

⇒ Query Methods.

↳ HQL & Native queries.

⇒ Representing cardinalities

↳ Mapped By
↳ Cascading

⇒ Fetch Types

↳ Eager & Lazy.

Query Methods.

1) Declared Queries. ✓

2) HQL queries

3) Native queries.

HQL.

⇒ Hibernate Query Language

⇒ Middle of SQL & OOPs.

↳ We can use class name instead of table name.

→ Attrs of class instead of columns.

⚡ → Called repository method.
productRepo.doSomething()
↳ HQL

↓
JPA checks the HQL
query & creates the corresponding
SQL & execute

Representing Cardinalities

1:1 ⇒ @OneToOne

1:M ⇒ @OneToMany

M:1 ⇒ @ManyToOne

M:M ⇒ @ManyToMany

Cardinality ⇒

When an object of 1 class has an attribute
of another class.

Class Product {

id

title

==

@ManyToOne

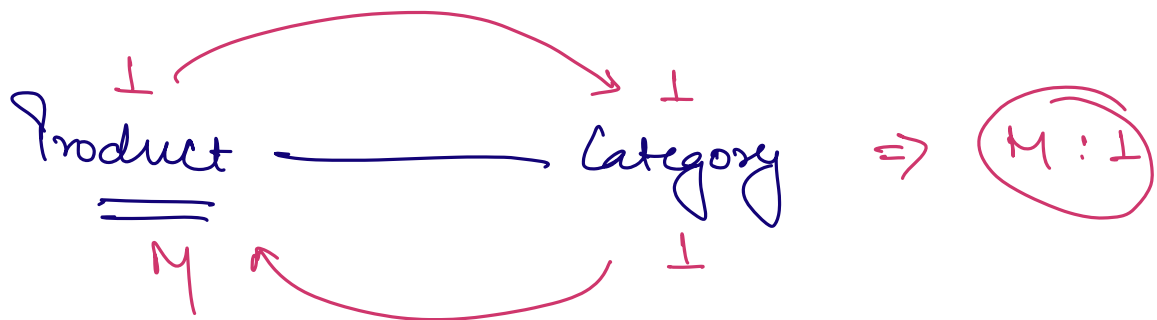
Category category;

==

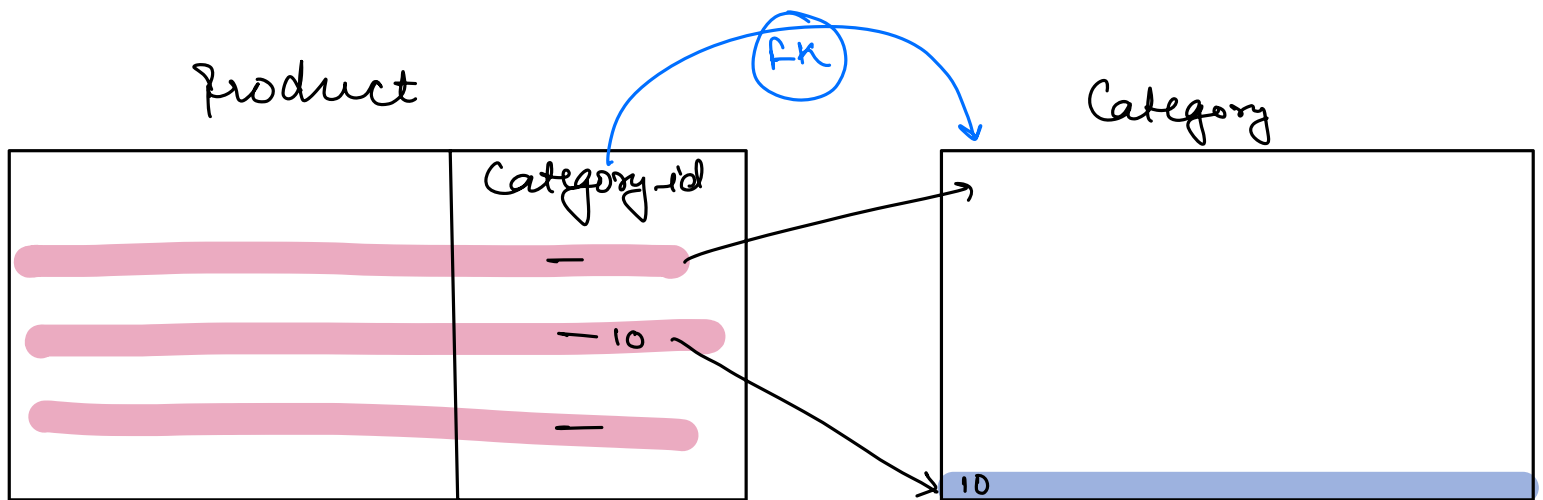
Category {

id

==



⇒ Category-id in Product table.



⇒ Category-id is acting like a fk in Product table.

If we delete category?

- 1) Update the category-id of products to be NULL
- 2) Delete the product as well.
- 3) Don't allow the category to be deleted if there's an associated product.

⇒ Foreign Key Constraints (Cascading).

Fetch Types.

Class Product {

Category category

Eager



3

Product p = productRepository.findById(-);

Join Query was executed
to fetch the underlying
Category for the Product.

