

## Criteria Query

Queries are now deprecated. Criteria queries are faster than normal Query object.

HCQL---- hibernate Criteria Query

//equivalent to select \* from Product

```
Criteria cr=session.createCriteria(Product.class)
```

```
List<Product> plist=Cr.list()
```

//to find product with name=chair

```
Criteria cr=session.createCriteria(Product.class);
```

```
Cr.add(Restrictions.eq("name","chair"))
```

//to find product with name!=chair

```
Criteria cr=session.createCriteria(Product.class);
```

```
Cr.add(Restrictions.ne("name","chair"))
```

//to find product with name starts with ch

```
Criteria cr=session.createCriteria(Product.class);
```

```
Cr.add(Restrictions.like("name","ch%",MatchMode.AYWHERE))
```

Matchmode.START --- search pattern only at the beginning

Matchmode.END --- search pattern only at the end

Matchmode.ANYWHERE --- search pattern anywhere in the string

Matchmode.EXACT --- search pattern for exact match

In Restrictions you can use functions like

1. isNull
2. isNotNull
3. eq
4. ne
5. like
6. gt
7. lt

Find all products with price <1000 or

Qty >100

```
Criteria cr=session.createCriteria(Product.class);
```

```
Criterion pricelt=Restrictions.lt("price",1000);
```

```
Criterion qtygt=Restrictions.gt("qty",100);
```

```
LogicalExpression orexp=Restrictions.or(pricelt, qtygt)
```

```
Cr.add(orexp)
```

```
List<Product> plist=Cr.list();
```

Pagination means, at a time show only one screenful records.

```
Criteria cr=session.createCriteria(Product.class);
```

```
Cr.setFirstResult(11)
```

```
Cr.setMaxResult(20)
```

```
List<Product> Cr.list();
```

To retrieve only first record.

```
Criteria cr=session.createCriteria(Product.class);
```

```
cr.add(Restrictions.gt("qty",100))
```

```
Cr.setMaxResult(1)
```

```
List<Product> Cr.uniqueResult()
```

//to display data in descending order of price for all products with qty>100

```
Criteria cr=session.createCriteria(Product.class);
```

```
cr.add(Restrictions.gt("qty",100))
```

```
cr.addOrder(Order.desc("price"))
```

//to create joins between 2 tables

```
Criteria cr=session.createCriteria(Product.class);
```

```
Criteria prodcatt=cr.createCriteria(Category.class)
```

```
Prodcart.add(Restrictions.gt("price",1000));
```

```
List result=cr.list()
```

JPA criteria Query

```
CriteriaBuilder cb=session.getCriteriaBuilder();
```

```
CriteriaQuery<Product> cr=cb.createQuery(Product.class);
```

```
Root<Product> root=cr.from(Product.class)
```

```
Cr.select(root)
```

```
Cr.select(root).where(cb.gt("price"),1000.00)
```

```
Cr.select(root).where(cb.between("price"),500,1000.00)
```