

ĐẠI HỌC BACH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

----- □ & □ -----



BÀI TẬP LỚN KỲ 2023.2

Thực hành Cơ sở dữ liệu

IT3290 – 147777

HỆ THỐNG QUẢN LÝ DỮ LIỆU THƯ VIỆN

Giảng viên: **Trần Văn Đặng**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Bùi Quốc Bảo	20225601
2	Vũ Trần Tuấn Minh	20225891
3	Đặng Kim Ngân	20225751

Hà Nội, năm 2024

MỤC LỤC

MỤC LỤC.....	2
I, Mô tả chủ đề.....	3
II, Mục đích	3
III, Chức năng, tình huống sử dụng	3
IV, Sơ đồ thực thể liên kết.....	4
V, Sơ đồ quan hệ	5
VI, Chi tiết bảng	5
VII. Ứng dụng của các câu truy vấn SQL để quản lý dữ liệu thư viện :.....	6
VIII. Ứng dụng Function và Trigger :	10
IX. Trang web quản lý thư viện :	25
X. Đánh giá báo cáo	31
XI. Thành viên nhóm	32

I, Mô tả chủ đề

Từ lâu, việc vận hành và quản lý thư viện là một nhiệm vụ rất khó khăn và thách thức đối với các nhân viên thư viện do dữ liệu ngày càng nhiều và phức tạp. Do đó sẽ rất khó khăn nếu như quản lý dữ liệu của thư viện bằng sổ sách hoặc bằng giấy tờ thông thường. Mục tiêu bài tập lớp này đem đến một hệ thống quản lý dữ liệu thư viện thông qua việc ứng dụng hệ quản trị cơ sở dữ liệu PostgreSQL và thiết kế một trang web giúp quản lý và trực quan hóa dữ liệu một cách dễ dàng hơn.

II, Mục đích

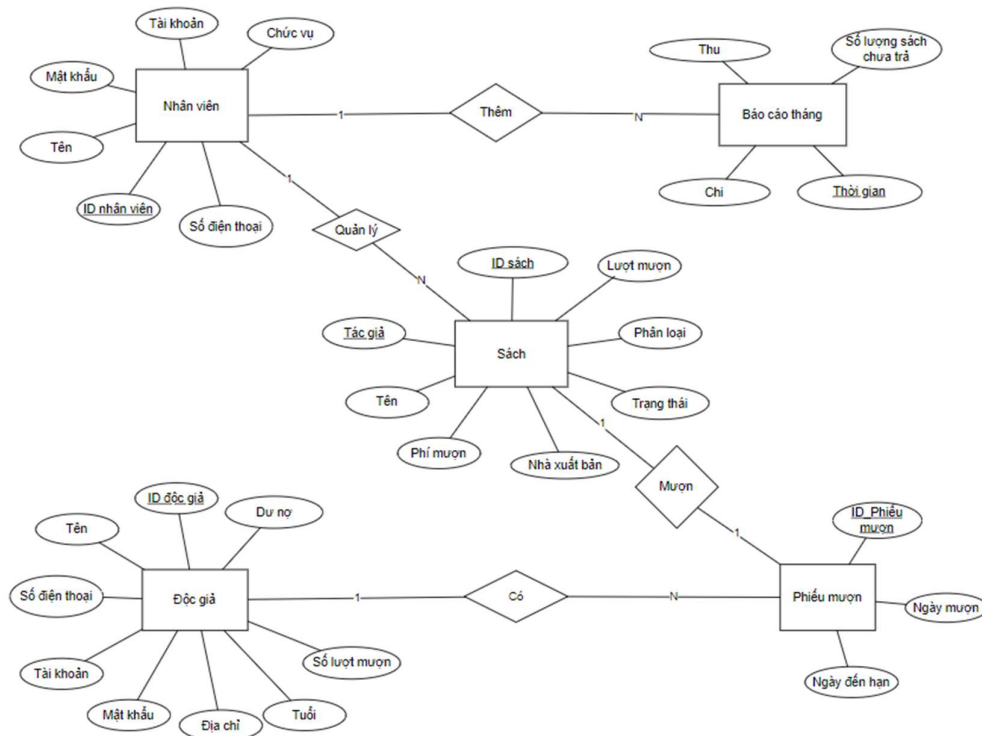
Mục đích của CSDL quản lý thư viện là lưu trữ và quản lý thông tin về các cuốn sách, độc giả, nhân viên và các hoạt động mượn/trả sách. CSDL này giúp cho việc quản lý thư viện trở nên hiệu quả hơn bằng cách cung cấp các chức năng như tìm kiếm, mượn/trả sách, quản lý tài liệu, thống kê và báo cáo. Đồng thời, CSDL cũng giúp cho việc theo dõi và quản lý tình trạng của các cuốn sách trong thư viện, từ đó tối ưu hóa việc quản lý tài nguyên và cung cấp dịch vụ tốt nhất cho độc giả.

