

# REPORT

## Báo cáo về Cơ sở dữ liệu quản lý lịch hẹn và hồ sơ bệnh án Nhóm 4

Thành viên:  
Vũ Quang Hoàng Lâm  
Nguyễn Trung Sơn  
Đỗ Quốc Bảo

### 1. Giới thiệu

Chào mừng cô và các bạn đến với báo cáo về Cơ sở dữ liệu (DB) được thiết kế dành riêng cho việc quản lý lịch hẹn giữa bệnh nhân và bác sĩ, lưu trữ thông tin bệnh án và điều trị cùng với các chức năng quản lý điều trị, thông tin dịch vụ và lịch sử bệnh của bệnh nhân. Được xây dựng với mục tiêu tăng tính hiệu quả và sự tổ chức trong việc quản lý thông tin, cơ sở dữ liệu này cung cấp khả năng truy cập nhanh chóng và dễ dàng đối với thông tin y tế quan trọng, từ đó hỗ trợ việc cải thiện chất lượng dịch vụ chăm sóc sức khỏe.

### 2. Mục đích và chức năng của cơ sở dữ liệu

#### 2.1 Mục đích:

- Quản lý thông tin liên quan đến việc đặt lịch hẹn giữa bệnh nhân và bác sĩ.
- Ghi lại thông tin về thời gian khám và lưu trữ hồ sơ điều trị và bệnh án.
- Quản lý các loại điều trị, thông tin về thuốc và lịch sử bệnh của bệnh nhân.

#### 2.2 Các chức năng chính:

##### 2.2.1 Bệnh nhân

- Xem thông tin cá nhân và lịch sử bệnh án của mình.
- Sửa đổi thông tin cá nhân
- Đặt lịch hẹn với bác sĩ vào một ngày giờ cụ thể.
- Xem lịch hẹn đã đặt và theo dõi thay đổi lịch trình.

- Hủy lịch hẹn nếu cần.
- Xem thông tin về bác sĩ để lựa chọn bác sĩ phù hợp với nhu cầu khám.
- Kiểm tra xem bác sĩ có lịch hẹn nào vào ngày cụ thể hay không
- Lấy danh sách bác sĩ theo chuyên khoa

#### 2.2.2 Bác sĩ

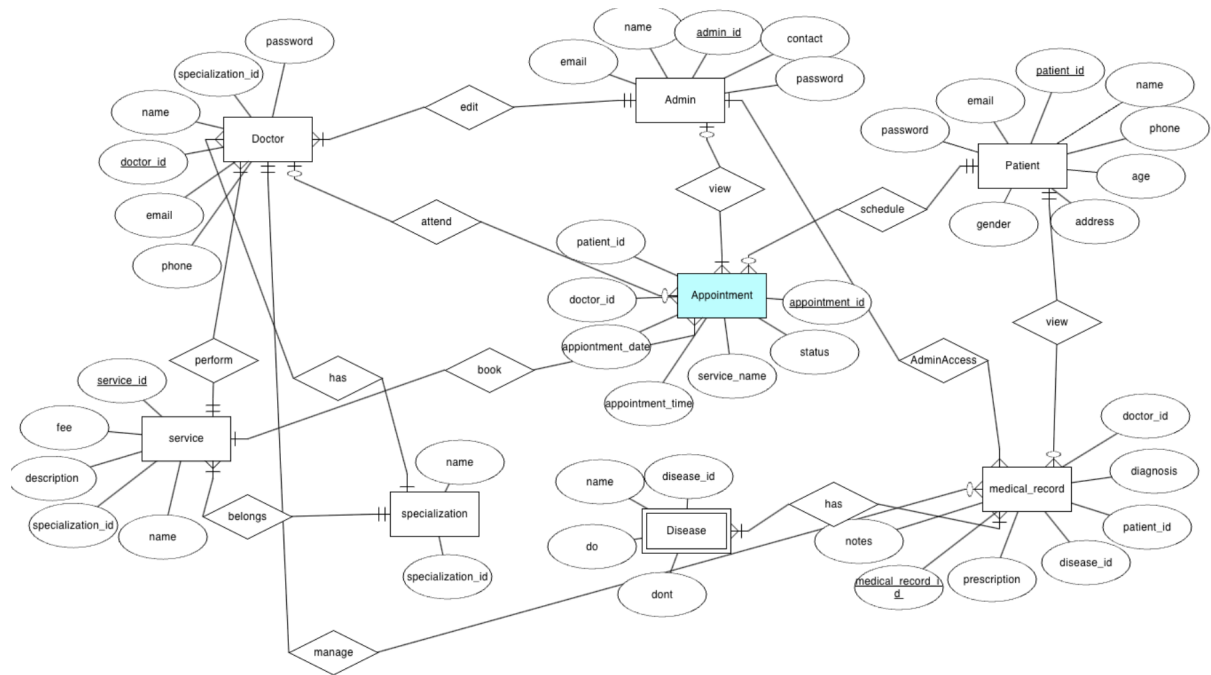
- Xem thông tin cá nhân và lịch sử cuộc hẹn của mình.
- Tìm kiếm bệnh nhân để xem thông tin bệnh án và lịch sử bệnh.
- Theo dõi sức khỏe của bệnh nhân, cập nhật thông tin tình trạng bệnh và đưa ra chỉ dẫn phù hợp.
- Thêm các chỉ thị và lời khuyên liên quan đến bệnh và điều trị.
- Tìm kiếm bệnh nhân theo tên
- Lấy danh sách bệnh nhân của một bác sĩ

#### 2.2.3 Admin

- Xác nhận các lịch khám và thực hiện các thay đổi cần thiết.
- Thêm thông tin mới cho bác sĩ.
- Có quyền chỉnh sửa thông tin cá nhân của bệnh nhân, bác sĩ và hồ sơ bệnh án dựa trên ID của từng thực thể.
- Xem và thống kê các bệnh án và lịch hẹn theo tháng và ngày

### 3. Thiết kế cơ sở dữ liệu

- Lược đồ ER (Entity-Relationship diagram) được sử dụng để mô tả mối quan hệ giữa các bảng



### 1. Bảng "patient":

- patient\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- email: Địa chỉ email của bệnh nhân, duy nhất (UNIQUE).
- password: Mật khẩu của bệnh nhân.
- name: Tên của bệnh nhân.
- age: Tuổi của bệnh nhân.
- gender: Giới tính của bệnh nhân.
- address: Địa chỉ của bệnh nhân.
- phone: Số điện thoại của bệnh nhân.

