

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ UEH**



**DỰ ÁN CUỐI KỲ
CƠ SỞ DỮ LIỆU**

**XÂY DỰNG MÔ HÌNH CƠ SỞ DỮ LIỆU MUA BÁN XE CỦA CỬA
HÀNG HONDA**

Giảng viên bộ môn : Viên Thanh Nhã

Lớp học phần : 22C1INF50900603

Họ và tên sinh viên 1 : Nguyễn Quang Nhật

Mã số sinh viên 1 : 31211027658

Họ và tên sinh viên 2 : Phan Đình Nhân

Mã số sinh viên 2 : 31211027657

TP. HỒ CHÍ MINH, NĂM 2022

MỤC LỤC

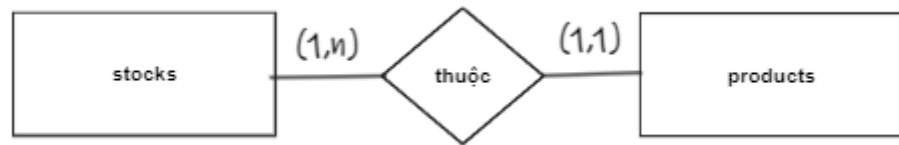
1. Diễn tả qui tắc mô tả, ràng buộc về bản số và xác định các thuộc tính để hình thành thực thể	1
a. Diễn tả các qui tắc mô tả, ràng buộc về bản số	1
b. Xác định thuộc tính hình thành thực thể.....	2
2. Xây dựng mô hình thực thể kết hợp (ERD).....	6
3. Chuyển mô hình mức quan niệm sang mô hình quan hệ.....	6
4. Chuẩn hoá	8
a. Dữ liệu các bảng	8
b. Dạng chuẩn 1 (1NF)	12
c. Dạng chuẩn 2 (2NF)	13
d. Dạng chuẩn 3 (3NF)	13
e. Dạng chuẩn Boyce-codd (BCNF)	13
5. Xây dựng các phép toán đại số quan hệ	13
a. Phép chọn.....	13
b. Phép chiếu	14
c. Phép giao.....	14
d. Phép hội.....	15
e. Phép trừ.....	16
f. Phép Tích Decac	16
g. Phép kết nối	17
h. Gom nhóm.....	17
i. Gom nhóm có điều kiện	18
6. Lệnh mô tả dữ liệu: DDL	18
a. Tạo database	18
b. Tạo các table dựa trên các quan hệ đã cho trên database	18
c. Thay đổi cấu trúc bảng nếu có	21
7. Lệnh thao tác dữ liệu: DDL	21
a. Thêm dữ liệu vào các table bằng lệnh SQL.....	21
b. Update dữ liệu vào các table bằng lệnh SQL	29
c. Xóa dữ liệu ở các bảng table bằng lệnh SQL	30
8. Lệnh truy vấn dữ liệu: SQL.....	30
a. Truy vấn 1 bảng.....	30
b. Truy vấn nhiều bảng (Phép kết).....	31
c. Truy vấn có điều kiện (and, or, like, between, ...).....	34
d. Truy vấn tính toán.....	36
e. Truy vấn có gom nhóm (group by)	39
f. Truy vấn gom nhóm có điều kiện (having).....	42
g. Truy vấn có sử dụng phép giao, hội, trừ	44
h. Truy vấn con.....	46
i. Truy vấn chéo	49

9. Viết store procedure và fuction	52
10. Viết trigger ràng buộc dữ liệu cho các bảng	55
11. Phân quyền.....	57
12. Sao lưu dữ liệu	57

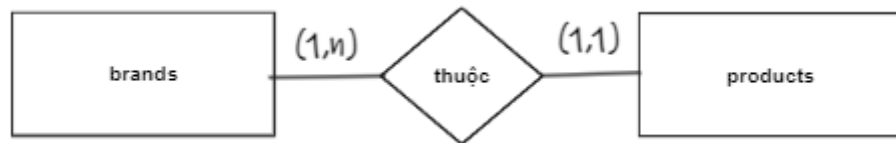
1. Diễn tả qui tắc mô tả, ràng buộc về bản số và xác định các thuộc tính để hình thành thực thể

a. Diễn tả các qui tắc mô tả, ràng buộc về bản số

1. Tối thiểu một kho hàng sẽ chứa được tối đa n xe. Tối thiểu một xe sẽ thuộc tối đa 1 kho hàng. => Thực thể **“stocks”** và **“products”** có quan hệ 1-N



2. Tối thiểu một thương hiệu sẽ có tối đa n xe. Tối thiểu một xe sẽ thuộc tối đa 1 thương hiệu. => Thực thể **“brands”** và **“products”** có quan hệ 1-N



3. Tối thiểu một phân khúc sẽ có tối đa n xe. Tối thiểu một xe sẽ thuộc tối đa 1 phân khúc. => Thực thể **“categories”** và **“products”** có quan hệ 1-N



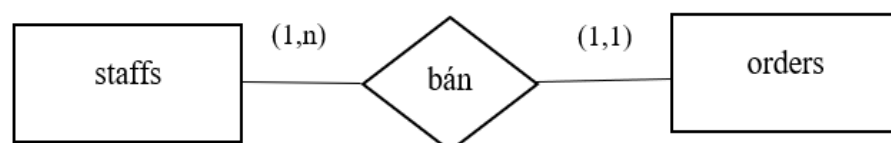
4. Tối thiểu một đơn hàng sẽ có tối đa n sản phẩm. Tối thiểu một sản phẩm sẽ thuộc tối đa n đơn hàng. => Thực thể **“orders”** và **“products”** có quan hệ N-M tạo ra bảng phụ là **“order_items”**



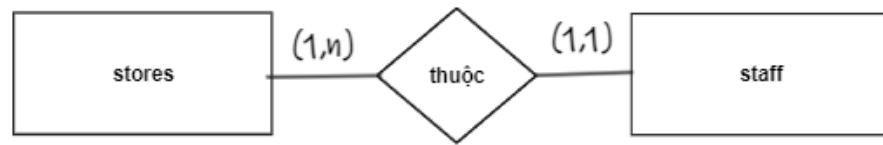
5. Tối thiểu một khách hàng sẽ mua được tối đa n đơn hàng. Tối thiểu một đơn hàng sẽ được mua tối đa bởi 1 khách hàng. => Thực thể **“customers”** và **“orders”** có quan hệ 1-N



6. Tối thiểu một nhân viên sẽ bán được tối đa n đơn hàng. Tối thiểu một đơn hàng sẽ được bán bởi tối đa 1 nhân viên. => Thực thể **“staffs”** và **“orders”** có quan hệ 1-N



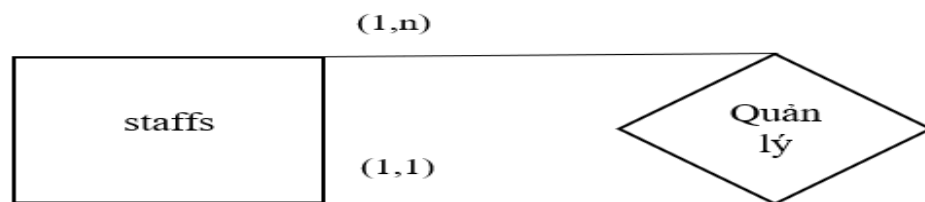
7. Tối thiểu một cửa hàng sẽ có tối đa n nhân viên. Tối thiểu một nhân viên sẽ thuộc tối đa 1 cửa hàng. => Thực thể “stores” và “staffs” có quan hệ 1-N



8. Tối thiểu một nhân viên sẽ có tối đa n thân nhân. Tối thiểu một thân nhân sẽ thuộc tối đa 1 nhân viên. => Thực thể “staffs” và “relatives” có quan hệ 1-N



9. Tối thiểu một nhân viên sẽ có tối đa 1 người quản lý. Tối thiểu 1 người quản lý sẽ quản lý tối đa n nhân viên. => Thực thể “staffs” có quan hệ 1-N với chính nó.



b. Xác định thuộc tính hình thành thành thực thể

Trong mô hình quản lý dữ liệu mua bán xe của cửa hàng Honda này gồm 10 thực thể với các thuộc tính:

- i. Thương hiệu: mã thương hiệu và tên thương hiệu.
- ii. Phân khúc: mã phân khúc và tên phân khúc.
- iii. Kho: mã kho và tình trạng kho (còn trống hay không)
- iv. Sản phẩm: mã sản phẩm, tên sản phẩm, mã thương hiệu, mã phân khúc, mã kho, năm sản xuất, giá bán và số lượng trong kho.
- v. Cửa hàng: mã cửa hàng, tên cửa hàng, số điện thoại, mã người quản lý và địa chỉ.
- vi. Khách hàng: mã khách hàng, tên khách hàng, email, số điện thoại
- vii. Nhân viên: mã nhân viên, tên nhân viên, số điện thoại, lương cơ bản và mã cửa hàng.
- viii. Đơn hàng: mã đơn hàng, ngày bán, mã nhân viên, mã khách hàng
- ix. Đơn hàng_Sản phẩm: mã sản phẩm, mã đơn hàng
- x. Thân nhân: mã nhân viên(của thân nhân), tên thân nhân, mối quan hệ

Giải thích các thực thể

Các khái niệm cơ bản:

- Thực thể (Entity): Là khái niệm mô tả một lớp các đối tượng có đặc trưng mà chúng ta cần quan tâm. Các thực thể là đối tượng cụ thể hoặc trừu tượng. Trong sơ đồ thì thực thể thường được ký hiệu là hình chữ nhật.
- Thuộc tính (Attribute): Là các tính chất, đặc điểm chung của đối tượng, nó là một giá trị dùng để mô tả một đặc trưng nào đó của một thực thể. Thuộc tính có thể là đơn trị, đa trị (lập), hoặc phức hợp.

Thực thể “**brands**”: mỗi thương hiệu xe sẽ có mã thương hiệu và tên riêng: brand_id, brand_name



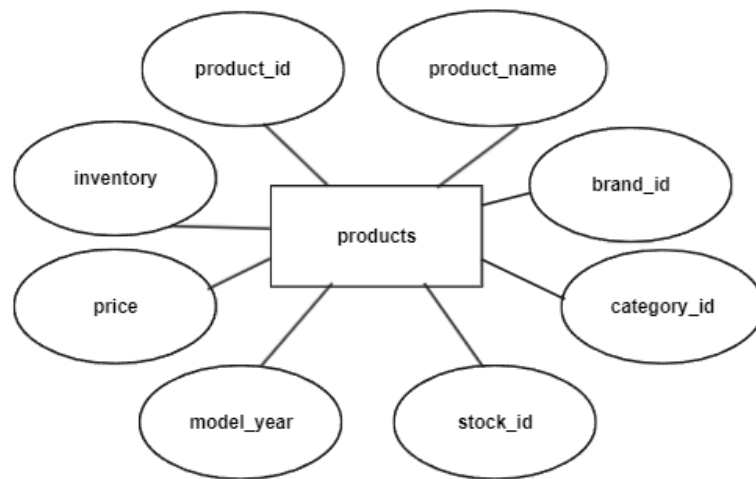
Thực thể “**categories**”: mỗi xe sẽ được chia vào một phân khúc khác nhau có mã và tên phân biệt: categories_id, categories_name



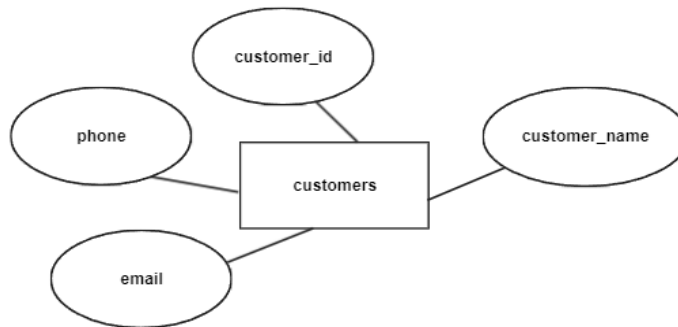
Thực thể “**stocks**”: mỗi kho của hãng sẽ được cấp một mã kho cũng như thông tin về khả năng nhận thêm xe: stock_id, vacant



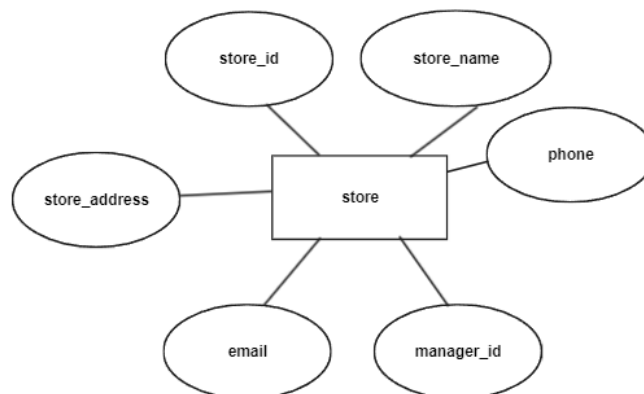
Thực thể “**products**”: bao gồm thông tin cơ bản về các sản phẩm: product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory.



