

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ - ĐHQGHN  
VIỆN TRÍ TUỆ NHÂN TẠO

---



CƠ SỞ DỮ LIỆU  
BÁO CÁO BÀI TẬP LỚN

NHÓM LOLI

Hệ thống Cơ sở dữ liệu quản lý cửa hàng điện thoại

Thành viên nhóm: Phạm Nhật Quang    23020413 (trưởng nhóm)  
Bùi Minh Quân    23020415  
Phan Quang Trường    23020443

Hà Nội, 2024

# Lời mở đầu

## Đường dẫn

- GitHub Source code: [Link GitHub](#)

Lời đầu tiên, nhóm tác giả chúng em xin gửi lời cảm ơn tới thầy cô trong Viện Trí tuệ nhân tạo, vì đã tạo điều kiện cho chúng em thử sức với nhiều điều mới mẻ.

Trong thời đại công nghệ hiện nay, nhu cầu sử dụng điện thoại thông minh tại Việt Nam ngày càng tăng cao, trở thành một phần không thể thiếu trong cuộc sống hàng ngày của người dân. Từ công việc, học tập đến giải trí, điện thoại không chỉ là phương tiện liên lạc mà còn là công cụ hỗ trợ đa năng. Trước xu hướng đó, các chuỗi cửa hàng bán lẻ điện thoại như Thế Giới Di Động, FPT Shop hay CellphoneS đã nhanh chóng phát triển mạnh mẽ để đáp ứng nhu cầu này. Những hệ thống này không chỉ cung cấp đa dạng các sản phẩm công nghệ chính hãng, mà còn chú trọng đến dịch vụ hậu mãi và trải nghiệm mua sắm của khách hàng. Sự tiện lợi, uy tín và các chương trình ưu đãi hấp dẫn đã giúp các chuỗi cửa hàng này trở thành điểm đến hàng đầu của người tiêu dùng trên cả nước.

Trong bản báo cáo này, ta sẽ thực hiện theo từng bước sau:

1. Liệt kê nghiệp vụ hệ thống, thực thể trong hệ cơ sở dữ liệu, tạo mô hình ER;
2. Chuyển mô hình ER sang mô hình Quan hệ cùng các quan hệ, phụ thuộc hàm, chuẩn hóa 3NF, vẽ đồ thị;
3. Tạo bảng cùng các ràng buộc, dựa trên mô hình quan hệ trong SQL;
4. Thao tác truy vấn với dữ liệu bên trong cơ sở dữ liệu: Query, Transaction, Trigger, Procedure.

Để bản báo cáo thêm phong phú, hệ thống của chúng em sẽ đáp ứng nhiều nghiệp vụ ở nhiều phần việc khác nhau, phần dữ liệu cũng sẽ được sinh để sao cho đa dạng nhất có thể. Chúng em sẽ dựa trên hệ thống hiện có của Thế Giới Di Động [1], kết hợp một phần với dữ liệu về vị trí cửa hàng thực tế [2].

Trong tài liệu sẽ có ghi thêm tên file tương ứng. Thầy cô có thể xem nội dung file thực tế ở trong repo GitHub.

# Mục lục

<b>Lời mở đầu, đường dẫn</b>	<b>1</b>
<b>1 Mô hình dữ liệu</b>	<b>4</b>
1.1 Nghiệp vụ hệ thống, thực thể, liên kết	4
1.2 Mô hình ER	5
1.3 Mô hình Quan hệ	5
1.3.1 Đặc tả dữ liệu	5
1.3.2 Liên kết	9
1.3.3 Quan hệ, phụ thuộc hàm khi chuẩn hóa về 3NF	9
1.4 Lược đồ mô hình quan hệ	11
<b>2 Khởi tạo cơ sở dữ liệu trên SQL</b>	<b>12</b>
2.1 CREATE TABLE (createDB.sql)	12
2.2 CONSTRAINTS (constraints.sql)	15
2.3 INSERT (insert.sql)	16
<b>3 Thao tác truy vấn trên SQL</b>	<b>21</b>
3.1 QUERY (query.sql)	21
3.1.1 Lấy thông tin kỹ thuật của máy	21
3.1.2 Lấy danh sách tất cả các điện thoại và thông tin khuyến mãi (nếu có)	21
3.1.3 Lấy danh sách các mẫu điện thoại có giá lớn hơn mức giá trung bình và đã bán được trên 50 máy	22
3.1.4 Lấy tổng doanh thu của từng cửa hàng từ bảng orders và order_detail	22
3.1.5 Lấy số lượng điện thoại tồn kho	22
3.1.6 Lấy danh sách điện thoại và rating trung bình tương ứng	23
3.2 TRANSACTION (transaction.sql)	23
3.2.1 Cập nhật thông tin người dùng, in ra lỗi nếu không thành công	23
3.2.2 Cập nhật thông tin sản phẩm, in ra lỗi nếu không thành công	24
3.2.3 Cập nhật thông tin sản phẩm vào đơn hàng khi người dùng chọn mua	25
3.2.4 Cập nhật trạng thái đơn hàng	26
3.2.5 Hủy đơn hàng nếu như khách hàng không còn như cầu mua	26
3.3 TRIGGER (trigger.sql)	27
3.3.1 Tự động cập nhật trạng thái đơn hàng khi đã giao	27
3.3.2 Tự động cập nhật trạng thái sản phẩm khi có đơn đặt hàng	27
3.3.3 Tự động cập nhật trạng thái điện thoại khi đơn hàng bị hủy	28
3.3.4 Tự động kích hoạt bảo hành cho sản phẩm mới mua	28
3.3.5 Tự động cập nhật số lượng đã bán của phone_model	29
3.3.6 Kiểm tra dữ liệu trước khi thêm vào bảng orders	29
3.3.7 Kiểm tra số lượng hàng trước khi thêm vào order_detail	29
3.4 PROCEDURE (procedure.sql)	30
3.4.1 Lọc sản phẩm theo nhà sản xuất	30
3.4.2 Lọc sản phẩm theo giá tiền	30
3.4.3 Lọc sản phẩm theo thông số kỹ thuật	31
3.4.4 Đề xuất sản phẩm bán chạy theo tháng	31
3.4.5 Kiểm tra bảo hành có còn khả dụng	32
3.4.6 Tìm kiếm địa chỉ cửa hàng gần người dùng nhất	32

3.4.7	Xuất hóa đơn . . . . .	33
3.4.8	Lịch sử mua hàng của khách hàng . . . . .	34
3.4.9	Thêm chi tiết đơn hàng . . . . .	34
<b>Tổng kết</b>		<b>36</b>
<b>Tài liệu tham khảo</b>		<b>36</b>

# Chương 1

## Mô hình dữ liệu

### 1.1 Nghiệp vụ hệ thống, thực thể, liên kết

Nhằm giải quyết những bài toán liên quan tới việc quản lý chuỗi cửa hàng điện thoại, chúng em sẽ tiến hành phát triển một hệ thống cơ sở dữ liệu của một chuỗi cửa hàng điện thoại, bao gồm các nghiệp vụ cùng các thực thể tương ứng sau:

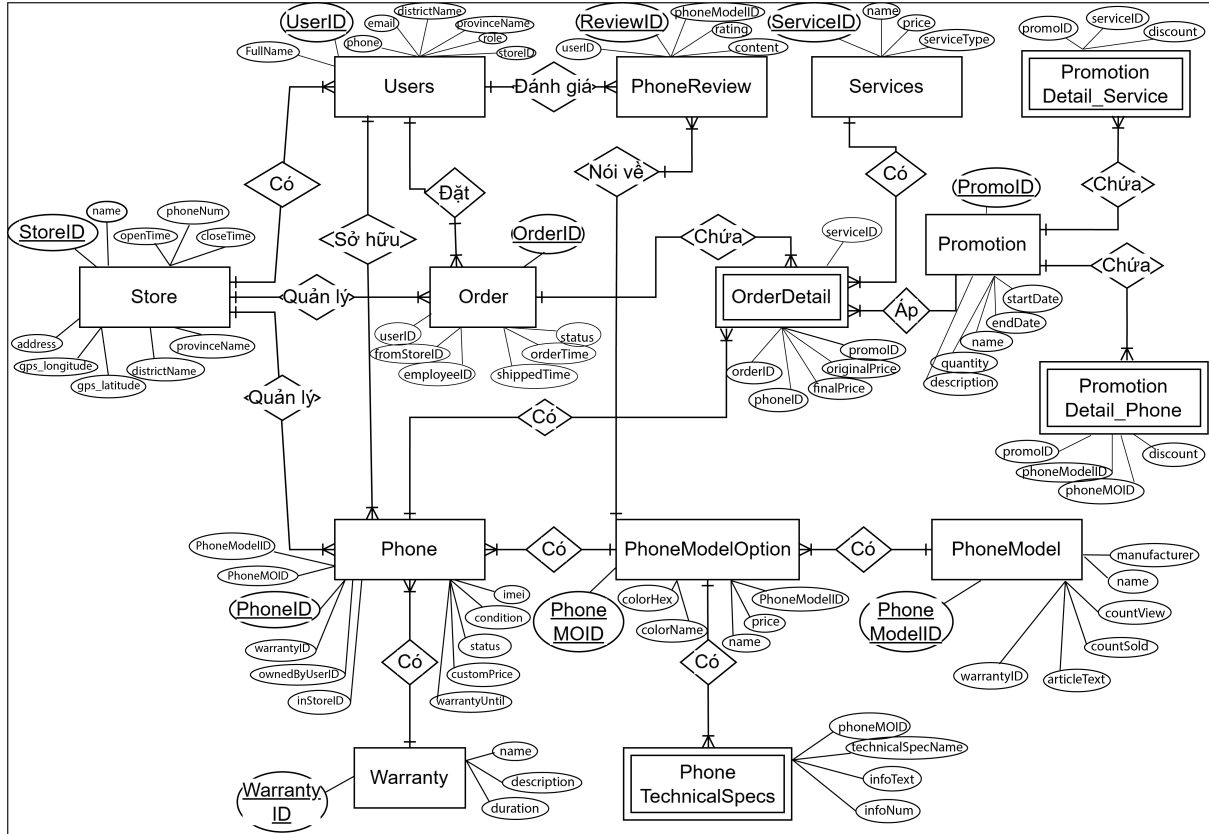
1. Quản lý danh mục điện thoại (**phoneModel**): Cho phép tìm kiếm, tra cứu các dòng điện thoại theo nhà sản xuất, theo từng tính năng; cho phép người dùng đánh giá, tìm hiểu bài phân tích về mẫu điện thoại; cho phép liệt kê nhiều phiên bản điện thoại khả dụng trong mẫu điện thoại này (**phoneModelOption**), mỗi phiên bản có thể có màu hoặc thông số khác nhau (**phoneTechnicalSpecs**).
2. Quản lý từng máy điện thoại (**phone**): Cho phép lưu trữ tìm kiếm, tra cứu những thông tin liên quan đến từng máy, như IMEI, trạng thái hiện tại, người dùng đang sở hữu, hạn bảo hành, ...
3. Quản lý chuỗi cửa hàng (**store**) và dữ liệu người dùng (**users**): Cho phép tra cứu thông tin của cửa hàng và người dùng, trạng thái kho máy ở cửa hàng, tra cứu danh sách nhân viên, ...
4. Quản lý dịch vụ đi kèm (**services**), ưu đãi (**promotion**), chương trình bảo hành (**warranty**): Quản lý quyền lợi của ưu đãi (**promotion\_detail\_service**, **promotion\_detail\_phone**), thời hạn bảo hành, dịch vụ đi kèm; cùng với giá của chúng.
5. Quản lý đơn hàng (**order**): Cho phép tra cứu lịch sử mua hàng, cụ thể tiểu mục trong đơn hàng như điện thoại, dịch vụ đi kèm (**order\_detail**), người mua hàng, giá trước và sau khi áp ưu đãi...

Ta có một số liên kết như sau:

- Cửa hàng (store) có thể quản lý nhiều Người dùng (users), nhiều Đơn hàng (orders), nhiều máy điện thoại (phone).
- Người dùng (users) có thể đặt nhiều Đơn hàng (orders), hoặc sở hữu nhiều điện thoại (phone). Ngoài ra còn có thể đăng nhiều Đánh giá (phoneReview), và mỗi Mẫu máy (phoneModel) có thể có nhiều Đánh giá.
- Một mẫu máy (phoneModel) có thể có nhiều phiên bản (phoneModelOption), một phiên bản điện thoại cũng có thể có nhiều máy điện thoại (phone). Mỗi phiên bản điện thoại cũng có nhiều thông số (phoneTechnicalSpecs) liên quan tới nó.
- Mỗi loại bảo hành (warranty) có thể áp cho nhiều máy điện thoại (phone) khác nhau.
- Mỗi đơn hàng (order) sẽ có nhiều tiểu mục (order\_detail), trong đó mỗi tiểu mục đề cập tới máy điện thoại (phone), dịch vụ đi kèm (services) và ưu đãi được áp (promotion).
- Mỗi ưu đãi (promotion) có nhiều quyền lợi ưu đãi ở nhiều mẫu máy (promotion\_detail\_phone), và/hoặc nhiều dịch vụ (promotion\_detail\_service). Mỗi quyền lợi ưu đãi lại có liên kết tới mẫu máy hoặc dịch vụ đó.

