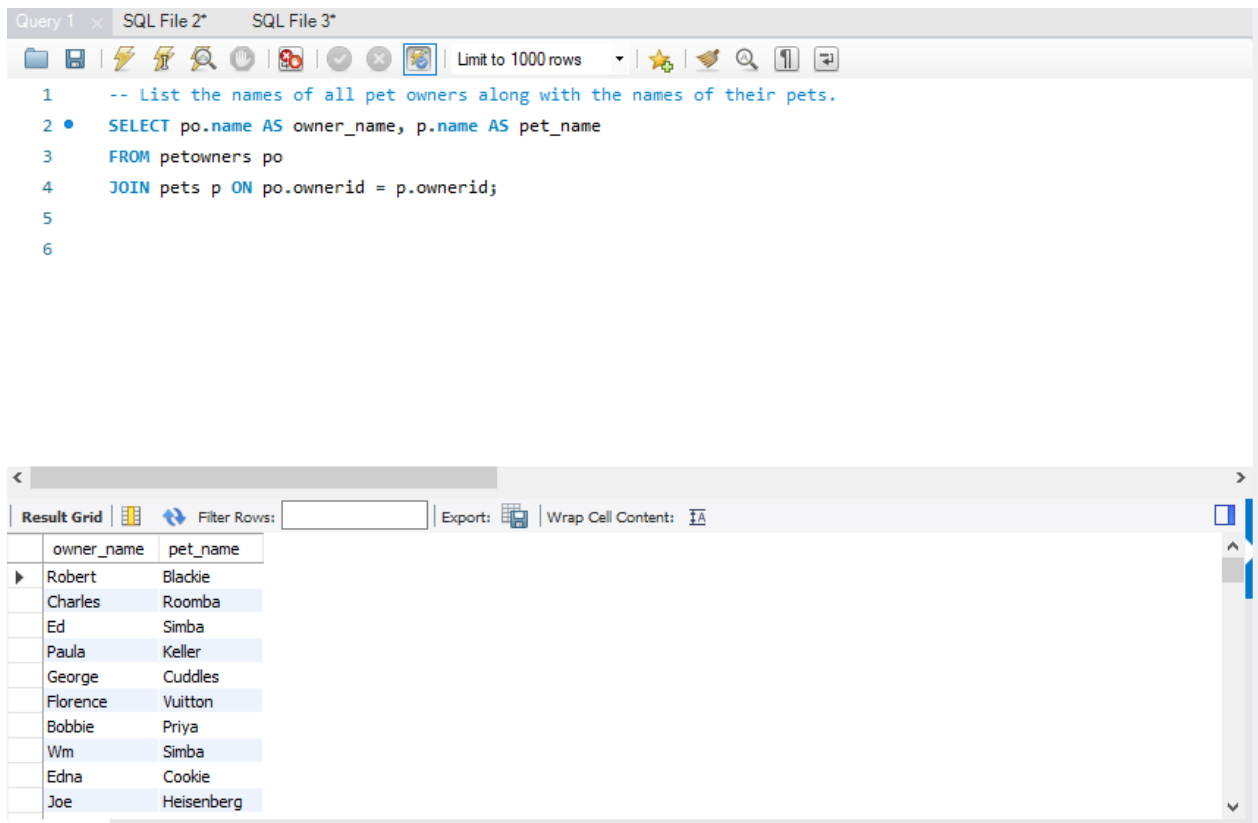


Q:1 List the names of all pet owners along with the names of their pets.



