

Optimistic Concurrency Control Kabiliyeti



@Version Alanı Olmadan Optimistic Lock

- Hibernate, ayrı bir versiyon alanı olmadan da optimistic lock yapmaya imkan tanır
- Versiyon alanı olmadan optimistic kilitleme **sadece** “**extended persistence context**” yaklaşımı için uygundur
- **Detached nesnelerle** çalışıldığı vakit “**lost update**” problemi ortaya çıkacaktır

@Version Alanı Olmadan Optimistic Lock

- Versiyon alanı olmadan optimistic lock yapabilmek için entity sınıfta **@OptimisticLocking** anotasyonu ile **OptimisticLockType** belirtilmelidir
- Mevcut OptimisticLockType'lar:
 - **DIRTY**: WHERE ifadesinde sadece değişen alanlar yer alır
 - **ALL**: WHERE ifadesinde bütün alanlar yer alır
 - **NONE**: optimistic kilitlemeyi devre dışı bırakır
 - **VERSION**: @Version alanı kullanılır

@Version Alanı Olmadan Optimistic Lock

```
@Entity
```

```
@org.hibernate.annotations.DynamicUpdate  
@org.hibernate.annotations.OptimisticLocking(  
    OptimisticLockType.DIRTY)  
public class Pet {
```

```
}
```

- **@OptimisticLocking** tipi **DIRTY** tanımlanırsa **@DynamicUpdate** anotasyonu da kullanılmalıdır

Collection İlişkileri ve @OptimisticLock

@Entity

```
public class Owner {
```

```
    @OneToMany
```

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

```
    @org.hibernate.annotations.OptimisticLock(excluded = true)
```

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

```
}
```

Collection'lara yapılan ekleme ve silmelerin **versiyon artışına neden olması istenmiyor** ise ilişki @OptimisticLock ile exclude edilebilir

@Entity

```
public class Pet {
```

```
}
```

İletişim

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