III, Chức năng, tính hướng sử dụng

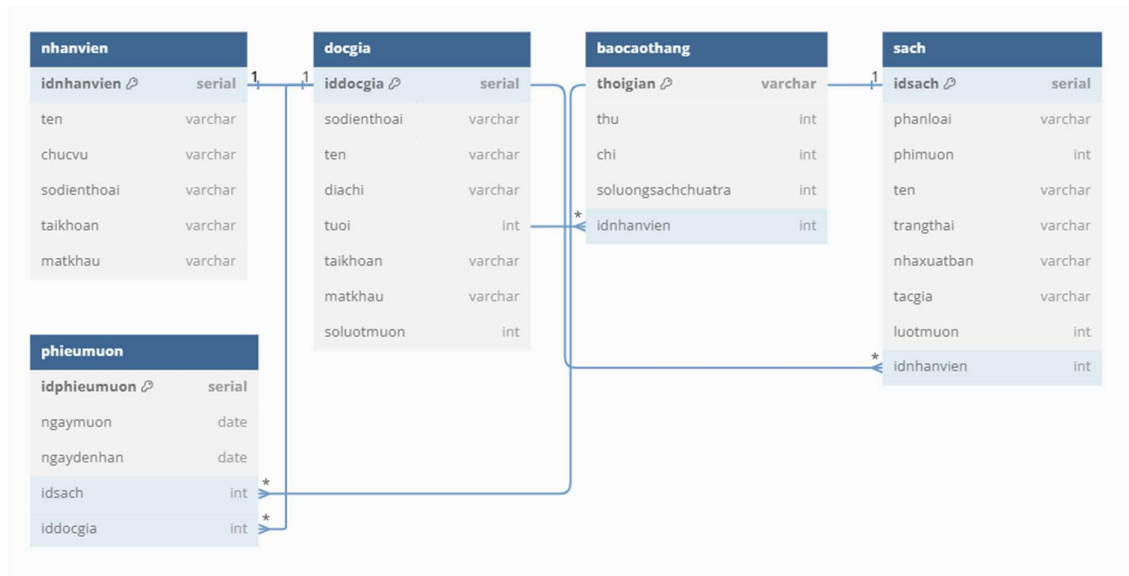
- CSDL quản lý thư viện cung cấp nhiều chức năng quan trọng để hỗ trợ việc quản lý và vận hành thư viện. Dưới đây là mô tả các chức năng chính của CSDL quản lý thư viện:
 - **Quản lý sách:** Cho phép thêm mới, cập nhật, xóa sách trong thư viện. Điều này bao gồm thông tin về sách như tựa đề, tác giả, nhà xuất bản, năm xuất bản, số lượng, vị trí lưu trữ.
 - **Quản lý độc giả:** Cho phép quản lý thông tin cá nhân của độc giả và các hoạt động của độc giả tại thư viện
 - **Quản lý mượn/trả sách:** Ghi nhận thông tin mượn sách của độc giả, bao gồm sách mượn, ngày mượn, ngày hẹn trả, ngày trả, số lần gia hạn (nếu có) và tính toán các khoản phạt (nếu có).
 - **Tìm kiếm sách:** Cho phép độc giả tìm kiếm sách trong thư viện theo nhiều tiêu chí như tựa đề, tác giả, chủ đề,....
 - **Thống kê và báo cáo:** Cung cấp chức năng thống kê số lượng sách, số lượng độc giả, số lần mượn/trả sách, thông tin về sách được mượn nhiều nhất, thông tin về độc giả mượn sách nhiều nhất, v.v. Báo cáo này giúp thư viện đánh giá hiệu suất và nhu cầu của độc giả.

- **Quản lý nhân viên:** Cho phép quản lý thông tin về nhân viên: thông tin cá nhân, chức vụ, tài khoản làm việc.
 - **Quản lý phiếu thu chi:** Ghi nhận thông tin về các khoản thu (ví dụ: phí mượn sách) và chi (ví dụ: chi phí mua sách mới).
 - **Bảo trì dữ liệu:** Cung cấp các chức năng để bảo trì và sao lưu dữ liệu, đảm bảo an toàn và tính nhất quán của dữ liệu.
- Các chức năng này giúp cho việc quản lý thư viện trở nên hiệu quả hơn, giúp thư viện cung cấp dịch vụ tốt nhất cho độc giả và duy trì hoạt động của mình một cách chuyên nghiệp.

IV, Sơ đồ thực thể liên kết



V, Sơ đồ quan hệ



VI, Chi tiết bảng

1. nhanvien

- **idnhanvien (PK)**: serial
- ten: string
- chucvu: string
- sodienthoai: string
- taikhoan: string
- matkhau: string

2. docgia

- **iddocgia (PK)**: serial
- sodienthoai: string
- ten: string
- diachi: string
- tuoi: string
- taikhoan: string
- matkhau: string
- soluotmuon: int

3. baocaothang

- **thoigian (PK)**: string
- thu: int
- chi: int
- soluongsachchuatra: int
- **idnhanvien (FK)**: int

4. sach

- **idsach (PK):** serial
- **phanloai:** string
- **phimuon:** int
- **ten:** string
- **trangthai:** string
- **nhaxuatban:** string
- **tacgia:** string
- **luotmuon:** int
- **idnhanvien (FK):** int

5. phieumuon

- **idphieumuon (PK):** serial
- **ngaymuon:** date
- **ngaydenhan:** date
- **idsach (FK):** int
- **iddocgia (FK):** int

VII. Ứng dụng của các câu truy vấn SQL để quản lý dữ liệu thư viện :

- CÁC CÂU TRUY VẤN TÌM KIẾM :

1. Tìm kiếm thông tin nhân viên theo tên:

```
SELECT *  
FROM nhanvien  
WHERE ten LIKE '%Nguyen Van A%';
```

2. Tìm kiếm thông tin độc giả theo tên:

```
SELECT *  
FROM docgia  
WHERE ten LIKE '%Le Van C%';
```

3. Tìm kiếm sách theo tên:

```
SELECT *  
FROM sach  
WHERE ten LIKE '%Sach Khoa Hoc%';
```

4. Tìm kiếm phiếu mượn theo ngày mượn:

```
SELECT *  
FROM phieumuon  
WHERE ngaymuon = '2024-01-01';
```

5. Tìm kiếm báo cáo tháng theo thời gian:

```
SELECT *  
FROM baocaothang  
WHERE thoigian = '2020-12';
```

6. Tìm 10 cuốn sách được mượn nhiều nhất:

```
SELECT ten, MAX(luotmuon) as max_luotmuon  
FROM sach  
GROUP BY ten  
ORDER BY max_luotmuon DESC  
LIMIT 10;
```

7. Tìm tất cả nhân viên có chức vụ là 'Thủ thư' và quản lý 10 cuốn sách có lượt mượn cao nhất:

```
SELECT nhanvien.ten, sach.ten  
FROM nhanvien  
JOIN sach ON nhanvien.idnhanvien = sach.idnhanvien  
WHERE nhanvien.chucvu = 'Thu Thu'  
ORDER BY sach.luotmuon DESC  
LIMIT 10;
```

8. Tìm tất cả sách đang được mượn bởi độc giả với tên cụ thể:

```
SELECT sach.*  
FROM sach  
JOIN phieumuon ON sach.idsach = phieumuon.idsach  
JOIN docgia ON phieumuon.iddocgia = docgia.iddocgia  
WHERE docgia.ten = 'Pham Thi D' AND sach.trangthai = 'Da muon';
```

9. Tìm tất cả phiếu mượn đã quá hạn trả:

```
SELECT *  
FROM phieumuon  
WHERE ngaydenhan < CURRENT_DATE;
```

- CÁC CÂU TRUY VẤN XEM:

1. Xem chi tiết tất cả phiếu mượn và sách đi kèm:

```
SELECT phieumuon.*, sach.ten  
FROM phieumuon  
JOIN sach ON phieumuon.idsach = sach.idsach;
```

2. Xem tất cả phiếu mượn và phí mượn của nó:

```
SELECT
    phieumuon.idphieumuon,
    phieumuon.ngaymuon,
    phieumuon.ngaydenhan,
    phieumuon.idsach,
    sach.phimuon AS phi_muon
FROM phieumuon
JOIN sach ON phieumuon.idsach = sach.idsach;
```

