

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Khoa Công nghệ Thông tin
Học phần: Thực hành cơ sở dữ liệu

BÁO CÁO HỆ THỐNG QUẢN LÝ THÔNG TIN KHÁM CHỮA BỆNH VÀ THANH TOÁN TẠI BỆNH VIỆN

Nhóm: 5

Thành viên:

20235731 - Võ Huy Hoàng

20235738 - Đỗ Phú Hưng

20235758 - Đoàn Đình Khôi

Hà Nội, 06/2025

MỤC LỤC

MỤC LỤC.....	2
Mô tả sơ bộ	3
Nhiệm vụ.....	3
Mô tả quy trình khám chữa bệnh và thanh toán:	3
Chức năng của hệ thống.....	4
Sơ đồ thực thể - liên kết.....	6
Bảng cơ sở dữ liệu.....	6
Các câu truy vấn cho hệ thống.....	7
Báo cáo các vấn đề gặp phải và cách giải quyết.....	21
Đánh giá về những lợi ích đạt được và các nhược điểm của kết quả.....	22
Nhiệm vụ mỗi người.....	22

Mô tả sơ bộ:

Hệ thống sẽ hỗ trợ đăng ký khám, khám lâm sàng và cận lâm sàng, quản lý điều trị và viện phí. Mỗi bước được tự động hoá qua giao diện nhập liệu và quy trình làm việc rõ ràng, giúp giảm thiểu sai sót, tăng tính minh bạch và rút ngắn thời gian chờ đợi.

Nhiệm vụ:

Mô tả các nghiệp vụ diễn ra trong bệnh viện chủ yếu liên quan đến bệnh nhân.

Đưa ra các chức năng cần có cho hệ thống, có sự phân quyền cho từng cá nhân tham gia.

Tạo nên mô hình cơ sở dữ liệu cho hệ thống.

Mô tả quy trình khám chữa bệnh và thanh toán:

1) Tiếp đón và tạo phiếu đăng ký khám

Khi bệnh nhân đến bệnh viện, trước tiên bệnh nhân sẽ lấy số, xếp hàng đợi đến lượt để gặp tiếp tân. Khi gặp tiếp tân, bệnh nhân sẽ nói, miêu tả tình trạng bệnh của mình và sẽ được hỗ trợ tạo phiếu đăng ký khám (mỗi lần khám sẽ có phiếu đăng ký khám riêng cho lần khám đó). Phiếu đăng ký khám sẽ có thông tin một bác sĩ tại chuyên khoa liên quan đến bệnh của bệnh nhân phụ trách khám cho bệnh nhân. VD bệnh nhân cảm thấy bị đau bụng sẽ được đăng ký khám với bác sĩ tại khoa nội.

2) Quy trình khám bệnh

Bệnh nhân và bác sĩ sẽ gặp nhau tại một phòng khám chuyên khoa. Tại đây bác sĩ sẽ khám cho bệnh nhân, tra hỏi về triệu chứng quan trọng của bệnh nhân. Bác sĩ sau khi khám sơ qua cho bệnh nhân đã đoán được khả năng cao bệnh nhân bị bệnh gì nhưng để xác định rõ và minh bạch thì bác sĩ sẽ chỉ định cho bệnh nhân đi thực hiện một loạt các xét nghiệm (gọi chung là khám cận lâm sàng, VD xét nghiệm máu, chụp x quang, chụp cắt lớp, ...).

Bệnh nhân sẽ tra cứu địa điểm từng nơi khám cận lâm sàng để đi thực hiện xét nghiệm, mỗi một ca khám cận lâm sàng này sẽ được phụ trách bởi viên bệnh viện (VD kĩ thuật viên xét nghiệm máu, kĩ thuật viên chụp x quang, ...) và nhân viên này sẽ ghi lại kết quả xét nghiệm vào hệ thống.

Sau khi thực hiện xong tất cả xét nghiệm, bác sĩ phụ trách bệnh nhân sẽ cho ra kết luận cuối cùng về bệnh của bệnh nhân (VD đã bị bệnh đau dạ dày) và cho ra kế hoạch điều trị cho bệnh nhân (VD không ăn cay, thức khuya nhiều,...).

3) Cấp đơn thuốc

Nếu bệnh nhân không bị bệnh nhẹ theo quan sát bác sĩ thì bác sĩ sẽ cấp cho bệnh nhân một đơn thuốc. Đơn thuốc này sẽ bao gồm các loại thuốc và cũng ghi cụ thể cả cách dung, số lượng đầy đủ. Bệnh nhân sẽ tự đi mua thuốc theo đơn thuốc đã cấp.

4) Nhập viện

Nếu bệnh nhân bị bệnh nặng, nghiêm trọng theo quan sát của bác sĩ thì bác sĩ sẽ yêu cầu bệnh nhân nhập viện. Khi nhập viện, bệnh nhân sẽ biết được phòng nơi mình nằm, ngày nhập và xuất viện. Ngoài ra bệnh nhân được cấp 1 nhân viên (y tá chăm sóc) để được tiến hành theo dõi. Khi thì bệnh sẽ phải trải qua một loạt các điều trị để chữa bệnh của mình cho đến khi xuất viện.

