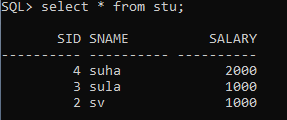
**1. Deleting.**

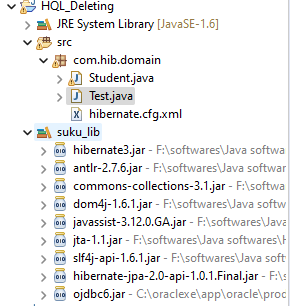
**2. Updating.**

**3. Inserting.**

**4. Seleting.**

**DELETING THE RECORDS**

****

****

**Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu")

**public** **class** Student **implements** Serializable{

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

**package** com.hib.domain;

**import** org.hibernate.Query;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.classic.Session;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration cfg=**new** Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Transaction t=s.beginTransaction();

Query q=s.createQuery("delete from Student where marks=1000");

**int** count=q.executeUpdate();

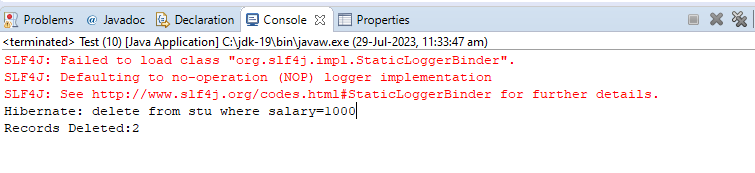
System.***out***.println("Records Deleted:"+count);

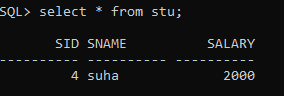
t.commit();

}

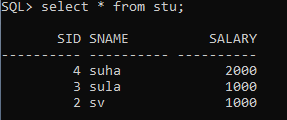
}

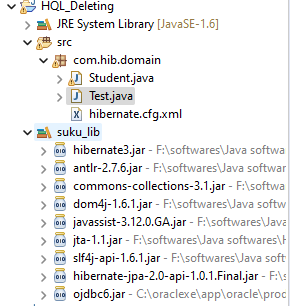
**Output:**

****

****

**UPDATING THE RECORDS**

****

****

**Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu")

**public** **class** Student **implements** Serializable{

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<property name=*"show\_sql"*>true</property>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

**package** com.hib.domain;

**import** org.hibernate.Query;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.classic.Session;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Configuration cfg=**new** Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Transaction t=s.beginTransaction();

Query q=s.createQuery("update Student set marks=2500 where marks=1000");

**int** count=q.executeUpdate();

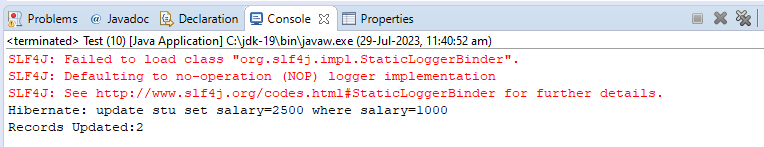
System.***out***.println("Records updated:"+count);

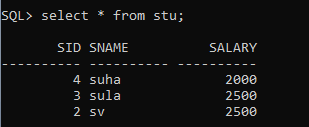
t.commit();

}

}

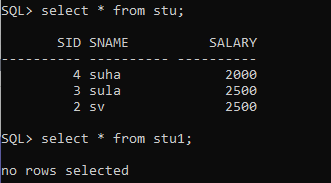
**Output:**

****

****

**INSERTING RECORDS**

**Example:1- T**ransfer records from one table to another table.



**Annotated classes/Domain classes:**

**1.Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu")

**public** **class** Student **implements** Serializable{

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**2.Student1.java**

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu1")

**public** **class** Student1 {

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**Hibernate configuration File**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping class=*"com.hib.domain.Student1"*/>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

**package** com.hib.domain;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Scanner;

**import** org.hibernate.NonUniqueResultException;

**import** org.hibernate.Query;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.classic.Session;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**try** {

