Hibernate ve Transaction Yönetimi

harezmi bilişim çözümleri

Hibernate ve Transaction Yönetimi

JAVA Eğitimleri

- Transaction bir veya daha fazla SQL işleminin **tek bir atomik unit** olarak ele alınmasını sağlar
- Transaction **demarcation**, TX'in başlatılması ve bitirilmesi arasında kalan kısımdır
  - TX **begin** ile başlatılır, **commit/rollback** ile sonlandırılır
- Hibernate ve JPA'da bütün SQL işlemleri **mutlaka bir transaction** içerisinde gerçekleştirilmelidir
- Transaction dışında bir SQL ifadesini çalıştırmak **mümkün değildir**

Hibernate ve Transaction Yönetimi

```
Session session = null;
Transaction transaction = null;
try {
    session = sessionFactory.openSession();
    transaction = session.beginTransaction();

    Pet kedi = new Pet();
    kedi.setName("boncuk");

    session.save(kedi);

    transaction.commit();
} catch (Exception ex) {
    transaction.rollback();
} finally {
    session.close();
}
```

Persistence işlemi mutlaka açık bir TX bloğu içerisinde gerçekleştirilmelidir

Aksi durumda Hibernate işlemi göz ardı eder

Hibernate 5, JPA uyumluluğu nedeni ile hata fırlatmaya başlamıştır

Hibernate Exception'ları

- PersistenceContext üzerinde gerçekleştirilen işlemler sırasında hata meydana geldiğinde **HibernateException** ve türevleri fırlatılır
- HibernateException, JPA'nın **PersistenceException** sınıfından türer
- NoResultException, NonUniqueResultException, LockTimeoutException, QueryTimeoutException dışındaki **bütün exception'lar TX rollback nedeni**dir

Hibernate Exception'ları

- Exception meydana geldiği vakit **Session o hali ile terk edilip, TX rollback yapılmalı**dır
- TX rollback yapılması allocate edilmiş **DB resource'larının beklemeden salıverilmesi** için önemlidir

İletişim

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