T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

CSE 414 PROJECT

Hospital Management System

Muhammetalp ERDEM
151044006

Deadline: 06.06.2021

İÇİNDEKİLER

1. USER REQUIREMENTS

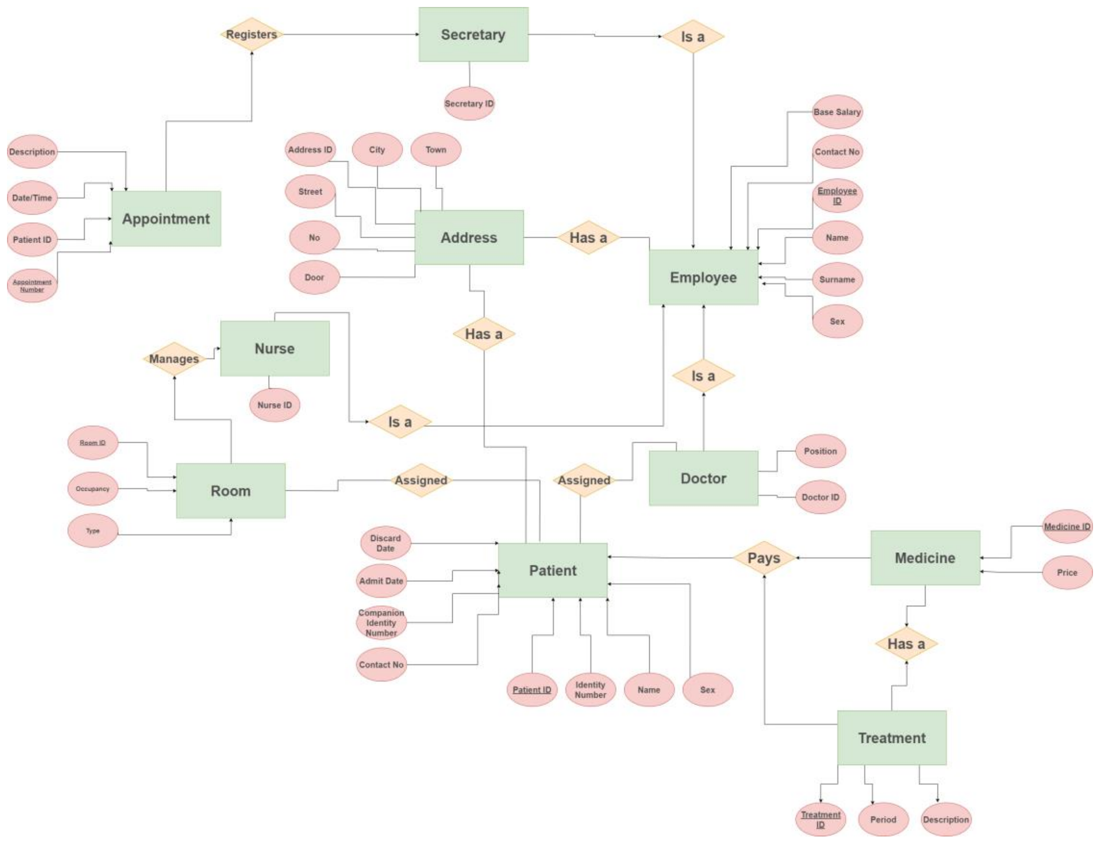
We need to aid the administration by keeping records of all the employees such as nurses and doctos as well as their information such as their name, address and phone number. The said information should be accessible all the time for emergency cases which should not be rare, considering this is a hospital.

The main customer/user of a hospital is patients, therefore, keeping record of the patients who registered and maintaining their data as their their threatment progresses is quite important in order to make the employees' life easier while rendering it possible to access historical data when the need for it arises.

Another issue concerning the administration is patient rooms and medicines which can be simplified as hospital equipment and expenses. Keeping track of these data is mandatory due to the need of reporting it monthly and being updated about the total and percentage expenses of the hospital.

Overall, a database is most certainly required for such a establishment indepentent of being public or private. Storing and updating data after every single process will reduce human error and the effects of slow communication. Yet it can be ahrd to keep up with without a constantly maintainin manager, therefore creating a robust system as well as automatic triggers will improve the everyday usage.

2. ENTITY RELATIONSHIP DIAGRAM



3. FUNCTIONAL DEPENDENCIES

1. Room(Room ID, Occupancy, Type)
    a. Room ID -> Type
    b. Room ID -> Occupancy

2. Appointment(Description, DateTime, Patient ID, Appointment ID)
    a. Appointment ID -> Patient ID
    b. Appointment ID -> DateTime
    c. Appointment ID -> Description
    d. DateTime, Patient ID -> Appointment ID

3. Employee(Employee ID, Contact No, Name, Surname, Sex, Base Salary, Address ID)
    a. Employee ID -> Contact No
    b. Employee ID -> Name
    c. Employee ID -> Surname
    d. Employee ID -> Sex
    e. Employee ID -> Base Salary
    f. Employee ID -> Address ID
    g. Contact No, Name, Surname -> Employee ID
    h. Contact No, Name, Surname, Sex, Base Salary, Address ID -> Employee ID

