

**COURSE ID: 20TS3103AA**

**COURSE NAME: JAVA FULL STACK DEVELOPMENT + MICROSERVICES**

**LAB:03**

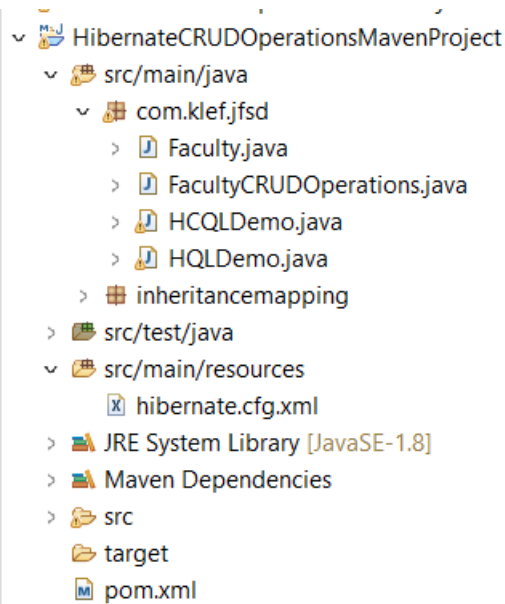
**ID:2000030185**

**NAME: SUPRITH CH**

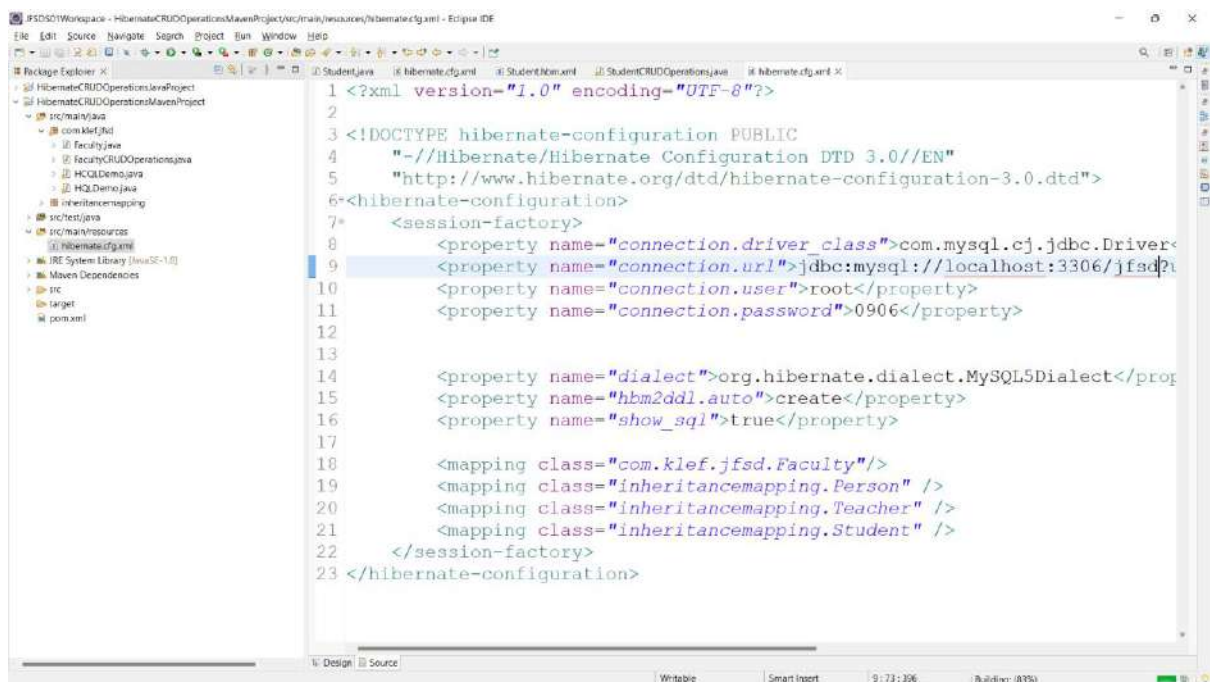
**QUESTION:**

Perform the following tasks by taking reference from Lab 2 with Hibernate Query Language (HQL) and Hibernate Criteria Query Language (HCQL).

