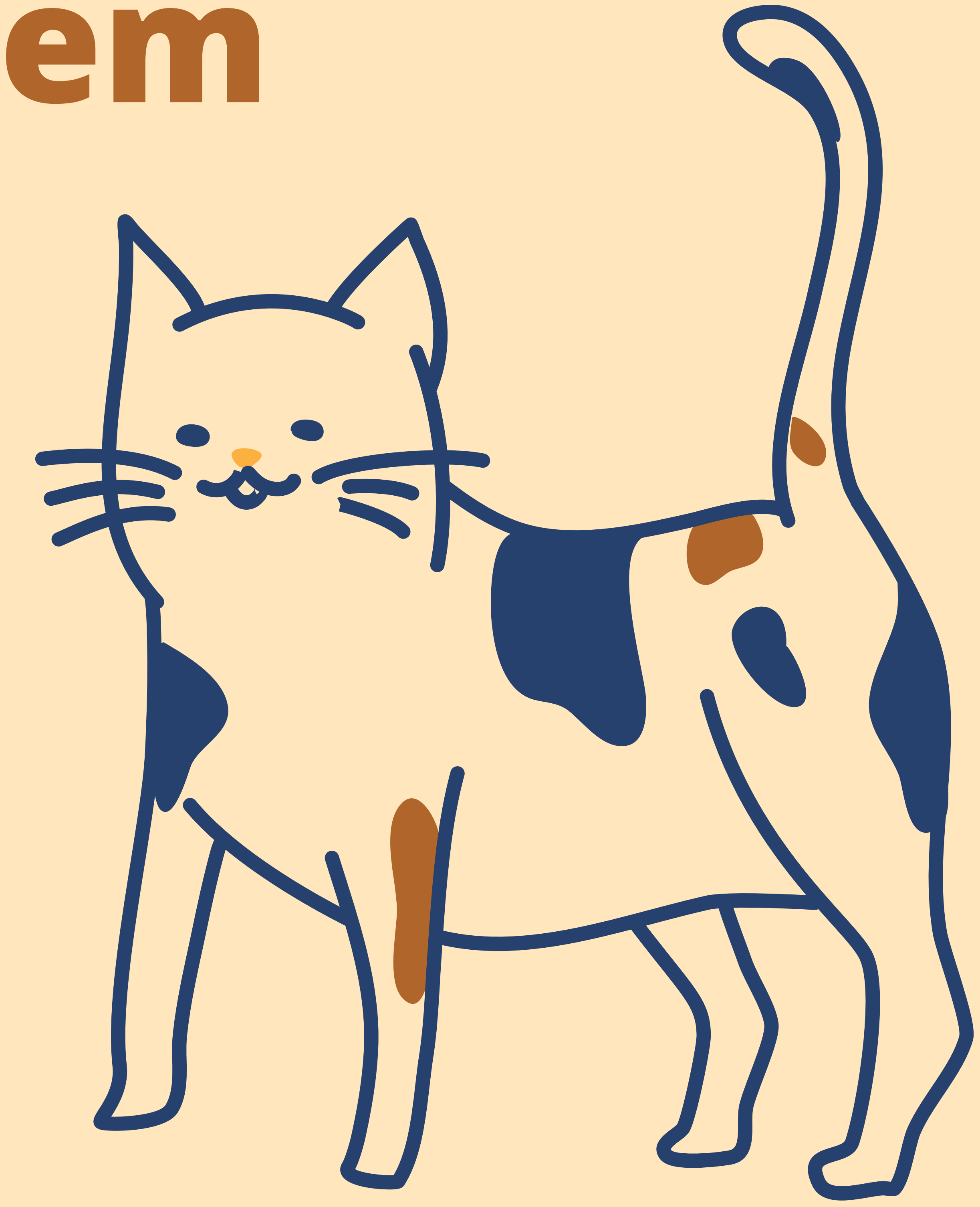
Pet Hospitality System

Data Base Project

SECTION: SE3
CCCS 215

GROUP MEMBERS & ID

Refal Najmi	2006271
Asool Rajab	2005623
Mai Alqurashi	2008060
Joury Alsulimani	2006121



Problem Description

Many pet owners struggle to find places to host their pets. Whenever a pet owner is exposed to any circumstance, they are forced to leave their pets on the street or put it up for adoption.

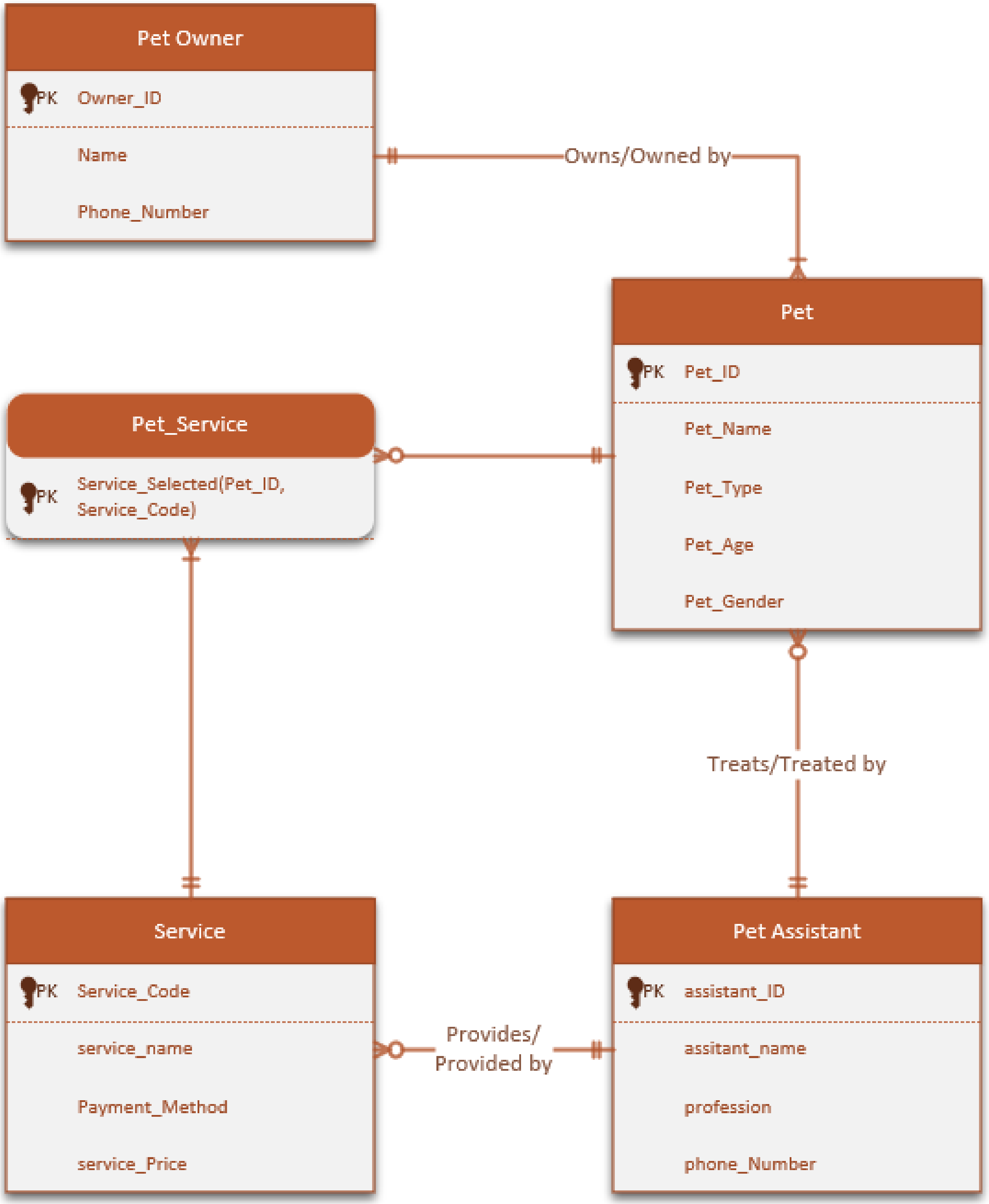
We collectively decided to create a reliable pet hospitality application that allows pet owners to drop their pets at. The application strives to provide a safe and secure environment for the pets that incorporates many services including pet kennels, boarding, day-care, grooming, and a veterinary clinic.

