

BÁO CÁO BÀI TẬP NHÓM BẢN CHÍNH THỨC

Học phần: Thực hành CSDL – IT3290

Nhóm 4 – 156778 – GV: TS. Nguyễn Thị Oanh

BÀI TOÁN QUẢN LÝ SIÊU THỊ

Hà Nội, 6/2025

MỤC LỤC

A. THÔNG TIN NHÓM	4
B. THÔNG TIN BÀI TẬP.....	4
1. Tổng hợp tài liệu – Hướng dẫn sử dụng:	4
a. Tài liệu bao gồm:.....	4
b. Hướng dẫn sử dụng báo cáo:	6
2. Mô tả nghiệp vụ:	6
A. Mua bán hàng:	6
B. Nhập hàng và Quản lí kho:	7
C. Quản lí chung:	8
3. Mô tả chức năng:.....	9
a. Đối với khách hàng	9
b. Đối với nhân viên	9
c. Đối với nhà quản lí.....	11
4. Sơ đồ thực thể liên kết:	13
5. Sơ đồ quan hệ:	15
6. Câu lệnh Truy vấn – Phân tích Hiệu năng:	16
a. Lương Quý Hiếu (20235711):	16
b. Lê Vũ Nguyên Hoàng (20235723):	22
c. Phạm Đào Việt Hoàng (20235727):.....	28
C. NHẬN XÉT – NÂNG CAO:	31
1. Kết quả tự đánh giá:	31
a. Kết quả chung:	31
b. Điểm mới của bài toán:.....	31
2. Nâng cao – Gợi mở:	31
3. Khó khăn:.....	31
4. Phân công nhiệm vụ:	32
5. Phụ lục:	32
Lời kết.....	33

A. THÔNG TIN NHÓM

Họ tên SV	MSSV	STT theo DS lớp
Lương Quý Hiếu (NT)	20235711	14
Lê Vũ Nguyên Hoàng	20235723	15
Phạm Đào Việt Hoàng	20235727	17

B. THÔNG TIN BÀI TẬP

Đề tài: CSDL cho quản lí siêu thị.

Nhiệm vụ:

- + Mô tả được nghiệp vụ các hoạt động diễn ra trong siêu thị.
- + Đưa ra các chức năng cần có của ứng dụng sẽ xây dựng, có sự phân quyền giữa các chủ thể tham gia.
- + Xây dựng sơ đồ thực thể liên kết và sơ đồ quan hệ.

Giải quyết:

1. Tổng hợp tài liệu – Hướng dẫn sử dụng:**a. Tài liệu bao gồm:**

- Báo cáo chính thức định dạng PDF: bao gồm toàn bộ thông tin của bài tập lớn.
- Slide thuyết trình buổi thi CK môn học: phục vụ cho mục đích thuyết trình bài thi cuối kỳ.
- File Group4_156778_Query.sql tổng hợp tài liệu 30 yêu cầu truy vấn của 3 sinh viên trong nhóm.
- File g4.market.sql lưu toàn bộ câu lệnh SQL khởi tạo CSDL và view, trigger lưu ở trên tài khoản admin_market và CSDL market.
- File CSDL_20242_OanhNT chứa source code chương trình Java để chạy chương trình thử nghiệm trên máy. Để chạy được chương trình, người dùng phải đáp ứng 2 điều kiện:
 1. Điều kiện cần: có cài sẵn phần mềm chạy ngôn ngữ Java, có sử dụng JavaFX để chạy FXML phục vụ vấn đề giao diện, trước khi chạy phải import CSDL từ

- file g4.market.sql về máy chạy trên hệ quản trị CSDL postgresSQL bằng giao diện dòng lệnh hoặc giao diện pgAdmin4.
2. Điều kiện đủ: sau khi tải file về máy, người dùng tìm đến file Main_App.java để bấm nút Run chương trình. Giao diện mở đầu sẽ là màn hình đăng nhập, người dùng có quyền test trên ứng dụng với 3 vai trò: customer, employee và admin.
- + Đối với customer: tài khoản là customer01 -> customer04, mật khẩu mặc định là 123456.
 - + Đối với employee: tài khoản là employee01 -> employee04, mật khẩu mặc định là abc123.
 - + Đối với admin chỉ có 1 tài khoản duy nhất: admin01, mật khẩu mặc định là admin123.
- Các giá trị từ 1 đến 4 là các tài khoản test để truy cập vào theo customer_id và employee_id từ 1 đến 4.
- Toàn bộ chương trình được lưu ở trên GitHub, giảng viên và các bạn có thể truy cập bằng 2 cách sau (hiện tại đang để chế độ public):**
- Cách 1:** [HieuLuong1/CSDL.20242.156778: Dự án code CSDL và chương trình Java](https://github.com/HieuLuong1/CSDL.20242.156778). Hoặc:
- <https://github.com/hieuluong1/csd1.20242.156778/tree/main>
- Cách 2:** Quét chương trình thông qua mã QR Code:



3. Hướng dẫn sử dụng báo cáo:

- Báo cáo cho biết thông tin cá nhân của các thành viên trong nhóm 4_156778_IT3290. Về nội dung chuyên môn, báo cáo sẽ trình bày lần lượt từ mô tả nghiệp vụ bài toán quản lý siêu thị, mô tả chức năng của hệ thống, thiết kế sơ đồ thực thể liên kết, chuyển sang sơ đồ quan hệ. Ngoài ra ở mục số 6 đã tổng hợp 30 câu hỏi truy vấn và phân tích hiệu năng của từng câu lệnh của mỗi thành viên. Mục nhận xét – nâng cao là những gì đã làm được của nhóm, những ý tưởng và gợi mở còn có thể phát triển của dự án này và những khó khăn mà nhóm gặp phải trong quá trình thiết kế. Ở phần cuối cùng, nhóm xin gửi đến giảng viên TS. Nguyễn Thị Oanh lời cảm ơn chân thành đối với nhóm trong học kì 2024.2.
- Phần phụ lục là các cập nhật sau mỗi phiên bản (mỗi phiên bản đều được gửi trên Teams của nhóm).

2. Mô tả nghiệp vụ:

Nghiệp vụ được chia thành 3 luồng hoạt động chính gồm hoạt động mua bán hàng hóa, hoạt động nhập hàng và quản lý hàng hóa và hoạt động quản lý nhân sự của siêu thị.

A. Mua bán hàng:

a. Quản lý thông tin khách hàng:

- Khi khách đến mua hàng, thu ngân có thể tạo hồ sơ khách hàng, mỗi **khách hàng** được lưu bởi 1 mã khách hàng khác nhau, tên, SDT, email (nếu có) của khách.
- Khách hàng có thể kiểm tra được lịch sử mua hàng của mình.

b. Bán hàng và thanh toán:

- Khách hàng mang giỏ hàng đến quầy thu ngân, nhân viên sẽ quét mã hàng hóa, nhập số lượng bán ra, hệ thống tự động đưa ra số tiền niêm yết của từng sản phẩm * số lượng sản phẩm đó, thuế VAT sau đó xuất ra tổng tiền trên **hóa đơn**. Hiện tại, siêu thị chưa áp dụng hình thức Khuyến mãi nào.
- Hóa đơn của khách hàng được hiển thị trên màn hình bao gồm thông tin khách hàng, ngày giờ mua hàng, tên siêu thị (nếu siêu thị rộng có thể thêm quầy số bao nhiêu), nhân viên thu ngân là ai, thông tin sản phẩm bán ra và số tiền cần chi trả.
- Sau đó, khách hàng lựa chọn hình thức thanh toán. Siêu thị nhận được 3 hình thức thanh toán bao gồm tiền mặt, tiền chuyển khoản hoặc ví điện tử. Khách hàng thanh toán đúng số tiền gửi, nhân viên xác nhận thanh toán, hệ thống ghi nhận đơn hàng, gửi số tiền thu được đến nghiệp vụ doanh thu, gửi số hàng đã bán đến nghiệp vụ quản lý kho cập nhật số lượng...

