**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**



**BÀI TẬP LỚN**

**HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI ORACLE**

**ĐỀ SỐ 10: XÂY DỰNG CƠ SỞ DỮ LIỆU HỆ THỐNG QUẢN LÝ SINH VIÊN**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sinh viên thực hiện** | **Khóa** | **Lớp** | **Mã sinh viên** |
| **Nguyễn Ngọc Minh** | **K12** | **DCCNTT12.10.4** | **20211084** |
| **Nguyễn Văn Lâm** | **K12** | **DCCNTT12.10.4** | **20211166** |
| **Lê Quý Mùi** | **K12** | **DCCNTT12.10.4** | **20211133** |

**Hà Nội, tháng 10 năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**BÀI TẬP LỚN**

**HỌC PHẦN: HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU VỚI ORACLE**

**ĐỀ SỐ 10: XÂY DỰNG CƠ SỞ DỮ LIỆU HỆ THỐNG QUẢN LÝ SINH VIÊN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Khóa** | **Lớp** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** | **Ký tên SV** |
| **1** | **Nguyễn Ngọc Minh** | **K12** | **DCCNTT12.10.4** | **20211084** |  |  |  |
| **2** | **Nguyễn Văn Lâm** | **K12** | **DCCNTT12.10.4** | **20211166** |  |  |  |
| **3** | **Lê Quý Mùi** | **K12** | **DCCNTT12.10.4** | **20211133** |  |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM 1**  *(Ký và ghi rõ họ tên)* | **CÁN BỘ CHẤM 2**  *(Ký và ghi rõ họ tên)* |

**Hà Nội, tháng 10 Năm 2023**

Mục lục

[Lời mở đầu 4](#_Toc149134041)

[Lời cảm ơn 5](#_Toc149134042)

[CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 6](#_Toc149134043)

[1.1 Giới Thiệu Đề Tài 6](#_Toc149134044)

[1.2 Các Thực Thể 6](#_Toc149134045)

[1.3 Mối Quan Hệ Thực Thể 11](#_Toc149134046)

[CHƯƠNG II: XÂY DỰNG CƠ SỞ DỮ LIỆU 14](#_Toc149134047)

[2.1 Thiết Kế Các Bảng 14](#_Toc149134048)

[2.2 Xây Dựng Các Ràng Buộc 17](#_Toc149134049)

[2.3 Nhập Dữ Liệu 18](#_Toc149134050)

[2.4 Code Trên SQL 32](#_Toc149134051)

[Chương III: Thiết kế các truy vấn 35](#_Toc149134052)

[3.1 Truy vấn cơ bản 35](#_Toc149134053)

[3.2 Truy vấn lồng nhau 35](#_Toc149134054)

[3.3 Truy vấn gộp nhóm 36](#_Toc149134055)

[3.4 Truy vấn nâng cao 38](#_Toc149134056)

[Chương 4: Lập trình PL/SQL 39](#_Toc149134057)

[4.1 Một số thủ tục 40](#_Toc149134058)

[4.2 Câu lệnh IF - ELSE 42](#_Toc149134059)

[4.3 Một số hàm 44](#_Toc149134060)

[Chương 5: Quản trị cơ sở dữ liệu Oracle 45](#_Toc149134061)

[5.1 Quản trị Instance 45](#_Toc149134062)

[5.2 Quản trị tablespace 47](#_Toc149134063)

[5.3 Quản trị người dùng 49](#_Toc149134064)

[5.4 Sao lưu, phục hồi 51](#_Toc149134065)

[Kết Luận 56](#_Toc149134066)

[Tài liệu tham khảo 57](#_Toc149134067)

# 

# Lời mở đầu

Trong thời đại công nghệ thông tin phát triển như hiện nay, các hệ thống quản lý bằng máy tính đang dần thay thế cho các phương pháp quản lý truyền thống. Trong đó, hệ thống quản lý sinh viên là một trong những hệ thống quan trọng nhất của các trường học. Hệ thống này giúp nhà trường quản lý thông tin sinh viên một cách hiệu quả và khoa học, từ đó nâng cao chất lượng đào tạo.

Tại các trường đại học , hệ thống quản lý sinh viên đang được sử dụng rộng rãi trong các hoạt động đào tạo, giảng dạy và nghiên cứu. Tuy nhiên, hệ thống này vẫn còn tồn tại một số hạn chế như:

* Cơ sở dữ liệu chưa đáp ứng đầy đủ các yêu cầu quản lý sinh viên hiện nay.
* Hệ thống chưa tích hợp với các hệ thống khác của nhà trường.
* Giao diện người dùng chưa thân thiện với người dùng.

Để khắc phục những hạn chế trên, tiểu luận này đề cập đến việc xây dựng cơ sở dữ liệu cho hệ thống quản lý sinh viên bằng Oracle. Mục tiêu của tiểu luận là xây dựng một cơ sở dữ liệu đáp ứng đầy đủ các yêu cầu quản lý sinh viên của nhà trường, đồng thời tích hợp với các hệ thống khác của nhà trường và có giao diện người dùng thân thiện với người dùng.

# Lời cảm ơn

Là sinh viên năm 3 khoa công nghệ thông tin của trường đại học công nghệ đông á, nhóm em nhận thấy rằng hệ thống quản lý sinh viên của nhà trường đóng vai trò quan trọng trong việc quản lý và đào tạo sinh viên. Hệ thống này giúp nhà trường quản lý thông tin sinh viên một cách đầy đủ và chính xác, từ đó giúp nhà trường đưa ra những quyết định chính xác trong công tác đào tạo và quản lý sinh viên.

Chính vì vậy, nhóm em đã chọn đề tài "Xây dựng cơ sở dữ liệu hệ thống quản lý sinh viên bằng Oracle" để thực hiện tiểu luận của mình. Em hy vọng rằng tiểu luận này sẽ góp phần nâng cao chất lượng của hệ thống quản lý sinh viên của nhà trường.

Em xin chân thành cảm ơn cô Trần Thị Thúy Hằngđã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện tiểu luận này. Em cũng xin cảm ơn các thầy/cô giáo trong khoa công nghệ thông tin đã giúp đỡ em trong quá trình học tập và nghiên cứu.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến cha mẹ, anh chị em, bạn bè đã luôn động viên và ủng hộ em trong suốt quá trình học tập và nghiên cứu.

# CHƯƠNG I. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## 1.1 Giới Thiệu Đề Tài

Mục Tiêu:

* Mục tiêu của dự án là xây dựng một hệ thống quản lý sinh viên hiệu quả, giúp quản lý thông tin cá nhân của sinh viên, thông tin học tập, và mối quan hệ với giảng viên.

Lý Do Chọn Đề Tài:

* Lý do chọn đề tài này là để giúp trường học/quản lý giảng dạy dễ dàng theo dõi tiến trình học tập của sinh viên, cung cấp thông tin chi tiết về mỗi sinh viên, và tối ưu hóa quá trình quản lý hồ sơ sinh viên.

## 1.2 Các Thực Thể

* Trong hệ thống này, có một số thực thể quan trọng bao gồm:
* Người dùng : Thực thể này chứa thông tin cá nhân của người dùng bao gồm: tên tài khoản , mật khẩu , mã xác nhận , email.

A diagram of a computer

Description automatically generated

Hình 2.1: Thực thể người dùng

* Sinh viên: Thực thể này chứa thông tin cá nhân của sinh viên, bao gồm: tên, mã số sinh viên, ngày tháng năm sinh, địa chỉ, số điện thoại , lớp ,giới tính

A diagram of a company

Description automatically generated with medium confidence

Hình 2.2: Thực thể sinh viên

* Giảng viên: Thực thể này lưu trữ thông tin về giảng viên, bao gồm: tên, mã giảng viên, mã môn học , số điện thoại và giới tính.

