Project1 Design & Implement a Relational Database

- **Patient Management System**

Highlight nouns and verbs

1.Introduction

The Patient Management System is a critical healthcare application designed to assist patients in managing patient appointments, medical prescriptions, and the delivery of medications. This system is particularly valuable for streamlining the process of test scheduling, medicine tracking, and ensuring patients receive timely updates and treatments. By integrating patient data, medical test reports, and doctor prescriptions, the system enables efficient tracking of patient health progress and ensures proper medication adherence.

The system is designed to handle various use cases, such as appointment for tests for patients, sending prescribed medications to the patient's home address, and recording the medication cycle. The goal is to improve the overall healthcare management experience for both the patients and medical professionals involved.

2. Rules of the business

1.The Patient Management APP stores each user's most fundamental personal information. This data forms the base layer for managing patient records.

2. Tests are conducted by Labs, and each test is associated with a labId. The Lab Reports are sent back to the system where doctors can review them to prescribe or adjust treatment. Lab Reports are generated based on the test results and contain important information such as illnessName and memo for the doctor’s review.

3. Doctors refer Prescriptions to patients after reviewing the Lab reports. Each Prescription contains information such as the date and dosage of Instructions. The system records multiple medications per prescription, each with a unique medicine, medicine name, and dosage frequency.
4. Patients can schedule appointments with doctors through the Patient Management APP. Each appointment records the date and time of the visit, ensuring that the doctor is assigned accordingly. Multiple appointments can be scheduled for the same doctor, and the system prevents double

booking by maintaining unique time slots per doctor.

5. Each medicine issued to a patient is tracked through a Medicine Record. This record logs the date and frequency of medication issuance, ensuring that doctors and the system have an accurate history of the patient's medication timeline. A single medicine can be recorded in multiple medicine issuance records, representing different times it was administered.

6. Doctors have access to all the lab reports for patients under their care. These reports include detailed test results and assist the doctor in making data-driven decisions about adjusting or prescribing treatments. The system ensures that only the assigned doctor has access to each patient's lab reports for privacy and security purposes.

7. After reviewing lab reports and assessing the patient's condition, doctors can adjust prescriptions by changing the dosage instructions or adding/removing medications. This ensures that prescriptions are always up-to-date based on the patient's current health status and the latest lab results.

3. nouns and verbs

Nouns:

Patient Management System

healthcare application

patients

appointments

medical prescriptions

delivery of medications

system

test scheduling

medicine tracking

patient data

medical test reports

doctor prescriptions

progress

medication adherence

appointment

prescribed medications

medication cycle

healthcare management

medical professionals

Patient Management APP

personal information

records

primary functions

Health Surveillance

Medicine Monitor

monitoring

medications

monitor Id (Patient Id)

dosages

Health Surveillance record

symptoms

disease history

Tests

Labs

test

labId

Lab Reports

doctors

Reports

illnessName

memo

doctor’s review

Doctors

Prescriptions

Lab reports

Prescription

information

date

dosage of Instructions

medications per prescription

medicine

medicine name

dosage frequency

Delivery

prescribed medications

address

patient

delivery Id

delivery date

Delivery

Medicine

Verbs

Appointment

Assist

Recording

Prescribed

Conducted

Review

Delivery

4. Assumption

Patient and Survey

Relationship Type: One-to-Many because a Patient can participate in multiple Surveys that record their health status and symptoms over time.

Lab and Test

Relationship Type: Aggregation because a Lab organizes and conducts multiple Tests, but each Test is an independent entity.

Lab and LabReport

Relationship Type: Aggregation because a Lab generates multiple LabReports containing test results and analyses, which exist independently within the system.

Patient and Test

Relationship Type: One-to-Many because a Patient can undergo multiple Tests, each providing important health assessments.

Patient and Doctor

Relationship Type: Many-to-Many because a Patient can have appointments with multiple Doctors, and a doctor can serve multiple Patients.

Doctor and Prescription

Relationship Type: One-to-Many because a doctor can issue multiple Prescriptions for different Patients.

