**Step 6.1 - Begin with the list of the tables that the entities and relationships from the E-R diagram mapped to naturally, from the sample project section at the end of chapter 4.**

For each table on the list, identify functional dependencies and normalize the relation to BCNF. Then decide whether the resulting tables should be implemented in that form. If not, explain why.

A table is in **First Normal Form (1NF)** if and only if the following conditions are satisfied:

1. Each attribute contains only one value (single-valued)
2. All attribute values are atomic meaning they cannot be broken down any further (there are no repeating groups).
3. There are no duplicated rows in the table

A table is in **Second Normal Form (2NF)** if and only if the following conditions are satisfied:

1. Table is in First Normal Form (1NF)
2. And all non-primary key attributes are fully functionally dependent on the primary key (no partial dependency)

Note: If primary key has only one attribute and the table is 1NF, then the table is automatically 2NF

A table is in **Third Normal Form (3NF)** if and only if the following conditions are satisfied:

1. Table is in Second Normal Form (2NF)
2. And all non-primary key attributes are transitive dependent on the primary key

A table is in **Boyce-Codd Normal Form (BCNF)** if and only if the following conditions are satisfied:

1. Table is in Third Normal Form (3NF)
2. And for any non-trivial functional dependency A→B, A is a superkey

policyType(company, policyType, medicalCoPay, labCoPay, pharmacyCoPay)

InsurancePolicy ( policyNo, *company*, *policyType,* insuredName, startDate, endingDate)

1. Values are single-valued; records are unique; attributes are atomic => The table is in 1NF.
2. 1NF; policyNo -> company, insuredName, policytype, medicalCoPay, labCoPay, pharmacyCoPay, startDate, endingDate Therefore, the table is in 2NF.
3. 2NF, There is no transitive dependencies => The table is in 3NF
4. The table is in BCNF.

Patient (patientNo, name, address, phone, dateOfBirth, sex, *insuranceCo, policyNo*, relationshipToInsured)

1. The values are single-valued and records are unique. However, attributes are not atomic => Therefore we will transform the ‘address’ attribute to a composite key consisting of street, zipCode, city, and state. We are also breaking up the ‘name’ into ‘firstName’ and ‘lastName’.
2. 1NF, patientNo → firstName, lastName, street, zipCode, city, state,1 phone, dateOfBirth, sex, insuranceCo, policyNo, relationshipToInsured

Table is in 2NF since all of non-prime attributes are fully functionally dependent on the primary key patientNo

1. 2NF, The insuranceCo attribute is transitively dependent on patientNo since patientNo → policyNo and policyNo → insuranceCo based on the insurance policy table. Therefore we will remove the insuranceCo attribute to avoid the redundancy.
2. The table is in BCNF.

The new tables are below.

Patient1 (patientNo, firstName, lastName, street, *zipCode*, phone, dateOfBirth, sex*, policyNo*, relationshipToInsured);

Zip (zipCode, city, state);

Staff (staffNo, name, title, specialty, address, phone)   
  
1. 1NF: Same as Patient, we will transform the ‘address’ attribute into Staff (staffNo, name, title, specialty, phone, address, city, state, zip)

2. 2NF: The table is in 2NF as it is in 1NF and all of it’s non-primary key attributes are fully functionally dependent on its primary key

3. 3NF: The table is in 3NF as it is in 2NF and there are no transitive dependencies.

4. The table is in BCNF

Staff (staffNo, firstName, lastName, title, specialty, street, zipCode, phone)

Availability (*staffNo*, availDate, startTime, endTime)

1. Values are single-valued; records are unique; attributes are atomic => The table is in 1NF.
2. 1NF, staffNo, availDate → startTime, endTime
3. 2NF, There are no transitive dependencies, therefore the table is in 3NF.
4. The table is not in BCNF since the ‘availDate’ under AvailabilityDate is not a superkey.

