

Name: Rohan Vinayak Chaudhari

Batch: Data Engineering

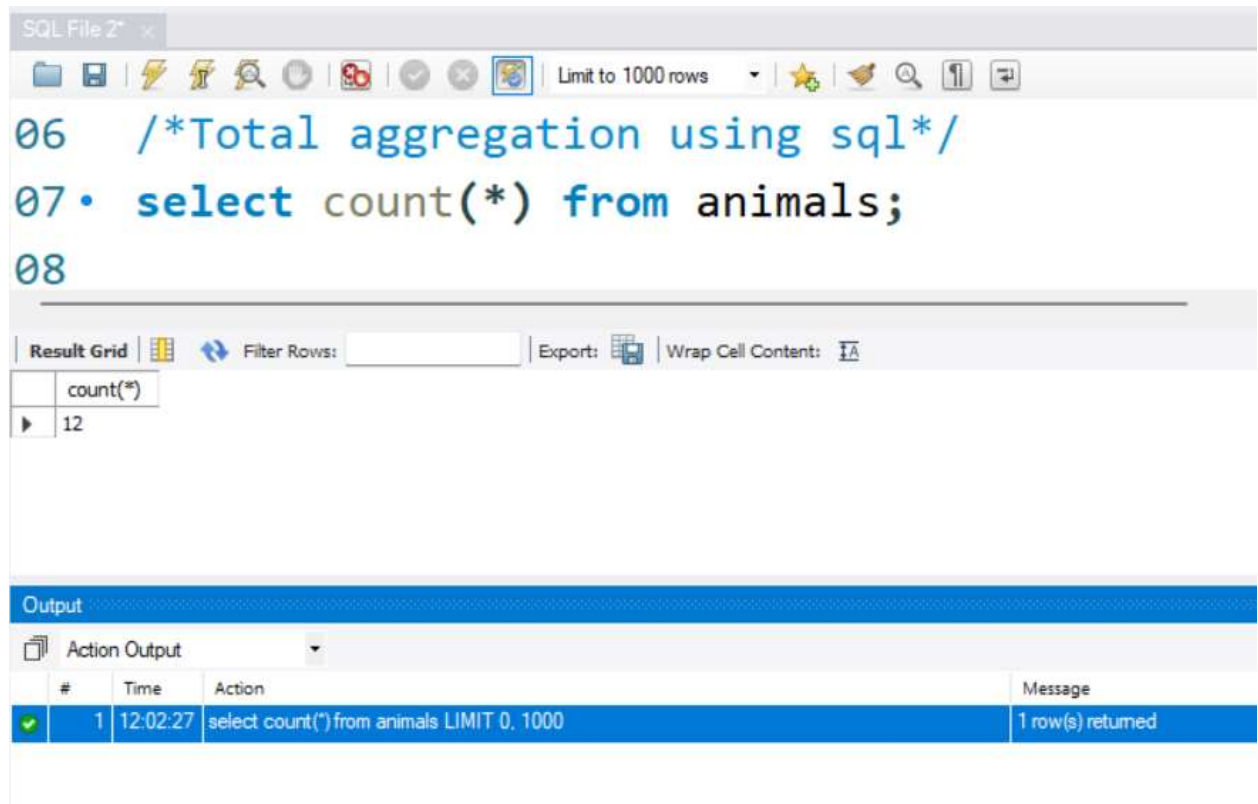
Date:25/01/2024

Topic: MYSQL(AGGREGATION,OVER,PARTITION,REGEX,SUBTOTALS,MATERIALIZED VIEWS)

Solution:

## 1. MYSQL:

Total Aggregations using SQL Queries:



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and settings. The query editor contains the following SQL code:

```
06  /*Total aggregation using sql*/
07  • select count(*) from animals;
08
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query:

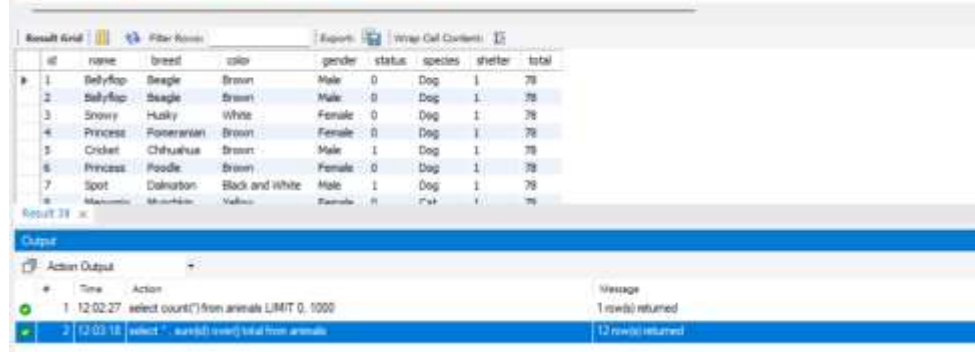
	count(*)
▶	12

At the bottom, the 'Output' tab is active, showing the 'Action Output' log:

#	Time	Action	Message
✓ 1	12:02:27	select count(*) from animals LIMIT 0, 1000	1 row(s) returned

## OVER CLAUSE:

```
29 /*Over clause */
10 select * , sum(id) over() total from animals;
11
12 /*partition clause*/
```



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with icons for file operations and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL query is displayed. The 'Result Grid' shows a table with 8 columns: id, name, breed, color, gender, status, species, shelter, and total. The data is as follows:

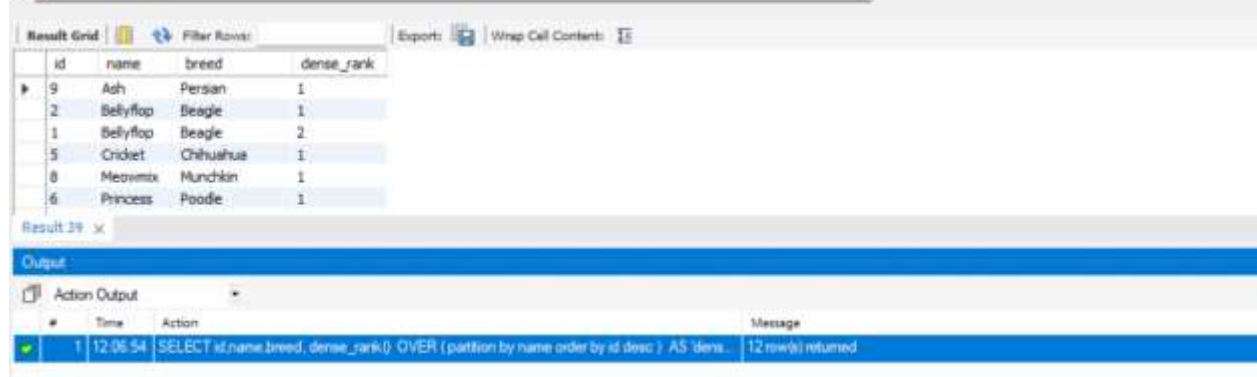
id	name	breed	color	gender	status	species	shelter	total
1	Bellyflop	Beagle	Brown	Male	0	Dog	1	78
2	Bellyflop	Beagle	Brown	Male	0	Dog	1	78
3	Snowy	Husky	White	Female	0	Dog	1	78
4	Princess	Pomeranian	Brown	Female	0	Dog	1	78
5	Cricket	Chihuahua	Brown	Male	1	Dog	1	78
6	Princess	Poodle	Brown	Female	0	Dog	1	78
7	Spot	Dalmatian	Black and White	Male	1	Dog	1	78
8	Meowmix	Munchkin	Various	Female	0	Cat	1	78

Below the result grid, the 'Output' section shows the 'Action Output' log. It contains two entries:

#	Time	Action	Message
1	12:02:27	select count(*) from animals LIMIT 0, 1000	1 row(s) returned
2	12:03:18	select * , sum(id) over() total from animals	12 row(s) returned

## PARTITION CLAUSE

