

# SQL API Java

- Pré-requisitos;
- Conexão;
- Declarações SQL;
- Consulta;
- Fechando a conexão;
- Exemplos.

# Pré-requisitos

- Instalar o Java JDK;
- Instalar um banco de dados com suporte JDBC ou ODBC;
- Instalar o driver fornecido pelo desenvolvedor do seu banco de dados para usar no Java;
- Importar a API do Java para SQL:
  - `Import java.sql.*;`

# Conexão ao Banco

O programa inicia tentando localizar o driver, passando o caminho de busca do mesmo. Este driver é fornecido pelo fabricante e pode ser atualizado com novas versões.

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); // Access  
Class.forName("com.mysql.jdbc.Driver"); // MySQL
```

Conexão ao banco de dados e Driver:

```
Connection con = DriverManager.getConnection(url, username, password);  
  
String username = "usuario";  
String password = "senha";  
String url = "jdbc:mysql://localhost/popov";
```

# Declarações SQL

Para efetuar as declarações SQL é necessário criar um objeto Statement a partir da conexão criada.

```
Statement stmt = con.createStatement();
```

# Declarações SQL

O envio da declaração SQL para o bando de dados é dado através do método `executeUpdate(String strSQL)`. Este método como parâmetro `strSQL`, que é uma declaração SQL.

```
String query =
```

```
    "insert into funcionario (nome) values ('Roberto')"
```

```
stmt.executeUpdate(query);
```

# Consulta

```
String query = "SELECT * FROM funcionario";
```

```
ResultSet rs = stmt.executeQuery(query);
```

```
while (rs.next()) {
```

```
    int codigo = rs.getInt(0);
```

```
    String nome = rs.getString(1);
```

```
    System.out.println(codigo + " - " + nome);
```

```
}
```

# Fechando a Conexão

No final o programa deve liberar os recursos das conexões e sentenças JDBC com o método `close()`.

Este procedimento acelera a liberação de recursos, mas não é imprescindível devido ao Garbage Collector.

```
stmt.close();  
conn.close();
```

# Estrutura Exemplo

```
CREATE TABLE funcionario (  
    codigo int(5) auto_increment,  
    nome varchar(50),  
    PRIMARY KEY (codigo));
```

```
CREATE TABLE dependente (  
    codigo int(5) auto_increment,  
    codigo_funcionario int(5),  
    nome varchar(50),  
    PRIMARY KEY (codigo));
```

# Inserindo dados

```
public class InserirFuncionario {
    public static void main(String args[]) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection(
                "jdbc:mysql://localhost/popov", "root", "");
            Statement stmt = con.createStatement();
            String query =
                "insert into funcionario (nome) values (" + args[0] + ")";
            stmt.executeUpdate(query);
            stmt.close();
            con.close();
        } catch(SQLException ex) {
            System.err.println("SQLException: " + ex.getMessage());
        } catch(java.lang.ClassNotFoundException e) {
            System.err.println("ClassNotFoundException: " + e.getMessage());
        }
    }
}
```

# Listando dados

```
String query = "select * from funcionario";  
ResultSet rs = stmt.executeQuery(query);  
  
while (rs.next()) {  
    int codigo = rs.getInt("codigo");  
    String nome = rs.getString("nome");  
    System.out.println(codigo + " - " + nome);  
}
```

# Excluindo dados

```
Statement stmt = con.createStatement();
```

```
String query =
```

```
    "delete from funcionario where codigo='" + args[0] + "'";
```

```
stmt.executeUpdate(query);
```

# Junção

```
String query =
```

```
    "SELECT f.nome as funcionarioNome, d.nome, d.idade " +  
    "FROM funcionario f, dependente d " +  
    "WHERE f.codigo = d.codigo_funcionario";
```

```
ResultSet rs = stmt.executeQuery(query);
```

```
while (rs.next()) {
```

```
    String nomeFuncionario = rs.getString("funcionarioNome");
```

```
    String nome = rs.getString("nome");
```

```
    int idade = rs.getInt("idade");
```

```
    System.out.println(nomeFuncionario + " - " + nome + " - " +  
        idade);
```

```
}
```

# Compilar e executar

COMPILAR:

```
javac Programa.java
```

EXECUTAR:

```
java -cp .::mysql-connector-java-3.1.13-bin.jar Programa
```