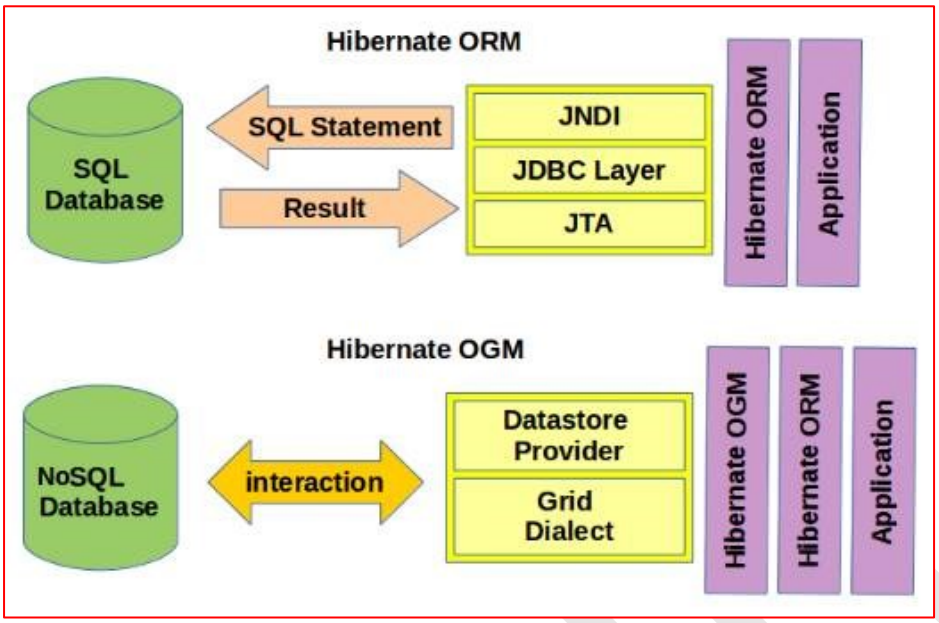
BÀI TẬP PHẦN JPA



PHẦN 1: HƯỚNG DẪN

ORM

Configuration

Drivers

Eclipselink: http://www.eclipse.org/eclipselink/downloads/

Hibernate: http://hibernate.org/orm/

OpenJPA: http://openjpa.apache.org/

…

persistence.xml

File persistence.xml trong thư mục META-INF

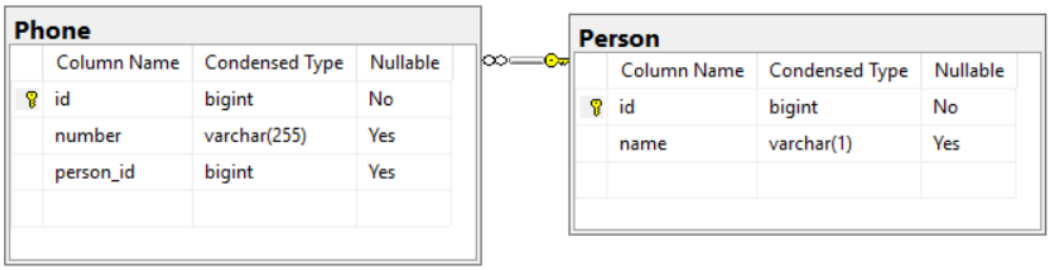
```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
	xmlns="http://xmlns.jcp.org/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
	<persistence-unit name="JPA_Exercises">
		<provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
		<class> your class names with fully package</class>
		<!-- Entities declrare herer -->
		<properties>
			<property name="eclipselink.target-database" value="SQLServer" />
			<property name="javax.persistence.jdbc.url"
				value="jdbc:sqlserver://localhost:1433;databaseName=test" />
			<property name="javax.persistence.jdbc.user" value="sa" />
			<property name="javax.persistence.jdbc.password"
				value="your_password" />
			<property name="javax.persistence.jdbc.driver"
				value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
			<property name="javax.persistence.schema-generation.database.action"
				value="drop-and-create" />
			<property name="eclipselink.logging.level" value="FINE"/>
```

```
            </properties>
        </persistence-unit>
</persistence>
```

Associations

1. @ManyToOne – undirectional

Cơ sở dữ liệu mong muốn được sinh ra



Lớp Person

```
package vovanhai.wordpress.com.ex01;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity(name = "Person")
public  class Person {

    @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;
    @Column(length=150, nullable=false)
    private String name;

    public Person() {
    }

       public Person(String name) {
              this.name = name;
       }

       public Long getId() {
              return id;
       }

       public void setId(Long id) {
              this.id = id;
       }

       public String getName() {
              return name;
       }

       public void setName(String name) {
              this.name = name;
       }
}
```

Lớp Phone

```
      package vovanhai.wordpress.com.ex01;
```

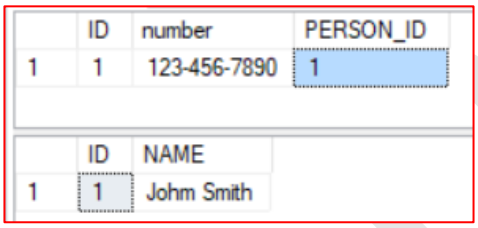
```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

@Entity(name = "Phone")
public class Phone {

    @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;

    @Column(name = "number")
    private String number;

    @ManyToOne()
    private Person person;

    public Phone() {
    }

    public Phone(String number) {
        this.number = number;
    }

    public Long getId() {
        return id;
    }

    public String getNumber() {
        return number;
    }

    public Person getPerson() {
        return person;
    }

    public void setPerson(Person person) {
        this.person = person;
    }
}
```

Driver:

```
package vovanhai.wordpress.com.ex01;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;

public class Test {

 public static void main(String[] args) {
        EntityManagerFactory fac =
Persistence.createEntityManagerFactory("JPA_Exercises");
        EntityManager entityManager = fac.createEntityManager();
        EntityTransaction transaction = entityManager.getTransaction();
        try {
              transaction.begin();

              Person person = new Person("Johm Smith");
              entityManager.persist( person );
```

```
                    Phone phone = new Phone( "123-456-7890" );
                    phone.setPerson( person );
                    entityManager.persist( phone );

                    transaction.commit();
                } catch (Exception e) {
                    transaction.rollback();
                    e.printStackTrace();
                }
            }
        }
```

Kết quả thêm vào cơ sở dữ liệu như sau

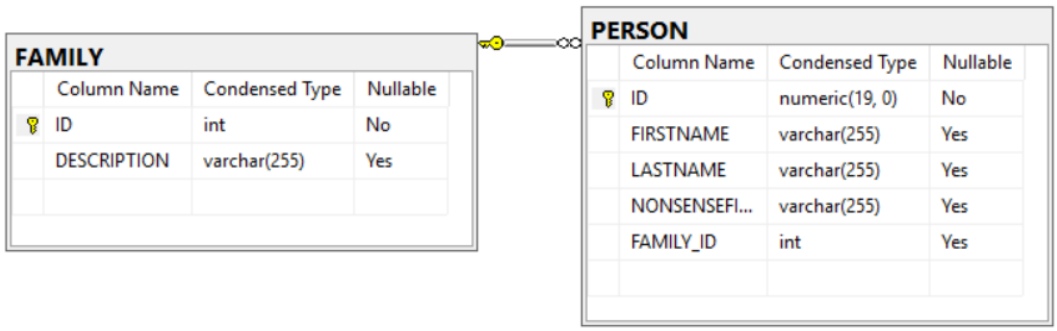


	ID	number	PERSON_ID
1	1	123-456-7890	1

	ID	NAME
1	1	Johm Smith

2. @OneToMany - Bidirectional

Cơ sở dữ liệu mong muốn được sinh ra




```java
package vovanhai.wordpress.com.ex02;
import java.util.ArrayList;
import java.util.List;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;

@Entity
public class Family {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;
    private String description;

    @OneToMany(cascade=CascadeType.ALL, mappedBy = "family")
    private final List<Person> members = new ArrayList<Person>();

    public Family() {
       }

    public Family(String description) {
            this.description = description;
       }

      public int getId() {
       return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public List<Person> getMembers() {
        return members;
    }
}
```

```java
package vovanhai.wordpress.com.ex02;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Transient;

@Entity
public class Person {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String firstName;
    private String lastName;

    @Transient
    private String nonsenseField = "";

    @ManyToOne
    private Family family;

    public Person() {
       }
    public Person(String firstName, String lastName) {
            this.firstName = firstName;
            this.lastName = lastName;
       }

    public long getId() {
            return id;
       }
       public void setId(long id) {
            this.id = id;
       }
       public String getFirstName() {
         return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    // Leave the standard column name of the table
    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }


    public Family getFamily() {
        return family;
    }

    public void setFamily(Family family) {
        this.family = family;
    }


    public String getNonsenseField() {
        return nonsenseField;
```

```
    }

    public void setNonsenseField(String nonsenseField) {
        this.nonsenseField = nonsenseField;
    }
}
```