We have considered that the application is the link between pet owners and tenants for ease and flexibility of handling. Our idea is likely to be the ideal solution to this problem, given the studies and statistics we have done

Business Rule

- In a pet hospitality application, a pet owner can have at least one pet or more, yet each pet must be owned by only one owner.
- A pet is treated by only one assistant always and forever.
- A pet assistant can treat zero, one, or many pets.
- One pet assistant can provide zero, one, or more services.
- A service is managed by only one pet assistant.
- A pet can have one or more services.
- A service is provided to at least one or many pets

ER



Pet Owner

PK Owner_ID

Name

Phone_Number

Owns/Owned by

Pet

PK Pet_ID

Pet_Name

Pet_Type

Pet_Age

Pet_Gender

Pet_Service

PK Service_Selected(Pet_ID, Service_Code)

Treats/Treated by

Service

PK Service_Code

service_name

Payment_Method

service_Price

Provides/ Provided by

Pet Assistant

PK assistant_ID

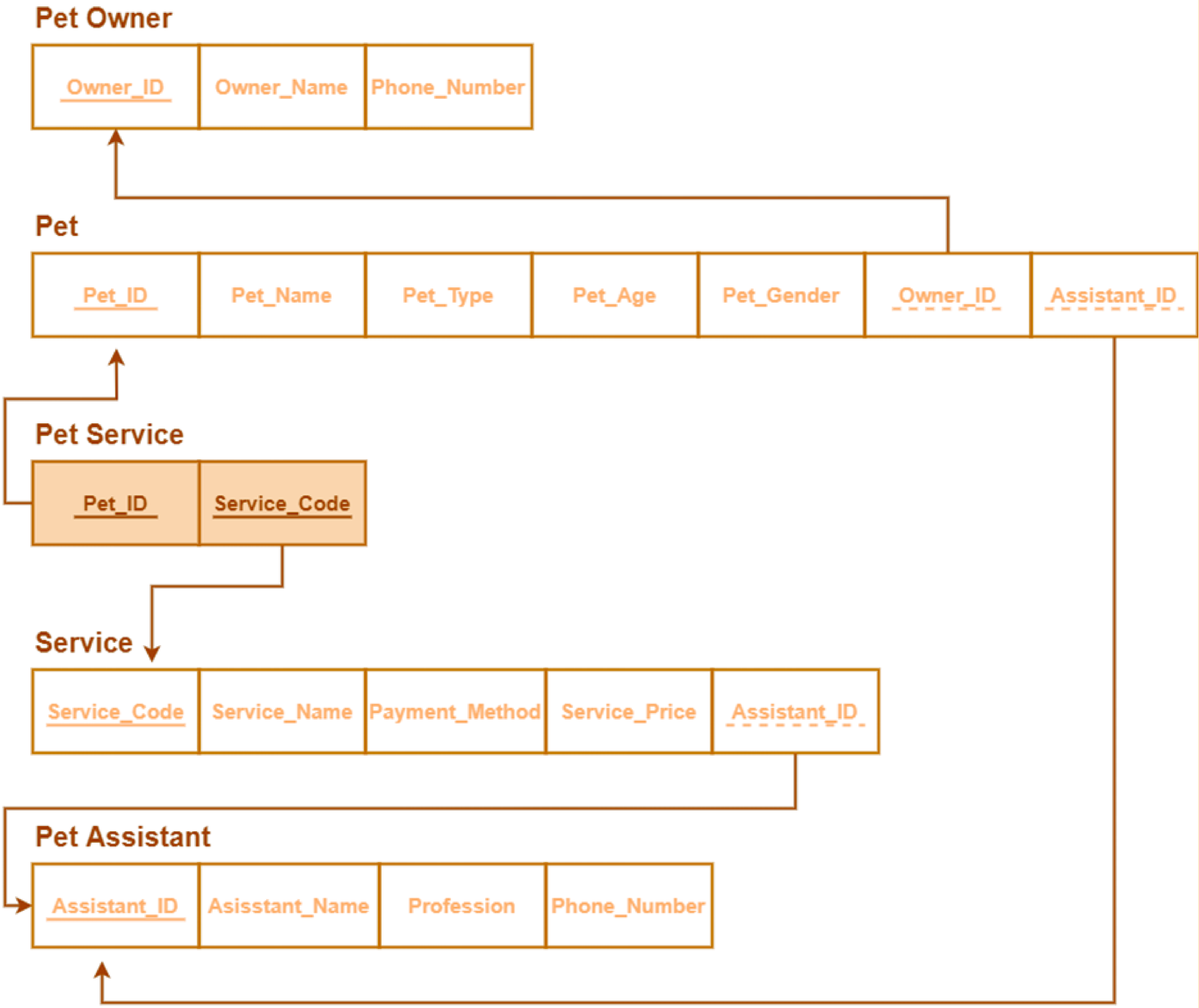
assitant_name

profession

phone_Number

Diagram

Relations



Pet Owner

Owner_ID	Owner_Name	Phone_Number

Pet

Pet_ID	Pet_Name	Pet_Type	Pet_Age	Pet_Gender	Owner_ID	Assistant_ID

Pet Service

Pet_ID	Service_Code

Service

Service_Code	Service_Name	Payment_Method	Service_Price	Assistant_ID

Pet Assistant

Assistant_ID	Asisstant_Name	Profession	Phone_Number

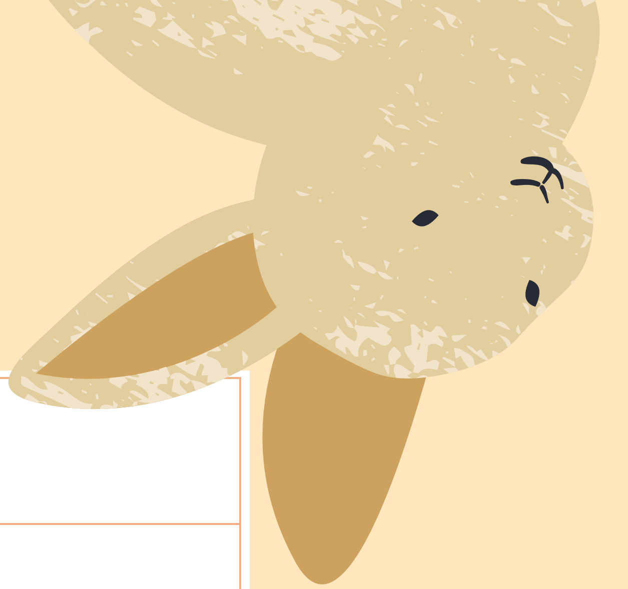
Normalization

- Pet Owner (Owner_ID, name, Phone_Number)
  In 1NF,2NF and 3NF
- Pet (Pet_ID, Pet_Name, Pet_Type, Pet_Age, Pet_Gender, Owner_ID, Assistant_ID)
  In 1NF,2NF and 3NF
- Pet Service (Pet_ID, Service_Code)
  In 1NF,2NF and 3NF
- Service (Service_Code, Service_Name, Payment_Method, Service_Price, Assistant_ID)
  In 1NF,2NF and 3NF
- Disclaimer: Service_Name and Service_Price do not have transitive dependency because the price differs from one pet to another
- Pet Assistant (Assistant_ID, Assistant_Name, Profession, Phone_Number)
  In 1NF,2NF and 3NF

All Relations are in the 3NF as you can see:

- there is no repeating group
- all of them are full dependency
- no transitive

Normalized Tables



Pet Owner

Column name	Data type Size	Constraint
Owner_ID	Number(7)	Primary key
Name	Varchar(12)	Not null
Phone_number	Number(10)	Not null

Assistant

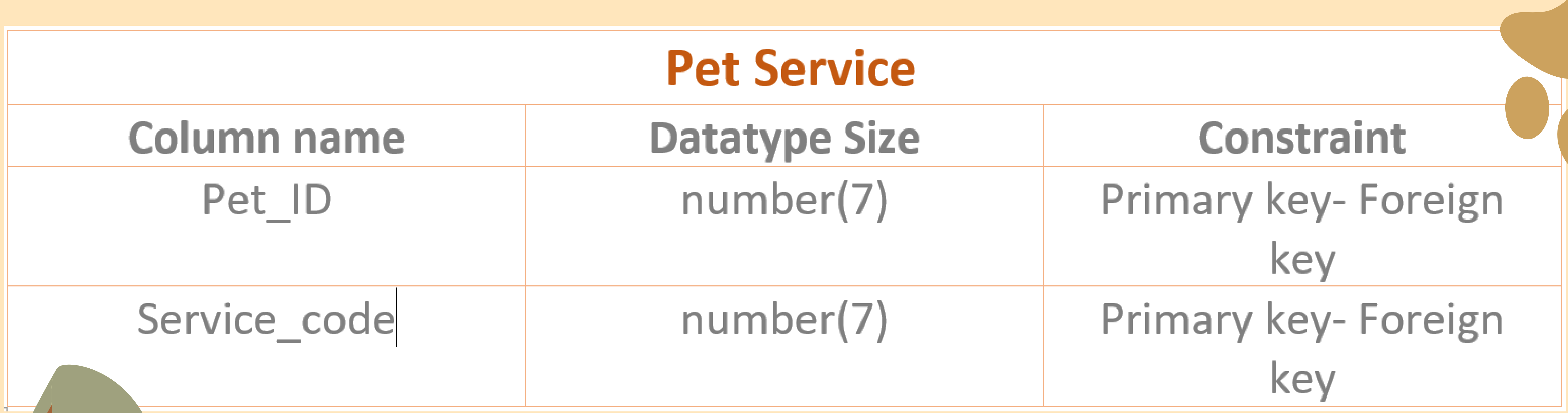
Column name	Datatype Size	Constraint
Assistant_ID	Number(7)	primary key
Assistant_name	Varchar2(12)	Not null
Profession	varchar2(25)	Not null
Phone_number	number(10)	Not null

Pet

Column name	Datatype Size	Constraint
Pet_ID	Number(7)	primary key
Pet_Name	Varchar2(12)	Not null
Pet_Type	Varchar2(25)	Not null
Pet_Age	Varchar2(15)	Not null
Pet_Gender	Varchar2(12)	Not null
Owner_ID	Number(7)	Foreign key
Assistant_ID	Number(7)	Foreign key

Service

Column name	Datatype Size	Constraint
Service_code	Number(7)	Primary key
Service_name	Varchar2(25)	Not null
Payment_method	Varchar2(14)	Not null
Service_price	Number(3)	Not null
Assistant_ID	Number(7)	Foreign key



Pet Service

Column name	Datatype Size	Constraint
Pet_ID	number(7)	Primary key- Foreign key
Service_code	number(7)	Primary key- Foreign key

