

Trường Đại học Bách khoa Hà nội

Viện CNTT & TT

CHỦ ĐỀ: QUẢN LÝ THƯ VIỆN SÁCH

GV hướng dẫn: T.S NGUYỄN THỊ OANH

Mã lớp: 121261

Nhóm sinh viên thực hiện:

Thiều Ngọc Hiền Kiệt: 20184128

Nguyễn Thị Ánh Hồng: 20184112

Nguyễn Thị Hà: 20184086

I. Mô tả bài toán

- Thư viện lưu trữ sách bao gồm các thông tin: mã sách, tên sách, năm xuất bản, số lượng đang có ở thư viện, nhà xuất bản, ngôn ngữ, thể loại, tác giả
 - Một cuốn sách được cung cấp bởi một nhà xuất bản, một nhà xuất bản có thể cung cấp một hoặc nhiều sách cho thư viện. Nhà xuất bản có mã nhà xuất bản, tên nhà xuất bản, địa chỉ.
 - Một cuốn sách có thể được viết bởi nhiều tác giả, một tác giả có thể viết một hoặc nhiều cuốn sách. Mỗi tác giả sẽ có mã tác giả, tên tác giả, địa chỉ.
 - Một loại sách có thể có một hoặc nhiều cuốn sách, mỗi cuốn sách chỉ thuộc một loại sách nhất định. Loại sách bao gồm mã loại, tên loại.
 - Một cuốn sách được viết bằng một ngôn ngữ nhất định. Ngôn ngữ có mã ngôn ngữ, tên ngôn ngữ.

- Thư viện quản lý sinh viên có nhu cầu mượn sách qua thẻ đọc giả bao gồm các thông tin như mã thẻ, mã số sinh viên, ngày cấp và ngày hết hạn.
 - Mỗi thẻ chỉ được sở hữu bởi một sinh viên. Thẻ sinh viên cần quản lý một số thông tin sau: mã sinh viên, tên sinh viên, lớp, địa chỉ.
 - Sinh viên phải làm thẻ đọc giả thì mới được mượn sách trong thư viện

- Thư viện quản lý việc mượn trả sách của sinh viên qua thẻ mượn bao gồm thông tin: mã thẻ mượn, mã số sinh viên, hình thức mượn, ngày mượn, ngày phải trả, ngày trả, mã sách, số lượng mượn của sách đó
 - Một sinh viên có thể mượn được một hoặc nhiều cuốn sách, một cuốn sách có thể không có sinh viên nào mượn hoặc có một hoặc nhiều sinh viên mượn.
 - Khi sinh viên mượn sách, thông tin sẽ được lưu lại trong thẻ mượn với tối đa 10 cuốn sách/lượt và thời gian mượn là 2 tháng/cuốn sách
 - Hình thức mượn gồm có online và offline
 - Sinh viên cần trả sách đúng hạn, nếu quá hạn thì sẽ bị cảnh cáo

