

TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN

----- * * * -----

BÁO CÁO ĐỒ ÁN

Chuyên đề Python (COS525)



**XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỦA HÀNG TIVI
VỚI PYTHON, TKINTER VÀ MYSQL**

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện:

DTH235719 - Lê Thị Huỳnh Như - DH24TH2_Nhóm2_Tô 1

DTH235724 - Phạm Thị Trúc Như - DH24TH2_Nhóm2_Tô 1

AN GIANG, 11-2025

MỤC LỤC

I. GIỚI THIỆU ĐỀ TÀI	1
1. Tính cần thiết của đề tài	1
2.Mục tiêu nghiên cứu	1
II. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT	2
1. Tổng quan về bài toán quản lý cửa hàng	2
2.Cơ sở lý thuyết	2
2.1. Ngôn ngữ lập trình Python	2
2.2. Thư viện Tkinter	2
2.3. Cơ sở dữ liệu MySQL	3
3. Phạm vi và đối tượng nghiên cứu	3
4. Phương pháp nghiên cứu và ứng dụng lý thuyết	3
III.KẾT QUẢ ĐẠT ĐƯỢC VÀ CODE	4
1. Giới thiệu chung về chức năng	4
2. Giao diện và chức năng	5
2.1. Giao diện bảng quản lý hệ thống chính	5
2.2. Bảng quản lý sản phẩm	6
2.3. Bảng quản lý kho hàng	7
2.4. Bảng quản lý hóa đơn và chi tiết hóa đơn	8
2.5. Bảng quản lý khách hàng	9
2.6. Bảng quản lý nhân viên	10
3. Chi tiết code	11
4. Cơ sở dữ liệu MySQL	42
TÀI LIỆU THAM KHẢO	42

LỜI NÓI ĐẦU

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các phần mềm vào hoạt động quản lý kinh doanh đã trở thành xu hướng tất yếu nhằm nâng cao hiệu quả làm việc, giảm thiểu sai sót và tối ưu hóa quy trình vận hành. Đặc biệt, đối với các cửa hàng kinh doanh thiết bị điện tử như tivi, khói lượng thông tin về sản phẩm, khách hàng, nhân viên và hóa đơn ngày càng nhiều, nếu quản lý thủ công sẽ dễ dẫn đến chồng chéo dữ liệu, sai sót trong tính toán hoặc mất nhiều thời gian tra cứu.

Xuất phát từ nhu cầu thực tế đó, nhóm chúng em đã thực hiện đề tài “Xây dựng ứng dụng quản lý cửa hàng Tivi với Python, Tkinter và MySQL” nhằm tạo ra một phần mềm hỗ trợ quản lý thông minh, dễ sử dụng và phù hợp với mô hình cửa hàng nhỏ và vừa. Ứng dụng được xây dựng dựa trên kiến thức đã học về lập trình Python, thiết kế giao diện, xử lý cơ sở dữ liệu cũng như các kỹ năng phân tích yêu cầu hệ thống.

Trong quá trình thực hiện đề tài, chúng em đã cố gắng tìm hiểu tài liệu, phân tích yêu cầu, xây dựng cơ sở dữ liệu, thiết kế giao diện và triển khai các chức năng quản lý sản phẩm, kho hàng, khách hàng, hóa đơn và nhân viên. Mặc dù đã nỗ lực hoàn thiện đề tài, nhưng khó tránh khỏi những thiếu sót do hạn chế về thời gian, kiến thức và kinh nghiệm thực tế. Nhóm rất mong nhận được sự góp ý từ thầy để đề tài được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn ThS. Nguyễn Ngọc Minh*, giảng viên hướng dẫn, đã tận tình hỗ trợ, định hướng và giúp đỡ chúng em trong suốt quá trình thực hiện đồ án.

Xin trân trọng cảm ơn thầy!

I. GIỚI THIỆU ĐỀ TÀI

1. Tính cần thiết của đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, các hoạt động quản lý kinh doanh đang được tự động hóa nhằm nâng cao hiệu quả làm việc và giảm thiểu sai sót. Đối với các cửa hàng kinh doanh mặt hàng tivi, việc quản lý sản phẩm, số lượng tồn kho, thông tin khách hàng, nhân viên hay hóa đơn bán hàng là vô cùng quan trọng. Tuy nhiên, nhiều cửa hàng nhỏ và vừa vẫn áp dụng phương pháp quản lý thủ công bằng sổ sách hoặc Excel, dẫn đến các vấn đề:

- Dễ xảy ra nhầm lẫn khi nhập liệu hoặc tính toán.
- Khó tìm kiếm và truy xuất thông tin khi dữ liệu lớn.
- Không đồng bộ và bảo mật dữ liệu thấp.
- Không hỗ trợ thống kê, báo cáo nhanh.
- Khó mở rộng hoặc chia sẻ thông tin khi cửa hàng phát triển.

Vì vậy, việc xây dựng phần mềm quản lý cửa hàng Tivi bằng Python, Tkinter và MySQL là rất cần thiết. Ứng dụng không chỉ giúp chủ cửa hàng quản lý dữ liệu khoa học, chính xác và nhanh chóng, mà còn giúp sinh viên ứng dụng kiến thức về lập trình, giao diện và cơ sở dữ liệu vào giải quyết một bài toán thực tế – hướng tới năng lực làm việc chuyên nghiệp trong tương lai.

2. Mục tiêu nghiên cứu

Đề tài hướng đến việc xây dựng một ứng dụng hoàn chỉnh với các mục tiêu cụ thể sau:

- Xây dựng giao diện quản lý trực quan, thân thiện, giúp người dùng dễ dàng thao tác dù không am hiểu công nghệ.
- Quản lý dữ liệu cửa hàng tivi bao gồm các chức năng:
 - Thêm – sửa – xóa và tìm kiếm thông tin sản phẩm Tivi.
 - Quản lý khách hàng.
 - Quản lý hóa đơn bán hàng.
 - Quản lý nhân viên .
 - Quản lý kho
- Tìm kiếm và lọc sản phẩm nhanh, hỗ trợ quá trình bán hàng.
- Kết nối và xử lý dữ liệu với MySQL, đảm bảo tính ổn định và an toàn dữ liệu.
- Áp dụng kiến thức đã học: lập trình Python, lập trình hướng đối tượng, Tkinter, thiết kế cơ sở dữ liệu quan hệ, và xử lý SQL.

II. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

1. Tổng quan về bài toán quản lý cửa hàng

Trong hoạt động kinh doanh, đặc biệt là kinh doanh thiết bị điện tử như tivi, một cửa hàng cần quản lý đồng bộ các thông tin như:

- Danh mục sản phẩm (mã tivi, hãng sản xuất, kích thước, giá bán...)
- Số lượng tồn kho.
- Thông tin khách hàng.
- Thông tin nhân viên.
- Hóa đơn và lịch sử giao dịch.

Nếu quản lý thủ công bằng giấy tờ hoặc Excel, cửa hàng thường gặp các vấn đề:

- Dữ liệu khó cập nhật, dễ trùng lặp.
- Không thể kiểm soát tồn kho theo thời gian thực.
- Dễ thất lạc khi lưu trữ lâu dài.
- Khó tạo báo cáo bán hàng.
- Không hỗ trợ đa người dùng.

Chính vì những hạn chế đó, việc ứng dụng phần mềm quản lý vào cửa hàng là xu hướng tất yếu để tăng hiệu suất làm việc, giảm chi phí vận hành và đáp ứng nhu cầu mở rộng.

2.Cơ sở lý thuyết

2.1. Ngôn ngữ lập trình Python

Python là ngôn ngữ bậc cao, cú pháp đơn giản, dễ đọc và được sử dụng nhiều trong phát triển phần mềm. Python có hệ sinh thái phong phú với nhiều thư viện hỗ trợ, đặc biệt phù hợp cho các ứng dụng quản lý.

2.2. Thư viện Tkinter

Tkinter là thư viện GUI mặc định của Python, hỗ trợ xây dựng:

- Cửa sổ giao diện
- Button, Label, Entry
- Combobox, Treeview
- Hộp thoại thông báo (messagebox)
- Menu, Frame,...

- Ưu điểm: nhẹ, dễ dùng, phù hợp cho ứng dụng quản lý nhỏ và vừa.

2.3. Cơ sở dữ liệu MySQL

MySQL là hệ quản trị CSDL quan hệ mạnh mẽ, miễn phí, tốc độ xử lý cao.

Các điểm cần nắm:

- Mỗi bảng gồm các hàng và cột.
- Khóa chính (Primary Key) đảm bảo mỗi bản ghi là duy nhất.
- Khóa ngoại (Foreign Key) tạo mối quan hệ giữa các bảng.
- Các câu lệnh SQL cơ bản: SELECT, INSERT, UPDATE, DELETE.

3. Phạm vi và đối tượng nghiên cứu

- Đối tượng sử dụng: nhân viên bán hàng, quản lý hoặc chủ cửa hàng tivi.
- Phạm vi chức năng của phần mềm:
 - Bảng quản lý sản phẩm Tivi
 - Bảng quản lý khách hàng
 - Bảng quản lý hóa đơn và chi tiết hóa đơn
 - Bảng quản lý nhân viên
 - Bảng quản lý kho
 - Tìm kiếm sản phẩm.
 - Lưu trữ dữ liệu vào MySQL.

