

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



HỆ CƠ SỞ DỮ LIỆU

---

# Thiết kế cơ sở dữ liệu & hiện thực Mô hình nhà sách trực tuyến (E-Book store)

---

GVHD: Đỗ Thanh Thái  
SV: Nguyễn Hữu Khang - 2011365  
Nguyễn Hoàng Trí Viễn - 2015043  
Đậu Xuân Thành - 2014486

TP. HỒ CHÍ MINH, THÁNG 10/2022

## Mục lục

<b>1</b>	<b>Insert dữ liệu mẫu vào cơ sở dữ liệu</b>	<b>2</b>
<b>2</b>	<b>Viết các Functions/Procedures/Assertions/Triggers</b>	<b>5</b>
<b>3</b>	<b>Định nghĩa các nhóm người dùng và phân quyền ở mức ứng dụng</b>	<b>37</b>
<b>4</b>	<b>Định nghĩa kiến trúc cho ứng dụng</b>	<b>38</b>
4.1	Mô hình MVC . . . . .	38
4.2	Kiến trúc ứng dụng thực tế của nhóm . . . . .	38
4.3	Mình họa kiến trúc ứng dụng . . . . .	38
<b>5</b>	<b>Thiết kế chi tiết cho ứng dụng</b>	<b>39</b>
<b>6</b>	<b>Chuẩn hóa lược đồ về dạng chuẩn BCNF cho lược đồ quan hệ:</b>	<b>47</b>
<b>7</b>	<b>So sánh dữ liệu trong cơ sở dữ liệu và dữ liệu trong các tập tin như text files khi phát triển ứng dụng cho đề tài của assignment</b>	<b>60</b>
<b>8</b>	<b>Cài đặt chỉ mục, minh họa đo đạc: so sánh thời gian thực thi chỉ mục với demo lớn:</b>	<b>76</b>



## 1 Insert dữ liệu mẫu vào cơ sở dữ liệu

- **Bảng Account:**

Insert một số dữ liệu mẫu vào bảng Account như sau:

```
INSERT INTO `account` VALUES
(1,'Viễn','Trí','0373015428','2022-11-13 23:37:20','Quang Ngai','2002-11-14','vien.nguyen999@hcmut.edu.vn','123456',4,0),
(2,'Hùng','Trần','0373015422','2022-11-14 02:03:56','Da Nang','2002-10-10','trivien@hcmut.com','123456789',4,0),
(3,'Đào','Lê','0984321486','2022-11-18 10:46:12','Khanh Hoa','2000-02-28','ledao@gmail.com','abcdef',4,0),
(4,'Lực','Nguyễn','0312212186','2022-11-18 10:47:49','Khanh Hoa','2000-01-01','abc@gmail.com','1234567',4,0),
(5,'Toàn','Trần','0955325514','2022-11-18 10:49:06','Da Nang','1994-01-01','toantran123@gmail.com','trantoan',4,0),
(6,'Dũng','Nguyễn','0373225421','2022-11-18 10:51:20','Hue','1999-02-02','nguyenvdung@gmail.com','123abc',4,0),
(7,'Ánh','Lê','0312125482','2022-11-18 10:52:42','TP Ho Chi Minh','1900-10-10','lethianh@gmail.com','abc1123',4,0),
(8,'Ngọc','Trần','0984821273','2022-11-18 10:54:15','Lam Dong','1900-11-11','tranthingoc@gmail.com','abc1234',4,0),
(9,'Dương','Nguyễn','0373015412','2022-11-18 10:56:06','Hue','2001-10-11','nguyenduong@hcmut.edu.vn','nd12345',4,0),
(10,'Đạt','Nguyễn','0379841848','2022-11-18 10:59:42','Phu Yen','2001-10-11','datnguyennn@gmail.com','ntdntd',4,0),
(11,'Thư','Anh','0984714416','2022-11-18 11:01:22','Phu Yen','2000-01-11','anhthunguyen@gmail.com','anhthu123',4,0),
(12,'Nam','Nguyễn','0355254417','2022-11-18 11:02:54','Phu Yen','1998-12-12','namnguyen1212@gmail.com','namnguyen123',4,0),
(13,'Hùng','Lê','0984812298','2022-11-18 11:04:32','Phu Yen','1998-11-11','hungleabc@gmail.com','lehungabc',4,0),
(14,'Ngọc','Ngọc','0122317173','2022-11-18 11:06:39','Quang Ngai','2000-02-15','ngocngoc@gmail.com','ngocngocabc@gmail.com',4,0),
(15,'Vũ','Trần','0916194255','2022-11-18 11:09:07','Da Nang','2000-11-14','tranthanhvu@gmail.com','tranvuabc',4,0),
(16,'Nguyễn','Nguyễn','0321219841','2022-11-18 11:09:07','Khanh Hoa','2000-01-01','nguyennguyen@gmail.com','nguyenabc',4,0);
```

Kết quả dữ liệu đã được insert thành công:

	Account_ID	FName	LName	TelephoneNum	Start_date	Address	Birthday	Email	Password	ROLE_ID
	1	Viễn	Trí	0373015428	2022-11-13 23:37:20	Quang Ngai	2002-11-14	vien.nguyen999@hcmut.edu.vn	123456	4
	2	Hùng	Trần	0373015422	2022-11-14 02:03:56	Da Nang	2002-10-10	trivien@hcmut.com	123456789	4
	3	Đào	Lê	0984321486	2022-11-18 10:46:12	Khanh Hoa	2000-02-28	ledao@gmail.com	abcdef	4
	4	Lực	Nguyễn	0312212186	2022-11-18 10:47:49	Khanh Hoa	2000-01-01	abc@gmail.com	1234567	4
	5	Toàn	Trần	0955325514	2022-11-18 10:49:06	Da Nang	1994-01-01	toantran123@gmail.com	trantoan	4
	6	Dũng	Nguyễn	0373225421	2022-11-18 10:51:20	Hue	1999-02-02	nguyenvdung@gmail.com	123abc	4
	7	Ánh	Lê	0312125482	2022-11-18 10:52:42	TP Ho Chi Minh	1900-10-10	lethianh@gmail.com	abc1123	4
	8	Ngọc	Trần	0984821273	2022-11-18 10:54:15	Lam Dong	1900-11-11	tranthingoc@gmail.com	abc1234	4
	9	Dương	Nguyễn	0373015412	2022-11-18 10:56:06	Hue	2001-10-11	nguyenduong@hcmut.edu.vn	nd12345	4
	10	Đạt	Nguyễn	0379841848	2022-11-18 10:59:42	Phu Yen	2001-10-11	datnguyennn@gmail.com	ntdntd	4
	11	Thư	Anh	0984714416	2022-11-18 11:01:22	Phu Yen	2000-01-11	anhthunguyen@gmail.com	anhthu123	4
	12	Nam	Nguyễn	0355254417	2022-11-18 11:02:54	Phu Yen	1998-12-12	namnguyen1212@gmail.com	namnguye...	4
	13	Hùng	Lê	0984812298	2022-11-18 11:04:32	Phu Yen	1998-11-11	hungleabc@gmail.com	lehungabc	4
	14	Ngọc	Ngọc	0122317173	2022-11-18 11:06:39	Quang Ngai	2000-02-15	ngocngoc@gmail.com	ngocngoc...	4
	15	Vũ	Trần	0916194255	2022-11-18 11:09:07	Da Nang	2000-11-14	tranthanhvu@gmail.com	tranvuabc	4
	16	Nguyễn	Nguyễn	0321219841	2022-11-18 11:09:07	Khanh Hoa	2000-01-01	nguyennguyen@gmail.com	nguyenabc	4
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### – Bảng Product:

Insert một số dữ liệu mẫu vào bảng Product như sau:

```
INSERT INTO `product` VALUES
('https://cf.shopee.vn/file/510df9c4781fec3532c03c59a355eaf0','Truyện thám tử lừng danh conan tập 98',1,24000,20000,'Nhà xuất bản Kim Đồng',
('https://salt.tikicdn.com/cache/w1200/media/catalog/product/d/a/dacnhamtam_2_1_1.jpg','Đặc nhân tâm',2,108000,97200,'Nhà xuất bản trẻ','Đồ
('https://cf.shopee.vn/file/740581c5b16270610a6420654fee77d5','Cha giàu cha nghèo',3,99000,69000,'Nhà xuất bản Lao động xã hội','Nếu bạn mu
('https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/16303853381/32836144.jpg','Khéo ăn nói sẽ có được thiên hạ',4,12000
```

Kết quả dữ liệu đã được insert thành công:

Thumbnail	Name	Product_ID	Price	Discount_price	Publisher	Description	Deleted	Typeproduct
https://cf.shopee...	Truyện thám t...	1	24000	20000	Nhà xuất bản Ki...	Mở đầu câu tru...	0	1
https://salt.tikic...	Đặc nhân tâm	2	108000	97200	Nhà xuất bản trẻ	Đặc nhân tâm -...	0	1
https://cf.shopee...	Cha giàu cha n...	3	99000	69000	Nhà xuất bản L...	Nếu bạn muốn ...	0	1
https://i.gr-asse...	Khéo ăn nói sẽ...	4	120000	113000	Nhà xuất bản V...	Trong xã hội th...	0	1

\* Ở đây nhóm có bổ sung thêm thuộc tính Typeproduct để phân biệt loại sản phẩm giữa sách với tạp chí (với Typeproduct của Sách là 1 và Typeproduct của tạp chí là 2).

### – Bảng Orders:

Insert một số dữ liệu mẫu vào bảng Orders như sau:

```
INSERT INTO `orders` VALUES
(1,'Completed',250000,'số 8, đường Mai Chí Thọ, quận 2, thành phố Hồ Chí Minh',NULL,1,2,'Giao nhanh giúp mình nha shop'),
(2,'Completed',99000,'số 20, đường Nguyễn Lương Bằng, quận 7',NULL,2,2,'Shop đóng gói cẩn thận giúp mình nha'),
(3,'Completed',120000,'123 đường abc',NULL,3,2,'Shop giao nhanh giúp mình'),
(4,'Completed',100000,'123 đường abc',NULL,4,2,'Shop giao nhanh giúp mình'),
(5,'Completed',500000,'123 đường abc',NULL,5,2,'Shop giao nhanh nhanh nha'),
(6,'Completed',200000,'123 đường abc',NULL,14,2,'Shop giao nhanh nha'),
(7,'Completed',999999,'123 đường abc',NULL,15,2,'Giao lẹ nha shop'),
(8,'Completed',1200000,'123 đường abc',NULL,16,2,'Giao sớm giúp mình'),
(9,'Delivering',100000,'số 32 đường Nguyễn Du',NULL,6,2,'abcd'),
(10,'Canceled',200000,'số 4 đường Mai Chí Thọ',NULL,7,2,'abcde'),
(12,'Completed',100000,'123 đường abc',NULL,1,2,'abcde'),
(13,'Completed',100000,'số 12 đường An Dương Vương',NULL,2,2,'abcd'),
(14,'Completed',299000,'123 đường abc',NULL,3,2,'abcd'),
(15,'Completed',399000,'123 đường abc',NULL,3,2,'abcd'),
(16,'Completed',390000,'123 đường Võ Văn Ngân',NULL,4,2,'abcd'),
(17,'Canceled',250000,'123 đường abc',NULL,14,2,'aaaa');
```



Kết quả dữ liệu đã được insert thành công:

Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID
1	Completed	250000	số 8, hẻm 2, dương 120	NULL	1	2
2	Completed	99000	số 20, đường Nguyễn L...	NULL	2	2
3	Completed	120000	123 đường abc	NULL	3	2
4	Completed	100000	123 đường abc	NULL	4	2
5	Completed	500000	123 đường abc	NULL	5	2
6	Completed	200000	123 đường abc	NULL	14	2
7	Completed	999999	123 đường abc	NULL	15	2
8	Completed	1200000	123 đường abc	NULL	16	2
9	Delivering	100000	số 32 đường Nguyễn Du	NULL	6	2
10	Canceled	200000	số 4 đường Mai Chí Thọ	NULL	7	2
12	Completed	100000	123 đường abc	NULL	1	2
13	Completed	100000	số 12 đường An Dương ...	NULL	2	2
14	Completed	299000	123 đường abc	NULL	3	2
15	Completed	399000	123 đường abc	NULL	3	2
16	Completed	390000	123 đường Võ Văn Ngân	NULL	4	2
17	Canceled	250000	123 đường abc	NULL	14	2

## 2 Viết các Functions/Procedures/Assertions/Triggers

- Bài làm của Nguyễn Hữu Khang

- **Trigger:**

**Trigger trên bảng book:** Trong một số trường hợp nhất định, nếu sách trong cửa hàng bị thất lạc, hoặc bị mất thì chủ cửa hàng phải cập nhật lại số lượng còn lại trong cửa hàng của cuốn sách đã bị thất lạc đó. Sau khi cập nhật lại số lượng sách trong cửa hàng, nếu số sách hiện có là ít hơn số sách có trong một đơn đặt hàng của một khách hàng và đơn hàng đó còn ở trạng thái đang xử lý (Processing) thì đơn hàng sẽ chuyển sang trạng thái đã hủy (Canceled)

```
CREATE DEFINER='root'@'localhost' TRIGGER 'book_AFTER_UPDATE' AFTER UPDATE
ON 'book' FOR EACH ROW BEGIN
DECLARE done INT DEFAULT FALSE;
DECLARE x int;
DECLARE mycur CURSOR FOR SELECT ORDERID
                           FROM order_detail, orders
                           WHERE Quantity > New.Quantity_in_store and
                              ORDERID=Order_ID and PID = New.Product_ID and
                              Status like '%Processing%';
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
open mycur;
My_Loop : loop
    fetch mycur into x;
    if done then Leave My_Loop;
    end if;
    UPDATE orders SET Status = 'Canceled' WHERE (Order_ID = x);
END loop;
CLOSE mycur;
END
```

**Câu lệnh để kiểm tra thủ tục hoạt động:** VD: Đơn hàng có ID là 102, đang có trạng thái đơn hàng là: Processing (đang xử lý).

Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID	Note
10	Canceled	200000	số 4 đường Mai Chí Thọ	NULL	7	2	abcdef
12	Completed	100000	123 đường abc	NULL	1	2	abcde
13	Completed	100000	số 12 đường An Dương...	NULL	2	2	abcd
14	Completed	299000	123 đường abc	NULL	3	2	abcd
15	Completed	399000	123 đường abc	NULL	3	2	abcd
16	Completed	390000	123 đường Võ Văn Ngân	NULL	4	2	abcd
17	Canceled	250000	123 đường abc	NULL	14	2	aaaa
101	Canceled	200000	123 vũ văn nhàn	NULL	1	2	aaaa
102	Processing	20000	123 Võ Thị Sáu	NULL	1	2	abcd

Ta xem chi tiết đơn hàng có ID là 102 ở bảng order\_detail thì thấy rằng đơn hàng này bao gồm 10 sản phẩm ID là 1.

	Detail_ID	Price	Quantity	Total_cost	ORDERID	PID
▶	1	20000	2	40000	1	1
	2	97200	1	97200	1	2
	3	20000	1	20000	2	1
	4	20000	10	200000	101	1
	5	20000	10	200000	102	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Lúc này chuyển sang bảng Book ta thấy ở cuốn sách có ID = 1 thì số lượng sách trong cửa hàng hiện tại là 50 cuốn. Lúc này ta thay đổi số lượng sách thành một số bất kỳ bé hơn 10 và lưu lại.

	Product_ID	Publish_year	Quantity_in_store	CATEG_ID
▶	1	2020	50	1
	2	2020	50	4
	3	2020	50	4
	4	2020	50	4
	5	NULL	50	NULL
*	NULL	NULL	NULL	NULL

	Product_ID	Publish_year	Quantity_in_store	CATEG_ID
	1	2020	8	1
	2	2020	50	4
	3	2020	50	4
	4	2020	50	4
▶	5	NULL	50	NULL
*	NULL	NULL	NULL	NULL

Lúc này ta quay lại bảng đơn hàng (orders) để kiểm tra thì thấy rằng đơn hàng có ID là 102 đã được chuyển sang trạng thái Canceled từ trạng thái ban đầu là Processing.



Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID	Note
4	Completed	100000	123 đường abc	NULL	4	2	Shop giao nhanh giú...
5	Completed	500000	123 đường abc	NULL	5	2	Shop giao nhanh nha...
6	Completed	200000	123 đường abc	NULL	14	2	Shop giao nhanh nha
7	Completed	999999	123 đường abc	NULL	15	2	Giao lệ nha shop
8	Completed	1200000	123 đường abc	NULL	16	2	Giao sớm giúp mình
9	Delivering	100000	số 32 đường Nguyễn Du	NULL	6	2	abcd
10	Canceled	200000	số 4 đường Mai Chí Thọ	NULL	7	2	abcdef
12	Completed	100000	123 đường abc	NULL	1	2	abcde
13	Completed	100000	số 12 đường An Dương ...	NULL	2	2	abcd
14	Completed	299000	123 đường abc	NULL	3	2	abcd
15	Completed	399000	123 đường abc	NULL	3	2	abcd
16	Completed	390000	123 đường Võ Văn Ngân	NULL	4	2	abcd
17	Canceled	250000	123 đường abc	NULL	14	2	aaaa
101	Canceled	200000	123 vô văn nản	NULL	1	2	aaaa
102	Canceled	20000	123 Vô Thị Sáu	NULL	1	2	abcd

#### – Procedures:

##### Thủ tục 1: Tìm top những cuốn sách bán chạy nhất.

Thủ tục này có tên là topsachbanchay. Chức năng của thủ tục này là để hiển thị các thông tin của top 5 các cuốn sách bán chạy nhất như: Thumbnail, Tên sách, Giá Tiền, Thể loại sách và tổng số sách đã bán. Lưu ý: tổng số lượng đã bán của từng cuốn sách chỉ được cộng lại từ những đơn hàng đã ở trạng thái hoàn thành (Completed).

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'topsachbanchay'()  
BEGIN  
select Thumbnail, P.Name as TenSach, P.Discount_price as GiaTien, C.Name as  
Theloaisach ,sum(O.Quantity) as Tongsosachdaban  
FROM product P, book B, category C, order_detail O, orders D  
where P.Typeproduct=1 and P.Product_ID = B.Product_ID and B.CATEG_ID =  
C.Category_ID and P.Product_ID = O.PID and O.ORDERID=D.Order_ID and  
D.Status like '%Completed%'  
group by Thumbnail, TenSach, GiaTien, Theloaisach  
ORDER BY sum(O.Quantity) desc  
limit 5;  
END
```

**Câu lệnh để kiểm tra thủ tục hoạt động:** Chúng ta gọi thủ tục topsachbanchay như sau:

```
|| call baitaplon.topsachbanchay();
```

Kết quả hiện thị ra màn hình thể hiện có 2 sản phẩm đang bán chạy là "Truyện thám tử lừng danh conan tập 98" và "Đắc nhân tâm".

	Thumbnail	TenSach	GiaTien	Theloaisach
▶	<a href="https://cf.shopee...">https://cf.shopee...</a>	Truyện thám tử lừng danh conan tập 98	20000	Truyen tranh
	<a href="https://salt.tikic...">https://salt.tikic...</a>	Đắc nhân tâm	97200	Sach ky nang

Tuy thủ tục này lấy ra top 5 sản phẩm bán chạy nhưng kết quả hiện ra chỉ có 2 sản phẩm vì trong bảng chi tiết đơn hàng (order\_detail) chỉ mới có 2 sản phẩm (PID = 1 và PID = 2) được mua.



	Detail_ID	Price	Quantity	Total_cost	ORDERID	PID
▶	1	20000	2	40000	1	1
	2	97200	1	97200	1	2
	3	20000	1	20000	2	1
✱	NULL	NULL	NULL	NULL	NULL	NULL

### • Bài làm của Nguyễn Hoàng Trí Viễn

#### – Trigger:

**Trigger trên bảng Product:** Trigger sẽ tự động thêm vào bảng Book hoặc bảng Magazine\_seri khi ta nhập một sản phẩm mới vào bảng Product. Ví dụ nếu ta thêm mới một sản phẩm ở bảng Product và loại sản phẩm này là sách (Typeproduct = 1) thì trigger này sẽ fire và sẽ thêm vào bảng Book với ProductID (khóa chính của bảng Book) cũng chính là ProductID khóa chính của bảng Product được thêm, tương tự với sản phẩm là tạp chí.

```
CREATE DEFINER='root'@'localhost' TRIGGER 'product_AFTER_INSERT' AFTER
INSERT ON 'product' FOR EACH ROW BEGIN
if(NEW.Typeproduct = 1) then INSERT INTO book(Product_ID) values
(NEW.Product_ID);
elseif(NEW.Typeproduct = 2) then INSERT INTO magazine_seri(Product_ID)
values (NEW.Product_ID);
end if;
END

CREATE DEFINER='root'@'localhost' TRIGGER 'product_AFTER_UPDATE' AFTER
UPDATE ON 'product' FOR EACH ROW BEGIN
if(NEW.Typeproduct = 1) then INSERT INTO book(Product_ID) values
(NEW.Product_ID);
elseif(NEW.Typeproduct = 2) then INSERT INTO magazine_seri(Product_ID)
values (NEW.Product_ID);
end if;
END
```

#### Câu lệnh để kiểm tra trigger hoạt động:

```
Insert du lieu cua mot cuon sach (Typeprodct = 1) vao trong bang Product
INSERT INTO 'baitaplon'. 'product' ('Thumbnail', 'Name', 'Price',
'Discount_price', 'Publisher', 'Description', 'Typeproduct') VALUES
('https://m.media-amazon.com/images/I/41SNoh5ZhOL._AC_SY780_.jpg',
'Introduction to Algorithms', '200000', '180000', 'Nhà xuất bản giao
duc', 'Cuon sach giúp bạn hiểu rõ về các giai thuật', '1');
```

#### Hình ảnh sau khi Insert thành công :

1 **SELECT \* FROM baitaplon.product;**

Result Grid

	Thumbnail	Name	Product_ID	Price	Discount_price	Publisher	Description	Deleted	Typeproduct
▶	https://cf.shopee...	Truyện thám t...	1	24000	20000	Nhà xuất bản K...	Mở đầu câu tru...	0	1
	https://salt.bic...	Đặc nhân tâm	2	108000	97200	Nhà xuất bản tré	Đặc nhân tâm -...	0	1
	https://cf.shopee...	Cha giàu cha n...	3	99000	69000	Nhà xuất bản L...	Nếu bạn muốn ...	0	1
	https://i.g-aspe...	Khéo ăn nói sẽ ..	4	120000	113000	Nhà xuất bản V...	Trong xã hội th...	0	1
	https://m.media-...	Introduction t...	5	200000	180000	Nhà xuất bản gi...	Cuốn sách giúp ...	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

product 1 x

Output

Action Output

#	Time	Action	Message
✓ 12	22:07:15	Apply changes to book	Changes applied
✓ 13	22:34:37	INSERT INTO 'baitaplon'.product ('Thumbnail', 'Name', 'Price', 'Discount_price', ...	1 row(s) affected
✓ 14	22:35:19	SELECT * FROM baitaplon.product LIMIT 0, 1000	5 row(s) returned

Kiểm tra thấy rằng trong bảng Book thì sách có Product\_ID = 5 đã được thêm vào thành công -> Trigger đã hoạt động.

Product_ID	Publish_year	Quantity_in_store	CATEG_ID
1	2020	50	1
2	2020	50	4
3	2020	50	4
4	2020	50	4
5	NULL	50	NULL
NULL	NULL	NULL	NULL

**Trigger trên bảng Account:** Vì một số lý do nên cửa hàng yêu cầu độ tuổi nhất định cho mỗi loại tài khoản người dùng. Với tài khoản là nhân viên của cửa hàng (nhân viên thường, nhân viên kho) và quản lý của hàng thì độ tuổi yêu cầu là từ 18 tuổi trở lên. Với tài khoản là khách hàng thì độ tuổi yêu cầu là từ 12 tuổi. Sau đây là trigger sau khi insert trên bảng account, nếu tuổi của tài khoản bé hơn độ tuổi quy định thì sẽ xuất hiện thông báo lỗi.

Ta sẽ phân biệt giữa khách hàng, nhân viên thường, nhân viên quản lý kho và quản lý thông qua khóa ngoại ROLE\_NO tham chiếu đến khóa chính Role\_ID của bảng Role.

	Role_ID	Role_name
▶	1	quanly
	2	nhanvienthuong
	3	nhanvienkho
	4	khachhang
*	NULL	NULL

