Electricity Grid Database Management System
Normalization and Schema Refinement

202303005 : Hemil Patel
2020303015 : Akshat shah
202303001 : Preet Siddhapura

Functional Dependencies

Along with current normal form

In General we have excluded the listing of trivial functional dependencies and excluded the functional dependencies of super keys of each relation.

- **Customer**

  1. Customer_ID → {Customer_Name, Phone_Number, Connection_Status, Billing_Cycle, Block/Flat_no., Street, City, District, State, Pin-Code, Customer_Type_ID}
  2. Phone_Number (Candidate Key) → {Customer_ID,Customer_Name, Connection_Status, Billing_Cycle, Block/Flat_no., Street, City, District, State, Pin-Code, Customer_Type_ID}
  3. (Customer_Name, Block/Flat_no., Street, Pin-Code) (Candidate Key) → {Customer_ID}
  4. Pin-Code → {City, District, State}

  Customer is in 1NF, since it has partial dependency. Pin-Code is a prime Attribute, while non prime attributes {City, District, State} depend on it.

- **Bill**

  1. Bill_ID → {Total_Price, Billing_Date, Payment_Status, Customer_ID, Meter_ID, Rate_ID}
  2. Meter_ID → {Customer_ID}
  3. (Billing_Date, Customer_ID) → {Rate_ID}
  4. (Billing_Date, Meter_ID) → {Rate_ID} (By transitivity of 2 and 3)
  5. (Billing_Date, Meter_ID) → {Bill_ID} (Then, by transitivity, (Billing_Date, Meter_ID) is a candidate key)

  Bill is in 1NF, since it has partial dependency. Meter_ID is a prime Attribute, while non prime attribute {Customer_ID} depends on it.

- **Substation**

  1. Substation_ID → {Substation_Name, Area, City, District, State, Pin_Code, Voltage_Level, Transformer_Capacity, Circuit_Breaker, Capacity, Status}
  2. (Area, Pin_Code) → {Substation_ID}
     (Thus , Candidate key)
  3. Area → {Substation_Name}
  4. Pin-Code → {City, District, State}

  Substation is in 1NF, since it has partial dependency. Area and Pin-Code are prime Attributes, while non prime attributes:

  – {Substation_Name} depends on Area

- {City, District, State} depends on Pin-Code.

- **Feeder**

  1. Feeder_ID → {Area_Name, Voltage_Level, Capacity, Load_Profile, Circuit_Breaker_Rating, Number_Of_Meters, Substation_ID}

  Feeder is in BCNF, since it has no partial or transitive dependencies. Also all it's FDs have super key on the left hand side.

- **Meter**

  1. Meter_ID → {Current_Reading, Status, Installation_Date, Last_Reading_Date, Feeder_ID, Customer_ID, Customer_Type_ID}
  2. Customer_ID → {Customer_Type_ID}

  Meter is in 2NF, since it has no partial dependencies. But it has transitive dependency.
  Customer_ID depends on Meter_ID
  while Customer_Type_ID depends on Customer_ID.

- **Outage**

  1. Outage_ID → {Start_Date_Time, End_Date_Time, Status, Outage_Type, Cause, Maintenance_ID}
  2. Maintenance_ID → {Outage_ID} (Since, it has one to one relation with outage. Thus, its a candidate key)

  Outage is in BCNF, since it has no partial or transitive dependencies. Also all it's FDs have super key on the left hand side.

- **Maintenance Team (BCNF)**

  1. Team_ID → {Team_Type}

- **Customer_Type**

  1. Customer_Type_ID → {Type_Name, Description}
  2. Type_Name → {Customer_Type_ID, Description}

  Customer_Type is in BCNF, since it has no partial dependencies neither it has transitive dependency of non prime attributes . While all the functional dependencies has super keys on the left hand side.

- **Employee**

  1. Employee_ID → {Employee_Name, Role, Department, Salary, Contact_Info, Substation_ID, Feeder_ID, Team_ID}
  2. Contact_Info (Candidate key) → {Employee_Name, Role, Department, Salary, Employee_ID, Substation_ID, Feeder_ID, Team_ID}
  3. **NOTE:** We have recently understood that all feeders are inherently embedded in substations. Therefore, an employee is associated with a substation only. Therefore, we must remove Feeder_ID from employee.

  Employee is in BCNF, since it has no partial dependencies neither it has transitive dependency of non prime attributes . While all the functional dependencies has super keys on the left hand side.

- **Rate**

  1. Rate_ID → {Electricity_Rate, Rate_Start_Date, Rate_End_Date, Customer_Type_ID}
  2. (Customer_Type_ID,Rate_Start_Date) → {Rate_ID}(Thus by transitivity, it is a candidate key)

3. (Customer_Type_ID,Rate_End_Date) → {Rate_ID}(Thus by transitivity, it is a candidate key)

Rate is in BCNF, since it has no partial dependencies neither it has transitive dependency of non prime attributes . While all the functional dependencies has super keys on the left hand side.

