SHIJU THOMAS M.Y
DEPT. OF COMPUTER SCIENCE
RCSS

Employee Management System – Using Reverse Engineering

Data base Part

```
create database empms
create table emp (eno integer AUTO_INCREMENT PRIMARY key ,
        ename varchar(30),
        esal float)
insert into emp (ename , esal) values('abcd',10000)
select * from emp
commit
```

Project : Emp_Ms

hibernate.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<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/empms?zeroDateTimeBehavior=con
vertToNull</property>
  <property name="hibernate.connection.username">root</property>
  <mapping resource="emp_ms/hibernate.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

hibernate.reveng.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse Engineering
DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-reverse-engineering-3.0.dtd">
<hibernate-reverse-engineering>
 <schema-selection match-catalog="empms"/>
 <table-filter match-name="emp"/>
</hibernate-reverse-engineering>
```

Emp.java

```
public class Emp  implements java.io.Serializable {
    private Integer eno;
    private String ename;
    private Float esal;
   public Emp() {
   }
   public Emp(String ename, Float esal) {
     this.ename = ename;
     this.esal = esal;
```

```
    }
    public Integer getEno() {
        return this.eno;
    }
    public void setEno(Integer eno) {
        this.eno = eno;
    }
    public String getEname() {
        return this.ename;
    }
    public void setEname(String ename) {
        this.ename = ename;
    }
    public Float getEsal() {
        return this.esal;
    }
    public void setEsal(Float esal) {
        this.esal = esal;
    }
}
```

hibernate.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 1 Jul, 2020 11:53:28 AM by Hibernate Tools 4.3.1 -->
<hibernate-mapping>
    <class name="emp_ms_rev.Emp" table="emp" catalog="empms" optimistic-lock="version">
        <id name="eno" type="java.lang.Integer">
```

```
        <column name="eno" />
        <generator class="identity" />
      </id>
      <property name="ename" type="string">
        <column name="ename" length="30" />
      </property>
      <property name="esal" type="java.lang.Float">
        <column name="esal" precision="12" scale="0" />
      </property>
    </class>
</hibernate-mapping>
```

Emp_Ms_Rev.java

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
/**
 *
 * @author Administrator
 */
public class Emp_Ms_Rev {

    private static SessionFactory factory;

    public static void init()
    {
        try{
```

```
      factory = new Configuration().configure().buildSessionFactory();
    }
    catch (Throwable ex) {
      System.err.println("Failed to create sessionFactory object." + ex);
      throw new ExceptionInInitializerError(ex);
    }
}

public static void insert(String na,float s)
{

    Session session = factory.openSession();
    Transaction tx = null;
    try {
     tx = session.beginTransaction();
     Emp e = new Emp();
          e.setEname(na);
          e.setEsal(s);
     session.save(e);
          tx.commit();
     System.out.println("Successfully inserted");
          factory.close();
    }
    catch(Exception e)
    {
      System.out.println(e);
    }
}
```

```
    public static void main(String[] args) {
        // TODO code application logic here
        init();
        insert("test",15000);
    }

}
```