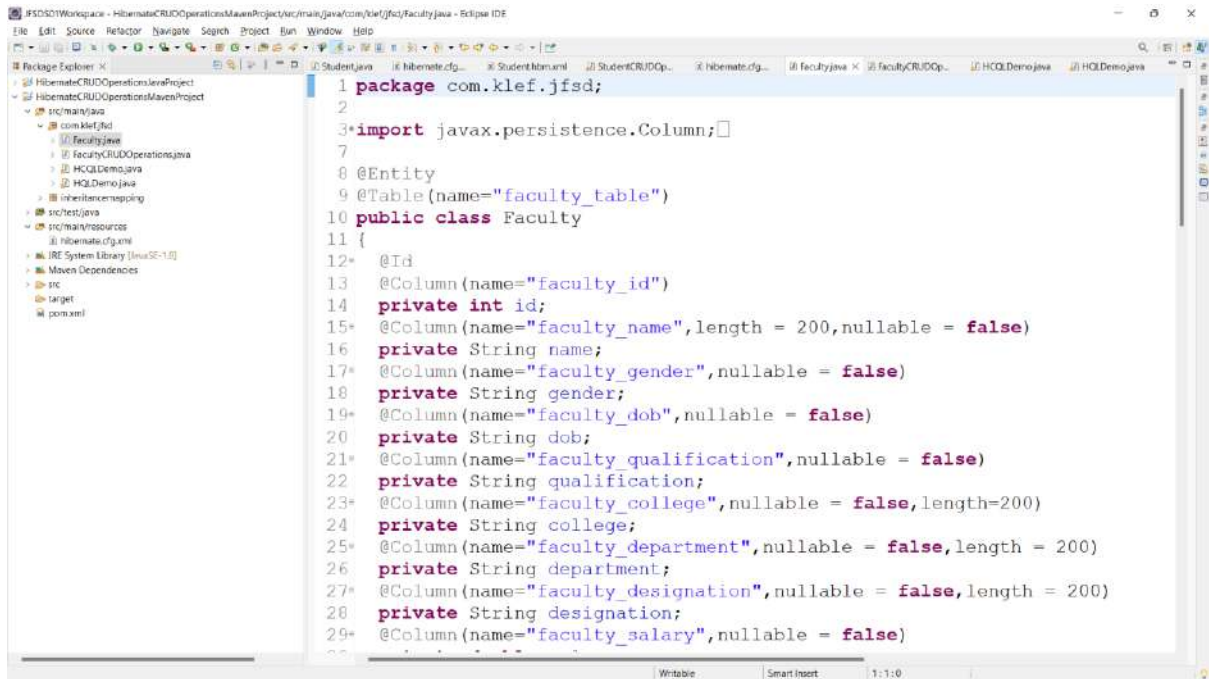
1. HQL query to display all student records with all columns.
2. HQL query to display all student records with specific columns.
3. HQL query to display only names of all students whose CGPA is greater than 7.
4. HQL query to delete a Student whose Student ID is 30101 using input parameter (?) as well as named parameter (:)
5. HQL query to update few details of a student whose Student ID is 30102 using input parameter (?) as well as named parameter (:)
6. HQL query to perform aggregate functions like count (\*), sum(), avg(), min(), max() functions on CGPA column.
7. HCQL query to display specific columns from set of student records.
8. HCQL query to get 5th to 10th records.
9. HCQL query to apply greater than, less than, greater than equal to, less than equal to, equal to and not equal to on CGPA column.
10. HCQL query to get the records in ascending as well as descending order based on Student Name.