A diagram of a company

Description automatically generated

Hình 2.3: Thực thể giảng viên

* Đề tài: Thực thể này đại diện cho thông tin về các đề tài, bao gồm: mã đề tài, tên đề tài, và giới hạn .

A diagram of a structure

Description automatically generated

Hình 2.4: Thực thể đề tài

* Môn học: Thực thể này chứa thông tin về các môn học, bao gồm tên môn học, mã môn học, số tín chỉ, giá tiền , và tổng số tiền .

A diagram of a network

Description automatically generated

Hình 2.5: Thực thể môn học

* Bảng điểm: Thực thể này đại diện cho thông tin về bảng điểm , bao gồm: điểm lần 1 điểm lần 2 , nhận xét , mã sinh viên , và mã đề tài .

A diagram of a network with Ice hockey rink in the background

Description automatically generated

Hình 2.6: Thực thể bảng điểm

* Điểm môn học : Thực thể này đại diện cho thông tin về điểm môn học bao gồm: mã sinh viên , mã môn học , điểm kiểm tra , điểm cuối kì , điểm tổng kết

A diagram of a network

Description automatically generated

Hình 2.7: Thực thể bảng điểm môn học

## 1.3 Mối Quan Hệ Thực Thể

* Mối quan hệ giữa bảng "nguoidung" và bảng "sinhvien":
* Một người dùng có thể có thông tin liên quan đến một sinh viên thông qua một trường khóa ngoại (khóa ngoại ma\_SV trong bảng "sinhvien").
* Mối quan hệ giữa bảng "sinhvien" và bảng "detai":
* Mỗi sinh viên có thể tham gia nhiều đề tài thông qua bảng "bangdiem" (khóa ngoại ma\_SV trong bảng "bangdiem").
* Một đề tài có thể được thực hiện bởi nhiều sinh viên thông qua bảng "bangdiem" (khóa ngoại ma\_DT trong bảng "bangdiem").
* Mối quan hệ giữa bảng "detai" và bảng "monhoc":
* Mỗi đề tài có một môn học liên quan thông qua bảng "giangvien" (khóa ngoại ma\_MH trong bảng "giangvien").
* Một môn học có thể có nhiều đề tài thông qua bảng "detai" (khóa ngoại ma\_DT trong bảng "detai").
* Mối quan hệ giữa bảng "sinhvien" và bảng "monhoc" thông qua bảng "diemMH":
* Một sinh viên có thể có điểm của nhiều môn học thông qua bảng "diemMH" (khóa ngoại ma\_SV trong bảng "diemMH").
* Một môn học có thể có nhiều sinh viên và điểm thông qua bảng "diemMH" (khóa ngoại ma\_MH trong bảng "diemMH").
* Mối quan hệ giữa bảng "giangvien" và bảng "monhoc":
* Một giảng viên có thể giảng dạy một môn học thông qua bảng "monhoc" (khóa ngoại ma\_MH trong bảng "giangvien").
* Một môn học có thể có nhiều giảng viên thông qua bảng "giangvien" (khóa ngoại ma\_MH trong bảng "giangvien").

**Mô hình quan hệ thực thể :**

**A diagram of a data flow

Description automatically generated**

Hình 3: Mô hình quan hệ thực thể

# CHƯƠNG II: XÂY DỰNG CƠ SỞ DỮ LIỆU

## 2.1 Thiết Kế Các Bảng

* Dựa trên các thực thể và mối quan hệ trên, ta có thể thiết kế các bảng cơ sở dữ liệu như sau:
* Bảng sinh viên: ma\_SV, hoten,SDT, NgaySinh, DiaChi, Lop, GioiTinh
* Bảng người dùng: usename, password, confirmpw, Email
* Bảng đề tài: ma\_DT, TenDT, gioihan
* Bảng môn học: ma\_MH, TenMH, MaMon, SoTinChi, GiaTien, TongTien
* Bảng điểm : diemlan1, diemlan2, nhanxet , ma\_SV , ma\_DT
* Bảng điểm môn học: ma\_SV, ma\_MH, diemcc, diemkt, diemtk

Ta có các bảng :

A black and white rectangular box with black text

Description automatically generated

Hình 1.1: Bảng người dùng

A white rectangular box with black text

Description automatically generated

Hình 1.2: Bảng sinh viên

A table with black text

Description automatically generated with medium confidence

Hình 1.3: Bảng môn học

A close-up of a sign

Description automatically generated

Hình 1.4: Bảng đề tài

A white rectangular box with black text

Description automatically generated

Hình 1.5: Bảng giảng viên

A white rectangular object with black text

Description automatically generated

Hình 1.6: Bảng điểm

A white rectangular object with black text

Description automatically generated

Hình 1.2: Bảng điểm môn học

## 2.2 Xây Dựng Các Ràng Buộc

* **Bảng nguoidung (người dùng):**

usename, password, comfirmpw, và email không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là trường usename, đảm bảo sự duy nhất của mỗi người dùng.

* **Bảng sinhvien (sinh viên):**

Tất cả các trường không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là trường ma\_SV, đảm bảo sự duy nhất của mỗi sinh viên.

* **Bảng detai (đề tài):**

ma\_DT, tenDT, và gioihan không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là trường ma\_DT, đảm bảo sự duy nhất của mỗi đề tài.

* **Bảng bangdiem (bảng điểm):**

diemlan1 và diemlan2 không thể trống (NOT NULL).

nhanxet, ma\_SV, và ma\_DT không thể trống (NOT NULL).

ma\_SV và ma\_DT là các khóa ngoại (FOREIGN KEY) tham chiếu đến bảng sinhvien và detai, đảm bảo ràng buộc liên quan đến sinh viên và đề tài, tạo mô hình liên kết 1-N giữa sinh viên và bảng điểm.

* **Bảng monhoc (môn học):**

Tất cả các trường không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là trường ma\_MH, đảm bảo sự duy nhất của mỗi môn học.

* **Bảng diemMH (điểm môn học):**

ma\_SV, ma\_MH, diemcc, diemkt, và diemtk không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là cặp trường ma\_SV và ma\_MH, đảm bảo sự duy nhất của điểm cho mỗi sinh viên trong từng môn học.

ma\_SV và ma\_MH là các khóa ngoại (FOREIGN KEY) tham chiếu đến bảng sinhvien và monhoc, đảm bảo ràng buộc liên quan đến sinh viên và môn học, tạo mô hình liên kết 1-N giữa sinh viên và bảng điểm môn học.

* **Bảng giangvien (giảng viên):**

ma\_GV, hoten, ma\_MH, sdt, và gioitinh không thể trống (NOT NULL).

Khóa chính (PRIMARY KEY) là trường ma\_GV, đảm bảo sự duy nhất của mỗi giảng viên.

ma\_MH là một khóa ngoại (FOREIGN KEY) tham chiếu đến bảng monhoc, đảm bảo ràng buộc liên quan đến môn học, tạo mô hình liên kết 1-N giữa giảng viên và môn học.

