

BÁO CÁO ĐỒ ÁN CUỐI KỲ THỰC HÀNH CƠ SỞ DỮ LIỆU

Nhóm 7:

Đặng Phương Nam - 20225892

Nguyễn Việt Phong – 20225901

Bàng Tiến Thành – 20225669

I. Mô tả

Tên đồ án:

- Hệ thống hỗ trợ khám chữa bệnh tại phòng khám.

Bối cảnh:

- Xây dựng CSDL để quản lý thông tin liên quan đến các bệnh án của bệnh nhân, áp dụng cho 1 phòng khám.
- Lưu trữ và quản lý các thông tin phục vụ khám chữa bệnh.

Mô tả cơ sở dữ liệu:

- Cơ sở dữ liệu dùng cho tổ chức phòng khám và cơ sở y tế
- Hệ thống phục vụ cho: Bác sĩ, nhân viên và người khám bệnh.
- Bác sĩ: Cung cấp tiền sử bệnh án của bệnh nhân, lưu trữ thông tin quá trình khám chữa bệnh từ đó cung cấp thông tin để bác sĩ đưa ra phương pháp chữa bệnh hợp lý với từng bệnh nhân. Cung cấp công cụ số hóa quá trình khám chữa bệnh tại các cơ sở y tế.
- Người khám bệnh: Lưu trữ thông tin bệnh án, giúp người khám theo dõi thông tin về tiền sử bệnh án của mình từ trước đến nay. Xem thông tin các bác sĩ đã từng khám cho người khám bệnh.
- Một hồ sơ bệnh án bao gồm: Thông tin cá nhân, thông tin khám sàng lọc, thông tin xét nghiệm, thông tin khám chữa bệnh, các hồ sơ bệnh án lần trước tái khám, kết quả điều trị và tên thuốc sử dụng trong lần điều trị đó.

Các chức năng:

- Hỗ trợ quy trình khám chữa bệnh tại 1 phòng khám. Quy trình khám chữa bệnh gồm: Tiếp nhận bệnh nhân, Khám sàng lọc, Xét nghiệm (nếu có), Khám bệnh chẩn đoán bệnh và kê thuốc.

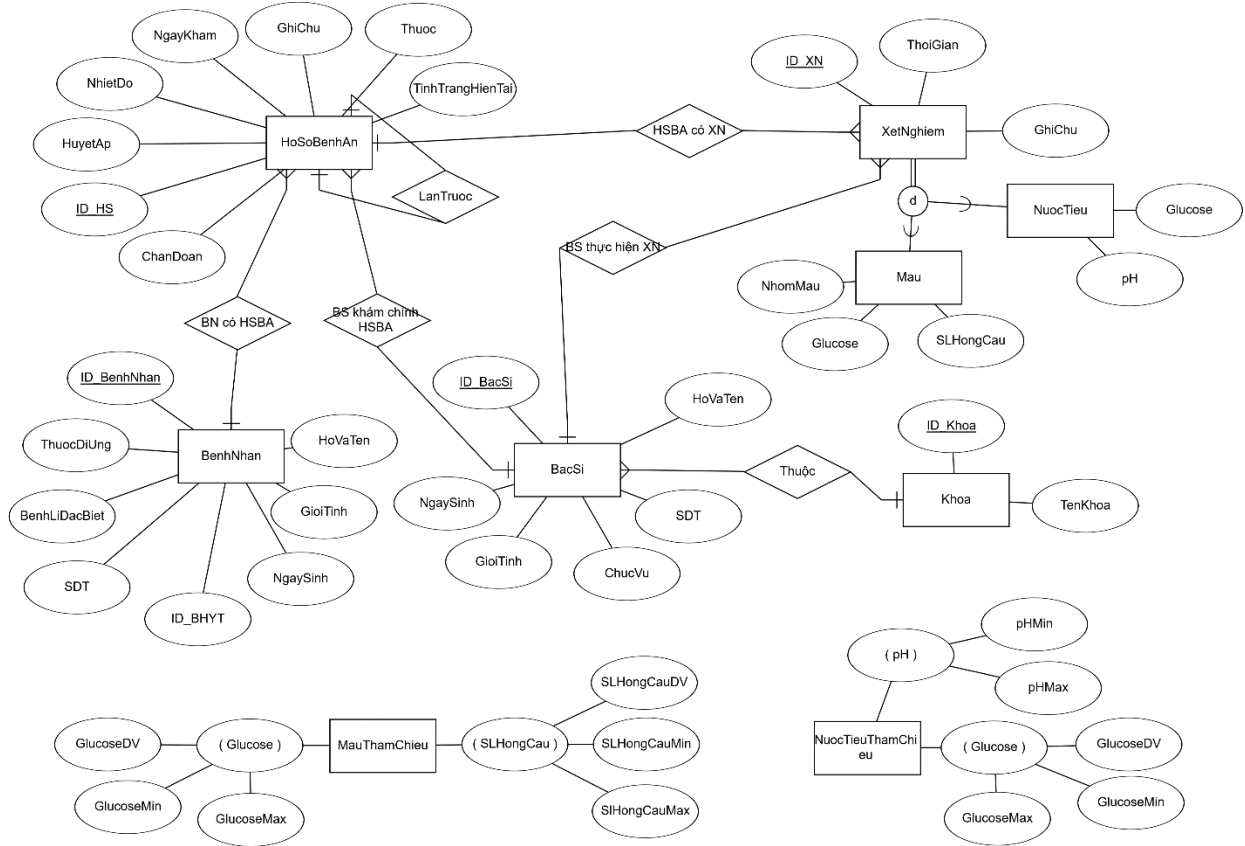
- Cung cấp cho bác sĩ dữ liệu cần thiết để phục vụ quá trình chẩn đoán bệnh.
- Cung cấp cho người bệnh quyền truy cập thông tin bệnh án hiện tại và quá khứ của mình.
- Quản lý thông tin bác sĩ và nhân viên: Lưu trữ thông tin cần thiết của bác sĩ để phục vụ quá trình liên lạc, quản lý bệnh án bệnh nhân.
- Quy trình:
 - Tiếp nhận hồ sơ: Người đến khám lần đầu hay tái khám đều gặp trực tiếp nhân viên ở quầy tiếp đón. Người khám lần đầu cần điền các thông tin cơ bản để tạo hồ sơ bệnh nhân. Tạo mới bệnh án cho lần khám này. Nhân viên tiếp tân sẽ chỉ dẫn người bệnh tới phòng khám chính để bắt đầu khám sàng lọc.
 - Khám sàng lọc: Khám sàng lọc bao gồm đo nhiệt độ, huyết áp, ... và chỉ định các xét nghiệm cần thiết để có thể đưa ra chẩn đoán. Bác sĩ đa khoa khám sàng lọc, cập nhật thông tin bệnh án, tra cứu thông tin bệnh án.
 - Xét nghiệm (Nếu cần): Do bác sĩ khám đa khoa yêu cầu để phục vụ khám bệnh. Thông tin xét nghiệm được bác sĩ xét nghiệm cập nhật vào bệnh án.
 - Khám bệnh: Bác sĩ đa khoa tiếp tục tiếp nhận bệnh nhân để tiến hành khám chi tiết dựa vào triệu chứng, kết quả sàng lọc, kết quả xét nghiệm và lịch sử y tế của bệnh nhân. Lưu kết quả khám bệnh vào bệnh án
 - Chẩn đoán bệnh: Bác sĩ dựa vào thông tin khám, xét nghiệm của bệnh nhân, kết hợp với những thông tin có sẵn trên cơ sở dữ liệu và kinh nghiệm khám chữa bệnh để đưa ra kết quả chẩn đoán cho người được khám. Bác sĩ cập nhật thông tin hồ sơ bệnh án của bệnh nhân.
 - kê thuốc: Bác sĩ đưa ra đơn thuốc phù hợp với tình trạng bệnh nhân; ghi lại tên thuốc bệnh nhân sử dụng để phục vụ việc điều trị sau này.
 - Điều trị: Bệnh nhân thực hiện điều trị theo yêu cầu của bác sĩ. Bác sĩ ghi lại kết quả điều trị của bệnh nhân qua các lần tái khám (nếu có).

