

**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
Lời cảm ơn.....	5
CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	6
1.1 Giới Thiệu Đề Tài.....	6
1.2 Các Thực Thể.....	6
1.3 Mối Quan Hệ Thực Thể.....	11
CHƯƠNG II: XÂY DỰNG CƠ SỞ DỮ LIỆU	14
2.1 Thiết Kế Các Bảng.....	14
2.2 Xây Dựng Các Ràng Buộc	17
2.3 Nhập Dữ Liệu	18
2.4 Code Trên SQL.....	32
Chương III: Thiết kế các truy vấn.....	35
3.1 Truy vấn cơ bản	35
3.2 Truy vấn lồng nhau	35
3.3 Truy vấn gộp nhóm.....	36
3.4 Truy vấn nâng cao	38
Chương 4: Lập trình PL/SQL.....	39
4.1 Một số thủ tục	40
4.2 Câu lệnh IF - ELSE.....	42
4.3 Một số hàm	44
Chương 5: Quản trị cơ sở dữ liệu Oracle.....	45
5.1 Quản trị Instance.....	45
5.2 Quản trị tablespace	47
5.3 Quản trị người dùng.....	49
5.4 Sao lưu, phục hồi	51
Kết Luận.....	56
Tài liệu tham khảo.....	57

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 Hằng đã 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:

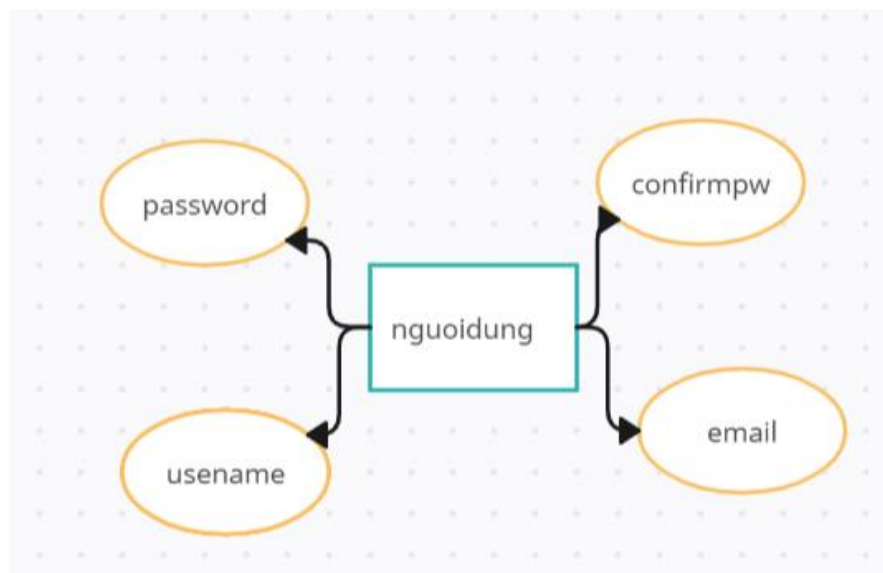
- Mục tiêu của dự án là xây dựng một hệ thống quản lý sinh viên hiệu quả, giúp quản lý thông tin cá nhân của sinh viên, thông tin học tập, và mối quan hệ với giảng viên.

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.

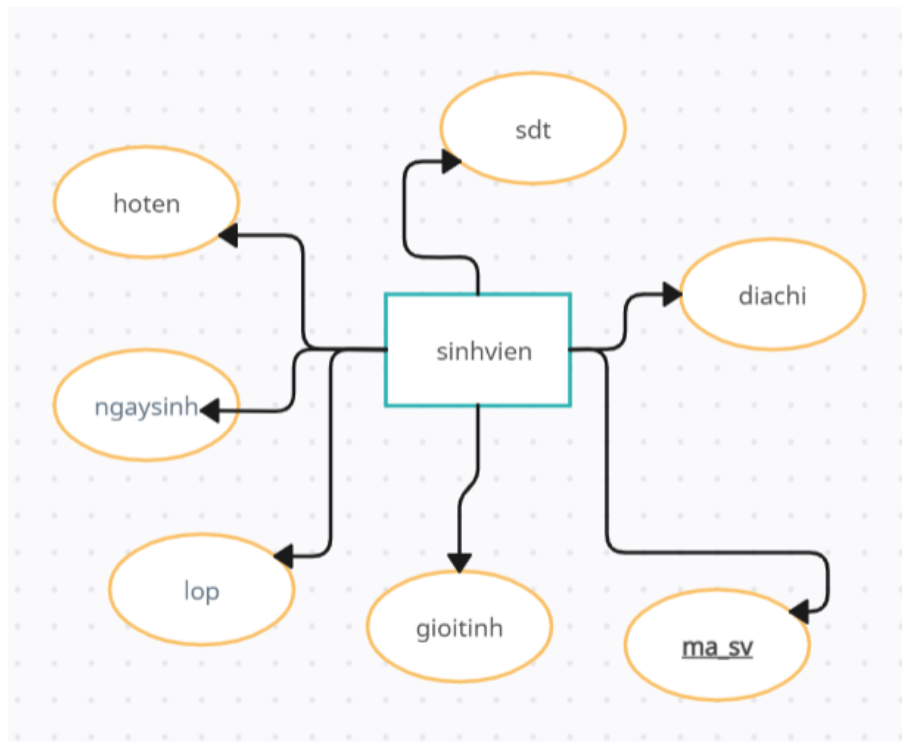
1.2 Các Thực Thể

- 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.



