

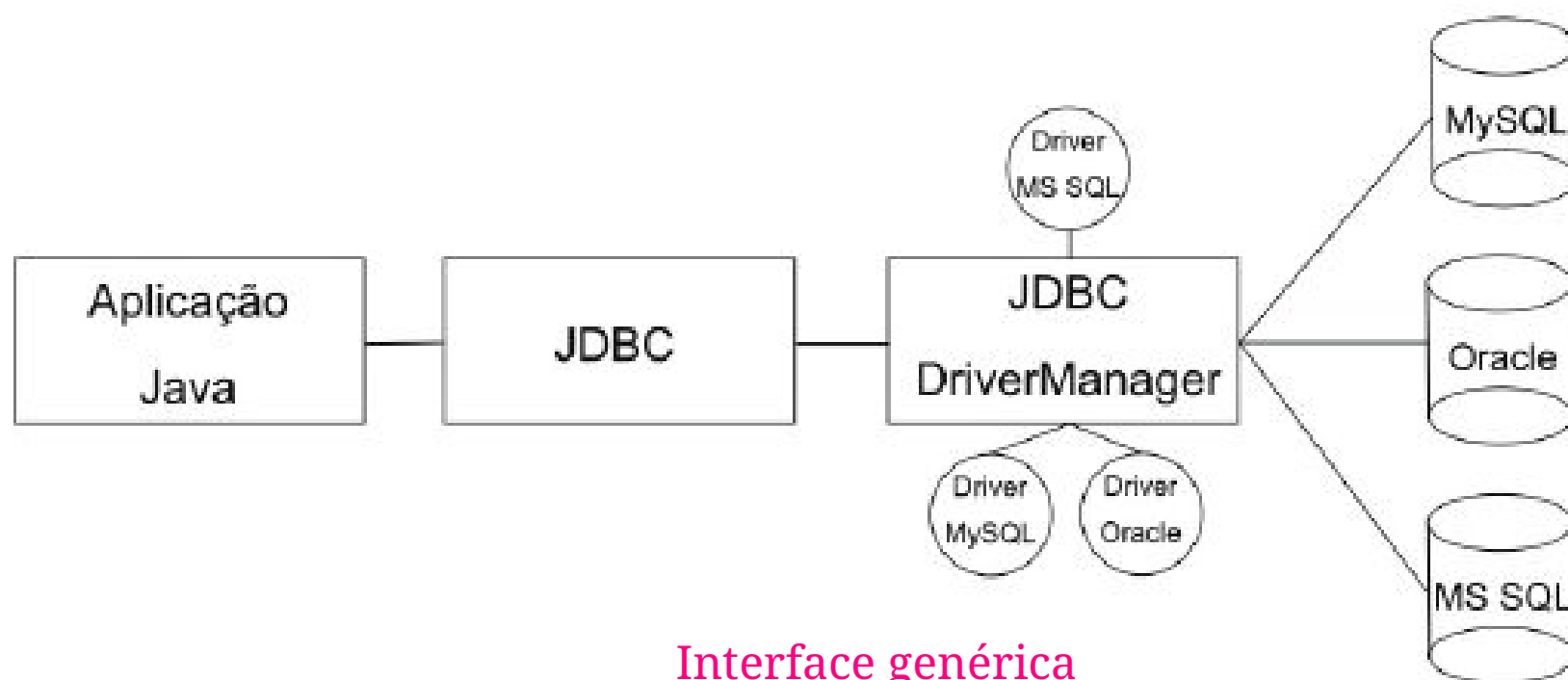
Programação em Java usando banco de dados relacional

Programação Orientada a Objetos

- Para utilizar o JDBC com qualquer SGBD é necessário executar cinco passos, e então será possível executar comandos para buscar dados ou enviar informações para o SGBD
 - Estabelecer a conexão
 - Criar um objeto da classe statement vinda da conexão para possibilitar a execução das consultas
 - Executar as consultas
 - Processar os resultados, sendo os dados enviados ou recebidos
 - Fechar a conexão

