

**TRƯỜNG CAO ĐẲNG THỰC HÀNH FPT-POLYTECHNIC HÀ NỘI**

----------

**BÁO CÁO ASSIGNMENT - COM2034**

**QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI SQL SERVER**

Sinh viên thực hiện: Lê Thị Nga

Mã số sinh viên: PH26840

Lớp: IT17320

Nhóm: 01

GVHD: Trần Thanh Long

HÀ NỘI – SUMMER 2022

MENU

[A.GIỚI THIỆU BÀI TOÁN 2](#_Toc105430037)

[I.ĐẶT VẤN ĐỀ 2](#_Toc105430038)

[B.NỘI DUNG THỰC HIỆN 3](#_Toc105430041)

[I.THIẾT KẾ CSDL 3](#_Toc105430042)

[1.Xây dựng cơ sở dữ liệu 3](#_Toc105430043)

[2.Sơ đồ thực thể ERD 3](#_Toc105430044)

[3.Tạo Database Diagram 4](#_Toc105430045)

[4.Chi tiết thiết kế CSDL 5](#_Toc105430046)

[**Bảng LOAINHA** 5](#_Toc105430047)

[**Bảng NGUOIDUNG** 5](#_Toc105430048)

[**Bảng NHATRO** 6](#_Toc105430049)

[**Bảng DANHGIA** 6](#_Toc105430050)

[II.Tiến hành cài đặt CSDL trên SQL SERVER 6](#_Toc105430051)

[1.Tạo Database:QLNHATRO 6](#_Toc105430052)

[2.Tạo bảng và các ràng buộc khoá ngoại 7](#_Toc105430053)

[**BẢNG LOAINHA** 7](#_Toc105430054)

[**BẢNG NGUOIDUNG** 7](#_Toc105430055)

[**BẢNG NHATRO** 8](#_Toc105430056)

[**BẢNG DANHGIA** 8](#_Toc105430057)

[**Các ràng buộc khoá chính, khoá ngoại** 8](#_Toc105430058)

[3.Chèn dữ liệu vào bảng: 9](#_Toc105430059)

[**BẢNG LOAINHA: 3 bản ghi** 9](#_Toc105430060)

[**BẢNG NGUOIDUNG:10 bản ghi** 9](#_Toc105430061)

[**BẢNG NHATRO:10 bản ghi** 9](#_Toc105430062)

[**BẢNG DANHGIA:10 bản ghi** 10](#_Toc105430063)

[III. CÁC YÊU CẦU VỀ CHỨC NĂNG 10](#_Toc105430065)

[1.Thêm thông tin vào bảng: 10](#_Toc105430066)

[2.Truy vấn thông tin: 12](#_Toc105430073)

[3.Xoá thông tin: 12](#_Toc105430074)

[4.Quản trị CSDL: 12](#_Toc105430075)

[C.KẾT LUẬN 12](#_Toc105430076)

[D.BÀI HỌC KINH NGHIỆM 12](#_Toc105430077)

# **A.GIỚI THIỆU BÀI TOÁN**

# **I.ĐẶT VẤN ĐỀ**

# Ngày nay, sự phát triển không ngừng công nghệ thông tin nói chung và các lĩnh vực ngành công nghệ phần mềm nói riêng, việc tin học hóa các công tác quản lý trong nhiều lĩnh vực, hoạt động khác nhau đã tạo cho chúng ta một thư viện dữ liệu, sẵn sàng phục vụ bắt cứ ai quan tâm trong các lĩnh vực đó thì lĩnh vực quản lý là thật sự giúp ích được rất nhiều cho con người. Công việc đi tìm một nhà trọ như ý theo cách thức truyền thống là một việc rất tốn thời gian, công sức, ...Đặc biệt đối với các sinh viên mới nhập trường thì đây quả là một công việc rất phức tạp.

