


Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# Java私塾 《深入浅出学Hibernate4》 ——系列精品教程

10101010101010101010101010101

私塾在线<http://si.shuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 私塾在线公告

n 私塾在线 <http://sishuok.com?frombook> 独家提供本教程的教学视频

n 私塾在线独家提供，已有精品视频课程：

Java基础、Java中级、Java高级、Java项目、Java Web开发、Java Web项目(中移动科技综合管理)、数据库SQL、PL/SQL、Jquery、研磨Struts2、Hibernate4实战、Spring3开发实战、log4j&slf4j实战、ant实战、maven实战 等

n 正在制作中的精品课程介绍：

Android开发和项目实战、Ext开发到实战、数据库设计和性能优化、WebLogic从入门到精通、JBPM4的开发、Spring MVC开发、Spring Security开发、SSH+JBPM综合项目、保险业核心业务系统开发、金融业核心业务系统开发。。。更多更多

机不可失，失不再来，敬请各位朋友奔走相告，Hold住这给力的惊喜！  
后续还会推出更多的活动，敬请关注私塾在线的公告，早加入早享受！

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程概览

#### n Hibernate的关系映射

包括：关系的表示、集合映射、关系映射、双向一对多映射示例、过滤器

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



## 《深入浅出学Hibernate4开发》——系列精品教程

### 第四部分：关系映射

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系的数据库表示

n 数据表之间的关系分为三类：一对一、一对多、多对多

n 一对一数据表（部门表 和 部门主管表）

uuid	depName
Dep1	开发部

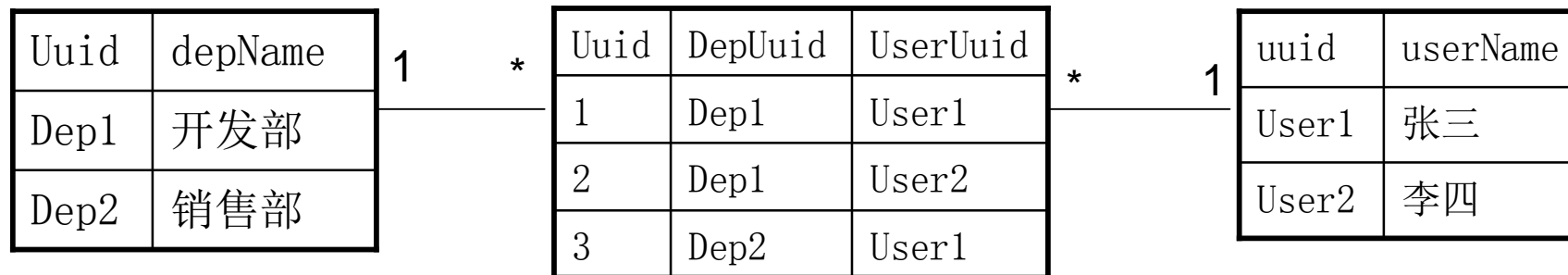
Uuid	Name	depUuid
1	张三	Dep1

n 一对多数据表（部门表 和 部门下的人员表）

Uuid	depName
Dep1	开发部

Uuid	userName	depUuid
User1	张三	Dep1
User2	李四	Dep1

n 多对多数据表（部门表 和 人员表）



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系的对象表示

**n** 根据相互寻找的关系又分：单向和双向

**n** 对象一对一（双向）

```
public class A {  
    private B b = null; }  
public class B {  
    private A a = null; }
```

**n** 对象一对多（双向）

```
public class A {  
    private B b = null; }  
public class B {  
    private Collection<A> colA = null; }
```

**n** 对象多对多（双向）

```
public class A {  
    private Collection<B> colB = null; }  
public class B {  
    private Collection<A> colA = null; }
```

**n** 双向一对多是最常用的映射关系

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### key的配置

#### n <key>元素

<key> 元素在父映射元素定义了对新表的连接，并且在被连接表中定义了一个外键引用原表的主键的情况下经常使用。

```
<key  
  column="columnname" (1)  
  on-delete="noaction|cascade" (2)  
  property-ref="propertyName" (3)  
  not-null="true|false" (4)  
  update="true|false" (5)  
  unique="true|false" (6)  
>
```

(1)column（可选）：外键字段的名称。也可以通过嵌套的 <column> 指定。

(2)on-delete（可选，默认是 noaction）：表明外键关联是否打开数据库级别的级联删除。

(3)property-ref（可选）：表明外键引用的字段不是原表的主键（提供给遗留数据）。

(4)not-null（可选）：表明外键的字段不可为空，意味着无论何时外键都是主键的一部分。

(5)update（可选）：表明外键决不应该被更新，这意味着无论何时外键都是主键的一部分。

(6)unique（可选）：表明外键应有唯一性约束，这意味着无论何时外键都是主键的一部分。

对那些看重删除性能的系统，推荐所有的键都应该定义为 on-delete="cascade"，这样 Hibernate 将使用数据库级的 ON CASCADE DELETE 约束，而不是多个 DELETE 语句

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合映射的配置-1

- n 用于映射集合类的Hibernate映射元素取决于接口的类型。比如，<set>元素用来映射Set类型的属性：

```
<class name="Product">
  <id name="serialNumber" column="productSerialNumber"/>
    <set name="parts">
      <key column="productSerialNumber" not-null="true"/>
      <one-to-many class="Part"/>
    </set>
</class>
```

- n 除了<set>, 还有<list>, <map>, <bag>, <array> 和 <primitive-array> 映射元素。<map>具有代表性，如下：

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合映射的配置-2

```
n  <map
    name="propertyName" (1)
    table="table_name" (2)
    schema="schema_name" (3)
    lazy="true|extra|false" (4)
    inverse="true|false" (5)
    cascade= "all|none|save-update|delete|all-delete-orphan|delete-orphan" (6)
    sort="unsorted|natural|comparatorClass" (7)
    order-by="column_name asc|desc" (8)
    where="arbitrary sql where condition" (9)
    fetch="join|select|subselect" (10)
    batch-size="N" (11)
    access="field|property|ClassName" (12)
    optimistic-lock="true|false" (13)
    mutable="true|false" (14)
>
    <key .... /> <map-key .... />
    <element .... />
</map>
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合映射的配置-3

- (1) name 集合属性的名称
- (2) table (可选——默认为属性的名称) 这个集合表的名称(不能在一对多的关联关系中使用)
- (3) schema (可选) 表的schema的名称，他将覆盖在根元素中定义的schema
- (4) lazy (可选——默认为true) 可以用来关闭延迟加载(false)，
- (5) inverse (可选——默认为false) 标记这个集合作为双向关联关系中的方向一端。
- (6) cascade (可选——默认为none) 让操作级联到子实体
- (7) sort (可选) 指定集合的排序顺序
- (8) order-by (可选，仅用于jdk1.4) 指定表的字段(一个或几个)再加上asc或者desc(可选)，定义Map, Set和Bag的迭代顺序
- (9) where (可选) 指定任意的SQL where条件，该条件将在重新载入或者删除这个集合时使用(当集合中的数据仅仅是所有可用数据的一个子集时这个条件非常有用)
- (10) fetch (可选，默认为select) 用于在外连接抓取、通过后续select抓取和通过后续subselect抓取之间选择。
- (11) batch-size (可选，默认为1) 指定通过延迟加载取得集合实例的批处理块大小
- (12) access (可选-默认为属性property):Hibernate取得集合属性值时使用的策略
- (13) 乐观锁 (可选 - 默认为 true): 对集合的状态的改变会是否导致其所属的实体的版本增长。(对一对多关联来说，关闭这个属性常常是有理的)
- (14) mutable (可变) (可选 - 默认为true): 若值为false, 表明集合中的元素不会改变 (在某些情况下可以进行一些小的性能优化)。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合映射的配置-4

#### n 集合外键

集合实例在数据库中依靠持有集合的实体的外键加以辨别。此外键作为集合关键字段加以引用。集合关键字段通过 <key> 元素映射。

在外键字段上可能具有非空约束。对于大多数集合来说，这是隐含的。对单向一对多关联来说，外键字段默认是可以为空的，因此你可能需要指明 not-null="true"。示例如下：

```
<key column="productSerialNumber" not-null="true"/>
```

外键约束可以使用 ON DELETE CASCADE, 示例如下：

```
<key column="productSerialNumber" on-delete="cascade"/>
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射-1

#### n one-to-one

通过 one-to-one 元素,可以定义持久化类的一对一关联。

<one-to-one

name="propertyName" (1)

class="ClassName" (2)

cascade="cascade\_style" (3)

constrained="true|false" (4)

fetch="join|select" (5)

property-ref="propertyNameFromAssociatedClass" (6)

access="field|property|ClassName" (7)

formula="any SQL expression" (8)

lazy="proxy|no-proxy|false" (9)

entity-name="EntityName" (10)

/>

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射-2

- (1) name: 属性名。
- (2) class (可选 — 默认是通过反射得到的属性类型): 被关联的类的名字。
- (3) cascade (级联) (可选) 表明操作是否从父对象级联到被关联的对象。
- (4) constrained (约束) (可选) 表明该类对应的表对应的数据库表，和被关联的对象所对应的数据库表之间，通过一个外键引用对主键进行约束。这个选项影响 save() 和 delete() 在级联执行时的先后顺序以及决定该关联能否被委托 (也在 schema export tool 中被使用)。
- (5) fetch (可选 — 默认为 select): 在外连接抓取 (outer-join fetching) 和序列选择抓取 (sequential select fetching) 两者中选择其一。
- (6) property-ref: (可选) 指定关联类的属性名，这个属性将会和本类的主键相对应。如果没有指定，会使用对方关联类的主键。
- (7) access (可选 — 默认为 property): Hibernate 用来访问属性值的策略。
- (8) formula (可选): 绝大多数一对一的关联都指向其实体的主键。在一些少见的情况中，你可能会指向其他的一个或多个字段，或者是一个表达式，这些情况下，你可以用一个 SQL 公式来表示。(可以在 org.hibernate.test.onetooneformula 找到例子)
- (9) lazy (可选 — 默认为 proxy): 默认情况下，单点关联是经过代理的。lazy="no-proxy" 指定此属性应该在实例变量第一次被访问时应该延迟抓取 (fetch lazily) (需要运行时字节码的增强)。lazy="false" 指定此关联总是被预先抓取。注意，如果 constrained="false"，不可能使用代理，Hibernate 会采取预先抓取。
- (10) entity-name (可选): 被关联的类的实体名。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射-3

#### n many-to-one

通过 many-to-one 元素,可以定义一种常见的与另一个持久化类的多对一关联,这个表的一个外键引用目标表的主键字段。

```
<many-to-one
  name="propertyName"           (1)
  column="column_name"          (2)
  class="ClassName"              (3)
  cascade="cascade_style" (4)
  fetch="join|select"           (5)
  update="true|false"           (6)
  insert="true|false"           (7)
  property-ref="propertyNameFromAssociatedClass" (8)
  access="field|property|ClassName" (9)
  unique="true|false"           (10)
  not-null="true|false"          (11)
  optimistic-lock="true|false"  (12)
  lazy="proxy|no-proxy|false"   (13)
  not-found="ignore|exception"  (14)
  entity-name="EntityName"      (15)
  formula="arbitrary SQL expression" (16)
/>
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射-4

- (1)name: 属性名。
- (2)column (可选): 外键字段的名称。也可以通过嵌套的 <column> 指定。
- (3)class (可选 — 默认是通过反射得到的属性类型): 被关联的类的名字。
- (4)cascade (级联) (可选) 表明操作是否从父对象级联到被关联的对象。
- (5)fetch (可选 — 默认为 select): 在外连接抓取和序列选择抓取两者中选择其一。
- (6)update, insert (可选 — 默认为 true) 指定对应的字段是否包含在用于 UPDATE 和/或 INSERT 的 SQL 语句中。如果二者都是false, 则这是一个纯粹的 “外源性 (derived)” 关联, 它的值是通过映射到同一个 (或多个) 字段的某些其他属性得到 或者通过 trigger (触发器)、或其他程序生成。
- (7)property-ref: (可选) 被关联到此外键的类中的对应属性的名字。如果不指定, 使用被关联类的主键
- (8)access (可选 — 默认为 property): Hibernate 用来访问属性值的策略。
- (9)unique (可选): 使用 DDL 为外键字段生成一个唯一约束。此外, 这也可以用作 propertyref 的目标属性。这使关联同时具有一对一的效果。
- (10)not-null (可选): 使用 DDL 为外键字段生成一个非空约束。
- (11)optimistic-lock (可选 — 默认为 true): 指定这个属性在做更新时是否需要获得乐观锁定。换句话说, 它决定这个属性发生脏数据时版本 (version) 的值是否增长。
- (12)lazy (可选 — 默认为 proxy): 默认情况下, 单点关联是经过代理的。lazy="no-proxy" 指定此属性应该在实例变量第一次被访问时应该延迟抓取 (fetch lazily) (需要运行时字节码的增强)。lazy="false" 指定此关联总是被预先抓取。
- (13)not-found (可选 - 默认为exception): 指定如何处理引用缺失行的外键: ignore 会把缺失的行作为一个空关联处理。
- (14)entity-name (可选): 被关联的类的实体名。
- (15)formula (可选): SQL 表达式, 用于定义 computed (计算出的) 外键值。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射-5

#### n one-to-many

通过 one-to-many 元素,可以定义一种常见的与另一个持久化类的一对多关联。

```
<one-to-many  
  class="ClassName" (1)  
  not-found="ignore|exception" (2)  
  entity-name="EntityName" (3)  
>
```

(1)class (必需): 被关联类的名称。

(2)not-found (可选 - 默认为exception): 指明若缓存的标示值关联的行缺失, 该如何处理: ignore 会把缺失的行作为一个空关联处理。

(3)entity-name (可选): 被关联的类的实体名, 作为 class 的替代。

注意: <one-to-many> 元素不需要定义任何字段。也不需要指定表名。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射的示例-1

还是用示例来看，下面作一个双向一对多的关系示例：

- 1：新建一个Parent对象和Child对象，Parent对象里面有一个Child对象的集合，Child对象里面有一个对Parent对象的引用，如下：

```
public class Parent{
    private String id;
    private String name;
    private Set children = new HashSet();
    .....
}

public class Child {
    private String uuid;
    private String address;
    private String postCode;
    private Parent parent;
    .....
}
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射的示例-2

Parent.hbm.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="cn.javass.h3.parentchild.Parent" table="tbl_parent" dynamic-
update="true" dynamic-insert="true" lazy="true">
        <id name="id" column="uuid" type="java.lang.String" unsaved-value="any">
            <generator class="assigned"> </generator>
        </id>
        <property name="name" update="true" insert="true"/>
        <set name="children" inverse="true" cascade="all">
            <key column="tbl_parent_fk"/>
            <one-to-many class="cn.javass.h3.parentchild.Child"/>
        </set>
    </class>
</hibernate-mapping>
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射的示例-3

Child.hbm.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="cn.javass.h3.parentchild.Child" table="tbl_child" dynamic-update="true"
        dynamic-insert="true" lazy="true">
        <id name="uuid" column="uuid" type="java.lang.String" unsaved-value="any">
            <generator class="assigned"> </generator>
        </id>
        <property name="address" update="true" insert="true"/>
        <property name="postCode" update="true" insert="true"/>
        <many-to-one
            name="parent"
            column="tbl_Parent_fk"
            class="cn.javass.h3.parentchild.Parent"
            not-null="true"
        />
    </class>
</hibernate-mapping>
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射的示例-4

客户端测试TestMR.java文件太长，直接看演示好了。

在这个测试里面分别演示了单独的操作和级联的操作。

- n inverse:** 指定由哪一方来维护双方的关联关系，默认是false，也就是双方都维护，主要用在一对多 和 多对多中。
- n** 在一对多中使用inverse的时候，通常是在”一”这一端设置inverse=true，他的意思就是由多的那一端去维护关系，非反向端用于把内存中的表示保存到数据库中。如下：

```
Parent p = new Parent();  
Child c = new Child();  
c.setParent(p); //维护父子之间关系  
p.getChildren().add(c);
```

