**Hibernate Mapping : -**

Hibernate mapping is one of the key features provided by hibernate which is used to establish the relationship between two database tables.

To achieve hibernate mapping ,Hibernate provides 4 different ways :

1. OneToOne
2. OneToMany
3. ManyToOne
4. ManyToMany

**Note:-**

* If the relationship is defines in only one of the tables , It is Known as

“Unidirectional Mapping”.

* If the relationship is defined both the tables , It is known as

“Bidirectional Mapping”.

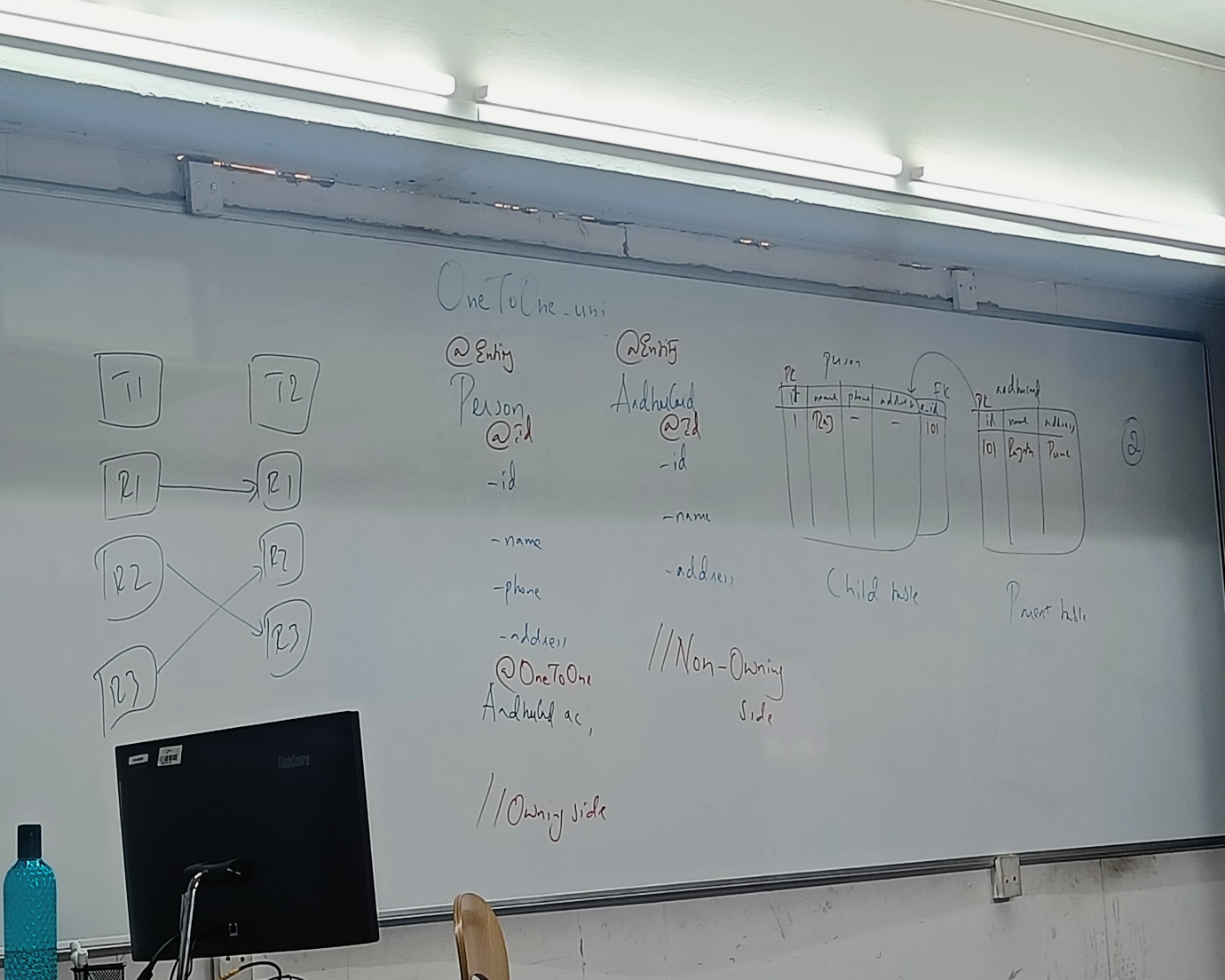
**OneToOne\_Uni:-**

When One Record From Table One Is Related To Exactly One Record From

Table 2. It Is Called As “OneToOne Mapping”.

