

Project 1: Electronic Medical Record System

Vasu Tiwari

801254277

Description: -

An electronic version of a patient's medical history, maintained over time by the healthcare provider, is known as an electronic medical record system (EMR). An EMR may contain all the essential administrative clinical data relevant to that person's care under a specific provider, including demographics, progress notes, issues, medications, vital signs, past medical history, immunizations, laboratory information, and radiology reports. The clinician's workflow could potentially be streamlined by the EMR, which automates information access. Through a variety of interfaces, the EMR could also directly or indirectly assist additional care-related tasks like evidence-based decision support, quality control, and outcomes reporting.

System Requirements: -

An electronic record management system for any hospital/clinic or any health provider. The system should contain all the required details of a patient like his name, address, phone number and the details of his diagnosis like medicines prescribed etc. The system should also maintain information for billing and the details of departments involved in the process.

Based on the above requirements, created the below structure of the database.

Patient – Table to store patient details like his name, address, email address, phone number etc.

Clinical Care – Table to store patient clinical care i.e., diagnosis information.

Facility- Table to store details about a particular facility like critical care, radiology etc.

Insurance- Table to store details about patients' insurance details.

Consultant – Table to store details about consultant i.e., Doctor.

Billing - Table to store details about billing for a particular diagnosis.

Prescription - Table to store details about the prescription suggested by the consultant.

Medicine- Table to store details about medicine, like its name manufacture etc.

3 new tables added for supporting the n*n relations

Facility_consultant – Table for n*n dependency between Facility and Consultant Table.

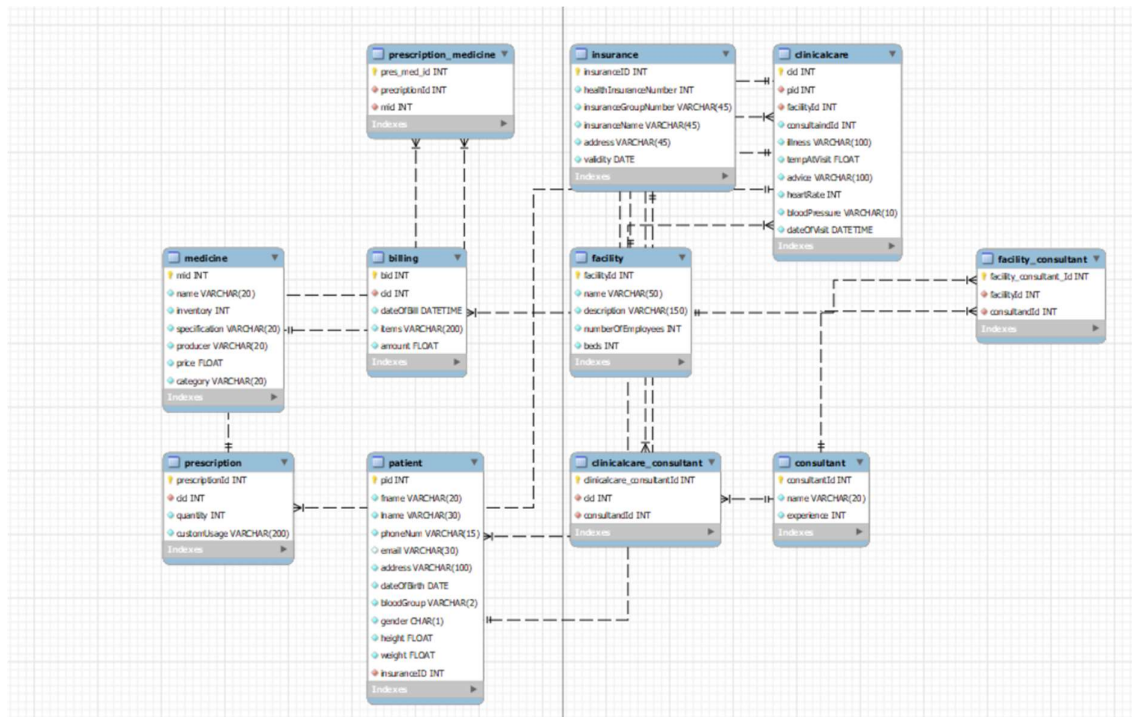
Prescription_medicine - Table for n*n dependency between Prescription and Medicine Table.

Clinicalcare_consultant - Table for n*n dependency between ClinicalCare and Consultant Table.

Storyboard: - New Patient Visit

- A new patient checks in at the hospital/clinic and his basic details are then filled in via a form. Basic details include - name, phone number, email, address, date of birth, blood group, gender, height, weight, details of illness, insurance details.
- Based on the details provided, if he is a new patient, a unique patient id is given to him at the patient table. A new request is created in the clinical care table and based on his illness a consultant and facility are assigned to him. Also, his temperature and blood pressure are taken.
- He is diagnosed and given some prescription which contains details of the medicines and quantity to be taken
- After he is diagnosed a billing Id is generated in the billing table which contains all the details of his bills, he needs to pay.

UML Diagram: -



Glossary: -

Below are the details of the tables in the system.