Patient and Medicine

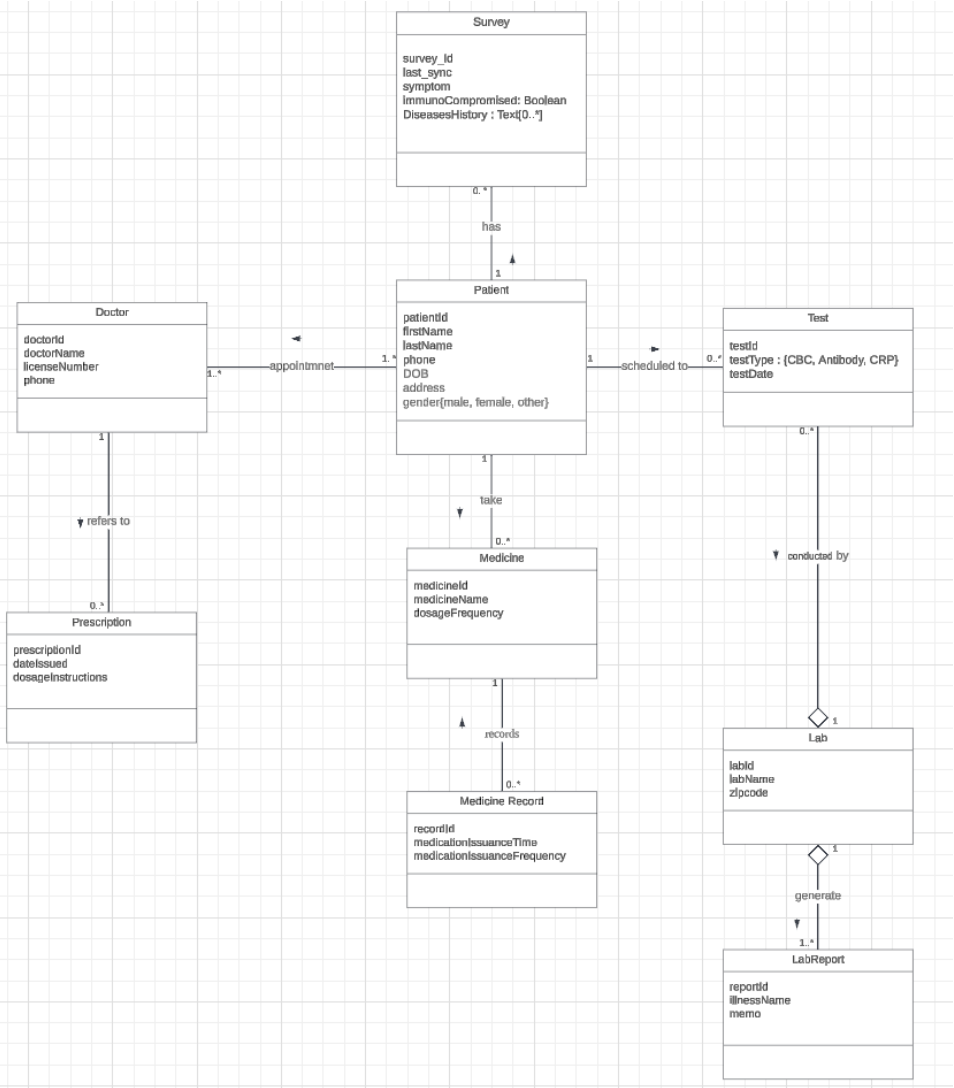
Relationship Type: One-to-Many because a Patient can be prescribed multiple Medicines, each with specific dosage instructions.

