**BÀI TẬP THỰC HÀNH 6**

**Câu 1: Tổ chức/Biểu diễn thông tin:**

* 1. Vẽ sơ đồ ERD cho các quan hệ trong CSDL QLDT.

A black background with white rectangles

Description automatically generated with low confidence

Hình 1 BIỂU ĐỒ ERD

**Câu 2: Xử lý thông tin:**

Hiện thực các ràng buộc như sau. Vẽ bảng tầm ảnh hưởng và hiện thực trong SQL Server (Check, Constraints, Trigger).

**2.1.** Một sinh viên chỉ được tham gia một đề tài.

Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Xoá | Sửa |
| SV\_DETAI | + | - | + |

CREATE OR ALTER TRIGGER CHECK\_SV\_DT

ON SV\_DETAI

AFTER INSERT, UPDATE

AS

BEGIN

DECLARE @MSSV CHAR(8)

SET @MSSV = (SELECT MSSV FROM inserted)

IF (SELECT COUNT(MSDT) FROM SV\_DETAI WHERE @MSSV = SV\_DETAI.MSSV) > 1

BEGIN

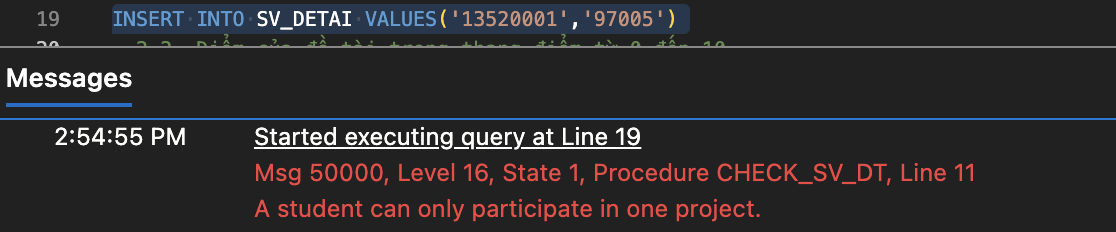
RAISERROR ('A student can only participate in one project.', 16, 1)

ROLLBACK TRANSACTION

RETURN

END

END



**2.2.** Điểm của đề tài trong thang điểm từ 0 đến 10.

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Xoá | Sửa |
| GV\_HDDT | + | - | + |
| GV\_PBDT | + | - | + |
| GV\_UVDT | + | - | + |

ALTER TABLE GV\_HDDT ADD CONSTRAINT CHECK\_GRADE\_HDDT CHECK(DIEM >= 0 AND DIEM < 10)

ALTER TABLE GV\_PBDT ADD CONSTRAINT CHECK\_GRADE\_PBDT CHECK(DIEM >= 0 AND DIEM < 10)

ALTER TABLE GV\_UVDT ADD CONSTRAINT CHECK\_GRADE\_UVDT CHECK(DIEM >= 0 AND DIEM < 10)

A screenshot of a computer

Description automatically generated

**2.3.** GV là chủ tịch hội đồng phải có học vị tiến sĩ.

|  |  |  |  |
| --- | --- | --- | --- |
| Quan hệ | Thêm | Xoá | Sửa |
| HOIDONG | + | - | + |

CREATE OR ALTER TRIGGER CHECK\_HV\_CTHD ON HOIDONG

AFTER INSERT, UPDATE

AS BEGIN

IF NOT EXISTS(SELECT MSHD

FROM INSERTED I, HOCVI HV , GV\_HV\_CN GHC

WHERE I.MSGV = GHC.MSGV AND HV.MSHV = GHC.MSHV AND HV.TENHV = N'Tiến sĩ')

begin

PRINT N'CHỦ TỊCH HỘI ĐỒNG PHẢI LÀ TIẾN SĨ'

ROLLBACK TRANSACTION

END

END

A screenshot of a computer

Description automatically generated with low confidence

Viết các truy vấn xử lý thông tin như dưới đây (tùy ý sử dụng Function, Stored Procedure hoặc Cursor):

**2.4.** Tính số lượng đề tài làm phản biện và số lượng đề tài làm ủy viên của từng GV.

CREATE TABLE GV\_DTPB\_DTUV

(

MSGV INT NOT NULL,

PBDT INT ,

UVDT INT ,

FOREIGN KEY (MSGV) REFERENCES GIAOVIEN(MSGV)

)