# Tuy nhiên, sinh viên thường có xu hướng thuê những nhà trọ có điều kiện an ninh trật tự tốt, đảm bảo vệ sinh sạch sẽ, điện nước đầy đủ, hợp lí, gần các trường đại học, cao đẳng…. Cho nên, những nhà trọ như vậy thường thu hút cầu của sinh viên cao hơn đặc biệt đối với đối tượng sinh viên có thu nhập cao, cầu của họ đối với những căn nhà có hình thức đẹp sẽ cao hơn. Để đảm bảo yêu cầu học tập của sinh viên, vấn đề mà các sinh viên và phụ huynh lo lắng nhất vẫn là trọ. Vì vậy, trọ tốt, ổn định, đảm bảo an ninh trật tự là điều kiện thuận lợi để các sinh viên học tập tốt. Những khu nhà trọ đạt tiêu chuẩn đảm bảo an ninh, văn minh phố phường thì thường thu hút sinh viên tìm đến thuê nhiều.

## 

## \_Là một sinh viên công nghệ thông tin, chúng tôi muốn xây dựng một Website nho nhỏ để giúp đỡ bạn bè và sinh viên trong trường có thể đăng tin và tìm kiếm thông tin nhà trọ cho thuê trong nội thành Hà Nội.

# **B.NỘI DUNG THỰC HIỆN**

# **I.THIẾT KẾ CSDL**

# **\_Yêu cầu: Thiết kế CSDL phải đảm bảo:** + Đầy đủ liên kết giữa các bảng

# **+**Kiểu dữ liệu và độ dài trường dữ liệu phải phù hợp, tốn ít dung lượng bộ nhớ

# và hỗ trợ Tiếng Việt với các trường cần thiết. + Có ràng buộc CHECK dữ liệu cho các trường cần thiết.

# + Thiết lập thuộc tính NULL/NOT NULL cho các cột một cách phù hợp

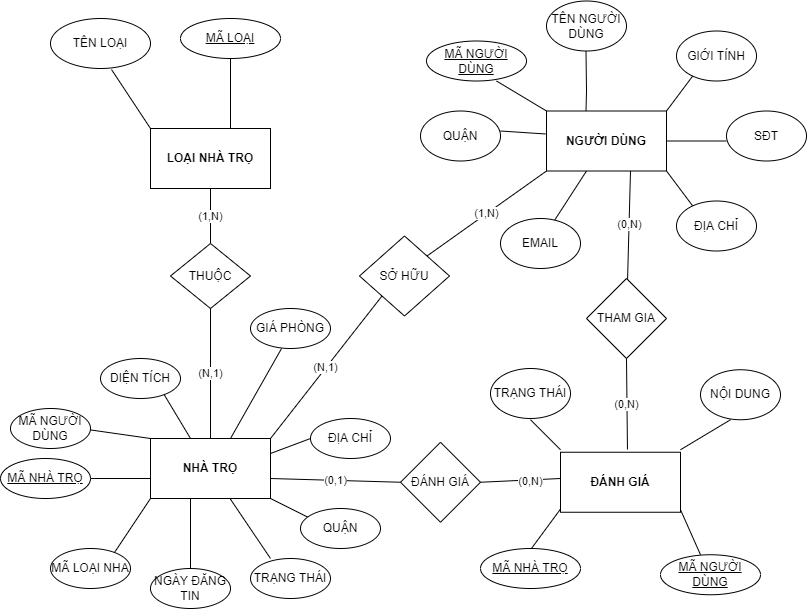
# 

## 1.Xây dựng cơ sở dữ liệu

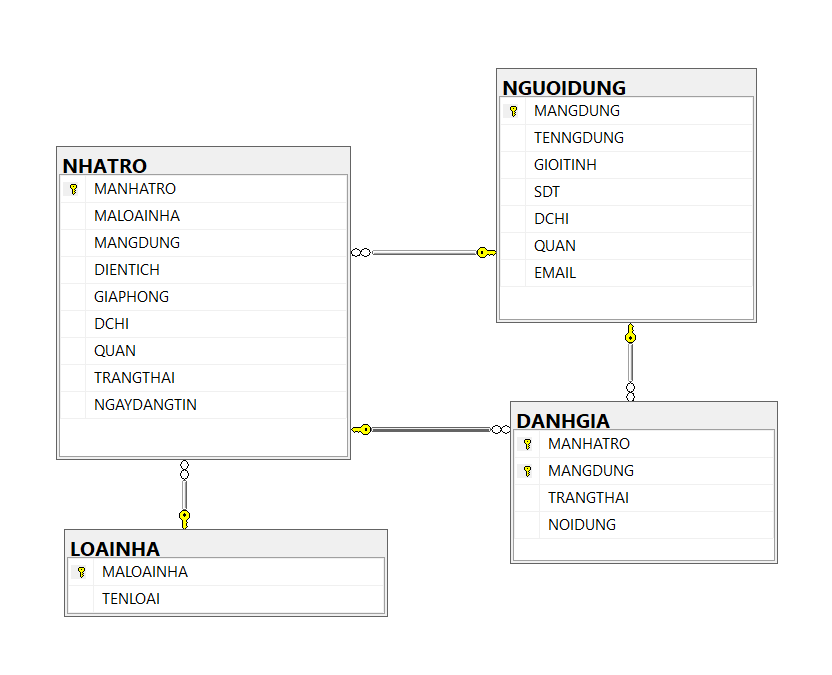
* Chủ đề:Quản lí nhà trọ
* Các đối tượng cần quản lí:

1. **LOAINHA:** lưu thông tin loại hình nhà trọ cho thuê
2. **NGUOIDUNG:** lưu toàn bộ thông tin các thành viên của Website,có thể đăng tin thuê trọ và đánh giá.
3. **NHATRO:** lưu thông tin nhà trọ cho thuê
4. **DANHGIA:** lưu thông tin đánh giá chất lượng nhà trọ

## 2.Sơ đồ thực thể ERD



## 3.Tạo Database Diagram



## 4.Chi tiết thiết kế CSDL

### **Bảng LOAINHA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **MALOAINHA** | varchar(5) | PK,not null | Mã loại nhà |
| TENLOAINHA | nvarchar(50) | null | Tên loại nhà |

### **Bảng NGUOIDUNG**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **MANGUOIDUNG** | varchar (5) | PK,not null | Mã người dùng |
| TENNGUOIDUNG | nvarchar (50) | null | Tên người dùng |
| GIOITINH | nvarchar (3) | null | Giới tính |
| SDT | varchar (13) | null, Unique | Số điện thoại |
| DCHI | nvarchar (50) | null | Địa chỉ |
| QUAN | nvarchar (50) | null | Quận |
| EMAIL | nvarchar (30) | null , Unique | Email |

### **Bảng NHATRO**

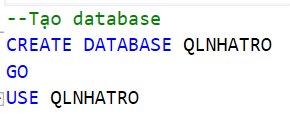
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **MANHATRO** | varchar(5) | PK,not null | Mã nhà trọ |
| **MALOAINHA** | varchar(5) | FK,not null | Mã loại nhà |
| **MANGDUNG** | varchar(5) | FK,not null | Mã người dùng |
| DIENTICH | float | Null,  Diện tích >0 | Diện tích |
| GIAPHONG | decimal(10,2) | Null,  Giá phòng>0 | Giá phòng |
| DCHI | nvarchar(30) | null | Địa chỉ |
| QUAN | nvarchar(20) | null | Quận |
| TRANGTHAI | nvarchar(20) | null | Trạng thái |
| NGAYDANGTIN | datetime | null | Ngày đăng tin |

### **Bảng DANHGIA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| **MANHATRO** | varchar(5) | PK,FK,  not null | Mã nhà trọ |
| **MANGDUNG** | varchar(5) | PK,FK,  not null | Mã người dùng |
| TRANGTHAI | nvarchar(20) | null | Trạng thái |
| NOIDUNG | nvarchar(250) | null | Nội dung |

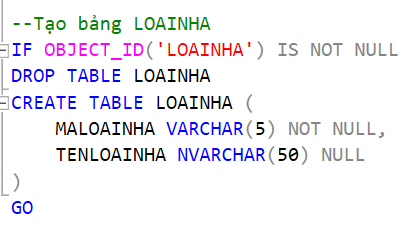
# **II.Tiến hành cài đặt CSDL trên SQL SERVER**

