

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**Khoa Khoa học - Kỹ thuật Máy tính**



**BÁO CÁO BÀI TẬP LỚN SỐ 2**  
**HỆ CƠ SỞ DỮ LIỆU**

---

**ĐỀ TÀI: : E-BOOKSTORE**

---

GVHD: Võ Thị Ngọc Châu

SV thực hiện: Tô Duy Hưng – 1810198  
Trương Thanh Lộc – 1812982  
Ngô Nguyễn Quốc Thịnh – 1814168

**TP. HỒ CHÍ MINH, THÁNG 12/2020**



## Mục lục

<b>1</b>	<b>Viết các Functions/Procedures/Assertions/Triggers</b>	<b>2</b>
1.1	Procedures . . . . .	2
1.2	Triggers . . . . .	22
<b>2</b>	<b>Định nghĩa và phân quyền cho các nhóm người dùng</b>	<b>23</b>
2.1	Định nghĩa nhóm người dùng . . . . .	23
2.2	Phân quyền nhóm người dùng . . . . .	24
2.3	Phân quyền mức ứng dụng . . . . .	25
<b>3</b>	<b>Kiến trúc mô hình ứng dụng</b>	<b>25</b>
3.1	Ứng với đề tài E-Book-Store . . . . .	26
<b>4</b>	<b>Thiết kế chi tiết cho ứng dụng</b>	<b>28</b>
4.1	Use Case diagram . . . . .	28
4.2	Class diagram . . . . .	29
4.3	Module interface . . . . .	29
4.4	Design pattern . . . . .	29
4.5	Working demonstration . . . . .	30
<b>5</b>	<b>Đánh giá các cơ sở dữ liệu</b>	<b>32</b>
5.1	Đánh giá về mặt ngữ nghĩa . . . . .	32
5.2	Đánh giá về dư thừa dữ liệu và các cập nhật dị thường . . . . .	33
5.3	Đánh giá về các giá trị không xác định (NULL) trong cơ sở dữ liệu . . . . .	33
5.4	Đánh giá về việc sinh ra các bộ giả . . . . .	33
5.5	Đánh giá dạng chuẩn hóa dữ liệu . . . . .	34
<b>6</b>	<b>So sánh dữ liệu trong cơ sở dữ liệu và dữ liệu trong các tập tin khi phát triển ứng dụng</b>	<b>34</b>
<b>7</b>	<b>Cài đặt chỉ mục</b>	<b>37</b>
7.1	Lý thuyết . . . . .	37
7.2	Tạo bảng dữ liệu demo . . . . .	37
7.3	Do đặc và đánh giá . . . . .	38
7.4	Kết luận . . . . .	39
<b>8</b>	<b>Video demo</b>	<b>39</b>
<b>9</b>	<b>Phân chia công việc</b>	<b>40</b>

# 1 Viết các Functions/Procedures/Assertions/Triggers

## 1.1 Procedures

### 1. Các quyền của nhân viên của hiệu sách

#### (a) Cập nhật thông tin về sách khi sách được nhập kho

drop procedure if exists NhapKho;

DELIMITER \$\$

CREATE procedure NhapKho (MaKho\_input varchar(20), ISBN\_input varchar(10), SoLuongBanIn\_input int)

BEGIN

if ISBN\_input in (select ISBN from BanIn) and MaKho\_input in (select MaKho from Kho) and SoLuongBanIn\_input >= 0 then

if exists (select ISBN from LuuTru where ISBN = ISBN\_input and MaKho = MaKho\_input) then

update LuuTru

set SoLuongBanIn = SoLuongBanIn + SoLuongBanIn\_input

where ISBN = ISBN\_input and LuuTru.Makho = MaKho\_input;

else

insert into LuuTru

values (MaKho\_input,ISBN\_input,SoLuongBanIn\_input);

end if;

end if;

end if;

END\$\$

DELIMITER ;

Kiểm tra:

call NhapKho('Kho1',1111111111,2);

select\* from LuuTru;

```
371 •  call NhapKho('Kho1',1111111111,2);
372 •  select* from LuuTru;
373
374 •  call NhapKho('Kho1',1111111111,5);
375 •  select* from LuuTru;
376 •  grant execute on procedure NhapKho to 'staff'@'localhost';
377
```

Makho	ISBN	SoLuongBanIn
Kho1	1111111111	2
NULL	NULL	NULL

Kết quả sau khi nhập vào "Kho1" 2 cuốn sách "1111111111"

call NhapKho('Kho1',1111111111,5);

select\* from LuuTru;

```
371 •  call NhapKho('Kho1',1111111111,2);
372 •  select* from LuuTru;
373
374 •  call NhapKho('Kho1',1111111111,5);
375 •  select* from LuuTru;
376 •  grant execute on procedure NhapKho to 'staff'@'localhost';
377
```

Makho	ISBN	SoLuongBanIn
Kho1	1111111111	7
NULL	NULL	NULL

Kết quả sau nhập vào "Kho1" tiếp 5 cuốn sách "1111111111"

**Phân quyền cho staff:**

```
grant execute on procedure NhapKho to 'staff'@'localhost';
```

(b) **Cập nhật thông tin về sách khi sách được xuất kho**

```
drop procedure if exists XuatKho;
```

```
DELIMITER $$
```

```
CREATE procedure XuatKho (MaKho_input varchar(20), ISBN_input varchar(10), SoLuongBanIn_input int)
```

```
BEGIN
```

```
if exists(select ISBN from LuuTru where ISBN = ISBN_input and MaKho = MaKho_input)  
and SoLuongBanIn_input <= (select SoLuongBanIn from LuuTru where ISBN = ISBN_input  
and MaKho = MaKho_input) then
```

```
if SoLuongBanIn_input < (select SoLuongBanIn from LuuTru where ISBN = ISBN_input  
and MaKho = MaKho_input) then
```

```
update LuuTru
```

```
set SoLuongBanIn = SoLuongBanIn - SoLuongBanIn_input
```

```
where ISBN = ISBN_input and Makho = MaKho_input;
```

```
else
```

```
delete from LuuTru
```

```
where ISBN = ISBN_input and Makho = MaKho_input;
```

```
end if;
```

```
end if;
```

```
END$$
```

```
DELIMITER ;
```

**Kiểm tra:**

```
call XuatKho('Kho1',1111111111,3);
```

```
select* from LuuTru;
```

```
399 • call XuatKho('Kho1',1111111111,3);  
400 • select* from LuuTru;  
401
```

	Makho	ISBN	SoLuongBanIn
▶	Kho1	1111111111	4
*	NULL	NULL	NULL

```
call XuatKho('Kho1',1111111111,4);
```

```
select* from LuuTru;
```

```
403 • call XuatKho('Kho1',1111111111,4);  
404 • select* from LuuTru;
```

	Makho	ISBN	SoLuongBanIn
*	NULL	NULL	NULL

**Phân quyền cho staff:**

```
grant execute on procedure XuatKho to 'staff'@'localhost';
```

(c) **Cập nhật thông tin giao dịch khi giao dịch trực tuyến gặp sự cố**

```
drop procedure if exists CapNhatGiaoDichLoi;
```

```
DELIMITER $$
```

```
CREATE procedure CapNhatGiaoDichLoi (UID_input varchar(10), ISBN_input varchar(10),  
NgayGio_input datetime)
```

```
BEGIN
```

```
if exists(select * from GiaoDich where UID = UID_input and ISBN = ISBN_input and NgayGio = NgayGio_input) then
    update ThanhToan
    set TrangThai = 'lỗi'
    where HoaDon = (select HoaDon from GiaoDich where UID = UID_input and ISBN = ISBN_input and NgayGio = NgayGio_input);
end if;
END$$
DELIMITER ;
```

**Kiểm tra:**

```
call CapNhatGiaoDichLoi('2222222','2222222222','2020-12-01');
select * from ThanhToan;
```

```
421 •  call CapNhatGiaoDichLoi('2222222','2222222222','2020-12-01');
422 •  select * from ThanhToan;
423 •  grant execute on procedure CapNhatGiaoDichLoi to 'staff'@'localhost';
424

| Result Grid | Filter Rows: | Edit: | Export/Import: | W |



| HoaDon | TrangThai       |
|--------|-----------------|
| ABC123 | chưa thanh toán |
| TDH166 | lỗi             |
| XYZ456 | lỗi             |


```

#### Phân quyền cho staff:

```
grant execute on procedure CapNhatGiaoDichLoi to 'staff'@'localhost';
```

- (d)
- Xem tất cả các sách tính theo ISBN được mua trong một ngày
  - Xem tổng số sách tính theo mỗi ISBN được mua trong một ngày
  - Xem tổng số sách truyền thông tính theo mỗi ISBN được mua trong một ngày
  - Xem tổng số sách điện tử được mua trong một ngày
  - Xem tổng số sách điện tử được thuê trong một ngày

```
drop table if exists DSMuaTrongMotNgay;
```

```
create table DSMuaTrongMotNgay(
```

```
    ISBN          varchar(10),
    TenSach       varchar(30),
    GiaTien       int not null,
    TenNhaXuatBan varchar(30) not null,
    primary key (ISBN),
    check (GiaTien > 0)
);
```

```
drop table if exists TongMuaTrongMotNgay;
```

```
create table TongMuaTrongMotNgay(
```

```
    ISBN          varchar(10),
    TenSach       varchar(30),
    GiaTien       int not null,
    TenNhaXuatBan varchar(30) not null,
    SoLuong       int,
    primary key (ISBN),
    check (GiaTien > 0 and SoLuong > 0)
);
```

```
drop table if exists TongTruyenThongMuaTrongMotNgay;
```

```
create table TongTruyenThongMuaTrongMotNgay(
```

```
    ISBN          varchar(10),
```



```
TenSach      varchar(30),
GiaTien       int not null,
TenNhaXuatBan  varchar(30) not null,
SLTruyenThong   int,
primary key (ISBN),
check (GiaTien > 0 and SLTruyenThong > 0)
);
drop table if exists TongDienTuMuaTrongMotNgay;
create table TongDienTuMuaTrongMotNgay(
    ISBN      varchar(10),
    TenSach    varchar(30),
    GiaTien     int not null,
    TenNhaXuatBan  varchar(30) not null,
    SLDienTuMua    int,
primary key (ISBN),
check (GiaTien > 0 and SLDienTuMua > 0)
);
drop table if exists TongDienTuThueTrongMotNgay;
create table TongDienTuThueTrongMotNgay(
    ISBN      varchar(10),
    TenSach    varchar(30),
    GiaTien     int not null,
    TenNhaXuatBan  varchar(30) not null,
    SLDienTuThue    int,
primary key (ISBN),
check (GiaTien > 0 and SLDienTuThue > 0)
);
drop procedure if exists XemSachMuaTrongMotNgay;
DELIMITER $$
CREATE procedure XemSachMuaTrongMotNgay (Ngay_input date, opt int)
BEGIN
    delete from DSMuaTrongMotNgay;
    delete from TongMuaTrongMotNgay;
    delete from TongTruyenThongMuaTrongMotNgay;
    delete from TongDienTuMuaTrongMotNgay;
    delete from TongDienTuThueTrongMotNgay;
    if opt = 1 then
        insert into DSMuaTrongMotNgay
        select distinct ISBN, TenSach, GiaTien, TenNhaXuatBan from GiaoDich natural join Sach
        where NgayGio = Ngay_input and Loai = 'mua';
    elseif opt = 2 then
        insert into TongMuaTrongMotNgay
        select ISBN, TenSach, GiaTien, TenNhaXuatBan, count(*) from GiaoDich natural join Sach
        where NgayGio = Ngay_input and Loai = 'mua'
        group by ISBN;
    elseif opt = 3 then
        insert into TongTruyenThongMuaTrongMotNgay
        select ISBN, TenSach, GiaTien, TenNhaXuatBan, count(*) from GiaoDich natural join Sach
        where NgayGio = Ngay_input and Loai = 'mua' and LoaiSach = 'bản in'
        group by ISBN;
```

```
elseif opt = 4 then
    insert into TongDienTuMuaTrongMotNgay
        select ISBN, TenSach, GiaTien, TenNhaXuatBan, count(*) from GiaoDich natural join Sach
        where NgayGio = Ngay_input and Loai = 'mua' and LoaiSach = 'bản điện tử'
    group by ISBN;
else
    insert into TongDienTuThueTrongMotNgay
        select ISBN, TenSach, GiaTien, TenNhaXuatBan, count(*) from GiaoDich natural join Sach
        where NgayGio = Ngay_input and Loai = 'thuê' and LoaiSach = 'bản điện tử'
    group by ISBN;
end if;
END$$
DELIMITER ;
call XemSachMuaTrongMotNgay('2020-12-01', 1);
select* from DSMuaTrongMotNgay;
```