```
12 /*partition clause*/
13 SELECT id,name,breed, dense_rank()
14 OVER ( partition by name order by id desc )
15 AS 'dense_rank' FROM animals;
16
```



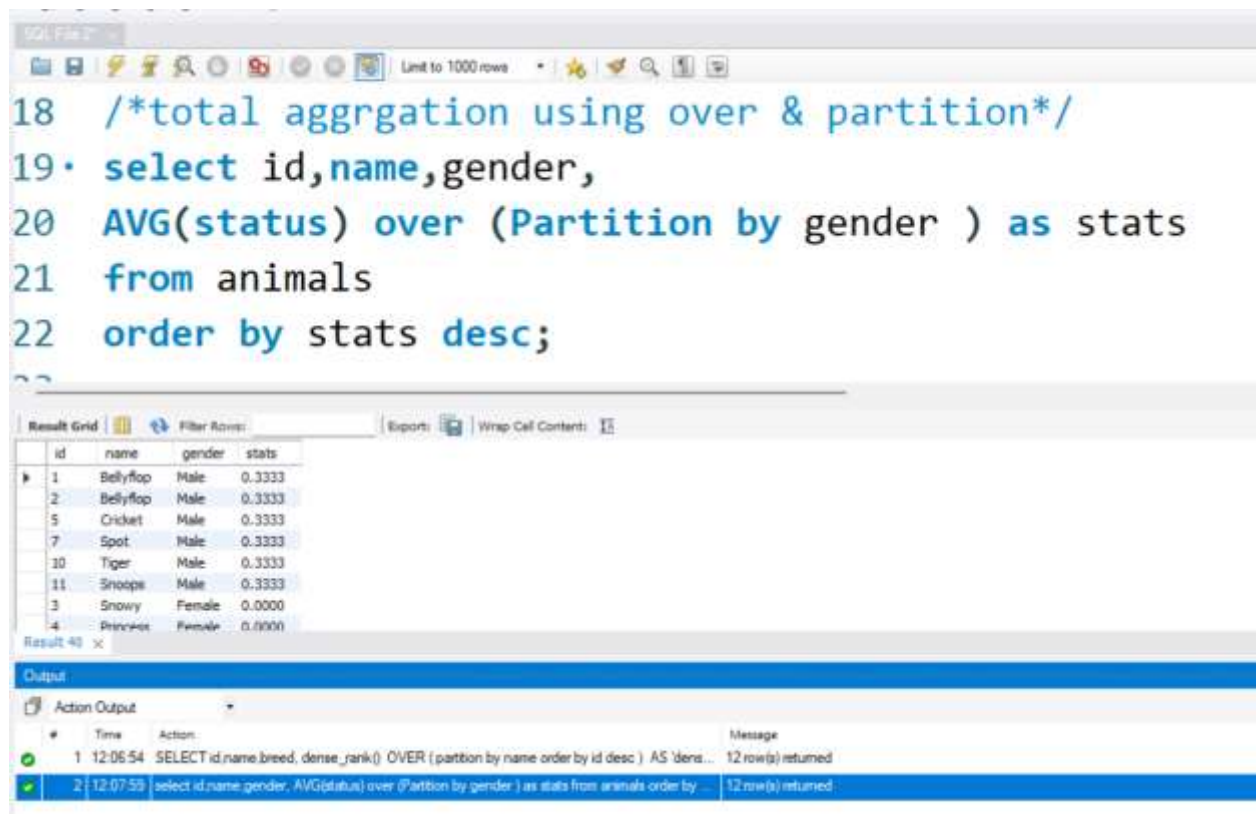
The screenshot shows a SQL IDE interface. At the top, there's a toolbar with icons for file operations and a 'Limit to 1000 rows' dropdown. Below the toolbar, the SQL query is displayed. The 'Result Grid' shows a table with 4 columns: id, name, breed, and dense\_rank. The data is as follows:

id	name	breed	dense_rank
9	Ash	Persian	1
2	Bellyflop	Beagle	1
1	Bellyflop	Beagle	2
5	Cricket	Chihuahua	1
8	Meowmix	Munchkin	1
6	Princess	Poodle	1

Below the result grid, the 'Output' section shows the 'Action Output' log. It contains one entry:

#	Time	Action	Message
1	12:06:54	SELECT id,name,breed, dense_rank() OVER ( partition by name order by id desc ) AS 'dense_rank' FROM animals;	12 row(s) returned

total aggregation using over & partition:



The screenshot shows a SQL IDE interface. The top pane contains a SQL query. The bottom pane is split into two sections: 'Result Grid' and 'Output'.

**SQL Query:**

