Database Systems
CSE 414

Section 5: Midterm Review

Consider a schema for a picture tagging website:

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the names of all members that were tagged in both 2011 and 2014 sorted in alphabetic order

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the names of all members that were tagged in both 2011 and 2014 sorted in alphabetic order.

Find a partner and try it out!

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the names of all members that were tagged in both 2011 and 2014 sorted in alphabetic order

```
select M.name
from Member M, Tagged T, Picture P1, Picture P2
where M.mid = T.mid
and P1.pid = T.pid and P2.pid = T.pid
and P1.year = 2011 and P2.year = 2014
order by M.name;
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

```
select M.name
from Member M, Tagged T, Picture P1, Picture P2
where M.mid = T.mid
and P1.pid = T.pid and P2.pid = T.pid
and P1.year = 2011 and P2.year = 2014
order by M.name;
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

```
select M.name
from Member M, Tagged T, Picture P1, Picture P2
where M.mid = T.mid
and P1.pid = T.pid and P2.pid = T.pid
and P1.year = 2011 and P2.year = 2014
order by M.name;
```

T.pid = P1.pid and T.pid = P2.pid =>

P1.pid = P2.pid

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

```
select M.name
from Member M, Tagged T, Picture P1, Picture P2
where M.mid = T.mid
and P1.pid = T.pid and P2.pid = T.pid
and P1.year = 2011 and P2.year = 2014
order by M.name;
```

```
T.pid = P1.pid and T.pid = P2.pid =>
P1.pid = P2.pid => P1.year = P2.year
```

Since pid is the primary key of Picture

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the names of all members that were tagged in both 2011 and 2014 sorted in alphabetic order

```
select x.name
from Member x, Tagged y1, Tagged y2,
Picture z1, Picture z2
where x.mid = y1.mid and y1.pid = z1.pid and z1.year = 2011
and x.mid = y2.mid and y2.pid = z2.pid and z2.year = 2014
order by x.name
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

Find a partner and try it yourself!

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q1 = select distinct x.name
     from Member x, Tagged y
     where x.mid = y.mid
     and not exists
          (select *
           from Picture z
           where y.pid = z.pid
           and z.year = 2015);
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q1 = select distinct x.name
     from Member x, Tagged y
     where x.mid = y.mid
     and not exists
          (select *
           from Picture z
           where y.pid = z.pid
           and z.year = 2015);
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q2 = select distinct x.name
     from Member x
     where not exists
          (select *
           from Tagged y, Picture z
           where x.mid = y.mid
           and y.pid = z.pid and z.year = 2015);
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q3 = select distinct x.name
     from Member x
     where not exists
         (select *
          from Tagged y
          where x.mid = y.mid
          and not exists
              (select *
               from Picture z
               where y.pid = z.pid
               and z.year = 2015));
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q3 = select distinct x.name
     from Member x
     where not exists
         (select *
          from Tagged y
          where x.mid = y.mid
          and exists
              (select *
               from Picture z
               where y.pid = z.pid
               and z.year = 2015));
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q4 = select distinct x.name
     from Member x, Tagged y, Picture z
     where x.mid = y.mid and y.pid = z.pid
     and z.year = 2015
     group by x.name
     having count(z.pid) = 0;
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q4 = select distinct x.name
     from Member x, Tagged y, Picture z
     where x.mid = y.mid and y.pid = z.pid
     and z.year = 2015
     group by x.name
     having count(z.pid) = 0;
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Return the name of all users who were never tagged in 2015.