## 1.2 Mô hình ER

Với nghiệp vụ hệ thống, thực thể, liên kết đã liệt kê, quy mô của dữ liệu sẽ khá lớn. Ta sẽ bổ sung các thuộc tính cần thiết, và phác thảo lược đồ ER (Hình (1.1)):



Hình 1.1: (img\_c1/diag-1.1-ERModel.png) Lược đồ ER (tạm thời) của hệ CSDL

Ta nhận thấy có một số thuộc tính có giá trị xuất hiện nhiều ở các thực thể khác nhau, như Tỉnh (province), Quận/Huyện (district), Loại dịch vụ (serviceType), Loại thông số (technicalSpec), ... Để làm rõ liên kết giữa các thực thể này, khi thực hiện mô hình Quan hệ, ta sẽ tách các thuộc tính trên thành những thực thể riêng.

Tương tự, một số thuộc tính có thể chứa dữ liệu lớn, như Nội dung đánh giá người dùng (phoneReview(content)), Bài phân tích mẫu điện thoại (phoneModel(article)), ... nếu để chung trong một thực thể, sẽ ảnh hưởng lớn đến thời gian nhập xuất và truy vấn. Để giải quyết, ta cũng tách những thuộc tính trên thành thuộc thể riêng.

Ta sẽ hoàn thiện và chuyển từ mô hình ER sang mô hình Quan hệ.

## 1.3 Mô hình Quan hệ

### 1.3.1 Đặc tả dữ liệu

Ta có 21 bảng ứng với 21 thực thể, với các thuộc tính và ràng buộc:

Kiểu thực thể: phone_model - Mẫu điện thoại				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>phoneModelID</u>	INT	Mã định danh mẫu điện thoại	Primary Key	6
manufacturerID	INT	Mã định danh nhà sản xuất	FK tới manufacturer (manufacturerID)	2
warrantyID	INT	Mã định danh chế độ bảo hành	FK tới warranty (warrantyID)	2
articleID	INT	Mã định danh bài phân tích	FK tới article (articleID)	6
name	VARCHAR(50)	Tên mẫu điện thoại		'iPhone 12 Pro Max'
countView	INT	Số lượt xem mẫu điện thoại		131230
countSold	INT	Số lượt mua mẫu điện thoại		65123

Kiểu thực thể: phone_model_option - Phiên bản điện thoại				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>phoneModelOptionID</u>	INT	Mã định danh phiên bản điện thoại	Primary Key	28
phoneModelID	INT	Mã định danh mẫu điện thoại	FK tới phone_model (phoneModelID)	6
name	VARCHAR(100)	Tên phiên bản điện thoại		'iPhone 12 Pro Max Gold 12/256GB'
price	INT	Giá phiên bản điện thoại		26122000
colorHex	INT	Mã màu riêng cho phiên bản điện thoại		16766720
colorName	VARCHAR(20)	Tên màu riêng cho phiên bản điện thoại		'Gold'

Kiểu thực thể: phone_tech_spec - Thông số điện thoại				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>phoneModelOptionID</u>	INT	Mã định danh phiên bản điện thoại	Primary Key	28
<u>techSpecID</u>	INT	Mã định danh kiểu thông số	Primary Key FK tới technical_spec (techSpecID)	1
infoText	VARCHAR(100)	Nội dung thông số		'iOS 17'
infoNum	DECIMAL(10,3)	Nội dung dạng số của thông số		1
Kiểu thực thể: technical_spec - Kiểu thông số				
techSpecID	INT	Mã định danh loại thông số	Primary Key	1
name	VARCHAR(30)	Tên loại thông số		'OS'
description	TEXT	Mô tả loại thông số		'He dieu hanh'

Kiểu thực thể: phone - Từng máy điện thoại				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>phoneID</u>	INT	Mã định danh điện thoại	Primary Key	468
phoneModelID	INT	Mã định danh mẫu điện thoại	FK tới phone_model (phoneModelID)	6
phoneModelOptionID	INT	Mã định danh phiên bản điện thoại	FK tới phone_model_option (phoneModelOptionID)	28
ownedByUserID	INT	Mã định danh người dùng đang sở hữu (0 nếu chưa bán)	FK tới users (userID)	189
warrantyID	INT	Mã định danh loại bảo hành	FK tới warranty (warrantyID)	2
inStoreID	INT	Mã định danh cửa hàng tương ứng với điện thoại này	FK tới store (storeID)	68
countSold	INT	Số lượt mua mẫu điện thoại		65123
imei	VARCHAR(15)	Mã IMEI của máy		'178691161272448'
customPrice	INT	Giá riêng của điện thoại này (null nếu không có giá riêng)		null
phoneCondition	ENUM	Tình trạng máy ('New', 'Used', 'Refurbished')		'New'
status	ENUM	Trạng thái hiện tại máy ('InStore', 'Active', 'Repairing', 'Inactive')		'Active'
warrantyUntil	DATE	Ngày kết thúc bảo hành máy		'2025-03-06'

Kiểu thực thể: promotion - Ưu đãi				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>promotionID</u>	INT	Mã định danh ưu đãi	Primary Key	1
quantity	INT	Số lượng ưu đãi		12
name	VARCHAR(50)	Tên ưu đãi		'Xmas 2023 10%'
description	TEXT	Mô tả ưu đãi		'Khuyến mai 10% giang sinh 2023'
startDate	DATE	Ngày bắt đầu ưu đãi		'2023-12-22'
endDate	DATE	Ngày kết thúc ưu đãi		'2024-01-01'
Kiểu thực thể: promotion_detail_phone - Nội dung ưu đãi điện thoại				
<u>promotionID</u>	INT	Mã định danh ưu đãi	Primary Key FK tới promotion (promotionID)	1
<u>phoneModelID</u>	INT	Mã định danh mẫu điện thoại được áp ưu đãi (0 để chỉ toàn bộ)	Primary Key FK tới phone_model (phoneModelID)	0
<u>phoneModelOptionID</u>	INT	Mã định danh phiên bản điện thoại được áp ưu đãi (0 để chỉ toàn bộ)	Primary Key FK tới phone_model_option (phoneModelOptionID)	0
discountPercent	DECIMAL(5,2)	Phần trăm ưu đãi (null để không ưu đãi)		10
discountFixed	INT	Phần trăm ưu đãi (null để không ưu đãi)		null
fixedNewPrice	INT	Giá mới ghi đè sau ưu đãi (null để không ưu đãi)		null
Kiểu thực thể: promotion_detail_service - Nội dung ưu đãi dịch vụ				
<u>promotionID</u>	INT	Mã định danh ưu đãi	Primary Key FK tới promotion (promotionID)	7
<u>serviceTypeID</u>	INT	Mã định danh kiểu dịch vụ được áp ưu đãi (0 để chỉ toàn bộ)	Primary Key FK tới service_type (serviceTypeID)	3
<u>serviceID</u>	INT	Mã định danh dịch vụ được áp ưu đãi (0 để chỉ toàn bộ)	Primary Key FK tới service (serviceID)	0
discountPercent	DECIMAL(5,2)	Phần trăm ưu đãi (null để không ưu đãi)		30
discountFixed	INT	Phần trăm ưu đãi (null để không ưu đãi)		null
fixedNewPrice	INT	Giá mới ghi đè sau ưu đãi (null để không ưu đãi)		null

Kiểu thực thể: services - Dịch vụ đi kèm				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>serviceID</u>	INT	Mã định danh dịch vụ	Primary Key	9
<u>serviceTypeID</u>	INT	Mã định danh kiểu dịch vụ	FK tới service_type (serviceTypeID)	4
name	VARCHAR(50)	Tên dịch vụ		'Giao hàng tan nơi'
price	INT	Giá dịch vụ		50000
Kiểu thực thể: service_type - Kiểu dịch vụ				
<u>serviceTypeID</u>	INT	Mã định danh kiểu dịch vụ	Primary Key	4
name	VARCHAR(30)	Tên kiểu dịch vụ		'Giao hàng'
description	TEXT	Mô tả kiểu dịch vụ		'Giao hàng tan nơi'

Kiểu thực thể: store - Cửa hàng				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>storeID</u>	INT	Mã định danh cửa hàng	Primary Key	68
<u>provinceID</u>	INT	Mã định danh tỉnh	FK tới province (provinceID)	3
<u>districtID</u>	INT	Mã định danh quận, huyện	FK tới district (districtID)	27
name	VARCHAR(50)	Tên cửa hàng		'HCM-32_3 Lê Văn Khương'
address	VARCHAR(100)	Địa chỉ cửa hàng		'32/3 Lê Văn Khương Q.12, TPHCM'
phoneNumber	VARCHAR(15)	Số điện thoại cửa hàng		'(08)38.102.102'
gps_longitude	DECIMAL(10,5)	Kinh độ GPS cửa hàng		10.8654
gps_latitude	DECIMAL(10,5)	Vĩ độ GPS cửa hàng		106.65



Kiểu thực thể: users - Người dùng				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>userID</u>	INT	Mã định danh người dùng	Primary Key	189
provinceID	INT	Mã định danh tỉnh	FK tới province (provinceID)	3
districtID	INT	Mã định danh quận, huyện	FK tới district (districtID)	31
storeID	INT	Mã định danh cửa hàng liên kết	FK tới store (storeID)	49
fullName	VARCHAR(50)	Họ tên người dùng		'Trần Hà Anh'
phone	VARCHAR(15)	Số điện thoại người dùng		'+84667712501'
role	ENUM	Vai trò người dùng ('Customer', 'Employee')		'Customer'
address	VARCHAR(100)	Địa chỉ người dùng		'ijoxq...dqkfp'
email	VARCHAR(50)	Email người dùng		'nodp...@gmail.com'

Kiểu thực thể: phone_review - Đánh giá người dùng				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>reviewID</u>	INT	Mã định danh bài đánh giá	Primary Key	1
phoneModelID	INT	Mã định danh mẫu điện thoại	FK tới phone_model (phoneModelID)	1
phoneModelOptionID	INT	Mã định danh phiên bản điện thoại (0 nếu không chỉ rõ phiên bản)	FK tới phone_model_option (phoneModelOptionID)	1
userID	INT	Mã định danh người dùng đăng bài	FK tới users (userID)	175
rating	DECIMAL(3,2)	Đánh giá người dùng (thang 5)		2
likes	INT	Số lượt thích bài đánh giá		146
timePosted	DATETIME	Thời điểm đăng bài		'2024-01-25 15:00:33'

Kiểu thực thể: orders - Đơn hàng				
Thuộc tính	Kiểu	Ý nghĩa	Ràng buộc	Ví dụ
<u>orderID</u>	INT	Mã định danh đơn hàng	Primary Key	181
userID	INT	Mã định danh người dùng ứng với đơn hàng này	FK tới users (userID)	189
fromStoreID	INT	Mã định danh cửa hàng ứng với đơn hàng này	FK tới store (storeID)	68
employeeID	INT	Mã định danh nhân viên ứng với đơn hàng này	FK tới users (userID)	318
orderTime	DATETIME	Ngày giờ đặt hàng		'2024-09-07 14:57:22'
shippedTime	DATETIME	Ngày giờ nhận hàng (null nếu chưa nhận hàng)		'2024-09-16 06:41:37'
status	ENUM	Trạng thái đơn hàng ('Pending', 'Preparing', 'Delivering', 'Completed', 'Cancelled')		'Completed'
comment	TEXT	Ghi chú đơn hàng		null
Kiểu thực thể: order_detail - Chi tiết đơn hàng				
<u>orderID</u>	INT	Mã định danh đơn hàng	Primary Key FK tới orders (orderID)	181 181
<u>phoneID</u>	INT	Mã định danh máy điện thoại	Primary Key FK tới phone (phoneID)	468 468
<u>serviceID</u>	INT	Mã định danh dịch vụ đi kèm (0 nếu không phải dịch vụ)	Primary Key FK tới service (serviceID)	0 9
<u>promotionID</u>	INT	Mã định danh ưu đãi (0 nếu không áp ưu đãi)	FK tới promotion (promotionID)	0 0
originalPrice	INT	Giá gốc chưa ưu đãi		26122000 50000
finalPrice	INT	Giá sau ưu đãi		26122000 50000

Các kiểu thực thể khác có dạng (ID, tên, mô tả):

- **manufacturer** - Nhà sản xuất: (manufacturerID, name, description)
- **article** - Bài phân tích: (articleID, content)
- **warranty** - Chế độ bảo hành: (warrantyID, name, description, warrantyDuration (DATE, số ngày được bảo hành))
- **phone\_review\_detail** - Chi tiết đánh giá người dùng: (reviewID, content)
- **province** - Tỉnh: (provinceID, name)
- **district** - Quận/Huyện: (districtID, name, provinceID (INT, FK tới province, tỉnh của quận/huyện)).

