

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU ORACLE

ĐỀ TÀI 10: Xây dựng cơ sở dữ liệu hệ thống quản lý sinh viên

Giảng viên hướng dẫn: Th.s Nguyễn Việt Hưng

Nhóm : 10

Lớp : CNTT1_K62

Danh sách thành viên:

STT	Họ và Tên	Mã Sinh Viên
1	Hoàng Trung Nguyên (Trưởng nhóm)	211243705
2	Phùng Tiến Việt	211213865

Hà nội, năm 2023

MỤC LỤC

LỜI NÓI ĐẦU	3
1. Phân tích bài toán	4
1.1. Giới thiệu đề tài.....	4
1.2. Mô tả các thực thể.....	5
1.2.1. Sơ đồ ERD – Mô hình thực thể quan hệ.....	5
1.2.2. Mô tả các thực thể.....	6
1.3. Mối quan hệ các thực thể	11
2. Xây dựng cơ sở dữ liệu với Oracle.....	13
2.1. Xây dựng các bảng dữ liệu	13
2.2. Xây dựng Diagram.....	17
2.3. Tạo các ràng buộc:	18
2.4. Nhập dữ liệu.....	20
3. Truy vấn.....	20
3.1. Truy vấn cơ bản	20
3.2. Truy vấn lồng nhau	21
3.3. Truy vấn nhóm.....	22
3.4. Các hàm thống kê.....	22
4. Lập trình chương trình PL/SQL	23
5. Quản trị CSDL Oracle	25
5.1. Quản lý Instance.....	25
5.2. Quản lý Table Space	26
5.3. Truy vấn thông tin về cấu trúc lưu trữ	28
5.4. Quản trị người dùng, quyền, chức danh.....	28
5.5. Import và Export 1 Schema	29

LỜI NÓI ĐẦU

Ngày nay, song song với quá trình phát triển công nghệ khoa học và kỹ thuật thì ngành khoa học tính toán đã đóng vai trò rất quan trọng. Nó đã đạt được những thành tựu khoa học kỹ thuật rực rỡ với những bước tiến nhảy vọt. Việc áp dụng các thành tựu của khoa học kỹ thuật vào đời sống của con người ngày càng tăng và đang không ngừng can thiệp vào hầu hết các lĩnh vực trong đời sống. Công nghệ thông tin là một trong những ngành khoa học đó. Đi đôi với sự phát triển của công nghệ chế tạo các thiết bị máy tính thì việc các sản phẩm phần mềm ứng dụng ra đời có tính chất quyết định đối với việc áp dụng ngành khoa học này .

Trên thế giới cũng như ở Việt Nam , công nghệ thông tin đã trở thành một ngành công nghiệp mũi nhọn nó là một ngành khoa học kỹ thuật không thể thiếu trong việc áp dụng vào các hoạt động xã hội như : Quản lý, kinh tế, thông tin ...

Ở nước ta hiện nay, việc áp dụng công nghệ thông tin trong việc quản lý sinh viên ở các trường Đại học có thể nói là đang cực kỳ toàn diện và càng ngày càng hiện đại. Tuy nhiên không phải ứng dụng hay hệ thống quản lý nào cũng đáp ứng được đầy đủ các yêu cầu của sinh viên và người quản lý thông tin ấy.

Với mong muốn giúp các bạn hiểu được tầm quan trọng của việc phân tích thiết kế một cơ sở dữ liệu tự động hoá trong lĩnh vực *quản lý sinh viên*. Chúng em đã cố gắng và mạnh dạn đưa ra một mô hình cơ sở dữ liệu *quản lý sinh viên* được xây dựng hệ quản trị cơ sở dữ liệu Oracle.

Qua đây chúng em xin được gửi lời cảm ơn đến Th.s Nguyễn Việt Hưng đã cung cấp cho chúng em những kiến thức cốt lõi trong môn Cơ sở dữ liệu Oracle để chúng em có thể hoàn thành bài báo cáo này. Bài báo cáo còn nhiều thiếu sót, em rất mong nhận được những góp ý cũng như đánh giá từ thầy và các bạn để bài báo cáo được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

Nhóm 10

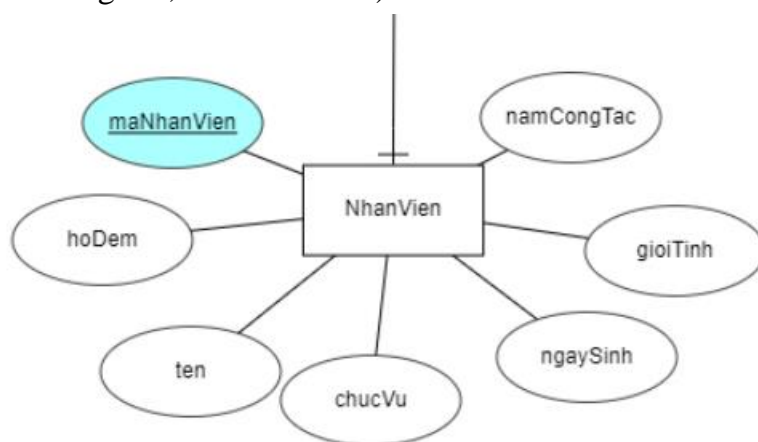
1. Phân tích bài toán

1.1. Giới thiệu đề tài

- Tên đề tài: Xây dựng cơ sở dữ liệu hệ thống quản lý sinh viên
- Mô tả đề tài: Hệ thống quản lý sinh viên là hệ thống thông tin dùng để quản lý thông tin của sinh viên, bao gồm thông tin cá nhân, thông tin học tập, thông tin hoạt động ngoại khóa,... Hệ thống này giúp cho nhà trường có thể dễ dàng quản lý thông tin của sinh viên, đồng thời giúp sinh viên dễ dàng tiếp cận thông tin của mình.
- Quy mô quản lý: Trường Đại học (Do dữ liệu của sinh viên trường Đại học là rất lớn nên nhóm chỉ tập trung quy hoạch vào một nhóm nhỏ mang tính chất demo cho dữ liệu)
- Phạm vi quản lý: Hệ thống quản lý sinh viên sẽ quản lý thông tin của tất cả các sinh viên đang theo học tại trường, bao gồm:
 - Thông tin cá nhân: Họ tên, mã sinh viên, ngày sinh, giới tính, địa chỉ, số điện thoại, email, điểm GPA
 - Thông tin học tập: Chương trình học, Môn học, Lớp học phần, Điểm thi, Điểm quá trình
 - Thông tin tổ chức chính trị: khối Đoàn, khối Đảng
 - Điểm trung bình: Từng kỳ học