### 2. Bảng "admin":

- admin\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- name: Tên của quản trị viên.
- email: Địa chỉ email của quản trị viên, duy nhất (UNIQUE).
- password: Mật khẩu của quản trị viên, không được để trống (NOT NULL).

### 3. Bảng "specialization":

- specialization\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- name: Tên các chuyên ngành

### 4. Bảng "service":

- service\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- name: tên các dịch vụ

- specialization\_id: Khóa ngoại liên kết với bảng "specialization" (specialization\_id).
- description: Mô tả thông tin dịch vụ
- fee: Giá dịch vụ

5. Bảng "doctor":

- doctor\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- password: Mật khẩu của bác sĩ.
- name: Tên của bác sĩ.
- specialization\_id: Khóa ngoại liên kết với bảng "specialization" (specialization\_id).
- phone: Số điện thoại của bác sĩ.
- email: Địa chỉ email của bác sĩ, duy nhất (UNIQUE).

6. Bảng "disease":

- disease\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- name: Tên của bệnh.
- dos: Các hướng dẫn về cách xử lý bệnh.
- donts: Các hướng dẫn về cách không xử lý bệnh.

7. Bảng "appointment":

- appointment\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- patient\_id: Khóa ngoại liên kết với bảng "patient" (patient\_id).
- doctor\_id: Khóa ngoại liên kết với bảng "doctor" (doctor\_id).
- appointment\_date: Ngày hẹn.
- appointment\_time: Giờ hẹn.
- service\_name: Khóa ngoại liên kết với bảng "service" (name).
- status: Trạng thái của cuộc hẹn.

8. Bảng "medical\_record":

- medical\_record\_id: Trường số tự tăng, đóng vai trò là khóa chính.
- patient\_id: Khóa ngoại liên kết với bảng "patient" (patient\_id).
- doctor\_id: Khóa ngoại liên kết với bảng "doctor" (doctor\_id).
- disease\_id: Khóa ngoại liên kết với bảng "disease" (disease\_id).
- service\_id: Khóa ngoại liên kết với bảng "service" (service\_id).
- diagnosis: Chuẩn đoán của bệnh.
- prescription: Đơn thuốc.
- notes: Ghi chú.

# Các câu lệnh SQL tương ứng cho từng chức năng

## 1-Doctor :

-- xem lịch hẹn

```
CREATE OR REPLACE FUNCTION ViewDoctorAppointments(  
    p_doctor_id INT  
)  
RETURNS TABLE (  
    appointment_id INT,  
    patient_id INT,  
    appointment_date DATE,  
    appointment_time TIME  
) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT appointment.appointment_id, appointment.patient_id,  
    appointment.appointment_date, appointment.appointment_time  
    FROM appointment  
    WHERE appointment.doctor_id = p_doctor_id;  
END;  
$$ LANGUAGE plpgsql;  
  
SELECT * FROM ViewDoctorAppointments(1);
```

```
testbooking=# SELECT * FROM ViewDoctorAppointments(1);  
 appointment_id | patient_id | appointment_date | appointment_time  
-----+-----+-----+-----  
          9 |          9 | 2023-07-08      | 12:00:00  
(1 row)
```

--cap nhat trang thai

```
CREATE OR REPLACE FUNCTION change_appointment_status(app_id INT, doc_id INT)  
RETURNS VOID AS  
$$  
BEGIN
```

```

IF EXISTS (SELECT 1 FROM appointment WHERE appointment_id = app_id AND
doctor_id = doc_id) THEN
    UPDATE appointment
    SET status = 1
    WHERE appointment_id = app_id;
ELSE
    RAISE EXCEPTION 'Bác sĩ không có quyền thay đổi trạng thái của cuộc hẹn này.';
END IF;
END;
$$
LANGUAGE plpgsql;

```

```
SELECT change_appointment_status(1, 6);
```

```

testbooking=# select * from appointment;
 appointment_id | patient_id | doctor_id | appointment_date | appointment_time | status
-----+-----+-----+-----+-----+-----
          2 |         2 |         3 | 2023-07-08      | 10:30:00         |      0
          3 |         1 |         4 | 2023-07-09      | 14:00:00         |      0
          4 |         7 |         2 | 2023-07-10      | 11:30:00         |      0
          5 |         5 |         8 | 2023-07-11      | 16:30:00         |      0
          6 |         8 |         5 | 2023-07-12      | 13:00:00         |      0
          7 |         3 |         7 | 2023-07-13      | 15:30:00         |      0
          8 |         6 |         4 | 2023-07-14      | 12:00:00         |      0
          9 |         9 |         1 | 2023-07-08      | 12:00:00         |      0
         10 |        10 |         5 | 2023-07-11      | 08:00:00         |      0
          1 |         4 |         6 | 2023-07-07      | 09:00:00         |      1
(10 rows)

```

--Lay danh sach benh nhan cua mot bac si (vi du bac si co id = 1)

```

SELECT patient.*
FROM patient
INNER JOIN appointment ON patient.patient_id = appointment.patient_id
WHERE appointment.doctor_id = 1;

```

```

--tao medical_record
CREATE OR REPLACE FUNCTION CreateMedicalRecord(
    p_doctor_id INT,
    p_patient_id INT,
    p_appointment_id INT,
    p_disease_id INT,
    p_diagnosis VARCHAR(50),
    p_prescription VARCHAR(50),
    p_notes TEXT
)
RETURNS VOID AS $$
DECLARE