Configuration cfg=**new** Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Transaction t=s.beginTransaction();

Query q;

Scanner s1=**new** Scanner(System.***in***);

System.***out***.println("It transfer all the records from one table to another table.");

q=s.createQuery("insert into Student1(sid,sname,marks)select s.sid,s.sname,s.marks from Student as s");

**int** count=q.executeUpdate();

t.commit();

System.***out***.println("Records Updated:"+count);

}

**catch**(Exception e) {

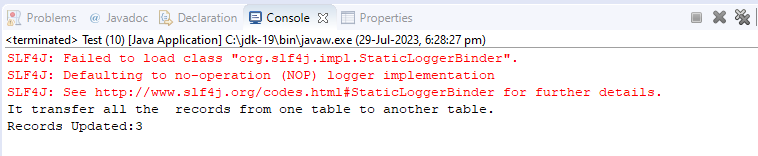
System.***out***.println(e.getMessage());

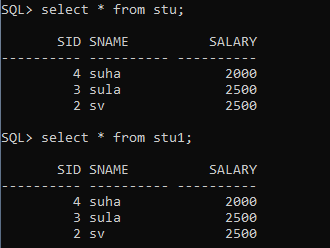
}

}

}

Output:-



****

**RETRIEVING THE RECORDS**

**Example:-1** In this example, the records are retrieved using either list(-),iterate(-), or uniqueResult(-) method.

**1.Annatotated Class:-**

package com.hib.domain;

import java.io.Serializable;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="stu")

public class Student implements Serializable{

@Id

@Column(name="sid")

private int sid;

@Column(name="sname")

private String sname;

@Column(name="salary")

private float marks;

public int getSid() {

return sid;

}

public void setSid(int sid) {

this.sid = sid;

}

public String getSname() {

return sname;

}

public void setSname(String sname) {

this.sname = sname;

}

public float getMarks() {

return marks;

}

public void setMarks(float marks) {

this.marks = marks;

}

}

**Hibernate.cfg.xml:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication: Test.java**

package com.hib.domain;

import java.util.Iterator;

import java.util.List;

import java.util.Scanner;

import org.hibernate.NonUniqueResultException;

import org.hibernate.Query;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import org.hibernate.classic.Session;

public class Test {

public static void main(String[] args) {

try {

Configuration cfg=new Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Transaction t=s.beginTransaction();

Query q;

Scanner s1=new Scanner(System.in);

System.out.println("Retrieve records Using. \n 1. List \n 2. Iterator \n 3. UniqueResult");

System.out.print("Enter the choice:");

int ch=s1.nextInt();

switch(ch) {

case 1:

q=s.createQuery("from Student");

List<Student>l=q.list();

System.out.println("sNo"+"\t"+"sName"+"\t"+"Marks");

System.out.println("===============================");

for(Student s2:l) {

System.out.println(s2.getSid()+"\t"+s2.getSname()+"\t"+s2.getMarks());

}

break;

case 2:

q=s.createQuery("from Student");

Iterator<Student> i=q.iterate();

System.out.println("sNo"+"\t"+"sName"+"\t"+"Marks");

System.out.println("===============================");

while(i.hasNext()) {

Student s3=i.next();

System.out.println(s3.getSid()+"\t"+s3.getSname()+"\t"+s3.getMarks());

}

break;

case 3:

q=s.createQuery("from Student");

try {

System.out.println("sNo"+"\t"+"sName"+"\t"+"Marks");

System.out.println("===============================");

Student e1=(Student)q.uniqueResult();

System.out.println(e1.getSid()+"\t"+e1.getSname()+"\t"+e1.getMarks());

}

catch(NonUniqueResultException e)

{

System.out.println("student table have more than one record");

}

break;

}

}

catch(Exception e) {

System.out.println(e.getMessage());

