CSCI 4380 Spring 2018 Quiz 3 [Time: 25 minutes]

Name 1:	RCS ID1:
Name 2:	RCS ID2:
Name 3:	RCS ID3:

Rules.

- Open book and notes. Do not use any electronic tools including your computer.
- You can talk to anyone in class. If you know the answer, help someone else in class.
- Each quiz must be by at least two and at most three people. Most quizzes should be by three people, so find someone to talk to. Put your name on one quiz only.

Question (25 points). The following is a database of heroes in movies and TV shows. For shows that are still ongoing, the endyear is null. All other attributes must have a value. Universe for shows, movies and heroes is either Marvel or DC. You can shorten `heroesintvshow` to `ht` and `heroesinmovie` to `hm` in your queries.

```
TVShows(id, name, tvchannel, startyear, endyear, universe)
Movies(id, name, year, gross, universe)
Heroes(id, name, realname, universe)
Actors(id, name, dob)
HeroesInTVShow(id, hero_id, show_id, actor_id)
HeroesInMovie(id, hero_id, show_id, actor_id)
```
Write the following queries in SQL. Use the simplest possible query for each.

Query 1. Find all actors who played the same hero both in a TV show and at least two movies. Return the id, name of the actor, sort by actor name.

Answer.

```
select distinct --distinct: because there may be multiple heroes for this actor
    a.id
    , a.name
from
    actors a
    , heroesintvshow ht
    , heroesinmovie hm
where
    a.id = ht.actor_id
    and a.id = hm.actor_id
    and hm.hero_id = ht.hero_id
group by
    a.id
    , ht.hero_id
having
    count(distinct hm.show_id)>=2
order by
    a.name;
```

Query 2. Return the id, name all movies that feature every hero in the same universe as the movie. Order by movie name.

Answer.

```
select
    id
    , name
from
    movies m
where
    not exists
    (select
        1
    from
        heroes h
    where
        h.universe=m.universe
        and not exists
            (select
                1
            from
                heroesinmovie hm
            where
                hm.hero_id = h.id
                and hm.show_id = m.id));
```

Alternate answer. This solution assumes there is at least one hero in each universe (otherwise we would have to do outer join!).

```
select
    m.id
    , m.name
from
    movies m
    , heroesinmovie hm
    , heroes h
where
    hm.show_id = m.id
    and hw.hero_id = h.id
    and h.universe = m.universe
group by
    m.id
    , m.universe  --not needed because universe is unique given movie id
                  --but does not hurt having it in group by either
having
    count(*) =
            (select count(*)
             from heroes h
             where h.universe=m.universe);
```