TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: HỆ QUẢN TRỊ CSDL VỚI ORACLE

MÃ ĐỀ THI: 18

TÊN ĐỀ TÀI: XÂY DỰNG CƠ SỞ DỮ LIỆU CHO PHẦN MỀM QUẢN LÝ CÀ PHÊ

LỚP TÍN CHỈ: HQTCO.03.K13.09.LH.C04.1_LT

Giảng viên hướng dẫn: ThS.Trần Thị Thúy Hằng

Danh sách sinh viên thực hiện: Nhóm 14

STT	Mã sinh viên	Sinh viên thực hiện	Lớp hành chính
1	20223155	Nguyễn Trí Dũng	DCCNTT13.10.16
2	20223110	Nguyễn Văn Duy	DCCNTT13.10.16
3	20223011	Hoàng Ngọc Thành	DCCNTT13.10.16

Bắc Ninh - 2025

MỤC LỤC

DANH MỤC BẢNG BIỂU 4

DANH MỤC HÌNH ẢNH 7

Mở Đầu 8

1. Giới thiệu tổng quan đề tài 8

1.1. Lý do chọn đề tài, tính thực tiễn hoặc cấp thiết. 8

1.2. Mục tiêu chung của đề tài. 8

1.3. Tổng quan chức năng chính của hệ thống. 9

2. Danh sách thành viên và phân công công việc 10

Chương I. Cơ Sở Lý Thuyết 12

1.1. Giới thiệu Hệ quản trị cơ sở dữ liệu 12

1.1.1. Oracle Database 12

1.1.2. PL/SQL 13

1.2. Giới thiệu công cụ sử dụng 13

1.2.1. SQL Developer 13

1.2.2. VSCode Studio 2022 14

1.2.3. Draw.io 15

Chương II. Thiết Kế và Xây Dựng CSDL Ứng Dụng 16

2.1. Mô tả chức năng ứng dụng 16

2.2. Thiết kế CSDL 16

2.3. Truy vấn SQL 20

2.4. Lập trình PL/SQL 33

2.5. Quản trị người dùng & sao lưu 72

2.5.1. Tạo User mới 72

2.5.2. Cấp quyền (GRANT) 73

2.5.3. Thu hồi quyền (REVOKE) 73

2.5.4. Xóa User (nếu cần) 73

2.5.5. Backup bằng expdp (Data Pump Export) 73

2.5.6. Phục hồi bằng impdp (Data Pump Import) 73

2.5.7. Tạo DIRECTORY cho xuất/nhập 74

2.5.8. Backup bảng cụ thể 74

Chương III. Kết luận 75

Tài liệu tham khảo 77

DANH MỤC BẢNG BIỂU

Bảng 1. 1: Bảng CRUD tổng quát của quản trị viên 9

Bảng 1. 2: Bảng chức năng của nhân viên 9

Bảng 1. 3: Bảng phân công công việc 10

Bảng 2. 1: Bảng mô tả cấu trúc bảng Employees 17

Bảng 2. 2: Bảng mô tả cấu trúc bảng Accounts 18

Bảng 2. 3: Bảng mô tả cấu trúc bảng Categories 18

Bảng 2. 4: Bảng mô tả cấu trúc bảng Product 19

Bảng 2. 5: Bảng mô tả cấu trúc bảng Invoices 20

Bảng 2. 6: Truy vấn cơ bản 1 20

Bảng 2. 7: Truy vấn cơ bản 2 21

Bảng 2. 8: Truy vấn cơ bản 3 22

Bảng 2. 9: Truy vấn cơ bản 4 22

Bảng 2. 10: Truy vấn cơ bản 5 23

Bảng 2. 11: Truy vấn lồng nhau 1 24

Bảng 2. 12: Truy vấn lồng nhau 2 24

Bảng 2. 13: Truy vấn lồng nhau 3 25

Bảng 2. 14: Truy vấn lồng nhau 4 26

Bảng 2. 15: Truy vấn lồng nhau 5 26

Bảng 2. 16: Truy vấn lồng nhau 6 27

Bảng 2. 17: Truy vấn có nhóm 1 28

Bảng 2. 18: Truy vấn có nhóm 2 29

Bảng 2. 19: Truy vấn có nhóm 3 30

Bảng 2. 20: Truy vấn có nhóm 4 30

Bảng 2. 21: Hàm tổng hợp 1 31

Bảng 2. 22: Hàm tổng hợp 2 32
Bảng 2. 23: Hàm tổng hợp 3 33
Bảng 2. 24: Hàm views 1 33
Bảng 2. 25: Hàm views 2 34
Bảng 2. 26: Hàm views 3 35
Bảng 2. 27: Hàm views 4 36
Bảng 2. 28: Hàm views 5 37
Bảng 2. 29: Hàm views 6 37
Bảng 2. 30: Hàm 1 38
Bảng 2. 31: Hàm 2 39
Bảng 2. 32: Hàm 3 40
Bảng 2. 33: Hàm 4 41
Bảng 2. 34: Hàm 5 42
Bảng 2. 35: Procedures Check tài khoản đăng nhập 43
Bảng 2. 36: Procedures Tạo tài khoản nhân viên 45
Bảng 2. 37: Procedures Đổi mật khẩu của account 48
Bảng 2. 38: Procedures Thêm danh mục mới 49
Bảng 2. 39: Procedures Cập nhật thông tin danh mục 50
Bảng 2. 40: Procedures Xóa danh mục 52
Bảng 2. 41: Procedures Tạo sản phẩm mới 52
Bảng 2. 42: Procedures Cập nhật sản phẩm 54
Bảng 2. 43: Procedures Xóa sản phẩm 56
Bảng 2. 44: Procedures Tạo hóa đơn mới 57
Bảng 2. 45: Procedures Cập nhật hóa đơn 59
Bảng 2. 46: Procedures Xóa hóa đơn 60

Bảng 2. 47: Procedures Cập nhật nhân viên 61

Bảng 2. 48: Procedures Khoá nhân viên 63

Bảng 2. 49: Procedures thống kê báo cáo 64

Bảng 2. 50: Trigger ghi log khi thêm danh mục 67

Bảng 2. 51: Trigger kiểm tra tên danh mục trùng lặp khi thêm mới 68

Bảng 2. 52: Trigger lưu lại thông tin trước khi sửa sản phẩm 70

DANH MỤC HÌNH ẢNH

Hình 1. 1: Hình minh họa Oracle 12

Hình 1. 2: Hình ảnh minh họa PL/SQL 13

Hình 1. 3: Hình ảnh minh họa SQL Developer 14

Hình 1. 4: Hình ảnh minh họa VSCode Studio 2022 14

Hình 1. 5: Hình ảnh minh họa Draw.io 15

Hình 2. 1: Sơ đồ ERD 17

Mở Đầu

1. Giới thiệu tổng quan đề tài

1.1. Lý do chọn đề tài, tính thực tiễn hoặc cấp thiết.

Trong thời đại công nghệ số hiện nay, ngành dịch vụ ăn uống, đặc biệt là các quán cà phê, đang phát triển mạnh mẽ. Tuy nhiên, việc quản lý quán cà phê theo phương pháp thủ công thường gặp nhiều bất cập như sai sót trong ghi chép, khó khăn trong việc kiểm soát doanh thu, tồn kho và chất lượng dịch vụ. Điều này ảnh hưởng trực tiếp đến hiệu quả kinh doanh và sự hài lòng của khách hàng.

Theo thông tin từ trang PosApp, phần mềm quản lý quán cà phê giúp tối ưu hóa quy trình order và tính tiền, hỗ trợ thanh toán đa phương thức, quản lý nguyên liệu và pha chế hiệu quả, nâng cao chất lượng phục vụ khách hàng. Việc sử dụng phần mềm quản lý giúp quán cà phê tự động hóa quy trình order, thanh toán, in hóa đơn, đồng bộ đơn hàng giữa các bộ phận Order - Pha chế - Thu ngân, giúp món được chế biến đúng yêu cầu khách, thanh toán chính xác.

Ngoài ra, theo Faceworks, phần mềm quản lý quán cà phê cung cấp các tính năng như quản lý menu, quản lý kho hàng, quản lý quy trình pha chế đồ uống, quản lý khách hàng, quản lý bán hàng và quản lý tài chính. Những tính năng này giúp chủ quán kiểm soát chặt chẽ hoạt động kinh doanh, giảm thiểu thất thoát và nâng cao hiệu quả quản lý.

Từ những lý do trên, việc xây dựng phần mềm quản lý quán cà phê không chỉ đáp ứng nhu cầu thực tiễn mà còn là một đề tài có tính ứng dụng cao trong lĩnh vực công nghệ thông tin. Đề tài này không chỉ giúp sinh viên áp dụng kiến thức đã học vào thực tế mà còn góp phần nâng cao hiệu quả kinh doanh cho các quán cà phê.

1.2. Mục tiêu chung của đề tài.

Thiết kế cơ sở dữ liệu tối ưu cho phần mềm quản lý quán cà phê.

Xây dựng các bảng dữ liệu, mối quan hệ giữa các bảng một cách hợp lý, khoa học.

Thực hiện các thao tác CRUD (Tạo, Đọc, Cập nhật, Xóa) dữ liệu bằng ngôn ngữ SQL trên Oracle.

Phát triển các chức năng quản lý như: danh mục sản phẩm, hóa đơn, tài khoản, nhân viên…

Hỗ trợ các chức năng tìm kiếm và thống kê báo cáo phục vụ công tác quản lý.

1.3. Tổng quan chức năng chính của hệ thống.

Hệ thống phần mềm quản lý quán cà phê được thiết kế với các chức năng chính sau, phân quyền theo hai nhóm người dùng: Administrator (Quản trị viên) và Employee (Nhân viên):

a, Nhóm chức năng CRUD:

Bảng 1. 1: Bảng CRUD tổng quát của quản trị viên

STT	Chức năng	Người dùng	Mô tả chung
1	Quản lý danh mục (Categories)	Administrator	Thêm, sửa, xoá danh mục sản phẩm (Cà phê, Nước ép...)
2	Quản lý sản phẩm (Products)	Administrator	Thêm, sửa, xoá sản phẩm thuộc danh mục
3	Quản lý nhân viên (Employees)	Administrator	Sửa thông tin nhân viên, khoá tài khoản khi cần thiết
4	Quản lý tài khoản (Accounts)	Administrator	Thêm, sửa, xoá tài khoản đăng nhập
5	Quản lý hoá đơn (Invoices)	Administrator + Employee	Thêm mới, sửa hoặc xoá hóa đơn bán hàng

b, Nhóm chức năng nhân viên:

Bảng 1. 2: Bảng chức năng của nhân viên

STT	Chức năng	Nhân viên	Mô tả chung
1	Đăng nhập	Administrator +	Xác thực người dùng vào hệ thống

		Employee	
2	Mua hàng	Employee	Tạo hóa đơn bán hàng, chọn sản phẩm
3	Đổi mật khẩu	Employee	Đổi mật khẩu cá nhân
4	Đăng ký tài khoản	Employee	Tạo tài khoản cho nhân viên mới (nếu được cấp quyền)

c, Chức năng tìm kiếm:

Tìm nhanh theo từ khóa: Tìm kiếm theo tên sản phẩm, tên danh mục, tên nhân viên…

Tìm nâng cao theo tiêu chí: Lọc theo giá, tên (A → Z), hoặc các thuộc tính khác.

d, Chức năng thống kê / báo cáo:

Thống kê doanh thu theo ngày.

Thống kê doanh thu theo tháng.

Liệt kê các mặt hàng bán chạy nhất.

Thống kê nhân viên bán hàng giỏi, theo số lượng hóa đơn hoặc doanh thu.

2. Danh sách thành viên và phân công công việc

Bảng 1. 3: Bảng phân công công việc

STT	**Họ và tên**	**Mã sinh viên**	**Vai trò chính**	**Nhiệm vụ cụ thể**
1	Nguyễn Trí Dũng	20223155	Trưởng nhóm, Thiết kế CSDL	- Phân tích yêu cầu hệ thống - Thiết kế mô hình ERD - Phân công công việc cho các thành viên
2	Nguyễn Văn Duy	20223155	Lập trình PL/SQL	- Viết các thủ tục (Procedures), hàm (Functions)
3	Hoàng Ngọc Thành	20223155	Lập trình	- Tạo và quản lý các View

			PL/SQL	- Viết các hàm Trigger

Chương I. Cơ Sở Lý Thuyết

1.1. Giới thiệu Hệ quản trị cơ sở dữ liệu

1.1.1. Oracle Database

Trong bài tập lớn này, nhóm chúng em lựa chọn sử dụng Oracle Database – một trong những hệ quản trị cơ sở dữ liệu (HQT CSDL) mạnh mẽ và phổ biến nhất hiện nay, đặc biệt phù hợp cho các ứng dụng doanh nghiệp với quy mô từ nhỏ đến lớn. Cùng với đó, nhóm sử dụng PL/SQL (Procedural Language/SQL) – ngôn ngữ lập trình mở rộng của SQL do Oracle phát triển, hỗ trợ xử lý logic phức tạp trong cơ sở dữ liệu.

