**<Datebase 201433707 이형욱>**

**8.13 Show that the decomposition in Practice Exercise 8.1 is not a dependency-reserving decomposition.**

Answer: The dependency B → D is not preserved. F1, the restriction of F to (A, B, C) is A → ABC, A → AB, A → AC, A → BC, Exercises 15 A → B, A → C, A → A, B → B, C → C, AB → AC, AB → ABC, AB → BC, AB → AB, AB → A, AB → B, AB → C, AC (same as AB), BC (same as AB), ABC (same as AB). F2, the restriction of F to (C, D, E) is A → ADE, A → AD, A → AE, A → DE, A → A, A → D, A → E, D → D, E (same as A), AD, AE, DE, ADE (same as A). (F1 ∪ F2) + is easily seen not to contain B → D since the only FD in F1 ∪ F2 with B as the left side is B → B, a trivial FD. We shall see in Practice Exercise 8.15 that B → D is indeed in F +. Thus B → D is not preserved. Note that C D → ABC DE is also not preserved. A simpler argument is as follows: F1 contains no dependencies with D on the right side of the arrow. F2 contains no dependencies with B on the left side of the arrow. Therefore for B → D to be preserved there must be an FD B → a in F + 1 and a → D in F + 2 (so B → D would follow by transitivity). Since the intersection of the two schemes is A, a = A. Observe that B → A is not in F + 1 since B + = B D. 8.21 Normalize the following schema, with given constraints, to 4NF.

books(accessionno, isbn, title, author, publisher)

users(userid, name, deptid, deptname)

accessionno → isbn

isbn → title

isbn → publisher

isbn →→ author

userid → name

userid → deptid

deptid → deptname

**8.21 Normalize the following schema, with given constraints, to 4NF.**

**books(accessionno, isbn, title, author, publisher)**

**users(userid, name, deptid, deptname)**

**accessionno→isbn**

**isbn→title**

**isbn→publisher**

**isbn→→author**

**userid→name**

**userid→deptid**

**deptid→deptname**

Answer:

In books we see that

isbn →→ title,publicher,author

And yet ,isbn in not a superkey. Thus, we break books into

books\_accnno(accessionno,isbn)

books\_details(isbn,title,publisher,author)

After this , we still have

isbn→→author

But neither is isbn a primary key of book\_details, not are the attributes of book\_details equal to {isbn} U {author}. Therefore we decompose book\_details again into

books\_details1(isbn,title,publisher)

books\_authors(isbn,author)

Similarly in users,

deptid→deptname

And yet, deptid is not a super key. Hence we break users to

users(userid,name,deptid)

departments(deptid,depname)

We verify that there are no further functional or multivalued dependencies that cause violation of 4NF, so the final set of relations are:

books\_accnno(accessionno,isbn)

books\_details1(isbn,title,publisher)

books\_author(isbn,author)users(userid,name,deptid)

departments(deptid,deptname)

**8.27 Using the functional dependencies of Practice Exercise 8.6, compute B+**.

Answer : Computing B+ by the algorithm in Figure 8.8 we start with result = [B]. Considering FDs of the form β ->γ in F, we find that the only dependencies satisfying β ⊆ result are B -> B and

B-> D. Therefore result = { B, D } . No more dependencies in F apply now.

Therefore B+ = { B, D }

**8.29 Consider the following set F of functional dependencies on the relation schema r(A, B, C, D, E, F):**

**A → BCD**

**BC → DE**

**B → D**

**D → A**

1. **Compute B+.**

B -> BD third dependency

BD ->ABD fourth dependency

ABD ->ABCD first dependency

ABCD ->ABCDE second dependency

1. **Prove (using Armstrong’s axioms) that AF is a superkey.**

A->BCD Given

A->ABCD Augmentation with A

BC->DE Given

ABCD->ABCDE Augmentation with ABCD

A->ABCDE Transitivity

AF->ABCDEF Augmentation with F

1. **Compute a canonical cover for the above set of functional dependencies F; give each step of your derivation with an explanation.**

Answer: we see that D is extraneous in dep. 1and 2, because of dep. 3. Removing these two, we get the now set of rules

A->BC

BC->E

B->D

D->A

Now notice that B+ is ABCDE, and in particular, the FD B->E can be determined from this set. Thus, the attribute C is extraneous in the third dependency. Removing this attribute, and combining with the third FD, we get the final canonical cover as:

A->BC

B->DE

D->A

Here, no attribute is extraneous in any FD.

1. **Give a 3NF decomposition of r based on the canonical cover.**

Answer: We see that there is no FD in the canonical cover such that the set of attributes is a subset of any other FD in the canonical cover. Thus, each each FD gives rise to its own relation,giving

r1(A, B ,C)

r2(B, D ,E)

r3(D, A)

Now the attribute F is not dependent on any attribute. Thus, it must be a part of every superkey. Also, non of the relations in the above schema have F, and hence, none of them have a superkey. Thus, we need to add a new relation with a superkey.

r4(A, F)

**e. Give a BCNF decomposition of r using the original set of functional dependencies.**

Answer: We start with

r(A, B, C, D, E, F)

we see that the relation in not in BCNF because of the first FD. Hence, we decompose it accordingly to get

r1(A, B, C, D)r2(A, E, F)

Now we notice that A->E is an FD in F+, and causes r2 to violate BCNF. Once again, decomposing r2 gives

r1(A, B, C, D)r2(A, F)r3(A, E)

this schema is now n BCNF

**f. Can you get the same BCNF decomposition of r as above, using the canonical cover?**

Answer: if we use the functional dependencies in the preceding canonical cover directly, we cannot get the above decomposition. However, we can infer the original dependencies from the canonical cover, and if we use those for BCNF decomposition, we would be able to get the same decomposition.

**8.33 Given a relational schema r(A, B,C, D), does A →→ BC logically imply A →→ B and A →→ C? If yes prove it, else give a counter example.**

Answer: A→→BC holds on the following table:

r:

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
| a1 | b1 | c1 | d1 |
| a2 | b2 | c2 | d2 |
| a3 | b3 | c3 | d3 |
| a4 | b4 | c4 | d4 |

If A→→B , then we know that there exists t1 and t3 such that t1[B] = t3[B]. Thus we must choose one of the following for t1 and t3:

-t1 = r1 and t3=r3, or t1= r3 and t3 = r1:

Choosing either t2 = r2 or t2 = r4, t3[C] !=t2[C]

-t1 = r2 and t3 = r4, or t1 = r4 and t3 = r2:

Choosing either t2 =r2 or t2 =r3, t3[C] !=t2[C]

Therefore, the condition t3[C] =t2[C] can not satisfied, so the conjecture is false.