```

```

    p_patient_id INT;
BEGIN
    SELECT patient_id INTO p_patient_id FROM appointment WHERE appointment_id =
p_appointment_id AND doctor_id = p_doctor_id;

    IF p_patient_id IS NULL THEN
        RAISE 'No appointment found for doctor_id: %, appointment_id: %', p_doctor_id,
p_appointment_id;
    ELSE
        INSERT INTO medical_record(patient_id, doctor_id, disease_id, diagnosis,
prescription, notes)
        VALUES (p_patient_id, p_doctor_id, p_disease_id, p_diagnosis, p_prescription,
p_notes);
    END IF;

END;
$$ LANGUAGE plpgsql;

```

```

SELECT CreateMedicalRecord(1,9, 9, 1, 'Chuan doan viem hong', 'Ke don Pain relievers',
'Khuyen nghi nghi ngoi va uong nhieu nuoc');

```

medical_record_id	patient_id	doctor_id	disease_id	service_id	diagnosis	prescription	
			notes				
1	4	6	8		Chuan doan viem hong cho Do Thi Lan	Ke don Pain relievers, Antibiotic	khuyen
nghĩ nghi ngoi va uong nhieu nuoc							
2	2	3	1		Chuan doan u nang buon trung cho Le Thi Dung	Ke don Ibuprofen giam dau	Benh n
han theo doi tinh hinh qua thoi gian							
3	1	4	4		Chuan doan viem loi cho Pham Van Cuong	Ke don khang sinh	Benh n
han can cai thien ve sinh rang mieng							
4	7	2	2		Chuan doan Gay xuong cho Dang Van Minh	Ke don bo sung canxi va vien chong loang xuong	Benh n
han can han che van dong nang							
5	5	8	1		Chuan doan Tieu duong cho Bui Van Khoa	Ke don insulin hang ngay	Benh n
han can kiem tra duong huyet thuong xuyen							
6	8	5	1		Chuan doan Can thi cho Nguyen Thi Nga	Ke don kinh ap trong	Benh n
han can kiem tra thi luc thuong xuyen							
7	3	7	4		Chuan doan Viem da co dia cho Hoang Van Hoan	Ke don Kem corticosteroid	Benh n
han can han che tiep xuc voi chat kích thích va dung kem theo huong dan							
8	6	4	1		Chuan doan viem loi cho Do Thi Lan	Ke don Ke don khang sinh	Benh n
han can cai thien ve sinh rang mieng							
9	9	1	1		Chuan doan Tieu duong cho Nguyen Van An	Ke don insulin hang ngay	Benh n
han can kiem tra duong huyet thuong xuyen							
10	10	5	1		Chuan doan Can thi cho Tran Thi Bich	Ke don kinh ap trong	Benh n
han can kiem tra thi luc thuong xuyen							
11	9	1	1		Chuan doan viem hong	Ke don Pain relievers	khuyen
nghĩ nghi ngoi va uong nhieu nuoc							
(11 rows)							

```

--update medical_record
UPDATE medical_record SET
    patient_id = '10',
    doctor_id = '5',
    disease_id = '1',
    diagnosis = 'Chuan doan Can thi cho Tran Thi Bich',
    prescription = 'Ke don kinh can ',
    notes = 'Benh nhan can kiem tra thi luc thuong xuyen'
WHERE id = '1010';

```

```

--Tim kiem benh nhan theo ten
CREATE FUNCTION find_patient_by_name(search_name VARCHAR) RETURNS SETOF
patient AS $$
BEGIN
    RETURN QUERY SELECT * FROM patient WHERE name ILIKE '%' || search_name || '%';

```

```

RETURN;
END;
$$ LANGUAGE plpgsql;

```

```

SELECT * FROM find_patient_by_name('Lan');

```

```

testbooking=# SELECT * FROM find_patient_by_name('Lan');
 patient_id | email | password | name | age | gender | address | phone
-----+-----+-----+-----+-----+-----+-----+-----
          6 | dothilan@patient.com | password16 | Do Thi Lan | 61 | F | 951 Le Duan, Ha Noi | 0287391654
(1 row)

```

-nhập tên bệnh nhân và in ra diagnosis và prescription

```

CREATE OR REPLACE FUNCTION get_patient_diagnosis_prescription(search_name
VARCHAR)
RETURNS TABLE(patient_id INT, name VARCHAR, diagnosis VARCHAR, prescription
VARCHAR) AS $$
BEGIN
RETURN QUERY
SELECT patient.patient_id, patient.name, medical_record.diagnosis,
medical_record.prescription
FROM patient
INNER JOIN medical_record ON patient.patient_id = medical_record.patient_id
WHERE patient.name = search_name;
END;
$$ LANGUAGE plpgsql;

```

```

SELECT * FROM get_patient_diagnosis_prescription('Tran Thi Bich');

```

```

testbooking=# SELECT * FROM get_patient_diagnosis_prescription('Tran Thi Bich');
 patient_id | name | diagnosis | prescription
-----+-----+-----+-----
         10 | Tran Thi Bich | Chuan doan Can thi cho Tran Thi Bich | Ke don kinh ap trong
(1 row)

```

-Thông tin bệnh nhân mà bác sĩ quân li

```

CREATE FUNCTION get_doctor_patient_info(d_id INT) RETURNS TABLE (
patient_name VARCHAR(30),
age INTEGER,
gender CHAR(1)
) AS $$
BEGIN
RETURN QUERY SELECT p.name AS patient_name, p.age, p.gender
FROM patient p
JOIN medical_record m ON p.patient_id = m.patient_id
WHERE m.doctor_id = d_id;
END;
$$ LANGUAGE plpgsql;
SELECT * FROM get_doctor_patient_info(1);

```



```

booking_care=# SELECT * FROM get_doctor_patient_info(1);
 patient_name | age | gender 
-----+-----+-----
  Nguyen Van An | 23 | M
(1 row)

```

## 2.Patient :

```

INSERT INTO appointment (patient_id, doctor_id, appointment_date, appointment_time,
status, service_name)
VALUES (1, 1, '2023-07-21', '10:00:00', 0, 1);

