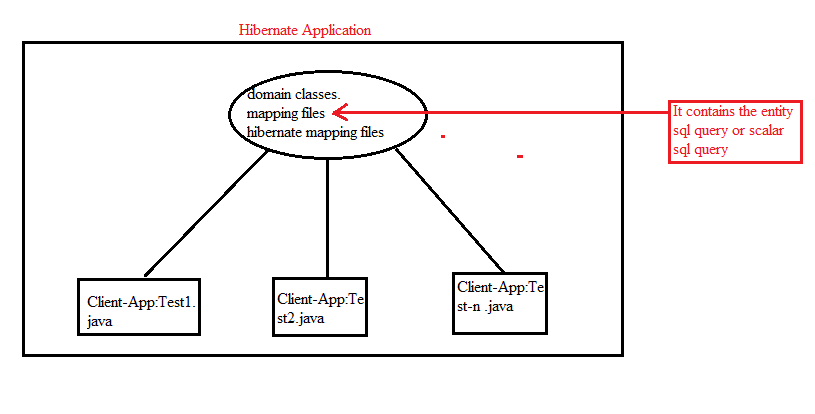
A named query is a HQL or SQL query and defined either using **@NamedQuery** annotation or an **XML file**. We can refer to a named query by its name, in the runtime, when we need to execute it.



All client applications wanted to select records from same table. All Client applications should have same Scalar sql query or Entity sql query.

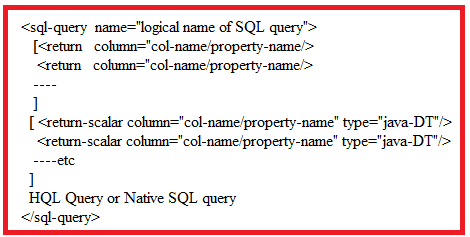
It leads to redudant problem .

It may leads to query inconsistency problem.

To solve above two problems, hibernate provide **named queries**.

**Approach:-1 XML File**

**Syntax:**



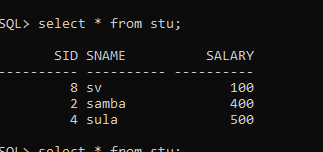
**The <sql-query> tag should be written in <hibernate-mapping>tag.**

The client application should invokes getNamedQuery(“query name”) to get the sql query from hibernate mapping file in the form “Query” object.



This method presents in the **Session** Interface.

**Example:-1**  Insertion,Deletion,Updation,Creation operations will be performed in this example.

****

**Domain class/persistence 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.Table;

public class Student implements Serializable{

private int sid;

private String sname;

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 Mapping File: 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=*"insert-1"*>

insert into stu values(9,'suma',700)

</sql-query>

<sql-query name=*"delete-1"*>

delete from stu where sid=4

</sql-query>

<sql-query name=*"update-1"*>

update stu set salary=700 where sid=8

</sql-query>

<sql-query name=*"create-1"*>

create table rock(rid number(2),rname varchar2(10))

</sql-query>

</hibernate-mapping>

**HibernateConfigurationFile:**

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

</session-factory>

</hibernate-configuration>

**ClientApplication:Test.java**

package com.hib.domain;

import java.util.List;

import java.util.Scanner;

import org.hibernate.Query;

import org.hibernate.SQLQuery;

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 sq=s.getNamedQuery("insert-1");

int count=sq.executeUpdate();

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

sq=s.getNamedQuery("delete-1");

sq.executeUpdate();

System.out.println("deleted records:"+count);

sq=s.getNamedQuery("update-1");

sq.executeUpdate();

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

sq=s.getNamedQuery("create-1");

sq.executeUpdate();

System.out.println("table Created");

t.commit();

}

catch(Exception e) {

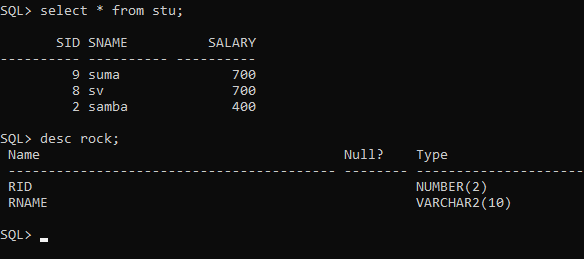
System.out.println(e.getMessage());

}

}

}

**Output:-**

****

Example:2 In this example , Records are selected using **Entity named SQL query.**

**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;

public class Student implements Serializable{

private int sid;

private String sname;

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;

}

}

**HibernateMapping File: 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=*"sv"*>**

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

**select \* from stu**

**</sql-query>**

</hibernate-mapping>

**HibernateConfigurtationFile: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 resource=*"com/hib/domain/Student.hbm.xml"*/>

</session-factory>

</hibernate-configuration>

**Client Application:Test.java**

**package** com.hib.domain;

**import** java.util.List;

**import** java.util.Scanner;

**import** org.hibernate.Query;

**import** org.hibernate.SQLQuery;

**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 q1;

q1=s.getNamedQuery("sv");

List<Student>l=q1.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());

}

}

**catch**(Exception e) {

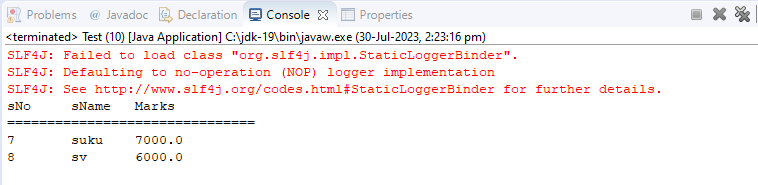
System.***out***.println(e.getMessage());

}

}

}

Output:-



Example:3 In this application, Records are selected using **Scalar Named SQL query.**

