**Assignment 4**

**Travel Agency**

* **Requirement Gathering :**

My friend Shradaya Ramtel provided me with the requirements as follow:

**Request for proposal**

|  |  |  |
| --- | --- | --- |
| RFP: **Travel Agency** | Proposal Due By: **09/20/2021** | **Manakamana Travel Agency** |
| **Project Overview:**  Our company Manakamana Travel Agency is willing to build a system in order to make it easy for us to communicate well with our customers. We have a number of travel plans and the customers can select a travel plan and plan their trip ahead. | | |
| **Project Goals:**   * Make Bookings * Make Payments * Add new travel packages * Add new travel locations * Add new means of transport | | |
| **Scope of Work:**  We would like the system that allows the general users to view the different available packages, make bookings for the packages, and make payments for their travel; while the employees have the ability to make changes in packages, locations and means of transport. The information regarding the past bookings, payments and expired packages should also be stored. | | |
| **Current Roadblocks and Barriers to Success**   * No proper data management * No area for staffs only admin area * No proper way to get full information about packages * Poor payment management | | |
| **Submission Requirements**   * Need a fully functional system * Must be able to handle hundreds of users at once * Abundant storage | | |
| Project Due By: **November 2021** | | Budget: **$20000** |
| Contact: **Shradaya Ramtel** | Email: **Shradays@gmail.com** | Phone #: **9865737373** |

**Analyzing and drafting requirements**

* Packages are listed with location of visit,hotel, transport and cost
* Packages history be maintained regarding expiry or ongoing
* Customer can book the packages
* Payment can be done through application
* Customer History be maintained
* Employee can modify packages, locations, hotel, transport info
* Admin has overall access
* **Conceptual Modeling**

1. Identifying Entities:

package - package\_id, name, start\_date, end\_date, description, cost,status

destination - destination\_id, location\_name,package\_id

hotel - hotel\_id, name, location, cost,package\_id

transport - transport\_id, type, cost,package \_id

customer - customer\_id, name, age, phone, email

booking - booking\_id, date , customer\_id

payment - payment\_id, type, amount

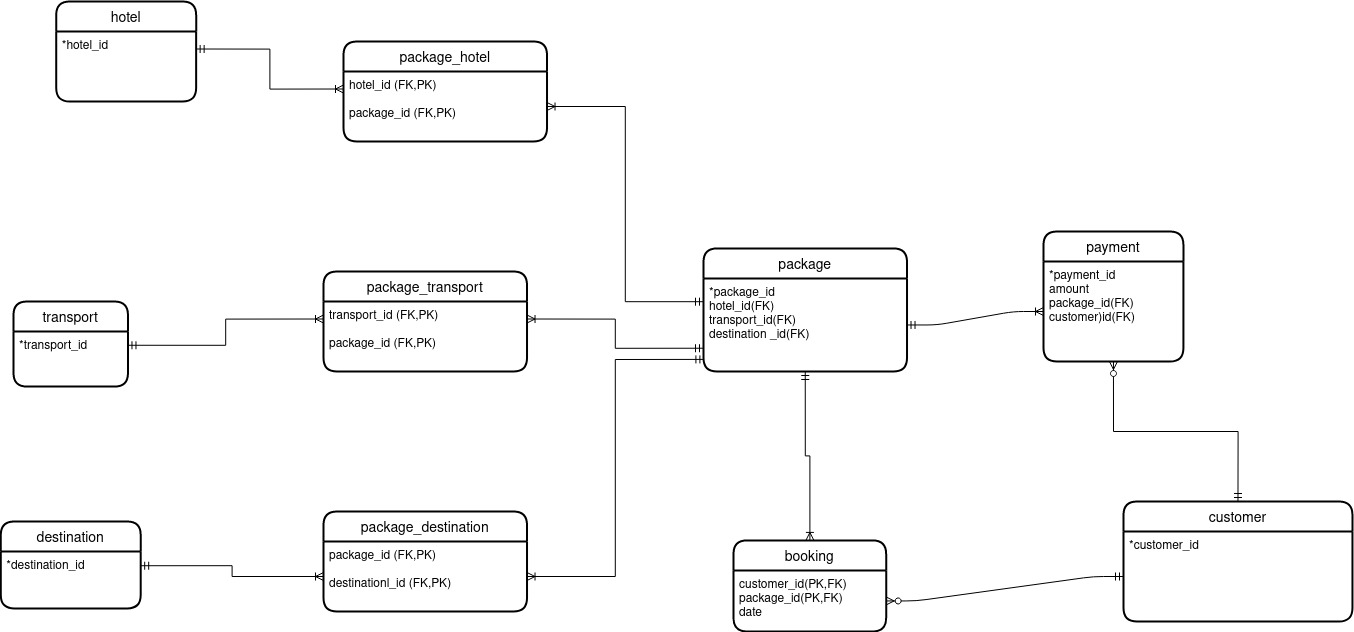
customer\_history - ch\_id, customer\_id, date, package\_id,payment\_id