Lớp driver

```
package vovanhai.wordpress.com.ex02;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;

public class Test {

	public static void main(String[] args) {
		EntityManagerFactory fac =
Persistence.createEntityManagerFactory("JPA_Exercises");
		EntityManager entityManager = fac.createEntityManager();
		EntityTransaction transaction = entityManager.getTransaction();
		try {
			transaction.begin();

			Family f=new Family("Johm happy family");

			Person person1 = new Person("John", "Smith");
			Person person2 = new Person("Harry", "Potter");
			person1.setFamily(f);
			person2.setFamily(f);

			f.getMembers().add(person1);
			f.getMembers().add(person2);

			entityManager.persist( f );

			transaction.commit();
		} catch (Exception e) {
			transaction.rollback();
			e.printStackTrace();
		}
	}
}
```

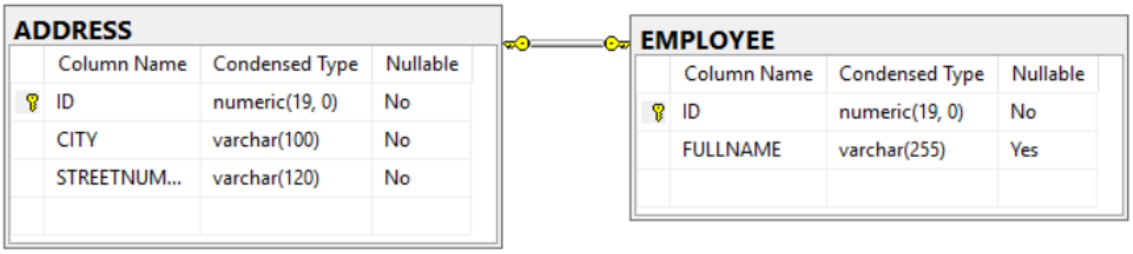
Kết quả

	ID	DESCRIPTION
1	1	Johm happy family

	ID	FIRSTNAME	LASTNAME	FAMILY_ID
1	1	Harry	Potter	1
2	2	John	Smith	1

3. @OneToOne

Cơ sở dữ liệu mong muốn được sinh ra



Lớp Employee

```
package vovanhai.wordpress.com.ex03;

import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;

@Entity
public class Employee {
	@Id @GeneratedValue(strategy=GenerationType.IDENTITY)
	private long id;

	private String fullname;

	@OneToOne(cascade=CascadeType.ALL,fetch=FetchType.EAGER)
	@JoinColumn(name="id",insertable=false,updatable=false)
	private Address address;

	public Employee() {
	}
	public Employee(String fullname) {
		this.fullname = fullname;
	}
	public long getId() {
		return id;
	}

	public String getFullname() {
		return fullname;
	}
	public void setFullname(String fullname) {
		this.fullname = fullname;
	}
	public Address getAddress() {
		return address;
	}
	public void setAddress(Address address) {
		this.address = address;
	}
}
```

Lớp Address

```
package vovanhai.wordpress.com.ex03;

import javax.persistence.CascadeType;
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class Address {
	@Id @GeneratedValue(strategy=GenerationType.IDENTITY)
	private long id;

	@Column(length=120,nullable=false)
	private String streetNumber;

	@Column(length=100,nullable=false)
	private String city;

	@OneToOne(mappedBy="address",cascade=CascadeType.ALL,fetch=FetchType.EAGER)
	private Employee owner;

	public Address() {
	}

	public Address(String streetNumber, String city) {
		this.streetNumber = streetNumber;
		this.city = city;
	}

	public String getStreetNumber() {
		return streetNumber;
	}

	public void setStreetNumber(String streetNumber) {
		this.streetNumber = streetNumber;
	}

	public String getCity() {
		return city;
	}

	public void setCity(String city) {
		this.city = city;
	}

	public Employee getOwner() {
		return owner;
	}

	public void setOwner(Employee owner) {
		this.owner = owner;
	}

	public long getId() {
		return id;
	}
}
```

Lớp Driver

```
package vovanhai.wordpress.com.ex03;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
```

```
import javax.persistence.Persistence;

public class Test {

	public static void main(String[] args) {
		EntityManagerFactory fac =
				Persistence.createEntityManagerFactory("JPA_Exercises");
		EntityManager entityManager = fac.createEntityManager();
		EntityTransaction transaction = entityManager.getTransaction();
		try {
			transaction.begin();

			Employee emp=new Employee("Than Thi Det");
			Address add=new Address("12 Nguyen Van Bao, F4,"
					+ " Go Vap", "Ho CHi Minh");
			emp.setAddress(add);

			entityManager.persist(emp);

			transaction.commit();
		} catch (Exception e) {
			transaction.rollback();
			e.printStackTrace();
		}
	}
}
```

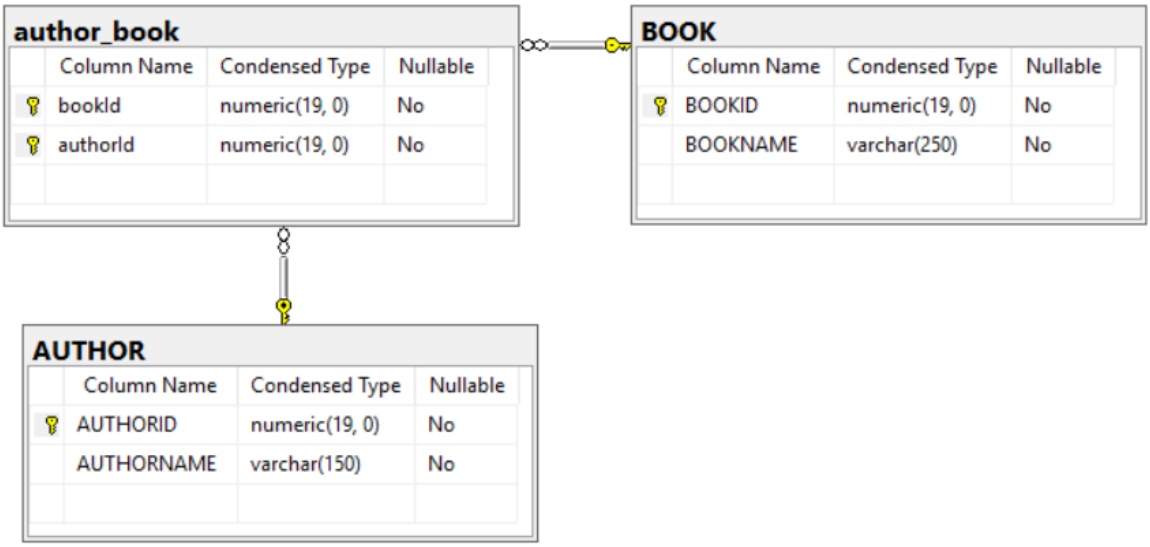
Kết quả

	ID	FULLNAME
1	1	Than Thi Det

	ID	CITY	STREETNUMBER
1	1	Ho CHi Minh	12 Nguyen Van Bao, F4, Go Vap

4. @ManyToMany

Cơ sở dữ liệu mong muốn



Lớp Book

```
package vovanhai.wordpress.com.ex04;

import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.JoinTable;
import javax.persistence.ManyToMany;

@Entity
public class Book
{
	private Long bookId;
	private String bookName;
	private Set<Author> authors;

	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	public Long getBookId()
	{
		return bookId;
	}
	public void setBookId(Long bookId)
	{
		this.bookId = bookId;
	}

	@Column(length=250, nullable=false)
	public String getBookName()
	{
		return bookName;
	}
	public void setBookName(String bookName)
	{
		this.bookName = bookName;
	}
```

```
	@ManyToMany(cascade=CascadeType.ALL)
	@JoinTable(
			name="author_book",
	joinColumns=@JoinColumn(name="bookId",referencedColumnName="bookId"),
			inverseJoinColumns=@JoinColumn(name="authorId",
			referencedColumnName="authorId")
	)
	public Set<Author> getAuthors()
	{
		return authors;
	}
	public void setAuthors(Set<Author> authors)
	{
		this.authors = authors;
	}
}
```

Lớp Author

```
package vovanhai.wordpress.com.ex04;

import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;

@Entity
public class Author
{
	private Long authorId;
	private String authorName;
	private Set<Book> books;

	@Id  @GeneratedValue(strategy=GenerationType.IDENTITY)
	public Long getAuthorId()
	{
		return authorId;
	}
	public void setAuthorId(Long authorId)
	{
		this.authorId = authorId;
	}

	@Column(length=150, nullable=false)
	public String getAuthorName()
	{
		return authorName;
	}
	public void setAuthorName(String authorName)
	{
		this.authorName = authorName;
	}
	@ManyToMany(cascade=CascadeType.ALL, mappedBy="authors")
	public Set<Book> getBooks() {
		return books;
	}
	public void setBooks(Set<Book> books) {
		this.books = books;
	}
```

```
}
```