Phân quyền cho người dùng hệ thống:

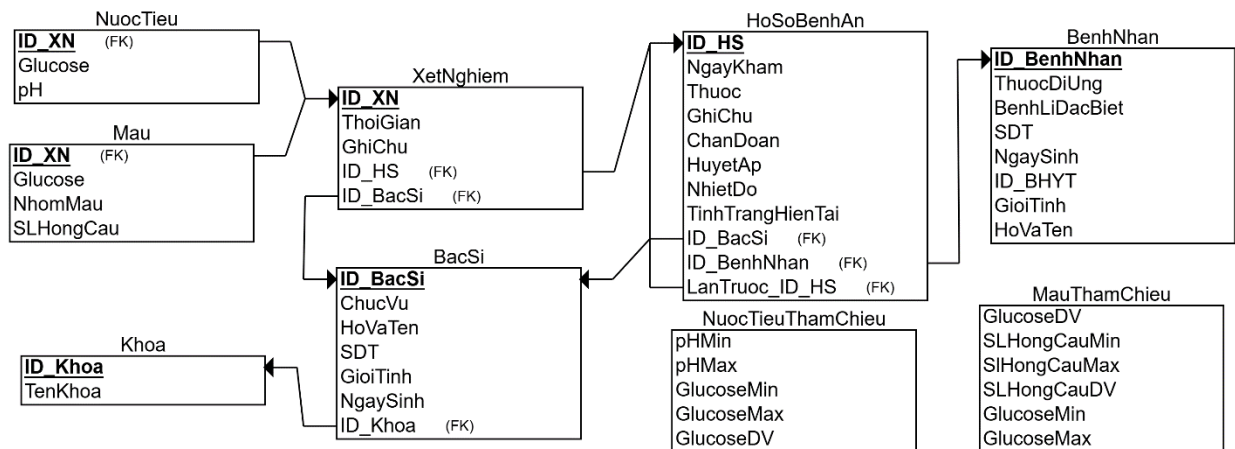
- Nhân viên tiếp tân: Tạo bản ghi bệnh án mới, cập nhật thông tin cơ bản của người đến khám.
- Bác sĩ: có quyền truy xuất thông tin các bệnh án của các bệnh nhân đang phụ trách, thêm bệnh án, sửa bệnh án (của những bệnh nhân đang chữa trị).
- Người bệnh: có quyền truy xuất thông tin y bác sĩ đang điều trị cho mình (tên, chuyên môn, khoa trực thuộc), truy xuất thông tin bệnh án của bản thân.

II. Thiết kế cơ sở dữ liệu

1. Thực thể liên kết



2. Sơ đồ quan hệ



3. Ràng buộc

- Các ràng buộc khoá chính: ID_HS, ID_BenhNhan, ID_BacSi, ID_XN, ID_Khoa
- Các ràng buộc NOT NULL:

- i. Họ và tên, Giới tính, Ngày sinh, SDT
- ii. Tên khoa
- iii. Chức vụ
- iv. Ngày khám, Huyết áp, Nhiệt độ, Tình trạng hiện tại, Thời gian XN
- v. Các giá trị của xét nghiệm máu, nước tiểu
- vi. Các giá trị của bảng tham chiếu

c. Các ràng buộc của khoá ngoài:

- i. FOREIGN KEY (ID_Khoa) REFERENCES Khoa(ID_Khoa) ON UPDATE CASCADE ON DELETE SET NULL
- ii. FOREIGN KEY (ID_BacSi) REFERENCES BacSi(ID_BacSi) ON UPDATE CASCADE ON DELETE CASCADE
- iii. FOREIGN KEY (ID_BenhNhan) REFERENCES BenhNhan(ID_BenhNhan) ON UPDATE CASCADE ON DELETE CASCADE
- iv. FOREIGN KEY (LanTruoc_ID_HS) REFERENCES HoSoBenhAn(ID_HS) ON UPDATE CASCADE ON DELETE SET NULL
- v. FOREIGN KEY (ID_HS) REFERENCES HoSoBenhAn(ID_HS) ON UPDATE CASCADE ON DELETE CASCADE

d. Ràng buộc về dữ liệu

- i. Giới tính là F hoặc M
- ii. Nhóm máu là một trong các giá trị: AB+, AB-, A+, A-, B+, B-, O+, O-
- iii. Các giá trị Min của bảng tham chiếu phải bé hơn Max

4. Index

- Index trên trường id_benhnhan ở bảng HoSoBenhAn
- Index trên trường id_hs ở bảng HoSoBenhAn
- Index trên trường id_hs ở bảng XetNghiem
- Index trên trường id_xn ở bảng Mau
- Index trên trường id_xn ở bảng NuocTieu