ISBN	TenSach	GiaTien	TenNhaXuatBan
1111111111	CSDL	50000	KimDong
2222222222	HD qua mon	100000	KimDong
*	HULL	NULL	NULL

```
call XemSachMuaTrongMotNgay('2020-12-01', 2);
select* from TongMuaTrongMotNgay;
```

ISBN	TenSach	GiaTien	TenNhaXuatBan	SoLuong
1111111111	CSDL	50000	KimDong	1
2222222222	HD qua mon	100000	KimDong	3
*	HULL	NULL	NULL	NULL

```
call XemSachMuaTrongMotNgay('2020-12-01', 3);
select* from TongTruyenThongMuaTrongMotNgay;
```

ISBN	TenSach	GiaTien	TenNhaXuatBan	SLTruyenThong
1111111111	CSDL	50000	KimDong	1
2222222222	HD qua mon	100000	KimDong	2
*	HULL	NULL	NULL	NULL

```
call XemSachMuaTrongMotNgay('2020-12-01', 4);
select* from TongDienTuMuaTrongMotNgay;
```

ISBN	TenSach	GiaTien	TenNhaXuatBan	SLDienTuMua
2222222222	HD qua mon	100000	KimDong	1
*	HULL	NULL	NULL	NULL

```
call XemSachMuaTrongMotNgay('2020-12-01', 5);
select* from TongDienTuThueTrongMotNgay;
```

ISBN	TenSach	GiaTien	TenNhaXuatBan	SLDienTuThue
1111111111	CSDL	50000	KimDong	2
2222222222	HD qua mon	100000	KimDong	1
*	NULL	NULL	NULL	NULL

### Phân quyền cho staff

```
grant execute on procedure XemSachMuaTrongMotNgay to 'staff'@'localhost';
grant select on DSMuaTrongMotNgay to 'staff'@'localhost';
grant select on TongMuaTrongMotNgay to 'staff'@'localhost';
grant select on TongTruyenThongMuaTrongMotNgay to 'staff'@'localhost';
grant select on TongDienTuMuaTrongMotNgay to 'staff'@'localhost';
grant select on TongDienTuThueTrongMotNgay to 'staff'@'localhost';
```

- (e) • **Xem danh sách tác giả có số sách được mua nhiều nhất trong một ngày.**  
• **Xem danh sách tác giả có số sách được mua nhiều nhất trong một tháng.**

```
drop table if exists TacGiaCoSachBanChayNhat;
```

```
create table TacGiaCoSachBanChayNhat(
```

```
    AID          varchar(10),
    Ho           varchar(30),
    Ten          varchar(10),
    Email        varchar(30),
    NamSinh      int,
    primary key (AID),
    check (AID like 'A%'),
    check (NamSinh > 0 AND NamSinh < 9999)
);
```

```
drop procedure if exists XemTacGiaCoSachBanChayNhat;
```

```
DELIMITER $$
```

```
CREATE procedure XemTacGiaCoSachBanChayNhat (Ngay_input date, opt int)
```

```
BEGIN
```

```
    delete from TacGiaCoSachBanChayNhat;
```

```
if opt = 1 then
```

```
    insert into TacGiaCoSachBanChayNhat
```

```
        select AID, Ho, Ten, Email, NamSinh from (GiaoDich natural join Viet) natural join
        TacGia
```

```
        where NgayGio = Ngay_input and Loai = 'mua'
```

```
        group by AID
```

```
        having count(*) = (select max(Soluong) from (select count(*) as Soluong from GiaoDich
        natural join Viet
```

```
            where NgayGio = Ngay_input and Loai = 'mua'
```

```
            group by AID) as alias);
```

```
else
```

```
    insert into TacGiaCoSachBanChayNhat
```

```
        select AID, Ho, Ten, Email, NamSinh from (GiaoDich natural join Viet) natural join
        TacGia
```

```
        where datediff(Ngay_input,NgayGio) <= 30 and datediff(Ngay_input,NgayGio) >= 0
        and Loai = 'mua'
```

```
group by AID
having count(*) = (select max(Soluong) from (select count(*) as Soluong from GiaoDich
natural join Viet
where datediff(Ngay_input,NgayGio) <= 30 and datediff(Ngay_input,NgayGio) >= 0
and Loai = 'mua'
group by AID) as alias);
end if;
END$$
DELIMITER ;
```

**Kiểm tra:**

- call XemTacGiaCoSachBanChayNhat('2020-12-31', 2);
- select\* from TacGiaCoSachBanChayNhat;

	AID	Ho	Ten	Email	NamSinh
▶	A987654321	Gosho	Aoyama	NULL	2000
*	HULL	HULL	HULL	HULL	HULL

**Phân quyền cho staff:**

```
grant execute on procedure XemTacGiaCoSachBanChayNhat to 'staff'@'localhost';
grant select on TacGiaCoSachBanChayNhat to 'staff'@'localhost';
```

**(f) Xem danh sách sách được mua nhiều nhất trong một tháng.**

```
drop table if exists sachmuanhieunhat;
create table sachmuanhieunhat
(
ISBN varchar(10),
TenSach varchar(30),
GiaTien int not null,
TenNhaXuatBan varchar(30),
primary key (ISBN),
check (GiaTien > 0)
);
drop procedure if exists xemsachmuanhieunhat;
DELIMITER $$
create procedure xemsachmuanhieunhat()
BEGIN
delete from sachmuanhieunhat;
insert into sachmuanhieunhat
select ISBN, TenSach, GiaTien, TenNhaXuatBan
from sach natural join giaoDich
where Loai = 'mua'
group by Loai
having count(Loai) = (select MAX(mycount)
from (select count(Loai) as mycount from giaoDich where Loai = 'mua' group by Loai) as T);
end $$
delimiter ;


Kiểm tra:



```
call xemsachmuanhieunhat();
select * from sachmuanhieunhat;
```


```

The screenshot shows a MySQL Workbench interface. At the top, there are two lines of SQL code:  
606 • call xemsachmuanhieunhat();  
607 • select \* from sachmuanhieunhat;A result grid is displayed below the code. The grid has four columns: ISBN, TenSach, GiaTien, and TenNhaXuatBan. There is one row of data:  
ISBN: 2222222222, TenSach: HD qua mon, GiaTien: 100000, TenNhaXuatBan: KimDong.

ISBN	TenSach	GiaTien	TenNhaXuatBan
2222222222	HD qua mon	100000	KimDong

**Phân quyền cho staff:**

```
grant execute on procedure xemsachmuanhieunhat to 'staff'@'localhost';
grant select on sachmuanhieunhat to 'staff'@'localhost';
```

- (g) Xem danh sách mua hàng được thanh toán bằng thẻ trong một ngày.

```
drop table if exists danhsachthanhtoanthe;
create table danhsachthanhtoanthe(
    hoadon varchar(20),
    primary key(hoadon)
);
drop procedure if exists xemdanhsachthanhtoanthe;
DELIMITER $$
create procedure xemdanhsachthanhtoanthe(date DATE)
BEGIN
    delete from danhsachthanhtoanthe;
    insert into danhsachthanhtoanthe
        select hoadon from thanhtoanthe natural join giao dich
        where NgayGio = date;
    end $$
DELIMITER ;
```

**Kiểm tra:**

```
call xemdanhsachthanhtoanthe('2020-12-01 00:00:00');
select * from danhsachthanhtoanthe;
```

The screenshot shows a MySQL Workbench interface. At the top, there are two lines of SQL code:  
628 • call xemdanhsachthanhtoanthe('2020-12-01 00:00:00');
629 • select \* from danhsachthanhtoanthe;A result grid is displayed below the code. The grid has one column: hoadon. There are three rows of data:  
hoadon: ABC123, XYZ456, NULL.

hoadon
ABC123
XYZ456
NULL

**Phân quyền cho staff:**

```
grant execute on procedure xemdanhsachthanhtoanthe to 'staff'@'localhost';
grant select on danhsachthanhtoanthe to 'staff'@'localhost';
```

- (h) Xem danh sách mua hàng được thanh toán bằng thẻ gấp sự cố trong một ngày.

```
drop table if exists danhsachsucothanhtoanthe;
create table danhsachsucothanhtoanthe(
    hoadon varchar(20),
    primary key(hoadon)
);
drop procedure if exists xemdanhsachsucothanhtoanthe;
DELIMITER $$
create procedure xemdanhsachsucothanhtoanthe(NgayGio_input DATE)
BEGIN
    delete from danhsachsucothanhtoanthe;
    insert into danhsachsucothanhtoanthe
        select hoadon from thanhtoanthe
        where NgayGio = NgayGio_input;
```

```
select hoaDon from (thanhtoanthe natural join thanhtoan) natural join giaoDich
where NgayGio = NgayGio_input and TrangThai = 'lỗi';
end $$

DELIMITER ;

Kiểm tra:
call xemdanhsachsucothanhtoanthe('2020-12-01');
select * from danhsachsucothanhtoanthe;
```

hoaDon	
XYZ456	
*	NULL

**Phân quyền cho staff:**

```
grant execute on procedure xemdanhsachsucothanhtoanthe to 'staff'@'localhost';
grant select on danhsachsucothanhtoanthe to 'staff'@'localhost';
```

- (i) Xem danh sách kho hàng có số sách tính theo mỗi ISBN dưới 10 quyển trong một ngày.

```
insert into Luutru values ('Kho1','1111111111',11);
drop table if exists isbn10;
create table isbn10(
makho varchar(20),
primary key (makho)
);
drop procedure if exists xemisbn10;
DELIMITER $$
create procedure xemisbn10(NgayGio_input DATE)
BEGIN
delete from isbn10;
insert into isbn10
select kho.MaKho
from ((kho natural join luutru) natural join banin) natural join giaoDich
where NgayGio = NgayGio_input
group by ISBN having count(*) < 10;
end $$

DELIMITER ;
Kiểm tra:
call xemisbn10('2020-12-01 00:00:00');
select * from isbn10;
```

makho	
Kho1	
*	NULL

**Phân quyền cho staff:**

```
grant execute on procedure xemisbn10 to 'staff'@'localhost';
grant select on isbn10 to 'staff'@'localhost';
```

- (j) Xem tổng số sách tính theo mỗi ISBN tại mỗi kho hàng trong một tháng.

```
drop table if exists khoisbnthang;
create table khoisbnthang(
    soluong decimal(10,2),
    primary key (soluong)
);
drop procedure if exists xemkhoisbnthang;
DELIMITER $$
create procedure xemkhoisbnthang(month INT, year INT)
BEGIN
    delete from khoisbnthang;
    insert into khoisbnthang
    select count(*)
    from kho join luutru l on kho.MaKho = l.MaKho join banin b on b.ISBN = l.ISBN join giao dich
    g on l.ISBN = g.ISBN
    where NgayGio like concat('%',year,'-',month,'%')
    group by g.ISBN;
end $$
DELIMITER ;
Kiểm tra:
call xemkhoisbnthang(12, 2020);
select * from khoisbnthang;
```

soluong	3.00	*
NULL		

**Phân quyền cho staff:**

```
grant execute on procedure xemkhoisbnthang to 'staff'@'localhost';
grant select on khoisbnthang to 'staff'@'localhost';
```

