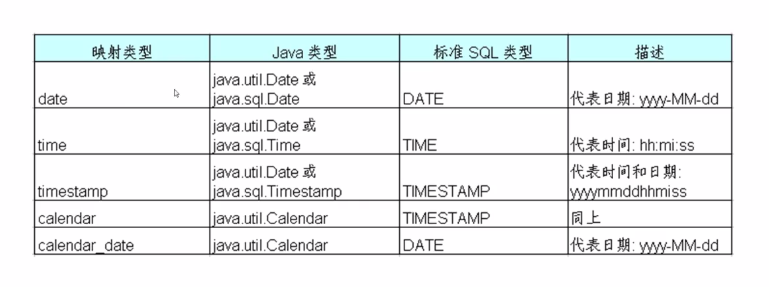
一．单一主键：

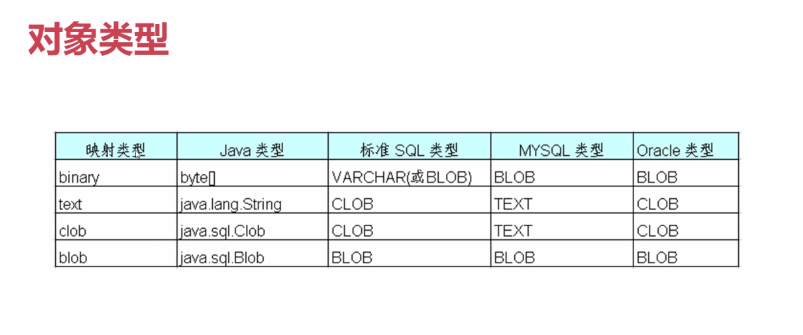
（1）assigned 由java应用程序负责生成（手工赋值）（必须要手工赋值，不然造成主键冲突）  
（2） native 由底层数据库自动生成标示符，如果是MySQL就是increment，如果是Oracle就是sequence，等等(自动增长)  
Ps1：assigned注意：如果实体类中设置的主键id是基本类型int的话，则可以不用赋值，系统默认值为0；如是引用类型Integer话，则默认值为null，不赋值系统则报错。  
Ps2：native注意：系统会自动选择该数据库对应的自动增值方式，从1开始。即使手动给他赋值，也不会起作用，但也不会报错。

二、数据类型





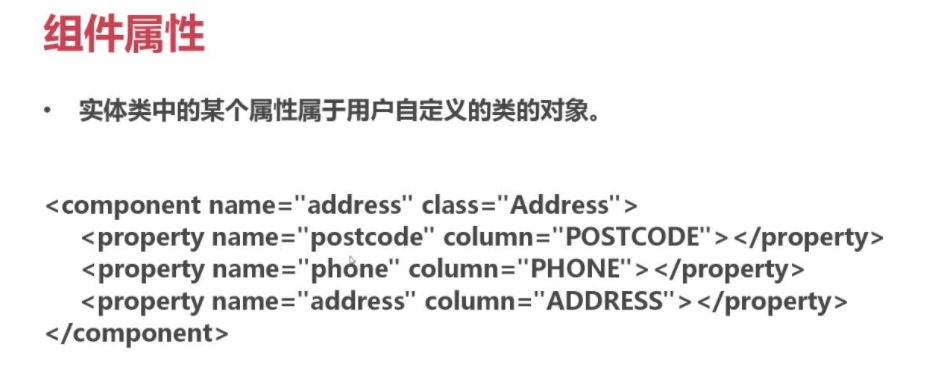
三、对象类型



Java.sql.Clob 大文本类型

Java.sql.Blob 大的数据类型

四、组件属性



<component name=”address” class=”package.Address”>

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

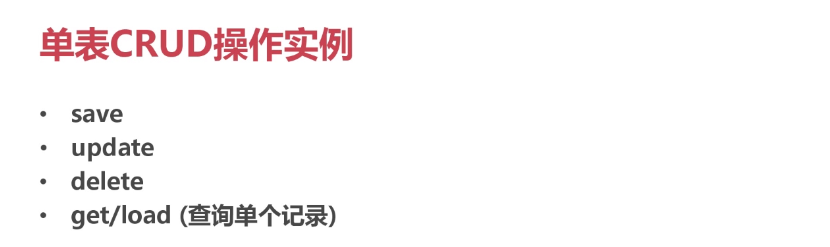
<property name=”postcode column=”POSTCODE”></property>

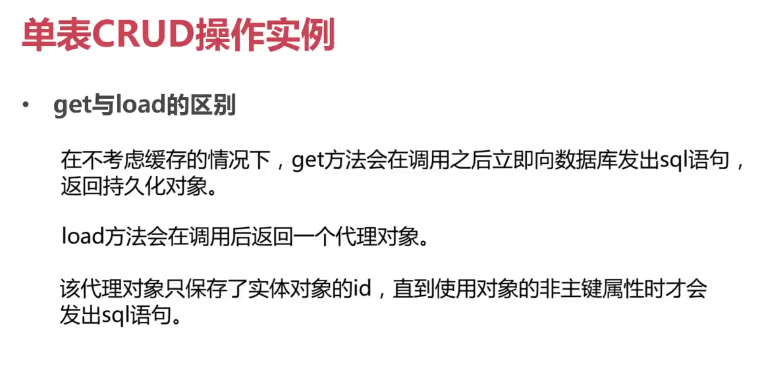
<property name=”tel” column=”TEL”></property>

