Đại học Đà Nẵng

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT



BÁO CÁO CÁ NHÂN

**THỰC HÀNH CƠ SỞ DỮ LIỆU II**

Đề tài:

**Đăng Kí Tín Chỉ Trực Tuyến Của Sinh Viên Trường ĐHSPKT**

Họ tên sinh viên:Nguyễn Văn Linh

Mã sinh viên: 22115053122225

Lớp HP:123TCSDL201

**Đà nẵng ngày 21 tháng 09 năm 2023**

**Phụ lục:**

[1. Sơ đồ ERD 9](#_Toc153899150)

[1.1 Ảnh ERD về đăng kí tín chỉ trực tuyến trường ĐHSPKT-ĐN. 9](#_Toc153899151)

[2. Bổ sung kiểu dữ liệu và ràng buộc cho các bảng từ sơ đồ ERD trên theo mẫu dưới đây: 9](#_Toc153899152)

[Tên bảng: chuongTrinhDaoTao 9](#_Toc153899153)

[Tên bảng: hocPhan 9](#_Toc153899154)

[Tên bảng: chuongTrinhDaoTao\_HocPhan 10](#_Toc153899155)

[Tên bảng: giaoVien 10](#_Toc153899156)

[Tên bảng: thoiGian 10](#_Toc153899157)

[Tên bảng: Phong 10](#_Toc153899158)

[Tên bảng: ThoiGian\_Phong 11](#_Toc153899159)

[Tên bảng: LopHocPhan 11](#_Toc153899160)

[Tên bảng: Khoa 11](#_Toc153899161)

[Tên bảng: LopSH 11](#_Toc153899162)

[Tên bảng: SinhVien 12](#_Toc153899163)

[Tên bảng: DangKiTinChi 12](#_Toc153899164)

[Tên bảng: DangKiChiTiet 13](#_Toc153899165)

[3. Sử dụng câu lệnh CREATE TABLE để tạo các bảng trong cơ sở dữ liệu từ ERD với các trường, kiểu dữ liệu và ràng buộc từ câu 1. 13](#_Toc153899166)

[4. Sử dụng câu lệnh ALTER TABLE 15](#_Toc153899167)

[a. Bảng SinhVien, thay đổi ngày sinh của cột ngaySinh thành Datetime, thêm ràng buộc cho ngày sinh phải lớn hơn hoặc = 18 tuổi 15](#_Toc153899168)

[b. Thay đổi kiểu dữ liệu cho cột gioiTinh bảng SinhVien và GiaoVien 15](#_Toc153899169)

[c. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho côt maSV của Bảng DangKiTinChi 15](#_Toc153899170)

[d. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột tenGV và email của bảng GiaoVien 15](#_Toc153899171)

[e. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột tenHP và soTC của bảng HocPhan 15](#_Toc153899172)

[f. Sử dụng ALTER TABLE để thêm ràng buộc NOT NULL cho các cột maHP, maGV, soLuongDK, maTG\_HP, maHocKi của bảng LopHocPhan 15](#_Toc153899173)

[g. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột maThoiGian và maPhong của bảng ThoiGian\_PhongHoc 16](#_Toc153899174)

[h. Thêm ràng buộc cho cột gioiTinh ở bảng SinhVien và GiaoVien 16](#_Toc153899175)

[i. Thêm ràng buộc cho cột CHECK SDT ở bảng SinhVien và bảng GiaoVien 16](#_Toc153899176)

[j. Thêm ràng buộc CHECK email ở bảng SinhVien và bảng GiaoVien 16](#_Toc153899177)

[k. Thêm ràng buộc NOT NULL cho cột matKhau ở bảng SinhVien và GiaoVien 16](#_Toc153899178)

[l. Thêm ràng buộc cho SoTC ở bảng HocPhan phải lớn hơn 0 17](#_Toc153899179)

[CHECK(SoTC>0); 17](#_Toc153899180)

[m. Thêm ràng buộc cho soLuongDK và soLuongDDK phải lớn hơn 0, soLuongDK<=soLuongDDK ở bảng LopHocPhan 17](#_Toc153899181)

[n. Thêm rang buộc cho tietBD và tietKT phải thuộc từ tiết 1đến tiết 15 , và tietBD<=tietKTcho bảng ThoiGian\_LHP 17](#_Toc153899182)

[o. Thêm ràng buộc cho namBD ở bảng lopSH phải <=năm hiện tại 17](#_Toc153899183)

[p. Thay đổi khóa chỉnh của bảng ChiTietDangKi thành hai cột maDKCT va maDK vì maDK là mã đăng kí chính thức của sinh viên quản lí các maDKCT con của từng lớp học phần mà sinh viên đăng kí (hay nói cách khác là một maDK có thể đăng kí nhiều maDKCT con) 17](#_Toc153899184)

[q. Sửa đổi cột maLHP và maDK của bảng ChiTietDangKi thành not null vì nếu có một hàng null sẽ không đăng kí được môn học 18](#_Toc153899185)

[5. Sử dụng câu lệnh INSERT để thêm dữ liệu cho các bảng 18](#_Toc153899186)

[6. Sử dụng câu lệnh UPDATE để UPDATE những dữ liệu cần thiết trong DATABASE 20](#_Toc153899187)

[a. Cập nhật tên khoa có mã’KHOA001’ thành tên ‘Khoa Công Nghệ Thông Tin’ 20](#_Toc153899188)

[b. Cập nhật tên lớp học có mã ‘LSH001’ thành tên ‘Lớp SH 22T2’ 20](#_Toc153899189)

[c. Cập nhật số lượng đăng kí mới cho lớp học phần có mã ‘LHP001’ 20](#_Toc153899190)

[d. Cập nhật thời gian đăng kí của sinh viên có mã ‘SV004’ thành thời gian’14:15:00’ 20](#_Toc153899191)

[e. Cập nhật số lượng đăng kí mới soLuongDDK=soLuongDDK-10 cho tất cả lớp học phần (LopHocPhan’) 21](#_Toc153899192)

[f. Cập nhật học phần có số tín chỉ =4 thành số tín chỉ = 3 nếu lớp đó có mã HP là ‘HP001’ 21](#_Toc153899193)

[g. Cập nhật giới tính của giáo Viên có giới tính là ‘M’ đổi thành ‘F’ và ngược lại. 21](#_Toc153899194)

[h. Cập nhật lại tên lớp sinh hoạt của những lớp có mã lớp sinh hoạt ‘LSH003’, ‘LSH004’, ‘LSH005’ lần lượt đổi lại thành ‘Lớp SH3’,’Lớp SH4’,’Lớp SH5’ 21](#_Toc153899195)

[i. UPDATE maHocKi của bảng LopHocPhan thành’123’ cho các lớp có mã học phần ‘HP001’ , ‘HP002’, ‘HP003’, ‘HP004’ và maHocKi của lớp có mã học phần ‘HP005’ thành ‘223’ 21](#_Toc153899196)

[7. Sử dụng câu lệnh DELETE để xóa những không dữ liệu cần thiết trong DATABASE 22](#_Toc153899197)

[a. Xóa những SinhVien đã bắt đầu học từ 8 năm về trước 22](#_Toc153899198)

[b. Xóa những lớp học phần có số lượng sinh Viên đăng kí <= 0.5 sô lượng đăng kí tối đa và soLuongDDK = 0; 22](#_Toc153899199)

[c. Xóa những lớp học phần không có maHP thuộc mã học phần quy đinh. 22](#_Toc153899200)

[d. Xóa tất cả lớp học phần từ bảng LopHocPhan mà không có phòng học tương ứng. 22](#_Toc153899201)

[e. Xóa tất cả sinh viên từ bảng sinhVien mà chưa đăng kí lớp học phần nào. 22](#_Toc153899202)

[f. Xóa tất cả khoa từ bảng Khoa mà không có sinh viên nào thuộc khoa đó. 22](#_Toc153899203)

[g. Xóa tất cả những bản ghi của bảng DangKiChiTiet mà có thời gian DK >’16:00:00’ 22](#_Toc153899204)

[h. Xóa tất cả lớp học phần từ bảng LopHocPhan mà chưa có sinh viên nào đăng kí. 22](#_Toc153899205)

[i. Xóa tất cả phòng từ bảng LopHP\_Phong mà tên phòng không có tí tự ‘Phòng’ 22](#_Toc153899206)

[8. Sử dụng câu lệnh SELECT để truy vấn dữ liệu 22](#_Toc153899207)

[a. Hiển thị các lớp thuộc khoa ‘Công nghệ thông tin’ 22](#_Toc153899208)

[b. Hiển thị mã sinh viên, họ và tên, ngày sinh, năm bắt đầu học của các sinh viên 23](#_Toc153899209)

[c. Hiển thị thông tin của sinh viên có mã sinh viên ‘SV001’ 23](#_Toc153899210)

[d. Hiển thị mã sinh viên, tên sinh viên, maDKCT, Tổng số tín chỉ của sinh viên có tổng số tín chỉ đăng kí nhỏ hơn 10 23](#_Toc153899211)

[e. Hiển thị thông tin của những sinh viên, không tham gia đăng kí tín chỉ trong học kì 123 23](#_Toc153899212)

[f. Hiển thị thông tin thông tin đăng kí học phần gồm mã sinh viên, tên sinh viên, maHP đăng kí, maLHP đăng kí, tenHP đăng kí của sinh viên có mã sinh viên ‘SV001’ 23](#_Toc153899213)

[g. Hiển thị các Học phần của các chương trình đào tạo tương ứng, bao gồm,maHP,tenHP,soTC,maChuongTrinhDaoTao,tenChuongTrinhDaoTao 24](#_Toc153899214)

[h. Cho biết chương trình đào tạo 1 có những học phần nào. 24](#_Toc153899215)

[i. Hiển thị mã sinh viên , tên sinh viên, ngày sinh và địa chỉ cửa những sinh viên thuộc lớp ‘LopSH1’ 24](#_Toc153899216)

[j. Hiển thị những sinh viên đã đăng kí lớp học phần có mã lớp học phần ‘LHP02’ 24](#_Toc153899217)

[k. Hiển thị số lượng sinh viên đã đăng kí lớp học phần của tất cả các lớp học phần trong học kì 123, bao gồm maLHP, maHP, tenHP, soLuongDDK, soLuongĐK, maHocKi 24](#_Toc153899218)

[l. Hiển thị thời gian đăng kí của sinh viên có mã sinh viên là ‘SV001’, thông tin hiển thị bao gồm, mã sinh viên, tên sinh viên, email sinh viên, maLHP đăng kí, thời gian đăng kí. 24](#_Toc153899219)

[m. Hãy cho biết những giáo viên nào không dạy lớp nào trong học kì 123. 25](#_Toc153899220)

[n. Hãy hiển thời gian giảng dạy của lớp học phần có mã ‘LHP001’ gồm, maLHP, maThoiGian,Thu,tietBD, tietKT 25](#_Toc153899221)

[o. Hãy hiển thị thông tin của phòng học đã có lớp giảng dạy vào thứ 2 tiết 1 – tiết 3, gồm maLHP, maPhong, tenPhong. 25](#_Toc153899222)

[r. Hiển thị những học phần không có trong chương trình đào tạo nào 25](#_Toc153899223)

[s. Hiển thị những giáo viên dạy từ 2 lớp học phần trở lên 25](#_Toc153899224)

[t. Hiển thị những lớp học phần nào học vào thứ 2 tiết BD là tiết 1 và tiết kết thúc là tiết 3 26](#_Toc153899225)

[u. Hiển thị danh sách sinh viên đã đăng kí học phần và lớp học phần tương ứng 26](#_Toc153899226)

[v. Hãy in ra thông tin sinh viên gồm mã sinh viên , tên sinh viên, và tổng số môn học mà sinh viên đó đã đăng kí trong học kì 123 26](#_Toc153899227)

[w. Hãy hiển thị các lớp sinh hoạt và tổng số sinh viên trong lớp sinh hoạt đó 26](#_Toc153899228)

[x. Hãy hiện thị các lớp sinh hoạt có tổng số sinh viên là 5 trở lên 26](#_Toc153899229)

[y. Hãy hiển thị thông tin các sinh viên nữ thuộc khoa công nghệ thông tin 26](#_Toc153899230)

[z. Hãy hiện thị mã khoa, tên khoa và số lượng sinh viên của mỗi khoa 27](#_Toc153899231)

[9. Sử dụng CREATE VIEW để tạo VIEW thích hợp. 27](#_Toc153899232)

[a. Tạo view chứa các sinh viên năm 1 (năm BD 2023) 27](#_Toc153899233)

[b. Tạo view chứa các sinh viên đã đăng kí học phần kì 123 27](#_Toc153899234)

[c. Tạo View chứa tất cả các học phần không nằm trong chương trình đào tạo nào 27](#_Toc153899235)

[d. Tạo View chứa danh sách lớp học phần nào vào thứ 2 tiết 3-4 27](#_Toc153899236)

[e. Tạo View chứa tất cả các sinh viên nữ của khoa công nghệ thông tin có maKhoa =’KHOA001’ 28](#_Toc153899237)

