**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BỘ MÔN THỰC HÀNH CƠ SỞ DỮ LIỆU**

**---o0o---**

****

**BÁO CÁO PROJECT**

**THỰC HÀNH CƠ SỞ DỮ LIỆU**

**Project:** *CSDL quản lí đặt lịch khám bệnh, dịch vụ của một*

*cơ sở y tế, bệnh viện*

**Giảng viên hướng dẫn : TS. Đỗ Bá Lâm**

**Lớp : 126882**

**Nhóm sinh viên thực hiện:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Hà Văn Đức(TL) | 20194510 |
| 2 | Nguyến Thế Đức | 20194515 |
| 3 | Bùi Tôn Điệp | 20194505 |

***Hà Nội, 01-2022***

Mục lục

[**I.** **Giới thiệu tổng quan** 3](#_Toc93440789)

[**II.** **Quá trình xây dựng cơ sở dữ liệu:** 5](#_Toc93440790)

[**a.** **Tìm hiểu về quy trình khám bệnh của bệnh viện** 5](#_Toc93440791)

[**b.** **Tập hợp các dữ liệu thu thập được** 5](#_Toc93440792)

[**c.** **Xây dựng sơ đồ thực thể liên kết** 6](#_Toc93440793)

[**d.** **Thiết kế cơ sở dữ liệu** 6](#_Toc93440794)

[**e.** **Nhập dữ liệu cho cơ sở dữ liệu** 7](#_Toc93440795)

[**Các câu lệnh tạo bảng:** 7](#_Toc93440796)

[**Sơ đồ quan hệ:** 11](#_Toc93440797)

[**Cách tạo dữ liệu:** 11](#_Toc93440798)

[**Kết quả** 11](#_Toc93440799)

[**III.** **Phân công công việc** 11](#_Toc93440800)

[**IV.** **Các câu truy vấn:** 13](#_Toc93440801)

1. **Giới thiệu tổng quan**

Có ý kiến cho rằng, các thông tin về bác sĩ và cơ sở y tế đầy rẫy trên Internet. Và người bệnh cũng không gặp khó khăn gì trong việc tìm hiểu thông tin và đặt lịch khám với bác sĩ, cơ sở y tế.

**Bệnh nhân có nhu cầu đặt khám thuận tiện, mọi lúc, mọi nơi**

Thông thường, khi cần đi khám với bác sĩ, bệnh nhân (hoặc người nhà) thường đến các cơ sở y tế lấy số, xếp hàng, chờ đợi, hoặc nhắn tin, gọi điện đặt lịch trước.

Tuy nhiên, lấy số xếp hàng chờ khám rất mất thời gian và mệt mỏi. Bệnh nhân ở tỉnh xa đi lại rất khó khăn, vất vả, tốn kém. Nếu là gọi điện đặt lịch thì có vẫn có một số tình huống sau đây xảy ra.

* Không gặp được lễ tân, trợ lý của bác sĩ vì không ai bắt máy
* Bệnh nhân gọi điện vào giờ nghỉ vì nhân viên chỉ làm việc vào giờ hành chính
* Bệnh nhân không nắm được lịch khám chi tiết của bác sĩ khi gọi điện đến
* Bệnh nhân không tìm được các thông tin quan trọng khác như: giá khám, giá dịch vụ, thông tin về bảo hiểm y tế, địa chỉ khám, các thay đổi lịch khám...
* Những trường hợp này đa số sẽ dấn đến các cuộc gọi liên tục qua lại giữa hai bên. Vì nhiều lý do, bệnh nhân không thể nhận điện thoại từ phòng khám vào giờ làm việc, khiến quá trình này kéo dài hơn. Và tình trạng không hẹn được kéo dài càng lâu thì càng có nhiều khả năng họ bỏ cuộc và đi tìm nơi khác.
* Những thuận lợi của đặt khám online là rất rõ ràng. Bệnh nhân có thể đặt lịch khám 24/7, mọi lúc, mọi nơi chỉ cần một máy tính hay điện thoại di động có kết nối Internet.
* Một cuộc khảo sát trong tháng 1/2017 (ở Mỹ),88% người dùng trả lời rằng họ thích đặt online hơn gọi điện.

**Bệnh nhân xem thông tin bác sĩ và đặt khám với bác sĩ chuyên khoa phù hợp**

Thông qua thiết bị có kết nối internet, bệnh nhân có thể tìm hiểu các thông tin về bác sĩ từ chuyên khoa, quá trình đào tạo, kinh nghiệm công tác, khám và chữa các bệnh phù hợp với vấn đề của người bệnh.

Với mạng lưới các bác sĩ chuyên khoa giỏi, người bệnh dễ dàng lựa chọn đúng bác sĩ chuyên khoa phù hợp, với thông tin đã xác thực và đặt lịch nhanh chóng, thuận tiện.

**Thông tin cập nhật, rõ ràng, minh bạch**

Y tế là lĩnh vực rất đặc thù, các thông tin liên quan đến khám chữa bệnh như: thông tin về bác sĩ, chuyên khoa, bệnh viện, phòng khám, giá cả, chi phí, dịch vụ y tế, trang thiết bị, qui trình, chính sách bảo hiểm y tế...chưa được chuẩn hóa, chưa rõ ràng hoặc có tính chuyên môn sâu nên người bệnh (và người dân nói chung) khó hiểu.

Mỗi khi đi khám và điều trị, người bệnh (và người thân) thường gặp nhiều khó khăn trong việc lựa chọn bác sĩ, đơn vị khám chữa bệnh phù hợp. Khó khăn trong việc tìm hiểu, xác thực thông tin, đặt lịch khám và chủ động kế hoạch đi khám.

* **Kết luận**: Với một ứng dụng đặt khám online đã cập nhật thông tin rõ ràng, qui trình dễ hiểu giúp người bệnh (và gia đình) yên tâm, tin tưởng và lựa chọn bác sĩ, cơ sở y tế phù hợp với tình trạng bệnh tật, thời gian, khả năng chi trả của mình để đi khám chữa bệnh hiệu quả.
* Chính vì những lí do trên, việc đặt lịch khám bệnh online là nhu cầu chính đáng của người dùng, đặc biệt là trong thời điểm dịch bệnh covid 19 như hiện nay khi mà việc khám bệnh trực tiếp không còn như thời điểm trước dịch bệnh nữa. Và với việc internet được phổ biến rộng rãi thì một ứng dụng đặt lịch khám bệnh là hoàn toàn khả thi và có tính thực tiễn.

