

# Spring Data Projesi ve Dinamik Sorgu Metotları



# Dinamik Sorgu Metotları

- Repository nesnelerinin sunacağı sorguların metotları **her bir entity için ayrı bir Repository arayüzünde** tanımlanır
- Spring Data bu arayüzleri implement eden proxy nesneler üreterek **sorgu metotlarını** veri erişim teknolojisine göre **dinamik olarak oluşturur**

```
public interface VetRepository extends JpaRepository<Vet, Long> {
    List<Vet> findByLastname(String lastname);
}
```

# Dinamik Sorgu Metotları

- Repository proxy metotlarından sorguyu üretmek için iki farklı yöntem vardır
  - Sorguyu doğrudan **metot isminden üretmek**
  - Manuel olarak **tanımlanmış sorguyu kullanmak**
- Bu yöntemler ilgili **veri erişim teknolojisine** göre mevcut olabilir veya olmayabilir

# Metot İsminden Dinamik Sorgu Üretmek

- Mekanizma sorgu metodunun isminden **find...By**, **read...By**, **query...By**, **get...By** gibi örnekleri çıkarır
- İlk “**By**” ifadesinden sonraki kısım “**where clause**”unu oluşturur
- “**And**” ve “**Or**” ifadeleri ile ayrılmış bölümler de “**condition**”ları oluşturur

# Metot İsminden Dinamik Sorgu Üretmek

```
public interface VetRepository extends JpaRepository<Vet, Long> {  
  
    List<Vet> findByLastName(String lastName);  
  
    List<Vet> findByFirstNameAndLastName(  
        String firstName, String lastName);  
  
    List<Vet> findByLastNameOrderByFirstNameAsc(String lastName);  
  
    List<Vet> findByLastNameOrderByFirstNameDesc(String lastName);  
  
    List<Vet> findByFirstNameAndLastNameIgnoreCase(  
        String firstName, String lastName);  
}
```

# Metot İsminden Dinamik Sorgu Üretmek

...

```
List<Vet> findByFirstNameAndLastNameAllIgnoreCase(  
    String firstName, String lastName);
```

```
List<Vet> findByGraduationYearGreaterThan(  
    Integer graduationYear);
```

```
List<Vet> findByGraduationYearLessThan(Integer  
graduationYear);
```

```
List<Vet> findByGraduationYearBetween(Integer  
graduationYearFrom, Integer graduationYearTo);
```

```
List<Vet> findFirstOrderByLastNameAsc();
```

...

# Metot İsminden Dinamik Sorgu Üretmek

...

```
List<Vet> findFirst5ByLastName(String lastName);
```

```
List<Vet> findTopByOrderByLastNameAsc();
```

```
List<Vet> findTop5ByLastName(String lastName);
```

```
List<Vet> findFirst5ByLastName(String lastName, Pageable  
pageable);
```

```
List<Vet> findFirst5ByLastName(String lastName, Sort sort);
```

```
}
```

# Property İfadelerinin Çözümlemesi

```
List<Owner> findByAddressZipCode(ZipCode zipCode);
```

Property çözümleme işi domain sınıfından başlar.

1. İlk olarak By ifadesinden sonraki bütün bölüm (AddressZipCode) domain sınıfında aranır.

2. Böyle bir property domain sınıfı içerisinde mevcut değil ise, sağ taraftan başlayarak property ismi “camel case” bölümlerden tokenize edilerek property’lerin altında çözümlenmeye çalışılır: addressZip.code, address.zipCode

```
List<Owner> findByAddress_ZipCode(ZipCode zipCode);
```

“\_” karakteri ile property ifadesinin hangi noktalardan tokenize edileceği belirlenebilir



# Manuel Sorgular

- **@Query** anotasyonu ile repository arayüzünde manuel sorgu da tanımlanabilir

```
public interface OwnerRepository extends JpaRepository<Owner, Long> {

    @Query("select distinct o from Owner o left join o.pets p
           where o.firstName = :firstName and p.name = :petName")
    List<Owner> findOwnersByFirstNameAndPetName(
        @Param("firstName")String firstName,
        @Param("petName")String petName);

    ...
}
```

# Native Sorgular

- Manuel sorgu **native SQL** kullanılarak da oluşturulabilir

```
public interface OwnerRepository extends JpaRepository<Owner, Long> {
    @Query(value="select * from T_OWNERS where FIRST_NAME = :firstName",
            nativeQuery=true)
    List<Owner> findOwnersByFirstName(
        @Param("firstName")String firstName);
    ...
}
```

# Değişiklik Yapan Sorgular

- Bulk update HQL veya native SQL ifadeleri yazarak **entity state'lerini doğrudan DB üzerinde değiştirmek** de mümkündür

```
public interface VetRepository extends JpaRepository<Vet, Long> {
    @Modifying
    @Query("update Vet v set o.enabled = :enabled where
           v.graduationYear > :year")
    int updateOwnersSetEnabled(@Param("enabled") boolean enabled,
                              @Param("year") int year);
}
```

Update işlemi sonucu değişen  
satır sayısını döner

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