3. Xem danh sách nhân viên và số lượng sách họ quản lý:

```
SELECT nhanvien.idnhanvien, nhanvien.ten, COUNT(sach.idsach) as
    total_books
FROM nhanvien
JOIN sach ON nhanvien.idnhanvien = sach.idnhanvien
GROUP BY nhanvien.ten, nhanvien.idnhanvien
ORDER BY nhanvien.idnhanvien ASC;
```

4. Xem danh sách độc giả và tổng số phiếu mượn của họ:

```
SELECT docgia.ten, COUNT(phieumuon.idphieumuon) as
    total_loans
FROM docgia
JOIN phieumuon ON docgia.iddocgia = phieumuon.iddocgia
GROUP BY docgia.ten;
```

5. Xem chi tiết 10 báo cáo tháng có doanh thu cao nhất:

```
SELECT *
FROM baocaothang
ORDER BY thu DESC
LIMIT 10;
```

6. Xem tất cả sách và tên nhân viên quản lý chúng:

```
SELECT sach.ten, nhanvien.ten
FROM sach
JOIN nhanvien ON sach.idnhanvien = nhanvien.idnhanvien;
```

7. Xem tất cả báo cáo tháng và tên nhân viên tạo báo cáo:

```
SELECT baocaothang.*, nhanvien.ten
FROM baocaothang
JOIN nhanvien ON baocaothang.idnhanvien = nhanvien.idnhanvien;
```


8. Xem tổng số sách theo trạng thái:

```
SELECT trangthai, COUNT(*)  
FROM sach  
GROUP BY trangthai;
```

9. Xem tổng số sách theo từng thể loại:

```
SELECT phanloai, COUNT(*)  
FROM sach  
GROUP BY phanloai;
```

10. Xem tổng số sách theo từng nhà xuất bản:

```
SELECT nhaxuatban, COUNT(*)  
FROM sach  
GROUP BY nhaxuatban;
```

11. Xem tổng số sách theo từng tác giả:

```
SELECT tacgia, COUNT(*)  
FROM sach  
GROUP BY tacgia;
```

12. Xem số lượng độc giả theo địa chỉ:

```
SELECT diachi, COUNT(*) AS so_luong_doc_gia  
FROM docgia  
GROUP BY diachi;
```

13. Xem số lượng độc giả theo độ tuổi:

```
SELECT  
CASE  
WHEN tuoi BETWEEN 1 AND 10 THEN '1-10'  
WHEN tuoi BETWEEN 11 AND 20 THEN '11-20'  
WHEN tuoi BETWEEN 21 AND 30 THEN '21-30'  
WHEN tuoi BETWEEN 31 AND 40 THEN '31-40'  
WHEN tuoi BETWEEN 41 AND 50 THEN '41-50'  
WHEN tuoi BETWEEN 51 AND 60 THEN '51-60'  
ELSE '60+'  
END AS do_tuai,  
COUNT(*) AS so_luong_doc_gia  
FROM docgia  
GROUP BY
```

```
CASE
WHEN tuoi BETWEEN 1 AND 10 THEN '1-10'
WHEN tuoi BETWEEN 11 AND 20 THEN '11-20'
WHEN tuoi BETWEEN 21 AND 30 THEN '21-30'
WHEN tuoi BETWEEN 31 AND 40 THEN '31-40'
WHEN tuoi BETWEEN 41 AND 50 THEN '41-50'
WHEN tuoi BETWEEN 51 AND 60 THEN '51-60'
ELSE '60+'
END
ORDER BY do_tuoi ASC
```

- CÁC CÂU TRUY VẤN TÍNH TOÁN

1. Tính tổng số lượt mượn của tất cả sách:

```
SELECT SUM(luotmuon) AS tong_so_luot_muon
FROM sach;
```

2. Tính tổng số lượt mượn sách theo từng thể loại sách:

```
SELECT phanloai,
SUM(luotmuon) AS tong_so_luot_muon
FROM sach
GROUP BY phanloai
```

3. Tính lãi trung bình của báo cáo tháng:

```
SELECT AVG(thu - chi) AS doanh_thu_trung_binh
FROM baocaothang
```

4. Tính số lượt mượn trung bình của độc giả:

```
SELECT AVG(soluotmuon) AS so_luot_muon_trung_binh
FROM docgia
```

5. Tính trung bình tuổi của độc giả theo từng thành phố:

```
SELECT diachi, AVG(tuoi) AS average_age
FROM docgia
GROUP BY diachi;
```

VIII. Ứng dụng Function và Trigger :

- CÁC FUNCTION HỮU ÍCH :

1. **Hàm tính tổng số tiền phạt của mỗi phiếu mượn(tiền phạt=0.05 * số ngày quá hạn * phí mượn) và thêm cột tiền phạt vào bảng phiếu mượn:**

```
ALTER TABLE phieumuon  
ADD COLUMN tienphat INT;
```

```
CREATE OR REPLACE FUNCTION  
tinh_tien_phat(ngaydenhan_param DATE, idsach_param INT)  
RETURNS NUMERIC AS $$  
DECLARE  
so_ngay_qua_han INT;  
sach_phimuon INT;  
tien_phat NUMERIC;  
BEGIN  
-- Lấy phí mượn từ bảng sách  
SELECT s.phimuon  
INTO sach_phimuon  
FROM sach s  
WHERE s.idsach = idsach_param;  
-- Tính số ngày quá hạn  
so_ngay_qua_han := CURRENT_DATE - ngaydenhan_param;  
IF so_ngay_qua_han > 0 THEN  
tien_phat := 0.05 * so_ngay_qua_han * sach_phimuon;  
ELSE  
tien_phat := 0;  
END IF;  
  
RETURN tien_phat;  
END;  
$$ LANGUAGE plpgsql;  
  
UPDATE phieumuon  
SET tienphat = tinh_tien_phat(ngaydenhan, idsach);
```

2. **Hàm nhận vào ID nhân viên và trả về danh sách sách NV ấy quản lý:**

```
CREATE OR REPLACE FUNCTION  
lay_danh_sach_sach(p_idnhanvien INT)  
RETURNS TABLE (  
idsach INT,
```

```
phanloai VARCHAR(50),
phimuon INT,
ten VARCHAR(100),
trangthai VARCHAR(20),
nhaxuatban VARCHAR(100),
tacgia VARCHAR(100),
luotmuon INT
) AS $$
BEGIN
RETURN QUERY
SELECT
s.idsach,
s.phanloai,
s.phimuon,
s.ten,
s.trangthai,
s.nhaxuatban,
s.tacgia,
s.luotmuon
FROM sach s
WHERE s.idnhanvien = p_idnhanvien;
END;
$$ LANGUAGE plpgsql;

SELECT * FROM lay_danh_sach_sach(1);
```

