**ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐHQG TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**BÁO CÁO ĐỒ ÁN**

**(SCRIPT)**

**Giáo viên lý thuyết :**

Cô: Phạm Thị Bạch Huệ

**Giáo viên hướng dẫn thực hành :**

Thầy: Lương Vĩ Minh

**Thành viên nhóm:**

Nguyễn Tiến Dũng – 1312096

Nguyễn Minh Dũng – 1312094

# STRUCTURE

|  |
| --- |
| USE MASTER  GO  IF DB\_ID('StudentCoursesManagement') IS NOT NULL  DROP DATABASE StudentCoursesManagement  GO  CREATE DATABASE StudentCoursesManagement  GO  USE StudentCoursesManagement  SET DATEFORMAT DMY  GO  CREATE TABLE SINHVIEN(  MASV INT IDENTITY(1,1),  HOTEN NVARCHAR(50),  NGAYSINH DATETIME,  MALOP CHAR(6),  PRIMARY KEY (MASV)  )  CREATE TABLE LOP(  MALOP CHAR(6),  TENLOP CHAR(6),  SISO INT,  PRIMARY KEY (MALOP)  )  CREATE TABLE MONHOC(  MAMH CHAR(5),  TENMH NVARCHAR(50),  SOTIETLT INT,  SOTIETTH INT,  SOCHOMAX INT,  CHECK (SOTIETLT >= 30 AND SOTIETLT <= 60),  CHECK (SOTIETTH >= 30 AND SOTIETTH <= 60),  CHECK (SOCHOMAX >= 30 AND SOCHOMAX <= 120),  PRIMARY KEY (MAMH)  )  CREATE TABLE MHMO(  MAMH CHAR(5),  NAM INT,  HOCKY INT,  CHECK (NAM >= 1990),  CHECK (HOCKY = 1 OR HOCKY = 2),  PRIMARY KEY (MAMH, NAM, HOCKY)  )  CREATE TABLE DANGKY(  MASV INT,  MAMH CHAR(5),  NAM INT,  HOCKY INT,  DIEM REAL,  CHECK (DIEM >= 0 AND DIEM <=10),  PRIMARY KEY (MASV, MAMH, NAM, HOCKY)  )  CREATE TABLE KETQUA(  MASV INT,  NAM INT,  DIEMTB REAL,  XEPLOAI CHAR(5),  CHECK (DIEMTB >=0 AND DIEMTB <=10),  CHECK (NAM >= 1990),  PRIMARY KEY (MASV, NAM)  ) |

|  |
| --- |
| --FOREIGN KEY  ALTER TABLE SINHVIEN  ADD CONSTRAINT FK\_SINHVIEN\_LOP  FOREIGN KEY (MALOP)  REFERENCES LOP(MALOP)  ALTER TABLE MHMO  ADD CONSTRAINT FK\_MHMO\_MONHOC  FOREIGN KEY (MAMH)  REFERENCES MONHOC(MAMH)  ALTER TABLE DANGKY  ADD CONSTRAINT FK\_DANGKY\_SINHVIEN  FOREIGN KEY (MASV)  REFERENCES SINHVIEN(MASV),  CONSTRAINT FK\_DANGKY\_MHMO  FOREIGN KEY (MAMH, NAM, HOCKY)  REFERENCES MHMO(MAMH, NAM, HOCKY)  ALTER TABLE KETQUA  ADD CONSTRAINT FK\_KETQUA\_SINHVIEN  FOREIGN KEY (MASV)  REFERENCES SINHVIEN(MASV) |