**Ex:-**

1. **Person & AadharCard ✅**
2. **Student & Id**
3. **Person & PanCard**
4. **Employee & Salary Account**



**Q.)**

Here We Will Create Reference Variable Of One Class In An Another Class,

That Will Be Annoted with @OneToOne

By considering this annotation ,hibernate will create a foreign key inside

“OWNING SIDE”.

Here ,two tables will be generated.

**One is for :-** Owning Side (Person)And **Another One Is For :-** Non-Owning Side

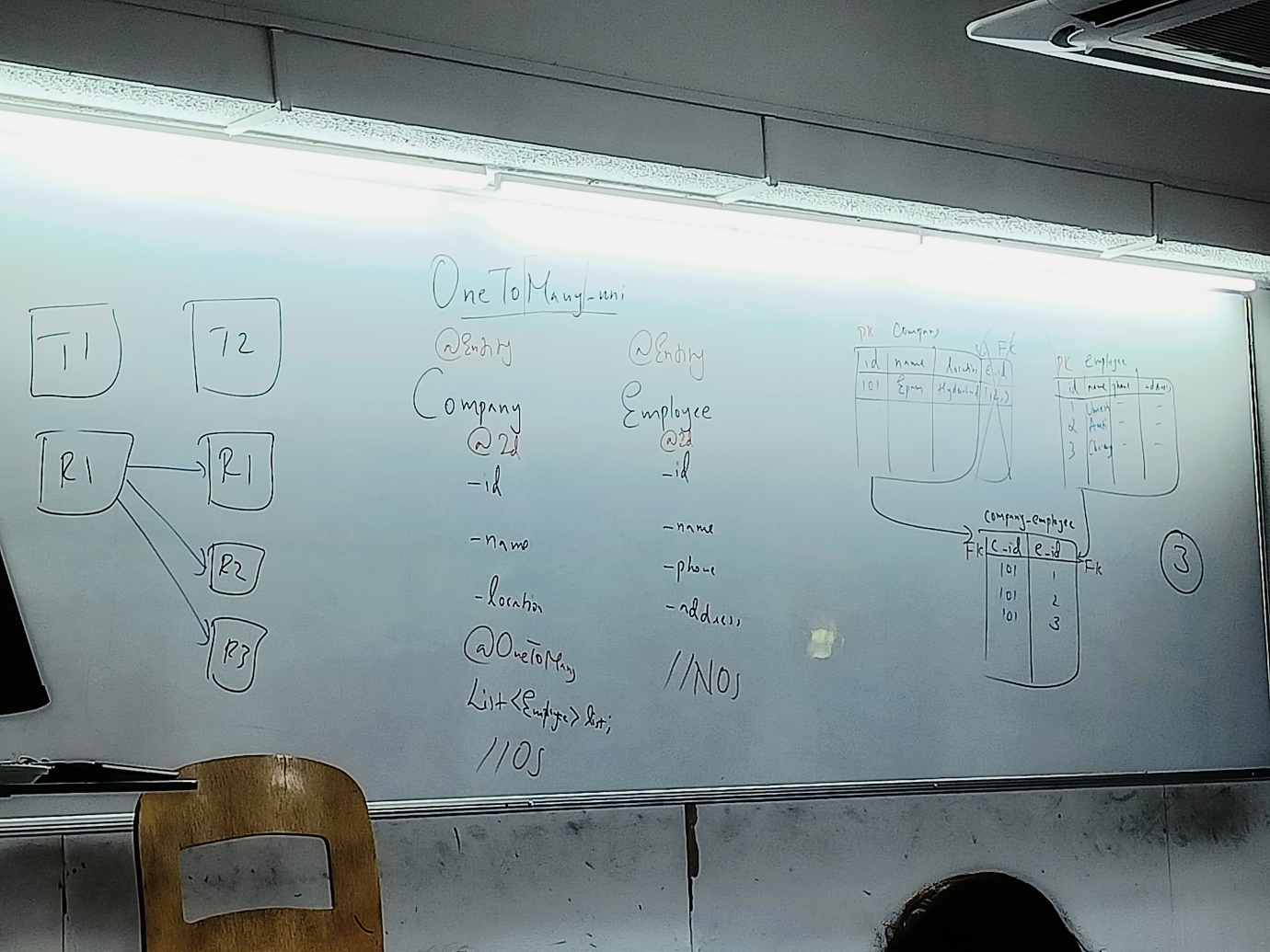
(AadharCard).

