Data & Applications

HW4

GROUP:- 4

Team Members:

1. ***Malla Sailesh (2021101106)***
2. ***Manohar Naga(2021101128)***
3. ***Naveen Chekkapalli (2021101025)***

1. Part One:-

(a).ABC

(i).set of FD's possible from F -
AB->C,AC->B,BC->A

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

(ii). Strongest normal form that is not Violated :- BCNF

(iii).Decompose to BCNF:- Already in BCNF

(b).ABCD

(i).set of FD's possible from F -
AB->C,AC->B,B->D,BC->A

Minimal cover - AB->C,AC->B,B->D,BC->A

Candidate Keys:- AB,AC,BC
(ii). Strongest normal form that is not
Violated :- 1NF
D is not fully functionally dependent B->D holds
and B is a proper subset of AB - violating 2NF
(iii).Decompose to BCNF:- {ABC,BD}

(c).ABCEG
(i).set of FD's possible from F -
AB->C,AC->B,BC->A,E->G
Minimal cover - AB->C,AC->B,BC->A,E->G
Candidate Keys:- ABE,ACE,BCE
(ii). Strongest normal form that is not
Violated :- 1NF
As G is dependent on a proper subset of ABE (E->G)
- violating 2NF
(iii).Decompose to BCNF:- {ABC,EG}

(d).DCEGH
(i).set of FD's possible from F -
E->G
Minimal cover - E->G

Candidate Keys:- DCEGH
(ii). Strongest normal form that is not
Violated :- 1NF
As G is dependent on a proper subset of DCEGH
(E->G) - violating 2NF
(iii).Decompose to BCNF:- {DCEH,EG}

(d).ACEH
(i).set of FD's possible from F - no FD's
Minimal cover - no FD's
Candidate Keys:- ACEH(trivial FD)
(ii). Strongest normal form that is not
Violated :- BCNF
X->X is always in BCNF
(iii).Decompose to BCNF:- {ACEH}

2.
(a).{AB, BC, ABDE, EG }:-
Let the relations for these be **R1,R2,R3,R4**
And their FD sets be **f1,f2,f3,f4**
(a).Dependency-preserving - NO,as the FD's
AB->C,BC->A,AC->B are not there in
(f1 U f2 U f3 U f4) and

(f1 U f2 U f3 U f4) != F

(b).lossless-join -

No,

Consider the relation state

r = {(a1,b,c1,d,e,g) , (a2,b,c2,d,e,g)}

AB->C & BC->A implies that **a1 = a2 iff c1 = c2**

for the natural join of AB,BC we have – AB*BC with entries of type (a1,b,c1),(a2,b,c1),(a1,b,c2),(a2,b,c2)

Thus in this case the natural join of all the decompositions(R1,R2,R3,R4) is not equal to the original relation state.

(b).{ABC, ACDE, ADG }:-

Let the relations for these be **R1,R2,R3**

And their FD sets be **f1,f2,f3**

(a).Dependency-preserving - NO, as B->D,E->G are not there in (f1 U f2 U f3) and

(f1 U f2 U f3) != F

(b).lossless-join - Yes,as the natural join of all the decompositions(R1,R2,R3)

is equal to the original relation.

2. Part Two:-

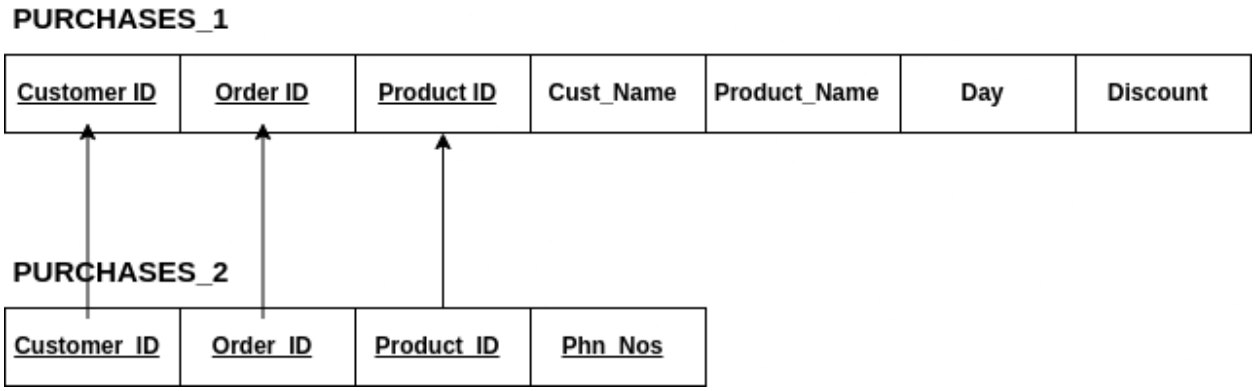
``Arrows represent Foreign Keys``

(a) 1 NF :

As Phn_Nos is a multi-valued attribute we make another relation with
primary key as
(Customer_ID,Order_ID,Product_ID,Phn_Nos)