II. Chức năng hệ thống

1. Quản trị hệ thống- Quản lý người dùng: Cung cấp cho user tài khoản trên hệ thống, với trường thông tin cơ

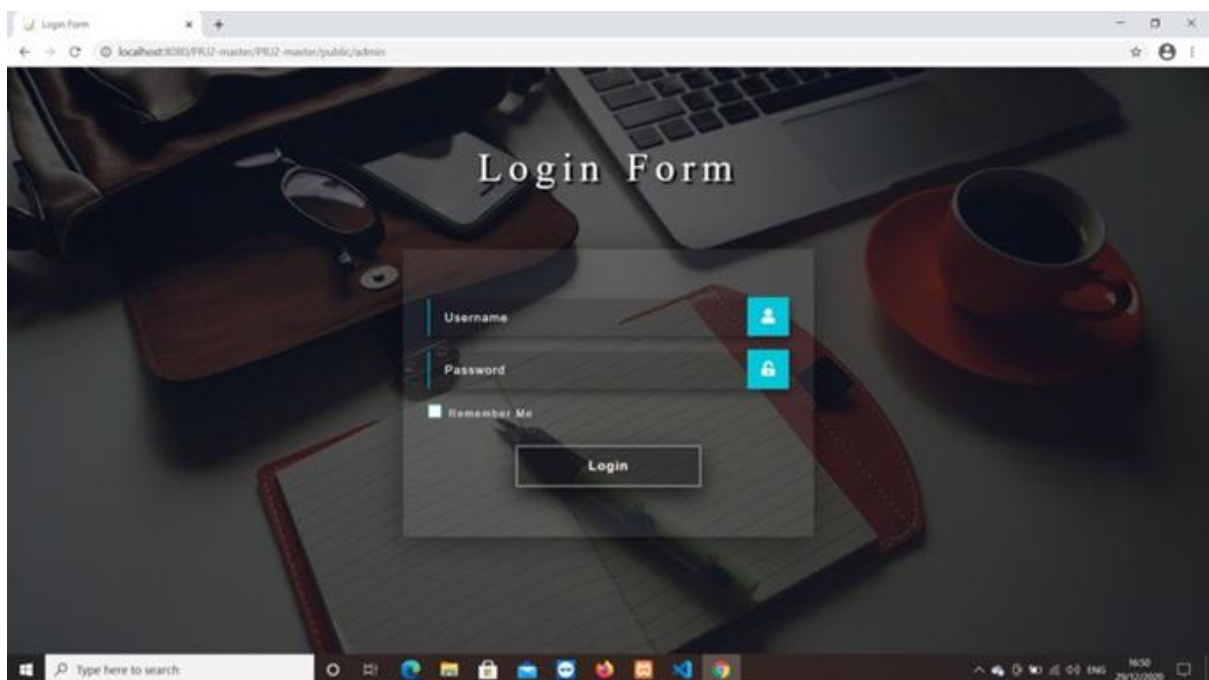
bản nhất là tên đăng nhập và mật khẩu

- Phân quyền: Mỗi user là người quản lý, nhân viên, khách hàng (bạn đọc) sẽ có những quyền khác nhau với các chức năng xem, thêm sửa, xóa, sửa các đối tượng cơ bản trên hệ thống

- Chỉnh sửa thông tin cá nhân

- Sao lưu, phục hồi thông tin: hỗ trợ trong các tình huống hệ thống gặp sự cố về thông tin, cho phép tìm kiếm hồ ớ, thông tin bạn đọc theo nhiều trường dữ liệu khác nhau giúp việc tìm kiếm dễ dàng, nhanh chóng