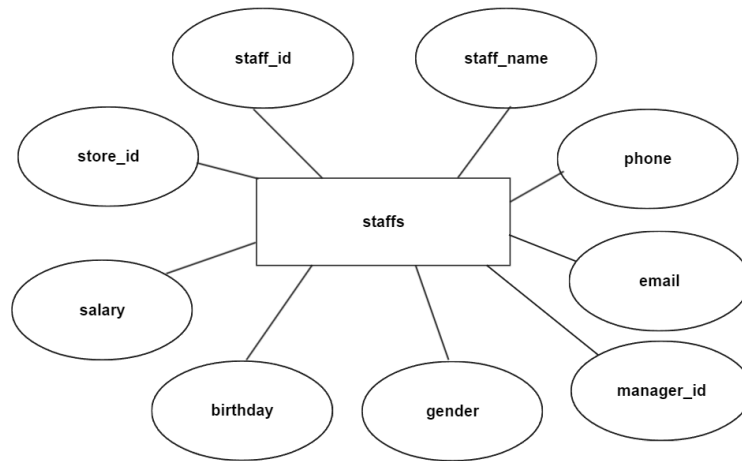
Thực thể “**customers**”: chứa thông tin cần thiết của khách hàng: customer_id, customer_name, phone, email.



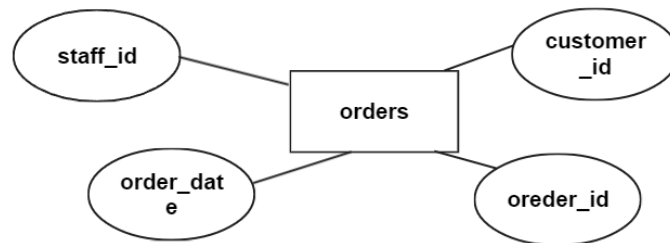
Thực thể “**store**”: bao gồm thông tin về các cửa hàng cũng như cách liên lạc và người quản lý: store_id, store_name, phone, email, store_address, manager_id



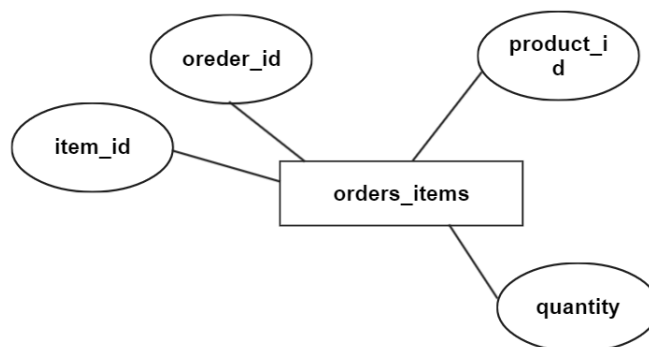
Thực thể “**staffs**”: chứa các thông tin cơ bản của các nhân viên: staff_id, staff_name, store_id, phone, email, salary, birthday, gender, manager_id



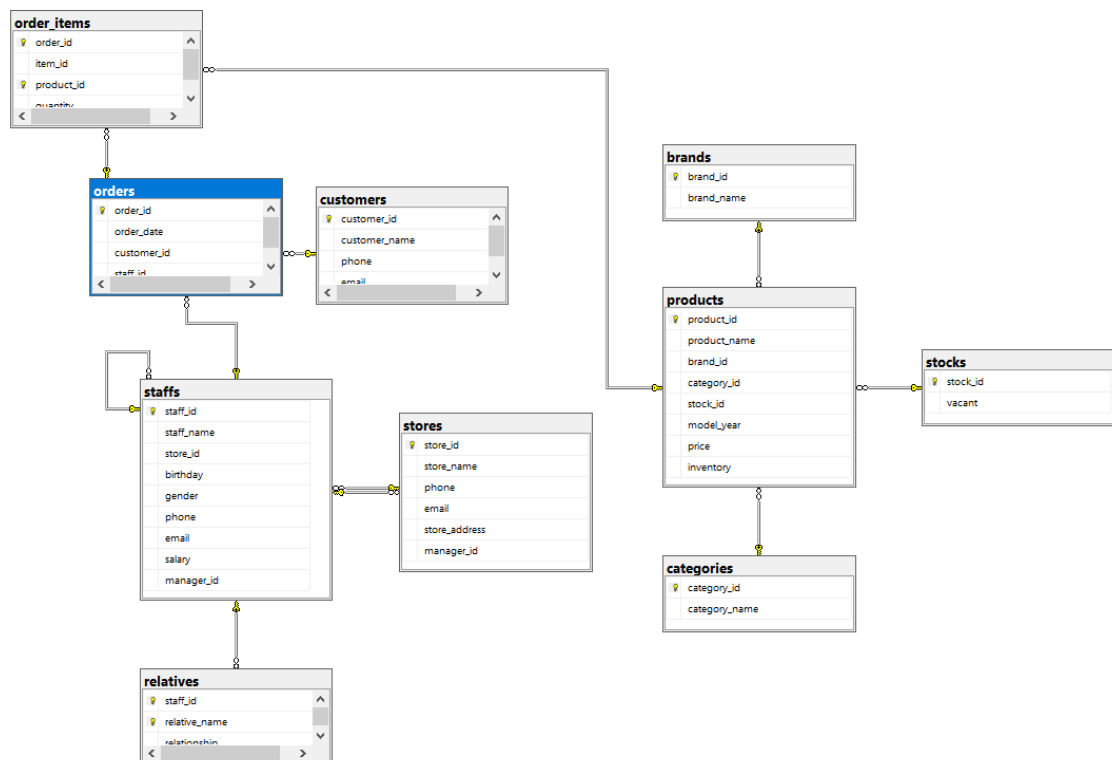
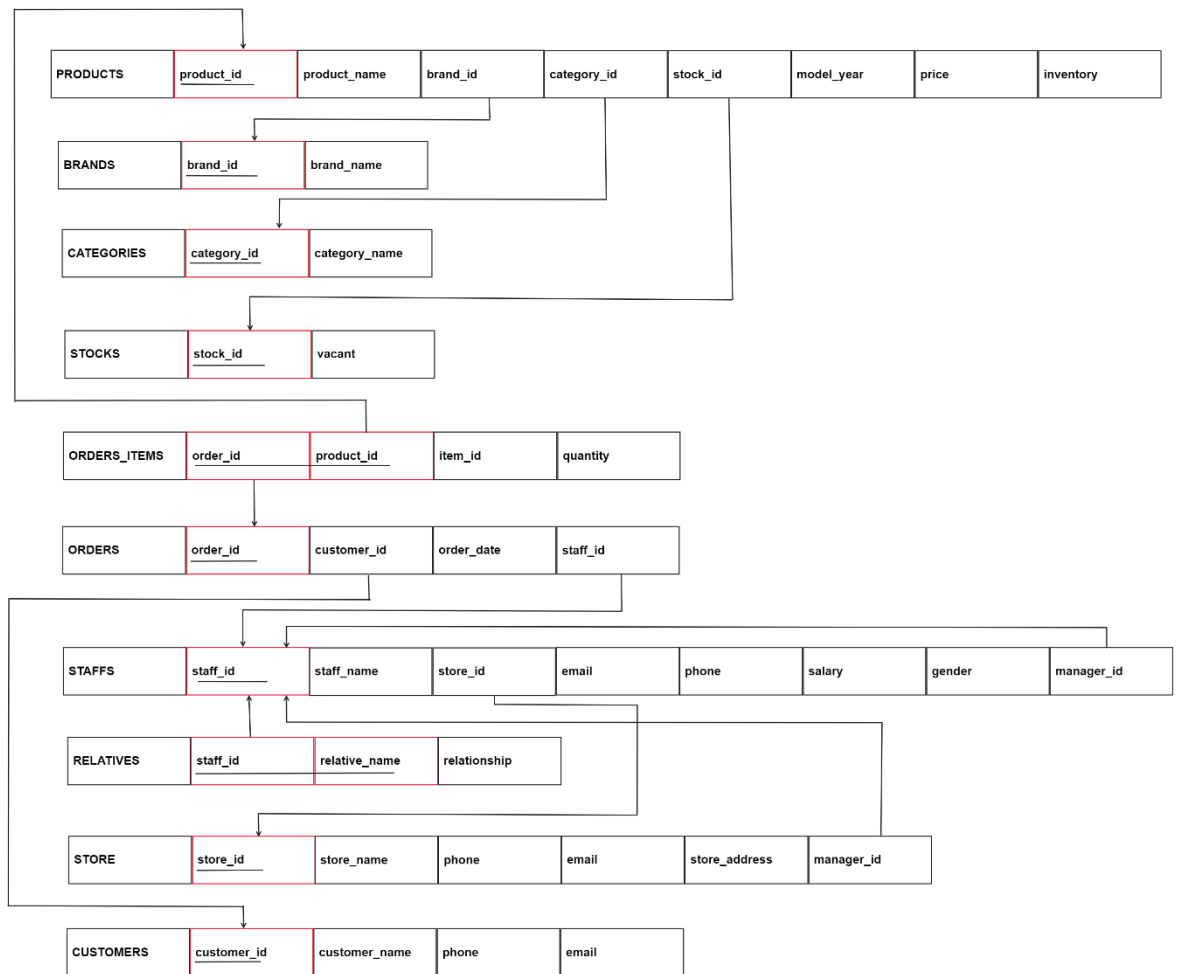
Thực thể “**orders**”: Mỗi đơn bán hàng gồm những thông tin như: order_id, customer_id, order_date, staff_id



Thực thể “**orders_items**”: đơn lấy sản phẩm có thông tin: order_id, product_id, item_id, quantity



Thực thể “**relatives**”: chứa thông tin thân nhân của nhân viên: staff_id, relative_name, relationship



4. Chuẩn hoá

a. Dữ liệu các bảng

Categories: (Với category_id là khóa chính)

category_id	category_name
1	Adventure
2	Cruiser
3	Naked Bike
4	Scooter
5	Sport Bike
6	Sport Touring
7	Touring
8	Underbone

Brands: (Với brand_id là khóa chính)

brand_id	brand_name
1	BMW
2	Ducati
3	Harley - Davidson
4	Honda
5	Kawasaki
6	Piaggio
7	Suzuki
8	Yamaha

Stocks: (Với stock_id là khóa chính)

stock_id	vacant
1	Y
2	Y
3	N

Products: (Với product_id là khóa chính)

product_id	product_name	brand_id	category_id	stock_id	model_year	price	inventory
1	Ducati Multistrada 950 S	2	1	1	2021	558000000	6
2	BMW R1200 GS Adventure	1	1	1	2017	719000000	4
3	Harley-Davidson Sportster SuperLow	3	2	2	2020	356000000	8
4	Kawasaki Vulcan S	5	2	2	2021	241000000	11
5	Suzuki GSX-S1000	7	3	2	2022	399000000	7
6	Vespa GTS 150	6	4	2	2022	115000000	15
7	Yamaha NVX	8	4	3	2020	52000000	19
8	Air Blade 125/160	4	4	3	2022	42000000	24
9	Suzuki Hayabusa GSX 1300R	7	5	1	2021	750000000	5
10	BMW S1000RR	1	5	1	2020	949000000	2
11	Kawasaki ZX-10R	5	5	1	2022	729000000	3
12	Harley-Davidson Road King	3	7	1	2021	965000000	2
13	Honda GoldWing	4	7	1	2021	1230000000	1
14	BMW K-1600 GTL	1	6	1	2022	634000000	7
15	Kawasaki Concours	5	6	2	2020	384000000	10
16	Yamaha Z125	8	8	2	2017	270000000	9
17	Honda Winner X	4	8	3	2022	50060000	20
18	Suzuki Raider 150 FI	7	8	3	2017	49190000	14

Customers: (Với customer_id là khóa chính)

customer_id	customer_name	phone	email
1	Nguyễn Thanh Vân	0912151114	vannguyen@gmail.com
2	Lê Văn Tạo	0954544154	taole@gmail.com
3	Đỗ Duy Trung	0931184518	trungdo@gmail.com
4	Lê Minh Nguyệt	0715166518	nguyetle@gmail.com
5	Nguyễn Kim Liên	0961484123	liennguyen@gmail.com
6	Ngô Văn Sở	0716151656	songo@gmail.com
7	Đinh Công Minh	0921612161	minhding@gmail.com
8	Đặng Văn Tài	0914653165	taidang@gmail.com
9	Trần Hạo Sơn	0765164516	sonhao@gmail.com
10	Nguyễn Công Nghĩa	0961646126	nghianguyen@gmail.com
11	Đỗ Tấn Tài	0764151545	taido@gmail.com
12	Phạm Hoàng Long	0961646546	longpham@gmail.com
13	Đặng Việt Trung	0713216489	trungdang@gmail.com
14	Lê Công Minh	0961618456	minhle@gmail.com
15	Trần Hoài Sơn	0754646465	sontran@gmail.com

Stores: (Với store_id là khóa chính)

store_id	store_name	phone	email	store_address	manager_id
1	Dũng Sơn	18006420	info@dungson.com	213 Tôn Đức, Phường 14, Quận 4, TP.HCM	4
2	Phát Tài	18004325	info@phattai.com	123 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM	2
3	An Khang	18007846	info@ankhang.com	442 Phạm Ngũ Lão, Phường Bến Nghé, Quận 1, TP.HCM	3