1. **Quá trình xây dựng cơ sở dữ liệu:**
2. **Tìm hiểu về quy trình khám bệnh của bệnh viện**

Bằng kinh nghiệm khám bênh của bản thân và đưa người nhà đi khám kết hợp với việc tìm kiếm trên internet. Nhóm đã tổn hợp được quy trinh khám bênh chung tại các bệnh viện.Khi đi khám bệnh ở các bệnh viện, chúng ta sẽ trải qua các bước sau đây:

**Bước 1**: Lấy số thứ tự

- Bấm máy lấy số thứ tự

- Ngồi ghế và theo dõi số thứ tự

**Bước 2**: Lập hồ sơ khám & Thanh toán  
- Lập hồ sơ khám  
- Thanh toán phí khám bệnh hoặc dịch vụ yêu cầu

**Bước 3**: Bác sĩ khám, tư vấn

- Khi đến lượt, chúng ta sẽ được gọi, nếu có bảo hiểm y tế thì sẽ đưa ra cùng CCCD, nếu không có thì chỉ cần cung cấp CCCD

-Nhận số thứ tự, địa chỉ phòng khám tại tầng, số phòng.

-Chờ gọi đến lượt

**Bước 4**:

-Bác sĩ gọi và khám bệnh cho chúng ta

-Bác sĩ kê đơn thuốc

**Bước 5**: Nhận thuốc

Đối tượng BHYT: Nhận tại quầy thuốc BHYT theo từng tầng khu C

Đối tượng viện phí: Quầy thuốc tư nhân tầng 1.

* Có thể thấy rằng việc chờ đợi trực tiếp là tốn thời gian, và gây trở ngại cho người sử dụng dịch vụ trong khi hoàn toàn có thể sử dụng ứng dụng đặt lịch online để giải quyết việc này.

1. **Tập hợp các dữ liệu thu thập được**

Với việc đã nắm rõ được quy trình khám bệnh, nhóm tiến hành xây dựng sơ đồ thực thể liên kết gồm các đối tượng và quan hệ sau:

1. Các đối tượng:

* Bệnh nhân: Họ và tên, ngày tháng năm sinh, số điện thoại, email, địa chỉ, giới tinh, CCCD
* Bác sĩ: Họ và tên , ngày tháng năm sinh , chức danh , số điện thoại , email , giá khám của bác sĩ, CCCD.
* Đơn đặt lịch: Bác sĩ, ngày đặt lịch, thời điểm đặt lịch, ngày đến khám, khoảng thời gian đến khám cụ thể.
* Dịch vụ: Tên dịch vụ, giá tiền.
* Các chuyên khoa: Tên chuyên khoa, mô tả về chuyên khoa (các loại bệnh)
* Thời gian: chia nhỏ thành các khoảng thời gian (VD: 8:00-9:0)

1. Các quan hệ:

* Xác định 1 bác sĩ thuộc 1 chuyên khoa nào đấy, làm việc trong bao nhiêu năm
* Lịch trinh làm việc của bác sĩ
* Thông tin đặt lịch của bệnh nhân
* Thông tin về việc sử dụng dịch vụ của bệnh nhân

1. Diagram

   Description automatically generated**Xây dựng sơ đồ thực thể liên kết**

Link: <https://drive.google.com/file/d/1c3Wsg57yZ236YDI-hxykl8Wvt9jvMmZb/view>

1. **Thiết kế cơ sở dữ liệu**

Từ sơ đồ thực thể liên kết để xây dựng cơ sở dữ liệu gồm các bảng sau:

1. BẢNG SPECIALTY:

SpecialtyId, Name, Description

1. BẢNG DOCTORS:

DoctorId, Name, Birthday, Sex,ID\_CARD, Position, PhoneNumber, Email, Address, ExaminationFee

1. BẢNG PATIENTS :

PatientId, Name, Birthday, Sex ID\_CARD, PhoneNumber, Email, Address

1. BẢNG DOCTOR\_SPECIALTY

DCSId, DoctorId, SpecialtyId, Experience

1. BẢNG TIMETYPE

TimeTypeId, Date, TimeType

1. BẢNG SCHEDULE :

ScheduleId, DoctorId,TimeTypeId

1. BẢNG BOOKING:

BookingId, PatientId, ScheduleId. BookingDate, BookingMoment, Status

1. BẢNG DỊCH VỤ

ServiceId, Name, Type, Price

1. BẢNG SỬ DỤNG DỊCH VỤ :

USId, BookingId, ServiceId

1. **Nhập dữ liệu cho cơ sở dữ liệu**

### **Các câu lệnh tạo bảng:**

1 . BẢNG SPECIALTY:

CREATE TABLE SPECIALTY(

SpecialtyId char(6) NOT NULL,

Name varchar(50),

Description varchar(2000),

primary key (SpecialtyId)

);

2 . TẠO BẢNG DOCTORS:

CREATE TABLE DOCTORS(

DoctorId char(6) NOT NULL,

Name nvarchar(50),

Birthday date,

Sex nvarchar(5),

ID\_CARD char(12),

Position nvarchar(30),

PhoneNumber varchar(15),

Email varchar(50),

Address nvarchar(50),

ExaminationFee int,

PRIMARY KEY (DoctorId)

);

3 . TẠO BẢNG PATIENTS :

CREATE TABLE PATIENTS (

PatientId char(9) NOT NULL,

Name nvarchar(50),

Birthday date,

Sex nvarchar(5),

ID\_CARD char(12),

PhoneNumber varchar(15),

Email varchar(50),

Address nvarchar(100),

PRIMARY KEY (PatientId)

);

4. TẠO BẢNG DOCTOR\_SPECIALTY

CREATE TABLE DOCTOR\_SPECIALTY (

DCSId char(6) not null,

DoctorId char(6),

SpecialtyId char(6),

Experience int,

PRIMARY KEY (DCSId),

CONSTRAINT FK\_DOCTORID\_DS FOREIGN KEY (DoctorId)

REFERENCES DOCTORS(DoctorId)

on update cascade

on delete cascade ,

CONSTRAINT FK\_SPECIALTY\_DS FOREIGN KEY (SpecialtyId)

REFERENCES SPECIALTY(SpecialtyId)

on update cascade

on delete cascade

);

5 . BẢNG TIMETYPE

CREATE TABLE TIMETYPE(

TimeTypeId char(5) not null,

Date date,

TimeType varchar(20) not null,

primary key(TimeTypeId)

);

