Họ và Tên : Trương Thái Đan Huy

MSSV : 52100222

ASSIGNMENT

**Phần 1:**

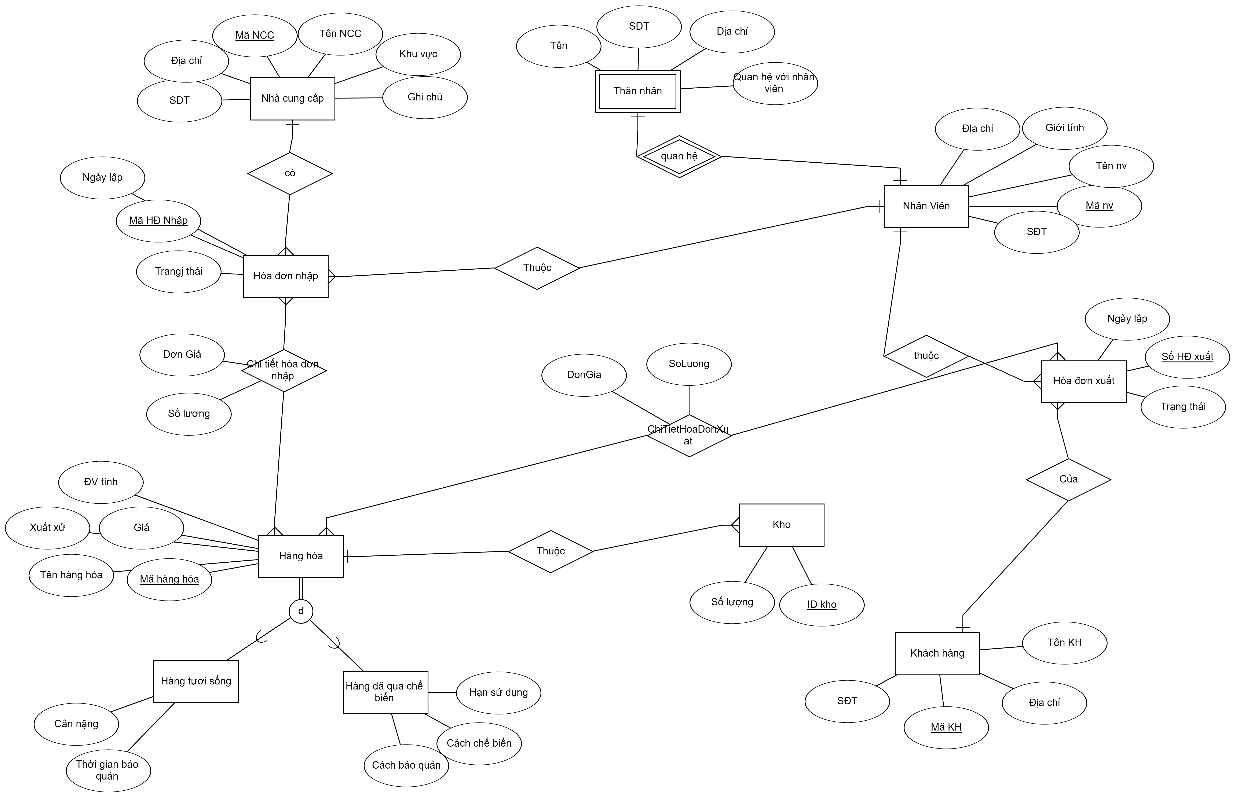
Đặc tả :

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 :

