**Tên project:** 22T1020362\_AbstractFactory

# **a. Nội dung mã nguồn tất cả class, interface, file tài nguyên trong project**

Trong src/

## **Package: \_22T1020362**

### **Program.java: File chạy chương trình**

package \_22T1020362;

import java.io.IOException;

import \_22T1020362.abstractfactory.MongoDBRepoFactory;

import \_22T1020362.abstractfactory.MySQLRepoFactory;

import \_22T1020362.abstractfactory.RepoFactory;

import \_22T1020362.abstractfactory.TextRepoFactory;

import \_22T1020362.config.Config;

import \_22T1020362.models.Nganh;

import \_22T1020362.models.SinhVien;

import \_22T1020362.nganhrepo.INganhRepo;

import \_22T1020362.sinhvienrepo.ISinhVienRepo;

public class Program {

public static void main(String[] args) throws IOException {

//Simple Factory

Config cg = new Config();

String type = cg.getConfigType(); //MySQL, MongoDB, Text

String object = cg.getObjectType(); //Nganh, SinhVien

RepoFactory f;

if(type.equals("MySQL")) f = new MySQLRepoFactory();

else if(type.equals("MongoDB")) f = new MongoDBRepoFactory();

else if(type.equals("Text")) f = new TextRepoFactory();

else f = null;

if(f != null) {

if(object.equals("Nganh")) {

INganhRepo ndtRepo = f.getNganhRepo();

System.out.println("-----------------------------Đọc ngành từ " + type + " -----------------------------");

//read

System.out.println("Danh sách SV:");

System.out.println("[Mã ngành" + " - " + "Tên ngành]");

for(Nganh ndt: ndtRepo.readNDT()) {

System.out.println(ndt.getMaNganh() + " - " + ndt.getTenNganh());

}

System.out.println("-----------------------------Thêm ngành từ " + type + " -----------------------------");

//insert

Nganh n = new Nganh(102, "Công nghệ thông tin");

ndtRepo.insertNDT(n);

}

else if(object.equals("SinhVien")) {

ISinhVienRepo svRepo = f.getSinhVienRepo();

System.out.println("-----------------------------Đọc sinh viên từ " + type + " -----------------------------");

//read

System.out.println("Danh sách SV:");

System.out.println("[Mã sinh viên" + " - " + "Họ tên" + " - " + "Mã NĐT]");

for(SinhVien sv: svRepo.readSV()) {

System.out.println(sv.getMaSinhVien() + " - " + sv.getHoTen() + " - " + sv.getMaNganh());

}

System.out.println("-----------------------------Xóa sinh viên từ " + type + " -----------------------------");

//delete

String msv = "22T1020485";

svRepo.deleteSV(msv);

}

else {

System.out.println("Đối tượng không có trong cơ sở dữ liệu");

}

}

else {

System.out.println("Hệ thống không hỗ trợ nguồn dữ liệu này");

}

}

}

## **Package: \_22T1020362.abstractfactory**

### **MongoDBRepoFactory.java: Factory cho MongoDB, áp dụng mẫu Abstract Factory**

package \_22T1020362.abstractfactory;

import \_22T1020362.config.MongoDBConfig;

import \_22T1020362.nganhrepo.INganhRepo;

import \_22T1020362.nganhrepo.MongoDBNganhRepo;

import \_22T1020362.sinhvienrepo.ISinhVienRepo;

import \_22T1020362.sinhvienrepo.MongoDBSinhVienRepo;

public class MongoDBRepoFactory extends RepoFactory{

@Override

public ISinhVienRepo getSinhVienRepo() {

return new MongoDBSinhVienRepo(MongoDBConfig.getInstance().getDB());

}

@Override

public INganhRepo getNganhRepo() {

return new MongoDBNganhRepo(MongoDBConfig.getInstance().getDB());

}

}

### **MySQLRepoFactory.java: Factory cho MySQL, áp dụng mẫu Apstract Factory**

package \_22T1020362.abstractfactory;

import \_22T1020362.config.MySQLConfig;

import \_22T1020362.nganhrepo.INganhRepo;

import \_22T1020362.nganhrepo.MySQLNganhRepo;

import \_22T1020362.sinhvienrepo.ISinhVienRepo;

import \_22T1020362.sinhvienrepo.MySQLSinhVienRepo;

public class MySQLRepoFactory extends RepoFactory {

@Override

public ISinhVienRepo getSinhVienRepo() {

return new MySQLSinhVienRepo(MySQLConfig.getInstance().getCn());

