

Explicit Optimistic ve Pessimistic Lock Yöntemleri



Explicit Lock Yöntemleri

- Version alanına sahip bir entity'nin update işlemi sırasında Hibernate **otomatik olarak versiyon kontrolü** yapar ve versiyonu bir artırır
- Buna **implicit lock** adı verilir
- JPA ve Hibernate **explicit olarak** da optimistic ve pessimistic **lock alınması** için farklı yöntemler sunmaktadır

Persistence Context Lock Metodu ile Explicit Lock

- EntityManager ve Session üzerinde tanımlı **lock() metodu** kullanılarak spesifik bir entity için **explicit lock** alınabilir

JPA API

```
em.getTransaction().begin();  
  
Pet pet = em.find(Pet.class, 1L);  
  
...  
  
em.lock(pet,  
LockModeType.READ);  
  
...  
  
em.getTransaction().commit();
```

Hibernate API

```
Session session = sf.openSession();  
session.beginTransaction();  
  
session.lock(owner,  
LockMode.PESSIMISTIC_WRITE);  
  
//...  
session.getTransaction().commit();
```

Session.lock() ve LockRequest

- Hibernate Session.lock() metodu arka tarafta **LockRequest** üzerinden çalışmaktadır
- Session üzerinden **LockRequest nesnesi** oluşturup bununla da explicit lock yapılabilir

```
Session session = sf.openSession();  
session.beginTransaction();
```

```
LockRequest lockRequest = session.buildLockRequest(  
    LockOptions.UPGRADE);
```

```
lockRequest.lock(owner);
```

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

Get ve Load Sırasında Explicit Lock

- Spesifik bir **entity**'yi **PK değeri ile yüklerken** de explicit biçimde optimistic ve pessimistic lock alınabilir

```
em.getTransaction().begin();
```

```
Pet pet = em.find(Pet.class, 1L,  
                  LockModeType.PESSIMISTIC_FORCE_INCREMENT);
```

```
...
```

```
em.getTransaction().commit();
```

Pet entity'si DB'den select edilirken sorguda WHERE ifadesinden sonra FOR UPDATE alır, böylece pessimistic lock uygulanmış olur

Eğer persistence context'de halihazırdan entity mevcut ise explicit lock çalışmaz

Sorgu Sırasında Explicit Lock

- HQL ve Criteria sorgularında da bir **grup entity üzerinde** explicit lock almak mümkündür

```
em.createQuery("from Owner o left join o.pets")  
  .setLockMode(LockModeType.PESSIMISTIC_FORCE_INCREMENT)  
  .getResultList();
```

DB destekliyorsa HQL'e karşılık gelen SQL sorgusunun WHERE ifadesinin sonunda FOR UPDATE yer alacak, ardından da sorgu sonucu dönen hem Owner hem de Pet entity'lerinin ayrı ayrı versiyon alanları güncellenecektir

Sorgular mutlaka DB'ye gittiği için entity'ler PersistenceContext de olsa bile lock çalışacaktır.

İsimlendirilmiş Sorgular ve Explicit Lock

- İsimlendirilmiş sorgu tanımlarında da explicit lock belirtilebilir

```
@NamedQuery (name="findAllOwnersWithLock", query="from  
Owner", lockMode=LockModeType.PESSIMISTIC_FORCE_INCREMENT)  
@Entity  
public class Owner {  
    ...  
}
```

```
em.createNamedQuery("findAllOwnersWithLock").getResultList();
```

İstenirse named query'de tanımlı lock mode ezilebilir

```
em.createNamedQuery("findAllOwnersWithLock")  
    .setLockMode(LockModeType.NONE).getResultList();
```

NaturalId ile Explicit Lock

- Hibernate'e özel **Naturel PK** üzerinden de explicit olarak **optimistic** veya **pessimistic kilit** elde edilebilir

```
User user = session.bySimpleNaturalId(User.class)
    .with(LockOptions.UPGRADE)
    .load("ksevindik@gmail.com");
```


Refresh İşlemi ve Explicit Lock

- Persistence Context'deki **entity state**'inin **DB'den yeniden yüklenmesi** sırasında da explicit lock yapılabilir

```
em.getTransaction().begin();
```

```
Owner o = em.find(Owner.class, 1L);
```

```
em.refresh(o, LockModeType.PESSIMISTIC_FORCE_INCREMENT);
```

```
em.getTransaction().commit();
```

İletişim

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

