**BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ MT**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

****

**BÁO CÁO BÀI TẬP LỚN HỌC PHẦN HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**HỆ THỐNG QUẢN LÝ KHÁCH SẠN**

**Giảng viên hướng dẫn :** TS. Nguyễn Ngọc Quỳnh Châu

**Lớp :** 65KTPM

**Nhóm sinh viên thực hiện :** Nguyễn Hồng Phúc : 2351170611

Ngô Đức Nam Khánh : 2351060452

Nguyễn Hải Ninh : 2351170609

Ngô Quang Tùng : 2351170631

Hà Nội, năm 2026

[**PHẦN I. THIẾT KẾ CƠ SỞ DỮ LIỆU 4**](#_c82ydgm6w5v5)

[1. Bài toán đặt ra 4](#_ja7opv45b53j)

[2. Mô tả quy tắc quản lý & Ràng buộc 5](#_hy7i60x31xxp)

[2.1. Quy tắc quản lý 5](#_nikvvsrw7zlg)

[2.2. Các ràng buộc toàn vẹn (Constraints) 6](#_rqqdwdc0ddik)

[2.3. Các công thức tính toán 11](#_v0q5x8taee3w)

[3. Mô hình Thực thể - Liên kết (ER Model) 12](#_cxfop4oy1umt)

[4. Mô hình Quan hệ (Relational Model) 13](#_gpsm679pjxse)

[**PHẦN II. QUẢN TRỊ CƠ SỞ DỮ LIỆU 15**](#_q262i0qnp8i3)

[**1. Thành viên: Nguyễn Hồng Phúc 15**](#_dvup8v732dmv)

[1.1. Function 15](#_ohs7prb1z32e)

[a. fn\_calculate\_room\_price 15](#_1dy5aztkktjs)

[b. fn\_check\_room\_availability 16](#_bq63y8cqw1fl)

[1.2. View 16](#_p39qwrhp0v7i)

[a. vw\_room\_availability 16](#_hu9lgl6wkzwv)

[b. vw\_occupancy\_statistics 17](#_ysyasi6nocbe)

[1.3. Procedure + Transaction 17](#_kppl3fk0fx25)

[a. sp\_create\_reservation 17](#_fwcpg9veolj2)

[b. sp\_cancel\_reservation 19](#_9fdalamudf6w)

[1.4. Trigger 20](#_c6zr6vbb3gay)

[a. trg\_reservation\_status\_change 20](#_fz42cb46637p)

[b. trg\_reservation\_audit 21](#_2w1y3zmwoabd)

[1.5. Cursor 21](#_xdp2iei2yfdd)

[a. checkin\_cursor 21](#_tsd0cq8rk8ex)

[b. noshow\_cursor 23](#_yjpki1xbmn4v)

[**2. Thành viên: Ngô Đức Nam Khánh 24**](#_n45jwnnurlqg)

[2.1. Function 24](#_3jaawqtsngcq)

[a. fn\_calculate\_total\_bill 24](#_32lgtzyqjmc8)

[b. fn\_calculate\_loyalty\_points 25](#_wn8ssngmya05)

[c. fn\_get\_customer\_tier 26](#_hwat4au23z3c)

[2.2. View 26](#_d0wgwskpywkr)

[a. vw\_daily\_revenue\_report 26](#_ile50g2hc92e)

[b. vw\_outstanding\_payments 26](#_78zflce66u8w)

[2.3. Procedure + Transaction 27](#_6glmwmdxnjfg)

[a. sp\_process\_payment 27](#_hopx971ai67m)

[b. sp\_generate\_invoice 28](#_8o31fvnhkpng)

[2.4. Trigger 29](#_gxeu1p9j60a8)

[a. trg\_payment\_loyalty\_update 29](#_bdib4zgmqjur)

[b. trg\_payment\_audit 30](#_23qaryvi0jeh)

[2.5. Cursor 30](#_1hxj8mwiv2j3)

[a. Nhắc nhở thanh toán 30](#_t8f0648nhs67)

[b. Tạo báo cáo doanh thu tháng 32](#_cucbrveduyi0)

[**3. Thành viên: Nguyễn Hải Ninh 33**](#_txcqumhdvb6d)

[3.1. Function 33](#_jd9vgv1ci7n0)

[a. fn\_get\_customer\_discount\_rate 33](#_wu9sjkrd12b9)

[b. fn\_get\_customer\_statistics 34](#_e25tdwb1ady2)

[3.2. View 34](#_rohawexb5x5r)

[a. vw\_customer\_history 34](#_ka5cejt15gmc)

[b. vw\_popular\_services 35](#_9qcp43a58yye)

[3.3. Procedure + Transaction 35](#_wi6iafaowcez)

[a. sp\_register\_customer 35](#_jh14zpyytxx0)

[b. sp\_add\_service\_to\_reservation 36](#_gfjahwkb0jeq)

[3.4. Trigger 37](#_ts6152x2c51v)

[a. trg\_customer\_tier\_upgrade 37](#_ynczu7on73vx)

[b. trg\_award\_service\_loyalty\_points 38](#_xl9afluot8jk)

[3.5. Cursor 39](#_z9c1wrpzqyjq)

[a. sp\_process\_loyalty\_tier\_upgrades 39](#_hn67pw6x8tvi)

[b. sp\_generate\_service\_usage\_report 40](#_r1ijashcsfnc)

[**4. Thành viên: Ngô Quang Tùng 41**](#_dgim5vzfvsiy)

[4.1. Function 41](#_7ek1s5abmv5x)

[a. fn\_calculate\_sla\_status 41](#_pdrtyfvb9cyi)

[b. fn\_calculate\_maintenance\_cost 42](#_99tghrlqfivq)

[4.2. View 43](#_pfu1chjlovgr)

[a. vw\_maintenance\_dashboard 43](#_hkqbje19ewcb)

[b. vw\_employee\_performance 44](#_9t4h92oeet5n)

[c. vw\_maintenance\_cost\_statistics 44](#_qoqhkjw0zorb)

[4.3. Procedure + Transaction 46](#_w21lhctbj6dv)

[a. sp\_create\_maintenance\_request 46](#_na62wtoj6uf2)

[b. sp\_complete\_maintenance 48](#_w2vrwfw0hdbj)

[4.4. Trigger 50](#_wpu42vcodm81)

[a. trg\_room\_status\_history 50](#_7fj12jr4quca)

[b. trg\_update\_employee\_availability 51](#_1oagyc9nqk5)

[c. trg\_restore\_employee\_availability 51](#_7zj5v0ehy4wh)

[4.5. Cursor 52](#_2rhq4dgoylbx)

[a. Cursor 1: Phân công tự động yêu cầu bảo trì 52](#_qx72mfa3frr7)

[b. Cursor 2: Thống kê task theo nhân viên 54](#_ui9aevpkjdny)

[**Kết luận 56**](#_vss9urgdv8lw)

[**Lời cảm ơn 57**](#_guflpdrfu0xa)

# 

# PHẦN I. THIẾT KẾ CƠ SỞ DỮ LIỆU

## 1. Bài toán đặt ra

* **Mô tả:** Trong ngành khách sạn hiện đại, việc quản lý vận hành thủ công gây ra nhiều bất cập như: thông tin phòng không đồng bộ, đặt phòng trùng lặp, thanh toán sai sót, và khó theo dõi lịch sử khách hàng.
* **Hệ thống Quản lý Khách sạn (Hotel Management System - HMS)** được xây dựng nhằm giải quyết các vấn đề trên cho khách sạn quy mô vừa, với các mục tiêu:
  + **Quản lý phòng:** Theo dõi trạng thái phòng (Available, Occupied, Maintenance, Cleaning), loại phòng, giá theo mùa/cuối tuần
  + **Quản lý đặt phòng:** Tạo, sửa, hủy đặt phòng; xử lý check-in/check-out; phát hiện no-show
  + **Quản lý thanh toán:** Xử lý thanh toán đa phương thức, hoàn tiền, theo dõi công nợ
  + **Quản lý khách hàng:** Đăng ký khách, tích điểm, phân hạng thành viên (Bronze/Silver/Gold/Platinum)
  + **Quản lý dịch vụ:** Ghi nhận dịch vụ sử dụng, tính chi phí dịch vụ vào hóa đơn
  + **Quản lý vận hành:** Yêu cầu bảo trì, phân công nhân viên, theo dõi SLA
  + **Quản lý nhân sự:** Ca làm việc, hiệu suất nhân viên

## 2. Mô tả quy tắc quản lý & Ràng buộc

### 2.1. Quy tắc quản lý

***a. Quy tắc về Khách hàng***

* Một khách hàng có thể có nhiều đặt phòng (đã hoàn thành, đang ở, sắp tới)
* Một khách hàng có thể có nhiều hóa đơn thanh toán
* Email khách hàng phải là duy nhất trong hệ thống
* Khách hàng mới được tặng 100 điểm chào mừng và hạng Bronze
* Hạng thành viên chỉ được nâng lên, không bị hạ xuống

***b. Quy tắc về Phòng***

* Mỗi phòng thuộc một loại phòng (Standard, Deluxe, Suite...)
* Một phòng có thể có nhiều đặt phòng (không trùng thời gian)
* Một phòng có thể có nhiều yêu cầu bảo trì
* Phòng chỉ có thể ở một trạng thái tại một thời điểm: Available, Occupied, Maintenance, Cleaning, Reserved
* Giá phòng cơ bản phải lớn hơn 0

***c. Quy tắc về Đặt phòng***

* Mỗi đặt phòng thuộc về một khách hàng và một phòng
* Một đặt phòng có thể có nhiều thanh toán (thanh toán nhiều lần)
* Khi đặt phòng bắt buộc phải đặt cọc 30% tiền phòng ban đầu
* Một đặt phòng có thể có nhiều dịch vụ sử dụng
* Ngày check-out phải sau ngày check-in
* Số khách không được vượt quá sức chứa của phòng
* Không được đặt phòng trùng thời gian với đặt phòng khác đang hoạt động

***d. Quy tắc về Thanh toán***

* Mỗi thanh toán thuộc về một đặt phòng và một khách hàng
* Số tiền thanh toán phải lớn hơn 0 (trừ hoàn tiền thì âm)
* Sau khi thanh toán, tự động cộng điểm loyalty cho khách hàng
* Sau khi thanh toán, tự động cập nhật tổng chi tiêu và kiểm tra nâng hạng

***e. Quy tắc về Dịch vụ***

* Mỗi dịch vụ thuộc một danh mục (Food & Beverage, Spa, Laundry...)
* Một dịch vụ có thể được sử dụng nhiều lần bởi nhiều đặt phòng
* Chỉ thêm dịch vụ khi đặt phòng đang ở trạng thái CheckedIn
* Giá dịch vụ được ghi nhận tại thời điểm sử dụng (không thay đổi khi giá gốc thay đổi)

***f. Quy tắc về Nhân viên***

* Mỗi nhân viên thuộc một phòng ban
* Một nhân viên có thể có nhiều ca làm việc
* Một nhân viên bảo trì có thể được giao nhiều yêu cầu bảo trì
* Nhân viên được giao task ưu tiên cao sẽ đánh dấu bận, không nhận task mới

***g. Quy tắc về Bảo trì***

* Mỗi yêu cầu bảo trì thuộc về một phòng
* Mỗi yêu cầu được giao cho một nhân viên bảo trì
* Yêu cầu Critical/High sẽ chuyển phòng sang trạng thái Maintenance
* Có SLA theo độ ưu tiên: Critical 4h, High 12h, Medium 24h, Low 48h

***h. Quy tắc về Hạng thành viên và Giảm giá***

* Bronze: Chi tiêu < $5,000 → Giảm 0%
* Silver: Chi tiêu ≥ $5,000 → Giảm 5%
* Gold: Chi tiêu ≥ $20,000 → Giảm 10%
* Platinum: Chi tiêu ≥ $50,000 → Giảm 15%
* Đơn hàng ≥ $1,000 được giảm thêm 2%
* Tổng giảm giá tối đa 25%

***i. Quy tắc về Chính sách hủy***

* Giữ lại tiền cọc khi khách hủy
* No-show (không đến): giữ lại tiền cọc

### 2.2. Các ràng buộc toàn vẹn (Constraints)

* Khóa chính (Primary Key)

| **Bảng** | **Khóa chính** | **Kiểu dữ liệu** |
| --- | --- | --- |
| DEPARTMENTS | department\_id | INT IDENTITY(1,1) |
| SERVICE\_CATEGORIES | category\_id | INT IDENTITY(1,1) |
| ROOM\_TYPES | type\_id | INT IDENTITY(1,1) |
| CUSTOMERS | customer\_id | INT IDENTITY(1,1) |
| EMPLOYEES | employee\_id | INT IDENTITY(1,1) |
| ROOMS | room\_id | INT IDENTITY(1,1) |
| SERVICES | service\_id | INT IDENTITY(1,1) |
| RESERVATIONS | reservation\_id | INT IDENTITY(1,1) |
| PAYMENTS | payment\_id | INT IDENTITY(1,1) |
| SERVICES\_USED | usage\_id | INT IDENTITY(1,1) |
| MAINTENANCE\_REQUESTS | request\_id | INT IDENTITY(1,1) |
| EMPLOYEE\_SHIFTS | shift\_id | INT IDENTITY(1,1) |
| ROOM\_STATUS\_HISTORY | history\_id | INT IDENTITY(1,1) |
| AUDIT\_LOGS | log\_id | INT IDENTITY(1,1) |
| ROLES | role\_id | INT IDENTITY(1,1) |

* Khóa ngoại (Foreign Key)

| **Bảng** | **Cột** | **Tham chiếu đến** |
| --- | --- | --- |
| EMPLOYEES | department\_id | DEPARTMENTS(department\_id) |
| EMPLOYEES | role\_id | ROLES(role\_id) |
| ROOMS | type\_id | ROOM\_TYPES(type\_id) |
| SERVICES | category\_id | SERVICE\_CATEGORIES(category\_id) |
| RESERVATIONS | customer\_id | CUSTOMERS(customer\_id) |
| RESERVATIONS | room\_id | ROOMS(room\_id) |
| RESERVATIONS | created\_by | EMPLOYEES(employee\_id) |
| PAYMENTS | reservation\_id | RESERVATIONS(reservation\_id) |
| PAYMENTS | customer\_id | CUSTOMERS(customer\_id) |
| PAYMENTS | processed\_by | EMPLOYEES(employee\_id) |
| SERVICES\_USED | reservation\_id | RESERVATIONS(reservation\_id) |
| SERVICES\_USED | service\_id | SERVICES(service\_id) |
| SERVICES\_USED | served\_by | EMPLOYEES(employee\_id) |
| MAINTENANCE\_REQUESTS | room\_id | ROOMS(room\_id) |
| MAINTENANCE\_REQUESTS | assigned\_to | EMPLOYEES(employee\_id) |
| MAINTENANCE\_REQUESTS | created\_by | EMPLOYEES(employee\_id) |
| EMPLOYEE\_SHIFTS | employee\_id | EMPLOYEES(employee\_id) |
| ROOM\_STATUS\_HISTORY | changed\_by | EMPLOYEES(employee\_id) |