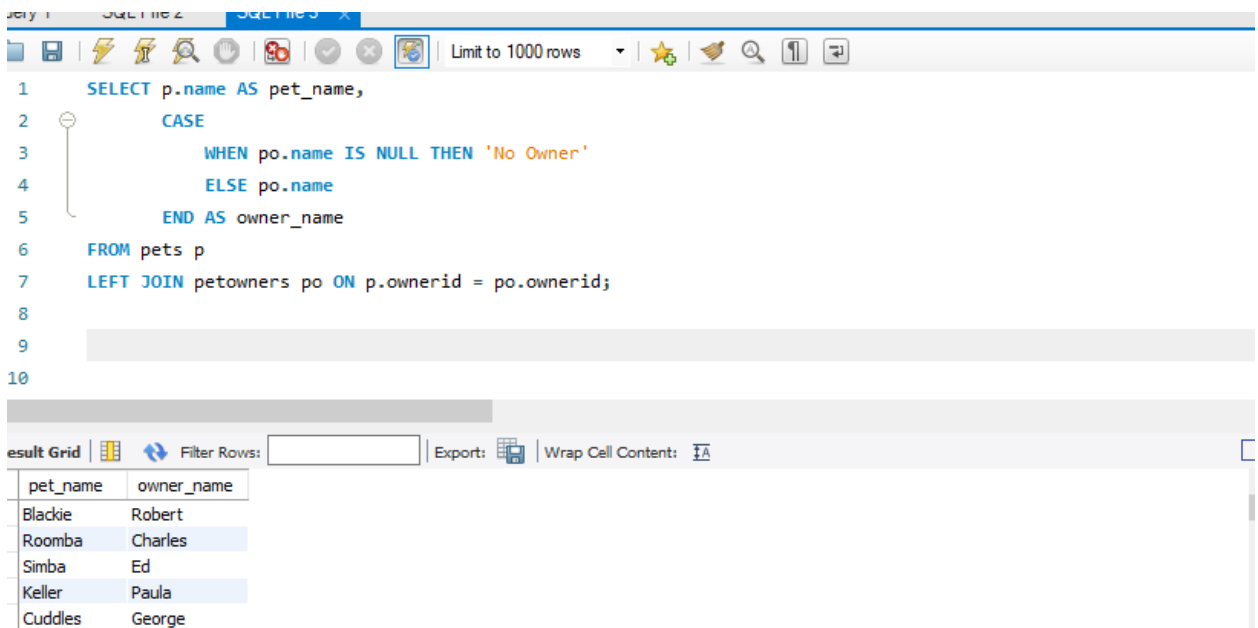
The screenshot shows a SQL IDE with a query editor and a results grid. The query editor contains the following SQL code:

```
1  -- List the names of all pet owners along with the names of their pets.
2  • SELECT po.name AS owner_name, p.name AS pet_name
3  FROM petowners po
4  JOIN pets p ON po.ownerid = p.ownerid;
5
6
```

The results grid displays the following data:

owner_name	pet_name
Robert	Blackie
Charles	Roomba
Ed	Simba
Paula	Keller
George	Cuddles
Florence	Vuitton
Bobbie	Priya
Wm	Simba
Edna	Cookie
Joe	Heisenberg

Q:2 List all pets and their owner names, including pets that don't have recorded owners