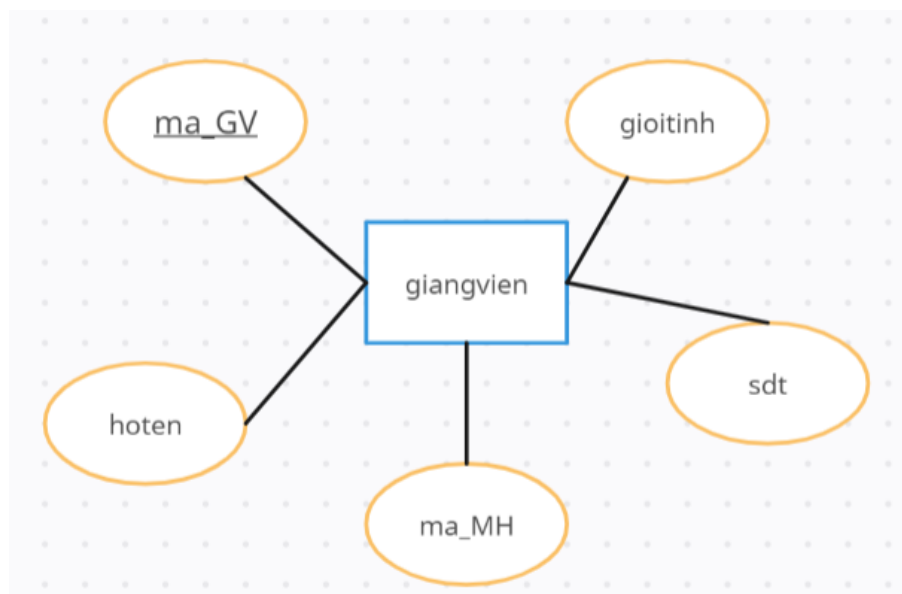
Hình 2.1: Thực thể người dùng

- 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



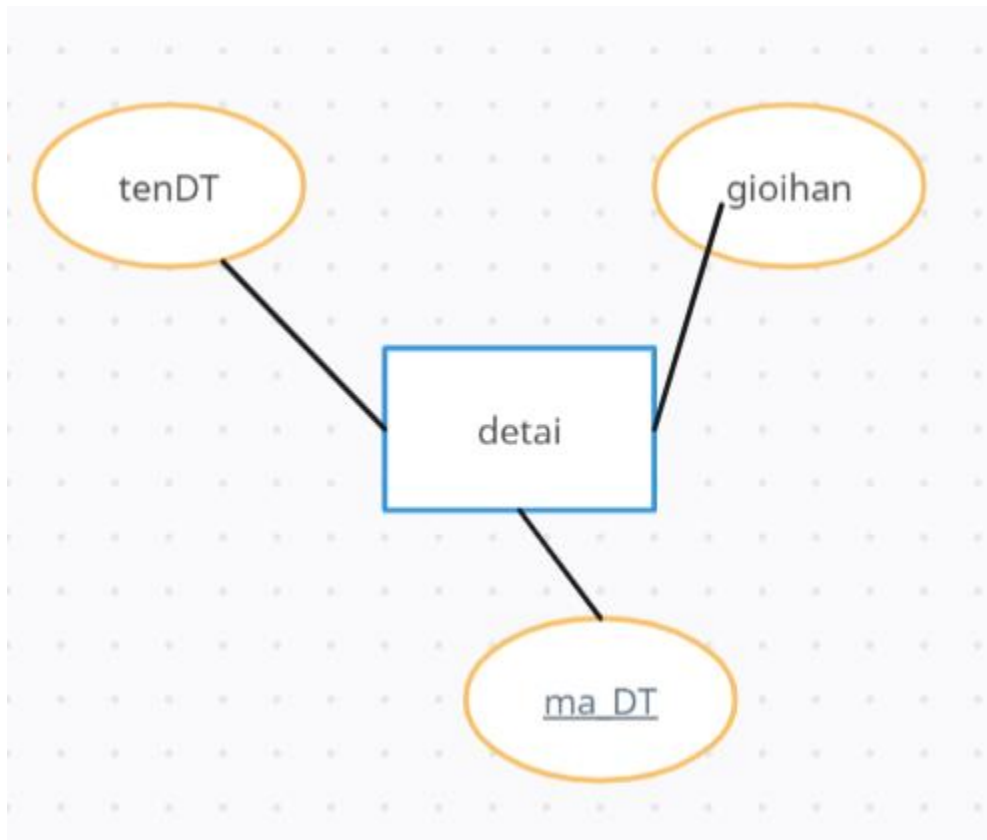
Hình 2.2: Thực thể sinh viên

- 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.



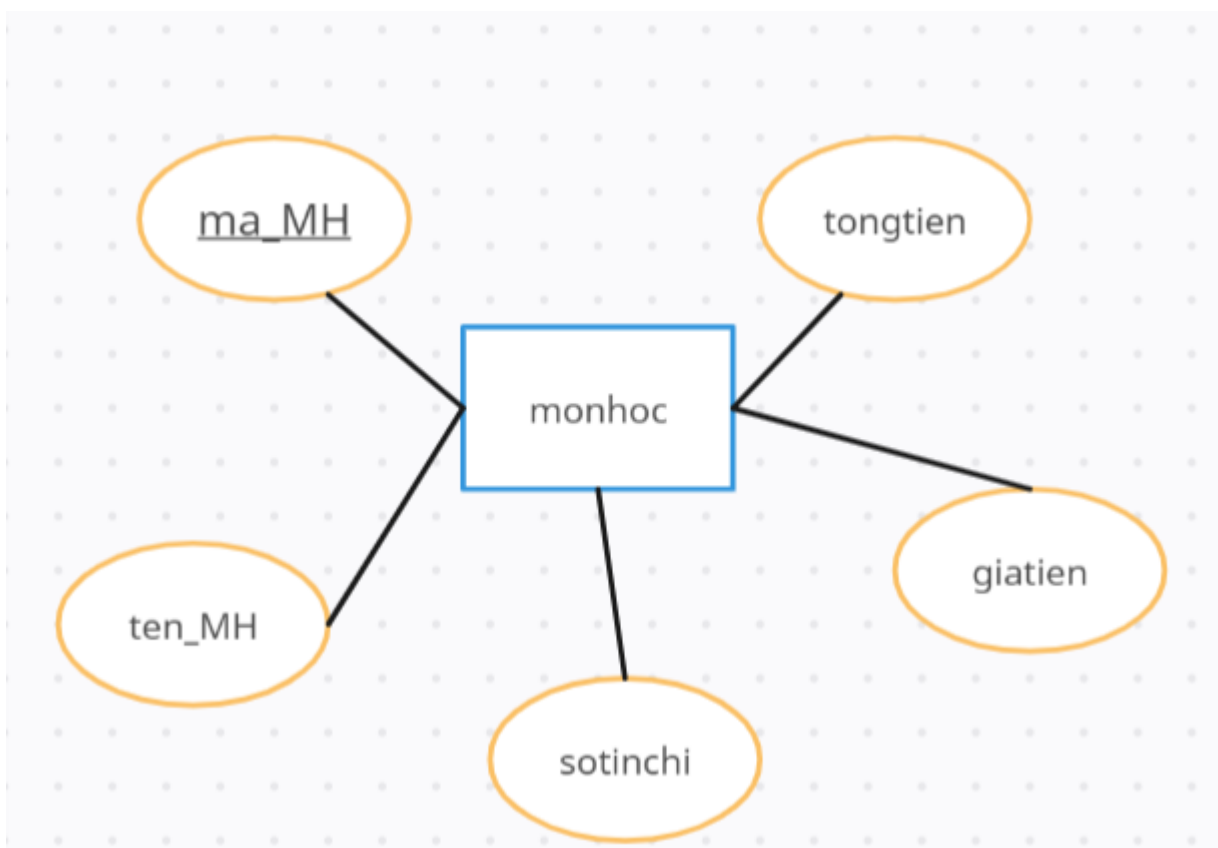
Hình 2.3: Thực thể giảng viên

- 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 .



