Phase 4 By Ethan Seem Yaseen Hassan Alan Nguyen

A. Problem Description: Replicate the provided medical database project description.

There is a PERSON table. They have a personID, date of birth, gender, and their name (first, middle, last). The personID must be in the format P-XXX where X is a number from 0-9.
There are also 3 tables that are the PERSON_EMAIL_ADDRESS, PERSON_PHONE_NUMBER, and PERSON_LOCATION. They tie email addresses, phone numbers, and state/street/city/zip together via personID.

There is a HEALTHCARE_PROFESSIONAL table. Any PERSON can be a Healthcare professional as long as they are over 18 years old, have non-negative years of experience and working hours. They have an employeeID, specialization, department, and a hire date as well. The employeeID must be in the form EMP-XXX where X is a number from 0-9. Each HEALTHCARE_PROFESSIONAL is tied to a unique personID as well.

There are tables for 4 types of specialization: DOCTOR, NURSE, TECHNICIAN, and ADMINISTRATIVE STAFF. They each have special IDs that have the prefixes DO ,NU ,TEC , and AD, followed by 3 numbers from 0-9. They are also tied to an employee ID, however ADMINISTRATIVE STAFF must be tied to a Doctor ID instead.

There is a PATIENT table. It has PatientID, EmployeeID, PersonID, and Enrollment_date. The PatientID must be in the format PT-XXXXX where X is a number from 0-9. Each patient is tied to their respective PersonID, as well as an employeeID if they have one.

There is a PHARMACY table that contains PharmacyID, Name, Location, and ContactInformation. The PharmacyID must be in the format PH-XXX where X is a number from 0-9.

There is a BILL table that contains BillingID, BillDate, ItemizedCosts, TotalCost, PaymentMethod, and PatientID. The BillingID must be in the format B-XXX where X is a number from 0-9. It also must be tied to a patient via their patient ID.

There is an APPOINTMENT table that contains AppointmentID, AppointmentDate, Purpose, AppointmentTime, PatientID, and EmployeeID. The AppointmentID must be in the format A-XXX where X is a number from 0-9. It also must be tied to a patient (who is visiting) and a healthcare professional (who is conducting the appointment) via their

respective IDs.

There is a MEDICAL_RECORD table that contains RecordID, PatientID, Healthcare_ProfessionalID, Diagnosis, Treatment_History, Allergies, and LabResults. The RecordID must be in the format R-XXXXX where X is a number from 0-9. It must also be tied to a patient and healthcare professional via their respective IDs.

There is a TREATMENT table that contains TreatmentID, StartDate, EndDate, Description, Outcome, PatientID, PharmacyID, and Healthcare_ProfessionalID. The last 3 have to be tied to their respective tables' IDs. The TreatmentID must be in the format T-XXX where X is a number from 0-9.

Finally, there is an INSURANCE table that has an InsuranceID, CompanyName, and BillAmount.

B. Project Questions: Address and answer the three project questions listed above.

How crucial is the ability to model superclass/subclass relationships in a medical database environment, particularly for entities like Healthcare Professionals and Patients?

> Modeling superclass/subclass relationships improves data organization, integrity, and access control in medical database environments. By categorizing entities like Healthcare Professionals and Patients hierarchically, data management can be streamlined, and data consistency rules can be enforced. Additionally, evolving healthcare roles and responsibilities can be accommodated by enabling targeted queries, optimizing database performance, and facilitating scalability. Thus, superclass/subclass modeling ensures effective medical data management through a secure and flexible framework.

Identify 5 additional business rules that the MedConnect database should accommodate. How would these rules impact your Extended Entity-Relationship (EER) model?

> Patient-Healthcare Professional Association
>
> - A patient can be associated with multiple healthcare professionals for various treatments and consultations, but each healthcare professional should be assigned to at least one patient. This rule impacts the EER model by introducing a many-to-many relationship between the Patient and the Healthcare Professional entities, requiring an association table to manage these relationships.

Allergy Constraint Enforcement

- The database should enforce allergy constraints, allowing only values from a predefined list of common allergens. This rule affects the Medical Records entity by adding a constraint on the Allergies attribute, ensuring that only valid allergens are recorded.

