Batch

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

@Fetch(FetchMode.***SELECT***)

@BatchSize(size = 2)

Sub Select:

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

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

Join:

Is not recommended using plain join, since Join is cartesian products, need to watch collection size. Best practice is using Lazy initialization.

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

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

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

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

**FetchMode SELECT**  
Use this when you want a quick response time when working on a single entity. SELECT creates small queries and only fetches the data which is absolutely needed. The use-case in our example could be an application to which displays one Customer with its Invoices.

**BatchSize**  
BatchSize is useful when working with a fixed set of data. When you have a batch processing 10 Customers at a time a BatchSize of 10 will drastically reduced the number of queries needed.If the BatchSize is not set too high the query will most likely return a manageable amount of data in a reasonable time.

**FetchMode JOIN**  
As indicated you’ll have to worry about duplicated results. On the other hand JOIN creates the least amount of queries. In a high latency environment a single JOIN could be considerable faster then multiple SELECTS. Keep in mind that joining too much data could put a strain on the database.

**FetchMode SUBSELECT**  
If you’ve got an entity of which you know that there aren’t that many of them, and almost all of them are in the session, then SUBSELECT should be a good choice. Just keep in mind that all collections are fetched, even if the parent is not in the session. A SUBSELECT when having a single Customer in session while there are 1000+ in the database will be wasteful.