### 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 Modelling**

In this section we will list out the possible entities , and shop their relationship with each other using the ER diagram.

The list of possible entities according to the requirement given are:

1. dental\_office
2. patient
3. dentist
4. dental\_hygenist
5. appointment
6. operation
7. invoice

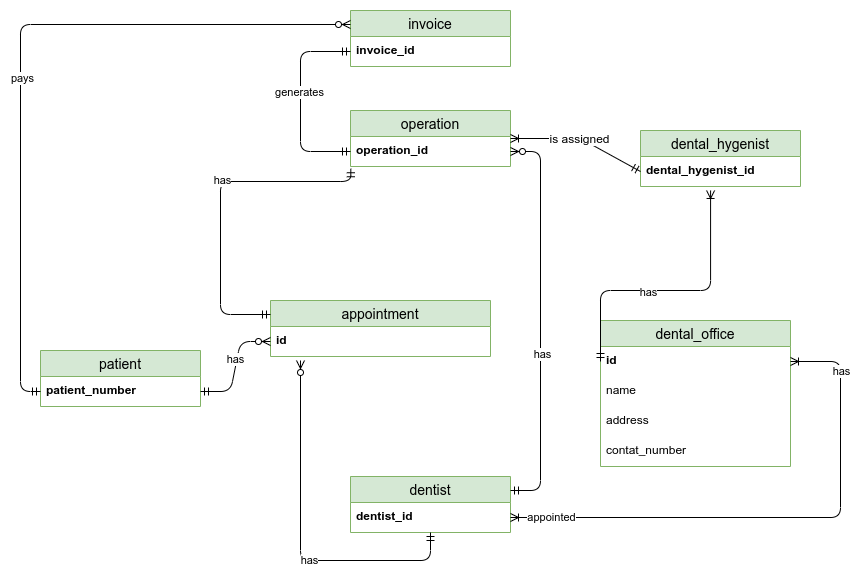
The business rules as per the requirement are.

* Their are dental offices (i.e Jawlakhel Office, the other the Putalisadak Office)
* the patient books the appointment.
* dentist is assigned to an appointment.
* dentist can be available to any of the listed dental offices throughout the week.
* After booking the appointment , the patient is assigned the operation during its appointment time.
* dental hygienist is assigned to the operation.
* Operation generates the invoice.
* Patient pays the amount generated in the invoice.

Identifying Relationship between entities.

* **patient** books **appointment**
* **dentist** is assigned to **appointment**
* **dentist** can be available at many **dental\_office**
* **dental\_office** has many **dentist**
* **dental\_office** has many **dental\_hygenist**
* **appointment** has **operation**
* **operation** has **dentist**
* **operation** has **patient**
* **dental\_hygenist** can be assigned to many **operation**
* **operation** generates **invoice**
* **patient** pays **invoice** amount.