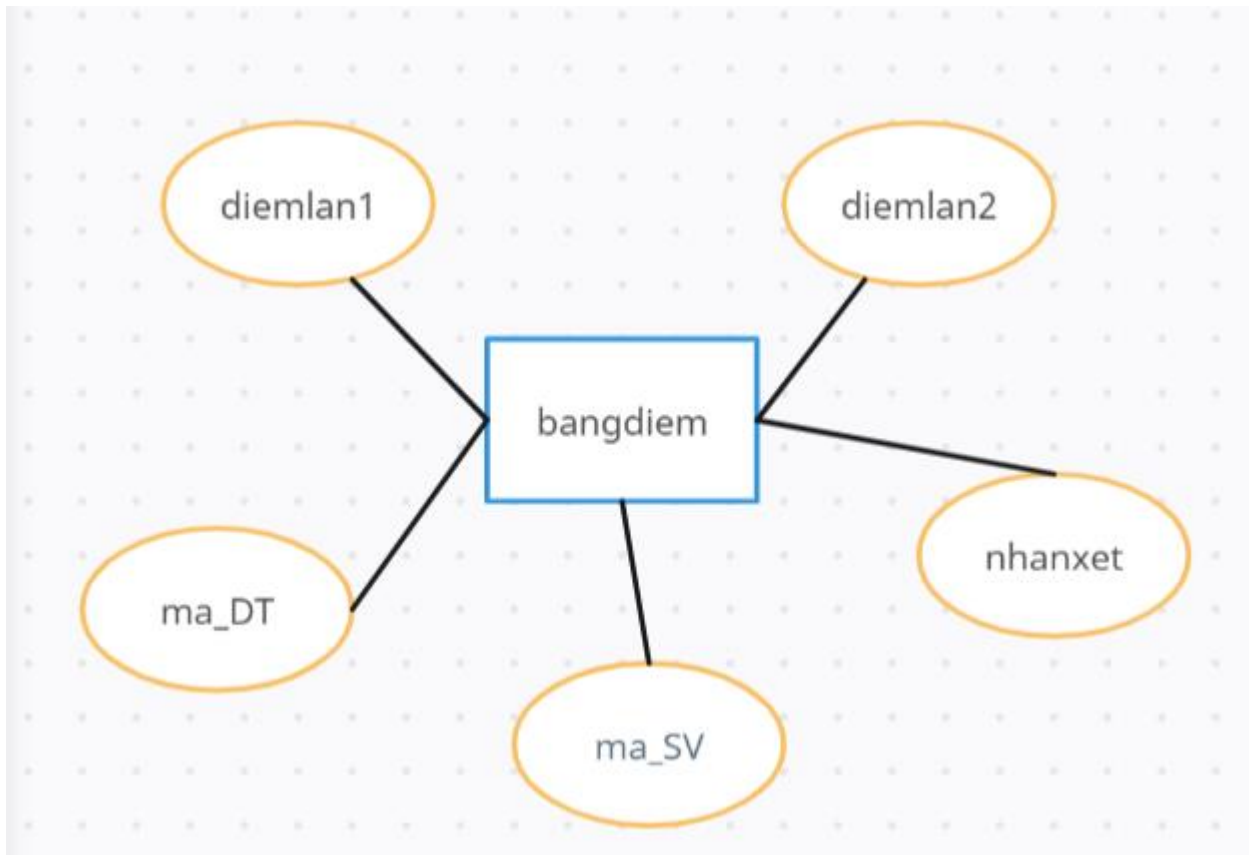
Hình 2.4: Thực thể đề tài

- 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 .



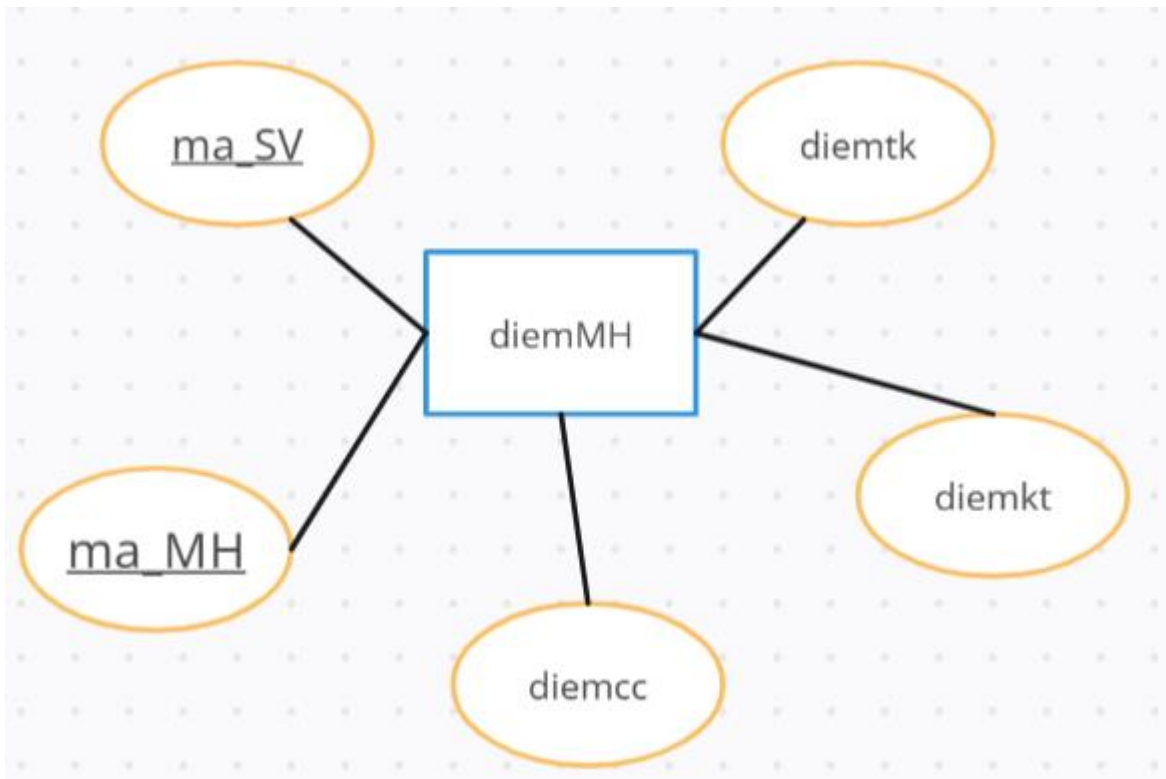
Hình 2.5: Thực thể môn học

- 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.



