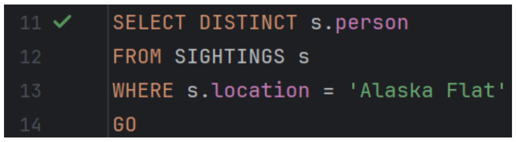
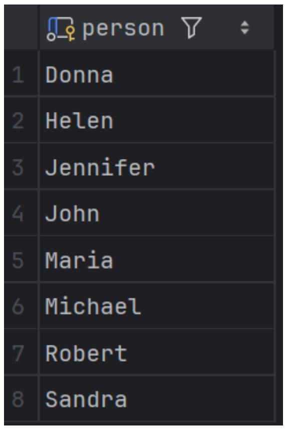
Assignment #1: Declarative SQL Programming

Yu-Tzu Chang (yc222)

1. Who has seen a flower at Alaska Flat?

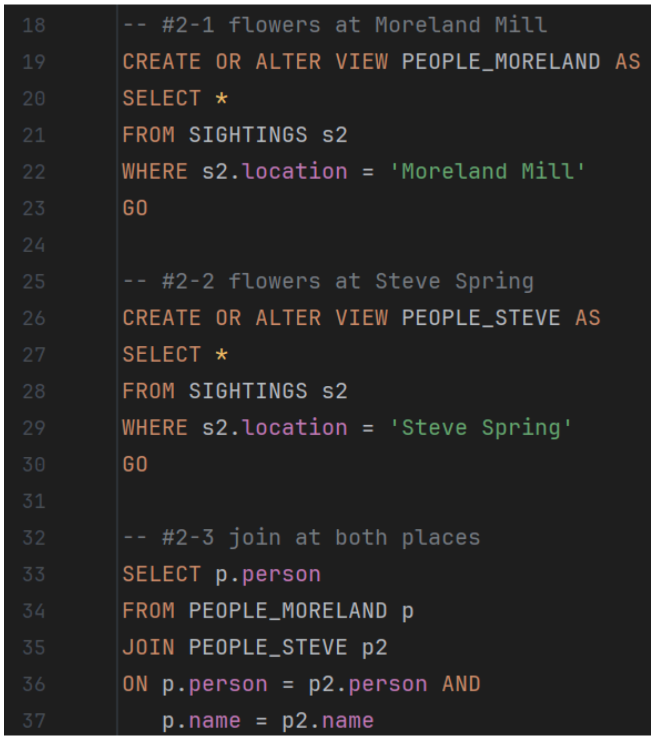


```
SELECT DISTINCT s.person
FROM SIGHTINGS s
WHERE s.location = 'Alaska Flat'
GO
```

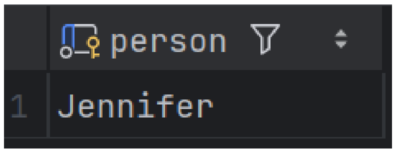


	person
1	Donna
2	Helen
3	Jennifer
4	John
5	Maria
6	Michael
7	Robert
8	Sandra

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

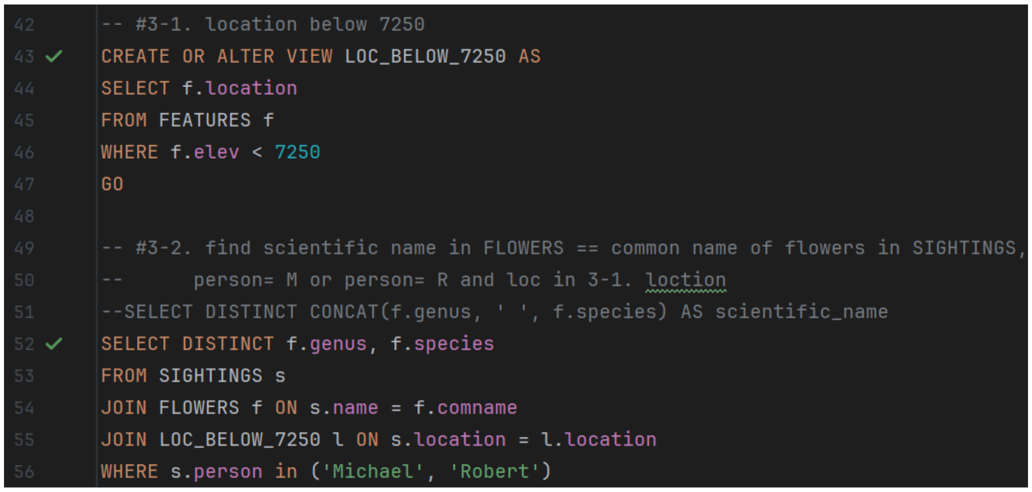


```
-- #2-1 flowers at Moreland Mill
CREATE OR ALTER VIEW PEOPLE_MORELAND AS
SELECT *
FROM SIGHTINGS s2
WHERE s2.location = 'Moreland Mill'
GO

-- #2-2 flowers at Steve Spring
CREATE OR ALTER VIEW PEOPLE_STEVE AS
SELECT *
FROM SIGHTINGS s2
WHERE s2.location = 'Steve Spring'
GO

-- #2-3 join at both places
SELECT p.person
FROM PEOPLE_MORELAND p
JOIN PEOPLE_STEVE p2
ON p.person = p2.person AND
   p.name = p2.name
```