5) Thanh toán

Sau khi khám tất cả xong và đã được cấp đơn thuốc, hoặc sau khi xuất viện, bệnh nhân sẽ được cấp 1 hóa đơn để thanh toán tại quầy. Nếu bệnh nhân nhập viện thì tiền nhập viện sẽ là số ngày nằm viện nhân giá nhập viện/ngày. Hệ thống sẽ tự tính tổng tiền tất cả các dịch vụ lại và ghi vào hóa đơn để bệnh nhân đi thanh toán. Bệnh nhân có thể tự tra bảng giá dịch vụ từng loại dịch vụ.

Chức năng của hệ thống

1) Bệnh nhân

Chức năng	Đầu vào	Đầu ra
Thông tin bệnh nhân		Có thể xem các thông tin cơ bản (tên, ngày sinh, số cccd, bhyt)
Thông tin bác sĩ		Có thể xem các thông tin như (tên, chuyên ngành)
Thông tin buổi khám		Có thể xem được các thông tin về thời gian, địa điểm, giá của buổi khám.
Đơn thuốc		Có thể xem được thông tin về thuốc như tên, liều lượng, HDSD
Nội trú		Biết được ngày nhập viện, xuất viện, giá của phiên nội trú.
Cận lâm sàng		Biết được loại xét nghiệm cận lâm sàng, thời gian, địa điểm, về kết quả.
Hóa đơn		Biết được số tiền phải trả

2) Bác sĩ/Kỹ thuật viên

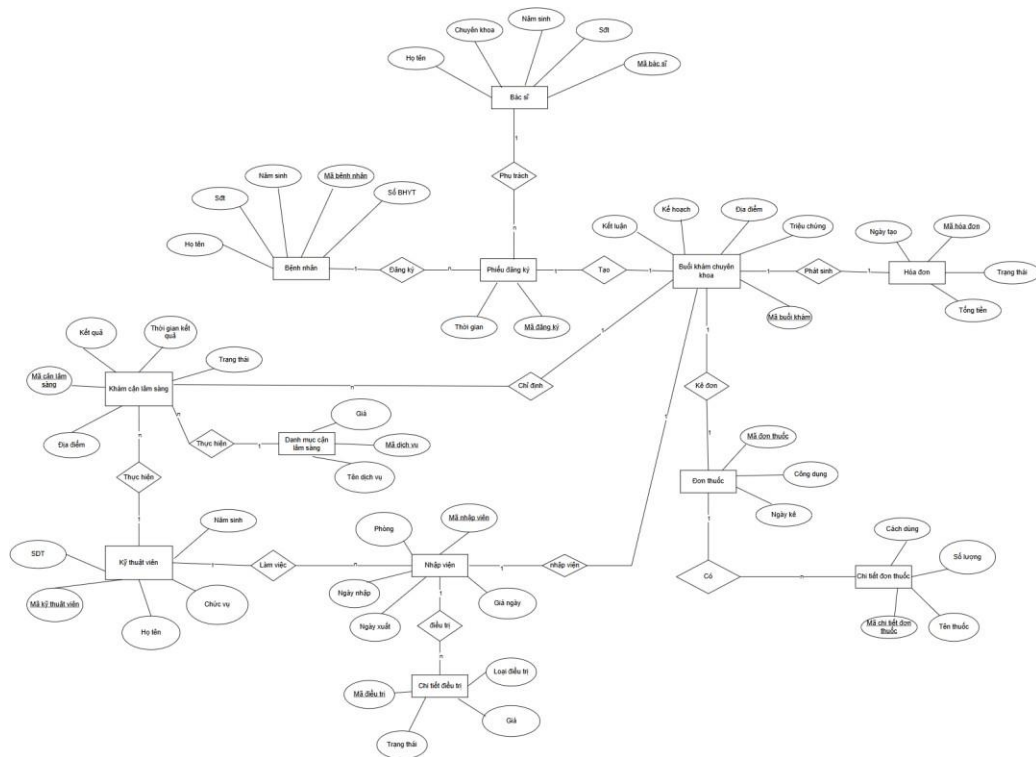
Chức năng	Đầu vào	Đầu ra
Thông tin bác sĩ/kỹ thuật viên		Có thể xem các thông tin cơ bản như (tên, ngày sinh, chuyên khoa)
Thông tin buổi khám		Có thể xem được các thông tin về thời gian, địa điểm của buổi khám.
Đơn thuốc(Bác sĩ)	Tạo đơn thuốc cho bệnh nhân	Thông tin về đơn thuốc được lưu lại
Cận lâm sàng(Bác sĩ)	Cập nhật loại xét nghiệm, thời gian, địa điểm	Thông tin về phiên cận lâm sàng được cập nhật
Cận lâm sàng(Kỹ thuật viên)	Cập nhật kết quả	Kết quả đã được cập nhật