Room (roomNo, *roomType*, condition)

1. Values are single-valued; records are unique; attributes are atomic. Therefore the table is in 1NF.
2. 1NF, roomNo → roomType, condition

Table is in 2NF since all of non-prime attributes are fully functionally dependent on the primary key roomNo

1. 2NF, There are no transitive dependencies, therefore the table is in 3NF.
2. The table is in BCNF.

roomType(roomType, condition)

Visit (visitNo, *patientNo*, visitdate*,* visittime, duration, reason, visitType, visitCost, *staffNo, roomNo)*1. 1NF: Visit (visitNo, patientNo, visitdate, visittime, duration, reason, visitType, visitCost, staffNo, roomNo)

2. 2NF: The table is in 2NF as it’s already in 1NF and it’s non-primary key attributes are fully functionally dependent on the primary key.

Visit (visitNo, visitdate, visittime, duration, reason, visitType, visitCost)

staffNo(patientNo, roomNo)

3. 3NF: The table is in 3NF as it is in 2NF and there are no transitive dependencies.

4. The table is in BCNF

Visit (visitNo, *patientNo*, visitdate*,* visittime, duration, reason,visitType, *staffNo, roomNo)*

visitType (visitType, visitCost)

Appointment (*patientNo*, apptdate, appttime, reason, *staffNo*, *visitNo*)

1. 1NF: Appointment (patientNo, apptdate, appttime, reason, staffno, visitNo)
2. 2NF: The table is in 2NF as it’s already in 1NF and it’s non-primary key attributes are fully functionally dependent on the primary key.
3. 3NF: The table is in 3NF as it is in 2NF and there are no transitive dependencies.
4. The table is in BCNF

Appointment *(patientNo,* apptdate, appttime, reason*, staffNo, visitNo)*

**Filippo’s part** :

Referral (refNo, *visitNo*, refTo, reason)

1. Tuple cells hold single-valued inputs; records are unique; attributes are atomic. Hence, the table is in first normal form (1FN).
2. 1NF, refNo → refTo, reason

The Referral table is in Second Normal Form since all 1NF requirements are satisfied and the table’s non-prime attributes are dependent on the primary key, refNo.

1. 2NF, there are no transitive dependencies in Referral, so the table is in 3NF as well.
2. Since all the previous normalization requirements are satisfied and that Referral has only one unique superkey, refNo, the table is also in BCNF.

Bill (invoiceNo, billDate, totalAmount, dueDate, *patientNo,* amountPaid)

1. Tuple cells hold single-valued inputs; records are unique; attributes are atomic. Thus, the table is in first normal form (1FN).
2. 1NF, invoiceNo → billDate, totalAmount, dueDate, patientNo, amountPaid

The Bill table is in Second normal Form since all 1NF requirements are met and the table’s non-prime attributes are dependent on the primary key, invoiceNo.

1. I am going to delete the attribute amountPaid from the Bill table as it critical to the Payment table and not to the Bill table. Therefore, Bill is now 3NF having met all 2NF requirements.
2. Since all the previous normalization requirements are met and that Referral has only one unique superkey, refNo, the table is also in BCNF.

Bill (invoiceNo, billDate, totalAmount, dueDate, *patientNo*)

Charge (*invoiceNo*, serviceType, serviceDate, amountCharged)

1. Tuple cells hold single-valued inputs; records are unique; attributes are atomic. Thus, the table is in first normal form (1FN).

2. 1NF, and there are three primary keys and one non-prime key in this relation.

invoiceNo, serviceType, serviceDate → amountCharge: we are going to define how

these dependencies of each primary key follows:

invoiceNo→serviceType

serviceType→serviceDate

serviceType→amountCharged

Thereby, we have the following tables:

Charge1 (invoiceNo, *serviceType*)

serviceType (serviceType, serviceDate, amountCharged)

3. 2NF, there are no transitive dependencies; consequently, the table is also in 3FN.