Staffs: (Với staff_id là khóa chính)

staff_id	staff_name	store_id	birthday	gender	phone	email	salary	manager_id
1	Đặng Công Thành	1	1998-01-01 00:00:00.000	Nam	0324546464	thanh dang@gmail.com	8000000	4
2	Đỗ Phương Nghi	2	1992-12-21 00:00:00.000	Nữ	0334446485	nghido@gmail.com	15000000	NULL
3	Đỗ Thị Thủy Vy	3	1994-04-03 00:00:00.000	Nữ	0346484838	vydo@gmail.com	15000000	NULL
4	Nguyễn Hoàn Mỹ	1	1991-07-03 00:00:00.000	Nữ	0355484354	mynguyen@gmail.com	15000000	NULL
5	Nguyễn Ngọc Hằng	2	1999-05-07 00:00:00.000	Nữ	0364464688	hangnguyen@gmail.com	8000000	2
6	Bùi Minh Đạt	3	2000-08-19 00:00:00.000	Nam	0375164845	datbui@gmail.com	8000000	3
7	Võ Lê Khang	1	2001-03-24 00:00:00.000	Nam	0381654861	khangvo@gmail.com	8000000	4
8	Đỗ Tuấn Cường	2	1990-11-30 00:00:00.000	Nam	0395485462	cuongdo@gmail.com	8000000	2
9	Trần Hạo Long	3	1997-06-04 00:00:00.000	Nam	0354546548	longtran@gmail.com	8000000	3
10	Phan Trung Tin	2	1998-07-02 00:00:00.000	Nam	0345134846	tinphan@gmail.com	8000000	2

Relatives: (Với staff_id và relative_name là khóa chính)

staff_id	relative_name	relationship
1	Đặng Việt Anh	Cha
2	Đỗ Phương Hoài	Cha
2	Lê Nhật Anh	Mẹ
3	Đỗ Trọng Nguyên	Cha
3	Phạm Thu Sương	Mẹ
4	Nguyễn Trung Hậu	Cha
4	Trần Huyền Trang	Mẹ
5	Lê Thị Mỹ Kim	Mẹ
6	Bùi Minh Trung	Cha
7	Võ Anh Tài	Cha
8	Nguyễn Thị Hồng	Mẹ
9	Trần Thanh Thiên	Cha
10	Lê Mỹ Trúc	Mẹ

Orders: (Với order_id là khóa chính)

order_id	order_date	customer_id	staff_id
1	2022-04-05 00:00:00.000	1	1
2	2022-05-12 00:00:00.000	2	4
3	2022-06-15 00:00:00.000	4	7
4	2022-04-18 00:00:00.000	3	1
5	2022-05-14 00:00:00.000	6	4
6	2022-06-23 00:00:00.000	7	4
7	2022-04-30 00:00:00.000	5	4
8	2022-05-01 00:00:00.000	4	7
9	2022-06-14 00:00:00.000	6	2
10	2022-04-15 00:00:00.000	7	5
11	2022-05-19 00:00:00.000	8	8
12	2022-06-07 00:00:00.000	9	10
13	2022-04-04 00:00:00.000	10	2
14	2022-05-04 00:00:00.000	12	2
15	2022-06-09 00:00:00.000	15	5
16	2022-04-30 00:00:00.000	13	3
17	2022-05-31 00:00:00.000	14	3
18	2022-06-22 00:00:00.000	8	6
19	2022-04-03 00:00:00.000	14	9
20	2022-05-07 00:00:00.000	11	3
21	2022-06-01 00:00:00.000	8	9
22	2022-04-09 00:00:00.000	13	3
23	2022-05-18 00:00:00.000	9	3
24	2022-06-27 00:00:00.000	11	6
25	2022-04-29 00:00:00.000	4	9

Order_items: (Với order_id và product_id là khóa chính)

order_id	item_id	product_id	quantity
1	1	5	1
1	2	8	3
2	1	1	1
3	1	7	2
4	1	9	1
4	2	18	3
5	1	2	1
5	2	14	1
6	1	17	2
7	1	13	1
8	1	11	1
9	2	6	1
9	1	12	1
10	1	3	1
11	1	4	1
12	1	10	1
12	2	15	1
12	3	16	1
13	1	7	2
13	2	18	2
14	1	9	1
15	1	4	1
16	1	2	1
17	2	3	1
17	1	4	1
18	1	6	1
19	1	11	1
20	2	5	1
20	3	8	2
20	1	9	1
21	1	1	1
22	1	16	1
23	1	17	2

b. **Dạng chuẩn 1 (1NF)**

Định nghĩa: Lược đồ quan hệ Q ở dạng 1NF nếu tất cả thuộc tính của Q đều là thuộc tính đơn/nguyên tố. Đồng thời để một lược đồ cơ sở dữ liệu C ở dạng 1NF thì tất cả Qi của C đều phải ở dạng 1NF.

Dễ dàng thấy được tất cả thuộc tính thuộc các thực thể trong hệ cơ sở dữ liệu qlyxemay_nhat_nhan đều là thuộc tính đơn.

⇒ Hệ cơ sở dữ liệu qlyxemay_nhat_nhan đạt chuẩn 1.

c. **Dạng chuẩn 2 (2NF)**

Định nghĩa: Lược đồ quan hệ Q ở dạng 2NF nếu ở 1NF và tất cả thuộc tính không khóa đều phụ thuộc đầy đủ vào khóa. Đồng thời để một lược đồ cơ sở dữ liệu C ở dạng 2NF thì tất cả Qi của C đều phải ở dạng 2NF.

Lấy ví dụ thực thể staffs đặt các thuộc tính staff_id, staff_name, store_id, birthday, gender, phone, email, salary, manager_id lần lượt là ABCDEFGHI.

Ta có: staffs(ABCDEFGHI) với tập phụ thuộc hàm $F = \{A \rightarrow BCDEFGHI\}$

Để thấy các thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào khóa. Chứng minh tương tự ở các bảng còn lại.

⇒ Hệ cơ sở dữ liệu glyxemay_nhat_nhan đạt chuẩn 2.

d. **Dạng chuẩn 3 (3NF)**

Định nghĩa: Lược đồ quan hệ Q ở dạng chuẩn 3NF nếu ở 2NF và tất cả thuộc tính không khóa không phụ thuộc bắc cầu vào khóa. Hay nói cách khác các thuộc tính không khóa phải phụ thuộc trực tiếp vào khóa. Đồng thời để một lược đồ cơ sở dữ liệu C ở dạng 3NF thì tất cả Qi của C đều phải ở dạng 3NF.

Lấy ví dụ thực thể staffs như trên. Dễ dàng thấy các thuộc tính không khóa (BCDEFGHI) không chỉ phụ thuộc đầy đủ vào khóa (A) mà còn phụ thuộc trực tiếp với khóa. Chứng minh tương tự với các bảng khác.

⇒ Hệ cơ sở dữ liệu glyxemay_nhat_nhan đạt chuẩn 3.

e. **Dạng chuẩn Boyce-codd (BCNF)**

Định nghĩa: Lược đồ quan hệ Q ở BCNF nếu ở dạng 1NF và tất cả phụ thuộc hàm không hiển nhiên $X \rightarrow Y$ của F^+ có X là một siêu khóa (X chứa một khóa nào đó). Đồng thời để một lược đồ cơ sở dữ liệu C ở dạng 3NF thì tất cả Qi của C đều phải ở dạng BCNF.

Lại lấy tiếp ví dụ về thực thể staffs. Từ tập phụ hàm $F = \{A \rightarrow BCDEFGHI\}$ ta có thể thấy tập bao đóng của tập phụ hàm F cũng bằng chính nó. $F^+ = F = \{A \rightarrow BCDEFGHI\}$. Bản thân A là khóa chính cũng là siêu khóa. Tương tự với các bảng còn lại.

⇒ Hệ cơ sở dữ liệu glyxemay_nhat_nhan đạt chuẩn Boyce-codd.

5. **Xây dựng các phép toán đại số quan hệ**

a. **Phép chọn**

- Ví dụ 1: Tìm ra những nhân viên trong cửa hàng 1. Thông tin hiện thị gồm staff_id, staff_name, store_id

$$\pi_{\text{staff_id, staff_name, store_id}} (\sigma_{\text{store_id} = 1} (\text{staffs}))$$

- Ví dụ 2: Tìm ra những nhân viên có tuổi lớn hơn 25 và làm ở cửa hàng Phát Tài. Thông tin hiển thị gồm staff_id, staff_name, age

$$\pi_{\text{staff_id, staff_name, age}}(\sigma_{(\text{YEAR}(\text{GETDATE>()) - \text{YEAR}(\text{birthday})) > 25 \wedge \text{store_id} = 2}(\text{staffs}))$$

- Ví dụ 3: Tìm ra những kho nào chưa đầy. Thông tin hiển thị gồm stock_id

$$\pi_{\text{stock_id}}(\sigma_{\text{vacant} = 'Y'}(\text{stocks}))$$

- Ví dụ 4: Tìm ra những khách hàng đã mua hàng trong tháng 5. Thông tin hiển thị gồm customer_id, customer_name, phone, email

$$\pi_{\text{customer_id, customer_name, phone, email}}(\sigma_{\text{MONTH}(\text{order_date})=5}(\text{customers } c \bowtie (\text{c.customer_id}=\text{o.customer_id}) \text{ orders } o))$$

b. Phép chiếu

- Ví dụ 1: Lấy ra các thuộc tính product_id, product_name, price

$$\pi_{\text{product_id, product_name, price}}(\text{products})$$

- Ví dụ 2: Tìm các sản phẩm đang đặt ở kho 1. Lấy các thuộc tính product_id, product_name, price.

$$\pi_{\text{product_id, product_name, price}}(\sigma_{\text{stock_id} = 1}(\text{products}))$$

- Ví dụ 3: Tìm các sản phẩm đang đặt ở kho 1 có lượng tồn kho nhỏ hơn 5 . Lấy các thuộc tính product_id, product_name, price.

$$\pi_{\text{product_id, product_name, price}}(\sigma_{\text{stock_id} = 1 \wedge \text{inventory} < 5}(\text{products}))$$

- Ví dụ 4: Tìm các sản phẩm đang đặt ở kho 1 có lượng tồn kho nhỏ hơn 5 . Lấy các thuộc tính product_id, product_name, price.

$$\pi_{\text{product_id, product_name, price}}(\sigma_{\text{stock_id} = 1 \wedge \text{inventory} < 5}(\text{products}))$$

c. Phép giao

- Ví dụ 1: Liệt kê các đơn hàng đã mua sản phẩm Yamaha NVX và Kawasaki Vulcan S. Thông tin hiển thị gồm: order_id

$$\pi_{\text{order_id}}(\sigma_{\text{product_id} = 4}(\text{orders_items})) \cap \pi_{\text{order_id}}(\sigma_{\text{product_id} = 7}(\text{orders_items}))$$

- Ví dụ 2: Liệt kê các thân nhân là mẹ của nhân viên và nhân viên đó làm ở cửa hàng Dũng Sơn. Thông tin hiển thị staff_id, relative_name

$$\pi_{\text{staff_id, relative_name}}(\sigma_{\text{relationship} = \text{'N'Me'}}(\text{relatives } r \bowtie_{(r.\text{staff_id}=s.\text{staff_id})} \text{staffs } s)) \cap \pi_{\text{staff_id, relative_name}}(\sigma_{\text{store_id} = 2}(\text{relatives } r \bowtie_{(r.\text{staff_id}=s.\text{staff_id})} \text{staffs } s))$$

- Ví dụ 3: Tìm những loại xe có phân khúc Adventure của hãng Ducati. Thông tin hiển thị product_id, product_name

$$\pi_{\text{product_id, product_name}}(\sigma_{\text{category_id} = 1}(\text{products } p \bowtie_{(p.\text{category_id} = c.\text{category_id})} \text{categories } c)) \cap \pi_{\text{product_id, product_name}}(\sigma_{\text{brand_id} = 1}(\text{products } p \bowtie_{(p.\text{brand_id} = b.\text{brand_id})} \text{brands } b))$$

- Ví dụ 4: Tìm những xe ở kho 1 và kho 2. Thông tin hiển thị product_id, product_name, price

$$\pi_{\text{product_id, product_name, price}}(\sigma_{\text{stock_id} = 1}(\text{products } p \bowtie_{(p.\text{stock_id}=s.\text{stock_id})} \text{stocks } s)) \cap \pi_{\text{product_id, product_name, price}}(\sigma_{\text{stock_id} = 2}(\text{products } p \bowtie_{(p.\text{stock_id}=s.\text{stock_id})} \text{stocks } s))$$

d. Phép hội

- Ví dụ 1: Tìm các hóa đơn thực hiện trong tháng 4 hoặc tháng 5. Lấy các thuộc tính order_id, order_date

$$\pi_{\text{order_id, order_date}}(\sigma_{\text{MONTH}(\text{order_date}) = 4}(\text{orders})) \cup \pi_{\text{order_id, order_date}}(\sigma_{\text{MONTH}(\text{order_date}) = 5}(\text{orders}))$$

- Ví dụ 2: Tìm các sản phẩm có giá lớn hơn 1 tỷ hoặc có giá nhỏ hơn 100 triệu. Lấy các thuộc tính product_id, product_name, price.