3. Hàm nhận vào ID độc giả và trả về danh sách phiếu mượn:

```
CREATE OR REPLACE FUNCTION
lay_danh_sach_phieu_muon(p_iddocgia INT)
RETURNS TABLE (
idphieumuon INT,
ngaymuon DATE,
ngaydenhan DATE,
idsach INT,
id_docgia INT
) AS $$
BEGIN
RETURN QUERY
SELECT
pm.idphieumuon,
```

```
pm.ngaymuon,  
pm.ngaydenhan,  
pm.idsach,  
pm.iddocgia  
FROM phieumuon pm  
WHERE pm.iddocgia = p_iddocgia;  
END;  
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM lay_danh_sach_phieu_muon(1);
```

4. Hàm nhận vào tên thể loại và in ra danh sách sách của thể loại đó:

```
CREATE OR REPLACE FUNCTION  
lay_danh_sach_sach_theo_loai(p_phanloai VARCHAR(50))  
RETURNS TABLE (  
idsach INT,  
phanloai VARCHAR(50),  
phimuon INT,  
ten VARCHAR(100),  
trangthai VARCHAR(20),  
nhaxuatban VARCHAR(100),  
tacgia VARCHAR(100),  
luotmuon INT  
) AS $$  
BEGIN  
RETURN QUERY  
SELECT  
s.idsach,  
s.phanloai,  
s.phimuon,  
s.ten,  
s.trangthai,  
s.nhaxuatban,  
s.tacgia,  
s.luotmuon  
FROM sach s  
WHERE s.phanloai = p_phanloai;  
END;  
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM lay_danh_sach_sach_theo_loai('Van Hoc');
```

5. Hàm nhận vào tên tác giả và trả về danh sách sách của tác giả đó:

```
CREATE OR REPLACE FUNCTION sach_cua_tacgia(ten_tacgia
VARCHAR)
RETURNS TABLE(
idsach INT,
ten VARCHAR,
phanloai VARCHAR,
nhaxuatban VARCHAR)
AS $$
BEGIN
RETURN QUERY
SELECT sach.idsach,
sach.ten,
sach.phanloai,
sach.nhaxuatban
FROM sach
WHERE sach.tacgia = ten_tacgia;
END;
$$ LANGUAGE plpgsql;
```

```
-- Gọi hàm để lấy danh sách sách của một tác giả cụ thể (ví dụ:
'Nguyễn Nhật Anh')
SELECT *
FROM sach_cua_tacgia('Nguyễn Nhật Anh');
```

6. Hàm cập nhật doanh thu cho sách (doanh thu = số lượt mượn * phí mượn) và thêm cột doanh thu vào bảng sách:

```
-- Thêm cột doanhthu vào bảng sach
ALTER TABLE sach
ADD COLUMN doanhthu INT;
```

```
-- Tạo hàm cập nhật doanh thu
CREATE OR REPLACE FUNCTION cap_nhat_doanh_thu()
RETURNS VOID AS $$
BEGIN
UPDATE sach
SET doanhthu = luotmuon * phimuon;
```

```
END;  
$$ LANGUAGE plpgsql;  
  
-- Gọi hàm để cập nhật doanh thu cho tất cả sách  
SELECT cap_nhat_doanh_thu();
```

7. Hàm trả về top 10 cuốn sách có doanh thu cao nhất:

```
-- Tạo hàm trả về top 10 cuốn sách có doanh thu cao nhất  
CREATE OR REPLACE FUNCTION top10_sach_doanhthu_cao()  
RETURNS TABLE(idsach INT,  
ten VARCHAR,  
doanhthu INT)  
AS $$  
BEGIN  
RETURN QUERY  
SELECT s.idsach, s.ten, s.doanhthu  
FROM sach s  
ORDER BY s.doanhthu DESC  
LIMIT 10;  
END;  
$$ LANGUAGE plpgsql;  
  
-- Gọi hàm để lấy top 10 cuốn sách có doanh thu cao nhất  
SELECT * FROM top10_sach_doanhthu_cao();
```

8. Hàm trả về danh sách 10 độc giả có số tiền phạt cao nhất(black-list):

```
CREATE OR REPLACE FUNCTION  
top_10_doc_gia_phat_cao_nhat()  
RETURNS TABLE (  
ten VARCHAR(100),  
sodienthoai VARCHAR(15),  
tong_phat NUMERIC(15,2)  
) AS $$  
BEGIN  
RETURN QUERY (  
SELECT dg.ten, dg.sodienthoai,  
SUM(pm.tienphat::NUMERIC(15,2))  
AS tong_phat  
FROM docgia dg
```

```
JOIN phieumuon pm ON dg.iddocgia = pm.iddocgia
GROUP BY dg.ten, dg.sodienthoai
ORDER BY tong_phat DESC
LIMIT 10
);
END;
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM top_10_doc_gia_phat_cao_nhat();
```

9. Hàm cập nhật số lượng sách chưa trả cho báo cáo của tháng hiện tại:

-- Khởi tạo báo cáo tháng

```
CREATE OR REPLACE FUNCTION khai_tao_bao_cao_thang()
RETURNS VOID AS $$
BEGIN
INSERT INTO baocaothang (thoigian, thu, chi, soluongsachchuatra,
idnhanvien)
VALUES (TO_CHAR(CURRENT_DATE, 'YYYY-MM'), 0, 0, 0, 1);
END;
$$ LANGUAGE plpgsql;
```

-- Hàm cập nhật số lượng sách chưa trả

```
CREATE OR REPLACE FUNCTION cap_nhat_sach_chua_tra()
RETURNS VOID AS $$
BEGIN
```

-- Khởi tạo báo cáo tháng nếu chưa có

```
PERFORM khai_tao_bao_cao_thang();
```

-- Cập nhật số lượng sách chưa trả

```
UPDATE baocaothang
SET soluongsachchuatra = (
SELECT COUNT(*)
FROM phieumuon
)
WHERE thoigian = TO_CHAR(CURRENT_DATE, 'YYYY-MM');
END;
$$ LANGUAGE plpgsql;
```



```
SELECT cap_nhat_sach_chua_tra();
```

10. Hàm nhận vào 1 giá trị và trả về danh sách sách có số lần mượn lớn hơn giá trị đó:

```
CREATE OR REPLACE FUNCTION  
sach_co_so_luot_muon_lon_hon(gia_tri INT)  
RETURNS TABLE(idsach INT,  
ten VARCHAR,  
luotmuon INT) AS $$  
BEGIN  
RETURN QUERY  
SELECT sach.idsach, sach.ten, sach.luotmuon  
FROM sach  
WHERE sach.luotmuon > gia_tri;  
END;  
$$ LANGUAGE plpgsql;
```