employee - employee\_id, name,age, phone, email

2. Defining Business Rules:

* Contains three groups : Customer, Employee, Admin
* Packages are listed with location of visit,hotel, transport and cost
* Packages history be maintained regarding expiry or ongoing
* Customer can book the packages
* Payment can be done through application
* Customer History be maintained
* Employee can modify packages, locations, hotel, transport info
* Admin has overall access

3. Conceptual Model



**Normalization:**

e.g. Package table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| package\_id | name | destination\_id | destination | hotel\_id | hotel |  | transport\_id | transport |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| package\_id | name | destination\_id | hotel\_id | transport\_id |

|  |  |
| --- | --- |
| hotel\_id | hotel |

|  |  |
| --- | --- |
| destination\_id | destination |

**Functional Dependencies :**

package\_id -> name, destination, transport , hotel

package\_id, hotel\_id -> hotel

hotel\_id -> hotel

package\_id -> hotel\_id -> hotel

* **Logical Modeling**

1. Identifying attributes and domains for entities and relationships:

|  |  |  |
| --- | --- | --- |
| **Entity** | **Description** | **Domain** |
| **package** | packages containing information about travel |  |
| Attributes:  package\_id  name  start\_date  end\_date  description  cost  status  hotel  destination  transport | Identifier for entity, SK, PK  Name of package  Date of start  Date of end  Info About package  Price for the package plan  active/inactive status  Hotel Code(FK)  Destination Code(FK)  Transport Code(FK) | Auto Generated  Text  Date  Date  Text  Number  Boolean Value  Valid id from hotel table  Valid id from destination table  Valid id from transport table |
| **hotel** | Hotel information |  |
| Attributes:  hotel\_id  name  description  location  cost | Identifier for entity, SK, PK  Name of Hotel  About Hotel  Hotel Location  Cost for hotel | Auto Generated  Text  Text  Valid id from city table  Number |
| **package\_hotel** | Relationship between packages and hotels |  |
| Attributes:  package\_id  hotel\_id | Package Code(PK,FK)  Hotel Code (PK,FK) | Valid id from package table  Valid id from hotel table |
| **destination** | Location information |  |
| Attributes:  destination\_id  location\_name  description | Identifier for entity, SK, PK  Location Name  About Location | Auto Generated  Text  Text |
| **package\_destination** | Relation Between packages and destinations |  |
| Attributes:  package\_id  destination\_id | Package Code(PK,FK)  Destination Code (PK,FK) | Valid id from package table  Valid id from destination table |
| **transport** | Transport information |  |
| Attributes:  transport\_id  type  cost | Identifier for entity, SK, PK  Transport ways(from air, from road)  Cost for transport | Auto Generated  Text  Number |
| **package\_transport** | Relation between packages and transports |  |
| Attributes:  package\_id  transport \_id | Package Code(PK,FK)  Transport Code (PK,FK) | Valid id from package table  Valid id from transport table |
| **customer** | Users of application |  |
| Attributes:  customer\_id  name  dob  phone  email  location | Identifier for entity, SK, PK  First name, Last name  Date of Birth  Can have one phone no.  Can have one email  Address | Auto Generated  Text  Date  Unique  Unique  Valid id from city table |
| **booking** | Booking info about customer and packages they book |  |
| Attributes:  customer\_id  package\_id  booking\_date  package\_status | Customer Code (PK,FK)  Package Code(PK,FK)  Date of booking  Can have active and expired package for history | Valid id from customer table  Valid id from package table  Valid Date  Boolean Value |
| **payment** | Payment details |  |
| Attributes:  payment\_id  customer\_id  package\_id  type  date  amount | Identifier for entity, SK, PK  Customer Code(FK)  Package Code (FK)  Different Payment options  Date of Payment  Amount of pay | Auto Generated  Valid id from customer table  Valid id from package table  Text  Date  Valid Number |

2. Logical model

<https://drive.google.com/drive/folders/1BQbZpGRJ17VY9wNK3QT1x-xRBJT3DLnp>

* **Physical Model**

CREATE DATABASE travelagency;