```
CREATE DEFINER='root'@'localhost' TRIGGER `account_BEFORE_INSERT` BEFORE
INSERT ON `account` FOR EACH ROW BEGIN
IF(NEW.ROLE_NO = 4 and (datediff(curdate(),NEW.Birthday)/365 < 12)) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Age is not allowed
    to be less than 12';
ELSEIF(datediff(curdate(),NEW.Birthday)/365 < 18) THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Age is not allowed
    to be less than 18';
END IF;
end
```

**Câu lệnh để kiểm tra trigger hoạt động:**

Ta thử nhập thông tin của khách hàng (với ROLE\_NO = 4) và một ngày sinh không thỏa yêu cầu, VD: 2015-10-10

```
INSERT INTO `baitaplon`.`account` ('FName', 'LName', 'TelephoneNum',
'Address', 'Birthday', 'Email', 'Password', 'ROLE_NO') VALUES
('An', 'Nguyen', '0123412516', 'Quang Ngai', '2015-10-10',
'anan@gmail.com', 'abc123', '4');
```

Hình ảnh thông báo lỗi vì ngày sinh của người dùng so với ngày hiện tại của hệ thống là bé hơn 12 năm :



```
2 • INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address`, `Birthday`, `Email`, `Password`, `ROLE_NO`) VALUES
3   ('An', 'Nguyen', '0123412516', 'Quang Ngai', '2015-10-10', 'anan@gmail.com', 'abc123', '4');
4
```

Result Grid

Account_ID	FName	LName	TelephoneNum	Start_date	Address	Birthday	Email	Password	ROLE_NO	Deleted
6	Dũng	Nguyễn	0373225421	2022-11-18 10:51:20	Hue	1999-02-02	nguyenvdung@gmail.com	123abc	4	0
7	Ảnh	Lê	0312125482	2022-11-18 10:52:42	TP Ho Chi Minh	1900-10-10	lethianh@gmail.com	abc1123	4	0
8	Ngọc	Trần	0984821273	2022-11-18 10:54:15	Lam Dong	1900-11-11	tranthingoc@gmail.com	abc1234	4	0
9	Dương	Nguyễn	0373015412	2022-11-18 10:56:06	Hue	2001-10-11	nguyenduong@hcmut.edu.vn	nd12345	4	0
10	Đạt	Nguyễn	0379841848	2022-11-18 10:59:42	Phu Yen	2001-10-11	datnguyennn@gmail.com	ntdntd	4	0
11	Thư	Anh	0984714416	2022-11-18 11:01:22	Phu Yen	2000-01-11	anhthunguyen@gmail.com	anhthu123	4	0
12	Nam	Nguyễn	0355254417	2022-11-18 11:02:54	Phu Yen	1998-12-12	namnguyen1212@gmail.com	namnguye...	4	0
13	Hùng	Lê	0984812298	2022-11-18 11:04:32	Phu Yen	1998-11-11	hungleabc@gmail.com	lehungabc	4	0

account 1 x Apply

Output

Action Output

#	Time	Action	Message	Duration / Fets
30	23:02:25	DELETE FROM `baitaplon`.`product` WHERE ('Product_ID' = '5')	1451: Cannot delete or update a parent row: a foreign key constraint fails (baitaplon...	
31	23:15:40	SELECT * FROM baitaplon.account LIMIT 0, 1000	16 row(s) returned	0.016 sec / 0
32	23:17:16	INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address...	1 row(s) affected	0.047 sec
33	23:17:38	INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address...	Error Code: 1644. Age is not allowed to be less than 12	0.000 sec

Ta thử nhập thông tin của nhân viên quản lý kho ( $ROLE\_NO = 3$ ) và một ngày sinh không thỏa yêu cầu, VD: 2008-10-10

```
INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`,
`Address`, `Birthday`, `Email`, `Password`, `ROLE_NO`)
VALUES ('An', 'Nguyen', '0123412516', 'Quang Ngai', '2008-10-10',
'anan@gmail.com', 'abc123', '3');
```

Hình ảnh thông báo lỗi vì ngày sinh của người dùng so với ngày hiện tại của hệ thống là bé hơn 18 năm :

```
2 • INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address`, `Birthday`, `Email`, `Password`, `ROLE_NO`)
3   VALUES ('An', 'Nguyen', '0123412516', 'Quang Ngai', '2008-10-10', 'anan@gmail.com', 'abc123', '3');
4 • use baitaplon
```

Output

Action Output

#	Time	Action	Message	D
86	14:00:03	SELECT * FROM baitaplon.role LIMIT 0, 1000	4 row(s) returned	0.0
87	14:02:30	INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address...	Error Code: 1644. Age is not allowed to be less than 18	0.0
88	14:03:21	INSERT INTO `baitaplon`.`account` (`FName`, `LName`, `TelephoneNum`, `Address...	Error Code: 1644. Age is not allowed to be less than 18	0.0

– Procedures:

**Thủ tục 1: Về nhập dữ liệu trên bảng Account**

Thủ tục insert trong bảng Account có tên là insertaccount. Chức năng của thủ tục này để kiểm tra xem giá trị của các thuộc tính nhập vào như: TelephoneNum, email, password ,.. có đúng định dạng hay chưa. Các định dạng để kiểm tra input đầu vào như sau:

- + ) TelephoneNum chỉ bao gồm các chữ số và bắt đầu bằng chữ số 0.
- + ) Email phải theo định dạng something@something.something.
- + ) Password phải có độ dài lớn hơn hoặc bằng 8 kí tự, chỉ bao gồm chữ viết thường, chữ viết hoa và chữ số.
- + ) Họ, tên của chủ tài khoản chỉ bao gồm chữ viết thường và viết hoa.

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'insertaccount'(FName
    varchar(255),LName varchar(255),TelephoneNum varchar(15),Address
    varchar(255),Birthday date,email varchar(255),passwordd
    varchar(255),Rolen int)
BEGIN
IF((TelephoneNum REGEXP '^0[0-9]+$')is false)
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format error
        (Telephone_number)';
ELSEIF(length(passwordd)<8 or ((passwordd REGEXP '^[a-zA-Z0-9]+$')is false))
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format error
        (Password)';
ELSEIF ((email REGEXP '^[\~@]+\.[\~@]{2,}$')is false)
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format error (Email)';
ELSEIF((FName REGEXP '^[a-zA-Z]+$')is false)
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format error (FName)';
ELSEIF((LName REGEXP '^[a-zA-Z]+$')is false)
    THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format error (LName)';
END IF;
INSERT INTO
    account (FName,LName,TelephoneNum,Address,Birthday,Email>Password,ROLE_NO)
values
    (FName,LName,TelephoneNum,Address,Birthday,email,passwordd,Rolen);
END
```

**Câu lệnh để kiểm tra thủ tục hoạt động:**

Ví dụ ta nhập một tài khoản có FName bao gồm cả chữ số: **vien1**

```
call baitaplon.insertaccount('vien1', 'tri', '0373015428', 'aaa',
    '2000-01-01', 'a@gmail.com', '12345678', 4);
```

Hình ảnh thông báo lỗi hiện ra vì FName không thỏa yêu cầu.

```
1 • call baitaplon.insertaccount('vien1', 'tri', '0373015428', 'aaa', '2000-01-01', 'a@gmail.com', '12345678', 4);  
2
```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
✓ 32	23:17:16	INSERT INTO 'baitaplon'.account ('FName', 'LName', 'TelephoneNum', 'Address...	1 row(s) affected	0.047 sec
✗ 33	23:17:38	INSERT INTO 'baitaplon'.account ('FName', 'LName', 'TelephoneNum', 'Address...	Error Code: 1644. Age is not allowed to be less than 12	0.000 sec
✓ 34	23:48:06	Apply changes to insertaccount	Changes applied	
✗ 35	23:49:08	call baitaplon.insertaccount('vien1', 'tri', '0373015428', 'aaa', '2000-01-01', 'a@gmail....	Error Code: 1644. Format error (FName)	0.032 sec

Nếu ta tiếp tục nhập một password có kí tự đặc biệt: **abc@12345**

```
|| call baitaplon.insertaccount('aaa', 'aaaa', '0123456789', 'abc',  
| '2020-01-01', 'trivien@gmail', 'abc@12345', 4);
```

Hình ảnh thông báo lỗi hiện ra vì Password không thỏa yêu cầu.

```
1 call baitaplon.insertaccount('aaa', 'aaaa', '0123456789', 'abc', '2020-01-01', 'trivien@gmail', 'abc@12345', 4);  
2
```

Output				
Action Output				
#	Time	Action	Message	
✗ 1	00:40:46	call baitaplon.insertaccount('aaa', 'aaaa', '0123456789', 'abc', '2020-01-01', 'trivien@g...	Error Code: 1644. Format error (Password)	

Nếu ta nhập các dữ liệu đều thỏa yêu cầu thì sẽ insert thành công.

```
|| call baitaplon.insertaccount('Vien', 'Tri', '0123456789', 'abc',  
| '2000-01-01', 'trivien@gmail.com', '123456789', 4);
```



```
1 • call baitaplon.insertaccount('Vien', 'Tri', '0123456789', 'abc', '2000-01-01', 'trivien@gmail.com', '123456789', 4);
```

Output				
Action Output				
#	Time	Action	Message	
✖ 1	00:40:46	call baitaplon.insertaccount('aaa', 'aaaa', '0123456789', 'abc', '2020-01-01', 'trivien@g...	Error Code: 1644. Format error (Password)	
✔ 2	00:43:30	call baitaplon.insertaccount('Vien', 'Tri', '0123456789', 'abc', '2000-01-01', 'trivien@gma...	1 row(s) affected	

## Thủ tục 2: Tìm sách theo thể loại

Thủ tục này có tên là `timsachtheotheo`. Chức năng của thủ tục này là để hiển thị dữ liệu về cuốn sách có cùng thể loại mà người dùng nhập vào bao gồm: Tên cuốn sách, Mã cuốn sách (`Product_ID`) và Giá tiền của cuốn sách; được sắp xếp theo thứ tự giá tiền từ thấp đến cao. Nếu người dùng nhập thể loại sách không có trong cửa hàng thì sẽ hiện thông báo lỗi.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `timsachtheotheo`(loaisach
    varchar(255))
BEGIN
    if(loaisach <> 'Truyen tranh' and loaisach <> 'Sach tham khao' and loaisach
        <> 'Sach giao khoa' and loaisach <> 'Sach ky nang song' and loaisach <>
        'Van hoc Viet Nam' and loaisach <> 'Hoc ngoai ngu')
        then SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'The loai sach nay khong
            co trong cua hang';
    else (select P.Name as TenSach, P.Product_ID as MaCuonSach,
        P.Discount_price as Giatien
        from product P, book B, category C
        where P.Product_ID = B.Product_ID and B.CATEG_ID=C.Category_ID and
            lower(C.Name)=lower(loaisach)
        order by P.Discount_price asc
        );
    end if;
END
```

**Câu lệnh để kiểm tra thủ tục hoạt động:** Nếu chúng ta gọi thủ tục và nhập một thể loại không tồn tại, ví dụ: `aaaa`

```
call baitaplon.timsachtheotheo('aaaa');
```

Hình ảnh thông báo lỗi xuất hiện trên màn hình.

```
1 • call baitaplon.timsachtheoLoai('aaaa');
2 |
```

#	Time	Action	Message	Duration
1	00:40:46	call baitaplon.insertaccount('aaa', 'aaaa', '0123456789', 'abc', '2020-01-01', 'trivien@g...	Error Code: 1644. Format error (Password)	0.000 se
2	00:43:30	call baitaplon.insertaccount('Vien', 'Trn', '0123456789', 'abc', '2000-01-01', 'trivien@gma...	1 row(s) affected	0.094 se
3	01:01:51	call baitaplon.timsachtheoLoai('aaaa')	Error Code: 1644. The loại sách này không có trong của hàng	0.016 se

Nếu thể loại sách nhập vào tồn tại thì sẽ hiện ra danh sách các cuốn sách thuộc thể loại ấy.

```
|| call baitaplon.timsachtheoLoai('Sách kỹ năng sống');
```

	TenSach	MaCuonSach	Giatien
▶	Cha giàu cha nghèo	3	69000
	Đắc nhân tâm	2	97200
	Khéo ăn nói sẽ có được thiên hạ	4	113000

### Thủ tục 3: Tìm top những khách hàng có số đơn đặt hàng cao nhất.

Thủ tục này có tên là topkhachhang. Chức năng của thủ tục này là để hiển thị các thông tin như: Mã số khách hàng (Account\_ID), Họ, Tên, Địa chỉ, Tổng số đơn hàng đã hoàn thành và tổng số tiền đã mua sản phẩm (chưa bao gồm phí vận chuyển) từ khách hàng có số đơn đặt hàng thành công cao nhất ở khu vực được đề cập. Nếu ở một khu vực có nhiều khách hàng có cùng số đơn đặt hàng là cao nhất thì sẽ sắp xếp theo thứ tự tổng số tiền sản phẩm mà khách hàng đã mua (chưa bao gồm phí vận chuyển) theo thứ tự giảm dần.

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'topkhachhang'(diachi
    varchar(255))
BEGIN
    SELECT K.Account_ID, K.FName, K.LName, K.Address, Count(*) as
        SoDon, Sum(D.Total_amount) as TongTien
```



```

FROM account K, orders D
WHERE D.Status like '%Completed%' and K.ROLE_NO=4 AND
      K.Account_ID=D.ACC_ID AND LOWER(K.Address)=LOWER(diachi) AND
      D.Total_amount is not null
GROUP BY K.Account_ID, K.FName, K.LName, K.Address
HAVING COUNT(*) IN (SELECT MAX(T.SoDon) as SoDonMax
                     FROM (SELECT COUNT(*) AS SoDon
                           FROM account K, orders D
                               where K.ROLE_NO=4 and K.Account_ID=D.ACC_ID and
                                   D.Total_amount is not null
                                   and LOWER(K.Address)=LOWER(diachi)
                                   group by K.Account_ID) T)
ORDER BY Sum(D.Total_amount) DESC;
END

```

**Câu lệnh để kiểm tra thủ tục hoạt động:** Chúng ta gọi thủ tục và truyền vào khu vực là **Khanh Hoa**

```
|| call baitaplon.topkhachhang('Khanh Hoa');
```

Kết quả hiện thị ra thể hiện rằng khách hàng có Account\_id = 3 là người có số đơn đặt hàng đã hoàn thành cao nhất, với tổng cộng 3 đơn hàng và số tiền đã bỏ ra để mua sản phẩm của cửa hàng là 818000 đ.

	Account_ID	FName	LName	Address	SoDon	TongTien
▶	3	Đào	Lê	Khanh Hoa	3	818000

Ta thử kiểm tra lại những tài khoản có địa chỉ ở Khanh Hoa, sẽ thấy có tổng cộng 3 tài khoản, trong đó có tài khoản có Account\_ID = 3.

	Account_ID	FName	LName	TelephoneNum	Start_date	Address
▶	3	Đào	Lê	0984321486	2022-11-18 10:46:12	Khanh H
	4	Lực	Nguyễn	0312212186	2022-11-18 10:47:49	Khanh H
	16	Nguyễn	Nguyễn	0321219841	2022-11-18 11:09:07	Khanh H
*	NULL	NULL	NULL	NULL	NULL	NULL

Sau đó kiểm tra lại ở bảng orders, ta sẽ thấy tài khoản có Account\_ID = 3 đã mua hàng tổng cộng 3 lần, và tổng số tiền cộng lại đúng bằng 818000. Vì vậy thủ tục đã hoạt động đúng.



	Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID
▶	1	Completed	250000	so 8, hẻm 2, duong 120	NULL	1	2
	2	Completed	99000	số 20, đường Nguyễn L...	NULL	2	2
	3	Completed	120000	123 đường abc	NULL	3	2
	4	Completed	100000	123 đường abc	NULL	4	2
	5	Completed	500000	123 đường abc	NULL	5	2
	6	Completed	200000	123 đường abc	NULL	14	2
	7	Completed	999999	123 đường abc	NULL	15	2
	8	Completed	1200000	123 đường abc	NULL	16	2
	9	Delivering	100000	số 32 đường Nguyễn Du	NULL	6	2
	10	Canceled	200000	số 4 đường Mai Chí Thọ	NULL	7	2
	12	Completed	100000	123 đường abc	NULL	1	2
	13	Completed	100000	số 12 đường An Dương ...	NULL	2	2
	14	Completed	299000	123 đường abc	NULL	3	2
	15	Completed	399000	123 đường abc	NULL	3	2
	16	Completed	390000	123 đường Võ Văn Ngân	NULL	4	2
	17	Canceled	250000	123 đường abc	NULL	14	2
	101	Canceled	200000	123 vô văn ngân	NULL	1	2
	102	Canceled	20000	123 Vô Thị Sáu	NULL	1	2

– **Functions:**

**Hàm 1: Hàm tặng mã giảm giá 20% cho khách hàng**

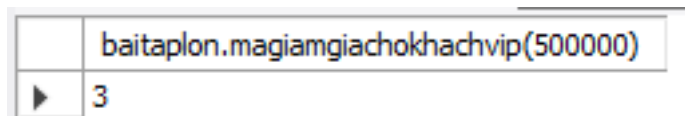
Hàm này có tên là magiamgiachokhachvip. Hàm này có một tham số cần nhập vào là số tiền mà cửa hàng mong muốn. Những khách hàng nào có tổng số tiền mua sản phẩm từ cửa hàng lớn hơn số tiền đã nhập thì sẽ nhận được một mã giảm giá với giá trị 0.2 (20%). Hàm này cũng sẽ trả về tổng số lượng mã giảm giá đã tặng khách hàng.

```
CREATE DEFINER='root'@'localhost' FUNCTION 'new_function'(giatri int)
  RETURNS int
  DETERMINISTIC
BEGIN
  DECLARE v_Found Integer default 1;
  DECLARE idcustomer int;
  DECLARE dem int;
  DECLARE mycur CURSOR FOR SELECT sum(D.Total_amount) as Tongtiensanpham,
    K.Account_ID
    FROM account K, orders D
    WHERE D.Status like '%Completed%' and
    K.Account_ID=D.ACC_ID and K.ROLE_NO=4
    GROUP BY K.Account_ID
    HAVING sum(D.Total_amount)>giatri;
  DECLARE CONTINUE HANDLER FOR NOT FOUND Set v_Found = 0;
  set dem=0;
  open mycur;
  My_Loop : loop
  fetch mycur into idcustomer;
  if v_Found=0 then Leave My_Loop;
  end if;
  INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID) VALUES
    (0.2, 2022-12-12, 'Giam 20%',idcustomer);
  SET dem=dem+1;
  END loop;
  CLOSE mycur;
  RETURN dem;
END
```

**Câu lệnh để kiểm tra hàm hoạt động:** Chúng ta gọi hàm và truyền vào tham số là số tiền mà cửa hàng mong muốn, ví dụ: 500000

```
|| select baitaplon.magiamgiachokhachvip(500000);
```

Sau khi chạy hàm thì sẽ hiển thị ra số mã giảm giá mà cửa hàng đã tặng khách hàng, ở đây là 3 mã.



Kiểm tra lại ở bảng Orders, ta thấy rằng có 3 tài khoản Account\_ID = 3, 15, 16 có tổng cộng số tiền ở các đơn hàng đã đặt lớn hơn 500000.

	Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID
▶	1	Completed	250000	so 8, hẻm 2, đường 120	NULL	1	2
	2	Completed	99000	số 20, đường Nguyễn L...	NULL	2	2
	3	Completed	120000	123 đường abc	NULL	3	2
	4	Completed	100000	123 đường abc	NULL	4	2
	5	Completed	500000	123 đường abc	NULL	5	2
	6	Completed	200000	123 đường abc	NULL	14	2
	7	Completed	999999	123 đường abc	NULL	15	2
	8	Completed	1200000	123 đường abc	NULL	16	2
	9	Delivering	100000	số 32 đường Nguyễn Du	NULL	6	2
	10	Canceled	200000	số 4 đường Mai Chí Thọ	NULL	7	2
	12	Completed	100000	123 đường abc	NULL	1	2
	13	Completed	100000	số 12 đường An Dương ...	NULL	2	2
	14	Completed	299000	123 đường abc	NULL	3	2
	15	Completed	399000	123 đường abc	NULL	3	2
	16	Completed	390000	123 đường Võ Văn Ngân	NULL	4	2
	17	Canceled	250000	123 đường abc	NULL	14	2
	101	Canceled	200000	123 vô văn ngân	NULL	1	2
	102	Canceled	20000	123 Vô Thị Sáu	NULL	1	2

Ta thử kiểm tra lại ở bảng Discount\_code xem liệu đã có 3 mã giảm giá được thêm vào hay chưa, và đây là dữ liệu hiển thị trong bảng Discount\_code:

	Code_ID	Discount	Expiration_date	Name	ACC_ID
▶	20	0.20	2022-12-30	Giảm 20%	3
	21	0.20	2022-12-30	Giảm 20%	15
	22	0.20	2022-12-30	Giảm 20%	16
*	NULL	NULL	NULL	NULL	NULL

## Hàm 2: Hàm tặng mã giảm giá cho khách hàng vào các ngày lễ hội mua sắm:

Hàm này có tên là tangmagiamgia. Vào các ngày mà nhu cầu mua sắm của khách hàng tăng cao hơn so với thường nhật, như ngày lễ Quốc khánh, ngày 11 tháng 11, Giáng sinh ... thì chủ cửa hàng quyết định sẽ tặng một loạt các mã giảm giá có giá trị 15% cho những khách hàng đã từng đặt hàng thành công, để họ có được trải nghiệm mua sắm tốt hơn. Hàm này sẽ trả về tổng số mã giảm giá mà cửa hàng đã dành tặng khách hàng.

Với những vị khách có số đơn đặt hàng thành công trong quá khứ từ 5 đơn trở lên sẽ nhận được 3 mã giảm giá.

Những khách hàng có từ 2 đến 3 đơn đặt hàng thành công sẽ nhận được 2 mã giảm giá.

Những khách hàng có 1 đơn đặt hàng thành công sẽ nhận được 1 mã giảm giá.

```
CREATE DEFINER='root'@'localhost' FUNCTION 'tangmagiamgia'() RETURNS int
DETERMINISTIC
BEGIN
DECLARE done INT DEFAULT FALSE;
DECLARE x int;
DECLARE idcustomer int;
```

```
DECLARE dem int;  
DECLARE mycur CURSOR FOR SELECT K.Account_ID, count(D.ACC_ID) as  
    Tongsodonhang  
        FROM account K, orders D  
        WHERE D.Status like '%Completed%' and  
            K.Account_ID=D.ACC_ID and K.ROLE_NO=4  
        GROUP BY K.Account_ID;  
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;  
set dem = 0;  
open mycur;  
My_Loop : loop  
    fetch mycur into idcustomer, x;  
    if done then Leave My_Loop;  
end if;  
    if(x>=5) then  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
set dem=dem+3;  
    elseif (x>=2 and x<=3) then  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
SET dem=dem+2;  
    elseif (x=1) then  
INSERT INTO discount_code (Discount,Expiration_date,Name,ACC_ID)  
    VALUES (0.15, '2022-12-30', 'Giam 15%',idcustomer);  
set dem=dem+1;  
    end if;  
END loop;  
CLOSE mycur;  
  
RETURN dem;  
END
```

**Câu lệnh để kiểm tra hàm hoạt động:** Trước khi chạy hàm ta sẽ thử chạy câu truy vấn bên dưới để lấy ra Id của các khách hàng và số đơn hàng thành công họ đã đặt.

```
SELECT K.Account_ID, count(D.ACC_ID) as Tongsodonhang  
    FROM account K, orders D  
    WHERE D.Status like '%Completed%' and  
        K.Account_ID=D.ACC_ID and K.ROLE_NO=4  
    GROUP BY K.Account_ID
```

Kết quả của câu truy vấn trên:

	Account_ID	Tongsodonhang
▶	1	2
	2	2
	3	3
	4	2
	5	1
	14	1
	15	1
	16	1

Ở đây ta có thể thấy được có 4 khách hàng với Id lần lượt là: 1,2,3,4 có số đơn đặt hàng từ 2 đến 3 đơn. 4 khách hàng khác với Id lần lượt là: 5,14,15,16 có cùng số đơn đặt hàng là 1. Vậy ta có thể tính được số mã giảm giá mà cửa hàng cần tặng khách hàng là:  $4*2 + 4*1 = 12$  mã.