$$\pi_{\text{product_id, product_name, price}}(\sigma_{\text{price} > 1000000000}(\text{products})) \cup \pi_{\text{product_id, product_name, price}}(\sigma_{\text{price} < 100000}(\text{products}))$$

- Ví dụ 3: Tìm những nhân viên có tuổi dưới 24 hoặc trên 30. Lấy các thuộc tính staff_id, staff_name, age

$$\pi_{\text{staff_id, staff_name, YEAR}(\text{GETDATE}()) - \text{YEAR}(\text{birthday}) \text{ AS age}}(\sigma_{\text{age} < 24}(\text{staffs})) \cup \pi_{\text{staff_id, staff_name, YEAR}(\text{GETDATE}()) - \text{YEAR}(\text{birthday}) \text{ AS age}}(\sigma_{\text{age} > 30}(\text{staffs}))$$

- Ví dụ 4: Tìm các sản phẩm thuộc loại xe 1 hoặc loại xe 8. Lấy các thuộc tính product_id, product_name, category_id, price

$$\pi_{\text{product_id, product_name, category_id, price}}(\sigma_{\text{category_id} = 1}(\text{products})) \cup \pi_{\text{product_id, product_name, category_id, price}}(\sigma_{\text{category_id} = 8}(\text{products}))$$

e. Phép trừ

- Ví dụ 1: Tìm ra những nhân viên có lương cơ bản lớn hơn 8.000.000vnd. Thông tin hiển thị staff_id, staff_name, salary

$$\pi_{\text{staff_id, staff_name, salary}}(\text{staffs}) - \pi_{\text{staff_id, staff_name, salary}}(\sigma_{\text{salary} \leq 8000000}(\text{staffs}))$$

- Ví dụ 2: Tìm ra những nhân viên làm ở cửa hàng An Khang. Thông tin hiển thị staff_id, staff_name, salary

$$\pi_{\text{staff_id, staff_name, salary}}(\text{staffs}) - \pi_{\text{staff_id, staff_name, salary}}(\sigma_{\text{store_id} = 1 \wedge \text{store_id} = 2}(\text{staffs} \bowtie (\text{s.store_id} = \text{sto.store_id}) \text{ stores sto}))$$

- Ví dụ 3: Tìm ra những xe có giá trên 1 tỷ. Thông tin hiển thị product_id, product_name, price

$$\pi_{\text{product_id, product_name, price}}(\text{products}) - \pi_{\text{product_id, product_name, price}}(\sigma_{\text{price} \leq 1000000000}(\text{products}))$$

- Ví dụ 4: Tìm ra những nhân viên không bán được đơn hàng nào trong tháng 4. Thông tin hiển thị staff_id, staff_name, salary

$$\pi_{\text{staff_id, staff_name, salary}}(\text{staffs}) - \pi_{\text{staff_id, staff_name, salary}}(\sigma_{\text{MONTH}(\text{order_date}) = 5 \wedge \text{MONTH}(\text{order_date}) = 6}(\text{staffs} \bowtie (\text{s.staff_id} = \text{o.staff_id}) \text{ orders o}))$$

f. Phép Tích Decac

- Ví dụ 1: Tìm cửa hàng tương ứng với mỗi hóa đơn được bán. Lấy các thuộc tính order_id, store_id

$$\pi_{\text{order_id, store_id}}(\sigma_{\text{o.staff_id} = \text{s.staff_id}}(\text{orders AS o X staffs AS s}))$$

- Ví dụ 2: Tìm nhân viên quản lý ở mỗi cửa hàng. Lấy các thuộc tính store_id, store_name, manager_id, manager_name

$$\pi_{\text{so.store_id, store_name, so.manager_id, staff_name AS manager_name}}(\sigma_{\text{so.manager_id} = \text{staff_id}}(\text{stores AS so X staffs}))$$

- Ví dụ 3: Tìm những viên khai báo người thân là mẹ. Lấy các thuộc tính staff_id, staff_name, relative_name

$$\pi_{\text{s.staff_id, staff_name, relative_name}}(\sigma_{\text{r.staff_id} = \text{s.staff_id} \wedge \text{relationship} = \text{N'Mẹ'}}(\text{relatives AS r X staffs AS s}))$$

- Ví dụ 4: Tìm tên người thân của 3 nhân viên cấp quản lý. Lấy các thuộc tính manager_id, manager_name, relative_name, relationship

$\pi_{so.manager_id, staff_name \text{ AS } manager_name, relative_name, relationship}(\sigma_{so.manager_id = sa.staff_id \wedge so.manager_id = r.staff_id}(\text{relatives AS r X staffs AS sa X stores AS so}))$

g. Phép kết nối

- Ví dụ 1: In ra danh sách các hoá đơn. Thông tin hiển thị. Order_id, staff_name, customer_name

$\pi_{order_id, staff_name, customer_name}(\text{staffs} \bowtie_{(staff_id)} \text{orders} \bowtie_{(customer_id)} \text{customer})$

- Ví dụ 2: In ra danh sách các sản phẩm. Thông tin hiển thị product_id. Product_name, brand_name, category_name

$\pi_{product_id, product_name, brand_name, category_name}(\text{brands} \bowtie_{(brand_id)} \text{product} \bowtie_{(category_id)} \text{categories})$

- Ví dụ 3: In ra danh sách các nhân viên. Thông tin hiển thị staff_id, staff_name, store_id

$\pi_{staff_id, staff_name, store_name}(\text{staffs s} \bowtie_{(s.staff_id = sto.staff_id)} \text{stores sto})$

- Ví dụ 4: Tìm ra những sản phẩm thuộc phân khúc Underbone. Thông tin hiển thị: product_id, product_name, category_name

$\pi_{product_id, product_name, category_name}(\sigma_{c.category_id = 8}((\text{product p} \bowtie_{(p.product_id=c.product_id)} \text{categories c}))$

h. Gom nhóm

- Ví dụ 1: Tìm số lượng đơn hàng được thực hiện mỗi tháng. Lấy các thuộc tính months, bill_amount

$\pi_{MONTH(order_date) \text{ AS } months, (months \mathrel{\mathcal{F}} COUNT(*)) \text{ AS } bill_amount}(\text{orders})$

- Ví dụ 2: Tính tổng tiền mỗi hóa đơn. Lấy các thuộc tính order_id, total

$\pi_{order_id, (order_id \mathrel{\mathcal{F}} SUM(price * quantity)) \text{ AS } total}(\sigma_{p.product_id = o.product_id}(\text{order_items AS o X products AS p}))$

- Ví dụ 3: Tìm số lượng mỗi sản phẩm được bán trong tháng 4. Lấy các thuộc tính product_id, product_name, sum_quantity

$\pi_{oi.product_id, p.product_name, (oi.product_id, p.product_name \mathrel{\mathcal{F}} SUM(quantity)) \text{ AS } sum_quantity}(\sigma_{oi.product_id = p.product_id \wedge oi.order_id = o.order_id \wedge MONTH(order_date) = 4}(\text{order_items AS oi X products AS p X orders AS o}))$

- Ví dụ 4: Đếm số lượng nhân viên làm ở mỗi cửa hàng. Lấy các thuộc tính store_id, store_name, amount

$$\pi_{so.store_id, store_name, (so.store_id, store_name \Join COUNT(*)) AS amount}(\sigma_{sa.store_id = so.store_id}(stores AS so X staffs AS sa))$$

i. Gom nhóm có điều kiện

- Ví dụ 1: Chọn ra các cửa hàng có số lượng nhân viên lớn hơn 3. Thông tin hiển thị store_id

$$K1 \leftarrow store_id \Join COUNT(*) (staffs)$$

$$KQ \leftarrow \pi_{store_id}(\sigma_{COUNT(*) > 3}(K1))$$

- Ví dụ 2: Chọn ra kho chứa hơn 50 xe. Thông tin hiển thị stock_id, total

$$K1 \leftarrow stock_id \Join SUM(inventory) (products)$$

$$KQ \leftarrow \pi_{stock_id, SUM(inventory) AS total}(\sigma_{SUM(inventory) > 50}(K1))$$

- Ví dụ 3: Chọn ra những đơn hàng được bán ra có tổng giá trị lớn hơn 800 triệu VNĐ. Thông tin hiển thị order_id, total

$$K1 \leftarrow order_id \Join SUM(quantity * price) AS total (order_items AS oi \Join_{(oi.product_id = p.product_id)} products AS p)$$

$$KQ \leftarrow \pi_{oi.order_id, SUM(quantity * price) AS total}(\sigma_{SUM(quantity * price) > 800000000}(K1))$$

- Ví dụ 4: Tìm những sản phẩm được đặt có số lượng lớn hơn 3. Thông tin hiển thị gồm product_id, product_name, sum_quantity

$$K1 \leftarrow p.product_id, product_name \Join SUM(quantity) (order_items oi \Join_{(oi.product_id = p.product_id)} products p)$$

$$KQ \leftarrow \pi_{p.product_id, product_name, SUM(quantity) AS sum_quantity}(\sigma_{SUM(quantity) > 3}(K1))$$

6. Lệnh mô tả dữ liệu: DDL

a. Tạo database

```
CREATE DATABASE qlyxemay_nhat_nhan;
```

b. Tạo các table dựa trên các quan hệ đã cho trên database

```
CREATE TABLE brands
```

```
(
```

```
        brand_id int PRIMARY KEY,  
        brand_name varchar(100)  
    )
```

CREATE TABLE categories

```
(  
    category_id int PRIMARY KEY,  
    category_name varchar(100)  
)
```

CREATE TABLE stocks

```
(  
    stock_id int PRIMARY KEY,  
    vacant char(2) CHECK(vacant IN ('Y','N'))  
)
```

CREATE TABLE products

```
(  
    product_id int PRIMARY KEY,  
    product_name varchar(100),  
    brand_id int FOREIGN KEY (brand_id) REFERENCES brands(brand_id),  
    category_id int FOREIGN KEY (category_id) REFERENCES categories(category_id),  
    stock_id int FOREIGN KEY (stock_id) REFERENCES stocks(stock_id),  
    model_year int,  
    price decimal,  
    inventory int  
)
```

CREATE TABLE customers

```
(  
    customer_id int PRIMARY KEY,  
    customer_name nvarchar(100),  
    phone char(10),  
    email varchar(100)  
)
```

CREATE TABLE stores

```
(
```

```

        store_id int PRIMARY KEY,
        store_name nvarchar(100),
        phone char(10),
        email varchar(100),
        store_address nvarchar(100),
        manager_id int
    )

```

CREATE TABLE staffs

```

(
    staff_id int PRIMARY KEY,
    staff_name nvarchar(100),
    store_id int FOREIGN KEY (store_id) REFERENCES stores(store_id),
    birthday datetime DEFAULT(GETDATE()),
    gender nchar(3) CHECK(gender IN (N'Nam',N'Nũ')),
    phone char(10),
    email varchar(100),
    salary int,
    manager_id int
)

```

CREATE TABLE orders

```

(
    order_id int PRIMARY KEY,
    order_date datetime DEFAULT(GETDATE()),
    customer_id int FOREIGN KEY (customer_id) REFERENCES customers(customer_id),
    staff_id int FOREIGN KEY (staff_id) REFERENCES staffs(staff_id)
)

```

CREATE TABLE order_items

```

(
    order_id int FOREIGN KEY (order_id) REFERENCES orders(order_id),
    item_id int,
    product_id int FOREIGN KEY (product_id) REFERENCES products(product_id),
    quantity int
    PRIMARY KEY (order_id, product_id)
)

```

```
CREATE TABLE relatives
(
    staff_id int FOREIGN KEY (staff_id) REFERENCES staffs(staff_id),
    relative_name nvarchar(100),
    relationship nvarchar(100)
    PRIMARY KEY (staff_id, relative_name)
)
```

c. Thay đổi cấu trúc bảng nếu có

```
ALTER TABLE stores
ADD CONSTRAINT fk_stores_staff FOREIGN KEY (manager_id) REFERENCES staffs(staff_id)
ALTER TABLE staffs
ADD CONSTRAINT fk_staffs_staffs FOREIGN KEY (manager_id) REFERENCES staffs(staff_id)
```

7. Lệnh thao tác dữ liệu: DDL

a. Thêm dữ liệu vào các table bằng lệnh SQL

- Thêm dữ liệu vào bảng categories:

```
INSERT INTO categories(category_id, category_name)
VALUES ('1','Adventure')
INSERT INTO categories(category_id, category_name)
VALUES ('2','Cruiser')
INSERT INTO categories(category_id, category_name)
VALUES ('3','Naked Bike')
INSERT INTO categories(category_id, category_name)
VALUES ('4','Scooter')
INSERT INTO categories(category_id, category_name)
VALUES ('5','Sport Bike')
INSERT INTO categories(category_id, category_name)
VALUES ('6','Sport Touring')
INSERT INTO categories(category_id, category_name)
VALUES ('7','Touring')
INSERT INTO categories(category_id, category_name)
VALUES ('8','Underbone')
```

