HIBERNATE

MAPPING INHERITANCE RELATIONSHIPS

GVHD: Thầy Nguyễn Hoàng Anh

Sinh viên: 0812463 – Hồ Văn Tấn
0812505 – Sử Bá Thuần

Các vấn đề trình bày

1. Đặt vấn đề.
2. Ba cách Mapping.
3. Demo + Hỏi đáp.

ĐẶT VẤN ĐỀ

Quan hệ kế thừa và Hibernate

- Hệ thống hướng đối tượng có thể mô hình cả 2 loại mối quan hệ “is a” và “has a”. Trong khi đó mô hình quan hệ chỉ hỗ trợ 1 mối quan hệ là “has a” giữa 2 thực thể.

→Hibernate hỗ trợ 3 chiến lược để mapping các đối tượng thành các bảng quan hệ.

  - One table per Concrete class
  - One table per Subclass
  - One table per class Hierarchy

3 CÁCH MAPPING

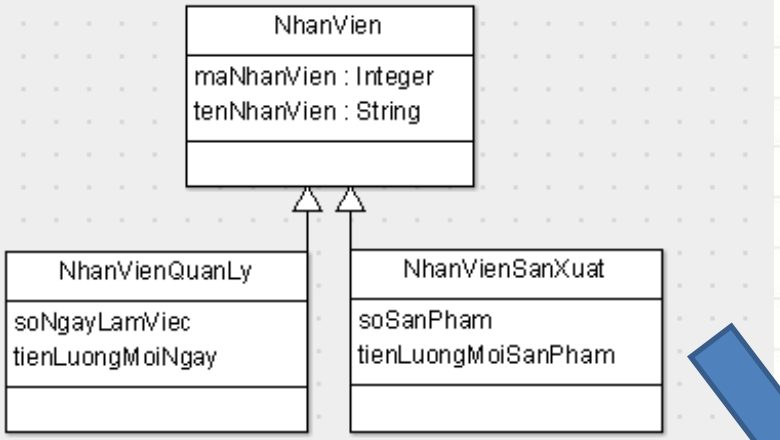
ONE TABLE PER CONCRETE CLASS

- Chiến lược
- Mô hình
- Mapping
- Ưu – nhược điểm

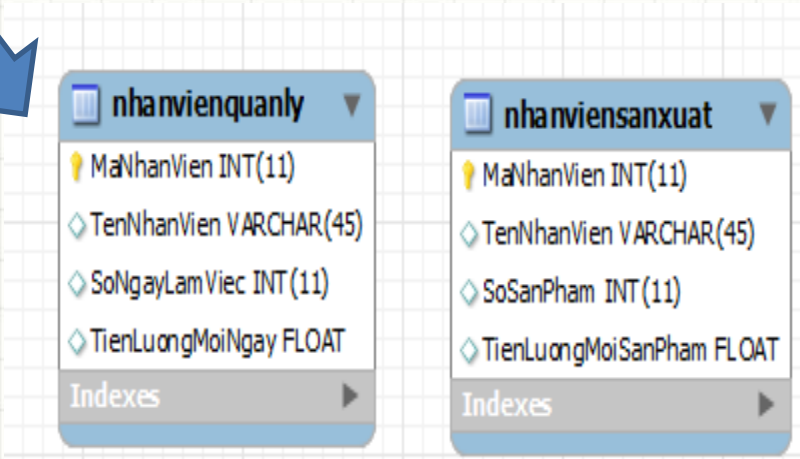
Chiến lược

- Mỗi class con sẽ được mapping bằng 1 table.
- Mỗi table sẽ chứa đầy đủ các thuộc tính của lớp cha.

Mô hình



Mô hình các class



Mô hình CSDL

POJO - NhanVien

```
  public class NhanVien implements java.io.Serializable{
12      private int maNhanVien;
13      private String tenNhanVien;
14
15      /**...*/
18      public int getMaNhanVien() {...}
21
22      /**...*/
25      public void setMaNhanVien(int maNhanVien) {...}
28
29      /**...*/
32      public String getTenNhanVien() {...}
35
36      /**...*/
39      public void setTenNhanVien(String tenNhanVien) {...}
42
43      public NhanVien(int maNhanVien, String tenNhanVien) {...}
47
48      public NhanVien() {...}
50  }
```

