Medical Database

Ricardo Barbosa
Lindsey Reynolds
Dan Sedano

Tuesday, February 11, 2020

The Helping Otters design team is at the edge of leading techniques, which will ensure reliable and efficient database management for your medical needs. In this document, we have recorded all components of the database's design through an ER diagram and description. We state our goals for the project and how we plan to create the most efficient and easy-to-maintain database as possible. We also state our major findings and unresolved design issues.

ER Diagram Description

- The following diagram maps out a medical database and indicates the relationships between each table. All relationships, primary and foreign keys are indicated on the ER diagram.
- Indices, however, are not indicated on the ER diagram. We have placed indices on the patient **last_name** and **age** columns, as well as the doctor's **last_name** and **specialty** columns. We chose to do this because these columns will be accessed through searches often, and their tables will not be updated frequently.

Goals

- To make the database fast and efficient while providing the necessary information.
- To create a database that links patients to doctors, doctors to prescriptions, prescriptions to their pharmaceutical companies, and pharmaceutical companies to contracts with pharmacies that sell their drug.
- To configure cardinalities based on foreign and primary keys.

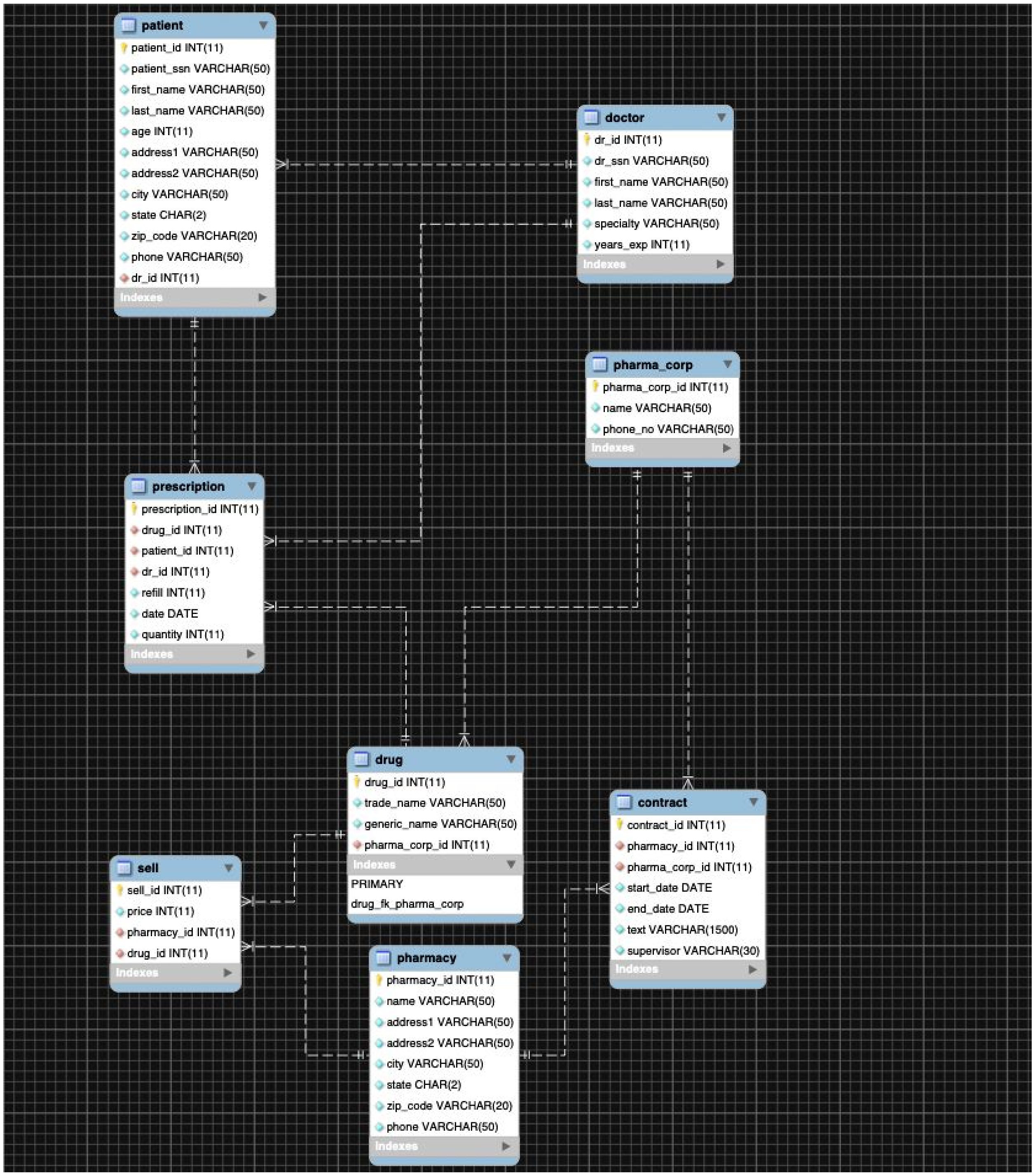
Resolved Findings

- Indexed most frequently accessed information, such as **last_name** and **age** for patients and **last_name** and **specialty** for doctors. It is also unlikely that these tables will be updated frequently.
- In order to normalize our tables, we decided to break down the patient and doctor **name** column into **first_name** and **last_name** columns. We also decided to break down the patient and pharmacy **address** columns into 5 single-value columns.
- Discovered the relationships between tables, deciphering whether each should be a one-to-one, one-to-many or many-to-many relationship. This included a one-to-many relationship from drug to prescription since we concluded that one drug can have many prescriptions for many patients.