-- Creating necessary tables from logical model

--hotel

CREATE TABLE hotel(

hotel\_id serial PRIMARY KEY,

name VARCHAR(50),

location INT REFERENCES city(city\_code),

description VARCHAR(150),

cost INT

);

--transport

CREATE TABLE transport(

transport\_id serial PRIMARY KEY,

type VARCHAR(20),

cost INT

);

--destination

CREATE TABLE destination(

destination\_id serial PRIMARY KEY,

location\_name VARCHAR(50),

description VARCHAR(150)

);

-- package

CREATE TABLE package(

package\_id serial PRIMARY KEY,

name VARCHAR(50),

start\_date DATE,

end\_date DATE,

description VARCHAR(150),

cost INT ,

active\_status BOOLEAN,

hotel\_id INT REFERENCES hotel(hotel\_id),

transport\_id INT REFERENCES transport(transport\_id),

destination\_id INT REFERENCES destination(destination\_id)

);

--package\_hotel

CREATE TABLE package\_hotel(

hotel\_id INT REFERENCES hotel(hotel\_id),

package\_id INT REFERENCES package(package\_id),

unique(hotel\_id, package\_id)

);

--package\_transport

CREATE TABLE package\_transport(

transport\_id INT REFERENCES transport(transport\_id),

package\_id INT REFERENCES package(package\_id),

unique(transport\_id,package\_id)

);

--package\_destination

CREATE TABLE package\_destination(

destination\_id INT REFERENCES destination(destination\_id),

package\_id INT REFERENCES package(package\_id),

unique(destination\_id,package\_id)

);

--city

CREATE TABLE city(

city\_code INT PRIMARY KEY,

city\_name VARCHAR(50)

);

--customer

CREATE TABLE customer(

customer\_id serial PRIMARY KEY,

name VARCHAR(50),

dob DATE ,

email VARCHAR(30) unique,

phone INT not null,

location INT REFERENCES city(city\_code)

);

--booking

CREATE TABLE booking(

customer\_id INT REFERENCES customer(customer\_id),

package\_id INT REFERENCES package(package\_id),

booking\_date DATE ,

package\_status BOOLEAN

)

--payment

CREATE TABLE payment(

payment\_id serial PRIMARY KEY,

customer\_id INT REFERENCES customer(customer\_id),

package\_id INT REFERENCES package(package\_id),

type VARCHAR(30),

payment\_date DATE,

amount INT

);

* Other forms of normalization were done while physical modeling:

Like: Hotel Location, Customer location attributes were split into city table.

**Health Care**

* **Requirement Gathering :**

**Request for Proposal (RPF):**

An Online Health Care system should be made. It should consists the details of registered Clinics, Hospitals and Pharmacies with their provided services. Users can register as a Doctor, Patient , Clinic, Hospital or Pharmacy. The registered Doctors should be listed in the site along with registered clinics,hospitals and pharmacies. User can have appointment to clinic,hospital or doctor. There can also be consultation with doctor directly. The billing can be done through payment like esewa to registered clinic/hospital/doctor/pharmacies. There can also be Discussion section to write about the problems related to users.

**Collecting Information:**

When collecting information from key stakeholders, it is essential to focus on two main questions that should serve as guidelines for the overall requirements gathering process.

1. How can an this system contribute to your practice’s strategic goals?
2. What features are the best for achieving your practice’s strategic goals?

So, this may include meeting sample like:

Meeting Notes

Date: Sept 7, 2021

Agenda:

* Discussion about the proposed site.
* Cost finalization.
* Gathering info for design.

Meeting Notes:

1. User friendly site to be made.

Decision Made:

1. Project Accepted.
2. Design for site finalized.

Action Items:

Discovery Questions

|  |  |
| --- | --- |
| **Questions/ Thoughts** | **Answers** |
| Who will be using the new system? | It should be focused to give public a online health care options. So, the system is frequently used by patients , doctors, pharmacies and admin to update regarding new registered details or for removing existing details. |
| So the users can register using any of the options. But how will be the authenticity be done if the doctor/pharmacy/clinic registered directly ? So I recommend admin verify before these registration.. do you consider this ? | Yeah definitely, it would be better if the users are real and valid. So, that should be done. |
| What patient population are we documenting? (i.e., pediatrics, geriatrics, patients, etc.) | There can be different categories of patients obviously i.e pediatrics, geriatrics but all can be kept under patients. |
| Should the payment system be other than esewa? | Since esewa is popular, it should be used as a payment method. Other payments would be plus if they could be add`ed like khalti. |
| What database to use? | MySQL or Posgres, any! |
|  |  |

