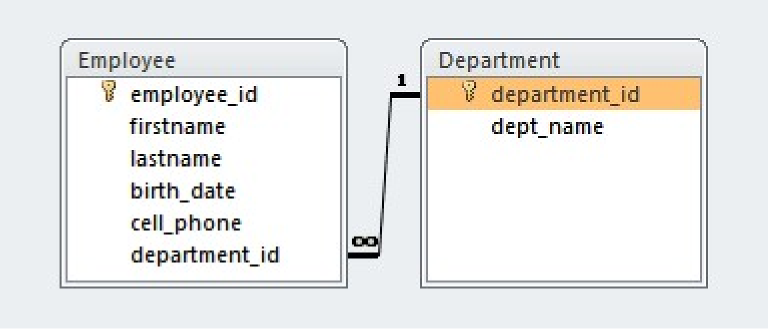
Hibernate Association with Annotation

One To Many

For this example, we will use MySQL database. Create following two tables in MySQL. Note that Employee and Department table exhibits One-to-many relationship. Each Department can be associated with multiple Employees and each Employee can have only one Department.

```
CREATE TABLE `department` (
        `department_id` BIGINT(20) NOT NULL AUTO_INCREMENT,
        `dept_name` VARCHAR(50) NOT NULL DEFAULT '0',
        PRIMARY KEY (`department_id`)
)
```

```
CREATE TABLE `employee` (
        `employee_id` BIGINT(10) NOT NULL AUTO_INCREMENT,
        `firstname` VARCHAR(50) NULL DEFAULT NULL,
        `lastname` VARCHAR(50) NULL DEFAULT NULL,
        `birth_date` DATE NULL DEFAULT NULL,
        `cell_phone` VARCHAR(15) NULL DEFAULT NULL,
        `department_id` BIGINT(20) NULL DEFAULT NULL,
        PRIMARY KEY (`employee_id`),
        INDEX `FK_DEPT` (`department_id`),
        CONSTRAINT `FK_DEPT` FOREIGN KEY (`department_id`) REFERENCES
`department` (`department_id`)
)
```



Department Entity Class:

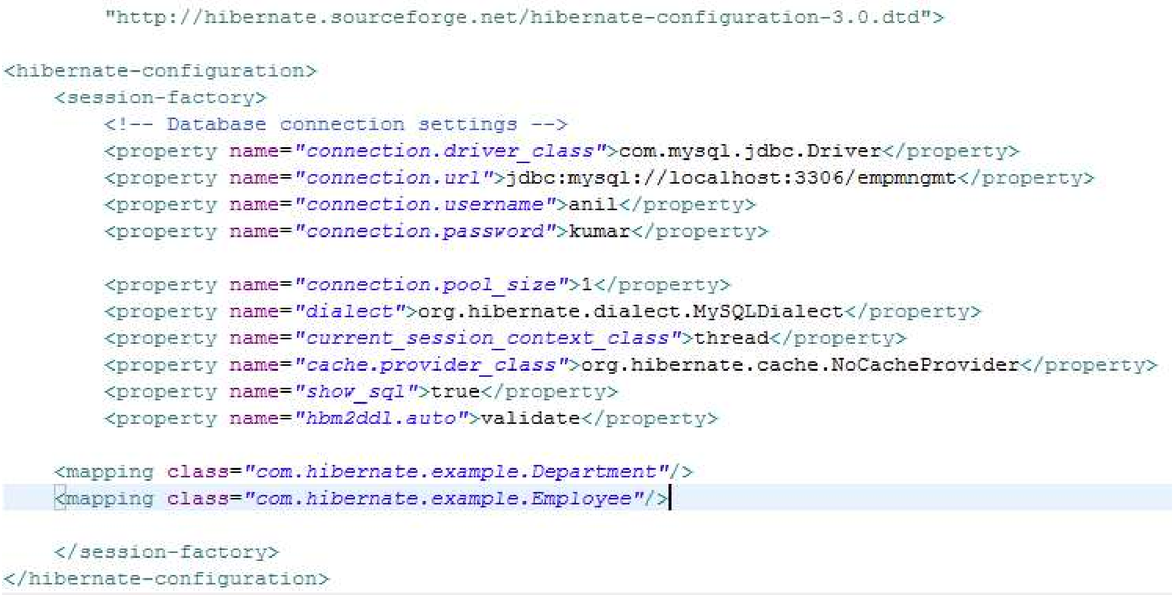
```
package com.hibernate.example;

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

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



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

        @Id
        @GeneratedValue
        @Column(name="department_id")
        private long deptid;

        @Column(name="dept_name")
        private String deptname;

        @OneToMany(mappedBy="department",cascade=CascadeType.ALL)
        private Set<Employee> employee=new HashSet<Employee>();

        //setters and getters
        -----
        -----
}
```