Functional dependencies

- Owner_ID ---------> Name, Phone_Number

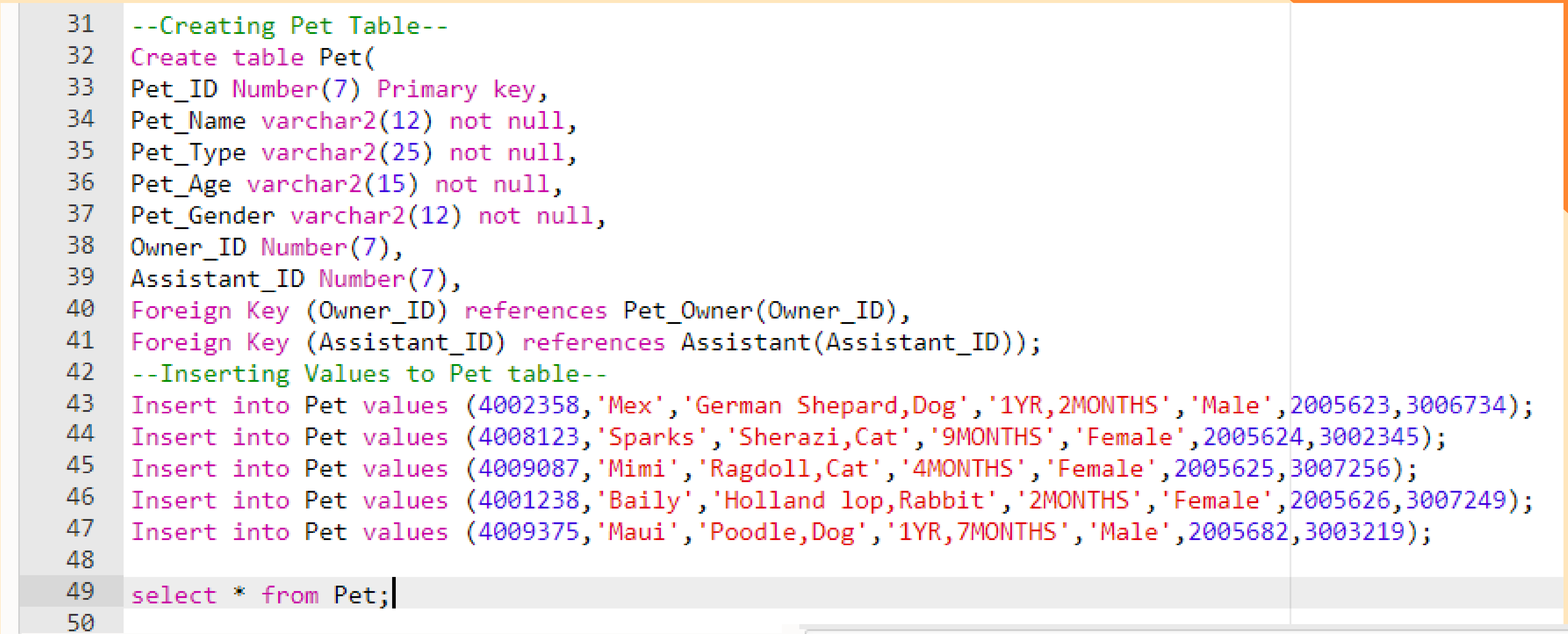
- Pet _ID ---------> Pet_Name, Pet_Type, Pet_Age, Pet_gender, Owner_ID (FK), Assistant_ID (FK)

- Assistant _ID ---------> Assistant_name, Profession, Phone_Number

- Service_Code ---------> Service_Name, Payment_Method, Service_Price, Assistant ID (FK)

- Pet Service ---------> (Pet ID (FK), Service Code (FK)) (PK)

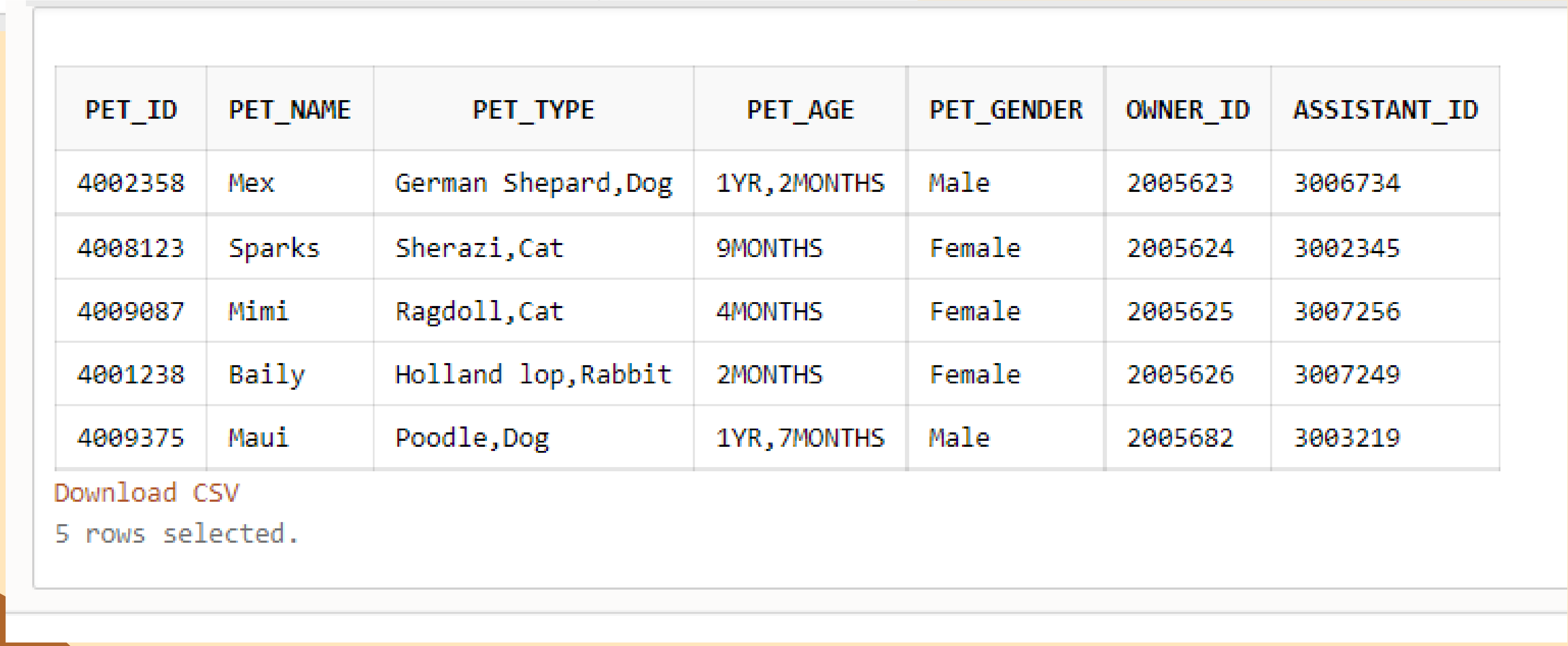
The coding Phase



```
--Creating Pet Table--
Create table Pet(
Pet_ID Number(7) Primary key,
Pet_Name varchar2(12) not null,
Pet_Type varchar2(25) not null,
Pet_Age varchar2(15) not null,
Pet_Gender varchar2(12) not null,
Owner_ID Number(7),
Assistant_ID Number(7),
Foreign Key (Owner_ID) references Pet_Owner(Owner_ID),
Foreign Key (Assistant_ID) references Assistant(Assistant_ID));
--Inserting Values to Pet table--
Insert into Pet values (4002358,'Mex','German Shepard,Dog','1YR,2MONTHS','Male',2005623,3006734);
Insert into Pet values (4008123,'Sparks','Sherazi,Cat','9MONTHS','Female',2005624,3002345);
Insert into Pet values (4009087,'Mimi','Ragdoll,Cat','4MONTHS','Female',2005625,3007256);
Insert into Pet values (4001238,'Baily','Holland lop,Rabbit','2MONTHS','Female',2005626,3007249);
Insert into Pet values (4009375,'Maui','Poodle,Dog','1YR,7MONTHS','Male',2005682,3003219);

select * from Pet;
```

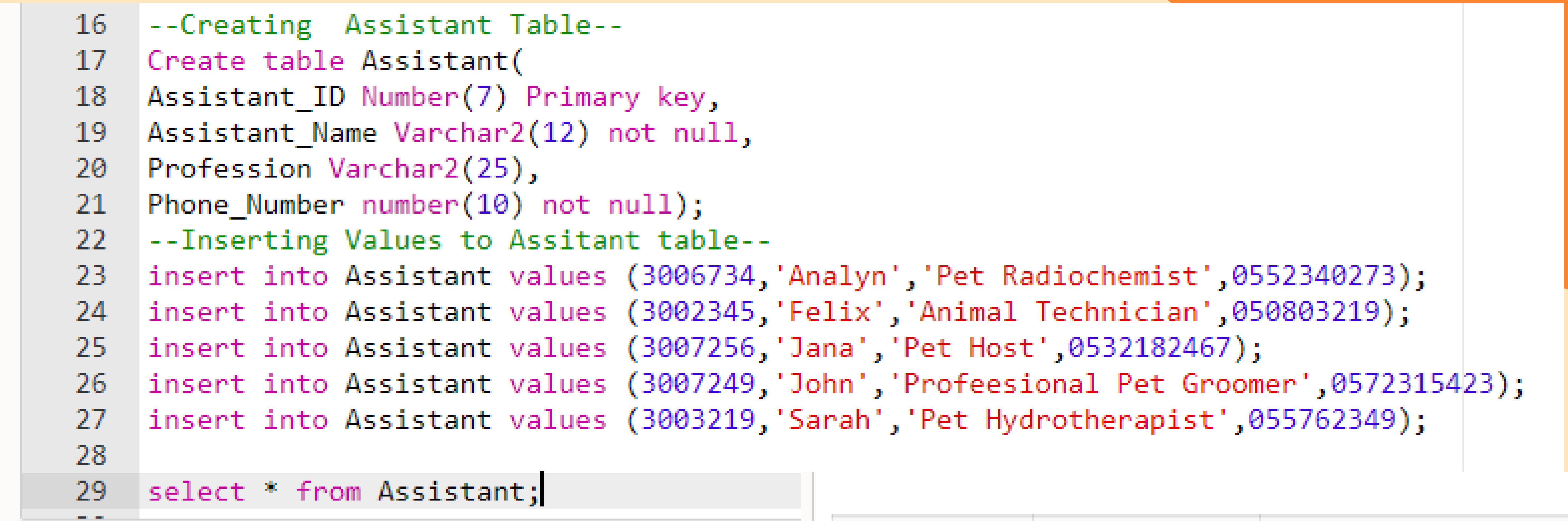
Creating The Pet Table and inserting the data