```
18 /*total aggrgation using over & partition*/
19 select id,name,gender,
20 AVG(status) over (Partition by gender ) as stats
21 from animals
22 order by stats desc;
```

**Result Grid:**

	id	name	gender	stats
1	Bellyflop	Male	0.3333	
2	Bellyflop	Male	0.3333	
5	Cricket	Male	0.3333	
7	Spot	Male	0.3333	
10	Tiger	Male	0.3333	
11	Snoops	Male	0.3333	
3	Snowy	Female	0.0000	
4	Princess	Female	0.0000	

**Output:**

#	Time	Action	Message
1	12:06:54	SELECT id,name,breed, dense_rank() OVER ( partition by name order by id desc ) AS 'dens...	12 row(s) returned
2	12:07:55	select id,name,gender, AVG(status) over (partition by gender ) as stats from animals order by ...	12 row(s) returned

## ORDER OF EXECUTION:

```
SQL File 2" x
Limit to 1000 rows

25  /*order of execution*/
26  • select                               /*5*/
27    gender,count(*) as count
28  from                               /*1*/
29    animals a
30  join
31    adoptions ad
32  on
33    a.id = ad.animal_id
34  where                               /*2*/
35    gender like 'Male'                /*4*/
36  group by gender                     /*3*/
37  Order by count desc                /*6*/
38  limit 1;                           /*7*/
```

## OUTPUT:

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	gender	count				
▶	Male	2				

Result 41 x				Output
Action Output				
#	Time	Action	Message	
✓ 1	12:09:23	select /*5*/ gender,count(*) as count from /*1*/ animals a join adoptions ad ...	1 row(s) returned	

## SUBTOTALS:

The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
139
140 /*subtotals*/
141 /* it is giving total adopted pets of each gender null is the grand total */
142
143 * SELECT gender, SUM(status) AS Total FROM animals
144     GROUP BY gender with rollup;
145
146
```

The results pane displays a table with two columns: 'gender' and 'Total'. The data is as follows:

gender	Total
Female	0
Male	2
Rollup	2

The results pane also shows a message: "SELECT gender, SUM(status) AS Total FROM animals GROUP BY gender with rollup; 3 rows returned".

## REGEX:

SQL File 7

Limit to 1000 rows

```
145
146
147 /*regex*/
148 /*finds the animal with either color crown or yellow*/
149 * SELECT * FROM animals WHERE color REGEXP 'Brown|yellow';
150
151 /*finds the name starting with b*/
152 * select * from animals where name regexp '^b';
```

Result Grid

	id	name	breed	color	gender	status	species	shelter
▶	1	Bellyflop	Beagle	Brown	Male	0	Dog	1
	2	Bellyflop	Beagle	Brown	Male	0	Dog	1
	4	Princess	Pomeranian	Brown	Female	0	Dog	1
	5	Crocket	Chihuahua	Brown	Male	1	Dog	1
	6	Princess	Poodle	Brown	Female	0	Dog	1
	8	Meowmix	Munchkin	Yellow	Female	0	Cat	1
	10	Tiger	Bengal	Brown	Male	0	Cat	1
	11	Snoops	Beagle	Brown	Male	0	Dog	2

animals 43 x

Output

Action Output

#	Time	Action	Message
1	12:09:58	SELECT gender, SUM(status) AS Total FROM animals GROUP BY gender with rollup LIMIT 0, 1000	3 row(s) returned
2	12:10:32	SELECT * FROM animals WHERE color REGEXP 'Brown yellow' LIMIT 0, 1000	8 row(s) returned

