

# ***FACEC***

## ***F a c u l d a d e***

# Cont. Linguagem SQL

Professor: Yuri Ferreira

- Revisão aula anterior:
  - Funções e Procedures;

## ➤ Conteúdo:

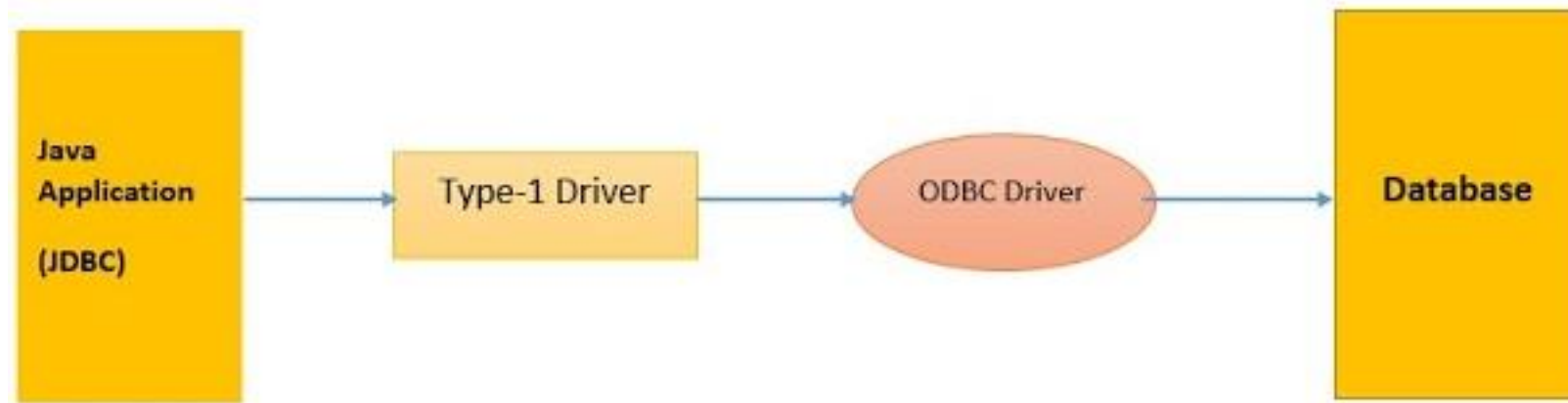
- API JDBC;
- Tipos de drivers JDBC;
- Criar um projeto Java;
- Carregar driver;
- Conectar no Banco;
- Executar Instrução Create Table;



- JDBC – Conceitos
- JDBC (Java Database Connectivity) é uma API para conectar e executar consultas em um banco de dados;
- pode trabalhar com **qualquer** BD, desde que os drivers adequados sejam fornecidos;
- Composto por um conjunto de **classes** e **interfaces** escritas em Java; (Ex: Connection, DriverManager, ResultSet e Statement);
- Facilita o acesso a múltiplos bancos de dados no mesmo programa de aplicação;

- JDBC – Conceitos
- Um **driver** JDBC é uma **implementação** usada para conectar-se a um determinado tipo de banco de dados;
- Existem vários **tipos de drivers** JDBC:
  - 1. Driver de ponte **JDBC-ODBC**;
  - 2. Driver de **API nativa**;
  - 3. Driver de **protocolo** de rede;
  - 4. Driver **Java** puro;

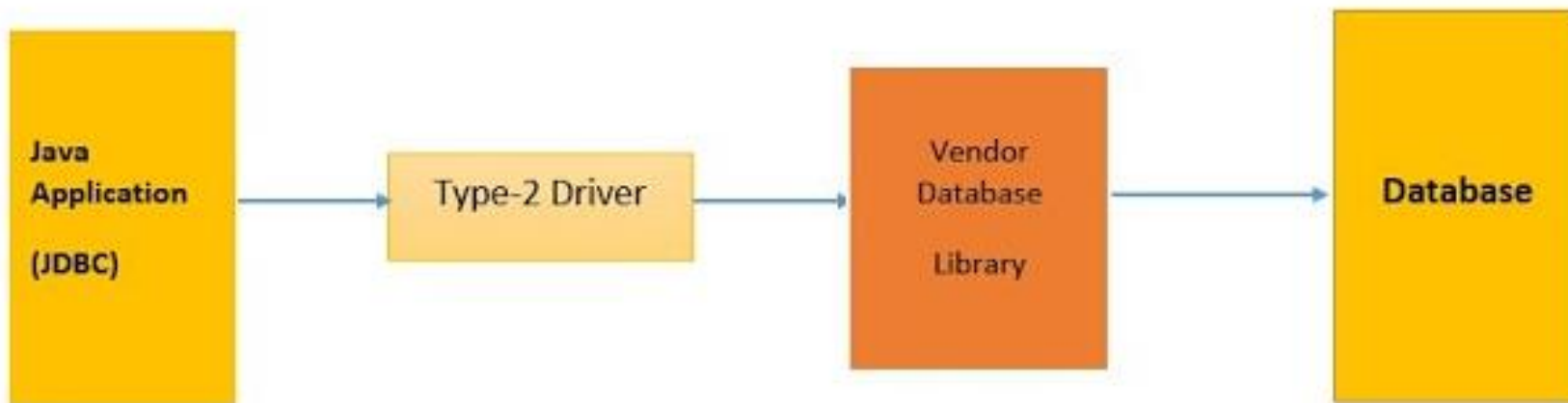
## ➤ 1. Driver de Ponte JDBC-ODBC:



Fonte: SRIVASTAVA, 2018

- É o tipo de **driver** mais lento, pois tem que fazer a migração a tradução de JDBC para ODBC;
- Depende internamente do driver ODBC;

## ➤ 2. Driver de API nativa:

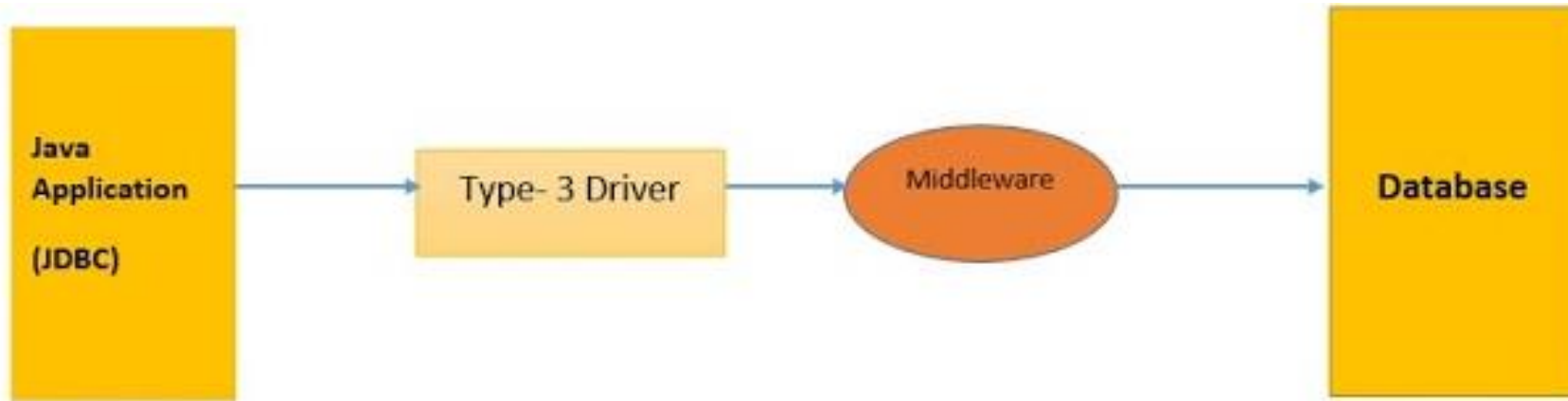


Fonte: SRIVASTAVA, 2018

- Grandes fornecedores como Oracle e IBM, usam driver Tipo 2 para bancos de dados corporativos;
- Não possui compatibilidade entre arquiteturas, pois são compilados para cada arquitetura;



## ➤ 3. Protocolo driver de Rede:



Fonte: SRIVASTAVA, 2018

- Suporte a rede é necessário na máquina cliente;
- Independente de plataforma;
- Depende de um middleware do banco de dados;