**OneToMany Mapping:-** Is Associated with many instances of other entity.***OR***

When One record from table 1 is related to more than one record from table 2 then it is known as **“OneToMany Mapping”.**

**Ex:-**

1. **Company & Employee ✅**
2. **School & Student**
3. **Country & State**
4. **Person & BankAccount**



By Considering The Scenario Of Company & Employee

To Achieve This We Have To Create Employee Object Inside Company Class With The Help Of List

(i.e.,List<Employee>list ;)

It Must Be Annoted With “@OneToMany”.

In OneToMany Mapping , 3 Tables Will Be Generated In The Database.

(One Is For :- Owning Side , One Is For :- Non-Owning Side & One Is For Storing Intermediate Tables.) To Achieve relationship This Is Done Because If 2 Tables Are created It Violets “E.F.CODD’s 1st Rule (ATOMICITY)”. & We Have To Make Primary Key Of Company As Duplicate.

It Is Not Possible hence memory are stored in separate table as foreign keys

**NOTE :-**

If the flow of relationship is keywords many,We have to create a collection of object can achieve HAS-A relationship.

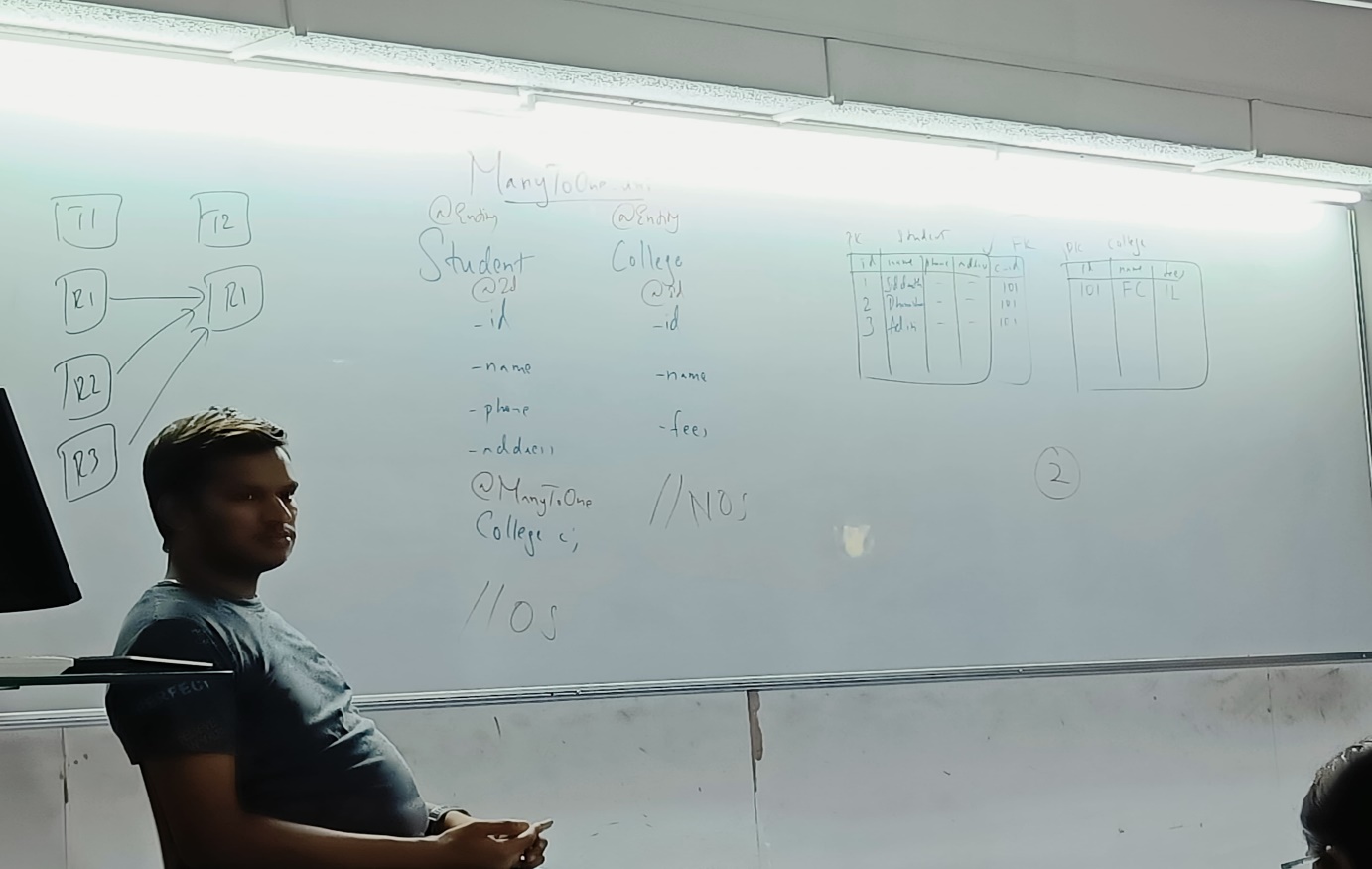
In this case ,hibernate will create an intermediate table to define the relationship.

