



# **BANCO DE DADOS I**

**MSc. Fernanda Dias**

**CENTRO UNIVERSITÁRIO UNESP**

```
CREATE DATABASE aula01;
```

```
USE aula01;
```

```
CREATE TABLE cidades (  
    id_cidade INT PRIMARY KEY,  
    nome_cidade VARCHAR(50),  
    estado VARCHAR(2)  
);
```

```
SHOW TABLES;
```

```
DESCRIBE CIDADES;
```

```
INSERT INTO CIDADES (id_cidade, nome_cidade, estado)  
VALUES (1, 'João Pessoa', 'PB')
```

```
SELECT * FROM cidades;
```

```
SELECT * FROM cidades WHERE id_cidade = 1;
```

```
SELECT nome_cidade FROM CIDADES WHERE id_cidade = 1;
```

```
SET SQL_SAFE_UPDATES = 0;
```

```
DELETE FROM cidades WHERE id_cidade = 3;
```

```
DROP TABLE cidades;
```

**ALTER TABLE CIDADES**

**ADD** quantidade\_habitantes int;

**ALTER TABLE CIDADES**

**DROP COLUMN** quantidade\_habitantes;

**ALTER TABLE ENDERECO**

**MODIFY COLUMN** cep VARCHAR(10) **AFTER** bairro;

**UPDATE CIDADES SET** quantidade\_habitantes = 10

**WHERE** id\_cidade = 1;

**CONECTANDO**

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conexao {
    public static Connection GeraConexao() {

        Connection conexao = null;

        try{
            String url = "jdbc:mysql://localhost/nome_do_banco";
            String usuario = "root";
            String senha = "root";
            conexao = DriverManager.getConnection(url, usuario, senha);
            System.out.println("Conexão realizada com sucesso.");
        } catch (SQLException e) {
            System.out.println("Erro ao tentar conectar: " + e.getMessage());
            conexao = null;
        }
        return conexao;
    }
}
```

# FILE -> Project Structure



<https://dev.mysql.com/downloads/connector/j/>

Código JAVA



```
public class Cidade {  
    private int idCidade;  
    private String nomeCidade;  
    private String estado;
```

```
    //get e set
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {
```

```
        Cidade c = new Cidade();  
        c.setIdCidade(3);  
        c.setNomeCidade("São Paulo");  
        c.setEstado("SP");
```

```
        CidadeDAO cidadeDAO = new CidadeDAO();  
        cidadeDAO.adiciona(c);
```

```
    }  
}
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class CidadeDAO {
    private Connection connection;

    public CidadeDAO() {
        this.connection = new Conexao().GeraConexao();
    }

    public void adiciona(Cidade c) {
        String sql = "INSERT INTO cidades(id_cidade, nome_cidade, estado) VALUES(?,?,?)";
        try {
            PreparedStatement stmt = connection.prepareStatement(sql);
            stmt.setInt(1, c.getIdCidade());
            stmt.setString(2, c.getNomeCidade());
            stmt.setString(3, c.getEstado());
            stmt.execute();
            stmt.close();
            System.out.println("Cidade adicionada com sucesso.");
        } catch (SQLException e) {
            throw new RuntimeException(e);
        }
    }
}
```

# PRATICANDO

```
CREATE TABLE cidades (  
  id_cidade INT PRIMARY KEY,  
  nome_cidade VARCHAR(50),  
  estado VARCHAR(2)  
);
```

	id_cidade	nome_cidade	estado
▶	1	João Pessoa	PB
	2	Cabedelo	PB
✱	NULL	NULL	NULL

```
CREATE TABLE endereco (  
  id_endereco INT PRIMARY KEY,  
  rua VARCHAR(50),  
  numero INT,  
  bairro VARCHAR(50),  
  cep VARCHAR(10),  
  id_cidade INT,  
  FOREIGN KEY (id_cidade) REFERENCES cidades (id_cidade)  
);
```

	id_endereco	rua	numero	bairro	cep	id_cidade
▶	1	abc	20	centro	58000	1
✱	NULL	NULL	NULL	NULL	NULL	NULL

CRUD

// Create

```
public void adiciona(Cidade c) {  
    String sql = "INSERT INTO cidades(id_cidade, nome_cidade, estado) VALUES(?,?,?)";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        stmt.setInt(1, c.getIdCidade());  
        stmt.setString(2, c.getNomeCidade());  
        stmt.setString(3, c.getEstado());  
        stmt.execute();  
        stmt.close();  
        System.out.println("Cidade adicionada com sucesso.");  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

