**Final Report**

Zhili Hu

The Pennsylvania State University

School of Electrical Engineering and Computer Science

CMPSC 431W: Database Management Systems

Steven Shaffer

December, 12 2022

**Introduction**

This report is based on midterm domain: **Hospital**

The purpose of this database is building a system for hospital to save information for doctors and patients, and realize some simple application scenarios. The specific description is in the following problem description section.

**Database:**

Database includes 6 tables: doctors, specialization\_department, bill, bill\_doctor, patients and patient\_history.

Table doctor includes 6 fields. They are doctor id(did), first name(fname), middle name(mname), last name(lname), specialization(spec), phone number(phone). This table stores information of doctors.

Table specialization\_department includes 2 fields. They are specialization(spec) and department(depart). This table can check the different specialization by specific department, or check the department of specific specialization.

Table bill includes following 5 fields: bill id(bid), date of bill(date), patient id(pid), if patient arrive(arrive), charge amount(charge). This table stores information of daily bills of all patients.

Table bill\_doctor includes 2 fields that bill id(bid) and doctor id(did). This table can check the doctor(s) of a specific bill, or all the related bills of specific doctor.

Table patients includes 8 fields. Patient id(pid), first name(fname), middle name(mname), last name(lname), age(age), gender(gender), insurance id(iid), phone number(phone). This table stores information of patients.

Table patient\_history includes 3 fields. They are patient id(pid), date(date), medical history(history). This table is for doctors to record the medical information of the patient. In the future, other doctors can check the table by pid to know the patient's medical history to help with the visit.

**PHP application:**

Here are 6 different applications in the system. The detail will be introduced in the following part.

**Description of the Problem to be Solved**

The main purpose of this database system is to help hospital store and provide information for doctors and patients. Except the basic personal information (name, age, gender, etc.) the system stores, this database has application of:

1.Add/delete/update information of doctors

2.Generate complete bill information for a patient

3.Save medical history for a patient after the treatment

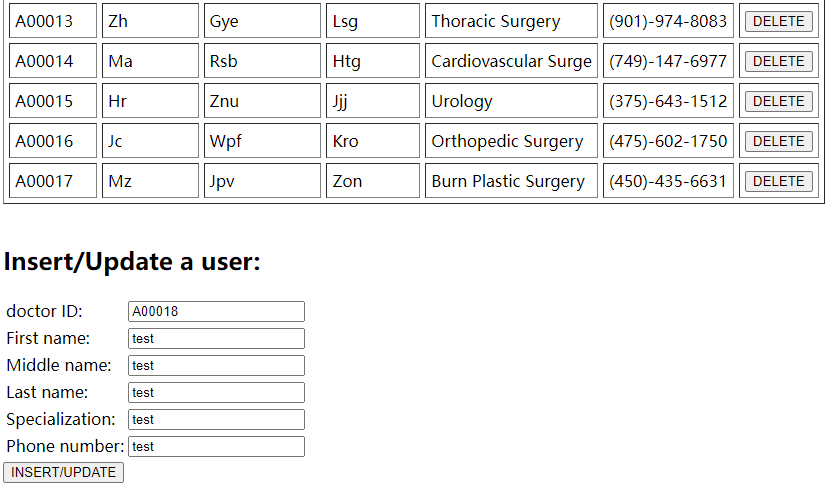
4.Generate monthly income for a department

