

HQL ile Toplu İşlemler



HQL ile Toplu İşlemler

- Toplu işlemler **doğrudan çalışan DML'lerdir**
- Hibernate ve JPA, **düz SQL'den biraz daha kabiliyetli DML** işlem desteği sunarlar
- Toplu işlemlerde **persistence context ve cache bypass** edilir
- HQL ifadeleri, entity ve property isimlerini kullanır
- HQL ifadesi sonucu güncelleme yapılan, yaratılan veya silinen “**nesne**” **sayısı** dönülür

HQL ile Toplu İşlemler

```
Query query = session.createQuery(
    "update Visit v set v.checkup = ?");
query.setParameter(0, true);

int updatedVisitCount = query.executeUpdate();
```

```
Query query = session.createQuery(
    "delete Pet p where p.owner is null");

int deletedPetCount = query.executeUpdate();
```

HQL ile Toplu İşlemler

- Doğrudan çalıştırılan DML işlemleri **default olarak** version/timestamp değerlerine dokunmazlar
- Ancak HQL kullanılarak doğrudan güncellenen nesnelerin **version değerleri artırılabilir**

```
Query query = session.createQuery(
    "update versioned Visit v set v.checkup = ?");
query.setParameter(0, true);

int updatedVisitCount = query.executeUpdate();
```

HQL ile Toplu İşlemler

```
Query query = session.createQuery(  
    "insert into CreditCard(ccNo,exprDate) select  
    ccNo,exprDate from Customer where type =  
    'Visa'");  
  
int createdCreditCards = query.executeUpdate();
```

- SELECT ile dönen tiplerin INSERT ifadesinin beklediği tiplerle **eşleşmesi** gerekir
- Her insert edilen entity için **otomatik olarak DB identifier üretilecektir**
- Üretilen nesneler **versioned** ise otomatik olarak version değeri atanır

HQL ile Toplu İşlemler

- HQL ile gerçekleştirilen toplu işlemlerde entity mapping'lerdeki **cascade** tanımları **dikkate alınmaz!**

İletişim

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