PET_ID	PET_NAME	PET_TYPE	PET_AGE	PET_GENDER	OWNER_ID	ASSISTANT_ID
4002358	Mex	German Shepard,Dog	1YR,2MONTHS	Male	2005623	3006734
4008123	Sparks	Sherazi,Cat	9MONTHS	Female	2005624	3002345
4009087	Mimi	Ragdoll,Cat	4MONTHS	Female	2005625	3007256
4001238	Baily	Holland lop,Rabbit	2MONTHS	Female	2005626	3007249
4009375	Maui	Poodle,Dog	1YR,7MONTHS	Male	2005682	3003219

Download CSV

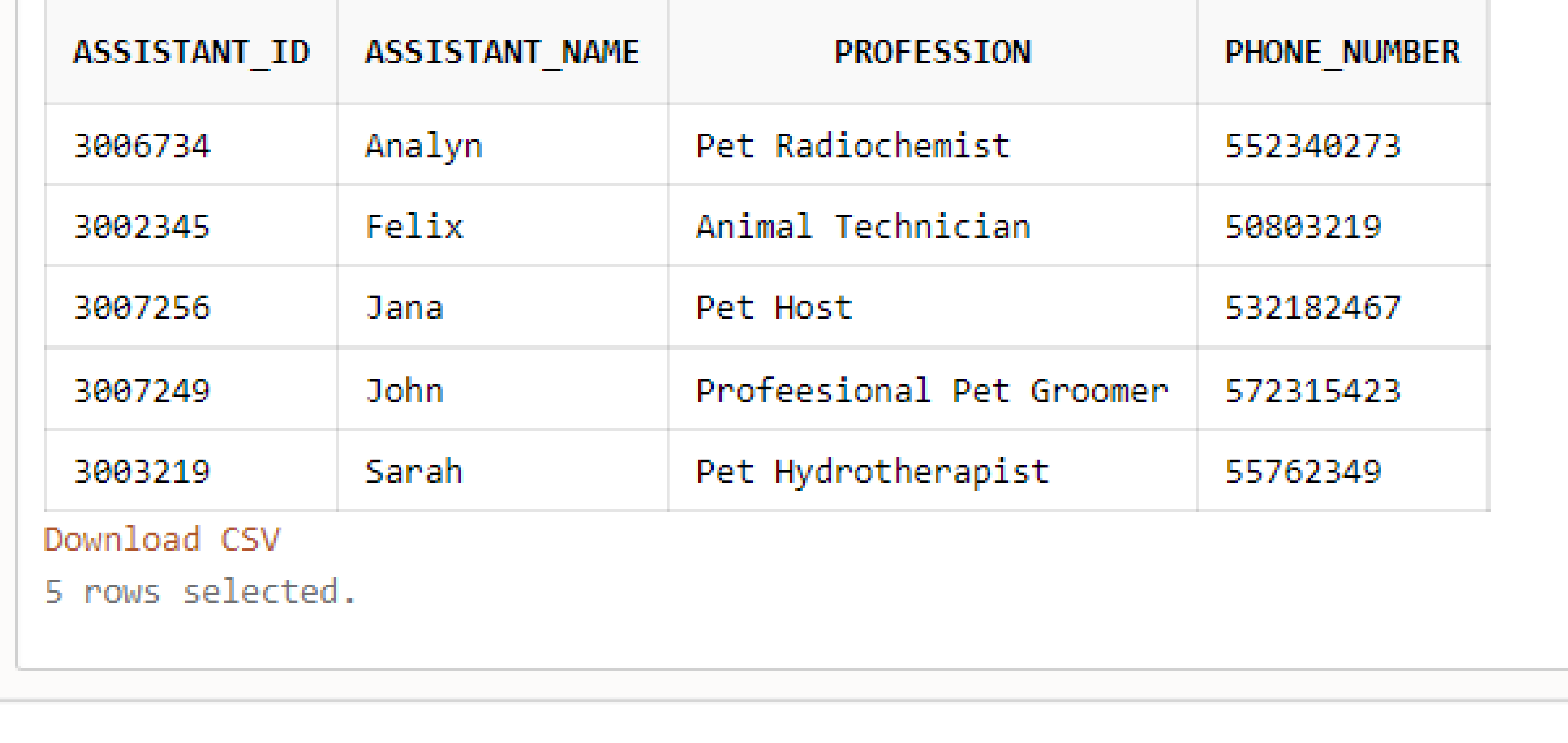
5 rows selected.



```
--Creating  Assistant Table--
Create table Assistant(
Assistant_ID Number(7) Primary key,
Assistant_Name Varchar2(12) not null,
Profession Varchar2(25),
Phone_Number number(10) not null);
--Inserting Values to Assitant table--
insert into Assistant values (3006734,'Analyn','Pet Radiochemist',0552340273);
insert into Assistant values (3002345,'Felix','Animal Technician',050803219);
insert into Assistant values (3007256,'Jana','Pet Host',0532182467);
insert into Assistant values (3007249,'John','Profeesional Pet Groomer',0572315423);
insert into Assistant values (3003219,'Sarah','Pet Hydrotherapist',055762349);

select * from Assistant;
```

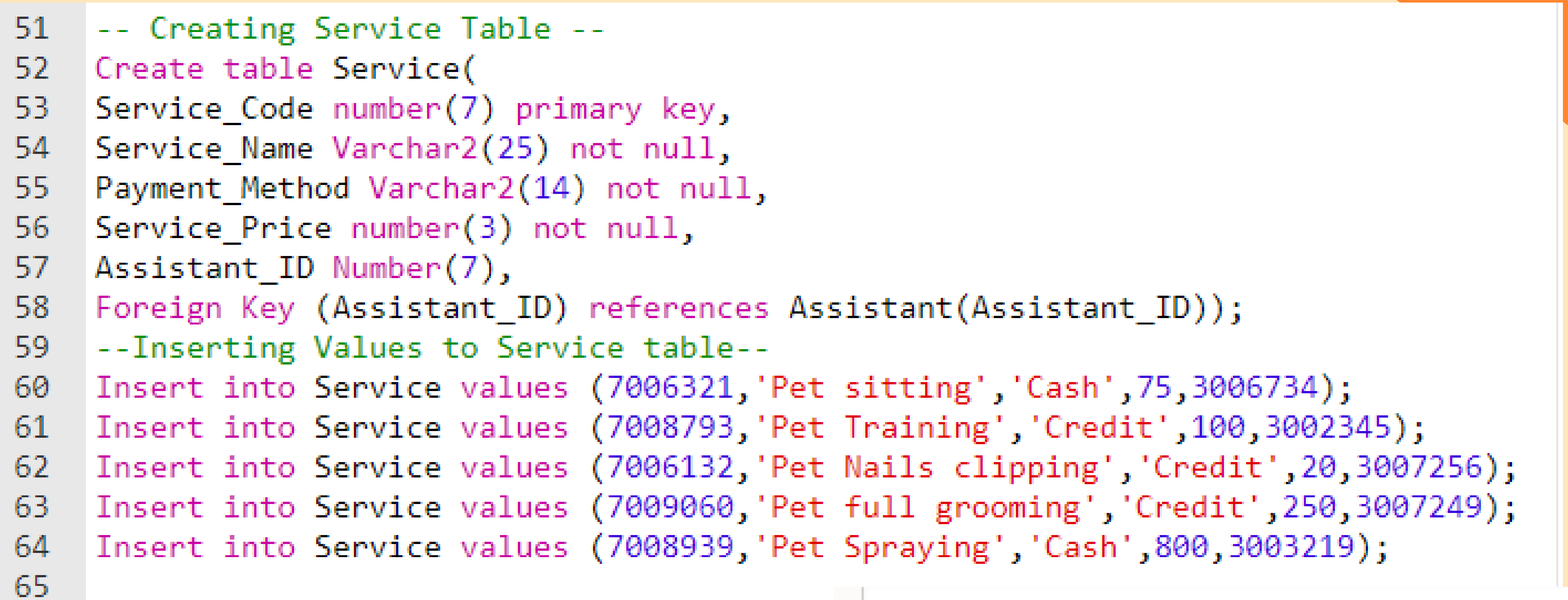
Creating The assistant Table and inserting the data



ASSISTANT_ID	ASSISTANT_NAME	PROFESSION	PHONE_NUMBER
3006734	Analyn	Pet Radiochemist	552340273
3002345	Felix	Animal Technician	50803219
3007256	Jana	Pet Host	532182467
3007249	John	Profeesional Pet Groomer	572315423
3003219	Sarah	Pet Hydrotherapist	55762349