- n inverse**还有一个重要特点就是会优化Sql

**n cascade:** 定义对象的级联关系

all : 所有情况下均进行关联操作

none: 所有情况下均不进行关联操作。这是默认值

save-update: 在执行save/update/saveOrUpdate时进行关联操作

delete: 在执行delete时进行关联操作

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 关系映射的示例-5

n 简单的示范一下双向的一对一

双向一对一的操作是类似于双向一对多的，只是配置上有一些不同：

1: Parent里面的配置，示例如下：

```
<one-to-one name="cc" class="cn.javass.h3.parentchild.Child"
cascade="all" property-ref="parent"/>
```

这里的property-ref参照的是Child对象里面的属性。

2: Child里面的配置，不是使用one-to-one，而是仍使用many-to-one，示例如下：

```
<many-to-one name="parent" class="cn.javass.h3.parentchild.Parent"
column="tbl_parent_fk"/>
```

3: 测试程序里面，原来调用集合的地方，变成调用一个Child对象，其他就差不多了，可以测试看看。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 过滤器-1

#### n Hibernate4的过滤器

Hibernate3 新增了对某个类或者集合使用预先定义的过滤器条件（filter criteria）的功能。过滤器条件相当于定义一个 非常类似于类和各种集合上的“where”属性的约束子句，但是过滤器条件可以带参数，应用程序可以在运行时决定是否启用给定的过滤器，以及使用什么样的参数值。过滤器的用法很像数据库视图，只不过是在应用程序中确定使用什么样的参数的。

#### n 定义过滤器

要使用过滤器，首先要在相应的映射节点中定义。而定义一个过滤器，要用到位于 <hibernatemapping/>节点之内的 <filter-def/> 节点：

示例如下：

```
<filter-def name="myFilter">
  <filter-param name="myFilterParam" type="string"/>
</filter-def>
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 过滤器-2

#### n 使用过滤器之配置

定义好之后，就可以在某个类中使用这个过滤器：

```
<class name="myClass" ...>
```

```
...
```

```
<filter name="myFilter" condition=":myFilterParam=FILTERED_COLUMN"/>
```

```
</class>
```

或者也可以在某个集合使用它：

```
<set ...>
```

```
<filter name="myFilter" condition=":myFilterParam=FILTERED_COLUMN"/>
```

```
</set>
```

在同时可以使用多个过滤器。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 过滤器-3

#### n 使用过滤器之程序

在程序中，需要使用session接口中的：`enableFilter(String filterName)`，`getEnabledFilter(String filterName)`，和 `disableFilter(String filterName)` 方法。Session中默认不启用过滤器，必须通过`enableFilter()` 方法显式的启用。示例代码`session.enableFilter("myFilter").setParameter("myFilterParam", "some-value");`

#### n 过滤器示例

在Parent.hbm.xml中定义有如下的过滤器：

```
<filter-def name="myFilter">
    <filter-param name="myFilterParam" type="string"/>
</filter-def>
<filter-def name="myFilter2">
    <filter-param name="myFilterParam" type="int"/>
    <filter-param name="myFilterParam2" type="int"/>
</filter-def>
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 过滤器-4

**n** 在定义Child集合中使用

```
<set name="children" inverse="true" cascade="all" fetch="select"
  lazy="false" batch-size="3">
  <key column="tbl_parent_fk"/>
  <one-to-many class="cn.javass.h3.parentchild.Child" />
  <filter name="myFilter"
    condition="address like :myFilterParam"></filter>
  <filter name="myFilter2" condition="uuid >= :myFilterParam and
    uuid <= :myFilterParam2"></filter>
</set>
```

**n** 程序中使用示例

```
s.enableFilter("myFilter").setParameter("myFilterParam", "%1%");
s.enableFilter("myFilter2").setParameter("myFilterParam", 1)
                           .setParameter("myFilterParam2", 3);
Query q = s.createQuery("select p from Parent as p ");
System.out.println("p==="+p.getChildren());
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程小结

- n 理解关系的表示
  - n 熟悉集合映射，掌握set
  - n 了解关系映射的知识
  - n 掌握双向一对多映射的开发
  - n 掌握过滤器的定义、配置和开发应用
- 
- n 作业：复习和练习这些知识，尤其是双向一对多映射的开发，还有过滤器的使用。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程概览

n Hibernate的事务和并发

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



## 《深入浅出学Hibernate4开发》——系列精品教程

### 第六部分：Hibernate的事务和并发

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 概述

- n Hibernate本身没有事务的实现  
Hibernate 直接使用 JDBC 连接和 JTA 资源，不添加任何附加锁定行为。也就是说你在Hibernate里面使用的事务要么是JDBC的事务，要么是JTA的事务。
- n Hibernate不锁定内存中的对象  
你的应用程序会按照你的数据库事务的隔离级别规定的那样运作，真正对事务的实现和支持也依赖于数据库。
- n 对于并发处理，Hibernate提供了乐观锁和悲观锁来进行并发处理  
Hibernate对自动乐观并发控制提供版本管理，针对行级悲观锁定，Hibernate也提供了辅助的（较小的）API，它使用了 SELECT FOR UPDATE 的 SQL 语法

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### Session和事务范围-1

- n Hibernate的Session是和事务联系在一起的  
可以通过Session去获取事务的接口，从而进行事务的控制。
- n 数据库事务应该尽可能的短  
这样能降低数据库中的锁争用。数据库长事务会阻止你的应用程序扩展到高的并发负载。因此，假若用户在思考期间让数据库事务开着，直到整个工作单元完成才关闭这个事务，这绝不是一个好的设计。  
这就引出一个问题：一个操作单元，也就是一个事务单元的范围应该是多大？  
一个操作一个？一个请求一个？一个应用一个？
- n 反模式：session-per-operation  
在单个线程中，不要因为一次简单的数据库调用，就打开和关闭一次Session！数据库事务也是如此。也就是说应该禁止自动事务提交（auto-commit）。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Session和事务范围-2

#### n session-per-request

最常用的模式是 每个请求一个会话。在这种模式下，来自客户端的请求被发送到服务器端，即 Hibernate 持久化层运行的地方，一个新的 Hibernate Session 被打开，并且执行这个操作单元中所有的数据库操作。一旦操作完成（同时对客户端的响应也准备就绪），session 被同步，然后关闭。会话和请求之间的关系是一对一的关系。

Hibernate 内置了对“当前 session (current session)”的管理，用于简化此模式。你要做的一切就是在服务器端要处理请求的时候，开启事务，在响应发送给客户之前结束事务，通常使用Servlet Filter来完成。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

# 数据库事务声明

### n 非托管环境下

所谓非托管，指的是：应用程序没有托管到J2EE环境中，通常由Hibernate自己来负责管理数据库连接池。应用程序开发人员必须手工设置事务声明，换句话说，就是手工启动，提交，或者回滚数据库事务。

### n 使用JTA

又有两种方式，一种是在Hibernate配置里面修改transaction的factory类，从而在程序里面可以直接使用Hibernate的事务API，也就是程序不用变化。

另外一种方式就是直接通过JNDI去查找UserTransaction，然后直接在程序里面使用JTA的接口来控制事务。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 乐观并发控制

#### n 应用程序级别的版本检查

简单点说，就是由应用程序自己实现版本检查来确保对话事务的隔离，从数据访问的角度来说是最低效的，不推荐使用。

#### n 扩展周期的session和自动版本化

Hibernate 使用扩展周期的 Session 的方式，或者脱管对象实例的方式来提供自动版本检查。单个 Session 实例和它所关联的所有持久化对象实例都被用于整个对话，这被称为 sessionper-conversation。

Hibernate 在同步的时候进行对象实例的版本检查，如果检测到并发修改则抛出异常。由开发人员来决定是否需要捕获和处理这个异常（通常的抉择是给用户 提供一个合并更改，或者在无脏数据情况下重新进行业务对话的机会）。

在等待用户交互的时候，Session 断开底层的 JDBC 连接。这种方式以数据库访问的角度来说是最高效的方式。应用程序不需要关心版本检查或脱管对象实例的重新关联，在每个数据库事务中，应用程序也不需要载入读取对象实例。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 悲观锁定-1

#### n 通常不需要自己去管理锁定策略

Hibernate 总是使用数据库的锁定机制，从不在内存中锁定对象。因而用户并不需要花很多精力去担心锁定策略的问题。通常情况下，只要为 JDBC 连接指定一下隔离级别，然后让数据库去搞定一切就够了。

然而，高级用户有时候希望进行一个排它的悲观锁定，或者在一个新的事务启动的时候，重新进行锁定。

#### n 类 LockMode 定义了Hibernate 所需的不同的锁定级别

- 1: 当更新或者插入一行记录的时候，锁定级别自动设置为LockMode.WRITE。
- 2: 当用户显式的使用数据库支持的 SQL 格式 SELECT ... FOR UPDATE 发送 SQL 的时候，锁定级别设置为 LockMode.UPGRADE。
- 3: 当用户显式的使用 Oracle 数据库的 SQL 语句 SELECT ... FOR UPDATE NOWAIT 的时候，锁定级别设置 LockMode.UPGRADE\_NOWAIT。
- 4: 当 Hibernate 在“可重复读”或者是“序列化”数据库隔离级别下读取数据的时候，锁定模式自动设置为 LockMode.READ。这种模式也可以通过用户显式指定进行设置。
- 5: LockMode.NONE 代表无需锁定。在 Transaction 结束时，所有的对象都切换到该模式上来。与session 相关联的对象通过调用 update() 或者 saveOrUpdate() 脱离该模式。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 悲观锁定-2

#### n 显示的指定锁定模式

- 1: 调用 Session.load() 的时候指定锁定模式 (LockMode)。
- 2: 调用 Session.lock()。
- 3: 调用 Query.setLockMode()。

#### n 显示指定锁定模式的说明

- 1: 如果在 UPGRADE 或者 UPGRADE\_NOWAIT 锁定模式下调用 Session.load(), 并且要读取的对象尚未被session 载入过, 那么对象通过 SELECT ... FOR UPDATE 这样的 SQL 语句被载入。如果为一个对象调用 load() 方法时, 该对象已经在另一个较少限制的锁定模式下被载入了, 那么 Hibernate 就对该对象调用 lock() 方法。
- 2: 如果指定的锁定模式是 READ, UPGRADE 或 UPGRADE\_NOWAIT, 那么 Session.lock() 就执行版本号检查。(在 UPGRADE 或者 UPGRADE\_NOWAIT 锁定模式下, 执行 SELECT ... FOR UPDATE这样的SQL语句。)
- 3: 如果数据库不支持用户设置的锁定模式, Hibernate 将使用适当的替代模式 (而不是抛出异常)。这一点可以确保应用程序的可移植性。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程小结

- n 理解和掌握J2EE事务相关的理论，尤其是事务传播属性，还有事务隔离级
  - n 掌握Hibernate的乐观并发控制
  - n 熟悉Session和事务范围，以及基本的处理方式
  - n 熟悉数据库事务声明的方式
  - n 了解悲观锁定的理论知识
- 
- n 作业：复习和掌握这些理论知识。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程概览

- n Hibernate的性能提升  
包括：抓取策略、实例化集合、集合性能
- n Hibernate的二级缓存  
对象的二级缓存、查询缓存、管理缓存、以及使用EHCache来实现二级缓存的示例

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



## 《深入浅出学Hibernate4开发》——系列精品教程

### 第七部分：性能提升和二级缓存

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 抓取策略-1

- n** 抓取策略 (fetching strategy) 是指：当应用程序需要在 (Hibernate 实体对象图的) 关联关系间进行导航的时候，Hibernate 如何获取关联对象的策略。抓取策略可以在 O/R 映射的元数据中声明，也可以在特定的 HQL 或条件查询 (Criteria Query) 中重载声明。Hibernate4 定义了如下几种抓取策略：
- 1: **连接抓取** (Join fetching) - Hibernate 通过 在 SELECT 语句使用 OUTER JOIN (外连接) 来 获得对象的关联实例或者关联集合。
  - 2: **查询抓取** (Select fetching) - 另外发送一条 SELECT 语句抓取当前对象的关联实体或集合。除非你显式的指定 lazy="false" 禁止 延迟抓取，否则只有当真正访问关联关系的时候，才会执行第二条 select 语句。
  - 3: **子查询抓取** (Subselect fetching) - 另外发送一条 SELECT 语句抓取在前面查询到 (或者抓取到) 的所有实体对象的关联集合。除非你显式的指定 lazy="false" 禁止 延迟抓取否则只有当你真正访问关联关系的时候，才会执行第二条 select 语句。
  - 4: **批量抓取** (Batch fetching) - 对查询抓取的优化方案， 通过指定一个主键或外键列表，Hibernate 使用单条 SELECT 语句获取一批对象实例或集合。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 抓取策略-2

**n** 映射文档中定义的抓取策略将会对以下列表条目产生影响：

- (1) 通过 `get()` 或 `load()` 方法取得数据。
- (2) 只有在关联之间进行导航时，才会隐式的取得数据。
- (3) 条件查询
- (4) 使用了 `subselect` 抓取的 HQL 查询

通常情况下，我们并不使用映射文档进行抓取策略的定制。更多的是，保持其默认值，然后在特定的事务中，使用 HQL 的左连接抓取（`left join fetch`）对其进行重载。这将通知 Hibernate 在第一次查询中使用外部关联（`outer join`），直接得到其关联数据。

**n** 条件查询 API 中，应该调用 `setFetchMode` 语句，例如：

```
User user = (User) session.createCriteria(User.class)
    .setFetchMode("permissions", FetchMode.JOIN)
    .add( Restrictions.idEq(userId) )
    .uniqueResult();
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 抓取策略-3

n Hibernate会区分下列各种情况：

- 1: **Immediate fetching, 立即抓取** - 当宿主被加载时，关联、集合或属性被立即抓取。
- 2: **Lazy collection fetching, 延迟集合抓取** - 直到应用程序对集合进行了一次操作时，集合才被抓取。（对集合而言这是默认行为。）
- 3: **"Extra-lazy" collection fetching, "Extra-lazy"集合抓取** - 对集合类中的每个元素而言，都是直到需要时才去访问数据库。除非绝对必要，Hibernate不会试图去把整个集合都抓取到内存里来（适用于非常大的集合）。
- 4: **Proxy fetching, 代理抓取** - 对返回单值的关联而言，当其某个方法被调用，而非对其关键字进行get操作时才抓取。
- 5: **"No-proxy" fetching, 非代理抓取** - 对返回单值的关联而言，当实例变量被访问的时候进行抓取。与上面的代理抓取相比，这种方法没有那么“延迟”得厉害（就算只访问标识符，也会导致关联抓取）但是更加透明，因为对应用程序来说，不再看到proxy。这种方法需要在编译期间进行字节码增强操作，因此很少需要用到。
- 6: **Lazy attribute fetching, 属性延迟加载** - 对属性或返回单值的关联而言，当其实例变量被访问的时候进行抓取。需要编译期字节码强化，因此这一方法很少用到

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 实例化集合和代理-1

#### n 何时实例化集合和代理

既然Hibernate会采取代理来延迟集合的实例化，那么何时实例化这个集合呢？因为如果session关闭了，再访问未初始化的集合活代理的话，将会抛出LazyInitializationException 异常。

常见的解决方法，如下：

#### n 在session打开期间去访问集合的数据

这是一种不好的做法，就是没有什么用的去访问集合的数据，触发其实例化

#### n 使用Hibernate.initialized()方法

静态方法 Hibernate.initialized() 为你的应用程序提供了一个便捷的途径来延迟加载集合或代理。只要它的 Session 处于 open 状态，Hibernate.initialize(cat) 将会为 cat 强制对代理实例化。同样，Hibernate.initialize(cat.getKittens())对kittens的集合具有同样的功能。

