



CS544 EA

Hibernate

Optimization: Join Fetch

# Join Fetch

- Before Fetch Graphs were added to JPA
  - Queries could already do “Join Fetch”
- Like EntityGraph Join Fetch-ed entities are:
  - **Added to the cache**
  - **Not added to the result set**
- Unlike EntityGraph
  - Join Fetch can use inner or outer join (just add left / outer)

# Join Fetch

No SELECT clause needed  
joined entities not added to ResultSet  
Although you may want it for DISTINCT

```
TypedQuery<Customer> query = em.createQuery(  
    "SELECT DISTINCT c from Customer c "  
    + "JOIN FETCH c.address a "  
    + "JOIN FETCH c.books b "  
    + "JOIN FETCH b.author "  
    + "WHERE c.firstName like :name", Customer.class);  
query.setParameter("name", "J%");  
List<Customer> customers = query.getResultList();  
System.out.println(customers.size());
```

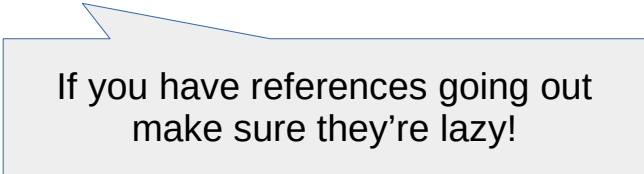
Remember: don't join (fetch)  
multiple collections!

Hibernate:

```
select  
    customer0_.id as id1_3_0_,  
    address1_.id as id1_0_1_,  
    books2_.id as id1_2_2_,  
    author3_.id as id1_1_3_,  
    customer0_.address_id as address_4_3_0_,  
    customer0_.firstName as firstNam2_3_0_,  
    customer0_.lastName as lastName3_3_0_,  
    address1_.city as city2_0_1_,  
    address1_.state as state3_0_1_,  
    books2_.author_id as author_i3_2_2_,  
    books2_.name as name2_2_2_,  
    books2_.books_id as books_id4_2_0_,  
    books2_.id as id1_2_0_,  
    author3_.name as name2_1_3_  
from  
    Customer customer0_  
inner join  
    Address address1_  
        on customer0_.address_id=address1_.id  
inner join  
    Book books2_  
        on customer0_.id=books2_.books_id  
inner join  
    Author author3_  
        on books2_.author_id=author3_.id  
where  
    customer0_.firstName like ?
```

# Join Fetch and N+1

- Join Fetch can be a solution for N+1
  - Load all the needed objects in one query
- Same potential problems:
  - You can not Join Fetch more than one collection eager
  - Eager associations from your graph / result to other entities **still cause N+1** (see eager references N+1)



If you have references going out  
make sure they're lazy!