- **Interconnects**

  We will enumerate all the trivial Functional Dependencies for **Interconnects**. Since it has no non-prime attributes.

  1. {Substation1_ID, Substation2_ID} → {Substation1_ID}
  2. {Substation1_ID, Substation2_ID} → {Substation2_ID}

  Interconnects is in BCNF, since it has only Prime attributes, thus it has only trivial FDs.

- **Power Source**

  1. Power_Source_ID → {Type, Status, Capacity, Generation_Data, Area, City, State, Pin-Code, Owner_ID, Substation_ID}
  2. Pin-Code → {City, DIstrict, State}
  3. (Owner_ID,Type,Pin-Code) → {Power_Source_ID} (Thus, by transitivity, this becomes a candidate key)

  Power Source is in 1NF, since it has partial dependency. Pin-Code is a prime Attribute, while non prime attributes {City, District, State} depend on it.

- **Power Source Owner**

  1. Owner_ID → {Name, Contact_Info, Power_Source_Ownership_Details, Office_No., Street, District, City, State, Pin-Code}
  2. Contact_Info → {Owner_ID} (Thus, contact info is a candidate key)
  3. (Name,Office_No., Street,Pin-Code) → {Owner_ID } (Thus,it is a candidate key)
  4. Pin-Code → {District, City, State}

  Power Source Owner is in 1NF, since it has partial dependency. Pin-Code is a prime Attribute, while non prime attributes {City, District, State} depend on it.

- **Maintenance_Schedule**

  1. Maintenance_ID → {Maintenance_Type, Status, Date, Start_Time, End_Time, Substation_ID, Feeder_ID, Team_ID}
  2. (Start_Date_Time,Team_ID) → {Maintenance_ID}
     Reason: At a given time a given team can only be performing one maintenance
  3. (End_Date_Time,Team_ID) → {Maintenance_ID}
     Reason: A given team can only be performing one maintenance before starting another one.
  4. (Maintenance_Type,Start_Date_Time,Substation_ID) → {Maintenance_ID}
     Assumption: At a given time for a given substation and a given type of maintenance type , there can only be one maintenance. Which implies we can not allocate more than one team for a given maintenance type at a given time and same substation.

  Maintenance_Schedule is in BCNF, since it has no partial dependencies neither it has transitive dependency of non prime attributes . While all the functional dependencies has super keys on the left hand side.

  NOTE: We have recently understood that all feeders are inherently embedded in substations. Therefore, an maintenance schedule is associated with a substation only. Therefore, we must remove Feeder_ID from Maintenance schedule.

- **Affected_Area**

  1. Affected_Area_ID → {Area_Type, Area, City, State, Pin-Code, Outage_ID}

2. (Area, Pin-Code,Outage_ID) → {Affected_Area_ID} (Thus, a candidate key)
3. Pin-Code → {City, State, District}

Affected_Area is in 1NF, since it has partial dependency. Pin-Code is a prime Attribute, while non prime attributes {City, District, State} depend on it.

Anomalies in Relations

- **Pin-Code Anomalies**
  These anomalies are present in the following relations:

  1. Customer
  2. Substation
  3. Power Source
  4. Power Source Owner
  5. Affected Area

  Therefore, we are explaining the anomalies for one relation CUSTOMER, but it holds true for all the 5 relations stated above

  - Insertion Anomalies: If we want to add a Pin-Code for a new Area, we have to compulsory add customer for that Pin-Code. But that is not how it should work. We must be able to add a new Pin-Code without having customers for it.
  - Deletion Anomalies: If we delete all Customer for a given Pin-Code, the Pin-Code to gets deleted, which should not happen.
  - Update Anomalies: If we are going to update the pincode for a given city, we will have to update every tuple of every customer associated with that city. In this process, there can occur inconsistencies.

- **Bill Anomalies**

  - Update Anomalies: Suppose there has been a mistake in associating the correct meter_ID with Customer_ID. Then when we are going to correct the mistake by updating the Meter_ID with the correct Customer_ID, we must also need to update every tuple of every bill associated with that meter_ID. In this process, there can occur inconsistencies.

- **Meter Anomalies**

  - Since Customer_Type_ID depends on Customer_ID, if you need to update a customer's type, you would have to update every occurrence of that Customer_ID wherever the Meter_ID appears. This can lead to inconsistency if one of the instances is not updated properly.

Decomposition of Relations

We will be using BCNF Decomposition Algorithm to Convert our Relations to BCNF. Since this algorithm is lossless but doesn't necessarily preserve FDs, for the relations where there is a loss of FDs, we will mention it. Furthermore, we will use 3NF synthesis algorithm for those relations.

- **Customer**
  We have already written minimal FD Set of Customer above. Now we will find the FDs which do not satisfy BCNF.

