



# Hibernate-Named Queries-Criteria-QBE

Presented by

# Named Queries

A named query is a statically defined query with a predefined unchangeable query string.

They're validated when the session factory is created, thus making the application to fail fast in case of an error.

Ex: **@NamedQueries( {**  
    **@NamedQuery(**  
        **name = "findempbyname",**  
        **query = "from Employee e where e.empName = :empName"**  
    **)**

**\*\* Demo**



# Criteria API

Criteria is a simplified API for retrieving entities by composing Criterion objects.

This is a very convenient approach for functionality like "search" screens where there is a variable number of conditions to be placed upon the result set.

Ex:

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

```
// Add restriction
```

```
cr.add(Restrictions.gt("salary", 70000));
```

```
List<Employee> totalSalary = cr.list();
```

```
//Add Projections
```

```
// To get total salary.
```

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

```
List<Employee> totalSalary = cr.list();
```

**\*\*Demo**



# QBE – Query By Example

QBE returns a result set depending on the properties that were set on an instance of the queried class.

Ex:

If an Employee entity is created and fill in the name field, then the query will select all the Employee entities having the same name field as the given Employee entity.

```
/*----- QBE -----*/
```

```
Criteria crit = session.createCriteria(Employee.class);  
Employee employee = new Employee();  
employee.setEmpName("Amar");  
crit.add(Example.create(employee));  
List<Employee> empNameList = crit.list();
```



# QBE Example

```
/*----- QBE -----*/  
Criteria crit = session.createCriteria(Employee.class);  
Employee employee = new Employee();  
employee.setEmpName("Amar");  
crit.add( Example.create(employee) );  
List<Employee> empNameList = crit.list();
```



# Thank You

