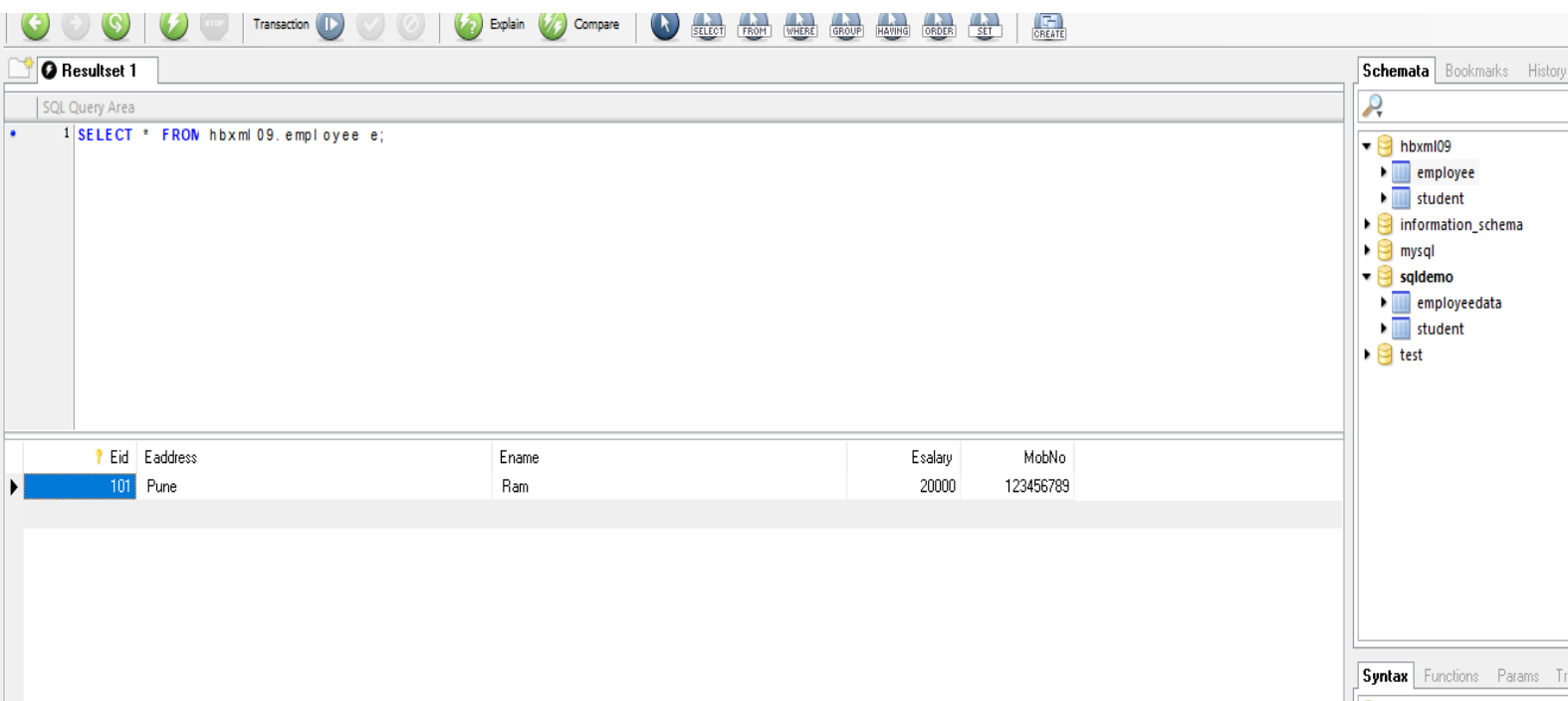


Add employee data to database using hibernate xml base

Before applying annotations except @Entity and @Id-



The screenshot shows a database query tool interface. The top toolbar includes buttons for navigation and execution. The 'SQL Query Area' contains the query: `SELECT * FROM hbxml09.employee e;`. The 'Resultset 1' tab is active, displaying a table with the following data:

Eid	Eaddress	Ename	Esalary	MobNo
101	Pune	Ram	20000	123456789

The right sidebar shows the 'Schemata' view with a tree structure of databases and tables:

- hbxml09
 - employee
 - student
- information_schema
- mysql
- sqldemo
 - employeeedata
 - student
- test

Entity Class-

```
1 package com.tech;
2
3 import javax.persistence.Column;
4
14 @Setter
15 @Getter
16 @Data
17 @Entity
18
19 @Table(name="Employee_Table")
20 public class Employee {
21
22     @Id
23     @GeneratedValue(strategy = GenerationType.AUTO)
24     @Column(name="Emp_Id")
25     private int Eid;
26
27     @Column(name="Emp_Name")
28     private String Ename;
29
30     private String Eaddress;
31
32     @Transient
33     private long MobNo;
34
35     private long Esalary;
36
37
38 }
39
```

XML file-

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <hibernate-configuration>
3 <session-factory>
4
5 <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
6 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hbxml09</property>
7 <property name="hibernate.connection.username">root</property>
8 <property name="hibernate.connection.password">root</property>
9
10 <property name="hibernate.hbm2ddl.auto">update</property>
11 <property name="hibernate.dialect">org.hibernate.dialect.MySQL55Dialect</property>
12 <property name="hibernate.show_sql">true</property>
13 <mapping class="com.tech.Employee"></mapping>
14
15 </session-factory>
16 </hibernate-configuration>
```

Main Class-

```
1 package com.tech;
2
3 import org.hibernate.Session;
4
5
6
7
8
9
10
11 public class Main_Employee {
12     public static void main(String[] args) {
13         StandardServiceRegistry registry=new StandardServiceRegistryBuilder().configure().build();
14         MetadataSources ms=new MetadataSources(registry);
15         Metadata md = ms.getMetadataBuilder().build();
16         SessionFactory sf = md.getSessionFactoryBuilder().build();
17         Session session = sf.openSession();
18
19         Employee e=new Employee();
20         e.setEname("Yogesh");
21         e.setEaddress("SambhajiNagar");
22         e.setMobNo(76838289);
23         e.setEsalary(22000);
24
25         session.save(e);
26         session.beginTransaction().commit();
27         session.close();
28         sf.close();
29         System.out.println("save");
30     }
31 }
32
33
```

Database output-

@Transient annotation apply after 5th employee data for MobNo

SQL Query Area					
1 SELECT * FROM hbxml 09. employee_table e;					
Emp_Id	Eaddress	Emp_Name	Esalary	MobNo	
1	Pune	Ram	20000	123456789	
2	Mumbai	Ramesh	25000	123456789	
3	Nagpur	Amol	30000	123456789	
4	Nashik	Suraj	15000	5638289	
5	Sambhajinagar	Yogesh	22000	76838289	
6	Sambhajinagar	Yogesh	22000	0	
7	Sambhajinagar	Yogesh	22000	0	