we have two approaches to call procedure or function.

**Approach-1:-** if procedure /function returns a scalar value then it can be invoked using callable statement of jdbc in hibernate application.

The scalar value means row value, string or date …etc.

This approach is added in hibernate 3.6.5 version onwards.

**Approach-2:-** If procedure/function returns a cursor object then it can be invoked using “native SQL Queries” concept in hibernate application.

**APPROACH-1**

The following are steps to invoke the procedure/approach in Approach-1;

1. Create a class which implements **org.hibernate.jdbc.Work** interface.
2. Implements only one abstrace method of Work interface called execute().
3. Write procedure/function invoking logic in execute() method.
4. Pass the class object as parameter to doWork(-) method of session interface.

Example:-

**Domain Class/Pojo 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.NamedNativeQueries;

**import** javax.persistence.NamedNativeQuery;

**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>

**Supporting Class: CallProcedure.java**

**package** com.hib.domain;

**import** java.sql.CallableStatement;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**import** java.sql.Types;

**import** org.hibernate.jdbc.Work;

**public** **class** CallProcedure **implements** Work {

**public** **void** execute(Connection con)**throws** SQLException {

CallableStatement cs=con.prepareCall("{call sum1(?,?,?)}");

cs.registerOutParameter(3, Types.***INTEGER***);

cs.setInt(1, 10);

cs.setInt(2, 20);

cs.execute();

System.***out***.println("Sum is:"+cs.getInt(3));

cs.close();

}

}

**ClientApplication:**Test.java

**package** com.hib.domain;

**import** java.sql.SQLException;

**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)**throws** SQLException {

**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();

CallProcedure cp=**new** CallProcedure();

s.doWork(cp);

t.commit();

}

**catch**(Exception e) {

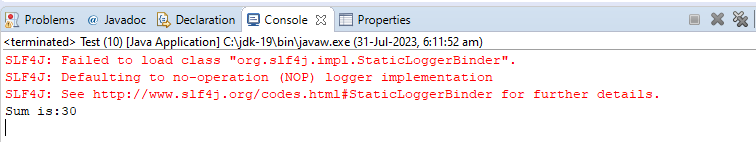
e.printStackTrace();

}

}

}

**Output:**-



**APPROACH-2**

Example: - call the procudure using Native SQL query with xml tags.

**Domain Class/PojoClass: 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.NamedNativeQueries;

import javax.persistence.NamedNativeQuery;

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;

}

}

**HibernateMappingFile: Student.hbm.xml**

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

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping>

<class name=*"com.hib.domain.Student"* table=*"stu"*>

<id name=*"sid"*/>

<property name=*"sname"*/>

<property name=*"marks"* column=*"salary"*/>

</class>

<sql-query name=*"cp"* callable=*'true'*>

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

{call sum1(?)}

</sql-query>

</hibernate-mapping>

**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.Oracle10gDialect</property>

<mapping resource=*"com/hib/domain/Student.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

package com.hib.domain;

import java.sql.SQLException;

import java.util.List;

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)throws SQLException {

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=s.getNamedQuery("cp");

System.out.println("sid\tsname\tmarks");

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

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

for(Student x:li) {

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

}

t.commit();

}

catch(Exception e) {

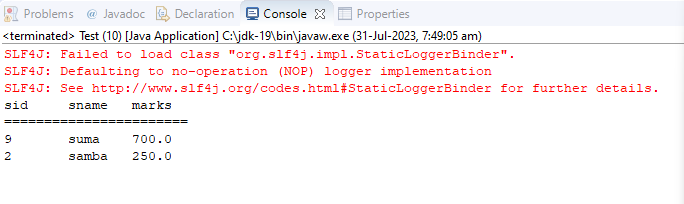
e.printStackTrace();

}

}

}

Output:-



Example:2 call the Procedure using Native SQL Query with Annotations.

Pending