- (k) Xem danh sách kho hàng được xuất kho nhiều nhất trong một tháng.

```
drop table if exists danhsachkhohangxuatkhonhieu;
create table danhsachkhohangxuatkhonhieu(
    makho varchar(20),
    primary key (makho)
);
drop procedure if exists xemdanhsachkhohangxuatkhonhieu;
DELIMITER $$
create procedure xemdanhsachkhohangxuatkhonhieu(month int, year int)
BEGIN
    delete from danhsachkhohangxuatkhonhieu;
    insert into danhsachkhohangxuatkhonhieu
    select makho from xuatkho
    group by makho
    having count(makho) = (select MAX(mycount)
    from (select count(makho) as mycount from xuatkho where soluongxuatnhapkho > 0 group by
    makho) as T);
end $$
DELIMITER ;
```

```
update luutru
set Soluongbanin = 3
where makho = 'Kho1' and isbn = '1111111111';
```

**Kiểm tra:**

```
call xemdanhsachkhohangxuatkhonhieutrongthang(12,2020);
select * from danhsachkhohangxuatkhonhieu;
```

makho
Kho1
NULL

**Phân quyền cho staff:**

```
grant execute on procedure xemdanhsachkhohangxuatkhonhieutrongthang to 'staff'@'localhost';
grant select on danhsachkhohangxuatkhonhieu to 'staff'@'localhost';
```

## 2. Các quyền của khách hàng

(a) **Cập nhật thông tin cá nhân.**

```
drop procedure if exists CapNhatThongTinCaNhan;
```

```
DELIMITER $$
```

```
CREATE procedure CapNhatThongTinCaNhan (UID_input varchar(10), Ho_input varchar(20),
Ten_input varchar(10), DiaChi_input varchar(30),
Phone_input varchar(11))
```

```
BEGIN
```

```
update KhachHang
```

```
set Ho = Ho_input, Ten = Ten_input, DiaChi = DiaChi_input, Phone = Phone_input
where UID = UID_input;
```

```
END$$
```

```
DELIMITER ;
```

**Kiểm tra:**

```
call CapNhatThongTinCaNhan('4444444','kudo','shinichi','japan','0123456789');
select* from KhachHang;
```

UID	Ho	Ten	DiaChi	Phone
1111111	NULL	NULL	aaa	0123456789
1810198	to	hung	KTX	0905983795
2222222	NULL	NULL	aaa	0123456789
3333333	NULL	NULL	aaa	0123456789
4444444	kudo	shinichi	japan	0123456789
*	NULL	NULL	NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure CapNhatThongTinCaNhan to 'customer'@'localhost';
```

(b) **Cập nhật thông tin thanh toán.**

```
drop procedure if exists ChonTheTinDung;
```

```
DELIMITER $$
```

```
CREATE procedure ChonTheTinDung (HoaDon_input varchar(20), MaThe_input varchar(10))
BEGIN
```

```
if not exists (select * from TheTinDung where MaThe = MaThe_input) then
```

```

insert into TheTinDung(MaThe,HoaDon) values (MaThe_input, HoaDon_input);
else
    update TheTinDung
    set HoaDon = HoaDon_input
where MaThe = MaThe_input;
end if;
END$$
DELIMITER ;
Kiểm tra:
call ChonTheTinDung('XYZ456','XXX456789');
select* from TheTinDung;

```

	MaThe	TenChuThe	TenNhanHang	NgayHetHan	UID	HoaDon
▶	XXX123456	toduyhung	ocb	NULL	1810198	ABC123
	XXX456789	toduyhung	bidv	NULL	1810198	XYZ456
*	NULL	NULL	NULL	NULL	NULL	NULL

#### Phân quyền cho customer:

```
grant execute on procedure ChonTheTinDung to 'customer'@'localhost';
```

#### (c) Cập nhật giao dịch mua hàng

```
drop procedure if exists CapNhatGiaoDich;
```

```
DELIMITER $$
```

```
CREATE procedure CapNhatGiaoDich (UID_input varchar(10), ISBN_input varchar(10),
NgayGio_input datetime, Loai_input varchar(10),
LoaiSach_input varchar(20), SoLuong_input int)
```

```
BEGIN
```

```
if not exists (select * from GiaoDich where UID = UID_input and ISBN = ISBN_input and NgayGio = NgayGio_input) then
```

```
    insert into GiaoDich(UID, ISBN, NgayGio, Loai, LoaiSach, SoLuong) values
```

```
(UID_input, ISBN_input, NgayGio_input, Loai_input, LoaiSach_input, SoLuong_input);
```

```
else
```

```
    update GiaoDich
```

```
    set Loai = Loai_input, LoaiSach = LoaiSach_input, SoLuong = SoLuong_input
where UID = UID_input and ISBN = ISBN_input and NgayGio = NgayGio_input;
```

```
end if;
```

```
END$$
```

```
DELIMITER ;
```

#### Kiểm tra:

```
call CapNhatGiaoDich('1111111','2222222222','2020-12-01','mua','bản điện tử',2);
```

```
select* from GiaoDich;
```

```
call CapNhatGiaoDich('1111111','2222222222',curdate(),'mua','bản điện tử',2);
```

```
select* from GiaoDich;
```

```

795 •  call CapNhatGiaoDich('1111111','2222222222','2020-12-01','mua','bản điện tử',2);
796 •  select* from GiaoDich;
797

```

	UID	ISBN	NgayGio	Loai	LoaiSach	SoLuong	HoaDon
▶	1111111	2222222222	2020-12-01 00:00:00	mua	bản điện tử	2	NULL
	1810198	1111111111	2020-12-01 00:00:00	thuê	bản điện tử	2	XYZ456
	1810198	2222222222	2020-12-01 00:00:00	mua	bản in	1	ABC123
	2222222	2222222222	2020-12-01 00:00:00	mua	bản điện tử	1	TDH166

**Phân quyền cho customer:**

```
grant execute on procedure CapNhatGiaoDich to 'customer'@'localhost';
```

(d) **Xem danh sách sách theo thể loại**

```
drop table if exists SachCungTheLoai;
```

```
create table SachCungTheLoai(
```

```
    ISBN          varchar(10),
```

```
    TenSach       varchar(30),
```

```
    GiaTien       int not null,
```

```
    TenNhaXuatBan  varchar(30),
```

```
        primary key (ISBN),
```

```
        check (GiaTien > 0)
```

```
);
```

```
drop procedure if exists XemSachTheoTheLoai;
```

```
DELIMITER $$
```

```
CREATE procedure XemSachTheoTheLoai (LinhVuc_input varchar(20))
```

```
BEGIN
```

```
    delete from SachCungTheLoai;
```

```
    insert into SachCungTheLoai
```

```
        select ISBN, TenSach, GiaTien, TenNhaXuatBan from Sach natural join LinhVuc
```

```
        where ALinhVuc = LinhVuc_input;
```

```
END$$
```

```
DELIMITER ;
```

**Kiểm tra:**

```
call XemSachTheoTheLoai('CNTT');
```

```
select* from SachCungTheLoai;
```

	ISBN	TenSach	GiaTien	TenNhaXuatBan
▶	1111111111	CSDL	50000	KimDong
*	NULL	NULL	NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure XemSachTheoTheLoai to 'customer'@'localhost';
```

```
grant select on SachCungTheLoai to 'customer'@'localhost';
```

(e) **Xem danh sách sách theo tác giả.**

```
drop table if exists SachCungTacGia;
```

```
create table SachCungTacGia(
```

```
    ISBN          varchar(10),
```

```
    TenSach       varchar(30),
```

```
    GiaTien       int not null,
```

```
    TenNhaXuatBan  varchar(30),
```

```
        primary key (ISBN),
```

```
        check (GiaTien > 0)
```

```
);
```

```
drop procedure if exists XemSachTheoTacGia;
```

```
DELIMITER $$
```

```
CREATE procedure XemSachTheoTacGia (Ho_input varchar(30), Ten_input varchar(10))
```

```
BEGIN
```

```
    delete from SachCungTacGia;
```

```
insert into SachCungTacGia
select ISBN, TenSach, GiaTien, TenNhaXuatBan from (Sach natural join Viet) natural join Tac-
Gia
where Ho = Ho_input and Ten = Ten_input;
END$$
DELIMITER ;
Kiểm tra:
call XemSachTheoTacGia('Conan','Doyle');
select* from SachCungTacGia;
```

	ISBN	TenSach	GiaTien	TenNhaXuatBan
▶	1111111111	CSDL	50000	KimDong
	3333333333	VaTheLaHet	100000	HCM
	4444444444	5phLaDL	100000	HCM
*	NULL	NULL	NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure XemSachTheoTacGia to 'customer'@'localhost';
grant select on SachCungTacGia to 'customer'@'localhost';
```

(f) **Xem danh sách sách theo từ khóa**

```
drop table if exists SachCungTuKhoa;
create table SachCungTuKhoa(
    ISBN          varchar(10),
    TenSach       varchar(30),
    GiaTien        int not null,
    TenNhaXuatBan varchar(30),
    primary key (ISBN),
    check (GiaTien > 0)
);
drop procedure if exists XemSachTheoTuKhoa;
DELIMITER $$
CREATE procedure XemSachTheoTuKhoa (TuKhoa_input varchar(20))
BEGIN
    delete from SachCungTuKhoa;
    insert into SachCungTuKhoa
    select ISBN, TenSach, GiaTien, TenNhaXuatBan from Sach natural join TuKhoa
    where ATuKhoa = TuKhoa_input;
END$$
DELIMITER ;
Kiểm tra:
call XemSachTheoTuKhoa('nghiên cứu');
select* from SachCungTuKhoa;
```

	ISBN	TenSach	GiaTien	TenNhaXuatBan
▶	1111111111	NULL	NULL	NULL
*	NULL	NULL	NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure XemSachTheoTuKhoa to 'customer'@'localhost';
grant select on SachCungTuKhoa to 'customer'@'localhost';
```

(g) **Xem danh sách sách theo năm xuất bản**

```
drop table if exists SachCungNamXB;
```

```
create table SachCungNamXB(
```

```
    ISBN          varchar(10),
    TenSach       varchar(30),
    GiaTien        int not null,
    TenNhaXuatBan  varchar(30),
    primary key (ISBN),
    check (GiaTien > 0)
);
```

```
drop procedure if exists XemSachTheoNamXB;
```

```
DELIMITER $$
```

```
CREATE procedure XemSachTheoNamXB (NamXB_input int)
```

```
BEGIN
```

```
    delete from SachCungNamXB;
```

```
    insert into SachCungNamXB
```

```
        select ISBN, TenSach, GiaTien, TenNhaXuatBan from Sach natural join NamSanXuat
        where ANamSanXuat = NamXB_input;
```

```
END$$
```

```
DELIMITER ;
```

**Kiểm tra:**

```
call XemSachTheoNamXB(2019);
```

```
select* from SachCungNamXB;
```

	ISBN	TenSach	GiaTien	TenNhaXuatBan
▶	1111111111	CSDL	50000	KimDong
*	HULL	NULL	NULL	HULL

**Phân quyền cho customer:**

```
grant execute on procedure XemSachTheoNamXB to 'customer'@'localhost';
grant select on SachCungNamXB to 'customer'@'localhost';
```

(h) **Xem danh sách sách mà mình đã mua trong một tháng.**

```
drop table if exists MuaTrong1Thang;
```

```
create table MuaTrong1Thang(
```

```
    UID          varchar(10),
    ISBN         varchar(10),
```

```
    NgayGio      datetime,
```

```
    Loai          varchar(10) not null,
```

```
    check (Loai in ('mua','thuê')),
```

```
    LoaiSach      varchar(20) not null,
```

```
    check(LoaiSach in ('bản in','bản điện tử')),
```

```
    SoLuong      int default(1),
```

```
    check(SoLuong > 0),
```

```
    HoaDon       varchar(20),
```

```
    primary key (UID,ISBN,NgayGio)
```

```
);  
drop procedure if exists XemMuaTrong1Thang;  
DELIMITER $$  
CREATE procedure XemMuaTrong1Thang (UID_input varchar(10))  
BEGIN  
    delete from MuaTrong1Thang;  
    insert into MuaTrong1Thang  
    select * from GiaoDich  
    where UID = UID_input and Loai = 'mua' and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0;  
END$$  
DELIMITER ;  
Kiểm tra:  
call XemMuaTrong1Thang('1810198');  
select* from MuaTrong1Thang;
```

