2. Hospital Patient Records (Up to 3NF & BCNF)

Table Data:

PatientID | PatientName | DoctorID | DoctorName | Specialization | VisitDate
601 | Sam | D101 | Dr. Patel | Cardiology | 2024-03-10
602 | John | D102 | Dr. Lee | Orthopedic | 2024-03-11
601 | Sam | D101 | Dr. Patel | Cardiology | 2024-03-12

Tasks:

- Identify functional dependencies.

- Check for 1NF, 2NF, and 3NF violations. - Normalize the table to BCNF.

Step 1: Identify Functional Dependencies

PatientID → PatientName

- A patient ID uniquely determines a patient's name.

DoctorID → DoctorName, Specialization

- A doctor ID uniquely determines the doctor's name and specialization.

(PatientID, DoctorID, VisitDate) → (PatientName, DoctorName, Specialization)

- The combination of PatientID, DoctorID, and VisitDate uniquely identifies a record.

Step 2: Checking Normal Forms

Checking Given Table Is In First Normal Form (1NF)
The given table is already in 1NF because:

- Each attribute contains atomic values.
- There are no repeating groups.

Checking Given Table Is In Second Normal Form (2NF)
The table is NOT in 2NF because PatientName and DoctorName, Specialization depend only on part of the primary key.

Partial dependency exists:

- PatientID → PatientName (Partial dependency: PatientName depends only on PatientID)
- DoctorID → DoctorName, Specialization (Partial dependency: DoctorName and Specialization depend only on DoctorID)

Checking Given Table Is In Second Normal Form (2NF)
The table is NOT in 3NF because there are transitive dependencies.

To achieve **BCNF**, we need to decompose the table into separate tables.

Decomposed Tables

click this link for see -> https://dbdiagram.io/d/Practice-question-2-67e5570b4f7afba18472ebae