

# Hibernate 与数据库视图映射的解决方案

张旋<sup>1</sup> 高波<sup>2</sup> 孙闻<sup>1</sup>

(1.中国矿业大学计算机学院 江苏 徐州 221116;2.中国矿业大学信电学院 江苏 徐州 221116)

**【摘要】**Hibernate 是一个持久化框架,主要用于进行 O/R 映射。作为一种轻量级的框架,Hibernate 的目标是开发人员从 95%的数据持久化操作中解脱出来,Hibernate 从某种程度上已经取代 EJB,成为事实上的 ORM 标准了。但 Hibernate 也存在缺点,Hibernate 限制程序员所使用的对象模型。例如,一个持久性类不能映射到多个表。

**【关键词】**Hibernate;映射;数据库;视图

## 1.Hibernate 框架的简介

Hibernate 是一种 Java 语言下的对象关系映射解决方案。它是一种自由、开源的软件。它用来把对象模型表示的对象映射到基于 SQL 的关系模型结构中去,为面向对象的领域模型到传统的关系型数据库的映射,提供了一个使用方便的框架。

Hibernate 不仅管理 Java 类到数据库表的映射(包括从 Java 数据类型到 SQL 数据类型的映射),还提供数据查询和获取数据的方法,可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间。

它的设计目标是将软件开发人员从大量相同的数据持久层相关编程工作中解放出来。无论是从设计草案还是从一个遗留数据库开始,开发人员都可以采用 Hibernate。

Hibernate 对 JDBC 进行了非常轻量级的对象封装,使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。Hibernate 可以应用在任何使用 JDBC 的场合,它既可以在 Java 的客户端程序使用,也可以在 Servlet/JSP 的 Web 应用中使用。最具革命意义的是,Hibernate 可以在应用 EJB(Enterprise JavaBeans 是 Java 应用于企业计算的框架)的 J2EE 架构中取代 CMP,完成数据持久化的重任。

## 2.Hibernate 对普通数据库表的映射

要实现 Hibernate 对数据库的操作需要完成以下几项工作,一个持久化类,数据库及与持久化类对应的数据表、配置文件、映射文件。现在有一个数据表如下:

```
create table student
(
  stud_id char(10) not null default '0401010101', --学号
  name nvarchar(10) not null, --姓名
  birthday datetime, --出生日期
  gender nchar(1), --性别
  address nvarchar(20), --家庭住址
  telcode char(12), --电话
  zipcode char(6), --邮编
  constraint PK_student primary key (stud_id)
)
```

一个与之对应的持久化类如下:

```
package model;
import java.util.Date;
public class student {
  private String stud_id; //学号
  private String name; //姓名
  private Date birthday; //出生日期
  private String gender; //性别
  private String address; //家庭住址
  private String telcode; //电话
  private String zipcode; //邮编
  ..... //此处是属性的 get/set 方法,在此略
}
```

下面需要编写数据库表与类的映射文件 student.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="hibernate.model.student" table="student">
```