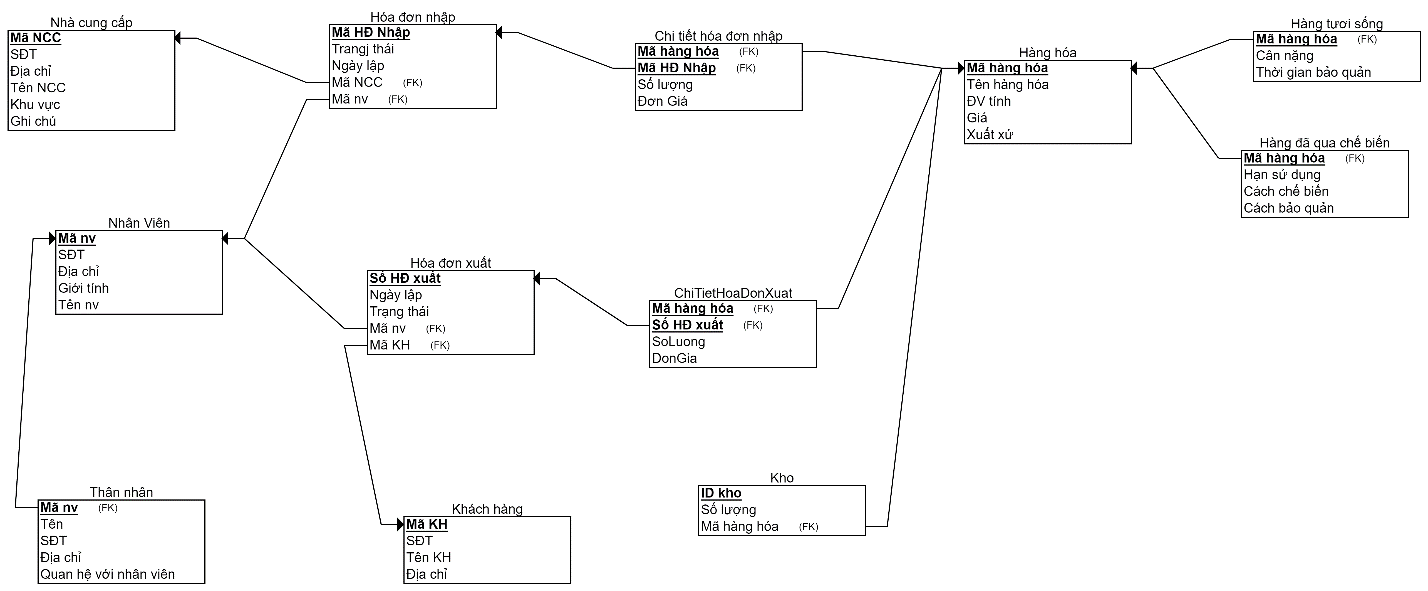
* Nhà cung cấp cần bán các loại hàng hóa có thông tin như nhau :Mã hàng,Tên hàng hóa, xuất xứ, giá bán, đơn vị tính. Ngoài các thuộc tính chung thì hàng hóa chia làm hai loại Hàng tươi sống :Cân nặng, Thời gian bảo quản. Hàng qua chế biến:Hạn sử dụng, cách sử dụng, cách bảo quản.
* Nhà cung cấp không 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, gới tính, năm sinh, dia chi, SDT.
* Mỗi nhân viên sẽ ghi thêm cho mình một thân nhân gồm các thông tin : Tên, SDT, dia chi, 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 rát nhiều sản phẩm, khi nhập cần có số lượng và đơn giá của hàng hóa đó.
* Các khách hàng khi sủ dụng dịch vụ bên nhà cung cấp cần để lại thông tin cá nhân như : Mã khách hàng, tên, SĐT, dia chi.
* 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 só 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, SDT, ghi chú, Khu vực, Tên nhà cung cấp.

Yêu cầu :

1. Các quan hệ:
2. Quan hệ 1-n: Nhà cung cấp – hóa đơn nhập, Nhân viên – Hóa đơn xuất, Khách hàng – HD xuất
3. Quan hệ n – n : Hóa đơn nhập – hàng hóa, Hóa đơn xuất – Hàng Hóa
4. Quan hệ 1 – 1: Nhân viên – Thân Nhân
5. Quan hệ cha con : Hàng Hóa – Hàng tươi sống, Hàng qua chế biến
6. Quan hệ giữa thực thể mạnh và yếu : Nhân Viên – Thân Nhân
7. Vẽ erd



1. Chuyển sang mô hình quan hệ :



1. Thực hiện lệnh SQL :

USE master

go

if exists (select name from sys.databases where name = 'QLCH')

drop database QLCH

go

create database QLCH

go

use QLCH

go

create table NCC(

MA\_NCC varchar(10) primary key,

SDT varchar(15),

DIACHI varchar(30),

KHUVUC varchar(20),

GHICHU varchar(30)

)

go

CREATE FUNCTION AUTO\_IDNV()

RETURNS VARCHAR(5)

AS

BEGIN

DECLARE @ID VARCHAR(5)

IF (SELECT COUNT(MA\_NV) FROM NHANVIEN) = 0

SET @ID = '0'

