

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**



---

**DATABASE SYSTEM**  
**ASSIGNMENT #2: HOSPITAL DATABASE**

---

**Giảng viên hướng dẫn: Phan Trọng Nhân**  
**Lớp: L02**

**Sinh viên thực hiện:**

- |                      |                      |
|----------------------|----------------------|
| • Nguyễn Hoàng Thịnh | <b>MSSV: 1814172</b> |
| • Nguyễn Phi Thông   | <b>MSSV: 1814205</b> |
| • Nguyễn Văn Thuận   | <b>MSSV: 1814220</b> |
| • Nguyễn Văn Tĩnh    | <b>MSSV: 1814356</b> |

Thành viên	Công việc
Nguyễn Hoàng Thịnh	Tạo bảng dữ liệu, Insert dữ liệu, Định nghĩa các constraint, Viết các trigger kiểm tra các semantic constraint, Viết các phép check kiểm tra constraint, Viết báo cáo
Nguyễn Văn Thuận	Tạo bảng dữ liệu, Định nghĩa các constraint,Viết các update trigger, Viết các function và procedure ở part B, hiện thực web app (fullstack)
Nguyễn Phi Thông	Hiện thực web app(back-end)
Nguyễn Văn Tĩnh	Hiện thực web app(front-end)

## Contents

<b>PART A: PHYSICAL DATABASE DESIGN</b>	<b>4</b>
1. EMPLOYEE	4
2. DOCTOR và NURSE	8
3. DEPARTMENT	9
4. INPATIENT	11
6. TREAT	17
7. MEDICATION	19
8. PRESCRIPT	21
9. EXAMINATION	22
10. EFFECT	24
11. PHONE:	25
<b>PART B: STORE PROCEDURE, FUNCTION, SQL</b>	<b>26</b>
a) Increase Inpatient Fee to 10% for all the inpatients who are admitted to hospital from 01/09/2020.	26
b) Select all the patients (outpatients and inpatients) of a named doctor.	27
c) Write a function to calculate the total medication price a patient has to pay for each treatment or examination.	28
d) Write a procedure to sort the doctor in increasing number of patients he/she takes care in a period of time.	29
<b>PART C: BUILDING APPLICATION</b>	<b>30</b>
I. Tạo một user với quyền DBA:	30
II. Các chức năng được yêu cầu:	32
1. Tìm thông tin bệnh nhân:	32
2. Thêm thông tin cho bệnh nhân:	33
3. Tìm kiếm các bệnh nhân được điều trị bởi một bác sĩ:	33
4. Tạo báo cáo về một lần khám, chữa trị của một bệnh nhân:	33



- Constraint về toàn vẹn dữ liệu
  - Khóa chính của bảng này là ECode có kiểu dữ liệu là INT
  - Với trường Gender thì có kiểu dữ liệu CHAR(1) vì nhóm chỉ sử dụng 2 ký tự là M (nam) và F (nữ) để biểu thị giới tính và bị bắt buộc chỉ được nhập vào 1 trong 2 dữ liệu là 'F' hoặc 'M' qua constraint EMPLOYEE\_GENDER\_VALIDITY
  - Với các trường BirthDate và StartDate thì nhóm sử dụng kiểu dữ liệu DATE vì 2 trường này là 2 trường biểu thị ngày tháng năm
  - FName và LName nhóm sử dụng kiểu dữ liệu VARCHAR(15) và ràng buộc điều kiện NOT NULL cho 2 trường này vì 2 trường này biểu thị họ tên của nhân viên nên bắt buộc không được để trống
  - Hai trường Deg\_Name và Address có kiểu dữ liệu là VARCHAR(30) với độ dài dữ liệu tối đa được nhóm ước lượng
  - Trường Deg\_Year nhóm sử dụng kiểu dữ liệu SMALLINT
  - Trường DepartmentCode là khóa ngoại tham chiếu đến departmentcode của bảng EMPLOYEE

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
<sup>1</sup> CHECK NEGATIVE ECODE	Check	ecode>0
<sup>2</sup> CHECK NEGATIVE YEAR	Check	deg year > 1900
<sup>3</sup> CHECK VALID YEAR	Check	startdate > birthDate and deg year <= extract(year from startdate)
<sup>4</sup> DEG NAME NO DIGIT	Check	not regexp like(deg name,'[0-9]')
<sup>5</sup> EMPLOYEE GENDER VALIDITY	Check	REGEXP LIKE(qender,'(M F)')
<sup>6</sup> FK EMP DEPT	Foreign Key	(null)
<sup>7</sup> FNAME NO DIGIT CHECK	Check	not regexp like(fname,'[0-9]')
<sup>8</sup> LNAME NO DIGIT CHEKC	Check	not regexp like(lname,'[0-9]')
<sup>9</sup> SYS C007260	Check	"FNAME" IS NOT NULL
<sup>10</sup> SYS C007261	Check	"LNAME" IS NOT NULL
<sup>11</sup> SYS C007262	Primary Key	(null)

#### Các constrain của bảng employee

- Constraint về toàn vẹn tham chiếu:
  - Constraint cho khóa ngoại là ON DELETE SET NULL để khi xóa phòng ban thì tại trường DepartmentCode sẽ set giá trị NULL và thông tin về nhân viên sẽ không bị xóa
- Constraint về ngữ nghĩa:
  - Các constraint trên được thêm vào bảng EMPLOYEE để kiểm tra thông tin khi nhập vào có hợp lệ hay không. Constraint CHECK\_NEGATIVE\_ECODE dùng để kiểm tra mã nhân viên nhập vào phải có giá trị lớn hơn 0, constraint CHECK\_NEGATIVE\_YEAR dùng để kiểm tra thông tin năm của bằng có hợp lệ đúng với thực tế hay không (một phần cũng để đảm bảo rằng người dùng sẽ nhập vào một số có 4 chữ số, do trường YEAR được định nghĩa ở kiểu NUMBER và có giá trị thuộc khoảng 1900<YEAR<Năm hiện tại),

- Constraint CHECK\_VALID\_YEAR để kiểm tra thông tin thời gian bắt đầu làm việc phải lớn hơn ngày tháng năm sinh và năm của bằng phải nhỏ hơn năm hiện tại (có nghĩa những người chưa tốt nghiệp một loại bằng bác sĩ nào đó sẽ không được làm trong bệnh viện, đồng thời set cận trên cho năm tốt nghiệp hợp lệ).

- Tên của người dùng sẽ không được chứa chữ số được ràng buộc thông qua 2 constraint FNAME\_NO\_DIGIT\_CHECK và LNAME\_NO\_DIGIT\_CHECK

- Các Triggers:

```
trigger deg_year_validity_check
before insert or update of deg_year
on employee
for each row
begin
    if TO_CHAR(extract(year from sysdate)) < TO_CHAR(:new.deg_year)
    then raise_application_error(-20003,'The Graduation year of the degree is not valid. Pls
make sure you have already graduated before applying');
    end if;
end;
```

- Trigger này dùng để báo lỗi khi người dùng nhập vào một năm tốt nghiệp lớn hơn năm hiện tại

```
trigger employee_valid_birthdate
before insert or update of birthdate
on employee
for each row
begin
    if (extract(year from :new.birthdate) > extract(year from sysdate))
    then raise_application_error(-20006,'An employee cannot be a new born or to be born');
    end if;
end;
```

- Trigger này sẽ ngăn chặn người dùng nhập vào một ngày sinh lớn hơn ngày hiện tại

**\* Lưu ý:** Chúng ta không thể viết check trực tiếp trên 2 bảng mà phải viết trigger do Oracle không hỗ trợ các câu lệnh check trên các biến không có trên bảng mà ta đang check (trong 2 trường hợp trên, biến không có trong bảng employee chính là biến sysdate thuộc vào một bảng khác có sẵn của hệ thống). **Các trường hợp tương tự cho các bảng khác bên dưới nhóm sẽ không giải thích lại.**

- Update Trigger:

```
TRIGGER update_fk_ecode
AFTER UPDATE OF ECODE ON EMPLOYEE
FOR EACH ROW
BEGIN
    UPDATE PHONE
    SET PHONE.ECODE = :NEW.ECODE
    WHERE :OLD.ECODE = PHONE.ECODE;

    UPDATE DOCTOR
    SET DOCTOR.DCODE = :NEW.ECODE
    WHERE :OLD.ECODE = DOCTOR.DCODE;

    UPDATE NURSE
    SET NURSE.NCODE = :NEW.ECODE
    WHERE :OLD.ECODE = NURSE.NCODE;
END;
```

\* **Lưu ý:** Một lần nữa, do Oracle không hỗ trợ từ khóa on update để mô tả hành động khi bảng được tham chiếu đến thay đổi giá trị trường được tham chiếu bởi khóa ngoại nên chúng ta phải viết trigger để tự update lại cho đúng giá trị các khóa ngoại. **Các trường hợp tương tự cho các bảng khác bên dưới nhóm sẽ không nhắc lại mà chỉ để tựa mục là Update Trigger.**