SQL File 7

Limit to 1000 rows

```
148 /*finds the animal with either color crown or yellow*/
149 * SELECT * FROM animals WHERE color REGEXP 'Brown|yellow';
150
151 /*finds the name starting with b*/
152 * select * from animals where name regexp '^b';
153
154
155 /*to search from end*/
```

Result Grid

	id	name	breed	color	gender	status	species	shelter
▶	1	Bellyflop	Beagle	Brown	Male	0	Dog	1
	2	Bellyflop	Beagle	Brown	Male	0	Dog	1

animals 45 x

Output

Action Output

#	Time	Action	Message
1	12:12:03	select * from animals where name regexp "b" LIMIT 0, 1000	2 row(s) returned

SQL File 2

Limit to 1000 rows

```

:52 * select * from animals where name regexp '^b';
:53
:54
:55 /*to search from end*/
:56 * select * from animals where name regexp 'ss$';
:57
:58
:59

```

Result Grid

#	id	name	breed	color	gender	status	species	shelter
4	Princess	Pomeranian	Brown	Female	0	Dog	1	
5	Princess	Poodle	Brown	Female	0	Dog	1	

animals 47 x

Output

Action Output

#	Time	Action	Message
1	12:12:45	select * from animals where name regexp '^b' LIMIT 0, 1000	2 row(s) returned
2	12:14:22	select * from animals where name regexp 'ss\$' LIMIT 0, 1000	2 row(s) returned

SQL File 2'

Limit to 1000 rows

```

:57
:58
:59
:60 /*anything containing b*/
:61 * select * from animals where name regexp 'b';
:62
:63
:64

```

Result Grid

#	id	name	breed	color	gender	status	species	shelter
1	Belyflop	Beagle	Brown	Male	0	Dog	1	
2	Belyflop	Beagle	Brown	Male	0	Dog	1	

animals 48 x

Output

Action Output

#	Time	Action	Message
1	12:15:01	select * from animals where name regexp 'b' LIMIT 0, 1000	2 row(s) returned



SQL Fiddle

Limit to 1000 rows

```
63
64
65
66
67 /*to find consecutive digit*/
68 * SELECT 'It124contains3consicutive67digits' REGEXP '[0-9]{3}';
69
70
```

