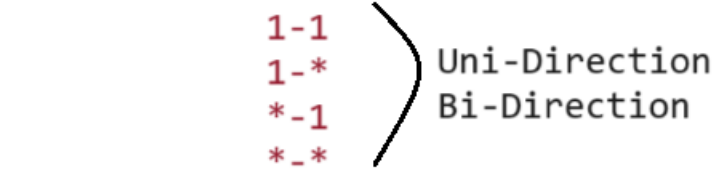
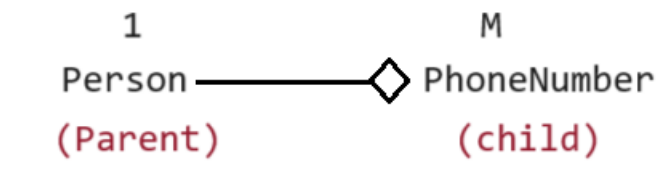


- a. Basic ORM  
b. Bulk Operations  
c. Advanced ORM  
    1. Component Mapping  
    2. Collection Mapping  
    3. Inheritance Mapping  
    4. Association Mapping



1. @OneToMany



Collection-Type

- a. List  
b. Set(no duplicate)  
c. Map

what is cascading?

=> Performing non-select operation on the object  
    cascade = CascadeType.ALL

(insert,update,delete)

done on parent should be  
reflected to child

Load child objects in **Eager loading**  
(default:::lazy loading)

Person.java  
@OneToMany(targetEntity = PhoneNumber.class,cascade = CascadeType.ALL,fetch = FetchType.EAGER)  
@JoinColumn(name="person\_id",referencedColumnName = "pid")//creating a fk in PhoneNumber Table  
private Set<PhoneNumber> numbers;//special property to hold many phoneNumbers

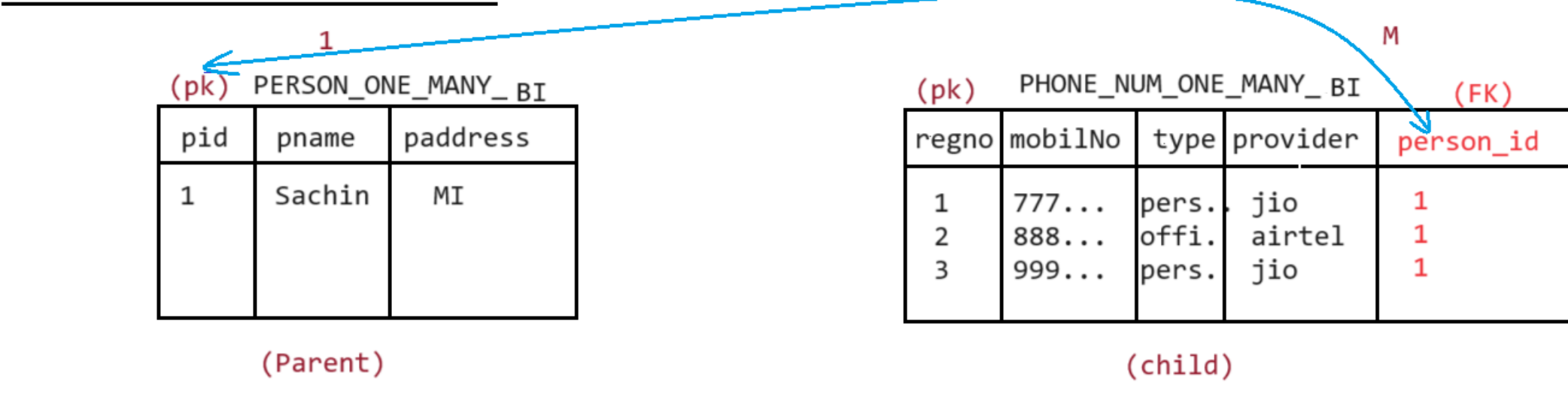
Solution  
@OneToMany(targetEntity = PhoneNumber.class,  
    cascade = CascadeType.ALL,  
    fetch = FetchType.EAGER,  
    orphanRemoval = true)  
//creating a fk in PhoneNumber Table  
@JoinColumn(name="person\_id",referencedColumnName = "pid")  
private Set<PhoneNumber> numbers;//special property to hold many phoneNumbers

regno	mobilNo	provider	type	person_id	pk
1	8889977	airtel	office	(NULL)	
2	7788999	jio	personal	(NULL)	
3	9977788	jio	personal	(NULL)	
4	8889977	airtel	office	3	
5	9977788	jio	personal	3	
6	7788999	jio	personal	3	
7	6677788	v1	office	(NULL)	

OrphanRecords

- src  
in.pawskills.nitin.bean  
  Person.java  
  PhoneNumber.java  
in.pawskills.nitin.dao  
  PersonDao.java  
in.pawskills.nitin.main  
  PersonDaoImpl.java  
  MainApp.java  
in.pawskills.nitin.util  
  HibernateUtil.java  
hibernate.cfg.xml  
JRE System Library (JawaSE-17)  
Hibernate  
MySQL  
hikaricp

