1. 什么是 Hibernate

ORM 框架-----对象/关系映射

Java类与数据库中-表-列名对应

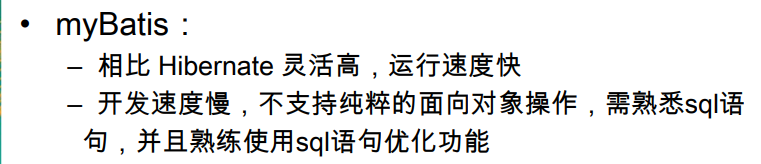
XML通常写映射

持久化框架-----数据库交互

CRUD+加载(把一个对象从数据库加载到内存中)

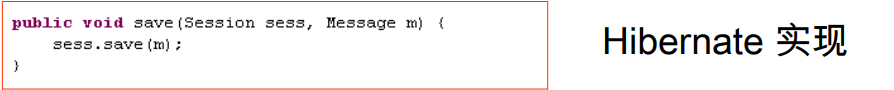
2. Hibernate

采用面向对象的方式  
 不需要掌握SQL



3. Hibernate 与 Jdbc 代码对比

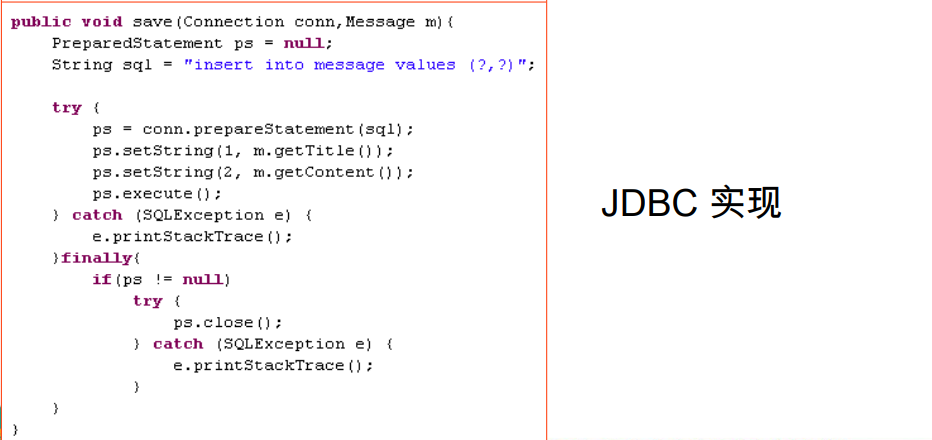
Java思想



写映射语句

填充语句

获取连接



4. hibernate使用

1.插件安装

2. hibernate的jar包+数据库驱动jar包

5.使用流程

1. hibernate.cfg.xml

hibernate配置文件

2. 创建持久化类

建数据库+表+列

3. \*.hbm.xml

对象-关系映射文件---指定Java类与数据库的映射关系

4. Hibernate API

当Java方法写具体的SQL操作

6.持久化Java类

1. 提供一个无参的构造器:使Hibernate可以使用 Constructor.newInstance() 来实例化持久化类

2. 提供一个标识属性(identifier property): 通常映射为数据库表的主 键字段. 如果没有该属性，一些功能将不起作用，如： Session.saveOrUpdate()

3. 为类的持久化类字段声明访问方法(get/set): Hibernate对 JavaBeans 风格的属性实行持久化。

4. 使用非 final 类: 在运行时生成代理是 Hibernate 的一个重要的功 能. 如果持久化类没有实现任何接口, Hibnernate 使用 CGLIB 生 成代理. 如果使用的是 final 类, 则无法生成 CGLIB 代理.