Lớp Driver

```
package vovanhai.wordpress.com.ex04;

import java.util.HashSet;
import java.util.Set;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.Persistence;

public class Test {
	public static void main(String[] args) {
		Set<Author> authors = new HashSet<Author>();

		Author author1 = new Author();
		author1.setAuthorName("Trevor Page");
		authors.add(author1);

		Author author2 = new Author();
		author2.setAuthorName("John Doe");

		authors.add(author1);
		authors.add(author2);

		Book book1 = new Book();
		book1.setBookName("How to Program with Java");

		book1.setAuthors(authors);

		EntityManagerFactory fac =
Persistence.createEntityManagerFactory("JPA_Exercises");
		EntityManager entityManager = fac.createEntityManager();
		EntityTransaction transaction = entityManager.getTransaction();
		try {
			transaction.begin();

			entityManager.persist(book1);

			transaction.commit();
		} catch (Exception e) {
			transaction.rollback();
			e.printStackTrace();
		}
	}
}
```

Kết quả

	BOOKID	BOOKNAME
1	1	How to Program with Java

	AUTHORID	AUTHORNAME
1	1	Trevor Page
2	2	John Doe

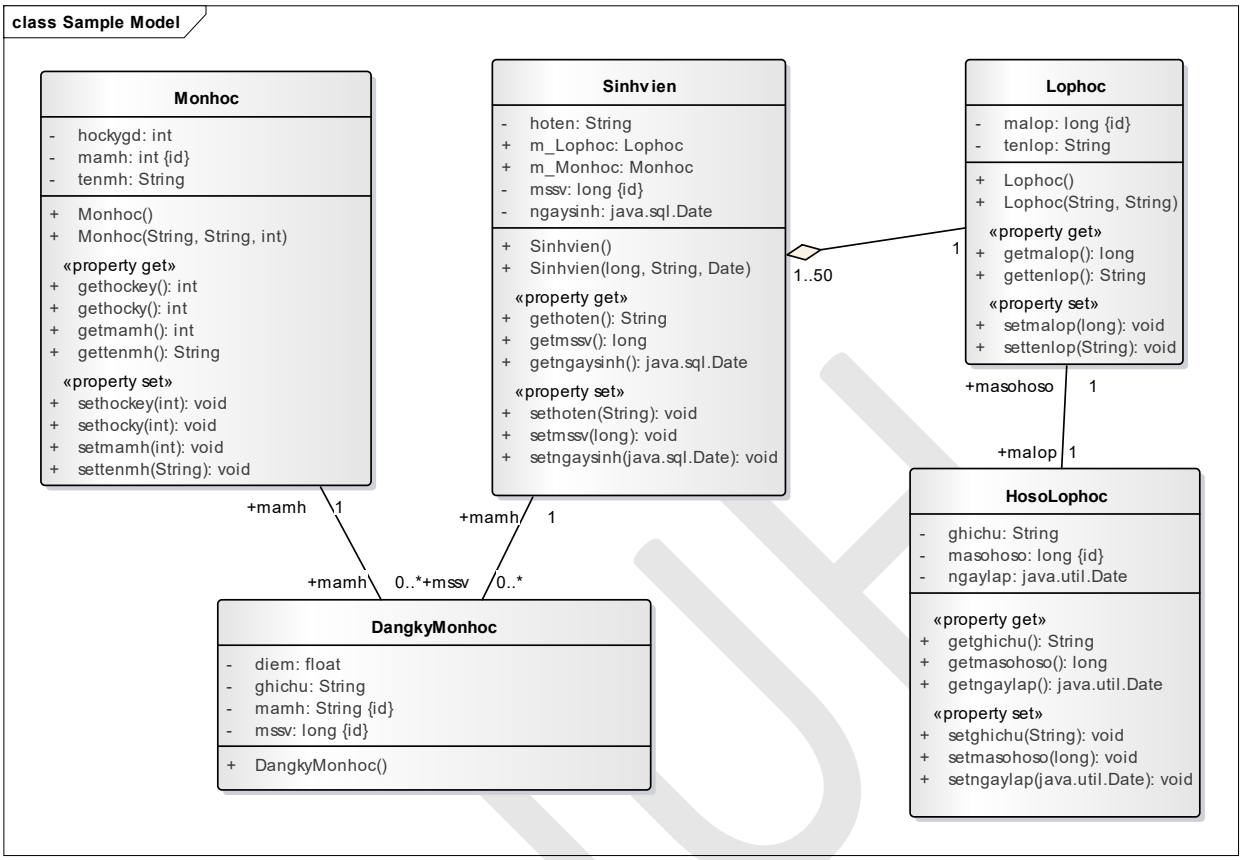
	bookId	authorId
1	1	1
2	1	2

Strategy

1. Mô tả các lớp đối tượng như bình thường (POJO)
2. Thêm các JPA annotations (thực thể)
3. Thêm các mối quan hệ (Associations)
4. Tùy theo các yêu cầu, thêm các named query

Hướng dẫn ORM tổng hợp

Cho class diagram sau:



Thực hiện việc ánh xạ các lớp trong mô hình trên thành các JPA entities.

Code

Lớp Lophoc

```
package vovanhai.wordpress.com.entities;

import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name="Lophoc")
public class Lophoc {
 @Id
 @GeneratedValue(strategy=GenerationType.IDENTITY)
 private long malop;
 private String tenlop;

 //lớp học <1:1> hồ sơ lớp học
 @OneToOne
 private HosoLophoc hosoLH;

 //Lớp học <1:n> sinh viên
 @OneToMany(mappedBy="lophoc",cascade=CascadeType.ALL,fetch=FetchType.EAGER)
```

```
	private Set<Sinhvien> dssv;


	public Lophoc() {
	}

	public Lophoc(String tenlop) {
		this.tenlop = tenlop;
	}
	public HosoLophoc getHosoLH() {
		return hosoLH;
	}

	public void setHosoLH(HosoLophoc hosoLH) {
		this.hosoLH = hosoLH;
	}
	public long getMalop() {
		return malop;
	}

	public void setMalop(long malop) {
		this.malop = malop;
	}

	public String getTenlop() {
		return tenlop;
	}

	public void setTenlop(String tenlop) {
		this.tenlop = tenlop;
	}
	@Override
	public String toString() {
		return tenlop;
	}
}
```