5. Trigger

- a. Khoá ngoài lần trước của 1 hàng của HoSoBenhAn không được chính là ID_HS của nó và phải của cùng một bệnh nhân (trước insert hoặc update)

```

30 -- Insert a previous record into HoSoBenhAn
31 INSERT INTO HoSoBenhAn (ID_BenhNhan, ID_Bacsi, NgayKham, HuyetAp, NhietDo, TinhTrangHienTai, ChanDoan, Thuoc, GhiChu)
32 VALUES
33 (1, 1, '2024-06-01', 120, 36.5, 'On dinh', ARRAY['Healthy'], ARRAY['Vitamin C'], 'First visit'),
34 (1, 2, '2024-06-10', 130, 37.0, 'Cai thien', ARRAY['Flu'], ARRAY['Paracetamol'], 'Follow-up visit'),
35 (1, 2, '2024-06-12', 135, 37.2, 'Tiep tuc cai thien', ARRAY['Cold'], ARRAY['Penicillin'], 'Second follow-up visit');
36
37 drop trigger trg_check_LanTruoc_ID_HS on hosobenhan;
38 -- Create the trigger function to enforce constraints
39 CREATE OR REPLACE FUNCTION check_LanTruoc_ID_HS()
40 RETURNS TRIGGER AS $$
41 BEGIN
42     -- Check that LanTruoc_ID_HS belongs to the same patient
43     IF NEW.LanTruoc_ID_HS IS NOT NULL THEN
44         IF NOT EXISTS (
45             SELECT 1
46             FROM HoSoBenhAn
47             WHERE ID_HS = NEW.LanTruoc_ID_HS AND ID_BenhNhan = NEW.ID_BenhNhan
48         ) THEN
49             RAISE EXCEPTION 'LanTruoc_ID_HS must belong to the same patient';
50         END IF;
51
52     -- Check that LanTruoc_ID_HS is not the same as ID_HS
53     IF NEW.LanTruoc_ID_HS = NEW.ID_HS THEN
54         RAISE EXCEPTION 'LanTruoc_ID_HS cannot be the same as ID_HS';
55     END IF;
56 END IF;
57
58     RETURN NEW;
59 END;
60 $$ LANGUAGE plpgsql;
61 CREATE TRIGGER trg_check_LanTruoc_ID_HS
62 BEFORE INSERT OR UPDATE ON HoSoBenhAn
63 FOR EACH ROW
64 EXECUTE FUNCTION check_LanTruoc_ID_HS();
65 --select * from hosobenhan
66 UPDATE HoSoBenhAn SET LanTruoc_ID_HS = 1 WHERE ID_HS = 1;

```

Data Output Messages Notifications

ERROR: LanTruoc_ID_HS cannot be the same as ID_HS
CONTEXT: PL/pgSQL function check_lantruoc_id_hs() line 15 at RAISE

SQL state: P0001

- b. Thuoc của HoSoBenhAn không được chứa phần tử trong ThuocDiUng của bệnh nhân (trước insert hoặc update)

Query	Query History
40	\$\$ LANGUAGE plpgsql;
41	
42	
43	-- Insert sample data into BenhNhan
44	INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
45	VALUES
46	('Nguyen Van A', 'M', '1990-01-01', '0912345678', '123456', ARRAY['Penicillin', 'Aspirin'], ARRAY['Diabetes']),
47	('Tran Thi B', 'F', '1985-05-05', '0987654321', '654321', NULL, ARRAY['Hypertension']);
48	
49	-- Insert sample data into Khoa
50	INSERT INTO Khoa (TenKhoa) VALUES
51	('Khoa Noi'),
52	('Khoa Ngoai');
53	
54	-- Insert sample data into BacSi
55	INSERT INTO BacSi (HoVaTen, SDT, GioiTinh, NgaySinh, ChucVu, ID_Khoa)
56	VALUES
57	('Le Van C', '0912345679', 'M', '1980-02-02', 'Bac Si Truong', 1),
58	('Pham Thi D', '0987654320', 'F', '1975-03-03', 'Bac Si Pho', 2);
59	
60	-- Insert a previous record into HoSoBenhAn
61	INSERT INTO HoSoBenhAn (ID_BenhNhan, ID_BacSi, NgayKham, HuyetAp, NhietDo, TinhTrangHienTai, ChanDoan, Thuoc, GhiChu)
62	VALUES
63	(1, 1, '2024-06-01', 120, 36.5, 'On dinh', ARRAY['Healthy'], ARRAY['Vitamin C'], 'First visit'),
64	(1, 2, '2024-06-10', 130, 37.0, 'Cai thien', ARRAY['Flu'], ARRAY['Paracetamol'], 'Follow-up visit'),
65	(1, 2, '2024-06-12', 135, 37.2, 'Tiep tuc cai thien', ARRAY['Cold'], ARRAY['Penicillin'], 'Second follow-up visit');
66	
67	-- Test the function
68	SELECT * FROM find_ketluan_by_benhnhan(1, 'Flu');
69	

Data Output	Messages	Notifications
ERROR: Ngươi bệnh bị dị ứng thuốc này: Penicillin		
CONTEXT: PL/pgSQL function check_thuocdiung() line 17 at RAISE		
SQL state: P0001		

CREATE OR REPLACE FUNCTION check_ThuocDiUng()

RETURNS TRIGGER AS \$\$

DECLARE

listThuoc TEXT[];

drug TEXT;

BEGIN

-- Get the list of allergic drugs for the patient

SELECT ThuocDiUng INTO listThuoc

FROM BenhNhan

WHERE ID_BenhNhan = NEW.ID_BenhNhan;

```

-- Loop through each drug in the Thuoc array

IF listThuoc IS NOT NULL THEN

    FOREACH drug IN ARRAY NEW.Thuoc

    LOOP

        -- Check if the drug is in the thuocDiUng array

        IF drug = ANY(listThuoc) THEN

            RAISE EXCEPTION 'Nguoi benh bi di ung thuoc nay: %', drug;

        END IF;

    END LOOP;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_check_ThuocDiUng

BEFORE INSERT OR UPDATE ON HoSoBenhAn

FOR EACH ROW

EXECUTE FUNCTION check_ThuocDiUng();

c. Trigger ngăn insert bệnh nhân mới nếu trùng cả tên và ngày sinh với dữ
liệu có sẵn. Cần set flag để force insert

```

```

15 DROP TRIGGER trg_check_duplicate ON BenhNhan;
16 CREATE OR REPLACE FUNCTION check_duplicate_benhnhhan() RETURNS TRIGGER AS $$
17 DECLARE
18     existing_benhnhhan RECORD;
19     force_insert BOOLEAN;
20 BEGIN
21     -- Lấy giá trị của biến session
22     SELECT current_setting('app.force_insert', true) INTO force_insert;
23     -- Kiểm tra bản ghi trùng lặp
24     SELECT * INTO existing_benhnhhan
25     FROM BenhNhan
26     WHERE HoVaTen = NEW.HoVaTen AND NgaySinh = NEW.NgaySinh;
27     IF FOUND THEN
28         -- Nếu tìm thấy bản ghi trùng, hiển thị thông báo
29         RAISE NOTICE 'Duplicate found: ID: %, Name: %, Birthdate: %, Phone: %, BHYT: %',
30             existing_benhnhhan.ID_BenhNhan, existing_benhnhhan.HoVaTen, existing_benhnhhan.NgaySinh,
31             existing_benhnhhan.SDT, existing_benhnhhan.ID_BHYT;
32         -- Nếu force_insert là true, cho phép insert
33         IF force_insert THEN
34             RETURN NEW;
35         ELSE
36             -- Nếu force_insert là false, chặn insert
37             RETURN NULL;
38         END IF;
39     ELSE
40         -- Nếu không có bản ghi trùng, cho phép insert
41         RETURN NEW;
42     END IF;
43 END;
44 $$ LANGUAGE plpgsql;
45 CREATE TRIGGER trg_check_duplicate
46 BEFORE INSERT ON BenhNhan
47 FOR EACH ROW
48 EXECUTE FUNCTION check_duplicate_benhnhhan();
49 -- Set the session variable to true to force the insert despite duplicates
50 --SET app.force_insert = 'true';
51 INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
52 VALUES ('John Doe', 'M', '1980-01-01', '1234567890', 'BHYT01', NULL, NULL);
53 INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
54 VALUES ('John Doe', 'M', '1980-01-01', '1234567893', 'BHYT02', NULL, NULL);

