

**Tạo ứng dụng kết nối tới DB TestingSystem bằng Hibernate sử dụng bảng Department trong DB.**

## 1. Giao diện chính chương trình

**Giao diện chính:**

```
Console
DemoCRUDHibernate [Java Application]
Picked up _JAVA_OPTIONS: -Djava.net.preferIPv4Stack=true
Chương trình quản lý phòng ban..
Mời bạn nhập vào chức năng muốn sử dụng
1. Lấy danh sách tất cả các phòng ban trên hệ thống
2. Tìm phòng ban theo ID
3. Tìm phòng ban theo Name
4. Tạo mới 1 phòng
```

**Lấy danh sách tất cả phòng ban:**

```
Hibernate:
select
  department0_.DepartmentID as departme1_0_,
  department0_.DepartmentName as departme2_0_
from
  testingsystem1.department department0_
ID: 15 Name:
ID: 16 Name: 123
ID: 6 Name: Accounting
ID: 3 Name: Business Development
ID: 1 Name: Engineering
ID: 10 Name: Human Resources
ID: 4 Name: Legal
ID: 12 Name: Marketing
ID: 2 Name: Product Management
ID: 11 Name: Research and Development
ID: 8 Name: Sale
ID: 7 Name: Services
ID: 5 Name: Support
ID: 13 Name: test
ID: 9 Name: Training
```

**Lấy phòng ban theo ID:**

```
Hibernate:
select
  department0_.DepartmentID as departme1_0_0_,
  department0_.DepartmentName as departme2_0_0_
from
  testingsystem1.department department0_
where
  department0_.DepartmentID=?
Department [id=1, name=Engineering]
```

## 2. Tạo DB.

```
-- create database
DROP DATABASE IF EXISTS TestingSystem;
CREATE DATABASE TestingSystem;
USE TestingSystem;
```