- Para utilizar os SGDBs em Java, especialmente em interfaces gráficas em Java Swing, é indicado utilizar o Java Database Connectivity (JDBC)
- Possibilita o acesso a diversos SGDBs de forma padronizada
 - MySQL, Oracle, Microsoft SQL, etc.
- O JDBC é uma interface genérica para diversos sistemas de bancos de dados

Figura 1.8 | Representação da ação do JDBC



- A API JDBC é implementada pelos pacotes `java.sql` e `javax.sql`, onde estão as classes para manipulação
 - `java.sql.DriverManager`: cria a conexão com SGBD
 - `java.sql.Connection`: representa a conexão com o SGBD e fornece acesso às consultas
 - `java.sql.Statement`: executar consultas e comandos
 - `java.sql.ResultSet`: recuperar dados que foram buscados, por exemplo, um comando de `select`
 - `javax.sql.DataSource`: agrupar conexões com o SGBD

- Para se conectar a um banco de dados, esteja ele implementado em qualquer SGBD, é necessário criar uma string de conexão, ou URL JDBC (Uniform Resource Locator JDBC)
- Informará o “caminho” do banco e apresenta a seguinte sintaxe:

`jdbc:<driver>:<detalhes da conexão>`

Banco de dados	URL JDBC
MySQL	<code>jdbc:mysql://localhost:3306/nomeBancoDeDados</code>
SQL Server	<code>jdbc:sqlserver://localhost;databaseName = nomeBancoDeDados</code>
Oracle	<code>jdbc:oracle:thin@myserver:1521:nomeBancoDeDados</code>

MySQL - criando usuário

```
$ sudo apt install mysql-server libmariadb-java
```

```
$ sudo mysql
```

```
CREATE USER 'teste'@'localhost' IDENTIFIED BY 'teste';
```

```
GRANT ALL PRIVILEGES ON teste.* TO 'teste'@'localhost' WITH GRANT  
OPTION;
```

```
exit
```

para apagar um usuário:
drop user teste@localhost;
flush privileges;

JShell (Java REPL)

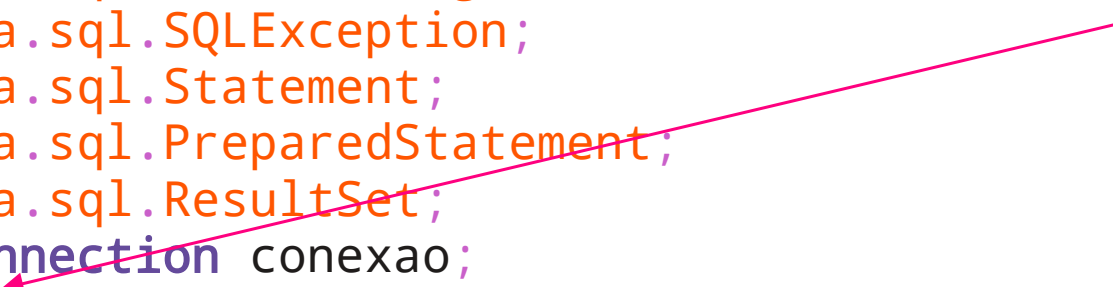
docs.oracle.com/en/java/javase/11/jshell/

```
$ CLASSPATH=$CLASSPATH:/usr/share/java/mariadb-java-client.jar &&  
jshell
```

```
| Welcome to JShell -- Version 18.0.2-ea  
| For an introduction type: /help intro
```

```
jshell> import java.sql.Connection;  
...> import java.sql.DriverManager;  
...> import java.sql.SQLException;  
...> import java.sql.Statement;  
...> import java.sql.PreparedStatement;  
...> import java.sql.ResultSet;  
...> private Connection conexao;  
...> private final String URLDB = "jdbc:mysql://localhost";  
...> private final String usuario = "teste";  
...> private final String senha = "teste";  
...>
```

atributo
imutável,
inalterável



```
conexao ==> null  
URLDB ==> "jdbc:mysql://localhost"  
usuario ==> "teste"  
senha ==> "teste"
```