4. The second functional dependency in Charge, serviceType → amountCharged is difficult to implement, because the serviceType can represent a missed appointment, a visit, a lab test, a prescription, or a procedure performed. Since a union is needed to represent these possibilities, we choose to ignore this dependency in the relational model, and allow amountCharged to remain in the Charge table. We assume the procedure for calculating bills will check the value of service type and be able to assign the correct charge. So, the table is also BCNF

Payment (*invoiceNo*, date, amountPaid, *insuranceCoPayer, patientPayer, insurancePolNoPayer)*

1. 1NF, values are single-valued, records are unique, and attributes are atomic
2. 1NF, and invoiceNo, date → amountPaid, insuranceCoPayer, patientPayer, insurancePolNoPayer
3. 2NF, insuranceCoPayer is unnecessary in Payment table because it is already a feature identifying patient in its table. patientPayer is also redundant because this attribute it is used anywherelese. Thus, it serves no purpose in this table as well. As a consequence, the table is now in 3NF.
4. The table is in 3NF, and has one superkey. Thereby, the Payment table is also in BCNF.

Payment (*invoiceNo*, date, amountPaid*, insurnacePolNoPayer)*

DiagnosisMenu (diagCode, diagName)

1. 1NF: DiagnosisMenu (daigCode, diagName)
2. 2NF: The table is in 2NF as it is in 1NF and its non-primary key attributes are fully functionally dependent on its primary key.
3. 3NF: The table is in 3NF as it is in 2NF and there are no transitive dependencies.
4. The table is in BCNF

ProcedureMenu (procCode, procName, cost)

1. 1NF: Procedure Menu (procCode, procName, cost)
2. 2NF: The table is in 2NF since it is already in 1NF and t’s non-primary key attributes are fully functionally dependant on the primary key.
3. 3NF: The table is in 3NF as it is in 2NF and there are no transitive dependencies.
4. The table is in BCNF

PrescriptionScript (scriptNo, *visitNo*, dateWritten, itemPrescribed, quantityPrescribed, directions, numberRefills) DAULET

1. Values are single-valued; records are unique; attributes are atomic => The table is in 1NF.
2. 1NF; itemPrescribed, quantityPrescribed and numberRefills are not functionally dependent on scriptNo. These attributes are dependent on specific Medication, so we add them to the PrescriptionMedication table.
3. 2NF, There is no transitive dependencies => The table is in 3NF
4. The table is in BCNF.

PrescriptionScript (scriptNo, *visitNo*, dateWritten, directions) DAULET

PrescriptionMedication (RXNumber, *scriptNo*, drugDispensed, quantityPrescribed, quantityDispensed , dateDispensed, refillsRemaining, cost) DAULET

1. Values are single-valued; records are unique; attributes are atomic => The table is in 1NF.
2. All of the attributes are functional dependent on the primary key => The table is in 2NF
3. There is transitive dependency drugDispensed -> cost.
4. The table is BCNF.

PrescriptionMedication (RXNumber, *scriptNo*, *drugDispensed*, quantityPrescribed, quantityDispensed , dateDispensed, refillsRemaining)

Drug(drugDispensed*, cost*)

LabTest (testNo, *prescriptionNo*, testType, testDate, testTime, cost, result)

1. Values are single-values; records are unique; attributes are atomic=> The table is in 1NF.
2. All of the attributes are functional dependent on the primary key => The table is in 2NF.
3. There is a transitive dependency testType -> cost.
4. Table in BCNF

LabTest(testNo, testDate, testTime, result)

TestType(testType, cost)

ProcedurePerformed (*visitNo*, *procCode*, result)

1. Values are single-values; records are unique; attributes are atomic => Table is in 1NF
2. All attributes are single-valued; records are unique; attributes are atomic=> The table is in 1NF
3. All non-primary attributes are functionally dependent on the primary key, in 2NF.
4. No transitive dependencies; table in 3NF
5. Table in BCNF

Diagnosis (*visitNo, diagCode*, dateOnset, symptoms, severity, prognosis)

1. Values are single-values; records are unique; attributes are atomic => The table is in 1NF.
2. All non-primary attributes are functionally dependent on the primary key.
3. No transitive dependencies; table is in 3NF
4. Table is in BCNF