### 1.3.2 Liên kết

Loại liên kết	Liên kết	Lý do
One-to-Many (Một-Nhiều)	manufacturer → phone_model	Một nhà sản xuất có thể có nhiều mẫu máy, và nhiều phiên bản điện thoại
	phone_model → phone_model_option	Một mẫu điện thoại có thể có nhiều phiên bản
	phone_model_option → phone_tech_spec	Một phiên bản điện thoại sẽ có nhiều thuộc tính đi với nó
	phone_model_option → phone_review	Một phiên bản điện thoại sẽ có nhiều đánh giá về nó
	phone_model_option → phone	Một phiên bản điện thoại sẽ có nhiều máy cùng phiên bản đó
	users → phone users → orders	Một người dùng có thể có nhiều điện thoại và cũng có thể đặt nhiều đơn hàng
	province → district district → store	Một tỉnh có thể có nhiều quận/huyện; Một quận/huyện có thể có nhiều cửa hàng
	promotion → promotion_detail_phone → promotion_detail_service	Một ưu đãi có thể có nhiều phần được ưu đãi
	service, phoneModelOptionID → promotion_detail_phone → promotion_detail_service	Mỗi dịch vụ đi kèm hoặc mỗi phiên bản điện thoại có thể có nhiều chương trình ưu đãi
	orders → order_detail	Một đơn hàng có thể gồm nhiều mặt hàng riêng biệt (điện thoại, dịch vụ, ...)
	service_type → service	Một loại dịch vụ có thể có nhiều dịch vụ riêng biệt
	store → users → phone → orders	Một cửa hàng có thể gồm nhiều người dùng, có thể chứa nhiều máy, và xử lý nhiều đơn hàng
	technical_spec → phone_tech_spec	Một loại thông số có thể xuất hiện trên nhiều máy
	warranty → phone	Một chương trình bảo hành có thể diễn ra đồng thời trên nhiều máy
One-to-One (Một-Một)	phone_model → article	Một mẫu máy sẽ chỉ có một bài phân tích
	phone_review → phone_review_detail	Một đánh giá sẽ có một nội dung tương ứng

### 1.3.3 Quan hệ, phụ thuộc hàm khi chuẩn hóa về 3NF

Dưới đây là các quan hệ và phụ thuộc hàm khi được chuẩn hóa về chuẩn 3NF:

- R1(province) = (provinceID, name)

- $F1 = \{ \text{provinceID} \rightarrow \text{name} \}$
- $R2(\text{district}) = (\underline{\text{districtID}}, \text{provinceID}, \text{name})$
- $F2 = \{ \text{districtID} \rightarrow \text{name}, \text{districtID} \rightarrow \text{province} \}$
- $R3(\text{store}) = (\underline{\text{storeID}}, \text{districtID}, \text{name}, \text{address}, \text{phoneNumber}, \text{gps\_longitude}, \text{gps\_latitude}, \text{openTime}, \text{closeTime})$
- $F3 = \{ \text{storeID} \rightarrow (\text{districtID}, \text{name}, \text{address}, \text{phoneNumber}, \text{gps\_longitude}, \text{gps\_latitude}, \text{openTime}, \text{closeTime}) \}$
- $R4(\text{users}) = (\underline{\text{userID}}, \text{districtID}, \text{storeID}, \text{role}, \text{fullName}, \text{email}, \text{phone}, \text{address})$
- $F4 = \{ \text{userID} \rightarrow (\text{districtID}, \text{storeID}, \text{role}, \text{fullName}, \text{email}, \text{phone}, \text{address}) \}$
- $R5(\text{orders}) = (\underline{\text{orderID}}, \text{userID}, \text{fromStoreID}, \text{employeeID}, \text{status}, \text{orderTime}, \text{shippedTime}, \text{comment})$
- $F5 = \{ \text{orderID} \rightarrow (\text{userID}, \text{fromStoreID}, \text{employeeID}, \text{status}, \text{orderTime}, \text{shippedTime}, \text{comment}) \}$
- $R6(\text{order\_detail}) = (\underline{\text{orderID}}, \underline{\text{phoneID}}, \underline{\text{serviceID}}, \text{promotionID}, \text{originalPrice}, \text{finalPrice})$
- $F6 = \{ (\text{phoneID}, \text{serviceID}) \rightarrow \text{originalPrice}, (\text{orderID}, \text{phoneID}, \text{serviceID}) \rightarrow (\text{promotionID}, \text{finalPrice}) \}$
- $R7(\text{phone}) = (\underline{\text{phoneID}}, \text{phoneModelOptionID}, \text{ownedByUserID}, \text{warrantyID}, \text{inStoreID}, \text{imei}, \text{customPrice}, \text{phoneCondition}, \text{status}, \text{warrantyUntil})$
- $F7 = \{ \text{phoneID} \rightarrow (\text{phoneModelOptionID}, \text{ownedByUserID}, \text{warrantyID}, \text{inStoreID}, \text{imei}, \text{customPrice}, \text{phoneCondition}, \text{status}, \text{warrantyUntil}) \}$
- $R8(\text{phone\_model\_option}) = (\underline{\text{phoneModelOptionID}}, \text{phoneModelID}, \text{price}, \text{name}, \text{colorHex}, \text{colorName})$
- $F8 = \{ \text{phoneModelOptionID} \rightarrow (\text{phoneModelID}, \text{price}, \text{name}, \text{colorHex}, \text{colorName}) \}$
- $R9(\text{phone\_model}) = (\underline{\text{phoneModelID}}, \text{manufacturerID}, \text{warrantyID}, \text{articleID}, \text{name}, \text{countView}, \text{countSold})$
- $F9 = \{ \text{phoneModelID} \rightarrow (\text{manufacturerID}, \text{warrantyID}, \text{articleID}, \text{name}, \text{countView}, \text{countSold}) \}$
- $R10(\text{phone\_review}) = (\underline{\text{reviewID}}, \text{phoneModelID}, \text{phoneModelOptionID}, \text{userID}, \text{rating}, \text{likes}, \text{timePosted})$
- $F10 = \{ \text{reviewID} \rightarrow (\text{phoneModelID}, \text{phoneModelOptionID}, \text{userID}, \text{rating}, \text{likes}, \text{timePosted}) \}$
- $R11(\text{phone\_review\_detail}) = (\underline{\text{reviewID}}, \text{content})$
- $F11 = \{ \text{reviewID} \rightarrow \text{content} \}$
- $R12(\text{phone\_tech\_spec}) = (\underline{\text{phoneModelOptionID}}, \underline{\text{techSpecID}}, \text{infoText}, \text{infoNum})$
- $F12 = \{ (\text{phoneModelOptionID}, \text{techSpecID}) \rightarrow (\text{infoText}, \text{infoNum}) \}$
- $R13(\text{technical\_spec}) = (\underline{\text{techSpecID}}, \text{name}, \text{description})$
- $F13 = \{ \text{techSpecID} \rightarrow (\text{name}, \text{description}) \}$
- $R14(\text{article}) = (\underline{\text{articleID}}, \text{content})$
- $F14 = \{ \text{articleID} \rightarrow \text{content} \}$
- $R15(\text{manufacturer}) = (\underline{\text{manufacturerID}}, \text{name}, \text{description})$
- $F15 = \{ \text{manufacturerID} \rightarrow (\text{name}, \text{description}) \}$
- $R16(\text{warranty}) = (\underline{\text{warrantyID}}, \text{name}, \text{description}, \text{warrantyDuration})$

- F16 = { warrantyID  $\rightarrow$  (name, description, warrantyDuration) }
- R17(service) = (serviceID, name, price, serviceTypeID)
- F17 = { serviceID  $\rightarrow$  (name, price, serviceTypeID) }
- R18(service\_type) = (serviceTypeID, name, description)
- F18 = { serviceTypeID  $\rightarrow$  (name, description) }
- R19(promotion) = (promotionID, startDate, endDate, name, description, quantity)
- F19 = { promotionID  $\rightarrow$  (startDate, endDate, name, description, quantity) }
- R20(promotion\_detail\_service) = (promotionID, serviceTypeID, serviceID, discountPercent, discountFixed, fixedNewPrice)
- F20 = { (promotionID, serviceTypeID, serviceID)  $\rightarrow$  (discountPercent, discountFixed, fixedNewPrice) }
- R21(promotion\_detail\_phone) = (promotionID, phoneModelID, phoneModelOptionID, discountPercent, discountFixed, fixedNewPrice)
- F21 = { (promotionID, phoneModelID, phoneModelOptionID)  $\rightarrow$  (discountPercent, discountFixed, fixedNewPrice) }

Cuối cùng, ta có thể sinh lược đồ trong MySQL Workbench (diagram.mwb):

Hình 1.2: (img\_c1/diag-1.2-QH.png) Lược đồ mô hình Quan hệ của hệ CSDL

## Chương 2

# Khởi tạo cơ sở dữ liệu trên SQL

### 2.1 CREATE TABLE (createDB.sql)

Đầu tiên ta sẽ tạo database mới có tên 'thegioididong'. Tương ứng với 21 bảng là 21 lệnh CREATE TABLE. Ta sẽ đặt ràng buộc primary key ngay trong lệnh.

**Code:** Thêm bảng

```
DROP DATABASE IF EXISTS thegioididong;
CREATE DATABASE thegioididong;
USE thegioididong;

CREATE TABLE phone_model (
    phoneModelID INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    manufacturerID INT NOT NULL,
    countView INT NOT NULL,
    countSold INT NOT NULL,
    warrantyID INT,
    articleID INT,
    PRIMARY KEY (phoneModelID)
);

CREATE TABLE phone_model_option (
    phoneModelOptionID INT NOT NULL AUTO_INCREMENT,
    phoneModelID INT NOT NULL,
    price INT NOT NULL,
    name VARCHAR(100) NOT NULL,
    colorHex INT,
    colorName VARCHAR(20),
    PRIMARY KEY (phoneModelOptionID)
);

CREATE TABLE phone_tech_spec (
    phoneModelOptionID INT NOT NULL,
    techSpecID INT NOT NULL,
    infoText VARCHAR(100),
    infoNum DECIMAL(10,3),
    PRIMARY KEY (phoneModelOptionID, techSpecID)
);

CREATE TABLE technical_spec (
    techSpecID INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    description TEXT,
    PRIMARY KEY (techSpecID)
);

CREATE TABLE manufacturer (
    manufacturerID INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(15) NOT NULL,
    description TEXT,
    PRIMARY KEY (manufacturerID)
);
```

```

CREATE TABLE warranty (
    warrantyID INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    description TEXT,
    warrantyDuration INT NOT NULL,
    PRIMARY KEY (warrantyID)
);

CREATE TABLE article (
    articleID INT NOT NULL AUTO_INCREMENT,
    content TEXT,
    PRIMARY KEY (articleID)
);

CREATE TABLE phone_review (
    reviewID INT NOT NULL AUTO_INCREMENT,
    phoneModelID INT NOT NULL,
    phoneModelOptionID INT,
    userID INT NOT NULL,
    rating DECIMAL(3,2) NOT NULL,
    likes INT NOT NULL,
    timePosted DATETIME NOT NULL,
    PRIMARY KEY (reviewID)
);

CREATE TABLE phone_review_detail (
    reviewID INT NOT NULL,
    content TEXT NOT NULL,
    PRIMARY KEY (reviewID)
);

CREATE TABLE phone (
    phoneID INT NOT NULL AUTO_INCREMENT,
    phoneModelID INT NOT NULL,
    phoneModelOptionID INT NOT NULL,
    imei VARCHAR(15) NOT NULL,
    customPrice INT,
    phoneCondition ENUM('New', 'Used', 'Refurbished') NOT NULL,
    status ENUM('InStore', 'Active', 'Repairing', 'Inactive') NOT NULL,
    ownedByUserID INT,
    warrantyID INT,
    warrantyUntil DATE,
    inStoreID INT,
    PRIMARY KEY (phoneID)
);

CREATE TABLE store (
    storeID INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    address VARCHAR(100) NOT NULL,
    phoneNumber VARCHAR(15) NOT NULL,
    gps_longitude DECIMAL(10, 5) NOT NULL,
    gps_latitude DECIMAL(10, 5) NOT NULL,
    provinceID INT NOT NULL,
    districtID INT NOT NULL,
    openTime TIME,
    closeTime TIME,
    PRIMARY KEY (storeID)
);

CREATE TABLE province (
    provinceID INT NOT NULL,
    name VARCHAR(50) NOT NULL,
    PRIMARY KEY (provinceID)
);

CREATE TABLE district (
    districtID INT NOT NULL,
    name VARCHAR(50) NOT NULL,
    provinceID INT NOT NULL,
    PRIMARY KEY (districtID)
);

```

```

CREATE TABLE services (
  serviceID INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  price INT NOT NULL,
  serviceTypeID INT NOT NULL,
  PRIMARY KEY (serviceID)
);

CREATE TABLE service_type (
  serviceTypeID INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(50) NOT NULL,
  description TEXT,
  PRIMARY KEY (serviceTypeID)
);

CREATE TABLE users (
  userID INT NOT NULL AUTO_INCREMENT,
  fullName VARCHAR(50) NOT NULL,
  email VARCHAR(50),
  phone VARCHAR(15),
  address VARCHAR(100),
  provinceID INT,
  districtID INT,
  role ENUM('Customer', 'Employee') NOT NULL,
  storeID INT,
  PRIMARY KEY (userID)
);

CREATE TABLE orders (
  orderID INT NOT NULL AUTO_INCREMENT,
  orderTime DATETIME NOT NULL,
  shippedTime DATETIME,
  status ENUM('Pending', 'Preparing', 'Delivering', 'Completed', 'Cancelled') NOT NULL,
  comment TEXT,
  userID INT NOT NULL,
  fromStoreID INT NOT NULL,
  employeeID INT,
  PRIMARY KEY (orderID)
);

CREATE TABLE order_detail (
  orderID INT NOT NULL,
  phoneID INT NOT NULL,
  serviceID INT NOT NULL,
  promotionID INT,
  originalPrice INT NOT NULL,
  finalPrice INT NOT NULL,
  PRIMARY KEY (orderID, phoneID, serviceID)
);

CREATE TABLE promotion (
  promotionID INT NOT NULL AUTO_INCREMENT,
  startDate DATE NOT NULL,
  endDate DATE NOT NULL,
  name VARCHAR(50) NOT NULL,
  description TEXT,
  quantity INT,
  PRIMARY KEY (promotionID)
);

CREATE TABLE promotion_detail_phone (
  promotionID INT NOT NULL,
  phoneModelID INT NOT NULL,
  phoneModelOptionID INT NOT NULL,
  discountPercent DECIMAL(5,2),
  discountFixed INT,
  fixedNewPrice INT,
  PRIMARY KEY (promotionID, phoneModelID, phoneModelOptionID)
);

CREATE TABLE promotion_detail_service (
  promotionID INT NOT NULL,