3) Quản lý

Chức năng	Đầu vào	Đầu ra
Thông tin bác sĩ/nhân viên		Có thể thêm/sửa/xóa các thông tin về bác sĩ và nhân viên
Thông tin bệnh nhân	Cập nhật thông tin bệnh nhân	Thông tin bệnh nhân sau chỉnh sửa
Thông tin buổi khám	Cập nhật các thông tin về thời gian, địa điểm, giá tiền	Thông tin về buổi khám được cập nhật
Nội trú	Cập nhật ngày nhập viện, ngày xuất, giá tiền	Thông tin về phiên nội trú được cập nhật

Cận lâm sàng	Cập nhật giá tiền	Thông tin về giá của buổi khám được cập nhật
--------------	-------------------	--

Sơ đồ thực thể - liên kết



Bảng cơ sở dữ liệu

Benh_nhan(**Ma_benh_nhan**, Ho_ten, Sdt, Nam_sinh, So BHYT)

Bac_si(**Ma_bac_si**, Ho_ten, Sdt, Nam_sinh, Chuyen_khoa)

Pheuu_dang_ky(**Ma_dang_ky**, Thoi_gian, Ma_benh_nhan, Ma_bac_si, Ma_buoi_kham)

Buoi_kham_chuyen_khoa(**Ma_buoi_kham**, Dia_diem, Trieu_chung, Ket_luan, Ke_hoach, Ma_nhap_vien, Ma_don_thuoc, Ma_hoa_don)

Kham_can_lam_sang(**Ma_can_lam_sang**, Dia_diem, Ket_qua, Thoi_gian_ket_qua, Trang_thai, Ma_dich_vu, Ma_ky_thuat_vien, ma_buoi_kham)

Danh_muc_can_lam_sang(**Ma_dich_vu**, Ten_dich_vu, Gia)

Ky_thuat_vien(**Ma_ky_thuat_vien**, Ho_ten, Sdt, Chuc_vu, Nam_sinh)

Nhap_vien(**Ma_nhap_vien**, Phong, Ngay_nhap, Ngay_xuat, Gia_ngay, Ma_ky_thuat_vien)

Chi_tiet_dieu_tri(**Ma_dieu_tri**, Loai_dieu_tri, Gia, Trang_thai, Ma_nhap_vien)

Don_thuoc(**Ma_don_thuoc**, Cong_dung, Ngay_ke)

Chi_tiet_don_thuoc(**Ma_chi_tiet_don_thuoc**, Ten_thuoc, Cach_dung, So_luong,
Ma_don_thuoc)

Hoa_don(**Ma_hoa_don**, Ngay_tao, Tong_tien, Trang_thai)

Các câu truy vấn cho hệ thống

Giao diện bệnh nhân (Hưng)

1) Xem lịch sử khám

```
SELECT thoi_gian AS ngay_kham, bs.ho_ten AS bac_si, bs.chuyen_khoa, trieu_chung, ket_luan,  
ke_hoach
```

```
FROM phieu_dang_ky
```

```
JOIN bac_si bs USING (ma_bac_si)
```

```
JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)
```

```
WHERE ma_benh_nhan = 'BN001'
```

```
ORDER BY thoi_gian DESC;
```

+ Index cho WHERE + ORDER BY

```
CREATE INDEX idx_pd_bn_thoi_gian ON phieu_dang_ky (ma_benh_nhan, thoi_gian DESC);
```

+ Index cho JOIN với bảng bac_si

```
CREATE INDEX idx_pd_ma_bac_si ON phieu_dang_ky (ma_bac_si);
```

+ Index cho JOIN với bảng buoi_kham_chuyen_khoa

```
CREATE INDEX idx_pd_ma_buoi_kham ON phieu_dang_ky (ma_buoi_kham);
```

2) Xem thông tin bác sĩ khám

```
SELECT DISTINCT bs.ho_ten, bs.sdt, bs.nam_sinh, chuyen_khoa, thoi_gian AS ngay_kham
```

```
FROM phieu_dang_ky
```

```
JOIN bac_si bs USING (ma_bac_si)
```

```
WHERE ma_benh_nhan = 'BN001'
```

```
ORDER BY thoi_gian DESC;
```

+ Index cho WHERE + ORDER BY

```
CREATE INDEX idx_pd_bn_thoi_gian ON phieu_dang_ky (ma_benh_nhan, thoi_gian DESC);
```

+ Index cho JOIN với bảng bac_si

```
CREATE INDEX idx_pd_ma_bac_si ON phieu_dang_ky (ma_bac_si);
```

3) Xem các chỉ định khám cận lâm sàng nào chưa thực hiện

```
SELECT ten_dich_vu, k.dia_diem, trang_thai
```

```
FROM phieu_dang_ky
```

```
JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)
```

```
JOIN kham_can_lam_sang k USING (ma_buoi_kham)
```

```
JOIN danh_muc_can_lam_sang USING (ma_dich_vu)
```

```
WHERE ma_benh_nhan = 'BN001' AND k.trang_thai = 'Chưa khám'
```

```
ORDER BY ma_can_lam_sang;
```