B. Nhập hàng và Quản lý kho:

c. Quản lý hàng hóa và kho:

- Cứ 1-2 tuần cố định, nhân viên sẽ tiến hành kiểm tra lại **kho** và **hàng hóa** tồn, hàng sắp hết ở trong kho. Từ đó người nhân viên có thể cập nhật kho (còn loại hàng nào, số lượng bao nhiêu, của **nhà cung cấp** (NCC) nào, chất lượng hàng, HSD...). Bên cạnh đó, để quản lý số lượng thật chặt chẽ hơn nữa để tránh thất thoát sản phẩm, quản lý cũng cần phải đi kiểm kê thực tế và đột xuất. Khi kiểm kê số lượng thực tế, người quản lý cần có 1 phiếu kiểm tra cụ thể, trên đó là đầy đủ thông tin mã đơn kiểm kê, ngày giờ, lô hàng đã kiểm, số lượng thực tế, số lượng được lưu trên hệ thống.
- Mỗi sản phẩm của siêu thị sẽ được phân loại vào từng chủng loại cụ thể như Sữa và sản phẩm chế biến từ sữa, Rau củ quả, Thịt...
- Hàng hóa khi được đưa ra khỏi kho, hệ thống sẽ cập nhật số lượng còn lại trong kho.
- Đối với những loại hàng hóa có HSD, siêu thị cần phải kiểm tra nghiêm ngặt sản phẩm theo từng lô hàng nhập về của sản phẩm đó. Giả sử sản phẩm Sữa Ông Thọ nhập về từ lô SD1111 có hạn sử dụng là 30/6/2025, thì người quản lý và nhân viên phải nắm được lô đó nhập từ khi nào, nhập số lượng bao nhiêu, còn bao nhiêu sản phẩm chưa bán trong lô đó để có thể chủ động trong vấn đề xử lý lô hàng tồn khi gần hết HSD.

d. Nhập hàng:

- Nhân viên tạo phiếu nhập hàng đến từng NCC về tên hàng, số lượng, địa điểm, thời gian nhận hàng. Siêu thị nhập từ mỗi nhà cung cấp nhiều sản phẩm và cũng không độc quyền sản phẩm nào, do đó mỗi sản phẩm cũng có nhập từ nhiều nhà cung cấp khác nhau.
- NCC gửi tới siêu thị hóa đơn thanh toán hàng hóa nhập bao gồm các thông tin về NCC (tên, địa chỉ, email, SĐT khi cần làm việc với ai), thông tin lô hàng, ngày nhập hàng, ngày hết hạn (nếu có), đơn giá, thuế VAT, tổng tiền phải thanh toán.
- Khi hàng về, nhân viên kiểm tra hàng hóa theo yêu cầu của người quản lý thông qua 1 biên bản nhập hàng, biên bản này là sự tổng hợp của tờ phiếu nhập hàng với hóa đơn nhập hàng. Nghĩa là đối với mỗi sản phẩm và số lượng yêu cầu, siêu thị sẽ kiểm tra xem đã đủ lô và số lượng nhận về chưa, nếu đủ rồi thì đó là những lô nào, HSD đến bao giờ?..

C. Quản lý chung:

e. Quản lý nhân viên:

- Khi **nhân viên** được nhận vào hệ thống, siêu thị và người quản lý cần thiết lập và lưu trữ mã nhân viên, thông tin (mã CCCD, tên, tuổi, ngày sinh, giới tính, địa chỉ nhà, SĐT và email liên lạc nhân viên đó).
- Quản lý cập nhật lịch làm việc theo tháng của nhân viên lên hệ thống. Mỗi ca làm việc có thể có nhiều nhân viên. Mỗi ngày, quản lý có thể điểm danh nhân viên và kèm ghi chú thưởng phạt theo ca làm (đồng phục, đầu tóc...) đối với mỗi nhân viên thông qua hệ thống giám sát. Nhân viên có thể xem được lịch làm việc của bản thân, xem được tình trạng ngày hôm đó (đi làm/ nghỉ/ muộn/ phép). Đây là cơ sở để nhà quản lý nắm được ngày công và ngày nghỉ, chế độ lương, thưởng phạt của nhân viên.
- Nếu có việc nghỉ phải báo lại cho quản lý trước thời gian nghỉ (bao lâu) để bố trí nhân sự thay thế, trừ trường hợp các lý do đặc biệt, ... để quản lý tính vào ngày phép và chấm công.
- Khi nhập hóa đơn thanh toán hay báo cáo kho, tên của nhân viên cũng sẽ được hiển thị cụ thể để nhà quản lý nắm được thông tin bán hàng, có thể truy xuất nếu có vấn đề bất ngờ gì đó.

f. Quản lý doanh số:

- Siêu thị cần quản lý được số tiền nhận/chi trong từng tháng.
- Về hoạt động chi, siêu thị quản lý được số tiền đã xuất ra để nhập hàng từng NCC, tính được bảng **lương** của nhân viên (dựa trên ngày công, thưởng phạt theo quy định, số tiền theo ngày, thưởng phạt), các chi phí phát sinh sửa chữa hoặc chi phí ổn định cần chi hàng tháng.
- Về hoạt động nhận, siêu thị quản lý được số tiền nhận của mỗi đơn hàng, cập nhật được theo ngày giờ cụ thể số tiền nhận được.

3. Mô tả chức năng:

Cần xây dựng 1 hệ thống phân quyền cho 3 đối tượng sử dụng chính gồm khách hàng, nhân viên, quản lý siêu thị.

a. Đối với khách hàng

- Ở màn hình khách hàng, khách có thể nhìn thấy 2 nút bấm “*Thông tin khách hàng*” và “*Lịch sử mua hàng*”. Khi bấm vào nút 1 sẽ hiển thị toàn bộ thông tin bản thân, do siêu thị chỉ bán trực tiếp nên nếu muốn sửa thông tin gì khách hàng hãy đến siêu thị để nhờ nhân viên chỉnh sửa. Đối với nút số 2 khi bấm vào sẽ hiển thị toàn bộ các hóa đơn mua hàng bao gồm mã hóa đơn, thời gian mua, tổng tiền đã chi trả cho từng hóa đơn. Ngoài ra khi bấm vào phần xem chi tiết mỗi hóa đơn, khách hàng có thể xem được đầy đủ từng sản phẩm và số lượng, tổng tiền đã mua của sản phẩm đó.

Chức năng	Đầu vào	Đầu ra
Thông tin khách hàng		Toàn bộ thông tin của Khách lưu trong CSDL.
Lịch sử mua hàng	Chọn 1 bản ghi	Chi tiết các sản phẩm và số lượng, giá tiền đã mua các sản phẩm.

b. Đối với nhân viên

- Nhân viên sẽ có các chức năng chính bao gồm Thông tin cá nhân, Lịch làm việc, Quản lý khách hàng, Quản lý kho, Tạo hóa đơn, Nhập hàng, Nhà cung cấp.
- Đối với chức năng “*Thông tin cá nhân*” khi bấm vào sẽ hiển thị đầy đủ thông tin của nhân viên đó lên hệ thống, nhân viên chỉ có thể xem không thể sửa các thông tin này, cũng như chỉ xem được của bản thân không xem được của người khác.
- Đối với chức năng “*Lịch làm việc*”, nhân viên có thể theo dõi được lịch làm việc của bản thân trong tháng cụ thể giờ cụ thể, xem được đi làm bao nhiêu buổi, vắng bao nhiêu buổi. Nhân viên có thể chọn lọc theo tháng, năm để lọc theo lịch làm việc của tháng, năm đó. Bấm vào nút xem chi tiết để xem cụ thể từng ngày. Nhân viên có thể lựa chọn nút “*Xem lương*” ở ngay bên cạnh trong tháng đó trong trường hợp quản lý đã cho phép xem. Bảng lương sẽ bao gồm đầy đủ mọi thông tin của nhân viên được lưu trên hệ thống siêu thị.
- Đối với chức năng “*Quản lý khách hàng*” khi bấm vào, nhân viên có thể tìm kiếm được khách hàng trong hệ thống bằng tên hoặc số điện thoại của họ, có thể tạo mới khách hàng nếu họ mới đến đây lần đầu hoặc cập nhật thông tin khách hàng nếu bấm vào ô đó.
- Chức năng “*Quản lý kho*” cho phép nhân viên xem và tìm kiếm được các sản phẩm của siêu thị hiện có. Ngoài ra khi lựa chọn phần “*Thêm sản phẩm*”, nhân viên cũng có thể tạo mới các sản phẩm được thêm vào trong siêu thị. Ở đây, nhân viên cũng

có thể xem được các mặt hàng mới được nhập về thông qua chức năng cùng tên “*Hàng mới nhập*”. Chức năng này cho phép nhân viên xem được các lô hàng mới cập nhật của siêu thị và có thể tìm kiếm theo ngày hoặc theo nhà cung cấp.