Category {

List<Products>

← lazy

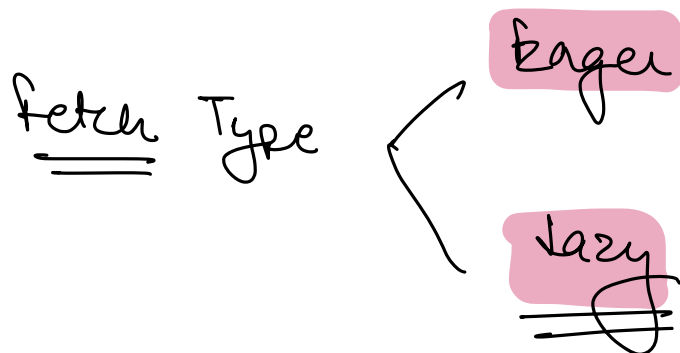
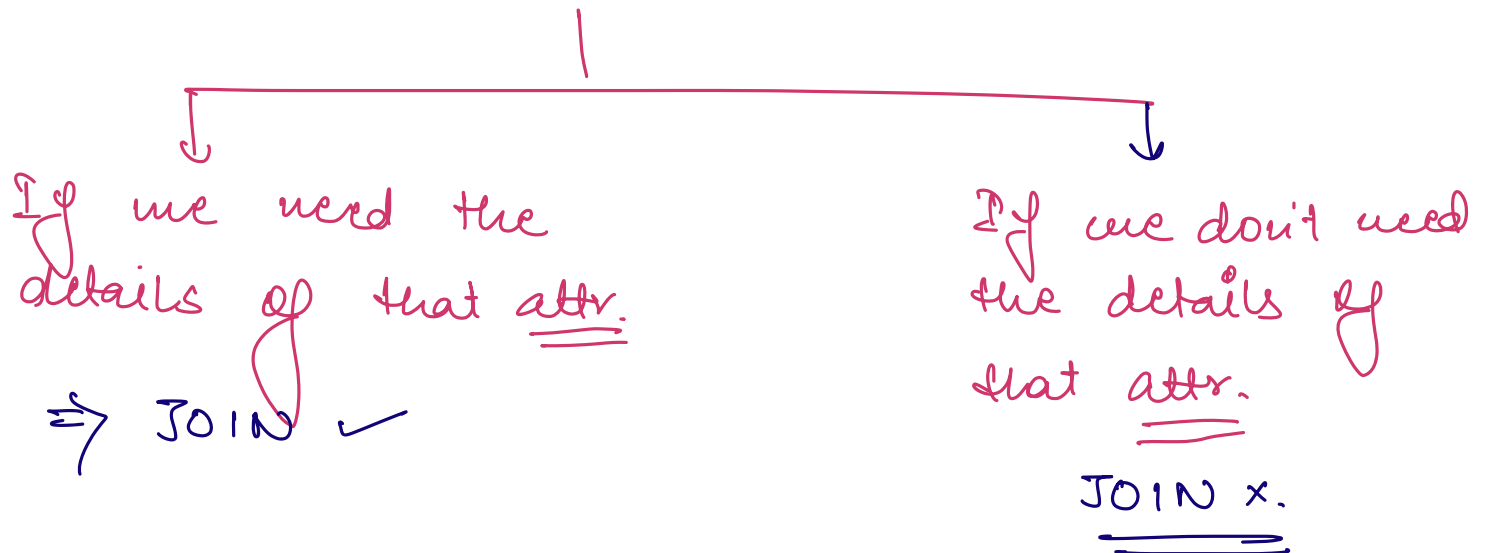
3

Category c = CategoryRepo.findById(10)

Category join Products

⇒ When a class has an attribute of another class, to fetch the details of that attr, a join operation will be performed internally.

⇒ It will make our query slow.



2 ways to fetch attrs of child class.

1) Eager fetch

2) lazy fetch.

Eager fetch \Rightarrow fetch attrs along with fetching the main object.

JOIN ✓

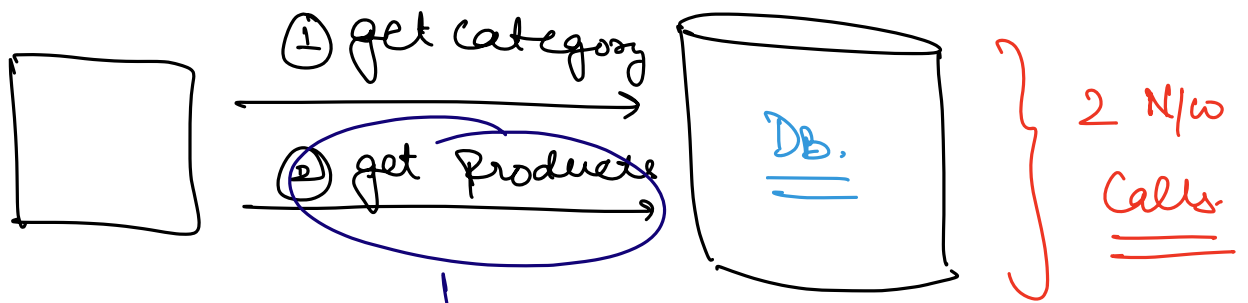
Lazy fetch \Rightarrow Don't fetch attrs along with fetching the main object.

\Rightarrow fetch only when required.

Category c = CategoryRepo.findById(10) \leftarrow No join
 \hookrightarrow Only fetch the category details & not the product details.

List<Product> products = (c.getProduct());

Products will be fetched now



↓

2nd query will only be executed
when required.

⇒ By default all the attrs are eagerly fetched
except the collection attr.