* Check (Kiểm tra miền giá trị)

| **Bảng** | **Ràng buộc** | **Mô tả** |
| --- | --- | --- |
| ROOM\_TYPES | base\_price > 0 | Giá phòng phải lớn hơn 0 |
| ROOM\_TYPES | capacity > 0 | Sức chứa phải lớn hơn 0 |
| CUSTOMERS | loyalty\_points >= 0 | Điểm tích lũy không âm |
| CUSTOMERS | membership\_tier IN ('Bronze', 'Silver', 'Gold', 'Platinum') | Hạng thành viên hợp lệ |
| ROOMS | status IN ('Available', 'Occupied', 'Maintenance', 'Cleaning', 'Reserved') | Trạng thái phòng hợp lệ |
| RESERVATIONS | num\_guests > 0 | Số khách phải lớn hơn 0 |
| RESERVATIONS | check\_out\_date > check\_in\_date | Ngày trả phải sau ngày nhận |
| RESERVATIONS | status IN ('Confirmed', 'CheckedIn', 'CheckedOut', 'Cancelled', 'NoShow') | Trạng thái đặt phòng hợp lệ |
| PAYMENTS | amount > 0 | Số tiền phải lớn hơn 0 |
| MAINTENANCE\_REQUESTS | priority IN ('Low', 'Medium', 'High', 'Critical') | Độ ưu tiên hợp lệ |

* Unique (Duy nhất)

| **Bảng** | **Cột** | **Mô tả** |
| --- | --- | --- |
| DEPARTMENTS | department\_name | Tên phòng ban không trùng |
| SERVICE\_CATEGORIES | category\_name | Tên danh mục dịch vụ không trùng |
| ROOM\_TYPES | type\_name | Loại phòng không trùng |
| CUSTOMERS | email | Email khách hàng không trùng |
| EMPLOYEES | email | Email nhân viên không trùng |
| ROOMS | room\_number | Số phòng không trùng |
| ROLES | role\_name | Tên vai trò không trùng |

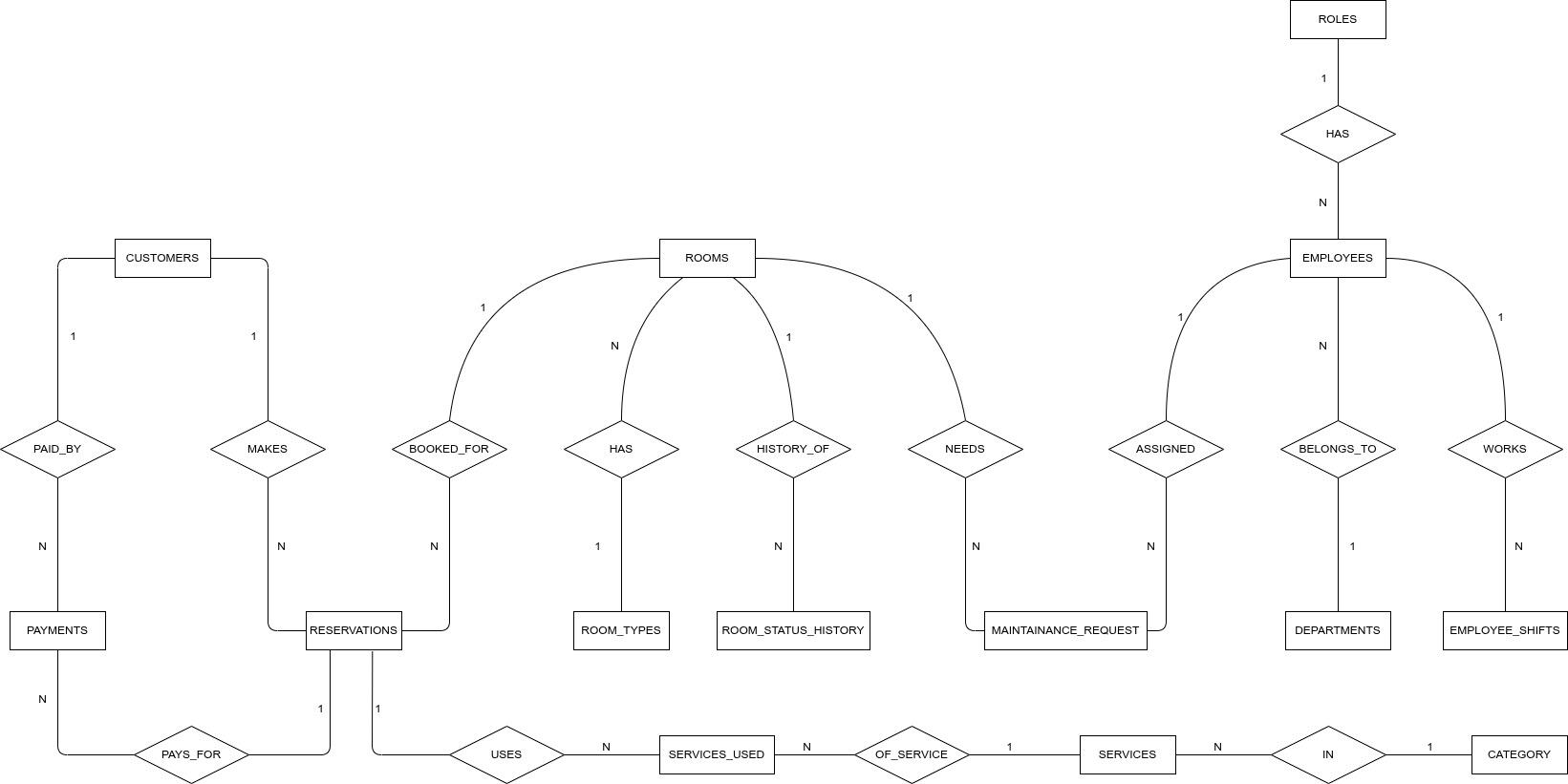
### 

### 

### 2.3. Các công thức tính toán

| **Công thức** | **Mô tả** |
| --- | --- |
| room\_charge = base\_price × seasonal × weekend | Tính tiền phòng |
| service\_charge = unit\_price × quantity | Tính tiền dịch vụ |
| discount\_rate = tier + loyalty\_points + high\_value\_bonus | Giảm giá theo hạng |
| tax\_amount = (room\_charge + service\_charge - discount) × 0.10 | Thuế 10% |
| total\_amount = room\_charge + service\_charge - discount + tax | Tổng hóa đơn |
| loyalty\_points = (amount / 10) × tier\_multiplier | Điểm tích lũy |
| deposit = room\_charge × 30% | Tính tiền đặt cọc |
| customer\_tier = f(total\_spending) | Hạng thành viên |

## 3. Mô hình Thực thể - Liên kết (ER Model)



## 4. Mô hình Quan hệ (Relational Model)

Bảng tóm tắt các quan hệ (Relational Schema Summary)

| **Tên bảng** | **Mô tả** | **Khóa chính** | **Khóa ngoại** |
| --- | --- | --- | --- |
| DEPARTMENTS | Phòng ban | department\_id | - |
| SERVICE\_CATEGORIES | Danh mục dịch vụ | category\_id | - |
| ROOM\_TYPES | Loại phòng | type\_id | - |
| ROLES | Vai trò phân quyền | role\_id | - |
| CUSTOMERS | Khách hàng | customer\_id | - |
| EMPLOYEES | Nhân viên | employee\_id | department\_id, role\_id |
| ROOMS | Phòng | room\_id | type\_id |
| SERVICES | Dịch vụ | service\_id | category\_id |
| RESERVATIONS | Đặt phòng | reservation\_id | customer\_id, room\_id |
| PAYMENTS | Thanh toán | payment\_id | reservation\_id, customer\_id |
| SERVICES\_USED | Dịch vụ sử dụng | usage\_id | reservation\_id, service\_id |
| MAINTENANCE\_REQUESTS | Yêu cầu bảo trì | request\_id | room\_id, assigned\_to |
| EMPLOYEE\_SHIFTS | Ca làm việc | shift\_id | employee\_id |
| ROOM\_STATUS\_HISTORY | Lịch sử trạng thái phòng | history\_id | room\_id |
| AUDIT\_LOGS | Nhật ký hệ thống | log\_id | - |

****

**Mô hình quan hệ**

# PHẦN II. QUẢN TRỊ CƠ SỞ DỮ LIỆU

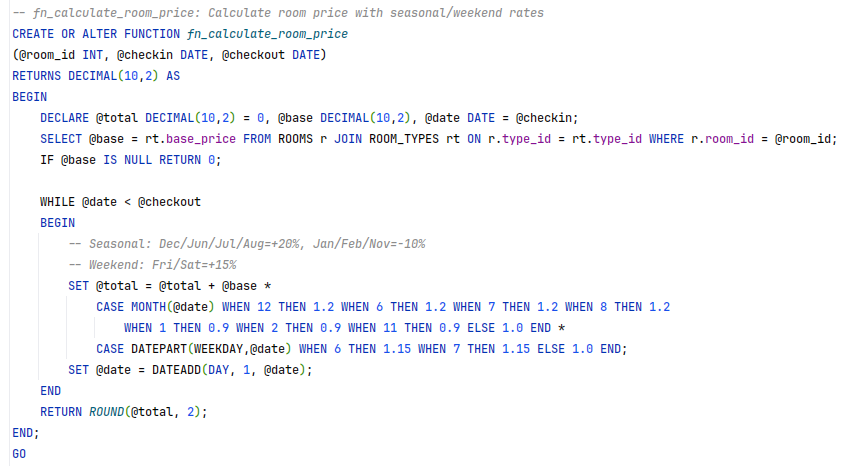
# 1. Thành viên: Nguyễn Hồng Phúc

*Nhiệm vụ: QUẢN LÝ ĐẶT PHÒNG*

### 1.1. Function

#### a. fn\_calculate\_room\_price

* **Mục đích:** Tính giá phòng theo mùa và cuối tuần
* **Tham số:** @room\_id INT, @checkin DATE, @checkout DATE
* **Trả về:** DECIMAL(10,2) - Tổng tiền phòng
* **Được sử dụng bởi:** PROCEDURE sp\_create\_reservation
* **Logic tính giá:**
  + Mùa cao điểm (tháng 6, 7, 8, 12): +20%
  + Mùa thấp điểm (tháng 1, 2, 11): -10%
  + Cuối tuần (Thứ 6, Thứ 7): +15%



#### 

#### 

#### 

#### *b. fn\_check\_room\_availability*

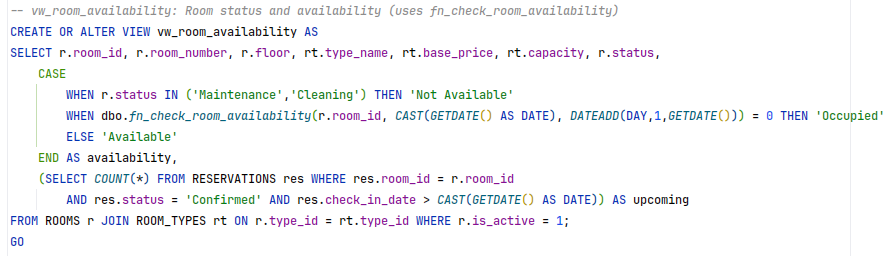
* **Mục đích:** Kiểm tra phòng có sẵn trong khoảng thời gian
* **Tham số:** @room\_id INT, @checkin DATE, @checkout DATE
* **Trả về:** BIT (1 = Sẵn sàng, 0 = Không sẵn)
* **Được sử dụng bởi:** PROCEDURE sp\_create\_reservation
* **Logic kiểm tra:**
  + Phòng phải active và không đang bảo trì
  + Không có đặt phòng trùng thời gian (Confirmed/CheckedIn)



### 1.2. View

#### *a. vw\_room\_availability*

* **Mục đích:** Hiển thị tình trạng phòng real-time
* **Các cột:** room\_id, room\_number, floor, type\_name, base\_price, capacity, status, availability, upcoming
* **Logic:** Tính availability dựa trên status hiện tại và đặt phòng đang hoạt động



#### 

#### *b. vw\_occupancy\_statistics*

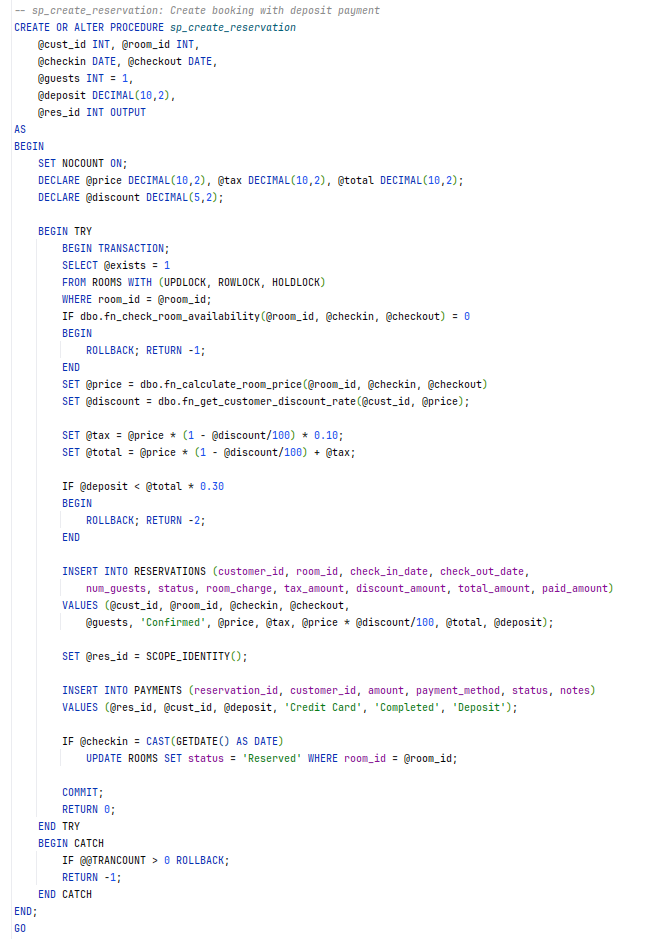
* **Mục đích:** Thống kê tỷ lệ lấp đầy phòng cho quản lý
* **Các cột:** report\_date, total\_rooms, occupied, occupancy\_pct, total\_revenue
* **Ứng dụng:** Dashboard quản lý, báo cáo công suất khách sạn