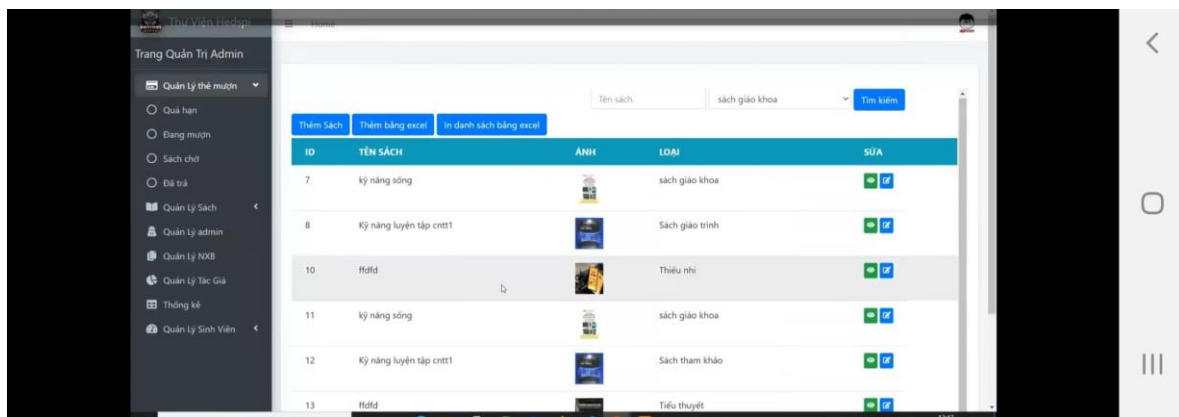
*Giao diện khi admin đăng nhập



2. Quản lý sách:

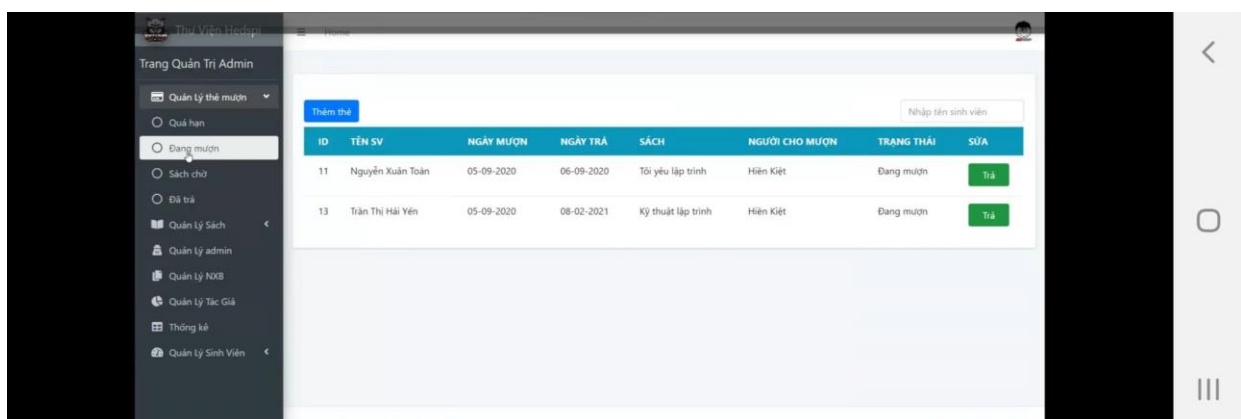
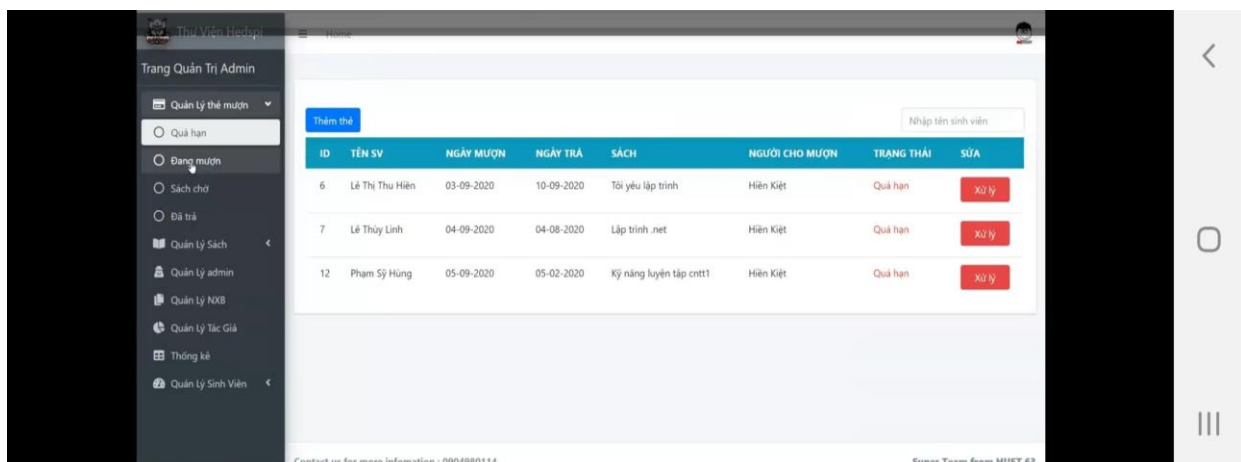
- Quản lý các thông tin đến sách: mã sách, tên sách, thể loại, ngôn ngữ, tác giả, người dịch, nhà xuất bản, số lượng,...

- Quản lý về sách như hư hỏng, tổn thất, mất sách,...



3. Quản lý thẻ thư viện

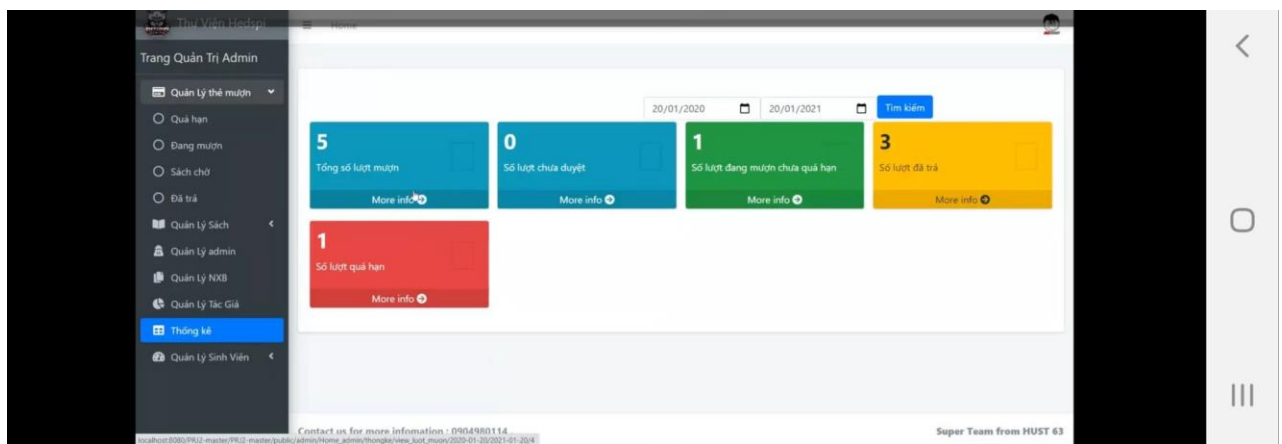
- Cho phép in và cấp thẻ cho độc giả mới trong việc quản lý thẻ bạn độc cũng như các dịch vụ mượn, trả và gia hạn tài liệu.
- Quản lý số lượng sách bạn đọc mượn, trả của thư viện
- Quản lý ngày, giờ, số lượng đầu sách độc giả mượn
- Quản lý số lượng sách độc giả được phép mượn tối đa (tối thiểu)