```

```

testbooking=# select * from appointment;
 appointment_id | patient_id | doctor_id | appointment_date | appointment_time | status 
-----+-----+-----+-----+-----+-----
                2 |          2 |          3 | 2023-07-08      | 10:30:00         |      0
                3 |          1 |          4 | 2023-07-09      | 14:00:00         |      0
                4 |          7 |          2 | 2023-07-10      | 11:30:00         |      0
                5 |          5 |          8 | 2023-07-11      | 16:30:00         |      0
                6 |          8 |          5 | 2023-07-12      | 13:00:00         |      0
                7 |          3 |          7 | 2023-07-13      | 15:30:00         |      0
                8 |          6 |          4 | 2023-07-14      | 12:00:00         |      0
                9 |          9 |          1 | 2023-07-08      | 12:00:00         |      0
               10 |         10 |          5 | 2023-07-11      | 08:00:00         |      0
                1 |          4 |          6 | 2023-07-07      | 09:00:00         |      1
               11 |          1 |          1 | 2023-07-21      | 10:00:00         |      0
(11 rows)

```

--Kiem tra xem bac si co cuoc hen chua

```

CREATE OR REPLACE FUNCTION doctor_has_appointments(
  doc_id INT,
  appt_date DATE,
  appt_time TIME
) RETURNS BOOLEAN AS $$
DECLARE
  has_appointments BOOLEAN;
  one_hour_interval INTERVAL := INTERVAL '1 hour';
  appt_datetime TIMESTAMPTZ;
BEGIN
  appt_datetime := (appt_date::text || ' ' || appt_time::text)::timESTAMPTZ;
  SELECT EXISTS (
    SELECT 1
    FROM appointment a
    WHERE a.doctor_id = doc_id
    AND a.appointment_date = appt_date
    AND (a.appointment_date::text || ' ' || a.appointment_time::text)::timESTAMPTZ

```

```
BETWEEN appt_datetime AND appt_datetime + one_hour_interval
) INTO has_appointments;
```

```
RETURN NOT has_appointments;
END;
$$ LANGUAGE plpgsql;
```

```
SELECT doctor_has_appointments(1, '2023-07-08', '11:30:00');
```

```
booking_care=# SELECT doctor_has_appointments(1, '2023-07-08', '8:00:00');
 doctor_has_appointments
-----
t
(1 row)
```

--Lay danh sach bac si theo chuyen khoa

```
CREATE OR REPLACE FUNCTION get_doctors_by_specialization(specialization_name
VARCHAR(50))
RETURNS TABLE (
    doctor_id INT,
    name VARCHAR(30),
    specialization VARCHAR(50),
    phone VARCHAR(10),
    email VARCHAR(50)
)
AS $$
BEGIN
    RETURN QUERY (
        SELECT
            doctor.doctor_id,
            doctor.name,
            specialization.name AS specialization,
            doctor.phone,
            doctor.email
        FROM
            doctor
        JOIN
            specialization ON doctor.specialization_id = specialization.specialization_id
        WHERE
            specialization.name = specialization_name
    );
END;
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM get_doctors_by_specialization('Da lieu');
```

```
testbooking=# SELECT * FROM get_doctors_by_specialization('Da lieu');
 doctor_id | name       | specialization | phone       | email
-----+-----+-----+-----+-----
          7 | Ngo Thi Hoa | Da lieu       | 0321487695 | ngothihoa@doctor.com
(1 row)
```

--benh nhan xoa lich hen cua ho

```
CREATE OR REPLACE FUNCTION DeleteAppointment(
```

```
    p_appointment_id INT,
```

```
    p_patient_id INT
```

```
)
```

```
RETURNS VOID AS $$
```

```
DECLARE
```

```
    original_patient_id INT;
```

```
BEGIN
```

```
    SELECT patient_id INTO original_patient_id
```

```
    FROM appointment
```

```
    WHERE appointment_id = p_appointment_id;
```

```
    IF original_patient_id = p_patient_id THEN
```

```
        DELETE FROM appointment
```

```
        WHERE appointment_id = p_appointment_id;
```

```
    ELSE
```

```
        RAISE EXCEPTION 'You are not authorized to delete this appointment';
```

```
    END IF;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT DeleteAppointment(11, 1);
```

```
testbooking=# select * from appointment
testbooking=# ;
 appointment_id | patient_id | doctor_id | appointment_date | appointment_time | status
-----+-----+-----+-----+-----+-----
1 |          2 |          2 | 2023-07-08       | 10:30:00         | 0
2 |          3 |          1 | 2023-07-09       | 14:00:00         | 0
3 |          4 |          7 | 2023-07-10       | 11:30:00         | 0
4 |          5 |          5 | 2023-07-11       | 16:30:00         | 0
5 |          6 |          8 | 2023-07-12       | 13:00:00         | 0
6 |          7 |          3 | 2023-07-13       | 15:30:00         | 0
7 |          8 |          6 | 2023-07-14       | 12:00:00         | 0
8 |          9 |          9 | 2023-07-08       | 12:00:00         | 0
9 |         10 |         10 | 2023-07-11       | 08:00:00         | 0
10 |          1 |          4 | 2023-07-07       | 09:00:00         | 1
(10 rows)
```

--benh nhan xem lich hen

```
SELECT appointment_id, doctor_id, appointment_date, appointment_time
```

```
FROM appointment
```

```
WHERE patient_id = 1;
```

```
testbooking=# SELECT * FROM ViewPatientAppointments(1);
 appointment_id | doctor_id | appointment_date | appointment_time
-----+-----+-----+-----
              3 |          4 | 2023-07-09      | 14:00:00
(1 row)
```

--benh nhan xem thong tin bac si

```
CREATE OR REPLACE FUNCTION ViewDoctorInfo(
    p_doctor_id INT
)
```

```
RETURNS TABLE (
    doctor_id INT,
    name VARCHAR(50),
    specialization VARCHAR(50),
    phone varchar(10)
)
```

```
AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT doctor.doctor_id, doctor.name, doctor.specialization, doctor.phone
```

```
    FROM doctor
```

```
    WHERE doctor.doctor_id = p_doctor_id;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM ViewDoctorInfo(2);
```

```
testbooking=# SELECT * FROM ViewDoctorInfo(2);
 doctor_id |      name      | specialization |      phone
-----+-----+-----+-----
          2 | Tran Van Hoang | Ngoai khoa    | 0724583916
(1 row)
```