#### n 保持Session一直处于open状态，直到集合被实例化

通常又有两种方法：

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 实例化集合和代理-2

#### n Open Session in View模式

在一个基于Web的应用中，可以利用过滤器，在用户请求时打开Hibernate的 session、页面生成结束时关闭 Session，示例如下：

```
public class HibernateSessionRequestFilter implements Filter {
    private SessionFactory sf;
    public void doFilter(...)throws... {
        sf.getCurrentSession().beginTransaction();
        chain.doFilter(request, response);
        sf.getCurrentSession().getTransaction().commit();
    }
    public void init(FilterConfig filterConfig) throws ServletException {
        sf = new Configuration().configure().buildSessionFactory();
    }
    public void destroy() {}
}
```

n 在逻辑层为表现层准备数据的时候，在session打开的情况下，实例化好所有需要的数据。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合性能-1

- n 要完全理解Hibernate各种集合的关系结构和性能特点，必须同时考虑“用于Hibernate更新或删除集合行数据的主键的结构”。因此得到了如下的分类：有序集合类、集合（sets）、包（bags）
- n 所有的有序集合类（maps, lists, arrays）都拥有一个由<key>和 <index>组成的主键。这种情况下集合类的更新是非常高效的——主键已经被有效的索引，因此当Hibernate试图更新或删除一行时，可以迅速找到该行数据
- n 集合（sets）的主键由<key>和其他元素组成。对于有些元素来说，这很低效，特别是组合元素，大文本或二进制数据。但是对于一对多或者多对多，set也可以达到同样的高效性能
- n Bag是最差的。因为bag允许重复的元素值，也没有索引字段，因此不可能定义主键。Hibernate无法判断出重复的行。当这种集合被更改时，Hibernate将会先完整地移除（通过一个（in a single DELETE））整个集合，然后再重新创建整个集合。因此Bag是非常低效的。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 集合性能-2

**n** Lists, maps 和sets用于更新效率最高

- 1: 有序集合类型和大多数set都可以在增加、删除、修改元素中拥有最好的性能。
- 2: 在多对多中，set性能不如有序集合类型
- 3: Hibernate中，set应该是最通用的集合类型

**n** Bag和list是反向集合类（也就是指定了inverse=true的集合类）中效率最高的在一种情况下，bag的性能(包括list)要比set高得多：对于指明了inverse="true"的集合类（比如说，标准的双向的一对多关联），我们可以在未初始化(fetch)包元素的情况下直接向bag或list添加新元素！这是因为Collection.add()或者Collection.addAll() 方法 对bag或者List总是返回true（这点与Set不同）。因此对于下面的相同代码来说，速度会快得多。

```
Parent p = (Parent) sess.load(Parent.class, id);
Child c = new Child();
c.setParent(p);
p.getChildren().add(c); //不用抓取集合
sess.flush();
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 监测性能

- n SessionFactory里面带着监测性能的数据，就是Statistics  
    首先需要在cfg.xml中打开统计功能，将 `hibernate.generate_statistics` 设置为 `true`。在运行期间，则可以可以通过  
    `sf.getStatistics().setStatisticsEnabled(true)` 或  
    `hibernateStatsBean.setStatisticsEnabled(true)` 来打开统计功能
- n 所有的测量值都可以由 Statistics 接口 API 进行访问，主要分为三类：
  - 1: 使用 Session 的普通数据记录，例如打开的 Session 的个数、取得的 JDBC 的连接数等；
  - 2: 实体、集合、查询、缓存等内容的统一数据记录。
  - 3: 和具体实体、集合、查询、缓存相关的详细数据记录

**真正高质量培训    签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-1

- n Hibernate的Session在事务级别进行持久化数据的缓存操作，也就是前边所讲的一级缓存
- n 当然，也可以分别为每个类（或集合），配置集群、或JVM级别(SessionFactory级别)的缓存（二级缓存）
- n 在Hibernate4以前的版本：通过在hibernate.cache.provider\_class属性中指定org.hibernate.cache.CacheProvider的某个实现的类名，你可以选择让Hibernate使用哪个缓存实现。注意，在3.2版本之前，默认使用EhCache 作为缓存实现，但从3.2起就不再这样了。如：<property name="cache.provider\_class">org.hibernate.cache.EhCacheProvider</property>
- n 在Hibernate4的版本中：通过在cache.region.factory\_class指定相应的缓存区域的实现，如：<property name="cache.region.factory\_class">org.hibernate.cache.EhCacheRegionFactory</property>

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-2

缓存策略提供商 (Cache Providers)

Cache	Provider class	Type	Cluster Safe	Query Cache Supported
Hashtable (not intended for production use)	org.hibernate.cache.HashtableCacheProvider	memory		yes
EHCache	org.hibernate.cache.EhCacheProvider	memory, disk		yes
OSCache	org.hibernate.cache.OSCacheProvider	memory, disk		yes
SwarmCache	org.hibernate.cache.SwarmCacheProvider	clustered (ip multicast)	yes (clustered invalidation)	
JBoss Cache 1.x	org.hibernate.cache.TreeCacheProvider	clustered (ip multicast), transactional	yes (replication)	yes (clock sync req.)
JBoss Cache 2	org.hibernate.cache.jbc2.JBossCacheRegionFactory	clustered (ip multicast), transactional	yes (replication or invalidation)	yes (clock sync req.)

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-3

**n** 缓存映射，类或者集合映射的“<cache>元素”可以有如下形式：

<cache

usage="transactional|read-write|nonstrict-read-write|read-only" (1)

region="RegionName" (2)

include="all|non-lazy" (3) />

(1) usage (必须) 说明了缓存的策略：transactional、read-write、nonstrict-read-write 或 read-only。

(2) region (可选，默认为类或者集合的名字) 指定第二级缓存的区域名

(3) include (可选，默认为 all) non-lazy 当属性级延迟抓取打开时，标记为 lazy="true" 的实体的属性可能无法被缓存

**n** 可以在hibernate.cfg.xml 中指定<class-cache>和<collection-cache>元素

**n** 策略：只读缓存

如果你的应用程序只需读取一个持久化类的实例，而无需对其修改，那么就可以对其进行只读 缓存。这是最简单，也是实用性最好的方法。甚至在集群中，它也能完美地运作。示例如下：

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-4

```
<class name="eg.Immutable" mutable="false">  
    <cache usage="read-only"/>  
    ....  
</class>
```

#### **n** 策略：读写缓存

如果应用程序需要更新数据，那么使用读/写缓存 比较合适。如果应用程序要求“serializable”的隔离级别，那么就决不能使用这种缓存策略。如果在JTA环境中使用缓存，必须指定hibernate.transaction.manager\_lookup\_class属性的值，通过它，Hibernate才能知道该应用程序中JTA的TransactionManager的具体策略。在其它环境中，你必须保证在Session.close()、或Session.disconnect()调用前，整个事务已经结束。如果你想在集群环境中使用此策略，你必须保证底层的缓存实现支持锁定(locking)。Hibernate内置的缓存策略并不支持锁定功能。

```
<class name="eg.Cat" .... >  
    <cache usage="read-write"/>  
    <set name="kittens" ... >  
        <cache usage="read-write"/>  
        ....  
    </set>  
</class>
```

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-5

**n** 策略：非严格读写缓存Strategy: nonstrict read/write)

如果应用程序只偶尔需要更新数据（也就是说，两个事务同时更新同一记录的情况很不常见），也不需要十分严格的事务隔离，那么比较适合使用非严格读/写缓存策略。如果在JTA环境中使用该策略，你必须为其指定hibernate.transaction.manager\_lookup\_class属性的值，在其它环境中，你必须保证在Session.close()、或Session.disconnect()调用前，整个事务已经结束

**n** 策略：事务缓存

Hibernate的事务缓存策略提供了全事务的缓存支持，例如对JBoss TreeCache的支持。这样的缓存只能用于JTA环境中，你必须指定为其hibernate.transaction.manager\_lookup\_class属性

各种缓存提供商对缓存并发策略的支持情况

Cache	read-only	nonstrict-read-write	read-write	transactional
Hashtable (not intended for production use)	yes	yes	yes	
EHCache	yes	yes	yes	
OSCache	yes	yes	yes	
SwarmCache	yes	yes		
JBoss Cache 1.x	yes			yes
JBoss Cache 2	yes			yes

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-6

#### n 管理缓存

- 1: 无论何时，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。
- 2: 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，如下示例：

```
ScrollableResult cats = sess.createQuery( "from Cat as cat" ).scroll(); //很大的结果集
while ( cats.next() ) {
    Cat cat = (Cat) cats.get(0);
    doSomethingWithACat(cat);
    sess.evict(cat);
}
```

- 3: Session还提供了contains()方法，用来判断某个实例是否处于当前session的缓存中。
- 4: 如若要把所有的对象从session缓存中彻底清除，则需要调用Session.clear()。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-7

5: 对于二级缓存来说，在SessionFactory中定义了许多方法，清除缓存中实例、整个类、集合实例或者整个集合。

```
sessionFactory.evict(Cat.class, catId); //evict a particular Cat  
sessionFactory.evict(Cat.class); //evict all Cats  
sessionFactory.evictCollection("Cat.kittens", catId); //evict a  
particular collection of kittens  
sessionFactory.evictCollection("Cat.kittens"); //evict all kitten
```

6: collections CacheMode参数用于控制具体的Session如何与二级缓存进行交互。

- (1) CacheMode.NORMAL - 从二级缓存中读、写数据。
- (2) CacheMode.GET - 从二级缓存读取数据，仅在数据更新时对二级缓存写数据。
- (3) CacheMode.PUT - 仅向二级缓存写数据，但不从二级缓存中读数据。
- (4) CacheMode.REFRESH - 仅向二级缓存写数据，但不从二级缓存中读数据。通过hibernate.cache.use\_minimal\_puts的设置，强制二级缓存从数据库中读取数据，刷新缓存内容。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存-8

- n 查询缓存：查询的结果集也可以被缓存。只有当经常使用同样的参数进行查询时，这才会有些用处
- 1： 要使用查询缓存，首先你必须打开hibernate.cache.use\_query\_cache为true
- 2： 该设置将会创建两个缓存区域 - 一个用于保存查询结果集(org.hibernate.cache.StandardQueryCache)； 另一个则用于保存最近查询的一系列表的时间戳(org.hibernate.cache.UpdateTimestampsCache)。 请注意：在查询缓存中，它并不缓存结果集中所包含的实体的确切状态；它只缓存这些实体的标识符属性的值、以及各值类型的结果。 所以查询缓存通常会和二级缓存一起使用。
- 3： 绝大多数的查询并不能从查询缓存中受益，所以Hibernate默认是不进行查询缓存的。如若需要进行缓存，请调用 Query.setCacheable(true)方法。这个调用会让查询在执行过程中时先从缓存中查找结果，并将自己的结果集放到缓存中去

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存示例-1

**n** 以EHCACHE为例来说明二级缓存的配置

**n** 第一步：配置ehcache.xml，放置到classpath下面，配置如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<ehcache>
<diskStore path="java.io.tmpdir"/>
<defaultCache
    maxElementsInMemory="10000"
    eternal="false"
    timeToIdleSeconds="120"
    timeToLiveSeconds="120"
    overflowToDisk="true"
/>
</ehcache>
```

**n** 配置说明：

maxElementsInMemory ： 缓存最大数目

eternal ： 缓存是否持久

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存示例-2

overflowToDisk : 是否保存到磁盘，当系统当机时

timeToIdleSeconds : 当缓存闲置n秒后销毁

timeToLiveSeconds : 当缓存存活n秒后销毁

**n** 第二步：在hibernate.cfg.xml中设置：

**Hibernate4以前的版本：**

```
<property name="hibernate.cache.use_second_level_cache">true</property>
<property name="cache.provider_class">org.hibernate.cache.EhCacheProvider</property>
<property name="cache.use_query_cache">false</property>
```

**Hibernate4的版本：**

```
<property name="cache.use_query_cache">false</property>
<property name="cache.use_second_level_cache">true</property>
<property
    name="cache.region.factory_class">org.hibernate.cache.EhCacheRegionFactory</property>
```

**n** 第三步：在hibernate.cfg.xml中设置需要缓存的类，还有缓存策略

```
<class-cache class="cn.javass.h3.hello.UserModel" usage="read-only"/>
```

**n** 在测试文件中，如下：

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存示例-3

```
s = sf.openSession();
t = s.beginTransaction();
UserModel um1 = (UserModel)s.load(UserModel.class, "1");
System.out.println("um1==" + um1);
t.commit();
s = sf.openSession();
t = s.beginTransaction();
UserModel um2 = (UserModel)s.load(UserModel.class, "1");
System.out.println("um2==" + um2);
t.commit();
s = sf.openSession();
t = s.beginTransaction();
UserModel um3 = (UserModel)s.load(UserModel.class, "1");
System.out.println("um3==" + um3);
t.commit();
```

- n** 然后就可以运行测试文件，看看输出的sql语句，一共有多少条？一条查询语句才是正确的，因为缓存起了作用。尝试去掉缓存的配置，测试文件不动，再次运行，看看输出的sql语句，一共有多少条？应该是三条了。

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存示例-4

- n 说明：如果不设置“查询缓存”，那么hibernate只会缓存使用load()方法获得的单个持久化对象，如果想缓存使用findall()、list()、Iterator()、createCriteria()、createQuery()等方法获得的数据结果集的话，就需要设置hibernate.cache.use\_query\_cache true才行。
- n 下面看看查询缓存的示例
- n 第一步：配置ehcache.xml，跟前面相同
- n 第二步：配置hibernate.cfg.xml，在前面配置的基础上，添加：  
配置是否使用查询缓存为true  
<property name="hibernate.cache.use\_query\_cache">true</property>
- n 第三步：在写程序的时候，还要设置Query的setCacheable(true);
- n 测试文件如下，运行一下看看，然后再去掉缓存配置运行一下看看：

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 二级缓存示例-5

```
s = sf.openSession();
t = s.beginTransaction();
Query query1 =s.createQuery("select Object(o) from UserModel o");
query1.setCacheable(true);
List list1 = query1.list();
System.out.println("list1==" +list1);
t.commit();
s = sf.openSession();
t = s.beginTransaction();
Query query2 =s.createQuery("select Object(o) from UserModel o");
query2.setCacheable(true);
List list2 = query2.list();
System.out.println("list2==" +list2);
t.commit();
s = sf.openSession();
t = s.beginTransaction();
Query query3 =s.createQuery("select Object(o) from UserModel o");
query3.setCacheable(true);
List list3 = query3.list();
System.out.println("list3==" +list3);
t.commit();
```

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程小结

- n 理解Hibernate的各种抓取策略
  - n 掌握何时、如何实例化集合
  - n 理解集合性能
  - n 了解如何监测Hibernate的性能
  - n 掌握二级缓存的基本知识
  - n 掌握使用EHCACHE来实现二级缓存
- 
- n 作业：
    - 1: 复习、整理和吸收这些理论知识。
    - 2: 动手去试验一下Hibernate的各种抓取策略
    - 3: 动手去使用EHCACHE来实现二级缓存，并体会用与不用的区别

**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程概览

- n Hibernate的基本实现原理

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



## 《深入浅出学Hibernate4开发》——系列精品教程

### 第八部分：基本实现原理

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Hibernate整体运行流程

#### n 整体流程

- 1: 通过configuration来读cfg.xml文件
- 2: 得到SessionFactory 工厂
- 3: 通过SessionFactory 工厂来创建Session实例
- 4: 通过Session打开事务
- 5: 通过session的api操作数据库
- 6: 事务提交
- 7: 关闭连接