| ID | TÊN SV | NGÀY MƯỢN | NGÀY TRẢ | SÁCH | NGƯỜI CHO MƯỢN | TRẠNG THÁI |
|----|-------------------|------------|------------|-------------------------|----------------|------------|
| 2 | Nguyễn Thành Long | 02-02-2020 | 02-03-2020 | Kỹ năng luyện tập cntt1 | Hiện Kiệt | Đã trả |
| 3 | Nguyễn Long Vũ | 03-09-2020 | 09-09-2020 | kỹ năng sống | Hiện Kiệt | Đã trả |
| 4 | Trần Ngọc Toàn | 03-09-2020 | 25-09-2020 | Tôi yêu lập trình | Hiện Kiệt | Đã trả |
| 5 | Dương Thành Đạt | 03-09-2020 | 27-01-0001 | Tôi yêu lập trình | Hiện Kiệt | Đã trả |
| 8 | Doãn Hải Phong | 04-09-2020 | 02-02-2222 | ffdfdf | Hiện Kiệt | Đã trả |
| 9 | Mai Đình Trung | 04-09-2020 | 02-06-2021 | kỹ năng sống | Hiện Kiệt | Đã trả |
| 10 | Vũ Nhật Vi | 04-09-2020 | 09-09-2029 | Kỹ năng luyện tập cntt1 | Hiện Kiệt | Đã trả |

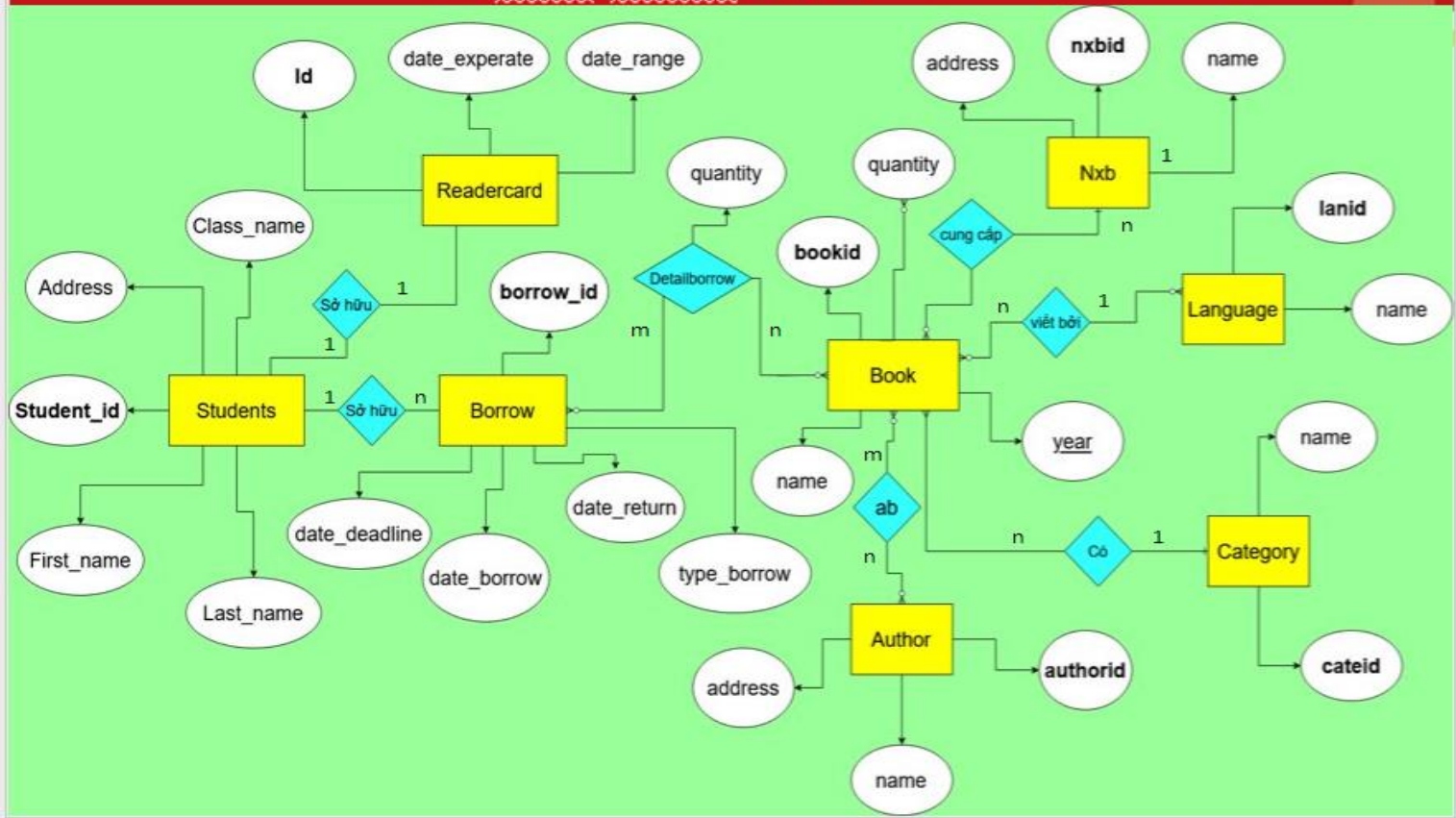
4. Thống kê

Cho phép biết số lượt mượn , số sách đã trả , số sách chưa trả và quá hạn trong thời gian nhập vào