ELSE

SELECT @ID = MAX(RIGHT(MA\_NV, 3)) FROM NHANVIEN

SELECT @ID = CASE

WHEN @ID >= 0 and @ID < 9 THEN 'NV00' + CONVERT(CHAR, CONVERT(INT, @ID) + 1)

WHEN @ID >= 9 THEN 'NV0' + CONVERT(CHAR, CONVERT(INT, @ID) + 1)

END

RETURN @ID

END

go

Create table NHANVIEN(

MA\_NV varchar(10) primary key constraint IDNV default DBO.AUTO\_IDNV() ,

SDT varchar(15),

DIACHI varchar(30),

GIOITINH int,

constraint check\_GIOITINH check (GIOITINH = 1 or GIOITINH = 0),

TEN\_NV nvarchar(30)

)

go

Create table THANNHAN(

TEN\_TN nvarchar(30),

SDT varchar(15),

DIACHI varchar(30),

QUANHE varchar(20),

MA\_NV varchar(10),

constraint FK\_THANNHAN foreign key (MA\_NV) references NHANVIEN(MA\_NV)

)

go

create table HANGHOA(

MA\_HANGHOA varchar(10) primary key,

TEN\_HANG nvarchar(30),

DV\_TINH varchar(10),

GIA int check(GIA > 0),

XUATXU varchar(30)

)

go

create table HANGTUOISONG(

MA\_HANGHOA varchar(10) foreign key (MA\_HANGHOA) references HANGHOA(MA\_HANGHOA),

CANNANG float,

TG\_BAOQUAN varchar(10),

)

go

create table HANGCHEBIEN(

MA\_HANGHOA varchar(10) foreign key (MA\_HANGHOA) references HANGHOA(MA\_HANGHOA),

HANSUDUNG varchar(10),

CACHCHEBIEN nvarchar(50),

CACHBAOQUAN nvarchar(50),

)

go

create table HDNHAP(

MA\_HDNHAP varchar(10) primary key,

TRANGTHAI int,

constraint check\_TT check (TRANGTHAI =1 or TRANGTHAI = 0),

NGAYLAP smalldatetime,

MA\_NCC varchar(10),

MA\_NV varchar(10),

constraint FK\_MANV foreign key (MA\_NV) references NHANVIEN(MA\_NV),

constraint FK\_MANCC foreign key (MA\_NCC) references NCC(MA\_NCC)

)

go

create table ChiTietHoaDonNhap(

ID\_NHAP varchar(10) primary key,

SOLUONG int,

DONGIA int,

MA\_HDNHAP varchar(10) foreign key (MA\_HDNHAP) references HDNHAP(MA\_HDNHAP),

MA\_HANGHOA varchar(10) foreign key (MA\_HANGHOA) references HANGHOA(MA\_HANGHOA)

)

go

create table KHO(

ID\_KHO varchar(10) primary key,

SOLUONG int,

MA\_HANGHOA varchar(10) foreign key (MA\_HANGHOA) references HANGHOA(MA\_HANGHOA)

)

go

CREATE FUNCTION AUTO\_IDKH()

RETURNS VARCHAR(5)

AS

BEGIN

DECLARE @ID VARCHAR(5)

IF (SELECT COUNT(MA\_KH) FROM KHACHHANG) = 0

SET @ID = '0'

ELSE

SELECT @ID = MAX(RIGHT(MA\_KH, 3)) FROM KHACHHANG

SELECT @ID = CASE

WHEN @ID >= 0 and @ID < 9 THEN 'KH00' + CONVERT(CHAR, CONVERT(INT, @ID) + 1)

WHEN @ID >= 9 THEN 'KH0' + CONVERT(CHAR, CONVERT(INT, @ID) + 1)

END

RETURN @ID

END

go

create table KHACHHANG(

MA\_KH varchar(10) primary key constraint IDKH default DBO.AUTO\_IDKH(),

SDT varchar(15),

TEN\_KH varchar(30),

DIACHI varchar(30)

)

go

create table HDXUAT(

MA\_HDXUAT varchar(10) primary key,

TRANGTHAI int,

constraint check\_TT\_xuat check (TRANGTHAI =1 or TRANGTHAI = 0),

NGAYLAP smalldatetime,

MA\_NV varchar(10) foreign key (MA\_NV) references NHANVIEN(MA\_NV),

MA\_KH varchar(10) foreign key (MA\_KH) references KHACHHANG(MA\_KH)