Medicine and MedicineRecord

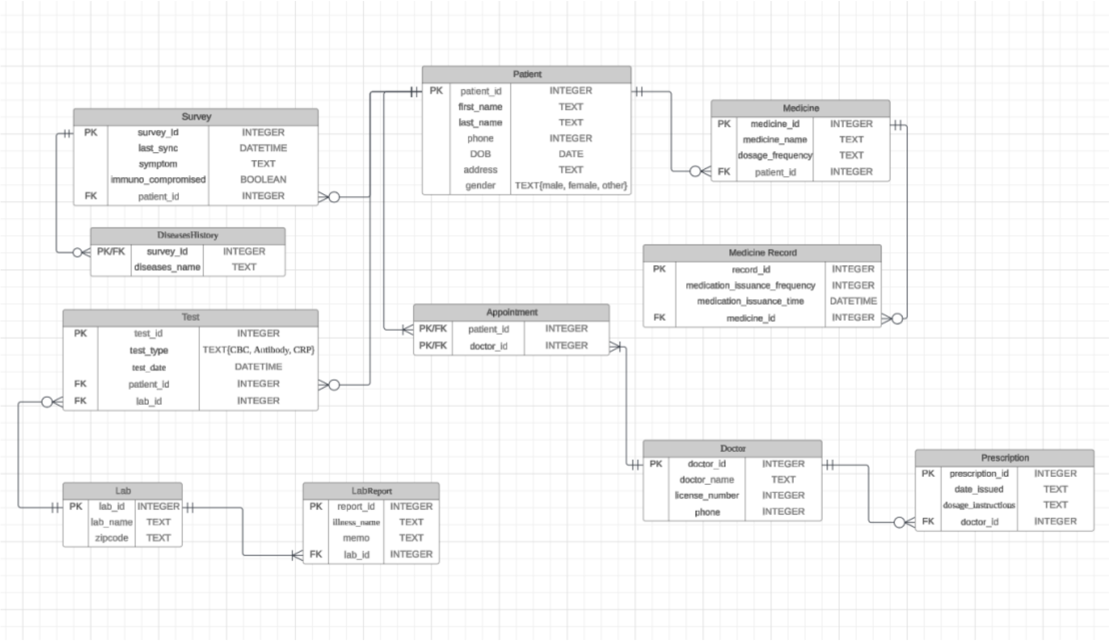
Relationship Type: One-to-Many because each MedicineRecord logs an instance of a specific Medicine being administered. One Medicine can have multiple records showing its issuance to different Patients at different times.

5.Conceptual model

LucidChart Tool



6. logical data model



7. Relational Schema Definitions

The logical schema shown above is resolved into the below relational schema. A relation R with attributes A1,A2,A3..An is shown as R(A1,A2,*A3*..An), where the primary key is underlined, and foreign keys are shown in *italics*.

Patient (patient_id, first_name, last_name, phone, DOB, address, gender)

Survey (survey_id, last_sync, symptom, immuno_compromised, *patient_id*)

DiseasesHistory(*survey_id*, diseases_name)

Test(test_id, test_type, test_date, *patient_id* , *lab_id*)

Lab(lab_id, lab_name, zipcode)

LabReport(report_id, illness_name, memo, *lab_id*)

Doctor(doctor_id, doctor_name, license_number, phone)

Appointment(*patient_id, doctor_id)*

Medicine(medicine_id, medicine_name, dosage_frequency, *patient_id)*

Medicine Record(record_id, medication_issuance_frequency, medication_issuance_time, *medicine_id*,)

Prescription(prescription_id, date_issued, dosage_instructions, *doctor_id*)

8. Proof using functional dependencies to show that schema is in BCNF

1.X → Y is a trivial functional dependency (i.e Y is a subset of X)

2.X is a super key for the schema

Patient

patient_id → {first_name, last_name, phone, DOB, address, gender}

Survey

survey_id→{ last_sync, symptom, immuno_compromised, patient_id }

DiseasesHistory

survey_id→{diseases_name}

Test

test_id→{test_type, test_date, patient_id, lab_id}

Lab

lab_id→{lab_name, zipcode}

LabReport

report_id→{illness_name, memo, lab_id }

Doctor

doctor_id→ {doctor_name, license_number, phone}

Medicine Record

record_id→{ medication_issuance_frequency, medication_issuance_time , medicine_id}

