

Hibernate Criteria API ile Sorgulama



Hibernate Criteria API

- Criteria API **programatik** olarak sorgu hazırlamak için tasarlanmıştır
- Genellikle **parametrik sorgu ekranlarında** kullanılır
- Kullanıcının seçtiği kriterlere göre **sorgu dinamik olarak** oluşturulur
- Hibernate 5.2 ile birlikte Criteria API deprecated olup, yerini ileride JPA Criteria API'ye bırakacaktır

Hibernate Criteria API

Öncelikle sorgu sonucu dönülecek root entity sınıfı ile Session üzerinden bir Criteria nesnesi yaratılır

```
Criteria criteria = session.createCriteria(Owner.class);
```

```
Criterion emailCriterion = Restrictions.eq("email",  
"ksevindik@gmail.com");  
criteria.add(emailCriterion);
```

Criterion nesneleri ile sorgu üzerindeki kısıtlar tanımlanır
Restrictions sınıfı built-in criterion nesneleri içerir

```
List owners = criteria.list();
```

Eğer tek bir entity döneceği biliniyor ise **Criteria.uniqueResult()** metodu da kullanılabilir

DetachedCriteria

```
DetachedCriteria detachedCriteria =  
DetachedCriteria.forClass(Owner.class);
```

Session'dan kopuk
Criteria sorgusudur

Serializable'dır
Herhangi bir yerde
saklanabilir

```
Criterion emailCriterion = Restrictions.eq("email",  
"ksevindik@gmail.com");
```

```
detachedCriteria.add(emailCriterion);
```

```
Criteria criteria =  
detachedCriteria.getExecutableCriteria(session)
```

İstendiğinde session'a attach edilerek
normal executable Criteria nesnesi oluşturularak
sorgu gerçekleştirilebilir

```
List result = criteria.list();
```

Criteria Sorgularına Örnekler

```
session.createCriteria(Owner.class)
.add(Restrictions.eq("firstName", "James"))
.add(Restrictions.eq("address.street", "1st Street"))
.addOrder(Order.asc("lastName"))
.addOrder(Order.asc("firstName"))
.list();
```

↓
Sorgu sonuçlarının sıralı gelmesi birden fazla field üzerinden sağlanabilir

↘ Birden fazla Criterion eklenebilir

Eklenen criterion'lar where clause'una AND'lenerek dahil edilmektedir

Criteria Sorgularına Örnekler

```
Criterion restriction = Restrictions.between("price",
    new BigDecimal(100), new BigDecimal(200) );
```

```
session.createCriteria(Item.class).add(restriction);
```

```
String[] emails = {"ksevindik@gail.com", "info@java-egitimleri.com"};
session.createCriteria(Owner.class).add(Restrictions.in("email", emails) );
```

Criteria Sorgularına Örnekler

```
session.createCriteria(Owner.class)  
    .add( Restrictions.isNull("email") );
```

```
session.createCriteria(Pet.class)  
    .add( Restrictions.isEmpty("visits") );
```

```
session.createCriteria(Pet.class)  
    .add( Restrictions.sizeGt("visits", 3) );
```

Criteria Sorgularına Örnekler

```
session.createCriteria(Owner.class).add( Restrictions
.eqProperty("firstName", "username") );
```

```
session.createCriteria(Owner.class).add( Restrictions
.like("username", "G%") );
```

```
session.createCriteria(Owner.class).add( Restrictions
.like("username", "G", MatchMode.START) );
```

```
session.createCriteria(Owner.class)
.add( Restrictions.eq("username", "foo").ignoreCase() );
```


Criteria Sorgularına Örnekler

```
session.createCriteria(Owner.class)
    .add( Restrictions.like("firstName", "G%") )
    .add( Restrictions.like("lastName", "K%") );
```

```
session.createCriteria(Owner.class)
    .add( Restrictions.or(
        Restrictions.and(
            Restrictions.like("firstName", "G%"),
            Restrictions.like("lastName", "K%")),
        Restrictions.in("email", emails)));
```

Criteria Sorgularına Örnekler

```
session.createCriteria(User.class)
    .add( Restrictions.sqlRestriction(
        "length({alias}.PASSWORD) < ?",
        5,Hibernate.INTEGER));
```

```
session.createCriteria(Category.class)
    .add( Restrictions.sqlRestriction(
        "'100' > all" +
        " (select i.PRICE from ITEM i" +
        " where i.CATEGORY_ID = {alias}.ID )"));
```

