

Laboratório de desenvolvimento de software

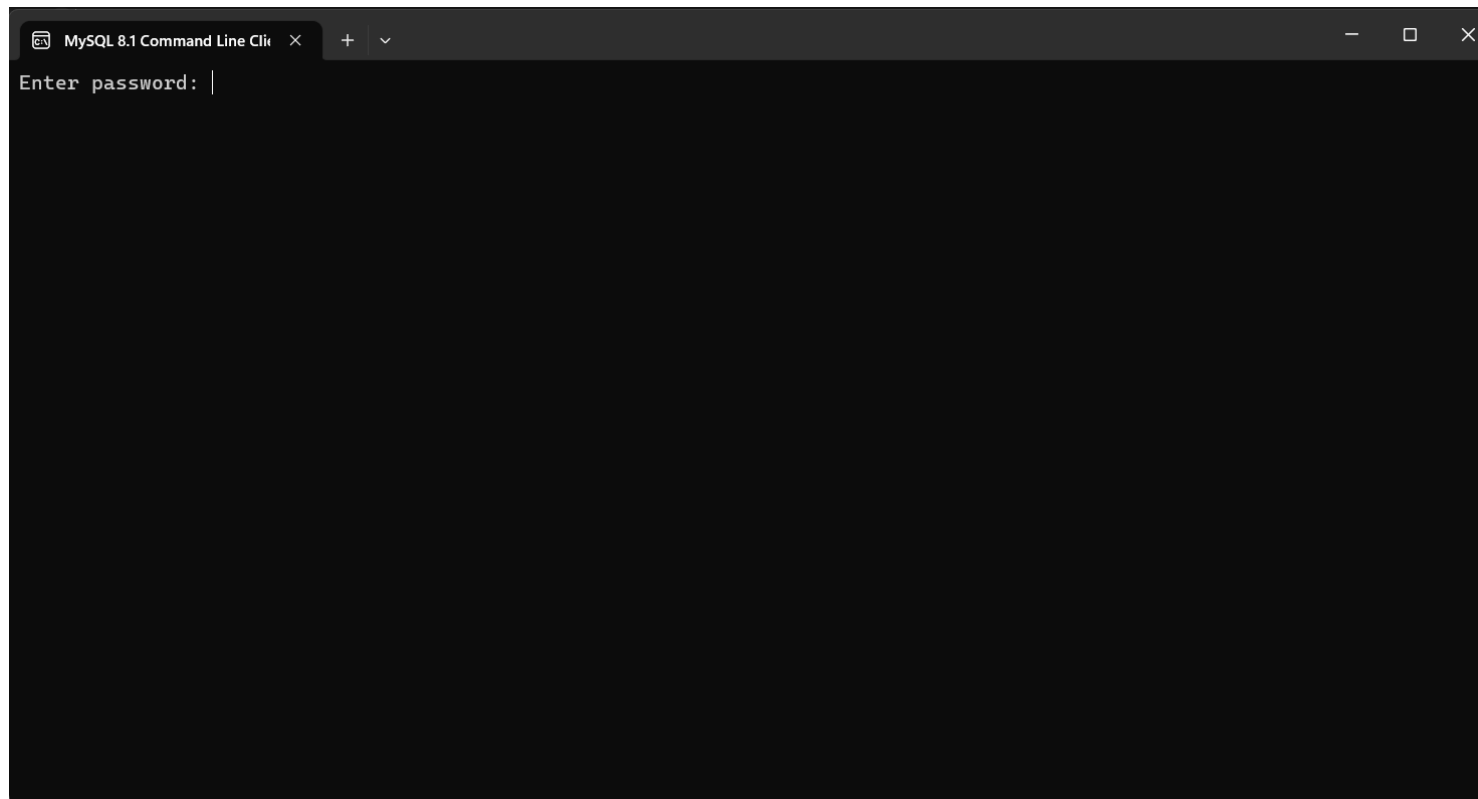
Java Swing + Banco de dados

Java Swing

- Na aula de hoje, vamos trabalhar com Java Swing e Banco de dados
- O banco de dados, utilizaremos o mysql e para utiliza-lo teremos algumas opções..
 - Podemos usar o MySQL nativo, instalado
 - Ou alguma ferramenta para desenvolvedores, como Xampp, Wamp, EasyPHP etc
 - Vocês podem utilizar o que preferir..

