**BÁO CÁO**

**QUÁ TRÌNH TRIỂN KHAI VÀ THAO TÁC**

**TRÊN CƠ SỞ DỮ LIỆU CỦA MỘT TRANG BÁO ĐIỆN TỬ**

**SV: NGUYỄN MẠNH HÀ - MS: FX13850**

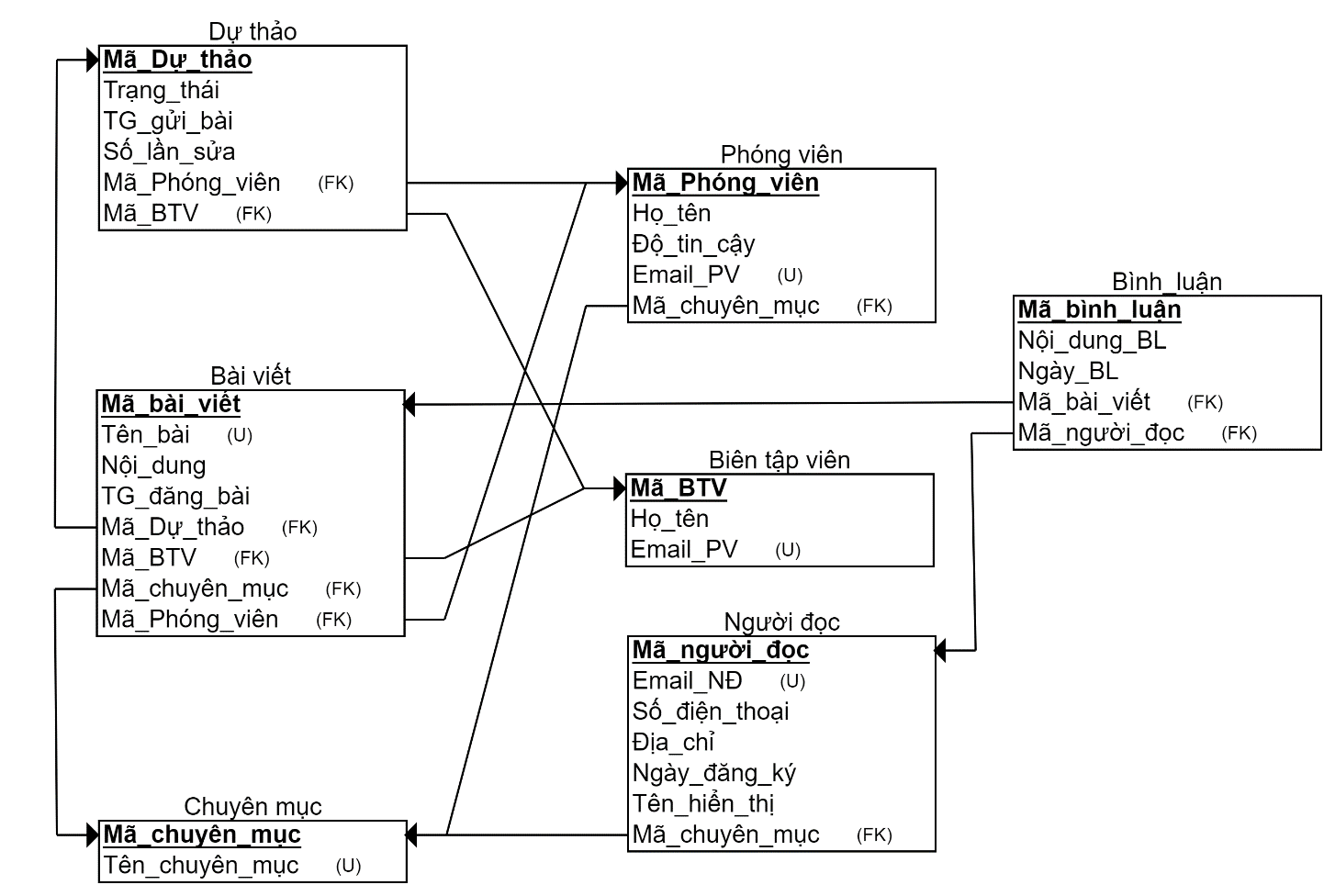
**ASSIGNMENT 2**

**MÔN HỌC DBI202x\_2.1-A\_VN**

**CÁC HỆ CƠ SỞ DỮ LIỆU**

HÀ NỘI, THÁNG 3/2022

**QUA BÀI ASM 1, TA CÓ LƯỢC ĐỒ CƠ SỞ DỮ LIỆU NHƯ SAU**

****

Link sơ đồ ERD

[https://erdplus.com/edit-diagram/ERD](https://erdplus.com/edit-diagram/c2c40e3b-45ab-4dad-90c8-2ecca9691490)

Link lược đồ CSDL

[https://erdplus.com/edit-diagram/Luoc\_do\_CSDL](https://erdplus.com/edit-diagram/d2e2af6f-5d69-45a0-a390-495062cd9323)

**CÁC NỘI DUNG CHÍNH CỦA BÀI ASSIGNMENT 2**

1. **Tạo cơ sở dữ liệu và bảng, cột thuộc tính và một số ràng buộc**

**--Tạo cơ sở dữ liệu**

CREATE DATABASE CSDL\_ASM\_2

**--Chọn cơ sở dữ liệu**

USE CSDL\_ASM\_2

**--Tạo bảng Dự thảo gồm 6 cột, khóa chính là Ma\_du\_thao**

CREATE TABLE Du\_thao(

Ma\_du\_thao VARCHAR(10) PRIMARY KEY,

TG\_gui\_bai DATE NOT NULL,

So\_lan\_sua INT DEFAULT 0,

Trang\_thai NVARCHAR(20) NOT NULL DEFAULT N'Chờ phê duyệt',

Ma\_phong\_vien VARCHAR(10) NOT NULL,

