**Weekend Assignment**

**Design a database for the following scenario. (Conceptual + Logical + Physical Phase)**

The client manages a couple of dental offices. One is called the Jawlakhel Office, the other the Putalisadak Office. The client needs the system to manage its patients and appointments, alerting the patients when and where their appointments occur, either by email or by phone, and then assisting in the selection of new appointments. The client wants to be able to keep up with the records of all the patients’ appointments without having to maintain lots of files. The dentists might spend time at each of the offices throughout the week. For each appointment, the client needs to have everything documented that went on and then invoice the patient.

Currently, the client uses a patient number in its computer system that corresponds to a particular folder that has the patient’s records. The system needs to track and manage several dentists and quite a few dental hygienists who the client needs to allocate to each appointment as well.

* **Conceptual Modeling :**

Here, we list out possible entities and their relationships.

1. Identifying Entities:

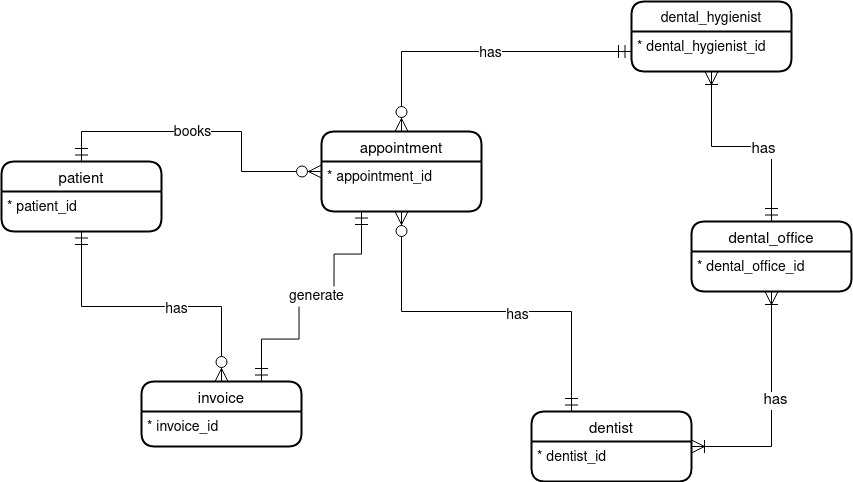
* dental\_office
* dentist
* dental\_hygienist
* patient
* appointment
* invoice

2. Defining Business Rules from requirements:

* two dental offices ( Jawlakhel and Putalisadak)
* patient gets appointed via phone or email
* dentist might spend a week at each offices
* dental hygienists are assigned to appointments
* invoice is generated per appointments
* patients pays the amount in invoice

3. Conceptual Model:

* dental\_office has many dentists
* patient can have appointments
* dentist is assigned to many appointments
* dental hygienist is assigned to many appointments
* an appointment has fixed dentist and dental hygienist
* invoice is generated per appointments
* invoice is paid by respective patient