## 2.3 Nhập Dữ Liệu

* **Bảng Người Dùng:**

DECLARE

usename1 VARCHAR2(40) := 'nguyen\_user1';

password1 VARCHAR2(40) := 'password123';

comfirmpw1 VARCHAR2(40) := 'password123';

email1 VARCHAR2(50) := 'nguyen@example.com';

usename2 VARCHAR2(40) := 'dung\_user';

password2 VARCHAR2(40) := 'password1232';

comfirmpw2 VARCHAR2(40) := 'password1232';

email2 VARCHAR2(50) := 'dung@example.com';

usename3 VARCHAR2(40) := 'user3';

password3 VARCHAR2(40) := 'pass3';

comfirmpw3 VARCHAR2(40) := 'pass3';

email3 VARCHAR2(50) := 'user3@example.com';

usename4 VARCHAR2(40) := 'user4';

password4 VARCHAR2(40) := 'pass4';

comfirmpw4 VARCHAR2(40) := 'pass4';

email4 VARCHAR2(50) := 'user4@example.com';

usename5 VARCHAR2(40) := 'user5';

password5 VARCHAR2(40) := 'pass5';

comfirmpw5 VARCHAR2(40) := 'pass5';

email5 VARCHAR2(50) := 'user5@example.com';

BEGIN

-- Insert user 1

INSERT INTO nguoidung (usename, password, comfirmpw, email)

VALUES (usename1, password1, comfirmpw1, email1);

dbms\_output.put\_line('Thêm người dùng 1 thành công!');

-- Insert user 2

INSERT INTO nguoidung (usename, password, comfirmpw, email)

VALUES (usename2, password2, comfirmpw2, email2);

dbms\_output.put\_line('Thêm người dùng 2 thành công!');

-- Insert user 3

INSERT INTO nguoidung (usename, password, comfirmpw, email)

VALUES (usename3, password3, comfirmpw3, email3);

dbms\_output.put\_line('Thêm người dùng 3 thành công!');

-- Insert user 4

INSERT INTO nguoidung (usename, password, comfirmpw, email)

VALUES (usename4, password4, comfirmpw4, email4);

dbms\_output.put\_line('Thêm người dùng 4 thành công!');

-- Insert user 5

INSERT INTO nguoidung (usename, password, comfirmpw, email)

VALUES (usename5, password5, comfirmpw5, email5);

dbms\_output.put\_line('Thêm người dùng 5 thành công!');

END;

* **Bảng Sinh Viên:**

DECLARE

hoten1 VARCHAR2(100) := 'Nguyễn Văn A';

ma\_SV1 VARCHAR(10) := 'SV001';

sdt1 VARCHAR(20) := '0123456789';

diachi1 VARCHAR2(100) := '123 Đường A, Quận 1, TP.HCM';

ngaysinh1 DATE := TO\_DATE('2000-01-15', 'YYYY-MM-DD');

lop1 VARCHAR(100) := 'K18-01';

gioitinh1 VARCHAR2(10) := 'Nam';

hoten2 VARCHAR2(100) := 'Trần Thị B';

ma\_SV2 VARCHAR(10) := 'SV002';

sdt2 VARCHAR(20) := '0987654321';

diachi2 VARCHAR2(100) := '456 Đường B, Quận 2, TP.HCM';

ngaysinh2 DATE := TO\_DATE('1999-05-20', 'YYYY-MM-DD');

lop2 VARCHAR(100) := 'K18-02';

gioitinh2 VARCHAR2(10) := 'Nữ';

hoten3 VARCHAR2(100) := 'Lê Văn C';

ma\_SV3 VARCHAR(10) := 'SV003';

sdt3 VARCHAR(20) := '0333333333';

diachi3 VARCHAR2(100) := '789 Đường C, Quận 3, TP.HCM';

ngaysinh3 DATE := TO\_DATE('2001-11-10', 'YYYY-MM-DD');

lop3 VARCHAR(100) := 'K18-03';

gioitinh3 VARCHAR2(10) := 'Nam';

hoten4 VARCHAR2(100) := 'Phạm Thị D';

ma\_SV4 VARCHAR(10) := 'SV004';

sdt4 VARCHAR(20) := '0555555555';

diachi4 VARCHAR2(100) := '101 Đường D, Quận 4, TP.HCM';

ngaysinh4 DATE := TO\_DATE('2002-07-05', 'YYYY-MM-DD');

lop4 VARCHAR(100) := 'K18-04';

gioitinh4 VARCHAR2(10) := 'Nữ';

hoten5 VARCHAR2(100) := 'Nguyễn Văn E';

ma\_SV5 VARCHAR(10) := 'SV005';

sdt5 VARCHAR(20) := '0777777777';

diachi5 VARCHAR2(100) := '202 Đường E, Quận 5, TP.HCM';

ngaysinh5 DATE := TO\_DATE('1998-03-30', 'YYYY-MM-DD');

lop5 VARCHAR(100) := 'K18-05';

gioitinh5 VARCHAR2(10) := 'Nam';

BEGIN

-- Insert student 1

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten1, ma\_SV1, sdt1, diachi1, ngaysinh1, lop1, gioitinh1);

dbms\_output.put\_line('Thêm sinh viên 1 thành công!');

-- Insert student 2

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten2, ma\_SV2, sdt2, diachi2, ngaysinh2, lop2, gioitinh2);

dbms\_output.put\_line('Thêm sinh viên 2 thành công!');

-- Insert student 3

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten3, ma\_SV3, sdt3, diachi3, ngaysinh3, lop3, gioitinh3);

dbms\_output.put\_line('Thêm sinh viên 3 thành công!');

-- Insert student 4

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten4, ma\_SV4, sdt4, diachi4, ngaysinh4, lop4, gioitinh4);

dbms\_output.put\_line('Thêm sinh viên 4 thành công!');

-- Insert student 5

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten5, ma\_SV5, sdt5, diachi5, ngaysinh5, lop5, gioitinh5);

dbms\_output.put\_line('Thêm sinh viên 5 thành công!');

END;

* **Bảng Đề Tài**

DECLARE

ma\_DT1 VARCHAR(10) := 'DT001';

tenDT1 VARCHAR2(255) := 'Đề tài 1';

gioihan1 VARCHAR(50) := '100 giờ';

ma\_DT2 VARCHAR(10) := 'DT002';

tenDT2 VARCHAR2(255) := 'Đề tài 2';

gioihan2 VARCHAR(50) := '80 giờ';

ma\_DT3 VARCHAR(10) := 'DT003';

tenDT3 VARCHAR2(255) := 'Đề tài 3';

gioihan3 VARCHAR(50) := '120 giờ';

ma\_DT4 VARCHAR(10) := 'DT004';

tenDT4 VARCHAR2(255) := 'Đề tài 4';

gioihan4 VARCHAR(50) := '90 giờ';

ma\_DT5 VARCHAR(10) := 'DT005';

tenDT5 VARCHAR2(255) := 'Đề tài 5';

gioihan5 VARCHAR(50) := '110 giờ';

BEGIN

-- Insert project 1

INSERT INTO detai (ma\_DT, tenDT, gioihan)

VALUES (ma\_DT1, tenDT1, gioihan1);

dbms\_output.put\_line('Thêm đề tài 1 thành công!');

-- Insert project 2

INSERT INTO detai (ma\_DT, tenDT, gioihan)

VALUES (ma\_DT2, tenDT2, gioihan2);

dbms\_output.put\_line('Thêm đề tài 2 thành công!');

-- Insert project 3

INSERT INTO detai (ma\_DT, tenDT, gioihan)

VALUES (ma\_DT3, tenDT3, gioihan3);

dbms\_output.put\_line('Thêm đề tài 3 thành công!');

-- Insert project 4

INSERT INTO detai (ma\_DT, tenDT, gioihan)

VALUES (ma\_DT4, tenDT4, gioihan4);

dbms\_output.put\_line('Thêm đề tài 4 thành công!');

-- Insert project 5

INSERT INTO detai (ma\_DT, tenDT, gioihan)

VALUES (ma\_DT5, tenDT5, gioihan5);

dbms\_output.put\_line('Thêm đề tài 5 thành công!');

END;

