COMP 5120/6120/6126
Database Systems I
July 29, 2011

Database Design Project

available at:
https://oitappstest.auburn.edu/BayView/Home.aspx

by:
Jared Adams
Patrick Carpenter
Heath Hopkins
Samantha Knight

Table of Contents

Analysis

Introduction

This section details the group's interpretation of the high-level design requirements as outlined in the *Problem Description* and *Problem Scope* sections of the *Project Description* document, realized in the form of a conceptual diagram. First, constraints and assumptions arising from the interpretation of the problem description are noted. Then, a conceptual diagram is given which seeks to capture these requirements using precise notation. Finally, assumptions concerning the intended use of the database system which justify the omission of certain details from the domain are stated.

Domain Analysis

An examination of the problem description led the group to define the following entities:

- **Patient**: A record of a person having received treatments in the hospital for which the patient still needs to pay;
- **Bed**: A record of the bed in which patients may reside during their stay at the hospital;
- **Employee**: A record of a person working for the hospital, who may assist in the delivery of treatments;
- **Treatment**: A procedure which may be carried out by employees on patients.

These entities were found to be related as follows:

- **Patients** are related to **Rooms**, in that they are assigned to a **Bed** inside a room. Each **Patient** is assigned to exactly one **Room**, whereas the number of **Patients** assigned to a given **Room** can be zero, one, or two (for a fully-occupied semiprivate room).
- **Patients**, **Employees** and **Treatments** are related by the fact that **Employees** administer **Treatments** to **Patients**.

These entities and relationships constitute the core of the conceptual model. Attributes were determined by identifying attributes in the sample reports and associating these with the appropriate entity and/or relationship. It is important to note that there are several other entities and relationships which might have been included in the ER model, but which were omitted. Some of the most noticeable omissions are motivated as follows:

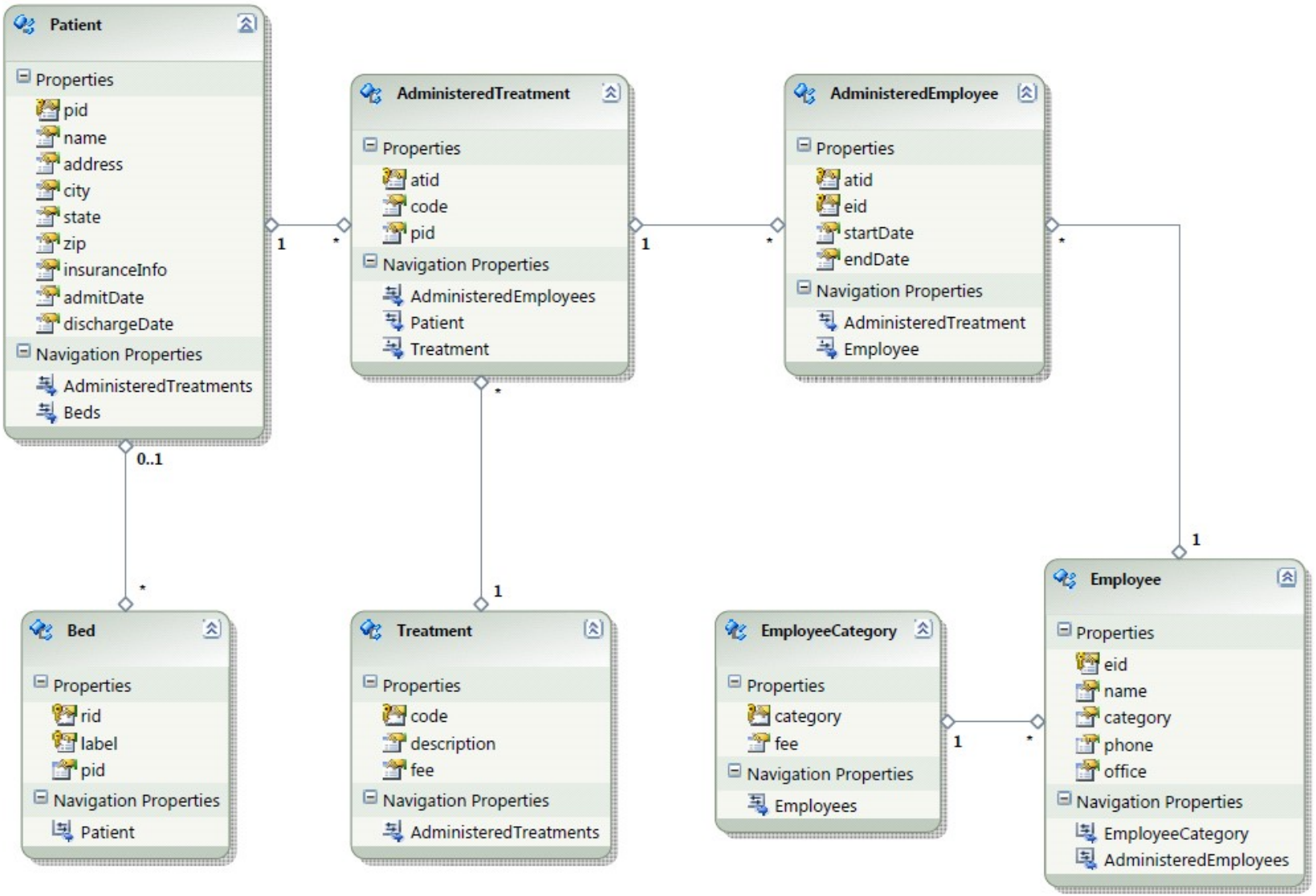
- **PrimaryDoctor** as a relationship: since none of the reports requires information concerning any **Patient**'s **PrimaryDoctor**, this relationship is deemed unnecessary. The definition of additional reports requiring this information would necessitate a redefinition of the data model to record this information.
- Constraints governing ordering and performance of treatments: these constraints were deemed complex, incomplete, and unnecessary at the data definition level.