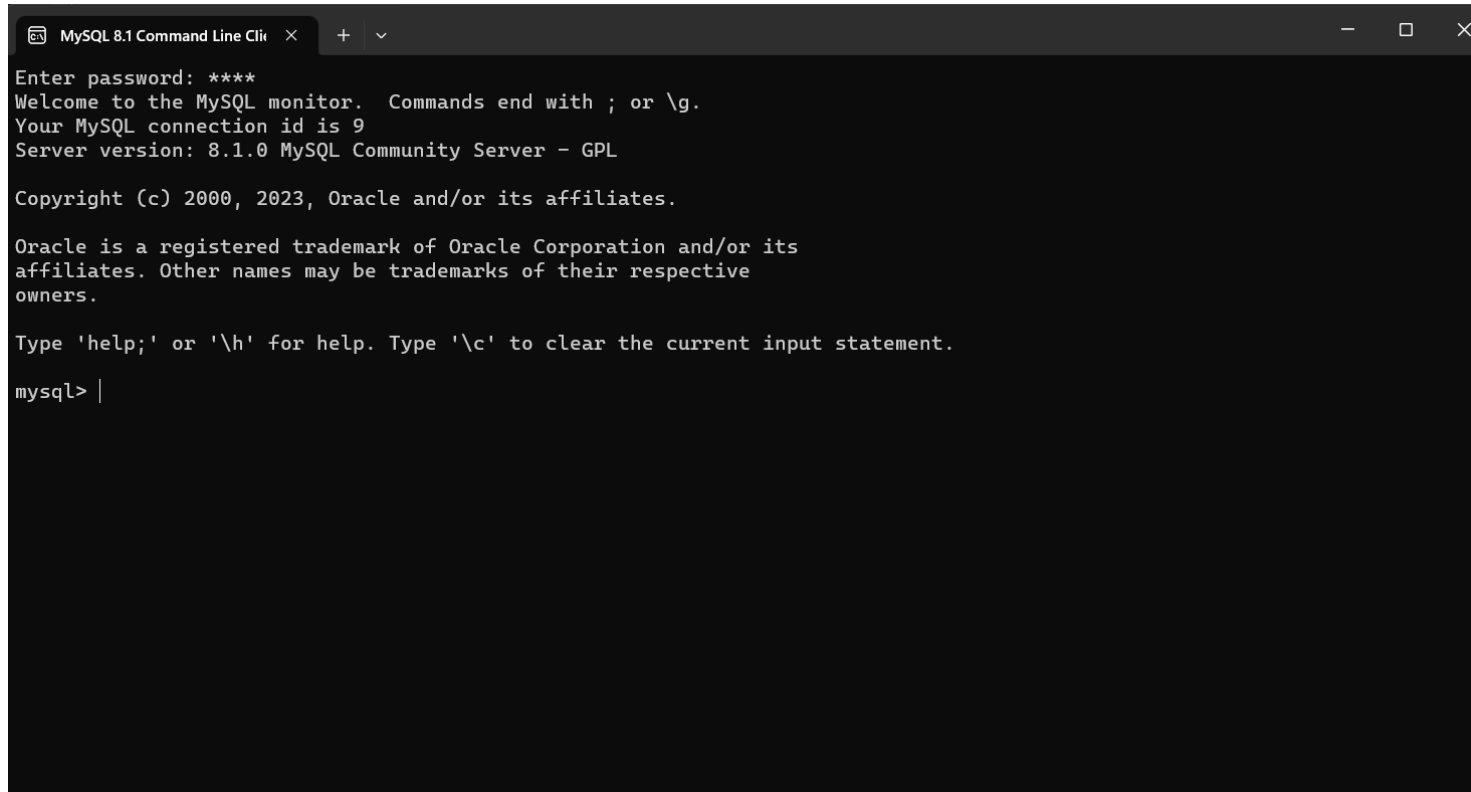
MySQL

- Na aula, será utilizado o MySQL nativo..
- Para isso abra o MySQL Command Line Client



MySQL

- Coloque a senha do root, no meu caso, é 1234 (no laboratório, provavelmente a senha é “laboratorio”)

