

# Lab 5: Tổng hợp DB4O

## 1 LINQ to DB4O

Nguồn tham khảo:

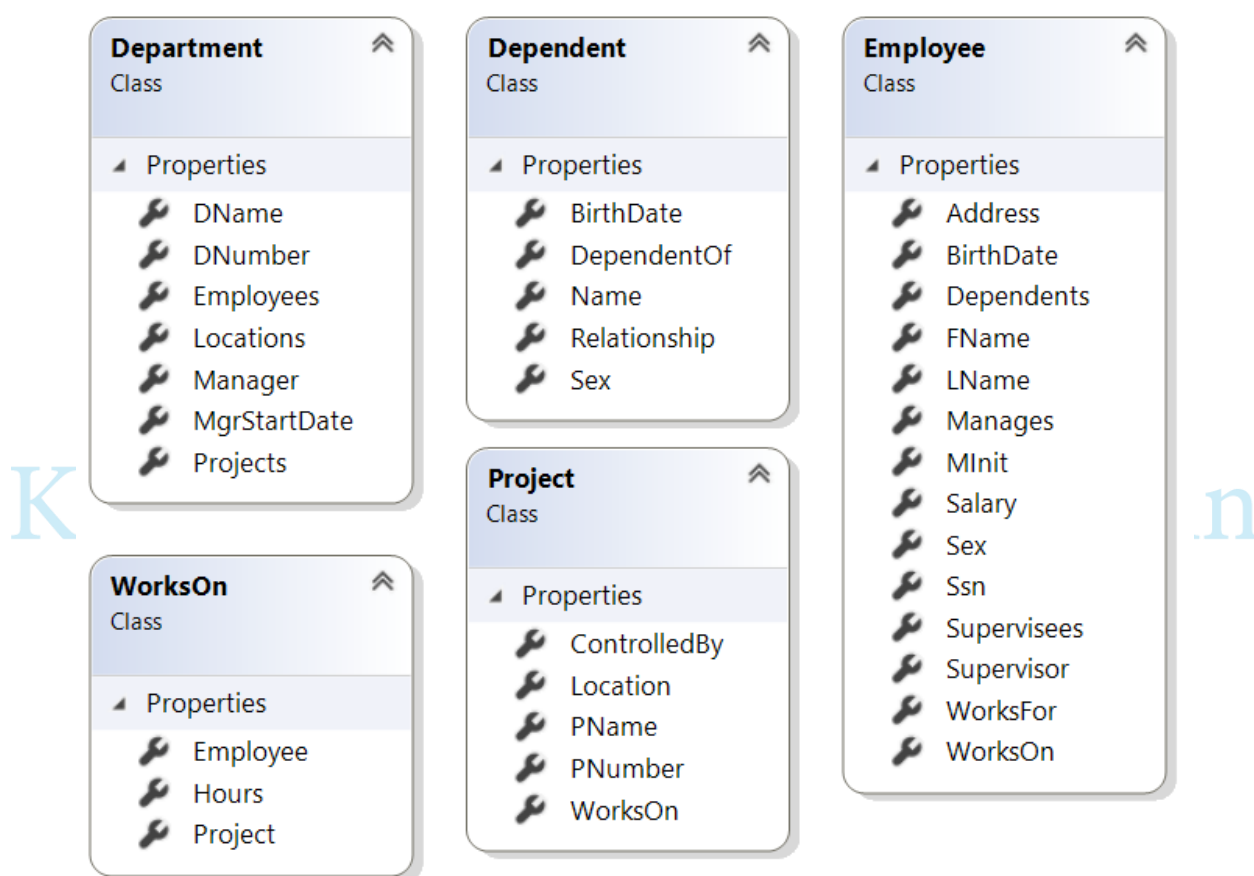
<https://www.codeproject.com/Articles/25669/An-introduction-to-LINQ-for-db4o>

Yêu cầu: Đọc và thực hiện lại bài viết trên nền Windows Form hoặc ASP.NET MVC Core.