	UID	ISBN	NgayGio	Loai	LoaiSach	SoLuong	HoaDon
▶	1810198	2222222222	2020-12-01 00:00:00	mua	bán in	1	ABC 123
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### Phân quyền cho customer:

```
grant execute on procedure XemMuaTrong1Thang to 'customer'@'localhost';  
grant select on MuaTrong1Thang to 'customer'@'localhost';
```

- (i)    • Xem danh sách các giao dịch mà mình đã thực hiện trong một tháng  
      • Xem danh sách các giao dịch gấp sự cố mà mình đã thực hiện trong một tháng  
      • Xem danh sách các giao dịch mà mình đã thực hiện nhưng chưa hoàn tất

```
drop table if exists GiaoDichTrong1Thang;
```

```
create table GiaoDichTrong1Thang(
```

```
    UID          varchar(10),  
    ISBN         varchar(10),  
    NgayGio      datetime,  
    Loai          varchar(10) not null,  
    check (Loai in ('mua','thue')),  
    LoaiSach     varchar(20) not null,  
    check(LoaiSach in ('bản in','bản điện tử')),  
    SoLuong      int default(1),  
    check(SoLuong > 0),  
    HoaDon        varchar(20),  
    primary key (UID,ISBN,NgayGio)
```

```
);
```

```
drop procedure if exists XemGiaoDichTrong1Thang;
```

```
DELIMITER $$
```

```
CREATE procedure XemGiaoDichTrong1Thang (UID_input varchar(10), option_num int)  
BEGIN
```

```
    delete from GiaoDichTrong1Thang;
```

```
    if option_num = 1 then
```

```
        insert into GiaoDichTrong1Thang  
        select * from GiaoDich
```

```
where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0;
elseif option_num = 2 then
    insert into GiaoDichTrong1Thang
        select UID, ISBN, NgayGio, Loai, LoaiSach, SoLuong, HoaDon from GiaoDich natural join
        ThanhToan
        where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0
        and TrangThai = 'lỗi';
    else
        insert into GiaoDichTrong1Thang
            select UID, ISBN, NgayGio, Loai, HoaDon from GiaoDich natural join ThanhToan
            where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0
            and TrangThai = 'chưa thanh toán';
        end if;
END$$
DELIMITER ;
```

**Kiểm tra:**

```
call XemGiaoDichTrong1Thang('1810198', 2);
select* from GiaoDichTrong1Thang;
```

	UID	ISBN	NgayGio	Loai	LoaiSach	SoLuong	HoaDon
▶	1810198	1111111111	2020-12-01 00:00:00	thuê	bán điện tử	2	XYZ456
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

#### Phân quyền cho customer:

```
grant execute on procedure XemGiaoDichTrong1Thang to 'customer'@'localhost';
grant select on GiaoDichTrong1Thang to 'customer'@'localhost';
```

#### (j) Xem danh sách tác giả của cùng một thể loại.

```
drop table if exists TacGiaCungTheLoai;
create table TacGiaCungTheLoai(
    AID          varchar(10),
    Ho           varchar(30),
    Ten          varchar(10),
    Email        varchar(30),
    NamSinh      int,
    primary key (AID),
    check (AID like 'A%'),
    check (NamSinh > 0 AND NamSinh < 9999)
);
drop procedure if exists XemTacGiaTheoTheLoai;
DELIMITER $$
CREATE procedure XemTacGiaTheoTheLoai (LinhVuc_input varchar(20))
BEGIN
    delete from TacGiaCungTheLoai;
    insert into TacGiaCungTheLoai
        select distinct AID, Ho, Ten, Email, NamSinh from (TacGia natural join Viet) natural join LinhVuc
```

where ALinhVuc = LinhVuc\_input;

END\$\$

DELIMITER ;

**Kiểm tra:**

call XemTacGiaTheoTheLoai('CNTT');

select\* from TacGiaCungTheLoai;

```
1040 •  call XemTacGiaTheoTheLoai('CNTT');
1041 •  select* from TacGiaCungTheLoai;
<
| Result Grid | Filter Rows: [ ] | Edit
+-----+-----+-----+-----+-----+
| AID | Ho | Ten | Email | NamSinh |
+-----+-----+-----+-----+-----+
| A123456789 | Conan | Doyle | NULL | 1999 |
| * | NULL | NULL | NULL | NULL |

```

**Phân quyền cho customer:**

grant execute on procedure XemTacGiaTheoTheLoai to 'customer'@'localhost';

grant select on TacGiaCungTheLoai to 'customer'@'localhost';

- (k) **Xem danh sách tác giả của cùng một số từ khóa.** -> trên CSDL sẽ xem của cùng 1 từ khóa, trên ứng dụng sẽ tách cụm từ khóa ra để kiểm tra cho từng từ khóa rồi gộp lại danh sách tác giả sau

drop table if exists TacGiaCungTuKhoa;

create table TacGiaCungTuKhoa(

    AID                  varchar(10),

    Ho                  varchar(30),

    Ten                  varchar(10),

    Email               varchar(30),

    NamSinh             int,

    primary key (AID),

    check (AID like 'A%'),

    check (NamSinh > 0 AND NamSinh < 9999)

);

drop procedure if exists XemTacGiaTheoTuKhoa;

DELIMITER \$\$

CREATE procedure XemTacGiaTheoTuKhoa (TuKhoa\_input varchar(20))

BEGIN

    delete from TacGiaCungTuKhoa;

    insert into TacGiaCungTuKhoa

        select distinct AID, Ho, Ten, Email, NamSinh from (TacGia natural join Viet) natural join TuKhoa

        where ATuKhoa = TuKhoa\_input;

END\$\$

DELIMITER ;

**Kiểm tra:**

call XemTacGiaTheoTuKhoa('nghiên cứu');

select\* from TacGiaCungTuKhoa;

```
1071 •  call XemTacGiaTheoTuKhoa('nghiên cứu');
1072 •  select* from TacGiaCungTuKhoa;
<
| Result Grid | Filter Rows: [ ] | Edit: [ ]
+-----+-----+-----+-----+-----+
| AID | Ho | Ten | Email | NamSinh |
+-----+-----+-----+-----+-----+
| A123456789 | Conan | Doyle | NULL | 1999 |
| * | NULL | NULL | NULL | NULL |

```

**Phân quyền cho customer:**

```
grant execute on procedure XemTacGiaTheoTuKhoa to 'customer'@'localhost';
grant select on TacGiaCungTuKhoa to 'customer'@'localhost';
```

- (l) **Xem tổng số sách theo từng thể loại mà mình đã mua trong một tháng.**

```
drop table if exists TongSoSachTheoTheLoai;
```

```
create table TongSoSachTheoTheLoai(
```

```
    LinhVuc          varchar(20) primary key,
```

```
    TongSoLuong      int
```

```
);
```

```
drop procedure if exists XemTongSoSachTheoTheLoai;
```

```
DELIMITER $$
```

```
CREATE procedure XemTongSoSachTheoTheLoai (UID_input varchar(10), LinhVuc_input var-
char(20))
```

```
BEGIN
```

```
    delete from TongSoSachTheoTheLoai;
```

```
    if LinhVuc_input = " then
```

```
        insert into TongSoSachTheoTheLoai
```

```
            select ALinhVuc, count(*) from GiaoDich natural join LinhVuc
```

```
            where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and dated-
iff(curdate(),NgayGio) >= 0
```

```
            group by ALinhVuc;
```

```
        else
```

```
            insert into TongSoSachTheoTheLoai
```

```
                select UID, count(*) from GiaoDich natural join LinhVuc
```

```
                where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and dated-
iff(curdate(),NgayGio) >= 0
```

```
                and ALinhVuc = LinhVuc_input;
```

```
        end if;
```

```
END$$
```

```
DELIMITER ;
```

**Kiểm tra:**

```
call XemTongSoSachTheoTheLoai('1810198', 'KHMT');
```

```
select* from TongSoSachTheoTheLoai;
```

LinhVuc	TongSoLuong
1810198	2
NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure XemTongSoSachTheoTheLoai to 'customer'@'localhost';
grant select on TongSoSachTheoTheLoai to 'customer'@'localhost';
```

- (m) **Xem các giao dịch mà mình đã thực hiện có số lượng sách được mua nhiều nhất trong một tháng.**

```
drop table if exists GDMuaNhiuNhatTrong1Thang;
```

```
create table GDMuaNhiuNhatTrong1Thang(
```

```
    UID          varchar(10),
```

```
    ISBN         varchar(10),
```

```
    NgayGio       datetime,
```

```
    Loai          varchar(10) not null,
```



```
check (Loai in ('mua','thuê')),
LoaiSach      varchar(20) not null,
check (LoaiSach in ('bản điện tử','bản in')),
SoLuong      int default(1),
check (SoLuong > 0),
HoaDon      varchar(20),
primary key (UID,ISBN,NgayGio)
);
drop procedure if exists XemGDMuaNhieuNhatTrong1Thang;
DELIMITER $$

CREATE procedure XemGDMuaNhieuNhatTrong1Thang (UID_input varchar(10))
BEGIN
    delete from GDMuaNhieuNhatTrong1Thang;
    insert into GDMuaNhieuNhatTrong1Thang
        select * from GiaoDich
        where UID = UID_input and Loai = 'mua' and datediff(curdate(),NgayGio) <= 30 and
        datediff(curdate(),NgayGio) >= 0
        and SoLuong = (select max(Soluong) from GiaoDich
        where UID = UID_input and Loai = 'mua' and datediff(curdate(),NgayGio) <= 30 and
        datediff(curdate(),NgayGio) >= 0);
END$$
DELIMITER ;

Kiểm tra:
call XemGDMuaNhieuNhatTrong1Thang('4444444');
select* from GDMuaNhieuNhatTrong1Thang;

Phân quyền cho customer:
grant execute on procedure XemGDMuaNhieuNhatTrong1Thang to 'customer'@'localhost';
grant select on GDMuaNhieuNhatTrong1Thang to 'customer'@'localhost';

(n) Xem các giao dịch vừa có sách truyền thống vừa có sách điện tử được mua hoặc thuê mà mình đã thực hiện trong một tháng.
drop table if exists GD2LoaiTrong1Thang;
create table GD2LoaiTrong1Thang(
    UID      varchar(10),
    ISBN     varchar(10),
    NgayGio   datetime,
    Loai      varchar(10) not null,
    check (Loai in ('mua','thuê')),
    LoaiSach    varchar(20) not null,
    check (LoaiSach in ('bản điện tử','bản in')),
    SoLuong    int default(1),
    check (SoLuong > 0),
    HoaDon    varchar(20),
    primary key (UID,ISBN,NgayGio)
);
drop procedure if exists XemGD2LoaiTrong1Thang;
DELIMITER $$

CREATE procedure XemGD2LoaiTrong1Thang (UID_input varchar(10))
BEGIN
    delete from GD2LoaiTrong1Thang;
    insert into GD2LoaiTrong1Thang
        select * from GiaoDich
```

```
where UID = UID_input and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0
and ISBN in (select distinct ISBN from GiaoDich
    where UID = UID_input and LoaiSach = 'bản in' and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0
    and ISBN in
        (select ISBN from GiaoDich
            where UID = UID_input and LoaiSach = 'bản điện tử' and datediff(curdate(),NgayGio) <= 30 and datediff(curdate(),NgayGio) >= 0));
END$$
DELIMITER ;
```

**Kiểm tra:**

```
call XemGD2LoaiTrong1Thang('4444444');
select* from GD2LoaiTrong1Thang;
```

	UID	ISBN	NgayGio	Loai	LoaiSach	Soluong	HoaDon
▶	4444444	2222222222	2020-12-01 00:00:00	mua	bản in	1	NULL
▶	4444444	2222222222	2020-12-02 00:00:00	thuê	bản điện tử	1	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Phân quyền cho customer:**