GO

CREATE OR ALTER PROC GV\_DTPB\_DTUV\_CUR

AS

BEGIN

DECLARE @DUYET CURSOR, @MSGV char(6), @PBDT INT, @UVDT INT

SET @DUYET = CURSOR FOR SELECT MSGV

FROM GIAOVIEN

OPEN @DUYET

FETCH NEXT FROM @DUYET INTO @MSGV

WHILE @@FETCH\_STATUS = 0

BEGIN

--logic

SELECT @PBDT = COUNT(MSDT)

FROM GV\_PBDT

WHERE @MSGV = GV\_PBDT.MSGV

SELECT @UVDT = COUNT(MSDT)

FROM GV\_UVDT

WHERE @MSGV = GV\_UVDT.MSGV

--INSERT

INSERT INTO GV\_DTPB\_DTUV

VALUES(@MSGV, @PBDT, @UVDT)

FETCH NEXT FROM @DUYET INTO @MSGV

END

CLOSE @DUYET

END

EXEC GV\_DTPB\_DTUV\_CUR

A screenshot of a computer

Description automatically generated with medium confidence

**2.5.** In ra danh sách tên các sinh viên có điểm trung bình đề tài cao nhất.

CREATE TABLE DETAI\_DIEM1

(

MSDT CHAR(6) NOT NULL,

DIEMTB FLOAT NOT NULL

)

GO

CREATE PROC DTB\_CUR1

AS

BEGIN

DECLARE @DUYET CURSOR, @msdt char(6), @dtb float

SET @DUYET = CURSOR FOR SELECT MSDT

FROM DETAI

OPEN @DUYET

FETCH NEXT FROM @DUYET INTO @msdt

WHILE @@FETCH\_STATUS = 0

BEGIN

--logic

SELECT @dtb = AVG(DIEM)

FROM ( SELECT \*

FROM GV\_HDDT

UNION

SELECT \*

FROM GV\_PBDT

UNION

SELECT \*

FROM GV\_UVDT) AS BANGTONGHOP

WHERE BANGTONGHOP.MSDT = @msdt

INSERT INTO DETAI\_DIEM1

VALUES(@msdt, @dtb)

FETCH NEXT FROM @DUYET INTO @msdt

END

CLOSE @DUYET

END

GO

--THUC THI

EXEC DTB\_CUR1

--IN RA DS HS CO DIEM TRUNG BINH CAO NHAT

GO

CREATE OR ALTER PROC HIGHEST\_AVG

AS

BEGIN

DECLARE @MAX\_DIEM FLOAT

SET @MAX\_DIEM = (SELECT MAX(DIEMTB) FROM DETAI\_DIEM1)

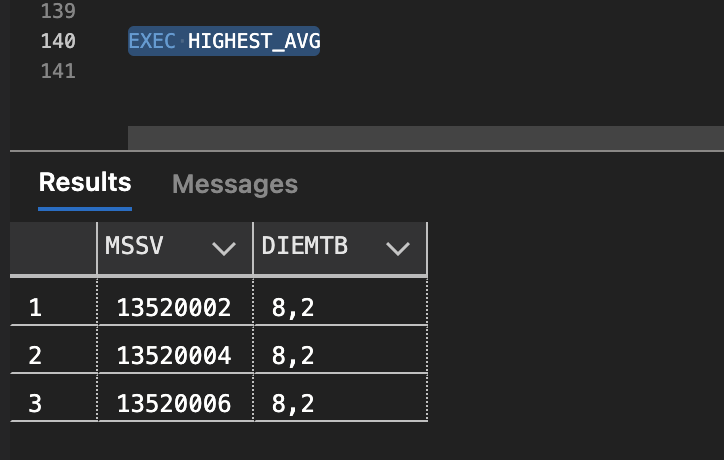
SELECT SV.MSSV, DIEM.DIEMTB

FROM DETAI\_DIEM1 DIEM, SV\_DETAI SV

WHERE DIEM.DIEMTB = @MAX\_DIEM AND DIEM.MSDT = SV.MSDT

END

EXEC HIGHEST\_AVG

****

**Câu 3: Phân quyền, xác thực:**

1. **3.1.**Tạo ra 3 users: GIANGVIEN, GIAOVU và SINHVIEN, đặt mật khẩu tuỳ ý.

CREATE LOGIN GIANGVIEN WITH PASSWORD = 'Matkhau\_giangvien1';

CREATE LOGIN GIAOVU WITH PASSWORD = 'Matkhau\_giaovu1';

CREATE LOGIN SINHVIEN WITH PASSWORD = 'Matkhau\_sinhvien1';

CREATE USER GIANGVIEN FOR LOGIN GIANGVIEN;

CREATE USER GIAOVU FOR LOGIN GIAOVU;

CREATE USER SINHVIEN FOR LOGIN SINHVIEN;

1. **3.2.**Phân quyền cho các users trên database như sau:

- GIAOVU có quyền xem và chỉnh sửa (cập nhật) trên tất cả các bảng

GRANT SELECT, UPDATE ON DATABASE::QUANLYDETAI TO GIAOVU;

- GIANGVIEN  
+ Có quyền xem trên các bảng có liên quan đến thông tin GV, các đề tài mà GV hướng dẫn, phản biện hay làm uỷ viên, xem thông tin hội đồng và danh sách các đề tài hiện có  
+ Có quyền cập nhật thông tin của mình

GRANT SELECT,UPDATE ON GIAOVIEN TO GIANGVIEN;

GRANT SELECT ON HOCVI TO GIANGVIEN;

GRANT UPDATE ON HOCHAM TO GIANGVIEN;

GRANT SELECT ON CHUYENNGANH TO GIANGVIEN;

GRANT SELECT ON GV\_HV\_CN TO GIANGVIEN;

GRANT SELECT ON DETAI TO GIANGVIEN;

GRANT SELECT ON GV\_UVDT TO GIANGVIEN;

GRANT SELECT ON GV\_PBDT TO GIANGVIEN;

GRANT SELECT ON GV\_HDDT TO GIANGVIEN;

GRANT SELECT ON HOIDONG\_GV TO GIANGVIEN;

GRANT SELECT ON HOIDONG TO GIANGVIEN;

- SINHVIEN có quyền xem thông tin của sinh viên, thông tin của hội đồng và danh sách các đề tài hiện có

GRANT SELECT ON SINHVIEN TO SINHVIEN

GRANT SELECT ON DETAI TO SINHVIEN

GRANT SELECT ON SV\_DETAI TO SINHVIEN

GRANT SELECT ON HOIDONG TO SINHVIEN

GRANT SELECT ON HOIDONG\_GV TO SINHVIEN

GRANT SELECT ON HOIDONG\_DT TO SINHVIEN

- Tất cả người dùng trên đều không có quyền xoá thông tin

DENY DELETE ON DATABASE::QUANLYDETAI TO GIAOVU;

DENY DELETE ON DATABASE::QUANLYDETAI TO GIANGVIEN;

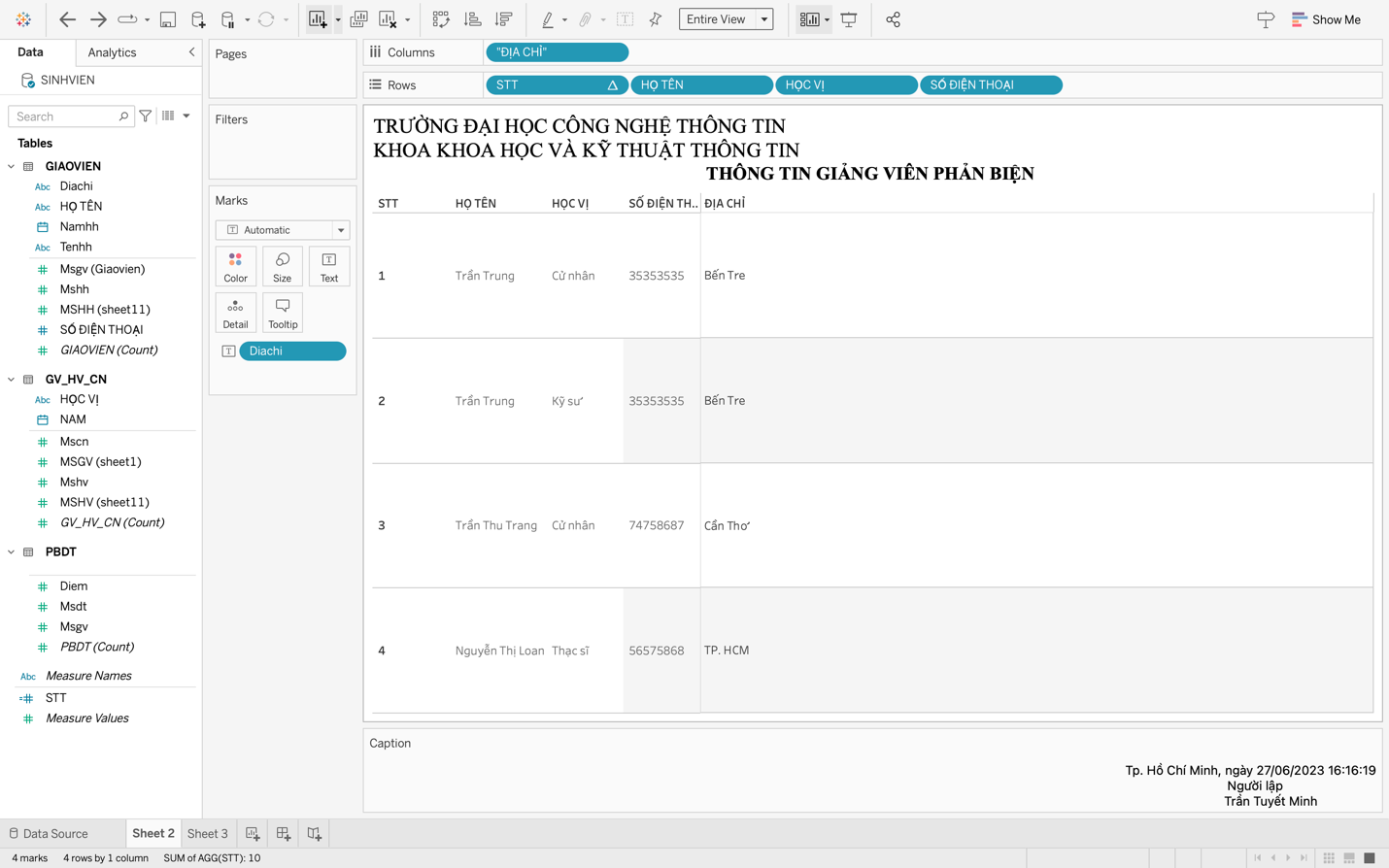
DENY DELETE ON DATABASE::QUANLYDETAI TO SINHVIEN;

**Câu 4: Trình bày thông tin:**

Dùng Crystal Report để thiết kế các Reports như sau (Nếu là máy MAC thì có thể sử dụng phần mềm Tableau):

**4.1.** Thông tin GV phản biện:

- Màn hình Design

****

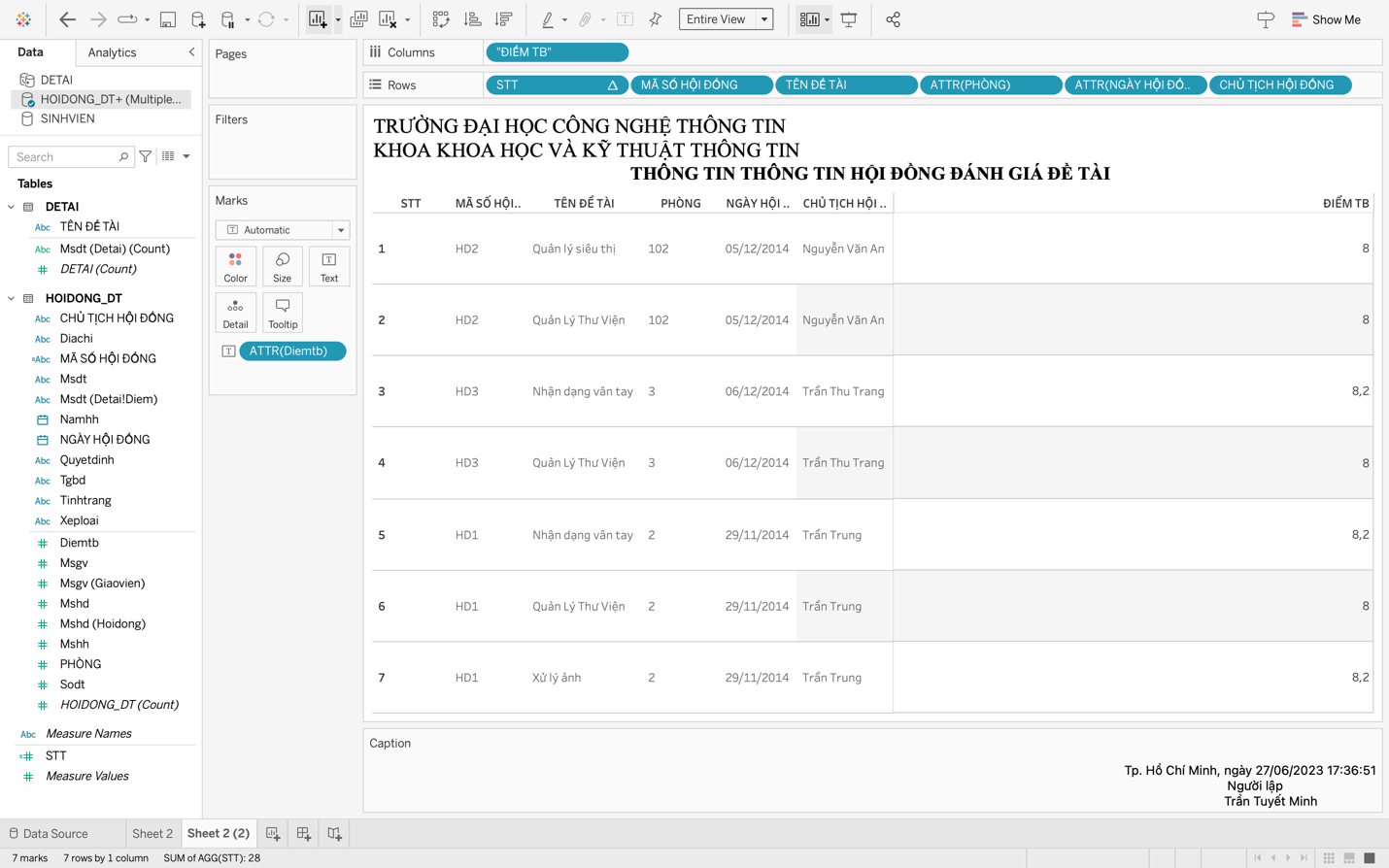
* Màn hình Preview

A screenshot of a computer

Description automatically generated with medium confidence

**4.2.** Thông tin hội đồng đánh giá đề tài

- Màn hình design



* Màn hình Preview

A screenshot of a computer

Description automatically generated with medium confidence

**5.1.** Giả sử cần chuyển bài toán QLDT sang một mô hình CSDL khác. Bạn sẽ lựa chọn mô hình nào (Phân tán, NoSQL, HĐT, Di động). Tại sao?  
**5.2.** Cho ví dụ chuyển đổi sang mô hình đã chọn (Chọn 2 bảng bất kỳ khi chuyển, không cần chuyển hết tất cả các bảng).

Lựa chọn mô hình CSDL phân tán trong trường hợp bài toán QLDT. Dưới đây là lý do:

1. Khối lượng dữ liệu lớn: Với bài toán QLDT, có thể dự đoán rằng dữ liệu sẽ tăng theo thời gian do số lượng sinh viên, giáo viên và đề tài tăng lên. Mô hình phân tán có thể xử lý khối lượng dữ liệu lớn và mở rộng hệ thống để đáp ứng nhu cầu lưu trữ và truy xuất dữ liệu của QLDT.
2. Tăng khả năng mở rộng: Mô hình phân tán cho phép chia nhỏ dữ liệu và phân phối trên nhiều máy chủ. Điều này giúp tăng khả năng mở rộng của hệ thống, cho phép QLDT mở rộng quy mô mà không gặp hạn chế về tài nguyên và hiệu suất.
3. Đảm bảo độ tin cậy: Với mô hình phân tán, dữ liệu được phân tán trên nhiều máy chủ. Điều này giúp đảm bảo độ tin cậy của hệ thống, vì một máy chủ gặp sự cố không ảnh hưởng đến toàn bộ hệ thống và dữ liệu vẫn có sẵn trên các máy chủ khác.
4. Tối ưu hiệu suất và thời gian truy xuất dữ liệu: Mô hình phân tán cho phép truy xuất dữ liệu song song từ nhiều máy chủ, giúp tối ưu hiệu suất và giảm thời gian truy xuất dữ liệu. Điều này quan trọng đối với QLDT để đáp ứng yêu cầu truy xuất thông tin nhanh chóng và đồng thời xử lý nhiều yêu cầu truy xuất cùng lúc.

Chuyển đổi Bảng SINHVIEN và DETAI sang csdl phân tán:

1. SINHVIEN (Sinh viên)

MSSV: char

TENSV: nvarchar(30)

SODT: varchar(10)

LOP: char(10)

DIACHI: nchar(50)

2. DETAI (Đề tài)

MSDT: char(6)

TENDT: nvarchar(30)

3. Bảng phân vùng SINHVIEN

SINHVIEN\_Partition1 (Máy chủ 1)

MSSV: char

TENSV: nvarchar(30)

SODT: varchar(10)

SINHVIEN\_Partition2 (Máy chủ 2)

MSSV: char

LOP: char(10)

DIACHI: nchar(50)

4. Bảng phân vùng DETAI

DETAI\_Partition1 (Máy chủ 1)

MSDT: char(6)

TENDT: nvarchar(30)