Ma\_BTV VARCHAR(10) NOT NULL);

ALTER TABLE Du\_thao ADD CONSTRAINT Check\_Trang\_thai CHECK( Trang\_thai IN

(N'Đã đăng tải', N'Được lên lịch đăng', N'Đã phê duyệt đăng', N'Đang sửa', N'Chờ phê duyệt'));

**-- Tạo bảng Bài viết gồm 8 cột, khóa chính là Ma\_bai\_viet**

CREATE TABLE Bai\_viet(

Ma\_bai\_viet VARCHAR(10) PRIMARY KEY,

Ten\_bai\_viet NVARCHAR(100) NOT NULL,

Noi\_dung NVARCHAR(4000) NOT NULL,

TG\_dang\_bai DATETIME NULL, --(TRIGGER cập nhật TG đăng bài theo giờ hiện tại)

Ma\_du\_thao VARCHAR(10) NOT NULL,

Ma\_chuyen\_muc VARCHAR(10) NULL, -- (TRIGGER để cập nhật tự động)

Ma\_phong\_vien VARCHAR(10) NULL, -- (TRIGGER để cập nhật tự động)

Ma\_BTV VARCHAR(10));

ALTER TABLE Bai\_viet ADD CONSTRAINT Check\_Ngay\_bai\_viet CHECK( TG\_dang\_bai <= CURRENT\_TIMESTAMP);

**-- Tạo bảng Phóng viên gồm 5 cột, khóa chính là Ma\_phong\_vien**

CREATE TABLE Phong\_vien(

Ma\_phong\_vien VARCHAR(10) PRIMARY KEY,

Email\_PV VARCHAR(30) NOT NULL UNIQUE,

Ho\_ten NVARCHAR(30) NOT NULL,

Do\_tin\_cay INT DEFAULT 50,

Ma\_chuyen\_muc VARCHAR(10) NOT NULL);