**ManyToOne\_uni Mapping :-**

In ManyToOne Mapping Many instances of one entity is associated with one instance of other entity **|OR|** When more than one records from table one are related to exactly one record of another table, It is known as “ManyToOne Mapping”.

EX:-

1. Employees & Company
2. Students & College ✅
3. Branches & Hospital



ManyToOne Mapping means many object are pointing towards one object.

In this scenario we are going to create a reference variable of collage class inside student class.

Here,2 tables will be generated in the database.

One For The Owning Side (Student) & another One is For Non-Owning side (College).

The foreign key is present in the owning side (i.e., The Primary Key Of College Table Acts As Foreign key in student table)

**ManyToMany\_uni Mapping :-**

When more than one records from table 1 are related to more than 1 records in table 2 , it is known as “ManyToMany Mapping”.

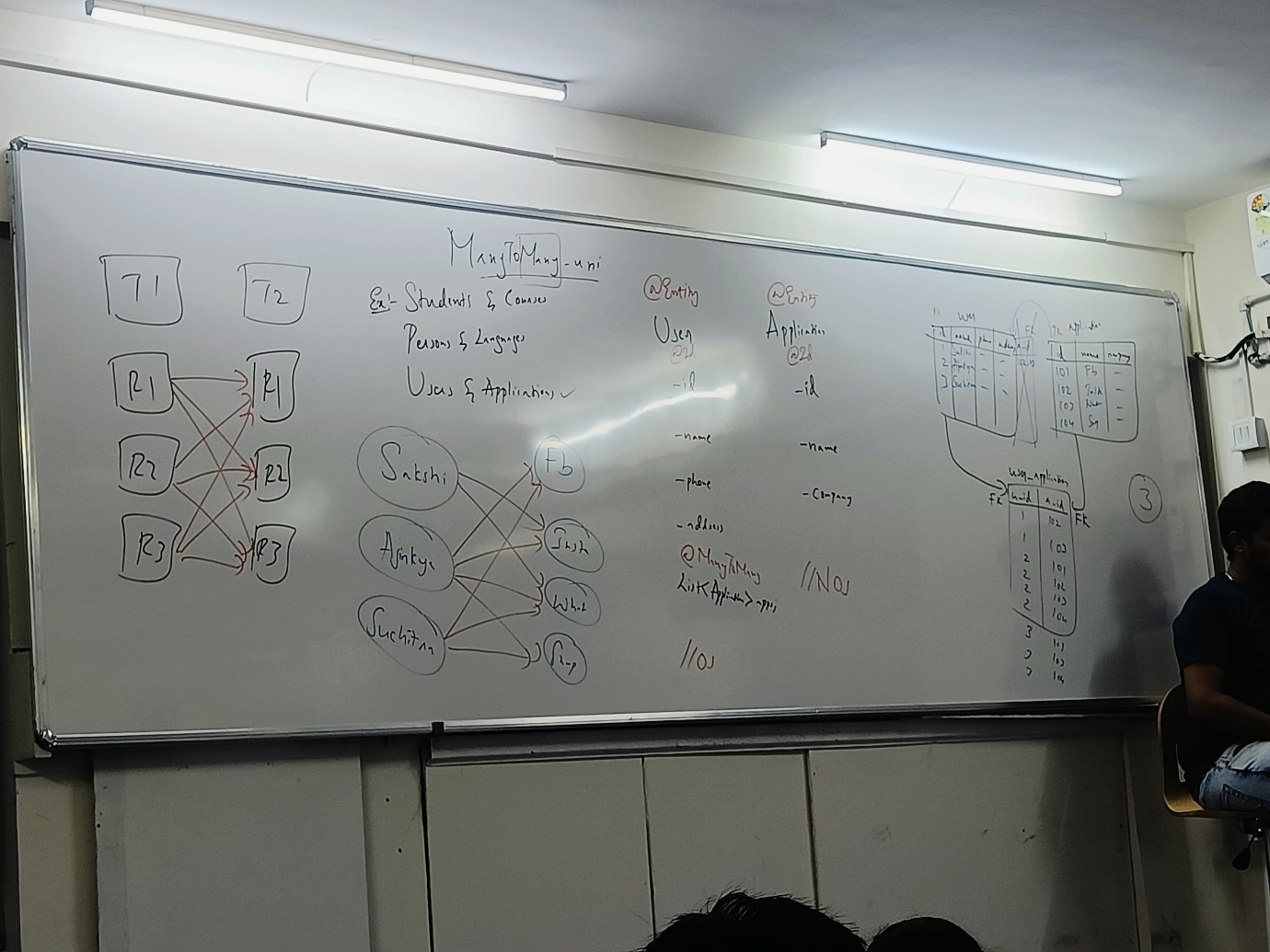
**|OR|**

In ManyToMany Mapping ,many instances of one entity is associated with many instances of other entity.