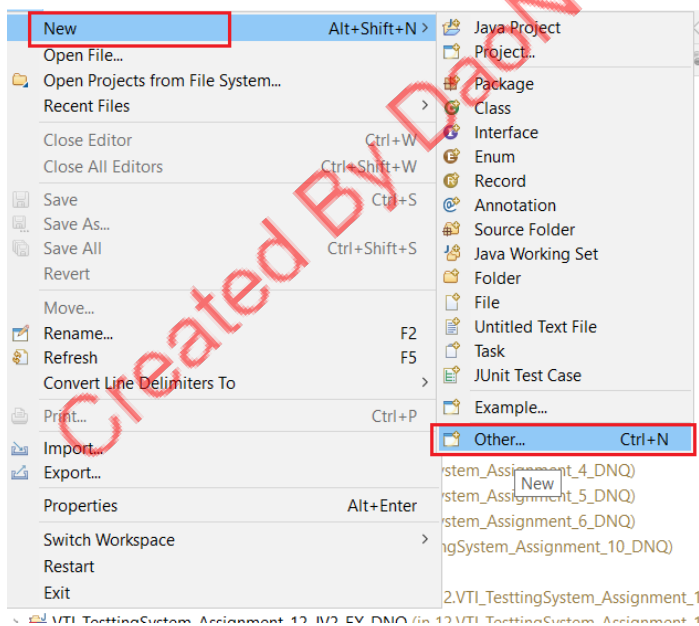
```

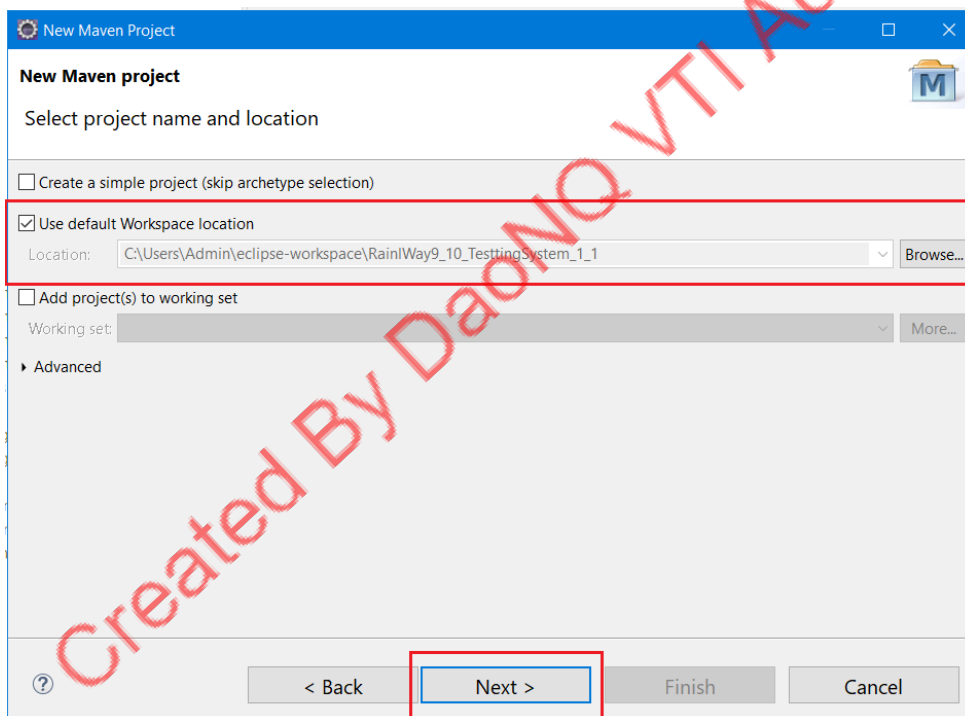
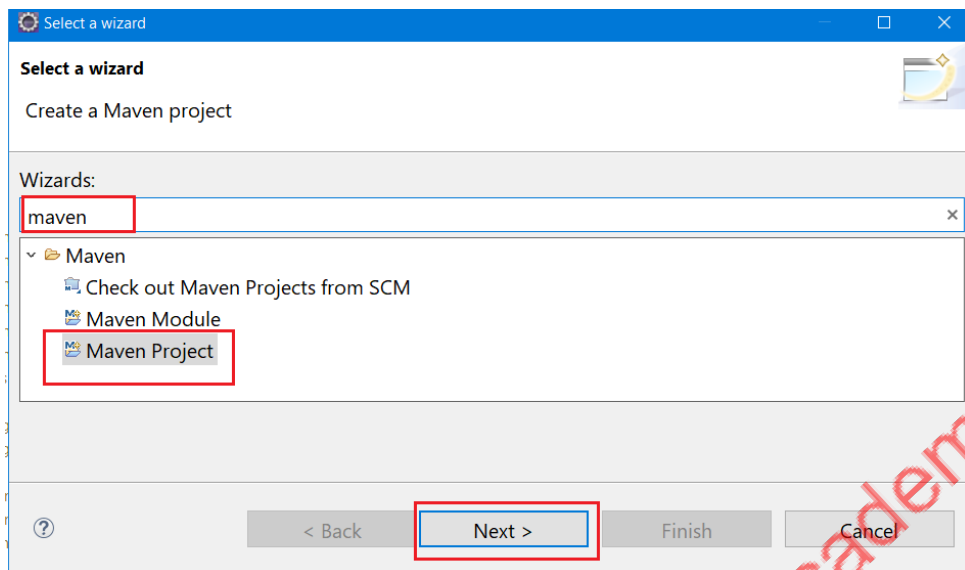
-- create table 1: Department
DROP TABLE IF EXISTS Department;
CREATE TABLE Department(
    DepartmentID          TINYINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    DepartmentName        NVARCHAR(30) NOT NULL UNIQUE KEY
);

-- Add data Department
INSERT INTO Department(DepartmentName)
VALUES
    (N'Marketing'      ),
    (N'Sale'            ),
    (N'Bảo vệ'         ),
    (N'Nhân sự'         ),
    (N'Kỹ thuật'        ),
    (N'Tài chính'       ),
    (N'Phó giám đốc'),
    (N'Giám đốc'        ),
    (N'Thư kí'          ),
    (N'Bán hàng'       );

```

### 3. Tạo Project maven: hình minh họa.





New Maven Project

New Maven project

Select an Archetype

Catalog: Internal

Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-j2ee-simple	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1

An archetype which contains a sample Maven project.