+ Index vừa lọc bệnh nhân, vừa JOIN sang buoi_kham_chuyen_khoa

```
CREATE INDEX idx_pd_bn_and_mb ON phieu_dang_ky (ma_benh_nhan, ma_buoi_kham);
```

4) Xem thông tin xét nghiệm và kết quả

```
SELECT ten_dich_vu, k.dia_diem, k.thoi_gian_ket_qua, k.ket_qua, ktv.ho_ten AS  
ky_thuat_vien
```

```
FROM phieu_dang_ky
```

```
JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)
```

```
JOIN kham_can_lam_sang k USING (ma_buoi_kham)
```

```
JOIN danh_muc_can_lam_sang USING (ma_dich_vu)
```

```
JOIN ky_thuat_vien ktv USING (ma_ky_thuat_vien)
```

```
WHERE ma_benh_nhan = 'BN001'
```

```
ORDER BY thoi_gian_ket_qua DESC;
```

+ Index vừa lọc bệnh nhân vừa join buoi_kham

```
CREATE INDEX idx_pd_bn_bk ON phieu_dang_ky (ma_benh_nhan, ma_buoi_kham);
```


+ Index trên kham_can_lam_sang giúp join buoi_kham và tìm nhanh thoi_gian_ket_qua kết quả

```
CREATE INDEX idx_kcls_bk_tkkq ON kham_can_lam_sang (ma_buoi_kham, thoi_gian_ket_qua DESC);
```

5) Xem toàn bộ đơn thuốc đã cấp

```
SELECT ma_don_thuoc, ngay_ke, cong_dung,  
ARRAY_AGG(ten_thuoc||'x'||so_luong) AS danh_sach_thuoc  
FROM don_thuoc JOIN chi_tiet_don_thuoc USING (ma_don_thuoc)  
JOIN buoi_kham_chuyen_khoa USING (ma_don_thuoc)  
JOIN phieu_dang_ky USING (ma_buoi_kham)  
WHERE ma_benh_nhan = 'BN001'  
GROUP BY ma_don_thuoc  
ORDER BY ngay_ke DESC;
```

+ Index giúp join nhanh buổi khám đến đơn thuốc

```
CREATE INDEX idx_bk_dn ON buoi_kham_chuyen_khoa (ma_don_thuoc);
```

+ Index giúp join nhanh từ đơn thuốc lên chi tiết đơn thuốc

```
CREATE INDEX idx_ctdt_dn ON chi_tiet_don_thuoc (ma_don_thuoc);
```

+ Index sắp xếp kết quả theo ngày đơn

```
CREATE INDEX idx_dt_ngay_ke ON don_thuoc (ngay_ke DESC);
```

6) Xem thông tin nhập viện

```
SELECT ma_nhap_vien, phong, ngay_nhap, ngay_xuat, gia_ngay, ktv.ho_ten AS ky_thuat_vien  
FROM phieu_dang_ky  
JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)  
JOIN nhap_vien USING (ma_nhap_vien)  
JOIN benh_nhan USING (ma_benh_nhan)  
JOIN ky_thuat_vien ktv USING (ma_ky_thuat_vien)  
WHERE ma_benh_nhan = 'BN001'
```

ORDER BY ngay_nhap DESC;

+ Index vừa lọc bệnh nhân, vừa join buổi khám

CREATE INDEX idx_pd_bn_buoi ON phieu_dang_ky (ma_benh_nhan, ma_buoi_kham);

+ Index join ngay buoi_kham_chuyen_khoa đến nhập viện

CREATE INDEX idx_bk_nv ON buoi_kham_chuyen_khoa (ma_buoi_kham, ma_nhap_vien);

+ Index vừa join kỹ thuật viên và sắp xếp ngày nhập

CREATE INDEX idx_nv_nv_ngay ON nhap_vien (ma_nhap_vien, ngay_nhap DESC);

7) Xem thông tin các điều trị

SELECT ma_dieu_tri, loai_dieu_tri, trang_thai

FROM chi_tiet_dieu_tri

JOIN nhap_vien USING (ma_nhap_vien)

JOIN buoi_kham_chuyen_khoa USING (ma_nhap_vien)

JOIN phieu_dang_ky USING (ma_buoi_kham)

WHERE ma_benh_nhan = 'BN001';

+ Index giúp lọc bệnh nhân và join buoi_kham

CREATE INDEX idx_pd_bn_buoi ON phieu_dang_ky (ma_benh_nhan, ma_buoi_kham);

+ Index giúp join buoi_kham_chuyen_khoa tiếp xuống nhap_vien

CREATE INDEX idx_bk_nv ON buoi_kham_chuyen_khoa (ma_buoi_kham, ma_nhap_vien);

8) Xem hóa đơn (tổng tiền khám chữa bệnh)

SELECT ma_hoa_don, ngay_tao, tong_tien

FROM phieu_dang_ky

JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)

JOIN hoa_don USING (ma_hoa_don)