```
Q4 = select distinct x.name
     from Member x
     left outer join Tagged y on x.mid = y.mid
     left outer join Picture z on y.pid = z.pid
     and z.year = 2015
     group by x.name
     having count(z.pid) = 0;
```

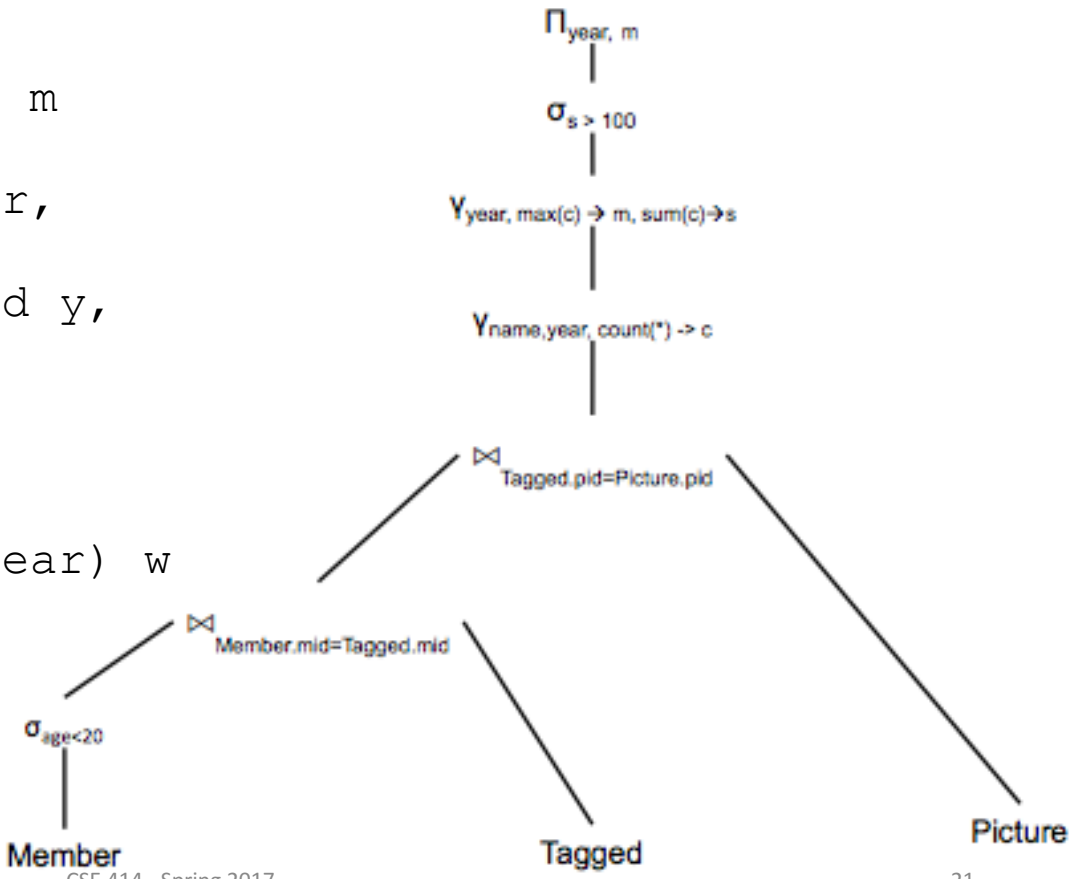
```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Write a Relational Algebra Expression (draw a tree) for the following query:

```
select w.year, max(w.c) as m
from
     (select x.name, z.year, count(*) as c
      from Member x, Tagged y, Picture z
      where x.mid = y.mid and y.pid = z.pid and
      age < 20 group by x.name, z.year) w
group by w.year
having sum(w.c) > 100;
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

```
select w.year, max(w.c) as m
from
      (select x.name, z.year,
       count(*) as c
       from Member x, Tagged y,
       Picture z
       where x.mid = y.mid
       and  y.pid = z.pid
       and  age < 20
       group by x.name, z.year) w
group by w.year
having sum(w.c) > 100;
```



```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Write a query in datalog with negation that returns the mids and names of all members that were tagged only in pictures where Alice was also tagged.

Try it out!

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Write a query in datalog with negation that returns the mids and names of all members that were tagged only in pictures were Alice was also tagged.

Try it out!

Hint:

```
aliceTagged(pid) :-
nonAnswer(mid) :-
answer(mid,name) :-
```

```
Member(mid, name, age)
Picture(pid, year)
Tagged(mid, pid)
```

Write a query in datalog with negation that returns the mids and names of all members that were tagged only in pictures were Alice was also tagged.

```
aliceTagged(pid) :- Member(mid, 'Alice',-),Tagged(mid, pid)
nonAnswer(mid) :- Tagged(mid,pid) not aliceTagged(pid)
answer(mid,name) :- Member(mid,name,-), not nonAnswer(mid)
```