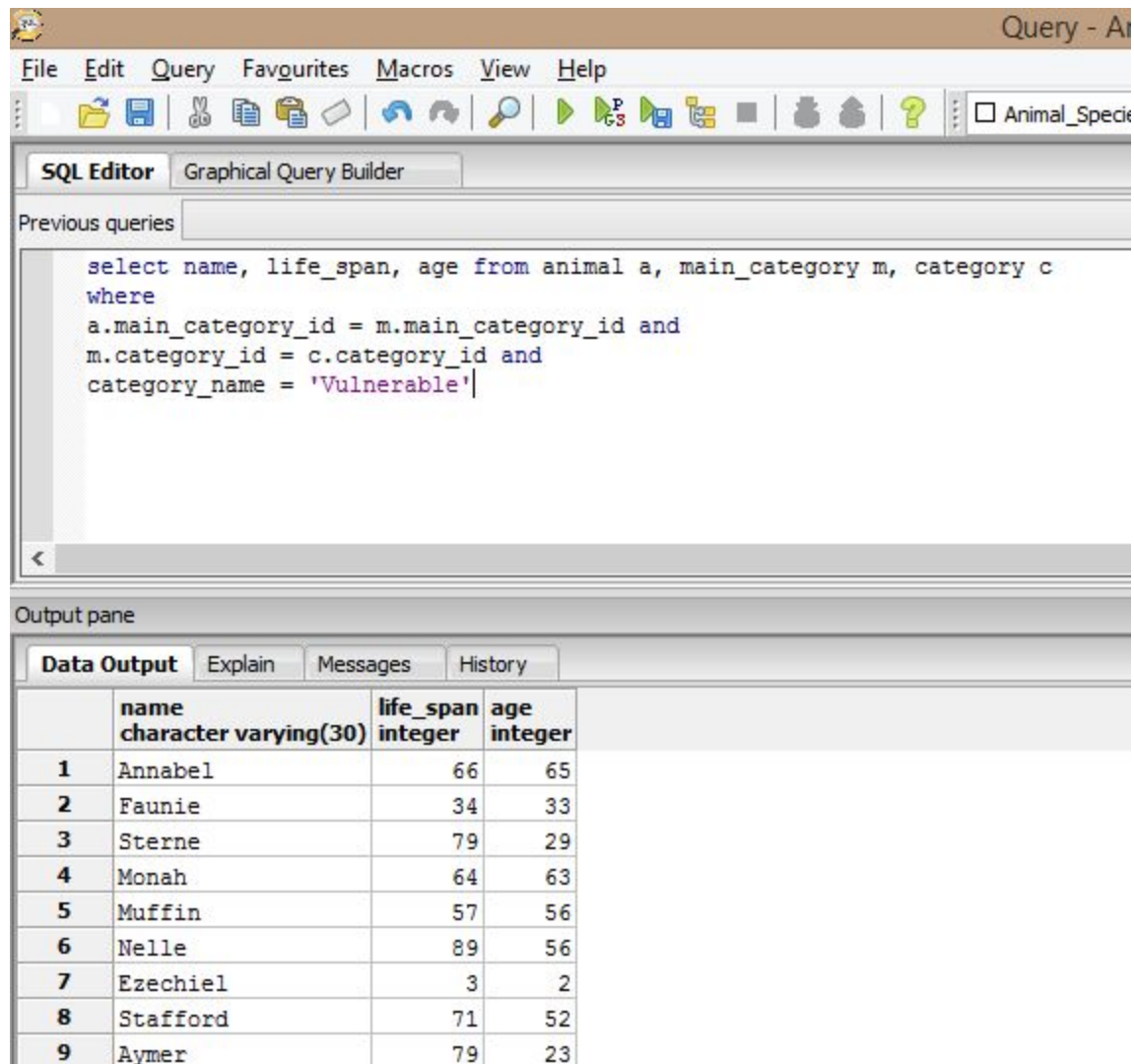

Animal Repository System

Queries

1. List all the names of animals of category 'Vulnerable'

```
SELECT name,  
       life_span,  
       age  
FROM animal a,  
     main_category m,  
     category c  
WHERE a.main_category_id = m.main_category_id  
      AND m.category_id = c.category_id  
      AND category_name = 'Vulnerable'
```



The screenshot shows a database query editor window titled "Query - Animal_Species". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar with various icons. Below the toolbar are two tabs: "SQL Editor" (selected) and "Graphical Query Builder". The "Previous queries" list is empty. The SQL Editor contains the following query:

```
select name, life_span, age from animal a, main_category m, category c  
where  
a.main_category_id = m.main_category_id and  
m.category_id = c.category_id and  
category_name = 'Vulnerable'
```

Below the SQL Editor is the "Output pane" with four tabs: "Data Output" (selected), "Explain", "Messages", and "History". The "Data Output" tab displays the results of the query in a table with the following columns: "name" (character varying(30)), "life_span" (integer), and "age" (integer). The table contains 9 rows of data:

	name character varying(30)	life_span integer	age integer
1	Annabel	66	65
2	Faunie	34	33
3	Sterne	79	29
4	Monah	64	63
5	Muffin	57	56
6	Nelle	89	56
7	Ezechiel	3	2
8	Stafford	71	52
9	Aymer	79	23

2. Find the number of animals in state Gujarat.

```
SELECT
    name,
    life_span,
    age
FROM animal a,
    animal_details d,
    city c,
    state s
WHERE a.animal_id = d.animal_id
AND d.city_id = c.city_id
AND d.state_id = s.state_id
AND state_name = 'Gujarat'
```

Query - Animal

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```

select name, life_span, age from animal a, animal_details d, city c, state s
where
a.animal_id = d.animal_id and
d.city_id = c.city_id and
c.state_id = s.state_id and
state_name = 'Gujarat'

```

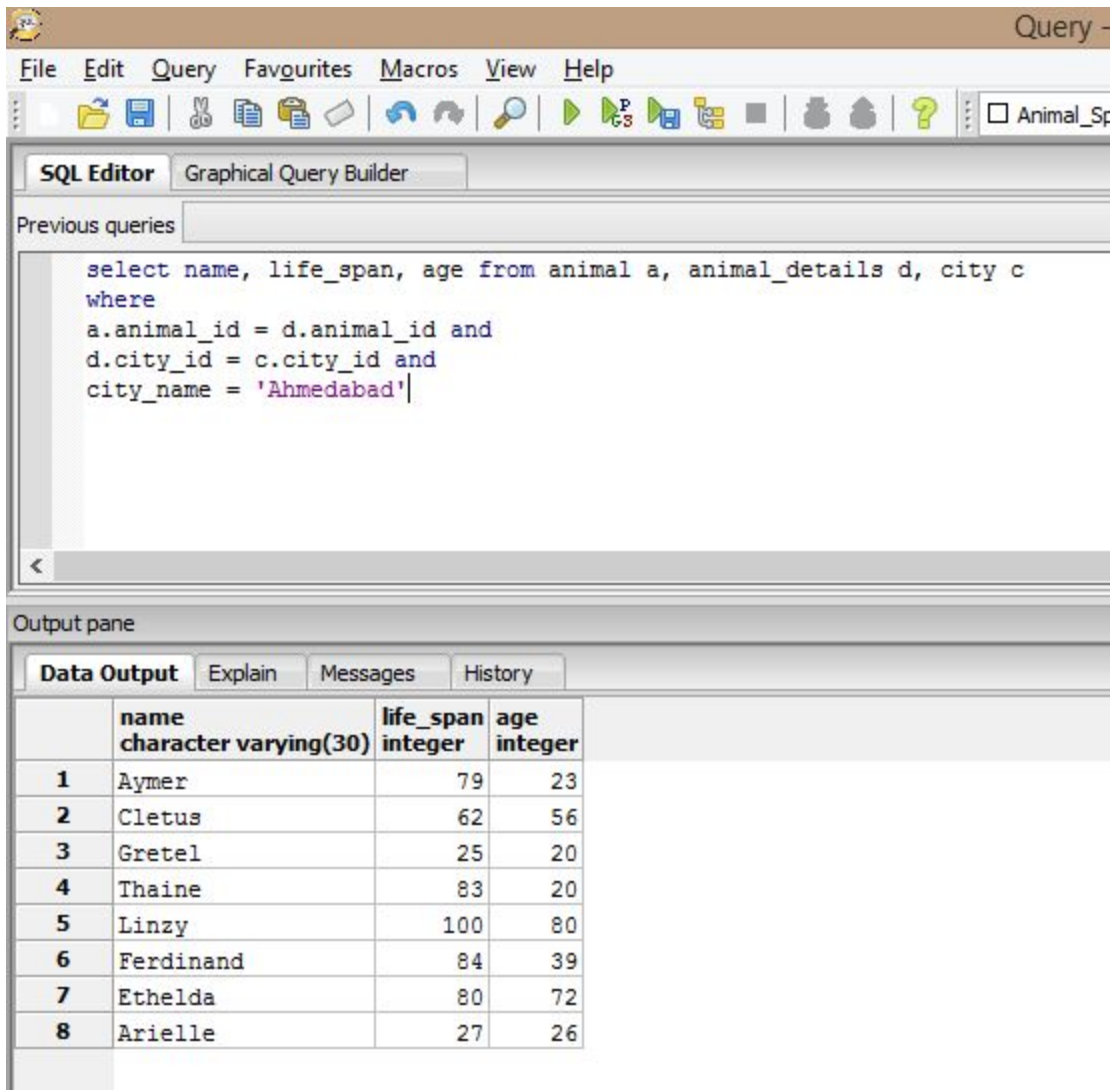
Output pane

Data Output Explain Messages History

	name character varying(30)	life_span integer	age integer
1	Marne	97	11
2	Gert	17	13
3	Talbot	71	47
4	Aland	13	12
5	Aymer	79	23
6	Cletus	62	56
7	Monti	40	14
8	Grannie	19	18
9	Arielle	27	26

3. Find the number of animals in city Ahmedabad

```
SELECT name,  
       life_span,  
       age  
FROM animal a,  
     animal_details d,  
     city c  
WHERE a.animal_id = d.animal_id  
      AND d.city_id = c.city_id  
      AND city_name = 'Ahmedabad'
```



The screenshot shows a database query editor window. The top menu bar includes File, Edit, Query, Favurites, Macros, View, and Help. Below the menu is a toolbar with various icons. The main window is divided into two panes: the SQL Editor and the Output pane. The SQL Editor contains the following query:

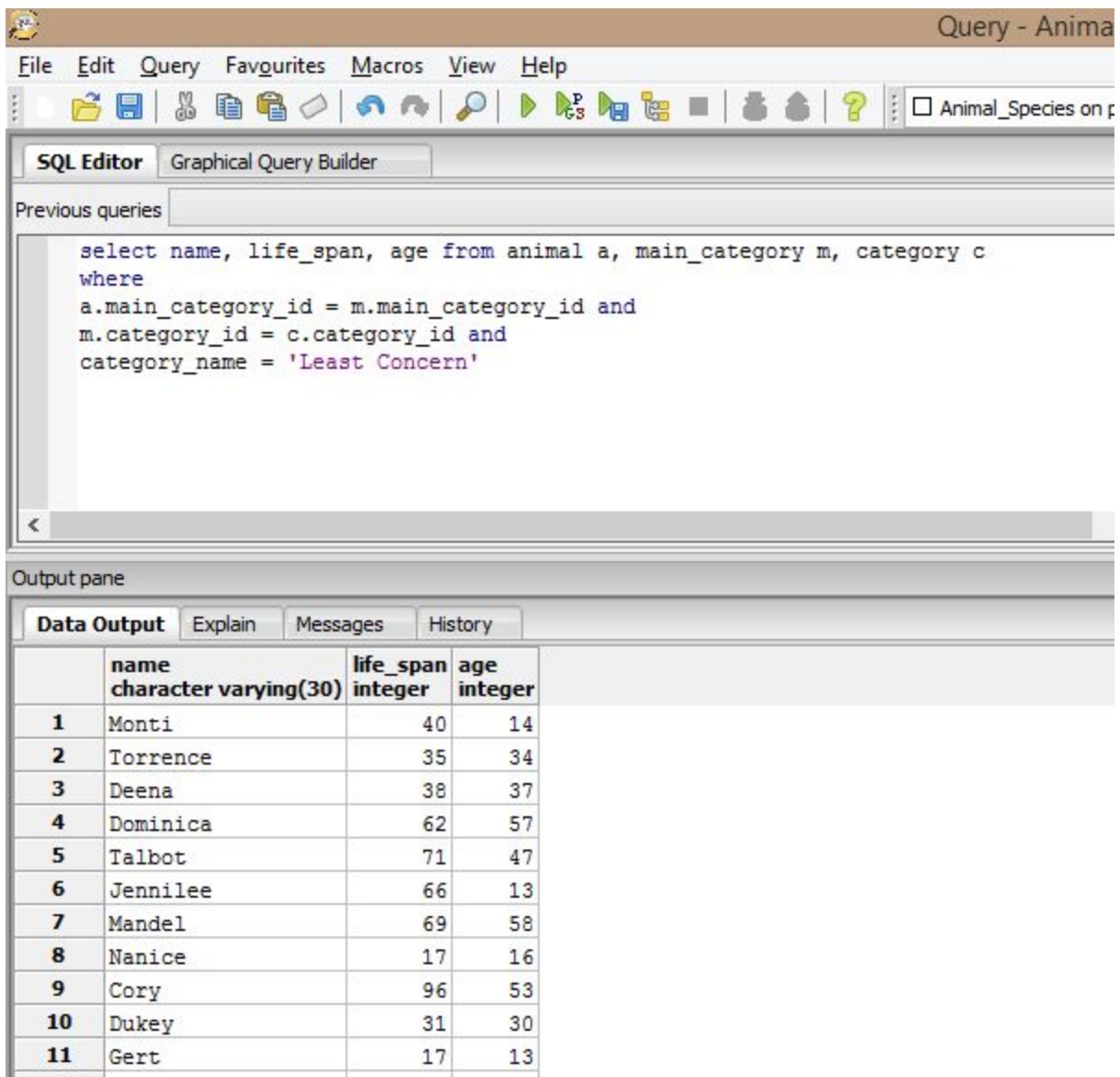
```
select name, life_span, age from animal a, animal_details d, city c  
where  
a.animal_id = d.animal_id and  
d.city_id = c.city_id and  
city_name = 'Ahmedabad'
```

The Output pane is currently displaying the 'Data Output' tab, which shows the results of the query in a table format. The table has four columns: name, life_span, and age. The data is as follows:

	name character varying(30)	life_span integer	age integer
1	Aymer	79	23
2	Cletus	62	56
3	Gretel	25	20
4	Thaine	83	20
5	Linzy	100	80
6	Ferdinand	84	39
7	Ethelda	80	72
8	Arielle	27	26

4. Find the number of animals of category 'Least Concern'

```
SELECT name,  
       life_span,  
       age  
FROM animal a,  
     main_category m,  
     category c  
WHERE a.main_category_id = m.main_category_id  
      AND m.category_id = c.category_id  
      AND category_name = 'Least Concern'
```



The screenshot shows a database query tool interface. The top menu bar includes File, Edit, Query, Favurites, Macros, View, and Help. Below the menu is a toolbar with various icons. The main window is divided into two panes. The top pane is the SQL Editor, which contains the following SQL query:

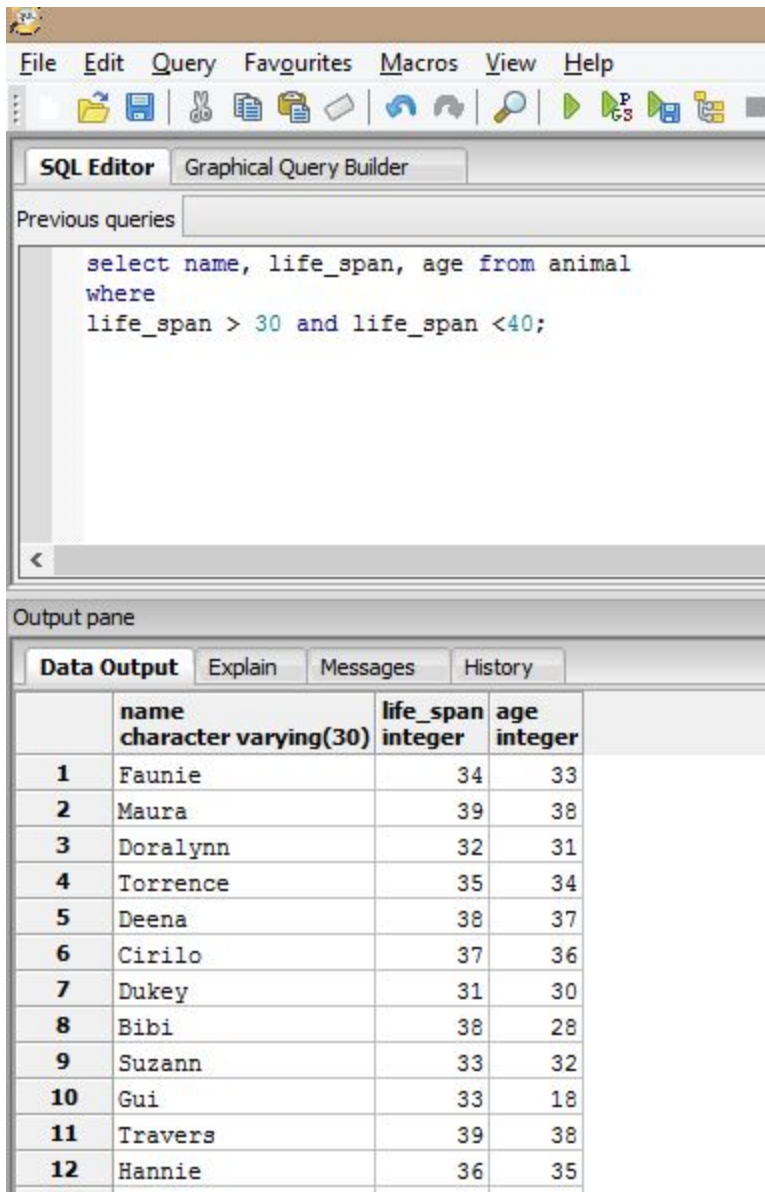
```
select name, life_span, age from animal a, main_category m, category c  
where  
a.main_category_id = m.main_category_id and  
m.category_id = c.category_id and  
category_name = 'Least Concern'
```

The bottom pane is the Output pane, which has tabs for Data Output, Explain, Messages, and History. The Data Output tab is selected, showing a table with 11 rows of results. The table has four columns: name, life_span, and age. The data is as follows:

	name character varying(30)	life_span integer	age integer
1	Monti	40	14
2	Torrence	35	34
3	Deena	38	37
4	Dominica	62	57
5	Talbot	71	47
6	Jennilee	66	13
7	Mandel	69	58
8	Nanice	17	16
9	Cory	96	53
10	Dukey	31	30
11	Gert	17	13

5. List all the animals whose 'life-span' is between 30 and 40.

```
SELECT name,  
       life_span,  
       age  
FROM animal  
WHERE life_span >30  
      AND life_span <40
```



The screenshot shows a database application window with a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The 'SQL Editor' tab is active, displaying the following query:

```
select name, life_span, age from animal  
where  
life_span > 30 and life_span <40;
```

Below the editor is the 'Output pane' with tabs for 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is selected, showing a table with 12 rows of results. The table has four columns: an index, 'name', 'life_span', and 'age'.

	name character varying(30)	life_span integer	age integer
1	Faunie	34	33
2	Maura	39	38
3	Doralynn	32	31
4	Torrence	35	34
5	Deena	38	37
6	Cirilo	37	36
7	Dukey	31	30
8	Bibi	38	28
9	Suzann	33	32
10	Gui	33	18
11	Travers	39	38
12	Hannie	36	35

6. List all the animals whose name starts with 'A' and weight is between 100 and 500.

```
SELECT name
FROM animal
WHERE weight > 100
      AND weight < 500
      AND name LIKE 'A%'
```

The screenshot shows a database application window with a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is the 'SQL Editor' tab, which contains the following SQL query:

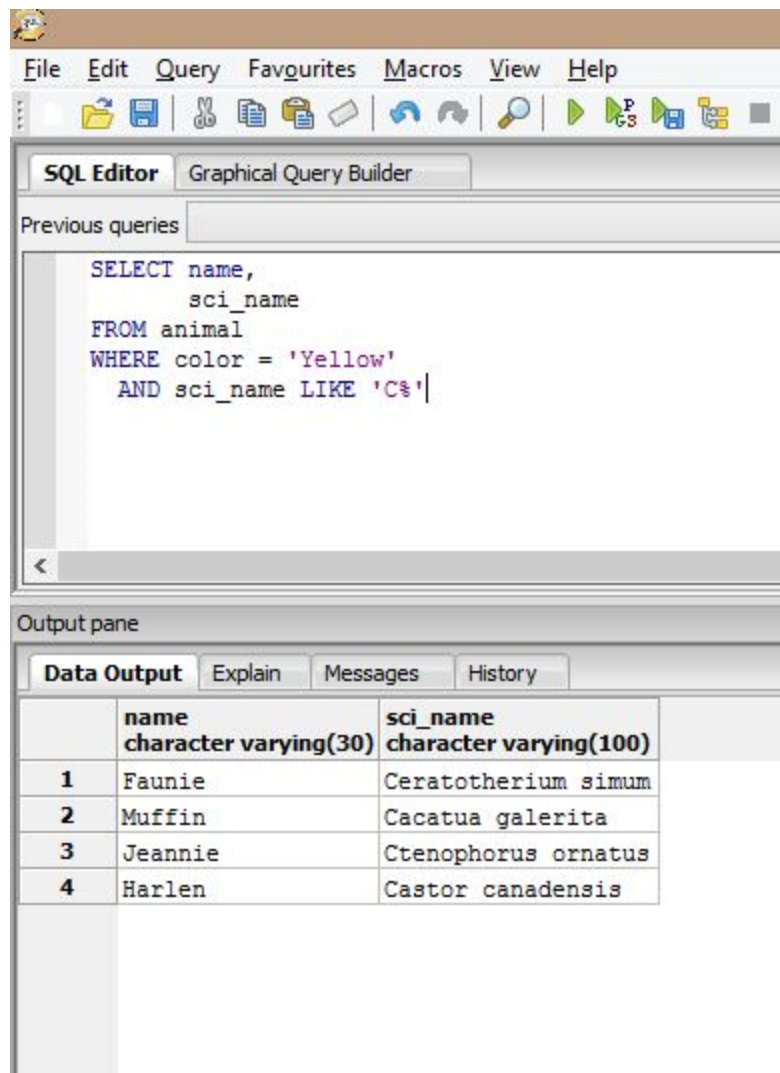
```
SELECT name
FROM animal
WHERE weight > 100
      AND weight < 500
      AND name LIKE 'A%'
```

Below the SQL Editor is the 'Output pane' with tabs for 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is active, displaying a table with the results of the query:

	name character varying(30)
1	Annabel
2	Aymer
3	Averell
4	Ange
5	Aland

7. List all the animals whose scientific name starts with 'C' and has color yellow.

```
SELECT name,  
       sci_name  
FROM animal  
WHERE color = 'Yellow'  
       AND sci_name LIKE 'C%'
```



The screenshot shows a database application window with a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is the SQL Editor, which contains the following query:

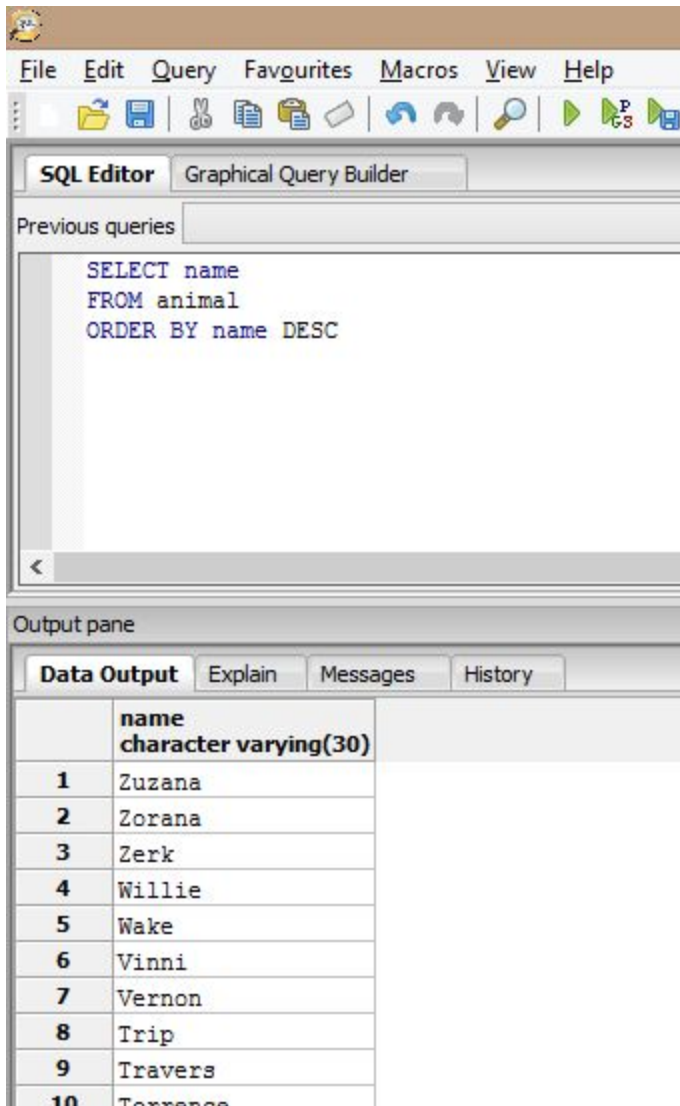
```
SELECT name,  
       sci_name  
FROM animal  
WHERE color = 'Yellow'  
       AND sci_name LIKE 'C%'
```

Below the SQL Editor is the Output pane, which has tabs for Data Output, Explain, Messages, and History. The Data Output tab is selected, showing a table with the results of the query:

	name character varying(30)	sci_name character varying(100)
1	Faunie	Ceratotherium simum
2	Muffin	Cacatua galerita
3	Jeannie	Ctenophorus ornatus
4	Harlen	Castor canadensis

8. List all the animals in descending order (names).

```
SELECT name  
FROM animal  
ORDER BY name DESC
```

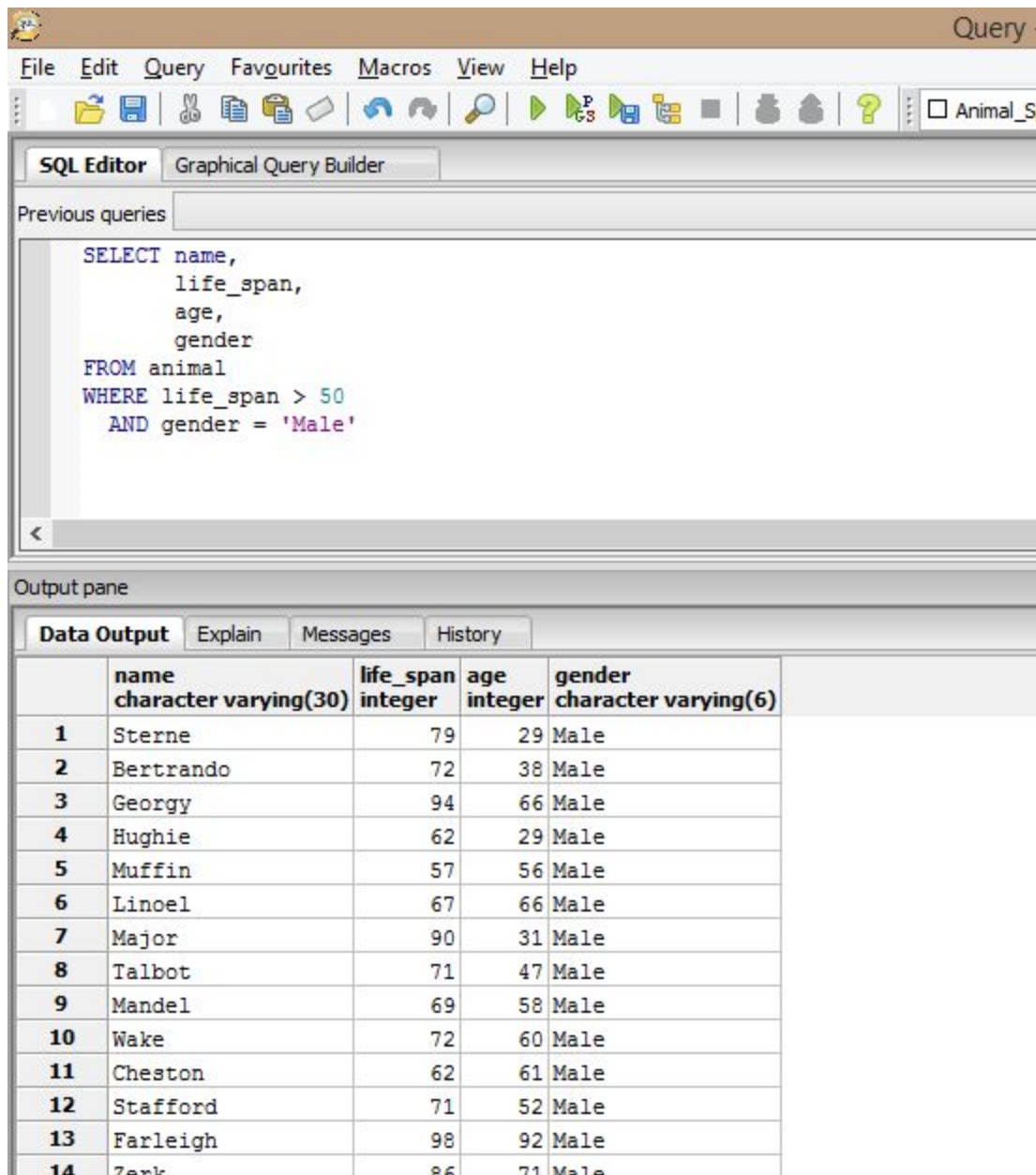


The screenshot shows a database application window with a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The main area is split into two panes: the SQL Editor and the Output pane. The SQL Editor contains the query: `SELECT name
FROM animal
ORDER BY name DESC`. The Output pane is active and shows the results of the query in a table format.

	name character varying(30)
1	Zuzana
2	Zorana
3	Zerk
4	Willie
5	Wake
6	Vinni
7	Vernon
8	Trip
9	Travers
10	Terrance

9. Count the number of male animals whose life-span is greater than 50.

```
SELECT name,  
       life_span,  
       age,  
       gender  
FROM animal  
WHERE life_span > 50  
      AND gender = 'Male'
```



The screenshot shows a database query editor window titled "Query". The "SQL Editor" tab is active, displaying the following SQL query:

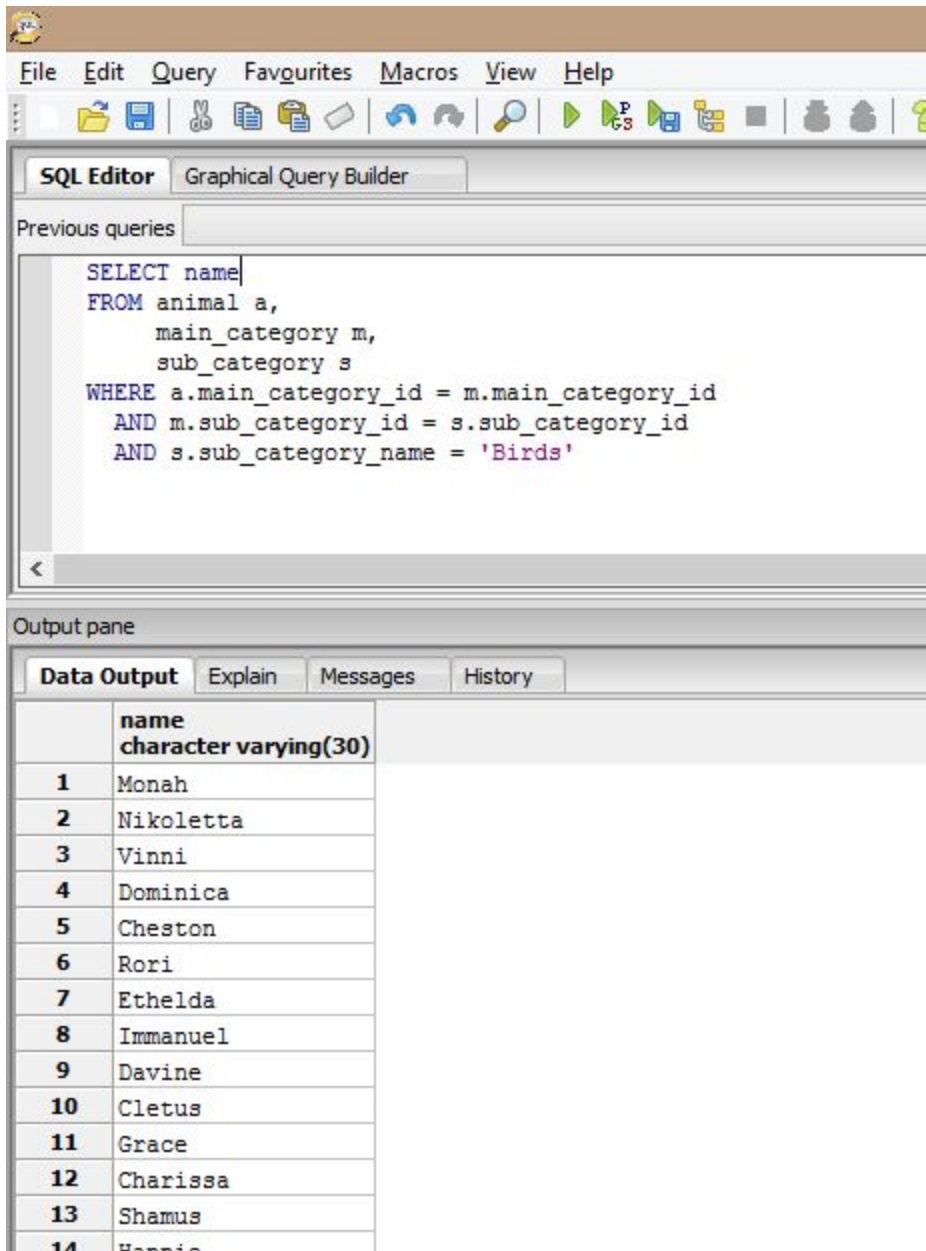
```
SELECT name,  
       life_span,  
       age,  
       gender  
FROM animal  
WHERE life_span > 50  
      AND gender = 'Male'
```

The "Output pane" at the bottom shows the "Data Output" tab with the following results:

	name character varying(30)	life_span integer	age integer	gender character varying(6)
1	Sterne	79	29	Male
2	Bertrando	72	38	Male
3	Georgy	94	66	Male
4	Hughie	62	29	Male
5	Muffin	57	56	Male
6	Linoel	67	66	Male
7	Major	90	31	Male
8	Talbot	71	47	Male
9	Mandel	69	58	Male
10	Wake	72	60	Male
11	Cheston	62	61	Male
12	Stafford	71	52	Male
13	Farleigh	98	92	Male
14	Dark	86	71	Male

10. List all the Birds.

```
SELECT name,  
FROM animal a,  
       main_category m,  
       sub_category s  
WHERE a.main_category_id = m.main_category_id  
      AND m.sub_category_id = s.sub_category_id  
      AND s.sub_category_name = 'Birds'
```



The screenshot shows a database application window with a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar. The main window is divided into two panes. The top pane, titled 'SQL Editor', contains the following SQL query:

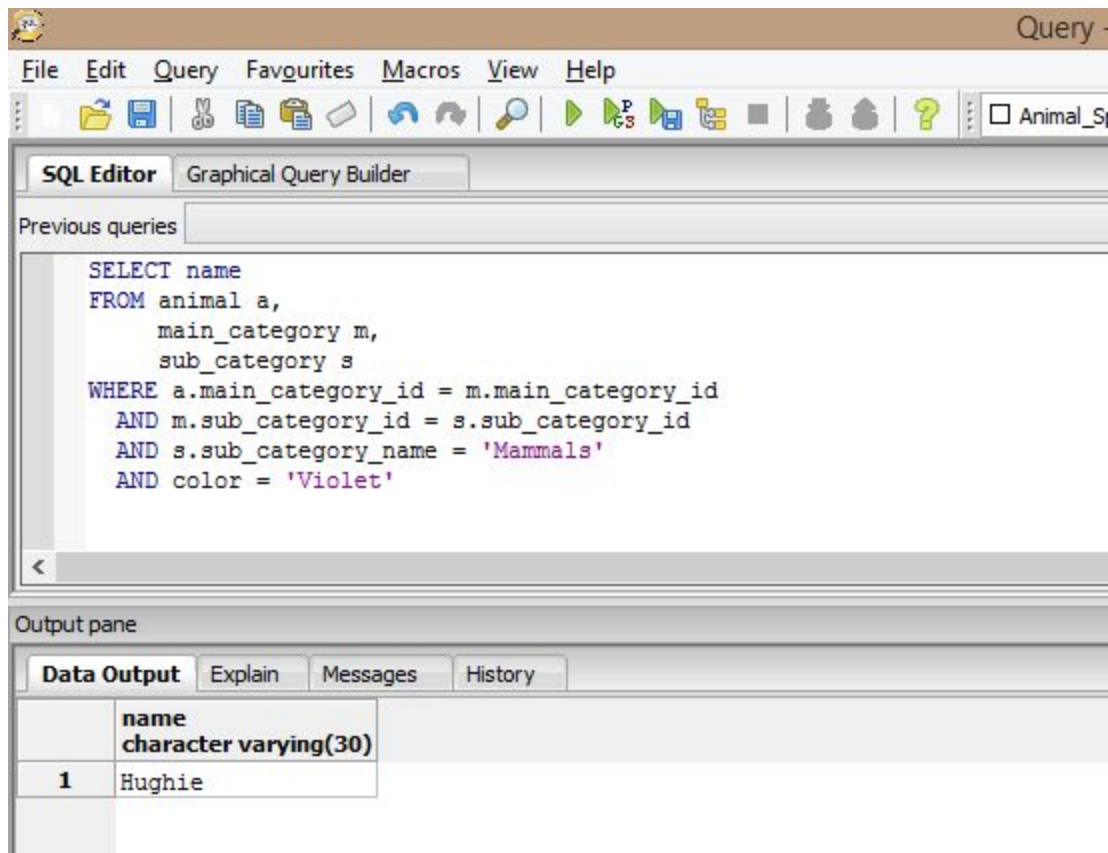
```
SELECT name  
FROM animal a,  
       main_category m,  
       sub_category s  
WHERE a.main_category_id = m.main_category_id  
      AND m.sub_category_id = s.sub_category_id  
      AND s.sub_category_name = 'Birds'
```

The bottom pane, titled 'Output pane', has tabs for 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is selected, displaying a table with 14 rows of bird names. The table has two columns: 'name' and 'character varying(30)'.

	name character varying(30)
1	Monah
2	Nikoletta
3	Vinni
4	Dominica
5	Cheston
6	Rori
7	Ethelda
8	Immanuel
9	Davine
10	Cletus
11	Grace
12	Charissa
13	Shamus
14	Hannie

11. List all the mammals whose color is violet.

```
SELECT name
FROM animal a,
     main_category m,
     sub_category s
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = s.sub_category_id
     AND s.sub_category_name = 'Mammals'
     AND color = 'Violet'
```



The screenshot shows a database query editor window. The top menu bar includes File, Edit, Query, Favurites, Macros, View, and Help. Below the menu is a toolbar with various icons. The main window is titled "SQL Editor" and contains the following SQL query:

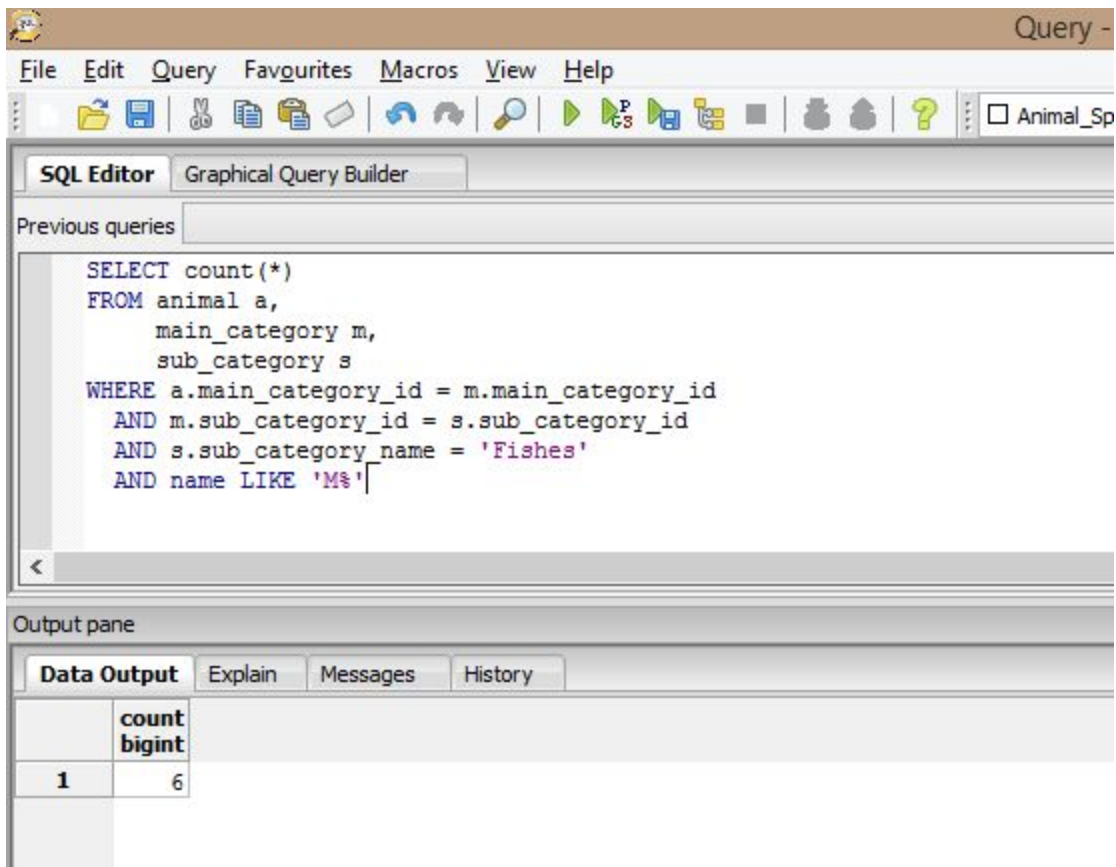
```
SELECT name
FROM animal a,
     main_category m,
     sub_category s
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = s.sub_category_id
     AND s.sub_category_name = 'Mammals'
     AND color = 'Violet'
```

Below the query editor is an "Output pane" with tabs for Data Output, Explain, Messages, and History. The "Data Output" tab is selected, showing the following result:

	name character varying(30)
1	Hughie

12. Count all the Fishes whose name starts with 'M'.

```
SELECT count(*)
FROM animal a,
     main_category m,
     sub_category s
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = s.sub_category_id
     AND s.sub_category_name = 'Fishes'
     AND name LIKE 'M%'
```



The screenshot shows a database query editor window titled "Query -". The window has a menu bar with "File", "Edit", "Query", "Favourites", "Macros", "View", and "Help". Below the menu bar is a toolbar with various icons. The main area is divided into two tabs: "SQL Editor" and "Graphical Query Builder". The "SQL Editor" tab is active, displaying the following SQL query:

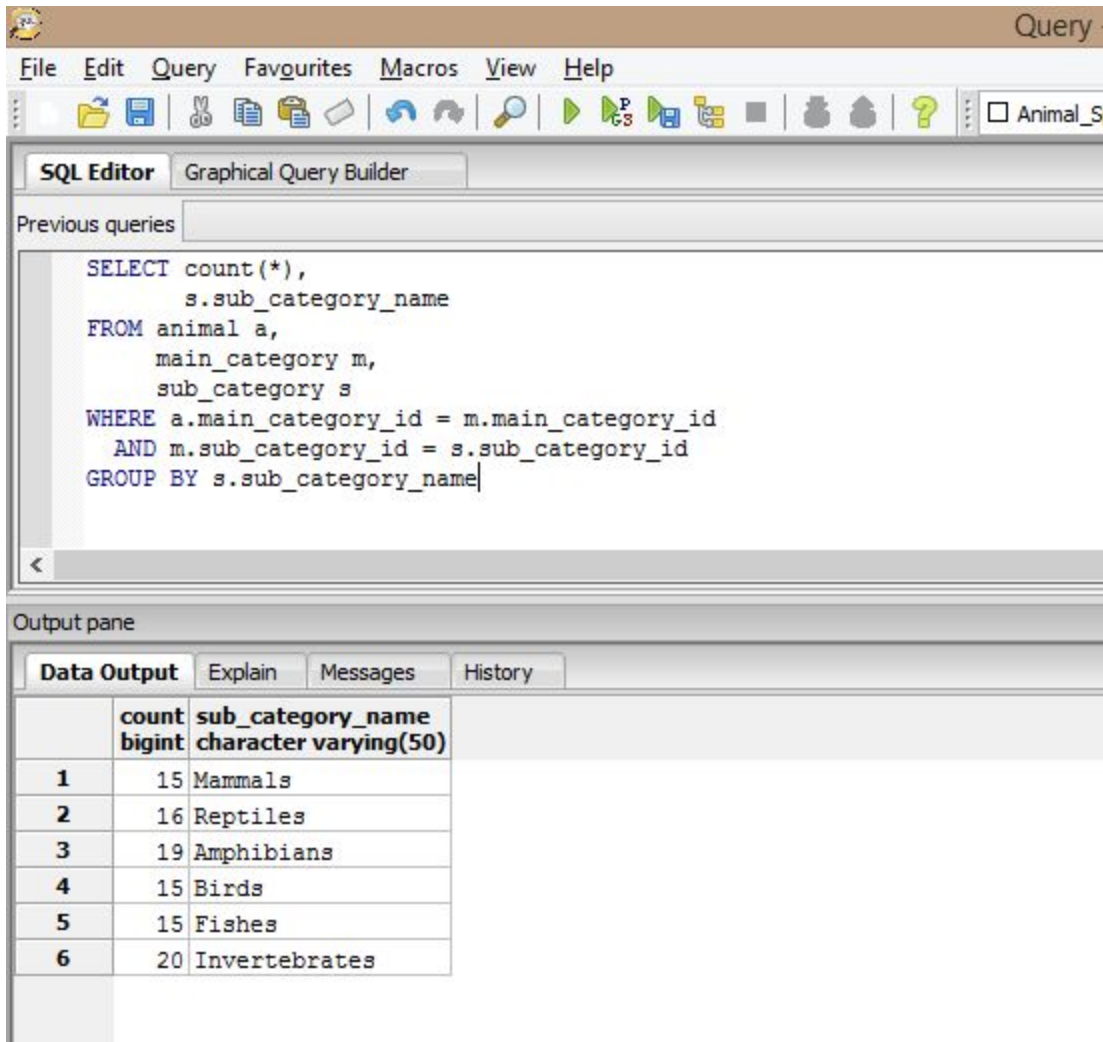
```
SELECT count(*)
FROM animal a,
     main_category m,
     sub_category s
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = s.sub_category_id
     AND s.sub_category_name = 'Fishes'
     AND name LIKE 'M%'
```

Below the SQL Editor is a section labeled "Output pane" with four tabs: "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, showing a table with the following data:

	count bigint
1	6

13. Count all the animals sub-category wise.

```
SELECT count(*),
       s.sub_category_name
FROM animal a,
       main_category m,
       sub_category s
WHERE a.main_category_id = m.main_category_id
      AND m.sub_category_id = s.sub_category_id
GROUP BY s.sub_category_name
```



The screenshot shows a database query editor window titled "Query". The "SQL Editor" tab is active, displaying the following SQL query:

```
SELECT count(*),
       s.sub_category_name
FROM animal a,
       main_category m,
       sub_category s
WHERE a.main_category_id = m.main_category_id
      AND m.sub_category_id = s.sub_category_id
GROUP BY s.sub_category_name
```

Below the query editor is the "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is selected, showing the results of the query in a table format:

	count bigint	sub_category_name character varying(50)
1	15	Mammals
2	16	Reptiles
3	19	Amphibians
4	15	Birds
5	15	Fishes
6	20	Invertebrates

14. Count the total number of animals who are either Birds or Mammals.

```
SELECT sum(t1.total) AS TOTAL from
(SELECT count(*) AS total, s.sub_category_name
FROM animal a, main_category m, sub_category s
WHERE a.main_category_id = m.main_category_id
AND m.sub_category_id = s.sub_category_id
AND (s.sub_category_name = 'Birds'
OR s.sub_category_name = 'Mammals')
GROUP BY s.sub_category_name) AS t1
```

The screenshot shows a database application interface. The top menu bar includes File, Edit, Query, Favourites, Macros, View, and Help. Below the menu is a toolbar with various icons. The main window is divided into two panes. The top pane is the SQL Editor, which contains the following SQL query:

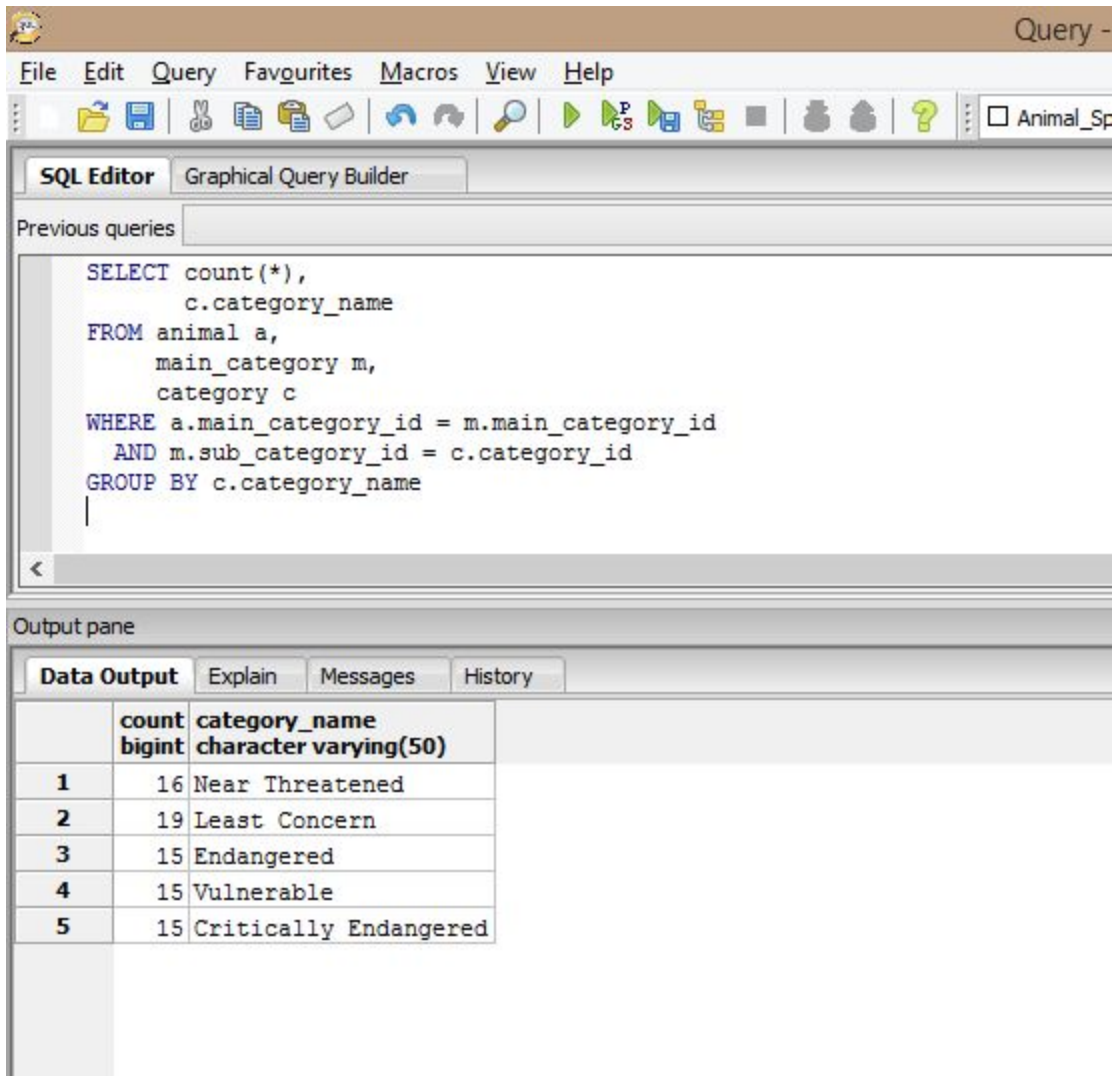
```
SELECT sum(t1.total) AS TOTAL from
(SELECT count(*) AS total, s.sub_category_name
FROM animal a, main_category m, sub_category s
WHERE a.main_category_id = m.main_category_id
AND m.sub_category_id = s.sub_category_id
AND (s.sub_category_name = 'Birds'
OR s.sub_category_name = 'Mammals')
GROUP BY s.sub_category_name) AS t1
```

The bottom pane is the Output pane, which has tabs for Data Output, Explain, Messages, and History. The Data Output tab is selected, showing a table with the following data:

	total numeric
1	30

15. Count all the animals category wise.(c).

```
SELECT count(*),
       c.category_name
FROM animal a,
     main_category m,
     category c
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = c.category_id
GROUP BY c.category_name
```



The screenshot shows a database query editor window titled "Query -". The window has a menu bar with "File", "Edit", "Query", "Favourites", "Macros", "View", and "Help". Below the menu bar is a toolbar with various icons. The main area is divided into two tabs: "SQL Editor" and "Graphical Query Builder". The "SQL Editor" tab is active, showing the following SQL query:

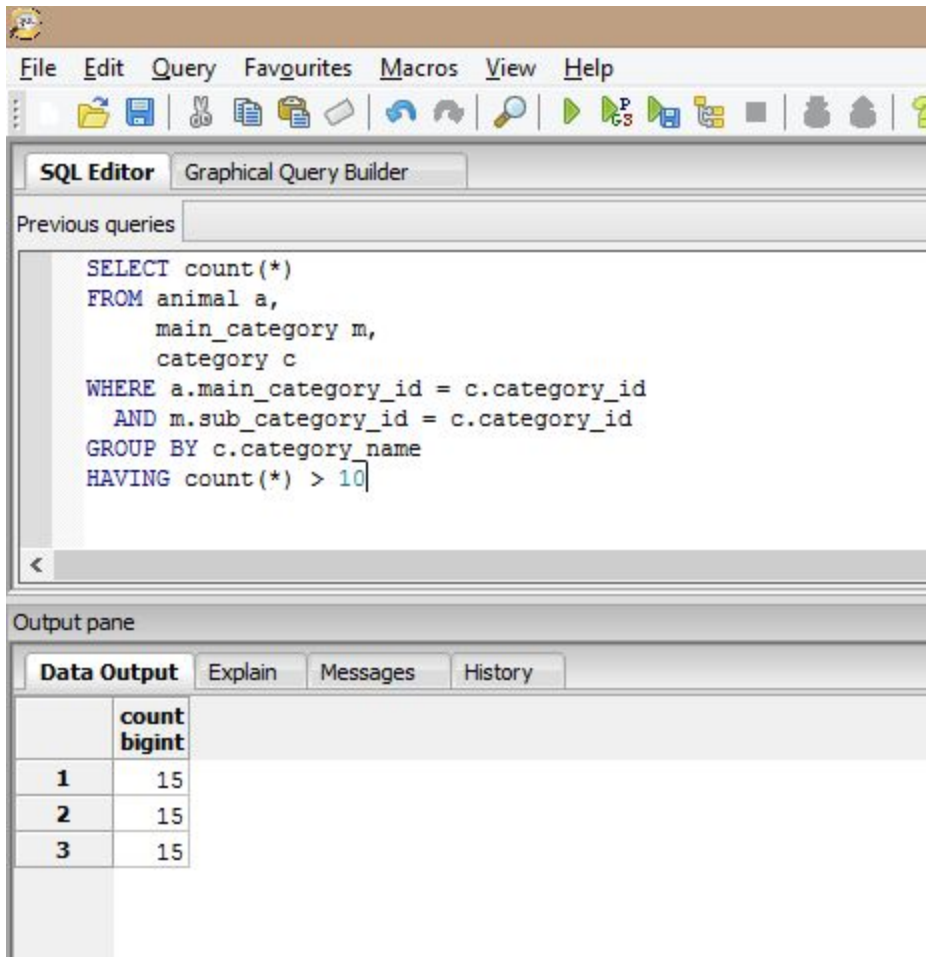
```
SELECT count(*),
       c.category_name
FROM animal a,
     main_category m,
     category c
WHERE a.main_category_id = m.main_category_id
     AND m.sub_category_id = c.category_id
GROUP BY c.category_name
```

Below the SQL editor is an "Output pane" with four tabs: "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is active, displaying the results of the query in a table format:

	count bigint	category_name character varying(50)
1	16	Near Threatened
2	19	Least Concern
3	15	Endangered
4	15	Vulnerable
5	15	Critically Endangered

16. List the animals category wise whose count is greater than 10.

```
SELECT count(*)  
FROM animal a,  
      main_category m,  
      category c  
WHERE a.main_category_id = c.category_id  
      AND m.sub_category_id = c.category_id  
GROUP BY c.category_name  
HAVING count(*) > 10
```

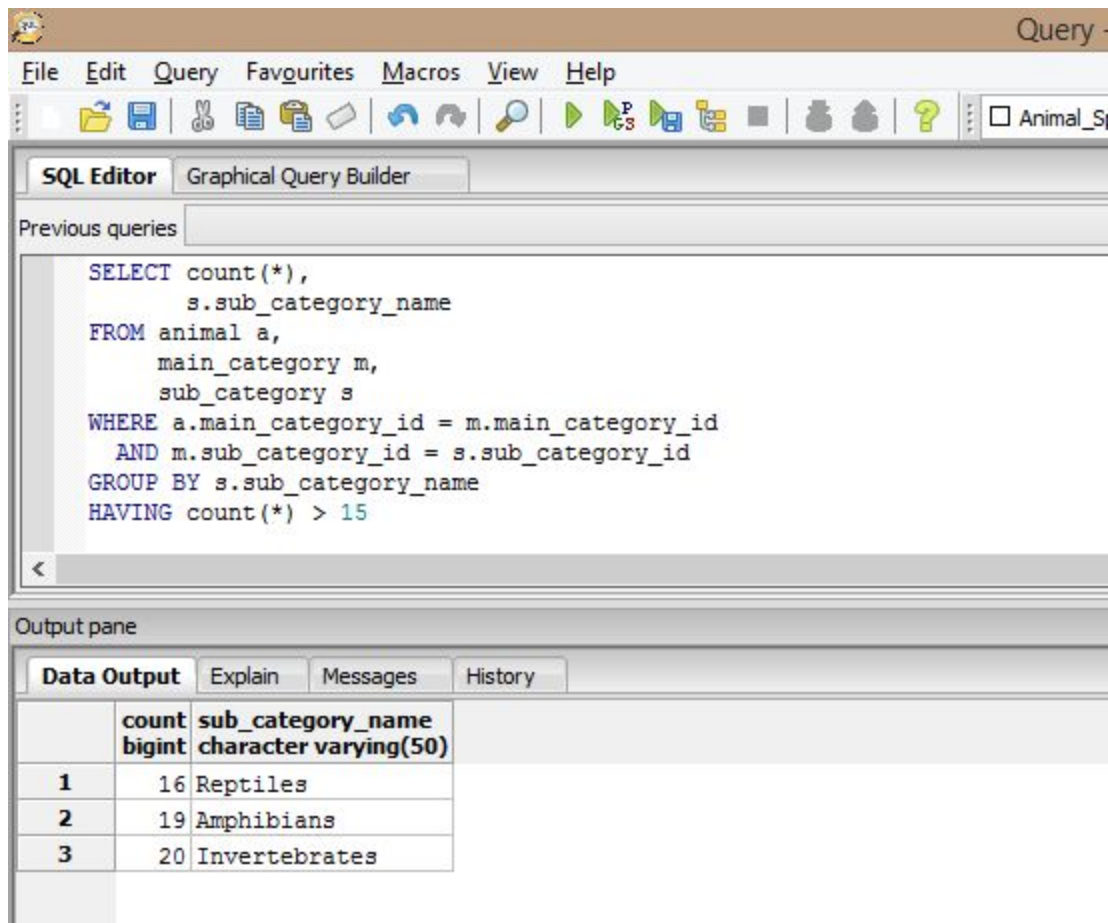


The screenshot shows a database application window with a menu bar (File, Edit, Query, Favurites, Macros, View, Help) and a toolbar. The main window is divided into two panes. The top pane, titled 'SQL Editor', contains the SQL query from the previous block. The bottom pane, titled 'Output pane', has tabs for 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is active, displaying a table with the results of the query.

	count bigint
1	15
2	15
3	15

17. List the animals sub-category wise whose count is greater than 15.

```
SELECT count(*),  
       s.sub_category_name  
FROM animal a,  
     main_category m,  
     sub_category s  
WHERE a.main_category_id = m.main_category_id  
     AND m.sub_category_id = s.sub_category_id  
GROUP BY s.sub_category_name  
HAVING count(*) > 15
```



The screenshot shows a database query editor window with the following components:

- SQL Editor:** Contains the SQL query from the previous block.
- Previous queries:** A list of previously executed queries.
- Output pane:** Displays the results of the query in a table format.

The output table is as follows:

	count bigint	sub_category_name character varying(50)
1	16	Reptiles
2	19	Amphibians
3	20	Invertebrates

18. Name the sub-category having highest number of animals.

```
SELECT s.sub_category_name,  
       count(*) AS total  
FROM animal a,  
     main_category m,  
     sub_category s  
WHERE a.main_category_id = m.main_category_id  
     AND m.sub_category_id = s.sub_category_id  
GROUP BY s.sub_category_name  
HAVING count(*) =  
     (SELECT max(total) from  
     (SELECT s.sub_category_name, count(*) AS total  
     FROM animal a, main_category m, sub_category s  
     WHERE a.main_category_id = m.main_category_id  
     AND m.sub_category_id = s.sub_category_id  
     GROUP BY s.sub_category_name) t1)
```

Query -

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT s.sub_category_name,
       count(*) AS total
FROM animal a,
     main_category m,
     sub_category s
WHERE a.main_category_id = m.main_category_id
      AND m.sub_category_id = s.sub_category_id
GROUP BY s.sub_category_name
HAVING count(*) =
(SELECT max(total) from
 (SELECT s.sub_category_name, count(*) AS total
  FROM animal a, main_category m, sub_category s
  WHERE a.main_category_id = m.main_category_id
        AND m.sub_category_id = s.sub_category_id
  GROUP BY s.sub_category_name) t1)
```

Output pane

Data Output Explain Messages History

	sub_category_name character varying(50)	total bigint
1	Invertebrates	20

19. Name the category having highest number of animals.

```
SELECT c.category_name,  
       count(*) AS total  
FROM animal a,  
     main_category m,  
     category c  
WHERE a.main_category_id = m.main_category_id  
      AND m.category_id = c.category_id  
GROUP BY c.category_name  
HAVING count(*) =  
      (SELECT max(total) from  
      (SELECT c.category_name, count(*) AS total  
      FROM animal a, main_category m, category c  
      WHERE a.main_category_id = m.main_category_id  
            AND m.category_id = c.category_id  
      GROUP BY c.category_name) t1)
```

Query -

File Edit Query Favouirites Macros View Help

Previous queries

```
SELECT c.category_name,
       count(*) AS total
FROM animal a,
     main_category m,
     category c
WHERE a.main_category_id = m.main_category_id
     AND m.category_id = c.category_id
GROUP BY c.category_name
HAVING count(*) =
  (SELECT max(total) from
   (SELECT c.category_name, count(*) AS total
    FROM animal a, main_category m, category c
    WHERE a.main_category_id = m.main_category_id
          AND m.category_id = c.category_id
    GROUP BY c.category_name) t1)
```

Output pane

Data Output Explain Messages History

	category_name character varying(50)	total bigint
1	Endangered	29

20. Name all the reptiles which are Least Concerned.

select name, c.category_name, s.sub_category_name as total from animal a,
main_category m, sub_category s, category c

where

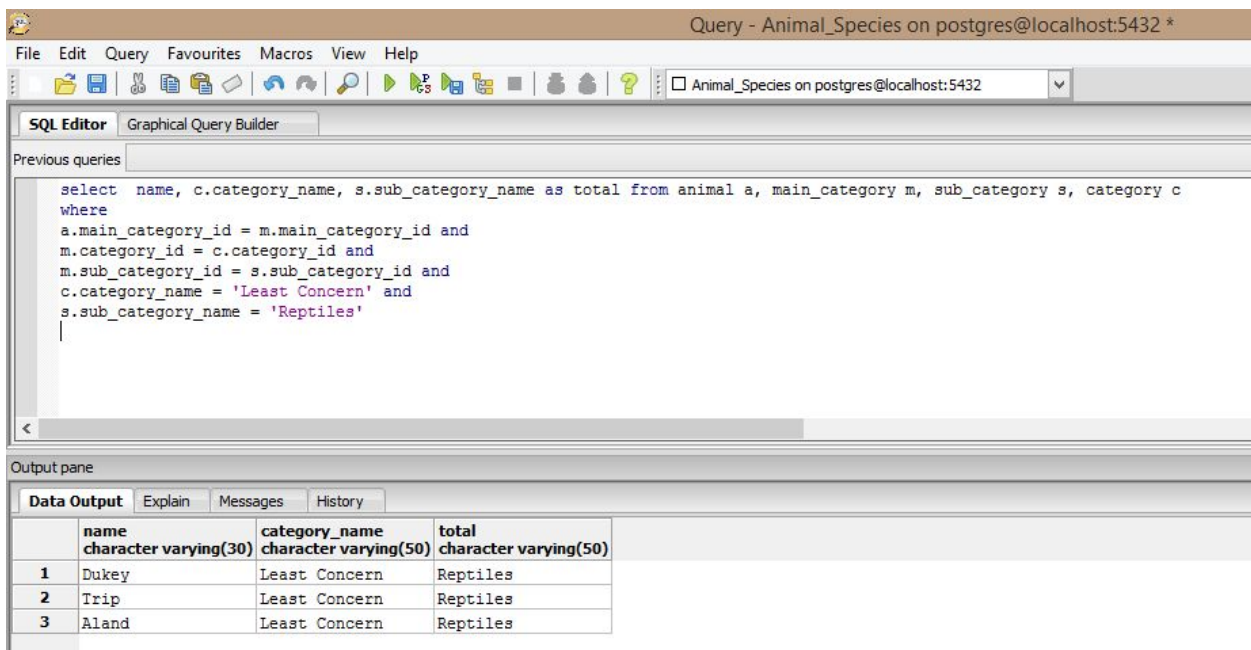
a.main_category_id = m.main_category_id and

m.category_id = c.category_id and

m.sub_category_id = s.sub_category_id and

c.category_name = 'Least Concern' and

s.sub_category_name = 'Reptiles'



The screenshot shows a PostgreSQL SQL Editor window titled "Query - Animal_Species on postgres@localhost:5432 *". The window has a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The SQL Editor tab is active, displaying the following query:

```
select name, c.category_name, s.sub_category_name as total from animal a, main_category m, sub_category s, category c
where
a.main_category_id = m.main_category_id and
m.category_id = c.category_id and
m.sub_category_id = s.sub_category_id and
c.category_name = 'Least Concern' and
s.sub_category_name = 'Reptiles'
```

Below the SQL Editor is the "Output pane" with tabs for "Data Output", "Explain", "Messages", and "History". The "Data Output" tab is selected, showing a table with 4 columns: "name", "category_name", and "total". The table contains 3 rows of data:

	name character varying(30)	category_name character varying(50)	total character varying(50)
1	Dukey	Least Concern	Reptiles
2	Trip	Least Concern	Reptiles
3	Aland	Least Concern	Reptiles

21. Name all the Vulnerable species except Birds.

```
SELECT name,  
       s.sub_category_name AS total  
FROM animal a,  
     main_category m,  
     sub_category s,  
     category c  
WHERE a.main_category_id = m.main_category_id  
     AND m.category_id = c.category_id  
     AND m.sub_category_id = s.sub_category_id  
     AND c.category_name = 'Vulnerable'  
EXCEPT  
(SELECT name,  
       s.sub_category_name AS total  
FROM animal a,  
     main_category m,  
     sub_category s,  
     category c  
WHERE a.main_category_id = m.main_category_id  
     AND m.category_id = c.category_id  
     AND m.sub_category_id = s.sub_category_id  
     AND c.category_name = 'Vulnerable'  
     AND s.sub_category_name = 'Birds' )
```

Previous queries

```
SELECT name,
       s.sub_category_name AS total
FROM animal a,
     main_category m,
     sub_category s,
     category c
WHERE a.main_category_id = m.main_category_id
     AND m.category_id = c.category_id
     AND m.sub_category_id = s.sub_category_id
     AND c.category_name = 'Vulnerable'
EXCEPT
( SELECT name,
       s.sub_category_name AS total
  FROM animal a,
       main_category m,
       sub_category s,
       category c
 WHERE a.main_category_id = m.main_category_id
       AND m.category_id = c.category_id
       AND m.sub_category_id = s.sub_category_id
       AND c.category_name = 'Vulnerable'
       AND s.sub_category_name = 'Birds' )
```

Output pane

Data Output Explain Messages History

	name character varying(30)	total character varying(50)
1	Zuzana	Invertebrates
2	Annabel	Reptiles
3	Arielle	Fishes
4	Hakeem	Reptiles
5	Faunie	Invertebrates
6	Muffin	Fishes

22. Name all the animals whose name length in minimum of all or maximum of all.

```
SELECT name
FROM animal
WHERE char_length(name) =
    (SELECT max(char_length(name))
     FROM animal)
OR char_length(name) =
    (SELECT min(char_length(name))
     FROM animal)
ORDER BY char_length(name)
```

Query -

File Edit Query Favourites Macros View Help

SQL Editor Graphical Query Builder

Previous queries

```
SELECT name
FROM animal
WHERE char_length(name) =
  (SELECT max(char_length(name))
   FROM animal)
OR char_length(name) =
  (SELECT min(char_length(name))
   FROM animal)
ORDER BY char_length(name)
```

Output pane

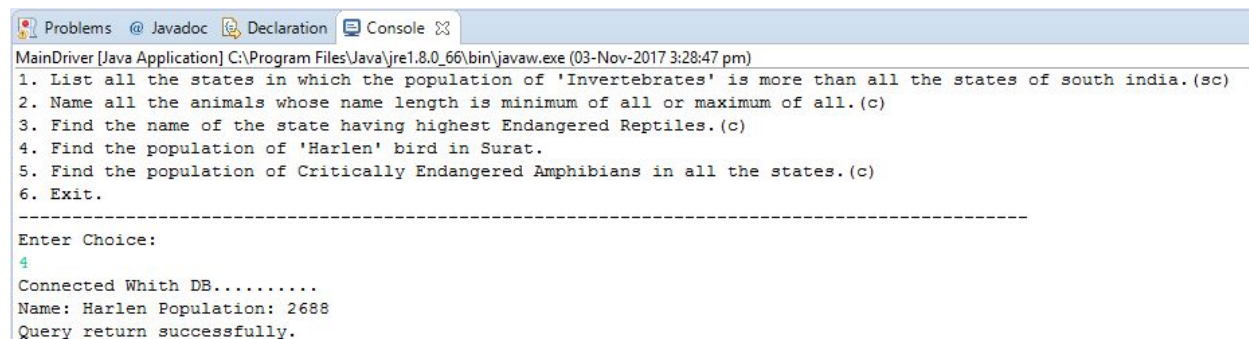
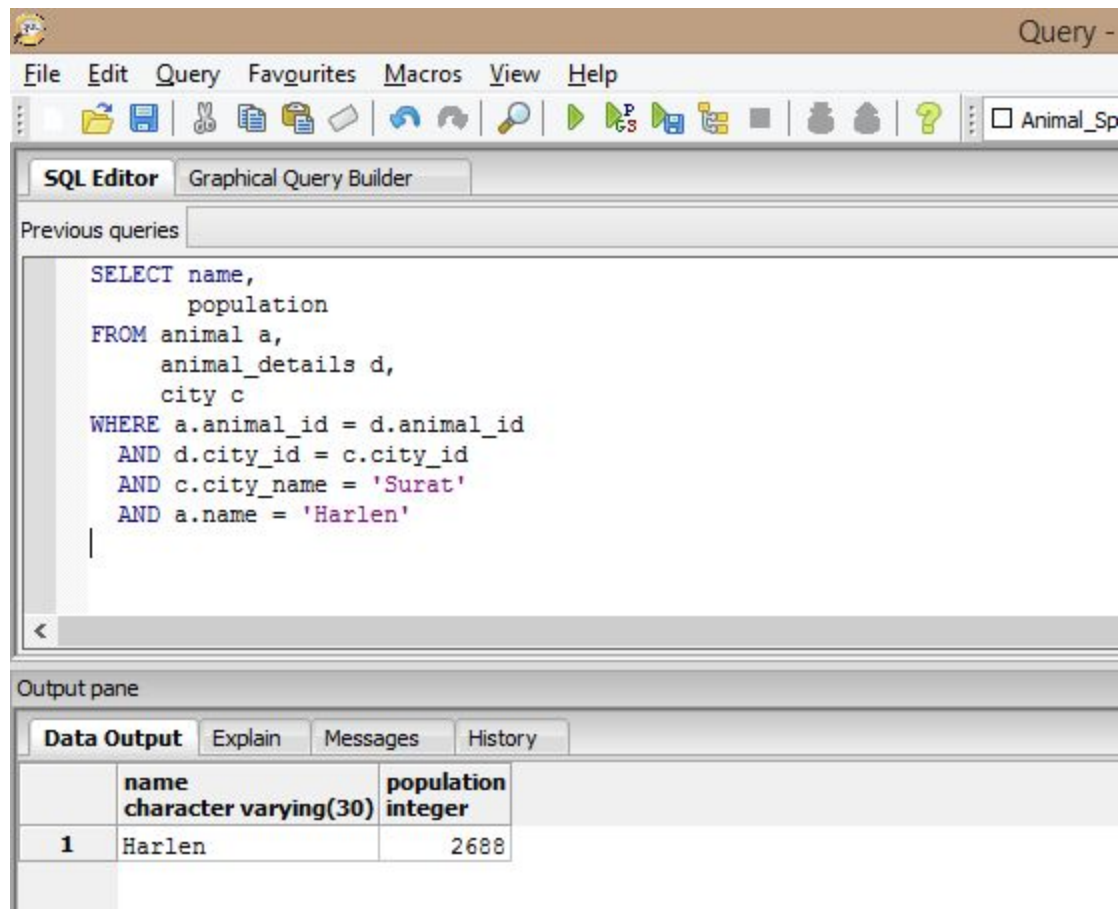
Data Output Explain Messages History

	name character varying(30)
1	May
2	Ham
3	Gar
4	Gui
5	Sanderson
6	Nikoletta
7	Bertrando
8	Ferdinand

```
Problems @ Javadoc Declaration Console
MainDriver [Java Application] C:\Program Files\Java\jre1.8.0_66\bin\javaw.exe (03-Nov-2017 3:28:47 pm)
2. Name all the animals whose name length is minimum of all or maximum of all.(c)
3. Find the name of the state having highest Endangered Reptiles.(c)
4. Find the population of 'Harlen' bird in Surat.
5. Find the population of Critically Endangered Amphibians in all the states.(c)
6. Exit.
-----
Enter Choice:
2
Connected Whith DB.....
Name: May
Name: Ham
Name: Gar
Name: Gui
Name: Sanderson
Name: Nikoletta
Name: Bertrando
Name: Ferdinand
Query return successfully.
```

23. Find the population of 'Harlen' bird in Surat.

```
SELECT name,
       population
FROM animal a,
     animal_details d,
     city c
WHERE a.animal_id = d.animal_id
     AND d.city_id = c.city_id
     AND c.city_name = 'Surat'
     AND a.name = 'Harlen'
```



24. Find the details of the animal whose population is greater than all other animals in Mumbai city.

```
SELECT name,  
       sci_name,  
       age,  
       gender,  
       population  
FROM animal a,  
     animal_details d,  
     city c  
WHERE a.animal_id = d.animal_id  
     AND d.city_id = c.city_id  
     AND c.city_name = 'Mumbai'  
     AND population =  
     (SELECT max(population)  
      FROM animal a,  
            animal_details d,  
            city c  
      WHERE a.animal_id = d.animal_id  
            AND d.city_id = c.city_id  
            AND c.city_name = 'Mumbai' )
```

Query - Animal_Species on postgres

File Edit Query Favurites Macros View Help

Animal_Species on postgres@localhost:5432

SQL Editor Graphical Query Builder

Previous queries

```
SELECT name,
       sci_name,
       age,
       gender,
       population
FROM animal a,
     animal_details d,
     city c
WHERE a.animal_id = d.animal_id
     AND d.city_id = c.city_id
     AND c.city_name = 'Mumbai'
     AND population =
       (SELECT max(population)
        FROM animal a,
             animal_details d,
             city c
        WHERE a.animal_id = d.animal_id
             AND d.city_id = c.city_id
             AND c.city_name = 'Mumbai' )
```

Output pane

Data Output Explain Messages History



























	name character varying(30)	sci_name character varying(100)	age integer	gender character varying(6)	population integer
1	Gert	Smithopsis crassicaudata	13	Female	8138

25. Find the details of the animal whose population is greater than all other animals in Maharashtra state.

```
SELECT name,  
       sci_name,  
       population  
FROM animal a,  
     animal_details d,  
     city c,  
     state s  
WHERE a.animal_id = d.animal_id  
     AND d.city_id = c.city_id  
     AND c.state_id = s.state_id  
     AND s.state_name = 'Maharashtra'  
     AND population =  
       (SELECT max(population)  
        FROM animal a,  
             animal_details d,  
             city c,  
             state s  
        WHERE a.animal_id = d.animal_id  
              AND d.city_id = c.city_id  
              AND c.state_id = s.state_id  
              AND s.state_name = 'Maharashtra' )
```

Query - Anim

FileEditQueryFavouritesMacrosViewHelp



☐ Animal_Species c

SQL EditorGraphical Query Builder

Previous queries

```
SELECT name,
      sci_name,
      population
FROM animal a,
      animal_details d,
      city c,
      state s
WHERE a.animal_id = d.animal_id
      AND d.city_id = c.city_id
      AND c.state_id = s.state_id
      AND s.state_name = 'Maharashtra'
      AND population =
      (SELECT max(population)
       FROM animal a,
            animal_details d,
            city c,
            state s
       WHERE a.animal_id = d.animal_id
            AND d.city_id = c.city_id
            AND c.state_id = s.state_id
            AND s.state_name = 'Maharashtra' )
```

Output pane

Data Output

Explain

Messages

History

	name character varying(30)	sci_name character varying(100)	population integer
1	Leyla	Trichoglossus haematodus moluccanus	9974

26. Find the details of vulnerable birds whose population is maximum in West Bengal.

```
SELECT name,
       sci_name,
       population
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
     AND d.city_id = c.city_id
     AND c.state_id = s.state_id
     AND m.category_id = ca.category_id
     AND m.sub_category_id = sc.sub_category_id
     AND s.state_name = 'West Bengal'
     AND ca.category_name = 'Vulnerable'
     AND sc.sub_category_name = 'Birds'
     AND population =
       (SELECT max(population)
        FROM animal a,
             animal_details d,
             city c,
             state s,
             main_category m,
             category ca,
             sub_category sc
        WHERE a.animal_id = d.animal_id
             AND d.city_id = c.city_id
             AND c.state_id = s.state_id
             AND m.category_id = ca.category_id
             AND m.sub_category_id = sc.sub_category_id
             AND s.state_name = 'West Bengal'
             AND ca.category_name = 'Vulnerable'
             AND sc.sub_category_name = 'Birds' )
```

Previous queries

▼

Delete

```
SELECT name,
       sci_name,
       population FROM animal a,
       animal_details d,
       city c,
       state s,
       main_category m,
       category ca,
       sub_category sc WHERE a.animal_id = d.animal_id
AND d.city_id = c.city_id AND c.state_id = s.state_id AND m.category_id = ca.category_id
AND m.sub_category_id = sc.sub_category_id AND s.state_name = 'West Bengal'
AND ca.category_name = 'Vulnerable' AND sc.sub_category_name = 'Birds'
AND population =
  (SELECT max(population)
   FROM animal a,
        animal_details d,
        city c,
        state s,
        main_category m,
        category ca,
        sub_category sc
   WHERE a.animal_id = d.animal_id AND d.city_id = c.city_id AND c.state_id = s.state_id
        AND m.category_id = ca.category_id AND m.sub_category_id = sc.sub_category_id AND s.state_name = 'West Bengal'
        AND ca.category_name = 'Vulnerable' AND sc.sub_category_name = 'Birds' )
```

Output pane

Data Output

Explain

Messages

History

	name character varying(30)	sci_name character varying(100)	population integer
1	Dominica	Cyrtodactylus lousiadensis	9261

27. Find the total population of mammals in Karnataka.

```
SELECT sum(population)
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     sub_category sc
WHERE a.animal_id = d.animal_id
     AND a.main_category_id = m.main_category_id
     AND d.city_id = c.city_id
     AND c.state_id = s.state_id
     AND m.sub_category_id = sc.sub_category_id
     AND s.state_name = 'Karnataka'
     AND sc.sub_category_name = 'Mammals'
```

SQL Editor

Previous queries

```
SELECT sum(population)
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     sub_category sc
WHERE a.animal_id = d.animal_id
      AND a.main_category_id = m.main_category_id
      AND d.city_id = c.city_id
      AND c.state_id = s.state_id
      AND m.sub_category_id = sc.sub_category_id
      AND s.state_name = 'Karnataka'
      AND sc.sub_category_name = 'Mammals'
```

Output pane

Data Output Explain Messages History

	sum bigint
1	31189

28. Find the name of the state having highest Endangered Reptiles.

```
SELECT s.state_name,
       population,
       ca.category_name,
       sc.sub_category_name
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
     AND a.main_category_id = m.main_category_id
     AND d.city_id = c.city_id
     AND c.state_id = s.state_id
     AND m.category_id = ca.category_id
     AND m.sub_category_id = sc.sub_category_id
     AND ca.category_name = 'Endangered'
     AND sc.sub_category_name = 'Reptiles'
     AND population =
       (SELECT max(population)
        FROM animal a,
             animal_details d,
             city c,
             state s,
             main_category m,
             category ca,
             sub_category sc
        WHERE a.animal_id = d.animal_id
             AND a.main_category_id = m.main_category_id
             AND d.city_id = c.city_id
             AND c.state_id = s.state_id
             AND m.category_id = ca.category_id
             AND m.sub_category_id = sc.sub_category_id
             AND ca.category_name = 'Endangered'
             AND sc.sub_category_name = 'Reptiles' )
```

Query - Animal

File Edit Query Favourites Macros View Help

☐ Animal_Species on p

SQL Editor

Graphical Query Builder

Previous queries

```

SELECT s.state_name,population,ca.category_name,sc.sub_category_name
FROM animal a,animal_details d,city c,state s,main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
      AND a.main_category_id = m.main_category_id
      AND d.city_id = c.city_id
      AND c.state_id = s.state_id
      AND m.category_id = ca.category_id
      AND m.sub_category_id = sc.sub_category_id
      AND ca.category_name = 'Endangered'
      AND sc.sub_category_name = 'Reptiles'
      AND population =
      (SELECT max(population)
       FROM animal a,animal_details d,city c,state s,main_category m,
            category ca,
            sub_category sc
       WHERE a.animal_id = d.animal_id
            AND a.main_category_id = m.main_category_id
            AND d.city_id = c.city_id
            AND c.state_id = s.state_id
            AND m.category_id = ca.category_id
            AND m.sub_category_id = sc.sub_category_id
            AND ca.category_name = 'Endangered'
            AND sc.sub_category_name = 'Reptiles' )

```

Output pane

Data Output Explain Messages History

	state_name character varying(50)	population integer	category_name character varying(50)	sub_category_name character varying(50)
1	Punjab	9123	Endangered	Reptiles

OK.


```
Problems @ Javadoc Declaration Console
MainDriver [Java Application] C:\Program Files\Java\jre1.8.0_66\bin\javaw.exe (03-Nov-2017 3:28:47 pm)
6.City
7.Complex Queries
8.Exit
Enter your Choice (1-8)
7
-----
1. List all the states in which the population of 'Invertebrates' is more than all the states of south india. (sc)
2. Name all the animals whose name length is minimum of all or maximum of all. (c)
3. Find the name of the state having highest Endangered Reptiles. (c)
4. Find the population of 'Harlen' bird in Surat.
5. Find the population of Critically Endangered Amphibians in all the states. (c)
6. Exit.
-----
Enter Choice:
3
Connected Whith DB.....
State Name: Punjab Population: 9123 Category Name: Endangered Sub Category NameReptiles
Query return successffully.
```

29. Find the population of Critically Endangered Amphibians in all the states.

```
SELECT s.state_name,
       population
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
     AND a.main_category_id = m.main_category_id
     AND d.city_id = c.city_id
     AND c.state_id = s.state_id
     AND m.category_id = ca.category_id
     AND m.sub_category_id = sc.sub_category_id
     AND ca.category_name = 'Critically Endangered'
     AND sc.sub_category_name = 'Amphibians'
```

SQL Editor

```

SELECT s.state_name,
       population
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
      AND a.main_category_id = m.main_category_id
      AND d.city_id = c.city_id
      AND c.state_id = s.state_id
      AND m.category_id = ca.category_id
      AND m.sub_category_id = sc.sub_category_id
      AND ca.category_name = 'Critically Endangered'
      AND sc.sub_category_name = 'Amphibians'

```

Output pane

Data Output

	state_name character varying(50)	population integer
1	Andhra Pradesh	2301
2	Tripura	6148
3	Gujarat	5280
4	Assam	3374
5	Tamil Nadu	7076
6	Kerala	5111
7	Punjab	2371
8	Nagaland	5981
9	Himachal Pradesh	5554
10	Himachal Pradesh	2569

```
Problems @ Javadoc Declaration Console
MainDriver [Java Application] C:\Program Files\Java\jre1.8.0_66\bin\javaw.exe (03-Nov-2017 3:28:47 pm)
4. Find the population of 'Harlen' bird in Surat.
5. Find the population of Critically Endangered Amphibians in all the states.(c)
6. Exit.
-----
Enter Choice:
5
Connected Whith DB.....
Name: Andhra Pradesh Population: 2301
Name: Tripura Population: 6148
Name: Gujarat Population: 5280
Name: Assam Population: 3374
Name: Tamil Nadu Population: 7076
Name: Kerala Population: 5111
Name: Punjab Population: 2371
Name: Nagaland Population: 5981
Name: Himachal Pradesh Population: 5554
Name: Himachal Pradesh Population: 2569
Query return successfully.
```

30. List all the animals whose population is greater than 5000.

```
SELECT name
FROM animal
WHERE animal_id = ANY
  (SELECT animal_id
   FROM animal_details
   WHERE population > 5000)
```

SQL Editor

```

SELECT name
FROM animal
WHERE animal_id = ANY
  (SELECT animal_id
   FROM animal_details
   WHERE population > 500)

```

Output pane

Data Output

	name character varying(30)
1	Annabel
2	Faunie
3	Sterne
4	Maura
5	Bertrando
6	Doralynn
7	Monti
8	Torrence
9	Paola
10	Monah
11	Deena
12	May
13	Michaela
14	Georgy
15	Zorana
16	Hughie
17	Emlynn

31. List all the states in which the population of 'Invertebrates' is more than all the states of south india.

```
SELECT sum(population),
       state_name
FROM animal a,
     animal_details d,
     city c,
     state s,
     main_category m,
     category ca,
     sub_category sc
WHERE a.animal_id = d.animal_id
     AND a.main_category_id = m.main_category_id
     AND d.city_id = c.city_id
     AND c.state_id = s.state_id
     AND m.category_id = ca.category_id
     AND m.sub_category_id = sc.sub_category_id
     AND sc.sub_category_name = 'Invertebrates'
GROUP BY state_name
HAVING sum(population) > ALL
      (SELECT sum(population)
       FROM animal a,
            animal_details d,
            city c,
            state s,
            main_category m,
            category ca,
            sub_category sc
       WHERE a.animal_id = d.animal_id
            AND a.main_category_id = m.main_category_id
            AND d.city_id = c.city_id
            AND c.state_id = s.state_id
            AND m.category_id = ca.category_id
            AND m.sub_category_id = sc.sub_category_id
            AND sc.sub_category_name = 'Invertebrates'
            AND s.state_name IN ('Andhra Pradesh',
                                'Karnataka',
                                'Kerala',
                                'Tamil Nadu', 'Telangana'))
GROUP BY state_name)
```

Query - Animal_Species on postgres@localhost:5432

File Edit Query Favurites Macros View Help

SQL Editor Graphical Query Builder Execute query

Previous queries

```

SELECT sum(population),
       state_name
FROM animal a,animal_details d,city c,state s,main_category m,category ca,sub_category sc
WHERE a.animal_id = d.animal_id
      AND a.main_category_id = m.main_category_id
      AND d.city_id = c.city_id
      AND c.state_id = s.state_id
      AND m.category_id = ca.category_id
      AND m.sub_category_id = sc.sub_category_id
      AND sc.sub_category_name = 'Invertebrates'
GROUP BY state_name
HAVING sum(population) > ALL
  (SELECT sum(population)
   FROM animal a,animal_details d,city c,state s,main_category m,category ca,sub_category sc
   WHERE a.animal_id = d.animal_id
         AND a.main_category_id = m.main_category_id
         AND d.city_id = c.city_id
         AND c.state_id = s.state_id
         AND m.category_id = ca.category_id
         AND m.sub_category_id = sc.sub_category_id
         AND sc.sub_category_name = 'Invertebrates'
         AND s.state_name IN ('Andhra Pradesh','Karnataka','Kerala','Tamil Nadu','Telangana'))
GROUP BY state_name)

```

Output pane

Data Output Explain Messages History

	sum bigint	state_name character varying(50)
1	42661	Chhattisgarh
2	49774	Gujarat
3	39220	Manipur
4	37772	Uttar Pradesh

Problems @ Javadoc Declaration Console

MainDriver [Java Application] C:\Program Files\Java\jre1.8.0_66\bin\javaw.exe (03-Nov-2017 3:28:47 pm)

Enter your Choice (1-8)

7

1. List all the states in which the population of 'Invertebrates' is more than all the states of south india.(sc)
2. Name all the animals whose name length is minimum of all or maximum of all.(c)
3. Find the name of the state having highest Endangered Reptiles.(c)
4. Find the population of 'Harlen' bird in Surat.
5. Find the population of Critically Endangered Amphibians in all the states.(c)
6. Exit.

Enter Choice:

1

Connected Whith DB.....

Population: 42661 and State Name : Chhattisgarh

Population: 49774 and State Name : Gujarat

Population: 39220 and State Name : Manipur

Population: 37772 and State Name : Uttar Pradesh

Query return successfully.