[f. Tạo View chứa số lượng sinh viên của mỗi khoa 28](#_Toc153899238)

[g. Tạo view chứa danh sách và số lượng sinh viên của mỗi lớp , lớp đó thuộc khoa nào? 28](#_Toc153899239)

[h. Tạo view chứa danh sách sinh viên đã đăng kí “Học phần 1” của “chương trình đào tạo 1” 28](#_Toc153899240)

[10. Từ những View đã tạo thực hiện 1 số câu truy vấn từ view trên 28](#_Toc153899241)

[a. Hiển thị thông tin của tất cả sinh viên khóa 23 có giới tính là nam(M) 28](#_Toc153899242)

[b. Hiển thị thông tin của tất cả sinh viên khóa 23 có giới tính là nữ và có mã lớp sinh hoạt là 'LSH001' 28](#_Toc153899243)

[c. Hiển thị Sinh viên nũ của khoa công nghệ thông tin 29](#_Toc153899244)

[d. Hiển thị những sinh viên khóa 23 và có đăng kí học phần 123 29](#_Toc153899245)

[e. Danh sách sinh viên của các lớp có nhiều hơn 3 sinh viên 29](#_Toc153899246)

[f. Hiển thị những sinh viên đã đang kí học phần của lớp SH có maLSH= 'LSH001' và có giới tính nữ(F) 29](#_Toc153899247)

[g. Danh sách sinh viên nữ thuộc Khoa Công nghệ thông tin và không tham gia đăng ký học phần kỳ 123: 29](#_Toc153899248)

[h. Danh sách các lớp( gồm maLSH và tenLSP) thuộc khóa K23 có sinh viên không tham gia dăng kí học phần 123 29](#_Toc153899249)

[11. Sử dụng câu lệnh IF\_ELSE và CASE trong SQL 29](#_Toc153899250)

[a. Kiểm tra xem có sinh viên nào có giới tính nam hay không nếu có thì in ra ‘Co sinh vien nam trong danh sách’ ngược lại in ra ‘Khong co sinh vien nam trong danh sach’. 29](#_Toc153899251)

[b. Câu truy vấn lấy thông tin về lớp sinh hoạt và hiển thị tên mới của các lớp có mã 'LSH003', 'LSH004', 'LSH005' 29](#_Toc153899252)

[c. Kiểm tra xem có tồn tại Giáo viên có mã GV=’GV001’ hay không nếu tồn tại thì in ra ‘Giáo viên có ma’ +maGV+’tồn tại’, ngược lại in ra ‘Không có giáo viên nào có mã’+maGV+ ‘ .’ 30](#_Toc153899253)

[d. Kiểm tra sinh viên có mã sinh viên SV001 có đăng kí môn học nào đó hay không nếu sinh viên có đăng kí môn học nào đó thì in ra’Sinh viên có mã’ + maSV+ ‘đã đăng kí môn học.’, ngược lại thì in ra ‘Sinh viên có mã’ +maSV + ‘ chưa đăng kí môn học 30](#_Toc153899254)

[e. Kiểm tra xem lớp học phần có maLHP ‘LHP001’ có soLuongDDK đạt 50% soLuongDK hay chưa , nêu đạt hãy in ra ‘Lớp học phần’ + maLHP+ ‘đã đạt đến 50% số Lượng đăng kí.’, ngược lại nếu chưa đạt hãy in ra ‘Lớp học phần’ + maLHP+ ‘chưa đạt đến 50% số Lượng đăng kí.’ 30](#_Toc153899255)

[f. Sử dụng câu lệnh CASE kiểm tra xem tất cả các lớp học phần đã có soLuongDDK đạt 50% của soLuongDK hay chưa, thêm cột trạng thái ở câu SELECT 30](#_Toc153899256)

[h. Câu truy vấn cập nhật tên lớp học phần có mã 'LHP001' thành 'Lớp Tin Học Cơ Sở' nếu số lượng đăng ký lớn hơn , 50 ngược lại giữ nguyên, Sử dụng CASE. 31](#_Toc153899257)

[i. Câu truy vấn cập nhật tên giáo viên có mã 'GV003' thành 'Nguyễn Minh Cường' nếu tên giáo viên là 'Hồ Minh C', ngược lại giữ nguyên, sử dụng CASE . 31](#_Toc153899258)

[j. Câu truy vấn lấy thông tin về sinh viên và hiển thị giới tính dưới dạng chuỗi 'Nam' hoặc 'Nữ', Sử dụng CASE để thêm một cột mới có tên là GioiTinh 31](#_Toc153899259)

[k. Câu truy vấn kiểm tra và hiển thị thông tin về sinh viên có tuổi từ 19 trở lên, sử dụng CASE để thêm một cột mới có tên là tình trạng tuổi 31](#_Toc153899260)

[l. Câu truy vấn lấy thông tin về lớp học phần và hiển thị 'Đã đủ đăng ký'nếu số lượng đăng ký bằng số lượng đăng ký đủ, ngược lại 'Chưa đủ đăng ký', sử dụng CASE và thêm một cột mới tên là TinhTrangDangKy. 31](#_Toc153899261)

[m. Câu truy vấn lấy thông tin về thời gian và hiển thị 'Buổi sáng' nếu thời gian nằm trong khoảng buổi sáng, ngược lại 'Buổi chiều', sử dụng CASE và thêm một cột mới tên là thoiGian 32](#_Toc153899262)

[n. Câu truy vấn lấy thông tin về sinh viên và hiển thị 'Email hợp lệ' nếu email có định dạng đúng, ngược lại 'Email không hợp lệ', sử dụng CASE và thêm một cột mới tên là TinhTrangEmail. 32](#_Toc153899263)

[12. Sử dụng câu lệnh UNION 32](#_Toc153899264)

[a. Truy vấn dữ liệu sử dụng UNION hãy truy vấn những sinh viên nào có tổng số đăng kí tín chỉ từ 5 đến 10 hoặc đăng kí lớp học phần có mã học phần maHP= ‘HP001’ 32](#_Toc153899265)

[b. Hãy truy vấn dữ liệu sử dụng UNION , những sinh viên nào có email kết thúc bằng ‘@gmail.com’ hoặc có số điện thoại bắt đầu bằng ‘09’ 32](#_Toc153899266)

[c. Hãy truy vấn dữ liệu sử dụng UNION , những sinh viên có giới tính là ‘Nam’ hoặc có tháng sinh từ tháng 1 đến tháng 6. 33](#_Toc153899267)

[d. Liệt kê tên, loại (sinh viên/giáo viên), và ngày đăng ký của tất cả các đăng ký môn học có sử dụng UNION 33](#_Toc153899268)

[e. Sử dụng câu lệnh UNION để thực hiện Liệt kê tất cả các lớp học phần và thời gian học tương ứng 33](#_Toc153899269)

[f. Truy vấn có sử dụng UNION liệt kê tất cả các môn học và số lượng sinh viên đã đăng ký 33](#_Toc153899270)

[13. Tạo những FUNCTION có thể có trong Database này : 33](#_Toc153899271)

[a. Tạo FUNCTION lấy tên khoa theo mã khoa 33](#_Toc153899272)

[b. Tạo FUNCTION lấy tên lớp Sinh Hoạt của một sinh viên với tham số đầu vào là maSV 34](#_Toc153899273)

[c. Tạo FUNCTION lấy tên giáo viên dựa vào mã giáo viên được truyền vào. 34](#_Toc153899274)

[d. Tạo FUNCTION lấy số lượng học phần mà sinh viên đã đăng kí với mã sinh viên là tham số được truyền vào . 34](#_Toc153899275)

[e. Tạo FUNCTION lấy số tên học phần theo mã học phần được truyền vào làm tham số 34](#_Toc153899276)

[f. Tạo FUNCTION lấy thôn tên thời gian gồm (Thứ -tietBD-tietKT) 35](#_Toc153899277)

[g. Tạo FUNCTION lấy tên phòng học theo mã phòng học được truyền vào. 35](#_Toc153899278)

[h. Tạo FUCNTION đếm số lượng đăng kí chi tiết trong một lớp học phần 35](#_Toc153899279)

[i. Tạo FUNCTION check điều kiện đủ tuổi của một sinh viên với mã sinh viên là tham số được truyền vào 35](#_Toc153899280)

[j. Tạo FUNCTION Kiểm Tra Số Lượng Đăng Ký Đã Đạt Mức Tối Đa Theo Lớp Học Phần 36](#_Toc153899281)

[k. Tạo FUNCTION kiểm tra lớp học phần đã có giáo viên dạy hay chưa 36](#_Toc153899282)

[14. Tạo PROCEDURE có thể có trong DATABASE 36](#_Toc153899283)

[14.1 Có sử dụng FUNCTION đã tạo ở bên trên trong PROCEDURE 36](#_Toc153899284)

[a. Tạo PROCEDURE lấy tên khoa theo mã khoa được truyền vào 36](#_Toc153899285)

[b. Tạo PROCEDURE kiểm tra số lượng đăng kí đã đạt mức tối đa theo lớp học phần 37](#_Toc153899286)

[c. Lấy lớp sinh hoạt của sinh viên dựa vào mã sinh viên truyền vào 37](#_Toc153899287)

[d. Tạo PROCEDURE lấy tên giáo viên thông qua maGiaoVien 37](#_Toc153899288)

[e. Tạo PROCEDURE lấy số lượng học phần mà sinh viên đã đăng kí thông qua mã sinh viên 37](#_Toc153899289)

[14.2 Tạo PROCEDURE không sử dụng FUNCTION đã tạo ở trên. 38](#_Toc153899290)

[f. Tạo PROCEDURE lấy tất cả sinh viên trong một lớp cụ thể 38](#_Toc153899291)

[g. Tạo PROCEDURE lấy lịch học của một sinh viên cụ thể thông qua mã sinh viên 38](#_Toc153899292)

[h. Tạo PROCEDURE cập nhật thông tin sinh viên 38](#_Toc153899293)

[i. Tạo PROCEDURE lấy thông tin chi tiết của một lớp học phần cụ thể 38](#_Toc153899294)

[j. Tạo PROCEDURE Lấy chi tiết chương trình đào tạo cho một mã chương trình cụ thể 39](#_Toc153899295)

[k. Tạo PROCEDURE lấy danh sách môn học sinh viên đã đăng kí theo maSV và maHocKi 39](#_Toc153899296)

[l. Tạo PROCEDURE INSERT 1 dòng dữ liệu vao bảng LopHocPhan đảm bảo không trùng khóa chính, thực hiện kiểm tra tính hợp lệ của dữ liệu , đảm bảo toàn vẹn tham chiếu . 39](#_Toc153899297)

[15. Tạo TRIGGER cho CƠ SỞ DỮ LIỆU 40](#_Toc153899298)

[a. Tạo Trigger cho bảng LopHocPhan thực hiện yêu cầu INSERT, DELETE, UPDATE như sau : 40](#_Toc153899299)

[b. Tạo trigger cho bảng LopHocPhan, trường hợp một lớp học phần không đủ số lượng sinh viên đăng kí bắt buộc phòng đào tạo phải thực hiện xóa lớp đó đi thì mình phải cập nhật những lớp học phần mà sinh viên đã đăng kí. 41](#_Toc153899300)

[c. Tạo Trigger kiểm tra khi sinh viên thực hiên đăng kí thêm một học phần vào bảng DangKiChiTiet, trigger có nhiệm vụ kiểm ra xem Lớp học phần vừa được thêm vào có trùng thời gian với lớp học phần đã đăng kí hay không nếu có thì thông báo cho người dung và thực hiện rollback , ngược lại nếu thỏa mãn không trùng thời gian thì cho phép insert dữ liệu vào bảng: 42](#_Toc153899301)

# Sơ đồ ERD

## 1.1 Ảnh ERD về đăng kí tín chỉ trực tuyến trường ĐHSPKT-ĐN.

A computer screen shot of a computer

Description automatically generated

# Bổ sung kiểu dữ liệu và ràng buộc cho các bảng từ sơ đồ ERD trên theo mẫu dưới đây:

## Tên bảng: chuongTrinhDaoTao

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maChuongTrinhDaoTao | char(20) | PRIMARY KEY | Mã chương trình đào tạo |
| tenChuongTrinhDaoTao | nvarchar(255) | NOT NULL | Tên chương trình đào tạo |

## Tên bảng: hocPhan

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maHP | char(20) | PRIMARY KEY | Mã học phần |
| tenHP | nvarchar(255) | NOT NULL | Tên học phần |
| soTC | int | NOT NULL | Số tín chỉ |

## Tên bảng: chuongTrinhDaoTao\_HocPhan

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maChuongTrinhDaoTao | Char(20) | PRIMARY KEY,  FOREIGN KEY | Mã chương trình |
| maHP | Char(20) | PRIMARY KEY,  FOREIGN KEY | Mã học phần |