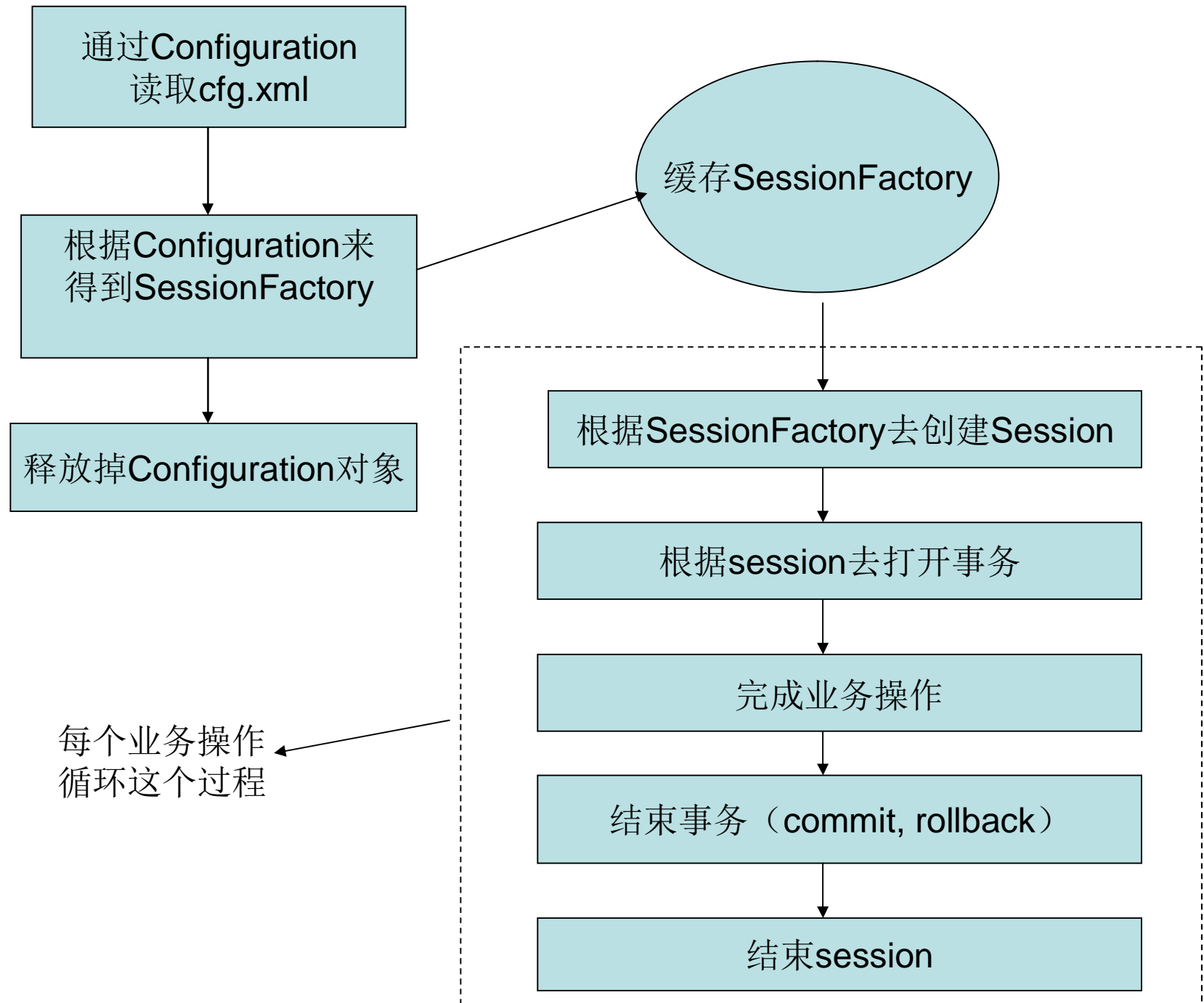
n 说明：以下分方法描述的实现流程并不是Hibernate的完整实现流程，也不是Hibernate的完整实现顺序，只是描述了Hibernate实现这些方法的主干和基本方式，主要是用来理解这些方法背后都发生了些什么，如果需要详细完整的实现流程，请查阅Hibernate相应文档和源代码

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### save方法基本流程

n 当我们调用了session.save(UserModel)后：

1: T0-->P0: Hibernate先在缓存中查找，如果发现在内部缓存中已经存在相同id的P0，就认为这个数据已经保存了，抛出例外。

如果缓存中没有，Hibernate会把传入的这个T0对象放到session控制的实例池去，也就是把一个瞬时对象变成了一个持久化对象。

如果需要Hibernate生成主键值，Hibernate就会去生成id并设置到P0上

2: 客户端提交事务或者刷新内存

3: 根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件

4: 根据hbm.xml文件和model来动态的拼sql，如下：

insert into 表名(来自hbm.xml) (字段名列表 (来自hbm.xml )) values(对应的值的列表(根据hbm.xml从传入的model中获取值))

5: 真正用JDBC执行sql，把值添加到数据库

6: 返回这个P0的id。

**真正高质量培训 签订就业协议**

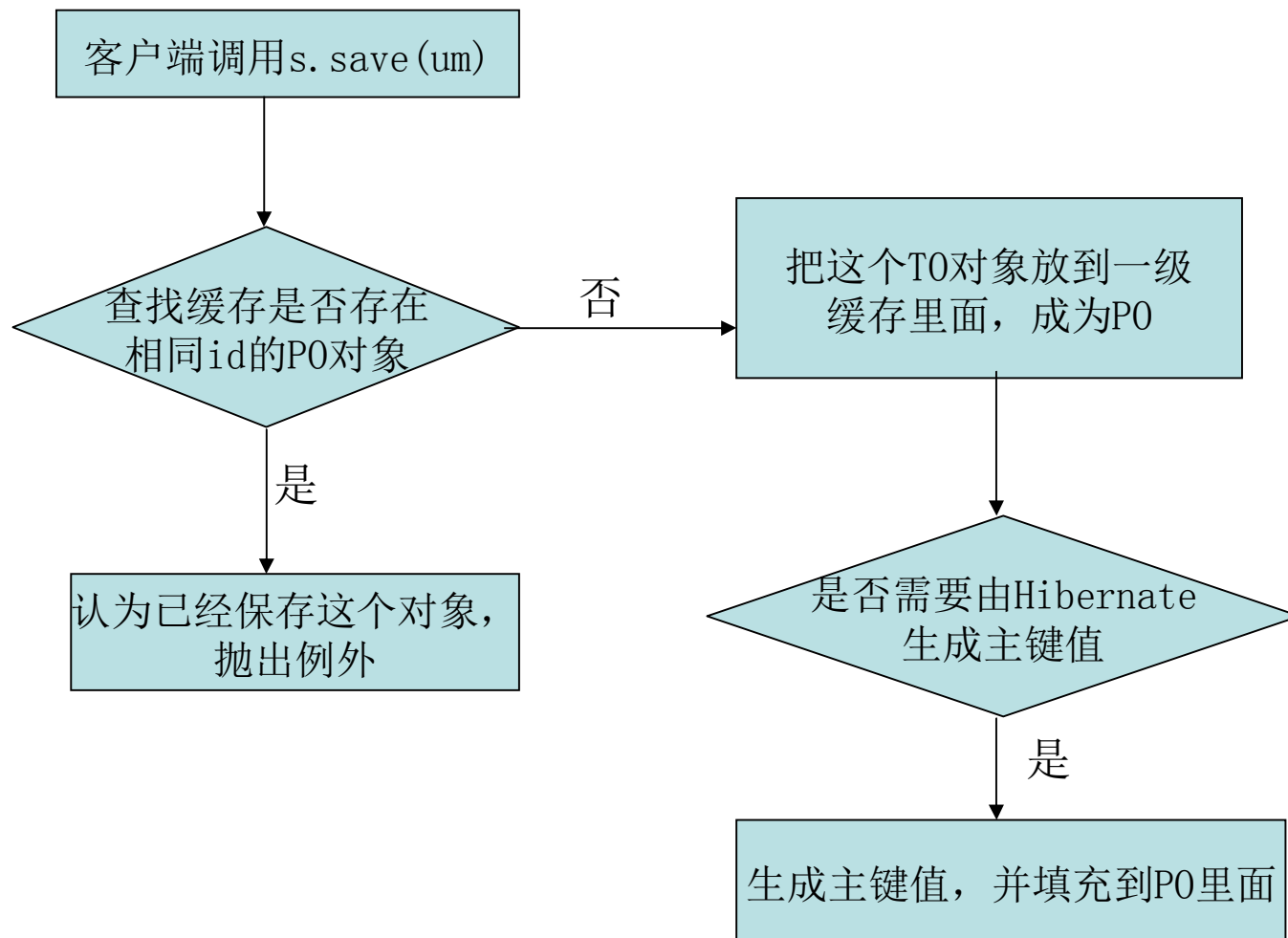
网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

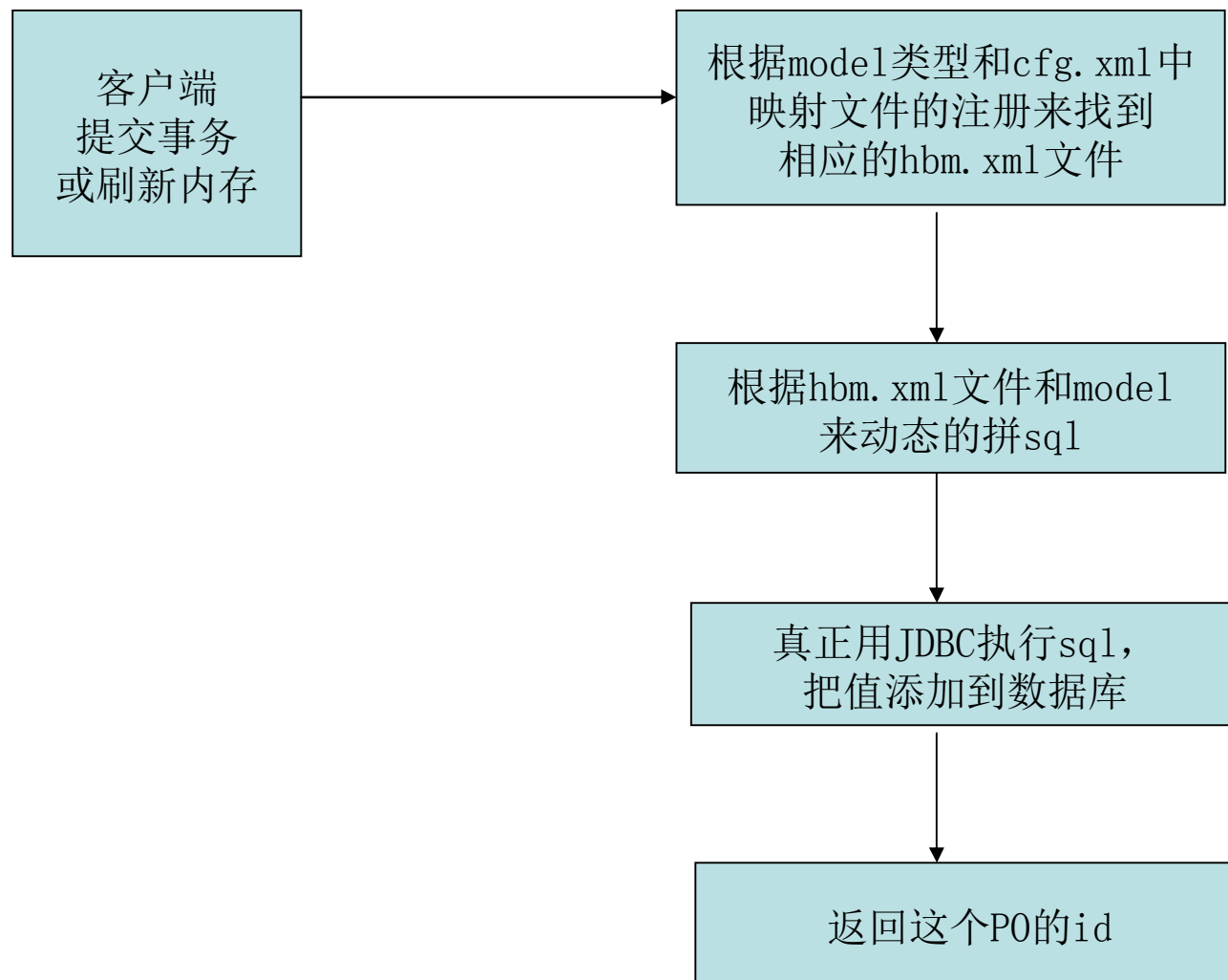
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Update (D0) 方法基本流程

n 当我们调用了session.update(UserModel)后：

1: D0--->P0: 首先根据model 的主键在hibernate的实例池中查找该对象，找到就抛出错误。

如果没有就D0--->P0, Hibernate会把传入的这个D0对象放到session控制的实例池去，也就是把一个瞬时对象变成了一个持久化对象

2: 客户端提交事务或者刷新内存

3: 根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件

4: 根据hbm.xml文件和model来动态的拼sql，不进行脏数据检查，如下：

update 表名(来自hbm.xml) set 字段名(来自hbm.xml) =值(根据hbm.xml从传入的model中获取值) where 条件

5: 真正用JDBC执行sql，把值修改到数据库

**真正高质量培训 签订就业协议**

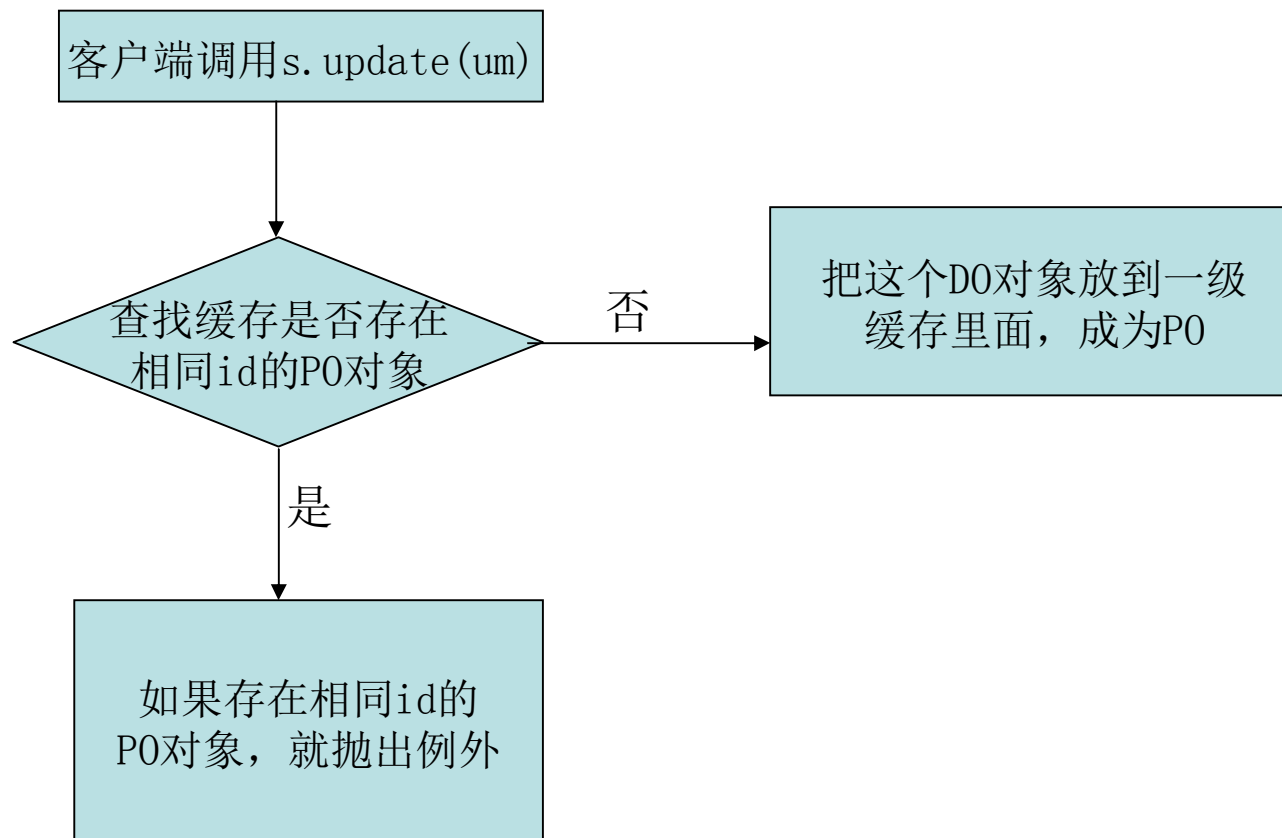
网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

