**第1章 Hibernate初识**

1-1 课程介绍

1-2 什么是ORM

·ORM(Object/Relationship Mapping)：对象/关系映射

利用面向对象思想编写的数据库应用程序最终都是把对象信息保存在关系型数据库中， 于是要编写很多和底层数据库相关的SQL语句。

·写SQL语句有什么不好吗？

·不同的数据库使用的SQL语法不同。比如：PL/SQL与T/SQL

·同样的功能在不同的数据库中有不同的实现方式。比如分页SQL

·程序过分依赖SQL对程序的移植及扩展，维护等带来很大的麻烦。

·有没有办法让程序员彻底抛弃书写SQL的思想，完全的使用面向对象思想开发软件呢？

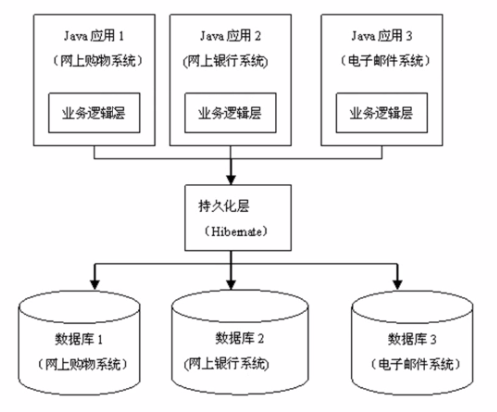
·需要一种可行的ROM框架技术

1-3 Hibernate简介

·什么是Hibernate

·Hibernate是Java领域的一款开源的ORM框架技术。

·Hibernate对JDBC进行了非常轻量级的对象封装



·其他主流的ORM框架技术

·其他主流的ORM框架技术：

·MyBatis：前身就是著名的iBatis

·Toplink：后被Oracle收购，并重新包装为Oracle As Toplink

·EJB：本身是JAVAEE的规范

1-4 Hibernate开发前准备

·编写第一个Hibernate例子

·开发工具：Eclipse

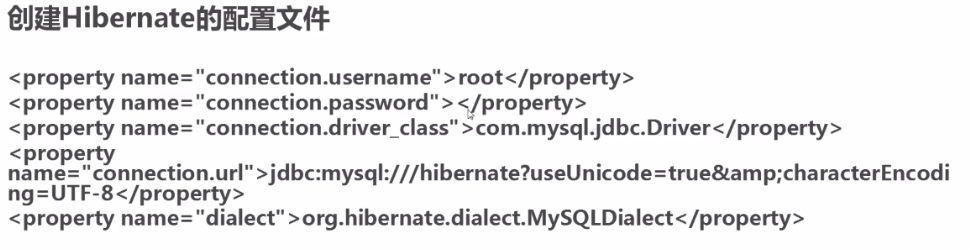
·Hibernate Tools for Eclipse Plugins

Hibernate Tools是由JBoss推出的一个Eclipse综合开发工具插件，该插件可以简化ORM框架Hibernate以及JBoss Seam，EJB3等的开发工作。