## 2 Company Manager Project

### 2.1 Ví dụ đơn giản

Trong phần này, một cơ sở dữ liệu của công ty Elmasri-Navathe được mô tả như một cơ sở dữ liệu hướng đối tượng. Thiết kế bao gồm các lớp đối tượng: Employee, Department, Project, và Dependent. Để biểu diễn mối quan quan hệ nhiều – nhiều, một lớp worksOn được thiết kế. Các lớp và thuộc tính được định nghĩa như mô hình sau đây:



Sau đây là các lớp tương ứng

```

public class Department
{
    // attributes
    public int DNumber { get; set; }
    public string DName { get; set; }
    public List<string> Locations { get; set; }
    // relationships
    public List<Employee> Employees { get; set; }
    public Employee Manager { get; set; }
    public List<Project> Projects { get; set; }
    // one-to-many relationship (manager) attribute
    public string MgrStartDate { get; set; }
}

public class Employee
{
    // attributes
    public int Ssn { get; set; }
    public string FName { get; set; }
    public char MInit { get; set; }
    public string LName { get; set; }
    public string Address { get; set; }
    public string BirthDate { get; set; }
    public float Salary { get; set; }
    public char Sex { get; set; }
    //relationships
    public Department WorksFor { get; set; }
    public Department Manages { get; set; }
    public List<WorksOn> WorksOn { get; set; }
    public List<Dependent> Dependents { get; set; }
    public Employee Supervisor { get; set; }
    public List<Employee> Supervisees { get; set; }
}

public class Project
{
    // attributes
    public int PNumber { get; set; }
    public string PName { get; set; }
    public string Location { get; set; }
    // relationships
    public Department ControlledBy { get; set; }
    public List<WorksOn> WorksOn { get; set; }
}

public class Dependent
{
    // attributes
    public string Name { get; set; }
    public char Sex { get; set; }
    public string BirthDate { get; set; }
    public string Relationship { get; set; }
    // relationships
    public Employee DependentOf { get; set; }
}

```

```
public class WorksOn
{
    // attribute
    public float Hours { get; set; }
    //owner attributes
    public Employee Employee { get; set; }
    public Project Project { get; set; }
}
```

## 2.2 Xây dựng ứng dụng

Xây dựng lớp Database đọc dữ liệu từ tập tin văn bản và tạo ra CSDL hướng đối tượng. Lớp này được xây dựng như sau:

### 2.2.1 Mở/đóng CSDL

```
public class Database
{
    public static string DbFileName { get; set; }
    public static IObjectContainer DB => Db4oEmbedded.OpenFile(DbFileName);
    public static void CloseDB(IObjectContainer db)
    {
        db.Close();
    }
}
```

### 2.2.2 Tạo đối tượng Employee

Phương thức tạo đối tượng Employee, được viết trong class DataBase. Phương thức này đọc dữ liệu từ tập tin *employee.dat*, tập tin này chứa dữ liệu như sau:

40

*James:E:Borg:888665555:10-NOV-27:450 Stone, Houston, TX:M:55000*

*Franklin:T:Wong:333445555:08-DEC-45:638 Voss, Houston, TX:M:40000*

*Jennifer:S:Wallace:987654321:20-JUN-31:291 Berry, Bellaire, TX:F:43000*

...

Phương thức tạo đối tượng Employee như sau:

```
public static void CreateEmployees(IObjectContainer db, string fileName)
{
    if (File.Exists(fileName))
    {
        FileStream fs = new FileStream(fileName, FileMode.Open);
        StreamReader fin = new StreamReader(fs);
        int nEmps = int.Parse(fin.ReadLine());
        for (int i = 0; i < nEmps; i++)
        {
            string line = fin.ReadLine();
            if (line != null)
            {
                string[] fields = line.Split(':');
            }
        }
    }
}
```

```

        string fname = fields[0];
        char minit = fields[1][0];
        string lname = fields[2];
        int ssn = int.Parse(fields[3]);
        string bdate = fields[4];
        string address = fields[5];
        char sex = fields[6][0];
        float salary = float.Parse(fields[7]);
        Employee e = new Employee
        {
            Ssn = ssn,
            FName = fname,
            MInit = minit,
            LName = lname,
            Address = address,
            BirthDate = bdate,
            Salary = salary,
            Sex = sex
        };
        db.Store(e);
    }
}
fin.Close();
fs.Close();
}
}