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



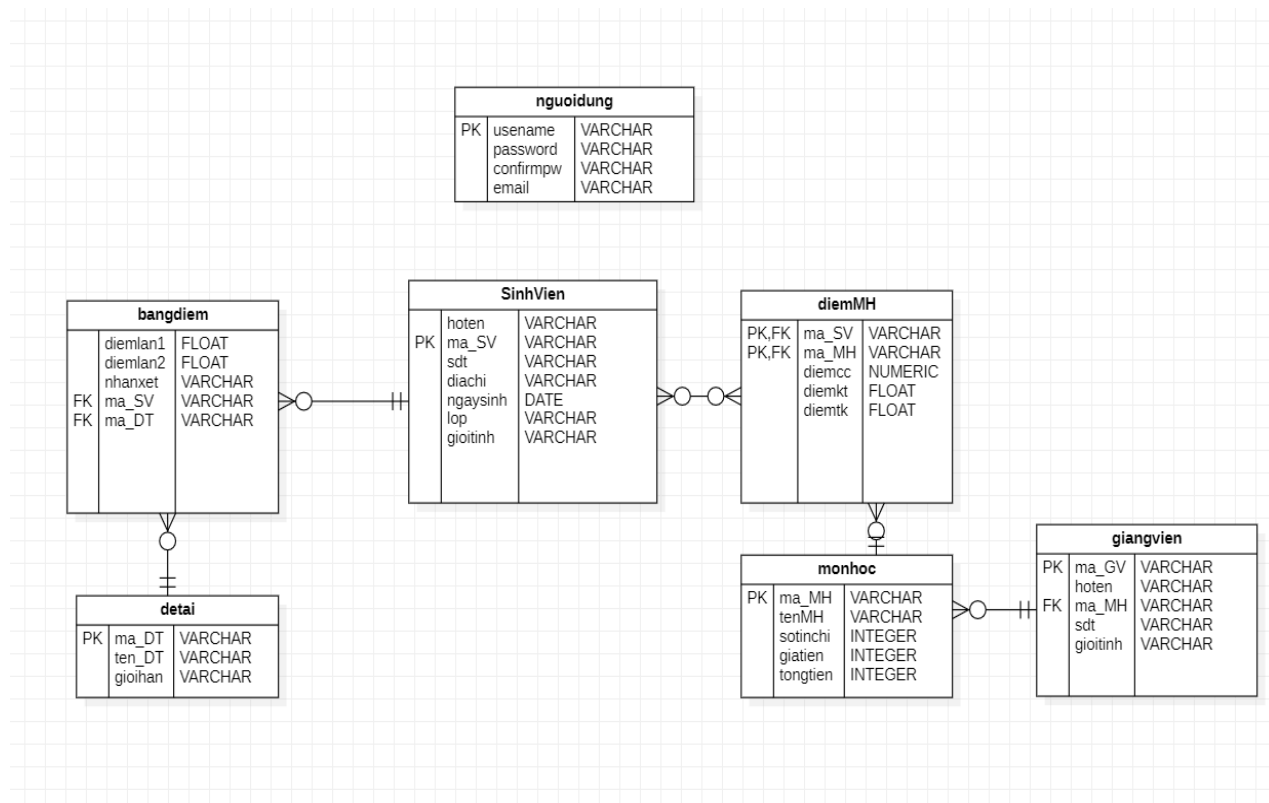
Hình 2.7: Thực thể bảng điểm môn học

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").
- Mỗi 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").
- Mỗi 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 :



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: username, 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 :

nguoidung		
PK	username	VARCHAR
	password	VARCHAR
	confirmpw	VARCHAR
	email	VARCHAR

Hình 1.1: Bảng người dùng

SinhVien		
PK	hoten ma_SV sdt diachi ngaysinh lop gioitinh	VARCHAR VARCHAR VARCHAR VARCHAR DATE VARCHAR VARCHAR

Hình 1.2: Bảng sinh viên

monhoc		
PK	ma_MH tenMH sotinchi giatien tongtien	VARCHAR VARCHAR INTEGER INTEGER INTEGER

Hình 1.3: Bảng môn học

detai		
PK	ma_DT ten_DT gioihan	VARCHAR VARCHAR VARCHAR

Hình 1.4: Bảng đề tài

giangvien		
PK	ma_GV	VARCHAR
	hoten	VARCHAR
FK	ma_MH	VARCHAR
	sdt	VARCHAR
	gioitinh	VARCHAR

Hình 1.5: Bảng giảng viên

bangdiem		
	diemlan1	FLOAT
	diemlan2	FLOAT
	nhanxet	VARCHAR
FK	ma_SV	VARCHAR
FK	ma_DT	VARCHAR

Hình 1.6: Bảng điểm

diemMH		
PK,FK	ma_SV	VARCHAR
PK,FK	ma_MH	VARCHAR
	diemcc	NUMERIC
	diemkt	FLOAT
	diemtk	FLOAT

Hình 1.2: Bảng điểm môn học

2.2 Xây Dựng Các Ràng Buộc

- Bảng nguoidung (người dùng):

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

Khóa chính (PRIMARY KEY) là trường username, 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

```
username1 VARCHAR2(40) := 'nguyen_user1';  
password1 VARCHAR2(40) := 'password123';  
comfirmpw1 VARCHAR2(40) := 'password123';  
email1 VARCHAR2(50) := 'nguyen@example.com';
```

```
username2 VARCHAR2(40) := 'dung_user';  
password2 VARCHAR2(40) := 'password1232';  
comfirmpw2 VARCHAR2(40) := 'password1232';  
email2 VARCHAR2(50) := 'dung@example.com';
```

```
username3 VARCHAR2(40) := 'user3';  
password3 VARCHAR2(40) := 'pass3';  
comfirmpw3 VARCHAR2(40) := 'pass3';  
email3 VARCHAR2(50) := 'user3@example.com';
```

```
username4 VARCHAR2(40) := 'user4';  
password4 VARCHAR2(40) := 'pass4';  
comfirmpw4 VARCHAR2(40) := 'pass4';  
email4 VARCHAR2(50) := 'user4@example.com';
```

```
username5 VARCHAR2(40) := 'user5';  
password5 VARCHAR2(40) := 'pass5';  
comfirmpw5 VARCHAR2(40) := 'pass5';  
email5 VARCHAR2(50) := 'user5@example.com';
```