```

Data Output Messages Notifications

```

NOTICE: Duplicate found: ID: 1, Name: John Doe, Birthdate: 1980-01-01, Phone: 1234567890, BHYT: BHYT01
INSERT 0 0

```

Query returned successfully in 41 msec.


```

48 -- Set the session variable to true to force the insert despite duplicates
49 SET app.force_insert = 'true';
50 INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SĐT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
51 VALUES ('John Doe', 'M', '1980-01-01', '1234567890', 'BHYT01', NULL, NULL);
52 INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SĐT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
53 VALUES ('John Doe', 'M', '1980-01-01', '1234567893', 'BHYT02', NULL, NULL);
54 -- Reset the session variable to false to prevent insert if duplicates exist
55 --SET app.force_insert = 'false';
56 select * from benhnhan
57
58

```

Data Output
Messages
Notifications

	id_benhnhan [PK] integer	hovaten text	gioitinh character	ngaysinh date	sdt character varying (11)	id_bhyt character varying (6)	thuocdiung text[]	benhli dacbiet text[]
1	1	John Doe	M	1980-01-01	1234567890	BHYT01	[null]	[null]
2	2	John Doe	M	1980-01-01	1234567893	BHYT02	[null]	[null]

- d. Trigger ngày khám đang được insert phải xảy ra sau ngày khám cuối cùng của bệnh nhân

	id_hs [PK] integer	id_benhnhan integer	id_bacsi integer	ngaykham date	huyetap integer	nhietdo numeric	tinhtranghientai text	chandoan text[]	thuoc text[]	ghichu text	lantruoc_id_hs integer
1	1	1	1	2024-06-01	120	36.5	On dinh	{Healthy}	{Vitamin C}	First visit	[null]
2	2	1	2	2024-06-10	130	37.0	Cai thien	{Flu}	{Paracetamol}	Follow-up visit	[null]
3	3	1	2	2024-06-12	135	37.2	Tiep tuc cai thien	{Cold}	{Penicillin}	Second follow-up visit	[null]
4	4	2	2	2024-06-09	133	37.1	cai thien	{Headache}	{ABC}	Second patient	[null]

Total rows: 4 of 4 Query complete 00:00:00.074

```

30 -- Insert a previous record into HoSoBenhAn
31 INSERT INTO HoSoBenhAn (ID_BenhNhan, ID_Bacsi, NgayKham, HuyetAp, NhietDo, TinhTrangHienTai, ChanDoan, Thuoc, GhiChu)
32 VALUES
33     (1, 1, '2024-06-01', 120, 36.5, 'On dinh', ARRAY['Healthy'], ARRAY['Vitamin C'], 'First visit'),
34     (1, 2, '2024-06-10', 130, 37.0, 'Cai thien', ARRAY['Flu'], ARRAY['Paracetamol'], 'Follow-up visit'),
35     (1, 2, '2024-06-12', 135, 37.2, 'Tiep tuc cai thien', ARRAY['Cold'], ARRAY['Penicillin'], 'Second follow-up visit'),
36     (2, 2, '2024-06-09', 133, 37.1, 'cai thien', ARRAY['Headache'], ARRAY['ABC'], 'Second patient'); --id hoso = 4
37
38 drop trigger trg_check_NgayKham on hosobenhan;
39 -- Create the trigger function to enforce NgayKham constraint
40 CREATE OR REPLACE FUNCTION check_NgayKham()
41 RETURNS TRIGGER AS $$
42 BEGIN
43     -- Check if there's any existing record for the same patient
44     IF EXISTS (
45         SELECT 1
46         FROM HoSoBenhAn
47         WHERE ID_BenhNhan = NEW.ID_BenhNhan
48         AND NgayKham > NEW.NgayKham
49     ) THEN
50         RAISE EXCEPTION 'NgayKham must be equal or greater than the previous exam date for the same patient';
51     END IF;
52
53     RETURN NEW;
54 END;
55 $$ LANGUAGE plpgsql;
56 CREATE TRIGGER trg_check_NgayKham
57 BEFORE INSERT OR UPDATE ON HoSoBenhAn
58 FOR EACH ROW
59 EXECUTE FUNCTION check_NgayKham();
60 INSERT INTO HoSoBenhAn (ID_BenhNhan, ID_Bacsi, NgayKham, HuyetAp, NhietDo, TinhTrangHienTai, ChanDoan, Thuoc, GhiChu)
61 VALUES
62     (1, 2, '2024-05-27', 126, 37.3, 'On dinh', ARRAY['Healthy'], ARRAY['Vitamin B'], 'time traveling');

```

Data Output Messages Notifications

ERROR: NgayKham must be equal or greater than the previous exam date for the same patient
CONTEXT: PL/pgSQL function check_ngaykham() line 10 at RAISE

SQL state: P0001

III. Thao tác trên cơ sở dữ liệu

1. Các truy vấn

a. Liệt kê tất cả các hồ sơ và kết quả xét nghiệm của một bệnh nhân.

```
SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, xn.*, m.glucose, m.nhommau,
m.slhongcau, nt.glucose, nt.ph
```

```
FROM hosobenhan h
```

```
LEFT JOIN xetnghiem xn ON xn.id_hs = h.id_hs
```

```
LEFT JOIN mau m ON m.id_xn = xn.id_xn
```

```
LEFT JOIN nuoctieu nt ON nt.id_xn = xn.id_xn
```

WHERE h.id_benhnhan = 47

ORDER BY ngaykham DESC

	id_hs integer	id_benhnhan integer	ngaykham date	id_bacsi integer	id_xn integer	id_hs integer	id_bacsn integer	thoigian timestamp without time zone	ghichu text	glucose numeric
1	79	47	2024-02-22	33	25	79	14	2024-02-22 10:27:08	Draw whose understand mouth.	[null]
2	79	47	2024-02-22	33	180	79	31	2024-02-22 05:19:45	Arrive next sometimes national bed production.	2.86
3	79	47	2024-02-22	33	148	79	32	2024-02-22 16:13:49	Budget gun issue their material level speech several.	1.39
4	79	47	2024-02-22	33	154	79	24	2024-02-22 06:59:52	Camera wind race material.	1.27
5	68	47	2024-02-13	43	197	68	4	2024-02-13 03:33:02	Radio nature health nothing resource so style we.	0.04
6	68	47	2024-02-13	43	163	68	35	2024-02-13 09:10:12	Result study role order.	2.5
7	39	47	2024-01-23	22	[null]	[null]	[null]	[null]	[null]	[null]
8	37	47	2024-01-21	44	176	37	21	2024-01-21 22:54:42	Early suffer measure single.	1.61

Total rows: 8 of 8 Query complete 00:00:00.051 Ln 2, Col 1

b. Liệt kê các bác sĩ từng khám cho một bệnh nhân.

