JDBC

Programar aplicativos comp. com integração de BD para desktop

Msc. Lucas G. F. Alves
e-mail: lgfalves@senacrs.com.br

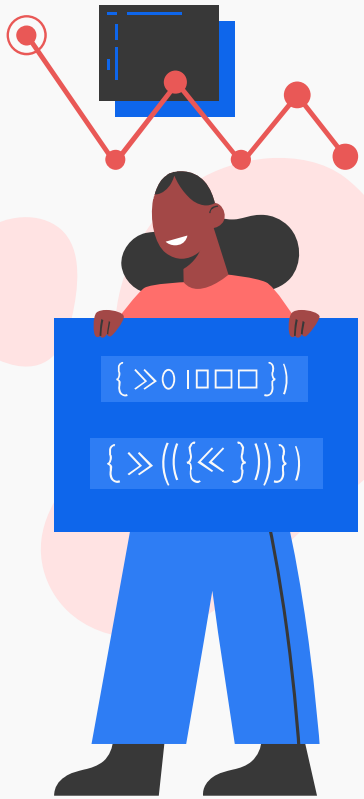
Senac

Planejamento de Aula

Revisão SGBDs

JDBC

Dinâmica



Revisão

SGBDs

Banco de Dados Relacional

Banco de Dados: Repositório de informações, organizadas e estruturadas. As informações podem ser acessadas, atualizadas e consultadas de forma rápida.

Relacional: Os dados são organizados em tabelas. Uma tabela é como uma planilha, com linhas (registros ou tuplas) e colunas (atributos ou campos). Cada linha com um número identificador.

Exemplo: Nossa tabela de Livros terá colunas como Título, Autor, ISBN. Cada linha será um livro específico.

SGBDs

Banco de Dados Relacional

Banco de Dados: Repositório de informações, organizadas e estruturadas. As informações podem ser acessadas, atualizadas e consultadas de forma rápida.

Relacional: Os dados são organizados em tabelas. Uma tabela é como uma planilha, com linhas (registros ou tuplas) e colunas (atributos ou campos). Cada linha com um número identificador.

Relacionamento: Cada tabela se relaciona com outra tabela. Por exemplo, um Aluno pode pegar vários Livros emprestados.

SGBDs

MySQL

O MySQL é o motor, e o **MySQL Workbench** é a interface gráfica para criar, gerenciar e consultar o banco de dados.

Instalando o MySQL.
Acesse: https://dev.mysql.com/downloads/installer/

SGBDs

Instalando o MySQL

Tipo de Servidor: Deixem como 'Development Computer'.
Porta: Deixem a porta padrão 3306.
Autenticação: Usem o método de autenticação padrão (Use Strong Password Encryption).
Senha Root: ATENÇÃO AQUI! Definam uma senha para o usuário root.
Nome do Serviço Windows: Podem deixar o padrão.
Inicialização: Marque para iniciar o MySQL Server como um serviço do Windows na inicialização do sistema.

SGBDs

Criando o Banco de Dados MySQL

Abrir o MySQL Workbench.
Conectar ao Servidor Local: Cliquem em Local Instance MySQL80 (ou o nome da conexão de vocês) na tela inicial. Irá pedir a senha que no caso é root.
Criar um Novo Esquema (Banco de Dados):
No menu superior, vá em File > New Model.
Cliquem duas vezes em Add Diagram.
Na aba Physical Schemas, cliquem no ícone de adicionar esquema (+).
Nomeiem o esquema como **biblioteca_db**.
Cliquem em Apply e depois Finish.

SGBDs

Instalando o PostgreSQL

Tipo de Servidor: Servidor Local.
Porta: Deixem a porta padrão 5432.
Autenticação: Usem o método de autenticação padrão (Use Strong Password Encryption).
Senha Root: ATENÇÃO AQUI! Definam uma senha para o usuário root.
Nome do Serviço Windows: Podem deixar o padrão.

SGBDs

Criando o Banco de Dados PostgreSQL

Abrir o PgAdmin.
Conectar ao Servidor Local: Cliquem em PostgreSQL (ou o nome da conexão de vocês) na tela inicial. Irá pedir a senha que no caso é root.
Criar um Novo Esquema (Banco de Dados):
No menu superior, vá em Object>Create> Database.
Nomeiem o esquema como **biblioteca_db**.
Cliquem em Query Tools.

Dinâmica

Criando o Banco de Dados MySQL/PostgreSQL

Dentro do esquema biblioteca_db:

1)Criar as tabelas **alunos** e **livros**.

2)Criar a tabela **emprestimos**.

3) Inserir os dados nas 3 tabelas **alunos**, **livros** e **emprestimos**.

JDBC

JDBC

O que é JDBC?

JDBC

O que é JDBC?

Java Database Connectivity

É a **API** (**Application Programming Interface**) da Java que permite que aplicações Java interajam com diversos bancos de dados relacionais.

Pensem no **JDBC** como uma **ponte** ou um tradutor universal entre o **Java** e o mundo dos **bancos de dados.**

JDBC

Como o JDBC funciona?

Aplicação Java: Seu código quer acessar o banco.