BEGIN

-- Insert user 1

```
INSERT INTO nguoidung (username, password, comfirmpw, email)
```

```
VALUES (username1, password1, comfirmpw1, email1);
```

```
dbms_output.put_line('Thêm người dùng 1 thành công!');
```

-- Insert user 2

```
INSERT INTO nguoidung (username, password, comfirmpw, email)
```

```
VALUES (username2, password2, comfirmpw2, email2);
```

```
dbms_output.put_line('Thêm người dùng 2 thành công!');
```

-- Insert user 3

```
INSERT INTO nguoidung (username, password, comfirmpw, email)
```

```
VALUES (username3, password3, comfirmpw3, email3);
```

```
dbms_output.put_line('Thêm người dùng 3 thành công!');
```

-- Insert user 4

```
INSERT INTO nguoidung (username, password, comfirmpw, email)
```

```
VALUES (username4, password4, confirmpw4, email4);
dbms_output.put_line('Thêm người dùng 4 thành công!');
-- Insert user 5
INSERT INTO nguoidung (username, password, confirmpw, email)
VALUES (username5, password5, confirmpw5, 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 (  
    username VARCHAR(40) NOT NULL,  
    password VARCHAR(40) NOT NULL,  
    comfirmpw VARCHAR(40) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    PRIMARY KEY (username)  
);
```

-- 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 detail(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

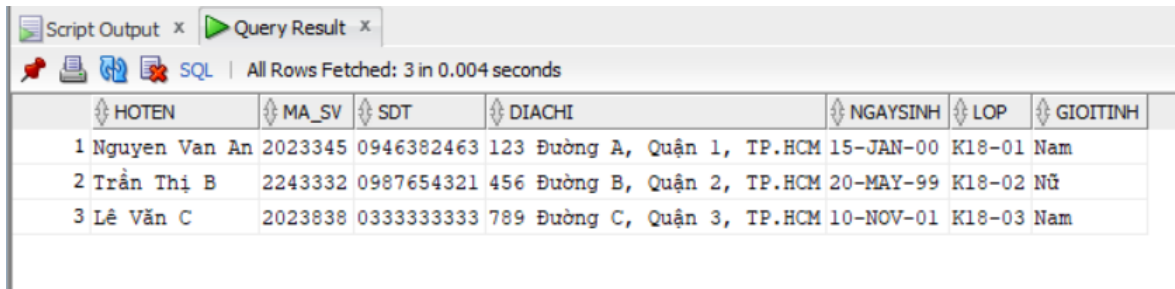
```
CREATE TABLE giangvien (  
    ma_GV VARCHAR(10) NOT NULL,  
    hoten VARCHAR2(100) NOT NULL,  
    ma_MH VARCHAR(10) NOT NULL,  
    sdt VARCHAR(20) NOT NULL,  
    gioitinh VARCHAR(20) NOT NULL,  
    PRIMARY KEY (ma_GV),  
    FOREIGN KEY (ma_MH) REFERENCES monhoc (ma_MH)  
);
```

Chương III: Thiết kế các truy vấn

3.1 Truy vấn cơ bản

- Lấy danh sách tất cả sinh viên:

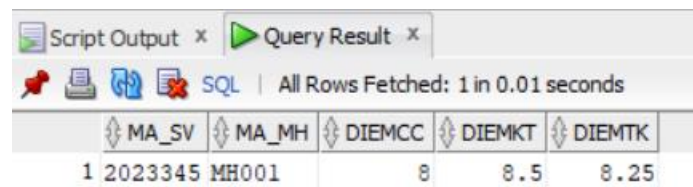
```
SELECT * FROM sinhvien11
```



	HOTEN	MA_SV	SDT	DIACHI	NGAYSINH	LOP	GIOITINH
1	Nguyen Van An	2023345	0946382463	123 Đường A, Quận 1, TP.HCM	15-JAN-00	K18-01	Nam
2	Trần Thị B	2243332	0987654321	456 Đường B, Quận 2, TP.HCM	20-MAY-99	K18-02	Nữ
3	Lê Văn C	2023838	0333333333	789 Đường C, Quận 3, TP.HCM	10-NOV-01	K18-03	Nam

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

```
SELECT * FROM diemMH WHERE ma_SV = '2023345';
```



	MA_SV	MA_MH	DIEMCC	DIEMKT	DIEMTK
1	2023345	MH001	8	8.5	8.25

3.2 Truy vấn lồng nhau

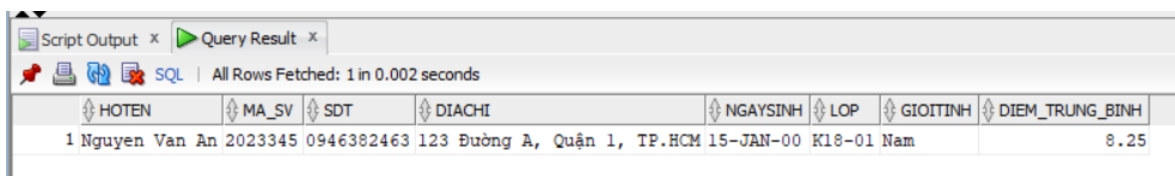
- Lấy danh sách các sinh viên và điểm trung bình của họ trong môn học 'MH001':

```
SELECT s.*, dmh.diemtk AS diem_trung_binh
```

```
FROM sinhvien11 s
```

```
INNER JOIN diemMH dmh ON s.ma_SV = dmh.ma_SV
```

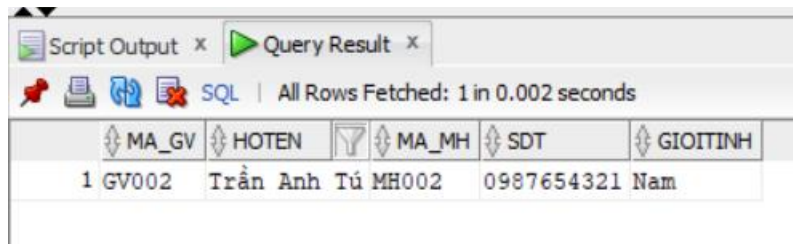
```
WHERE dmh.ma_MH = 'MH001';
```



	HOTEN	MA_SV	SDT	DIACHI	NGAYSINH	LOP	GIOITINH	DIEM_TRUNG_BINH
1	Nguyen Van An	2023345	0946382463	123 Đường A, Quận 1, TP.HCM	15-JAN-00	K18-01	Nam	8.25

