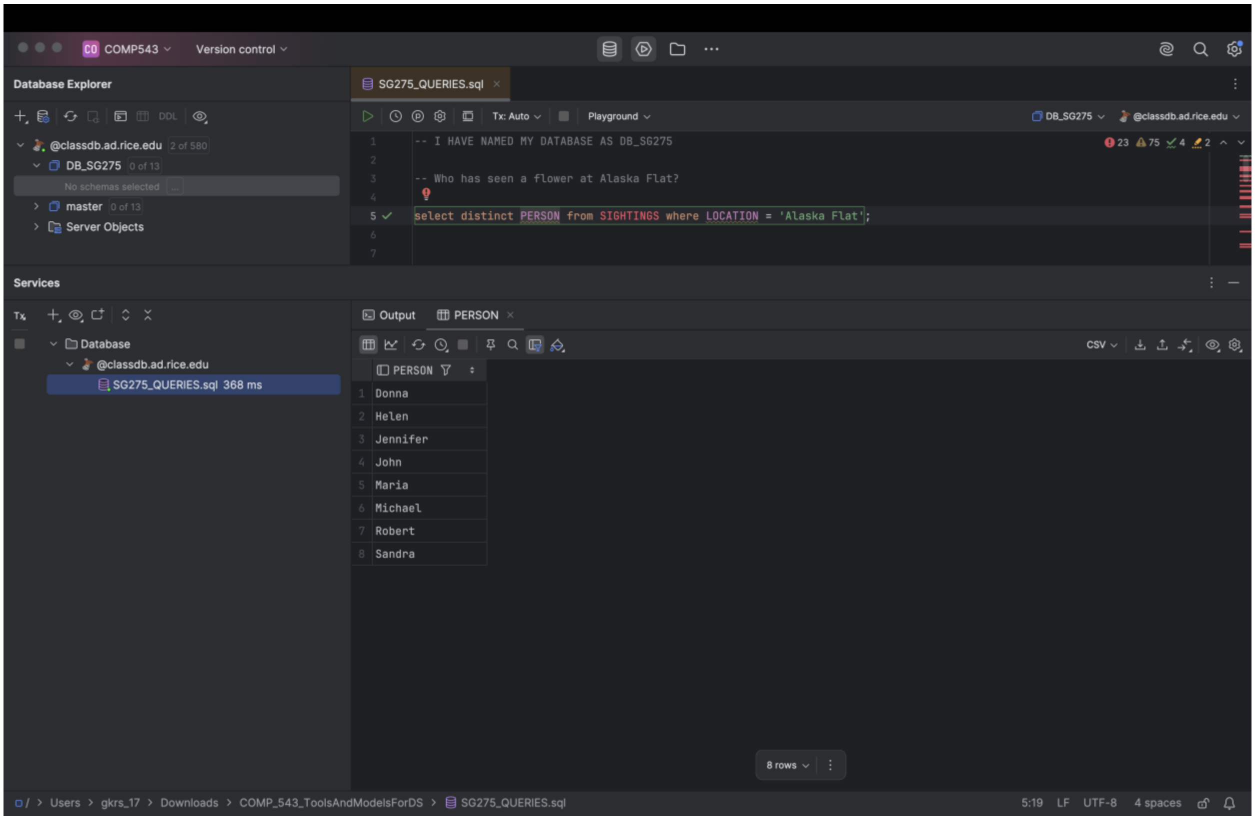
```sql
-- Who has seen a flower at Alaska Flat?

SELECT DISTINCT PERSON FROM SIGHTINGS WHERE LOCATION = 'Alaska Flat';
```

Output

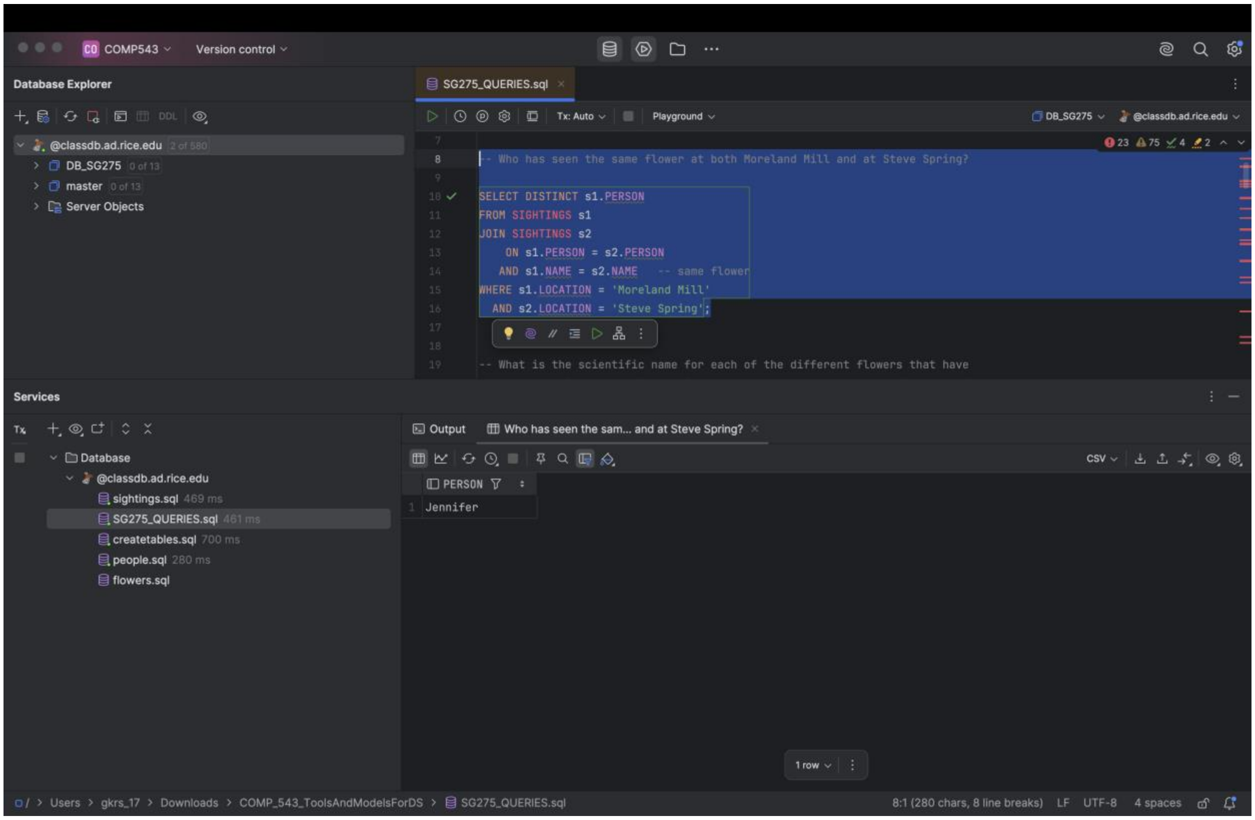


```sql
-- I HAVE NAMED MY DATABASE AS DB_SG275

-- Who has seen a flower at Alaska Flat?

select distinct PERSON from SIGHTINGS where LOCATION = 'Alaska Flat';
```