-- Gọi hàm để lấy danh sách sách có số lần mượn lớn hơn (ví dụ: 10)

```
SELECT * FROM sach_co_so_luot_muon_lon_hon(10);
```

11. Hàm trả về danh sách 10 độc giả có số lượt mượn nhiều nhất:

```
CREATE OR REPLACE FUNCTION  
top_10_docgia_muon_nhieu_nhat()  
RETURNS TABLE(iddocgia INT,  
ten VARCHAR,  
soluotmuon INT) AS $$  
BEGIN  
RETURN QUERY  
SELECT d.iddocgia, d.ten, d.soluotmuon  
FROM docgia d  
ORDER BY d.soluotmuon DESC  
LIMIT 10;  
END;  
$$ LANGUAGE plpgsql;
```

-- Gọi hàm để lấy danh sách 10 độc giả có số lượt mượn nhiều nhất

```
SELECT * FROM top_10_docgia_muon_nhieu_nhat();
```

- **CÁC TRIGGER HỮU ÍCH:**

1. Trigger kiểm tra trạng thái của cuốn sách khi thêm phiếu mượn(nếu cuốn sách có trạng thái là đang mượn thì ko được thêm phiếu mượn đó):

-- Tạo function kiểm tra trạng thái của cuốn sách khi thêm phiếu mượn

```
CREATE OR REPLACE FUNCTION kiem_tra_trang_thai_sach()  
RETURNS TRIGGER AS $$
```

```
BEGIN
```

-- Kiểm tra trạng thái của cuốn sách

```
IF (SELECT trangthai
```

```
FROM sach
```

```
WHERE idsach = NEW.idsach) like '%Da muon%'
```

```
THEN
```

```
RAISE EXCEPTION 'Cuon sach nay dang duoc muon. Khong the  
them phieu muon nay.';
```

```
ELSE
```

```
RAISE NOTICE 'Them phieu muon moi';
```

```
RETURN NEW;
```

```
END IF;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

-- Tạo trigger để gọi function trước khi thêm phiếu mượn

```
CREATE OR REPLACE TRIGGER trg_kiem_tra_trang_thai_sach
```

```
BEFORE INSERT ON phieumuon
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION kiem_tra_trang_thai_sach();
```

2. Trigger cập nhật trạng thái sách khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION cap_nhat_trang_thai_sach()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
UPDATE sach
```

```
SET trangthai = 'Da muon'
```

```
WHERE idsach = NEW.idsach;
```

```
RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trg_cap_nhat_trang_thai_sach
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION cap_nhat_trang_thai_sach();
```

3. Trigger cập nhật số lượng sách chưa trả của tháng hiện tại trong báo cáo tháng khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION cap_nhat_sach_chua_tra1()
RETURNS TRIGGER AS $$
BEGIN
UPDATE baocaothang
SET soluongsachchuatra = soluongsachchuatra + 1
WHERE thoigian = TO_CHAR(CURRENT_DATE, 'YYYY-MM');

RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trg_cap_nhat_sach_chua_tra
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION cap_nhat_sach_chua_tra1();
```

4. Trigger cập nhật lượt mượn của sách khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION cap_nhat_luot_muon_sach()
RETURNS TRIGGER AS $$
BEGIN
UPDATE sach
SET luotmuon = luotmuon + 1
WHERE idsach = NEW.idsach;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trg_cap_nhat_luot_muon_sach
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION cap_nhat_luot_muon_sach();
```

5. Trigger cập nhật soluotmuon của bảng độc giả khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION update_soluotmuon()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE docgia
    SET soluotmuon = soluotmuon + 1
    WHERE iddocgia = NEW.iddocgia;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER update_soluotmuon_trigger
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION update_soluotmuon();
```

6. Trigger cập nhật doanh thu của sách sau khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION cap_nhat_doanh_thu_sach()
RETURNS TRIGGER AS $$
BEGIN
    -- Cập nhật doanh thu của sách
    UPDATE sach
    SET doanhthu = doanhthu + (SELECT phimuon FROM sach WHERE
    idsach = NEW.idsach)
    WHERE idsach = NEW.idsach;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_cap_nhat_doanh_thu_sach
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION cap_nhat_doanh_thu_sach();
```

7. Trigger cập nhật cột "thu" trong bảng báo cáo tháng(+ phí mượn) khi thêm phiếu mượn:

```
CREATE OR REPLACE FUNCTION
cap_nhat_thu_khi_them_phieu_muon()
RETURNS TRIGGER AS $$
DECLARE
phimuon_sach INT;
BEGIN
-- Lấy phí mượn từ bảng sách
SELECT s.phimuon INTO phimuon_sach FROM sach s WHERE
s.idsach = NEW.idsach;
-- Cập nhật cột "thu" trong bảng báo cáo tháng
UPDATE baocaothang
SET thu = thu + phimuon_sach
WHERE thoigian = TO_CHAR(CURRENT_DATE, 'YYYY-MM');

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_cap_nhat_thu_khi_them_phieu_muon
AFTER INSERT ON phieumuon
FOR EACH ROW
EXECUTE FUNCTION cap_nhat_thu_khi_them_phieu_muon();
```

**8. Trigger cập nhật cột "thu" trong bảng báo cáo tháng(+ tiền phạt)
khi xóa phiếu mượn:**

```
CREATE OR REPLACE FUNCTION
cap_nhat_thu_khi_xoa_phieu_muon()
RETURNS TRIGGER AS $$
DECLARE
so_ngay_qua_han INT;
tien_phat NUMERIC;
phimuon_sach INT;
BEGIN
-- Lấy phí mượn từ bảng sách
SELECT s.phimuon INTO phimuon_sach FROM sach s WHERE
s.idsach = OLD.idsach;
IF CURRENT_DATE > OLD.ngaydenhan THEN
so_ngay_qua_han := CURRENT_DATE - OLD.ngaydenhan;
ELSE
so_ngay_qua_han := 0;
```

```
END IF;
```

```
tien_phat := 0.05 * so_ngay_qua_han * phimuon_sach;
```

```
UPDATE baocaothang
```

```
SET thu = thu + tien_phat
```

```
WHERE thoigian = TO_CHAR(CURRENT_DATE, 'YYYY-MM');
```

```
RETURN OLD;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_cap_nhat_thu_khi_xoa_phieu_muon
```

```
AFTER DELETE ON phieumuon
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION cap_nhat_thu_khi_xoa_phieu_muon();
```

