Database Project 23/24

Hospital Management System

3. Problem Domain Definition

Database Topic Proposal: Hospital Management System

The goal of our project is to design and implement a Hospital Management System database. It will contain relevant entities to a hospital system such as 'Patients', 'Doctors', 'Nurses', 'Departments', 'Appointments', 'Rooms', 'Records', 'Inventory', 'Medication', 'Diagnoses' and 'Suppliers'.

Expected attributes are 'patientName', 'doctorSpeciality', 'doctorRank', 'patientDateOfBirth', 'doctorOffice', 'inventoryQuantity', 'inventoryType', 'medicationSideEffects' and such.

Relationships will link patients to doctors, nurses to departments, appointments to patients and doctors, rooms to departments, medication to inventory and patient, suppliers to medication, diagnosis to records, records to patients.

4. First Submission: Conceptual Modeling

A. Domain Definition

The management of a hospital group pretends to find a way to store all information regarding their hospitals. Each hospital has set specifications, such as the name of the hospital itself; its address; ways to contact, like phone number, email and website; the administration and the date of foundation. Each hospital has a number of departments which are part of it. When the manager wants to know how the department is doing, they check for the name, the description and head of the department, number of beds occupied in that department by the patients, the location of that department inside the hospital, patient wait times and quality metrics. Departments also have rooms, with IDs and types. Nurses and Doctors are part of the departments, being able to be part of multiple departments as a doctor can multiple specialities. Nurses and Doctors are identified by their ID numbers. It's important to know their names, hour rates, their speciality/type and gender.

Patients are a vital part of a hospital and so it's essential to store their data. The management will need their ID, name, age, gender and ways to contact them, such as email and phone number.

When a patient wants to make an appointment, they can either have it with a doctor or a nurse, and it's important to store the information related to it as well, such as the room where it will be held, the date, reason and status of the appointment. If the appointment results in a diagnosis, it's also crucial to keep the information about the disease, the treatment options and the date of the diagnosis which may coincide with the appointment date.

The hospital also owns a pharmacy, where the patients can get their medication. The pharmacy has defined operating hours and also information about the patient's medication, like the primary and secondary effects, name and date of the prescription.