- 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;
```



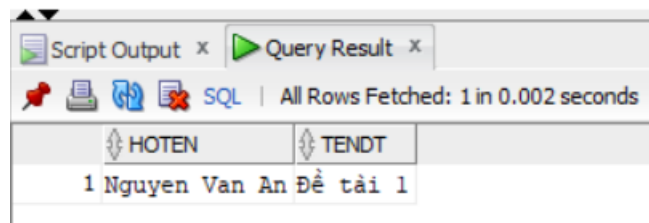
Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.002 seconds

	MA_GV	HOTEN	MA_MH	SDT	GIOITINH
1	GV002	Trần Anh Tú	MH002	0987654321	Nam

- 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';
```



Script Output x Query Result x

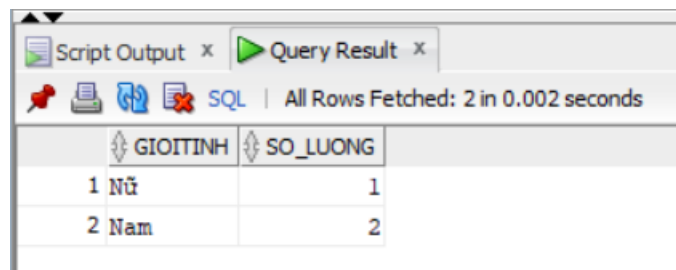
SQL | All Rows Fetched: 1 in 0.002 seconds

	HOTEN	TENDT
1	Nguyen Van An	Đề tài 1

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;
```



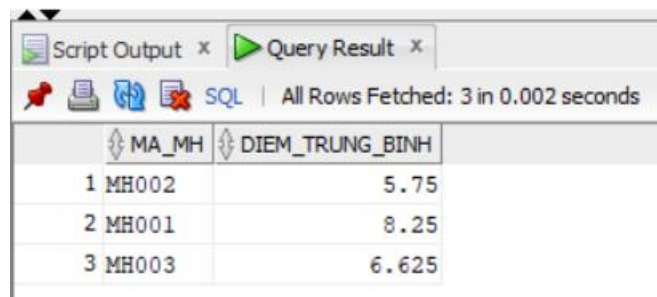
Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.002 seconds

	GIOITINH	SO_LUONG
1	Nữ	1
2	Nam	2

- Tính điểm trung bình của từng môn học:

```
SELECT ma_MH, AVG(diemtk) AS diem_trung_binh  
FROM diemMH  
GROUP BY ma_MH;
```



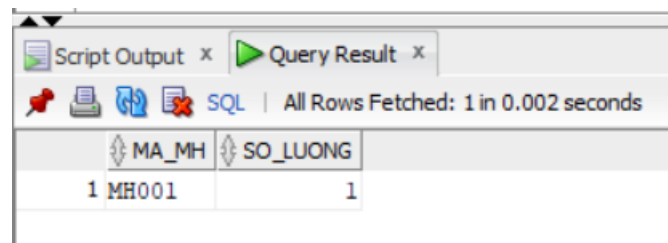
Script Output x Query Result x

SQL | All Rows Fetched: 3 in 0.002 seconds

	MA_MH	DIEM_TRUNG_BINH
1	MH002	5.75
2	MH001	8.25
3	MH003	6.625

- 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;
```



Script Output x Query Result x

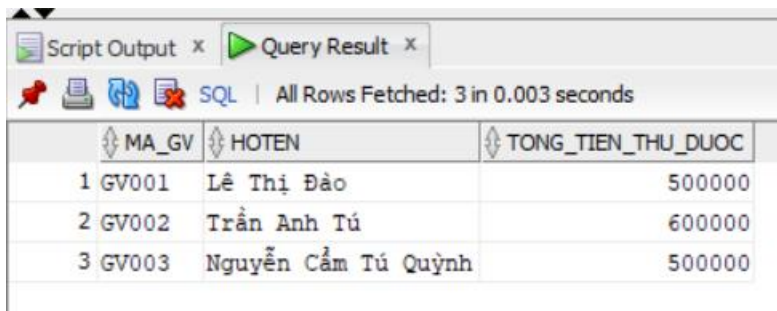
SQL | All Rows Fetched: 1 in 0.002 seconds

	MA_MH	SO_LUONG
1	MH001	1

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;
```



MA_GV	HOTEN	TONG_TIEN_THU_DUOC
1 GV001	Lê Thị Đào	500000
2 GV002	Trần Anh Tú	600000
3 GV003	Nguyễn Cẩm Tú Quỳnh	500000

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.

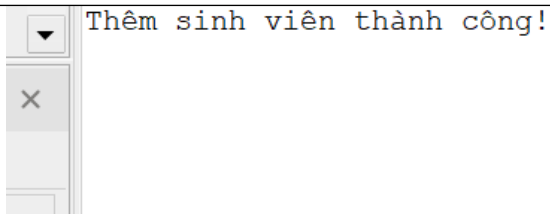
- Ứng dụng di động: PL/SQL được sử dụng để truy cập và thao tác dữ liệu trong cơ sở dữ liệu từ các ứng dụng di động.
- Ứng dụng doanh nghiệp: PL/SQL được sử dụng để phát triển các ứng dụng doanh nghiệp phức tạp.

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;
```



Trước khi thêm

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	HOTEN	MA_SV	SDT	DIACHI	NGAYSINH	LOP	GIOITINH					
1	pham van minh	1111	05245043	ha noi	16-OCT-03	4	nam					
2	nguyen minh van	2222	230843024	thai binh	15-JUL-03	3	nu					
3	minh deptra	3333	44243432	nam dinh	17-JUN-02	4	nam					
4	nguyen van a	4444	29384723	bac ninh	14-MAY-02	2	nu					
5	nguyen thi b	5555	3928742384	ca mau	19-DEC-03	3	nsm					

Sau khi thêm

Columns	Data	Model	Constraints	Grants	Statistics	Triggers	Flashback	Dependencies	Details	Partitions	Indexes	SQL
	HOTEN	MA_SV	SDT	DIACHI	NGAYSINH	LOP	GIOITINH					
1	pham van minh	1111	05245043	ha noi	16-OCT-03	4	nam					
2	nguyen minh van	2222	230843024	thai binh	15-JUL-03	3	nu					
3	minh deptra	3333	44243432	nam dinh	17-JUN-02	4	nam					
4	nguyen van a	4444	29384723	bac ninh	14-MAY-02	2	nu					
5	nguyen thi b	5555	3928742384	ca mau	19-DEC-03	3	nsm					
6	Nguyễn Văn A	SV1	0123456789	Hà Nội	01-JAN-01	K18	Nam					
7	Nguyễn Văn 3	8888	0123456789	Hà Nội	01-JAN-01	6	Nam					
8	Nguyễn Văn 4	9999	0123456789	Hà Nội	01-JAN-01	6	Nam					