- Chức năng “*Tạo hóa đơn*” sẽ được sử dụng khi khách mua hàng, chỉ đơn giản là tạo hóa đơn và các thông tin cơ bản trong hóa đơn đó. Nhân viên điền thông tin sản phẩm, sau đó hiển thị ra các lô hàng và số lượng còn của chúng (trên thực tế siêu thị đã có gắn nhãn các lô được đưa ra quầy ngày hôm đó còn trong phạm vi môn học này nhóm sẽ lựa chọn hình thức chọn lô hàng coi như nhân viên biết lô nào đã được bán ra), các sản phẩm này khi bấm vào sẽ được lựa chọn và điền đầy đủ các thông tin khác, nhân viên có thể tạo 1 đơn hàng mới.
- Ở phần “*Nhập hàng*” nhân viên có quyền tạo 1 biên bản nhập hàng sau khi hàng hóa đến, nhân viên nhập vào các lô được nhận về trong ngày hôm đó, đếm đúng số lượng rồi điền vào máy, sau đó sẽ được lưu trên hệ thống.
- Ở phần “*Nhà cung cấp*”, nhân viên có thể xem được chi tiết các nhà cung cấp và các thông tin của họ.

Chức năng	Đầu vào	Đầu ra
Thông tin cá nhân		Toàn bộ thông tin của nhân viên.
Lịch làm việc	Chọn tháng, năm	Thời gian làm việc cụ thể, điểm danh từng bản ghi cụ thể (D, M, V, P)
Lương	*Chỉ xem được khi thông báo có lương của tháng đó, quản lí đã tạo 1 bảng lương.	Chi tiết bảng lương tháng.
Tìm kiếm khách hàng	SDT Khách hàng	Thông tin Khách Hàng
Tạo Khách hàng	Họ tên, SDT, Email nếu có	Khách hàng được tạo.
Quản lí kho	Tìm kiếm tên sản phẩm	Thông tin sản phẩm.
Thêm hàng hóa	Nhập thông tin hàng hóa	Hàng hóa mới được tạo.
Hàng mới nhập		Toàn bộ các lô hàng mới nhập về.
Cập nhật/Xóa sản phẩm	Thông tin sản phẩm, số tiền, chủng loại mới	Kết quả mong muốn.
Tạo hóa đơn	Các thông tin của hóa đơn, sản phẩm trong hóa đơn đó.	Hóa đơn mới được tạo.
Nhập hàng	Thông tin biên bản nhập hàng, các lô được nhập.	Biên bản được nhập.

Nhà cung cấp		Xem thông tin Nhà cung cấp.
--------------	--	-----------------------------

c. Đối với nhà quản lí

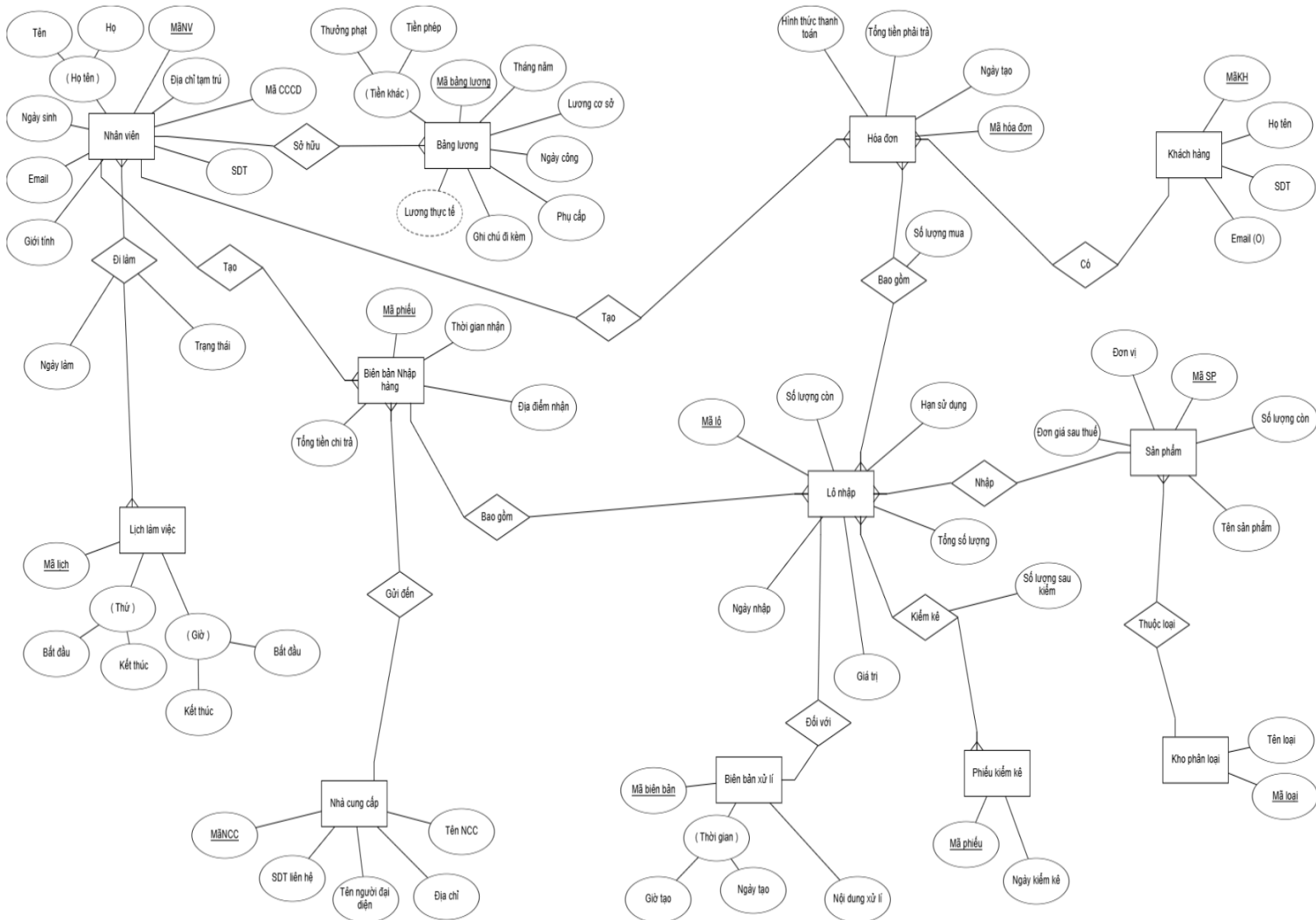
- Quản lí sẽ có các chức năng chính bao gồm Lịch làm việc, Quản lí khách hàng, Quản lí kho, Lương, Tạo hóa đơn, Nhập hàng, Nhà cung cấp, Biên bản kiểm kê, xử lí.
- Đối với chức năng “*Lịch làm việc*”, quản lí có thể tìm kiếm theo thông tin của người nhân viên đó. Ngoài ra khi bấm vào từng dòng nhân viên và từng cột giá trị của nhân viên, quản lí có thể cập nhật thông tin của người đó cho đúng với sự thay đổi giá trị. Quản lí có thể thêm 1 nhân viên mới vào hệ thống. Hơn nữa, quản lí cũng có thể bấm vào nút “*Thêm lịch làm việc*” để chọn giờ làm và lịch làm cho nhân viên trong tháng đó. Khi bấm vào đây và chọn các giá trị tháng, người nhân viên, ngày đầu tiên làm việc của tháng và lịch làm việc, quản lí có thể gán lịch làm việc cho nhân viên. Ngoài ra, sau khi có lịch làm việc rồi, quản lí có thể chọn ngày làm việc rồi lựa chọn tình trạng của ngày hôm đó để “*Điểm danh*” cho nhân viên.
- Chức năng “*Lương*” cho phép quản lí xem được chi tiết các bảng lương. Quản lí có thể lựa chọn tháng năm cụ thể, sau đó bấm “*Thêm bảng lương mới*”, quản lí có thể bấm chọn các mã nhân viên, các thông tin cần thiết để bảng lương được tạo lập thành công.
- Đối với chức năng “*Quản lí khách hàng*” khi bấm vào, quản lí có thể tìm kiếm được khách hàng trong hệ thống bằng tên hoặc số điện thoại của họ, có thể tạo mới khách hàng nếu họ mới đến đây lần đầu.
- Chức năng “*Quản lí kho*” cho phép quản lí xem và tìm kiếm được các sản phẩm của siêu thị hiện có. Ngoài ra khi lựa chọn phần thêm sản phẩm, quản lí cũng có thể tạo mới các sản phẩm được thêm vào trong siêu thị. Ở đây, quản lí cũng có thể xem được các mặt hàng mới được nhập về thông qua chức năng cùng tên “*Hàng mới nhập*”. Chức năng này cho phép quản lí xem được các lô hàng mới cập nhật của siêu thị.
- Chức năng “*Tạo hóa đơn*” sẽ được sử dụng khi khách mua hàng, chỉ đơn giản là tạo hóa đơn và các thông tin cơ bản trong hóa đơn đó. Quản lí điền tên sản phẩm, sau đó hiển thị ra các lô hàng và số lượng còn của chúng, các sản phẩm này khi bấm vào sẽ được lựa chọn và điền đầy đủ các thông tin khác, nhân viên có thể tạo 1 đơn hàng mới.
- Ở phần “*Nhập hàng*”, quản lí có quyền tạo 1 biên bản nhập hàng sau khi hàng hóa đến, nhân viên nhập vào các lô được nhận về trong ngày hôm đó, đếm đúng số lượng rồi điền vào máy, sau đó sẽ được lưu trên hệ thống.