1.2.2. Mô tả các thực thể

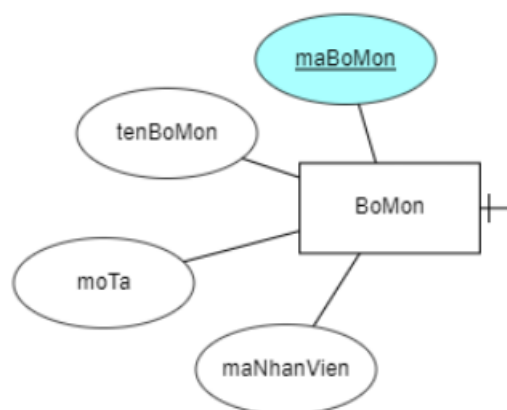
1. NhanVien (maNhanVien, hoDem, ten, chucVu, ngaySinh, gioiTinh, namCongTac, NamXuatBan)



Thực thể nhân viên: Mô tả nhân viên (giáo viên và cán bộ công chức) trong trường Đại học bao gồm:

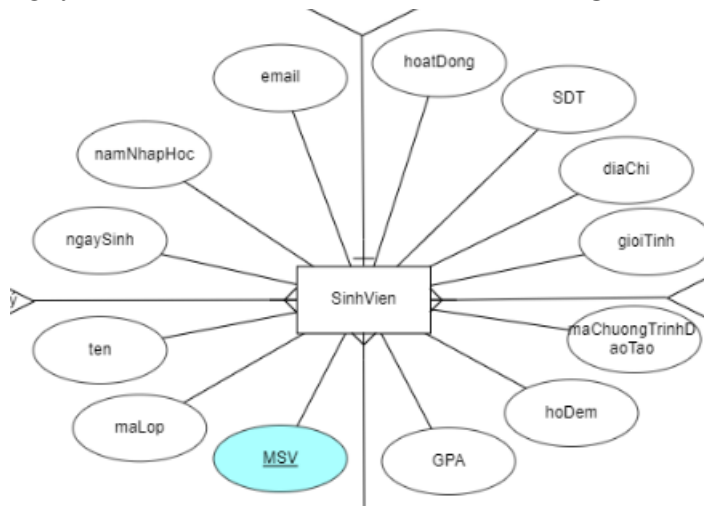
- + maNhanVien: chứa thông tin mã nhân viên
- + hoDem: thông tin họ và tên đệm
- + ten: thông tin tên
- + chucVu: thông tin về chức vụ
- + ngaySinh: thông tin về ngày sinh
- + gioiTinh: thông tin về giới tính
- + namCongTac: thông tin về năm công tác
- + namXuatBan: thông tin năm xuất bản

2. BoMon (maBoMon, tenBoMon, moTa): Mô tả thông tin bộ môn



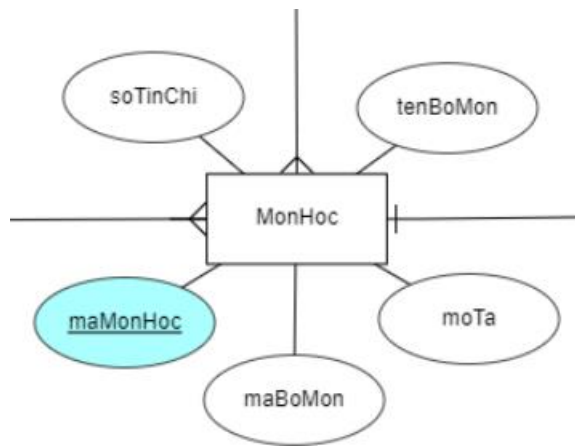
- + maBoMon: thông tin mã bộ môn – Khóa chính
- + tenBoMon: thông tin tên bộ môn
- + moTa: thông tin mô tả bộ môn
- + maNhanVien: thông tin mã nhân viên giảng dạy bộ môn – Khóa phụ bảng NhanVien

3. SinhVien (MSV, maChuongTrinhDaoTao, maLop, gioiTinh, namNhapHoc, ngaySinh, ten, hoDem, GPA): Mô tả thông tin của sinh viên



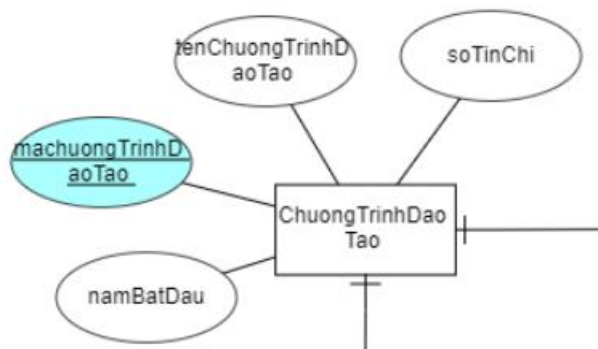
- + MSV: thông tin mã sinh viên – Khóa chính
- + maChuongTrinhDaoTao: mã chương trình đào tạo – Khóa phụ bảng chương trình đào tạo
- + maLop: mã lớp – Khóa phụ bảng Lop
- + gioiTinh: thông tin giới tính
- + namNhapHoc: thông tin năm nhập học
- + ngaySinh: thông tin ngày sinh
- + ten: thông tin tên
- + hoDem: thông tin họ và tên đệm
- + GPA: thông tin điểm GPA
- + email: Email sinh viên được tạo tự động
- + hoatDong: tình trạng sinh viên (0 là chưa tham gì gì, 1 là đoàn viên và 2 là đảng viên)
- + SDT: Thông tin về số điện thoại sinh viên
- + diaChi: Thông tin về địa chỉ sinh viên