Hình 1. 1: Hình minh họa Oracle

Tổng quan về Oracle Database là một HQT CSDL quan hệ (RDBMS – Relational Database Management System) do hãng Oracle Corporation phát triển. Đây là một hệ thống mạnh mẽ, được sử dụng rộng rãi trong các tổ chức tài chính, doanh nghiệp lớn, chính phủ, và nhiều lĩnh vực khác nhờ vào khả năng mở rộng, bảo mật và hiệu năng cao.

Trong bài này, nhóm sử dụng phiên bản Oracle Database 21c – một trong những phiên bản ổn định và lâu dài (Long Term Support) được khuyến nghị dùng trong các môi trường doanh nghiệp và giáo dục. Các đặc trưng nổi bật của Oracle Database:

- Quản lý dữ liệu mạnh mẽ: Hỗ trợ các loại dữ liệu phức tạp, khả năng kiểm soát truy cập, sao lưu, khôi phục và tối ưu hóa hiệu suất.
- Tính toàn vẹn dữ liệu cao: Hỗ trợ ràng buộc khóa chính, khóa ngoại, kiểm tra dữ liệu đầu vào và trigger để đảm bảo dữ liệu nhất quán.
- Bảo mật cao: Có các cơ chế bảo mật nâng cao như mã hóa dữ liệu, phân quyền truy cập, kiểm tra lịch sử thao tác.

- Khả năng mở rộng và đồng thời: Hỗ trợ nhiều người dùng truy cập cùng lúc với cơ chế quản lý giao dịch và khóa (transaction & locking) chặt chẽ.
- Tích hợp với PL/SQL: PL/SQL cho phép viết các chương trình logic (thủ tục, hàm, trigger) chạy trực tiếp trên server, giảm thiểu độ trễ và tăng hiệu quả xử lý.

1.1.2. PL/SQL

PL/SQL là ngôn ngữ lập trình thủ tục tích hợp với SQL, được Oracle phát triển nhằm bổ sung khả năng lập trình vào SQL. Một số đặc điểm nổi bật:

Tích hợp chặt chẽ với SQL: Cho phép thực hiện các truy vấn SQL trong các khối lệnh logic.

Hình 1. 2: Hình ảnh minh họa PL/SQL

Hỗ trợ lập trình hướng thủ tục: Có thể khai báo biến, vòng lặp, điều kiện, xử lý ngoại lệ.

Tái sử dụng mã: Thông qua các thủ tục, hàm, gói và trigger.

Cải thiện hiệu năng: Vì mã PL/SQL chạy trong server nên giảm được chi phí truyền dữ liệu qua lại giữa ứng dụng và cơ sở dữ liệu.

1.2. Giới thiệu công cụ sử dụng

Trong quá trình thực hiện bài tập lớn, nhóm sử dụng ba công cụ chính để thiết kế, lập trình và trình bày mô hình cơ sở dữ liệu, bao gồm: SQL Developer, Visual Studio Code 2022, và Draw.io.

1.2.1. SQL Developer

SQL Developer là một công cụ miễn phí do Oracle cung cấp, hỗ trợ quản trị cơ sở dữ liệu và viết mã PL/SQL một cách thuận tiện. Đây là phần mềm tích hợp giao diện đồ

họa (GUI), giúp người dùng thực hiện các truy vấn SQL, quản lý bảng, thủ tục, trigger và dữ liệu một cách trực quan. Tính năng nổi bật:

Kết nối và quản lý nhiều cơ sở dữ liệu Oracle cùng lúc.

Soạn thảo, chạy và debug mã PL/SQL.

Hình 1. 3: Hình ảnh minh họa SQL Developer

Xem cấu trúc bảng, dữ liệu, quan hệ giữa các đối tượng.

Hỗ trợ xuất dữ liệu và sao lưu.

Phiên bản sử dụng: Oracle SQL Developer 21.4.3

1.2.2. VSCode Studio 2022

Visual Studio Code (VSCode) là một trình soạn thảo mã nguồn mạnh mẽ, hỗ trợ đa nền tảng và đa ngôn ngữ lập trình. Trong đề tài này, nhóm sử dụng VSCode để viết tài liệu markdown, soạn báo cáo định dạng văn bản và hỗ trợ làm việc với các tập tin SQL hoặc PL/SQL. Tính năng nổi bật:

Giao diện gọn nhẹ, hỗ trợ mở nhiều tệp cùng lúc

Hỗ trợ highlight và gợi ý cú pháp PL/SQL khi cài plugin.

Tích hợp hệ thống quản lý phiên bản Git.

Hình 1. 4: Hình ảnh minh họa VSCode Studio 2022

Phù hợp cho việc biên soạn tài liệu, đặc biệt là viết báo cáo.

Phiên bản sử dụng: Visual Studio Code 1.77 (2022)

1.2.3. Draw.io

Draw.io (nay còn gọi là diagrams.net) là một công cụ trực tuyến dùng để vẽ sơ đồ, lưu đồ và mô hình ERD (Entity Relationship Diagram) phục vụ thiết kế cơ sở dữ liệu. Tính năng nổi bật:

Miễn phí, dễ sử dụng, hỗ trợ vẽ sơ đồ ERD, Use Case, Flowchart...

Tích hợp lưu trữ trên Google Drive, OneDrive hoặc máy tính cá nhân.

Hình 1. 5: Hình ảnh minh họa Draw.io

Cho phép xuất sơ đồ sang nhiều định dạng như PNG, PDF, XML.

Hữu ích cho việc trình bày mô hình thực thể quan hệ một cách trực quan trong báo cáo bài giảng.

Phiên bản sử dụng: Draw.io bản web – online

Chương II. Thiết Kế và Xây Dựng CSDL Ứng Dụng

2.1. Mô tả chức năng ứng dụng

Các chức năng nên được mô tả theo nhóm:

Chức năng CRUD (Thêm/Sửa/Xoá dữ liệu): Quản lý danh mục (Categories)

Chức năng CRUD (Thêm/Sửa/Xoá dữ liệu): Quản lý sản phẩm (Product)

Chức năng CRUD (Sửa/Khóa dữ liệu): Quản lý nhân viên (Employees)

Chức năng CRUD (Thêm/Sửa/Xoá dữ liệu): Quản lý tài khoản (Accounts)

Chức năng CRUD (Thêm/Sửa/Xoá dữ liệu): Quản lý hóa đơn (Invoices)

Chức năng Đăng nhập

Chức năng Mua hàng

Chức năng Đổi mật khẩu

Chức năng Đăng ký

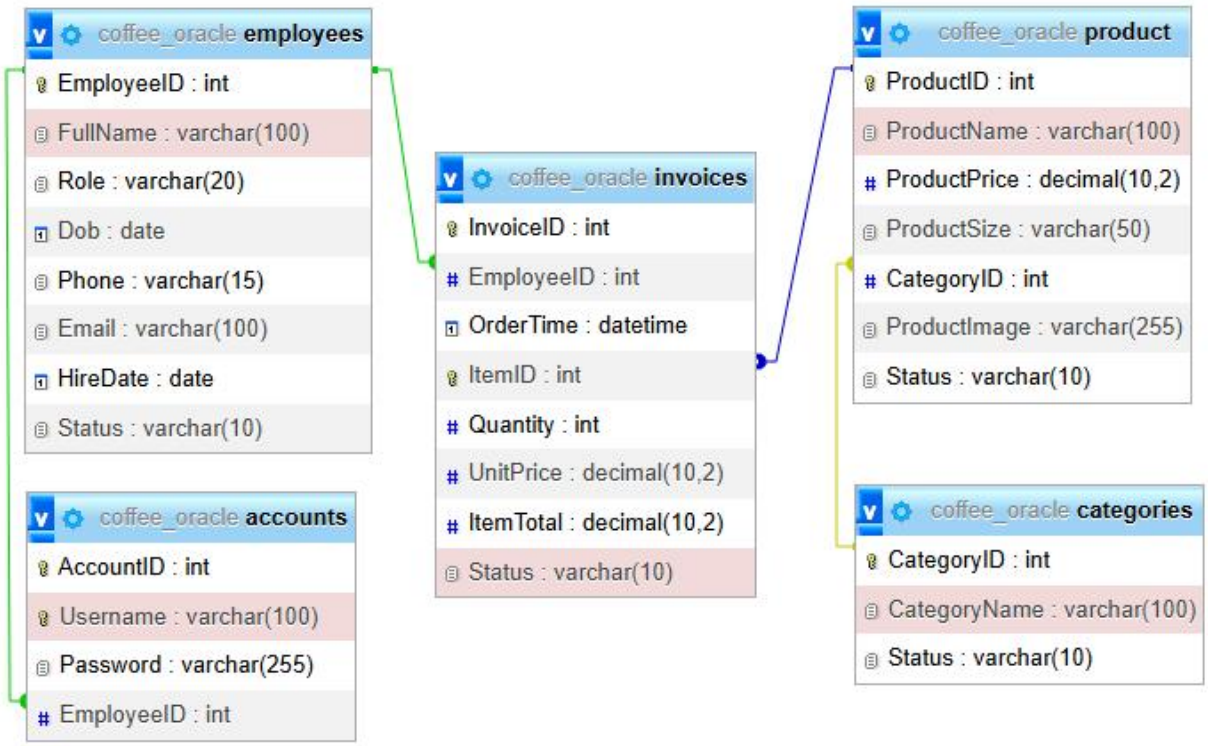
Tìm kiếm:

Tìm nhanh theo từ khoá (tên sản phẩm, tên danh mục, tên nhân viên...).

Tìm nâng cao theo tiêu chí (giá, A -> Z,...).

Thống kê/Báo cáo: Thống kê doanh thu ngày, thống kê doanh thu tháng, mặt hàng bán chạy, nhân viên bán được hàng nhiều...

2.2. Thiết kế CSDL



Hình 2. 1: Sơ đồ ERD

Mô tả các bảng dữ liệu:

Bảng 2. 1: Bảng mô tả cấu trúc bảng Employees

Tên trường	Mô tả	Kiểu dữ liệu	Khóa Chính	Khóa Ngoại	Ràng buộc
EmployeeID	Mã nhân viên	NUMBER	Primary Key		
FullName	Họ và tên nhân viên	VARCHAR2(100)			NOT NULL
Role	Vai trò (Chức vụ)	VARCHAR2(20)			
Dob	Ngày sinh	DATE			
Phone	Số điện thoại	VARCHAR2(15)			

Email	Email	VARCHAR2(100)			
HireDate	Ngày vào làm	DATE			
Status	Trạng thái làm việc	VARCHAR2(10)			DEFAULT 'Active'

Bảng 2. 2: Bảng mô tả cấu trúc bảng Accounts

Tên trường	**Mô tả**	**Kiểu dữ liệu**	**Khóa Chính**	**Khóa Ngoại**	**Ràng buộc**
AccountID	Mã tài khoản	NUMBER	Primary Key		
Username	Tên đăng nhập	VARCHAR2 (100)			NOT NULL, UNIQUE
Password	Mật khẩu	VARCHAR2 (255)			NOT NULL
EmployeeID	Mã nhân viên liên kết	NUMBER		fk_accounts_ employee	Tham chiếu Employees(E mployeeID)

Bảng 2. 3: Bảng mô tả cấu trúc bảng Categories

Tên trường	**Mô tả**	**Kiểu dữ liệu**	**Khóa Chính**	**Khóa Ngoại**	**Ràng buộc**
CategoryID	Mã danh mục	NUMBER	Primary Key		
CategoryName	Tên danh	VARCHAR2(100)			NOT NULL

	mục				
Status	Trạng thái danh mục	VARCHAR2(10)			DEFAULT 'Active'

