Normalization (BCNF)

* MaintenanceNeedsStationedAt(maintenanceID: INTEGER, complexityLevel: INTEGER, downtimeDuration: TIME, maintenanceDate: DATE, parts: VARCHAR[20], **coordinateX**: FLOAT (NOT NULL), **coordinateY**: FLOAT (NOT NULL), **aircraftID**: INTEGER (NOT NULL))
  + Find all closures
    - maintenanceID+ = {maintenanceID, complexityLevel, downtimeDuration, maintenanceDate, parts, coordinateX, coordinateY, aircraftID}
    - downtimeDuration+ = {downtimeDuration, complexityLevel}
    - parts+ = {parts, complexityLevel, downtimeDuration}
  + All the explicit and implicit non-trivial FDs:
    - maintenanceID → complexityLevel
    - maintenanceID → downtimeDuration
    - maintenanceID → maintenanceDate
    - maintenanceID → parts
    - maintenanceID → coordinateX
    - maintenanceID → coordinateY
    - maintenanceID → aircraftID
    - downtimeDuration → complexityLevel
    - parts → complexityLevel
    - parts → downtimeDuration
      * Note: No implicit FD found
  + Checking each FD for BCNF violations
    - maintenanceID → downtimeDuration, maintenanceDate, coordinateX, coordinateY, parts, complexityLevel
      * Since maintenanceID is a superkey, this FD does not violate BCNF
    - downtimeDuration → complexityLevel
      * downtimeDuration is not a superkey, therefore it violates BCNF. Hence, decompose MaintenanceNeedsStationedAt to:
        + MaintenanceNeedsStationedAt1(downtimeDuration, complexityLevel)
        + MaintenanceNeedsStationedAt2(downtimeDuration, maintenanceID, maintenanceDate, parts, **coordinateX, coordinateY, aircraftID**)
    - parts → complexityLevel
      * This FD is not considered since complexityLevel is not a part of the relation MaintenanceNeedsStationedAt2
    - parts → downtimeDuration
      * Parts is not a superkey, therefore it violates BCNF. Hence, decompose MaintenanceNeedsStationedAt2 to:
        + MaintenanceNeedsStationedAt3(parts, complexityLevel)
        + MaintenanceNeedsStationedAt4(parts, maintenanceID, maintenanceDate, **coordinateX, coordinateY, aircraftID**)
    - This is the final set of relations in BCNF
      * MaintenanceNeedsStationedAt1(downtimeDuration: TIME, complexityLevel: INTEGER)
      * MaintenanceNeedsStationedAt3(parts: VARCHAR[20] , complexityLevel: INTEGER)
      * MaintenanceNeedsStationedAt4(parts: VARCHAR[20], maintenanceID: INTEGER, maintenanceDate: DATE, **coordinateX**: FLOAT**, coordinateY**: FLOAT**, aircraftID**: INTEGER)
* Mechanic(mechanicID, name, contact, availabilityStatus):
  + Find all closures:
    - mechanicID+ = {mechanicID, name, contact, availabilityStatus} (Key)
    - name+ = {name, availabilityStatus}
    - contact+ = {contact, name, availabilityStatus}
  + All the explicit and implicit non-trivial FDs:
    - mechanicID → name
    - mechanicID → contact
    - mechanicID → availabilityStatus
    - name → availabilityStatus
    - contact → name
    - contact → availabilityStatus
  + Checking each FD for BCNF violations and decompose into BCNF
    - mechanicID → name, mechanicID → contact, mechanicID → availabilityStatus
      * Since mechanicID is a superkey, they do not violate BCNF
    - contact → name
      * contact is not a superkey, therefore it violates BCNF
      * Decompose Mechanic:
        + Mechanic1(mechanicID, availabilityStatus, contact)
        + Mechanic2(contact, name)
    - name → availabilityStatus
      * This FD is not considered since availabilityStatus is not a part of relation Mechanic2
    - contact → availabilityStatus
      * Contact is not a superkey, therefore it violates BCNF
      * Decompose Mechanic1:
        + Mechanic3(mechanicID, contact)
        + Mechanic4(contact, availabilityStatus)
  + This is the final set of relations in BCNF:
    - Mechanic2(contact: VARCHAR[20], name: VARCHAR[20])
    - Mechanic3(mechanicID: INTEGER, contact: VARCHAR[20])
    - Mechanic4(contact: VARCHAR[20], availabilityStatus: BOOLEAN)
* SquadronFrom(squadronID, dateFormed, name, aircraftCount, homeBaseID)
  + Find all closures:
    - squadronID+ = {squadronID, dateFormed, name, aircraftCount, homeBaseID}
    - Name+ = {Name, aircraftCount, dateFormed}
  + All the explicit and implicit non-trivial FDs:
    - squadronID → dateFormed
    - squadronID → name
    - squadronID → aircraftCount
    - squadronID → homeBaseID
    - name → dateFormed
    - name → aircraftCount
      * Note: No implicit FD found
