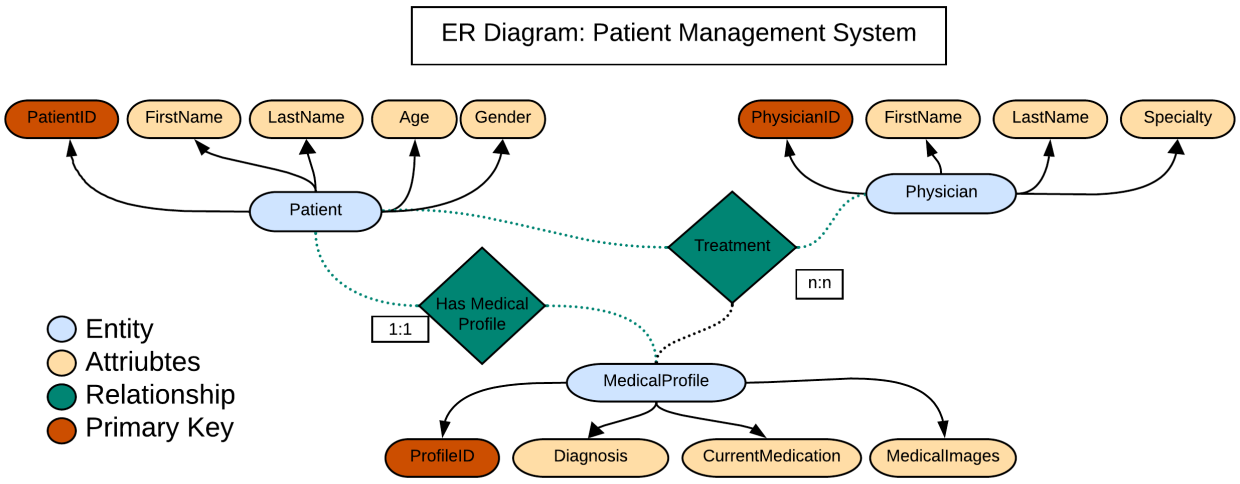
Advanced Databases Project - Assignment 2

Team: Sifael Sebastian Ndandala
Email: sifael.ndandala@gmail.com
Deadline: 12 März 2024, 23:59

Mapping Approach for Patient Management System Database

The Patient Management System database consists of four tables designed to manage information regarding patients, physicians, medical profiles, treatments, and access records. This document describes the mapping approach for each table, illustrating how they interrelate to efficiently store and retrieve health-related data.



1. Patient Entity

The `Patient` table serves as the foundation for patient-related information. It includes basic personal details of patients such as their ID, first name, last name, age, and gender. The `PatientID` acts as a unique identifier (primary key) for each patient, ensuring that each

record represents a distinct individual within the healthcare system.

2. Physician Entity

The `Physician` table stores information about healthcare professionals, including their ID, first name, last name, and specialty. The `PhysicianID` is a unique identifier (primary key) for each physician, facilitating the association between physicians and their areas of expertise, patients they are treating, and access to patient records.

3. Medical Profile Entity

The `MedicalProfile` table links directly to the `Patient` table via the `ProfileID`, which is a foreign key referencing `PatientID`. This design establishes a one-to-one relationship between each patient and their medical profile. The table includes detailed medical information such as diagnosis, current medication, and medical images. The use of `ProfileID` as both a primary key and a foreign key to `PatientID` simplifies the process of accessing a patient's medical history directly through their unique identifier.

4. Treatment Relationship

The `Treatment` table captures the many-to-many relationship between patients and physicians, indicating which physicians are treating which patients. This is achieved through composite primary keys (`PatientID`, `PhysicianID`) and corresponding foreign keys that reference the `Patient` and `Physician` tables. This setup allows for the recording of multiple physicians treating a single patient and vice versa, without duplicating patient or physician details.

2. Relationship Mapping and Variants

1. Patients to MedicalProfile

The Patient to Medical Profile data has a one-to-one relationship based on `PatientID` and `ProfileID`. The mapping of this relationship can be achieved directly with primary and foreign keys.

```
SELECT
    P.PatientID,
    P.FirstName,
    P.LastName,
    P.Age,
    P.Gender,
    MP.Diagnosis,
    MP.CurrentMedication,
    MP.MedicalImages
FROM
    Patient P
JOIN
    MedicalProfile MP
ON
    P.PatientID = MP.ProfileID;
```

2. Patients and Physicians through `Treatment` Table

The Patient to Physician is a many-to-many relationship mapping. It can be achieved through the `Treatment` relationship which maps individual `Patient` with `Physician`.

```
SELECT
    P.PatientID,
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName,
    Phy.PhysicianID,
    Phy.FirstName AS PhysicianFirstName,
    Phy.LastName AS PhysicianLastName,
    Phy.Specialty
FROM
    Patient P
JOIN
    Treatment T
ON
    P.PatientID = T.PatientID
JOIN
    Physician Phy
ON
    T.PhysicianID = Phy.PhysicianID;
```

3. Physician To Medical Records

Finally, the Physician To MedicalProfile Mapping can happen directly through the Treatment

Relationship. This removes the need for an extra relationship table.

```
SELECT
    Phy.PhysicianID,
    Phy.FirstName,
    Phy.LastName,
    Phy.Specialty,
    MP.Diagnosis,
    MP.CurrentMedication,
    MP.MedicalImages
FROM
    Physician Phy
JOIN
    Treatment T
ON
    Phy.PhysicianID = T.PhysicianID
JOIN
    MedicalProfile MP
ON
    T.PatientID = MP.ProfileID;
```

3. ISA Hierarchies

The nature of the entity design of my project and therefore the database lends itself to a non-superclass and sub-classes as the entities are independent of each other. Patients are independent of Physicians and viceversa and do not belong to a higher SuperClass. Therefore, ISA Hierarchies for this project are not reasonable to create.