2021

Database Project Report

Máire Murphy
G00375722
4/8/2021

Contents

Introduction

The *dentist_db* database is designed to store patient booking information, treatment history performed by the Mulcahy practice and any specialist treatment history. Also, the data base stores billing and payment information.

Table 1: dentist_db Tables

Tables	Explanation
Patient, *Address,* *Patient_Phone_No*	Store information for the patient. Also stores an amount owed by the patient (if any) and how many days overdue (if any). There can be more than one phone number per address i.e. family of patient
Patient_Chart_Details	Contains treatment history – linked to Appointments and Treatment Guide tables
Appointments	Stores administration info about the appointment: date and time, for what patient, reminder sent info, type of appointment i.e. first visit
Bill	stores status of bill (open or closed), amount owed, balance owed, amount paid
Bill_Items	Contains inks to the Patient_Chart_Details table to pull itemized information for a bill (linked to Bill)
Treatment_Guide	Contain information about treatments that Mulcahy practice offers
Patient_Specialist_Treatments	Stores treatment history by other specialist dentists. Keeps track of date referral sent
Dentists	Stores dentist ID, name, their category (specialist or regular i.e. Dr. Mary MulCahy) and links to Office.
Office	Stores address and phone information for Specialist Dentist and Mulcahy's Practice. Also stores setting information for the MulCahy practice: cancellation fee, thresholds for amount due\day overdue
Payments	Stores information of payments made by the patient. Payment type (instalment\full), method of payment (card, cash, cheque) links to a bill

Normalization

Normalization was an iterative process that was performed during ERD modelling. Later other normalization non-compliance issues were identified when test data was inserted into the database and when SQL queries were executed. For example, redundancy was identified when parent and children were added to the *Patient* table. The address was duplicated. This may have caused an update\insertion anomaly.

1NF

Repeating groups have been put them into records of their own. Examples of tables specially created to hold detail info are *Bill_Items* and *Patient_Chart_Details*.

2NF

The database complies to 1NF. All non-key data items are fully functionally dependent on the primary key. Individual items in all tables can be retrieved using the primary key. For example, *notes* recorded for one treatment can be retrieved using the primary key *chart_item_ID*.

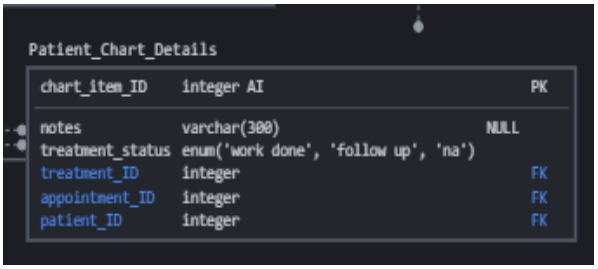

Figure 1: Patient_Chart_Details Table

3NF

The database complies with 2NF and no non-key data item is functionally dependent on any other-non-key data item. An example is where the patient address is functionally dependent on the postcode. New *Contact_Details* and *Patient_Contact*_Details were created to conform to 3NF. The patient's postcode will find a unique address and a phone number.

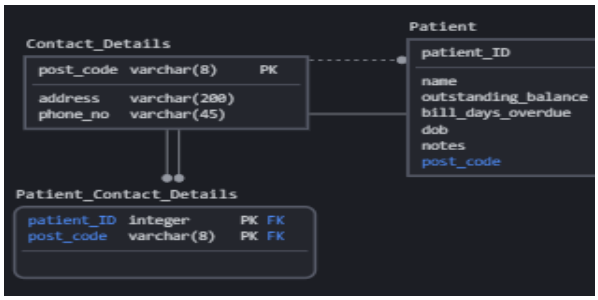

Figure 2: 3NF Example

BCNF

The database complies with 3NF and for every functional dependency X->Y, X should be the super key of the table. The use case here is that an address can have more than one phone number. A minor patient at one address may have several phone numbers (guardian\parent) for that address. In order to comply to BCNF, phone number is now in a new table called *Patient_Phone_No*.

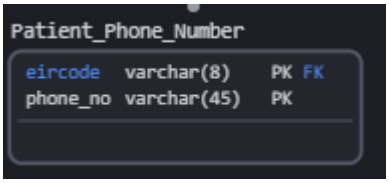

Figure 3: BCNF Example

Future Proofing

Dr. Mary Mulcahy is currently the only dentist in the practice. However, she may expand in the future and hire more dentists, or she may even just need a temporary replacement. A new dentist working for Dr. Mulcahy can easily be added to the database. The system includes the dentist who performed the work in the patient history. If another branch of the Mulcahy practice should open in another town; the database can handle a new local office being added.

A miscellaneous fee option was included in the *Office* table for unexpected costs such as Personal Protective Equipment.

Implementation Instructions

To implement the dentist_db:

1. Create a new data base using phpMyAdmin.
2. Import the dentist_db_structure_G00375722.sql.
3. To import the data in the import screen, select dentist_db_data_G00375722.sql and **Disable foreign key checks.** Then click 'Go'**.**