

```
1 package model;
2
3 import jakarta.persistence.*;
4
5 @Entity
6 @Table(name="USER")
7 public class User {
8     @Column(name="USER_ID")
9     @Id
10    @GeneratedValue(strategy = GenerationType.
IDENTITY)
11    private Integer id;
12    private String fullName;
13    private String email;
14    private String password;
15    private int age;
16    private double salary;
17    private String city;
18
19    public User(String fullName, String email, String
password, int age, double salary, String city) {
20        this.fullName = fullName;
21        this.email = email;
22        this.password = password;
23        this.age = age;
24        this.salary = salary;
25        this.city = city;
26    }
27
28    public User(){
29
30    }
31
32    public Integer getId() {
33        return id;
34    }
35
36    public void setId(Integer id) {
37        this.id = id;
38    }
39
```

```
40     public String getFullName() {
41         return fullName;
42     }
43
44     public void setFullName(String fullName) {
45         this.fullName = fullName;
46     }
47
48     public String getEmail() {
49         return email;
50     }
51
52     public void setEmail(String email) {
53         this.email = email;
54     }
55
56     public String getPassword() {
57         return password;
58     }
59
60     public void setPassword(String password) {
61         this.password = password;
62     }
63
64     public int getAge() {
65         return age;
66     }
67
68     public void setAge(int age) {
69         this.age = age;
70     }
71
72     public double getSalary() {
73         return salary;
74     }
75
76     public void setSalary(double salary) {
77         this.salary = salary;
78     }
79
80     public String getCity() {
```

```
81         return city;
82     }
83
84     public void setCity(String city) {
85         this.city = city;
86     }
87 }
88
```

```
1 package controller;
2
3 import model.User;
4 import org.hibernate.Session;
5 import org.hibernate.SessionFactory;
6 import org.hibernate.Transaction;
7 import org.hibernate.cfg.Configuration;
8
9 public class UserController {
10     public static void main(String[] args) {
11         SessionFactory factory = new Configuration().
            configure().buildSessionFactory();
12         Session session = factory.openSession();
13         //         addUser(factory,session);
14         //         findUser(factory,session,3);
15         //
16         //         updateUser(session,3);
17         //
18         deleteUser(session,4);
19
20         factory.close();
21         session.close();
22     }
23     public static void addUser(SessionFactory factory
24     , Session session){
25         Transaction transaction = session.
            beginTransaction();
26         User uOne = new User();
27         uOne.setEmail("haseeb@gmail.com");
28         uOne.setFullName("Moh Haseeb");
29         uOne.setPassword("has123");
30         uOne.setSalary(2000.69);
31         uOne.setAge(20);
32         uOne.setCity("NYC");
33
34         User uTwo = new User();
35         uTwo.setEmail("James@gmail.com");
36         uTwo.setFullName("James Santana");
37         uTwo.setPassword("James123");
38         uTwo.setSalary(2060.69);
39         uTwo.setAge(25);
```

```
39         uTwo.setCity("Dallas");
40
41         User uThree = new User();
42         uThree.setEmail("Shahparan@gmail.com");
43         uThree.setFullName("AH Shahparan");
44         uThree.setPassword("Shahparan123");
45         uThree.setSalary(3060.69);
46         uThree.setAge(30);
47         uThree.setCity("Chicago");
48
49         User uFour = new User("Christ", "christ@gmail
.com", "147852", 35, 35000.3, "NJ");
50         User uFive = new User("Sid", "Sid", "s258",
29, 4000.36, "LA");
51
52         session.persist(uOne);
53         session.persist(uTwo);
54         session.persist(uThree);
55         session.persist(uFour);
56         session.persist(uFive);
57
58         transaction.commit();
59         System.out.println("successfully saved");
60         factory.close();
61         session.close();
62
63
64     }
65
66     public static void findUser(SessionFactory
factory, Session session, int userId){
67         Transaction tx = session.beginTransaction();
68
69         User u = session.get(User.class, userId);
70         System.out.println("FullName: " + u.
getFullName());
71         System.out.println("Email: " + u.getEmail());
72         System.out.println("password: " + u.
getPassword());
73         tx.commit();
74         factory.close();
```

```
75         session.close();
76     }
77     public static void updateUser(Session session,
78     int userId) {
79         // Todo comment out findUser method and
80         uncomment updateUser method
81         Transaction tx = session.beginTransaction();
82         User u = new User();
83         u.setId(userId);
84         u.setEmail("mhaseeb@perscholas");
85         u.setFullName("M Haseeb");
86         u.setPassword("123456");
87         session.merge(u);
88         session.getTransaction().commit();
89         session.close();
90     }
91     public static void deleteUser(Session session,
92     int userId){
93         // Todo comment out updateUser method and
94         uncomment deleteUser method
95         SessionFactory factory = new Configuration
96         ().configure().buildSessionFactory();
97         Transaction tx = session.beginTransaction();
98         User u = new User();
99         u.setId(userId);
100        session.remove(u);
101        tx.commit();
102        session.close();
103        factory.close();
104    }
105 }
106
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE hibernate-configuration PUBLIC
3     "-//Hibernate/Hibernate Configuration DTD 3.0
4     //EN"
5     "http://www.hibernate.org/dtd/hibernate-
6     configuration-3.0.dtd">
7 <hibernate-configuration>
8     <session-factory>
9         <!-- Drop and re-create the database on
10         startup -->
11         <property name="hibernate.hbm2ddl.auto">
12             update </property>
13
14         <!-- Database connection settings -->
15         <property name="connection.driver_class">com.
16             mysql.cj.jdbc.Driver</property>
17         <property name="connection.url">jdbc:mysql://
18             localhost:3306/usersDb?createDatabaseIfNotExist=true
19             </property>
20         <property name="connection.username">root</
21             property>
22         <property name="connection.password">
23             SzhengSQL123</property>
24
25         <!-- MySQL DB dialect -->
26         <property name="dialect">org.hibernate.
27             dialect.MySQLDialect</property>
28         <!-- print all executed SQL on console -->
29         <property name="hibernate.show_sql" >true </
30             property>
31         <property name="hibernate.format_sql" >true
32             </property>
33
34         <!-- Mapping entity file -->
35         <mapping class="model.User"/>
36
37     </session-factory>
38 </hibernate-configuration>
```