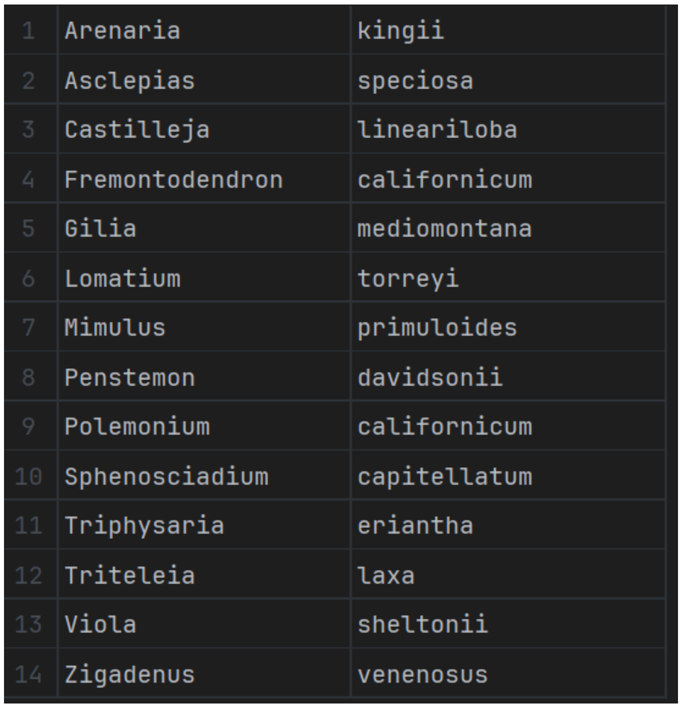


```
person
Jennifer
```

3. What is the scientific name for each of the different flowers that have been sighted by either Michael or Robert below 7250 feet in elevation?

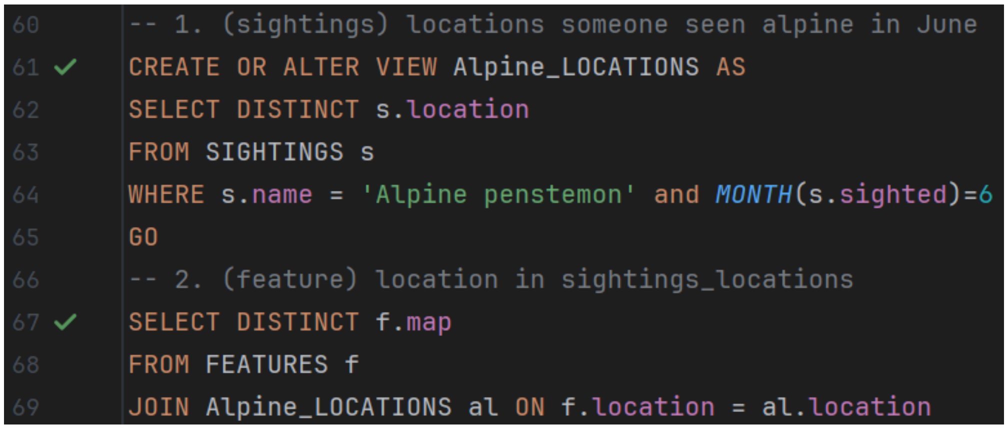


```sql
-- #3-1. location below 7250
CREATE OR ALTER VIEW LOC_BELOW_7250 AS
SELECT f.location
FROM FEATURES f
WHERE f.elev < 7250
GO

-- #3-2. find scientific name in FLOWERS == common name of flowers in SIGHTINGS,
--       person= M or person= R and loc in 3-1. loction
--SELECT DISTINCT CONCAT(f.genus, ' ', f.species) AS scientific_name
SELECT DISTINCT f.genus, f.species
FROM SIGHTINGS s
JOIN FLOWERS f ON s.name = f.comname
JOIN LOC_BELOW_7250 l ON s.location = l.location
WHERE s.person in ('Michael', 'Robert')
```