}

@Override

public INganhRepo getNganhRepo() {

return new MySQLNganhRepo(MySQLConfig.getInstance().getCn());

}

}

### **RepoFactory.java: Áp dụng mẫu Abstract Factory, là nhà máy ảo tổng quát**

package \_22T1020362.abstractfactory;

import \_22T1020362.nganhrepo.INganhRepo;

import \_22T1020362.sinhvienrepo.ISinhVienRepo;

public abstract class RepoFactory {

public abstract ISinhVienRepo getSinhVienRepo();

public abstract INganhRepo getNganhRepo();

}

### **TextRepoFactory: Factory cho File Text, áp dụng mẫu Abstract Factory**

package \_22T1020362.abstractfactory;

import \_22T1020362.config.TextConfig;

import \_22T1020362.nganhrepo.INganhRepo;

import \_22T1020362.nganhrepo.TextNganhRepo;

import \_22T1020362.sinhvienrepo.ISinhVienRepo;

import \_22T1020362.sinhvienrepo.TextSinhVienRepo;

public class TextRepoFactory extends RepoFactory {

@Override

public ISinhVienRepo getSinhVienRepo() {

return new TextSinhVienRepo(TextConfig.getInstance().getTextFileUrl());

}

@Override

public INganhRepo getNganhRepo() {

return new TextNganhRepo(TextConfig.getInstance().getTextFileUrl());

}

}

## **Package: \_22T1020362.config**

### **Config.java: Đọc kiểu dữ liệu và đối tượng từ file “config.txt”**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.io.IOException;

import java.util.Scanner;

public class Config {

public String getConfigType() throws IOException{

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = sc.nextLine();

sc.close();

return line;

}

public String getObjectType() throws IOException{ //Nganh/SinhVien ?

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = sc.nextLine();

if(line.equals(getConfigType())) {

line = sc.nextLine();

}

sc.close();

return line;

}

}

### **MongoDBConfig.java: Áp dụng mẫu Singleton cho kết nối đến MongoDB**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.util.Scanner;

import com.mongodb.client.MongoClient;

import com.mongodb.client.MongoClients;

import com.mongodb.client.MongoDatabase;

public class MongoDBConfig {

private MongoDatabase db;

public MongoDatabase getDB() {

return this.db;

}

// 1. Khai báo private constructor

private MongoDBConfig() {

try {

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = null;

while(sc.hasNextLine()) {

line = sc.nextLine();

}

String [] lst = line.split("\\|");

MongoClient mongoClient = MongoClients.create(lst[0]);

db = mongoClient.getDatabase(lst[1]);

sc.close();

} catch (Exception e) {

e.printStackTrace();

}

}

// 2. Khai báo INSTANCE

private static MongoDBConfig instance = null;

// 3. Khai báo getInstance

public static MongoDBConfig getInstance() {

if (instance == null) {

instance = new MongoDBConfig();

}

return instance;

}

}

### **MySQLConfig.java: Áp dụng mẫu Singleton cho kết nối đến MySQL**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.Scanner;

public class MySQLConfig {

// 0. Khai báo các thuộc tính, hàm như bình thường

//Ket noi CSDL

private Connection cn;

public Connection getCn() {

return this.cn;

}

// 1. Khai báo private constructor

private MySQLConfig() {

try {

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = null;

while(sc.hasNextLine()) {

line = sc.nextLine();

}

String [] lst = line.split("\\|");

cn = DriverManager.getConnection(lst[0], lst[1], lst[2]);

sc.close();

} catch (Exception e) {

e.printStackTrace();

}

}

// 2. Khai báo INSTANCE

private static MySQLConfig instance = null;

// 3. Khai báo getInstance

public static MySQLConfig getInstance() {

if (instance == null) {

instance = new MySQLConfig();

}

return instance;

}

}

### **TextConfig.java: Áp dụng mẫu Singleton cho kết nối đến File Text**

package \_22T1020362.config;

import java.io.FileInputStream;

import java.util.Scanner;

public class TextConfig {

private String txtfileUrl;

public String getTextFileUrl() {

return this.txtfileUrl;

}

private TextConfig() {

try {

FileInputStream fis = new FileInputStream("config.txt");

Scanner sc = new Scanner(fis);

String line = null;

while(sc.hasNextLine()) {

line = sc.nextLine();

}

txtfileUrl = line;

sc.close();

// mất 10s!!!