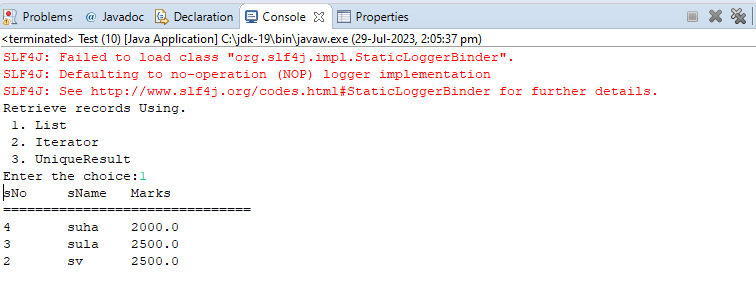
}

}

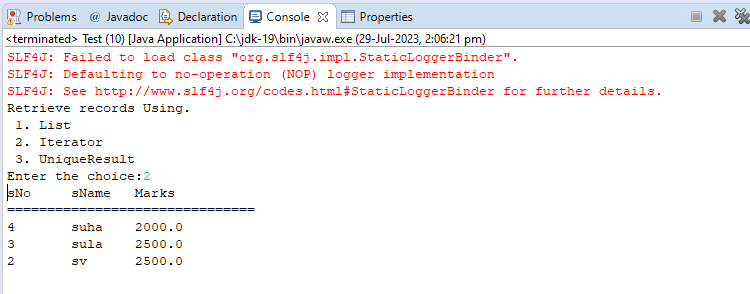
}

**Output:**

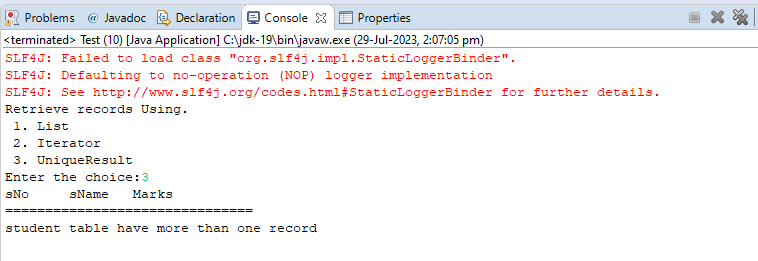
**Run1:**

****

**Run2:**

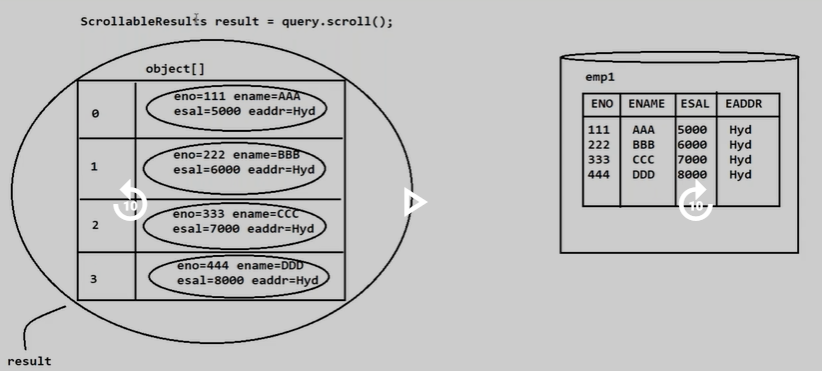
****

**Run3:**

****

**\*\*\*\*Example:2**  The records are retrieved from table using scroll(-) method.

Scroll(-) method returns the ScrollableResults object. This ScrollableResults object has object type array. Actual records are stored in Object type[] in the form of Objects.