  + Checking each FD for BCNF violations
    - squadronID → dateFormed, name, aircraftCount, homeBaseID (PK)
      * Since squadronID is a superkey, they are do not violate BCNF
    - name → dateFormed
      * Name is not a superkey, therefore it violates BCNF. Hence, decompose SquadronFrom to:
        + SquadronFrom1(name, dateFormed)
        + SquadronFrom2(name, squadronID, aircraftCount, homeBaseID)
    - Name → aircraftCount
      * Name is not a superkey, therefore it violates BCNF. Hence, decompose SquadronFrom2 to:
        + SquadronFrom3(name, aircraftCount)
        + SquadronFrom4(name, squadronID, homeBaseID)
  + This is the final set of relations in BCNF
    - SquadronFrom1(name: VARCHAR[20], dateFormed: DATE)
    - SquadronFrom3(name: VARCHAR[20], aircraftCount: INTEGER)
    - SquadronFrom4(name: VARCHAR[20], squadronID: INTEGER, **homeBaseID**: INTEGER)
* Aircraft(aircraftID, yearIntroduced, model, manufacturer, mainWeapon)
  + Find all closures:
    - aircraftID+ = {aircraftID, yearIntroduced, model, manufacturer, mainWeapon} (Key)
    - manufacturer, yearIntroduced+ = {manufacturer, yearIntroduced, model, mainWeapon}
    - model+ = {model, mainWeapon}
  + All the explicit and implicit non-trivial FDs:
    - aircraftID → yearIntroduced
    - aircraftID → model
    - aircraftID → manufacturer
    - aircraftID → mainWeapon
    - manufacturer, yearIntroduced → model
    - manufacturer, yearIntroduced → mainWeapon
    - model → mainWeapon
  + Checking each FD for BCNF violations and decompose into BCNF
    - aircraftID → yearIntroduced, aircraftID → model, aircraftID → manufacturer, aircraftID → mainWeapon
      * Since aircraftID is a superkey, they do not violate BCNF
    - manufacturer, yearIntroduced → model
      * manufacturer, yearIntroduced is not superkey, therefore it violates BCNF
      * Decompose Aircraft:
        + Aircraft1(aircraftID, yearIntroduced, manufacturer, mainWeapon)
        + Aircraft2(yearIntroduced, manufacturer, model)
    - manufacturer, yearIntroduced → mainWeapon
      * manufacturer, yearIntroduced is not superkey in the relation Aircraft1, therefore it violates BCNF
      * Decompose Aircraft1:
      * Aircraft3(aircraftID, yearIntroduced, manufacturer)
      * Aircraft4(yearIntroduced, manufacturer, mainWeapon)
    - model → mainWeapon
      * This FD is not considered since mainWeapon is not a part of relation Aircraft2
  + This is the final set of relations in BCNF
    - Aircraft2(yearIntroduced: INTEGER, manufacturer:VARCHAR[20], model: VARCHAR[20])
    - Aircraft3(aircraftID: INTEGER, yearIntroduced: INTEGER, manufacturer: VARCHAR[20])
    - Aircraft4(yearIntroduced: INTEGER, manufacturer: VARCHAR[20], mainWeapon: VARCHAR[20])
* RepairBase(coordinateX, coordinateY, city, country, certification, specialization)
  + Find all closures:
    - coordinateX, coordinateY+ = {coordinateX, coordinateY, city, country, certification, specialization} (Key)
    - city+ = {city, country}
    - certification+ = {certification, specialization}
  + All the explicit and implicit non-trivial FDs:
    - coordinateX, coordinateY → city
    - coordinateX, coordinateY → country
    - coordinateX, coordinateY → certification
    - coordinateX, coordinateY → specialization
    - city → country
    - certification → specialization
  + Checking each FD for BCNF violations and decompose into BCNF
    - coordinateX, coordinateY → city, coordinateX, coordinateY → country, coordinateX, coordinateY → certification, coordinateX, coordinateY → specialization
      * Since coordinateX, coordinateY is a superkey, they do not violate BCNF
    - city → country
      * city is not superkey, therefore it violates BCNF
      * Decompose RepairBase:
        + RepairBase1(coordinateX, coordinateY, city, certification, specialization)
        + RepairBase2(city, country)
    - certification → specialization
      * certification is not a superkey, therefore it violates BCNF
      * Decompose RepairBase1:
      * RepairBase3(coordinateX, coordinateY, city, certification)
      * RepairBase4(certification, specialization)
  + This is the final set of relations in BCNF
    - RepairBase2(city: VARCHAR[20], country: VARCHAR[20])
    - RepairBase3(coordinateX: FLOAT, coordinateY: FLOAT, city: VARCHAR[20], certification: VARCHAR[20] )
    - RepairBase4(certification: VARCHAR[20], specialization: VARCHAR[20] )