## 2. DOCTOR và NURSE

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 DCODE	NUMBER (38,0)	No	(null)	1	(null)

Bảng định nghĩa cột dữ liệu cho bảng Doctor

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 NCODE	NUMBER (38,0)	No	(null)	1	(null)

Bảng định nghĩa cột dữ liệu cho bảng Nurse

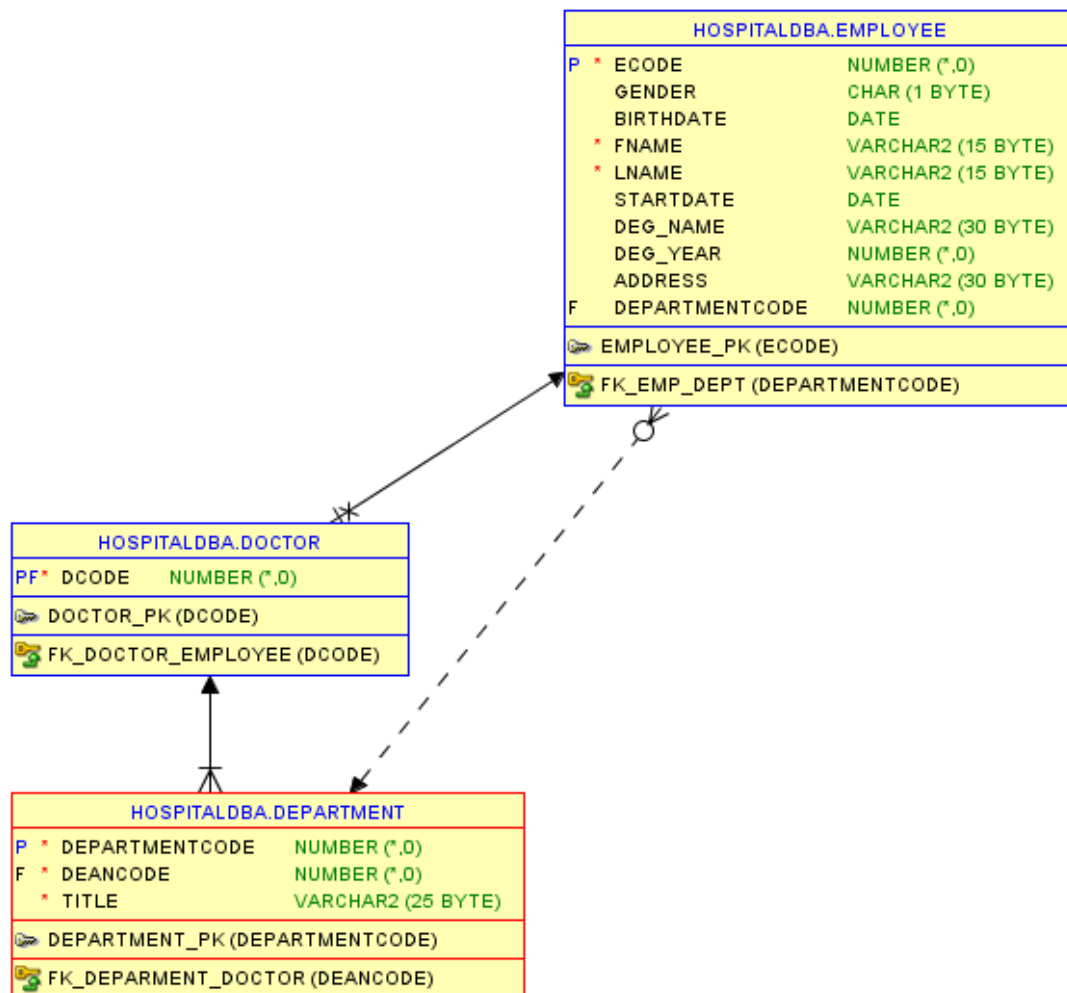
- Constraint về toàn vẹn dữ liệu:
  - Hai bảng này là hai bảng kế thừa từ bảng EMPLOYEE
  - Bảng DOCTOR có khóa chính là doccode và bảng NURSE có khóa chính là ncode. Hai trường doccode và ncode đều tham chiếu đến trường Ecode của bảng EMPLOYEE
  - Cả 2 khóa chính của 2 bảng nhóm đều sử dụng kiểu dữ liệu là INT, ban đầu nhóm có quyết định là sử dụng kiểu VARCHAR để trong mã nhân viên có cả ký tự thể hiện chuyên ngành của nhân viên nhưng sau khi xem xét lại nhóm nhận thấy 2 bảng DOCTOR và NURSE kế thừa bảng EMPLOYEE và trong bảng EMPLOYEE lại có trường Deg\_Name đã thể hiện được chuyên ngành của nhân viên nên nhóm quyết định chuyển mã nhân viên qua dạng số.
- Constraint về toàn vẹn tham chiếu:
  - Constraint cho 2 trường doccode và ncode là ON DELETE CASCADE để khi xóa nhân viên ở bảng EMPLOYEE thì nhân viên tương ứng ở bảng DOCTOR hoặc NURSE cũng sẽ xóa theo



### 3. DEPARTMENT

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DEPARTMENTCODE	NUMBER (38,0)	No	(null)	1	(null)
2	DEANCODE	NUMBER (38,0)	No	(null)	2	(null)
3	TITLE	VARCHAR2 (25 BYTE)	No	(null)	3	(null)

Định nghĩa các cột dữ liệu của bảng DEPARTMENT



Mô hình biểu diễn các khóa ngoại của bảng Department

- Constraint về toàn vẹn dữ liệu:
  - Khóa chính của bảng là departmentcode có kiểu dữ liệu là INT, cũng như khóa chính của 2 bảng DOCTOR và NURSE trên thì ban đầu nhóm có sử dụng kiểu VARCHAR cho trường departmentcode để thể hiện được phòng của khoa nào, tuy nhiên trong bảng DEPARTMENT đã có trường title (kiểu dữ liệu là VARCHAR(25)) thể hiện được tên của khoa nên nhóm quyết định chuyển từ VARCHAR sang kiểu INT cho khóa chính departmentcode

- Trường deancode có kiểu dữ liệu là INT và được ràng buộc NOT NULL vì deancode là mã của nhân viên là trưởng phòng ban nên không được để trống và trường này tham chiếu đến doccode của bảng DOCTOR
- Trường title có kiểu dữ liệu là VARCHAR(25) thể hiện tên của phòng ban với độ dài tên tối đa được nhóm ước lượng.
- Constraint về toàn vẹn tham chiếu:
  - Constraint cho khóa ngoại deancode là ON DELETE SET NULL để khi xóa dữ liệu về một bác sĩ ở bảng DOCTOR thì giá trị ở trường deancode sẽ là NULL và dữ liệu về phòng ban vẫn sẽ không bị xóa
- Các trigger:

```
trigger dean_experience_check
before insert or update on department
for each row
declare
  dean_deg_year employee.deg_year%TYPE;
begin
  select deg_year into dean_deg_year
  from employee
  where ecode = :new.deancode;
  if (TO_CHAR(dean_deg_year) > TO_CHAR(EXTRACT(YEAR FROM Sysdate) - 5))
  then raise_application_error(-20002,'The dean does not have enough experience.Choose
another guy!');
  end if;
end;
```

- Trigger này đảm bảo rằng một người bác sĩ làm trưởng phòng ban sẽ có kinh nghiệm lớn hơn 5 năm theo yêu cầu của nghiệp vụ.

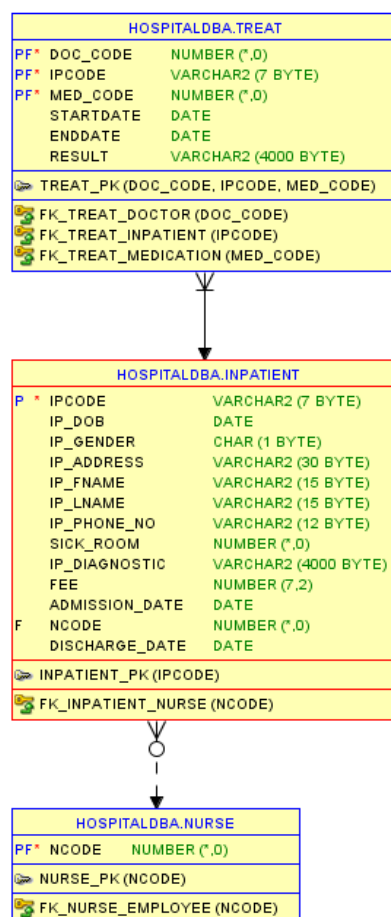
```
TRIGGER update_fk_departmentcode
AFTER UPDATE OF DEPARTMENTCODE ON DEPARTMENT
FOR EACH ROW
BEGIN
  UPDATE EMPLOYEE
  SET EMPLOYEE.DEPARTMENTCODE = :NEW.DEPARTMENTCODE
  WHERE :OLD.DEPARTMENTCODE = EMPLOYEE.DEPARTMENTCODE;
END;
```