4. Phương pháp nghiên cứu và ứng dụng lý thuyết

Để thực hiện đề tài, các phương pháp sau được áp dụng:

- Nghiên cứu lý thuyết: tìm hiểu kiến thức về Python, Tkinter, SQL và mô hình CSDL.
- Phân tích yêu cầu hệ thống: xác định các chức năng chính, luồng xử lý.
- Thiết kế CSDL quan hệ: xây dựng mô hình ERD, xác định các bảng:
 - Bảng quản lý sản phẩm Tivi
 - Bảng quản lý khách hàng
 - Bảng quản lý hóa đơn và chi tiết hóa đơn
 - Bảng quản lý nhân viên
 - Bảng quản lý kho
- Thiết kế giao diện: xây dựng bộ cục các cửa sổ, nút bấm, thanh menu, bảng dữ liệu.
- Lập trình chức năng: thêm – sửa – xóa – tìm kiếm – thống kê.

III.KẾT QUẢ ĐẠT ĐƯỢC VÀ CODE

1. Giới thiệu chung về chức năng

Khi cửa hàng nhập sản phẩm mới hoặc có khách đến mua Tivi, chương trình quản lý cửa hàng Tivi sẽ giúp người dùng lưu trữ đầy đủ thông tin liên quan đến sản phẩm và hoạt động bán hàng. Những thông tin bao gồm: mã sản phẩm, tên sản phẩm, hãng sản xuất, kích thước, loại, giá nhập, giá bán, số lượng tồn kho, tình trạng và mô tả của từng loại Tivi. Bên cạnh đó còn có tìm kiếm sản phẩm, chức năng tìm kiếm sản phẩm giúp người dùng nhanh chóng tra cứu sản phẩm Tivi trong kho dựa trên tên sản phẩm. Khi người dùng nhập từ khóa, hệ thống sẽ lọc danh sách và chỉ hiển thị những sản phẩm phù hợp. Nhờ đó, việc quản lý kho trở nên thuận tiện, tiết kiệm thời gian và tránh nhầm lẫn khi số lượng sản phẩm lớn.

Đồng thời chương trình giúp theo dõi tình trạng kho của cửa hàng, cho phép người dùng kiểm tra số lượng tồn của từng sản phẩm, sản phẩm nào còn hàng, sản phẩm nào sắp hết, sản phẩm nào đã ngừng kinh doanh. Hệ thống cũng hỗ trợ xem danh sách toàn bộ sản phẩm trong kho, phân loại theo loại phiếu nhập hoặc xuất, ngày lập, nhà cung cấp, mã sản phẩm, đơn giá và ghi chú của từng loại Tivi

Khi người sử dụng cần quản lý hóa đơn bán hàng và chiết khấu hóa đơn bán hàng, chương trình sẽ giúp lưu trữ từng hóa đơn của khách, hiển thị chi tiết thông tin của khách hàng và chi tiết bán hàng gồm danh sách hóa đơn và chiết khấu hóa đơn. Danh sách hóa đơn bao gồm mã hóa đơn, ngày bán hàng, tên khách hàng, tổng tiền mà khách phải trả, nhân viên bán hàng, hình thức thanh toán và ghi chú cần thiết. Chiết khấu hóa đơn bao gồm mã hóa đơn, mã sản phẩm, tên sản phẩm, số lượng, đơn giá và thành tiền.

Bên cạnh đó, chương trình còn có chức năng quản lý thông tin nhân viên: hệ thống lưu trữ thông tin như mã nhân viên, họ tên, vị trí công việc, lương, thưởng, ca làm và hiệu suất làm việc. Chức năng này giúp theo dõi công việc của từng nhân viên, đánh giá hiệu quả và phân công công việc hợp lý.

Và cuối cùng là quản lý khách hàng hệ thống lưu trữ mã khách hàng, họ tên, số điện thoại, địa chỉ, lịch sử mua hàng và các chính sách ưu đãi. Chức năng này giúp theo dõi nhu cầu khách hàng, chăm sóc khách hàng sau bán và áp dụng chính sách phù hợp để tăng sự hài lòng.

Tóm lại, chương trình sẽ có 5 chức năng chính:

- Quản lý sản phẩm Tivi
- Quản lý kho hàng
- Quản lý hóa đơn và chi tiết hóa đơn
- Quản lý khách hàng
- Quản lý nhân viên

2. Giao diện và chức năng

2.1. Giao diện bảng quản lý hệ thống chính



Hình 1: Bảng quản lý hệ thống chính

Đây là giao diện chính của hệ thống quản lý cửa hàng Tivi. Màn hình hiển thị các chức năng quang trọng nhất của phần mềm bao gồm **Quản lý sản phẩm**, **Quản lý kho hàng**, **Quản lý hóa đơn**, **Quản lý khách hàng**, **Quản lý nhân viên** và giao diện còn có nút **Thoát** cho phép người dùng đóng khi không sử dụng tiếp tục.

2.2. Bảng quản lý sản phẩm

masanpham	tensanpham	hangsanxuat	kichthuoc	loai	gianhap	giaban	soluongton	tinhtrang	mota
1	Sony Bravia 55X	Sony	55	Smart TV 4K	12000000.00	18000000.00	20	Mới	Màn hình 55 inch, i
2	Samsung QLED 65C	Samsung	65	QLED 4K	20000000.00	25000000.00	9	Mới	TV QLED 65 inch 4k

Tên sản phẩm: _____ Hàng SX: _____ Kích thước: _____
Loại: _____ Giá nhập: _____ Giá bán: _____
Số lượng: _____ Tình trạng: Mới _____ Mô tả: _____
Tim kiếm: _____ Tim Hiển thị tất cả _____
Thêm Lưu Sửa Hủy Xóa Thoát

Hình 2: Giao diện bảng quản lý sản phẩm TiVi

- Bảng hiển thị tất cả thông tin của các ti vi trong kho bao gồm:
 - **masanpham:** Mã sản phẩm.
 - **tensanpham:** Tên sản phẩm.
 - **hangsanxuat:** Hãng sản xuất.
 - **kichthuoc:** Kích thước màn hình.
 - **loai:** Loại TV (Smart TV 4K, QLED 4K,...).
 - **gianhap:** Giá nhập của sản phẩm.
 - **giaban:** Giá bán ra.
 - **soluongton:** Số lượng tồn kho.
 - **tinhtrang:** Tình trạng (Mới, Cũ,...).
 - **mota:** Mô tả chi tiết sản phẩm.
- Chức năng: Quan sát nhanh, chọn sản phẩm để sửa, xóa, hoặc xem chi tiết.
- Các ô phía dưới dùng để **thêm hoặc sửa thông tin sản phẩm:**
 - **Thêm:** Nhập thông tin vào các ô và thêm sản phẩm mới vào bảng.
 - **Lưu:** Lưu dữ liệu vừa thêm hoặc sửa vào danh sách/kho.
 - **Sửa:** Chọn một sản phẩm từ bảng, chỉnh sửa thông tin trong khung nhập liệu.
 - **Hủy:** Hủy thao tác thêm hoặc sửa, khôi phục dữ liệu cũ.
 - **Xóa:** Xóa sản phẩm đã chọn khỏi danh sách.
 - **Thoát:** Đóng phần mềm

2.3. Bảng quản lý kho hàng

maphieu	loaiphieu	ngaylap	nhacungcap	masanpham	soluong	dongia	ghichu
1	nhập	2025-10-21	Công ty Samsung	2	10	20000000.00	Samsung QLED56C
5	nhập	2025-11-11	Sony	1	10	12000000.00	Sony Bravia 55X
6	nhập	2025-11-14	Samsung	2	5	20000000.00	khong co

Hình 3: Giao diện bảng quản lý kho hàng TiVi

- Bảng dữ liệu kho gồm các thông tin:
 - **maphieu:** Mã phiếu nhập hoặc xuất kho.
 - **loaiphieu:** Loại phiếu (Nhập hoặc Xuất).
 - **ngaylap:** Ngày lập phiếu.
 - **nhacungcap:** Tên nhà cung cấp.
 - **masanpham:** Mã sản phẩm được nhập hoặc xuất.
 - **soluong:** Số lượng sản phẩm nhập hoặc xuất .
 - **dongia:** Đơn giá nhập hoặc xuất.
 - **ghichu:** Thông tin ghi chú.
- Chức năng
 - Quan sát nhanh danh sách phiếu nhập kho.
 - Chọn phiếu để xem chi tiết, sửa hoặc xóa.
 - Theo dõi lịch sử nhập, xuất hàng.
- Các thao tác bên dưới dùng để quản lý kho
 - **Thêm:** Nhập dữ liệu phiếu và thêm phiếu mới vào bảng.
 - **Lưu:** Lưu thông tin vừa thêm hoặc sửa.
 - **Sửa:** Chọn phiếu để chỉnh sửa nội dung.
 - **Xóa:** Xóa phiếu đang chọn khỏi danh sách.
 - **Tải lại:** Làm mới bảng dữ liệu.
 - **Đóng:** Thoát giao diện quản lý kho.