Sinhvien

```
package vovanhai.wordpress.com.entities;

import java.util.Date;
import java.util.Set;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
@Entity
public class Sinhvien {
	@Id @GeneratedValue(strategy=GenerationType.IDENTITY)
	private long mssv;
	private String hoten;
	@Temporal(TemporalType.DATE)
	private Date ngaysinh;

	//sinh viên <n:1> Lớp học
	@ManyToOne(fetch=FetchType.EAGER)
	private Lophoc lophoc;

	@OneToMany(mappedBy="pk.sinhvien")
	private Set<DangkyMonhoc>svdkmh;
```

```
	public Set<DangkyMonhoc> getSvdkmh() {
		return svdkmh;
	}

	public void setSvdkmh(Set<DangkyMonhoc> svdkmh) {
		this.svdkmh = svdkmh;
	}

	public Lophoc getLophoc() {
		return lophoc;
	}

	public void setLophoc(Lophoc lophoc) {
		this.lophoc = lophoc;
	}

	public Sinhvien() {
	}

	public Sinhvien(String hoten, Date ngaysinh) {
		this.hoten = hoten;
		this.ngaysinh = ngaysinh;
	}

	public long getMssv() {
		return mssv;
	}

	public void setMssv(long mssv) {
		this.mssv = mssv;
	}

	public String getHoten() {
		return hoten;
	}

	public void setHoten(String hoten) {
		this.hoten = hoten;
	}

	public Date getNgaysinh() {
		return ngaysinh;
	}

	public void setNgaysinh(Date ngaysinh) {
		this.ngaysinh = ngaysinh;
	}

	@Override
	public String toString() {
		return hoten;
	}
}
```

```
package vovanhai.wordpress.com.entities;

import java.util.Set;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
```

```
import javax.persistence.OneToMany;

@Entity
public class Monhoc {
 @Id
 @GeneratedValue(strategy=GenerationType.IDENTITY)
 private long mamh;
 private String tenmh;
 private int hockygd;

 //Môn học <n:n> sinh viên
 @OneToMany(mappedBy="pk.monhoc")
 private Set<DangkyMonhoc> mhdk;

 public Monhoc() {
 }

 public Monhoc( String tenmh, int hockygd) {
        this.tenmh = tenmh;
        this.hockygd = hockygd;
 }

 public Set<DangkyMonhoc> getMhdk() {
        return mhdk;
 }

 public void setMhdk(Set<DangkyMonhoc> mhdk) {
        this.mhdk = mhdk;
 }

 public long getMamh() {
        return mamh;
 }


 public void setMamh(long mamh) {
        this.mamh = mamh;
 }


  public String getTenmh() {
        return tenmh;
 }
 public void setTenmh(String tenmh) {
        this.tenmh = tenmh;
 }
 public int getHockygd() {
        return hockygd;
 }
 public void setHockygd(int hockygd) {
        this.hockygd = hockygd;
 }

}
```

```
package vovanhai.wordpress.com.entities;

import java.util.Date;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
```

```java
@Entity
public class HosoLophoc {
 @Id
 @GeneratedValue(strategy=GenerationType.IDENTITY)
 private long masohoso;
 private Date ngaylap;
 private String ghichu;

 //hồ sơ lớp học <1:1> lớp học
 @OneToOne(mappedBy="hosoLH")
 private Lophoc lophoc;

 public HosoLophoc() {
 }
 public HosoLophoc( Date ngaylap, String ghichu) {
        this.ngaylap = ngaylap;
        this.ghichu = ghichu;
 }

 public Lophoc getLophoc() {
        return lophoc;
 }
 public void setLophoc(Lophoc lophoc) {
        this.lophoc = lophoc;
 }
 public long getMasohoso() {
        return masohoso;
 }
 public void setMasohoso(long masohoso) {
        this.masohoso = masohoso;
 }
 public Date getNgaylap() {
        return ngaylap;
 }
 public void setNgaylap(Date ngaylap) {
        this.ngaylap = ngaylap;
 }
 public String getGhichu() {
        return ghichu;
 }
 public void setGhichu(String ghichu) {
        this.ghichu = ghichu;
 }
 @Override
 public String toString() {
        return masohoso+"";
 }
}
```

```java
package vovanhai.wordpress.com.entities;

import java.io.Serializable;
import javax.persistence.CascadeType;
import javax.persistence.Embeddable;
import javax.persistence.ManyToOne;

@Embeddable
public class DangkyMonhoc_PK implements Serializable{
 @ManyToOne(cascade=CascadeType.ALL)
 private Monhoc monhoc;
 @ManyToOne(cascade=CascadeType.ALL)
 private Sinhvien sinhvien;

 public DangkyMonhoc_PK() {
```

```
	}

	public DangkyMonhoc_PK(Monhoc monhoc, Sinhvien sinhvien) {
		this.monhoc = monhoc;
		this.sinhvien = sinhvien;
	}

	public Monhoc getMonhoc() {
		return monhoc;
	}

	public void setMonhoc(Monhoc monhoc) {
		this.monhoc = monhoc;
	}
	public Sinhvien getSinhvien() {
		return sinhvien;
	}

	public void setSinhvien(Sinhvien sinhvien) {
		this.sinhvien = sinhvien;
	}

	@Override
	public int hashCode() {
		final int prime = 31;
		int result = 1;
		result = prime * result + ((monhoc == null) ? 0 : monhoc.hashCode());
		result = prime * result + ((sinhvien == null) ? 0 :
sinhvien.hashCode());
		return result;
	}

	@Override
	public boolean equals(Object obj) {
		if (this == obj)
			return true;
		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		DangkyMonhoc_PK other = (DangkyMonhoc_PK) obj;
		if (monhoc == null) {
			if (other.monhoc != null)
				return false;
		} else if (!monhoc.equals(other.monhoc))
			return false;
		if (sinhvien == null) {
			if (other.sinhvien != null)
				return false;
		} else if (!sinhvien.equals(other.sinhvien))
			return false;
		return true;
	}
}
```

```
package vovanhai.wordpress.com.entities;

import javax.persistence.AssociationOverride;
import javax.persistence.AssociationOverrides;
import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.JoinColumn;
import javax.persistence.Transient;
```

```java
@Entity
@AssociationOverrides({
 @AssociationOverride(name = "pk.sinhvien",
             joinColumns = @JoinColumn(name = "sinhvien")),
 @AssociationOverride(name = "pk.monhoc",
 joinColumns = @JoinColumn(name = "monhoc")) })
public class DangkyMonhoc {
 // composite-id key
 @EmbeddedId
 private DangkyMonhoc_PK pk=new DangkyMonhoc_PK();

 private float diem;
 private String ghichu;

 public DangkyMonhoc() {
 }

 public DangkyMonhoc(DangkyMonhoc_PK pk, float diem, String ghichu) {
       this.pk = pk;
       this.diem = diem;
       this.ghichu = ghichu;
 }

 public DangkyMonhoc_PK getPk() {
       return pk;
 }

 public void setPk(DangkyMonhoc_PK pk) {
       this.pk = pk;
 }

 public float getDiem() {
       return diem;
 }

 public void setDiem(float diem) {
       this.diem = diem;
 }

 public String getGhichu() {
       return ghichu;
 }

 public void setGhichu(String ghichu) {
       this.ghichu = ghichu;
 }
 @Transient
 public Monhoc getMonhoc() {
       return getPk().getMonhoc();
 }

 public void setMonhoc(Monhoc monhoc) {
       getPk().setMonhoc(monhoc);
 }
 @Transient
 public Sinhvien getSinhvien() {
       return  getPk().getSinhvien();
 }

 public void setSinhvien(Sinhvien sinhvien) {
       getPk().setSinhvien(sinhvien);
 }
}
```

File persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
            http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd"
  version="2.1">
  <persistence-unit name="demo-orm" transaction-type="RESOURCE_LOCAL">
        <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>

        <class>vovanhai.wordpress.com.entities.Lophoc</class>
        <class>vovanhai.wordpress.com.entities.Monhoc</class>
        <class>vovanhai.wordpress.com.entities.Sinhvien</class>
        <class>vovanhai.wordpress.com.entities.HosoLophoc</class>
        <class>vovanhai.wordpress.com.entities.DangkyMonhoc</class>
        <properties>
              <property name="eclipselink.target-database" value="SQLServer" />
              <property name="javax.persistence.jdbc.url"
                    value="jdbc:sqlserver://localhost:1433;databaseName=test" />
              <property name="javax.persistence.jdbc.user" value="sa" />
              <property name="javax.persistence.jdbc.password" value="12345678" />
              <property name="javax.persistence.jdbc.driver"
                    value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
              <property name="javax.persistence.schema-generation.database.action"
                    value="drop-and-create" />
        </properties>
  </persistence-unit>
</persistence>
```

Code cho phần daos

```
package vovanhai.wordpress.com.daos;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class MyEntityManagerFactory {
  private static MyEntityManagerFactory instance;

  private EntityManager entityManager;

  private MyEntityManagerFactory() {
        EntityManagerFactory fac =
Persistence.createEntityManagerFactory("demo-orm");
        entityManager=fac.createEntityManager();
  }
  public synchronized static MyEntityManagerFactory getInstance() {
        if(instance==null)
              instance=new MyEntityManagerFactory();
        return instance;
  }

  public EntityManager getEntityManager() {
        return entityManager;
  }


}
```

```
package vovanhai.wordpress.com.daos;

import javax.persistence.EntityManager;

public abstract class AbstractCRUD<T> {
  protected EntityManager em;

  public AbstractCRUD() {
```

```
            em=MyEntityManagerFactory.getInstance().getEntityManager();
      }

      public void persist(T t) {
            em.persist(t);
      }

      public void remove(T t) {
            em.remove(em.merge(t));
      }

      public void merge(T t) {
            em.merge(t);
      }
}
```

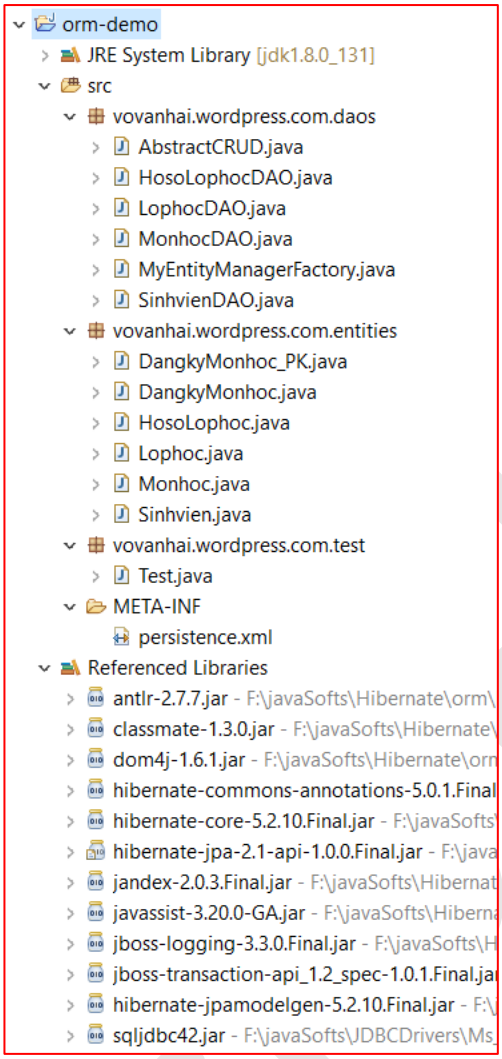
```
package vovanhai.wordpress.com.daos;

import vovanhai.wordpress.com.entities.HosoLophoc;

public class HosoLophocDAO extends AbstractCRUD<HosoLophoc> {
      public HosoLophoc findById(long id) {
            return em.find(HosoLophoc.class, id);
      }
}
```