III. Mô hình ERD

Mô hình ERD



IV. Mô hình quan hệ

author (authorid , name ,address)

category (cateid , name)

language (lanid , name)

nxb (nxbid , name , address)

book (bookid , name, year, quantity, *cateid*, *lanid*, *nxbid*)

students (student_id , first_name, last_name, class_name, address)

readercard (id , student_id , date_range, date_experate)

borrow(borrow_id , *student_id* , type_borrow , date_borrow, date_deadline, date_return)

ab (*bookid*, *authorid*)

detailborrow (*borrow_id*, *bookid*, quantity)

V. Phân chia công việc

| | |
|----------------------|--|
| Thiều Ngọc Hiền Kiệt | - Thiết kế cơ sở dữ liệu - Làm website |
| Nguyễn Thị Hà | - Thiết kế cơ sở dữ liệu, thêm và chỉnh sửa dữ liệu - Làm powperpoint |
| Nguyễn Thị Ánh Hồng | - Thiết kế cơ sở dữ liệu, thêm và chỉnh sửa dữ liệu - Làm báo cáo |

VI. Câu truy vấn của các thành viên

1. Nguyễn Thị Ánh Hồng 20184112

Câu 1. đưa ra danh sách những sách viết bằng tiếng nhật đã đc mượn

```
select b.name  
from language lan join book b using(lanid)  
join detailborrow using (bookid)  
where lan.name='Tiếng nhật'
```

Câu 2. in ra sách chưa đc ai mượn

```
select b.*  
from book b  
where bookid not in ( select bookid from detailborrow)
```

******Câu 3. tác giả đc ưa chuộng nhất

```
--c1 25.44 .. 69.74
```

explain

```

select a.name, count (db.borrow_id)
from author a join ab using (authorid)
      join book b using (bookid)
      join detailborrow db using(bookid)
group by (a.authorid)
having count(db.borrow_id)>= all (select count(db.borrow_id)
      from ab join book using (bookid)
      join detailborrow db using(bookid)
      where ab.authorid is not null
      group by (authorid))
--c2 12.38 .. 14.04
explain
with tmp_auth as(
  select a.name, count(db.borrow_id) count_quantity
  from author a, ab, detailborrow db
  where a.authorid = ab.authorid
        and ab.bookid = db.bookid
  group by a.name)

select name, count_quantity
from tmp_auth
where count_quantity = (select max(count_quantity) from tmp_auth);

select * from author
select * from detailborrow

```

Đánh giá : Cách thứ nhất là duyệt 2 truy vấn lấy ra số lượng lượt được mượn sách của mỗi tác giả rồi mới so sánh với nhau

Cách thứ hai dùng bảng tạm tmp lấy ra số lượng lượt mượn sách của mỗi tác giả và sau đó chỉ cần gọi bảng tmp và lấy max của số lượt đó nên cách 2 chỉ duyệt 1 lần

Nên cách thứ 2 tối ưu hơn

```
select * from author
select * from detailborrow
```

Câu 4. in ra quyển sách của tác giả ".." đc mượn nhiều nhất

```
select b.name, count(borrow_id)
from author a join ab using (authorid)
      join book b using (bookid)
      join detailborrow db using (bookid)
where a.name='Robin Sharma'
group by(b.bookid)
having count(borrow_id) >= all (select count(borrow_id)
      from author a join ab using (authorid)
            join book b using (bookid)
            join detailborrow db using (bookid)
            where a.name='Robin Sharma'
            group by(b.bookid))
```

Câu 5. in ra số phiếu mượn trong năm 2020

```
select count(borrow_id)
from borrow
where date_part('year',date_borrow)=2020
```

****Câu 6.** đưa ra những sinh viên đã mượn 2 quyển sách 'Mặc kệ nó làm tới đi', 'Đời ngắn lắm đừng ngủ dài'

```
--c1
explain --3.5..12.03
select s.first_name,s.last_name
from students s join borrow br using (student_id)
```

```

join detailborrow db using (borrow_id)
      join book b using (bookid)
      where b.name='Đánh bại phố Wall'
intersect
select s.first_name, s.last_name
from students s join borrow br using (student_id)
      join detailborrow db using (borrow_id)
      join book b using (bookid)
      where b.name='Đời ngắn đừng ngủ dài'

```

--c2

```

explain --6.9..12.02
select s.first_name,s.last_name
from students s join borrow br using (student_id)
      join detailborrow db using (borrow_id)
      join book b using (bookid)
where b.name='Đánh bại phố Wall'
      and s.student_id in (select s.student_id
                           from students s join borrow br using (student_id)
                           join detailborrow db using (borrow_id)
                           join book b using (bookid)
                           where b.name='Đời ngắn đừng ngủ dài')

