Problem 1

(a) (fnum, deptdate) is a foreign key from Flights to Books.

(pID, fnum, deptdate) is a foreign key in Books table.

(originAp, destinAp) is a foreign key from Airport to Flight

(b) I. select P.pName

from Passenger P, Airport A1, Airport A2, Flight F, Books B

where B.pID=P.pID and A1.apCountry= “Japan” and A1.apCode=F.orginAp and

A2.apCountry= “US” and A2.apCode =F.destinAp and B.fnum=F.fnum

B.deptdate=“11/5/2018” and B.deptdate=F.deptdate

II.select P.pName

From Passenger P, Flight F1 natural join Books B1

Where P.pID= B1.pID and exist (select *

From Flights F2 natural join Books B2

Where B1.pID=B2.pID and F1.fnum <>F2.fnum and

F1.deptdate=F2.deptdate and F1.deptTime-F2.arrTime<=one hour

and F1.deptTime-F2.arrTime>0)

III. select destinAp, count(distinct pID)

From Flight F, Books B

Where originAp= “JFK” and B.year(deptdate)=2017 and F.fnum=B.fnum and

F.deptdate=B.deptdate

Group by destinAp

IV. Select aName

From Airline A, (Select aID, count(pID) as number

From Flight F natural join Books B

Where F.deptdate(year) =2017

Group by aID) Num_book

Where A.aID= Num_book.aID and Num_book.number =(select max(number)

From (Select aID, count(pID) as number

From Flight F natural join Books B

Where F.deptdate(year) =2017

Group by aID) Num_book);

V. Select P1.pName,

From Passenger P1, (select pID, count(*) as total

from Airport A1, Flight F, Books B, Airport A2

where A1.apCode=F.originAp and A1.apCountry= "Japan" and

F.deptdate(year)=2017 and A2.apCode=F.destinAp and A2.apCountry ="US" and F.fnum=B.fnum and F.deptdate=B.deptdate

Group by pID) P2

Where P1.pID=P2.pID and P2.total>1

VI. select P1.pID, P2.pID

From Passenger P1, Passenger P2

Where P1.pID>P2.pID and not exists(select *

From Books B1 natural join P1

Where not exists(select *

From Books B2 natural join P2

Where B1.fnum=B2.fnum and B1.deptdate =B2.deptdate))

VII select distinct destinAp

From Flight

Where originAp= “JFK”

Union

Select distinct F2.destinAp

From Flight F1, Flight F2

Where F1.originAp= “JFK” and F1.destinAp=F2.originAp

(3)I. $\Pi_{pName}(\Pi_{fnum,deptdate}((\sigma_{apCountry=US \wedge destinAp=apCode}(\sigma_{apCountry=Japan \wedge originAp=apCode}(Airport \times Flight)) \times Airport) \bowtie Books) \bowtie passengers)$

II. $D \leftarrow \rho_{newtable(pID,f,deptdate,dT,aT)}(\Pi_{pID,fnum,deptdate,deptTime,arrTime}(Flight \bowtie Books \bowtie Passengers)$

$\Pi_{pName}(\sigma_{fnum \neq f \wedge deptTime-aT>0 \wedge deptTime-aT \leq one\ hour}(D \bowtie Flight \bowtie Books \bowtie Passengers))$

III. $_{destinAp}\mathcal{G}_{count(distinct\ pID)}(\sigma_{originAp=\text{JFK} \wedge year(deptdate)=2017}(Flight \bowtie Books))$

IV. $D \leftarrow {}_{aID}\mathcal{G}_{count(pID)\ as\ num}(\sigma_{year(deptdate)=2017}(Flights \bowtie Books))$

$E \leftarrow \mathcal{G}_{\max(num) as\ m} D$

$\Pi_{aName}(\sigma_{D.num=E.m}(D \bowtie Airline))$

V. $D \leftarrow \Pi_{fnum,deptdate,destinAp}($

$Flight \bowtie_{originAp=apCode \wedge apCountry=\text{Japan} \wedge deptdate=2017} Airline)$

$E \leftarrow \Pi_{fnum,deptdate}(D \bowtie_{destinAp=apCode \wedge apCountry="US"}$Airline)

$F \leftarrow {}_{pID}\mathcal{G}_{count(*)\ as\ num}(E \bowtie Books)$

$\Pi_{pName}(\sigma_{num>2}(E \bowtie Books \bowtie Passengers)$

VI.