Bảng 2. 4: Bảng mô tả cấu trúc bảng Product

Tên trường	**Mô tả**	**Kiểu dữ liệu**	**Khóa Chính**	**Khóa Ngoại**	**Ràng buộc**
ProductID	Mã sản phẩm	NUMBER	Primary Key		
ProductName	Tên sản phẩm	VARCHAR2 (100)			NOT NULL
ProductPrice	Giá sản phẩm	NUMBER(10, 2)			NOT NULL
ProductSize	Kích thước sản phẩm	VARCHAR2 (50)			
CategoryID	Mã danh mục sản phẩm	NUMBER		fk_product_category	Tham chiếu Categories(CategoryID)
ProductImage	Hình ảnh sản phẩm	VARCHAR2 (255)			
Status	Trạng thái sản phẩm	VARCHAR2 (10)			DEFAULT 'Active'

Bảng 2. 5: Bảng mô tả cấu trúc bảng Invoices

Tên trường	Mô tả	Kiểu dữ liệu	Khóa Chính	Khóa Ngoại	Ràng buộc
InvoiceID	Mã hóa đơn	NUMBER	Primary Key		
EmployeeID	Mã nhân viên lập hóa đơn	NUMBER		fk_invoice_employee	Tham chiếu Employees(EmployeeID)
OrderTime	Thời gian đặt hàng	DATE			DEFAULT SYSDATE
ItemID	Mã sản phẩm trong hóa đơn	NUMBER		fk_invoice_item	Tham chiếu Product(ProductID)
Quantity	Số lượng	NUMBER			DEFAULT 1
UnitPrice	Giá đơn vị	NUMBER (10, 2)			
ItemTotal	Thành tiền	NUMBER (10, 2)			
Status	Trạng thái hóa đơn	VARCHAR2(10)			DEFAULT 'Unpaid'

2.3. Truy vấn SQL

- **Truy vấn cơ bản (SELECT, WHERE,...)**

Bảng 2. 6: Truy vấn cơ bản 1

Chức năng cơ bản 1: Liệt kê tất cả các sản phẩm thuộc danh mục "Trà"

Câu lệnh	SELECT p.ProductID, p.ProductName, p.ProductPrice, p.ProductSize FROM Product p JOIN Categories c ON p.CategoryID = c.CategoryID WHERE c.CategoryName = 'Trà';
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 4 in 0 seconds PRODUCTID, PRODUCTNAME, PRODUCTPRICE, PRODUCTSIZE 1 7 Trà Thạch Vải 45000 S 2 6 Trà Ngọc Trai Dâu Tằm 59000 M 3 5 Trà Xanh Đậu Đỏ 45000 S 4 8 Trà Thạch Đào 45000 M

Bảng 2. 7: Truy vấn cơ bản 2

Chức năng cơ bản 2: Tìm thông tin nhân viên có vai trò là "Administrator" và đang làm việc (Status = 'Active')	
Câu lệnh	SELECT EmployeeID, FullName, Role, HireDate FROM Employees WHERE Role = 'Administrator' AND Status = 'Active';
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 2 in 0,001 seconds EMPLOYEEID, FULLNAME, ROLE, HIREDATE 1 1 Nguyễn Trí Dũng Administrator 10-JAN-25 2 8 Bùi Thị H Administrator 01-APR-15

Bảng 2. 8: Truy vấn cơ bản 3

<table>
<tr><th colspan="2">Chức năng cơ bản 3: Liệt kê các hóa đơn đã thanh toán (Paid), sắp xếp theo thời gian giảm dần</th></tr>
<tr><td>Câu lệnh</td><td>SELECT DISTINCT InvoiceID, EmployeeID, OrderTime, Status
FROM Invoices
WHERE Status = 'Paid'
ORDER BY OrderTime DESC;</td></tr>
<tr><td>Hình ảnh thực tế</td><td>Script Output x | Query Result x
SQL | All Rows Fetched: 6 in 0,001 seconds
<table>
<tr><th></th><th>INVOICEID</th><th>EMPLOYEEID</th><th>ORDERTIME</th><th>STATUS</th></tr>
<tr><td>1</td><td>1009</td><td>9</td><td>23-MAY-25</td><td>Paid</td></tr>
<tr><td>2</td><td>1008</td><td>8</td><td>23-MAY-25</td><td>Paid</td></tr>
<tr><td>3</td><td>1006</td><td>6</td><td>22-MAY-25</td><td>Paid</td></tr>
<tr><td>4</td><td>1004</td><td>1</td><td>22-MAY-25</td><td>Paid</td></tr>
<tr><td>5</td><td>1002</td><td>3</td><td>20-MAY-25</td><td>Paid</td></tr>
<tr><td>6</td><td>1001</td><td>2</td><td>20-MAY-25</td><td>Paid</td></tr>
</table></td></tr>
</table>

Bảng 2. 9: Truy vấn cơ bản 4

Chức năng cơ bản 4: Thống kê tổng tiền đã bán được theo từng sản phẩm	
Câu lệnh	SELECT p.ProductName, SUM(i.Quantity) AS TotalSold, SUM(i.ItemTotal) AS TotalRevenue FROM Invoices i JOIN Product p ON i.ItemID = p.ProductID WHERE i.Status = 'Paid' GROUP BY p.ProductName ORDER BY TotalRevenue DESC;

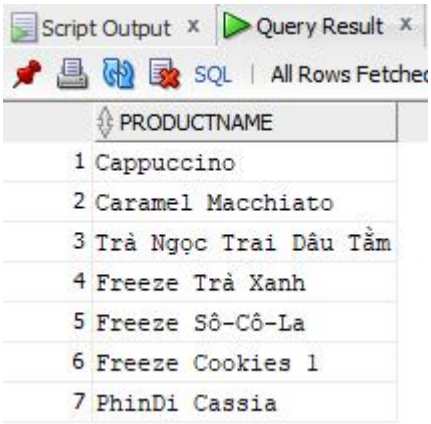
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 10 in 0,002 seconds |	PRODUCTNAME	TOTALSOLD	TOTALREVENUE
---	---	---	---
1	Trà Xanh Đậu Đỏ	3	135000
2	Caramel Macchiato	1	69000
3	Cappuccino	1	65000
4	Trà Ngọc Trai Dâu Tằm	1	59000
5	Freeze Sô-Cô-La	1	55000
6	Phindi Hạnh Nhân	1	55000
7	Phindi Choco	1	45000
8	Trà Thạch Vải	1	45000
9	PhinDi Kem Sữa	1	45000
10	Americano	1	45000

Bảng 2. 10: Truy vấn cơ bản 5

Chức năng cơ bản 5: Liệt kê các tài khoản và tên đầy đủ của nhân viên tương ứng	
Câu lệnh	SELECT a.Username, e.FullName, e.Role FROM Accounts a JOIN Employees e ON a.EmployeeID = e.EmployeeID;
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 14 in 0,001 seconds |	USERNAME	FULLNAME	ROLE
---	---	---	---
1	admin	Nguyễn Trí Dũng	Administrator
2	duyahihi	Nguyễn Văn Duy	Employee
3	hoangnt	Hoàng Ngọc Thành	Employee
4	phamtd	Phạm Thị D	Employee
5	hoangve	Hoàng Văn E	Employee
6	dothif	Đỗ Thị F	Employee
7	ngovang	Ngô Văn G	Employee
8	buithih	Bùi Thị H	Administrator
9	vuvan	Vũ Văn I	Employee
10	tathik	Tạ Thị K	Employee
11	newuser1	Guest 1	Employee
12	asasd1	Guest 2	Employee
13	nguyenabc	Guest 3	Employee
14	user456	Nguyễn Văn Duy	Employee

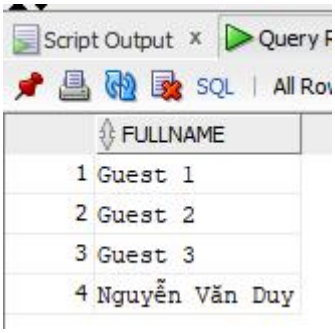
- **Truy vấn lồng nhau (subquery)**

Bảng 2. 11: Truy vấn lồng nhau 1

Chức năng 1: Sản phẩm có giá cao hơn giá trung bình	
Câu lệnh	SELECT ProductName FROM Product WHERE ProductPrice > (SELECT AVG(ProductPrice) FROM Product);
Hình ảnh thực tế	

Bảng 2. 12: Truy vấn lồng nhau 2

Chức năng 2: Nhân viên có tài khoản nhưng chưa lập hóa đơn nào	
Câu lệnh	SELECT FullName FROM Employees WHERE EmployeeID IN (SELECT EmployeeID FROM Accounts)

	AND EmployeeID NOT IN (SELECT DISTINCT EmployeeID FROM Invoices);
Hình ảnh thực tế	

Bảng 2. 13: Truy vấn lồng nhau 3

Chức năng 3: Nhân viên có số hóa đơn nhiều hơn trung bình	
Câu lệnh	SELECT FullName FROM Employees WHERE EmployeeID IN (SELECT EmployeeID FROM Invoices GROUP BY EmployeeID HAVING COUNT(*) > (SELECT AVG(sohd) FROM (SELECT COUNT(*) AS sohd FROM Invoices GROUP BY EmployeeID)

<table>
<tr><td></td><td>)

);</td></tr>
<tr><td>Hình ảnh thực tế</td><td>Script Output x | Query Re
SQL | All Rows
<table><tr><th></th><th>FULLNAME</th></tr><tr><td>1</td><td>Hoàng Ngọc Thành</td></tr><tr><td>2</td><td>Phạm Thị D</td></tr><tr><td>3</td><td>Hoàng Văn E</td></tr><tr><td>4</td><td>Đỗ Thị F</td></tr><tr><td>5</td><td>Bùi Thị H</td></tr><tr><td>6</td><td>Vũ Văn I</td></tr></table></td></tr>
</table>

Bảng 2. 14: Truy vấn lồng nhau 4

<table>
<tr><td colspan="2">Chức năng 4: Tìm các hóa đơn có tổng thành tiền (ItemTotal) cao hơn tổng trung bình tất cả hóa đơn</td></tr>
<tr><td>Câu lệnh</td><td>SELECT *

FROM Invoices

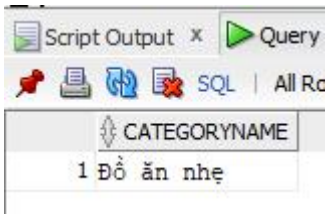
WHERE ItemTotal > (

SELECT AVG(ItemTotal)

FROM Invoices

);</td></tr>
<tr><td>Hình ảnh thực tế</td><td>Script Output x | Query Result x
SQL | All Rows Fetched: 4 in 0,002 seconds
<table><tr><th></th><th>INVOICEID</th><th>EMPLOYEEID</th><th>ORDERTIME</th><th>ITEMID</th><th>QUANTITY</th><th>UNITPRICE</th><th>ITEMTOTAL</th><th>STATUS</th></tr><tr><td>1</td><td>1002</td><td>3</td><td>20-MAY-25</td><td>5</td><td>2</td><td>45000</td><td>90000</td><td>Paid</td></tr><tr><td>2</td><td>1005</td><td>5</td><td>22-MAY-25</td><td>2</td><td>2</td><td>65000</td><td>130000</td><td>Cancelled</td></tr><tr><td>3</td><td>1006</td><td>6</td><td>22-MAY-25</td><td>3</td><td>1</td><td>69000</td><td>69000</td><td>Paid</td></tr><tr><td>4</td><td>1008</td><td>8</td><td>23-MAY-25</td><td>2</td><td>1</td><td>65000</td><td>65000</td><td>Paid</td></tr></table></td></tr>
</table>

Bảng 2. 15: Truy vấn lồng nhau 5

Chức năng 5: Danh mục không có sản phẩm nào	
Câu lệnh	SELECT CategoryName FROM Categories WHERE CategoryID NOT IN (SELECT DISTINCT CategoryID FROM Product);
Hình ảnh thực tế	

Bảng 2. 16: Truy vấn lồng nhau 6

Chức năng 6: Sản phẩm có giá thấp nhất trong từng danh mục	
Câu lệnh	SELECT ProductName FROM Product p WHERE ProductPrice = (SELECT MIN(ProductPrice) FROM Product WHERE CategoryID = p.CategoryID);

