

1 .. 1

```
class Associate {
    int associateNo;
    String firstName;
    String lastName;
    LocalDate dob;
    String gender;
    String designation;
    String mobileNo;
    String emailAddress;
    // accessors
}
```

*..1

```
class Project {
    int projectNo;
    String projectTitle;
    int duration;
    String businessUnit;
    String clientName;
    int teamSize;
    Set<Associate> associates;
    // accessors
}
```

```
associate
associate_no (pk)
first_nm
last_nm
dob
gender
designation
mobile_no
email_address
project_no (fk) (nullable)
```

```
project
project_no (pk)
project_title
duration
business_unit
client_nm
team_size
```

How to perform the database operations on the entity objects?

1. PERSIST/SAVE

1.1 Association

directly store the Associate object into the associate table. as we dont have association relationship between Associate to Project, we need to persist the relationship column as "NULL"

1.2 Project

A Project object holds Set<Associate> associates working in that project, so when we ask hibernate/jpa to persist/save the project it performs the below operations on the underlying tables

- store the project object data into the project table directly
- but when it comes to associate object "associates", as all these associates are working in the project, to represent that association in table relationship for each associate in the Set we need to update foreign key column project_no with project_no of the project object we persisted above

That means while storing/persisting the project object we have to perform operations on #2 tables

1. insert into project able

2. update associate table all the associated records with the project_no as foreign key

2. FETCH/GET

2.1 Associate

```
Associate associate = session.get(Associate.class, 1);
```

directly we can query the data from associate table and retrieve the data for Associate object based on primary key value supplied.

2.2 Project

```
Project project = session.get(Project.class, 1);
```

Since Project entity class has Set<Associate> associates objects within it, so while fetching the data for the project, along with project the associates working in the project also should be retrieved

- query the data from project table for fetching the project entity class attributes
- for associate object (Set:-associates) : take the primary key column of project table and join with foreign key column of associate table to find which associates are working on which project and filter them with where clause of project_no to find all the associates working in the project

In-Short:

1. While persisting the data: take the primary key of my object and store it as foreign key in another table

2. While fetching the data: take the primary key of my object and join with foriegn key of another table to fetch the related objects

Associate.hbm.xml

```
<hibernate-mapping package="com.otm.entities">
    <class name="associate" table="associate">
        <id name="associateNo" column="associate_no">
            <generator class="increment"/>
        </id>
        <property name="firstName" column="first_nm"/>
        <property name="lastName" column="last_nm"/>
        // all properties
    </class>
</hibernate-mapping>
```

Project.hbm.xml

```
<hibernate-mapping package="com.otm.entities">
    <class name="Project" table="project">
        <id name="projectNo" column="project_no">
            <generator class="increment"/>
        </id>
        <property name="projectTitle" column="project_title"/>
        <property name="duration"/>
        <property name="teamSize" column="team_size"/>
        <property name="clientName" column="client_nm"/>
        <set name="associates">
            <key column="project_no"/>
            <one-to-many class="Associate"/>
        </set>
    </class>
</hibernate-mapping>
```