**Simple Named Native Query**

* Student.java - see @NamedNativeQuery and @SqlResultSetMapping annotations.
* @NamedNativeQuery(name = "nativeQuery",
* query = "select id as sid, name as sname from Student where name=:name", resultSetMapping = "nativeQueryMapping")
* @SqlResultSetMapping(name = "nativeQueryMapping",
* entities = { @EntityResult(entityClass = Student.class, fields = {
* @FieldResult(name = "id", column = "sid"),
* @FieldResult(name = "name", column = "sname") }) })
* @Cacheable
* @Cache(usage = CacheConcurrencyStrategy.READ\_ONLY)
* @Entity
* @Table(name = "STUDENT", schema = "HIBERNATE")
* public class Student implements Serializable {
* private Integer id;
* private String name;
* @Id
* @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "hib\_seq")
* @SequenceGenerator(name = "hib\_seq", allocationSize = 1)
* @Column(name = "ID")
* public Integer getId() {
* return id;
* }
* public void setId(Integer pId) {
* id = pId;
* }
* @Column(name = "NAME")
* public String getName() {
* return name;
* }
* public void setName(String pName) {
* name = pName;
* }
* @Override
* public String toString() {
* return name;
* }
* }
* test method
* public void test1() {
* Session session = HibernateSessionFactory.openSession();
* Query query = session.getNamedQuery("nativeQuery");
* query.setParameter("name", "Name1");
* List list = query.list();
* System.out.println(list);
* }

* Console output
* select id as sid, name as sname from Student where name=?

[Name1]