Hình ảnh thực tế	Script Output x Query Resu SQL | All Rows F PRODUCTNAME 1 Phin Đen Đá 2 Trà Xanh Đậu Đỏ 3 Trà Thạch Vải 4 Trà Thạch Đào 5 Freeze Trà Xanh 6 Freeze Sô-Cô-La 7 Freeze Cookies 1 8 Phindi Hạnh Nhân 9 PhinDi Kem Sữa 10 Phindi Choco 11 Chanh Dây Đá Viên

- **Truy vấn có nhóm (GROUP BY, HAVING,...)**

Bảng 2. 17: Truy vấn có nhóm 1

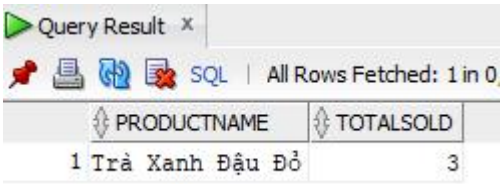
Chức năng 1: Thống kê tổng doanh thu theo từng nhân viên (chỉ tính đơn đã thanh toán)	
Câu lệnh	SELECT e.FullName, SUM(i.ItemTotal) AS TotalRevenueFROM Invoices iJOIN Employees e ON i.EmployeeID = e.EmployeeIDWHERE i.Status = 'Paid'GROUP BY e.FullNameORDER BY TotalRevenue DESC;

Hình ảnh thực tế	Query Result All Rows Fetched: 6 in 0,002 | | FULLNAME | TOTALREVENUE | | 1 | Bùi Thị H | 165000 | | 2 | Hoàng Ngọc Thành | 149000 | | 3 | Đỗ Thị F | 114000 | | 4 | Vũ Văn I | 100000 | | 5 | Nguyễn Trí Dũng | 45000 | | 6 | Nguyễn Văn Duy | 45000 |

Bảng 2. 18: Truy vấn có nhóm 2

Chức năng 2: Doanh thu theo từng loại danh mục sản phẩm (chỉ tính đơn đã thanh toán)	
Câu lệnh	SELECT c.CategoryName, SUM(i.ItemTotal) AS RevenueByCategoryFROM Invoices iJOIN Product p ON i.ItemID = p.ProductIDJOIN Categories c ON p.CategoryID = c.CategoryIDWHERE i.Status = 'Paid'GROUP BY c.CategoryNameORDER BY RevenueByCategory DESC;
Hình ảnh thực tế	Query Result All Rows Fetched: 4 in 0,002 se | | CATEGORYNAME | REVENUEBYCATEGORY | | 1 | Trà | 239000 | | 2 | Cà phê | 179000 | | 3 | Phindi | 145000 | | 4 | Freeze | 55000 |

Bảng 2. 19: Truy vấn có nhóm 3

Chức năng 3: Số lượng sản phẩm đã bán ra theo tên sản phẩm (chỉ Paid)	
Câu lệnh	SELECT p.ProductName, SUM(i.Quantity) AS TotalSoldFROM Invoices iJOIN Product p ON i.ItemID = p.ProductIDWHERE i.Status = 'Paid'GROUP BY p.ProductNameHAVING SUM(i.Quantity) >= 2 -- Lọc ra những sản phẩm bán từ 2 cái trở lênORDER BY TotalSold DESC;
Hình ảnh thực tế	

Bảng 2. 20: Truy vấn có nhóm 4

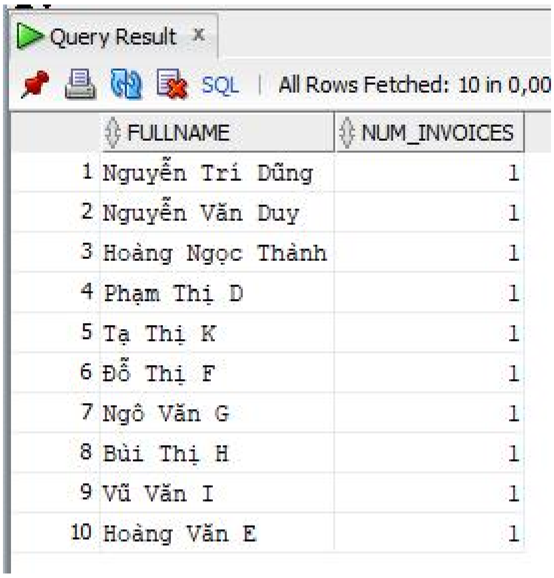
Chức năng 4: Thống kê tổng tiền hóa đơn theo trạng thái đơn hàng (Paid, Unpaid, Cancelled)	
Câu lệnh	SELECT i.Status, SUM(i.ItemTotal) AS TotalAmountFROM Invoices iGROUP BY i.StatusORDER BY

	TotalAmount DESC;
Hình ảnh thực tế	Query Result SQL | All Rows Fetched STATUS, TOTALAMOUNT 1 Paid 618000 2 Cancelled 234000 3 Unpaid 210000

➢ **Dùng hàm tổng hợp (SUM, COUNT, AVG,...)**

Bảng 2. 21: Hàm tổng hợp 1

Chức năng 1: Số lượng đơn hàng đã xử lý bởi từng nhân viên	
Câu lệnh	SELECT e.FullName, COUNT(DISTINCT i.InvoiceID) AS Num_Invoices FROM Invoices i JOIN Employees e ON i.EmployeeID = e.EmployeeID GROUP BY e.FullName ORDER BY Num_Invoices DESC;

Hình ảnh thực tế	

	FULLNAME	NUM_INVOICES
1	Nguyễn Trí Dũng	1
2	Nguyễn Văn Duy	1
3	Hoàng Ngọc Thành	1
4	Phạm Thị D	1
5	Tạ Thị K	1
6	Đỗ Thị F	1
7	Ngô Văn G	1
8	Bùi Thị H	1
9	Vũ Văn I	1
10	Hoàng Văn E	1

Bảng 2. 22: Hàm tổng hợp 2

Chức năng 2: Giá trung bình của sản phẩm theo từng danh mục	
Câu lệnh	SELECT c.CategoryName, ROUND(AVG(p.ProductPrice), 2) AS Avg_Price FROM Product p JOIN Categories c ON p.CategoryID = c.CategoryID GROUP BY c.CategoryName ORDER BY Avg_Price DESC;

Hình ảnh thực tế	Query Result SQL | All Rows Fetched: 5 in CATEGORYNAME, AVG_PRICE 1 Freeze 55000 2 Cà phê 52000 3 Trà 48500 4 Phindi 47500 5 Thức Uống Khác 38000

Bảng 2. 23: Hàm tổng hợp 3

Chức năng 3: Sản phẩm có giá thấp nhất và cao nhất	
Câu lệnh	SELECT MIN(ProductPrice) AS Min_Price, MAX(ProductPrice) AS Max_Price FROM Product;
Hình ảnh thực tế	Query Result SQL | All Rows Fetche MIN_PRICE, MAX_PRICE 1 29000 69000

2.4. Lập trình PL/SQL

- Views: Truy vấn ảo để hiển thị dữ liệu tổng hợp.

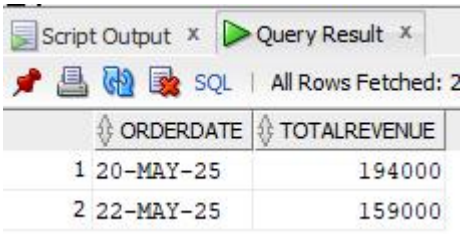
Bảng 2. 24: Hàm views 1

View 1: Tổng số hóa đơn và tổng doanh thu mỗi nhân viên	
Câu lệnh	CREATE OR REPLACE VIEW View_EmployeeSales AS SELECT e.EmployeeID,

<table>
<tr><td></td><td>e.FullName,
COUNT(DISTINCT i.InvoiceID) AS TotalInvoices,
SUM(i.ItemTotal) AS TotalRevenue
FROM Employees e
JOIN Invoices i ON e.EmployeeID = i.EmployeeID
WHERE i.Status = 'Paid'
GROUP BY e.EmployeeID, e.FullName;</td></tr>
<tr><td>Cách sử dụng</td><td>SELECT * FROM View_EmployeeSales ORDER BY TotalRevenue DESC;</td></tr>
<tr><td>Hình ảnh thực tế</td><td>Script Output × Query Result ×
SQL | All Rows Fetched: 6 in 0,009 seconds
<table><tr><th></th><th>EMPLOYEEID</th><th>FULLNAME</th><th>TOTALINVOICES</th><th>TOTALREVENUE</th></tr><tr><td>1</td><td>8</td><td>Bùi Thị H</td><td>1</td><td>165000</td></tr><tr><td>2</td><td>3</td><td>Hoàng Ngọc Thành</td><td>1</td><td>149000</td></tr><tr><td>3</td><td>6</td><td>Đỗ Thị F</td><td>1</td><td>114000</td></tr><tr><td>4</td><td>9</td><td>Vũ Văn I</td><td>1</td><td>100000</td></tr><tr><td>5</td><td>1</td><td>Nguyễn Trí Dũng</td><td>1</td><td>45000</td></tr><tr><td>6</td><td>2</td><td>Nguyễn Văn Duy</td><td>1</td><td>45000</td></tr></table></td></tr>
</table>

Bảng 2. 25: Hàm views 2

<table>
<tr><th colspan="2">View 2: Doanh thu theo ngày</th></tr>
<tr><td>Câu lệnh</td><td>CREATE OR REPLACE VIEW View_DailyRevenue AS
SELECT
TRUNC(OrderTime) AS OrderDate,
SUM(ItemTotal) AS TotalRevenue
FROM Invoices
WHERE Status = 'Paid'</td></tr>
</table>

	GROUP BY TRUNC(OrderTime) ORDER BY OrderDate;
Cách sử dụng	SELECT * FROM View_DailyRevenue WHERE OrderDate BETWEEN TO_DATE('2025-05-20', 'YYYY-MM-DD') AND TO_DATE('2025-05-22', 'YYYY-MM-DD');
Hình ảnh thực tế	

Bảng 2. 26: Hàm views 3

View 3: Sản phẩm bán chạy (top sản phẩm theo số lượng)	

Câu lệnh	CREATE OR REPLACE VIEW View_TopSellingProducts AS
	SELECT
	  p.ProductName,
	  SUM(i.Quantity) AS TotalSold,
	  SUM(i.ItemTotal) AS Revenue
	FROM Product p
	JOIN Invoices i ON p.ProductID = i.ItemID
	WHERE i.Status = 'Paid'
	GROUP BY p.ProductName
	ORDER BY TotalSold DESC;

Cách sử dụng	SELECT * FROM View_TopSellingProducts WHERE ROWNUM <= 5;

Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 5 in 0,004 seconds PRODUCTNAME, TOTALSOLD, REVENUE 1 Trà Xanh Đậu Đỏ, 3, 135000 2 Americano, 1, 45000 3 Cappuccino, 1, 65000 4 Caramel Macchiato, 1, 69000 5 Trà Ngọc Trai Dâu Tằm, 1, 59000

Bảng 2. 27: Hàm views 4

View 4: Tổng doanh thu theo danh mục sản phẩm	
Câu lệnh	CREATE OR REPLACE VIEW View_CategoryRevenue AS SELECT c.CategoryName, SUM(i.ItemTotal) AS TotalRevenue FROM Categories c JOIN Product p ON c.CategoryID = p.CategoryID JOIN Invoices i ON p.ProductID = i.ItemID WHERE i.Status = 'Paid' GROUP BY c.CategoryName;
Cách sử dụng	SELECT * FROM View_CategoryRevenue ORDER BY TotalRevenue DESC;
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 4 in 0, CATEGORYNAME, TOTALREVENUE 1 Trà, 239000 2 Cà phê, 179000 3 Phindi, 145000 4 Freeze, 55000

Bảng 2. 28: Hàm views 5

View 5: Hóa đơn chi tiết (dễ xem hơn JOIN nhiều bảng)	
Câu lệnh	CREATE OR REPLACE VIEW View_InvoiceDetails AS SELECT i.InvoiceID, e.FullName AS EmployeeName, i.OrderTime, p.ProductName, i.Quantity, i.UnitPrice, i.ItemTotal, i.Status FROM Invoices i JOIN Employees e ON i.EmployeeID = e.EmployeeID JOIN Product p ON i.ItemID = p.ProductID;
Cách sử dụng	SELECT * FROM View_InvoiceDetails WHERE InvoiceID = 1002;
Hình ảnh thực tế	Script Output × Query Result × SQL | All Rows Fetched: 2 in 0,005 seconds INVOICEID, EMPLOYEENAME, ORDERTIME, PRODUCTNAME, QUANTITY, UNITPRICE, ITEMTOTAL, STATUS 1 1002 Hoàng Ngọc Thành 20-MAY-25 Trà Xanh Đậu Đỏ 2 45000 90000 Paid 2 1002 Hoàng Ngọc Thành 20-MAY-25 Trà Ngọc Trai Dâu Tằm 1 59000 59000 Paid