Appointment Overlap Prevention

- Appointments for a healthcare professional should not overlap. The database must check for time conflicts when scheduling appointments to ensure that a healthcare professional is available during the specified time slot. This rule influences the Appointments entity by introducing time constraints and validation checks during appointment scheduling.

Minimum Age for Treatment

- Patients must be at least 18 years old to receive certain treatments. This rule impacts the Treatment entity by adding an age verification check before allowing treatment records to be created for a patient.

Payment Plan Options

- The billing system should support multiple payment methods and installment plans for patients. This rule affects the Billing entity by adding attributes to capture payment method details (Cash, Credit, Insurance) and installment plan information, allowing for flexible patient billing options.

These five additional business rules impact the EER model by introducing new relationships, constraints, attributes, and validation checks to ensure data accuracy and integrity while adhering to business policies and regulations. Through these additional business rules, the model becomes more comprehensive and realistic to fit real-world scenarios, making the database more functional.

Argue the case for using a Relational DBMS such as Oracle for the MedConnect project. (Detailed explanation with design justifications to be provided in the final report).

Utilizing Oracle as the Relational Database Management System for the MedConnect project ensures data integrity and consistency, scalability, and security. Oracle ensures data integrity via constraints, triggers, and foreign key relationships, which are essential for managing accurate healthcare data and vital for patient safety and regulatory adherence. Scalability features such as partitioning, clustering, and parallel processing ensure efficient management of

large healthcare data volumes as the database expands. Security measures like role-based access control, encryption, and auditing protect patient information to maintain data confidentiality and integrity. Furthermore, Oracle's SQL capabilities and advanced analytical tools facilitate comprehensive data analysis, reporting, and business intelligence. Finally, Oracle offers high availability features, disaster recovery solutions, and industry compliance, making it a safe choice for MedConnect to ensure continuous service delivery, minimize downtime, and mitigate data loss or system failure risk.

Design Justifications:

- Entity-Relationship Model: Oracle supports implementing complex entity-relationship models, accommodating entities like Patients, Healthcare Professionals, Medical Records, etc., along with their relationships, attributes, and constraints.
- Normalization: Oracle's support for normalization ensures efficient data storage, reduces redundancy and enhances data integrity within the database schema.
- Indexing and Query Optimization: Oracle's indexing capabilities and query optimization techniques ensure fast and efficient data retrieval, crucial for MedConnect's real-time healthcare services and reporting needs.
- Stored Procedures and Triggers: Oracle allows the implementation of stored procedures and triggers for automating tasks, enforcing business rules, and improving database performance.
- Backup and Recovery: Oracle's robust backup and recovery solutions, including RMAN (Recovery Manager), Data Guard, and Flashback technologies, provide MedConnect with reliable data protection and disaster recovery capabilities.

D.Relational Schema after Normalization: Present all tables in 3NF with primary and foreign keys indicated.

PERSON(PERSON_ID (Primary Key), DATE_OF_BIRTH, GENDER, FIRST_NAME, LAST_NAME, MIDDLE_NAME)

PERSON_EMAIL_ADDRESS(PERSON_ID(foreign/primary key from PERSON), EMAIL_ADDRESS (primary key))

PERSON_PHONE_NUMBER(PERSON_ID(foreign/primary key from PERSON), PHONE_NUMBER (primary key))

PERSON_LOCATION(PERSON_ID(foreign/primary key from PERSON), STATE, STREET, CITY, ZIP)

HEALTHCARE_PROFESSIONAL(EMPLOYEE_ID (primary key), PERSON_ID (foreign/primary key from PERSON), SPECIALIZATION, DEPARTMENT, HIRE_DATE, YEARS_OF_EXPERIENCE, WORKING_HOURS,ENROLLMENT_DATE)

DOCTOR(DOCTOR_ID (primary key), EMPLOYEE_ID (foreign key from HEALTHCARE_PROFESSIONAL))

NURSE(NURSE_ID (primary key), EMPLOYEE_ID (foreign key from HEALTHCARE_PROFESSIONAL))

TECHNICIAN(TECHNICIAN_ID (primary key), EMPLOYEE_ID (foreign key from HEALTHCARE_PROFESSIONAL))

