

Họ và Tên : Trương Thái Đan Huy

MSSV : 52100222

ASSIGNMENT

Phần 1:

Đặc tả :

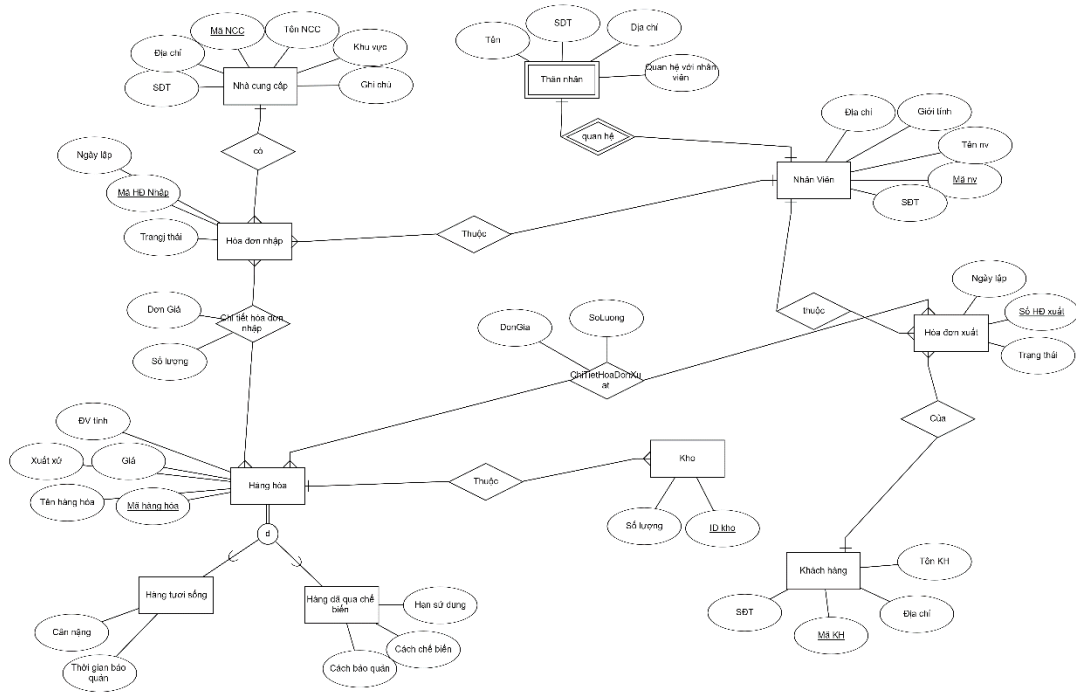
Một nhà cung cấp cần xây dựng cho mình một mô hình quản lý chuỗi cung ứng của mình. Nhà cung cấp hiện tại cần các hoạt động sau :

- Nhà cung cấp cần bán các loại hàng hóa có thông tin như nhau :Mã hàng,Tên hàng hóa, xuất xứ, giá bán, đơn vị tính. Ngoài các thuộc tính chung thì hàng hóa chia làm hai loại Hàng tươi sống :Cân nặng, Thời gian bảo quản. Hàng qua chế biến:Hạn sử dụng, cách sử dụng, cách bảo quản.
- Nhà cung cấp không thể làm được hết các công việc nên thuê thêm nhân viên, mỗi nhân viên đều có mã riêng và cần lưu các thông tin thêm như : Tên, giới tính, năm sinh, địa chỉ, SĐT.
- Mỗi nhân viên sẽ ghi thêm cho mình một thân nhân gồm các thông tin : Tên, SĐT, địa chỉ, quan hệ với nhân viên là gì.
- Nhà cung cấp cũng cần các hoạt động nhập, xuất hàng hóa, các hoạt động này thông qua một kho chung.
- Kho cần có mã kho và số lượng hàng hóa.
- Nhà cung cấp sẽ quản rất nhiều hóa đơn nhập, nhân viên phụ trách tạo và xử lý các hóa đơn nhập này. Hóa đơn nhập gồm có thông tin sau : trạng thái hóa đơn, ngày lập, Mã hóa đơn. Hóa đơn nhập có rất nhiều sản phẩm, khi nhập cần có số lượng và đơn giá của hàng hóa đó.
- Các khách hàng khi sử dụng dịch vụ bên nhà cung cấp cần để lại thông tin cá nhân như : Mã khách hàng, tên, SĐT, địa chỉ.
- Khi tạo hóa đơn xuất cần có các thông tin sau : Ngày lập, mã hóa đơn xuất, trạng thái. Do hóa đơn có nhiều hàng hóa nên cần có thêm đơn giá và số lượng hàng hóa trong lúc nhập. Một khách hàng có thể mua nhiều hóa đơn. Một nhân viên sẽ trực nhiều hóa đơn khác nhau.
- Cần phải làm tự động các thao tác nhập xuất sẽ tự tăng giảm số lượng hàng hóa trong kho. Mặc định tạo mã nhân viên theo thứ tự 'NV001' và khách hàng 'KH0001'
- Nhà cung cấp có các thông tin cho khách hàng biết : Mã NCC, Địa chỉ, SĐT, ghi chú, Khu vực, Tên nhà cung cấp.

