UPDATE pessoa SET nome='Paulo Henrique', email='paulo.henri@globo.com', cargo='Aluno' WHERE nome='Alberto Santos'";
    final String db_user = "root";//usuário do SGBD
    final String db_password = "";//senha do SGBD
    Pessoa[] p;
    int resultSetRows = 0;
    System.out.println("Iniciando conexao ao DB");
    //utilizando try com recursos para conexão e tratamento de exceções
    try {
        Connection c = DriverManager.getConnection(db_url, db_user, db_password);
        System.out.println("Conectado ao MySql");
        Statement statement = c.createStatement();//configuração de acesso aos cursores
        int linhas = statement.executeUpdate(db_query);
        System.out.println("A query afetou "+linhas);
    }
```

Iniciando conexao ao DB
Conectado ao MySql
A query afetou 1

Executando Alterações no Banco de Dados

Deletar um Registro

`query: "DELETE FROM pessoa WHERE nome='Paulo Henrique'";`

Notem que o método retorna um inteiro *linhas*. Esse retorno indica quantas linhas foram afetadas pela nossa “query” de deleção.

```
public static void main(String[] args) {  
    final String db_url = "jdbc:mysql://localhost:3306/reuniao";//caminho de acesso ao banco de dados  
    final String db_query = "DELETE FROM pessoa WHERE nome='Paulo Henrique'";// query a ser executada pelo banco de dados  
    final String db_user = "root";//usuário do SGBD  
    final String db_password = "";//senha do SGBD  
    Pessoa[] p;  
    int resultSetRows = 0;  
    System.out.println("Iniciando conexao ao DB");  
    //utilizando try com recursos para conexão e tratamento de exceções  
    try {  
        Connection c = DriverManager.getConnection(db_url, db_user, db_password);  
        System.out.println("Conectado ao MySql");  
        Statement statement = c.createStatement();//configuração de acesso aos cursores  
        int linhas = statement.executeUpdate(db_query);  
        System.out.println("A query afetou "+linhas);  
    }  
}
```

Iniciando conexao ao DB
Conectado ao MySql
A query afetou 1



Fechando a Conexão com o BD:

Ao executar Update, Delete ou Insert é importante fechar a conexão após o uso. Enquanto a conexão não for fechada, outras alterações podem ser feitas, através do método `executeUpdate()`, passando uma nova query com a alteração desejada.

```
try {  
    Connection c = DriverManager.getConnection(db_url, db_user, db_password);  
    System.out.println("Conectado ao MySql");  
    Statement statement = c.createStatement();//configuração de acesso aos cursors  
    int linhas = statement.executeUpdate(db_query);  
    System.out.println("A query afetou "+linhas);  
    statement.close();  
    c.close();|
```

Fechando a Conexão com o BD:

Ao fazer uma busca no banco, ao se utilizar os recursos do try(), não é necessário o fechamento das conexões, pois, existe a propriedade AutoClose que se encarrega dessa operação.

Também é possível configurar o cursor do ResultSet para que o mesmo ir e voltar, na tabela, pois, por padrão ele só vai a frente.

Outra possibilidade tornar o resultSet editável, nesse caso está como “somente leitura”;

```
System.out.println("Iniciando conexao ao DB");
// utilizando try com recursos para conexão e tratamento de exceções
try (Connection c = DriverManager.getConnection(db_url, db_user, db_password);
     Statement statement = c.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
     ResultSet resultSet = statement.executeQuery(db_query);) {
    System.out.println("Conectado ao MySql");

    while (resultSet.next()) {
        System.out.println(resultSet.getString(1) + " " + resultSet.getString(2) + " " + resultSet.getString(3)
                           + " " + resultSet.getString(4));
    }
}
```



Prepared Statement:

Em substituição ao Statement utilizado nos exemplos anteriores, podemos e podemos até dizer, devemos utilizar para o PreparedStatement.

O PreparedStatement funciona de forma similar ao Statement, porém, possibilita que sejam passados parâmetros, anterior à execução da consulta, ou inserção no banco.

Ex. Statement:

```
Statement st = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,  
ResultSet.CONCUR_READ_ONLY);  
ResultSet rs = st.executeQuery(db_query);
```

Ex. PreparedStatement:

```
PreparedStatement ps = conn.prepareStatement("SELECT * FROM pessoa WHERE ID>?"  
);  
ps.setInt(1,1);  
ResultSet rs = ps.executeQuery();
```



Prepared Statement: Exemplo

```
try {  
    Connection c =  
        DriverManager.getConnection("jdbc:mysql://localhost:3306/reuniao", "root", "");  
    PreparedStatement ps = c.prepareStatement("INSERT INTO pessoa  
VALUES(?,?,?,?)"); // equivale a c.prepareStatement("INSERT INTO pessoa VALUES ('',  
'Alberto Roberto', 'alberto@mail.com', 'Professor'");  
    ps.setInt(1, 0);  
    ps.setString(2, "Alberto Roberto");  
    ps.setString(3, "alberto@mail.com");  
    ps.setString(4, "Professor");  
    ps.executeUpdate();  
} catch (Exception e) {  
    System.out.println("Problema: "+e);  
}
```



Exercício

Dada a interface abaixo, escreva um código, o qual faça a implementação da interface abaixo:

```
public interface InterfaceBancoDados {  
    public void conectar(String db_url, String db_user, String db_password);  
    public void desconectar();  
    public void consultar(String db_query);  
    public int inserirAlterarExcluir(String db_query);  
}
```

Por questão de padronização, implemente-a em uma classe cujo nome deverá ser BancoDados.java

Faça, pelo menos 3 inserções em uma tabela no MySql, e após, faça a exibição dos valores inseridos no console.



Exercício Continuação

O esperado é que o fluxo para inserção, por exemplo seja algo do tipo:

Ex.:

```
bd.conectar(" ", " ", " ");
```

```
bd.inserirAlterarExcluir("query");
```

```
bd.consultar("query");
```

Obs.:

- Não é necessário, utilizar parâmetros no PreparedStatement
- Não é necessário, nesse exemplo, carregar as informações obtidas no objeto correlato.
- Preferencialmente, exibir o número de linhas que foram afetadas ao deletar, alterar ou inserir algo no banco.

Estrutura do Banco de Dados desse Exemplo:

Servidor: 127.0.0.1 » Banco de dados: reuniao » Tabela: pessoa

Procurar Estrutura SQL Pesquisar Inserir Exportar Importar Privilégios Operações Rast

Estrutura da tabela Visão de relação(ões)

#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Acções
<input type="checkbox"/>	1 ID	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2 nome	varchar(40)	utf8mb4_general_ci		Sim	NULL			Muda Eliminar Mais
<input type="checkbox"/>	3 email	varchar(40)	utf8mb4_general_ci		Sim	NULL			Muda Eliminar Mais
<input type="checkbox"/>	4 cargo	varchar(30)	utf8mb4_general_ci		Sim	NULL			Muda Eliminar Mais

☐ Marcar todos Com os seleccionados: Procurar Muda Eliminar Primária Único Índice Espacial

Texto Completo Adicionar à(s) coluna(s) central(is) Remover da(s) coluna(s) central(is)

Estrutura do Banco de Dados desse Exemplo:

Servidor de base de dados

- Servidor: 127.0.0.1 via TCP/IP
- Tipo de servidor: MariaDB
- Conexão com o servidor: SSL não está sendo usado ⓘ
- Versão do servidor: 10.4.24-MariaDB - mariadb.org binary distribution
- Versão do protocolo: 10
- Utilizador: root@localhost
- Conjunto de caracteres do servidor: UTF-8 Unicode (utf8mb4)

Apache + MySQL = XAMPP

https://www.apachefriends.org/pt_br/index.html



Estrutura de Strings Para Outros Bancos

MySQL

URL: "jdbc:mysql://localhost:3306/reuniao?user=root&password="

Driver: "com.mysql.cj.jdbc.Driver"

SQL Server

URL: "jdbc:sqlserver://localhost:1433;databaseName=master;user=sa;password=your_password"

Driver: "com.microsoft.sqlserver.jdbc.SQLServerDriver"

<https://github.com/microsoft/mssql-jdbc/blob/main/README.md>



Referências

<https://dev.mysql.com/doc/connector-j/8.0/en/>

<https://www.microsoft.com/en-us/sql-server/developer-get-started/java/windows>

<https://www.mysql.com/downloads/>

https://www.apachefriends.org/pt_br/index.html