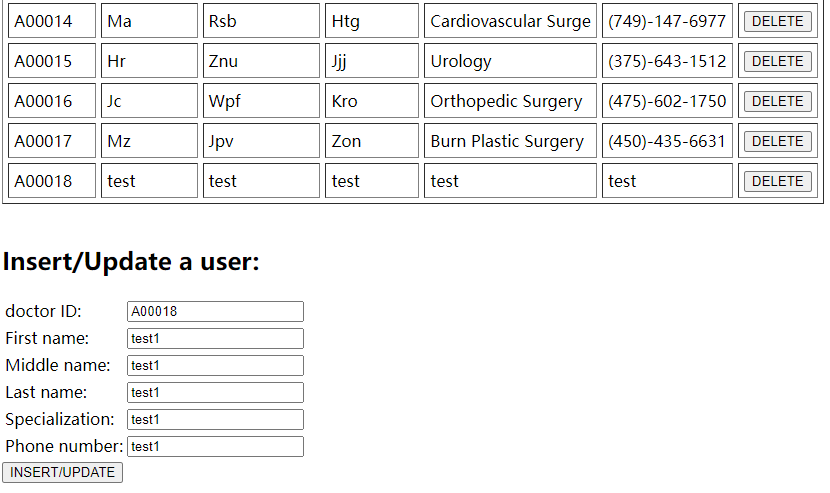
5.Generate an Annual Billing Report(sorting by date)

6.Cancle an appointment by a bill id (charge 5$) (roll back)

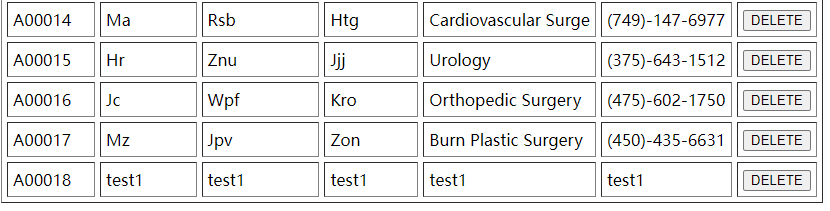
1.Add/delete/update a doctor.

insert into doctors values('A00018','test','test','test','Burn Plastic Surgery','(123)-456-7890'); 

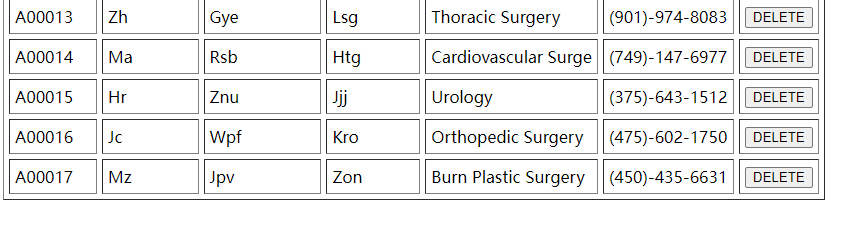


Click the button in the bottom when a new doctor id input, it inserts.

Then Click a button in the bottom when a doctor id already in the database, it updates.



Then click the DELETE button in the right, it deletes.



2.Show the patient C00083’s bill that contain bill id, date, insurance id, doctor’s first name, last name, charge amount and patient’s first name, last name.

SELECT b.bid as 'bill id', b.date as date, p.iid as 'insurance id', GROUP\_CONCAT(DISTINCT d.fname) as 'doctor first name', GROUP\_CONCAT(DISTINCT d.lname) as 'doctor last name', b.charge as 'charge amount', p.fname as 'patient first name', p.lname as 'patient last name'

FROM doctors d, bill\_doctor bd, bill b, patients p

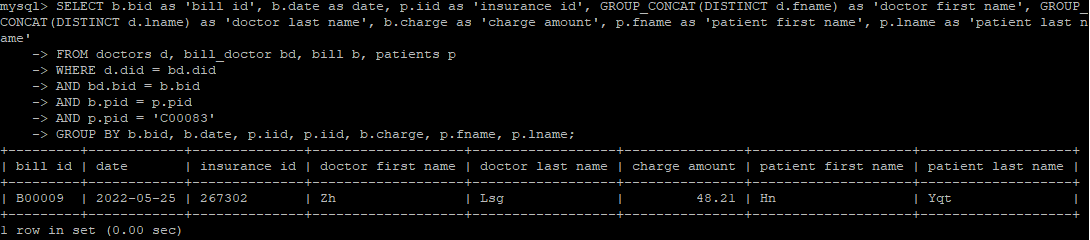
WHERE d.did = bd.did

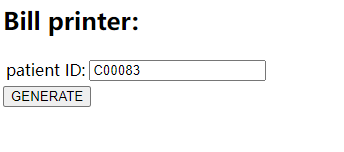
AND bd.bid = b.bid

AND b.pid = p.pid

AND p.pid = 'C00083'