} catch (Exception e) {

e.printStackTrace();

}

}

private static TextConfig instance = null;

public static TextConfig getInstance() {

if (instance == null) {

instance = new TextConfig();

}

return instance;

}

}

## **Package: \_22T1020362.models**

### **Nganh.java: chứa class Nganh, các thuộc tính, hàm khởi tạo, getter, setter, hàm riêng áp dụng Lombok**

package \_22T1020362.models;

import lombok.\*;

@Data

@AllArgsConstructor

@NoArgsConstructor

public class Nganh {

private int maNganh;

private String tenNganh;

public String toString() {

return maNganh + "," + tenNganh + "\n";

}

}

### **SinhVien.java: chứa class SinhVien, các thuộc tính, hàm khởi tạo, hàm getter, setter, hàm riêng áp dụng Lombok để code thuận tiện**

package \_22T1020362.models;

import lombok.\*;

@Data

@AllArgsConstructor

@NoArgsConstructor

public class SinhVien {

private String maSinhVien;

private String hoTen;

private int maNganh;

public String toString() {

return maSinhVien + "," + hoTen + "," + maNganh + "\n";

}

}

## **Package: \_22T1020362.nganhrepo**

### **INganhRepo.java: Interface cho Repo của Nganh**

package \_22T1020362.nganhrepo;

import java.util.List;

import \_22T1020362.models.Nganh;

public interface INganhRepo {

List<Nganh> readNDT();

void insertNDT(Nganh ndt);

}

### **MongoDBNganhRepo.java: Repo cho ngành có dữ liệu trong MongoDB, chứa các phương thức**

package \_22T1020362.nganhrepo;

import java.util.ArrayList;

import java.util.List;

import org.bson.Document;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.model.Filters;

import \_22T1020362.models.Nganh;

import lombok.\*;

@AllArgsConstructor

public class MongoDBNganhRepo implements INganhRepo{

private MongoDatabase db;

@Override

public List<Nganh> readNDT(){

List<Nganh> lst = new ArrayList<>();

try {

MongoDatabase db = this.db;

MongoCollection<Document> collection = db.getCollection("nganh");

FindIterable<Document> documents = collection.find();

// Chuyển đổi Document thành Student

for (Document doc : documents) {

int maNganh = doc.getInteger("MaNganh");

String tenNganh = doc.getString("TenNganh");

Nganh ndt = new Nganh(maNganh, tenNganh);

lst.add(ndt);

}

} catch (Exception e) {

System.err.println("Lỗi kết nối MongoDB: " + e.getMessage());

}

return lst;

}

@Override

public void insertNDT(Nganh ndt){

try {

MongoDatabase db = this.db;

MongoCollection<Document> collection = db.getCollection("nganh");

Document found = collection.find(Filters.eq("MaNganh", ndt.getMaNganh())).first();

if(found != null) {

System.out.println("Ngành có MaNganh: " + ndt.getMaNganh() + " đã tồn tại trong hệ thống");

}

else {

Document newNDT = new Document("MaNganh", ndt.getMaNganh()).append("TenNganh", ndt.getTenNganh());

collection.insertOne(newNDT);

System.out.println("Đã chèn thành công ngành: " + newNDT.toJson());

}

} catch (Exception e) {

System.err.println("Lỗi khi chèn dữ liệu: " + e.getMessage());

}

}

}

### **MySQLNganhRepo.java: Repo cho Nganh chứa dữ liệu trong MySQL, chứa các phương thức**

package \_22T1020362.nganhrepo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import \_22T1020362.models.Nganh;

import lombok.\*;

@AllArgsConstructor

public class MySQLNganhRepo implements INganhRepo{

private Connection cn;

@Override

public List<Nganh> readNDT(){

List<Nganh> lst = new ArrayList<>();

String sql = "SELECT \* FROM Nganh;";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql);

ResultSet rs = pr.executeQuery();

while(rs.next()) {

Nganh ndt = new Nganh(rs.getInt(1), rs.getString(2));

lst.add(ndt);

}

rs.close(); pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

return lst;

}

@Override

public void insertNDT(Nganh ndt){

String sql1 = "SELECT \* FROM Nganh WHERE MaNganh LIKE " + "'" + ndt.getMaNganh() + "'";

String sql2 = "INSERT INTO Nganh (MaNganh, TenNganh) VALUES (?, ?)";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql1);

ResultSet rs = pr.executeQuery();

if(rs.isBeforeFirst()) {

System.out.println("Mã ngành đã tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