- Thêm dữ liệu vào bảng brands:

```
INSERT INTO brands(brand_id, brand_name)
VALUES ('1','BMW')
INSERT INTO brands(brand_id, brand_name)
VALUES ('2','Ducati')
INSERT INTO brands(brand_id, brand_name)
```



```
VALUES ('3','Harley - Davidson')
INSERT INTO brands(brand_id, brand_name)
VALUES ('4','Honda')
INSERT INTO brands(brand_id, brand_name)
VALUES ('5','Kawasaki')
INSERT INTO brands(brand_id, brand_name)
VALUES ('6','Piaggio')
INSERT INTO brands(brand_id, brand_name)
VALUES ('7','Suzuki')
INSERT INTO brands(brand_id, brand_name)
VALUES ('8','Yamaha')
```

- Thêm dữ liệu vào bảng stocks:

```
INSERT INTO stocks(stock_id, vacant)
VALUES ('1','Y')
INSERT INTO stocks(stock_id, vacant)
VALUES ('2','Y')
INSERT INTO stocks(stock_id, vacant)
VALUES ('3','N')
```

- Thêm dữ liệu vào bảng products:

```
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('1', 'Ducati Multistrada 950 S', '2','1','1','2021','558000000','6')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('2', 'BMW R1200 GS Adventure', '1','1','1','2017','719000000','4')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('3', 'Harley-Davidson Sportster SuperLow', '3','2','2','2020','356000000','8')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('4', 'Kawasaki Vulcan S', '5','2','2','2021','241000000','11')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('5', 'Suzuki GSX-S1000', '7','3','2','2022','399000000','7')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('6', 'Vespa GTS 150', '6','4','2','2022','115000000','15')
INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price, inventory)
VALUES ('7', 'Yamaha NVX', '8','4','3','2020','52000000','19')
```

```

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('8', 'Air Blade 125/160', '4','4','3','2022','42000000','24')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('9', 'Suzuki Hayabusa GSX 1300R', '7','5','1','2021','750000000','5')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('10', 'BMW S1000RR', '1','5','1','2020','949000000','2')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('11', 'Kawasaki ZX-10R', '5','5','1','2022','729000000','3')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('12', 'Harley-Davidson Road King', '3','7','1','2021','965000000','2')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('13', 'Honda GoldWing', '4','7','1','2021','1230000000','1')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('14', 'BMW K-1600 GTL', '1','6','1','2022','634000000','7')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('15', 'Kawasaki Concours', '5','6','2','2020','384000000','10')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('16', 'Yamaha Z125', '8','8','2','2017','270000000','9')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('17', 'Honda Winner X', '4','8','3','2022','50060000','20')

INSERT INTO products(product_id, product_name, brand_id, category_id, stock_id, model_year, price,
inventory)
VALUES ('18', 'Suzuki Raider 150 FT', '7','8','3','2017','49190000','14')

```

- Thêm dữ liệu vào bảng customers:

```

INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('1', 'N'Nguyễn Thanh Vân', '0912151114','vannguyen@gmail.com')

INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('2', 'N'Nguyễn Văn Tạo', '0954544154','taole@gmail.com')

INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('3', 'N'Đỗ Duy Trung', '0931184518','trungdo@gmail.com')

INSERT INTO customers(customer_id, customer_name, phone, email)

```

```
VALUES ('4', N'Lê Minh Nguyệt', '0715166518', 'nguyetle@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('5', N'Nguyễn Kim Liên', '0961484123', 'liennguyen@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('6', N'Ngô Văn Sở', '0716151656', 'songo@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('7', N'Đinh Công Minh', '0921612161', 'minhding@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('8', N'Đặng Văn Tài', '0914653165', 'taidang@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('9', N'Trần Hạo Sơn', '0765164516', 'sonhao@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('10', N'Nguyễn Công Nghĩa', '0961646126', 'nghianguyen@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('11', N'Đỗ Tấn Tài', '0764151545', 'taido@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('12', N'Phạm Hoàng Long', '0961646546', 'longpham@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('13', N'Đặng Việt Trung', '0713216489', 'trungdang@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('14', N'Lê Công Minh', '0961618456', 'minhle@gmail.com')
INSERT INTO customers(customer_id, customer_name, phone, email)
VALUES ('15', N'Trần Hoài Sơn', '0754646465', 'sontran@gmail.com')
```

- Thêm dữ liệu vào bảng stores:

```
INSERT INTO stores(store_id, store_name, phone, email, store_address)
VALUES ('1', N'Dũng Sơn', '18006420', 'info@dungson.com', N'213 Tôn Đản, Phường 14, Quận 4, TP.HCM')
INSERT INTO stores(store_id, store_name, phone, email, store_address)
VALUES ('2', N'Phát Tài', '18004325', 'info@phattai.com', N'123 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM')
INSERT INTO stores(store_id, store_name, phone, email, store_address)
VALUES ('3', N'An Khang', '18007846', 'info@ankhang.com', N'442 Phạm Ngũ Lão, Phường Bến Nghé, Quận 1, TP.HCM')
```

- Thêm dữ liệu vào bảng staffs:

```
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('1', N'Đặng Công Thành', '1', '1/1/1998', N'Nam', '0324546464', 'thanhdang@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('2', N'Đỗ Phương Nghi', '2', '12/21/1992', N'Nữ', '0334446485', 'nghido@gmail.com', '15000000')
```

```

INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('3', N'Đỗ Thị Thúy Vy', '3', '4/3/1994', N'Nữ', '0346484838', 'vydo@gmail.com', '15000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('4', N'Nguyễn Hoàn Mỹ', '1', '7/3/1991', N'Nữ', '0355484354', 'mynguyen@gmail.com', '15000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('5', N'Nguyễn Ngọc Hằng', '2', '5/7/1999', N'Nữ', '0364464688', 'hangnguyen@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('6', N'Bùi Minh Đạt', '3', '8/19/2000', N'Nam', '0375164845', 'datbui@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('7', N'Võ Lê Khang', '1', '3/24/2001', N'Nam', '0381654861', 'khangvo@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('8', N'Đỗ Tuấn Cường', '2', '11/30/1990', N'Nam', '0395465462', 'cuongdo@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('9', N'Trần Hào Long', '3', '6/4/1997', N'Nam', '0354546548', 'longtran@gmail.com', '8000000')
INSERT INTO staffs(staff_id, staff_name, store_id, birthday, gender, phone, email, salary)
VALUES ('10', N'Phan Trung Tín', '2', '7/2/1998', N'Nam', '0345134846', 'tinphan@gmail.com', '8000000')

```

- Thêm dữ liệu vào bảng relatives:

```

INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('1', N'Đặng Việt Anh', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('2', N'Đỗ Phương Hoài', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('2', N'Lê Nhật Anh', N'Mẹ')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('3', N'Đỗ Trọng Nguyên', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('3', N'Phạm Thu Sương', N'Mẹ')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('4', N'Nguyễn Trung Hậu', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('4', N'Trần Huyền Trang', N'Mẹ')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('5', N'Lê Thị Mỹ Kim', N'Mẹ')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('6', N'Bùi Minh Trung', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('7', N'Võ Anh Tài', N'Cha')

```

```

INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('8', N'Nguyễn Thị Hồng', N'Mẹ')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('9', N'Trần Thanh Thiên', N'Cha')
INSERT INTO relatives(staff_id, relative_name, relationship)
VALUES ('10', N'Lê Mỹ Trúc', N'Mẹ')

```

- Thêm dữ liệu vào bảng orders:

```

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('1', '4/5/2022', '1', '1')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('2', '5/12/2022', '2', '4')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('3', '6/15/2022', '4', '7')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('4', '4/18/2022', '3', '1')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('5', '5/14/2022', '6', '4')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('6', '6/23/2022', '7', '4')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('7', '4/30/2022', '5', '4')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('8', '5/1/2022', '4', '7')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('9', '6/14/2022', '6', '2')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('10', '4/15/2022', '7', '5')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('11', '5/19/2022', '8', '8')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('12', '6/7/2022', '9', '10')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('13', '4/4/2022', '10', '2')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('14', '5/4/2022', '12', '2')
INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('15', '6/9/2022', '15', '5')

```

```

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('16', '4/30/2022', '13', '3')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('17', '5/31/2022', '14', '3')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('18', '6/22/2022', '8', '6')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('19', '4/3/2022', '14', '9')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('20', '5/7/2022', '11', '3')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('21', '6/1/2022', '8', '9')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('22', '4/9/2022', '13', '3')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('23', '5/18/2022', '9', '3')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('24', '6/27/2022', '11', '6')

INSERT INTO orders(order_id, order_date, customer_id, staff_id)
VALUES ('25', '4/29/2022', '4', '9')

```

- Thêm dữ liệu vào bảng orders_items:

```

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('1', '1', '5', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('1', '2', '8', '3')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('2', '1', '1', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('3', '1', '7', '2')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('4', '1', '9', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('4', '2', '18', '3')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('5', '1', '2', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('5', '2', '14', '1')

```

```

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('6', '1', '17', '2')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('7', '1', '13', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('8', '1', '11', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('9', '1', '12', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('9', '2', '6', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('10', '1', '3', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('11', '1', '4', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('12', '1', '10', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('12', '2', '15', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('12', '3', '16', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('13', '1', '7', '2')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('13', '2', '18', '2')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('14', '1', '9', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('15', '1', '4', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('16', '1', '2', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('17', '1', '4', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('17', '2', '3', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('18', '1', '6', '1')
INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('19', '1', '11', '1')

```

```

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('20', '1', '9', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('20', '2', '5', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('20', '3', '8', '2')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('21', '1', '1', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('22', '1', '16', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('23', '1', '17', '2')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('24', '1', '6', '1')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('25', '1', '8', '4')

INSERT INTO order_items(order_id, item_id, product_id, quantity)
VALUES ('25', '2', '18', '3')

```

b. Update dữ liệu vào các table bằng lệnh SQL

```

UPDATE stores
SET manager_id = '4'
WHERE store_id = '1'

```

```

UPDATE stores
SET manager_id = '3'
WHERE store_id = '3'

```

```

UPDATE stores
SET manager_id = '2'
WHERE store_id = '2'

```

```

UPDATE staffs
SET manager_id = '2'
WHERE staff_id IN ('5', '8', '10')

```

```

UPDATE staffs
SET manager_id = '3'

```



```
WHERE staff_id IN ('6','9')
```

```
UPDATE staffs
```

```
SET manager_id = '4'
```

```
WHERE staff_id IN ('1', '7')
```

c. Xóa dữ liệu ở các bảng table bằng lệnh SQL

```
DELETE FROM products
```

```
WHERE product_id = '19'
```

8. Lệnh truy vấn dữ liệu: SQL

a. Truy vấn 1 bảng

--Truy vấn 1 bảng

--1) Lọc ra những nhân viên có tuổi nhỏ hơn 25. Thông tin hiển thị: staff_id, staff_name, Age

```
SELECT
```

```
    staff_id,
```

```
    staff_name,
```

```
    YEAR(GETDATE())-YEAR(birthday) Age
```

```
FROM staffs
```

```
WHERE YEAR(GETDATE())-YEAR(birthday) <25
```

--2) Lọc ra sản phẩm có trong kho 1 Thông tin hiển thị: product_id, product_name

```
SELECT
```

```
    product_id,
```

```
    product_name
```

```
FROM products
```

```
WHERE stock_id=1
```

--3) Lọc ra những đơn hàng được hoàn thành trong tháng 5 năm nay. Thông tin hiển thị order_id, order_date, customer_id, staff_id

```
SELECT *
```

```
FROM orders
```

```
WHERE MONTH(order_date)=5
```

--4) Tìm ra những thân nhân có quan hệ là cha. Thông tin hiển thị *

```
SELECT *
```

```
FROM relatives
```

```
WHERE relationship=N'cha'
```

Kết quả các bảng

	staff_id	staff_name	Age
1	1	Đặng Công Thành	24
2	5	Nguyễn Ngọc Hằng	23
3	6	Bùi Minh Đạt	22
4	7	Võ Lê Khang	21
5	10	Phan Trung Tín	24

	product_id	product_name
1	1	Ducati Multistrada 950 S
2	2	BMW R1200 GS Adventure
3	9	Suzuki Hayabusa GSX 1300R
4	10	BMW S1000RR
5	11	Kawasaki ZX-10R
6	12	Harley-Davidson Road King
7	13	Honda GoldWing
8	14	BMW K-1600 GTL

	order_id	order_date	customer_id	staff_id
1	2	2022-05-12 00:00:00.000	2	4
2	5	2022-05-14 00:00:00.000	6	4
3	8	2022-05-01 00:00:00.000	4	7
4	11	2022-05-19 00:00:00.000	8	8
5	14	2022-05-04 00:00:00.000	12	2
6	17	2022-05-31 00:00:00.000	14	3
7	20	2022-05-07 00:00:00.000	11	3
8	23	2022-05-18 00:00:00.000	9	3

	staff_id	relative_name	relationship
1	1	Đặng Việt Anh	Cha
2	2	Đỗ Phương Hoài	Cha
3	3	Đỗ Trang Nguyễn	Cha
4	4	Nguyễn Trung Hậu	Cha
5	6	Bùi Minh Trung	Cha
6	7	Võ Anh Tài	Cha
7	9	Trần Thanh Thiện	Cha

b. Truy vấn nhiều bảng (Phép kết)

-- Truy vấn nhiều bảng

-- Đếm số lượng nhân viên làm ở mỗi cửa hàng. Thông tin gồm: store_id, store_name, amount