```

```

serviceTypeID INT NOT NULL,
serviceID INT NOT NULL,
discountPercent DECIMAL(5,2),
discountFixed INT,
fixedNewPrice INT,
PRIMARY KEY (promotionID, serviceTypeID, serviceID)
);

```

## 2.2 CONSTRAINTS (constraints.sql)

**Code:** Định ràng buộc foreign key

```

use thegioididong;

ALTER TABLE phone_model
ADD FOREIGN KEY (manufacturerID) REFERENCES manufacturer(manufacturerID),
ADD FOREIGN KEY (warrantyID) REFERENCES warranty(warrantyID),
ADD FOREIGN KEY (articleID) REFERENCES article(articleID);

ALTER TABLE phone_model_option
ADD FOREIGN KEY (phoneModelID) REFERENCES phone_model(phoneModelID);

ALTER TABLE phone_tech_spec
ADD FOREIGN KEY (phoneModelOptionID) REFERENCES phone_model_option(phoneModelOptionID),
ADD FOREIGN KEY (techSpecID) REFERENCES technical_spec(techSpecID);

ALTER TABLE phone_review
ADD FOREIGN KEY (phoneModelID) REFERENCES phone_model(phoneModelID),
ADD FOREIGN KEY (phoneModelOptionID) REFERENCES phone_model_option(phoneModelOptionID),
ADD FOREIGN KEY (userID) REFERENCES users(userID);

ALTER TABLE phone_review_detail
ADD FOREIGN KEY (reviewID) REFERENCES phone_review(reviewID);

ALTER TABLE phone
ADD FOREIGN KEY (ownedByUserID) REFERENCES users(userID),
ADD FOREIGN KEY (warrantyID) REFERENCES warranty(warrantyID),
ADD FOREIGN KEY (inStoreID) REFERENCES store(storeID),
ADD FOREIGN KEY (phoneModelID) REFERENCES phone_model(phoneModelID),
ADD FOREIGN KEY (phoneModelOptionID) REFERENCES phone_model_option(phoneModelOptionID);

ALTER TABLE store
ADD FOREIGN KEY (provinceID) REFERENCES province(provinceID),
ADD FOREIGN KEY (districtID) REFERENCES district(districtID);

ALTER TABLE district
ADD FOREIGN KEY (provinceID) REFERENCES province(provinceID);

ALTER TABLE services
ADD FOREIGN KEY (serviceTypeID) REFERENCES service_type(serviceTypeID);

ALTER TABLE users
ADD FOREIGN KEY (provinceID) REFERENCES province(provinceID),
ADD FOREIGN KEY (districtID) REFERENCES district(districtID),
ADD FOREIGN KEY (storeID) REFERENCES store(storeID);

ALTER TABLE order_detail
ADD FOREIGN KEY (orderID) REFERENCES orders(orderID),
ADD FOREIGN KEY (phoneID) REFERENCES phone(phoneID),
ADD FOREIGN KEY (serviceID) REFERENCES services(serviceID),
ADD FOREIGN KEY (promotionID) REFERENCES promotion(promotionID);

ALTER TABLE orders
ADD FOREIGN KEY (userID) REFERENCES users(userID),
ADD FOREIGN KEY (employeeID) REFERENCES users(userID),
ADD FOREIGN KEY (fromStoreID) REFERENCES store(storeID);

ALTER TABLE promotion_detail_phone
ADD FOREIGN KEY (phoneModelID) REFERENCES phone_model(phoneModelID),
ADD FOREIGN KEY (phoneModelOptionID) REFERENCES phone_model_option(phoneModelOptionID),

```



```

ADD FOREIGN KEY (promotionID) REFERENCES promotion(promotionID);

ALTER TABLE promotion_detail_service
ADD FOREIGN KEY (serviceID) REFERENCES services(serviceID),
ADD FOREIGN KEY (serviceTypeID) REFERENCES service_type(serviceTypeID),
ADD FOREIGN KEY (promotionID) REFERENCES promotion(promotionID);

```

Tiếp theo, ta sẽ đặt giá trị khởi tạo mặc định cho một số thuộc tính:

**Code:** Giá trị mặc định

```

-- Add default values
ALTER TABLE phone_model
ALTER COLUMN countView SET DEFAULT 0,
ALTER COLUMN countSold SET DEFAULT 0;

ALTER TABLE phone_model_option
ALTER COLUMN colorHex SET DEFAULT 0,
ALTER COLUMN colorName SET DEFAULT 'Black';

ALTER TABLE orders
ALTER COLUMN status SET DEFAULT 'Pending';

ALTER TABLE users
ALTER COLUMN role SET DEFAULT 'Customer';

ALTER TABLE store
ALTER COLUMN openTime SET DEFAULT '08:00',
ALTER COLUMN closeTime SET DEFAULT '22:30';

```

Tiếp theo, ta sẽ đặt ràng buộc index:

**Code:** Ràng buộc chỉ số index

```

-- Add index constraints
ALTER TABLE phone_model ADD INDEX idx_phoneModelID (phoneModelID);
ALTER TABLE phone_model_option ADD INDEX idx_phoneModelOptionID (phoneModelOptionID);
ALTER TABLE technical_spec ADD INDEX idx_techSpecID (techSpecID);
ALTER TABLE manufacturer ADD INDEX idx_manufacturerID (manufacturerID);
ALTER TABLE warranty ADD INDEX idx_warrantyID (warrantyID);
ALTER TABLE article ADD INDEX idx_articleID (articleID);
ALTER TABLE phone_review ADD INDEX idx_reviewID (reviewID);
ALTER TABLE phone ADD INDEX idx_phoneID (phoneID);
ALTER TABLE store ADD INDEX idx_storeID (storeID);
ALTER TABLE services ADD INDEX idx_serviceID (serviceID);
ALTER TABLE service_type ADD INDEX idx_serviceTypeID (serviceTypeID);
ALTER TABLE users ADD INDEX idx_userID (userID);
ALTER TABLE orders ADD INDEX idx_orderID (orderID);
ALTER TABLE promotion ADD INDEX idx_promotionID (promotionID);

```

Cuối cùng là ràng buộc điều kiện.

**Code:** Ràng buộc điều kiện

```

-- Rating must be between 1 to 5
ALTER TABLE phone_review ADD CHECK (rating >= 1 AND rating <= 5);

-- Prices cannot be negative values
ALTER TABLE phone ADD CHECK (customPrice >= 0);
ALTER TABLE order_detail ADD CHECK (originalPrice >= 0 AND finalPrice >= 0);

```

## 2.3 INSERT (insert.sql)

Với bước thêm dữ liệu vào bảng, ta cần xử lý tính chất của dữ liệu có các trường hợp đặc biệt, ví dụ như những thuộc tính ID foreign key có giá trị bằng 0. Để xử lý những trường hợp này, ta sẽ định giá trị cho những hàng có index = 0 trong một số bảng trước:

**Code:** Định giá trị cho index 0

```

INSERT INTO
promotion(promotionID, name, startDate, endDate)

```

```

VALUES
(0, 'Khong co gi', '2000-01-01', '2000-01-01');
UPDATE 'promotion' SET 'promotionID'=0 WHERE 'promotionID'=1;
ALTER TABLE 'promotion' AUTO_INCREMENT = 1;

INSERT INTO
manufacturer(manufacturerID, name)
VALUES
(0, 'Khong');
UPDATE 'manufacturer' SET 'manufacturerID'=0 WHERE 'manufacturerID'=1;
ALTER TABLE 'manufacturer' AUTO_INCREMENT = 1;

INSERT INTO
phone_model(phoneModelID, name, manufacturerID)
VALUES
(0, 'Toan bo mau may', 0);
UPDATE 'phone_model' SET 'phoneModelID'=0 WHERE 'phoneModelID'=1;
ALTER TABLE 'phone_model' AUTO_INCREMENT = 1;

INSERT INTO
phone_model_option(phoneModelOptionID, phoneModelID, price, name)
VALUES
(0, 0, 0, 'Toan bo lua chon cua mau may');
UPDATE 'phone_model_option' SET 'phoneModelOptionID'=0 WHERE 'phoneModelOptionID'=1;
ALTER TABLE 'phone_model_option' AUTO_INCREMENT = 1;

INSERT INTO
service_type(serviceTypeID, name)
VALUES
(0, 'Khong dich vu');
UPDATE 'service_type' SET 'serviceTypeID'=0 WHERE 'serviceTypeID'=1;
ALTER TABLE 'service_type' AUTO_INCREMENT = 1;

INSERT INTO
services(serviceID, name, price, serviceTypeID)
VALUES
(0, 'Khong dich vu di kem', 0, 0);
UPDATE 'services' SET 'serviceID'=0 WHERE 'serviceID'=1;
ALTER TABLE 'services' AUTO_INCREMENT = 1;

INSERT INTO
warranty(warrantyID, name, warrantyDuration)
VALUES
(0, 'Khong bao hanh', 0);
UPDATE 'warranty' SET 'warrantyID'=0 WHERE 'warrantyID'=1;
ALTER TABLE 'warranty' AUTO_INCREMENT = 1;

INSERT INTO
users(userID, fullName, role)
VALUES
(0, "Dai dien Cua hang", 'Employee');
UPDATE 'users' SET 'userID'=0 WHERE 'userID'=1;
ALTER TABLE 'users' AUTO_INCREMENT = 1;

```

Sau đó ta thực hiện chèn các dữ liệu theo thứ tự cụ thể để đảm bảo ràng buộc. Do dữ liệu nhập bằng rất lớn, chúng em sẽ chỉ ghi 5 hàng mẫu của mỗi bảng.

**Code:** Thêm dữ liệu (tối đa 5 hàng mẫu)

```

INSERT INTO manufacturer(name, description)
VALUES
('Samsung', 'Samsung Electronics Co., Ltd. ...'),
('Apple', 'Apple Inc. ...'),
('Xiaomi', 'Xiaomi Corporation ...'),
('vivo', 'vivo Communication Technology Co. Ltd. is a C...');

INSERT INTO warranty(name, description, warrantyDuration)
VALUES
('1 nam Samsung va Xiaomi', 'Bao hanh 1 nam xin xo con bo cho Samsung va Xiaomi', 365),
('1 nam AppleCare+', 'Bao hanh 1 nam cho toan bo san pham Apple moi nhat', 365),
('6 thang AppleCare', 'Bao hanh 6 thang cho san pham Apple', 180),
('3 thang bao hanh chung', 'Bao hanh 3 thang cho toan bo san pham', 90),
('6 thang bao hanh chung', 'Bao hanh 6 thang cho toan bo san pham', 180),

```

```

...

INSERT INTO article(content)
VALUES
('Samsung Galaxy S21 Ultra is a great phone, with ...'),
('Samsung Galaxy S21 Plus is a great phone, ...'),
('Samsung Galaxy S21 is a great phone, with a lot of features. ...'),
('Samsung Galaxy S20 Ultra is a...'),
('Samsung Galaxy S20 is...'),
...

INSERT INTO service_type(name, description)
VALUES
('Bao hanh', 'Bao hanh xin xo hehehe'),
('Kinh cuong luc', 'Khong so var dap'),
('Sac nhanh', 'Sac sieu toc do cam phat day pin luon'),
('Giao hang tan noi', 'Giao hang tan giuong');

INSERT INTO services(name, price, serviceTypeID)
VALUES
('Bao hanh mo rong 3 thang', 1000000, 1),
('Bao hanh mo rong 6 thang', 2000000, 1),
('Bao hanh mo rong 1 nam', 3000000, 1),
('Kinh cuong luc cho iPhone', 300000, 2),
('Kinh cuong luc cho Samsung/Xiaomi/vivo', 500000, 2),
...

INSERT INTO province(provinceID, name)
VALUES
(3, 'TP.Ho Chi Minh'),
(5, 'Ha Noi'),
(6, 'Dak Lak'),
(7, 'Can Tho'),
(8, 'Dong Nai'),
...

INSERT INTO district(districtID, name, provinceID)
VALUES
(16, 'Quan 1', 3),
(18, 'Quan 3', 3),
(1308, 'Huyen Hoa Thanh', 146),
(1311, 'Thi xa Tay Ninh', 146),
(1313, 'Huyen Dong Hung', 147),
...

INSERT INTO
store(address, districtID, name, phoneNumber, gps_longitude, gps_latitude, provinceID)
VALUES
('11A Thai Ha, P.Trung Liet, Q. Dong Da, Ha Noi', 37, 'HN -11A Thai Ha', '(04)37.757.757', 21.013, 105.82, 5),
('Quoc lo 91, KV4, Phuong Chau Van Liem, Quan O Mon, Can Tho', 57, 'CTH - O Mon', '(08) 38.102.102', 10.1091, 105.62, 7),
('362A Nguyen Van Nghi, Quan Go Vap, TPHCM', 29, 'HCM-362A Nguyen Van Nghi', '(08) 38.102.102', 10.8244, 106.686, 3),
('15 Nguyen Huu Tho, KP3, Thi Tran Ben Luc, Long An', 1139, 'LA - Ben Luc', '(08) 38.102.102', 10.6372, 106.489, 132),
('848 Lac Long Quan, P.8, Quan Tan Binh, TPHCM', 30, 'HCM - Lac Long Quan', '(08) 38.102.102', 10.7815, 106.65, 3),
...

INSERT INTO
users(fullName, email, phone, address, provinceID, districtID, role, storeID)
VALUES
('Vu Dinh Khanh', 'emyfvjbbdf@gmail.com', '+84436421181', 'ariujjkszyobbuxbtyc', 3, 31, 'Customer', 76),
('Luong Thu Quan', 'wqswcfjygu@gmail.com', '+84719506905', 'mqbnkhvgdabtjnayyszx', 8, 723, 'Customer', 39),
('Ngo Ha Hung', 'stirrocjfx@gmail.com', '+84663735064', 'adtkpqvamignhgxbcbgh', 5, 43, 'Customer', 5),
('Duong Ha Nam', 'besjnxmvot@gmail.com', '+84946873561', 'hfundybvzvkioiovfdcfxq', 3, 31, 'Customer', 76),
('Do Thi Anh', 'fqqupcrdrc@gmail.com', '+84951467973', 'xvnumsdccqwaadivsy', 147, 1313, 'Customer', 66),