```

### 2.2.3 Tạo đối tượng Dependents

Phương thức tạo đối tượng Dependents, được viết trong class DataBase. Phương thức này đọc dữ liệu từ tập tin Dependents.dat, tập tin này chứa dữ liệu như sau:

11

333445555,Alice,F,05-APR-1976,Daughter

333445555,Theodore,M,25-OCT-1973,Son

333445555,Joy,F,03-MAY-1948,Spouse

987654321,Abner,M,29-FEB-1932,Spouse

123456789,Michael,M,01-JAN-1978,Son

123456789,Alice,F,31-DEC-1978,Daughter

123456789,Elizabeth,F,05-MAY-1957,Spouse

444444400,Johnny,M,04-APR-1997,Son

444444400,Tommy,M,07-JUN-1999,Son

444444401,Chris,M,19-APR-1969,Spouse

444444402,Sam,M,14-FEB-1964,Spouse

Phương thức tạo đối tượng *Dependents* tương tự như phương thức tạo đối tượng *Employee*, có nhiệm vụ đọc dữ liệu từ tập tin và đưa vào CSDL.

```
public static void CreateDependents(IObjectContainer db, string fileName)
{
    // Viết code ở đây ...
}
```

#### 2.2.4 Tạo đối tượng Department

Phương thức tạo đối tượng *Department*, được viết trong class *DataBase*. Phương thức này đọc dữ liệu từ tập tin *department.dat*, tập tin này chứa dữ liệu như sau:

```
6
1:Headquarters:Houston
4:Administration:Stafford
5:Research:Bellaire,Sugarland,Houston
6:Software:Atlanta,Sacramento
7:Hardware:Milwaukee
8:Sales:Chicago,Dallas,Philadelphia,Seattle,Miami
```

Phương thức tạo đối tượng *Department* tương tự như phương thức tạo đối tượng *Employee*, có nhiệm vụ đọc dữ liệu từ tập tin *department.dat* và đưa vào CSDL:

```
public static void CreateDepartment(IObjectContainer db, string fileName)
{
    // Viết code ở đây ...
}
```

#### 2.2.5 Tạo đối tượng Project

Phương thức tạo đối tượng *Project*, được viết trong class *DataBase*. Phương thức này đọc dữ liệu từ tập tin *project.dat*, tập tin này chứa dữ liệu như sau:

```
11
1,Product X,Bellaire
2,Product Y,Sugarland
3,Product Z,Houston
10,Computerization,Stafford
```

20,Reorganization,Houston

30,New Benefits,Stafford

61,Operating Systems,Jacksonville

62,Database Systems,Birmingham

63,Middleware,Jackson

91,Inkjet Printers,Phoenix

92,Laser Printers,Las Vegas

Phương thức tạo đối tượng Project tương tự như phương thức tạo đối tượng Employee, có nhiệm vụ đọc dữ liệu từ tập tin *project.dat* và đưa vào CSDL:

```
public static void CreateProject(IObjectContainer db, string fileName)
{
    // Viết code ở đây ...
}
```

### 2.2.6 Tạo đối tượng WorksOn

Phương thức tạo đối tượng WorksOn, được viết trong class DataBase. Phương thức này đọc dữ liệu từ tập tin *workson.dat*, tập tin này chứa dữ liệu như sau:

48

123456789,1,32.5

123456789,2,7.5

666884444,3,40.0

453453453,1,20.0

Phương thức tạo đối tượng WorksOn tương tự như phương thức tạo đối tượng Employee, có nhiệm vụ đọc dữ liệu từ tập tin *workson.dat* và đưa vào CSDL:

```
public static void CreateWorkOn(IObjectContainer db, string fileName)
{
    // Viết code ở đây ...
}
```

### 2.2.7 Thiết lập quan hệ Manager cho đối tượng Employee:

Phương thức này thiết lập đối tượng tham chiếu cho quan hệ một – một, người quản lý. Tập tin dữ liệu chứa số lượng department, một mã số cho người quản lý và một ngày bắt đầu. Dữ liệu được cho như sau (lưu trữ trong tập tin *manager.dat*):