## Tên bảng: giaoVien

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maGV | Char(15) | PRIMARY KEY | Mã giáo viên( đồng thời cũng là tên user khi đăng nhập) |
| tenGV | Nvarchar(50) | NOT NULL | Tên giáo viên |
| gioiTinh | Nvarchar(3) | ‘F’ OR ‘M’ ,DEFAULT ‘F’ | Giới tính |
| SDT | Char(10) | 10 số | Số điện thoại |
| email | Varchar(255) | Unique, kiểm tra đó phải là email. | email |
| matKhau | Varchar(255) | NOT NULL | Mật khẩu tài khoản đăng nhập |

## Tên bảng: thoiGian

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maThoiGian | Char(10) | PRIMARY KEY | Mã thời gian |
| Thu | Char(5) | NOT NULL | Thứ( thứ 2- thứ 6) |
| tietBD | int | NOT NULL | Tiết bắt đầu |
| tietKT | int | NOT NULL | Tiết kết thúc |

## Tên bảng: Phong

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maPhong | Char(10) | PRIMARY KEY | Mã phòng |
| tenPhong | Varchar(20) | NOT NULL | Tên phòng |

## Tên bảng: ThoiGian\_Phong

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maTG\_PH | Char(15) | PRIMARY KEY | Mã thời gian – phòng lừa |
| maPhong | Char(10) | PRIMARY KEY,  FOREIGN KEY | Mã phòng, (tham chiếu tới khóa chính của bảng Phong) |
| maThoiGian | Char(10) | PRIMARY KEY,  FOREIGN KEY | Mã thời gian,(tham chiếu tới khóa chính của bảng ThoiGian) |

## Tên bảng: LopHocPhan

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maLHP | Char(15) | PRIMARY KEY | Mã lớp học phần |
| maHP | Char(20) | FOREIGN KEY, | Mã học phần |
| maGV | Char(15) | FOREIGN KEY | Mã giáo viên |
| soLuongDK | int | NOT NULL | Số lượng đăng kí tối đa |
| soLuongDDK | int | NOT NULL | Số lượng đã đăng kí |
| maTG\_PH | Char(10) | FOREIGN KEY | Mã thời gian-phòng học |
| maHocKi | Char(5) | NOT NULL | Mã học kì |
| ghiChu | Nvarchar(255) | NULL | Ghi chú |

## Tên bảng: Khoa

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maKhoa | Char(10) | PRIMARY KEY | Mã khoa |
| tenKhoa | Nvarchar(50) | NOT NULL | Tên khoa |

## Tên bảng: LopSH

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maLSH | Char(10) | PRIMARY KEY | Mã lớp sinh hoạt |
| maKhoa | Char(10) | FOREIGN KEY | Mã khoa |
| tenLSH | Nvarchar(50) | NOT NULL | Tên lớp sinh hoạt |
| namBD | int | NOT NULL | Năm bắt đầu học |

## Tên bảng: SinhVien

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maSV | Char(15) | PRIMARY KEY | Mã sinh viên(đồng thời cũng là tên user khi đăng nhập) |
| maLSH | Char(10) | FOREIGN KEY | Mã lớp sinh hoạt |
| tenSV | Nvarchar(100) | NOT NULL | Tên sinh viên |
| gioiTinh | char(1) | ‘F’ OR ‘M’ ,DEFAULT ‘F’ | Giới tính |
| ngaySinh | Date | <=ngày hiện tại | Ngày sinh |
| diaChi | Nvarchar(255) | NULL | Địa chỉ |
| email | Varchar(100) | UNIQUE, NOT NULL | email |
| SDT | Char(10) | Check(10 số) | Số điện thoại |
| matKhau | Varchar(255) | NOT NULL | Mật khẩu |

## Tên bảng: DangKiTinChi

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maDK | Char(15) | PRIMARY KEY | Mã đăng kí |
| maSV | Char(15) | FOREIGN KEY | Mã sinh viên |
| thoiGianDK | Time | NOT NULL; | Thời gian đăng kí |

## Tên bảng: DangKiChiTiet

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| maDKCT | Char(10) | PRIMARY KEY | Mã đăng kí chi tiết |
| maLHP | Char(15) | PRIMARY KEY | Mã lớp học phần |
| maDK | Char(15) | FOREIGN KEY | Mã đăng kí |

# Sử dụng câu lệnh CREATE TABLE để tạo các bảng trong cơ sở dữ liệu từ ERD với các trường, kiểu dữ liệu và ràng buộc từ câu 1.

CREATE DATABASE BTCANHAN

GO

USE BTCANHAN;

CREATE TABLE Khoa (

maKhoa CHAR(10) PRIMARY KEY,

tenKhoa NVARCHAR(50) NOT NULL

);

CREATE TABLE LopSH (

maLSH CHAR(10) PRIMARY KEY,

maKhoa CHAR(10),

tenLSH NVARCHAR(50),

namBD INT,

FOREIGN KEY (maKhoa) REFERENCES Khoa (maKhoa)

);

CREATE TABLE GiaoVien (

maGV CHAR(15) PRIMARY KEY,

tenGV NVARCHAR(255),

gioiTinh NVARCHAR(3),

SDT CHAR(10),

email VARCHAR(255) UNIQUE,

matkhau VARCHAR(255)

);

CREATE TABLE SinhVien (

maSV CHAR(15) PRIMARY KEY,

maLSH CHAR(10),

tenSV NVARCHAR(100),

gioiTinh NVARCHAR(3),

ngaySinh DATE,

diaChi NVARCHAR(255),

email VARCHAR(100) UNIQUE,

matkhau VARCHAR(255) ,

FOREIGN KEY (maLSH) REFERENCES lopSH(maLSH)

);

CREATE TABLE ChuongTrinhDaoTao (

maChuongTrinhDaoTao CHAR(20) PRIMARY KEY,

tenChuongTrinhDaoTao NVARCHAR(255)

);

CREATE TABLE HocPhan (

maHP CHAR(20) PRIMARY KEY,

tenHP NVARCHAR(255),

soTC INT

);

CREATE TABLE ChuongTrinhDaoTao\_HocPhan (

maChuongTrinhDaoTao CHAR(20),

maHP CHAR(20),

PRIMARY KEY (maChuongTrinhDaoTao, maHP),

FOREIGN KEY (maChuongTrinhDaoTao) REFERENCES ChuongTrinhDaoTao (maChuongTrinhDaoTao),

FOREIGN KEY (maHP) REFERENCES HocPhan (maHP)

);

CREATE TABLE LopHocPhan (

maLHP CHAR(15) PRIMARY KEY,

maHP CHAR(20),

maGV CHAR(15),

soLuongDK INT,

soLuongDDK INT,

ghiChu NVARCHAR(255),

FOREIGN KEY (maHP) REFERENCES HocPhan (maHP),

FOREIGN KEY (maGV) REFERENCES GiaoVien (maGV)

);

CREATE TABLE LopHP\_Phong (

maLHP CHAR(15),

maPhong CHAR(10),

tenPhong nvarchar(20),

PRIMARY KEY (maLHP, maPhong),

FOREIGN KEY (maLHP) REFERENCES LopHocPhan (maLHP)

);

CREATE TABLE DangKiTinChi (

maDK CHAR(15) PRIMARY KEY,

maSV CHAR(15),

thoiGianDK TIME,

FOREIGN KEY (maSV) REFERENCES SinhVien (maSV)

);

CREATE TABLE DangKiChiTiet (

maDKCT CHAR(10),

maLHP CHAR(15),

maDK CHAR(15),

PRIMARY KEY (maDKCT, maLHP),

FOREIGN KEY (maLHP) REFERENCES LopHocPhan (maLHP),

FOREIGN KEY (maDK) REFERENCES DangKiTinChi (maDK)

);

CREATE TABLE ThoiGian\_LopHP (

maLHP CHAR(15),

maThoiGian CHAR(10),

Thu NVARCHAR(5),

tietBD INT,

tietKT INT,

PRIMARY KEY (maLHP, maThoiGian),

FOREIGN KEY (maLHP) REFERENCES LopHocPhan (maLHP),

);

# Sử dụng câu lệnh ALTER TABLE

* 1. Bảng SinhVien, thay đổi ngày sinh của cột ngaySinh thành Datetime, thêm ràng buộc cho ngày sinh phải lớn hơn hoặc = 18 tuổi
* thay đổi ngày sinh của côt ngaySinh thành Datetime

ALTER TABLE SinhVien

ALTER column ngaySinh DATETIME

* Thêm rang buộc check cho cột ngaySinh phải lớn hoặc = 18 tuổi

ALTER TABLE SinhVien

ADD CONSTRAINT CK\_SinhVien\_ngaySinh

CHECK(DATEDIFF(year, '0:0', getDate()- ngaySinh)>=18);

* 1. Thay đổi kiểu dữ liệu cho cột gioiTinh bảng SinhVien và GiaoVien

-Bảng SinhVien

ALTER TABLE SinhVien

ALTER COLUMN gioiTinh char(1);

* Bảng GiaoVien

ALTER TABLE GiaoVien

ALTER column gioiTinh char(1);

* 1. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho côt maSV của Bảng DangKiTinChi

ALTER TABLE DangKiTinChi

ALTER COLUMN maSV char(15) NOT NULL

* 1. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột tenGV và email của bảng GiaoVien

ALTER TABLE GiaoVien

ALTER COLUMN tenGV nvarchar(255) NOT NULL

ALTER TABLE GiaoVien

ALTER COLUMN email varchar(255) NOT NULL

* 1. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột tenHP và soTC của bảng HocPhan

ALTER TABLE HocPhan

ALTER COLUMN tenHP nvarchar(255) NOT NULL

ALTER TABLE HocPhan

ALTER COLUMN soTC int NOT NULL;

* 1. Sử dụng ALTER TABLE để thêm ràng buộc NOT NULL cho các cột maHP, maGV, soLuongDK, maTG\_HP, maHocKi của bảng LopHocPhan

ALTER TABLE LopHocPhan

ALTER COLUMN maHP char(20) NOT NULL

ALTER TABLE LopHocPhan

ALTER COLUMN soLuongDK int NOT NULL

ALTER TABLE LopHocPhan

ALTER COLUMN maGV char(15) NOT NULL

ALTER TABLE LopHocPhan

ALTER COLUMN maTG\_HP char(15) NOT NULL

ALTER TABLE LopHocPhan

ALTER COLUMN maHocKi char(5) NOT NULL

* 1. Sử dụng ALTER TABLE để để thêm ràng buộc NOT NULL cho cột maThoiGian và maPhong của bảng ThoiGian\_PhongHoc

ALTER TABLE ThoiGian\_PhongHoc

ALTER COLUMN maPhong char(10) NOT NULL

ALTER TABLE ThoiGian\_PhongHoc

ALTER COLUMN maThoiGian char(10) NOT NULL

* 1. Thêm ràng buộc cho cột gioiTinh ở bảng SinhVien và GiaoVien
* Bảng SinhVien

ALTER TABLE SinhVien

ADD CONSTRAINT CK\_SinhVien\_GT

CHECK(gioiTinh Like 'F' OR gioiTinh like 'M'),

CONSTRAINT DF\_SinhVien\_GT Default 'F' for gioiTinh;

* Bảng GiaoVien

ALTER TABLE GiaoVien

ADD CONSTRAINT CK\_GiaoVien\_GT

CHECK(gioiTinh Like 'F' OR gioiTinh like 'M'),

CONSTRAINT DF\_GiaoVien\_GT Default 'F' for gioiTinh;

* 1. Thêm ràng buộc cho cột CHECK SDT ở bảng SinhVien và bảng GiaoVien
* Bảng SinhVien

ALTER TABLE SinhVien ADD CONSTRAINT CK\_SinhVien\_SDT

CHECK(SDT Like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');

* Bảng GiaoVien

ALTER TABLE GiaoVien ADD CONSTRAINT CK\_GiaoVien\_SDT

CHECK(SDT Like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');

* 1. Thêm ràng buộc CHECK email ở bảng SinhVien và bảng GiaoVien
* Bảng SinhVien

ALTER TABLE SinhVien

ADD CONSTRAINT CK\_SinhVien\_email

CHECK(email Like '[a-z]%@gmail.com');

* Bảng GiaoVien

ALTER TABLE GiaoVien

ADD CONSTRAINT CK\_GiaoVien\_email

CHECK(email Like '[a-z][%@gmail.com](mailto:%25@gmail.com)');

* 1. Thêm ràng buộc NOT NULL cho cột matKhau ở bảng SinhVien và GiaoVien

-Bảng SinhVien:

ALTER TABLE GiaoVien

ALTER column matKhau varchar(255) NOT NULL;

-Bảng GiaoVien:

ALTER TABLE SinhVien

ALTER column matKhau varchar(255) NOT NULL;

* 1. Thêm ràng buộc cho SoTC ở bảng HocPhan phải lớn hơn 0

ALTER TABLE HocPhan

ADD CONSTRAINT CK\_HocPhan\_SoTC

CHECK(SoTC>0);

* 1. Thêm ràng buộc cho soLuongDK và soLuongDDK phải lớn hơn 0, soLuongDK<=soLuongDDK ở bảng LopHocPhan
* Check (soLuongDK>=0)