A screenshot of a terminal window titled "MySQL 8.1 Command Line Client". The window has a dark background and a light-colored text. The text inside the window shows the MySQL login process: "Enter password: ****", "Welcome to the MySQL monitor. Commands end with ; or \g.", "Your MySQL connection id is 9", "Server version: 8.1.0 MySQL Community Server - GPL", "Copyright (c) 2000, 2023, Oracle and/or its affiliates.", "Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.", "Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.", and finally "mysql> |".

```
MySQL 8.1 Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

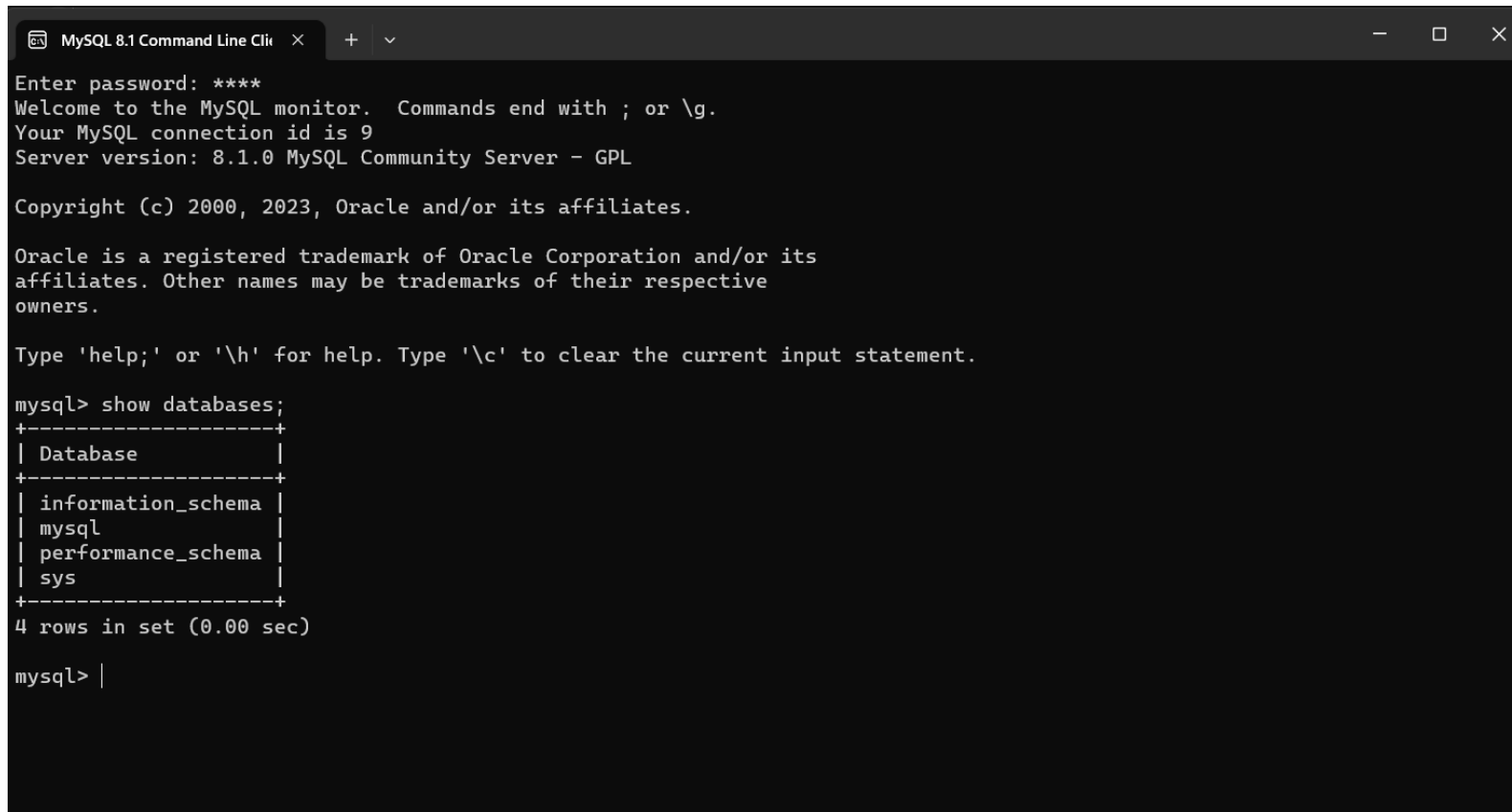
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> |
```

MySQL

- Para apresentar os bancos de dados, digite “show databases;”