6. TAO BẢNG SCHEDULE :

CREATE TABLE SCHEDULE(

ScheduleId char(6) NOT NULL,

DoctorId char(6),

TimeTypeId char(5),

PRIMARY KEY (ScheduleId),

CONSTRAINT FK\_DOCTORID\_SCHEDULE FOREIGN KEY (DoctorId) REFERENCES DOCTORS(DoctorId)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_TIMETYPEID\_SCHEDULE FOREIGN KEY (TimeTypeId) REFERENCES TIMETYPE(TimeTypeId)

ON DELETE CASCADE

ON UPDATE CASCADE

);

7 . BẢNG BOOKING:

create table BOOKING(

BookingId char(8) not null,

PatientId char(9) not null,

ScheduleId char(6) not null,

BookingDate date,

BookingMoment time,

Status varchar(20),

PRIMARY KEY(BookingId),

constraint FK\_SCHEDULEID\_BOOKING foreign key(ScheduleId) references SCHEDULE(ScheduleId)

on delete cascade

on update cascade,

constraint FK\_PATIENTID\_BOOKING foreign key(PatientId) references PATIENTS(PatientId)

on delete cascade

on update cascade

);

8.BẢNG DỊCH VỤ

CREATE TABLE SERVICES(

ServiceId char(5) not null,

Name varchar(200),

Type varchar(100),

Price int,

primary key(ServiceId)

);

9. BẢNG SỬ DỤNG DỊCH VỤ :

CREATE TABLE USED\_SERVICES(

USId char(8) not null,

BookingId char(8),

ServiceId char(5),

primary key(USId),

CONSTRAINT FK\_BOOKINGID\_USED\_SERVICES FOREIGN KEY(BookingId)

REFERENCES BOOKING(BookingId)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_SERVICES\_USED\_SERVICES FOREIGN KEY(ServiceId)

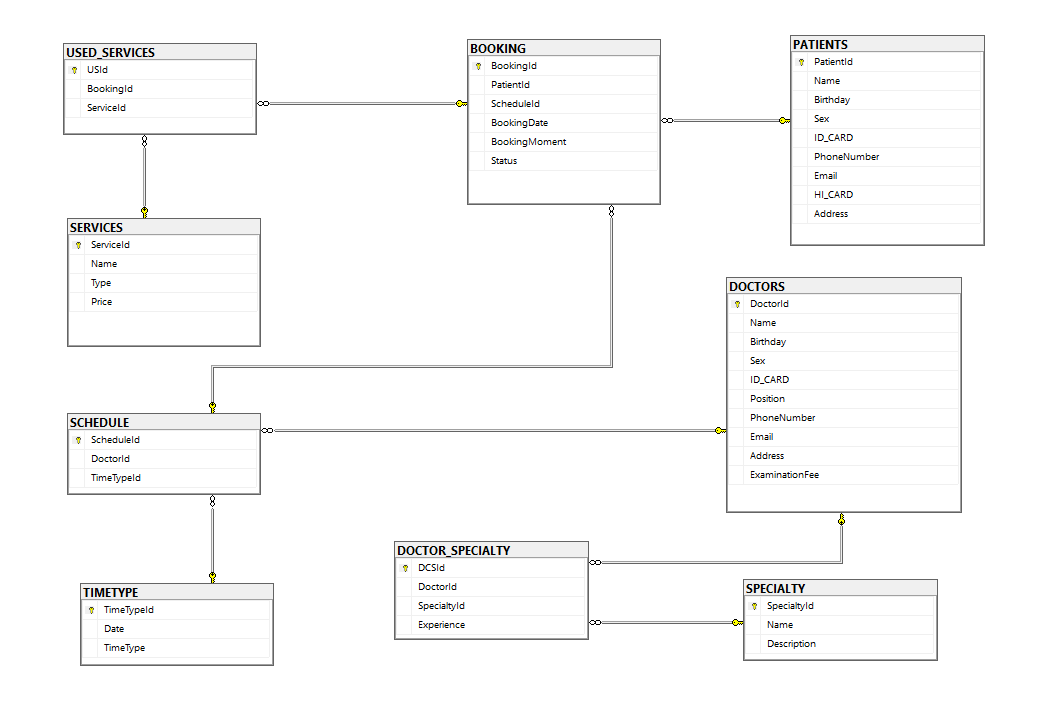
REFERENCES SERVICES(ServiceId)

ON DELETE CASCADE

ON UPDATE CASCADE

);

### **Sơ đồ quan hệ:**

****

### **Cách tạo dữ liệu:**

* 1 triệu bản ghi dữ liệu bệnh nhân: Sử dụng exel để sinh ngẫu nhiên các trường
* Các bảng khác cũng sử dụng exel để cùng sinh ngẫu nhiên và nhập dữ liệu.
* Sử dụng file .csv để load vào mysql

### **Kết quả**

* Sau nhiều tuần nỗ lực làm việc hết mình, nhóm đã hoàn thành cơ sở dữ liệu đúng hạn

1. **Phân công công việc**

**Làm việc nhóm thông qua phần mềm MS TEAMS hàng tuần, có báo cáo tiến độ.**

* Lên ý tưởng project: Cùng thảo luận
* Khảo sát thực tế: Bùi Tôn Điệp
* Tổng hợp dữ liệu thu được: Nguyễn Thế Đức
* Vẽ sơ đồ thực thể liên kết: Bùi Tôn Điệp
* Tạo bảng: Hà Văn Đức
* Tạo dữ liệu: Nguyễn Thế Đức, Hà Văn Đức, Bùi Tôn Điệp
* insert dữ liệu: Hà Văn Đức insert, cùng nhau sửa lỗi.
* Viết báo cáo, làm slide: share màn hình cùng nhau làm.
* Thuyết trình: Hà Văn Đức, Nguyễn Thế Đức
* Thảo luận đánh giá và rút kinh nghiệm hàng tuần: Cùng thảo luận

1. **Các câu truy vấn:**
2. **Các câu truy vấn của Hà Văn Đức**

1. Cài đặt Trigger cho bảng DOCTORS để phản hồi lại một sự kiện insert , update , delete

Tạo bảng DOCTORS\_AUDIT :

CREATE TABLE DOCTORS\_AUDIT (

Id INT AUTO\_INCREMENT PRIMARY KEY,

DoctorId char(6) NOT NULL,

Name VARCHAR(50) NOT NULL,

ID\_CARD char(12) NOT NULL,

Position nvarchar(30),

ExaminationFee int,

ChangeDate DATETIME DEFAULT NULL,

action VARCHAR(50) DEFAULT NULL

);

