



- ALPOO – Princípios de Banco de Dados e SQL



Professor: Me. Gustavo Molina

Introdução



- Diferentes bancos de dados relacionais possuem diferentes formas de se comunicar com uma aplicação que necessite acessar seus dados.



Introdução



- Isso causa um **grande problema de codificação e manutenção** nas aplicações que precisam trabalhar com vários bancos de dados e também requer o aprendizado de uma nova API para cada banco de dados diferentes.

JDBC



- A Sun desenvolveu a API JDBC, com a intenção de uniformizar os acessos aos diferentes bancos de dados relacionais, dando maior flexibilidade aos sistemas

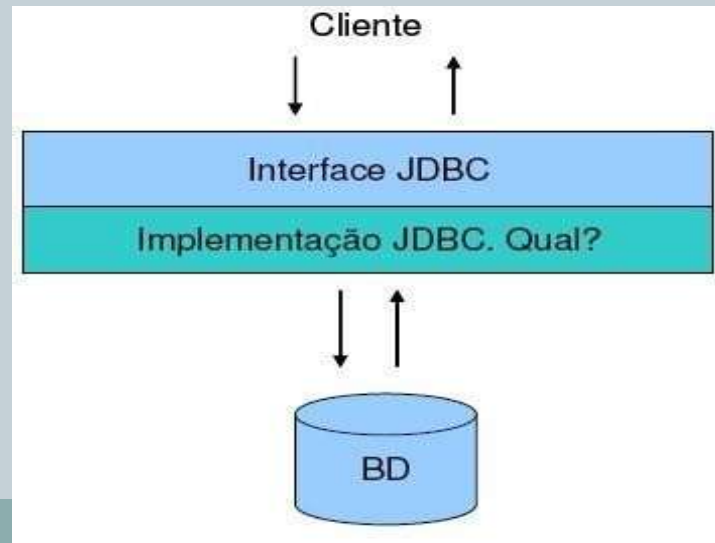
JDBC = Java Database Connectivity

- Com o JDBC as chamadas aos bancos de dados são padronizadas.

JDBC



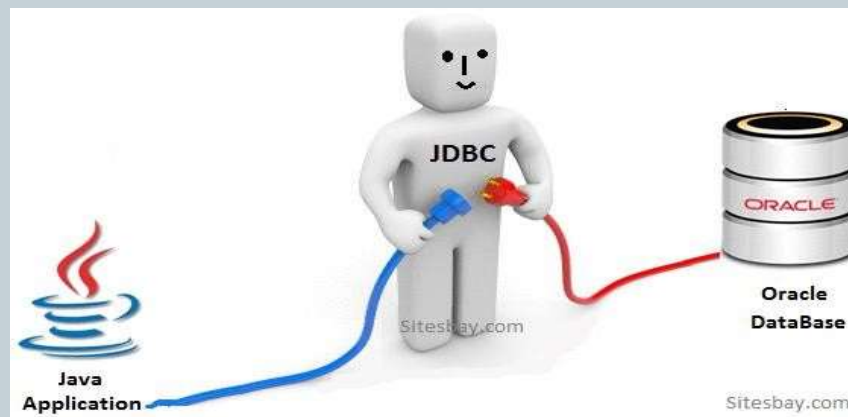
- AAPI JDBC (Java DataBase Connectivity) fornece um conjunto de classes e interfaces para manipular as Bases de Dados.
- AAPI JDBC é uma camada de abstração que permite a uma aplicação Java utilizar uma interface padrão para acessar um banco de dados relacional através da linguagem SQL.