2.4. Bảng quản lý hóa đơn và chi tiết hóa đơn

Mahoadon	Ngayban	Tenkhachhang	Tongtien	Nhanvienbanhang	Hinhhuctt	Ghichu
10	2025-10-31	Lê Thị Huỳnh Như	18000000.00	Trúc Như	tiền mặt	Sinh Viên giảm 5%
12	2025-11-11	Dương Nguyễn Ngọc Linh	18000000.00	Truc Nhu	tiền mặt	

Thông tin hóa đơn:

Ngày (YYYY-MM-DD): 2025-11-14 Khách hàng: _____
Nhân viên bán: _____ Hình thức TT: tiền mặt
Ghi chú: _____

Thêm sản phẩm vào hóa đơn:

Mã hóa đơn: _____ Mã SP: _____
Tên SP: _____ Số lượng: _____
Đơn giá: _____

Buttons:

- Tạo hóa đơn
- Sửa hóa đơn
- Thêm chi tiết
- Xóa hóa đơn
- Lưu
- Đóng

Hình 4: Giao diện bảng quản lý hóa đơn và chi tiết hóa đơn

- Bảng danh sách hóa đơn gồm các thông tin:
 - **Mahoadon:** mã hóa đơn.
 - **Ngayban:** ngày lập hóa đơn.
 - **Tenkhachhang:** tên khách mua hàng.
 - **Tongtien:** tổng tiền hóa đơn.
 - **Nhanvienbanhang:** nhân viên thực hiện bán hàng.
 - **Hinhhuctt:** hình thức thanh toán.
 - **Ghichu:** ghi chú hóa đơn.
- Chi tiết hóa đơn bao gồm:
 - **Id:** mã chi tiết.
 - **Masanpham:** mã sản phẩm.
 - **Tensanpham:** tên sản phẩm.
 - **Soluong:** số lượng sản phẩm.
 - **Dongia:** đơn giá.
 - **Thanhtien:** thành tiền.
- Chức năng:
 - Xem danh sách hóa đơn.
 - Xem chi tiết hóa đơn.
 - Sửa hoặc xóa hóa đơn.
 - Thêm sản phẩm vào hóa đơn.

- Các thao tác:

- **Tạo hóa đơn:** Lập hóa đơn mới.
- **Sửa hóa đơn:** Chính sửa nội dung hóa đơn.
- **Thêm chi tiết:** Thêm sản phẩm vào hóa đơn.
- **Xóa hóa đơn:** Xóa hóa đơn đang chọn.
- **Lưu:** Lưu toàn bộ thay đổi.
- **Đóng:** Thoát giao diện.

2.5. Bảng quản lý khách hàng

The screenshot shows a Windows application window titled "Quản lý khách hàng". The main area displays a table with the following columns: makhachhang, hoten, sodienthoai, diachi, lichsumuahang, and chinh sach. There are two rows of data:

makhachhang	hoten	sodienthoai	diachi	lichsumuahang	chinh sach
2	Huỳnh Như	1243573873	An Giang	1	Sinh viên giảm 5%
7	Dương Nguyễn Ngọc Linh	0243573831	An Giang	1	Sinh viên giảm 5%

At the bottom of the window, there is a search/filter bar with fields for Họ tên, SĐT, Địa chỉ, and Lịch sử mua. Below the search bar are buttons for Thêm, Cập nhật, Lưu, Xóa, and Đóng.

Hình 5: Giao diện bảng quản lý thông tin khách hàng

- Bảng dữ liệu quản lý khách hàng gồm các thông tin:

- **makhachhang:** Mã khách hàng.
- **hoten:** Họ tên khách hàng.
- **sodienthoai:** Số điện thoại liên hệ.
- **diachi:** Địa chỉ cư trú.
- **lichsumuahang:** Số lần mua hàng.
- **chinh sach:** Chính sách áp dụng.

- Chức năng
 - Quan sát danh sách khách hàng.
 - Chọn khách để xem chi tiết, sửa hoặc xóa.
 - Theo dõi lịch sử mua hàng.
- Các thao tác dùng để quản lý
 - **Thêm:** Nhập dữ liệu và thêm khách hàng.
 - **Lưu:** Lưu thông tin vừa thêm hoặc sửa.
 - **Sửa:** Chọn khách để chỉnh sửa.
 - **Xóa:** Xóa khách khỏi danh sách.
 - **Tải lại:** Làm mới bảng dữ liệu.
 - **Đóng:** Thoát giao diện.

2.6. Bảng quản lý nhân viên

Quản lý nhân viên

manhanvien	hoten	vitri	luong	thuong	calam	hieusuat
1	Trúc Như	nhân viên	8000000.00	1000000.00	8h_21h	7.00

Họ tên: Vị trí: Lương:
 Thường: Ca làm: Hiệu suất:

Bảng 6: Giao diện bảng quản lý nhân viên của cửa hàng

- Bảng dữ liệu quản lý nhân viên gồm các thông tin:
 - **manhanvien:** Mã nhân viên.
 - **hoten:** Họ tên nhân viên.
 - **vitri:** Chức vụ.
 - **luong:** Lương cơ bản.
 - **thuong:** Thưởng.
 - **calam:** Ca làm việc.

- **hieusuat:** Điểm đánh giá hiệu suất.
- Chức năng
 - Quan sát danh sách nhân viên.
 - Chọn nhân viên để xem chi tiết hoặc sửa.
 - Theo dõi ca làm và hiệu suất làm việc.
- Các thao tác dùng để quản lý
 - **Thêm:** Nhập dữ liệu nhân viên mới.
 - **Lưu:** Lưu thông tin đã chỉnh sửa.
 - **Sửa:** Điều chỉnh dữ liệu nhân viên.
 - **Xóa:** Xóa nhân viên khỏi danh sách.
 - **Tải lại:** Làm mới bảng nhân viên.
 - **Đóng:** Thoát giao diện.

3. Chi tiết code

```

import tkinter as tk
from tkinter import ttk, messagebox
import mysql.connector
from datetime import datetime

# -----
# Cấu hình DB - chỉnh ở đây
# -----
DB_CONFIG = {
    "host": "localhost",
    "user": "root",
    "password": "15082005",      # sửa mật khẩu nếu cần
    "database": "qlcuahangTivi"
}

# -----
# Hàm kết nối
# -----
def connect_db():
    try:

```

```

conn = mysql.connector.connect(**DB_CONFIG)

return conn

except mysql.connector.Error as e:
    messagebox.showerror("Lỗi kết nối MySQL", str(e))
    return None

# -----
# UTIL: Hiển thị lỗi / commit
# -----

def db_commit(conn):
    try:
        conn.commit()
    except Exception as e:
        messagebox.showerror("Lỗi commit", str(e))

# -----
# QUẢN LÝ SẢN PHẨM
# -----


def open_product_manager():
    conn = connect_db()

    if not conn:
        return

    cursor = conn.cursor()

    win = tk.Tk()
    win.title("Quản lý sản phẩm Tivi")
    win.geometry("1100x650")

    cols = ("masanpham", "tensanpham", "hangsanxuat", "kichthuoc", "loai",
            "gianhap", "giaban", "soluongton", "tinhtrang", "mota")
    tree = ttk.Treeview(win, columns=cols, show="headings")

    for c in cols:
        tree.heading(c, text=c)
        tree.column(c, width=100, anchor="center")

```

```

tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)

# --- Form nhập ---
frm = tk.Frame(win)
frm.pack(padx=10, pady=5, fill="x")

labels = ["Tên sản phẩm", "Hãng SX", "Kích thước", "Loại",
          "Giá nhập", "Giá bán", "Số lượng", "Tình trạng", "Mô tả"]
entries = {}

for i, lbl in enumerate(labels):
    tk.Label(frm, text=lbl).grid(row=i//3, column=(i%3)*2, sticky="e", padx=5, pady=5)
    if lbl == "Mô tả":
        entries[lbl] = tk.Text(frm, height=3, width=40)
        entries[lbl].grid(row=i//3, column=(i%3)*2+1, padx=5, pady=5)
    elif lbl == "Tình trạng":
        var = tk.StringVar(value="Mới")
        cb = ttk.Combobox(frm, textvariable=var, values=["Mới", "Trung bày", "Đã qua sử dụng"])
        cb.grid(row=i//3, column=(i%3)*2+1, padx=5, pady=5)
        entries[lbl] = var
    else:
        e = tk.Entry(frm)
        e.grid(row=i//3, column=(i%3)*2+1, padx=5, pady=5)
        entries[lbl] = e

# --- Biến trạng thái ---
edit_mode = {"status": None, "id": None}

# --- Load dữ liệu ---
def load_products():
    tree.delete(*tree.get_children())
    cursor.execute("SELECT * FROM sanpham")
    for r in cursor.fetchall():
        tree.insert("", tk.END, values=r)

```

--- Làm sạch form ---

```
def clear_form():
    for e in entries.values():
        e.delete(0, tk.END)
    edit_mode["status"] = None
    edit_mode["id"] = None
```

