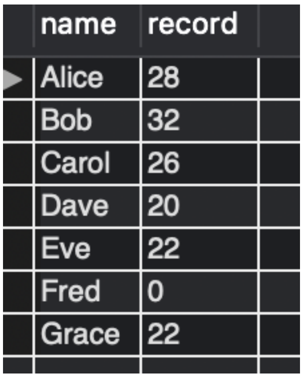
1. Print each user’s name, along with the number of times they have recorded a location.

```
SELECT User.name, COUNT(Location.id) AS record
FROM User LEFT OUTER JOIN Location
      ON User.id = Location.user
      GROUP by User.name;
```

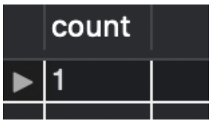


name	record
Alice	28
Bob	32
Carol	26
Dave	20
Eve	22
Fred	0
Grace	22

7 Rows Returned

2. How many cities are in the same state as Melbourne? (Don’t count Melbourne in your answer.)

```
SELECT COUNT(cityName) AS count
FROM City
WHERE state =
      (SELECT state FROM City
      WHERE cityName = "Melbourne")
AND cityName != "Melbourne";
```

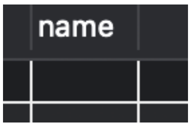


count
1

1 Row Returned

3. List the names of any members of Academia gym who have been north of Brunswick gym.

```
SELECT DISTINCT User.name
FROM User INNER JOIN Location
      ON User.id = Location.user
      WHERE Location.latitude >
            (SELECT latitude FROM Gym
            WHERE name = "Brunswick")
      AND User.gym IN
            (SELECT id FROM Gym
            WHERE name = "Academia");
```

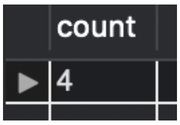


name

0 Row Returned

4. How many users are registered with gyms in the state of Vic?

```
SELECT COUNT(id) AS count
FROM User
WHERE gym IN
    (SELECT id FROM Gym
    WHERE city IN
        (SELECT id FROM City
        WHERE state = "Vic"));
```

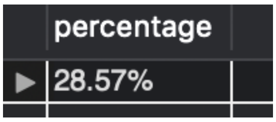


count
4

1 Row Returned

5. What percentage of the total number of users are not affiliated with gyms?

```
SELECT CONCAT
(FORMAT
    ((SELECT COUNT(id)/
        (SELECT COUNT(id) FROM User)*100
        FROM User
        WHERE gym IS NULL),2),"%")
AS percentage;
```

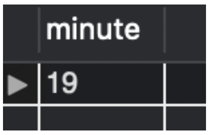


percentage
28.57%

1 Row Returned

6. How much time elapsed between the first and last recorded locations of the user with id 4?

```
SELECT TIMESTAMPDIFF(minute,
    (SELECT whenRecorded FROM Location
    WHERE user = "4"
        ORDER BY whenRecorded
        LIMIT 1),
    (SELECT whenRecorded FROM Location
    WHERE user = "4" ORDER BY whenRecorded DESC
    LIMIT 1))
AS minute;
```

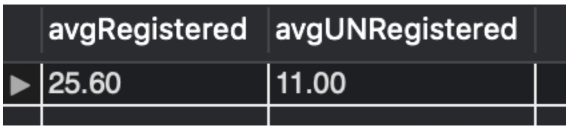


minute
19

1 Row Returned

7. Print as two columns: the average number of locations recorded by registered users, and the average number of locations recorded by unregistered users.

```
SELECT FORMAT
    ((SELECT COUNT(Location.id)/
        (SELECT COUNT(id) FROM User
        WHERE gym IN
            (SELECT id FROM Gym))
    FROM User LEFT JOIN Location
        ON User.id = Location.user
        WHERE gym IN
            (SELECT id FROM Gym)),2)
AS avgRegistered,
FORMAT
    ((SELECT COUNT(Location.id)/
        (SELECT count(id) FROM user
        WHERE gym IS NULL)
    FROM user LEFT JOIN Location
        ON User.id = Location.user
        WHERE gym IS NULL),2)
AS avgUNRegistered;
```

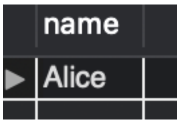


avgRegistered	avgUNRegistered
25.60	11.00

1 Row Returned

8. List the names of users who have run within 100m of the Doug McDonell building. (DMD is at longitude 144.9630, latitude -37.7990 .)

```
SELECT DISTINCT name
FROM user INNER JOIN Location
    ON User.id = Location.user
    WHERE SQRT
        (POWER(Location.latitude-(-37.7990),2)+
        POWER(Location.longitude-144.9630,2))*100<0.1;
```



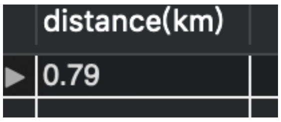
name
Alice

1 Row Returned

9. What is the distance between the northern-most and southern-most locations to

which Alice has run?

```
SELECT FORMAT
(SQRT(POWER
    ((SELECT longitude
    FROM User INNER JOIN Location
        ON User.id = Location.user
        WHERE latitude =
            (SELECT MAX(latitude)
            FROM User INNER JOIN Location
                ON User.id = Location.user
                WHERE name = "Alice")
        AND name = "Alice"
        LIMIT 1) -
    (SELECT longitude
    FROM user INNER JOIN location
        ON User.id = Location.user
        WHERE latitude =
            (SELECT MIN(latitude)
            FROM User INNER JOIN Location
                ON User.id = Location.user
                WHERE name = "Alice")
        AND name = "Alice" limit 1),2) +
POWER
    ((SELECT MAX(latitude)
    FROM User INNER JOIN Location
        ON User.id = Location.user
        WHERE name = "Alice") -
    (SELECT MIN(latitude)
    FROM User INNER JOIN Location
        ON User.id = Location.user
        WHERE name = "Alice"),2))*100,2)
AS "distance(km)";
```



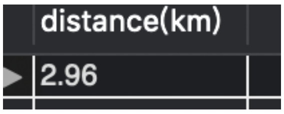
```
distance(km)
0.79
```

1 Row Returned

10. Show the total distance that Alice has run. Calculate this by summing the individual distances between each successive pair of locations.

```
SELECT FORMAT
(SUM(SQRT(POWER(a.latitude-b.latitude,2) +
POWER(a.longitude-b.longitude,2))*100),2)
AS "distance(km)"
```

```
FROM Location a INNER JOIN Location b
        ON a.whenRecorded - b.whenRecorded = -100
        WHERE a.user = 1
        AND b.user = 1;
```



```
distance(km)
2.96
```

1 Row Returned