JDBC



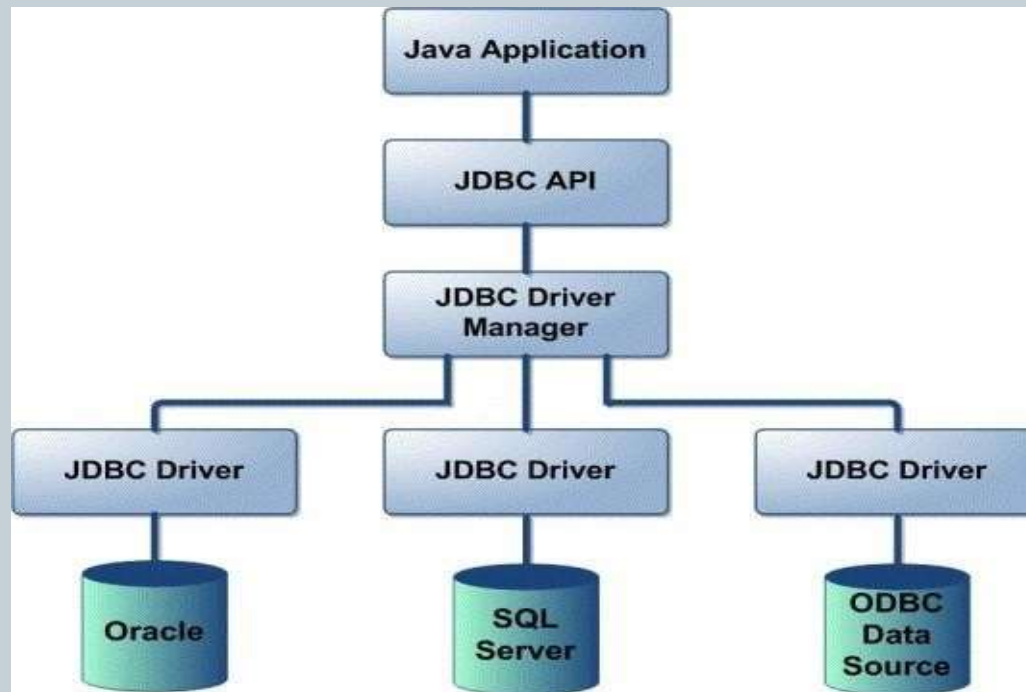
- **A Versão atual do JDBC é a 4.2.(para o JDK 8)**
- É composta pelos pacotes java.sql e javax.sql já incluídos no Java 8.
- O pacote javax.sql contém outras classes e pacotes que permitem o uso de conexões JDBC de forma mais eficiente e portátil.



Arquitetura JDBC



- **JDBC consiste em duas partes:**
 - (1) API JDBC, puramente escrita em Java;
 - (2) Gerenciador de Driver JDBC, o qual se comunica com os drivers dos fabricantes de Bancos de Dados.



Driver JDBC



- É o componente de software utilizado para que uma aplicação escrita em Java tenha acesso a um Banco de Dados.
- Todos os principais bancos de dados do mercado possuem os seus drivers JDBC.
- O driver JDBC atua como tradutor entre uma aplicação Java e o Banco de Dados, implementando um protocolo de comunicação para a transferência de consultas e resultados.

SGBD'S



PostgreSQL



ORACLE®

IBM®

DB2®

Vantagens SGBD



- Uma aplicação Java utiliza uma API JDBC única que independe do banco de dados ou driver que estiver sendo utilizado.
- Os drivers para conexão e acesso aos principais bancos de dados são fornecidos pelos seus próprios fabricantes.
- O desenvolvedor precisa apenas conhecer a API JDBC e utilizar o driver adequado.

Etapas



Sete etapas para manipulação de banco de dados em Java

- 1 Carregue o driver JDBC.** Para carregar um driver, você deve especificar o nome do driver do banco de dados que será utilizado no método `Class.forName`. Ao fazer isso, você cria automaticamente uma instância do driver que é registrado no gerenciador de drivers do JDBC.
- 2 Defina a URL de conexão.** Em JDBC, uma URL de conexão especifica o servidor host, a porta e nome do banco de dados com o qual se quer estabelecer uma conexão.
- 3 Estabeleça a conexão.** Com o nome de usuário, a URL e uma senha, a conexão de rede para o banco de dados pode ser estabelecida. Uma vez que a conexão é estabelecida, as consultas de banco de dados podem ser realizadas até que a conexão seja fechada.

Etapas



- 4 Crie um objeto do tipo Statement.** Criando um objeto do tipo Statement é possível enviar consultas e comandos para o banco de dados.
- 5 Execute uma consulta ou atualização no banco de dados.** Dado um objeto do tipo Statement, você pode enviar instruções SQL para o banco de dados usando `execute` ou os métodos `executeQuery`, `executeUpdate` ou `executeBatch`.
- 6 Processe os resultados.** Quando uma consulta de banco de dados é executada, um `ResultSet` é retornado. O `ResultSet` representa um conjunto de linhas e colunas que podem ser processados por chamadas a métodos `getXXX`.
- 7 Feche a conexão.** Quando as consultas e processamento de resultados se encerrarem, a conexão deve ser fechada, liberando os recursos do banco de dados.

Etapas



- As principais Classes e Interfaces JDBC estão listadas abaixo:
 - `public class DriverManager`
 - `public interface Connection`
 - `public interface Statement`
 - `public interface PreparedStatement`
 - `public interface ResultSet`

Exemplo de URL MySql