SELECT

so.store_id,

store_name,

COUNT(*) amount

FROM stores so

JOIN staffs sa

ON sa.store_id = so.store_id

GROUP BY so.store_id, store_name

ORDER BY Amount,store_name;

-- Tính tổng thành tiền của mỗi hóa đơn. Thông tin gồm: order_id, total

SELECT

order_id,

SUM(quantity*price) total

FROM order_items o

JOIN products p

ON o.product_id = p.product_id

GROUP BY order_id

ORDER BY order_id;

-- Tìm số hóa đơn mỗi cửa hàng đã thực hiện trong tháng 4. Thông tin gồm: store_id, store_name, total_bill

SELECT

sta.store_id,

store_name,

COUNT(*) total_bill

FROM orders o

JOIN staffs sta ON o.staff_id = sta.staff_id

JOIN stores sto ON sta.store_id = sto.store_id

WHERE MONTH(order_date) = 4

GROUP BY sta.store_id, store_name;

-- Tính tổng doanh thu mà mỗi nhân viên đã bán được tháng 4. Thông tin gồm: staff_id, staff_name, total_per_staff

WITH total AS

(

SELECT

order_id,

SUM(quantity*price) total

FROM order_items o

JOIN products p

ON o.product_id = p.product_id

GROUP BY order_id

),

sale AS

(

SELECT

o.staff_id,

s.staff_name,

SUM(total) total_per_staff

FROM orders o

JOIN total t ON o.order_id = t.order_id

JOIN staffs s ON o.staff_id = s.staff_id

WHERE MONTH(order_date) = 4

GROUP BY o.staff_id, s.staff_name

),

non_sale AS

```

(
    SELECT
        staff_id,
        staff_name,
        0 total_per_staff
    FROM staffs
    WHERE staff_id NOT IN
        (
            SELECT
                DISTINCT o.staff_id
            FROM orders o
            JOIN staffs s
            ON o.staff_id = s.staff_id
            WHERE MONTH(order_date) = 4
        )
)

SELECT *
FROM non_sale
    UNION
SELECT *
FROM sale
ORDER BY staff_id;

```

Kết quả các bảng

	store_id	store_name	amount
1	3	An Khang	3
2	1	Dũng Sơn	3
3	2	Phát Tài	4

	order_id	total
18	18	115000000
19	19	729000000
20	20	1233000000
21	21	558000000
22	22	270000000
23	23	100120000
24	24	115000000
25	25	315570000

	store_id	store_name	total_bill
1	1	Dũng Sơn	3
2	2	Phát Tài	2
3	3	An Khang	4

	staff_id	staff_name	total_per_staff
1	1	Đặng Công Thành	1422570000
2	2	Đỗ Phương Nghi	202380000
3	3	Đỗ Thị Thủy Vy	989000000
4	4	Nguyễn Hoàn Mỹ	1230000000
5	5	Nguyễn Ngọc H...	356000000
6	6	Bùi Minh Đạt	0
7	7	Võ Lê Khang	0
8	8	Đỗ Tuấn Cường	0
9	9	Trần Hạo Long	1044570000
10	10	Phan Trung Tín	0

c. Truy vấn có điều kiện (and, or, like, between, ...)

--Truy vấn có điều kiện (and, or, like, between)

--1) Tìm ra những nhân viên ở cửa hàng Dũng Sơn và có tuổi lớn hơn 25. Thông tin hiển thị staff_id, staff_name, store_name, Age

SELECT

sta.staff_id,

sta.staff_name,

s.store_name,

YEAR(GETDATE())-YEAR(birthday) Age

FROM staffs sta JOIN stores s On sta.store_id=s.store_id

WHERE S.store_id=1 and YEAR(GETDATE())-YEAR(birthday) >25

--2) Tìm ra những đơn hàng được mua trong nửa đầu tháng 4 năm nay. Thông tin hiển thị: order_id, order_date, store_name

SELECT

o.order_id,

o.order_date,

s.store_name

FROM orders o

JOIN staffs sta ON o.staff_id=sta.staff_id

JOIN stores s ON sta.store_id=s.store_id

WHERE (DAY(o.order_date) BETWEEN 1 and 15) and MONTH(o.order_date) =4

--3) Lọc ra những khách hàng mua xe có phân khúc Adventure hoặc Scooter. Thông tin hiển thị: customer_id, customer_name, category_name, product_name

SELECT DISTINCT

c.customer_id,
c.customer_name,
cat.category_name,
p.product_name

FROM customers c

JOIN orders o **ON** c.customer_id=o.customer_id

JOIN order_items oi **ON** o.order_id=oi.order_id

JOIN products p **ON** oi.product_id=p.product_id

JOIN categories cat **ON** p.category_id=cat.category_id

WHERE p.category_id=1 or p.category_id=4

--4) Tìm ra những nhân viên có tuổi trong khoảng từ 30-40 và đã bán được đơn hàng trong tháng 6. Thông tin hiển thị staff_id, staff_name, salary

SELECT

s.staff_id,
s.staff_name,
salary

FROM staffs s

JOIN orders o **ON** s.staff_id=o.staff_id

WHERE (YEAR(GETDATE())-YEAR(s.birthday) between 30 and 40) and MONTH(o.order_date) =6

Kết quả các bảng

	staff_id	staff_name	store_name	Age
1	4	Nguyễn Hoàn Mỹ	Dũng Sơn	31

	order_id	order_date	store_name
1	1	2022-04-05 00:00:00.000	Dũng Sơn
2	10	2022-04-15 00:00:00.000	Phát Tài
3	13	2022-04-04 00:00:00.000	Phát Tài
4	19	2022-04-03 00:00:00.000	An Khang
5	22	2022-04-09 00:00:00.000	An Khang

	customer_id	customer_name	category_name	product_name
1	1	Nguyễn Thanh Văn	Scooter	Air Blade 125/160
2	2	Lê Văn Tạo	Adventure	Ducati Multistrada 950 S
3	4	Lê Minh Nguyệt	Scooter	Air Blade 125/160
4	4	Lê Minh Nguyệt	Scooter	Yamaha NVX
5	6	Ngô Văn Sô	Adventure	BMW R1200 GS Adventure
6	6	Ngô Văn Sô	Scooter	Vespa GTS 150
7	8	Đặng Văn Tài	Adventure	Ducati Multistrada 950 S
8	8	Đặng Văn Tài	Scooter	Vespa GTS 150
9	10	Nguyễn Công Nghĩa	Scooter	Yamaha NVX
10	11	Đỗ Tấn Tài	Scooter	Air Blade 125/160
11	11	Đỗ Tấn Tài	Scooter	Vespa GTS 150
12	13	Đặng Việt Trung	Adventure	BMW R1200 GS Adventure

	staff_id	staff_name	salary
1	4	Nguyễn Hoàn Mỹ	15000000
2	2	Đỗ Phương Nghi	15000000

d. Truy vấn tính toán

-- Truy vấn tính toán

-- Tính giá trị trung bình các sản phẩm ở từng kho. Thông tin gồm stock_id, avg_value

SELECT

stock_id,

CAST(AVG(price * inventory) AS DECIMAL(18,0)) avg_value

FROM products

GROUP BY stock_id;

-- Sản phẩm đắt nhất mỗi loại xe mà cửa hàng sở hữu. Thông tin gồm category_id, category_name, product_name, price

WITH max_price AS

(

SELECT

c.category_id,

c.category_name,

MAX(price) price

FROM products p

JOIN categories c

ON p.category_id = c.category_id

GROUP BY c.category_id, c.category_name

)

SELECT

m.category_id,

```

        category_name,
        product_name,
        m.price
FROM max_price m
JOIN products p
ON p.price = m.price
ORDER BY m.category_id;

```

-- Tốc độ tăng trưởng doanh thu tháng 5 so với tháng 4.

```

WITH total AS
(
    SELECT
        order_id,
        SUM(quantity*price) total
    FROM order_items o
    JOIN products p
    ON o.product_id = p.product_id
    GROUP BY order_id
),
total_per_month AS
(
    SELECT
        MONTH(order_date) months,
        SUM(total) total_per_month
    FROM orders o
    JOIN total t
    ON o.order_id = t.order_id
    GROUP BY MONTH(order_date)
)

```

```

SELECT
    ((SELECT total_per_month FROM total_per_month WHERE months = 5)
    - (SELECT total_per_month FROM total_per_month WHERE months = 4))
    / (SELECT total_per_month FROM total_per_month WHERE months = 4) rate;

```

/* Tính lương của mỗi nhân viên trong tháng 4 biết rằng với mỗi đơn hàng được thực hiện họ sẽ được trích 1% đơn hàng

Thông tin gồm staff_id, staff_name, income */

```

WITH total AS
(

```



```

SELECT
    order_id,
    SUM(quantity*price) total
FROM order_items o
JOIN products p
ON o.product_id = p.product_id
GROUP BY order_id
),
sale AS
(
    SELECT
        o.staff_id,
        s.staff_name,
        salary,
        SUM(total) total_per_staff
    FROM orders o
    JOIN total t ON o.order_id = t.order_id
    JOIN staffs s ON o.staff_id = s.staff_id
    WHERE MONTH(order_date) = 4
    GROUP BY o.staff_id, s.staff_name, salary
),
sale_income AS
(
    SELECT
        staff_id,
        staff_name,
        CONVERT(INT, (salary + 1.0/100 * total_per_staff)) income
    FROM sale
),
non_sale_income AS
(
    SELECT
        staff_id,
        staff_name,
        salary income
    FROM staffs
    WHERE staff_id NOT IN
        (

```

```

SELECT
    DISTINCT o.staff_id
FROM orders o
JOIN staffs s
ON o.staff_id = s.staff_id
WHERE MONTH(order_date) = 4
)

SELECT *
FROM non_sale_income

UNION

SELECT *
FROM sale_income
ORDER BY staff_id;

```

Kết quả các bảng

	stock_id	avg_value
1	1	2707125000
2	2	2714500000
3	3	921465000

	category_id	category_name	product_name	price
1	1	Adventure	BMW R1200 GS Adventure	719000000
2	2	Cruiser	Harley-Davidson Sportster SuperLow	356000000
3	3	Naked Bike	Suzuki GSX-S1000	399000000
4	4	Scooter	Vespa GTS 150	115000000
5	5	Sport Bike	BMW S1000RR	949000000
6	6	Sport Touring	BMW K-1600 GTL	634000000
7	7	Touring	Honda GoldWing	1230000000
8	8	Underbone	Yamaha Z125	270000000

	rate
1	0.060367

	staff_id	staff_name	income
1	1	Đặng Công Thành	22225700
2	2	Đỗ Phương Nghi	17023800
3	3	Đỗ Thị Thúy Vy	24890000
4	4	Nguyễn Hoàn Mỹ	27300000
5	5	Nguyễn Ngọc Hằng	11560000
6	6	Bùi Minh Đạt	8000000
7	7	Võ Lê Khang	8000000
8	8	Đỗ Tuấn Cường	8000000
9	9	Trần Hạo Long	18445700
10	10	Phan Trung Tín	8000000

e. Truy vấn có gom nhóm (group by)

--Truy vấn có gom nhóm(GROUP BY)

--1) Gom nhóm những khách hàng tiềm năng (mua trên 2 tỷ). Thông tin hiển thị: customer_id, customer_name, total_bill

WITH total AS

(

```

SELECT
    order_id,
    SUM(quantity*price) total
FROM order_items o
JOIN products p
ON o.product_id = p.product_id
GROUP BY order_id
)

```

```

SELECT
    c.customer_id,
    c.customer_name,
    SUM(t.total) total_bill
FROM customers c
JOIN orders o ON c.customer_id=o.customer_id
JOIN order_items oi ON o.order_id=oi.order_id
JOIN total t ON t.order_id=oi.order_id
GROUP BY c.customer_id, c.customer_name
HAVING SUM(t.total) > 2000000000

```

--2) Gom nhóm những nhân viên bán được từ 2 đơn hàng trở lên trong tháng 4. Thông tin hiển thị: staff_id, staff_name, total

```

SELECT
    s.staff_id,
    s.staff_name,
    COUNT(o.order_id) total
FROM staffs s
JOIN orders o ON s.staff_id=o.staff_id
WHERE MONTH(o.order_date) = 5
GROUP BY s.staff_id, s.staff_name
HAVING COUNT(o.order_id)>=2

```

--3) Tìm ra những nhân viên bán được doanh thu cao nhất mỗi cửa hàng. Thông tin hiển thị: staff_id, staff_name, store_id, sales, total_orders

```

WITH total AS
(
    SELECT
        order_id,
        SUM(quantity*price) total

```

```

        FROM order_items o
        JOIN products p
        ON o.product_id = p.product_id
        GROUP BY order_id
    )
SELECT
    sta.staff_id,
    sta.staff_name,
    s.store_id,
    MAX(t.total) sales,
    COUNT(o.order_id) total_orders
FROM stores s
JOIN staffs sta ON s.store_id=sta.staff_id
JOIN orders o ON o.staff_id=sta.staff_id
JOIN total t ON t.order_id=o.order_id
GROUP BY s.store_id, sta.staff_id, sta.staff_name;