```

Đánh giá : *Cách thứ nhất dùng 2 truy vấn độc lập sau đó lấy giao kết quả cả 2 truy vấn đó.*

Cách thứ 2 dùng 2 truy vấn lồng nhau, mỗi lần duyệt 1 bản ghi thì câu truy vấn con. Nên cách thứ 2 tối ưu hơn

Câu 7. in ra sách và số lượng sách ban đầu trong kho (hiện tại + đang mượn)

```
alter table book add so_luong_ban_dau int;
```

```
select *from book
```

```
with tmp as(
```

```
    select b.bookid, sum(db.quantity) a
```

```
    from book b join detailborrow db using (bookid)
```

```
    join borrow br using (borrow_id)
```

```
    where date_return is null
```

```
    group by (b.bookid)
```

```
)
```

```
select b.name, (quantity+a) so_luong_ban_dau
```

```
from book b join tmp using (bookid)
```

```
union
```

```
select name, quantity so_luong_ban_dau
```

```
from book
```

```
where bookid not in (select bookid from tmp)
```

Câu 8. thống kê theo lớp học về số lượt mượn cho mỗi thể loại, sx giảm dần

```
select class_name , cate.name,count(db.bookid)
```

```
from category cate join book b using (cateid)
```

```
    join detailborrow db using (bookid)
```

```
join borrow br using (borrow_id)
```

```
join students s using (student_id)
```

```
group by (class_name,cate.cateid)
```

```
order by (class_name,cate.cateid),count(db.bookid)
```

Câu 9. trigger update số lượng sách sau khi đc trả

```
create trigger af_quantity
```

```
after update on borrow
```

```
for each row
when (new.date_return is not null)
execute procedure tg_af_quantity();
```

```
create or replace function tg_af_quantity() returns trigger as
$$
begin
update book set quantity = quantity +1
      where bookid in (select bookid
      from detailborrow
      where borrow_id = old.borrow_id) ;
return new;
end;
$$ language plpgsql;
```

```
drop trigger af_quantity on borrow
drop function tg_af_quantity()
```

```
select * from detailborrow where borrow_id=02
select * from borrow where borrow_id =02
select * from book where bookid=12 or bookid=33
update borrow set date_return='2021-01-04' where borrow_id=02
```

```
select * from detailborrow
```

Câu 10. function cập nhật trạng thái của thẻ mượn (0: đã trả, 1:đang mượn, -1: quá hạn)

```
alter table borrow add trang_thai integer;
alter table borrow drop column trang_thai;
```

```
create function update_trang_thai() returns void as
```

\$\$

begin

update borrow set trang_thai = -1 where borrow_id in (select borrow_id from borrow
where date_deadline < current_date and date_return is null);

update borrow set trang_thai = 0 where borrow_id in (select borrow_id from borrow
where date_return is not null);

update borrow set trang_thai = 1 where borrow_id in (select borrow_id from borrow
where date_deadline > current_date and date_return is null);

end;

\$\$ language plpgsql;

select * from borrow

select update_trang_thai()

2. Nguyễn Thị Hà 20184086

***Câu 11. in ra những sv mượn quyển sách "Đời ngắn đừng ngủ dài"

--c1: câu truy vấn con

explain --4.43...6.61

SELECT s.student_id, s.last_name || ' ' || s.first_name fullname,
s.class_name

FROM students s, detailborrow d, borrow b

WHERE s.student_id = b.student_id

AND b.borrow_id = d.borrow_id

AND d.bookid = (SELECT bookid

FROM book

WHERE name = 'Đời ngắn đừng ngủ dài');

--c2:

Explain --2.50....4.95

select s.student_id, s.last_name || ' ' || s.first_name fullname, s.class_name

```

from detailborrow db, borrow br, book b, students s
where db.bookid=b.bookid and
db.borrow_id=br.borrow_id and
br.student_id=s.student_id and
b.name='Đời ngắn đừng ngủ dài'

```

Đánh giá : ở cách 1 thì sử dụng 2 câu truy vấn lồng nhau mỗi lần duyệt 1 bản ghi thì câu truy vấn con lại được gọi. Còn cách 2 kết nối nhiều bảng với nhau, nó sẽ chạy lần lượt nên chi phí để chạy câu truy vấn ở cách 2 sẽ thấp hơn cách 1.