A screenshot of the MySQL 8.1 Command Line Client window. The window title is "MySQL 8.1 Command Line Cli". The terminal shows the following text: "Enter password: ****", "Welcome to the MySQL monitor. Commands end with ; or \g.", "Your MySQL connection id is 9", "Server version: 8.1.0 MySQL Community Server - GPL", "Copyright (c) 2000, 2023, Oracle and/or its affiliates.", "Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.", "Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.", "mysql> show databases;", followed by a table with 4 rows: information_schema, mysql, performance_schema, and sys. The table is enclosed in a box with a dashed border. Below the table, it says "4 rows in set (0.00 sec)". The prompt "mysql> |" is shown at the bottom.

```
MySQL 8.1 Command Line Cli x + v
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> |
```

MySQL

- Depois, para criar uma base de dados, digite “create database BDAula01;”

```
mysql> create database BDAula01;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| bdaula01 |  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
5 rows in set (0.00 sec)
```

MySQL

- E depois, precisaremos selecionar a base de dados criada para inserirmos uma tabela nova:

```
mysql> use bdaula01;  
Database changed
```

- E vamos criar uma tabela chamada Pessoa:

```
mysql> CREATE TABLE pessoa (  
-> id smallint AUTO_INCREMENT PRIMARY KEY,  
-> nome Varchar(50) NOT NULL,  
-> sexo Varchar(1) NOT NULL,  
-> idioma Varchar(10) NOT NULL  
-> );  
Query OK, 0 rows affected (0.01 sec)
```

MySQL

- E para ver a tabela, usamos o comando “desc NOME_DA_TABELA”:

```
mysql> desc pessoa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | smallint      | NO   | PRI | NULL    | auto_increment |
| nome  | varchar(50)   | NO   |     | NULL    |                |
| sexo  | varchar(1)    | NO   |     | NULL    |                |
| idioma | varchar(10)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```


MySQL

- E também podemos usar o comando “show tables;” para ver todas as tabelas do banco de dados:

```
mysql> show tables;  
+-----+  
| Tables_in_bdaula01 |  
+-----+  
| pessoa              |  
+-----+  
1 row in set (0.00 sec)
```

MySQL

- E para inserir dados, usamos o comando Insert

```
mysql> INSERT INTO pessoa (nome, sexo, idioma)
      -> VALUES ("Ricardo", "M", "Português");
Query OK, 1 row affected (0.01 sec)
```

- E vamos consultar a inserção:

```
mysql> select * from pessoa;
+----+-----+-----+-----+
| id | nome   | sexo | idioma   |
+----+-----+-----+-----+
|  1 | Ricardo | M    | Português |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

MySQL

- Podemos inserir mais de um dado na mesmo insert:

```
mysql> INSERT INTO pessoa (nome, sexo, idioma)
-> VALUES
-> ("Gerald", "M", "Inglês"),
-> ("William", "M", "Inglês"),
-> ("Umberto", "M", "Espanhol"),
-> ("Jostein", "M", "Alemão"),
-> ("Stephen", "M", "Holandês");
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

MySQL

```
mysql> select * from pessoa;
```

id	nome	sexo	idioma
1	Ricardo	M	Português
2	Gerald	M	Inglês
3	William	M	Inglês
4	Umberto	M	Espanhol
5	Jostein	M	Alemão
6	Stephen	M	Holandês

```
6 rows in set (0.00 sec)
```

MySQL

- Atividades:
- Exercício 1: Criar um Banco de Dados
 - Crie um banco de dados chamado escola.

MySQL

- Atividades:
- Exercício 1: Criar um Banco de Dados
 - Crie um banco de dados chamado escola.

```
mysql> CREATE DATABASE escola;  
Query OK, 1 row affected (0.04 sec)
```

MySQL

- Atividades:
- Exercício 2: Usar o Banco de Dados Criado
 - Selecione o banco de dados escola para usá-lo.

```
mysql> USE escola;  
Database changed
```

MySQL

- Atividades:
 - Exercício 3: Criar uma Tabela de Alunos
 - Crie uma tabela chamada alunos com os seguintes campos:
 - id (chave primária, auto-incremento)
 - nome (texto, tamanho máximo de 50 caracteres)
 - idade (inteiro)
 - curso (texto, tamanho máximo de 50 caracteres)

MySQL

- Atividades:
- Exercício 3: Criar uma Tabela de Alunos

```
mysql> CREATE TABLE alunos (  
->     id INT AUTO_INCREMENT PRIMARY KEY,  
->     nome VARCHAR(50),  
->     idade INT,  
->     curso VARCHAR(50)  
-> );  
Query OK, 0 rows affected (0.01 sec)
```

MySQL

- Atividades:
 - Exercício 4: Inserir Alunos na Tabela
 - Insira cinco registros de alunos na tabela alunos.