GROUP BY b.bid, b.date, p.iid, p.iid, b.charge, p.fname, p.lname;





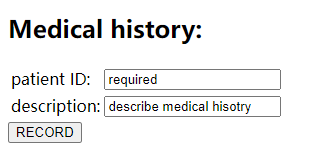
Input patient id and click generate button, it generates a bill with all bill information

(Time limit and code ability limited do not complete PHP code part)

3.Save a medical history for patient C00153

insert into patient\_history values('C00153',' 2022-4-8','test');





Input patient id and description then click generate button. it records a patient id, current date and description into patient\_history table.

(Time limit and code ability limited do not complete PHP code part)

4.Given a 2022 November income of Internal Medicine department.

SELECT SUM(b.charge) as '2022 Nov income of Internal Medicine department'

FROM bill b, bill\_doctor bd, doctors d, specialization\_department s

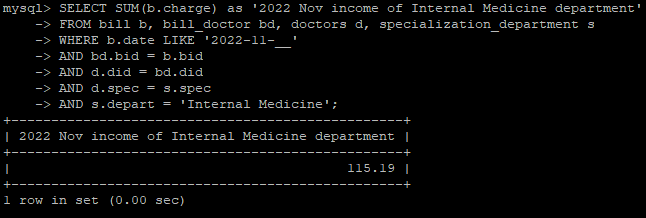
WHERE b.date LIKE '2022-11-\_\_'

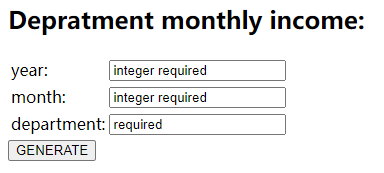
AND bd.bid = b.bid

AND d.did = bd.did

AND d.spec = s.spec

AND s.depart = 'Internal Medicine';





Input year, month department then click generate button, it generates a report about the department income in that month.

(Time limit and code ability limited do not complete PHP code part)

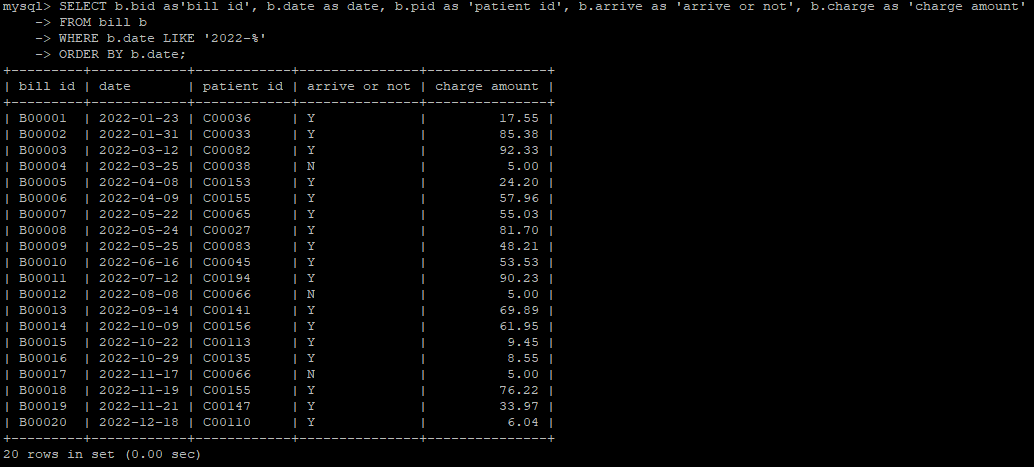
5. Generate Annual Billing Report of 2022, sorting by date