}

else {

pr = cn.prepareStatement(sql2);

pr.setInt(1, ndt.getMaNganh());

pr.setString(2, ndt.getTenNganh());

int rowInserted = pr.executeUpdate();

if(rowInserted > 0) {

System.out.println("Thêm mới ngành đào tạo có mã: " + ndt.getMaNganh() + " thành công");

}

else {

System.out.println("Lỗi không xác định, hệ thống đang kiểm tra lại, xin quý khách vui lòng đợi trong giây lát");

}

}

rs.close();

pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

}

}

### **TextNganhRepo.java: Repo cho Nganh có dữ liệu trong File Text, chứa các phương thức**

package \_22T1020362.nganhrepo;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.sql.Date;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import \_22T1020362.models.Nganh;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class TextNganhRepo implements INganhRepo{

private String fileUrl;

@Override

public List<Nganh> readNDT() {

List<Nganh> list = new ArrayList<>();

FileInputStream fis = null;

try {

fis = new FileInputStream(fileUrl);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Scanner sc = new Scanner(fis);

String line;

while(sc.hasNextLine()) {

line = sc.nextLine();

if(line.isEmpty()) break;

String [] lst = line.split(",");

Nganh ndt = new Nganh(Integer.valueOf(lst[0]), lst[1]);

list.add(ndt);

}

sc.close();

return list;

}

@Override

public void insertNDT(Nganh ndt) {

List<Nganh> lst = readNDT();

boolean check = false;

for(Nganh n: lst) {

if(n.getMaNganh() == ndt.getMaNganh()) {

System.out.println("Mã ngành đã tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

check = true;

}

}

if(!check) {

try {

FileWriter fw;

fw = new FileWriter(fileUrl, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write(ndt.toString());

bw.newLine();

bw.close();

System.out.println("Thêm mới ngành đào tạo có mã: " + ndt.getMaNganh() + " thành công");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

}

## **Package: \_22T1020362.sinhvienrepo**

### **INganhRepo.java: Interface cho Repo của SinhVien**

package \_22T1020362.sinhvienrepo;

import java.util.List;

import \_22T1020362.models.SinhVien;

public interface ISinhVienRepo {

List<SinhVien> readSV();

void deleteSV(String msv);

}

### **MongoDBSinhVienRepo.java: Repo cho sinh viên có dữ liệu trong MongoDB, chứa các phương thức**

package \_22T1020362.sinhvienrepo;

import java.util.ArrayList;

import java.util.List;

import org.bson.Document;

import com.mongodb.client.FindIterable;

import com.mongodb.client.MongoCollection;

import com.mongodb.client.MongoDatabase;

import com.mongodb.client.model.Filters;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class MongoDBSinhVienRepo implements ISinhVienRepo{

private MongoDatabase db;

@Override

public List<SinhVien> readSV(){

List<SinhVien> lst = new ArrayList<>();

try {

MongoDatabase db = this.db;

MongoCollection<Document> collection = db.getCollection("sinhvien");

FindIterable<Document> documents = collection.find();

// Chuyển đổi Document thành Student

for (Document doc : documents) {

String maSinhVien = doc.getString("MaSinhVien");

String hoTen = doc.getString("HoTen");

int maNganh = doc.getInteger("MaNganh");

SinhVien sv = new SinhVien(maSinhVien, hoTen, maNganh);

lst.add(sv);

}

} catch (Exception e) {

System.err.println("Lỗi kết nối MongoDB: " + e.getMessage());

}

return lst;

}

@Override

public void deleteSV(String msv){

try {

MongoDatabase db = this.db;

MongoCollection<Document> collection = db.getCollection("sinhvien");

//Tìm có SV không

Document found = collection.find(Filters.eq("MaSinhVien", msv)).first();

if(found != null) {

collection.deleteOne(Filters.eq("MaSinhVien", msv));

System.out.println("Đã xóa sinh viên có MaSinhVien: " + msv);

}

else {

System.out.println("Không tìm thấy sinh viên có MaSinhVien: " + msv);

}

} catch (Exception e) {

System.err.println("Lỗi khi xóa dữ liệu: " + e.getMessage());

}

}

}

### **MySQLSinhVienRepo.java: Repo cho SinhVien chứa dữ liệu trong MySQL, chứa các phương thức**

package \_22T1020362.sinhvienrepo;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class MySQLSinhVienRepo implements ISinhVienRepo{

private Connection cn;

@Override

public List<SinhVien> readSV(){

List<SinhVien> lst = new ArrayList<>();