Update trigger cho bảng department

## 4. INPATIENT

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	IPCODE	VARCHAR2 (7 BYTE)	No	(null)	1	(null)
2	IP DOB	DATE	Yes	(null)	2	(null)
3	IP GENDER	CHAR (1 BYTE)	Yes	(null)	3	(null)
4	IP ADDRESS	VARCHAR2 (30 BYTE)	Yes	(null)	4	(null)
5	IP FNAME	VARCHAR2 (15 BYTE)	Yes	(null)	5	(null)
6	IP LNAME	VARCHAR2 (15 BYTE)	Yes	(null)	6	(null)
7	IP PHONE NO	VARCHAR2 (12 BYTE)	Yes	(null)	7	(null)
8	SICK ROOM	NUMBER (38,0)	Yes	(null)	8	(null)
9	IP DIAGNOSTIC	VARCHAR2 (4000 ...	Yes	(null)	9	(null)
10	FEE	NUMBER (7,2)	Yes	(null)	10	(null)
11	ADMISSION DATE	DATE	Yes	(null)	11	(null)
12	NCODE	NUMBER (38,0)	Yes	(null)	12	(null)
13	DISCHARGE DATE	DATE	Yes	(null)	13	(null)

Bảng định nghĩa các cột dữ liệu của INPATIENT



Mô hình dữ liệu và tham chiếu của bảng INPATIENT

- Constraint về toàn vẹn dữ liệu:
  - Khóa chính của bảng là IPCode có kiểu dữ liệu là VARCHAR(7) với ràng buộc bắt đầu bằng 2 ký tự và theo sau đó là 5 chữ số, 2 ký tự đầu phân loại bệnh nhân (IP là inpatient)
  - Trường IP\_Gender sử dụng kiểu dữ liệu CHAR(1) như ở bảng EMPLOYEE chỉ sử dụng 1 trong 2 ký tự M (nam) hoặc F (nữ) để biểu thị giới tính của bệnh nhân
  - Với các trường IP\_Address, IP\_FName và IP\_LName thì có kiểu dữ liệu lần lượt là VARCHAR(30), VARCHAR(15) và VARCHAR(15) chứa các dữ liệu về địa chỉ, họ tên của bệnh nhân
  - Trường IP\_Phone\_No có kiểu dữ liệu VARCHAR(12), nhóm sử dụng kiểu này để lưu số điện thoại bệnh nhân vì khi nhóm muốn lưu số điện thoại có số 0 đứng đầu hoặc +84 đứng đầu thì không thể sử dụng kiểu NUMBER được vì kiểu dữ liệu này sẽ bỏ số 0 đứng đầu, vì vậy nhóm sử dụng kiểu VARCHAR cho trường này, nhóm chọn độ dài chuỗi tối đa là 12 vì trường hợp thông tin số điện thoại nhập dài nhất là +84 đứng đầu có 12 ký tự
  - Trường IP\_Diagnosis ban đầu nhóm có sử dụng kiểu TEXT nhưng trong oracle không chấp nhận kiểu dữ liệu này nên sau khi tìm hiểu thì nhóm quyết định chọn kiểu VARCHAR2(4000), nhóm chọn kiểu dữ liệu này vì kiểu này có độ dài lớn hơn VARCHAR
  - Các trường IP\_DoB và Admission\_Date lưu giữ ngày tháng năm nên sử dụng kiểu dữ liệu DATE
  - Trường Sick\_Room có kiểu dữ liệu là SMALLINT lưu giữ số của phòng bệnh
  - Trường Fee nhóm sử dụng kiểu dữ liệu DECIMAL(7,2) theo định dạng tiền Dollar Mỹ
- Constraint về toàn vẹn tham chiếu:
  - Trường NCode là khóa ngoại của bảng tham chiếu tới ncode của bảng NURSE
  - Constraint cho khóa ngoại NCode là ON DELETE SET NULL để thông tin của một inpatient vẫn sẽ được lưu lại khi mà dữ liệu của y tá của bệnh nhân bị xóa đi.
- Constraint về toàn vẹn ngữ nghĩa:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 CHECK_NEGATIVE_FEE	Check	fee >= 0
2 CHECK_NEGATIVE_SICK_ROOM	Check	sick room >=0
3 FK_INPATIENT_NURSE	Foreign Key	(null)
4 INPATIENT_GENDER	Check	regexp like(ip gender, '(F M)')
5 IP_NO_DIGIT_FNAME	Check	not regexp like(ip fname, '[0-9]*')
6 IP_NO_DIGIT_LNAME	Check	not regexp like(ip lname, '[0-9]*')
7 SYS_C007269	Primary Key	(null)
8 VALID_INPATIENT_DATE	Check	ip dob < admission date and admission date <= discharge date

#### Các constraint của bảng INPATIENT

- Constraint CHECK\_NEGATIVE\_FEE đảm bảo rằng chi phí nằm viện của một bệnh nhân luôn dương

- Constraint CHECK\_NEGATIVE\_SICK\_ROOM đảm bảo rằng phòng bệnh là một số dương
- Constraint INPATIENT GENDER bắt buộc giới tính một bệnh nhân phải là 'F' hoặc 'M'
- Constraint IP\_NO\_DIGIT\_FNAME và IP\_NO\_DIGIT\_LNAME đảm bảo tên bệnh nhân sẽ không chứa chữ số
- Constraint VALID\_INPATIENT\_DATE đảm bảo rằng bệnh nhân chỉ được vào viện sau khi đã được sinh ra và ngày xuất viện của bệnh nhân phải lớn hơn ngày mà bệnh nhân nhập viện.
- Các Trigger:

```
trigger inpatient_valid_birthdate
before insert or update of ip_dob
on inpatient
for each row
begin
  if (extract(year from :new.ip_dob) > extract(year from sysdate))
  then raise_application_error(-20007,'An person will not exist until today');
  end if;
end;
```

- Trigger ngăn chặn người dùng nhập ngày sinh lớn hơn thời điểm hiện tại

```
trigger patient_format_code_check
before insert or update of ipcode
on inpatient
for each row
begin
  if not REGEXP_LIKE(:new.ipcode,'IP\d{5}')
  then raise_application_error(-20004,'Wrong format code for inpatient. Please check if the
code is in format IPXXXX where X is a digit!');
  end if;
end;
```

- Trigger kiểm tra người dùng có nhập đúng format của mã số bệnh nhân là 'IP' theo sau là 5 chữ số trong hệ thập phân.

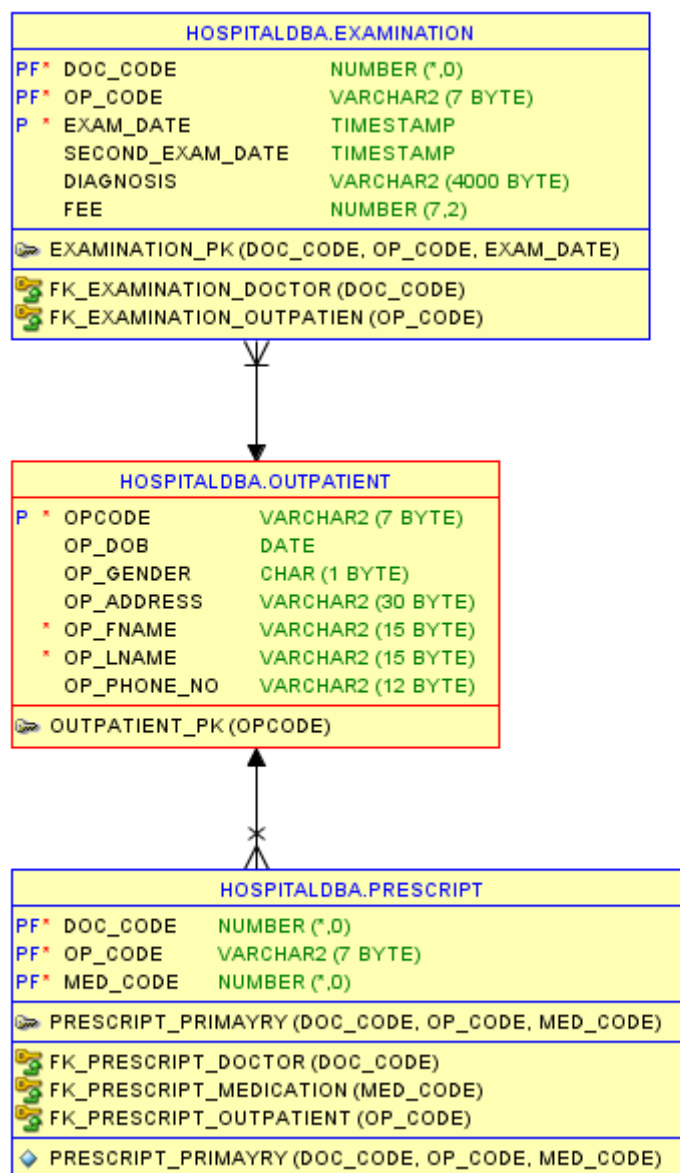
```
TRIGGER update_fk_ipcode
AFTER UPDATE OF IPCODE ON INPATIENT
FOR EACH ROW
BEGIN
  UPDATE TREAT
  SET TREAT.IPCODE = :NEW.IPCODE
  WHERE :OLD.IPCODE = TREAT.IPCODE;
END;
```