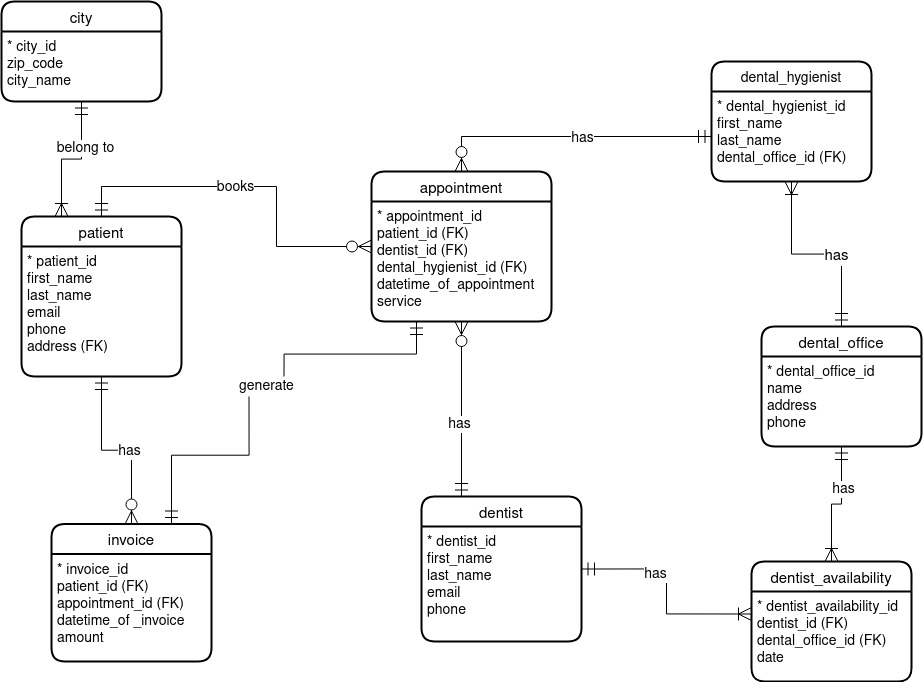


* **Logical Modeling :**

1. Identifying attributes and domains for entities and relationships:

|  |  |  |
| --- | --- | --- |
| **Entity** | **Description** | **Domain** |
| **dental\_office** | The office which provides dental service. |  |
| Attributes:  dental\_office \_id  name  address  phone | Identifier for user entity, SK, PK  Name of office  Address of office  Valid phone no. of office | Auto Generated  Text  Text  Number |
| **dentist** | Dentist information |  |
| Attributes:  dentist \_id  name  email  phone | Identifier for user entity, SK, PK  First Name, Last Name of Dentist  Valid email  Valid Phone no. | Auto Generated  Text  Text  Number |
| **dentist\_availability** | Dentist availability at particular office at particular date |  |
| Attributes:  availability \_id  dentist\_id  dental\_office\_id  date | Identifier for user entity, SK, PK  Dentist Code (FK)  Dental Office Code (FK)  Valid date of availability of dentist | Auto Generated  Valid id from dentist table  Valid id from dental\_office table  Date |
| **dental\_hygienist** | Information about dental hygienist |  |
| Attributes:  dental\_hygienist \_id  name  phone  dental\_office\_id | Identifier for user entity, SK, PK  First Name, Last Name of Dental Hygienist  Valid phone no.  Dental Office code (FK) | Auto Generated  Text  Number  Valid id from dental\_office table |
| **patient** | Patient information |  |
| Attributes:  patient \_id  patient \_name  email  phone  address | Identifier for user entity, SK, PK  First Name , Last Name  Valid email  Valid Phone no.  Address code (FK) | Auto Generated  Text  Text  Number  Valid id from city table |
| **city** | Information about cities |  |
| Attributes:  city \_id  zip\_code  city\_name | Identifier for user entity, SK, PK  Zip code of city  Name of city | Auto Generated  Number  Text |
| **appointment** | Appointment information about patient |  |
| Attributes:  appointment \_id  patient\_id  dentist \_id  dental\_hygienist\_id  datetime\_of\_appointment  service | Identifier for user entity, SK, PK  Patient Code (FK)  Dentist Code(FK)  Dental Hygienist Code(FK)  Valid date and time for appointment  Service type | Auto Generated  Valid id from patient table  Valid id from dentist table  Valid id from dental\_hygienist table  Timestamp  Text |
| **invoice** | Invoice information for appointments |  |
| Attributes:  invoice \_id  patient \_id  appointment\_id  datetime\_of\_invoice  amount | Identifier for user entity, SK, PK  Patient Code (FK)  Appointment Code (FK)  Valid date and time of invoice.  Cost for appointment | Auto Generated  Valid id from patient table  Valid id from appointment table  Timestamp  Number |

2. Logical model :



* **Physical Modeling :**

Here, I implemented required entities and relations into Postgresql relational database.