WHERE ma_benh_nhan = 'BN001'

ORDER BY ngay_tao DESC;

+ Index join buổi khám

```
CREATE INDEX idx_pd_bn_buoi ON phieu_dang_ky (ma_buoi_kham);
```

+ Index join nhanh sang hóa đơn

```
CREATE INDEX idx_bk_hoa_don ON buoi_kham_chuyen_khoa (ma_hoa_don);
```

+ Index sắp xếp theo ngày tạo giảm dần

```
CREATE INDEX idx_hd_ngay_tao ON hoa_don (ngay_tao DESC);
```

9) Xem chi tiết giá từng loại điều trị

```
SELECT ma_dieu_tri, loai_dieu_tri, gia
```

```
FROM chi_tiet_dieu_tri
```

```
JOIN nhap_vien USING (ma_nhap_vien)
```

```
JOIN buoi_kham_chuyen_khoa USING (ma_nhap_vien)
```

```
JOIN phieu_dang_ky USING (ma_buoi_kham)
```

```
WHERE ma_benh_nhan = 'BN001' AND ngay_nhap = '2025-04-01';
```

+ Index JOIN chi_tiet_dieu_tri đến nhap_vien

```
CREATE INDEX idx_ctdt_ma_nhap_vien ON chi_tiet_dieu_tri (ma_nhap_vien);
```

+ Index JOIN nhap_vien đến buoi_kham_chuyen_khoa

```
CREATE INDEX idx_bk_ma_nhap_vien ON buoi_kham_chuyen_khoa (ma_nhap_vien);
```

+ Index JOIN buoi_kham_chuyen_khoa đến phieu_dang_ky

```
CREATE INDEX idx_pd_ma_buoi_kham ON phieu_dang_ky (ma_buoi_kham);
```

10) Xem chi tiết tiền từng dịch vụ khám cận lâm sàng

```
SELECT ma_dich_vu, ten_dich_vu, gia
```

```
FROM phieu_dang_ky
```

```
JOIN buoi_kham_chuyen_khoa USING (ma_buoi_kham)
```

```
JOIN kham_can_lam_sang USING (ma_buoi_kham)
```

```
JOIN danh_muc_can_lam_sang USING (ma_dich_vu)
```

```
WHERE ma_benh_nhan = 'BN001' AND thoi_gian = '2025-04-01';
```

+ Index trên phieu_dang_ky để JOIN nhanh sang buoi_kham_chuyen_khoa

```
CREATE INDEX idx_pd_ma_buoi_kham ON phieu_dang_ky (ma_buoi_kham);  
+ Index trên kham_can_lam_sang để JOIN nhanh từ buoi_kham_chuyen_khoa  
CREATE INDEX idx_kcls_ma_buoi_kham ON kham_can_lam_sang (ma_buoi_kham);
```

Giao diện bác sĩ và kỹ thuật viên (Hoàng)

1) Tính số bệnh nhân mà bác sĩ cần khám trong một ngày nào đấy

```
SELECT ma_bac_si, COUNT(ma_benh_nhan) as so_luot_dang_ky  
FROM phieu_dang_ky  
WHERE thoi_gian = '2025-06-19'  
GROUP BY ma_bac_si;
```

```
Index: CREATE INDEX idx_phieu_dang_ky_thoi_gian ON phieu_dang_ky(thoi_gian);  
CREATE INDEX idx_phieu_dang_ky_ma_bac_si ON phieu_dang_ky(ma_bac_si);
```

2) Liệt kê các buổi khám có nhiều chỉ định cận lâm sàng nhất

```
WITH tmp AS (  
SELECT bk.ma_buoi_kham, COUNT(kcls.ma_can_lam_sang) AS so_chi_dinh_cls  
FROM buoi_kham_chuyen_khoa bk  
JOIN kham_can_lam_sang kcls USING(ma_buoi_kham)  
GROUP BY bk.ma_buoi_kham)  
SELECT ma_buoi_kham  
FROM tmp  
WHERE so_chi_dinh_cls = (SELECT max(so_chi_dinh_cls) FROM tmp)  
Index: CREATE INDEX idx_kham_cls_ma_buoi_kham ON  
kham_can_lam_sang(ma_buoi_kham);
```

3) Tìm các dịch vụ cận lâm sàng ít được chỉ định

```
with tmp as (  
select ma_dich_vu, ten_dich_vu, count(ma_can_lam_sang) as so_lan_chi_dinh  
from danh_muc_can_lam_sang  
join kham_can_lam_sang using(ma_dich_vu)  
group by ma_dich_vu, ten_dich_vu )  
select ma_dich_vu, ten_dich_vu  
from tmp  
where so_lan_chi_dinh <= all(select so_lan_chi_dinh from tmp)  
Index: CREATE INDEX idx_kham_cls_ma_dich_vu ON kham_can_lam_sang(ma_dich_vu);
```