Relations we get after converting to 1NF:-

1.NF

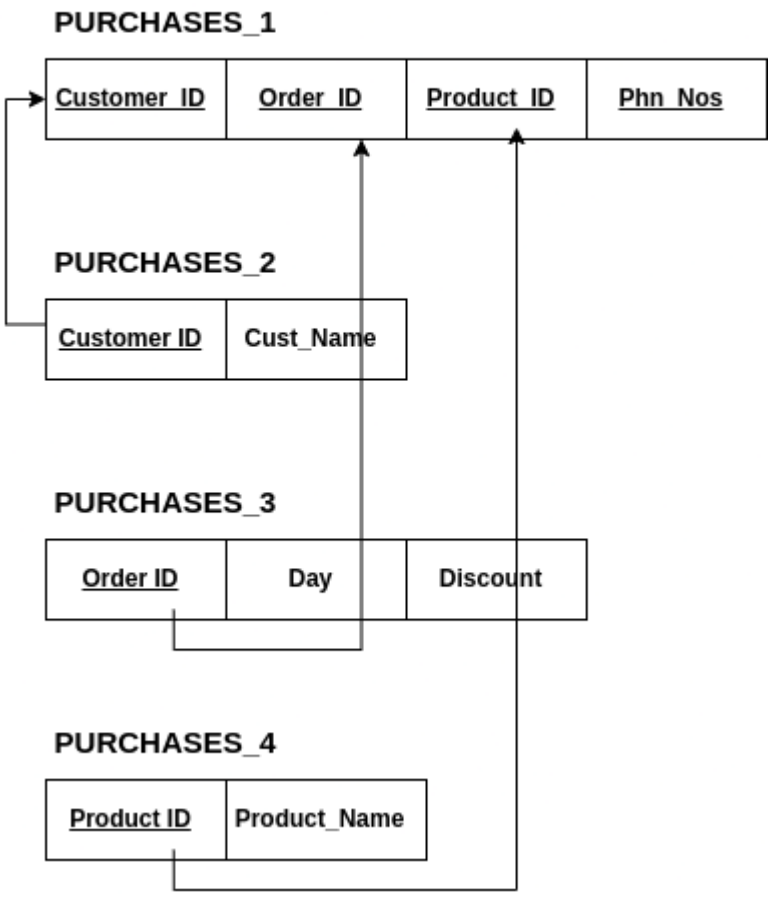


(b) 2 NF :

As there are 3
non-prime attributes

(Cust_Name,Product_Name,Day)
that are not fully functionally dependent on primary key we make 3 new relations for them from 1 NF -
PURCHASES_2,PURCHASES_3,PURCHASES_4
Here we get an extra table (Customer_ID,Order_ID,Product_ID) while making 2NF. But to minimize redundancy we don't consider it as we already have a table PURCHASES_1 with all these attributes as shown below

2.NF



(c) 3 NF :

The FD

(Customer_ID,Order_ID,Product_ID)->Discount

- is transitive as we have the following 2 FDs

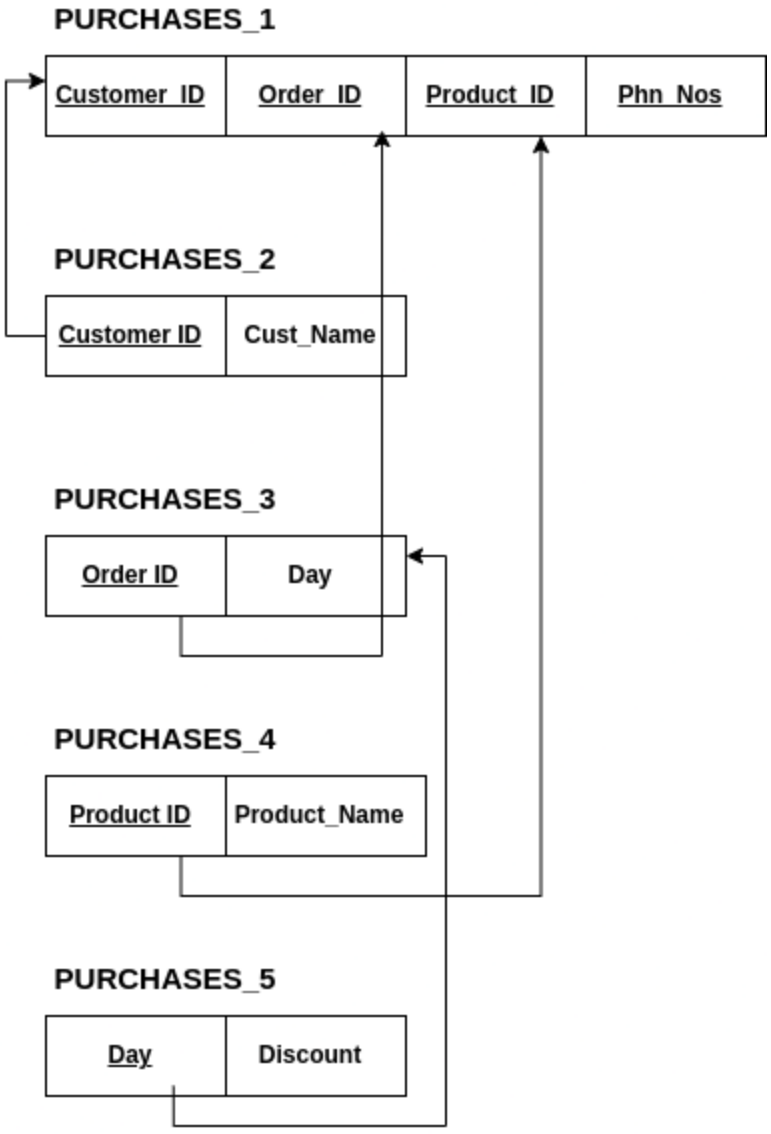
(Customer_ID,Order_ID,Product_ID)->Day

Day->Discount

As a non-key attribute **Day** is determining another non-key attribute **Discount** this violates 3NF.

FD we update the relations of 2NF above so that the **Discount** can be determined by the **Primary key Day in a new relation.**

3.NF


PURCHASES_1
Customer_ID
Order_ID
Product_ID
Phn_Nos
PURCHASES_2
Customer ID
Cust_Name
PURCHASES_3
Order ID
Day
PURCHASES_4
Product ID
Product_Name
PURCHASES_5
Day
Discount