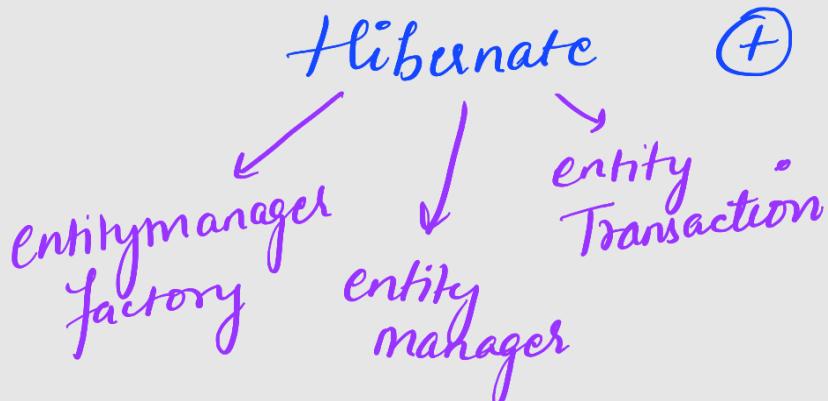


Introduction to Mapping

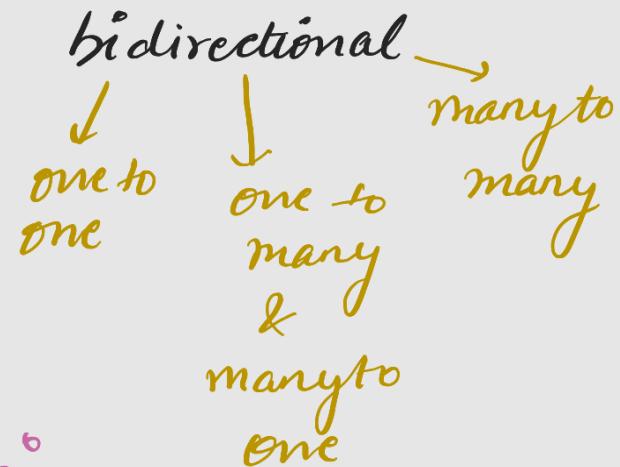
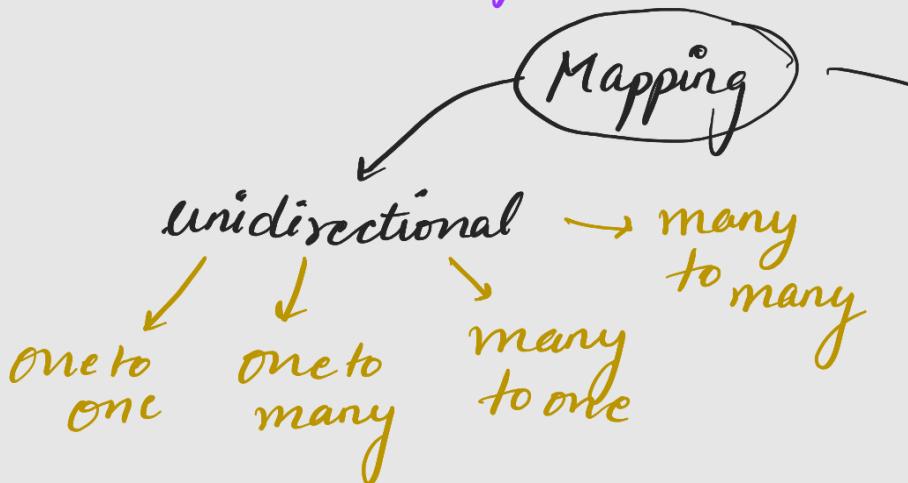
↳ linking two tables together

↳ achieved using hibernate & JPA

Java Persistence API



JPA → cascade
↳ cache
fetchType



Unidirectional Mapping

One to One Mapping

One person has one aadhar card.



@Entity

→ To identify the class as an entity

public class PersonDto {

@Id

→ To set id as p. key

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
private int id;
```

```
private String name;
```

```
@OneToOne
```

```
private AadharDto aadhar;
```

```
}
```

```
@Entity
```

```
public class AadharDto {
```

```
@Id
```

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
private int id;
```

```
private long number;
```

```
private String fName;
```

```
}
```

to generate id automatically so we don't need to handle it

connect one aadhar object to one person object

One to Many Mapping

One singer can have many songs.