SELECT DISTINCT b* FROM Bacsi b

JOIN Hosobenhan h ON h.ID_Benhnhan = 47 AND h.id_bacsi = b.id_Bacsi;

	id_bacsi [PK] integer	hovaten text	sdt character	gioitinh character	ngaysinh date	chucvu text	id_khoa integer
1	22	Tammy Briggs	550.331.655	F	1998-04-10	International aid/development worker	4
2	33	Mark Brewer	(785)949-28	M	1964-07-24	Chief Financial Officer	4
3	43	Kaitlyn Navarro	+1-692-343-	F	1955-02-18	Jewellery designer	3
4	44	Veronica Koch	001-747-368	F	1979-07-19	Medical technical officer	9

Total rows: 4 of 4 Query complete 00:00:00.070

c. Liệt kê tất cả hồ sơ bệnh án của một bệnh nhân.

SELECT id_hs, id_benhnhan, id_bacsi, ngaykham

FROM HoSoBenhAn

WHERE ID_BenhNhan = 47

	id_hs [PK] integer	id_benhnhan integer	id_bacsi integer	ngaykham date
1	37	47	44	2024-01-21
2	39	47	22	2024-01-23
3	68	47	43	2024-02-13
4	79	47	33	2024-02-22
Total rows: 4 of 4 Query complete 00:00:00.064				

- d. Liệt kê các hồ sơ bệnh án trước đó có liên quan đến nhau của một bệnh nhân.

WITH RECURSIVE HoSoCTE AS (

SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, h.lantruoc_id_hs

FROM HoSoBenhAn h

WHERE h.ID_HS = 84

UNION

SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, h.lantruoc_id_hs

FROM HoSoBenhAn h

JOIN HoSoCTE c ON c.LanTruoc_ID_HS = h.ID_HS

)

SELECT * FROM HoSoCTE;

	id_hs integer	id_benhnhan integer	ngaykham date	id_bacsi integer	lantruoc_id_hs integer
1	84	96	2024-02-25	1	27
2	27	96	2024-01-14	19	17
3	17	96	2024-01-09	39	14
4	14	96	2024-01-08	6	[null]
Total rows: 4 of 4 Query complete 00:00:00.070 Ln 18, Col 1					

- e. Tìm hồ sơ bệnh nhân theo tên.

```

SELECT ID_BenhNhan, HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng,
BenhLiDacBiet

FROM BenhNhan

WHERE HoVaTen ILIKE '%' || 'David Johnson' || '%'

```

	id_benhnhân [PK] integer	hovaten text	gioitinh character	ngaysinh date	sdt character varying (11)	id_bhyt character varying (6)	thuocdiung text[]	benhli dac biet text[]
1	37	David Johnson	M	1953-05-24	(960)505-74	SJqOpd	{two,case,type}	{shoulder,prove}
Total rows: 1 of 1 Query complete 00:00:00.075 Ln 35, Col 1								

f. Liệt kê các bệnh theo số người mắc giảm dần trong một khoảng thời gian

```

SELECT unnest(ChanDoan) AS bệnh, COUNT(*) AS soluongmac

FROM HoSoBenhAn

WHERE NgayKham BETWEEN '2024-1-1' AND '2024-3-1'

AND lantruoc_id_hs IS NULL

GROUP BY bệnh

ORDER BY soluongmac DESC;

```

	bệnh text	soluongmac bigint
1	reflect	2
2	third	2
3	condition	2
4	manage	2
Total rows: 156 of 156 Query complete 00:00:00.049		

2. Các hàm

a. Insert xét nghiệm máu

```

37
38 CREATE OR REPLACE FUNCTION insert_xn_mau(
39     p_id_bacsi INT,
40     p_id_hs INT,
41     p_glucose NUMERIC,
42     p_nhommau VARCHAR(3),
43     p_slhongcau NUMERIC,
44     p_thoigian TIMESTAMP,
45     p_ghichu TEXT
46 )
47 RETURNS VOID AS $$
48 DECLARE
49     v_id_xn INT;
50 BEGIN
51     -- Insert into XetNghiem (test result)
52     INSERT INTO XetNghiem (ID_HS, ID_BacSiXN, Thoigian, GhiChu)
53     VALUES (p_id_hs, p_id_bacsi, p_thoigian, p_ghichu)
54     RETURNING ID_XN INTO v_id_xn;
55
56     -- Insert into Mau (blood test) using the captured test result ID
57     INSERT INTO Mau (ID_XN, Glucose, NhomMau, SLHongCau)
58     VALUES (v_id_xn, p_glucose, p_nhommau, p_slhongcau);
59 END;
60 $$ LANGUAGE plpgsql;
61 select insert_xn_mau(2, 1, 30, 'AB-', 7.3, '2025-06-23 19:10:25-07', NULL);
62
63 -- Test the function
64 SELECT h.id_hs, h.id_bacsi, xn.*, m.glucose, m.nhommau, m.slhongcau
65 FROM hosobenhan h
66 LEFT JOIN xetnghiem xn ON xn.id_hs = h.id_hs
67 LEFT JOIN mau m ON m.id_xn = xn.id_xn
68 LEFT JOIN nuoctieu nt ON nt.id_xn = xn.id_xn
69

```

Data Output
Messages
Notifications

	id_hs integer	id_bacsi integer	id_xn integer	id_hs integer	id_bacsixn integer	thoigian timestamp without time zone	ghichu text	glucose numeric	nhommau character varying (3)	slhongcau numeric
1	1	1	1	1	2	2025-06-23 19:10:25	[null]	30	AB-	7.3
2	2	2	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]
3	3	2	[null]	[null]	[null]	[null]	[null]	[null]	[null]	[null]