- Quản lí còn có thêm 1 quyền nữa đó là “*Nhà cung cấp*” đến của siêu thị. Quản lí có thể thêm, xóa các thông tin của nhà cung cấp đến với siêu thị.
- Chức năng “*Tạo biên bản*” cho phép quản lí tạo 2 loại biên bản thường dùng trong siêu thị: biên bản kiểm kê và biên bản xử lí hàng hóa. Đối với “*Biên bản kiểm kê*”, khi lựa chọn tạo 1 biên bản mới, khi quản lí chọn mã lô mà mình muốn kiểm tra thì ngay lập tức hiển thị số lượng nhập vào và số lượng còn của lô đó, tuy nhiên khi có thêm 1 ô trống để quản lí có thể cập nhật vào đó số lượng thực tế mà nhà quản lí đã đi kiểm tra. Bấm lưu kết quả để tạo mới 1 bản ghi. Đối với “*Biên bản xử lí*”, nhà quản lí thêm đầy đủ các thông tin như ngày giờ, mã lô, nội dung xử lí để tiến hành xử lí lô hàng đó, bấm lưu để tạo mới 1 bản ghi xử lí sản phẩm.

Chức năng	Đầu vào	Đầu ra
Quản lí nhân viên	Họ tên NV	Thông tin nhân viên.
Thêm lịch làm việc	Tháng, nhân viên, lịch làm việc, ngày bắt đầu làm	Nhân viên đã được gán lịch làm việc.
Điểm danh	Ngày, tên NV, trạng thái	Kết quả điểm danh nhân viên sẽ nhìn thấy được.
Lương	Cập nhật, chỉnh sửa bảng lương	Bảng lương mới được tạo.
Thêm bảng lương mới		Tạo thêm 1 bảng lương rỗng chờ quản lí cập nhật.
Quản lí kho	Tìm kiếm tên sản phẩm	Thông tin sản phẩm.
Thêm hàng hóa	Nhập thông tin hàng hóa	Hàng hóa mới được tạo.
Hàng mới nhập		Toàn bộ các lô hàng mới nhập về.
Cập nhật/Xóa sản phẩm	Thông tin sản phẩm, số tiền, chủng loại mới	Kết quả mong muốn.
Tạo hóa đơn	Các thông tin của hóa đơn, sản phẩm trong hóa đơn đó.	Hóa đơn mới được tạo.
Nhập hàng	Thông tin biên bản nhập hàng, các lô được nhập.	Biên bản được nhập.
Nhà cung cấp		Xem thông tin Nhà cung cấp.
Biên bản xử lí	Tạo biên bản xử lí, chọn 1 lô hàng, điền nội dung xử lí.	Biên bản được khởi tạo.
Biên bản kiểm kê	Tạo 1 biên bản kiểm kê, chọn nhiều lô hàng, điều chỉnh giá trị thực tế	Biên bản được khởi tạo.

4. Sơ đồ thực thể liên kết:

Từ nghiệp vụ chi tiết, nhóm đã xây dựng được lên sơ đồ thực thể liên kết.



Một số giải thích thực thể:

- Khách hàng: là người mua hàng khi đến siêu thị, được lưu trữ thông tin gồm Mã KH, Họ tên, SĐT và Email nếu có.
- Hóa đơn: là biên bản giao dịch mỗi lần sử dụng dịch vụ của khách hàng khi đến siêu thị, hóa đơn cần phải cho biết Mã hóa đơn, Ngày tạo, Tổng tiền phải trả và Hình thức thanh toán.
- Lô nhập là các lô hàng chứa các sản phẩm của siêu thị, trên lô nhập là Mã lô, ngày nhập, số lượng còn, hạn sử dụng, tổng số lượng và giá trị của lô.
- Sản phẩm của siêu thị bao gồm Mã sản phẩm, Tên sản phẩm, Số lượng còn, Đơn vị và Đơn giá sau thuế.
- Kho phân loại là các loại hàng chung của sản phẩm, bao gồm Mã kho và Tên kho.
- Phiếu kiểm kê là biên bản mà quản lý có thể nhập để đánh giá kiểm tra sản phẩm trong kho theo từng lô, bao gồm Mã phiếu và Ngày kiểm kê.
- Biên bản xử lý là các biên bản dùng để xử lý sản phẩm bị hỏng hoặc hết hạn sử dụng, cần xử lý sản phẩm ngay lập tức và có nội dung cụ thể. Biên bản xử lý bao gồm Mã biên bản, Ngày tạo, Giờ tạo, Nội dung xử lý.
- Biên bản nhập hàng là biên bản sau cùng trong nghiệp vụ nhập hàng mà nhân viên phải quan tâm bao gồm Mã phiếu, Thời gian nhận, Địa điểm nhận, Tổng số tiền chi trả.
- Nhà cung cấp là thông tin các đối tác mà siêu thị hợp tác, bao gồm Mã NCC, Tên NCC, Tên người đại diện, SĐT liên hệ, Địa chỉ.
- Nhân viên là những người làm việc cho siêu thị, được siêu thị quản lý bằng Mã NV, Họ tên đầy đủ, Ngày sinh, Giới tính, Địa chỉ tạm trú, SĐT, Mã CCCD, Email.
- Bảng lương là tổng hợp lương của nhân viên, được phân biệt bởi Mã bảng lương, Tháng năm, Lương cơ sở, Ngày công, Phụ cấp, Thưởng phạt, Tiền phép, Ghi chú đi kèm, Lương thực tế.
- Lịch làm việc là chi tiết các giờ làm và ngày làm của nhân viên, bao gồm Mã lịch, Thứ bắt đầu, Thứ kết thúc, Giờ bắt đầu, Giờ kết thúc.

Các liên kết trong sơ đồ:

Thực thể A	Liên kết	Thực thể B
Khách hàng	Có (1 – n)	Hóa đơn
Hóa đơn	Bao gồm (n – n)	Lô nhập
Lô nhập	Nhập (n – 1)	Sản phẩm
Sản phẩm	Thuộc loại (n – 1)	Kho phân loại
Phiếu kiểm kê	Kiểm kê (n – n)	Lô nhập
Biên bản xử lý	Xử lý (1 – 1)	Lô nhập
Biên bản nhập hàng	Bao gồm (1 – n)	Lô nhập
Biên bản nhập hàng	Gửi đến (n – 1)	Nhà cung cấp
Nhân viên	Tạo (1 – n)	Hóa đơn

Nhân viên	Tạo (1 – n)	Biên bản nhập hàng
Nhân viên	Đi làm (n – n)	Lịch làm việc
Nhân viên	Sở hữu (1 – n)	Bảng lương

Bảng trên có 7 liên kết bậc 2, 0 liên kết bậc 1 (đệ quy).

Dựa vào bảng trên, đối với các liên kết n – n như Bao gồm, Đi làm có thể tạo ra 1 quan hệ mới, các liên kết 1 – n, n – 1, 1 – 1 có thể tạo thành khóa ngoài trong sơ đồ quan hệ.

5. Sơ đồ quan hệ:

Sơ đồ quan hệ được chuyển từ sơ đồ thực thể liên kết:

Nhân viên (MãNV, Họ, Tên, Ngày sinh, Email, Giới tính, Địa chỉ tạm trú, SĐT, Mã CCCD)

Lịch làm việc (Mã lịch, Thứ bắt đầu, Thứ kết thúc, Giờ bắt đầu, Giờ kết thúc)

Đi làm (MãNV, Mã lịch, Ngày làm, Trạng thái)

Bảng lương (Mã bảng lương, Tháng năm, Lương cơ sở, Ngày công, Phụ cấp, Ghi chú đi kèm, Thưởng phạt, Tiền phép, Lương thực tế, MãNV)

Khách hàng (MãKH, Họ tên, SĐT, Email)

Hóa đơn (Mã hóa đơn, Ngày tạo, Tổng tiền phải trả, Hình thức thanh toán, MãKH, MãNV tạo)

Lô nhập (Mã lô, Ngày nhập, Hạn sử dụng, Tổng số lượng, Số lượng còn, Mã SP, Mã Nhập hàng, Giá trị)

Bao gồm (Mã hóa đơn, Mã lô, Số lượng mua)

Sản phẩm (Mã SP, Tên sản phẩm, Đơn vị, Đơn giá sau thuế, Số lượng còn, Mã loại)

Kho phân loại (Mã loại, Tên loại)

Phiếu kiểm kê (Mã phiếu, Ngày kiểm kê)