4. MonHoc (maMonHoc, tenBoMon, moTa, maBoMon, soTinChi, maChuongTrinhDaoTao)



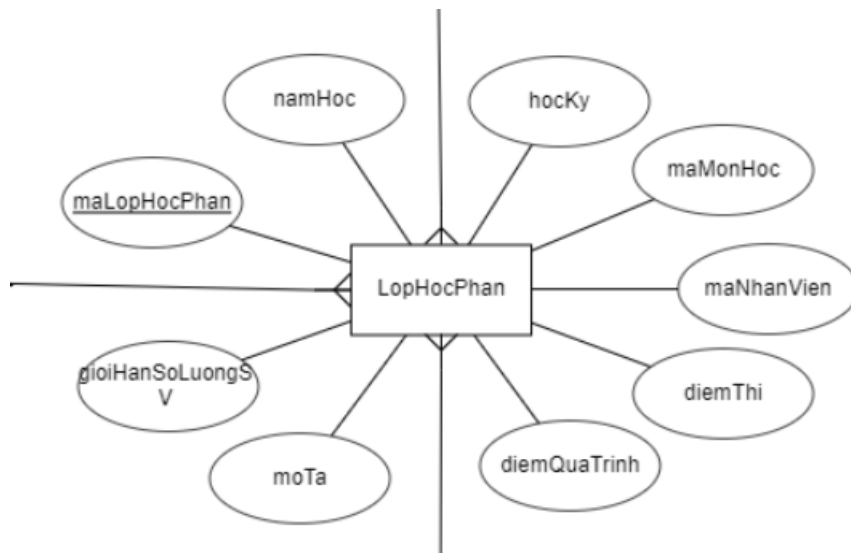
- + maMonHoc: thông tin mã môn học – Khóa chính
- + tenMonHoc: thông tin tên môn học
- + moTa: thông tin mô tả môn học
- + maBoMon: thông tin mã bộ môn – Khóa phụ từ bảng BoMon
- + soTinChi: thông tin số tín chỉ
- + maChuongTrinhDaoTao: thông tin mã chương trình đào tạo – Khóa phụ bảng ChuongTrinhDaoTao

5. ChuongTrinhDaoTao (machuongTrinhDaoTao, tenChuongTrinhDaoTao, namBatDau, soTinChi)



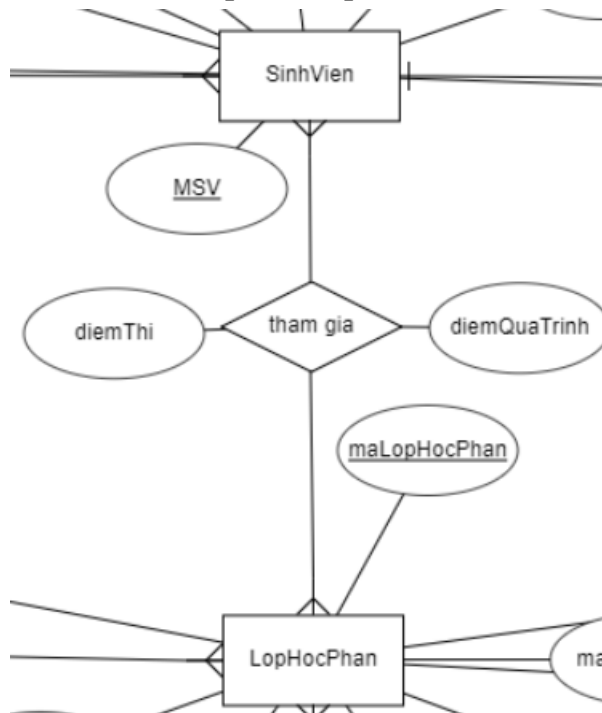
- + maChuongTrinhDaoTao: thông tin mã chương trình đào tạo – Khóa chính
- + tenChuongTrinhDaoTao: thông tin tên chương trình đào tạo
- + namBatDau: thông tin năm bắt đầu của chương trình đào tạo
- + soTinChi: thông tin số tín chỉ của chương trình đào tạo

6. LopHocPhan (maLopHocPhan, tenLopHocPhan, hocKy, soLuongSinhVien, namHoc, moTa, maMonHoc, maNhanVien)



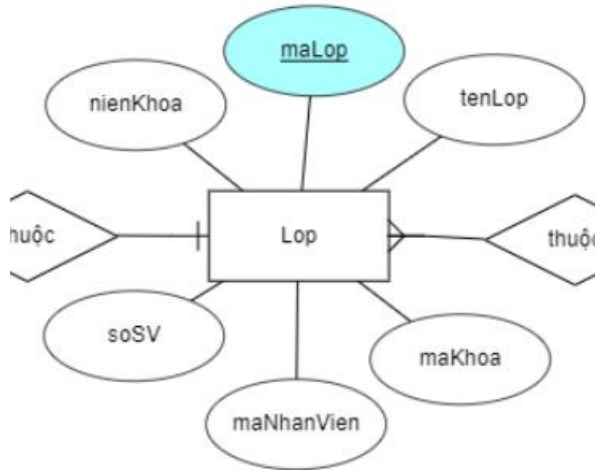
- + maLopHocPhan: thông tin mã lớp học phần – Khóa chính
- + tenLopHocPhan: thông tin tên lớp học phần
- + hocKy: thông tin học kì áp dụng lớp học phần
- + soLuongSinhVien: số lượng sinh viên đăng ký của lớp học phần
- + namHoc: thông tin năm học áp dụng lớp học phần
- + moTa: thông tin mô tả về lớp học phần
- + maMonHoc: thông tin mã môn học – Khóa phụ từ bảng MonHoc
- + maNhanVien: thông tin mã nhân viên phụ trách lớp học phần - Khóa phụ từ bảng NhanVien

7. SinhVienTGLop (maLopHocPhan, MSV, diemQuaTrinh, diemThi)

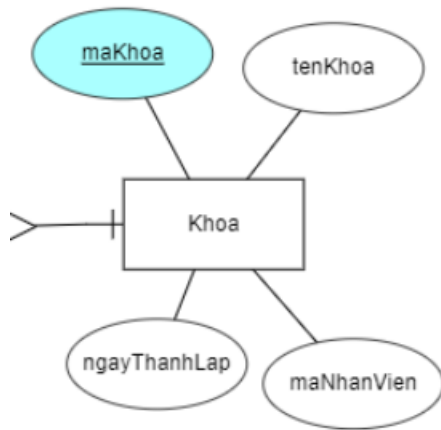


- + maLopHocPhan – Khóa phụ từ bảng LopHocPhan

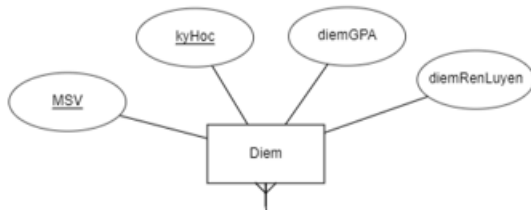
- + MSV – Khóa phụ từ bảng SinhVien
 - + diemQuaTrinh: thông tin điểm quá trình của lớp học phần
 - + diemThi: thông tin điểm thi của lớp học phần
8. Lop (maLop, tenLop, maKhoa, nienKhoa, soSinhVien, maNhanVien)



- + maLop: thông tin mã lớp – Khóa chính
 - + tenLop: thông tin tên lớp
 - + maKhoa: thông tin mã khoa – Khóa phụ bảng Khoa
 - + nienKhoa: thông tin niên khóa
 - + soSinhVien: thông tin số lượng sinh viên trong lớp
 - + maNhanVien: thông tin mã nhân viên là cố vấn học tập của lớp
9. Khoa (maKhoa, tenKhoa, maNhanVien, ngayThanhLap)