Conceptual Diagram

A conceptual diagram corresponding to the text description in the previous subsection is given on the next page. In it, a notation similar to entity-relation (ER) diagramming notation is used, where:

- Boxes are used to represent entities and N:M relationships;
- 1:N and 1:1 relationships are represented as foreign key attributes;
- Attributes are shown inside the entity/relationship box to which they refer;
- Key attributes are shown with a key symbol;
- Multiplicities are shown as annotations on the connecting lines.

Note: A relationship and one entity (EmployeeCategory) have been added to the diagram in an attempt to maintain consistency between set of relations defined in the next section and the conceptual model. Additionally, two relationships – AdministeredTreatment and AdministeredEmployee – are added in order to correctly model the complicated ternary relationship between Employees, Patients and Treatments.


Patient
Properties
pid
name
address
city
state
zip
insuranceInfo
admitDate
dischargeDate
Navigation Properties
AdministeredTreatments
Beds
AdministeredTreatment
Properties
atid
code
pid
Navigation Properties
AdministeredEmployees
Patient
Treatment
AdministeredEmployee
Properties
atid
eid
startDate
endDate
Navigation Properties
AdministeredTreatment
Employee
Employee
Properties
eid
name
category
phone
office
Navigation Properties
EmployeeCategory
AdministeredEmployees
EmployeeCategory
Properties
category
fee
Navigation Properties
Employees
Bed
Properties
rid
label
pid
Navigation Properties
Patient
Treatment
Properties
code
description
fee
Navigation Properties
AdministeredTreatments
1
*
0..1

Design

Introduction

This section details the group's transformation of the high-level conceptual model developed in the previous section into a set of suitable relation schemas. First, the translation process is briefly described. Then, a set of relation schemas is given which constitute a logical database design corresponding to the conceptual model. Finally, these schemas' adherence to Boyce-Codd Normal Form (BCNF) is investigated and the normalization process used to ensure compliance (where appropriate) is summarized.

Conceptual Diagram Translation

The conceptual model developed in the previous section contains five relatively simple entities. Translation of these into relations is accomplished by considering them to be tuples over their attributes.

The conceptual model developed in the previous section contains (a) a single one-to-one (**Employee**/**EmployeeCategory**) binary relationship, (b) a generally complex many-to-many-to-many (**Treatment**/**Patient**/**Employee**) ternary relationship, and (c) a one-to-many (**Room**/**Patient**) binary relationship. Case (a) is handled in a fairly simple fashion by allowing **Employee** to refer to its category, which can be used as a foreign key to access the appropriate entry in **EmployeeCategory**. Case (b) is handled by creating a table associating **Patient**/**Treatment** pairs with the (potentially many) **Employees** administering the **Treatment** to the **Patient**. Case (c) is handled by allowing **Patient** to refer to the **Room** to which it has been assigned.

These considerations are sufficient for translating the conceptual diagram developed in the previous section.

Relational Schemas

Relational schemas are given below. Primary keys are underlined and bolded and foreign keys appear in bold italics.

Patient(**pid**, name, address, city, state,
zip, insuranceInfo, admitDate, dischargeDate)

Bed(**rid, label**, ***pid***)

Treatment(**code**, fee, description)

Employee(**eid**, name, ***category***, phone, office)

EmployeeCategory(**category**, fee)

AdministeredTreatment(**atid, *pid, code***)

AdministeredEmployee(***atid, eid***, startDate, endDate)

Normalization

EmployeeCategory was created in order to normalize **Employee** into BCNF. **EmployeeCategory** is guaranteed to be in BCNF as it consists of only two columns. **Treatment** is in BCNF since the only functional dependencies are on the kind of treatment (referred to by the associated code). **AdministeredEmployee** is in BCNF since its attributes are fundamentally unrelated foreign keys. **AdministeredTreatment** is in BCNF since the atid determines all other attributes, pid and code are unrelated foreign keys, and neither of these determines the date on which the treatment was administered.

Under reasonable assumptions, **Patient** is most likely not in BCNF. This is due to the fact that the Patient's address contains information about the city, state, and zip code; depending on the rules governing the locality of zip codes, these could be used to determine the state and, possibly, the city. However, it has been deemed acceptable to keep this redundancy since (a) this information is required by at least one report and (b) fixing this would require significant unrelated work in determining zip code to city/state mappings, which is not worth the extra complexity and effort.

Implementation

Table Creation Scripts

See the file "Database-Tables.sql" for commands used to create the tables used in the provided implementation.

Data Initialization Scripts

See the file "Database-Data.sql" for commands used to populate the tables used in the provided implementation.

Web Application Source

See the ZIP archive "Website (ASP.NET MVC 3 + C# .NET 4).zip" for the web application which provides access to the database. This implementation is deployed at https://oitappstest.auburn.edu/BayView/Home.aspx.

Test Output

See the file "Website Screenshots.docx" for screenshots of the database and application deployed at https://oitappstest.auburn.edu/BayView/Home.aspx.