  1. Pin-Code → {City, District, State}
     This FD violates BCNF, since left hand side is not a super key.
  2. $(\text{Pin-Code})^+$ ={Pin-Code,City, District,State}
  3. We create a new relation for the $(\text{Pin-Code})^+$

     Pin-Code(Pin-Code, City, District, State)

4. Thus we have Remaining Customer Relation.

   Customer(Customer_ID, Customer_Name, Phone_Number, Connection_Status, Billing_Cycle, Block/Flat_no., Street, Pin-Code, Customer_Type_ID)

5. Both the relations are now in BCNF, and there is no loss of FDs. Certain FDs which may appear to be lost, are actually implied by transitivity.

- **Bill**

1. Meter_ID $\rightarrow$ {Customer_ID}
   This FD violates BCNF, since left hand side is not a super key.
2. $(\text{Meter\_ID})^+$ ={Meter_ID,Customer_ID}
3. We create a new relation for the $(\text{Meter\_ID})^+$

   Meter_ID(Meter_ID, Customer_ID)
4. BUT, There already exists a meter_ID relation which comsists both of these. So we just need to Remove Customer_ID from the Table As it is making the Relation Transitively Dependent.
   Bill(Bill_ID, Total_Price, Billing_Date, Payment_Status, Meter_ID, Rate_ID)
5. Both the relations are now in BCNF, and there is no loss of FDs. Certain FDs which may appear to be lost, are actually implied by transitivity.

- **Substation**

1. Since we have realised there is no point in storing substation_name data in another table. Therefore we will simply remove it.
2. Pin-Code $\rightarrow$ {City, District, State}
   This FD violates BCNF, since left hand side is not a super key.
3. $(\text{Pin-Code})^+$ ={Pin-Code,City, District,State}
4. We create a new relation for the $(\text{Pin-Code})^+$

   Pin-Code(Pin-Code, City, District, State)
5. Thus we have Remaining Substation Relation.

   Substation(Substation_ID, Area, Pin_Code, Voltage_Level, Transformer_Capacity, Circuit_Breaker, Capacity, Status)
6. Both the relations are now in BCNF, and there is no loss of FDs. Certain FDs which may appear to be lost, are actually implied by transitivity.

- **Feeder**
  Feeder(Feeder_ID, Area_Name, Voltage_Level, Capacity, Load_Profile, Circuit_Breaker_Rating, Number_Of_Meters, Substation_ID)

- **Meter**

1. Customer_ID $\rightarrow$ {Customer_Type_ID} is an FD which doesn't follow BCNF rules, since left hand side is not a super key.
2. $(\text{Customer\_ID})^+$ ={Customer_Type_ID, Customer_ID}
3. We create a new relation for the $(\text{Customer\_ID})^+$

   Customer_ID(Customer_ID,Customer_Type_ID)
4. BUT , we already have a relation Customer_ID which consists of both these attributes.Thus, we have to simply remove Customer_Type_ID from this relation

5. Thus we have Remaining Meter Relation.

   Meter(Meter_ID, Current_Reading, Status, Installation_Date, Last_Reading_Date, Feeder_ID, Customer_ID)

6. Both these relations are now in BCNF form.

- **Outage**
  Outage(Outage_ID, Start_Date_Time, End_Date_Time, Status, Outage_Type, Cause, Maintenance_ID)
- **Maintenance Team**
  Team(Team_ID, Team_Type)
- **Customer Type**
  Customer Type (Customer_Type_ID, Type_Name, Description)
- **Employee**
  Employee(Employee_ID, Employee_Name, Role, Department, Salary, Contact_Info, Substation_ID, Team_ID)
- **Rate**
  Rate(Rate_ID, Electricity_Rate, Rate_Start_Date, Rate_End_Date, Customer_Type_ID)
- **Power Source**
  1. Here too we have the FD of Pin-Code that violates the BCNF rules. Thus, we will perform identical steps as performed in Customer and Substation. We will remove City, State, and District from the Relation as we have Created new Table for it.

     Power Source(Power_Source_ID, Type, Status, Capacity, Generation_Data, Area, Pin-Code, Owner_ID, Substation_ID)

- **Power Source Owner**
  1. We would need to remove City, State, and District from this Relation too.
     Power Source Owner(Owner_ID, Name, Contact_Info, Power_Source_Ownership_Details, Office_No., Street, Pin-Code)
- **Maintenance Schedule**
  Maintenance(Maintenance_ID, Maintenance_Type, Status, Start_Date_Time, End_Date_Time, Substation_ID, Team_ID)
- **Affected Area**
  1. We would need to remove City, State, and District from this Relation too.
     Affected Area(Affected_Area_ID, Area_Type, Area, Pin-Code, Outage_ID)
- **Substation_Connects**
  Substation_Connects(Substation1_ID, Substation2_ID)