Update trigger cho bảng INPATIENT

## 5. OUTPATIENT

	↕ COLUMN_NAME	↕ DATA_TYPE	↕ NULLABLE	DATA_DEFAULT
1	OPCODE	VARCHAR2(7 BYTE)	No	(null)
2	OP DOB	DATE	Yes	(null)
3	OP GENDER	CHAR(1 BYTE)	Yes	(null)
4	OP ADDRESS	VARCHAR2(30 BYTE)	Yes	(null)
5	OP FNAME	VARCHAR2(15 BYTE)	No	(null)
6	OP LNAME	VARCHAR2(15 BYTE)	No	(null)
7	OP PHONE NO	VARCHAR2(12 BYTE)	Yes	(null)

Định nghĩa các cột dữ liệu của bảng OUTPATIENT



Mô hình dữ liệu và tham chiếu của bảng OUTPATIENT

- Constraint về toàn vẹn dữ liệu:

- Khóa chính của bảng là OPCode có kiểu dữ liệu VARCHAR(7) có các ràng buộc như bắt đầu bằng 2 ký tự và còn lại là các chữ số, cũng như khóa chính của bảng INPATIENT thì OPCode cũng được sử dụng để phân loại bệnh nhân (OP là outpatient)
- Trường OP\_Gender thì như ở các bảng khác sử dụng kiểu dữ liệu CHAR(1) cũng như các trường OP\_Address, OP\_FName, OP\_LName, OP\_DOB và OP\_Phone\_No sử dụng các kiểu dữ liệu tương tự như trong bảng INPATIENT

- Constraint về toàn vẹn ngữ nghĩa:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
1 OP_GENDER_CHECK	Check	regexp like(op gender, '(F M)')
2 OP_NO_DIGIT_FNAME	Check	not regexp like(op fname, '[0-9]')
3 OP_NO_DIGIT_LNAME	Check	not regexp like(op lname, '[0-9]')
4 SYS_C007265	Check	"OP_FNAME" IS NOT NULL
5 SYS_C007266	Check	"OP_LNAME" IS NOT NULL
6 SYS_C007267	Primary Key	(null)

- Constraint OP\_GENDER\_CHECK giúp đảm bảo trường giới tính chỉ có thể mang giá trị 'F' hoặc 'M'
- Constraint OP\_NO\_DIGIT\_FNAME và OP\_NO\_DIGIT\_LNAME đảm bảo tên bệnh nhân không chứa chữ số

- Các trigger:

```
trigger outpatient_code_check
before insert or update of opcode
on outpatient
for each row
begin
    if not REGEXP_LIKE(:new.opcode, 'OP\d{5}')
    then raise_application_error(-20005, 'Wrong format code for outpatient. Correct format is:
OPXXXXX where X is a digit');
    end if;
end;
```

- Trigger kiểm tra người dùng có nhập đúng format của mã số bệnh nhân là 'OP' theo sau là 5 chữ số trong hệ thập phân.

```
trigger outpatient_valid_birthdate
before insert or update of op_dob
on outpatient
for each row
begin
    if (extract(year from :new.op_dob) > extract(year from sysdate))
    then raise_application_error(-20008, 'In valid birthdate of outpatient');
    end if;
```

```
end;
```

- Trigger kiểm tra người dùng có nhập vào một ngày sinh hợp lệ hay không (so sánh với thời điểm hiện tại)

```
TRIGGER update_fk_opcode  
AFTER UPDATE OF OPCODE ON OUTPATIENT  
FOR EACH ROW  
BEGIN  
    UPDATE EXAMINATION  
    SET EXAMINATION.OP_CODE = :NEW.OPCODE  
    WHERE :OLD.OPCODE = EXAMINATION.OP_CODE;  
  
    UPDATE PRESCRIPT  
    SET PRESCRIPT.OP_CODE = :NEW.OPCODE  
    WHERE :OLD.OPCODE = PRESCRIPT.OP_CODE;  
END;
```

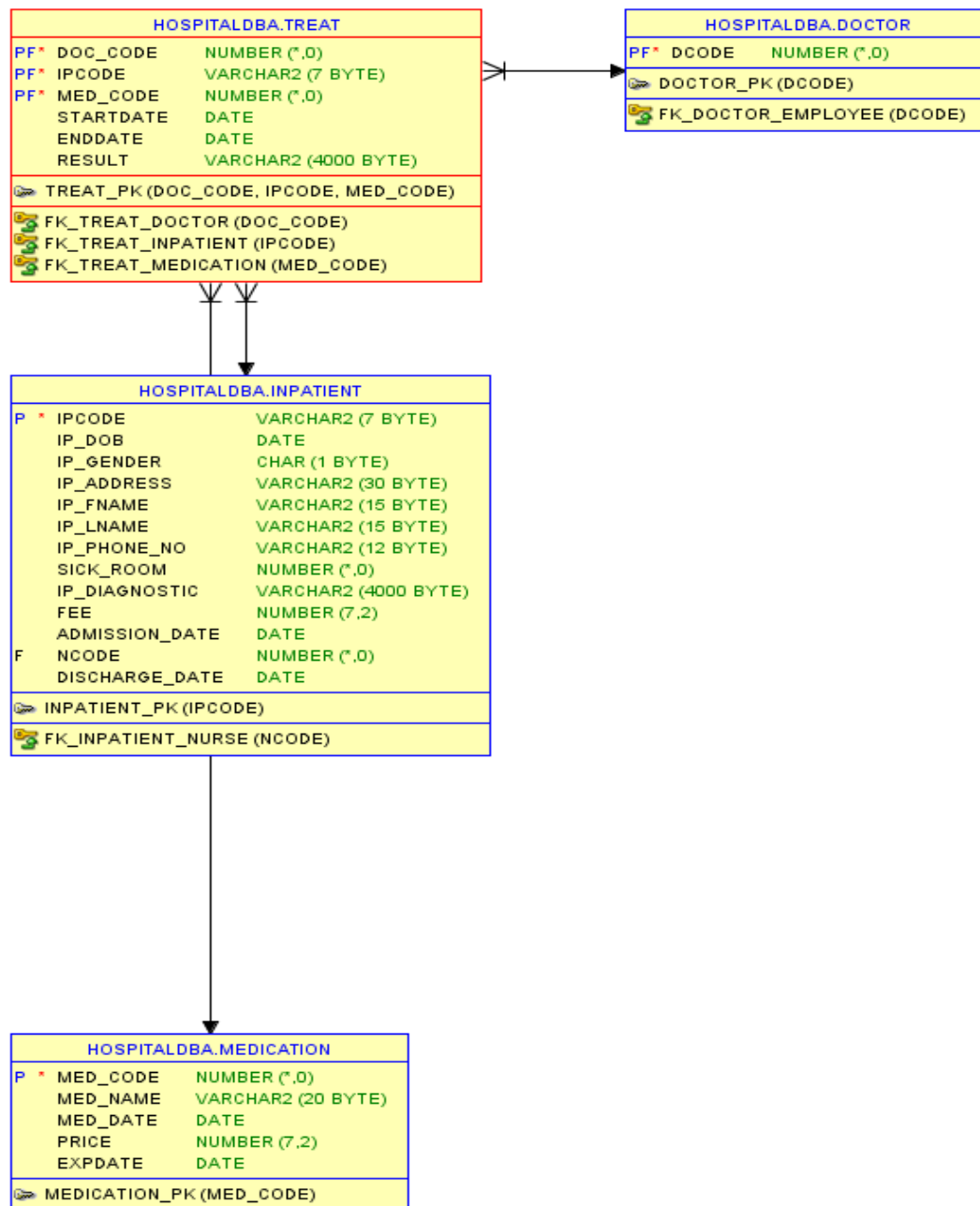
Update trigger cho bảng OUTPATIENT



## 6. TREAT

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	DOC CODE	NUMBER (38,0)	No	(null)
2	IPCODE	VARCHAR2 (7 BYTE)	No	(null)
3	MED CODE	NUMBER (38,0)	No	(null)
4	STARTDATE	DATE	Yes	(null)
5	ENDDATE	DATE	Yes	(null)
6	RESULT	VARCHAR2 (4000 ...)	Yes	(null)

Bảng định nghĩa các cột dữ liệu cho TREAT



Mô hình dữ liệu và tham chiếu của TREAT

- Constraint về toàn vẹn dữ liệu:

- Bảng này sử dụng 3 trường Doc\_Code, IPCODE và Med\_Code làm khóa chính, trong đó Doc\_code được tham chiếu đến trường doccode của bảng DOCTOR, IPCODE tham chiếu đến trường IPCODE của bảng INPATIENT và Med\_Code tham chiếu đến trường Med-code của bảng MEDICATION
- Các trường StartDate và EndDate lưu dưới định dạng ngày tháng năm nên sử dụng kiểu dữ liệu DATE cho 2 trường này
- Trường Result thì nhóm nhận thấy trường này sẽ lưu nhiều ký tự nên nhóm sử dụng kiểu VARCHAR2(4000) như đã sử dụng ở bảng INPATIENT

- Constraint về toàn vẹn tham chiếu:

- Constraint cho cả 3 trường tham chiếu đến các bảng khác là giá trị của các trường sẽ không thay đổi khi mà dữ liệu được lưu trong các trường doccode, IPCODE và Med\_code ở các bảng DOCTOR, INPATIENT và MEDICATION bị xóa. Lý do nhóm chọn như vậy vì nhóm muốn thông tin của một lần điều trị bác sĩ đảm nhận, bệnh nhân được điều trị và loại thuốc sử dụng sẽ được lưu lại để có thể xem lại sau này khi có vấn đề xảy ra (bệnh nhân khám ở bệnh viện khác và muốn xem lại lịch sử khám cũ để có thêm thông tin cho bác sĩ ở bệnh viện khác)

- Constraint về toàn vẹn ngữ nghĩa:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
CHECK_VALID_TREAT_PERIOD	Check	StartDate < EndDate
FK TREAT DOCTOR	Foreign Key	(null)
FK TREAT INPATIENT	Foreign Key	(null)
FK TREAT MEDICATION	Foreign Key	(null)
SYS C007277	Primary Key	(null)

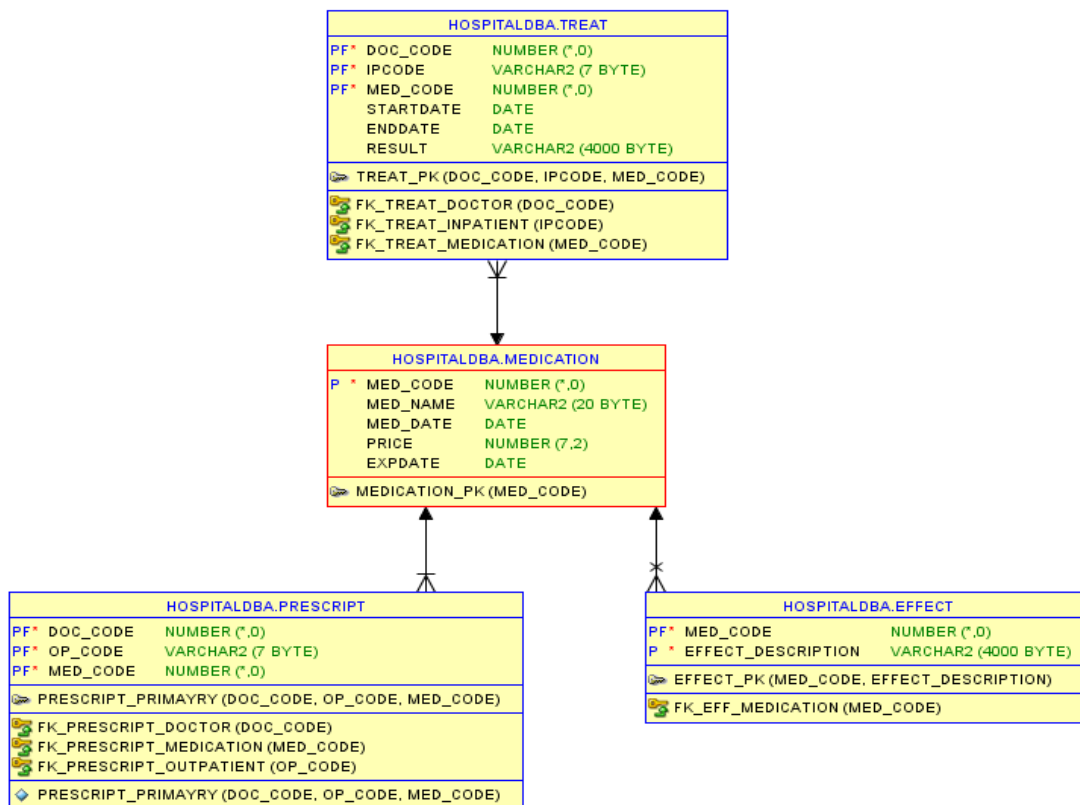
#### Các constraint của bảng TREAT

- Constraint CHECK\_VALID\_TREAT\_PERIOD giúp chúng ta đảm bảo rằng thời điểm bắt đầu sẽ nhỏ hơn thời điểm kết thúc một phiên chữa trị.

## 7. MEDICATION

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEFAULT
1	MED CODE	NUMBER(38,0)	No	(null)
2	MED NAME	VARCHAR2(20 BYTE)	Yes	(null)
3	MED DATE	DATE	Yes	(null)
4	PRICE	NUMBER(7,2)	Yes	(null)
5	EXPDATE	DATE	Yes	(null)

Định nghĩa cho các cột dữ liệu của bảng MEDICATION



Mô hình dữ liệu và tham chiếu của bảng MEDICATION

- Constraint về toàn vẹn dữ liệu:
  - Với trường Med\_code là khóa chính có kiểu dữ liệu là INT để lưu các mã của thuốc dưới dạng số
  - Trường Med\_Name để lưu tên các loại thuốc nên có kiểu dữ liệu VARCHAR(20)
  - Hai trường Med\_Date và ExpDate lưu dữ liệu ngày nhập thuốc và ngày hết hạn dưới định dạng ngày tháng năm nên có kiểu dữ liệu là DATE
  - Trường Price giống như trường Fee ở bảng INPATIENT lưu ở dạng DECIMAL(7,2) lấy 2 số thập phân sau dấu chấm

- Constraint về toàn vẹn ngữ nghĩa:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
<sup>1</sup> CHECK_MED_NAME	Check	not regexp like (med name, '[0-9]')
<sup>2</sup> MEDICATION_PRICE_CHECK	Check	price >=0
<sup>3</sup> MEDICATION_VALIDITY_C...	Check	med date < expDate
<sup>4</sup> SYS_C007274	Primary Key	(null)

- Constraint CHECK\_MED\_NAME đảm bảo rằng tên một loại thuốc sẽ không chứa chữ số.
- Constraint MEDICATION\_PRICE\_CHECK đảm bảo rằng giá của một đơn vị thuốc sẽ dương.
- Constraint MEDICATION\_VALIDITY\_CHECK đảm bảo rằng các thuốc không hợp lệ sẽ không được nhập vào cơ sở dữ liệu. (với med\_date là ngày thuốc được nhập vào và expDate là ngày hết hạn của thuốc)
- Các trigger

```
trigger check_valid_exp_date
before insert or update of expdate
on medication
for each row
begin
    if (:new.expdate < sysdate)
    then raise_application_error(-20006, 'The medicine is outdated. Check or use another
one');
    end if;
end;
```

- Trigger kiểm tra rằng thời hạn sử dụng của thuốc vẫn còn tính cho đến ngày hôm nay. Lưu ý rằng Trigger này không bị lặp tác dụng với constraint MEDICATION\_VALIDITY\_CHECK ở trên.

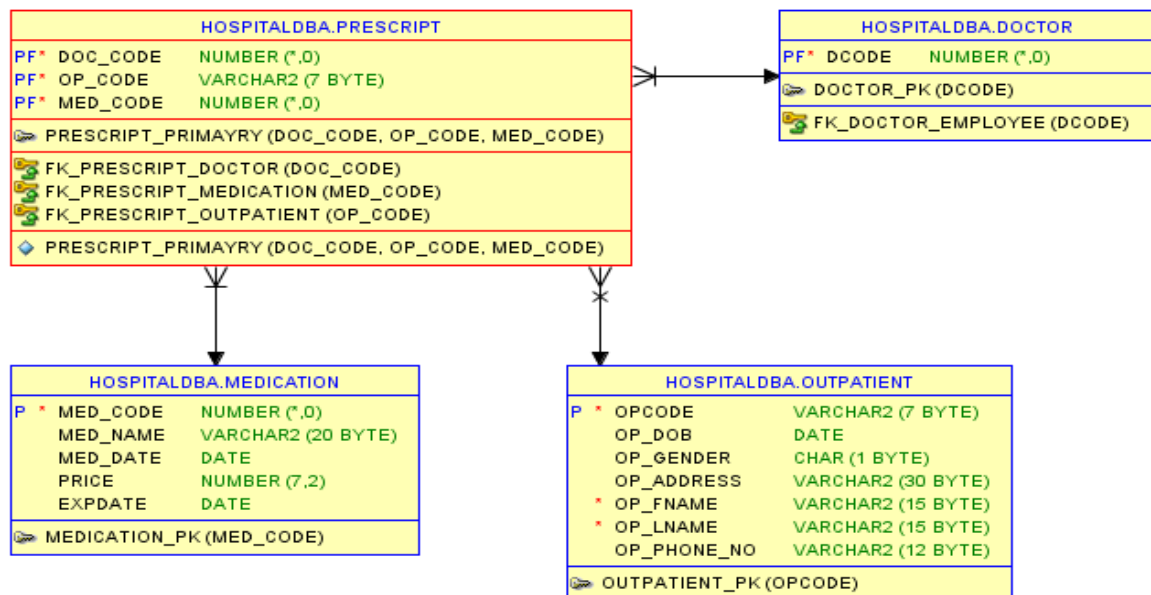
```
trigger check_med_date_validity
before insert or update of med_date
on medication
for each row
begin
    if (:new.med_date > sysdate)
    then raise_application_error(-20010, 'Invalid medication date. It should be earlier than
today');
    end if;
end;
```

- Trigger kiểm tra rằng ngày nhập thuốc là một thời điểm trước thời điểm hiện tại.