### 1.3. Procedure + Transaction

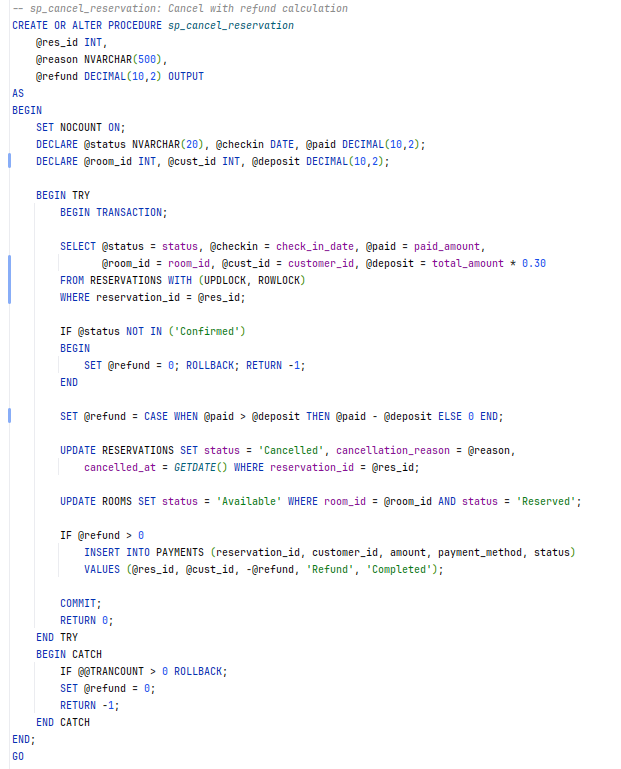
#### *a. sp\_create\_reservation*

* **Mục đích:** Tạo đặt phòng với tính giá tự động
* **Tham số:** @cust\_id, @room\_id, @checkin, @checkout, @guests, @res\_id OUTPUT
* **Transaction:** CÓ (BEGIN TRANSACTION, COMMIT, ROLLBACK)
* **Gọi Function:** fn\_check\_room\_availability, fn\_calculate\_room\_price
* **Logic xử lý:**
  + Kiểm tra phòng còn trống
  + Tính giá phòng theo ngày, giảm giá theo hạng, thuế 10%
  + Tính tiền cọc phải đặt = 30% giá phòng
  + Kiểm tra có đủ tiền cọc không
  + Tạo bản ghi RESERVATIONS
  + Cập nhật trạng thái phòng nếu check-in là hôm nay



#### *b. sp\_cancel\_reservation*

* **Mục đích:** Hủy đặt phòng với chính sách hoàn tiền
* **Tham số:** @res\_id, @reason, @refund OUTPUT
* **Transaction:** CÓ
* **Chính sách hoàn tiền:** Giữ lại tiền cọc, hoàn tiền còn lại

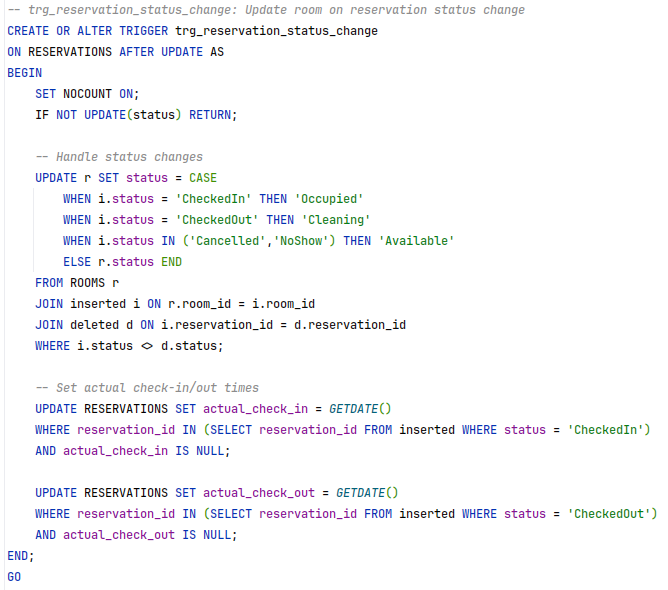


### 

### 1.4. Trigger

#### *a. trg\_reservation\_status\_change*

* **Bảng áp dụng:** RESERVATIONS
* **Loại trigger:** AFTER UPDATE
* **Mục đích:** Tự động cập nhật trạng thái phòng khi đặt phòng thay đổi khi có thay đổi đặt phòng
* **Logic:**
  + CheckedIn → Phòng = Occupied
  + CheckedOut → Phòng = Cleaning
  + Cancelled/NoShow → Phòng = Available
  + Tự động ghi nhận actual\_check\_in, actual\_check\_out



#### 

#### *b. trg\_reservation\_audit*

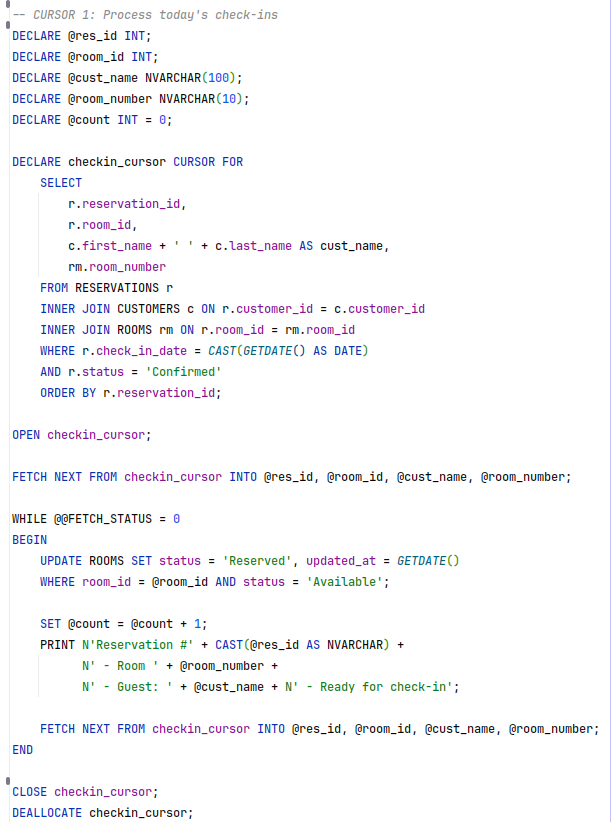
* **Bảng áp dụng:** RESERVATIONS
* **Loại trigger:** AFTER INSERT, UPDATE, DELETE
* **Mục đích:** Ghi log đặt phòng vào bảng AUDIT\_LOGS
* **Thông tin ghi:** Bảng, thao tác, mã bản ghi, giá trị cũ/mới, người thực hiện



### 1.5. Cursor

#### checkin\_cursor

* **Mục đích:** Xử lý các check-in trong ngày
* **OUTPUT:** @count INT - Số đặt phòng được xử lý
* **Logic Cursor:** Duyệt qua tất cả đặt phòng có check\_in\_date = hôm nay và status = 'Confirmed', cập nhật phòng thành 'Reserved'



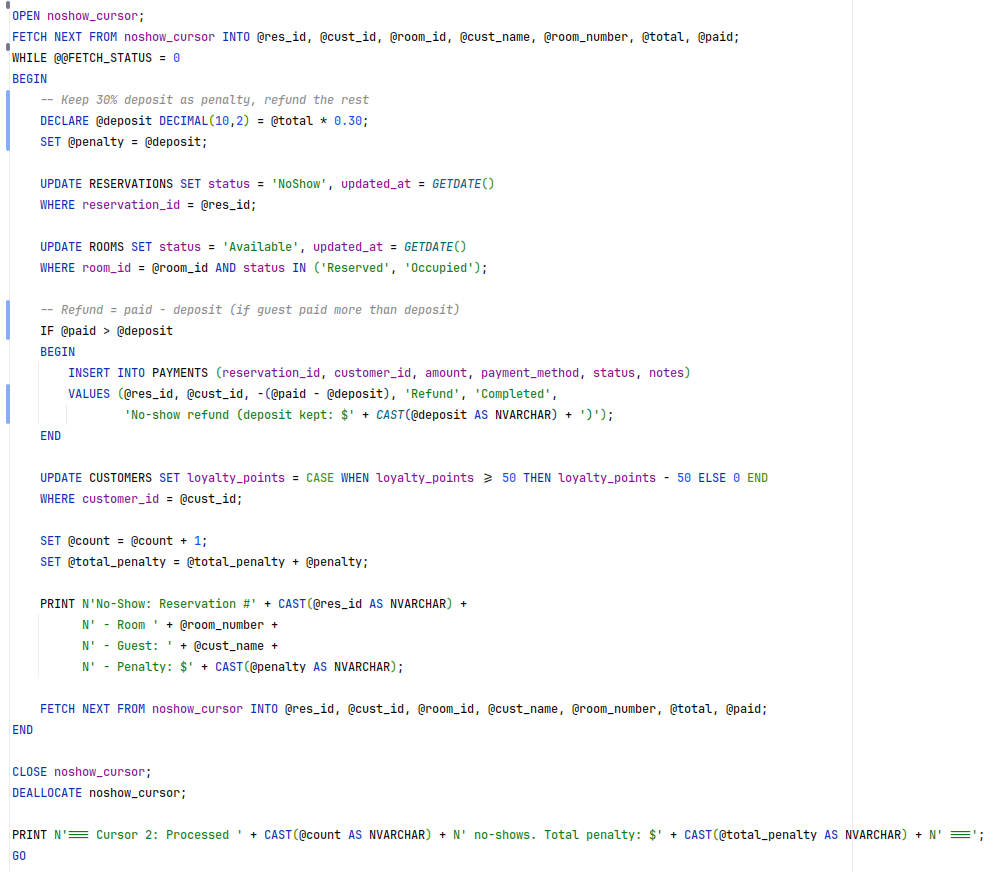
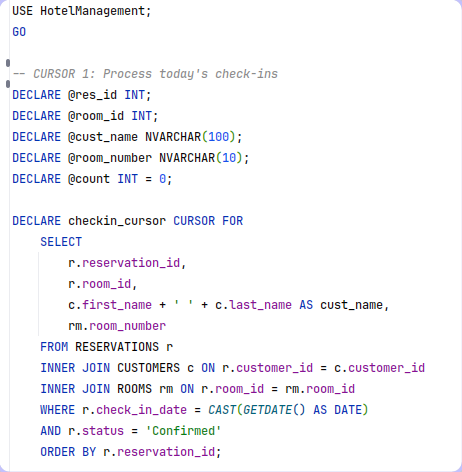
#### 

#### 

#### 

#### *b.* noshow\_cursor

* **Mục đích:** Xử lý các đặt phòng quá hạn check in
* **OUTPUT:** @count INT, @penalty DECIMAL(10,2)
* **Logic Cursor:**
  + Tìm đặt phòng quá hạn check-in chưa nhận phòng
  + Đánh dấu NoShow, giữ lại tiền cọc
  + Giải phóng phòng về Available
  + Trừ 50 điểm loyalty của khách



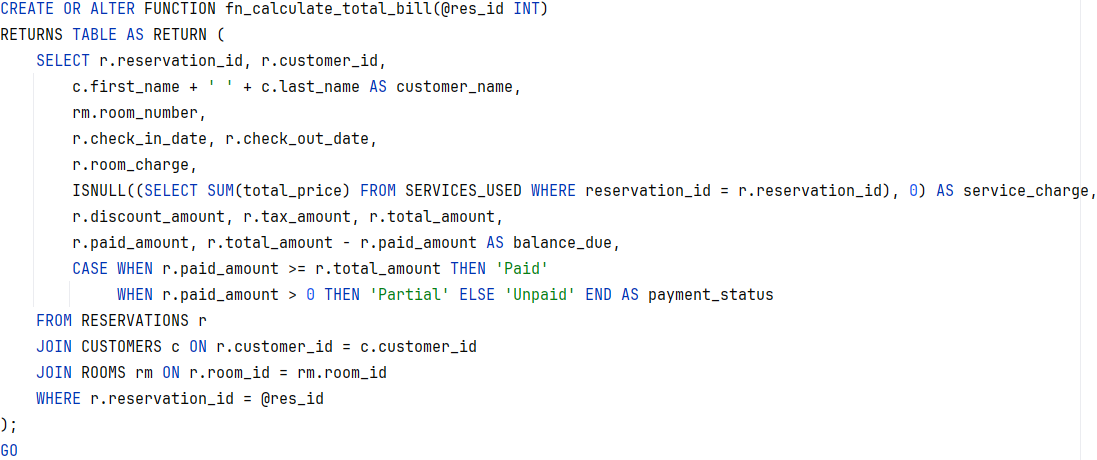
## 2. Thành viên: Ngô Đức Nam Khánh

*Nhiệm vụ: QUẢN LÝ THANH TOÁN & TÀI CHÍNH*

### 2.1. Function

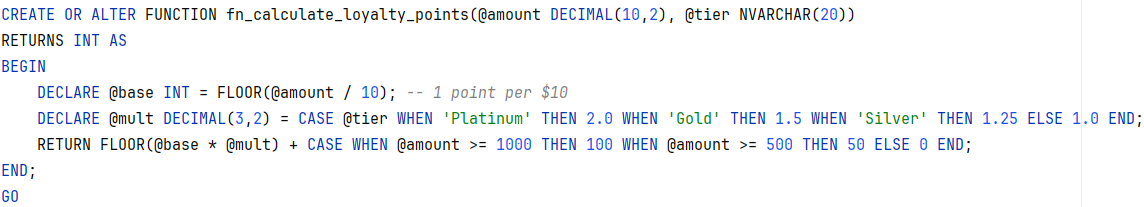
#### *a. fn\_calculate\_total\_bill*

* **Mục đích:** Tạo bảng hóa đơn chi tiết cho đặt phòng
* **Tham số:** @res\_id INT
* **Trả về:** TABLE (reservation\_id, customer\_name, room\_charge, service\_charge, check\_in\_date, check\_out\_date, discount\_amount, tax\_amount, total\_amount, paid\_amount, balance\_due, payment\_status)
* **Được sử dụng bởi:** PROCEDURE sp\_generate\_invoice



#### *b. fn\_calculate\_loyalty\_points*

* **Mục đích:** Tính điểm thưởng dựa trên số tiền và hạng thành viên
* **Tham số:** @amount DECIMAL(10,2), @tier NVARCHAR(20)
* **Trả về:** INT - Số điểm được cộng
* **Được sử dụng bởi:** PROCEDURE sp\_process\_payment
* **Logic tính điểm:**
  + Cơ bản: 1 điểm / $10
  + Hệ số nhân: Platinum 2x, Gold 1.5x, Silver 1.25x, Bronze 1x
  + Bonus: ≥$1,000 +100 điểm, ≥$500 +50 điểm