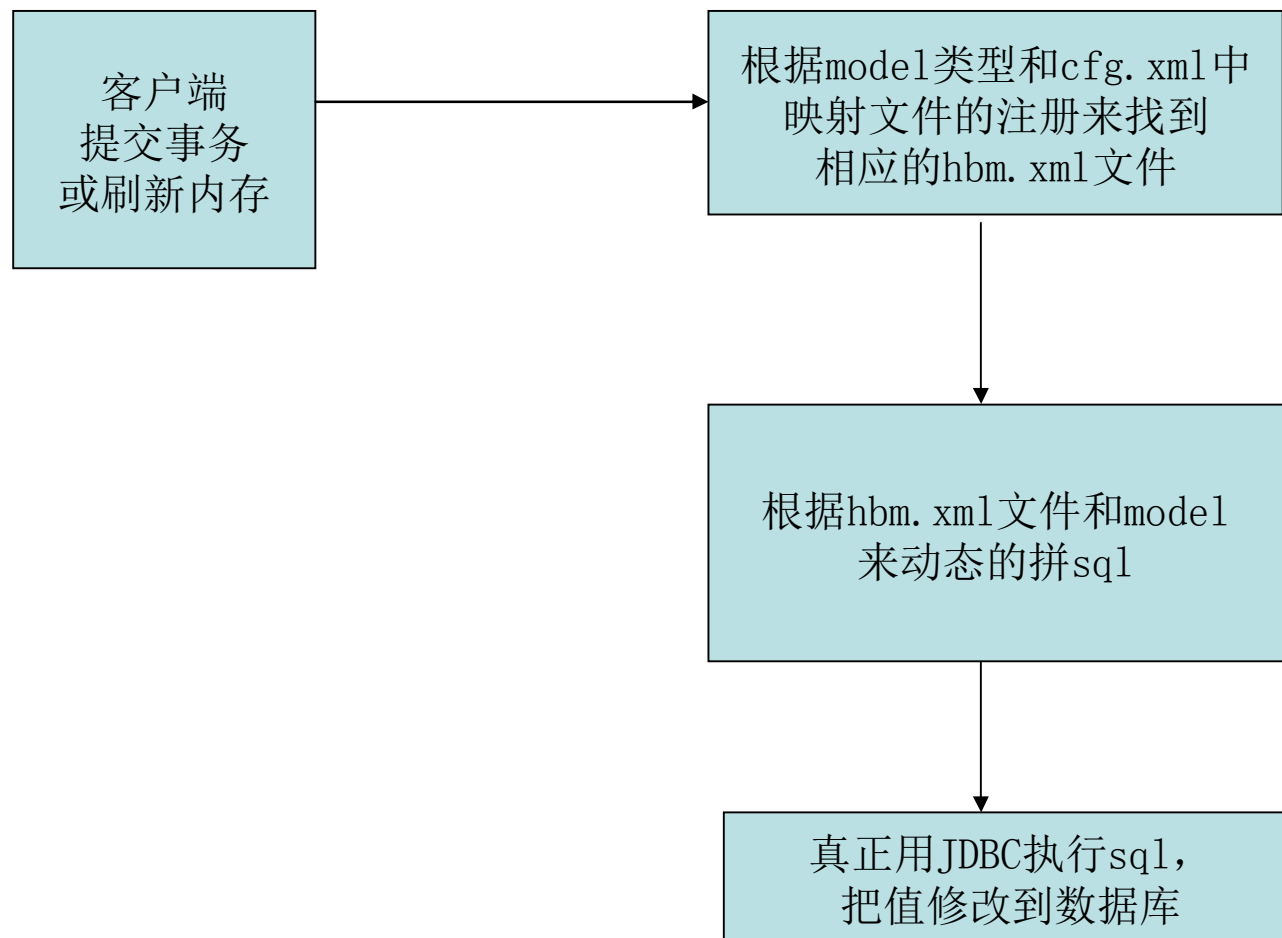
网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Update (P0) 方法基本流程

n 当我们调用了session.update(UserModel)后：

- 1: 首先根据model 的主键在hibernate的实例池中查找该对象，找到就使用该P0对象（用来检查脏数据）。
- 2: 客户端提交事务或者刷新内存
- 3: **Hibernate**会进行脏数据检查，如果没有数据被修改，就不执行下面的步骤了。
- 4: 根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
- 5: 根据hbm.xml文件和model来动态的拼sql，进行脏数据检查（如果开启了dynamic-update的话），如下：  
update 表名(来自hbm.xml) set 字段名（来自hbm.xml）=值(根据hbm.xml从传入的model中获取值) where 条件
- 6: 真正用JDBC执行sql，把值修改到数据库

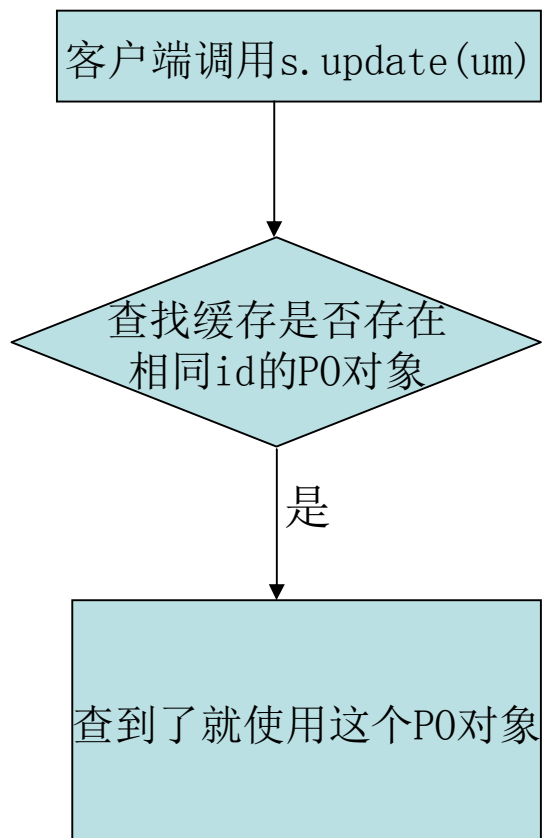
**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

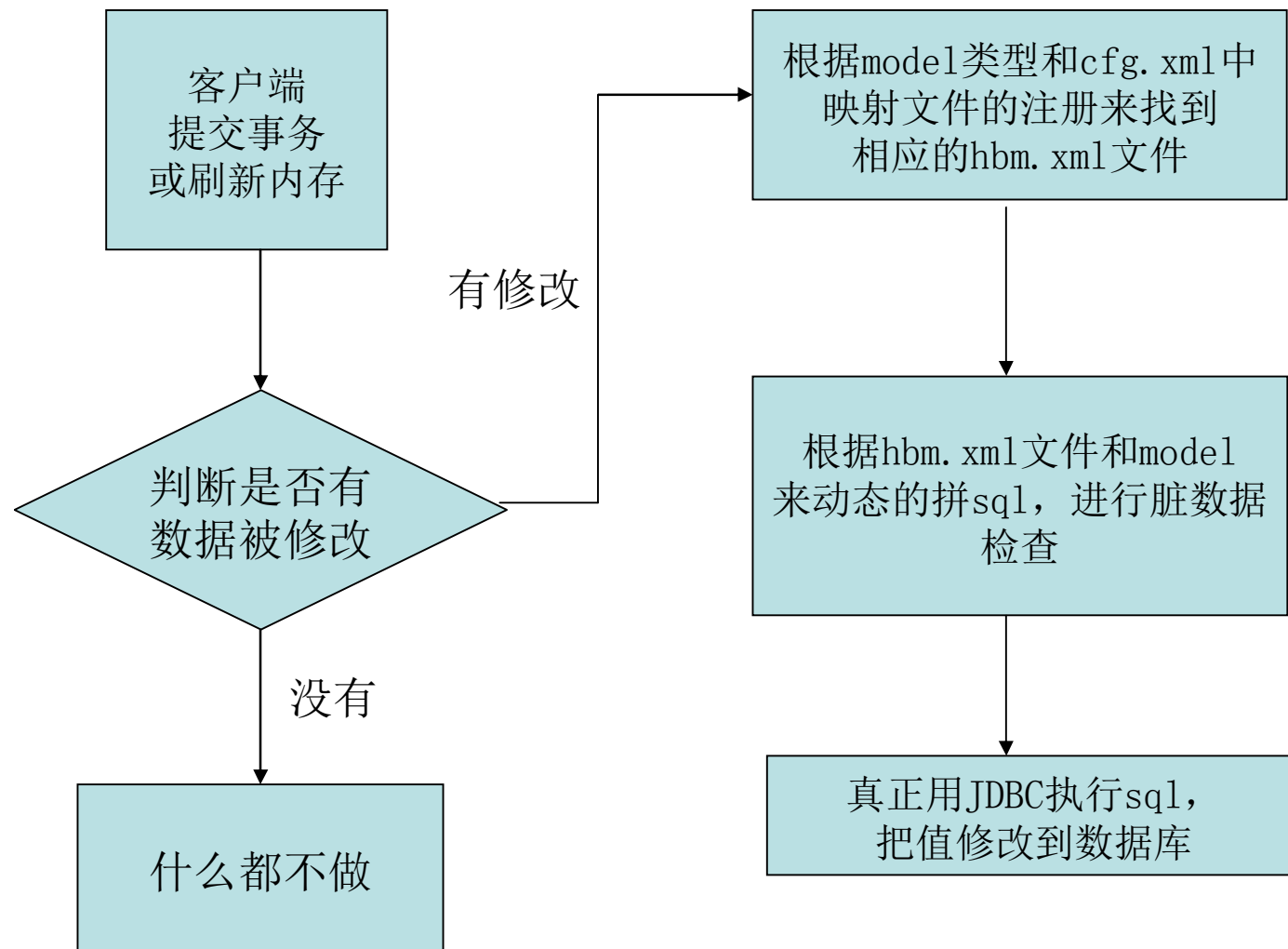
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Delete (D0) 方法基本流程

n Id的生成方式为assigned的情况

n 当我们调用了session.delete(UserModel)后：

- 1: 根据model的主键在数据库里面查找数据, 来保证对象的存在, 然后把找到的对象放到内存里面, 如果此时在hibernate的实例池中已经存在对应的实体对象(注意: 代理对象不算实体对象), 就抛出例外。
- 2: 如果此时在hibernate的实例池中不存在对应的实体对象, 那么就把对象放到内存里面, 但会标识成待删除的对象, 就不可以被load等使用了。
- 3: 如果对象还是不存在, 那么就直接返回了(注意, 这个时候是不抛出例外的)。也就是说, delete之前会执行一个查询语句。
- 4: 客户端提交事务或者刷新内存
- 5: 判断待删除的PO是否存在, 存在才需要删除, 否则不需要删除
- 6: 如果要删除, 才执行以下的步骤。先根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
- 7: 根据hbm.xml文件和model来动态的拼sql, 如下:  
delete from 表名(来自hbm.xml) where 主键=值(来自model)
- 8: 真正用JDBC执行sql, 把数据从数据库中删除

**真正高质量培训 签订就业协议**

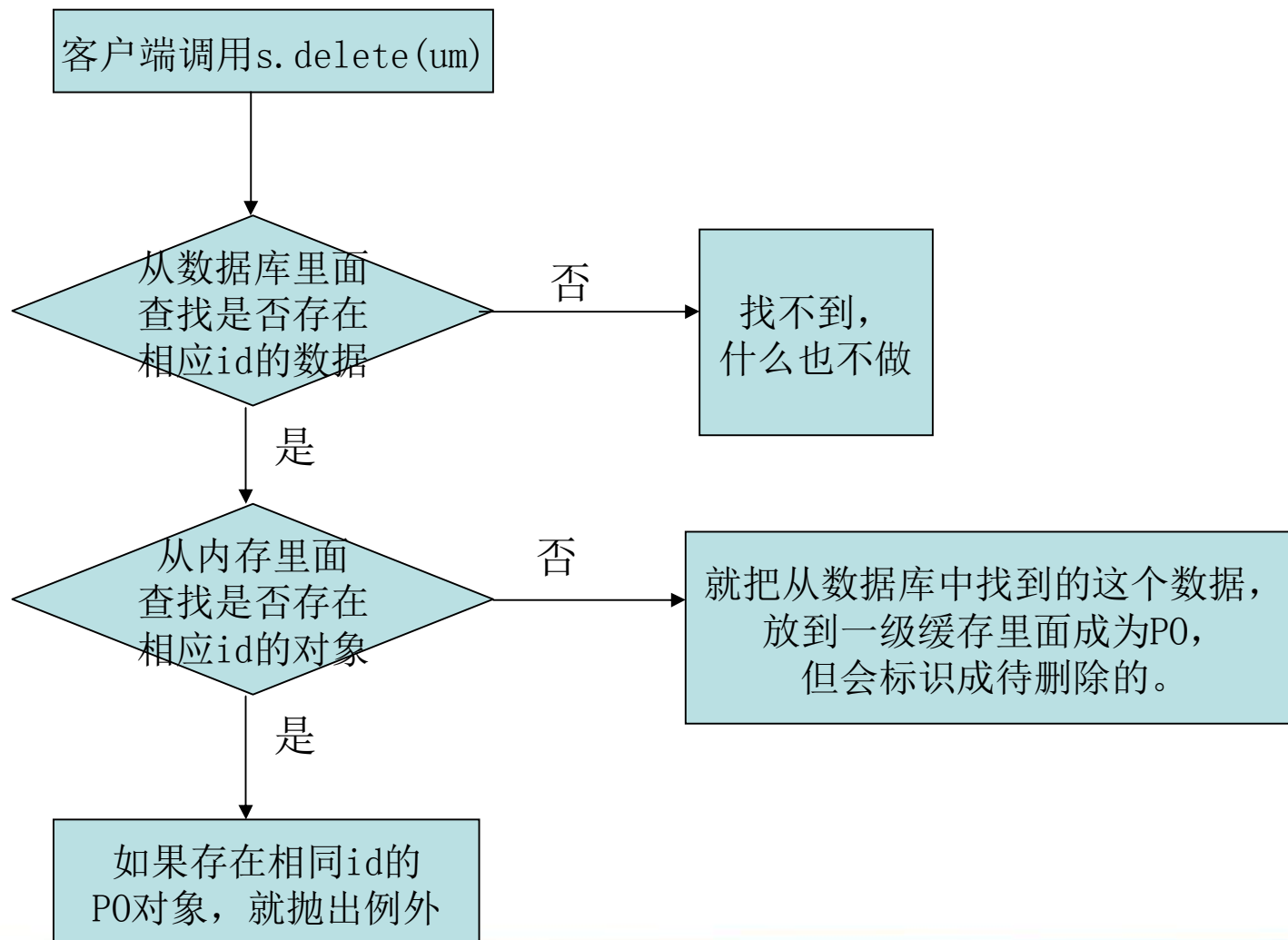
网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频, 更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