Employee Entity Class:

```
package com.hibernate.example;

import java.sql.Date;

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

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

        @Id
        @GeneratedValue
        @Column(name="employee_id")
        private long empid;

        @Column(name="firstname")
        private String empfirstname;

        @Column(name="lastname")
```

```
    private String emplastname;

    @Column(name="birth_date")
    private Date birthDate;

    @Column(name="cell_phone")
    private String cell_phone;

    @ManyToOne(cascade=CascadeType.ALL)
    @JoinColumn(name="department_id")
    private Department department;


    //setters and getters
    -----
    -----

}
```

Hibernate Configuration File:



```
        "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/empmngmt</property>
        <property name="connection.username">anil</property>
        <property name="connection.password">kumar</property>

        <property name="connection.pool_size">1</property>
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="current_session_context_class">thread</property>
        <property name="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>
        <property name="show_sql">true</property>
        <property name="hbm2ddl.auto">validate</property>

    <mapping class="com.hibernate.example.Department"/>
    <mapping class="com.hibernate.example.Employee"/>

    </session-factory>
</hibernate-configuration>
```

Hibernate One To Many Demo:

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class HibernateOneToManyDemo {

    public static void main(String[] args) {

        SessionFactory sf= HibernateUtil.getSessionFactory();

        Session session=sf.openSession();
        session.beginTransaction();

        Department sales_dept=new Department();
        sales_dept.setDeptname("Sales");

        session.save(sales_dept);

        Employee emp1=new Employee();
        emp1.setEmpfirstname("Amit");
        emp1.setEmplastname("Ahuja");
        emp1.setCell_phone("+918850453234");

        Employee emp2=new Employee();
        emp2.setEmpfirstname("Suman");
        emp2.setEmplastname("Pradha");
        emp2.setCell_phone("+918850453000");

        emp1.setDepartment(sales_dept);
        emp2.setDepartment(sales_dept);

        session.save(emp1);
        session.save(emp2);

        session.getTransaction().commit();
        session.close();
    }
}
```

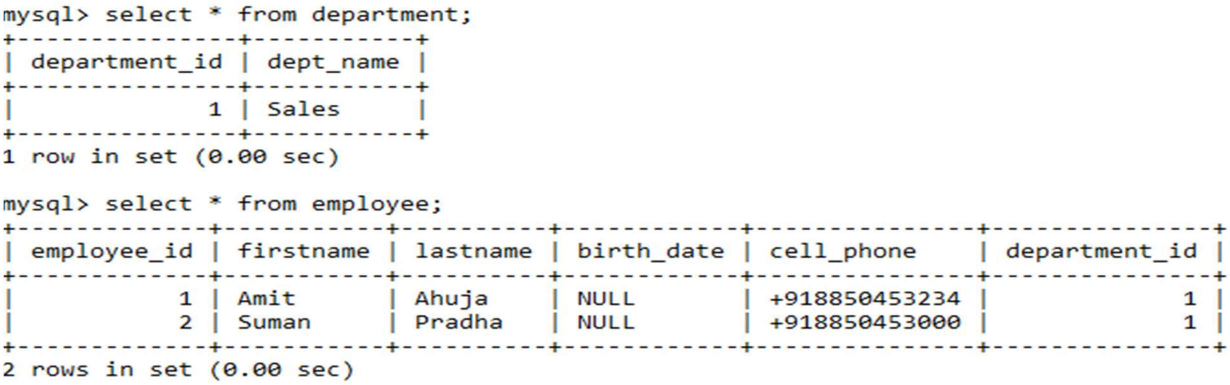
OUTPUT:

Hibernate: insert into department (dept_name) values (?)

Hibernate: insert into employee (birth_date, cell_phone, department_id, firstname, lastname) values (?, ?, ?, ?, ?)

Hibernate: insert into employee (birth_date, cell_phone, department_id, firstname, lastname) values (?, ?, ?, ?, ?)

Database Table:



```
mysql> select * from department;
+---------------+-----------+
| department_id | dept_name |
+---------------+-----------+
|             1 | Sales     |
+---------------+-----------+
1 row in set (0.00 sec)

mysql> select * from employee;
+-------------+-----------+----------+------------+---------------+---------------+
| employee_id | firstname | lastname | birth_date | cell_phone    | department_id |
+-------------+-----------+----------+------------+---------------+---------------+
|           1 | Amit      | Ahuja    | NULL       | +918850453234 |             1 |
|           2 | Suman     | Pradha   | NULL       | +918850453000 |             1 |
+-------------+-----------+----------+------------+---------------+---------------+
2 rows in set (0.00 sec)
```

ONE TO ONE

So we have created two tables “EMPLOYEE” and “EMPLOYEEDETAILS” which have One-to-one relational mapping. These two tables are mapped by primary key Employee_ID.

```sql
CREATE TABLE `employee` (
    `employee_id` BIGINT(10) NOT NULL AUTO_INCREMENT,
    `firstname` VARCHAR(50) NULL DEFAULT NULL,
    `lastname` VARCHAR(50) NULL DEFAULT NULL,
    `birth_date` DATE NOT NULL,
    `cell_phone` VARCHAR(15) NOT NULL,
    PRIMARY KEY (`employee_id`)
)
```

```sql
/* EMPLOYEEDETAIL table */
CREATE TABLE `employeedetail` (
    `employee_id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `street` VARCHAR(50) NULL DEFAULT NULL,
    `city` VARCHAR(50) NULL DEFAULT NULL,
    `state` VARCHAR(50) NULL DEFAULT NULL,
    `country` VARCHAR(50) NULL DEFAULT NULL,
    PRIMARY KEY (`employee_id`),
    CONSTRAINT `FKEMPL` FOREIGN KEY (`employee_id`) REFERENCES `employee`
(`employee_id`)
)
```

Employee Entity Class:

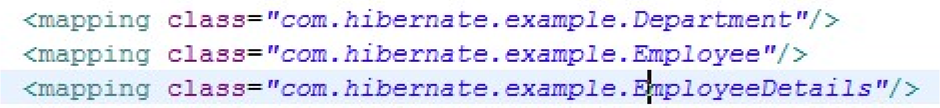
```java
package com.hibernate.example;

import java.sql.Date;

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

@Entity
@Table(name="employee")
public class Employee {
      @Id
      @GeneratedValue
      @Column(name="employee_id")
      private long empid;

      @Column(name="firstname")
      private String empfirstname;

      @Column(name="lastname")
      private String emplastname;

      @Column(name="birth_date")
      private Date birthDate;

      @Column(name="cell_phone")
      private String cell_phone;

      @ManyToOne(cascade=CascadeType.ALL)
      @JoinColumn(name="department_id")
      private Department department;

      // One to One relationship mapping
      @OneToOne(mappedBy="employee",cascade=CascadeType.ALL)
      private EmployeeDetails employee_details;

      //setters and getters
      -----
      -----
}
```

EmployeeDetails Entity Class:

```
package com.hibernate.example;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.PrimaryKeyJoinColumn;
import javax.persistence.Table;

import org.hibernate.annotations.GenericGenerator;
import org.hibernate.annotations.Parameter;

@Entity
@Table(name="employeedetail")
public class EmployeeDetails {

    //Foreign key strategy to map one to one relationship
    @Id
    @Column(name="employee_id")
    @GeneratedValue(generator="gen")
    @GenericGenerator(name="gen",strategy="foreign",
          parameters=@Parameter(name="property",value="employee"))
```

```
	private long empid;

	@Column(name="street")
	private String street;

	@Column(name="city")
	private String city;

	@Column(name="state")
	private String state;

	@Column(name="country")
	private String country;

	//One to one relationship mapping
	@OneToOne(cascade=CascadeType.ALL)
	@PrimaryKeyJoinColumn
	private Employee employee;

	//setters and getters
	----
	----

}
```

Hibernate Configuration File:



```
 <mapping class="com.hibernate.example.Department"/>
 <mapping class="com.hibernate.example.Employee"/>
 <mapping class="com.hibernate.example.EmployeeDetails"/>
```

HibernateOneToOneDemo Class:

```
package com.hibernate.example;

import org.hibernate.Session;
import org.hibernate.SessionFactory;

public class HibernateOneToOneDemo {


	public static void main(String[] args) {

		SessionFactory sf= HibernateUtil.getSessionFactory();
		Session session=sf.openSession();
		session.beginTransaction();