--- Khi chọn dòng ---

```
def on_select(ev):
    sel = tree.focus()
    if not sel:
        return
    v = tree.item(sel)["values"]
    edit_mode["id"] = v[0]
    entries["Tên sản phẩm"].delete(0, tk.END); entries["Tên sản phẩm"].insert(0, v[1])
    entries["Hãng SX"].delete(0, tk.END); entries["Hãng SX"].insert(0, v[2])
    entries["Kích thước"].delete(0, tk.END); entries["Kích thước"].insert(0, v[3])
    entries["Loại"].delete(0, tk.END); entries["Loại"].insert(0, v[4])
    entries["Giá nhập"].delete(0, tk.END); entries["Giá nhập"].insert(0, v[5])
    entries["Giá bán"].delete(0, tk.END); entries["Giá bán"].insert(0, v[6])
    entries["Số lượng"].delete(0, tk.END); entries["Số lượng"].insert(0, v[7])
    entries["Tình trạng"].delete(0, tk.END); entries["Tình trạng"].insert(0, v[8])
    entries["Mô tả"].delete(0, tk.END); entries["Mô tả"].insert(0, v[9])
```

--- Thêm mới ---

```
def add_product():
    clear_form()
    edit_mode["status"] = "add"
    messagebox.showinfo("Thêm sản phẩm", "Nhập thông tin và bấm Lưu để thêm mới.")
```

--- Lưu ---

```
def save_product():
    data = (
        entries["Tên sản phẩm"].get(),
```

```

        entries["Hãng SX"].get(),
        entries["Kích thước"].get(),
        entries["Loại"].get(),
        float(entries["Giá nhập"].get() or 0),
        float(entries["Giá bán"].get() or 0),
        int(entries["Số lượng"].get() or 0),
        entries["Tình trạng"].get() or "mới",
        entries["Mô tả"].get()
    )
}

try:
    if edit_mode["status"] == "add":
        cursor.execute("""
            INSERT INTO sanpham (tensanpham, hangsanxuat, kichthuoc, loai,
            gianhap, giaban, soluongton, tinhtrang, mota)
            VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s)
        """, data)
        db_commit(conn)
        messagebox.showinfo("Thành công", "Đã thêm sản phẩm mới.")
    elif edit_mode["status"] == "edit" and edit_mode["id"]:
        cursor.execute("""
            UPDATE sanpham SET tensanpham=%s, hangsanxuat=%s, kichthuoc=%s, loai=%s,
            gianhap=%s, giaban=%s, soluongton=%s, tinhtrang=%s, mota=%s
            WHERE masanpham=%s
        """, data + (edit_mode["id"],))
        db_commit(conn)
        messagebox.showinfo("Thành công", "Đã cập nhật sản phẩm.")
    else:
        messagebox.showwarning("Chú ý", "Vui lòng chọn Thêm hoặc Sửa trước khi Lưu.")
        load_products()
        clear_form()
except Exception as e:
    messagebox.showerror("Lỗi lưu", str(e))

```

--- Sửa ---

```
def edit_product():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Vui lòng chọn sản phẩm cần sửa.")
        return
    edit_mode["status"] = "edit"
    messagebox.showinfo("Chế độ sửa", "Bạn có thể chỉnh thông tin và bấm Lưu.")
```

--- Hủy ---

```
def cancel_edit():
    clear_form()
    messagebox.showinfo("Hủy", "Đã hủy thao tác.")
```

--- Xóa ---

```
def delete_product():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Vui lòng chọn sản phẩm cần xóa.")
        return
    v = tree.item(sel)["values"]
    if messagebox.askyesno("Xác nhận", f"Bạn chắc chắn muốn xóa sản phẩm '{v[1]}'?"):
        cursor.execute("DELETE FROM sanpham WHERE masanpham=%s", (v[0],))
        db_commit(conn)
        load_products()
        clear_form()
        messagebox.showinfo("Đã xóa", "Sản phẩm đã bị xóa.")
```

--- Thanh tìm kiếm ---

```
search_frame = tk.Frame(win)
search_frame.pack(pady=5)
tk.Label(search_frame, text="Tìm kiếm:").pack(side=tk.LEFT, padx=5)
search_entry = tk.Entry(search_frame, width=40)
search_entry.pack(side=tk.LEFT, padx=5)
```

```

def search_product():

    key = search_entry.get().strip()

    if not key:
        messagebox.showwarning("Tìm kiếm", "Vui lòng nhập từ khóa.")

        return

    tree.delete(*tree.get_children())

    cursor.execute("""
        SELECT * FROM sanpham
        WHERE tensanpham LIKE %s OR hangsanxuat LIKE %
        """ , (f"%{key}%", f"%{key}%"))

    for r in cursor.fetchall():
        tree.insert("", tk.END, values=r)

def show_all():

    search_entry.delete(0, tk.END)

    load_products()

tk.Button(search_frame, text="Tìm", command=search_product).pack(side=tk.LEFT, padx=5)
tk.Button(search_frame, text="Hiển thị tất cả", command=show_all).pack(side=tk.LEFT, padx=5)

# --- Thoát ---

def exit_window():

    cursor.close()

    conn.close()

    win.destroy()

# --- Nút chức năng ---

btnf = tk.Frame(win)

btnf.pack(pady=10)

tk.Button(btnf, text="Thêm", width=12, command=add_product).grid(row=0, column=0, padx=5)
tk.Button(btnf, text="Lưu", width=12, command=save_product).grid(row=0, column=1, padx=5)
tk.Button(btnf, text="Sửa", width=12, command=edit_product).grid(row=0, column=2, padx=5)
tk.Button(btnf, text="Hủy", width=12, command=cancel_edit).grid(row=0, column=3, padx=5)
tk.Button(btnf, text="Xóa", width=12, command=delete_product).grid(row=0, column=4, padx=5)

```

```

tk.Button(btnf, text="Thoát", width=12, command=exit_window).grid(row=0, column=5, padx=5)

tree.bind("<<TreeviewSelect>>", on_select)
load_products()
win.mainloop()

# -----
# QUAN LI KHO HANG
# -----


def open_stock_manager():
    conn = connect_db()
    if not conn:
        return
    cursor = conn.cursor()

    win = tk.Tk()
    win.title("Quản lý kho hàng")
    win.geometry("950x600")

    cols = ("maphieu", "loaiphieu", "ngaylap", "nhacungcap", "masanpham", "soluong", "dongia", "ghichu")
    tree = ttk.Treeview(win, columns=cols, show="headings")
    for c in cols:
        tree.heading(c, text=c)
        tree.column(c, width=110, anchor="center")
    tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)

# --- Form nhập ---
frm = tk.Frame(win); frm.pack(padx=10, pady=5, fill="x")
tk.Label(frm, text="Loại phiếu").grid(row=0, column=0)
loaivar = tk.StringVar(value="Nhập")
cb_loai = ttk.Combobox(frm, textvariable=loaivar, values=["Nhập", "Xuất"], width=15)
cb_loai.grid(row=0, column=1)

```

```

tk.Label(frm, text="Ngày (YYYY-MM-DD)").grid(row=0, column=2)
ngay_entry = tk.Entry(frm, width=20); ngay_entry.grid(row=0, column=3)
ngay_entry.insert(0, datetime.now().strftime("%Y-%m-%d"))

tk.Label(frm, text="Nhà cung cấp").grid(row=1, column=0)
nhacungcap_entry = tk.Entry(frm, width=25); nhacungcap_entry.grid(row=1, column=1)

tk.Label(frm, text="Mã sản phẩm").grid(row=1, column=2)
masp_entry = tk.Entry(frm, width=20); masp_entry.grid(row=1, column=3)

tk.Label(frm, text="Số lượng").grid(row=2, column=0)
soluong_entry = tk.Entry(frm, width=20); soluong_entry.grid(row=2, column=1)

tk.Label(frm, text="Đơn giá").grid(row=2, column=2)
dongia_entry = tk.Entry(frm, width=20); dongia_entry.grid(row=2, column=3)

tk.Label(frm, text="Ghi chú").grid(row=3, column=0)
ghichu_entry = tk.Entry(frm, width=70); ghichu_entry.grid(row=3, column=1, columnspan=3,
pady=5)

# Biến trạng thái
edit_mode = {"status": None, "id": None}

# --- Load dữ liệu ---

def load_stock():
    tree.delete(*tree.get_children())
    try:
        cursor.execute("SELECT
maphieu,loaiphieu,ngaylap,nhacungcap,masanpham,soluong,dongia,ghichu FROM khohang")
        for r in cursor.fetchall():
            tree.insert("", tk.END, values=r)
    except Exception as e:
        messagebox.showerror("Lỗi", str(e))