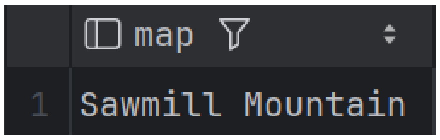


1	Arenaria	kingii
2	Asclepias	speciosa
3	Castilleja	lineariloba
4	Fremontodendron	californicum
5	Gilia	mediomontana
6	Lomatium	torreyi
7	Mimulus	primuloides
8	Penstemon	davidsonii
9	Polemonium	californicum
10	Sphenosciadium	capitellatum
11	Triphysaria	eriantha
12	Triteleia	laxa
13	Viola	sheltonii
14	Zigadenus	venenosus

4. Which maps hold a location where someone has seen Alpine penstemon in June?

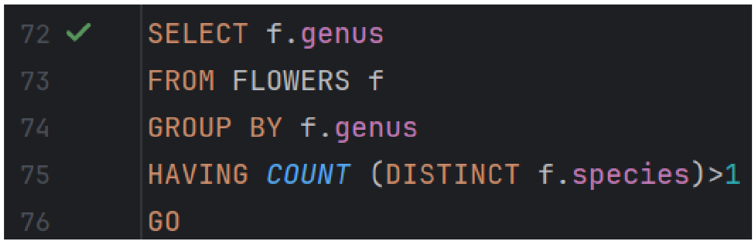


```sql
-- 1. (sightings) locations someone seen alpine in June
CREATE OR ALTER VIEW Alpine_LOCATIONS AS
SELECT DISTINCT s.location
FROM SIGHTINGS s
WHERE s.name = 'Alpine penstemon' and MONTH(s.sighted)=6
GO
-- 2. (feature) location in sightings_locations
SELECT DISTINCT f.map
FROM FEATURES f
JOIN Alpine_LOCATIONS al ON f.location = al.location
```

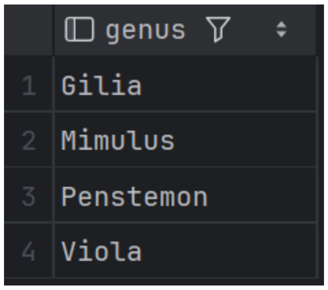


	map
1	Sawmill Mountain

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

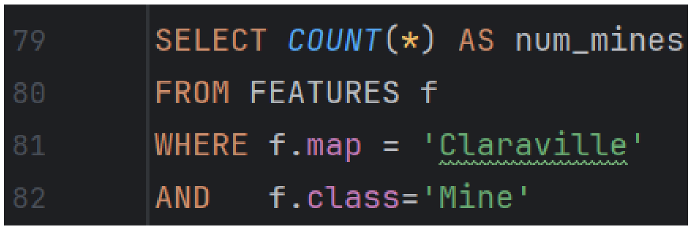


```sql
SELECT f.genus
FROM FLOWERS f
GROUP BY f.genus
HAVING COUNT (DISTINCT f.species)>1
GO
```

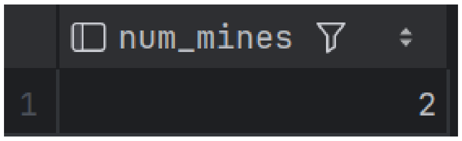


	genus
1	Gilia
2	Mimulus
3	Penstemon
4	Viola

6. How many mines are on the Claraville map?

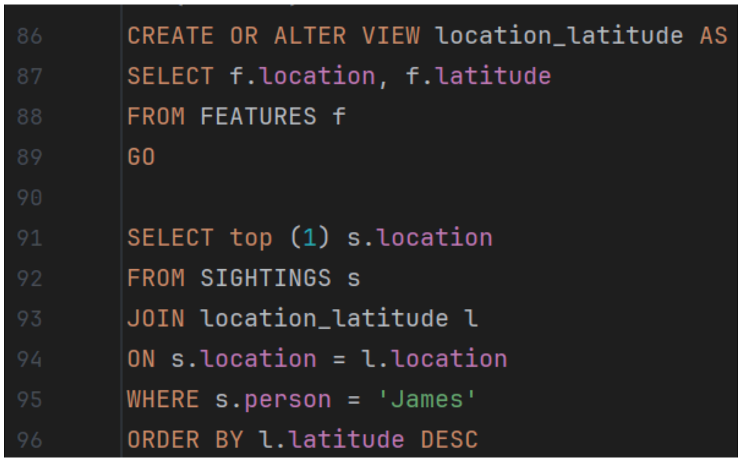


```sql
SELECT COUNT(*) AS num_mines
FROM FEATURES f
WHERE f.map = 'Claraville'
AND   f.class='Mine'
```

