

1:M ve M:N İlişkilerde OrderBy & Sort



JPA ile OrderBy

```
@OneToMany
@JoinColumn(name = "PET_ID")
@OrderBy("filePath asc")
private List<Image> images = new ArrayList<Image>();
```



“**Ordered**” collection veritabanı düzeyinde SQL sorgusundaki “**order by**” ifadesi ile sıralanır

@OrderBy attribute değeri bir **JPQL ifadesidir**.

Hibernate ile OrderBy

```
@OneToMany
@JoinColumn(name = "PET_ID")
@org.hibernate.annotations.OrderBy(
clause="FILE_PATH asc")
private List<Image> images = new ArrayList<Image>();
```



“**Ordered**” collection veritabanı düzeyinde SQL sorgusundaki
“**order by**” ifadesi ile sıralanır

Hibernate'in @OrderBy attribute değeri bir **SQL ifadesidir**,
Burada bir **SQL function da kullanılabilir**

Hibernate Sort

```
@OneToMany
@JoinColumn(name = "PET ID")
@org.hibernate.annotations.SortNatural
private SortedSet<Image> images = new TreeSet<Image>();
```



Collection elemanları Comparable arayüzünü implement ediyorsa
SortNatural kullanılabilir

Sorted* arayüzlerini kullanmak yerine veritabanı düzeyinde "ordering" yapmak daha efektifir

Hibernate Sort

```
@OneToMany
@JoinColumn(name = "PET_ID")
@org.hibernate.annotations.SortComparator(ImageComparator.class)
private SortedSet<Image> images = new TreeSet<Image>();
```



“Sorted” collection Java **Comparator** sınıfı ile hafızada sıralanır

Sorted* arayüzlerini kullanmak yerine veritabanı düzeyinde “ordering” yapmak daha etkilidir

İletişim

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