TREATMENT(TREATMENT_ID (primary key), END_DATE, START_DATE, OUTCOME, DESCRIPTION, PRESCRIPTION_RECORDS, HEALTHCARE_PROFESSIONAL_ID(foreign key from HEALTHCARE_PROFESSIONAL), PHARMACY_ID(foreign key from PHARMACY), PATIENT_ID(foreign key from PATIENT))

BILL(BILLING_ID(primary key), DATE, ITEMIZED_COSTS, TOTAL_COST, PAYMENT_METHOD, PATIENT_ID(foreign key from PATIENT))

PATIENT(PATIENT_ID(primary key), PERSON_ID(foreign key from PERSON), EMPLOYEE_ID (foreign key from HEALTHCARE_PROFESSIONAL))

PHARMACY(PHARMACY_ID(primary key), NAME, LOCATION, PHONE_NUMBER)

MEDICAL_RECORD(RECORD_ID(primary key), DIAGNOSIS, TREATMENT_HISTORY, ALLERGIES, CURRENT_MEDICATIONS, LAB_RESULTS, HEALTHCARE_PROFESSIONAL_ID(foreign key from HEALTHCARE_PROFESSIONAL), PATIENT_ID(foreign key from PATIENT))

APPOINTMENT(APPOINTMENT_ID(primary key), TIME, PURPOSE, DATE, HEALTHCARE_PROFESSIONAL_ID(foreign key from HEALTHCARE_PROFESSIONAL), PATIENT_ID(foreign key from PATIENT))

ADMINISTRATIVE_STAFF(ADMIN_ID(primary key), DOCTOR_ID(foreign key from DOCTOR))

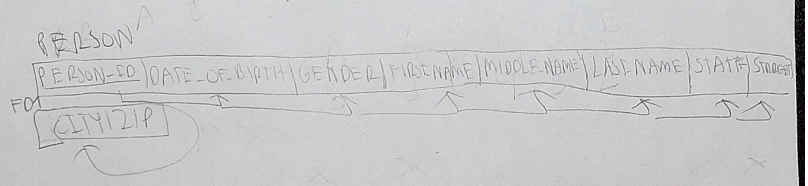
INSURANCE(INSURANCE_ID(primary key), COMPANY_NAME, BILL_AMOUNT)

In table form:

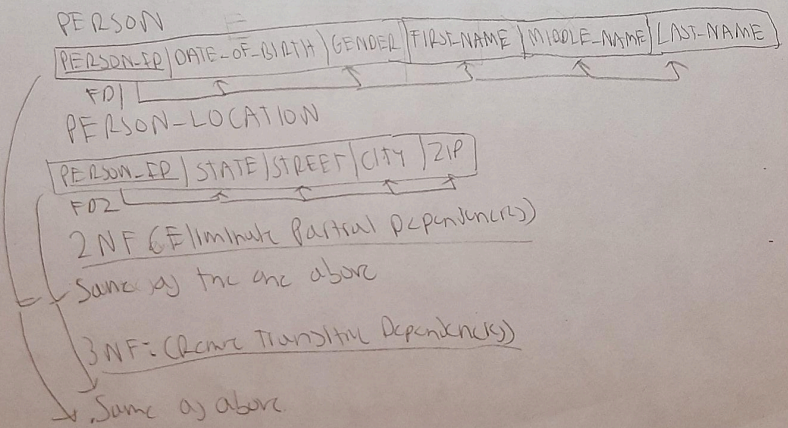
E. Dependency Diagram: Provide a dependency diagram for each table from Phase III-b.

PERSON (PERSON_ID, DATE_OF_BIRTH, GENDER, FIRST_NAME, MIDDLE_NAME, LAST_NAME, STATE, STREET, CITY, ZIP)

F = { PERSON_ID → DATE_OF_BIRTH, GENDER, FIRST_NAME, MIDDLE_NAME, LAST_NAME, STATE, STREET, CITY, ZIP }



1NF: (Eliminate any multivalued, composite attributes and repeating columns and create a seperate table for it)