```

```
# --- Làm sạch form ---
```

```
def clear_form():
    loaivar.set("Nhập")
    ngay_entry.delete(0, tk.END); ngay_entry.insert(0, datetime.now().strftime("%Y-%m-%d"))
    nhacungcap_entry.delete(0, tk.END)
    masp_entry.delete(0, tk.END)
    soluong_entry.delete(0, tk.END)
    dongia_entry.delete(0, tk.END)
    ghichu_entry.delete(0, tk.END)
    edit_mode["status"] = None
    edit_mode["id"] = None
```

```
# --- Khi chọn dòng ---
```

```
def on_select(ev):
    sel = tree.focus()
    if not sel: return
    v = tree.item(sel)["values"]
    edit_mode["id"] = v[0]
    loaivar.set(v[1])
    ngay_entry.delete(0, tk.END); ngay_entry.insert(0, v[2])
    nhacungcap_entry.delete(0, tk.END); nhacungcap_entry.insert(0, v[3])
    masp_entry.delete(0, tk.END); masp_entry.insert(0, v[4])
    soluong_entry.delete(0, tk.END); soluong_entry.insert(0, v[5])
    dongia_entry.delete(0, tk.END); dongia_entry.insert(0, v[6])
    ghichu_entry.delete(0, tk.END); ghichu_entry.insert(0, v[7])
```

```
# --- Thêm phiếu ---
```

```
def add_phieu():
    try:
        sql = """INSERT INTO khohang (loaiphieu, ngaylap, nhacungcap, masanpham, soluong,
dongia, ghichu)
VALUES (%s,%s,%s,%s,%s,%s)"""
        vals = (
            edit_mode["id"], date, id, name, quantity, price, note)
```

```

        loaivar.get(),
        ngay_entry.get(),
        nhacungcap_entry.get(),
        int(masp_entry.get()),
        int(soluong_entry.get()),
        float(dongia_entry.get() or 0),
        ghichu_entry.get()
    )
cursor.execute(sql, vals)
db_commit(conn)
load_stock()
clear_form()
messagebox.showinfo("Thành công", "Đã ghi nhận phiếu mới.")

except Exception as e:
    messagebox.showerror("Lỗi thêm phiếu", str(e))

```

--- Sửa phiếu ---

```

def edit_phieu():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Vui lòng chọn phiếu cần sửa.")
        return
    edit_mode["status"] = "edit"
    messagebox.showinfo("Chế độ sửa", "Hãy chỉnh thông tin và bấm 'Lưu thay đổi!'")

```

--- Lưu thay đổi ---

```

def save_edit():
    if edit_mode["status"] != "edit" or not edit_mode["id"]:
        messagebox.showwarning("Chú ý", "Bạn cần chọn phiếu và bấm 'Sửa' trước.")
        return
    try:
        sql = """UPDATE khohang SET loaiphieu=%s, ngaylap=%s, nhacungcap=%s,
masanpham=%s,
soluong=%s, dongia=%s, ghichu=%s WHERE maphieu=%s"""

```

```

vals = (
    loaivar.get(),
    ngay_entry.get(),
    nhacungcap_entry.get(),
    int(masp_entry.get()),
    int(soluong_entry.get()),
    float(dongia_entry.get() or 0),
    ghichu_entry.get(),
    edit_mode["id"]
)
cursor.execute(sql, vals)
db_commit(conn)
load_stock()
clear_form()
messagebox.showinfo("Thành công", "Đã cập nhật phiếu kho.")
except Exception as e:
    messagebox.showerror("Lỗi lưu", str(e))

```

--- Xóa phiếu ---

```

def delete_phieu():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn dòng", "Chọn phiếu để xóa.")
        return
    item = tree.item(sel)
    maphieu = item['values'][0]
    if not messagebox.askyesno("Xác nhận", f"Xóa phiếu {maphieu}?"):
        return
    try:
        cursor.execute("DELETE FROM khohang WHERE maphieu=%s", (maphieu,))
        db_commit(conn)
        load_stock()
        clear_form()
        messagebox.showinfo("Đã xóa", "Xóa thành công.")
    
```

```
except Exception as e:
```

```
    messagebox.showerror("Lỗi xóa", str(e))
```

--- Nút chức năng ---

```
btnf = tk.Frame(win); btnf.pack(pady=10)
```

```
    tk.Button(btnf, text="Thêm phiếu", command=add_phieu, width=14).grid(row=0, column=0, padx=5)
```

```
    tk.Button(btnf, text="Sửa phiếu", command=edit_phieu, width=14).grid(row=0, column=1, padx=5)
```

```
    tk.Button(btnf, text="Lưu thay đổi", command=save_edit, width=14).grid(row=0, column=2, padx=5)
```

```
    tk.Button(btnf, text="Xóa phiếu", command=delete_phieu, width=14).grid(row=0, column=3, padx=5)
```

```
    tk.Button(btnf, text="Tải lại", command=load_stock, width=14).grid(row=0, column=4, padx=5)
```

```
    tk.Button(btnf, text="Đóng", command=lambda:[cursor.close(), conn.close(), win.destroy()], width=14).grid(row=0, column=5, padx=5)
```

```
tree.bind("<<TreeviewSelect>>", on_select)
```

```
load_stock()
```

```
win.mainloop()
```

```
btnf = tk.Frame(win); btnf.pack(pady=8)
```

```
    tk.Button(btnf, text="Ghi nhận phiếu", command=add_phieu, width=15).grid(row=0, column=0, padx=5)
```

```
    tk.Button(btnf, text="Xóa phiếu", command=delete_phieu, width=15).grid(row=0, column=1, padx=5)
```

```
    tk.Button(btnf, text="Sửa", width=12, command=edit_phieu).grid(row=0, column=2, padx=5)
```

```
    tk.Button(btnf, text="Đóng", command=lambda:[cursor.close(), conn.close(), win.destroy()], width=15).grid(row=0, column=3, padx=5)
```

```
load_stock()
```

```
win.mainloop()
```

```
# -----
```

```
# QUAN LI KHACH HANG
```

```
# -----
```

```

def open_customer_manager():
    conn = connect_db()
    if not conn:
        return
    cursor = conn.cursor()

    win = tk.Tk()
    win.title("Quản lý khách hàng")
    win.geometry("900x600")

    cols = ("makhachhang","hoten","sodienthoai","diachi","lichsumuahang","chinh sach")
    tree = ttk.Treeview(win, columns=cols, show="headings")
    for c in cols:
        tree.heading(c, text=c)
        tree.column(c, width=130)
    tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)

    frm = tk.Frame(win); frm.pack(padx=10, pady=5, fill="x")
    tk.Label(frm, text="Họ tên").grid(row=0, column=0)
    hoten_entry = tk.Entry(frm); hoten_entry.grid(row=0, column=1)
    tk.Label(frm, text="SĐT").grid(row=0, column=2)
    sdt_entry = tk.Entry(frm); sdt_entry.grid(row=0, column=3)
    tk.Label(frm, text="Địa chỉ").grid(row=1, column=0)
    diachi_entry = tk.Entry(frm); diachi_entry.grid(row=1, column=1)
    tk.Label(frm, text="Lịch sử mua").grid(row=1, column=2)
    lichsu_entry = tk.Entry(frm); lichsu_entry.grid(row=1, column=3)
    tk.Label(frm, text="Chính sách").grid(row=2, column=0)
    chinh sach_entry = tk.Entry(frm); chinh sach_entry.grid(row=2, column=1)

    def load_customers():
        tree.delete(*tree.get_children())
        cursor.execute("SELECT makhachhang,hoten,sodienthoai,diachi,lichsumuahang,chinh sach
FROM khachhang")
        for r in cursor.fetchall():

```

```

tree.insert("", tk.END, values=r)

def add_customer():
    try:
        sql = """INSERT INTO khachhang (hoten, sodienthoai, diachi, lichsumuahang, chinh sach)
                 VALUES (%s,%s,%s,%s,%s)"""
        vals = (
            hoten_entry.get(),
            sdt_entry.get(),
            diachi_entry.get(),
            lichsu_entry.get(),
            chinh sach_entry.get()
        )
        cursor.execute(sql, vals)
        db_commit(conn)
        load_customers()
        messagebox.showinfo("Thành công", "Đã thêm khách hàng.")
    except Exception as e:
        messagebox.showerror("Lỗi", str(e))

def update_customer():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn khách hàng để sửa.")
        return
    mak h = tree.item(sel)["values"][0]
    try:
        sql = """UPDATE khachhang SET hoten=%s, sodienthoai=%s, diachi=%s,
                lichsumuahang=%s, chinh sach=%s
                WHERE mak hachhang=%s"""
        vals = (
            hoten_entry.get(), sdt_entry.get(), diachi_entry.get(), lichsu_entry.get(),
            chinh sach_entry.get(), mak h
        )

```

```

cursor.execute(sql, vals)
db_commit(conn)
load_customers()
messagebox.showinfo("Thành công", "Đã cập nhật.")

except Exception as e:
    messagebox.showerror("Lỗi", str(e))

def delete_customer():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn khách hàng để xóa.")
        return
    makh = tree.item(sel)["values"][0]
    if not messagebox.askyesno("Xác nhận", f"Xóa khách hàng {makh}?"):
        return
    try:
        cursor.execute("DELETE FROM khachhang WHERE makhachhang=%s", (makh,))
        db_commit(conn)
        load_customers()
        messagebox.showinfo("Đã xóa", "Xóa thành công.")
    except Exception as e:
        messagebox.showerror("Lỗi", str(e))

def save_customer():
    sel = tree.focus()
    if sel: # có chọn => cập nhật
        makh = tree.item(sel)["values"][0]
        sql = """UPDATE khachhang
                  SET hoten=%s, sodienthoai=%s, diachi=%s, lichsumuahang=%s, chinh sach=%s
                  WHERE makhachhang=%s"""
        vals = (
            hoten_entry.get(),
            sdt_entry.get(),
            diachi_entry.get(),

