**JPA Queries VS Custome Query**

@Modifying -> only returns int/void type

If we want to change data in db -> @Transactional -> update-delete-create

Where to put @Transactional?

@Query("select c from Person c where c.id = ?1")

public Person findById(int id);

Hibernate: select person0\_.id as id1\_1\_, person0\_.laptop\_lid as laptop\_l4\_1\_, person0\_.name as name2\_1\_, person0\_.surname as surname3\_1\_ from person person0\_ where person0\_.id=?

Jpa default findById(int id)

Hibernate: select person0\_.id as id1\_1\_0\_, person0\_.laptop\_lid as laptop\_l4\_1\_0\_, person0\_.name as name2\_1\_0\_, person0\_.surname as surname3\_1\_0\_, laptop1\_.lid as lid1\_0\_1\_, laptop1\_.lname as lname2\_0\_1\_ from person person0\_ left outer join laptop laptop1\_ on person0\_.laptop\_lid=laptop1\_.lid where person0\_.id=?

@Query("select c from Person c")

publicIterable<Person> find();

Hibernate: select person0\_.id as id1\_1\_, person0\_.laptop\_lid as laptop\_l4\_1\_, person0\_.name as name2\_1\_, person0\_.surname as surname3\_1\_ from person person0\_

Jpa default findAll()

Hibernate: select person0\_.id as id1\_1\_, person0\_.laptop\_lid as laptop\_l4\_1\_, person0\_.name as name2\_1\_, person0\_.surname as surname3\_1\_ from person person0\_

@Modifying -> only returns int/void type

@Query("UPDATE Person c SET c.name = ?1 WHERE c.id = 1")

public void update(String name);

Hibernate: update person set name=? where id=1

Jpa default save(Person person)

Hibernate: select person0\_.id as id1\_1\_0\_, person0\_.laptop\_lid as laptop\_l4\_1\_0\_, person0\_.name as name2\_1\_0\_, person0\_.surname as surname3\_1\_0\_ from person person0\_ where person0\_.id=?

Hibernate: update person set laptop\_lid=?, name=?, surname=? where id=?

@Modifying

@Query("delete from Person c where c.id = ?1")

public void delete(int id);

Hibernate: delete from person where id=?

Jpa default deleteById(int id)

Hibernate: select person0\_.id as id1\_1\_0\_, person0\_.laptop\_lid as laptop\_l4\_1\_0\_, person0\_.name as name2\_1\_0\_, person0\_.surname as surname3\_1\_0\_, laptop1\_.lid as lid1\_0\_1\_, laptop1\_.lname as lname2\_0\_1\_ from person person0\_ left outer join laptop laptop1\_ on person0\_.laptop\_lid=laptop1\_.lid where person0\_.id=?

Hibernate: delete from person where id=?

**Lazy Eager loading**

Eager ->

Hibernate: select person0\_.id as id1\_1\_0\_, person0\_.name as name2\_1\_0\_, person0\_.surname as surname3\_1\_0\_ from person person0\_ where person0\_.id=?

Hibernate: select laptops0\_.person\_id as person\_i3\_0\_0\_, laptops0\_.lid as lid1\_0\_0\_, laptops0\_.lid as lid1\_0\_1\_, laptops0\_.lname as lname2\_0\_1\_, laptops0\_.person\_id as person\_i3\_0\_1\_ from laptop laptops0\_ where laptops0\_.person\_id=?

Lazy ->

Hibernate: select person0\_.id as id1\_1\_0\_, person0\_.name as name2\_1\_0\_, person0\_.surname as surname3\_1\_0\_ from person person0\_ where person0\_.id=?

**Optional**

@OneToOne(Optional = false) -> makes the association mandatory

The target entity can not be persisted when its association reference is null

**CascadeType.Removal VS OrphanRemoval**

cascadeType.Removal -> if you remove the parent class , child class will be removed

OrphanRemoval -> if we remove child entity the parent entity will be removed