# JPA 之 一对一

1. @OneToOne关系映射

JPA使用@OneToOne来标注一对一的关系。实体 People ：用户。 实体 Address：家庭住址。

People 和 Address 是一对一的关系。

这里用两种方式描述JPA的一对一关系。

1 一种是通过外键的方式(一个实体通过外键关联到另一个实体的主键)；

2 另外一种是通过一张关联表来保存两个实体一对一的关系。

**1、通过外键的方式**

people 表（id，name，sex，birthday，address\_id）

address 表（id，phone，zipcode，address）

**People.java**

@Entity

@Data

@Table(name = "people")

public class People {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", nullable = false)

private Long id;

@Column(name = "name", nullable = true, length = 20)

private String name;

@Column(name = "sex", nullable = true, length = 1)

private String sex;

@Column(name = "birthday", nullable = true)

private Timestamp birthday;

@OneToOne(fetch=FetchType.LAZY, cascade=CascadeType.ALL)

//People是关系的维护端，当删除 people，会级联删除 address , 懒加载

@JoinColumn(name = "address\_id", referencedColumnName = "id")

//people中的address\_id字段参考address表中的id字段

private Address address;//地址

}

关联的实体的主键一般是用来做外键的。但如果此时不想主键作为外键，则需要设置referencedColumnName属性。当然这里关联实体(Address)的主键 id 是用来做主键，所以这里第20行的 referencedColumnName = "id" 实际可以省略。

**Address.java**

@Entity

@Data

@Table(name = "address ")

public class Address {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", nullable = false)

private Long id;

@Column(name = "phone", nullable = true, length = 11)

private String phone;

@Column(name = "zipcode", nullable = true, length = 6)

private String zipcode;

@Column(name = "address", nullable = true, length = 100)

private String address;

//如果不需要根据Address级联查询People，可以注释掉

// @OneToOne(mappedBy = "address", cascade = {CascadeType.MERGE, CascadeType.REFRESH}, optional = false)

// private People people;

}

**2、通过关联表的方式来保存一对一的关系。**

people 表（id，name，sex，birthday）

address 表 (id，phone，zipcode，address）

people\_address (people\_id，address\_id)

只需要创建 People 和 Address 两个实体

**People.java**

@Entity

public class People {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "id", nullable = false)

private Long id;//id

@Column(name = "name", nullable = true, length = 20)

private String name;//姓名

@Column(name = "sex", nullable = true, length = 1)

private String sex;//性别

@Column(name = "birthday", nullable = true)

private Timestamp birthday;//出生日期

@OneToOne(cascade=CascadeType.ALL)//People是关系的维护端

@JoinTable(name = "people\_address",

joinColumns = @JoinColumn(name="people\_id"),

inverseJoinColumns = @JoinColumn(name = "address\_id"))//通过关联表保存一对一的关系

private Address address;//地址

}

**Address.java**

不变