Data and Applications

Homework 4

Abhinav Reddy Boddu	Gowlapalli Rohit	Gnana Prakash Punnavajhala
2021101034	2021101113	2021111027

1. Part One

1. Q1:

A minimal cover of a set of functional dependencies (FD) E is a minimal set of dependencies F that is equivalent to E.

A set FD is minimal if it satisfies the following conditions:
1. Every dependency in F has a single attribute for its right hand-side
2. We cannot replace any dependency X -> A in F with a dependency Y -> A, where Y is a proper subset of X, and still have a set of dependencies that is equivalent to F
3. We cannot remove any dependency from F and still have a set of dependencies that are equivalent to F

A relation is said to be in BCNF in DBMS if the relation is already in 3NF and also , for every functional dependency (say X -> Y), X is either the super key or candidate key.

(a) ABC:

i. The FD’s are:
   - AB → C
   - AC → B
   - BC → A

   S = {AB → C, AC → B, BC → A} is the minimal cover and is also the set of dependencies that hold over the set

ii. The strongest normal form that is not violated by these set of attributes: **BCNF**
   as AB, AC and BC are candidate keys for R1 where R1 = ABC

iii. R1 is in BCNF where R1 = ABC

(b) ABCD:

i. The FD's are:
   - AB → C
   - AC → B
   - B → D
   - BC → A

   S = {AB → C, AC → B, B → D, BC → A} is the minimal cover and is also the set of dependencies that hold over the set

ii. The strongest normal form that is not violated by the relation containing these attributes: **1NF**.
   Here AB, BC, AC are keys, therefore....
   the Functional dependency B → D violates 2NF and thereby **violates BCNF** as B is a proper subset of AB.

iii. Decomposing to: ABC, BD
   This is a BCNF decomposition
   ABC is in BCNF as stated in (a)
   BD is in BCNF as the Functional Dependency B -> D is in accordance with regulations of BCNF in the relation BD

(c) ABCEG:

i. The FD's are:
   - AB → C
   - AC → B
   - BC → A
   - E → G

   S = {AB → C, AC → B, BC → A, E → G} is the minimal cover and is also the set of dependencies that hold over the set.

ii. The strongest normal form that is not violated by the relation containing these attributes: **1NF**.
   Here ABE, BCE, ACE are keys, therefore....
   the Functional dependency E → G violates 2NF and thereby **violates BCNF** as E is a proper subset of ABE.

iii. Decompose to: ABE, ABC, EG
   This is a BCNF decomposition.
   ABC is in BCNF as stated in (a)
   ABE is in BCNF as it doesn't violate regulations of BCNF
   EG is in BCNF as the Functional Dependency E -> G is in accordance with the regulations of BCNF

(d) DCEGH:

i. The FD's are:
   - E → G

   S = {E → G} is the minimal cover and is also the set of dependencies that hold over the set.

ii. The strongest normal form that is not violated by the relation containing these attributes: **1NF**
   Here DCEH is key, therefore...
   the Functional dependency E → G violates 2NF and thereby **violates BCNF** as E is a proper subset of DCEH.

iii. Decompose to: DCEH, EG
   This is a BCNF decomposition.
   DCEH is in BCNF as it doesn't violate any regulations of BCNF
   EG is in BCNF as the Functional Dependency E -> G is in accordance with the regulations of BCNF

(e) ACEH:

i. No Functional Dependencies exist.

   S = {} is the minimal cover and is also the set of dependencies that hold over the set.

ii. The strongest normal form that is not violated by the relation containing these attributes: **BCNF**
   Here ACEH is the key

iii. Already in BCNF

2. # Q2:

The Closure of Functional Dependency means the complete set of all possible attributes that can be functionally derived from given functional dependency.
If "F" is a functional dependency, then closure of functional dependency can be denoted using "$\{F\}^+$"

A Decomposition D = {R1, R2, R3…. Rn} of R is dependency preserving with respect to a set F of Functional dependency if
$(F1 \cup F2 \cup \ldots \cup Fm) = \{F^+\}$

A decomposition D = {R1, R2, R3…. Rn} of a set R of functional dependencies is said to be lossless join if:
$\{R1 \cup R2 \cup \ldots Rn\} = F$
$\{Ri \cap Rj\} \neq \emptyset$ for all $1 \leq i, j \leq n$
$\{Ri \cap Rj\}$ must be key of at least one of Ri, Rj

a) The decomposition D = {AB, ABDE, EG, BC} is not dependency preserving as the FD AC -> B and AB -> C are not preserved
The decomposition D = {AB, ABDE, EG, BC} is not a lossless decomposition of R with set of dependencies given by F.
Now consider the following instances of R
{(a2, b, c2, d2, e2, g2), (a1, b, c1, d1, e1, g1)

Due to the functional dependencies in F given by AB->C and BC->A, c1≠c2 iff a1≠a2.
Hence the join AB |><| BC contain the tuples
{(a1, b, c1), (a2, b, c1), (a2, b, c2), (a1, b, c2)}
Hence the join of AB, EG, BC, ABDE contains >=4 tuples (8 in reality).
Hence D is a lossy decomposition.

b) The decomposition D = {ABC, ACDE, ADG} is lossless Decomposition.
Proof: The join of ADG, ACDE and ABC can be constructed in 2 steps
Step-1: We construct the join of ABC and ACDE.
This is lossless decomposition because intersection of their attributes is given by AC which is a key for ABCDE and ABCDEG. Hence this decomposition is lossless decomposition.
Step-2: Now we join this intermediate join with ADG.
This is lossless decomposition because the intersection of their attributes is given by AD and AD -> ADG.
Hence from Step-1 and Step 2, D is a lossless decomposition
The decomposition D = {ABC, ACDE, ADG} is not dependency preserving as the FD B -> D and E -> G are not preserved.