Driver JDBC: Software específico que sabe como "falar" com um tipo particular de banco de dados (MySQL, PostgreSQL, Oracle, etc.).

Banco de Dados: Onde os dados estão armazenados.

JDBC

Funcionalidades

Conectar: Abrir uma conexão com o banco de dados.
Enviar Comandos: Executar instruções SQL (SELECT, INSERT, UPDATE, DELETE).
Receber Resultados: Trazer os dados do banco de volta para a aplicação Java.

Para o MySQL, vamos usar o Driver JDBC, também conhecido como Connector/J."

JDBC

Configurando o Driver no IntelliJ IDEA

Abrir o IntelliJ IDEA: Abriremos o projeto que criamos (ou criem um novo projeto Java vazio).
Criar um Novo Projeto (se necessário): File > New > Project...
Selecionem Java (ou Maven se já estiverem familiarizados, mas vamos começar simples).
Escolham o SDK (Java Development Kit) que vocês instalaram (ex: OpenJDK 17).
Deem um nome ao projeto, como SistemaBiblioteca ou ConexaoDB.
Cliquem em Finish.

JDBC

Configurando o Driver no IntelliJ IDEA

Baixar o Driver JDBC um arquivo zip (MySQL Connector/J): Acessem o site oficial: https://dev.mysql.com/downloads/connector/j/
Descompactem o arquivo ZIP. Dentro, vocês encontrarão um arquivo .jar (ex: **mysql-connector-j-8.0.33.jar**). Guarde esse arquivo em um local do seu projeto.
Adicionar o Driver ao Projeto no IntelliJ: No IntelliJ, vá em File > Project Structure... (ou Ctrl+Alt+Shift+S).
Na janela que se abre, no menu da esquerda, selecionem 'Libraries'.
Cliquem no botão + (Adicionar Nova Biblioteca) e escolham 'Java'.
Naveguem até a pasta onde vocês salvaram o arquivo .jar do MySQL Connector/J e selecionem-no.
Cliquem em OK e depois em Apply e OK novamente.

JDBC

Criando a Classe de conexão

Criar um novo pacote: No painel Project (geralmente à esquerda), clique com o botão direito na pasta src. New > Package.
Nomeiem como **conexao** (ou util.conexao).

Criar a classe **ConexaoDB**:
Clique com o botão direito no pacote conexao.
New > Java Class.
Nomeiem como **ConexaoDB**.

JDBC

Exemplo de classe de conexão.

```
package conexao; // Certifiquem-se de que o nome do pacote está correto

import java.sql.Connection; // Importa a interface Connection
import java.sql.DriverManager; // Importa a classe DriverManager para gerenciar drivers
import java.sql.SQLException; // Importa a exceção SQLException para tratamento de erros
```

JDBC

Atributos da classe.

```
private static final String URL = "jdbc:mysql://localhost:3306/biblioteca_db";
private static final String USUARIO = "root"; // O usuário que configuramos
private static final String SENHA = "root"; // A senha que configuramos
```

JDBC

Dentro da classe criada.

```
public static Connection conectar() {
    Connection conexao = null; // Inicializa a conexão como nula
    try {
      conexao = DriverManager.getConnection(URL, USUARIO, SENHA);
      System.out.println("Conexão com o banco de dados estabelecida com
sucesso!");
    } catch (SQLException e) {
      System.err.println("Erro ao conectar com o bd: " + e.getMessage());
       } return conexao; // Retorna a conexão (pode ser null em caso de erro)
  }
```

JDBC

Dentro da classe criada.

```
public static void fecharConexao(Connection conexao) {
    if (conexao != null) { // Verifica se a conexão não é nula antes de tentar fechar
      try {
        conexao.close(); // fecha a conexão
        System.out.println("Conexão com o bd fechada!");
      } catch (SQLException e) {
          System.err.println("Erro ao fechar a conexão com o bd: " +
e.getMessage());
      }
    }}
```

JDBC

Dentro da classe criada.

```
public static void main(String[] args) {
    Connection testeConexao = ConexaoDB.conectar();
    if (testeConexao != null) {
      // Se a conexão for bem-sucedida, podemos fechá-la
      ConexaoDB.fecharConexao(testeConexao);
    }
  }
```

JDBC

Testando a conexão.

Na classe **ConexaoDB.java.**
Cliquem com o botão direito dentro da classe ou no nome da classe no painel Project.
Selecionem **Run 'ConexaoDB.main()'.**
Se tudo der certo: Vocês verão a mensagem Conexão com o bd estabelecida com sucesso! e depois Conexão com o bd fechada!.
Se der erro: Vocês verão a mensagem Erro ao conectar com o banco de dados: ... seguida da descrição do erro. (Aproveitem para debugar juntos!)

Atividade

Atividade

1. Criar o projeto do PI.
2. Criar a conexão com o Banco de Dados do P.I no PostgreSQL ou MySQL.
3. Testar a conexão tanto o PostgreSQL quanto o MySQL.
4. Enviar os repositório do Drive ou Git individualmente.

Professor

Lucas G. F. Alves

Senac

Obrigado!

E-mail: lgfalves@senacrs.com.br