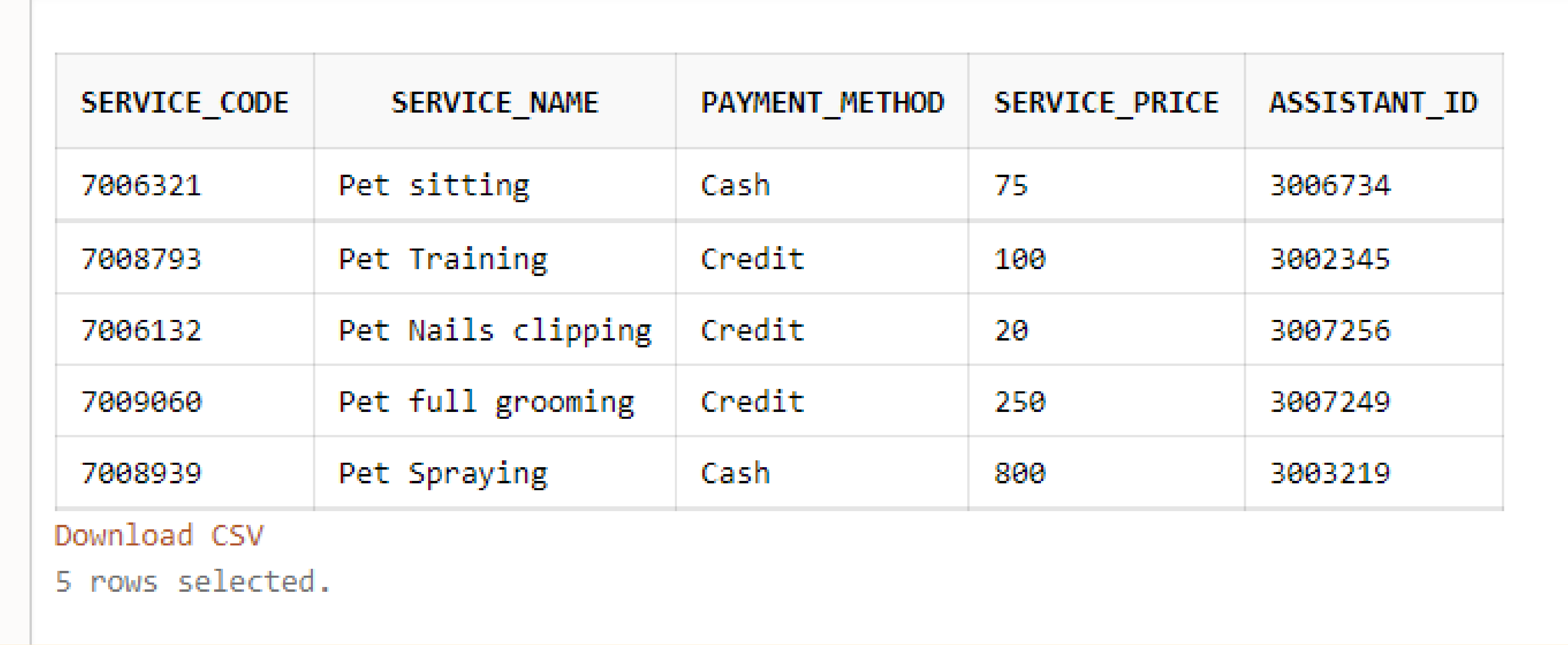
Download CSV
5 rows selected.

Creating The Service Table and inserting the data



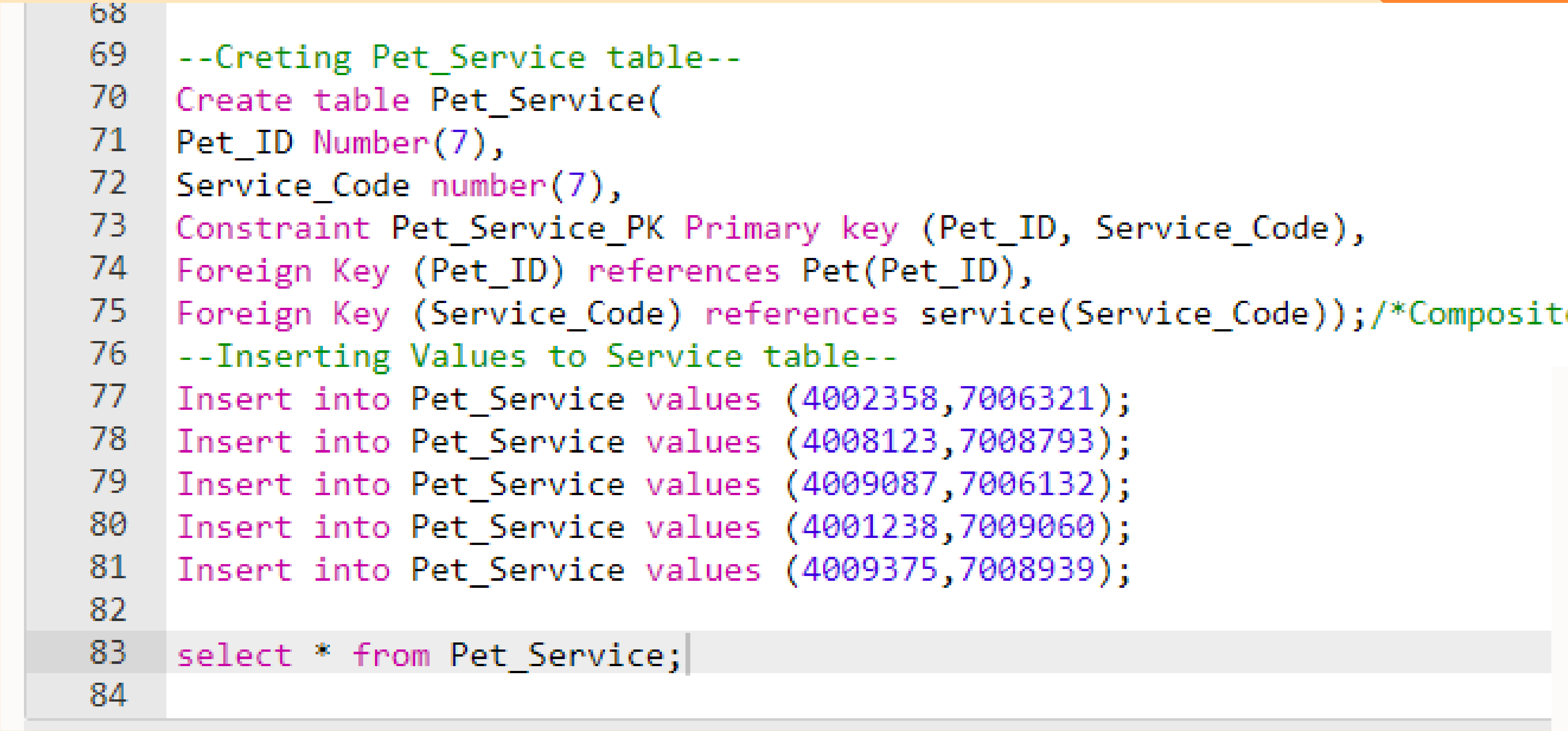
```
-- Creating Service Table --
Create table Service(
Service_Code number(7) primary key,
Service_Name Varchar2(25) not null,
Payment_Method Varchar2(14) not null,
Service_Price number(3) not null,
Assistant_ID Number(7),
Foreign Key (Assistant_ID) references Assistant(Assistant_ID));
--Inserting Values to Service table--
Insert into Service values (7006321,'Pet sitting','Cash',75,3006734);
Insert into Service values (7008793,'Pet Training','Credit',100,3002345);
Insert into Service values (7006132,'Pet Nails clipping','Credit',20,3007256);
Insert into Service values (7009060,'Pet full grooming','Credit',250,3007249);
Insert into Service values (7008939,'Pet Spraying','Cash',800,3003219);

select * from Service;
```



SERVICE_CODE	SERVICE_NAME	PAYMENT_METHOD	SERVICE_PRICE	ASSISTANT_ID
7006321	Pet sitting	Cash	75	3006734
7008793	Pet Training	Credit	100	3002345
7006132	Pet Nails clipping	Credit	20	3007256
7009060	Pet full grooming	Credit	250	3007249
7008939	Pet Spraying	Cash	800	3003219

