There are two varieties of one-to-one association:

• primary key associations

• unique foreign key associations

|  |
| --- |
|  |
| **primary key associations** |  | **unique foreign key associations** |  |
|  |  |  |  |
| Primary key associations **don't need an extra table column** |  | **Need extra column** |  |
|  |  |  |  |
| Hibernate identifier generation strategy called **foreign** |  | Hibernate identifier generation strategy called native or other |  |
|  |  |  |  |
| Generator needs param |  |  |  |
|  |  |  |  |
| <one to one >tag needs constrained=true |  | <many to one> unique=true |  |
|  |  |  |  |
|  |  |  |  |

unidirectional one-to-one association on a primary key  
uses a special id generator foreign in exam.hbm.xml  
reversed the direction of the association  
direction is exam to stud

it does not need extra column  
uses a special id generator foreign  
use  constrained=true

Two tables related by a primary key association share the same primary key values.  
The primary key of one table is also a foreign key of the other. The main difficulty  
with this approach is ensuring that associated instances are assigned the same primary  
key value when the objects are saved

stud is not having exam object but exam is having stud object

package pack;

public class Stud  {  
 private int roll;  
 private String name;

 public int getRoll() {  
  return roll;  
 }

 public void setRoll(int roll) {  
  this.roll = roll;  
 }

 public String getName() {  
  return name;  
 }

 public void setName(String name) {  
  [this.name](http://www.google.com/url?q=http%3A%2F%2Fthis.name&sa=D&sntz=1&usg=AFQjCNHGHAckObDvjanVgmuORUnQyyIaAg) = name;  
 }

}

package pack;

public class Exam  {  
 private int roll;  
 private String subject;  
 private Stud stud;

 public int getRoll() {  
  return roll;  
 }

 public void setRoll(int roll) {  
  this.roll = roll;  
 }

 public String getSubject() {  
  return subject;  
 }

 public void setSubject(String subject) {  
  this.subject = subject;  
 }

 public Stud getStud() {  
  return stud;  
 }

 public void setStud(Stud stud) {  
  this.stud = stud;  
 }  
}

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-mapping-3.0.dtd&sa=D&sntz=1&usg=AFQjCNEjlBfisNKuNhGBCp0Ww1ceZaDd5Q)">  
<hibernate-mapping>  
 <class name="pack.Stud">  
  <id name="roll" type="int">  
   <generator class="native" />  
  </id>  
  <property name="name" />  
 </class>  
</hibernate-mapping>

A unidirectional one-to-one association on a primary key usually uses a special id generator. (Notice that we've  
reversed the direction of the association in this example.)  
generator is foreign

direction is exam to stud  
in one to one constrained=true

It tells Hibernate that there is a foreign key constraint on the primary key of Exam that refers  
to the primary key of stud.

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-mapping-3.0.dtd&sa=D&sntz=1&usg=AFQjCNEjlBfisNKuNhGBCp0Ww1ceZaDd5Q)">  
<hibernate-mapping>  
 <class name="pack.Exam">  
  <id name="roll">  
   <generator class="foreign">  
    <param name="property">stud</param>  
   </generator>  
  </id>  
  <one-to-one name="stud" class="pack.Stud" constrained="true">  
  </one-to-one>  
  <property name="subject" type="string"></property>  
 </class>  
</hibernate-mapping>

package pack;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
import pack.Exam;  
import pack.Stud;

public class app1 {  
 public static void main(String[] args) {  
  SessionFactory sf = new Configuration().configure("pack/hibernate.cfg.xml")  
    .buildSessionFactory();  
  Session session = sf.openSession();  
  Stud stud = new Stud();  
  Exam exam = new Exam();  
    
  stud.setName("ram");  
    
  exam.setSubject("math");  
  exam.setStud(stud);  
     
  session.save(stud);  
  session.beginTransaction().commit();  
  session.close();  
  System.out.println("ADDED");  
 }  
}