**-- Tạo bảng Biên tập viên gồm 3 cột, khóa chính là Ma\_BTV**

CREATE TABLE Bien\_tap\_vien(

Ma\_BTV VARCHAR(10) PRIMARY KEY,

Email\_BTV VARCHAR(30) NOT NULL UNIQUE,

Ho\_ten NVARCHAR(30) NOT NULL);

**-- Tạo bảng Người đọc gồm 7 cột, khóa chính là Ma\_nguoi\_doc**

CREATE TABLE Nguoi\_doc(

Ma\_nguoi\_doc VARCHAR(10) PRIMARY KEY,

Email\_ND VARCHAR(30) NOT NULL UNIQUE,

Ten\_hien\_thi NVARCHAR(30),

So\_dien\_thoai VARCHAR(11),

Dia\_chi NVARCHAR(100),

Ngay\_dang\_ky DATE NOT NULL DEFAULT CURRENT\_TIMESTAMP,

Ma\_chuyen\_muc VARCHAR(10) NOT NULL);

ALTER TABLE Nguoi\_doc ADD CONSTRAINT Check\_Ngay\_dang\_ky CHECK( Ngay\_dang\_ky <= CURRENT\_TIMESTAMP);

**-- Tạo bảng Chuyên mục gồm 2 cột, khóa chính là Ma\_chuyen\_muc**

CREATE TABLE Chuyen\_muc(

Ma\_chuyen\_muc VARCHAR(10) PRIMARY KEY,

Ten\_chuyen\_muc NVARCHAR(30) NOT NULL UNIQUE);

**-- Tạo bảng Bình luận gồm 5 cột, khóa chính là 2 cột Ma\_bai\_viet và Ma\_binh\_luan, có thể tạo khóa chính sau khi tạo bảng.**

CREATE TABLE Binh\_luan(

Ma\_bai\_viet VARCHAR(10) NOT NULL, --PRIMARY KEY sẽ được tạo sau

Ma\_binh\_luan INT IDENTITY NOT NULL, --PRIMARY KEY sẽ được tạo sau

Noi\_dung\_BL NVARCHAR(300) NOT NULL,

Ngay\_BL DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

Ma\_nguoi\_doc VARCHAR(10) NOT NULL);

ALTER TABLE Binh\_luan ADD CONSTRAINT Check\_Ngay\_BL CHECK( Ngay\_BL <= CURRENT\_TIMESTAMP);

1. **Tạo các ràng buộc khóa chính, khóa ngoại**

**-- Tạo ràng buộc khóa chính cho bảng Bình luận**

ALTER TABLE Binh\_luan ADD CONSTRAINT PK\_BL PRIMARY KEY (Ma\_binh\_luan, Ma\_bai\_viet);

**-- Tạo ràng buộc khóa** **ngoại:**

ALTER TABLE Du\_thao ADD CONSTRAINT FK\_DT\_PV FOREIGN KEY (Ma\_phong\_vien) REFERENCES Phong\_vien(Ma\_phong\_vien);

ALTER TABLE Du\_thao ADD CONSTRAINT FK\_DT\_BTV FOREIGN KEY (Ma\_BTV) REFERENCES Bien\_tap\_vien(Ma\_BTV);

ALTER TABLE Bai\_viet ADD CONSTRAINT FK\_BV\_DT FOREIGN KEY (Ma\_du\_thao) REFERENCES Du\_thao(Ma\_du\_thao);

ALTER TABLE Bai\_viet ADD CONSTRAINT FK\_BV\_PV FOREIGN KEY (Ma\_phong\_vien) REFERENCES Phong\_vien(Ma\_phong\_vien);

ALTER TABLE Bai\_viet ADD CONSTRAINT FK\_BV\_BTV FOREIGN KEY (Ma\_BTV) REFERENCES Bien\_tap\_vien(Ma\_BTV);

