

Çift Yönlü İlişkiler ve MappedBy



Çift Yönlü İlişkiler ve MappedBy

```
public class Owner {
```

```
    @OneToMany(mappedBy = "owner")
```

```
    private Set<Pet> pets = new HashSet<Pet>();
```

```
}
```

mappedBy attribute çift yönlü ilişkide, ilişkiyi yöneten tarafın kim olduğunu tanımlar

mappedBy attribute olmadığı takdirde aynı FK'yı güncellemeye çalışan **iki farklı SQL ifadesi** söz konusu olur

```
public class Pet {
```

```
    @ManyToOne
```

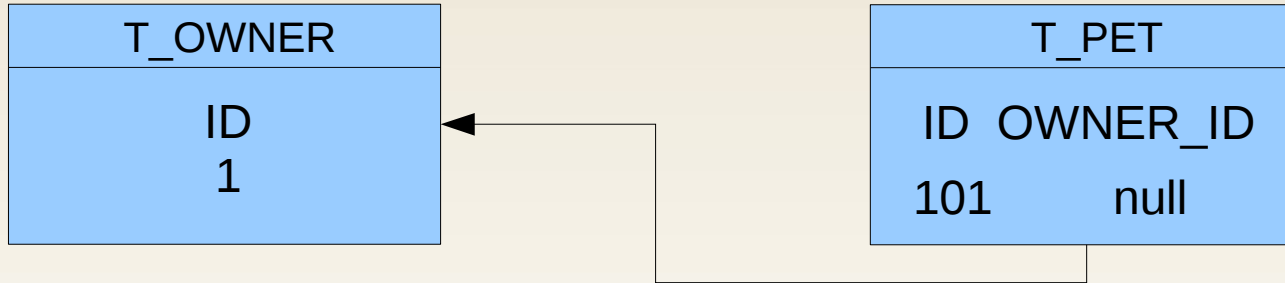
```
    @JoinColumn(name = "OWNER_ID")
```

```
    private Owner owner;
```

```
}
```

JoinColumn ve **JoinTable** anotasyonları mappedBy ile işaret edilen tarafta tanımlanmalıdır

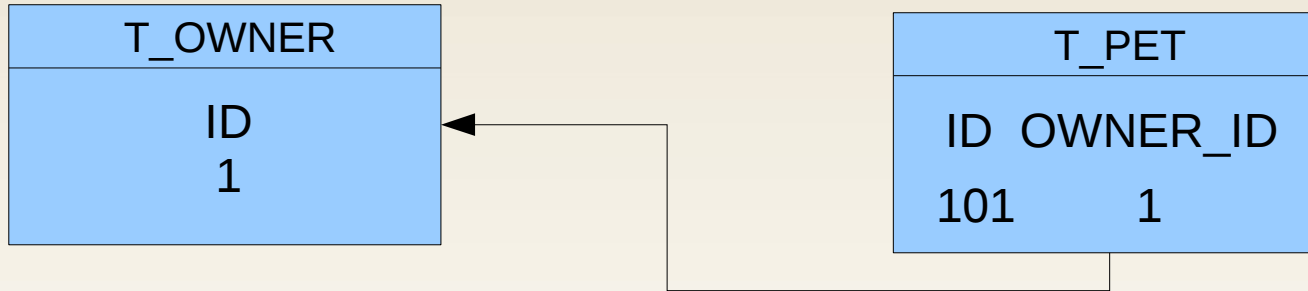
mappedBy'ın İlişki Kurma ve Kaldırmaya Etkisi



```
session.beginTransaction();  
  
Owner owner = session.get(Owner.class, 1L);  
  
Pet pet = session.get(Pet.class, 101L);  
  
owner.getPets().add(pet);  
  
session.getTransaction().commit();
```

İlişki
kurulmaz!

mappedBy'ın İlişki Kurma ve Kaldırmaya Etkisi



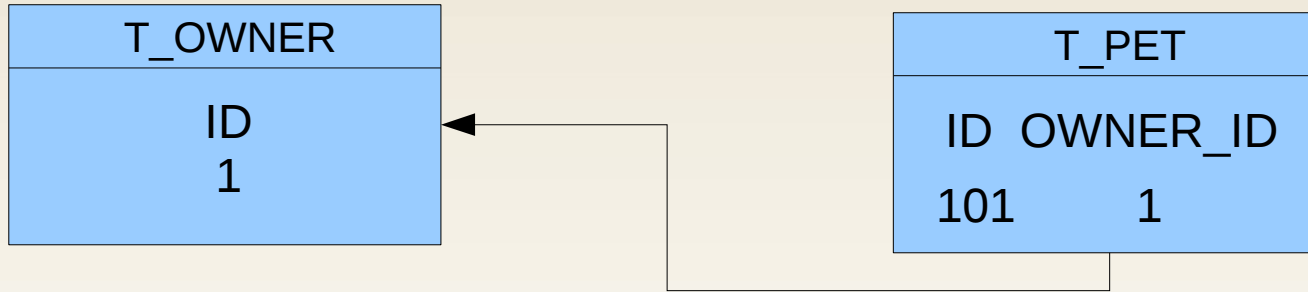
```
session.beginTransaction();  
  
Owner owner = session.get(Owner.class, 1L);  
  
Pet pet = session.get(Pet.class, 101L);
```

```
pet.setOwner(owner);
```

```
session.getTransaction().commit();
```

İlişki
kurulur

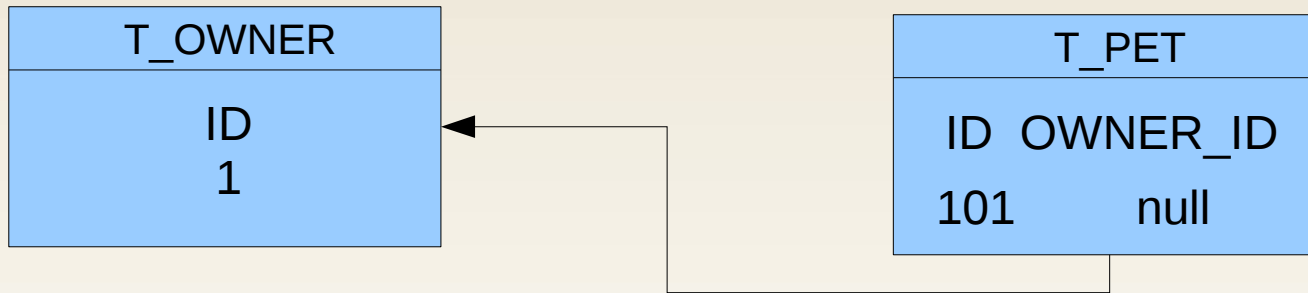
mappedBy'ın İlişki Kurma ve Kaldırmaya Etkisi



```
session.beginTransaction();  
  
Owner owner = session.get(Owner.class, 1L);  
  
Pet pet = session.get(Pet.class, 101L);  
  
owner.getPets().remove(pet);  
  
session.getTransaction().commit();
```

İlişki
kaldırılmaz!

mappedBy'ın İlişki Kurma ve Kaldırmaya Etkisi



```
session.beginTransaction();  
  
Owner owner = session.get(Owner.class, 1L);  
  
Pet pet = session.get(Pet.class, 101L);  
  
pet.setOwner(null);  
  
session.getTransaction().commit();
```

İlişki
kaldırılır

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- info@java-egitimleri.com