// Read

```
public ArrayList<Cidade> listar() {  
    ArrayList<Cidade> cidades = new ArrayList<>();  
    String sql = "SELECT id_cidade, nome_cidade, estado FROM cidades";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        ResultSet resultSet = stmt.executeQuery();  
        while (resultSet.next()) {  
            Cidade c = new Cidade();  
            c.setIdCidade(resultSet.getInt("id_cidade"));  
            c.setNomeCidade(resultSet.getString("nome_cidade"));  
            c.setEstado(resultSet.getString("estado"));  
            cidades.add(c);  
        }  
        stmt.close();  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
    return cidades;  
}
```

// Read por ID

```
public Cidade buscarPorId(int idCidade) {  
    String sql = "SELECT id_cidade, nome_cidade, estado FROM cidades WHERE id_cidade = ?";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        stmt.setInt(1, idCidade);  
        ResultSet resultSet = stmt.executeQuery();  
        if (resultSet.next()) {  
            Cidade c = new Cidade();  
            c.setIdCidade(resultSet.getInt("id_cidade"));  
            c.setNomeCidade(resultSet.getString("nome_cidade"));  
            c.setEstado(resultSet.getString("estado"));  
            stmt.close();  
            return c;  
        } else {  
            stmt.close();  
            return null;  
        }  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

*// Update*

```
public void atualiza(Cidade c) {  
    String sql = "UPDATE cidades SET nome_cidade = ?, estado = ? WHERE id_cidade = ?";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        stmt.setString(1, c.getNomeCidade());  
        stmt.setString(2, c.getEstado());  
        stmt.setInt(3, c.getIdCidade());  
        stmt.executeUpdate();  
        stmt.close();  
        System.out.println("Cidade atualizada com sucesso.");  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```



*// Delete*

```
public void exclui(int idCidade) {  
    String sql = "DELETE FROM cidades WHERE id_cidade = ?";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        stmt.setInt(1, idCidade);  
        stmt.executeUpdate();  
        stmt.close();  
        System.out.println("Cidade excluída com sucesso.");  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```

Main

*// Create*

```
Cidade c = new Cidade();  
c.setIdCidade(1);  
c.setNomeCidade("São Paulo");  
c.setEstado("SP");  
  
CidadeDAO cidadeDAO = new CidadeDAO();  
cidadeDAO.adiciona(c);
```

*// Update*

```
Cidade cidadeAtt = cidadeDAO.buscarPorId(1);  
if (cidadeAtt != null) {  
    cidadeAtt.setNomeCidade("Nova Cidade");  
    cidadeDAO.atualiza(cidadeAtt);  
}
```

*// Delete*

```
cidadeDAO.exclui(2);
```

*// Read*

```
ArrayList<Cidade> cidades = cidadeDAO.listar ();  
for (Cidade c : cidades) {  
    System.out.println("ID: " + c.getIdCidade() +  
        ", Nome: " + c.getNomeCidade() +  
        ", Estado: " + c.getEstado());  
}
```

*// Read (Leitura) - Buscar uma cidade por ID*

```
Cidade cidadeEncontrada = cidadeDAO.buscarPorId(1);  
if (cidadeEncontrada != null) {  
    System.out.println("Cidade encontrada: " +  
        cidadeEncontrada.getNomeCidade());  
} else {  
    System.out.println("Cidade não encontrada.");  
}
```

**Observação!**

```
public void adiciona(Cidade c) {  
    // Verifica se a cidade já existe antes de adicioná-la  
    if (cidadeJaExiste(c.getIdCidade())) {  
        System.out.println("Cidade com id " + c.getIdCidade() + " já existe na tabela.");  
        return;  
    }  
    //Código do CREATE aqui  
}  
private boolean cidadeJaExiste(int idCidade) {  
    // Verifica se a cidade já existe no banco de dados  
    String sql = "SELECT COUNT(*) FROM cidades WHERE id_cidade = ?";  
    try {  
        PreparedStatement stmt = connection.prepareStatement(sql);  
        stmt.setInt(1, idCidade);  
        ResultSet resultSet = stmt.executeQuery();  
        resultSet.next();  
        int count = resultSet.getInt(1);  
        stmt.close();  
        return count > 0;  
    } catch (SQLException e) {  
        throw new RuntimeException(e);  
    }  
}
```