```

--4) Giả sử khách hàng mua đơn hàng có giá trị từ 1 tỷ VNĐ trở lên được tặng 1 phần quà trị giá bằng 1% đơn hàng đó, tính hoa hồng của mỗi khách hàng đó. Thông tin gồm: customer_id, customer_name, commission

```

WITH total AS
(
    SELECT
        order_id,
        SUM(quantity*price) total
    FROM order_items o
    JOIN products p
    ON o.product_id = p.product_id
    GROUP BY order_id
    HAVING SUM(quantity*price)>1000000000
)
SELECT
    c.customer_id,
    c.customer_name,
    FORMAT(((SUM(t.total)/100), '#.0') commission
FROM customers c
JOIN orders o ON c.customer_id=o.customer_id
JOIN total t ON t.order_id=o.order_id
GROUP BY c.customer_id, c.customer_name

```

Kết quả các bảng

	customer_id	customer_name	total_bill
1	6	Ngô Văn Sở	4866000000
2	9	Trần Hạo Sơn	4909120000
3	11	Đỗ Tấn Tài	3814000000

	staff_id	staff_name	total
1	3	Đỗ Thị Thúy Vy	3
2	4	Nguyễn Hoàn Mỹ	2

	staff_id	staff_name	store_id	sales	total_orders
1	1	Đặng Công Thành	1	897570000	2
2	2	Đỗ Phương Nghi	2	1080000000	3
3	3	Đỗ Thị Thúy Vy	3	1233000000	5

	customer_id	customer_name	commission
1	5	Nguyễn Kim Liên	12300000.0
2	6	Ngô Văn Sở	24330000.0
3	9	Trần Hạo Sơn	16030000.0
4	11	Đỗ Tấn Tài	12330000.0

f. Truy vấn gom nhóm có điều kiện (having)

-- Truy vấn gom nhóm có điều kiện

-- Tìm những sản phẩm được đặt với số lượng lớn hơn 3 trong tháng 4. Thông tin gồm product_id, product_name, sum_quantity

SELECT

oi.product_id,

p.product_name,

SUM(quantity) sum_quantity

FROM order_items oi

JOIN orders o ON oi.order_id = o.order_id

JOIN products p ON oi.product_id = p.product_id

WHERE MONTH(order_date) = 4

GROUP BY oi.product_id, p.product_name

HAVING SUM(quantity) > 3

ORDER BY sum_quantity DESC;

--Tìm những hãng xe được mua với số lượng lớn hơn 5 trong tháng 4. Thông tin gồm brand_id, brand_name, sum_quantity

WITH sum_quantity AS

(

SELECT

oi.product_id,

p.product_name,

```

        brand_id,
        SUM(quantity) sum_quantity
    FROM order_items oi
    JOIN orders o ON oi.order_id = o.order_id
    JOIN products p ON oi.product_id = p.product_id
    WHERE MONTH(order_date) = 4
    GROUP BY oi.product_id, p.product_name, brand_id
)
SELECT
    s.brand_id,
    b.brand_name,
    SUM(s.sum_quantity) sum_quantity
FROM sum_quantity s
JOIN brands b ON s.brand_id = b.brand_id
GROUP BY s.brand_id, b.brand_name
HAVING SUM(s.sum_quantity) > 5
ORDER BY SUM(s.sum_quantity) DESC;

```

-- Tìm những hóa đơn có giá trị lớn hơn 500 triệu. Thông tin gồm order_id, total

```

SELECT
    order_id,
    SUM(quantity*price) total
FROM order_items o
JOIN products p
ON o.product_id = p.product_id
GROUP BY order_id
HAVING SUM(quantity*price) > 500000000
ORDER BY total DESC;

```

/*Tìm nhân viên đạt được mức hoa hồng lớn hơn 10 triệu trong tháng 4. Biết mỗi nhân viên có thể trích 1% từ hóa đơn bán được

Thông tin gồm staff_id, staff_name, tip*/

```

WITH total AS
(
    SELECT
        order_id,
        SUM(quantity*price) total
    FROM order_items o
    JOIN products p
    ON o.product_id = p.product_id
    GROUP BY order_id
)

```

```

)
SELECT
    o.staff_id,
    s.staff_name,
    CONVERT(INT, (1.0/100 * SUM(total))) tip
FROM orders o
JOIN total t ON o.order_id = t.order_id
JOIN staffs s ON o.staff_id = s.staff_id
WHERE MONTH(order_date) = 4
GROUP BY o.staff_id, s.staff_name, salary
HAVING CONVERT(INT, (1.0/100 * SUM(total))) > 10000000;

```

Kết quả các bảng

	product_id	product_name	sum_quantity
1	18	Suzuki Raider 150 FI	8
2	8	Air Blade 125/160	7

	brand_id	brand_name	sum_quantity
1	7	Suzuki	10
2	4	Honda	8

	order_id	total
1	12	1603000000
2	5	1353000000
3	20	1233000000
4	7	1230000000
5	9	1080000000
6	4	897570000
7	14	750000000
8	8	729000000

	staff_id	staff_name	tip
1	1	Đặng Công Thành	14225700
2	4	Nguyễn Hoàn Mỹ	12300000
3	9	Trần Hạo Long	10445700

g. Truy vấn có sử dụng phép giao, hội, trừ

--Truy vấn có sử dụng phép giao hội trừ

--1)Tìm ra những nhân viên ở cửa hàng 1 hoặc 3. Thông tin hiển thị *

```

SELECT *
FROM staffs s
WHERE s.store_id=1

```

UNION

```

SELECT *
FROM staffs s
WHERE s.store_id=3

```

--2)Tìm ra những nhân viên bán được nhiều hơn 1 đơn hàng trong tháng 4

```

SELECT
    sta.staff_id,
    sta.staff_name,
    COUNT(o.order_id) total_per_staff
FROM staffs sta
JOIN orders o ON sta.staff_id=o.staff_id
WHERE MONTH(o.order_date)=4
GROUP BY sta.staff_id, sta.staff_name
EXCEPT
SELECT

```

```

    sta.staff_id,
    sta.staff_name,
    COUNT(o.order_id) total_per_staff
FROM staffs sta
JOIN orders o ON sta.staff_id=o.staff_id
WHERE MONTH(o.order_date)=4
GROUP BY sta.staff_id, sta.staff_name
HAVING COUNT(o.order_id)<2

```

--3) Tìm ra những đơn hàng có sản phẩm Harley-Davidson Sportster SuperLow. Thông tin hiển thị order_id, product_id, product_name

```

SELECT
    oi.order_id,
    p.product_id,
    product_name
FROM products p
JOIN order_items oi ON p.product_id=oi.product_id
INTERSECT
SELECT
    oi.order_id,
    p.product_id,
    product_name
FROM products p
JOIN order_items oi ON p.product_id=oi.product_id
WHERE p.product_id = 3

```

--4) Tìm ra những xe ở kho 1 hoặc kho 2. Thông tin hiển thị product_id, product_name

```

SELECT
    product_id,
    product_name

```



```

FROM products
WHERE stock_id=1
UNION
SELECT
    product_id,
    product_name
FROM products
WHERE stock_id=2

```

Kết quả các bảng

	staff_id	staff_name	store_id	birthday	phone	email	salary
1	1	Đặng Công Thành	1	1998-01-01 00:00:00.000	0324546464	thanh dang@gmail.com	8000000
2	4	Nguyễn Hoàn Mỹ	1	1991-07-03 00:00:00.000	0355484354	mynguyen@gmail.com	15000000
3	7	Võ Lê Khang	1	2001-03-24 00:00:00.000	0381654861	khangvo@gmail.com	8000000
4	3	Đỗ Thị Thủy Vy	3	1994-04-03 00:00:00.000	0346484838	vydo@gmail.com	15000000
5	6	Bùi Minh Đạt	3	2000-08-19 00:00:00.000	0375164845	datbui@gmail.com	8000000
6	9	Trần Hạo Long	3	1997-06-04 00:00:00.000	0354546548	longtran@gmail.com	8000000

	staff_id	staff_name	total_per_staff
1	1	Đặng Công Thành	2
2	3	Đỗ Thị Thủy Vy	2
3	9	Trần Hạo Long	2

	order_id	product_id	product_name
1	10	3	Harley-Davidson Sportster SuperLow
2	17	3	Harley-Davidson Sportster SuperLow

	product_id	product_name
1	1	Ducati Multistrada 950 S
2	2	BMW R1200 GS Adventure
3	3	Harley-Davidson Sportster SuperLow
4	4	Kawasaki Vulcan S
5	5	Suzuki GSX-S1000
6	6	Vespa GTS 150
7	9	Suzuki Hayabusa GSX 1300R
8	10	BMW S1000RR
9	11	Kawasaki ZX-10R
10	12	Harley-Davidson Road King
11	13	Honda GoldWing
12	14	BMW K-1600 GTL
13	15	Kawasaki Concours
14	16	Yamaha Z125

h. Truy vấn con

-- Truy vấn con

-- Tìm tổng hóa đơn do các nhân viên cấp quản lý thực hiện trong tháng 4. Thông tin gồm: manager_id, manager_name, total_bill

```

SELECT
    o.staff_id manager_id,
    s.staff_name manager_name,
    COUNT(*) total_bill
FROM orders o
JOIN staffs s
ON o.staff_id = s.staff_id
WHERE o.staff_id IN (SELECT manager_id FROM stores)
GROUP BY o.staff_id, s.staff_name;

```

-- Tìm nhân viên có số tuổi lớn nhất trong mỗi cơ sở. Thông tin gồm: store_id, store_name, staff_name, birthday, age

```
SELECT
    sa.store_id,
    store_name,
    staff_name,
    FORMAT(birthday, 'dd-MM-yyyy') birthday,
    YEAR(GETDATE()) - YEAR(birthday) age
FROM staffs sa
JOIN stores so
ON sa.store_id = so.store_id
WHERE YEAR(GETDATE()) - YEAR(birthday) IN
    (
        SELECT
            MAX(YEAR(GETDATE()) - YEAR(birthday))
        FROM staffs
        GROUP BY store_id
    )
ORDER BY store_id;
```

-- Tìm thông tin khách hàng đã thanh toán nhiều hóa đơn nhất trong tháng 4. Thông tin gồm: customer_id, customer_name, total_bill

```
SELECT
    o.customer_id,
    customer_name,
    COUNT(*) total_bill
FROM orders o
JOIN customers c
ON o.customer_id = c.customer_id
GROUP BY o.customer_id, customer_name
HAVING COUNT(*) IN
    (
        SELECT
            TOP 1 COUNT(*)
        FROM orders
        WHERE MONTH(order_date) = 4
        GROUP BY customer_id
        ORDER BY COUNT(*) DESC
    );
```

/* Biết rằng mỗi nhân viên cấp quản lý ngoài 1% từ hóa đơn còn có 10% hoa hồng của nhân viên dưới cấp.

Tính lương thực nhận của nhân viên cấp quản lý trong tháng 4. Thông tin gồm: store_name, manager_id, manager_name, real_income*/

WITH total AS

```
(
    SELECT
        order_id,
        SUM(quantity*price) total
    FROM order_items o
    JOIN products p
    ON o.product_id = p.product_id
    GROUP BY order_id
),
sale AS
(
    SELECT
        o.staff_id,
        s.staff_name,
        salary,
        store_id,
        SUM(total) total_per_staff
    FROM orders o
    JOIN total t ON o.order_id = t.order_id
    JOIN staffs s ON o.staff_id = s.staff_id
    WHERE MONTH(order_date) = 4
    GROUP BY o.staff_id, s.staff_name, salary, store_id
),
```

sale_income AS

```
(
    SELECT
        store_id,
        CONVERT(INT, SUM(salary + 1.0/100 * total_per_staff)) income
    FROM sale
    WHERE staff_id NOT IN (SELECT manager_id FROM stores)
    GROUP BY store_id
)
```

SELECT

```

store_name,
manager_id,
sta.staff_name manager_name,
CONVERT(INT, (sta.salary + 1.0/10 * income + 1.0/100 * total_per_staff)) real_income
FROM sale_income si
JOIN stores s ON si.store_id = s.store_id
JOIN staffs sta ON sta.staff_id = s.manager_id
JOIN sale sal ON sal.staff_id = s.manager_id;

```

Kết quả các bảng

	manager_id	manager_name	total_bill
1	2	Đỗ Phương Nghi	3
2	3	Đỗ Thị Thúy Vy	5
3	4	Nguyễn Hoàn Mỹ	4

	store_id	store_name	staff_name	birthday	age
1	1	Dũng Sơn	Nguyễn Hoàn Mỹ	03-07-1991	31
2	2	Phát Tài	Đỗ Tuấn Cường	30-11-1990	32
3	3	An Khang	Đỗ Thị Thúy Vy	03-04-1994	28

	customer_id	customer_name	total_bill
1	6	Ngô Văn Sò	2
2	7	Đinh Công Minh	2
3	9	Trần Hạo Sơn	2
4	11	Đỗ Tấn Tài	2
5	13	Đặng Việt Trung	2
6	14	Lê Công Minh	2

	store_name	manager_id	manager_name	real_income
1	Phát Tài	2	Đỗ Phương Nghi	18179800
2	An Khang	3	Đỗ Thị Thúy Vy	26734570
3	Dũng Sơn	4	Nguyễn Hoàn Mỹ	29522570

i. Truy vấn chéo

--Truy vấn chéo (PIVOT)

