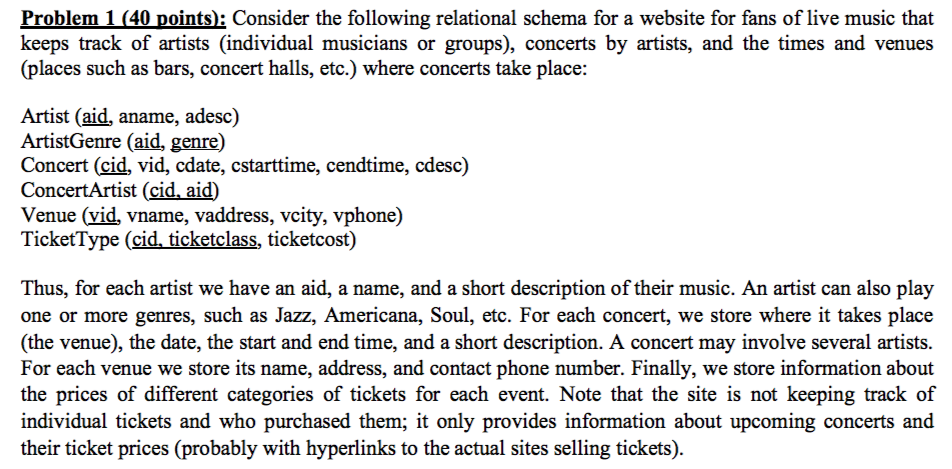
**ASSIGNMENT 1**



1. (2 points) Why does TicketType and ConcertArtist have two primary keys? What would be the problem of using only one primary key? Explain.

Primary key is used to identify a tuple uniquely. The relations TicketType and ConcertArtist require two attributes to be able to identify tuples uniquely. For example, if ConcertArtist

had only aid as primary key then all it wouldn’t be able to identify the different Concert with different cid. Similarly, if we would take only cid as the primary key for TicketType then we would get many items with the same cid thus not uniquely identifying the tuple.

1. (2 points) Identify suitable foreign key representing the tables.

A foreign key is a field (or collection of fields) in one table that refers to the primary key in another table.

In this schema,

* aid is the foreign key in relation ArtistGenre
* cid and aid are the foreign key in relation ConcertArtist
* vid is the foreign key in relation Concert
* cid is the foreign key in relation TicketType

1. (12 points) Write statements in SQL for the following queries.
2. Output the names of the venues in city ‘New York’.

SELECT vname FROM Venue WHERE vcity= “New York”;

1. Output all the concerts that were played in ‘Chicago’ in year 2017.

SELECT cid FROM Concert natural join Venue WHERE vcity=”Chicago” and extract(year from cdate) = 2017;i

1. For each venue, output the number of artists played in year 2017.

SELECT vid, count(aid)

FROM Venue natural join Concert natural join ConcertArtist

WHERE extract(year from cdate) = 2017;

1. Give the cid and ticket class where the cost of ticket exceeded $100.

SELECT cid ,ticketclass FROM TicketType WHERE ticketcost > 100;

1. Output the aids of all artists who have appeared together with “Bruno Mars” in at least two concerts during 2017.

SELECT aid, aname

FROM Artist NARTURAL JOIN ConcertArtist

WHERE aname = "Bruno Mars" AND cid in (

SELECT cid FROM Concert NATURAL JOIN ConcertArtist NATURAL JOIN Artist

WHERE name = "]] Mars" and YEAR(cdate) = 2016)

