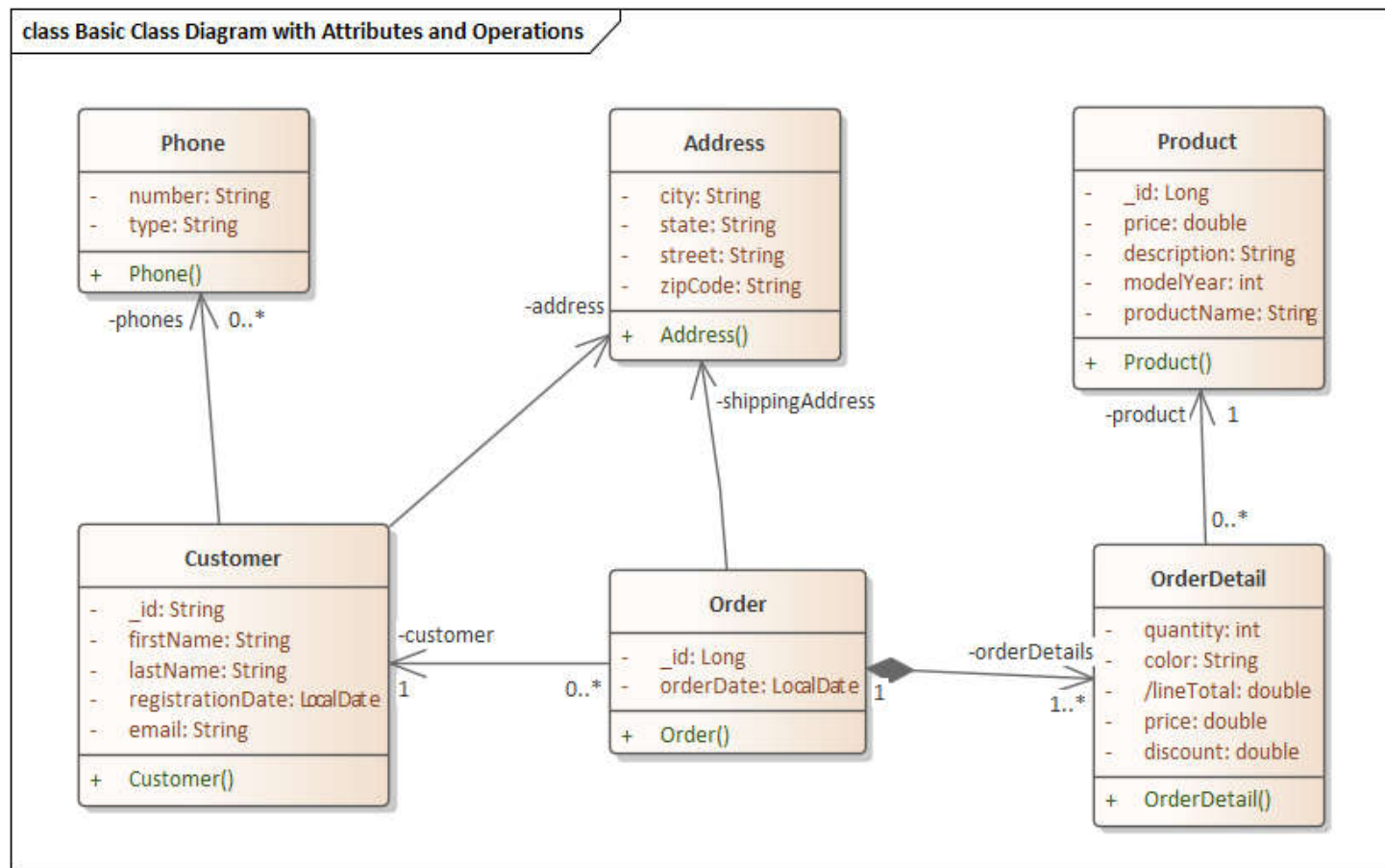


BÀI TẬP

Một khách hàng có thể có nhiều hóa đơn, một hóa đơn chỉ thuộc một khách hàng. Một hóa đơn có thể mua nhiều sản phẩm và một sản phẩm có thể bán trên nhiều hóa đơn.

Sơ đồ lớp như sau:



Cấu trúc dữ liệu json của các đối tượng như sau:

```
Customer.json
{
  "_id": "AGSI14215",
  "firstName": "Agnes",
  "lastName": "Sims",
  "address": {
    "city": "Buffalo",
    "state": "NY",
    "street": "170 Queen Lane ",
    "zipCode": "14215"
  },
  "registrationDate": "1999-11-02",
  "email": "agnes.sims@aol.com",
  "phones": [
    {
      "number": "799 724-303",
      "type": "home"
    }
  ]
}
```

```

    },
    {
      "number" : "33 556-775",
      "type" : "personal"
    }
  ]
}

```

Order.json

```

{
  "_id" : NumberLong(2),
  "shippingAddress" : {
    "city" : "Oswego",
    "zipCode" : "13126",
    "street" : "7800 Magnolia Street ",
    "state" : "NY"
  },
  "customerID" : "ELDE13126",
  "orderDate" : "2016-04-15",
  "orderDetails" : [
    {
      "quantity" : 2,
      "color" : "blue",
      "productID" : NumberLong(16),
      "lineTotal" : 1139.981,
      "price" : 599.99,
      "discount" : 0.05000000074505806
    },
    {
      "quantity" : 1,
      "color" : "blue",
      "productID" : NumberLong(20),
      "lineTotal" : 557.9907,
      "price" : 599.99,
      "discount" : 0.07000000029802322
    }
  ]
}

```

Product.json

```

{
  "_id" : NumberLong(4),
  "price" : 469.99,
  "description" : "Brand: Surly, Category: Mountain Bikes",
  "modelYear" : 2016,
  "productName" : "Surly Ice Cream Truck Frameset - 2016"
}

```

1/ Ánh xạ mô hình lớp trên sang mô hình cơ sở dữ liệu No-SQL (MongoDB)

2/ Thực hiện các thao tác Create, Update, Delete, Find by Id trên từng collection.