```
<!-- 将 stud_id 属性和 stud_id 字段关联起来 -->
<id name="stud_id" column="stud_id" type="string"></id>
<!-- 将 name 属性和 name 字段关联起来 -->
<property name="name" column="name" type="string"></
property>
..... //其他属性的映射,略
</class>
</hibernate-mapping>
最后需要在 hibernate.cfg.xml 中引入 mapping 文件。
<?xml version="1.0" encoding="UTF-8"?>
<! DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0/EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<!-- Generated by MyEclipse Hibernate Tools.-->
<hibernate-configuration>
<session-factory>
<property name="dialect">
org.hibernate.dialect.SQLServerDialect
</property>
<property name="connection.url">
jdbc:sqlserver://127.0.0.1:1433;DatabaseName=hibernate
</property>
<property name="connection.username">sa</property>
<property name="connection.password">aaa</property>
<property name="connection.driver_class">
com.microsoft.sqlserver.jdbc.SQLServerDriver
</property>
<property name="myeclipse.connection.profile">
com.microsoft.sqlserver.jdbc.SQLServerDriver
</property>
<mapping resource="hibernate/model/student.hbm.xml" />
</session-factory>
</hibernate-configuration>
```

到此为止,Hibernate 与普通数据库表映射的配置工作就完成了,下面就可以通过 Hibernate 提供的接口来实现对数据库的操作。

## 3.Hibernate 与数据库视图的映射

现在我们还有三个表,分别为 teacher、lesson、stud\_grade 表,它们拥有的关键字段如下:teacher 表:teacher\_id、name、course\_id、...

Lesson 表:course\_id、course\_name、...

stud\_grade 表:stud\_id、course\_id、grade、...

现在我们想分别获得 student 表的 name 字段、teacher 表 name 字段、Lesson 表 course\_name 字段、stud\_grade 表的 grade 字段,并且把它们封装到一个持久化类里,这样我们就可以方便的查询出实体,然后再前台通过<s:iterator>等标签来显示了。解决的方法其实很简单,我们可以再数据库里创建一个视图,然后再建立一个与之对应的持久化类,通过映射文件将两者关联起来。最后引进 hibernate.cfg.xml 文件里,就可以了。具体实现步骤如下:

创建视图:

```
create view stud_grade
as
(
select s.name,l.course_name,g.grade,t.name
```

(下转第 28 页)

# 子网掩码

孙斌<sup>1</sup> 金梦<sup>2</sup> 柴婷婷<sup>2</sup>

(1.中国矿业大学环境与测绘学院 江苏 徐州 221116;

2.中国矿业大学计算机科学与技术学院 江苏 徐州 221116)

**【摘要】**本文首先介绍了子网掩码的概念,然后介绍了子网掩码的作用,最后通过举例分析介绍了子网掩码的划分方法,可以使我们对子网有了更加深刻的了解。

**【关键词】**子网掩码的概念;子网掩码的作用;子网掩码的划分方法

## 1.子网掩码的概念

子网掩码是一个32位地址,是与IP地址结合使用的一种技术。它的主要作用有两个,一是用于屏蔽IP地址的一部分以区别网络标识和主机标识,并说明该IP地址是在局域网,还是在远程网上。二是用于将一个大的IP网络划分为若干小的子网络。

## 2.子网掩码的作用

通过IP地址的二进制与子网掩码的二进制进行与运算,确定某个设备的网络地址和主机号,也就是说通过子网掩码分辨一个网络的网络部分和主机部分。子网掩码一旦设置,网络地址和主机地址就固定了。子网一个最显著的特征就是具有子网掩码。与IP地址相同,子网掩码的长度也是32位,也可以使用十进制的形式。例如,为二进制形式的子网掩码:1111 1111.1111 1111 1111.0000 0000,采用十进制的形式为:255.255.255.0。

## 3.子网掩码的划分方法

子网划分IP地址,也称为子网标识。子网掩码是可用点分十进制表示的32位二进制数,掩码告诉网络中的端系统IP地址的多少位用于识别网络和子网,这些被称为扩展的网络前缀。剩下的位标识子网内的主机,掩码中用于标识网络号的位置为1,主机位置为0。

如果要将一个网络划分成多个子网,如何确定这些子网的子网掩码和IP地址中的网络号和主机号呢。下面将会详细介绍子网划分的方法。

第1步,将要划分的子网数目转换为2的m次方。如要分8个子网,8=2<sup>3</sup>。如果不是恰好是2的多少次方,则取大为原则,如要划分为6个,则同样要考虑2<sup>3</sup>。

第2步,将上一步确定的幂m按高序占用主机地址m位后,转换为十进制。如m为3表示主机位中有3位被划为“网络标识号”占用,因网络标识号应全为“1”,所以主机号对应的字节段为“11100000”。转换成十进制后为224,这就最终确定的子网掩码。如果是C类网,则子网掩码为255.255.224.0;如果是B类网,则子网掩码为255.255.224.0;如果是A类网,则子网掩码为255.224.0.0。

在这里,子网个数与占用主机地址位数有如下等式成立:2<sup>m</sup>≥n。其中,m表示占用主机地址的位数;n表示划分的子网个数。根据这些原则,将一个C类网络分成4个子网。

为了说明问题,现再举例。若我们用的网络号为192.9.200,则该C类网内的主机IP地址就是192.9.200.1~192.9.200.254,现将网络划分为4个子网,按照以上步骤,4=2<sup>2</sup>,则表示要占用主机地址的2个高序位,即为11000000,转换为十进制为192。这样就可确定该子网掩码为:192.9.200.192。4个子网的IP地址的划分是根据被网络号占住的两位排列进行的,这四个IP地址范围分别为:

(1)第1个子网的IP地址是从“11000000 00001001 11001000 00000001”到“11000000 00001001 11001000 00111110”,注意它们的最后8位中被网络号占住的两位都为“00”,因为主机号不能全为“0”和“1”,所以没有11000000 00001001 11001000 00000000和11000000 00001001 11001000 00111111这两个IP地址(下同)。注意实际上此时的主机号只有最后面的6位。对应的十进制IP地址范围为192.9.200.1~192.9.200.62。而这个子网的子网掩码(或网络地址)为11000000 00001001 11001000 00000000,为192.9.200.0。

(2)第2个子网的IP地址是从“11000000 00001001 11001000 01000001”到“11000000 00001001 11001000 01111110”,注意此时被网络号所占住的2位主机号为“01”。对应的十进制IP地址范围为192.9.200.65~192.9.200.126。对应这个子网的子网掩码(或网络地址)为11000000 00001001 11001000 01000000,为192.9.200.64。

(3)第3个子网的IP地址是从“11000000 00001001 11001000 10000001”到“11000000 00001001 11001000 10111110”,注意此时被网络号所占住的2位主机号为“10”。对应的十进制IP地址范围为192.9.200.129~192.9.200.190。对应这个子网的子网掩码(或网络地址)为11000000 00001001 11001000 10000000,为192.9.200.128。

(4)第4个子网的IP地址是从“11000000 00001001 11001000 11000001”到“11000000 00001001 11001000 11111110”,注意此时被网络号所占住的2位主机号为“11”。对应的十进制IP地址范围为192.9.200.193~192.9.200.254。对应这个子网的子网掩码(或网络地址)为11000000 00001001 11001000 11000000,为192.9.200.192。

## 4.总结

本文首先介绍了子网的概念、作用,然后介绍了划分子网的方法,使我们对子网有了更加深刻的认识。

(上接第97页)from student s,teacher t,lesson l,stud\_grade g  
where s.stud\_id=g.stud\_id and  
l.course\_id=g.course\_id and  
g.course\_id=t.course\_id

```
)
创建持久化类:
package model;
public class stud_grade {
    private String stud_name ;           //姓名
    private String teac_name;           //教师姓名
    private String course_name;         //课程名
    private String grade;               //分数
    .....                               //此处是属性的 get/set 方法,在此略
```

}
映射文件和配置文件的写法与前面的相同。
这样我们就可以像操作普通数据表一样来操作视图了。

## 4.结语

作为一种应用前景非常广阔的数据库持久化框架,Hibernate的发展是非常迅速的,它的未来发展也是非常光明的,我们期待着Hibernate带给我们更大的喜悦。

## 【参考文献】

- [1]孙卫琴.精通 Hibernate:Java 对象持久化技术详解[M].电子工业出版社,2005.
- [2]李宁.Java Web 开发技术大全.清华大学出版社,2009.
- [3]蔡雪燕.Hibernate 开发及整合应用[M].清华大学出版社,2006.