ER DIAGRAM



*Fig: Conceptual ER diagram of dental appointment management system*

**Logical modelling**

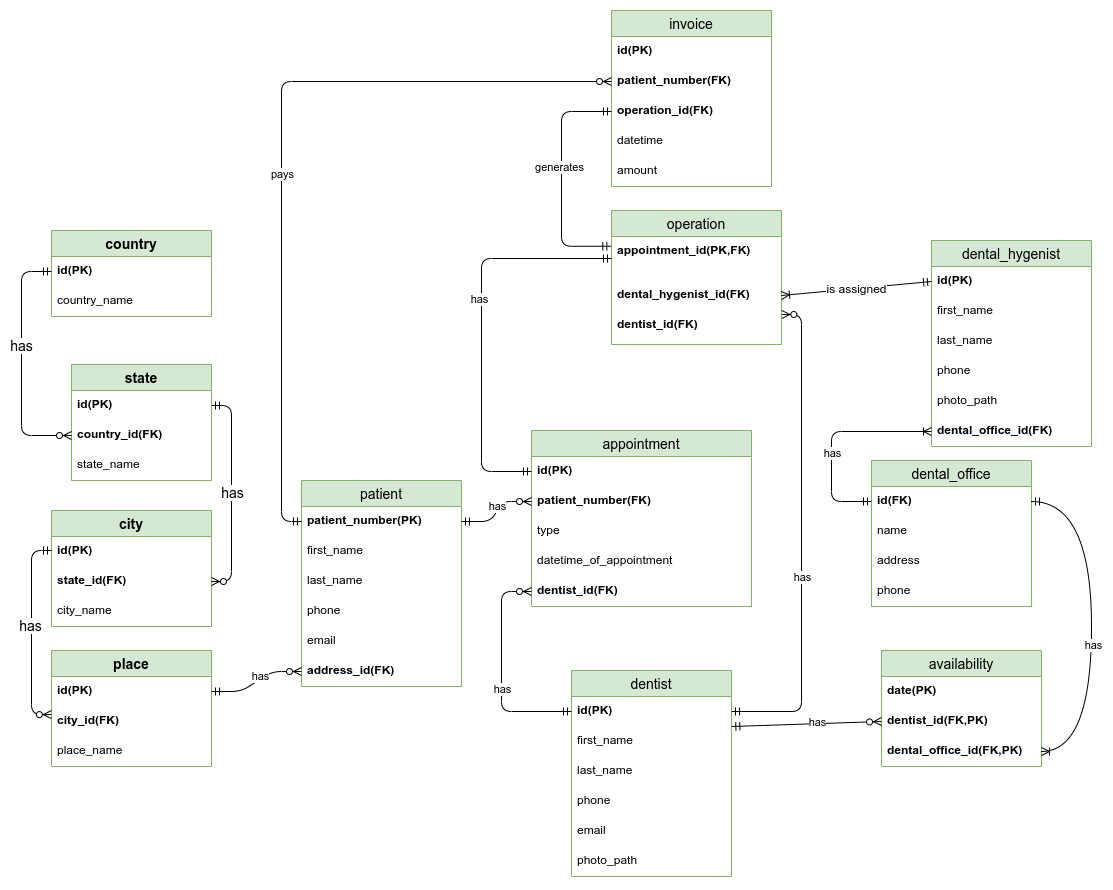
We will refine the work done in the conceptual phase and represent the data in detail as much as possible for building the basis for developing the physical mode.

We will also normalize different tables and represent it into more entities possible to maintain less redundancy in the data.

We already found the entities and their simple relationship .Now we will give the description of the entities , description of their attributes and the attributes domain.