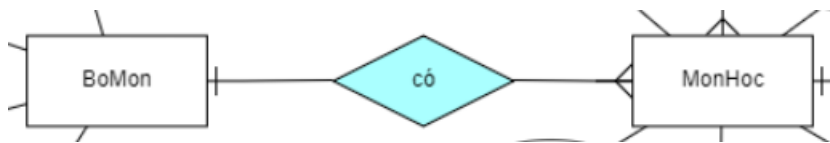
- + maKhoa: thông tin mã khoa – Khóa chính
 - + tenKhoa: thông tin về tên khoa
 - + maNhanVien: thông tin mã nhân viên đóng vai trò là trưởng khoa
 - + ngayThanhLap: thông tin về ngày thành lập khoa
10. Diem (MSV, kyHoc, diemGPA, diemRenLuyen)



- + (MSV, kyHoc): Thông tin mã sinh viên và kỳ học – Khóa chính
- + diemGPA: Thông tin GPA điểm từng kỳ
- + diemRenLuyen: Thông tin điểm rèn luyện từng kỳ

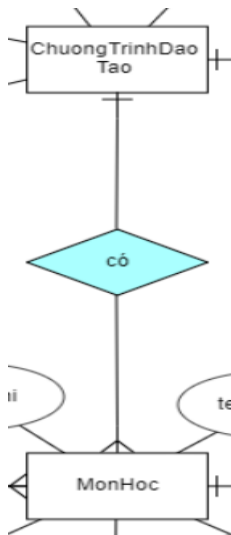
1.3. Mối quan hệ các thực thể

1. Quan hệ một-nhiều của thực thể BoMon và thực thể MonHoc



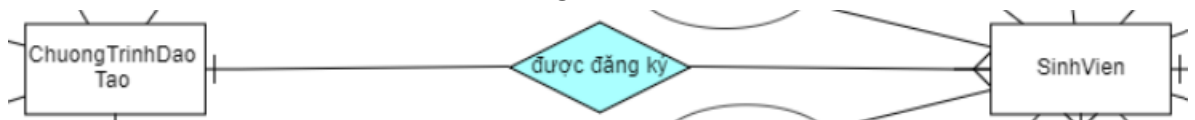
➔ Một bộ môn có nhiều môn học thuộc về nó

2. Quan hệ một-nhiều của thực thể ChươngTrìnhDaoTao với MonHoc



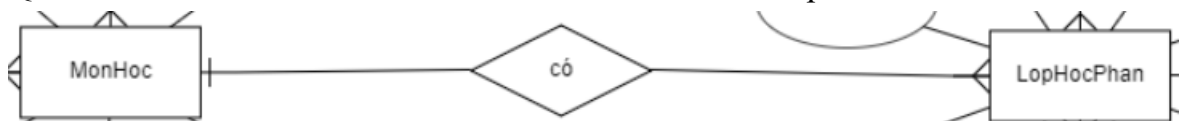
➔ Một chương trình đào tạo có nhiều môn học thuộc về nó

3. Quan hệ một-nhiều của thực thể ChươngTrìnhDaoTao với SinhVien



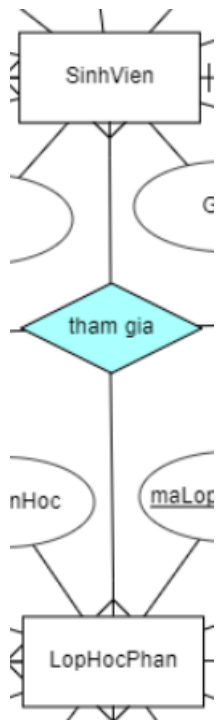
➔ Một chương trình đào tạo được đăng ký bởi nhiều sinh viên

4. Quan hệ một-nhiều của thực thể MonHoc với thực thể LopHocPhan



➔ Một môn học có nhiều lớp học phần thuộc về nó

5. Quan hệ nhiều-nhiều của thực thể SinhVien với thực thể LopHocPhan



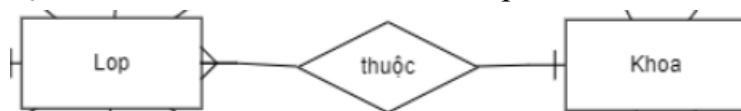
➔ Một sinh viên tham gia nhiều lớp học phần và ngược lại, một lớp học phần được tham gia bởi nhiều sinh viên

6. Quan hệ một-nhiều của thực thể SinhVien với thực thể Lop



➔ Một lớp có nhiều sinh viên

7. Quan hệ một-nhiều của thực thể Lop với thực thể Khoa



➔ Một khoa có nhiều lớp

8. Quan hệ một-nhiều của thực thể NhanVien và thực thể LopHocPhan



→ Một nhân viên giảng dạy nhiều lớp học phần

2. Xây dựng cơ sở dữ liệu với Oracle

2.1. Xây dựng các bảng dữ liệu

1. Bảng tblSinhVien đại diện cho thực thể SinhVien – Khóa chính: MSV (Mã sinh viên)

```
CREATE TABLE tblSinhVien (  
  MSV NUMBER PRIMARY KEY,  
  maLop VARCHAR2(10) NOT NULL,  
  maChuongTrinhDaoTao VARCHAR2(10) NOT NULL,  
  gioiTinh NUMBER(1) NOT NULL,  
  namNhapHoc NUMBER NOT NULL,  
  ngaySinh DATE NOT NULL,  
  ten NVARCHAR2(50) NOT NULL,  
  hoDem NVARCHAR2(50) NOT NULL,  
  GPA NUMBER NOT NULL,  
  CONSTRAINT fk_chuongtrinhdaotao_sv  
    FOREIGN KEY (maChuongTrinhDaoTao)  
    REFERENCES tblChuongTrinhDaoTao(maChuongTrinhDaoTao),  
  CONSTRAINT fk_lop_sv  
    FOREIGN KEY (maLop)  
    REFERENCES tblLop(maLop)  
);
```

2. Bảng tblChuongTrinhDaoTao đại diện cho thực thể ChuongTrinhDaoTao – Khóa chính: maChuongTrinhDaoTao

```
CREATE TABLE tblChuongTrinhDaoTao (  
  maChuongTrinhDaoTao VARCHAR2(10) PRIMARY KEY,  
  tenChuongTrinhDaoTao NVARCHAR2(100) NOT NULL,  
  soTinChi NUMBER NOT NULL,  
  namBatDau NUMBER NOT NULL  
);
```