☒ Show the last version of Archetype only ☐ Include snapshot archetypes

Advanced

< Back Next > Finish Cancel

New Maven Project

New Maven project

Specify Archetype parameters

Group Id: com.vti

Artifact Id: Rainlway9\_10\_TestingSystem\_1\_1

Version: 0.0.1-SNAPSHOT

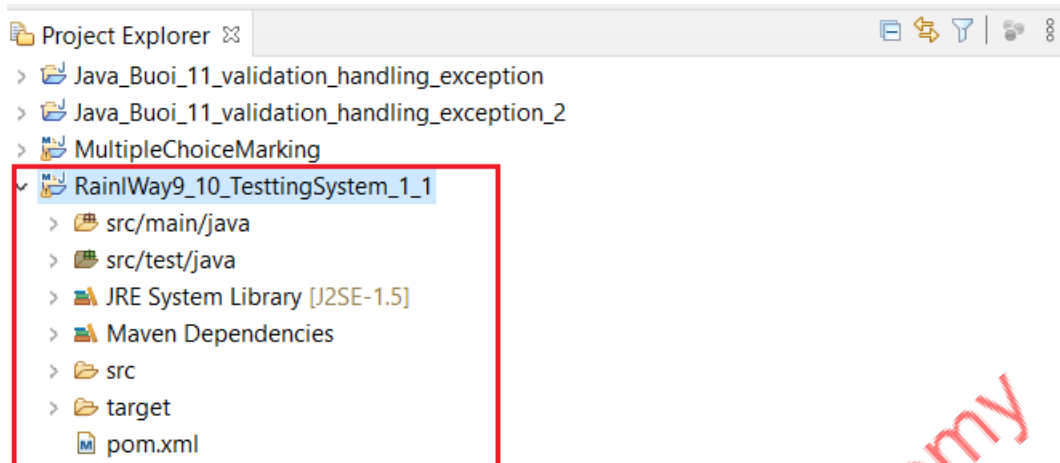
Package: com.vti.entity

Properties available from archetype:

Name	Value

Advanced

< Back Next > Finish Cancel

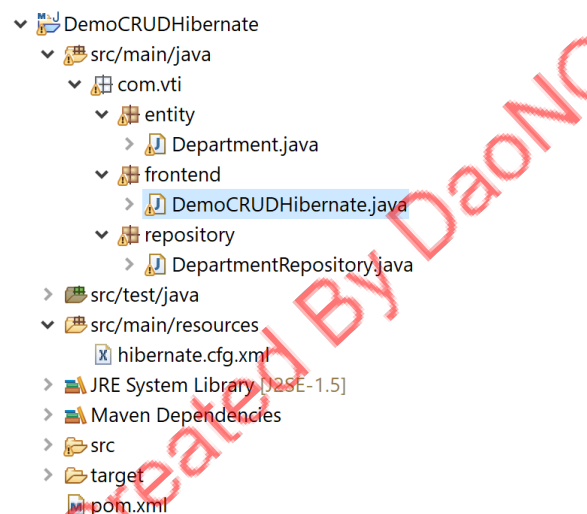


#### 4. Tạo khung chương trình:

Tạo mới 1 project: DemoCRUDHibernate, Tạo mới các package và class như dưới:

Package: com.vti.entity, com.vti.frontend, com.vti.repository

Class: DepartmentRepository, Department, DemoCRUDHibernate



#### 5. Thêm các cấu hình cho file: pom.xml

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.17.Final</version>
</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.20</version>
</dependency>
```