* + Trigger trước khi update

DELIMITER $$

CREATE TRIGGER before\_DOCTORS\_update

BEFORE UPDATE ON DOCTORS FOR EACH ROW

BEGIN

INSERT INTO DOCTORS\_AUDIT

SET action = 'update',

DoctorId = OLD.DoctorId,

Name = OLD.Name,

ID\_CARD = OLD.ID\_CARD,

Position = OLD.Position,

ExaminationFee = OLD. ExaminationFee,

ChangeDate = NOW();

END$$

DELIMITER ;

* Sau khi chạy cài đặt trigger thành công , ta kiểm tra hoạt động của trigger bằng cách sau:

Update dữ liệu của bảng DOCTORS

UPDATE thông tin chức danh của bác sĩ có DoctorId = ‘DT0032’ từ ‘Phó giáo sư’ thành ‘Giáo sư’

UPDATE DOCTORS

SET Position = 'Giáo sư'

WHERE DoctorId = 'DT0032';

* Xem phản hồi được lưu tại bảng DOCTORS\_AUDIT:

SELECT \* FROM DOCTORS\_AUDIT;

A screenshot of a computer

Description automatically generated with medium confidence

* + Trigger sau khi insert

DELIMITER $$

CREATE TRIGGER after\_DOCTORS\_insert

AFTER INSERT ON DOCTORS FOR EACH ROW

BEGIN

INSERT INTO DOCTORS\_AUDIT

SET action = 'insert',

DoctorId = NEW.DoctorId,

Name = NEW.Name,

ID\_CARD = NEW.ID\_CARD,

Position = NEW.Position,

ExaminationFee = NEW. ExaminationFee,

ChangeDate = NOW();

END$$

DELIMITER ;

* INSERT 1 bản ghi sau vào bảng DOCTORS để kiểm tra :

INSERT INTO DOCTORS

VALUES

(‘DT0033’,N’Nguyễn Hoàng Anh’,’1995-02-18’,’Nam’ , ‘001383214568’ , N‘Thạc sĩ’ , ‘0963021448’, ‘hoanganhdeptraicool@gmail.com’,’PhườngTương Mai – Quận Hoàng Mai – Hà Nội’ ,250000);

Xem phản hồi tại bảng DOCTORS\_AUDIT:

SELECT \* FROM DOCTORS\_AUDIT;

A screenshot of a computer

Description automatically generated with medium confidence

* + Trigger trước khi delete

DELIMITER $$

CREATE TRIGGER before\_DOCTORS\_delete

BEFORE DELETE ON DOCTORS FOR EACH ROW

BEGIN

INSERT INTO DOCTORS\_AUDIT

SET action = 'delete',

DoctorId = OLD.DoctorId,

Name = OLD.Name,

ID\_CARD = OLD.ID\_CARD,

Position = OLD.Position,

ExaminationFee = OLD. ExaminationFee,

ChangeDate = NOW();

END$$

DELIMITER ;

DELETE Bác sĩ có DoctorId = ‘DT0033’:

DELETE FROM DOCTORS

WHERE DoctorId = 'DT0033';

Xem phản hồi tại bảng DOCTORS\_AUDIT:

SELECT \* FROM DOCTORS\_AUDIT;

A screenshot of a computer

Description automatically generated with medium confidence

2 . Tạo PROCEDURE chỉ ra số đơn đặt lịch theo mã số bác sĩ , ngày bệnh nhân đến khám và trạng thái của đơn đặt lịch đó

DELIMITER $$

CREATE PROCEDURE GetBookingCountByDoctor\_InADate( IN DoctorId CHAR(6), IN Date DATE , IN Status VARCHAR(20), OUT total INT)

BEGIN

SELECT COUNT(BookingId)

INTO total

FROM BOOKING B, SCHEDULE S , TIMETYPE T

WHERE B.ScheduleId = S.ScheduleId AND S.TimeTypeId = T.TimeTypeId AND S.DoctorId = DoctorId AND T.Date = Date AND B.Status = Status;

END$$

DELIMITER ;

* Xác định số đơn đã khám xong của bác sĩ ‘DT0001’ vào ngày 18-01-2022

CALL GetBookingCountByDoctor\_InADate('DT0001','2022-01-18','Đã khám xong',@total);

Text

Description automatically generated

* Kiểm tra lại kết quả :

select BookingId , PatientId , S.ScheduleId , DoctorId , S.TimeTypeId , T.Date , T.TimeType ,BookingDate , BookingMoment , Status FROM BOOKING B, SCHEDULE S , TIMETYPE T

WHERE B.ScheduleId = S.ScheduleId AND S.TimeTypeId = T.TimeTypeId AND DoctorId = 'DT0001' AND T.Date = '2022-01-18' AND B.Status = 'Đã khám xong';

Graphical user interface

Description automatically generated with medium confidence

3 . Cài đặt INDEX lên thuộc tính Name của bảng PATIENTS

* Trước khi cài INDEX:

select \* from PATIENTS

where Name = 'Âu Bích Huyền';

A screenshot of a computer

Description automatically generated with medium confidence

explain SELECT \* FROM PATIENTS WHERE Name = 'Âu Bích Huyền';

Graphical user interface

Description automatically generated with low confidence

* Sau khi cài INDEX:

A screenshot of a computer

Description automatically generated with medium confidence

4 . Đưa ra tổng số lượt bệnh nhân đã đến khám bệnh vào ngày 19-01-2022 theo từng chuyên khoa sắp xếp theo thứ tự giảm dần số lượng lượt đến khám của bệnh nhân

select SP.SpecialtyId , SP.Name , count(BookingId) as SoLuotKham

from BOOKING B , SCHEDULE SC , DOCTORS D , DOCTOR\_SPECIALTY DS , SPECIALTY SP , TIMETYPE T

where B.ScheduleId = SC.ScheduleId and SC.DoctorId = D.DoctorId and D.DoctorId = DS.DoctorId and DS.SpecialtyId = SP.SpecialtyId and SC.TimeTypeId = T.TimeTypeId

and T.Date = '2022-01-19' and B.Status = 'Đã khám xong'

group by SP.SpecialtyId

order by SoLuotKham desc;

Text

Description automatically generated

5 . Full text search : Bệnh nhân muốn tìm kiếm thêm những bệnh liên quan đến dạ dày qua từ khóa 'dạ dày'

ALTER TABLE SPECIALTY

ADD FULLTEXT(Description);

SELECT Name , Description

FROM SPECIALTY

WHERE MATCH (Description)

AGAINST ('Dạ dày');