3. Bảng tblNhanVien đại diện cho thực thể NhanVien – Khóa chính maNhanVien

```
CREATE TABLE tblNhanVien (  
  maNhanVien NUMBER PRIMARY KEY,  
  hoDem NVARCHAR2(50) NOT NULL,  
  ten NVARCHAR2(50) NOT NULL,  
  chucVu NVARCHAR2(100) NOT NULL,  
  ngaySinh DATE NOT NULL,  
  gioiTinh NUMBER(1) NOT NULL,  
  namCongTac NUMBER NOT NULL  
);
```

4. Bảng tblBoMon đại diện cho thực thể BoMon – Khóa chính maBoMon

```
CREATE TABLE tblBoMon (  
    maBoMon NUMBER PRIMARY KEY,  
    tenBoMon NVARCHAR2(50) NOT NULL,  
    moTa NVARCHAR2(500),  
    maNhanVien NUMBER NOT NULL,  
    CONSTRAINT fk_nhanvien_bomon  
        FOREIGN KEY (maNhanVien)  
        REFERENCES tblNhanVien(maNhanVien)  
);
```

5. Bảng tblMonHoc đại diện cho thực thể MonHoc – Khóa chính maMonHoc

```
CREATE TABLE tblMonHoc (  
    maMonHoc NUMBER PRIMARY KEY,  
    tenMonHoc NVARCHAR2(100) NOT NULL,  
    maBoMon NUMBER NOT NULL,  
    soTinChi NUMBER NOT NULL,  
    moTa NVARCHAR2(500),  
    maChuongTrinhDaoTao VARCHAR2(10) NOT NULL,  
  
    CONSTRAINT fk_mabomon_monhoc  
        FOREIGN KEY (maBoMon)  
        REFERENCES tblBoMon(maBoMon),  
  
    CONSTRAINT fk_machuongtrinhdaotao_monhoc  
        FOREIGN KEY (maChuongTrinhDaoTao)  
        REFERENCES tblChuongTrinhDaoTao(maChuongTrinhDaoTao)  
);
```

6. Bảng tblLopHocPhan đại diện cho thực thể LopHocPhan – Khóa chính maLopHocPhan

```
CREATE TABLE tblLopHocPhan (  
    maLopHocPhan NUMBER PRIMARY KEY,  
    tenLopHocPhan NVARCHAR2(100) NOT NULL,  
    soLuongSinhVien NUMBER NOT NULL,  
    moTa NVARCHAR2(500),  
    hocKy NUMBER NOT NULL,  
    namHoc NUMBER NOT NULL,  
    maMonHoc NUMBER NOT NULL,  
    maNhanVien NUMBER NOT NULL,  
  
    CONSTRAINT fk_monhoc_lop  
        FOREIGN KEY (maMonHoc)  
        REFERENCES tblMonHoc(maMonHoc),  
  
    CONSTRAINT fk_nhanvien_lop  
        FOREIGN KEY (maNhanVien)  
        REFERENCES tblNhanVien(maNhanVien)  
);
```

7. Bảng tblSinhVienTGLop – Được sinh ra từ mối quan hệ nhiều-nhiều giữa 2 bảng tblSinhVien và tblLop

```
CREATE TABLE tblSinhVienTGLop (  
  
    maLopHocPhan NUMBER NOT NULL,  
    MSV NUMBER NOT NULL,  
  
    diemQuaTrinh NUMBER,  
    diemThi NUMBER,  
  
    CONSTRAINT fk_malophocphan  
        FOREIGN KEY (maLopHocPhan)  
        REFERENCES tblLopHocPhan (maLopHocPhan),  
  
    CONSTRAINT fk_masinhvien  
        FOREIGN KEY (MSV)  
        REFERENCES tblSinhVien (MSV)  
);
```

8. Bảng tblKhoa đại diện cho thực thể Khoa – Khóa chính maKhoa

```
CREATE TABLE tblKhoa (  
    maKhoa VARCHAR2(10) PRIMARY KEY,  
    tenKhoa NVARCHAR2(100) NOT NULL,  
    ngayThanhLap DATE NOT NULL,  
    maNhanVien NUMBER NOT NULL,  
  
    CONSTRAINT fk_nhanvien_khoa  
        FOREIGN KEY (maNhanVien)  
        REFERENCES tblNhanVien (maNhanVien)  
);
```

9. Bảng tblLop đại diện cho thực thể Lop – Khóa chính maLop

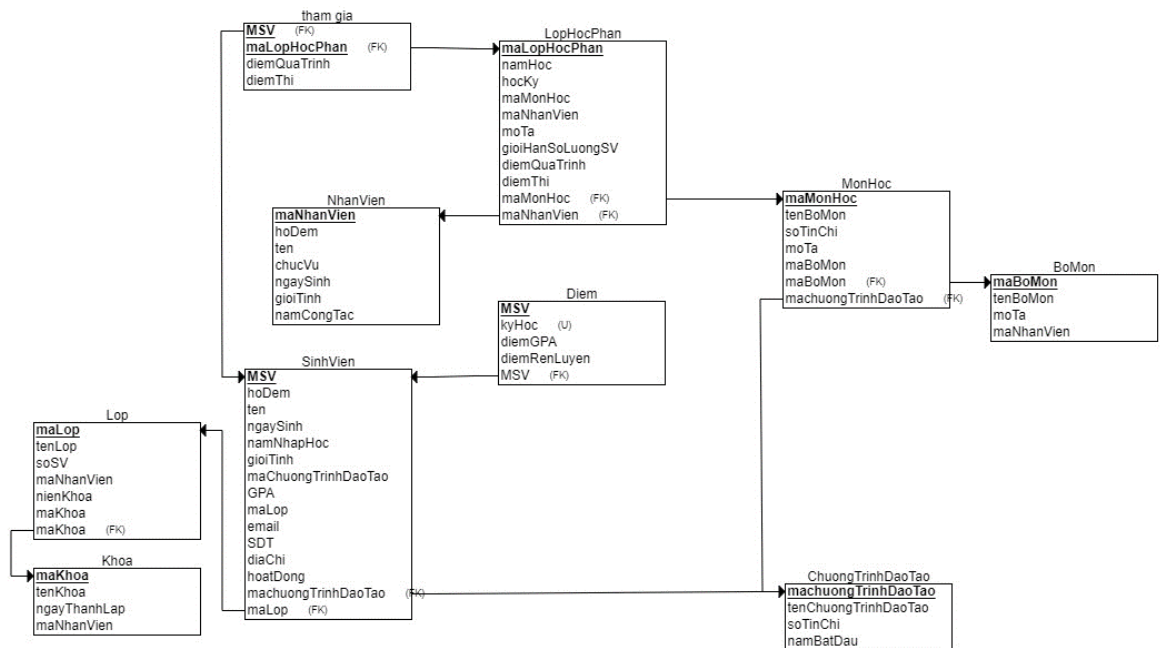
```
CREATE TABLE tblLop (  
    maLop VARCHAR2(10) PRIMARY KEY,  
    tenLop NVARCHAR2(100) NOT NULL,  
    nienKhoa VARCHAR2(5) NOT NULL,  
    soSV NUMBER NOT NULL,  
    maKhoa VARCHAR2(10) NOT NULL,  
    maNhanVien NUMBER NOT NULL,  
    CONSTRAINT fk_lop_khoa  
        FOREIGN KEY (maKhoa)  
        REFERENCES tblKhoa (maKhoa),  
    CONSTRAINT fk_lop_nhanvien  
        FOREIGN KEY (maNhanVien)  
        REFERENCES tblNhanVien (maNhanVien)  
);
```