GROUP BY aid, aname

HAVING COUNT(\*) >= 2

6. Give the names of the artist who have done the concerts at all the venues at least once.

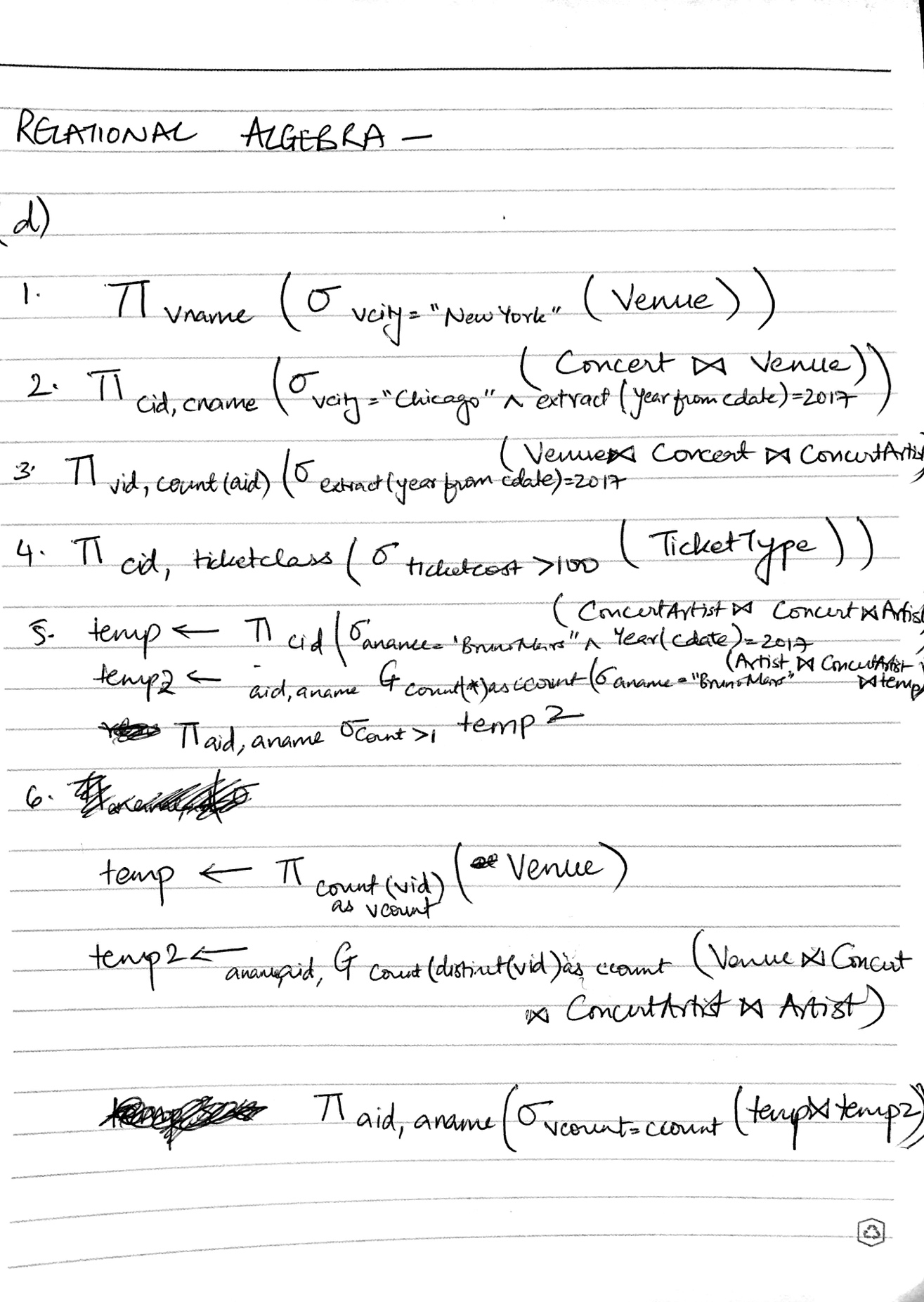
SELECT aid,aname

FROM (Venue natural join Concert natural join ConcertArtist natural join Artist)