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

```sql
-- Who has seen the same flower at both Moreland Mill and at Steve Spring?

SELECT DISTINCT s1.PERSON
FROM SIGHTINGS s1
JOIN SIGHTINGS s2
    ON s1.PERSON = s2.PERSON
   AND s1.NAME = s2.NAME
WHERE s1.LOCATION = 'Moreland Mill'
  AND s2.LOCATION = 'Steve Spring';
```

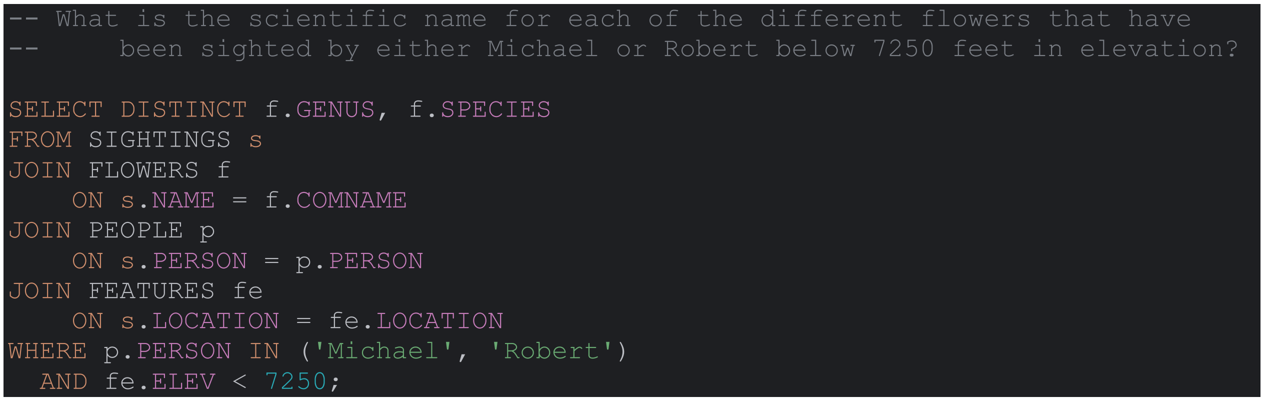
Output



```sql
-- Who has seen the same flower at both Moreland Mill and at Steve Spring?

SELECT DISTINCT s1.PERSON
FROM SIGHTINGS s1
JOIN SIGHTINGS s2
    ON s1.PERSON = s2.PERSON
   AND s1.NAME = s2.NAME   -- same flower
WHERE s1.LOCATION = 'Moreland Mill'
  AND s2.LOCATION = 'Steve Spring';


-- What is the scientific name for each of the different flowers that have
```

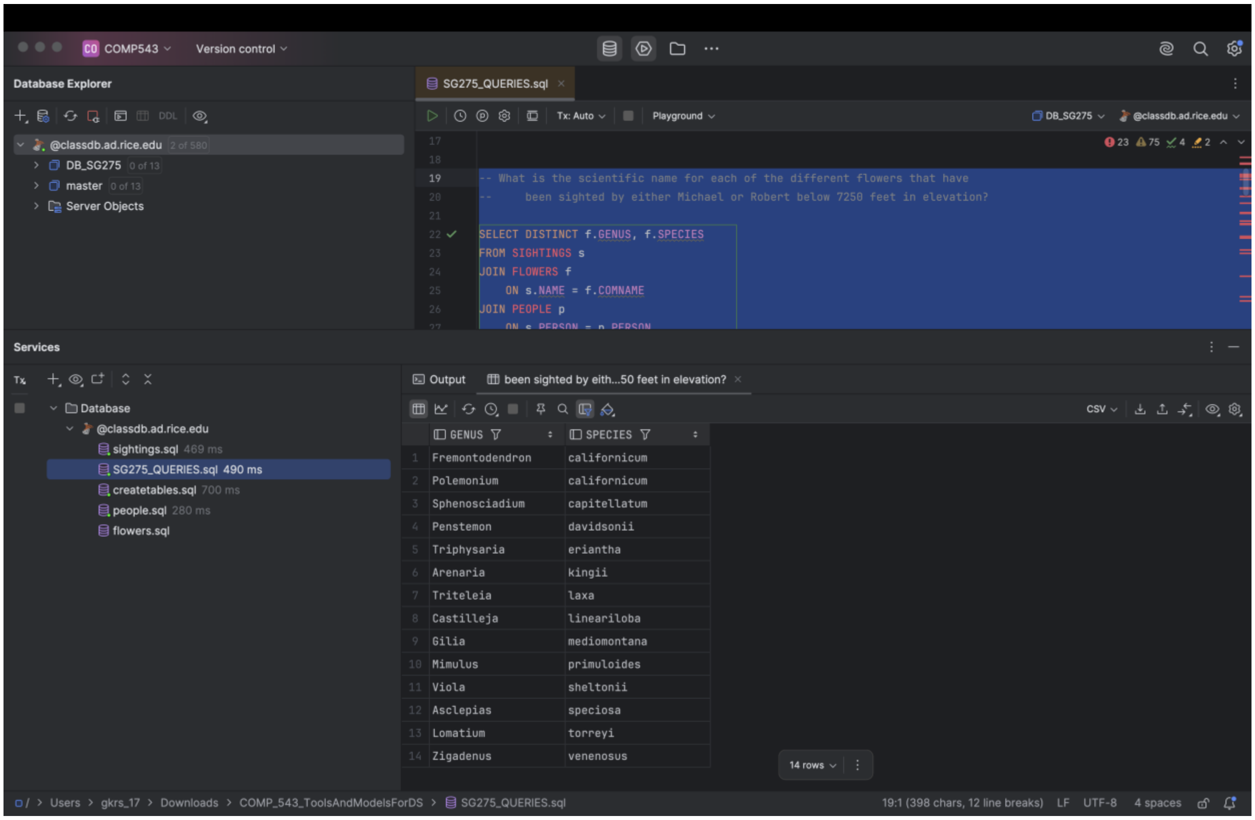
	PERSON
1	Jennifer

1 row



```sql
-- 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?

SELECT DISTINCT f.GENUS, f.SPECIES
FROM SIGHTINGS s
JOIN FLOWERS f
    ON s.NAME = f.COMNAME
JOIN PEOPLE p
    ON s.PERSON = p.PERSON
JOIN FEATURES fe
    ON s.LOCATION = fe.LOCATION
WHERE p.PERSON IN ('Michael', 'Robert')
  AND fe.ELEV < 7250;
```

