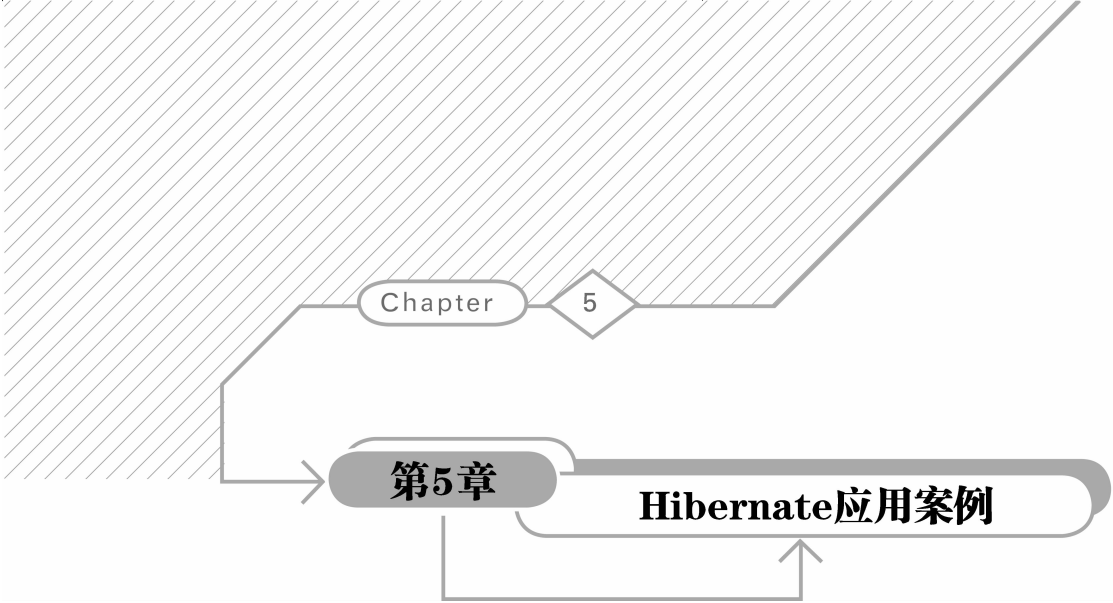




5.1 案例 1——多对一和一对多关联

多对一和一对多关联是互逆的两种关系,可通过设立两张表的外键关系,同时实现多对一和一对多关联。

5.1.1 工程框架搭建

1. 新建数据库和表

新建数据库 RelationTest,然后新建 person 表和 room 表。

1) person 表

person 表字段如图 5.1 所示,主键设置如图 5.2 所示。

SKY-20160123UKM....Test - dbo.person			
	列名	数据类型	允许 Null 值
	id	int	☐
	name	varchar(50)	☒
	room_id	int	☒

图 5.1 person 表

表设计器	
Text/Image 文件组	PRIMARY
标识列	id
常规数据空间规范	id
是可索引的	room_id
锁升级	表

图 5.2 设置主键

注意：id 字段是主键,而且必须是标识列(即自增型)。标识列必须在保存表之前就设置好,一旦保存表后,就不能再做设置标识列的操作。

2) room 表

room 表字段如图 5.3 所示,主键设置如图 5.4 所示。

SKY-20160123UKM...onTest - dbo.room		
列名	数据类型	允许 Null 值
id	int	☐
address	varchar(100)	☒