1,888665555,19-JUN-1971

4,987654321,01-JAN-1985

5,333445555,22-MAY-1978

6,111111100,15-MAY-1999

7,444444400,15-MAY-1998

8,555555500,01-JAN-1997

Sau đây là phương thức đọc dữ liệu từ tập tin văn bản và thiết lập manager cho lớp Employee:

```
public static void SetManagers(IObjectContainer db, string fileName)
{
    if (File.Exists(fileName))
    {
        FileStream fs = new FileStream(fileName, FileMode.Open);
        StreamReader fin = new StreamReader(fs);
        int nMgrs = int.Parse(fin.ReadLine());
        for (int i = 0; i < nMgrs; i++)
        {
            string line = fin.ReadLine();
            string[] fields = line.Split(',');
            int dno = int.Parse(fields[0]);
            int essn = int.Parse(fields[1]);
            string startDate = fields[2];
            IList<Department> depts = db.Query(delegate (Department dept)
            {
                return (dept.DNumber == dno);
            });
            Department d = null;
            if (depts != null)
            {
                d = depts[0];
            }
            IList<Employee> emps = db.Query(delegate (Employee emp)
            {
                return (emp.Ssn == essn);
            });
            Employee e = null;
            if (emps != null && emps.Count != 0)
            {
                e = emps[0];
            }
            if (e != null && d != null)
            {
                d.MgrStartDate = startDate;
                e.Manages = d;
                d.Manager = e;
                db.Store(d);
                db.Store(e);
            }
        }
    }
}
```

Tương tự, ta tiến hành thiết lập các phương thức:

**setControls:** thiết lập cho thuộc tính `controlledBy` của lớp `Project`:

```
5
1:20
4:10,30
5:1,2,3
6:61,62,63
7:91,92
```

**setWorksFor:** thiết lập thuộc tính liên quan giữa hai đối tượng: *Employee & Department*:

```
6
1:888665555
4:987654321,987987987,999887777
5:123456789,333445555,666884444,453453453
```

**setSupervisors:** thiết lập thuộc tính liên quan giữa *Employee & Employee*:

```
19
888665555:333445555,987654321
333445555:123456789
987654321:999887777,987987987
333445555:666884444,453453453
111111100:111111101,111111102,111111103
222222200:222222201,222222202,222222203
222222201:222222204,222222205
```

## 2.2.8 Truy vấn phức tạp

Trong phần này, ta thực hiện một câu truy vấn như sau: *lấy những departments mà có location tại Houston hoặc có ít hơn 4 employees hoặc điều khiển 1 dự án tọa lạc tại Phoenix.*

Ta thực hiện code như sau:

```
public void ComplexRetrievalExample(IObjectContainer db)
{
    try
    {
        IList<Department> depts = db.Query(delegate (Department dept)
        {
            int nEmps = dept.Employees.Count;
            List<Project> prjs = dept.Projects;
            bool foundPhoenix = false;
            for (int i = 0; i < prjs.Count; i++)
            {
                Project p = prjs[i];
                if (p.Location.Equals("Phoenix"))
                {
                    foundPhoenix = true;
                    break;
                }
            }
            return dept.Locations.Contains("Houston") || (nEmps < 4) ||
foundPhoenix;
        });
        //for (int i = 0; i < depts.Count; i++)
```



```
        // Console.WriteLine("Department: " + depts[i].DName);  
    }  
    catch { }  
}
```

## 2.3 Yêu cầu

- Thực hiện hết những phần còn lại của các phần ở trên.
- Thực hiện các câu lệnh thêm, xóa, sửa trên các bảng.
- Thực hiện các câu truy vấn sau đây:
  - *Tìm tất cả những người làm thuê có giới tính là nam.*
  - *Tìm tất cả những người làm thuê đã làm việc cho ban ngành có tên có chữ T ở đầu.*
  - *Tìm tất cả những người làm thuê đã làm cho dự án có tên chứa chữ “A” và có thời gian làm việc là <10h.*
  - *Tìm tất cả các công việc được thực hiện bởi dự án có địa điểm là Houston.*

Khoa Công Nghệ Thông Tin  
ĐẠI HỌC SƯ PHẠM