```

```

...

INSERT INTO
phone_model(name, manufacturerID, countView, countSold, warrantyID, articleID)
VALUES
('Samsung Galaxy S21 Ultra', 1, 121210, 52521, 1, 1),
('Samsung Galaxy S21 Plus', 1, 2210, 1120, 1, 2),
('Samsung Galaxy S21', 1, 3120, 1512, 1, 3),
('Samsung Galaxy S20 Ultra', 1, 4120, 2730, 1, 4),
('Samsung Galaxy S20', 1, 610, 310, 1, 5),
...

INSERT INTO
phone_model_option(phoneModelID, price, name, colorHex, colorName)
VALUES
(1, 25220000, 'Samsung Galaxy S21 Ultra Green 6/256GB', 32768, 'Green'),
(1, 26602000, 'Samsung Galaxy S21 Ultra Green 16/512GB', 32768, 'Green'),
(2, 20902000, 'Samsung Galaxy S21 Plus Gold 12/128GB', 16766720, 'Gold'),
(2, 20566000, 'Samsung Galaxy S21 Plus Gold 4/256GB', 16766720, 'Gold'),
(2, 21494000, 'Samsung Galaxy S21 Plus Gold 16/512GB', 16766720, 'Gold'),
...

INSERT INTO technical_spec(name, description)
VALUES
('OS', 'He dieu hanh'),
('CPU', 'Bo xu ly trung tam'),
('RAM', 'Bo nho tam'),
('ROM', 'Bo nho trong'),
('Display Size', 'Kich co man hinh'),
...

INSERT INTO phone_tech_spec(phoneModelOptionID, techSpecID, infoText, infoNum)
VALUES
(1, 1, 'Android 13', 0),
(1, 2, 'Snapdragon 835 3.0GHz', 3.0),
(1, 3, '6GB', 6),
(1, 4, '256GB', 256),
(1, 5, '4.9inch', 4.9),
...

INSERT INTO
phone(phoneModelID, phoneModelOptionID, imei, customPrice, phoneCondition, status,
ownedByUserID, warrantyID, warrantyUntil, inStoreID)
VALUES
(1, 1, '231597703200072', null, 'New', 'Active', 157, 1, '2023-07-26', 8),
(1, 1, '968012177541041', 22120000, 'Used', 'Active', 180, 1, '2023-12-30', 40),
(1, 1, '471551095813510', null, 'New', 'InStore', 0, 0, null, 62),
(1, 1, '329888516079876', 21820000, 'Used', 'Active', 231, 1, '2025-01-13', 77),
(1, 1, '712043561549760', null, 'New', 'Active', 71, 1, '2024-06-17', 48),
...

INSERT INTO
phone_review(phoneModelID, phoneModelOptionID, userID, rating, likes, timePosted)
VALUES
(1, 1, 175, 2, 146, '2024-01-25 15:00:33'),
(1, 1, 35, 3, 31, '2023-11-03 02:15:12'),
(1, 1, 39, 3, 69, '2023-10-23 16:18:37'),
(1, 1, 41, 1, 37, '2024-04-01 12:40:05'),
(1, 1, 154, 1, 23, '2024-07-19 14:10:42'),
...

INSERT INTO phone_review_detail(reviewID, content)
VALUES
(1, 'Review 1: This phone is awful. I hate the performance.'),
(2, 'Review 2: This phone is average. I think the camera could be better.'),
(3, 'Review 3: This phone is average. I think the build quality could be better.'),
(4, 'Review 4: This phone is horrible. I hate the performance.'),
(5, 'Review 5: This phone is poor. I hate the features.'),
...

INSERT INTO promotion(name, description, startDate, endDate, quantity)
VALUES
('Xmas 2023 10%', 'Khuyen mai 10% giang sinh 2023', '2023-12-22', '2024-01-01', 12),

```

```

('Back to School 2024 2000k', 'Khuyen mai 2 trieu back to school', '2024-08-01', '
2024-09-30', 43),
('Black Friday 2024 15%', 'Khuyen mai black friday sap san hhuhuhu', '2024-11-20', '
2024-11-30', 1),
('vivo giam 1000k', 'Khuyen mai danh rieng cho khach hang su dung dien thoai vivo 1000k'
, '2024-04-01', '2024-07-30', 69),
('Uu dai len doi iPhone', 'Khuyen mai uu dai len doi iPhone 12 Pro Max gia uu dai', '
2024-09-15', '2024-09-30', 3),
...

INSERT INTO
promotion_detail_phone(promotionID, phoneModelID, phoneModelOptionID, discountPercent,
discountFixed, fixedNewPrice)
VALUES
(1, 0, 0, 10, null, null),
(2, 0, 0, null, 2000000, null),
(3, 0, 0, 15, null, null),
(4, 17, 0, null, 1000000, null),
(4, 18, 0, null, 1000000, null),
...

INSERT INTO
promotion_detail_service(promotionID, serviceTypeID, serviceID, discountPercent,
discountFixed, fixedNewPrice)
VALUES
(6, 2, 0, null, null, 0),
(7, 3, 0, 30, null, null);

INSERT INTO
orders(orderTime, shippedTime, status, userID, fromStoreID, employeeID)
VALUES
('2023-01-27 11:11:30', '2023-01-31 19:00:21', 'Completed', 157, 8, 258),
('2023-07-03 17:19:57', '2023-07-05 23:49:31', 'Completed', 180, 40, 290),
('2024-07-17 14:44:54', '2024-07-24 21:19:38', 'Completed', 231, 77, 327),
('2023-12-20 07:09:08', '2023-12-28 00:41:42', 'Completed', 71, 48, 298),
('2023-10-13 11:25:44', '2023-10-23 08:48:47', 'Completed', 172, 20, 270),
...

INSERT INTO
order_detail(orderID, phoneID, serviceID, promotionID, originalPrice, finalPrice)
VALUES
(1, 1, 0, 0, 25220000, 25220000),
(1, 1, 9, 0, 50000, 50000),
(1, 1, 5, 0, 500000, 500000),
(1, 1088, 0, 0, 7974000, 7974000),
(1, 1088, 9, 0, 50000, 50000),
...

```

## Chương 3

# Thao tác truy vấn trên SQL

### 3.1 QUERY (query.sql)

Với yêu cầu có ít nhất một truy vấn và các truy vấn đáp ứng các yêu cầu sau:

- Query using inner join.
- Query using outer join.
- Using subquery in where.
- Using subquery in from.
- Query using group by and aggregate functions.

Trong hệ thống quản lý của ta sẽ có những truy vấn sau:

#### 3.1.1 Lấy thông tin kỹ thuật của máy

Code:

```
SELECT pm.name AS PhoneModelName, ts.name AS TechSpecName, pts.infoText AS InfoText,
       pts.infoNum AS InfoNum FROM phone_tech_spec pts
INNER JOIN phone_model_option pmo ON pts.phoneModelOptionID = pmo.phoneModelOptionID
INNER JOIN phone_model pm ON pmo.phoneModelID = pm.phoneModelID
INNER JOIN technical_spec ts ON pts.techSpecID = ts.techSpecID
ORDER BY pm.name, ts.name;
```

	PhoneModelName	TechSpecName	InfoText	InfoNum
▶	iPhone 11	Battery	4470mAh	4470.000
	iPhone 11	Battery	5330mAh	5330.000
	iPhone 11	Battery	5950mAh	5950.000
	iPhone 11	Battery	4150mAh	4150.000
	iPhone 11	Battery	3460mAh	3460.000
	iPhone 11	CPU	Apple A12 Bionic 3.7GHz	3.700
	iPhone 11	CPU	Apple A13 Bionic 2.1GHz	2.100

#### 3.1.2 Lấy danh sách tất cả các điện thoại và thông tin khuyến mãi (nếu có)

Code:

```
SELECT pm.name AS PhoneModelName, p.name AS PromotionName, CONCAT(p.startDate, ' - ', p.
       endDate) AS PromotionDate FROM phone_model pm
LEFT OUTER JOIN promotion_detail_phone pdp ON pm.phoneModelID = pdp.phoneModelID
LEFT OUTER JOIN promotion p ON pdp.promotionID = p.promotionID;
```

	PhoneModelName	PromotionName	PromotionDate
▶	Toàn bộ màu máy	Xmas 2023 10%	2023-12-22 - 2024-01-01
	Toàn bộ màu máy	Back to School 2024 2000k	2024-08-01 - 2024-09-30
	Toàn bộ màu máy	Black Friday 2024 15%	2024-11-20 - 2024-11-30
	Samsung Galaxy S21 Ultra	NULL	NULL
	Samsung Galaxy S21 Plus	NULL	NULL
	Samsung Galaxy S21	NULL	NULL
	Samsung Galaxy S21	NULL	NULL

### 3.1.3 Lấy danh sách các mẫu điện thoại có giá lớn hơn mức giá trung bình và đã bán được trên 50 máy

Code:

```
SELECT pm.name AS PhoneModelName, pmo.name AS PhoneOptionName, pmo.price AS Price,
       COUNT(od.phoneID) AS TotalModelSold FROM phone_model pm
LEFT JOIN phone p ON pm.phoneModelID = p.phoneModelID
LEFT JOIN phone_model_option pmo ON pm.phoneModelID = pmo.phoneModelID
LEFT JOIN order_detail od ON p.phoneID = od.phoneID
GROUP BY pm.phoneModelID, pmo.phoneModelOptionID
HAVING pmo.price > (SELECT AVG(price) FROM phone_model_option) AND COUNT(od.phoneID) >
50;
```

	PhoneModelName	PhoneOptionName	Price	TotalModelSold
▶	Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Green 4/256GB	21026000	85
	Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Green 12/128GB	21358000	85
	Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Gold 16/512GB	21494000	85
	Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Gold 4/256GB	20566000	85
	Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Gold 12/128GB	20902000	85
	Samsung Galaxy S21	Samsung Galaxy S21 Green 12/256GB	18014000	100
	Samsung Galaxy S21	Samsung Galaxy S21 Green 8/512GB	19562000	100

### 3.1.4 Lấy tổng doanh thu của từng cửa hàng từ bảng orders và order\_detail

Code:

```
SELECT s.name AS StoreName, SUM(od.finalPrice) AS TotalRevenue FROM orders o
INNER JOIN order_detail od ON o.orderID = od.orderID
INNER JOIN store s ON o.fromStoreID = s.storeID
GROUP BY s.name
ORDER BY TotalRevenue DESC;
```

	StoreName	TotalRevenue
▶	HCM - 229 Nguyễn Thị Tú	322510000
	NB - Trần Hưng Đạo	282258000
	HCM - Lê Đức Thọ	273712000
	VT - đường 30/4 (Vũng Tàu)	250522000
	HN - 753 Giải Phóng	220450000
	CM - Nguyễn Tất Thành	204444000
	HCM-362A Nguyễn Văn Nghi	202658000
	STR - Hai Bà Trưng	199862000

### 3.1.5 Lấy số lượng điện thoại tồn kho

Code:

```
SELECT s.name AS StoreName, COUNT(p.phoneID) AS TotalPhonesLeft FROM phone p
INNER JOIN store s ON p.inStoreID = s.storeID
WHERE p.status = 'InStore'
GROUP BY s.name
ORDER BY TotalPhonesLeft DESC;
```

	StoreName	TotalPhonesLeft
▶	8_THB - Đông Hưng (Thái Bình)	20
	HNO - Ngọc Hồi	18
	HCM-182 Nguyễn Thị Minh Khai	17
	HN - 291 Nguyễn Văn Cừ	17
	VT - Quốc Lộ 55	17
	HCM - 287A Lê Văn Quới	16
	1_CBA - Kim Đồng (Cao Bằng)	16

### 3.1.6 Lấy danh sách điện thoại và rating trung bình tương ứng

Code:

```
SELECT pm.name AS PhoneModelName, COUNT(pr.reviewID) AS TotalReviews, AVG(pr.rating) AS
AverageRating FROM phone_review pr
INNER JOIN phone_model pm ON pr.phoneModelID = pm.phoneModelID
GROUP BY pm.name
ORDER BY AverageRating DESC;
```

	PhoneModelName	TotalReviews	AverageRating
▶	Xiaomi Mi 11 Lite	32	3.468750
	iPhone 11	24	3.458333
	iPhone 11 Pro	17	3.411765
	iPhone 11 Pro Max	45	3.355556
	Samsung Galaxy S21 Plus	32	3.343750
	Xiaomi Mi 11 Ultra	17	3.294118
	iPhone 12 Pro	20	3.250000

## 3.2 TRANSACTION (transaction.sql)

Với yêu cầu có ít nhất một TRANSACTION và phải đáp ứng yêu cầu sử dụng ROLLBACK. Trong hệ thống quản lý của ta sẽ có những TRANSACTION sau:

### 3.2.1 Cập nhật thông tin người dùng, in ra lỗi nếu không thành công

```
1 • SET SQL_SAFE_UPDATES = 0;
2 • select * from users;
3 • CALL UpdateUser(
4     1, -- userID
5     'Nguyễn Văn A', -- fullName
6     'example@gmail.com', -- email
7     '0909123456', -- phone
8     '123 Đường ABC', -- address
9     3, -- provinceID
10    31, -- districtID
11    'Customer', -- role
12    31 -- storeID
13 );
14 • select * from users
15 where userID = 1;
```

	userID	fullName	email	phone	address	provinceID	districtID	role	storeID
▶	1	Nguyễn Văn A	example@gmail.com	0909123456	123 Đường ABC	3	31	Customer	31
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



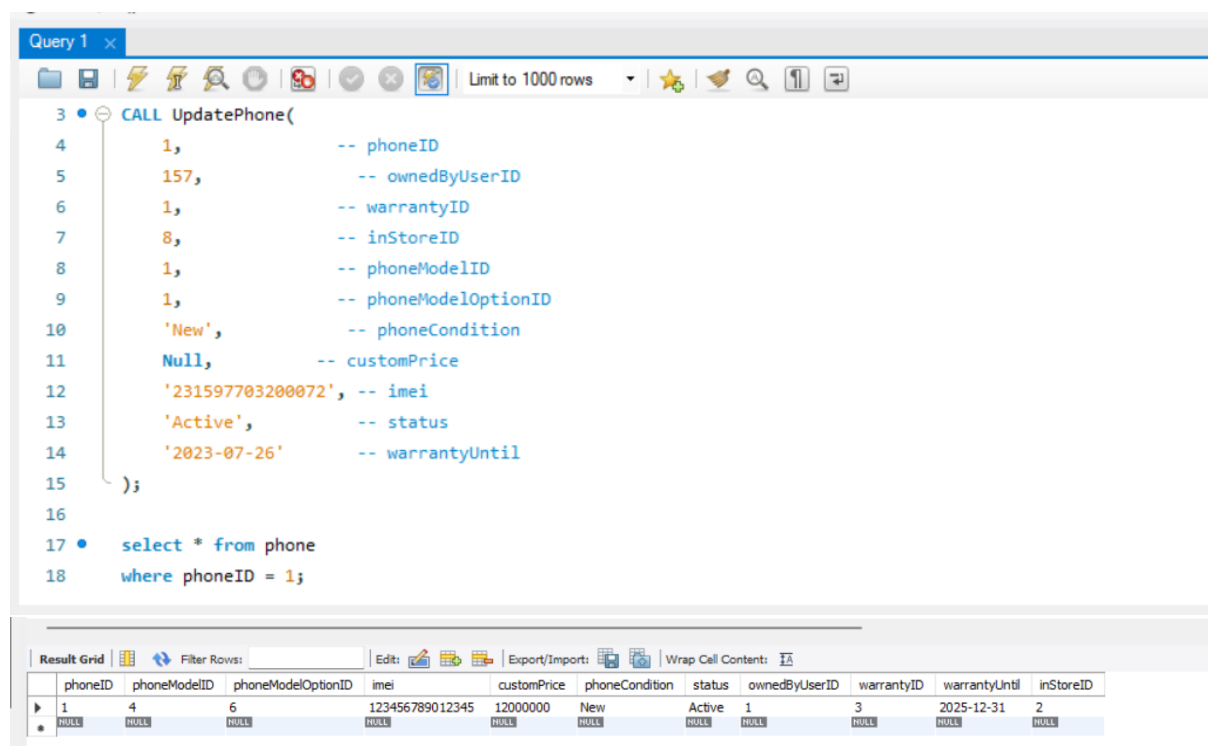
### Code:

```
START TRANSACTION;
BEGIN
    UPDATE users
    SET fullName = fullName, email = email, phone = phone,
        address = address, provinceID = provinceID,
        districtID = districtID, role = role, storeID = storeID
    WHERE userID = userID;

    IF ROW_COUNT() = 0 THEN
        ROLLBACK;
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Update failed. Rolling back';
    END IF;
END;

COMMIT;
```

### 3.2.2 Cập nhật thông tin sản phẩm, in ra lỗi nếu không thành công



The screenshot shows a SQL query editor window titled "Query 1". The query is a stored procedure call: `CALL UpdatePhone(1, 157, 1, 8, 1, 1, 'New', Null, '231597703200072', 'Active', '2023-07-26');` followed by a `select * from phone where phoneID = 1;` query. The result grid below shows the data for the phone with ID 1.

phoneID	phoneModelID	phoneModelOptionID	imei	customPrice	phoneCondition	status	ownedByUserID	warrantyID	warrantyUntil	inStoreID
1	4	6	123456789012345	12000000	New	Active	1	3	2025-12-31	2

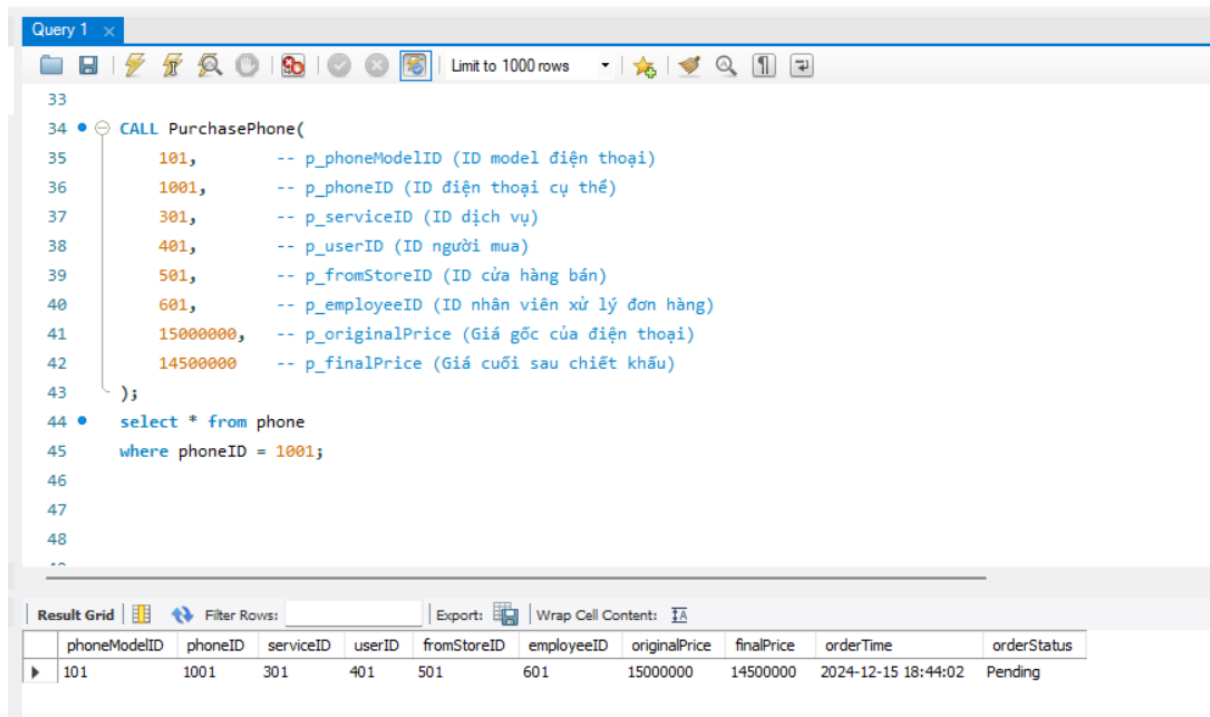
### Code:

```
START TRANSACTION;
BEGIN
    UPDATE phone
    SET ownedByUserID = ownedByUserID, warrantyID = warrantyID, inStoreID = inStoreID,
        phoneModelID = phoneModelID, phoneModelOptionID = phoneModelOptionID,
        phoneCondition = phoneCondition, customPrice = customPrice,
        imei = imei, status = status, warrantyUntil = warrantyUntil
    WHERE phoneID = phoneID;

    IF ROW_COUNT() = 0 THEN
        ROLLBACK;
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Update failed. Rolling back transaction.';
    END IF;
END;

COMMIT;
```

### 3.2.3 Cập nhật thông tin sản phẩm vào đơn hàng khi người dùng chọn mua



The screenshot shows a SQL query editor window titled "Query 1". The query is a stored procedure call `CALL PurchasePhone(` followed by parameters: `101,` (commented as `-- p_phoneModelID (ID model điện thoại)`), `1001,` (`-- p_phoneID (ID điện thoại cụ thể)`), `301,` (`-- p_serviceID (ID dịch vụ)`), `401,` (`-- p_userID (ID người mua)`), `501,` (`-- p_fromStoreID (ID cửa hàng bán)`), `601,` (`-- p_employeeID (ID nhân viên xử lý đơn hàng)`), `15000000,` (`-- p_originalPrice (Giá gốc của điện thoại)`), and `14500000` (`-- p_finalPrice (Giá cuối sau chiết khấu)`). The query ends with `);`. Below the query, there is a `select * from phone` statement with a `where phoneID = 1001;` filter. The result grid at the bottom shows a single row of data for the phone with ID 1001.

phoneModelID	phoneID	serviceID	userID	fromStoreID	employeeID	originalPrice	finalPrice	orderTime	orderStatus
101	1001	301	401	501	601	15000000	14500000	2024-12-15 18:44:02	Pending

**Code:** Tăng số lượng sản phẩm được bán ra lên 1

```
UPDATE phone_model
SET countSold = countSold + 1
WHERE phoneModelID = p_phoneModelID;

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to update phone model sales count. Rolling back.'
;
END IF;
```

**Code:** Tạo thông tin mới trong order

```
INSERT INTO orders (orderTime, status, userID, fromStoreID, employeeID)
VALUES (NOW(), 'Pending', p_userID, p_fromStoreID, p_employeeID);

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to create order. Rolling back.';
END IF;
```

**Code:** Thêm thông tin vào orderdetail và kết thúc

```
SET @newOrderID = LAST_INSERT_ID();
INSERT INTO order_detail (orderId, phoneID, serviceID, originalPrice, finalPrice)
VALUES (@newOrderID, p_phoneID, p_serviceID, p_originalPrice, p_finalPrice);

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to add order details. Rolling back.';
END IF;

COMMIT;
```

### 3.2.4 Cập nhật trạng thái đơn hàng

Query 1 x

Limit to 1000 rows

```
1 • use thegioiiddong;
2 • CALL UpdateOrderStatusToDelivering(80);
3 • select * from orders
4 • where orderID = 80 and status = 'Delivering';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	orderID	orderTime	shippedTime	status	comment	userID	fromStoreID	employeeID
▶	80	2024-12-14 07:16:24	NULL	Delivering	NULL	93	11	261
	158	2024-12-16 05:11:52	NULL	Delivering	NULL	177	25	275
	180	2024-12-17 15:44:00	NULL	Delivering	NULL	99	36	286
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Code:

```
START TRANSACTION;
UPDATE orders
SET status = 'Delivering', shippedTime = NOW()
WHERE orderID = p_orderID AND status = 'Preparing';
COMMIT;
END $$
```

### 3.2.5 Hủy đơn hàng nếu như khách hàng không còn như cầu mua

Query 1 x

Limit to 1000 rows

```
1 • CALL CancelOrder(2098);
2 • SELECT * FROM orders WHERE orderID = 2098;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	orderID	orderTime	shippedTime	status	comment	userID	fromStoreID	employeeID
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Code: Bước 1 –Khôi phục trạng thái sản phẩm về cửa hàng

```
UPDATE phone
SET status = 'InStore'
WHERE phoneID IN (
    SELECT phoneID
    FROM order_detail
    WHERE orderID = p_orderID
);

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to restore phone status. Rolling back transaction
.';
END IF;
```

**Code:** Bước 2 –Xóa thông tin đơn hàng trong orderdetail

```
DELETE FROM order_detail
WHERE orderID = p_orderID;

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to delete order details. Rolling back transaction
.';
END IF;
```

**Code:** Bước 3 –Chuyển thông tin đơn hàng thành Cancelled

```
UPDATE orders
SET status = 'Cancelled'
WHERE orderID = p_orderID;

IF ROW_COUNT() = 0 THEN
    ROLLBACK;
    SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Failed to update order status. Rolling back transaction.
';
END IF;

END;

COMMIT;
```

### 3.3 TRIGGER (trigger.sql)

#### 3.3.1 Tự động cập nhật trạng thái đơn hàng khi đã giao

Trigger after\_update\_shipped\_time tự động cập nhật trạng thái đơn hàng thành 'Completed' khi cột shippedTime được cập nhật và không rỗng.

**Code:**

```
drop trigger if exists after_update_shipped_time $$
create trigger after_update_shipped_time
after update on orders
for each row
begin
    if new.shippedTime is not null then
        update orders
        set status = 'Completed'
        where orderID = new.orderID;
    end if;
end $$
```

#### 3.3.2 Tự động cập nhật trạng thái sản phẩm khi có đơn đặt hàng

Trigger after\_insert\_order\_detail tự động cập nhật trạng thái sản phẩm thành 'Active' khi có đơn đặt hàng mới được thêm vào bảng order\_detail.

**Code:**

```
DROP TRIGGER IF EXISTS after_insert_order_detail $$
CREATE TRIGGER after_insert_order_detail
AFTER INSERT ON order_detail
FOR EACH ROW
BEGIN
    UPDATE phone
    SET status = 'Active'
    WHERE phoneID = NEW.phoneID;
END $$
```

### 3.3.3 Tự động cập nhật trạng thái điện thoại khi đơn hàng bị hủy

Trigger `after_delete_order_detail` tự động cập nhật trạng thái của điện thoại khi đơn hàng bị hủy, cụ thể là khi xóa chi tiết đơn hàng khỏi bảng `order_detail`.