图 5.3 room 表

表设计器	
Text/Image 文件组	PRIMARY
标识列	id
常规数据空间规范	id
是否索引的	是

图 5.4 设置主键

2. 设置外键

在设置外键前,必须将这两张表中的数据全部删除,否则无法创建外键。在 SQL Server 2008 中必须在外键表中才能建外键,所以必须首先打开外键表。

打开外键表 person,如图 5.5 所示。

单击“关系”按钮,如图 5.6 所示。

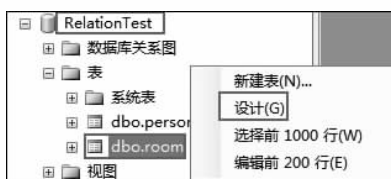


图 5.5 打开外键表 person



图 5.6 单击“关系”按钮

单击“添加”按钮,如图 5.7 所示。

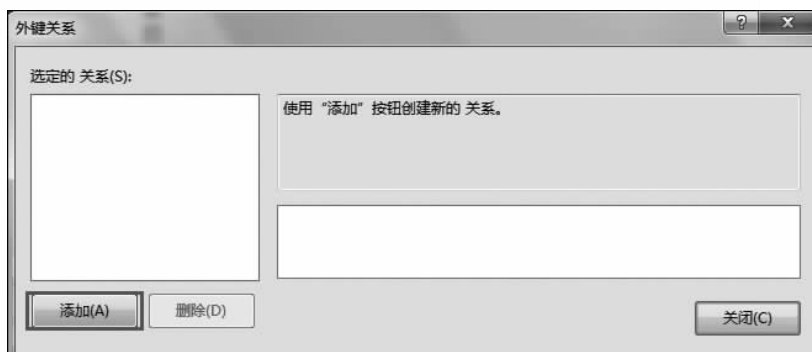


图 5.7 单击“添加”按钮

单击“表和列规范”项的按钮,如图 5.8 所示。

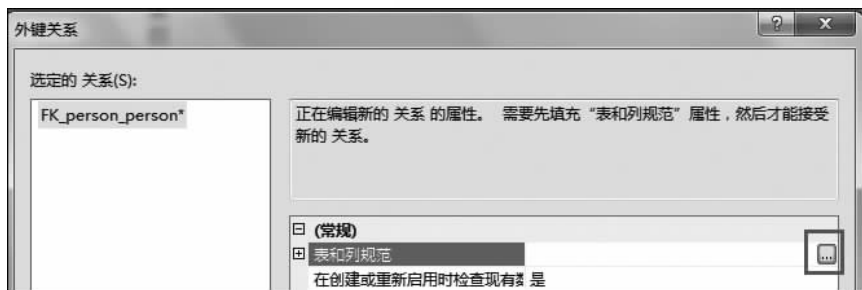


图 5.8 单击“表和列规范”项的按钮

选择主键表 room 和主键表的外键列 id,再选择外键表 person 的外键列 room_id,如图 5.9 所示。

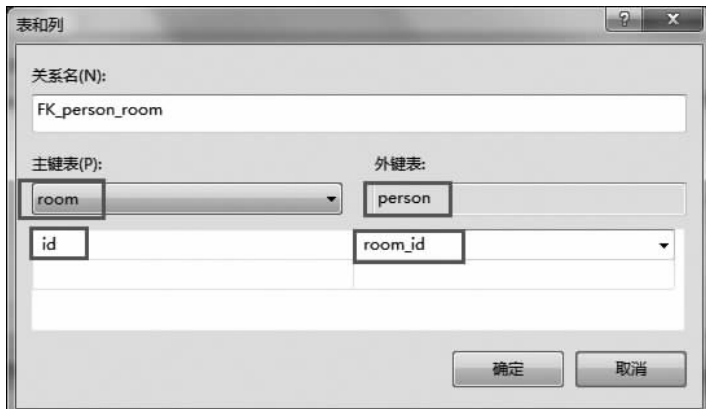


图 5.9 设置主键和外键

注意：主键表通常都是简单表,即字段少的表,而外键表是复杂表,即字段多的表,主键表是被引用的对象。在一对多的关系中,多方是外键表,而且包含一个外键字段。

打开数据库关系图,新建一个数据库关系图,可以看到两张表之间存在一个外键关系 FK_person_room,如图 5.10 所示。

3. 新建 Java 工程

工程名为 MulToOne,如图 5.11 所示。

新建 lib 目录,如图 5.12 所示。

复制 sqljdbc4.jar 包到 lib 目录下,并添加该 jar 包的引用,如图 5.13 所示。

打开 Db Browser 视图,然后新建一个连接,选择 Microsoft SQL Server 2005 模板,然后修改 Connection URL,单击 Add JARs 按钮,选择 Microsoft SQL Server 的驱动连接 jar 包所在的位置,如图 5.14 所示。

