Bi-Directional, Fetch and Cascading Codes for Hibernate

1) One-to-One example : Employee (1…1) Profile:

Employee Model:-

```
package com.app.model;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="emptab")
public class Employee {

	@Id
	@Column(name="eid")
	private int empId;
	@Column(name="ename")
	private String empName;
	@Column(name="esal")
	private double empSal;


	@ManyToOne(fetch=FetchType.EAGER,cascade=CascadeType.ALL)
	@JoinColumn(name="eidFk",unique=true)
	private Profile prof;


	public Employee() {
		super();
	}

	public Employee(int empId, String empName, double empSal) {
		super();
		this.empId = empId;
		this.empName = empName;
		this.empSal = empSal;
	}

	public int getEmpId() {
		return empId;
	}
```

```
	public void setEmpId(int empId) {
		this.empId = empId;
	}
	public String getEmpName() {
		return empName;
	}
	public void setEmpName(String empName) {
		this.empName = empName;
	}
	public double getEmpSal() {
		return empSal;
	}
	public void setEmpSal(double empSal) {
		this.empSal = empSal;
	}
	public Profile getProf() {
		return prof;
	}
	public void setProf(Profile prof) {
		this.prof = prof;
	}
	@Override
	public String toString() {
		return "Employee [empId=" + empId + ", empName=" + empName +
", empSal=" + empSal + ", prof=" + prof + "]";
	}


}
```

Profile Model:-

```
package com.app.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name="proftab")
public class Profile {
	@Id
	@Column(name="pid")
	private int profId;
	@Column(name="pcode")
	private String profCode;
	@Column(name="pdata")
	private String otherData;
```

```
	//Bi-Directional code
	@OneToOne(mappedBy="prof")
	private Employee emp;

	public Profile() {
		super();
	}
	public Profile(int profId, String profCode, String otherData) {
		super();
		this.profId = profId;
		this.profCode = profCode;
		this.otherData = otherData;
	}
	public int getProfId() {
		return profId;
	}
	public void setProfId(int profId) {
		this.profId = profId;
	}
	public String getProfCode() {
		return profCode;
	}
	public void setProfCode(String profCode) {
		this.profCode = profCode;
	}
	public String getOtherData() {
		return otherData;
	}
	public void setOtherData(String otherData) {
		this.otherData = otherData;
	}
	@Override
	public String toString() {
		return "Profile [profId=" + profId + ", profCode=" + profCode
+ ", otherData=" + otherData + "]";
	}


}
```


hibernate.cfg.xml file:-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
	<session-factory>
```

```
            <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
            <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property
>
            <property name="hibernate.connection.username">root</property>
            <property name="hibernate.connection.password">root</property>
            <property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

            <property name="hibernate.show_sql">true</property>
            <property name="hibernate.format_sql">true</property>
            <property name="hibernate.hbm2ddl.auto">create</property>

            <mapping class="com.app.model.Profile" />
            <mapping class="com.app.model.Employee" />
      </session-factory>
</hibernate-configuration>
```

Test class:-

```
package com.app.test;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.app.model.Employee;
import com.app.model.Profile;
import com.app.util.HibernateUtil;

public class Test {

      public static void main(String[] args) {
            Transaction tx=null;
            try(Session ses=HibernateUtil.getSf().openSession()){
                  //Step#1 run this first to create tables with data
                  tx=ses.beginTransaction();

                  Profile prof=new Profile(101, "P1", "NONE");
                  Employee emp1=new Employee(11,"B",2.2);

                  emp1.setProf(prof);

                  ses.save(emp1);

                  tx.commit();
            } catch (Exception e) {
                  e.printStackTrace();
            }
      }
}
```

2) many-to-One example : Employee (...1) Department:*

parent class:-

```
package com.app.model;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="emptab")
public class Employee {

	@Id
	@Column(name="eid")
	private int empId;
	@Column(name="ename")
	private String empName;
	@Column(name="esal")
	private double empSal;

	@ManyToOne(fetch=FetchType.EAGER,cascade=CascadeType.ALL)
	@JoinColumn(name="eidFk")
	private Department dept;


	public Employee() {
		super();
	}

	public Employee(int empId, String empName, double empSal) {
		super();
		this.empId = empId;
		this.empName = empName;
		this.empSal = empSal;
	}

	public int getEmpId() {
		return empId;
	}
	public void setEmpId(int empId) {
		this.empId = empId;
	}
	public String getEmpName() {
		return empName;
	}
	public void setEmpName(String empName) {
		this.empName = empName;
```