ALTER TABLE Bai\_viet ADD CONSTRAINT FK\_BV\_CM FOREIGN KEY (Ma\_chuyen\_muc) REFERENCES Chuyen\_muc(Ma\_chuyen\_muc);

ALTER TABLE Phong\_vien ADD CONSTRAINT FK\_PV\_CM FOREIGN KEY (Ma\_chuyen\_muc) REFERENCES Chuyen\_muc(Ma\_chuyen\_muc);

ALTER TABLE Nguoi\_doc ADD CONSTRAINT FK\_ND\_CM FOREIGN KEY (Ma\_chuyen\_muc) REFERENCES Chuyen\_muc(Ma\_chuyen\_muc);

ALTER TABLE Binh\_luan ADD CONSTRAINT FK\_BL\_BV FOREIGN KEY (Ma\_bai\_viet) REFERENCES Bai\_viet(Ma\_bai\_viet);

ALTER TABLE Binh\_luan ADD CONSTRAINT FK\_BL\_ND FOREIGN KEY (Ma\_nguoi\_doc) REFERENCES Nguoi\_doc(Ma\_nguoi\_doc);

1. **Chèn dữ liệu vào bảng đã tạo**

3.1 Để chèn dữ liệu vào các bảng của DATABASE ta dùng lệnh INSERT, Do lượng dữ liệu rất lớn nên lệnh ghi dữ liệu xem file Query đính kèm

**-- VD chèn dữ liệu vào bảng Chuyên mục**

INSERT INTO Chuyen\_muc VALUES

('CM001', N'Thời sự'),

('CM002', N'Khoa học'),

('CM003', N'Giải trí'),

('CM004', N'Số hóa'),

('CM005', N'Sức khỏe')

**-- VD chèn dữ liệu vào bảng Biên tập viên**

INSERT INTO Bien\_tap\_vien VALUES

('BTV2595', 'thuytrinh.btv@vnexpress.net', N'Nguyễn Thị Thúy Trinh'),

('BTV1390', 'trunghieu.btv@vnexpress.net', N'Nguyễn Trung Hiếu'),

('BTV4632', 'quynhchi.btv@vnexpress.net', N'Trần Quỳnh Chi'),

('BTV3306', 'huucong.btv@vnexpress.net', N'Quách Hữu Công'),

('BTV8362', 'ductam.btv@vnexpress.net', N'Tưởng Đức Tâm')

**-- VD chèn dữ liệu vào bảng Phóng viên**

INSERT INTO Phong\_vien VALUES

('PV22820', 'manhlinh.pv@vnexpress.net', N'Nguyễn Mạnh Linh', 40, 'CM001'),

('PV20785', 'quachhuucong.pv@vnexpress.net', N'Quách Hữu Công', 90, 'CM001'),

('PV83122', 'bichquyen.pv@vnexpress.net', N'Nguyễn Thị Bích Quyền', 60, 'CM002')...

**-- VD chèn dữ liệu vào bảng Dự thảo...**

INSERT INTO Du\_thao VALUES

('DT46204822', '2022-02-13 04:57:22', 3, N'Đã đăng tải', 'PV76705', 'BTV4632'),

('DT11625222', '2022-02-19 18:49:32', 2, N'Đã đăng tải', 'PV83122', 'BTV1390')...