**Analyzing and drafting requirements**

After Analyzing , the entities may include:

* Hospitals , Clinics, Pharmacies,Doctor, Patients can be registered
* Services are provided by Hospitals, Clinics, Doctors
* Patient can appoint to Hospitals, Clinics, Doctors
* Patient can consult to specific Doctor
* Patient can give feedback to doctor
* Patient can open conversation in Discussion section
* Esewa or other payments be done by patients to hospital, pharmacies, doctors
* History is maintained for users
* Admin verifies Hospital, Clinic, Doctor and Pharmacy registration
* Admin can modify registered user details, payments and other features
* **Conceptual Modeling**

1. Identifying Entities:

clinic - clinic\_id ,name,location

hospital - hospital\_id, name, location

doctor - doctor\_id, doctor\_name, phone\_no,qualification, specialization(service)

services - service\_id, name ,cost

pharmacy - pharmacy\_id,name, location

medicine - med\_id, name, brand,cost

patient - patient\_id , name, age, sex, address ,phone, blood\_group

patient\_history - disease\_history , treatment

appointment - appointment\_id , patient\_id , date

consultation - c\_id, doctor\_id, date,patient\_id

feedback - feedback\_id, description,doctor\_id, patient\_id

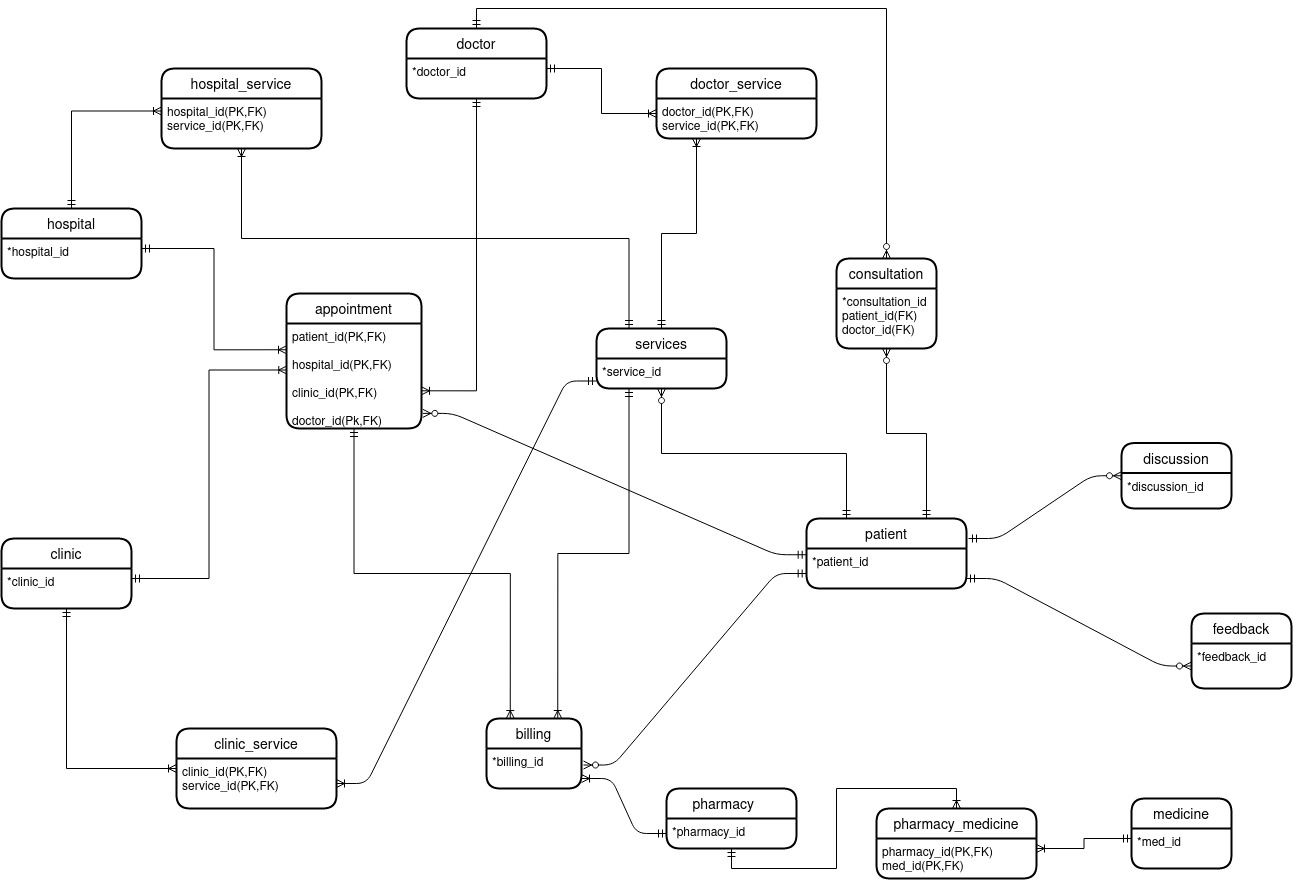
discussion - p\_id, description

billing - billing\_id, amount , type

2. Defining business rules

* Hospitals , Clinics, Pharmacies,Doctor, Patients can be registered
* Services are provided by Hospitals, Clinics, Doctors
* Patient can appoint to Hospitals, Clinics, Doctors
* Patient can consult to specific Doctor
* Patient can give feedback to doctor
* Patient can open conversation in Discussion section
* Esewa or other payments be done by patients to hospital, pharmacies, doctors
* History is maintained for users
* Admin verifies Hospital, Clinic, Doctor and Pharmacy registration
* Admin can modify registered user details, payments and other features

3. Conceptual Model :



* **Logical Modeling**

1. Identifying attributes and domains for entities and relationships:

|  |  |  |
| --- | --- | --- |
| **Entity** | **Description** | **Domain** |
| **hospital** | Hospital information |  |
| Attributes:  hospital\_id  name  location | Identifier for entity, SK, PK  Name of hospital  Location of hospital | Auto Generated  Text  Text |
| **clinic** | Clinic information |  |
| Attributes:  clinic\_id  name  location | Identifier for entity, SK, PK  Name of clinic  Location of clinic | Auto Generated  Text  Text |