4.2 Câu lệnh IF - ELSE

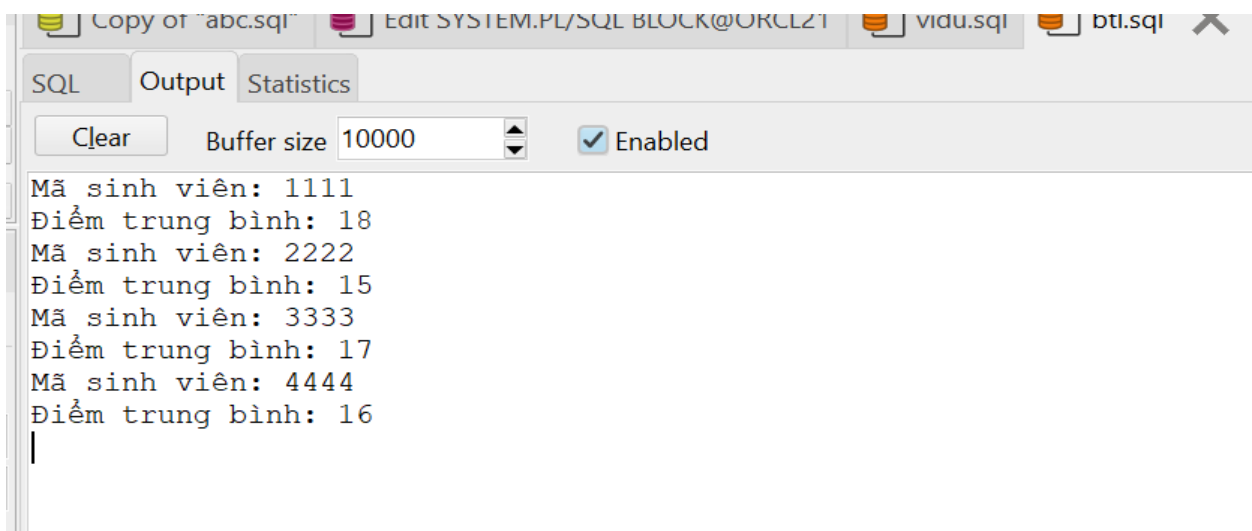
❖ Sinh viên có điểm trung bình >7 for

```

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



```

Copy of abc.sql  Edit SYSTEM:PL/SQL BLOCK@ORCL21  vidu.sql  bti.sql
SQL  Output  Statistics
Clear  Buffer size 10000  Enabled
Mã sinh viên: 1111
Điểm trung bình: 18
Mã sinh viên: 2222
Điểm trung bình: 15
Mã sinh viên: 3333
Điểm trung bình: 17
Mã sinh viên: 4444
Điểm trung bình: 16
    
```

❖ 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;

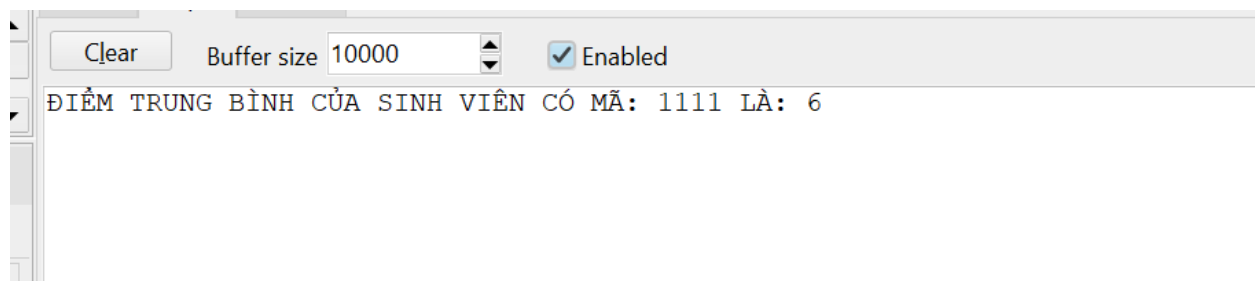
    -- In ra sinh viên có điểm cao nhất
    dbms_output.put_line('Sinh viên có điểm cao nhất: ' ||
ma_SV);
END;
```

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



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

```
SQL> show parameters shared_pool_size;
```

NAME	TYPE	VALUE
shared_pool_size	big integer	0

Chỉnh sửa dung lượng

```
SQL> ALTER SYSTEM SET shared_pool_size=64M SCOPE=BOTH
2 /

System altered.
```

Xem lại

```
SQL> show parameters shared_pool_size;
```

NAME	TYPE	VALUE
shared_pool_size	big integer	64M

```
SQL>
```

```
*.shared_pool_size=67108864
*.undo_tablespace='UNDOTBS1'
```

Khởi động instance

```
SQL> startup
ORACLE instance started.

Total System Global Area 1073741488 bytes
Fixed Size                  9862832 bytes
Variable Size              339738624 bytes
Database Buffers          717225984 bytes
Redo Buffers               6914048 bytes
Database mounted.
Database opened.
SQL>
```

Tạo một SPFILE từ PFILE

```
SQL> create spfile='D:\u01\app\oracle\database\minhDB.ora' from pfile;
File created.

SQL>
```

Tắt instance

```
SQL> shutdown
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> █
```

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

```
CREATE TEMPORARY TABLESPACE minh2
TEMPFILE 'D:\u01\app\oracle\oradata\ORCL21\minh2.dbf'
SIZE 30M;
```

Script Output x | Task completed in 0.22 seconds

TABLESPACE MINH2 created.