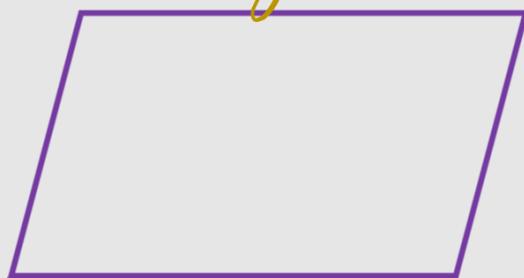
SingerDto

@OneToMany

private List<SongDto> song;

list of songs is assigned to one singer

SongDto



Many to One Mapping

Many to One Mapping

Many actors can work in one movie

Moviedto

actor dto

@manytoone

private moviedto mov;
assign all actors to
one movie.

Many to Many Mapping

Many students can have many subjects.

Student dto

assign a list of subjects to a student

@manytomany

private list<subject dto> subl;

Subject dto

The same list of subjects
can be assigned to
multiple students as
well.

Bidirectional Mapping

Person

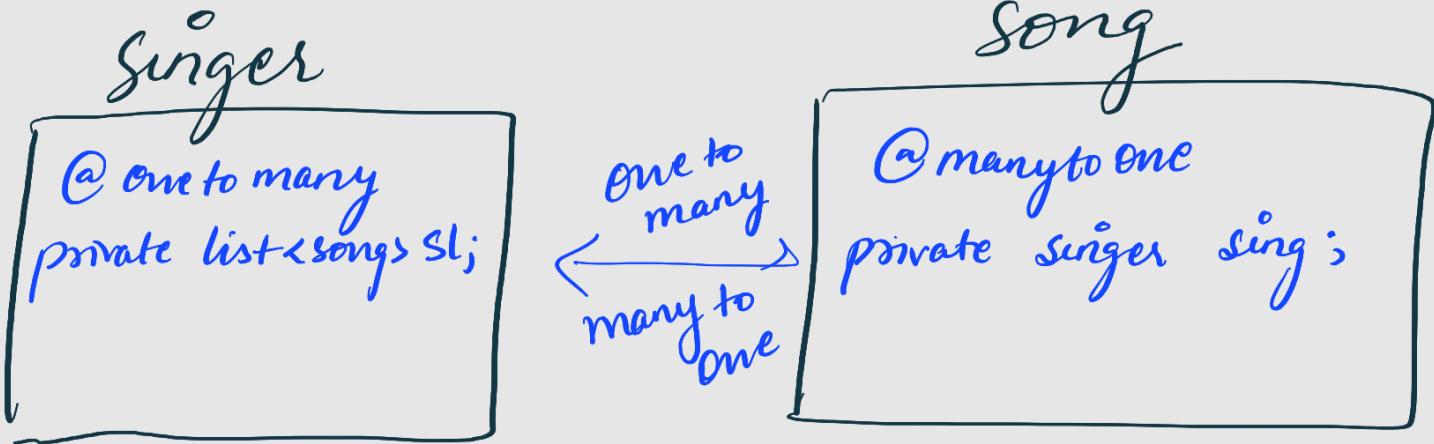
@onetooner

one
to
one
one
to
one

aadhar

@onetooner

One aadhar
belongs to one
person and
vice versa



@joinColumn

And so on....

↳ In case of bidirectional mapping, foreign key is created in both tables

To avoid that, we use join column in second table (aadhar)



We will use "mappedBy" in first column

Person dto

@OneToOne (mappedBy = "p")
private Aadhar a;

aadhar dto

@OneToOne
@join column
private Person p;

Introduction to Cascading

When we perform some action on target entity then the same action is applied on the associated entity.

For instance if person class has set CascadeType as

PERSIST then \Rightarrow when person class is saved/stored
 \hookrightarrow Aadhar class is also stored/saved.
(mappedBy = "P", cascade = CascadeType.ALL)
all operations

Introduction to fetch type

- * One of the strategies for fetching data from database
- + present in javax.persistence package

lazy loading fetch

when called then loaded

Data initialization occurs
as long as it is possible

Eager loading fetch

Everything loaded
even if not called

Data initialization
occurs on the spot

By default \Rightarrow eager fetch

(mappedBy = "P", fetchType.Lazy)

Default fetch Types

One to One \Rightarrow Eager fetch

One to Many \Rightarrow Lazy fetch

Many to One \Rightarrow eager fetch

Many to Many \Rightarrow lazy Fetch