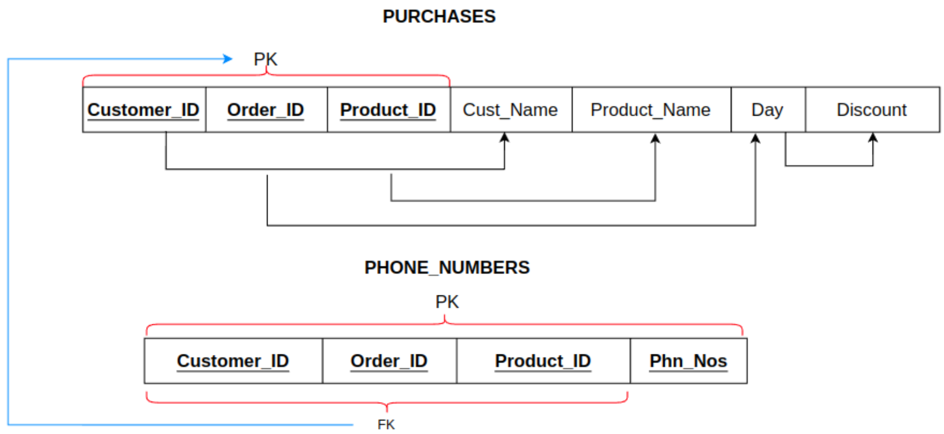
2. Part Two

Note: Blue arrows in the Relational diagrams denote foreign key constraints and Black arrows show functional dependencies

(a) 1NF:

i. The composite key attribute Customer_ID + Order_ID + Product_ID has been broken down into the corresponding simple attributes – Customer_ID, Order_ID and Product_ID

ii. A new relation "PHONE_NUMBERS" was created with {Customer_ID, Order_ID, Product_ID} as foreign key referencing {Customer_ID, Order_ID, Product_ID} of "PURCHASES" and Phn_Nos was added as a simple attribute to remove the multivalued attribute Phn_Nos of the original "PURCHASES" and {Customer_ID, Order_ID, Product_ID, Phn_Nos} was set as the primary key attribute of the relation "PHONE_NUMBERS"

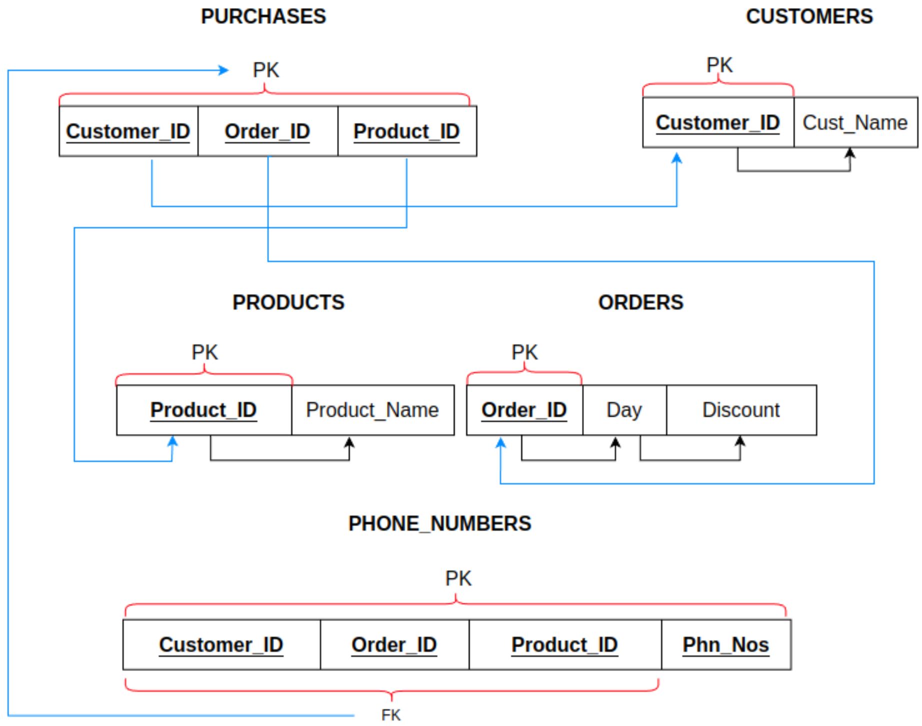


(b) 2NF:

i. Since there are partial functional dependencies involving Customer_ID, Order_ID and ProductID individually as shown in the above 1NF representation where {Customer_ID, Order_ID, Product_ID} was the primary key, they violated the guidelines of 2NF.

ii. Thus, new tables "CUSTOMERS", "PRODUCTS" and "ORDERS" were added as follows:

- "CUSTOMERS" table was created with Customer_ID as primary key and Cust_Name as a simple attribute. The Customer_ID attribute of "PURCHASES" is set to be a foreign key that references Customer_ID of "CUSTOMERS.
- "PRODUCTS" table was created with Product_ID as primary key and Product_Name as a simple attribute. The Product_ID attribute of "PURCHASES" is set to be a foreign key that references Product_ID of "PRODUCTS"
- "ORDERS" table was created with Order_ID as primary key and Day and Discount as simple attributes. The Order_ID attribute of "PURCHASES" is set to be a foreign key that references ORDER_ID of "ORDERS"



(c) 3NF:

i. Since there is a transitive functional dependency from Order_ID to Day and from Day to Discount but Day is a non-prime attribute. This violates the guidelines of 3NF.

ii. Thus, another table “DISCOUNTS” was created with Day as key attribute and functional dependency from Day to Discount in the “DISCOUNTS” table. Also, the “ORDERS” table was updated by removing the Discount column and setting the Day attribute as foreign key referencing Day of “DISCOUNTS”