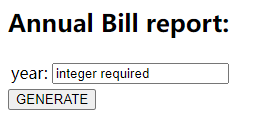
SELECT b.bid as'bill id', b.date as date, b.pid as 'patient id', b.arrive as 'arrive or not', b.charge as 'charge amount'

FROM bill b

WHERE b.date LIKE '2022-%'

ORDER BY b.date;





Input year then click generate button. It generates a report about all the information of fields of the table bills.

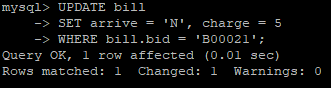
(Time limit and code ability limited do not complete PHP code part)

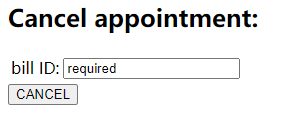
6.Cancle an appointment by a bill id

UPDATE bill

SET arrive = 'N', charge = 5

WHERE bill.bid = 'B00021';



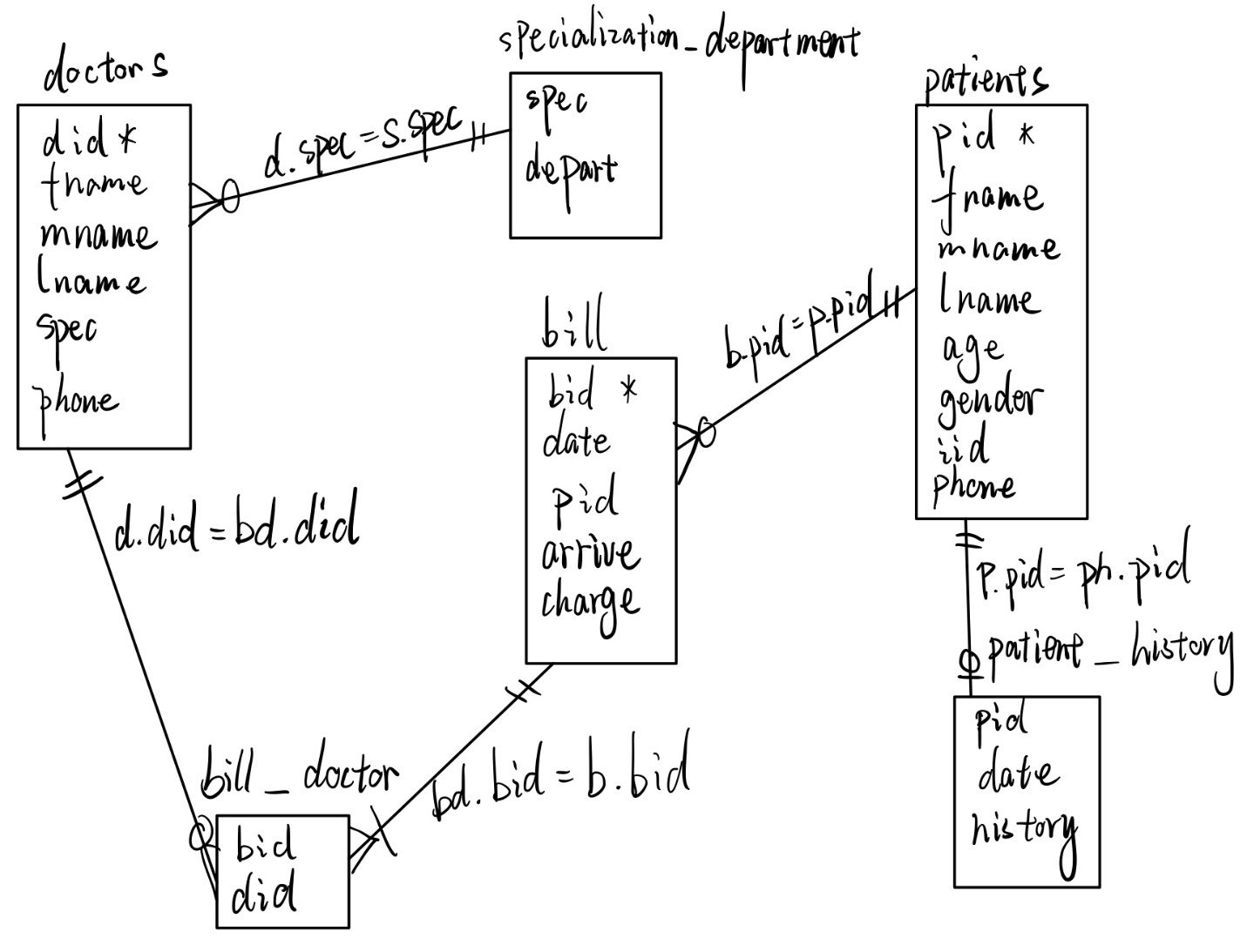