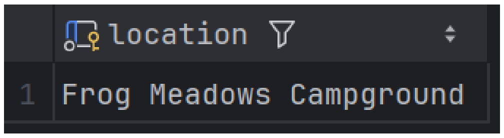


num_mines
2

7. What is the furthest north location that James has seen a flower? "Furthest north" means highest latitude.

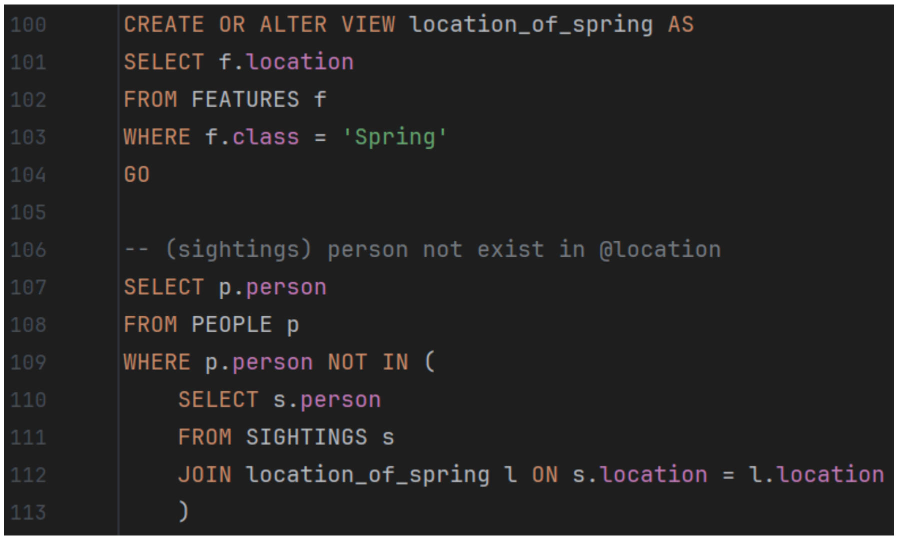


```sql
CREATE OR ALTER VIEW location_latitude AS
SELECT f.location, f.latitude
FROM FEATURES f
GO

SELECT top (1) s.location
FROM SIGHTINGS s
JOIN location_latitude l
ON s.location = l.location
WHERE s.person = 'James'
ORDER BY l.latitude DESC
```

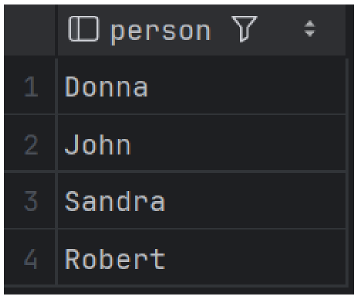


location
Frog Meadows Campground

8. Who has not seen a flower at a location of class Spring?

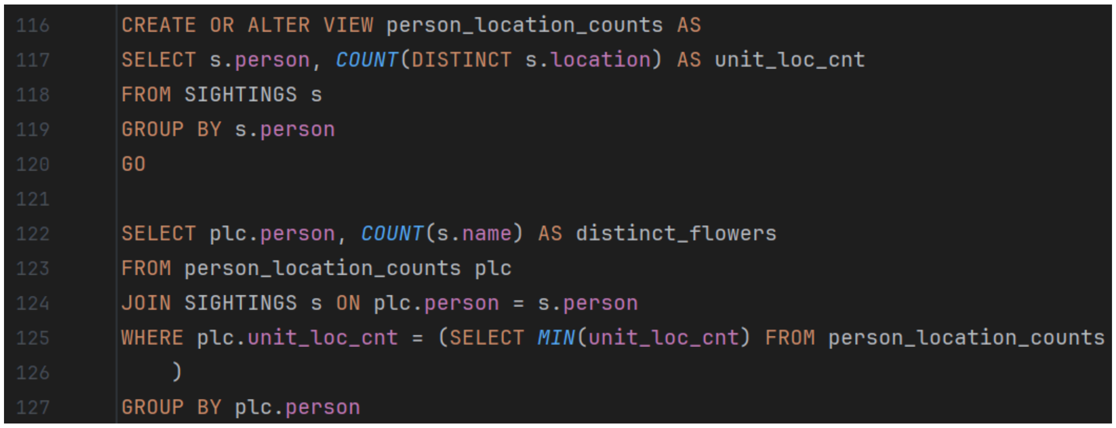


```sql
CREATE OR ALTER VIEW location_of_spring AS
SELECT f.location
FROM FEATURES f
WHERE f.class = 'Spring'
GO

-- (sightings) person not exist in @location
SELECT p.person
FROM PEOPLE p
WHERE p.person NOT IN (
    SELECT s.person
    FROM SIGHTINGS s
    JOIN location_of_spring l ON s.location = l.location
    )
```

