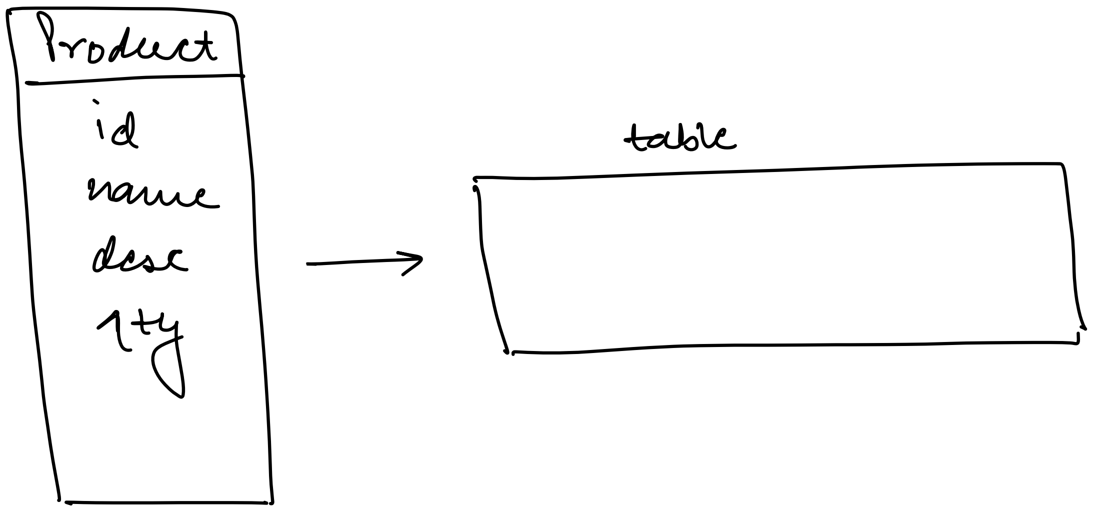
Agenda.

→ Represent inheritance in Database.