Input bill id then click cancel button. It will change the arrive value of the bill to N then change charge amount to 5.

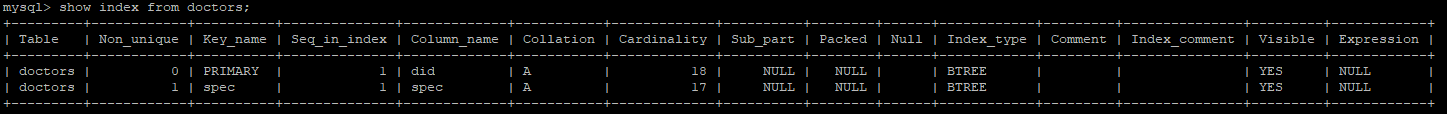
(Time limit and code ability limited do not complete PHP code part)

**Database Design**

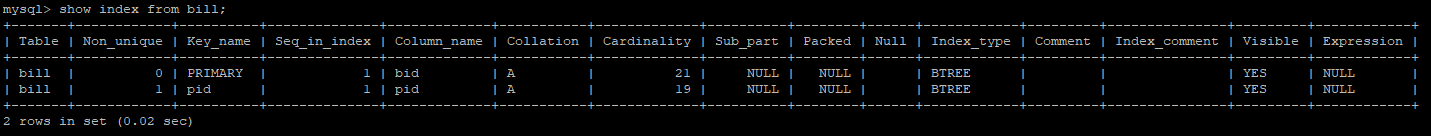
**Crow's foot diagram**



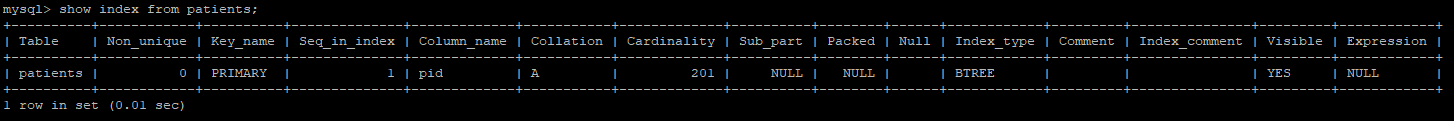
Data declaration: I build this data by my own, except specialization\_department. The department and specialization refer to Wiki. Other data including name, phone, charge, date, age, gender, insurance id is made by random function of excel table.