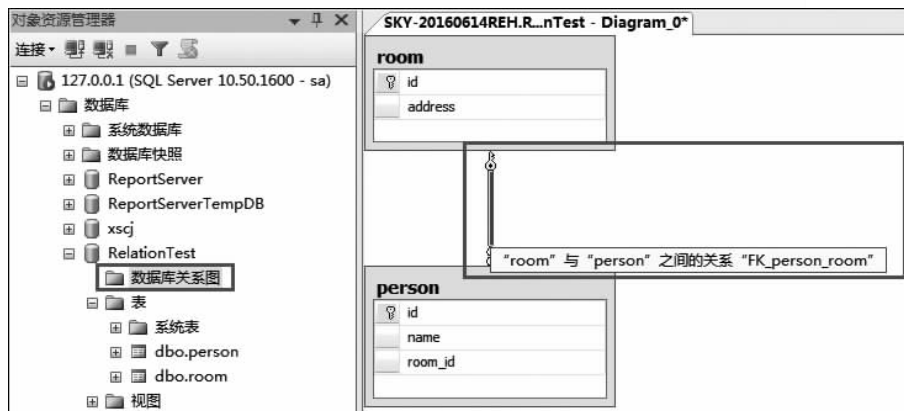


图 5.10 数据库关系图

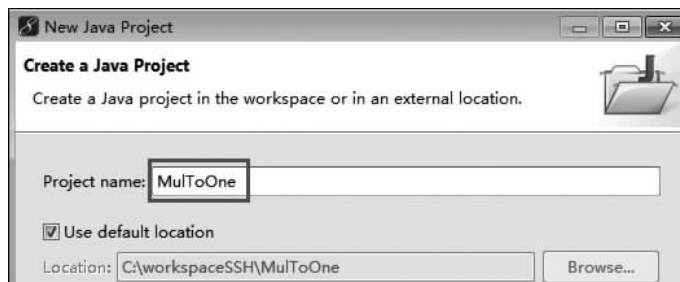


图 5.11 新建 Java 工程

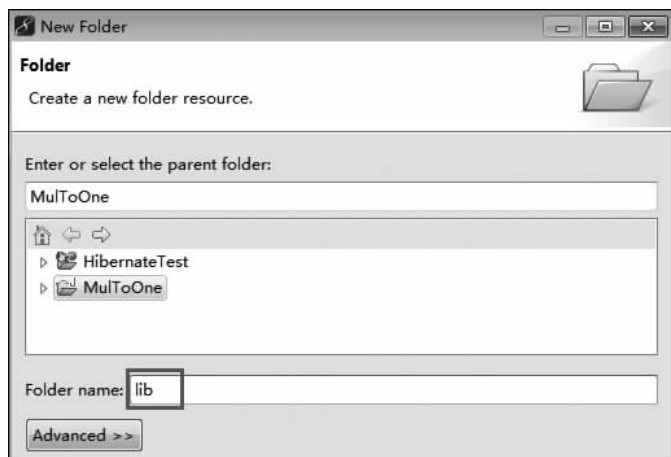


图 5.12 新建 lib 目录

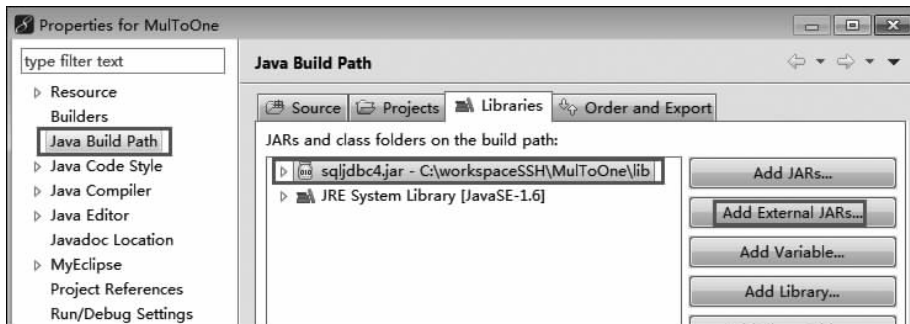


图 5.13 添加 sqljdbc4.jar 包的引用

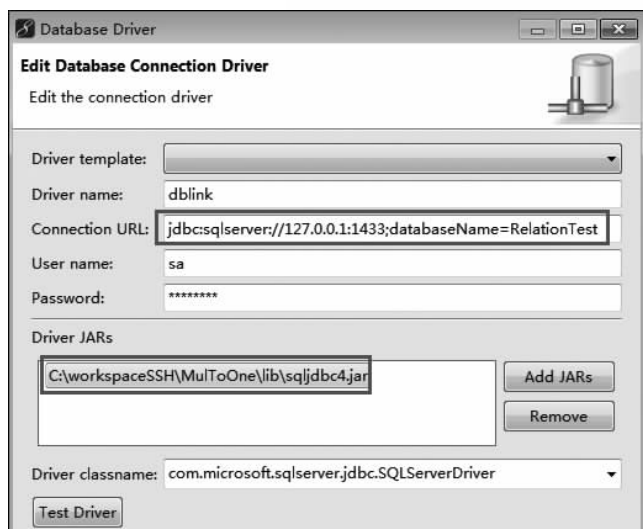


图 5.14 新建数据库连接