| Entity | Description | Domain |
| --- | --- | --- |
| patient | The person who books for the appointment. |  |
| **Attributes:**  patient\_number  name  phone  email  address\_id | Identifier for user entity,,PK  First Name + Last Name  Valid phone number of patient  Valid email address of patient  FK which references the entity place of the patient. | Text  Phone number verification  Email verification  Valid id from place table. |
| place | The local street where the customer lives |  |
| **Attributes**:  id  city\_id  place\_name | Identifier of the street entity, SK,PK  FK which references the city.  The name of the place inside the city where the customer lives. | Auto generated  Valid id from table city  Text |
| city | The name of the city. |  |
| **Attributes**:  id  state\_id  city\_name | Identifier of the city entity, SK,PK  FK which references the state of the city.  The name of the city | Auto generated  Valid id from state table  Text |
| state | The name of the state. |  |
| **Attributes**:  id  country\_id  state\_name | Identifier of the state entity, SK,PK.  FK which references the country of the state.  The name of the state | Auto generated  Valid id from country table  Text |
| country | The name of the country. |  |
| **Attributes**:  id  country\_name | Identifier of the country entity, SK,PK  The name of the country | Auto generated  Text |
| dentist | The person who checks the person |  |
| **Attributes:**  id  name  phone  email  photo\_path | Identifier of the dentist entity,SK,FK  First\_name + last\_name  Valid phone number of dentist  Valid email of the dentist  The photo path of the dentist | Auto generated  Text  Text  Text  Text |
| dental\_office | The offices where the patient comes for services. |  |
| **Attributes:**  id  name  address  phone | Identifier of the entity,SK,PK  first\_name+last\_name  Valid address of the dental office  Valid contact number of the office | Auto generated  Text  Text  Text |
| availability | The entity showing the availability of the dentist at the dental office at the particular date. |  |
| **Attributes:**  date  dentist\_id  dental\_office\_id | Date for which the dentist is available to the particular dental office,PK  The id referencing the dentist, FK , PK  The id referencing the dental office,FK,PK | Valid Date  Valid id from dentist table  Valid id from dentist\_office table |
| dental\_hygenist | The person who is appointed to the operation of patient along with the dentist |  |
| **Attributes:**  id  name  phone  photo\_path  dental\_office\_id | Identifier of the dental\_hygenist entity,SK,PK  first\_name+last\_name  Valid phone number of the dental hygienist  The URL of the photo  ID referencing the entity dental\_office,FK | Auto generated  Text  Text  Text  Valid if from table dental office |
| appointment |  |  |
| **Attributes:**  id  patient\_number  type  datetime\_of\_appointment  dentist\_id | Identifier of the entity,SK,PK  ID referencing the entity patient , FK  The type of the appointment  Valid date and time of the appointment  ID referencing the entity dentist,FK | Auto generated  Valid id from patient table  Text  Timestamp  Valid id from dentist table |
| operation | The actual operation of patient who has the appointment at the particular time |  |
| **Attributes:**  appointment\_id  dental\_hygenist\_id  dentist\_id | ID referencing the appointment entity,FK,PK  ID referencing the dental\_hygenist,FK  ID referencing the dentist who performs the operation, FK | Valid id from table appointment  Valid id from table dental\_hygenist  Valid id from table dentist |
| invoice |  |  |
| **Attributes:**  id  patient\_number  operation\_id  datetime  amount | Identifier of the entity,SK,PK  Id referencing the patient with the invoice,FK  ID referencing the operation ,FK  The valid date and time of the invoice.  The total cost of the operation. | Auto generated  Valid id from table patient  Valid id from table operation  TIMESTAMP  Money |

*Fig:Table showing the entities,attributes,attributes description and attributes domain of the entities.*



**Verification**

Now , I am going to check whether each table satisfies a different Normalization Form(NF).

Here,All of the tables are already in 1NF as it has atomic data and no repeating groups.

So, if it satisfies the 1NF , I am going to check for the 2NF.

As every table has its own unique id as its primary key which is also a surrogate key, it already satisfies the condition of 2NF.

As the table is already in 2NF form, we can now check for 3NF conditions.

For the table not to satisfy the 3NF condition , there should be the transitive dependency . But looking at all of the mentioned tables I don’t see any non-prime attributes determining the other non-prime attributes.So,we can say there is no transitive dependency.

Hence, the tables are already in 3NF form.

**Physical modelling**

This is the part where I will physically implement the model into the database.For that purpose we are gonna use PostgreSQL relational database.It is an open source DBMS.

It uses the Standard Query Language(SQL) to create,update,alter ,delete the tables or any rows in the database.

Firstly , I made the dummy datas for each table with some tuples in it in the Excel sheet.

Then , I have used python and the driver of PostgreSQL which is Psycopg2 to push the

Data of .xlsx file into the Postgre DBMS.I have used Jupyter Notebook to write all of the code .

The physical implementation code of the E-comm is presented below in my github repository link:

LINK: <https://github.com/callingsandesh/Leapfrog-Assignment/blob/main/Data/week_2/weekend_assignment/dentist%20appointment%20(physical%20implementation).ipynb>

**References:**

1. *Draw.io (*[*https://app.diagrams.net/*](https://app.diagrams.net/) *)*
2. PostgreSQL ( <https://www.postgresql.org/> )
3. Psycopg2 ( <https://www.psycopg.org/docs/> )
4. Python Programming language (<https://www.python.org/> )
5. Jupyter Notebook (<https://jupyter.org/> )