Erica Filgueras
CSC 355- hw5
Section 501

1. Consider the relation R with schema R(a, b, c, d), and the following set of functional dependencies:
F = { a -> c,d ; b,d -> a ; b -> c }.

a. For each of the fifteen non-empty subsets S of the set of attributes {a, b, c, d}, find the closure of S using the set of functional dependencies F.
Closures

A= {A,C,D}
B= {B,C}
C={C}
D={D}
AB= {A,B,C,D}
AC= {A,C,D}
BC= {B,C}
BD= {A,B,C,D}
CD= {C,D}
ABC= {A,B,C,D}
ABD= {A,B,C,D}
ACD= {A,C,D}
BCD= {A,B,C,D}
ABCD= {A,B,C,D}

b. superkeys
{AB,BD,ABC,ABD,BCD,ABCD}

c. candidate keys
{AB, BD}

2. Consider the following relational schema with seven attributes, which stores information on contacts between lawyers at a law firm and their clients:

CONTACT(LawyerID, ClientID, ClientName, Company, City, Date, Hours)

ClientID -> ClientName, Company, City
LawyerID, ClientID, Date -> Hours
Company -> City

a. candidate key: {LawyerID, ClientID, Date }
b. Two that violate BCNF:

* ClientID -> ClientName, Company, City
- The closure of ClientID does not contain LawyerID, Date, and Hours. Therefore, ClientID is not a superkey and it violate BCNF.

* Company -> City
- The closure of company also doesn't contain every attribute in the relation of contact. Since Company isn't a superkey, it violates the rules of BCNF.

c. **Construct a decomposition of CONTACT into a collection of relations that are all in BCNF.**
Use the algorithm given in class, and show your work

Algorithm:

While there is some Q in D that is not in BCNF:
- **Choose a Q that's not in BCNF**
- **Find an X-> Y that violates BCNF**
- **Replace Q with 2 relations:**

(Q-Y) and (XUY)

Q's not in BCNF:

R(LawyerID, ClientID, ClientName, Company, City, Date, Hours)

I. ClientID (X)-> ClientName, Company, City(Y)

R1(LawyerID,ClientID,Date, Hours) =(Q-Y)
R2(ClientID,ClientName, Company, City) =(X U Y)

II. Company (X)-> City(Y)

R3(ClientID,ClientName, Company) =(Q-Y from R2)
R4(Company, City) =(X U Y)

3. For the universal relation R(A, B, C, D), consider the decomposition D consisting of R_1(A, B, C) and
R_2(C, D), and the set F of functional dependencies { A -> B ; C -> B,D ; D ->A }.

Split into singletons only on the right:

FDs:

A -> B
C -> B
C -> D

D ->A

a. Compute the projection of F on R_1.
R_1(A, B, C)
= {A-> B, C -> B}

b. Compute the projection of F on R_2.
R_2(C, D),
={C -> D}

c. Does the decomposition D preserve the set of dependencies F? Give a detailed explanation
why or why not.
****
find the closures of A and C using R1 and R2's functional dependency:

Closures from F:
{ A -> B ; C -> B,D ; D ->A }
A= {A,B}
C= {A,B,C,D}

Closures from projection:
R1,R2 = {A-> B, C -> B,C -> D}

A= {A,B}
C= {D,B,C}

The closures we arrived using the projection of the relations didn't match the original functional dependency. Therefore, didn't preserve the set of dependencies in F.