## 8. PRESCRIPT

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	DOC_CODE	NUMBER (38,0)	No	(null)
2	OP_CODE	VARCHAR2 (7 BYTE)	No	(null)
3	MED_CODE	NUMBER (38,0)	No	(null)

Bảng định nghĩa các cột dữ liệu của bảng PRESCRIPT



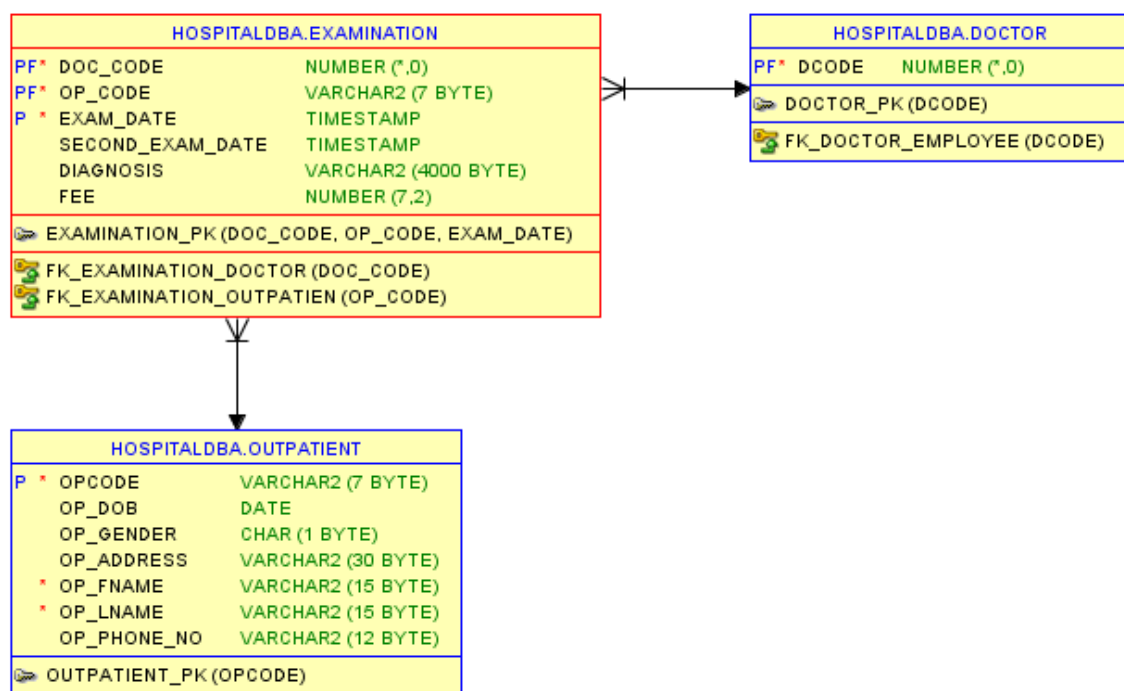
Mô hình dữ liệu của bảng PRESCRIPT

- Constraint về toàn vẹn dữ liệu:
  - Có khóa chính là 3 trường Doc\_Code tham chiếu đến doccode của bảng DOCTOR, OP\_Code tham chiếu đến OPCODE của bảng OUTPATIENT và Med\_Code tham chiếu đến Med\_code của bảng MEDICATION
- Constraint về toàn vẹn tham chiếu
  - Constraint cho trường Doc\_Code là không thay đổi dữ liệu của trường này khi bác sĩ bị xóa ở bảng DOCTOR vì mặc dù dữ liệu của một bác sĩ đã xóa đi khi một bác sĩ không hoạt động trong bệnh viện nữa thì bệnh viện vẫn có thể xem lại thông tin đơn thuốc của bác sĩ nào kê đơn (dùng để truy vấn trách nhiệm hoặc truy vấn thêm thông tin trước khi ra quyết định khác cho bệnh nhân trong tương lai).
  - Constraint cho trường OP\_Code là ON DELETE CASCADE, nhóm chọn constraint này vì khi bệnh viện xóa thông tin bệnh nhân thì các đơn thuốc liên quan sẽ bị xóa theo
  - Constraint cho trường Med\_Code là ON DELETE SET NULL vì để có thể xem lại các thông tin trong đơn thuốc như bác sĩ kê đơn thuốc thì khi xóa một loại thuốc thì ở giá trị ở trường Med\_Code này sẽ là NULL

## 9. EXAMINATION

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEFAULT
1	DOC CODE	NUMBER (38,0)	No	(null)
2	OP CODE	VARCHAR2 (7 BYTE)	No	(null)
3	EXAM DATE	TIMESTAMP (6)	No	(null)
4	SECOND EXAM DATE	TIMESTAMP (6)	Yes	(null)
5	DIAGNOSIS	VARCHAR2 (4000 ...)	Yes	(null)
6	FEE	NUMBER (7,2)	Yes	(null)

Định nghĩa cho các cột dữ liệu của bảng EXAMINATION



Mô hình dữ liệu và tham chiếu của bảng EXAMINATION

- Constraint về toàn vẹn dữ liệu:
  - Exam\_Date có kiểu dữ liệu là DATE lưu trữ thông tin ngày kiểm tra
  - Trường SECOND\_EXAMDATE lưu trữ thông tin ngày tháng năm kiểm tra lần 2 của bệnh nhân nên có kiểu dữ liệu là DATE
  - Trường Diagnosis có kiểu dữ liệu VARCHAR2(4000) vì trường này để lưu trữ các chẩn đoán nên có thể sẽ có độ dài ký tự lớn nên nhóm sử dụng kiểu dữ liệu này
  - Trường Fee sử dụng kiểu dữ liệu DECIMAL(7,2) để lưu trữ như những trường lưu trữ thông tin về giá tiền ở các bảng khác
- Constraint về toàn vẹn tham chiếu:

- Khóa chính có 3 trường là Doc\_Code tham chiếu đến doccode của bảng DOCTOR, OP\_Code tham chiếu đến OPCode của bảng OUTPATIENT.
- Constraint cho cả hai trường Doc\_Code và OP\_Code đều là không thay đổi khi dữ liệu ở các bảng DOCTOR và OUTPATIENT bị xóa vì để xem lại thông tin về một lần kiểm tra dù cho bác sĩ không còn hoạt động ở bệnh viện hoặc thông tin bệnh nhân đã bị xóa
- Constraint về toàn vẹn ngữ nghĩa:

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
<sup>1</sup> CHECK_VALID_REEXAM	Check	exam date < second exam date
<sup>2</sup> FK_EXAMINATION_DOCTOR	Foreign Key	(null)
<sup>3</sup> FK_EXAMINATION_OUTPATIENT	Foreign Key	(null)
<sup>4</sup> NEGATIVE_FEE_CHECK	Check	fee >= 0
<sup>5</sup> SYS_C007271	Primary Key	(null)

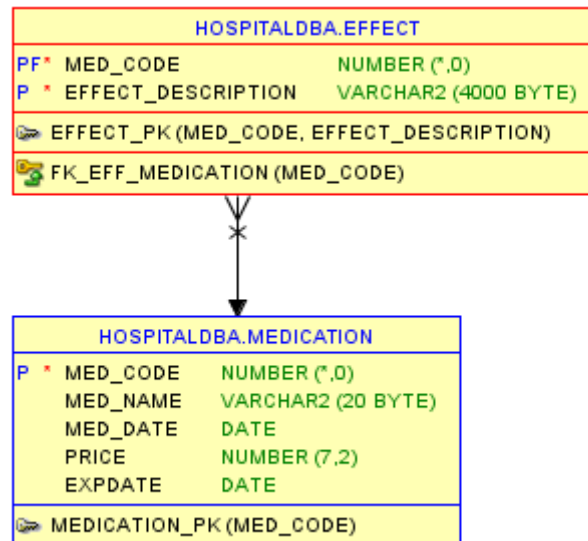
#### Các constraint của bảng EXAMINATION

- Constraint CHECK\_VALID\_REEXAM đảm bảo rằng ngày tái khám sẽ là một thời điểm sau lần khám được lưu trong mỗi record.
- Constraint NEGATIVE\_FEE\_CHECK đảm bảo rằng chi phí mỗi lần khám sẽ là một số dương.

