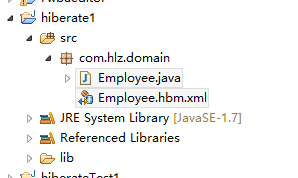


1.1



# 1.上面的Empolyee.hbm.xml就是这个对象关系映射文件，表示表和这个domain类的对应关系

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<!--该文件要清楚地表述出 类 和 表 的对应关系-->

<!-- 映射文件需要一个dtd来制定格式-->

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<!-- package : 表示该类在哪个包下 -->

<hibernate-mapping package=*"com.hsp.domain"*>

<!-- name : 表示类名 table 表示 该类和哪个表映射 -->

<class name=*"Employee"* table=*"employee"*>

<!-- id元素专门用于指定主键是如何生成,hibernate设计者认为，我们每一个表都应该有一个主键 -->

<!-- name:表示类的哪个属性是主键 -->

<id name=*"id"* type=*"java.lang.Integer"*>

<generator class=*"increment"*/>

<!-- 指定主键生成策略 -->

<!--

<generator class="sequence">

<param name="sequence">emp\_seq</param>

</generator> -->

</id>

<property name=*"name"* type=*"java.lang.String"*>

<column name=*"name"* not-null=*"true"*/>

</property>

<property name=*"email"* type=*"java.lang.String"*>

<column name=*"email"* not-null=*"true"*/>

</property>

<property name=*"hiredate"* type=*"java.util.Date"*>

<column name=*"hiredate"* not-null=*"true"*/>

</property>

</class>

</hibernate-mapping>

3.按照上面的表示添加一个序列

<!-- name:表示类的哪个属性是主键 -->

<id name=*"id"* type=*"java.lang.Integer"*>

<!-- 指定主键生成策略 -->

<generator class=*"sequence"*>

<param name=*"sequence"*>emp\_seq</param>

</generator>

</id>

# 2.创建一个序列，将来用于主键的自增长 :

--创建一个序列 ，但是MySQL是没有序列的

create sequence emp\_seq

start with 1

increment by 1

minvalue 1

nomaxvalue

nocycle

nocache

之前的数据库中的雇员表是，因为这个数据库是MySQL不是orcal，所以和老师讲的不一样。

create table employee(

id int primary key,

name varchar(64) not null,

email varchar(64) not null,

hiredate date not null);

我们使用第二种开发项目

创建employe 表.

create table employee(

id number primary key,

name varchar2(64) not null,

email varchar2(64) not null,

hiredate date not null)

创建一个序列，将来用于主键的自增长 :

--创建一个序列

create sequence emp\_seq

(start with 1,

increment by 1,

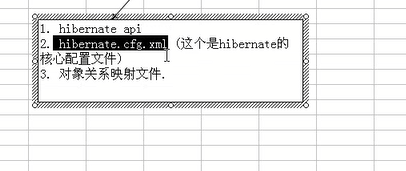
minvalue 1,

nomaxvalue,

nocycle,

nocache)

# 3.上面的步奏完成了 3，这里下一步就是第二个了



# 4.这里是手动配置的，但是其实将来做的时候都是自动配置的

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<!-- 这个我们自己配置的 -->

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- 将来改变的数据库的时候，只要改了下面这个就是可以了啦 -->

<!-- 该文件用于配置连接数据的种类,用户名，密码,ul ,驱动.. 连接池,二级缓存.. 有点类似strus struts-config.xml -->

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/hiberate1</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>147094</property>

<!-- 配置数据库的方言/ -->

<property name=*"dialect"*>org.hibernate.dialect.MySQLDialect</property>

<!-- 配置显示hibernate生成的 sql ,特别说明，在开发阶段设为true利于调试，在使用项目则设为false--><property name=*"show\_sql"*>true</property>

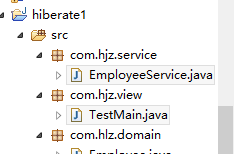
<!-- 配置管理的对象映射文件 -->

<mapping resource=*"com/hlj/domain/Employee.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

# 5然后我们建立service 和 view 界面，不管用不用，先建立起来



# 6.然后我们首先不用service，直接使用view 来进行操作数据库

public class TestMain {

public static void main(String[] args) {

// TODO Auto-generated method stub

//对数据持久层操作

//1. 该类主要是用于读取hibernate.cfg.xml

Configuration cf=new Configuration().configure();

//2. 创建一个会话工厂.是一个重量级的对象

SessionFactory sf=cf.buildSessionFactory();

//3. 创建Session(会话)<->Connection

Session session=sf.openSession();

//4.没有下面事务，下面的东西是提交不了，也就是保存不到数据中的 ，

//开始事务

Transaction ts=session.beginTransaction();

//加入一个对象(记录)Employee.java

Employee u=new Employee();

u.setEmail("1318830916@qq.com");

u.setHiredate(new Date());

u.setName("zhangyujin");

//insert

//保存，

session.save(u);//添加该对象到数据库 //等于==inset into 。。被hibernate封装，将来不管用什么数据库都是可以试实现的

ts.commit();//提交事务

session.close();

}

}

# 7.直接运行出现的就是下面的东西了

14 [main] INFO org.hibernate.cfg.Environment - Hibernate 3.3.1.GA

19 [main] INFO org.hibernate.cfg.Environment - hibernate.properties not found

24 [main] INFO org.hibernate.cfg.Environment - Bytecode provider name : javassist

522 [main] INFO org.hibernate.connection.DriverManagerConnectionProvider -

816 [main] INFO org.hibernate.cfg.SettingsFactory - Automatic session close at end of transaction: disabled

816 [main] INFO org.hibernate.cfg.SettingsFactory - JDBC batch size: 15

816 [main] INFO org.hibernate.cfg.SettingsFactory - JDBC batch updates for versioned data: disabled

817 [main] INFO org.hibernate.cfg.SettingsFactory - Scrollable result sets:

e.hql.ast.ASTQueryTranslatorFactory - Using ASTQueryTranslatorFactory

820 [main] INFO org.hibernate.cfg.SettingsFactory - Query language substitutions: {}

820

……… ……… …… ….

877 [main] INFO org.hibernate.impl.SessionFactoryImpl - building session factory

1036 [main] INFO org.hibernate.impl.SessionFactoryObjectFactory - Not binding factory to JNDI, no JNDI name configured

Hibernate: select max(id) from employee

Hibernate: insert into employee (name, email, hiredate, id) values (?, ?, ?, ?)