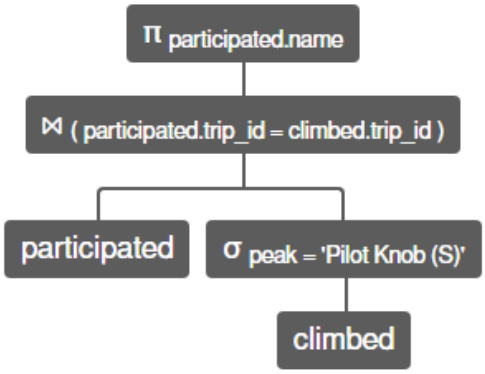
Blaise Bowman 11/11/2019

CIS4301 Assignment #3

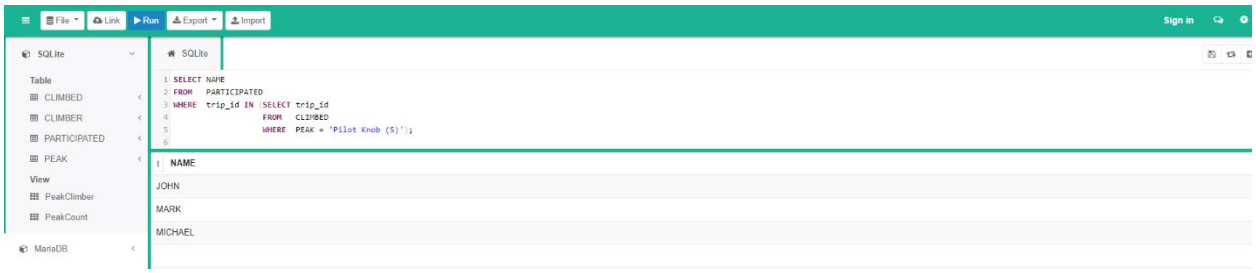
Each problem spans about a page

1.

```
SELECT NAME
FROM   PARTICIPATED
WHERE  trip_id IN (SELECT trip_id
                   FROM   CLIMBED
                   WHERE  PEAK = 'Pilot Knob (S)');
```

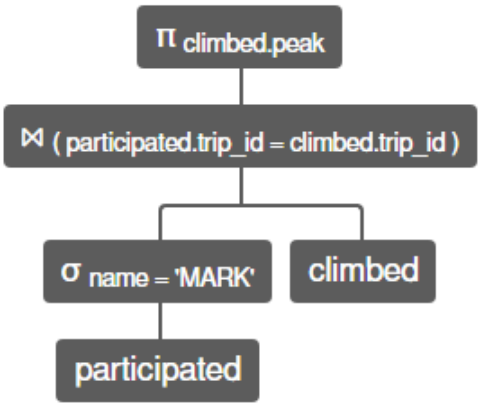


π participated.name (participated ⋈ (participated.trip_id = climbed.trip_id) (σ peak = 'Pilot Knob (S)' (climbed)))

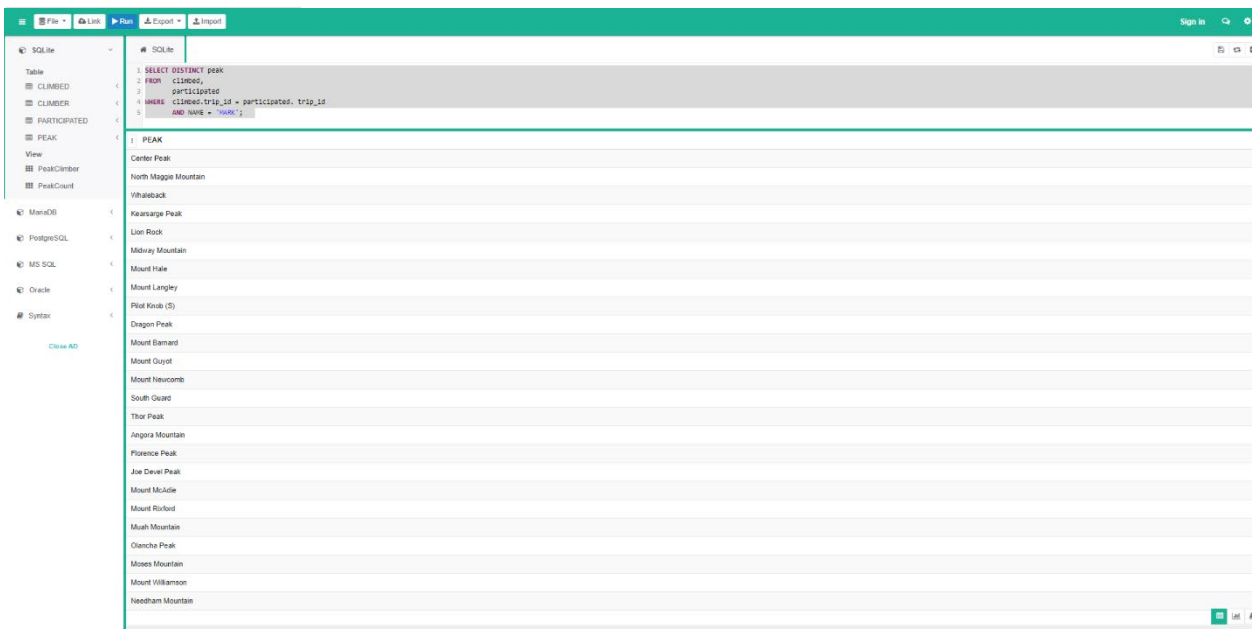


2.

```sql
SELECT DISTINCT peak
FROM   climbed,
       participated
WHERE  climbed.trip_id = participated. trip_id
       AND NAME = 'MARK';
```

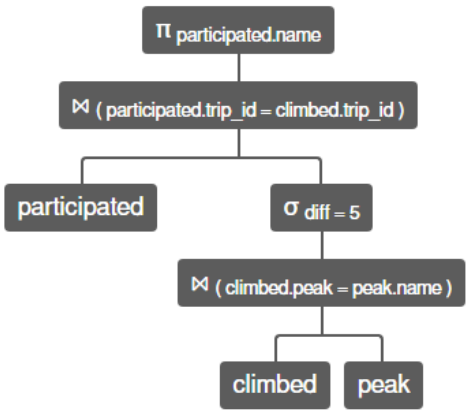


$$\pi_{\text{climbed.peak}}\left(\left(\sigma_{\text{name = 'MARK'}}(\text{participated})\right) \bowtie_{(\text{participated.trip\_id = climbed.trip\_id})} (\text{climbed})\right)$$

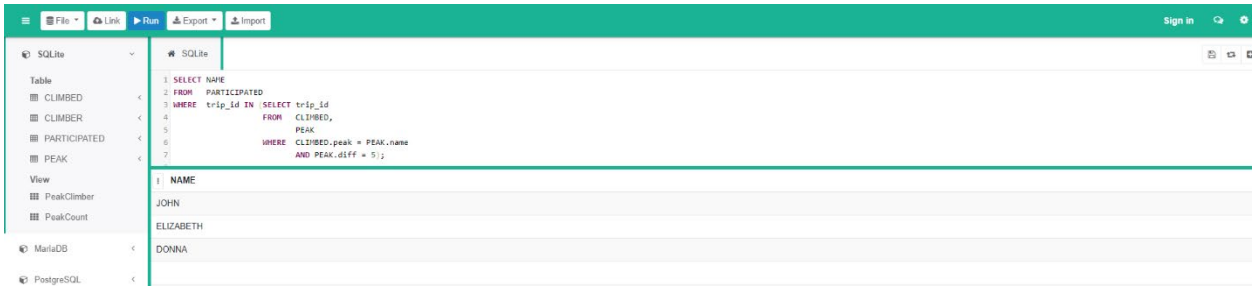


3.

```
SELECT NAME
FROM   PARTICIPATED
WHERE  trip_id IN (SELECT trip_id
                   FROM   CLIMBED,
                          PEAK
                   WHERE  CLIMBED.peak = PEAK.name
                AND PEAK.diff = 5);
```

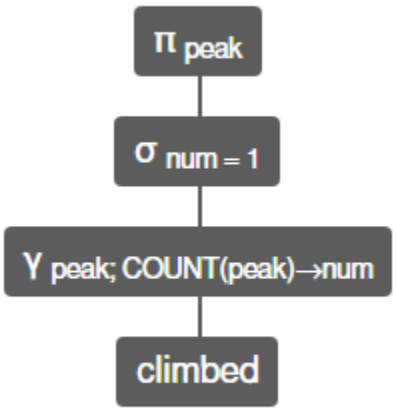


π participated.name (participated ⋈ (participated.trip_id = climbed.trip_id) (σ diff = 5 (climbed ⋈ (climbed.peak = peak.name) peak)))

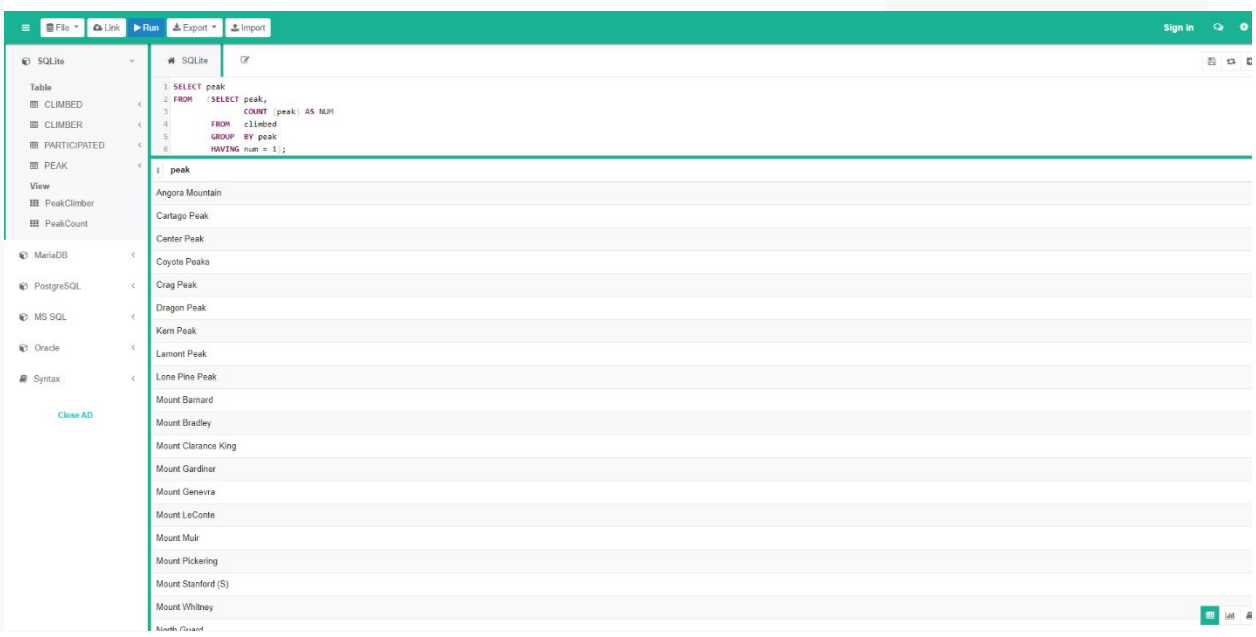


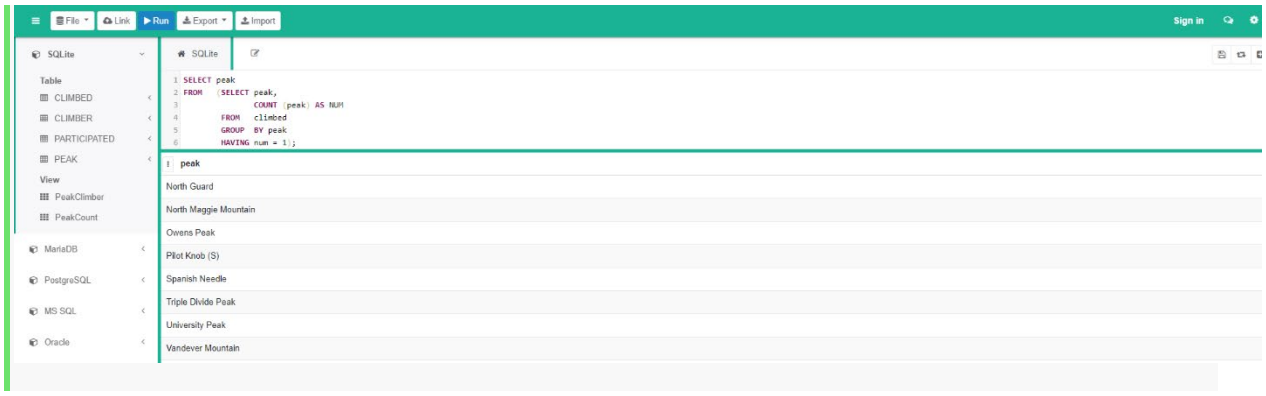
4.

```sql
SELECT peak
FROM   (SELECT peak,
               Count (peak) AS NUM
        FROM   climbed
        GROUP  BY peak
        HAVING num = 1);
```



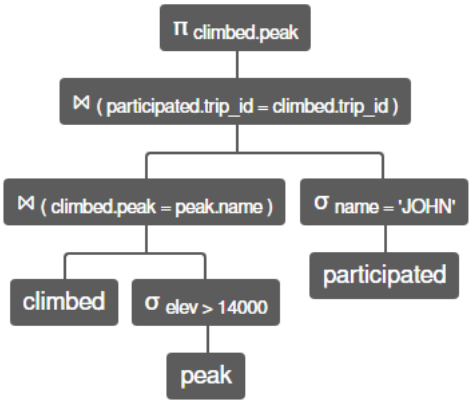
$\pi_{peak}(\sigma_{num = 1}(\gamma_{peak;\ COUNT(peak) \rightarrow num}(climbed)))$



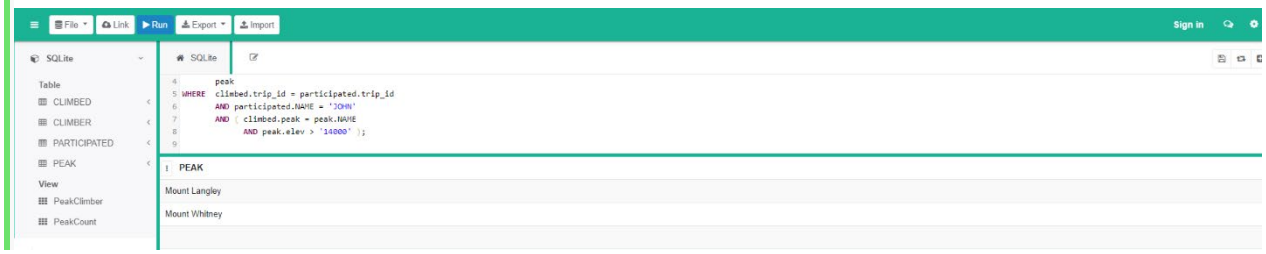


5.

```
SELECT DISTINCT peak
FROM   climbed,
       participated,
       peak
WHERE  climbed.trip_id = participated.trip_id
       AND participated.NAME = 'JOHN'
       AND ( climbed.peak = peak.NAME
             AND peak.elev > '14000' );
```

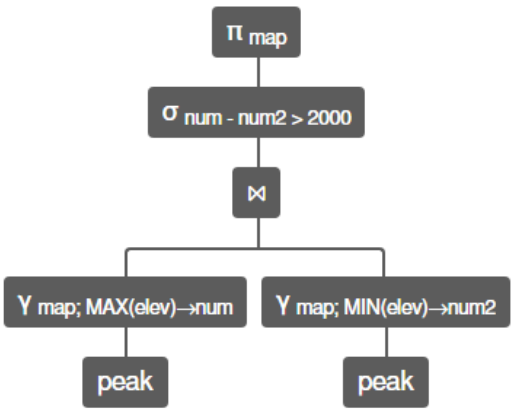


π climbed.peak ((climbed ⋈ (climbed.peak = peak.name) (σ elev > 14000 peak)) ⋈ (participated.trip_id = climbed.trip_id) (σ name = 'JOHN' (participated)))

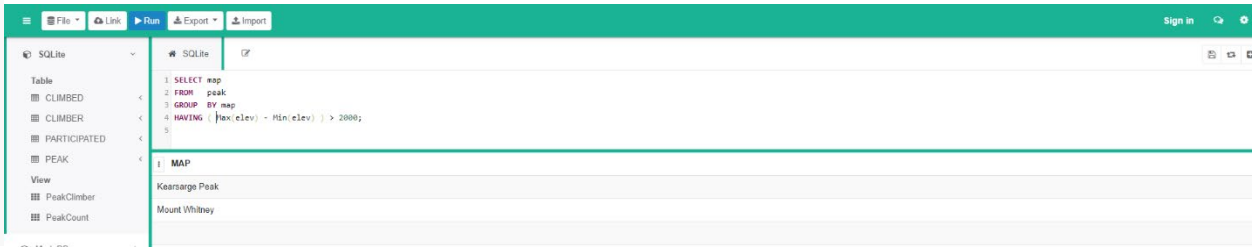


6.

```sql
SELECT map
FROM   peak
GROUP  BY map
HAVING ( Max(elev) - Min(elev) ) > 2000;
```

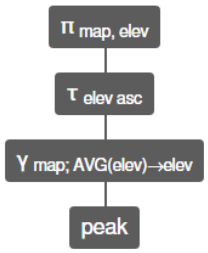


$\pi_{map}(\sigma_{num - num2 > 2000}((\gamma_{map;\ MAX(elev)\rightarrow num}(peak)) \bowtie (\gamma_{map;\ MIN(elev)\rightarrow num2}(peak))))$

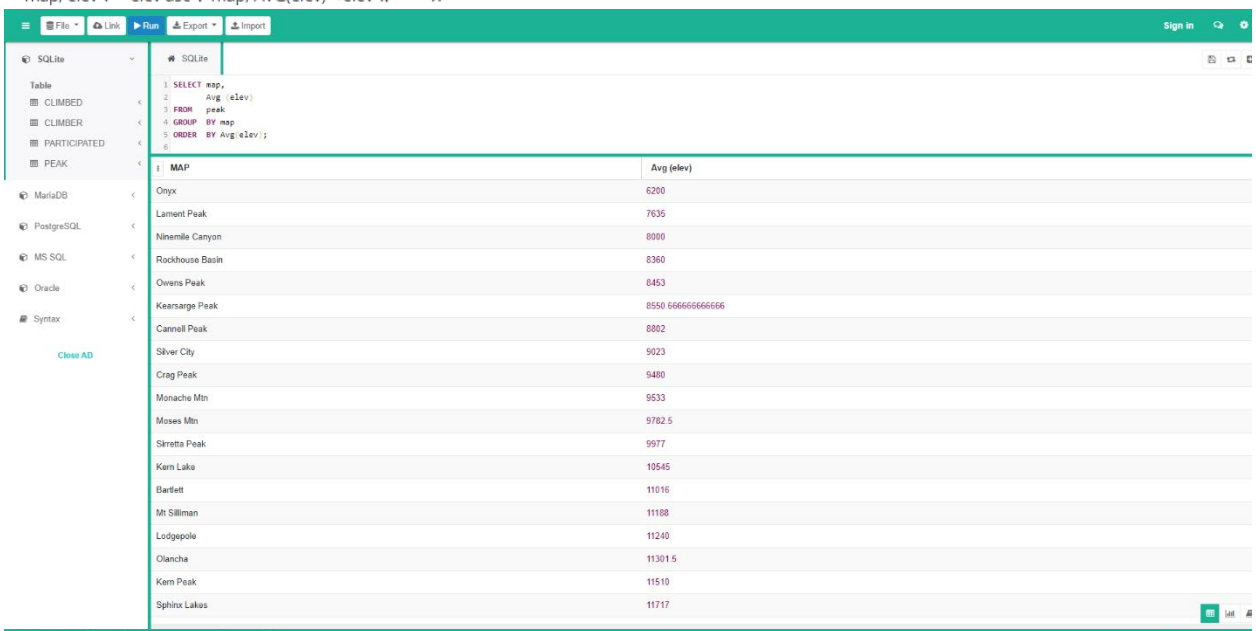


7.

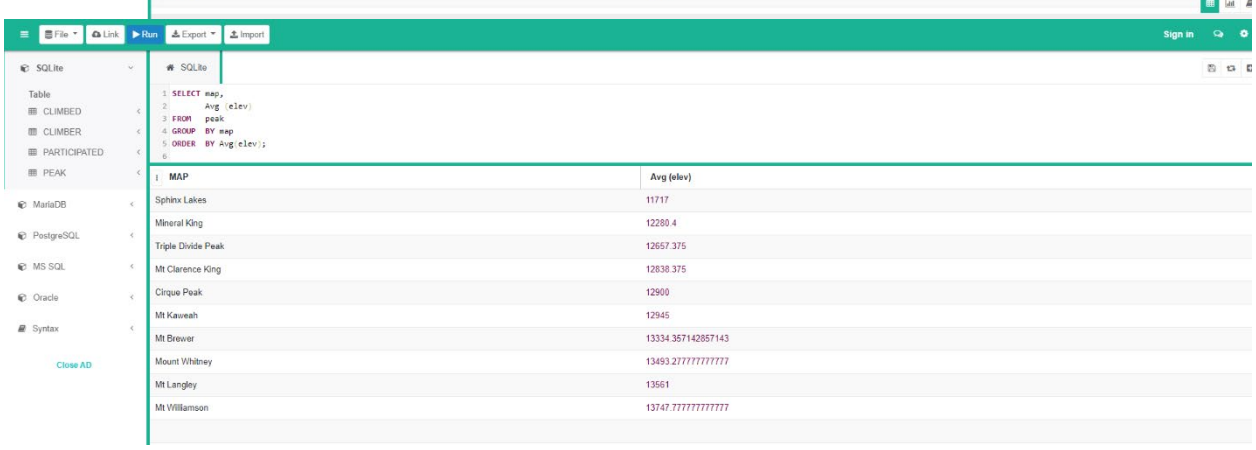
```sql
SELECT map,
       Avg (elev)
FROM   peak
GROUP  BY map
ORDER  BY Avg(elev);
```



π map, elev (τ elev asc γ map; AVG(elev)→elev (peak))



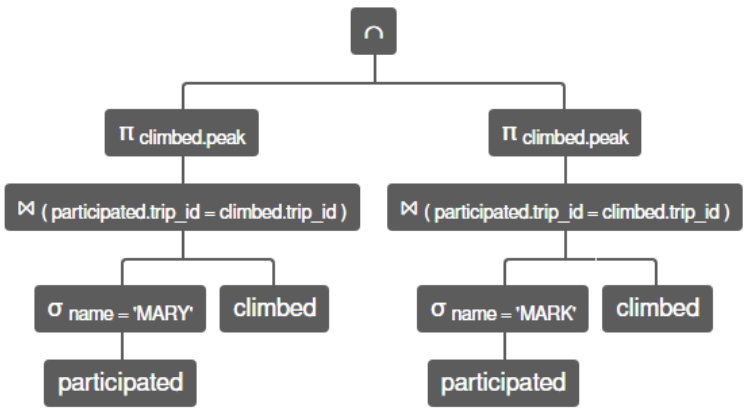
MAP	Avg (elev)
Onyx	6200
Lament Peak	7635
Ninemile Canyon	8000
Rockhouse Basin	8360
Owens Peak	8453
Kearsarge Peak	8550.666666666666
Cannell Peak	8802
Silver City	9023
Crag Peak	9480
Monache Mtn	9533
Moses Mtn	9782.5
Sirretta Peak	9977
Kern Lake	10545
Bartlett	11016
Mt Silliman	11188
Lodgepole	11240
Olancha	11301.5
Kern Peak	11510
Sphinx Lakes	11717



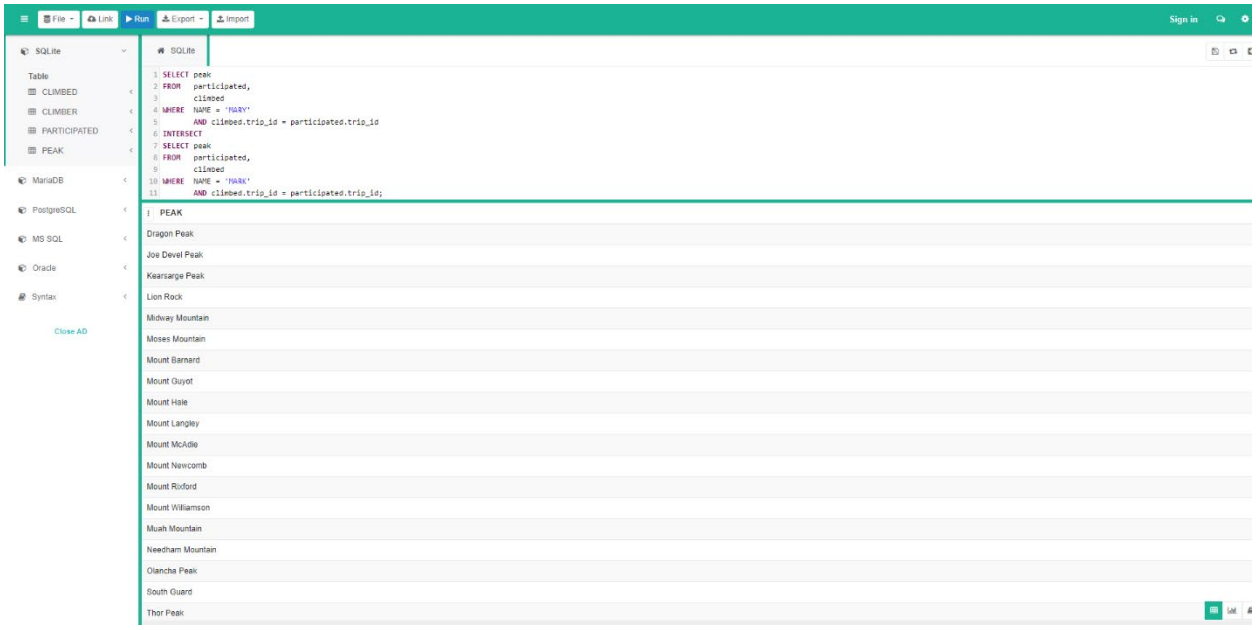
MAP	Avg (elev)
Sphinx Lakes	11717
Mineral King	12280.4
Triple Divide Peak	12657.375
Mt Clarence King	12838.375
Cirque Peak	12900
Mt Kaweah	12945
Mt Brewer	13334.357142857143
Mount Whitney	13493.277777777777
Mt Langley	13561
Mt Williamson	13747.777777777777

8.

```sql
SELECT peak
FROM   participated,
       climbed
WHERE  NAME = 'MARY'
       AND climbed.trip_id = participated.trip_id
INTERSECT
SELECT peak
FROM   participated,
       climbed
WHERE  NAME = 'MARK'
       AND climbed.trip_id = participated.trip_id;
```

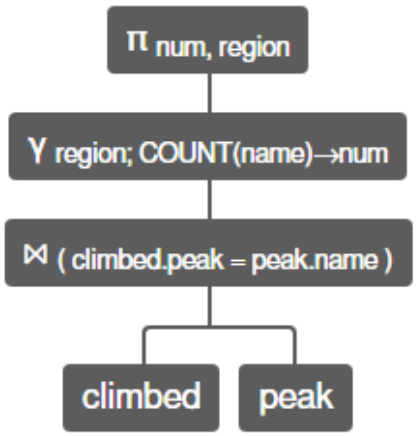


$\pi_{\text{climbed.peak}}((\sigma_{\text{name = 'MARY'}}(\text{participated})) \bowtie_{(\text{participated.trip\_id = climbed.trip\_id})} (\text{climbed})) \cap \pi_{\text{climbed.peak}}((\sigma_{\text{name = 'MARK'}}(\text{participated})) \bowtie_{(\text{participated.trip\_id = climbed.trip\_id})} (\text{climbed}))$

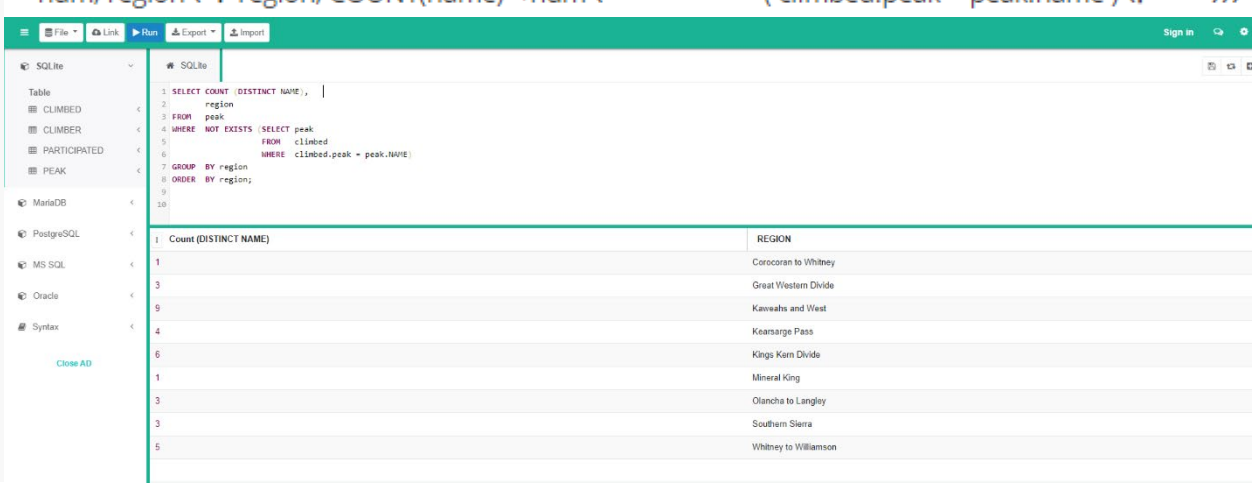


9.

```sql
SELECT Count (DISTINCT NAME),
       region
FROM   peak
WHERE  NOT EXISTS (SELECT peak
                   FROM   climbed
                   WHERE  climbed.peak = peak.NAME)
GROUP  BY region
ORDER  BY region;
```

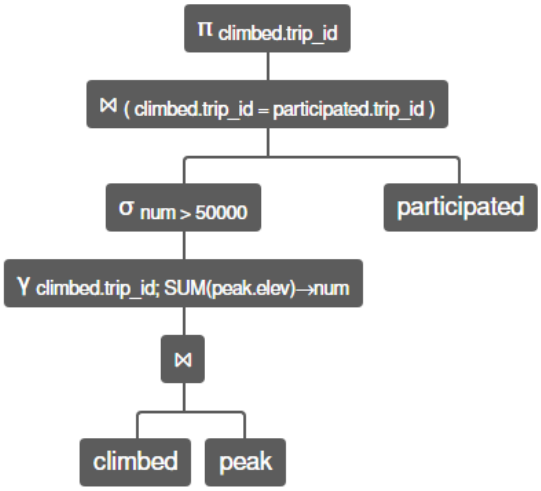


$$\pi_{\text{num, region}} (\gamma_{\text{region; COUNT(name)}\rightarrow\text{num}} (\text{climbed} \bowtie_{\text{climbed.peak = peak.name}} (\text{peak})))$$

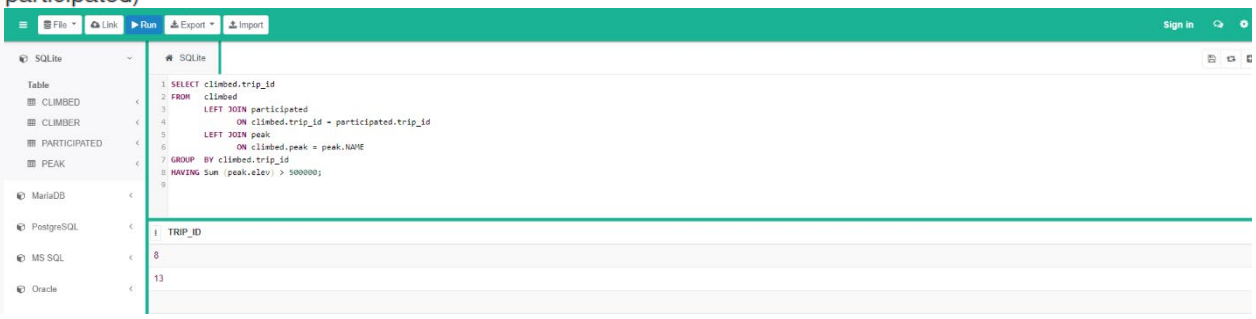


10.

```sql
SELECT climbed.trip_id
FROM   climbed
       LEFT JOIN participated
              ON climbed.trip_id = participated.trip_id
       LEFT JOIN peak
              ON climbed.peak = peak.NAME
GROUP  BY climbed.trip_id
HAVING Sum (peak.elev) > 500000;
```

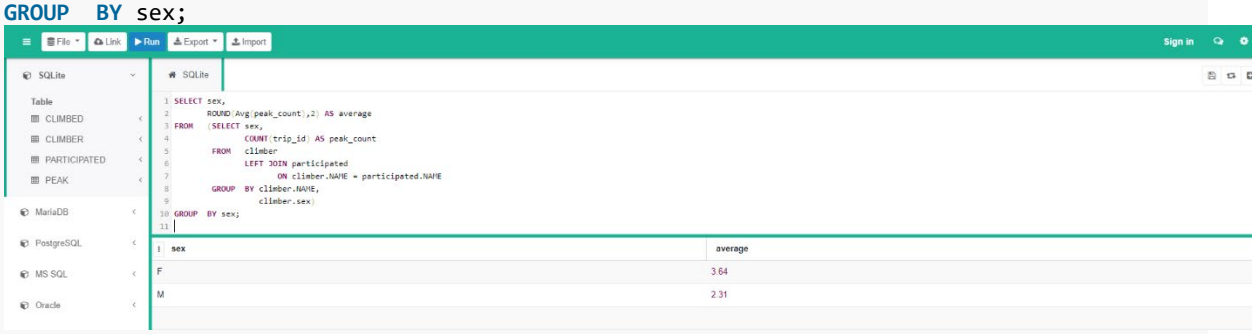


$\pi_{\text{climbed.trip\_id}}((\sigma_{\text{num} > 50000}(\gamma_{\text{climbed.trip\_id; SUM(peak.elev)}\rightarrow\text{num}}(\text{climbed} \bowtie \text{peak}))) \bowtie_{(\text{climbed.trip\_id} = \text{participated.trip\_id})} \text{participated})$



11.

```sql
SELECT sex,
       ROUND(Avg(peak_count),2) AS average
FROM   (SELECT sex,
               Count(trip_id) AS peak_count
        FROM   climber
               LEFT JOIN participated
                      ON climber.NAME = participated.NAME
        GROUP  BY climber.NAME,
                  climber.sex)
GROUP  BY sex;
```

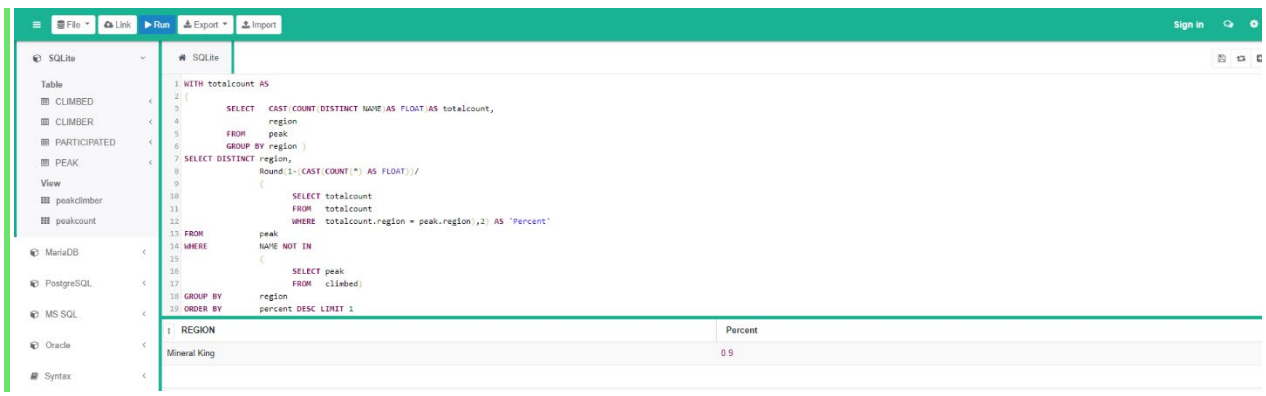


12.

```sql
CREATE VIEW peakclimber
AS
  SELECT DISTINCT NAME,
                  peak
  FROM   climbed,
         participated
  WHERE  climbed.trip_id = participated.trip_id;

CREATE VIEW peakcount
AS
  SELECT NAME,
         Count(peak) AS peakNumber
  FROM   peakclimber
  GROUP  BY NAME;

SELECT NAME
FROM   peakclimber
WHERE  peak IN (SELECT peak
                FROM   peakclimber
                WHERE  NAME = 'MARIA')
       AND NAME != 'MARIA'
GROUP  BY NAME
HAVING Count(peak) IN (SELECT peaknumber
                       FROM   peakcount
                       WHERE  NAME = 'MARIA');
```

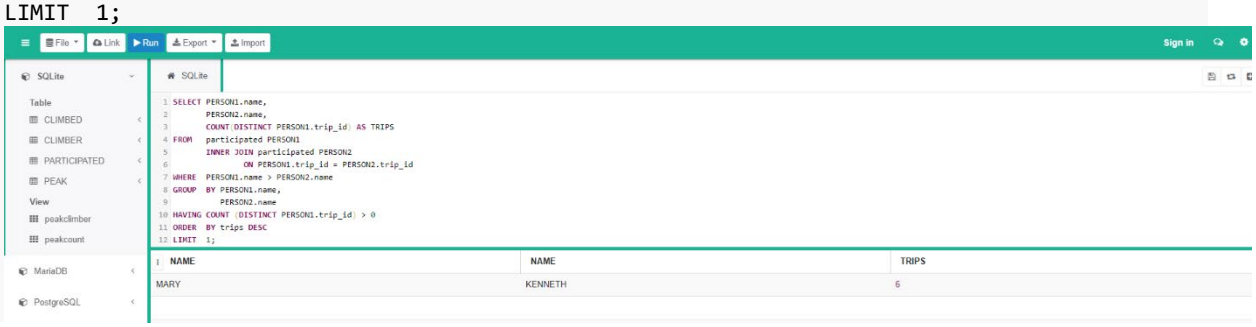
13.

```sql
WITH totalCount as(
    SELECT
        CAST(COUNT(distinct name)as FLOAT)as totalCount,region
FROM peak
  group by region
 )

SELECT DISTINCT REGION,
    round(1-
(CAST(COUNT(*) as FLOAT))/(select totalCount from totalCount where totalCount.region = peak.re
gion),2) as 'Percent'
FROM PEAK
WHERE NAME not IN
(SELECT PEAK FROM CLIMBED)

GROUP BY region
order by percent desc
limit 1
```



14.

```sql
SELECT PERSON1.name,
       PERSON2.name,
       Count(DISTINCT PERSON1.trip_id) AS TRIPS
FROM   participated PERSON1
       INNER JOIN participated PERSON2
               ON PERSON1.trip_id = PERSON2.trip_id
WHERE  PERSON1.name > PERSON2.name
GROUP  BY PERSON1.name,
          PERSON2.name
HAVING Count (DISTINCT PERSON1.trip_id) > 0
ORDER  BY trips DESC
LIMIT  1;
```



15.

```sql
WITH table_peaks
     AS (SELECT climbed.peak,
                climbed.when_climbed,
                participated.NAME
         FROM   climbed
                INNER JOIN participated
                        ON participated.trip_id = climbed.trip_id)
SELECT DISTINCT NAME
FROM   (SELECT t1.NAME
        FROM   table_peaks t1
               LEFT JOIN table_peaks t2
                      ON t1.NAME = t2.NAME
                         AND ( t2.when_climbed > t1.when_climbed
                                OR ( t2.when_climbed = t1.when_climbed
                                     AND t2.peak <> t1.peak ) )
                         AND NOT ( t2.when_climbed > Date(
       t1.when_climbed, 60) )) AS
       query
GROUP  BY NAME
HAVING Count (*) > 20;
```