4. 添加 Hibernate 支持

右键单击工程,添加 Hibernate Capabilities 功能支持,如图 5.15 所示。

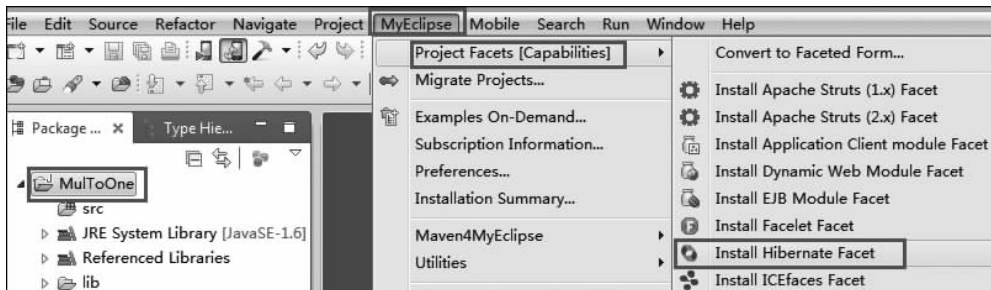


图 5.15 添加 Hibernate 支持

选择 Hibernate 的版本为 3.2,如图 5.16 所示。

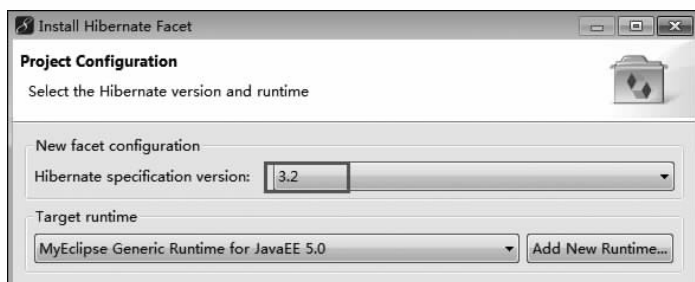


图 5.16 选择 Hibernate 的版本

单击 Java package 后的 New 按钮,新建一个包,并在该包下,自动生成类 HibernateSessionFactory 类,如图 5.17 所示。