)

go

create table ChiTietHoaDonXuat(

SOLUONG int,

MA\_HANGHOA varchar(10) foreign key (MA\_HANGHOA) references HANGHOA(MA\_HANGHOA),

MA\_HDXUAT varchar(10) foreign key (MA\_HDXUAT) references HDXUAT(MA\_HDXUAT),

DONGIA int

)

go

create proc ADDNHANVIEN

@SDT varchar(15),

@DIACHI varchar(30),

@GIOITINH int,

@TEN\_NV varchar(30)

as

insert into NHANVIEN(SDT,DIACHI,GIOITINH,TEN\_NV) values(@SDT,@DIACHI,@GIOITINH,@TEN\_NV)

go

create proc ADDKHACHHANG

@SDT varchar(15),

@TEN varchar(30),

@DIACHI varchar(30)

as

insert into KHACHHANG(SDT,TEN\_KH,DIACHI) values(@SDT,@TEN,@DIACHI)

go

CREATE TRIGGER trg\_NhapHang ON ChiTietHoaDonNhap AFTER INSERT AS

BEGIN

UPDATE KHO

SET SOLUONG = KHO.SOLUONG + (

SELECT SOLUONG

FROM inserted

WHERE MA\_HANGHOA = KHO.MA\_HANGHOA

)

FROM KHO

JOIN inserted ON KHO.MA\_HANGHOA = inserted.MA\_HANGHOA

END

GO

/\* cập nhật hàng trong kho sau khi đặt hàng hoặc cập nhật \*/

CREATE TRIGGER trg\_DatHang ON ChiTietHoaDonXuat AFTER INSERT AS

BEGIN

UPDATE KHO

SET SOLUONG = KHO.SOLUONG - (

SELECT SOLUONG

FROM inserted

WHERE MA\_HANGHOA = KHO.MA\_HANGHOA

)

FROM KHO

JOIN inserted ON KHO.MA\_HANGHOA = inserted.MA\_HANGHOA

END

GO

/\* cập nhật hàng trong kho sau khi cập nhật đặt hàng \*/

CREATE TRIGGER trg\_CapNhatDatHang on ChiTietHoaDonXuat after update AS

BEGIN

UPDATE KHO SET SOLUONG = KHO.SOLUONG -

(SELECT SOLUONG FROM inserted WHERE MA\_HANGHOA = KHO.MA\_HANGHOA) +

(SELECT SOLUONG FROM deleted WHERE MA\_HANGHOA = KHO.MA\_HANGHOA)

FROM KHO

JOIN deleted ON KHO.MA\_HANGHOA = deleted.MA\_HANGHOA

end

GO

/\* cập nhật hàng trong kho sau khi hủy đặt hàng \*/

create TRIGGER trg\_HuyDatHang ON ChiTietHoaDonXuat FOR DELETE AS

BEGIN

UPDATE KHO

SET SOLUONG = KHO.SOLUONG + (SELECT SOLUONG FROM deleted WHERE MA\_HANGHOA = KHO.MA\_HANGHOA)

FROM KHO

JOIN deleted ON KHO.MA\_HANGHOA = deleted.MA\_HANGHOA

END

go

exec ADDKHACHHANG '909111056','ME DAN HUY','CAO LANH DONG THAP'

exec ADDKHACHHANG '915725805','BA DAN HUY','CAO LANH DONG THAP'

exec ADDKHACHHANG '939726205','DAN HUY','HCM'

exec ADDNHANVIEN '0939726205','60/56 LVB',1,'Truong Thai Dan Huy'

exec ADDNHANVIEN '0939777111','60/78/4',0,'Nguyen Cat Tuong'

exec ADDNHANVIEN '0903572602','1150 HTP',1,'Pham Phuc Xuyen'

insert into NCC values

('N001','0939111xxx','LVB HCM','HCM','anh nay vip'),

('N002','0939222xxx','NLB HCM','HCM','anh nay cung vip luon'),

('N003','0939333xxx','CL Dong Thap','Dong Thap','anh nay la developer');

go

insert into THANNHAN values

('Tanh Tu','0903227504','Q8','Chong Phuc Xuyen','NV003'),

('Cat Tuong','0907777125','Q7','Vo Dan Huy','NV001')