```
package vovanhai.wordpress.com.daos;

import vovanhai.wordpress.com.entities.Lophoc;

public class LophocDAO extends AbstractCRUD<Lophoc> {
      public Lophoc findById(long id) {
            return em.find(Lophoc.class, id);
      }
}
```

```
package vovanhai.wordpress.com.daos;

import vovanhai.wordpress.com.entities.Monhoc;

public class MonhocDAO extends AbstractCRUD<Monhoc> {
      public Monhoc findById(long id) {
            return em.find(Monhoc.class, id);
      }
}
```

```
package vovanhai.wordpress.com.daos;

import vovanhai.wordpress.com.entities.Sinhvien;

public class SinhvienDAO extends AbstractCRUD<Sinhvien> {
      public Sinhvien findById(long id) {
            return em.find(Sinhvien.class, id);
      }
}
```

Cấu trúc code



Code test

```
package vovanhai.wordpress.com.test;

import java.util.GregorianCalendar;
import javax.persistence.EntityManager;
import javax.persistence.EntityTransaction;
import vovanhai.wordpress.com.daos.HosoLophocDAO;
import vovanhai.wordpress.com.daos.LophocDAO;
import vovanhai.wordpress.com.daos.MonhocDAO;
import vovanhai.wordpress.com.daos.MyEntityManagerFactory;
import vovanhai.wordpress.com.entities.HosoLophoc;
import vovanhai.wordpress.com.entities.Lophoc;
import vovanhai.wordpress.com.entities.Monhoc;

public class Test {
	public static void main(String[] args) {
		EntityManager em =
MyEntityManagerFactory.getInstance().getEntityManager();
		EntityTransaction trans = em.getTransaction();
		try {
			trans.begin();
			Lophoc lh=new Lophoc("Lop dhth 11A");
			HosoLophoc hslh=new HosoLophoc(new GregorianCalendar(2010,
12, 24).getTime(),
					"ghi chu cho lop hoc");
			lh.setHosoLH(hslh);
			LophocDAO lhdao=new LophocDAO();
			HosoLophocDAO hsdao=new HosoLophocDAO();
```

```
                hsdao.persist(hslh);
                lhdao.persist(lh);
                Monhoc mh=new Monhoc("thuc hanh", 1);
                new MonhocDAO().persist(mh);
                trans.commit();
            } catch (Exception e) {
                trans.rollback();
                e.printStackTrace();
            }
            LophocDAO x=new LophocDAO();
            System.out.println(x.findById(1));

            System.exit(1);
        }
}
```

OGM

Configuration

1. Sử dụng JPA

Dùng JPA để kết nối, phải cấu hình persistence.xml như sau

```
<?xml version="1.0" encoding="utf-8"?>

<persistence xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">
    <persistence-unit name="ogm-jpa" transaction-type="RESOURCE_LOCAL">
        <provider>org.hibernate.ogm.jpa.HibernateOgmPersistence</provider>
            <class>entities.Lophoc</class>
            <class>entities.Sinhvien</class>
            <class>entities.Monhoc</class>
        <properties>
            <property name="hibernate.ogm.datastore.provider"
      value="org.hibernate.ogm.datastore.mongodb.impl.MongoDBDatastoreProvider"
/>
            <property name="hibernate.ogm.datastore.database" value="demodb" />
             <property name="hibernate.ogm.mongodb.host" value="localhost"/>
             <property name="hibernate.ogm.datastore.create_database"
value="true"/>
        </properties>
    </persistence-unit>
</persistence>
```

Để sử dụng cấu hình này, ta dùng EntityManager

```
EntityManagerFactory emf =Persistence.createEntityManagerFactory("ogm-jpa");
EntityManager em = emf.createEntityManager();
```

2. Sử dụng Hibernate OGM APIs

```
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.AvailableSettings;
import org.hibernate.ogm.OgmSession;
import org.hibernate.ogm.OgmSessionFactory;
import org.hibernate.ogm.boot.OgmSessionFactoryBuilder;
import org.hibernate.ogm.cfg.OgmProperties;
import org.hibernate.ogm.datastore.mongodb.MongoDB;

import entities.Lophoc;
import entities.Monhoc;
import entities.Sinhvien;

public class MyOgmSessionFactory {
       private static MyOgmSessionFactory factory;
       private OgmSession ogmSession;
       private MyOgmSessionFactory() {
              init();
       }

       public synchronized static MyOgmSessionFactory getInstance() {
              if(factory==null)
                     factory=new MyOgmSessionFactory();
              return factory;
       }
```

```
	public OgmSession getOgmSession() {
		return ogmSession;
	}

	private void init() {
		// create the StandardServiceRegistry
		StandardServiceRegistry registry = new
StandardServiceRegistryBuilder()
	.applySetting(OgmProperties.ENABLED, true)
	// assuming you are using JDBCTransactionFactory
	.applySetting(AvailableSettings.TRANSACTION_COORDINATOR_STRATEGY,
		"org.hibernate.transaction.JDBCTransactionFactory")
	// configure current session context
	.applySetting(AvailableSettings.CURRENT_SESSION_CONTEXT_CLASS, "thread")
	// assuming MongoDB as the backend
	.applySetting(OgmProperties.DATASTORE_PROVIDER,
				MongoDB.DATASTORE_PROVIDER_NAME)
	.applySetting(OgmProperties.DATABASE, "demodb")
	.applySetting(OgmProperties.CREATE_DATABASE, "true")
	.applySetting(OgmProperties.HOST, "127.0.0.1:27017")
	.build();

	OgmSessionFactory ogmSessionFactory= new MetadataSources(registry)
				.addAnnotatedClass(Sinhvien.class)
				.addAnnotatedClass( Lophoc.class )
				.addAnnotatedClass( Monhoc.class )
				.buildMetadata()
				.getSessionFactoryBuilder()
				.unwrap(OgmSessionFactoryBuilder.class)
				.build();
		ogmSession = ogmSessionFactory.openSession();
	}

	public void closeSession() {
		ogmSession.close();
	}
}
```

Để sử dụng session, ta có ví dụ như sau

```
SQLQuery query = ogmSession
			.createNativeQuery("{}") //lấy tất cả
			.addEntity(Sinhvien.class);
List<Sinhvien> lst = query.list();
System.out.println(lst);
```

3. Chuyển đổi

Lấy OgmSession từ EntityManager

```
OgmSession session = (OgmSession)em.getDelegate();
Hoặc
OgmSession session = em.unwrap(OgmSession.class);
```

Mapping

1. Id do người dùng định nghĩa

Ánh xạ lớp "News" thành collection "news" với id do người dùng qui định

```
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Table(name="news")
```

```
@Entity public class News {
	@Id private String id;
	private String title;

	//constuctor, setter/getter

}
```

Test

```
	OgmSession session = ogmSessionFactory.openSession();
	Transaction trans = session.getTransaction();
	try {
		trans.begin();
		session.persist(new News("n1001","how to become a good men"));
		trans.commit();
	} catch (Exception e) {
		trans.rollback();
		e.printStackTrace();
	}
```

Kết quả

```
> db.news.find()
{ "_id" : "n1001", "title" : "how to become a good men" }
>
```

2. Id là ObjectID

Nên dùng id kiểu này

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

import org.hibernate.annotations.Type;

@Table(name="news")
@Entity public class News {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Type(type="objectid")
	private String id;
	private String title;
	public News() {}
	public News(String title) {
		this.title = title;
	}
// setter/getter
}
```

Hoặc

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Table(name="news")
@Entity public class News {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	private org.bson.types.ObjectId id;
	private String title;
	public News() {}
	public News(String title) {
		this.title = title;
	}
	public org.bson.types.ObjectId getId() {
		return id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
}
```

Kết quả

```
> db.news.find()
{ "_id" : ObjectId("59c1e39d8d43e11a3c6699ec"), "title" : "how to become a good men" }
>
```

3. @EmbededId

```
package vovanhai.wordpresss.com;

import java.io.Serializable;
import javax.persistence.Embeddable;
@Embeddable
public class NewsID implements Serializable{
	private String author;
	private String title;
	public NewsID() {}
	public NewsID(String author, String title) {
		this.author = author;
		this.title = title;
	}
//setters, getters
//hashCode, equals
	}
```

```
package vovanhai.wordpresss.com;

import javax.persistence.EmbeddedId;
import javax.persistence.Entity;
import javax.persistence.Table;