Tiếp theo ta sẽ thử gọi hàm và xem kết quả trả về:

```
1 • select baitaplon.tangmagiamgia();  
2
```

<	
Result Grid	Filter Rows: <input type="text"/> Export:
	baitaplon.tangmagiamgia()
▶	12

Cũng như trong bảng discount\_code cũng được thêm vào tổng cộng 12 mã tương ứng:

1 • `SELECT * FROM baitaplon.discount_code;`

	Code_ID	Discount	Expiration_date	Name	ACC_ID
▶	20	0.20	2022-12-30	Giam 20%	3
	21	0.20	2022-12-30	Giam 20%	15
	22	0.20	2022-12-30	Giam 20%	16
	23	0.15	2022-12-30	Giam 15%	1
	24	0.15	2022-12-30	Giam 15%	1
	25	0.15	2022-12-30	Giam 15%	2
	26	0.15	2022-12-30	Giam 15%	2
	27	0.15	2022-12-30	Giam 15%	3
	28	0.15	2022-12-30	Giam 15%	3
	29	0.15	2022-12-30	Giam 15%	4
	30	0.15	2022-12-30	Giam 15%	4
	31	0.15	2022-12-30	Giam 15%	5
	32	0.15	2022-12-30	Giam 15%	14
	33	0.15	2022-12-30	Giam 15%	15
	34	0.15	2022-12-30	Giam 15%	16

**Hàm 3: Hàm tính số sản phẩm có điểm đánh giá trung bình đáp ứng yêu cầu:**

Hàm này có tên là sosanpham. Nhằm muốn đánh giá chất lượng chung của sản phẩm trong cửa hàng, chủ cửa hàng mong muốn xem liệu có bao nhiêu sản phẩm trong cửa hàng có số điểm đánh giá trung bình của khách hàng cao hơn một điểm số nhất định. Hàm này nhận đầu vào là một số có kiểu decimal(3,2). Và kết quả trả về là số sản phẩm có điểm trung bình lớn hơn hoặc bằng giá trị truyền vào.

```
CREATE DEFINER='root'@'localhost' FUNCTION 'sosanpham'(giatri decimal(3,2))
RETURNS int
DETERMINISTIC
BEGIN
DECLARE dem int;
```

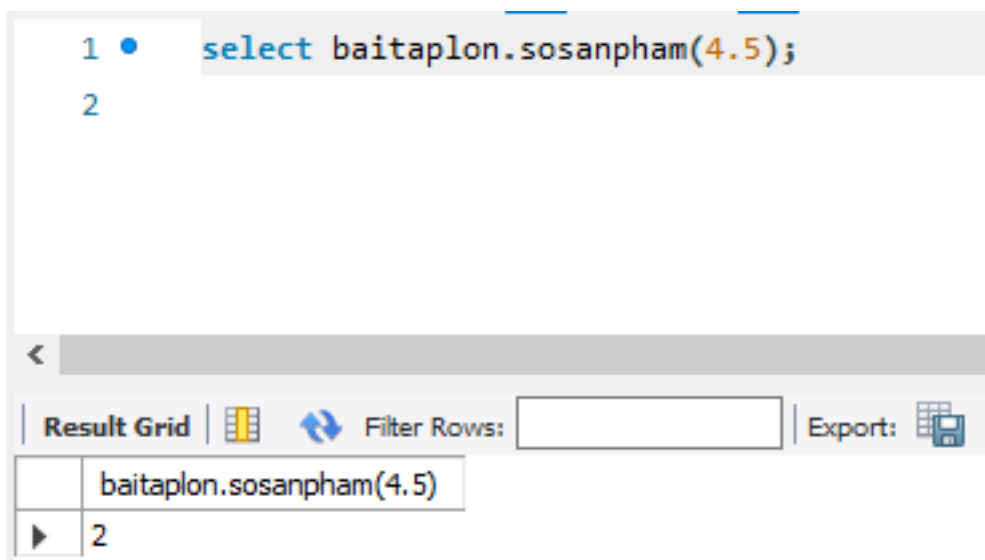
```
set dem=0;
select count(*) into dem from
(select avg(R.Rating), Product_ID
from review R
group by Product_ID
having avg(R.Rating)>=giatri) as a;
return dem;
END
```

**Câu lệnh để kiểm tra hàm hoạt động:** Trước khi chạy hàm, ta sẽ thử chạy câu truy vấn bên dưới để lấy ra ID của các sản phẩm được đánh giá và điểm đánh giá trung bình của nó từ bảng Review. Ở đây ta được 2 sản phẩm: Product\_ID = 1 với rating trung bình là 4.5 và Product\_ID = 2 với rating trung bình là 4.6667.

```
SELECT Product_ID,avg(Rating)
FROM review
group by Product_ID;
```

	Product_ID	avg(Rating)
▶	1	4.5000
	2	4.6667

Bây giờ ta gọi hàm và truyền vào tham số là 4.5 thì kết quả nhận được là 2, vì có 2 sản phẩm có điểm đánh giá từ 4.5 trở lên.



The screenshot shows a SQL query editor with the following code:

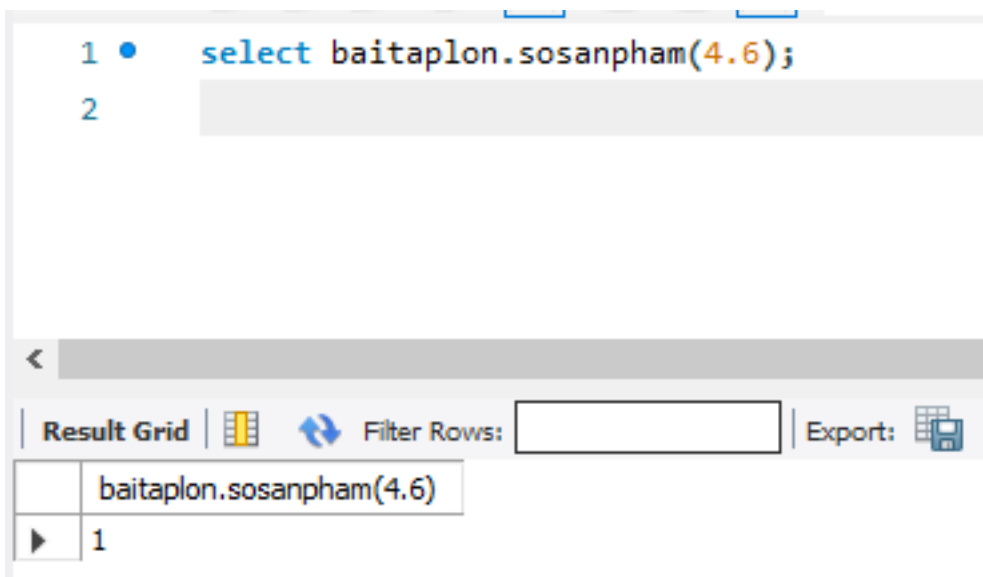
```
1 • select baitaplon.sosanpham(4.5);
2
```

Below the editor is a toolbar with a "Result Grid" button, a "Filter Rows" input field, and an "Export" button. The "Result Grid" is active, displaying the following data:

	baitaplon.sosanpham(4.5)
▶	2

Nếu truyền vào tham số là 4.6 thì chỉ có 1 sản phẩm. Vì chỉ có Product\_ID = 2 có rating trung bình là: 4.6667 > 4.6 .





- Bài làm của Đậu Xuân Thành

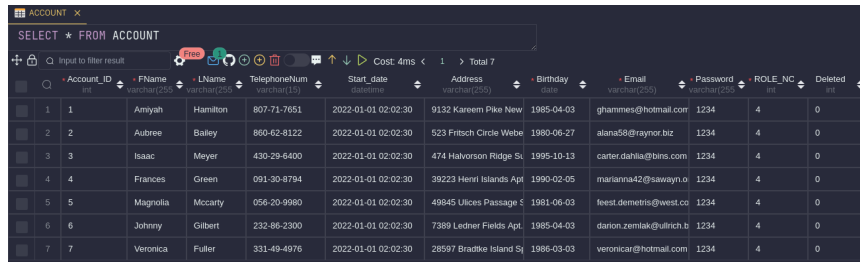
- Trigger:

**Trigger trên bảng Account:** Để khuyến khích người dùng mua sắm, mỗi người dùng sẽ được tặng miễn phí 1 mã giảm giá 10% sau khi tạo tài khoản mới và mã giảm giá này sẽ có hạn sử dụng là 30 ngày.

```
CREATE TRIGGER NEWUSER_DISCOUNT AFTER INSERT ON ACCOUNT
FOR EACH ROW
BEGIN
    DECLARE LAST_ID INTEGER;
    DECLARE EXP_DATE DATE;
    SET @EXP_DATE := DATE_ADD(NEW.Start_date, interval 30 day);
    SELECT COUNT('Code_ID') INTO @LAST_ID FROM DISCOUNT_CODE;
    INSERT INTO DISCOUNT_CODE VALUES (@LAST_ID + 1, 0.1, @EXP_DATE,
        'Discount 10% for new customer', NEW.Account_ID);
END;
DELIMITER;
```

Câu lệnh kiểm tra Trigger hoạt động:

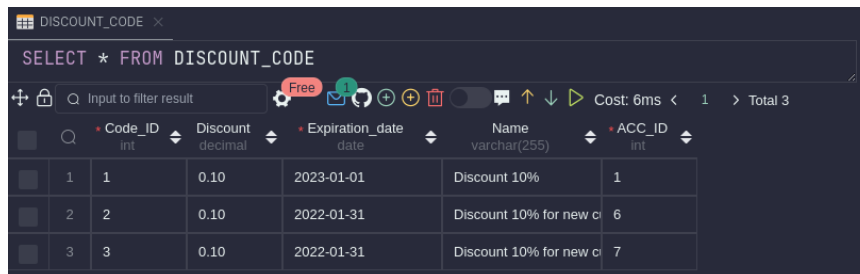
```
INSERT INTO ACCOUNT VALUES ('7','Veronica', 'Fuller', '331-49-4976',
    '2022-01-01 02:02:30', '28597 Bradtke Island Spencerfurt, IL
    93028-7647', '1986-03-03 02:02:30', 'veronicar@hotmail.com', '1234', 4,
    0);
```



Account_ID	FName	LName	TelephoneNum	Start_date	Address	Birthday	Email	Password	ROLE_NC	Deleted
1	Amiyah	Hamilton	807-71-7651	2022-01-01 02:02:30	9132 Kareem Pike New	1985-04-03	ghammes@hotmail.com	1234	4	0
2	Aubree	Bailey	860-62-8122	2022-01-01 02:02:30	523 Fritsch Circle Webe	1980-06-27	alana58@raynor.biz	1234	4	0
3	Isaac	Meyer	430-29-6400	2022-01-01 02:02:30	474 Hakonson Ridge St	1995-10-13	carter.dahlia@bms.com	1234	4	0
4	Frances	Green	091-30-8794	2022-01-01 02:02:30	39223 Henri Islands Apt	1990-02-05	marianna42@sawayn.o	1234	4	0
5	Magnolia	Mccarty	056-20-9980	2022-01-01 02:02:30	49845 Ulices Passage E	1981-06-03	leest.demetris@west.co	1234	4	0
6	Johnny	Gilbert	232-86-2300	2022-01-01 02:02:30	7389 Ledner Fields Apt	1985-04-03	darion.zemlak@vulrich.b	1234	4	0
7	Veronica	Fuller	331-49-4976	2022-01-01 02:02:30	28597 Bradtke Island St	1986-03-03	veronicar@hotmail.com	1234	4	0

Hình 1: Sau khi insert thành công vào bảng ACCOUNT

Sau khi thêm 1 khách hàng có ID = 7 vào table ACCOUNT thì bảng DISCOUNT\_CODE cũng thêm một record có ACC\_ID = 7



Code_ID	Discount	Expiration_date	Name	ACC_ID
1	0.10	2023-01-01	Discount 10%	1
2	0.10	2022-01-31	Discount 10% for new c	6
3	0.10	2022-01-31	Discount 10% for new c	7

Hình 2: Trigger hoạt động - thêm 1 record vào bảng DISCOUNT\_CODE

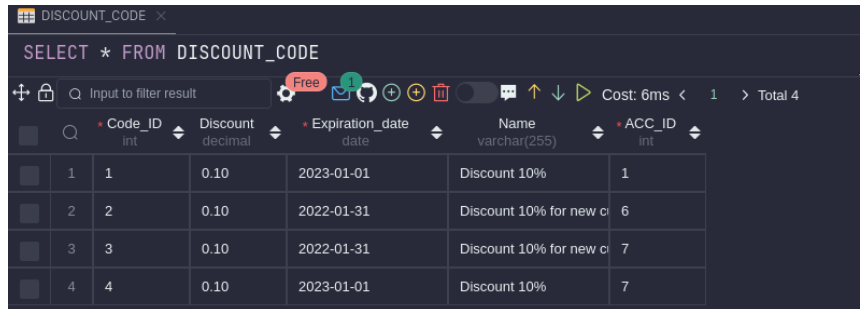
**Trigger trên bảng Account:** Sau khi xóa một khách hàng, thì tất cả các mã giảm giá của người này cũng bị xóa theo.

```
CREATE TRIGGER DELETE_USER_DISCOUNT BEFORE DELETE ON ACCOUNT
FOR EACH ROW
BEGIN
    DELETE FROM 'DISCOUNT_CODE' WHERE 'ACC_ID' = OLD.Account_ID;
END;
DELIMITER;
```

Câu lệnh kiểm tra Trigger hoạt động:

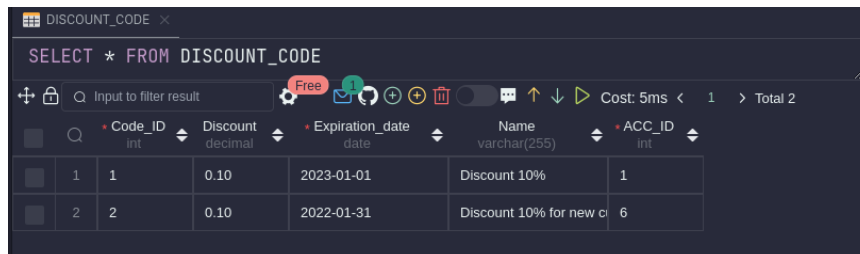
```
DELETE FROM 'ACCOUNT' WHERE 'Account_ID' = 7;
```

Ở đây, chúng ta xóa Account có ID = 7 ở bảng Account



	Code_ID int	Discount decimal	Expiration_date date	Name varchar(255)	ACC_ID int
1	1	0.10	2023-01-01	Discount 10%	1
2	2	0.10	2022-01-31	Discount 10% for new c	6
3	3	0.10	2022-01-31	Discount 10% for new c	7
4	4	0.10	2023-01-01	Discount 10%	7

Hình 3: Trước khi chạy câu lệnh delete ở bảng DISCOUNT



	Code_ID int	Discount decimal	Expiration_date date	Name varchar(255)	ACC_ID int
1	1	0.10	2023-01-01	Discount 10%	1
2	2	0.10	2022-01-31	Discount 10% for new c	6

Hình 4: Sau khi chạy câu lệnh delete ở bảng DISCOUNT

**Trigger trên bảng Product:** Trong trường hợp người thêm dữ liệu thêm nhầm sản phẩm, để giải quyết các ràng buộc tham chiếu về khóa ngoại, chúng ta có thể viết thêm trigger. Khi xóa một sản phẩm ở bảng **Product**, thì các record có tham chiếu đến bảng **Product** cũng bị xóa theo, như là record ở bảng **Book**, **Magazine\_Seri**, **WRITE**,... Như đã nói, đây là phương thức xóa "cứng", tức là xóa sản phẩm khỏi cơ sở dữ liệu vĩnh viễn, nên phải cân nhắc kỹ trước khi xóa sản phẩm.

```
CREATE TRIGGER DELETE_PRODUCT BEFORE DELETE ON
PRODUCT FOR EACH ROW BEGIN
    DELETE FROM 'WRITEE' WHERE 'PRODUCT_ID' = OLD.Product_ID;
    DELETE FROM 'BOOK' WHERE 'Product_ID' = OLD.Product_ID;
    DELETE FROM 'MAGAZINE_SERI' WHERE 'Product_ID' = OLD.Product_ID;
END;
```

Ở đây, giả sử chúng ta đã thêm nhầm sản phẩm có  $ID = 6$  vào cơ sở dữ liệu:

Câu lệnh kiểm tra trigger:

```
DELETE FROM 'PRODUCT' WHERE 'Product_ID' = 6;
```

Ta có thể thấy, dữ liệu của sản phẩm có  $ID = 6$  đã được xóa ra khỏi cả hai bảng Product và & Book, ngoài ra dữ liệu cũng được xóa ra khỏi bảng Write

#### – Procedure:

**Procedure trên bảng Product:** Giá của sản phẩm cần được điều chỉnh để kích cầu mua sắm cho khách hàng, vì vậy chúng ta cần một procedure để cập nhật giá của sản phẩm sau khi được giảm.

```
CREATE PROCEDURE SET_DISCOUNTPRICE(IN PID INT, IN P_FACTOR
```

Q Input to filter result

Cost: 2ms < 1 > Total 6

	Thumbnail varchar(255)	Name varchar(255)	Product_ID int	Price int	Discount_price int	Publisher varchar(255)	Description varchar(500)	Deleted int	Typeproduct int
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê	1	110000	99000	NXB Lao Động	(NULL)	0	(NULL)
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0	(NULL)
3	https://cdn0.fahasa.com	Huyền Thuật Giá Kim	3	139000	118000	Thanh Niên	(NULL)	0	(NULL)
4	https://cdn0.fahasa.com	Hoàng Hôn Đỏ Rực	4	30000	30000	NXB Phụ Nữ	NOT FOUND	0	1
5	https://cdn0.fahasa.com	Trên Đồi Đất Đỏ	5	32000	32000	NXB Kim Đồng	NOT FOUND	0	1
6	https://cdn0.fahasa.com	Nhà Già Kim (Tái Bản 2	6	55	55	NXB Hội Nhà Văn	NOT FOUND	0	1

SELECT \* FROM `BOOK` LIMIT 100;

Q Input to filter result

Cost: 5ms

1 > Total 3

	Product_ID int	Publish_year year	Quantity_in_store int	CATEG_IC int
1	4	2018	10	1
2	5	2019	15	1
3	6	2020	20	1

Hình 5: Dữ liệu ở bảng Product & Book trước khi xóa

PRODUCT

Q Input to filter result

Cost: 2ms < 1 > Total 5

	Thumbnail varchar(255)	Name varchar(255)	Product_ID int	Price int	Discount_price int	Publisher varchar(255)	Description varchar(500)	Deleted int	Typeproduct int
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê	1	110000	99000	NXB Lao Động	(NULL)	0	(NULL)
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0	(NULL)
3	https://cdn0.fahasa.com	Huyền Thuật Giá Kim	3	139000	118000	Thanh Niên	(NULL)	0	(NULL)
4	https://cdn0.fahasa.com	Hoàng Hôn Đỏ Rực	4	30000	30000	NXB Phụ Nữ	NOT FOUND	0	1
5	https://cdn0.fahasa.com	Trên Đồi Đất Đỏ	5	32000	32000	NXB Kim Đồng	NOT FOUND	0	1

SELECT \* FROM `BOOK` LIMIT 100;

Q Input to filter result

Cost: 5ms < 1 > Total 2

	Product_ID int	Publish_year year	Quantity_in_store int	CATEG_IC int
1	4	2018	10	1
2	5	2019	15	1

Hình 6: Dữ liệu ở bảng Product & Book sau khi xóa ở bảng Product

```

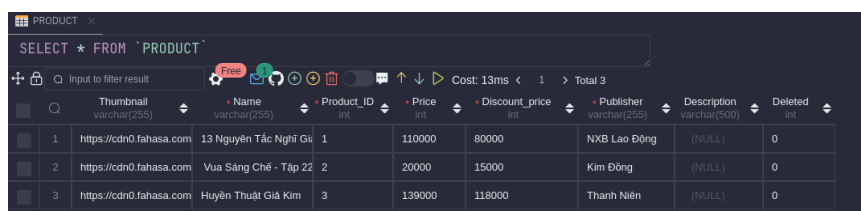
NUMERIC(3, 2)) BEGIN
  IF P_FACTOR < 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'P_FACTOR MUST BE > 0';
  ELSEIF P_FACTOR = 0 THEN
    SIGNAL SQLSTATE '01000'
    SET MESSAGE_TEXT = 'THIS PRODUCT WILL BE FREE';
  ELSEIF P_FACTOR > 1.2 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'DISCOUNT PRICE MUST BE LESS THAN 120% * PRICE';
  END IF;

```

```
UPDATE 'PRODUCT'  
SET  
    'Discount_price' = 'Price' * 'P_Factor'  
WHERE 'Product_ID' = PID;  
END;
```

Ở đây, procedure sẽ tăng hoặc giảm giá trên giá trị thực của sản phẩm, ví dụ sản phẩm X có giá thực là 110.000 VND thì sau khi giảm 10% thì ta có giá trị Discount\_price sẽ là 99.000 VND. Tuy nhiên, ở đây có ràng buộc ràng sản phẩm không được giảm giá xuống dưới 0VND và không được tăng giá quá 20% giá trị thực của sản phẩm.

Ví dụ: Chúng ta chạy câu lệnh kiểm tra procedure, bằng cách giảm giá 10% cho sản phẩm có ID = 1:

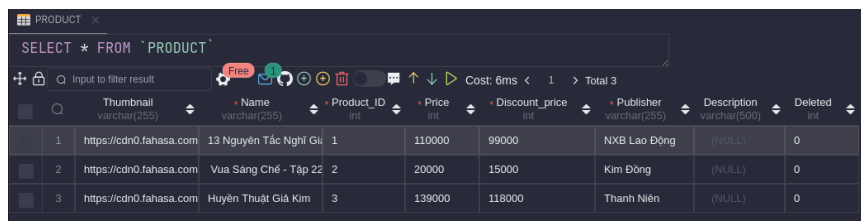


	Thumbnail	Name	Product_ID	Price	Discount_price	Publisher	Description	Deleted
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê Gi	1	110000	80000	NXB Lao Động	(NULL)	0
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0
3	https://cdn0.fahasa.com	Huyền Thuật Giả Kim	3	139000	118000	Thanh Niên	(NULL)	0

Hình 7: Trước khi chạy procedure

```
|| CALL 'SET_DISCOUNTPRICE'(1, 0.9);
```

Kết quả:



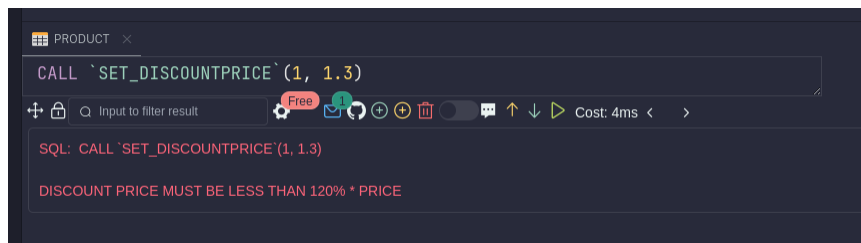
	Thumbnail	Name	Product_ID	Price	Discount_price	Publisher	Description	Deleted
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê Gi	1	110000	99000	NXB Lao Động	(NULL)	0
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0
3	https://cdn0.fahasa.com	Huyền Thuật Giả Kim	3	139000	118000	Thanh Niên	(NULL)	0

Hình 8: Sau khi chạy procedure

Giả sử khi giá sản phẩm bị tăng lên 30% so với giá trị thực:

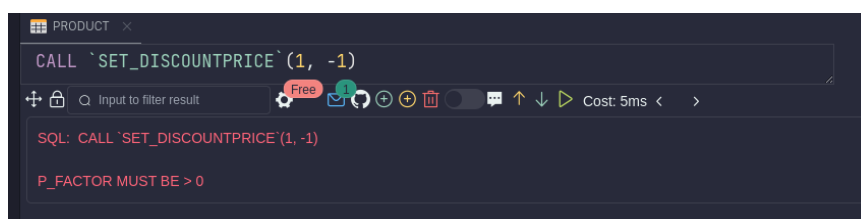
```
|| CALL 'SET_DISCOUNTPRICE'(1, 1.3);
```

Thì procedure sẽ báo lỗi cho người dùng biết:



Hình 9: Procedure báo lỗi khi tăng giá sản phẩm lên quá 20%

```
|| CALL 'SET_DISCOUNTPRICE'(1, -1);
```



Hình 10: Procedure báo lỗi khi giảm giá sản phẩm xuống dưới 0VND

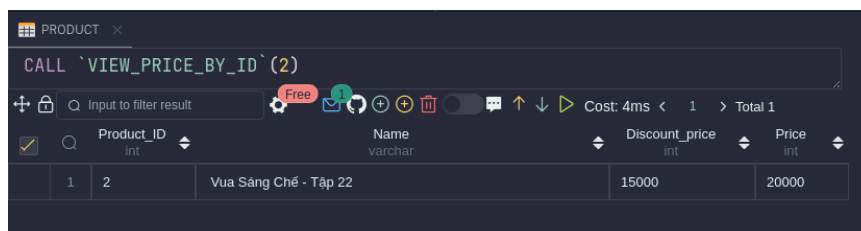
- Procedure cho phép xem giá hiện tại của sản phẩm và giá thực của sản phẩm thông qua ID của sản phẩm

```
CREATE PROCEDURE VIEW_PRICE_BY_ID(IN P_ID INT)
BEGIN
    DECLARE COUNT_ID INT;
    DECLARE _Message VARCHAR(200);
    SET @_Message := CONCAT ('Not have product with id = ', CAST(P_ID AS
        VARCHAR(20)));
    SELECT COUNT(*) INTO @COUNT_ID FROM 'PRODUCT' WHERE 'Product_ID' = P_ID;
    IF (@COUNT_ID = 1) THEN
        SELECT 'Product_ID', 'Name', 'Discount_price', 'Price' FROM PRODUCT
            WHERE 'Product_ID' = P_ID;
    ELSE
        SIGNAL SQLSTATE '02200'
        SET MESSAGE_TEXT = @_Message;
    END IF;
END;
```

Procedure sẽ trả về một bảng có thông tin bao gồm ID của sản phẩm, Tên của sản phẩm, Giá trị thực và giá sau khi đã được giảm. Nếu không tìm thấy sản phẩm có ID trùng với sản phẩm đang tìm, thì sẽ hiển thị ra một message để cảnh báo cho người dùng.

