// uni directional

class Post {

@OneToMany(cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.LAZY)

@JoinColumn(name = "post\_id")

private List<PostComment> comments = new ArrayList<>();

}

class PostComment {

//not required.

}

//bi directional

class Post {

@OneToMany(

mappedBy = "post",

cascade = CascadeType.ALL,

orphanRemoval = true

)

private List<PostComment> comments = new ArrayList<>();

}

class PostComment {

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "post\_id")

private Post post;

}

// uni directional

class Post {

//not required

}

// you can use Student and University also.

class PostComment {

@ManyToOne(cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.LAZY)

@JoinColumn(name = "post\_id")

private Post post;

}

//bi directional

class Post {

@OneToMany(

mappedBy = "post",

cascade = CascadeType.ALL,

orphanRemoval = true

)

private List<PostComment> comments = new ArrayList<>();

}

class PostComment {

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "post\_id")

private Post post;

}

Many-To-Many Unidirectional

class Post {

@ManyToMany(cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.LAZY)

@JoinTable(

name = "post\_tag",

joinColumns = { @JoinColumn(name = "post\_id") },

inverseJoinColumns = { @JoinColumn(name = "tag\_id") } private List<PostComment> comments = new ArrayList<>();

}

class Tag {

//not required.

}

Many-To-Many Biidirectional

class Post {

@ManyToMany(cascade = CascadeType.ALL, orphanRemoval = true, fetch = FetchType.LAZY)

@JoinTable(

name = "post\_tag",

joinColumns = { @JoinColumn(name = "post\_id") },

inverseJoinColumns = { @JoinColumn(name = "tag\_id") } private List<Tag> tags = new ArrayList<Tag>();

}

class Tag {

@ManyToMany(mappedBy="tags")

private List<Post> post = new ArrayList<Post>();

}