A picture containing background pattern

Description automatically generated

6 . Đưa ra bác sĩ có giá khám rẻ nhất thuộc chuyên khoa mắt và lịch làm việc của bác sĩ đó trong ngày 21-01-2022

select D.DoctorId , D.Name , ExaminationFee , T.Date , T.TimeType

from DOCTORS D, DOCTOR\_SPECIALTY DS , SPECIALTY SP, SCHEDULE SC , TIMETYPE T

WHERE D.DoctorId = DS.DoctorId AND DS.SpecialtyId = SP.SpecialtyId AND D.DoctorId = SC.DoctorId

AND SC.TimeTypeId = T.TimeTypeId AND SP.Name like '%Chuyên khoa mắt%'

AND T.Date = '2022-01-21' AND ExaminationFee = (select min(ExaminationFee) from DOCTORS D, DOCTOR\_SPECIALTY DS , SPECIALTY SP WHERE D.DoctorId = DS.DoctorId AND DS.SpecialtyId = SP.SpecialtyId

AND SP.Name like '%Chuyên khoa mắt%');

Text

Description automatically generated

7 . Đưa ra thông tin về mã bác sĩ , tên bác sĩ , chuyên khoa , Ngày đặt , thời điểm đặt , trạng thái của những lần đặt lịch khám bệnh của người bệnh có số điện thoại 0826438618 sắp xếp theo ngày đặt lịch , thời điểm đặt lịch

select D.DoctorId , D.Name , SP.Name , BookingDate , BookingMoment , Status Status from PATIENTS P , BOOKING B , DOCTORS D , SCHEDULE SC , DOCTOR\_SPECIALTY DS , TIMETYPE T , SPECIALTY SP where P.PatientId = B.PatientId and B.ScheduleId = SC.ScheduleId and SC.DoctorId = D.DoctorId and SC.TimeTypeId = T.TimeTypeId and D.DoctorId = DS.DoctorId and DS.SpecialtyId = SP.SpecialtyId and P.PhoneNumber = '0826438618' order by BookingDate , BookingMoment;

A screenshot of a computer screen

Description automatically generated with medium confidence

8 . Đưa ra thông tin bác sĩ nào không có lịch khám bệnh vào 8:30 - 9:00 ngày

17-01-2022

select DoctorId , Name , PhoneNumber , ExaminationFee from DOCTORS WHERE DoctorId NOT IN (Select DoctorId from SCHEDULE SC , BOOKING B , TIMETYPE T WHERE SC.ScheduleId = B.ScheduleId AND SC.TimeTypeId = T.TimeTypeId AND T.Date = '2022-01-17' AND T.TimeType = '8:30 - 9:00');

Text

Description automatically generated

9 . Đưa ra mã , tên , số đơn đã hủy của bệnh nhân đã hủy đơn đặt lịch nhiều nhất từ ngày 14-01-2022 đến ngày 16-01-2022(dựa trên BookingDate)

select P.PatientId , P.Name , count(BookingId) AS SoDonHuy from PATIENTS P , BOOKING B WHERE P.PatientId = B.PatientId AND BookingDate BETWEEN '2022-01-14' AND '2022-01-16' AND Status = 'Đã hủy' GROUP BY P.PatientId having SoDonHuy >= ALL(select count(BookingId) from BOOKING B where Status = 'Đã hủy'AND BookingDate BETWEEN '2022-01-14' AND '2022-01-16' group by PatientId);

Text

Description automatically generated

10 . Đưa ra mã số , tên , giá khám của 3 bác sĩ có lịch hẹn mới nhiều nhất trong ngày 22-01-2022

select D.DoctorId , D.Name , ExaminationFee , count(BookingId) AS SoLichHenMoi from DOCTORS D , BOOKING B , SCHEDULE SC , TIMETYPE T where D.DoctorId = SC.DoctorId AND B.ScheduleId = SC.ScheduleId AND SC.TimeTypeId = T.TimeTypeId AND Status = 'Lịch hẹn mới' GROUP BY DoctorId ORDER BY SoLichHenMoi DESC LIMIT 3;

Text

Description automatically generated

1. **Các câu truy vấn của Nguyễn Thế Đức:**

1. Đưa ra thông tin bệnh nhân đã khám bệnh tại bệnh viện và số lần khám( tức Status của BOOKNG là đã khám xong), sắp xếp theo thứ tự giảm dần của số lần khám.

select PATIENTS.\*, count(BookingId) as solankham from PATIENTS, BOOKING where PATIENTS.PatientId=BOOKING.PatientId and Status = 'Đã khám xong' group by PatientId order by count(BookingId) desc ;

Timeline

Description automatically generated

2. Đưa ra DoctorId, họ tên, chuyên khoa, lịch làm việc các bác sĩ.

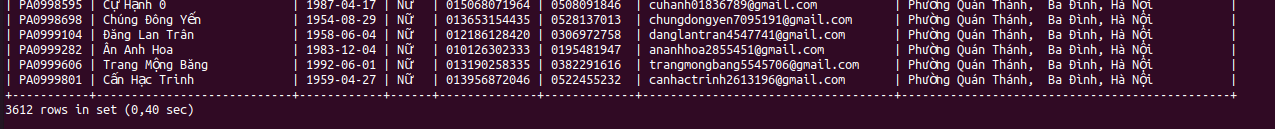
select d.DoctorId, d.Name,sp.name, concat("Ngày:", Date," Giờ:", TimeType) from DOCTORS d, SCHEDULE s, TIMETYPE t, SPECIALTY sp, DOCTOR\_SPECIALTY dsp where d.DoctorId = s.DoctorId and t.TimeTypeId = s.TimeTypeId and dsp.SpecialtyId = sp.SpecialtyId and d.DoctorId = dsp.DoctorID;

Text

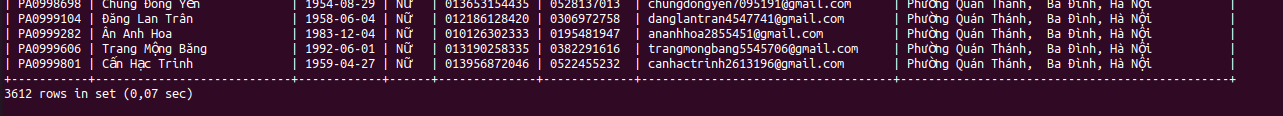
Description automatically generated with medium confidence

3. Tạo chỉ mục để truy vấn thông tin những bệnh nhân ở "Phường Quán Thánh, Ba Đình, Hà Nội", so sánh độ hiệu quả của câu truy vấn với không dùng chỉ mục.