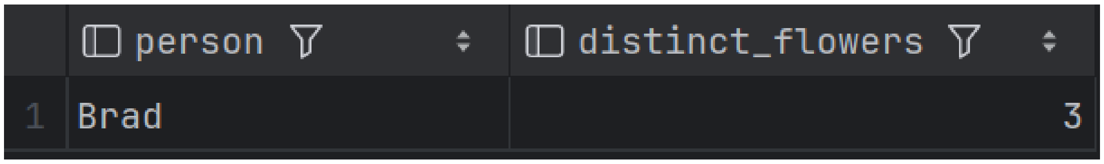


	person
1	Donna
2	John
3	Sandra
4	Robert

9. Who has seen flowers at the least distinct locations, and how many distinct flowers was that?

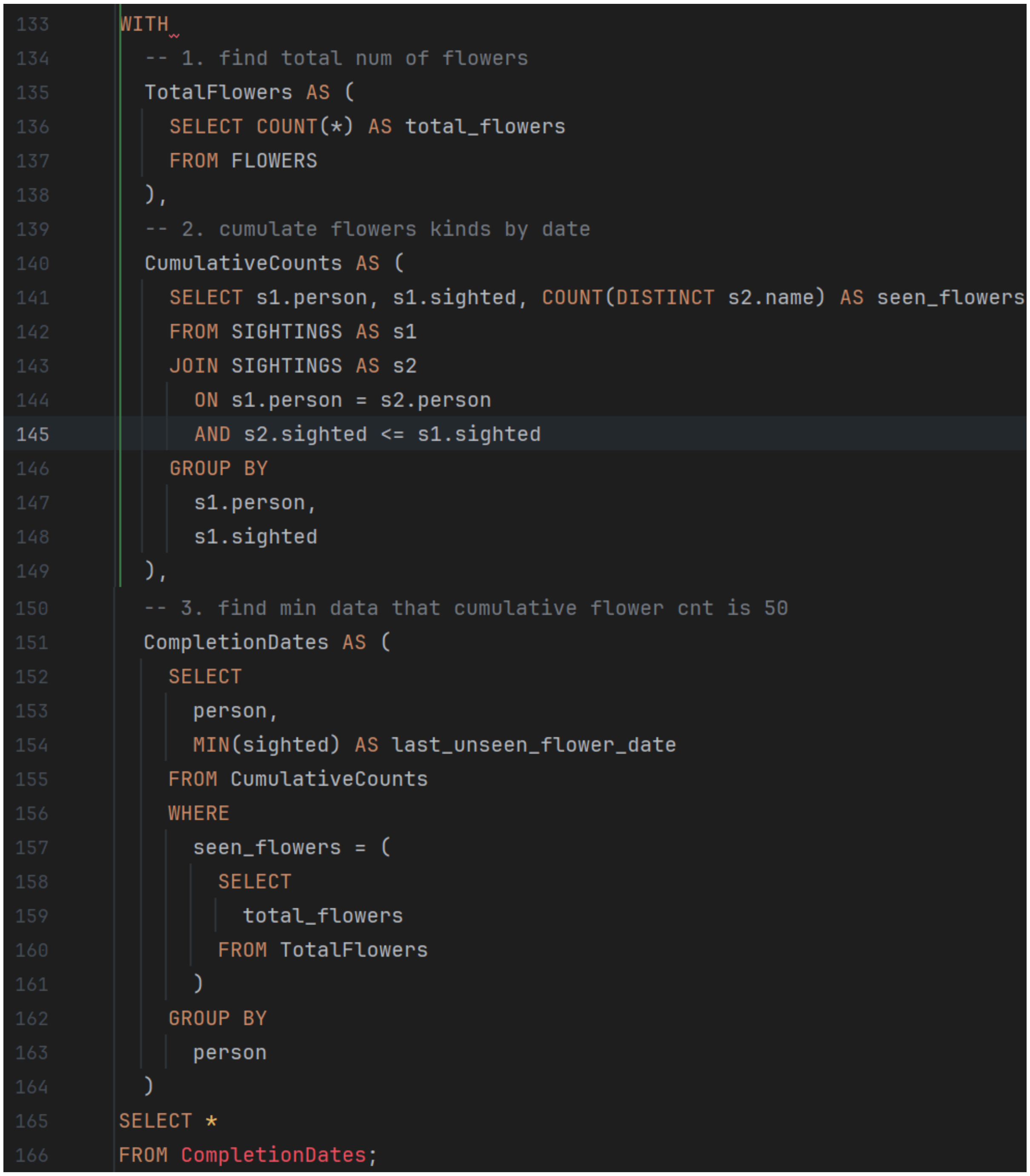


```
CREATE OR ALTER VIEW person_location_counts AS
SELECT s.person, COUNT(DISTINCT s.location) AS unit_loc_cnt
FROM SIGHTINGS s
GROUP BY s.person
GO

SELECT plc.person, COUNT(s.name) AS distinct_flowers
FROM person_location_counts plc
JOIN SIGHTINGS s ON plc.person = s.person
WHERE plc.unit_loc_cnt = (SELECT MIN(unit_loc_cnt) FROM person_location_counts
    )
GROUP BY plc.person
```