4. Patient(Patient ID, Identity Number, Name, Sex, Address ID, Contact No, Companion Identity Number, Admit Date, Discard Date)
    a. Patient ID -> Identity Number
    b. Patient ID -> Name
    c. Patient ID -> Sex
    d. Patient ID -> Address ID
    e. Patient ID -> Contact No
    f. Patient ID -> Companion Identity Number

   g. Patient ID -> Admit Date
   h. Patient ID -> Discard Date
   i. Identity Number -> Patient ID
   j. Identity Number -> Name
   k. Identity Number -> Sex
   l. Identity Number -> Address ID
   m. Identity Number -> Contact No
   n. Identity Number -> Companion Identity Number
   o. Identity Number -> Admit Date
   p. Identity Number -> Discard Date
   q. Companion Identity Number, Admit Date -> Identity Number, Name, Sex, Address, Contact No, Discard Date
   r. Companion Identity Number, Discard Date -> Identity Number, Name, Sex, Address, Contact No, Admit Date
   s. Companion Identity Number, Admit Date, Discard Date -> Identity Number, Name, Sex, Address, Contact No

5. Medicine(Medicine ID, Price)
   a. Medicine ID -> Price

6. Doctor (Employee ID, Patient ID, Position)
   a. Employee ID, Patient ID -> Position
   b. Employee ID -> Position

7. Address (Address ID, City, Town, Street, No, Door)
   a. Address ID -> City
   b. Address ID -> Town
   c. Address ID -> Street
   d. Address ID -> No
   e. Address ID -> Door

f. City, Town, Street, No, Door -> <u>Address ID</u>

8. Treatment (<u>Treatment ID</u>, Period, Description)
    a. <u>Treatment ID</u> -> Period
    b. <u>Treatment ID</u> -> Description

4. TABLES

1. Room (<u>Room ID</u>, Occupancy, Type)
2. Appointment (Description, DateTime, Patient ID, <u>Appointment ID</u>)
3. Employee (<u>Employee ID</u>, Contact No, Address ID, Name, Surname, Sex, Base Salary)
4. Patient (<u>Patient ID</u>, Identity Number, Name, Sex, Address ID, Contact No, Companion No, Admit Date, Discard Date, Room ID)
5. Medicine (<u>Medicine ID</u>, Price)
6. Doctor (<u>Employee ID</u>, <u>Patient ID</u>, Position)
7. Secretary (<u>Employee ID</u>, <u>Appointment Number</u>)
8. Nurse (<u>Employee ID</u>, <u>Room ID</u>)
9. Address (<u>Address ID</u>, City, Town, Street, No, Door)
10. Treatment (<u>Treatment ID</u>, Period, Description)

5. NORMALIZATION

1. 1NF
    a. Every attribute name in the table should be unique.
    b. Attribute types in the table should not change over time.
    c. Every attribute in the table should bear only one value.
2. 2NF

    a. The table should be in 1NF.
    b. The table should not include partial dependencies because partial dependency causes unneccessary data usage.

3. 3NF
    a. The table should be in 2NF.
    b. Dependencies in the table should respect at least one of these rules:
        i. A -> B is trivial.
        ii. A is a superkey of B.
        iii. Attributes derived from B – A should not be in any candidate key of the table.

4. BCNF
    a. The table should be in 3NF.
    b. Dependencies in the table should respect at least one of these rules:
        i. A -> B is trivial.
        ii. A is B’s superkey.

Room (<u>Room ID</u>, Occupancy, Type)

<u>Room ID</u> -> Type

<u>Room ID</u> -> Occupancy

Every single table of ours, satistify the 1NF due to having unique names and their types don’t change over time while having only one value.

In order to satisfy the 2NF standart, the table should be in 1NF while including no partial dependencies. Spotting a partial dependency is not a hard process, all we need to do is check if there are any foreign keys in the table that is not associated with the primary key.

Satisfying this condition as well, all tables are indeed in 2NF.

3NF requires the table to satisfy one of the additional three conditions. Our tables satisfy the a is b’s superkey part. Therefore the tables are in the 3NF.

BCNF is the most limiting standart but does not affect our tables because it only removes the third condition from the 3NF which we did not use, our tables already satisfy the second condition.

6. VIEWS

- Display Employee Information
  - Name
  - Sex
  - Contact No
- Display Patient Admitted to The Hospital
  - Patient Name
  - Sex
  - Admit Date
- Display current Appointments
  - Appointment ID
  - Appointment Date Time
  - Appointment Description
- Display Rooms that are available
  - Room ID
  - Room Type
- Display Treatments that are not finished
  - Treatment ID
  - Period

        - Treatment Description

7. TRIGGERS

- When a patient's discard date is assigned, empty the that he/she used to stay.
- When a doctor is removed, delete him/her from the employee table and assigned patient.
- When a patient is added, assign a room to him/her and mark the room full.
- When a treatment ends, decrease the stock of the assigned medicine.
- When a nurse is removed, he/she should removed from the assigned room as well.

8. ATOMIC TRANSACTION

- Removing the third trigger should be made atomically since it could cause assigning two patient to a single room which we should definitely avoid.
- The fourth trigger is at upmost importance since the stock of a medicine cannot go below 0 and that would cause some chaotic elements in the establishment, therefore, when removing this trigger it should be atomic.
- When removing an employee from the table, it should be made atomic since another query can read from another table without the knowledge that the employee is removed and assign him/her to somewhere which could cause untracable problems.

- When a nurse is removed, he/she should removed from the assigned room as well.

9. CONCURRENCY

MySQL inclused its very own snapshot isolation and lock mechanism and uses repeatable read as the default isolation level. When a transaction includes consistent reads then te snapshot from the first consistent read is used which means these transactions don’t take the lock of the table they’re accessing. Therefore the system lets the other transactions update the table, yet, this doesn’t cause any inconsistency because consistent reads use the first snapshot.