10. Bảng tblDiem đại diện cho thực thể Diem – Khóa chính (MSV, kyHoc)

```
--tblDiem
CREATE TABLE tblDiem (
    MSV NUMBER NOT NULL,
    kyHoc NUMBER NOT NULL,
    diemGPA NUMBER(3, 2),
    diemRenLuyen NUMBER,
    CONSTRAINT pk_tblDiem PRIMARY KEY (MSV, kyHoc),
    CONSTRAINT fk_tblDiem_SinhVien FOREIGN KEY (MSV) REFERENCES tblSinhVien(MSV)
);
```


➔ Kéo thả các bảng trong database để tự sinh Diagram Quản lý sinh viên

- Mô hình lược đồ quan hệ tự sinh:



2.3. Tạo các ràng buộc:

1. Bảng tblSinhVien:

1.1. CHECK (GPA >= 0 AND GPA <= 4): Giới hạn điểm GPA từ 0 đến 4

1.2. Mã sinh viên có 9 chữ số

1.3. Trigger ngaySinh_tblSinhVien: Ngày sinh phải trước ngày hôm nay

```
--Ngày sinh phải trước ngày hôm nay
CREATE OR REPLACE TRIGGER ngaySinh_tblSinhVien
BEFORE INSERT ON tblSinhVien
FOR EACH ROW
DECLARE
BEGIN
    IF :NEW.ngaySinh >= SYSDATE THEN
        RAISE_APPLICATION_ERROR(-20001, 'Ngày sinh không hợp lệ');
    END IF;
END;
```

1.4. Trigger namNhapHoc_tblSinhVien: Giới hạn từ 20 năm trước đến năm nay

```

--namNhapHoc phai nho hon hoac bang nam nay va khong duoc nho hon qua 20 nam
CREATE OR REPLACE TRIGGER namNhapHoc_tblSinhVien
BEFORE INSERT ON tblSinhVien
FOR EACH ROW
DECLARE
    v_current_year NUMBER;
BEGIN
    -- Lay nam hien tai
    SELECT EXTRACT(YEAR FROM SYSDATE) INTO v_current_year FROM DUAL;

    -- Kiem tra dieu kien nam nhap hoc
    IF :NEW.namNhapHoc > v_current_year OR :NEW.namNhapHoc < v_current_year - 20 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Nam nhap hoc phai tu ' || v_current_year - 20 || ' den ' || v_current_year || '.');
    END IF;
END;

```

1.5. Trigger email sinh viên được tạo tự động dựa trên ten + MSV đuôi lms.utc.edu.vn

```

CREATE OR REPLACE TRIGGER trg_sinhvien_email
BEFORE INSERT OR UPDATE ON tblSinhVien
FOR EACH ROW
BEGIN
    :NEW.email := :NEW.ten || :NEW.MSV || '@lms.utc.edu.vn';
END;

UPDATE tblSinhVien
SET email = ten || MSV || '@lms.utc.edu.vn';

```

2. Bảng tblLop:

UNIQUE (tenLop): Tránh trùng tên lớp

CHECK (soSV >= 5 AND soSV <= 200): Giới hạn số lớp từ 5 đến 200 sinh viên

3. Bảng tblMonHoc:

CHECK (soTinChi >=1 AND soTinChi <= 10): Giới hạn số tín chỉ từ 1 đến 10

4. Bảng tblNhanVien

Trigger namCongTac_tblNhanVien: Năm công tác phải từ năm nay đổ về trước

```
--tblNhanVien
CREATE OR REPLACE TRIGGER namCongTac_tblNhanVien
BEFORE INSERT ON tblNhanVien
FOR EACH ROW
DECLARE
    v_current_year NUMBER;
BEGIN
    -- Lay nam hoc hien tai
    SELECT EXTRACT(YEAR FROM SYSDATE) INTO v_current_year FROM DUAL;

    -- Kiem tra dieu kien nam cong tac
    IF :NEW.namCongTac > v_current_year THEN
        RAISE_APPLICATION_ERROR(-20003, 'Nam cong tac khong hop li');
    END IF;
END;
```

5. Bảng tblDiem:

CHECK điểm quá trình và điểm thi cần phải nằm trong khoảng 0 đến 10

```
--tblSinhVienTGLop
-- Thêm ràng bu?c CHECK cho c?t diemQuaTrinh
ALTER TABLE tblSinhVienTGLop
ADD CONSTRAINT chk_diemQuaTrinh CHECK (diemQuaTrinh >= 0 AND diemQuaTrinh <= 10);

-- Thêm ràng buoc CHECK cho cot diemThi
ALTER TABLE tblSinhVienTGLop
ADD CONSTRAINT chk_diemThi CHECK (diemThi >= 0 AND diemThi <= 10);
```

2.4. Nhập dữ liệu

Dữ liệu demo cho các bảng dữ liệu. Mỗi bảng từ 5 trở lên

3. Truy vấn

3.1. Truy vấn cơ bản

1. Danh sách tất cả các sinh viên
SELECT * FROM tblSinhVien;
2. Danh sách các môn học và số tin chỉ
SELECT tenMonHoc, soTinChi FROM tblMonHoc;
3. Thông tin các học phần trong một năm học cụ thể
SELECT * FROM tblLopHocPhan WHERE namHoc = 2023;
4. Số lượng sinh viên theo từng lớp
SELECT maLop, COUNT(*) AS SoLuongSinhVien

```
FROM tblSinhVien
GROUP BY maLop;
```

5. Điểm GPA của sinh viên theo từng lớp

```
SELECT maLop, AVG(GPA) AS DiemTrungBinh
FROM tblSinhVien
GROUP BY maLop;
```

3.2. Truy vấn lồng nhau

➔ Truy vấn lồng nhau là quá trình thực hiện một truy vấn SQL bên trong một truy vấn SQL khác. Truy vấn ở mức bên trong được gọi là truy vấn con hoặc truy vấn lồng nhau.