Kiểm tra (Mã phiếu, Mã lô, Số lượng sau kiểm)

Phiếu nhập hàng (Mã phiếu, Thời gian nhập, Địa điểm nhận, Tổng tiền chi trả, MãNV, Mã NCC)

Nhà cung cấp (MãNCC, Tên NCC, Địa chỉ, Tên người đại diện, SĐT liên hệ)

Biên bản xử lý (Mã biên bản, Ngày tạo, Giờ tạo, Nội dung xử lý, Mã lô)

- **Biên dịch CSDL sang bản tiếng Anh để đưa vào lập trình đảm bảo tính thống nhất giữa các thành viên trong nhóm:**

Employee (EmployeeID, Lastname, Firstname, DOB, Email, Gender, Address, Phone, IdentityID)

Schedule (ScheduleID, FromDay, ToDay, TimeStart, TimeEnd)

Working (EmployeeID, ScheduleID, WorkDate, Status)

Salary (SalaryID, MonthYear, BasicSalary, Workdays, Bonus, Note, Reward/Punish, Leavepay, ActualSalary, EmployeeID)

Customer (CustomerID, Fullname, Phone, Email)

Orders (OrderID, Orderdate, Totalamount, Method, *CustomerID*, *EmployeeID*)

Batch (Batch, ImportDate, Expiration, Totalnumber, QuantInStock, *ProductID*, *ImportID*, Value)

Including (*OrderID*, *BatchID*, Quantity)

Products (ProductID, Name, Unit, Price, QuantInStock, *Category*)

Categories (Category, Title)

CheckReport (ReportID, Checkdate)

Checking (*ReportID*, *Batch*, RealQuant)

ImportReport (ImportID, Date, Address, TotalAmount, *EmployeeID*, *SupplierID*)

Supplier (SupplierID, Name, Address, ContactName, Contact)

IncidentReport (IncidentID, Date, Hours, Content, *Batch*)

Lưu ý: phân xem thông tin các cột trong bảng khi đưa lên PostgreSQL đã được mô tả chi tiết ở File g4.market.sql; phần báo cáo này chỉ thể hiện các thuộc tính trong từng quan hệ và các liên kết giữa các bảng để đảm bảo thuộc tính khóa chính và khóa ngoại trong sơ đồ quan hệ.

6. Câu lệnh Truy vấn – Phân tích Hiệu năng:

a. Lương Quý Hiếu (20235711):

--1. Cho biết các đơn hàng và số tiền của từng đơn hàng nhận được trong ngày 5/6/2025:

```
select order_id, total_amount
```

```
from orders
```

```
where order_date = '2025/06/05';
```

--Câu lệnh có thể sử dụng index để tối ưu do phép toán = thỏa mãn điều kiện tìm kiếm của index btree. Nó sẽ sắp xếp dữ liệu theo 1 trình tự rồi lấy ra các kết quả orderdate thỏa mãn.

```
create index idx_orders_date on orders using btree (order_date);
```

--2. Cho biết các khách hàng đã mua hàng ở cửa hàng vào ngày 5/6/2025. Thông tin in ra gồm mã khách hàng, họ tên đầy đủ của khách hàng đó. Lưu ý nếu khách hàng mua từ 2 lần trên 1 ngày cũng chỉ hiện thông tin 1 lần.

```
select distinct customer_id, fullname
```

```
from customer
```

```
where customer_id in (
```

```
    select customer_id
```



```

from orders
where order_date = '2025/06/05');

```

--câu lệnh có thể viết theo nhiều cách khác nhau:

```

select distinct customer_id, fullname
from customer
join orders using (customer_id)
where order_date = '2025/06/05';

```

--Có thể sử dụng chỉ mục trên cột order_date do phép toán = giúp index sắp xếp dữ liệu, thực thi tìm ra nhanh các dòng có giá trị là 2025/06/05.

```

create index idx_orders_date on orders using btree (order_date);

```

--Ở 2 cách viết trên thì cách số 1 dùng truy vấn con, nó sẽ lọc ra các bản ghi customerid thỏa mãn, sau đó sẽ group theo các giá trị trùng nhau rồi mới so sánh với các bản ghi customerid ở câu truy vấn chính. Cách số 2 thì nó lọc ra các bản ghi theo orderid thỏa mãn, rồi kết nối với bảng customer để trả ra kết quả. 2 cách đều có thể dùng index tuy nhiên cách số 2 nhanh hơn cách 1 do máy ko cần lọc để loại truy trong truy vấn con.

--3. Đưa ra sản phẩm được bán nhiều nhất trong tháng 6/2025.

```

with temp as(
    select product_id, product_name, sum(quantity) as banra
    from orders
    join order_details using (order_id)
    join batch using (batch_id)
    join products using (product_id)
    where order_date between '2025/06/01' and '2025/06/30'
    group by product_id, product_name)

```

```

select product_id, product_name, banra
from temp

```

```

where banra = (select max(banra) from temp);

```

--Sử dụng 1 bảng temp để đưa ra các sản phẩm và số lượng được bán ra của chúng.

--Có thể tạo chỉ mục trên order_date (đã tạo) để tăng tốc độ.

```
create index idx_orders_date on orders using btree (order_date);
```

```
with temp as(
```

```
    select product_id, product_name, sum(quantity) as banra
```

```
    from orders
```

```
    join order_details using (order_id)
```

```
    join batch using (batch_id)
```

```
    join products using (product_id)
```

```
    where order_date between '2025/06/01' and '2025/06/30'
```

```
    group by product_id, product_name)
```

```
select product_id, product_name, banra
```

```
from temp
```

```
where banra >= all(select banra from temp);
```

--Ở 2 cách trên, cách số 1 sẽ nhanh hơn do đặt phép toán bằng và 1 truy vấn tìm max con bên trong, do đó hệ thống chỉ cần tính 1 lần ra giá trị max, sau đó nó sẽ chỉ làm việc trên bảng chính theo quy tắc đường ống. Còn ở cách số 2 nó sẽ liên quan đến bộ nhớ, tức là cứ duyệt 1 bản ghi lại so sánh với mọi bản ghi của truy vấn con, sau đó nó mới ghi vào bộ nhớ xem giá trị mới có vượt giá trị max hay không.

--4. Chi tiết đơn hàng có order_id = 1 biết 1 đơn hàng cần bao gồm mã sản phẩm, tên sản phẩm, số lượng bán ra, giá cả mỗi SP, thành tiền mỗi SP

```
select product_id, product_name, quantity, price_with_tax, (quantity *
price_with_tax) as thanhtien
```

```
from order_details
```

```
join batch using (batch_id)
```

```
join products using (product_id)
```

```
where order_id = '1';
```

--Nếu viết theo cách này, order_id = 1 sẽ được lọc ra đầu tiên để giảm kích cỡ của bảng sau đó sẽ thực hiện join. Mặc dù đã có sẵn index khóa chính orderid với batch_id, nhưng hệ thống không lựa chọn sử dụng index này do phép where chỉ có 1 điều kiện và sử dụng index trên 2 cột cũng không nhanh bằng phép lọc thông thường (do máy tự chọn). Tuy nhiên vẫn có thể sử dụng 1 index trên 1 cột

```
create index idx_details_orderid on order_details using btree (order_id);
```

--thì hệ thống sẽ nhận ra chỉ có 1 điều kiện duy nhất và trùng khớp với điều kiện trên index này nên nó dừng => cải thiện truy vấn.