		Department it_dept=new Department();
		it_dept.setDeptname("IT");

		session.save(it_dept);

		Employee emp=new Employee();
		emp.setEmpfirstname("Anil");
		emp.setEmplastname("Kumar");
		emp.setCell_phone("+918860543789");
```

```
        EmployeeDetails empdetails=new EmployeeDetails();
        empdetails.setStreet("4th Cross");
        empdetails.setCity("Bangalore");
        empdetails.setState("Karnataka");
        empdetails.setCountry("India");

        emp.setDepartment(it_dept);
        emp.setEmployee_details(empdetails);
        empdetails.setEmployee(emp);

        session.save(emp);

        session.getTransaction().commit();
        session.close();
    }
}
```

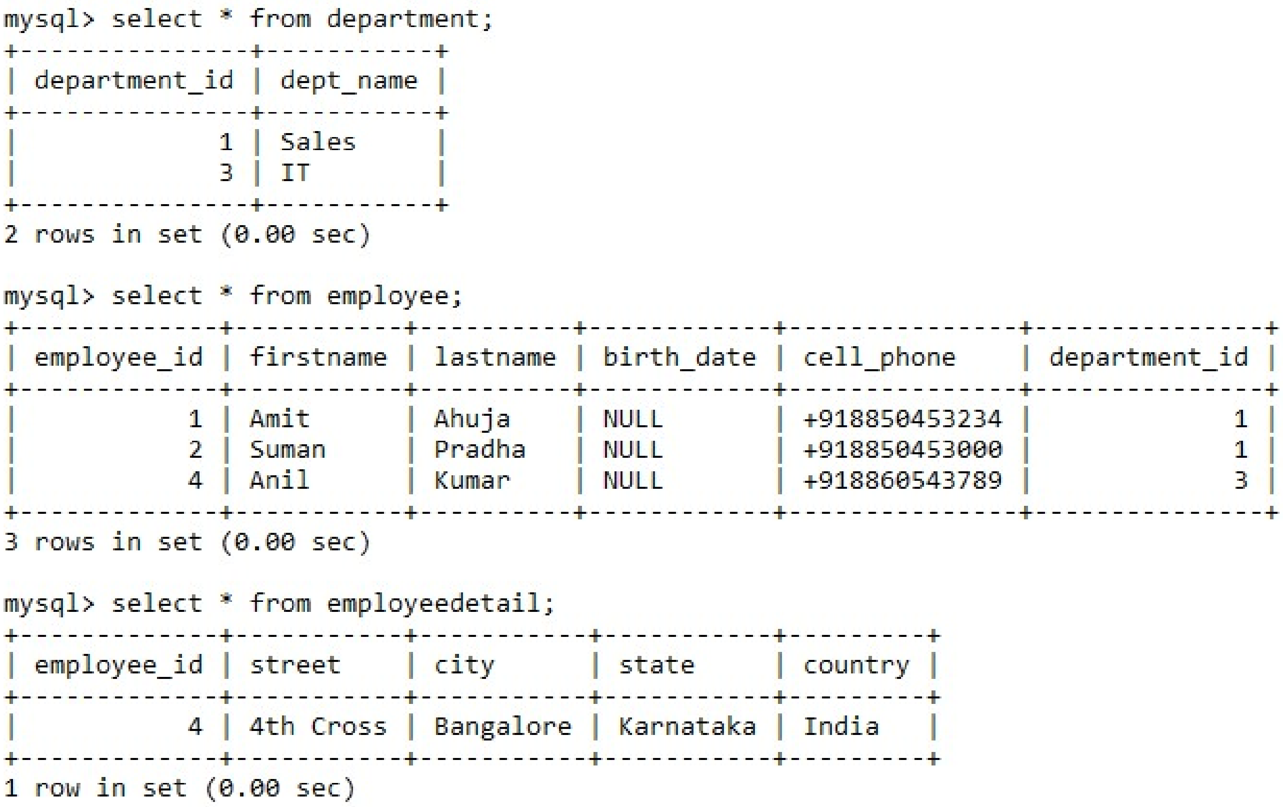
OUTPUT:

Hibernate: insert into department (dept_name) values (?)

Hibernate: insert into employee (birth_date, cell_phone, department_id, firstname, lastname) values (?, ?, ?, ?, ?)

Hibernate: insert into employeedetail (city, country, state, street, employee_id) values (?, ?, ?, ?, ?)

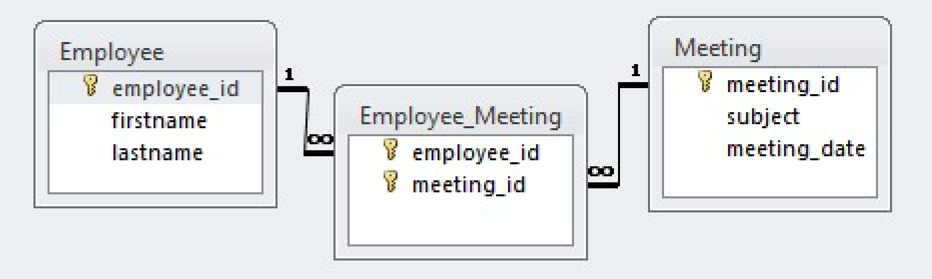
Database Tables:



```
mysql> select * from department;
+---------------+-----------+
| department_id | dept_name |
+---------------+-----------+
|             1 | Sales     |
|             3 | IT        |
+---------------+-----------+
2 rows in set (0.00 sec)

mysql> select * from employee;
+-------------+-----------+----------+------------+---------------+---------------+
| employee_id | firstname | lastname | birth_date | cell_phone    | department_id |
+-------------+-----------+----------+------------+---------------+---------------+
|           1 | Amit      | Ahuja    | NULL       | +918850453234 |             1 |
|           2 | Suman     | Pradha   | NULL       | +918850453000 |             1 |
|           4 | Anil      | Kumar    | NULL       | +918860543789 |             3 |
+-------------+-----------+----------+------------+---------------+---------------+
3 rows in set (0.00 sec)

mysql> select * from employeedetail;
+-------------+-----------+-----------+-----------+---------+
| employee_id | street    | city      | state     | country |
+-------------+-----------+-----------+-----------+---------+
|           4 | 4th Cross | Bangalore | Karnataka | India   |
+-------------+-----------+-----------+-----------+---------+
1 row in set (0.00 sec)
```

MANY TO MANY

For this example, we will MySQL database. We are using Employee-Meeting relationship as a many to many relationship example. Each Employee can attain more than one meetings and each meetings can have more than one employee



```
CREATE TABLE `employee` (
        `employee_id` BIGINT(10) NOT NULL AUTO_INCREMENT,
        `firstname` VARCHAR(50) NULL DEFAULT NULL,
        `lastname` VARCHAR(50) NULL DEFAULT NULL,
        `birth_date` DATE NULL DEFAULT NULL,
        `cell_phone` VARCHAR(15) NULL DEFAULT NULL,
        `department_id` BIGINT(20) NULL DEFAULT NULL,
        PRIMARY KEY (`employee_id`),
        INDEX `FK_DEPT` (`department_id`),
        CONSTRAINT `FK_DEPT` FOREIGN KEY (`department_id`) REFERENCES
`department` (`department_id`)
)
```

```
CREATE TABLE `meeting` (
        `meeting_id` BIGINT(20) NOT NULL AUTO_INCREMENT,
        `subject` VARCHAR(50) NOT NULL,
        `meeting_date` DATE NOT NULL,
        PRIMARY KEY (`meeting_id`)
)
```

```
CREATE TABLE `employee_meeting` (
        `employee_id` BIGINT(20) NOT NULL,
        `meeting_id` BIGINT(20) NOT NULL,
        PRIMARY KEY (`employee_id`, `meeting_id`),
        INDEX `FK_MEETING` (`meeting_id`),
        CONSTRAINT `FK_EMPLOYEE` FOREIGN KEY (`employee_id`) REFERENCES
`employee` (`employee_id`),
        CONSTRAINT `FK_MEETING` FOREIGN KEY (`meeting_id`) REFERENCES `meeting`
(`meeting_id`)
)
```

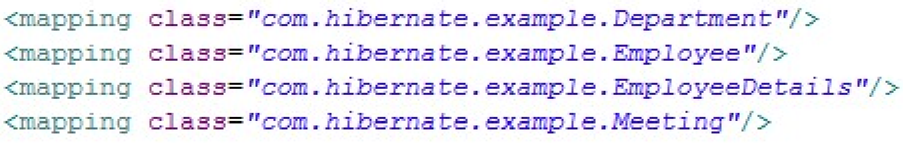
Employee Entity Class:

```java
package com.hibernate.example;

import java.sql.Date;
import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;
import javax.persistence.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.Table;

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

      @Id
      @GeneratedValue
      @Column(name="employee_id")
      private long empid;

      @Column(name="firstname")
      private String empfirstname;

      @Column(name="lastname")
      private String emplastname;

      @Column(name="birth_date")
      private Date birthDate;

      @Column(name="cell_phone")
      private String cell_phone;

      @ManyToOne(cascade=CascadeType.ALL)
      @JoinColumn(name="department_id")
      private Department department;

      @OneToOne(mappedBy="employee",cascade=CascadeType.ALL)
      private EmployeeDetails employee_details;

      //Many to Many relationship mapping with jointable (intermediate table)
      and join columns
      @ManyToMany(cascade=CascadeType.ALL)
      @JoinTable(name="employee_meeting",
            joinColumns={@JoinColumn(name="employee_id")},
            inverseJoinColumns={@JoinColumn(name="meeting_id")})
      private Set<Meeting> meetings=new HashSet<>();

      //setters and getters

      }
```

Meeting Entity Class:

```java
package com.hibernate.example;

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

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

@Entity
@Table(name="meeting")
public class Meeting {

	@Id
	@Column(name="meeting_id")
	@GeneratedValue
	private long meetingid;

	@Column(name="subject")
	private String subject;

	@Column(name="meeting_date")
	private Date meetingDate;

	@ManyToMany(mappedBy="meetings",cascade=CascadeType.ALL)
	private Set<Employee> employees=new HashSet<>();
	//setters and getters

	---
	---
}
```

Hibernate Config File:



```xml
<mapping class="com.hibernate.example.Department"/>
<mapping class="com.hibernate.example.Employee"/>
<mapping class="com.hibernate.example.EmployeeDetails"/>
<mapping class="com.hibernate.example.Meeting"/>
```

HibernateManyToManyDemo Class:

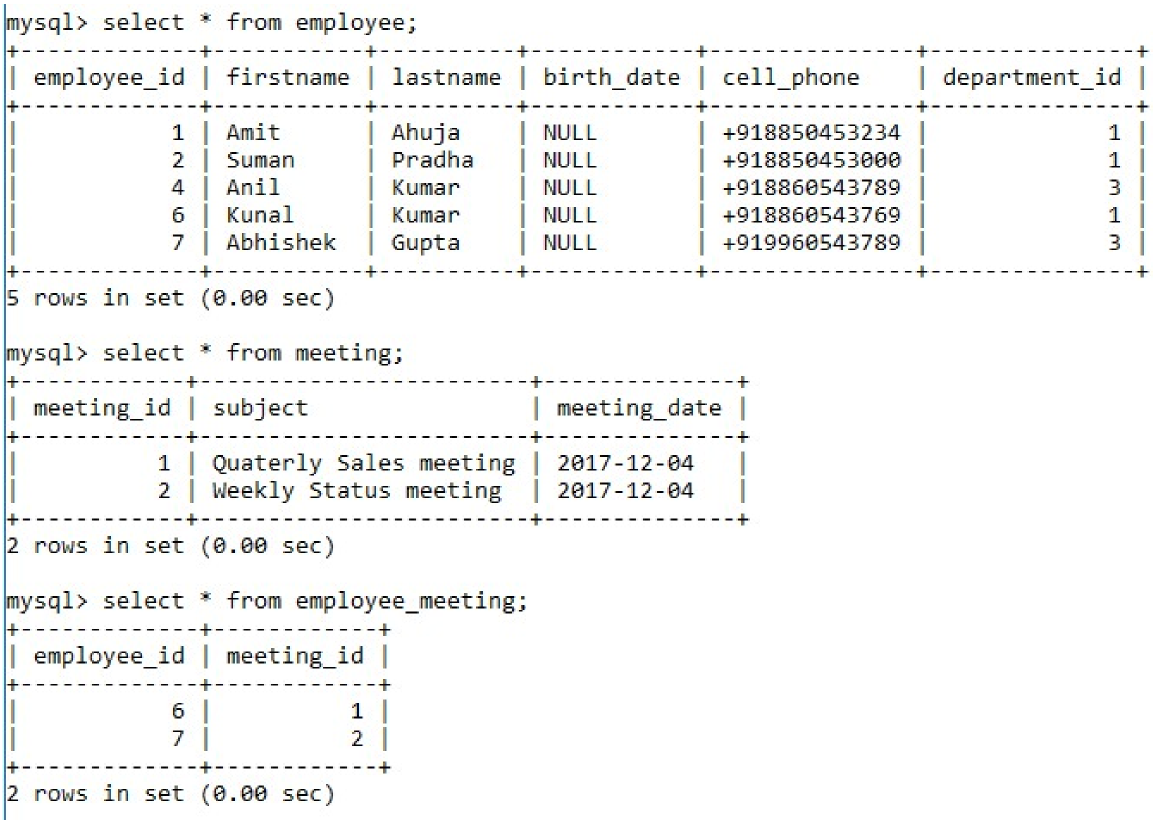
```java
package com.hibernate.example;