Criteria ve Join : 1. Yol

```
Criteria petCriteria =  
session.createCriteria(Pet.class);
```

```
petCriteria.add(Restrictions.like("name",  
    "Kitty", MatchMode.ANYWHERE));
```

Root entity'nin Criteria nesnesi üzerinden
join yapılacak property için ayrı bir
Criteria nesnesi yaratılarak yapılır

```
Criteria visitCriteria = petCriteria  
    .createCriteria("visits");  
visitCriteria.add(Restrictions.gt("date", new Date()));
```

```
List result = petCriteria.list();
```

Criteria ve Join : 2. Yol

```
session.createCriteria(Pet.class)
    .createAlias("visits", "v")
    .add(Restrictions.like("name", "%Kitty%"))
    .add(Restrictions.eq("v.date", new
Date())));
```

Diğer bir yol ise join yapılacak property için alias tanımlamaktır

- Criteria API'de **tek değerli ilişkinin dot notasyonu ile implicit join yapılması** mümkün değildir

```
Restrictions.like("p.owner.firstName", "%Jo%")
```

Hata!

Criteria ve ResultTransformer

```
List result = session.createCriteria(Pet.class)
                    .createAlias("visits", "v")
                    .setResultTransformer(Criteria.ROOT_ENTITY).list();
```

ResultTransformer ile sorgu sonucunda
sadece root entity'den nesnelerin olması
sağlanabilir

```
List result = session.createCriteria(Pet.class)
                    .createAlias("visits", "v")
                    .setResultTransformer(Criteria.DISTINCT_ROOT_ENTITY)
                    .list();
```

Criteria sorguları da duplike sonuçlar dönebilir
Duplikasyon LinkedHashSet'e konarak veya
ResultTransformer ile elemine edilebilir

Criteria ve ResultTransformer

```
Criteria crit =
    session.createCriteria(Pet.class)
        .createAlias("visits", "v")
        .createAlias("owner", "o")
        .setResultTransformer(Criteria.ALIAS_TO_ENTITY_MAP);

for (Object result : crit.list()) {

    Map map = (Map) result;
    Pet pet = (Pet) map.get(Criteria.ROOT_ALIAS);
    Visit visit = (Visit) map.get("v");
    Owner owner = (Owner) map.get("o");

}
```

Projeksiyon ve Rapor Sorguları

```
session.createCriteria(Visit.class).setProjection(  
Projections.projectionList()  
    .add(Projections.id())  
    .add(Projections.property("description"))  
    .add(Projections.property("date")));
```

```
session.createCriteria(Visit.class).setProjection(  
Projections.projectionList()  
    .add(Property.forName("id"))  
    .add(Property.forName("description"))  
    .add(Property.forName("date")));
```

Projeksiyon ve Rapor Sorguları

```
session.createCriteria(Visit.class)
    .setProjection(Projections.projectionList()
        .add(Projections.id()
            .as("visitId"))
        .add(Projections.property("description")
            .as("visitDescription"))
        .add(Projections.property("date")
            .as("visitDate")))
    .setResultTransformer(
        new AliasToBeanResultTransformer(VisitDTO.class));
```


Projeksiyon ve Rapor Sorguları

```
session.createCriteria(Pet.class)
    .createAlias("owner", "o")
    .setProjection( Projections.projectionList()
        .add( Projections.groupProperty("o.id") )
        .add( Projections.groupProperty("o.lastName") )
        .add( Projections.count("id") )
        .add( Projections.avg("age") ) );
```

Criteria ve Alt Sorgular

```
DetachedCriteria subquery =  
    DetachedCriteria.forClass(Pet.class, "p");
```

DetachedCriteria alt sorgu oluşturmak için de kullanılabilir

```
subquery  
    .add(Restrictions.eqProperty("p.owner.id", "o.id"))  
    .add(Restrictions.isNull("p.name"))  
    .setProjection(  
        Property.forName("p.id").count());
```

```
Criteria crt = session.createCriteria(Owner.class, "o")  
    .add(Subqueries.lt(10L, subquery));  
crt.list();
```

İletişim

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