Total rows: 3 of 3
Query complete 00:00:00.091

b. Insert xét nghiệm nước tiểu

Query

Query History

```

38 CREATE OR REPLACE FUNCTION insert_xn_nuocTieu(
39     p_id_bacsi INT,
40     p_id_hs INT,
41     p_glucose NUMERIC,
42     p_ph NUMERIC,
43     p_thoigian TIMESTAMP,
44     p_ghichu TEXT
45 )
46 RETURNS VOID AS $$
47 DECLARE
48     v_id_xn INT;
49 BEGIN
50     -- Insert into XetNghiem (test result)
51     INSERT INTO XetNghiem (ID_HS, ID_BacSiXN, Thoigian, GhiChu)
52     VALUES (p_id_hs, p_id_bacsi, p_thoigian, p_ghichu)
53     RETURNING ID_XN INTO v_id_xn;
54
55     -- Insert into NuocTieu (urine test) using the captured test result ID
56     INSERT INTO NuocTieu (ID_XN, Glucose, pH)
57     VALUES (v_id_xn, p_glucose, p_ph);
58
59     -- Optionally, you can return a success message or the inserted record ID
60     --RETURN v_id_xn;
61 END;
62 $$ LANGUAGE plpgsql;
63 -- Test the function
64 SELECT insert_xn_nuocTieu(2, 1, 0.5, 7, '2024-06-22 19:10:25-07', NULL);
65 SELECT h.id_hs, h.id_bacsi, xn.*, nt.glucose, nt.ph
66 FROM hosobenhan h
67 LEFT JOIN xetnghiem xn ON xn.id_hs = h.id_hs
68 LEFT JOIN mau m ON m.id_xn = xn.id_xn
69 LEFT JOIN nuocTieu nt ON nt.id_xn = xn.id_xn
70

```

Data Output

Messages

Notifications

	id_hs integer	id_bacsi integer	id_xn integer	id_hs integer	id_bacsiXN integer	thoigian timestamp without time zone	ghichu text	glucose numeric	ph numeric
1	1	1	1	1	2	2024-06-22 19:10:25	[null]	0.5	7
2	2	2	[null]	[null]	[null]	[null]	[null]	[null]	[null]
3	3	2	[null]	[null]	[null]	[null]	[null]	[null]	[null]

c. Update hồ sơ bệnh án

```

69 $$ LANGUAGE plpgsql;
70 select * from hosobenhan

```

Data Output

Messages

Notifications

	id_hs [PK] integer	id_benhnhan integer	id_bacsi integer	ngaykham date	huyetap integer	nhietdo numeric	tinhtranghiental text	chandoan text[]	thuoc text[]	ghichu text	lantruoc_id_hs integer
1	1	1	1	2024-06-01	120	36.5	On dinh	{Healthy}	{Vitamin C}	First visit	[null]
2	2	1	2	2024-06-10	130	37.0	Cai thien	{Flu}	{Paracetamol}	Follow-up visit	[null]
3	3	1	2	2024-06-12	135	37.2	Tiep tục cai thien	{Cold}	{Penicillin}	Second follow-up visit	[null]

```

69 $$ LANGUAGE plpgsql;
70 SELECT update_hoso(1, 'New Diagnosis 1', 'New Medicine 1', 'Additional note 1');
71 SELECT update_hoso(2, 'New Diagnosis 2', 'New Medicine 2', 'Additional note 2');
72 select * from hosobenhan

```

Data Output

Messages

Notifications

	id_hs [PK] integer	id_benhnhan integer	id_bacsi integer	ngaykham date	huyetap integer	nhietdo numeric	tinhtranghiental text	chandoan text[]	thuoc text[]	ghichu text	lantruoc_id_hs integer
1	3	1	2	2024-06-12	135	37.2	Tiep tục cai thien	{Cold}	{Penicillin}	Second follow-up visit	[null]
2	1	1	1	2024-06-01	120	36.5	On dinh	{Healthy,'New Diagnosis 1'}	{Vitamin C,'New Medicine 1'}	First visit Additional note 1	[null]
3	2	1	2	2024-06-10	130	37.0	Cai thien	{Flu,'New Diagnosis 2'}	{Paracetamol,'New Medicine 2'}	Follow-up visit Additional note 2	[null]

```

CREATE OR REPLACE FUNCTION update_hoso(
    id_hoso INT,
    new_chandoan TEXT DEFAULT NULL,
    new_thuoc TEXT DEFAULT NULL,
    additional_note TEXT DEFAULT NULL
)
RETURNS VOID AS $$
BEGIN
    IF new_chandoan IS NOT NULL THEN
        UPDATE HoSoBenhAn
        SET
            ChanDoan = array_append(ChanDoan, new_chandoan)
        WHERE
            ID_HS = id_hoso;
    END IF;

    IF new_thuoc IS NOT NULL THEN
        UPDATE HoSoBenhAn
        SET
            Thuoc = array_append(Thuoc, new_thuoc)
        WHERE
            ID_HS = id_hoso;
    END IF;

    IF additional_note IS NOT NULL THEN
        UPDATE HoSoBenhAn

```


SET

GhiChu = COALESCE(GhiChu, '') || ' ' || additional_note

WHERE

ID_HS = id_hoso;

END IF;

END;

\$\$ LANGUAGE plpgsql;

d. Insert bệnh nhân mới

```
13
14 CREATE OR REPLACE FUNCTION insert_benh_nhan(
15     ho_va_ten TEXT,
16     gioi_tinh CHAR(1),
17     ngay_sinh DATE,
18     sdt VARCHAR(11),
19     id_bhyt VARCHAR(6),
20     thuoc_di_ung TEXT[],
21     benh_li_dac_biet TEXT[]
22 ) RETURNS VOID AS $$
23 BEGIN
24     INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
25     VALUES (ho_va_ten, gioi_tinh, ngay_sinh, sdt, id_bhyt, thuoc_di_ung, benh_li_dac_biet);
26 END;
27 $$ LANGUAGE plpgsql;
28
29 SELECT insert_benh_nhan('Nguyễn Văn A', 'M', '2000-01-01', '0123456789', 'BHYT01', ARRAY['Paracetamol'], ARRAY['ABC']);
30 select * from benhnhan
```

Data Output Messages Notifications

	id_benhnhan [PK] integer	hovaten text	gioitinh character	ngaysinh date	sdt character varying (11)	id_bhyt character varying (6)	thuocdiung text[]	benhlidacbiet text[]
1	1	Nguyễn Văn A	M	2000-01-01	0123456789	BHYT01	{Paracetamol}	{ABC}