POJO - NhanVienQuanLy

```
11  public class NhanVienQuanLy extends NhanVien implements java.io.Serializable {
12      private int soNgayLamViec;
13      private float tienLuongMoiNgay;
14
15      public int getSoNgayLamViec() {...}
18
19      public void setSoNgayLamViec(int soNgayLamViec) {...}
22
23      public float getTienLuongMoiNgay() {...}
26
27      public void setTienLuongMoiNgay(float tienLuongMoiNgay) {...}
30
31      public NhanVienQuanLy(int soNgayLamViec, float tienLuongMoiNgay,
32              int maNhanVien, String tenNhanVien) {...}
37
38      public NhanVienQuanLy() {...}
40
41  }
```

POJO - NhanVienSanXuat

```
11  public class NhanVienSanXuat extends NhanVien implements java.io.Serializable {
12      private int soSanPham;
13      private float tienLuongMoiSanPham;
14
15      public int getSoSanPham() {...}
18
19      public void setSoSanPham(int soSanPham) {...}
22
23      public float getTienLuongMoiSanPham() {...}
26
27      public void setTienLuongMoiSanPham(float tienLuongMoiSanPham) {...}
30
31      public NhanVienSanXuat(int soSanPham, float tienLuongMoiSanPham) {...}
35
36      public NhanVienSanXuat(int soSanPham, float tienLuongMoiSanPham,
37              int maNhanVien, String tenNhanVien) {...}
42
43      public NhanVienSanXuat() {...}
45  }
```

Mapping

- Thực hiện mapping như bình thường.
- Không yêu cầu phải có file mapping cho lớp cha.

Mapping – NhanVienQuanLy

```
3  <hibernate-mapping>
4      <class name="pojo.NhanVienQuanLy" table="nhanvienquanly">
5          <id name="maNhanVien" type="integer">
6              <generator class="assigned" />
7          </id>
8          <property name="tenNhanVien" column="TenNhanVien" type="string" />
9          <property name="soNgayLamViec" column="SoNgayLamViec" type="integer" />
10         <property name="tienLuongMoiNgay" column="TienLuongMoiNgay" type="float" />
11     </class>
12 </hibernate-mapping>
```

Mapping – NhanVienSanXuat

```
3  <hibernate-mapping>
4      <class name="pojo.NhanVienSanXuat" table="nhanviensanxuat">
5          <id name="maNhanVien" type="integer">
6              <generator class="assigned" />
7          </id>
8          <property name="tenNhanVien" column="TenNhanVien" type="string" />
9          <property name="soSanPham" column="SoSanPham" type="integer" />
10         <property name="tienLuongMoiSanPham" column="TienLuongMoiSanPham" type="float" />
11     </class>
12 </hibernate-mapping>
```

DAO

```
21      public static List<NhanVienQuanLy> layDsNVQL() {
22          List<NhanVienQuanLy> list = null;
23          SessionFactory sf = MyHibernateUtil.getSessionFactory();
24          Session ss = sf.getCurrentSession();
25          Transaction trans = ss.beginTransaction();
26          trans.begin();
27          list = ss.createQuery("from pojo.NhanVienQuanLy").list();
28          trans.commit();
29          return list;
30      }
```

```
32      public static List<NhanVien> layDsNVQL1() {
33          List<NhanVien> list = null;
34          SessionFactory sf = MyHibernateUtil.getSessionFactory();
35          Session ss = sf.getCurrentSession();
36          Transaction trans = ss.beginTransaction();
37          trans.begin();
38          list = ss.createQuery("from pojo.NhanVienQuanLy").list();
39          trans.commit();
40          return list;
41      }
```

App

```
24              List<NhanVienQuanLy> list = dao.NVQuanLyDAO.layDsNVQL();
25
26              for (int i = 0; i < list.size(); ++i) {
27                  System.out.print(list.get(i).getTenNhanVien());
28                  System.out.print(" - ");
29                  System.out.print(list.get(i).getSoNgayLamViec());
30                  System.out.print(" - ");
31                  System.out.println(list.get(i).getTienLuongMoiNgay());
32              }
```

MaNhanVien	TenNhanVien	SoNgayLamViec	TienLuongMoiNgay
1	Ho Van Tan	10	1000
2	Su Ba Thuan	29	299
NULL	NULL	NULL	NULL