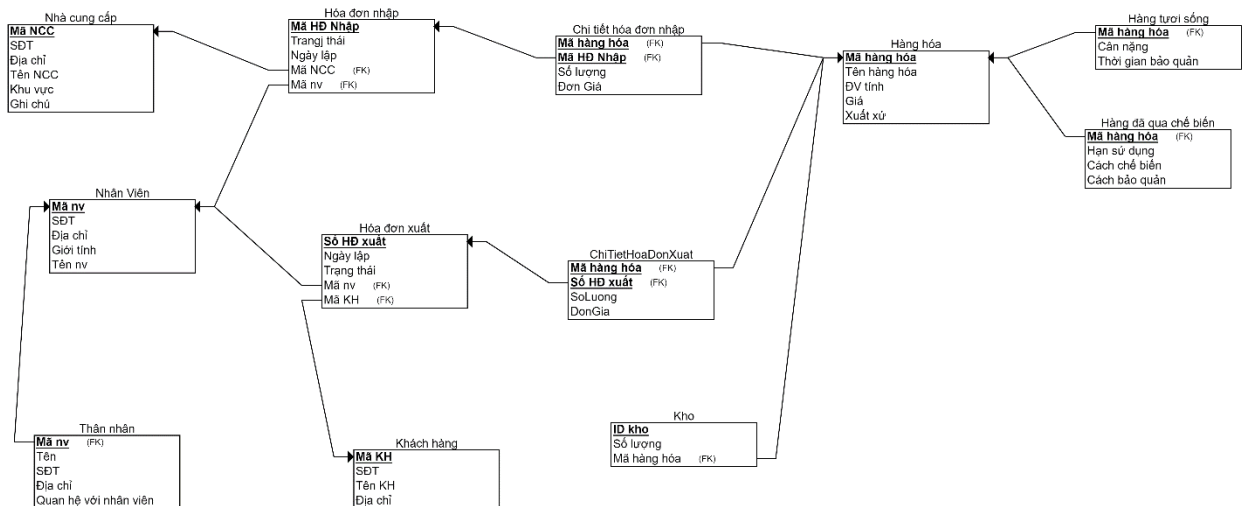
Yêu cầu :

1. Các quan hệ:
 - a) Quan hệ 1-n: Nhà cung cấp – hóa đơn nhập, Nhân viên – Hóa đơn xuất, Khách hàng – HD xuất
 - b) Quan hệ n – n : Hóa đơn nhập – hàng hóa, Hóa đơn xuất – Hàng Hóa
 - c) Quan hệ 1 – 1: Nhân viên – Thân Nhân
 - d) Quan hệ cha con : Hàng Hóa – Hàng tươi sống, Hàng qua chế biến
 - e) Quan hệ giữa thực thể mạnh và yếu : Nhân Viên – Thân Nhân

2. Vẽ erd



3. Chuyển sang mô hình quan hệ :



4. Thực hiện lệnh SQL trong file QLCH.sql

Phần 2 :