Code:

```
-- DROP TRIGGER IF EXISTS after_delete_order_detail $$
CREATE TRIGGER after_delete_order_detail
AFTER DELETE ON order_detail
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
        SELECT 1
        FROM order_detail
        WHERE phoneID = OLD.phoneID
    ) THEN
        UPDATE phone
        SET status = 'InStore'
        WHERE phoneID IN (SELECT phoneID FROM order_detail WHERE orderID = OLD.orderID);
    END IF;
END $$
```

### 3.3.4 Tự động kích hoạt bảo hành cho sản phẩm mới mua

Trigger `after_update_order_status` sử dụng con trỏ `cur_order_details` để duyệt qua và lấy thông tin, tự động kích hoạt bảo hành cho sản phẩm khi đơn hàng có trạng thái "Completed" (hoàn thành).

Code:

```
DROP TRIGGER IF EXISTS after_update_order_status $$
CREATE TRIGGER after_update_order_status
AFTER UPDATE ON orders
FOR EACH ROW
BEGIN
    DECLARE v_phoneID INT;
    DECLARE v_serviceID INT;
    DECLARE v_warrantyDuration INT;
    DECLARE done INT DEFAULT 0;

    DECLARE cur_order_details CURSOR FOR
        SELECT od.phoneID, od.serviceID
        FROM order_detail od
        INNER JOIN services s ON od.serviceID = s.serviceID
        WHERE od.orderID = NEW.orderID AND s.serviceTypeID = 1 limit 1;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    IF NEW.status = 'Completed' AND OLD.status != 'Completed' THEN
        OPEN cur_order_details;
        read_order_details: LOOP
            FETCH cur_order_details INTO v_phoneID, v_serviceID;

            IF done = 1 THEN
                LEAVE read_order_details;
            END IF;

            SELECT warrantyDuration INTO v_warrantyDuration
            FROM warranty
            WHERE warrantyID = v_serviceID;

            UPDATE phone
            SET warrantyUntil = DATE_ADD(NEW.shippedTime, INTERVAL v_warrantyDuration
DAY)
            WHERE phoneID = v_phoneID;
        END LOOP;

        CLOSE cur_order_details;
    END IF;
END $$
```

### 3.3.5 Tự động cập nhật số lượng đã bán của phone\_model

Trigger `after_order_insert` tự động cập nhật số lượng đã bán của một mẫu điện thoại mỗi khi có đơn hàng mới được thêm vào.

Code:

```
DROP TRIGGER IF EXISTS after_order_insert $$
CREATE TRIGGER after_order_insert
AFTER INSERT ON order_detail
FOR EACH ROW
BEGIN
    UPDATE phone_model
    SET countSold = countSold + 1
    WHERE phoneModelID = (
        SELECT phoneModelID
        FROM phone
        WHERE phoneID = NEW.phoneID
    );
END $$
```

### 3.3.6 Kiểm tra dữ liệu trước khi thêm vào bảng orders

Trigger `before_insert_orders` kiểm tra tính hợp lệ của `userID` và `employeeID` trước khi thêm một đơn hàng mới vào bảng `orders`.

Code:

```
DROP TRIGGER IF EXISTS before_insert_orders $$

CREATE TRIGGER before_insert_orders
BEFORE INSERT ON orders
FOR EACH ROW
BEGIN
    IF NOT EXISTS (
        SELECT 1 FROM users WHERE userID = NEW.userID AND role = 'Customer'
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid userID: Must be a Customer';
    END IF;

    IF NOT EXISTS (
        SELECT 1 FROM users WHERE userID = NEW.employeeID AND role = 'Employee'
    ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid employeeID: Must be an Employee';
    END IF;
END $$
```

### 3.3.7 Kiểm tra số lượng hàng trước khi thêm vào order\_detail

Trigger `before_insert_order_detail` kiểm tra số lượng hàng có sẵn trong kho trước khi thêm chi tiết đơn hàng vào bảng `order_detail`. Nếu số lượng sản phẩm có sẵn là 0, trigger sẽ báo lỗi "The selected phone model and option are out of stock"

Code:

```
DROP TRIGGER IF EXISTS before_insert_order_detail $$
CREATE TRIGGER before_insert_order_detail
BEFORE INSERT ON order_detail
FOR EACH ROW
BEGIN
    DECLARE available_in_stock INT;
    SELECT COUNT(*)
    INTO available_in_stock
    FROM phone
    WHERE phoneID = NEW.phoneID
    AND phoneModelID = (SELECT phoneModelID FROM phone WHERE phoneID = NEW.phoneID)
    AND phoneModelOptionID = (SELECT phoneModelOptionID FROM phone WHERE phoneID = NEW
    .phoneID)
```

```

        AND status = 'InStore';

    IF available_in_stock = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'The selected phone model and option are out of stock.';
    END IF;
END $$

```

### 3.4 PROCEDURE (procedure.sql)

#### 3.4.1 Lọc sản phẩm theo nhà sản xuất

Code:

```

DROP PROCEDURE IF EXISTS GetPhonesByManufacturer $$
CREATE PROCEDURE GetPhonesByManufacturer(IN manufacturerName VARCHAR(50))
BEGIN
    SELECT pm.name AS PhoneName, pm.countView, pm.countSold,
    mf.name AS Manufacturer FROM phone_model pm
    JOIN manufacturer mf ON pm.manufacturerID = mf.manufacturerID
    WHERE mf.name = manufacturerName;
END $$

```

Query 1 x

Limit to 1000 rows

```

1 • use thegioididong;
2 • select * from manufacturer;
3 • call GetPhonesByManufacturer('Samsung')
4

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

PhoneName	countView	countSold	Manufacturer
Samsung Galaxy S21 Ultra	121210	52521	Samsung
Samsung Galaxy S21 Plus	2210	1120	Samsung
Samsung Galaxy S21	3120	1512	Samsung
Samsung Galaxy S20 Ultra	4120	2730	Samsung
Samsung Galaxy S20	610	310	Samsung

#### 3.4.2 Lọc sản phẩm theo giá tiền

Code:

```

DROP PROCEDURE IF EXISTS GetPhonesByPrice $$
CREATE PROCEDURE GetPhonesByPrice(IN minPrice INT, IN maxPrice INT)
BEGIN
    SELECT pm.name AS PhoneModel, pmo.name AS OptionName, pmo.price AS Price,
    mf.name AS Manufacturer FROM phone_model pm
    JOIN phone_model_option pmo ON pm.phoneModelID = pmo.phoneModelID
    JOIN manufacturer mf ON pm.manufacturerID = mf.manufacturerID
    WHERE pmo.price BETWEEN minPrice AND maxPrice
    ORDER BY pmo.price ASC;
END $$

```

Query 1 x

Limit to 1000 rows

```

1 • use thegioididong;
2
3 • call GetPhonesByPrice(3000000, 6000000);

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

PhoneModel	OptionName	Price	Manufacturer
iPhone 11	iPhone 11 Black 4/128GB	4902000	Apple
iPhone 11	iPhone 11 Purple 8/64GB	5126000	Apple
iPhone 11	iPhone 11 Black 8/64GB	5150000	Apple

### 3.4.3 Lọc sản phẩm theo thông số kỹ thuật

Code:

```
DROP PROCEDURE IF EXISTS GetPhonesByTechSpec $$
CREATE PROCEDURE GetPhonesByTechSpec( IN p_techSpecID INT,
    IN p_infoNumMin DECIMAL(10,3),
    IN p_infoNumMax DECIMAL(10,3))
BEGIN
    SELECT pm.name AS PhoneModel, pmo.name AS OptionName,
        pts.infoNum AS SpecValue, pts.infoText AS TechSpecDescription FROM phone_model
    pm
    INNER JOIN phone_model_option pmo ON pm.phoneModelID = pmo.phoneModelID
    INNER JOIN phone_tech_spec pts ON pmo.phoneModelOptionID = pts.phoneModelOptionID
    WHERE pts.techSpecID = p_techSpecID
    AND pts.infoNum BETWEEN p_infoNumMin AND p_infoNumMax;
END $$
```

429 • CALL GetPhonesByTechSpec(2, 0, 2.6);

PhoneModel	OptionName	Price	SpecValue	TechSpecDescription
Samsung Galaxy S21 Plus	Samsung Galaxy S21 Plus Gold 12/128GB	20902000	2.000	Snapdragon 835 2.0GHz
Samsung Galaxy S21	Samsung Galaxy S21 Silver 8/512GB	17530000	2.600	Exynos 990 2.6GHz
Samsung Galaxy S20 Ultra	Samsung Galaxy S20 Ultra Blue 4/128GB	11822000	2.100	Exynos 990 2.1GHz
Samsung Galaxy S20 Ultra	Samsung Galaxy S20 Ultra Blue 16/512GB	15094000	2.100	Snapdragon 855 2.1GHz
Samsung Galaxy S20	Samsung Galaxy S20 Red 6/256GB	7098000	2.500	Snapdragon 888 2.5GHz
iPhone 12	iPhone 12 Red 16/256GB	16094000	2.100	Apple A13 Bionic 2.1GHz

### 3.4.4 Đề xuất sản phẩm bán chạy theo tháng

Code:

```
DROP PROCEDURE IF EXISTS GetBestSellingPhonesByMonth $$
CREATE PROCEDURE GetBestSellingPhonesByMonth(IN targetMonth INT, IN targetYear INT)
BEGIN
    SELECT pm.name AS PhoneModel, COUNT(DISTINCT(CONCAT(od.orderID, '-', od.phoneID)))
    AS TotalSold FROM phone_model pm
    JOIN phone p ON p.phoneModelID = pm.phoneModelID
    JOIN order_detail od ON p.phoneID = od.phoneID
    JOIN orders o ON o.orderID = od.orderID
    WHERE MONTH(o.orderTime) = targetMonth AND YEAR(o.orderTime) = targetYear
    GROUP BY pm.phoneModelID
    ORDER BY TotalSold DESC
    LIMIT 5;
END $$
```

Query 1 x

1 • use thegioididong;

2 • call GetBestSellingPhonesByMonth(12, 2024);

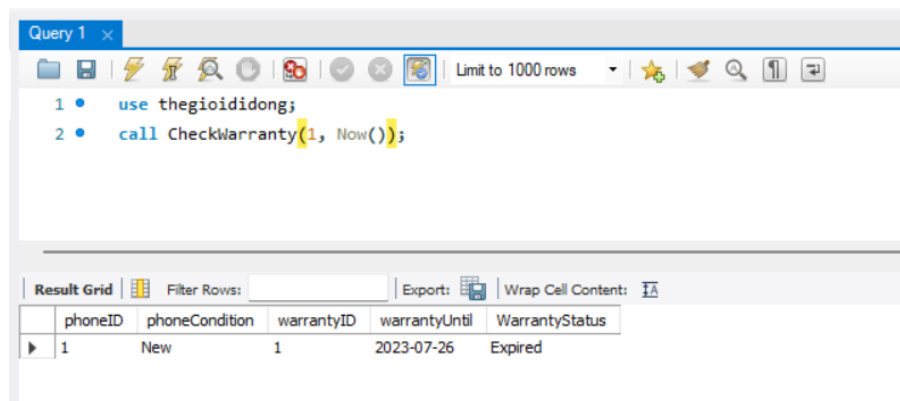
PhoneModel	TotalSold
Xiaomi Mi 10 Pro	5
Samsung Galaxy S21	3
Samsung Galaxy S20 Ultra	3
Samsung Galaxy S21 Plus	2
vivo X50	2



### 3.4.5 Kiểm tra bảo hành có còn khả dụng

Code:

```
DROP PROCEDURE IF EXISTS CheckWarranty $$
CREATE PROCEDURE CheckWarranty(IN phoneID INT, IN currentDate DATE)
BEGIN
    SELECT p.phoneID, p.phoneCondition, p.warrantyID, p.warrantyUntil,
           CASE
               WHEN p.warrantyUntil >= currentDate THEN 'Valid'
               ELSE 'Expired'
           END AS WarrantyStatus
    FROM phone p
    WHERE p.phoneID = phoneID;
END $$
```



### 3.4.6 Tìm kiếm địa chỉ cửa hàng gần người dùng nhất

Code:

```
DROP PROCEDURE IF EXISTS GetNearbyStores $$
CREATE PROCEDURE GetNearbyStores(IN userLongitude DECIMAL(10, 5), IN userLatitude
DECIMAL(10, 5))
BEGIN
    SELECT storeID, name AS StoreName, address, phoneNumber,
           SQRT(POW(gps_longitude - userLongitude, 2) + POW(gps_latitude - userLatitude, 2))
           ) AS Distance FROM store
    ORDER BY Distance ASC;
END $$
```