## 1.Tạo Database:QLNHATRO

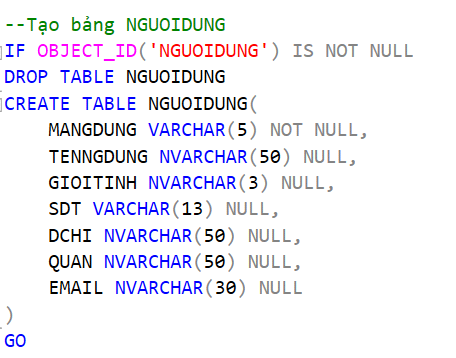


## 2.Tạo bảng và các ràng buộc khoá ngoại

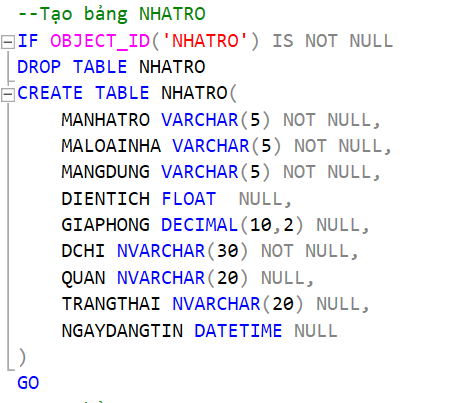
### **BẢNG LOAINHA**



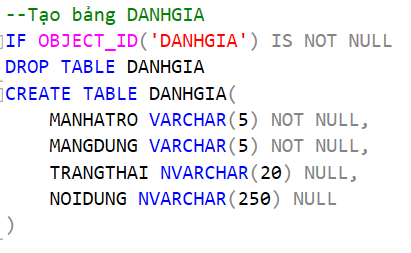
### **BẢNG NGUOIDUNG**



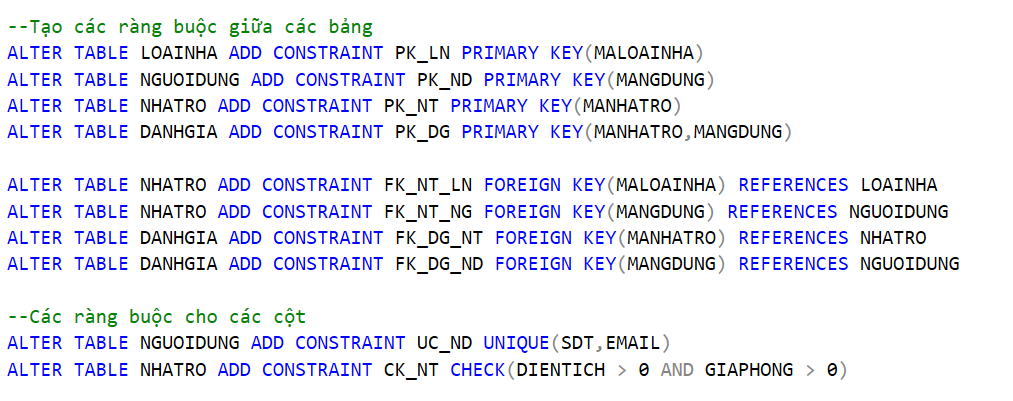
### **BẢNG NHATRO**



### **BẢNG DANHGIA**



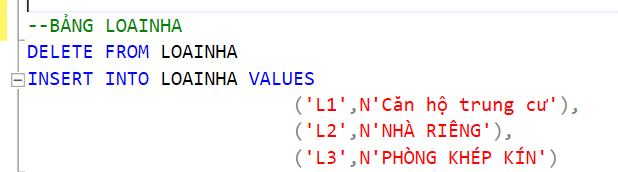
### **Các ràng buộc khoá chính, khoá ngoại**



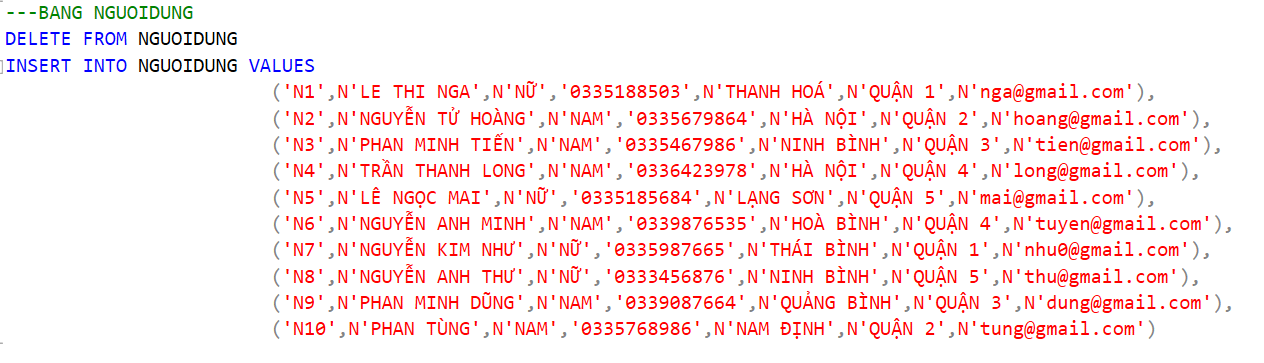
## 3.Chèn dữ liệu vào bảng:

## Lưu ý: Dữ liệu nhập vào các bảng phải có ý nghĩa thực tế, logic, đúng quy định về kiểu dữ liệu và đảm bảo có thể sử dụng để chạy thử tất cả các yêu cầu bên dưới đều trả về kết quả

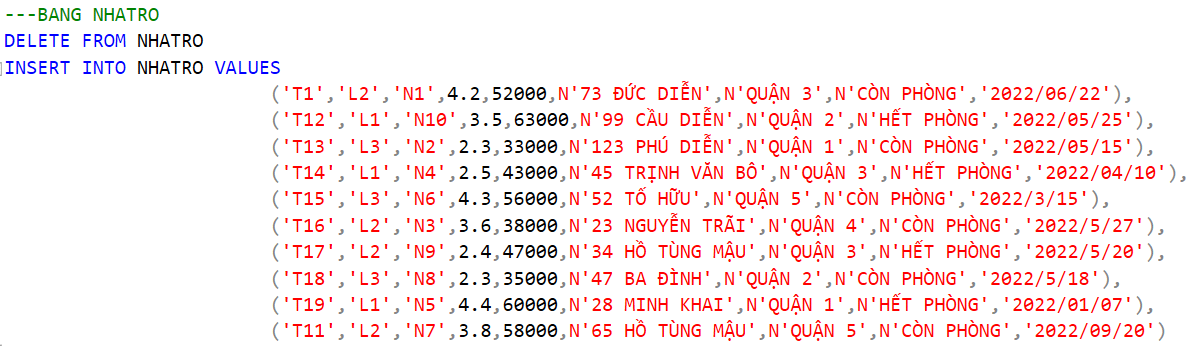
### **BẢNG LOAINHA: 3 bản ghi**



### **BẢNG NGUOIDUNG:10 bản ghi**



### **BẢNG NHATRO:10 bản ghi**



### **BẢNG DANHGIA:10 bản ghi**

# 

# **III.** **CÁC YÊU CẦU VỀ CHỨC NĂNG**

## 1.Thêm thông tin vào bảng:

# - Tạo ba Stored Procedure (SP) với các tham số đầu vào phù hợp. Yêu cầu đối với các SP: Trong mỗi SP phải kiểm tra giá trị các tham số đầu vào. Với các cột không chấp nhận thuộc tính NULL, nếu các tham số đầu vào tương ứng với chúng không được truyền giá trị, thì không thực hiện câu lệnh chèn mà in một thông báo yêu cầu người dùng nhập liệu đầy đủ. - Với mỗi SP, viết hai lời gọi. Trong đó, một lời gọi thực hiện chèn thành công dữ liệu, và một lời gọi trả về thông báo lỗi cho người dùng

# **o SP thứ nhất thực hiện chèn dữ liệu vào bảng NGUOIDUNG**

# **o SP thứ hai thực hiện chèn dữ liệu vào bảng NHATRO**

# 

# **o SP thứ ba thực hiện chèn dữ liệu vào bảng DANHGIA**

# 

## 2.Truy vấn thông tin:

## \_Truy vấn có thể thực hiện nhiều chức năng khác nhau trong cơ sở dữ liệu. Chức năng phổ biến nhất là truy xuất dữ liệu cụ thể từ các bảng. Dữ liệu mà bạn muốn xem thường trải ra nhiều bảng và truy vấn giúp bạn xem chúng trong một biểu dữ liệu duy nhất. Truy vấn cho phép bạn thêm các tiêu chí để "lọc" dữ liệu đến đúng các bản ghi mà bạn muốn.

## \_\_Tiến hành thực hiện truy vấn thông tin theo yêu cầu của đề bài

## \_Với câu a: ta thực hiện tìm kiếm thông tin các phòng trọ thoả mãn điều kiện bằng cách dùng thủ tục Store Procedure với cú pháp:

## 

## 

## \_Câu B: Ta dùng hàm Function giá trị là một bảng với cú pháp:

## 

## 

## \_Ở câu C:Dùng hàm Function với giá trị vô hướng với cú pháp:

## 

## \_Ở câu D: Sử dụng View để lưu thông tin nhà trọ với cú pháp:

## 

## \_Ở câu E: SỬ dụng thủ tục Stoce Procedure để tìm kiếm các thông tin :

## 

## 3.Xoá thông tin:

## \_Yêu cầu: Sử dụng giao dịch trong thân SP, để đảm bảo tính toàn vẹn dữ liệu khi một thao tác xóa thực hiện không thành công.

## \_Với thông tin các nhà trọ và thông tin đánh giá :thực hiện xoá ĐÁNH GIÁ 🡪 NGƯỜI DÙNG 🡪 XOÁ NHÀ TRỌ

## \_Với các thông tin đánh giá:

## 4.Quản trị CSDL:

# **C.KẾT LUẬN**

# **\_**Bài viết đã xây dựng được quy trình tạo ra một cơ sở dữ liệu từ những thông tin về Nhà Trọ cho thuê ở Hà Nội. Tất cả những thông tin được xây dựng dựa trên các yêu cầu của bài toán.

# \_Việc xây dựng cơ sở dữ liệu giúp cho các doanh nghiệp xây dựng chuyển đổi việc lưu trữ thông tin cá nhân thủ công trở thành số hóa giúp giảm bớt thời gian cũng như duy trì dữ liệu kế thừa qua từng năm,đồng thời giúp các chủ dễ quản lý Nhà Trọ hơn.

# **D.BÀI HỌC KINH NGHIỆM**

\_Người dùng phải biết được mục đích của việc sử dụng dữ liệu

\_ Mọi CSDL nên ít nhất được chuẩn hóa theo dạng 3NF vì đó là mô hình tốt nhất để thể hiện các thực thể dữ liệu và sẽ cân bằng được hiệu năng truy vấn và thêm-sửa-xóa bản ghi.

\_ Phải xác định ràng buộc khoá chính khoá ngoại giúp bảo toàn độ chính xác của dữ liệu và tránh lỗi.

\_Cần nắm rõ các câu lệnh truy vấn cơ bản như:

* SELECT: Hiển thị kết quả truy vấn
* FROM: Sử dụng data từ bảng nào
* WHERE: Khai báo điều kiện
* GROUP BY: gom nhóm theo đối tượng
* ORDER BY: Sắp xếp dữ liệu
* HAVING: sắp xếp dữ liệu sau khi gom nhóm

\_ Các function JOINS như:

* INNER JOIN: Trả về tất cả các hàng khi có ít nhất một giá trị ở cả hai bảng
* LEFT JOIN: Trả lại tất cả các dòng từ bảng bên trái, và các dòng đúng với điều kiện từ bảng bên phải
* RIGHT JOIN: Trả lại tất cả các hàng từ bảng bên phải, và các dòng thỏa mãn điều kiện từ bảng bên trái
* OUTER JOIN: Trả về tất cả các dòng đúng với 1 trong các bảng.

- Các function làm việc với thời gian (ngày, tháng, năm, timestamp,..)

- Các function để clean dữ liệu (CAST, TRY\_CAST, CONVERT, TRY\_CONVERT, IS NULL,…)

\_Phải nắm rõ về biến:

+Có 2 loại biến:

* Biến vô hướng
* Biến bảng:

\_ 2 hàm chuyển đổi kiểu dữ liệu:

* Chuyển đổi ngầm (chuyển kiểu có độ ưu tiên thấp  
  sang kiểu có độ ưu tiên cao hơn)
* Chuyển đổi tường minh (chuyển dữ liệu có kiểu dữ liệu với độ ưu tiên cao hơn về kiểu dữ liệu có độ ưu tiên thấp hơn) dùng CAST or CONVERT

+Hàm CAST:

cú pháp: cast(biểu thức as kiểu\_dl)

+Hàm CONVERT :

cú pháp: convert(kiểu\_dữ\_liệu, biểu thức [,định đạng])

\_Các hàm toán học:

\_Các hàm xử lý chuỗi:

+ LEN(string) Trả về số lượng ký tự trong chuỗi, tính cả ký tự trắng đầu chuỗi

+ LTRIM(string) Loại bỏ khoảng trắng bên trái

+ RTRIM(string) Loại bỏ khoảng trắng bên phải

+ LEFT(string,length) Cắt chuỗi theo vị trí chỉ định từ trái

+ RIGHT(string,legnth) Cắt chuỗi theo vị trí chỉ định từ phải

+ TRIM(string) Cắt chuỗi 2 đầu nhưng từ bản SQL 2017 trở lên mới hoạt động

+ Replace Hàm thay thế chuỗi theo giá trị cần thay thế và cần thay thế

+REVERSE(string) Đảo ngược chuỗi truyền vào

+LOWER(string) Biến tất cả chuỗi truyền vào thành chữ thường

+UPPER(string) Biến tất cả chuỗi truyền vào thành chữ hoa

+SPACE(integer) Đếm số lượng khoảng trắng trong chuỗi.

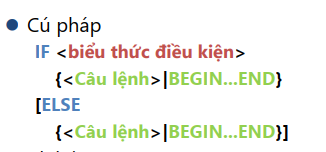
\_Các hàm ngày tháng:

+Hàm GETDATE()

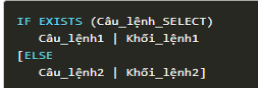
+Hàm DATENAME()

+HÀM DATEDIFF(DATEPART,DATE1,DATE2)  
\_Các câu lệnh điều kiện :

+ IF : kiểm tra một biểu thức có đúng hay không, nếu đúng thì thực thi nội dung bên trong của IF, nếu sai thì bỏ qua.



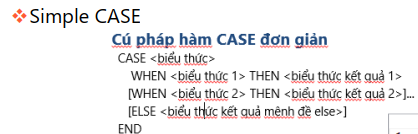
+IF Exists:

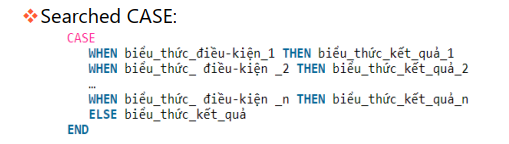


+IFF : trả về một giá trị nếu một điều kiện là TRUE hoặc một giá trị khác nếu một điều kiện là FALSE.

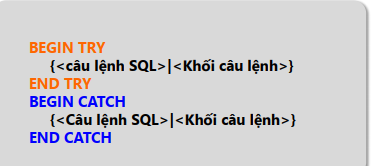


\_Câu lệnh CASE: đi qua các điều kiện và trả về một giá trị khi điều kiện đầu tiên được đáp ứng





\_Hàm sử lí lỗi TRY…CATCH:



Một số hàm ERROR thường dùng

\_

ERROR\_NUMBER() : Trả về mã số của lỗi dưới dạng số

ERROR\_MESSAGE() Trả lại thông báo lỗi dưới hình thức văn bản

ERROR\_SEVERITY() Trả lại mức độ nghiêm trọng của lỗi kiểu int

ERROR\_STATE() Trả lại trạng thái của lỗi dưới dạng số

ERROR\_LINE() : Trả lại vị trí dòng lệnh đã phát sinh ra lỗi

ERROR\_PROCEDURE() Trả về tên thủ tục/Trigger gây ra lỗi

\_Vòng lặp: Vòng lặp WHILE (WHILE LOOP) được sử dụng nếu bạn muốn

chạy lặp đi lặp lại một đoạn mã khi điều kiện cho trước trả về giá trị là TRUE.

\_ Lệnh Break (Ngắt vòng lặp)

\_ Lệnh Continue: Thực hiện bước lặp tiếp theo bỏ qua các lệnh trong

\_**Câu lệnh STORE PROCEDURE:**

--Hỗ trợ các ứng dụng tương tác nhanh và chính xác

-- Cho phép thực thi nhanh hơn cách viết từng câu lệnh SQL

-- Stored procedure có thể làm giảm bớt vấn đề kẹt đường truyền mạng, dữ liệu được gởi theo gói.