3.2 Để cập nhật dữ liệu ta dùng lệnh UPDATE, ví dụ như sau

/\* Do biên tập viên Thúy Trinh bị cách ly Covid19 trong 2 tuần, bài dự thảo có mã DT46720222' được chuyển sang cho biên tập viên Quỳnh Chi phụ trách, ta thực hiện lệnh UPDATE biên tập viên như sau: \*/

UPDATE Du\_thao

SET Ma\_BTV = 'BTV4632'

WHERE Ma\_du\_thao = 'DT46720222';

3.3 Để xóa dữ liệu ta dùng lệnh DELETE, ví dụ như sau

/\*Bài dự thảo có mã 'DT64480122' được biên tập viên yêu cầu chỉnh sửa, nhưng phóng viên để quá thời hạn nộp bài nên bài dự thảo sẽ bị xóa khỏi cơ sở dữ liệu, ta thực hiện lệnh DELETE bài dự thảo như sau: \*/

DELETE FROM Du\_thao

WHERE Ma\_du\_thao = 'DT64480122';

1. **Thực hiện tạo TRIGGER**

Khi BTV đăng một bài viết mới từ bản dự thảo hoàn chỉnh, Biên tập viên chỉ cần điền các thông tin gồm Mã bài viết, Tên bài, Nội dung và Mã dự thảo của bài viết lên DATABASE, các thông tin còn lại như:

1. Mã phóng viên tự động UPDATE theo thông tin từ bản dự thảo có trước

2. Mã biên tập tự động UPDATE theo thông tin từ bản dự thảo có trước

3. Thời gian đăng bài sẽ tự động UPDATE theo giờ hiện tại

4. Mã chuyên mục tự động UPDATE theo mã chuyên mục mà phóng viên đăng ký

5. Trạng thái bài dự thảo sẽ UPDATE thành 'Đã đăng tải'

CREATE TRIGGER InsertBV\_UpdateBV\_UpdateDT

ON Bai\_viet AFTER INSERT AS

BEGIN

-- (1) UPDATE Ma\_phong\_vien

UPDATE Bai\_viet SET Ma\_phong\_vien =

(SELECT Du\_thao.Ma\_phong\_vien FROM Du\_thao join inserted ON Du\_thao.Ma\_du\_thao = inserted.Ma\_du\_thao)

FROM Bai\_viet join inserted ON Bai\_viet.Ma\_bai\_viet = inserted.Ma\_bai\_viet;

-- (2) UPDATE Ma\_BTV

UPDATE Bai\_viet SET Ma\_BTV =

(SELECT Du\_thao.Ma\_BTV FROM Du\_thao join inserted ON Du\_thao.Ma\_du\_thao = inserted.Ma\_du\_thao)

FROM Bai\_viet join inserted ON Bai\_viet.Ma\_bai\_viet = inserted.Ma\_bai\_viet;

-- (3) UPDATE TG\_dang\_bai

UPDATE Bai\_viet SET TG\_dang\_bai = CURRENT\_TIMESTAMP

FROM Bai\_viet join inserted ON Bai\_viet.Ma\_bai\_viet = inserted.Ma\_bai\_viet;

-- (4) UPDATE mã chuyên mục

UPDATE Bai\_viet SET Ma\_chuyen\_muc =

(SELECT Phong\_vien.Ma\_chuyen\_muc FROM Phong\_vien WHERE Phong\_vien.Ma\_phong\_vien = --SELECT Ma\_CM tu bang PV

(SELECT Du\_thao.Ma\_phong\_vien FROM Du\_thao join inserted ON Du\_thao.Ma\_du\_thao = inserted.Ma\_du\_thao)) -- SELECT Ma\_PV

FROM Bai\_viet join inserted ON Bai\_viet.Ma\_bai\_viet = inserted.Ma\_bai\_viet;

-- (5) UPDATE Trang\_thai Du\_thao

UPDATE Du\_thao SET Trang\_thai = N'Đã đăng tải'

FROM Du\_thao join inserted ON Du\_thao.Ma\_du\_thao = inserted.Ma\_du\_thao;

END

-- VD Bài dự thảo 'DT48968222' được đăng tải với mã bài viết 'BV48968222'

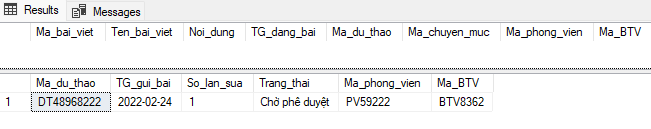
INSERT INTO Bai\_viet(Ma\_bai\_viet, Ten\_bai\_viet, Noi\_dung, Ma\_du\_thao) VALUES