Medicine

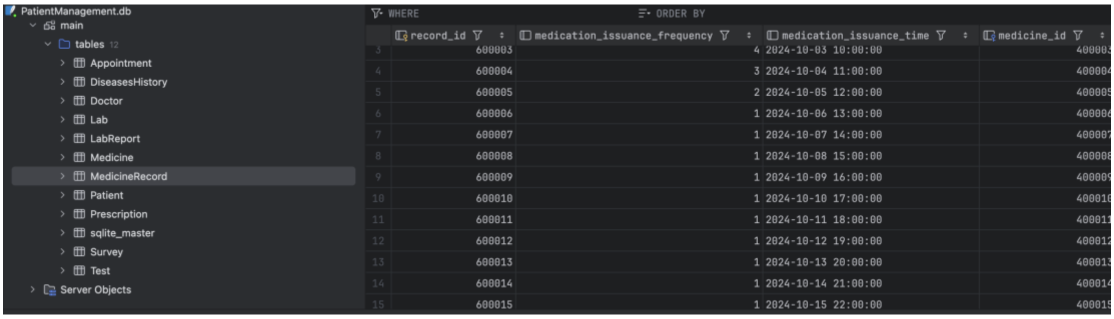
medicine_id→ {medicine_name, dosage_frequency, patient_id }

Prescription

prescription_id→{date_issued, dosage_instructions, doctor_id}

Appointment

patient_id, doctor_id form a composite primary key(trivial FD hence in BCNF)



	record_id	medication_issuance_frequency	medication_issuance_time	medicine_id
3	600003	4	2024-10-03 10:00:00	400003
4	600004	3	2024-10-04 11:00:00	400004
5	600005	2	2024-10-05 12:00:00	400005
6	600006	1	2024-10-06 13:00:00	400006
7	600007	1	2024-10-07 14:00:00	400007
8	600008	1	2024-10-08 15:00:00	400008
9	600009	1	2024-10-09 16:00:00	400009
10	600010	1	2024-10-10 17:00:00	400010
11	600011	1	2024-10-11 18:00:00	400011
12	600012	1	2024-10-12 19:00:00	400012
13	600013	1	2024-10-13 20:00:00	400013
14	600014	1	2024-10-14 21:00:00	400014
15	600015	1	2024-10-15 22:00:00	400015

9.Schema creation and Table definition in MySQL

```
CREATE TABLE Survey (
    survey_id INTEGER PRIMARY KEY,
    last_sync DATETIME,
    symptom TEXT,
    immuno_compromised BOOLEAN,
    patient_id INTEGER,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);

CREATE TABLE DiseasesHistory (
    survey_id INTEGER PRIMARY KEY,
    diseases_name TEXT,
    FOREIGN KEY (survey_id) REFERENCES Survey(survey_id)
);

CREATE TABLE Patient (
    patient_id INTEGER PRIMARY KEY,
    first_name TEXT,
    last_name TEXT,
    phone INTEGER,
    DOB DATE,
    address TEXT,
    gender TEXT CHECK(gender IN ('male', 'female', 'other'))
);

CREATE TABLE Test (
    test_id INTEGER PRIMARY KEY,
    test_type TEXT CHECK(test_type IN ('CBC', 'Antibody', 'CRP')),
    test_date DATETIME,
    patient_id INTEGER,
    lab_id INTEGER,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (lab_id) REFERENCES Lab(lab_id)
);

CREATE TABLE Lab (
    lab_id INTEGER PRIMARY KEY,
    lab_name TEXT,
    zipcode TEXT
);

CREATE TABLE Appointment (
    patient_id INTEGER NOT NULL,
    doctor_id INTEGER NOT NULL,
    PRIMARY KEY (patient_id, doctor_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);

CREATE TABLE LabReport (
    report_id INTEGER PRIMARY KEY,
    illness_name TEXT,
    memo TEXT,
    lab_id INTEGER,
    FOREIGN KEY (lab_id) REFERENCES Lab(lab_id)
);

CREATE TABLE Doctor (
    doctor_id INTEGER PRIMARY KEY,
    doctor_name TEXT,
    license_number INTEGER,
    phone INTEGER
);

CREATE TABLE Prescription (
    prescription_id INTEGER PRIMARY KEY,
    date_issued DATETIME,
    dosage_instructions TEXT,
    doctor_id INTEGER,
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);

CREATE TABLE Medicine (
    medicine_id INTEGER PRIMARY KEY,
    medicine_name TEXT,
    dosage_frequency TEXT,
    patient_id INTEGER,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);

CREATE TABLE MedicineRecord (
    record_id INTEGER PRIMARY KEY,
    medication_issuance_frequency INTEGER,
    medication_issuance_time DATETIME,
    medicine_id INTEGER,
    FOREIGN KEY (medicine_id) REFERENCES Medicine(medicine_id)
);
```