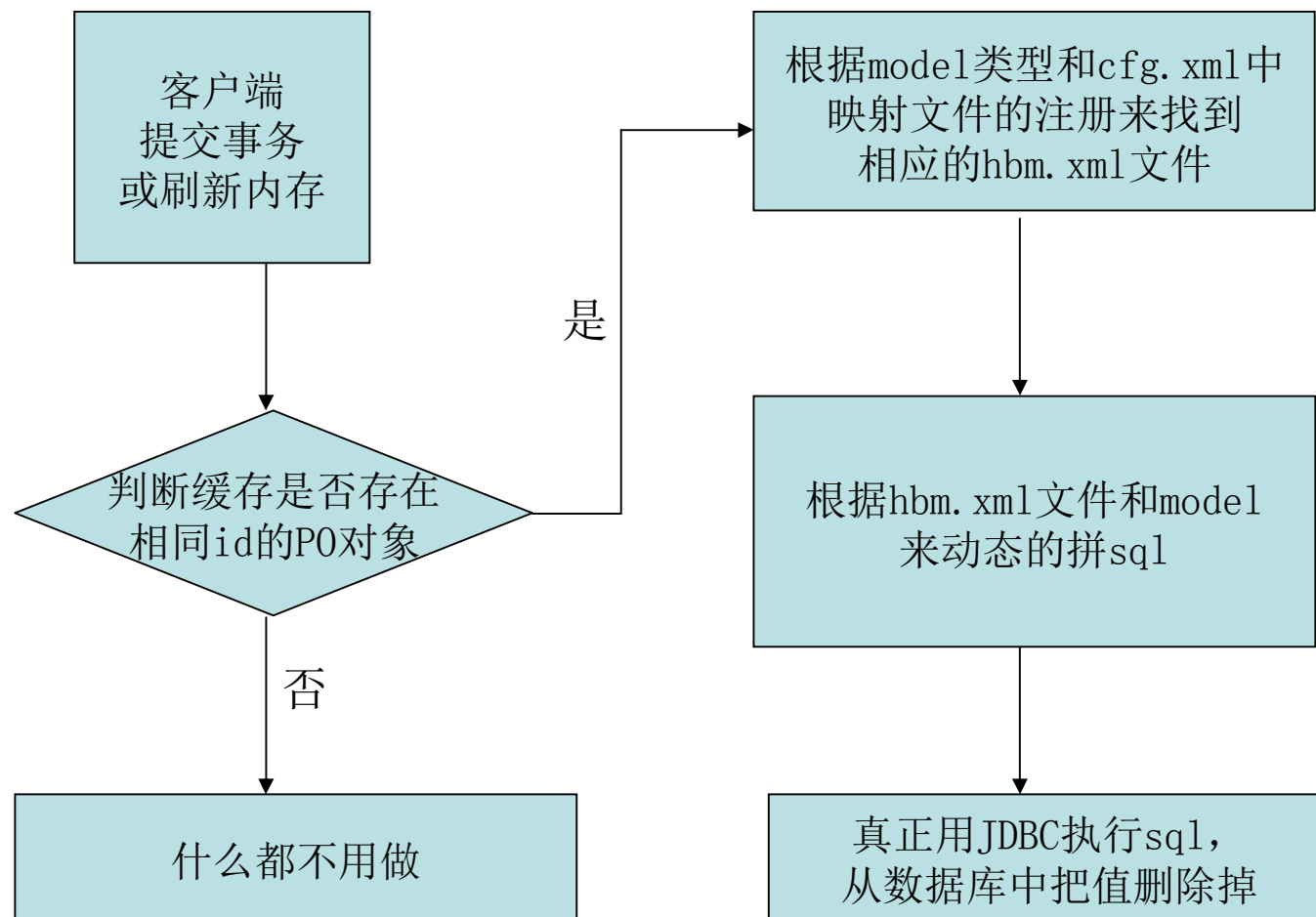
网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>

咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Delete (D0) 方法基本流程

- n Id的生成方式为非assigned的情况
- n 当我们调用了session.delete(UserModel)后：
  - 1: 根据model的主键在hibernate的实例池中查找对应的实体对象(注意：代理对象不算实体对象)，找到就抛出例外。
  - 2: 如果内存中没有对应的实体对象，就什么都不做。
  - 3: 客户端提交事务或者刷新内存
  - 4: 先根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
  - 5: 根据hbm.xml文件和model来动态的拼sql，如下：  
delete from 表名(来自hbm.xml) where 主键=值(来自model)
  - 6: 真正用JDBC执行sql，把数据从数据库中删除，如果数据不存在，就抛出例外

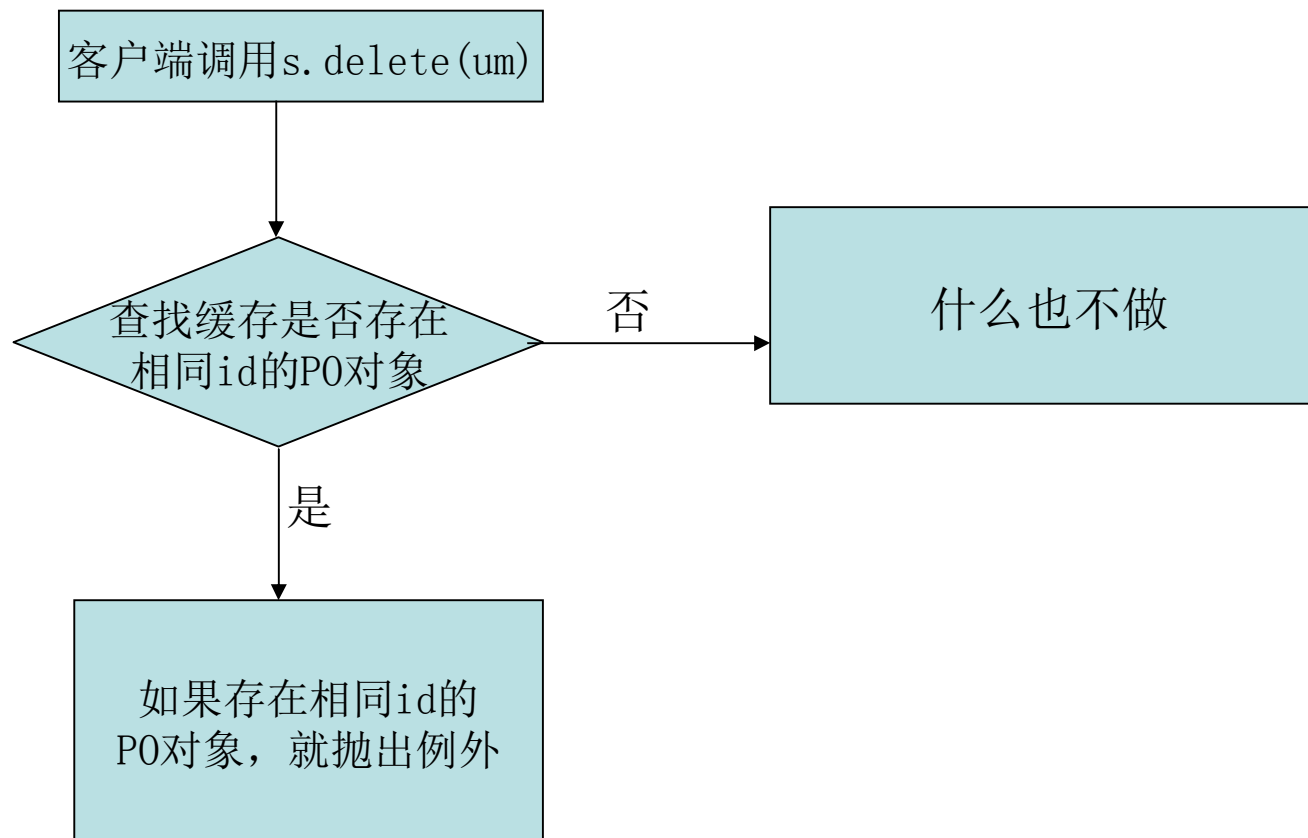
**真正高质量培训 签订就业协议**

网 址: <http://www.javass.cn>  
咨询QQ: 460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

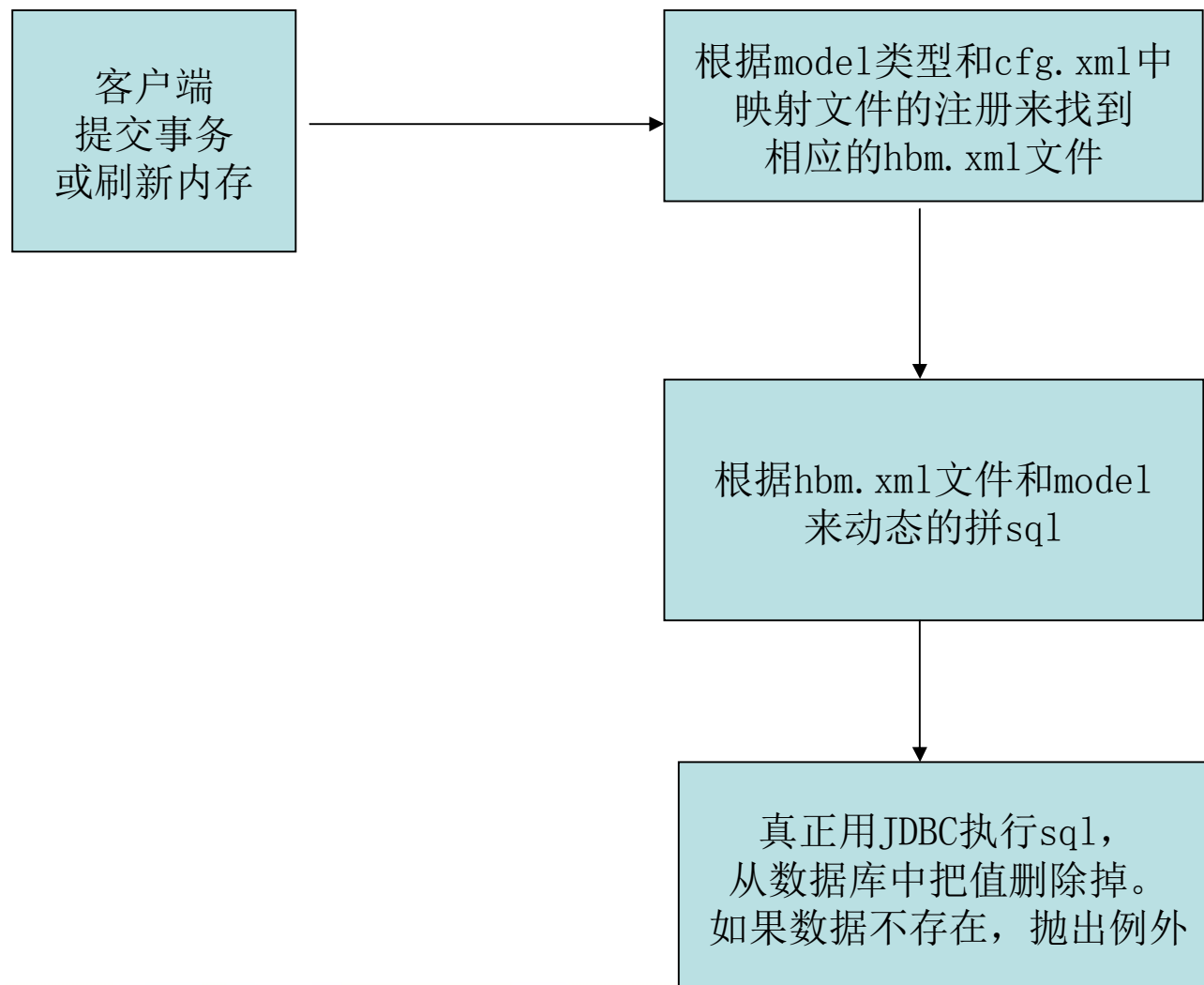
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### Delete(P0)方法基本流程

**n** 当我们调用了session.delete(UserModel)后：

- 1: 根据model的主键在hibernate的实例池中查找对应的实体对象(注意：代理对象不算实体对象)，找到就使用该对象。
- 2: 如果内存中没有对应的实体对象，就到数据库中查找来保证对象的存在，把找到的对象放到内存里面，而且不会标识成待删除的对象，可以继续被load等使用。代理对象也需要去数据库中查找数据。
- 3: 如果对象还是不存在，那么就抛出例外。也就是说，delete之前可能会执行一个查询语句。
- 4: 客户端提交事务或者刷新内存
- 5: 根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
- 6: 根据hbm.xml文件和model来动态的拼sql，如下：  
delete from 表名(来自hbm.xml) where 主键=值(来自model)
- 7: 真正用JDBC执行sql，把数据从数据库中删除

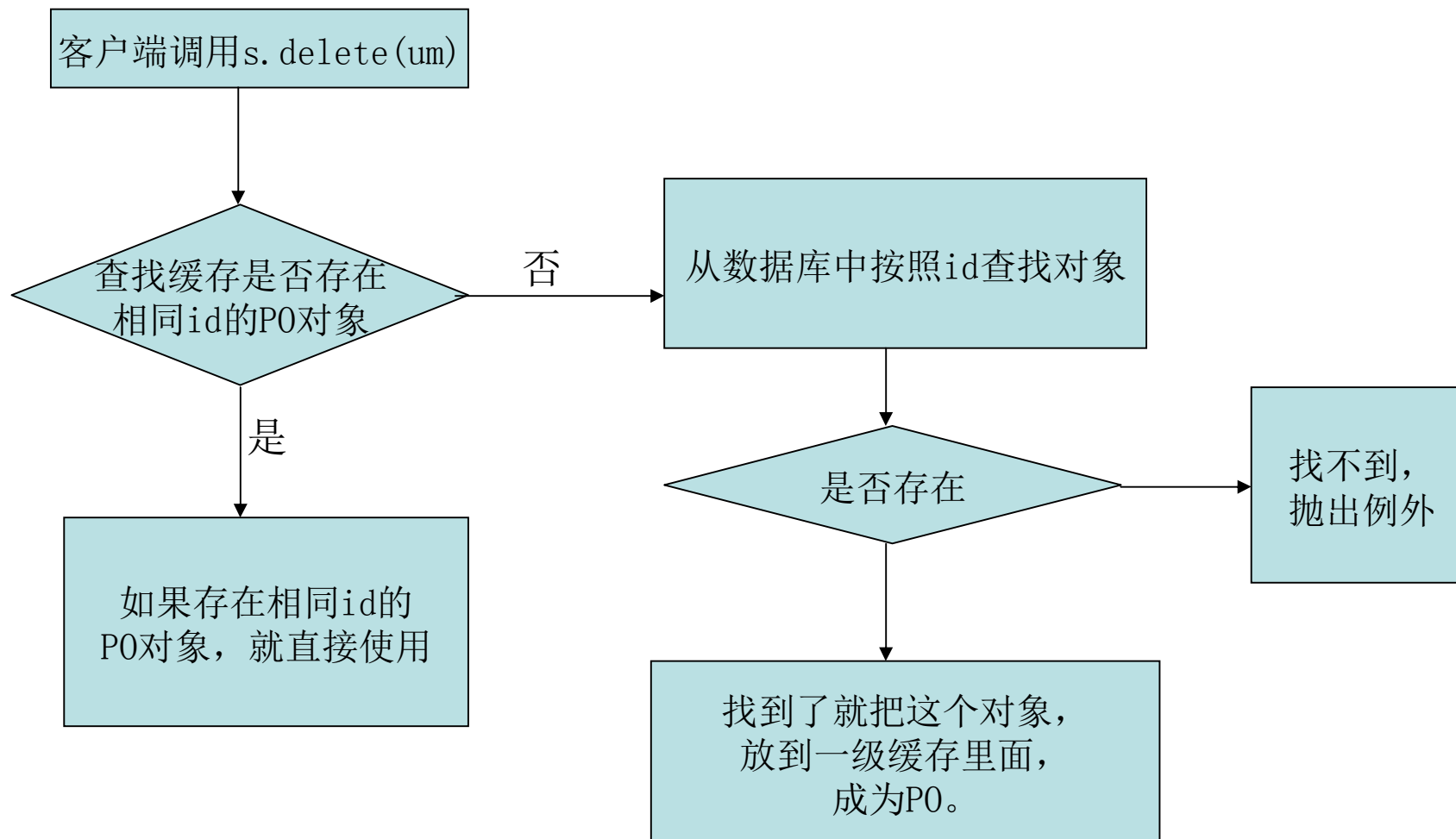
**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