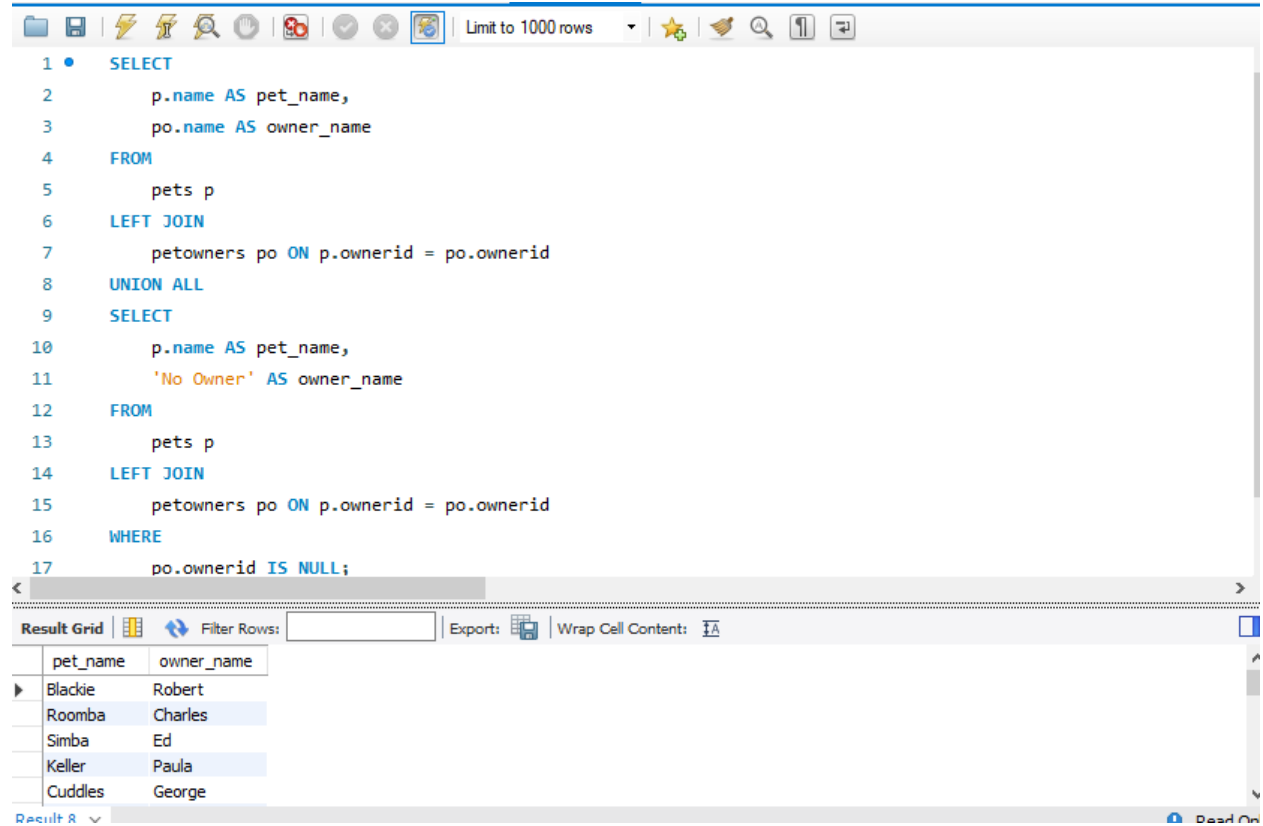
The screenshot shows a SQL IDE with a query editor and a results grid. The query editor contains the following SQL code:

```
1  SELECT p.name AS pet_name,
2  CASE
3      WHEN po.name IS NULL THEN 'No Owner'
4      ELSE po.name
5  END AS owner_name
6  FROM pets p
7  LEFT JOIN petowners po ON p.ownerid = po.ownerid;
8
9
10
```

The results grid displays the following data:

pet_name	owner_name
Blackie	Robert
Roomba	Charles
Simba	Ed
Keller	Paula
Cuddles	George

Q:3 Combine the information of pets and their owners, including those pets without owners and owners without pets.



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 a SQL query that combines data from the 'pets' and 'petowners' tables using a UNION ALL. The query is as follows:

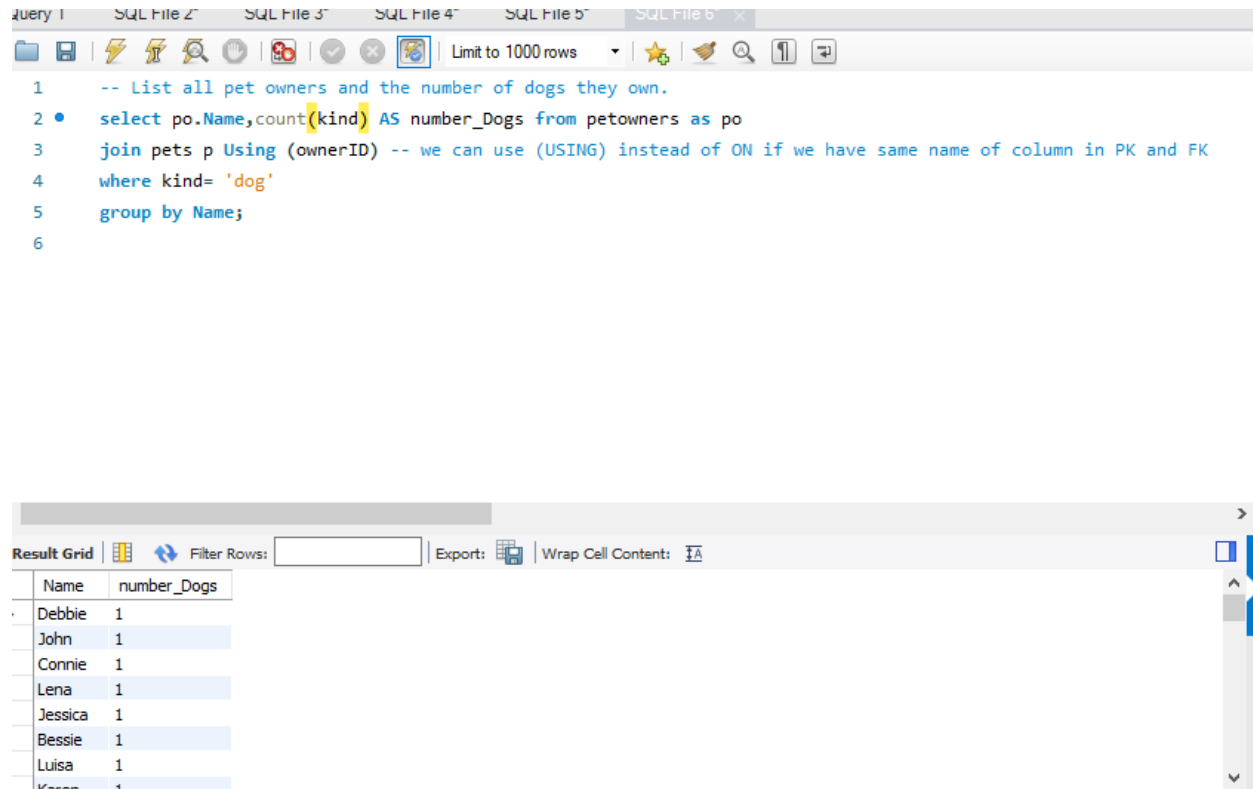
```
1 • SELECT
2     p.name AS pet_name,
3     po.name AS owner_name
4 FROM
5     pets p
6 LEFT JOIN
7     petowners po ON p.ownerid = po.ownerid
8 UNION ALL
9 SELECT
10    p.name AS pet_name,
11    'No Owner' AS owner_name
12 FROM
13     pets p
14 LEFT JOIN
15     petowners po ON p.ownerid = po.ownerid
16 WHERE
17     po.ownerid IS NULL;
```