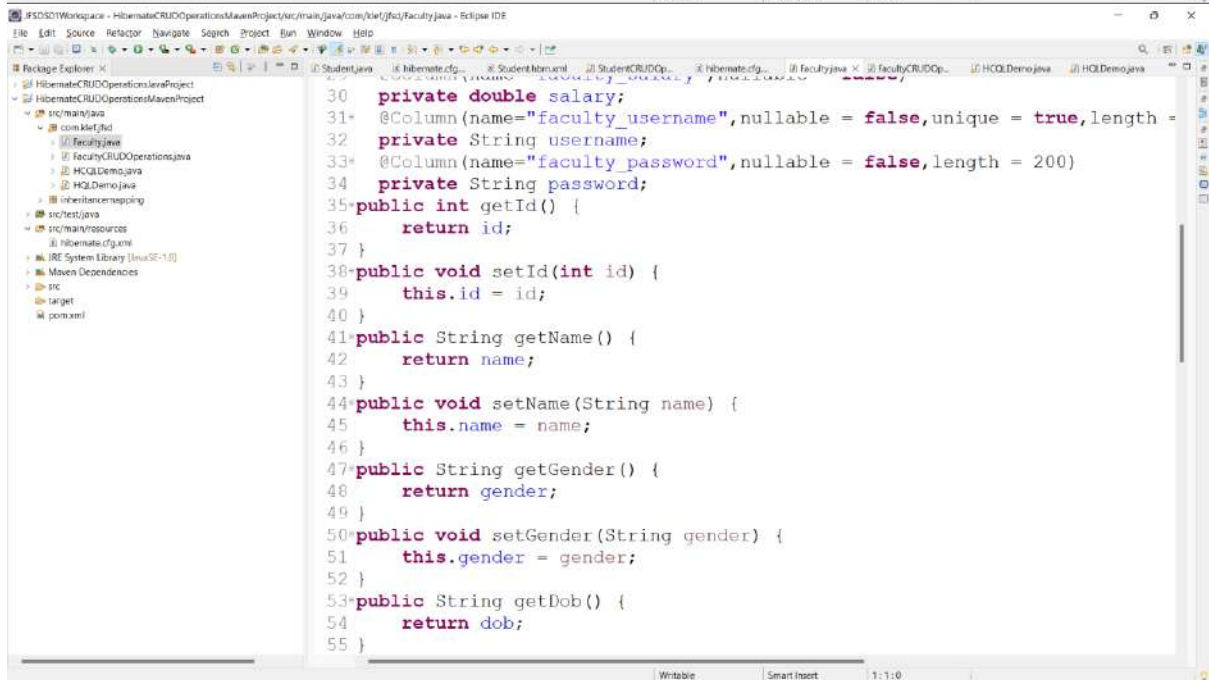
## Hibernate.cfg.xml



Faculty.java



```
1 package com.klef.jfsd;
2
3 import javax.persistence.Column;
4
5 @Entity
6 @Table(name="faculty_table")
7 public class Faculty
8 {
9     @Id
10    @Column(name="faculty_id")
11    private int id;
12    @Column(name="faculty_name",length = 200,nullable = false)
13    private String name;
14    @Column(name="faculty_gender",nullable = false)
15    private String gender;
16    @Column(name="faculty_dob",nullable = false)
17    private String dob;
18    @Column(name="faculty_qualification",nullable = false)
19    private String qualification;
20    @Column(name="faculty_college",nullable = false,length=200)
21    private String college;
22    @Column(name="faculty_department",nullable = false,length = 200)
23    private String department;
24    @Column(name="faculty_designation",nullable = false,length = 200)
25    private String designation;
26    @Column(name="faculty_salary",nullable = false)
```

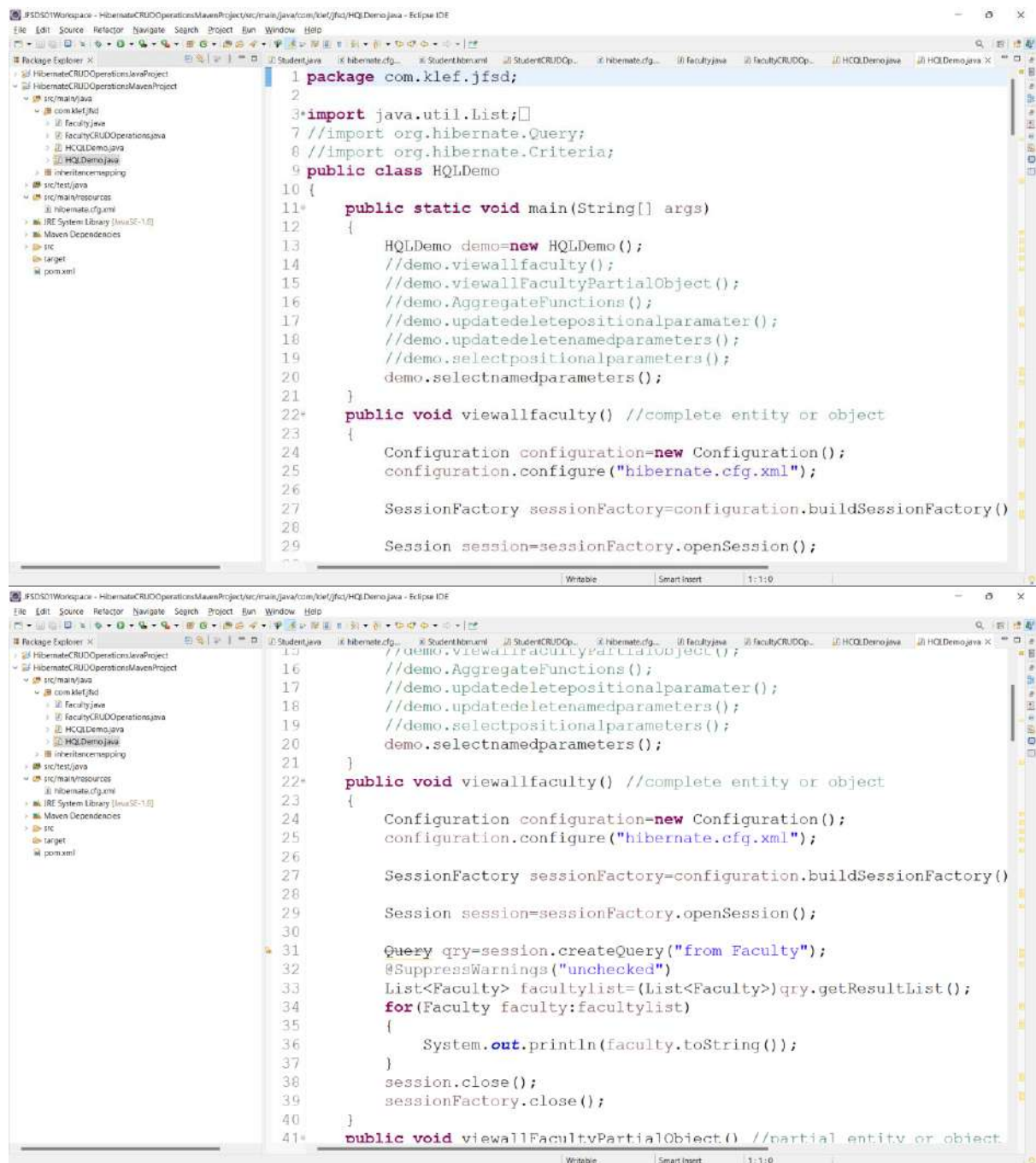


```
27    private double salary;
28    @Column(name="faculty_username",nullable = false,unique = true,length =
29    private String username;
30    @Column(name="faculty_password",nullable = false,length = 200)
31    private String password;
32    public int getId() {
33        return id;
34    }
35    public void setId(int id) {
36        this.id = id;
37    }
38    public String getName() {
39        return name;
40    }
41    public void setName(String name) {
42        this.name = name;
43    }
44    public String getGender() {
45        return gender;
46    }
47    public void setGender(String gender) {
48        this.gender = gender;
49    }
50    public String getDob() {
51        return dob;
52    }
```

```
59 public String getQualification() {
60     return qualification;
61 }
62 public void setQualification(String qualification) {
63     this.qualification = qualification;
64 }
65 public String getCollege() {
66     return college;
67 }
68 public void setCollege(String college) {
69     this.college = college;
70 }
71 public String getDepartment() {
72     return department;
73 }
74 public void setDepartment(String department) {
75     this.department = department;
76 }
77 public String getDesignation() {
78     return designation;
79 }
80 public void setDesignation(String designation) {
81     this.designation = designation;
82 }
83 public double getSalary() {
84     return salary;
85 }
```

```
83 public double getSalary() {
84     return salary;
85 }
86 public void setSalary(double salary) {
87     this.salary = salary;
88 }
89 public String getUsername() {
90     return username;
91 }
92 public void setUsername(String username) {
93     this.username = username;
94 }
95 public String getPassword() {
96     return password;
97 }
98 public void setPassword(String password) {
99     this.password = password;
100 }
101 @Override
102 public String toString() {
103     return "Faculty [id=" + id + ", name=" + name + ", gender=" + gender
104         + qualification + ", college=" + college + ", department=" +
105         + ", salary=" + salary + ", username=" + username + ", passwo
106 }
107 }
108 }
```

