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

1. 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.
2. Patient and Appointment
Relationship Type: One-to-Many because a Patient can schedule multiple Appointments for consultations or treatments.
3. Appointment and Doctor
Relationship Type: Many-to-One because multiple Appointments can be scheduled with the same Doctor, who manages these patient consultations.
4. Medicine and Patient
Relationship Type: Many-to-Many because a Patient can take multiple Medicines, and each Medicine can be prescribed to or taken by multiple patients. This relationship is tracked by MedicineRecord, which logs instances of medicine issuance to patients.
5. Aggregation between Lab and Test

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

6. Aggregation between Lab and LabReport

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

7. Patient and Test

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

8. Prescription and Medicine

Relationship Type: One-to-Many because a single Prescription can include multiple Medicines, allowing doctors to prescribe various treatments to patients.

9. Prescription and Doctor

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

10. Doctor and LabReport

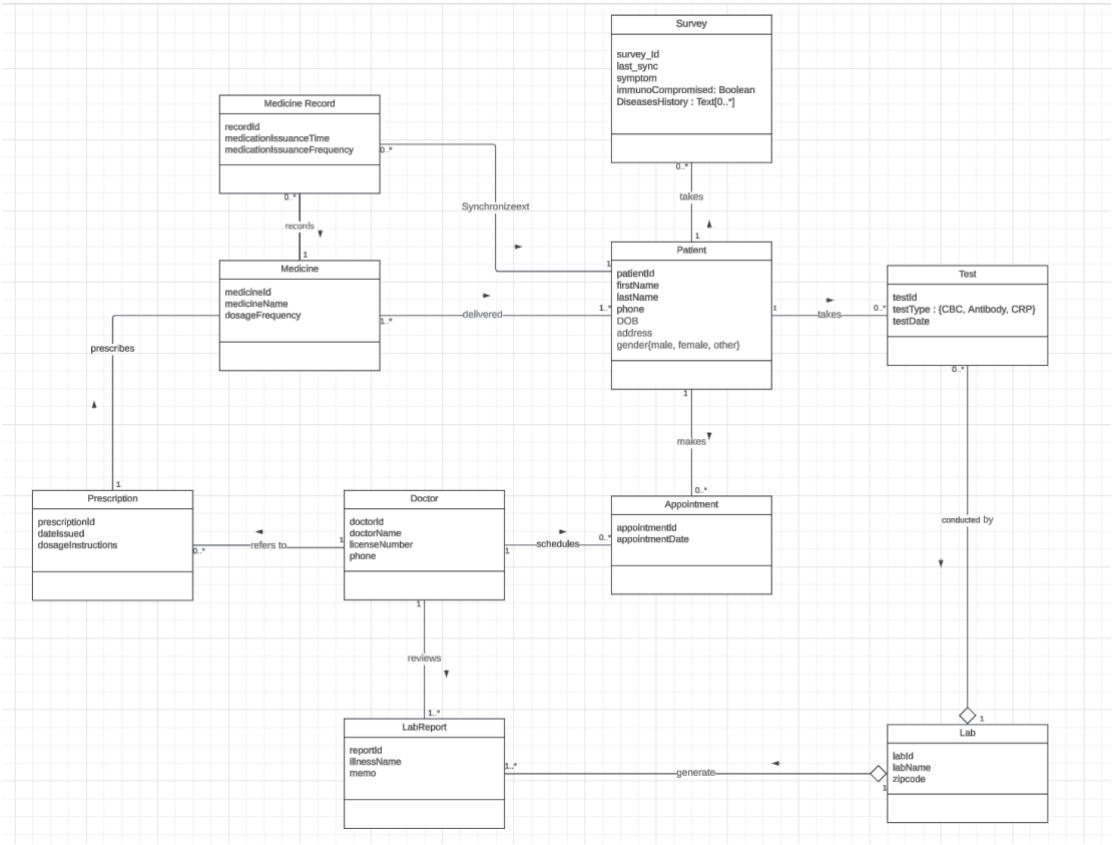
Relationship Type: One-to-Many because a doctor can review and evaluate multiple LabReports to make informed diagnoses and treatment decisions.