CSDL

```
Ho Van Tan - 10 - 1000.0
Su Ba Thuan - 29 - 299.0
```

Kết quả

App

```
26              List<NhanVien> list = dao.NVQuanLyDAO.layDsNVQL1();
27
28              for (int i = 0; i < list.size(); ++i) {
29
30                  System.out.print(list.get(i).getTenNhanVien());
31                  System.out.print(" - ");
32                  System.out.print(((NhanVienQuanLy)(list.get(i))).getSoNgayLamViec());
33                  System.out.print(" - ");
34                  System.out.println(((NhanVienQuanLy)(list.get(i))).getTienLuongMoiNgay());
35              }
```

MaNhanVien	TenNhanVien	SoNgayLamViec	TienLuongMoiNgay
1	Ho Van Tan	10	1000
2	Su Ba Thuan	29	299
NULL	NULL	NULL	NULL

CSDL

```
Ho Van Tan - 10 - 1000.0
Su Ba Thuan - 29 - 299.0
```

Kết quả

Ưu điểm

- Đơn giản, dễ thực hiện nhất.

Nhược điểm

- Khi muốn tạo liên kết từ lớp cha tới 1 lớp khác thì phải tạo mapping tới tất cả các lớp con của nó.
- Khi muốn thay đổi các thuộc tính của lớp cha sẽ phải thay đổi lại trong những table của các lớp con.
- Nếu muốn truy vấn lớp cha, bạn phải truy vấn lớp con nhiều lần.

→"quick-and-dirty solutions."

Demo

ONE TABLE PER SUBCLASS

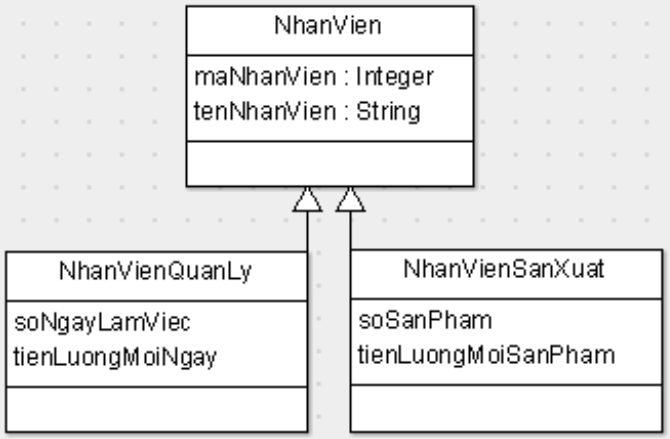
- Chiến lược
- Mô hình
- Mapping
- Ưu – nhược điểm

Chiến lược

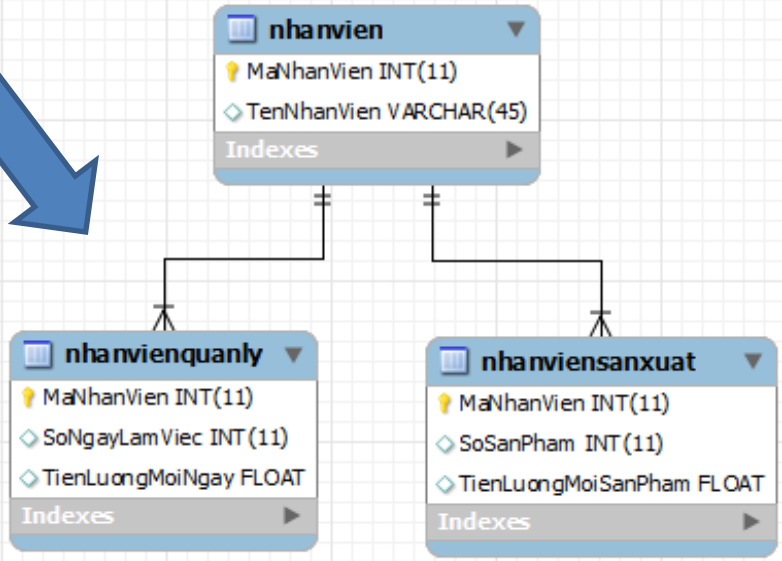
- Mỗi class sẽ được mapping bằng 1 table, bao gồm cả class abstract và interface class.

- Chuyển mối quan hệ “is a” trong phân cấp class thành mối quan hệ “has a” cho mỗi thực thể trong schema.

