CS2055 HW7 Group Homework Submission

Group Member: Fangzheng Guo (fag24@pitt.edu), Shibo Xing (shx26@pitt.edu), Zhuolun Li (zhl137@pitt.edu)

1.

(a): Transform dependencies to canonical forms and drop extraneous attributes:

~~ABE->ED:~~ AB->D
B->D
B->E: ~~AB->E~~
C->D
C->E
C->F
DC->A:
DF->A:
E->D

Get the minimum canonical cover

B->E, E->D implies B->D, thus B->D is redundant

C->E, E->D, implies C->D, thus C->D is redundant

C->F, DF->A, implies DF->A, thus DC->A is redundant

Thus, the minimum canonical cover is:

B->E
C->F
C->E
E->D
DF->A

Since B and C do not appear on the right-hand side, B and C are the primary keys.

Group some dependencies with the same determinant: C->EF

Construct relation for each group**:**

R1: (B,E), R2: (C,E,F), R3: (E,D), R4(DF,A), R5(B,C) (B,C is the superkey)

All of these relations are in BCNF (also 3NF).

(b): Construct the table

	A	B	C	D	E	F
R1: (B,E)	U	K	U	U	K	U
R2: (C,E,F)	U	U	K	U	K	K
R3: (E,D)	U	U	U	K	K	U
R4: (DF,A)	K	U	U	K	U	K
R5: (B,C)	U	K	K	U	U	U

Enforce FDs, we get

	A	B	C	D	E	F
R1: (B,E)	U	K	U	K	K	U
R2: (C,E,F)	K	U	K	K	K	K
R3: (E,D)	U	U	U	K	K	U
R4: (DF,A)	K	U	U	K	U	K
R5: (B,C)	K	K	K	K	K	K

Since all columns at row 5 are known, the relation decomposition is lossless.

2. The set of relations we have at the beginning is simply the universal relation itself: R={R1= (MedicineID, Ingredients, Uses, Warnings, Directions, OrderID, OrderDate, PatientID, TotalPrice, Address, City, State, ZipCode, PhoneNumber, MedicineQuantity)}

FD1: MedicineID→Ingredients, Uses, Warnings, Directions
FD2: OrderID→OrderDate, PatientID, TotalPrice
FD3: PatientID→Address, City, State, ZipCode, PhoneNumber
FD4: MedicineID, OrderID→MedicineQuantity

Decomposition:

R1 is not BCNF due to FD1
R2=(MedicineID, OrderID, OrderDate, PatientID, TotalPrice, Address, City, State, ZipCode, PhoneNumber, MedicineQuantity)
R3=(MedicineID, Ingredients, Uses, Warnings, Directions)

R= (R-R1) ∪ R2 ∪ R3 -> R={R2,R3}
R2 is not BCNF due to FD3

R4=(MedicineID, OrderID, OrderDate, PatientID, TotalPrice, MedicineQuantity)
R5=(PatientID, Address, City, State, ZipCode, PhoneNumber)