Câu 12. in thông tin sách xb năm 2012 của nxb trẻ

```

SELECT b.*
FROM book b, nxb n
WHERE b.nxbid = n.nxbid
AND b.year = 2012
AND n.name = 'NXB Trẻ';

```

Câu 13. in ra danh sách những sv cần gia hạn thẻ trong năm nay (hết hạn năm 2021)

--c1

```

SELECT s.*,date_range,date_experate
FROM students s, readercard r
WHERE s.student_id = r.student_id
AND extract(year from r.date_experate) = 2021;

```

--c2:

```

select s.*,date_range,date_experate
from students s, readercard rc
where s.student_id=rc.student_id and
date_part('year',date_experate)=2021;

```

Câu 14. in ra sách đang được mượn

```
SELECT b.*  
FROM book b, borrow br, detailborrow d  
WHERE b.bookid = d.bookid  
AND d.borrow_id = br.borrow_id  
AND br.date_return is null;
```

***Câu 15. thể loại nào đc ưa chuộng nhất

--c1

Explain --7.86...8.22

```
WITH tmp_cate AS(  
SELECT c.name, COUNT(c.name) count  
FROM category c, book bk, detailborrow d  
WHERE c.cateid = bk.cateid  
AND d.bookid = bk.bookid  
GROUP BY c.name)  
SELECT name, count  
FROM tmp_cate  
WHERE count = (SELECT max(count)  
FROM tmp_cate);
```

--c2:

Explain --13.23....15.53

```
select ca.name, count(db.bookid)  
from category ca join book b using (cateid)  
join detailborrow db using (bookid)  
group by (ca.cateid)  
having count(db.bookid)>= all (select count(db.bookid)  
from book b join detailborrow db using (bookid)  
group by (cateid));
```

Đánh giá: Cách 1 sử dụng hàm WITH... AS coi câu truy vấn như là một bảng tạm, giúp đơn giản hóa câu truy vấn hơn. Còn với cách 2 sử dụng câu truy vấn lồng nhau

*nó sẽ duyệt tất cả các bản ghi trong truy vấn con sao cho thỏa mãn câu truy vấn chính.
Do đó, mệnh đề with sử dụng ở cách 1 tối hơn*

Câu 16. sv mượn sách thường xuyên nhất

```
SELECT s.*, COUNT(d.borrow_id)
FROM students s JOIN borrow b USING(student_id)
JOIN detailborrow d USING(borrow_id)
GROUP BY (s.student_id)
HAVING COUNT(d.borrow_id) >= ALL(SELECT COUNT(d.borrow_id)
FROM students s JOIN borrow b USING(student_id)
JOIN detailborrow d USING(borrow_id)
GROUP BY (s.student_id));
```

Câu 17. lớp có số sv mượn là nhiều nhất và số sv mượn

```
SELECT s.class_name, COUNT(student_id) count_of_s_borrowed
FROM students s JOIN borrow b USING(student_id)
JOIN detailborrow d USING(borrow_id)
GROUP BY s.class_name
HAVING COUNT(student_id) >= ALL(SELECT COUNT(student_id)
FROM students s JOIN borrow b USING(student_id)
JOIN detailborrow d USING(borrow_id)
GROUP BY s.class_name);
```

Câu 18. Thống kê thông tin những sv thường xuyên trong tháng 11/2020 của thư viện

```
CREATE VIEW regular_students_view AS(
    SELECT s.first_name, s.last_name, COUNT(borrow_id)
FROM students s JOIN borrow USING (student_id)
WHERE date_borrow BETWEEN '2020-11-01' AND '2020-11-30')
```



```
GROUP BY (s.student_id))
```

```
SELECT * FROM regular_students_view;
```

Câu 19. thống kê lần mượn của mỗi sách của mỗi thể loại, sx giảm dần

```
SELECT c.name category_name, b.name book, COUNT (borrow_id)
FROM category c JOIN book b USING (cateid)
JOIN detailborrow USING (bookid)
GROUP BY(c.cateid,b.bookid)
ORDER BY c.cateid, COUNT(borrow_id) DESC
```

Câu 20. kiểm tra độc giả mượn sách (nếu còn sách quá hạn chưa trả thì k insert phiếu mượn)

```
CREATE TRIGGER t_insert
BEFORE INSERT ON borrow
FOR EACH ROW
EXECUTE PROCEDURE tg_t_insert();
```

```
CREATE OR REPLACE FUNCTION tg_t_insert()
RETURNS TRIGGER AS
$$
DECLARE x int;
BEGIN
SELECT student_id INTO x
FROM borrow
WHERE student_id = NEW.student_id AND date_return is null
AND date_deadline < current_date;

IF(x is not null) THEN RETURN null;
END IF;
```

```

IF(x is null) THEN RETURN NEW;
END IF;
END;
$$ LANGUAGE plpgsql;

```

```

DROP TRIGGER t_insert ON borrow
DROP FUNCTION tg_t_insert
SELECT * FROM borrow
INSERT INTO borrow VALUES (72,20184279,'offline','2020-01-16','2020-03-16')

```

```
delete from borrow where borrow_id=72
```

3. Thiều Ngọc Hiền Kiệt 20184128

Câu 21 sinh viên mượn sách ngày ... và cuốn sách mượn hôm do

```

select s.*,b.name
from students s, borrow br, detailborrow db, book b
where s.student_id=br.student_id and
br.borrow_id=db.borrow_id and
db.bookid=b.bookid and
date_borrow = '2020-10-30';