#### 

#### 

#### 

#### 

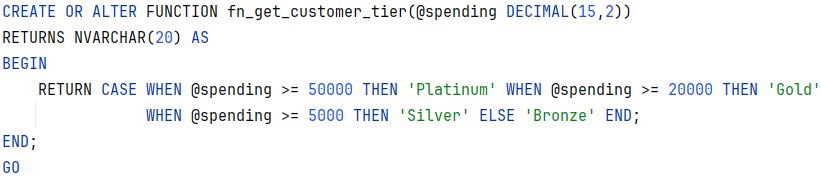
#### 

#### 

#### 

#### *c. fn\_get\_customer\_tier*

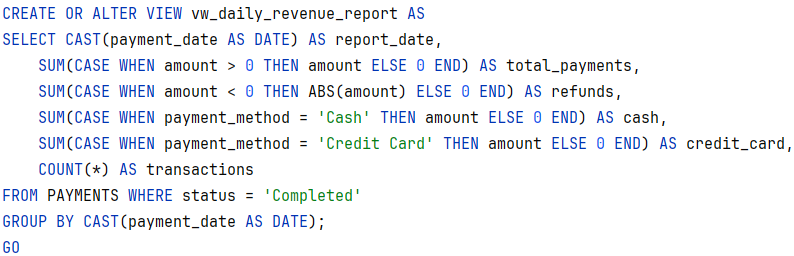
* **Mục đích:** Xác định hạng thành viên từ tổng chi tiêu
* **Tham số:** @spending DECIMAL(15,2)
* **Trả về:** NVARCHAR(20) - Bronze/Silver/Gold/Platinum
* **Được sử dụng bởi:** TRIGGER trg\_payment\_loyalty\_update
* **Ngưỡng:** ≥$50k: Platinum, ≥$20k: Gold, ≥$5k: Silver



### 2.2. View

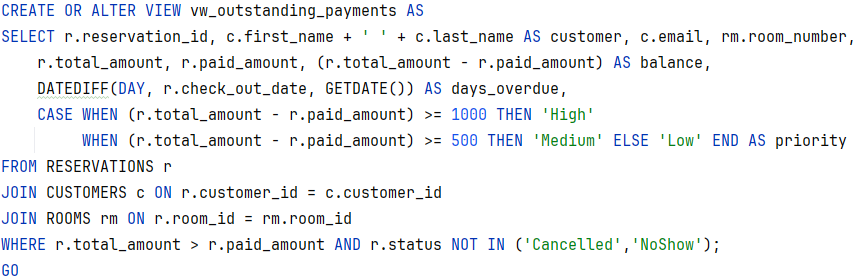
#### *a. vw\_daily\_revenue\_report*

* **Mục đích:** Báo cáo doanh thu hàng ngày cho kế toán
* **Các cột:** report\_date, total\_payments, refunds, cash, credit\_card, transactions
* **Ứng dụng:** Đối chiếu cuối ngày, báo cáo tài chính



#### *b. vw\_outstanding\_payments*

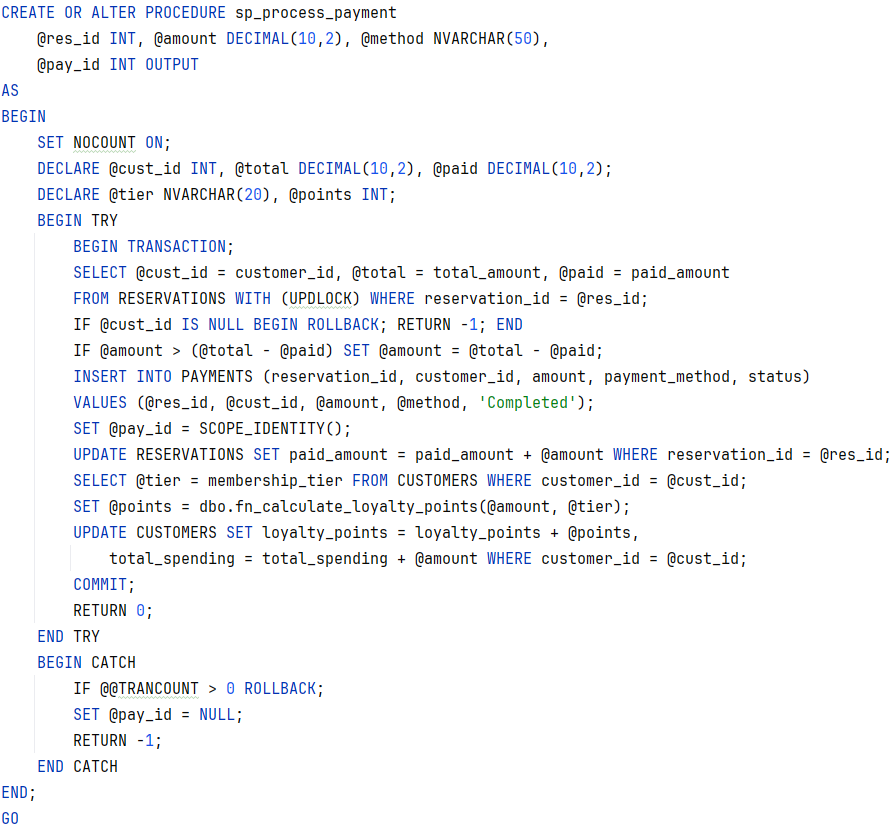
* **Mục đích:** Theo dõi công nợ chưa thanh toán
* **Các cột:** reservation\_id, customer, email, room\_number, total\_amount, paid\_amount, balance, days\_overdue, priority
* **Logic phân loại:** High (≥$1,000), Medium (≥$500), Low (còn lại)



### 2.3. Procedure + Transaction

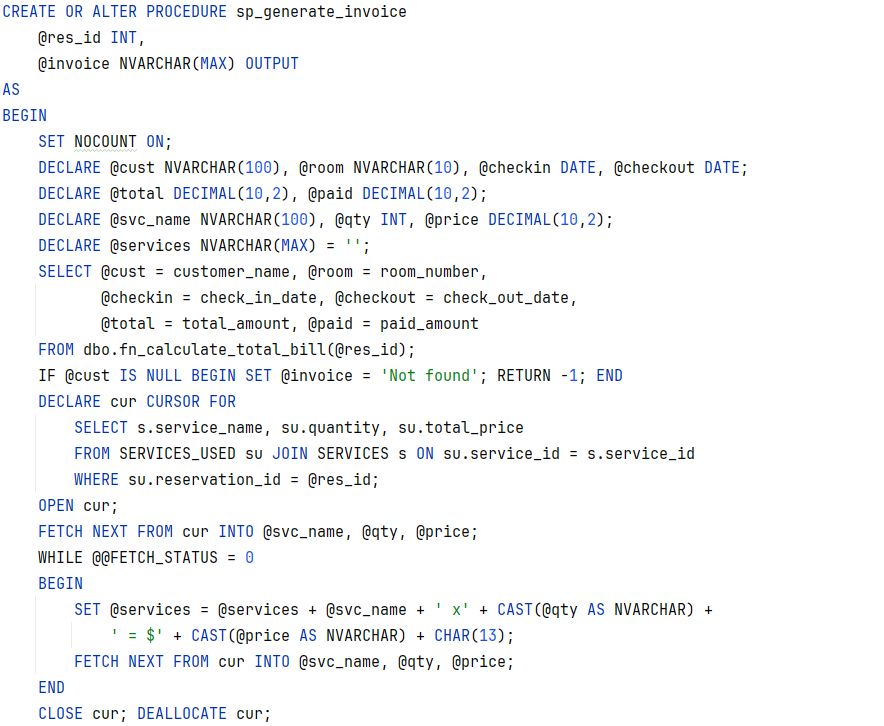
#### *a. sp\_process\_payment*

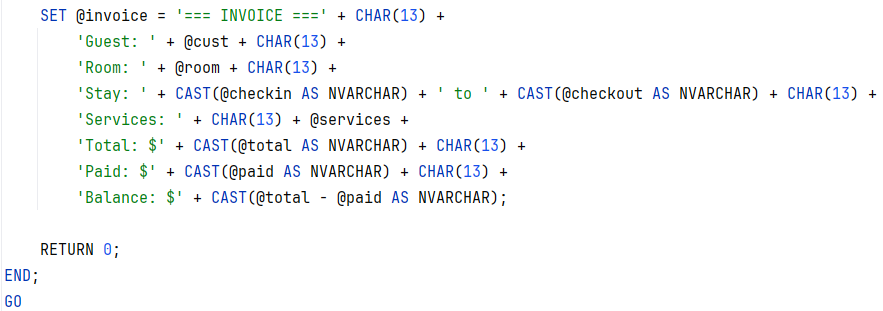
* **Mục đích:** Xử lý thanh toán và cộng điểm loyalty
* **Tham số:** @res\_id INT, @amount DECIMAL(10,2), @method NVARCHAR(50), @pay\_id INT OUTPUT
* **Transaction:** CÓ
* **Gọi Function:** fn\_calculate\_loyalty\_points
* **Logic xử lý:**
  + Giới hạn thanh toán không vượt quá số còn nợ
  + Tạo bản ghi PAYMENTS
  + Cập nhật paid\_amount của đặt phòng
  + Cộng điểm loyalty và total\_spending cho khách



#### *b. sp\_generate\_invoice*

* **Mục đích:** Tạo hóa đơn chi tiết với cursor duyệt dịch vụ
* **Tham số:** @res\_id INT, @invoice NVARCHAR(MAX) OUTPUT
* **Sử dụng Cursor:** Duyệt qua SERVICES\_USED để liệt kê từng dịch vụ
* **Gọi Function:** fn\_calculate\_total\_bill
* **Kết quả:** Chuỗi hóa đơn chi tiết với thông tin khách, phòng, dịch vụ, tổng tiền

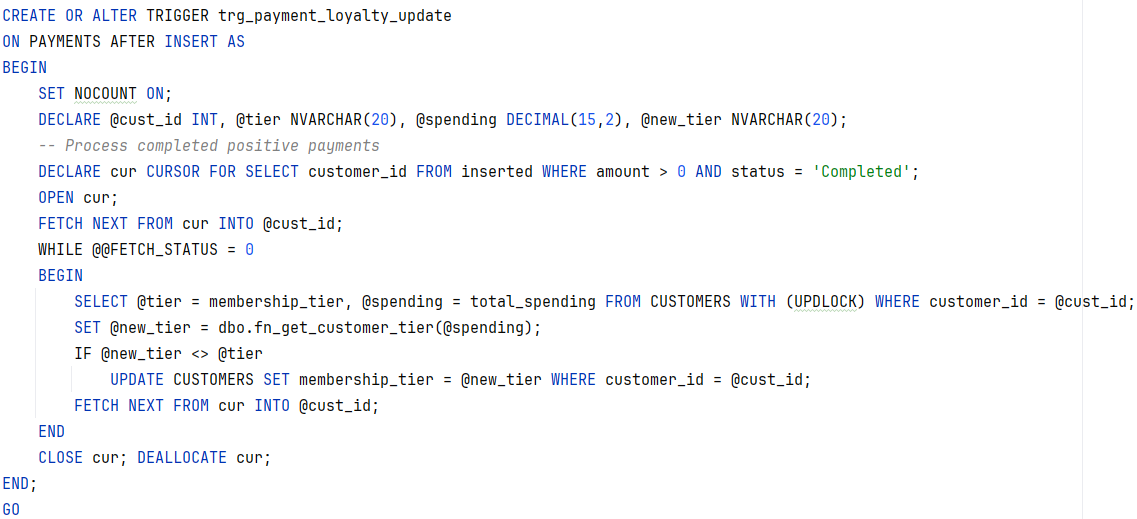




### 2.4. Trigger

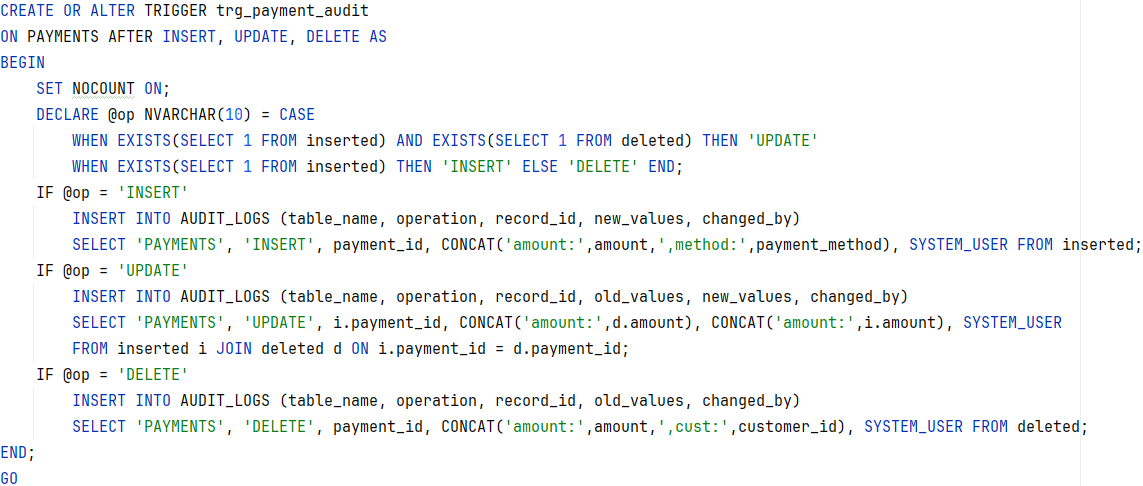
#### *a. trg\_payment\_loyalty\_update*

* **Bảng áp dụng:** PAYMENTS
* **Loại trigger:** AFTER INSERT
* **Mục đích:** Tự động nâng hạng thành viên sau thanh toán
* **Gọi Function:** fn\_get\_customer\_tier
* **Logic:** Kiểm tra total\_spending mới, so sánh với hạng hiện tại, nâng hạng nếu đủ điều kiện



#### *b. trg\_payment\_audit*

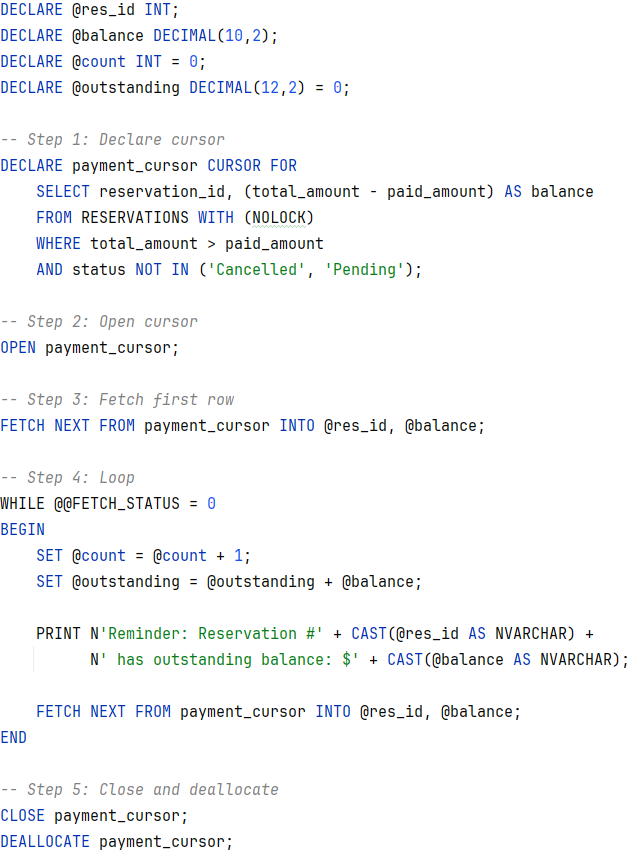
* **Bảng áp dụng:** PAYMENTS
* **Loại trigger:** AFTER INSERT, UPDATE, DELETE
* **Mục đích:** Ghi log tất cả giao dịch thanh toán vào AUDIT\_LOGS