@Table(name="news")
@Entity public class News {
	@EmbeddedId
	private NewsID newsid;
	private String content;
	public News() {}
	public News(NewsID newsid, String content) {
		this.newsid = newsid;
		this.content = content;
	}
//setters/getters
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();
	NewsID nid=new NewsID("Dr Jackson", "Make it easy");
	session.persist(new News(nid,"paper's content..."));
	trans.commit();
	} catch (Exception e) {
		trans.rollback();
		e.printStackTrace();
	}
```

Kết quả

```
> db.news.find().pretty()
{
	"_id" : {
		"author" : "Dr Jackson",
		"title" : "Make it easy"
	},
	"content" : "paper's content..."
}
>
```

4. Embedded objects and collections

```java
package vovanhai.wordpresss.com;

import javax.persistence.Embedded;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

import org.hibernate.annotations.Type;

@Entity
@Table(name="news")
public class News {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Type(type="objectid")
	private String id;
	private String title;
	@Embedded
	private NewsPaper paper;

	public News() {	}
	public News(String title, NewsPaper paper) {
		this.title = title;
		this.paper = paper;
	}
	public String getId() {
		return id;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public NewsPaper getPaper() {
		return paper;
	}
	public void setPaper(NewsPaper paper) {
		this.paper = paper;
	}
}
```

```java
package vovanhai.wordpresss.com;

import javax.persistence.Embeddable;

@Embeddable
public class NewsPaper {
	private String owner;
	private String name;

	public NewsPaper() {
	}

	public NewsPaper(String owner, String name) {
		this.owner = owner;
		this.name = name;
	}
// getters, setters ...
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();
	NewsPaper np=new NewsPaper("Dr Jackson", "Make it easy");
	session.persist(new News("paper's content...",np));
	trans.commit();
} catch (Exception e) {
	trans.rollback();
	e.printStackTrace();
}
```

Kết quả

```
> db.news.find().pretty()
{
	"_id" : ObjectId("59c1e9b58d43e12658b437d5"),
	"paper" : {
		"name" : "Make it easy",
		"owner" : "Dr Jackson"
	},
	"title" : "paper's content..."
}
>
```

5. Embedded collections

```
package vovanhai.wordpresss.com;

import java.util.List;

import javax.persistence.ElementCollection;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

import org.hibernate.annotations.Type;

@Entity
@Table(name="news")
public class News {
	@Id
	@GeneratedValue(strategy=GenerationType.IDENTITY)
	@Type(type="objectid")
	private String id;
	private String title;

	@ElementCollection
	private List<NewsPaper> papers;

	public News() {	}
	public News(String title, List<NewsPaper> papers) {
		this.title = title;
		this.papers = papers;
	}
	public String getTitle() {
		return title;
	}
	public void setTitle(String title) {
		this.title = title;
	}
	public List<NewsPaper> getPapers() {
		return papers;
	}
	public void setPapers(List<NewsPaper> papers) {
		this.papers = papers;
	}
	public void setId(String id) {
		this.id = id;
	}
}
```

```
package vovanhai.wordpresss.com;

import javax.persistence.Embeddable;

@Embeddable
public class NewsPaper {
	private String owner;
	private String name;

	public NewsPaper() {}

	public NewsPaper(String owner, String name) {
		this.owner = owner;
		this.name = name;
	}
```

```
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getOwner() {
		return owner;
	}
	public void setOwner(String owner) {
		this.owner = owner;
	}
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();
	NewsPaper np1=new NewsPaper("Dr Jackson", "Java Programming");
	NewsPaper np2=new NewsPaper("John Allen", "MongoDB intro");
	NewsPaper np3=new NewsPaper("Darling, Jr.", "We are friends");

	ArrayList<NewsPaper> lst=new ArrayList<>();
	lst.add(np1);
	lst.add(np2);
	lst.add(np3);

	session.persist(new News("paper's content...",lst));
	trans.commit();
} catch (Exception e) {
	trans.rollback();
	e.printStackTrace();
}
```

Kết quả

```
> db.news.find().pretty()
{
	"_id" : ObjectId("59c1f1c28d43e103b43f451f"),
	"title" : "paper's content...",
	"papers" : [
		{
			"owner" : "Dr Jackson",
			"name" : "Java Programming"
		},
		{
			"owner" : "John Allen",
			"name" : "MongoDB intro"
		},
		{
			"owner" : "Darling, Jr.",
			"name" : "We are friends"
		}
	]
}
>
```

Associations

1. One-To-One Undirectional

```java
package vovanhai.wordpresss.com.associations;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name="vehicle")
public class Vehicle {

    @Id
    private String id;
    private String brand;
    @OneToOne
    @JoinColumn(name="license_plate")
    private LicensePlate vehicule;

    public Vehicle() {
       }

       public Vehicle(String id, String brand, LicensePlate vehicule) {
              this.id = id;
              this.brand = brand;
              this.vehicule = vehicule;
       }

       public String getId() {
              return id;
       }

       public void setId(String id) {
              this.id = id;
       }

       public String getBrand() {
              return brand;
       }

       public void setBrand(String brand) {
              this.brand = brand;
       }

       public LicensePlate getVehicule() {
              return vehicule;
       }

       public void setVehicule(LicensePlate vehicule) {
              this.vehicule = vehicule;
       }
}
```

```java
package vovanhai.wordpresss.com.associations;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
```

```java
@Table(name="license_plate")
public class LicensePlate {
	@Id
	private String id;
	private String number;

	public LicensePlate() {
	}

	public LicensePlate(String id, String number) {
		this.id = id;
		this.number = number;
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getNumber() {
		return number;
	}

	public void setNumber(String number) {
		this.number = number;
	}
}
```

Kết quả

```
> db.vehicle.find().pretty()
{ "_id" : "V_02", "license_plate" : "L_001", "brand" : "Mercedes" }
>
```

2. One-To-One Bidirectional

```java
package vovanhai.wordpresss.com.asso.one.bidirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class Husband {

	@Id
	private String id;
	private String name;

	@OneToOne
	private Wife wife;

	public Husband() {}

	public Husband(String id, String name) {
		this.id = id;
		this.name = name;
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public Wife getWife() {
		return wife;
	}

	public void setWife(Wife wife) {
		this.wife = wife;
	}
}
```

```java
package vovanhai.wordpresss.com.asso.one.bidirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;

@Entity
public class Wife {
    @Id
    private String id;
    private String name;

    @OneToOne
    private Husband husband;
```

```
    public Wife() {}

       public Wife(String id, String name) {
              this.id = id;
              this.name = name;
       }

       public String getId() {
              return id;
       }

       public void setId(String id) {
              this.id = id;
       }

       public String getName() {
              return name;
       }

       public void setName(String name) {
              this.name = name;
       }

       public Husband getHusband() {
              return husband;
       }

       public void setHusband(Husband husband) {
              this.husband = husband;
       }

}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
       trans.begin();

       Husband h=new Husband("H_1001", "Trần Văn Tũn");
       Wife w=new Wife("V_1002", "Thgan6 Thị Đẹt");
       h.setWife(w);
       w.setHusband(h);

       session.persist(h);
       session.persist(w);

       trans.commit();
} catch (Exception e) {
       trans.rollback();
       e.printStackTrace();
}
```

Kết quả

```
> db.Husband.find().pretty()
{ "_id" : "H_1001", "wife_id" : "V_1002", "name" : "Trần Văn Tũn" }
> db.Wife.find().pretty()
{ "_id" : "V_1002", "husband_id" : "H_1001", "name" : "Thgan6 Thị Đẹt" }
>
```

3. Many-To-One Unidirectional

```java
package vovanhai.wordpresss.com.asso.many2one.undirect;

import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class JavaUserGroup {

	@Id
	private String jugId;
	private String name;

	public JavaUserGroup() {
	}

	public JavaUserGroup(String jugId, String name) {
		super();
		this.jugId = jugId;
		this.name = name;
	}

	public String getJugId() {
		return jugId;
	}

	public void setJugId(String jugId) {
		this.jugId = jugId;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}
}
```

```java
package vovanhai.wordpresss.com.asso.many2one.undirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;

@Entity
public class Member {

    @Id
    private String id;
    private String name;

    @ManyToOne
    private JavaUserGroup memberOf;

    public Member() {
	}

	public Member(String id, String name) {
		this.id = id;
		this.name = name;
	}

	public String getId() {
```

```
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public JavaUserGroup getMemberOf() {
		return memberOf;
	}

	public void setMemberOf(JavaUserGroup memberOf) {
		this.memberOf = memberOf;
	}

}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();

	JavaUserGroup jsg=new JavaUserGroup("summer_camp", "JUG Summer Camp");

	Member m1=new Member("jerome","Jerome");
	Member m2=new Member("emmanuel","Emmanuel Bernard");
	//Member m3=new Member("teonguyen","Nguyễn văn Tèo");

	m1.setMemberOf(jsg);
	m2.setMemberOf(jsg);

	session.persist(jsg);
	session.persist(m1);
	session.persist(m2);

	trans.commit();
} catch (Exception e) {
	trans.rollback();
	e.printStackTrace();
}
```

Kết quả