1-5 编写第一个Hibernate例子

1-6 创建hibernate工程

1-7 hibernate配置文档



1-8 创建持久化类

1-9 创建对象-关系映射文件和数据库

1-10 使用JUnit进行测试

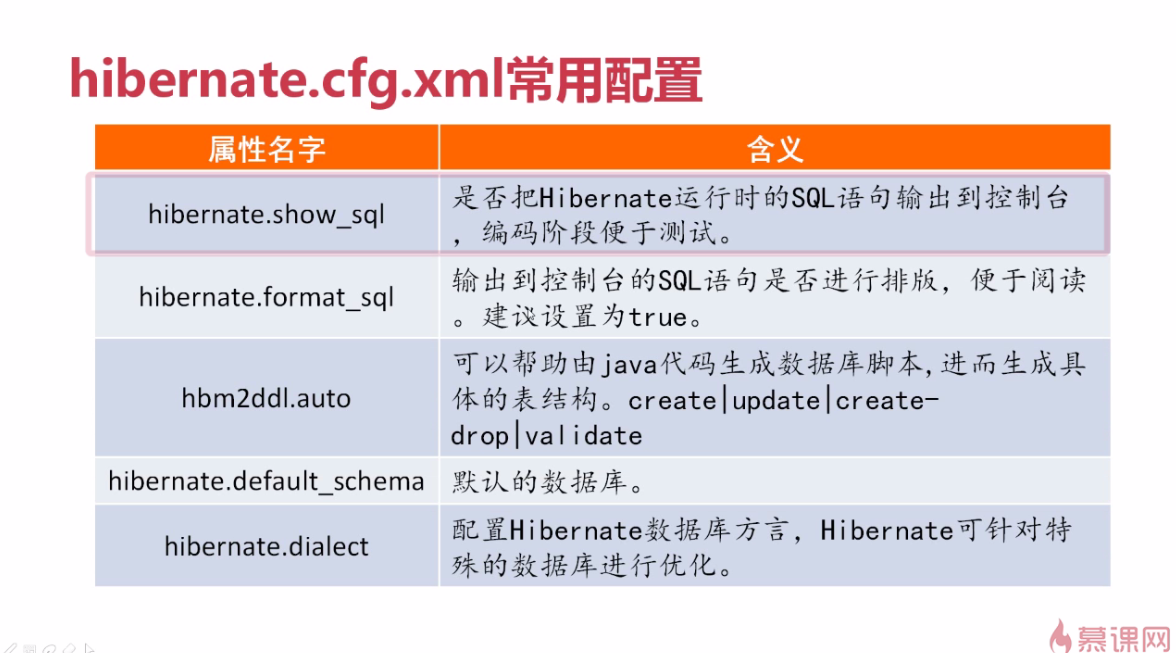
1-11 通过Hibernate API编写访问数据库的代码



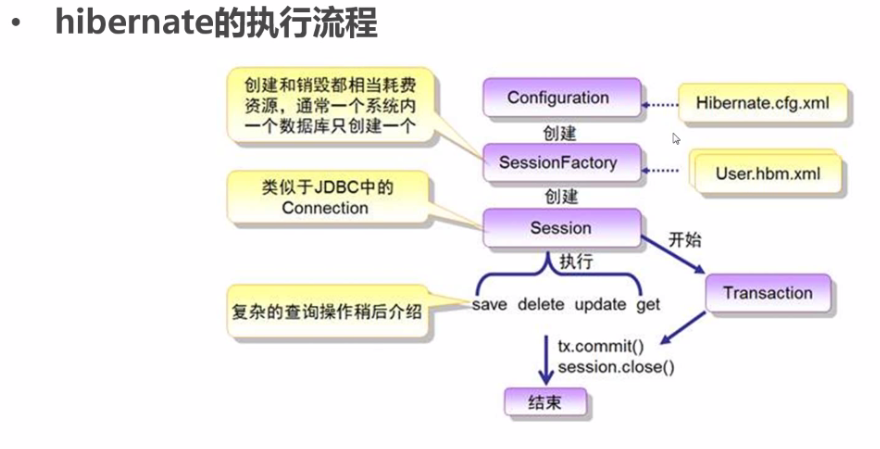
**第2章 Hibernate进阶**

2-1 本章简介

2-2 hibernate常用配置



2-3 session简介



·session 与connection，是一对多关系，每个session都有一个与之对应的connection,一个connection不同时刻可以供多个session使用。

·不建议直接使用jdbc的connection操作数据库，而是通过session操作数据库。

·session可以理解为操作数据库的对象

·session与connection，是多对一关系，每个session都有一个与之对应的connection,一个connection不同时刻可以供多个session使用。

·把对象保存在关系数据库中需要调用session的各种方法，如： save(),update(),delete(),createQuery()等。

2-4 transaction简介

·hibernate对数据的操作都是封装在事务当中，并且默认是非自动提交的方式。所以用session保存对象时，如果不开启事务，并且手工提交事务，对象并不会真正保存在数据库中。

·如果你想让hibernate像jdbc那样自动提交事务，必须调用session对象的doWork()方法，获得jdbc的connection后，设置其为自动提交事务模式。(注意:通常并不推荐这样做)

·transaction = session.beginTransaction();

...