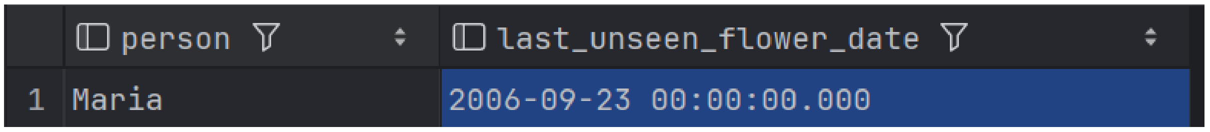


	person	distinct_flowers
1	Brad	3

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?



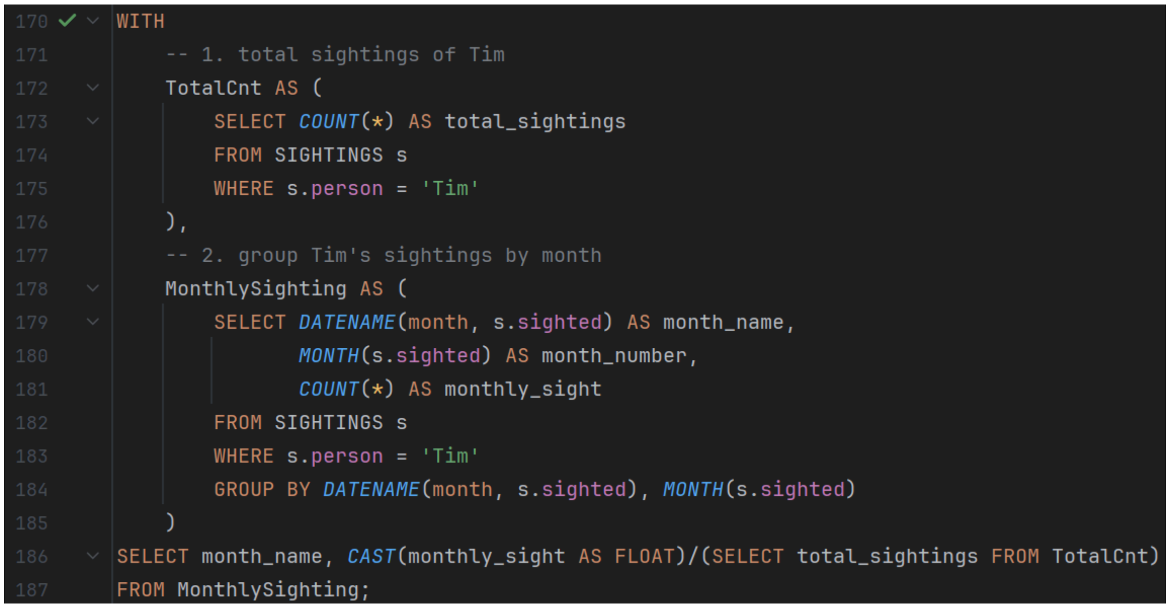
```sql
WITH
  -- 1. find total num of flowers
  TotalFlowers AS (
    SELECT COUNT(*) AS total_flowers
    FROM FLOWERS
  ),
  -- 2. cumulate flowers kinds by date
  CumulativeCounts AS (
    SELECT s1.person, s1.sighted, COUNT(DISTINCT s2.name) AS seen_flowers
    FROM SIGHTINGS AS s1
    JOIN SIGHTINGS AS s2
      ON s1.person = s2.person
      AND s2.sighted <= s1.sighted
    GROUP BY
      s1.person,
      s1.sighted
  ),
  -- 3. find min data that cumulative flower cnt is 50
  CompletionDates AS (
    SELECT
      person,
      MIN(sighted) AS last_unseen_flower_date
    FROM CumulativeCounts
    WHERE
      seen_flowers = (
        SELECT
          total_flowers
        FROM TotalFlowers
      )
    GROUP BY
      person
  )
SELECT *
FROM CompletionDates;
```



