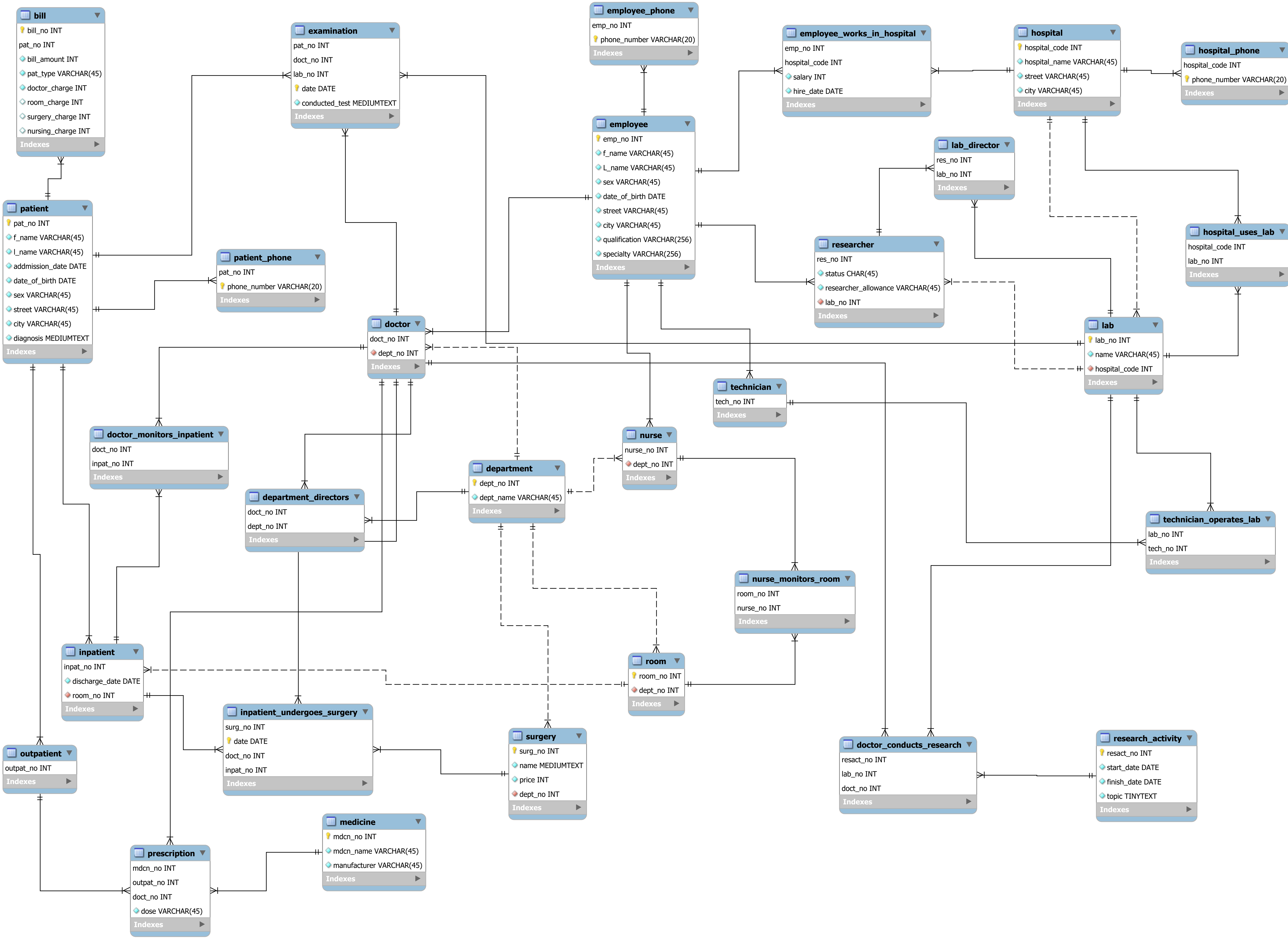
bill
bill_no INT
pat_no INT
bill_amount INT
pat_type VARCHAR(45)
doctor_charge INT
room_charge INT
surgery_charge INT
nursing_charge INT
Indexes
examination
pat_no INT
doct_no INT
lab_no INT
date DATE
conducted_test MEDIUMTEXT
Indexes
employee_phone
emp_no INT
phone_number VARCHAR(20)
Indexes
employee_works_in_hospital
emp_no INT
hospital_code INT
salary INT
hire_date DATE
Indexes
hospital
hospital_code INT
hospital_name VARCHAR(45)
street VARCHAR(45)
city VARCHAR(45)
Indexes
hospital_phone
hospital_code INT
phone_number VARCHAR(20)
Indexes
employee
emp_no INT
f_name VARCHAR(45)
L_name VARCHAR(45)
sex VARCHAR(45)
date_of_birth DATE
street VARCHAR(45)
city VARCHAR(45)
qualification VARCHAR(256)
specialty VARCHAR(256)
Indexes
lab_director
res_no INT
lab_no INT
Indexes
patient
pat_no INT
f_name VARCHAR(45)
l_name VARCHAR(45)
addmission_date DATE
date_of_birth DATE
sex VARCHAR(45)
street VARCHAR(45)
city VARCHAR(45)
diagnosis MEDIUMTEXT
Indexes
patient_phone
pat_no INT
phone_number VARCHAR(20)
Indexes
researcher
res_no INT
status CHAR(45)
researcher_allowance VARCHAR(45)
lab_no INT
Indexes
hospital_uses_lab
hospital_code INT
lab_no INT
Indexes
doctor
doct_no INT
dept_no INT
Indexes
lab
lab_no INT
name VARCHAR(45)
hospital_code INT
Indexes
technician
tech_no INT
Indexes
doctor_monitors_inpatient
doct_no INT
inpat_no INT
Indexes
nurse
nurse_no INT
dept_no INT
Indexes
department
dept_no INT
dept_name VARCHAR(45)
Indexes
department_directors
doct_no INT
dept_no INT
Indexes
technician_operates_lab
lab_no INT
tech_no INT
Indexes
nurse_monitors_room
room_no INT
nurse_no INT
Indexes
inpatient
inpat_no INT
discharge_date DATE
room_no INT
Indexes
room
room_no INT
dept_no INT
Indexes
inpatient_undergoes_surgery
surg_no INT
date DATE
doct_no INT
inpat_no INT
Indexes
surgery
surg_no INT
name MEDIUMTEXT
price INT
dept_no INT
Indexes
doctor_conducts_research
resact_no INT
lab_no INT
doct_no INT
Indexes
research_activity
resact_no INT
start_date DATE
finish_date DATE
topic TINYTEXT
Indexes
outpatient
outpat_no INT
Indexes
medicine
mdcn_no INT
mdcn_name VARCHAR(45)
manufacturer VARCHAR(45)
Indexes
prescription
mdcn_no INT
outpat_no INT
doct_no INT
dose VARCHAR(45)
Indexes