* **Bảng Điểm**

DECLARE

diemlan1\_1 FLOAT := 8.5;

diemlan2\_1 FLOAT := 9.2;

nhanxet1 VARCHAR2(255) := 'Rất xuất sắc';

ma\_SV1 VARCHAR(10) := 'SV001';

ma\_DT1 VARCHAR(10) := 'DT001';

diemlan1\_2 FLOAT := 7.8;

diemlan2\_2 FLOAT := 8.6;

nhanxet2 VARCHAR2(255) := 'Khá tốt';

ma\_SV2 VARCHAR(10) := 'SV002';

ma\_DT2 VARCHAR(10) := 'DT002';

diemlan1\_3 FLOAT := 9.0;

diemlan2\_3 FLOAT := 9.5;

nhanxet3 VARCHAR2(255) := 'Xuất sắc';

ma\_SV3 VARCHAR(10) := 'SV003';

ma\_DT3 VARCHAR(10) := 'DT003';

diemlan1\_4 FLOAT := 8.0;

diemlan2\_4 FLOAT := 8.2;

nhanxet4 VARCHAR2(255) := 'Tốt';

ma\_SV4 VARCHAR(10) := 'SV004';

ma\_DT4 VARCHAR(10) := 'DT004';

diemlan1\_5 FLOAT := 9.4;

diemlan2\_5 FLOAT := 9.8;

nhanxet5 VARCHAR2(255) := 'Xuất sắc';

ma\_SV5 VARCHAR(10) := 'SV005';

ma\_DT5 VARCHAR(10) := 'DT005';

BEGIN

-- Insert score 1

INSERT INTO bangdiem (diemlan1, diemlan2, nhanxet, ma\_SV, ma\_DT)

VALUES (diemlan1\_1, diemlan2\_1, nhanxet1, ma\_SV1, ma\_DT1);

dbms\_output.put\_line('Thêm bảng điểm 1 thành công!');

-- Insert score 2

INSERT INTO bangdiem (diemlan1, diemlan2, nhanxet, ma\_SV, ma\_DT)

VALUES (diemlan1\_2, diemlan2\_2, nhanxet2, ma\_SV2, ma\_DT2);

dbms\_output.put\_line('Thêm bảng điểm 2 thành công!');

-- Insert score 3

INSERT INTO bangdiem (diemlan1, diemlan2, nhanxet, ma\_SV, ma\_DT)

VALUES (diemlan1\_3, diemlan2\_3, nhanxet3, ma\_SV3, ma\_DT3);

dbms\_output.put\_line('Thêm bảng điểm 3 thành công!');

-- Insert score 4

INSERT INTO bangdiem (diemlan1, diemlan2, nhanxet, ma\_SV, ma\_DT)

VALUES (diemlan1\_4, diemlan2\_4, nhanxet4, ma\_SV4, ma\_DT4);

dbms\_output.put\_line('Thêm bảng điểm 4 thành công!');

-- Insert score 5

INSERT INTO bangdiem (diemlan1, diemlan2, nhanxet, ma\_SV, ma\_DT)

VALUES (diemlan1\_5, diemlan2\_5, nhanxet5, ma\_SV5, ma\_DT5);

dbms\_output.put\_line('Thêm bảng điểm 5 thành công!');

END;

* **Bảng Môn Học**

DECLARE

ma\_MH1 VARCHAR(10) := 'MH001';

tenMH1 VARCHAR2(50) := 'Môn học 1';

sotinchi1 INT := 3;

giatien1 INT := 500000;

tongtien1 INT := sotinchi1 \* giatien1;

ma\_MH2 VARCHAR(10) := 'MH002';

tenMH2 VARCHAR2(50) := 'Môn học 2';

sotinchi2 INT := 4;

giatien2 INT := 600000;

tongtien2 INT := sotinchi2 \* giatien2;

ma\_MH3 VARCHAR(10) := 'MH003';

tenMH3 VARCHAR2(50) := 'Môn học 3';

sotinchi3 INT := 3;

giatien3 INT := 550000;

tongtien3 INT := sotinchi3 \* giatien3;

ma\_MH4 VARCHAR(10) := 'MH004';

tenMH4 VARCHAR2(50) := 'Môn học 4';

sotinchi4 INT := 2;

giatien4 INT := 450000;

tongtien4 INT := sotinchi4 \* giatien4;

ma\_MH5 VARCHAR(10) := 'MH005';

tenMH5 VARCHAR2(50) := 'Môn học 5';

sotinchi5 INT := 4;

giatien5 INT := 600000;

tongtien5 INT := sotinchi5 \* giatien5;

BEGIN

-- Insert subject 1

INSERT INTO monhoc (ma\_MH, tenMH, sotinchi, giatien, tongtien)

VALUES (ma\_MH1, tenMH1, sotinchi1, giatien1, tongtien1);

dbms\_output.put\_line('Thêm môn học 1 thành công!');

-- Insert subject 2

INSERT INTO monhoc (ma\_MH, tenMH, sotinchi, giatien, tongtien)

VALUES (ma\_MH2, tenMH2, sotinchi2, giatien2, tongtien2);

dbms\_output.put\_line('Thêm môn học 2 thành công!');

-- Insert subject 3

INSERT INTO monhoc (ma\_MH, tenMH, sotinchi, giatien, tongtien)

VALUES (ma\_MH3, tenMH3, sotinchi3, giatien3, tongtien3);

dbms\_output.put\_line('Thêm môn học 3 thành công!');

-- Insert subject 4

INSERT INTO monhoc (ma\_MH, tenMH, sotinchi, giatien, tongtien)

VALUES (ma\_MH4, tenMH4, sotinchi4, giatien4, tongtien4);

dbms\_output.put\_line('Thêm môn học 4 thành công!');

-- Insert subject 5

INSERT INTO monhoc (ma\_MH, tenMH, sotinchi, giatien, tongtien)

VALUES (ma\_MH5, tenMH5, sotinchi5, giatien5, tongtien5);

dbms\_output.put\_line('Thêm môn học 5 thành công!');

END;

* **Bảng điểm Môn Học**

DECLARE

ma\_SV1 VARCHAR(10) := 'SV001';

ma\_MH1 VARCHAR(10) := 'MH001';

diemcc1 FLOAT := 8.0;

diemkt1 FLOAT := 8.5;

diemtk1 FLOAT := (diemcc1 + diemkt1) / 2;

ma\_SV2 VARCHAR(10) := 'SV002';

ma\_MH2 VARCHAR(10) := 'MH002';

diemcc2 FLOAT := 7.5;

diemkt2 FLOAT := 7.0;

diemtk2 FLOAT := (diemcc2 + diemkt2) / 2;

ma\_SV3 VARCHAR(10) := 'SV003';

ma\_MH3 VARCHAR(10) := 'MH003';

diemcc3 FLOAT := 9.0;

diemkt3 FLOAT := 9.5;

diemtk3 FLOAT := (diemcc3 + diemkt3) / 2;

ma\_SV4 VARCHAR(10) := 'SV004';

ma\_MH4 VARCHAR(10) := 'MH004';

diemcc4 FLOAT := 8.5;

diemkt4 FLOAT := 8.0;

diemtk4 FLOAT := (diemcc4 + diemkt4) / 2;

ma\_SV5 VARCHAR(10) := 'SV005';

ma\_MH5 VARCHAR(10) := 'MH005';

diemcc5 FLOAT := 7.0;

diemkt5 FLOAT := 7.5;

diemtk5 FLOAT := (diemcc5 + diemkt5) / 2;

BEGIN

-- Insert score 1

INSERT INTO diemMH (ma\_SV, ma\_MH, diemcc, diemkt, diemtk)

VALUES (ma\_SV1, ma\_MH1, diemcc1, diemkt1, diemtk1);

dbms\_output.put\_line('Thêm điểm môn học 1 thành công!');

-- Insert score 2

INSERT INTO diemMH (ma\_SV, ma\_MH, diemcc, diemkt, diemtk)