## 10. EFFECT

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	MED_CODE	NUMBER (38,0)	No	(null)
2	EFFECT_DESCRIPTION	VARCHAR2 (4000 ...)	No	(null)

Định nghĩa các cột dữ liệu của bảng EFFECT



Mô hình dữ liệu và tham chiếu của EFFECT

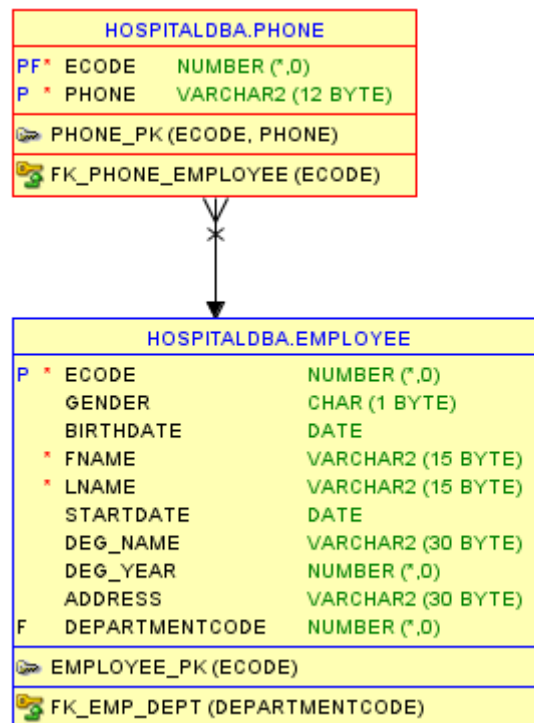
- Constraint về toàn vẹn dữ liệu:
  - Khóa chính có 2 trường là Med\_Code tham chiếu đến Med\_code của bảng MEDICATION và Effect\_Description có kiểu dữ liệu là VARCHAR2(4000) để lưu trữ thông tin về tác dụng của thuốc
- Constraint về toàn vẹn tham chiếu:
  - Constraint cho trường Med\_Code là ON DELETE CASCADE vì khi dữ liệu của một loại thuốc đã bị xóa thì thông tin về tác dụng của thuốc cũng sẽ không cần và bị xóa theo



## 11. PHONE:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT
1	ECODE	NUMBER (38,0)	No	(null)
2	PHONE	VARCHAR2 (12 BYTE)	No	(null)

Định nghĩa các cột dữ liệu của bảng PHONE



Mô hình dữ liệu và tham chiếu của bảng PHONE

- Constraint về toàn vẹn dữ liệu của bảng PHONE:
  - Bảng có 2 khóa chính là Ecode tham chiếu đến ECode của bảng EMPLOYEE và trường phone lưu trữ thông tin số điện thoại của nhân viên có kiểu dữ liệu là VARCHAR(12)
- Constraint về toàn vẹn tham chiếu của bảng PHONE:
  - Constraint cho trường Ecode là ON DELETE CASCADE vì khi xóa thông tin của một nhân viên không còn hoạt động ở bệnh viện thì thông tin về số điện thoại cũng sẽ bị xóa theo

## PART B: STORE PROCEDURE, FUNCTION, SQL

**a) Increase Inpatient Fee to 10% for all the inpatients who are admitted to hospital from 01/09/2020.**

Source Code:

```
CREATE OR REPLACE PROCEDURE increase_fee
(patientID IN INPATIENT.IPCODE%TYPE)
AS
    admitted_date DATE;
BEGIN
    SELECT ADMISSION_DATE INTO admitted_date
    FROM INPATIENT
    WHERE IPCODE = patientID;

    IF (admitted_date > TO_DATE('01-SEP-2020')) THEN
        UPDATE INPATIENT
        SET FEE = FEE*1.1
        WHERE IPCODE = patientID;
        COMMIT;
    END IF;
END;
```

Ở đây, nhóm đã tạo một procedure để tăng lương cho các bệnh nhân nhập viện từ ngày 1 tháng 9 năm 2020. Procedure được định nghĩa với một tham số đầu vào là Mã số bệnh nhân và một biến tạm là ngày nhập viện của bệnh nhân này. Nếu như ngày nhập viện của bệnh nhân là sau ngày 1 tháng 9 năm 2020 thì viện phí của bệnh nhân sẽ bị tăng lên 10%.

Việc cung cấp cho procedure một mã số bệnh nhân có thể giúp chúng ta viết các trigger update dễ dàng hơn tương lai với tùy chọn FOR EACH ROW của trigger

**b) Select all the patients (outpatients and inpatients) of a named doctor.**

Source code:

```
WITH patient AS
(
  SELECT DOC_CODE, OP_FNAME AS FNAME, OP_LNAME AS LNAME
  FROM OUTPATIENT, EXAMINATION
  WHERE OUTPATIENT.OPCODE = EXAMINATION.OP_CODE
  UNION ALL
  SELECT DOC_CODE, IP_FNAME AS FNAME, IP_LNAME AS LNAME
  FROM INPATIENT, TREAT
  WHERE INPATIENT.IPCODE = TREAT.IPCODE
)
SELECT patient.FNAME, patient.LNAME
FROM patient, DOCTOR, EMPLOYEE
WHERE DOCTOR.DCODE = EMPLOYEE.ECODE AND patient.DOC_CODE = DOCTOR.DCODE
AND EMPLOYEE.FNAME = 'Nguyen Hoang' AND EMPLOYEE.LNAME = 'Thinh';
```

Ở đây, nhóm lưu lại toàn bộ tên và doc\_code của mỗi bệnh nhân (bao gồm cả outpatient và inpatient) trong một lần khám qua phép UNION vào một bảng tạm là patient và select từ bảng tạm này các bệnh nhân có tên bác sĩ trùng với bác sĩ cần tìm.

**c) Write a function to calculate the total medication price a patient has to pay for each treatment or examination.**

Source code:

```
CREATE OR REPLACE FUNCTION total_price
(patientID IN INPATIENT.IPCODE%TYPE)
RETURN NUMBER
AS
    res NUMBER;
BEGIN
    IF (SUBSTR(patientID, 1, 2) = 'IP') THEN
        SELECT SUM(PRICE) INTO res
        FROM MEDICATION, TREAT
        WHERE TREAT.MED_CODE = MEDICATION.MED_CODE AND IPCODE = patientID
        GROUP BY IPCODE;
    ELSE
        SELECT SUM(FEE) INTO res
        FROM EXAMINATION
        WHERE OP_CODE = patientID
        GROUP BY OP_CODE;
    END IF;
    RETURN res;
END;
```

Ở đây, Function được khai báo với tham số là mã số của bệnh nhân và kiểu trả về là một số (tổng tiền viện phí của bệnh nhân) với biến tạm res cùng kiểu trả về để tạm lưu kết quả. Đầu tiên, Hàm sẽ kiểm tra đây là outpatient hay inpatient thông qua phép lấy chuỗi con **SUBSTR(patientID, 1, 2) = 'IP'.**

- Nếu đây là bệnh nhân nội trú thì ta sẽ tính viện phí của bệnh nhân thông qua giá của các loại thuốc được sử dụng trong các lần điều trị (TREAT).
- Ngược lại, tổng viện phí được tính bằng tổng phí chi trả cho các lần khám(EXAMINATION) đối với outpatient này.