--benh nhan xem thong tin benh an

```
CREATE OR REPLACE PROCEDURE GetMedicalRecord(
```

```
    p_patient_id INT,
```

```
    INOUT ref_cursor REFCURSOR
```

```
)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
    OPEN ref_cursor FOR SELECT * FROM medical_record WHERE patient_id =
    p_patient_id;
```

```
END $$;
```

```
BEGIN;
```

```
CALL GetMedicalRecord(1, 'my_cursor');
```

```
FETCH ALL FROM "my_cursor";
```

```
COMMIT;
```

```

2 testbooking=# FETCH ALL FROM my_cursor ;
3 medical_record_id | patient_id | doctor_id | disease_id | service_id |          diagnosis          | presc
4 ription      | notes
5 -----+-----+-----+-----+-----+-----+-----
6          3 |          1 |          4 |          4 |          | Chuan doan viem loi cho Pham Van Cuong | Ke don
7 khang sinh | Benh nhan can cai thien ve sinh rang mieng
8 (1 row)

```

--view patient

```
CREATE OR REPLACE FUNCTION ViewOwnPatientInfo(
```

```
    p_patient_id INT
```

```
)
```

```
RETURNS TABLE (
```

```
    patient_id INT,
```

```
    name varchar(30),
```

```
    age integer,
```

```
    gender char(1),
```

```
    address varchar(100),
```

```
    phone varchar(10)
```

```
) AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT patient.patient_id, patient.name, patient.age, patient.gender,patient.address,
patient.phone
```

```
    FROM patient
```

```
    WHERE patient.patient_id = p_patient_id;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM ViewOwnPatientInfo(1);
```

```

testbooking=# SELECT * FROM ViewOwnPatientInfo(1);
 patient_id |      name      | age | gender |      address      |      phone
-----+-----+-----+-----+-----+-----
          1 | Pham Van Cuong |  37 |   M    | 789 Xa Dan, Ha Noi | 0739284561
1 row

```

--update patient;

```
CREATE OR REPLACE FUNCTION UpdatePatientInfo(
```

```
    p_patient_id INT,
```

```
    p_name VARCHAR(50),
```

```
    p_gender VARCHAR(1),
```

```
    p_address varchar(100),
```

```
    p_phone VARCHAR(20)
```

```
) RETURNS VOID AS $$
```

```
BEGIN
```

```
    UPDATE patient
```

```
    SET name = p_name,
```

```
        gender = p_gender,
```

```
        address = p_address,
```

```
        phone = p_phone
```

```

WHERE patient_id = p_patient_id;
END;
$$ LANGUAGE plpgsql;

```

```

SELECT UpdatePatientInfo(1, 'New Name', 'M', 'New address', 'Phone');

```

```

testbooking=# select * from patient
testbooking=# ;

```

patient_id	email	password	name	age	gender	address	phon
3915	2   lethidung@patient.com	password12	Le Thi Dung	43	F	321 Cau Giay, Ha Noi	064827
7461	3   hoangvanhoan@patient.com	password13	Hoang Van Hoan	47	M	159 Pham Hung, Ha Noi	058392
3615	4   vuthihuyen@patient.com	password14	Vu Thi Huyen	51	F	753 Giang Vo, Ha Noi	092847
7465	5   buivankhoa@patient.com	password15	Bui Van Khoa	57	M	357 Ton Duc Thang, Ha Noi	038291
1654	6   dothilan@patient.com	password16	Do Thi Lan	61	F	951 Le Duan, Ha Noi	028739
7654	7   dangvanminh@patient.com	password17	Dang Van Minh	67	M	246 Nguyen Trai, Ha Noi	018392
3645	8   nguyenthinga@patient.com	password18	Nguyen Thi Nga	71	F	842 Ba Trieu, Ha Noi	098271
7465	9   nguyenvanan@patient.com	password19	Nguyen Van An	23	M	123 Doi Can, Ha Noi	091283
1645	10   tranthibich@patient.com	password20	Tran Thi Bich	31	F	456 Truong Chinh, Ha Noi	082739
	1   phamvancuong@patient.com	password11	New Name	37	M	New address	Phone

```

(10 rows)

```

### 3.Admin:

--1. Fuction giúp bệnh nhân lựa chọn bác sĩ và dịch vụ mong muốn

```

CREATE OR REPLACE FUNCTION search_services(keyword VARCHAR)
RETURNS TABLE (service_name VARCHAR, doctor_id INT, doctor_name
VARCHAR) AS

```

```

$$

```

```

BEGIN

```

```

    RETURN QUERY

```

```

        SELECT s.name AS service_name, d.doctor_id, d.name AS doctor_name
        FROM service s
        JOIN doctor d ON s.specialization_id = d.specialization_id
        WHERE s.name ILIKE '%' || keyword || '%';

```

```

END;

```

```

$$

```

```

LANGUAGE plpgsql;

```

```

SELECT * FROM search_services('rang');

```

```
final_db=# SELECT * FROM search_services('rang');
      service_name      | doctor_id | doctor_name
-----+-----+-----
Phau thuat rang ham mat |         4 | Vu Minh Tuan
Kham rang ham mat       |         4 | Vu Minh Tuan
Dieu tri rang ham mat   |         4 | Vu Minh Tuan
(3 rows)
```

```
--2.trigger kiểm tra về thời gian đầu vào và sự xung đột cuộc hẹn có thể xảy ra
CREATE OR REPLACE FUNCTION check_appointment_conflict()
RETURNS TRIGGER AS
$$
DECLARE
    existing_appointment RECORD;
BEGIN
    -- Kiểm tra xung đột thời gian với các cuộc hẹn trước đó chỉ khi status được cập
    nhật lên 1 (đồng ý)
    IF NEW.status = 1 THEN
        -- Kiểm tra xem có cuộc hẹn nào khác cùng thời gian và bác sĩ không
        SELECT INTO existing_appointment
            appointment_id
        FROM appointment
        WHERE doctor_id = NEW.doctor_id
        AND appointment_date = NEW.appointment_date
        AND appointment_time = NEW.appointment_time
        AND status = 1
        AND appointment_id <> NEW.appointment_id; -- Loại trừ cuộc hẹn đang
        được cập nhật

        -- Nếu tìm thấy cuộc hẹn khác trùng thời gian và bác sĩ
        IF FOUND THEN
            RAISE EXCEPTION 'Xung đột cuộc hẹn với cuộc hẹn khác trước đó';
        END IF;
    END IF;

    -- Kiểm tra thời gian hẹn
    IF NEW.appointment_time < '08:00:00' OR NEW.appointment_time > '16:00:00'
    THEN
        RAISE EXCEPTION 'Thời gian hẹn phải nằm trong khoảng từ 8h đến 16h';
    END IF;

    RETURN NEW;
END;
```

