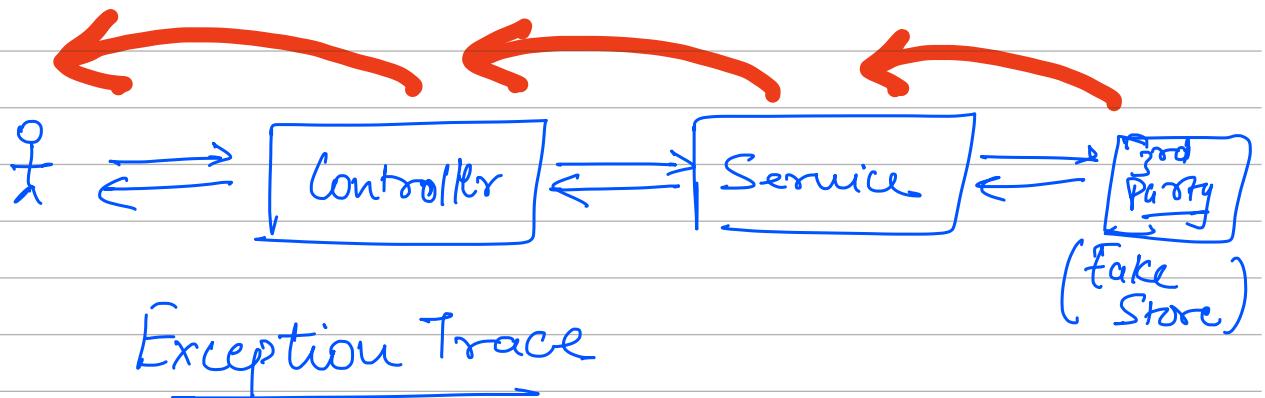


DEception Handling

2) Introduction to JPA



ProductController {

public get ProductById (—)

{

try {

productService . get — ()

4

catch (Exception e)

}

Responsibility ("Something went wrong
with 3rd party,
Please try
after sometime")

} }

Spring Works Annotations

@ExceptionHandler

Ideal we should Exception specific handling



Controller Advice

↳ final check on whatever is being returned by controller.

This class will modify the data sent by controller to client

@ControllerAdvice

Global Exception Handler

public class ExceptionHandler {

 @ExceptionHandler(—)

 {

 ↳

- ① Less code
- ② Modular



Intro to JPA

Product Service

→ As of now we have built the Product Service using fake store

→ Now we are going to build product service using our own DB



→ Connection

→ Schema ⇒ Create tables
Build Relations

→ CRUD operations

Create New Product (Product P) ↴

DBConn cbc = new DBConn();

cbc.connect();

cbc.fetchTable("Products");

If (nub) ↴

↳ Query q = "CreateTable —"

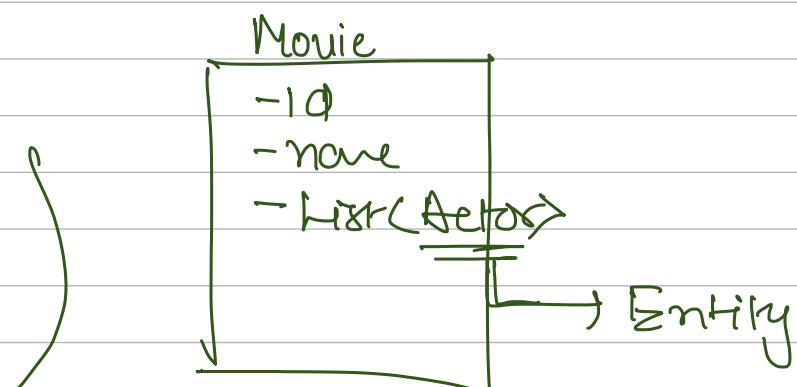
Query q = "Insert product"

q.execute()

↳

↳

Class Diagram = Schema



Movie — Actor $\equiv \underline{M:M}$

Mapping Table

Amazon Retail → u work

ORM Object Relation Mapping

Provides us a easy way to work with
Databases based on the Models that
are there in Code base

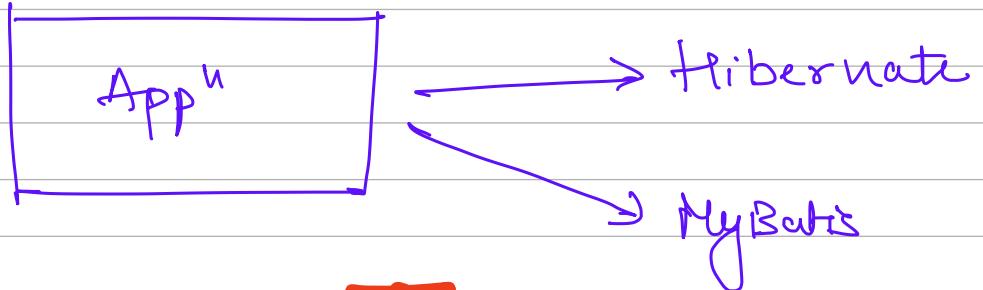
- ① Automatically creates corresponding table for Models
- ② Automatically performs CRUD operation

productrepo.findbyId(id);