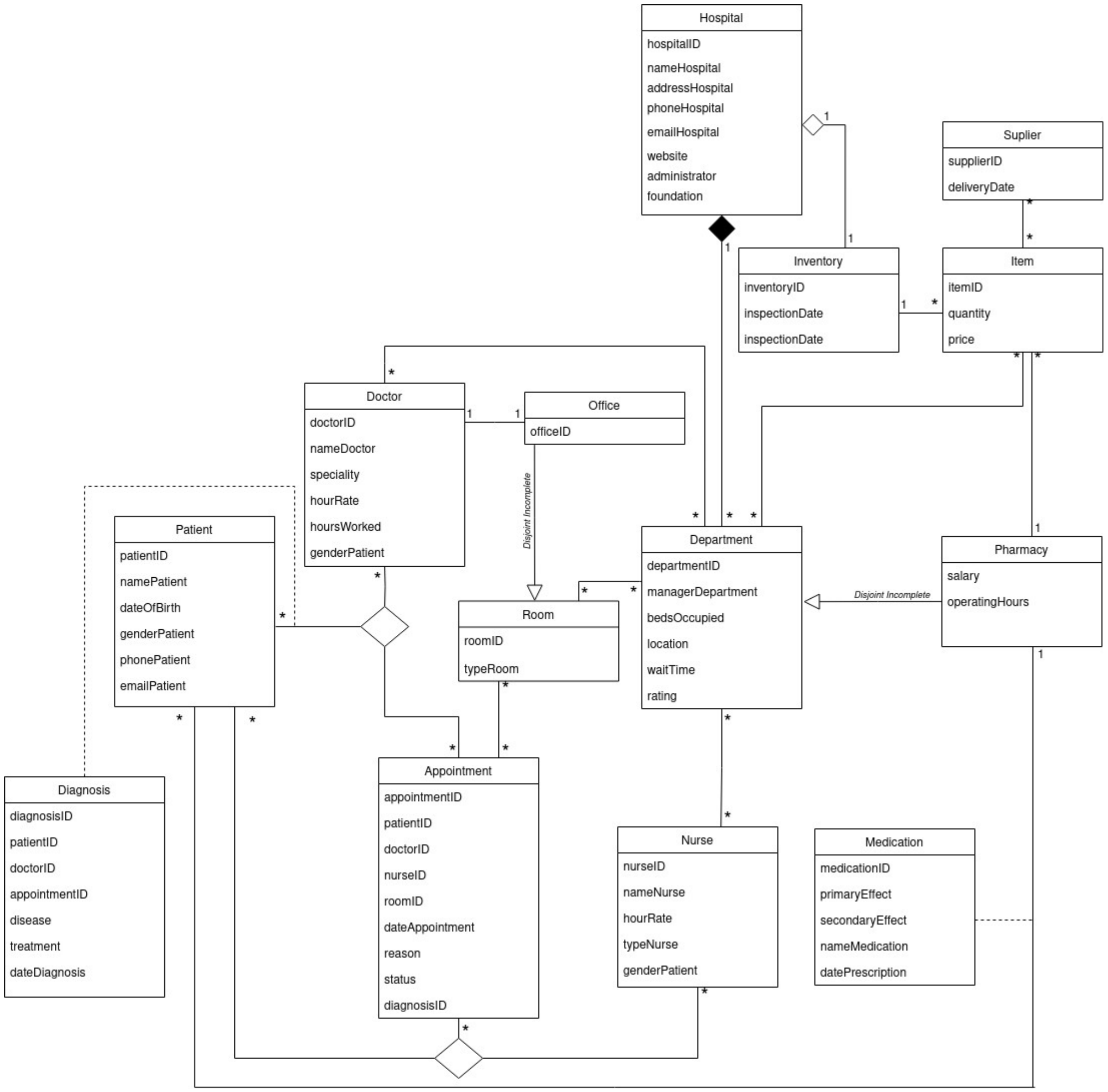
The pharmacy, like every other department, needs materials for their activities, so it's essential that the hospital keeps record of those items, such as their ID, quantity and price. For this, the hospital has an inventory that stores the data for all the items that the hospital keeps.

To get those items into the hospital, it needs various suppliers. Because of this, it's important that the hospital stores the information related to them, like the supplier's ID and date of delivery.

B. Conceptual Modeling

2.

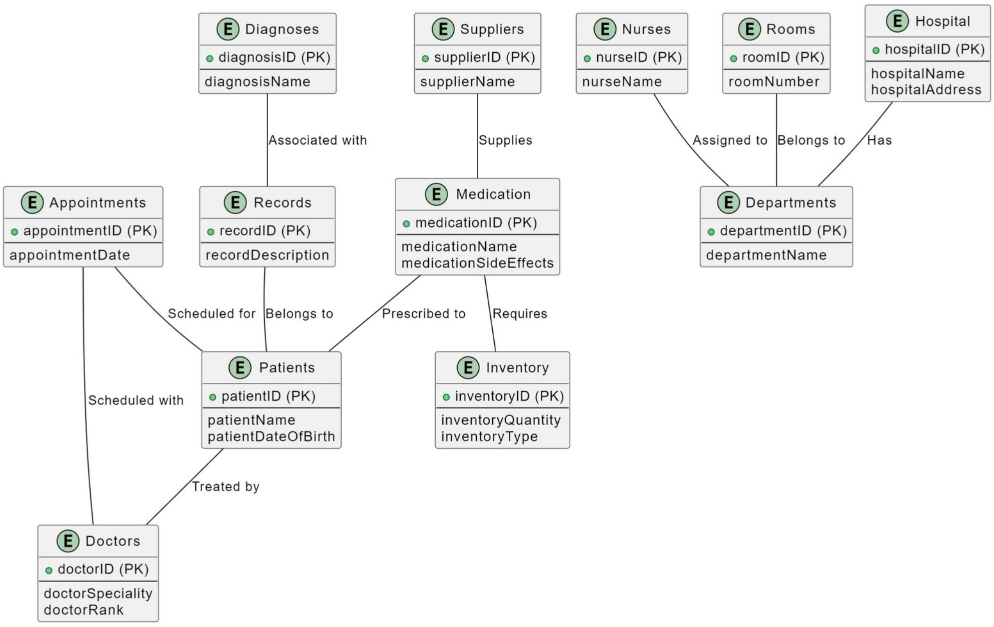
Initial Solution for the Conceptual Model