Kiểm tra:

```
|| CALL 'VIEW_PRICE_BY_ID'(2);
```

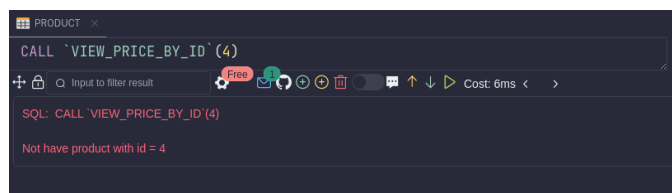


Product_ID	Name	Discount_price	Price
2	Vua Sáng Chế - Tập 22	15000	20000

Hình 11: Kết quả sau khi chạy procedure

Chúng ta thử tìm xem với một sản phẩm có **ID = 4** (đang không có trong cơ sở dữ liệu)

```
|| CALL 'VIEW_PRICE_BY_ID'(4);
```



```
SQL: CALL 'VIEW_PRICE_BY_ID'(4)
Not have product with id = 4
```

Hình 12: Kết quả sau khi chạy procedure

Kết quả hiển thị ra một lỗi: **Not have product with id = 4**, vậy procedure đã thực hiện thành công.

#### – Procedure cho phép thêm sách

Ở đây, để đồng bộ trong cơ sở dữ liệu, chúng ta sẽ sử dụng stored procedure để thêm sách vào cơ sở dữ liệu.

```
CREATE PROCEDURE INSERT_BOOK(IN THUMBNAİL VARCHAR(255)
), IN P_NAME VARCHAR(255), IN P_ID INT, IN P_PRICE
INT, IN P_PUBLISHER VARCHAR(255), IN P_DES VARCHAR
(500), IN P_YEAR YEAR, IN Q INT, IN CATE INT, IN AUTH_ID
INT) BEGIN
    DECLARE _Message VARCHAR(200);
    DECLARE COUNT_ID INT;
    --Check condition
    SET
        @_Message := CONCAT (
            'Not have author with id = ',
            CAST(AUTH_ID AS VARCHAR(20))
        );
    SELECT
        COUNT(*) INTO @COUNT_ID
    FROM 'AUTHOR'
    WHERE 'Author_ID' = AUTH_ID;
    IF (@COUNT_ID != 1) THEN
        SIGNAL SQLSTATE '02200' SET MESSAGE_TEXT = @_Message;
    END IF;

    ---insert
    INSERT INTO
        'PRODUCT'
    VALUES (
        THUMBNAİL,
        P_NAME,
        P_ID,
        P_PRICE,
        P_PRICE,
        P_PUBLISHER,
        P_DES,
        0,
        1
    );
    INSERT INTO 'BOOK' VALUES (P_ID, P_YEAR, Q, CATE);
    INSERT INTO 'WRITEE'
    VALUES (AUTH_ID, P_ID);
END;
```

#### Tham số:

Procedure nhận vào tất cả các tham số của **Product**, và thêm các tham số đặc biệt của riêng **Book** như ID của tác giả, năm xuất bản, số lượng thêm. Trong đó, procedure cũng kiểm tra các điều kiện về khóa ngoại, như kiểm tra và sẽ thông báo nếu **ID** của tác giả chưa có trong hệ thống.

#### Các bảng ảnh hưởng:



Khi chạy procedure này, các bảng liên quan đến **Product** sẽ ảnh hưởng, như: **WRITE**, **BOOK**

Kiểm tra:

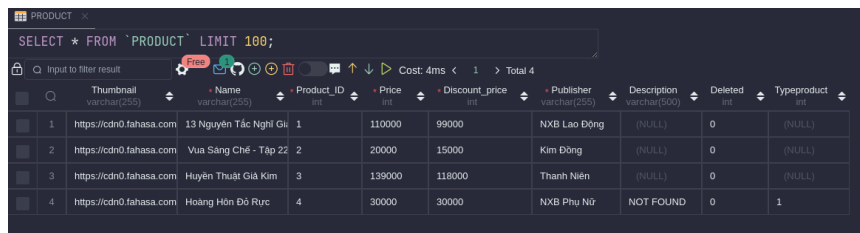
```
CALL 'INSERT_BOOK' (
    'https://cdn0.fahasa.com/media/catalog/product/i/m/image_195509_1_9622.jpg',
    'Hoang Hon Do Ruc',
    4,
    30000,
    'NXB Phu Nu',
    'NOT FOUND',
    '2018',
    10,
    1,
    1
);
```

Khi ta thêm thông tin trên, hệ thống hiển thị thông báo thành công:

```
CALL 'INSERT_BOOK' ( 'https://cdn0.fahasa.com/media/catalog/product/i/m/image_195509_1_9622.jpg', "Hoàng Hôn Đỏ Rục", 4, 30000, 'NXB Phụ Nữ', 'NOT FOUND', '2018', 10, 1, 1 )

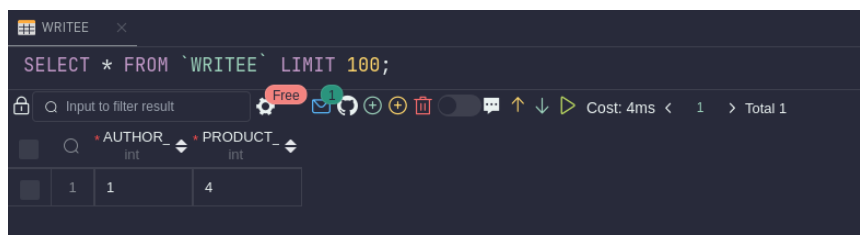
AffectedRows : 4
```

Hình 13: Kết quả sau khi chạy procedure



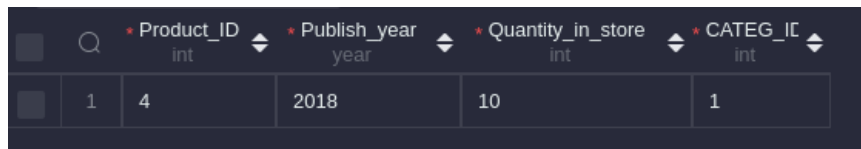
	Thumbnail varchar(255)	Name varchar(255)	Product_ID int	Price int	Discount_price int	Publisher varchar(255)	Description varchar(255)	Deleted int	Typeproduct int
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghi Gi	1	110000	99000	NXB Lao Động	(NULL)	0	(NULL)
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0	(NULL)
3	https://cdn0.fahasa.com	Huyền Thuật Giá Kim	3	139000	118000	Thanh Niên	(NULL)	0	(NULL)
4	https://cdn0.fahasa.com	Hoàng Hôn Đỏ Rục	4	30000	30000	NXB Phụ Nữ	NOT FOUND	0	1

Hình 14: Kết quả sau khi chạy procedure trên bảng Product



	AUTHOR_ int	PRODUCT_ int
1	1	4

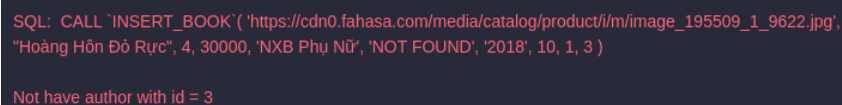
Hình 15: Kết quả sau khi chạy procedure trên bảng WRITE



	* Product_ID int	* Publish_year year	* Quantity_in_store int	* CATEG_IC int
	1	4	2018	10
				1

Hình 16: Kết quả sau khi chạy procedure trên bảng BOOK

Để chắc chắn procedure chạy thành công, chúng ta thêm một thông tin với ID của tác giả là **5** (hiện không có trong database)



```
SQL: CALL 'INSERT_BOOK' ( 'https://cdn0.fahasa.com/media/catalog/product/i/m/image_195509_1_9622.jpg',  
"Hoàng Hôn Đỏ Rực", 4, 30000, 'NXB Phụ Nữ', 'NOT FOUND', '2018', 10, 1, 3 )  
  
Not have author with id = 3
```

Hình 17: Kết quả sau khi chạy procedure khi không có tác giả có id phù hợp

Như ta đã thấy, hệ thống hiển thị lỗi **"Not have author with id = 5"**

#### – Procedure cho phép thêm Magazine seri

Tương tự như thêm sách, ta có thể hiển thị được procedure cho phép thêm magazine seri.

- **Function:**

- **Function** cho phép xem tổng số lượng sách theo thể loại.

Để quản lý, cũng như cân bằng số lượng sách để bổ sung kịp thời cho phù hợp với nhu cầu của người dùng.

```
CREATE FUNCTION SL_BOOK_BY_CATE(CATE VARCHAR(255))
RETURNS INT BEGIN
    DECLARE SL INT;
    SELECT
        SUM(BOOK.Quantity_in_store) INTO SL
    FROM BOOK, CATEGORY
    WHERE
        CATEG_ID = Category_ID
        AND CATEGORY.Name = CATE;
    RETURN SL;
END;
```

Câu lệnh kiểm tra:

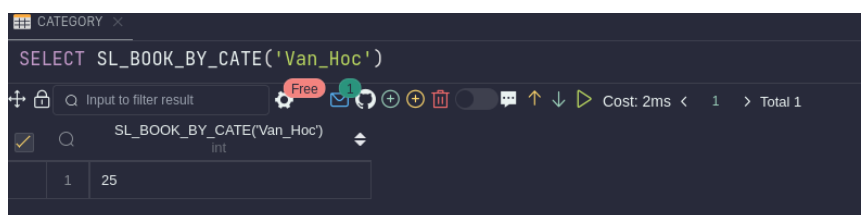
```
SELECT SL_BOOK_BY_CATE('Van_Hoc');
```



Product_ID	Publish_year	Quantity_in_store	CATEG_ID
1	4	2018	10
2	5	2019	15

Hình 18: Bảng **Book**.

Bảng **Book** có 10 cuốn có ID = 4 và 15 cuốn có ID = 5, 2 cuốn này đều có cùng thể loại là '**Van\_hoc**'.



SL_BOOK_BY_CATE('Van_Hoc')
25

Hình 19: Kết quả.

Hàm trả về số lượng là 25 cuốn cho thể loại '**Van\_hoc**'.

- **Function** cho phép điều chỉnh giá các loại sách.

Để kích cầu đối với một loại sách bất kỳ, đáp ứng kịp thời nhu cầu của khách hàng, cũng như để tăng doanh thu cho cửa hàng. Những thể loại sách mà người dùng ưa chuộng có thể được cửa hàng giảm giá, cũng như những loại sách mà có giá quá cao cũng cần được giảm giá. Hoặc trong một số đợt mua sắm, các loại sách bất kỳ cũng

cần điều chỉnh giá, ví dụ như vào thời điểm bắt đầu năm học mới, nhu cầu mua sách Giáo Khoa tăng cao – cửa hàng có thể điều chỉnh giá cho phù hợp với nhu cầu của khách hàng.

**Tham số đầu vào:** Hàm nhận vào ID của loại sách cần điều chỉnh giá, một biến **factor** cho biết tỷ lệ cần điều chỉnh so với giá gốc.

**Kết quả trả về:** Hàm trả về True (1) nếu điều chỉnh thành công và False (0) nếu không thành công.

```
CREATE FUNCTION AdjustPrice_BY_CATE (Cate_ID INT, D_FACTOR
NUMERIC(3, 2))
RETURNS BOOL BEGIN
    IF D_FACTOR < 0 THEN SIGNAL SQLSTATE '45000'
    SET
        MESSAGE_TEXT = 'D_FACTOR MUST BE > 0';
    RETURN FALSE;
    ELSEIF D_FACTOR = 0 THEN SIGNAL SQLSTATE '01000'
    SET
        MESSAGE_TEXT = 'THIS PRODUCT WILL BE FREE';
    ELSEIF D_FACTOR > 1.2 THEN SIGNAL SQLSTATE '45000'
    SET
        MESSAGE_TEXT = 'DISCOUNT PRICE MUST BE LESS THAN 120% * PRICE';
    RETURN FALSE;
    END IF;
    UPDATE 'PRODUCT'
    SET
        'Discount_price' = 'Price' * 'D_Factor'
    WHERE 'Product_ID' IN (SELECT 'Product_ID' FROM 'BOOK' WHERE 'CATEG_ID'
        = Cate_ID);
    RETURN TRUE;
END;
```

**Kiểm tra hoạt động của hàm:** Ở bảng Book, ta thấy được các sách có ID từ 9 – 13 có cùng thể loại có ID = 3 – "Tam\_ly". Giả sử ta muốn điều chỉnh giá tất cả các

	Product_ID int	Publish_year year	Quantity_in_store int	CATEG_ID int
1	4	2018	10	1
2	5	2019	15	1
3	6	2020	20	1
4	7	2017	100	1
5	8	2020	200	1
6	9	2022	150	3
7	10	2021	100	3
8	11	2004	50	3
9	12	2018	120	3
10	13	2019	200	3

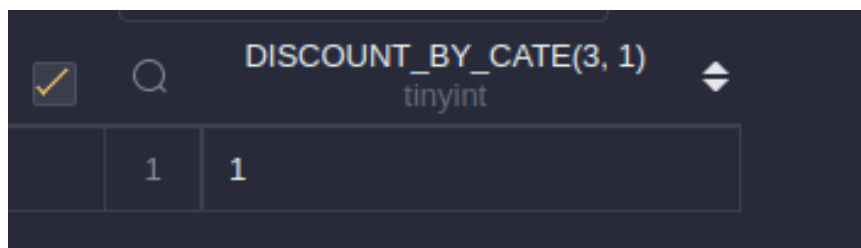
Hình 20: Bảng Book

cuốn sách có thể loại là "Tam\_ly" giảm xuống 10% ta sẽ chạy câu lệnh sau:

	Thumbnail varchar(255)	Name varchar(255)	Product_ID int	Price int	Discount_price int	Publisher varchar(255)	Description varchar(200)	Deleted int	Typeproduct int
1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê	1	110000	99000	NXB Lao Động	(NULL)	0	(NULL)
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0	(NULL)
3	https://cdn0.fahasa.com	Huyền Thuật Giá Kim	3	139000	118000	Thanh Niên	(NULL)	0	(NULL)
4	https://cdn0.fahasa.com	Hoàng Hôn Đỏ Rục	4	30000	30000	NXB Phụ Nữ	NOT FOUND	0	1
5	https://cdn0.fahasa.com	Trên Đồi Đất Đỏ	5	32000	32000	NXB Kim Đồng	NOT FOUND	0	1
6	https://cdn0.fahasa.com	Nhà Giả Kim (Tái Bản 2	6	55000	55000	NXB Hội Nhà Văn	NOT FOUND	0	1
7	https://cdn0.fahasa.com	Trí Khung Tự Truyền	7	72000	72000	NXB Hội Nhà Văn	NOT FOUND	0	1
8	https://cdn0.fahasa.com	Ma Bùn, Lưu Manh Va	8	85000	85000	NXB Trẻ	NOT FOUND	0	1
9	https://cdn0.fahasa.com	Đắc Nhân Tâm	9	68000	68000	NXB Đại Học Quốc Gia	NOT FOUND	0	1
10	https://cdn0.fahasa.com	Quảng Gánh Lo Đì & Vù	10	108000	108000	NXB Tổng Hợp TPHCM	NOT FOUND	0	1
11	https://cdn0.fahasa.com	How to Enjoy Your Life	11	179000	179000	Ebury Press	NOT FOUND	0	1
12	https://cdn0.fahasa.com	Chuẩn Bị Để Thắng Tiế	12	145000	145000	NXB Lao Động	NOT FOUND	0	1
13	https://cdn0.fahasa.com	Nghệ Thuật Nói Trước	13	85000	85000	NXB Dân Trí	NOT FOUND	0	1

Hình 21: Giá ban đầu của các sản phẩm ở bảng Product

```
|| SELECT AdjustPrice_BY_CATE(3, 0.9);
```



Hình 22: Hàm trả về 1 (True) – Thành công

Sau đó, ta kiểm tra giá sau khi điều chỉnh:

1	https://cdn0.fahasa.com	13 Nguyễn Tác Nghiê	1	110000	99000	NXB Lao Động	(NULL)	0	(NULL)
2	https://cdn0.fahasa.com	Vua Sáng Chế - Tập 22	2	20000	15000	Kim Đồng	(NULL)	0	(NULL)
3	https://cdn0.fahasa.com	Huyền Thuật Giá Kim	3	139000	118000	Thanh Niên	(NULL)	0	(NULL)
4	https://cdn0.fahasa.com	Hoàng Hôn Đỏ Rục	4	30000	30000	NXB Phụ Nữ	NOT FOUND	0	1
5	https://cdn0.fahasa.com	Trên Đồi Đất Đỏ	5	32000	32000	NXB Kim Đồng	NOT FOUND	0	1
6	https://cdn0.fahasa.com	Nhà Giả Kim (Tái Bản 2	6	55000	55000	NXB Hội Nhà Văn	NOT FOUND	0	1
7	https://cdn0.fahasa.com	Trí Khung Tự Truyền	7	72000	72000	NXB Hội Nhà Văn	NOT FOUND	0	1
8	https://cdn0.fahasa.com	Ma Bùn, Lưu Manh Va	8	85000	85000	NXB Trẻ	NOT FOUND	0	1
9	https://cdn0.fahasa.com	Đắc Nhân Tâm	9	68000	61200	NXB Đại Học Quốc Gia	NOT FOUND	0	1
10	https://cdn0.fahasa.com	Quảng Gánh Lo Đì & Vù	10	108000	97200	NXB Tổng Hợp TPHCM	NOT FOUND	0	1
11	https://cdn0.fahasa.com	How to Enjoy Your Life	11	179000	161100	Ebury Press	NOT FOUND	0	1
12	https://cdn0.fahasa.com	Chuẩn Bị Để Thắng Tiế	12	145000	130500	NXB Lao Động	NOT FOUND	0	1
13	https://cdn0.fahasa.com	Nghệ Thuật Nói Trước	13	85000	76500	NXB Dân Trí	NOT FOUND	0	1

Hình 23: Giá sau khi được điều chỉnh

Ta có thể thấy giá của tất cả các sản phẩm có thể loại "Tam\_ly" đã được giảm thành công.

### 3 Định nghĩa các nhóm người dùng và phân quyền ở mức ứng dụng

Nhóm chúng em sẽ phân chia người dùng bên phía ứng dụng thành 4 nhóm, đó là: quản lý, nhân viên thường, nhân viên kho, khách hàng.

Ở bảng Role sẽ bao gồm 2 thuộc tính đó là Role\_ID và Role\_name. Với Role\_ID là ID của nhóm người dùng, Role\_name là tên nhóm người dùng.

	Role_ID	Role_name
▶	1	quanly
	2	nhanvienthuong
	3	nhanvienkho
	4	khachhang
✖	NULL	NULL

Quyền hạn	Khách hàng	Nhân viên kho	Nhân viên thường	Quản lý
Xem sản phẩm của cửa hàng	x	x	x	x
Tạo và chỉnh sửa tk khách hàng	x			x
Tạo và chỉnh sửa tk nhân viên thường/ kho				x
Thêm, xóa sản phẩm		x		x
Đặt hàng	x			
Xử lý các đơn hàng			x	x
Xem thông tin tất cả đơn hàng			x	x

## 4 Định nghĩa kiến trúc cho ứng dụng

Sau khi thảo luận, kiến trúc mà nhóm sử dụng để thiết kế cho ứng dụng là sử dụng mô hình MVC. Nhưng trong quá trình làm việc gặp nhiều vấn đề nên kiến trúc ứng dụng của nhóm không hoàn toàn tuân thủ theo mô hình MVC.

### 4.1 Mô hình MVC

MVC là viết tắt của cụm từ “**Model-View-Controller**“. Là viết tắt của cụm "Model-View-Controller". Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

- **Model (dữ liệu):** Quản lý xử lý các dữ liệu.
- **View (giao diện):** Nơi hiển thị giao diện người dùng.
- **Controller (bộ điều khiển):** Điều khiển sự tương tác của hai thành phần Model và View.

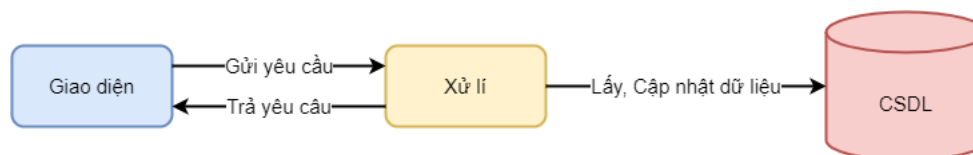
### 4.2 Kiến trúc ứng dụng thực tế của nhóm

**Mô tả:** Kiến trúc ứng dụng gồm có 3 phần chính là Giao diện, Xử lý, Cơ sở dữ liệu. Giao diện đóng vai trò hiển thị và gửi các yêu cầu của người dùng qua phương thức POST. Phần xử lý đóng vai trò tiếp nhận thông tin từ giao diện và xử lý yêu cầu của người dùng, lấy và cập nhật dữ liệu cơ sở dữ liệu. Cơ sở dữ liệu đóng vai trò lưu trữ toàn bộ dữ liệu. Mỗi trang trong ứng dụng đều có 2 phần là hiển thị và xử lý.

- **View (giao diện):** Nơi hiển thị giao diện người dùng.
- **Model + Controller (Xử lý):** Phần xử lý yêu cầu của người dùng.
- **Database (Cơ sở dữ liệu):** Chứa toàn bộ dữ liệu của ứng dụng.