Patient(Strong): - Table to store the details of patient

1. pid: PK of the table
2. fname: first name of the patient
3. lname: last name of the patient
4. phoneNum: phone number of the patient
5. email: email of the patient
6. address: address of the patient
7. dateOfBirth: date of birth of the patient
8. bloodGroup: blood group of the patient
9. gender: gender of the patient
10. height: height of the patient
11. weight: weight of the patient
12. insuranceID: FK from the table Insurance

Clinical Care (Strong): - Table to store the details of clinical care provided to the patient

1. cid: PK of the table
2. pid: FK from the table Patient
3. facilityId: FK from the table Facility
4. consultandId: FK from the table consultant
5. dateOfVisit: date and time of the visit
6. illness : details of his illness for his visit.
7. tempAtVisit: temperature of the patient at the visit
8. advice: comments given to the patient
9. heartrate: heartrate of the patient at the visit
10. bloodPresurre: blood pressure of the patient at the visit

Facility (Strong): Table to store the details of given facility like critical care, eye care etc

1. facilityId: PK of the table
2. name: name of the facility
3. description: description of the facility
4. numberOFEmployees: number of employees in the facility
5. beds: beds available in the facility

Insurance (Strong): Table to store insurance details of patient

1. insuranceId: PK of the table
2. healthInsuranceNumber: health insurance number of the patient
3. validity: validity of the insurance

Billing (Strong): Table to store billing details of patient

1. bid: PK of the table
2. cid: FK from the table Clinical Care
3. dateOfBill: billing date

4. item: item details
5. amount: total amount of the bill

Prescription (Strong): Table to store prescription details

1. prescriptionId: PK of the table
2. cid: FK from the table Clinical care
3. mid: FK from the table Medicine
4. quantity: quantity of the medicines
5. customUsage: details of the prescription

Medicine (Strong): Table to store medicine details

1. mid: PK of the table
2. name: name of the medicine
3. inventory: total stock of the medicine
4. specification: details about the medicine
5. producer: name of the producer of the medicine
6. price: price of the medicine
7. category: category of the medicine

Consultant (Strong): Table to store Dr/Consultant details

1. consultantId: PK of the table
2. name: name of the consultant
3. experience: years of experience of consultant
4. facilityID: FK from the table Facility

Relationship between different tables.

1. Patient -> Clinical Care
 - 1 to n
 - 1 patient can have multiple clinical requests.
2. Patient -> Insurance
 - 1 to n
 - 1 patient can have multiple insurances.
3. Clinical Care -> Facility
 - 1 to n
 - A patient can be admitted to 1 or more facilities.
4. Clinical Care -> Consultant
 - n to n
 - A patient can be consulted with multiple consultants.

5. Clinical Care -> Billing
 - 1 to 1
 - A patient can have 1 bill per visit.
6. Clinical Care -> Prescription
 - 1 to n
 - A patient can be given 1 or multiple prescriptions.
7. Facility -> Consultant
 - n to n
 - A consultant can be part of multiple Facilities.
8. Prescription -> Medicine
 - n to n
 - A prescription can have multiple medicines, and a medicine can be part of multiple prescriptions.

Functional Dependencies and BCNF Normalization –

All the tables do not contain data in composite or multi-valued form hence the tables are in **1-NF**.

All relational tables do not contain any partial dependencies hence the tables are in **2-NF**.

There is no transitive dependency in any of the tables hence the tables are in **3-NF**.

A relation is in BCNF iff in every non-trivial functional dependency $X \rightarrow Y$, X is a super key.

Consider the below relations –

Consultant (consultantId, name, experience)

consultantId \rightarrow name, experience

Billing (bid, cid, dateOfBill, items, amount)

bid \rightarrow cid, dateOfBill, items, amount

facility (facilityId, name, description, numberOfEmployees, beds)

facilityId \rightarrow facilityId, name, description, numberOfEmployees, beds

medicine (mid, name, inventory, specification, producer, price, category)

mid \rightarrow name, inventory, specification, producer, price, category

Insurance (insuranceId, healthInsuranceNumber, validity)

insuranceId -> healthInsuranceNumber, validity

Indexes created

1. billing Table
 - a. cid_idx
2. clinicalcare Table
 - a. pid_idx
 - b. facility_idx
 - c. clinicalcare_pid_idx
3. clinicalcare_consultant table
 - a. cid_idx
 - b. consultantId_idx
4. consultant table
 - a. consultant_name_idx
5. facility table
 - a. facility_name_idx
6. facility_consultant table
 - a. facilityId_idx
 - b. cid_fac_idx
7. insurance table
 - a. insurance_name_idx
8. medicine table
 - a. medicine_name_idx
9. patient table
 - a. insuranceID_idx
 - b. patient_fname_idx
 - c. patient_lname_idx
10. prescription table
 - a. cid_clic_idx
11. prescription_medicine table
 - a. prescriptionId_idx
 - b. mid_idx

Stored Procedure Created

1. add_billing() : For adding billing information.
2. add_clinicalcare: For adding clinical care information.
3. add_consultant: For adding consultant details
4. add_facility: For adding facility details.
5. add_insurance: For adding insurance details
6. add_medicine: For adding medicine details
7. add_patient: For adding patient details.
8. add_prescription: For adding prescription details.
9. get_billing() : For getting billing information.
10. get_clinicalcare: For getting clinical care information.
11. get_consultant: For getting consultant details
12. get_facility: For getting facility details.
13. get_insurance: For getting insurance details
14. get_medicine: For getting medicine details
15. get_patient: For getting patient details.

16. get_prescription: For getting prescription details.
17. get_insurance_last_id: For getting last patient insurance id.
18. Update_clinicalcare: update illness details for given patient id and dateofVisit.

Java APIs created.

1. add_new_patient : For adding new patient
2. add_new_insurance: For adding new insurance.
3. update_clinicalcare: For updating illness of given patient id and date of visit.

Note: - I have include 1 document to run the Java program.

Lessons learned and other conclusions:

Creating database schema with the constraints was a little challenging. Creating java API was tedious but fun.

Thoroughly enjoyed the course and learned a lot.