ALTER TABLE LopHocPhan

ADD CONSTRAINT CK\_LopHocPhan\_SoLDK

CHECK(soLuongDK>0);

* Check(soLuongDDK>0)

ALTER TABLE LopHocPhan

ADD CONSTRAINT CK\_LopHocPhan\_SoLDDK

CHECK(soLuongDDK>0);

* Check(soLuongDDK<=soLuongDK):

ALTER TABLE LopHocPhan

ADD CONSTRAINT CK\_LopHocPhan\_SoLDDK\_soLDK

CHECK(soLuongDDK<soLuongDK);

* 1. Thêm rang buộc cho tietBD và tietKT phải thuộc từ tiết 1đến tiết 15 , và tietBD<=tietKTcho bảng ThoiGian\_LHP
* Ràng buộc cho tiết bắt đầu(tietBD)

ALTER TABLE ThoiGian

ADD CONSTRAINT CK\_TGLHP\_tietBD

CHECK(tietBD>0 and tietBD<15);

* Ràng buộc cho tiết kết thưc(tietKT)

ALTER TABLE ThoiGian

ADD CONSTRAINT CK\_TGLHP\_tietKT

CHECK(tietKT>0 and tietBD<15);

* Ràng buộc cho tietKT>=tietBD

ALTER TABLE ThoiGian

ADD CONSTRAINT CK\_TGLHP\_tietKT\_tietBD

CHECK(tietKT>=tietBD);

* 1. Thêm ràng buộc cho namBD ở bảng lopSH phải <=năm hiện tại

ALTER TABLE LopSH

ADD CONSTRAINT CK\_LopSH\_namBD

CHECK(namBD<getDate());

* 1. Thay đổi khóa chỉnh của bảng ChiTietDangKi thành hai cột maDKCT va maDK vì maDK là mã đăng kí chính thức của sinh viên quản lí các maDKCT con của từng lớp học phần mà sinh viên đăng kí (hay nói cách khác là một maDK có thể đăng kí nhiều maDKCT con)

ALTER TABLE DangKiChiTiet

DROP CONSTRAINT [PK\_\_DangKiCh\_\_361DBC101107CFE1];

ALTER TABLE DangKiChiTiet

ADD CONSTRAINT PK\_DangKiChiTiet PRIMARY KEY (maDKCT, maDK);

* 1. Sửa đổi cột maLHP và maDK của bảng ChiTietDangKi thành not null vì nếu có một hàng null sẽ không đăng kí được môn học

ALTER TABLE DangKiChiTiet

ALTER COLUMN maLHP char(15) not null

ALTER TABLE DangKiChiTiet

ALTER COLUMN maDK char(15) not null

# Sử dụng câu lệnh INSERT để thêm dữ liệu cho các bảng

-- INSERT DỮ LIỆU

-- Insert data into Khoa

INSERT INTO Khoa (maKhoa, tenKhoa)

VALUES

('KHOA001', N'Khoa Học Máy Tính'),

('KHOA002', N'Khoa Kinh Tế'),

('KHOA003', N'Khoa Sư Phạm'),

('KHOA004', N'Khoa Y Học'),

('KHOA005', N'Khoa Khoa Học Xã Hội');

-- Insert data into LopSH

INSERT INTO LopSH (maLSH, maKhoa, tenLSH, namBD)

VALUES

('LSH001', 'KHOA001', N'Lớp SH1', 2023),

('LSH002', 'KHOA001', N'Lớp SH2', 2023),

('LSH003', 'KHOA002', N'Lớp SH1', 2023),

('LSH004', 'KHOA002', N'Lớp SH2', 2023),

('LSH005', 'KHOA003', N'Lớp SH1', 2023);

-- Insert data into SinhVien

INSERT INTO SinhVien (maSV, maLSH, tenSV, gioiTinh, ngaySinh, diaChi, email, SDT, matkhau)

VALUES

('SV001', 'LSH001', N'Nguyễn Thị X', 'M', '2000-01-01', N'123 Đường ABC', 'nguyen.thi.x@gmail.com', '0123456789', 'matkhau1'),

('SV002', 'LSH002', N'Trần Văn Y', 'F', '2001-02-02', N'456 Đường XYZ', 'tran.van.y@gmail.com', '0987654321', 'matkhau2'),

('SV003', 'LSH003', N'Hồ Thị Z', 'F', '2002-03-03', N'789 Đường DEF', 'ho.thi.z@gmail.com', '0369258741', 'matkhau3'),

('SV004', 'LSH004', N'Lê Văn W', 'M', '2003-04-04', N'987 Đường LMN', 'le.van.w@gmail.com', '0897456321', 'matkhau4'),

('SV005', 'LSH005', N'Phạm Thị U', 'M', '2004-05-05', N'741 Đường UVW', 'pham.thi.u@gmail.com', '0321456987', 'matkhau5');

-- Insert data into GiaoVien

INSERT INTO GiaoVien (maGV, tenGV, gioiTinh, SDT, email, matkhau)

VALUES

('GV001', N'Nguyễn Văn A', 'F', '0123456789', 'nguyen.va@gmail.com', 'matkhau1'),

('GV002', N'Trần Thị B', 'F', '0987654321', 'tran.thi.b@gmail.com', 'matkhau2'),

('GV003', N'Hồ Minh C', 'M', '0369258741', 'ho.minh.c@gmail.com', 'matkhau3'),

('GV004', N'Lê Thị D', 'M', '0897456321', 'le.thi.d@gmail.com', 'matkhau4'),

('GV005', N'Phạm Văn E', 'F', '0321456987', 'pham.van.e@gmail.com', 'matkhau5'),

('GV006', N'Nguyễn Văn Hải', 'M', '0123416759', 'nguyenr.va@gmail.com', 'matkhau6'),

('GV007', N'Nguyễn Thủy Tiên', 'F', '0122416759', 'levan.va@gmail.com', 'matkhau7');

-- Insert data into ThoiGian

INSERT INTO ThoiGian (maThoiGian, Thu, tietBD, tietKT)

VALUES

('TG001', N'Thứ 2', 1, 3),

('TG0011',N'Thứ 2', 3, 5),

('TG002', N'Thứ 3', 2, 4),

('TG008', N'Thứ 3', 4, 6),

('TG003', N'Thứ 4', 3, 5),

('TG004', N'Thứ 5', 4, 6),

('TG005', N'Thứ 6', 5, 7),

('TG0051',N'Thứ 6', 9, 11);

-- Insert data into Phong

INSERT INTO Phong (maPhong, tenPhong)

VALUES

('PH001', N'Phòng 1'),

('PH002', N'Phòng 2'),

('PH003', N'Phòng 3'),

('PH004', N'Phòng 4'),

('PH005', N'Phòng 5');

-- Insert data into ThoiGian\_PhongHoc

INSERT INTO ThoiGian\_PhongHoc (maTG\_PH, maPhong, maThoiGian)

VALUES

('TG\_PH001', 'PH001', 'TG001'),

('TG\_PH0011', 'PH001', 'TG0011'),

('TG\_PH0012', 'PH001', 'TG003'),

('TG\_PH0013', 'PH001', 'TG005'),

('TG\_PH0021', 'PH002', 'TG002'),

('TG\_PH002', 'PH002', 'TG004'),

('TG\_PH003', 'PH003', 'TG003'),

('TG\_PH004', 'PH004', 'TG004'),

('TG\_PH005', 'PH005', 'TG005');

-- Insert data into HocPhan

INSERT INTO HocPhan (maHP, tenHP, soTC)

VALUES

('HP001', N'Học Phần 1', 3),

('HP002', N'Học Phần 2', 4),

('HP003', N'Học Phần 3', 3),

('HP004', N'Học Phần 4', 4),

('HP005', N'Học Phần 5', 2),

('HP00CNTT', N'Học Phần Tin Học Cơ bản', 2),

('HP008', N'Học Phần Toán rời rạc', 3),

('HP00GT', N'Học Phần Giải tích', 3);

-- Insert data into ChuongTrinhDaoTao

INSERT INTO ChuongTrinhDaoTao (maChuongTrinhDaoTao, tenChuongTrinhDaoTao)

VALUES

('CTDT001', N'Chương Trình Đào Tạo 1'),

('CTDT002', N'Chương Trình Đào Tạo 2'),

('CTDT003', N'Chương Trình Đào Tạo 3'),

('CTDT004', N'Chương Trình Đào Tạo 4'),

('CTDT005', N'Chương Trình Đào Tạo 5');

-- Insert data into ChuongTrinhDaoTao\_HocPhan

INSERT INTO ChuongTrinhDaoTao\_HocPhan (maChuongTrinhDaoTao, maHP)

VALUES

('CTDT001', 'HP001'),

('CTDT001', 'HP002'),

('CTDT002', 'HP003'),

('CTDT002', 'HP004'),

('CTDT003', 'HP005'),

('CTDT001', 'HP00CNTT'),

('CTDT001', 'HP008');

-- Insert data into LopHocPhan

INSERT INTO LopHocPhan (maLHP, maHP, maGV, soLuongDK, soLuongDDK, maTG\_PH, maHocKi)

VALUES

('LHP001', 'HP001', 'GV001', 50, 45, 'TG\_PH001', 'HK001'),

('LHP002', 'HP002', 'GV002', 60, 55, 'TG\_PH002', 'HK001'),

('LHP003', 'HP003', 'GV003', 40, 35, 'TG\_PH003', 'HK001'),

('LHP004', 'HP004', 'GV004', 70, 65, 'TG\_PH004', 'HK001'),

('LHP005', 'HP005', 'GV005', 30, 25, 'TG\_PH005', 'HK001');

-- Insert data into DangKiTinChi

INSERT INTO DangKiTinChi (maDK, maSV, thoiGianDK)

VALUES

('DK001', 'SV001', '08:00:00'),

('DK002', 'SV002', '09:15:00'),

('DK003', 'SV003', '10:30:00'),

('DK004', 'SV004', '11:45:00'),

('DK005', 'SV005', '13:00:00');

-- Insert data into DangKiChiTiet

INSERT INTO DangKiChiTiet (maDKCT, maLHP, maDK)

VALUES

('DKCT001', 'LHP001', 'DK001'),

('DKCT002', 'LHP002', 'DK002'),

('DKCT003', 'LHP003', 'DK003'),

('DKCT004', 'LHP004', 'DK004'),

('DKCT005', 'LHP005', 'DK005');

# Sử dụng câu lệnh UPDATE để UPDATE những dữ liệu cần thiết trong DATABASE

* 1. Cập nhật tên khoa có mã’KHOA001’ thành tên ‘Khoa Công Nghệ Thông Tin’

UPDATE Khoa

SET tenKhoa = N'Khoa Công Nghệ Thông Tin'

WHERE maKhoa = 'KHOA001';

* 1. Cập nhật tên lớp học có mã ‘LSH001’ thành tên ‘Lớp SH 22T2’

UPDATE LopSH

SET tenLSH = N'Lớp SH 22T2'

WHERE maLSH = 'LSH001';

* 1. Cập nhật số lượng đăng kí mới cho lớp học phần có mã ‘LHP001’

UPDATE LopHocPhan

SET soLuongDDK = 48

WHERE maLHP = 'LHP001';

* 1. Cập nhật thời gian đăng kí của sinh viên có mã ‘SV004’ thành thời gian’14:15:00’

UPDATE DangKiTinChi

SET thoiGianDK = '14:15:00'

WHERE maSV = 'SV004';

* 1. Cập nhật số lượng đăng kí mới soLuongDDK=soLuongDDK-10 cho tất cả lớp học phần (LopHocPhan’)

UPDATE LopHocPhan

SET soLuongDDK = soLuongDK - 10

WHERE maHP IN (SELECT maHP FROM HocPhan );

* 1. Cập nhật học phần có số tín chỉ =4 thành số tín chỉ = 3 nếu lớp đó có mã HP là ‘HP001’

UPDATE HocPhan

SET soTC = 1 WHERE maHP ='HP001'

* 1. Cập nhật giới tính của giáo Viên có giới tính là ‘M’ đổi thành ‘F’ và ngược lại.

