1.

Insertion Anomaly: Adding a new patient into the database requires that we add a visit date, as well as provider information and the diagnosis for that visit. This is extra information that most likely is not useful if we simply want to add a new patient into the database.

Update Anomaly: If we need to update the age of the patient, we will utilize extra resources because we need to update the age of the patient in all rows with the corresponding PatNo. In this table, if we need to increase the age of the patient with PatNo = 1, we need to update rows 1 and 2.

Deletion Anomaly: We may need to delete the records of the patient with PatNo = 1. To do this, we must delete the associated visits they had, as well as the diagnoses performed. This could be undesired if we still want to see which providers performed which diagnoses on which dates, regardless of the patients involved.

### 2.

Steps of BCNF Normalization Process:

1. Group FDs with the same determinant (LHS)

PatNo -> PatAge, PatZip PatZip -> PatCity VisitNo -> VisitDate, PatNo ProvNo -> ProvSpecialty, ProvEmail VisitNo, ProvNo -> Diagnosis ProvEmail -> ProvNo

1. Create Tables using Determinant as primary key, adding foreign keys where necessary (**Bold** indicates primary key, Italics indicates foreign key)

* Patient(**PatNo**, PatAge, PatZip) FOREIGN KEY (PatZip) REFERENCES Zip
* Zip(**PatZip**, PatCity)
* Visit(**VisitNo**, VisitDate, PatNo) FOREIGN KEY (PatNo) REFERENCES Patient
* Provider(**ProvNo**, ProvSpecialty, ProvEmail) FOREIGN KEY (ProvEmail) REFERENCES ProviderEmail
* Appointment(**VisitNo**, **ProvNo**, Diagnosis) FOREIGN KEY (VisitNo) REFERENCES Visit FOREIGN KEY (ProvNo) REFERENCES Provider
* ProviderEmail(**ProvEmail**, ProvNo) FOREIGN KEY (ProvNo) REFERENCES Provider

1. Merge tables if one contains columns that are a subset of another table

Through this process we can only remove the ProviderEmail table. We can also remove the foreign key constraints on the other table definitions that reference these tables, as well as add additional UNIQUE constraints as required, resulting in our normalized list of final tables:

* Patient(**PatNo**, PatAge, PatZip) FOREIGN KEY (PatZip) REFERENCES Zip
* Zip(**PatZip**, PatCity)
* Visit(**VisitNo**, VisitDate, PatNo) FOREIGN KEY (PatNo) REFERENCES Patient
* Provider(**ProvNo**, ProvSpecialty, ProvEmail) UNIQUE ProviderEmail
* Appointment(**VisitNo**, **ProvNo**, Diagnosis) FOREIGN KEY (VisitNo) REFERENCES Visit FOREIGN KEY (ProvNo) REFERENCES Provider

### 3.

| **Functional Dependencies** | **Falsifying Rows** |
| --- | --- |
| OrdNo -> ItemNo | (1, 2), (3, 4) |
| OrdNo -> QtyOrd | (3, 4) |
| OrdNo -> CustNo | None |
| OrdNo -> CustBal | None |
| OrdNo -> CustDisc | None |
| OrdNo -> ItemPrice | (1, 2), (3, 4) |
| OrdNo -> OrdDate | None |