**d) Write a procedure to sort the doctor in increasing number of patients he/she takes care in a period of time.**

Source code:

```
CREATE OR REPLACE PROCEDURE sort_doc
(start_date IN DATE, end_date IN DATE)
AS
    res TABTYPE := TABTYPE();
BEGIN
    WITH info AS
    (
        SELECT DOC_CODE, IPCODE AS PCODE, STARTDATE AS DATEIN
        FROM TREAT
        UNION ALL
        SELECT DOC_CODE, OP_CODE AS PCODE, EXAM_DATE AS DATEIN
        FROM EXAMINATION
    )
    SELECT TEMP(COUNT(PCODE), FNAME, LNAME) BULK COLLECT INTO res
    FROM info, EMPLOYEE
    WHERE info.DOC_CODE = EMPLOYEE.ECODE AND TO_DATE(TO_CHAR(info.DATEIN, 'DD-
MON-YY')) >= TO_DATE(start_date) AND TO_DATE(TO_CHAR(info.DATEIN, 'DD-MON-YY'))
<= TO_DATE(end_date)
    GROUP BY DOC_CODE, FNAME, LNAME
    ORDER BY COUNT(PCODE) ASC;
END;
```

Thủ tục `sort_doc` liệt kê một danh sách bác sĩ theo thứ tự tăng dần của số bệnh nhân mà bác sĩ đó tham gia điều trị hoặc khám bệnh trong một khoảng thời gian. Thủ tục được khai báo với một biến tạm là một bảng **res** và các tham số đầu vào là ngày bắt đầu và ngày kết thúc của khoảng thời gian đang xét. Đầu tiên, hàm sẽ lưu vào bảng tạm **info** toàn bộ các lần khám, chữa bệnh cùng với ngày tháng bắt đầu của toàn bộ các bệnh nhân và bác sĩ. Từ đó, thủ tục sẽ lưu vào bảng tạm **res** lưu kết quả của câu lệnh select trả về một bảng chứa tên đầy đủ (FNAME, LNAME) của bác sĩ và tổng số bệnh nhân mà bác sĩ đó điều trị thông qua hàm gộp COUNT(PCODE) sắp xếp theo thứ tự tăng dần thông qua đoạn lệnh **ORDER BY COUNT(PCODE) ASC;**

# PART C: BUILDING APPLICATION

## I. Tạo một user với quyền DBA:

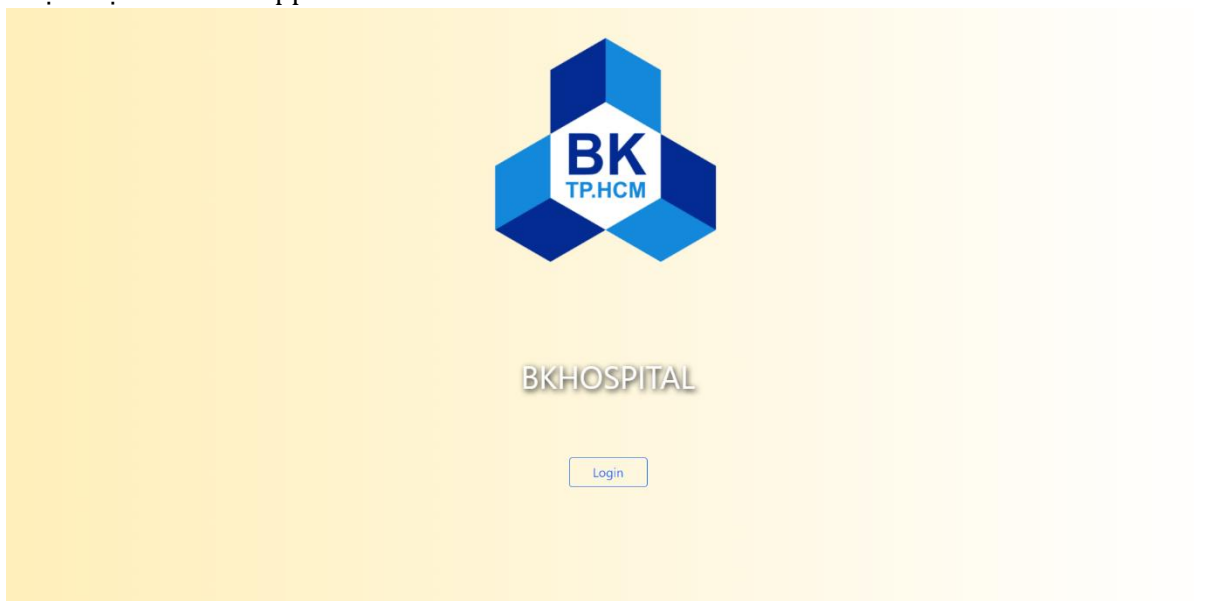
Để tạo một user ta sử dụng lệnh:

```
CREATE USER manager IDENTIFIED BY abcd1234;
```

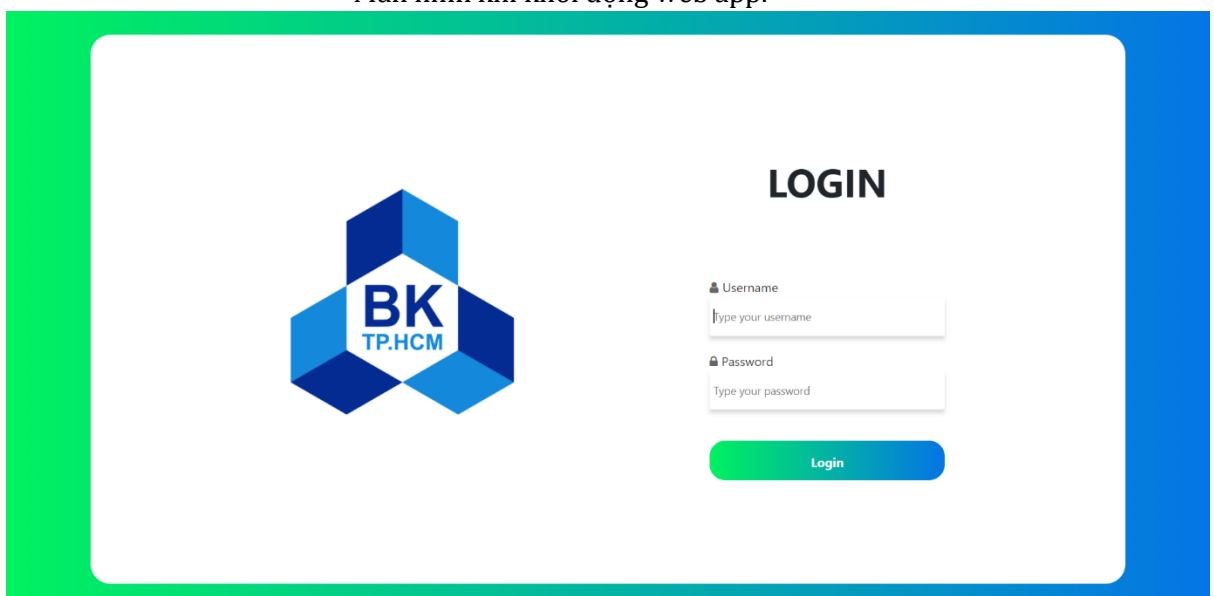
Sau đó ta gán tất cả các quyền cho user trên theo lệnh:

```
GRANT ALL PRIVILEGES TO manager;
```


Hiện thực trên web app:



Màn hình khi khởi động web app.



Người dùng cần điền thông tin vào form để đăng nhập sau khi nhấn nút Login ở màn hình khởi động



## LOGIN

Login failed. Username/Password is incorrect.

Username

askdjas

Password

.....

Login

Nếu nhập sai thông tin. Một tin nhắn báo lỗi sẽ xuất hiện.

## II. Các chức năng được yêu cầu:

### 1. Tìm thông tin bệnh nhân:

BKHospital

Home

Patients

Doctors

Payment Report

Logout

PATIENTS

Searching

nguyen hoang

Search

Add New Patient

Name	Birth Date	Gender	
Nguyen Hoang	3/12/2000	M	<div>View</div>

Kết quả khi tìm kiếm các bệnh nhân

BKHospital

Home

Patients

Doctors

Payment Report

Logout

Patient Information

Name: NGUYEN TINH

Start Date: 11/16/2012

Sick Room: 2

Diagnostic: Perfect health with some minor mental issues

Result: Thanks to whoa. The patient eyesight is increased

Medication: VRohip

Phone Number: 0889991122

End Date: 12/23/2013

Sau khi người dùng chọn vào một trong các kết quả để xem chi tiết hơn



## 2. Thêm thông tin cho bệnh nhân:

BKHospital Home Patients Doctors Payment Report Logout

ADD NEW PATIENT

First Name: Nguyen Van Last Name: A Kind of Patient: Choose...

Phone Number: 0967123456 Date Of Birth: mm/dd/yyyy Gender: Choose...

Address: Ho Chi Minh City

**Below information is of inpatient:**

Diagnosis: Perfect health with some minor mental issues

Admission Date: mm/dd/yyyy Room: 1,2,3... Fee: 15000

Add Patient

Form để cho bệnh nhân có thể chỉnh sửa thông tin cá nhân của mình

## 3. Tìm kiếm các bệnh nhân được điều trị bởi một bác sĩ:

BKHospital Home Patients Doctors Payment Report Logout

Patients Of Doctor

nguyen hoang thinh

Search

Name	Phone Number	Date of Birth
Le Bang	0996671122	3/22/1991
Ly Tinh	0889933322	3/12/2000
Nguyen tinh	0889991122	3/12/2000
Tran Ngoc Duy	+8491238121	3/3/2000

Kết quả khi chúng ta tìm các bệnh nhân được điều trị hoặc khám bởi một bác sĩ

## 4. Tạo báo cáo về một lần khám, chữa trị của một bệnh nhân:

BKHospital Home Patients Doctors Payment Report Logout

PAYMENT

Searching

le bang

Search

Payment of Le Bang	Price
VRohto	1040
paracetamol	1030
Fee for sick room	33200
Total	35270

Báo cáo về một lần thăm khám/điều trị của bệnh nhân