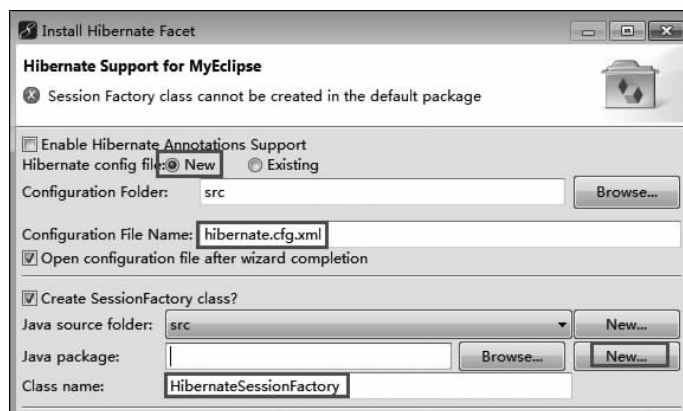


图 5.17 生成 HibernateSessionFactory 类

输入新的包名 org.util,如图 5.18 所示。

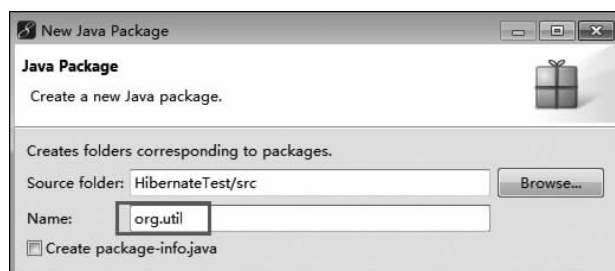


图 5.18 输入新的包 org.util

此时,可以看到新的包名 org.util,并在该包下新建 HibernateSessionFactory 类,如图 5.19 所示。

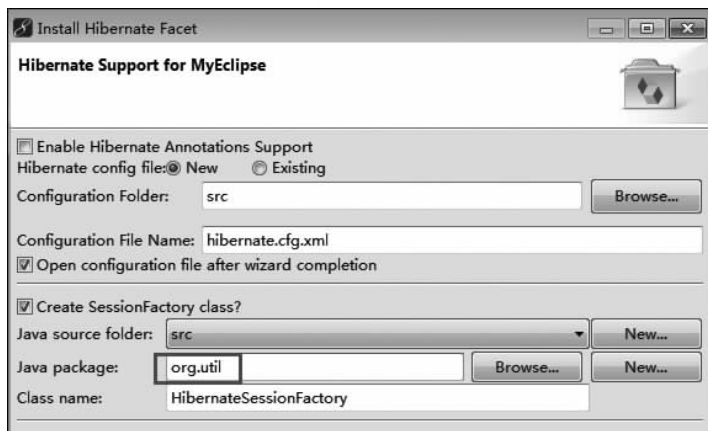


图 5.19 新建 HibernateSessionFactory 类

选择刚才建好的 dblink, 如图 5.20 所示。

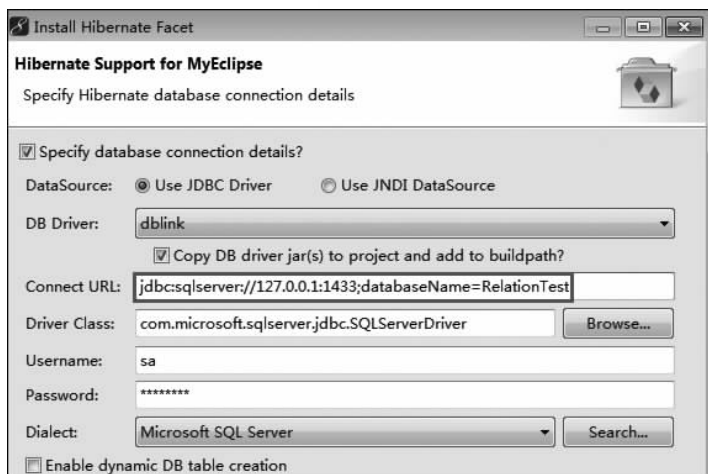


图 5.20 选择 dblink

选择使用 MyEclipse 自带的 Hibernate 库包, 如图 5.21 所示。



图 5.21 选择自带的 Hibernate 库包

不要选择 MyEclipse 自带的 sqljdbc4.jar,如图 5.22 所示。



图 5.22 不选择自带的 sqljdbc4.jar

5.1.2 实体类创建

1. 反向工程 room 表

反向工程 room 表,room 表是主键表,将生成 Room 类和 Room.hbm.xml 文件。
单击 Java src folder 边的 Browse 按钮,选择工程下的 src 文件夹,如图 5.23 所示。