VII.

$\rho_{F(f,a,dd,dT,aT,oA,dA)}Flight$

D $\leftarrow \Pi_{destinAp}(\sigma_{oA=}JFK\wedge dA=originAP}(F \times Flight)$

$\Pi_{destinAp}(\sigma_{orginAp="JFK"}Flight) \cup D$

(4)I. $\{t|\exists s \in Passenger(t[pName] = s[pName]) \wedge \exists r \in Books(s[pID] = r[pID]) \wedge \exists q \in Airport(q[apCountry] = Japan) \wedge \exists p \in Airport(p[apCountry] = US) \wedge \exists u \in Flight(u[fnum] = r[fnum] \wedge r[deptdate] = u[deptdate] \wedge u[originAp] = q[apCode] \wedge u[destinAp] = p[apCode])\}$

II. $\{t|\ \exists s \in Passenger(r[pName] = s[pName]) \wedge \exists\ r \in Books\ (r[pID] = s[pID]) \wedge \exists\ p \in Flight\ (p[fnum] = r[fnum] \wedge p[deptdate] = r[deptdate]) \wedge \exists q \in Books\ (q[pID] = r[pID] \wedge q[fnum] \neq p[fnum] \wedge p[deptdate] = q[deptdate]) \wedge \exists w \in Flight(w[fnum] = q[fnum] \wedge w[deptdata] = q[deptdate] \wedge p[deptTime] - w[arriTime] \leq one\ hour \wedge p[deptTime] - w[arriTime] > 0)\}$

III.cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are aggregate functions.

IV. cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are aggregate functions.

V. cannot express in Domain Relational Calculus or Tuple Relational Calculus as there are aggregate functions.

VI. $\{t|\exists p \in Passengers(\ t[p1] = p[pID]) \wedge \exists q \in Passenger(t[p2] = q[pID] \wedge p[pID] > q[pID]) \wedge (\forall s \in Books((s[pID] = p[pID]) \Rightarrow \exists r \in Books(r[pID] = q[pID] \wedge r[fnum] = s[fnum] \wedge r[deptdate] = s[deptdate]))\}$

VII.

$\{t|\exists s \in Flight(s[originAp] =$ JFK $\wedge\ t[destinAp] = s[destinAp]) \vee \exists r \in Flight(r[originAp] = s[destinAp] \wedge t[destinAp] = r[destinAp])\}$

Problem 2

(1) Assuming that each agent can only organize one open house for a certain property in the same day. There exist some agencies who represent a property but does not organize an open house for it yet.

Schema:Property(pID, street number, street name, apartment name, city, zipcode, neighborhood, asking price)

Agents(aID, name, phone number)

Represent(pID, aID)

Organization(pID,aID, date, start time, end time)

Buyers(bID, name, phone number)

Reservation(bID, pID,aID,date)

In Property, pID is primary key.

In Agents, aID is primary key.

In Represent, (pID,aID) is primary key, pID is also a foreign key referencing pID(Property), aID is a foreign key referencing aID(Agents).

In Organization,(pID,aID,date) is primary key, pID is also a foreign key referencing pID(Property), aID is a foreign key referencing aID(Agents)

In Buyers, bID is primary key.

In Reservation,(bID,pID,aID,date) is primary key. bID is a foreign key referencing bID(Buyers), pID is also a foreign key referencing pID(Property), aID is a foreign key referencing aID(Agents),date is a foreign key referencing date(Organization).

(2)I. select P.pID

from Property P, Organization O

where P.neighborhood= “Park Slop” and P.pID=O.pID

group by pID

having count(distinct aID)>5

II.select distinct P.pID, P.street number, P.street name

From Property P, Organization O

Where P.pID=R.pID and O.date= “10/10/2018” and P.neighborhood= “Park Slop”

III.select B.name

From Buyers B, Reservation R

Where R.pID=61734 and R.pID=B.pID