Below the editor, the 'Result Grid' tab is active, displaying the query results in a table with two columns: 'pet\_name' and 'owner\_name'. The results are as follows:

pet_name	owner_name
Blackie	Robert
Roomba	Charles
Simba	Ed
Keller	Paula
Cuddles	George

At the bottom left, it says 'Result 8' with a dropdown arrow. At the bottom right, there is a 'Read On' button.

Q:4 List all pet owners and the number of dogs they own.



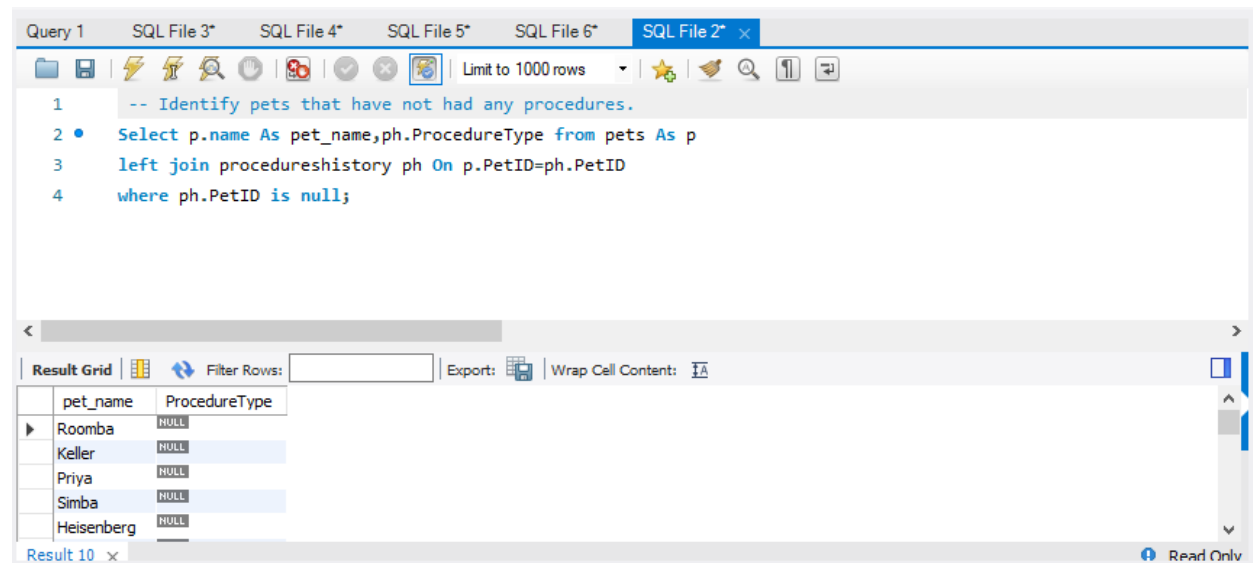
The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 -- List all pet owners and the number of dogs they own.
2 • select po.Name,count(kind) AS number_Dogs from petowners as po
3 join pets p Using (ownerID) -- we can use (USING) instead of ON if we have same name of column in PK and FK
4 where kind= 'dog'
5 group by Name;
6
```

The result grid displays the following data:

Name	number_Dogs
Debbie	1
John	1
Connie	1
Lena	1
Jessica	1
Bessie	1
Luisa	1
Karen	1

Q:5 Identify pets that have not had any procedures.



The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 -- Identify pets that have not had any procedures.
2 • Select p.name As pet_name,ph.ProcedureType from pets As p
3 left join procedureshistory ph On p.PetID=ph.PetID
4 where ph.PetID is null;
```