DROP DATABASE if EXISTS dentist\_appointment;

CREATE DATABASE dentist\_appointment;

**-- dental office**

CREATE TABLE dental\_office(

dental\_office\_id serial PRIMARY KEY,

name VARCHAR(150) NOT NULL,

address VARCHAR (150) NOT NULL,

phone int NOT NULL

);

SELECT \* FROM dental\_office ;

INSERT into dental\_office VALUES (1,'Jawlakhel Office','Jawlakhel,Kathmandu',5090000);

INSERT into dental\_office VALUES (2,'Putalisadak Office','Putalisadak,Kathmandu',5091111);

**-- dentist**

CREATE TABLE dentist(

dentist\_id serial PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(150) NOT NULL,

phone int NOT NULL

);

SELECT \* FROM dentist;

INSERT into dentist VALUES

(1,'Rajan','Giri','rajan@gmail.com',984100000),

(2,'Pranav','Pudasaini','prav12@gmail.com',984300000),

(3,'Mohit','Bhatta','mohit@gmail.com',984311111),

(4,'Saphal','Shakha','safal@gmail.com',984312346),

(5,'Samyog','Shah','samyog@gmail.com',984343434);

**-- dentist\_availability**

CREATE TABLE dentist\_availability(

dentist\_availability\_id serial PRIMARY KEY,

dentist\_id int NOT NULL,

dental\_office\_id int NOT NULL,

available\_date date,

CONSTRAINT fk\_availability\_dentist FOREIGN KEY(dentist\_id)

REFERENCES dentist(dentist\_id),

CONSTRAINT fk\_availability\_dentalOffice FOREIGN KEY(dental\_office\_id)

REFERENCES dental\_office(dental\_office\_id)

);

SELECT \* FROM dentist\_availability ;

INSERT into dentist\_availability VALUES

(1,1,1,'2021-09-11'),

(2,1,2,'2021-09-18'),

(3,2,1,'2021-09-12'),

(4,2,2,'2021-09-16'),

(5,3,1,'2021-09-11'),

(6,3,2,'2021-09-12'),

(7,4,1,'2021-09-18'),

(8,4,2,'2021-09-11'),

(9,5,1,'2021-09-13'),

(10,5,2,'2021-09-19');

**-- dental\_hygienist**

CREATE TABLE dental\_hygienist(

dental\_hygienist\_id serial PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

second\_name VARCHAR(50) NOT NULL,

phone int NOT NULL,

dental\_office\_id int NOT NULL,

CONSTRAINT fk\_hygienist\_office FOREIGN KEY(dental\_office\_id)

REFERENCES dental\_office(dental\_office\_id)

);

SELECT \* FROM dental\_hygienist dh ;

INSERT into dental\_hygienist VALUES

(1,'Ram','Giri',985100000,1),

(2,'Prem','Prajapati',984300000,1),

(3,'Mohan','Lal',984333333,2),

(4,'Sita','Siwakoti',984156656,2),

(5,'Sa','Shah',984332404,2);

**-- city**

CREATE TABLE city (

city\_id serial PRIMARY KEY,

zip\_code int NOT NULL,

city\_name VARCHAR(50) NOT NULL,

unique(zip\_code),

unique(city\_name)

);

SELECT \* FROM city;

INSERT into city VALUES

(1,44600,'Kathmandu'),

(2,44800,'Bhaktapur'),

(3,44700,'Lalitpur');

**-- patient**

CREATE TABLE patient(

patient\_id serial PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(150) unique NOT NULL,

phone int unique NOT NULL,

address int NOT NULL,

CONSTRAINT fk\_patient\_city FOREIGN KEY(address)

REFERENCES city(city\_id)

);

SELECT \* FROM patient p ;

INSERT into patient VALUES

(1,'Ram','Shr','ram@gmail.com',5055000,1),

(2,'Sita','Ghr','sita@gmail.com',565654,1),

(3,'Hari','Sree','hari@gmail.com',985000,2),

(4,'Sam','Sri','sam@gmail.com',8055000,3),

(5,'Hom','Shpp','hom@gmail.com',855000,2),

(6,'Jam','Sho','jam@gmail.com',895000,3),

(7,'Tan','Shy','tan@gmail.com',55000,1),

(8,'Raju','Sho','raju@gmail.com',98008,2);

