Worksheet

Use the Mondial dataset in hw5 to solve the following problems

1. Return the set of all mountains.

   ```
   SELECT x.mondial.mountain FROM world x;
   ```

2. Return each mountain one by one. Compare it to Problem 1.

   ```
   SELECT y as mountain FROM world x, x.mondial.mountain y;
   ```

3. Return name and type for each mountain, in descending order of the height.

   ```
   SELECT y.name, y.`-type`, y.height
   FROM world x, x.mondial.mountain y
   ORDER BY INT(y.height) DESC;
   ```

4. Find mountains located in more than 1 country. Your query should return mountain name and the count.

   ```
   SELECT y.name as mountainName, Count(*) as numCountries
   FROM world x,  x.mondial.mountain y, split(y.`-country`, ' ') r
   GROUP BY y.name
   HAVING numCountries > 1;
   ```

5. For each country, return the country name and a list of all the mountain names in that country.

   ```
   SELECT y.name as countryName, m as mountainList
   FROM world x,  x.mondial.country y
   let m = ( SELECT z.name as mountain
             FROM world x2, x2.mondial.mountain z, split(z.`-country`, ' ') r
            where y.`-car_code` = r);
   ```

Suppose that we store all the data for our social network in a single dataset of Users:

```
[{“handle”: “biebs”,
  “name”: “Justin Bieber”,
  “home_city”: “Somewhere, Canada”,
  “bio”: “...”,
  “friends”: [“kimkardashian”, “shaq”, ...],
  “messages”: [ {“text”: “:-* :-* :-* :-*”, “from_city”: “Los Angeles,
CA”},      {“text”: “New. Music. Friday.”, “from_city”: “Los Angeles,
CA”},
  ... ] }
...]
```

1. For each home city, compute a list of users from that home city. Your query should return a list where each element consists of city name and list of User handles.

   ```
   SELECT DISTINCT x.home_city as homeCity,
                       (SELECT y.handler
                         FROM Users y
                         WHERE y.home_city = x.home_city ) as userHandleList
   FROM Users x;
   ```

2. Return pairs of users that have at least one common friend.

   ```
   SELECT x.handle, y.handle
   FROM Users x, x.Friends xf, Users y, y.Friends yf
   WHERE x.handle < y.handle
           AND xf = yf;
   ```

   We use ‘<’ operator to remove duplicate pairs – (a,b) and (b,a)