# TRIGGER

|  |
| --- |
| -- MÃ R1  -- LOAI PHAI TUONG XUNG VOI DTB  -- BẢNG TẦM ẢNH HƯỞNG: KETQUA: I(+), D(-), U(+(DIEMTB, XEPLOAI))  IF OBJECT\_ID('UTR\_RB1', 'TR') IS NOT NULL  DROP TRIGGER UTR\_RB1  GO  CREATE TRIGGER UTR\_RB1  ON KETQUA  FOR INSERT, UPDATE  AS BEGIN  DECLARE @MASV INT, @NAM INT  SET @MASV = (SELECT MASV FROM INSERTED)  SET @NAM = (SELECT NAM FROM INSERTED)  --ĐIỂM TRUNG BÌNH THAY ĐỔI THÌ PHẢI SỬA LẠI XẾP LOẠI  IF UPDATE(DIEMTB)  BEGIN  DECLARE @DIEMTB REAL;  SET @DIEMTB = (SELECT DIEMTB FROM INSERTED);  IF (@DIEMTB < 6.5)  BEGIN  UPDATE dbo.KETQUA  SET XEPLOAI = 'KHA'  WHERE MASV = @MASV AND NAM = @NAM;  END;    IF (@DIEMTB >= 6.5 AND @DIEMTB < 9)  BEGIN  UPDATE dbo.KETQUA  SET XEPLOAI = 'GIOI'  WHERE MASV = @MASV AND NAM = @NAM;  END;    IF (@DIEMTB > 9)  BEGIN  UPDATE dbo.KETQUA  SET XEPLOAI = 'XS'  WHERE MASV = @MASV AND NAM = @NAM;  END;  END;    --THAY ĐỔI XẾP LOẠI THÌ PHẢI KIỂM TRA ĐIỂM TRUNG BÌNH  IF UPDATE(XEPLOAI)  BEGIN  DECLARE @DIEMTB1 REAL, @XEPLOAI CHAR(5);    SET @DIEMTB1 = (SELECT DIEMTB  FROM dbo.KETQUA  WHERE MASV = @MASV AND NAM = @NAM);    SET @XEPLOAI = (SELECT XEPLOAI FROM Inserted);    IF ((@DIEMTB1 < 6.5 AND @XEPLOAI <> 'KHA') OR  (@DIEMTB1 >= 6.5 AND @DIEMTB1 < 9 AND @XEPLOAI <> 'GIOI') OR  (@DIEMTB1 >= 9 AND @XEPLOAI <> 'XS'))  BEGIN  RAISERROR(N'Lỗi: Xếp hạng không phù hợp với điểm trung bình ',16,1);  ROLLBACK;  END;  END;  END;  --------------------------------------------------------------------------------------  --------------------------------------------------------------------------------------  --R2  --DTB BANG TONG DIEM CHIA TONG MON  --NẾU UPDATE ĐIỂM Ở ĐĂNG KÍ THÌ CẬP NHẬT LẠI ĐIỂM TRUNG BINGH  --BẢNG TẦM ẢNH HƯỜNG: DANGKY: I(+), D(+), U(+(DIEM))  IF OBJECT\_ID('UTR\_RB2', 'TR') IS NOT NULL  DROP TRIGGER UTR\_RB2  GO  CREATE TRIGGER UTR\_RB2  ON DANGKY  FOR INSERT, UPDATE, DELETE  AS BEGIN  IF UPDATE(DIEM)  BEGIN  DECLARE @MASV INT, @NAM INT, @TONGMON INT, @TONGDIEM INT, @DIEMTB REAL;  SET @MASV = (SELECT MASV FROM INSERTED);  SET @NAM = (SELECT NAM FROM INSERTED);  SET @TONGMON = (SELECT COUNT(\*)  FROM dbo.DANGKY  WHERE MASV = @MASV AND  NAM = @NAM);    SET @TONGDIEM = (SELECT SUM(DIEM)  FROM dbo.DANGKY  WHERE MASV = @MASV AND  NAM = @NAM);    SET @DIEMTB = @TONGDIEM / @TONGMON;  UPDATE dbo.KETQUA  SET DIEMTB = @DIEMTB  WHERE MASV = @MASV AND NAM = @NAM;  END;  END;  --------------------------------------------------------------------------------------  --------------------------------------------------------------------------------------  --R3  --KHÔNG CHO ĐĂNG KÍ QUÁ 8 MÔN  --BẢNG TẦM ẢNH HƯỞNG: DANGKY: I(+), D(-), U(-)  IF OBJECT\_ID('UTR\_RB3', 'TR') IS NOT NULL  DROP TRIGGER UTR\_RB3  GO  CREATE TRIGGER UTR\_RB3  ON dbo.DANGKY  FOR INSERT  AS BEGIN  IF ((SELECT COUNT(\*)  FROM Inserted) > 0)  BEGIN  DECLARE @MASV INT, @MAMH CHAR(5), @NAM INT, @HOCKY INT;  SET @MASV = (SELECT Inserted.MASV FROM Inserted);  SET @MAMH = (SELECT Inserted.MAMH FROM Inserted);  SET @NAM = (SELECT Inserted.NAM FROM Inserted);  SET @HOCKY = (SELECT Inserted.HOCKY FROM Inserted);  IF ((SELECT COUNT(\*)  FROM dbo.DANGKY  WHERE MASV = @MASV AND MAMH = @MAMH AND  NAM = @NAM AND HOCKY = @HOCKY) >=8)  BEGIN  RAISERROR('RB1: ERROR',16,1);  ROLLBACK;  END;  END;  END; |