### 4.3 Minh họa kiến trúc ứng dụng

- Sơ đồ minh họa kiến trúc ứng dụng



- Minh họa kiến trúc ứng dụng với trang shop.php, là trang hiển thị toàn bộ sản phẩm của cửa hàng.
  - Phần đầu của trang (phần xử lý): Gồm các câu lệnh SQL để lấy và cập nhật dữ liệu trên Database.

```
1 <?php
2
3 include 'config.php';
4
5 session_start();
6
7 $user_id = $_SESSION['user_id'];
8
9 if(!isset($user_id)){
10     header('location:login.php');
11 }
12
13
14 if(isset($_POST['add_to_cart'])){
15
16     $product_name = $_POST['product_name'];
17     $product_price = $_POST['product_price'];
18     $product_image = $_POST['product_image'];
19     $product_quantity = $_POST['product_quantity'];
20     $product_id = $_POST['product_id'];
21
22     $check_cart_numbers = mysqli_query($conn, "SELECT * FROM 'cart' WHERE name = '$product_name' AND user_id = '$user_id'" ) or die('query failed');
23
24     if(mysqli_num_rows($check_cart_numbers) > 0){
25         $message[] = 'already added to cart!';
26     }else{
27         mysqli_query($conn, "INSERT INTO 'cart' (user_id,Product_ID, name, price, quantity, image) VALUES('$user_id','$product_id', '$product_name', '$product_price', '$product_quantity', '$product_image')") or die('query failed');
28         $message[] = 'product added to cart!';
29     }
30 }
31
32 ?>
```

- Phần thứ hai của trang (phần hiển thị): Gồm các mã HTML, CSS và phương thức POST. Tuy nhiên cũng có một vài câu lệnh SQL để truy cập vào Database nhằm lấy thông tin hiển thị.

```
<section class="products">
<h1 class="title">latest products</h1>
<div class="box-container">
<?php
$select_products = mysqli_query($conn, "SELECT * FROM 'product'") or die('query failed');
if(mysqli_num_rows($select_products) > 0){
    while($fetch_products = mysqli_fetch_assoc($select_products)){
    }
<form action="" method="post" class="box">

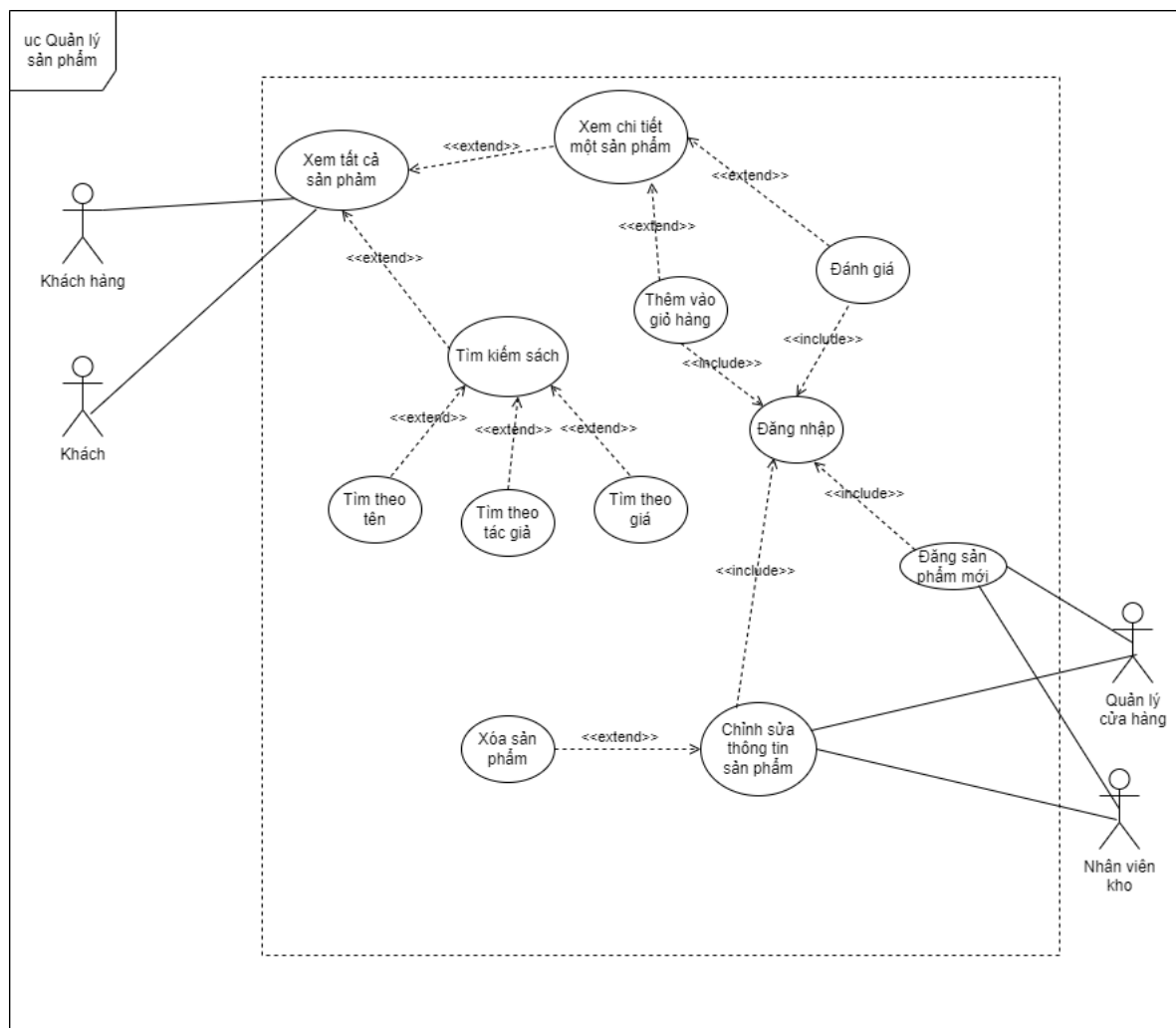
<div class="name"><?php echo $fetch_products['Name']; ?></div>
<div class="price"><?php
    $price=$fetch_products['Price'];
    $discount_price=$fetch_products['Discount_price'];
    if($discount_price) {
        echo "<s style='text-decoration: line-through'>$.price.</s>";
        echo " | $<?php echo $discount_price;
    }
    else {
        echo "$<?php echo $price;
    }
    ?></div>
<input type="number" min="1" name="product_quantity" value="1" class="qty">
<input type="hidden" name="product_name" value="<?php echo $fetch_products['Name']; ?>">
<input type="hidden" name="product_price" value="<?php echo $fetch_products['Price']; ?>">
    $price=$fetch_products['Price'];
    $discount_price=$fetch_products['Discount_price'];
    if($discount_price) {
        echo $discount_price;
    }
    else {
        echo $price;
    }
    ?>">
<input type="hidden" name="product_image" value="<?php echo $fetch_products['Thumbnail']; ?>">
<input type="hidden" name="product_id" value="<?php echo $fetch_products['Product_ID']; ?>">
<input type="submit" value="add to cart" name="add_to_cart" class="btn">
```

## 5 Thiết kế chi tiết cho ứng dụng

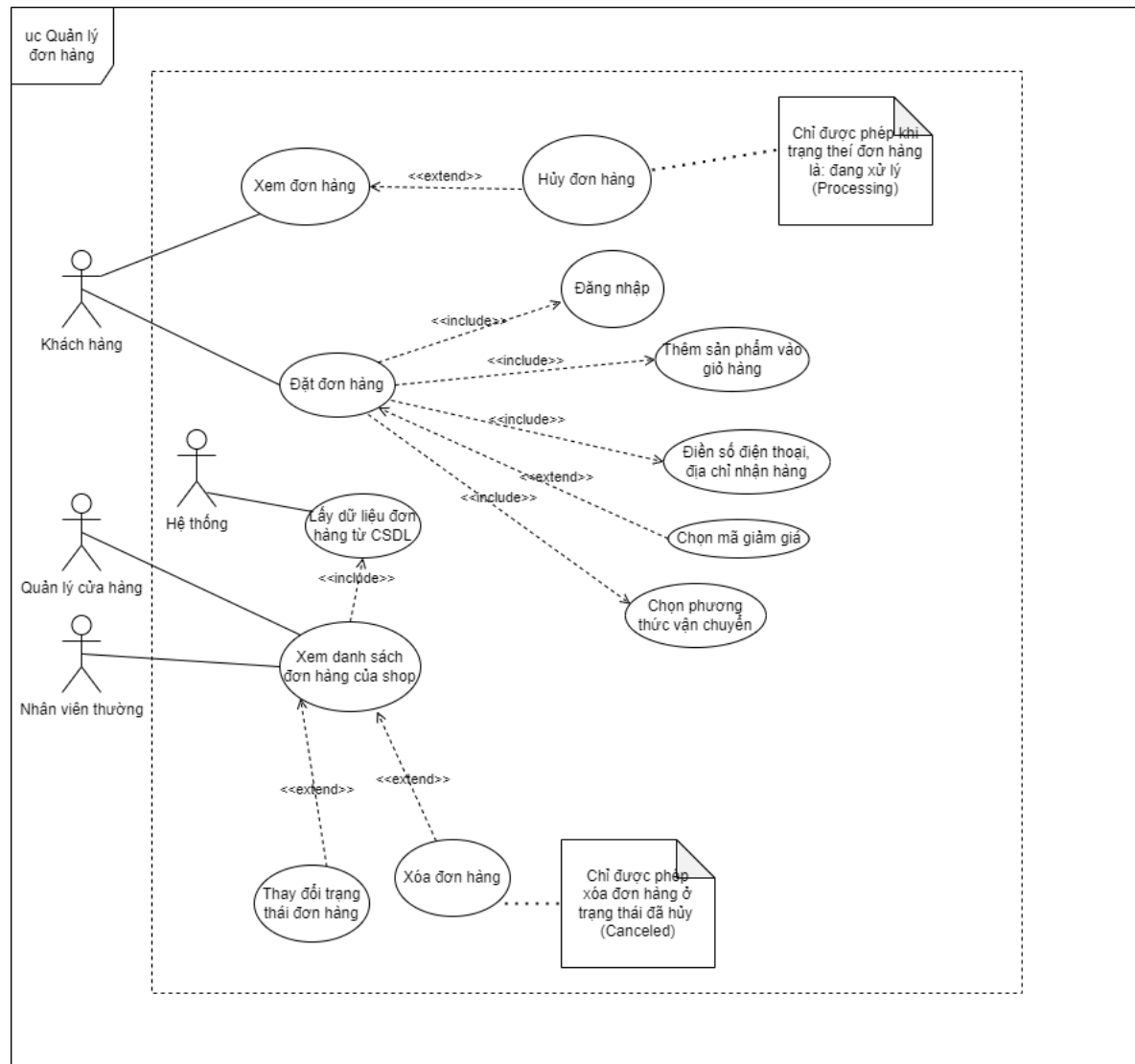
### Usecase diagram:

Sơ đồ usecase Quản lý sản phẩm.



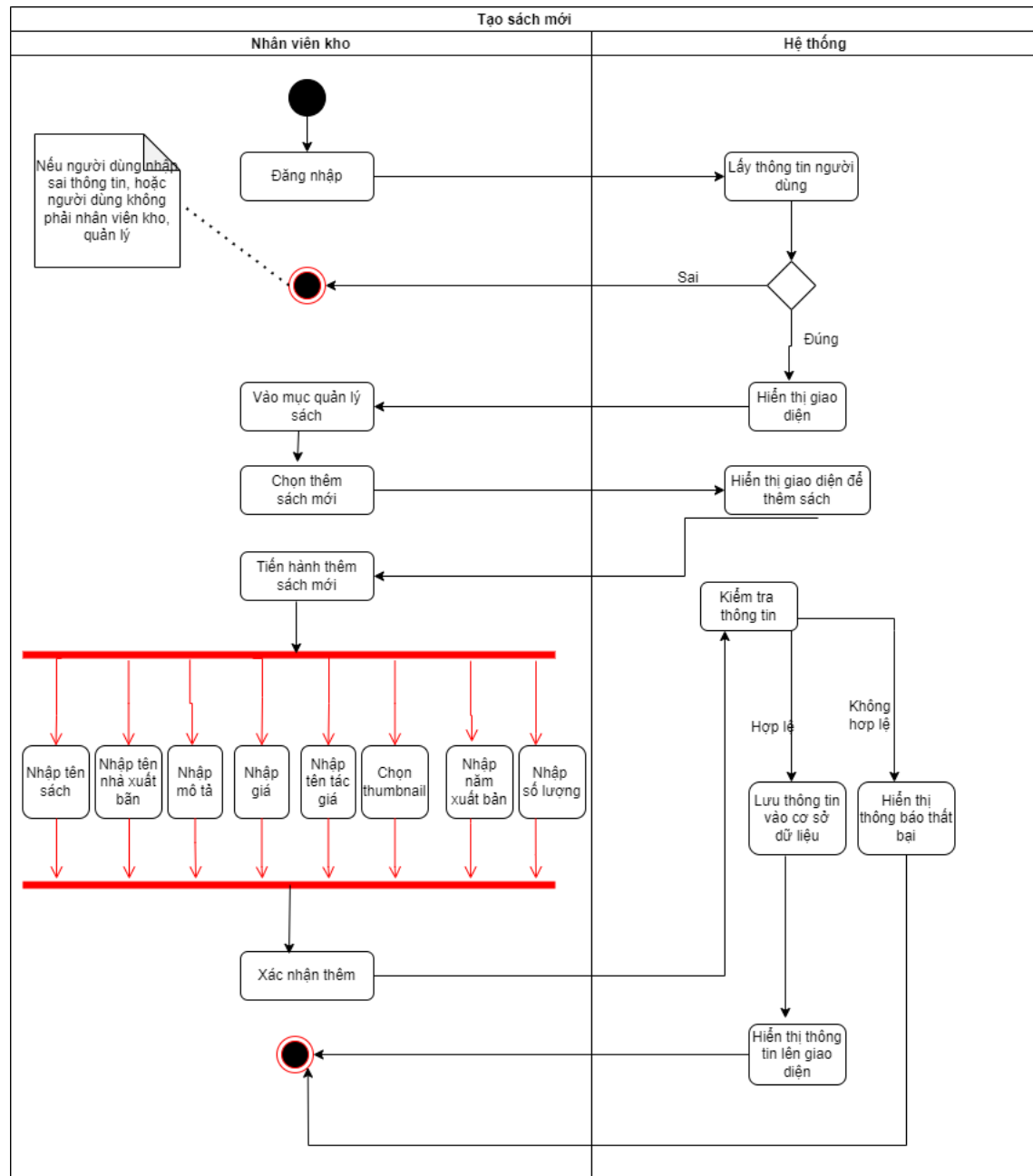


Sơ đồ usecase Quản lý đơn hàng.

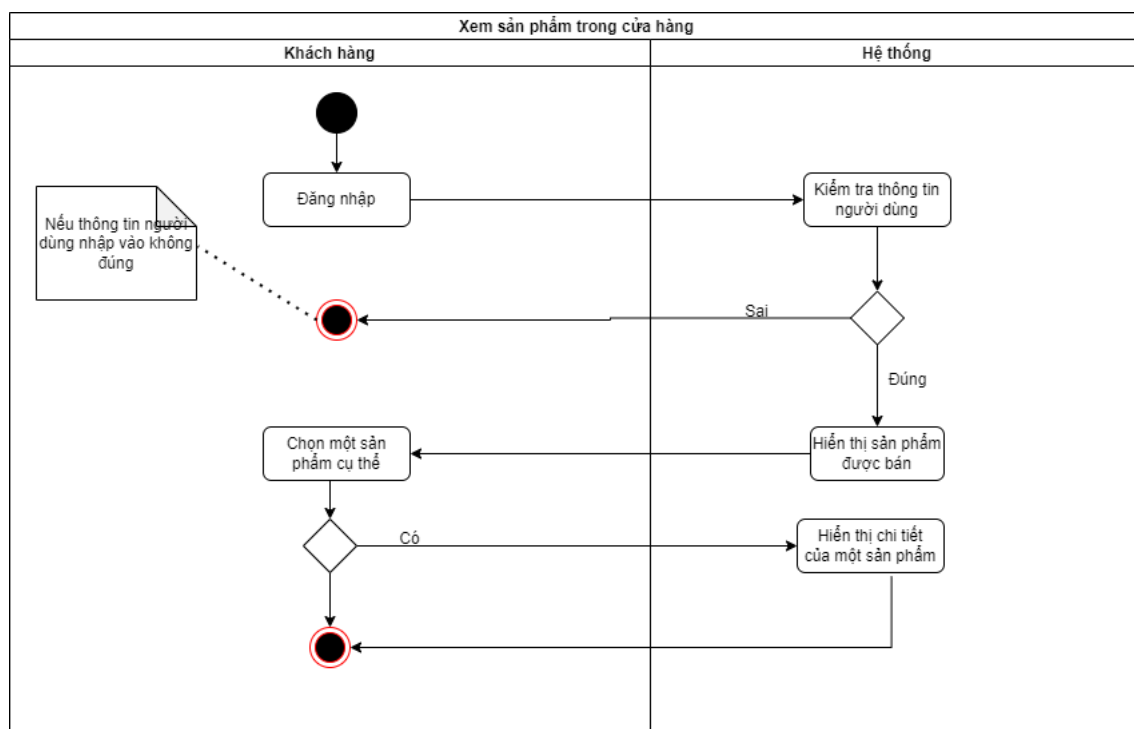


### Activity diagram:

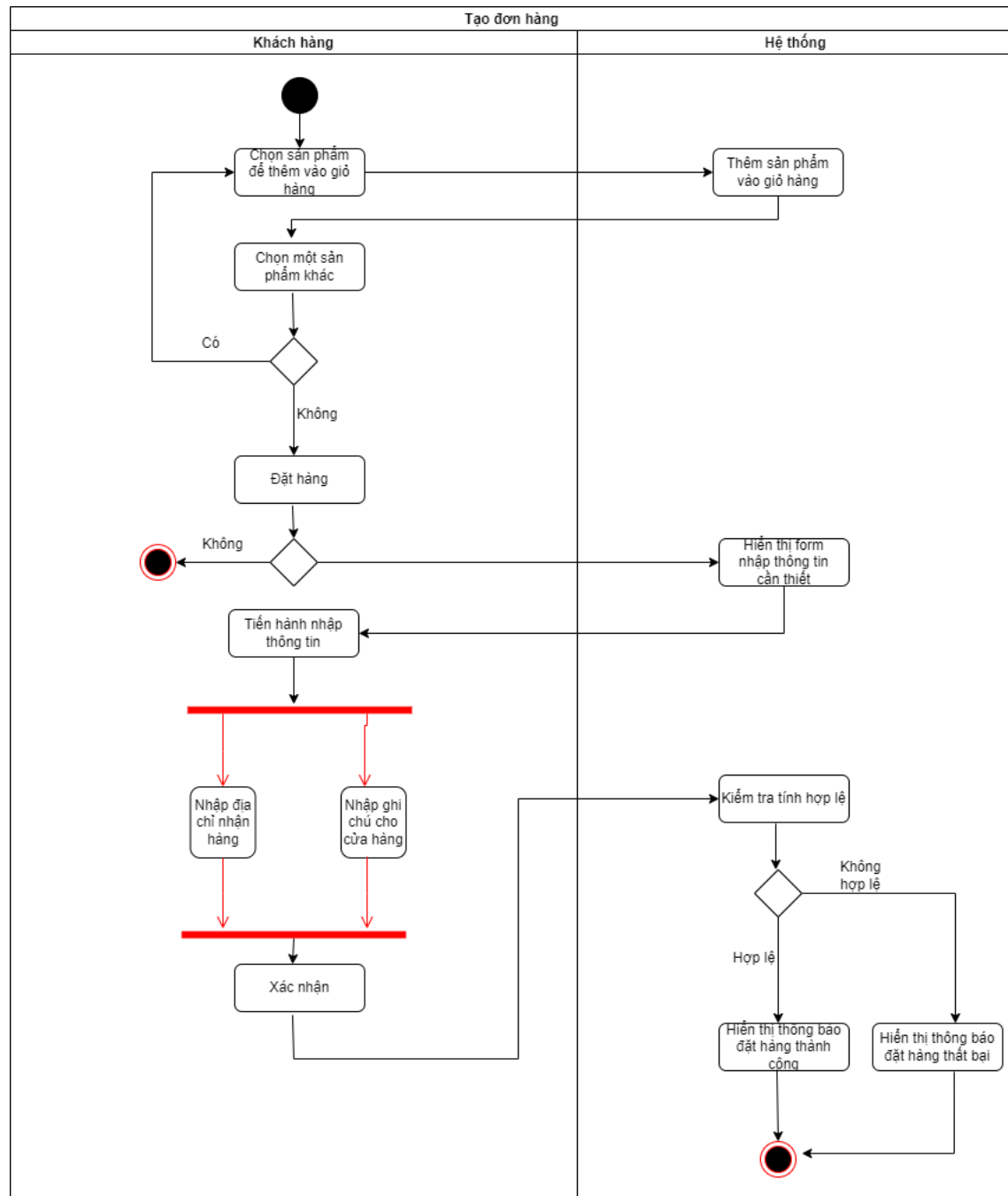
Sơ đồ hoạt động thêm sách mới vào cửa hàng.



Sơ đồ hoạt động khi khách hàng xem sản phẩm của cửa hàng.