--5. Tạo 1 khung nhìn cho biết các loại sản phẩm hiện có trong siêu thị và số lượng của mỗi loại. Sắp xếp kết quả theo thứ tự giảm dần về số lượng.

create or replace view categories_number as

```
select category_id, category_name, count(product_id) as soluong
from categories
join products using (category_id)
group by category_id, category_name
order by soluong desc;
```

--6. Cho biết nhân viên chăm chỉ nhất siêu thị vào tháng 5/2025 để quản lý tặng thưởng. Nhân viên chăm chỉ là nhân viên đi làm đủ từ 25 buổi/ tháng nhưng không được Vắng (nghỉ không phép) buổi nào.

with temp as(

```
select employee_id, count(*) as dilam
from working
where status = 'D' and (work_date between '2025/05/01' and '2025/05/31')
group by employee_id
having count(*) >= 25),
```

temp_absent as(

```
select employee_id, count(*) as vang
from working
where status = 'V' and (work_date between '2025/05/01' and '2025/05/31')
group by employee_id)
```

select employee_id, concat(lastname, ' ', firstname) as fullname, dilam

from employee

join temp using (employee_id)

left join temp_absent using (employee_id)

where dilam = (select max(dilam) from temp) and vang is null;

--Câu này có thể cải thiện hiệu năng câu lệnh truy vấn bằng cách đặt index trên 2 điều kiện ở phép where trong bảng tạm. Vấn đề cần thiết ở đây là nên đặt index theo 2 cột (status, work_date) hay (work_date, status) hay 2 index trên 2 cột đơn lẻ status với workdate.

```
--create index idx_working_status_date on working using btree (status, work_date);
```

```
--create index idx_date_status on working using btree (work_date, status);
```

```
create index idx_working_date on working using btree (work_date);
```

```
--create index idx_working_status on working using btree (status);
```

--Nhận xét do cột status ít dữ liệu khác nhau ('D', 'M', 'V', 'P') nên thứ tự bị lặp lại nhiều, có nhiều khả năng đặt index ở đây sẽ không hiệu quả do hệ thống sẽ lựa chọn cách duyệt toàn bộ thay vì sắp xếp nhiều bản ghi giống hệt nhau trên cột status rồi mới lọc ra. => index trên status ko hiệu quả. Index trên work_date theo em đáp ứng được nhất trong trường hợp này, nó sẽ giúp lọc ra khoảng ngày tháng năm thỏa mãn rồi sau đó lấy ra các bản ghi có status phù hợp. Đối với index đôi tùy vào 2 cách viết truy vấn thì thứ tự trong where có sự thay đổi => dẫn đến nếu viết theo cách 1 thì index (status, work_date) thực hiện được, còn index (work_date, status) lại ko được, cách 2 thì ngược lại. => có thể nhanh trong 1 số trường hợp, nhưng lại thiếu tính linh hoạt, bắt buộc theo thứ tự. Ngoài ra việc đặt 1 index 2 cột ở đây cũng khá tốn kém chi phí, trừ trường hợp cần truy vấn việc này nhiều lần, ... => gây tốn kém bảo trì, trong khi việc lập theo ngày đã cực hiệu quả rồi (do status bị lặp lại dữ liệu nhiều).

--7. Viết hàm trả về các đơn hàng mà khách hàng đã mua với đầu vào là số điện thoại của khách, kết quả trả về là mã hóa đơn, ngày mua hàng, hình thức thanh toán, tổng số tiền phải trả.

```
create or replace function order_bought(in v_phone varchar(20)) returns table
(orderid int, orderdate date, method varchar(50), totalamount numeric(12,2)) as
```

```
$$
```

```
begin
```

```
    return query select order_id, order_date, payment_method, total_amount
```

```
        from orders
```

```
        join customer using (customer_id)
```

```
        where phone = v_phone;
```

```
end;
```

```
$$ language plpgsql;
```

```
select * from order_bought('0909123456');
```

--8. Cho biết Công ty TNHH Sao Mai đã cung cấp những sản phẩm gì cho siêu thị.

```
select distinct product_id, product_name
```

```
from products
```

```
join batch using (product_id)
```

```
join import_reports using (import_id)
```

```
join suppliers using (supplier_id)
```

```
where supplier_name = 'Cong ty TNHH Sao Mai';
```

--Có thể đặt index trên supplier_name để truy vấn hiệu quả hơn.

```
create index idx_suppliers_name on suppliers using btree (supplier_name);
```

--Tuy nhiên trên thực tế người ta chỉ lưu đầy đủ chứ tìm kiếm thì hiếm khi. Thường khi tìm tên công ty người ta chỉ tìm Sao Mai, hoặc cùng lắm là Công ty Sao Mai, do đó dòng where trên thực tế sẽ có dạng là (tùy vào tên của công ty nữa):

```
where supplier_name like 'Sao Mai%';
```

```
where supplier_name like '%Sao Mai';
```

```
where supplier_name like '%Sao Mai%';
```

--Như vậy trong 3 cách trên thì có đến 2 cách thường xuyên được dùng nhưng lại kém index btree (%name), do đó ở đây việc đặt index sẽ không tối ưu về việc sử dụng lắm.

--9. Viết 1 hàm có đầu vào là mã nhân viên, ngày đi làm và trả về số đơn mà nhân viên đó đã thực hiện trong ngày hôm đó.

```
create or replace function total_order (in v_employeeid int, v_date date) returns integer as
```

```
$$
```

```
declare v_count integer;
```

```
begin
```

```
    select into v_count count(order_id)
```

```
    from orders
```

```
    where employee_id = v_employeeid and order_date = v_date;
```

```
    return v_count;
```

end;

\$\$ language plpgsql;

select total_order(1, '2025/06/09');

--10. Cho biết khách hàng nào mua nhiều đơn nhất từ trước đến giờ.

with temp as(

select customer_id, count(order_id) as mua

from orders

group by customer_id)

select customer_id, fullname, mua

from customer

join temp using (customer_id)

where mua = (select max(mua) from temp);

--Sử dụng 1 bảng tạm để tính số lần mua của khách hàng sau đó đưa ra số lần mua là lớn nhất, phép toán = trong câu lệnh truy vấn chính góp phần giúp truy vấn được thực hiện nhanh hơn sử dụng phép toán >= all.

b. Lê Vũ Nguyễn Hoàng (20235723):

-- 1. Cho biết 5 sản phẩm có đơn giá cao nhất trong siêu thị

SELECT product_id, product_name, unit, price_with_tax

FROM products

ORDER BY price_with_tax DESC

LIMIT 5;

-- Hệ thống phải quét toàn bộ bảng, sau đó sắp xếp toàn bộ theo price_with_tax giảm dần rồi mới lấy ra được 5 dòng đầu tiên. Do đó có thể tạo một chỉ mục cho price_with_tax theo thứ tự giảm để có thể truy cập trực tiếp vào các phần tử lớn nhất mà không cần sắp xếp toàn bộ bảng:

CREATE INDEX idx_products_price_desc ON products USING btree (price_with_tax DESC);

-- 2. Hiện thị toàn bộ danh sách hóa đơn gồm mã hóa đơn, ngày đặt hàng, tên khách hàng, tổng tiền và hình thức thanh toán

```
SELECT o.order_id, o.order_date, c.fullname AS customer_name,
o.total_amount, o.payment_method
```

```
FROM orders o
```

```
JOIN customer c ON o.customer_id = c.customer_id;
```

-- Cách viết khác dùng subquery

```
SELECT o.order_id, o.order_date, (SELECT fullname FROM customer c
WHERE c.customer_id = o.customer_id) AS customer_name, o.total_amount,
o.payment_method
```

```
FROM orders o;
```

-- Ở cách viết 2, hệ thống quét toàn bộ bảng orders và đối với mỗi bản ghi, hệ thống quét tiếp toàn bộ bảng customer để tìm bản ghi có customer_id trùng khớp. Do đó ở cách viết 1, sử dụng JOIN để ghép 2 bảng với nhau sẽ có tốc độ nhanh hơn

-- 3. Thống kê số loại sản phẩm, số lượng hàng tồn kho của mỗi danh mục hàng

```
SELECT ca.category_name, COUNT(p.product_id) AS so_loai_san_pham,
SUM(p.quantity_in_stock) AS tong_so_hang_trong_kho
```

```
FROM categories ca
```

```
LEFT JOIN products p ON ca.category_id = p.category_id
```

```
GROUP BY ca.category_name;
```

-- Hệ thống quét toàn bộ bảng products để tìm các bản ghi có category_id tương ứng với từng bản ghi của bảng categories, sau đó nhóm các bản ghi theo danh mục hàng, rồi thực hiện COUNT và tính SUM

-- 4. Thống kê số lần nhập hàng từ các nhà cung cấp, và đưa ra giá trị đơn nhập cao nhất và thấp nhất của từng nhà cung cấp

```
SELECT s.supplier_id, s.supplier_name, COUNT(ir.import_id) AS
so_lan_nhap, MAX(ir.total_amount) AS tong_cao_nhat, MIN(ir.total_amount) AS
tong_thap_nhat
```

```
FROM suppliers s
```

```
LEFT JOIN import_reports ir ON s.supplier_id = ir.supplier_id
```

```
GROUP BY s.supplier_id, s.supplier_name;
```

-- Hệ thống quét toàn bộ bảng import_reports để tìm các bản ghi có supplier_id tương ứng với mỗi bản ghi của bảng suppliers, nhóm các bản ghi theo nhà cung cấp rồi COUNT, tìm MAX, MIN

-- 5. Đưa ra danh sách khách hàng đã đến siêu thị mua hàng vào tháng 5 năm 2025

```
SELECT DISTINCT c.customer_id, c.fullname
FROM customer c
JOIN orders o ON c.customer_id = o.customer_id
WHERE o.order_date BETWEEN '2025-05-01' AND '2025-05-31'
ORDER BY c.fullname;
```

