Assignment 2: Relational Model

Q 1. Comment on following designs (whether the primary keys are appropriate).

(a) SALE (<u>CustomerName</u>, CustomerPhone, CustomerAddress, ItemName, SalePrice, SaleDate)

Answer: No. The primary key, CustomerName is inappropriate because it does not allow the record to be uniquely identified. Two or more customers can have the name and hence, CustomerName cannot be used to describe the row to be unique.

(b) SALE (CustomerName, CustomerPhone, CustomerAddress, <u>ItemName</u>, SalePrice, SaleDate)

Answer: No. The primary key, ItemName is inappropriate because it does not allow the record to be uniquely identified. Two or more customers could purchase the same item and hence, ItemName cannot be used to describe the row to be unique.

However, for the given scenario, it's true because we can uniquely access the record since each row contain different item names.

(c) SALE (CustomerName, CustomerPhone, CustomerAddress, <u>ItemName</u>, SalePrice, <u>SaleDate</u>)

Answer: No, The Primary key i.e (ItemName, SaleDate) is inappropriate because it does not allow the record to be uniquely identified. Item name could be the same for more than one row because there are chances of same items sold more than once. Hence, (ItemName, SaleDate) cannot be used to describe the uniqueness of the row. However, this holds good if we consider the given data where each row has a different item name, but in general it is not valid.

(d) SALE (<u>CustomerName</u>, CustomerPhone, CustomerAddress, <u>ItemName</u>, SalePrice, SaleDate)

Answer: No, The Primary key (CustomerName, ItemName) is inappropriate because it does not allow the record to be uniquely identified. Two or more

customers can have the name and purchase the same item on the same day. Hence, the (CustomerName, ItemName) cannot describe the row to be unique.

Q 2. Add ID columns called CustomerID and SaleID into the SALE table. Answer the following questions. (35 points)

Customer ID	CustomerName	CustomerPhone	CustomerAddress	ItemName	SalePrice	SaleDate	SaleID
1	Anderson	425-125-8461	533 Main street, Chicago, IL, 62803	Antique Desk	3000	14-Dec-12	1
1	Anderson	425-125-8461	533 Main street, Chicago, IL, 62803	Lamp	200	17-Dec-12	2
2	Judy	231-234-1232	112 University Ave., State College, PA, 16802	Chair	1000	2-Dec-12	3
2	Judy	231-234-1232	112 University Ave., State College, PA, 16802	Coffee Table	150	2-Dec-12	4
3	Davis	131-122-9987	17 Green Street, Seattle, WA, 10592	Book Shelf	500	20-Dec-12	5

(a) List one candidate key in this relation.
Answer: After adding SaleID and CustomerID, SaleID is used to describe each record uniquely. Hence, SaleID can be set as the candidate key since will be unique for every sale.

(b) List at least two functional dependencies. What is the determinant in each functional dependency?
Answer:

1. SaleID →(CustomerID, CustomerName, CustomerPhone, CustomerAddress, ItemName, SalePrice, SaleDate).

   Dependent: SaleID

2. CustomerID →(CustomerName, CustomerPhone, CustomerAddress).

Dependent: CustomerID

NOTE: We can also have one more functional dependency where we can access the SalePrice from ItemName and SaleDate,
I.e. (ItemName, SaleDate) → SalePrice
Dependent: (ItemName, SaleDate)

SaleDate is considered along with ItemName because the item price can change frequently. So, given the date, SalePrice can be accessed.

(c) Is the SALE relation well-formed? Why?
Answer: The relationship is not well-formed, because there are two dependents through which the records are being accessed. SaleID is the candidate key whereas CustomerID is not.

(d) If the relation is not well-formed, normalize it to a well-formed relation. In the well-formed relation, please indicate the primary key and foreign key
Answer: Normalization procedures to normalize it into well-formed relations are as follows.

a. Place columns of the functional dependency in a new relation of their own.
CUSTOMER (CustomerID, CustomerName, CustomerPhone, CustomerAddress)

b. Make the determinant of the functional dependency the primary key of the new relation.
CUSTOMER (<u>CustomerID</u>, CustomerName, CustomerPhone, CustomerAddress)

c. Leave a copy of the determinant as a foreign key in the original relation. (italic)
 SALE(<u>SaleID</u>, ItemName, SalePrice, SaleDate, *CustomerID*)

Well-formed Relation model design:
CUSTOMER (<u>CustomerID</u>, CustomerName, CustomerPhone, CustomerAddress)

     SALE(<u>SaleID</u>, ItemName, SalePrice, SaleDate, *CustomerID*)

Q 3. Comment on following designs (whether the primary keys are appropriate).

(a) PURCHASE(ItemName, PurchasePrice, PurchaseDate, <u>VendorName</u>, VendorPhone, VendorAddress)

Answer: No. The primary key, VendorName is inappropriate because it does not allow the record to be uniquely identified. Two or more customers can have the name and hence, VendorName cannot be used to describe the row to be unique.

(b) PURCHASE(<u>ItemName</u>, PurchasePrice, PurchaseDate, <u>VendorName</u>, VendorPhone, VendorAddress)

Answer: No. The primary key, (ItemName, VendorName) is inappropriate because it does not allow the record to be uniquely identified. Same vendor can sell the same items to different customers, hence, cannot be used to describe the row to be unique.

(c) PURCHASE(ItemName, PurchasePrice, PurchaseDate, VendorName, VendorPhone, V<u>endorAddress</u>)

Answer: No. The primary key, (VendorAddress) is inappropriate because it does not allow the record to be uniquely identified. same vendor can sell the multiple items to different users hence, multiple customers can have same vendor address where they have purchased the items. Hence, VendorAddress cannot be used to describe the row to be unique.