import java.util.Date;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
```

```
public class HibernateManyToManyDemo {

      public static void main(String[] args) {

            SessionFactory sf=HibernateUtil.getSessionFactory();
            Session session=sf.openSession();
            session.beginTransaction();

            long deptId=1;
            Department sales_dept=(Department)session.load(Department.class,
deptId);
            deptId=3;
            Department it_dept=(Department)session.load(Department.class,
deptId);

            Employee emp1=new Employee();
            emp1.setEmpfirstname("Kunal");
            emp1.setEmplastname("Kumar");
            emp1.setCell_phone("+918860543769");
            emp1.setDepartment(sales_dept);

            Employee emp2=new Employee();
            emp2.setEmpfirstname("Abhishek");
            emp2.setEmplastname("Gupta");
            emp2.setCell_phone("+919960543789");
            emp2.setDepartment(it_dept);

            Meeting meeting1 = new Meeting();
            meeting1.setSubject("Quaterly Sales meeting");
            meeting1.setMeetingDate(new Date());

        Meeting meeting2 = new Meeting();
        meeting2.setSubject("Weekly Status meeting");
        meeting2.setMeetingDate(new Date());

        emp1.getMeetings().add(meeting1);
        emp2.getMeetings().add(meeting2);

        session.save(emp1);
        session.save(emp2);

        session.getTransaction().commit();
        session.close();
      }
}
```

OUTPUT:

```
Hibernate: insert into employee (birth_date, cell_phone, department_id, firstname, lastname) values (?, ?, ?, ?, ?)
Hibernate: insert into meeting (meeting_date, subject) values (?, ?)
Hibernate: insert into employee (birth_date, cell_phone, department_id, firstname, lastname) values (?, ?, ?, ?, ?)
Hibernate: insert into meeting (meeting_date, subject) values (?, ?)
Hibernate: insert into employee_meeting (employee_id, meeting_id) values (?, ?)
Hibernate: insert into employee_meeting (employee_id, meeting_id) values (?, ?)
```

Database Tables:



```
mysql> select * from employee;
+-------------+-----------+----------+------------+---------------+---------------+
| employee_id | firstname | lastname | birth_date | cell_phone    | department_id |
+-------------+-----------+----------+------------+---------------+---------------+
|           1 | Amit      | Ahuja    | NULL       | +918850453234 |             1 |
|           2 | Suman     | Pradha   | NULL       | +918850453000 |             1 |
|           4 | Anil      | Kumar    | NULL       | +918860543789 |             3 |
|           6 | Kunal     | Kumar    | NULL       | +918860543769 |             1 |
|           7 | Abhishek  | Gupta    | NULL       | +919960543789 |             3 |
+-------------+-----------+----------+------------+---------------+---------------+
5 rows in set (0.00 sec)

mysql> select * from meeting;
+------------+-------------------------+--------------+
| meeting_id | subject                 | meeting_date |
+------------+-------------------------+--------------+
|          1 | Quaterly Sales meeting  | 2017-12-04   |
|          2 | Weekly Status meeting   | 2017-12-04   |
+------------+-------------------------+--------------+
2 rows in set (0.00 sec)

mysql> select * from employee_meeting;
+-------------+------------+
| employee_id | meeting_id |
+-------------+------------+
|           6 |          1 |
|           7 |          2 |
+-------------+------------+
2 rows in set (0.00 sec)
```

HibernateUtil Class:

```
package com.hibernate.example;

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

public class HibernateUtil {

      public static SessionFactory getSessionFactory(){
            Configuration cfg=new
Configuration().configure("hibernate.cfg.xml");
            return cfg.buildSessionFactory();
      }
}
```

Thanks
Anil Kumar