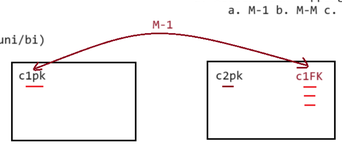
O
R ⇨ Object Relational Mapping
M

1. Basic ORM
2. Bulk Operation
3. Advanced ORM
   a. Composition Mapping
   b. Inheritance Mapping
   c. Collection Mapping
   d. Association Mapping(uni-dir and bi-dir)
      a. M-1 b. M-M c. 1-1 d. 1-M

1. JOINS

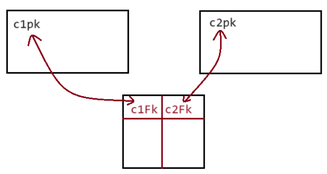
2. 1+N Select problem
3. Locking in hibernate

1. M-1(uni/bi)

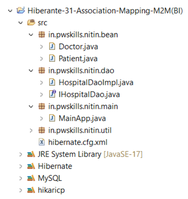


Annotations used are
 a. @ManyToOne(....,mappedBy="")
 b. @OneToMany
    @JoinColumn(....)

2. M-M(bi)



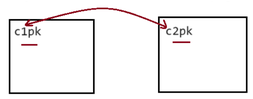
Annotations used are
 a.@ManyToMany(........,mappedBy="")
 b.@JoinTable(name="",
            joinColumns = @JoinColumn(...),
            inverseJoinColumn = @JoinColumn(..))



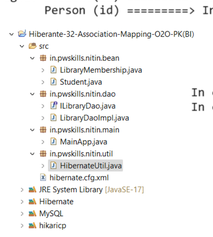
3. 1-1(BI)

 a. one-one (PK) [Both parent and child class would use same id values as pk]
 b. one-one (FK) [Both parent and child class are not ready to use same id values as pk]

eg#1. Student(id) =========> LibraryMembership(id)
      Person (id) =========> IncomeTaxFilling(id)



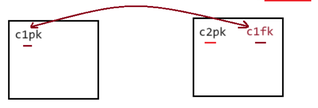
Annotations used are
 a. @OneToOne(....,mappedBy="")
 b. @PrimaryKeyJoinColumn(,,)

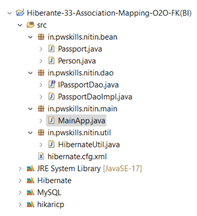


In case of One-One mapping, by default fetch type will be "EAGER".
In case of Many-One or Many-Many by default fetch type will be "LAZY".

1-1(FK)

eg: Person(adharNo) -------> PASSPORT(PassPortNo,adharNo)
      (pk)   (pk)   (FK)





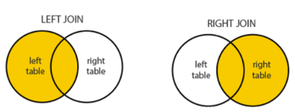
HQL JOINS

 => They are the implicit conditions that are given to common and uncommon data of DB table that are in relationship

 => HQL joins are similar to SQL-Joins
    a. Inner join
    b. left join
    c. right join
    d. full join

(person)

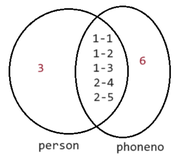
pid	paddress	pname
1	MI	sachin
2	rcb	kohli
3	csk	dhoni



(phoneNo)

regNo	mobileNo	provider	type	person_id
1	888999777	airtel	office	1
2	777888999	jio	personal	1
3	999777888	jio	personal	1
4	666888999	aircel	company	2
5	666777888	vi	office	2
6	77789989	vi	personal	(NULL)

1. inner join :: common data(.....)
2. left join :: 3 +common data(.....)
3. right join :: 6 +common data(.....)
4. full join :: 3,commondata(...),6




```
String  HQL_INNER_JOIN="select
                        per.pid,per.pname,per.paddress, |parent attributes
                        num.regNo,num.mobileNo,num.type,num.provider |child attributes
                        from Person per //parent
                             inner join <-------------- //type of join
                                per.numbers num "; //child


String  HQL_RIGHT_JOIN="select
                             num.regNo,num.mobileNo,num.type,num.provider, |parent attributes
                             per.pid,per.pname,per.paddress |child attributes
                             from PhoneNumber num //parent
                                  inner  join <-------------- //type of join
                                       num.person per " //child
```