9. Trigger cập nhật trạng thái sách khi xóa phiếu mượn:

```
CREATE OR REPLACE FUNCTION
```

```
cap_nhat_trang_thai_sach_khi_xoa()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
IF NOT EXISTS (
```

```
SELECT 1
```

```
FROM phieumuon
```

```
WHERE idsach = OLD.idsach AND idphieumuon !=
```

```
OLD.idphieumuon
```

```
) THEN
```

```
UPDATE sach
```

```
SET trangthai = 'Chua muon'
```

```
WHERE idsach = OLD.idsach;
```

```
END IF;
```

```
RETURN OLD;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_cap_nhat_trang_thai_sach_khi_xoa
```

```
AFTER DELETE ON phieumuon
```

```
FOR EACH ROW  
EXECUTE FUNCTION cap_nhat_trang_thai_sach_khi_xoa();
```

10. Trigger cập nhật số lượng sách chưa trả của tháng hiện tại trong báo cáo tháng khi xóa phiếu mượn:

```
CREATE OR REPLACE FUNCTION  
cap_nhat_bao_cao_thang_sach_chua_tra_khi_xoa()  
RETURNS TRIGGER AS $$  
BEGIN  
UPDATE baocaothang  
SET soluongsachchuatra = soluongsachchuatra - 1  
WHERE thoigian = TO_CHAR(CURRENT_DATE, 'YYYY-MM')  
AND soluongsachchuatra > 0; -- Đảm bảo số lượng sách chưa trả  
không âm  
  
RETURN OLD;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trg_cap_nhat_sach_chua_tra_khi_xoa  
AFTER DELETE ON phieumuon  
FOR EACH ROW  
EXECUTE FUNCTION  
cap_nhat_bao_cao_thang_sach_chua_tra_khi_xoa();
```

11. Trigger kiểm tra tài khoản của độc giả khi thêm mới(kiểm tra đã tồn tại chưa):

```
CREATE OR REPLACE FUNCTION kiem_tra_tai_khoan_doc_gia()  
RETURNS TRIGGER AS $$  
DECLARE  
count_account INT;  
BEGIN  
SELECT COUNT(*) INTO count_account  
FROM docgia  
WHERE taikhoan = NEW.taikhoan;  
  
IF count_account > 0 THEN  
RAISE EXCEPTION 'Tài khoản % đã tồn tại trong hệ thống!',  
NEW.taikhoan;  
END IF;
```

```
RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
CREATE TRIGGER trg_kiem_tra_tai_khoan_doc_gia  
BEFORE INSERT ON docgia  
FOR EACH ROW  
EXECUTE FUNCTION kiem_tra_tai_khoan_doc_gia();
```

12. Trigger kiểm tra tài khoản của nhân viên khi thêm mới(kiểm tra đã tồn tại chưa):

```
CREATE OR REPLACE FUNCTION  
kiem_tra_tai_khoan_nhan_vien()  
RETURNS TRIGGER AS $$  
DECLARE  
count_account INT;  
BEGIN  
SELECT COUNT(*) INTO count_account  
FROM nhanvien  
WHERE taikhoan = NEW.taikhoan;  
  
IF count_account > 0 THEN  
RAISE EXCEPTION 'Tài khoản % đã tồn tại trong hệ thống!',  
NEW.taikhoan;  
END IF;  
  
RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
CREATE TRIGGER trg_kiem_tra_tai_khoan_nhan_vien  
BEFORE INSERT ON nhanvien  
FOR EACH ROW  
EXECUTE FUNCTION kiem_tra_tai_khoan_nhan_vien();
```

- MỘT VÀI CÂU INDEX:

-- Tạo index cho cột idnhanvien trong bảng nhanvien:

```
CREATE INDEX idx_nhanvien_idnhanvien ON nhanvien (idnhanvien);
```

-- Tạo index cho cột taikhoan trong bảng nhanvien:


```
CREATE INDEX idx_nhanvien_taikhoan ON nhanvien (taikhoan);
```

-- Tạo index cho cột iddocgia trong bảng docgia:

```
CREATE INDEX idx_docgia_iddocgia ON docgia (iddocgia);
```

-- Tạo index cho cột sodienthoai trong bảng docgia:

```
CREATE INDEX idx_docgia_sodienthoai ON docgia (sodienthoai);
```

-- Tạo index cho cột thoigian trong bảng baocaothang:

```
CREATE INDEX idx_baocaothang_thoigian ON baocaothang (thoigian);
```

-- Tạo index cho cột idsach trong bảng sach:

```
CREATE INDEX idx_sach_idsach ON sach (idsach);
```

-- Tạo index cho cột phanloai trong bảng sach:

```
CREATE INDEX idx_sach_phanloai ON sach (phanloai);
```

-- Tạo index cho cột idphieumuon trong bảng phieumuon:

```
CREATE INDEX idx_phieumuon_idphieumuon ON phieumuon  
(idphieumuon);
```

-- Tạo index cho cột ngaymuon trong bảng phieumuon:

```
CREATE INDEX idx_phieumuon_ngaymuon ON phieumuon  
(ngaymuon);
```

-- Tạo index cho cột tacgia trong bảng sach:

```
CREATE INDEX idx_sach_tacgia ON sach (tacgia);
```

-- Tạo index cho cột nhaxuatban trong bảng sach:

```
CREATE INDEX idx_sach_nhaxuatban ON sach (nhaxuatban);
```

IX. Trang web quản lý thư viện :

1. Công nghệ tạo web :

- Giao diện : HTML, CSS
- Tính năng : PHP
- Cơ sở dữ liệu : SQL
- Localhost : XAMPP

2. Mô tả web :

- Cho phép độc giả và nhân viên đăng nhập vào tài khoản cá nhân :
- + Tài khoản nhân viên : hiển thị thông tin cá nhân của nhân viên

STAFF DETAILS

PROFILE
CUSTOMER
REPORT
BOOK
LOAN SLIP

STAFF ID:

1

FULL NAME:

Nguyen Van A

POSITION:

Thu Thu

PHONE NUMBER:

0123456789

- + Tài khoản độc giả : hiển thị thông tin cá nhân của độc giả

READER DETAILS

PROFILE
LOAN SLIP
REQUEST
BOOK

USER ID:

1

USERNAME:

levanc

FULL NAME:

Le Van C

PHONE NUMBER:

0123456789

AGE:

20

ADRESS:

Ha Noi

TIME BORROWEDS:

0

- Cho phép nhân viên quản lý danh sách độc giả : hiển thị danh sách các độc giả, cho phép tìm kiếm độc giả, chỉnh sửa thông tin và thêm mới độc giả