```

```

        lichsu_entry.get(),
        chinh sach_entry.get(),
        makh
    )
    cursor.execute(sql, vals)
    db_commit(conn)
    messagebox.showinfo("Thành công", "Đã cập nhật thông tin khách hàng.")

else: # chưa chọn => thêm mới
    sql = """INSERT INTO khachhang (hoten, sodienthoai, diachi, lichsumuahang, chinh sach)
              VALUES (%s, %s, %s, %s, %s)"""
    vals = (
        hoten_entry.get(),
        sdt_entry.get(),
        diachi_entry.get(),
        lichsu_entry.get(),
        chinh sach_entry.get()
    )
    cursor.execute(sql, vals)
    db_commit(conn)
    messagebox.showinfo("Thành công", "Đã lưu khách hàng mới.")

load_customers()

except Exception as e:
    messagebox.showerror("Lỗi", str(e))

def on_select(ev):
    sel = tree.focus()
    if not sel: return
    v = tree.item(sel)["values"]
    hoten_entry.delete(0, tk.END); hoten_entry.insert(0, v[1])
    sdt_entry.delete(0, tk.END); sdt_entry.insert(0, v[2])
    diachi_entry.delete(0, tk.END); diachi_entry.insert(0, v[3])
    lichsu_entry.delete(0, tk.END); lichsu_entry.insert(0, v[4] if v[4] else "")
    chinh sach_entry.delete(0, tk.END); chinh sach_entry.insert(0, v[5] if v[5] else "")

```

```

btmf = tk.Frame(win); btnf.pack(pady=8)

tk.Button(btnf, text="Thêm", command=add_customer, width=12).grid(row=0, column=0, padx=5)
tk.Button(btnf, text="Cập nhật", command=update_customer, width=12).grid(row=0, column=1, padx=5)
tk.Button(btnf, text="Lưu", command=save_customer, width=12).grid(row=0, column=2, padx=5)
tk.Button(btnf, text="Xóa", command=delete_customer, width=12).grid(row=0, column=3, padx=5)
tk.Button(btnf, text="Đóng", command=lambda:[cursor.close(), conn.close(), win.destroy()], width=12).grid(row=0, column=4, padx=5)

tree.bind("<<TreeviewSelect>>", on_select)
load_customers()
win.mainloop()

# -----
# QUAN LI HOA DON (hoadon + chitiethoadon)
# -----



def open_invoice_manager():
    conn = connect_db()
    if not conn:
        return
    cursor = conn.cursor()

    win = tk.Tk()

    win.title("📄 Quản lý Hóa đơn Bán Hàng")
    win.geometry("1150x750")
    win.configure(bg="#f4f6f8")

# ---- FRAME DANH SÁCH HÓA ĐƠN ----

    frm_hd = tk.LabelFrame(win, text="Danh sách hóa đơn", bg="#f4f6f8", fg="#333", font=("Arial", 11, "bold"))
    frm_hd.pack(fill="both", expand=False, padx=10, pady=8)

```

```

cols_hd = ("mahoadon", "ngayban", "tenkhachhang", "tongtien", "nhanvienbanhang", "hinhthuctt",
"ghichu")

tree_hd = ttk.Treeview(frm_hd, columns=cols_hd, show="headings", height=5)
for c in cols_hd:
    tree_hd.heading(c, text=c.title())
    tree_hd.column(c, width=150)
tree_hd.pack(fill=tk.BOTH, expand=True, padx=10, pady=8)

# ---- FRAME CHI TIẾT ----

frm_ct = tk.LabelFrame(win, text="Chi tiết hóa đơn", bg="#f4f6f8", fg="#333", font=("Arial", 11,
"bold"))
frm_ct.pack(fill="both", expand=True, padx=10, pady=8)

cols_ct = ("id","mahoadon", "masanpham", "tensanpham", "soluong", "dongia", "thanhtien")
tree_ct = ttk.Treeview(frm_ct, columns=cols_ct, show="headings",height=5)
for c in cols_ct:
    tree_ct.heading(c, text=c.title())
    tree_ct.column(c, width=120)
tree_ct.pack(fill=tk.BOTH, expand=True, padx=10, pady=8)

# ---- FORM HÓA ĐƠN ----

frm = tk.LabelFrame(win, text="Thông tin hóa đơn", bg="#f4f6f8", fg="#333", font=("Arial", 11,
"bold"))
frm.pack(fill="x", padx=10, pady=5)

tk.Label(frm, text="Ngày (YYYY-MM-DD):", bg="#f4f6f8").grid(row=0, column=0, sticky="e",
pady=3)
ngay_entry = tk.Entry(frm, width=20)
ngay_entry.grid(row=0, column=1, padx=5, pady=3)
ngay_entry.insert(0, datetime.now().strftime("%Y-%m-%d"))

tk.Label(frm, text="Khách hàng:", bg="#f4f6f8").grid(row=0, column=2, sticky="e", pady=3)
tenkh_entry = tk.Entry(frm, width=25)
tenkh_entry.grid(row=0, column=3, padx=5, pady=3)

```

```

tk.Label(frm, text="Nhân viên bán:", bg="#f4f6f8").grid(row=1, column=0, sticky="e", pady=3)
nv_entry = tk.Entry(frm, width=20)
nv_entry.grid(row=1, column=1, padx=5, pady=3)

tk.Label(frm, text="Hình thức TT:", bg="#f4f6f8").grid(row=1, column=2, sticky="e", pady=3)
hinhthuc_var = tk.StringVar(value="tiền mặt")
ttk.Combobox(frm, textvariable=hinhthuc_var, values=["tiền mặt", "chuyển khoản", "trả góp",
"quẹt thẻ"], width=22).grid(row=1, column=3, padx=5, pady=3)

tk.Label(frm, text="Ghi chú:", bg="#f4f6f8").grid(row=2, column=0, sticky="e", pady=3)
ghichu_entry = tk.Entry(frm, width=60)
ghichu_entry.grid(row=2, column=1, columnspan=3, padx=5, pady=3)

# ---- FORM CHI TIẾT ----
frm2 = tk.LabelFrame(win, text="Thêm sản phẩm vào hóa đơn", bg="#f4f6f8", fg="#333",
font=("Arial", 11, "bold"))
frm2.pack(fill="x", padx=10, pady=5)

tk.Label(frm2, text="Mã hóa đơn:", bg="#f4f6f8").grid(row=0, column=0)
mahoadon_entry = tk.Entry(frm2, width=10); mahoadon_entry.grid(row=0, column=1, padx=5)

tk.Label(frm2, text="Mã SP:", bg="#f4f6f8").grid(row=0, column=2)
masp_entry = tk.Entry(frm2, width=10); masp_entry.grid(row=0, column=3, padx=5)

tk.Label(frm2, text="Tên SP:", bg="#f4f6f8").grid(row=1, column=0)
tensp_entry = tk.Entry(frm2, width=20); tensp_entry.grid(row=1, column=1, padx=5)

tk.Label(frm2, text="Số lượng:", bg="#f4f6f8").grid(row=1, column=2)
soluong_entry = tk.Entry(frm2, width=10); soluong_entry.grid(row=1, column=3, padx=5)

tk.Label(frm2, text="Đơn giá:", bg="#f4f6f8").grid(row=2, column=0)
dongia_entry = tk.Entry(frm2, width=10); dongia_entry.grid(row=2, column=1, padx=5)