The result grid displays the following data:

pet_name	ProcedureType
Roomba	NULL
Keller	NULL
Priya	NULL
Simba	NULL
Heisenberg	NULL

Q:6 Find the name of the oldest pet.

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

```
1 -- Find the name of the oldest pet.
2 • select * from pets;
3 • select name from pets
4 where age =(select max(age) from pets);
5
```

The results pane shows a table with one column, 'name', and three rows of data:

name
Stowe
Tiger
Simba

Q 6 can also be done on this query

SELECT name AS oldest\_pet\_name

FROM pets

ORDER BY age DESC

LIMIT 1;

Q:7 Find the details of procedures performed on 'Cuddles'.

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

```
1 -- Find the details of procedures performed on 'Cuddles'.
2 • select p.name As petName,ProcedureType From Pets As p
3 join procedureshistory ph on p.PetID=ph.PetID
4 where Name='cuddles';
```

The results pane shows a table with two columns, 'petName' and 'ProcedureType', and three rows of data:

petName	ProcedureType
Cuddles	ORTHOPEDIC
Cuddles	VACCINATIONS
Cuddles	VACCINATIONS

Q:8 List the pets who have undergone a procedure called 'VACCINATIONS'

Query 1   SQL File 3\*   SQL File 4\*   SQL File 5\*   SQL File 6\*   SQL File 2\*   SQL File 7\*   SQL File 8\*   SQL File 9\* x

Limit to 1000 rows

```
1 -- List the pets who have undergone a procedure called 'VACCINATIONS'
2 • select Name,procedureType from pets As p
3 join procedureshistory ph on p.PetID=ph.PetID
4 where ProcedureType='Vaccinations';
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:   Read Only

	Name	procedureType
▶	Humbert	VACCINATIONS
	Jake	VACCINATIONS
	Bandit	VACCINATIONS
	Cuddles	VACCINATIONS
	Rumba	VACCINATIONS

Result 2 x

Q:9 Count the number of pets of each kind

ry 1   SQL File 10\* x   SQL File 3\*   SQL File 4\*   SQL File 5\*   SQL File 6\*   SQL File 2\*   SQL File 7\*   SQL File 8\*   SQL File 9\*

Limit to 1000 rows

```
1 -- Count the number of pets of each kind
2 • select kind,count(*) As Total_pet from pets
3 group by kind;
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:   Read Only

	kind	Total_pet
▶	Dog	57
	Cat	31
	Parrot	12

Q:10 Group pets by their kind and gender and count the number of pets in each group.