Mô hình



Mô hình các class



Mô hình CSDL

Mapping - NhanVien

```
3  <hibernate-mapping>
4      <class name="pojo.NhanVien" table="nhanvien">
5          <id name="maNhanVien" type="integer" column="MaNhanVien">
6              <generator class="assigned" />
7          </id>
8          <property name="tenNhanVien" column="TenNhanVien" type="string" />
9      </class>
10 </hibernate-mapping>
```

Mapping - NhanVienSanXuat

```
3  <hibernate-mapping>
4      <joined-subclass name="pojo.NhanVienSanXuat"
5          extends="pojo.NhanVien" table="nhanviensanxuat">
6          <key column="MaNhanVien"/>
7          <property name="soSanPham" type="integer">
8              <column name="SoSanPham"/>
9          </property>
10         <property name="tienLuongMoiSanPham" type="float" >
11             <column name="TienLuongMoiSanPham" />
12         </property>
13     </joined-subclass>
14 </hibernate-mapping>
```

Mapping - NhanVienQuanLy

```
3  <hibernate-mapping>
4      <joined-subclass name="pojo.NhanVienQuanLy"
5          extends="pojo.NhanVien" table="nhanvienquanly">
6          <key column="MaNhanVien"/>
7          <property name="soNgayLamViec" type="integer"
8              column="SoNgayLamViec"/>
9          <property name="tienLuongMoiNgay" type="float"
10             column="TienLuongMoiNgay"/>
11     </joined-subclass>
12 </hibernate-mapping>
```

DAO

```
21      public static List<NhanVienSanXuat> layDsNVSX() {
22          List<NhanVienSanXuat> list = null;
23          String sql = "from NhanVienSanXuat";
24          SessionFactory sf = MyHibernateUtil.getSessionFactory();
25          Session ss = sf.getCurrentSession();
26          Transaction trans = ss.beginTransaction();
27          list = ss.createQuery(sql).list();
28          trans.commit();
29          return list;
30      }
```

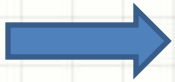
```
32      public static void themNV(NhanVienSanXuat info) {
33          SessionFactory sf = MyHibernateUtil.getSessionFactory();
34          Session ss = sf.getCurrentSession();
35          Transaction trans = ss.beginTransaction();
36          ss.saveOrUpdate(info);
37          trans.commit();
38      }
```

```
40      public static List<NhanVien> layDsNV() {
41          List<NhanVien> list = null;
42          String sql = "from NhanVien";
43          SessionFactory sf = MyHibernateUtil.getSessionFactory();
44          Session ss = sf.getCurrentSession();
45          Transaction trans = ss.beginTransaction();
46          list = ss.createQuery(sql).list();
47          trans.commit();
48          return list;
49      }
```

App

```
24          List<NhanVien> list = dao.NhanVienSanXuatDAO.layDsNV();
25          for (int i = 0; i < list.size(); ++i) {
26              System.out.println(list.get(i).getTenNhanVien());
27          }
```

	MaNhanVien	TenNhanVien
▶	1	Hồ Văn Tấn
	2	Sử Bá Thuần
	3	Hoàng Đình Trung
	4	Lê Văn Tám
	5	Nguyễn Viết Xuân
*	NULL	NULL



```
Hồ Văn Tấn
Sử Bá Thuần
Hoàng Đình Trung
Lê Văn Tám
Nguyễn Viết Xuân
```

Table nhanvien

App

```
25              List<NhanVienSanXuat> list = dao.NhanVienSanXuatDAO.layDsNVSX();
26              for (int i = 0; i < list.size(); ++i) {
27                  System.out.print(list.get(i).getMaNhanVien());
28                  System.out.print("-");
29                  System.out.print(list.get(i).getTenNhanVien());
30                  System.out.print("-");
31                  System.out.print(list.get(i).getSoSanPham());
32                  System.out.print("-");
33                  System.out.println(list.get(i).getTienLuongMoiSanPham());
34              }
```

MaNhanVien	TenNhanVien
1	Hồ Văn Tấn
2	Sử Bá Thuần
3	Hoàng Đình Trung
4	Lê Văn Tám
5	Nguyễn Viết Xuân
NULL	NULL

Table nhanvien

MaNhanVien	SoSanPham	TienLuongMoiSanPham
2	20	2000
5	30	3000
NULL	NULL	NULL

Table nhanviensanxuat

```
2-Sử Bá Thuần-20-2000.0
5-Nguyễn Viết Xuân-30-3000.0
```