CONTROL01.CTL	18,288 KB	10/16/2023 8:02 PM
CONTROL02.CTL	18,288 KB	10/16/2023 8:02 PM
MINH2.DBF	30,728 KB	10/16/2023 8:26 PM
REDO01.LOG	204,801 KB	10/16/2023 5:37 PM
REDO02.LOG	204,801 KB	10/16/2023 4:47 PM
REDO03.LOG	204,801 KB	10/16/2023 4:47 PM
SYSAUX01.DBF	506,328 KB	10/16/2023 8:07 PM

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.

```
SQL> ALTER SESSION SET CONTAINER = ORCLPDB1;  
Session altered.
```

```
SQL> ALTER DATABASE OPEN;  
Database altered.
```

Tạo user

```
SQL> conn / as sysdba  
Connected.  
SQL> CREATE USER c##minh2003 IDENTIFIED BY 1 CONTAINER=ALL;  
User created.
```

Kết nối database, cấp quyền cho user

```
SQL> GRANT CREATE SESSION TO c##minh2003 CONTAINER=ALL;  
Grant succeeded.
```

Unlock user

```
SQL> alter user c##minh2003 account unlock;  
User altered.
```

Đổi mật khẩu user

```
SQL> alter user c##minh2003 identified by 2;  
User altered.
```

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.

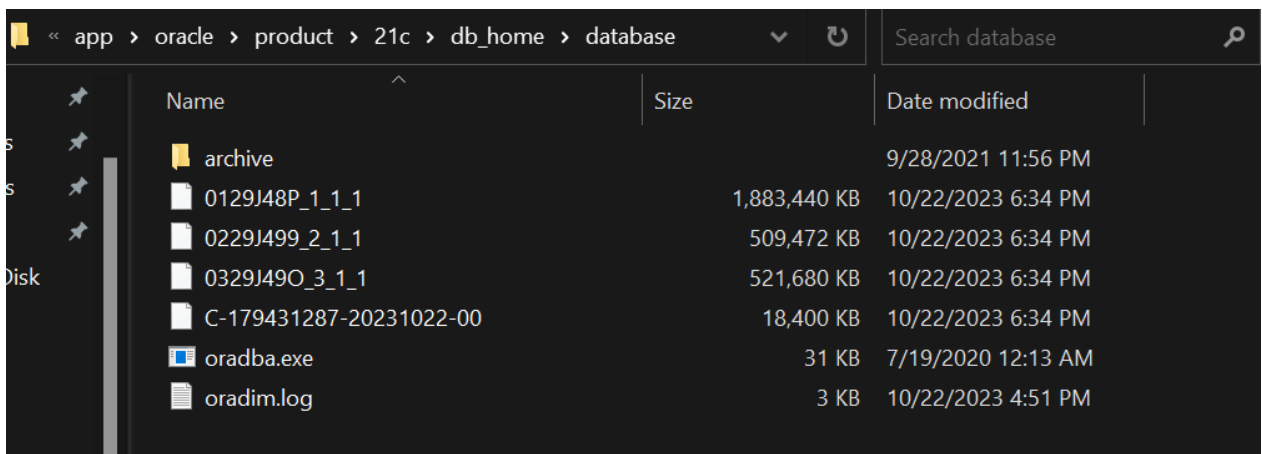
Kết nối với oracle database và sử dụng lệnh backup database:

```
RMAN> backup database;

Starting backup at 22-OCT-23
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
.....
Finished backup at 22-OCT-23

Starting Control File and SPFILE Autobackup at 22-OCT-23
piece handle=D:\U01\APP\ORACLE\PRODUCT\21C\DB_HOME\DATABASE\C-179431287-20231022-00 comment=NONE
Finished Control File and SPFILE Autobackup at 22-OCT-23
```

Kiểm tra ở vị trí lưu file backup:



Name	Size	Date modified
archive		9/28/2021 11:56 PM
0129J48P_1_1_1	1,883,440 KB	10/22/2023 6:34 PM
0229J499_2_1_1	509,472 KB	10/22/2023 6:34 PM
0329J49O_3_1_1	521,680 KB	10/22/2023 6:34 PM
C-179431287-20231022-00	18,400 KB	10/22/2023 6:34 PM
oradba.exe	31 KB	7/19/2020 12:13 AM
oradim.log	3 KB	10/22/2023 4:51 PM

Xác nhận bản sao lưu:

```
RMAN> BACKUP VALIDATE database;

Starting backup at 22-OCT-23
using channel ORA_DISK_1
.....

Finished backup at 22-OCT-23
```

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

```

C:\Administrator: C:\Windows\system32\cmd.exe - rman target/
RMAN> RESTORE DATABASE PREVIEW;

Starting restore at 22-OCT-23
using channel ORA_DISK_1

List of Backup Sets
=====
BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
1 Full 1.80G DISK 00:00:15 22-OCT-23
BP Key: 1 Status: AVAILABLE Compressed: NO Tag: TAG20231022T183401
Piece Name: D:\U01\APP\ORACLE\PRODUCT\21C\DB_HOME\DATABASE\0129J48P_1_1_1
List of Datafiles in backup set 1
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
1 Full 15324682 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\SYSTEM01.DBF
2 Full 15324682 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\SYS_AUX01.DBF
4 Full 15324682 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\UNDOTBS01.DBF
7 Full 15324682 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\USERS01.DBF

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
3 Full 509.45M DISK 00:00:14 22-OCT-23
BP Key: 3 Status: AVAILABLE Compressed: NO Tag: TAG20231022T183401
Piece Name: D:\U01\APP\ORACLE\PRODUCT\21C\DB_HOME\DATABASE\0329J490_3_1_1
List of Datafiles in backup set 3
Container ID: 2, PDB Name: PDB$SEED
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
5 Full 2795509 13-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\PDBSEED\SYSTEM01.DBF
6 Full 2795509 13-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\PDBSEED\SYS_AUX01.DBF
8 Full 2795509 13-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\PDBSEED\UNDOTBS01.DBF

BS Key Type LV Size Device Type Elapsed Time Completion Time
-----
2 Full 497.52M DISK 00:00:14 22-OCT-23
BP Key: 2 Status: AVAILABLE Compressed: NO Tag: TAG20231022T183401
Piece Name: D:\U01\APP\ORACLE\PRODUCT\21C\DB_HOME\DATABASE\0229J499_2_1_1
List of Datafiles in backup set 2
Container ID: 3, PDB Name: ORCLPDB1
File LV Type Ckp SCN Ckp Time Abs Fuz SCN Sparse Name
-----
9 Full 15322937 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\ORCLPDB1\SYSTEM01.DBF
10 Full 15322937 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\ORCLPDB1\SYS_AUX01.DBF
11 Full 15322937 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\ORCLPDB1\UNDOTBS01.DBF
12 Full 15322937 22-OCT-23 NO D:\U01\APP\ORACLE\ORADATA\ORCL21\ORCLPDB1\USERS01.DBF

archived logs generated after SCN 15324682 not found in repository
recovery will be done up to SCN 15324682
Media recovery start SCN is 15324682
Recovery must be done beyond SCN 15324682 to clear datafile fuzziness
Finished restore at 22-OCT-23

```

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/>