</component>

Ps：Class 一定要写包名.类名；

五、单表增删改查操作





查询数据库中不存在的东西的时候，get方法返回的是null。

Load方法会报错。

**hibernate 简介：**  
hibernate是一个开源框架，它是对象关联关系映射的框架，它对JDBC做了轻量级的封装，而我们java程序员可以使用面向对象的思想来操纵数据库。  
hibernate核心接口  
session：负责被持久化对象CRUD操作  
sessionFactory:负责初始化hibernate，创建session对象  
configuration:负责配置并启动hibernate，创建SessionFactory  
Transaction:负责事物相关的操作  
Query和Criteria接口：负责执行各种数据库查询  
  
**hibernate工作原理：**  
1.通过Configuration config = new Configuration().configure();//读取并解析hibernate.cfg.xml配置文件  
2.由hibernate.cfg.xml中的<mapping resource="com/xx/User.hbm.xml"/>读取并解析映射信息  
3.通过SessionFactory sf = config.buildSessionFactory();//创建SessionFactory  
4.Session session = sf.openSession();//打开Sesssion  
5.Transaction tx = session.beginTransaction();//创建并启动事务Transation  
6.persistent operate操作数据，持久化操作  
7.tx.commit();//提交事务  
8.关闭Session  
9.关闭SesstionFactory

**为什么要用hibernate：**  
1. 对JDBC访问数据库的代码做了封装，大大简化了数据访问层繁琐的重复性代码。  
2. Hibernate是一个基于JDBC的主流持久化框架，是一个优秀的ORM实现。他很大程度的简化DAO层的编码工作  
3. hibernate使用Java反射机制，而不是字节码增强程序来实现透明性。  
4. hibernate的性能非常好，因为它是个轻量级框架。映射的灵活性很出色。它支持各种关系数据库，从一对一到多对多的各种复杂关系。

 Hibernate是如何延迟加载?get与load的区别

1. 对于Hibernate get方法，Hibernate会确认一下该id对应的数据是否存在，首先在session缓存中查找，然后在二级缓存中查找，还没有就查询数据库，数据 库中没有就返回null。

2. Hibernate load方法加载实体对象的时候，根据映射文件上类级别的lazy属性的配置(默认为true)，分情况讨论：

(1)若为true,则首先在Session缓存中查找，看看该id对应的对象是否存在，不存在则使用延迟加载，返回实体的代理类对象(该代理类为实体类的子类，由CGLIB动态生成)。等到具体使用该对象(除获取OID以外)的时候，再查询二级缓存和数据库，若仍没发现符合条件的记录，则会抛出一个ObjectNotFoundException。

(2)若为false,就跟Hibernateget方法查找顺序一样，只是最终若没发现符合条件的记录，则会抛出一个ObjectNotFoundException。

这里get和load有两个重要区别:

如果未能发现符合条件的记录，Hibernate get方法返回null，而load方法会抛出一个ObjectNotFoundException。

load方法可返回没有加载实体数据的代 理类实例，而get方法永远返回有实体数据的对象。

总之对于get和load的根本区别，一句话，hibernate对于 load方法认为该数据在数据库中一定存在，可以放心的使用代理来延迟加载，如果在使用过程中发现了问题，只能抛异常；而对于get方 法，hibernate一定要获取到真实的数据，否则返回null。

Hibernate中怎样实现类之间的关系?(如：一对多、多对多的关系)

类与类之间的关系主要体现在表与表之间的关系进行操作，它们都是对对象进行操作，我们程序中把所有的表与类都映射在一起，它们通过配置文件中的many-to-one、one-to-many、many-to-many、

说下Hibernate的缓存机制：

Hibernate缓存的作用：

    Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据

Hibernate缓存分类：

  Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存

Hibernate一级缓存又称为“Session的缓存”，它是内置的，意思就是说，只要你使用hibernate就必须使用session缓存。由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。在第一级缓存中，持久化类的每个实例都具有唯一的OID。

Hibernate二级缓存又称为“SessionFactory的缓存”，由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。

什么样的数据适合存放到第二级缓存中？

1 很少被修改的数据

2 不是很重要的数据，允许出现偶尔并发的数据

3 不会被并发访问的数据

4 常量数据

不适合存放到第二级缓存的数据？

1经常被修改的数据

2 .绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发

3 与其他应用共享的数据。

Hibernate查找对象如何应用缓存？

当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；查不到，如果配置了二级缓存，那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照ID放入到缓存

删除、更新、增加数据的时候，同时更新缓存

Hibernate管理缓存实例

无论何时，我们在管理Hibernate缓存（Managing the caches）时，当你给save()、update()或saveOrUpdate()方法传递一个对象时，或使用load()、 get()、list()、iterate() 或scroll()方法获得一个对象时, 该对象都将被加入到Session的内部缓存中。

当随后flush()方法被调用时，对象的状态会和数据库取得同步。 如果你不希望此同步操作发生，或者你正处理大量对象、需要对有效管理内存时，你可以调用evict() 方法，从一级缓存中去掉这些对象及其集合。

 Hibernate的查询方式

Sql、Criteria,object comptosition

Hql：

1、 属性查询

2、 参数查询、命名参数查询

3、 关联查询

4、 分页查询

**如何优化Hibernate**？

1.使用双向一对多关联，不使用单向一对多

2.灵活使用单向一对多关联

3.不用一对一，用多对一取代

4.配置对象缓存，不使用集合缓存

5.一对多集合使用Bag,多对多集合使用Set

6. 继承类使用显式多态

7. 表字段要少，表关联不要怕多，有二级缓存撑腰