UPDATE GiaoVien

SET gioiTinh = CASE

WHEN gioiTinh = 'M' THEN 'F'

ELSE 'M'

END;

* 1. Cập nhật lại tên lớp sinh hoạt của những lớp có mã lớp sinh hoạt ‘LSH003’, ‘LSH004’, ‘LSH005’ lần lượt đổi lại thành ‘Lớp SH3’,’Lớp SH4’,’Lớp SH5’
* Cập nhật lại tên của lớp có mã ‘LSH003’

UPDATE LopSH

SET tenLSH=N'Lớp SH3'

WHERE maLSH='LSH003';

* Cập nhật lại tên của lớp có mã ‘LSH04’

UPDATE LopSH

SET tenLSH=N'Lớp SH4'

WHERE maLSH='LSH004';

* Cập nhật lại tên của lớp có mã ‘LSH05’

UPDATE LopSH

SET tenLSH=N'Lớp SH5'

WHERE maLSH='LSH005';

* 1. UPDATE maHocKi của bảng LopHocPhan thành’123’ cho các lớp có mã học phần ‘HP001’ , ‘HP002’, ‘HP003’, ‘HP004’ và maHocKi của lớp có mã học phần ‘HP005’ thành ‘223’

-- Cập nhật lại mã học kì của LopHocPhan ='123' cho tất cả các lớp

UPDATE LopHocPhan

SET maHocki ='123'

-- Cập nhật lại mã học kì của LopHocPhan ='223 cho những lớp có maHP ='HP005'

UPDATE LopHocPhan

SET maHocki ='223'

WHERE maHP='HP005'

# Sử dụng câu lệnh DELETE để xóa những không dữ liệu cần thiết trong DATABASE

1. Xóa những SinhVien đã bắt đầu học từ 8 năm về trước

DELETE FROM SinhVien

WHERE maLSH IN (SELECT maLSH FROM LopSH WHERE namBD + 8 < YEAR(GETDATE()));

1. Xóa những lớp học phần có số lượng sinh Viên đăng kí <= 0.5 sô lượng đăng kí tối đa và soLuongDDK = 0;

DELETE LopHocPhan

WHERE soLuongDDK<=0.5\*soLuongDK and soLuongDDK=0;

1. Xóa những lớp học phần không có maHP thuộc mã học phần quy đinh.

DELETE FROM LopHocPhan

WHERE maHP NOT IN (SELECT maHP FROM HocPhan);

1. Xóa tất cả lớp học phần từ bảng LopHocPhan mà không có phòng học tương ứng.

DELETE FROM LopHocPhan

WHERE maTG\_PH IS NULL

1. Xóa tất cả sinh viên từ bảng sinhVien mà chưa đăng kí lớp học phần nào.

DELETE FROM SinhVien

WHERE maSV NOT IN (SELECT maSV FROM DangKiTinChi);

1. Xóa tất cả khoa từ bảng Khoa mà không có sinh viên nào thuộc khoa đó.

DELETE FROM Khoa

WHERE maKhoa NOT IN (SELECT maKhoa FROM LopSH);

1. Xóa tất cả những bản ghi của bảng DangKiChiTiet mà có thời gian DK >’16:00:00’

DELETE DangKiChiTiet

WHERE maDK IN(SELECT maDK FROM DangKiTinChi WHERE thoiGianDK>'16:00:00');

1. Xóa tất cả lớp học phần từ bảng LopHocPhan mà chưa có sinh viên nào đăng kí.

DELETE FROM LopHocPhan

WHERE maLHP NOT IN (SELECT maLHP FROM DangKiChiTiet);

1. Xóa tất cả phòng từ bảng LopHP\_Phong mà tên phòng không có tí tự ‘Phòng’

DELETE FROM LopHP\_Phong

WHERE tenPhong NOT LIKE '%Phòng%';

# Sử dụng câu lệnh SELECT để truy vấn dữ liệu

* 1. Hiển thị các lớp thuộc khoa ‘Công nghệ thông tin’

SELECT tenLSH, maLSH

FROM dbo.LopSH AS l, dbo.Khoa AS k

WHERE l.maKhoa = k.maKhoa AND k.tenKhoa = N'Khoa Công nghệ thông tin';

* 1. Hiển thị mã sinh viên, họ và tên, ngày sinh, năm bắt đầu học của các sinh viên

SELECT maSV, TenSV,ngaySinh,namBD

FROM SinhVien as s, LopSH as l

WHERE s.maLSH= l.maLSH;

* 1. Hiển thị thông tin của sinh viên có mã sinh viên ‘SV001’

SELECT \* FROM dbo.SinhVien

WHERE maSV='SV001'

* 1. Hiển thị mã sinh viên, tên sinh viên, maDKCT, Tổng số tín chỉ của sinh viên có tổng số tín chỉ đăng kí nhỏ hơn 10

SELECT

SV.maSV,

SV.tenSV,

SUM(HP.soTC) AS TongSoTinChi

FROM SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

GROUP BY SV.maSV, SV.tenSV

HAVING SUM(HP.soTC) < 10;

* 1. Hiển thị thông tin của những sinh viên, không tham gia đăng kí tín chỉ trong học kì 123

SELECT

SV.maSV,SV.tenSV,SV.maLSH,SV.ngaySinh,SV.SDT,SV.email,

SV.gioiTinh,SV.diaChi

FROM SinhVien SV

WHERE

SV.maSV NOT IN (

SELECT DISTINCT SV.maSV

FROM SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

WHERE LHP.maHocKi='123'

);

* 1. Hiển thị thông tin thông tin đăng kí học phần gồm mã sinh viên, tên sinh viên, maHP đăng kí, maLHP đăng kí, tenHP đăng kí của sinh viên có mã sinh viên ‘SV001’

SELECT

SV.maSV,SV.tenSV,HP.maHP,LHP.maLHP,HP.tenHP,HP.soTC

FROM

SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

WHERE SV.maSV = 'SV001';

* 1. Hiển thị các Học phần của các chương trình đào tạo tương ứng, bao gồm,maHP,tenHP,soTC,maChuongTrinhDaoTao,tenChuongTrinhDaoTao

SELECT H.maHP, tenHP,soTC,C.maChuongTrinhDaoTao,tenChuongTrinhDaoTao

FROM dbo.HocPhan H

INNER JOIN dbo.ChuongTrinhDaoTao\_HocPhan CH ON CH.maHP=H.maHP

INNER JOIN dbo.ChuongTrinhDaoTao C ON C.maChuongTrinhDaoTao=CH.maChuongTrinhDaoTao;

* 1. Cho biết chương trình đào tạo 1 có những học phần nào.

SELECT maHP,tenHP

FROM dbo.HocPhan

WHERE maHP in (SELECT CH.maHP FROM dbo.ChuongTrinhDaoTao\_HocPhan as CH , dbo.ChuongTrinhDaoTao as C

WHERE CH.maChuongTrinhDaoTao= C.maChuongTrinhDaoTao and C.tenChuongTrinhDaoTao=N'Chương Trình Đào Tạo 1');

* 1. Hiển thị mã sinh viên , tên sinh viên, ngày sinh và địa chỉ cửa những sinh viên thuộc lớp ‘LopSH1’

SELECT maSV,tenSV,diaChi

FROM SinhVien

WHERE maLSH = (SELECT maLSH FROM dbo.LopSH WHERE tenLSH like N'Lớp SH2');

* 1. Hiển thị những sinh viên đã đăng kí lớp học phần có mã lớp học phần ‘LHP02’

SELECT SV.\*

FROM SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

WHERE LHP.maLHP = 'LHP002';

* 1. Hiển thị số lượng sinh viên đã đăng kí lớp học phần của tất cả các lớp học phần trong học kì 123, bao gồm maLHP, maHP, tenHP, soLuongDDK, soLuongĐK, maHocKi

SELECT maLHP,H.maHP,tenHP,soLuongDDK,soLuongDK,maHocKi

FROM dbo.LopHocPhan as L

INNER JOIN HocPhan as H ON H.maHP=L.maHP

WHERE maHocKi ='123'

* 1. Hiển thị thời gian đăng kí của sinh viên có mã sinh viên là ‘SV001’, thông tin hiển thị bao gồm, mã sinh viên, tên sinh viên, email sinh viên, maLHP đăng kí, thời gian đăng kí.

SELECT SV.maSV,SV.tenSV,SV.email,LHP.maLHP,DKTC.thoiGianDK

FROM dbo.SinhVien as SV

INNER JOIN dbo.DangKiTinChi as DKTC ON SV.maSV=DKTC.maSV

INNER JOIN dbo.DangKiChiTiet as DKCT ON DKCT.maDK=DKTC.maDK

INNER JOIN dbo.LopHocPhan as LHP ON DKCT.maLHP=LHP.maLHP

WHERE SV.maSV='SV001'

* 1. Hãy cho biết những giáo viên nào không dạy lớp nào trong học kì 123.

SELECT \*

FROM GiaoVien

WHERE maGV NOT IN(SELECT maGV FROM dbo.LopHocPhan WHERE maHocKi='123');

* 1. Hãy hiển thời gian giảng dạy của lớp học phần có mã ‘LHP001’ gồm, maLHP, maThoiGian,Thu,tietBD, tietKT

SELECT LopHocPhan.maLHP, ThoiGian.maThoiGian, ThoiGian.Thu, ThoiGian.tietBD, ThoiGian.tietKT

FROM LopHocPhan

JOIN ThoiGian\_PhongHoc ON LopHocPhan.maTG\_PH = ThoiGian\_PhongHoc.maTG\_PH

JOIN ThoiGian ON ThoiGian\_PhongHoc.maThoiGian = ThoiGian.maThoiGian

WHERE LopHocPhan.maLHP = 'LHP001';

* 1. Hãy hiển thị thông tin của phòng học đã có lớp giảng dạy vào thứ 2 tiết 1 – tiết 3, gồm maLHP, maPhong, tenPhong.

SELECT LopHocPhan.maLHP, ThoiGian\_PhongHoc.maPhong, Phong.tenPhong

FROM LopHocPhan

JOIN ThoiGian\_PhongHoc ON LopHocPhan.maTG\_PH = ThoiGian\_PhongHoc.maTG\_PH

JOIN Phong ON ThoiGian\_PhongHoc.maPhong = Phong.maPhong

JOIN ThoiGian ON ThoiGian\_PhongHoc.maThoiGian = ThoiGian.maThoiGian

WHERE ThoiGian.Thu = N'Thứ 2' AND ThoiGian.tietBD = 1

AND ThoiGian.tietKT = 3;

* 1. Hãy hiển thị tổng số tín chỉ đã đăng kí của từng sinh viên gồm maSV, tenSV, lopSH, tongSoTC\_DK.

SELECT SV.maSV, SV.tenSV,LSH.tenLSH, SUM(HP.soTC) AS TongSoTC

FROM LopSH as LSH

JOIN SinhVien SV ON SV.maLSH= LSH.maLSH

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

GROUP BY SV.maSV, SV.tenSV,tenLSH;

* 1. Hãy hiển thị số phòng còn rảnh ở Thứ 3 tiết 1 đến tiết 2

SELECT maPhong, tenPhong

FROM Phong

WHERE maPhong NOT IN (

SELECT DISTINCT ThoiGian\_PhongHoc.maPhong

FROM ThoiGian

JOIN ThoiGian\_PhongHoc

ON ThoiGian.maThoiGian = ThoiGian\_PhongHoc.maThoiGian

WHERE ThoiGian.Thu = N'Thứ 3' AND ThoiGian.tietBD BETWEEN 1 AND 2);

* 1. Hiển thị những học phần không có trong chương trình đào tạo nào

SELECT maHP, tenHP

FROM HocPhan

WHERE maHP NOT IN (

SELECT maHP

FROM ChuongTrinhDaoTao\_HocPhan

);

* 1. Hiển thị những giáo viên dạy từ 2 lớp học phần trở lên

SELECT GV.maGV, GV.tenGV,GV.email, GV.SDT,

COUNT(DISTINCT maLHP) AS SoLuongLopHocPhan

FROM GiaoVien GV

JOIN LopHocPhan LHP ON GV.maGV = LHP.maGV

GROUP BY GV.maGV, GV.tenGV,GV.email, GV.SDT

HAVING COUNT(DISTINCT maLHP) >= 2;

* 1. Hiển thị những lớp học phần nào học vào thứ 2 tiết BD là tiết 1 và tiết kết thúc là tiết 3

SELECT LHP.maLHP, LHP.maHP, LHP.maGV, LHP.soLuongDK, LHP.soLuongDDK, LHP.maTG\_PH, LHP.maHocKi

FROM LopHocPhan LHP

JOIN ThoiGian\_PhongHoc TG\_PH ON LHP.maTG\_PH = TG\_PH.maTG\_PH

JOIN ThoiGian TG ON TG\_PH.maThoiGian = TG.maThoiGian

WHERE TG.Thu = N'Thứ 2' AND TG.tietBD = 1AND TG.tietKT = 3;

* 1. Hiển thị danh sách sinh viên đã đăng kí học phần và lớp học phần tương ứng

SELECT DangKiTinChi.maDK, SinhVien.tenSV, HocPhan.tenHP, LopHocPhan.maLHP, LopHocPhan.maHocKi

FROM DangKiTinChi

JOIN SinhVien ON DangKiTinChi.maSV = SinhVien.maSV

JOIN DangKiChiTiet ON DangKiTinChi.maDK = DangKiChiTiet.maDK

JOIN LopHocPhan ON DangKiChiTiet.maLHP = LopHocPhan.maLHP

JOIN HocPhan ON LopHocPhan.maHP = HocPhan.maHP;

* 1. Hãy in ra thông tin sinh viên gồm mã sinh viên , tên sinh viên, và tổng số môn học mà sinh viên đó đã đăng kí trong học kì 123

SELECT SinhVien.maSV, SinhVien.tenSV, COUNT(DISTINCT LopHocPhan.maLHP) AS TongSoMonHoc