```
grant execute on procedure XemGD2LoaiTrong1Thang to 'customer'@'localhost';
grant select on GD2LoaiTrong1Thang to 'customer'@'localhost';
```

## 1.2 Triggers

- Trong các giao dịch, thuê chỉ được áp dụng đối với sách điện tử. Ngoài ra, nếu trong một giao dịch không có sẵn bản in hay bản điện tử trong cơ sở dữ liệu thì giao dịch này là không hợp lệ.  
drop trigger if exists LoaiSachGiaoDich ;
delimiter \$\$

create trigger LoaiSachGiaoDich after insert on GiaoDich
for each row
begin
if new.Loai = 'thuê' and new.LoaiSach = 'bản in' then
 asignal sqlstate '45000' set message\_text = 'Thuê chỉ được áp dụng cho sách điện tử';
end if;
if new.LoaiSach = 'bản in' and not exists(select \* from BanIn where ISBN = new.ISBN) then
 signal sqlstate '45000' set message\_text = 'Không có bản in';
elseif new.LoaiSach = 'bản điện tử' and not exists(select\* from BanDienTu where ISBN = new.ISBN)
then
 signal sqlstate '45000' set message\_text = 'Không có bản điện tử';
end if;
end\$\$
delimiter ;

- Một hóa đơn sau khi được thanh toán bằng thẻ tín dụng thành công thì sẽ bị xóa đi khỏi bảng TheTimDung.  
drop trigger if exists ThanhToanBangTheTD ;
delimiter \$\$



```
create trigger ThanhToanBangTheTD after update on ThanhToan
```

```
for each row
```

```
begin
```

```
if new.TrangThai = 'thành công' then
```

```
    delete from TheTinDung
```

```
    where HoaDon = old.HoaDon;
```

```
end if;
```

```
end$$
```

```
delimiter ;
```

- Nếu như số lượng bản in trong kho giảm đi (đồng nghĩa với việc bản in của cuốn sách đó đã bị xuất kho) thì cơ sở dữ liệu sẽ tự động cập nhật isbn của cuốn sách đó và số lượng sách xuất kho vào 1 bảng phụ tên là xuatkho. Bảng này sẽ giúp ta lưu trữ và truy xuất những thông tin liên quan đến quá trình xuất kho như: Xem danh sách kho hàng xuất nhiều nhất trong 1 tháng,...

```
drop trigger if exists danhsachkhohangxuatkho;
```

```
drop table if exists xuatkho;
```

```
create table xuatkho(
```

```
    makho varchar(20),
```

```
    soluongxuatnhapkho int
```

```
);
```

```
delimiter $$
```

```
create trigger danhsachkhohangxuatkho after update on luutru
```

```
for each row
```

```
begin
```

```
if new.SoLuongBanIn < old.SoLuongBanIn then
```

```
    insert into xuatkho values (new.MaKho, old.SoLuongBanIn-new.SoLuongBanIn);
```

```
end if;
```

```
end$$
```

```
delimiter ;
```

## 2 Định nghĩa và phân quyền cho các nhóm người dùng

### 2.1 Định nghĩa nhóm người dùng

Tiếp đó, bằng cách sử dụng lệnh: **CREATE USER '<user\_name>'@'localhost' IDENTIFIED BY <password>**; ta có thể tạo ra 2 user có tên là staff và customer - tượng trưng cho 2 nhóm người dùng lần lượt là Nhân viên và Khách Hàng.



```
mysql> create user 'staff'@'localhost' identified by 'mypass';
Query OK, 0 rows affected (0.03 sec)

mysql> select host,user from mysql.user;
+-----+-----+
| host | user |
+-----+-----+
| %   | root  |
| localhost | mysql.infoschema |
| localhost | mysql.session |
| localhost | mysql.sys |
| localhost | root  |
| localhost | staff |
+-----+-----+
6 rows in set (0.00 sec)

mysql> create user 'customer'@'localhost' identified by 'mypass';
Query OK, 0 rows affected (0.02 sec)

mysql> select host,user from mysql.user;
+-----+-----+
| host | user |
+-----+-----+
| %   | root  |
| localhost | customer |
| localhost | mysql.infoschema |
| localhost | mysql.session |
| localhost | mysql.sys |
| localhost | root  |
| localhost | staff |
+-----+-----+
7 rows in set (0.00 sec)
```

Kết quả khi tạo ra 2 user

## 2.2 Phân quyền nhóm người dùng

Đối với những tài khoản đăng ký là nhân viên của ứng dụng, họ sẽ được phân quyền tương tự như user có role là staff. Khi đó họ có thể truy cập vào các bảng và sử dụng các tính năng thông qua các procedure dưới đây:

```
mysql> show grants for 'staff'@'localhost';
+-----+
| Grants for staff@localhost |
+-----+
| GRANT USAGE ON *.* TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`dansachkhohangxuatkhonhieu` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`dansachsucanthoanthe` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`dansachthanhtoanthe` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`dsmuatrongmotngay` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`isbn10` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`khoisbnthang` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`sachmuahieunhat` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tacgiaacosachbanchaynhat` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tongdientumuatrongmotngay` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tongdientuthuetrongmotngay` TO `staff`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tongtruyenthongmuatrongmotngay` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`capnhatgiaodichloi` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`nhapkho` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemdanhsachkhohangxuatkhonhieu` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemdanhsucsucanthoanthe` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemdanhsachthanhtoanthe` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemisbn10` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemkhoisbnthang` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachmuahieunhat` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachmuatrongmotngay` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemtacgiaacosachbanchaynhat` TO `staff`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xuatkho` TO `staff`@`localhost` |
+-----+
24 rows in set (0.00 sec)
```

Kết quả sau khi phân quyền cho staff

Ngược lại, đối với những tài khoản đăng ký là khách hàng của ứng dụng, họ sẽ được phân quyền tương tự như user có role là customer. Khi đó họ có thể truy cập vào các bảng và sử dụng các tính năng thông qua các procedure này:

```
mysql> show grants for 'customer'@'localhost';
+-----+
| Grants for customer@localhost |
+-----+
| GRANT USAGE ON *.* TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`gd2loaitrong1thang` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`gdmuanhieunhatrong1thang` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`giaodichtrong1thang` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`muatrong1thang` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`sachcungnamxb` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`sachcungtacgia` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`sachcungtheloi` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`sachcungtukhoa` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tacgiacungtheloi` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tacgiacungtukhoa` TO `customer`@`localhost` |
| GRANT SELECT ON `ebookstore`.`tongsosachtheotheloi` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`capnhatgiaodich` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`capnhatthongtincanhan` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`chonthetindung` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemgd2loaitrong1thang` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemgdumanhieunhatrong1thang` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemgiaodichtrong1thang` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemmatratrong1thang` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachtheonamxb` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachtheotacgia` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachtheotheloi` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemsachtheotukhoa` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemtacgiatheotukhoa` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemtacgiatheotukhoa` TO `customer`@`localhost` |
| GRANT EXECUTE ON PROCEDURE `ebookstore`.`xemtongsosachtheotheloi` TO `customer`@`localhost` |
+-----+
26 rows in set (0.00 sec)
```

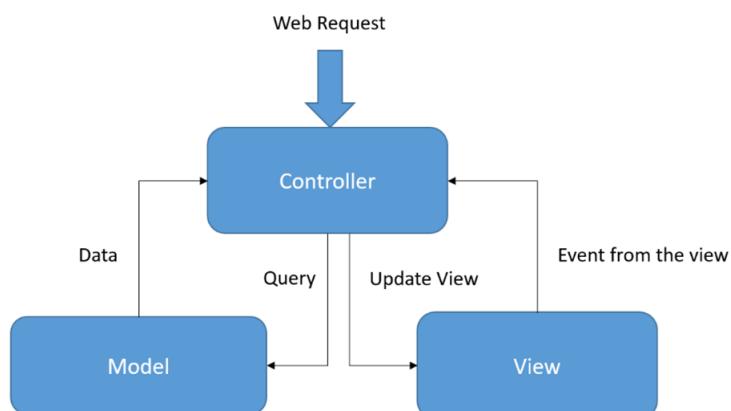
Kết quả sau khi phân quyền cho customer

### 2.3 Phân quyền mức ứng dụng

## 3 Kiến trúc mô hình ứng dụng

### Mô hình MVC

- MVC là viết tắt của Model - View - Controller
- Mô hình MVC là một mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. Nó đóng vai trò quan trọng trong quá trình xây dựng - phát triển - vận hành và bảo trì một hệ thống hay một ứng dụng phần mềm.
- Hay nói cách khác, nó là một mô hình 3 lớp Model-View-Controller, mỗi lớp có một nhiệm vụ riêng biệt và độc lập với các lớp khác





## Các thành phần trong MVC

### Module

- Là nơi chứa tất cả nghiệp vụ tương tác với cơ sở dữ liệu hay là hệ quản trị cơ sở dữ liệu. Nó bao gồm các class và function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm và xóa dữ liệu ...

### View

- Dảm nhận nhiệm vụ hiển thị dữ liệu, giúp người dùng tương tác với hệ thống. là nơi chứa tất cả các file html, form, image ...

### Controller

- Là nơi tiếp nhận và điều hướng những yêu cầu từ phía người dùng bao gồm các class hoặc function xử lý yêu cầu lấy dữ liệu từ Model và trả kết quả cho người dùng thông qua View.

## 3.1 Ứng với đề tài E-Book-Store

### Directory structure

```
> node_modules
└ src
  > middlewares
  └ models
    JS book.js
  └ routes
    JS books-Route.js
    JS index.js
    JS users-Route.js
  └ services
    JS books-service.js
    JS users-services.js
    JS userValidator.js
  └ views
    > layouts
      ~ home.handlebars
      ~ userPage.handlebars
    JS app.js
```

### Model gồm có:

- Folder Model

### Tác dụng

1. Query data từ data base lên và trả lại cái biến chứ data ở dạng Json

### Controllser gồm có:

- Folder Routes
- Folder Services

### Tác dụng

1. Routes dùng để sắp xếp cái url liên quan đến từng thực thể khác nhau. Để dùng phải tạo file index.js trong folder routes. xong rồi có thể tạo tùy thích các route khác nhau như hình

```
const { Router } = require("express");
const booksRouter = require("./books-Route");
const usersRouter = require("./users-Route");

const router = Router({ mergeParams: true });

router.use("/books", booksRouter);
router.use("/users", usersRouter);

module.exports = router;
```



2. Services là dùng để chứa các hàm để hỗ trợ cho hàm route dễ dàng xử lý hơn ví dụ như:

```
const getAllBooks = (callback) => {
  return callback(null, books)
};

module.exports = {
  // createBook,
  getAllBooks
}
```

View gồm có:

- Các file handlebars

Tác dụng

1. xuất các dữ liệu query được lên web server
2. chỉnh giao diện app

Ví dụ:

1. Dùng phương thức req.render bên file app.js để đưa dữ liệu lên file home.handlebars

```
app.get("/", (req, res) => {
  getAllBooks(() => {
    res.render("home", { results });
  });
});
```

2. File handlebar sẽ nhận được dữ liệu gửi lên và có thể xuất ra. Code file home.handlebars như hình