3/ Import các file json cho sẵn vào các collection tương ứng có trong database. Viết các phương thức với các yêu cầu sau:

- 1/ Tìm danh sách sản phẩm có giá cao nhất.
- 2/ Tìm danh sách sản phẩm chưa bán được lần nào.
- 3/ Thống kê số khách hàng theo từng bang.
+getNumberCustomerByState() : Map<String, Integer>
- 4/ Tính tổng tiền của đơn hàng khi biết mã số đơn hàng.
- 5/ Đếm số đơn hàng của từng khách hàng.
+ getOrdersByCustomers() : Map<Customer, Integer>
- 6/ Tính tổng số lượng của từng sản phẩm đã bán ra.
+ getTotalProduct(): Map<Product, Integer>
- 7/ Dùng text search để tìm kiếm sản phẩm theo tên sản phẩm và mô tả của sản phẩm.
- 8/ Tính tổng tiền của tất cả các hóa đơn trong một ngày nào đó.
- 9/ Liệt kê danh sách các hóa đơn có bán sản phẩm nào đó khi biết tên sản phẩm.
- 10/ Danh sách các khách hàng có từ 2 số điện thoại liên lạc trở lên.
- 11/ Dùng Text Search để tìm khách hàng theo họ và tên.
- 12/ Tạo index và tìm kiếm sản phẩm theo giá sản phẩm.
- 13/ Tìm khách hàng khi biết số điện thoại của khách hàng.
- 14/ Lập danh sách khách hàng đăng ký thành viên trong một ngày nào đó.
- 15/ Đếm số khách hàng đăng ký thành viên trong một năm nào đó.

Dependencies

```
<dependencies>
  <dependency>
    <groupId>org.hibernate.ogm</groupId>
    <artifactId>hibernate-ogm-mongodb</artifactId>
    <version>5.4.1.Final</version>
  </dependency>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.5</version>
  </dependency>
</dependencies>
```

Configuration

Hibernate Native



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>

    <!-- Database connection settings -->
    <property name="hibernate.ogm.datastore.provider">mongodb</property>
    <!-- Enable Hibernate's automatic session context management -->
    <property name="current_session_context_class">thread</property>
    <property name="hibernate.ogm.datastore.host">localhost</property>
    <property name="hibernate.ogm.mongodb.port">27017</property>
    <property name="hibernate.ogm.datastore.database">mydb</property>

  </session-factory>
</hibernate-configuration>
```

JPA



persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2" 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_2.xsd">
  <persistence-unit name="JPA_UNIT_NAME">
    <provider>org.hibernate.ogm.jpa.HibernateOgmPersistence</provider>
    <properties>
      <class>entity.Phone</class>
      <class>entity.Customer</class>
```

```

        <class>entity.Address</class>
        <class>entity.Order</class>
        <class>entity.OrderDetail</class>
        <class>entity.Product</class>

        <property name="hibernate.ogm.datastore.provider" value="mongodb"/>
        <property name="hibernate.ogm.datastore.database" value="mydb"/>
        <property name="hibernate.ogm.datastore.host" value="localhost"/>
        <property name="hibernate.ogm.datastore.port" value="27017"/>
        <property name="hibernate.ogm.datastore.create_database" value="true"/>

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

```

Bootstrapping Hibernate OGM

Hibernate Native

```

StandardServiceRegistry registry = new StandardServiceRegistryBuilder()
    .applySetting(OgmProperties.ENABLED, true)
    .configure()

//
//    .applySetting(AvailableSettings.CURRENT_SESSION_CONTEXT_CLASS, "thread")
//    .applySetting(OgmProperties.DATASTORE_PROVIDER,
MongoDB.DATASTORE_PROVIDER_NAME)
//    .applySetting(OgmProperties.DATABASE, "mydb")
//    .applySetting(OgmProperties.HOST, "localhost")
//    .applySetting(OgmProperties.CREATE_DATABASE, "true")
//    .build();

Metadata meta = new MetadataSources(registry)
    .addAnnotatedClass(entity.Address.class)
    .addAnnotatedClass(entity.Customer.class)
    .addAnnotatedClass(entity.Order.class)
    .addAnnotatedClass(entity.OrderDetail.class)
    .addAnnotatedClass(entity.Product.class)
    .getMetadataBuilder()
    .build();

OgmSessionFactory ogmSessionFactory = meta.getSessionFactoryBuilder()
    .unwrap(OgmSessionFactoryBuilder.class)
    .build();

OgmSession session = ogmSessionFactory.getCurrentSession();

```

JPA

```

EntityManager em = Persistence.createEntityManagerFactory("JPA_UNIT_NAME ")
    .createEntityManager();

```