FROM SinhVien

JOIN DangKiTinChi ON SinhVien.maSV = DangKiTinChi.maSV

JOIN DangKiChiTiet ON DangKiTinChi.maDK = DangKiChiTiet.maDK

JOIN LopHocPhan ON DangKiChiTiet.maLHP = LopHocPhan.maLHP

WHERE LopHocPhan.maHocKi = '123'

GROUP BY SinhVien.maSV, SinhVien.tenSV;

* 1. Hãy hiển thị các lớp sinh hoạt và tổng số sinh viên trong lớp sinh hoạt đó

SELECT l.maLSH, l.tenLSH, COUNT(s.maSV) as soLuongSinhVien

FROM dbo.LopSH as l

INNER JOIN dbo.SinhVien as s ON s.maLSH= l.maLSH

GROUP BY l.maLSH, l.tenLSH;

* 1. Hãy hiện thị các lớp sinh hoạt có tổng số sinh viên là 5 trở lên

SELECT l.maLSH, l.tenLSH, COUNT(s.maSV) as soLuongSinhVien

FROM dbo.LopSH as l

INNER JOIN dbo.SinhVien as s ON s.maLSH= l.maLSH

GROUP BY l.maLSH, l.tenLSH

HAVING COUNT(s.maSV) >5;

* 1. Hãy hiển thị thông tin các sinh viên nữ thuộc khoa công nghệ thông tin

SELECT SinhVien.maSV, SinhVien.tenSV, SinhVien.gioiTinh, SinhVien.ngaySinh, SinhVien.diaChi, SinhVien.email, SinhVien.SDT

FROM SinhVien

JOIN LopSH ON SinhVien.maLSH = LopSH.maLSH

JOIN Khoa ON LopSH.maKhoa = Khoa.maKhoa

WHERE SinhVien.gioiTinh = 'F' AND Khoa.tenKhoa = N'Khoa Công Nghệ Thông Tin';

* 1. Hãy hiện thị mã khoa, tên khoa và số lượng sinh viên của mỗi khoa

SELECT Khoa.maKhoa, Khoa.tenKhoa, COUNT(SinhVien.maSV) AS SoLuongSinhVien

FROM Khoa

JOIN LopSH ON Khoa.maKhoa = LopSH.maKhoa

JOIN SinhVien ON LopSH.maLSH = SinhVien.maLSH

GROUP BY Khoa.maKhoa, Khoa.tenKhoa;

# Sử dụng CREATE VIEW để tạo VIEW thích hợp.

* 1. Tạo view chứa các sinh viên năm 1 (năm BD 2023)

CREATE VIEW v\_sinhVienK23

AS

SELECT s.\*, namBD

FROM dbo.SinhVien AS s

JOIN dbo.LopSH AS l ON s.maLSH = l.maLSH

WHERE l.namBD =2023

* 1. Tạo view chứa các sinh viên đã đăng kí học phần kì 123

CREATE VIEW v\_sinhVien\_K\_ĐKHP

AS

SELECT

SV.maSV,SV.tenSV,SV.maLSH,SV.ngaySinh,SV.SDT,SV.email,

SV.gioiTinh,SV.diaChi

FROM SinhVien SV

WHERE

SV.maSV NOT IN (

SELECT DISTINCT SV.maSV

FROM SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

WHERE LHP.maHocKi='123'

);

* 1. Tạo View chứa tất cả các học phần không nằm trong chương trình đào tạo nào

CREATE VIEW v\_HP\_KhongCoTrongChuongTrinhDaoTao

AS

SELECT maHP, tenHP

FROM HocPhan

WHERE maHP NOT IN (

SELECT maHP

FROM ChuongTrinhDaoTao\_HocPhan

);

* 1. Tạo View chứa danh sách lớp học phần nào vào thứ 2 tiết 3-4

CREATE VIEW LopHocPhan\_T2\_Tiet34 AS

SELECT \*

FROM LopHocPhan

WHERE maTG\_PH IN (SELECT maThoiGian FROM ThoiGian

WHERE Thu = N'Thứ 2' AND (tietBD = 3 OR tietBD = 4));

* 1. Tạo View chứa tất cả các sinh viên nữ của khoa công nghệ thông tin có maKhoa =’KHOA001’

CREATE VIEW SinhVien\_Nu\_KhoaCNTT AS

SELECT \*

FROM SinhVien

WHERE gioiTinh = 'F' AND maLSH IN (SELECT maLSH FROM LopSH

WHERE maKhoa = 'KHOA001');

* 1. Tạo View chứa số lượng sinh viên của mỗi khoa

CREATE VIEW SoLuongSinhVien\_Khoa AS

SELECT Khoa.maKhoa, tenKhoa,COUNT(\*) AS SoLuongSV

FROM Khoa

JOIN LopSH ON Khoa.maKhoa = LopSH.maKhoa

JOIN SinhVien ON LopSH.maLSH = SinhVien.maLSH

GROUP BY Khoa.maKhoa, tenKhoa;

* 1. Tạo view chứa danh sách và số lượng sinh viên của mỗi lớp , lớp đó thuộc khoa nào?

CREATE VIEW v\_SoLuongSV\_Lop\_Khoa

AS

SELECT l.maLSH, tenLSH,k.maKhoa,tenKhoa, COUNT(s.maLSH) as SoLuongSV

FROM dbo.SinhVien s

JOIN dbo.LopSH l on s.maLSH = l.maLSH

JOIN dbo.Khoa k on l.maKhoa = k.maKhoa

GROUP BY l.maLSH, tenLSH,k.maKhoa,tenKhoa

* 1. Tạo view chứa danh sách sinh viên đã đăng kí “Học phần 1” của “chương trình đào tạo 1”

CREATE VIEW SinhVien\_HocHP1\_CTD1 AS

SELECT SV.\*,HP.tenHP,CTD.tenChuongTrinhDaoTao

FROM SinhVien SV

JOIN DangKiTinChi DK ON SV.maSV = DK.maSV

JOIN DangKiChiTiet DKCT ON DK.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

JOIN ChuongTrinhDaoTao\_HocPhan CTD\_HP ON HP.maHP = CTD\_HP.maHP

JOIN ChuongTrinhDaoTao CTD ON CTD\_HP.maChuongTrinhDaoTao = CTD.maChuongTrinhDaoTao

WHERE CTD.tenChuongTrinhDaoTao = N'Chương Trình Đào Tạo 1' AND HP.tenHP = N'Học Phần 1';

1. Từ những View đã tạo thực hiện 1 số câu truy vấn từ view trên
   1. Hiển thị thông tin của tất cả sinh viên khóa 23 có giới tính là nam(M)

SELECT \* FROM dbo.v\_sinhVienK23

WHERE gioiTinh ='M'

* 1. Hiển thị thông tin của tất cả sinh viên khóa 23 có giới tính là nữ và có mã lớp sinh hoạt là 'LSH001'

SELECT \* FROM dbo.v\_sinhVienK23

WHERE gioiTinh='F' and maLSH ='LSH001'

* 1. Hiển thị Sinh viên nũ của khoa công nghệ thông tin

SELECT \* FROM dbo.SinhVien\_Nu\_KhoaCNTT

* 1. Hiển thị những sinh viên khóa 23 và có đăng kí học phần 123

SELECT v.\*

FROM v\_sinhVienK23 v

JOIN DangKiTinChi DK ON v.maSV = DK.maSV

WHERE namBD = '123';

* 1. Danh sách sinh viên của các lớp có nhiều hơn 3 sinh viên

SELECT \* FROM [dbo].[SoLuongSinhVien\_Khoa]

WHERE SoLuongSV > 3;

* 1. Hiển thị những sinh viên đã đang kí học phần của lớp SH có maLSH= 'LSH001' và có giới tính nữ(F)

SELECT \* FROM dbo.v\_sinhVien\_K\_ĐKHP

WHERE maLSH ='LSH001' and gioiTinh ='M'

* 1. Danh sách sinh viên nữ thuộc Khoa Công nghệ thông tin và không tham gia đăng ký học phần kỳ 123:

SELECT s.\* FROM [dbo].[SinhVien\_Nu\_KhoaCNTT] as s

JOIN [dbo].[v\_sinhVien\_K\_ĐKHP] as sv ON s.maSV= sv.maSV

* 1. Danh sách các lớp( gồm maLSH và tenLSP) thuộc khóa K23 có sinh viên không tham gia dăng kí học phần 123

SELECT l.maLSH,l.tenLSH

FROM [dbo].[v\_sinhVienK23] v\_sv

JOIN dbo.LopSH l ON v\_sv.maLSH = l.maLSH

and l.maLSH NOT IN (

SELECT DISTINCT maLSH

FROM v\_sinhVien\_K\_ĐKHP

);

1. Sử dụng câu lệnh IF\_ELSE và CASE trong SQL
   1. Kiểm tra xem có sinh viên nào có giới tính nam hay không nếu có thì in ra ‘Co sinh vien nam trong danh sách’ ngược lại in ra ‘Khong co sinh vien nam trong danh sach’.

IF EXISTS (SELECT \* FROM SinhVien WHERE gioiTinh = 'M')

PRINT 'Có sinh viên nam trong danh sách.';

ELSE

PRINT 'Không có sinh viên nam trong danh sách.';

* 1. Câu truy vấn lấy thông tin về lớp sinh hoạt và hiển thị tên mới của các lớp có mã 'LSH003', 'LSH004', 'LSH005'

SELECT maLSH, tenLSH,

CASE

WHEN maLSH = 'LSH003' THEN N'Lớp SH3'

WHEN maLSH = 'LSH004' THEN N'Lớp SH4'

WHEN maLSH = 'LSH005' THEN N'Lớp SH5'

ELSE tenLSH

END AS TenMoi

FROM LopSH

WHERE maLSH IN ('LSH003', 'LSH004', 'LSH005');

* 1. Kiểm tra xem có tồn tại Giáo viên có mã GV=’GV001’ hay không nếu tồn tại thì in ra ‘Giáo viên có ma’ +maGV+’tồn tại’, ngược lại in ra ‘Không có giáo viên nào có mã’+maGV+ ‘ .’

DECLARE @maGV CHAR(15);

SET @maGV = 'GV001';

IF EXISTS (SELECT 1 FROM GiaoVien WHERE maGV = @maGV)

PRINT N'Giáo viên có mã ' + @maGV + N' tồn tại.';

ELSE

PRINT N'Không có giáo viên nào có mã ' + @maGV + '.';

* 1. Kiểm tra sinh viên có mã sinh viên SV001 có đăng kí môn học nào đó hay không nếu sinh viên có đăng kí môn học nào đó thì in ra’Sinh viên có mã’ + maSV+ ‘đã đăng kí môn học.’, ngược lại thì in ra ‘Sinh viên có mã’ +maSV + ‘ chưa đăng kí môn học

DECLARE @maSV CHAR(15);

SET @maSV = 'SV001';

IF EXISTS (SELECT 1 FROM DangKiTinChi WHERE maSV = @maSV)

PRINT 'Sinh viên có mã ' + @maSV + ' đã đăng ký môn học.';

ELSE

PRINT 'Sinh viên có mã ' + @maSV + ' chưa đăng ký môn học.';

* 1. Kiểm tra xem lớp học phần có maLHP ‘LHP001’ có soLuongDDK đạt 50% soLuongDK hay chưa , nêu đạt hãy in ra ‘Lớp học phần’ + maLHP+ ‘đã đạt đến 50% số Lượng đăng kí.’, ngược lại nếu chưa đạt hãy in ra ‘Lớp học phần’ + maLHP+ ‘chưa đạt đến 50% số Lượng đăng kí.’

DECLARE @maLHP CHAR(15) = 'LHP001';

IF EXISTS (

SELECT \*

FROM LopHocPhan

WHERE maLHP = @maLHP

AND soLuongDDK >= 0.5 \* soLuongDK

)

PRINT N'Lớp học phần ' + @maLHP + N' đã đạt đến 50% số lượng đăng ký.';

ELSE

PRINT N'Lớp học phần ' + @maLHP + N' chưa đạt đến 50% số lượng đăng ký.';

* 1. Sử dụng câu lệnh CASE kiểm tra xem tất cả các lớp học phần đã có soLuongDDK đạt 50% của soLuongDK hay chưa, thêm cột trạng thái ở câu SELECT

SELECT

maLHP,

soLuongDDK,

soLuongDK,

CASE

WHEN soLuongDDK >= 0.5 \* soLuongDK THEN N'Đạt 50% trở lên'

ELSE N'Chưa đạt 50%'

END AS TrangThai

FROM LopHocPhan

* 1. Câu truy vấn cập nhật giới tính của sinh viên có mã 'SV001', 'SV002', 'SV003' thành 'M' nếu giới tính là 'F' và ngược lại, sử dụng CASE:

UPDATE SinhVien

SET gioiTinh = CASE

WHEN gioiTinh = 'F' THEN 'M'

ELSE 'F'

END

WHERE maSV IN ('SV001', 'SV002', 'SV003');

* 1. Câu truy vấn cập nhật tên lớp học phần có mã 'LHP001' thành 'Lớp Tin Học Cơ Sở' nếu số lượng đăng ký lớn hơn , 50 ngược lại giữ nguyên, Sử dụng CASE.