```
<h1>Example App: EBook-Store</h1>
<h2>Show all books</h2>
{#if results}
  {#each results}
    <p>{{this.ISBN}}</p>
    <p>{{this.TenSach}}</p>
    <p>{{this.GiaTien}}</p>
    <p>{{this.TenNhaXuatBan}}</p>
  {/each}
{#/if}}
```

3. Kết quả sẽ được xuất ra trên web server



## Example App: EBook-Store

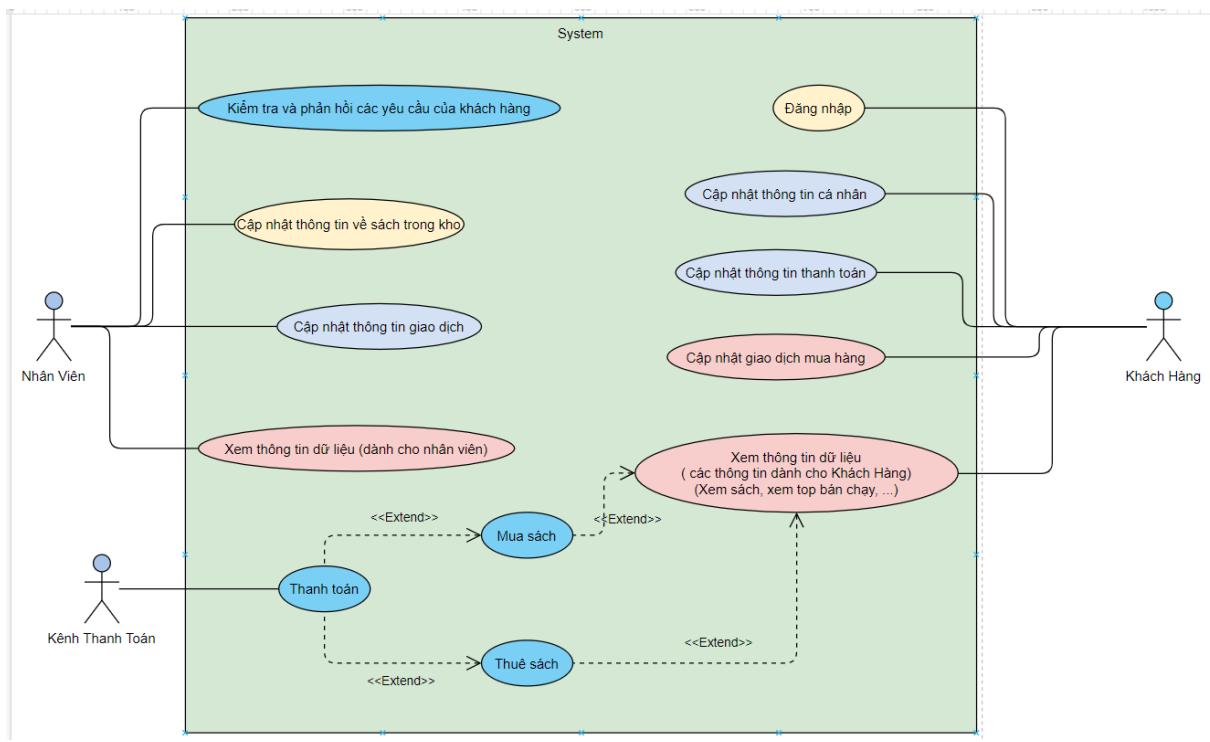
### Show all books

ISBN	TenSach	GiaTien	TenNhaXuatBan
1111111111	CSDL 50000	KimDong	
2222222222	HD qua mon 100000	KimDong	
3333333333	VaTheLaHet 100000	HCM	
4444444444	5phLaDL 100000	HCM	

Tóm lại mô hình MVC được ứng dụng vô app này gồm có controller là routes và services sẽ nhận event từ view và update view là các file handlebars, controller còn nhận các data query được từ Model và dùng nó để update lên view

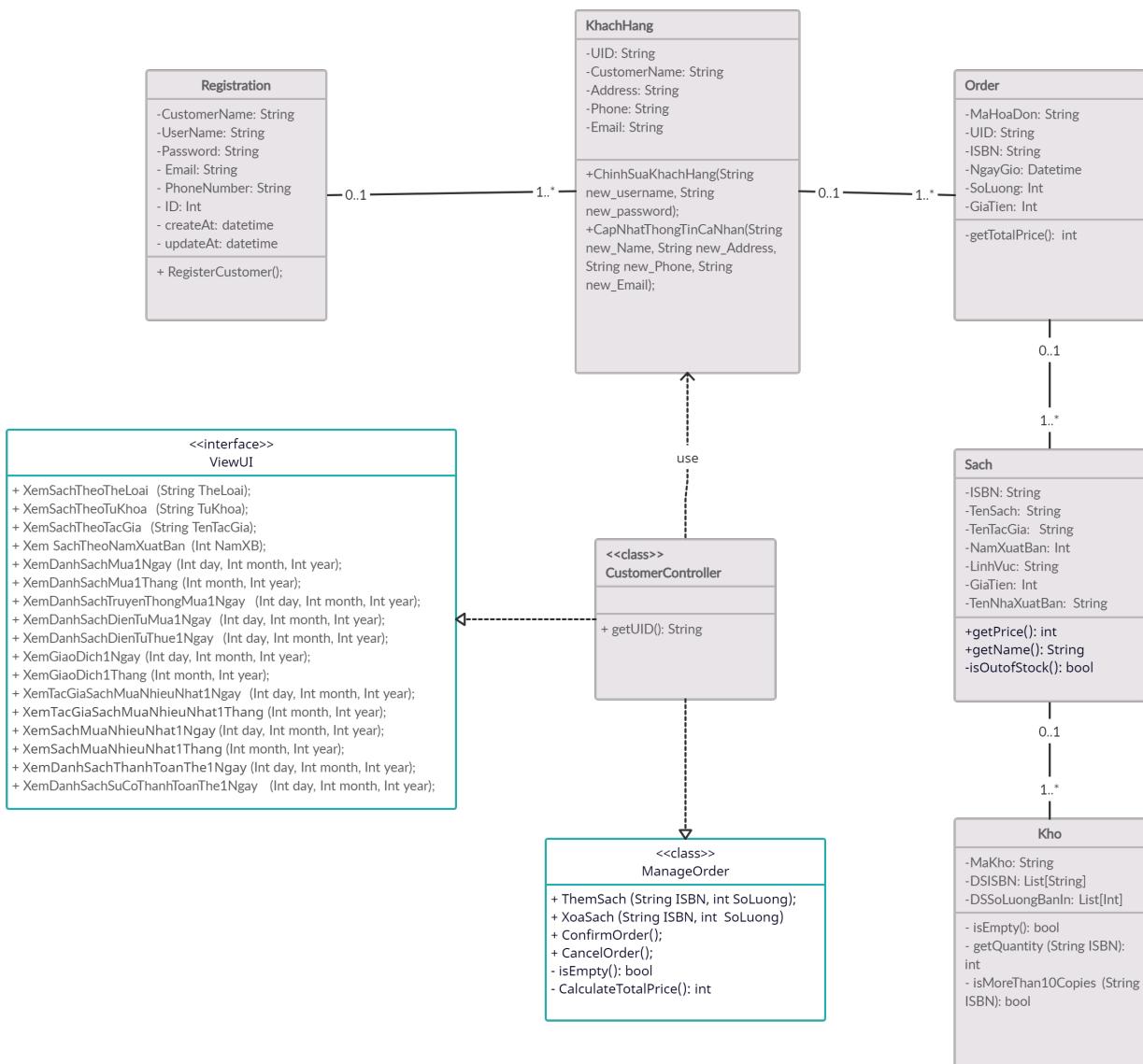
## 4 Thiết kế chi tiết cho ứng dụng

### 4.1 Use Case diagram



Use Case Diagram ở trên thể hiện các actor chính mà hệ thống hướng đến cũng như các chức năng hệ thống hỗ trợ

## 4.2 Class diagram



## 4.3 Module interface

Trong class diagram trên, Interface ManageOrder được dùng để quản lý các order của Khách Hàng. Mỗi Khách Hàng có thể thêm hoặc xóa (một/nhiều) cuốn sách trong giỏ hàng của mình trước khi đưa ra quyết định là hủy hay xác thực đơn hàng đó. Ngoài ra, interface này còn có nhiệm vụ kiểm tra order có rỗng hay không và tính tổng giá tiền của đơn hàng một cách tự động.

## 4.4 Design pattern

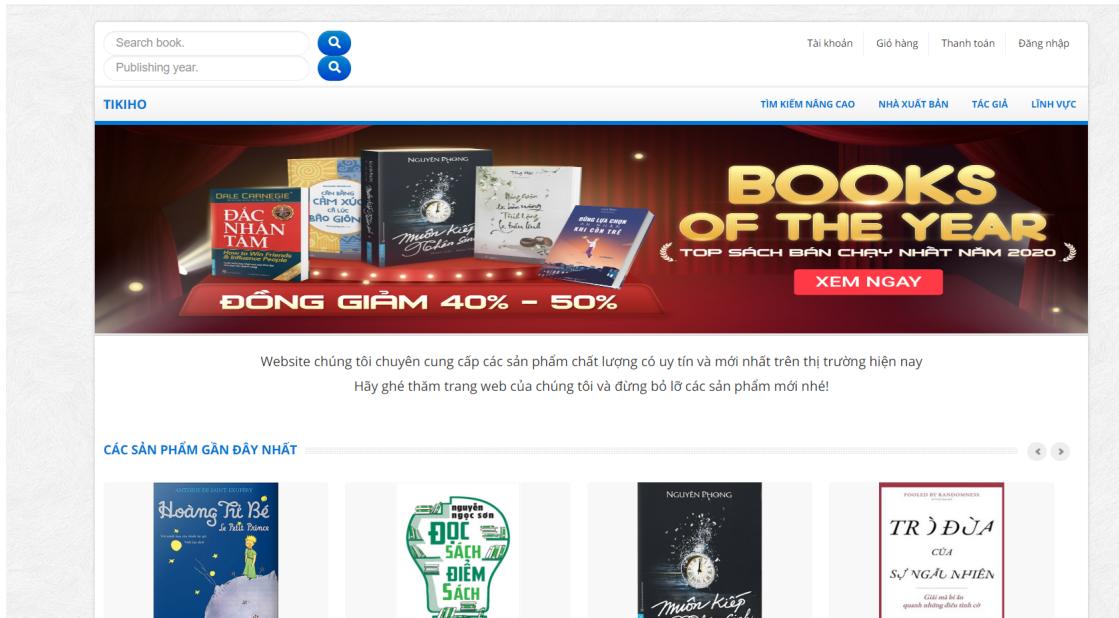
Như đã mô tả ở phần kiến trúc mô hình ứng dụng, ứng dụng của nhóm em được thiết kế theo mẫu MVC pattern chi tiết trong class diagram trên như sau:

- Model** là thành phần trung tâm của mẫu thiết kế. Nó bao gồm các class: KhachHang, Registration, Order, Sach và Kho có chức năng quản lý dữ liệu, logic và các quy tắc của ứng dụng.
- View** gồm có interface ViewUI. Interface này có các chức năng hiển thị những dữ liệu được xử lý từ lớp Controller cho người dùng như: Xem tổng số sách/sách truyền thống/sách điện tử mua trong 1 ngày/tháng, xem tác giả có sách bán chạy nhất trong 1 ngày/tháng, xem danh sách các giao dịch/sự cố thanh toán thẻ trong 1 ngày/tháng,...

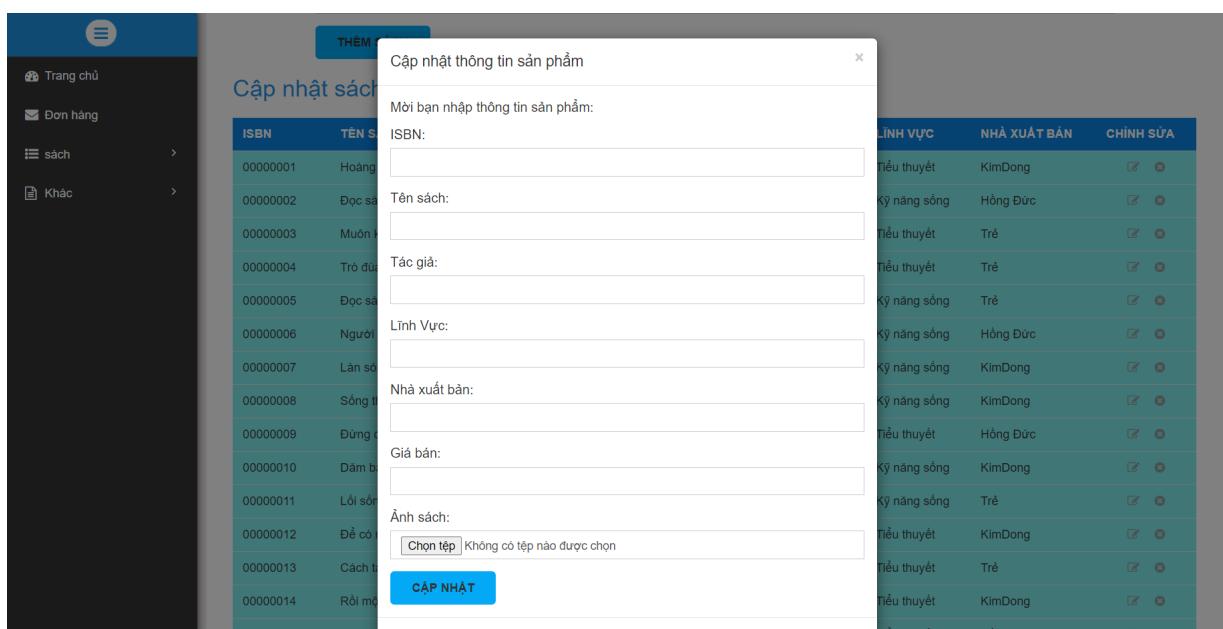


- **Controller** gồm có class CustomerController. Class này là thành phần nhận yêu cầu từ người dùng. Nó có nhiệm vụ lấy dữ liệu nhờ các operation được cung cấp bởi các lớp Model, sau đó hiện thực interface ViewUI và interface ManageOrder (chức năng của interface này đã được trình bày ở mục 4.3).

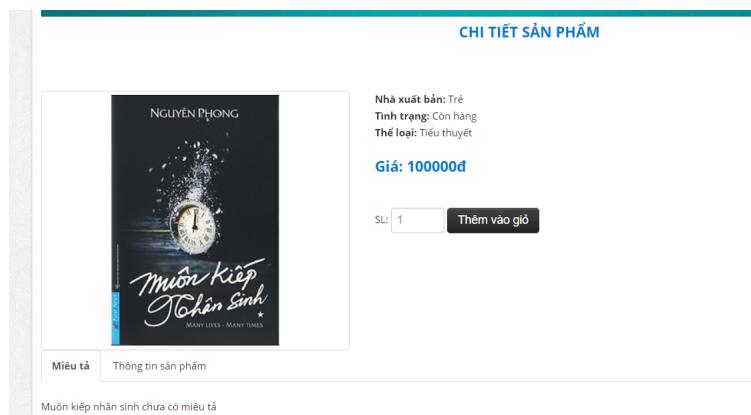
## 4.5 Working demonstration



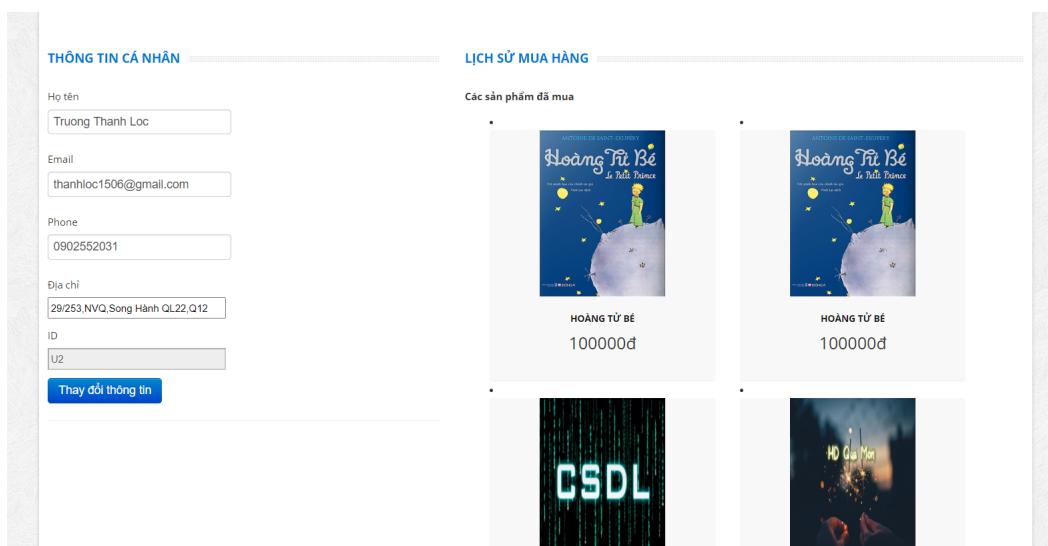
Hình 4.5.1. Giao diện trang chủ của ứng dụng



Hình 4.5.2. Giao diện trang thêm sản phẩm của nhân viên



Hình 4.5.3. Giao diện trang thông tin của sách



Hình 4.5.4. Giao diện trang thông tin của khách hàng

Hình 4.5.5. Giao diện trang giỏ hàng



Hình 4.5.6. Các tính năng tìm kiếm nâng cao của khách hàng

Hình 4.5.7. Tính năng tìm sách theo tên tác giả Nguyễn Nhật Ánh

## 5 Đánh giá các cơ sở dữ liệu

### 5.1 Đánh giá về mặt ngữ nghĩa

Nhóm đã thiết kế một lược đồ quan hệ sao cho có thể dễ dàng giải thích ý nghĩa của các thuộc tính. Không tổ hợp các thuộc tính từ nhiều kiểu thực thể và kiểu liên kết vào một quan hệ đơn. Theo nhóm, một lược đồ quan hệ tương ứng với một kiểu liên kết thì ý nghĩa trở nên rõ ràng. Ngược lại, một quan hệ tương ứng với một hỗn hợp các thực thể và liên kết thì ý nghĩa trở nên không rõ ràng. Lấy ví dụ view một bảng thẻ tín dụng, dựa theo yêu cầu người dùng ta thấy các thuộc tính trong bảng là đầy đủ ngữ nghĩa và không mập mờ



1 • SELECT \* FROM ebookstore.thetindung;

MaThe	TenChuThe	TenNganHang	NgayHetHan	UID	HoaDon
XXX123456	toduyhung	ocb	2021-12-31	1810198	ABC123
XXX456789	toduyhung	bidv	2021-12-31	1810198	XYZ456
*	NULL	NULL	NULL	NULL	NULL

## 5.2 Đánh giá về dư thừa dữ liệu và các cập nhật dị thường

Từ cơ sở dữ liệu đã xây dựng được trong bài tập lớn số 1, nhóm em đã vừa sử dụng để hiện thực ứng dụng vừa liên tục đánh giá và đưa ra những sửa đổi đối với những hạn chế dư thừa dữ liệu và dị thường cập nhật và đã hoàn thành được 90% tiêu chí để được xem là một cơ sở dữ liệu tốt.

### Các trường hợp còn mắc phải trong cơ sở dữ liệu ban đầu

- Trong quan hệ TheTinDung(#MaThe, TenChuThe, TenNganHang, NgayHetHan, UID, HoaDon)

Với mỗi HoaDon được thanh toán bằng MaThe của người dùng A(UID), nhóm em thấy rằng các thông tin MaThe, TenChuThe, TenNganHang, NgayHetHan, UID bị lặp lại một cách không cần thiết dẫn đến dư thừa.

Do đó, nhóm đã tạo ra 1 trigger tên là "ThanhToanBangTheTD" nhằm đảm bảo điều kiện: Mỗi một hóa đơn sau khi được thanh toán bằng thẻ tín dụng thành công thì sẽ bị xóa đi khỏi bảng TheTinDung. Nhờ đó tránh được việc dư thừa dữ liệu trong cơ sở dữ liệu.

Nếu như là không đặt ra một trigger như trên thì việc dư thừa dữ liệu cũng sẽ ảnh hưởng đến việc dễ xuất hiện của các cập nhật dị thường trong cơ sở dữ liệu. Lấy ví dụ, khi ta muốn xóa hoặc sửa đổi một giá trị HoaDon trong bảng GiaoDich nhưng hóa đơn đó cũng là hóa đơn duy nhất của thẻ tín dụng Y vậy thì ta sẽ mất tất cả dữ liệu của thẻ tín dụng Y.

Để tránh việc phát sinh các cập nhật dị thường làm mất tính nhất quán của dữ liệu, nhóm em đã kiểm tra rất kỹ điều kiện của các quan hệ có khóa ngoại tham khảo đến các quan hệ khác. Cách thêm khóa ngoại theo cú pháp: foreign key(< thuộc tính của quan hệ A>) references <quan hệ B>(< khóa dùng để tham khảo của quan hệ B>) **on delete cascade on update cascade** sẽ giúp đồng bộ dữ liệu giữa 2 quan hệ với nhau theo mục đích sử dụng của nhóm.

