

//create object of criteria first

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

// To get all records (at the end of the below codes)

```
List results = cr.list();
```

// To get all records equal to

```
cr.add(Restrictions.eq("salary", 2000));
```

```
List results = cr.list();
```

// To get all records

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

// To get records having salary more than 2000

```
cr.add(Restrictions.gt("salary", 2000));
```

// To get records having salary less than 2000

```
cr.add(Restrictions.lt("salary", 2000));
```

// To get records having firstName starting with zara

```
cr.add(Restrictions.like("firstName", "zara%"));
```

// Case sensitive form of the above restriction.

```
cr.add(Restrictions.ilike("firstName", "zara%"));
```

// To get records having salary in between 1000 and 2000

```
cr.add(Restrictions.between("salary", 1000, 2000));
```

// To check if the given property is null

```
cr.add(Restrictions.isNull("salary"));
```

// To check if the given property is not null

```
cr.add(Restrictions.isNotNull("salary"));
```

// To check if the given property is empty

```
cr.add(Restrictions.isEmpty("salary"));
```

// To check if the given property is not empty

```
cr.add(Restrictions.isNotEmpty("salary"));
```

// To get records matching with OR conditions
// To get records matching with AND conditions

((A='X') and (B in('X',Y))) or ((A='Y') and (B='Z'))

Below is for the above condition

```
Criteria criteria = getSession().createCriteria(clazz);
```

```
Criterion rest1= Restrictions.and(Restrictions.eq("A", "X"),  
    Restrictions.in("B", Arrays.asList("X","Y")) );
```

```
Criterion rest2= Restrictions.and(Restrictions.eq("A", "Y"),  
    Restrictions.eq("B", "Z")) ;
```

```
criteria.add(Restrictions.or(rest1, rest2));
```

// To get records from 1st to 10th row

```
cr.setFirstResult(1);  
cr.setMaxResults(10);  
List results = cr.list();
```

// To sort records in descening order

```
cr.addOrder(Order.desc("salary"));
```

// To sort records in ascending order

```
cr.addOrder(Order.asc("salary"));
```

// To get total row count.

```
cr.setProjection(Projections.rowCount());
```

// To get average of a property.

```
cr.setProjection(Projections.avg("salary"));
```

// To get distinct count of a property.

```
cr.setProjection(Projections.countDistinct("firstName"));
```

// To get maximum of a property.

```
cr.setProjection(Projections.max("salary"));
```

```
// To get minimum of a property.  
cr.setProjection(Projections.min("salary"));
```

```
// To get sum of a property.  
cr.setProjection(Projections.sum("salary"));
```