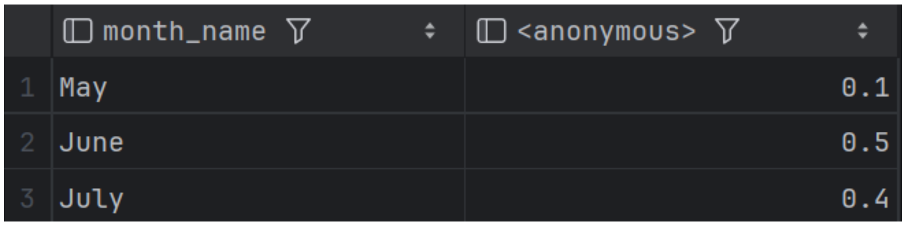
	person	last_unseen_flower_date
1	Maria	2006-09-23 00:00:00.000

11. For Tim, compute the fraction of his 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.

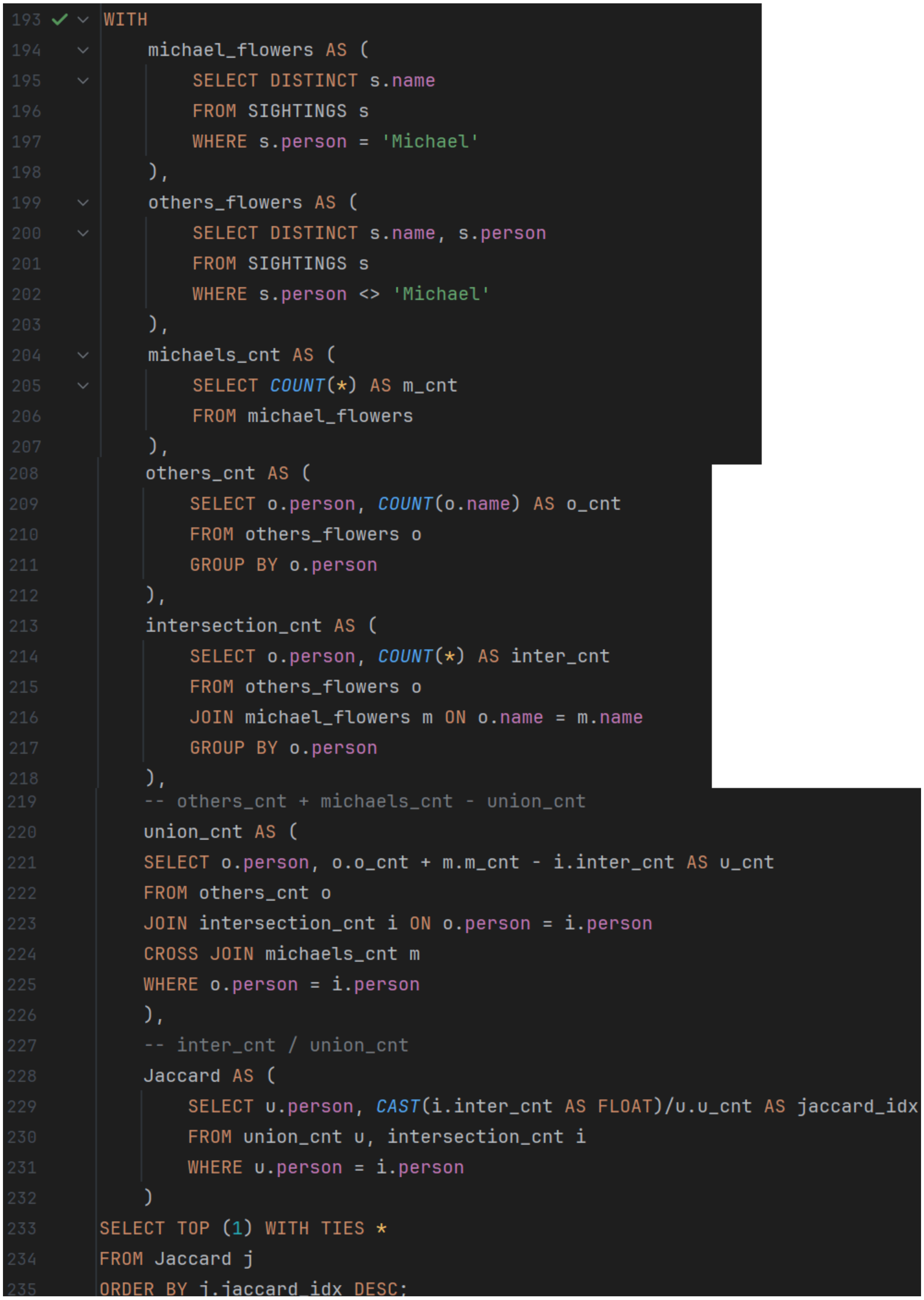


```sql
WITH
    -- 1. total sightings of Tim
    TotalCnt AS (
        SELECT COUNT(*) AS total_sightings
        FROM SIGHTINGS s
        WHERE s.person = 'Tim'
    ),
    -- 2. group Tim's sightings by month
    MonthlySighting AS (
        SELECT DATENAME(month, s.sighted) AS month_name,
               MONTH(s.sighted) AS month_number,
               COUNT(*) AS monthly_sight
        FROM SIGHTINGS s
        WHERE s.person = 'Tim'
        GROUP BY DATENAME(month, s.sighted), MONTH(s.sighted)
    )
SELECT month_name, CAST(monthly_sight AS FLOAT)/(SELECT total_sightings FROM TotalCnt)
FROM MonthlySighting;
```