## HQLDemo.java



```
1 package com.klef.jfsd;
2
3 import java.util.List;
4 //import org.hibernate.Query;
5 //import org.hibernate.Criteria;
6 public class HQLDemo
7 {
8     public static void main(String[] args)
9     {
10         HQLDemo demo=new HQLDemo();
11         //demo.viewallfaculty();
12         //demo.viewallFacultyPartialObject();
13         //demo.AggregateFunctions();
14         //demo.updatedeletepositionalparamater();
15         //demo.updatedeletenamedparameters();
16         //demo.selectpositionalparameters();
17         demo.selectnamedparameters();
18     }
19     public void viewallfaculty() //complete entity or object
20     {
21         Configuration configuration=new Configuration();
22         configuration.configure("hibernate.cfg.xml");
23
24         SessionFactory sessionFactory=configuration.buildSessionFactory();
25
26         Session session=sessionFactory.openSession();
27
28         //demo.viewallFacultyPartialObject();
29         //demo.AggregateFunctions();
30         //demo.updatedeletepositionalparamater();
31         //demo.updatedeletenamedparameters();
32         //demo.selectpositionalparameters();
33         demo.selectnamedparameters();
34     }
35     public void viewallFacultyPartialObject() //partial entity or object
36     {
37         Configuration configuration=new Configuration();
38         configuration.configure("hibernate.cfg.xml");
39
40         SessionFactory sessionFactory=configuration.buildSessionFactory();
41
42         Session session=sessionFactory.openSession();
43
44         Query qry=session.createQuery("from Faculty");
45         @SuppressWarnings("unchecked")
46         List<Faculty> facultylist=(List<Faculty>)qry.getResultList();
47         for(Faculty faculty:facultylist)
48         {
49             System.out.println(faculty.toString());
50         }
51         session.close();
52         sessionFactory.close();
53     }
54 }
```



```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
hibernateCRUDOperationsMavenProject
src/main/java
com.kiet.jtd
Faculty.java
FacultyCRUDOperations.java
HQLDemo.java
HQLDemo.java
src/test/java
InheritanceMapping
src/main/resources
hibernate.cfg.xml
JRE System Library [JnuSE-1.0]
Maven Dependencies
src
target
pom.xml

36      System.out.println(faculty.toString());
37    }
38    session.close();
39    sessionFactory.close();
40  }
41+ public void viewallFacultyPartialObject() //partial entity or object
42  {
43    Configuration configuration=new Configuration();
44    configuration.configure("hibernate.cfg.xml");
45
46    SessionFactory sessionFactory=configuration.buildSessionFactory()
47
48    Session session=sessionFactory.openSession();
49
50    Query qry=session.createQuery("select f.id,f.name,f.designation,f
51    List<Object[]> facultylist=qry.getResultList();
52    System.out.println("Total Records="+facultylist.size());
53    for(Object[] obj:facultylist)
54    {
55      System.out.println("Faculty ID: "+obj[0]);
56      System.out.println("Faculty Name: "+obj[1]);
57      System.out.println("Faculty Designation: "+obj[2]);
58      System.out.println("Faculty Salary: "+obj[3]);
59      System.out.println("Faculty Username: "+obj[4]);
60    }
61  }
```

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
hibernateCRUDOperationsMavenProject
src/main/java
com.kiet.jtd
Faculty.java
FacultyCRUDOperations.java
HQLDemo.java
HQLDemo.java
src/test/java
InheritanceMapping
src/main/resources
hibernate.cfg.xml
JRE System Library [JnuSE-1.0]
Maven Dependencies
src
target
pom.xml