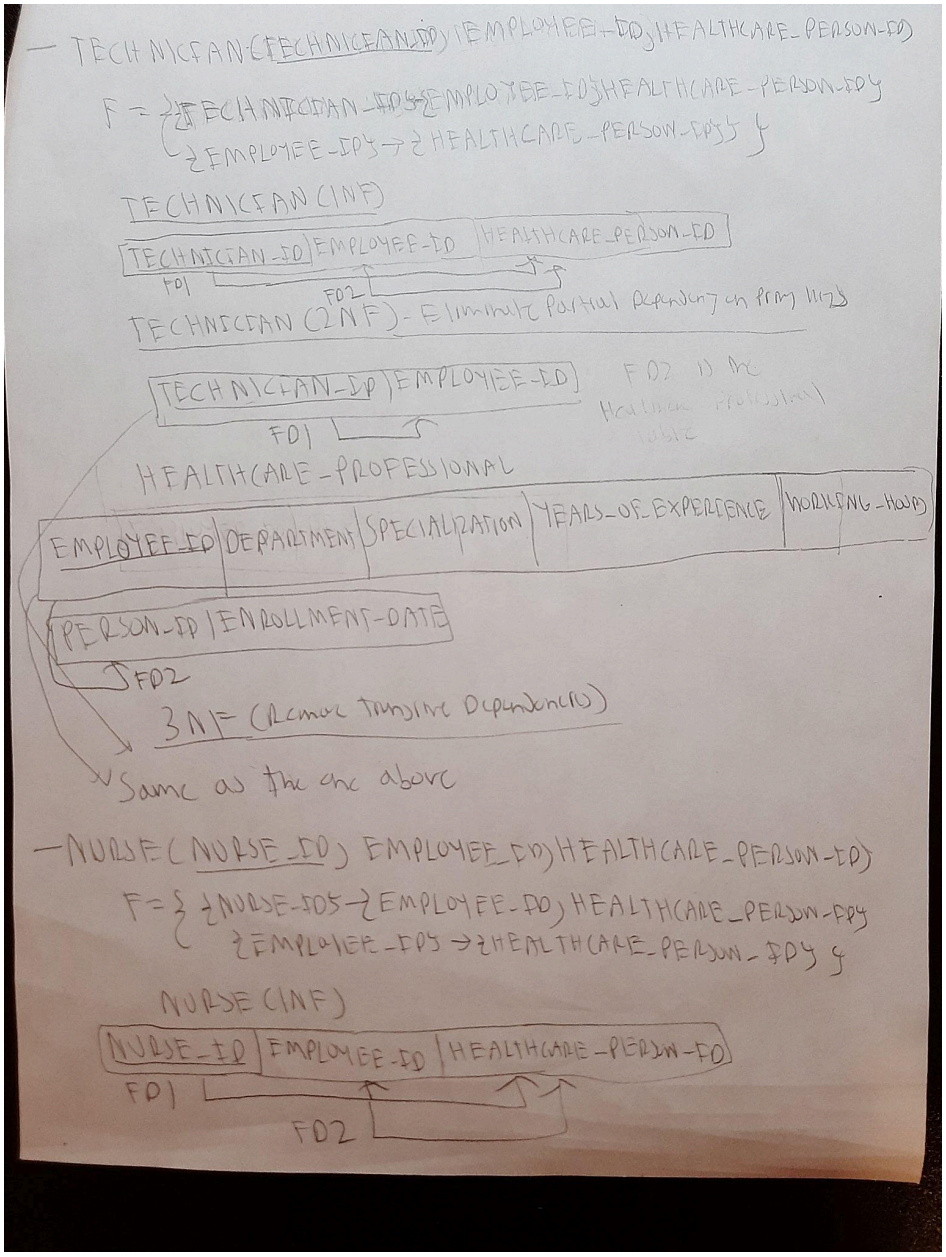


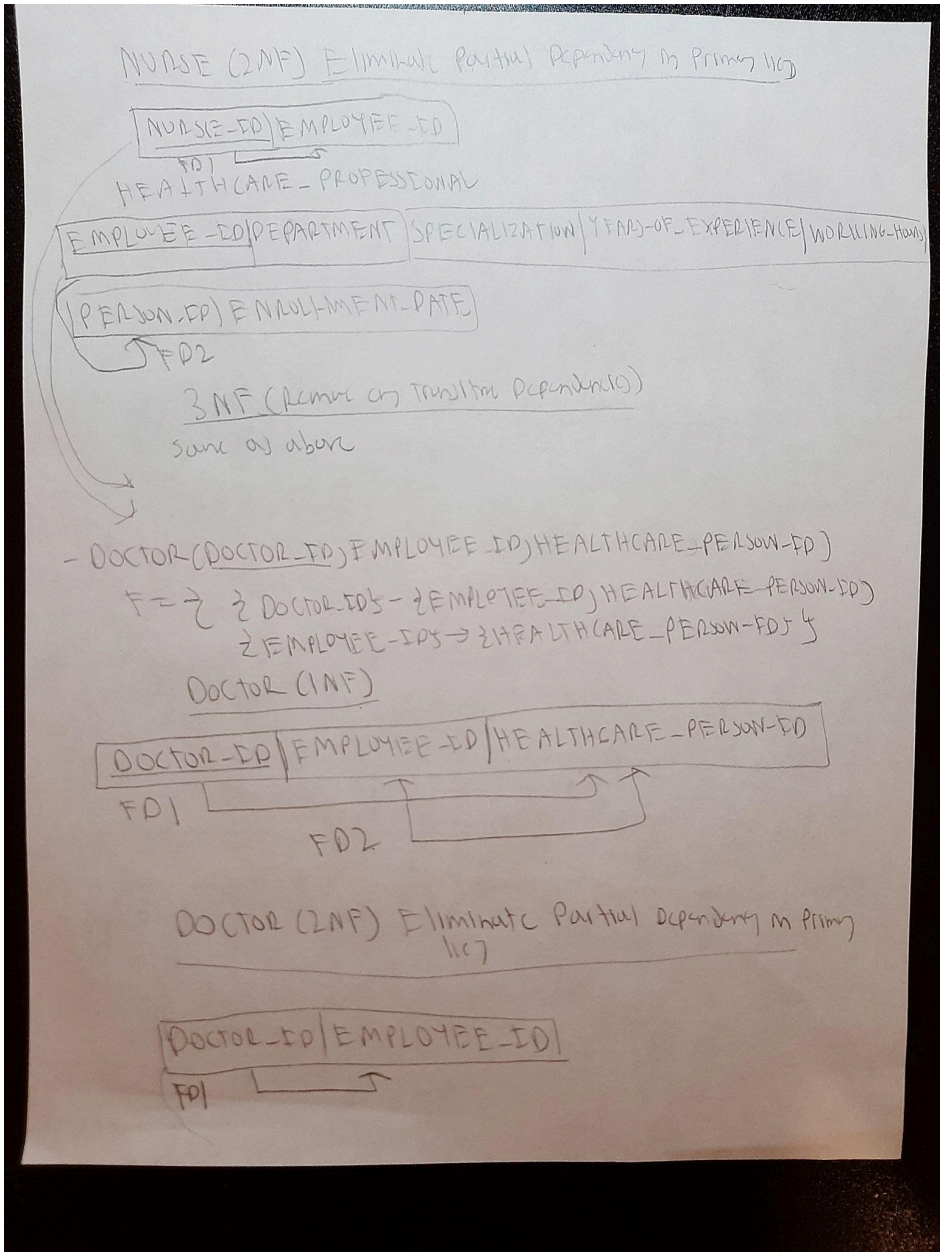
2NF (Eliminate Partial Dependencies)

Same as the one above

3NF: (Remove Transitive Dependencies)

Same as above


TECHNICIAN(TECHNICIAN_ID, EMPLOYEE_ID, HEALTHCARE_PERSON_ID)
F = { TECHNICIAN_ID → EMPLOYEE_ID, HEALTHCARE_PERSON_ID
EMPLOYEE_ID → HEALTHCARE_PERSON_ID }
TECHNICIAN (INF)
TECHNICIAN_ID | EMPLOYEE_ID | HEALTHCARE_PERSON_ID
FD1
FD2
TECHNICIAN (2NF) - Eliminate Partial dependency on Primary key
TECHNICIAN_ID | EMPLOYEE_ID
FD1
FD2 is the Healthcare professional table
HEALTHCARE_PROFESSIONAL
EMPLOYEE_ID | DEPARTMENT | SPECIALIZATION | YEARS_OF_EXPERIENCE | WORKING_HOURS
PERSON_ID | ENROLLMENT_DATE
FD2
3NF (Remove Transitive Dependencies)
Same as the one above
NURSE(NURSE_ID, EMPLOYEE_ID, HEALTHCARE_PERSON_ID)
F = { NURSE_ID → EMPLOYEE_ID, HEALTHCARE_PERSON_ID
EMPLOYEE_ID → HEALTHCARE_PERSON_ID }
NURSE (INF)
NURSE_ID | EMPLOYEE_ID | HEALTHCARE_PERSON_ID
FD1
FD2


