Name :- Merga Tafa

Subject :- CSC 355

Section:-510

Due date 02/26/2020

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.
   1, {a}+ = {a, c, d}
   2, {b}+ = {b, c}
   3, {c}+ = {c}
   4, {d}+ = {d}
   5, {a, b}+ = {a, b, c, d}
   6, {a, c}+ = {a, c, d}
   7, {a, d}+ = {a, d, c}
   8, {b, c}+ = {b, c}
   9, { b, d}+ ={b, d, a, c}
   10, {c, d}+ = {c, d}
   11, {a, b, c}+ ={a, b, c, d}
   12, {a, b, d}+ ={a, b, d, c}
   13, {b, c, d}+ ={b, c, d, a}
   14, {a, c, d}+ ={a, c, d,}
   15, {a, b, c, d}+ ={a, b, c, d}

b. List all of the superkeys of R.

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

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

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

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

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

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

c. List all of the candidate keys of R.

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

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

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)

The functional dependencies in CONTACT are:

ClientID -> ClientName, Company, City

LawyerID, ClientID, Date -> Hours

Company -> City

a. CONTACT has only one candidate key. What is it?
{LawyerID, ClientID, Date}

b. Two of the three functional dependencies in CONTACT violate the condition for CONTACT to be in BCNF. Identify which two they are, and explain how they violate the BCNF condition.

The two functional dependencies that violate BCNF condition in CONTACT are {ClientID -> ClientName, Company, City} and { Company -> City }. The reason is that ClientID and Company are not superkeys in the relations

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. (That is, at each step, state which functional dependency you are removing from which relation, and what the resulting decomposition is after you remove it. Answers that show only the final result without showing the steps taken to obtain it will receive little to no credit.)

1, Remove ClientID -> ClientName, Company, City from CONTACT as (X ->Y similar to removing Y where X is not Superkey in functional dependencies relations)

R1(LawyerID, ClientID, Date, Hours)

2, Then, with X union Y algorithms application R2 becomes R2(ClientID, ClientName, Company, City) as similar to (X ->Y become X union Y where X is not a Supper key in functional dependencies in the relation)

Therefore, the decomposition becomes

R1(LawyerID, ClientID, Date, Hours)

R2(ClientID, ClientName, Company, City)

3. For the universal relation R(A, B, C, D), consider the decomposition D consisting of R1(A, B, C) and R2(C, D), and the set F of functional dependencies { A ->B ; C ->B,D ; D ->A }.

a. Compute the projection of F on R1.
   {A -> B, C -> B}

b. Compute the projection of F on R2.
   {C->D}

c. Does the decomposition D preserve the set of dependencies F? Give a detailed explanation why or why not. (Don't just repeat back the definition of the dependency preservation property, but rather show why the decomposition D either has or does not have this property by showing whether or not each functional dependency in F can be derived from the union of the projections.)
   The decomposition D does not preserve the set of dependencies because only A-> B and C->B come from the projection on R1. C->D is the only dependency come from the projection on R2. The union of these dependencies do not give set of F because of dependency of D->A is missing. Therefore, the union of the two(R1 and R2) doesn't preserve dependencies that come from F.

.