Result Grid

	It124contains3consicutive67digits' REGEXP '[0-9]{3}
1	

Result 49

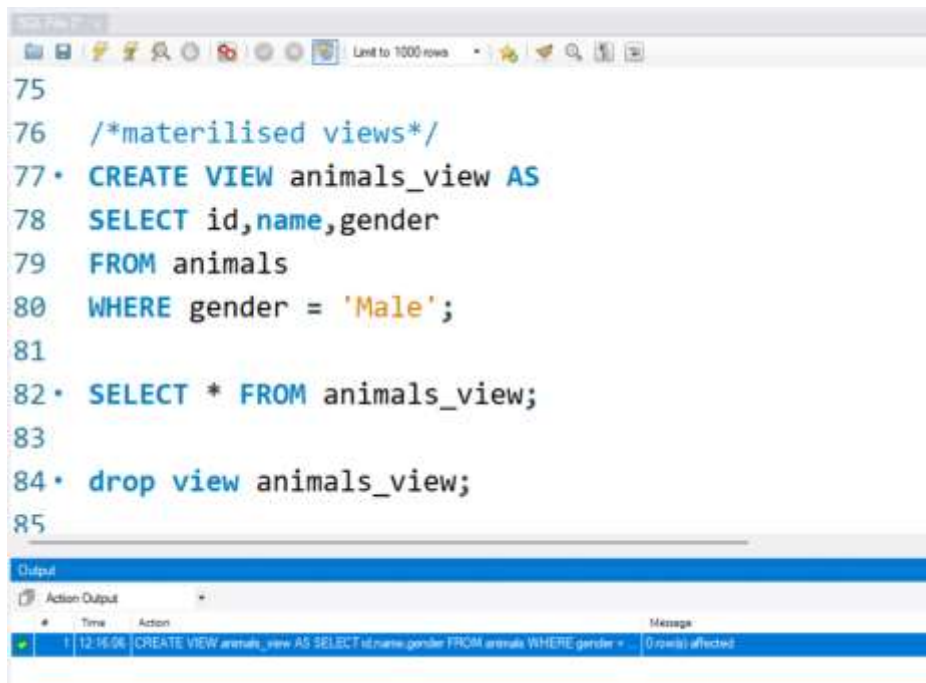
Output

Action Output

#	Time	Action	Message
1	12:15:01	select * from animals where name regexp b' LIMIT 0, 1000	2 row(s) returned
2	12:15:30	SELECT 'It124contains3consicutive67digits' REGEXP '[0-9]{3}' LIMIT 0, 1000	1 row(s) returned



## MATERIALIZED VIEW:

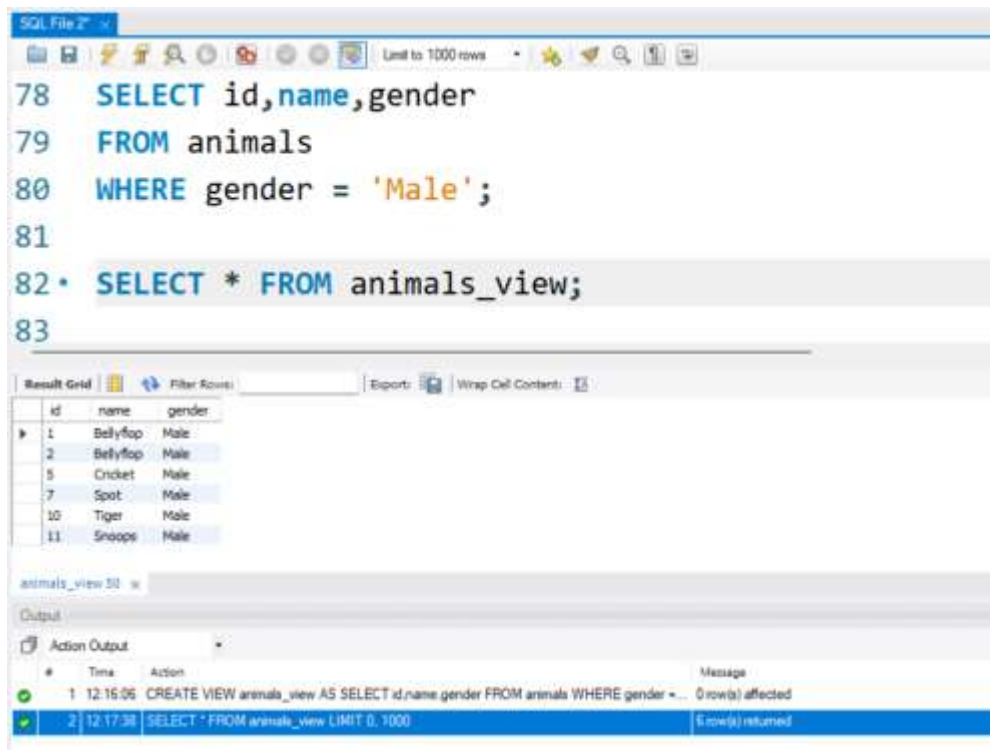


The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The main editor contains SQL code with line numbers 75 through 85. The code defines a materialized view named 'animals\_view' based on the 'animals' table, filtered by 'gender = Male'. It then queries the view and drops it. Below the editor is an 'Output' pane with a tab for 'Action Output'. It displays a single log entry with a green status icon, a timestamp of 12:16:06, the executed SQL statement, and a message indicating '0 rows affected'.

```
75
76  /*materilised views*/
77 • CREATE VIEW animals_view AS
78  SELECT id,name,gender
79  FROM animals
80  WHERE gender = 'Male';
81
82 • SELECT * FROM animals_view;
83
84 • drop view animals_view;
85
```