→ Setup our DB

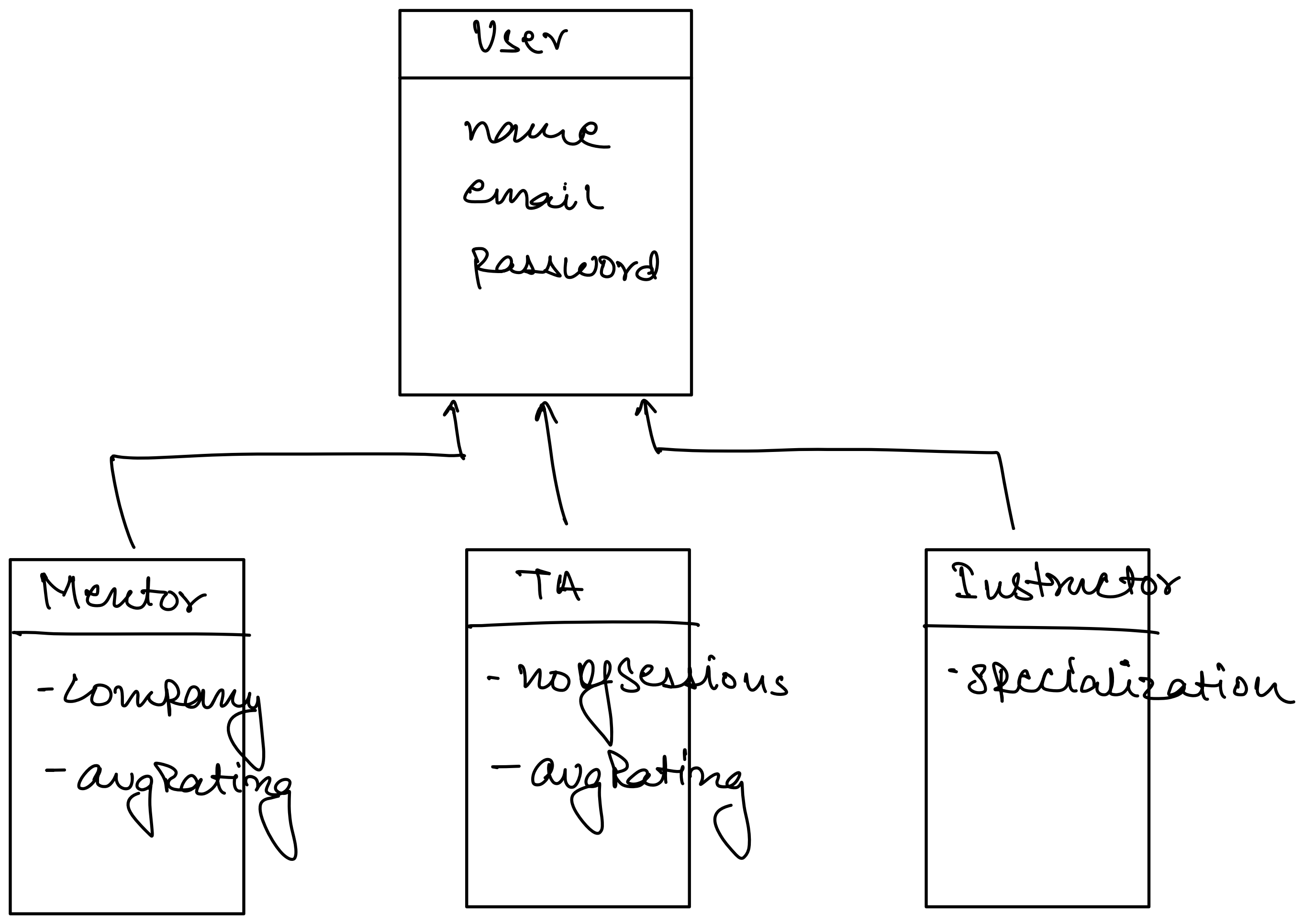
→ Integerate DB with ProductService



⇒ How to represent Inheritance in DB.

↳ 4 ways.

- Mapped Super Class
- JOINED Table
- Table Per Class
- Single Table class.



Store the data of instructors, ta & mentors in the DB, How?

① MappedSuperClass.

⇒ When there's No object of parent class.

⇒ Parent class can be marked as an Abstract Class.

Approach.

1) No table for parent class

2) One table for each of the class with their own attributes as well as attributes from parent class.

mentors

Company	avgRating	name	email	Password

ta

noOfSessions	avgRating	name	email	password

instructor

specialization	name	email	Passwo

Q. Get email of all the users.

Select email from mentors

UNION

Select email from ta

UNION

Select email from instructors

② Joined Table. ⇒ 99.99% (Best solⁿ)

→ Every data wrt objects of parent class, we'll store in the parent table.

→ for each sub class also, we'll create a table with only their own attributes.

→ We'll get parent class attrs in child classes via foreign key.

users

id	name	email	password

mentor

company	avgRating	user_id

ta

noOfSessions	avgRating	user_id

instructor

specialization	user_id

Q. Get email ids of all the Mentors.

⇒ JOIN Mentor & User.

Q. Get email of all the users.

↳ Single query

select email from users.

Usecase

⇒ There can be some user which is neither ta, nor Instructor & nor Mentor?

↳ MappedSuperClass ✗

↳ Joined Table.

③ Table per Class.

→ Exactly similar to MappedSuperClass, only difference, here we'll also create table for parent class as well.

→ table for each class will have their own attributes as well as parent class attributes.

users

name	email	Password
Suraj	—	—

mentors

Company	avgRating	name	email	Password

ta

noOfSessions	avgRating	name	email	Password

instructor

specialization	name	email	Passwo

<u>Note</u>

→ First define the inheritance relⁿ in the Codebase

→ Then identify the $\boxed{\text{query pattern.}}$ & go with on of the ways.

↓

Most frequently executed query.

④ Single Table. (Worst Solⁿ)

⇒ Create one table with all the columns across the tables.

⇒ Add one extra column user_type to recognize the type of user.

Users-

name	email	password	company	avgRating	noOfSess	avg	speciality
xyz	—	—	—	—	NULL	NULL	NULL
Deepr	—	—	NULL	NULL	NULL	NULL	LLD

⇒ Too many Nulls. (Sparse table)

⇒ Waste of space.

⇒ If a user can have multiple roles.

⇒ 4 ways to represent inheritance in DB.

- Mapped Super Class. ⇒ No object for Parent Class.
- ☆ Joined Table. ⇒ Table for parent class & Child tables will refer the parent class via FK.
- Table Per Class

  Exactly similar to Mapped Super Class, but table for parent class.
- Single Table. ✓

——— * ———