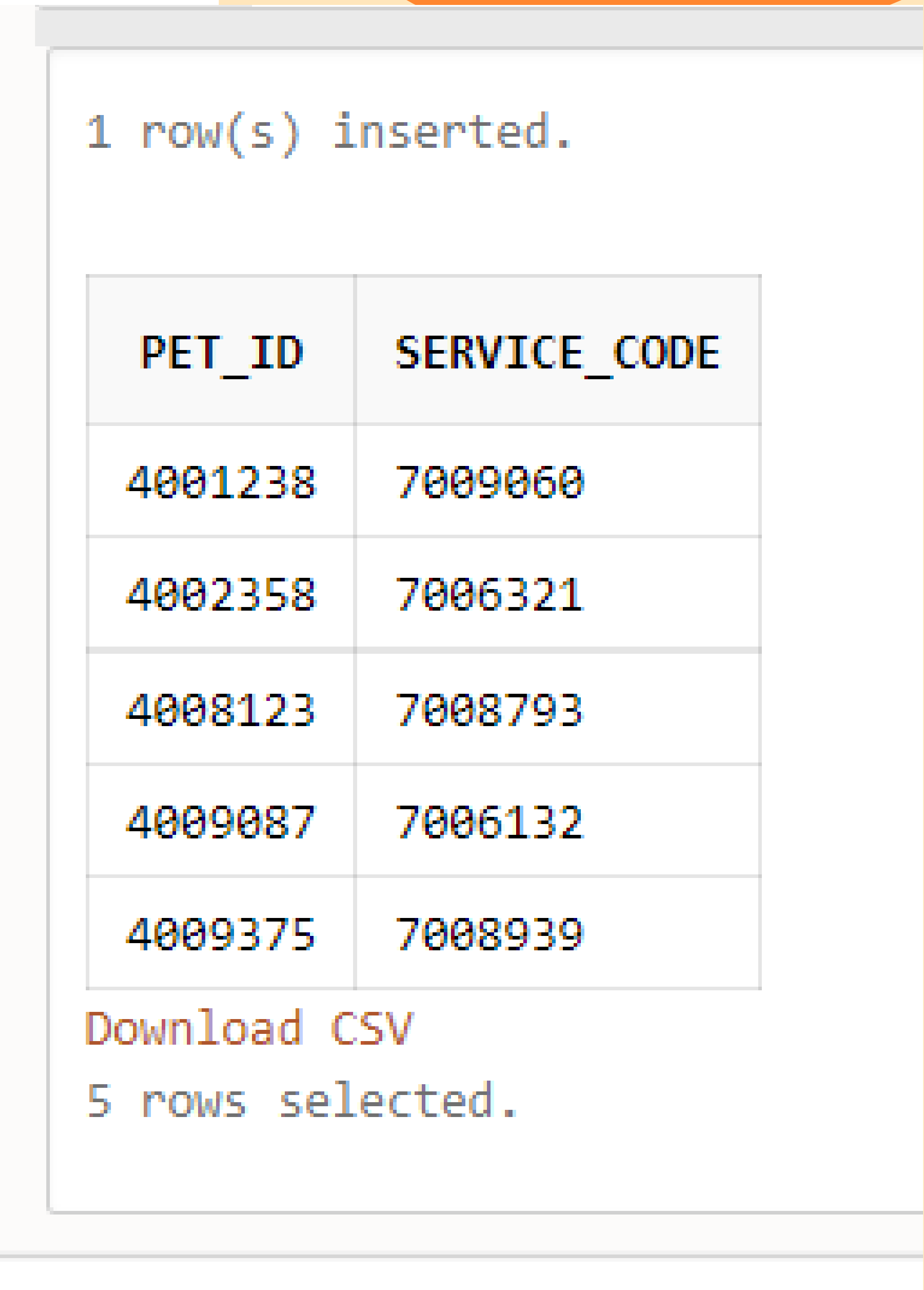
5 rows selected.



```
--Creting Pet_Service table--
Create table Pet_Service(
Pet_ID Number(7),
Service_Code number(7),
Constraint Pet_Service_PK Primary key (Pet_ID, Service_Code),
Foreign Key (Pet_ID) references Pet(Pet_ID),
Foreign Key (Service_Code) references service(Service_Code));/*Composite PK*/
--Inserting Values to Service table--
Insert into Pet_Service values (4002358,7006321);
Insert into Pet_Service values (4008123,7008793);
Insert into Pet_Service values (4009087,7006132);
Insert into Pet_Service values (4001238,7009060);
Insert into Pet_Service values (4009375,7008939);

select * from Pet_Service;

```

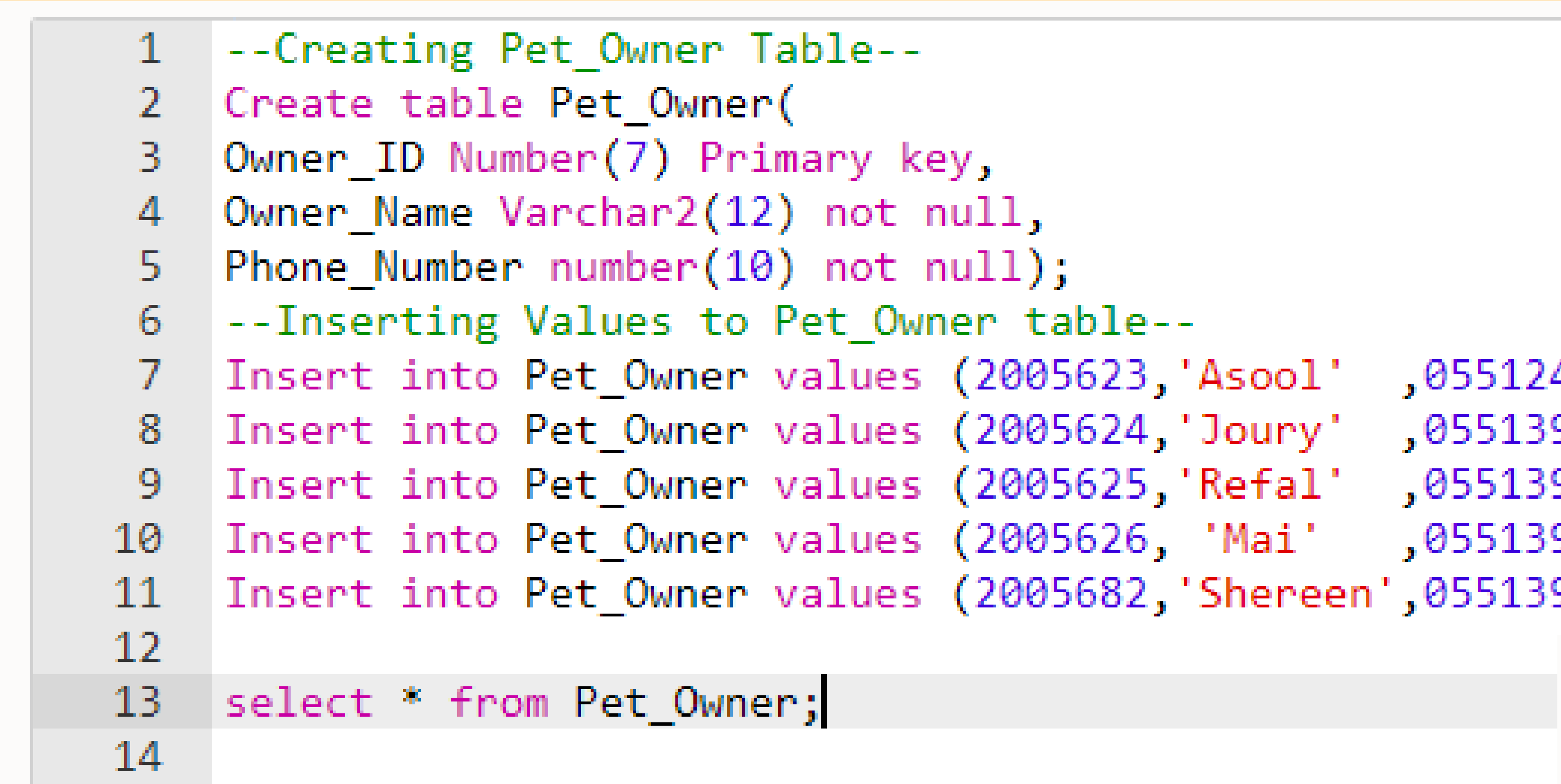


```
1 row(s) inserted.
```

PET_ID	SERVICE_CODE
4001238	7009060
4002358	7006321
4008123	7008793
4009087	7006132
4009375	7008939

```
5 rows selected.
```