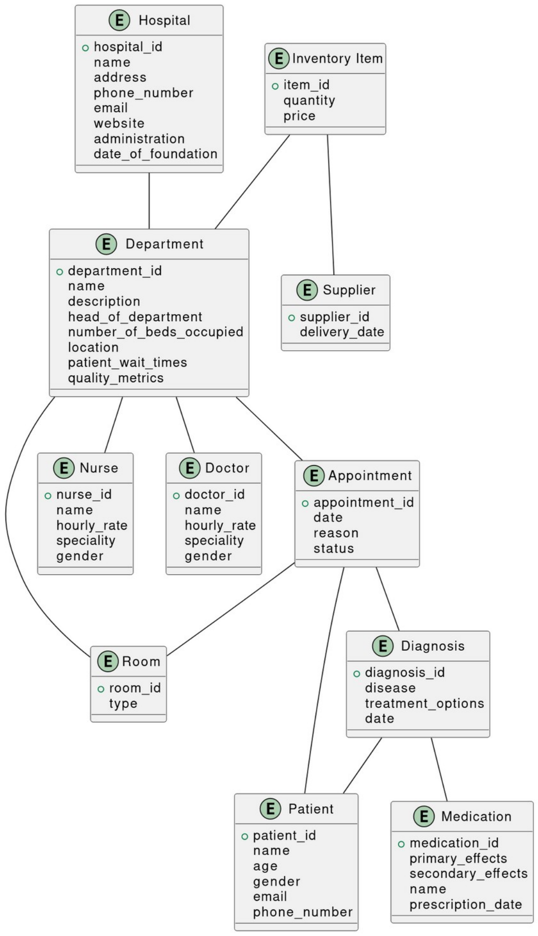
3.