65+ public void AggregateFunctions()
66  {
67    Configuration configuration=new Configuration();
68    configuration.configure("hibernate.cfg.xml");
69
70    SessionFactory sessionFactory=configuration.buildSessionFactory()
71
72    Session session=sessionFactory.openSession();
73
74    Query qry1=session.createQuery("select count(*) from Faculty");
75    List list1=qry1.list();
76    System.out.println("****count(*)****");
77    System.out.println("Total records count= "+list1.get(0));
78    Query qry2=session.createQuery("select min(salary) from Faculty");
79    List list2=qry2.list();
80    System.out.println("****min(salary)****");
81    System.out.println("Minimum Salary among all faculties= "+list2.get(0));
82    Query qry3=session.createQuery("select max(salary) from Faculty");
83    List list3=qry3.list();
84    System.out.println("****max(salary)****");
85    System.out.println("Maximum Salary among all faculties= "+list3.get(0));
86    Query qry4=session.createQuery("select sum(salary) from Faculty");
87    List list4=qry4.list();
88    System.out.println("****sum(salary)****");
89    System.out.println("sum of Salary of all faculties= "+list4.get(0));
90  }
```

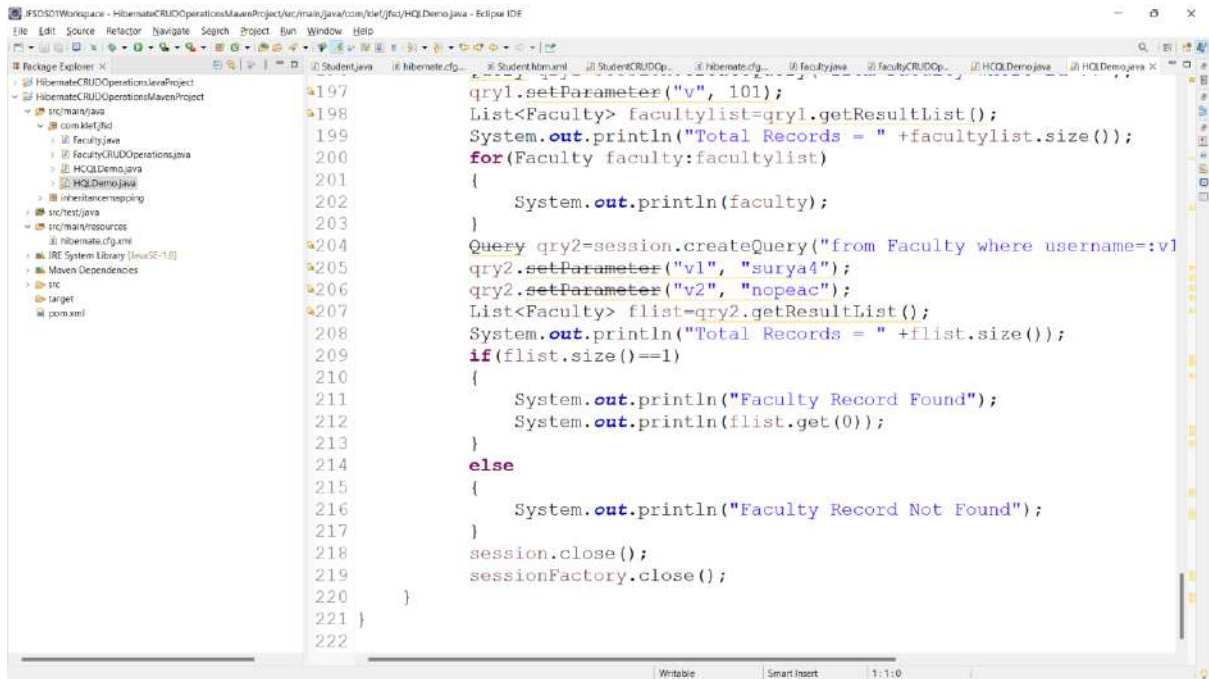
```
98 }
99 public void updatedeletepositionalparamater()
100 {
101     Configuration configuration=new Configuration();
102     configuration.configure("hibernate.cfg.xml");
103
104     SessionFactory sessionFactory=configuration.buildSessionFactory()
105
106     Session session=sessionFactory.openSession();
107
108     Transaction transaction=session.beginTransaction();
109
110     Query qry1=session.createQuery("update Faculty set name=?1,salary
111     qry1.setString(1, "Hari Kiran Vege");
112     qry1.setDouble(2, 67000.50);
113     qry1.setInteger(3, 101);
114     int count1=qry1.executeUpdate();
115     System.out.println(count1 + " records updated succesfully");
116
117     Query qry2 = session.createQuery("delete from Faculty where id=?1
118     qry2.setInteger(1, 105);
119     int count2 = qry2.executeUpdate();
120     System.out.println(count2+" Records(s) Deleted Successfully");
121
122     transaction.commit();
123 }
```