### 2.5. Cursor

#### *a.* Nhắc nhở thanh toán

* **Mục đích:** Thống kê công nợ cần nhắc nhở thanh toán
* **OUTPUT:** @count INT, @outstanding DECIMAL(12,2)
* **Logic Cursor:** Duyệt qua tất cả đặt phòng còn nợ, tính tổng số tiền và số lượng



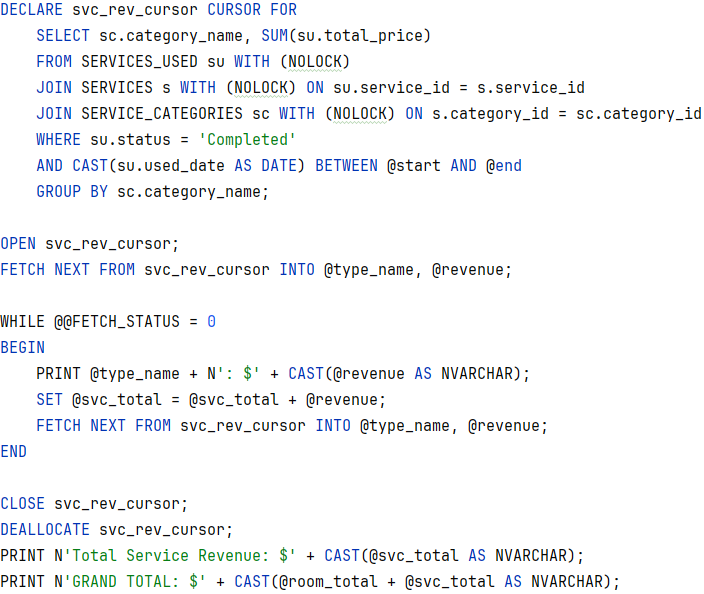
#### 

#### 

#### 

#### *b.* Tạo báo cáo doanh thu tháng

* **Mục đích:** Tạo báo cáo doanh thu tháng chi tiết
* **Tham số:** @year INT, @month INT, @output NVARCHAR(MAX) OUTPUT
* **Sử dụng 2 Cursor:**
  + Cursor 1: Doanh thu phòng theo loại phòng
  + Cursor 2: Doanh thu dịch vụ theo danh mục



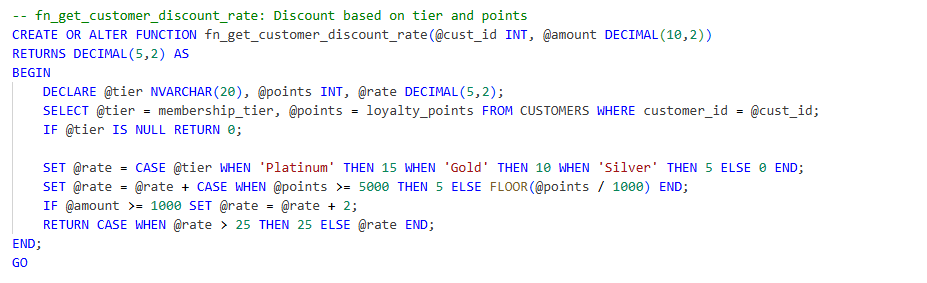
## 3. Thành viên: Nguyễn Hải Ninh

*Nhiệm vụ: QUẢN LÝ KHÁCH HÀNG & DỊCH VỤ*

### 3.1. Function

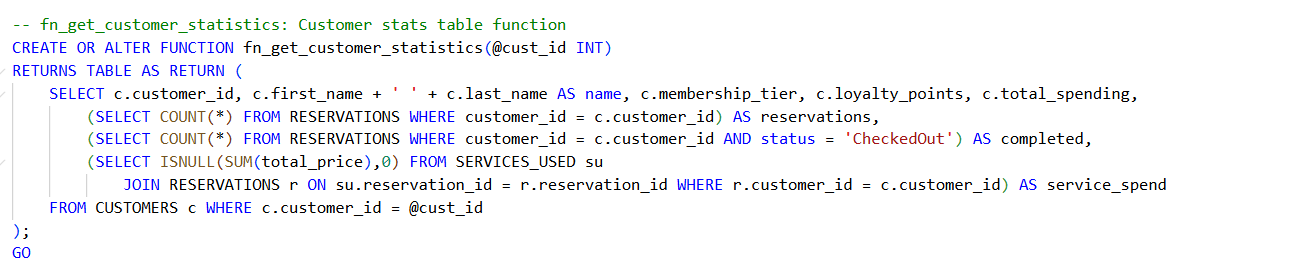
#### *a. fn\_get\_customer\_discount\_rate*

* **Mục đích:** Tính tỷ lệ giảm giá tổng hợp cho khách hàng
* **Tham số:** @cust\_id INT, @amount DECIMAL(10,2)
* **Trả về:** DECIMAL(5,2) - Phần trăm giảm giá (tối đa 25%)
* **Được sử dụng bởi:** PROCEDURE sp\_add\_service\_to\_reservation
* **Logic tính:**
  + Giảm giá theo hạng: Platinum 15%, Gold 10%, Silver 5%
  + Bonus theo điểm: +1% mỗi 1,000 điểm (tối đa +5%)
  + Đơn lớn ≥$1,000: +2%



#### *b. fn\_get\_customer\_statistics*

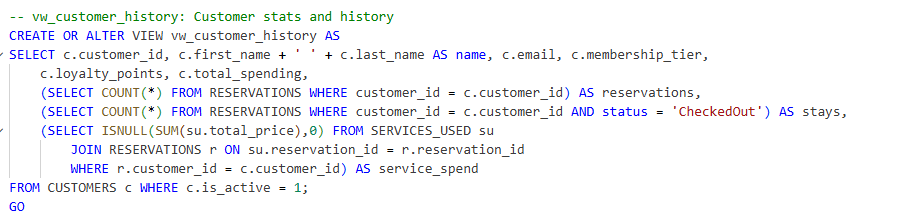
* **Mục đích:** Lấy thống kê toàn diện về khách hàng
* **Tham số:** @cust\_id INT
* **Trả về:** TABLE (customer\_id, name, membership\_tier, loyalty\_points, total\_spending, reservations, completed, service\_spend)
* **Ứng dụng:** Dashboard khách hàng, CRM



### 3.2. View

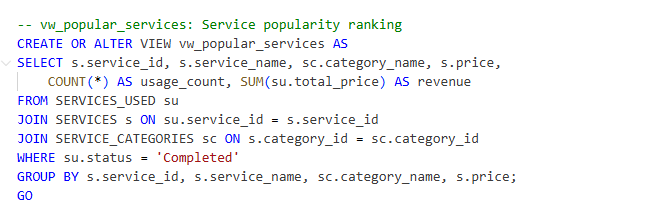
#### *a. vw\_customer\_history*

* **Mục đích:** Hiển thị lịch sử và thống kê khách hàng
* **Các cột:** customer\_id, name, email, membership\_tier, loyalty\_points, total\_spending, reservations, stays, service\_spend
* **Ứng dụng:** Báo cáo khách hàng thân thiết, marketing



#### *b. vw\_popular\_services*

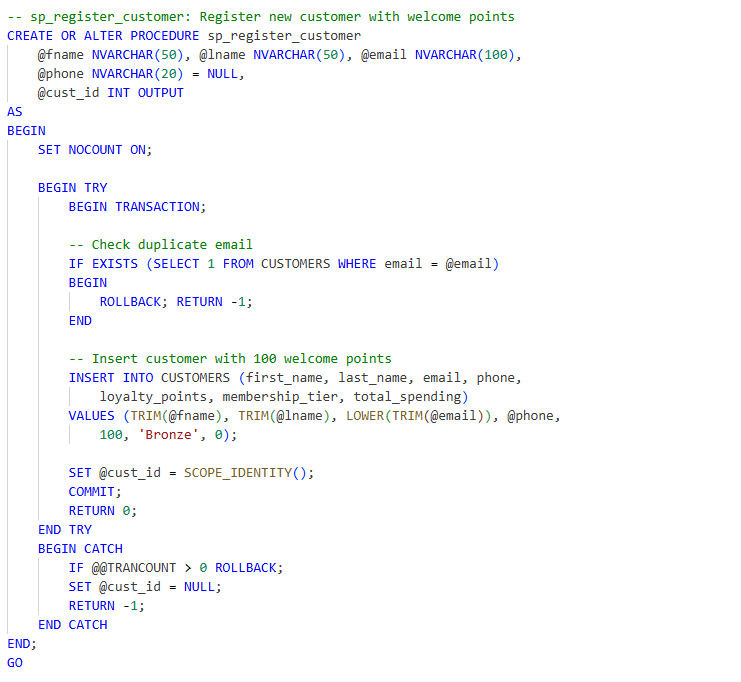
* **Mục đích:** Xếp hạng dịch vụ theo mức độ phổ biến
* **Các cột:** service\_id, service\_name, category\_name, price, usage\_count, revenue
* **Ứng dụng:** Phân tích dịch vụ, quyết định kinh doanh



### 3.3. Procedure + Transaction

#### *a. sp\_register\_customer*

* **Mục đích:** Đăng ký khách hàng mới với điểm chào mừng
* **Tham số:** @fname NVARCHAR(50), @lname NVARCHAR(50), @email NVARCHAR(100), @phone NVARCHAR(20), @cust\_id INT OUTPUT
* **Transaction:** CÓ
* **Logic xử lý:**
  + Kiểm tra email không trùng lặp
  + Chuẩn hóa dữ liệu (trim, lowercase email)
  + Tặng 100 điểm chào mừng
  + Đặt hạng Bronze, tổng chi tiêu = 0



#### *b. sp\_add\_service\_to\_reservation*

* **Mục đích:** Thêm dịch vụ vào đặt phòng và cập nhật hóa đơn
* **Tham số:** @res\_id INT, @svc\_id INT, @qty INT, @usage\_id INT OUTPUT
* **Transaction:** CÓ
* **Logic xử lý:**
  + Kiểm tra đặt phòng đang CheckedIn
  + Lấy giá dịch vụ hiện tại, tạo bản ghi SERVICES\_USED
  + Cập nhật service\_charge, tax\_amount, total\_amount của đặt phòng



### 3.4. Trigger

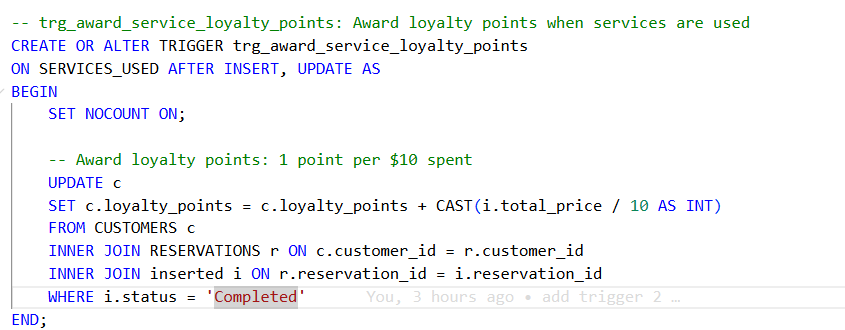
#### *a. trg\_customer\_tier\_upgrade*

* **Bảng áp dụng:** CUSTOMERS
* **Loại trigger:** AFTER UPDATE
* **Mục đích:** Tự động nâng hạng khi tổng chi tiêu tăng
* **Logic đặc biệt:**
  + Chỉ kích hoạt khi total\_spending thay đổi
  + Ngưỡng: $50k Platinum, $20k Gold, $5k Silver



#### b. trg\_award\_service\_loyalty\_points

* **Bảng áp dụng:** SERVICES\_USED
* **Loại trigger:** AFTER INSERT, UPDATE
* **Mục đích:** Tự động tặng điểm khi khách hàng sử dụng dịch vụ
* **Logic đặc biệt:**
  + Chỉ xử lý dịch vụ có status = 'Completed'
  + Tỷ lệ: 1 loyalty point cho mỗi $10 chi tiêu
  + Tránh tặng điểm trùng lặp khi cập nhật



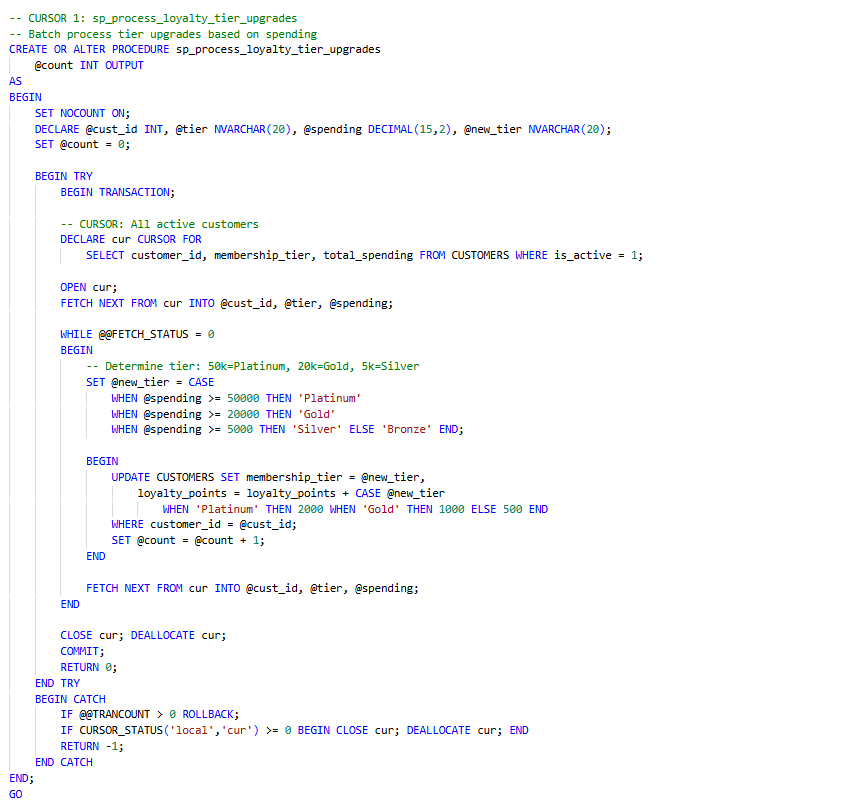
### 

### 

### 3.5. Cursor

#### *a. sp\_process\_loyalty\_tier\_upgrades*

* **Mục đích:** Xử lý hàng loạt nâng hạng với điểm thưởng
* **OUTPUT:** @count INT - Số khách được nâng hạng
* **Transaction:** CÓ
* **Logic Cursor:**
  + Duyệt qua tất cả khách hàng active
  + Kiểm tra và nâng hạng nếu đủ điều kiện
  + Thưởng điểm: Platinum +2,000, Gold +1,000, Silver +500