4) Thống kê các bệnh với số lượng bệnh nhân nhiều nhất trong tháng

```
with tmp as  
(SELECT ket_luan, count(distinct ma_benh_nhan) AS tong_so_benh_nhan  
FROM buoi_kham_chuyen_khoa  
JOIN phieu_dang_ky USING(ma_buoi_kham)  
WHERE extract(month from thoi_gian) = extract(month from current_date)  
GROUP BY ket_luan)  
SELECT ket_luan FROM tmp WHERE tong_so_benh_nhan >= ALL(select  
tong_so_benh_nhan from tmp)  
Index: CREATE INDEX idx_phieu_dang_ky_thoi_gian ON phieu_dang_ky(thoi_gian);
```

5) Xem kết quả khám gần nhất của bệnh nhân

```
select ket_luan  
from buoi_kham_chuyen_khoa  
join phieu_dang_ky using(ma_buoi_kham)  
where ma_benh_nhan = 'BN005'  
order by thoi_gian desc limit 1;
```

Index: CREATE INDEX idx_phieu_dang_ky_ma_benh_nhan ON
phieu_dang_ky(ma_benh_nhan);

CREATE INDEX idx_phieu_dang_ky_thoi_gian_desc ON phieu_dang_ky(thoi_gian DESC);

6) Xem danh sách bệnh nhân đã khám với bác sĩ trong 7 ngày gần nhất

```
select ma_benh_nhan, ho_ten
```

```
from benh_nhan join phieu_dang_ky using(ma_benh_nhan)
```

```
where ma_bac_si = 'BS001' and thoi_gian >= current_date - INTERVAL '7 days';
```

Index: CREATE INDEX idx_pdk_bac_si_thoi_gian ON phieu_dang_ky(ma_bac_si, thoi_gian);

7) Thống kê số lần khám và số lượng bệnh nhân đã khám theo chuyên khoa của bác sĩ

```
select count(ma_dang_ky) as so_lan_kham, count(distinct ma_benh_nhan) as  
so_luong_benh_nhan, chuyen_khoa
```

```
from phieu_dang_ky
```

```
join bac_si using(ma_bac_si)
```

```
group by chuyen_khoa;
```

Index: CREATE INDEX idx_bac_si_chuyen_khoa ON bac_si(chuyen_khoa);

8) Danh sách bệnh nhân đang nhập viện chưa xuất viện (vẫn đang điều trị nội trú)

```
select ma_benh_nhan, ho_ten
```

```
from benh_nhan
```

```
join phieu_dang_ky using(ma_benh_nhan)
```

```
join buoi_kham_chuyen_khoa using(ma_buoi_kham)
```

```
join nhap_vien using(ma_nhap_vien)
```

```
where ngay_xuat > current_date;
```

Index: CREATE INDEX idx_nhap_vien_ngay_xuat ON nhap_vien(ngay_xuat);

9) Danh sách bệnh nhân đã từng được kê đơn với hơn 3 loại thuốc trong một buổi khám

```
with tmp as (
```

```

select ma_benh_nhan, ho_ten, ma_dang_ky, count(ma_chi_tiet_don_thuoc) as
so_luong_loai_thuoc

from benh_nhan bn

join phieu_dang_ky using(ma_benh_nhan)

join buoi_kham_chuyen_khoa using(ma_buoi_kham)

join don_thuoc using(ma_don_thuoc)

join chi_tiet_don_thuoc using(ma_don_thuoc)

group by ma_benh_nhan, ho_ten, ma_dang_ky

having count(ma_chi_tiet_don_thuoc) > 3)

select distinct ma_benh_nhan, ho_ten from tmp;

```

Index: CREATE INDEX idx_ct_don_thuoc_ma_don_thuoc ON
chi_tiet_don_thuoc(ma_don_thuoc);

10) Tìm các bệnh nhân khám nhiều hơn 2 lần trong tháng hiện tại

```

select ma_benh_nhan, ho_ten, count(ma_dang_ky) as so_lan_kham

from benh_nhan

join phieu_dang_ky using(ma_benh_nhan)

where extract(month from thoi_gian) = extract(month from current_date)

group by ma_benh_nhan, ho_ten

having count(ma_dang_ky)>2;

Index: CREATE INDEX idx_phieu_dang_ky_thoi_gian ON phieu_dang_ky(thoi_gian);

```

Giao diện quản lý(Khôi)

1) Quản lý đếm số bệnh nhân nội trú trong khoảng thời gian nhất định

```

SELECT COUNT(*) AS so_benh_nhan_noi_tru

FROM nhap_vien

WHERE

```

```
ngay_nhap <= DATE '2025-04-30'
```

```
AND (ngay_xuat >= DATE '2025-04-01' OR ngay_xuat IS NULL);
```

```
CREATE INDEX idx_nhap_vien_ngay ON nhap_vien(ngay_nhap, ngay_xuat);
```

Index này sẽ giúp tối ưu hóa điều kiện lọc theo ngày nhập và ngày xuất viện.

