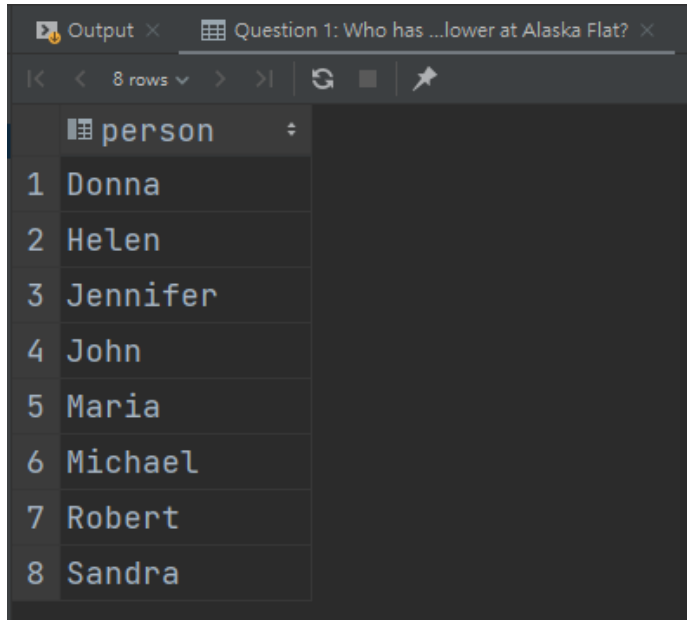


## Comp 543 – Assignment 1

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

Ans:

```
SELECT DISTINCT s.person
FROM sightings s
WHERE s.location = 'Alaska Flat'
```



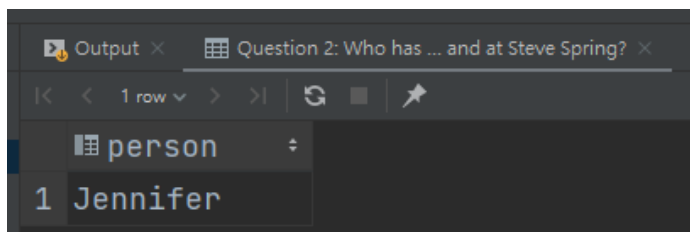
The screenshot shows a database query output window titled "Question 1: Who has ...lower at Alaska Flat?". The output is a table with one column labeled "person" and 8 rows of names. The names are: Donna, Helen, Jennifer, John, Maria, Michael, Robert, and Sandra.

person
1 Donna
2 Helen
3 Jennifer
4 John
5 Maria
6 Michael
7 Robert
8 Sandra

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'
)
```



The screenshot shows a database query output window titled "Question 2: Who has ... and at Steve Spring?". The output is a table with one column labeled "person" and 1 row of the name: Jennifer.

person
1 Jennifer

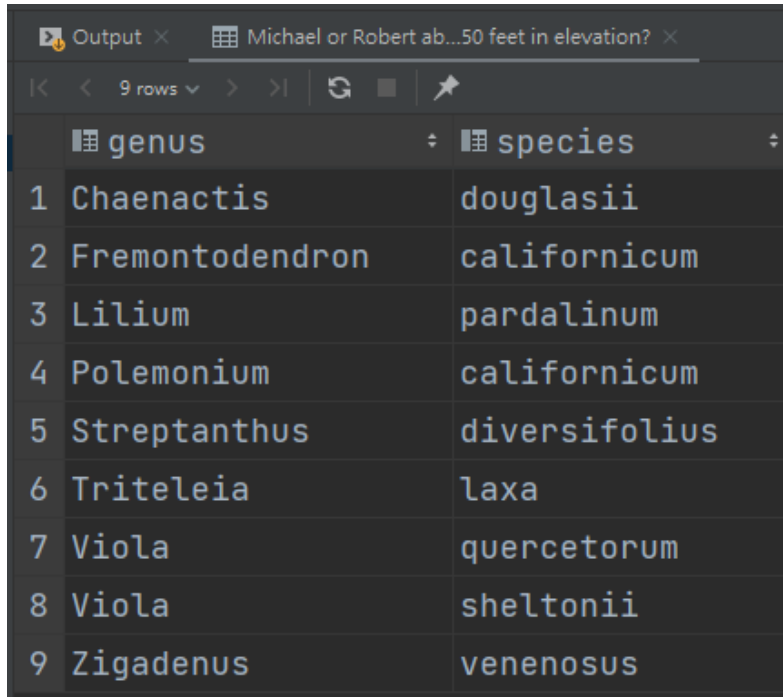
Question 3: What is the scientific name for each of the different flowers that have been sighted by either Michael or Robert above 8250 feet in elevation?

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