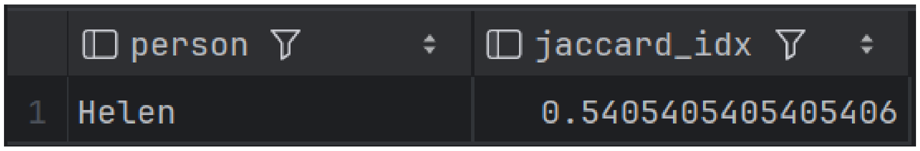


	month_name	<anonymous>
1	May	0.1
2	June	0.5
3	July	0.4

12. Whose set of flower sightings is most similar to Michael'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.



```sql
WITH
    michael_flowers AS (
        SELECT DISTINCT s.name
        FROM SIGHTINGS s
        WHERE s.person = 'Michael'
    ),
    others_flowers AS (
        SELECT DISTINCT s.name, s.person
        FROM SIGHTINGS s
        WHERE s.person <> 'Michael'
    ),
    michaels_cnt AS (
        SELECT COUNT(*) AS m_cnt
        FROM michael_flowers
    ),
    others_cnt AS (
        SELECT o.person, COUNT(o.name) AS o_cnt
        FROM others_flowers o
        GROUP BY o.person
    ),
    intersection_cnt AS (
        SELECT o.person, COUNT(*) AS inter_cnt
        FROM others_flowers o
        JOIN michael_flowers m ON o.name = m.name
        GROUP BY o.person
    ),
    -- others_cnt + michaels_cnt - union_cnt
    union_cnt AS (
    SELECT o.person, o.o_cnt + m.m_cnt - i.inter_cnt AS u_cnt
    FROM others_cnt o
    JOIN intersection_cnt i ON o.person = i.person
    CROSS JOIN michaels_cnt m
    WHERE o.person = i.person
    ),
    -- inter_cnt / union_cnt
    Jaccard AS (
        SELECT u.person, CAST(i.inter_cnt AS FLOAT)/u.u_cnt AS jaccard_idx
        FROM union_cnt u, intersection_cnt i
        WHERE u.person = i.person
    )
SELECT TOP (1) WITH TIES *
FROM Jaccard j
ORDER BY j.jaccard_idx DESC;
```



	person	jaccard_idx
1	Helen	0.5405405405405406