```
123 transaction.commit();
124 session.close();
125 sessionFactory.close();
126 }
127 public void updatedeletenamedparameters()
128 {
129     Configuration configuration=new Configuration();
130     configuration.configure("hibernate.cfg.xml");
131
132     SessionFactory sessionFactory=configuration.buildSessionFactory()
133
134     Session session=sessionFactory.openSession();
135
136     Transaction transaction=session.beginTransaction();
137
138     Query qry1=session.createQuery("update Faculty set salary=:v1,r
139     qry1.setParameter("v1", 86000.00);
140     qry1.setParameter("v2", "EX183 RedHat");
141     qry1.setParameter("v3", 101);
142     int count1=qry1.executeUpdate();
143     System.out.println(count1 + " records updated succesfully");
144     Query qry2=session.createQuery("delete from Faculty where id=:v
145     qry2.setParameter("v1", 104);
146     int count2 = qry2.executeUpdate();
147     System.out.println(count2+" Records(s) Deleted Successfully");
148     transaction.commit();
149 }
```

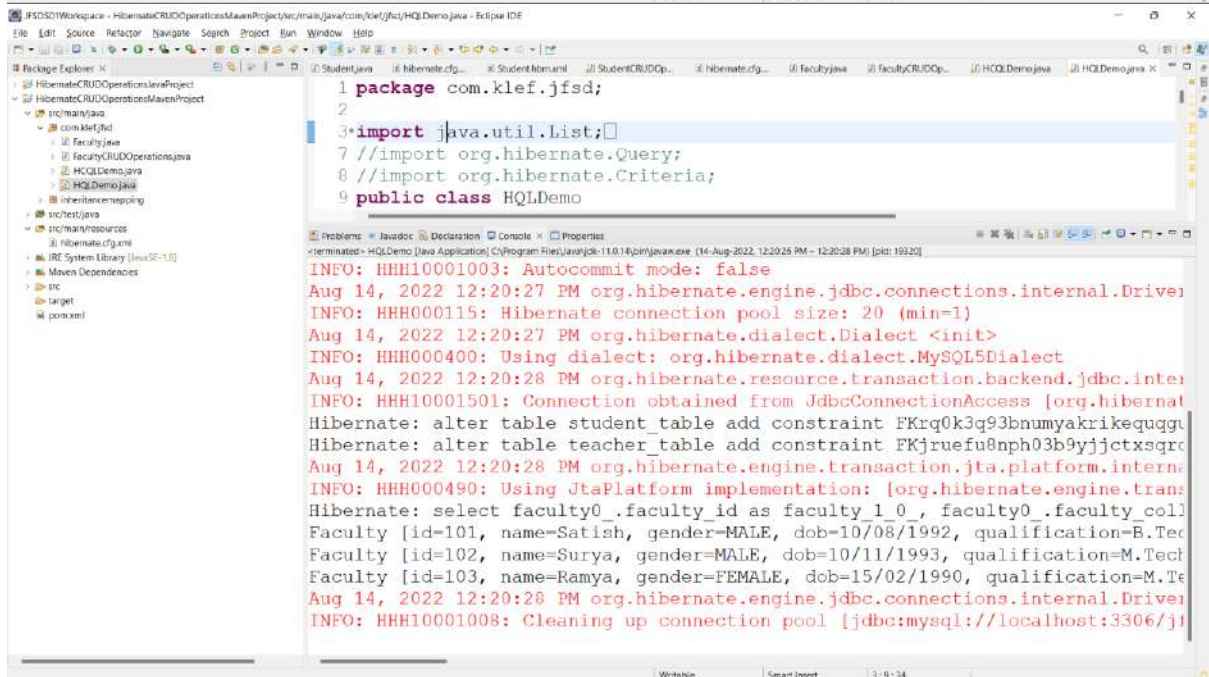
```
153 }
154
155 public void selectpositionalparameters()
156 {
157     Configuration configuration=new Configuration();
158     configuration.configure("hibernate.cfg.xml");
159
160     SessionFactory sessionFactory=configuration.buildSessionFactory();
161     Session session = sessionFactory.openSession();
162
163     Query qry1=session.createQuery("from Faculty where id=?1");
164     qry1.setInteger(1, 101);
165     List<Faculty> facultylist=qry1.getResultList();
166     System.out.println("Total Records = " +facultylist.size());
167     for(Faculty faculty:facultylist)
168     {
169         System.out.println(faculty.toString());
170     }
171     Query qry2=session.createQuery("from Faculty where username=?1");
172     qry2.setString(1, "surya4");
173     qry2.setString(2, "nopeac");
174     List<Faculty> flist=qry2.getResultList();
175     System.out.println("Total Records = " +flist.size());
176     if(flist.size()==1)
177     {
178         System.out.println("Faculty Record Found");
179     }
180 }
```

```
187 }
188
189 public void selectnamedparameters()
190 {
191     Configuration configuration=new Configuration();
192     configuration.configure("hibernate.cfg.xml");
193
194     SessionFactory sessionFactory=configuration.buildSessionFactory();
195     Session session = sessionFactory.openSession();
196
197     Query qry1=session.createQuery("from Faculty where id=:v");
198     qry1.setParameter("v", 101);
199     List<Faculty> facultylist=qry1.getResultList();
200     System.out.println("Total Records = " +facultylist.size());
201     for(Faculty faculty:facultylist)
202     {
203         System.out.println(faculty);
204     }
205     Query qry2=session.createQuery("from Faculty where username=:v1");
206     qry2.setParameter("v1", "surya4");
207     qry2.setParameter("v2", "nopeac");
208     List<Faculty> flist=qry2.getResultList();
209     System.out.println("Total Records = " +flist.size());
210     if(flist.size()==1)
211     {
212         System.out.println("Faculty Record Found");
213     }
214     System.out.println(flist.get(0));
215 }
```