In simple words Many rows of one table is associated with another table

**EX:-**

1. Many Students Are associated with many courses.
2. Many Persons Are Associated With Many Languages.
3. Many Users Are Associated With Many Applications. ✅



ManyToMany Represents Collections-Value\_Association

Here Any nu. Of enitites can be associated with a collection of other entities.

Consider 2 entity classes,user and application

In this scenario, One user can have many applications,

As well as one application can have many users

In this case we will crate list of one entity type inside another entity class. It should be anoted with “@ManyToMany”.

In this mapping ,3 tables are generated in the database. Two for Two entity classes and one table to store foreign keys.

Here,3rd table is having primary keys of both the tables as foreign keys.

Note:-

The name of the intermediate table will always be

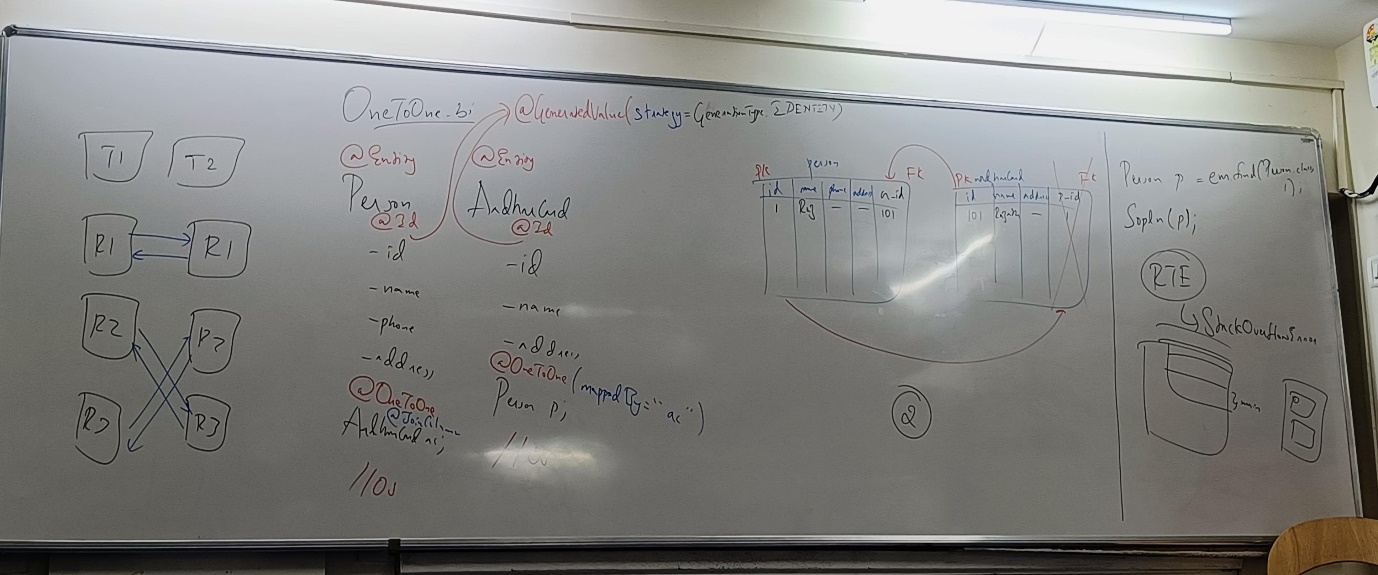
“owning Side \_non-Owning Side “.

