

Bileşenlerin Eşleştirilmesi



Bileşenlerin Eşleştirilmesi

- Bileşenlerin **kendilerine ait bir kimlikleri yoktur**
- Mutlaka **bir entity'ye ait** olmalılar
- Ait oldukları **entity'nin yaşam döngüsüne** tabidirler
- Genellikle **ait oldukları entity'nin** tablosu içerisinde persist edilirler
- Bileşenler **başka bileşenleri içerebilirler** veya **başka entity'lerle referans** verebilirler

Bileşenlerin Eşleştirilmesi

```
@Entity
public class User {
    //...
    @Embedded
    private Address homeAddress;
    //...
}
```

İki annotasyondan birisi
bileşen tanımlamak için
yeterlidir

```
@Embeddable
public class Address {
    @Column(name = "ADDRESS_STREET", nullable = false)
    private String street;
    @Column(name = "ADDRESS_ZIPCODE", nullable = false)
    private String zipcode;
    @Column(name = "ADDRESS_CITY", nullable = false)
    private String city;

    //...
}
```

Bileşenlerin Eşleştirilmesi

- Eğer parent entity sınıf içerisindeki property üzerinde **@Embedded** **mevcut değil ise** ve property tipi JDK tipi değilse,
- Hibernate hedef sınıfın **@Embeddable** annotasyonuna bakar
- Hedef sınıf üzerinde **@Embeddable** **mevcut ise** property yine bileşen olarak eşlenir

Bileşenlerin Eşleştirilmesi

- Bileşenin **bütün property'leri NULL** ise **bileşen değeri NULL** kabul edilir
- Bütün property değerleri NULL olan bir bileşen de persist edilip tekrar yüklenmek istenirse **Hibernate, DB'den NULL dönecektir**

Bileşenler ve Secondary Table

- Bileşenin alanları ait olduğu entity'nin tablosundan **ayrı bir tabloda** da tutulabilir

```
@Entity
```

```
@Table(name = "T_USER")
```

```
@SecondaryTable(name = "T_ADDRESS",  
    pkJoinColumns={@PrimaryKeyJoinColumn(name="USER_ID")})
```

```
public class User {  
    //...  
    private Address homeAddress;  
}
```



Bileşenler ve Secondary Table

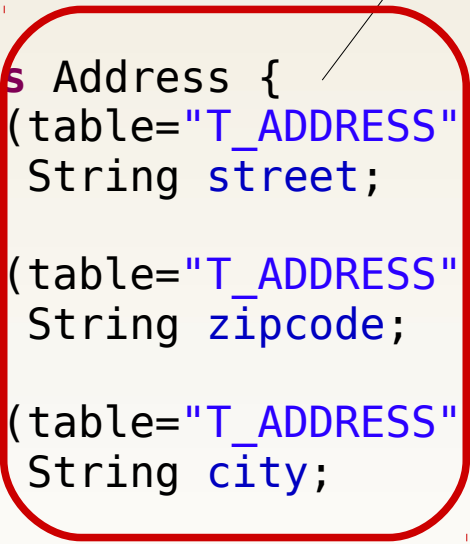
Sütunların hangi tablo içerisinde tanımlı olacakları **table** attribute ile belirtilir

```
@Embeddable
public class Address {
    @Column(table="T_ADDRESS", name="ADDRESS_STREET", nullable=false)
    private String street;

    @Column(table="T_ADDRESS", name="ADDRESS_ZIPCODE", nullable=false)
    private String zipcode;

    @Column(table="T_ADDRESS", name="ADDRESS_CITY", nullable=false)
    private String city;

    //...
}
```



Bileşenlerin Ait Oldukları Entity Nesneye Referans Vermesi

```
@Entity
public class User {
    //...
    private Address homeAddress;
    //...
}
```

```
@Embeddable
public class Address {

    @Parent
    private User user;
    //...
}
```

@Parent anotasyonu ile bileşen içerisinde ait olduğu entity nesneye referans verilebilir.

Böylece @ManyToOne mapping'e gerek kalmaz.

Hibernate'e özel anotasyondur

İletişim

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

