SHIJU THOMAS M.Y
DEPT. OF COMPUTER SCIENCE
RCSS

HN_ONE_TO_ONE

database

```
create database hn_stud_address


create table student (
student_id integer primary key AUTO_INCREMENT,
        student_name varchar(300),
            student_email varchar(200))
create table address(address_id integer primary key,
        address_street varchar(300),
        address_city varchar(300),
        address_state varchar(300),
        address_zipcode varchar(30),
            foreign key(address_id) references student(student_id))
```

---

hibernate.cfg.xml

```
<hibernate-configuration>
 <session-factory>
  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
  <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/hn_stud_address</property>
  <property name="hibernate.connection.username">root</property>
  <mapping resource="Student.hbm.xml"/>
  <mapping resource="Address.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

Student.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 2 Aug, 2019 9:52:12 AM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
  <class name="hn_stud_add.Student" table="student" catalog="hn_stud_address" optimistic-
lock="version">
    <id name="studentId" type="java.lang.Integer">
      <column name="student_id" />
      <generator class="identity" />
    </id>
    <property name="studentName" type="string">
```

```
    <column name="student_name" length="300" />
  </property>
  <property name="studentEmail" type="string">
    <column name="student_email" length="200" />
  </property>
   <one-to-one name="address" cascade="all"></one-to-one>
 </class>
</hibernate-mapping>
```

Address.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<!-- Generated 2 Aug, 2019 9:52:12 AM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
 <class name="hn_stud_add.Address" table="address" catalog="hn_stud_address" optimistic-lock="version">
  <id name="addressId" type="int" column="address_id" >
    <generator class="foreign">
      <param name="property">student</param>
    </generator>
  </id>
  <property name="addressStreet" type="string">
    <column name="address_street" length="300" />
  </property>
  <property name="addressCity" type="string">
    <column name="address_city" length="300" />
```

```
    </property>
    <property name="addressState" type="string">
      <column name="address_state" length="300" />
    </property>
    <property name="addressZipcode" type="string">
      <column name="address_zipcode" length="30" />
    </property>
    <one-to-one name="student"></one-to-one>
  </class>
</hibernate-mapping>
```

Address.java

```
package hn_stud_add;

public class Address  implements java.io.Serializable {


    private int addressId;
    private String addressStreet;
    private String addressCity;
    private String addressState;
    private String addressZipcode;
    private Student student;

   public Address() {
```

```
}

public Address(int addressId) {
    this.addressId = addressId;
}
public Address(int addressId, String addressStreet, String addressCity, String addressState, String addressZipcode) {
    this.addressId = addressId;
    this.addressStreet = addressStreet;
    this.addressCity = addressCity;
    this.addressState = addressState;
    this.addressZipcode = addressZipcode;
}

public int getAddressId() {
    return this.addressId;
}

public void setAddressId(int addressId) {
    this.addressId = addressId;
}
public String getAddressStreet() {
    return this.addressStreet;
}

public void setAddressStreet(String addressStreet) {
    this.addressStreet = addressStreet;
}
```

```
public String getAddressCity() {
   return this.addressCity;
}

public void setAddressCity(String addressCity) {
   this.addressCity = addressCity;
}
public String getAddressState() {
   return this.addressState;
}

public void setAddressState(String addressState) {
   this.addressState = addressState;
}
public String getAddressZipcode() {
   return this.addressZipcode;
}

public void setAddressZipcode(String addressZipcode) {
   this.addressZipcode = addressZipcode;
}

public void setStudent(Student student) {
   this.student = student;
}

public Student getStudent() {
   return student;
}
```

```
}
```

Student.java

```
package comp_map;

public class Student  implements java.io.Serializable {



    private Integer studentId;
    private String studentName;
    private String studentEmail;
    private Address address;

   public Student() {
   }

   public Student(String studentName, String studentEmail) {
      this.studentName = studentName;
      this.studentEmail = studentEmail;
   }

   public Integer getStudentId() {
       return this.studentId;
   }

   public void setStudentId(Integer studentId) {
```

```
    this.studentId = studentId;
  }
  public String getStudentName() {
    return this.studentName;
  }

  public void setStudentName(String studentName) {
    this.studentName = studentName;
  }
  public String getStudentEmail() {
    return this.studentEmail;
  }

  public void setStudentEmail(String studentEmail) {
    this.studentEmail = studentEmail;
  }

  public void setAddress(Address address) {
    this.address = address;
  }

  public Address getAddress() {
    return address;
  }
}
```

---

main

```
package hn_stud_add;


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



public class Hn_stud_add {


  /**
   * @param args the command line arguments
   */
   private static SessionFactory factory;
  public static void main(String[] args) {
     // TODO code application logic here


     try {
      factory =new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();
    } catch (Throwable ex) {
      System.err.println("Failed to create sessionFactory object." + ex);
      throw new ExceptionInInitializerError(ex);
    }
     Session session = factory.openSession();
                    Transaction transaction = null;
                    try {
                            transaction = session.beginTransaction();
```

```
                Address address = new Address();
            address.setAddressStreet("rajagiri");
            address.setAddressCity("Cochin");
            address.setAddressState("Kerala");
            address.setAddressZipcode("682104");
                Student student = new Student();
            student.setStudentName("shiju");
            student.setStudentEmail("one@one.com");
            student.setAddress(address);
            address.setStudent(student);
                session.save(student);
                session.save(address);
                transaction.commit();
            } catch (Exception e) {
                transaction.rollback();
                e.printStackTrace();
            } finally {
                session.close();
            }

    System.exit(0);
   }

}


/*
//////////////////////
public static void main(String[] args) {
```

```
    // TODO code application logic here


     try {
      factory =new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();
    } catch (Throwable ex) {
      System.err.println("Failed to create sessionFactory object." + ex);
      throw new ExceptionInInitializerError(ex);
    }
     Session session = factory.openSession();
     List list = session.createQuery("FROM Student").list();



   Iterator<Student> itr=list.iterator();
   while(itr.hasNext()){
    Student st=itr.next();
    System.out.print(st.getStudentId()+" "+st.getStudentName()+" "+st.getStudentEmail() + "  " );
    Address address=st.getAddress();
    System.out.println(" "+address.getAddressCity()+" "+
      address.getAddressStreet()+" "+address.getAddressState()+" "+address.getAddressZipcode());
   }


   session.close();
    System.exit(0);
   }
*/
```