VALUES (ma\_SV2, ma\_MH2, diemcc2, diemkt2, diemtk2);

dbms\_output.put\_line('Thêm điểm môn học 2 thành công!');

-- Insert score 3

INSERT INTO diemMH (ma\_SV, ma\_MH, diemcc, diemkt, diemtk)

VALUES (ma\_SV3, ma\_MH3, diemcc3, diemkt3, diemtk3);

dbms\_output.put\_line('Thêm điểm môn học 3 thành công!');

-- Insert score 4

INSERT INTO diemMH (ma\_SV, ma\_MH, diemcc, diemkt, diemtk)

VALUES (ma\_SV4, ma\_MH4, diemcc4, diemkt4, diemtk4);

dbms\_output.put\_line('Thêm điểm môn học 4 thành công!');

-- Insert score 5

INSERT INTO diemMH (ma\_SV, ma\_MH, diemcc, diemkt, diemtk)

VALUES (ma\_SV5, ma\_MH5, diemcc5, diemkt5, diemtk5);

dbms\_output.put\_line('Thêm điểm môn học 5 thành công!');

END;

* **Bảng Giảng Viên**

DECLARE

ma\_GV1 VARCHAR(10) := 'GV001';

hoten1 VARCHAR2(100) := 'Giang Viên 1';

ma\_MH1 VARCHAR(10) := 'MH001';

sdt1 VARCHAR(20) := '0123456789';

gioitinh1 VARCHAR(20) := 'Nam';

ma\_GV2 VARCHAR(10) := 'GV002';

hoten2 VARCHAR2(100) := 'Giang Viên 2';

ma\_MH2 VARCHAR(10) := 'MH002';

sdt2 VARCHAR(20) := '0987654321';

gioitinh2 VARCHAR(20) := 'Nữ';

ma\_GV3 VARCHAR(10) := 'GV003';

hoten3 VARCHAR2(100) := 'Giang Viên 3';

ma\_MH3 VARCHAR(10) := 'MH003';

sdt3 VARCHAR(20) := '0333333333';

gioitinh3 VARCHAR(20) := 'Nam';

ma\_GV4 VARCHAR(10) := 'GV004';

hoten4 VARCHAR2(100) := 'Giang Viên 4';

ma\_MH4 VARCHAR(10) := 'MH004';

sdt4 VARCHAR(20) := '0555555555';

gioitinh4 VARCHAR(20) := 'Nữ';

ma\_GV5 VARCHAR(10) := 'GV005';

hoten5 VARCHAR2(100) := 'Giang Viên 5';

ma\_MH5 VARCHAR(10) := 'MH005';

sdt5 VARCHAR(20) := '0777777777';

gioitinh5 VARCHAR(20) := 'Nam';

BEGIN

-- Insert teacher 1

INSERT INTO giangvien (ma\_GV, hoten, ma\_MH, sdt, gioitinh)

VALUES (ma\_GV1, hoten1, ma\_MH1, sdt1, gioitinh1);

dbms\_output.put\_line('Thêm giảng viên 1 thành công!');

-- Insert teacher 2

INSERT INTO giangvien (ma\_GV, hoten, ma\_MH, sdt, gioitinh)

VALUES (ma\_GV2, hoten2, ma\_MH2, sdt2, gioitinh2);

dbms\_output.put\_line('Thêm giảng viên 2 thành công!');

-- Insert teacher 3

INSERT INTO giangvien (ma\_GV, hoten, ma\_MH, sdt, gioitinh)

VALUES (ma\_GV3, hoten3, ma\_MH3, sdt3, gioitinh3);

dbms\_output.put\_line('Thêm giảng viên 3 thành công!');

-- Insert teacher 4

INSERT INTO giangvien (ma\_GV, hoten, ma\_MH, sdt, gioitinh)

VALUES (ma\_GV4, hoten4, ma\_MH4, sdt4, gioitinh4);

dbms\_output.put\_line('Thêm giảng viên 4 thành công!');

-- Insert teacher 5

INSERT INTO giangvien (ma\_GV, hoten, ma\_MH, sdt, gioitinh)

VALUES (ma\_GV5, hoten5, ma\_MH5, sdt5, gioitinh5);

dbms\_output.put\_line('Thêm giảng viên 5 thành công!');

END;

## 2.4 Code Trên SQL

-- Tạo bảng người dùng

CREATE TABLE nguoidung (

usename VARCHAR(40) NOT NULL,

password VARCHAR(40) NOT NULL,

comfirmpw VARCHAR(40) NOT NULL,

email VARCHAR(50) NOT NULL,

PRIMARY KEY (usename)

);

-- Tạo bảng sinh viên

CREATE TABLE sinhvien (

hoten VARCHAR2(100) NOT NULL,

ma\_SV VARCHAR(10) NOT NULL,

sdt VARCHAR(20) NOT NULL,

diachi VARCHAR2(100) NOT NULL,

ngaysinh DATE NOT NULL,

lop VARCHAR(100) NOT NULL,

gioitinh VARCHAR2(10) NOT NULL,

PRIMARY KEY (ma\_SV)

);

-- Tạo bảng đề tài

CREATE TABLE detai (

ma\_DT VARCHAR(10) NOT NULL,

tenDT VARCHAR2(255) NOT NULL,

gioihan VARCHAR(50) NOT NULL,

PRIMARY KEY (ma\_DT)

);

-- Tạo bảng điểm

CREATE TABLE bangdiem (

diemlan1 FLOAT NOT NULL,

diemlan2 FLOAT NOT NULL,

nhanxet VARCHAR2(255) NOT NULL,

ma\_SV VARCHAR(10) NOT NULL,

ma\_DT VARCHAR(10) NOT NULL,

FOREIGN KEY (ma\_SV) REFERENCES sinhvien (ma\_SV),

FOREIGN KEY (ma\_DT) REFERENCES detai(ma\_DT)

);

-- Tạo bảng môn học

CREATE TABLE monhoc (

ma\_MH VARCHAR(10) NOT NULL,

tenMH VARCHAR2(50) NOT NULL,

sotinchi INT NOT NULL,

giatien INT NOT NULL,

tongtien INT NOT NULL,

PRIMARY KEY (ma\_MH)

);

-- Tạo bảng điểm môn học

CREATE TABLE diemMH (

ma\_SV VARCHAR(10) NOT NULL,

ma\_MH VARCHAR(10) NOT NULL,

diemcc FLOAT NOT NULL,

diemkt FLOAT NOT NULL,

diemtk FLOAT NOT NULL,

PRIMARY KEY (ma\_SV, ma\_MH),

FOREIGN KEY (ma\_SV) REFERENCES sinhvien (ma\_SV),

FOREIGN KEY (ma\_MH) REFERENCES monhoc (ma\_MH)

);

-- Tạo bảng giảng viên

CREATE TABLE giangvien (

ma\_GV VARCHAR(10) NOT NULL,

hoten VARCHAR2(100) NOT NULL,

ma\_MH VARCHAR(10) NOT NULL,

sdt VARCHAR(20) NOT NULL,

gioitinh VARCHAR(20) NOT NULL,

PRIMARY KEY (ma\_GV),

FOREIGN KEY (ma\_MH) REFERENCES monhoc (ma\_MH)

);

# Chương III: Thiết kế các truy vấn

## 3.1 Truy vấn cơ bản

* Lấy danh sách tất cả sinh viên:

SELECT \* FROM sinhvien11

A screenshot of a computer

Description automatically generated

* Lấy điểm của sinh viên có mã sinh viên '2023345':

SELECT \* FROM diemMH WHERE ma\_SV = '2023345';

A screenshot of a computer

Description automatically generated

## 3.2 Truy vấn lồng nhau

* Lấy danh sách các sinh viên và điểm trung bình của họ trong môn học 'MH001':