UPDATE dbo.LopHocPhan

SET tenLHP = CASE

WHEN soLuongDK > 40 THEN 'Lớp Tin Học Cơ Sở'

ELSE tenLHP

END

WHERE maLHP = 'LHP001';

* 1. Câu truy vấn cập nhật tên giáo viên có mã 'GV003' thành 'Nguyễn Minh Cường' nếu tên giáo viên là 'Hồ Minh C', ngược lại giữ nguyên, sử dụng CASE .

UPDATE GiaoVien

SET tenGV = CASE

WHEN tenGV = N'Hồ Minh C' THEN N'Nguyễn Minh Cường'

ELSE tenGV

END

WHERE maGV = 'GV003';

* 1. Câu truy vấn lấy thông tin về sinh viên và hiển thị giới tính dưới dạng chuỗi 'Nam' hoặc 'Nữ', Sử dụng CASE để thêm một cột mới có tên là GioiTinh

SELECT maSV, tenSV,

CASE

WHEN gioiTinh = 'M' THEN N'Nam'

WHEN gioiTinh = 'F' THEN N'Nữ'

ELSE N'Khác'

END AS GioiTinh

FROM SinhVien;

* 1. Câu truy vấn kiểm tra và hiển thị thông tin về sinh viên có tuổi từ 19 trở lên, sử dụng CASE để thêm một cột mới có tên là tình trạng tuổi

SELECT maSV, tenSV, ngaySinh,

CASE

WHEN DATEDIFF(YEAR, ngaySinh,GETDATE()) >= 19 THEN N'Tuổi đủ 19+'

ELSE N'Chưa đủ 19 tuổi'

END AS TinhTrangTuoi

FROM SinhVien;

* 1. Câu truy vấn lấy thông tin về lớp học phần và hiển thị 'Đã đủ đăng ký'nếu số lượng đăng ký bằng số lượng đăng ký đủ, ngược lại 'Chưa đủ đăng ký', sử dụng CASE và thêm một cột mới tên là TinhTrangDangKy.

SELECT maLHP, soLuongDK, soLuongDDK,

CASE

WHEN soLuongDDK = soLuongDK THEN N'Đã đủ đăng ký'

ELSE N'Chưa đủ đăng ký'

END AS TinhTrangDangKy

FROM LopHocPhan;

* 1. Câu truy vấn lấy thông tin về thời gian và hiển thị 'Buổi sáng' nếu thời gian nằm trong khoảng buổi sáng, ngược lại 'Buổi chiều', sử dụng CASE và thêm một cột mới tên là thoiGian

SELECT maThoiGian, Thu, tietBD, tietKT,

CASE

WHEN tietBD >= 1 AND tietKT <= 5 THEN N'Buổi sáng'

ELSE N'Buổi chiều'

END AS BuoiHoc

FROM ThoiGian;

* 1. Câu truy vấn lấy thông tin về sinh viên và hiển thị 'Email hợp lệ' nếu email có định dạng đúng, ngược lại 'Email không hợp lệ', sử dụng CASE và thêm một cột mới tên là TinhTrangEmail.

SELECT maSV, tenSV, email,

CASE

WHEN email LIKE '%@gmail.com' THEN N'Email hợp lệ'

ELSE N'Email không hợp lệ'

END AS TinhTrangEmail

FROM SinhVien;

1. Sử dụng câu lệnh UNION
   1. Truy vấn dữ liệu sử dụng UNION hãy truy vấn những sinh viên nào có tổng số đăng kí tín chỉ từ 5 đến 10 hoặc đăng kí lớp học phần có mã học phần maHP= ‘HP001’

SELECT SV.\*

FROM SinhVien SV

JOIN DangKiTinChi DKT ON SV.maSV = DKT.maSV

JOIN DangKiChiTiet DKCT ON DKT.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

WHERE SV.maSV IN (

SELECT maSV

FROM DangKiTinChi DKT

JOIN DangKiChiTiet DKCT ON DKT.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

GROUP BY maSV

HAVING SUM(HP.soTC) BETWEEN 2 AND 10

)

UNION

SELECT SV.\*

FROM SinhVien SV

JOIN DangKiTinChi DKT ON SV.maSV = DKT.maSV

JOIN DangKiChiTiet DKCT ON DKT.maDK = DKCT.maDK

JOIN LopHocPhan LHP ON DKCT.maLHP = LHP.maLHP

JOIN HocPhan HP ON LHP.maHP = HP.maHP

WHERE HP.maHP = 'HP001'

* 1. Hãy truy vấn dữ liệu sử dụng UNION , những sinh viên nào có email kết thúc bằng ‘@gmail.com’ hoặc có số điện thoại bắt đầu bằng ‘09’

SELECT \*

FROM SinhVien

WHERE email LIKE '%@gmail.com%'

UNION

SELECT \*

FROM SinhVien

WHERE SDT LIKE '09%';

* 1. Hãy truy vấn dữ liệu sử dụng UNION , những sinh viên có giới tính là ‘Nam’ hoặc có tháng sinh từ tháng 1 đến tháng 6.

SELECT \*

FROM SinhVien

WHERE gioiTinh = 'Nam'

UNION

SELECT \*

FROM SinhVien

WHERE MONTH(ngaySinh) BETWEEN 1 AND 6;

* 1. Liệt kê tên, loại (sinh viên/giáo viên), và ngày đăng ký của tất cả các đăng ký môn học có sử dụng UNION

SELECT maSV AS Ma, 'SinhVien' AS Loai, thoiGianDK AS NgayDangKy

FROM DangKiTinChi

UNION

SELECT maGV AS Ma, 'GiaoVien' AS Loai, NULL AS NgayDangKy

FROM GiaoVien;

* 1. Sử dụng câu lệnh UNION để thực hiện Liệt kê tất cả các lớp học phần và thời gian học tương ứng

SELECT maLHP AS Ma, 'LopHocPhan' AS Loai, NULL AS ThoiGianHoc

FROM LopHocPhan

UNION

SELECT tg.maThoiGian AS Ma, 'ThoiGianHoc' AS Loai,

tg.Thu + ' ' + CAST(tietBD AS NVARCHAR) + '-' + CAST(tietKT AS NVARCHAR) AS ThoiGianHoc

FROM ThoiGian\_PhongHoc AS tg\_ph

JOIN ThoiGian AS tg ON tg.maThoiGian = tg\_ph.maThoiGian;

* 1. Truy vấn có sử dụng UNION liệt kê tất cả các môn học và số lượng sinh viên đã đăng ký

SELECT tenHP AS MonHoc, COUNT(DISTINCT maDKCT) AS SoLuongSinhVien

FROM HocPhan

LEFT JOIN LopHocPhan ON HocPhan.maHP = LopHocPhan.maHP

LEFT JOIN DangKiChiTiet ON LopHocPhan.maLHP = DangKiChiTiet.maLHP

GROUP BY tenHP

ORDER BY SoLuongSinhVien DESC;

1. Tạo những FUNCTION có thể có trong Database này :
   1. Tạo FUNCTION lấy tên khoa theo mã khoa

CREATE FUNCTION dbo.GetTenKhoa (

@maKhoa CHAR(10)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @tenKhoa NVARCHAR(50)

SELECT @tenKhoa = tenKhoa FROM Khoa WHERE maKhoa = @maKhoa

RETURN @tenKhoa

END;

* 1. Tạo FUNCTION lấy tên lớp Sinh Hoạt của một sinh viên với tham số đầu vào là maSV

CREATE FUNCTION dbo.GetTenLopSH (

@maSV CHAR(15)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @tenLopSH NVARCHAR(50)

SELECT @tenLopSH = l.tenLSH

FROM SinhVien AS s

INNER JOIN LopSH AS l ON s.maLSH = l.maLSH

WHERE s.maSV = @maSV

RETURN @tenLopSH

END;

* 1. Tạo FUNCTION lấy tên giáo viên dựa vào mã giáo viên được truyền vào.

CREATE FUNCTION dbo.GetTenGiaoVien (

@maGV CHAR(15)

)

RETURNS NVARCHAR(255)

AS

BEGIN

DECLARE @tenGV NVARCHAR(255)

SELECT @tenGV = tenGV FROM GiaoVien WHERE maGV = @maGV

RETURN @tenGV

END;

* 1. Tạo FUNCTION lấy số lượng học phần mà sinh viên đã đăng kí với mã sinh viên là tham số được truyền vào .

CREATE FUNCTION dbo.GetSoLuongDangKy (

@maSV CHAR(15)

)

RETURNS INT

AS

BEGIN

DECLARE @soLuong INT

SELECT @soLuong = COUNT(\*) FROM DangKiTinChi WHERE maSV = @maSV

RETURN @soLuong

END;

* 1. Tạo FUNCTION lấy số tên học phần theo mã học phần được truyền vào làm tham số

CREATE FUNCTION dbo.GetTenHocPhan (

@maHP CHAR(20)

)

RETURNS NVARCHAR(255)

AS

BEGIN

DECLARE @tenHP NVARCHAR(255)

SELECT @tenHP = tenHP FROM HocPhan WHERE maHP = @maHP

RETURN @tenHP

END;

* 1. Tạo FUNCTION lấy thôn tên thời gian gồm (Thứ -tietBD-tietKT)

CREATE FUNCTION dbo.GetTenThoiGianHoc (

@maThoiGian CHAR(10)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @tenThoiGian NVARCHAR(50)

SELECT @tenThoiGian = Thu + ' ' + CAST(tietBD AS NVARCHAR) + '-' + CAST(tietKT AS NVARCHAR)

FROM ThoiGian

WHERE maThoiGian = @maThoiGian

RETURN @tenThoiGian

END;

* 1. Tạo FUNCTION lấy tên phòng học theo mã phòng học được truyền vào.

CREATE FUNCTION dbo.GetTenPhongHoc (

@maPhong CHAR(10)

)

RETURNS NVARCHAR(20)

AS

BEGIN

DECLARE @tenPhong NVARCHAR(20)

SELECT @tenPhong = tenPhong FROM Phong WHERE maPhong = @maPhong

RETURN @tenPhong

END;

* 1. Tạo FUCNTION đếm số lượng đăng kí chi tiết trong một lớp học phần

CREATE FUNCTION dbo.GetSoLuongDangKyChiTiet (

@maLHP CHAR(15)

)

RETURNS INT

AS

BEGIN

DECLARE @soLuong INT

SELECT @soLuong = COUNT(\*) FROM DangKiChiTiet WHERE maLHP = @maLHP

RETURN @soLuong

END;

* 1. Tạo FUNCTION check điều kiện đủ tuổi của một sinh viên với mã sinh viên là tham số được truyền vào

CREATE FUNCTION dbo.CheckAgeCondition (

@ngaySinh DATETIME

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @tuoi INT

SET @tuoi = DATEDIFF(YEAR, @ngaySinh, GETDATE())

DECLARE @ketQua NVARCHAR(50)

IF @tuoi >= 18

SET @ketQua = 'Đủ 18 tuổi'

ELSE

SET @ketQua = 'Chưa đủ 18 tuổi'

RETURN @ketQua

END;

* 1. Tạo FUNCTION Kiểm Tra Số Lượng Đăng Ký Đã Đạt Mức Tối Đa Theo Lớp Học Phần

CREATE FUNCTION dbo.CheckSoLuongSVDaDatToiDaTheoLHP (

@maLHP CHAR(15)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @soLuongDK INT

DECLARE @soLuongDDK INT

SELECT @soLuongDK = soLuongDK, @soLuongDDK = soLuongDDK

FROM LopHocPhan WHERE maLHP = @maLHP

DECLARE @ketQua NVARCHAR(50)

IF @soLuongDDK >= @soLuongDK

SET @ketQua = N'Đã đạt số lượng đăng ký tối đa'

ELSE

SET @ketQua = N'Chưa đạt số lượng đăng ký tối đa'

RETURN @ketQua

END;

* 1. Tạo FUNCTION kiểm tra lớp học phần đã có giáo viên dạy hay chưa

CREATE FUNCTION dbo.CheckLHPDaCoGiaoVien (

@maLHP CHAR(15)

)

RETURNS NVARCHAR(50)

AS

BEGIN

DECLARE @ketQua NVARCHAR(50)

IF EXISTS (SELECT 1 FROM LopHocPhan WHERE maLHP = @maLHP AND maGV IS NOT NULL)

SET @ketQua = N'Lớp học phần đã có giáo viên dạy'

ELSE

SET @ketQua = N'Lớp học phần chưa có giáo viên dạy'

RETURN @ketQua

END;

1. Tạo PROCEDURE có thể có trong DATABASE

14.1 Có sử dụng FUNCTION đã tạo ở bên trên trong PROCEDURE

* 1. Tạo PROCEDURE lấy tên khoa theo mã khoa được truyền vào

CREATE PROCEDURE dbo.Pro\_GetTenKhoa

@maKhoa CHAR(10)