2) Quản lý đếm số ca cận lâm sàng mà 1 nhân viên phải làm trong ngày

```
SELECT
```

```
ktv.ho_ten,
```

```
kcls.thoi_gian_ket_qua::date AS ngay,
```

```
COUNT(*) AS so_ca
```

```
FROM kham_can_lam_sang kcls
```

```
JOIN ky_thuat_vien ktv ON kcls.ma_ky_thuat_vien = ktv.ma_ky_thuat_vien
```

```
WHERE ktv.ho_ten = 'KTV. Vũ Bình'
```

```
AND kcls.thoi_gian_ket_qua::date = DATE '2025-04-02'
```

```
GROUP BY ktv.ho_ten, kcls.thoi_gian_ket_qua::date;
```

```
CREATE INDEX idx_kcls_kythuat_ngay ON kham_can_lam_sang(ma_ky_thuat_vien,  
thoi_gian_ket_qua);
```

Index này sẽ giúp tối ưu hóa điều kiện lọc theo mã kỹ thuật viên và thời gian thực hiện kết quả xét nghiệm.

3) Gán mặc định trạng thái = 'Chưa thanh toán' cho hóa đơn

```
CREATE OR REPLACE FUNCTION fn_default_trang_thai_hd()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
IF NEW.trang_thai IS NULL THEN
```

```
NEW.trang_thai := 'Chưa thanh toán';
```

```
END IF;
```

```
RETURN NEW;
```

```
END;
```



```

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_default_trang_thai_hd

BEFORE INSERT ON hoa_don

FOR EACH ROW EXECUTE FUNCTION fn_default_trang_thai_hd();

```

4) Tự động kiểm tra số ca cận lâm sàng của kỹ thuật viên trong ngày không vượt quá giới hạn (cảnh báo nhưng vẫn cho chèn)

```

CREATE OR REPLACE FUNCTION fn_canh_bao_gioi_han_ktv()

RETURNS TRIGGER AS $$

DECLARE

ca_trong_ngay INT;

BEGIN

SELECT COUNT(*) INTO ca_trong_ngay

FROM kham_can_lam_sang

WHERE

ma_ky_thuat_vien = NEW.ma_ky_thuat_vien

AND thoi_gian_ket_qua::date = NEW.thoi_gian_ket_qua::date;

IF ca_trong_ngay >= 10 THEN

    RAISE WARNING ' Kỹ thuật viên % đã thực hiện % ca trong ngày % – vượt quá giới hạn khuyến nghị!',

    NEW.ma_ky_thuat_vien, ca_trong_ngay, NEW.thoi_gian_ket_qua::date;

END IF;

RETURN NEW;

END; $$ LANGUAGE plpgsql; CREATE TRIGGER trg_canh_bao_ktv BEFORE INSERT ON

kham_can_lam_sang FOR EACH ROW EXECUTE FUNCTION fn_canh_bao_gioi_han_ktv();

```

5) Kiểm tra ngày nhập sớm hơn ngày xuất

```

CREATE OR REPLACE FUNCTION fn_chk_ngay_xuat()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.ngay_xuat IS NOT NULL

```

```

AND NEW.ngay_xuat < NEW.ngay_nhap THEN

RAISE EXCEPTION 'ngay_xuat (%) không được trước ngay_nhap (%)', NEW.ngay_xuat,
NEW.ngay_nhap;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_chk_ngay_xuat BEFORE INSERT OR UPDATE ON nhap_vien FOR EACH
ROW EXECUTE FUNCTION fn_chk_ngay_xuat();

```

6) Gán mặc định trạng_thai = 'Chưa khám' cho KCLS

```

CREATE OR REPLACE FUNCTION fn_default_kcls_status()

RETURNS TRIGGER AS $$

BEGIN

IF NEW.trang_thai IS NULL THEN

NEW.trang_thai := 'Chưa khám';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_default_kcls_status

BEFORE INSERT ON kham_can_lam_sang

FOR EACH ROW EXECUTE FUNCTION fn_default_kcls_status();

```

7) Ghi lại mã và họ tên bệnh nhân sau khi xóa

```

CREATE TABLE IF NOT EXISTS benh_nhan_log (

ma_benh_nhan VARCHAR(20),

ho_ten VARCHAR(100),

xoa_luc TIMESTAMP DEFAULT NOW() );

```

```

CREATE OR REPLACE FUNCTION fn_log_delete_bn()
RETURNS TRIGGER AS $$
BEGIN
INSERT INTO benh_nhan_log (ma_benh_nhan, ho_ten)
VALUES (OLD.ma_benh_nhan, OLD.ho_ten);
RETURN OLD; END; $$ LANGUAGE plpgsql;

CREATE TRIGGER trg_log_delete_bn
AFTER DELETE ON benh_nhan
FOR EACH ROW EXECUTE FUNCTION fn_log_delete_bn();