**POJO CLASSES:**

package com.hib.domain;

import java.io.Serializable;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

public class Student implements Serializable{

private int sid;

private String sname;

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=*"select-1"*>

<return-scalar column=*"sid"*/>

<return-scalar column=*"sname"*/>

<return-scalar column=*"salary"*/>

select sid,sname,salary from stu

</sql-query>

</hibernate-mapping>

**HibernateConfigurationFile: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 resource=*"com/hib/domain/Student.hbm.xml"*/>

</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.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 sq=s.getNamedQuery("select-1");

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

System.***out***.println("sid"+"\t"+"sname"+"\t"+"salary");

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

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

**for**(Object y:x) {

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

}

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

}

}

**catch**(Exception e) {

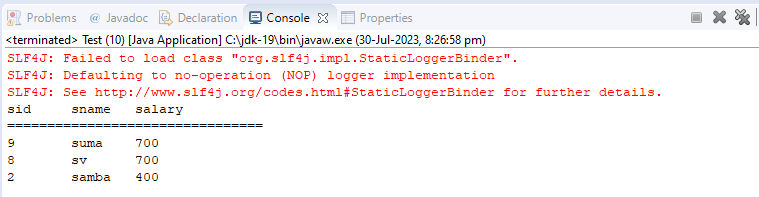
System.***out***.println(e.getMessage());

}

}

}

**Output:-**

****

**APPROACH-2 ANNOTATION**

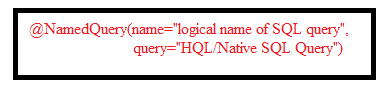
**The following two annotations are used in HQL:**

**1.@NamedQuery**

Named query definition has two important attributes:

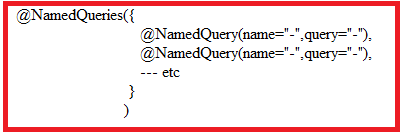
* name: The name of the named query by which it will be located using the *Session* or *EntityManager* interface.
* query: The HQL or SQL statement to get executed in the database.

Syntax:-



**2. @NamedQueries**

If we havenamed multiples queries on single pojo class, we can group them using @NamedQueries.



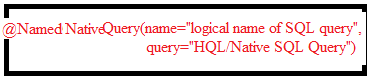
2.**The following two annotations are used in Native SQL:**

**1.@NamedNativeQuery**

Named query definition has two important attributes:

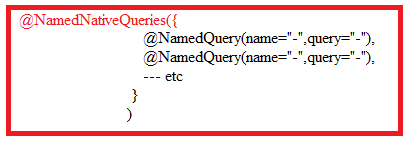
* name: The name of the named query by which it will be located using the *Session* or *EntityManager* interface.
* query: The HQL or SQL statement to get executed in the database.

Syntax:-



**2. @NamedNativeQueries**

If we havenamed multiples queries on single pojo class, we can group them using @NamedQueries.

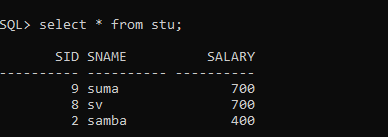


The client application should invokes getNamedQuery(“query name”) to get the sql query from hibernate mapping file in the form “Query” object.



This method presents in the **Session** Interface.

**Example:-1 deletion,updation,selection operations are performed in this application using HQL.**



**Annotated Class: Student.java**

**package** com.hib.domain;

**import** java.io.Serializable;

**import** javax.persistence.Column;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.NamedQueries;

**import** javax.persistence.NamedQuery;

**import** javax.persistence.Table;

@NamedQueries({

//@NamedQuery(name="inserteion-1",query="insert into Student(sid,sname,marks)values(17,'rock',7000f)"),

@NamedQuery(name="deletion-2",query="delete from Student where sid=8"),

@NamedQuery(name="update-3",query="update Student set salary=250 where sid=2"),

@NamedQuery(name="select-1",query="select sid,sname,marks from Student")

}

)

@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;

}

}

**HibernateconfigurationFile**

<?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.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 sq=s.getNamedQuery("select-1");

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

System.***out***.println("sid"+"\t"+"sname"+"\t"+"salary");

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

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

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

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

}

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

}

sq=s.getNamedQuery("deletion-2");

**int** x=sq.executeUpdate();

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

sq=s.getNamedQuery("update-3");

x=sq.executeUpdate();

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

t.commit();

}

**catch**(Exception e) {

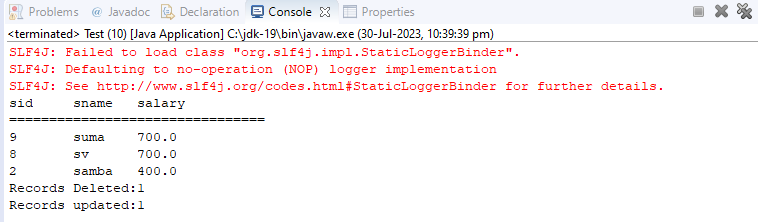
e.printStackTrace();

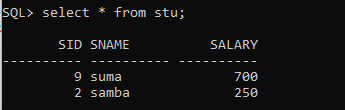
}

}

}

**Output:-**

****

****

**Example:-2Creation,insertion,deletion,updation,selection operations are performed in this application using Native SQL.**

**Domain class:Student.java**

@NamedNativeQueries({

@NamedNativeQuery(name="creation-1",query="create table Student-1(sid number(2),sname varchar2(10))"),

@NamedNativeQuery(name="insertion-1",query="insert into stu(sid,sname,salary)values(17,'rock',7000f)"),

@NamedNativeQuery(name="deletion-2",query="delete from stu where sid=8"),

@NamedNativeQuery(name="update-3",query="update stu set salary=250 where sid=2"),

@NamedNativeQuery(name="select-1",query="select sid,sname,salary from stu")

}

)

@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;

}

}