Optimizasyon Yöntemleri (Sayfalama/Paging)

harezmi bilişim çözümleri

Sorgu Sonuçlarının Sayfalanması (Paging)

- Sorgu sonuçları tek seferde değil de **parça parça** yüklenebilir
- HQL ve Criteria API'si destekler
- Query.**setFirstResult()** ve Query.**setMaxResult()** metotları ile gerçekleştirilir
- Hibernate arka tarafta **LIMIT** ve **OFFSET** SQL ifadeleri ile sayfalamayı gerçekleştirmektedir

Sorgu Sonuçlarının Sayfalanması (Paging)

- Sayfalamaya öncelikle sorgu sonucunda ekranda gösterilecek **toplam kayıt sayısı**nı tespit ederek başlanır

```
Long entityCount = session.createQuery("select
count(p.id) from Pet p").getSingleResult();
```

- Ardından bulunan sonuç her sayfada gösterilecek kayıt sayısına bölünerek **sayfa sayısı** tespit edilir

```
Long pageSize = 10;
Long pageCount = entityCount/pageSize;
pageCount += entityCount % pageSize != 0?1:0;
```

Sorgu Sonuçlarının Sayfalanması (Paging)

- Artık **setFirstResult** ve **setMaxResults** değerleri kullanılarak herhangi bir sayfaya karşılık gelen kayıtlar sorgu ile elde edilebilir ve sayfada gösterilebilir

```
Query query = session.createQuery(
                        "from Pet p order by p.id asc");

query.setFirstResult(0);
query.setMaxResults(10);

query.list();
```

Hangi kayıttan gösterime devam edileceği belirtilir

Her sayfada kaç kayıt olacağı (pageSize) belirtilir

Sorgu Sonuçlarının Sayfalanması (Paging)

- Her sayfa için aynı sorgu tekrar ama **farklı setFirstResult** değeri ile çalıştırılacaktır

```
Query query = session.createQuery(
                    "from Pet p order by p.id asc");

query.setFirstResult(10);
query.setMaxResults(10);
query.list();
```

- Sorgu sonuçlarının her seferinde aynı ID veya unique bir alan üzerinden **aynı şekilde sıralı olması önemli**dir!

Sorgu Sonuçlarının Sayfalanması (Paging)

```
Query query = session.createQuery(
    "from Pet p order by p.id asc");

query.setFirstResult(20);
query.setMaxResults(10);
query.list();
```

```
select
    pet0_.id as id1_1_,
    pet0_.created_by as created_2_1_,
    pet0_.created_date as created_3_1_,
    pet0_.updated_by as updated_4_1_,
    pet0_.updated_date as updated_5_1_,
    pet0_.birth_date as birth_da6_1_,
    pet0_.name as name7_1_,
    pet0_.owner_id as owner_id8_1_
from
    t_pet pet0_
order by
    pet0_.id asc limit ? offset ?
```

Fetch Join ve Paging

- Limit/offset, **fetch join uygulanan sorgular**da **ilişkilerin eksik gelmesi**ne neden olur
- Bu nedenle fetch join içeren sorgularda **paging disable** edilir, sorgu sonucu **bütün kayıtlar** dönülür
- Paging, sorgu çalıştıktan sonra **hafıza** da yapılır!

```
Query query = session.createQuery("from Pet p left join
fetch p.visits order by p.id asc");
query.setFirstResult(0);
query.setMaxResults(10);
query.list();
```

Yukarıdaki sorguyu çalıştırdığımız vakit Console'a şöyle bir **uyarı mesajı** görülecektir:
HHH000104: firstResult/maxResults specified with collection fetch; applying in memory!

İletişim

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