OneToOne\_bidirection Mapping :-

Bidirectional relationship provides access in both directions.

In bidirectional OneToOne a class “A” has an object reference of another class “B” & Class “B” has the object reference of Class “A”

The Anotation “@OneToOne” is used in both the classes.





Without “mappedBy” attribute,hibernate will generate foreign key column in both the tables.To avoid that “mappedBy ” should be used .

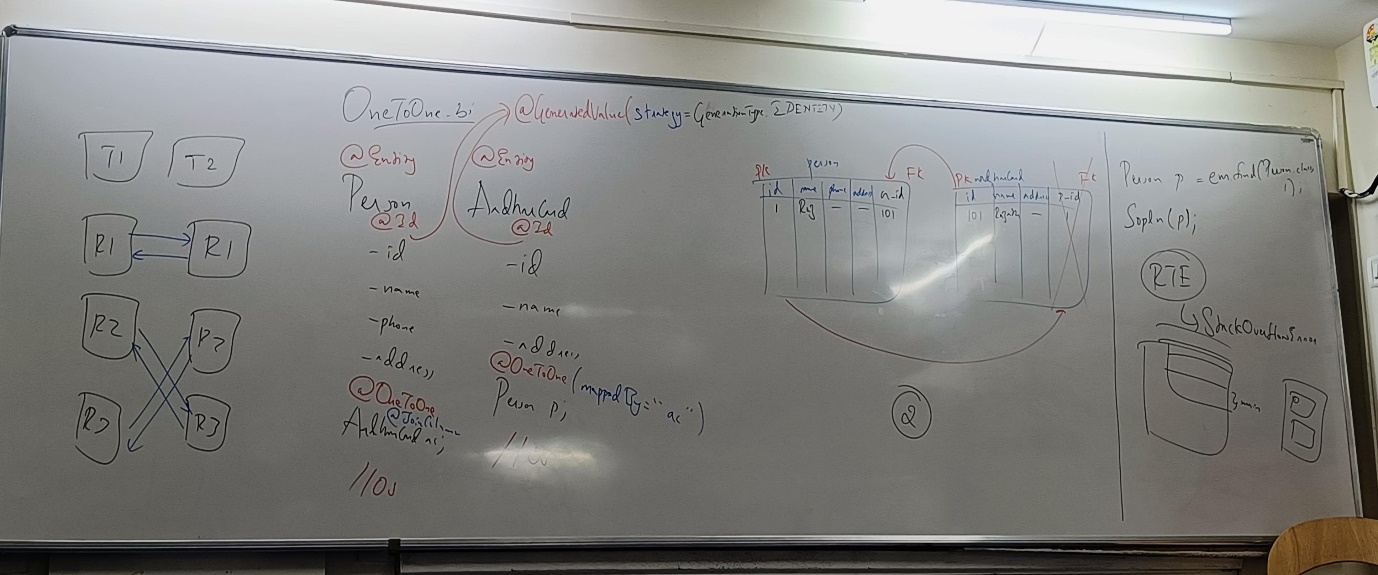
We must use “mappedBy” element with one of the “@OneToOne” annotation

The value of “mappedBy” should be the name of reference variable used in another class.

The class which has “mappedBy” attribute becomes the Non-owning side and the class which dosen’t have “mappedBy” element becomes “Owning side”.



In Owning Side we can use annotation “@JoinColumn”.The purpose of this annotation is to specify a foreign key column.





**Annotations Used :-**

**1)@GeneratedValue :-**

This Annotation is used to enable hibernate to automatically generate a value for mention a property.

The value is generated based on some generation strategy(Provided by an ENUM called “javax.persistence.GenerationType”).

The available strategies are “IDENTITY” , “AUTO” , ”SEQUENCE” ,”TABLE”.

**2)@JoinColumn:-**

It is used to specify a column for joining an entity association.

This annotation indicates the enclosing entity is the owner of the relationship & the corresponding table has a foreign key column which refers to the table of the “Non-Owning side”.

**3)mappedBy element / attribute:-**

We use mappedBy element with one of the mapping annotations

(except @ManyToOne)

Write on the top of page :-

Mapping Annotations:-

@OneToOne

@OneToMany

@ManyToOne

@ManyToMany

The value of mappedBy element should be the name of reference variable used in other class’s reference variable.