**Indexes and keys**

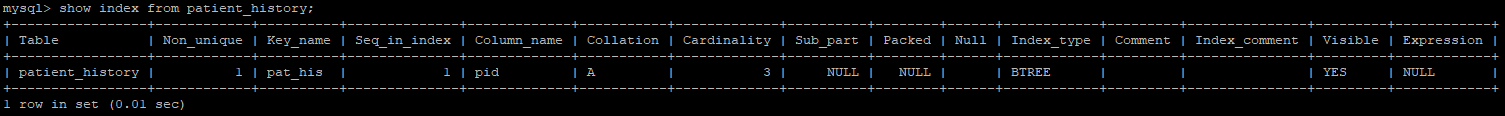
In table **doctors**, there is a primary key of doctor id and one B tree index of specialization. Because in this table, doctor id needs to be unique. And in most of query, all information in the table searched by doctor id through equality. And spec is also searched in the application of generate department monthly income so it needs an index.



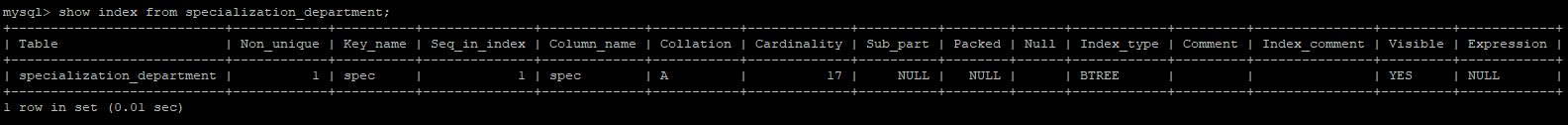
In table **bill**, there is a primary key of bill id and one B tree index of patient id. Because in this table, bill id needs to be unique and most of selection access information by use equality search bill id. Patient id also needs an index because application of generate bill for patient searches patient id in bill table.



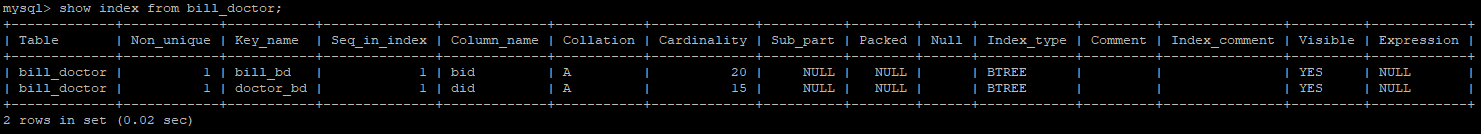
In table **patients**, there is a primary key of patient id. Because patient id needs to be unique and applications of generating bill and recording medical history use equality search with patient id.



In table **patient\_history**, there is one B tree index of patient id. There is no primary key because nothing must to be unique. And application of recording medical history needs to search by patient id.



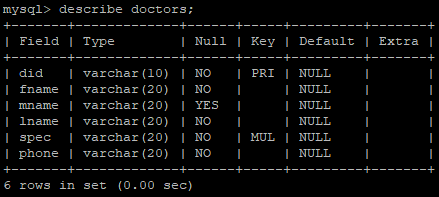
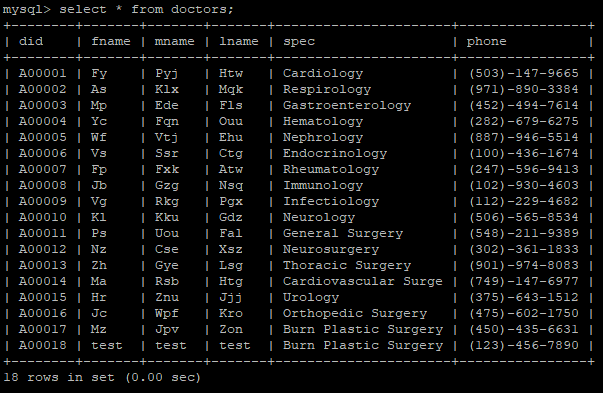
In table **specialization\_department**, there is one B tree index of spec. This table is small relatively because number of specialization and department is limited. In application of generating department monthly bill, we use equality search in specialization.