* HomeBase(homeBaseID: INTEGER, capacity: INTEGER, commissionDate: DATE, region: VARCHAR[20], baseType: VARCHAR[20])
  + Find all closures
    - homeBaseID+ = {homeBaseID, capacity, commisionDate, region, baseType}
    - commissionDate, baseType, region+ = {commissionDate, baseType, capacity}
    - commissionDate = {commisionDate, baseType}
  + All the explicit and implicit non-trivial FDs:
    - homeBaseID → capacity
    - homeBaseID → commissionDate
    - homeBaseID → region
    - homeBaseID → baseType
    - commissionDate, baseType, region → capacity
    - commissionDate → baseType
      * Note: No implicit FDs found
  + Checking each FD for BCNF violations and decompose into BCNF
    - homeBaseID → capacity, commissionDate, region, baseType (PK)
      * Since homeBaseID is a superkey, they do not violate BCNF
    - CommissionDate, baseType, region → capacity
      * CommissionDate, baseType, region is not a superkey, therefore it violates BCNF. Hence, decompose HomeBase to:
        + HomeBase1(CommissionDate, baseType, region, capacity)
        + HomeBase2(CommissionDate, baseType, region, homeBaseID)
    - CommissionDate → baseType
      * CommissionDate is not a superkey, therefore it violates BCNF. Hence, decompose HomeBase2 to:
        + HomeBase3(commissionDate, baseType)
        + HomeBase4(commissionDate, region, homeBaseID)
  + This is the final set of relations in BCNF
    - HomeBase1(commissionDate: DATE, baseType: VARCHAR[20], region: VARCHAR[20], capacity: INTEGER)
    - HomeBase3(commissionDate: DATE, baseType: VARCHAR[20])
    - HomeBase4(commissionDate: DATE, region: VARCHAR[20], homeBaseID: INTEGER)
* PilotsAndCrew(crewID, name, role, rank)
  + Find all closures:
    - crewID+ = {crewID, name, role, rank} (Key)
    - name+ = {name, role, rank}
  + All the explicit and implicit non-trivial FDs:
    - crewID → name
    - crewID → role
    - crewID → rank
    - name → role
    - name → rank
  + Checking each FD for BCNF violations and decompose into BCNF:
    - crewID → name, crewID → role, crewID → rank
      * Since crewID is a superkey, they do not violate BCNF
    - name → role
      * name is not superkey, therefore it violates BCNF
      * Decompose PilotsAndCrew:
        + PilotsAndCrew1(crewID, name, rank)
        + PilotsAndCrew2(name, role)
    - name → rank
      * name is not superkey, therefore it violates BCNF
      * Decompose PilotsAndCrew1:
        + PilotsAndCrew3(crewID, name)
        + PilotsAndCrew4(name, rank)
  + Result:
    - PilotsAndCrew2(name: VARCHAR[20], role: VARCHAR[20])
    - PilotsAndCrew3(crewID: INTEGER, name: VARCHAR[20])
    - PilotsAndCrew4(name: VARCHAR[20], rank: VARCHAR[20])
* Missions(missionID, missionDate, missionLocation, weather, outcome, duration):
  + Find all closures:
    - missionID+ = {missionID, missionDate, missionLocation, weather, outcome, duration} (Key)
    - missionDate, missionLocation+ = {missionDate, missionLocation, weather}
    - outcome+ = {outcome, duration}
  + All the explicit and implicit non-trivial FDs:
    - missionID → missionDate
    - missionID → missionLocation
    - missionID → weather
    - missionID → outcome
    - missionID → duration
    - missionDate, missionLocation → weather
    - outcome → duration
  + Checking each FD for BCNF violations and decompose into BCNF:
    - missionID → missionDate, missionID → missionLocation, missionID → weather, missionID → outcome, missionID → duration
      * Since missionID is a superkey, they do not violate BCNF
    - missionDate, missionLocation → weather
      * missionDate, missionLocation is not superkey, therefore it violates BCNF
      * Decompose Missions:
        + Missions1(missionID, missionDate, missionLocation, outcome, duration)
        + Missions2(missionLocation, missionDate, weather)
    - outcome → duration
      * outcome is not superkey, therefore it violates BCNF
      * Decompose Missions1:
        + Missions3(missionID, missionDate, missionLocation, outcome)
        + Missions4(outcome, duration)
  + Result:
    - Missions2(missionLocation: VARCHAR[20], missionDate: DATE, weather: VARCHAR[20])
    - Missions3(missionID: INTEGER, missionDate: DATE, missionLocation: VARCHAR[20], outcome: CHAR[1])
    - Missions4(outcome: CHAR[1], duration: TIME)

**Note: The associative relations created by M:N cardinality are omitted from the normalization process, since the are all already in BCNF.**