11. Medicine and MedicineRecord

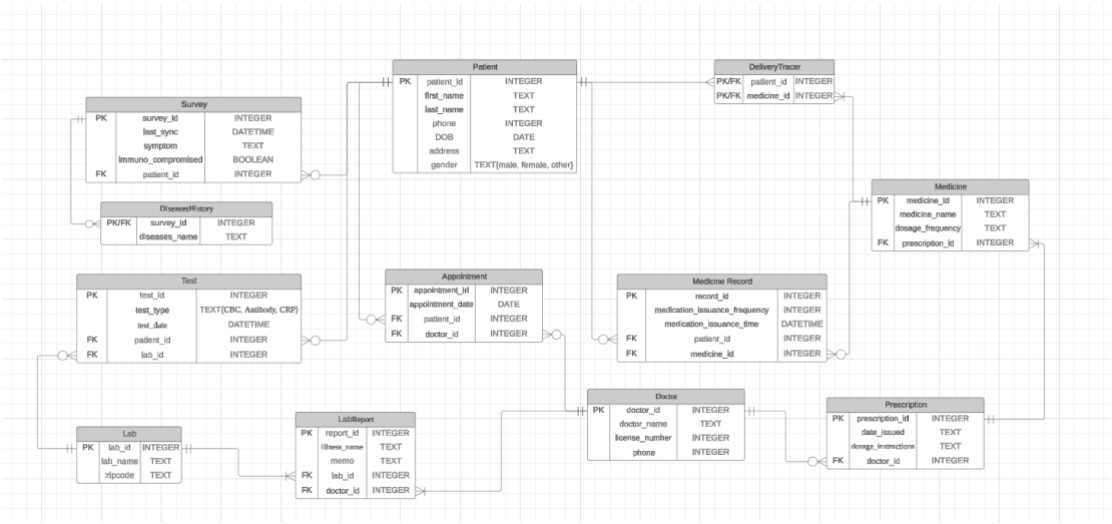
Relationship Type: One-to-Many because each MedicineRecord logs an instance of a specific Medicine being administered to a patient. One Medicine can have multiple records showing its issuance to different patients at different times.

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_id*)

Doctor(doctor_id, doctor_name, license_number, phone)

Appointment(appointment_Id, appointment_date, *patient_id, doctor_id)*

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

Medicine Record(record id, medication_issuance_frequency, medication_issuance_time, *patient_id, medicine_id*,)

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

DeliveryTracer(*patient_id*, *medicine_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_id}

Doctor

doctor_id→ {doctor_name, license_number, phone}

Medicine Record

record_id→{ medication_issuance_frequency, medication_issuance_time , patient_id, medicine_id}

Medicine

medicine_id→ {medicine_name, dosage_frequency, prescription_id }

Prescription

prescription_id→{date_issued, dosage_instructions, doctor_id}

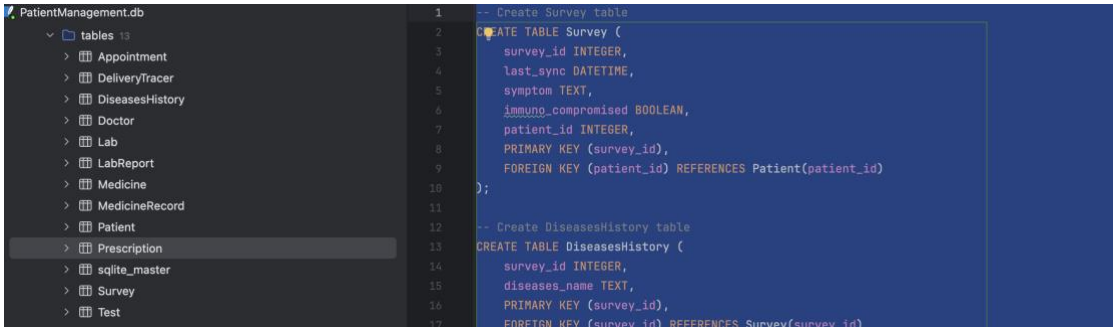
Appointment

appointment_Id→ {appointment_date, patient_id, doctor_id }

DeliveryTracer

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

9.Schema creation and Table definition in MySQL




```
CREATE TABLE Survey (
    survey_id INTEGER,
    last_sync DATETIME,
    symptom TEXT,
    immuno_compromised BOOLEAN,
    patient_id INTEGER,
    PRIMARY KEY (survey_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
CREATE TABLE DiseasesHistory (
    survey_id INTEGER,
    diseases_name TEXT,
    PRIMARY KEY (survey_id),
    FOREIGN KEY (survey_id) REFERENCES Survey(survey_id)
);
CREATE TABLE Patient (
    patient_id INTEGER,
    first_name TEXT,
    last_name TEXT,
    phone INTEGER,
    DOB DATE,
    address TEXT,
    gender TEXT CHECK(gender IN ('male', 'female', 'other')),
    PRIMARY KEY (patient_id)
);

CREATE TABLE Test (
    test_id INTEGER,
```

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

CREATE TABLE Appointment (
    appointment_id INTEGER,
    appointment_date DATETIME,
    patient_id INTEGER,
    doctor_id INTEGER,
    PRIMARY KEY(appointment_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);

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



CREATE TABLE Doctor (
```

```
    doctor_id INTEGER,
    doctor_name TEXT,
    license_number INTEGER,
    phone INTEGER,
    PRIMARY KEY(doctor_id)
);

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

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

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

CREATE TABLE DeliveryTracer (
    patient_id INTEGER,
```

```
    medicine_id INTEGER,
    PRIMARY KEY (patient_id, medicine_id),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (medicine_id) REFERENCES Medicine(medicine_id)
);
```