```

---- HÀM XỬ LÝ ----

```
def load_invoices():
    tree_hd.delete(*tree_hd.get_children())
    cursor.execute("SELECT mahoadon, ngayban, tenkhachhang, tongtien, nhanvienbanhang,
hinhthuctt, ghichu FROM hoadon")
    for r in cursor.fetchall():
        tree_hd.insert("", tk.END, values=r)

def load_invoice_details(mahd):
    tree_ct.delete(*tree_ct.get_children())
    cursor.execute("SELECT id, mahoadon, masanpham, tensanpham, soluong, dongia, thanhtien
FROM chitiethoadon WHERE mahoadon=%s", (mahd,))
    for r in cursor.fetchall():
        tree_ct.insert("", tk.END, values=r)

def create_invoice():
    try:
        sql = """INSERT INTO hoadon (ngayban, tenkhachhang, tongtien, nhanvienbanhang,
hinhthuctt, ghichu)
VALUES (%s,%s,%s,%s,%s,%s)"""
        vals = (ngay_entry.get(), tenkh_entry.get(), 0.0, nv_entry.get(), hinhthuc_var.get(),
ghichu_entry.get())
        cursor.execute(sql, vals)
        db_commit(conn)
        load_invoices()
        messagebox.showinfo("Thành công", "Đã tạo hóa đơn mới.")
    except Exception as e:
        messagebox.showerror("Lỗi tạo hóa đơn", str(e))

def edit_invoice():
    sel = tree_hd.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn hóa đơn để sửa.")
        return
    mahd = tree_hd.item(sel)["values"][0]
```

```

try:
    sql = """UPDATE hoadon SET ngayban=%s, tenkhachhang=%s, nhanvienbanhang=%s,
    hinhthuctt=%s, ghichu=%s WHERE mahoadon=%s"""
    vals = (ngay_entry.get(), tenkh_entry.get(), nv_entry.get(), hinhthuc_var.get(),
            ghichu_entry.get(), mahd)
    cursor.execute(sql, vals)
    db_commit(conn)
    load_invoices()
    messagebox.showinfo("Thành công", "Đã cập nhật hóa đơn.")
except Exception as e:
    messagebox.showerror("Lỗi sửa hóa đơn", str(e))

def add_invoice_item():
    try:
        mahd = int(mahoadon_entry.get())
        masp = int(masp_entry.get())
        tensp = tensp_entry.get()
        sl = int(soluong_entry.get())
        dg = float(dongia_entry.get())

        sql = """INSERT INTO chitiethoadon (mahoadon, masanpham, tensanpham, soluong, dongia)
                 VALUES (%s,%s,%s,%s,%s)"""
        cursor.execute(sql, (mahd, masp, tensp, sl, dg))
        db_commit(conn)

        cursor.execute("SELECT SUM(soluong * dongia) FROM chitiethoadon WHERE
        mahoadon=%s", (mahd,))
        tong = cursor.fetchone()[0] or 0
        cursor.execute("UPDATE hoadon SET tongtien=%s WHERE mahoadon=%s", (tong, mahd))
        db_commit(conn)

        load_invoice_details(mahd)
        load_invoices()
        messagebox.showinfo("Thành công", "Đã thêm sản phẩm vào hóa đơn.")
    except Exception as e:

```

```

messagebox.showerror("Lỗi thêm chi tiết", str(e))

def delete_invoice():
    sel = tree_hd.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn hóa đơn để xóa.")
        return
    mahd = tree_hd.item(sel)["values"][0]
    if not messagebox.askyesno("Xác nhận", f"Xóa hóa đơn {mahd}"):
        return
    try:
        cursor.execute("DELETE FROM chitiethoadon WHERE mahoadon=%s", (mahd,))
        cursor.execute("DELETE FROM hoadon WHERE mahoadon=%s", (mahd,))
        db_commit(conn)
        load_invoices()
        tree_ct.delete(*tree_ct.get_children())
        messagebox.showinfo("Đã xóa", "Xóa hóa đơn thành công.")
    except Exception as e:
        messagebox.showerror("Lỗi xóa", str(e))

def on_invoice_select(ev):
    sel = tree_hd.focus()
    if not sel:
        return
    vals = tree_hd.item(sel)["values"]
    mahd = vals[0]
    mahoadon_entry.delete(0, tk.END)
    mahoadon_entry.insert(0, mahd)
    ngay_entry.delete(0, tk.END)
    ngay_entry.insert(0, vals[1])
    tenkh_entry.delete(0, tk.END)
    tenkh_entry.insert(0, vals[2])
    nv_entry.delete(0, tk.END)
    nv_entry.insert(0, vals[4])

```

```

hinhthuc_var.set(vals[5])
ghichu_entry.delete(0, tk.END)
ghichu_entry.insert(0, vals[6])
load_invoice_details(mahd)

def save_invoice():
    """Lưu thông tin hóa đơn hiện đang nhập (nếu có mã hóa đơn)."""
    try:
        mahd = mahoadon_entry.get().strip()
        if not mahd:
            messagebox.showwarning("Thiếu mã", "Chưa có mã hóa đơn để lưu.")
            return

        sql = """UPDATE hoadon
                 SET ngayban=%s, tenkhachhang=%s, nhanvienbanhang=%s,
                     hinhthuctt=%s, ghichu=%s
                WHERE mahoadon=%s"""

        vals = (ngay_entry.get(), tenkh_entry.get(), nv_entry.get(),
                hinhthuc_var.get(), ghichu_entry.get(), mahd)
        cursor.execute(sql, vals)
        db_commit(conn)
        load_invoices()
        messagebox.showinfo("Thành công", f"Đã lưu thay đổi cho hóa đơn {mahd}.")
    except Exception as e:
        messagebox.showerror("Lỗi lưu", str(e))

```

---- NÚT CHỨC NĂNG ----

```

bt_nf = tk.Frame(win, bg="#f4f6f8")
bt_nf.pack(side="bottom", fill="x", padx=10, pady=10)

style_btn = {"width": 14, "bg": "#4CAF50", "fg": "white", "font": ("Arial", 10, "bold")}

```

NUT CHUC NANG

```

tk.Button(btnf, text="Tạo hóa đơn", command=create_invoice, width=14).grid(row=0, column=0,
padx=6, pady=6)

tk.Button(btnf, text="Sửa hóa đơn", command=edit_invoice, width=14 ).grid(row=0, column=1,
padx=6, pady=6)

tk.Button(btnf, text="Thêm chi tiết", command=add_invoice_item, width=14).grid(row=0,
column=2, padx=6, pady=6)

tk.Button(btnf, text="Xóa hóa đơn", command=delete_invoice, width=14).grid(row=0, column=3,
padx=6, pady=6)

tk.Button(btnf, text="Tải lại", command=load_invoices, width=14).grid(row=0, column=4, padx=6,
pady=6)

tk.Button(btnf, text="Đóng", command=lambda: [cursor.close(), conn.close(), win.destroy()],
width=14).grid(row=0, column=5, padx=6, pady=6)

tk.Button(btnf, text="Lưu", command=save_invoice, width=14).grid(row=0, column=4, padx=6,
pady=6)

for i in range(6):
    btnf.grid_columnconfigure(i, weight=1)

win.geometry("1350x750")

tree_hd.bind("<<TreeviewSelect>>", on_invoice_select)

load_invoices()
win.mainloop()

# -----
# QUAN LI NHAN VIEN
# -----



def open_employee_manager():

    conn = connect_db()

    if not conn:

        return

    cursor = conn.cursor()

    win = tk.Tk()

    win.title("Quản lý nhân viên")

```

```

win.geometry("950x600")

cols = ("manhanvien", "hoten", "vitri", "luong", "thuong", "calam", "hieusuat")
tree = ttk.Treeview(win, columns=cols, show="headings")
for c in cols:
    tree.heading(c, text=c)
    tree.column(c, width=120, anchor="center")
tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)

# --- Form nhập ---
frm = tk.Frame(win)
frm.pack(padx=10, pady=5, fill="x")

labels = ["Họ tên", "Vị trí", "Lương", "Thưởng", "Ca làm", "Hiệu suất"]
entries = {}
for i, lbl in enumerate(labels):
    tk.Label(frm, text=lbl).grid(row=i // 3, column=(i % 3) * 2, sticky="e", padx=5, pady=5)
    e = tk.Entry(frm)
    e.grid(row=i // 3, column=(i % 3) * 2 + 1, padx=5, pady=5)
    entries[lbl] = e

# --- Load danh sách ---
def load_employees():
    tree.delete(*tree.get_children())
    try:
        cursor.execute("SELECT manhanvien, hoten, vitri, luong, thuong, calam, hieusuat FROM nhanvien")
        for r in cursor.fetchall():
            tree.insert("", tk.END, values=r)
    except Exception as e:
        messagebox.showerror("Lỗi tải", str(e))

# --- Thêm ---
def add_employee():

```

```

try:
    sql = """INSERT INTO nhanvien (hoten, vitri, luong, thuong, calam, hieusuat)
              VALUES (%s,%s,%s,%s,%s,%s)"""
    vals = (
        entries["Họ tên"].get(),
        entries["Vị trí"].get(),
        float(entries["Lương"].get() or 0),
        float(entries["Thưởng"].get() or 0),
        entries["Ca làm"].get(),
        float(entries["Hiệu suất"].get() or 0)
    )
    cursor.execute(sql, vals)
    db_commit(conn)
    load_employees()
    messagebox.showinfo("Thành công", "Đã thêm nhân viên.")
except Exception as e:
    messagebox.showerror("Lỗi thêm", str(e))

```

--- Cập nhật ---

```

def update_employee():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn nhân viên để sửa.")
        return
    manv = tree.item(sel)["values"][0]
    try:
        sql = """UPDATE nhanvien SET hoten=%s, vitri=%s, luong=%s, thuong=%s, calam=%s,
hieusuat=%s
                  WHERE manhanvien=%s"""
        vals = (
            entries["Họ tên"].get(),
            entries["Vị trí"].get(),
            float(entries["Lương"].get() or 0),
            float(entries["Thưởng"].get() or 0),

```

```

        entries["Ca làm"].get(),
        float(entries["Hiệu suất"].get() or 0),
        manv
    )
    cursor.execute(sql, vals)
    db_commit(conn)
    load_employees()
    messagebox.showinfo("Thành công", "Đã cập nhật nhân viên.")
except Exception as e:
    messagebox.showerror("Lỗi cập nhật", str(e))