CUSTOMER'S LIST

CUSTOMER

PROFILE

REPORT

BOOK

LOAN SLIP

ID

Fullname

Username

Phone Number

Times Borroweds

ADD

SEARCH

ID	FULL NAME	USERNAME	PHONE NUMBER	TIMES BORROWEDS	ACTION
1	Le Van C	levanc	0123456789	0	<div>Remove</div> <div>Edit</div>
2	Pham Thi D	phamthid	0987654321	0	<div>Remove</div> <div>Edit</div>
3	Le Van Cao	levanco	0123456789	0	<div>Remove</div> <div>Edit</div>
4	Pham Thi Duong	phamthiduong	0987654321	0	<div>Remove</div> <div>Edit</div>
5	Tran Van E	tranve	0111111111	0	<div>Remove</div> <div>Edit</div>
6	Hoang Thi F	hoangf	0222222222	0	<div>Remove</div> <div>Edit</div>
7	Nguyen Van G	nguyeng	0333333333	0	<div>Remove</div> <div>Edit</div>
8	Le Thi H	lethih	0444444444	0	<div>Remove</div> <div>Edit</div>
9	Tran Thi I	tranthi	0555555555	1	<div>Remove</div> <div>Edit</div>
10	Pham Van K	phamvk	0666666666	1	<div>Remove</div> <div>Edit</div>
11	Hoang Van L	hoangvl	0777777777	1	<div>Remove</div> <div>Edit</div>
12	Nguyen Thi M	nguyenm	0888888888	1	<div>Remove</div> <div>Edit</div>
13	Le Van N	levann	0999999999	1	<div>Remove</div> <div>Edit</div>
14	Tran Van O	tranvo	0101010101	1	<div>Remove</div> <div>Edit</div>
15	Pham Thi P	phamtp	0202020202	1	<div>Remove</div> <div>Edit</div>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

- Cho phép nhân viên quản lý báo cáo tháng : hiển thị danh sách thu chi các tháng, cho phép tìm kiếm, thêm mới và chỉnh sửa thông tin

REPORT

PROFILE

CUSTOMER

BOOK

LOAN SLIP

Time

Income

Expense

Quantity

Staffid

ADD

SEARCH

TIME	INCOME	EXPENSE	QUANTITY	STAFF ID	ACTION
2004-01	220000	170000	5	12	<div>Remove</div> <div>Edit</div>
2005-01	220000	170000	5	133	<div>Remove</div> <div>Edit</div>
2005-02	240000	160000	6	134	<div>Remove</div> <div>Edit</div>
2005-03	260000	190000	4	135	<div>Remove</div> <div>Edit</div>
2005-04	280000	200000	3	136	<div>Remove</div> <div>Edit</div>
2005-05	230000	180000	5	137	<div>Remove</div> <div>Edit</div>
2005-06	250000	180000	4	138	<div>Remove</div> <div>Edit</div>
2005-07	270000	190000	3	139	<div>Remove</div> <div>Edit</div>
2005-08	290000	200000	2	140	<div>Remove</div> <div>Edit</div>
2005-09	210000	160000	6	141	<div>Remove</div> <div>Edit</div>
2005-10	230000	180000	5	142	<div>Remove</div> <div>Edit</div>
2005-11	250000	180000	4	143	<div>Remove</div> <div>Edit</div>
2005-12	270000	190000	3	144	<div>Remove</div> <div>Edit</div>
2006-01	220000	170000	5	145	<div>Remove</div> <div>Edit</div>
2006-02	240000	160000	6	146	<div>Remove</div> <div>Edit</div>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

- Cho phép nhân viên quản lý thông tin các đầu sách : hiển thị thông tin, tìm kiếm và thêm mới sách, chỉnh sửa thông tin và sắp xếp sách theo yêu cầu

BOOK

PROFILE

CUSTOMER

REPORT

LOAN SLIP

ID

Name

Genre

Times Borrowed

Price

Status

ADD

SEARCH

ID

Ascending

Sort

ID	NAME	GENRE	TIMES BORROWED	PRICE	STATUS	ACTION
1	Sach Khoa Hoc	Khoa Hoc	10	100000	Da muon	Remove Edit
2	Sach Van Hoc	Van Hoc	5	120000	Da muon	Remove Edit
3	Sach Lich Su	Lich Su	8	150000	Da muon	Remove Edit
4	Sach Kinh Te	Kinh Te	12	110000	Da muon	Remove Edit
5	Sach Toan Hoc	Toan Hoc	15	90000	Da muon	Remove Edit
6	Sach Van Hoc	Van Hoc	7	130000	Da muon	Remove Edit
7	Sach Lich Su	Lich Su	9	140000	Da muon	Remove Edit
8	Sach Kinh Te	Kinh Te	11	160000	Da muon	Remove Edit
9	Sach Toan Hoc	Toan Hoc	13	95000	Da muon	Remove Edit
10	Sach Van Hoc	Van Hoc	6	135000	Da muon	Remove Edit
11	Sach Khoa Hoc	Khoa Hoc	10	100000	Da muon	Remove Edit
12	Sach Van Hoc	Van Hoc	5	120000	Da muon	Remove Edit
13	Sach Lich Su	Lich Su	8	150000	Da muon	Remove Edit
14	Sach Kinh Te	Kinh Te	12	110000	Da muon	Remove Edit
15	Sach Toan Hoc	Toan Hoc	15	90000	Da muon	Remove Edit

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

- Cho phép nhân viên quản lý các phiếu mượn : hiển thị thông tin các phiếu, cho phép thêm mới và tìm kiếm các phiếu, trao quyền chỉnh sửa các phiếu

LOAN SLIP

PROFILE

CUSTOMER

REPORT

BOOK

ID

Loan Fee

Loan Day

Expired Day

Book ID

ADD

SEARCH

LOAN SLIP ID	LOAN FEE	LOAN DAY	EXPIRED DAY	BOOK ID	ACTION
1	50000	2024-01-01	2024-01-15	1	Remove Edit
2	75000	2024-01-03	2024-01-17	2	Remove Edit
3	100000	2024-01-05	2024-01-20	3	Remove Edit
4	65000	2024-01-07	2024-01-22	4	Remove Edit
5	80000	2024-01-09	2024-01-24	5	Remove Edit
6	55000	2024-01-11	2024-01-26	6	Remove Edit
7	90000	2024-01-13	2024-01-28	10	Remove Edit
8	85000	2024-01-15	2024-01-30	7	Remove Edit
9	70000	2024-01-17	2024-02-01	8	Remove Edit
10	95000	2024-01-19	2024-02-03	9	Remove Edit
11	45000	2024-01-21	2024-02-05	13	Remove Edit
12	65000	2024-01-23	2024-02-07	12	Remove Edit
13	75000	2024-01-25	2024-02-09	11	Remove Edit
14	80000	2024-01-27	2024-02-11	17	Remove Edit
15	50000	2024-01-29	2024-02-13	16	Remove Edit

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

- Cho phép độc giả xem phiếu thông tin phiếu mượn của bản thân :