# TRANSACTION

|  |
| --- |
| --------------------------------------------------------------------------------------  -- Trasaction: Get All Opened Courses  IF OBJECT\_ID('sp\_GetAllOpenedCourses', 'P') is not null  DROP PROC sp\_GetAllOpenedCourses  GO  CREATE PROC sp\_GetAllOpenedCourses  @year INT,  @semester INT  AS  BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    BEGIN TRAN  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  SELECT \*  FROM MHMO  WHERE (NAM = @year AND HOCKY = @semester)  COMMIT TRAN    RETURN 1;    END;  --------------------------------------------------------------------------------------  -- Trasaction: Register a Course  IF OBJECT\_ID('sp\_RegisterCourse', 'P') is not null  DROP PROC sp\_RegisterCourse  GO  CREATE PROC sp\_RegisterCourse  @studentID INT,  @objectID CHAR(6),  @year INT,  @semester INT  AS  BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2  OR @objectID IS NULL OR @studentID IS NULL)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    -- Kiểm tra mã môn học mở có tồn tại  IF (NOT EXISTS (SELECT MAMH  FROM MHMO  WHERE (MAMH = @objectID AND NAM = @year AND HOCKY = @semester)))  BEGIN  PRINT(N'Lỗi: Thông tin môn học mở không tồn tại');  RETURN 0;  END;    -- Kiểm tra mã sinh viên có tồn tại  IF (NOT EXISTS (SELECT MASV  FROM SINHVIEN  WHERE (MASV = @studentID)))  BEGIN  PRINT(N'Lỗi: Mã sinh viên không tồn tại');  RETURN 0;  END;    BEGIN TRAN  SET TRANSACTION ISOLATION LEVEL SERIALIZABLE  -- Kiểm tra còn chỗ  IF ((SELECT DISTINCT COUNT(\*)  FROM DANGKY WITH (ROWLOCK, XLOCK)  WHERE (MAMH = @objectID AND NAM = @year AND HOCKY = @semester)) <  (SELECT SOCHOMAX FROM MONHOC WITH (ROWLOCK, XLOCK)  WHERE MAMH = @objectID))  BEGIN  INSERT INTO DANGKY (MASV, MAMH, NAM, HOCKY, DIEM)  VALUES (@studentID, @objectID, @year, @semester, NULL);  END;  ELSE  BEGIN  PRINT (N'Đã hết chỗ');  RETURN 0;  END;    COMMIT TRAN      PRINT (N'Đã đăng ký thành công');  RETURN 1;    END;  --------------------------------------------------------------------------------------  -- Trasaction: Cancel a Course  IF OBJECT\_ID('sp\_CancelCourse', 'P') is not null  DROP PROC sp\_CancelCourse  GO  CREATE PROC sp\_CancelCourse  @studentID INT,  @objectID CHAR(6),  @year INT,  @semester INT  AS  BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2  OR @objectID IS NULL OR @studentID IS NULL)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    -- Kiểm tra mã môn học có tồn tại  IF (NOT EXISTS (SELECT MAMH  FROM MHMO  WHERE (MAMH = @objectID AND NAM = @year AND HOCKY = @semester)))  BEGIN  PRINT(N'Lỗi: Thông tin môn học mở không tồn tại');  RETURN 0;  END;    -- Kiểm tra mã sinh viên có tồn tại  IF (NOT EXISTS (SELECT MASV  FROM SINHVIEN  WHERE (MASV = @studentID)))  BEGIN  PRINT(N'Lỗi: Thông tin sinh viên không tồn tại');  RETURN 0;  END;    BEGIN TRAN  SET TRANSACTION ISOLATION LEVEL READ COMMITTED  DELETE FROM DANGKY  WHERE (MASV = @studentID AND MAMH = @objectID AND NAM = @year  AND HOCKY = @semester);  COMMIT TRAN    RETURN 1;    END;  -------------------------------------------------------------------------------------  -- Trasaction: Open a Course  IF OBJECT\_ID('sp\_OpenCourse', 'P') is not null  DROP PROC sp\_OpenCourse  GO  CREATE PROC sp\_OpenCourse  @objectID CHAR(6),  @year INT,  @semester INT  AS  BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());      IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2  OR @objectID IS NULL)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    BEGIN TRAN  SET TRANSACTION ISOLATION LEVEL READ COMMITTED  -- Kiểm tra mã khóa học có tồn tại  IF (NOT EXISTS (SELECT MAMH  FROM MHMO  WHERE (MAMH = @objectID AND NAM = @year AND HOCKY = @semester)))  BEGIN  INSERT INTO MHMO (MAMH, NAM, HOCKY)  VALUES (@objectID, @year, @semester);  RETURN 1;  END;  COMMIT TRAN    RETURN 0;    END;  --------------------------------------------------------------------------------------  -- Trasaction: Update Max Number Of Students Of Course  IF OBJECT\_ID('sp\_UpdateMaxStudentsOfCourse', 'P') is not null  DROP PROC sp\_UpdateMaxStudentsOfCourse  GO  CREATE PROC sp\_UpdateMaxStudentsOfCourse  @objectID CHAR(6),  @number INT  AS  BEGIN  -- Kiểm tra điều kiện  IF (@number <= 0 OR @objectID IS NULL)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    BEGIN TRAN  SET TRANSACTION ISOLATION LEVEL READ COMMITTED    UPDATE MONHOC  SET SOCHOMAX = SOCHOMAX + @number  WHERE MAMH = @objectID;    COMMIT TRAN    RETURN 1;    END;  --------------------------------------------------------------------------------------  --Transaction: Get Result of Student  IF OBJECT\_ID('sp\_GetResultofStudent', 'P') IS NOT NULL  DROP PROC sp\_GetResultofStudent  GO  CREATE PROC sp\_GetResultofStudent  @studentID INT,  @year INT,  @semester INT  AS BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2  OR @studentID IS NULL)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    --Kiểm tra điều kiện  IF (@studentID NOT IN (SELECT MASV  FROM SINHVIEN))  BEGIN  PRINT(N'Lỗi: Mã sinh viên không tồn tại');  RETURN 0;  END;    BEGIN TRAN  SET TRAN ISOLATION LEVEL READ COMMITTED    SELECT \*  FROM DANGKY WITH (ROWLOCK)  WHERE (MASV = @studentID AND NAM = @year AND HOCKY = @semester);  WAITFOR DELAY '00:00:05';  COMMIT TRAN    RETURN 1;  END;  ---------------------------------------------------------------------------------------------------  --Transaction: Update Result of Student  IF OBJECT\_ID('sp\_UpdateResultofStudent', 'P') IS NOT NULL  DROP PROC sp\_UpdateResultofStudent  GO  CREATE PROC sp\_UpdateResultofStudent  @studentID INT,  @objectID CHAR(6),  @year INT,  @semester INT,  @score REAL  AS BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year OR @semester < 1 OR @semester > 2  OR @objectID IS NULL OR 0 > @score OR @score > 10)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;    --Kiểm tra điều kiện  IF (@studentID NOT IN (SELECT MASV  FROM SINHVIEN))  BEGIN  PRINT(N'Lỗi: Mã sinh viên không tồn tại');  RETURN 0;  END;      BEGIN TRAN  SET TRAN ISOLATION LEVEL READ COMMITTED    UPDATE DANGKY  SET DIEM = @score  WHERE (MASV = @studentID AND NAM = @year AND HOCKY = @semester  AND MAMH = @objectID);    COMMIT TRAN    RETURN 1;  END;  ---------------------------------------------------------------------------------------------------  --Transaction: Get Result of Student in a Year  IF OBJECT\_ID('sp\_GetResultofStudentInYear', 'P') IS NOT NULL  DROP PROC sp\_GetResultofStudentInYear  GO  CREATE PROC sp\_GetResultofStudentInYear  @studentID INT,  @year INT  AS BEGIN  -- Kiểm tra điều kiện  DECLARE @currentYear INT;  SELECT @currentYear = YEAR(GETDATE());    IF (@year < 1990 OR @currentYear < @year)  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;  --Kiểm tra điều kiện  IF (@studentID NOT IN (SELECT MASV  FROM SINHVIEN))  BEGIN  PRINT(N'Lỗi: Mã sinh viên không tồn tại');  RETURN 0;  END;    BEGIN TRAN  SET TRAN ISOLATION LEVEL READ COMMITTED    SELECT \*  FROM KETQUA WITH (ROWLOCK)  WHERE (MASV = @studentID AND NAM = @year);    COMMIT TRAN    RETURN 1;  END;  --------------------------------------------------------------------------------------  --Transaction: Get All Students  IF OBJECT\_ID('sp\_GetAllStudents', 'P') is not null  DROP PROC sp\_GetAllStudents  GO  CREATE PROC sp\_GetAllStudents  AS BEGIN  BEGIN TRAN  SET TRAN ISOLATION LEVEL SERIALIZABLE    SELECT \*  FROM SINHVIEN;    COMMIT TRAN    RETURN 1;  END;  -------------------------------------------------------------------------------------  --Transaction: Add New Student  IF OBJECT\_ID('sp\_AddNewStudent', 'P') IS NOT NULL  DROP PROC sp\_AddNewStudent  GO  CREATE PROC sp\_AddNewStudent  @studentID INT,  @fullname NVARCHAR(50),  @birthday DATETIME,  @classID CHAR(6)  AS BEGIN  --Kiểm tra điều kiện  IF (@studentID IN (SELECT MASV FROM SINHVIEN))  BEGIN  PRINT(N'Lỗi: Mã sinh viên đã tồn tại');  RETURN 0;  END;  --Kiểm tra điều kiện  IF (@classID NOT IN (SELECT MALOP FROM LOP))  BEGIN  PRINT(N'Lỗi: Mã lớp không tồn tại');  RETURN 0;  END;    BEGIN TRAN  SET TRAN ISOLATION LEVEL READ COMMITTED    INSERT SINHVIEN (MASV, HOTEN, NGAYSINH, MALOP)  VALUES(@studentID, @fullname, @birthday, @classID);    UPDATE LOP  SET SISO = SISO + 1  WHERE MALOP = @classID;    COMMIT TRAN  RETURN 1;  END;  --------------------------------------------------------------------------------------  --Transaction: Move class for Student  IF OBJECT\_ID('sp\_ChangeClass', 'P') IS NOT NULL  DROP PROC sp\_ChangeClass  GO  CREATE PROC sp\_ChangeClass  @studentID INT,  @newClassID CHAR(6)  AS BEGIN  --Kiểm tra điều kiện  IF ((@studentID NOT IN (SELECT MASV FROM SINHVIEN)) OR  (@newClassID NOT IN (SELECT MALOP FROM LOP)))  BEGIN  PRINT(N'Lỗi: Dữ liệu đầu vào không hợp lệ');  RETURN 0;  END;  BEGIN TRAN  SET TRAN ISOLATION LEVEL READ COMMITTED  DECLARE @oldClassID CHAR(6)  SET @oldClassID = (SELECT MALOP FROM SINHVIEN WHERE MASV = @studentID);    UPDATE SINHVIEN  SET MALOP = @newClassID  WHERE MASV = @studentID;  UPDATE LOP  SET SISO = SISO + 1  WHERE MALOP = @newClassID;  UPDATE LOP  SET SISO = SISO - 1  WHERE MALOP = @oldClassID;  COMMIT TRAN    RETURN 1;  END;  -------------------------------------------------------------------------------------- |