DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SUBJECT CODE: 19CS2107
ENTERPRISE PROGRAMMING WORKBOOK

HQL, HCQL #9

Date of the Session:__/__/____ **Time of the Session: __to__**

Prerequisite:

- ☐ Basic idea on Hibernate Query Language, Query Interface
- ☐ Basic idea on hibernate Criteria Query Language

Pre-Lab Task:

1. Write the HQL query to find the count of red colored bottles and of quantity 1 litre.

```
Select Count(*) from Bottle as B where B.Colour = Red &
B.quantity = 1"
```

2. Write the HQL query to get data of the first 10 bottles which are microwave safe (i.e, the value of microwave safe is "yes") in ascending order with respect to the quantity.

```
String hql = "Select from Bottle as B where B.microwave = "Yes"
ORDER BY B.quantity ASC"
    Query = Session.(Create query, hql);
    Query.Set First Result(10);
    Query.Set Results(10);
```

3. Write the HCQL query to get the 15th to 30th record.

```
C.Set First Result(15); C.Set Max Result(30);
    List list = C.list();
```

4. Write the HCQL query to get the records who salary is greater than 150000.

```
c.add(Restrictions.gt("Salary",150000));
List list = c.list();
```

5. Write the HCQL query to get the records in ascending order on the basis of salary.

```
Criteria c = Session.createCriteria(Emp.class
c.add order(Order.asc("Salary"));
List list = c.list();
```

In Lab Task:

1. Stoins is the manager of Minimal Cube company. He maintains the records of employees working in his company. Create a java class where he gets and sets the values of EmpID, EmpName, EmpSalary, EmpAddress. Use Hibernate Frame-work to reduce manual work. When employees are terminated he is likely to delete the record of employee in his database and update the data when it is required. After performing all the operations he is likely to know the employees working in his company at the end of every month, so retrieve the data those who are working in his company. Create separate java class for retrieving, updating and deleting so that the Mr. Stoins can easily work when an employee data needs to update, delete or retrieve. Use Concept of HCQL (Hibernate Criteria Query Language).

Writing space for the Problem:(For Student's use only)

Employee.java

```
Package myPackage;
  Public class Employee{
    Private int empId;
    Private String empName;
    Private string empAddress;

  Public int getEmpId() {
        return empId; }
  Public void setEmpId( int empId){.
        this.empId = empId;
      }.
    Public String getEmpName(){.
        return Emp Name; }.
    Public void Set Emp Name (String name) {
        this.empName = emp Name
        }.

Public String get EmpAddress().
 {
  return emp Address;
 }.
Public void Set Emp Address (String emp Address){.
```

this.empAddress = emp.Address; }

Employee.hbm.xml :

```
<? xml version = "1.0" encoding = "UTF-8" ?>
<! DocTYPE hibernate mapping PUBLIC "-//Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name = "myPackage.employee">
    <id name = "empId">
    <generator class = "increment" ></generator>
    </id>
  <Property name = "empName" ></Property>
  <Property name = "empSalary" ></Property>
  <Property name = "empAddress></Property>
  </class>
  </hibernate-mapping>
```

Employee.cfg.xml :

```
<? xml version = "1.0", encoding = "UTF-8" ? >
<! DocTYPE hibernate-Configuration - SYSTEM
"http://www.hibernate.org/dtd/hibernate-Configuration-3.0.dtd">
<hibernate-Configuration>
  <session-factory>
  <Property name = "Connection.driver_class"> oracle.jdbc
driver.oracleDriver</Property>
<Property name = "Connection.user" >System</Property>
<Property name = "Connection.password" > manager</Property>
<Property name = "Show-SQL" > true</Property>
```

Writing space for the Problem:(For Student's use only)

<Property name = "connection.url"> driver.jdbc:oracle:
this.@localhost:1521:XE </Property>.

<Property name = "Connection.user"> System </Property>.

<Property name = "Connection.Password"> manager </Property>.

<Property name = "show-sql"> true </Property>

<Property name = "hbm2ddl.auto"> update </Property>

<Property name = "dialet"> update </Property>.

oracle.dialect </Property>.

<mapping resource = "employee.hbm.xml"> </mapping>.

</Session-factory>.

</hibernate-Configuration>.

Test.java :

```
Package mypackage;
import org.hibernate.*; import org.hibernate.cfg.*;
Public class Test {
  Public static void main (String [] args) {
  Configuration cfg = new Configuration()
  Cfg.Configure("hibernate.cfg.xml");
  SessionFactory sf = cfg.buildSessionFactory();
      Session Session = sf.openSession();
  Transaction transaction = session.beginTransaction();
      Employee employee = new Employee();
      employee.setEmpName("sriram");
      employee.setEmpSalary("10000");
      employee.setEmpAddress("Nellore");
```

```
Session . Save (employee);
transaction.Commit ();
Session.close ();
System. out .Print ln ("Inserted successfully");

Query query = Session.Create Query ("update Employee Set empName: n
where emp Id = I ");
query. set Parameter ("n", "RAM ");
Query . Set Parameter (": ", 2 );
int Status . query execute update ();
transaction Commit ();
System . out . Print ln ("updated successfully");
query = session Create Query ("delete from Employee where id=4");
query . execute update ();
transaction commit ();
System. out . Print ln ("Deleted successfully");
query = session Create Query ("from Employee");
List List = query list ();
iterator itr = list . iterator ();
while itr . has Next {.
Employee employee = (Employee) itr next ();

}
```

Post Lab Task:

1. Mr. Deekshit is running an Online Shopping website where shopping take place in his website regularly. So he decided to maintain two different tables namely ORDER containing attributes like ID, ORDERDATE, ORDRENUMBER, CUSTOMERID, TOTALAMOUNT where ID is primary key and another table describing CUSTOMER (ID, FIRSTNAME, LASTNAME, CITY, COUNTRY, PHONE) where ID will not have a null value. By using JOINS through Hibernate Frame-work help Mr.Deekshit to know the details of customer and items customer purchased. Display Order-number, Total-amount, First-name, Last-name, City, Country using ORDER.ID and CUSTOMER.ID.

Writing space for the Problem:(For Student's use only)

```
import java.io.*;
import java.Servlet.*;
import javax.servlet.http.*;
import java.sql.*;

Public class Display extends HttpServlet{.
  Public void doGet(HttpServletRequest req, HttpServletResponse
  ).
  throws IOexception, ServletException{.
    Print writes out.res.getWriter();
    res.SetContentType("text/html");
    try{.
      Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection Con = DriverManager.getConnection(jdbc:oracle:thin:@localhost
    1521:xe", "System", "root");
    Sql = "select orderNumber, Total amount, First Name, Last Name,
    City, Country FROM [order Details] JOIN CUSTOMER DETAILS,
    ON [order Details].CustomerId = Customer Details Id"
  PreparedStatement stmt = Con.PrepareStatement(Sql);
      Statement Stmt = Con.CreateStatement();
```

Writing space for the Problem:(For Student's use only)

```
Result Set rs = Stmt.executeQuery(SQL);
out.println("<table border=1 width=50% height=50%>");
out.println("<tr><th>empid</th><th>empName</th>
<th>Salary</th></tr>");
while(rs.next())
{
Integer n= rs.getInt("order number");
Integer nn = rs.getInt("total Amount");
Integer n = rs.getString("first Name");
String n=rs.getString("last Name");
String n=rs.getString("city");
String n=rs.getString("country");
String
}
out.println("</table>");
out.println("</html></body>");
Con.close();
}
Catch(Exception e)
{
out.println(e);
}
}
```