5. 重写 eqauls 和 hashCode 方法: 如果需要把持久化类的实例放到 Set 中(当需要进行关联映射时), 则应该重写这两个方法

7.\*.hbm.xml 映射文件

1.指定包所在位置

2.表与类的对应

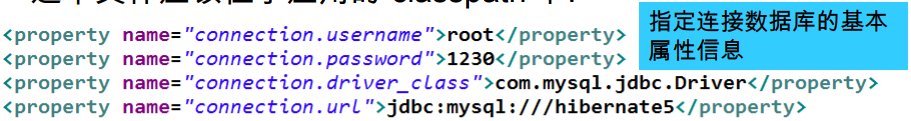
<class name=”?” table=”?”>

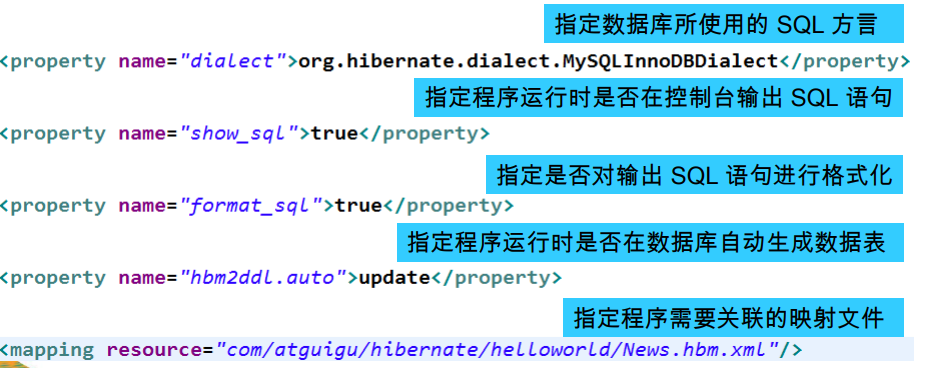
3.表类中写属性-列对应





8. Hibernate 配置文件





连什么数据库？

这个数据库独有方言是什么？

运行时候输出Sql吗？---易于排查

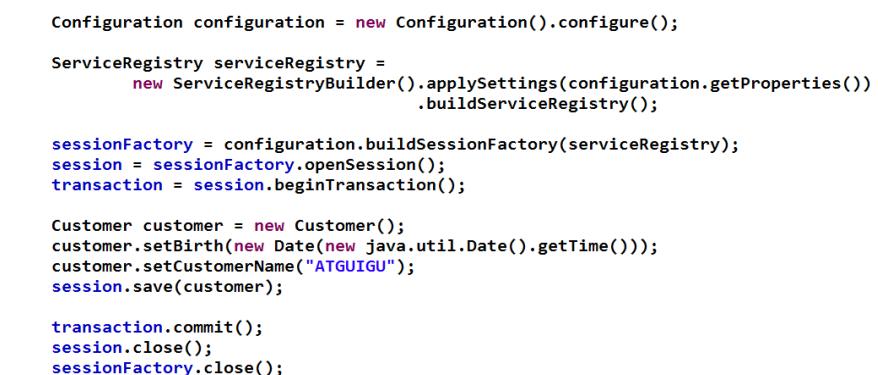
格式化SQL吗？

自动生成表吗？

映射关系是什么？在哪里？

9.实操方式

1.Java方面



配置文件---配置文件获取属性

注册类

会话工厂----通过注册类产生

Session---会话工厂产生

转换类----Session调用方法产生

实体类---Customer

实体类设置属性

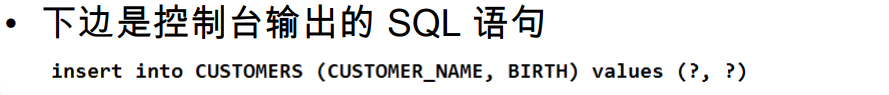
Session保存实体类

转换提交

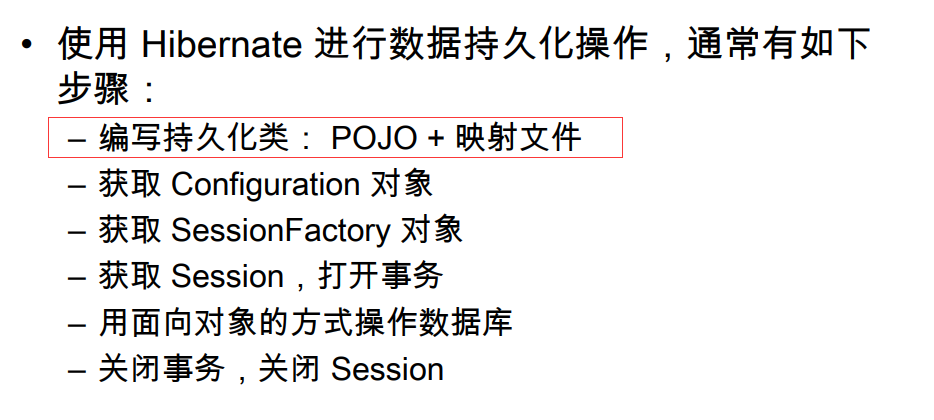
会话关闭

会话工厂关闭

2.控制台输出的SQL语句---因为配置文件里设置了输出



10.抽象化步骤



11. Configuration 类

Configuration 类负责管理 Hibernate 的配置信息。包括如下 内容：

1. – Hibernate 运行的底层信息：数据库的URL、用户名、密码、JDBC 驱动类，数据库Dialect,数据库连接池等（对应 hibernate.cfg.xml 文 件）。
2. – 持久化类与数据表的映射关系（\*.hbm.xml 文件）

创建 Configuration 的两种方式

– 属性文件（hibernate.properties）:

• Configuration cfg = new Configuration();

– Xml文件（hibernate.cfg.xml）

• Configuration cfg = new Configuration().configure();

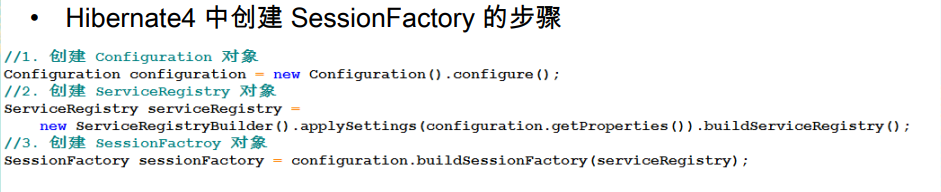
– Configuration 的 configure 方法还支持带参数的访问：

• File file = new File(“simpleit.xml”);