e. Tìm kết luận của một bệnh nhân có chứa bệnh Flu

```

42
43 -- Insert sample data into BenhNhan
44 INSERT INTO BenhNhan (HoVaTen, GioiTinh, NgaySinh, SDT, ID_BHYT, ThuocDiUng, BenhLiDacBiet)
45 VALUES
46 ('Nguyen Van A', 'M', '1990-01-01', '0912345678', '123456', ARRAY['Aspirin'], ARRAY['Diabetes']),
47 ('Tran Thi B', 'F', '1985-05-05', '0987654321', '654321', NULL, ARRAY['Hypertension']);
48
49 -- Insert sample data into Khoa
50 INSERT INTO Khoa (TenKhoa) VALUES
51 ('Khoa Noi'),
52 ('Khoa Ngoai');
53
54 -- Insert sample data into BacSi
55 INSERT INTO BacSi (HoVaTen, SDT, GioiTinh, NgaySinh, ChucVu, ID_Khoa)
56 VALUES
57 ('Le Van C', '0912345679', 'M', '1980-02-02', 'Bac Si Truong', 1),
58 ('Pham Thi D', '0987654320', 'F', '1975-03-03', 'Bac Si Pho', 2);
59
60 -- Insert a previous record into HoSoBenhAn
61 INSERT INTO HoSoBenhAn (ID_BenhNhan, ID_BacSi, NgayKham, HuyetAp, NhietDo, TinhTrangHienTai, ChanDoan, Thuoc, GhiChu)
62 VALUES
63 (1, 1, '2024-06-01', 120, 36.5, 'On dinh', ARRAY['Healthy'], ARRAY['Vitamin C'], 'First visit'),
64 (1, 2, '2024-06-10', 130, 37.0, 'Cai thien', ARRAY['Flu'], ARRAY['Paracetamol'], 'Follow-up visit'),
65 (1, 2, '2024-06-12', 135, 37.2, 'Tiep tuc cai thien', ARRAY['Cold'], ARRAY['Penicillin'], 'Second follow-up visit');
66
67 -- Test the function
68 SELECT * FROM find_ketluan_by_benhnhan(1, 'Flu');
69

```

Data Output Messages Notifications											
	id_hs integer	id_benhnhan integer	ngaykham date	huyetap integer	nhietdo numeric	tinhtanghientai text	chandoan text[]	thuoc text[]	ghichu text	lantruoc_id_hs integer	
1	2	1	2024-06-10	130	37.0	Cai thien	{Flu}	{Paracetamol}	Follow-up visit	[null]	

CREATE OR REPLACE FUNCTION find_ketluan_by_benhnhan(p_id_benhnhan INT,
p_search_text TEXT)

RETURNS TABLE (

ID_HS INT,

ID_BenhNhan INT,

NgayKham DATE,

HuyetAp INT,

NhietDo NUMERIC,

TinhTrangHienTai TEXT,

ChanDoan TEXT[],

Thuoc TEXT[],

GhiChu TEXT,

LanTruoc_ID_HS INT

) AS \$\$

```

BEGIN
RETURN QUERY
SELECT
    h.ID_HS,
    h.ID_BenhNhan,
    h.NgayKham,
    h.HuyetAp,
    h.NhietDo,
    h.TinhTrangHienTai,
    h.ChanDoan,
    h.Thuoc,
    h.GhiChu,
    h.LanTruoc_ID_HS
FROM HoSoBenhAn h
WHERE h.ID_BenhNhan = p_id_benhnhan
    AND p_search_text = ANY(h.ChanDoan);
END;
$$ LANGUAGE plpgsql;

```

IV. Đánh giá hiệu năng các truy vấn

Thử nghiệm các truy vấn với dữ liệu lớn (10000 bản ghi hồ sơ bệnh án)

Các thông số thời gian được tính trung bình với 5 lượt chạy

1. Liệt kê tất cả các hồ sơ và kết quả xét nghiệm của một bệnh nhân.

```

SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, xn.*, m.glucose, m.nhommau,
m.slhongcau, nt.glucose, nt.ph FROM hosobenhan h

```

LEFT JOIN xetnghiem xn ON xn.id_hs = h.id_hs

LEFT JOIN mau m ON m.id_xn = xn.id_xn

LEFT JOIN nuoctieu nt ON nt.id_xn = xn.id_xn

WHERE h.id_benhnhan = --id benh nhan

ORDER BY ngaykham DESC

100 bản ghi: 56 ms

10000 bản ghi: 63 ms

	QUERY PLAN text	
1	Sort (cost=1593.23..1593.26 rows=12 width=99)	
2	Sort Key: h.ngaykham DESC	
3	-> Hash Right Join (cost=1391.46..1593.02 rows=12 width=99)	
4	Hash Cond: (m.id_xn = xn.id_xn)	
5	-> Seq Scan on mau m (cost=0.00..164.00 rows=10000 width=19)	
6	-> Hash (cost=1391.31..1391.31 rows=12 width=84)	
7	-> Hash Right Join (cost=1191.75..1391.31 rows=12 width=84)	
8	Hash Cond: (nt.id_xn = xn.id_xn)	
9	-> Seq Scan on nuoctieu nt (cost=0.00..162.00 rows=10000 width=16)	
10	-> Hash (cost=1191.60..1191.60 rows=12 width=72)	
11	-> Hash Right Join (cost=711.08..1191.60 rows=12 width=72)	
12	Hash Cond: (xn.id_hs = h.id_hs)	
13	-> Seq Scan on xetnghiem xn (cost=0.00..428.00 rows=20000 width=...)	
14	-> Hash (cost=711.00..711.00 rows=6 width=16)	
15	-> Seq Scan on hosobenhan h (cost=0.00..711.00 rows=6 width=...)	
16	Filter: (id_benhnhan = 4895)	

10000 bản ghi có index: 57 ms

index được cài đặt: hosobenhan(id_benhnhan), xetnghiem(id_hs), mau(id_xn), nuoctieu(id_xn).

	QUERY PLAN	
	text	
1	Sort (cost=105.46..105.49 rows=12 width=99)	
2	Sort Key: h.ngaykham DESC	
3	-> Nested Loop Left Join (cost=9.20..105.25 rows=12 width=99)	
4	-> Nested Loop Left Join (cost=8.92..101.28 rows=12 width=87)	
5	-> Nested Loop Left Join (cost=8.63..97.30 rows=12 width=72)	
6	-> Bitmap Heap Scan on hosobenhan h (cost=4.33..26.59 rows=6 width=16)	
7	Recheck Cond: (id_benhnhan = 4895)	
8	-> Bitmap Index Scan on idx_hosobenhan_id_benhnhan (cost=0.00..4.33 rows=6 width=...	
9	Index Cond: (id_benhnhan = 4895)	
10	-> Bitmap Heap Scan on xetnghiem xn (cost=4.30..11.77 rows=2 width=56)	
11	Recheck Cond: (id_hs = h.id_hs)	
12	-> Bitmap Index Scan on idx_xetnghiem_id_hs (cost=0.00..4.30 rows=2 width=0)	
13	Index Cond: (id_hs = h.id_hs)	
14	-> Index Scan using idx_mau_id_xn on mau m (cost=0.29..0.32 rows=1 width=19)	
15	Index Cond: (id_xn = xn.id_xn)	
16	-> Index Scan using idx_nuocieu_id_xn on nuocieu nt (cost=0.29..0.32 rows=1 width=16)	
17	Index Cond: (id_xn = xn.id_xn)	
Total rows: 17 of 17 Query complete 00:00:00.040		

2. Liệt kê tất cả hồ sơ bệnh án của một bệnh nhân.

SELECT id_hs, id_benhnhan, id_bacsi, ngaykham FROM HoSoBenhAn WHERE
ID_BenhNhan = --id bệnh nhân