Possible Solutions Shown to us by the AI

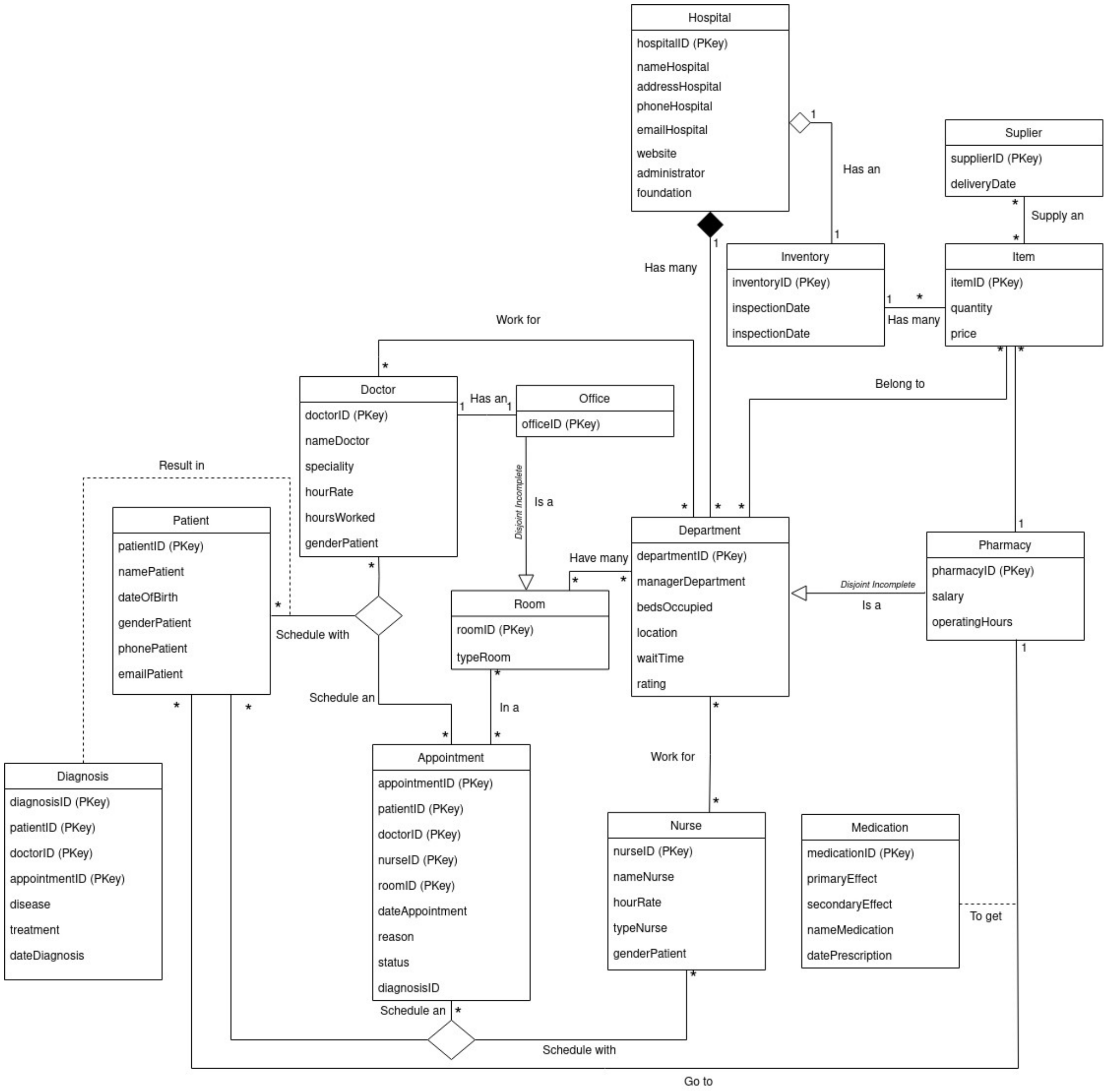
First model:



Second model:



Final Solution for the Conceptual Model with the help of AI



4.

The AI helped us attribute primary keys for each entity and add names for the relations.

C. Generative AI Integration

2.

The generative AI tool that we used was the ChatUML (https://chatuml.com/) that uses the GPT-3.5 model. The prompts for the first model were:

1- "can you do a uml based on this: Database Topic Proposal: Hospital Management System The goal of our project is to design and implement a Hospital Management System database. It will contain relevant entities to a hospital system such as 'Patients', 'Doctors', 'Nurses', 'Departments', 'Appointments', 'Rooms', 'Records', 'Inventory', 'Medication', 'Diagnoses' and 'Suppliers'. Expected attributes are 'patientName', 'doctorSpeciality', 'doctorRank', 'patientDateOfBirth', 'doctorOffice', 'inventoryQuantity', 'inventoryType', 'medicationSideEffects' and such. Relationships will link patients to doctors, nurses to departments, appointments to patients and doctors, rooms to departments, medication to inventory and patient, suppliers to medication, diagnosis to records, records to patients."

3- "add attributes to every class"

For the second model we just used the domain definition as the input.

3.

In general, the output was good, since we only used 3 prompts, two for the first model and only one for the second model. The limitations were the fact that there were no generalizations, composition or aggregations, all relations between entities were very simple.

Group Work

Everyone worked together, sharing ideas and adding content to the project, with no problems.