- We decided to use the same naming conventions across all tables as ambiguity would be eliminated when adding the table name to our queries e.g., **patient.first_name** and **doctor.first_name**.

Unresolved Findings (Design Issues)

- Should we create a separate table for the **supervisor** in the **contract** table because the person supervising the contract would most likely need to have a name and contact information associated in case issues with the contract need to be resolved?
- We took the creative liberty to split **name** and **address** into multiple columns; however, we are unsure about whether or not to change the **age** column. If we convert the **age** column to a **birth_date** column and then derive the actual age at the time that the information is called from the database, this may save on processing time overall. Since we have an index at the **age** column, it does not make sense to store a patient's age rather than their birth date, as this would mean many rows would need to be updated and reindexed daily. Should age be removed as a column and instead be replaced with **birth_date;** allowing a query to calculate the patient's age?

Entity-Relationship (ER) Model


patient
patient_id INT(11)
patient_ssn VARCHAR(50)
first_name VARCHAR(50)
last_name VARCHAR(50)
age INT(11)
address1 VARCHAR(50)
address2 VARCHAR(50)
city VARCHAR(50)
state CHAR(2)
zip_code VARCHAR(20)
phone VARCHAR(50)
dr_id INT(11)
Indexes
doctor
dr_id INT(11)
dr_ssn VARCHAR(50)
first_name VARCHAR(50)
last_name VARCHAR(50)
specialty VARCHAR(50)
years_exp INT(11)
Indexes
pharma_corp
pharma_corp_id INT(11)
name VARCHAR(50)
phone_no VARCHAR(50)
Indexes
prescription
prescription_id INT(11)
drug_id INT(11)
patient_id INT(11)
dr_id INT(11)
refill INT(11)
date DATE
quantity INT(11)
Indexes
drug
drug_id INT(11)
trade_name VARCHAR(50)
generic_name VARCHAR(50)
pharma_corp_id INT(11)
Indexes
PRIMARY
drug_fk_pharma_corp
contract
contract_id INT(11)
pharmacy_id INT(11)
pharma_corp_id INT(11)
start_date DATE
end_date DATE
text VARCHAR(1500)
supervisor VARCHAR(30)
Indexes
sell
sell_id INT(11)
price INT(11)
pharmacy_id INT(11)
drug_id INT(11)
Indexes
pharmacy
pharmacy_id INT(11)
name VARCHAR(50)
address1 VARCHAR(50)
address2 VARCHAR(50)
city VARCHAR(50)
state CHAR(2)
zip_code VARCHAR(20)
phone VARCHAR(50)
Indexes