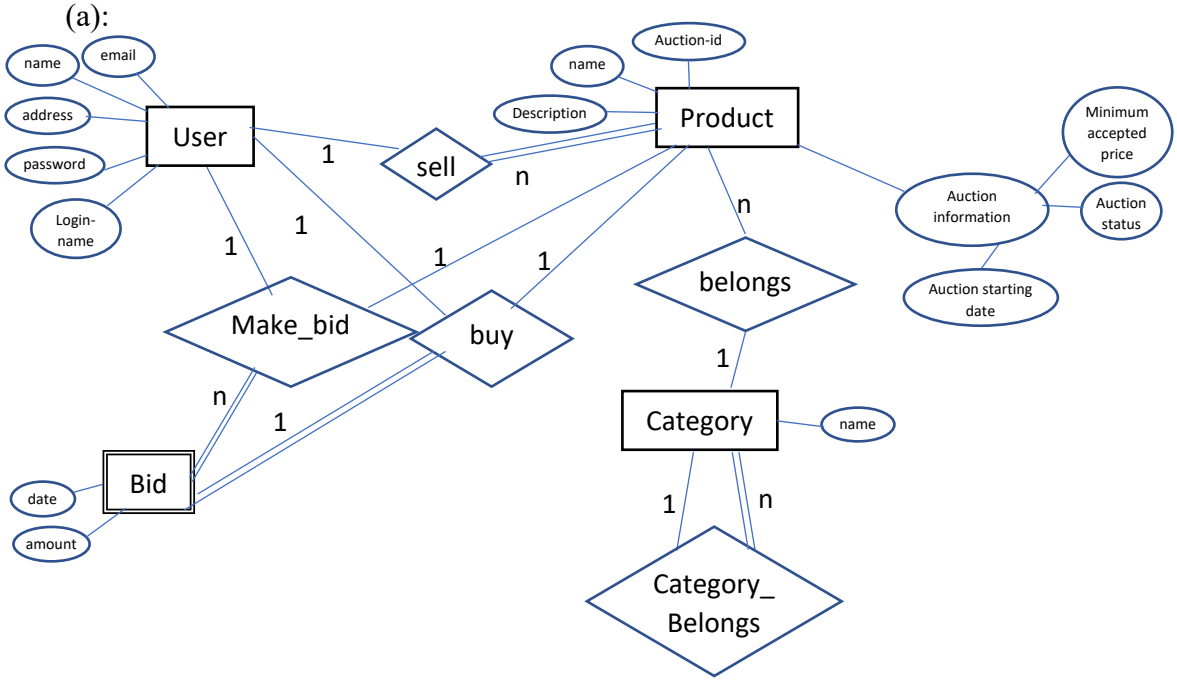
R= (R-R2) ∪ R4 ∪ R5 -> R={R3,R4,R5}
R4 is not BCNF due to FD2

R6=(MedicineID, OrderID, MedicineQuantity)
R7=(OrderID, OrderDate, PatientID, TotalPrice)
R=(R-R4) ∪ R6 ∪ R7 -> R={R3,R5,R6,R7}

Final result:

R3=(MedicineID, Ingredients, Uses, Warnings, Directions)
R5=(PatientID, Address, City, State, ZipCode, PhoneNumber)
R6=(MedicineID, OrderID, MedicineQuantity)
R7=(OrderID, OrderDate, PatientID, TotalPrice)

3. For each registered user, record his/her name, address, email, a unique login-name and pass-word.
•For each product put for auction, record its name, an (optional) description, one or several categories that it belongs to (e.g., 'books-and-records', 'software', 'automobiles', 'appliances',etc). Each product should have a unique auction-id.
•Keep track of information about a product for auction such as who is selling it, the minimum acceptable price, auction starting date and its status (i.e., 'under auction', 'sold', 'withdrawn').
•Keep track of every bid made by registered users, such as the bidder's name, the date when the bid was made, and the amount of the bid, etc
•If a product was sold successfully, we want to know who bought the product with what bidding price, and when it was sold.
•For each product category, record its (unique) name. We want to organize the categories into a hierarchical structure such that one category can contain 0 or more subcategories.
(a):



(b):

Create tables for strong and weak entities:

User:(Login-name, name, email, address, password)

Product:(Auction-id, name, description, seller, minimum_accepted_price, auction_status, auction_starting_date)

Category:(name)

Bid:(Login-name, Auction-id, date, amount)

modify tables based on binary relationships:

Product:(Auction-id, name, description, seller) where seller is a foreign key that references User (sell)

Product:(Auction-id, name, description, category_name, seller) where category_name is a foreign key that references Category (belongs)

Product:(Auction-id, name, description, category_name, seller, buyer) where buyer is a foreign key that references User (buy)

Sold:

Category:(name, parent_category_name) where parent_category_name is a foreign that references Category (category belongs)

modify tables for multi-value entity:

Auction_Information:(Auction_id, Minimum_accepted_price, Auction_status, Action_starting_date)

Final result:

User:(Login-name, name, email, address, password)

Bid:(Login-name, Auction-id, date, amount)

Product:(Auction-id, name, description, category_name, seller, buyer)

Category:(name, parent_category_name)

Auction_Information:(Auction_id, Minimum_accepted_price, Auction_status, Action_starting_date)