\$\$

LANGUAGE plpgsql;

```
CREATE TRIGGER appointment_check_conflict_time
BEFORE INSERT OR UPDATE ON appointment
FOR EACH ROW
EXECUTE FUNCTION check_appointment_conflict();
```

-- INSERT dữ liệu mới

```
INSERT INTO appointment (patient_id, doctor_id, appointment_date,
appointment_time, service_name)
VALUES (1, 4, '2023-07-20', '10:00', 'Phau thuat tim');
```

```
INSERT INTO appointment (patient_id, doctor_id, appointment_date,
appointment_time, service_name, status)
VALUES (1, 4, '2023-06-03', '10:00', 'Phau thuat rang ham mat', 0);
```

-- UPDATE dữ liệu hiện có

```
UPDATE appointment
SET appointment_time = '06:30'
WHERE appointment_id = 1;
```

```
UPDATE appointment
SET status = 1
WHERE appointment_id = 26;
```

```
final_db=# INSERT INTO appointment (patient_id, doctor_id, appointment_date, appointment_time, service_name)
VALUES (5, 9, '2023-07-20', '10:00', 'Phau thuat ngoai khoa');
INSERT 0 1
final_db=# INSERT INTO appointment (patient_id, doctor_id, appointment_date, appointment_time, service_name)
VALUES (5, 9, '2023-07-20', '03:00', 'Phau thuat ngoai khoa');
ERROR:  Thời gian hẹn phải nằm trong khoảng từ 8h đến 16h
CONTEXT:  PL/pgSQL function check_appointment_conflict() line 25 at RAISE
```

/\*#####ADMIN#####\*/

--3.Xoá tất cả các lệnh hẹn thừa

```
DELETE FROM appointment
WHERE status = 0 AND appointment_date < CURRENT_DATE;
```

--4.Xem thông tin bệnh án theo ngày

```
CREATE OR REPLACE FUNCTION
viewMedicalRecordsByDateAndDoctor(appointmentDate DATE) RETURNS TABLE (
    patient_name VARCHAR(30),
    doctor_name VARCHAR(30),
    disease_name VARCHAR(50),
```



```

        diagnosis VARCHAR(50),
        prescription VARCHAR(50),
        notes VARCHAR(255)
    ) AS $$
BEGIN
    RETURN QUERY
    SELECT p.name AS patient_name, d.name AS doctor_name, di.name AS
disease_name,
        mr.diagnosis, mr.prescription, mr.notes
    FROM medical_record mr
    JOIN patient p ON p.patient_id = mr.patient_id
    JOIN doctor d ON d.doctor_id = mr.doctor_id
    JOIN disease di ON di.disease_id = mr.disease_id
    JOIN appointment a ON a.patient_id = p.patient_id AND a.doctor_id =
d.doctor_id
    WHERE a.appointment_date = appointmentDate;
END;
$$ LANGUAGE plpgsql;

```

```
SELECT * FROM viewMedicalRecordsByDateAndDoctor('2023-07-09');
```

```

final_db=# SELECT * FROM get_medical_records_by_date('2023-07-20');
 patient_name | doctor_name | disease_name |          diagnosis          | prescription
-----
|-----+-----+-----+-----+-----
|          notes          |
|-----+-----+-----+-----+-----
Vu Thi Huyen | Bui Trung Duc | Viem hong | Chuan doan viem hong cho Do Thi Lan | Ke don Pain relievers, Antibiotic73
| khuyen nghi nghi ngoi va uong nhieu nuoc
Le Thi Dung | Hoang Bao Anh | Tieu duong | Chuan doan u nang buon trung cho Le Thi Dung | Ke don Ibuprofen giam dau
| Benh nhan theo doi tinh hinh qua thoi gian
Pham Van Cuong | Vu Minh Tuan | Ung thu da day | Chuan doan viem loi cho Pham Van Cuong | Ke don khang sinh
| Benh nhan can cai thien ve sinh rang mieng
Dang Van Minh | Tran Van Hoang | Gay xuong | Chuan doan Gay xuong cho Dang Van Minh | Ke don bo sung canxi va vien chong loang xuong
| Benh nhan can han che van dong nang
Bui Van Khoa | Duong Quang Dong | Tieu duong | Chuan doan Tieu duong cho Bui Van Khoa | Ke don insulin hang ngay
| Benh nhan cankiem tra duong huyet thuong xuyen
Nguyen Thi Nga | Dang Thi Huong | Tieu duong | Chuan doan Can thi cho Nguyen Thi Nga | Ke don kinh ap trong
| Benh nhan cankiem tra thi luc thuong xuyen
Hoang Van Hoan | Ngo Thi Hoa | Ung thu da day | Chuan doan Viem da co dia cho Hoang Van Hoan | Ke don Kem corticosteroid
| Benh nhan can han che tiep xuc voi chat kich thich va dung kem theo huong dan cua bac si
Do Thi Lan | Vu Minh Tuan | Tieu duong | Chuan doan viem loi cho Do Thi Lan | Ke don Ke don khang sinh
| Benh nhan can cai thien ve sinh rang mieng
Nguyen Van An | Nguyen Thi Lan | Tieu duong | Chuan doan Tieu duong cho Nguyen Van An | Ke don insulin hang ngay

