

需求：公民和身份证之间的关系是明显的一对一的关系。

person端：

@Id

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Column(name = "id")
private Integer id;
@Column(name = "name")
private String name;
@Column(name = "age")
private Integer age;
// 关联身份证
@OneToOne(mappedBy="person")
//mappedBy="本端实体在另一端实体中的属性名"
private card card;
```

card端：

@Id

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
@Column(name = "id")
private Integer id;
@Column(name = "cardname")
private String cardname;
// 关联人
@OneToOne
@JoinColumn(name="idd")
//name="另一端主键在本端作为外键的列名"
private person person;
```

hibernate.cfg.xml配置文件：

```
<mapping class="第一个实体类的路径" />
<mapping class="第二个实体类的路径" />
```

demo实操代码：

```
/**
```

```
 * 一对一映射
```

```
 */
```

```
@Test
```

```
public void test4() {
```

```
    Session session = hibernateUtils.getSession();
```

```
    Transaction ts = session.beginTransaction();
```

```
    person p1 = new person();
```

```
    p1.setId(1);
```

```
    p1.setName("www");
```

```
    p1.setAge(23);
```

```
    card c1 = new card();
```

```
    c1.setId(1);
```

```
    c1.setCardname("金融卡");
```

```
    c1.setPerson(p1);
```

```
    session.save(p1);
```

```
    session.save(c1);
```

```
    //其中需要注意的是，没有含有外键的一端要先操作，含有外键的一端要在后面操作，否则会报错。
```

```
    ts.commit();
```

```
    session.close();
```

```
}
```