OneToMany-BiDirectional Mapping

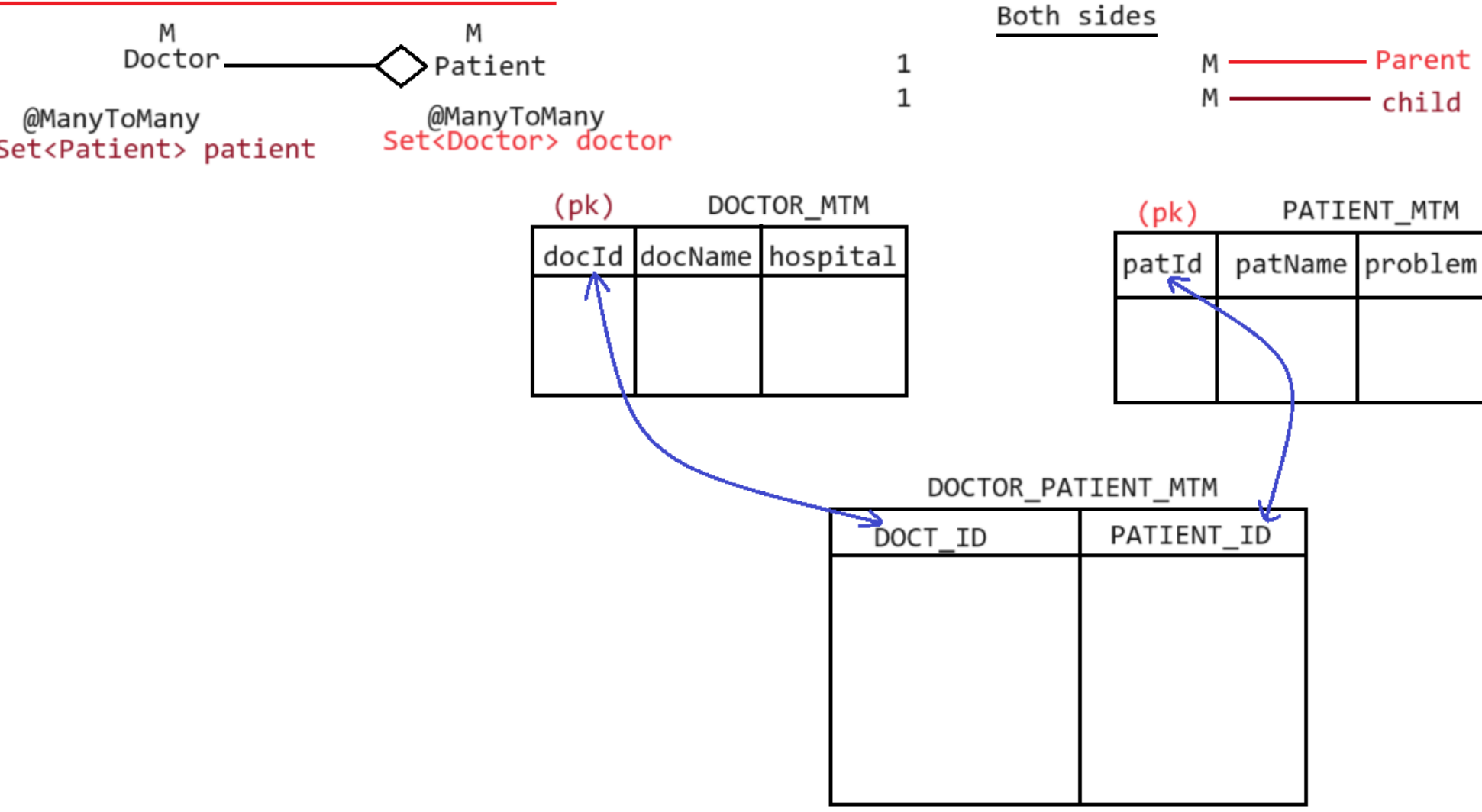


Person.java  
@OneToMany(targetEntity = PhoneNumber.class,  
    cascade = CascadeType.ALL,  
    fetch = FetchType.EAGER,  
    orphanRemoval = true,  
    mappedBy = "person")  
Set<PhoneNumber> number;

PhoneNumber.java  
@ManyToOne(targetEntity = Person.class,  
    cascade = CascadeType.ALL,  
    fetch = FetchType.EAGER,  
    mappedBy = "person")  
Person person;

Created a link by referring to Person

ManyToManyAssociation(Bi-Direction)



Doctor.java  
@ManyToMany(targetEntity = Patient.class,cascade = CascadeType.ALL,fetch = FetchType.EAGER)  
@JoinTable(name = "DOCTOR\_PATIENT\_MTM",  
    joinColumns = @JoinColumn(name="DOCT\_ID",referencedColumnName = "docId"),  
    inverseJoinColumns = @JoinColumn(name="PAT\_ID",referencedColumnName = "patId"))  
private Set<Patient> patients;

Patient.java  
@ManyToMany(targetEntity = Doctor.class,  
    cascade = CascadeType.ALL,  
    fetch = FetchType.EAGER,  
    mappedBy = "patients")  
private Set<Doctor> doctor;

public interface IHospitalDao {  
    //performing insert operation using parent  
    public void saveRecordUsingParent();  
    public void saveRecordUsingChild();  
  
    //performing select operation using parent  
    public void loadRecordUsingParent();  
    public void loadRecordUsingChild();  
}

Note: mappedBy should be placed in "@OneToMany" side in OneToMany-BiDirection  
where as for ManyToMany and OneToOne Bi-direction we can keep on any side.