```
	}
	public double getEmpSal() {
		return empSal;
	}
	public void setEmpSal(double empSal) {
		this.empSal = empSal;
	}

	public Department getDept() {
		return dept;
	}

	public void setDept(Department dept) {
		this.dept = dept;
	}

	@Override
	public String toString() {
		return "Employee [empId=" + empId + ", empName=" + empName +
", empSal=" + empSal + ", dept=" + dept + "]";
	}


}
```

child class:-

```
package com.app.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name="DeptTab")
public class Department {

	@Id
	@Column(name="did")
	private int deptId;
	@Column(name="dcde")
	private String deptCode;
	@Column(name="dname")
	private String deptName;

	@OneToOne(mappedBy="dept")
	private Employee emp;


	public Department() {
		super();
```

```
	}
	public Department(int deptId, String deptCode, String deptName) {
		super();
		this.deptId = deptId;
		this.deptCode = deptCode;
		this.deptName = deptName;
	}
	public int getDeptId() {
		return deptId;
	}
	public void setDeptId(int deptId) {
		this.deptId = deptId;
	}
	public String getDeptCode() {
		return deptCode;
	}
	public void setDeptCode(String deptCode) {
		this.deptCode = deptCode;
	}
	public String getDeptName() {
		return deptName;
	}
	public void setDeptName(String deptName) {
		this.deptName = deptName;
	}
	public Employee getEmp() {
		return emp;
	}
	public void setEmp(Employee emp) {
		this.emp = emp;
	}
	@Override
	public String toString() {
		return "Department [deptId=" + deptId + ", deptCode=" +
deptCode + ", deptName=" + deptName + ", emp=" + emp
				+ "]";
	}

}
```


hibernate.cfg.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
	<session-factory>
		<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
```

```
            <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property
>
            <property name="hibernate.connection.username">root</property>
            <property name="hibernate.connection.password">root</property>
            <property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

            <property name="hibernate.show_sql">true</property>
            <property name="hibernate.format_sql">true</property>
            <property name="hibernate.hbm2ddl.auto">create</property>

            <mapping class="com.app.model.Department" />
            <mapping class="com.app.model.Employee" />
      </session-factory>
</hibernate-configuration>
```

HibernateUtil :-

```
package com.app.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

      private static SessionFactory sf=null;
      static {
            sf=new Configuration()
                        .configure()
                        .buildSessionFactory();
      }
      public static SessionFactory getSf() {
            return sf;
      }
}
```

Test class:

```
package com.app.test;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.app.model.Department;
import com.app.model.Employee;
import com.app.util.HibernateUtil;

public class Test {
```

```
	public static void main(String[] args) {
		Transaction tx=null;
		try(Session ses=HibernateUtil.getSf().openSession()){
			//Step#1 run this first to create tables with data
			tx=ses.beginTransaction();

			Department dept=new Department(101, "DEPT-A", "DEV");

			Employee emp1=new Employee(11,"B",2.2);
			emp1.setDept(dept);

			Employee emp2=new Employee(12,"C",3.3);
			emp2.setDept(dept);

			Employee emp3=new Employee(13,"D",4.4);
			emp3.setDept(dept);


			ses.save(emp1);
			ses.save(emp2);
			ses.save(emp3);

			tx.commit();
		} catch (Exception e) {
			e.printStackTrace();
		}
	}
}
```


3) One-to-Many example : Employee (1...) Address:*

parent class:-

```
package com.app.model;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name="emptab")
public class Employee {

	@Id
	@Column(name="eid")
	private int empId;
	@Column(name="ename")
	private String empName;
	@Column(name="esal")
	private double empSal;

	@OneToMany(fetch=FetchType.EAGER,cascade=CascadeType.ALL
mappedBy="emp")
	//@JoinColumn(name="eidFk") //-Must not be provided
	private Set<Address> addr=new HashSet<Address>(0);


	public Employee() {
		super();
	}

	public Employee(int empId, String empName, double empSal) {
		super();
		this.empId = empId;
		this.empName = empName;
		this.empSal = empSal;
	}

	public int getEmpId() {
		return empId;
	}
	public void setEmpId(int empId) {
		this.empId = empId;
	}
	public String getEmpName() {
		return empName;
	}
	public void setEmpName(String empName) {
		this.empName = empName;
```

```
	}
	public double getEmpSal() {
		return empSal;
	}
	public void setEmpSal(double empSal) {
		this.empSal = empSal;
	}

	public Set<Address> getAddr() {
		return addr;
	}

	public void setAddr(Set<Address> addr) {
		this.addr = addr;
	}

	@Override
	public String toString() {
		return "Employee [empId=" + empId + ", empName=" + empName +
", empSal=" + empSal + ", addr=" + addr + "]";
	}
}
```