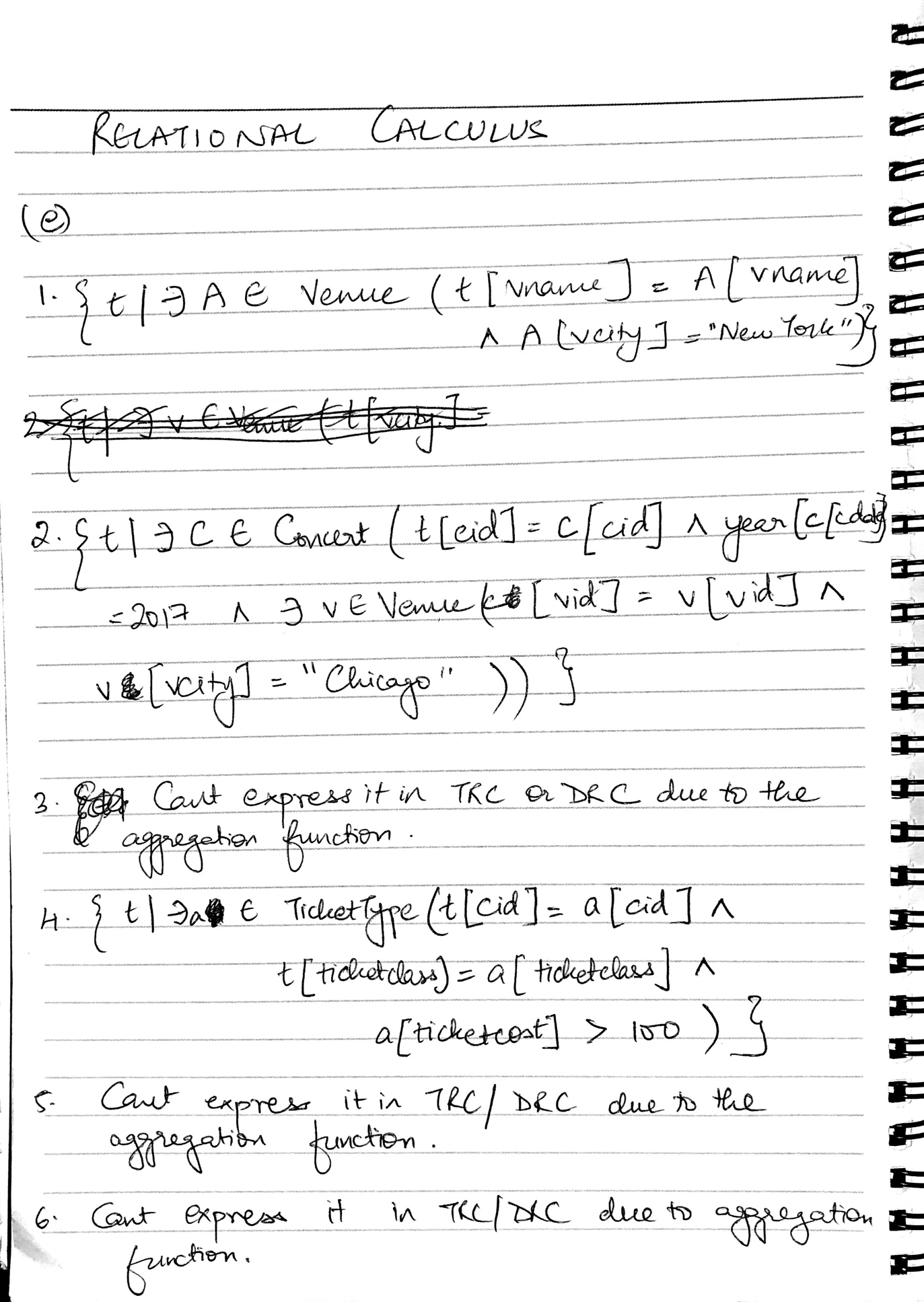
group by aid

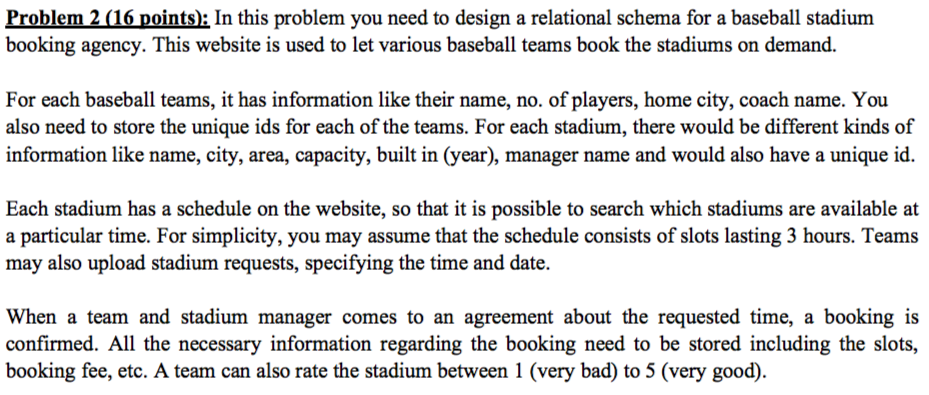
having (count(distinct(vid)) = (select count(vid) from Venue));

1. (12 points) Write relational algebra statements for the above queries if it is possible, or give a reason why is it not possible.



1. (12 points) Write statements in (Domain or Tuple) Relational Calculus for the above queries or explain why it is not possible





(a) (8 points) Design a relational database schema for the given problem, with suitable table, attributes, primary keys, and foreign keys. Discuss any assumptions you are making in your design!

Team(tid, tname, number\_players, tcity, coach\_name)

Stadium(sid, sname, scity,sarea, capacity, built\_year, manager\_name)

StadiumRating(sid, tid,rating, rdate)

Schedule(slotid, sid, start\_time, date, bookingid)

StadiumRequest(reqid, sid, tid, time, date)

Booking(bookingid, tid, sid, reqid, fee)

Primary Key- tid for relation Team, sid for relation Stadium, (sid,tid) for StadiumRating, slotid for Schedule,reqid for StadiumRequest, bookingid for relation Booking.

Foreign Key-

Assumptions: slots are 3 hours long with their start time stored in the relation Schedule. Rating ranges from 1 to 5 in StadiumRating.

(b) (8 points) Write SQL statements for the following queries. If your schema does not support these, you need to modify it appropriately.

1. Output the names of all stadiums available between 9 am to 6 pm on Feb 10th, 2018. (They should be available all the hours, not just part of the time.)

SELECT sid, sname

FROM Stadium

WHERE NOT EXISTS( SELECT sid, sname FROM Stadium natural join Schedule WHERE (start\_time between 7am and 5pm) and date= “Feb 10th, 2018” and bookingid=NOT NULL);

1. For each stadium, output their ids, and the number of distinct team they have been booked to in 2017.

SELECT sid, distinct(tid)

FROM Schedule

WHERE extract (year from date) = 2017

INNER JOIN Booking ON Schedule.bookingid = Booking.bookingid;

1. Output the stadium(s) who earned the most rating overall during 2017.

SELECT max (avg\_rating)

FROM(SELECT sid, avg(rating) as avg\_rating

FROM StadiumRating

WHERE extract (year from rdate) = 2017

INNER JOIN Stadium ON StadiumRating.sid = Stadium.sid

GROUP BY sid);

1. Output the names of teams who have booked at least 5 stadiums, but who have never given a rating of 4 stars or higher.

SELECT tid, avg\_rating

FROM (

SELECT tid, sid, avg(rating) as avg\_rating

FROM Booking natural join StadiumRating

GROUP BY tid

HAVING count(sid) >= 5)

WHERE avg\_rating < 4;