insert into HANGHOA values

('HH001','Rau muong','Bo','5000','Nong Trai'),

('HH002','Carot','Kg','10000','Nong Trai'),

('HH003','Thit ba roi ','Kg','30000','Nong Trai'),

('HH004','Hat Nem','Goi','5000','Sieu Thi'),

('HH005','Mi Goi','Goi','4000','Sieu Thi')

insert into HANGTUOISONG values

('HH001','10','5 ngay'),

('HH002','10','1 tuan'),

('HH003','20','2 tuan')

insert into HANGCHEBIEN values

('HH004','6 Thang','Dung ngay','kho rao'),

('HH005','6 thang','Dung Ngay','kho rao')

insert into HDNHAP values

('HD001','0','5/1/2022','N001','NV001'),

('HD002','0','6/2/2022','N002','NV002'),

('HD003','1','7/3/2022','N003','NV003'),

('HD004','1','8/4/2022','N001','NV001'),

('HD005','1','9/5/2022','N002','NV002')

insert into KHO values

('KHO001',0,'HH001'),

('KHO002',10,'HH002'),

('KHO003',2,'HH003'),

('KHO004',4,'HH004'),

('KHO005',5,'HH005')

insert into ChiTietHoaDonNhap values

('IDN001',20,10000,'HD001','HH001')

insert into HDXUAT values

('IDX001',1,'6/22/2022','NV003','KH001')

insert into ChiTietHoaDonXuat values

(2,'HH002','IDX001',5000)

select \* from NCC

select \* from NHANVIEN

select \* from THANNHAN

select \* from HDNHAP

select \* from ChiTietHoaDonNhap

select \* from HANGHOA

select \* from HANGTUOISONG

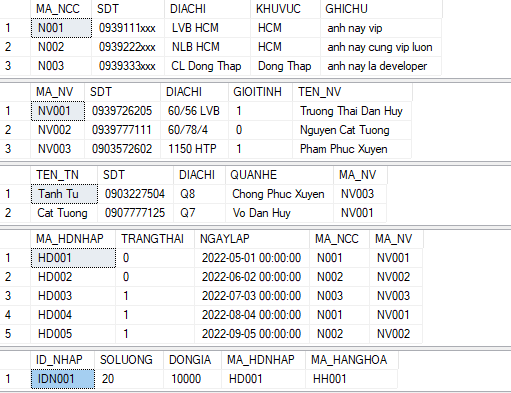
select \* from HANGCHEBIEN

select \* from KHO

select \* from HDXUAT

select \* from ChiTietHoaDonXuat

select \* from KHACHHANG

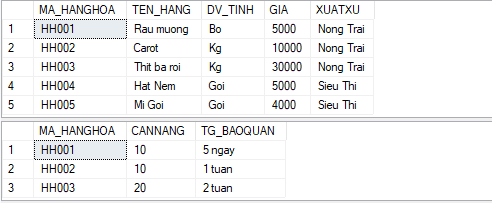
Nhà cung cấp

Nhân Viên

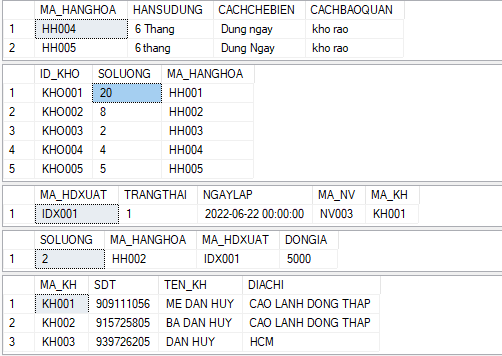
Thân Nhân

HD Nhập

ChiTietHoaDonNhap

 Hàng Hóa

Hàng Tươi sống

Hàng qua chế biến

Kho

Hóa đơn xuất

Chi tiết hóa đơn xuất

Khách hàng

**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* mohinhquanhe11(*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* timdatri(*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)):

    listthucthe.append(tachthuoctinh(thucthe[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(listthucthe)):

           if(y[0]==listthucthe[z][0]):

              a=z

           if(y[1]==listthucthe[z][0]):

              b=z

        mohinhquanhe11(listthucthe[a], listthucthe[b])

        bang11='1-1:'+listthucthe[a][0]+'-'+listthucthe[b][0]