Creating The Pet_Service Table and inserting the data

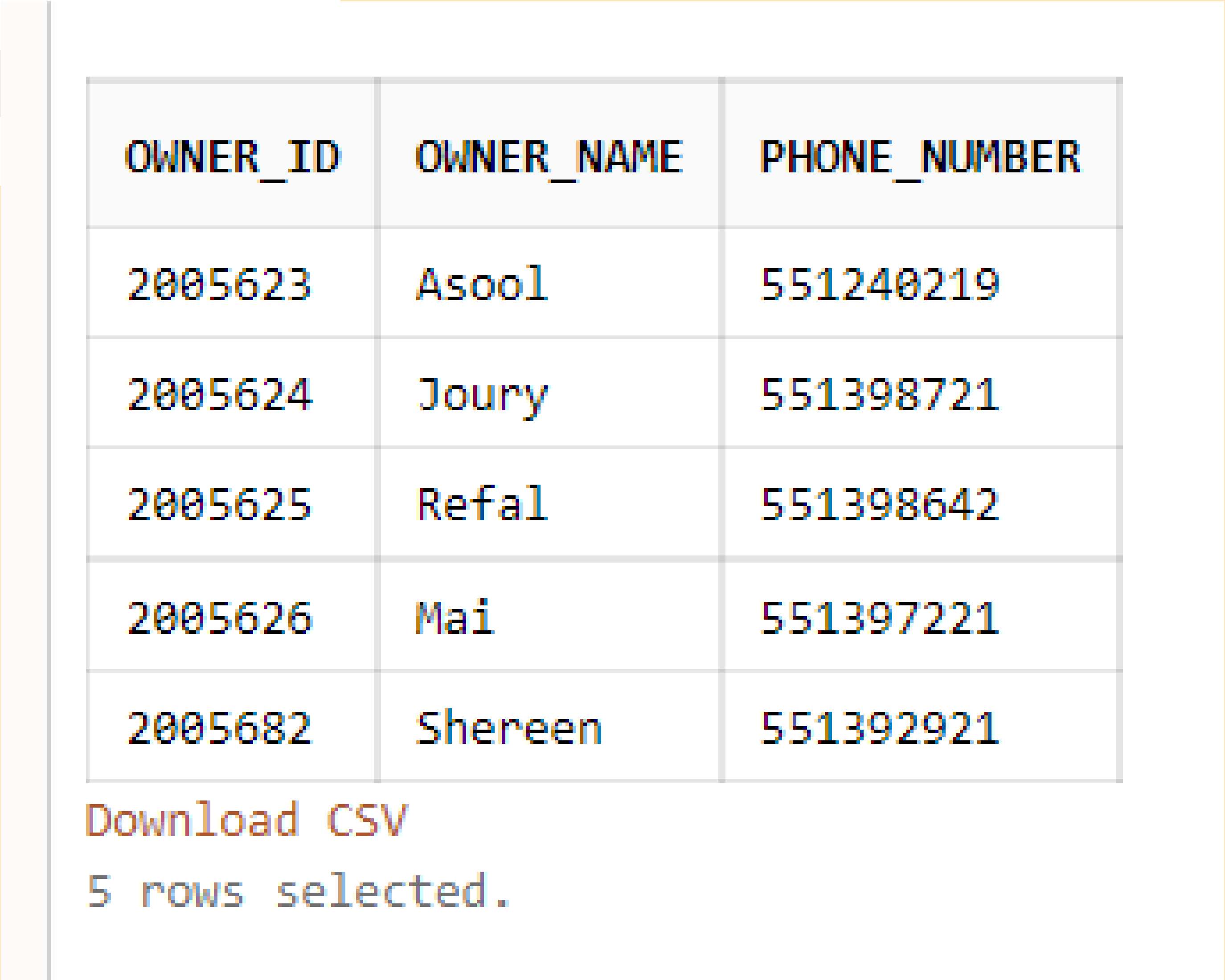


```
--Creating Pet_Owner Table--
Create table Pet_Owner(
Owner_ID Number(7) Primary key,
Owner_Name Varchar2(12) not null,
Phone_Number number(10) not null);
--Inserting Values to Pet_Owner table--
Insert into Pet_Owner values (2005623,'Asool'  ,0551240219);
Insert into Pet_Owner values (2005624,'Joury'  ,0551398721);
Insert into Pet_Owner values (2005625,'Refal'  ,0551398642);
Insert into Pet_Owner values (2005626, 'Mai'   ,0551397221);
Insert into Pet_Owner values (2005682,'Shereen',0551392921);

select * from Pet_Owner;

```

Creting The Pet_Owner Table and inserting the data

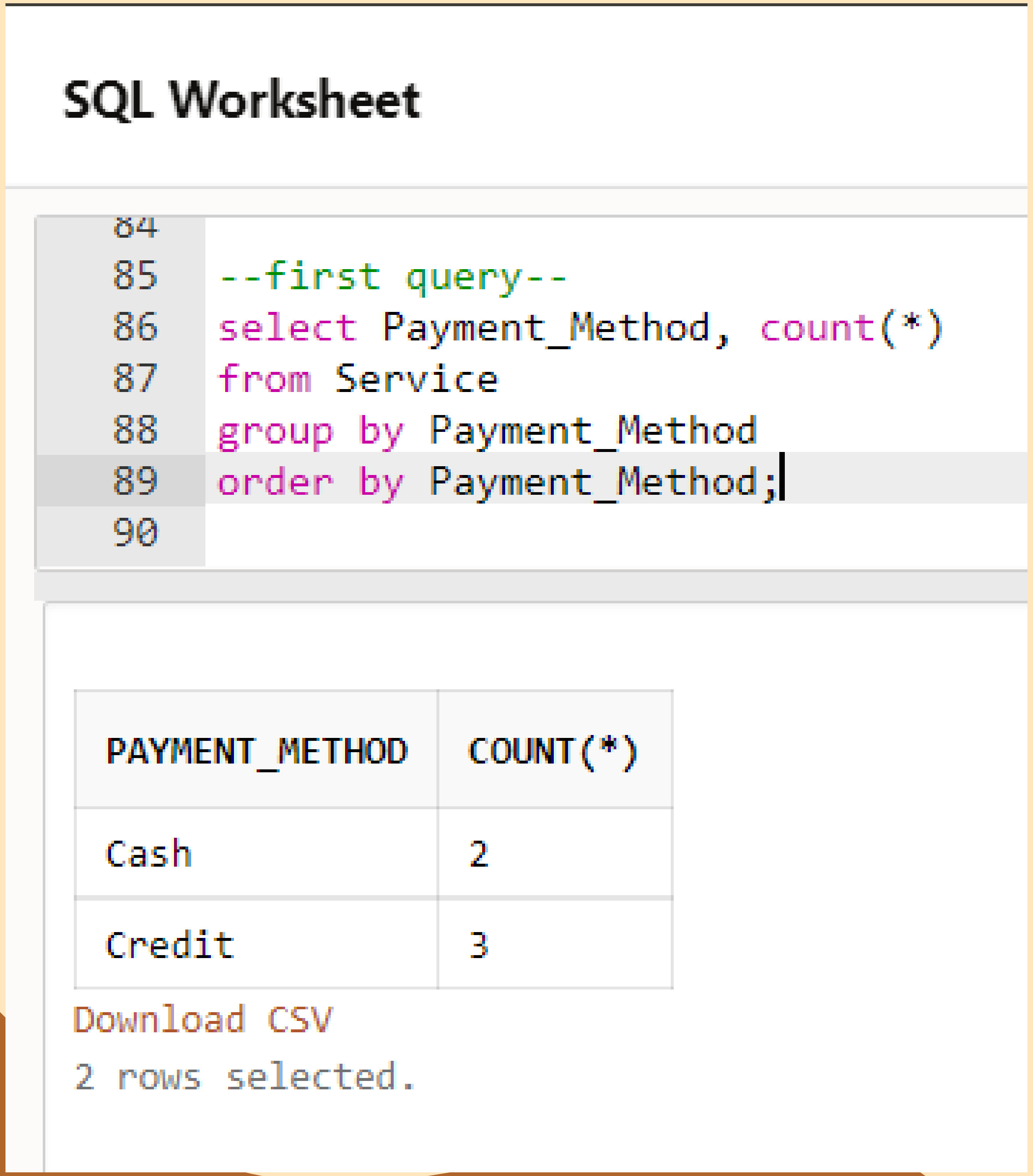


OWNER_ID	OWNER_NAME	PHONE_NUMBER
2005623	Asool	551240219
2005624	Joury	551398721
2005625	Refal	551398642
2005626	Mai	551397221
2005682	Shereen	551392921

Download CSV

5 rows selected.

Queries



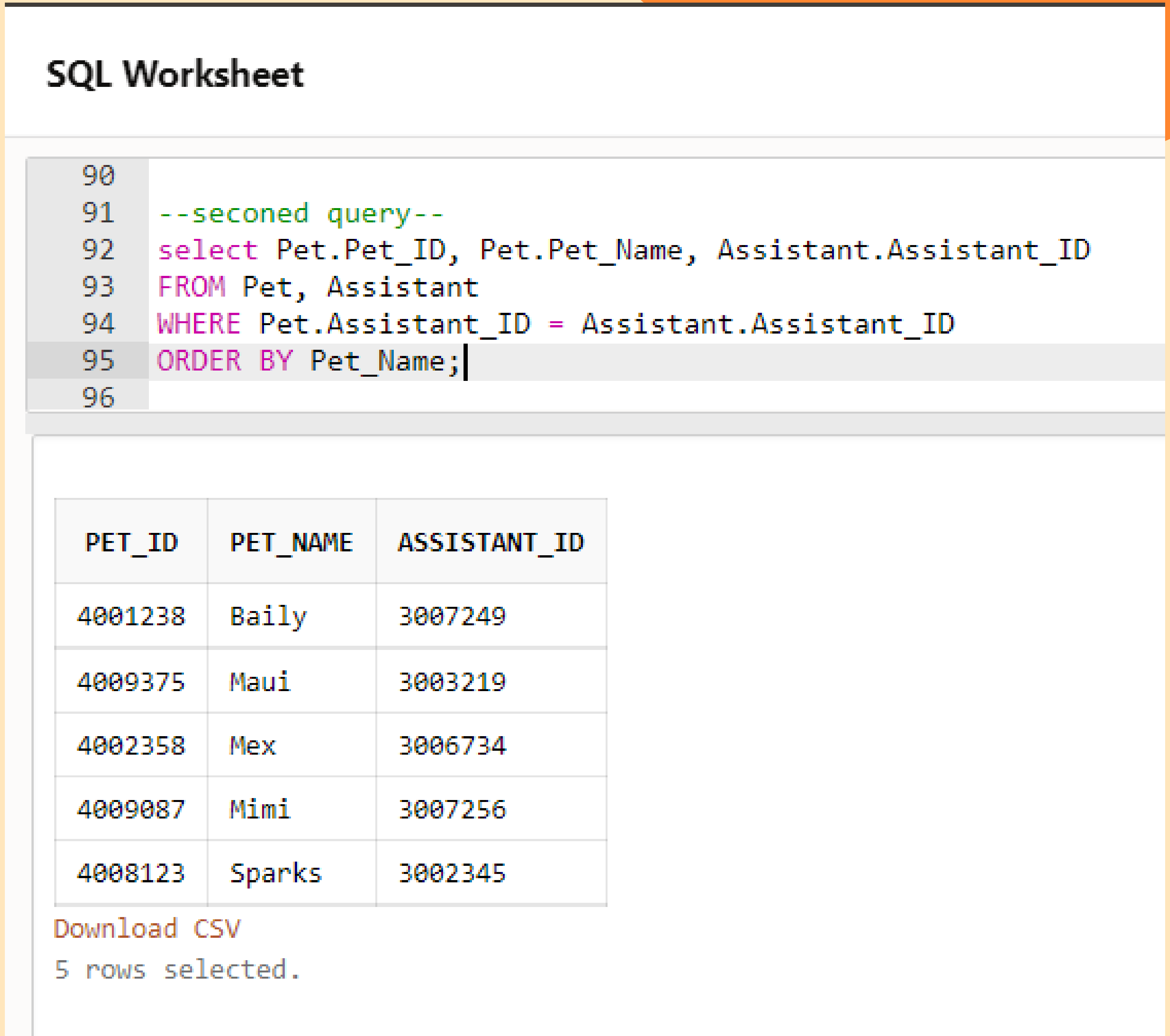
SQL Worksheet

```
--first query--
select Payment_Method, count(*)
from Service
group by Payment_Method
order by Payment_Method;
```

PAYMENT_METHOD	COUNT(*)
Cash	2
Credit	3

Download CSV

2 rows selected.