<?xml version='1.0' encoding='utf-8'?>  
<!DOCTYPE hibernate-configuration PUBLIC  
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-configuration-3.0.dtd&sa=D&sntz=1&usg=AFQjCNFF8pC3CTwnLXmor30uuf81LA9gSg)">  
<hibernate-configuration>  
 <session-factory>  
  <property name="connection.driver\_class">org.hsqldb.jdbcDriver</property>  
  <property name="connection.url">jdbc:hsqldb:hsql://localhost/</property>  
  <property name="connection.username">sa</property>  
  <property name="connection.password"></property>  
  <property name="connection.pool\_size">1</property>  
  <property name="dialect">org.hibernate.dialect.HSQLDialect</property>  
  <property name="show\_sql">true</property>  
  <property name="hbm2ddl.auto">create</property>  
  <mapping resource="pack/exam.hbm.xml" />  
  <mapping resource="pack/stud.hbm.xml" />  
 </session-factory>  
</hibernate-configuration>

unidirectional one-to-one association on a foreign  key  
direction stu-->exam  
can have additional column  
unique constraint on the foreign key

Using a foreign key association  
use a <many-to-one> mapping with a unique constraint on the foreign key.

package pack2;

import java.io.Serializable;

public class Stud  {  
 private int roll;  
 private String name;  
 private Exam exam;

 public int getRoll() {  
  return roll;  
 }

 public void setRoll(int roll) {  
  this.roll = roll;  
 }

 public String getName() {  
  return name;  
 }

 public void setName(String name) {  
  [this.name](http://www.google.com/url?q=http%3A%2F%2Fthis.name&sa=D&sntz=1&usg=AFQjCNHGHAckObDvjanVgmuORUnQyyIaAg) = name;  
 }

 public Exam getExam() {  
  return exam;  
 }

 public void setExam(Exam exam) {  
  this.exam = exam;  
 }  
}

package pack2;

import java.io.Serializable;

public class Exam {  
 private int exam\_roll;  
 private int marks;  
 private String subject;

 public int getExam\_roll() {  
  return exam\_roll;  
 }

 public void setExam\_roll(int exam\_roll) {  
  this.exam\_roll = exam\_roll;  
 }

 public int getMarks() {  
  return marks;  
 }

 public void setMarks(int marks) {  
  this.marks = marks;  
 }

 public String getSubject() {  
  return subject;  
 }

 public void setSubject(String subject) {  
  this.subject = subject;  
 }  
}

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-mapping-3.0.dtd&sa=D&sntz=1&usg=AFQjCNEjlBfisNKuNhGBCp0Ww1ceZaDd5Q)">  
<hibernate-mapping>  
 <class name="pack2.Stud">  
  <id name="roll" type="int">  
   <generator class="native" />  
  </id>

  <one-to-one name="exam" class="pack2.Exam" cascade="save-update" />  
  <property name="name" />  
 </class>  
</hibernate-mapping>

use a <many-to-one> mapping with a unique constraint on the foreign key.

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-mapping-3.0.dtd&sa=D&sntz=1&usg=AFQjCNEjlBfisNKuNhGBCp0Ww1ceZaDd5Q)">  
<hibernate-mapping>  
 <class name="pack2.Stud">  
  <id name="roll" type="int">  
   <generator class="native" />  
  </id>

  <many-to-one name="exam" class="pack2.Exam" cascade="save-update" unique="true"/>  
  <property name="name" />  
 </class>  
</hibernate-mapping>

<?xml version="1.0"?>  
<!DOCTYPE hibernate-mapping PUBLIC  
        "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
        "[http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd](http://www.google.com/url?q=http%3A%2F%2Fhibernate.sourceforge.net%2Fhibernate-mapping-3.0.dtd&sa=D&sntz=1&usg=AFQjCNEjlBfisNKuNhGBCp0Ww1ceZaDd5Q)">  
<hibernate-mapping>  
 <class name="pack2.Exam">  
  <id name="exam\_roll">  
   <generator class="native">  
   </generator>  
  </id>  
  <property name="subject" type="string" />  
  <property name="marks" type="int" />  
 </class>  
</hibernate-mapping>

package pack2;

import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;

public class app1 {  
 public static void main(String[] args) {  
  SessionFactory sf = new Configuration().configure(  
    "pack2/hibernate.cfg.xml").buildSessionFactory();  
  Session session = sf.openSession();  
  Stud stud = new Stud();  
  Exam exam = new Exam();  
  exam.setSubject("math");  
  exam.setMarks(50);  
  stud.setName("ram");  
  stud.setExam(exam);  
  session.save(stud);  
  session.beginTransaction().commit();  
  session.close();  
  System.out.println("ADDED");  
 }  
}