String sql = "SELECT \* FROM SinhVien;";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql);

ResultSet rs = pr.executeQuery();

while(rs.next()) {

SinhVien s = new SinhVien(rs.getString(1), rs.getString(2), rs.getInt(3));

lst.add(s);

}

rs.close(); pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

return lst;

}

@Override

public void deleteSV(String msv){

String sql1 = "SELECT \* FROM SinhVien WHERE MaSinhVien LIKE " + "'" + msv + "'";

String sql2 = "DELETE FROM SinhVien WHERE MaSinhVien LIKE ?";

Connection cn = this.cn;

try {

PreparedStatement pr = cn.prepareStatement(sql1);

ResultSet rs = pr.executeQuery();

if(!rs.isBeforeFirst()) {

System.out.println("Mã sinh viên không tồn tại trong hệ thống, quý khách xin vui lòng kiểm tra lại.");

}

else {

pr = cn.prepareStatement(sql2);

pr.setString(1, msv);

int rowInserted = pr.executeUpdate();

if(rowInserted > 0) {

System.out.println("Xóa thành công sinh viên có mã: " + msv);

}

else {

System.out.println("Lỗi không xác định, hệ thống đang kiểm tra lại, xin quý khách vui lòng đợi trong giây lát");

}

}

rs.close();

pr.close();

}

catch(Exception e) {

e.printStackTrace();

}

}

}

### **TextSinhVienRepo.java: Repo cho SinhVien có dữ liệu trong File Text, chứa các phương thức**

package \_22T1020362.sinhvienrepo;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import \_22T1020362.models.SinhVien;

import lombok.\*;

@AllArgsConstructor

public class TextSinhVienRepo implements ISinhVienRepo{

private String fileUrl;

@Override

public List<SinhVien> readSV() {

List<SinhVien> list = new ArrayList<>();

FileInputStream fis = null;

try {

fis = new FileInputStream(fileUrl);

} catch (FileNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Scanner sc = new Scanner(fis);

String line;

while(sc.hasNextLine()) {

line = sc.nextLine();

if(line.isEmpty()) break;

String [] lst = line.split(",");

SinhVien sv = new SinhVien(lst[0], lst[1] , Integer.valueOf(lst[2]));

list.add(sv);

}

sc.close();

return list;

}

@Override

public void deleteSV(String msv) {

List<SinhVien> lst = readSV();

boolean removed = lst.removeIf(sv -> sv.getMaSinhVien().equals(msv));

if(removed) {

FileWriter fw;

try {

fw = new FileWriter(fileUrl, false);

BufferedWriter bw = new BufferedWriter(fw);

for(SinhVien sv: lst) {

bw.write(sv.toString());

bw.newLine();

}

bw.close();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Hệ thống đã xóa sinh viên có mã: " + msv);

}

else {

System.out.println("Hệ thống không tìm thấy sinh viên có mã: " + msv);

}

}

}

## **pom.xml:**

<project xmlns="https://maven.apache.org/POM/4.0.0"

xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>22T1020362</groupId>

<artifactId>22T1020362\_AbstractFactory</artifactId>

<version>0.0.1-SNAPSHOT</version>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.13.0</version>

<configuration>

<release>22</release>

</configuration>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>org.mongodb</groupId>

<artifactId>mongodb-driver-sync</artifactId>

<version>4.10.2</version>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<version>9.0.0</version>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.30</version>

<scope>provided</scope>

</dependency>

</dependencies>

</project>

# **b. Hình minh họa chạy chương trình (Cửa sổ Eclipse)**

A white background with black text

Description automatically generated

# **c. Script của cơ sở dữ liệu -- bao gồm cả lược đồ & dữ liệu**

Trong database/

## **mongodb/**

### **mtk\_abstractfactory.nganh.json:**

[{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d748"

},

"MaNganh": 101,

"TenNganh": "Điện tử viễn thông"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d749"

},

"MaNganh": 102,

"TenNganh": "Công nghệ thông tin"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d74a"

},

"MaNganh": 103,

"TenNganh": "Toán ứng dụng"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d74b"

},

"MaNganh": 104,

"TenNganh": "Văn học"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d74c"

},

"MaNganh": 105,

"TenNganh": "Báo chí & Truyền thông"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d74d"

},

"MaNganh": 109,

"TenNganh": "Lịch sử"

},

{

"\_id": {

"$oid": "67dd1da97ccdf56527c5d74e"

},

"MaNganh": 836,

"TenNganh": "Quản lý nhà nước"

}]