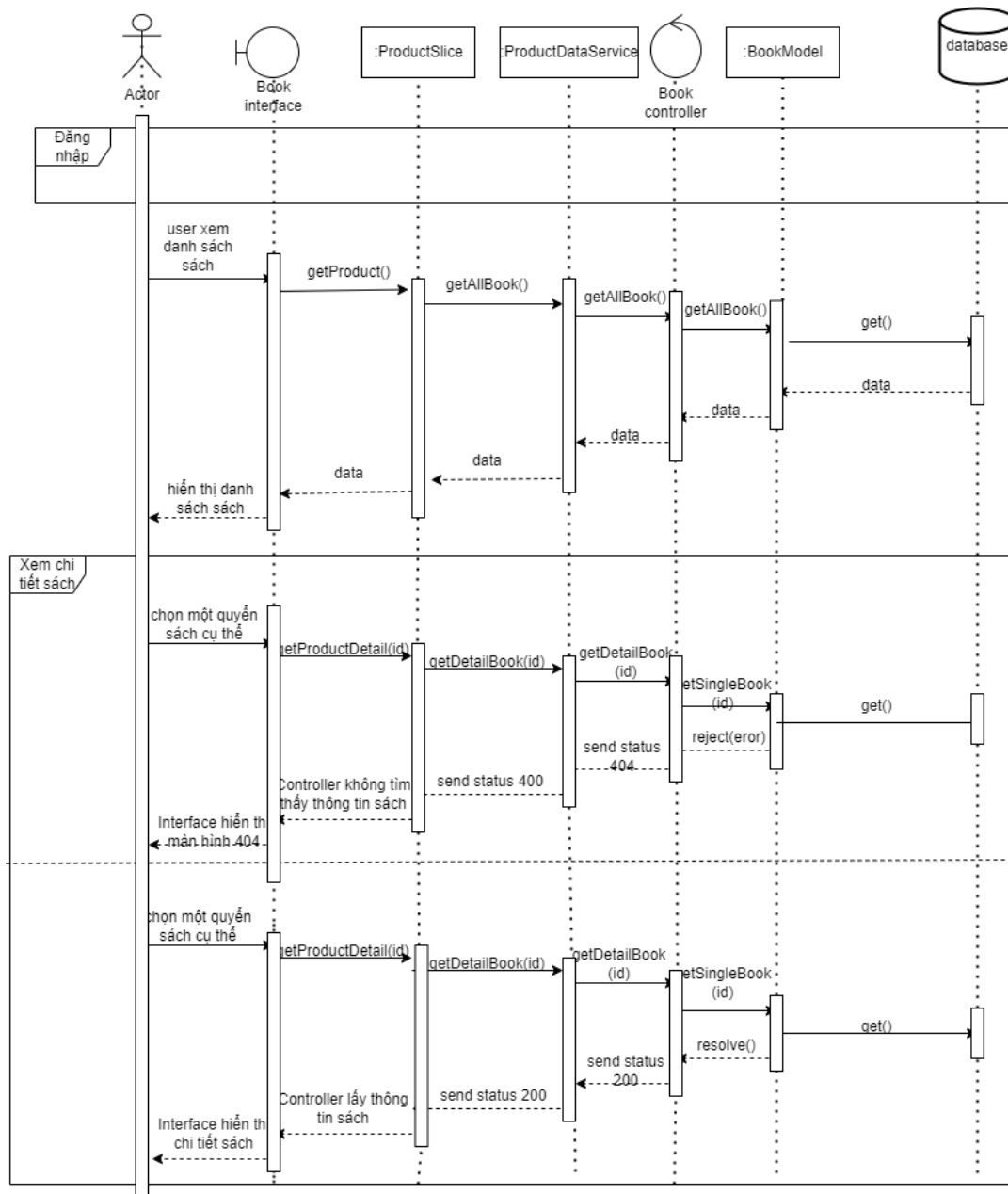


Sơ đồ hoạt động khi khách hàng đặt một đơn hàng:

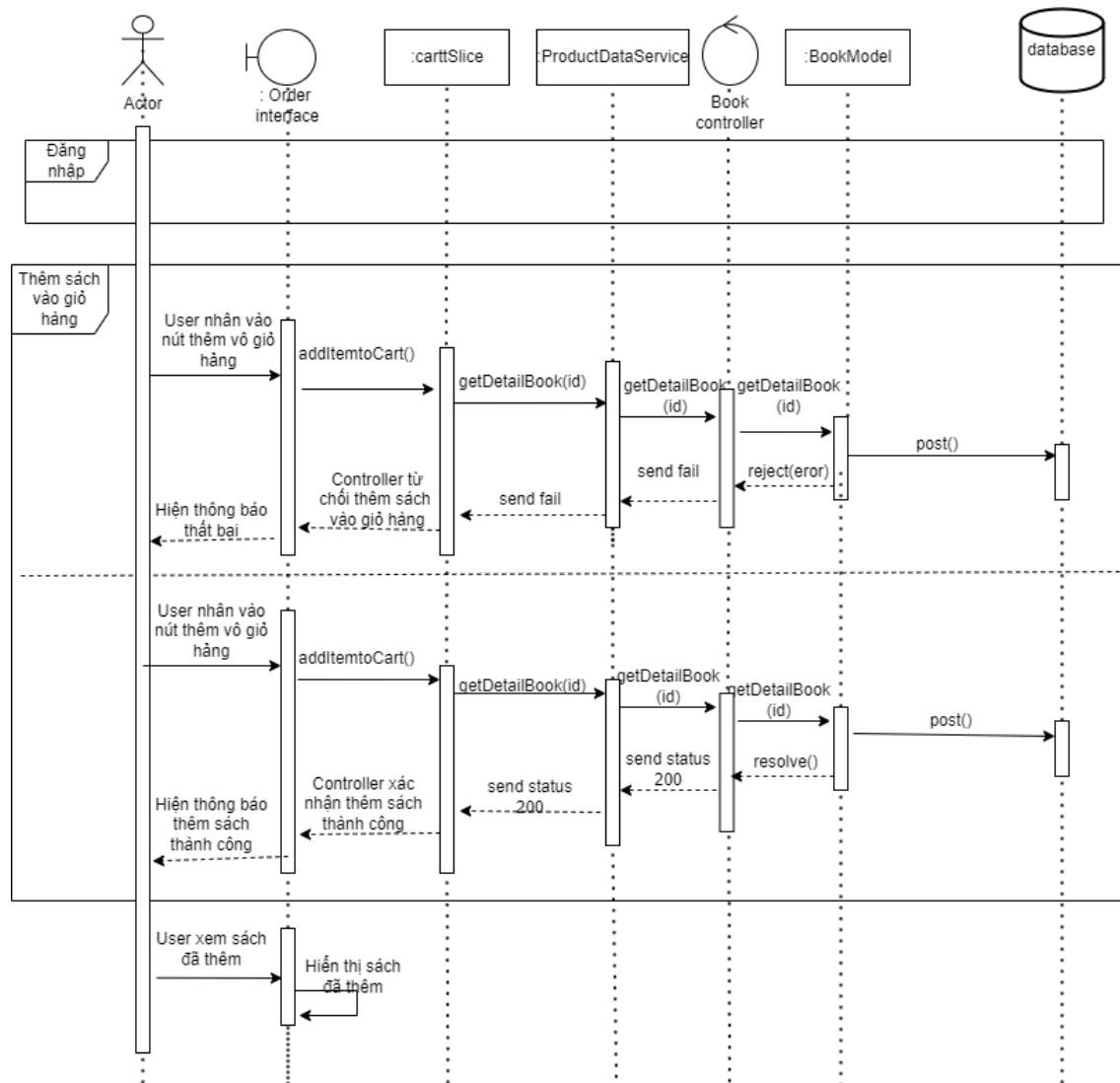


### Sequence diagram:

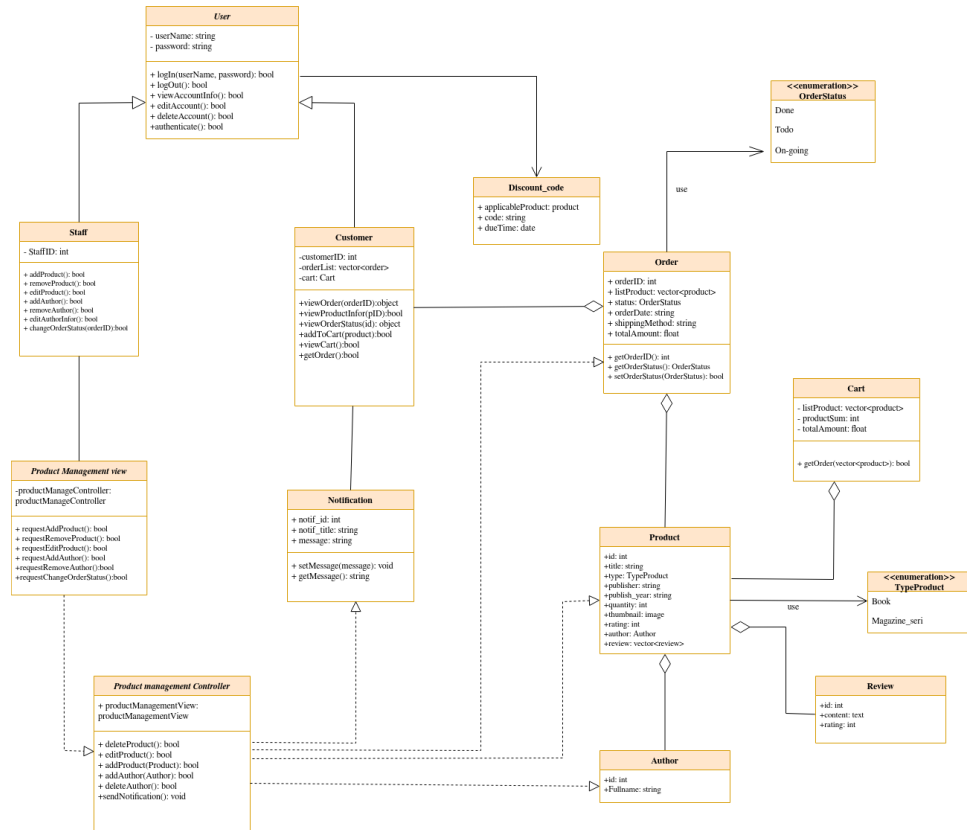
Sơ đồ tuần tự cho việc tìm kiếm sách và xem chi tiết một cuốn sách.



Sơ đồ tuần tự cho việc thêm sách vào giỏ hàng.



### Class diagram:



Hình 24: Class diagram

## 6 Chuẩn hóa lược đồ về dạng chuẩn BCNF cho lược đồ quan hệ:

Để đạt được dạng chuẩn cao (BCNF) ta cần phải đạt được những dạng chuẩn thấp hơn (1NF, 2NF, 3NF).

### Dạng chuẩn 1NF:

Định nghĩa dạng chuẩn 1NF: Một bảng (quan hệ) được gọi là ở dạng chuẩn 1NF nếu và chỉ nếu toàn bộ các miền giá trị của các cột có mặt trong bảng (quan hệ) đều chỉ chứa các giá trị nguyên tử (nguyên tố).

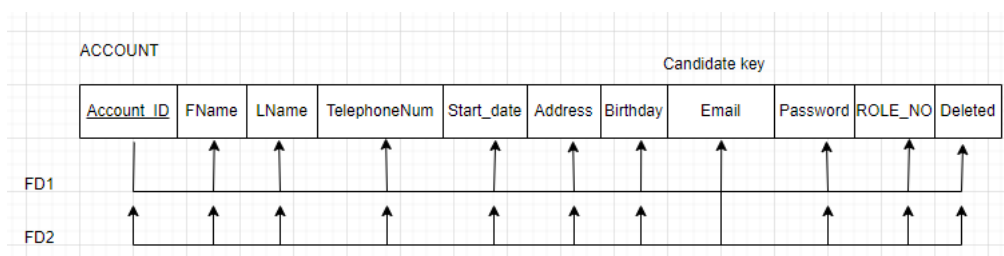
Vì tất cả các ô dữ liệu ở các bảng trong database chỉ lưu trữ những giá trị đơn, không lưu trữ giá trị đa hay các mối quan hệ lồng nhau. Vì vậy lược đồ đã ở dạng chuẩn 1NF.



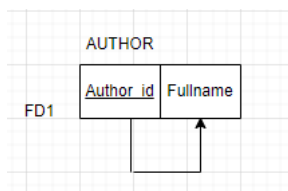
### Dạng chuẩn 2NF:

Định nghĩa dạng chuẩn 2NF: Một bảng (quan hệ) đạt dạng chuẩn 2NF thì nó cũng phải đạt dạng chuẩn 1NF, và các thuộc tính không khóa phải phụ thuộc hàm đầy đủ vào khóa chính.

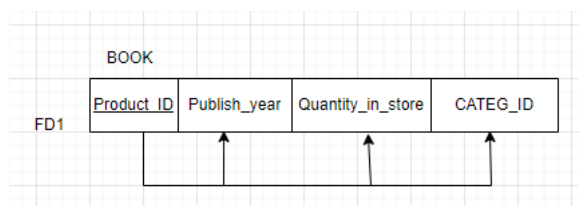
Ở bảng account có các thuộc tính: Account\_ID, FName, LName, TelephoneNum, Start\_date, Address, Birthday, Email, Password, ROLE\_NO, Deleted. Trong đó Account\_ID là khóa chính và Email là khóa dự tuyển. Các thuộc tính khác phụ thuộc hàm đầy đủ vào cả khóa chính Account\_ID và khóa dự tuyển Email, vì vậy bảng account đã đạt được dạng chuẩn 2NF.



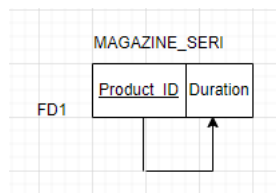
Ở bảng author có các thuộc tính: Author\_ID và Fullname. Trong đó Author\_ID là khóa chính, thuộc tính Fullname phụ thuộc hàm đầy đủ vào thuộc tính Author\_ID. Vì vậy bảng author đạt chuẩn 2NF.



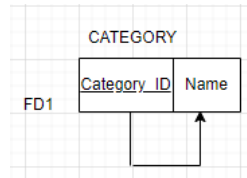
Ở bảng book có các thuộc tính: Product\_ID, Publish\_year, Quantity\_in\_store và CATEG\_ID. Trong đó Product\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm đầy đủ vào Product\_ID. Vì vậy bảng book đạt chuẩn 2NF.



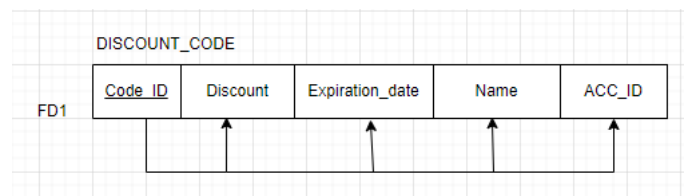
Ở bảng Magazine\_seri có các thuộc tính: Product\_ID và Duration. Trong đó thuộc tính Product\_ID là khóa chính, thuộc tính Duration phụ thuộc hàm đầy đủ vào Product\_ID. Vì vậy bảng Magazine\_seri đạt chuẩn 2NF.



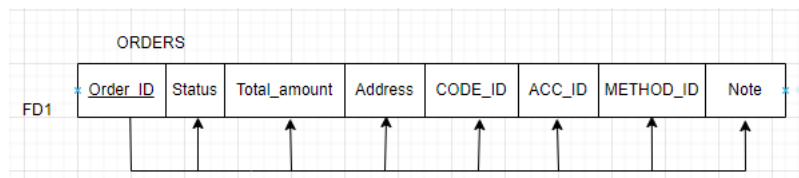
Ở bảng category có các thuộc tính: Category\_ID và Name. Trong đó Category\_ID là khóa chính, thuộc tính Name phụ thuộc hàm đầy đủ vào Category\_ID. Vì vậy bảng Category đạt chuẩn 2NF.



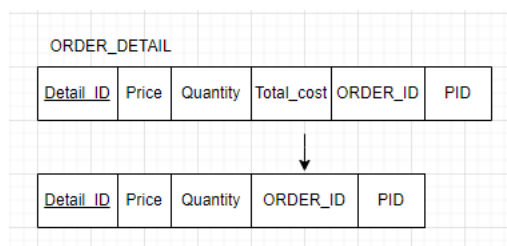
Ở bảng discount\_code có các thuộc tính: Code\_ID, Discount, Expiration\_date, Name, ACC\_ID. Trong đó thuộc tính Code\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm đầy đủ vào thuộc tính Code\_ID. Vì vậy bảng Discount\_code đạt chuẩn 2NF.



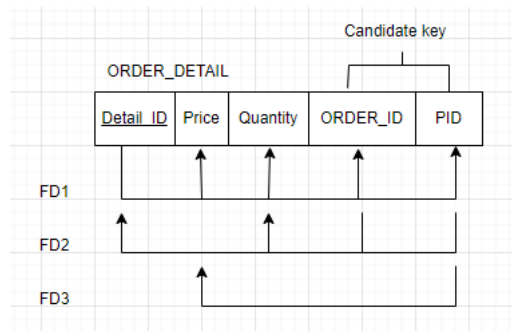
Ở bảng Orders có các thuộc tính: Order\_ID, Status, Total\_amount, Address, CODE\_ID, ACC\_ID, METHOD\_ID, Note. Trong đó thuộc tính Order\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm đầy đủ vào Order\_ID. Vì vậy bảng Orders đạt dạng chuẩn 2NF.



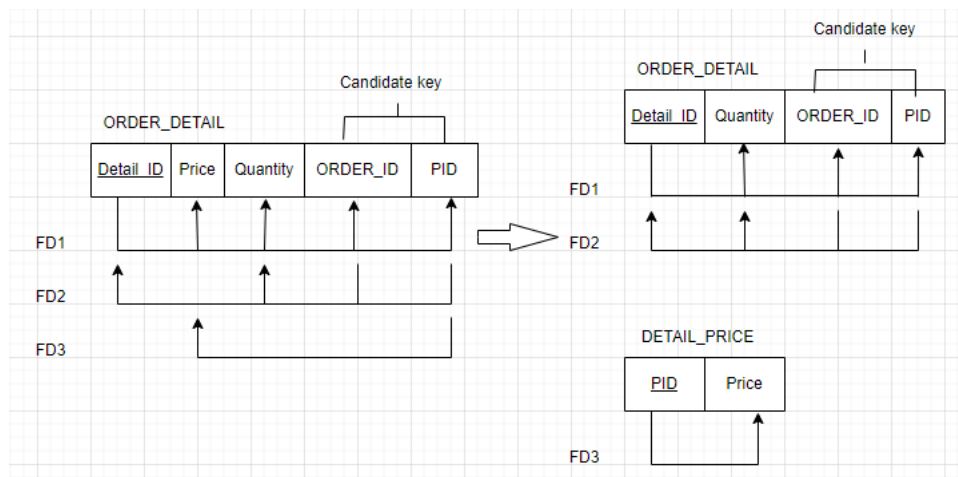
Ở bảng Order\_detail có các thuộc tính Detail\_ID, Price, Quantity, Total\_cost, ORDERID, PID. Ta thấy rằng thuộc tính Total\_cost là một thuộc tính dẫn xuất, có thể được suy ra từ các thuộc tính khác, ở đây Total\_cost sẽ bằng tích của Price với Quantity. Vì vậy ta có thể lược đi thuộc tính này mà vẫn đảm bảo được việc không mất dữ liệu.



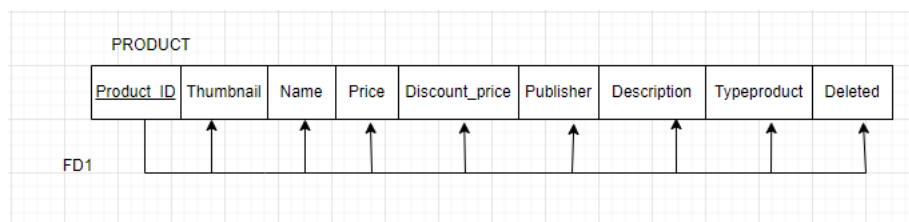
Và ở bảng này thuộc tính Detail\_ID là khóa chính, các thuộc tính khác phụ thuộc hàm đầy đủ vào khóa chính. Nhưng ngoài ra còn có một khóa dự tuyển là kết hợp của 2 thuộc tính: ORDERID và PID. Lúc này xét lại thì ta thấy thuộc tính Price phụ thuộc hàm riêng phần (vì nó chỉ phụ thuộc vào PID).



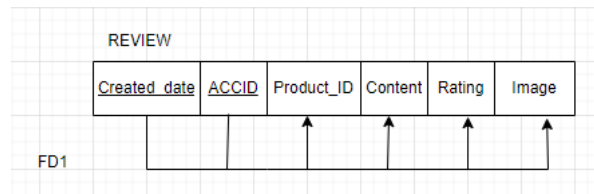
Vì vậy ta sẽ tạo bảng mới tên là Detail\_price gồm 2 thuộc tính PID, Price trong đó PID là khóa chính. Và ở bảng Order\_detail lúc này chỉ còn các thuộc tính: Detail\_ID, Quantity, ORDERID, PID. Cả 2 bảng Order\_detail và Detail\_price lúc này đều đạt được dạng chuẩn 2NF.



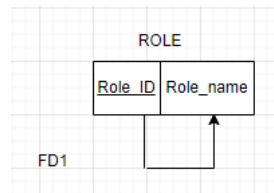
Ở bảng Product có các thuộc tính: Thumbnail, Name, Product\_ID, Price, Discount\_price, Publisher, Description, Deleted, Typeproduct. Trong đó thuộc tính Product\_ID là thuộc tính khóa, các thuộc tính còn lại phụ thuộc hàm đầy đủ vào Product\_ID. Vì vậy bảng Product đạt được dạng chuẩn 2NF.



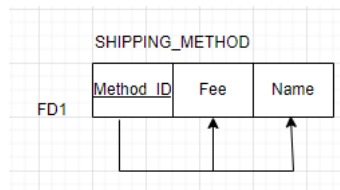
Ở bảng Review có các thuộc tính: Created\_date, ACCID, Product\_ID, Content, Rating, Image. Trong đó 2 thuộc tính Created\_date và ACCID kết hợp lại làm khóa chính. Các thuộc tính khác phụ thuộc hàm đầy đủ vào khóa chính. Vì vậy bảng Review đạt được dạng chuẩn 2NF.



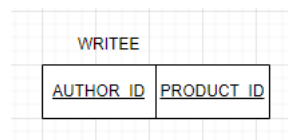
Ở bảng Role có các thuộc tính: Role\_ID và Role\_name. Trong đó Role\_ID là khóa chính, Role\_name phụ thuộc hàm đầy đủ vào Role\_ID. Vì vậy bảng Role đạt được dạng chuẩn 2NF.



Ở bảng shipping\_method có các thuộc tính: Method\_ID, Fee, Name. Trong đó Method\_ID là khóa chính, các thuộc tính khác phụ thuộc hàm đầy đủ vào Method\_ID. Vì vậy bảng Shii-  
ping\_method đạt dạng chuẩn 2NF.



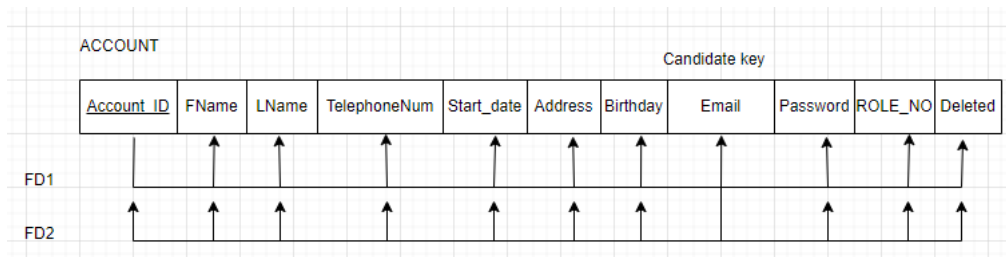
Ở bảng Writee có các thuộc tính AUTHOR\_ID và PRODUCT\_ID, cả 2 thuộc tính cùng kết hợp làm khóa chính. Vì vậy bảng Writee đạt dạng chuẩn 2NF.



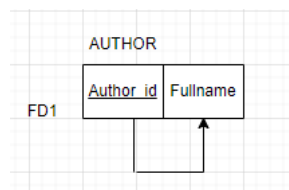
### Dạng chuẩn 3NF:

Định nghĩa dạng chuẩn 3NF: Một bảng (quan hệ) được gọi là ở dạng chuẩn 3NF nếu nó thỏa dạng chuẩn 2NF, và các thuộc tính không khóa phụ thuộc hàm trực tiếp vào khóa chính.

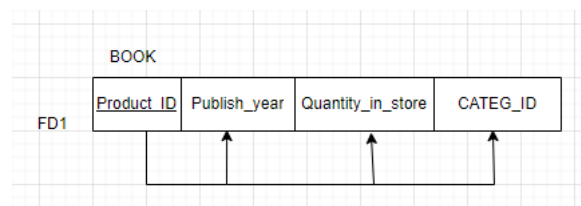
Ở bảng account có các thuộc tính: Account\_ID, FName, LName, TelephoneNum, Start\_date, Address, Birthday, Email, Password, ROLE\_NO, Deleted. Trong đó Account\_ID là khóa chính và Email là khóa dự tuyển. Các thuộc tính đều phụ thuộc hàm trực tiếp vào cả khóa chính và khóa dự tuyển. Vì vậy bảng Account đạt dạng chuẩn 3NF.



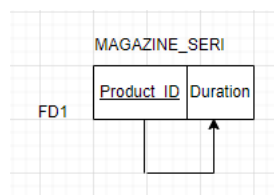
Ở bảng author có các thuộc tính: Author\_ID và Fullname. Trong đó Author\_ID là khóa chính, và thuộc tính Fullname phụ thuộc trực tiếp vào Author\_ID. Vì vậy bảng author đạt dạng chuẩn 3NF.



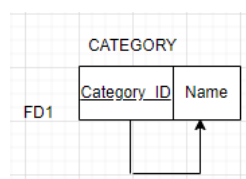
Ở bảng book có các thuộc tính: Product\_ID, Publish\_year, Quantity\_in\_store và CATEG\_ID. Trong đó Product\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm trực tiếp vào khóa chính. Vì vậy bảng Book đạt chuẩn 3NF.



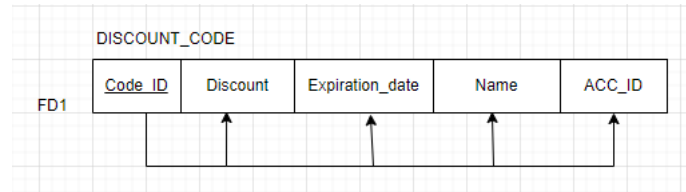
Ở bảng Magazine\_seri có các thuộc tính: Product\_ID và Duration. Trong đó thuộc tính Product\_ID là khóa chính, thuộc tính Duration phụ thuộc hàm trực tiếp vào Product\_ID. Vì vậy bảng Magazine\_seri đạt chuẩn 3NF.



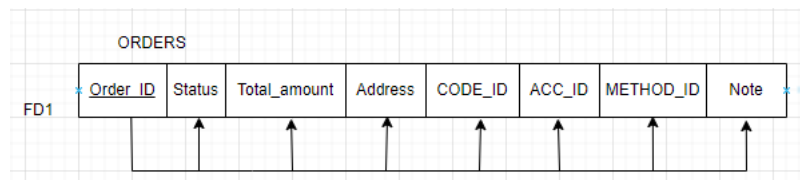
Ở bảng category có các thuộc tính: Category\_ID và Name. Trong đó Category\_ID là khóa chính, thuộc tính Name phụ thuộc hàm trực tiếp vào Category\_ID, vì vậy bảng Category đạt chuẩn 3NF.