1. Lấy họ và tên và GPA trong 1 lớp cụ thể:

```
SELECT hoDem || ' ' || ten AS "HoTen", GPA
FROM tblSinhVien
WHERE maLop = (SELECT maLop FROM tblLop WHERE tenLop =
'CNTT1-K62');
```

2. Liệt kê sinh viên và điểm của họ trong một môn học cụ thể

```
SELECT hoDem || ' ' || ten AS "HoTen", diemThi
FROM tblSinhVienTGLop
WHERE maLopHocPhan IN (SELECT maLopHocPhan FROM
tblLopHocPhan WHERE maMonHoc = (SELECT maMonHoc FROM
tblMonHoc WHERE tenMonHoc = '));
```

3. Danh sách các sinh viên theo lớp và được sắp xếp theo điểm GPA giảm dần

```
SELECT ten, GPA
FROM tblSinhVien
WHERE maLop = (SELECT maLop FROM tblLop WHERE tenLop =
'CNTT1-K62')
ORDER BY GPA DESC;
```

4. Thông tin các lớp học phần và giáo viên giảng dạy

```
SELECT tenLopHocPhan, hoDem || ' ' || ten AS "Ho Ten Giang vien"
FROM tblLopHocPhan
INNER JOIN tblNhanVien ON tblLopHocPhan.maNhanVien =
tblNhanVien.maNhanVien;
```

5. Danh sách các sinh viên và điểm GPA trong một học phần của một giáo viên cụ thể

```
SELECT SV.ten, SV.GPA, LHP.tenLopHocPhan
```

```

FROM tblSinhVien SV
JOIN tblSinhVienTGLop SVTL ON SV.MSV = SVTL.MSV
JOIN tblLopHocPhan LHP ON SVTL.maLopHocPhan =
LHP.maLopHocPhan
WHERE LHP.maNhanVien = (SELECT maNhanVien FROM tblNhanVien
WHERE ten = ' ');

```

3.3. Truy vấn nhóm

1. Tính điểm GPA trung bình theo từng mã lớp

```

SELECT maLop, AVG(GPA) AS avgGPA
FROM tblSinhVien
GROUP BY maLop;

```
2. Đếm số sinh viên từng khoa

```

SELECT maKhoa, COUNT(*) AS soSinhVien
FROM tblLop
GROUP BY maKhoa;

```
3. Tính số tín chỉ từng chương trình đào tạo

```

SELECT maChuongTrinhDaoTao, SUM(soTinChi) AS tongTinChi
FROM tblMonHoc
GROUP BY maChuongTrinhDaoTao;

```
4. Tính điểm trung bình từng môn học theo mỗi lớp học phần

```

SELECT maLopHocPhan, maMonHoc, AVG(diemThi) AS avgDiemThi
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan, maMonHoc;

```
5. Đếm số lượng sinh viên từng lớp học

```

SELECT maLopHocPhan, COUNT(MSV) AS soLuongSinhVien
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan;

```

3.4. Các hàm thống kê

1. Hàm 'COUNT': Đếm số lượng sinh viên trong từng lớp học

```

SELECT maLopHocPhan, COUNT(MSV) AS soLuongSinhVien
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan;

```
2. Hàm 'SUM': Tính tổng số tín chỉ của từng chương trình đào tạo

```

SELECT maChuongTrinhDaoTao, SUM(soTinChi) AS tongTinChi

```

- FROM tblMonHoc
GROUP BY maChuongTrinhDaoTao;
3. Hàm ‘AVG’: Tính điểm trung bình của từng môn học trong mỗi lớp học phần
SELECT maLopHocPhan, maMonHoc, AVG(COALESCE(diemThi, 0)) AS
avgDiemThi
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan, maMonHoc;
4. Hàm ‘MAX’: Tìm điểm thi cao nhất của mỗi lớp học phần:
SELECT maLopHocPhan, maMonHoc, MAX(COALESCE(diemThi, 0)) AS
maxDiemThi
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan, maMonHoc;
5. Hàm ‘MIN’: Tìm điểm thi thấp nhất trong mỗi lớp học phần
SELECT maLopHocPhan, maMonHoc, MIN(COALESCE(diemThi, 0)) AS
minDiemThi
FROM tblSinhVienTGLop
GROUP BY maLopHocPhan, maMonHoc;

4. Lập trình chương trình PL/SQL

1. Tính GPA trung bình toàn khóa của sinh viên dựa trên dữ liệu trong bảng Điểm và cập nhật vào thông tin Sinh Viên

```
--1. Tính GPA của mọi sinh viên dựa vào bảng Điểm và cập nhật lại vào bảng tblSinhVien
CREATE OR REPLACE PROCEDURE tinhGPADiem(p_msv NUMBER) AS
    v_diemGPA NUMBER;
BEGIN
    SELECT AVG(COALESCE(diemGPA, 0))
    INTO v_diemGPA
    FROM tblDiem
    WHERE MSV = p_msv
    GROUP BY MSV;

    -- Update điểm GPA trung bình của sinh viên trong bảng tblSinhVien
    UPDATE tblSinhVien
    SET GPA = v_diemGPA
    WHERE MSV = p_msv;

    COMMIT;
END tinhGPADiem;
/
```

```

BEGIN
    tinhGPADiem(211213123); -- Thay 123 bằng mã số sinh viên cần tính GPA
END;
/

select * from tblSinhVien where tblSinhVien.MSV = 211213123;

```

2. Cập nhật tên lớp cho sinh viên

```

CREATE OR REPLACE PROCEDURE capNhatTenLop(maLop
VARCHAR2, tenLopMoi NVARCHAR2) IS
BEGIN
    UPDATE tblLop
    SET tenLop = tenLopMoi
    WHERE maLop = maLop;
    COMMIT;
END;
/

```

3. Hàm: Lấy danh sách các sinh viên theo khoa

```

CREATE OR REPLACE FUNCTION
layDanhSachSinhVienTheoKhoa(maKhoa VARCHAR2) RETURN
SYS_REFCURSOR IS
    cur SYS_REFCURSOR;
BEGIN
    OPEN cur FOR
        SELECT MSV, ten, maLop
        FROM tblSinhVien S
        JOIN tblLop L ON S.maLop = L.maLop
        WHERE L.maKhoa = maKhoa;
    RETURN cur;
END;
/

```

4. Thủ tục: Xóa tất cả điểm của sinh viên

```

CREATE OR REPLACE PROCEDURE xoaDiemSinhVien(maSV NUMBER)
IS
BEGIN
    DELETE FROM tblDiem
    WHERE MSV = maSV;
    COMMIT;

