Comp 543 – Assignment 1

Question 1: Who has seen a flower at Alaska Flat?

Ans:

SELECT DISTINCT s.person

FROM sightings s

WHERE s.location = 'Alaska Flat'

一張含有 文字 的圖片

自動產生的描述

Question 2: Who has seen the same flower at both Moreland Mill and at Steve Spring?

Ans:

SELECT s.person

FROM sightings s

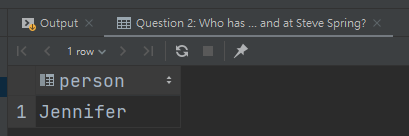
WHERE s.location = 'Moreland Mill' AND EXISTS (

SELECT s2.person

FROM sightings s2

WHERE s.name = s2.name AND s2.location = 'Steve Spring'

)



Question 3: What is the scientific name for each of the different flowers that have been sighted by either Michael or Robert above 8250 feet in elevation?

Ans:

SELECT DISTINCT fl.genus, fl.species

FROM flowers fl, sightings s, features f

WHERE fl.comname = s.name AND f.location = s.location AND (s.person = 'Michael' OR s.person = 'Robert') AND f.elev > 8250

一張含有 文字, 黑色, 計分板, 螢幕擷取畫面 的圖片

自動產生的描述

Question 4: Which maps hold a location where someone has seen Alpine penstemon in August?

Ans:

SELECT f.map

FROM features f

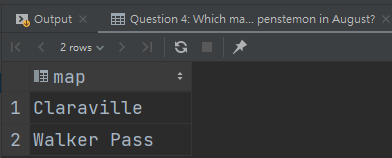
WHERE EXISTS (

SELECT s.location

FROM sightings s

WHERE s.location = f.location AND s.name = 'Alpine penstemon' AND MONTH(s.sighted) = 8

)



Question 5: Which genus have more than one species recorded in the SSWC database?

Ans:

SELECT DISTINCT fl.genus

FROM flowers fl

WHERE (SELECT COUNT(fl2.species)

FROM flowers fl2

WHERE fl.genus = fl2.genus) > 1

一張含有 文字, 計分板, 室外 的圖片

自動產生的描述

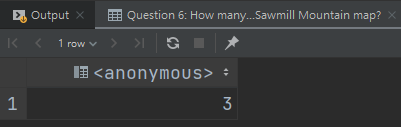
Question 6: How many summits are on the Sawmill Mountain map?

Ans:

SELECT COUNT(\*)

FROM features f

WHERE f.class = 'summit' AND f.map = 'Sawmill Mountain'



Question 7: What is the furthest south location that James has seen a flower? (“Furthest south” means lowest latitude)

Ans:

SELECT f.location

FROM features f

WHERE f.latitude = (

SELECT MIN(f2.latitude)

FROM features f2, sightings s

WHERE s.location = f2.location AND s.person = 'James'

)

一張含有 文字 的圖片

自動產生的描述

Question 8: Who has not seen a flower at a location of class Tower?

Ans:

SELECT DISTINCT s.person

FROM sightings s

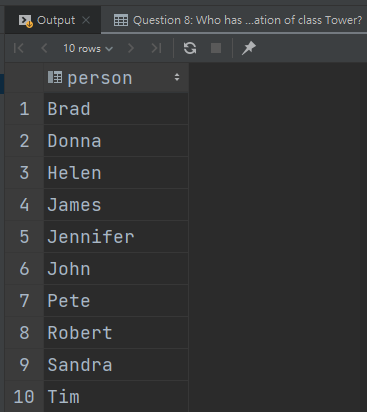
WHERE NOT EXISTS (

SELECT s2.person

FROM sightings s2, features f

WHERE s.person = s2.person AND f.location = s2.location AND f.class = 'Tower'

)



Question 9: Who has seen flowers at the most distinct locations, and how many flowers was that?

Ans:

CREATE VIEW UNILOC AS

SELECT s.person, COUNT(DISTINCT s.location) AS LOC\_NUM

FROM sightings s

GROUP BY s.person

SELECT s2.person, COUNT(DISTINCT s2.name)

FROM sightings s2

WHERE s2.person = (

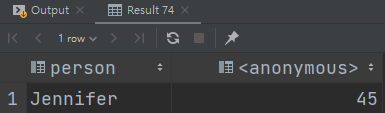
SELECT TOP(1) ul.person

FROM UNILOC ul

ORDER BY ul.LOC\_NUM DESC

)

GROUP BY s2.person



Question 10: For those people who have seen all of the flowers in the SSWC database, what was the date at which they saw their last unseen flower? In other words, at which date did they finish observing all of the flowers in the database?