-- Stored procedure có thể sử dụng trong vấn đề bảo mật, phân quyền

--1. Cú pháp Store Procedure

Create Proc Ten\_Thủ\_tuc

@Tham\_so1 kieu\_DL, -- Tham số đầu vào (nếu có) ko ghi chữ input

@Tham\_so2 kieu\_DL output -- Tham số đầu ra (nếu có) ghi chữ output\

As

BEGIN

-- Các lệnh

END

ĐỂ GỌI SP dùng EXEC hoặc EXECUTE

\_Để cập nhật STORE PROCEDURE ta thay Create bằng Alter

--**Trigger** là một dạng đặc biệt của thủ tục lưu trữ (store procedure), được thực thi một cách tự động khi có sự thay đổi dữ liệu (do tác động của câu lệnh INSERT, UPDATE, DELETE) trên một bảng nào đó.

--❑Không thể gọi thực hiện trực tiếp Trigger bằng lệnh EXECUTE

-- 1. Trigger Insert/Update/Delete.

--cú pháp:

create Trigger tên\_Trigger on Tên\_bảng

for [Insert/update/Delete]

as

Các lệnh sql

❖Bảng Inserted:chứa bản sao các bản ghi được sửa đổi với hoạt động INSERT và UPDATE trên bảng trigger.

Hoạt động INSERT và UPDATE sẽ tiến hành chèn các bản ghi mới vào bảng Inserted và bảng trigger.

❖Bảng Deleted:chứa bản sao của các bản ghi được sửa đổi với hoạt động DELETE và UPDATE trên bảng trigger

2. Trigger After

Trigger AFTER thực thi khi hoàn thành các hoạt động INSERT, UPDATE và DELETE.

-- cú pháp:

create trigger tên\_Trigger on Tên\_bảng

after [insert/update/delete]

as

Các lệnh

Go

-HÀM FUNCTION

+Note:

--➢Các hàm luôn phải trả về một giá trị, sử dụng câu lệnh RETURN

--➢Hàm không có tham số đầu ra

--➢Không được chứa các câu lệnh INSERT, UPDATE, DELETE một bảng hoặc view đang tồn tại trong CSDL

--➢Có thể tạo bảng, bảng tạm, biến bảng và thực hiện các câu lệnh INSERT, UPDATE, DELETE trên các bảng, bảng tạm,biến bảng vừa tạo trong thân hàm

+Hàm giá trị vô hướng :

-- cú pháp

create function ten\_ham (@TS1\_neu\_co kieuDL= giatriMacDinh\_neuco,

@TS2\_neu\_co kieuDL= giatriMacDinh\_neuco,

...

)

RETURNS kieu\_DL\_vohuong

as

BEGIN

-- Cac lenh

RETURN gia\_tri\_vohuong

END

GO

**- Lưu ý:**

-Không thể truyền tham số theo tên

-Truyền đầy đủ các tham số theo vị trí. Kể cả tham số

-Tùy chọn, nếu muốn sử dụng giá trị mặc định, phải

-Đặt từ khóa DEFAULT tại đúng vị trí tham số tùy chọn đó

+Hàm giá trị bảng đa câu lệnh:

Cú pháp:

create function ten\_ham (@TS1\_neu\_co kieuDL= giatriMacDinh\_neuco,

@TS2\_neu\_co kieuDL=giatriMacDinh\_neuco,

...

)

RETURNS @Bien\_bang TABLE

(

-- giống tạo bảng

tencot1 kieuDL,

tencot2 kieuDL,

...

)

as

BEGIN

-- Các lệnh. Nên có lệnh đổ dữ liệu vào @Bien\_bang

RETURN

END

-VIEW

-- cú pháp:

create view Vw\_Ten\_view

as

<câu lệnh select>

-- gọi

select \* Vw\_Ten\_view

-- Lưu ý:

-- + Không được chứa mệnh đề INTO, hoặc ORDER BY trừ khi chứa từ khóa TOP

-- + Cột chứa giá trị được tính toán từ nhiều cột khác phải được đặt tên

------------END-------------