-- Hệ thống quét bảng orders và lọc các đơn hàng có order_date nằm trong tháng 5/2025. Sau đó JOIN với bảng customer theo customer_id, và loại bỏ trùng lặp bằng DISTINCT. Cuối cùng, kết quả được sắp xếp theo tên khách hàng. Có thể tạo chỉ mục theo order_date và customer_id để tăng tốc độ lọc các đơn hàng trong tháng 5 năm 2025 và tăng tốc độ JOIN 2 bảng theo customer_id

```
CREATE INDEX idx_orders_date_customer ON orders(order_date, customer_id);
```

-- 6. Liệt kê chi tiết các biên bản kiểm kê trong năm 2025

```
SELECT cr.report_id, cr.check_date, b.batch_id, b.product_id,
b.quantity_in_stock AS so_luong_ghi_nhan, cd.real_quantity AS so_luong_thuc_te
FROM check_reports cr
JOIN check_details cd ON cr.report_id = cd.report_id
JOIN batch b ON cd.batch_id = b.batch_id
WHERE cr.check_date >= '2025-01-01'
ORDER BY cr.check_date, cr.report_id, b.batch_id;
```

-- Hệ thống quét toàn bộ bảng check_reports để lọc ra các bản ghi có ngày đúng điều kiện, sau đó JOIN với 2 bảng check_details, batch dựa trên id tương ứng và đưa ra thông tin

-- 7. Tính lợi nhuận (thu - chi) của siêu thị trong tháng 5/2025

```
WITH thu AS (
    SELECT SUM(total_amount) AS tong_thu
    FROM orders
    WHERE order_date BETWEEN '2025-05-01' AND '2025-05-31'
),
```



```

chi_luong AS (
    SELECT SUM(actual_salary) AS tong_chi_luong
    FROM salary
    WHERE monthyear = '5/2025'
),
chi_nhap AS (
    SELECT SUM(total_amount) AS tong_chi_nhap
    FROM import_reports
    WHERE import_date BETWEEN '2025-05-01' AND '2025-05-31'
)
SELECT
    thu.tong_thu,
    chi_luong.tong_chi_luong,
    chi_nhap.tong_chi_nhap,
    thu.tong_thu - chi_luong.tong_chi_luong - chi_nhap.tong_chi_nhap AS
loi_nhuan
FROM thu, chi_luong, chi_nhap;

-- Sử dụng WITH để tạo ra 3 bảng phụ thu là tổng doanh thu từ orders trong
tháng 5 năm 2025, chi_luong là tổng số tiền để trả lương cho nhân viên trong tháng 5
năm 2025, chi_nhap là tổng số tiền nhập hàng trong tháng 5 năm 2025 rồi tổng hợp lại
thành lợi nhuận của siêu thị

-- Có thể tạo các chỉ mục với thời gian trên các bảng để lọc thời gian nhanh hơn
CREATE INDEX idx_orders_date ON orders(order_date);
CREATE INDEX idx_salary_monthyear ON salary(monthyear);
CREATE INDEX idx_import_date ON import_reports(import_date);

-- 8. Tạo bảng view thống kê tổng chi tiêu của mỗi khách hàng
CREATE OR REPLACE VIEW customer_spending_view AS
SELECT c.customer_id, c.fullname, COUNT(o.order_id) AS so_don_hang,
SUM(o.total_amount) AS tong_chi_tieu
FROM customer c

```

```
LEFT JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.fullname;
```

-- 9. Hàm thống kê số ngày đi làm đúng giờ, đi muộn và vắng của nhân viên theo ID trong tháng 5 năm 2025

```
CREATE OR REPLACE FUNCTION
employee_attendance_summary(v_employee_id INT)
RETURNS TABLE (
    employee_id INT,
    so_ngay_dung_gio INT,
    so_ngay_di_muon INT,
    so_ngay_vang INT
) AS
$$
BEGIN
    RETURN QUERY
    SELECT
        w.employee_id,
        COUNT(*) FILTER (WHERE w.status = 'D')::INT AS so_ngay_dung_gio,
        COUNT(*) FILTER (WHERE w.status = 'M')::INT AS so_ngay_di_muon,
        COUNT(*) FILTER (WHERE w.status = 'V')::INT AS so_ngay_vang
    FROM working w
    WHERE w.employee_id = v_employee_id
    AND w.work_date BETWEEN DATE '2025-05-01' AND DATE '2025-05-
31'

    GROUP BY w.employee_id;
END;
$$ LANGUAGE plpgsql;
SELECT * FROM employee_attendance_summary(1);
```

-- Khi nhập vào employee_id, hệ thống sẽ quét bảng working để tìm ra bản ghi có employee_id tương ứng và có ngày nằm trong tháng 5 năm 2025, sau đó thống kê trạng thái từng ngày theo 3 loại đúng giờ, muộn, vắng và trả kết quả ra một bảng mới

-- 10. Hàm đưa ra các sản phẩm sắp hết hạn sau số ngày nhập vào

```
CREATE OR REPLACE FUNCTION expiring_batches(n_days INT)
RETURNS TABLE (
    batch_id INT,
    product_name VARCHAR(100),
    expiration_date DATE,
    quantity_in_stock INT,
    days_left INT
) AS
$$
BEGIN
    RETURN QUERY
    SELECT
        b.batch_id,
        p.product_name,
        b.expiration_date,
        b.quantity_in_stock,
        (b.expiration_date - CURRENT_DATE)::INT AS days_left
    FROM batch b
    JOIN products p ON b.product_id = p.product_id
    WHERE b.expiration_date BETWEEN CURRENT_DATE AND
CURRENT_DATE + n_days;
END;
$$ LANGUAGE plpgsql;
SELECT * FROM expiring_batches(200);
```

-- Khi nhập vào số ngày `n_days`, hệ thống quét bảng `batch` để tìm những lô hàng có hạn sử dụng từ ngày hiện tại đến sau `n_days` ngày nữa, sau đó JOIN với bảng `products` thông qua `product_id` để lấy được tên sản phẩm. Cuối cùng trả kết quả ra bảng với thông tin hàng đó và số ngày còn lại

c. Phạm Đào Việt Hoàng (20235727):

--1. Xem tất cả các đơn hàng của khách hàng "Nguyễn Minh Khoa", sắp xếp theo ngày mua.

```
SELECT o.order_id, o.order_date, o.total_amount, o.payment_method
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
WHERE c.fullname = 'Nguyễn Minh Khoa'
ORDER BY o.order_date DESC;
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_customer_fullname ON customer(fullname);
CREATE INDEX idx_orders_customer_id ON orders(customer_id);
```

--2. Hiện thị tên sản phẩm, số lượng và giá trong hóa đơn có `order_id = 1`.

```
SELECT p.product_name, od.quantity, p.price_with_tax
FROM order_details od
JOIN batch b ON od.batch_id = b.batch_id
JOIN products p ON b.product_id = p.product_id
WHERE od.order_id = 1;
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_batch_product_id ON batch(product_id);
```

--3. Lấy danh sách sản phẩm có số lượng tồn kho < 50.

```
SELECT product_name, quantity_in_stock
FROM products
WHERE quantity_in_stock < 50;
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_products_quantity_in_stock ON  
products(quantity_in_stock);
```

--4. Lấy lương tháng 5/2025 của nhân viên "An Nguyen".

```
SELECT s.monthyear, s.actual_salary  
FROM salary s  
JOIN employee e ON s.employee_id = e.employee_id  
WHERE e.firstname = 'An' AND e.lastname = 'Nguyen' AND s.monthyear =  
'5/2025';
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_salary_employee_month ON salary(employee_id,  
monthyear);
```

```
CREATE INDEX idx_employee_name ON employee(firstname, lastname);
```

--5. Lấy tổng tiền thu được từ đơn hàng theo từng ngày.

```
SELECT order_date, SUM(total_amount) AS daily_revenue  
FROM orders  
GROUP BY order_date  
ORDER BY order_date DESC;
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_orders_order_date ON orders(order_date);
```

--6. Liệt kê 5 sản phẩm được bán nhiều nhất.

```
SELECT p.product_name, SUM(od.quantity) AS total_sold  
FROM order_details od  
JOIN batch b ON od.batch_id = b.batch_id  
JOIN products p ON b.product_id = p.product_id  
GROUP BY p.product_name  
ORDER BY total_sold DESC  
LIMIT 5;
```

-- chỉ mục nên tạo

```
CREATE INDEX idx_batch_product_id ON batch(product_id);
```

--7. Tìm các lô có hạn sử dụng trước ngày 2025-12-31.

```
SELECT batch_id, product_id, expiration_date
FROM batch
WHERE expiration_date < '2025-12-31'
ORDER BY expiration_date;
```

-- chỉ mục cần tạo

```
CREATE INDEX idx_batch_expiration ON batch(expiration_date);
```