Bảng 2. 29: Hàm views 6

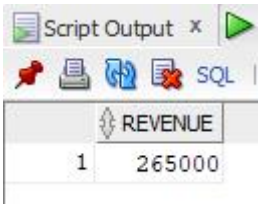
View 6: Thống kê trạng thái hóa đơn (Paid, Unpaid, Cancelled)	
Câu lệnh	CREATE OR REPLACE VIEW View_InvoiceStatusSummary AS

	SELECT Status, COUNT(DISTINCT InvoiceID) AS InvoiceCount, SUM(ItemTotal) AS TotalAmount FROM Invoices GROUP BY Status;
Cách sử dụng	SELECT * FROM View_InvoiceStatusSummary;
Hình ảnh thực tế	Script Output x Query Result x SQL | All Rows Fetched: 3 in 0,008 seconds 1 Paid 6 618000 2 Unpaid 2 210000 3 Cancelled 2 234000

- **Functions: Hàm trả về kết quả.**

Bảng 2. 30: Hàm 1

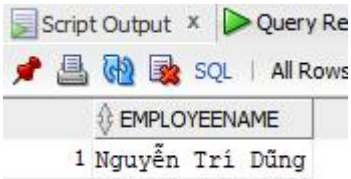
Hàm 1: Hàm GetTotalRevenueByDate Trả về tổng doanh thu trong một ngày cụ thể	
Câu lệnh	CREATE OR REPLACE FUNCTION GetTotalRevenueByDate(p_date IN DATE) RETURN NUMBER IS v_total NUMBER; BEGIN SELECT NVL(SUM(ItemTotal), 0) INTO v_total

	FROM Invoices WHERE TRUNC(OrderTime) = TRUNC(p_date) AND Status = 'Paid'; RETURN v_total; END;
Cách sử dụng	SELECT GetTotalRevenueByDate(TO_DATE('2025-05-23', 'YYYY-MM-DD')) AS Revenue FROM dual;
Hình ảnh thực tế	

Bảng 2. 31: Hàm 2

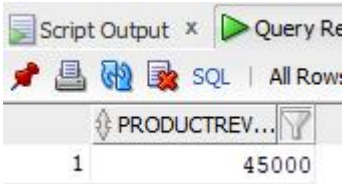
Hàm 2: Hàm GetEmployeeFullName Trả về tên đầy đủ của nhân viên theo mã nhân viên	

Câu lệnh	CREATE OR REPLACE FUNCTION GetEmployeeFullName(p_empID IN NUMBER)
	RETURN VARCHAR2
	IS
	v_name VARCHAR2(100);
	BEGIN
	SELECT FullName
	INTO v_name
	FROM Employees
	WHERE EmployeeID = p_empID;

	RETURN v_name; EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 'Unknown Employee'; END;
Cách sử dụng	SELECT GetEmployeeFullName(1) AS EmployeeName FROM dual;
Hình ảnh thực tế	

Bảng 2. 32: Hàm 3

Hàm 3: Hàm GetProductRevenue Tính tổng doanh thu của một sản phẩm cụ thể theo mã sản phẩm	
Câu lệnh	CREATE OR REPLACE FUNCTION GetProductRevenue(p_productID IN NUMBER) RETURN NUMBER IS v_revenue NUMBER; BEGIN SELECT NVL(SUM(ItemTotal), 0) INTO v_revenue FROM Invoices WHERE ItemID = p_productID

	AND Status = 'Paid'; RETURN v_revenue; END;
Cách sử dụng	SELECT GetProductRevenue(1) AS ProductRevenue FROM dual;
Hình ảnh thực tế	

Bảng 2. 33: Hàm 4

Hàm 4: Hàm GetInvoiceStatus(p_invoiceID) Trả về trạng thái (Status) của một hóa đơn theo mã hóa đơn	

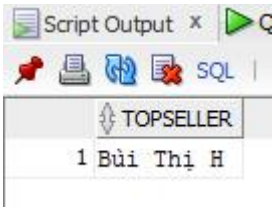
Câu lệnh	CREATE OR REPLACE FUNCTION GetInvoiceStatus(p_invoiceID IN NUMBER)
	RETURN VARCHAR2
	IS
	v_status VARCHAR2(20);
	BEGIN
	SELECT Status
	INTO v_status
	FROM Invoices
	WHERE InvoiceID = p_invoiceID;
	RETURN v_status;
	EXCEPTION

	WHEN NO_DATA_FOUND THEN RETURN 'Invoice Not Found'; END;
Cách sử dụng	SELECT GetInvoiceStatus(1001) AS Status FROM dual;
Hình ảnh thực tế	

Bảng 2. 34: Hàm 5

Hàm 5: Hàm GetTopEmployeeByRevenue(p_date) Trả về tên nhân viên có doanh thu cao nhất trong ngày	

Câu lệnh	CREATE OR REPLACE FUNCTION GetTopEmployeeByRevenue(p_date IN DATE)
	RETURN VARCHAR2
	IS
	v_name VARCHAR2(100);
	BEGIN
	SELECT e.FullName
	INTO v_name
	FROM (
	SELECT EmployeeID, SUM(ItemTotal) AS TotalRevenue
	FROM Invoices
	WHERE TRUNC(OrderTime) = TRUNC(p_date)

	AND Status = 'Paid' GROUP BY EmployeeID ORDER BY TotalRevenue DESC FETCH FIRST 1 ROWS ONLY) top_emp JOIN Employees e ON e.EmployeeID = top_emp.EmployeeID; RETURN v_name; EXCEPTION WHEN NO_DATA_FOUND THEN RETURN 'No Employee Found'; END;
Cách sử dụng	SELECT GetTopEmployeeByRevenue(TO_DATE('2025-05-23', 'YYYY-MM-DD')) AS TopSeller FROM dual;
Hình ảnh thực tế	

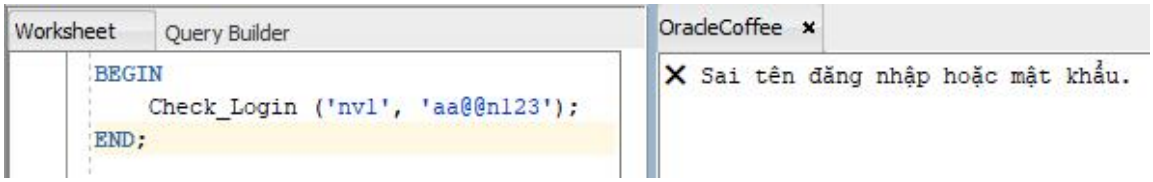
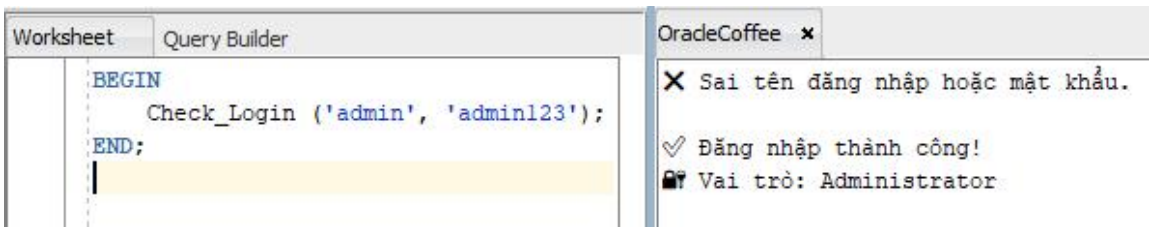
- **Procedures: Thủ tục xử lý logic nghiệp vụ.**

Bảng 2. 35: Procedures Check tài khoản đăng nhập

Câu lệnh	CREATE OR REPLACE PROCEDURE Check_Login (p_Username IN VARCHAR2, p_Password IN VARCHAR2) IS

	v_emp_id NUMBER;
	v_role VARCHAR2(50);
	BEGIN
	-- Kiểm tra username và password
	SELECT EmployeeID INTO v_emp_id
	FROM Accounts
	WHERE LOWER(Username) = LOWER(p_Username)
	AND Password = p_Password;
	-- Lấy Role tương ứng từ Employees
	SELECT Role INTO v_role
	FROM Employees
	WHERE EmployeeID = v_emp_id;
	-- In kết quả
	DBMS_OUTPUT.PUT_LINE('✅ Đăng nhập thành công!');
	DBMS_OUTPUT.PUT_LINE(' Vai trò: ' || NVL(v_role, 'Chưa xác định'));
	EXCEPTION
	WHEN NO_DATA_FOUND THEN
	DBMS_OUTPUT.PUT_LINE('❌ Sai tên đăng nhập hoặc mật khẩu.');
	WHEN OTHERS THEN
	DBMS_OUTPUT.PUT_LINE('❌ Lỗi hệ thống: ' || SQLERRM);
	END;

Cách sử dụng	BEGIN
	Check_Login ('admssin', 'admin123');
	END;

Hình ảnh thực tế	Đây là hình ảnh thông báo đăng nhập lỗi:  Đây là hình ảnh thông báo đăng nhập thành công: 

Bảng 2. 36: Procedures Tạo tài khoản nhân viên

Câu lệnh	CREATE OR REPLACE PROCEDURE Create_Account (p_Username IN VARCHAR2, p_Password IN VARCHAR2, p_Email IN VARCHAR2, p_Phone IN VARCHAR2) IS v_email_count NUMBER; v_phone_count NUMBER; v_emp_id NUMBER; BEGIN -- Kiểm tra trùng Email SELECT COUNT(*) INTO v_email_count

```
    FROM Employees
    WHERE LOWER(Email) = LOWER(p_Email);
    IF v_email_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20001, '❌ Email đã tồn tại trong bảng Employees.');
    END IF;
    -- Kiểm tra trùng Số điện thoại
    SELECT COUNT(*) INTO v_phone_count
    FROM Employees
    WHERE Phone = p_Phone;
    IF v_phone_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, '❌ Số điện thoại đã tồn tại trong bảng Employees.');
    END IF;
    -- Thêm nhân viên mới
    INSERT INTO Employees (
        FullName, Role, Dob, Phone, Email, HireDate, Status
    ) VALUES (
        'Guest', 'Employee', NULL, p_Phone, p_Email, SYSDATE, 'Active'
    )
    RETURNING EmployeeID INTO v_emp_id;
    -- Thêm tài khoản gắn với EmployeeID vừa tạo
    INSERT INTO Accounts (Username, Password, EmployeeID)
    VALUES (p_Username, p_Password, v_emp_id);
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('✅ Tạo tài khoản thành công. EmployeeID = '
```

<table>
<tr><td></td><td>

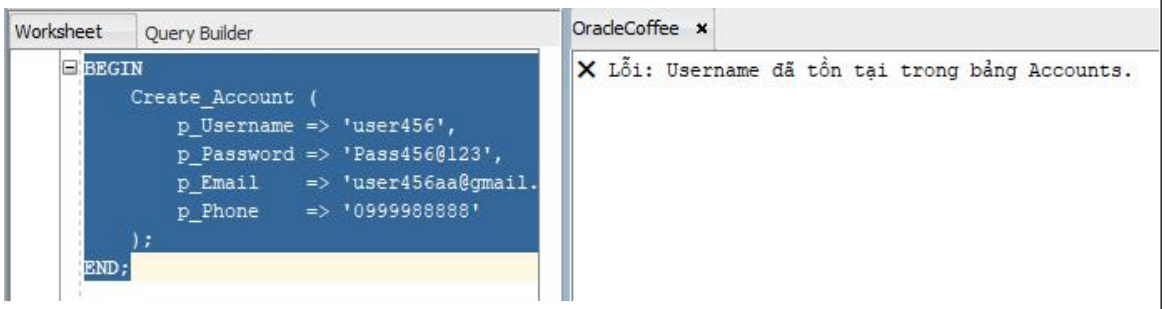
```
|| v_emp_id);
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('✕ Lỗi: Username đã tồn tại trong bảng Accounts.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('✕ Lỗi không xác định: ' || SQLERRM);
END;
```

