Abel Marin

February 26, 2020

1. Problem 1
   1. {a}+ = {a,c,d}

{b}+ = {b,c}

{c}+ = {c}

{d}+ = {d}

{a,b}+ = {a,b,c,d}

{a,c}+ = {a,c,d}

{a,d}+ = {a,c,d}

{b,c}+ = {b,c}

{b,d}+ = {a,b,c,d}

{c,d}+ = {c,d}

{a,b,c}+ = {a,b,c,d}

{a,b,d}+ = {a,b,c,d}

{a,c,d}+ = {a,c,d}

{b,c,d}+ = {a,b,c,d}

{a,b,c,d}+ = {a,b,c,d}

* 1. Superkeys: {a,b} {b,d} {a,b,c} {a,b,d} {b,c,d} {a,b,c,d}
  2. Candidate Keys: {a,b} {b,d}

1. Problem 2
   1. {LawyerID,ClientID ,Date}
   2. ClientID -> ClientName, Company, City and Company -> City both violate the BCNF condition. This is because neither of them has a determinate which is a superkey. Lawyer, ClientID, Date -> Hours does not violate the BCNF condition because the determinate is a superkey.
   3. We begin with the initial schema CONTACT(LawyerID, ClientID, ClientName, Company, City, Date, Hours) and proceed by removing all functional dependencies which violate BCNF. Two functional dependencies violate BCNF which are ClientID -> ClientName, Company, City and Company -> City. We start with ClientID -> ClientName, Company, City. In order to remove this BCNF violation we must decompose CONTACT into 2 relations R1 and R2:

R1(LawyerID, ClientID, Date, Hours) R2(ClientID, ClientName, Company, City).

Next we look to see if there is another functional dependency which violates BCNF and that is Company -> City in R2. In order to remove this dependency, we must decompose this relation into R2 and R3:

R2(ClientID, ClientName, Company) R3(Company, City).

Once we are done with this we check again to see if there are any functional decencies which violate BCNF and there are none. Thus, the final collection of relations which is in BCNF is:

R1(LawyerID, ClientID, Date, Hours) R2(ClientID, ClientName, Company) R3(Company, City)

1. Problem 3
   1. F on R1 = {A->B,C->B,C->A}
   2. F on R2 = {C->D}
   3. This decomposition D does not preserve the set of dependencies F. In order for the decomposition D to preserve the set of dependencies F the union of the projections must be equal to the original set of functional dependencies. The union of F on R1 and F on R2 is {A->B,C->B,D} This is missing the functional dependency D->A and it cannot be derived from the set of dependencies.