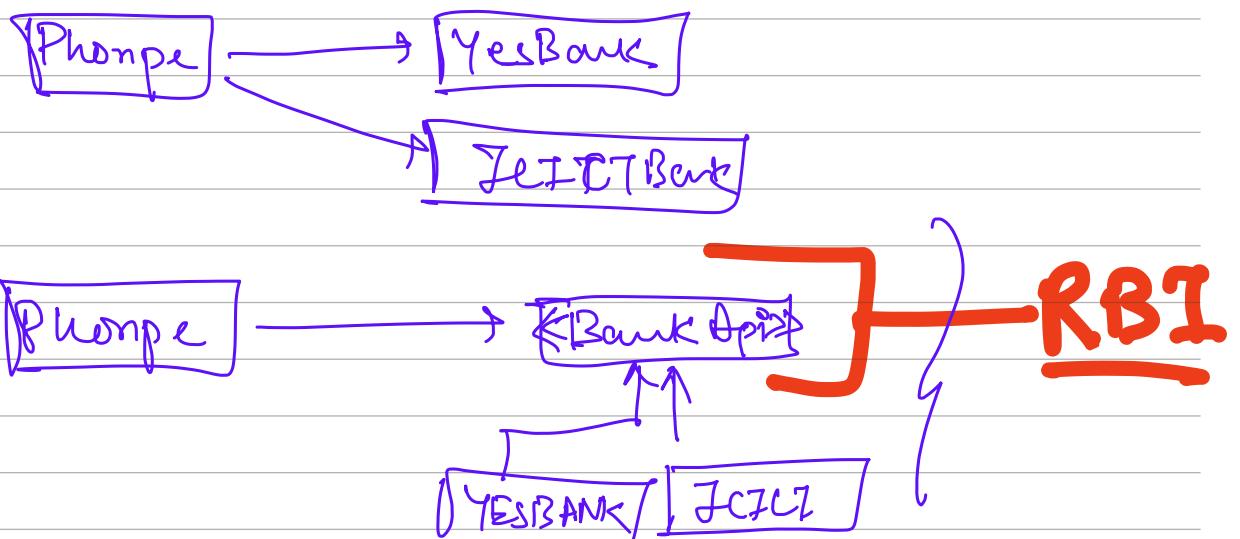
↓
Select * from products
where productId=id