Ans:

CREATE VIEW PEO AS

SELECT s.person, COUNT(DISTINCT s.name) AS FL\_CNT

FROM sightings s

GROUP BY s.person

SELECT p.person, MAX(s2.sighted)

FROM PEO p, sightings s2

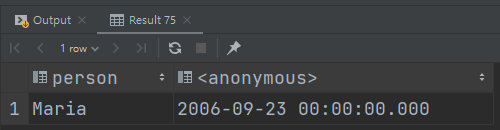
WHERE s2.person = p.person AND p.FL\_CNT = (

SELECT COUNT(fl.comname)

FROM flowers fl

)

GROUP BY p.person



Question 11: For Jennifer, compute the fraction of her sightings on a per-month basis. For example, we might get {(September, .12), (October, .74), (November, .14)}. The fractions should add up to one across all months.

Ans:

CREATE VIEW MON\_AVG AS

SELECT DATENAME(month, s.sighted) AS MON, COUNT(\*) AS MON\_CNT

FROM (SELECT s1.sighted

FROM sightings s1

WHERE s1.person = 'Jennifer') s

GROUP BY DATENAME(month, s.sighted)

SELECT ma.MON, CAST(ma.MON\_CNT AS FLOAT) / CAST(m.SUM\_CNT AS FLOAT)

FROM MON\_AVG ma, (

SELECT SUM(ma2.MON\_CNT) AS SUM\_CNT

FROM MON\_AVG ma2

) m

一張含有 文字 的圖片

自動產生的描述

Question 12: Whose set of flower sightings is most similar to John’s? Set similarity is here defined in terms of the Jaccard Index, where JI (A, B) for two sets A and B is (size of the intersection of A and B) / (size of the union of A and B). A larger Jaccard Index means more similar.

Ans:

CREATE VIEW JI\_INTERSEC AS

SELECT DISTINCT s.name, s.person

FROM sightings s

LEFT JOIN sightings s2 ON s.name = s2.name AND s2.person = 'John'

INTERSECT

SELECT DISTINCT s3.name, s3.person

FROM sightings s3

RIGHT JOIN sightings s5 ON s3.name = s5.name AND s3.person = 'John'

CREATE VIEW JI\_UNION AS

SELECT DISTINCT s.name, s.person

FROM (SELECT DISTINCT s1.name, s1.person

FROM sightings s1) s

LEFT JOIN sightings s2 ON s.name = s2.name AND s.person = 'John'

UNION

SELECT DISTINCT s3.name, s3.person

FROM (SELECT DISTINCT s4.name, s4.person

FROM sightings s4) s3

RIGHT JOIN sightings s5 ON s3.name = s5.name AND s3.person = 'John'

CREATE VIEW RESULT AS

SELECT ju.person AS ANS, CAST(COUNT(ji.name) AS FLOAT) / CAST(COUNT(ju.name) AS FLOAT) AS JACCARD\_INDEX

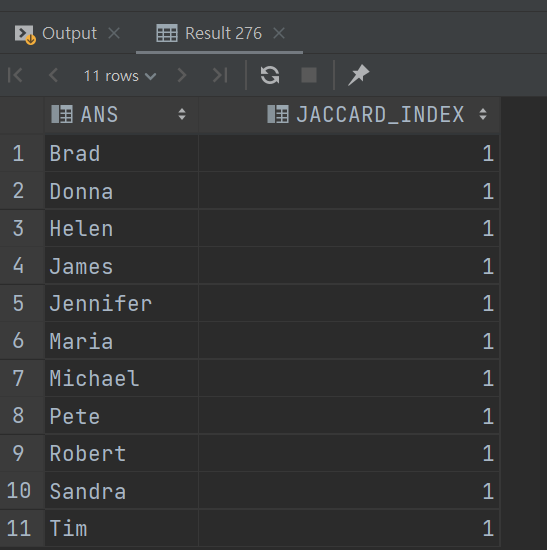
FROM JI\_INTERSEC ji, JI\_UNION ju

GROUP BY ju.person

SELECT r.ANS, r.JACCARD\_INDEX

FROM RESULT r

WHERE r.ANS <> 'John'



(Union seems to be correct but the intersection is wrong)