Hudson River Psychiatric Center and 3NF Consulting, Inc. present

Developing Cross-Departmental Database Applications

with the

HRPC Patient and Staff Management System
Design

• Group the analysis results into collections of attributes and then form entities.

• Normalize those entities (tables)
  – We first need to identify the functional dependencies **within** the entities so we know how to decompose large tables into normalized smaller ones.
Design

• Determine the functional dependencies within the tables
  – patientId --> C number, name, gender, DOB, ward/unit, Medicaid number, etc.
    • Note: patientID does not functionally determine relative values because there can be many relatives per patient.
  – staffId --> name, position, shift, FTE, seniority date, tier, etc.
    • Note: staffID does not functionally determine URC values.
Design

- Unnormalized Patient attributes

Unnormalized patients table with relatives

<table>
<thead>
<tr>
<th>Patientid</th>
<th>Lastname</th>
<th>Firstname</th>
<th>Cnumber</th>
<th>Rel_last</th>
<th>Rel_first</th>
<th>Rel_addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Smith</td>
<td>John</td>
<td>5544556</td>
<td>Smith</td>
<td>Mary</td>
<td>123 Elm St.</td>
</tr>
<tr>
<td>101</td>
<td>Smith</td>
<td>Jack</td>
<td>5544556</td>
<td>Smith</td>
<td>Frank</td>
<td>123 Elm St.</td>
</tr>
<tr>
<td>102</td>
<td>Doe</td>
<td>Jane</td>
<td>4455443</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>Jones, Jr.</td>
<td>Gerrald</td>
<td>3322112</td>
<td>Jones</td>
<td>Gerrald</td>
<td>10 W Main.</td>
</tr>
</tbody>
</table>

Problems
Duplicate person record needed to store both relatives
Inconsistent first names for the same person
We cannot delete Gerrald Jones, Jr.’s relative without deleting Gerrald himself
Design

- Decompose by functional dependencies and their determinants to form normalized tables

Normalized Patients table and Relatives table

<table>
<thead>
<tr>
<th>Patients</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Patientid</td>
<td>Lastname</td>
<td>Firstname</td>
<td>Cnumber</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>Smith</td>
<td>John</td>
<td>5544556</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>Doe</td>
<td>Jane</td>
<td>4455443</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>Jones, Jr.</td>
<td>Gerald</td>
<td>3322112</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relatives</th>
<th>Relativeid</th>
<th>Rel_last</th>
<th>Rel_first</th>
<th>Rel_addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>R01</td>
<td>Smith</td>
<td>Mary</td>
<td>123 Elm St</td>
</tr>
<tr>
<td>101</td>
<td>R02</td>
<td>Smith</td>
<td>Frank</td>
<td>123 Elm St</td>
</tr>
<tr>
<td>103</td>
<td>R03</td>
<td>Jones</td>
<td>Gerald</td>
<td>10 W. Main</td>
</tr>
</tbody>
</table>

Tables broken up by similar items and related with a common key (PatientID). PatientID is the primary key of the Patients table. It’s a foreign key in the Relative table which lets us connect relatives to patients.
Design

• Identify the primary key for each table
  – definition: an attribute (column) or collection of attributes that uniquely identifies every row in the table.

• Associate the tables with foreign key-primary key relationships.
  – definition: an attribute or collection of attributes in one table which must exist in another table or be null.
Design

• Normal forms
  – Measure of the degree of normalization of your database
  – More normalized = Higher data integrity
  – Higher data integrity = LESS ERRORS
  – Third normal form is the goal of every database
Design

Third Normal Form:

All attributes (columns) are dependent on the key, the whole key, and nothing but the key. (So help me, Codd!)
Design

• Relationships and E-R diagrams
  – Document the relationships between the tables (which are represented through your primary and foreign keys) with an Entity-Relationship diagram.
    • Rectangles for the entities (tables)
    • Lines for the relationships
    • Labels on the lines to document the primary key / foreign key links
Design

• Check constraints
  – Used to enforce legal data entry values
  – In a SQL create table statement:
    • `Shift` varchar(7) default 'Day'
      check (Shift='Day' or Shift='Evening' or Shift='Night')

• Iterations and changes
  – Filters back to earlier design steps.
  – Applications are never done, they’re just due.