Output

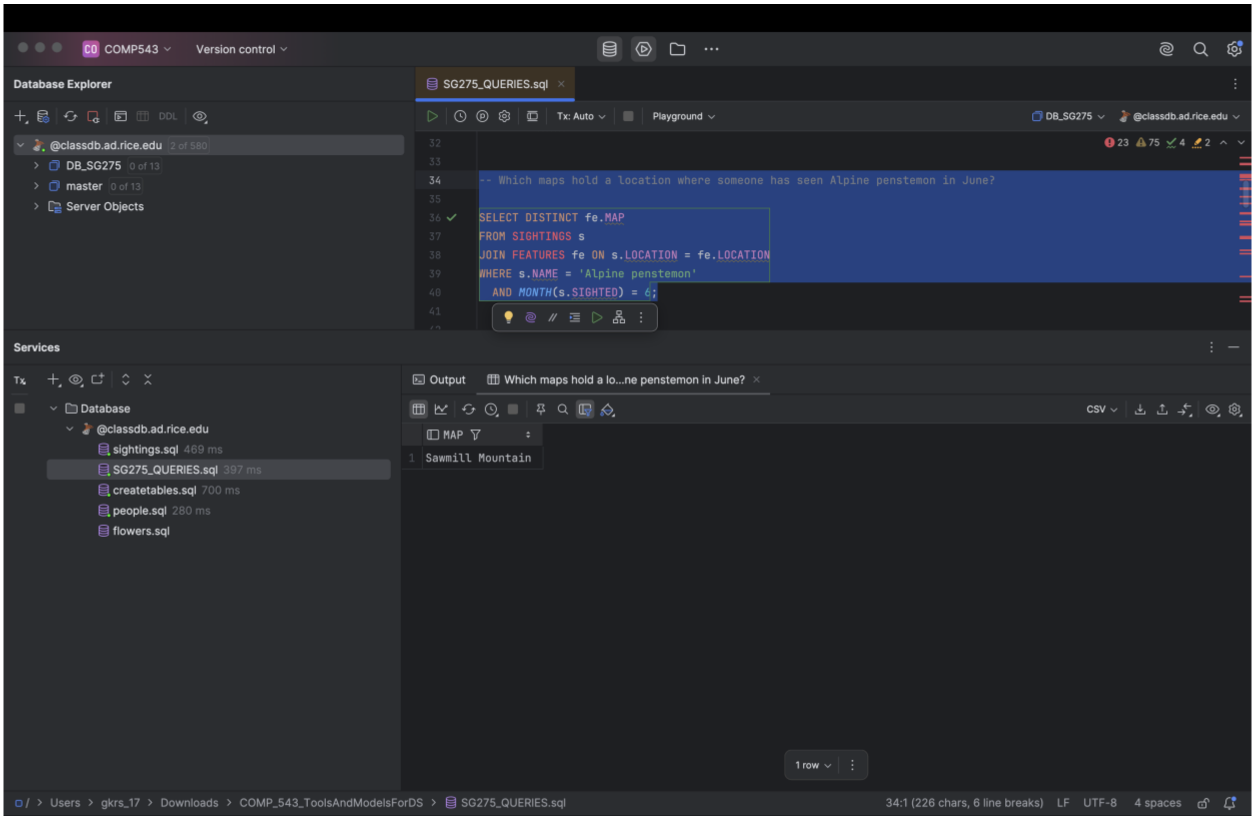


```sql
-- What is the scientific name for each of the different flowers that have
--     been sighted by either Michael or Robert below 7250 feet in elevation?

SELECT DISTINCT f.GENUS, f.SPECIES
FROM SIGHTINGS s
JOIN FLOWERS f
    ON s.NAME = f.COMNAME
JOIN PEOPLE p
    ON s.PERSON = p.PERSON
```

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

```sql
-- Which maps hold a location where someone has seen Alpine penstemon in
June?

SELECT DISTINCT fe.MAP
FROM SIGHTINGS s
JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
WHERE s.NAME = 'Alpine penstemon'
  AND MONTH(s.SIGHTED) = 6;
```

Output

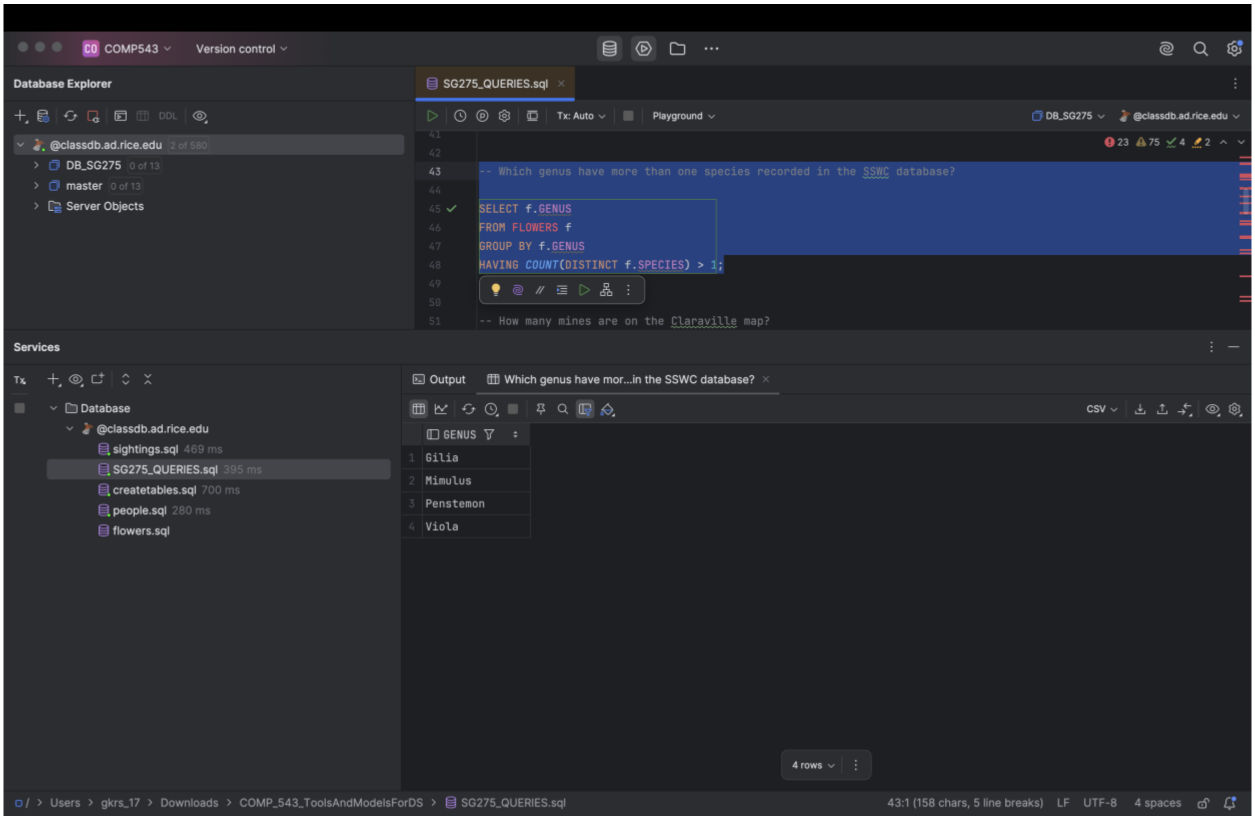


```sql
-- Which maps hold a location where someone has seen Alpine penstemon in June?

SELECT DISTINCT fe.MAP
FROM SIGHTINGS s
JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
WHERE s.NAME = 'Alpine penstemon'
  AND MONTH(s.SIGHTED) = 6;
```

```
  MAP
1 Sawmill Mountain
```

```sql
-- Which genus have more than one species recorded in the SSWC database?

SELECT f.GENUS
FROM FLOWERS f
GROUP BY f.GENUS
HAVING COUNT(DISTINCT f.SPECIES) > 1;
```

Output

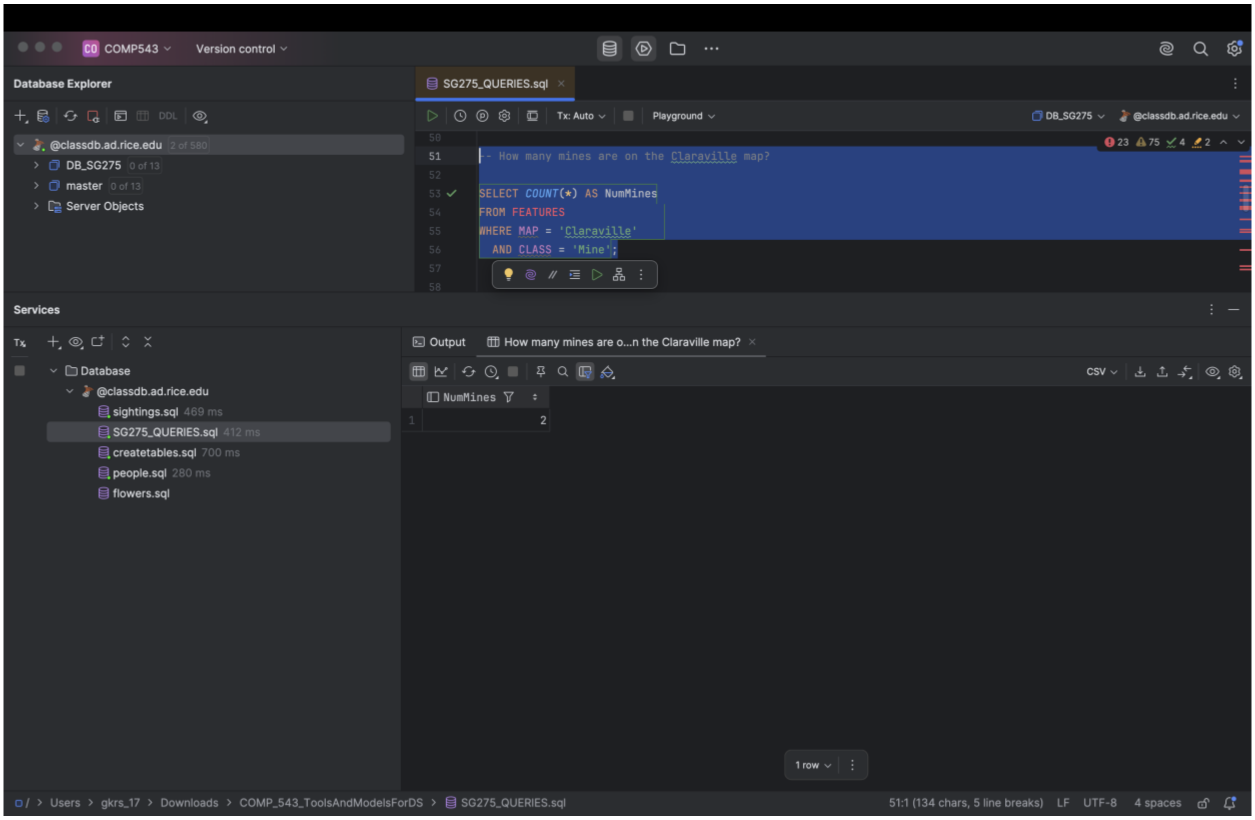


```sql
-- Which genus have more than one species recorded in the SSWC database?

SELECT f.GENUS
FROM FLOWERS f
GROUP BY f.GENUS
HAVING COUNT(DISTINCT f.SPECIES) > 1;


-- How many mines are on the Claraville map?
```

	GENUS
1	Gilia
2	Mimulus
3	Penstemon
4	Viola

```sql
-- How many mines are on the Claraville map?

SELECT COUNT(*) AS NumMines
FROM FEATURES
WHERE MAP = 'Claraville'
  AND CLASS = 'Mine';
```

Output

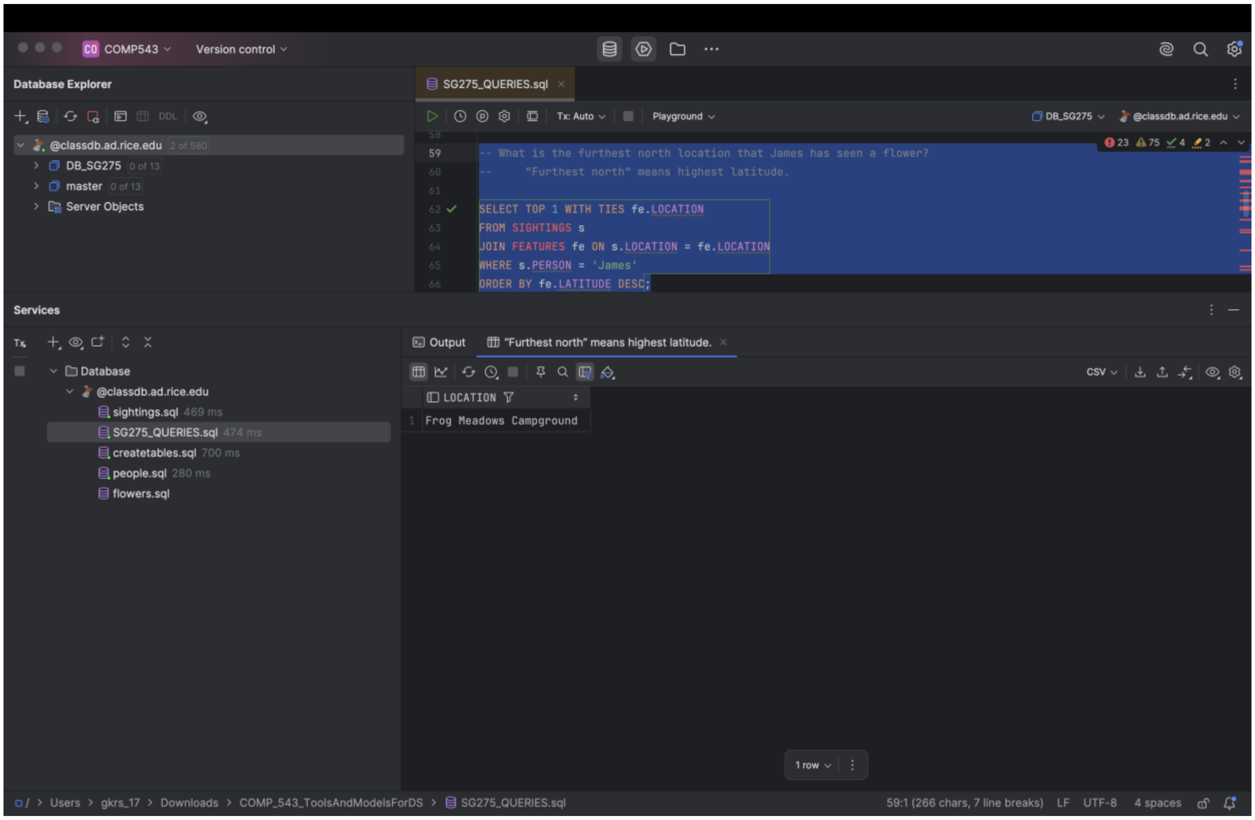


```sql
-- How many mines are on the Claraville map?

SELECT COUNT(*) AS NumMines
FROM FEATURES
WHERE MAP = 'Claraville'
  AND CLASS = 'Mine';
```

```
  NumMines
1        2
```

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

SELECT TOP 1 WITH TIES fe.LOCATION
FROM SIGHTINGS s
JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
WHERE s.PERSON = 'James'
ORDER BY fe.LATITUDE DESC;
```

Output

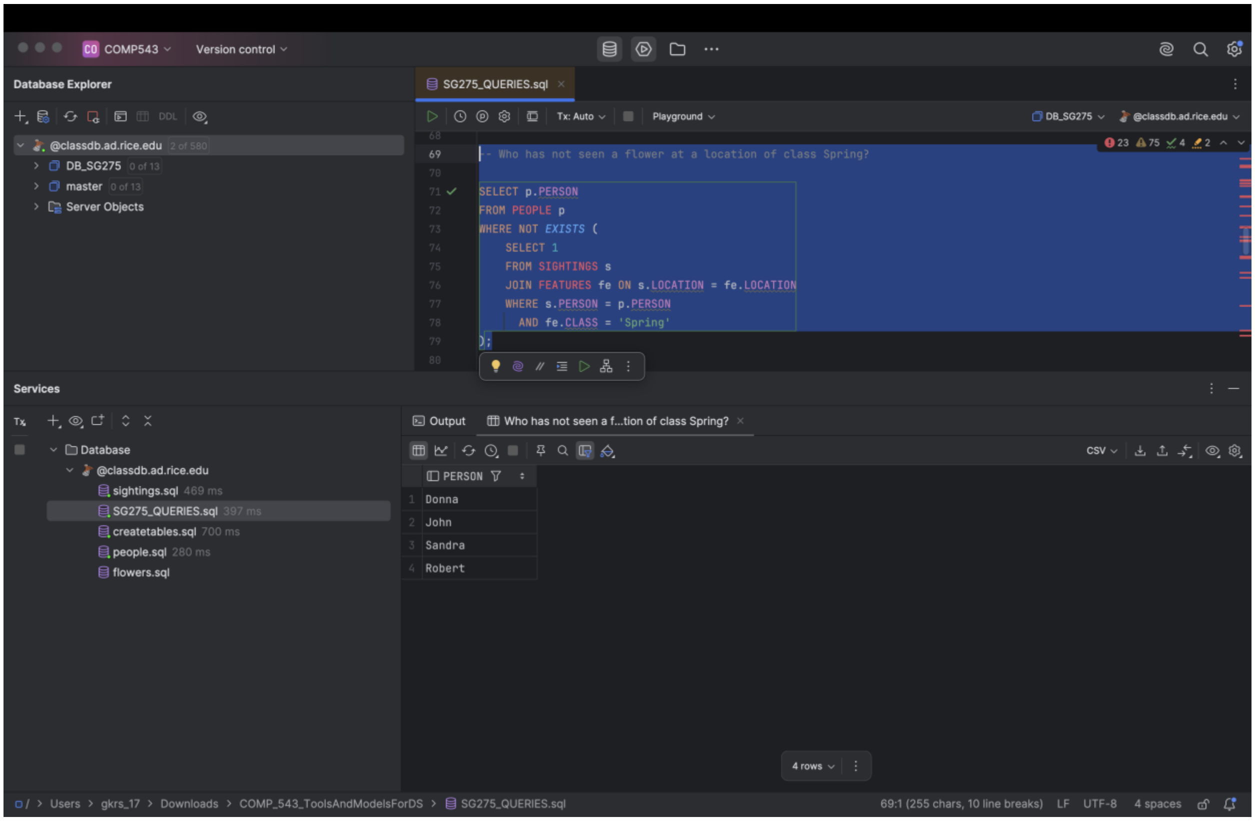


```
-- What is the furthest north location that James has seen a flower?
--     "Furthest north" means highest latitude.

SELECT TOP 1 WITH TIES fe.LOCATION
FROM SIGHTINGS s
JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
WHERE s.PERSON = 'James'
ORDER BY fe.LATITUDE DESC;
```

```
  LOCATION
1 Frog Meadows Campground
```

```sql
-- Who has not seen a flower at a location of class Spring?

SELECT p.PERSON
FROM PEOPLE p
WHERE NOT EXISTS (
    SELECT 1
    FROM SIGHTINGS s
    JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
    WHERE s.PERSON = p.PERSON
      AND fe.CLASS = 'Spring'
);
```

Output



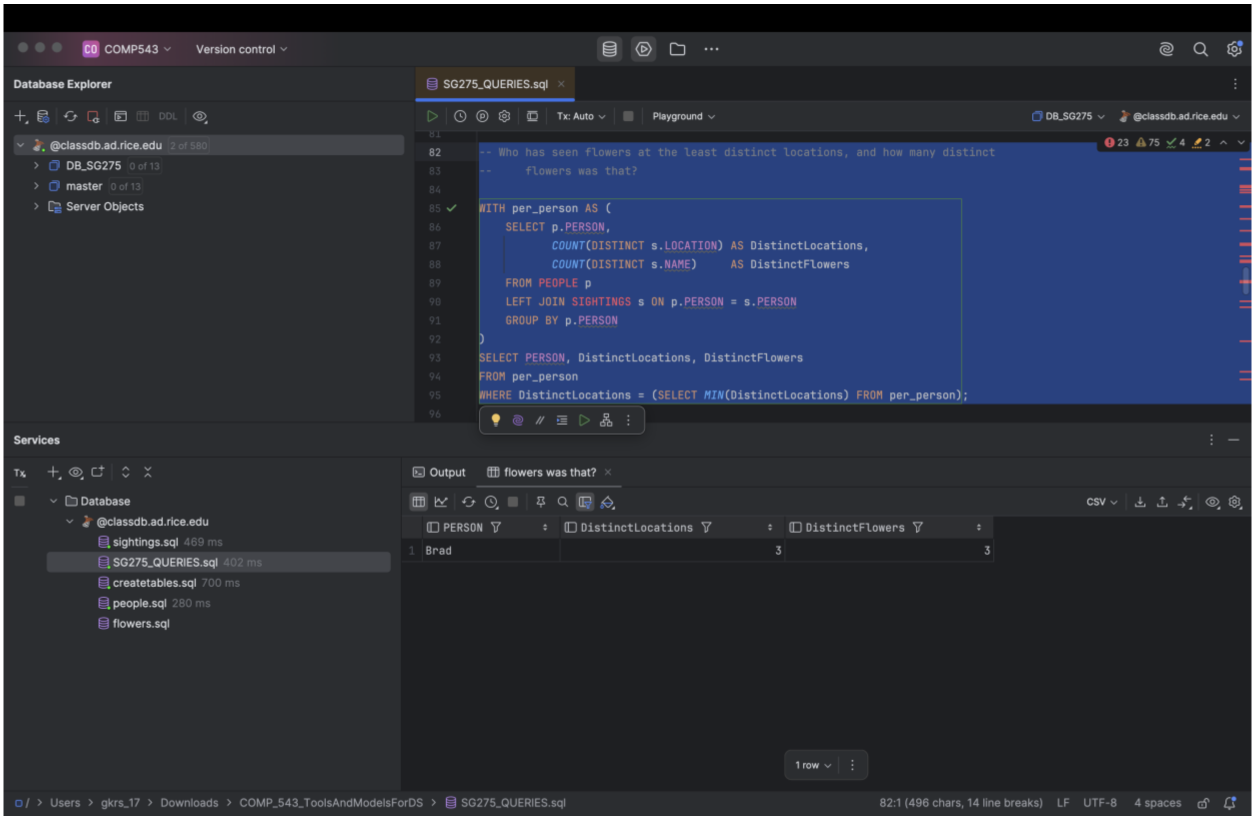
```sql
-- Who has not seen a flower at a location of class Spring?

SELECT p.PERSON
FROM PEOPLE p
WHERE NOT EXISTS (
    SELECT 1
    FROM SIGHTINGS s
    JOIN FEATURES fe ON s.LOCATION = fe.LOCATION
    WHERE s.PERSON = p.PERSON
      AND fe.CLASS = 'Spring'
);
```

```
  PERSON
1 Donna
2 John
3 Sandra
4 Robert
```

/ > Users > gkrs_17 > Downloads > COMP_543_ToolsAndModelsForDS > SG275_QUERIES.sql   69:1 (255 chars, 10 line breaks) LF UTF-8 4 spaces

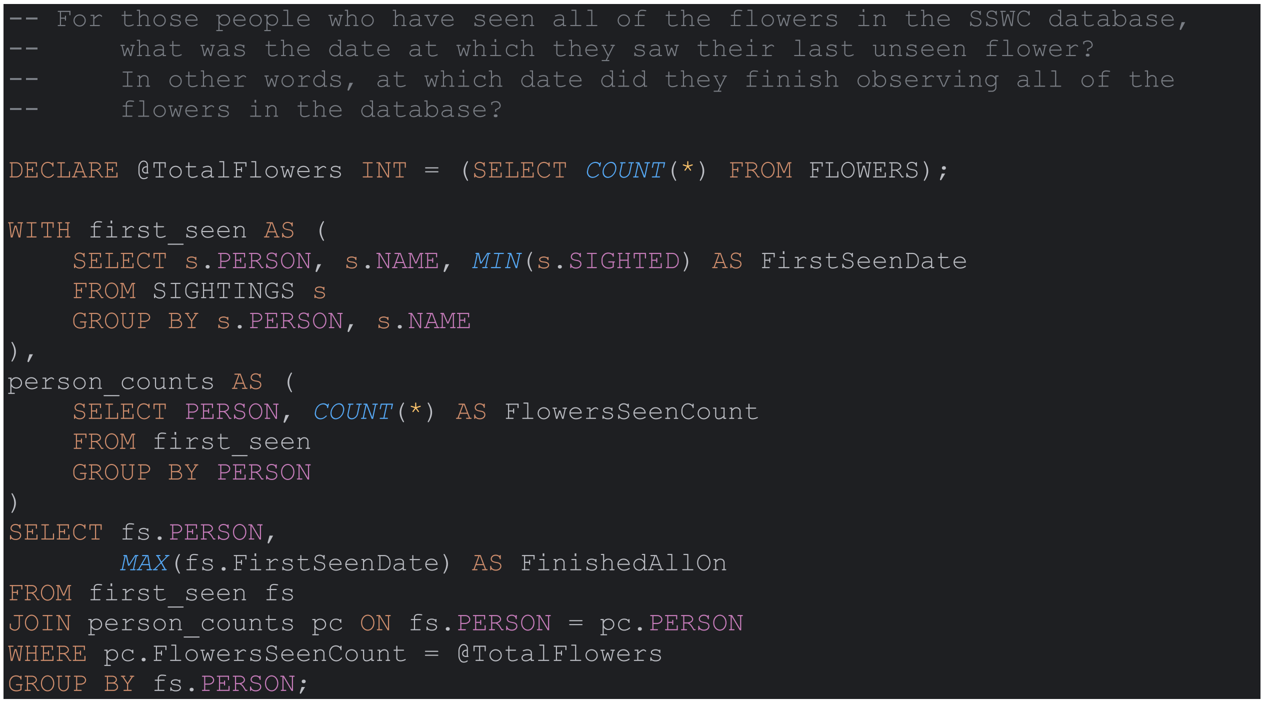
```
-- Who has seen flowers at the least distinct locations, and how many
distinct
--      flowers was that?

WITH per_person AS (
    SELECT p.PERSON,
           COUNT(DISTINCT s.LOCATION) AS DistinctLocations,
           COUNT(DISTINCT s.NAME)     AS DistinctFlowers
    FROM PEOPLE p
    LEFT JOIN SIGHTINGS s ON p.PERSON = s.PERSON
    GROUP BY p.PERSON
)
SELECT PERSON, DistinctLocations, DistinctFlowers
FROM per_person
WHERE DistinctLocations = (SELECT MIN(DistinctLocations) FROM per_person);
```

Output



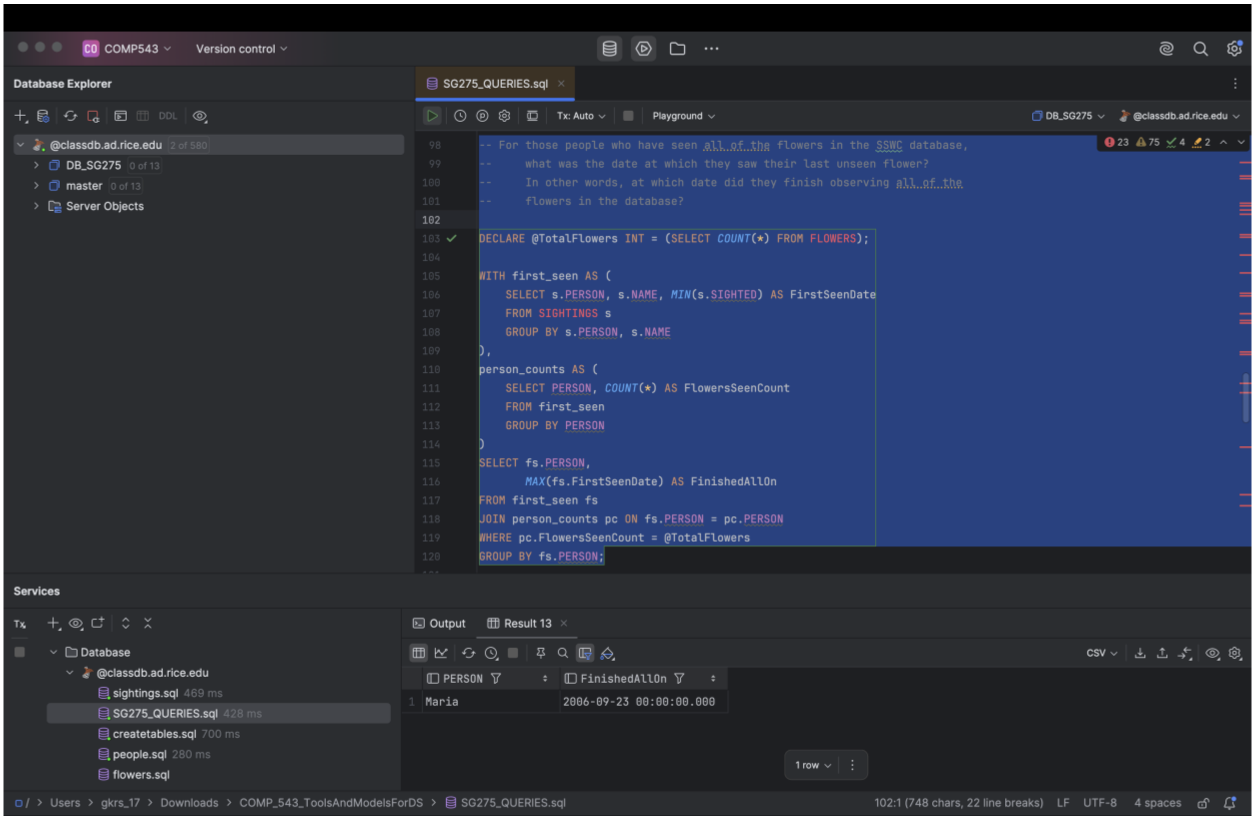
```
-- Who has seen flowers at the least distinct locations, and how many distinct
--     flowers was that?

WITH per_person AS (
    SELECT p.PERSON,
           COUNT(DISTINCT s.LOCATION) AS DistinctLocations,
           COUNT(DISTINCT s.NAME)     AS DistinctFlowers
    FROM PEOPLE p
    LEFT JOIN SIGHTINGS s ON p.PERSON = s.PERSON
    GROUP BY p.PERSON
)
SELECT PERSON, DistinctLocations, DistinctFlowers
FROM per_person
WHERE DistinctLocations = (SELECT MIN(DistinctLocations) FROM per_person);
```

	PERSON	DistinctLocations	DistinctFlowers
1	Brad	3	3



```sql
-- 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?

DECLARE @TotalFlowers INT = (SELECT COUNT(*) FROM FLOWERS);

WITH first_seen AS (
    SELECT s.PERSON, s.NAME, MIN(s.SIGHTED) AS FirstSeenDate
    FROM SIGHTINGS s
    GROUP BY s.PERSON, s.NAME
),
person_counts AS (
    SELECT PERSON, COUNT(*) AS FlowersSeenCount
    FROM first_seen
    GROUP BY PERSON
)
SELECT fs.PERSON,
       MAX(fs.FirstSeenDate) AS FinishedAllOn
FROM first_seen fs
JOIN person_counts pc ON fs.PERSON = pc.PERSON
WHERE pc.FlowersSeenCount = @TotalFlowers
GROUP BY fs.PERSON;
```

Output

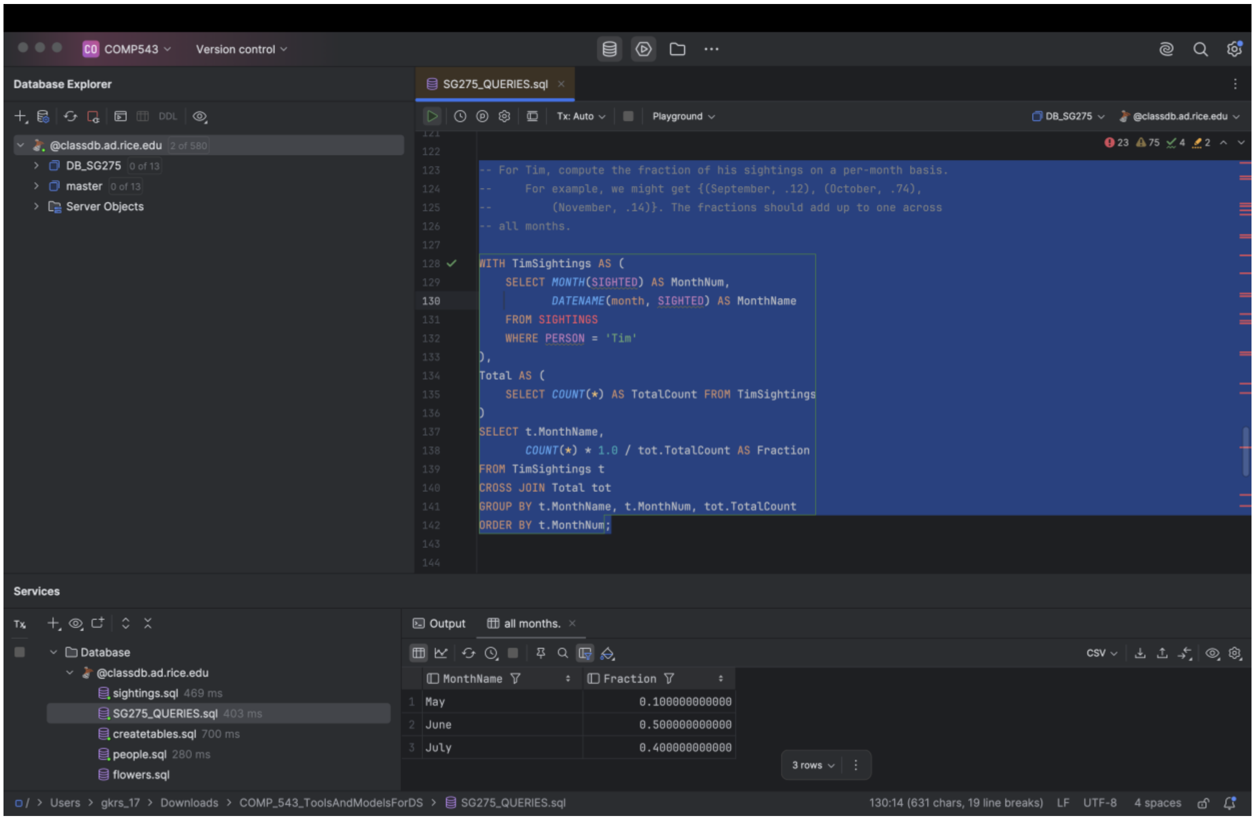