```
mysql> INSERT INTO alunos (nome, idade, curso)
-> VALUES
->      ('João', 20, 'Matemática'),
->      ('Maria', 22, 'História'),
->      ('Pedro', 21, 'Ciência da Computação'),
->      ('Ana', 19, 'Biologia'),
->      ('Carlos', 23, 'Economia');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

MySQL

- Atividades:
- Exercício 5: Criar uma Tabela de Professores
 - Crie uma tabela chamada professores com os seguintes campos:
 - id (chave primária, auto-incremento)
 - nome (texto, tamanho máximo de 50 caracteres)
 - idade (inteiro)
 - disciplina (texto, tamanho máximo de 50 caracteres)

MySQL

- Atividades:
- Exercício 6: Inserir Professores na Tabela
 - Insira três registros de professores na tabela professores.

MySQL

- Atividades:
- Exercício 7: Criar uma Tabela de Matrículas
 - Crie uma tabela chamada matriculas com os seguintes campos:
 - id (chave primária, auto-incremento)
 - id_aluno (chave estrangeira referenciando a tabela alunos)
 - id_professor (chave estrangeira referenciando a tabela professores)
 - data_matricula (data)

MySQL

- Atividades:
- Exercício 8: Inserir Matrículas na Tabela
 - Insira cinco registros de matrículas, relacionando alunos e professores fictícios.

MySQL

- Atividades:
- Exercício 9: Consulta de Alunos por Curso
 - Crie uma consulta SQL que liste todos os alunos e seus cursos.

MySQL

- Atividades:
- Exercício 10: Consulta de Professores por Disciplina
 - Crie uma consulta SQL que liste todos os professores e as disciplinas que eles ministram.

MySQL

- Atividades:
- Exercício 5: Criar uma Tabela de Professores

```
mysql> CREATE TABLE professores (  
->     id INT AUTO_INCREMENT PRIMARY KEY,  
->     nome VARCHAR(50),  
->     idade INT,  
->     disciplina VARCHAR(50)  
-> );  
Query OK, 0 rows affected (0.06 sec)
```

MySQL

- Atividades:
- Exercício 6: Inserir Professores na Tabela
 - Insira três registros de professores na tabela professores.

```
mysql> INSERT INTO professores (nome, idade, disciplina)
-> VALUES
->      ('Prof. Mario', 35, 'Matemática'),
->      ('Prof. Augusto', 40, 'História'),
->      ('Prof. Ricardo', 38, 'Sistemas de Informação');
Query OK, 3 rows affected (0.05 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

MySQL

- Atividades:
- Exercício 7: Criar uma Tabela de Matrículas

```
mysql> CREATE TABLE matriculas (  
->     id INT AUTO_INCREMENT PRIMARY KEY,  
->     id_aluno INT,  
->     id_professor INT,  
->     data_matricula DATE,  
->     FOREIGN KEY (id_aluno) REFERENCES alunos(id),  
->     FOREIGN KEY (id_professor) REFERENCES professores(id)  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

MySQL

- Atividades:
- Exercício 8: Inserir Matrículas na Tabela

```
mysql> INSERT INTO matriculas (id_aluno, id_professor, data_matricula)
-> VALUES
->      (1, 1, '2023-01-15'),
->      (2, 2, '2023-02-20'),
->      (3, 3, '2023-03-10'),
->      (4, 1, '2023-04-05'),
->      (5, 2, '2023-05-12');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

MySQL

- Atividades:
- Exercício 9: Consulta de Alunos por Curso
- Crie uma consulta SQL que liste todos os alunos e seus cursos.

```
mysql> SELECT nome, curso FROM alunos;
```

nome	curso
João	Matemática
Maria	História
Pedro	Ciência da Computação
Ana	Biologia
Carlos	Economia