transaction.commit();//提交事务

·session.doWork(new Work(){

@Override

public void execute(Connection connection) throws SQLException {

connection.setAutoCommit(true);

}

});

session.save(s);//保存对象进入数据库

session.flush();

2-5 session详解（上）

·如何获得session对象

·openSession

·getCurrentSession

·如果使用getCurrentSession需要在hibernate.cfg.xml文件中进行配置：

·如果是本地事务（jdbc事务）

<propertyname="hibernate.current\_session\_context\_class">thread</property>

·如果是全局事务（jta事务）

<propertyname="hibernate.current\_session\_context\_class">jta</property>

2-6 session详解（下）

·openSession与getCurrentSession的区别

·getCurrentSession在事务提交或者回滚之后会自动关闭，而openSession需要你手工 关闭。如果使用openSession而没有手动挂你，多次之后会导致连接池溢出。

·openSession每次创建新的session对象，getCurrentSession使用现有的session对象

2-7 hbm配置文档

·<hibernate-mapping

schema="schemaName"

catalog="catalogName"

default-cascade="cascade\_style" //级联风格

default-access="field|property|ClassName" //访问策略

default-lazy="true|false" //加载策略

package="packagename"

/>

·<class

name="ClassName"

table="tableName"

batch-size="N" //一次可以抓取N条记录

where="condition" //抓取条件

entity-name="EntityName" //一个实体类可以映射很多表

/>

·<id //主键

name="propertyName"

type="typename" //对应的java类型

column="column\_name"

length="length" //数据长度

<generator class="generatorClass"/> //主键生成策略

</id>

·主键生成策略（generatorClass）



·<generator>表示一个主键的生成机制。即具体通过何种方式来生成。   
 生成方式包括：  
 ·increment：生成long, short或者int类型的主键，不能在cluster环境下使用。适 用于所有数据库   
 ·identity：生成long, short或者int类型的主键。适用于[DB](https://www.baidu.com/s?wd=DB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank)2, MySQL, MS SQL Server, [Sybase](https://www.baidu.com/s?wd=Sybase&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank) and HypersonicSQL   
 ·sequence ：生成long, short或者int类型的主键。适用于[DB](https://www.baidu.com/s?wd=DB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank)2, PostgreSQL, [Oracle](https://www.baidu.com/s?wd=Oracle&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank), [SAP](https://www.baidu.com/s?wd=SAP&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank) [DB](https://www.baidu.com/s?wd=DB&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank), McKoi，Interbase.   
 ·hilo：生成long, short或者int类型的主键。需要提供一个数据库的表来存放生成 的主键信息。当采用应用服务器的[JTA](https://www.baidu.com/s?wd=JTA&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank)提供的数据库连接或者用户自定义的数据库 连接的时候，不要使用这种主键生成方式。适用于所有数据库   
 ·seqhilo：采用给定的数据库的sequence来生成long, short或者int类型的主键。 适用于DB2, PostgreSQL, [Oracle](https://www.baidu.com/s?wd=Oracle&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank), [SAP](https://www.baidu.com/s?wd=SAP&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank) DB, McKoi，Interbase.   
 ·[uuid](https://www.baidu.com/s?wd=uuid&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank).hex：采用128位的算法来生成一个32位字符串。最通用的一种方式。适用于所 有数据库   
 ·[uuid](https://www.baidu.com/s?wd=uuid&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank).string：同样采用128位的[UUID](https://www.baidu.com/s?wd=UUID&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YLmycknyf1n16vmHN-PWfY0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHR1rHRYnW0dnWcknjTvnHcvPs" \t "https://zhidao.baidu.com/question/_blank)算法。将生成的字符编码位16位。适用于除 PostgreSQL.以外的数据库   
 ·native：根据具体连接的数据库从identity, sequence或者hilo选择一种来生成主键。 适用的数据库根据选择的生成方式确定。   
 ·assigned： 交给应用自己给主键赋值。要注意的是赋值必须在调用save()方法之前完 成。适用的数据库根据选择的生成方式确定。