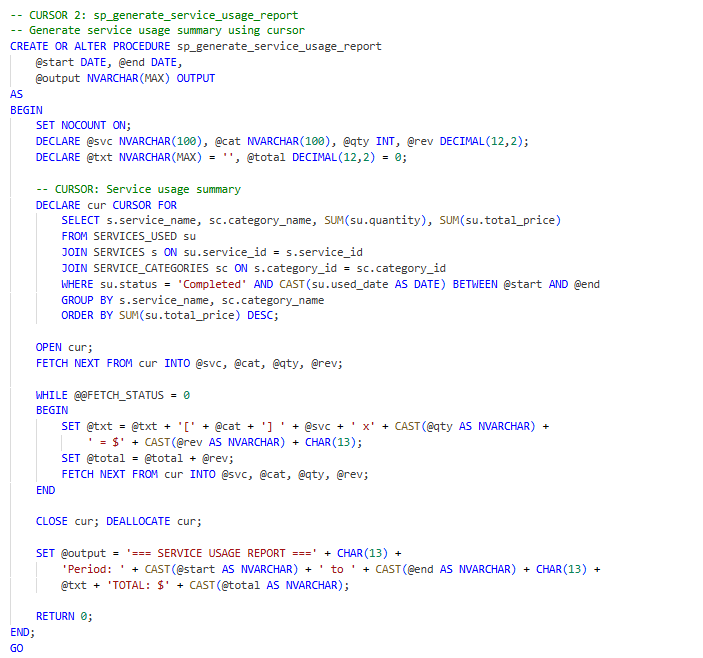


#### 

#### 

#### *b. sp\_generate\_service\_usage\_report*

* **Mục đích:** Tạo báo cáo sử dụng dịch vụ theo thời gian
* **Tham số:** @start DATE, @end DATE, @output NVARCHAR(MAX) OUTPUT
* **Logic Cursor:** Duyệt qua dịch vụ đã sử dụng, nhóm theo tên và danh mục, tính tổng số lượng và doanh thu



## 

## 4. Thành viên: Ngô Quang Tùng

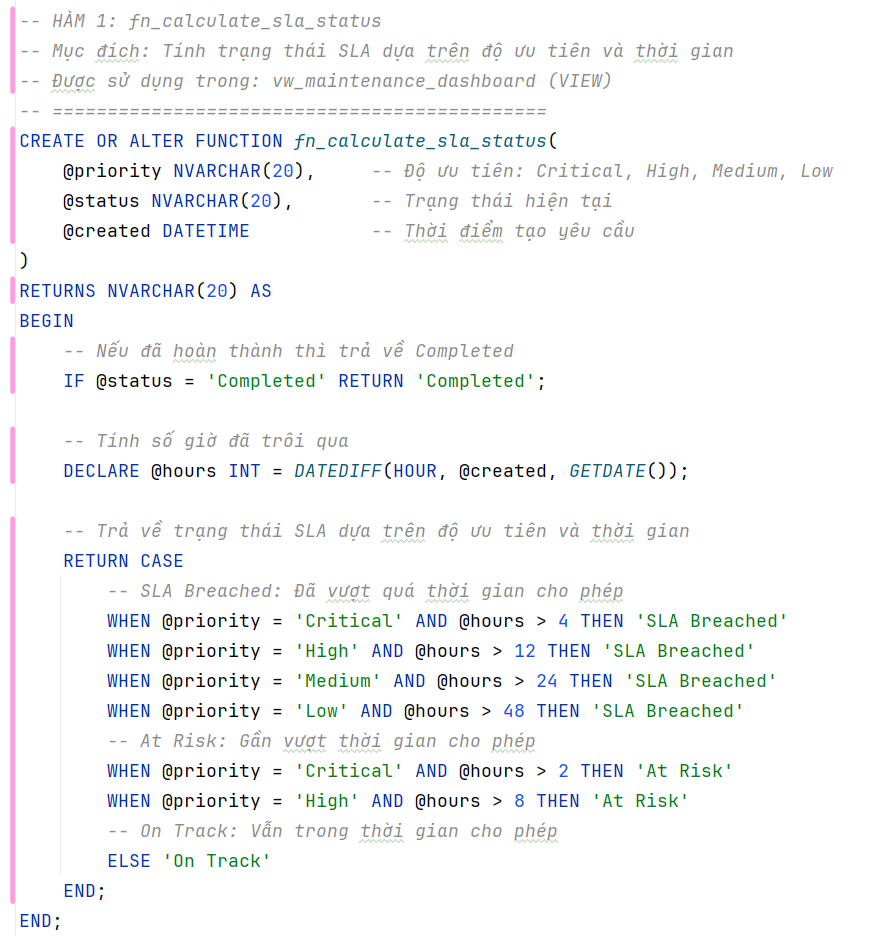
*Nhiệm vụ: QUẢN LÝ VẬN HÀNH & NHÂN SỰ*

### 4.1. Function

#### *a. fn\_calculate\_sla\_status*

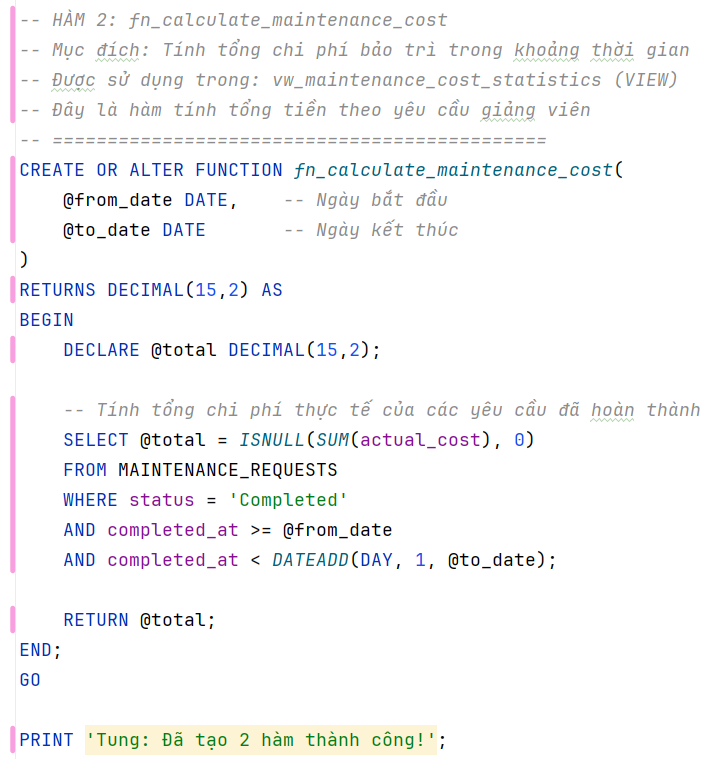
* **Mục đích:** Tính trạng thái SLA (Service Level Agreement) cho yêu cầu bảo trì
* **Tham số:** @priority NVARCHAR(20), @status NVARCHAR(20), @created DATETIME
* **Trả về:** NVARCHAR(20) - On Track, At Risk, SLA Breached, Completed
* **Được gọi bởi:** VIEW vw\_maintenance\_dashboard
* **Logic SLA:**

| **Priority** | **SLA Breached (giờ)** | **At Risk (giờ)** |
| --- | --- | --- |
| Critical | > 4 | > 2 |
| High | > 12 | > 8 |
| Medium | > 24 | - |
| Low | > 48 | - |



#### *b. fn\_calculate\_maintenance\_cost*

* **Mục đích:** Tính tổng chi phí bảo trì trong khoảng thời gian
* **Tham số:** @from\_date DATE, @to\_date DATE
* **Trả về:** DECIMAL(15,2) - Tổng chi phí thực tế
* **Được gọi bởi:** VIEW vw\_maintenance\_cost\_statistics
* **Logic:** Tổng tiền của các yêu cầu Completed trong khoảng thời gian



### 4.2. View

#### *a. vw\_maintenance\_dashboard*

* **Mục đích:** Dashboard tổng quan yêu cầu bảo trì cho quản lý
* **Các cột:** request\_id, room\_number, title, priority, status, assigned\_to, created\_at, hours\_elapsed, sla\_status
* **Gọi Function:** fn\_calculate\_sla\_status
* **Ứng dụng:** Giám sát real-time, cảnh báo SLA



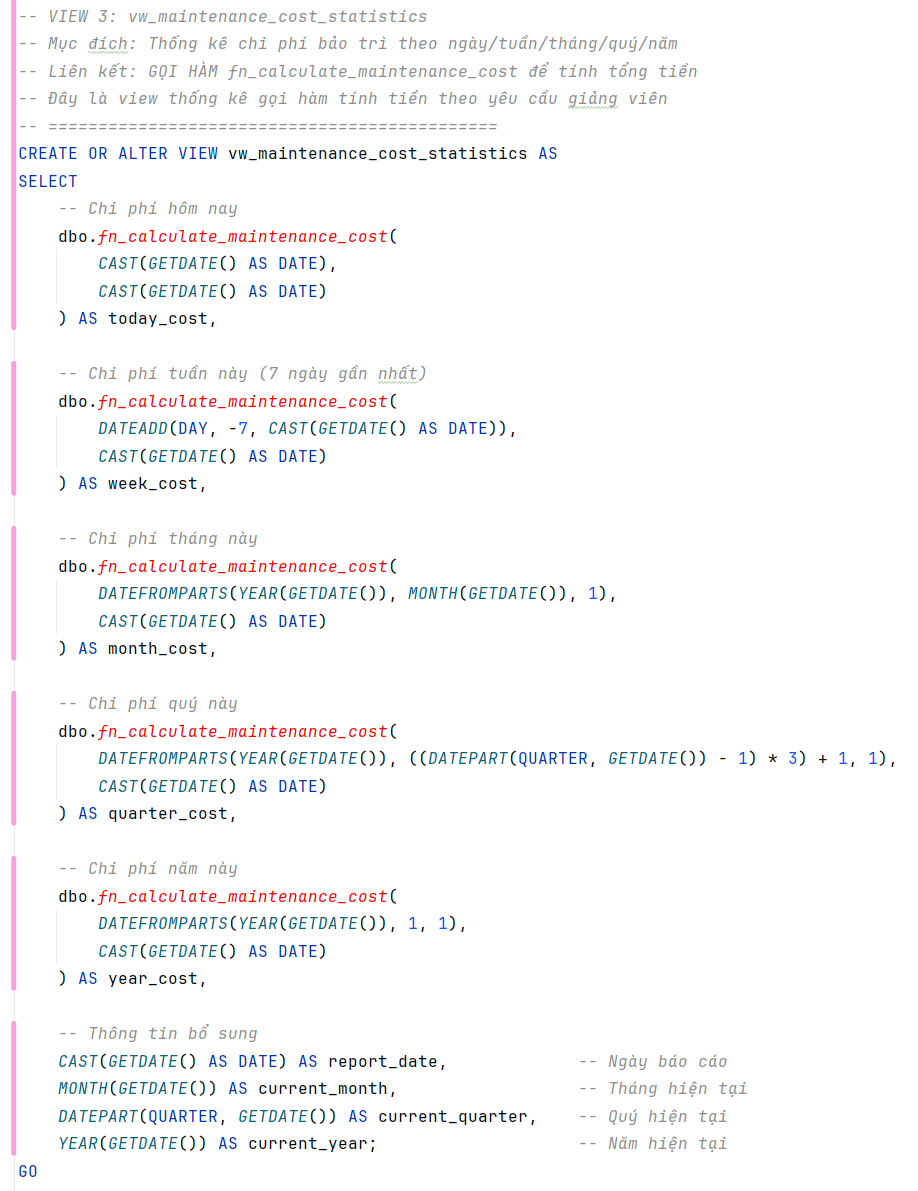
#### *b. vw\_employee\_performance*

* **Mục đích:** Thống kê hiệu suất làm việc của nhân viên
* **Các cột:** employee\_id, name, department\_name, total\_shifts, completed, tasks\_done
* **Ứng dụng:** Đánh giá nhân viên, phân công công việc



#### *c. vw\_maintenance\_cost\_statistics*

* **Mục đích:** Thống kê chi phí bảo trì theo các khoảng thời gian
* **Các cột:** today\_cost, week\_cost, month\_cost, quarter\_cost, year\_cost, report\_date, current\_month, current\_quarter, current\_year
* **Gọi Function:** fn\_calculate\_maintenance\_cost (5 lần với các khoảng thời gian khác nhau)
* **Ứng dụng:** Báo cáo tài chính bảo trì, lập ngân sách