NURSE (2NF) Eliminate Partial Dependency on Primary key
NURSE_ID | EMPLOYEE_ID
FD1
HEALTHCARE_PROFESSIONAL
EMPLOYEE_ID | DEPARTMENT | SPECIALIZATION | YEARS_OF_EXPERIENCE | WORKING_HOURS
PERSON_ID | ENROLLMENT_DATE
FD2
3NF (Remove any Transitive Dependencies)
same as above
- DOCTOR(DOCTOR_ID, EMPLOYEE_ID, HEALTHCARE_PERSON_ID)
F = { {DOCTOR_ID} → {EMPLOYEE_ID, HEALTHCARE_PERSON_ID}
{EMPLOYEE_ID} → {HEALTHCARE_PERSON_ID} }
DOCTOR (1NF)
DOCTOR_ID | EMPLOYEE_ID | HEALTHCARE_PERSON_ID
FD1
FD2
DOCTOR (2NF) Eliminate Partial Dependency on Primary key
DOCTOR_ID | EMPLOYEE_ID
FD1

HEALTHCARE_PROFESSIONAL

EMPLOYEE_ID	DEPARTMENT	SPECIALIZATION	YEARS_OF_EXPERIENCE	WORKING_HOURS

PERSON_ID	ENROLLMENT_DATE

FD2

3NF (Remove transitive dependency)

(same as above)

TREATMENT(TREAMENT_ID, END_DATE, START_DATE, OUTCOME, DESCRIPTION, PRESCRIPTION_RECORDS, HEALTHCARE_PROFESSIONAL_ID, PHARMACY_ID, PATIENT_ID, EMPLOYEE_PERSON_ID, PATIENT_PERSON_ID)

F = {TREATMENT_ID → {END_DATE, START_DATE, OUTCOME, DESCRIPTION, PRESCRIPTION_RECORD, HEALTHCARE_PROFESSIONAL_ID, PHARMACY_ID, PATIENT_ID}, {HEALTHCARE_PROFESSIONAL_ID → {EMPLOYEE_PERSON_ID}, {PATIENT_ID → {PATIENT_PERSON_ID}

TREATMENT

TREATMENT_ID	END_DATE	START_DATE	OUTCOME	DESCRIPTION	PRESCRIPTION_RECORD

HEALTHCARE_PROFESSIONAL_ID	PHARMACY_ID	PATIENT_ID	EMPLOYEE_PERSON_ID	PATIENT_PERSON_ID

FD2

FD3

1NF (Eliminate any multivalued/composite attributes and repeating columns and create a seperate table for it)

TREATMENT

TREATMENT_ID	END_DATE	START_DATE	OUTCOME	DESCRIPTION	PRESCRIPTION_RECORD

FD1

HEALTHCARE_PROFESSIONAL_ID	PHARMACY_ID	PATIENT_ID

2NF (Eliminate Partial Dependency)

Same as the one above

3NF (Remove Transitive Dependency)

Same as the one above

- APPOINTMENT (APPOINTMENT_ID, TIME, PURPOSE, DATE, HEALTHCARE_PROFESSIONAL_ID, PATIENT_ID, EMPLOYEE_PERSON_ID, PATIENT_PERSON_ID)

F = { {APPOINTMENT_ID} → {TIME, PURPOSE, DATE, HEALTHCARE_PROFESSIONAL_ID, PATIENT_ID}
{HEALTHCARE_PROFESSIONAL_ID} → {EMPLOYEE_PERSON_ID}
{PATIENT_ID} → {PATIENT_PERSON_ID}

APPOINTMENT

APPOINTMENT_ID	TIME	PURPOSE	DATE	HEALTHCARE_PROFESSIONAL_ID	PATIENT_ID

FD1

EMPLOYEE_PERSON_ID	PATIENT_PERSON_ID

FD2

FD3

1NF (Eliminate any multivalued/composite attributes and repeating columns and create a separate table for it)