SQL File 4\* SQL File 5\* SQL File 6\* SQL File 2\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\*

Limit to 1000 rows

```

1  -- Group pets by their kind and gender and count the number of pets in each group.
2  • select kind,gender,count(name) from pets
3    group by kind,gender;

```

Result Grid

Filter Rows:  Export: Wrap Cell Content:

kind	gender	count(name)
Dog	male	35
Cat	male	19
Parrot	female	7
Cat	female	12
Dog	female	22

Result 1 x

Q:11 Show the average age of pets for each kind, but only for kinds that have more than 5 pets

SQL File 5\* SQL File 6\* SQL File 2\* SQL File 7\* SQL File 8\* SQL File 9\* SQL File 10\* SQL File 11\*

Limit to 1000 rows

```

1  -- Show the average age of pets for each kind, but only for kinds that have more than 5
2  -- pets
3  • select kind,avg(age) As average_age from pets
4    group by kind
5    having count(*)>5;

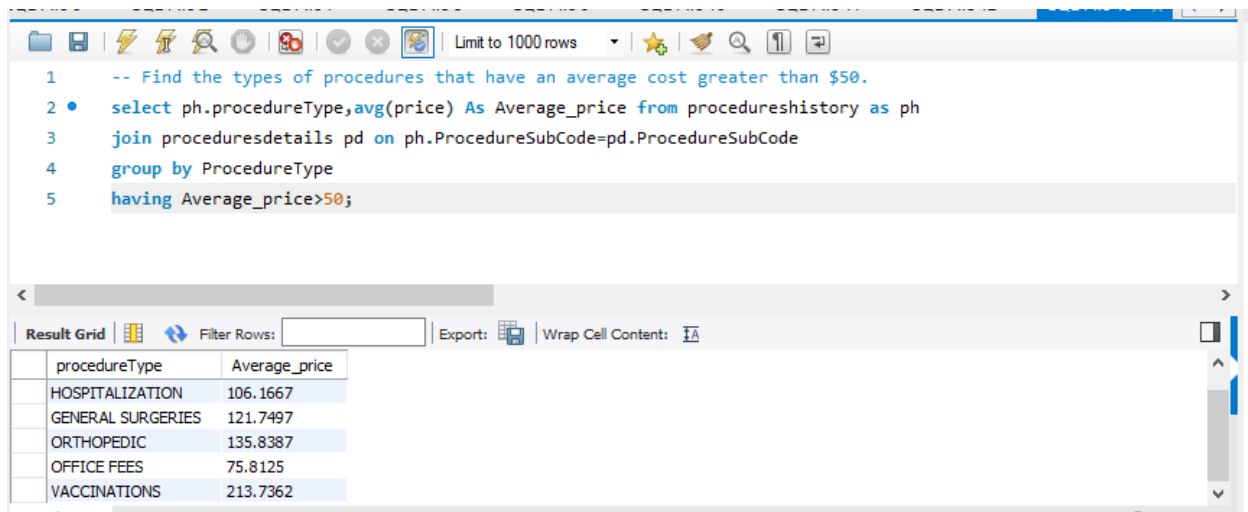
```

Result Grid

Filter Rows:  Export: Wrap Cell Content:

kind	average_age
Dog	6.7895
Cat	7.3226
Parrot	6.5833

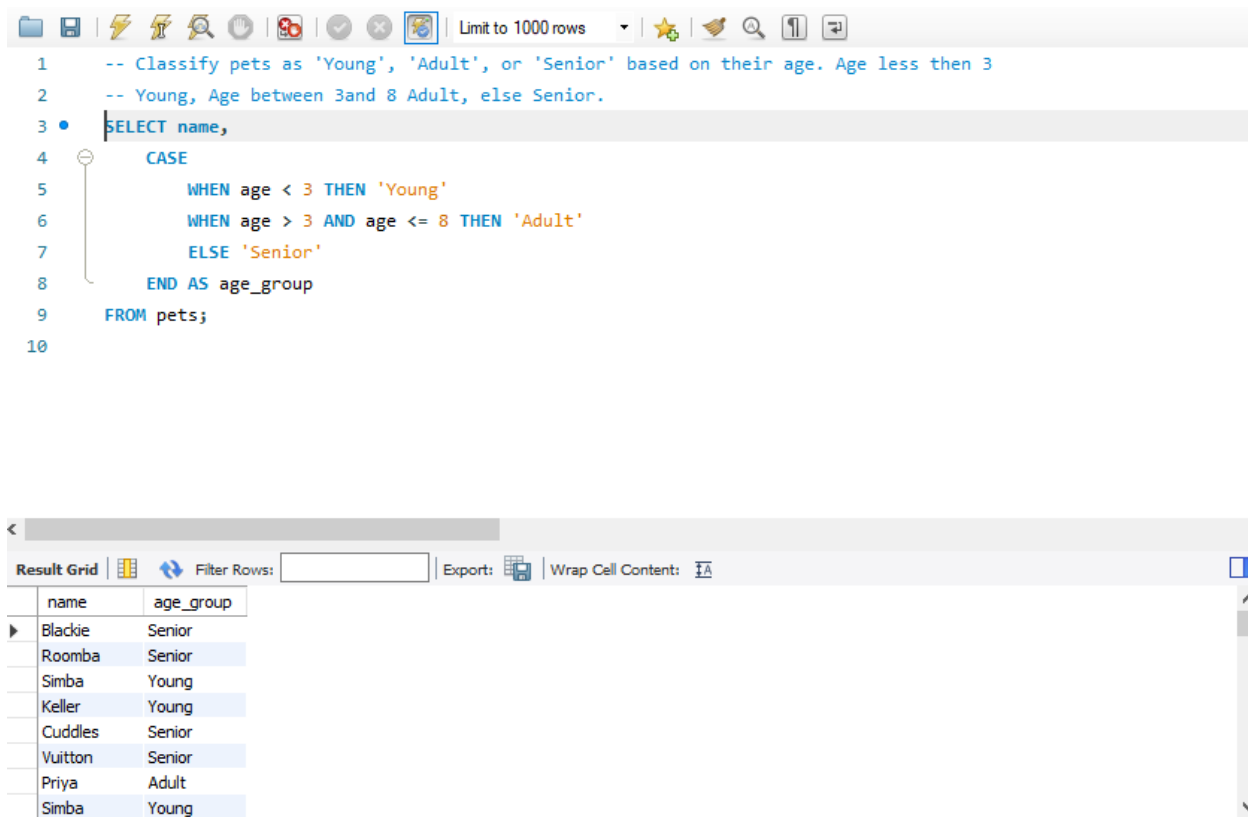
Q:12 Find the types of procedures that have an average cost greater than \$50.



The screenshot shows a SQL IDE window with a query editor and a results grid. The query is designed to find the types of procedures with an average cost greater than \$50. The results grid displays the following data:

procedureType	Average_price
HOSPITALIZATION	106.1667
GENERAL SURGERIES	121.7497
ORTHOPEDIC	135.8387
OFFICE FEES	75.8125
VACCINATIONS	213.7362

Q:13 Classify pets as 'Young', 'Adult', or 'Senior' based on their age. Age less than 3 Young, Age between 3 and 8 Adult, else Senior.



The screenshot shows a SQL IDE window with a query and its results. The query uses a CASE statement to classify pets based on their age. The results grid displays the following data:

name	age_group
Blackie	Senior
Roomba	Senior
Simba	Young
Keller	Young
Cuddles	Senior
Vuitton	Senior
Priya	Adult
Simba	Young

Q:14 Show the gender of pets with a custom label ('Boy' for male, 'Girl' for female).

```
1 -- Show the gender of pets with a custom label ('Boy' for male, 'Girl' for female).
2 • Select Name as pet_name,
3     CASE
4         WHEN Gender='Male' Then 'boy'
5         Else 'Girl'
6     End As Label_Group
7 from pets;
8
```

Result Grid

pet_name	Label_Group
Blackie	boy
Roomba	boy
Simba	boy
Keller	Girl
Cuddles	boy
Vuitton	Girl
Priya	Girl
Simba	boy
Cookie	Girl
Heisenberg	boy
Stowe	Girl
Scout	Girl
Lily	Girl
Danger	boy
Danger	boy
Scooter	boy

Q:15 For each pet, display the pet's name, the number of procedures they've had, and a status label: 'Regular' for pets with 1 to 3 procedures, 'Frequent' for 4 to 7 procedures, and 'Super User' for more than 7 procedures.

```
1 • SELECT
2     p.name AS pet_name,
3     COUNT(ph.proceduretype) AS num_procedures,
4     CASE
5         WHEN COUNT(ph.procedureType) BETWEEN 1 AND 3 THEN 'Regular'
6         WHEN COUNT(ph.procedureType) BETWEEN 4 AND 7 THEN 'Frequent'
7         ELSE 'Super User'
8     END AS status_label
9 from pets ph;
```

Result Grid

pet_name	num_procedures	status_label
HOLD	2243	Super User
Humbert	1	Regular
Jake	2	Regular
Cuddles	1	Regular
Bandit	1	Regular