**-- appointment**

CREATE TABLE appointment(

appointment\_id serial PRIMARY KEY,

patient\_id int NOT NULL,

dentist\_id int NOT NULL,

dental\_hygienist\_id int NOT NULL,

datetime\_of\_appointment timestamp NOT NULL,

service VARCHAR(100),

CONSTRAINT fk\_appointment\_patient FOREIGN KEY(patient\_id)

REFERENCES patient(patient\_id),

CONSTRAINT fk\_appointment\_dentist FOREIGN KEY(dentist\_id)

REFERENCES dentist(dentist\_id),

CONSTRAINT fk\_appointment\_hygienist FOREIGN KEY(dental\_hygienist\_id)

REFERENCES dental\_hygienist(dental\_hygienist\_id)

);

SELECT \* FROM appointment ;

INSERT into appointment VALUES

(1,1,3,4,'2021-09-12 10:00:00','Removing Teeth'),

(2,3,5,1,'2021-09-13 9:15:00','Plaster Teeth'),

(3,4,2,3,'2021-09-16 1:00:00','Cleaning teeth'),

(4,8,4,1,'2021-09-18 12:00:00','Filling'),

(5,2,1,1,'2021-09-11 11:00:00','Regular checkup'),

(6,7,4,3,'2021-09-11 1:30:00','Wire fit'),

(7,1,2,1,'2021-09-12 3:00:00','Removing Teeth');

**-- invoice**

CREATE TABLE invoice(

invoice\_id serial NOT NULL,

patient\_id int NOT NULL,

appointment\_id int NOT NULL,

datetime\_of\_invoice timestamp NOT NULL,

amount int NOT NULL,

CONSTRAINT fk\_invoice\_patient FOREIGN KEY(patient\_id)

REFERENCES patient(patient\_id),

CONSTRAINT fk\_invoice\_appointment FOREIGN KEY(appointment\_id)

REFERENCES appointment(appointment\_id)

);

SELECT \* FROM invoice;

INSERT into invoice VALUES

(1,1,1,'2021-09-11 10:00:00',2000),

(2,3,2,'2021-09-12 9:15:00',5000),

(4,4,3,'2021-09-14 1:00:00',2000),

(5,8,4,'2021-09-16 12:00:00',3000),

(6,2,5,'2021-09-09 11:00:00',2000),

(3,7,6,'2021-09-10 1:30:00',10000),

(7,1,7,'2021-09-11 3:00:00',2000);

**-- DQL**

**-- all patients**

SELECT \* FROM patient p ;

**-- all dentist**

SELECT \* FROM dentist d ;

**-- all dental\_hygienist**

SELECT \* FROM dental\_hygienist dh ;

**-- all the dental\_offices**

SELECT \*

FROM dental\_office;

**-- dentist availability at offices ;**

SELECT da.dentist\_availability\_id ,

concat( d.first\_name,' ',d.last\_name) as Dentist ,

available\_date,

doff."name"

FROM dentist\_availability da

JOIN dentist d

on da.dentist\_id = d.dentist\_id

JOIN dental\_office doff

on da.dental\_office\_id = doff.dental\_office\_id ;

**-- all appointments information with dentist, dental\_office and dental\_hygienist**

SELECT \* FROM appointment a ;

SELECT appointment\_id ,

concat(p.first\_name,' ',p.last\_name) as patient ,

concat( d.first\_name,' ',d.last\_name ) as Dentist,

concat(dh.first\_name,' ',dh.second\_name) as Dental\_Hygienist,

do2."name"

datetime\_of\_appointment ,

service

FROM appointment a

JOIN patient p on

a.patient\_id = p.patient\_id

JOIN dentist d on

a.dentist\_id = d.dentist\_id

JOIN dental\_hygienist dh on

a.dental\_hygienist\_id = dh.dental\_hygienist\_id

JOIN dental\_office do2 on

dh.dental\_office\_id = do2.dental\_office\_id