--8. Đếm số ngày "Đi làm" (status = 'D') của mỗi nhân viên trong tháng 5/2025.

```
SELECT e.firstname, e.lastname, COUNT(*) AS workdays
FROM working w
JOIN employee e ON w.employee_id = e.employee_id
WHERE w.status = 'D'
AND EXTRACT(MONTH FROM w.work_date) = 5
AND EXTRACT(YEAR FROM w.work_date) = 2025
GROUP BY e.employee_id, e.firstname, e.lastname
ORDER BY workdays DESC;
```

-- chỉ mục cần tạo

```
CREATE INDEX idx_working_employee_date_status ON
working(employee_id, work_date, status);
```

--9. Tính tổng tiền nhập hàng từ mỗi nhà cung cấp.

```
SELECT s.supplier_name, SUM(ir.total_amount) AS total_spent
FROM import_reports ir
JOIN suppliers s ON ir.supplier_id = s.supplier_id
GROUP BY s.supplier_name
ORDER BY total_spent DESC;
```

-- chỉ mục cần tạo

```
CREATE INDEX idx_import_reports_supplier ON import_reports(supplier_id);
```

--10. Xác định ngày nhập gần nhất cho từng sản phẩm.

```

SELECT p.product_name, MAX(b.import_date) AS last_import
FROM batch b
JOIN products p ON b.product_id = p.product_id
GROUP BY p.product_name
ORDER BY last_import DESC;

-- chỉ mục cần tạo

CREATE INDEX idx_batch_product_import_date ON batch(product_id,
import_date);

```

C. NHẬN XÉT – NÂNG CAO:

1. Kết quả tự đánh giá:

a. Kết quả chung:

- Đã biết cách viết bài mô tả nghiệp vụ, phân biệt với mô tả chức năng ứng dụng. Phần này nhóm cũng đã có thêm 1 hoạt động trải nghiệm đó là đi phỏng vấn từ các nhân viên ở siêu thị, nhờ đó am hiểu thêm về nghiệp vụ thực tế mà các chị phải làm mỗi ngày. Đồng thời việc mua 1 hóa đơn của siêu thị cũng là tài liệu quý giá để nhóm tham khảo những gì mà người mua cần quan tâm trên hóa đơn.
- Sơ đồ thực thể liên kết tự thiết kế hoàn toàn.
- Viết 10 câu truy vấn cho mỗi thành viên, có giải thích kèm chú ý về cách đặt chỉ mục nếu có.
- Xây dựng được 1 chương trình kết nối CSDL market.

b. Điểm mới của bài toán:

- Bài toán xây dựng CSDL quản lý siêu thị có sự mở rộng hơn về quản lý HSD, nhờ có sự trợ giúp của giảng viên về ý tưởng lô nên nhóm đã thực hiện bổ sung. Ngoài ra đối với mỗi lô nhập còn có cả biên bản kiểm kê, biên bản xử lý lô hỏng.
- Mở rộng từ phiếu nhập hàng thành biên bản nhập hàng: đây là cụm từ chính xác từ siêu thị để quản lý cụ thể các sản phẩm lô được nhập khi nào.
- Phần quản lý chung có thêm mục Lịch làm việc và Lương thưởng cho nhân viên.

2. Nâng cao – Gợi mở:

- Nếu siêu thị phát triển lâu dài, hoàn toàn có thể thêm phần Khuyến mãi và Điểm thưởng khách hàng vào để có thể phát triển.
- Phần bảng lương còn nhập thủ công, chưa có đủ kịp thời gian để thiết kế nhập tự động cho đúng với tính chất thuộc tính tự suy.

3. Khó khăn:

- Chắc chắn phần khó nhất trong CSDL chính là vẽ được sơ đồ ERD. Sau 1 buổi được cô chữa trên lớp, nhóm cũng đã phải tự về vẽ lại toàn bộ sơ đồ để theo đúng tính chất mô tả nghiệp vụ.
- Phần kết nối CSDL và xây dựng chương trình tương đối nhiều mà bài hướng dẫn kết nối trong chương trình lại ở khá muộn, do đó nhóm phải gấp rút thực hiện nên vẫn còn nhiều chỗ chưa thực sự mượt mà.

4. Phân công nhiệm vụ:

- Nhóm 4 hoàn toàn không có phân công nhiệm vụ, nhóm trưởng đã xác định tinh thần này ngay từ đầu. Các bạn phải lắng nghe nhau trong mọi công việc và tham gia ở mọi khâu của bài toán thiết kế. Ở bước thiết kế (B.1 – B.4) thông thường cứ sau 1 tuần nhóm trưởng sẽ tổ chức buổi họp, đề ra các nhiệm vụ cần làm trong tuần (viết nghiệp vụ, chức năng, vẽ ERD...) và các bạn trong nhóm chủ động báo cáo. Mọi người đều phải tự làm để thực sự hiểu về hệ thống và CSDL mà mình thiết kế để có thể bổ sung và tranh biện với nhau. Ở phần lập trình và khai thác dữ liệu, các thành viên xác định nhiệm vụ là cần đưa market lên máy tính, viết 10 câu truy vấn, xây dựng 1 chương trình. Sẽ có người đảm nhiệm phần giao diện, có người làm phần code và CSDL. Tuy nhiên trong quá trình làm luôn hỏi đáp, bổ sung cho nhau và thậm chí nếu đang làm mà người này bận thì người kia hoàn toàn sẵn sàng làm tiếp các phần còn lại. Do vậy, phần phân công nhiệm vụ không có sự rõ ràng mà làm chung tất cả nên nhóm cũng chưa biết nên ghi thể nào cho đúng, nếu có thì chắc ở phần cuối cùng nhóm trưởng sẽ đảm nhiệm phần báo cáo với slide thuyết trình, nhưng trong thời gian đó thì các bạn cũng đóng vai trò tester kiểm định chương trình Java và bổ sung các chức năng như View, Trigger nếu còn thiếu.

5. Phụ lục:

***Nội dung cập nhật so với bản cũ:**

Bản hiện tại: Bổ sung hoàn chỉnh nội dung theo yêu cầu đề nộp chính thức.

Bản chính thức 9/5:

Cập nhật phiên bản tiếng Anh cho CSDL để bắt đầu code.

Bản 4:

Cập nhật lại thông tin báo cáo và loại bỏ câu chuyện cho báo cáo nghiêm túc.

Bản 3:

- Nghiệp vụ: bổ sung thêm 1 câu chuyện cho dễ tưởng tượng, sau này đến báo cáo chính sẽ xóa. Bổ sung thêm một số nghiệp vụ quan trọng: quản lý lô cho vấn đề hạn sử dụng; quản lý phiếu kiểm kê, biên bản hủy bỏ hàng hóa...
- Chức năng: bổ sung chức năng xem thống kê các sản phẩm sắp hết, các sản phẩm sắp hết HSD, sản phẩm không bán được.

- Cập nhật lại sơ đồ ERD, sơ đồ quan hệ dựa trên quan hệ ngữ nghĩa nghiệp vụ, chưa áp dụng kiến thức phụ thuộc hàm do chưa học.

Bản 2:

- Sửa đổi sơ đồ ERD phần doanh thu.

Bản 1:

- Chỉnh sửa lại chi tiết phần mô tả chức năng theo cấu trúc Input – Output đối với từng chức năng cụ thể.
- Hiện tại nội dung liên quan đến Khuyến mãi bị loại bỏ.
- Xây dựng sơ đồ thực thể liên kết và sơ đồ quan hệ.

Lời kết

Nhóm xin chân thành cảm ơn giảng viên TS. Nguyễn Thị Oanh đã nhiệt tình với lớp mỗi tuần học. Đặc biệt là sau buổi nhận xét trình bày sản phẩm thử, nhờ đó mà các thành viên đã có một bài học đáng nhớ về việc tự thiết kế sản phẩm mà không dựa vào 1 DB mẫu nào có sẵn, phải tự lên ý tưởng và xóa đi vẽ lại sơ đồ từ đầu... Càng thiết kế lại thấy mình càng chưa đủ, những kiến thức ở những buổi lý thuyết còn ít quá. Vậy mà đến buổi cuối cùng, các thành viên cũng tự vẽ được 1 DB “nhà làm” (xin tự nhận là “chỉn chu”) để tự tin đi báo cáo. Học ở giảng viên, nhóm cũng học được nhiều bài học về cách tổ chức công việc, sự nghiêm túc trong bài giảng và lắng nghe bài trình bày của sinh viên. Nhóm chúc giảng viên luôn nhiều sức khỏe, đạt được nhiều thành công hơn nữa trong sự nghiệp trồng “trẻ lớn”.

Nhóm chúc cô Oanh một Hè vui!

Nhóm 4_156778_IT3290