--lọc ra danh sách các loại xe trong mỗi đơn hàng

```
SELECT DISTINCT product_id, product_name, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]
```

FROM

(SELECT DISTINCT

p.product_id,

p.product_name,

oi.order_id,

```

        oi.item_id
FROM products p
JOIN order_items oi ON p.product_id=oi.product_id
) as bangnguồn
pivot
(
COUNT(item_id)
for order_id in ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20],
[21], [22], [23], [24], [25])
)
as bangchuyen

```

--Lọc ra danh sách số lượng các xe trong mỗi đơn hàng

```

SELECT DISTINCT product_id, product_name, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14],
[15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]
FROM
(SELECT DISTINCT
        p.product_id,
        p.product_name,
        oi.order_id,
        quantity
FROM products p
JOIN order_items oi ON p.product_id=oi.product_id
) as bangnguồn
pivot
(
SUM(quantity)
for order_id in ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20],
[21], [22], [23], [24], [25])
)
as bangchuyen

```

--Danh sách các nhân viên ứng với mỗi cửa hàng

```

SELECT DISTINCT staff_id, staff_name, [1], [2], [3]
FROM
(SELECT DISTINCT
        sta.staff_id,
        sta.staff_name,
        s.store_id,

```

```

s.store_name
FROM staffs sta
JOIN stores s ON sta.store_id=s.store_id
) as bangnguồn
pivot
(
COUNT(store_name)
for store_id in ([1], [2], [3])
)
as bangchuyen

```

--Số lượng các hoá đơn được nhân viên bán ra theo từng tháng

```

SELECT DISTINCT staff_id, staff_name, [4], [5], [6]
FROM
(SELECT
s.staff_id,
s.staff_name,
o.order_id,
MONTH(order_date) month
FROM staffs s
JOIN orders o ON o.staff_id=s.staff_id
) as bangnguồn
pivot
(
COUNT(order_id)
FOR month IN ([4], [5], [6])) as bangchuyen

```

Kết quả các bảng

	product_id	product_name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	Ducati Multistrada 950 S	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	2	BMW R1200 GS Adventure	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	3	Harley-Davidson Sportster SuperLow	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
4	4	Kawasaki Vulcan S	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
5	5	Suzuki GSX-S1000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
6	6	Vespa GTS 150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0
7	7	Yamaha NXX	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	8	Air Blade 125/160	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
9	9	Suzuki Hayabusa GSX 1300R	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
10	10	BMW S1000RR	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
11	11	Kawasaki ZX-10R	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
12	12	Harley-Davidson Road King	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	13	Honda GoldWing	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	14	BMW K-1600 GTL	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	15	Kawasaki Concours	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
16	16	Yamaha Z125	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
17	17	Honda Winner X	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
18	18	Suzuki Raider 150 FI	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

product_id	product_name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	Ducati Multistrada 950 S	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL
2	BMW R1200 GS Adventure	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	Harley-Davidson Sportster SuperLow	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	Kawasaki Vulcan S	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	1	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	Suzuki GSX-S1000	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL
6	Vespa GTS 150	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL
7	Yamaha NIX	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	Air Blade 125/160	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	4	NULL	NULL
9	Suzuki Hayabusa GSX 1300R	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	BMW S1000RR	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
11	Kawasaki ZX-10R	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
12	Harley-Davidson Road King	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
13	Honda GoldWing	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
14	BMW K-1600 GTL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
15	Kawasaki Concours	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
16	Yamaha Z125	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	1	NULL	NULL	NULL	NULL
17	Honda Winner X	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL
18	Suzuki Raider 150 FI	NULL	NULL	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	3	NULL

staff_id	staff_name	1	2	3
1	Đặng Công Thành	1	0	0
2	Đỗ Phương Nghi	0	1	0
3	Đỗ Thị Thúy Vy	0	0	1
4	Nguyễn Hoàn Mỹ	1	0	0
5	Nguyễn Ngọc Hằng	0	1	0
6	Bùi Minh Đạt	0	0	1
7	Võ Lê Khang	1	0	0
8	Đỗ Tuấn Cường	0	1	0
9	Trần Hào Long	0	0	1
10	Phan Trung Tín	0	1	0

staff_id	staff_name	4	5	6
1	Đặng Công Thành	2	0	0
2	Đỗ Phương Nghi	1	1	1
3	Đỗ Thị Thúy Vy	2	3	0
4	Nguyễn Hoàn Mỹ	1	2	1
5	Nguyễn Ngọc H...	1	0	1
6	Bùi Minh Đạt	0	0	2
7	Võ Lê Khang	0	1	1
8	Đỗ Tuấn Cường	0	1	0
9	Trần Hào Long	2	0	1
10	Phan Trung Tín	0	0	1

9. Viết store procedure và fuction

-----Function + Procedure-----

-- hàm tìm mức giá trị cao nhất trong các sản phẩm ở cửa hàng

CREATE FUNCTION dbo.MaxPrice()

RETURNS VARCHAR(100)

AS

BEGIN

RETURN

(

SELECT

TOP(1) price

FROM products

ORDER BY price DESC

)

END;

-- Sử dụng hàm MaxPrice

SELECT

product_id,

product_name,

price

FROM products

WHERE price IN (dbo.MaxPrice())

-- hàm xuất ra tổng tiền mỗi hóa đơn

```

CREATE FUNCTION dbo.SumBill()
RETURNS @SumBill TABLE (order_id int, total_bill int)
AS
BEGIN
    INSERT INTO @SumBill
        SELECT
            order_id,
            SUM(price*quantity) total_bill
        FROM order_items o
        JOIN products p
        ON o.product_id = p.product_id
        GROUP BY order_id
    RETURN;
END;

-- sử dụng hàm SumBill
SELECT *
FROM dbo.SumBill();

-- Thủ tục thêm dữ liệu của một hóa đơn mới
CREATE PROCEDURE InsertOrder(@order_id int, @order_date datetime, @customer_id int, @staff_id int)
AS
BEGIN
    IF EXISTS(SELECT customer_id FROM customers WHERE @customer_id = customer_id)
    BEGIN
        IF EXISTS(SELECT staff_id FROM staffs WHERE @staff_id = staff_id)
        BEGIN
            IF NOT EXISTS(SELECT order_id FROM orders WHERE order_id = @order_id)
            BEGIN
                INSERT INTO orders(order_id, order_date, customer_id, staff_id)
                VALUES (@order_id, @order_date, @customer_id, @staff_id);
                PRINT N'Thêm thành công';
            END
        ELSE
            PRINT N'Thêm thất bại do hóa đơn '+' LTRIM(STR(@order_id)) + N' đã tồn tại';
        END
    ELSE
        PRINT N'Thêm thất bại do nhân viên '+' LTRIM(STR(@staff_id)) + N' chưa tồn tại';
    END
ELSE
    PRINT N'Thêm thất bại do nhân viên '+' LTRIM(STR(@staff_id)) + N' chưa tồn tại';
END

```



```

END

ELSE

    PRINT N'Thêm thất bại do khách hàng '+' + LTRIM(STR(@customer_id)) + N' chưa tồn tại';

END;

-- Sử dụng thủ tục InsertOrder

EXEC InsertOrder '1', '4/13/2022', '1', '1'

-- Thủ tục thêm dữ liệu một sản phẩm mới

CREATE PROCEDURE InsertProduct(@product_id int, @product_name varchar(100), @brand_id int, @category_id
int, @stock_id int, @model_year int, @price decimal(18, 0), @inventory int)
AS
BEGIN
    IF EXISTS(SELECT brand_id FROM brands WHERE @brand_id = brand_id)
    BEGIN
        IF EXISTS(SELECT category_id FROM categories WHERE @category_id = category_id)
        BEGIN
            IF EXISTS(SELECT stock_id FROM stocks WHERE stock_id = @stock_id)
            BEGIN
                IF NOT EXISTS(SELECT product_id FROM products WHERE product_id =
@product_id)
                BEGIN
                    INSERT INTO products(product_id, product_name, brand_id,
category_id, stock_id, model_year, price, inventory)
                    VALUES (@product_id, @product_name, @brand_id, @category_id,
@stock_id, @model_year, @price, @inventory);

                    PRINT N'Thêm thành công';

                END
            ELSE
                PRINT N'Thêm thất bại do sản phẩm '+' + LTRIM(STR(@product_id)) +
N' đã tồn tại';

            END
        ELSE
            PRINT N'Thêm thất bại do kho '+' + LTRIM(STR(@stock_id)) + N' chưa tồn tại';

        END
    ELSE
        PRINT N'Thêm thất bại do loại xe '+' + LTRIM(STR(@category_id)) + N' chưa tồn tại';

    END
END

```

```

ELSE
    PRINT N'Thêm thất bại do hãng xe ' + LTRIM(STR(@brand_id)) + N' chưa tồn tại';
END;

```

```

-- Sử dụng thủ tục InsertProduct

```

```

EXEC InsertProduct '19', 'Wave RSX ', '1', '1', '1', '2022', '30000000', '13'

```

10. Viết trigger ràng buộc dữ liệu cho các bảng

```

-----Trigger-----

```

```

--> INSERT

```

```

CREATE TRIGGER KiemtraInsert

```

```

ON dbo.order_items FOR INSERT

```

```

AS

```

```

BEGIN

```

```

    DECLARE @tonkho INT;

```

```

    SELECT @tonkho = dbo.products.inventory FROM dbo.products, inserted WHERE
    inserted.product_id=dbo.products.product_id

```

```

    IF @tonkho=0

```

```

    BEGIN

```

```

        RAISERROR(N'Out of stock', 16,1)

```

```

        ROLLBACK TRANSACTION

```

```

    END

```

```

    ELSE

```

```

    BEGIN

```

```

        UPDATE dbo.products SET inventory = inventory - inserted.quantity FROM inserted, order_items
        WHERE inserted.product_id=dbo.products.product_id

```

```

    END

```

```

END

```

```

GO

```

```

--> DELETE

```

```

CREATE TRIGGER KiemtraDelete

```

```

ON dbo.order_items FOR DELETE

```

```

AS

```

```

BEGIN

```

```

    DECLARE @tonkho INT;

```

```

    SELECT @tonkho = dbo.products.inventory FROM dbo.products, deleted WHERE
    deleted.product_id=dbo.products.product_id

```

```

    BEGIN

```

```

UPDATE dbo.products SET inventory = inventory + deleted.quantity FROM deleted, order_items
WHERE deleted.product_id=dbo.products.product_id

END

END

GO

--> UPDATE

CREATE TRIGGER KiemtraUpdate
ON dbo.order_items FOR UPDATE
AS
BEGIN

    DECLARE @tonkho INT;

    SELECT @tonkho = dbo.products.inventory FROM dbo.products, inserted WHERE
inserted.product_id=dbo.products.product_id

    IF @tonkho=0
    BEGIN

        RAISERROR(N'Out of stock', 16,1)

        ROLLBACK TRANSACTION

    END

    ELSE

    BEGIN

        UPDATE dbo.products SET inventory = inventory - (inserted.quantity-deleted.quantity) FROM
deleted, inserted, order_items WHERE deleted.product_id=dbo.products.product_id and
inserted.product_id=dbo.products.product_id

    END

END

GO

-----Kiemtra trigger

SELECT * FROM order_items

SELECT * FROM products

SELECT * FROM orders

INSERT INTO dbo.order_items(order_id,item_id,product_id,quantity)
VALUES (25,3,1,1)

DELETE FROM order_items
WHERE order_id=25 and item_id = 3

```

```

UPDATE order_items
    SET quantity=quantity+1
    WHERE order_id=25 and item_id=3

```

11. Phân quyền

-- Phân quyền cho cấp nhân viên truy cập vào cơ sở dữ liệu

```

CREATE LOGIN Staffs WITH PASSWORD = '31211027658'
CREATE USER NgocHang FOR LOGIN Staffs
GRANT SELECT ON categories TO NgocHang
GRANT SELECT ON brands TO NgocHang
GRANT SELECT ON stocks TO NgocHang
GRANT SELECT ON products TO NgocHang
GRANT SELECT ON stores TO NgocHang
GRANT SELECT(order_id, order_date), UPDATE ON orders TO NgocHang
GRANT UPDATE ON order_items TO NgocHang

```

12. Sao lưu dữ liệu

-----Sao lưu dữ liệu -----

```

CREATE PROCEDURE SaoLuuDuLieu(@tencsdl nvarchar(200), @tentaptin nvarchar(200))
AS
BEGIN
    BACKUP DATABASE @tencsdl TO DISK = @tentaptin
END

```

```

EXEC SaoLuuDuLieu 'qlyxemay_nhat_nhan', 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\qlyxemay_nhat_nhan.bak'

```

-----Khôi phục dữ liệu -----

```

CREATE PROCEDURE PhucHoiDuLieu(@tencsdl nvarchar(200), @tentaptin nvarchar(200))
AS
BEGIN
    RESTORE DATABASE @tencsdl FROM DISK = @tentaptin
END

```

```

EXEC PhucHoiDuLieu 'qlyxemay_nhat_nhan', 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Backup\qlyxemay_nhat_nhan.bak'

```