saída do jshell

tipo objeto

```
jshell> conexao = DriverManager.getConnection(URLDB, usuario, senha);
...> Statement comando = conexao.createStatement();
...> comando.execute("CREATE DATABASE teste");
...> comando.execute("USE teste");
...> comando.execute("CREATE TABLE pessoa" +
...>                  "(id          INT unsigned NOT NULL," +
...>                  "nome          VARCHAR(150)," +
...>                  "sobrenome      VARCHAR(150)," +
...>                  "nivelusuario  INT)");
...> comando.execute("INSERT INTO pessoa" +
...>                  "(id, nome, sobrenome, nivelusuario) VALUES"
+
...>                  "(1, 'NomePessoa', 'Sobrenome', 1)");
conexao ==> org.mariadb.jdbc.MariaDbConnection@5474c6c
comando  ==> org.mariadb.jdbc.MariaDbStatement@16c0663d
$13 ==> false
$14 ==> false
$15 ==> false
$16 ==> false
```

```
jshell> ResultSet res = comando.executeQuery("SELECT * FROM pessoa");
...> while(res.next()) {
...>     System.out.println(res.getString("nome")); }
res ==>
org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet@47f37ef1
NomePessoa

jshell> comando.execute("UPDATE pessoa SET nome = 'OutroNome' WHERE
id = 1");
...> ResultSet res = comando.executeQuery("SELECT * FROM pessoa");
...> while(res.next()) {
...>     System.out.println(res.getString("nome")); }
$19 ==> false ← = método não tem retorno
res ==>
org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet@19bb089b
OutroNome
```

```
jshell> comando.execute("DELETE FROM pessoa WHERE id = 1");
...> comando.execute("INSERT INTO pessoa" +
...>                    "(id, nome, sobrenome, nivelusuario) VALUES" +
...>                    "(1, 'Fulano', 'Sobrenome', 1)");
...> ResultSet res = comando.executeQuery("SELECT * FROM pessoa");
...> while(res.next()) {
...>     System.out.println(res.getString("nome")); }
$22 ==> false
$23 ==> false
res ==>
org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet@1fbc7afb
Fulano
```

```

jshell> PreparedStatement psInsert = conexao.prepareStatement(
...>     "INSERT INTO pessoa (id,nome,sobrenome,nivelusuario)
VALUES (?, ?, ?, ?)");
...> psInsert.setInt(1, 2);
...> psInsert.setString(2, "Ciclano");
...> psInsert.setString(3, "Sobrenome");
...> psInsert.setInt(4, 1);
...> psInsert.execute();
...> ResultSet res = comando.executeQuery("SELECT * FROM pessoa");
...> while(res.next()) {
...>     System.out.println(res.getString("nome")); }
psInsert ==> sql : 'INSERT INTO pessoa (id,nome,sobrenome,nive ...
rs : [null,null,null,null]
$31 ==> false
res ==>
org.mariadb.jdbc.internal.com.read.resultset.SelectResultSet@48140564
Fulano
Ciclano

jshell> /exit
| Goodbye
$

```

MySQL - verificando resultados

```
$ sudo mysql
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 22
```

```
Server version: 8.0.31-0ubuntu0.22.04.1 (Ubuntu)
```


```
Copyright (c) 2000, 2022, 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> SHOW DATABASES;
```



Database
information_schema
mysql
performance_schema
pets
sys
teste

```
6 rows in set (0.30 sec)
```

```
mysql> use teste  
Database changed
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_teste |
+-----+
| pessoa          |
+-----+
1 row in set (0,00 sec)
```

```
mysql> describe pessoa;
```

```
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id             | int unsigned  | NO   |     | NULL    |       |
| nome          | varchar(150)  | YES  |     | NULL    |       |
| sobrenome      | varchar(150)  | YES  |     | NULL    |       |
| nivelusuario  | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0,11 sec)
```

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

id	nome	sobrenome	nivelusuario	
1	NomePessoa	Sobrenome		1

```
1 row in set (0,00 sec)
```

```
mysql> exit
```

```
Bye
```

```
$
```