```

9      <packaging>jar</packaging>
10
11      <name>DemoCRUD</name>
12      <url>http://maven.apache.org</url>
13
14      <properties>
15          <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
16      </properties>
17
18      <dependencies>
19          <dependency>
20              <groupId>junit</groupId>
21              <artifactId>junit</artifactId>
22              <version>3.8.1</version>
23              <scope>test</scope>
24          </dependency>
25
26          <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
27          <dependency>
28              <groupId>org.hibernate</groupId>
29              <artifactId>hibernate-core</artifactId>
30              <version>5.4.17.Final</version>
31          </dependency>
32
33          <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
34          <dependency>
35              <groupId>mysql</groupId>
36              <artifactId>mysql-connector-java</artifactId>
37              <version>8.0.20</version>
38          </dependency>
39      </dependencies>
40  </project>
41
42

```



pom.xml

## 6. Thêm các cấu hình cho file: hibernate.cfg.xml trong resources

```

<?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="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/testingsystem1</property>
        <property name="connection.username">root</property>
        <property name="connection.password">root</property>

        <!-- format code SQL -->
        <property name="show_sql">true</property>
        <property name="hibernate.format_sql">true</property>
        <property name="connection.pool_size">10</property>

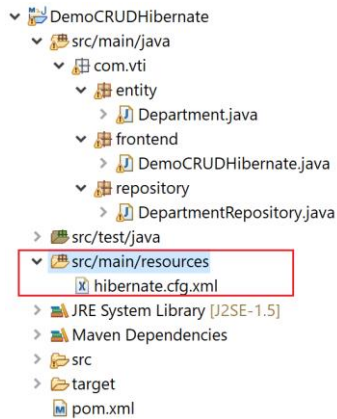
        <!-- other -->
        <property name="hibernate.connection.characterEncoding">utf8</property>

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

```



hibernate.cfg.xml



## 7. Tạo Class Department trong package com.vti.entity:

```
package com.vti.entity;

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.Table;

@Entity
@Table(name = "department", catalog = "testingsystem1")
public class Department implements Serializable {
    @Column(name = "DepartmentID")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private short id;
    @Column(name = "DepartmentName", nullable = false, unique = true)
    private String name;

    /**
     * @return the id
     */
    public short getId() {
        return id;
    }

    /**
     * @param id the id to set
     */
    public void setId(byte id) {
        this.id = id;
    }

    /**
     * @return the name
     */
    public String getName() {
        return name;
    }

    /**
     * @param name the name to set
     */
    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Department [id=" + id + ", name=" + name + "];"
    }
}
```

```

    }
}

```



Department.java

## 8. Tạo Class DepartmentRepository trong package com.vti.repository:

```

package com.vti.repository;

import java.util.List;

import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

import com.vti.entity.Department;

public class DepartmentRepository {
    private org.hibernate.SessionFactory sessionFactory;

    public DepartmentRepository() {
        sessionFactory = buildSessionFactory();
    }

    private org.hibernate.SessionFactory buildSessionFactory() {
        Configuration configuration = new Configuration();
        configuration.configure("hibernate.cfg.xml");
        configuration.addAnnotatedClass(Department.class);
        ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()
            .applySettings(configuration.getProperties()).build();

        return configuration.buildSessionFactory(serviceRegistry);
    }

    @SuppressWarnings("deprecation")
    public List<Department> getAllDepartments() {

        Session session = null;

        try {
            session = sessionFactory.openSession();
            Query<Department> query = session.createQuery("FROM Department");

            return query.list();
        } finally {
            if (session != null) {
                session.close();
            }
        }
    }

    public Department getDepartmentByID(short id) {
        Session session = null;
        try {
            session = sessionFactory.openSession();
            Department department = session.get(Department.class, id);
            return department;
        } finally {
            if (session != null) {
                session.close();
            }
        }
    }

    public Department getDepartmentByName(String name) {
        Session session = null;
        try {

```