```
197 qry1.setParameter("v", 101);
198 List<Faculty> facultylist=qry1.getResultList();
199 System.out.println("Total Records = " +facultylist.size());
200 for(Faculty faculty:facultylist)
201 {
202     System.out.println(faculty);
203 }
204 Query qry2=session.createQuery("from Faculty where username=:v1");
205 qry2.setParameter("v1", "surya4");
206 qry2.setParameter("v2", "nopeac");
207 List<Faculty> flist=qry2.getResultList();
208 System.out.println("Total Records = " +flist.size());
209 if(flist.size()==1)
210 {
211     System.out.println("Faculty Record Found");
212     System.out.println(flist.get(0));
213 }
214 else
215 {
216     System.out.println("Faculty Record Not Found");
217 }
218 session.close();
219 sessionFactory.close();
220 }
221 }
222 }
```



```
1 package com.klef.jfsd;
2
3 import java.util.List;
4 //import org.hibernate.Query;
5 //import org.hibernate.Criteria;
6 public class HQLDemo
```

Problems: Javadoc Declaration Console Properties

<terminated> HQLDemo (Java Application) C:\Program Files\Java\jdk-11.0.14\bin\javaw.exe (14-Aug-2022, 12:20:25 PM - 12:20:28 PM) [pid: 19320]

INFO: HHH10001003: Autocommit mode: false

Aug 14, 2022 12:20:27 PM org.hibernate.engine.jdbc.connections.internal.Driver

INFO: HHH000115: Hibernate connection pool size: 20 (min=1)

Aug 14, 2022 12:20:27 PM org.hibernate.dialect.Dialect <init>

INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL5Dialect

Aug 14, 2022 12:20:28 PM org.hibernate.resource.transaction.backend.jdbc.inte

INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hiberna

Hibernate: alter table student\_table add constraint FKrq0k3g93bnmyakrikequqg

Hibernate: alter table teacher\_table add constraint FKjruefu8nph03b9yjjctxsgr

Aug 14, 2022 12:20:28 PM org.hibernate.engine.transaction.jta.platform.interna

INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.trans

Hibernate: select faculty0\_.faculty\_id as faculty\_1\_0 , faculty0\_.faculty\_coll

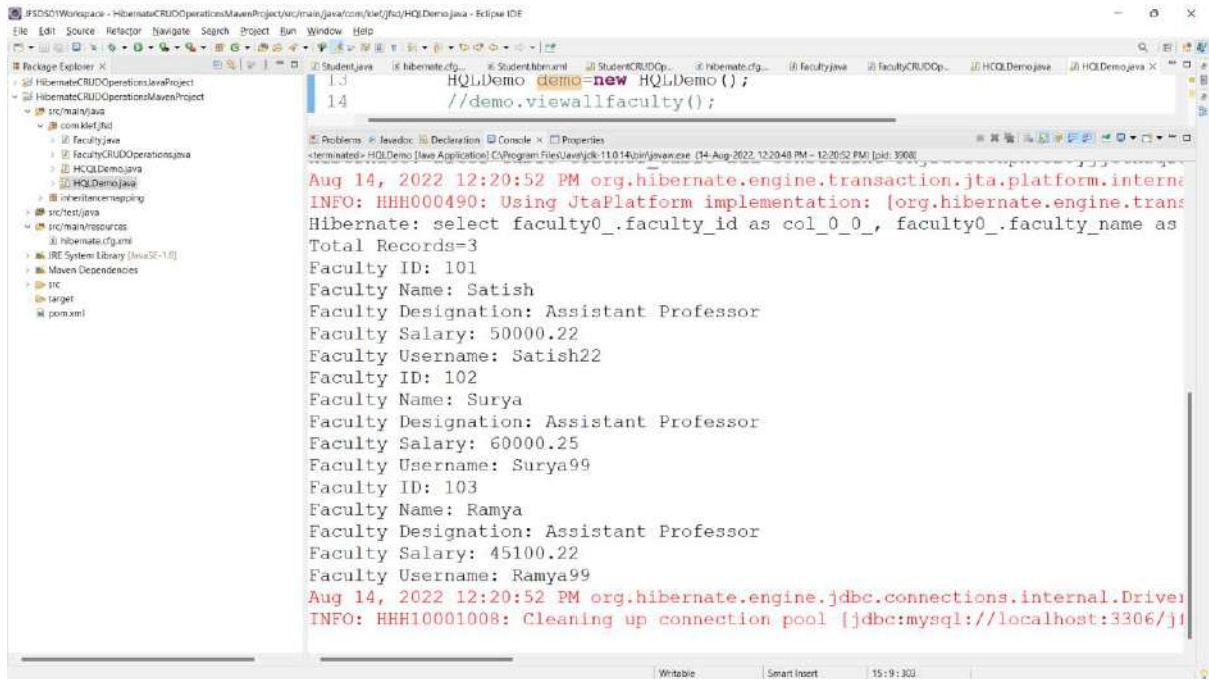
Faculty [id=101, name=Satish, gender=MALE, dob=10/08/1992, qualification=B.Tec

Faculty [id=102, name=Surya, gender=MALE, dob=10/11/1993, qualification=M.Tec

Faculty [id=103, name=Ramya, gender=FEMALE, dob=15/02/1990, qualification=M.Te

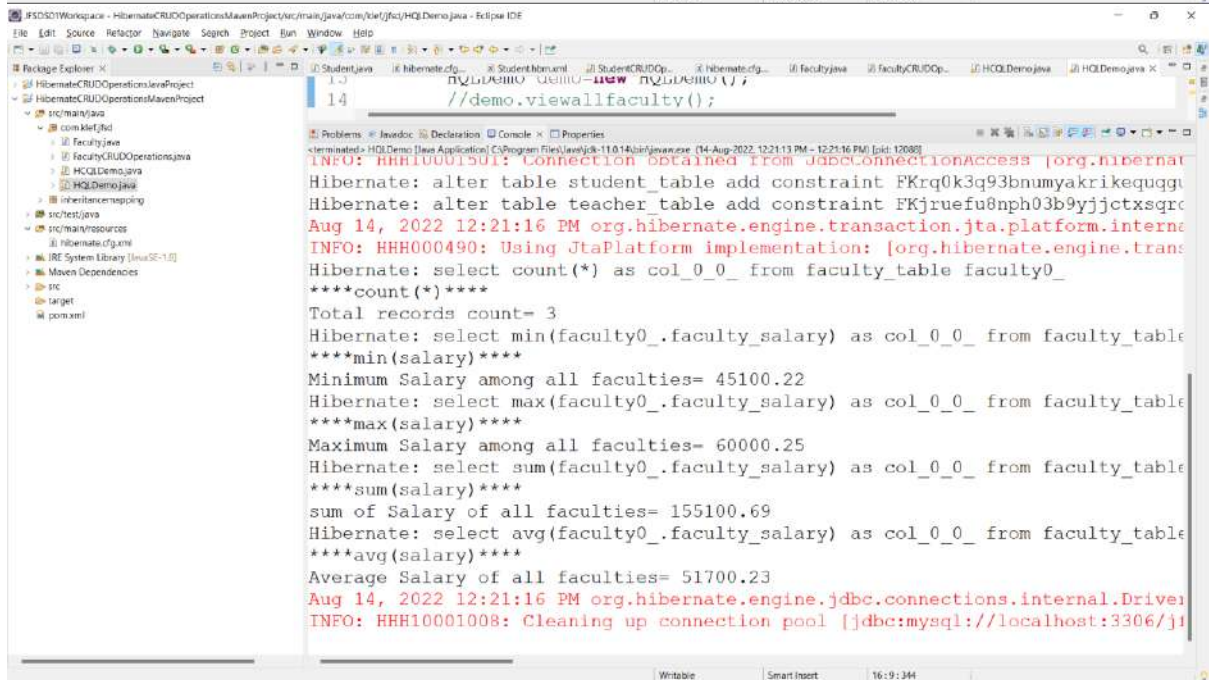
Aug 14, 2022 12:20:28 PM org.hibernate.engine.jdbc.connections.internal.Driver

INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/jf