| **pharmacy** | Pharmacy information |  |
| Attributes:  pharmacy\_id  name  location | Identifier for entity, SK, PK  Name of pharmacy  Location of pharmacy | Auto Generated  Text  Text |
| **doctor** | Doctor information |  |
| Attributes:  doctor\_id  name  phone\_no  qualification  specialization  service\_id | Identifier for entity, SK, PK  Name of doctor  Valid phone no. of doctor  Qualification of doctor  Specialization info  Service Code (FK) | Auto Generated  Text  Number  Text  Text  Valid id from service table |
| **services** | Services provided for health care |  |
| Attributes:  service\_id  name  cost  patient\_id | Identifier for entity, SK, PK  Name of service  Price of service  Patient Code | Auto Generated  Text  Valid Number  Valid Code from patient table |
| **appointment** | Relation for patient appointments |  |
| Attributes:  patient\_id  hospital\_id  clinic\_id  doctor\_id  date | Patient Code (PK, FK)  Hospital Code (PK, FK)  Clinic Code(PK,FK)  Doctor Code(PK,FK)  Date of appointment | Valid patient id from patient table  Valid id from hospital table  Valid id from clinic table  Valid id from doctor table  Date |
| **patient** | Patient Details |  |
| Attributes:  patient\_id  name  dob  phone\_no  location  blood\_group | Identifier for entity, SK, PK  Name of patient  Date of Birth  Valid Phone number  Address  Blood Group | Auto Generated  Text  Date  Number  Text  Text |
| **consultation** | Relation for consulting doctor |  |
| Attributes:  patient\_id  doctor\_id | Patient Code (PK,FK)  Doctor Code (PK,FK) | Valid id from patient table  Valid id from doctor table |
| feedback | Feedback give by patient to doctor |  |
| Attributes:  patient\_id  doctor \_id  description | Patient Code(PK,FK)  Doctor Code (PK,FK)  Description given by patient | Valid id from patient table  Valid id from doctor table  Text |
| **discussion** | Discussion about patient problems |  |
| Attributes:  patient\_id  description | Patient Code(PK,FK)  Discussion opened by patient | Valid id from patient table  Text |
| **billing** | Payment Details |  |
| Attributes:  billing\_id  type  amount | Identifier for entity, SK, PK  Type of pay  Valid amount | Auto Generated  Text  Number |
| **medicine** | Medicine Details |  |
| Attributes:  medicine\_id  name  brand  cost | Identifier for entity, SK, PK  Name of medicine  Brand of medicine  Price of medicine |  |