```

        // get session
        session = sessionFactory.openSession();

        // create hql query
        Query<Department> query = session.createQuery("FROM Department WHERE name =
:nameParameter");

        // set parameter
        query.setParameter("nameParameter", name);

        // get result
        Department department = query.uniqueResult();
        return department;
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

public void createDepartment(Department department) {
    Session session = null;

    try {
        // get session
        session = sessionFactory.openSession();
        session.beginTransaction();

        // create
        session.save(department);

        session.getTransaction().commit();
    } finally {
        if (session != null) {
            session.close();
        }
    }
}

public boolean isDepartmentExistsByID(short id) {
    // get department
    Department department = getDepartmentByID(id);

    // return result
    if (department == null) {
        return false;
    }

    return true;
}

public boolean isDepartmentExistsByName(String name) {
    Department department = getDepartmentByName(name);

    if (department == null) {
        return false;
    }

    return true;
}
}

```



DepartmentRepository.java

## 9. Tạo Class DemoCRUDHibernate trong com.vti.frontend:

```
package com.vti.frontend;

import java.util.List;
import java.util.Scanner;

import com.vti.entity.Department;
import com.vti.repository.DepartmentRepository;

public class DemoCRUDHibernate {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int choose = menu();
        switch (choose) {
            case 1:
                showAllDepartmet();
                break;
            case 2:
                System.out.println("Nhập vào ID của phòng ban");
                short id = scanner.nextShort();
                showDepartmentByID(id);
                break;
            case 3:
                System.out.println("Nhập vào tên phòng ban cần tìm kiếm");
                String nameDep = scanner.nextLine();
                showDepartmentByName(nameDep);
                break;
            case 4:
                System.out.println("Tạo mới phòng ban");
                System.out.println("Nhập tên phòng cần tạo:");
                String nameDepCreate = scanner.nextLine();
                Department dep = new Department();
                dep.setName(nameDepCreate);
                createDep(dep);
                showAllDepartmet();
                break;
            default:
                break;
        }
    }

    private static void createDep(Department dep) {
        DepartmentRepository depRepository = new DepartmentRepository();
        depRepository.createDepartment(dep);
    }

    private static void showDepartmentByName(String nameDep) {
        DepartmentRepository depRepository = new DepartmentRepository();
        Department depByName = depRepository.getDepartmentByName(nameDep);
        if (depByName == null) {
            System.out.println("Không có phòng ban này trên hệ thống");
        } else {
            System.out.println(depByName.toString());
        }
    }

    private static void showDepartmentByID(Short id) {
        DepartmentRepository depRepository = new DepartmentRepository();
        Department dep = depRepository.getDepartmentByID(id);
        System.out.println(dep.toString());
    }

    private static void showAllDepartmet() {
        System.out.println("Danh sách phòng ban trên hệ thống");
        DepartmentRepository depRepository = new DepartmentRepository();
        List<Department> listDep = depRepository.getAllDepartments();
        for (Department department : listDep) {
```

```

        System.out.println("ID: " + department.getId() + " Name: " +
        department.getName());
    }

    @SuppressWarnings("resource")
    private static int menu() {
        while (true) {
            System.out.println("Chương trình quản lý phòng ban..");
            System.out.println("Mời bạn nhập vào chức năng muốn sử dụng");
            System.out.println("1. Lấy danh sách tất cả các phòng ban trên hệ thống");
            System.out.println("2. Tìm phòng ban theo ID");
            System.out.println("3. Tìm phòng ban theo Name");
            System.out.println("4. Tạo mới 1 phòng");
            Scanner sc = new Scanner(System.in);
            if (sc.hasNextInt()) {
                int i = sc.nextInt();
                if ((i == 1) || (i == 2) || (i == 3) || (i == 4)) {
                    return i;
                } else {
                    System.out.println("Nhập lại");
                }
            } else {
                System.out.println("Nhập lại");
            }
        }
    }
}

```

Created By DaoNQ VTI Academy