```
13 HQLDemo demo=new HQLDemo();
14 //demo.viewAllFaculty();
```

Aug 14, 2022 12:20:52 PM org.hibernate.engine.transaction.jta.platform.internal.  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.trans  
Hibernate: select faculty0\_.faculty\_id as col\_0\_0\_, faculty0\_.faculty\_name as  
Total Records=3  
Faculty ID: 101  
Faculty Name: Satish  
Faculty Designation: Assistant Professor  
Faculty Salary: 50000.22  
Faculty Username: Satish22  
Faculty ID: 102  
Faculty Name: Surya  
Faculty Designation: Assistant Professor  
Faculty Salary: 60000.25  
Faculty Username: Surya99  
Faculty ID: 103  
Faculty Name: Ramya  
Faculty Designation: Assistant Professor  
Faculty Salary: 45100.22  
Faculty Username: Ramya99  
Aug 14, 2022 12:20:52 PM org.hibernate.engine.jdbc.connections.internal.Driver  
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/jl



```
13 HQLDemo demo=new HQLDemo();
14 //demo.viewAllFaculty();
```

Aug 14, 2022 12:21:16 PM org.hibernate.engine.jdbc.connections.internal.Driver  
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate  
Hibernate: alter table student\_table add constraint FKrq0k3q93bnmyakrikequqg  
Hibernate: alter table teacher\_table add constraint FKjruefu8nph03b9yjjctxsgrc  
Aug 14, 2022 12:21:16 PM org.hibernate.engine.transaction.jta.platform.internal.  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.trans  
Hibernate: select count(\*) as col\_0\_0\_ from faculty\_table faculty0\_  
\*\*\*\*count(\*)\*\*\*\*  
Total records count= 3  
Hibernate: select min(faculty0\_.faculty\_salary) as col\_0\_0\_ from faculty\_table  
\*\*\*\*min(salary)\*\*\*\*  
Minimum Salary among all faculties= 45100.22  
Hibernate: select max(faculty0\_.faculty\_salary) as col\_0\_0\_ from faculty\_table  
\*\*\*\*max(salary)\*\*\*\*  
Maximum Salary among all faculties= 60000.25  
Hibernate: select sum(faculty0\_.faculty\_salary) as col\_0\_0\_ from faculty\_table  
\*\*\*\*sum(salary)\*\*\*\*  
sum of Salary of all faculties= 155100.69  
Hibernate: select avg(faculty0\_.faculty\_salary) as col\_0\_0\_ from faculty\_table  
\*\*\*\*avg(salary)\*\*\*\*  
Average Salary of all faculties= 51700.23  
Aug 14, 2022 12:21:16 PM org.hibernate.engine.jdbc.connections.internal.Driver  
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/jl