```

8) Cập nhật tổng tiền hóa đơn khi xuất viện

```

CREATE OR REPLACE FUNCTION fn_update_hd_after_discharge()
RETURNS TRIGGER AS $$
DECLARE
v_days INT;
v_ma_hd VARCHAR(20);
BEGIN
IF NEW.ngay_xuat IS NULL THEN
RETURN NEW;
END IF;

v_days := GREATEST( (NEW.ngay_xuat - NEW.ngay_nhap) + 1 , 1 );

SELECT ma_hoa_don
INTO v_ma_hd
FROM buoi_kham_chuyen_khoa
WHERE ma_nhap_vien = NEW.ma_nhap_vien LIMIT 1;

IF v_ma_hd IS NOT NULL THEN
UPDATE hoa_don

```

```

SET tong_tien = COALESCE(tong_tien, '0')::NUMERIC + (v_days * NEW.gia_ngay)::NUMERIC
WHERE ma_hoa_don = v_ma_hd;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_update_hd_after_discharge
AFTER UPDATE OF ngay_xuat, gia_ngay ON nhap_vien
FOR EACH ROW EXECUTE FUNCTION fn_update_hd_after_discharge();

```

9) Tự động kiểm tra nếu tổng tiền hóa đơn vượt 10 triệu thì gán trạng thái “Chờ duyệt”

```

CREATE OR REPLACE FUNCTION fn_check_tong_tien_hd()
RETURNS TRIGGER AS $$

BEGIN

IF NEW.tong_tien::NUMERIC > 10000000 THEN

NEW.trang_thai := 'Chờ duyệt';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_check_tong_tien_hd
BEFORE INSERT OR UPDATE ON hoa_don
FOR EACH ROW EXECUTE FUNCTION fn_check_tong_tien_hd();

```

10) Giới hạn đăng ký trong giờ hành chính

```

CREATE OR REPLACE FUNCTION fn_gio_hanh_chinh_only()
RETURNS TRIGGER AS $$

```

```

DECLARE

v_gio INTEGER;

BEGIN

v_gio := EXTRACT(HOUR FROM NEW.thoi_gian);

IF v_gio < 8 OR v_gio >= 17 THEN

RAISE EXCEPTION ' Chỉ cho phép đăng ký trong giờ hành chính (08:00 - 17:00). Thời
gian bạn chọn: %', NEW.thoi_gian;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_gio_hanh_chinh_only

BEFORE INSERT ON phieu_dang_ky FOR EACH ROW

EXECUTE FUNCTION fn_gio_hanh_chinh_only();

```

Báo cáo các vấn đề gặp phải và cách giải quyết

1) Không xác định rõ yêu cầu nghiệp vụ ngay từ đầu, mô hình bị lan man, thiếu và thừa.

Cách giải quyết: chỉnh sửa lại nghiệp vụ và sửa lại sơ đồ ERD sao cho chúng ăn khớp với nhau.

2) Liên kết 2 liên kết n-n bằng khóa ngoại (Bệnh nhân - Bác sĩ)

Cách giải quyết: Thêm thuộc tính buổi khám giữa bác sĩ và bệnh nhân (Bác sĩ-> Buổi khám <- Bệnh nhân).

3) Tạo ra các chu kỳ vòng tròn khi tạo sơ đồ ERD khiến mô hình trở nên mơ hồ.

(Bệnh nhân -> session -> bác sĩ -> đơn thuốc -> bệnh nhân)

Cách giải quyết: Cho đơn thuốc là khóa ngoại của buổi khám để không tạo ra các chu kỳ vòng tròn

4) Có các thực thể liên kết 1-1 nhưng có ít thông tin phụ (bệnh nhân -> hồ sơ bệnh nhân) khiến tăng độ phức tạp không cần thiết của sơ đồ ERD và database

Cách giải quyết: gộp chung bệnh nhân và hồ sơ bệnh nhân làm một, giúp đơn giản hóa sơ đồ ERD và database.

Đánh giá về những lợi ích đạt được và các nhược điểm của kết quả

Lợi ích

Database cho phép lưu trữ và cập nhật dữ liệu của bệnh nhân, bác sĩ vào các buổi khám một cách tường minh và dễ kiểm soát.

Có thể thực hiện các chức năng cơ bản của bệnh viện như đăng kí khám, xuất đơn thuốc, ghi lại các kết quả khám của bệnh nhân.

Người dùng có thể dễ dàng truy xuất các thông tin đã được lưu trên hệ thống.

Nhược điểm

Chưa thực hiện các chức năng phức tạp như ghi chép các xét nghiệm lâm sàng của bệnh nhân nội trú.

Chưa thiết kế chi tiết được các dữ liệu liên quan đến kho thuốc, xuất nhập thuốc cho bệnh viện.

Chưa có các chức năng tính lương cho bác sĩ và các kỹ thuật viên.

Cần chia nhỏ các chức vụ của các kỹ thuật viên. (Kỹ thuật viên x-quang thì chỉ thực hiện chiếu chụp, các kỹ thuật viên liên quan đến xét nghiệm máu thì chỉ có thể xét nghiệm máu...)

Nhiệm vụ mỗi người

Đỗ Phú Hưng: Thiết kế nghiệp vụ và thiết kế database

Đoàn Đình Khôi: Mô tả chức năng

Võ Huy Hoàng: Thiết kế database và vẽ sơ đồ ERD