SELECT s.\*, dmh.diemtk AS diem\_trung\_binh

FROM sinhvien11 s

INNER JOIN diemMH dmh ON s.ma\_SV = dmh.ma\_SV

WHERE dmh.ma\_MH = 'MH001';

A screenshot of a computer

Description automatically generated

* Lấy thông tin của các giảng viên dạy các môn học có số tín chỉ lớn hơn 3:

SELECT gv.\*

FROM giangvien11 gv

INNER JOIN monhoc mh ON gv.ma\_MH = mh.ma\_MH

WHERE mh.sotinchi > 3;

A screenshot of a computer

Description automatically generated

* Lấy tên các sinh viên đã tham gia vào các đề tài có giới hạn thời gian là '100 giờ':

SELECT s.hoten, dt.tenDT

FROM sinhvien11 s

INNER JOIN bangdiem bd ON s.ma\_SV = bd.ma\_SV

INNER JOIN detai dt ON bd.ma\_DT = dt.ma\_DT

WHERE dt.gioihan = '100 giờ';

A screenshot of a computer

Description automatically generated

## 3.3 Truy vấn gộp nhóm

* Đếm số lượng sinh viên nam và nữ trong cơ sở dữ liệu:

SELECT gioitinh, COUNT(\*) AS so\_luong

FROM sinhvien11

GROUP BY gioitinh;

A screenshot of a computer

Description automatically generated

* Tính điểm trung bình của từng môn học:

SELECT ma\_MH, AVG(diemtk) AS diem\_trung\_binh

FROM diemMH

GROUP BY ma\_MH;

A screenshot of a computer

Description automatically generated

* Liệt kê số lượng sinh viên đã đạt điểm từ 8.0 trở lên trong từng môn học:

SELECT ma\_MH, COUNT(\*) AS so\_luong

FROM diemMH

WHERE diemtk >= 8.0

GROUP BY ma\_MH;

A screenshot of a computer

Description automatically generated

## 3.4 Truy vấn nâng cao

* Tính tổng số tiền được thu bởi mỗi giảng viên:

SELECT gv.ma\_GV, gv.hoten, SUM(mh.giatien) AS tong\_tien\_thu\_duoc

FROM giangvien11 gv

INNER JOIN monhoc mh ON gv.ma\_MH = mh.ma\_MH

GROUP BY gv.ma\_GV, gv.hoten;

A screenshot of a computer

Description automatically generated

# Chương IV: Lập trình PL/SQL

**Định nghĩa:**

PL/SQL là một ngôn ngữ lập trình thủ tục được sử dụng để kết hợp các truy vấn SQL với các khối mã thủ tục để tạo các ứng dụng cơ sở dữ liệu mạnh mẽ và linh hoạt. PL/SQL cho phép người dùng viết các chương trình để:

* Tạo, sửa đổi và xóa các đối tượng cơ sở dữ liệu
* Thao tác dữ liệu trong cơ sở dữ liệu
* Truy xuất dữ liệu từ cơ sở dữ liệu
* Quản lý các hoạt động của cơ sở dữ liệu

**Các thành phần của PL/SQL:**

PL/SQL bao gồm hai thành phần chính:

* SQL: PL/SQL sử dụng SQL để truy vấn và thao tác dữ liệu trong cơ sở dữ liệu.
* Các thành phần lập trình thủ tục: PL/SQL bao gồm các thành phần lập trình thủ tục như các biến, điều kiện, vòng lặp, hàm, và thủ tục.

**Ưu điểm của PL/SQL:**

PL/SQL mang lại một số lợi ích cho việc phát triển ứng dụng cơ sở dữ liệu, bao gồm:

* Tính khả chuyển: PL/SQL là một ngôn ngữ lập trình thủ tục tiêu chuẩn, được hỗ trợ bởi nhiều cơ sở dữ liệu khác nhau.
* Tính hiệu quả: PL/SQL được biên dịch thành mã máy, giúp tăng hiệu suất của ứng dụng.
* Tính linh hoạt: PL/SQL cho phép người dùng kết hợp các truy vấn SQL với các khối mã thủ tục để tạo các ứng dụng phức tạp.

**Ứng dụng của PL/SQL:**

PL/SQL được sử dụng trong nhiều ứng dụng cơ sở dữ liệu, bao gồm:

* Ứng dụng web: PL/SQL được sử dụng để truy cập và thao tác dữ liệu trong cơ sở dữ liệu từ các ứng dụng web.
* Ứng dụng di động: PL/SQL được sử dụng để truy cập và thao tác dữ liệu trong cơ sở dữ liệu từ các ứng dụng di động.
* Ứng dụng doanh nghiệp: PL/SQL được sử dụng để phát triển các ứng dụng doanh nghiệp phức tạp.

PL/SQL là một ngôn ngữ lập trình thủ tục mạnh mẽ và linh hoạt được sử dụng để phát triển các ứng dụng cơ sở dữ liệu. PL/SQL cho phép người dùng kết hợp các truy vấn SQL với các khối mã thủ tục để tạo các ứng dụng phức tạp và hiệu quả.

## 4.1 Một số thủ tục

* Sử dụng thủ tục thêm sinh viên vào danh sách

DECLARE

hoten VARCHAR2(100) := 'Nguyễn Văn 3';

ma\_SV VARCHAR2(10) := '8888';

sdt VARCHAR2(20) := '0123456789';

diachi VARCHAR2(100) := 'Hà Nội';

ngaysinh DATE := TO\_DATE('2001-01-01', 'YYYY-MM-DD');

lop VARCHAR2(100) := '6';

gioitinh VARCHAR2(10) := 'Nam';

BEGIN

*-- Thêm dữ liệu vào bảng*

INSERT INTO sinhvien (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh)

VALUES (hoten, ma\_SV, sdt, diachi, ngaysinh, lop, gioitinh);

dbms\_output.put\_line('Thêm sinh viên thành công!');

END;

Kết quả

A screenshot of a computer

Description automatically generated

Trước khi thêm

A screenshot of a computer

Description automatically generated

Sau khi thêm

A screenshot of a computer

Description automatically generated

## 4.2 Câu lệnh IF - ELSE

* Sinh viên có điểm trung bình >7 for

DECLARE

*-- Khai báo biến*

v\_ma\_SV VARCHAR2(10);

v\_diem\_tb NUMBER;

BEGIN

*-- Lặp qua tất cả các hàng trong bảng điểm môn học*

FOR rec IN (SELECT ma\_SV, AVG(diemcc + diemkt + diemtk) AS diem\_tb

FROM diemMH

GROUP BY ma\_SV) LOOP

*-- Lấy điểm trung bình của sinh viên*

v\_diem\_tb := rec.diem\_tb;

*-- Kiểm tra xem điểm trung bình có lớn hơn hoặc bằng 7.0 hay không*

IF v\_diem\_tb >= 7.0 THEN

*-- In thông tin sinh viên*

dbms\_output.put\_line('Mã sinh viên: ' || rec.ma\_SV);

dbms\_output.put\_line('Điểm trung bình: ' || rec.diem\_tb);

END IF;

END LOOP;

END;

Kết quả

A screenshot of a computer

Description automatically generated

* Tìm sinh viên có điểm cao nhất (for)

DECLARE

diemcao FLOAT;

ma\_SV VARCHAR2(40);

BEGIN

*-- Khởi tạo điểm cao nhất*

diemcao := 0;

*-- Lặp qua tất cả các sinh viên*

FOR rec IN (SELECT diemlan1, diemlan2, ma\_SV

FROM bangdiem)