App

```
36          NhanVienSanXuat nvsx = new NhanVienSanXuat(13, 6000, 7, "Văn Kinh Luân");
37          dao.NhanVienSanXuatDAO.themNV(nvsx);
```

MaNhanVien	TenNhanVien
1	Hồ Văn Tấn
2	Sử Bá Thuần
3	Hoàng Đình Trung
4	Lê Văn Tám
5	Nguyễn Viết Xuân
7	V?n Kinh Luân

Table nhanvien khi thêm

MaNhanVien	SoSanPham	TienLuongMoiSanPham
2	20	2000
5	30	3000
7	13	6000
NULL	NULL	NULL

Table nhanviensanxuat khi thêm

Lỗi font tiếng Việt khi lưu xuống CSDL

→ Thêm đoạn sau vào phần cấu hình CSDL trong file config Hibernate

?useUnicode=true&characterEncoding=UTF-8

```
jdbc:mysql://localhost:3306/quanlynhanvien?useUnicode=true&amp;characterEncoding=UTF-8
```

Ưu điểm

- Dễ quản lý
- Không đòi hỏi sự thay đổi phức tạp trong schema khi mà lớp cha được thay đổi.
- Giống với hầu hết các JVMs manager là dữ liệu được ẩn.

Nhược điểm

- Hiệu suất thấp.
- Khi phân cấp lớp lớn lên thì số lượng yêu cầu tham gia vào khởi tạo của lớp lá cũng lớn lên.
- Kỹ thuật này dùng tốt với các phân lớp thấp. Các phân lớp sâu thì không nên sử dụng kỹ thuật này vì hiệu suất sẽ rất thấp.

→ Nếu như vấn đề hiệu suất được bỏ qua thì nên chọn kỹ thuật này.