```

Câu 22 cuốn sách được mượn trong thời gian ngắn nhất

```

select b.*,age(date_return,date_borrow)
from book b, borrow br, detailborrow db
where b.bookid=db.bookid and db.borrow_id=br.borrow_id and
age(date_return,date_borrow) <=all (select age(date_return,date_borrow)
from borrow
where age(date_return,date_borrow) is not null);

```

Câu 23. sv có ngày hôm nay là ngày cuối cùng phải trả

```
select s.*  
from students s, borrow br  
where s.student_id=br.student_id and  
date_deadline= current_date
```

Câu 24 query sách có số lượng mượn nhiều nhất

```
select b.*, count(borrow_id)  
from book b join detailborrow br using (bookid)  
group by (b.bookid)  
having count(borrow_id) >= all (select count(borrow_id)  
from detailborrow  
group by(bookid));
```

Câu 25. số lượng sách nhiều nhất mà 1 sv mượn

```
select s.first_name,s.last_name, sum(quantity)  
from students s join borrow using (student_id)  
join detailborrow db using (borrow_id)  
group by (s.student_id)  
having sum(quantity) >= all ( select sum(quantity)  
from borrow join detailborrow using(borrow_id)  
group by(student_id));
```

Câu 26 sv ít mượn sách nhất

```
select s.first_name, s.last_name, count (borrow_id)
```

```

from students s left join borrow using (student_id)
group by (s.student_id)
having count(borrow_id) <= all ( select count(borrow_id)
from students s left join borrow using (student_id)
group by (s.student_id));

```

Câu 27 số sách đc mượn của sinh viên lớp ".."

```

select class_name, sum(quantity)
from students s join borrow using( student_id)
join detailborrow using(borrow_id)
where class_name='Dệt02-K62'
group by( class_name)
order by sum(quantity);

```

Câu 28 thống kê lần mượn của mỗi sách của mỗi nxb

```

select nxb.name, b.name, count (borrow_id)
from nxb join book b using (nxbid)
join detailborrow using (bookid)
group by(nxb.nxbid,b.bookid)
order by nxb.nxbid ,count(borrow_id) desc

```

Câu 29 in ra tháng có số sách đc mượn nhiều nhất trong năm 2019

```

select date_part('month',date_borrow) as monthx,sum(quantity)
from borrow join detailborrow using( borrow_id)
where date_part('year',date_borrow)=2019
group by (date_part('month',date_borrow))
having sum(quantity)>= all (select sum(quantity)
from borrow join detailborrow using( borrow_id)
where date_part('year',date_borrow)=2019
group by (date_part('month',date_borrow)));

```

Câu 30 tạo lời nhắc nhở cho sinh viên mượn sách đến hạn chưa trả !

create or replace function notice_student(in mssv char,out tag text) as

\$\$

begin

if ((select borrow_id from borrow

where student_id=mssv and date_deadline < current_date and date_return is null)
is not null)

then tag := 'Cảnh báo :SV có sách hết hạn chưa trả';

end if;

if ((select borrow_id from borrow

where student_id=mssv and date_return is not null) is not null)

then tag:='SV đã trả hết sách';

end if;

if ((select borrow_id from borrow

where student_id=mssv and date_deadline > current_date and date_return is null)
is not null)

then tag:='SV có sách đang mượn';

end if;

end;

\$\$ language plpgsql;

select notice_student('20200027');

Câu 31 (2trigger) sinh viên mượn sách nếu chưa có thể đọc thì chưa thể mượn !

create trigger n_insert

before insert on borrow

for each row

execute procedure rd_t_insert();

create or replace function rd_t_insert() returns trigger as

\$\$

declare x varchar;

begin

select student_id into x

from readercard rd

where student_id=new.student_id;

if(x is not null) then return new;

end if;

if(x is null) then return null;

end if;

end;

\$\$ language plpgsql;

drop trigger n_insert on borrow

drop trigger rd_insert on borrow

drop trigger rd1_insert on borrow

drop trigger new_insert on borrow

drop function rd_t_insert

drop function tg_af_insert

drop trigger af_insert on borrow

insert into borrow values (72,20184112,'online','2020-01-16','2020-03-16');

Câu 32 danh sách sách đc trả vào đúng ngày hết hạn mượn sách -- back up

select b.name,date_borrow,date_return

from borrow br, detailborrow db, book b

```
where br.borrow_id=db.borrow_id and  
      db.bookid = b.bookid and  
date_deadline=date_return
```

Tài liệu tham khảo :

1. Bài giảng Cơ sở dữ liệu, Thầy Nguyễn Hồng Phương, Viện Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội
2. Bài giảng Nhập môn cơ sở dữ liệu, Cô Nguyễn Thị Oanh, Viện Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội