

## Day 16: *SQL II: Join and Group by*

The rest of the trip was a blur. Dot allowed themselves to be dragged along by the whirlwind of a woman. When they arrived at the airport in Nepal, a man in a new black suit and dark sunglasses met them, nodding at the woman before leading the two of them to a shiny stretch limo. Seated in the plush of the vehicle's interior, the woman conjured a bottle of champagne from a mini-fridge and popped it open. "To adventure!" she laughed, pouring Dot a foaming glass and clinking it with her own. The car stopped after a long drive, during which the woman speed-crocheted an entire lime-green bodysuit for Dot. They got out and switched to a decidedly less swanky all-terrain vehicle. Dot found a parka, snow pants, a balaclava, and climbing gear underneath their seat. After a while, the car skid to a stop. Dot opened the door to find that they were at the foot of a massive mountain. All around them were mountains and valleys as far as the eye could see.

"Come on, sidekick, let's go!" the woman yelled, bounding out of the vehicle and running up the mountain at full speed. Dot cautiously began to walk after her. Inside them was a strange mix of fear and exhilaration — they were scared that they wouldn't be able to climb the mountain, but they wanted to try. They slowly began the ascent, with the energetic woman rapidly climbing high above them. When Dot looked up, the mountain stretched to the heavens, impossibly tall. "How long will it take to climb this mountain, anyway?" Dot yelled up at the woman. "About two months!" she yelled back down. Dot stopped in their tracks. "What?! How will we eat?" The woman's laughter came from high above them, "we'll hunt animals, of course!" Wordlessly, Dot stepped back down the mountain and walked back to the car. The driver nodded at Dot and asked where to take them. "Back to the airport, please," Dot said. "I'm going to Japan. No more mountains!" Dot thought about the woman climbing up the mountain on the drive back. Was it even possible for her to reach the peak? Let's examine the data to see whether past expeditions in the Himalayas have been successful.

## Tutorial

We will continue with more complex SQL queries in this challenge. We will learn how to join data from tables, and how to aggregate data.

### Relations between the tables

We call the database relational when there are relations between tables. See the diagram that displays simple database relations below:

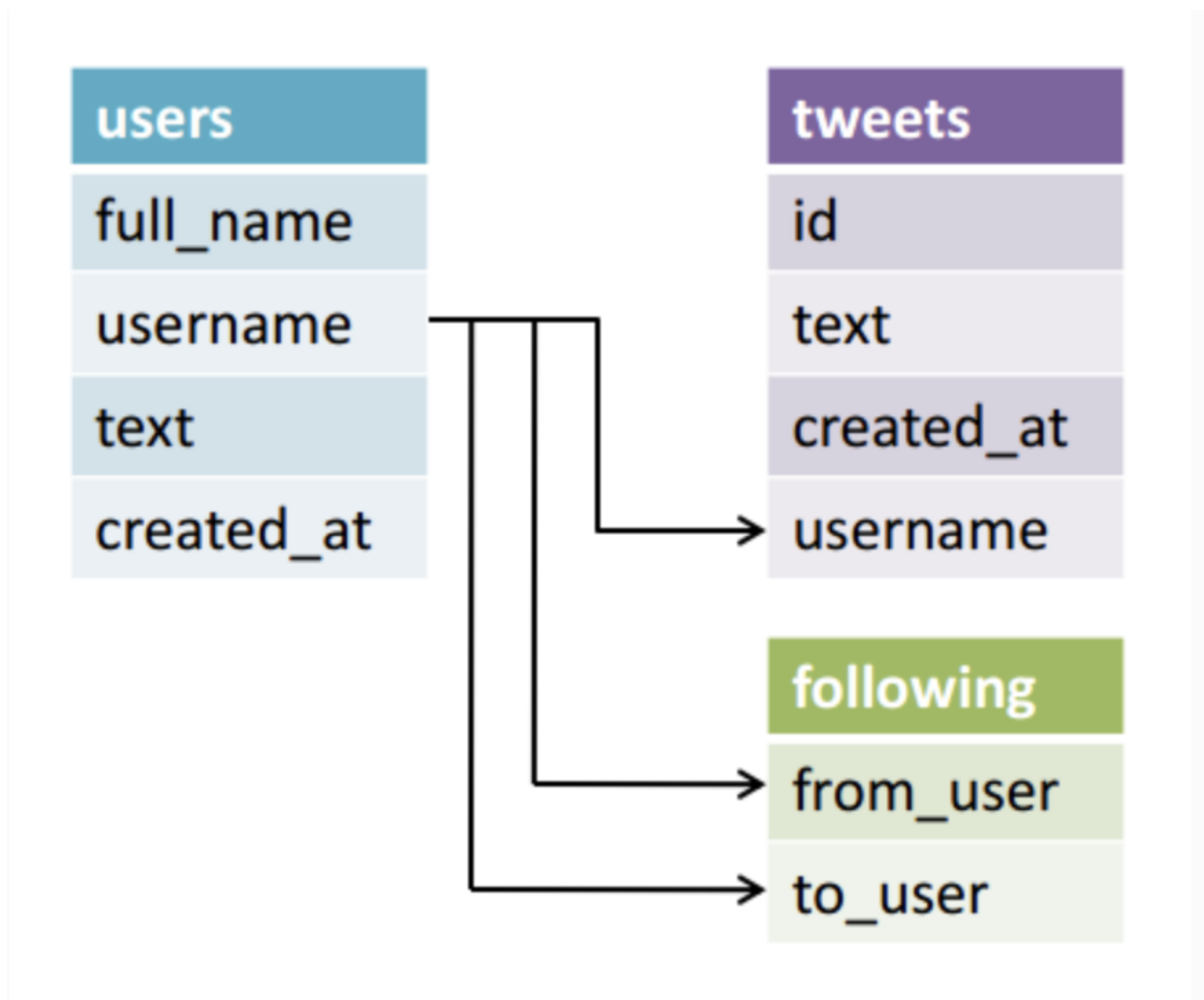


Image Taken from [here](#)

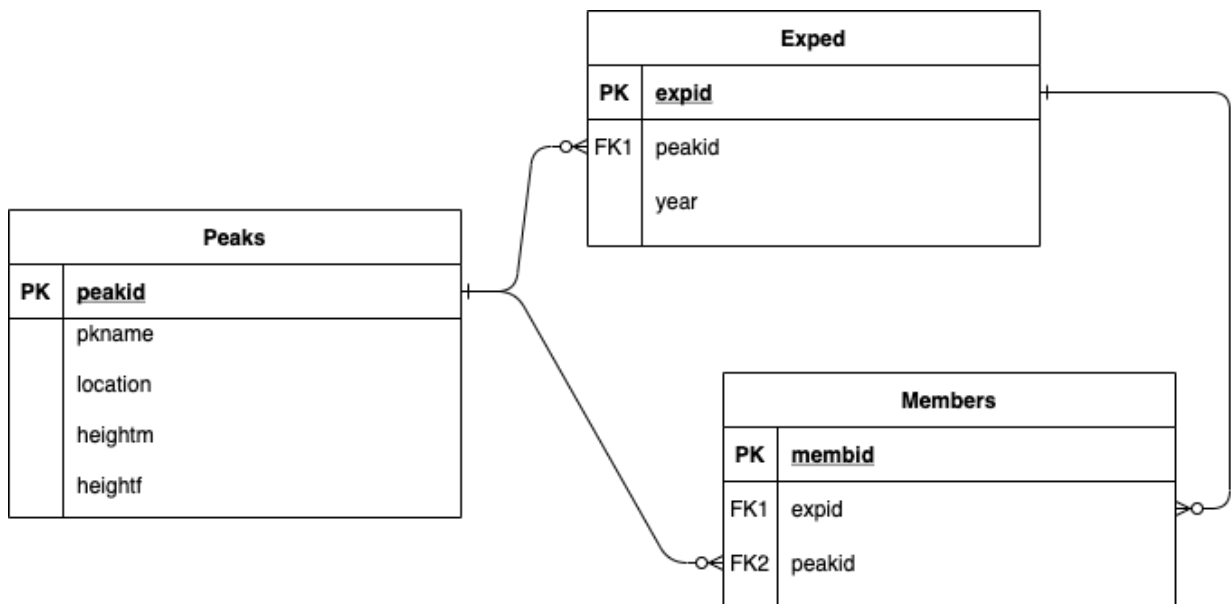
We can see that the table **users** is related to both **tweets** and **following** by using its key **username**.

We can display the same kind of diagram for our database **himalayas.db**.

```
import sqlite3
import pandas as pd
conn = sqlite3.connect("himalayas.db")
```

The tables are connected with primary (PK) and foreign (FK) keys, and there can be more than one type of relationship between them:

- **1 to 1** - when one line from table A is connected to one line in table B
- **1 to many** - when one line from table A is connected to more lines in table B
- **many to many** - when more lines from table A are connected to more lines in table B

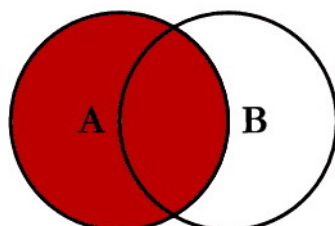


In our database we deal with **1 to many** relationships because we can have:

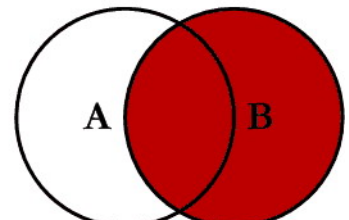
- more expeditions that went to a specific peak
- more expedition members who went to a specific peak

When the tables are connected through primary and foreign keys, we can use SQL JOIN to query the data from both tables. There are more types of JOIN which are shown below:

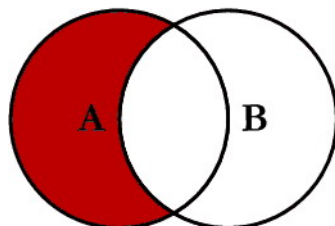
## SQL JOINS



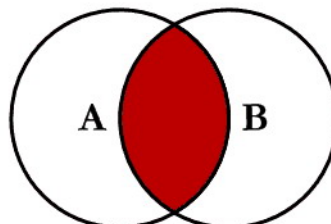
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



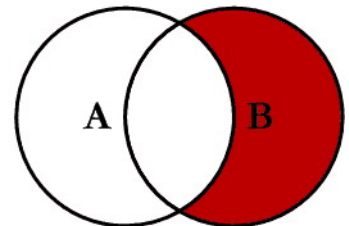
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



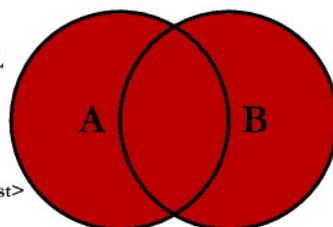
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



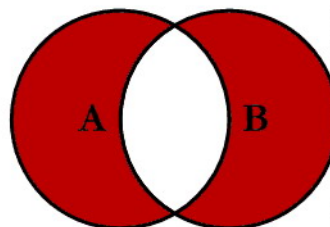
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

Image Taken from [here](#)

```
exped = pd.read_sql('select * from exped',conn)
members = pd.read_sql('select * from members',conn)
```

We use table alias `from exped as E` to identify which columns come from which tables:

```
pd.read_sql("""
    select E.expid, --this is from the table E -> exped
           M.fname, --this is from the table M -> members
           M.calcage as age
    from exped as E
    inner join members as M
           on E.expid = M.expid
""",conn)
```

We can also select all columns from one table:

```
pd.read_sql("""
    select E.*, --all columns from the table E -> exped
           M.fname, --this is from the table M -> members
           M.calcage as age
    from exped as E
    inner join members as M
           on E.expid = M.expid
""",conn)
```

To further play around with simple SQL JOIN queries, you can watch this [interactive video](#).

## Challenge

Answer following questions using our database:

1. **How many expeditions went to peak Everest?**
2. **How many people went to peak Everest? (One person could have gone more than once)**

```
In [2]: import sqlite3
import pandas as pd
conn = sqlite3.connect("himalayas.db")
```

```
In [6]: # SOLUTION

len(pd.read_sql("""
    select P.pkname,
           E.expid
    from peaks as P
    inner join exped as E
           on P.peakid = E.peakid
    where P.pkname = 'Everest'
""",conn))
```

```
Out[6]: 2161
```

```
In [7]: len(pd.read_sql("""
```

```
select P.pkname,  
       M.fname,  
       M.lname,  
       M.expid  
from peaks as P  
inner join members as M  
      on P.peakid = M.peakid  
where P.pkname = 'Everest'  
""",conn))
```

Out[7]: 21900