Demo

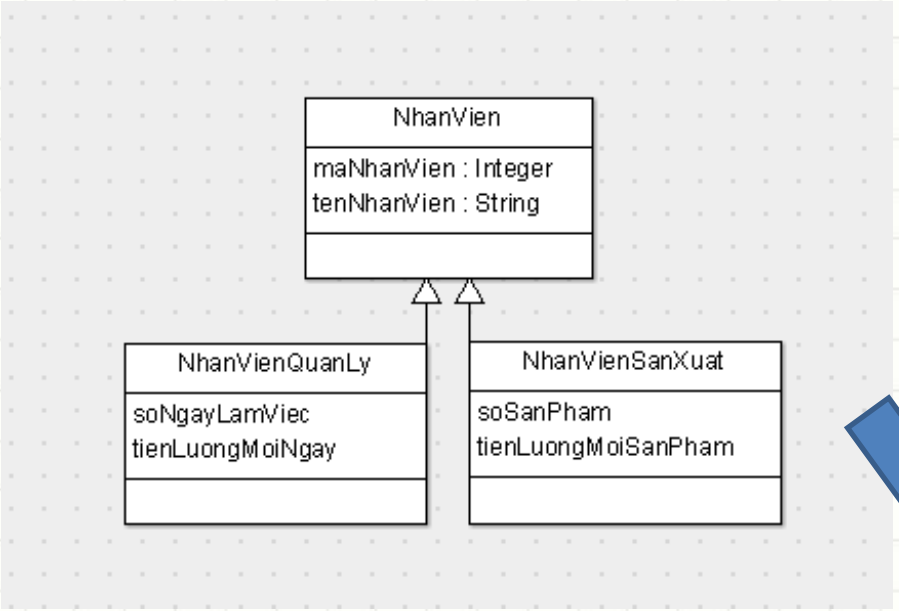
ONE TABLE PER CLASS HIERARCHY

- Chiến lược
- Mô hình
- Mapping
- Ưu – nhược điểm

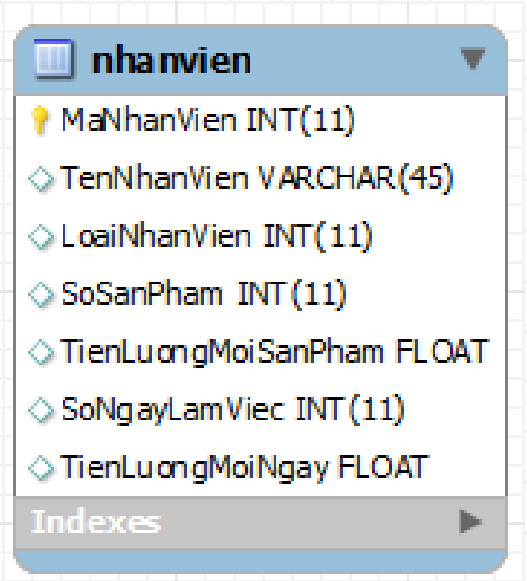
Chiến lược

- Sử dụng 1 table duy nhất.
- Các thuộc tính của các class con sẽ được thêm vào table này.
- Sử dụng 1 cột chứa khóa để xác định lớp con nào đang được đại diện bởi mỗi hàng trong table.

Mô hình



Mô hình các class



Mô hình CSDL

Mapping - NhanVien

```
<hibernate-mapping>
    <class name="entity.Nhanvien" table="nhanvien" catalog="quanlynhanvien2">
        <id name="maNhanVien" type="integer">
            <column name="MaNhanVien" />
            <generator class="assigned" />
        </id>
        <discriminator column="LoaiNhanVien" type="string"/>
        <property name="tenNhanVien" type="string">
            <column name="TenNhanVien" length="45" />
        </property>
         <subclass name="entity.Nhanviensanxuat" discriminator-value="1">
            <property name="soSanPham" type="integer" column="SoSanPham"/>
            <property name="tienLuongMoiSanPham" column="TienLuongMoiSanPham" type="float"/>
        </subclass>
        <subclass name="entity.Nhanvienquanly" discriminator-value="2">
            <property name="soNgayLamViec" column="SoNgayLamViec" type="integer"/>
            <property name="tienLuongMoiNgay" column="TienLuongMoiNgay" type="float"/>
        </subclass>

    </class>
</hibernate-mapping>
```

DAO

```
20      public static List<Nhanvien> getDSNV()
21      {
22          SessionFactory factory = HibernateUtil.getSessionFactory();
23          Session ss = factory.getCurrentSession();
24          ss.getTransaction().begin();
25          String hql = "from Nhanvien";
26          Query query = ss.createQuery(hql);
27          List<Nhanvien> dsnv = query.list();
28          return dsnv;
29      }
```

Thêm hành tính lương cho các pojo

- NhanVien

```
43      public float tinhLuong()
44      {
45          return 0;
46      }
```

- NhanVienQuanLy

```
40      public float tinhLuong()
41      {
42          return soNgayLamViec*tienLuongMoiNgay;
43      }
```

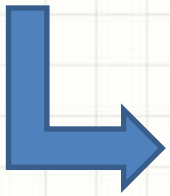
- NhanVienSanXuat

```
40      public float tinhLuong()
41      {
42          return soSanPham*tienLuongMoiSanPham;
43      }
```

App

```
21          List<Nhanvien> dsnv = dao.NhanVienDAO.getDSNV();
22          System.out.println("Danh sách nhân viên: ");
23          for(int i =0 ; i<dsnv.size(); i++)
24          {
25              System.out.println(dsnv.get(i).getTenNhanVien() + " - Tien luong: "
26                      + dsnv.get(i).tinhLuong());
27          }
```

MaNhanVien	TenNhanVien	LoaiNhanVien	SoSanPham	TienLuongMoiSanPham	SoNgayLamViec	TienLuongMoiNgay
1	Sử Bá Thuần	1	12	20	NULL	NULL
2	Hồ Văn Tấn	2	NULL	NULL	19	80000
3	Trương Đan Phong	1	55	50	NULL	NULL
4	Trần Thái Hòa	2	NULL	NULL	1	100000
NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
Danh sách nhân viên:
Sử Bá Thuần - Tien luong: 240.0
Hồ Văn Tấn - Tien luong: 1520000.0
Trương Đan Phong - Tien luong: 2750.0
Trần Thái Hòa - Tien luong: 100000.0
```

Ưu điểm

- Cung cấp hiệu suất tốt ngay cả với các hệ thống phân cấp kế thừa sâu.

Nhược điểm

- Thay đổi các thành phần của phân lớp thường sẽ phải yêu cầu các column được sửa đổi, thêm hoặc xóa khỏi table trong CSDL

→ Chậm.

- Khi phân cấp lớp lớn lên thì sẽ có rất nhiều column được yêu cầu bởi table này.

Demo

QUESTIONS?