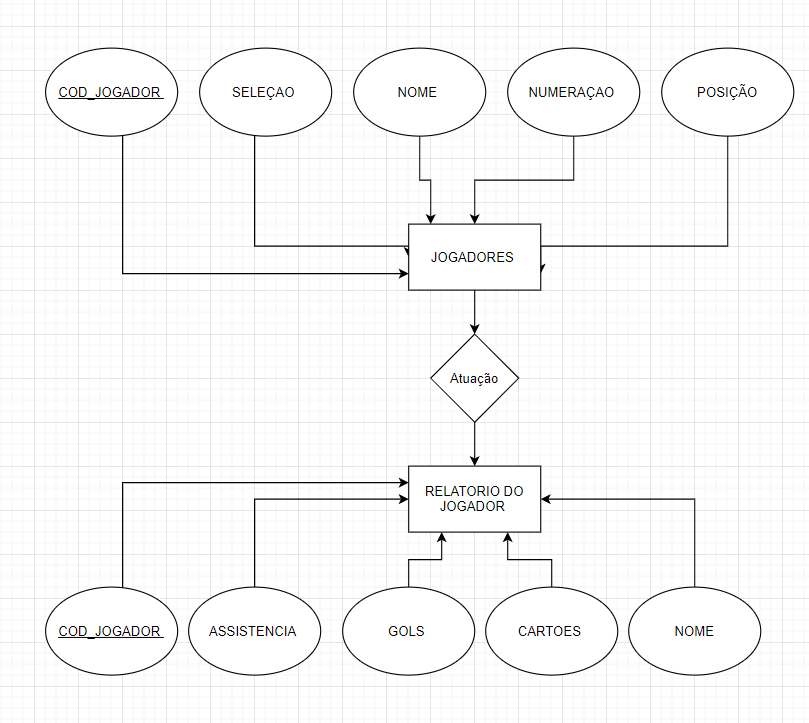
## Requisitos:

* Diagrama ER. (Pelo menos 2 tabelas e uma relação (FKey))

Obs: Quem fez o primeiro trabalho pode aproveitar o tema e o diagrama feito, porém arrumar caso necessário.

* Projeto no netbeans que crie as tabelas/ insira dados nas tabelas/visualize as tabelas modeladas no diagrama ER.
* Breve apresentação mostrando o que fez (códigos), quais foram os desafios e as dificuldades encontradas.

e caio, deixa esse diagrama salvo no teu pc, pq hoje não vamos terminar aqui o bagulho



Primeiro código:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.sql.ResultSet;

public class Copa {

private static Connection connection;

public static Connection getConnection(){

if(connection == null){

try{

//Class.forName("com.mysql.jdbc.Driver"); //para mysql

Class.forName("org.postgresql.Driver");//para postgresql

String host = "localhost";

String port = "5432";

String database = "postgres";

String user = "postgres";

String pass = "senai";//digitar a senha do seu banco

//String url = "jdbc:mysql://"+host+":"+port+"/"+database; //para mysql

String url = "jdbc:postgresql://"+host+":"+port+"/"+database;//para postgresql

connection = DriverManager.getConnection(url, user, pass);

} catch (ClassNotFoundException e) {

e.printStackTrace();

} catch (SQLException e) {

e.printStackTrace();

}

}

return connection;

}

public static void close(){

if (connection == null){

throw new RuntimeException("Nenhuma conexão aberta!");

}

else{

try{

connection.close();

}

catch (SQLException e){

e.printStackTrace();

}

}

}

public static void salvar (Jogador jogador){

try{

Connection con = Copa.getConnection();

PreparedStatement ps = con.prepareStatement("INSERT INTO jogador (cod, cpf, selecao, idade) values(?, ?, ?, ?)");

ps.setInt(1, jogador.getCod());

ps.setString(2, jogador.getCpf());

ps.setString(3, jogador.getTelefone());

ps.setString(4, jogador.getEndereco());

ps.execute();

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void salvar (Fase Fase){

try{

Connection con = Copa.getConnection();

PreparedStatement ps = con.prepareStatement("INSERT INTO fase (classificacao, selecao, gols) values(?, ?, ?)");

ps.setLong(1, Fase.getIsbn());

ps.setString(2, Fase.getNome\_autor());

ps.setString(3, livro.getEditora());

ps.setString(4, livro.getAssunto());

ps.setInt(5, livro.getQtd\_estoque());

ps.execute();

} catch (SQLException e) {

e.printStackTrace();

}

}

public static void deletaJogador(int cod){

try{

Connection con = Copa.getConnection();

PreparedStatement ps = con.prepareStatement("Delete FROM jogador WHERE cod = ?");

ps.setInt(1, cod);

ps.executeUpdate();

}

catch (SQLException e) {

e.printStackTrace();

}

}

public static void deletaFase(long isbn){

try{

Connection con = Copa.getConnection();

PreparedStatement ps = con.prepareStatement("Delete FROM fase WHERE isbn = ?");

ps.setLong(1, isbn);

ps.executeUpdate();

}

catch (SQLException e) {

e.printStackTrace();

}

}

public static void atualizaCopa(Jogador Jogador){

deletaJogador(jogador.getCod());

salvar(jogador);

}

public static void atualizaCopa(Fase fase){

deletaLivro(fase.getIsbn());

salvar(fase);

}

public static void visualiza\_tabela(String tabela, String... atributo){

try{

Connection con = Copa.getConnection();

PreparedStatement ps = con.prepareStatement("SELECT \* FROM "+ tabela);

ResultSet rs = ps.executeQuery();

String s = "";

while(rs.next()){

for(String i : atributo){

s = s + " | " +rs.getString(i);

}

s = s + "\n";

}

System.out.println(s);

}

catch (SQLException e) {

e.printStackTrace();

}

}

public static void criar\_tabela (String tabela, String... atributo){

try{

Connection con = Copa.getConnection();

String comando = "CREATE TABLE " + tabela + "(";

for(String i : atributo){

comando = comando + i + ",";

}

comando = comando.substring(0, comando.length()-1);

comando = comando + ");";

PreparedStatement ps = con.prepareStatement(comando);

ps.execute();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

Segundo código:

public class Jogador{

private int cod;

private String cpf;

private String selecao;

private String idade;

public int getCod() {

return cod;

}

public void setCod(int cod) {

this.cod = cod;

}

public String getCpf() {

return cpf;

}

public void setCpf(String cpf) {

this.cpf = cpf;

}

public String getSelecao() {

return selecao;

}

public void setSelecao(String selecao) {

this.selecao = selecao;

}

public String getIdade() {

return idade;

}

public void setIdade(String idade) {

this.idade = idade;

}

}

Terceiro código:

public class Fase {

private String classificacao;

private String selecao;

private String gols;

public String getClassificacao() {

return Classificacao;

}

public void setClassificaocao(String classificacao{

this.classificacao = classificacao;

}

public String getSelecao() {

return selecao;

}

public void setSelecao(String selecao) {

this.selecao = selecao;

}

public String getGols() {

return gols;

}

public void setGols(String gols) {

this.gols = gols;

}

}