</td></tr>
<tr><td>Cách sử dụng</td><td>

```
BEGIN
    Create_Account (
        p_Username => 'user456',
        p_Password => 'Pass456@123',
        p_Email    => 'user456aa@gmail.com',
        p_Phone    => '0999988888'
    );
END;
```

</td></tr>
<tr><td>Hình ảnh thực tế</td><td>

Đây là hình ảnh tạo tài khoản lỗi:



Đây là hình ảnh tạo tài khoản thành công:

</td></tr>
</table>

	Worksheet Query Builder
	BEGIN
	Create_Account (
	p_Username => 'titn2022',
	p_Password => 'Paadsv#3',
	p_Email => 'ugg6aa@gmail.com
	p_Phone => '0991598888'
	);
	END;
	
	OracleCoffee
	✕ Lỗi: Username đã tồn tại trong bảng Accounts.
	✓ Tạo tài khoản thành công. EmployeeID = 22

Bảng 2. 37: Procedures Đổi mật khẩu của account

Câu lệnh	CREATE OR REPLACE PROCEDURE Update_password (
	p_username IN VARCHAR2,
	p_mat_khau_cu IN VARCHAR2,
	p_mat_khau_moi IN VARCHAR2
	) AS
	v_count NUMBER;
	BEGIN
	-- Kiểm tra username và mật khẩu cũ có khớp không
	SELECT COUNT(*) INTO v_count
	FROM Accounts
	WHERE LOWER(Username) = LOWER(p_username)
	AND Password = p_mat_khau_cu;
	IF v_count = 0 THEN
	DBMS_OUTPUT.PUT_LINE('✕ Sai tên đăng nhập hoặc mật khẩu cũ.');
	ELSE
	UPDATE Accounts
	SET Password = p_mat_khau_moi

	WHERE LOWER(Username) = LOWER(p_username);
	DBMS_OUTPUT.PUT_LINE('✅ Đổi mật khẩu thành công cho tài khoản: ' || p_username);
	END IF;
	END;

Cách sử dụng	BEGIN
	Update_password('user456', 'Pass456@123', 'EautDung2022@');
	END;

Hình ảnh thực tế	Đây là hình ảnh đổi mật khẩu lỗi: Đây là hình ảnh đổi mật khẩu thành công:

Bảng 2. 38: Procedures Thêm danh mục mới

Câu lệnh	CREATE OR REPLACE PROCEDURE Add_Categories (p_ten_danh_muc IN VARCHAR2) AS v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count FROM Categories

	WHERE LOWER(CategoryName) = LOWER(p_ten_danh_muc);
	IF v_count > 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Tên danh mục đã tồn tại.');
	ELSE
	INSERT INTO Categories (CategoryName)
	VALUES (p_ten_danh_muc);
	DBMS_OUTPUT.PUT_LINE('✅ Thêm danh mục thành công: ' ‖ p_ten_danh_muc);
	END IF;
	END;

Cách sử dụng	BEGIN
	Add_Categories('Sản phẩm test');
	END;

Hình ảnh thực tế	

Bảng 2. 39: Procedures Cập nhật thông tin danh mục

Câu lệnh	CREATE OR REPLACE PROCEDURE Update_Categories (p_id IN NUMBER, p_ten_moi IN VARCHAR2) AS v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count

<table>
<tr><td></td><td>
FROM Categories
WHERE LOWER(CategoryName) = LOWER(p_ten_moi)
AND CategoryID != p_id;
IF v_count > 0 THEN
DBMS_OUTPUT.PUT_LINE('✕ Tên danh mục mới đã tồn tại.');
ELSE
UPDATE Categories
SET CategoryName = p_ten_moi
WHERE CategoryID = p_id;
IF SQL%ROWCOUNT = 0 THEN
DBMS_OUTPUT.PUT_LINE('✕ Không tìm thấy danh mục để sửa.');
ELSE
DBMS_OUTPUT.PUT_LINE('✓ Cập nhật danh mục thành công.');
END IF;
END IF;
END;
</td></tr>
<tr><td>Cách sử dụng</td><td>
BEGIN
Update_Categories(21, 'Máy tính bảng');
END;
</td></tr>
<tr><td>Hình ảnh thực tế</td><td>

</td></tr>
</table>

Bảng 2. 40: Procedures Xóa danh mục

Câu lệnh	CREATE OR REPLACE PROCEDURE Delete_Categories (
	p_id IN NUMBER
	) AS
	BEGIN
	DELETE FROM Categories
	WHERE CategoryID = p_id;
	IF SQL%ROWCOUNT = 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Không tìm thấy danh mục để xóa.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✅ Xóa danh mục thành công.');
	END IF;
	END;

Cách sử dụng	BEGIN
	Delete_Categories(21);
	END;

Hình ảnh thực tế	

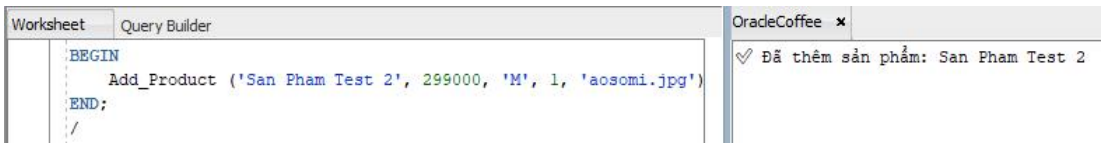
Bảng 2. 41: Procedures Tạo sản phẩm mới

Câu lệnh	CREATE OR REPLACE PROCEDURE Add_Product (p_ten IN VARCHAR2, p_gia IN NUMBER,

```
    p_size IN VARCHAR2,
    p_category_id IN NUMBER,
    p_image IN VARCHAR2
) AS
    v_count NUMBER;
    v_cat_exists NUMBER;
BEGIN
    -- Kiểm tra tên sản phẩm có trùng không
    SELECT COUNT(*) INTO v_count
    FROM Product
    WHERE LOWER(ProductName) = LOWER(p_ten);
    IF v_count > 0 THEN
        DBMS_OUTPUT.PUT_LINE('❌ Tên sản phẩm đã tồn tại.');
        RETURN;
    END IF;
    -- Kiểm tra danh mục có tồn tại không
    SELECT COUNT(*) INTO v_cat_exists
    FROM Categories
    WHERE CategoryID = p_category_id;
    IF v_cat_exists = 0 THEN
        DBMS_OUTPUT.PUT_LINE('❌ Danh mục không tồn tại.');
        RETURN;
    END IF;
    -- Thêm sản phẩm
        INSERT INTO Product (ProductName, ProductPrice, ProductSize,
```

	CategoryID, ProductImage) VALUES (p_ten, p_gia, p_size, p_category_id, p_image); DBMS_OUTPUT.PUT_LINE('✅ Đã thêm sản phẩm: ' || p_ten); END;

Cách sử dụng	BEGIN
	Add_Product ('San Pham Test 1', 299000, 'M', 1, 'aosomi.jpg');
	END;

Hình ảnh thực tế	

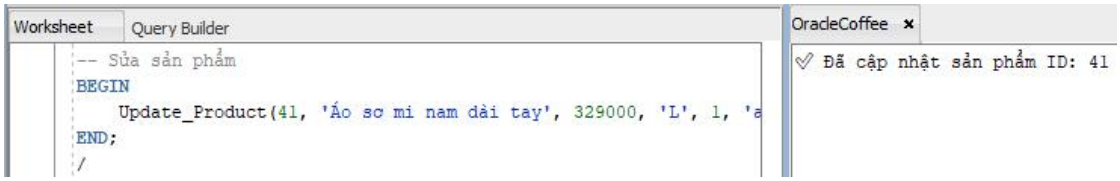
Bảng 2. 42: Procedures Cập nhật sản phẩm

Câu lệnh	CREATE OR REPLACE PROCEDURE Update_Product (p_id IN NUMBER, p_ten_moi IN VARCHAR2, p_gia_moi IN NUMBER, p_size_moi IN VARCHAR2, p_category_id IN NUMBER, p_image_moi IN VARCHAR2) AS v_count NUMBER; v_cat_exists NUMBER; BEGIN -- Kiểm tra trùng tên mới với sản phẩm khác

```
SELECT COUNT(*) INTO v_count
FROM Product
WHERE LOWER(ProductName) = LOWER(p_ten_moi)
  AND ProductID != p_id;
IF v_count > 0 THEN
    DBMS_OUTPUT.PUT_LINE('❌ Tên sản phẩm mới đã tồn tại.');
    RETURN;
END IF;
-- Kiểm tra danh mục tồn tại
SELECT COUNT(*) INTO v_cat_exists
FROM Categories
WHERE CategoryID = p_category_id;
IF v_cat_exists = 0 THEN
    DBMS_OUTPUT.PUT_LINE('❌ Danh mục không tồn tại.');
    RETURN;
END IF;
-- Cập nhật
UPDATE Product
SET ProductName = p_ten_moi,
    ProductPrice = p_gia_moi,
    ProductSize = p_size_moi,
    CategoryID = p_category_id,
    ProductImage = p_image_moi
WHERE ProductID = p_id;
IF SQL%ROWCOUNT = 0 THEN
```

	DBMS_OUTPUT.PUT_LINE('❌ Không tìm thấy sản phẩm để cập nhật.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✅ Đã cập nhật sản phẩm ID: ' || p_id);
	END IF;
	END;

Cách sử dụng	BEGIN
	Update_Product(21, 'Áo sơ mi nam dài tay', 329000, 'L', 1, 'aosomi_new.jpg');
	END;

Hình ảnh thực tế	

Bảng 2. 43: Procedures Xóa sản phẩm

Câu lệnh	REATE OR REPLACE PROCEDURE Delete_Product (
	p_id IN NUMBER
	) AS
	BEGIN
	DELETE FROM Product
	WHERE ProductID = p_id;
	IF SQL%ROWCOUNT = 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Không tìm thấy sản phẩm để xóa.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✅ Đã xóa sản phẩm ID: ' || p_id);

	END IF; END;

Cách sử dụng	BEGIN
	Delete_Product(21);
	END;

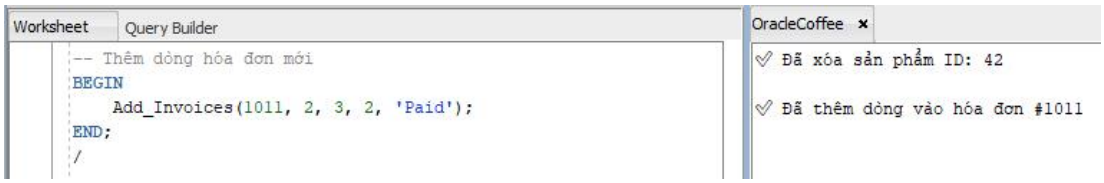
Hình ảnh thực tế	

Bảng 2. 44: Procedures Tạo hóa đơn mới

Câu lệnh	CREATE OR REPLACE PROCEDURE Add_Invoices (p_invoice_id IN NUMBER, p_employee_id IN NUMBER, p_item_id IN NUMBER, p_quantity IN NUMBER DEFAULT 1, p_status IN VARCHAR2 DEFAULT 'Unpaid') AS v_price NUMBER; v_emp_exist NUMBER; v_item_exist NUMBER; BEGIN -- Kiểm tra nhân viên tồn tại SELECT COUNT(*) INTO v_emp_exist FROM Employees WHERE EmployeeID = p_employee_id;

	IF v_emp_exist = 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Nhân viên không tồn tại.');
	RETURN;
	END IF;
	-- Kiểm tra sản phẩm tồn tại
	SELECT COUNT(*) INTO v_item_exist FROM Product WHERE ProductID = p_item_id;
	IF v_item_exist = 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Sản phẩm không tồn tại.');
	RETURN;
	END IF;
	-- Lấy đơn giá sản phẩm
	SELECT ProductPrice INTO v_price FROM Product WHERE ProductID = p_item_id;
	-- Thêm hóa đơn
	INSERT INTO Invoices (
	InvoiceID, EmployeeID, OrderTime, ItemID, UnitPrice, Quantity, ItemTotal, Status
	) VALUES (
	p_invoice_id, p_employee_id, SYSDATE, p_item_id, v_price, p_quantity, v_price * p_quantity, p_status
	);
	DBMS_OUTPUT.PUT_LINE('✅ Đã thêm dòng vào hóa đơn #' || p_invoice_id);
	END;