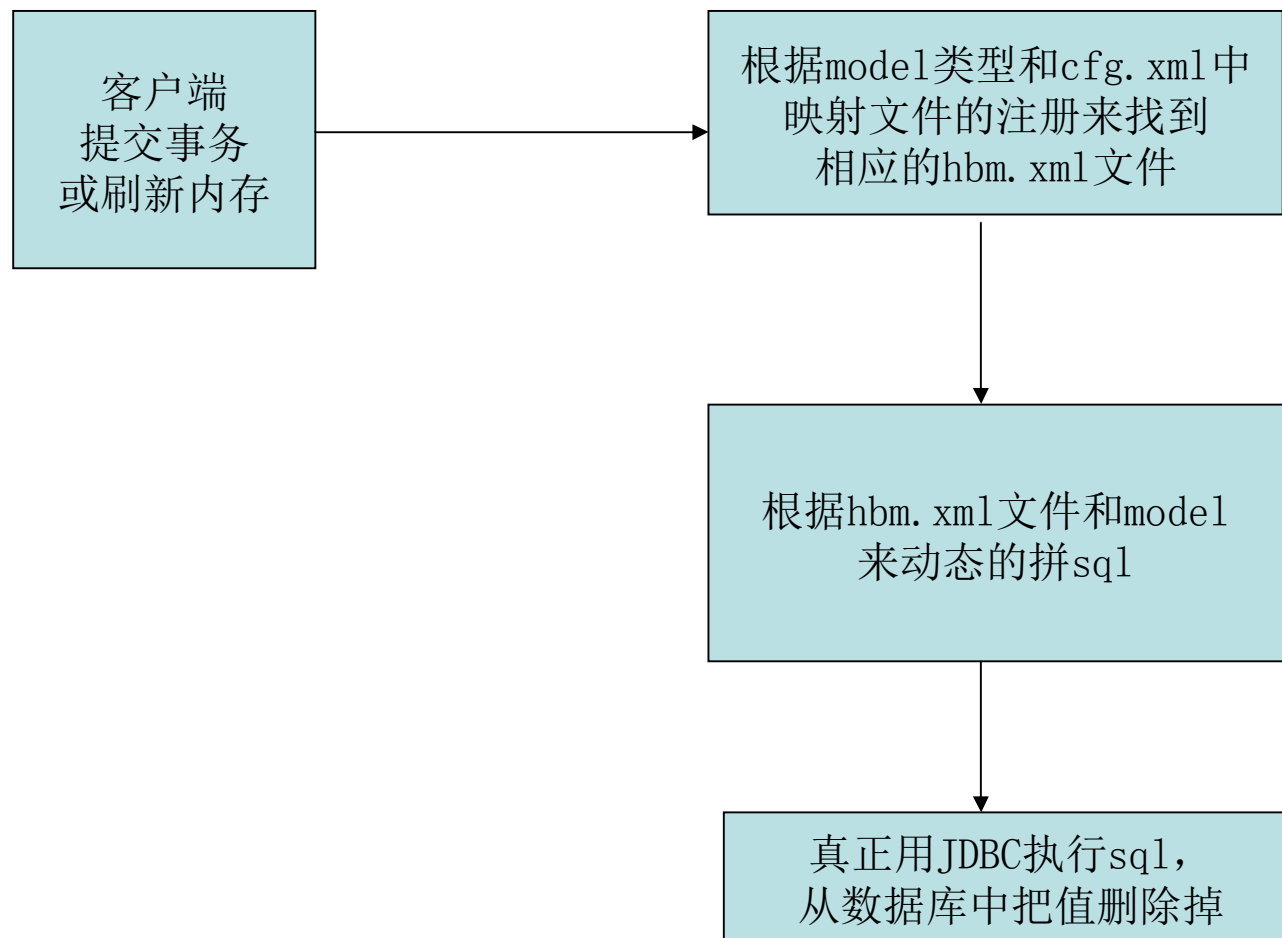
网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### load方法基本流程

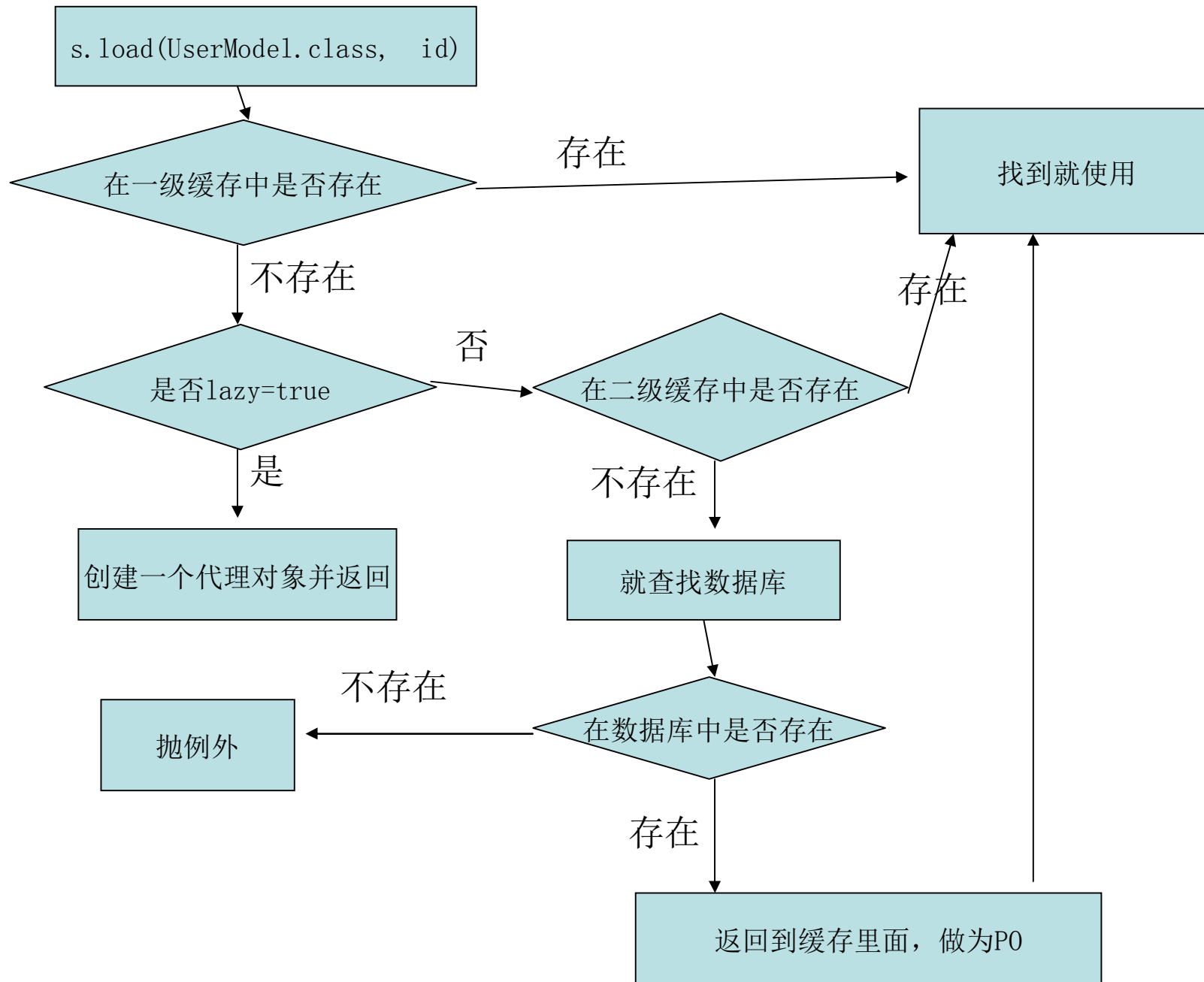
- n** 当我们调用了s.load(UserModel.class, “主键值”);后:
- 1: 根据model类型和主键值在一级缓存中查找对象，找到就返回该对象
  - 2: 如果没有找到，判断是否lazy=true，如果是，那就生成一个代理对象并返回；否则就先查找二级缓存，二级缓存没有，就查找数据库。如果是返回代理对象的，在第一次访问非主键属性时，先查找二级缓存，二级缓存中没有才真正查找数据库。
  - 3: 如果需要查找数据库的话，会根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
  - 4: 根据hbm.xml文件和model来动态的拼sql，如下：  
select 字段列表(来自hbm.xml) from 表名(来自hbm.xml) where 主键=值
  - 5: 真正用JDBC执行sql，把数据从数据库中查询出来到rs里面。如果找不到就报错
  - 6: 从结果集---> Model，然后返回model

注意：load方法开不开事务都可以执行查询语句。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

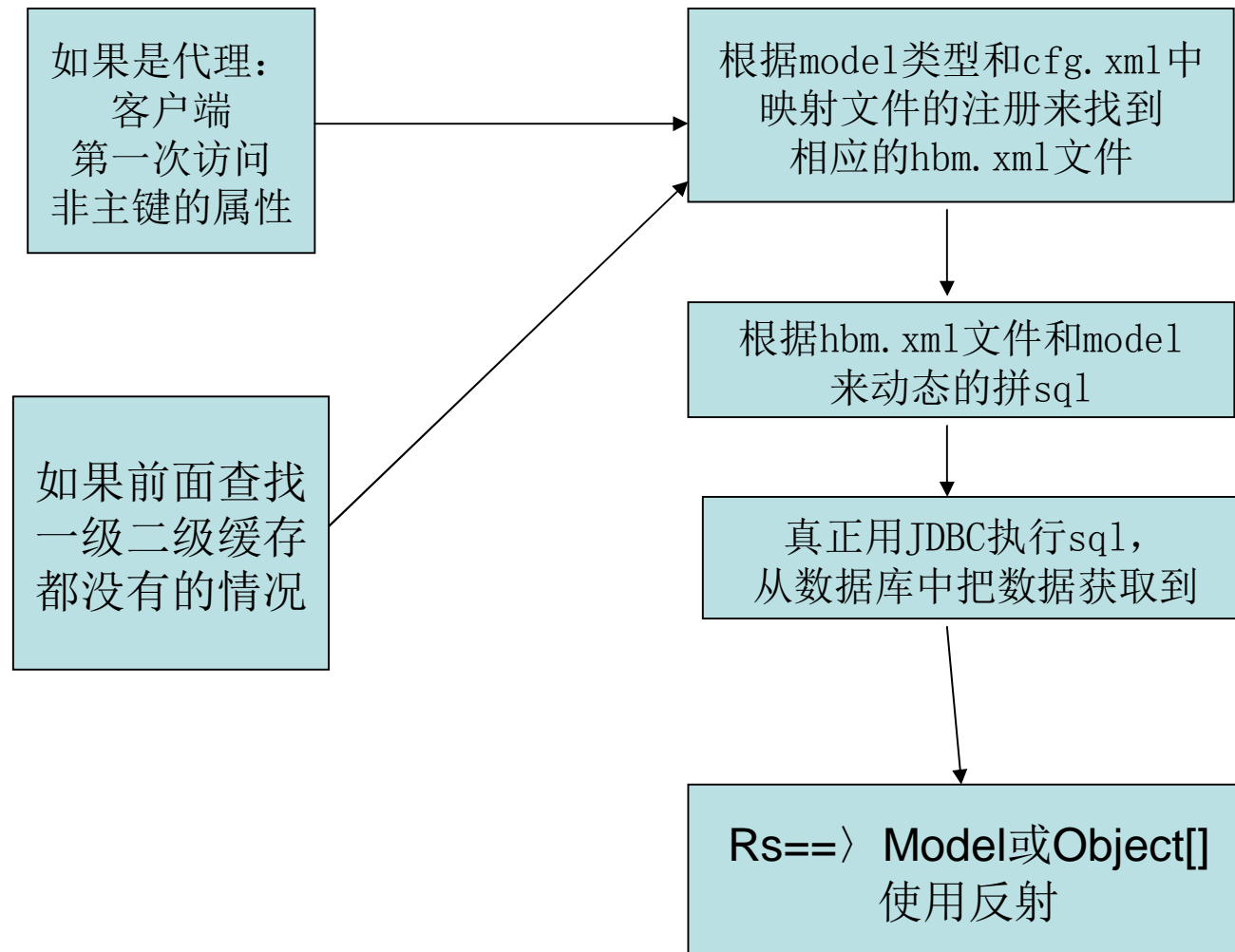
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送







## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### get方法基本流程

**n** 当我们调用了s.get(UserModel.class, “主键值”);后:

- 1: 先根据model类型和主键值查找缓存，如果存在具体的实体对象，就返回；如果存在实体的代理对象（比如前面load这条数据，但是还没有使用，那么load生成的是一个只有主键值的代理对象），那么查找数据库，把具体的数据填充到这个代理对象里面，然后返回这个代理对象，当然这个代理对象此时已经完全装载好数据了，跟实体对象没有什么区别了。
- 2: 如果要查找数据库，先根据model类型和cfg.xml中映射文件的注册来找到相应的hbm.xml文件
- 3: 根据hbm.xml文件和model来动态的拼sql，如下：  
select 字段列表(来自hbm.xml) from 表名(来自hbm.xml) where 主键=值
- 4: 真正用JDBC执行sql，把数据从数据库中查询出来到rs里面，没有值就返回null
- 5: 从结果集---> Model，然后返回model

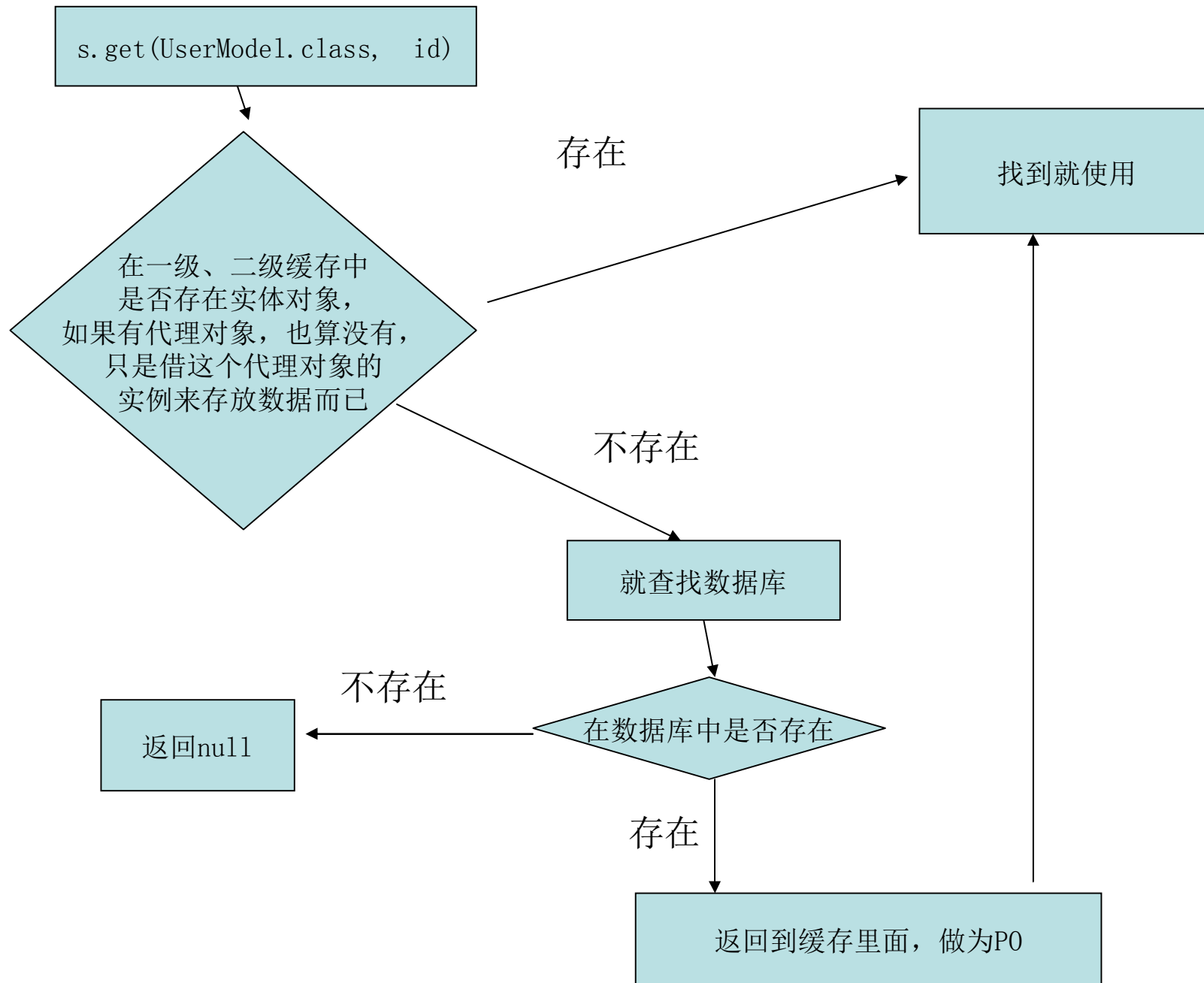
注意：get方法开不开事务都可以执行查询语句。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

