Hospital Management System Database Project

By: Sasha Rider

* + 1. **Describe the enterprise:** This database is very practical and can be applied to a computer system in a hospital setting to keep track of the patients, staff, scheduling, billing, and hospital rooms. I wanted to do a project on healthcare as I have an associates in General Biology and am familiar with the operation of hospitals. I could see myself using this in a real-world application, so this is a great introduction to later move on to more complex databases which could manage multiple hospitals, etc. Without the precise organization of medical records, there would be chaos and people would not be treated as efficiently.
    2. The system should perform various functions such as keeping track of patients, their diseases and treatment required, which doctor/nurse(s)/ward boy(s) is treating the patient(s), and which room they are staying in. Based on the number of days and the treatment received, a bill should be generated.
    3. I chose to create a hospital management database system as this was listed as one of the options for the project in the handout and it stood out to me as interesting and useful.
  1. **Entity Relationship Design:**
     1. **Entities:**  
        1. Patient – this is the person who is being treated and is admitted at the hospital. Their records are important for billing reasons, as the number of days they have stayed and treatment received will be recorded.

2. Doctor – this is the doctor(s) treating the patient at the hospital. Doctor’s will have specialty and will be assigned to patients accordingly.

3. Nurse – this is the nurse(s) assisting the doctor in treating/caring for the patient. A record of nurses is important so that there are enough nurses who can assist the doctors in caring for the patients.

4. WardBoys – this is the ward boy(s) who conduct hospital maintenance and patient care. A record of them is needed in order to assign shifts proportionally to the patient’s needs so the hospital can be well maintained.

5. Treatment- this is the management/care to combat the patient’s disease, the people involved in treating the patient will be looking at this. This is also important for billing.

6. Disease- this is the sickness/injury that the patient is sick with. Doctors and nurses will need to know this in order to treat the patient. Treatment is dependent upon disease.

7. Hospital- this is the medical facility providing care. In case records get lost, it is important to know which hospital they belong to so they are easier to trace.

8. Rooms- these are the rooms that are inside of the hospital, the staff needs to know which room patients are staying in.

9. ICU- These are the intensive care unit rooms. These should be separate from the other rooms as they require specialized medical/nursing care. Nurses and doctors need to have access to the ICU rooms to see where they are needed.

10. OperationTheaters- These are the operation rooms where surgeries happen. These are also a different kind of room and require their own entity. These are important for doctors who are surgeons as they need to know which room their surgery is assigned to.

11. Patient\_has\_WardBoys – This is an intersection table for ward boys and patient as multiple ward boys can have multiple patients.

12. Patient\_has\_Nurse – This is an intersection table for nurse and patient as multiple nurses can have multiple patients.

