Team 58 : Homework 4

Team members

Vansh Marda

Yatharth Gupta

Shambhavi Jahagirdar

PART 1:

1. R= ABCDEGH and F = {AB->C, AC->B, AD->E, B->D, BC->A, E->G}
    a. ABC
        i. The set of dependencies that hold over ABC is {AB->C, AC->B, BC->A}. This is already a minimal cover.
        ii. This is in BCNF as AB, AC, and BC are all candidate keys of the set ABC.
        iii. It is already in BCNF form.
    b. ABCD
        i. The set of dependencies that hold over ABCD is {AB->C, AC->B, B->D, BC->A}. This is already a minimal cover.
        ii. The keys are AB, AC, and BC. The FD B->D violates 2NF as B is a proper subset of the key AB. So, this set is in 1NF form.
        iii. Decompose the set into ABC, BD and it's now in BCNF form.

    c. ABCEG
        i. The set of dependencies that hold over ABCEG is {AB->C, AC->B, BC->A, E->G}. This is already in minimal cover.
        ii. The keys are ABE, ACE, and BCE. The FD E->G violates 2NF form as E is a proper subset of a key. So, this set is in 1NF form.

iii. Decompose the set into ABC, ABE, EG and it's now in BCNF form.

d. DCEGH

i. The set of dependencies that hold over DCEGH is {E->G}. This is already in minimal cover .

ii. The key is DCEH. The FD E->G violates 2NF form as E is a proper subset of the key DCEH. So, this set is in 1NF form.

iii. Decompose the set into DCEH, EG and it's now in BCNF form.

e. ACEH

i.The set of dependencies that hold over the set ACEH is {}. No FDs exist. This is already in minimal cover.

ii. The key is ACEH. It is in BCNF form.

iii. It is already in BCNF form.

2. R = ABCDEG, F = {AB->C, AC->B, AD->E, B->D, BC->A, E->G}.

a. {AB, BC, ABDE, EG}.

Consider the following instance of R:

{(a1, b, c1, d1, e1, g1), (a2, b, c2, d2, e2, g2)}

Because of the FDs BC->A and AB->C, a1 = a2 only when c1 = c2.

So, if we do AB join BC, we get

{(a1, b, c1), (a1, b, c2), (a2, b, c1), (a2, b, c2)}

So the join on AB, BE, ABDE, EG will contain at least 4 tuples.

Thus, we have a lossy join.

Also, the FDs AB->C and AC->B are not preserved.

Thus, this decomposition is not dependency preserving either.

b. {ABC, ACDE, ADG}

First, join ABC and ACDE.

This is lossless because their intersection AC is a key for ABCDEG.

Now, join (ABC join ACDE) with ADG.

This is also lossless because their intersection is AD and AD->ADG.

Thus, this decomposition is lossless.

The projection of FDs of R onto ABC gives-

AB->C, AC->B, BC->A

The projection of FDs of R into ACDE gives-

AD->E

The projection of FDs of R onto ADG gives-

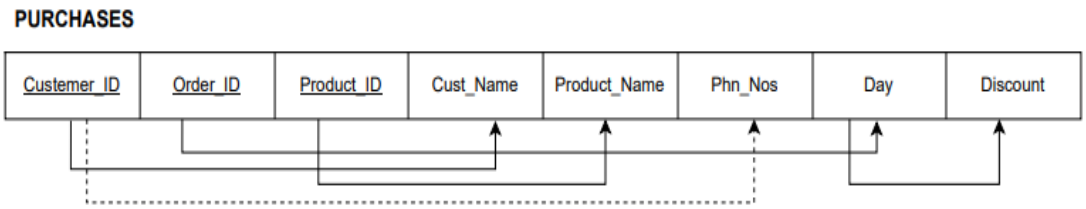
AD->G by transitivity

The closure of this set of dependencies doesn't contain E->G and B->D.

Thus, this decomposition is not dependency preserving.