• Configuration cfg = new Configuration().configure(file);

12. SessionFactory 接口

• Hibernate4 新增了一个 ServiceRegistry 接口，所有基于 Hibernate 的配 置或者服务都必须统一向这个 ServiceRegistry 注册后才能生效



13. Session 接口

相当于 JDBC 中的 Connection。

1. • 持久化类与 Session 关联起来后就具有了持久化的能 力。

2.• Session 类的方法：

– 取得持久化对象的方法： get() load()

– 持久化对象都得保存，更新和删除： save(),update(),saveOrUpdate(),delete()

– 开启事务: beginTransaction().

– 管理 Session 的方法：isOpen(),flush(), clear(), evict(), close()等

14. Transaction(事务)

• 代表一次原子操作，它具有数据库事务的概念。所有 持久层都应该在事务管理下进行，即使是只读操作。

Transaction tx = session.beginTransaction();

• 常用方法:

– commit():提交相关联的session实例

– rollback():撤销事务操作

– wasCommitted():检查事务是否提交

15. Hibernate 配置文件的两个配置项

• hbm2ddl.auto：该属性可帮助程序员实现正向工程, 即由 java 代码生成数据库脚本, 进而生成具体的表结构. 。取值 create | update | create-drop | validate

– create : 会根据 .hbm.xml 文件来生成数据表, 但是每次运行都会 删除上一次的表 ,重新生成表, 哪怕二次没有任何改变

– create-drop : 会根据 .hbm.xml 文件生成表,但是SessionFactory一 关闭, 表就自动删除

– update : 最常用的属性值，也会根据 .hbm.xml 文件生成表, 但 若 .hbm.xml 文件和数据库中对应的数据表的表结构不同, Hiberante 将更新数据表结构，但不会删除已有的行和列

– validate : 会和数据库中的表进行比较, 若 .hbm.xml 文件中的列在 数据表中不存在，则抛出异常

• format\_sql：是否将 SQL 转化为格式良好的 SQL . 取值 true | false

16. Session 操纵对象