**Assignment 2**

**Name: Hongzhi Fu**

**Student Number:1058170**

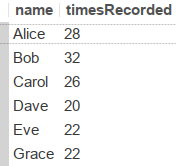
1. Print each user’s name, along with the number of times they have recorded a location.

SELECT name, COUNT(\*) AS timesRecorded

FROM User INNER JOIN Location

ON User.id = Location.user

GROUP BY Location.user;



6 Rows Returned

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

SELECT COUNT(\*) AS numofCities

FROM City AS C1

WHERE C1.cityName != 'Melbourne'

AND C1.state =

(SELECT state

FROM City AS C2

WHERE C2.cityName = 'Melbourne');



1 Row Returned

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

SELECT User.name

FROM User INNER JOIN Gym INNER JOIN Location

ON User.gym = Gym.id AND User.id = Location.user

WHERE Gym.name = 'Academia'

AND Location.latitude >

(SELECT latitude

FROM Gym

WHERE name = 'Brunswicks');

Question 3

0 Rows Returned

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

SELECT COUNT(\*) AS registeredUsers

FROM User

WHERE User.gym IN

(SELECT Gym.id

FROM Gym INNER JOIN City

ON Gym.city = City.id

WHERE City.state = 'Vic');



1 Row Returned

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

SELECT CONCAT(TRUNCATE((COUNT(\*) /

(SELECT COUNT(\*)

FROM User)) \* 100, 2), '%')

AS percentage

FROM User

WHERE User.gym IS NULL;

Question 5

1 Row Returned

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

SELECT TIMESTAMPDIFF(second, MIN(whenRecorded), MAX(whenRecorded))

AS timeElapsed

FROM Location

WHERE user = 4;

Question 6

Note: The unit of time here is second.

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 ROUND(COUNT(\*) /

(SELECT COUNT(\*)

FROM User

WHERE User.gym IS NOT NULL), 2)

AS avgRegMem,

ROUND((((SELECT COUNT(\*)

FROM Location) - COUNT(\*)) /

(SELECT COUNT(\*)

FROM User

WHERE User.gym IS NULL)), 2)

AS avgUnregMem

FROM Location

WHERE Location.user IN

(SELECT User.id

FROM User

WHERE User.gym IS NOT NULL);

Question 7

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(User.name)

FROM User INNER JOIN Location

ON User.id = Location.user

WHERE SQRT(POW(Location.longitude - 144.9630, 2) + POW(Location.latitude - (-37.7990), 2)) \* 100 < 0.1;



1 Row Returned

9. What is the distance between the northern-most and southern-most locations to Alice has run?

SELECT ROUND(MAX(SQRT(POW(L1.latitude - L2.latitude, 2) + POW(L1.longitude - L2.longitude, 2)) \* 100), 2) AS distance

FROM User INNER JOIN Location L1 INNER JOIN Location L2

ON User.id = L1.user and User.id = L2.user

WHERE User.name = 'Alice'

AND L1.latitude =

(SELECT MAX(Location.latitude)

FROM Location INNER JOIN User

ON Location.user = User.id

WHERE User.name = 'Alice')

AND L2.latitude =

(SELECT MIN(Location.latitude)

FROM Location INNER JOIN User

ON Location.user = User.id

WHERE User.name = 'Alice');

Question 9

Note: The maximum distance between northern-most and southern-most locations is selected in the result.

1 Row Returned

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

SELECT ROUND(SUM(SQRT(POW(L1.latitude - L2.latitude, 2) + POW(L1.longitude - L2.longitude, 2)) \* 100), 2) AS totalDistance

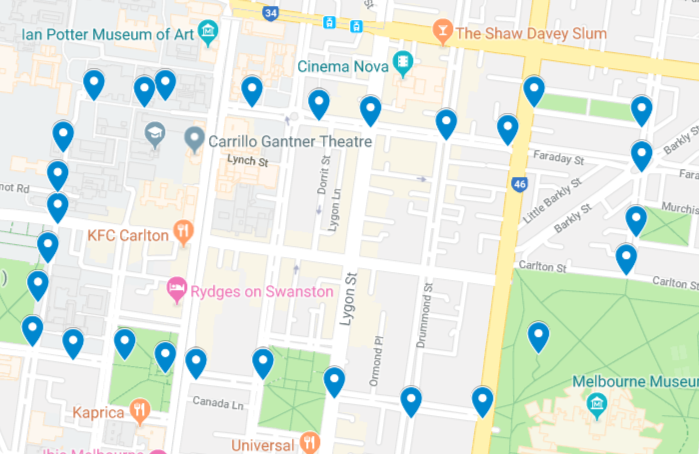
FROM User INNER JOIN Location L1 INNER JOIN Location L2

ON User.id = L1.user and User.id = L2.user and L2.id - L1.id = 2

WHERE User.name = 'Alice';

Question 10

1 Row Returned



**Figure 1 The total distance that Alice has run**