Sử dụng ngôn ngữ python để làm bài 1 và 2

Bài 1 gồm các file : Bai1.py, input1.txt, output1.txt

```

import numpy as np
f=open('input1.txt','r')
qa=f.readlines()
thucthe=qa[0].strip().split(';')
quanhe=qa[1].split(';')
def tachthuoctinh(a):
    khoavsthuoctinh=[]
    khoavsthuoctinh.append(a.split(':')[0])
    tt=a.split(':')[1]
    thuoctinh=tt.split(',')
    for i in range(0,len(thuoctinh)):
        khoavsthuoctinh.append(thuoctinh[i])
    return khoavsthuoctinh
def timkhoachinh(a):
    for i in range(0,len(a)):
        if(a[i][:3]=='PK.'):
            return a[i].split('.')[1]
def tachquanhe(a):
    qhevsthuoctinh=a.split(':')
    qhe=qhevsthuoctinh[0].split('-')
    thuoctinh=qhevsthuoctinh[1].split('-')
    return qhe,thuoctinh
def mohinhquanhe1n(a,b):
    b.append('FK.'+timkhoachinh(a))
def mohinhquanhe1l(a,b):
    b.append('FK.'+timkhoachinh(a))
def mohinhquanhenn(a,b):
    c=[]
    c.append('PK.FK.'+timkhoachinh(a))
    c.append('PK.FK.'+timkhoachinh(b))
    return c
def mohinhquanhechacon(a,b):
    b.append('PK.FK.'+timkhoachinh(a))
def convertstr(a):
    r=a[0]+':'
    for i in range(1,len(a)):
        r=r+a[i]+','
    result = ','.join([s.strip() for s in r.split(',') if s.strip()!=''])
    return result
def timdatribi(a):
    for i in range(0,len(a)):
        if(a[i][:3]=='DT.'):
            return a[i].split('.')[1]
listthucthe=[]
for i in range(0,len(thucthe)):

```

```

listthucethe.append(tachthuoctinh(thucethe[i]))
quanhebang=[]
for i in range(0,len(quanhe)):
    x,y=tachquanhe(quanhe[i])
    if(x[0]=='1' and x[1]=='1'):
        for z in range(0,len(listthucethe)):
            if(y[0]==listthucethe[z][0]):
                a=z
            if(y[1]==listthucethe[z][0]):
                b=z
        mohinhquanhe11(listthucethe[a], listthucethe[b])
        bang11='1-1:'+listthucethe[a][0]+'-'+listthucethe[b][0]
        quanhebang.append(bang11)
    if(x[0]=='1' and x[1]=='n'):
        for z in range(0,len(listthucethe)):
            if(y[0]==listthucethe[z][0]):
                a=z
            if(y[1]==listthucethe[z][0]):
                b=z
        mohinhquanhe1n(listthucethe[a], listthucethe[b])
        bang1n='1-n:'+listthucethe[a][0]+'-'+listthucethe[b][0]
        quanhebang.append(bang1n)
    if(x[0]=='cha' and x[1]=='con'):
        for z in range(0,len(listthucethe)):
            if(y[0]==listthucethe[z][0]):
                a=z
            if(y[1]==listthucethe[z][0]):
                b=z
        mohinhquanhechacon(listthucethe[a], listthucethe[b])
        bang1n='cha-con:'+listthucethe[a][0]+'-'+listthucethe[b][0]
        quanhebang.append(bang1n)
    if(x[0]=='n' and x[1]=='n'):
        for z in range(0,len(listthucethe)):
            if(y[0]==listthucethe[z][0]):
                a=z
            if(y[1]==listthucethe[z][0]):
                b=z
        phu=[]
        tenphu=str(listthucethe[a][0])+'_'+str(listthucethe[b][0])
        phu.append(tenphu)
        phu.append(mohinhquanhenn(listthucethe[a], listthucethe[b]))
        phu=list(np.hstack(phu))
        listthucethe.append(phu)
        bang1n1='1-n:'+listthucethe[a][0]+'-'+phu[0]
        quanhebang.append(bang1n1)