```
5 rows in set (0.00 sec)
```

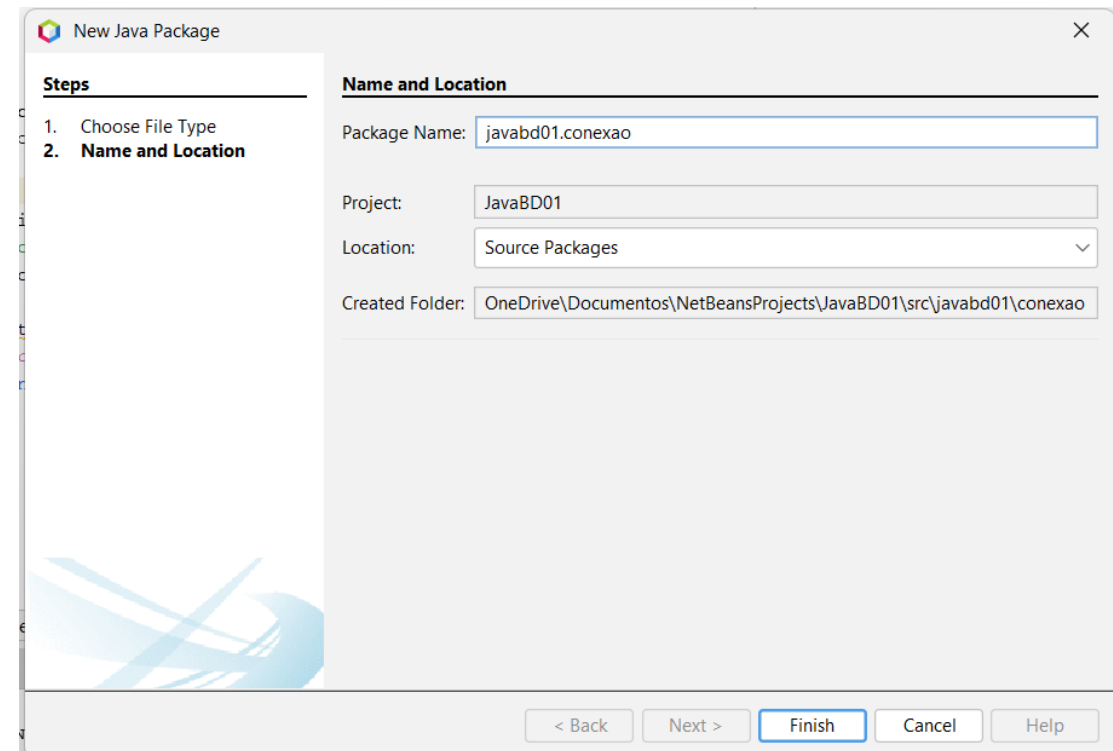
MySQL

- Atividades:
- Exercício 10: Consulta de Professores por Disciplina
- Crie uma consulta SQL que liste todos os professores e as disciplinas que eles ministram.

```
mysql> SELECT nome, disciplina FROM professores;
+-----+-----+
| nome          | disciplina          |
+-----+-----+
| Prof. Mario   | Matemática         |
| Prof. Augusto | História            |
| Prof. Ricardo | Sistemas de Informação |
+-----+-----+
3 rows in set (0.00 sec)
```

Projeto com banco MySQL

- Crie um projeto usando NetBeans -> Java with Ant -> Java Application.
- Crie um pacote chamado “conexao” e dentro deste pacote, crie uma classe chamada “Conexao”

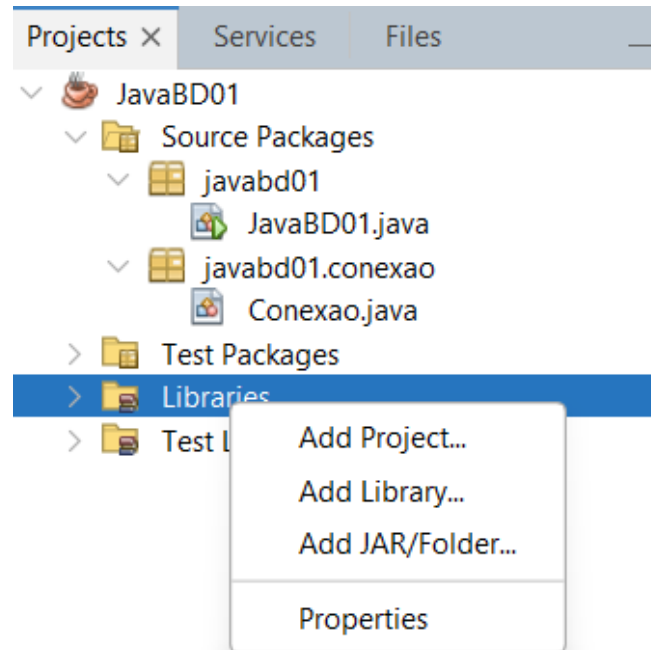


Projeto com banco MySQL

- Clique com o botão direito no nome do projeto
- E clique em “Open in System”
 - Abrirá a pasta do projeto
 - Dentro, crie uma pasta “drivers”
 - E baixe o zip do Agenda e coloque os arquivos lá

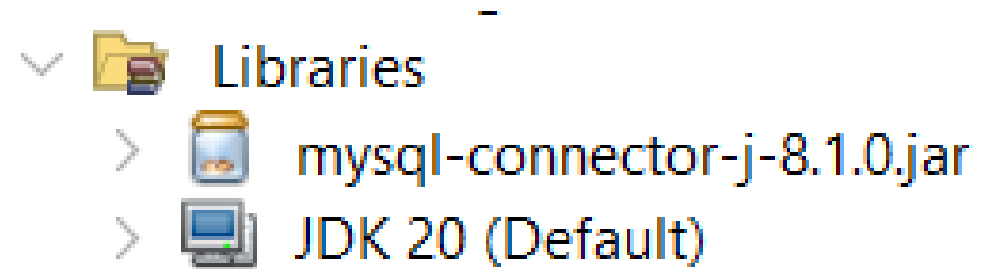
Projeto com banco MySQL

- Agora, clique com o botão direito em “Libraries”
- Selecione Add JAR/Folder



Projeto com banco MySQL

- Agora, clique com o botão direito em “Libraries”
 - Selecione Add JAR/Folder
 - Selecione o arquivo jar



Projeto com banco MySQL

- Agora na classe Conexao, vamos programar para conectar-se ao banco:

```
public class Conexao {  
    public Connection getConexao() {  
        try{  
  
            Connection conn = DriverManager.getConnection(  
                url: "jdbc:mysql://localhost:3306/bdaula01?useTimezone=true&serverTimezone=UTC",  
                user: "root", password: "1234");  
            System.out.println("Conexão realizada com sucesso!");  
            return conn;  
        }  
        catch (Exception e) {  
            System.out.println("Erro ao conectar no BD"+e.getMessage());  
            return null;  
        }  
    }  
}
```

Projeto com banco MySQL

- E na classe principal, vamos instanciar um objeto da classe e chamar o método `getConexao()`

```
public class JavaBD01 {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        Conexao c = new Conexao();  
        c.getConexao();  
    }  
}
```

Projeto com banco MySQL

- Se não apareceu mensagem de erro e apareceu somente esta mensagem, conseguimos conectar com o banco:

```
run:
Mon Sep 25 22:12:49 BRT 2023 WARN: Establishing SSL connection without server's identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must
BUILD SUCCESSFUL (total time: 1 second)
```

Projeto com banco MySQL

- Mas agora, vamos criar dois pacotes novos:
 - Crie um pacote chamado beans
 - Neste pacote, teremos as classes que mapearão as tabelas do banco de dados, ou seja, as classes terão como atributos as colunas da tabela.
 - É interessante em nossos projetos que tenhamos um "bean" para cada tabela do banco.

Projeto com banco MySQL

- Dentro do pacote beans, vamos criar a classe Pessoa:

```
public class Pessoa {  
    private int id;  
    private String nome;  
    private String sexo;  
    private String idioma;
```

```
    public int getId() {  
        return id;  
    }  
    public String getNome() {  
        return nome;  
    }  
    public String getSexo() {  
        return sexo;  
    }  
    public String getIdioma() {  
        return idioma;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
    public void setSexo(String sexo) {  
        this.sexo = sexo;  
    }  
    public void setIdioma(String idioma) {  
        this.idioma = idioma;  
    }  
}
```

Projeto com banco MySQL

- Mas agora, vamos criar dois pacotes novos:
 - E outro pacote chamado dao:
 - DAO = Data Access Object ou Objeto de Acesso aos Dados
 - Neste pacote, teremos as classes responsáveis por persistir os dados (inserção, edição, exclusão e pesquisa) da aplicação nas tabelas.
 - Resumindo, é ela que executará os códigos SQL no banco.
 - É importante que tenhamos também um DAO para cada tabela.

Projeto com banco MySQL

- E no pacote DAO, vamos fazer o insert no banco:

```
public class PessoaDAO {
    private Conexao conexao;
    private Connection conn;

    public PessoaDAO() {
        this.conexao = new Conexao();
        this.conn = this.conexao.getConexao();
    }

    public void inserir (Pessoa pessoa) {
        String sql = "INSERT INTO pessoa (nome, sexo, idioma) VALUES (?, ?, ?);";

        try {
            PreparedStatement stmt = this.conn.prepareStatement(string:sql);
            stmt.setString(i: 1, string:pessoa.getNome());
            stmt.setString(i: 2, string:pessoa.getSexo());
            stmt.setString(i: 3, string:pessoa.getIdioma());

            stmt.execute();
        } catch (SQLException ex) {
            System.out.println("Erro ao inserir pessoa: "+ex.getMessage());
        }
    }
}
```

Projeto com banco MySQL