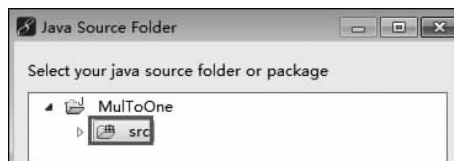


图 5.23 选择 src 文件夹

此时,对话框中其他选项被激活,如图 5.24 所示。

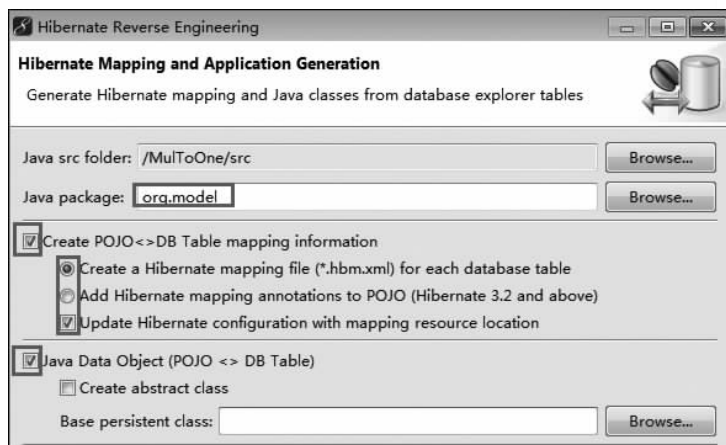


图 5.24 新建包,选择正确的选项

选择主键生成方式为 native,如图 5.25 所示。

此时,工程中将会有 3 个地方发生变化。

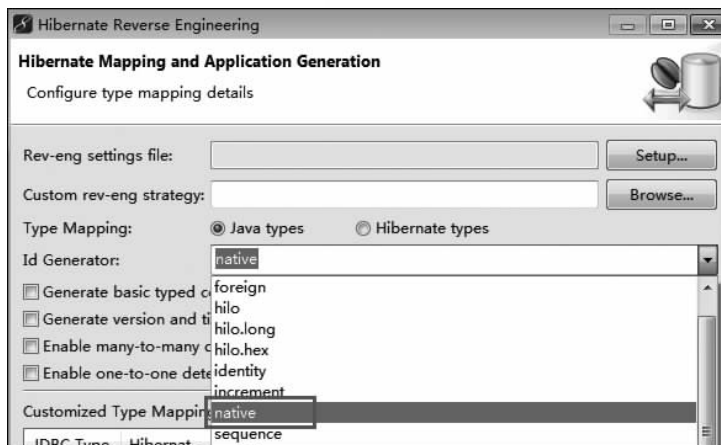


图 5.25 选择主键生成方式

1) Room 类

Room 类中增加了 persons 这个集合属性,Room.java 代码如下:

```
package org.model;
import java.util.HashSet;
import java.util.Set;
public class Room implements java.io.Serializable {
    private Integer id;
    private String address;
    //一个房间对应多个人,采用外键方式将自动增加该字段,在 room 表中是没有这个字段的
    private Set persons=new HashSet(0);
    public Room() {
    }
    public Room(String address) {
        this.address=address;
    }
    public Room(String address, Set persons) {
        this.address=address;
        this.persons=persons;
    }
    public Integer getId() {
        return this.id;
    }
    public void setId(Integer id) {
        this.id=id;
    }
}
```

```

    public String getAddress() {
        return this.address;
    }
    public void setAddress(String address) {
        this.address=address;
    }
    public Set getPersons() {
        return this.persons;
    }
    public void setPersons(Set persons) {
        this.persons=persons;
    }
}

```

2) Room.hbm.xml

在 Room.hbm.xml 中增加了 persons 这个 Set 集合属性的描述,即自动建立了一对多关联 one-to-many。一对多关联必定是一个集合 set 属性。

room 表只有两个字段,如图 5.26 所示。

SKY-20160123UKM....onTest - dbo.room			
	列名	数据类型	允许 Null 值
	id	int	☐
	address	varchar(100)	☒

图 5.26 room 表(主键表,“一”方)

但由于增加了外键,将新增一个 person 对象的 set 属性 persons。person 表有 3 个字段,但由于增加了外键,外键列 room_id 变成主键表对象 room 属性。

Room.hbm.xml 代码如下所示:

```

<?xml version="1.0" encoding="utf-8"?>
<hibernate-mapping>
    <class name="org.model.Room" table="room" schema="dbo" catalog="xscj">
        <id name="id" type="java.lang.Integer">
            <column name="id" />
            <generator class="native" />
        </id>
        <property name="address" type="java.lang.String">
            <column name="address" length="100" not-null="true" />
        </property>
        <!--该 persons 字段 Room 类中的一个集合 Set 属性,注意: 由于它是集合对象,不能
        能用 property,必须用 set 属性-->
        <set name="persons" inverse="true">
            <key>
                <!--person 字段对应外键表(Person 表)中的外键字段(room_id)-->

```