**1.2.2 Describe your relationship**

**1.2.2.1 Relationships**

1. Patient may suffer from 1 or more diseases 1:N

2. There is one or more treatment for each disease (in this scenario). 1:N

3. One doctor may have many patients 1:N

4. Many nurses may have many patients N:M

5. A hospital has many nurses 1:N

6. A hospital has many patients 1:N

7. A hospital has many doctors 1:N

8. A hospital has many ward boys 1:N

9. Many ward boys take care of many patients N:M

10. A hospital has many rooms 1:N

11. Operation theaters may have many rooms 1:N

12. ICU may have many rooms 1:N

**1.2.2.2 Minimum & Maximum cardinality**

1. Patient has a minimum cardinality of one and a maximum cardinality of many

2. Doctor has a minimum cardinality of one and a maximum cardinality of many

3. Nurse has a minimum cardinality of one and a maximum cardinality of many

4. Ward boys has a minimum cardinality of one and a maximum cardinality of many

5. Treatment has a minimum cardinality of one and a maximum cardinality of many

6. Disease has a minimum cardinality of zero and a maximum cardinality of many

7. Hospital has a minimum and maximum cardinality of one

8. Rooms has a minimum cardinality of one and a maximum cardinality of many

9. ICU has a minimum cardinality of one and a maximum cardinality of one

10. OperationTheaters has a minimum and maximum cardinality of one

11. Patient\_has\_WardBoys has a minimum cardinality of one and a maximum cardinality of many

12. Patient\_has\_Nurse has a minimum cardinality of one and a maximum cardinality of many

**1.2.2.3 Weak and Strong Entities**

1. OperationTheaters is a non-ID dependent weak entity

2. ICU is a non-ID-dependent weak entity

3. Patient\_has\_WardBoys is an ID-dependent weak entity

4. Patient\_has\_Nurse is an ID-dependent weak entity

5. Patient is a strong entity

6. WardBoys is a strong entity

7. Nurse is a strong entity

8. Doctor is a strong entity

9. Treatment is a strong entity

10. Disease is a strong entity

11. Hospital is a strong entity

12. Rooms is a weak entity

**1.2.2.4 Derived Attribute**

Don’t put derived attribute in E-R diagram, must show derived attribute in SQL statement (so something like age can be derived from DOB)

**1.2.2.5 Composite Attribute**

1. In Patient\_has\_WardBoys PatientID and WardBoyID act as composite keys

2. Similarly, in Patient\_has\_Nurse PatientID and NurseID act as composite keys

* + - 1. **Multivalued Attribute**

1. Symptoms in disease table is a multivalued attribute as a patient can have more than one symptom
2. DiseaseName in treatment table is also multivalued as a patient can suffer with multiple diseases.

**1.2.2.7 Total Participation**

Mandatory participation (each entity in an entity set must participate in at least one relationship instance in that entity set)

1. One hospital may admit one or more patients
2. One hospital is maintained by one to many ward boy(s)
3. One hospital is maintained by one to many nurse(s)
4. One hospital has employed one or more doctors
5. One hospital has one or more rooms
6. One or more rooms belong to the ICU
7. One or more rooms belong to Operation Theaters
8. One doctor may handle one or more patients
9. One or more patients may be treated by one doctor
10. Many nurses take care of many patients, many patients are taken care of by many nurses
11. Many ward boys take care of many patients, many patients are taken care of by many ward boys
12. One patient can suffer from zero or more diseases
13. Every disease has one or more treatments, every treatment has one disease

**1.2.2.8 Recursive Relationship**

Doctor has a recursive relationship to itself, as a doctor can refer another doctor who can refer another doctor, etc. One doctor can refer another doctor so it is 1:1

* 1. **Conceptual Level**

**1.3.1 Primary Keys**

1. Hospital – HospitalName

2. Patient – PatientID

3. Doctor – DoctorID

4. Nurse – NurseID  
5. WardBoys – WardBoyID

6. Disease – DiseaseName

7. Treatment – TreatmentID

8. Rooms – Primary composite key of RoomNumber, Op\_Theater\_ID, and ICUID

9. OperationTheaters – Op\_Theater\_ID

10. ICU – ICUID

**1.3.2 Foreign Keys**

1. Patient – HospitalName, DoctorID

2. Disease – PatientID

3. Treatment – DiseaseName

4. Doctor – HospitalName, DoctorRef (recursive)

5. Patient\_has\_Nurse – PatientID, NurseID (Intersection table so composite primary key is made up of foreign keys)

6. Patient\_has\_Wardboys – PatientID, WardBoyID (another intersection table)

7. Nurse – HospitalName

8. WardBoys – HospitalName

9. Rooms – Op\_Theater\_ID, ICUID, HospitalName

**1.3.3 Column Domains**

Column domains are the data types associated with the tables, they are listed next to the column names

**1.3.4 Domain Integrity Checks**

Domain integrity checks ensure that there are rules which restrict the format, type and volume of data rerecorded in the database

**1.3.5 Functional Dependencies**

Constraints (this includes foreign keys), a relationship between two attributes