2. Logical Mode:

<https://drive.google.com/drive/folders/1BQbZpGRJ17VY9wNK3QT1x-xRBJT3DLnp?usp=sharing>

* **Physical Model**

CREATE DATABASE healthcare;

-- Creating necessary TABLE from logical model

--city

CREATE TABLE city(

city\_code INT PRIMARY KEY,

city\_name VARCHAR(50)

);

--hospital

CREATE TABLE hospital(

hospital\_id serial PRIMARY KEY,

name VARCHAR(50),

location INT REFERENCES city(city\_code)

);

--clinic

CREATE TABLE clinic(

clinic\_id serial PRIMARY KEY,

name VARCHAR(50),

location INT REFERENCES city(city\_code)

);

--pharmacy

CREATE TABLE pharmacy(

pharmacy\_id serial PRIMARY KEY,

pharmacy\_name VARCHAR(50),

location INT REFERENCES city(city\_code)

);

--medicine

CREATE TABLE medicine(

med\_id serial PRIMARY KEY,

name VARCHAR(50),

brand VARCHAR(50),

cost INT

);

--services

CREATE TABLE services(

service\_id serial PRIMARY KEY,

service\_name VARCHAR(50),

cost INT

);

--specialization

CREATE TABLE specialization(

specialization\_id serial PRIMARY KEY,

specialization\_name VARCHAR(50)

);

--doctor

CREATE TABLE doctor(

doctor\_id serial PRIMARY KEY,

name VARCHAR(50),

phone\_no INT unique,

qualification VARCHAR(50),

specialization INT REFERENCES specialization(specialization\_id),

service\_id INT REFERENCES services(service\_id)

);

--patient

CREATE TABLE patient(

patient\_id serial PRIMARY KEY,

name VARCHAR(50),

dob DATE,

phone INT unique,

location INT REFERENCES city(city\_code),

blood\_group VARCHAR(10)

);

--hospital\_service

CREATE TABLE hospital\_service(

hospital\_id INT REFERENCES hospital(hospital\_id),

service\_id INT REFERENCES services(service\_id),

unique(hospital\_id, service\_id)

);

--clinic\_service

CREATE TABLE clinic\_service(

clinic\_id INT REFERENCES clinic(clinic\_id),

service\_id INT REFERENCES services(service\_id),

unique(clinic\_id, service\_id)

);

--doctor\_service

CREATE TABLE doctor\_service(

doctor\_id INT REFERENCES doctor(doctor\_id),

service\_id INT REFERENCES services(service\_id),

unique(doctor\_id, service\_id)

);

--consultaion

CREATE TABLE consultation(

consultation\_id serial PRIMARY KEY,

patient\_id INT REFERENCES patient(patient\_id),

doctor\_id INT REFERENCES doctor(doctor\_id),

consultation\_date DATE

);

--patient\_services

CREATE TABLE patient\_services(

id serial PRIMARY KEY,

patient\_id INT REFERENCES patient(patient\_id),

service\_id INT REFERENCES services(service\_id)

);

-- discussion

CREATE TABLE discussion(

discussion\_id serial PRIMARY KEY,

patient\_id INT REFERENCES patient(patient\_id),

question VARCHAR(200),

solution VARCHAR(200)

);

--feedback

CREATE TABLE feedback(

feedback\_id serial PRIMARY KEY,

patinent\_id INT REFERENCES patient(patient\_id),

doctor\_id INT REFERENCES doctor(doctor\_id),

description VARCHAR(300)

);

-- appointment

CREATE TABLE appointment(

appointment\_id serial PRIMARY KEY,

patient\_id INT REFERENCES patient(patient\_id),

hospital\_id INT REFERENCES hospital(hospital\_id),

clinic\_id INT REFERENCES clinic(clinic\_id),

doctor\_id INT REFERENCES doctor(doctor\_id),

appointment\_date DATE ,

cost INT

);

--pharmacy\_medicine

CREATE TABLE pharmacy\_medicine(

id serial PRIMARY KEY,

pharmacy\_id INT REFERENCES pharmacy(pharmacy\_id),

med\_id INT REFERENCES medicine(med\_id)

);

--billing

CREATE TABLE billing(

billing\_id serial PRIMARY KEY,

patient\_id INT REFERENCES patient(patient\_id),

appointment\_id INT REFERENCES appointment(appointment\_id),

service\_id INT REFERENCES services(service\_id),

pharmacy\_id INT REFERENCES pharmacy(pharmacy\_id),

type VARCHAR(30),

amount INT

);