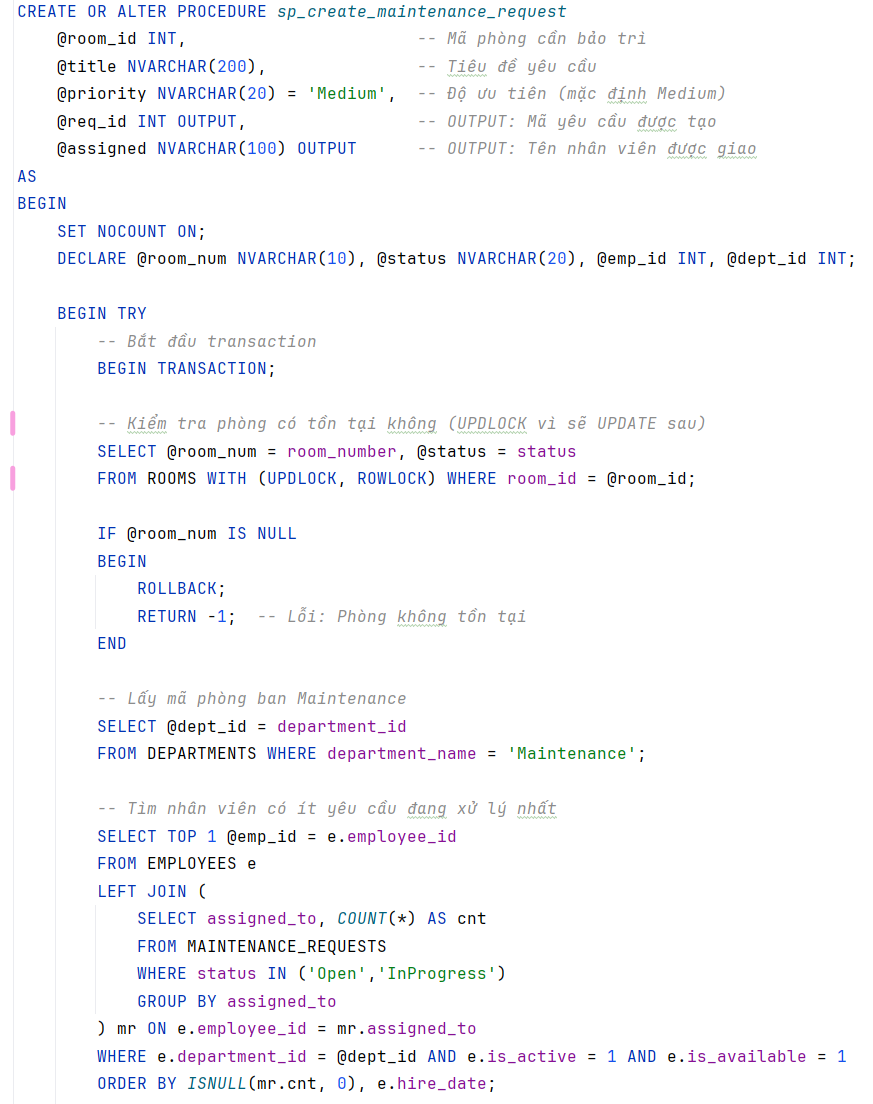
### 

### 

### 4.3. Procedure + Transaction

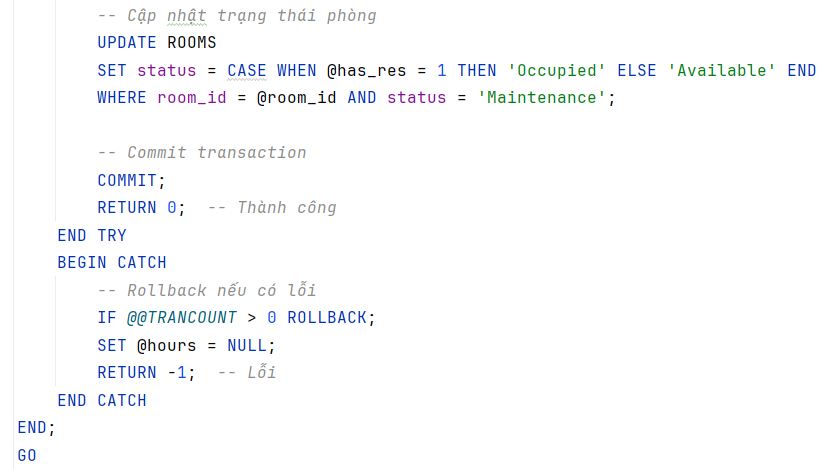
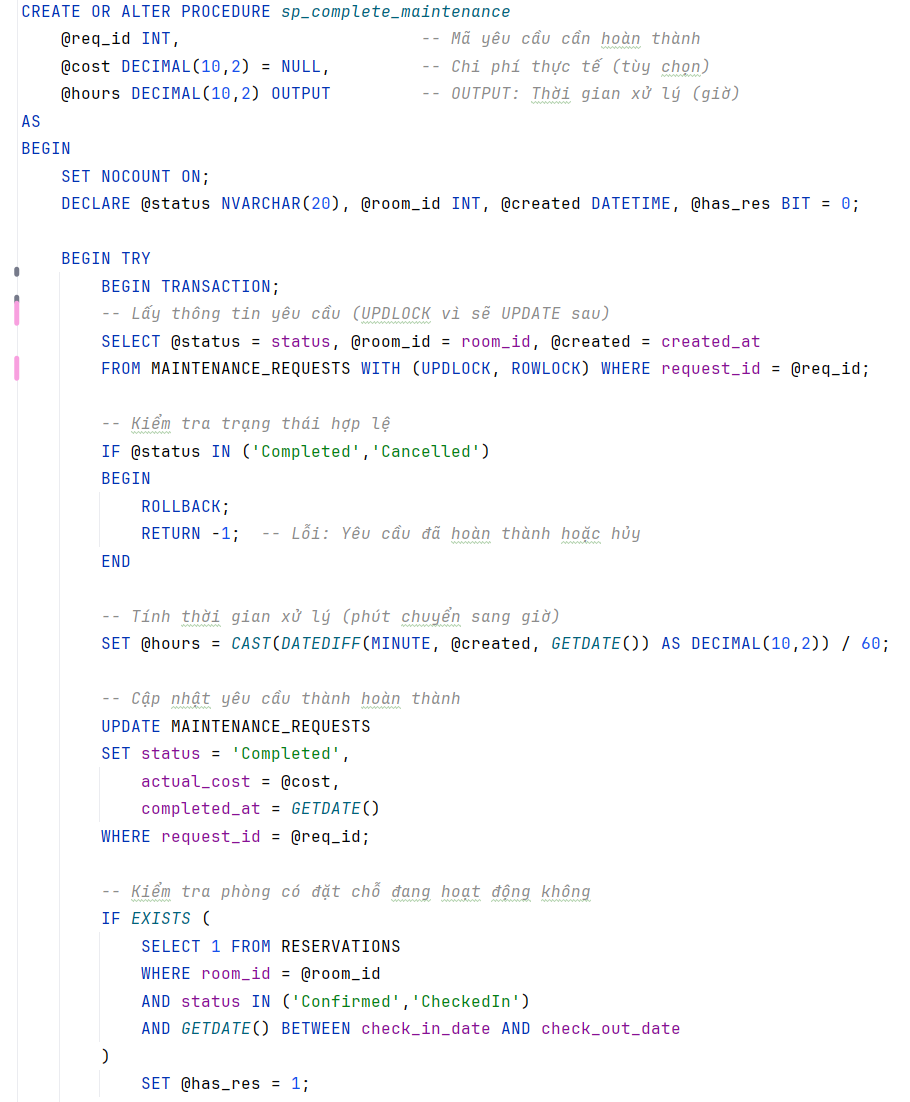
#### *a. sp\_create\_maintenance\_request*

* **Mục đích:** Tạo yêu cầu bảo trì và tự động phân công nhân viên
* **Tham số:**
  + @room\_id INT - Mã phòng cần bảo trì
  + @title NVARCHAR(200) - Tiêu đề yêu cầu
  + @priority NVARCHAR(20) - Độ ưu tiên (mặc định Medium)
  + @req\_id INT OUTPUT - Mã yêu cầu được tạo
  + @assigned NVARCHAR(100) OUTPUT - Tên nhân viên được giao
* **Transaction:** CÓ (BEGIN TRANSACTION, COMMIT, ROLLBACK)
* **Chống Deadlock:** Sử dụng WITH (UPDLOCK, ROWLOCK) khi SELECT bảng ROOMS để khóa row ngay từ đầu, tránh deadlock với các transaction khác.
* **Logic xử lý:**
  + Kiểm tra phòng tồn tại (có UPDLOCK)
  + Tìm nhân viên bảo trì có ít task đang xử lý nhất
  + Tạo yêu cầu bảo trì với trạng thái Open
  + Nếu priority Critical/High: Chuyển phòng sang Maintenance

****

#### *b. sp\_complete\_maintenance*

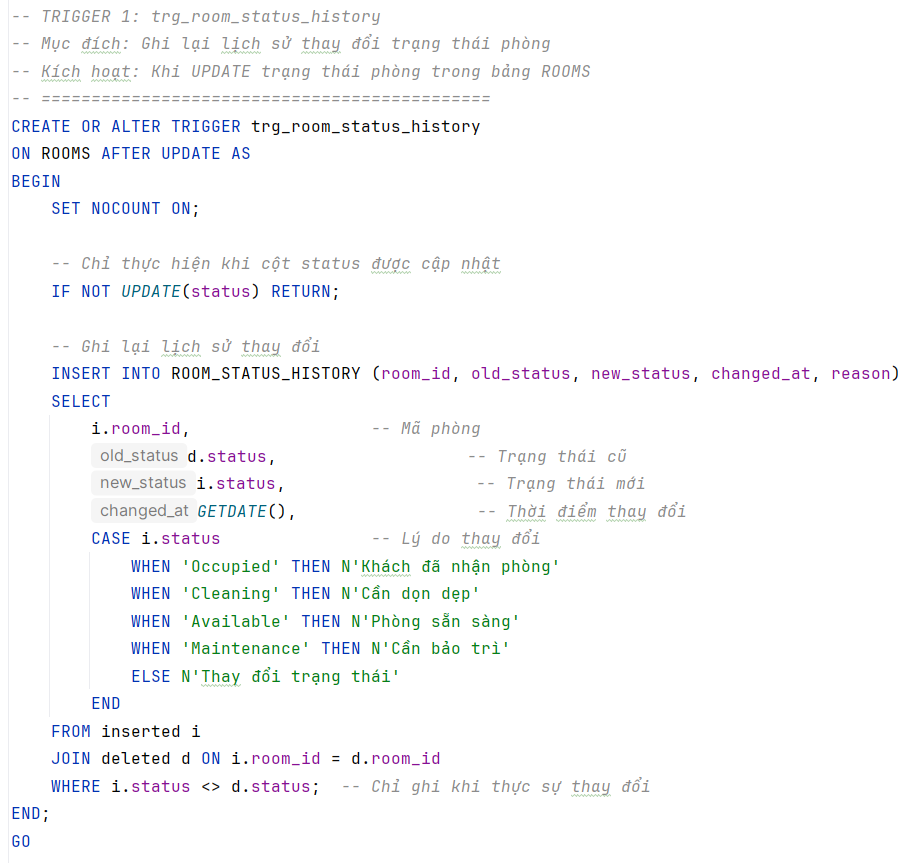
* **Mục đích:** Hoàn thành yêu cầu bảo trì và cập nhật trạng thái phòng
* **Tham số:**
  + @req\_id INT - Mã yêu cầu cần hoàn thành
  + @cost DECIMAL(10,2) - Chi phí thực tế (tùy chọn)
  + @hours DECIMAL(10,2) OUTPUT - Thời gian xử lý (giờ)
* **Transaction:** CÓ
* **Chống Deadlock:** Sử dụng WITH (UPDLOCK, ROWLOCK) khi SELECT bảng MAINTENANCE\_REQUESTS để khóa row ngay từ đầu.
* **Logic xử lý:**
  + Lấy thông tin yêu cầu (có UPDLOCK)
  + Tính thời gian xử lý (giờ) từ lúc tạo đến hoàn thành
  + Cập nhật status = Completed, actual\_cost, completed\_at
  + Khôi phục trạng thái phòng (Occupied nếu có đặt phòng, Available nếu không)



### 4.4. Trigger

#### *a. trg\_room\_status\_history*

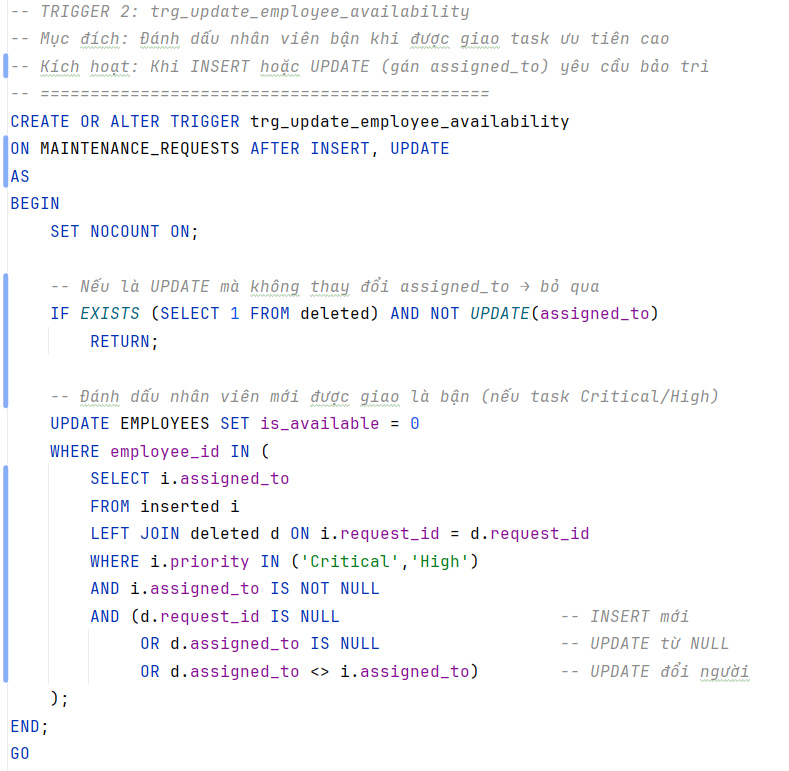
* **Bảng áp dụng:** ROOMS
* **Loại trigger:** AFTER UPDATE
* **Mục đích:** Ghi lại lịch sử thay đổi trạng thái phòng vào ROOM\_STATUS\_HISTORY
* **Thông tin ghi:** room\_id, old\_status, new\_status, changed\_at, reason (tự động xác định theo trạng thái mới)
* **Logic:** Chỉ ghi khi cột status thực sự thay đổi (IF UPDATE(status) và old\_status <> new\_status)



#### 

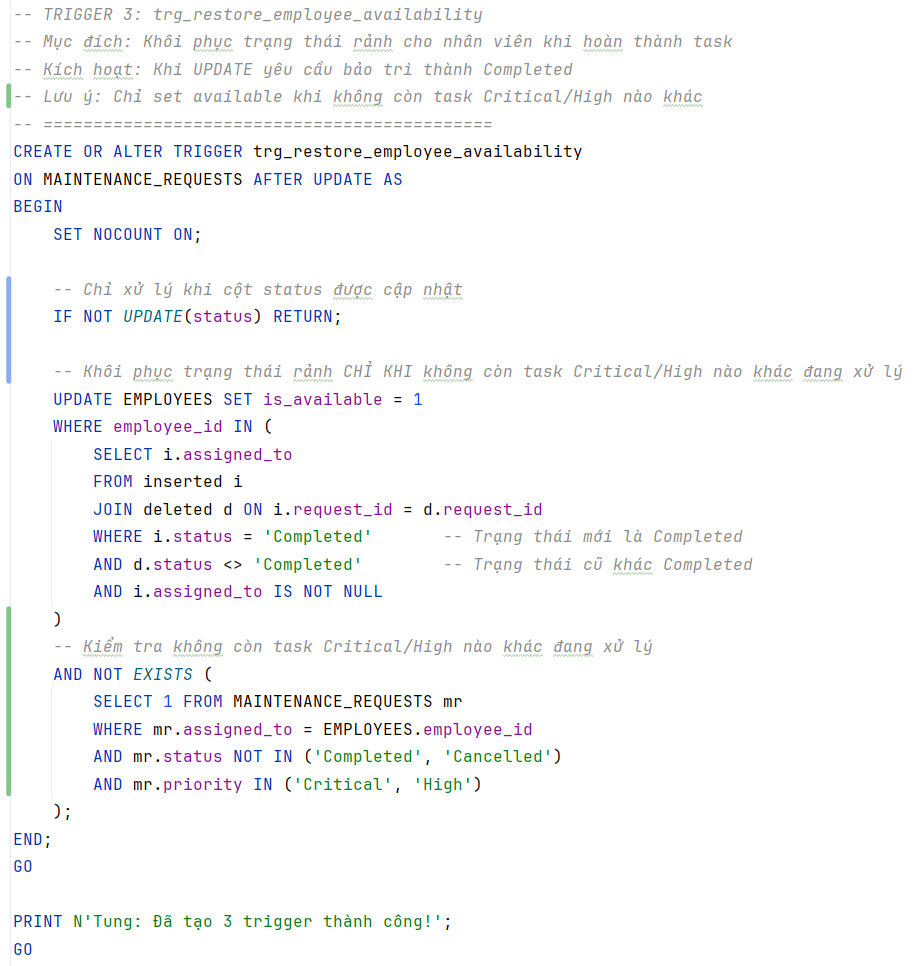
#### *b. trg\_update\_employee\_availability*