The element mappedBy is needed to specify which side is the owning side

Without mappedBy , hibernate will generate foreign key column in both the

Tables.

The side which has “mappedBy” element becomes the “Non-Owning Side” & the other side becomes the “Owning Side”.

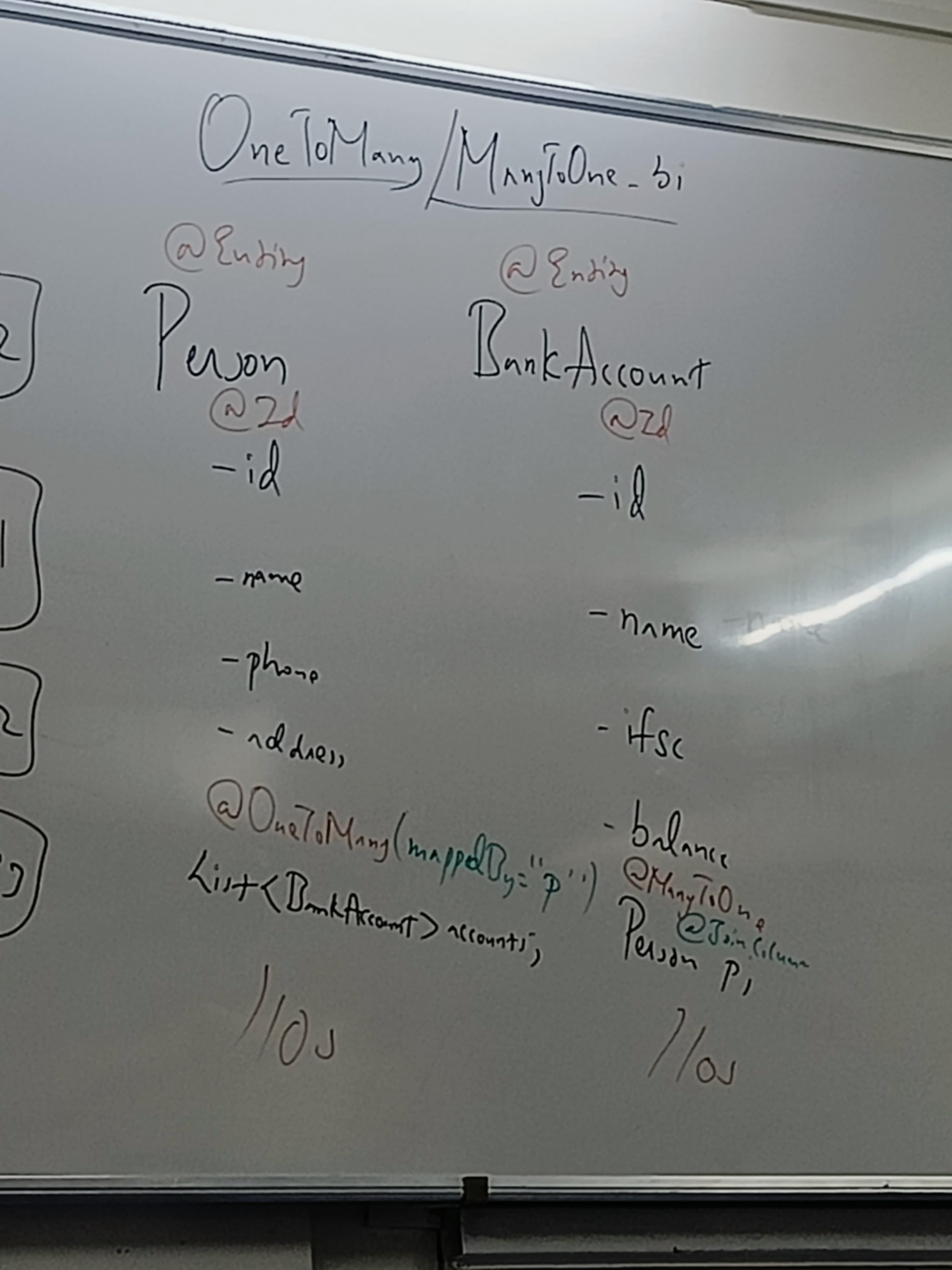
**OneToMany/ManyToOne\_bi Mapping :-**

**Ex :-**

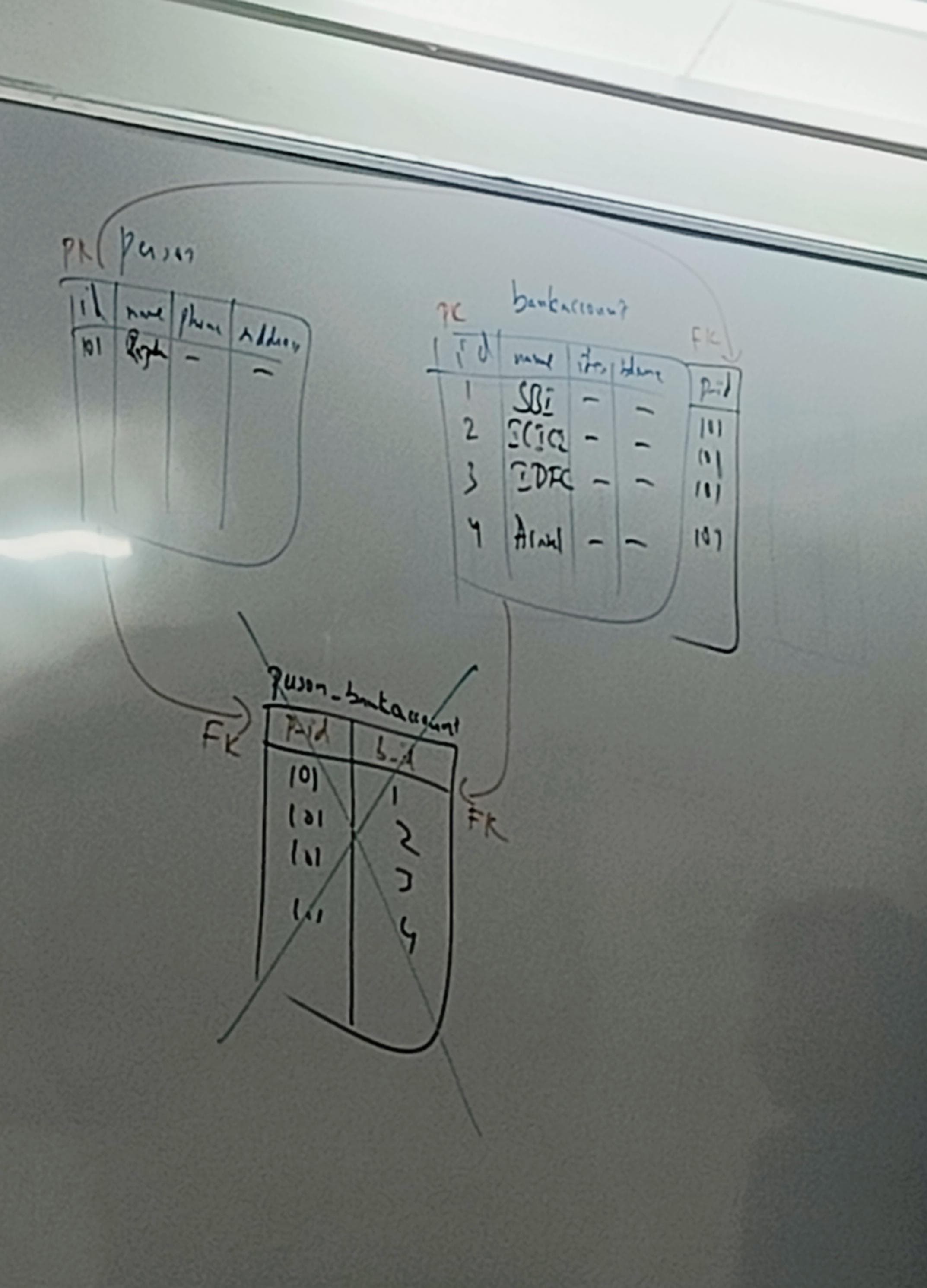
**1)Person & BankAccounts :-**

In this relationship ,we use annotations “@OneToMany” & “@ManyToOne”

The annotations called as “OneToMany” is used on the side which has collection reference . The annotations @ManyToOne is used on the side which has single valued object refrence.









We use mappedBy element to specify the which side is the owning side.

We use mappedBy with “@OneToMany” Annotation.

In OneToMany Unidirection, there will be 3 tables created for 2 entity classes.

Where as In ManyToOne Unidirection , 2 tables will created for 2 entity classes.

