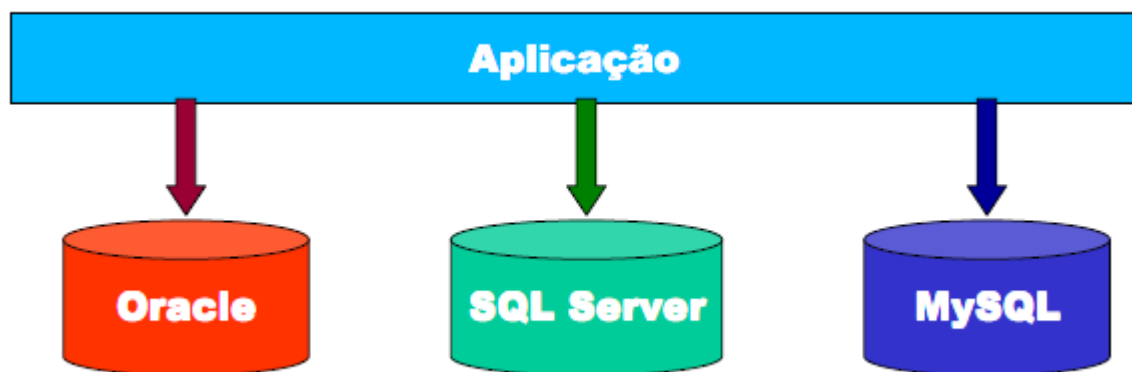


## JAVA JDBC

---

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

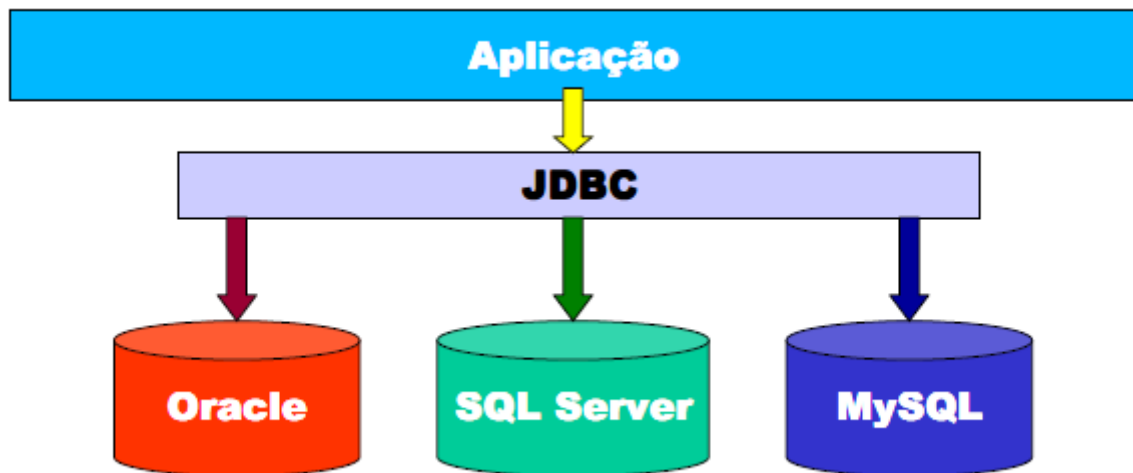


Isto causa um grande problema de codificação e manutenção nas aplicações que precisam trabalhar com vários bancos de dados e também requer o aprendizado de uma nova API para cada banco de dados diferente. Isso não torna a aplicação flexível.

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 JDBC as chamadas aos bancos de dados são padronizadas, apesar de que os comandos SQL podem variar de banco para banco, se não for usado o SQL padrão.



A biblioteca da JDBC provê um conjunto de interfaces de acesso ao banco de dados. Uma implementação em particular dessas interfaces é chamada de driver. Os próprios fabricantes dos bancos de dados (ou terceiros) são quem implementam os drivers JDBC para cada banco de dados, pois são eles que conhecem detalhes dos bancos de dados. Cada banco de dados possui um Driver JDBC específico (que é usado de forma padrão - JDBC).

A API padrão do Java já vem com o driver JDBC-ODBC, que é uma ponte entre a aplicação Java e o banco através da configuração de um recurso ODBC na máquina. Para a aplicação Java se comunicar com um banco de dados e acessar os seus dados, uma conexão com o banco de dados deve ser estabelecida. A conexão é estabelecida de seguinte forma:

- Carregamento do driver JDBC específico
- Criação da conexão com o banco de dados

A partir da conexão é possível interagir com o banco de dados, fazendo consultas, atualização e busca às informações da base e das tabelas.

---