The abstract EERD model was translated to the relational schema and by applying normalization concepts a richer model was generated.

EERD to Relational Schema

The translation of EERD to a relational schema is a process which should follow the following guidelines:

• Turn each entity to a relation(table) with the same set of single-valued attributes. All the entities that the preliminary EER diagram contained are represented as relations (e.g. hospital, lab, patient, employee and etc.). The composite attributes are decomposed and stored as simple attributes (e.g. Name as f_name and l_name, Address as city and street)
• The HSP EER diagram illustrates different relationships between entities. M:M
• The mapping of generalization-inheritance can be handled using various solutions. One of them is to create one relation for the superclass and one relation for each subclass. The root entity(superclass) of the hierarchy is identified. The key of the root entity serves as an identifier of all the specialized entities(subclasses). Thus, the subclasses – doctor, nurse, researcher and technician contain the primary key of superclass – employee. In this case, emp_no acts as both foreign key and primary key. For better visibility, the emp_no PK was renamed in subclasses (e.g. in doctor table, emp_no was renamed to doct_no, in nurse table- nurse_no and etc) The cardinality between superclass and subclasses is 1:1. The same strategy was used for mapping patient – outpatient/inpatient inheritance, where the pat_no identifies both subclasses. This solution is a true reflection of superclass-subclass relationship formulated in EERD.
• The binary relationships illustrated in HSP EER diagram have various cardinalities, such as 1:1, 1: M and M:M. For mapping these relationships the following rules were considered:
    •For binary 1:1 relationships,such as "doctor manages department", the new tbale was created to store the departments with respective directors. The same was applied to full time researcher, who is a lab director and to the respective lab. This 1:1 relationships could be solved by referencing directors number in department table, however this would cause the cycling reference problem, therefore the above explained method is rational approach is this case.
    • For each binary 1:M relationship the primary key on the one's side of the relationship is added to the many's side as a foreign key. The examples of this relationship are doctors and nurses working in a department, hospital owning the labs, department having many rooms and etc. All the examples are mapped using the above explained rule.
    •For each binary M:M relationship the junction table, also referred to as a Bridge table or Associative Table, is created. The junction table references the primary keys of the both tables. "doctor_monitors_inpatient", "Employee_Works_in_Hospital", "nurse_monitors_room" and more are the examples of junction tables. The "Employee_Works_in_Hospital" junction table also includes respective attributes such as hire_date, specialization and etc.

• The problem of representing multivalued attribute, such as "phone" can be easily solved by modeling its own table. The relationship between employee and phone is 1:M therefore foreign key was established. In order to, uniquely identify each row of the table, the composite key for "Employee_Phone" is established, containing emp_no – FK, and phone-number. The phone number for hospital and patients are stored in the same manner.

• Besides the binary relationship, HSP EERD includes ternary relationships. For each n-ary relationship, where n>2 the new relation was created that references the participating entities and stores the respective attributes. Examples are “examination”, “prescription”,“doctor_conducts_research”, and “inpatient_undergoes_surgery”. Moreover, it was decided to include “date” attribute as part of the composite key in the “prescription”, “inpatient_undergoes_surgery”, and “examination”.

- The doctor can examine the same patient more than one time.
- There is a possibility of the inpatient getting the same surgery more than once.
- The doctor can prescribe the same medicine more than once to an outpatient.

Therefore, “date” attribute allows differentiation between examinations, prescriptions and undergone surgeries of the patient.

• Based on the above analysis of the HSP structure, the relationships were defined as either identifying or non-identifying. The relationship between superclass and subclasses is identifying notated with solid line. Child tables like doctor, nurse, researcher, technician, in/outpatients cannot be identified without respective parent tables. The identifying relationship is also between regular and junction tables. The nonidentifying relationship is between doctor and department as department can be recognized without doctor. The same logic is behind other non-identifying relationships.

Normalization

In order to avoid data redundancy and related anomalies, the database was normalized based on normal form requirements. In HSP schema the attributes of each relation are single-valued and all of them have unique names. Therefore, the database satisfies the first normal form. The violation of 1st NF was the existence of multivalued attribute – phone. The problem was solved by splitting the data and creating separate table for phones.

In junction tables, the attributes depend on the whole key, thus there is no partial dependency. Therefore, the HSP schema satisfies the 2nd NF, as well.

The database does not have a transitive dependency, meaning, no non-key attribute depends on other non-key attributes. Thus,3rd normal form is accomplished, as well. Moreover, the BCNF is verified by having no non-trivial dependencies.