```

--5. xem số lượng bệnh án theo ngày

```
CREATE OR REPLACE FUNCTION get_date_with_most_appointments(month INT,
year INT)
```

```

    RETURNS TABLE (date_with_most_appointments DATE, appointment_count
BIGINT) AS $$
BEGIN

```

```
    RETURN QUERY
```

```

    SELECT appointment_date, COUNT(*)::BIGINT AS appointment_count
    FROM appointment

```

```

    WHERE EXTRACT(MONTH FROM appointment_date) = month

```

```

    AND EXTRACT(YEAR FROM appointment_date) = year

```

```

    GROUP BY appointment_date

```

```

ORDER BY COUNT(*) DESC
LIMIT 10;
END;
$$ LANGUAGE plpgsql;

```

```
SELECT * FROM get_date_with_most_appointments(7, 2023);
```

```

final_db=# SELECT * FROM get_date_with_most_appointments(7, 2023);
 date_with_most_appointments | appointment_count
-----+-----
 2023-07-20                  |                  11
 2023-07-16                  |                   1
 2023-07-30                  |                   1
(3 rows)

```

--6.in ra dịch vụ có số lượng được dùng nhiều nhất tháng 7 ---

```

SELECT s.name AS service_name, COUNT(*) AS booking_count
FROM appointment a
JOIN service s ON a.service_name = s.name
WHERE EXTRACT(MONTH FROM a.appointment_date) = 7 -- Replace with the
desired month
GROUP BY s.name
ORDER BY booking_count DESC
LIMIT 3;

```

```

final_db=# LIMIT 3;
 service_name | booking_count
-----+-----
  Dieu tri ngoai khoa |          3
  Kham mat       |          2
  Kham rang ham mat |          2
(3 rows)

```

--Tính tiền phải trả của 1 bệnh nhân theo ngày

```

CREATE OR REPLACE FUNCTION calculate_patient_total_fee(p_id INT, appoint_date
DATE)
RETURNS INT AS $$
DECLARE
    total_fee INT;
BEGIN
    SELECT COALESCE(SUM(s.fee), 0)
    INTO total_fee
    FROM appointment a

```

```

LEFT JOIN service s ON a.service_name = s.name
WHERE a.patient_id = p_id
      AND a.appointment_date = appoint_date;

RETURN total_fee;
END;
$$ LANGUAGE plpgsql;

SELECT calculate_patient_total_fee(4, '2023-07-07');

```

```

booking_care=# COMMIT;
booking_care=# CREATE OR REPLACE FUNCTION calculate_patient_total_fee(p_id INT, appoint_date DATE)
booking_care=# RETURNS INT AS $$
booking_care=# DECLARE
booking_care=#     total_fee INT;
booking_care=# BEGIN
booking_care=#     SELECT COALESCE(SUM(s.fee), 0)
booking_care=#     INTO total_fee
booking_care=#     FROM appointment a
booking_care=#     LEFT JOIN service s ON a.service_name = s.name
booking_care=#     WHERE a.patient_id = p_id
booking_care=#           AND a.appointment_date = appoint_date;
booking_care=# RETURN total_fee;
booking_care=# END;
booking_care=# $$ LANGUAGE plpgsql;
booking_care=# SELECT calculate_patient_total_fee(4, '2023-07-07');
calculate_patient_total_fee
-----
1000000
(1 row)

```

-----TRIGGER-----

--7.Tìm kiếm thông tin bác sĩ theo chuyên khoa

```

CREATE OR REPLACE FUNCTION get_doctors_by_specialization(spec
VARCHAR)
RETURNS TABLE (doctor_id INT, name VARCHAR, specialization VARCHAR,
phone VARCHAR, email VARCHAR)
AS $$
BEGIN
    RETURN QUERY SELECT d.doctor_id, d.name, s.name AS specialization,
d.phone, d.email
FROM doctor d
JOIN specialization s ON d.specialization_id = s.specialization_id
WHERE s.name ILIKE '%' || spec || '%';
END;
$$ LANGUAGE plpgsql;

```

```

SELECT *
FROM get_doctors_by_specialization('Noi');

```

```

final_db=# SELECT * FROM get_doctors_by_specialization('Noi');

```

doctor_id	name	specialization	phone	email
1	Nguyen Thi Lan	Noi khoa	0918234760	nguyenthilan@doctor.com
8	Duong Quang Dong	Noi tiet	0987152643	duongquangdong@doctor.com
9	Le Thi Nhung	Noi khoa	0631928745	lethinhung@doctor.com

(3 rows)

-----8.tạo view xem appointment-----

```
CREATE VIEW appointment_info AS
SELECT a.appointment_id, p.name AS patient_name, d.name AS doctor_name,
a.appointment_date, a.appointment_time, a.status
FROM appointment a
JOIN patient p ON a.patient_id = p.patient_id
JOIN doctor d ON a.doctor_id = d.doctor_id;
```

```
SELECT *
FROM appointment_info
ORDER BY appointment_date, appointment_time;
```

```
final_db=# ORDER BY appointment_date, appointment_time;
 appointment_id | patient_name | doctor_name | appointment_date | appointment_time | status
-----+-----+-----+-----+-----+-----
      17 | Pham Van Cuong | Tran Van Hoang | 2023-07-16 | 09:00:00 |
      35 | Bui Van Khoa | Le Thi Nhung | 2023-07-20 | 10:00:00 |
      55 | Vu Thi Huyen | Bui Trung Duc | 2023-07-20 | 10:00:00 | 0
      63 | Nguyen Van An | Nguyen Thi Lan | 2023-07-20 | 11:00:00 | 0
      56 | Le Thi Dung | Hoang Bao Anh | 2023-07-20 | 11:00:00 | 0
      57 | Pham Van Cuong | Vu Minh Tuan | 2023-07-20 | 12:00:00 | 0
      62 | Do Thi Lan | Vu Minh Tuan | 2023-07-20 | 12:00:00 | 0
      58 | Dang Van Minh | Tran Van Hoang | 2023-07-20 | 13:00:00 | 0
      61 | Hoang Van Hoan | Ngo Thi Hoa | 2023-07-20 | 13:00:00 | 0
      59 | Bui Van Khoa | Duong Quang Dong | 2023-07-20 | 14:00:00 | 0
      60 | Nguyen Thi Nga | Dang Thi Huong | 2023-07-20 | 15:00:00 | 0
      64 | Tran Thi Bich | Dang Thi Huong | 2023-07-20 | 15:30:00 | 0
      26 | Pham Van Cuong | Tran Van Hoang | 2023-07-30 | 09:00:00 | 1
      27 | Pham Van Cuong | Vu Minh Tuan | 2023-08-03 | 10:00:00 | 0
```