child class:-

```
package com.app.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="addrtab")
public class Address {

	@Id
	@Column(name="aid")
	private int addrId;
	@Column(name="loc")
	private String loc;
	@Column(name="pin")
	private int pinCode;

	@ManyToOne
	private Employee emp;

	public Address() {
		super();
	}
```

```
	public Address(int addrId, String loc, int pinCode) {
		super();
		this.addrId = addrId;
		this.loc = loc;
		this.pinCode = pinCode;
	}

	public int getAddrId() {
		return addrId;
	}

	public void setAddrId(int addrId) {
		this.addrId = addrId;
	}

	public String getLoc() {
		return loc;
	}

	public void setLoc(String loc) {
		this.loc = loc;
	}

	public int getPinCode() {
		return pinCode;
	}

	public void setPinCode(int pinCode) {
		this.pinCode = pinCode;
	}

	public Employee getEmp() {
		return emp;
	}

	public void setEmp(Employee emp) {
		this.emp = emp;
	}

	@Override
	public String toString() {
		return "Address [addrId=" + addrId + ", loc=" + loc + ",
pinCode=" + pinCode + ", emp=" + emp + "]";
	}


}
```

hibernate.cfg.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
	<session-factory>
		<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
		<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property
>
		<property name="hibernate.connection.username">root</property>
		<property name="hibernate.connection.password">root</property>
		<property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

		<property name="hibernate.show_sql">true</property>
		<property name="hibernate.format_sql">true</property>
		<property name="hibernate.hbm2ddl.auto">create</property>

		<mapping class="com.app.model.Address" />
		<mapping class="com.app.model.Employee" />
	</session-factory>
</hibernate-configuration>
```

HibernateUtil:-

```
package com.app.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

	private static SessionFactory sf=null;
	static {
		sf=new Configuration()
				.configure()
				.buildSessionFactory();
	}
	public static SessionFactory getSf() {
		return sf;
	}
}
```

Test class:-

```
package com.app.test;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.app.model.Address;
import com.app.model.Employee;
import com.app.util.HibernateUtil;

public class Test {

	public static void main(String[] args) {
		Transaction tx=null;
		try(Session ses=HibernateUtil.getSf().openSession()){
			tx=ses.beginTransaction();


			Address addr1=new Address(101, "HYD", 5000);
			Address addr2=new Address(102, "BAN", 6000);
			Address addr3=new Address(103, "CHN", 7000);

			Employee emp1=new Employee(11,"B",2.2);
			emp1.getAddr().add(addr1);
			emp1.getAddr().add(addr2);
			emp1.getAddr().add(addr3);

			ses.save(emp1);


			tx.commit();
		} catch (Exception e) {
			e.printStackTrace();
		}
	}
}
```

4) Many-to-Many example : Student (1…1) Course:

Parent class:-

```
package com.app.model;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity
@Table(name="stdtab")
public class Student {

	@Id
	@Column(name="sid")
	private int stdId;
	@Column(name="sname")
	private String stdName;
	@Column(name="sfee")
	private double stdFee;

	@ManyToMany(fetch=FetchType.EAGER,cascade=CascadeType.ALL)
	@JoinTable(name="std_crd_tab",
	joinColumns=@JoinColumn(name="sidFk"),
	inverseJoinColumns=@JoinColumn(name="cidFk"))
	private Set<Course> courses=new HashSet<Course>(0);

	public Student() {
		super();
	}
	public Student(int stdId, String stdName, double stdFee) {
		super();
		this.stdId = stdId;
		this.stdName = stdName;
		this.stdFee = stdFee;
	}
	public int getStdId() {
		return stdId;
	}
	public void setStdId(int stdId) {
		this.stdId = stdId;
	}
	public String getStdName() {
```