**Domain class:Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu")

**public** **class** Student **implements** Serializable{

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**Hibernate configuration file:hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

**package** com.hib.domain;

**import** org.hibernate.Query;

**import** org.hibernate.ScrollableResults;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.Transaction;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.classic.Session;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**try** {

Configuration cfg=**new** Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Query query=s.createQuery("from Student");

ScrollableResults scr=query.scroll();

System.***out***.println("sNo"+"\t"+"sName"+"\t"+"Marks");

System.***out***.println("===============================");

**while**(scr.next()) {

Object [] obj=scr.get();

**for**(**int** i=0;i< obj.length;i++) {

Student e1=(Student)obj[i];

System.***out***.println(e1.getSid()+"\t"+e1.getSname()+"\t"+e1.getMarks());

}

}

}

**catch**(Exception e) {

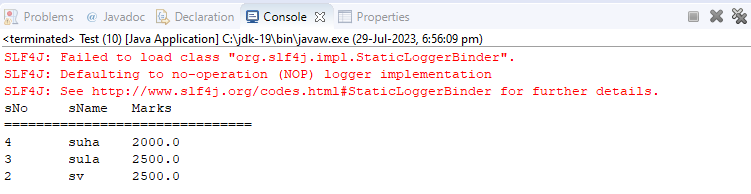
System.***out***.println(e.getMessage());

}

}

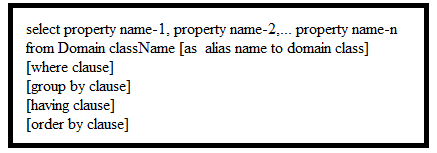
}

**Output:**

****

**SELECT QUERY**

**Syntax:**



Note:- If we use property names along with select clause in HQL query then results are generated in the form of Object[] where object contains the individual column values.

**Domain class:Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.Id;

**import** javax.persistence.Table;

@Entity

@Table(name="stu")

**public** **class** Student **implements** Serializable{

@Id

@Column(name="sid")

**private** **int** sid;

@Column(name="sname")

**private** String sname;

@Column(name="salary")

**private** **float** marks;

**public** **int** getSid() {

**return** sid;

}

**public** **void** setSid(**int** sid) {

**this**.sid = sid;

}

**public** String getSname() {

**return** sname;

}

**public** **void** setSname(String sname) {

**this**.sname = sname;

}

**public** **float** getMarks() {

**return** marks;

}

**public** **void** setMarks(**float** marks) {

**this**.marks = marks;

}

}

**Hibernate.cfg.xml**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hibernate.connection.driver\_class"*>oracle.jdbc.OracleDriver</property>

<property name=*"hibernate.connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"hibernate.connection.username"*>system</property>

<property name=*"hibernate.connection.password"*>tiger</property>

<property name=*"hibernate.dialect"*>org.hibernate.dialect.OracleDialect</property>

<mapping class=*"com.hib.domain.Student"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

**package** com.hib.domain;

**import** java.util.List;

**import** org.hibernate.Query;

**import** org.hibernate.SessionFactory;

**import** org.hibernate.cfg.Configuration;

**import** org.hibernate.classic.Session;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**try** {

Configuration cfg=**new** Configuration();

cfg.configure("com/hib/domain/hibernate.cfg.xml");

SessionFactory sf=cfg.buildSessionFactory();

Session s=sf.openSession();

Query query=s.createQuery("select sid,sname,marks from Student");

List<Object[]> li=query.list();

System.***out***.println("sNo"+"\t"+"sName"+"\t"+"Marks");

System.***out***.println("===============================");

**for**(Object[] obj:li) {

**for**(Object obj1:obj) {

System.***out***.print(obj1+"\t");

}

System.***out***.println();

}

}

**catch**(Exception e) {

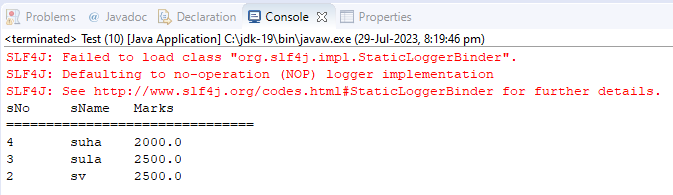
System.***out***.println(e.getMessage());

}

}

}

**Output:**

****