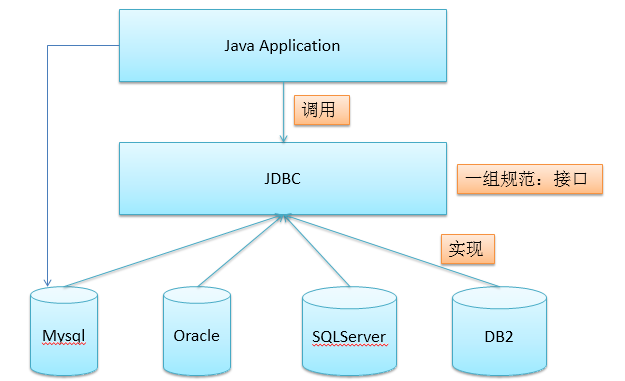
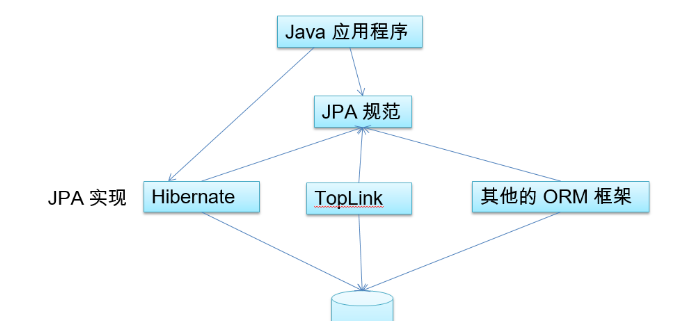
# 统一规范

## Jdbc统一访问数据库规范



## jpa统一使用orm的方式

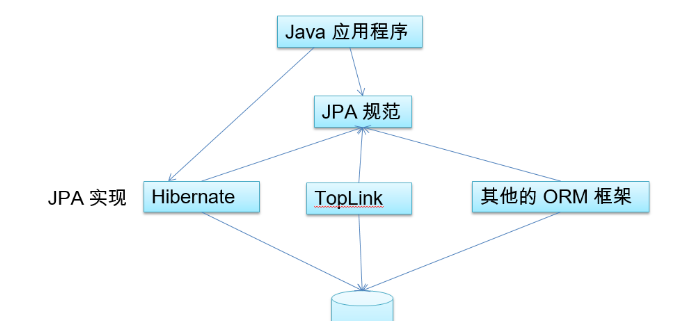


# jpa

## 什么是JPA

Java Persistence API：用于对象持久化的 API

Java EE 5.0 平台标准的 ORM 规范，使得应用程序以统一的方式访问持久层



与JDBC类似，JPA统一了java应用程序访问ORM框架的规范。

以前我们的应用程序直接使用ORM框架，如hibernate，mybatis。但是不同的框架使用方法不一样，而JPA让我们以同样的方式访问不同的ORM框架。常用的框架是hibernate。

## JPA和Hibernate的关系

JPA 是 hibernate 的一个抽象（就像JDBC和JDBC驱动的关系）：

JPA 是规范：JPA 本质上就是一种 ORM 规范，不是ORM 框架 —— 因为 JPA 并未提供 ORM 实现，它只是制订了一些规范，提供了一些编程的 API 接口，但具体实现则由 ORM 厂商提供实现

Hibernate 是实现：Hibernate 除了作为 ORM 框架之外，它也是一种 JPA 实现

从功能上来说， JPA 是 Hibernate 功能的一个子集

## Jpa的实现

JPA 的目标之一是制定一个可以由很多供应商实现的 API，目前Hibernate 3.2+、TopLink 10.1+ 以及 OpenJPA 都提供了 JPA 的实现

Hibernate：JPA 的始作俑者就是 Hibernate 的作者，Hibernate 从 3.2 开始兼容 JPA

OpenJPA：OpenJPA 是 Apache 组织提供的开源项目

TopLink：TopLink 以前需要收费，如今开源了

## JPA的优势

**标准化:** 提供相同的 API，这保证了基于JPA 开发的企业应用能够经过少量的修改就能够在不同的 JPA 框架下运行。

**简单易用，集成方便:** JPA 的主要目标之一就是提供更加简单的编程模型，在 JPA 框架下创建实体和创建 Java 类一样简单，只需要使用 javax.persistence.Entity 进行注释；JPA 的框架和接口也都非常简单，

**可媲美JDBC的查询能力:** JPA的查询语言是面向对象的，JPA定义了独特的JPQL，而且能够支持批量更新和修改、JOIN、GROUP BY、HAVING 等通常只有 SQL 才能够提供的高级查询特性，甚至还能够支持子查询。

**支持面向对象的高级特性: J**PA 中能够支持面向对象的高级特性，如类之间的继承、多态和类之间的复杂关系，最大限度的使用面向对象的模型