(14 rows)

--9+10.Cập nhật dữ liệu trên view xem số lượng appiointment sau khi có sự thay đổi

```
CREATE VIEW doctors_with_appointments AS
SELECT d.doctor_id, d.name AS doctor_name, COUNT(a.appointment_id) AS
appointment_count
FROM doctor d
JOIN appointment a ON d.doctor_id = a.doctor_id
GROUP BY d.doctor_id, d.name;
select * from doctors_with_appointments;
```

```
[final_db=# select * from doctors_with_appointments;
```

doctor_id	doctor_name	appointment_count
9	Le Thi Nhung	1
3	Hoang Bao Anh	1
5	Dang Thi Huong	2
4	Vu Minh Tuan	3
6	Bui Trung Duc	1
2	Tran Van Hoang	3
7	Ngo Thi Hoa	1
1	Nguyen Thi Lan	1
8	Duong Quang Dong	1

```
(9 rows)
```

```
-- Tạo trigger trên view doctors_with_appointments khi có xóa hoặc thêm  
appointment mới
```

```
CREATE OR REPLACE FUNCTION update_doctors_with_appointments_insert()
```

```
RETURNS TRIGGER AS
```

```
$$
```

```
BEGIN
```

```
UPDATE doctors_with_appointments
```

```
SET appointment_count = appointment_count + 1
```

```
WHERE doctor_id = NEW.doctor_id;
```

```
RETURN NULL;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_update_doctors_with_appointments_insert
```

```
AFTER INSERT ON appointment
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION update_doctors_with_appointments_insert();
```

```
CREATE OR REPLACE FUNCTION update_doctors_with_appointments_delete()
```

```
RETURNS TRIGGER AS
```

```
$$
```

```
BEGIN
```

```
UPDATE doctors_with_appointments
```

```
SET appointment_count = appointment_count - 1
```

```
WHERE doctor_id = OLD.doctor_id;
```

```
RETURN NULL;
```

```
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_update_doctors_with_appointments_delete
AFTER DELETE ON appointment
FOR EACH ROW
EXECUTE FUNCTION update_doctors_with_appointments_delete();
```

```
[final_db=# DELETE FROM appointment WHERE appointment_id = 63;
DELETE 1
```

```
[final_db=# select * from doctors_with_appointments;
```

doctor_id	doctor_name	appointment_count
9	Le Thi Nhung	1
3	Hoang Bao Anh	1
5	Dang Thi Huong	2
4	Vu Minh Tuan	3
6	Bui Trung Duc	1
2	Tran Van Hoang	3
7	Ngo Thi Hoa	1
8	Duong Quang Dong	1

```
(8 rows)
```

```
[final_db=# DELETE FROM appointment WHERE appointment_id = 57;
DELETE 1
```

```
[final_db=# select * from doctors_with_appointments;
```

doctor_id	doctor_name	appointment_count
9	Le Thi Nhung	1
3	Hoang Bao Anh	1
5	Dang Thi Huong	2
4	Vu Minh Tuan	2
6	Bui Trung Duc	1
2	Tran Van Hoang	3
7	Ngo Thi Hoa	1
8	Duong Quang Dong	1

```
(8 rows)
```

```
CREATE OR REPLACE FUNCTION instead_of_insert_doctors_with_appointments()
RETURNS TRIGGER AS
```

```
$$
```

```
BEGIN
```

```
INSERT INTO appointment (appointment_id, patient_id, doctor_id,
appointment_date, appointment_time, service_name)
```

```
VALUES (NEW.appointment_id, NEW.patient_id, NEW.doctor_id,
NEW.appointment_date, NEW.appointment_time, NEW.service_name);
```

```
UPDATE doctors_with_appointments
SET appointment_count = appointment_count + 1
WHERE doctor_id = NEW.doctor_id;
```

```
RETURN NULL;
END;
$$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_instead_of_insert_doctors_with_appointments
INSTEAD OF INSERT ON doctors_with_appointments
FOR EACH ROW
EXECUTE FUNCTION instead_of_insert_doctors_with_appointments();
```

```
SELECT * FROM doctors_with_appointments;
```

```
DELETE FROM appointment
WHERE appointment_id = 9;
```

```
final_db=# INSERT INTO appointment (patient_id, doctor_id, appointment_date, appointment_time, service_name)
[final_db=# VALUES (1, 4, '2023-07-22', '11:00', 'Dieu tri ngoai khoa');
INSERT 0 1
final_db=# INSERT INTO appointment (patient_id, doctor_id, appointment_date, appointment_time, service_name)
[final_db=# VALUES (1, 4, '2023-07-24', '11:00', 'Dieu tri ngoai khoa');
INSERT 0 1
[final_db=# select * from doctors_with_appointments;
```

doctor_id	doctor_name	appointment_count
9	Le Thi Nhung	1
3	Hoang Bao Anh	1
5	Dang Thi Huong	2
4	Vu Minh Tuan	4
6	Bui Trung Duc	1
2	Tran Van Hoang	3
7	Ngo Thi Hoa	1
8	Duong Quang Dong	1

```
(8 rows)
```

## 4.Nhiệm vụ của từng thành viên

Vũ Quang Hoàng Lâm:sản phẩm web + view + function+trigger(it)+slide+relational diagrams

Đỗ Quốc Bảo:ER+bảng sql +trigger(phần lớn)+function+mô tả nghiệp vụ

Nguyễn Trung Sơn:function+insert data+trigger(it)+tìm kiếm thông tin+sửa lỗi data