```



	genus	species
1	Chaenactis	douglasii
2	Fremontodendron	californicum
3	Lilium	pardalinum
4	Polemonium	californicum
5	Streptanthus	diversifolius
6	Triteleia	laxa
7	Viola	quercetorum
8	Viola	sheltonii
9	Zigadenus	venenosus

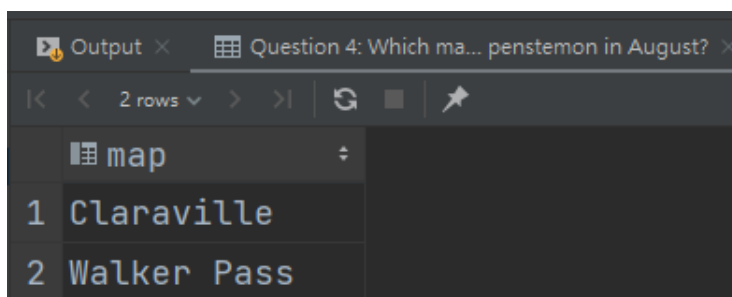
Question 4: Which maps hold a location where someone has seen Alpine penstemon in August?

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
)

```

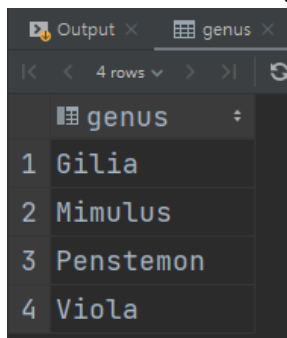


	map
1	Claraville
2	Walker Pass

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
```

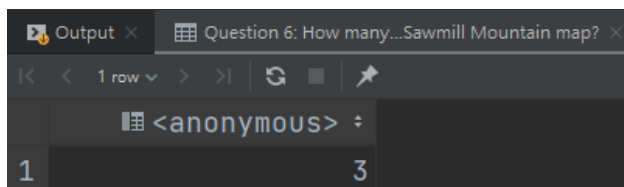


	genus
1	Gilia
2	Mimulus
3	Penstemon
4	Viola

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'
```

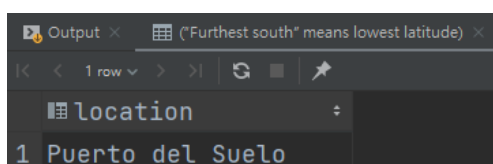


	<anonymous>
1	3

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'
)
```

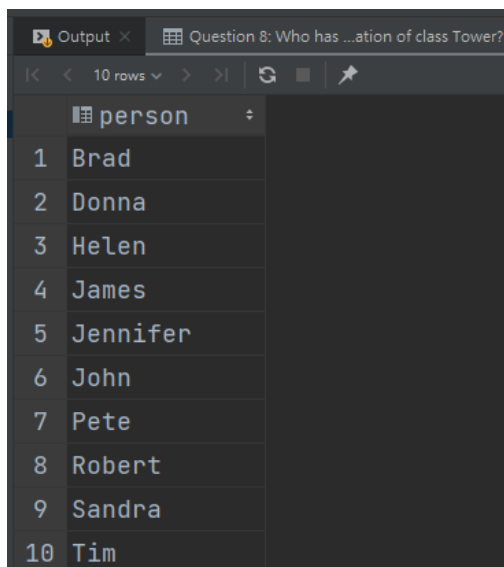


	location
1	Puerto del Suelo

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'
)
```



	person
1	Brad
2	Donna
3	Helen
4	James
5	Jennifer
6	John
7	Pete
8	Robert
9	Sandra
10	Tim

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
```