Part 2:

ASSUMING ONLY 2 PH_NO IN MULTIVALUED ATTRIBUTE Phn_no which can be extended for n



a) 1 NF

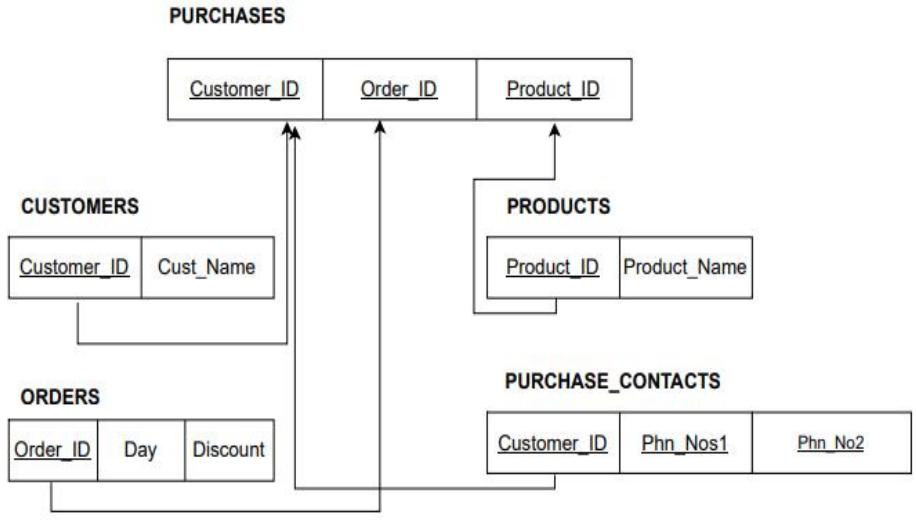
PURCHASES

Customer_ID	Order_ID	Product_ID	Cust_Name	Product_Name	Day	Discount	

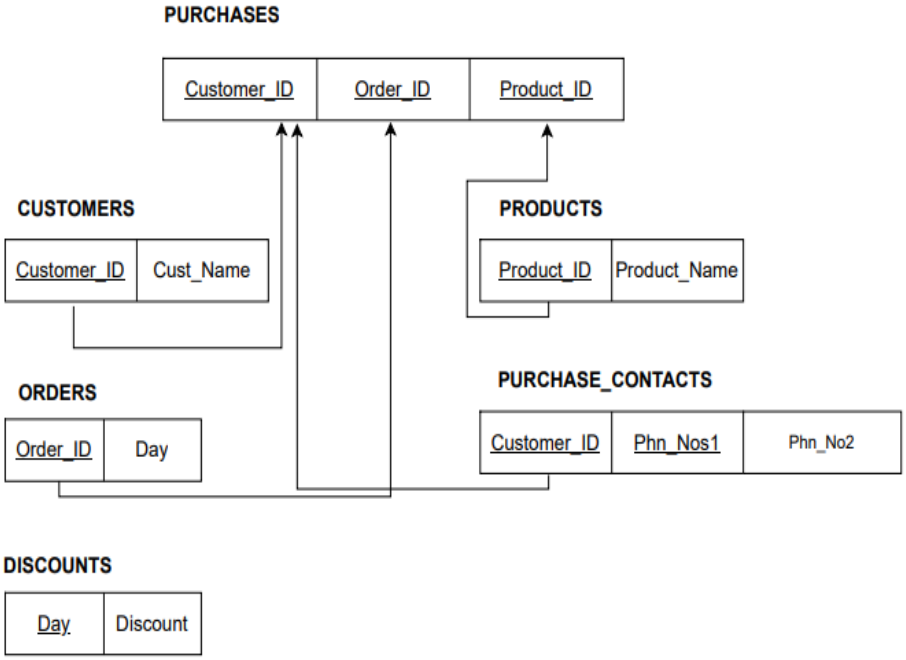
PURCHASE_CONTACTS

Customer_ID	Phn_Nos1	Phn_No2

b) 2 NF



c) 3NF


PURCHASES
Customer_ID
Order_ID
Product_ID
CUSTOMERS
Customer_ID
Cust_Name
PRODUCTS
Product_ID
Product_Name
ORDERS
Order_ID
Day
PURCHASE_CONTACTS
Customer_ID
Phn_Nos1
Phn_No2
DISCOUNTS
Day
Discount