LOOP

*-- Kiểm tra xem điểm hiện tại có lớn hơn điểm cao nhất hay không*

IF rec.diemlan1 + rec.diemlan2 > diemcao THEN

*-- Cập nhật điểm cao nhất*

diemcao := rec.diemlan1 + rec.diemlan2;

ma\_SV := rec.ma\_SV;

END IF;

END LOOP;

*-- In ra sinh viên có điểm cao nhất*

dbms\_output.put\_line('Sinh viên có điểm cao nhất: ' || ma\_SV);

END;

Kết quả

A screenshot of a computer

Description automatically generated

## 4.3 Một số hàm

* Sử dụng hàm để tính điểm trung bình sinh viên

CREATE OR REPLACE FUNCTION tinh\_diem\_tb(

p\_ma\_SV IN diemMH.ma\_SV%TYPE

) RETURN FLOAT IS

v\_diem FLOAT;

BEGIN

SELECT AVG((diemcc + diemkt + diemtk)/3) INTO v\_diem FROM diemMH WHERE ma\_SV = p\_ma\_SV;

RETURN v\_diem;

END;

DECLARE

v\_diemtb FLOAT;

BEGIN

v\_diemtb := tinh\_diem\_tb(p\_ma\_SV => '1111');

DBMS\_OUTPUT.PUT\_LINE('điểm trung bình của sinh viên có mã: ' || '1111' ||' LÀ: '|| v\_diemtb);

END;

DROP FUNCTION tinh\_diem\_tb

Kết quả

A screenshot of a computer

Description automatically generated

# Chương V: Quản trị cơ sở dữ liệu Oracle

## 5.1 Quản trị Instance

Trong Oracle Database, instance là một bộ các tiến trình và vùng nhớ được sử dụng để truy cập và quản lý một cơ sở dữ liệu cụ thể. Instance bao gồm:

* Quá trình Oracle: Quá trình Oracle là các tiến trình thực thi các lệnh SQL và PL/SQL.
* Vùng nhớ: Vùng nhớ chứa dữ liệu và mã của cơ sở dữ liệu.

Instance được khởi động khi bạn tạo cơ sở dữ liệu và bị đóng khi bạn tắt cơ sở dữ liệu.

Các thành phần của instance:

Instance bao gồm các thành phần chính sau:

* Quá trình Oracle: Quá trình Oracle là các tiến trình thực thi các lệnh SQL và PL/SQL. Quá trình Oracle chính là quá trình lắng nghe các yêu cầu truy cập cơ sở dữ liệu từ các ứng dụng.
* Vùng nhớ: Vùng nhớ chứa dữ liệu và mã của cơ sở dữ liệu. Vùng nhớ được chia thành các khu vực sau:
  + Vùng dữ liệu: Vùng chứa dữ liệu của cơ sở dữ liệu, bao gồm các bảng, chỉ mục, và các đối tượng khác.
  + Vùng mã: Vùng chứa mã của cơ sở dữ liệu, bao gồm các thủ tục, hàm, và các đối tượng khác.
  + Vùng bộ nhớ cache: Vùng chứa dữ liệu và mã được sử dụng thường xuyên để tăng hiệu suất truy cập cơ sở dữ liệu.

Tên instance:

Mỗi instance có một tên duy nhất, được tạo bởi người dùng khi tạo cơ sở dữ liệu. Tên instance được sử dụng để kết nối với cơ sở dữ liệu.

Số lượng instance:

Một máy chủ Oracle có thể có nhiều instance, mỗi instance có thể truy cập một cơ sở dữ liệu khác nhau.

Trạng thái instance:

Instance có thể ở một trong các trạng thái sau:

* OPEN: Instance đang mở và sẵn sàng để truy cập.
* MOUNTED: Instance đã được khởi động nhưng chưa sẵn sàng để truy cập.
* SHUTDOWN: Instance đã bị đóng.

Quản lý instance:

Instance được quản lý bằng các lệnh SQL và PL/SQL. Bạn có thể sử dụng các lệnh này để tạo, khởi động, dừng, và xóa instance.

Instance là một thành phần quan trọng của Oracle Database. Instance cho phép người dùng truy cập và quản lý cơ sở dữ liệu.

* Điều chỉnh các tham số trong SPFILE

A black background with blue text

Description automatically generated

Chỉnh sửa dung lượng

A screen shot of a computer

Description automatically generated

Xem lạiA black background with blue text

Description automatically generated

A black text on a white background

Description automatically generated

Khởi động instance

A screenshot of a computer

Description automatically generated

Tạo một SPFILE từ PFILE

A black screen with blue text

Description automatically generated

Tắt instance

A black background with white text

Description automatically generated

## 5.2 Quản trị tablespace

Tablespace là một đơn vị lưu trữ logic trong cơ sở dữ liệu Oracle. Nó là một nơi lưu trữ dữ liệu thực tế của các đối tượng cơ sở dữ liệu, chẳng hạn như bảng, thủ tục, trình kích hoạt và view.

Tablespace được tạo bởi quản trị viên cơ sở dữ liệu và được gán cho các đối tượng cơ sở dữ liệu khi chúng được tạo. Đối tượng cơ sở dữ liệu có thể được lưu trữ trên một hoặc nhiều khoảng chứa bảng.

Có hai loại tablespace chính:

* System tablespace: Khoảng chứa bảng hệ thống chứa các đối tượng cơ sở dữ liệu quan trọng, chẳng hạn như các bảng dữ liệu từ điển hệ thống và các rollback segment.
* User tablespace: Khoảng chứa bảng người dùng chứa các đối tượng cơ sở dữ liệu do người dùng tạo, chẳng hạn như bảng, thủ tục và view.

Các thuộc tính tablespace:

Mỗi khoảng chứa bảng có các thuộc tính sau:

* Tên: Tên của khoảng chứa bảng.
* Loại: Loại khoảng chứa bảng, chẳng hạn như hệ thống hoặc người dùng.
* Kích thước: Dung lượng lưu trữ của khoảng chứa bảng.
* Định dạng: Định dạng dữ liệu được sử dụng để lưu trữ dữ liệu trong khoảng chứa bảng.

Quản lý tablespace:

Quản trị viên cơ sở dữ liệu chịu trách nhiệm quản lý các khoảng chứa bảng. Các tác vụ quản lý tablespace bao gồm:

* Tạo tablespace mới.
* Xóa tablespace.
* Mở rộng tablespace.
* Thu hẹp tablespace.
* Đặt giới hạn dung lượng cho tablespace.
* Theo dõi việc sử dụng tablespace.

Tablespace cung cấp một số lợi ích cho cơ sở dữ liệu, bao gồm:

* Tính khả dụng: Tablespace có thể được đặt trên các ổ đĩa vật lý khác nhau để cải thiện tính khả dụng của cơ sở dữ liệu.
* Tính bảo mật: Tablespace có thể được bảo vệ bằng mật khẩu để bảo vệ dữ liệu trong đó.
* Tính dễ quản lý: Tablespace giúp quản trị viên cơ sở dữ liệu dễ dàng quản lý dữ liệu trong cơ sở dữ liệu.

Tạo tablespace minh2 30M

A close up of a computer screen

Description automatically generated

A computer screen with text

Description automatically generated A screen shot of a computer

Description automatically generated

## 5.3 Quản trị người dùng

Quản trị người dùng trong database là một tập hợp các quy trình và thủ tục được sử dụng để tạo, quản lý và xóa người dùng database. Quản trị người dùng là một phần quan trọng của quản trị database, vì nó giúp đảm bảo rằng chỉ những người có quyền truy cập cần thiết mới có thể truy cập vào database.