```
import beans.Pessoa;
import javabd01.conexao.Conexao;
import java.sql.Connection;
import java.sql.*;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
```

```
public class PessoaDAO {
    private Conexao conexao;
    private Connection conn;

    public PessoaDAO() {
        this.conexao = new Conexao();
        this.conn = this.conexao.getConexao();
    }

    public void inserir (Pessoa pessoa) {
        String sql = "INSERT INTO pessoa (nome, sexo, idioma) VALUES (?, ?, ?);";

        try {
            PreparedStatement stmt = this.conn.prepareStatement(string:sql);
            stmt.setString(i: 1, string:pessoa.getNome());
            stmt.setString(i: 2, string:pessoa.getSexo());
            stmt.setString(i: 3, string:pessoa.getIdioma());

            stmt.execute();
        } catch (SQLException ex) {
            System.out.println("Erro ao inserir pessoa: "+ex.getMessage());
        }
    }
}
```

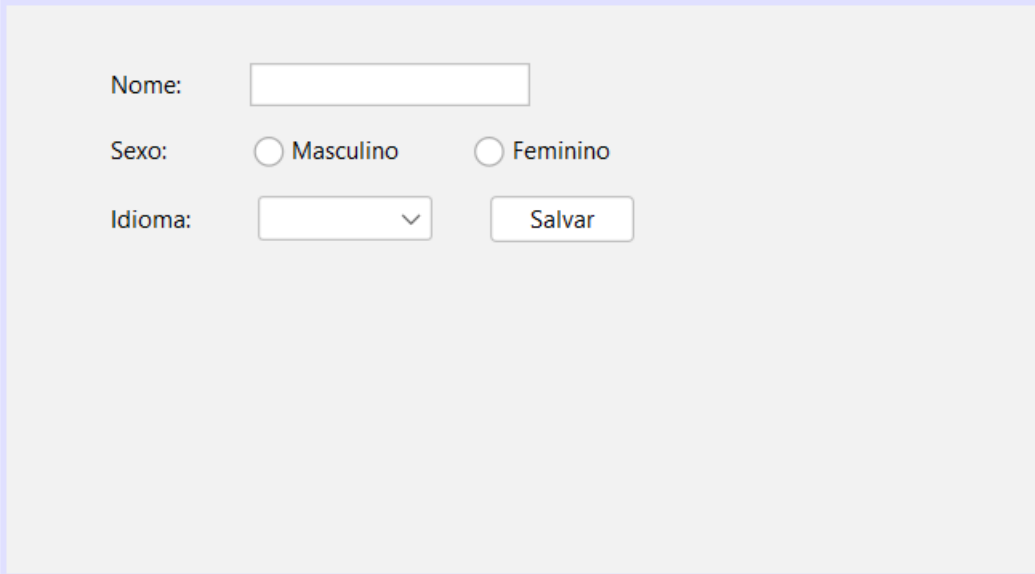
Projeto com banco MySQL

- E na main, vamos fazer somente um teste:

```
public static void main(String[] args) {  
  
    Pessoa p = new Pessoa();  
    p.setNome(nome: "Ricardo");  
    p.setSexo(sexo: "M");  
    p.setIdioma(idioma: "PORTUGUÊS");  
  
    PessoaDAO pDAO = new PessoaDAO();  
    pDAO.inserir(pessoa:p);  
}
```

Projeto com banco MySQL

- Vamos fazer a interface?
- Atividade 1: Faça a interface e faça com que seja cadastrado pessoas a partir dela no banco de dados

A user registration form with a light gray background and a thin blue border. It contains three input fields: a text box for 'Nome:', a radio button group for 'Sexo:' with options 'Masculino' and 'Feminino', and a dropdown menu for 'Idioma:'. A 'Salvar' button is positioned to the right of the 'Idioma:' dropdown.

Nome:

Sexo: ☐ Masculino ☐ Feminino

Idioma:

Atividade 2

- Crie um outro projeto e faça a conexão com o banco de dados escola (desenvolvido no início da aula)
- Faça o cadastro de Alunos e de Professores, em dois formulários ou, se preferir, em um somente