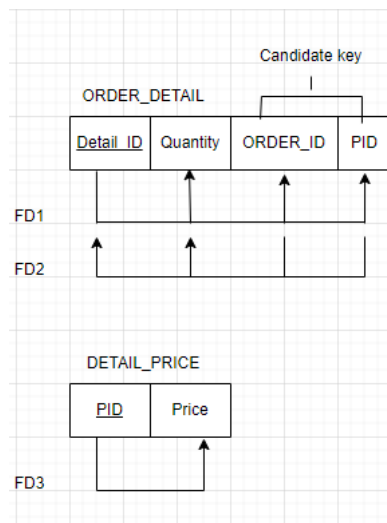
Ở bảng discount\_code có các thuộc tính: Code\_ID, Discount, Expiration\_date, Name, ACC\_ID. Trong đó thuộc tính Code\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm trực tiếp vào thuộc tính Code\_ID. Vì vậy bảng Discount\_code đạt chuẩn 3NF.



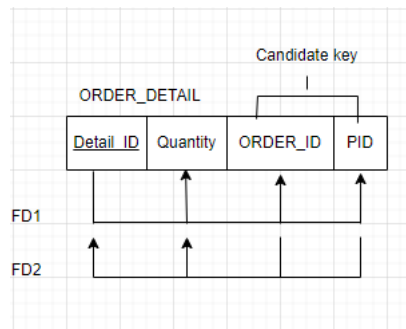
Ở bảng Orders có các thuộc tính: Order\_ID, Status, Total\_amount, Address, CODE\_ID, ACC\_ID, METHOD\_ID, Note. Trong đó thuộc tính Order\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm trực tiếp vào Order\_ID. Vì vậy bảng Orders đạt dạng chuẩn 3NF.



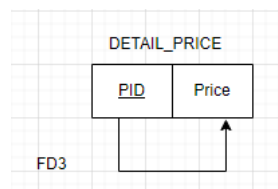
Ở bảng Order\_detail lúc ban đầu chưa đạt chuẩn 2NF, nhưng sau khi ta tách ra thành 2 bảng là: Order\_detail và Detail\_price, thì cả 2 bảng này đã đạt được dạng chuẩn 2NF:



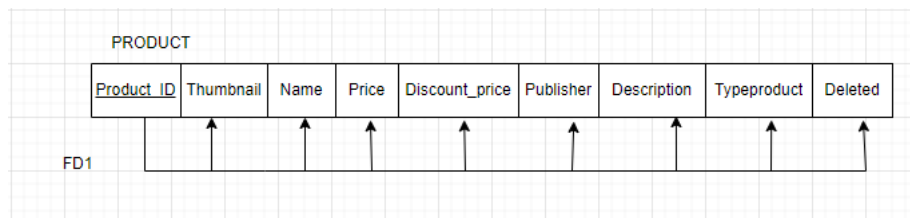
+ Bảng Order\_detail có các thuộc tính: Detail\_ID, Quantity, ORDERID, PID. Trong đó Detail\_ID là khóa chính và ORDERID, PID kết hợp làm khóa dự tuyển. Nếu xét ở khóa chính, ta thấy rằng các thuộc tính còn lại (Quantity, ORDERID, PID) phụ thuộc trực tiếp vào khóa chính. Tương tự nếu xét ở khóa dự tuyển, thì các thuộc tính còn lại (Detail\_ID, Quantity) cũng phụ thuộc trực tiếp vào khóa dự tuyển. Vì vậy bảng Order\_detail đã đạt được dạng chuẩn 3NF.



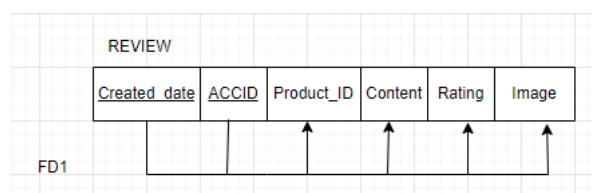
+ Bảng Detail\_price có 2 thuộc tính là PID, Price. Trong đó PID là khóa chính. Thuộc tính Price phụ thuộc hàm trực tiếp vào PID, vì vậy bảng Detail\_price đạt dạng chuẩn 3NF.



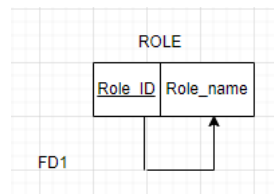
Ở bảng Product có các thuộc tính: Thumbnail, Name, Product\_ID, Price, Discount\_price, Publisher, Description, Deleted, Typeproduct. Trong đó thuộc tính Product\_ID là khóa chính, các thuộc tính còn lại phụ thuộc hàm trực tiếp vào Product\_ID. Vì vậy bảng Product đạt dạng chuẩn 3NF.



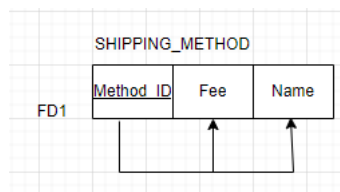
Ở bảng Review có các thuộc tính: Created\_date, ACCID, Product\_ID, Content, Rating, Image. Trong đó 2 thuộc tính Created\_date và ACCID kết hợp lại làm khóa chính. Các thuộc tính khác phụ thuộc hàm trực tiếp vào khóa chính. Vì vậy bảng Review đạt được dạng chuẩn 3NF.



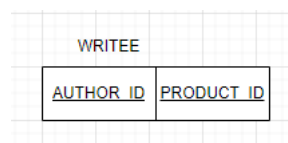
Ở bảng Role có các thuộc tính: Role\_ID và Role\_name. Trong đó Role\_ID là khóa chính, Role\_name phụ thuộc hàm trực tiếp vào Role\_ID. Vì vậy bảng Role đạt được dạng chuẩn 3NF.



Ở bảng shipping\_method có các thuộc tính: Method\_ID, Fee, Name. Trong đó Method\_ID là khóa chính, các thuộc tính khác phụ thuộc hàm trực tiếp vào Method\_ID. Vì vậy bảng Shipping\_method đạt dạng chuẩn 3NF.



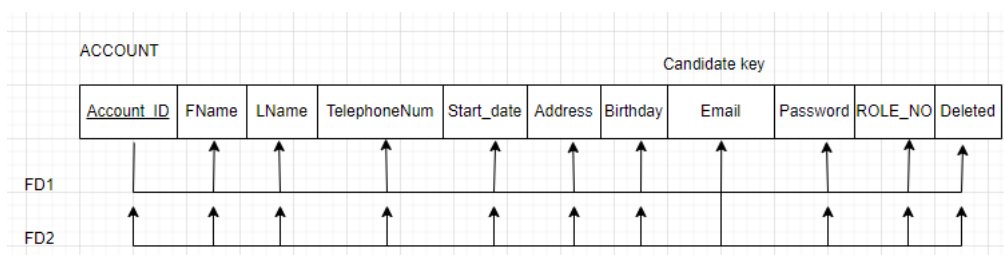
Ở bảng Writee có các thuộc tính AUTHOR\_ID và PRODUCT\_ID, cả 2 thuộc tính cùng kết hợp làm khóa chính. Không có thuộc tính nào phụ thuộc hàm gián tiếp vào khóa chính. Vì vậy bảng Writee đạt dạng chuẩn 3NF.



### Dạng chuẩn BCNF:

Định nghĩa dạng chuẩn BCNF là: Một lược đồ quan hệ (R) là ở dạng chuẩn BCNF nếu trong tất cả các phụ thuộc hàm xác định được:  $FD: X \rightarrow A$ . Thì vế trái của phụ thuộc hàm (X) phải là siêu khóa của R.

Ở bảng account có các thuộc tính: Account\_ID, FName, LName, TelephoneNum, Start\_date, Address, Birthday, Email, Password, ROLE\_NO, Deleted. Có các phụ thuộc hàm xác định được là:



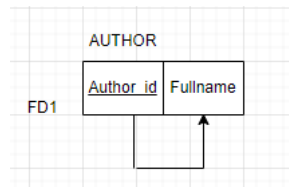
- + **FD1:** Account\_ID  $\rightarrow$  FName, LName, TelephoneNum, Start\_date, Address, Birthday, Email, Password, ROLE\_NO, Deleted.
- + **FD2:** Email  $\rightarrow$  Account\_ID, FName, LName, TelephoneNum, Start\_date, Address, Birthday, Password, ROLE\_NO, Deleted.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Account\_ID. Và Account\_ID là khóa chính của bảng account. Vì vậy Account\_ID là siêu khóa. Tương tự vế trái của phụ thuộc hàm FD2



là Email. Và Email là khóa dự tuyển của bảng account. Vì vậy Email là siêu khóa.  
Vậy bảng account đạt dạng chuẩn BCNF.

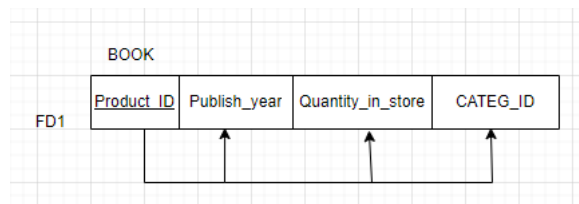
Ở bảng author có các thuộc tính: Author\_ID và Fullname. Có các phụ thuộc hàm xác định được là:



+ **FD1:** Author\_ID -> Fullname.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Author\_ID. Và Author\_ID là khóa chính của bảng author. Vì vậy Author\_ID là siêu khóa.  
Vậy bảng author đạt dạng chuẩn BCNF.

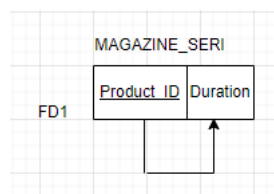
Ở bảng book có các thuộc tính: Product\_ID, Publish\_year, Quantity\_in\_store và CATEG\_ID. Có các phụ thuộc hàm xác định được là:



+ **FD1:** Product\_ID -> Publish\_year, Quantity\_in\_store, CATEG\_ID.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Product\_ID. Và Product\_ID là khóa chính của bảng book. Vì vậy Product\_ID là siêu khóa.  
Vậy bảng book đạt dạng chuẩn BCNF.

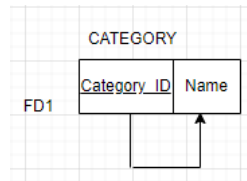
Ở bảng magazine\_seri có các thuộc tính: Product\_ID, Duration. Có các phụ thuộc hàm xác định được là:



+ **FD1:** Product\_ID -> Duration.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Product\_ID. Và Product\_ID là khóa chính của bảng magazine\_seri. Vì vậy Product\_ID là siêu khóa. Vậy bảng magazine\_seri đạt dạng chuẩn BCNF.

Ở bảng category có các thuộc tính: Category\_ID, Name. Có các phụ thuộc hàm xác định được là:

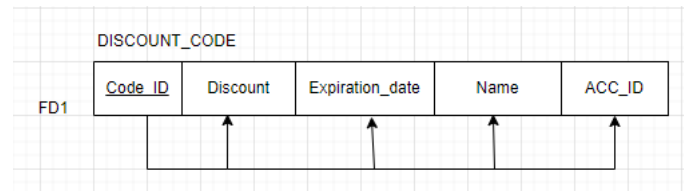


+ **FD1:** Category\_ID → Name.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Category\_ID. Và Category\_ID là khóa chính của bảng category. Vì vậy Category\_ID là siêu khóa.

Vậy bảng category đạt dạng chuẩn BCNF.

Ở bảng discount code có các thuộc tính: Code\_ID, Discount, Expiration\_date, Name, ACC\_ID. Có các phụ thuộc hàm xác định được là:

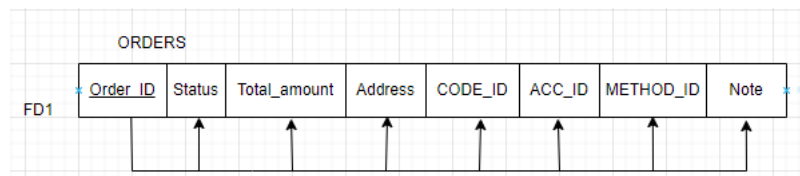


+ **FD1:** Code\_ID → Discount, Expiration\_date, Name, ACC\_ID.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Code\_ID. Và Code\_ID là khóa chính của bảng discount\_code. Vì vậy Code\_ID là siêu khóa.

Vậy bảng discount\_code đạt dạng chuẩn BCNF.

Ở bảng Orders có các thuộc tính: Order\_ID, Status, Total\_amount, Address, CODE\_ID, ACC\_ID, METHOD\_ID, Note. Có các phụ thuộc hàm xác định được là:

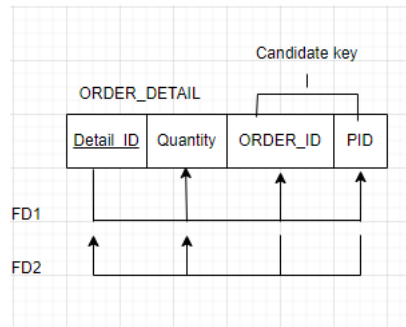


+ **FD1:** Order\_ID → Status, Total\_amount, Address, CODE\_ID, ACC\_ID, METHOD\_ID, Note.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Order\_ID. Và Order\_ID là khóa chính của bảng Orders. Vì vậy Order\_ID là siêu khóa.

Vậy bảng Orders đạt dạng chuẩn BCNF.

Bảng order\_detail ban đầu đã được tách ra làm 2 bảng: order\_detail và detail\_price. + Bảng order\_detail có các thuộc tính: Detail\_ID, Quantity, ORDERID, PID. Có các phụ thuộc hàm xác định được là:

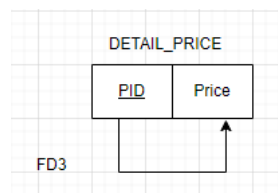


+ **FD1**: Detail\_ID -> Quantity, ORDERID, PID.

+ **FD2**: ORDERID, PID -> Detail\_ID, Quantity. Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Detail\_ID. Và Detail\_ID là khóa chính của bảng order\_detail, vì vậy Detail\_ID cũng là siêu khóa. Vế trái của phụ thuộc hàm FD2 là ORDERID, PID và ORDERID,PID kết hợp lại làm khóa dự tuyển của order\_detail, vì vậy cũng là siêu khóa.

Vậy bảng order\_detail đạt chuẩn BCNF.

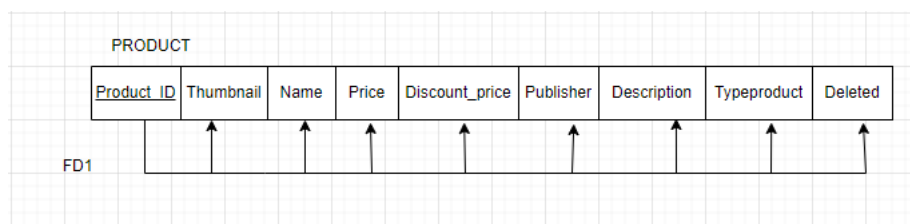
Bảng detail\_price có các thuộc tính: PID, Price. Có các phụ thuộc hàm xác định được là:



+ **FD3**: PID -> Price.

Ta thấy rằng vế trái của phụ thuộc hàm FD3 là PID. Và PID là khóa chính của bảng detail\_price, vì vậy PID cũng là siêu khóa. Vậy bảng detail\_price đạt chuẩn BCNF.

Ở bảng Product có các thuộc tính: Thumbnail, Name, Product\_ID, Price, Discount\_price, Publisher, Description, Deleted, Typeproduct. Có các phụ thuộc hàm xác định được là:

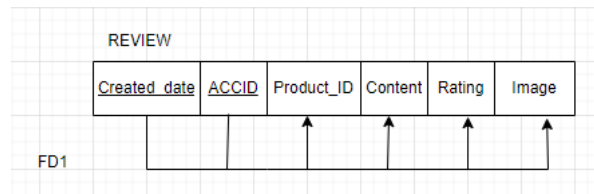


+ **FD1**: Product\_ID -> Thumbnail, Name, Price, Discount\_price, Publisher, Description, Deleted, Typeproduct.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Product\_ID. Và Product\_ID là khóa chính của bảng Product, vì vậy Product\_ID cũng là siêu khóa.

Vậy bảng product đạt chuẩn BCNF.

Ở bảng Review có các thuộc tính: Created\_date, ACCID, Product\_ID, Content, Rating, Image. Có các phụ thuộc hàm xác định được là:

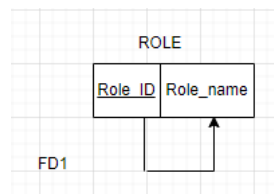


+ **FD1**: Created\_date, ACCID -> Product\_ID, Content, Rating, Image.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Created\_date, ACCID. Và Created\_date, ACCID kết hợp làm khóa chính của bảng Review, vì vậy Created\_date, ACCID cũng là siêu khóa.

Vậy bảng Review đạt chuẩn BCNF.

Ở bảng Role có các thuộc tính: Role\_ID và Role\_name. Có các phụ thuộc hàm xác định được là:

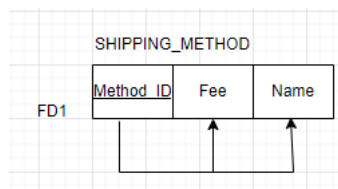


+ **FD1**: Role\_ID -> Role\_name.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Role\_ID. Và Role\_ID là khóa chính của bảng Role, vì vậy Role\_ID cũng là siêu khóa.

Vậy bảng Role đạt chuẩn BCNF.

Ở bảng shipping\_method có các thuộc tính: Method\_ID, Fee, Name. Có các phụ thuộc hàm xác định được là:

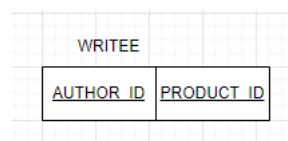


+ **FD1**: Method\_ID -> Fee, Name.

Ta thấy rằng vế trái của phụ thuộc hàm FD1 là Method\_ID. Và Method\_ID là khóa chính của bảng shipping\_method, vì vậy Method\_ID cũng là siêu khóa.

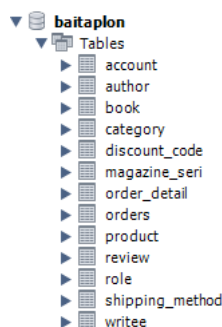
Vậy bảng shipping\_method đạt chuẩn BCNF.

Ở bảng Writee chỉ có hai thuộc tính AUTHOR\_ID và PRODUCT\_ID, cả 2 thuộc tính cùng kết hợp làm khóa chính. Vì vậy bảng Writee đạt dạng chuẩn BCNF.

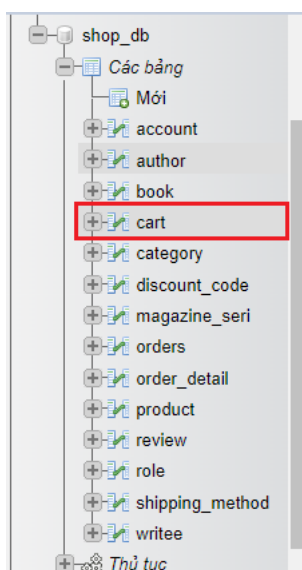


## 7 So sánh dữ liệu trong cơ sở dữ liệu và dữ liệu trong các tập tin như text files khi phát triển ứng dụng cho đề tài của assignment

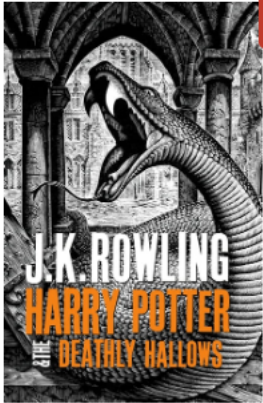
Cơ sở dữ liệu ban đầu mà nhóm thiết kế gồm tổng cộng 13 bảng để chứa các dữ liệu cần thiết.



Khi thiết kế ứng dụng thực tế, cơ sở dữ liệu mà nhóm thực hiện gồm 14 bảng, nhiều hơn so với khi thiết kế ban đầu 1 bảng, đó là bảng cart. Bảng này nhằm mục đích lưu lại những item mà khách hàng thêm vào giỏ hàng trong quá trình mua sắm. Nếu sau đó, khách hàng đặt hàng thành công thì những record trong bảng sẽ biến mất.



Ví dụ trong quá trình mua sắm, khách hàng có thêm vào giỏ hàng 3 sản phẩm. Lúc này ta kiểm tra dữ liệu ở bảng cart sẽ thấy có 3 record mới được thêm vào, tương ứng với 3 item hiện có trong giỏ hàng.

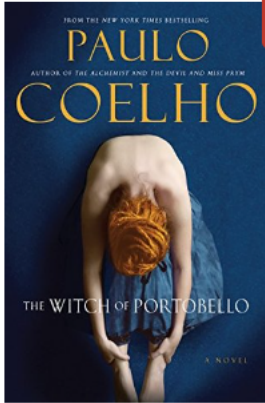


Harry Potter And The Deathly Hallows

\$21/-

1 [Update](#)

sub total : \$21/-

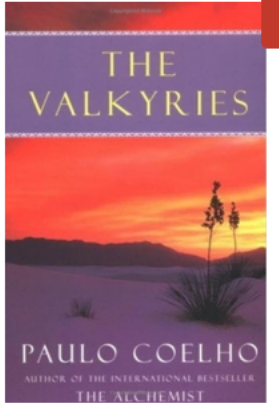


The Witch Of Portobello

\$10/-

2 [Update](#)

sub total : \$20/-



The Valkyries

\$20/-

1 [Update](#)

sub total : \$20/-

✓ Hiện thị các hàng 0 - 2 (tổng 3, Truy vấn cần 0,0003 giây.)

`SELECT * FROM `cart``

☐ Hồ sơ [\[ Sửa chung dòng \]](#) [\[ Sửa \]](#) [\[ Giải thích SQL \]](#) [\[ Tạo mã PHP \]](#) [\[ Làm tươi lại \]](#)

☐ Hiện tắt | Số hàng: 25 | Số hàng:  Sort by key:

Extra options

				id	user_id	Product_ID	name	price	quantity	image
<input type="checkbox"/>	<a href="#">Sửa</a>	<a href="#">Chép</a>	<a href="#">Xóa bỏ</a>	110	27	16	Harry Potter And The Deathly Hallows	21	1	9781408865453.jpg
<input type="checkbox"/>	<a href="#">Sửa</a>	<a href="#">Chép</a>	<a href="#">Xóa bỏ</a>	111	27	18	The Witch Of Portobello	10	2	38313371_SY475_.jpg
<input type="checkbox"/>	<a href="#">Sửa</a>	<a href="#">Chép</a>	<a href="#">Xóa bỏ</a>	112	27	19	The Valkyries	20	1	image_128701_thanh_ly.jpg

Khi khách hàng đặt hàng thành công, thì những record trong bảng sẽ được xóa.



order placed successfully!



## PLACE YOUR ORDER

your first name :

vien

your last name :

tri

shipping method :

Giao hàng tiết kiệm | \$20

your address :

Kí túc xá khu A

your note :

please enter your note

Order Now

✓ MySQL trả về kết quả là trống rỗng (vd. không có hàng nào). (Truy vấn cần 0,0019 giây.)

```
SELECT * FROM `cart`
```

☐ Hồ sơ [ Sửa chung dòng ] [ Sửa ] [ Giải thích SQL ] [ Tạo mã PHP ] [ Làm tươi lại ]

id	user_id	Product_ID	name	price	quantity	image
----	---------	------------	------	-------	----------	-------





Kết quả truy vấn các thao tác

Tạo view

Ngoài ra ở bảng Orders ban đầu nhóm thiết kế gồm 8 thuộc tính là: Order\_ID, Status, Total\_amount, Address, CODE\_ID, ACC\_ID, METHOD\_ID, Note.

Columns:  
**Order\_ID** int AI PK  
Status varchar(255)  
Total\_amount int  
Address varchar(255)  
**CODE\_ID** int  
**ACC\_ID** int  
**METHOD\_ID** int  
Note varchar(255)

Nhưng trong quá trình phát triển ứng dụng, nhóm nhận thấy cần lưu thêm một thuộc tính ngày tạo đơn hàng. Thuộc tính này có kiểu dữ liệu là datetime (YYYY-MM-DD HH:MI:SS) nhằm mục đích lưu lại thời gian mà đơn hàng được thêm vào bảng Orders.