```


END;

/

5. Hàm: Tính điểm trung bình của lớp học phần

CREATE OR REPLACE FUNCTION

tinhdienTrungBinhLHP(maLopHocPhan NUMBER) RETURN NUMBER IS

totalDien NUMBER := 0;

soSinhVien NUMBER := 0;

BEGIN

FOR rec IN (SELECT diemThi FROM tblSinhVienTGLop WHERE
maLopHocPhan = maLopHocPhan) LOOP

totalDien := totalDien + rec.diemThi;

soSinhVien := soSinhVien + 1;

END LOOP;

IF soSinhVien = 0 THEN

RETURN 0; -- Tránh chia cho 0

ELSE

RETURN totalDien / soSinhVien;

END IF;

END;

/

5. Quản trị CSDL Oracle

5.1. Quản lý Instance

- a. Chế Độ Nomount:

- Trong chế độ này, Oracle Database sẽ mở các tiến trình như instance, nhưng không mở bất kỳ tệp dữ liệu nào.

- Sử dụng lệnh sau: STARTUP NOMOUNT;

- b. Chế Độ Mount:

- Trong chế độ này, Oracle Database sẽ mở các tệp dữ liệu control file và thiết lập các tham số instance, nhưng không mở cơ sở dữ liệu.

- Sử dụng lệnh sau: STARTUP MOUNT;

c. Chế Độ Open:

- Trong chế độ này, Oracle Database sẽ mở tất cả các tệp dữ liệu và trở thành sẵn sàng để chấp nhận các kết nối.

- Sử dụng lệnh sau: `STARTUP OPEN;`

2. Tắt Instance:

a. Tắt Bình Thường:

- Sử dụng lệnh sau để tắt instance một cách bình thường: `SHUTDOWN;`

b. Tắt Với Option IMMEDIATE:

- Sử dụng lệnh sau để tắt instance ngay lập tức, không chờ đợi các kết nối kết thúc: `SHUTDOWN IMMEDIATE;`

c. Tắt Với Option TRANSACTIONAL:

- Sử dụng lệnh sau để tắt instance sau khi tất cả các giao dịch kết thúc: `SHUTDOWN TRANSACTIONAL;`

3. Chế Độ Read-Only và Restricted Session:

a. Chế Độ Read-Only:

- Sử dụng lệnh sau để chuyển instance sang chế độ chỉ đọc: `ALTER DATABASE OPEN READ ONLY;`

b. Chế Độ Restricted Session:

- Sử dụng lệnh sau để mở instance trong chế độ hạn chế, chỉ cho phép người dùng SYSDBA hoặc SYSOPER kết nối:

`ALTER SYSTEM ENABLE RESTRICTED SESSION;`

- Để tắt chế độ hạn chế, sử dụng:

`ALTER SYSTEM DISABLE RESTRICTED SESSION;`

5.2. Quản lý Table Space

1. Thêm Tablespace:

- **Tạo permanent tablespace tên nguoidung gồm 2 datafile**

Create tablespace nguoidung datafile ‘%oracle_home%\oradata\usedata1.dbf’ size 10M, ‘%oracle_home%\oradata\usedata2.dbf’ size 20M;

- **Tạo temporary tablespace tên tempo**
Create temporary tablespace tempo tempfile
‘%oracle_home%\oradata\tempo.dbf’ size 100M;
 - **Tạo undo tablespace tên undotbl**
Create undo tablespace undotbl datafile ‘%oracle_home%\oradata\undotbl.dbf’
size 100M;
2. Sửa Tablespace
- Thay đổi kích thước Datafile
ALTER DATABASE DATAFILE ‘%oracle_home%\oradata\usedata1.dbf’
RESIZE 200M;
 - Bật tự động mở rộng:
ALTER DATABASE DATAFILE ‘%oracle_home%\oradata\usedata1.dbf’
AUTOEXTEND ON NEXT 50M MAXSIZE 500M;
 - Tắt tự động mở rộng:
ALTER DATABASE DATAFILE ‘%oracle_home%\oradata\usedata1.dbf’
AUTOEXTEND OFF;
3. Xóa Tablespace
- DROP TABLESPACE nguoidung INCLUDING CONTENTS AND
DATAFILES;
4. Đổi tên hoặc thay đổi vị trí của datafiles
- ALTER TABLE SPACE
 - Tablespace phải offline, database phải open.
 - Đổi tên hoặc di chuyển datafiles
 ALTER TABLESPACE userdata RENAME
DATAFILE 'u01/oradata/userdata01.dbf'
TO 'u02/oradata/userdata01.dbf';
 - ALTER DATABASE
 - Hoặc CSDL phải mounted, hoặc Tablespace phải offline nếu CSDL đang open.
 - Đổi tên hoặc di chuyển datafiles
 ALTER DATABASE RENAME
FILE 'u01/oradata/system01.dbf'
TO 'u03/oradata/system01.dbf';
5. Khôi phục datafiles bị mất
- B1: Tạo lại datafile:

Alter database create datafile full_path_file_name;

B2: Khôi phục dữ liệu:

Recover datafile full_path_file_name;

6. Truy vấn thông tin Tablespace và Datafiles

a. Xem Danh Sách Tablespace:

SELECT tablespace_name FROM dba_tablespaces;

b. Xem Danh Sách Datafiles:

SELECT file_name FROM dba_data_files;

c. Xem Kích Thước và Tình Trạng Tự Động Mở Rộng:

SELECT tablespace_name, file_name, bytes, autoextensible, maxbytes
FROM dba_data_files;

d. Xem Tình Trạng Tự Động Mở Rộng Của Một Datafile Cụ Thể:

SELECT file_name, autoextensible, maxbytes

FROM dba_data_files

WHERE tablespace_name = 'ten_tablespace';

5.3. Truy vấn thông tin về cấu trúc lưu trữ

1. Xem thông tin Segment:

SELECT segment_name, segment_type, bytes, extents
FROM user_segments;

2. Xem thông tin Extent của một Segment cụ thể

SELECT segment_name, extent_id, file_id, block_id, bytes
FROM user_extents
WHERE segment_name = 'TEN_SEGMENT';

3. Hiển thị thông tin về các segment được sở hữu bởi user SCOTT

SELECT segment_name, tablespace_name, blocks FROM dba_segments
WHERE owner = 'SCOTT'

5.4. Quản trị người dùng, quyền, chức danh

1. Xem thông tin người dùng

SELECT username, account_status, created
FROM dba_users;

2. Xem quyền cụ thể của người dùng:

SELECT grantee, privilege
FROM dba_sys_privs
WHERE grantee = 'TEN_NGUOI_DUNG';

