TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**VÕ NHẬT HÀO – 522H0090**

**BÀI TẬP LỚN**

**HỆ CƠ SỞ DỮ LIỆU**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**VÕ NHẬT HÀO – 522H0090**

**BÀI TẬP LỚN**

**HỆ CƠ SỞ DỮ LIỆU**

Người hướng dẫn

**TS. Trần Thanh Phước**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

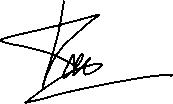
**LỜI CẢM ƠN**

Em xin chân thành cảm ơn thầy Trần Thanh Phước đã đồng hành cùng em. Cảm ơn thầy đã dạy và hướng dẫn cho em hoàn thành bài tiểu luận giữa kì môn Hệ Cơ Sở Diệu. Cảm ơn khoa Công Nghệ Thông Tin đã cho em tiếp xúc bài tiểu luận này và em tin chắc rằng bài tiểu luận này sẽ cho em tiếp cận với những kĩ năng làm việc về các bài báo cáo và kĩ năng trình bày nhằm tiếp cận cho công việc, khi đi làm có thể thích nghi với môi trường mới hơn. Ngoài ra, em chưa được tiếp cận đến nhiều bài tập về tiểu luận nên có thể còn nhiều lỗi và một số sai sót. Hy vọng thầy/cô chấm điểm và góp ý để em có thể càng phát triển, tốt hơn trên lĩnh vực này. Lời cuối cùng, em xin cảm ơn thầy/cô đã đọc và chấm điểm, đánh giá. Chúc cho thầy/cô thật nhiều sức khỏe để có thể truyền tải những kiến thức mới cho em và các bạn sinh viên.

*TP. Hồ Chí Minh, ngày 21 tháng 11 năm 2023.*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*



*Võ Nhật Hào*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Trần Thanh Phước. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

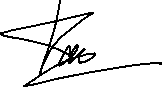
Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 21 tháng 11 năm 2023.*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*



*Võ Nhật Hào*

**TÓM TẮT**

(Time New Romans – 13)

**MỤC LỤC**

[PHẦN 1. 1](#_Toc151587502)

[1.1 Đặc tả 1](#_Toc151587503)

[1.2 Mô hình ERD 1](#_Toc151587504)

[1.3 Mô hình quan hệ 2](#_Toc151587506)

[1.4 Các câu lệnh SQL 3](#_Toc151587507)

[1.4.1 Tạo bảng 3](#_Toc151587508)

[1.4.2 Tạo fuction 5](#_Toc151587509)

[1.4.3 Tạo trigger 7](#_Toc151587510)

[PHẦN 2. 8](#_Toc151587511)

[2.1 Câu 1 8](#_Toc151587512)

[2.2 Câu 2 8](#_Toc151587513)

[TÀI LIỆU THAM KHẢO 9](#_Toc151587514)

# PHẦN 1.

## Đặc tả

* Quản lý dự án của một công ty về phần mềm:
* Nhà đầu tư muốn đầu tư cho một công ty phần mềm. Thông tin của nhà đầu tư bao gồm: mã nhà đầu tư, tên nhà đầu tư và số vốn đầu tư. Nhà đầu tư có thể đầu tư nhiều dự án. Thông tin của dự án bao gồm: mã dự án, tên dự án, tổng số vốn của nhà đầu tư.
* Mỗi dự án công ty có thể phân công nhiều nhân viên. Thông tin của nhân viên gồm: mã nhân viên, tên nhân viên, địa chỉ và số điện thoại nhân viên. Nhiều nhân viên thuộc một phòng ban. Thông tin của phòng ban gồm: mã phòng ban, tên phòng ban, email và số điện thoại phòng ban.
* Nhân viên trong công ty bao gồm 2 nhóm chính là: Lập trình viên và Designer. Thông tin Lập trình viên bao gồm chuyên môn và email. Thông tin Designer bao gồm kinh nghiệm và ngày sinh.
* Mỗi lập trình viên công ty sẽ được công ty cung cấp một tài khoản duy nhất. Thông tin tài khoản gồm: tên tài khoản và mật khẩu.
* Một Designer có thể có nhiều người thân. Thông tin về người thân bao gồm: tên người thân và số điện thoại

## Mô hình ERD

## Ảnh có chứa biểu đồ, hình vẽ, bản phác thảo, màu trắng Mô tả được tạo tự động

## Mô hình quan hệ

Ảnh có chứa biểu đồ, Bản vẽ kỹ thuật, Kế hoạch, sơ đồ

Mô tả được tạo tự động

## Các câu lệnh SQL

### Tạo bảng

Create table NhaDauTu

(

maNDT varchar(10) not null,

tenNDT nvarchar(30) not null,

SoVon varchar(30) not null,

primary key (maNDT)

);

Create table DuAn

(

maDA varchar(10) not null,

tenDA nvarchar(30) not null,

TongVon varchar(30) not null,

primary key (maDA)

);

Create table TaiKhoan

(

tenTK nvarchar(30) not null,

MatKhau varchar(30) not null,

primary key (tenTK)

);

Create table PhongBan

(

maPB varchar(10) not null,

tenPB nvarchar(30) not null,

emailPB varchar(30) not null,

sdtPB varchar(30) not null,

primary key (maPB)

);

Create table DauTu

(

maNDT varchar(10) not null,

maDA varchar(10) not null,

primary key (maNDT, maDA),

foreign key (maNDT) references NhaDauTu(maNDT),

foreign key (maDA) references DuAn(maDA)

);

Create table NhanVien

(

maNV varchar(10) not null,

tenNV nvarchar(30) not null,

sdtNV varchar(30) not null,

DiaChi varchar(30) not null,

maDA varchar(10) not null,

maPB varchar(10) not null,

primary key (maNV),

foreign key (maDA) references DuAn(maDA),

foreign key (maPB) references PhongBan(maPB)

);

Create table LapTrinhVien

(

emailLTV varchar(30) not null,

ChuyenMon varchar(30) not null,

maNV varchar(10) not null,

tenTK nvarchar(30) not null,

primary key (maNV),

foreign key (maNV) references NhanVien(maNV),

foreign key (tenTK) references TaiKhoan(tenTK)

);

Create table Designer

(

KinhNghiem nvarchar(30) not null,

NgaySinh date not null,

maNV varchar(10) not null,

primary key (maNV),

foreign key (maNV) references NhanVien(maNV)

);

Create table NguoiThan

(

tenNT nvarchar(30) not null,

sdtNT varchar(30) not null,

maNV varchar(10) not null,

primary key (maNV),

foreign key (maNV) references Designer(maNV)

);

### Tạo fuction

-- Tạo mã Nhà đầu tư tự động

--drop function TaoMaNDT

go

Create function TaoMaNDT()

returns varchar(10)

as

begin

if (select top 1 maNDT from NhaDauTu) is null

return 'NDT0001'

declare @vt int

select top 1 @vt = convert(int, right(maNDT, 4)) from NhaDauTu

order by maNDT desc

set @vt = @vt + 1

declare @MaNDT varchar(10)

set @MaNDT = 'NDT'

declare @s int

set @s = len(convert(varchar(4), @vt))

while @s < 4

begin

set @MaNDT = @MaNDT + '0'

set @s = @s + 1

end

set @MaNDT = @MaNDT + convert(varchar(4), @vt)

return (@MaNDT)

end

--drop proc P\_NDT

go

Create proc P\_NDT @TenNDT nvarchar(30), @SoVon varchar(30)

as

begin

insert into NhaDauTu values (dbo.TaoMaNDT(), @TenNDT, @SoVon)

end

Exec P\_NDT N'Nhà đầu tư 4', '10000000'

Select \* from NhaDauTu

-- Tạo mã Dự Án tự động

--drop function TaoMaDA

go

Create function TaoMaDA()

returns varchar(10)

as

begin

if (select top 1 maDA from DuAn) is null

return 'DA0001'

declare @vt int

select top 1 @vt = convert(int, right(maDA, 4)) from DuAn

order by maDA desc

set @vt = @vt + 1

declare @MaDA varchar(10)

set @MaDA = 'DA'

declare @s int

set @s = len(convert(varchar(4), @vt))

while @s < 4

begin

set @MaDA = @MaDA + '0'

set @s = @s + 1

end

set @MaDA = @MaDA + convert(varchar(4), @vt)

return (@MaDA)

end

--drop proc P\_NV

go

Create proc P\_DA @TenDA nvarchar(30), @TongVon varchar(30)

as

begin

insert into DuAn values (dbo.TaoMaDA(), @TenDA, @TongVon)

end

Exec P\_DA N'Dự án 4', '55000000'

Select \* from DuAn

### Tạo trigger

--Trigger

--drop trigger T\_TaiKhoan

go

Create trigger T\_TaiKhoan on TaiKhoan

after insert, update

as

begin

declare @tenTk nvarchar(30)

declare @MK varchar(30)

select @tenTk = tenTK, @MK = MatKhau from inserted

--Kiểm tra ràng buộc

if (@tenTk is null or exists (select @tenTk from LapTrinhVien where tenTK = @tenTk))

begin

if ((select count(tenTK) from TaiKhoan where tenTK = @tenTk) = 2)

begin

print (N'Lập trình viên đã có tài khoản')

rollback tran

end

if (@MK is null)

begin

print (N'Mật khẩu không được trống')

rollback tran

end

if (len(@MK) > 30)

begin

print (N'Mật khẩu phải nhỏ hơn 30 kí tự')

rollback tran

end

if (len(@tenTk) > 30)

begin

print (N'Tên tài khoản phải nhỏ hơn 30 kí tự')

rollback tran

end

end

else

begin

print (N'Tên tài khoản không tồn tại')

rollback tran

end

end

Insert into TaiKhoan values ('user4', 'passwordusersagchseasd@1')

# PHẦN 2.

## Câu 1

* File input.txt:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

* Định dạng input như sau:
* Các thực thể mô hình ERD: Tên thực thể <khoảng cách> Tập thuộc tính <khoảng cách> Khóa chính (sau dấu “ , ”).
* Các mối quan hệ giữa các thực thể mô hình ERD: Thực thể 1-Thực thể 2 <khoảng cách> Loại quan hệ <khoảng cách> Tên mối quan hệ.
* Code python:

def **read\_lines**(file\_name):

    with **open**(file\_name, encoding='utf-8') as f:

        return f.**readlines**()

pkey\_names = []

prikeys = []

def **erd\_entities**(lines):

    tables = []

    for line in lines:

        if '-' not in line:

            parts = line.split(',')

            entity = parts[0].split(' ')[0]

            attrs = parts[0].split(' ')[1:]

            pkeys = []

            if **len**(parts) > 1:

                pkeys.**append**(parts[1].strip())

            else:

                pkeys.**extend**(attrs)

            table = f"{entity} \n"

            for attr in attrs:

                if attr in pkeys:

                    table += f'{attr}\n'

                else:

                    table += f'{attr}\n'

            if **len**(pkeys) > 1:

                refs = ' '.**join**(pkeys)

                table += f'PRIMARY KEY: {refs}'

            else:

                table += f'PRIMARY KEY: {pkeys[0]}'

            tables.**append**(table)

            prikeys.**append**(pkeys)

            pkey\_names.**append**(entity)

    return tables, prikeys, pkey\_names

def **erd\_relationship**(lines, prikeys, pkey\_names):

    relationships = []

    for line in lines:

        if '-' in line:

            parts = line.strip().split(' ')

            entities = parts[0]

            rel = parts[1]

            if rel == "Inheritance":

                newtable = " "

            else:

                newtable = parts[2]

            ent1, ent2 = entities.split('-')

            if (rel == '1-1'):

                relationships.**append**(f'{entities}: RELATIONSHIP(1-1)')

                pkey1 = prikeys[pkey\_names.index(ent1)]

                fkey1 = prikeys[pkey\_names.index(ent2)]

                table = f'{newtable}'

                table += f'\nPRIMARY KEY: {pkey1[0]}'

                table += f'\nFOREIGN KEY: {fkey1[0]}'

                relationships.**append**(table)

            elif (rel == '1-N'):

                relationships.**append**(f'{entities}: RELATIONSHIP(1-N)')

                pkey1 = prikeys[pkey\_names.index(ent1)]

                fkey1 = prikeys[pkey\_names.index(ent2)]

                table = f'{newtable}'

                table += f'\nPRIMARY KEY: {pkey1[0]}'

                table += f'\nFOREIGN KEY: {fkey1[0]}'

                relationships.**append**(table)

            elif (rel == 'N-N'):

                relationships.**append**(f'{entities}: RELATIONSHIP({rel}) Create Entity: 'f'{newtable}')

                pkey1 = prikeys[pkey\_names.index(ent1)]

                pkey2 = prikeys[pkey\_names.index(ent2)]

                table = f'{newtable}'

                table += f'\nPRIMARY KEY: {", ".**join**(pkey1 + pkey2)}'

                table += f'\nFOREIGN KEY: {", ".**join**(pkey1 + pkey2)}'

                relationships.**append**(table)

            else:

                relationships.**append**(f'{entities}: RELATIONSHIP(Inheritance)')

    return relationships

def **write\_tables**(tables, rels):

    with **open**('output.txt','w', encoding='utf-8') as f:

       f.**write**('\n\n'.**join**(tables + rels))

def **main**():

    lines = **read\_lines**('input.txt')

    entities = []

    relationships = []

    for line in lines:

       if '-' not in line:

           entities.**append**(line)

       else:

           relationships.**append**(line)

    tables, prikeys, pkey\_names = **erd\_entities**(entities)

    rels = **erd\_relationship**(relationships, prikeys, pkey\_names)

**write\_tables**(tables, rels)

**main**()

* File output.txt:

Ảnh có chứa văn bản, thực đơn, ảnh chụp màn hình, tài liệu

Mô tả được tạo tự động

Ảnh có chứa văn bản, thực đơn, tài liệu, biên lai

Mô tả được tạo tự động

Ảnh có chứa văn bản, thực đơn, tài liệu, ảnh chụp màn hình

Mô tả được tạo tự động

* Định dạng output như sau:
* Các thực thể mô hình ERD: Tên thực thể <xuống dòng> Tập thuộc tính <xuống dòng> PRIMARY KEY:
* Các mối quan hệ giữa các thực thể mô hình ERD: Thực thể 1-Thực thể 2 <khoảng cách> Loại quan hệ <khoảng cách> Tên mối quan hệ.
* Đối với mối quan hệ N-N thì sẽ tạo ra bảng mới:
* Tên các mối quan hệ <xuống dòng> PRIMARY KEY: <xuống dòng> FOREIGN KEY:

## Câu 2

* Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

  Mô tả được tạo tự độngFile input.txt:
* Định dạng input như sau:
* Thuộc tính ban đầu .
* Các phụ thuộc hàm: về phải phụ thuộc -> vế trái
* Bao đóng cần tìm.
* Code python:

from **itertools** import **chain**, **combinations**

def **find\_closure**(attributes, dependencies):

    closure = **set**(attributes)

    old\_closure = **set**()

    while closure != old\_closure:

        old\_closure = **set**(closure)

        for dependency in dependencies:

            if **set**(dependency[0]).**issubset**(closure):

                closure.**update**(dependency[1])

    return **sorted**(**list**(closure))

from **itertools** import **chain**, **combinations**

def **find\_keys**(attributes, dependencies):

    keys = []

    def **is\_superkey**(subset):

        closure = **find\_closure**(subset, dependencies)

        return **set**(attributes) == **set**(closure)

    for i in **range**(1, **len**(attributes) + 1):

        all\_subsets = **combinations**(attributes, i)

        for subset **in** all\_subsets:

            if **is\_superkey**(subset):

                is\_minimal = True

                for existing\_key in keys:

                    if **set**(subset).**issuperset**(existing\_key):

                        is\_minimal = False

                        break

                if is\_minimal:

                    keys.**append**(**sorted**(**list**(subset)))

    return keys

def **powerset**(iterable):

    s = **list**(iterable)

    return **list**(**chain**.**from\_iterable**(**combinations**(s, r) for r in **range**(1, **len**(s) + 1)))

def **read\_file**(file\_name):

    with **open**(file\_name, 'r', encoding='utf-8') as file:

        lines = file.**readlines**()

        attributes = lines[0].**split**('=')[1].**strip**().**replace**('{', '').**replace**('}', '').**replace**(' ', '').**split**(',')

        dependencies = [**tuple**(dep.**split**('->')) for dep in lines[1].**split**('=')[1].**strip**().**replace**('{', '').**replace**('}', '').**replace**(' ', '').**split**(',')]

        X = **set**(lines[2].**split**('=')[1].**strip**().**replace**(' ', ''))

    return attributes, dependencies, X

def **write\_file**(file\_name, closure, keys, dependencies):

    with **open**(file\_name, 'w', encoding='utf-8') as file:

        file.**write**(f'Bao Dong: {"".**join**(closure)}\n')

        keys\_str = ', '.**join**(''.**join**(key) for key in keys)

        file.**write**(f'Cac Khoa: {keys\_str}')

def **main**():

    attributes, dependencies, X = **read\_file**('input2.txt')

    closure = **find\_closure**(X, dependencies)

    keys = **find\_keys**(attributes, dependencies)

**write\_file**('output2.txt', closure, keys, dependencies)

if \_\_name\_\_ == '\_\_main\_\_':

**main**()

* File output.txt:

Ảnh có chứa văn bản, Phông chữ, đại số, thiết kế

Mô tả được tạo tự động

* Định dạng output như sau:
* Bao Dong: chứa các bao đóng của tập thuộc tính X.
* Cac Khoa: chứa các khóa của lược đồ quan hệ.

# TÀI LIỆU THAM KHẢO

* Tài liệu tham khảo cho Phần 2 câu 1:

<https://www.thuvientailieu.vn/tai-lieu/chuyen-doi-erd-thanh-luoc-do-quan-he-47014/>

* Tài liệu tham khảo cho Phần 2 câu 2:

<https://phanmemhaynhat.wordpress.com/2011/12/08/tim-bao-dong-c%E1%BB%A7a-t%E1%BA%ADp-thu%E1%BB%99c-tinh-ph%E1%BB%A7-t%E1%BB%91i-thi%E1%BB%83u/>