        quanhebang.append(bang11)

    if(x[0]=='1' and x[1]=='n'):

        for z in range(0,len(listthucthe)):

           if(y[0]==listthucthe[z][0]):

              a=z

           if(y[1]==listthucthe[z][0]):

              b=z

        mohinhquanhe1n(listthucthe[a], listthucthe[b])

        bang1n='1-n:'+listthucthe[a][0]+'-'+listthucthe[b][0]

        quanhebang.append(bang1n)

    if(x[0]=='cha' and x[1]=='con'):

        for z in range(0,len(listthucthe)):

           if(y[0]==listthucthe[z][0]):

              a=z

           if(y[1]==listthucthe[z][0]):

              b=z

        mohinhquanhechacon(listthucthe[a], listthucthe[b])

        bang1n='cha-con:'+listthucthe[a][0]+'-'+listthucthe[b][0]

        quanhebang.append(bang1n)

    if(x[0]=='n' and x[1]=='n'):

        for z in range(0,len(listthucthe)):

           if(y[0]==listthucthe[z][0]):

              a=z

           if(y[1]==listthucthe[z][0]):

              b=z

        phu=[]

        tenphu=*str*(listthucthe[a][0])+'\_'+*str*(listthucthe[b][0])

        phu.append(tenphu)

        phu.append(mohinhquanhenn(listthucthe[a], listthucthe[b]))

        phu=*list*(np.hstack(phu))

        listthucthe.append(phu)

        bang1n1='1-n:'+listthucthe[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()

**Input đầu vào :**

NCC:PK.MaNCC,SDT,TenNCC,KhuVuc,GhiChu;HDNhap:NgayLap,PK.MaHDNhap,TrangThai;HangHoa:DVTinh,Gia,XuatXu,TenHang,PK.MaHang;Kho:PK.IDKho,SoLuong;NhanVien:PK.MaNV,DiaChi,Ten,SDT,GioiTinh;ThanNhan:Ten,DiaChi,QuanHe;HDXuat:NgayLap,PK.MaHDXuat,TrangThai,SoLuong;KhachHang:PK.MaKH,Ten,DiaChi,SDT

1-n:NCC-HDNhap;1-n:KhachHang-HDXuat;1-n:HangHoa-Kho;1-n:NhanVien-HDNhap;1-1:NhanVien-ThanNhan;n-n:HDNhap-HangHoa

**Output đầu ra :**

['NCC:PK.MaNCC,SDT,TenNCC,KhuVuc,GhiChu', 'HDNhap:NgayLap,PK.MaHDNhap,TrangThai,FK.MaNCC,FK.MaNV', 'HangHoa:DVTinh,Gia,XuatXu,TenHang,PK.MaHang', 'Kho:PK.IDKho,SoLuong,FK.MaHang', 'NhanVien:PK.MaNV,DiaChi,Ten,SDT,GioiTinh', 'ThanNhan:Ten,DiaChi,QuanHe,FK.MaNV', 'HDXuat:NgayLap,PK.MaHDXuat,TrangThai,SoLuong,FK.MaKH', 'KhachHang:PK.MaKH,Ten,DiaChi,SDT', 'HDNhap\_HangHoa:PK.FK.MaHDNhap,PK.FK.MaHang']

['1-n:NCC-HDNhap', '1-n:KhachHang-HDXuat', '1-n:HangHoa-Kho', '1-n:NhanVien-HDNhap', '1-1:NhanVien-ThanNhan', '1-n:HDNhap-HDNhap\_HangHoa', '1-n:HangHoa-HDNhap\_HangHoa']

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

# -\*- coding: utf-8 -\*-

from asyncore 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 xuât 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=[]

    # Chuyễn 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()

**Input đầu vào :**

ABCDEFG

A->B;AD->F;G->D;F->C;CG->

**Output đầu ra :**

**{'B', 'A'}**

**Tất cả các khóa của lược đồ quan hệ là:**

**['E,F,G,A', 'E,A,G,D', 'E,C,G,A', 'E,B,G,A']**

AFG

EGA

AGD

AGC

ABG

FGA

EGA

GDA

CGA

AGB

EAFG

EAGD

EAGC

EBAG

Link code và các thông tin liên quan :

<https://drive.google.com/drive/u/3/folders/1SVKXks-dkC77BeDhY2C2YjZynQMXPHeb>