100 bản ghi: 78 ms

10000 bản ghi: 80 ms

10000 bản ghi + index id_benhnhan: 75 ms

	QUERY PLAN	
	text	
1	Bitmap Heap Scan on hosobenhan (cost=4.33..26.59 rows=6 width=16)	
2	Recheck Cond: (id_benhnhan = 4895)	
3	-> Bitmap Index Scan on idx_hosobenhan_id_benhnhan (cost=0.00..4.33 rows=6 width=...	
4	Index Cond: (id_benhnhan = 4895)	
Total rows: 4 of 4 Query complete 00:00:00.040		

3. Liệt kê các bác sĩ từng khám cho một bệnh nhân.

SELECT DISTINCT b.* FROM Bacsi b

JOIN Hosobenhan h ON h.ID_Benhnhnhan = --nhap id-- AND h.id_bacsi = b.id_Bacsi;

100 bản ghi: 70 ms

10000 bản ghi: 78 ms

	QUERY PLAN	
	text	
1	Unique (cost=745.59..745.71 rows=6 width=62)	
2	-> Sort (cost=745.59..745.61 rows=6 width=62)	
3	Sort Key: b.id_bacsi, b.hovaten, b.sdt, b.gioitinh, b.ngaysinh, b.chucvu, b.id_kh...	
4	-> Hash Join (cost=34.50..745.52 rows=6 width=62)	
5	Hash Cond: (h.id_bacsi = b.id_bacsi)	
6	-> Seq Scan on hosobenhan h (cost=0.00..711.00 rows=6 width=4)	
7	Filter: (id_benhnhnhan = 4895)	
8	-> Hash (cost=22.00..22.00 rows=1000 width=62)	
9	-> Seq Scan on bacsi b (cost=0.00..22.00 rows=1000 width=62)	
Total rows: 9 of 9 Query complete 00:00:00.044		

10000 bản ghi hồ sơ, 1000 bản ghi bacsi, index hosobenhan(id_benhnhnhan), index bacsi(id_bacsi): 72 ms

	QUERY PLAN	
	text	
1	Unique (cost=58.80..58.92 rows=6 width=62)	
2	-> Sort (cost=58.80..58.81 rows=6 width=62)	
3	Sort Key: b.id_bacsi, b.hovaten, b.sdt, b.gioitinh, b.ngaysinh, b.chucvu, b.id_khoa	
4	-> Hash Join (cost=26.66..58.72 rows=6 width=62)	
5	Hash Cond: (b.id_bacsi = h.id_bacsi)	
6	-> Seq Scan on bacsi b (cost=0.00..22.00 rows=1000 width=62)	
7	-> Hash (cost=26.59..26.59 rows=6 width=4)	
8	-> Bitmap Heap Scan on hosobenhan h (cost=4.33..26.59 rows=6 width=4)	
9	Recheck Cond: (id_benhnhnhan = 2111)	
10	-> Bitmap Index Scan on idx_hosobenhan_id_benhnhnhan (cost=0.00..4.33 rows=6 width=...	
11	Index Cond: (id_benhnhnhan = 2111)	
Total rows: 11 of 11 Query complete 00:00:00.046		

10000 bản ghi hồ sơ, 6000 bản ghi bacsi, index hosobenhan(id_benhnhnhan), index bacsi(id_bacsi): 65 ms

	QUERY PLAN text	
1	Unique (cost=76.12..76.24 rows=6 width=61)	
2	-> Sort (cost=76.12..76.14 rows=6 width=61)	
3	Sort Key: b.id_bacsi, b.hovaten, b.sdt, b.gioitinh, b.ngaysinh, b.chucvu, b.id_khoa	
4	-> Nested Loop (cost=0.57..76.04 rows=6 width=61)	
5	-> Index Scan using idx_hosobenhan_id_benhnhan on hosobenhan h (cost=0.29..26.24 rows=6 width=...	
6	Index Cond: (id_benhnhan = 2111)	
7	-> Index Scan using idx_bacsi_id_bacsi on bacsi b (cost=0.28..8.30 rows=1 width=61)	
8	Index Cond: (id_bacsi = h.id_bacsi)	

4. Liệt kê các hồ sơ bệnh án trước đó có liên quan đến nhau của một bệnh nhân

WITH RECURSIVE HoSoCTE AS (

```
SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, h.lantruoc_id_hs
FROM HoSoBenhAn h
WHERE h.ID_HS = 4895
```

UNION

```
SELECT h.id_hs, h.id_benhnhan, h.ngaykham, h.id_bacsi, h.lantruoc_id_hs
FROM HoSoBenhAn h
JOIN HoSoCTE c ON c.LanTruoc_ID_HS = h.ID_HS
```

)

SELECT * FROM HoSoCTE;

100 bản ghi: 76 ms

10000 bản ghi: 89 ms

	QUERY PLAN text	
1	CTE Scan on hosocte (cost=7951.26..7953.28 rows=101 width=20)	
2	CTE hosocte	
3	-> Recursive Union (cost=0.00..7951.26 rows=101 width=20)	
4	-> Seq Scan on hosobenhan h (cost=0.00..711.00 rows=1 width=20)	
5	Filter: (id_hs = 4895)	
6	-> Hash Join (cost=0.33..723.93 rows=10 width=20)	
7	Hash Cond: (h_1.id_hs = c.lantruoc_id_hs)	
8	-> Seq Scan on hosobenhan h_1 (cost=0.00..686.00 rows=10000 wid...	
9	-> Hash (cost=0.20..0.20 rows=10 width=4)	
10	-> WorkTable Scan on hosocte c (cost=0.00..0.20 rows=10 width=...	
Total rows: 10 of 10 Query complete 00:00:00.054		

10000 bản ghi + index hosobenhan(id_hs): 51 ms

	QUERY PLAN text	
1	CTE Scan on hosocte (cost=801.58..803.60 rows=101 width=20)	
2	CTE hosocte	
3	-> Recursive Union (cost=0.29..801.58 rows=101 width=20)	
4	-> Index Scan using idx_hosobenhan_id_hs on hosobenhan h (cost=0.29..8.30 rows=1 width=20)	
5	Index Cond: (id_hs = 4895)	
6	-> Nested Loop (cost=0.29..79.23 rows=10 width=20)	
7	-> WorkTable Scan on hosocte c (cost=0.00..0.20 rows=10 width=4)	
8	-> Index Scan using idx_hosobenhan_id_hs on hosobenhan h_1 (cost=0.29..7.90 rows=1 width=...	
9	Index Cond: (id_hs = c.lantruoc_id_hs)	
Total rows: 9 of 9 Query complete 00:00:00.046		

V. Demo app

VI. Teamwork (Tiêu chí: Góp ý, Làm)

	Mô tả, báo cáo	Thiết kế	Thao tác DB	Đánh giá hiệu năng	Demo app
Nam	35	48	48	49	5
Phong	35	47	47	51	0
Thành	30	5	5	0	95