Cách sử	BEGIN

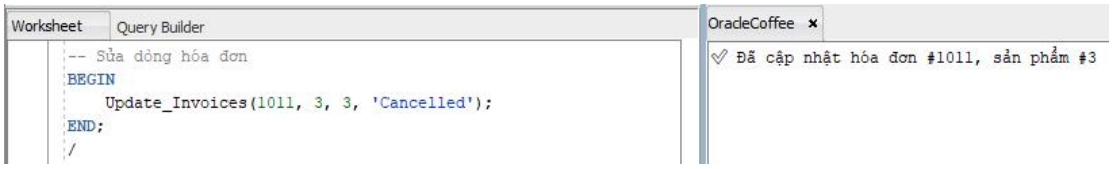
dụng	Add_Invoices(1011, 2, 3, 2, 'Paid'); END;
Hình ảnh thực tế	

Bảng 2. 45: Procedures Cập nhật hóa đơn

Câu lệnh	CREATE OR REPLACE PROCEDURE Update_Invoices (p_invoice_id IN NUMBER, p_item_id IN NUMBER, p_quantity_moi IN NUMBER, p_status_moi IN VARCHAR2) AS v_price NUMBER; BEGIN -- Lấy đơn giá hiện tại SELECT UnitPrice INTO v_price FROM Invoices WHERE InvoiceID = p_invoice_id AND ItemID = p_item_id; -- Cập nhật UPDATE Invoices SET Quantity = p_quantity_moi, ItemTotal = v_price * p_quantity_moi, Status = p_status_moi

	WHERE InvoiceID = p_invoice_id AND ItemID = p_item_id;
	IF SQL%ROWCOUNT = 0 THEN
	DBMS_OUTPUT.PUT_LINE('✕ Không tìm thấy dòng hóa đơn để cập nhật.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✓ Đã cập nhật hóa đơn #' || p_invoice_id || ', sản phẩm #' || p_item_id);
	END IF;
	END;

Cách sử dụng	BEGIN
	Update_Invoices(1011, 3, 3, 'Cancelled');
	END;

Hình ảnh thực tế	

Bảng 2. 46: Procedures Xóa hóa đơn

Câu lệnh	CREATE OR REPLACE PROCEDURE Delete_Invoices (
	p_invoice_id IN NUMBER,
	p_item_id IN NUMBER
	) AS
	BEGIN
	DELETE FROM Invoices
	WHERE InvoiceID = p_invoice_id AND ItemID = p_item_id;
	IF SQL%ROWCOUNT = 0 THEN

	DBMS_OUTPUT.PUT_LINE('✕ Không tìm thấy dòng hóa đơn để xóa.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✓ Đã xóa dòng hóa đơn #' || p_invoice_id || ', sản phẩm #' || p_item_id);
	END IF;
	END;

Cách sử dụng	BEGIN
	Delete_Invoices(1011, 3);
	END;

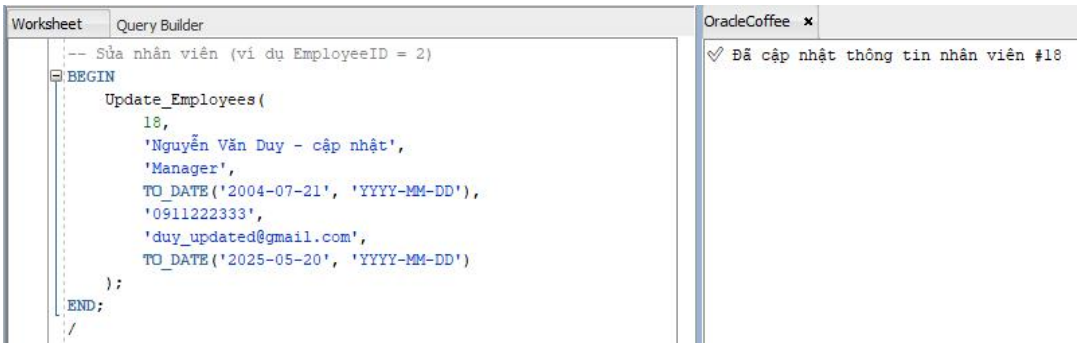
Hình ảnh thực tế	

Bảng 2. 47: Procedures Cập nhật nhân viên

Câu lệnh	CREATE OR REPLACE PROCEDURE Update_Employees (p_employee_id IN NUMBER, p_fullname IN VARCHAR2, p_role IN VARCHAR2, p_dob IN DATE, p_phone IN VARCHAR2, p_email IN VARCHAR2, p_hire_date IN DATE) AS BEGIN

	UPDATE Employees
	SET FullName = p_fullname,
	Role = p_role,
	Dob = p_dob,
	Phone = p_phone,
	Email = p_email,
	HireDate = p_hire_date
	WHERE EmployeeID = p_employee_id;
	IF SQL%ROWCOUNT = 0 THEN
	DBMS_OUTPUT.PUT_LINE('❌ Không tìm thấy nhân viên cần sửa.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✅ Đã cập nhật thông tin nhân viên #' || p_employee_id);
	END IF;
	END;

Cách sử dụng	BEGIN Update_Employees(18, 'Nguyễn Văn Duy - cập nhật', 'Manager', TO_DATE('2004-07-21', 'YYYY-MM-DD'), '0911222333', 'duy_updated@gmail.com', TO_DATE('2025-05-20', 'YYYY-MM-DD')

	); END;
Hình ảnh thực tế	

Bảng 2. 48: Procedures Khoá nhân viên

Câu lệnh	CREATE OR REPLACE PROCEDURE Lock_Employees (
	p_employee_id IN NUMBER
	) AS
	BEGIN
	UPDATE Employees
	SET Status = 'Banned'
	WHERE EmployeeID = p_employee_id;
	IF SQL%ROWCOUNT = 0 THEN
	DBMS_OUTPUT.PUT_LINE('✕ Không tìm thấy nhân viên để khóa.');
	ELSE
	DBMS_OUTPUT.PUT_LINE('✓ Đã khóa nhân viên #' || p_employee_id || ' (Status = Banned)');
	END IF;
	END;

Cách sử dụng	BEGIN
	Lock_Employees(18);
	END;

Hình ảnh thực tế	

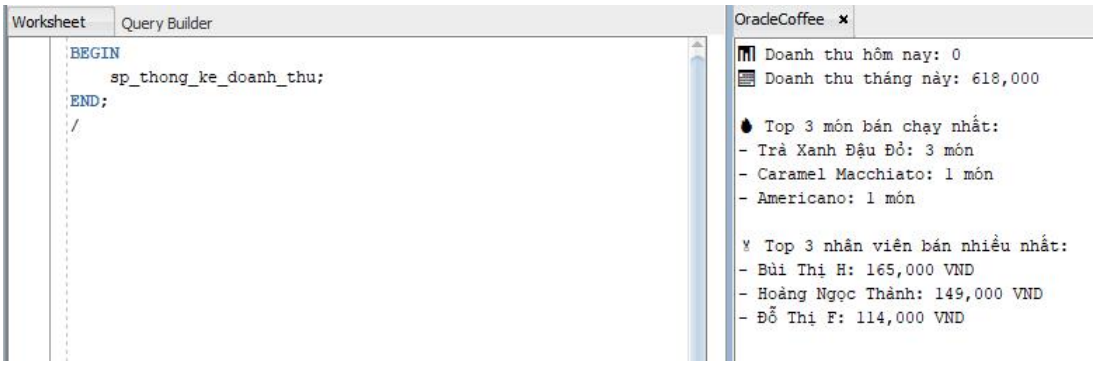
Bảng 2. 49: Procedures thống kê báo cáo

Câu lệnh	CREATE OR REPLACE PROCEDURE sp_thong_ke_doanh_thu IS
	v_doanh_thu_ngay NUMBER := 0;
	v_doanh_thu_thang NUMBER := 0;
	-- Cursor cho 3 món bán chạy nhất
	CURSOR c_top_items IS
	SELECT p.ProductName, SUM(i.Quantity) AS TotalQty
	FROM Invoices i
	JOIN Product p ON i.ItemID = p.ProductID
	WHERE i.Status = 'Paid'
	GROUP BY p.ProductName
	ORDER BY TotalQty DESC
	FETCH FIRST 3 ROWS ONLY;
	-- Cursor cho 3 nhân viên bán nhiều nhất (theo doanh thu)
	CURSOR c_top_employees IS
	SELECT e.FullName, SUM(i.ItemTotal) AS TotalSales
	FROM Invoices i

```
        JOIN Employees e ON i.EmployeeID = e.EmployeeID
        WHERE i.Status = 'Paid'
        GROUP BY e.FullName
        ORDER BY TotalSales DESC
        FETCH FIRST 3 ROWS ONLY;
BEGIN
    -- Doanh thu hôm nay
    SELECT NVL(SUM(ItemTotal), 0)
    INTO v_doanh_thu_ngay
    FROM Invoices
    WHERE Status = 'Paid'
      AND TRUNC(OrderTime) = TRUNC(SYSDATE);
    -- Doanh thu tháng hiện tại
    SELECT NVL(SUM(ItemTotal), 0)
    INTO v_doanh_thu_thang
    FROM Invoices
    WHERE Status = 'Paid'
       AND EXTRACT(MONTH FROM OrderTime) = EXTRACT(MONTH
FROM SYSDATE)
         AND EXTRACT(YEAR FROM OrderTime) = EXTRACT(YEAR
FROM SYSDATE);
    -- Xuất doanh thu
          DBMS_OUTPUT.PUT_LINE('    Doanh thu hôm nay: ' ||
TO_CHAR(v_doanh_thu_ngay, 'FM999,999,990'));
          DBMS_OUTPUT.PUT_LINE('    Doanh thu tháng này: ' ||
TO_CHAR(v_doanh_thu_thang, 'FM999,999,990'));
```

	-- Top món bán chạy
	DBMS_OUTPUT.PUT_LINE(CHR(10) || ' Top 3 món bán chạy nhất:');
	FOR r IN c_top_items LOOP
	  DBMS_OUTPUT.PUT_LINE('- ' || r.ProductName || ': ' || r.TotalQty || ' món');
	END LOOP;
	
	-- Top nhân viên bán nhiều nhất
	DBMS_OUTPUT.PUT_LINE(CHR(10) || ' Top 3 nhân viên bán nhiều nhất:');
	FOR r IN c_top_employees LOOP
	  DBMS_OUTPUT.PUT_LINE('- ' || r.FullName || ': ' || TO_CHAR(r.TotalSales, 'FM999,999,990') || ' VND');
	END LOOP;
	EXCEPTION
	WHEN OTHERS THEN
	  DBMS_OUTPUT.PUT_LINE('Lỗi: ' || SQLERRM);
	END;

Cách sử dụng	BEGIN
	sp_thong_ke_doanh_thu;
	END;

Hình ảnh thực tế	

➢ **Triggers: Kích hoạt tự động khi có sự kiện (INSERT, UPDATE,...)**

Bảng 2. 50: Trigger ghi log khi thêm danh mục

Tạo bảng log	CREATE TABLE Category_Log (LogID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY, CategoryID NUMBER, CategoryName VARCHAR2(100), Status VARCHAR2(10), Action VARCHAR2(20), ActionDate DATE, PerformedBy VARCHAR2(100));
Trigger	CREATE OR REPLACE TRIGGER trg_log_new_category AFTER INSERT ON Categories FOR EACH ROW BEGIN INSERT INTO Category_Log (CategoryID, CategoryName, Status, Action, ActionDate,

<table>
<tr><td></td><td>PerformedBy
) VALUES (
:NEW.CategoryID,
:NEW.CategoryName,
:NEW.Status,
'INSERT',
SYSDATE,
USER
);
END;</td></tr>
<tr><td>Cách sử dụng</td><td>INSERT INTO Categories (CategoryName) VALUES ('Soda');
SELECT * FROM Category_Log ORDER BY ActionDate DESC;</td></tr>
<tr><td>Hình ảnh thực tế</td><td>Script Output × | Query Result ×
SQL | All Rows Fetched: 2 in 0,005 seconds
<table><tr><th></th><th>LOGID</th><th>CATEGORYID</th><th>CATEGORYNAME</th><th>STATUS</th><th>ACTION</th><th>ACTIONDATE</th><th>PERFORMEDBY</th></tr><tr><td>1</td><td>1</td><td>63</td><td>Soda</td><td>(null)</td><td>INSERT</td><td>29-MAY-25</td><td>(null)</td></tr><tr><td>2</td><td>2</td><td>63</td><td>Soda</td><td>Active</td><td>INSERT</td><td>29-MAY-25</td><td>ORACLEUSER</td></tr></table></td></tr>
</table>

