Phase-1: Hospital Database Management System

Team Name : 5heads

Member Name:

1. Dhruv Hirpara (2020102029)
2. Shreya Patil (2021121009)
3. Srija Narra (2020101094)

Introduction

The miniworld for this database is a Hospital. The hospital has employees that are organised hierarchically in departments, and patients that are treated by doctors and other medical personnel. The hospital must track these interactions and several others that arise from them.

Purpose

Hospitals, despite being a crucial part of our health care system, often have a primitive and inefficient filing system.
Prescriptions written on paper are often misinterpreted or misplaced. Hefty files are inefficient to use and error prone. This often leads to patients being overbilled or incorrectly medicated.
This Database provides an integrated solution to these problems. It is a comprehensive data management system that keeps track of all hospital operations, and provides transparency between the hospital, it's employees and it's patients.

Users

- Hospital Owners/Administrators : They are database Administrators who can access every functionality provided by the database.
- Med Staff : They can keep track of their patients and their treatment.
- Patients : They are able to keep track of all their treatments / No. days spent on the certain bed and bill.
- Department Heads : They are able to keep track of employees under them and hire/expel an employee .

Applications

This database is comprehensive yet easy to implement, and requires little to no cost to run and maintain. Hence it can be used in small and large medical facilities.
It provides a way to maintain records of the following :

- Hospital employees, their salaries, roles, and relationship with other employees of the hospital.
- Patients and any relevant details of the patient.
- The treatments administered to the patients by the Employees of the hospital.
- Hospital revenue and expenses in terms of employee salary.

Database Requirements

Entities

1. Patient
    a. PID: (Primary Key)
        i. Integer >0
        ii. NOT NULL
    b. Name : Complex Attribute
        i. First Name : - Varchar(40), NOT NULL
        ii. Last Name : - Varchar(40), NOT NULL
    c. Gender
        i. Char(1):- ‘M’ for ‘male’ and ‘f’ for ‘Female’ and ‘o’ for ‘Others’
        ii. NOT NULL

d. Date of Birth
    i. Date- MM/DD/YYYY
    ii. NOT NULL
e. Age : derived Attribute
    i. INTEGER>0
    ii. NOTNULL
f. Patient Phone no. : (Alternate key)
    i. INTEGER >0
    ii. NOT NULL
g. Patient Email Address (Alternate key) :
    i. VARCHAR(70)

h. Emergency Contact (composite attribute) :
    i. Consists of :
    Name - VARCHAR(50)
    Phone Number - INTEGER >0
    Age INTEGER> 18
    Relationship to patient VARCHAR(30)

2. Visit : weak entity of Patient
    a. VID (Partial Key):
        i. INTEGER > 0
    b. Visit Bill : derived attribute
        i. INTEGER > 0
        ii. NOT NULL
    c. Diagnosis :
        i. VARCHAR(50)
    d. Doctor's Report
        i. Large Text

    Subclass(Disjoint, total Participation):

        I. OutPatient
            A. Arrival Date
                1. Date- MM/DD/YYYY
                2. NOT NULL
            B. Arrival Time
                1. Timestamp HH:MM
                2. NOT NULL
            C. Departure Time
                1. Timestamp HH:MM

II. InPatient
    A. Arrival Date
        1. Date- MM/DD/YYYY
        2. NOT NULL
    B. Departure Date
        1. Date- MM/DD/YYYY

3. Treatment
    a. UID (Primary Key)
        i. Integer > 0
        ii. NOT NULL
    b. Name
        i. VARCHAR(60)
        ii. NOT NULL
    c. Type
        i. VARCHAR(30)
        ii. NOT NULL
    d. Cost
        i. INTEGER >=0
        ii. NOT NULL

4. Rooms
    a. Room no. (Primary Key)
        i. INTEGER > 0
        ii. NOT NULL
    b. On Floor no.
        i. INTEGER >= 0
    c. Room Capacity
        i. INTEGER > 0
    d. Room Type
        i. VARCHAR(20)

5. Beds :- Weak entity of Rooms
    a. Bed no. (Partial Key)
        i. INTEGER >0
    b. Bed cost per night
        i. > 0
        ii. NOT NULL

c. Is vacant
    i. 0 if Vacant, else 1
    ii. NOT NULL

6. Department:-
    a. Department Name (Alternate Key)
        i. VARCHAR(50)
        ii. NOT NULL
    b. Department ID (Primary Key)
        i. INTEGER >0
        ii. NOT NULL
    c. No. of Employees (derived Attribute)
        i. INTEGER > 0
        ii. NOT NULL

7. Employee
    a. Employee ID (Primary Key)
        i. INTEGER > 0
    b. Name; Complex Attribute
        i. First Name : - Varchar(40), NOT NULL
        ii. Last Name : - Varchar(40), NOT NULL
    c. Gender
        i. Char(1):- ‘M’ for ‘male’ and ‘f’ for ‘Female’ and ‘o’ for ‘Others’
        ii. NOT NULL
    d. Phone No.(Alternate Key)
        i. Integer >0
    e. Working time (Composite Attribute):
        i. IN-TIME : Timestamp HH:MM
        ii. OUT-TIME : Timestamp HH:MM
    f. Pay/Salary
        i. INTEGER >0
        ii. NOT NULL