AS

BEGIN

SELECT k.maKhoa, dbo.GetTenKhoa(k.maKhoa) as tenKhoa

FROM dbo.Khoa AS k

WHERE k.maKhoa = @maKhoa

END;

* 1. Tạo PROCEDURE kiểm tra số lượng đăng kí đã đạt mức tối đa theo lớp học phần

CREATE PROCEDURE dbo.Pro\_CheckSoLuongSVDaDatToiDaTheoLHP

@maLHP CHAR(15)

AS

BEGIN

SELECT lhp.\*, dbo.CheckSoLuongSVDaDatToiDaTheoLHP(lhp.maLHP) as KetQuaDK

FROM dbo.LopHocPhan AS lhp

WHERE lhp.maLHP = @maLHP

END;

* 1. Lấy lớp sinh hoạt của sinh viên dựa vào mã sinh viên truyền vào

CREATE PROCEDURE dbo.GetTenLopSHProcedure

@maSV CHAR(15)

AS

BEGIN

SELECT s.\*, dbo.GetTenLopSH(s.maSV) as tenLSH

FROM dbo.SinhVien AS s

WHERE s.maSV = @maSV

END;

* 1. Tạo PROCEDURE lấy tên giáo viên thông qua maGiaoVien

CREATE PROCEDURE dbo.GetTenGiaoVienProcedure

@maGV CHAR(15)

AS

BEGIN

SELECT gv.\*, dbo.GetTenGiaoVien(gv.maGV) as tenGV

FROM dbo.GiaoVien AS gv

WHERE gv.maGV = @maGV

END;

* 1. Tạo PROCEDURE lấy số lượng học phần mà sinh viên đã đăng kí thông qua mã sinh viên

CREATE PROCEDURE dbo.GetSoLuongDangKyProcedure

@maSV CHAR(15)

AS

BEGIN

SELECT s.\*, dbo.GetSoLuongDangKy(s.maSV) as SoLuongDK

FROM dbo.SinhVien s

WHERE s.maSV = @maSV

END;

* 1. Tạo PROCEDURE không sử dụng FUNCTION đã tạo ở trên.
  2. Tạo PROCEDURE lấy tất cả sinh viên trong một lớp cụ thể

CREATE PROCEDURE pro\_LaySinhVienTrongLop

@maLop CHAR(10)

AS

BEGIN

SELECT \*

FROM SinhVien

WHERE maLSH = @maLop;

END;

### Tạo PROCEDURE lấy lịch học của một sinh viên cụ thể thông qua mã sinh viên

CREATE PROCEDURE pro\_LayLichHocSinhVien

@maSinhVien CHAR(15)

AS

BEGIN

SELECT LopHocPhan.\*, ThoiGian.Thu, ThoiGian.tietBD, ThoiGian.tietKT, Phong.tenPhong

FROM LopHocPhan

INNER JOIN ThoiGian\_PhongHoc

ON LopHocPhan.maTG\_PH = ThoiGian\_PhongHoc.maTG\_PH

INNER JOIN ThoiGian ON ThoiGian\_PhongHoc.maThoiGian = ThoiGian.maThoiGian

INNER JOIN Phong ON ThoiGian\_PhongHoc.maPhong = Phong.maPhong

WHERE EXISTS (

SELECT 1

FROM DangKiChiTiet

WHERE DangKiChiTiet.maLHP = LopHocPhan.maLHP

AND DangKiChiTiet.maDK = @maSinhVien

);

END;

### Tạo PROCEDURE cập nhật thông tin sinh viên

CREATE PROCEDURE pro\_CapNhatThongTinSinhVien

@maSinhVien CHAR(15),

@diaChiMoi NVARCHAR(255),

@emailMoi VARCHAR(100),

@soDienThoaiMoi CHAR(10)

AS

BEGIN

UPDATE SinhVien

SET diaChi = @diaChiMoi,

email = @emailMoi,

SDT = @soDienThoaiMoi

WHERE maSV = @maSinhVien;

END

### Tạo PROCEDURE lấy thông tin chi tiết của một lớp học phần cụ thể

CREATE PROCEDURE LayChiTietLopHoc

@maLopHoc CHAR(15)

AS

BEGIN

SELECT \*

FROM LopHocPhan

WHERE maLHP = @maLopHoc;

END;

### Tạo PROCEDURE Lấy chi tiết chương trình đào tạo cho một mã chương trình cụ thể

CREATE PROCEDURE LayChiTietChuongTrinhDaoTao

@maChuongTrinh CHAR(20)

AS

BEGIN

SELECT \*

FROM ChuongTrinhDaoTao

WHERE maChuongTrinhDaoTao = @maChuongTrinh;

END;

### Tạo PROCEDURE lấy danh sách môn học sinh viên đã đăng kí theo maSV và maHocKi

CREATE PROCEDURE LayDanhSachHocPhanSinhVienDK

@maSV CHAR(15),

@maHocKi CHAR(5)

AS

BEGIN

SELECT

dkc.maDKCT,

lh.maLHP,

hp.tenHP,

lh.maHocKi,

tg.Thu,

tg.tietBD,

tg.tietKT

FROM

DangKiChiTiet dkc

INNER JOIN LopHocPhan lh ON dkc.maLHP = lh.maLHP

INNER JOIN HocPhan hp ON lh.maHP = hp.maHP

INNER JOIN ThoiGian tg ON lh.maTG\_PH = tg.maThoiGian

INNER JOIN DangKiTinChi dktc ON dktc.maDK = dkc.maDK

WHERE

dktc.maSV = @maSV

AND lh.maHocKi = @maHocKi;

END;

### Tạo PROCEDURE INSERT 1 dòng dữ liệu vao bảng LopHocPhan đảm bảo không trùng khóa chính, thực hiện kiểm tra tính hợp lệ của dữ liệu , đảm bảo toàn vẹn tham chiếu .

CREATE PROCEDURE sp\_ThemLopHocPhan

@maLHP CHAR(15),

@maHP CHAR(20),

@maGV CHAR(15),

@soLuongDK INT,

@soLuongDDK INT,

@maTG\_PH CHAR(15),

@maHocKi CHAR(5),

@ghiChu NVARCHAR(255)

AS

BEGIN

-- Kiểm tra dữ liệu đầu vào phải đảm bảo không null ở các cột đã xét not null

IF @maLHP IS NULL OR @maHP IS NULL OR @maGV IS NULL OR @maTG\_PH IS NULL OR @maHocKi IS NULL

BEGIN

print N'Các tham số không được để trống.';

return;

END

-- Kiểm tra khóa chính không trùng

IF NOT EXISTS (SELECT 1 FROM LopHocPhan WHERE maLHP = @maLHP)

BEGIN

-- Kiểm tra khóa ngoại Mã Học Phần

IF NOT EXISTS (SELECT 1 FROM HocPhan WHERE maHP = @maHP)

BEGIN

print N'Mã Học Phần không tồn tại.';

return ;

END

-- Kiểm tra khóa ngoại Mã Giáo Viên

IF NOT EXISTS (SELECT 1 FROM GiaoVien WHERE maGV = @maGV)

BEGIN

print N'Mã Giáo Viên không tồn tại.';

return ;

END

-- Kiểm tra khóa ngoại Mã Thời Gian - Phòng Học

IF NOT EXISTS (SELECT 1 FROM ThoiGian\_PhongHoc WHERE maTG\_PH = @maTG\_PH)

BEGIN

print N'Mã Thời Gian - Phòng Học không tồn tại.';

return;

END

-- Thêm dòng vào bảng LopHocPhan

INSERT INTO LopHocPhan (maLHP, maHP, maGV, soLuongDK, soLuongDDK, maTG\_PH, maHocKi, ghiChu)

VALUES (@maLHP, @maHP, @maGV, @soLuongDK, @soLuongDDK, @maTG\_PH, @maHocKi, @ghiChu);

PRINT N'Thêm lớp học phần thành công.';

END

ELSE

BEGIN

print N'Mã Lớp Học Phần đã tồn tại.';

return ;

END

END;

1. Tạo TRIGGER cho CƠ SỞ DỮ LIỆU

### Tạo Trigger cho bảng LopHocPhan thực hiện yêu cầu INSERT, DELETE, UPDATE như sau :

* Khi có một sinh viên đăng kí một lớp học phần thì số lượng DDK thì phải tăng lên 1. (nếu soLuongDDK +1 của lớp học phần không lớn hơn số lượng DK tối đa, ngược lại thì thông báo cho sinh viên “N'Số lượng học sinh đăng ký không thể vượt quá số lượng đăng ký cho phép!”
* Trường hợp khi một sinh viên xóa lớp học phần thì thực hiện tăng số lượng ĐK lên 1 và số lượng DDK giảm xuống 1

CREATE TRIGGER trig\_LHP

ON [dbo].[DangKiChiTiet]

AFTER INSERT, DELETE, UPDATE

AS

BEGIN

if not exists (select \* from deleted)

-- INSERT data

BEGIN

IF EXISTS (SELECT 1 FROM inserted i

JOIN LopHocPhan l ON i.maLHP = l.maLHP

WHERE l.soLuongDDK + 1 > l.soLuongDK)

BEGIN

print N'Số lượng đăng ký vượt quá số lượng đăng ký cho phép!'

ROLLBACK;

END

UPDATE LopHocPhan

SET soLuongDDK = l.soLuongDDK + 1

FROM inserted i, LopHocPhan l

WHERE i.maLHP = l.maLHP

END;

if not exists (select \* from inserted)

BEGIN

-- DELETE data

UPDATE LopHocPhan

SET soLuongDDK = l.soLuongDDK - 1

FROM deleted d, LopHocPhan l

WHERE d.maLHP = l.maLHP

END;

END;

### Tạo trigger cho bảng LopHocPhan, trường hợp một lớp học phần không đủ số lượng sinh viên đăng kí bắt buộc phòng đào tạo phải thực hiện xóa lớp đó đi thì mình phải cập nhật những lớp học phần mà sinh viên đã đăng kí.

CREATE TRIGGER Trig\_LopHocPhan

ON LopHocPhan

AFTER DELETE

AS

BEGIN

DELETE DangKiChiTiet

FROM deleted d

WHERE DangKiChiTiet.maLHP = d.maLHP

END;

### Tạo Trigger kiểm tra khi sinh viên thực hiên đăng kí thêm một học phần vào bảng DangKiChiTiet, trigger có nhiệm vụ kiểm ra xem Lớp học phần vừa được thêm vào có trùng thời gian với lớp học phần đã đăng kí hay không nếu có thì thông báo cho người dung và thực hiện rollback , ngược lại nếu thỏa mãn không trùng thời gian thì cho phép insert dữ liệu vào bảng:

CREATE TRIGGER trg\_AfterInsert\_DangKiChiTiet

ON dbo.DangKiChiTiet

AFTER INSERT

AS

BEGIN

-- Kiểm tra xem có hai lớp học phần trùng thời gian học cho một sinh viên hay không

IF NOT EXISTS (

SELECT 1

FROM inserted i

JOIN LopHocPhan l ON i.maLHP = l.maLHP

JOIN dbo.ThoiGian\_PhongHoc tgph ON tgph.maTG\_PH = l.maTG\_PH

JOIN dbo.ThoiGian tg ON tg.maThoiGian = tgph.maThoiGian

WHERE EXISTS (

SELECT 1

FROM DangKiChiTiet dkt

JOIN LopHocPhan lh ON dkt.maLHP = lh.maLHP

JOIN dbo.ThoiGian\_PhongHoc tgphdkt ON tgphdkt.maTG\_PH = lh.maTG\_PH

JOIN dbo.ThoiGian tgdkt ON tgdkt.maThoiGian = tgphdkt.maThoiGian

JOIN DangKiTinChi dktc ON dkt.maDK = dktc.maDK

WHERE dktc.maSV = ( SELECT maSV From dbo.DangKiTinChi DKTC,

DangKiChiTiet DKCT WHERE DKTC.maDK = DKCT.maDK)

AND dkt.maDKCT <> i.maDKCT

AND EXISTS(SELECT COUNT(TG.maThoiGian) FROM dbo.DangKiChiTiet DKCT

JOIN dbo.LopHocPhan LHP ON DKCT.maLHP = LHP.maHP

JOIN dbo.ThoiGian\_PhongHoc TG\_PH ON LHP.maTG\_PH = TG\_PH.maThoiGian

JOIN dbo.ThoiGian TG ON TG.maThoiGian = TG\_PH.maThoiGian

GROUP BY DKCT.maDK

HAVING COUNT(TG.maThoiGian) >1)

)

)

BEGIN

-- Tất cả các môn học phần không có thời gian trùng nhau, tiến hành insert thêm một Lớp học phần mới

PRINT N'Lớp học phần vừa đăng kí thỏa mãn yêu cầu về thời gian cho phép thêm mới ';

END

ELSE

BEGIN

-- Có ít nhất một môn học phần trùng thời gian thực hiện rollback

PRINT N'Có ít nhất một môn học phần trùng thời gian. Rollback!';

ROLLBACK;

END;

END;