Bảng 2. 51: Trigger kiểm tra tên danh mục trùng lặp khi thêm mới

Trigger	CREATE OR REPLACE TRIGGER trg_check_duplicate_category BEFORE INSERT ON Categories FOR EACH ROW DECLARE v_count NUMBER; BEGIN

	SELECT COUNT(*) INTO v_count
	FROM Categories
	WHERE UPPER(CategoryName) = UPPER(:NEW.CategoryName);
	IF v_count > 0 THEN
	RAISE_APPLICATION_ERROR(-20002, 'Tên danh mục đã tồn tại!');
	END IF;
	END;

Cách sử dụng	INSERT INTO Categories(CategoryName) VALUES('Cà phê'); -- Nếu 'Cà phê' đã tồn tại, trigger sẽ chặn và báo lỗi

Hình ảnh thực tế

Worksheet | Query Builder

```oracle
CREATE OR REPLACE TRIGGER trg_check_duplicate_category
BEFORE INSERT ON Categories
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM Categories
    WHERE UPPER(CategoryName) = UPPER(:NEW.CategoryName);

    IF v_count > 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Tên danh mục đã tồn tại!');
    END IF;
END;

INSERT INTO Categories(CategoryName) VALUES('Cà phê');
-- Nếu 'Cà phê' đã tồn tại, trigger sẽ chặn và báo lỗi
```

Script Output | Query Result

Task completed in 0,057 seconds

```
Table ACCOUNTLOG created.


Trigger TRG_LOG_DELETE_ACCOUNT compiled


Trigger TRG_CHECK_DUPLICATE_CATEGORY compiled


Error starting at line : 16 in command -
INSERT INTO Categories(CategoryName) VALUES('Cà phê')
Error at Command Line : 16 Column : 13
Error report -
SQL Error: ORA-20002: Tên danh mục đã tồn tại!
ORA-06512: at "ORACLEUSER.TRG_CHECK_DUPLICATE_CATEGORY", line 9
ORA-04088: error during execution of trigger 'ORACLEUSER.TRG_CHECK_DUPLICATE_CAT

https://docs.oracle.com/error-help/db/ora-20002/

More Details :
https://docs.oracle.com/error-help/db/ora-20002/
https://docs.oracle.com/error-help/db/ora-06512/
https://docs.oracle.com/error-help/db/ora-04088/
```

Bảng 2. 52: Trigger lưu lại thông tin trước khi sửa sản phẩm

Tạo bảng log	CREATE TABLE Product_Update_Log (LogID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY PRIMARY KEY, ProductID NUMBER, OldName VARCHAR2(100), NewName VARCHAR2(100), OldPrice NUMBER(10,2), NewPrice NUMBER(10,2), OldSize VARCHAR2(50), NewSize VARCHAR2(50), OldStatus VARCHAR2(10), NewStatus VARCHAR2(10), UpdateDate DATE, UpdatedBy VARCHAR2(100));
Trigger	CREATE OR REPLACE TRIGGER trg_update_product_log AFTER UPDATE ON Product FOR EACH ROW BEGIN INSERT INTO Product_Update_Log (ProductID, OldName, NewName, OldPrice, NewPrice, OldSize, NewSize, OldStatus, NewStatus,

<table>
<tr>
<td></td>
<td>
UpdateDate,
UpdatedBy
) VALUES (
:OLD.ProductID,
:OLD.ProductName, :NEW.ProductName,
:OLD.ProductPrice, :NEW.ProductPrice,
:OLD.ProductSize, :NEW.ProductSize,
:OLD.Status, :NEW.Status,
SYSDATE,
USER
);
END;
</td>
</tr>
<tr>
<td>Cách sử dụng</td>
<td>
UPDATE Product
SET ProductPrice = 60000, ProductName = 'Phin Đen Đá (Mới)', ProductSize = 'M'
WHERE ProductID = 4;
SELECT * FROM Product_Update_Log
ORDER BY UpdateDate DESC;
</td>
</tr>
<tr>
<td>Hình ảnh thực tế</td>
<td>
Script Output × Query Result ×
SQL | All Rows Fetched: 1 in 0,003 seconds
<table>
<tr><th></th><th>LOGID</th><th>PRODUCTID</th><th>OLDNAME</th><th>NEWNAME</th><th>OLDPRICE</th><th>NEWPRICE</th><th>OLDSIZE</th><th>NEWSIZE</th><th>OLDSTATUS</th><th>NEWSTATUS</th><th>UPDATEDATE</th><th>UPDATEDBY</th></tr>
<tr><td>1</td><td>1</td><td>4</td><td>Phin Đen Đá</td><td>Phin Đen Đá (Mới)</td><td>29000</td><td>60000</td><td>S</td><td>M</td><td>Active</td><td>Active</td><td>29-MAY-25</td><td>ORACLEUSER</td></tr>
</table>
</td>
</tr>
</table>

2.5. Quản trị người dùng & sao lưu

2.5.1. Tạo User mới

- CREATE USER coffee_user IDENTIFIED BY coffee123DEFAULT TABLESPACE users TEMPORARY TABLESPACE temp QUOTA 100M ON users;

✅ coffee_user: tên người dùng

✅ coffee123: mật khẩu

✅ users, temp: tablespace mặc định và tạm

✅ QUOTA: dung lượng được phép dùng

2.5.2. Cấp quyền (GRANT)

- Cấp quyền cơ bản: GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE, CREATE TRIGGER TO coffee_user;
- Cấp toàn quyền trên 1 bảng: GRANT SELECT, INSERT, UPDATE, DELETE ON Product TO coffee_user;
- Cấp quyền DBA (toàn quyền): GRANT DBA TO coffee_user;

2.5.3. Thu hồi quyền (REVOKE)

- REVOKE UPDATE ON Product FROM coffee_user;

2.5.4. Xóa User (nếu cần)

- DROP USER coffee_user CASCADE;

2.5.5. Backup bằng expdp (Data Pump Export)

Ví dụ: Backup schema COFFEE_SHOP thành file .dmp

- expdp system/password DIRECTORY=DATA_PUMP_DIR DUMPFILE=coffee_shop.dmp LOGFILE=coffee_shop.log SCHEMAS=COFFEE_SHOP

✅ DIRECTORY: thư mục logic đã khai báo (ví dụ DATA_PUMP_DIR)

✅ DUMPFILE: file xuất ra

✅ SCHEMAS: tên schema cần backup

2.5.6. Phục hồi bằng impdp (Data Pump Import)

- impdp system/password DIRECTORY=DATA_PUMP_DIR DUMPFILE=coffee_shop.dmp LOGFILE=imp_coffee_shop.log REMAP_SCHEMA=COFFEE_SHOP:NEWUSER

✅ Dùng REMAP_SCHEMA nếu bạn muốn phục hồi sang user khác.

2.5.7. Tạo DIRECTORY cho xuất/nhập

- CREATE OR REPLACE DIRECTORY DATA_PUMP_DIR AS 'C:\oracle\backups';GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR TO system;

✅ Đảm bảo thư mục vật lý tồn tại trên server.

2.5.8. Backup bảng cụ thể

- expdp system/password TABLES=Product,DanhMuc DIRECTORY=DATA_PUMP_DIR DUMPFILE=table_backup.dmp

Chương III. Kết luận

Tổng kết các nội dung đã đạt được:

Qua quá trình thực hiện đề tài "Xây dựng cơ sở dữ liệu cho phần mềm quản lý quán cà phê" sử dụng hệ quản trị cơ sở dữ liệu Oracle, nhóm chúng em đã hoàn thành được các nội dung chính như sau:

Phân tích và xây dựng sơ đồ thực thể quan hệ (ERD) thể hiện rõ các thực thể chính trong hệ thống như: Nhân viên, Sản phẩm, Danh mục, Tài khoản, Hóa đơn, Khách hàng,...

Thiết kế cơ sở dữ liệu bao gồm việc xác định các bảng, khóa chính – khóa ngoại, các ràng buộc toàn vẹn dữ liệu nhằm đảm bảo tính logic, chặt chẽ và tối ưu trong quản lý dữ liệu.

Hiện thực các truy vấn và thủ tục thao tác dữ liệu (CRUD) với Oracle SQL thông qua các lệnh: INSERT, UPDATE, DELETE, SELECT,… đảm bảo hỗ trợ đầy đủ cho các chức năng của phần mềm.

Mô tả và triển khai các chức năng nghiệp vụ tiêu biểu như: đăng nhập, quản lý hóa đơn, đổi mật khẩu, thống kê doanh thu, v.v., phục vụ nhu cầu thực tế của một phần mềm quản lý quán cà phê.

Hạn chế còn tồn tại:

Mặc dù đã hoàn thiện cơ sở dữ liệu cơ bản và đảm bảo các chức năng cốt lõi, đề tài vẫn còn một số hạn chế nhất định như:

Chưa xây giao diện người dùng và các chức năng xử lý phía ứng dụng (front-end) chưa được xây dựng đầy đủ để kiểm chứng toàn diện các truy vấn CSDL.

Một số tính năng nâng cao như phân quyền tài khoản, xử lý tình huống lỗi hoặc tính năng sao lưu/phục hồi dữ liệu chưa được tích hợp.

Việc kiểm thử hệ thống mới dừng ở mức cơ bản, chưa có đánh giá hiệu năng trên tập dữ liệu lớn hoặc môi trường thực tế.

Hướng phát triển nếu tiếp tục mở rộng đề tài:

Nếu được tiếp tục mở rộng, đề tài có thể phát triển thêm ở các hướng sau:

Tích hợp giao diện người dùng (web hoặc mobile) để kết nối với cơ sở dữ liệu Oracle, nâng cao trải nghiệm người dùng cuối.

Phân quyền người dùng: Xây dựng hệ thống phân quyền theo vai trò (quản lý, nhân viên, khách hàng), tăng cường tính bảo mật.

Chức năng sao lưu, khôi phục dữ liệu định kỳ nhằm đảm bảo an toàn dữ liệu.

Tối ưu hóa truy vấn và chỉ số (indexing) để cải thiện hiệu năng khi xử lý lượng dữ liệu lớn.

Tích hợp các công nghệ mới như mã QR để đặt hàng nhanh, hoặc sử dụng AI để gợi ý món bán chạy, phân tích hành vi khách hàng.

Tài liệu tham khảo

[1] Faceworks. (n.d.). *Phần mềm quản lý quán cà phê*. https://faceworks.vn/phan-mem-quan-ly-quan-ca-phe/

[2] GeeksforGeeks. (n.d.). *PL/SQL - Introduction*. https://www.geeksforgeeks.org/pl-sql-introduction/

[3] Kamereo. (2023, October 25). *Top 10 phần mềm quản lý quán cafe tốt nhất hiện nay*. https://kamereo.vn/blog/vi/phan-mem-quan-ly-quan-cafe/

[4] KiotViet. (2023, June 13). *Phần mềm quản lý quán cafe, bán cafe tốt nhất*. https://www.kiotviet.net/2023/06/phan-mem-quan-ly-quan-cafe-ban-cafe-tot-nhat.html

[5] Mona Media. (n.d.). *Phần mềm quản lý quán cafe chuyên nghiệp - đơn giản - dễ dùng*. https://mona.solutions/phan-mem-quan-ly-quan-cafe/

[6] Oracle. (n.d.). *Oracle database documentation. Oracle.* https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html

[7] Oracle. (n.d.). *Oracle database technologies.* https://www.oracle.com/database/technologies/

[8] Oracle. (n.d.). *SQL Developer downloads.* https://www.oracle.com/tools/downloads/sqldev-downloads.html

[9] Phần mềm Biz. (2023, March 24). *Cách tiết kiệm chi phí quản lý quán cà phê hiệu quả.* https://phanmembiz.com/blogs/cach-tiet-kiem-chi-phi-quan-ly-quan-ca-phe/

[10] PosApp. (n.d.). *Phần mềm quản lý quán cafe - cà phê hiệu quả nhất hiện nay.* https://posapp.vn/phan-mem-quan-ly-quan-cafe-ca-phe/

[11] Visual Studio Code. (n.d.). *Code editing. Redefined.* https://code.visualstudio.com/

[12] W3Schools. (n.d.). *SQL PL/SQL.* https://www.w3schools.com/sql/sql_plsql.asp

[13] draw.io. (n.d.). *Diagram editor.* https://www.draw.io/