```
		return stdName;
	}
	public void setStdName(String stdName) {
		this.stdName = stdName;
	}
	public double getStdFee() {
		return stdFee;
	}
	public void setStdFee(double stdFee) {
		this.stdFee = stdFee;
	}
	public Set<Course> getCourses() {
		return courses;
	}
	public void setCourses(Set<Course> courses) {
		this.courses = courses;
	}
	@Override
	public String toString() {
		return "Student [stdId=" + stdId + ", stdName=" + stdName + ",
stdFee=" + stdFee + ", courses=" + courses + "]";
	}


}
```

child class:-

```
package com.app.model;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity
@Table(name="crstab")
public class Course {

	@Id
	@Column(name="cid")
	private int crsId;
	@Column(name="ccode")
	private String crsCode;
	@Column(name="ccost")
	private double crsCost;
```

```
	@ManyToMany(mappedBy="courses")
	private Set<Student> students=new HashSet<Student>(0);

	public Course() {
		super();
	}
	public Course(int crsId, String crsCode, double crsCost) {
		super();
		this.crsId = crsId;
		this.crsCode = crsCode;
		this.crsCost = crsCost;
	}
	public int getCrsId() {
		return crsId;
	}
	public void setCrsId(int crsId) {
		this.crsId = crsId;
	}
	public String getCrsCode() {
		return crsCode;
	}
	public void setCrsCode(String crsCode) {
		this.crsCode = crsCode;
	}
	public double getCrsCost() {
		return crsCost;
	}
	public void setCrsCost(double crsCost) {
		this.crsCost = crsCost;
	}
	public Set<Student> getStudents() {
		return students;
	}
	public void setStudents(Set<Student> students) {
		this.students = students;
	}

	@Override
	public String toString() {
		return "Course [crsId=" + crsId + ", crsCode=" + crsCode + ",
crsCost=" + crsCost + ", students=" + students
				+ "]";
	}

}
```

hibernate.cfg.xml:-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
```

```
<hibernate-configuration>
	<session-factory>
		<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
		<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/test</property
>
		<property name="hibernate.connection.username">root</property>
		<property name="hibernate.connection.password">root</property>
		<property
name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

		<property name="hibernate.show_sql">true</property>
		<property name="hibernate.format_sql">true</property>
		<property name="hibernate.hbm2ddl.auto">create</property>

		<mapping class="com.app.model.Student" />
		<mapping class="com.app.model.Course" />
	</session-factory>
</hibernate-configuration>
```

HibernatUtil :-

```
package com.app.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {

	private static SessionFactory sf=null;
	static {
		sf=new Configuration()
				.configure()
				.buildSessionFactory();
	}
	public static SessionFactory getSf() {
		return sf;
	}
}
```

Test class:-

```
package com.app.test;

import org.hibernate.Session;
import org.hibernate.Transaction;

import com.app.model.Course;
import com.app.model.Student;
import com.app.util.HibernateUtil;
```

```
public class Test {

	public static void main(String[] args) {
		Transaction tx=null;
		try(Session ses=HibernateUtil.getSf().openSession()){
			tx=ses.beginTransaction();


			Course c1=new Course(101, "HIBERNATE", 1500.0);
			Course c2=new Course(102, "SPRING", 2500.0);
			Course c3=new Course(103, "ADV JAVA", 1000.0);


			Student s1=new Student(10, "A", 4000.0);
			s1.getCourses().add(c1);
			s1.getCourses().add(c2);

			Student s2=new Student(11, "B", 3500.0);
			s2.getCourses().add(c2);
			s2.getCourses().add(c3);

			ses.save(s1);
			ses.save(s2);

			tx.commit();
		} catch (Exception e) {
			tx.rollback();
			e.printStackTrace();
		}
	}
}
```

Parent class (Uni-Direction code)	Child class (Bi-Direction code)
@ManyToOne(fetch=FetchType.EAGER,cascade=CascadeType.ALL) @JoinColumn(name="__",unique=true)	@OneToOne(mappedBy="___")
@ManyToOne(fetch=FetchType.EAGER,cascade=CascadeType.ALL)	@OneToOne(mappedBy="___")
@OneToMany(fetch=FetchType.EAGER,cascade=CascadeType.ALL mappedBy=" ")	@ManyToOne
@ManyToMany(fetch=FetchType.EAGER,cascade=CascadeType.ALL)	@ManyToMany(mappedBy=" ")

FB : https://www.facebook.com/groups/thejavatemple/

email : javabyraghu@gmail.com