🡪 Câu truy vấn: select \* from PATIENTS where Address = "Phường Quán Thánh, Ba Đình, Hà Nội ";

Trước khi tạo chỉ mục:0.41 sec

Tạo chỉ mục: create index Address on PATIENTS(Address);

Sau khi có index:

Thời gian chạy: 0.02 sec - 0.05 sec

4. Tạo TRIGGER lưu trữ sự kiện delete của bảng BOOKING

create table booking\_audit(

id int auto\_increment primary key,

BookingId char(8) not null,

PatientId char(9) not null,

ScheduleId char(6) not null,

BookingDate date,

BookingMoment time,

Status varchar(20),

changdate datetime default null,

action varchar(50) default null

);

DELIMITER $$

CREATE TRIGGER after\_booking\_delete

AFTER DELETE ON BOOKING

FOR EACH ROW

BEGIN

INSERT INTO booking\_audit

set action = 'delete',

BookingId = old.BookingId,

PatientId = old.PatientId,

ScheduleId = old.ScheduleId,

BookingDate = old.BookingDate,

BookingMoment = old.BookingMoment,

Status = old.Status,

changdate = NOW();

END$$

DELIMITER ;

delete from BOOKING where BookingId = 'BK001100';

Text

Description automatically generated

5. In ra số tiền cần trả cho mỗi lần khám của các bệnh nhân.

select p.\*, u.BookingId, SUM(price) from USED\_SERVICES u, SERVICES s, BOOKING b, PATIENTS p where u.ServiceId = s.ServiceId and b.BookingId = u.BookingId and p.PatientId = b.PatientId group by u.BookingId;

Text

Description automatically generated

6. Tạo procedure để tính tiền sử dụng dịch vụ của bệnh nhân trong các lần đến khám tại bệnh viện.

procedure cần tạo: calculatePrice(PatientId);

DELIMITER $$

create procedure calculatePrice(IN Order\_PatientId char(9))

BEGIN

select p.\*, u.BookingId, SUM(price) from USED\_SERVICES u, SERVICES s, BOOKING b, PATIENTS p where u.ServiceId = s.ServiceId and b.BookingId = u.BookingId and p.PatientId = b.PatientId and b.PatientId = Order\_PatientId group by u.BookingId;

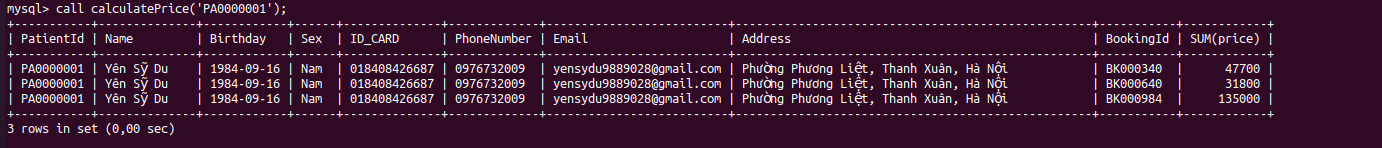
END$$

DELIMITER ;

câu lệnh test procedure:

call calculatePrice('PA0000001');

call calculatePrice('PA0000002');



7. Tạo function cho biết rank của bác sĩ, nếu kinh nghiệm >=10 năm thì là rank gold, 5<= số năm kinh nghiệm < 10 thì là rank silver, <5 năm sẽ là bronze

DELIMITER $$

CREATE FUNCTION DoctorLevel(

Experience int

)

RETURNS VARCHAR(20)

DETERMINISTIC

BEGIN

DECLARE DoctorLevel VARCHAR(20);

IF Experience >= 10 THEN

SET DoctorLevel = 'GOLD';

ELSEIF (Experience >= 5 AND

Experience < 10) THEN

SET DoctorLevel = 'SILVER';

ELSEIF Experience < 5 THEN

SET DoctorLevel = 'BRONZE';

END IF;

-- return the DOCTOR level

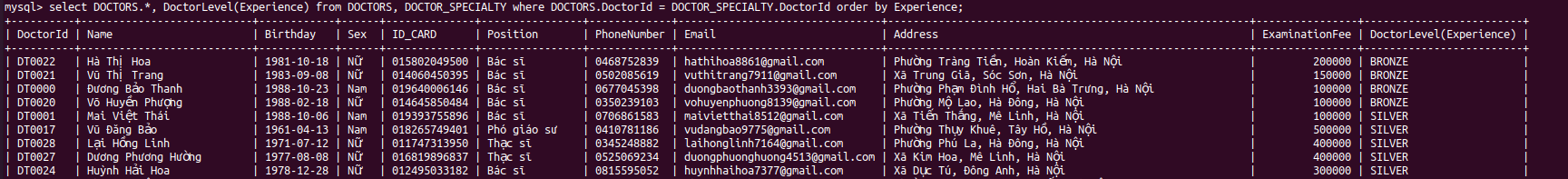
RETURN (DoctorLevel);

END$$

DELIMITER ;

Câu lệnh test Function:

select DOCTORS.\*, DoctorLevel(Experience) from DOCTORS, DOCTOR\_SPECIALTY where DOCTORS.DoctorId = DOCTOR\_SPECIALTY.DoctorId order by Experience;



8. Tìm kiếm các dịch vụ liên quan đến xương khớp.

alter table SERVICES ADD FULLTEXT(Name);

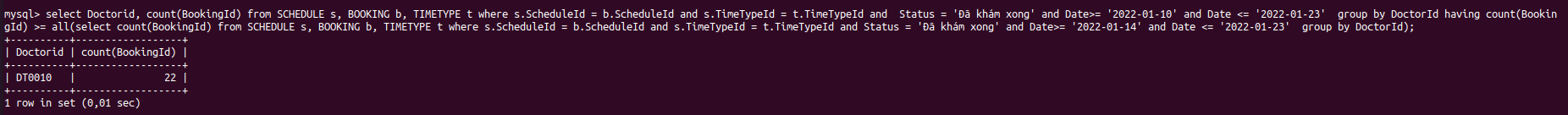
select Name from SERVICES where MATCH(Name) against ('Xương,Khớp');