	#	Tên	Kiểu
<input type="checkbox"/>	1	Order_ID 	int(11)
<input type="checkbox"/>	2	Status	varchar(255)
<input type="checkbox"/>	3	Total_amount	int(11)
<input type="checkbox"/>	4	Address	varchar(255)
<input type="checkbox"/>	5	CODE_ID 	int(11)
<input type="checkbox"/>	6	ACC_ID 	int(11)
<input type="checkbox"/>	7	METHOD_ID 	int(11)
<input type="checkbox"/>	8	Note	varchar(255)
<input type="checkbox"/>	9	<u>pay_date</u>	datetime

Sau đó dựa vào giá trị của thuộc tính, chúng ta sử dụng câu truy vấn để xác định được ID của đơn hàng vừa thêm vào (vì thuộc tính Order\_ID trong bảng đơn hàng là một thuộc tính được gán AUTO\_INCREMENT, vì vậy khi thêm một đơn hàng vào bảng Orders, ta không cần nhập giá trị cho nó). Sau khi có được ID của đơn hàng, ta sẽ sử dụng nó để thêm các record tương ứng với đơn hàng vào bảng order\_detail.

VD: Ở đây, ta sẽ truy vấn trên bảng đơn hàng để lấy ra Order\_ID của đơn hàng có ngày tạo là: 2022-12-04 17:28:09, ta sẽ được kết quả: 26. Sau đó ta sẽ dùng Order\_ID này để thêm các record vào bảng Order\_detail với thuộc tính ORDID = 26 (lưu các item của đơn hàng có ID = 26 vào bảng Order\_detail).





		Order_ID	Status	Total_amount	Address	CODE_ID	ACC_ID	METHOD_ID	Note	pay_date
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	22		7	323, Phạm Văn Chiêu, P14, Go Vap, TPHCM	NULL	18		1 cảm on	2022-12-04 01:52:19
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	23	Pending	30	323, Phạm Văn Chiêu, P14, Go Vap, TPHCM	NULL	18		3 dsadsadasda	2022-12-04 09:02:55
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	24	Completed	50	Kí túc xá khu A	NULL	27		3 note*	2022-12-04 09:30:07
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	25	Pending	8000	Kí túc xá khu A	NULL	27		1 abc	2022-12-04 10:24:03
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	26	Pending	61	Kí túc xá khu A	NULL	27		1 note*	2022-12-04 17:28:09

Ở bảng Order\_detail lúc này các chi tiết đơn hàng có Detail\_ID = 12, 13, 14 đều thuộc đơn hàng Order\_ID = 26.

```
SELECT * FROM `order_detail`
```

☐ Hồ sơ ☐ Sửa chung dòng ☐ Sửa ☐ Giải thích SQL ☐ Tạo mã PHP ☐ Làm tươi lại

☐ Hiện tất | Số hàng: 25  Sort by key: Khó

Extra options

		Detail_ID	Price	Quantity	Total_cost	ORDERID	PID
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	6	3	1	3	22	18
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	7	4	1	4	22	22
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	8	30	1	30	23	15
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	9	30	1	30	24	15
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	10	10	2	20	24	18
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	11	20	400	8000	25	15
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	12	21	1	21	26	16
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	13	10	2	20	26	18
<input type="checkbox"/>	Sửa  Chép  Xóa bỏ	14	20	1	20	26	19


Ngoài ra còn có sự thay đổi khác ở bảng product, thiết kế ban đầu bảng này gồm 9 thuộc tính: Product\_ID, Thumbnail, Name, Price, Discount\_price, Publisher, Description, Deleted, Typeproduct. Thuộc tính Typeproduct nhằm để sau khi thêm 1 sản phẩm mới vào bảng Product, sẽ dựa vào Typeproduct của sản phẩm đó để dùng trigger thêm tự động Product\_ID của sản phẩm vào bảng Book hoặc bảng Magazine\_seri (như đã đề cập ở phần Trigger).

**Table: product**

**Columns:**

Thumbnail	varchar(255)
Name	varchar(255)
<b>Product_ID</b>	int AI PK
Price	int
Discount_price	int
Publisher	varchar(255)
Description	varchar(1000)
Deleted	int
Typeproduct	int

Nhưng trong quá trình thiết kế nhóm nhận thấy không cần phải có thuộc tính này, vì có thể tách việc thêm sách và tạp chí thành 2 lựa chọn riêng trong ứng dụng. Vì vậy nhóm quyết định bỏ thuộc tính Typeproduct.

#	Tên	Kiểu
1	Thumbnail	varchar(255)
2	Name	varchar(255)
3	Product_ID 	int(11)
4	Price	int(11)
5	Discount_price	int(11)
6	Publisher	varchar(255)
7	Description	varchar(500)
8	Deleted	int(11)



## SHOP BOOKS

### ADD BOOK

## SHOP MAGAZINES

### ADD MAGAZINE

Tiếp theo ta kiểm tra sự nhất quán các ràng buộc ở mức DBMS và ứng dụng:

Ở các bảng trong DBMS có các ràng buộc về ngữ nghĩa như sau:

Bảng Account:



- Họ, tên chủ tài khoản chỉ chứa các ký tự.
- Password phải là một chuỗi có độ dài lớn hơn 8 ký tự.
- Email phải theo định dạng something@something.something.

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.

Ví dụ nếu ta nhập FName khi đăng ký tài khoản không thỏa điều kiện thì sẽ có thông báo lỗi xuất hiện.

REGISTER NOW

1234

vien

0373015428

kí túc xá khu B

12/11/2002

abcdef@gmail.com

.....

.....

First name is invalid



REGISTER NOW

enter your first name

Bảng Shipping\_method:

- Phí giao hàng (Fee) phải là một số lớn hơn 0.

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.



### PLACE YOUR ORDER

your first name :

your last name :

shipping method :

Giao hàng tiết kiệm | \$20

Giao hàng tiết kiệm | \$20

Giao hàng nhanh | \$25

Hỏa tốc | \$30

please enter your note

your address :

Order Now

Bảng Review:

- Giá trị của thuộc tính Rating phải là số nguyên từ 1->5

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.

Rating :

!

Giá trị phải nhỏ hơn hoặc bằng 5.

Rating :

!

Giá trị phải lớn hơn hoặc bằng 1.

Bảng Product:

- Giá tiền gốc (Price), cũng như giá tiền đang khuyến mãi hiện tại (Discount\_price) của một sản phẩm phải là một số lớn hơn hoặc bằng 0.

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.

### ADD BOOK

### ADD BOOK

 Giá trị phải lớn hơn hoặc bằng 0

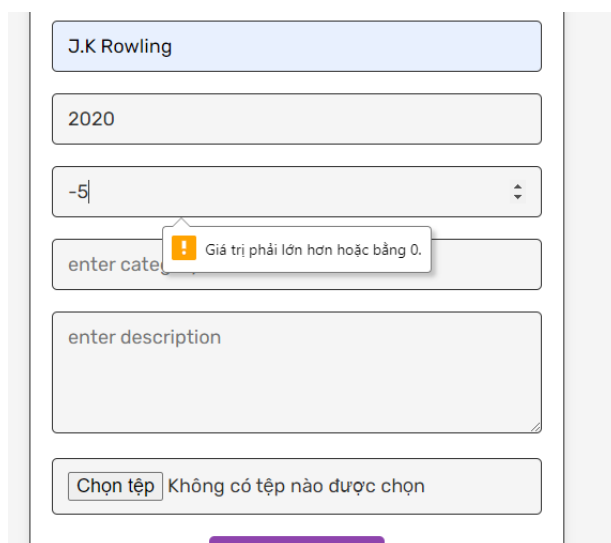
Bảng Book:

- Số lượng sách trong cửa hàng (Quantity\_in\_store) phải lớn hơn hoặc bằng 0.

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.

Chọn tệp

Không có tệp nào được chọn



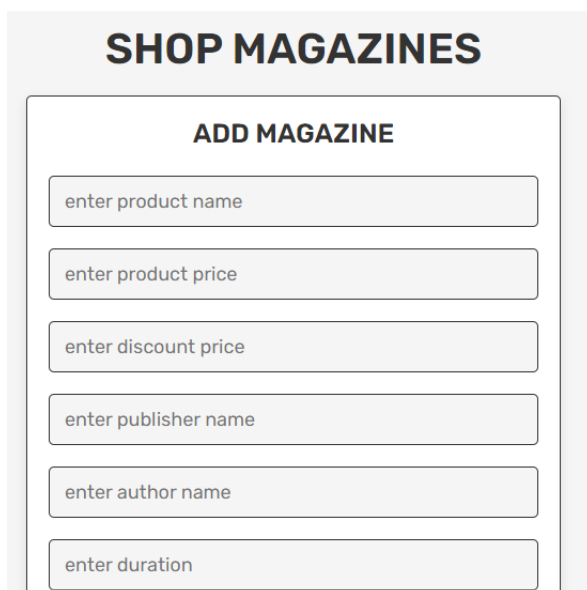
The form contains the following fields and elements:

- Author: J.K Rowling
- Year: 2020
- Category: -5 (with a validation error message: "Giá trị phải lớn hơn hoặc bằng 0.")
- Description: enter cate
- Image: Chọn tệp (Không có tệp nào được chọn)

Bảng Magazine\_seri:

- Khoảng thời gian (Duration) của một seri tạp chí được tính theo đơn vị tháng và lớn hơn 0.

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.



The form is titled "SHOP MAGAZINES" and "ADD MAGAZINE". It contains the following fields:

- enter product name
- enter product price
- enter discount price
- enter publisher name
- enter author name
- enter duration

**ADD MAGAZINE**

 Giá trị phải lớn hơn hoặc bằng 0.

Bảng Orders:

- Đơn hàng chỉ được có một trong 4 trạng thái (status) là: Processing (Đang xử lý), Delivering (Đang giao hàng), Completed (Giao thành công), Canceled (Đã hủy).

Ở ứng dụng thực tế đã đảm bảo được các ràng buộc này.

Quản lý và nhân viên thường của cửa hàng có thể cập nhật trạng thái cho đơn hàng gồm: Processing, Delivering, Canceled.

AdminPanel

home all products books magazines orders users

user id : 27

placed on : 2022-12-04 21:55:31

name : 1234 vien

email : trivien@gmail.com

address : kí túc xá khu B

total products : Harry Potter and the Order of the Phoenix(1).

shipping method : Giao hàng tiết kiệm

total price : \$30 + \$20

Processing

Processing

Processing

Delivering

Cancelled

Trạng thái Completed được cập nhật bởi khách hàng khi nhận hàng thành công.





Status updated successfully !



## PLACED ORDERS

placed on : 2022-12-04 21:55:31

address : kí túc xá khu B

shipping method : Giao hàng tiết kiệm

your orders : Harry Potter and the Order of the Phoenix(1),

total price : \$30 + \$20

payment status : Completed

Received

Một số ràng buộc về cấu trúc khác vẫn được đảm bảo như:

**Ràng buộc tham gia (Participation constraint):** Một quyển sách (Book) phải có một thể loại.

**ADD BOOK**

truyện kinh dị tháng 11

200000

100000

nhà xuất bản trẻ

J.K Rowling

2020

20

enter category

Vui lòng điền vào trường này.

Một sản phẩm phải được viết bởi một tác giả.



## SHOP BOOKS

### ADD BOOK

truyện kinh dị tháng 11


200000

100000

nhà xuất bản trẻ

enter author name

2020

 Vui lòng điền vào trường này.

Một đơn hàng (Order) phải chứa các chi tiết đơn hàng. Nếu giỏ hàng trống thì khách hàng không thể thanh toán, vì vậy sẽ không thêm 1 đơn hàng trống vào bảng Orders được.

## PRODUCTS ADDED

your cart is empty

Delete All

grand total : \$0/-

Continue Shopping

Proceed To Checkout

Một đơn hàng phải có phương thức vận chuyển. Mỗi đơn hàng sẽ có một phương thức vận chuyển cụ thể với chi phí khác nhau.



### PLACE YOUR ORDER

your first name :

your last name :

shipping method :

Giao hàng tiết kiệm | \$20

Giao hàng tiết kiệm | \$20

Giao hàng nhanh | \$25

Hỏa tốc | \$30

please enter your note

your address :

Order Now

**Ràng buộc về tỉ số :** Một sản phẩm có thể được viết bởi nhiều tác giả.

Ở đây ta thêm vào một cuốn sách mới trong cửa hàng, có 2 tác giả viết là: J.K Rowling và Paulo Coelho.

### ADD BOOK

truyện kinh dị mới

200000

100000

nhà xuất bản trẻ

J.K Rowling,Paulo Coelho

2020

Khi ta thêm thành công thì kiểm tra lại ở bảng Product trong cơ sở dữ liệu, sản phẩm vừa thêm có Product\_ID = 27.



	Thumbnail	Name	Product_ID	Price	Discount_price	Publisher	Description	Deleted
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	38313371_SY475_.jpg	The Witch Of Portobello	18	10	3	Macmillan Publisher	Dream is endless, Paulo Coelho imagination is endl...	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	image_128701_thanh_ly.jpg	The Valkyries	19	20	10	Macmillan Publisher	The Valkyries of God	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	INC_Cover_AndreaHenricks.jpg	INCLUSION	21	2	0	PlayBoy	good Magazine	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	lex (1).jpg	Vogue	22	10	4	Vogue	good Magazine for woman	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	A1iilPI1rL-785x1024.jpg	Vogue 3	23	2	1	Vogue	Another Magazine by Kendall Jenner	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	51orT80mDAL_AC_SY780_.jpg	Cristiano Ronaldo: The Biography	24	25	10	Amazon	The Greatest Of All Times	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	app24.png	truyện kinh dị tháng 12	26	200000	100000	nhà xuất bản trẻ	sách hay	0
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	app27.png	truyện kinh dị mới	27	200000	100000	nhà xuất bản trẻ	abc	0

Ta kiểm tra ở bảng Writtee, ở record với PRODUCT\_ID = 27 sẽ có 2 AUTHOR\_ID là 4 và 10.

	AUTHOR_ID	PRODUCT_ID
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	4	13
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	4	15
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	4	16
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	4	26
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	4	27
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	5	17
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	5	18
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	5	19
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	7	21
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	7	23
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	8	22
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	9	24
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	10	26
<input type="checkbox"/> Sửa  Chép  Xóa bỏ	10	27

Một sản phẩm có thể có nhiều đánh giá.



Bookly.

[home](#) [about](#) [shop](#) [contact](#) [orders](#)

(0)

PRICE: \$24 | \$9

Add To Cart

## CLIENT'S REVIEWS



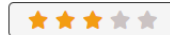
Sách dở ẹt



Mạnh Toan



Sách hay lắm



khang nguyen



SÁCH HAY



khang nguyen

## 8 Cài đặt chỉ mục, minh họa đo đạc: so sánh thời gian thực thi chỉ mục với demo lớn:

Để minh họa cho việc truy vấn dữ liệu để thể hiện sự khác biệt khi không sử dụng chỉ mục và sử dụng chỉ mục. Ở đây nhóm đã sử dụng thủ tục có tên: `create_million_record` để nhập một loạt giá trị vào bảng account.

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `create_million_record`()
2 BEGIN
3 DECLARE i INT DEFAULT 30000000;
4 START TRANSACTION;
5 WHILE (i <= 31000000) DO
6 INSERT INTO account (FName, LName, TelephoneNum, Address, Birthday, Email, Password, ROLE_NO) values('hun',
7 SET i=i+1;
8 END WHILE;
9 COMMIT;
10 END
```

Lúc này bảng account có tổng cộng hơn 12 triệu records.



12013963	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999992@gmail.com
12013964	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999993@gmail.com
12013965	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999994@gmail.com
12013966	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999995@gmail.com
12013967	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999996@gmail.com
12013968	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999997@gmail.com
12013969	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999998@gmail.com
12013970	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	109999999@gmail.com
12013971	hung	tran	0123456788	2022-12-01 20:42:45	abc	2000-01-01	110000000@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

account 1 x

Output



Action Output

#	Time	Action
1	20:47:09	SELECT * FROM baitaplon.account

Message

12013954 row(s) returned

Ở đây giả sử ta muốn xem thông tin của một tài khoản có `Account_ID = 11686736`.

11686731	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672760@gmail.com
11686732	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672761@gmail.com
11686733	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672762@gmail.com
11686734	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672763@gmail.com
11686735	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672764@gmail.com
11686736	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672765@gmail.com
11686737	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672766@gmail.com
11686738	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672767@gmail.com
11686739	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672768@gmail.com
11686740	hung	tran	0123456788	2022-12-01 20:42:22	abc	2000-01-01	109672769@gmail.com

Nếu ta thực hiện một câu truy vấn để lấy FName và LName của tài khoản này mà không sử dụng chỉ mục gì thì thời gian để trả về kết quả là: 6.813 giây.

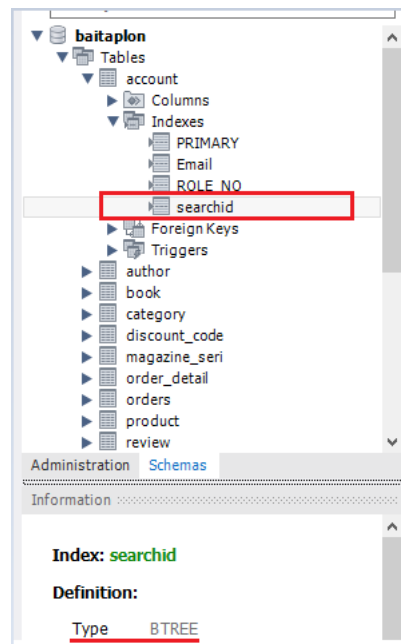
```
1 • select FName, LName from account USE INDEX ()
2   where Account_ID = 11686736
3
```

4	21:12:52	use baitaplon	0 row(s) affected	0.031 sec
5	21:12:55	select FName, LName from account USE INDEX () where Account_ID = 11686736	1 row(s) returned	6.813 sec / 0.0

Lúc này ta thử tạo một index có tên là: **searchid**

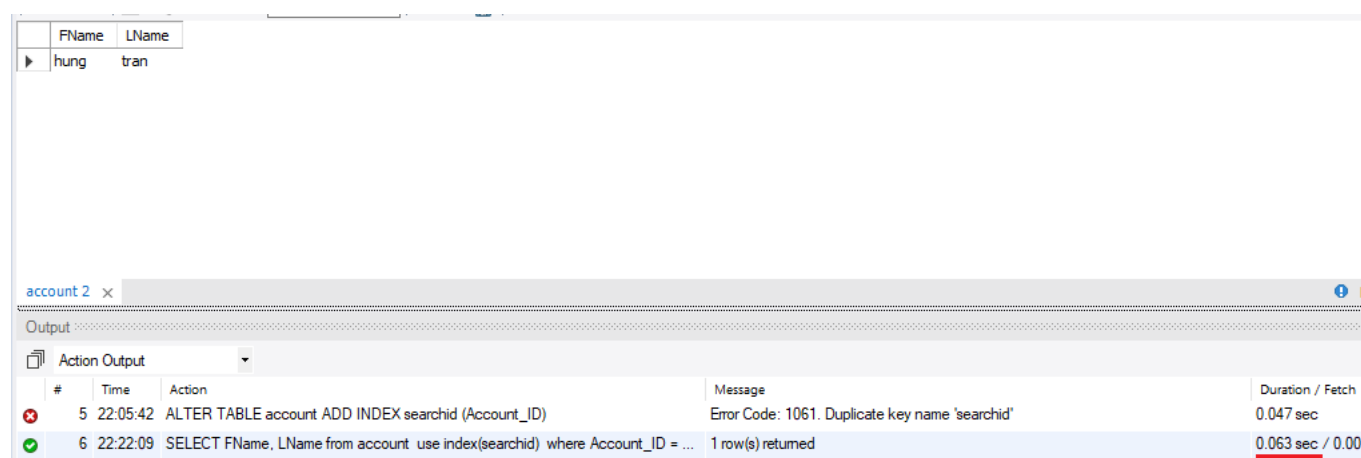
```
|| ALTER TABLE account ADD INDEX searchid (Account_ID);
```

Sau khi tạo thành công thì chỉ mục sẽ được hiển thị ở danh sách những chỉ mục trong bảng Account. Kiểu của chỉ mục này là cây cân bằng (Btree).



Bây giờ ta thử tìm kiếm lại tài khoản có ID = 11686736, thì thời gian thực thi lúc này chỉ còn: 0.063 giây (bé hơn nhiều so với 6.813 giây nếu không dùng chỉ mục)

```
|| SELECT FName, LName from account  
|| use index(searchid)  
|| where Account_ID = 11686736
```





Tiếp theo ta thử một câu truy vấn khác lấy ra FName, LName và TelephoneNum của tài khoản có Email : 106217829@gmail.com.

Result Grid	Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:		Fetch rows:	
Account_ID	FName	LName	TelephoneNum	Start_date	Address	Birthday	Email	Password	ROLE_NO	Deleted
8231792	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217821@gmail.com	123456789	4	0
8231793	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217822@gmail.com	123456789	4	0
8231794	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217823@gmail.com	123456789	4	0
8231795	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217824@gmail.com	123456789	4	0
8231796	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217825@gmail.com	123456789	4	0
8231797	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217826@gmail.com	123456789	4	0
8231798	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217827@gmail.com	123456789	4	0
8231799	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217828@gmail.com	123456789	4	0
8231800	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217829@gmail.com	123456789	4	0
8231801	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217830@gmail.com	123456789	4	0
8231802	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217831@gmail.com	123456789	4	0
8231803	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217832@gmail.com	123456789	4	0
8231804	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217833@gmail.com	123456789	4	0
8231805	hung	tran	0123456788	2022-12-01 20:38:07	abc	2000-01-01	106217834@gmail.com	123456789	4	0

Nếu ta thực hiện một câu truy vấn này mà không sử dụng chỉ mục gì thì thời gian để trả về kết quả là: 8.234 giây.

```
1 • SELECT FName, LName, TelephoneNum
2   from account
3   use index()
4  where Email = '106217829@gmail.com'
```

	FName	LName	TelephoneNum
▶	hung	tran	0123456788

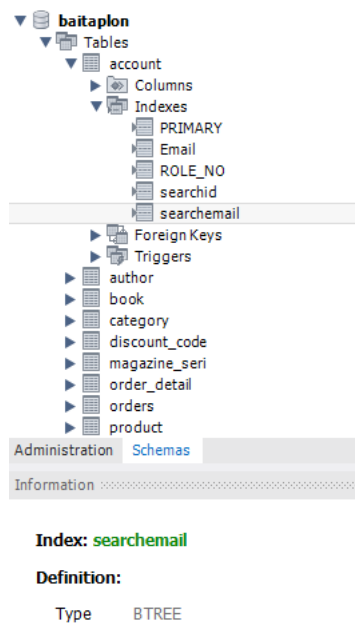
#	Time	Action	Message	Duration / Fetch
2	23:15:45	use baitaplon	0 row(s) affected	0.000 sec
3	23:15:51	SELECT FName, LName, TelephoneNum from account use index() where Email = '...	1 row(s) returned	8.234 sec / 0.0

Lúc này ta thử tạo một index có tên là: searchemail

```
|| create index searchemail on account(Email);
```

Sau khi tạo thành công thì chỉ mục sẽ được hiển thị ở danh sách những chỉ mục trong bảng Account.Kiểu của chỉ mục này là cây cân bằng (Btree).





Bây giờ ta thử lại câu truy vấn và sử dụng chỉ mục searchemail vừa mới tạo, thì thời gian thực thi lúc này chỉ còn: 0.093 giây (bé hơn nhiều so với 8.234 giây nếu không dùng chỉ mục)

```
SELECT FName, LName, TelephoneNum from account
use index(searchemail)
where Email = '106217829@gmail.com'
```

Output				
Action Output				
#	Time	Action	Message	Duration / Fetch
6	23:33:05	create index searchemail on account(Email)	Error Code: 1061. Duplicate key name 'searchemail'	0.000 sec
7	23:39:40	SELECT FName, LName, TelephoneNum from account use index(searchemail) w...	1 row(s) returned	0.093 sec / 0.000 sec