('BV48968222', N'Sự thật về Trái đất', N'Trái đất có hình ELIP', 'DT48968222')

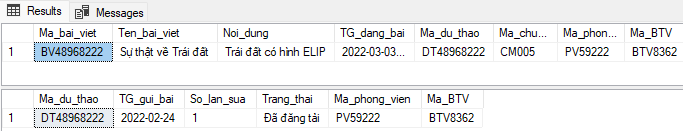
Trước khi đăng tải bài viết, ta truy vấn

SELECT \* FROM Bai\_viet WHERE Ma\_bai\_viet = 'BV48968222'

SELECT \* FROM Du\_thao WHERE Ma\_du\_thao = 'DT48968222'



Sau khi đăng tải bài viết, vẫn lệnh truy vấn cũ ta thấy sự thay đổi



Bảng bài viết sau khi INSERT 4 thông tin Mã bài viết, Tên bài, Nội dung và Mã dự thảo , 4 thông tin còn lại đều đã được TRIGGER tự động cập nhật, ngoài ra TRIGGER còn tự động UPDATE thông tin trạng thái của bài dự thảo từ Chờ phê duyệt sang Đã đăng tải.

1. **Thực hiện tạo STORE PROCEDURE**

Ta tạo một thủ tục để lập danh sách những bạn đọc bình luận từ N lần trở lên như sau:

CREATE PROCEDURE Nguoi\_doc\_N @SoBL INT

AS

BEGIN

SELECT Nguoi\_doc.Ma\_nguoi\_doc, Ten\_hien\_thi, So\_dien\_thoai, Dia\_chi, SL\_Binh\_luan

FROM (

SELECT Ma\_nguoi\_doc, count(Ma\_binh\_luan) AS SL\_Binh\_luan

FROM Binh\_luan GROUP BY Ma\_nguoi\_doc

) AS Dem\_SL\_BL

join Nguoi\_doc

ON Dem\_SL\_BL.Ma\_nguoi\_doc = Nguoi\_doc.Ma\_nguoi\_doc

WHERE SL\_Binh\_luan >= @SoBL

ORDER BY Ma\_nguoi\_doc

END

-- Thực thi thủ tục Store Procedure vừa tạo ta dùng lệnh EXCE:

EXEC Nguoi\_doc\_N @SoBL = 3

1. **Thực hiện tạo FUNCTIONS**

Ta tạo một function trả về 1 bảng gồm các bài viết được lọc theo một chuyên mục cụ thể

CREATE FUNCTION Filter\_BV\_CM(@MCM varchar(10))

RETURNS TABLE AS RETURN

(SELECT [dbo].[Bai\_viet].Ma\_bai\_viet, [dbo].[Bai\_viet].Ten\_bai\_viet,

[dbo].[Bai\_viet].Noi\_dung, convert(DATE, [dbo].[Bai\_viet].TG\_dang\_bai) as Ngay\_dang,

[dbo].[Phong\_vien].Ho\_ten as Tac\_gia, [dbo].[Chuyen\_muc].Ten\_chuyen\_muc

FROM [dbo].[Bai\_viet]

JOIN [dbo].[Phong\_vien]

ON [dbo].[Bai\_viet].Ma\_phong\_vien = [dbo].[Phong\_vien].Ma\_phong\_vien

JOIN [dbo].[Chuyen\_muc]

ON [dbo].[Bai\_viet].Ma\_chuyen\_muc = [dbo].[Chuyen\_muc].Ma\_chuyen\_muc

WHERE [dbo].[Bai\_viet].Ma\_chuyen\_muc = @MCM)