Query 1 x

Limit to 1000 rows

```

1 • use thegioididong;
2   CALL GetNearbyStores(105.854444, 21.028511);

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [fA](#)

	storeID	StoreName	address	phoneNumber	Distance
▶	46	1_YBA - Điện Biên (Yên Bái)	763 Điện Biên, Tổ 48, P. Minh Tân, TP Yên Bái, T...	(04)37.757.757	118.80527812498778
	34	1_CBA - Kim Đồng (Cao Bằng)	số 1 Kim Đồng, Phường Hợp Giang, TP.Cao Bắ...	(04)37.757.757	119.10070146696745
	14	1_HNO - Phố Thanh Vy	04-06 Phố Thanh Vy, P.Sơn Lộc, Thị Xã Sơn Tây,...	(04)37.757.757	119.64600290722504
	15	1_HNO - Phố Nhổn	Quốc Lộ 32, 18 Phố Nhổn, Xã Xuân Phương, H. ...	(04)37.757.757	119.85909436019321
	12	HNO - Xuân Mai	Số 34,Tổ 2, Khu Xuân Hà, Thị Trấn Xuân Mai, H...	(04)37.757.757	119.86104834974412
	10	HNO - Cổ Nhuế	338 Phạm Văn Đồng, Cổ Nhuế, Hà Nội	(04)37.757.757	119.87774341034995
	8	HN - Thôn Phan Xá	Thôn Phan Xá, Xã Uyển, Huyện Đông Anh, Hà ...	(04)37.757.757	119.88448610836892
	2	HN - 351 Cầu Giấy	351 Cầu Giấy, P. Dịch Vọng, Q. Cầu Giấy, Hà Nội	(04)37.757.757	119.91169062146402
	4	HNO - Lạc Long Quân	Số 199 - 201 -203, Lạc Long Quân, P.Nghĩa Đô,...	(04)37.757.757	119.9133824137644
	13	HN - 403 Quang Trung	403 Quang Trung, P. Quang Trung, Q. Hà Đông...	(04)37.757.757	119.94616741965415
	1	HN -11A Thái Hà	11A Thái Hà, P.Trung Liệt, Q. Đống Đa, Hà Nội	(04)37.757.757	119.9486003156923
	16	1_HNO - Nguyễn Trãi	Số 497 Đường Nguyễn Trãi, Q. Thanh Xuân, TP...	(04)37.757.757	119.95073602581061
	6	HN - 215 Xã Đàn	215 Xã Đàn, P.Nam Đồng, Q. Đống Đa, TP. Hà Nội	(04)37.757.757	119.95474944677972
	5	HN - 291 Nguyễn Văn Cừ	291Nguyễn Văn Cừ, P. Ngọc Lâm, Q. Long Biên...	(04)37.757.757	119.96464934662919
	7	HN - Ngô Gia Tự	807 Ngô Gia Tự, P. Đức Giang, Q.Long Biên, Hà ...	(04)37.757.757	119.9665028441177
	3	HN - Lê Duẩn	Số 468 - 470 - 472, Lê Duẩn,P.Phương Liền, Q...	(04)37.757.757	119.9667701194781

### 3.4.7 Xuất hóa đơn

Code:

```

DROP PROCEDURE IF EXISTS ExportInvoice $$
CREATE PROCEDURE ExportInvoice(IN orderID INT)
BEGIN
    SELECT o.orderID AS OrderID,
           CASE WHEN od.serviceID = 0 THEN pmo.name ELSE s.name END AS ItemName,
           o.orderTime AS OrderTime, od.originalPrice AS OriginalPrice,
           od.finalPrice AS FinalPrice, p.name AS DiscountName FROM orders o
    JOIN order_detail od ON o.orderID = od.orderID
    JOIN phone ph ON ph.phoneID = od.phoneID
    LEFT JOIN phone_model_option pmo ON ph.phoneModelOptionID = pmo.phoneModelOptionID
    LEFT JOIN services s ON od.serviceID = s.serviceID
    LEFT JOIN promotion p ON od.promotionID = p.promotionID
    WHERE o.orderID = orderID;
END $$

```

Query 1 x

Limit to 1000 rows

```

1 • use thegioididong;
2   CALL ExportInvoice(3);

```

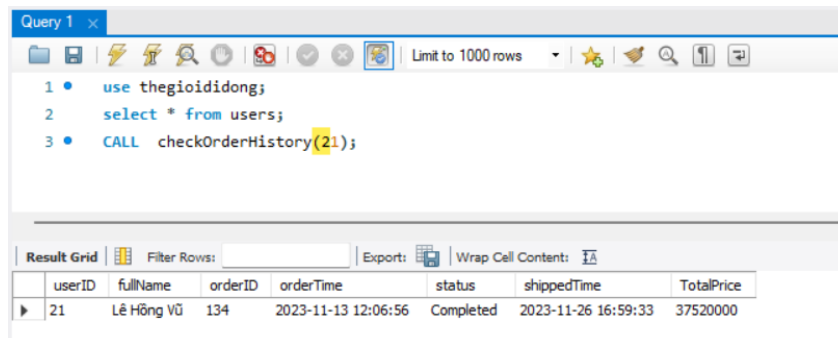
Result Grid | Filter Rows: | Export: | Wrap Cell Content: [fA](#)

	OrderID	ItemName	OrderTime	OriginalPrice	FinalPrice	DiscountName
▶	3	Samsung Galaxy S21 Ultra Green 6/256GB	2024-07-17 14:44:54	21820000	21820000	Không có gì
	3	Sạc nhanh 30W	2024-07-17 14:44:54	300000	300000	Không có gì
	3	vivo X60 Pro Black 4/256GB	2024-07-17 14:44:54	23526000	23526000	Không có gì
	3	Kính cường lực cho Samsung/Xiaomi/vivo	2024-07-17 14:44:54	500000	500000	Không có gì
	3	Sạc nhanh 18W	2024-07-17 14:44:54	100000	70000	Giảm giá 30% cho sạc nhanh

### 3.4.8 Lịch sử mua hàng của khách hàng

Code:

```
DROP PROCEDURE IF EXISTS checkOrderHistory $$
CREATE PROCEDURE checkOrderHistory(IN userID INT)
BEGIN
    SELECT o.userID, u.fullName, o.orderID, o.orderTime, o.status, o.shippedTime, SUM(od
    .finalPrice) AS TotalPrice FROM orders o
    JOIN order_detail od ON o.orderID = od.orderID
    JOIN users u ON u.userID = o.userID
    WHERE o.userID = 21
    GROUP BY o.orderID;
END $$
```



Query 1

1 • use thegioiiddong;

2 • select \* from users;

3 • CALL checkOrderHistory(21);

Result Grid

	userID	fullName	orderID	orderTime	status	shippedTime	TotalPrice
▶	21	Lê Hồng Vũ	134	2023-11-13 12:06:56	Completed	2023-11-26 16:59:33	37520000

### 3.4.9 Thêm chi tiết đơn hàng

Code:

```
DROP PROCEDURE IF EXISTS AddOrderDetail $$
CREATE PROCEDURE AddOrderDetail(IN p_orderID INT, IN p_phoneID INT, IN p_serviceID INT,
    IN p_promotionID INT)
BEGIN
    DECLARE v_original_price INT; DECLARE v_final_price INT;
    DECLARE v_original_service_price INT; DECLARE v_final_service_price INT;
    DECLARE v_discount_percentage DECIMAL(4,2); DECLARE v_discount_fixed INT;
    DECLARE v_fixed_new_price INT; DECLARE v_phone_model_id INT;
    DECLARE v_phone_model_option_id INT; DECLARE v_service_type_id INT;
    DECLARE v_start_promotion_date DATE; DECLARE v_end_promotion_date DATE;
    DECLARE v_order_time DATE;

    SELECT o.orderTime, p.startDate, p.endDate
    INTO v_order_time, v_start_promotion_date, v_end_promotion_date FROM orders o
    JOIN promotion p ON p.promotionID = p_promotionID WHERE o.orderID = p_orderID;

    SELECT COALESCE(customPrice, (SELECT price FROM phone_model_option WHERE
    phone_model_option.phoneModelOptionID = phone.phoneModelOptionID)),
    phoneModelID, phoneModelOptionID
    INTO v_original_price, v_phone_model_id, v_phone_model_option_id
    FROM phone WHERE phoneID = p_phoneID;

    SELECT discountPercent, discountFixed, fixedNewPrice
    INTO v_discount_percentage, v_discount_fixed, v_fixed_new_price
    FROM promotion_detail_phone
    WHERE promotionID = p_promotionID AND (phoneModelID = 0 OR phoneModelID =
    v_phone_model_id)
    AND (phoneModelOptionID = 0 OR phoneModelOptionID = v_phone_model_option_id);

    IF v_fixed_new_price IS NOT NULL THEN
        SET v_final_price = v_fixed_new_price;
    ELSEIF v_discount_percentage IS NOT NULL THEN
        SET v_final_price = v_original_price * (1 - (v_discount_percentage / 100));
    ELSEIF v_discount_fixed IS NOT NULL THEN
        SET v_final_price = GREATEST(0, v_original_price - v_discount_fixed);
    ELSE
        SET v_final_price = v_original_price;
    END IF;
```

```

END IF;

SELECT price, serviceTypeID
INTO v_original_service_price, v_service_type_id
FROM services WHERE serviceID = p_serviceID;

IF EXISTS (SELECT 1 FROM promotion_detail_service WHERE promotionID = p_promotionID)
THEN
    SELECT discountPercent, discountFixed, fixedNewPrice
    INTO v_discount_percentage, v_discount_fixed, v_fixed_new_price
    FROM promotion_detail_service
    WHERE promotionID = p_promotionID AND serviceTypeID = v_service_type_id LIMIT 1;

    IF v_discount_percentage IS NOT NULL THEN
        SET v_final_service_price = v_original_service_price * (1 -
v_discount_percentage / 100);
    ELSEIF v_discount_fixed IS NOT NULL THEN
        SET v_final_service_price = GREATEST(0, v_original_service_price -
v_discount_fixed);
    ELSEIF v_fixed_new_price IS NOT NULL THEN
        SET v_final_service_price = v_fixed_new_price;
    ELSE
        SET v_final_service_price = v_original_service_price;
    END IF;
ELSE
    SET v_final_service_price = v_original_service_price;
END IF;

IF v_order_time NOT BETWEEN v_start_promotion_date AND v_end_promotion_date THEN
    SET v_final_price = v_original_price;
    SET v_final_service_price = v_original_service_price;
END IF;

SET v_final_price = v_final_price + v_final_service_price;

INSERT INTO order_detail(orderID, phoneID, serviceID, promotionID, originalPrice,
finalPrice)
VALUES (p_orderID, p_phoneID, p_serviceID, p_promotionID, v_original_price,
v_final_price);
END $$

```

## Giải thích

Thủ tục lưu trữ `AddOrderDetail` trong MySQL được thiết kế để thêm thông tin chi tiết đơn hàng vào bảng `order_detail`, đồng thời tính toán giá điện thoại và dịch vụ sau khi áp dụng các khuyến mãi.

Thủ tục bắt đầu bằng việc khai báo các biến lưu trữ thông tin cần thiết. Sau đó, thời gian đặt hàng và thời gian khuyến mãi được lấy từ cơ sở dữ liệu để kiểm tra tính hợp lệ. Tiếp theo, giá gốc và các loại khuyến mãi áp dụng cho điện thoại và dịch vụ được xử lý:

- Nếu có giá cố định mới (`fixedNewPrice`), giá này sẽ được ưu tiên.
- Nếu không, các mức giảm giá phần trăm (`discountPercent`) hoặc giảm giá cố định (`discountFixed`) sẽ được áp dụng.
- Trường hợp không có khuyến mãi hoặc đơn hàng không nằm trong thời gian khuyến mãi, giá gốc sẽ được giữ nguyên.

Sau khi tính toán giá cuối cùng (bao gồm cả điện thoại và dịch vụ), thông tin chi tiết đơn hàng được chèn vào bảng `order_detail` với các trường như giá gốc, giá cuối cùng và các thông tin liên quan khác. Thủ tục này đảm bảo tính chính xác trong việc quản lý giá cả và áp dụng khuyến mãi.

Query 1

```

1 • use thegioididong;
2 • CALL AddOrderDetail(1, 308, 0, 0);
3 • select * from order_detail
4 • where orderID = 1 and phoneID = 308;

```

Result Grid

orderID	phoneID	serviceID	promotionID	originalPrice	finalPrice
1	308	0	1	14534000	14534000
*	NULL	NULL	NULL	NULL	NULL

# Tổng kết

Trong báo cáo này, ta đã đi qua hết các công đoạn thiết lập hệ thống cơ sở dữ liệu, cùng với quan hệ, ràng buộc đầy đủ, chặt chẽ. Chúng em đã thêm vào hệ thống một nguồn dữ liệu mẫu đa dạng với **hơn 6000 giá trị trong 21 bảng**. Mô hình và các truy vấn đều được thiết kế đảm bảo dữ liệu toàn vẹn.

Hơn nữa, nhóm đã thực hiện nhiều thao tác truy vấn, làm việc với hệ cơ sở dữ liệu, đáp ứng vượt yêu cầu đề ra. Nhiều vấn đề trong thực tế về quản lý dữ liệu, tra cứu đã được giải quyết thông qua các câu lệnh SQL. Hệ thống có thể làm việc trơn tru và có thể mở rộng trong tương lai.

Mặc dù vậy, ta vẫn còn có thể cải tiến hơn nữa. Từ những gì mà dự án đã thực hiện được, nhóm xin trình bày một số đề xuất cải tiến:

1. Mở rộng mô hình, cải thiện hiệu suất để phù hợp với những thao tác phức tạp hơn;
2. Xây dựng giao diện trực quan, giúp người sử dụng và các nhà quản lý dễ dàng tiếp cận với các tính năng của hệ thống;
3. Áp dụng song song với nhiều hệ cơ sở dữ liệu khác, để đem lại hiệu suất cao hơn với đặc thù dữ liệu của hệ thống.

Trong quá trình hoàn thành bài tập lớn, nhóm em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Chúng em đã cố gắng thực hiện nhiều hơn mức yêu cầu, tuy nhiên không thể tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của thầy cô, để báo cáo và kỹ năng của chúng em ngày được hoàn thiện hơn.

Trân trọng,

Nhóm Loli.

# Tài liệu tham khảo

[1] Thế giới di động. Trang chủ. <https://www.thegioididong.com/dtdd>, 2024.

[2] Cờu đường thần công. Dữ liệu cào được từ trang Thế giới di động. <https://drive.google.com/file/d/1fleyJt5LM4pVb23rivXwA4eCRVzebnLi/view?usp=sharing>, 2013.