**第3章 Hibernate单表操作**

3-1 本章简介

·单一主键

·基本类型

·对象类型

·组件属性

·单表操作CRUD实例

3-2 [Hibernate单表操作] 单一主键

·assigned 由java应用程序负责生成（手工赋值）。

·native 由底层数据库自动生成标示符，如果是MySQL就是increment，如果是Oracle就是sequence等等。此时人为赋值不起作用

·若基本数据类型作为类成员的时候，即使没有进行初始化。java也会给定默认的初始值。默认是如下所示：

boolean        false

char              null

byte              (byte)0

short             (short)0

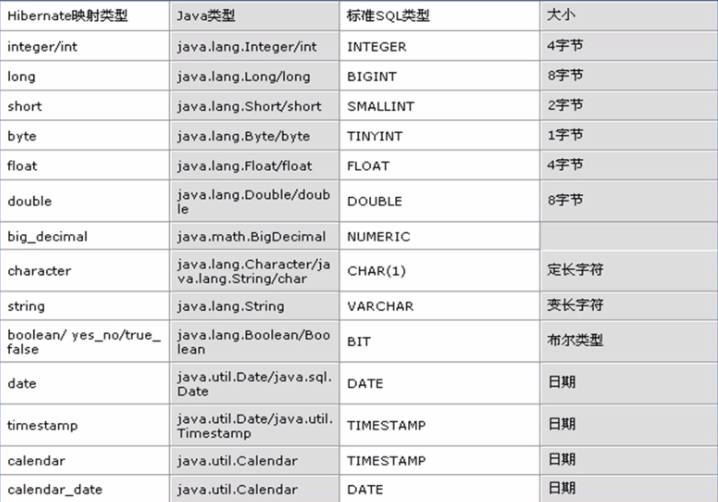
int                  0

long               0L

float               0.0f

double           0.0d

3-3 [Hibernate单表操作] 基本类型





3-4 [Hibernate单表操作] 对象类型



·MySQL不支持标准SQL的CLOB类型，在MySQL中，用TEXT，MEDIUMTEXT及LONGTEXT类型来表示长度超过255的长文本数据。

3-5 [Hibernate单表操作] 组件属性

·实体类中的某个属性属于用户自定义的类的对象

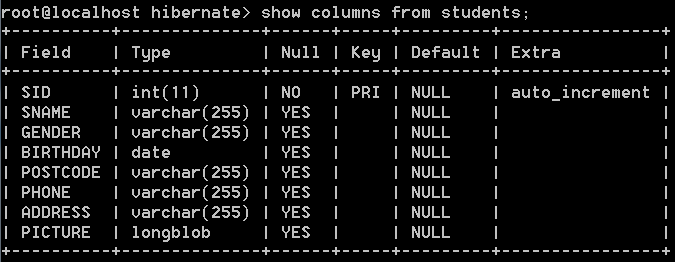
<component name="address" class="Address">

<property name="postcode" column="POSTCODE"> </property>

<property name="phone" column="PHONE"> </property>

<property name="address" column="ADDRESS"> </property>

</component>



3-6 [Hibernate单表操作] 单表操作CRUD实例

·单表CRUD操作实例

·save

·update

·delete

·get/load （查询单个记录）

·get与load区别

·在不考虑缓存的情况下，get方法会在调用之后立即向数据库发出sql语句，返回持 久化对象。

load方法会在调用后返回一个代理对象。该代理对象只保存了实体对象的id，直到使 用对象的非主键属性时才 会发出sql语句。

·查询数据库中不存在的数据时，get方法返回null，load方法抛出异常 org.hibernate.ObjectNotFoundException

· Students s1 = (Students)session.load(Students.class, 2);

Students s2= (Students)session.load(Students.class, 2);

System.out.println(s1.getClass().getName());

System.out.println(s2.getClass().getName());

输出：Students

Students\_$$\_javassist\_0

**第4章 课程总结**

4-1 Hibernate课程总结

·什么是ORM？为什么要使用Hibernate？

·Hibernate开发的基本步骤？

·编写配置文档hibernate.cfg.xml

·编写实体类

·生成对应实体类的映射文件并添加到配置文档中

·调用Hibernate API进行测试

·什么是session？

·openSession与getCurrentSession

·单表操作常用方法有哪些？

save delete update get load

·get与load