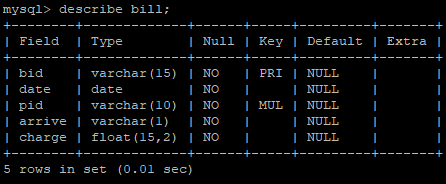
In table **bill\_doctor**, there are two B tree indexes for bid and did. Because in this table no field needs to be unique. Both bid and did are searched by equality in the applications.

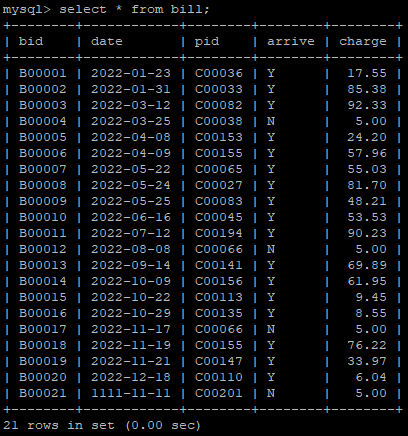
**Database description and show**

doctors:

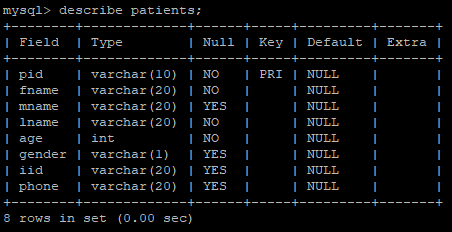
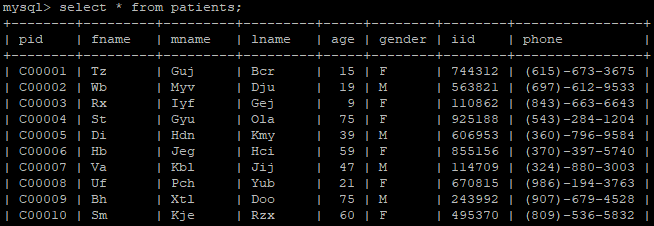
 

bill:

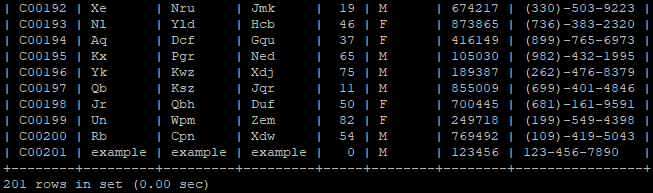




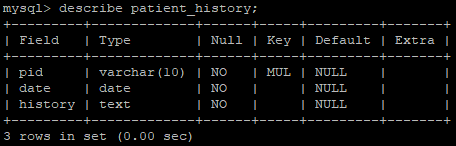
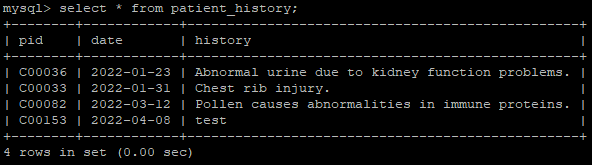
patients:

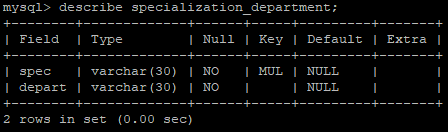
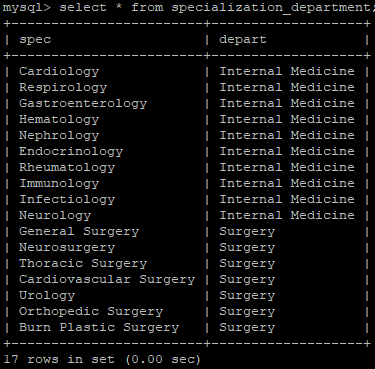
…



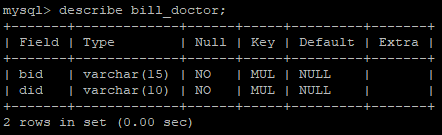
patient\_history:

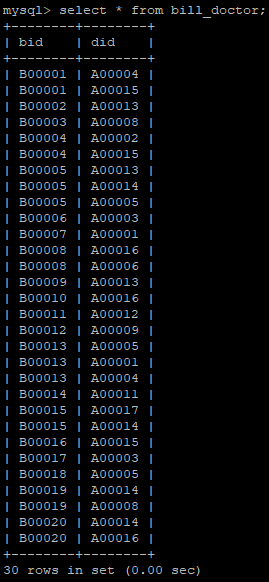
 

specialization\_department:



bill\_doctor:





**Transactions**

Start:

Start page includes a list of doctors. There includes delete button on each row. There are 6 input boxes in the bottom. They are doctor id, first name, middle name, last name, specialization and phone number. And a button of insert/update.

After insert/update, there is a bill printer part. Here is an input box of bill id and a button of generate bill.

Then is an application for doctor to record patients’ medical history. There are two input boxes, including patient id and history description. And a record button.

Then is an income report application. It needs three input boxes. Year, month and department. And a generate button also.

Then is an Annual Report generator. It has an input box of year, and a generate button.

Last one is an appointment cancel application. It needs a bill id input box and a cancel button.

Insert/update doctor:

If doctor id exists in the database, it updates. Otherwise, it inserts information. For input/update, middle name can be empty. It uses the input from the boxes to create an INSERT statement or UPDATE statement in the table doctors.

Bill printer:

It uses input of patient id to create a SELECT query, search the basic information of a standard bill. Including bill id, date, insurance id, doctor’s first name, last name, charge amount and patient’s first name, last name.

Medical history record:

It uses input of patient id, the input date from system and history description to create an INSERT statement in the patient\_history table.

Department Monthly income:

It uses input of month, year with number format and a department to generate a SELECT query, get the sum number of charge of that month by searching the table bill.

Annual Bill report:

With the input of the year in number format, it creates a SELECT query to search the bill table and give all the fields information of that table.

Cancel appointment(roll back):

With the input of the bill id, it searches the bill table and creates UPDATE statement to change the value of arrive to N and charge amount to 5.

**Error Trapping**

Insert/update doctor:

If any necessary information missing or any input types are different than expect, an input error jumps out. Field spec have error check that if input not in the specializaiton\_department table, an input error jumps out. If the same data input twice, the error that data exists appears.

Bill printer:

If the input bill is not in database, it shows an error that bill not found.

Medical history record:

If the input patent id is not in data base, it shows and error that patient not found.

Department Monthly income:

If input type is different than expect, an input error jumps out. If input department is not in the specialization\_department table, it shows an error that department not found.

Annual Bill report:

If input type is different than expect, an input error jumps out.

Cancel appointment:

If the input bill is not in database, it shows an error that bill not found. If the input bill is already cancelled, it shows an error that bill already cancelled.

**Summary**

**Problem solving**

Each application solves the problem correspondingly. In PHP interface, input property data and click insert/update button to add/update a doctor. Click delete button to remove it out of the database. Input a proper patient id and click the generate button to get the bill. Input patient id and medical history, then click the record button to insert medical history data into database. Input year and month into monthly income area then click the generate button, then a small report of monthly income of hospital appears. Input property year into annual bill report area, then click the generate button. A list of all bill information in that year appears. Input property bill id and click cancel, then that appointment will be cancelled.

**Technical Difficulty and Solving**

The mainly technical difficulty in this project appears in tree places.

First one is data generation. I use excel random function to finish this and learn to create random result in number and letter. Example: =CHAR(RAND()\*26+65)&CHAR(RAND()\*26+97)

This generates random one uppercase letter and two lowercase letters.

The second one is creating indexes and keys. I solve it by thinking according to the 6 applications.

The third part is about PHP realization and error trapping in PHP. We need to use SELECT query in the application and complete it by error trapping. Some PHP application and error trapping is not done because my PHP coding skill limitation.