```
> db.JavaUserGroup.find()
{ "_id" : "summer_camp", "name" : "JUG Summer Camp" }
> db.Member.find()
{ "_id" : "jerome", "name" : "Jerome", "memberOf_jugId" : "summer_camp" }
{ "_id" : "emmanuel", "name" : "Emmanuel Bernard", "memberOf_jugId" : "summer_camp" }
>
```

4. Many-To-One Bidirectional

```
package vovanhai.wordpresss.com.asso.many2one.bidirect;

import java.util.HashSet;
import java.util.Set;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity
@Table(name = "salesforce")
public class SalesForce {

	@Id
	private String id;
	private String corporation;

	@OneToMany(mappedBy = "salesForce")
	private Set<SalesGuy> salesGuys = new HashSet<SalesGuy>();

	public SalesForce() {
	}

	public SalesForce(String id, String corporation) {
		this.id = id;
		this.corporation = corporation;
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getCorporation() {
		return corporation;
	}

	public void setCorporation(String corporation) {
		this.corporation = corporation;
	}

	public Set<SalesGuy> getSalesGuys() {
		return salesGuys;
	}

	public void setSalesGuys(Set<SalesGuy> salesGuys) {
		this.salesGuys = salesGuys;
	}

}
```

```
package vovanhai.wordpresss.com.asso.many2one.bidirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name="salesguy")
```

```
public class SalesGuy {
	@Id
    private String id;
    private String name;

    @ManyToOne
    private SalesForce salesForce;

    public SalesGuy() {
	}
	public SalesGuy(String id, String name) {
		this.id = id;
		this.name = name;
	}
	public String getId() {
		return id;
	}
	public void setId(String id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public SalesForce getSalesForce() {
		return salesForce;
	}
	public void setSalesForce(SalesForce salesForce) {
		this.salesForce = salesForce;
	}
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();
	SalesForce sf=new SalesForce("red_hat","Red Hat");

	SalesGuy sg1=new SalesGuy("eric", "Eric Cantona");
	SalesGuy sg2=new SalesGuy("simon", "Simon Joey");

	sg1.setSalesForce(sf);
	sg2.setSalesForce(sf);

	Set<SalesGuy> salesGuys=new HashSet<>();
	salesGuys.add(sg1);
	salesGuys.add(sg2);

	sf.setSalesGuys(salesGuys);

	session.persist(sf);
	session.persist(sg1);
	session.persist(sg2);

	trans.commit();
} catch (Exception e) {
	trans.rollback();
	e.printStackTrace();
}
```

Kết quả

```
> db.salesforce.find().pretty()
{
        "_id" : "red_hat",
        "corporation" : "Red Hat",
        "salesGuys" : [
                "eric",
                "simon"
        ]
}
> db.salesguy.find().pretty()
{ "_id" : "eric", "salesForce_id" : "red_hat", "name" : "Eric Cantona" }
{ "_id" : "simon", "salesForce_id" : "red_hat", "name" : "Simon Joey" }
>
```

5. One-To-Many Unidirectional

```
package vovanhai.wordpresss.com.asso.one2many.undirect;

import java.util.ArrayList;
import java.util.List;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.Table;

@Entity @Table(name="basket")
public class Basket {

    @Id
    private String id;
    private String owner;

    @OneToMany
    private List<Product> products = new ArrayList<Product>();

    public Basket() {
       }

       public Basket(String id, String owner) {
              this.id = id;
              this.owner = owner;
       }

       public String getId() {
              return id;
       }

       public void setId(String id) {
              this.id = id;
       }

       public String getOwner() {
              return owner;
       }

       public void setOwner(String owner) {
              this.owner = owner;
       }

       public List<Product> getProducts() {
              return products;
       }

       public void setProducts(List<Product> products) {
              this.products = products;
       }
}
```

```
package vovanhai.wordpresss.com.asso.one2many.undirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity @Table(name="product")
public class Product {
    @Id
    private String name;
    private String description;
```

```
    public Product() {
       }
       public Product(String name, String description) {
              this.name = name;
              this.description = description;
       }
       public String getName() {
              return name;
       }
       public void setName(String name) {
              this.name = name;
       }
       public String getDescription() {
              return description;
       }
       public void setDescription(String description) {
              this.description = description;
       }
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
       trans.begin();

       Product p1=new Product("Pretzel", "Glutino Pretzel Sticks");
       Product p2=new Product("Beer", "Tactical nuclear penguin");

       Basket basket=new Basket("davide_basket", "Davide");

       List<Product> products=new ArrayList<>();
       products.add(p1);
       products.add(p2);
       basket.setProducts(products);

       session.persist(p1);
       session.persist(p2);

       session.persist(basket);

       trans.commit();
} catch (Exception e) {
       trans.rollback();
       e.printStackTrace();
```

Kết quả

```
> db.basket.find().pretty()
{
        "_id" : "davide_basket",
        "owner" : "Davide",
        "products" : [
                "Pretzel",
                "Beer"
        ]
}
> db.product.find().pretty()
{ "_id" : "Beer", "description" : "Tactical nuclear penguin" }
{ "_id" : "Pretzel", "description" : "Glutino Pretzel Sticks" }
>
```

Trường hợp có OrderColumn

```
package vovanhai.wordpresss.com.asso.one2many.undirect;

import java.util.ArrayList;
import java.util.List;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.OrderColumn;
import javax.persistence.Table;

@Entity @Table(name="basket")
public class Basket {

    @Id
    private String id;
    private String owner;

    @OneToMany
    @OrderColumn
    private List<Product> products = new ArrayList<Product>();

}
```

Kết quả

```
> db.product.find().pretty()
{ "_id" : "Beer", "description" : "Tactical nuclear penguin" }
{ "_id" : "Pretzel", "description" : "Glutino Pretzel Sticks" }
> db.basket.find().pretty()
{
        "_id" : "davide_basket",
        "owner" : "Davide",
        "products" : [
                {
                        "products_ORDER" : 0,
                        "products_name" : "Pretzel"
                },
                {
                        "products_ORDER" : 1,
                        "products_name" : "Beer"
                }
        ]
}
>
```

6. Many-To-Many **Unidirectional**

```
package vovanhai.wordpresss.com.asso.many2many.undirect;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity @Table(name="student")
public class Student {
	@Id
	private String id;
	private String name;

	public Student() {
	}

	public Student(String id, String name) {
		this.id = id;
		this.name = name;
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}
}
```

```
package vovanhai.wordpresss.com.asso.many2many.undirect;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity @Table(name="classroom")
public class ClassRoom {
	@Id
	private long id;
	private String lesson;

	@ManyToMany
	private List<Student> students = new ArrayList<Student>();

	public ClassRoom() {
	}

	public ClassRoom(long id, String lesson) {
		this.id = id;
```

```
		this.lesson = lesson;
	}

	public long getId() {
		return id;
	}

	public void setId(long id) {
		this.id = id;
	}

	public String getLesson() {
		return lesson;
	}

	public void setLesson(String lesson) {
		this.lesson = lesson;
	}

	public List<Student> getStudents() {
		return students;
	}

	public void setStudents(List<Student> students) {
		this.students = students;
	}
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();

	ClassRoom cr1=new ClassRoom(1001l, "Math");
	ClassRoom cr2=new ClassRoom(1002l, "English");

	Student s1=new Student("john","John Doe");
	Student s2=new Student("mario","Mario Rossi");
	Student s3=new Student("kate","Kate Winles");


	List<Student> students1=new ArrayList<>();
	students1.add(s1);
	students1.add(s2);

	List<Student> students2=new ArrayList<>();
	students2.add(s1);
	students2.add(s3);

	cr1.setStudents(students1);
	cr2.setStudents(students2);

	session.persist(s1);
	session.persist(s2);
	session.persist(s3);

	session.persist(cr1);
	session.persist(cr2);

	trans.commit();
} catch (Exception e) {
```

```
        trans.rollback();
        e.printStackTrace();
}
```

Kết quả

```
> db.student.find().pretty()
{ "_id" : "mario", "name" : "Mario Rossi" }
{ "_id" : "kate", "name" : "Kate Winles" }
{ "_id" : "john", "name" : "John Doe" }
> db.classroom.find().pretty()
{
        "_id" : NumberLong(1001),
        "lesson" : "Math",
        "students" : [
                "john",
                "mario"
        ]
}
{
        "_id" : NumberLong(1002),
        "lesson" : "English",
        "students" : [
                "john",
                "kate"
        ]
}
>
```

7. Many-To-Many Bidirectional

```java
package vovanhai.wordpresss.com.asso.many2many.bidirect;

import java.util.HashSet;
import java.util.Set;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity @Table(name="bankaccount")
public class BankAccount {
    @Id
    private String id;
    private String accountNumber;

