Consider a schema for a picture tagging website:

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

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

Find a partner and try it yourself!
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);
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);
Return the name of all users who were never tagged in 2015.

\[
Q2 = \text{select distinct } x.\text{name} \\
\text{from Member } x \\
\text{where not exists} \\
(\text{select *} \\
\text{from Tagged } y, \text{Picture } z \\
\text{where } x.\text{mid} = y.\text{mid} \\
\text{and } y.\text{pid} = z.\text{pid and } z.\text{year} = 2015); \\
\]
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));
Return the name of all users who were never tagged in 2015.

$$Q3 = \text{select distinct } x\text{.name}$$
$$\text{from Member } x$$
$$\text{where not exists}$$
$$\quad (\text{select *}$$
$$\quad \text{from Tagged } y$$
$$\quad \text{where } x\text{.mid} = y\text{.mid}$$
$$\quad \text{and exists}$$
$$\quad \quad (\text{select *}$$
$$\quad \quad \text{from Picture } z$$
$$\quad \quad \text{where } y\text{.pid} = z\text{.pid}$$
$$\quad \quad \text{and } z\text{.year} = 2015))$$;
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;
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;
Return the name of all users who were never tagged in 2015.

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

Hint:

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

```
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)
```