    Subclass(overlapping,partial participation):

        I. Doctor :-
            A. Specialization : Multivalued Attribute
                1. Each value should be of VARCHAR(30)
            B. Qualifications : Multivalued Attribute
                1. Each value should be of VARCHAR(30)

II. Medical Staff
    A. Qualifications : Multivalued Attribute
        1. Each value should be of VARCHAR(30)

Weak Entity:

1. VISIT is a weak entity related to PATIENT.
2. BEDS is a weak entity related to ROOM.

Relationships

1. *Visits* (VISIT, PATIENT)
    - Identifying Relationship for VISIT
    - n = 2
    - (1, n) : A patient may be related to any number of VISITs.
      (1, 1) : A VISIT is related to 1 PATIENT

2. *Assigned to* (BED, INPATIENT)
    - n = 2
    - BED *assigned to* INPATIENT
    - (1, 1) : A BED must be *assigned to* at most 1 INPATIENT
      (1, 1) : An INPATIENT gets at most 1 BED

3. *Belongs to* (BED, ROOM)
    - Identifying Relationship for BED
    - n = 2
    - BED *belongs to* ROOM
    - (0, n) : A ROOM may have any number of BEDs
      (1, 1) : 1 BED must *belong to* in a particular ROOM

4. *Works in* (EMPLOYEE, ROOM)
    - This relationship maps EMPLOYEEs to their offices (ROOMs)
    - n = 2
    - EMPLOYEE works in a ROOM
    - (0, 1) : An EMPLOYEE must *work in* at most 1 office (ROOM)
      (0, n) : A ROOM may be the office of any number of EMPLOYEEs

5. *Supervises* (EMPLOYEE, EMPLOYEE)
    - n = 2
    - An EMPLOYEE *supervises* another EMPLOYEE
    - (0, n) : An EMPLOYEE *supervises* any number of EMPLOYEEs
      (1, 1) : An EMPLOYEE must be supervised by 1 EMPLOYEE

6. *Member of* (EMPLOYEE, DEPARTMENT)
    - n = 2
    - An EMPLOYEE is a *member of* a DEPARTMENT.
    - (1, n) : An EMPLOYEE must be a *member of* at least 1 DEPARTMENT
      (1, n) : A DEPARTMENT must have at least 1 EMPLOYEE.

7. *Manages* (EMOPOYEE, DEPARTMENT)
    - n = 2
    - A EMPLOYEE manages a DEPARTMENT
    - (0, n) : A EMPLOYEE *manages* any number of DEPARTMENTS
    - (1, 1) : A DEPARTMENT is managed by 1 EMPLOYEE

8. *Treats* (VISIT, DOCTOR, MEDICAL STAFF, TREATMENT)
    - n = 4
    - A MEDICAL STAFF *treats* a PATIENT(in a VISIT) with a TREATMENT under the supervision of (prescription of) a DOCTOR.
    - Has the Attributes
        - Charge
        - Date & Time
    - (1, n) : For a given instance of (VISIT, DOCTOR, TREATMENT), there has to be at least 1 MEDICAL STAFF.
      (1, n) : For a given instance of (VISIT, TREATMENT, MEDICAL STAFF), there has to be at least 1 DOCTOR.
      (1, n) : For a given instance of (VISIT, DOCTOR, MEDICAL STAFF), there has to be at least 1 TREATMENT.
      (1, n) : For a given instance of (DOCTOR, TREATMENT, MEDICAL STAFF), there has to be at least 1 VISIT (PATIENT).

n>3 Relationships:

The relationship '*treats*' has a degree = 4.
Another way to model this relationship is as 4 binary relationships and 1 weak entity
This weak entity (let's call it TRT_INST) will have the same attributes as the relationship '*treats*' (Charge, Date&Time).
The 4 aforementioned relationships will be the identifying relationships of TRT_INST. They are :

- Prescribed by
  TRT_INST is prescribed by a DOCTOR.
- Performed by
  TRT_INST is performed by a MEDICAL STAFF.
- Consists of
  TRT_INST CONSISTS of a TREATMENT.
- Performed on
  TRT_INST is performed on a VISIT (of a PATIENT).

Functional Requirements

Modifications

1. Insert:
   - insert <entity type> object (instances of any entity may be created)
2. Delete:
   - delete <entity type> object (instances of any entity may be deleted)
3. Update:
   - Update Treatment object
   - Update Visit bill attribute for a Patient
   - Update pay/salary attribute for an Employee
   - Update instance of Visit
   - Append to doctor's Report of a Patient's VISIT

Retrievals

1. Selection:
   - Select data tuples of all patients
   - select data tuples of all employees

2. Projection:
    - Select doctor(s) specialized in <specialization>
    - Select bed numbers for which ‘Is vacant’ = 0
3. Aggregate:
    - sum of cost of all the treatments taken by a patient
    - sum of revenue generated by the hospital through treatments and room charges
4. Search:
    - Search for <patient name> in Patient
    - Search for Patient VISITS with partial match of diagnosis.
5. Analysis:
    - Reports of total bill (room bill + cost for all the treatments taken) paid by patients
    - Reports of total amount of money given to all employees in the form of salary
    - Number of Patient Visits with a particular diagnosis in the last <Time period>
    - Number of TREATMENTS (with type = surgery) performed by <DoctorID>