A picture containing graphical user interface

Description automatically generated

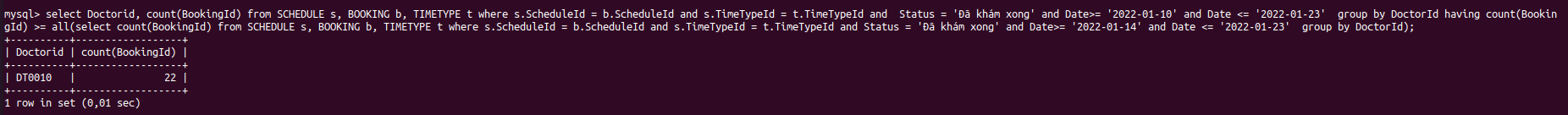
9. In ra thông tin bác sĩ thực hiện khám bệnh nhiều nhất của mỗi chuyên khoa trong tuần từ 17/01/2022 đến 23/01/2022

select Doctorid, count(BookingId) from SCHEDULE s, BOOKING b, TIMETYPE t where s.ScheduleId = b.ScheduleId and s.TimeTypeId = t.TimeTypeId and Status = 'Đã khám xong' and Date>= '2022-01-10' and Date <= '2022-01-23' group by DoctorId having count(BookingId) >= all(select count(BookingId) from SCHEDULE s, BOOKING b, TIMETYPE t where s.ScheduleId = b.ScheduleId and s.TimeTypeId = t.TimeTypeId and Status = 'Đã khám xong' and Date>= '2022-01-14' and Date <= '2022-01-23' group by DoctorId);



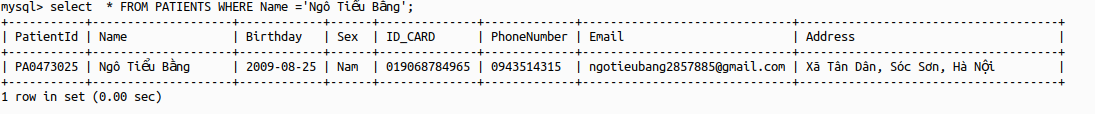
10. Đưa ra khoảng thời gian được đặt lịch nhiều nhất trong tuần từ 17/01/2022 đến 23/01/2022

select TimeType,count(BookingId) from SCHEDULE s, BOOKING b, TIMETYPE t where s.ScheduleId = b.ScheduleId and s.TimeTypeId = t.TimeTypeId and Date>= '2022-01-14' and Date <= '2022-01-23' group by TimeType having count(BookingId) >= all(select count(BookingId) from SCHEDULE s, BOOKING b, TIMETYPE t where s.ScheduleId = b.ScheduleId and s.TimeTypeId = t.TimeTypeId and Date>= '2022-01-14' and Date <= '2022-01-23' group by TimeType);



1. Các câu truy vấn của Bùi Tôn Điệp:
2. Đưa ra thông tin bệnh nhân có tên NGô Tiểu Bằng.

select \* FROM PATIENTS WHERE Name ='Ngô Tiểu Bằng';



1. Đưa ra thông tin trạng thái đặt lịch của bệnh nhân Kim Thế Tâm.

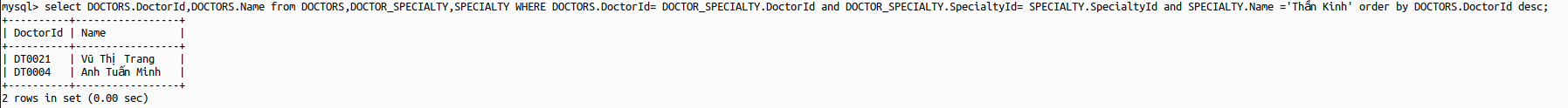
select PATIENTS.PatientId,Name, Status from PATIENTS,BOOKING WHERE PATIENTS.PatientId=BOOKING.PatientId and Name= 'Kim Thế Tâm';

Text

Description automatically generated

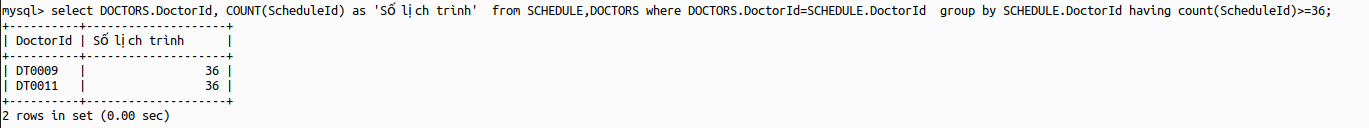
1. Sắp xếp Giảm dần các Bác sĩ làm việc ở khoa THẦN kinh.

select DOCTORS.DoctorId,DOCTORS.Name from DOCTORS,DOCTOR\_SPECIALTY,SPECIALTY WHERE DOCTORS.DoctorId= DOCTOR\_SPECIALTY.DoctorId and DOCTOR\_SPECIALTY.SpecialtyId= SPECIALTY.SpecialtyId and SPECIALTY.Name ='Thần Kinh' order by DOCTORS.DoctorId desc;



1. Đưa ra thông tin bác sĩ có ít nhất 36 lịch trình khám bệnh.

select DOCTORS.DoctorId, COUNT(ScheduleId) as 'Số lịch trình' from SCHEDULE,DOCTORS where DOCTORS.DoctorId=SCHEDULE.DoctorId group by SCHEDULE.DoctorId having count(ScheduleId)>=36;



1. Cài đặt index vào trường PhoneNumber bảng PATIENTS.

Trước khi cài index:

select \* from PATIENTS WHERE Name= 'Vàng Hạnh Trà';

explain select \* from PATIENTS WHERE Name= 'Vàng Hạnh Trà';

Cài index:

create index idx\_Name on PATIENTS(Name);

Sau khi cài index:

select \* from PATIENTS WHERE Name= 'Vàng Hạnh Trà';

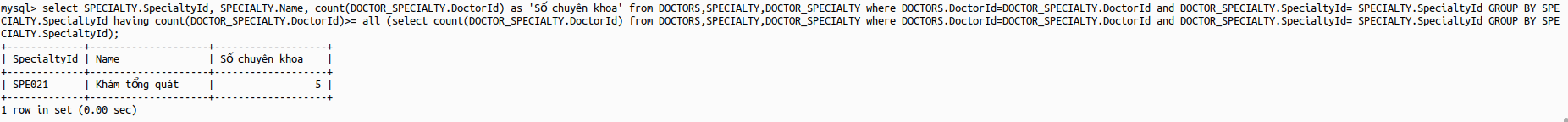
explain select \* from PATIENTS WHERE Name= 'Vàng Hạnh Trà';

Graphical user interface, application, table, Excel

Description automatically generated

1. Đưa ra tên chuyên khoa có nhiều bác sỹ nhất.

select SPECIALTY.SpecialtyId, SPECIALTY.Name, count(DOCTOR\_SPECIALTY.DoctorId) as 'Số chuyên khoa' from DOCTORS,SPECIALTY,DOCTOR\_SPECIALTY where DOCTORS.DoctorId=DOCTOR\_SPECIALTY.DoctorId and DOCTOR\_SPECIALTY.SpecialtyId= SPECIALTY.SpecialtyId GROUP BY SPECIALTY.SpecialtyId having count(DOCTOR\_SPECIALTY.DoctorId)>= all (select count(DOCTOR\_SPECIALTY.DoctorId) from DOCTORS,SPECIALTY,DOCTOR\_SPECIALTY where DOCTORS.DoctorId=DOCTOR\_SPECIALTY.DoctorId and DOCTOR\_SPECIALTY.SpecialtyId= SPECIALTY.SpecialtyId GROUP BY SPECIALTY.SpecialtyId);



1. Đưa ra bảng thay đổi trạng thái đặt lịch sau khi xóa đơn đặt BK000140.