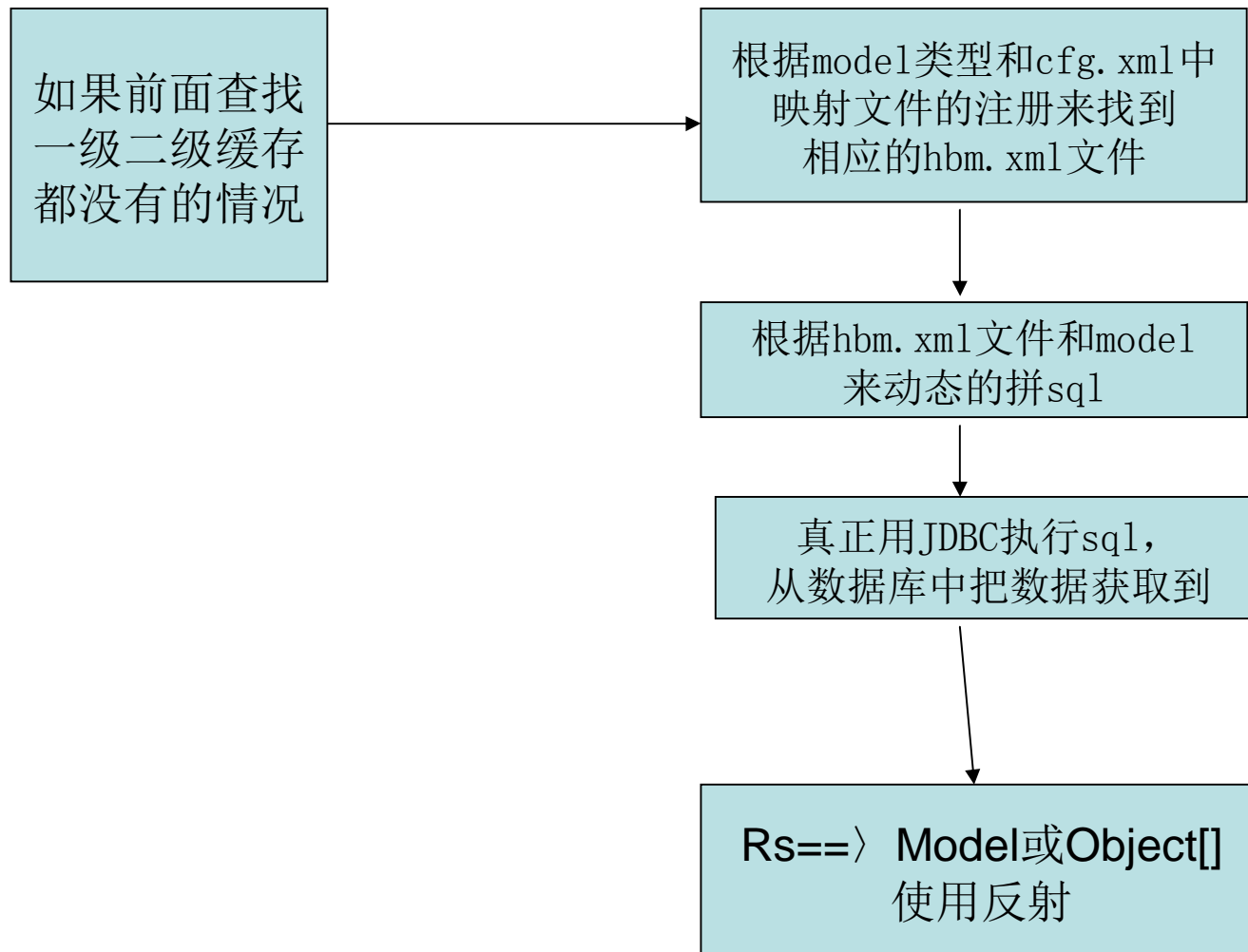
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程



**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### query方法基本流程

**n** 当我们调用了`q.list()`后：

- 1: 对HQL进行语义分析，分析出model来
- 2: 根据model类型和`cfg.xml`中映射文件的注册来找到相应的`hbm.xml`文件
- 3: 根据`hbm.xml`文件和model，来解析HQL，从而实现动态的把HQL转换成对应的sql，（从hql--->sql这个过程是非常复杂的，不但区分不同的数据库，还包括了对sql进行自动的优化），这里只能简单的示例如下：

select 字段列表(来自`hbm.xml`) from 表名(来自`hbm.xml`) where 条件

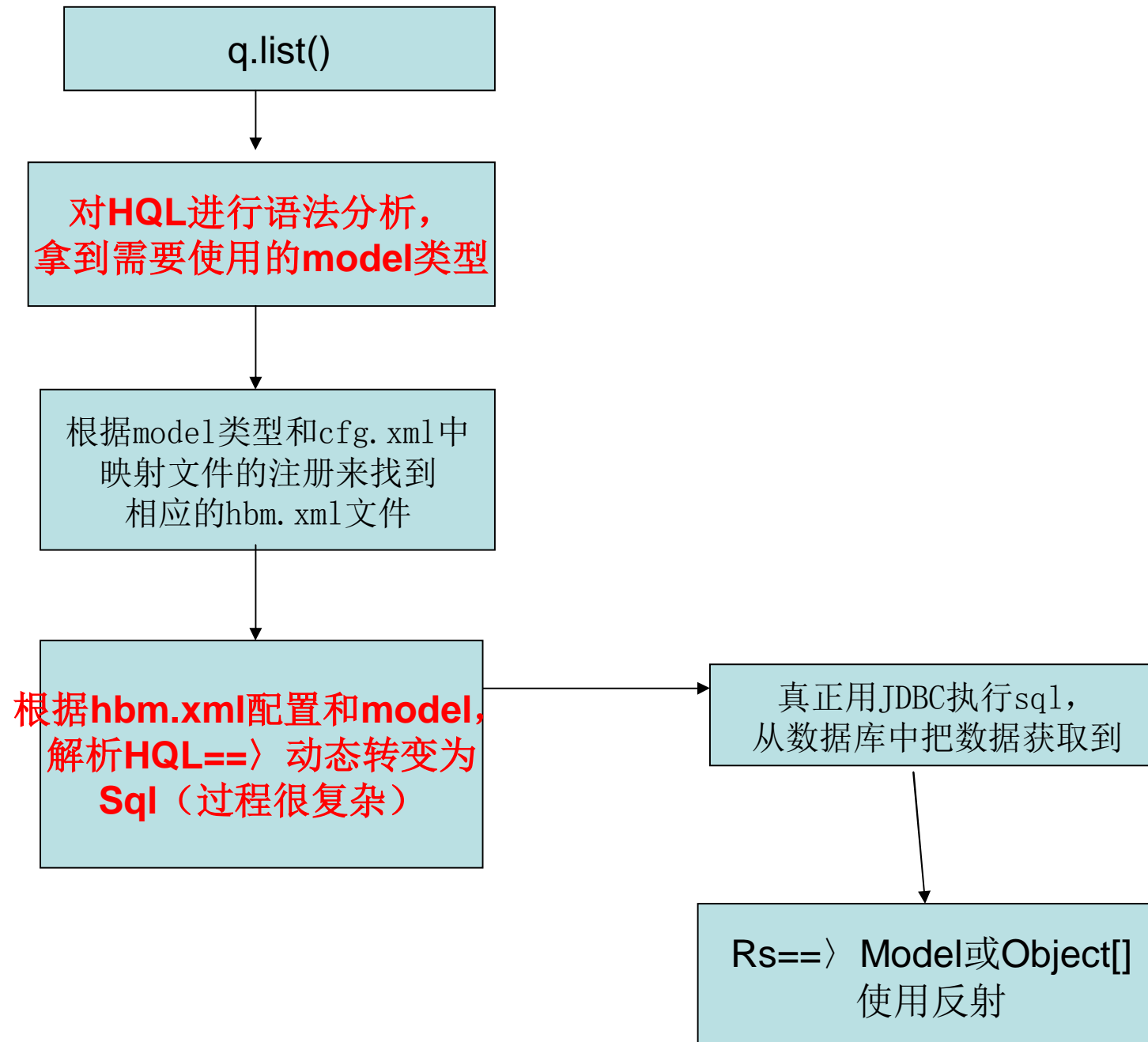
- 4: 真正用JDBC执行sql，把数据从数据库中查询出来到rs里面
- 5: 从结果集--->Model集合（或对象数组），然后返回model集合（或对象数组）

注意：`list()`方法开不开事务都可以执行查询语句。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送







## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程小结

- n 理解和掌握Hibernate的基本原理  
包括整体的运行流程，还有各个分方法描述的流程
  
- n 作业：
  - 1: 通过代码反复测试检验这些理论知识，以求理解和掌握。
  - 2: 所有流程图，最好是能够独立的、按照顺序画出来，并能解释每一步的功能。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程概览

n Hibernate的最佳实践

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



## 《深入浅出学Hibernate4开发》——系列精品教程

### 第九部分：最佳实践

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-1

- n 设计细颗粒度的持久类并且使用<component>来实现映射。

例如使用一个Address持久类来封装 street, suburb, state, postcode. 这有利于代码重用和简化代码重构(refactoring)的工作。

- n 对持久类声明标识符属性( identifier properties)。

Hibernate中标识符属性是可选的，不过有很多原因来说明你应该使用标识符属性。我们建议标识符应该是“人造”的(自动生成，不涉及业务含义)。

- n 使用自然键(natural keys)标识

对所有的实体都标识出自然键，用<natural-id>进行映射。实现equals()和hashCode()，在其中用组成自然键的属性进行比较。

- n 为每个持久类写一个映射文件

不要把所有的持久类映射都写到一个大文件中。把 com.eg.Foo 映射到 com/eg/Foo.hbm.xml中，在团队开发环境中，这一点显得特别有意义。

- n 把映射文件作为资源加载

把映射文件和他们的映射类放在一起进行部署。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-2

#### n 考虑把查询字符串放在程序外面

如果你的查询中调用了非ANSI标准的SQL函数，那么这条实践经验对你适用。把查询字符串放在映射文件中可以让程序具有更好的可移植性。

#### n 使用绑定变量

就像在JDBC编程中一样，应该总是用占位符“?”来替换非常量值，不要在查询中用字符串值来构造非常量值！更好的办法是在查询中使用命名参数。

#### n 不要自己来管理JDBC connections

Hibernate允许应用程序自己来管理JDBC connections，但是应该作为最后没有办法的办法。如果你不能使用Hibernate内建的connections providers，那么考虑实现自己来实现org.hibernate.connection.ConnectionProvider

#### n 考虑使用用户自定义类型(custom type)

假设你有一个Java类型，来自某些类库，需要被持久化，但是该类没有提供映射操作需要的存取方法。那么你应该考虑实现org.hibernate.UserType接口。这种方法使程序代码写起来更加自如，不再需要考虑类与Hibernate type之间的相互转换。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-3

#### n 在性能瓶颈的地方使用硬编码的JDBC

在系统中对性能要求很严格的一些部分，某些操作也许直接使用JDBC会更好。但是请先确认这的确是一个瓶颈，并且不要想当然认为JDBC一定会更快。如果确实需要直接使用JDBC，那么最好打开一个 Hibernate Session 然后从 Session获得connection，按照这种办法你仍然可以使用同样的transaction策略和底层的connection provider。

#### n 理解Session清洗（flushing）

Session会不时的向数据库同步持久化状态，如果这种操作进行的过于频繁，性能会受到一定的影响。有时候你可以通过禁止自动flushing，尽量最小化非必要的flushing操作，或者更进一步，在一个特定的transaction中改变查询和其它操作的顺序。

#### n 在三层结构中，考虑使用托管对象（detached object）

当使用一个servlet / session bean 类型的架构的时候，你可以把已加载的持久对象在session bean层和servlet / JSP 层之间来回传递。使用新的session来为每个请求服务，使用 Session.merge() 或者Session.saveOrUpdate() 来与数据库同步。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送





## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-4

- n 在两层结构中，考虑使用长持久上下文(long persistence contexts).

为了得到最佳的可伸缩性，数据库事务(Database Transaction)应该尽可能的短。但是，程序常常需要实现长时间运行的“应用程序事务(Application Transaction)”，包含一个从用户的观点来看的原子操作。这个应用程序事务可能跨越多次从用户请求到得到反馈的循环。用脱管对象(与session脱离的对象)来实现应用程序事务是常见的。或者，尤其在两层结构中，把Hibernate Session从JDBC连接中脱离开，下次需要用的时候再连接上。绝不要把一个Session用在多个应用程序事务(Application Transaction)中，否则你的数据可能会过期失效。

- n 不要把异常看成可恢复的

这一点甚至比“最佳实践”还要重要，这是“必备常识”。当异常发生的时候，必须要回滚 Transaction，关闭Session。如果你不这样做的话，Hibernate无法保证内存状态精确的反应持久状态。尤其不要使用 Session.load() 来判断一个给定标识符的对象实例在数据库中是否存在，应该使用Session.get() 或者进行一次查询。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-5

#### n 对于关联优先考虑lazy fetching

谨慎的使用主动抓取(eager fetching)。对于关联来说，若其目标是无法在第二级缓存中完全缓存所有实例的类，应该使用代理(proxies)与/或具有延迟加载属性的集合(lazy collections)。若目标是可以被缓存的，尤其是缓存的命中率非常高的情况下，应该使用lazy="false"，明确的禁止掉eager fetching。如果那些特殊的确实适合使用join fetch 的场合，请在查询中使用left join fetch。

#### n 使用open session in view模式，或者执行严格的装配期(assembly phase)策略来避免再次抓取数据带来的问题

Hibernate让开发者们摆脱了繁琐的Data Transfer Objects (DTO)。在传统的EJB结构中，DTO有双重作用：首先，他们解决了entity bean无法序列化的问题；其次，他们隐含地定义了一个装配期，在此期间，所有在view层需要用到的数据，都被抓取、集中到了DTO中，然后控制才被装到表示层。Hibernate终结了第一个作用。然而，除非你做好了在整个渲染过程中都维护一个打开的持久化上下文(session)的准备，你仍然需要一个装配期（想象一下，你的业务方法与你的表示层有严格的契约，数据总是被放置到托管对象中）。这并非是Hibernate的限制！这是实现安全的事务化数据访问的基本需求。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 最佳实践-6

#### n 考虑把Hibernate代码从业务逻辑代码中抽象出来

把Hibernate的数据存取代码隐藏到接口(interface)的后面，组合使用DAO和Thread Local Session模式。通过Hibernate的UserType，你甚至可以用硬编码的JDBC来持久化那些本该被Hibernate持久化的类。（该建议更适用于规模足够大应用软件中，对于那些只有5张表的应用程序并不适合。）

#### n 不要用怪异的连接映射

多对多连接用得好的例子实际上相当少见。大多数时候你在“连接表”中需要保存额外的信息。这种情况下，用两个指向中介类的一对多的连接比较好。实际上，我们认为绝大多数的连接是一对多和多对一的，你应该谨慎使用其它连接风格，用之前问自己一句，是否真的必须这么做。

#### n 偏爱双向关联

单向关联更加难于查询。在大型应用中，几乎所有的关联必须在查询中可以双向导航。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>

咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 《深入浅出学Hibernate4开发》——系列精品教程

### 本节课程小结

- n 理解这些Hibernate的最佳实践
- n 作业：复习和理解这些理论知识。

**真正高质量培训 签订就业协议**

网 址：<http://www.javass.cn>  
咨询QQ：460190900

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送