```

```

        bang1n2='1-n:'+listthucthe[b][0]+'-'+phu[0]
        quanhebang.append(bang1n2)
for i in range(0,len(listthucthe)):
    if(timdatri(listthucthe[i])!=None):
        bangdatri=[]
        bangdatri.append('Table_'+str(timdatri(listthucthe[i])))
        bangdatri.append('PK.FK_'+str(timkhoachinh(listthucthe[i])))
        bangdatri.append('PK_'+str(timdatri(listthucthe[i])))
        bangdatri=np.hstack(bangdatri)
        listthucthe.append(bangdatri)
for i in range(0,len(listthucthe)):
    listthucthe[i]=convertstr(listthucthe[i])
wf = open('output1.txt', 'w')
wf.write(str(listthucthe))
wf.writelines('\n')
wf.writelines(str(quanhebang))
f.close()
wf.close()

```

Bài 2 gồm các file : Bai2.py, input2.txt, output2.txt

```
# -*- coding: utf-8 -*-
from asyncio import write
from operator import index
import numpy as np
import re
import itertools
import collections
#Đọc file
f = open('input2.txt', 'r')
qa=f.readlines()
qa1=qa[1]
F=list(qa1.split(';'))

# X=input('Thuộc tính cần tìm bao đóng: (Nhập theo định dạng
A,B,C,D,...)\n')
X = 'A,B'

le=[] #biến xuất hiện bên trái
ri=[] #biến xuất hiện bên phải
for i in F:
    le.append(i.split('->')[0])
    ri.append(i.split('->')[1])

# F là list của các phụ thuộc hàm
# X là thuộc tính người dùng nhập
def Closure(F,X):
    result=[]

    # Chuyển xâu X thành list X tên input
    input=" ".join(X)
    input=input.split(',')

    # Bao đóng của các thuộc tính đầu vào là chính chúng
    for i in input:
        result.append(i)

    check=0
    while check!=len(F)*2:
        for i in range(0,len(le)): # duyệt từng thuộc tính trong vế trái
            if(all(item in result for item in le[i].split(','))): # nếu
chúng có trong tập bao đóng
```

```

        for i in ri[i]: # thêm thuộc tính bên phải tương ứng vào
tập bao đóng
            if(i!=' , '):
                result.append(i)
            check+=1

    return(set(result)) # trả về kết quả kiểu set để loại bỏ phần tử trùng

print(Closure(F,list(X.split(' '))))

# TÌM BAO ĐÓNG

# List chứa toàn bộ thuộc tính
listUniqueInput = [i for i in qa[0] if i.isalpha()]

# TÌM N VÀ L
N=[]
for i in listUniqueInput:
    if i not in ''.join(ri).replace(',',''):
        N.append(i)
Lbe=[]
Lbe=Lbe+N
for i in listUniqueInput:
    if(i not in N and i not in ri):
        Lbe.append(i)

Laf=[]
for i in listUniqueInput:
    if(i not in Lbe):
        Laf.append(i)
keys = []
tempLaf=Laf
table=list(itertools.product({0,1},repeat=len(Laf)))
for i in range(0,len(table)):
    Laf=table[i]
    X=[]
    key=[]
    for i in range(0,len(Laf)):
        if(Laf[i]!=0):
            X.append(tempLaf[i])
            X=X+N
    X=set(X)
    Xaf=list(','.join(list(X)).split(' '))

```

```

    # Nếu bao đóng của Xaf chứa các phần tử của listUniqueInput thì Xaf là
khóa
    if (all(item in listUniqueInput for item in List(Closure(F, Xaf)))):
        keys.extend(Xaf)
# print(keys)
# Loại bỏ superkey

temp = [] # list chứa các tập cha của candidate keys
for i in range(len(keys)): # duyệt từng khóa
    for j in range(i+1, len(keys)): # và xét với các khóa còn lại
        if (all(item in keys[j] for item in keys[i])): # nếu khóa keys[i] là
con của keys[j]
            temp.append(keys[j]) # thêm khóa keys[j] vào temp
# Tìm các phần tử chỉ có trong candidateKeys mà không có trong temp
candidateKeys = [item for item in keys if item not in temp]

print("Tất cả các khóa của lược đồ quan hệ là:")
print(candidateKeys)
# GHI FILE
file = open("output2.txt", "a")
for i in candidateKeys:
    key = "".join(i.split(','))
    file.write(key + "\n")
file.close()

```