#	Time	Action	Message
1	12:16:06	CREATE VIEW animals_view AS SELECT id,name,gender FROM animals WHERE gender = 'Male';	0 rows affected

## RETRIVEING VIEWS:



The screenshot shows a SQL File 2 window with the following queries:

```
78 SELECT id,name,gender
79 FROM animals
80 WHERE gender = 'Male';
81
82 • SELECT * FROM animals_view;
83
```

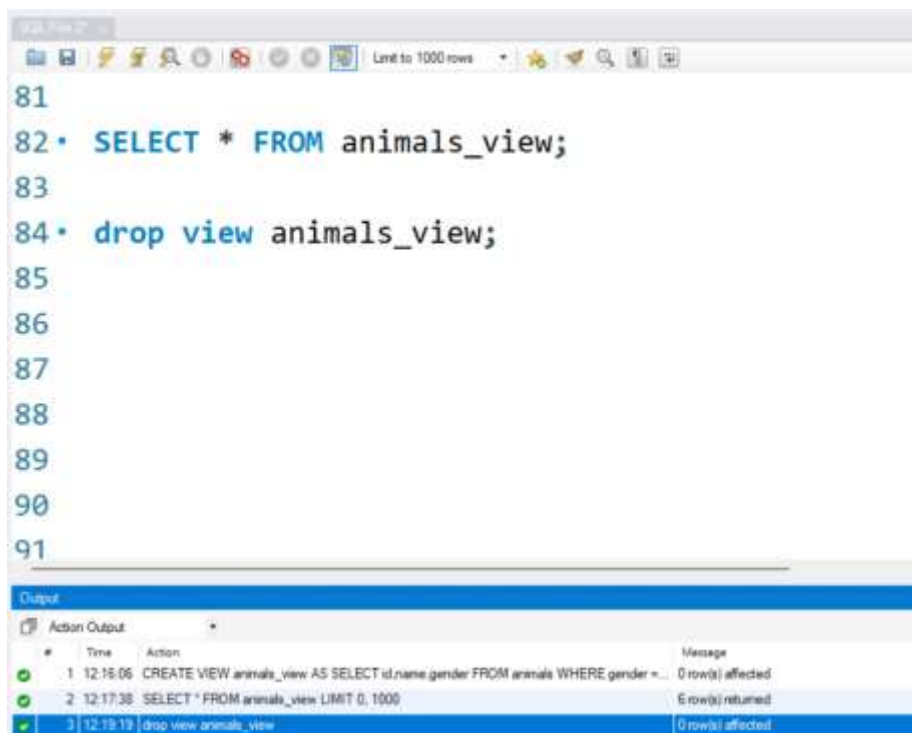
The Result Grid shows the following data:

	id	name	gender
1	1	Bellyflop	Male
2	2	Bellyflop	Male
5	5	Cricket	Male
7	7	Spot	Male
10	10	Tiger	Male
11	11	Snoops	Male

The Output window shows the following messages:

#	Time	Action	Message
1	12:16:06	CREATE VIEW animals_view AS SELECT id,name,gender FROM animals WHERE gender = ...	0 row(s) affected
2	12:17:38	SELECT * FROM animals_view LIMIT 0, 1000	6 row(s) returned

## DROP VIEW:



The screenshot shows a SQL File 2 window with the following queries:

```
81
82 • SELECT * FROM animals_view;
83
84 • drop view animals_view;
85
86
87
88
89
90
91
```

The Output window shows the following messages:

#	Time	Action	Message
1	12:16:06	CREATE VIEW animals_view AS SELECT id,name,gender FROM animals WHERE gender = ...	0 row(s) affected
2	12:17:38	SELECT * FROM animals_view LIMIT 0, 1000	6 row(s) returned
3	12:19:19	drop view animals_view	0 row(s) affected