## ➤ 4. Driver Java Puro:



Fonte: SRIVASTAVA, 2018

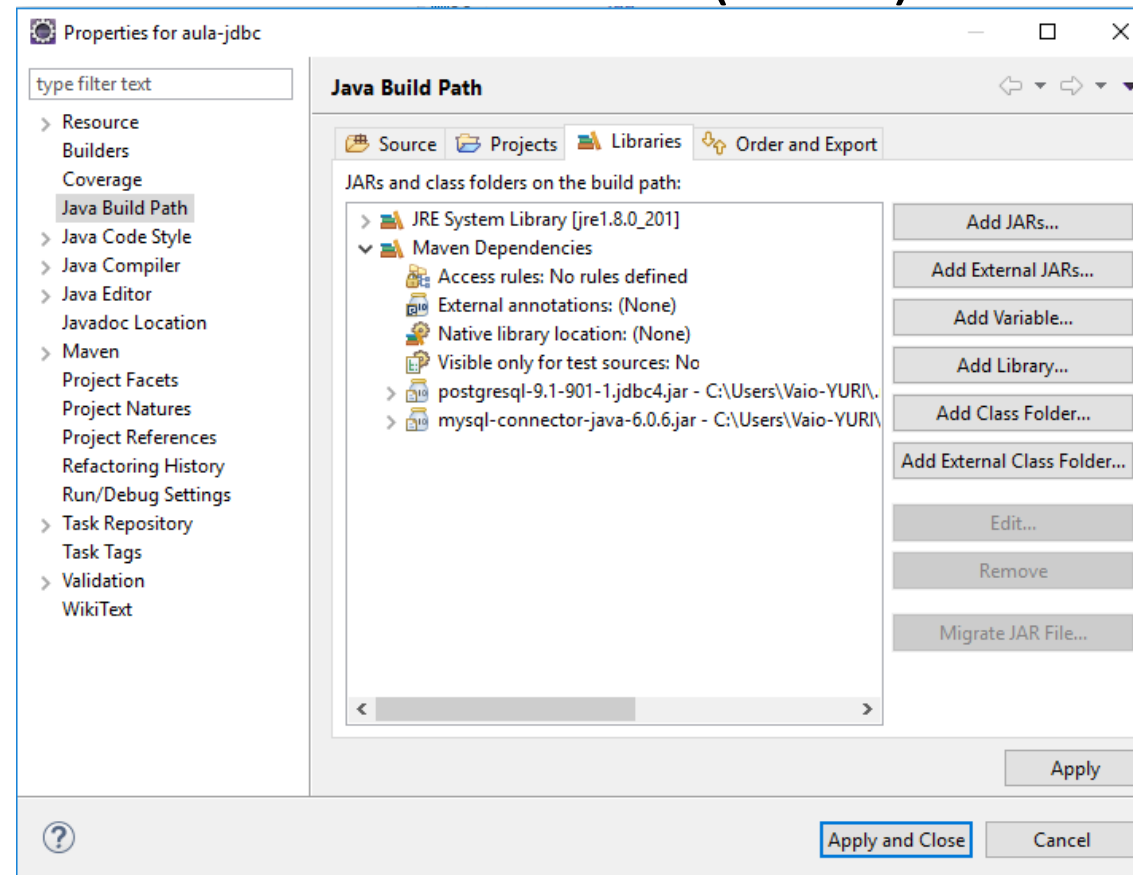
- Converte chamadas específicas do banco de dados diretamente;
- Utiliza protocolo nativo específico do BD;
- Melhor desempenho de todos os drivers;
- Independente de plataforma;

Criar um Projeto Java;

- Projeto Java:
- 1) Conectando a um banco de dados
  - Adicionar os **drivers** nas dependências do projeto Java com Maven:

```
<dependencies>
  <dependency>
    <groupId>postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>9.1-901-1.jdbc4</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>6.0.6</version>
  </dependency>
</dependencies>
```

- Projeto Java:
- 1) Conectando a um banco de dados (Cont.)



## ➤ Projeto Java:

### ➤ 1) Conectando a um banco de dados (Cont.)

#### ➤ Como invocar um **driver** JDBC no Java:

```
Class.forName("org.postgresql.Driver");
```

#### ➤ Isso registrará o driver no gerenciador e disponibilizará no programa Java;

#### ➤ Leitura do driver pela linha de comando:

```
java -jar arquivo.jar -Djdbc.drivers=org.postgresql.Driver
```

- Projeto Java:
- 2) Executando instruções SQL
  - A interface **Statement** contém funções essenciais para executar comandos SQL;

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

- Executar instruções SQL utiliza-se um dos 3 métodos:
  - execute();
  - executeUpdate();
  - executeQuery();

- Projeto Java:
- 2) Executando instruções SQL: método execute();
  - **Create Table**

```
String tableSql = "create table if not exists "  
    + "categoria_projeto(num_cat_proj serial primary key,"  
    + "descricao varchar(100) not null);";  
stm.execute(tableSql);
```



## ➤ Referências:

- SILBERSCHATZ, A.; KORTH, F.; SUDARSHA, S. Database System Concepts. 6. ed. Nova York: MC Graw Hill, 2011.
- ELMASRI, R.; NAVATHE B. Sistemas de banco de dados. 6. Ed. São Paulo, SP: Pearson Addison-Wesley, 2011.
- SRIVASTAVA Prashant. Driver Java JDBC. Medium, 2018. Acessado em 05/2019, disponível em: <https://medium.com/@prashant.srivastava7744/java-jdbc-driver-b62fd5f6a33a>.