CREATE TABLE BOOKING\_audit (

BookingId char(8) not null,

PatientId char(9) not null,

ScheduleId char(6) not null,

BookingDate date,

BookingMoment time,

Status varchar(20),

PRIMARY KEY(BookingId),

changedat DATETIME DEFAULT NULL,

action VARCHAR(50) DEFAULT NULL

);

DELIMITER $$

CREATE TRIGGER BOOKING\_update

BEFORE DELETE ON BOOKING

FOR EACH ROW

BEGIN

INSERT INTO BOOKING\_audit

SET action = 'delete',

BookingId =OLD.BookingId,

PatientId= OLD.PatientId,

ScheduleId=OLD.ScheduleId,

BookingDate=OLD.BookingDate,

BookingMoment= OLD.BookingMoment,

Status= OLD.Status,

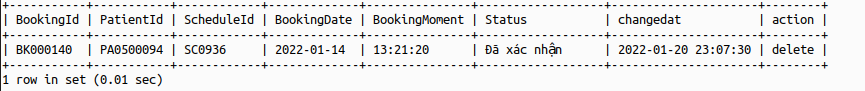
changedat = NOW();

END$$

DELIMITER ;

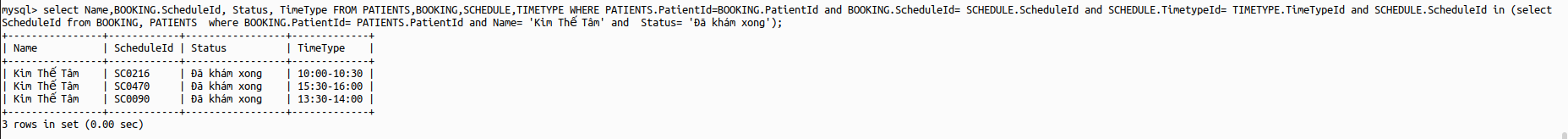
DELETE from BOOKING where BookingId='BK000140';

select\* from BOOKING\_audit;



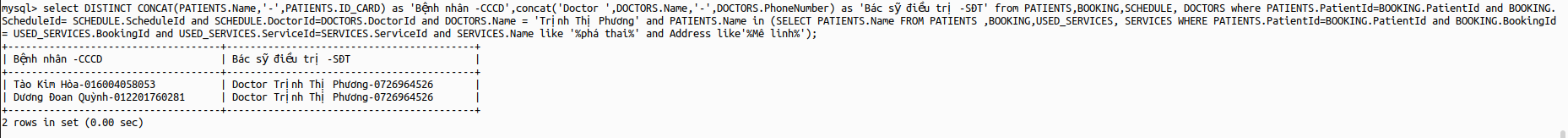
8. Đưa ra các khoảng thời gian mà bệnh nhân Kim Thế Tâm đã khám xong.

select Name,BOOKING.ScheduleId, Status, TimeType FROM PATIENTS,BOOKING,SCHEDULE,TIMETYPE WHERE PATIENTS.PatientId=BOOKING.PatientId and BOOKING.ScheduleId= SCHEDULE.ScheduleId and SCHEDULE.TimetypeId= TIMETYPE.TimeTypeId and SCHEDULE.ScheduleId in (select ScheduleId from BOOKING, PATIENTS where BOOKING.PatientId= PATIENTS.PatientId and Name= 'Kim Thế Tâm' and Status= 'Đã khám xong');



9. Đưa ra tên các bệnh nhân ở huyện Mê Linh hà nội đã đăng ký sử dụng dịch vụ phá thai bởi bác sỹ Trịnh Thị Phương.

select DISTINCT CONCAT(PATIENTS.Name,'-',PATIENTS.ID\_CARD) as 'Bệnh nhân -CCCD',concat('Doctor ',DOCTORS.Name,'-',DOCTORS.PhoneNumber) as 'Bác sỹ điều trị -SĐT' from PATIENTS,BOOKING,SCHEDULE, DOCTORS where PATIENTS.PatientId=BOOKING.PatientId and BOOKING.ScheduleId= SCHEDULE.ScheduleId and SCHEDULE.DoctorId=DOCTORS.DoctorId and DOCTORS.Name = 'Trịnh Thị Phương' and PATIENTS.Name in (SELECT PATIENTS.Name FROM PATIENTS ,BOOKING,USED\_SERVICES, SERVICES WHERE PATIENTS.PatientId=BOOKING.PatientId and BOOKING.BookingId= USED\_SERVICES.BookingId and USED\_SERVICES.ServiceId=SERVICES.ServiceId and SERVICES.Name like '%phá thai%' and Address like'%Mê linh%');



1. Tính xem có bao nhiêu bệnh nhân sử dụng dịch vụ Hồng ngoại trong từng khoảng thời gian.

select TimeType ,count(PATIENTS.PatientId) as 'Sô bệnh nhân sử dụng dịch vụ Hồng Ngoại' from TIMETYPE, SCHEDULE,BOOKING,PATIENTS where TIMETYPE.TimeTypeId= SCHEDULE.TimeTypeId and SCHEDULE.ScheduleId=BOOKING.ScheduleId and BOOKING.PatientId= PATIENTS.PatientId and PATIENTS.PatientId in (select PATIENTS.PatientId from SERVICES, USED\_SERVICES, BOOKING,PATIENTS WHERE SERVICES.ServiceId= USED\_SERVICES.ServiceId and USED\_SERVICES.BookingId=BOOKING.BookingId and BOOKING.PatientId= PATIENTS.PatientId and SERVICES.Name='Hồng ngoại') group by TimeType;

Graphical user interface, application

Description automatically generated with medium confidence