person	<anonymous>
1 Jennifer	45

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
```

person	<anonymous>
1 Maria	2006-09-23 00:00:00.000

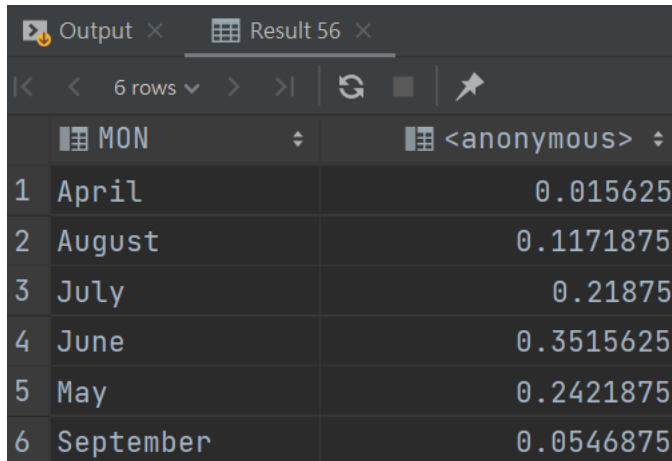
Question 11: For Jennifer, compute the fraction of her sightings on a per-month basis. For example, we might get {(September, .12), (October, .74), (November, .14)}. The fractions should add up to one across all months.

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
```



	MON	<anonymous>
1	April	0.015625
2	August	0.1171875
3	July	0.21875
4	June	0.3515625
5	May	0.2421875
6	September	0.0546875

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  $(\text{size of the intersection of A and B}) / (\text{size of the union of A and B})$ . A larger Jaccard Index means more similar.

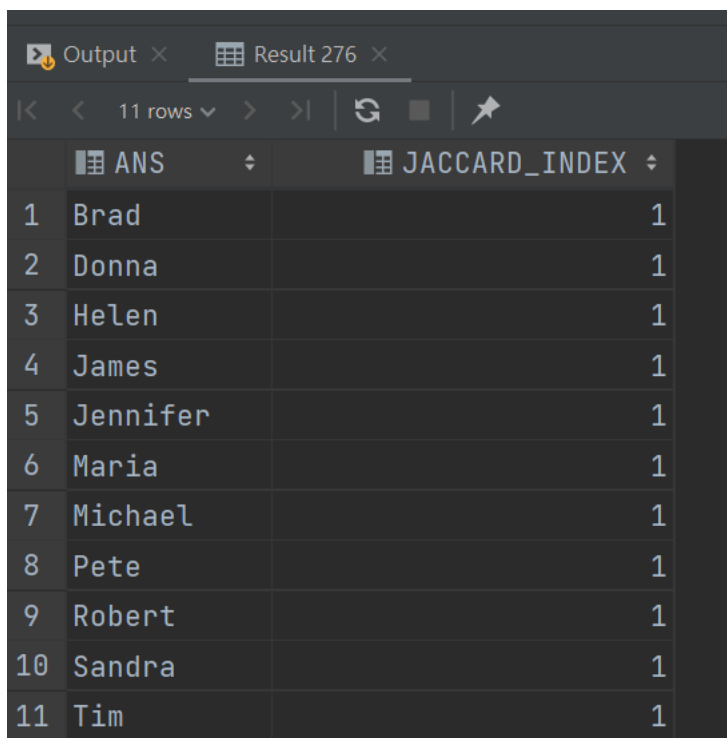
Ans:

```
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'
```



	ANS	JACCARD_INDEX
1	Brad	1
2	Donna	1
3	Helen	1
4	James	1
5	Jennifer	1
6	Maria	1
7	Michael	1
8	Pete	1
9	Robert	1
10	Sandra	1
11	Tim	1

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