## 5.3 Đánh giá về các giá trị không xác định (NULL) trong cơ sở dữ liệu

Trong cơ sở dữ liệu, nhóm đã hạn chế tối đa việc sử dụng các thuộc tính có khả năng NULL cao. Bởi lẽ, một giá trị null có thể làm sai sót những kết quả thống kê sử dụng các hàm toán học trong MySQL như: SUM, COUNT, AVG,... những hàm này gặp giá trị null sẽ ngay lập tức bỏ qua và khiến chúng ta mất đi 1 hàng dữ liệu.

Do đó, với những thuộc tính NULL trong cơ sở dữ liệu, nhóm em đều ghi chép và kiểm thử để tránh ảnh hưởng đến việc truy suất và sửa đổi dữ liệu. Đặc biệt, đối với các thuộc tính cần sử dụng các hàm toán học để thống kê như SoLuong, GiaTien, ... luôn được nhóm chú ý thêm điều kiện not null để đảm bảo tránh sự sai sót.

## 5.4 Đánh giá về việc sinh ra các bộ giả

Đối với cơ sở dữ liệu hiện tại của nhóm, nhóm em không tìm thấy trường hợp nào sinh ra các bộ giả sau khi thực hiện các phép nối như: JOIN, NATURAL JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN,... Bởi vì các thuộc tính dùng để nối đều là khóa chính hoặc khóa ngoại tương ứng trong các quan hệ của chúng nên khi thực hiện các phép nối sẽ không bao giờ xuất hiện tình huống phát sinh các bộ giả được. Ngay từ khâu thiết kế luận lí, nhóm đã cực kỳ lưu tâm đến vấn đề và tránh không tạo những quan hệ không đúng có khả năng gây ra các bộ giả.



## 5.5 Đánh giá dạng chuẩn hóa dữ liệu

Các quan hệ trong cơ sở dữ liệu của nhóm em đều:

- Có miền giá trị của các cột có mặt trong bảng chỉ chứa các giá trị nguyên tố (không thể chia nhỏ ra nữa). Do đó, cơ sở dữ liệu của nhóm thỏa dạng chuẩn 1NF.
- Có các thuộc tính không khóa phụ thuộc hàm đầy đủ vào khóa chính. Do đó, cơ sở dữ liệu của nhóm cũng thỏa dạng chuẩn 2NF.

Ví dụ như trong quan hệ GiaoDich(UID, ISBN, NgayGio, Loai, LoaiSach, Soluong, HoaDon), ta có các thuộc tính trong khóa chính lần lượt là {*UID, ISBN, NgayGio*} và các phụ thuộc hàm là: *UID, ISBN, NgayGio* → *Loai, LoaiSach, SoLuong, HoaDon*. Trong đó, thuộc tính *LoaiSach* là loại sách mua trong giao dịch của người dùng (là bản in/bản điện tử) chớ không phải loại hình/thể loại của sách nên không có bất cứ phụ thuộc hàm vào liên quan với ISBN. Cho nên các thuộc tính không khóa trong quan hệ này đều phụ thuộc hàm đầy đủ vào khóa chính.

- Không có thuộc tính không khóa nào phụ thuộc gián tiếp vào khóa chính. Do đó, cơ sở dữ liệu của nhóm đồng thời thỏa dạng chuẩn 3NF.

Cũng lấy ví dụ trong quan hệ GiaoDich như trên, trong các thuộc tính không khóa gồm {*Loai, LoaiSach, SoLuong, HoaDon*} không có bất kỳ thuộc tính nào phụ thuộc vào những thuộc tính không khóa khác, hay nói cách khác mọi thuộc tính không khóa đều phụ thuộc trực tiếp vào khóa chính của quan hệ.

- Không có thuộc tính khóa nào mà phụ thuộc hàm vào thuộc tính không khóa.

Kết luận lại, dạng chuẩn cao nhất của cơ sở dữ liệu mà nhóm xây dựng được là chuẩn BCNF.

## 6 So sánh dữ liệu trong cơ sở dữ liệu và dữ liệu trong các tập tin khi phát triển ứng dụng

- Cơ sở dữ liệu của nhóm em hoàn toàn nhất quán với ràng buộc được cài đặt ở DBMS:

Ví dụ khi nhân viên thêm mới sách vào cơ sở dữ liệu:



THÊM SÁCH

## Cập nhật sách

ISBN	TÊN SÁCH
00000001	Hoàng
00000002	Đọc sá
00000003	Muôn k
00000004	Trò đùa
00000005	Đọc sá
00000006	Người
00000007	Làn só
00000008	Sống th
00000009	Đừng c
00000010	Dăm b
00000011	Lối sốn
00000012	Để có
00000013	Cách t
00000014	Rồi m
00000015	Xe lău

Cập nhật thông tin

Mời bạn nhập thông tin

ISBN:

Tên sách:

Tác giả:

Lĩnh Vực:

Nhà xuất bản:

Giá bán:

Ảnh sách:

Chọn tệp

Kích thước

CẬP NHẬT

Nếu như năm xuất bản nhập vào không là một số tự nhiên từ 1000 đến 9999 hay nhập ngẫu nhiên tên của một tác giả mà không hề xuất hiện trong cơ sở dữ liệu (điều này làm vi phạm điều kiện của foreign key trong quan hệ Sach), khi đó hệ thống của ứng dụng sẽ báo lỗi như sau:



→ ⌂ ⓘ localhost:8080/users/error





Nhờ việc kiểm soát điều kiện chặt chẽ như vậy mà các đối tượng được thêm mới vào luôn đảm bảo tính nhất quán so với các ràng buộc được cài đặt ở DBMS.

- Ngoài ra, những điều kiện trên cũng được đặt ra nhằm đảm bảo tính nhất quán với ràng buộc được cài đặt ở chương trình ứng dụng.

Nói tóm lại, giữa dữ liệu trong cơ sở dữ liệu trên DBMS và dữ liệu được thêm mới khi phát triển ứng dụng luôn đảm bảo được tính nhất quán, chính xác, và đồng bộ với nhau. Khi thêm, xóa hoặc chỉnh sửa từ ứng dụng đều làm thay đổi dữ liệu trên DBMS và đồng thời vẫn đảm bảo được các ràng buộc mà nhóm đã đề ra như: not null, kiểu dữ liệu, foreign key,...

## 7 Cài đặt chỉ mục

### 7.1 Lý thuyết

Index là dữ liệu có cấu trúc như B-Tree giúp cải thiện tốc độ tìm kiếm trên một bảng, làm giảm chi phí thực hiện truy vấn. Việc tối ưu hóa chỉ mục sẽ giúp xác định được vị trí của dữ liệu cần tìm thay vì phải dò theo thứ tự hàng triệu record trong bảng.

Khi ta tạo một khóa chính thì đồng nghĩa với việc ta đã tạo một chỉ mục duy nhất, các hàng không được trùng dữ liệu với nhau, tên gọi của nó là PRIMARY, thuộc thể loại clustered index.

Những chỉ mục khác không phải là PRIMARY thì ta gọi là chỉ mục phụ secondary index hoặc non-clustered indexes.

### 7.2 Tạo bảng dữ liệu demo

Trong phần này, nhóm em sẽ tạo một bảng dữ liệu demo cho bảng *Sach* trong cơ sở dữ liệu có kích thước là 50000 dòng. Các trường sẽ được tạo một cách ngẫu nhiên riêng và giả sử rằng các Tên Nhà Xuất Bản được tạo ra đều tồn tại trong bảng *NhaXuatBan* nhằm vẫn đảm bảo được điều kiện của khóa ngoại.

Dầu tiên, nhóm em điều chỉnh lại cho thuộc tính ISBN trong bảng Sach là INT AUTO\_INCREMENT. Nhờ có điều kiện này mà khi thêm vào bảng một hàng dữ liệu với giá trị của ISBN là null, bảng sẽ tự động gán cho ISBN giá trị là 1 (lúc bắt đầu) và sau đó sẽ là các số tự nhiên tăng dần lên 1 đơn vị.

Sau đó, nhóm em xây dựng 1 procedure như sau:

```
DROP PROCEDURE IF EXISTS BigData;
DELIMITER $$

CREATE PROCEDURE BigData()
BEGIN
    SET @a = 0;
    simple_loop: LOOP
        SET @a=@a+1;
        INSERT INTO Sach() VALUES (null,SUBSTRING(MD5(RAND()), 1, 20),CEIL(RAND() * 1000000),SUBSTRING(MD5(RAND()), 1, 30));
        IF @a=50000 THEN
            LEAVE simple_loop;
        END IF;
    END LOOP simple_loop;
END $$
```

Trong đó:

- Vòng lặp simple\_loop sẽ được thực hiện tổng cộng 50000
- CEIL(RAND() \* 1000000) sẽ tạo ra một số tự nhiên ngẫu nhiên từ 1 đến 1000000
- SUBSTRING(MD5(RAND()), 1, 30)) sẽ tạo ra một chuỗi có các ký tự ngẫu nhiên có kích thước là 30.

Kết quả sau khi tạo ra tập dữ liệu demo như sau:



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 •   SELECT * FROM test.sach;
```

The result grid displays the following data:

	ISBN	TenSach	GiaTien	TenNhaXuatBan
49983	59496deedfed65a4a0c7	3414	0a04cbe18be41e5976fa061e15e931	
49984	e742ed48bf9f115d78fc	2653	2afca7e0f7f88c2ae665b12f3f368b	
49985	787c1ec5d27d85c65501	35	c3d0b1238f3ba148f096de55daf63e	
49986	08f56aa64c98d7c94c95	1148	5f02d4926d41c0f7091dc7f3319e75	
49987	fdfeeee7140d37bf752e	864	1e18dfa2c6e248ce57b9fc4f7addbc	
49988	6d0d7e7aeefad76bbcb8	8133	017d30ff436ab4e4a69c9ff47fb9d9	
49989	c161895b2d1f270af3d8	7284	734f36273719fe8b5e1a453a0af6c0	
49990	499c1bb28c371eba00a3	3112	2bb6f29332fa2ce015b4f01d66e77c	
49991	004d4bc99fbcc1845d9	2279	6d220d3da5a1341d73556df87eda9b	
49992	e5df29196e7335f39989	8275	108892e45be51f759e0a14b6d4af5	
49993	8274de4416710ce18fe5	469	e613abdbce1b745bf709007b5d2b83	
49994	9dd9905d1a326c8b70e9	4028	3f2ad15fb3ee9eb36be2861e685b8	
49995	34d04c0a1e3ee4df2fdb	9544	71d9ae9ec392c2ea4212c6aa76f2e3	
49996	ba1b1b06e80ada2fff43	4773	3b4637cc1a46064d0bf371da60d54	
49997	82bd7914b836ec4667d5	4095	e004ff61f87a4e1cb96cbe8317803	
49998	79ffb7e1de43f9da88f1	8711	2c4ff77414e0931ca22943e0840b1	
49999	970549ceb757fd3b6a38	6279	3f93ba70a28aecf1937fe87d68dcf6	
50000	1c9fd72ddaa3325da192	7116	56d28b26d6d7ab60883c7736da35a9	
*	NULL	NULL	NULL	NULL

### 7.3 Đo đặc và đánh giá

- Dối với bảng dữ liệu ban đầu:

Xem tất cả những index trong bảng bằng lệnh "Show index":

```
148 •   show indexes from Sach;
149
150 •   explain select TenSach from Sach where GiaTien = 4095;
```

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
150 •   explain select TenSach from Sach where GiaTien = 4095;
```

The result grid displays the following data:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
▶	sach	0	PRIMARY	1	ISBN	A	8889	NULL	NULL	NULL	BTREE

Bảng dữ liệu lúc này chỉ chứa khóa chính, đây cũng là một loại index có Key\_name là PRIMARY và index\_type là BTREE. Để truy xuất được thông tin về sách có giá tiền là 4095 thì ta cần phải duyệt qua 48454 dòng dữ liệu (xem ở cột rows trong kết quả khi thêm EXPLAIN trước câu truy vấn):

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
150 •   explain select TenSach from Sach where GiaTien = 4095;
```

The result grid displays the following data:

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	Sach	HULL	ALL	HULL	NULL	NULL	NULL	48454	10.00	Using where

Kết quả khi xem chi tiết câu truy vấn trong bảng cài đặt chỉ mục mặc định

- Dối với bảng dữ liệu sau khi được cài chỉ mục tương ứng với thuộc tính điều kiện dùng để tìm kiếm: Sử dụng lại yêu cầu truy xuất thông tin như trên, trước tiên ta cài đặt chỉ mục cho thuộc tính GiaTien của bảng bằng lệnh "create index GiaTien\_idx on Sach(GiaTien);". Xem tất cả những index trong bảng bằng lệnh "Show index"



```
152 •  create index GiaTien_idx on Sach(GiaTien);
153 •  show indexes from Sach;
154
155 •  explain select TenSach from Sach where GiaTien = 4095;
```

Result Grid   Filter Rows: [ ]   Export: [ ]   Wrap Cell Content: [ ]											
	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
▶	sach	0	PRIMARY	1	ISBN	A	8889	NULL	NULL	NULL	BTREE
▶	sach	1	GiaTien_idx	1	GiaTien	A	6313	NULL	NULL	YES	BTREE

Ta thấy được rằng bảng dữ liệu đã chứa thêm 1 Key\_name là GiaTien\_idx có index\_type cũng là BTREE. Nhờ đó, việc tìm kiếm theo chỉ mục "GiaTien" thay vì khóa "ISBN" chỉ phải duyệt qua 7 dòng dữ liệu để truy xuất được thông tin về sách có giá tiền là 4095:

Result Grid   Filter Rows: [ ]   Export: [ ]   Wrap Cell Content: [ ]													
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra	
▶	1	SIMPLE	Sach	NULL	ref	GiaTien_idx	GiaTien_idx	5	const	7	100.00	NULL	

Kết quả khi xem chi tiết câu truy vấn trong bảng cài đặt chỉ mục cho thuộc tính GiaTien

## 7.4 Kết luận

Sau nhiều thử nghiệm lần lượt trên các tập dữ liệu có kích thước từ nhỏ đến lớn, cho đến các câu truy vấn với điều kiện duyệt từ đơn giản đến phức tạp, nhóm em đã rút ra được một số kết luận về ưu điểm lẫn nhược điểm của việc đặt chỉ mục trong cơ sở dữ liệu như sau:

- **Ưu điểm:**
  - Hiệu suất cao hơn rõ rệt khi nhóm duyệt theo dữ liệu group trong một tập dữ liệu thật sự lớn. Ví dụ như trường hợp duyệt điều kiện TenNhaXuatBan like "Kim%" trong tập dữ liệu Sach với kích thước là 200.000 bản ghi.
  - Quá trình Select, Order dữ liệu cần phải duyệt số dòng dữ liệu ít hơn, nhờ đó giúp thời gian thực hiện nhanh hơn.
- **Nhược điểm:**
  - Tốc độ duyệt tương tự như khi không có Index nếu điều kiện duyệt bắt buộc phải "Vét cạn" hay kích thước của tập dữ liệu không phải là quá lớn. Ví dụ như trường hợp duyệt điều kiện TenNhaXuatBan like "%Kim%". Điều kiện duyệt này buộc cơ sở dữ liệu dù đã đánh index cũng phải tiến hành kiểm tra cẩn thận từng dòng dữ liệu, do đó không thật sự đem lại hiệu quả.
  - Quá trình Insert, update, delete dữ liệu diễn ra khá chậm.
  - Nếu thực hiện nhiều lệnh thay đổi dữ liệu trong bảng sẽ dẫn đến phân mảnh (Fragmentation), dẫn đến hiệu năng sẽ bị giảm.

## 8 Video demo

- Youtube: <https://youtu.be/uZCfRUQnYD8>



## 9 Phân chia công việc

### Tô Duy Hưng (33.33%)

- Bổ sung và hoàn thiện cơ sở dữ liệu
- Cài đặt chỉ mục cho sở dữ liệu
- Thiết kế chi tiết ứng dụng

### Trương Thanh Lộc (33.33%)

- Thiết kế ứng dụng
- Chuẩn bị video demo

### Ngô Nguyễn Quốc Thịnh (33.33%)

- Dánh giá cơ sở dữ liệu

Trong quá trình làm việc thì cả nhóm có trao đổi với nhau. Và cả nhóm đều tham gia vào phần viết báo cáo.