Các tác vụ quản trị người dùng bao gồm:

* Tạo người dùng: Tạo một người dùng mới trong database.
* Cấp quyền cho người dùng: Cấp quyền cho người dùng để truy cập và thao tác các đối tượng trong database.
* Xóa người dùng: Xóa một người dùng khỏi database.
* Quản lý mật khẩu người dùng: Thay đổi, đặt lại hoặc khóa mật khẩu người dùng.
* Theo dõi hoạt động của người dùng: Theo dõi hoạt động của người dùng trong database để phát hiện bất kỳ hoạt động đáng ngờ nào.

Quản trị người dùng giúp đảm bảo:

* Bảo mật: Chỉ những người có quyền truy cập cần thiết mới có thể truy cập vào database.
* Tính toàn vẹn dữ liệu: Chỉ những người có quyền cần thiết mới có thể sửa đổi dữ liệu trong database.
* Hiệu suất: Quản trị người dùng tốt có thể giúp cải thiện hiệu suất của database.

Các nguyên tắc quản trị người dùng

Có một số nguyên tắc cơ bản cần tuân theo khi quản trị người dùng trong database, bao gồm:

* Sử dụng các tên người dùng và mật khẩu mạnh: Sử dụng các tên người dùng và mật khẩu mạnh sẽ giúp bảo vệ database khỏi truy cập trái phép.
* Cấp quyền cho người dùng chỉ những quyền mà họ cần: Chỉ cấp quyền cho người dùng những quyền mà họ cần để thực hiện công việc của họ.
* Theo dõi hoạt động của người dùng: Theo dõi hoạt động của người dùng để phát hiện bất kỳ hoạt động đáng ngờ nào.
* Tạo các chính sách và quy trình: Tạo các chính sách và quy trình để quản lý người dùng trong database.

A black background with blue text

Description automatically generatedA black background with blue text

Description automatically generated

Tạo user

A black background with blue text

Description automatically generated

Kết nối database, cấp quyền cho user

A black background with blue text

Description automatically generated

Unlock user

A black background with blue text

Description automatically generated

Đổi mật khẩu user

A black background with blue text

Description automatically generated

## 5.4 Sao lưu, phục hồi

* Sao lưu

Sao lưu được thực hiện để bảo vệ dữ liệu khỏi bị mất do các sự cố như lỗi phần cứng, lỗi phần mềm hoặc thiên tai. Bản sao lưu có thể được lưu trữ trên đĩa cứng, băng từ hoặc các thiết bị lưu trữ khác.

Có hai loại sao lưu chính:

* Sao lưu toàn bộ: Sao chép tất cả dữ liệu trong cơ sở dữ liệu.
* Sao lưu theo phân vùng: Sao chép một phần dữ liệu trong cơ sở dữ liệu, chẳng hạn như một bảng hoặc một nhóm bảng.

Oracle Database cung cấp một số công cụ để thực hiện sao lưu, bao gồm:

* RMAN: Trình quản lý sao lưu và phục hồi tích hợp sẵn của Oracle Database.
* Oracle Data Pump: Công cụ sao chép dữ liệu cho phép bạn sao chép dữ liệu giữa các cơ sở dữ liệu Oracle.
* Oracle GoldenGate: Bộ công cụ sao chép dữ liệu thời gian thực cho phép bạn sao chép dữ liệu giữa các cơ sở dữ liệu Oracle trong thời gian thực.
* Phục hồi

Phục hồi được thực hiện để khôi phục dữ liệu từ bản sao lưu trong trường hợp dữ liệu gốc bị mất hoặc bị hỏng.

Có hai loại phục hồi chính:

* Phục hồi toàn bộ: Khôi phục tất cả dữ liệu trong cơ sở dữ liệu từ một bản sao lưu toàn bộ.
* Phục hồi theo phân vùng: Khôi phục một phần dữ liệu trong cơ sở dữ liệu từ một bản sao lưu theo phân vùng.

Oracle Database cung cấp một số công cụ để thực hiện phục hồi, bao gồm:

* RMAN: Trình quản lý sao lưu và phục hồi tích hợp sẵn của Oracle Database.
* Oracle Data Pump: Công cụ sao chép dữ liệu cho phép bạn sao chép dữ liệu giữa các cơ sở dữ liệu Oracle.
* Oracle GoldenGate: Bộ công cụ sao chép dữ liệu thời gian thực cho phép bạn sao chép dữ liệu giữa các cơ sở dữ liệu Oracle trong thời gian thực.

Sao lưu và phục hồi là hai công cụ quan trọng để đảm bảo tính khả dụng và tính toàn vẹn của dữ liệu trong cơ sở dữ liệu Oracle. Bằng cách sao lưu dữ liệu thường xuyên, bạn có thể giảm thiểu rủi ro mất dữ liệu do các sự cố. Bằng cách phục hồi dữ liệu từ bản sao lưu, bạn có thể khôi phục dữ liệu trong trường hợp dữ liệu gốc bị mất hoặc bị hỏng.

Có một số nguyên tắc cơ bản cần tuân theo khi thực hiện sao lưu và phục hồi, bao gồm:

* Sao lưu thường xuyên: Sao lưu dữ liệu thường xuyên để giảm thiểu rủi ro mất dữ liệu.
* Lưu trữ bản sao lưu ở nơi an toàn: Lưu trữ bản sao lưu ở nơi an toàn để bảo vệ khỏi bị mất hoặc bị hỏng.
* Kiểm tra bản sao lưu: Kiểm tra bản sao lưu thường xuyên để đảm bảo dữ liệu vẫn được sao lưu chính xác.
* Phục hồi thử nghiệm: Thực hiện phục hồi thử nghiệm thường xuyên để đảm bảo bạn có thể khôi phục dữ liệu thành công khi cần thiết.

Kết nối với oracle database và sử dụng lệnh backup database:

A screen shot of a computer

Description automatically generated

........A screen shot of a computer

Description automatically generated

Kiểm tra ở vị trí lưu file backup:

A screenshot of a computer

Description automatically generated

Xác nhận bản sao lưu:

A black background with blue text

Description automatically generated

.......



Thực hiện một bản xem trước của khôi phục lại cơ sở dữ liệu:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Kết Luận

Dự án quản lý sinh viên đã được thiết kế và xây dựng để giúp trường học hoặc trường đại học quản lý thông tin về sinh viên, giảng viên, lớp học, môn học và điểm số một cách hiệu quả. Dự án này có mục tiêu giúp cải thiện quá trình quản lý học tập và theo dõi tiến trình học tập của sinh viên.

Dự án bao gồm các bảng cơ sở dữ liệu chính như SinhVien, GiangVien, LopHoc, MonHoc và Diem, được thiết kế với các ràng buộc để đảm bảo tính toàn vẹn dữ liệu và quan hệ giữa chúng. Mỗi bảng có trường khóa chính đảm bảo tính duy nhất của dữ liệu và các trường khóa ngoại kết nối các bảng với nhau.

Dự án này giúp cải thiện quá trình quản lý sinh viên và hỗ trợ các chức năng quan trọng như nhập điểm, theo dõi tiến trình học tập của sinh viên và quản lý thông tin giảng viên và lớp học. Nó cung cấp một nền tảng đáng tin cậy để lưu trữ và truy xuất thông tin liên quan đến giáo dục và học tập.

Trong tương lai, dự án này có thể mở rộng và phát triển thêm các tính năng như lập kế hoạch học tập, báo cáo thống kê, và tương tác trực tuyến giữa sinh viên và giảng viên. Điều này sẽ cung cấp lợi ích lớn cho cả trường học và cộng đồng giáo dục trong việc quản lý và cải thiện quá trình học tập của sinh viên.

# Tài liệu tham khảo

<https://doan.edu.vn/do-an/do-an-he-quan-tri-co-so-du-lieu-oracle-trong-quan-ly-ban-hang-21331/>

<https://topdev.vn/blog/pl-sql-la-gi-hieu-sau-ve-pl-sql/>