1. **Thực hiện tạo chỉ mục**

Trong SQL, các PRIMARY KEY của mỗi bảng đều là các Chỉ\_mục (Index) tự động, ta tạo thêm một số chỉ mục khác như sau:

7.1 Chỉ mục Ma\_phong\_vien, Ma\_chuyen\_muc cho bảng Bài viết để tăng tốc độ truy vấn bài báo bằng tác giả và chuyên mục

CREATE NONCLUSTERED INDEX idx\_Bai\_viet\_MPV\_and\_MCM ON Bai\_viet(Ma\_phong\_vien, Ma\_chuyen\_muc)

7.2 Chỉ mục Ma\_nguoi\_doc, Noi\_dung\_BL cho bảng Bình luận để tăng tốc độ truy vấn bằng mã người đọc hoặc nội dung BL

CREATE NONCLUSTERED INDEX idx\_Binh\_luan\_MND ON Binh\_luan(Ma\_nguoi\_doc);

CREATE NONCLUSTERED INDEX idx\_Binh\_luan\_NDBL ON Binh\_luan(Noi\_dung\_BL);

1. **Thực hiện tạo TRANSACTION**

Trường hợp giả định, Người đọc 'Đậu Quang Huy - Hải Phòng' vì lý do cá nhân muốn xóa tài khoản của mình kèm các nội dung liên quan, trên CSDL ta biết mã người đọc là 'ND96008317'. Do bảng Bình\_luận có ForeignKey đến bảng Người\_đọc, ta tiến hành xóa đồng thời thông tin của người đọc trên cả 2 bảng Bình\_Luận và Người\_đọc bằng truy vấn sau:

BEGIN TRANSACTION

DECLARE @MND Varchar(10) = 'ND96008317'

DELETE Binh\_luan WHERE Ma\_nguoi\_doc = @MND

DELETE Nguoi\_doc WHERE Ma\_nguoi\_doc = @MND

COMMIT TRANSACTION;

1. **Thực hiện các truy vấn đơn giản trên CSDL đã tạo**

9.1 Truy vấn dữ liệu trên một bảng (VD Truy vấn tất cả các bản ghi trong bảng Dự thảo)

SELECT \* FROM Du\_thao;

9.2 Truy vấn có sử dụng ORDER BY (VD Truy vấn tên bài viết, mã phóng viên, ngày đăng của các bài viết được sắp xếp từ mới đến cũ)

SELECT Ten\_bai\_viet, Ma\_chuyen\_muc, Ma\_phong\_vien, convert(DATE, TG\_dang\_bai) as Ngay\_dang

FROM Bai\_viet

ORDER BY Ngay\_dang DESC

9.3 Truy vấn sử dụng toán tử Like và các so sánh xâu ký tự (VD Truy vấn các thông tin về những người đọc có địa chỉ tại Hà Nội)

SELECT \*

FROM Nguoi\_doc

WHERE Dia\_chi like N'%Hà Nội'

9.4 Truy vấn liên quan tới điều kiện về thời gian (VD truy vấn các bình luận được viết ngoài giờ hành chính - từ sau 17h30 chiều đến trước 8h sáng)

SELECT \*

FROM Binh\_luan

WHERE convert(time, Ngay\_BL) between '00:00:00' and '07:59:59'

Or convert(time, Ngay\_BL) between '17:30:01' and '23:59:59.999'

1. **Thực hiện truy vấn dữ liệu từ nhiều bảng**

10.1 Truy vấn dữ liệu từ nhiều bảng sử dụng Inner join:

Truy vấn họ tên, email, tên bài viết của những phóng viên có điểm tin cậy lớn hơn hoặc bằng 90

SELECT PV.Ho\_ten, Email\_PV, BV.Ten\_bai\_viet

FROM Phong\_vien AS PV

join Bai\_viet AS BV

on PV.Ma\_phong\_vien = BV.Ma\_phong\_vien

WHERE Do\_tin\_cay >=90

10.2 Truy vấn sử dụng Self join:

Ta truy vấn tên các phóng viên cùng viết bài cho một chuyên mục bằng câu lệnh:

SELECT P1.Ma\_chuyen\_muc, C.Ten\_chuyen\_muc, P1.Ho\_ten, P2.Ho\_ten

FROM Phong\_vien as P1

join Phong\_vien as P2

on P1.Ma\_chuyen\_muc = P2.Ma\_chuyen\_muc and P1.Ma\_phong\_vien < P2.Ma\_phong\_vien

join Chuyen\_muc as C on P1.Ma\_chuyen\_muc = C.Ma\_chuyen\_muc

ORDER BY P1.Ma\_chuyen\_muc

10.3 Truy vấn sử dụng Outer join:

Truy vấn Mã phóng viên, họ tên, email của những phóng viên chưa có bài báo nào được đăng

SELECT PV.Ma\_phong\_vien, PV.Ho\_ten, Email\_PV

FROM Phong\_vien AS PV

Full Outer Join Bai\_viet AS BV

on PV.Ma\_phong\_vien = BV.Ma\_phong\_vien

WHERE Ma\_bai\_viet is NULL

1. **Thực hiện truy vấn dữ liệu sử dụng truy vấn con**

Truy vấn chuyên mục được người đọc quan tâm nhất dựa trên số lượng bình luận

WITH So\_luong\_binh\_luan AS(

SELECT BV.Ma\_chuyen\_muc, count(BL.Ma\_binh\_luan) as SL\_BL

FROM Bai\_viet as BV

join Binh\_luan as BL

on BV.Ma\_bai\_viet = BL.Ma\_bai\_viet

GROUP BY BV.Ma\_chuyen\_muc)

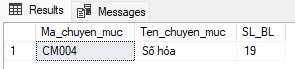
SELECT TOP(1) S.Ma\_chuyen\_muc, C.Ten\_chuyen\_muc, S.SL\_BL

FROM So\_luong\_binh\_luan as S

join Chuyen\_muc as C

on S.Ma\_chuyen\_muc = C.Ma\_chuyen\_muc

ORDER BY SL\_BL DESC



1. **Thực hiện truy vấn sử dụng GROUP BY và các hàm AGGREGATE FUNCTION**

Lập danh sách Tên bài viết, số lượng bình luận của mỗi bài viết, Ngày đăng bài, sắp xếp theo số bình luận từ lớn đến nhỏ, nếu số bình luận như nhau thì sắp xếp bài viết từ mới đến cũ, mỗi bài viết phải tối thiểu có 3 bình luận

SELECT BV.Ten\_bai\_viet, count(BL.Ma\_binh\_luan) as SL\_Binh\_luan, convert(date, TG\_dang\_bai) AS Ngày\_đăng

FROM Bai\_viet as BV

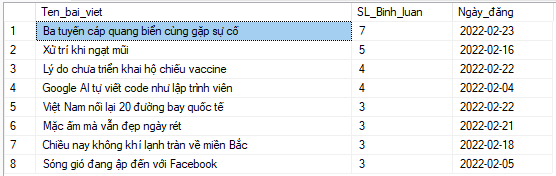
join Binh\_luan as BL

on BV.Ma\_bai\_viet = BL.Ma\_bai\_viet

GROUP BY BV.Ten\_bai\_viet, convert(date, TG\_dang\_bai)

HAVING count(BL.Ma\_binh\_luan)>= 3

ORDER BY SL\_Binh\_luan DESC, ngày\_đăng DESC



1. **Thực hiện truy vấn sử dụng FUNCTION đã viết ở mục trước**

Truy vấn các bài viết được lọc theo chuyên mục Số Hóa có Mã chuyên mục là ‘CM004’.

SELECT \*

FROM dbo.Filter\_BV\_CM('CM004')

ORDER BY Ngay\_dang DESC