```sql
-- For those people who have seen all of the flowers in the SSWC database,
--     what was the date at which they saw their last unseen flower?
--     In other words, at which date did they finish observing all of the
--     flowers in the database?

DECLARE @TotalFlowers INT = (SELECT COUNT(*) FROM FLOWERS);

WITH first_seen AS (
    SELECT s.PERSON, s.NAME, MIN(s.SIGHTED) AS FirstSeenDate
    FROM SIGHTINGS s
    GROUP BY s.PERSON, s.NAME
),
person_counts AS (
    SELECT PERSON, COUNT(*) AS FlowersSeenCount
    FROM first_seen
    GROUP BY PERSON
)
SELECT fs.PERSON,
       MAX(fs.FirstSeenDate) AS FinishedAllOn
FROM first_seen fs
JOIN person_counts pc ON fs.PERSON = pc.PERSON
WHERE pc.FlowersSeenCount = @TotalFlowers
GROUP BY fs.PERSON;
```

	PERSON	FinishedAllOn
1	Maria	2006-09-23 00:00:00.000

```sql
-- 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.

WITH TimSightings AS (
    SELECT MONTH(SIGHTED) AS MonthNum,
           DATENAME(month, SIGHTED) AS MonthName
    FROM SIGHTINGS
    WHERE PERSON = 'Tim'
),
Total AS (
    SELECT COUNT(*) AS TotalCount FROM TimSightings
)
SELECT t.MonthName,
       COUNT(*) * 1.0 / tot.TotalCount AS Fraction
FROM TimSightings t
CROSS JOIN Total tot
GROUP BY t.MonthName, t.MonthNum, tot.TotalCount
ORDER BY t.MonthNum;
```

Output



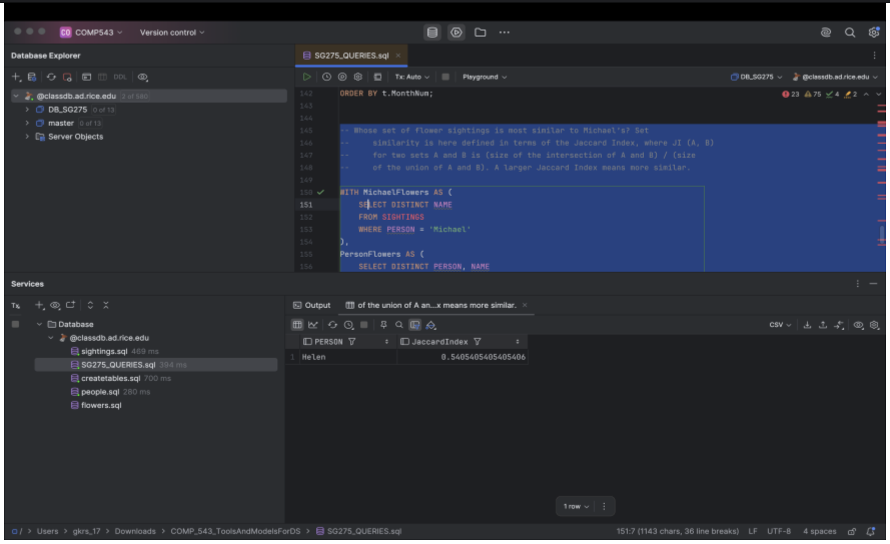
```sql
-- For Tim, compute the fraction of his sightings on a per-month basis.
--      For example, we might get {(September, .12), (October, .74),
--          (November, .14)}. The fractions should add up to one across
-- all months.

WITH TimSightings AS (
    SELECT MONTH(SIGHTED) AS MonthNum,
           DATENAME(month, SIGHTED) AS MonthName
    FROM SIGHTINGS
    WHERE PERSON = 'Tim'
),
Total AS (
    SELECT COUNT(*) AS TotalCount FROM TimSightings
)
SELECT t.MonthName,
       COUNT(*) * 1.0 / tot.TotalCount AS Fraction
FROM TimSightings t
CROSS JOIN Total tot
GROUP BY t.MonthName, t.MonthNum, tot.TotalCount
ORDER BY t.MonthNum;
```

	MonthName	Fraction
1	May	0.100000000000
2	June	0.500000000000
3	July	0.400000000000

3 rows

```
-- 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.

WITH MichaelFlowers AS (
    SELECT DISTINCT NAME
    FROM SIGHTINGS
    WHERE PERSON = 'Michael'
),
PersonFlowers AS (
    SELECT DISTINCT PERSON, NAME
    FROM SIGHTINGS
),
PersonCounts AS (
    SELECT PERSON, COUNT(*) AS PersonCount
    FROM PersonFlowers
    GROUP BY PERSON
),
MichaelCount AS (
    SELECT COUNT(*) AS MCount FROM MichaelFlowers
),
Intersections AS (
    SELECT pf.PERSON, COUNT(*) AS InterCount
    FROM PersonFlowers pf
    JOIN MichaelFlowers mf ON pf.NAME = mf.NAME
    GROUP BY pf.PERSON
)
SELECT TOP 1 WITH TIES
       pc.PERSON,
       CAST(ISNULL(ic.InterCount, 0) AS FLOAT) /
       (pc.PersonCount + mc.MCount - ISNULL(ic.InterCount, 0)) AS
JaccardIndex
FROM PersonCounts pc
LEFT JOIN Intersections ic ON pc.PERSON = ic.PERSON
CROSS JOIN MichaelCount mc
WHERE pc.PERSON <> 'Michael'
ORDER BY JaccardIndex DESC;
```

Output



```
ORDER BY t.MonthNum;


-- 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.

WITH MichaelFlowers AS (
    SELECT DISTINCT NAME
    FROM SIGHTINGS
    WHERE PERSON = 'Michael'
),
PersonFlowers AS (
    SELECT DISTINCT PERSON, NAME
```

	PERSON	JaccardIndex
1	Helen	0.5405405405405406