    @ManyToMany( mappedBy = "bankAccounts" )
    private Set<AccountOwner> owners = new HashSet<AccountOwner>();

    public BankAccount() {}

       public BankAccount(String id, String accountNumber) {
              this.id = id;
              this.accountNumber = accountNumber;
       }

       public String getId() {
              return id;
       }

       public void setId(String id) {
              this.id = id;
       }

       public String getAccountNumber() {
              return accountNumber;
       }

       public void setAccountNumber(String accountNumber) {
              this.accountNumber = accountNumber;
       }

       public Set<AccountOwner> getOwners() {
              return owners;
       }

       public void setOwners(Set<AccountOwner> owners) {
              this.owners = owners;
       }
}
```

```java
package vovanhai.wordpresss.com.asso.many2many.bidirect;

import java.util.Set;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.Table;

@Entity @Table(name="accountowner")
public class AccountOwner {
       @Id
```

```
	private String id;
	private String SSN;

	@ManyToMany
	private Set<BankAccount> bankAccounts;

	public AccountOwner() {
	}

	public AccountOwner(String id, String sSN) {
		this.id = id;
		SSN = sSN;
	}

	public String getId() {
		return id;
	}

	public void setId(String id) {
		this.id = id;
	}

	public String getSSN() {
		return SSN;
	}

	public void setSSN(String sSN) {
		SSN = sSN;
	}

	public Set<BankAccount> getBankAccounts() {
		return bankAccounts;
	}

	public void setBankAccounts(Set<BankAccount> bankAccounts) {
		this.bankAccounts = bankAccounts;
	}
}
```

Test

```
OgmSession session = ogmSessionFactory.openSession();
Transaction trans = session.getTransaction();
try {
	trans.begin();
	AccountOwner acc1=new AccountOwner("owner0001", "0123456");
	AccountOwner acc2=new AccountOwner("owner0002", "6543210");

	BankAccount ba1=new BankAccount("account_1", "X2345000");
	BankAccount ba2=new BankAccount("account_2", "Y8647382");

	Set<BankAccount> bankAccounts=new HashSet<>();
	bankAccounts.add(ba1);
	bankAccounts.add(ba2);


	Set<AccountOwner> owners=new HashSet<>();
	owners.add(acc1);
	owners.add(acc2);

	acc1.setBankAccounts(bankAccounts);
	ba1.setOwners(owners);
```

```
        session.persist(acc1);
        session.persist(acc2);
        session.persist(ba1);
        session.persist(ba2);

        trans.commit();
} catch (Exception e) {
        trans.rollback();
        e.printStackTrace();
}
```

Kết quả

```
> db.bankaccount.find().pretty()
{
        "_id" : "account_1",
        "accountNumber" : "X2345000",
        "owners" : [
                "owner0001"
        ]
}
{
        "_id" : "account_2",
        "accountNumber" : "Y8647382",
        "owners" : [
                "owner0001"
        ]
}
```

BankAccount collection

```
> db.accountowner.find().pretty()
{ "_id" : "owner0002", "SSN" : "6543210" }
{
        "_id" : "owner0001",
        "SSN" : "0123456",
        "bankAccounts" : [
                "account_2",
                "account_1"
        ]
}
>
```

AccountOwner collection

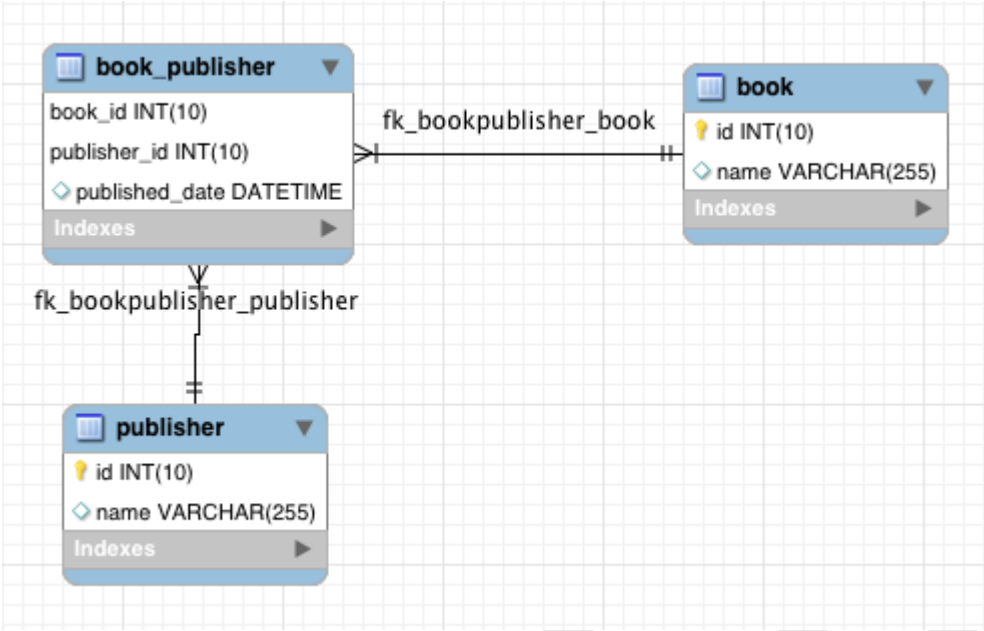
JP-QL queries

1. Native MongoDB queries

PHẦN 2: BÀI TẬP

BÀI 1

Cho mô hình thực thể quan hệ sau



Mô tả các thực thể này và viết các lớp xử lý data access (các thao tác crud). Thêm một số câu truy vấn bằng JPQL (named query) thực thi các yêu cầu sau:

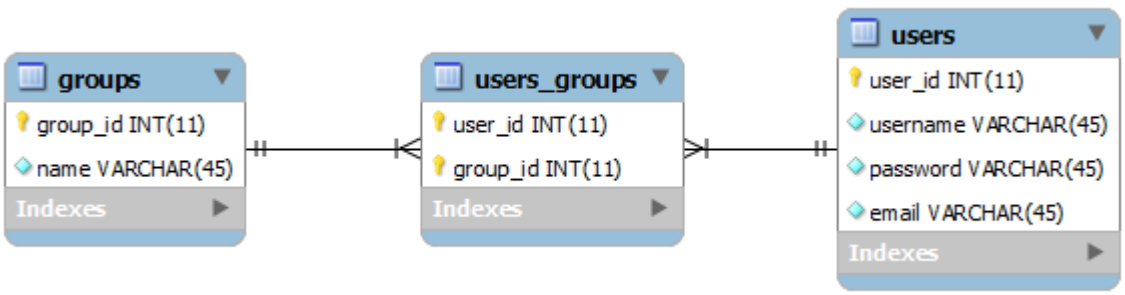
- Lấy danh sách các sách của cùng một nhà xuất bản phát hành.
- Tìm các nhà xuất bản cùng xuất bản một cuốn sách
- Tìm những cuốn sách được phát hành trong một ngày nào đó.
- Tìm sách theo tên sách

BÀI 2

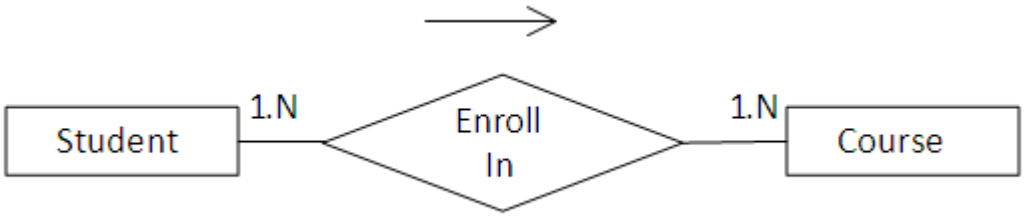
Source: http://www.codejava.net/frameworks/hibernate/hibernate-many-to-many-association-annotations-example

https://howtoprogramwithjava.com/hibernate-manytomany-unidirectional-bidirectional

Cho cơ sở dữ liệu mô tả như sau

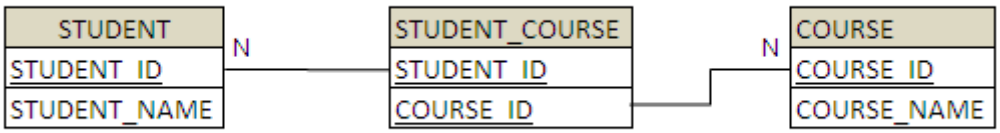


BÀI 3



According to the relationship a student can enroll in any number of courses and the course can have any number of students.

To create this relationship you need to have a *STUDENT*, *COURSE* and *STUDENT_COURSE* table. The relational model is shown below



Vài options

```
Show câu sql

<property name="eclipselink.logging.level" value="FINE"/>
<property name = "hibernate.show_sql" value = "true" />
```