NYU Tandon School of Engineering Oct. 3, 2019

Computer Science and Engineering

CS6083, Fall 2019

**Problem Set #1 Sample Solution**

**Problem 1:**

**(a)**

No. A seller can sell many items with the same item type, so just seller and iid would not suffice.

**(b)**

Yes. If we remove bidtime from the primary key, a bidder would not be able to bid multiple times during an auction.

**(c)**

I.

select seller, count(\*)

from AUCTION

where year(starttime) = 2018

group by seller

II.

with max\_price(seller, iid, starttime, maxbidprice) as

select seller, iid, starttime, max(bidprice) as maxbidprice

from AUCTION natural join BID

where bidtime <= endtime and bidprice >= minbid

group by seller, iid, starttime

select bidder, seller, iid, maxbidprice

from BID natural join max\_price

where BID.bidprice = max\_price.maxbidprice

III.

select \*

from BID,

(select max(bidprice) as maxbidprice

from BID natural join ITEMTYPE

where itemname = 'IPhone 12')

where bidprice = maxbidprice and itemname = 'IPhone 12'

IV.

*(using the 'max\_price' clause defined in c.II)*

with more\_than\_ten\_bidder(bidder) as

select bidder

from BID

group by bidder

having count(\*) > 10

with ever\_won\_bidder(bidder) as

select distinct bidder

from BID natural join max\_price

where bidprice = maxbidprice and bidtime <= endtime

(select bidder

from more\_than\_ten\_bidder)

except

(select bidder

from ever\_won\_bidder)

V.

*(using the 'max\_price' clause defined in c.II)*

select itemname, condition, avg(maxbidprice)

from AUCTION natural join max\_price natural join ITEMTYPE

where year(starttime) = 2018

group by itemname, condition

VI.

select distinct seller

from BID

where bidder = seller

VII.

with adversary\_pair(bidder1, bidder2, seller, iid, starttime) as

select

adversary1.bidder as bidder1,

adversary2.bidder as bidder2,

adversary1.seller as seller,

adversary1.iid as iid,

adversary1.starttime as starttime

from BID as adversary1, BID as adversary2

where

adversary1.seller = adversary2.seller and

adversary1.iid = adversary2.iid and

adversary1.starttime = adversary2.starttime and

adversary1.bidder < adversary2.bidder

select bidder1, bidder2

from adversary\_pair

where year(starttime) = 2018

group by bidder1, bidder2

having count(distinct seller, iid, starttime) > 10

**(d)**

I.

II.

III.

IV.

*(using the 'maxprice' clause defined in d.II)*

V.

*(using the 'maxprice' clause defined in d.II)*

VI.

VII.

**(e)**

I.

Cannot be expressed with Relational Calculus as it requires aggregate functions.

II.

III.

IV.

In principle, this query could be expressed in DRC/TRC, since it only counts to a constant, ten. But doing so would result in an extremely complicated query, so this is not expected of you. You need aggregate functions to write a query of reasonable length for this.

V.

Cannot be expressed with Relational Calculus as it requires aggregate functions.

**Problem 2:**

(a)

User(**uid**, uname, hometown, email)

Board(**bid**, uid, bname, create\_time)

Picture(**pid**, uid, description, upload\_time)

Pin(**pid**, **bid**)

Follow(**bid**, **uid**)

Comment(**pid**, **bid**, **uid**, content, **comment**\_time)

Liked(**pid**, **bid**, **uid**)

Board.bid is the foreign key referencing to User.uid

Picture.pid is the foreign key referencing to User.uid

Pin.pid is the foreign key referencing to Picture.pid

Pin.bid is the foreign key referencing to Board.bid

Follow.bid is the foreign key referencing to the Board.bid

Follow.uid is the foreign key referencing to the User.uid

Comment.pid is the foreign key referencing to Picture.pid

Comment.bid is the foreign key referencing to Board.bid

Comment.uid is the foreign key referencing to the User.uid

Liked.pid is the foreign key referencing to Picture.pid

Liked.bid is the foreign key referencing to Board.bid

Liked.uid is the foreign key referencing to User.uid

Assumptions: We assume in this design that when a user re-pins a picture, this cannot really be distinguished from an “original” pin. So we do not store from the which a picture gets repinned, only to which boards it was pinned. Also, we assume comments for a picture are associated with the picture in the context of a particular board, not in general.

(b)

I.

**with** Like3Times(u1, u2) **as** (

**select** **Liked**.uid **as** u1, Board.uid **as** u2

**from** **Liked** **join** Board **on** (**Liked**.bid = Board.bid)

**where** **Liked**.uid != Board.uid

**group** **by** **Liked**.uid, Board.uid

**having** **count**(\*) >= 3

)

**select**

**from** **User** U1, **User** U2, Like3Times

**where** U1.uid > U2.uid **and** (U1.uid, U2.uid) **in** Like3Times **and** (U2.uid, U1.uid) **in** Like3Times

II.

**select** pid, **count**(\*)

**from** Picture **left** **join** Pin **join** **on** (Board.bid = Pin.bid)

**where** Board.uid != Picture.uid

**group** **by** pid;

III.

**with** BoardPinedSunsetPicture **as** (

**select** bid

**from** Board, Pin, Picture

**where** description = 'cactus at sunset' **and** Board.bid = Pin.bid **and** Pin.pid = Picture.pid)

**select** uname

**from** **User** **natural** **join** Follow **natural** **join** BoardPinedSunsetPicture;

IV.

**select** uname

**from** **User** **natural** **join** **Comment**

**where** hour(currenttime()-comment\_time) < 24

**group** **by** uid

**having** **count**(\*) > 10