(d) PURCHASE(<u>ItemName</u>, PurchasePrice, <u>PurchaseDate</u>, <u>VendorName</u>, VendorPhone, VendorAddress)

Answer: No, The Primary key i.e (ItemName, PurchaseDate, VendorName) is inappropriate because it does not allow the record to be uniquely identified. There are chances that different users can purchase the same item on same date and from same vendor. Hence, the (ItemName, PurchaseDate, VendorName) cannot describe the row to be unique.

Q 4. Add ID columns called VendorID and PurchaseID into the PURCHASE table. Answer the following questions. (35 points)

PurchaseID	ItemName	Purchase Price	PurchaseDate	VendorName	VendorPhone	VendorAddress	VendorID
1	Coffee Table	100	20-Nov-12	European Specialities	206-329-1920	18 Union St., San Francisco, CA, 28192	1
2	Antique Desk	2500	14-Nov-12	European Specialities	206-329-1920	18 Union St., San Francisco, CA, 28192	1
3	Crystal Lights	900	2-Nov-12	Lamp and Lights	231-129-1232	210 Broadway, Dallas, TX, 23212	2
4	Lamp	170	1-Nov-12	Lamp and Lights	231-129-1232	210 Broadway, Dallas, TX, 23212	2
5	Book Shelf	320	24-Oct-12	Antique Things	321-329-1203	190 Alley St., Miami, 96802	3
6	Chair	700	13-Nov-12	European Specialities	206-329-1920	18 Union St., San Francisco, CA, 28192	1

(a) List one candidate key in this relation.
Answer: After adding PurchaseID and VendorID to the PURCHASE table, PurchaseID can be used to describe each record uniquely. Hence, we can set the candidate key as PurchaseID since PurchaseID will be unique for every purchase done.

(b) List at least two functional dependencies. What is the determinant in each functional dependency?
Answer: Following are two functional dependency which I conclude:

1. PurchaseID →(ItemName, PurchasePrice, PurchaseDate, VendorName, VendorPhone, VendorAddress, VendorID).

   Dependent: PurchaseID

2. VendorID →(VendorName, VendorPhone, VendorAddress).

Dependent: VendorID

NOTE: We can also have one more functional dependency where we can access the PurchasePrice from ItemName and PurchaseDate,
I.e. (ItemName, PurchaseDate) → PurchasePrice
    Dependent: (ItemName, PurchaseDate)

PurchaseDate is considered along with ItemName because the Purchase price can change frequently. So, given the date PurchasePrice can be accessed.
.
(c) Is the SALE relation well-formed? Why?
Answer: In the above case, there are two dependents which are being used for accessing the records. But PurchaseID is the candidate key whereas VendorID is not. Hence, the relationship is not well-formed.

(d) If the relation is not well-formed, normalize it to a well-formed relation. In the well-formed relation, please indicate the primary key and foreign key
Answer: Normalization procedures to normalize it into well-formed relations are as followed.

a. Place the columns of the functional dependency in a new relation of their own.
    VENDOR(VendorID, VendorName, VendorPhone, VendorAddress).

b. Make the determinant of the functional dependency the primary key of the new relation.
    VENDOR(<u>VendorID</u>, VendorName, VendorPhone, VendorAddress).

c. Leave a copy of the determinant as a foreign key in the original relation. (italic)
    PURCHASE(<u>PurchaseID</u>,   ItemName,   PurchasePrice,   PurchaseDate,   *VendorID*)

Well-formed Relation model design:
    VENDOR(<u>VendorID</u>, VendorName, VendorPhone, VendorAddress).

PURCHASE(<u>PurchaseID</u>,   ItemName,   PurchasePrice,   PurchaseDate, *VendorID*)

Q 5. You may realize that items in SALE and PURCHASE tables are not connected. For example, for the lamp sold to customer Anderson, we did not keep the information of where this lamp was purchased. Add an additional ITEM table to remedy this problem. How would you design ITEM table and how would you modify the tables you designed in Question 2d and 4d accordingly? Discuss how you keep track of the items in both SALE and PURCHASE in this design. (10 points)

Answer:
To connect two table, ITEM table is introduced. The ITEM table has ItemID as Primary Key along with ItemName. ItemName is removed from both PURCHASE and SALE table and replaced it with ItemID, making it Foreign Key in both tables. By doing so, both the tables can be connected and we can query item purchased from vendor from sale table.

CUSTOMER (<u>CustomerID</u>, CustomerName, CustomerPhone, CustomerAddress)
SALE(<u>SaleID</u>, SalePrice, SaleDate, *CustomerID, ItemID*)
VENDOR(<u>VendorID</u>, VendorName, VendorPhone, VendorAddress)
PURCHASE(<u>PurchaseID</u>, PurchasePrice, PurchaseDate, *VendorID, ItemID*)
ITEM(<u>ItemID</u>, ItemName)

But there might be one scenerio, that can cause ambiguity. If the same item is purchased from different vendors and to track the quantity we introduce a batch table that will maintain purchase and sale stock quantity and maintain separate batches using the BatchId.

CUSTOMER (<u>CustomerID</u>, CustomerName, CustomerPhone, CustomerAddress)
SALE(<u>SaleID</u>, SalePrice, SaleDate, *CustomerID, ItemID*)
VENDOR(<u>VendorID</u>, VendorName, VendorPhone, VendorAddress)
PURCHASE(<u>PurchaseID</u>, PurchasePrice, PurchaseDate, *VendorID, ItemID*)
ITEM(<u>ItemID</u>, ItemName)
BATCH(<u>BatchID</u>, *VendorID, ItemID*, Quantity)

BATCH table is used because whenever there is a purchase of the same item from a different vendor, there will be an increase in Quantity which can be tracked.