

Section 10 Worksheet Solutions

Use the Mondial dataset in homework 7 to solve the following problems

1. Return the set of all mountains, i.e., as a single tuple with an attribute containing the collection of all mountains.

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

2. Return each mountain one by one, i.e., as a collection of tuples that each contain a single mountain. Compare it to Problem 1.

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

3. Return name, type, and height 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 count(*) > 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.handle  
                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 DISTINCT 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)