ORM

- ① Hibernate
- ② My Batis
- ③ JDOQ



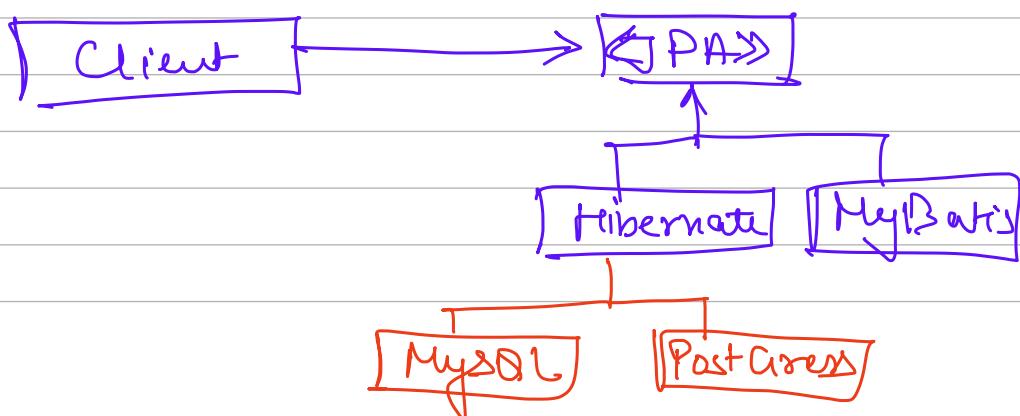
Right Coupling



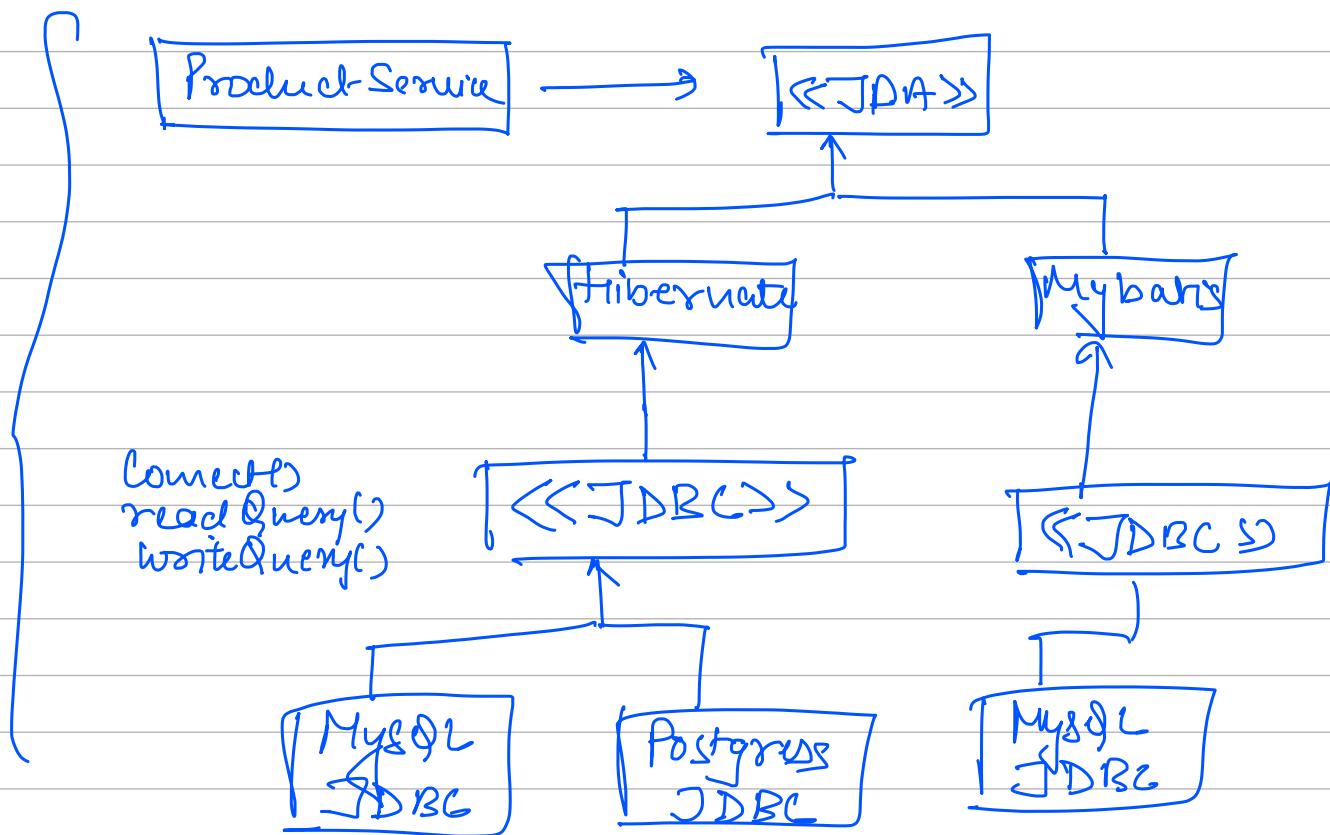
Programm to interface

JPA Java Persistence API

By default Java has an interface that implemented by different ORMs



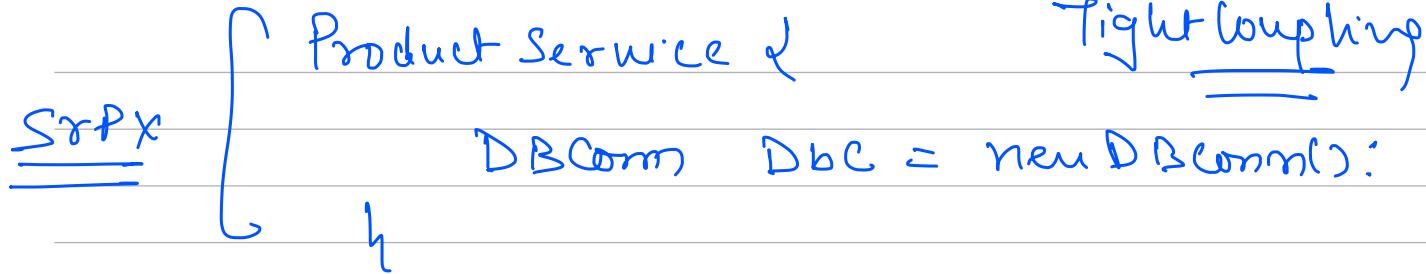
JDBC Java Database Connectivity



Repository Pattern

Code to interact with persistence layer should be separate from the business logic

Code to interact with DB should be present in Repo layer NOT in service layer



Product Repo ↴

DBComm dBC; ————— DI

Save() { }
Read() { }
Update() { }
—————
—————
—————