* **Bảng áp dụng**: MAINTENANCE\_REQUESTS
* **Loại trigger:** AFTER INSERT, UPDATE
* **Mục đích:** Đánh dấu nhân viên bận khi được giao task ưu tiên cao
* **Logic:** Khi có task Critical/High mới hoặc đổi assigned\_to, đặt is\_available = 0 cho nhân viên được giao



#### *c. trg\_restore\_employee\_availability*

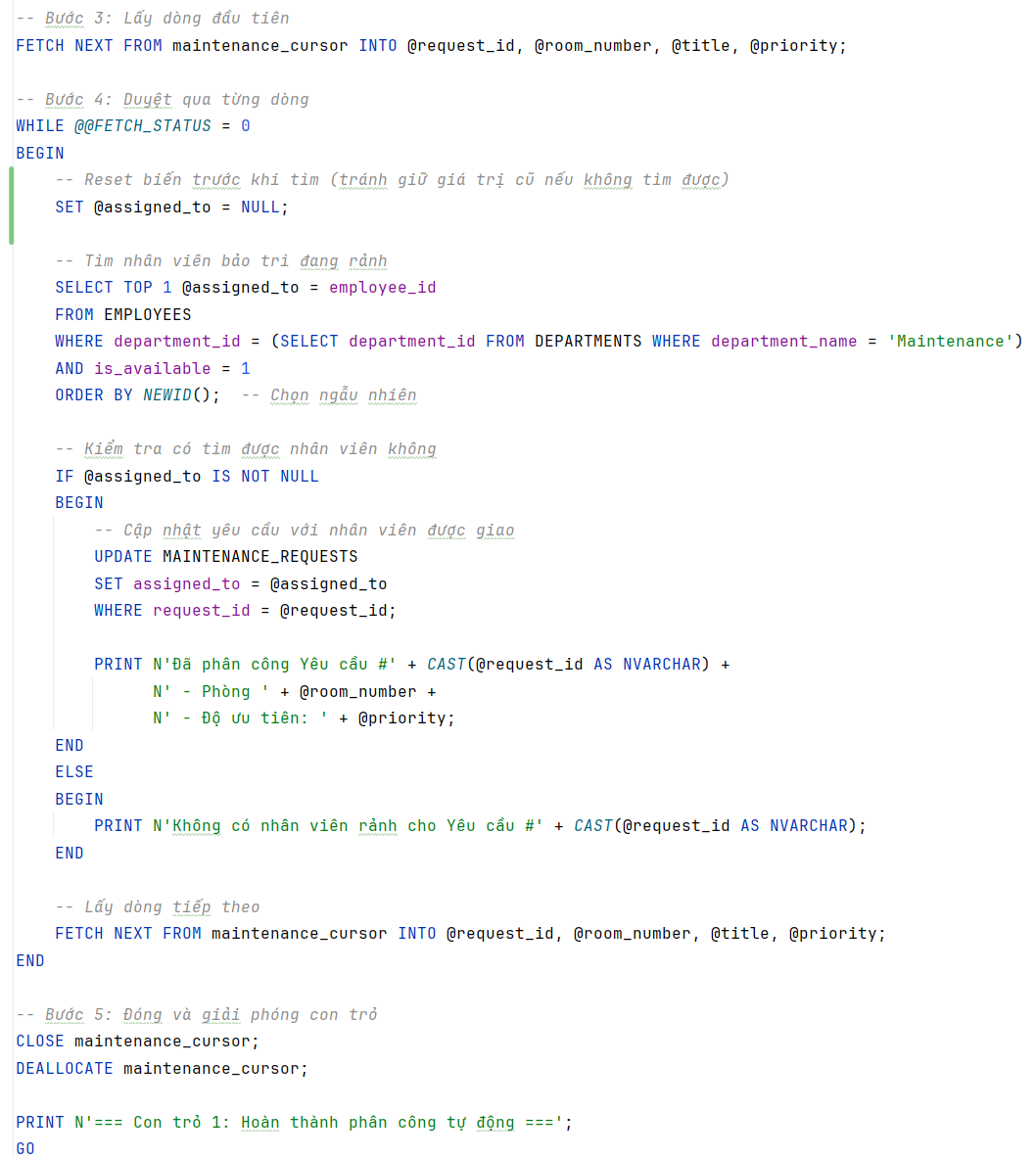
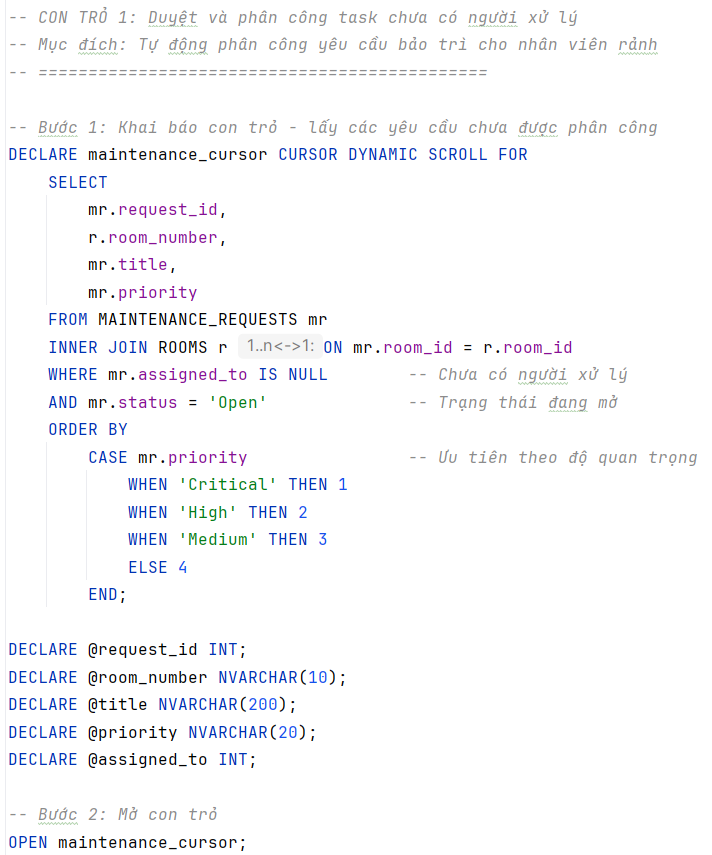
* **Bảng áp dụng:** MAINTENANCE\_REQUESTS
* **Loại trigger:** AFTER UPDATE
* **Mục đích:** Khôi phục trạng thái rảnh cho nhân viên khi hoàn thành task
* **Logic:** Khi status chuyển sang Completed, đặt is\_available = 1 CHỈ KHI nhân viên không còn task Critical/High nào khác đang xử lý



### 4.5. Cursor

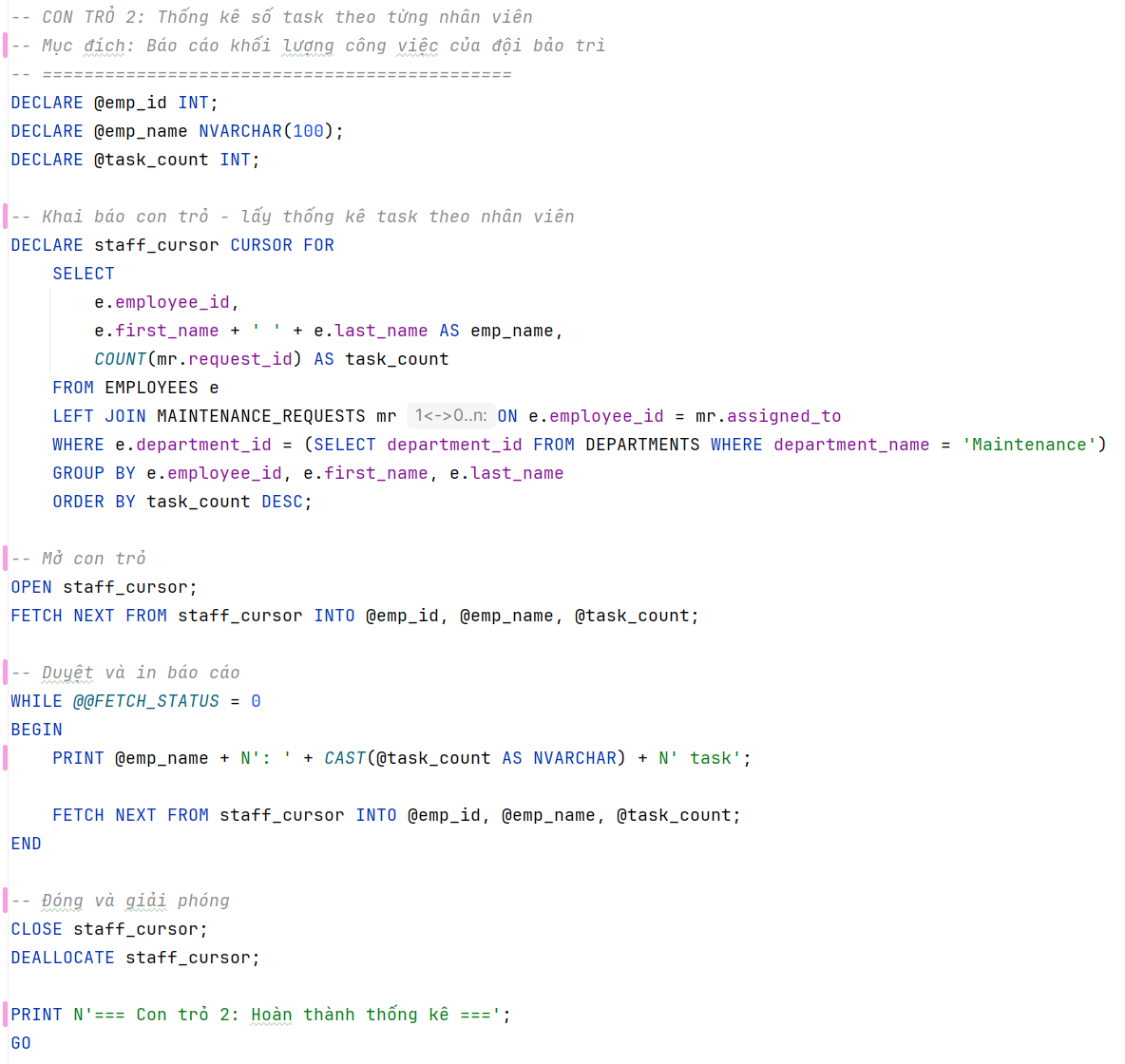
#### *a. Cursor 1: Phân công tự động yêu cầu bảo trì*

* **Mục đích:** Duyệt và phân công task chưa có người xử lý
* **Loại Cursor:** DYNAMIC SCROLL
* **Logic Cursor:**
  + Lấy các yêu cầu có assigned\_to IS NULL và status = 'Open'
  + Sắp xếp theo priority (Critical → High → Medium → Low)
  + Tìm nhân viên bảo trì đang rảnh (is\_available = 1)
  + Cập nhật assigned\_to cho từng yêu cầu
  + Xử lý trường hợp không có nhân viên rảnh (in thông báo)



#### *b. Cursor 2: Thống kê task theo nhân viên*

* **Mục đích:** Báo cáo khối lượng công việc của đội bảo trì
* **Logic Cursor:**
  + Duyệt qua từng nhân viên phòng Maintenance
  + Đếm số task được giao cho mỗi nhân viên
  + Sắp xếp theo số task giảm dần
  + In báo cáo: Tên nhân viên + số task



# 

# **Kết luận**

Qua quá trình thực hiện bài tập lớn môn Hệ Quản Trị Cơ Sở Dữ Liệu, nhóm chúng em đã hoàn thành việc xây dựng hệ thống quản lý khách sạn **Fantastic Hotel** với đầy đủ các chức năng cơ bản và nâng cao. Dự án bao gồm:

* **Thiết kế cơ sở dữ liệu** với mô hình ER logic, chuyển đổi sang lược đồ quan hệ chuẩn hóa, đảm bảo tính toàn vẹn dữ liệu.
* **Xây dựng các stored procedures** để xử lý nghiệp vụ đặt phòng, thanh toán, quản lý dịch vụ và vận hành khách sạn.
* **Triển khai triggers** để tự động hóa các quy trình và đảm bảo tính nhất quán của dữ liệu.
* **Tạo views và functions** hỗ trợ báo cáo, thống kê doanh thu theo nhiều chiều khác nhau.
* **Thiết kế hệ thống phân quyền** đảm bảo an toàn và bảo mật thông tin.

Bài tập lớn đã giúp nhóm củng cố kiến thức về thiết kế cơ sở dữ liệu, các kỹ thuật lập trình SQL Server, cũng như rèn luyện kỹ năng làm việc nhóm và quản lý dự án.

# 

# 

# **Lời cảm ơn**

Nhóm chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến **Thạc sĩ Nguyễn Ngọc Quỳnh Châu** – giảng viên hướng dẫn môn Hệ Quản Trị Cơ Sở Dữ Liệu.

Trong suốt quá trình học tập và thực hiện bài tập lớn, cô đã tận tình giảng dạy, truyền đạt những kiến thức nền tảng và chuyên sâu về cơ sở dữ liệu. Những góp ý quý báu và sự hướng dẫn nhiệt tình của cô đã giúp nhóm chúng em hoàn thiện dự án một cách tốt nhất.

Chúng em xin chúc cô luôn mạnh khỏe, hạnh phúc và tiếp tục đạt được nhiều thành công trong sự nghiệp giảng dạy.

**Trân trọng cảm ơn!**