LOAN SLIP'S LIST

LOAN SLIP

PROFILE

REQUEST

BOOK

LOAN SLIP ID	LOAN FEE	LOAN DAY	EXPIRED DAY	BOOK ID
1	50000	2024-01-01	2024-01-15	1

- Cho phép độc giả gửi yêu cầu mượn sách :

LOAN SLIP'S LIST

REQUEST

PROFILE

LOAN SLIP

BOOK

Loan Request

Loan ID:
238

Book ID:

Reason for Request:

Submit Request

+ Nếu sách có trạng thái chưa mượn, độc giả có thêm 1 phiếu mượn :

LOAN SLIP

PROFILE

REQUEST

BOOK

LOAN SLIP'S LIST

LOAN SLIP ID	LOAN FEE	LOAN DAY	EXPIRED DAY	BOOK ID
1	50000	2024-01-01	2024-01-15	1
237	140000	2024-06-15	2024-07-15	237

- Cho phép độc giả xem, tìm kiếm sách theo tên hoặc theo thứ tự được sắp xếp theo yêu cầu :

BOOK

PROFILE

LOAN SLIP

REQUEST

BOOK'S LIST

ID

Name

Genre

Times Borrowed

Price

Status

SEARCH

ID

Ascending

Sort

ID	NAME	GENRE	TIMES BORROWED	PRICE	STATUS
1	Sach Khoa Hoc	Khoa Hoc	10	100000	Da mượn
2	Sach Van Hoc	Van Hoc	5	120000	Da mượn
3	Sach Lich Su 5	Lich Su	8	150000	Da mượn
4	Sach Kinh Te 6	Kinh Te	12	110000	Da mượn
5	Sach Toan Hoc 7	Toan Hoc	15	90000	Da mượn
6	Sach Van Hoc 8	Van Hoc	7	130000	Da mượn
7	Sach Lich Su 9	Lich Su	9	140000	Da mượn
8	Sach Kinh Te 10	Kinh Te	11	160000	Da mượn
9	Sach Toan Hoc 11	Toan Hoc	13	95000	Da mượn
10	Sach Van Hoc 12	Van Hoc	6	135000	Da mượn
11	Sach Khoa Hoc Tre	Khoa Hoc	10	100000	Da mượn
12	Sach Van Hoc Vui	Van Hoc	5	120000	Da mượn
13	Sach Lich Su Hay	Lich Su	8	150000	Da mượn
14	Sach Kinh Te Li Thu	Kinh Te	12	110000	Da mượn
15	Sach Toan Hoc 27	Toan Hoc	15	90000	Da mượn

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

X. Đánh giá báo cáo

1. Một vài khó khăn khi thực hiện dự án và giải pháp
 - Khó khăn trong việc thiết kế cấu trúc dữ liệu sao cho tối ưu và dễ mở rộng
Giải pháp: Áp dụng kiến thức lý thuyết đã học để thiết kế và sử dụng các công cụ giúp trực quan hóa cấu trúc dữ liệu
 - Khó khăn trong việc khởi tạo dữ liệu: Để có cái nhìn trực quan và tổng quát, cần một lượng lớn dữ liệu. Tuy nhiên, không tìm được dữ liệu có sẵn mà việc khởi tạo dữ liệu gặp một số khó khăn như: lượng dữ liệu cần khởi tạo quá lớn, các dữ liệu không có mối liên kết với nhau
Giải pháp: Ứng dụng một số công cụ hỗ trợ trong việc khởi tạo dữ liệu.
 - Khó khăn trong việc tạo web: Muốn tạo web và vận hành web trơn tru thì cần nhiều kiến thức liên quan đến nhiều lĩnh vực khác nhau. Tuy nhiên, trong nhóm chưa ai có kinh nghiệm trong lĩnh vực này, dẫn đến mất nhiều thời gian để hoàn thành
Giải pháp: Tìm hiểu và ứng dụng từng phần nhỏ kiến thức để viết các chương trình nhỏ, sau đó mới tổng hợp lại thành chương trình hoàn chỉnh
 - Khó khăn trong việc kiểm tra tính đúng sai của các câu truy vấn và các hàm: Sau khi viết ra các câu truy vấn áp dụng cho các tính năng của cơ sở dữ liệu, thì cần nhiều thời gian để có thể kiểm tra xem hàm có chính xác hay không
Giải pháp: Ứng dụng một số công cụ hỗ trợ và dành thời gian kiểm tra thủ công
 - Khó khăn trong việc tối ưu hóa hệ thống: Thiết kế CSDL cần phải tối ưu hóa hiệu suất truy vấn và xử lý dữ liệu, đặc biệt khi khối lượng dữ liệu lớn.
Giải pháp: Áp dụng Index
2. Đánh giá sản phẩm
 - Ưu điểm:
 - + Đây là một hệ thống hoàn thiện, với đầy đủ các phần cần có cho một hệ cơ sở dữ liệu
 - + Hệ thống hoạt động ổn định mà không tạo ra lỗi
 - + Hệ thống đáp ứng được một vài các tính năng cơ bản và cần có
 - + Hệ thống đã được tối ưu hóa
 - Nhược điểm:
 - + Sản phẩm vẫn chưa thực sự là một sản phẩm hoàn hảo vì còn nhiều hạn chế về mặt tính năng
 - + Trang web được tạo ra thiếu tính thẩm mỹ và chuyên nghiệp
 - + Thiếu tính năng bảo mật dữ liệu

XI. Thành viên nhóm

1. Bùi Quốc Bảo: (Nhóm trưởng)
 - Mã số sinh viên: 20225601
 - Email: bao.bq225601@sis.hust.edu.vn
 - Lớp: CNTT Việt Nhật 04-K67
 - Nhiệm vụ:
 - + Xây dựng sơ đồ thực liên kết
 - + Khởi tạo data
 - + Làm Back-End
 - + Viết câu truy vấn
 - + Viết Function, Trigger
2. Đặng Kim Ngân:
 - Mã số sinh viên: 20225751
 - Email: ngan.dk225751@sis.hust.edu.vn
 - Lớp: CNTT Việt Nhật 05-K67
 - Nhiệm vụ:
 - + Khởi tạo data
 - + Viết câu truy vấn
 - + Làm Front- End
 - + Viết báo cáo, làm Power Point
3. Vũ Trần Tuấn Minh:
 - Mã số sinh viên: 20225891
 - Email: minh.vtt225891@sis.hust.edu.vn
 - Lớp: CNTT Việt Nhật 04-K67
 - Nhiệm vụ:
 - + Xây dựng sơ đồ quan hệ
 - + Viết câu truy vấn
 - + Viết Function, Trigger
 - + Viết báo cáo