**JOIN:**

@OneToMany(fetch=FetchType.***LAZY***, cascade = {CascadeType.***PERSIST***,CascadeType.***MERGE***})

@Fetch(FetchMode.***JOIN***)

Join is only one query fetch,and it causes EAGER fetch of the child collection, a good practice of using it is to initialize it as LAZY.

EX: Query query = entityManager.createQuery("SELECT DISTINCT u FROM User AS u JOIN FETCH u.boughtItems");

**return** (List<User>) query.getResultList();

**SUBSELECT:**

SubSelect does two fetches, one for the all parents(roots) and another one for all the children(associated relations).

@OneToMany(fetch=FetchType.***LAZY***, cascade = CascadeType.***PERSIST***,CascadeType.***MERGE***})

@Fetch(FetchMode.***SUBSELECT***)

We have also to “Hydrate” collection

List<User> users = **this**.findAll();

users.get(0).getBoughtItems().get(0);

**Batch:**

The number of fetches will depend on how the batch is declared

@OneToMany(fetch=FetchType.***LAZY***, cascade = { CascadeType.***PERSIST***,CascadeType.***MERGE***})

@org.hibernate.annotations.Fetch(org.hibernate.annotations.FetchMode.***SELECT***)

@org.hibernate.annotations.BatchSize(size = 2) // batch

Fetch= members%batch/2

For example If member = 3; BatchSize = 2; 2 Collection fetches.

We have also to “Hydrate” collection