### **mtk\_abstractfactoy.sinhvien.json:**

[{

"\_id": {

"$oid": "67dd1e347ccdf56527c5d751"

},

"MaSinhVien": "T1020111",

"HoTen": "Lê Thị Hồng Hoa",

"MaNganh": 104

},

{

"\_id": {

"$oid": "67dd1e347ccdf56527c5d752"

},

"MaSinhVien": "T1020122",

"HoTen": "Trần Văn Quyến",

"MaNganh": 102

},

{

"\_id": {

"$oid": "67dd1e347ccdf56527c5d753"

},

"MaSinhVien": "T1020133",

"HoTen": "Nguyễn Văn Tiến",

"MaNganh": 105

},

{

"\_id": {

"$oid": "67dd1e347ccdf56527c5d754"

},

"MaSinhVien": "T1020234",

"HoTen": "Trần Nguyên Chính",

"MaNganh": 109

},

{

"\_id": {

"$oid": "67dd1e347ccdf56527c5d755"

},

"MaSinhVien": "T1020912",

"HoTen": "Lê Thị Linh",

"MaNganh": 836

}]

## **mysql/**

### **mysql-script.json:**

-- MySQL dump 10.13 Distrib 8.0.41, for Win64 (x86\_64)

--

-- Host: localhost Database: mtk\_abstractfactory

-- ------------------------------------------------------

-- Server version 8.0.41

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!50503 SET NAMES utf8 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Table structure for table `nganh`

--

DROP TABLE IF EXISTS `nganh`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `nganh` (

`MaNganh` **int** NOT NULL,

`TenNganh` **varchar**(50) CHARACTER SET utf8mb4 **COLLATE** utf8mb4\_0900\_ai\_ci NOT NULL,

**PRIMARY KEY** (`MaNganh`)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4 **COLLATE**=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `nganh`

--

LOCK TABLES `nganh` WRITE;

/\*!40000 ALTER TABLE `nganh` DISABLE KEYS \*/;

INSERT INTO `nganh` VALUES (101,'Điện tử viễn thông'),(102,'Công nghệ thông tin'),(103,'Toán ứng dụng'),(104,'Văn học'),(105,'Báo chí & Truyền thông'),(109,'Lịch sử'),(836,'Quản lý nhà nước');

/\*!40000 ALTER TABLE `nganh` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `sinhvien`

--

DROP TABLE IF EXISTS `sinhvien`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!50503 SET character\_set\_client = utf8mb4 \*/;

CREATE TABLE `sinhvien` (

`MaSinhVien` **varchar**(10) CHARACTER SET utf8mb4 **COLLATE** utf8mb4\_0900\_ai\_ci NOT NULL,

`HoTen` **varchar**(50) CHARACTER SET utf8mb4 **COLLATE** utf8mb4\_0900\_ai\_ci NOT NULL,

`MaNganh` **int** NOT NULL,

**PRIMARY KEY** (`MaSinhVien`),

KEY `PK\_SinhVien\_Nganh` (`MaNganh`),

**CONSTRAINT** `PK\_SinhVien\_Nganh` **FOREIGN KEY** (`MaNganh`) **REFERENCES** `nganh` (`MaNganh`)

) ENGINE=InnoDB **DEFAULT** CHARSET=utf8mb4 **COLLATE**=utf8mb4\_0900\_ai\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `sinhvien`

--

LOCK TABLES `sinhvien` WRITE;

/\*!40000 ALTER TABLE `sinhvien` DISABLE KEYS \*/;

INSERT INTO `sinhvien` VALUES ('T1020111','Lê Thị Hồng Hoa',104),('T1020122','Trần Văn Quyến',102),('T1020133','Nguyễn Văn Tiến',105),('T1020234','Trần Nguyên Chính',109),('T1020912','Lê Thị Linh',836);

/\*!40000 ALTER TABLE `sinhvien` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Dumping events for database 'mtk\_abstractfactory'

--

--

-- Dumping routines for database 'mtk\_abstractfactory'

--

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2025-03-21 21:21:41

# **d. File minh hoạ nguồn dữ liệu TEXT**

## **config.txt**

MySQL

Nganh

jdbc:mysql://localhost:3306/mtk\_abstractfactory?useUSSL=false|root|quang

## **sinhvien.txt**

T012001,Nguyễn Thị Hoa,102

T012005,Trần Văn Huy,104

T102002,Lê Văn Nam,105

## **nganh.txt**

102,Công nghệ thông tin

103,Báo chí