```

--- Xóa ---

```

def delete_employee():
    sel = tree.focus()
    if not sel:
        messagebox.showwarning("Chọn", "Chọn nhân viên để xóa.")
        return
    manv = tree.item(sel)["values"][0]
    if not messagebox.askyesno("Xác nhận", f"Xóa nhân viên {manv}?"):
        return
    try:
        cursor.execute("DELETE FROM nhanvien WHERE manhanvien=%s", (manv,))
        db_commit(conn)
        load_employees()
        messagebox.showinfo("Đã xóa", "Xóa thành công.")
    except Exception as e:
        messagebox.showerror("Lỗi xóa", str(e))

```

--- Lưu (tự động thêm hoặc cập nhật) ---

```

def save_employee():
    sel = tree.focus()
    try:
        if sel: # có dòng được chọn → cập nhật
            manv = tree.item(sel)["values"][0]

```

```

sql = """UPDATE nhanvien
        SET hoten=%s, vitri=%s, luong=%s, thuong=%s, calam=%s, hieusuat=%s
        WHERE manhanvien=%s"""

vals = (
    entries["Họ tên"].get(),
    entries["Vị trí"].get(),
    float(entries["Lương"].get() or 0),
    float(entries["Thưởng"].get() or 0),
    entries["Ca làm"].get(),
    float(entries["Hiệu suất"].get() or 0),
    manv
)
cursor.execute(sql, vals)
db_commit(conn)
messagebox.showinfo("Thành công", "Đã cập nhật thông tin nhân viên.")

else: # chưa chọn → thêm mới
    sql = """INSERT INTO nhanvien (hoten, vitri, luong, thuong, calam, hieusuat)
              VALUES (%s,%s,%s,%s,%s,%s)"""
    vals = (
        entries["Họ tên"].get(),
        entries["Vị trí"].get(),
        float(entries["Lương"].get() or 0),
        float(entries["Thưởng"].get() or 0),
        entries["Ca làm"].get(),
        float(entries["Hiệu suất"].get() or 0)
    )
    cursor.execute(sql, vals)
    db_commit(conn)
    messagebox.showinfo("Thành công", "Đã thêm nhân viên mới.")

load_employees()

except Exception as e:
    messagebox.showerror("Lỗi lưu", str(e))

```

--- Khi chọn dòng ---

```

def on_select(ev):
    sel = tree.focus()
    if not sel:
        return
    v = tree.item(sel)["values"]
    entries["Họ tên"].delete(0, tk.END); entries["Họ tên"].insert(0, v[1])
    entries["Vị trí"].delete(0, tk.END); entries["Vị trí"].insert(0, v[2])
    entries["Lương"].delete(0, tk.END); entries["Lương"].insert(0, v[3])
    entries["Thưởng"].delete(0, tk.END); entries["Thưởng"].insert(0, v[4])
    entries["Ca làm"].delete(0, tk.END); entries["Ca làm"].insert(0, v[5])
    entries["Hiệu suất"].delete(0, tk.END); entries["Hiệu suất"].insert(0, v[6])

# --- Nút chức năng ---
btmf = tk.Frame(win)
btmf.pack(pady=8)
tk.Button(btmf, text="Thêm", command=add_employee, width=12).grid(row=0, column=0, padx=5)
tk.Button(btmf, text="Cập nhật", command=update_employee, width=12).grid(row=0, column=1, padx=5)
tk.Button(btmf, text="Xóa", command=delete_employee, width=12).grid(row=0, column=2, padx=5)
tk.Button(btmf, text="Lưu", command=save_employee, width=12).grid(row=0, column=3, padx=5)
tk.Button(btmf, text="Đóng", command=lambda:[cursor.close(), conn.close(), win.destroy()], width=12).grid(row=0, column=4, padx=5)

tree.bind("<<TreeviewSelect>>", on_select)
load_employees()
win.mainloop()

# -----
# QUAN LY HE THONG CHINH
# -----

def main_menu():
    root = tk.Tk()
    root.title("MENU CHÍNH - QUẢN LÝ CỦA HÀNG TIVI")

```

```
root.geometry("600x450")
root.resizable(False, False)

tk.Label(root, text="QUẢN LÝ CỦA HÀNG TIVI", font=("Arial", 16, "bold")).pack(pady=20)

tk.Button(root, text="Quản lý sản phẩm", width=30, height=2, command=lambda:[root.destroy(), open_product_manager()]).pack(pady=8)

tk.Button(root, text="Quản lý kho hàng", width=30, height=2, command=lambda:[root.destroy(), open_stock_manager()]).pack(pady=8)

tk.Button(root, text="Quản lý hóa đơn", width=30, height=2, command=lambda:[root.destroy(), open_invoice_manager()]).pack(pady=8)

tk.Button(root, text="Quản lý khách hàng", width=30, height=2, command=lambda:[root.destroy(), open_customer_manager()]).pack(pady=8)

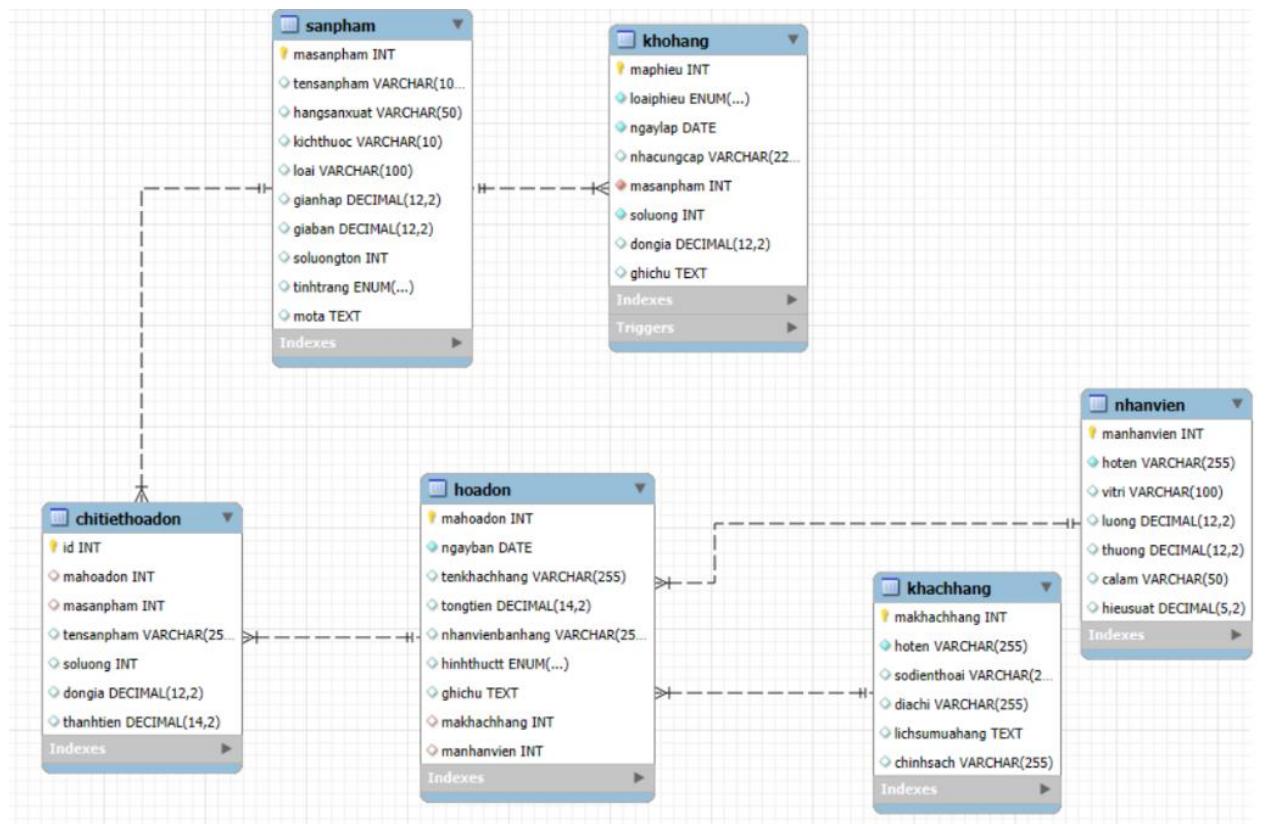
tk.Button(root, text="Quản lý nhân viên", width=30, height=2,
         command=lambda:[root.destroy(), open_employee_manager()]).pack(pady=8)

tk.Button(root, text="Thoát", width=30, height=2, command=root.destroy).pack(pady=8)

root.mainloop()

if __name__ == "__main__":
    main_menu()
```

4. Cơ sở dữ liệu MySQL



Hình 7: Bảng khóa chính khóa ngoại của cơ sở dữ liệu MySQL

TÀI LIỆU THAM KHẢO

Ths. Nguyễn Ngọc Minh(2025). Tài liệu hướng dẫn làm đồ án lập trình Python. Khoa Công Nghệ Thông Tin – Trường Đại Học An Giang