APPOINTMENT

APPOINTMENT_ID	TIME	PURPOSE	DATE	HEALTHCARE_PROFESSIONAL_ID	PATIENT

FD)

2NF (Eliminate Partial Dependency)

Same as the one above

3NF (Eliminate Transitive Dependency)

Same as the one above

- MEDICAL RECORD (RECORD_ID, DIAGNOSIS, TREATMENT_HISTORY, ALLERGIE
CURRENT_MEDICATIONS, LAB_RESULT, HEALTHCARE_PROFESSIONAL_I
PATIENT_ID, EMPLOYEE_PERSON_ID, PATIENT_PERSON_ID)

F = { {RECORD_ID} → {DIAGNOSIS, TREATMENT_HISTORY, ALLERGIES, CURRENT_
MEDICATIONS, LAB_RESULTS, HEALTHCARE_PROFESSIONAL_ID,
PATIENT_ID}, {HEALTHCARE_PROFESSIONAL_ID} →
{EMPLOYEE_PERSON_ID}
{PATIENT_ID} → {PATIENT_PERSON_ID} }

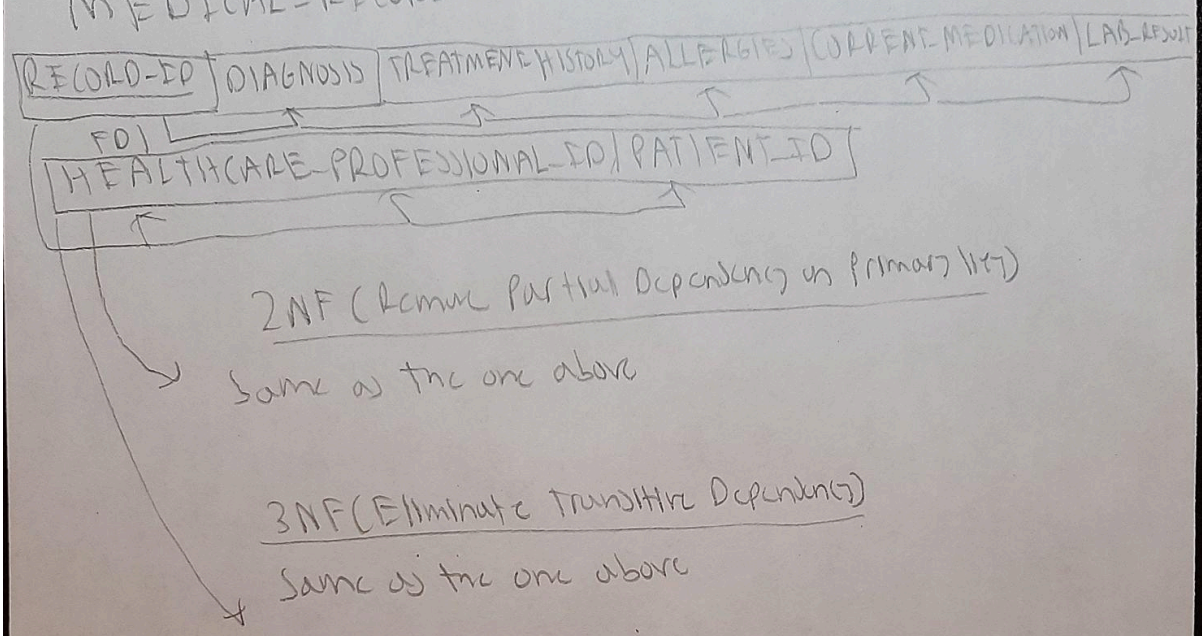
MEDICAL_RECORD

RECORD_ID	DIAGNOSIS	TREATMENT_HISTORY	ALLERGIES	CURRENT_MEDICATIONS	LAB_RESULT

HEALTHCARE_PROFESSIONAL_ID	PATIENT_ID	EMPLOYEE_PERSON_ID	PATIENT_PERSON_ID

1NF (Eliminate any multivalued composite attributes and repeating columns and create a separate table for it)

MEDICAL-RECORD



2NF (Remove Partial Dependency on Primary Key)

Same as the one above

3NF (Eliminate Transitive Dependency)

Same as the one above