SQL Worksheet

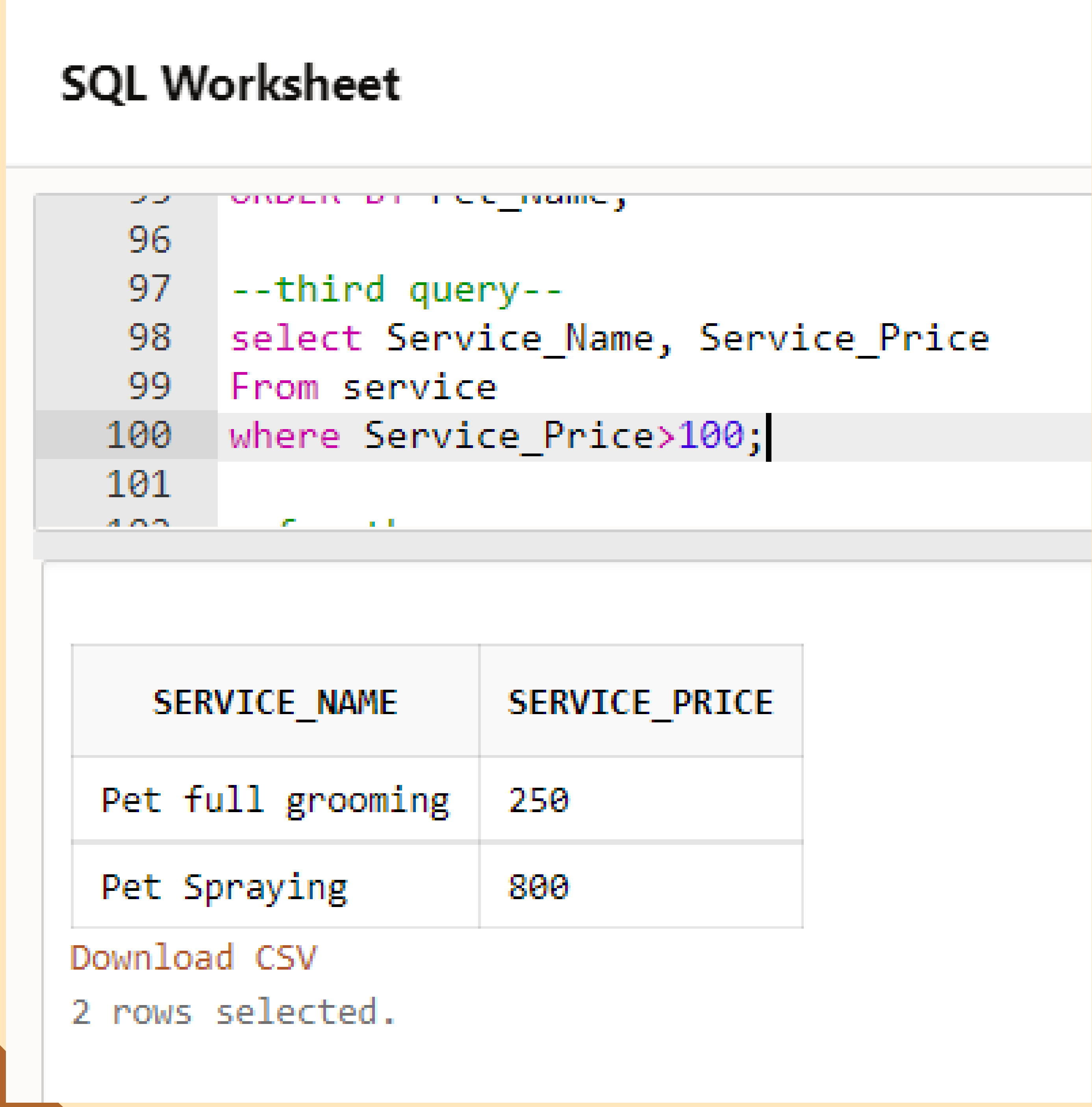
```
--seconed query--
select Pet.Pet_ID, Pet.Pet_Name, Assistant.Assistant_ID
FROM Pet, Assistant
WHERE Pet.Assistant_ID = Assistant.Assistant_ID
ORDER BY Pet_Name;
```

PET_ID	PET_NAME	ASSISTANT_ID
4001238	Baily	3007249
4009375	Maui	3003219
4002358	Mex	3006734
4009087	Mimi	3007256
4008123	Sparks	3002345

Download CSV

5 rows selected.

Queries



SQL Worksheet

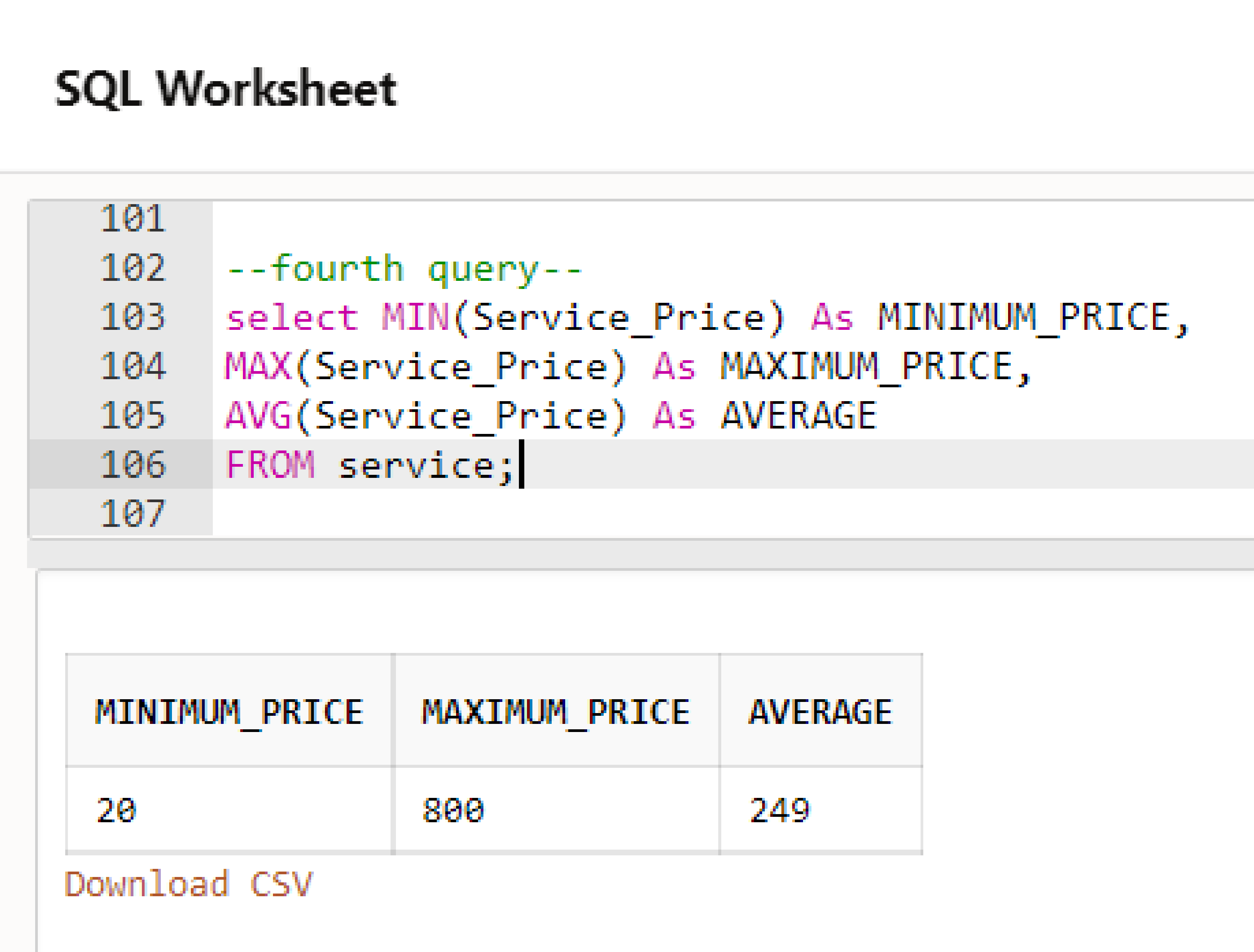
```
ORDER BY Pet_Name;

--third query--
select Service_Name, Service_Price
From service
where Service_Price>100;

```

SERVICE_NAME	SERVICE_PRICE
Pet full grooming	250
Pet Spraying	800

2 rows selected.



SQL Worksheet

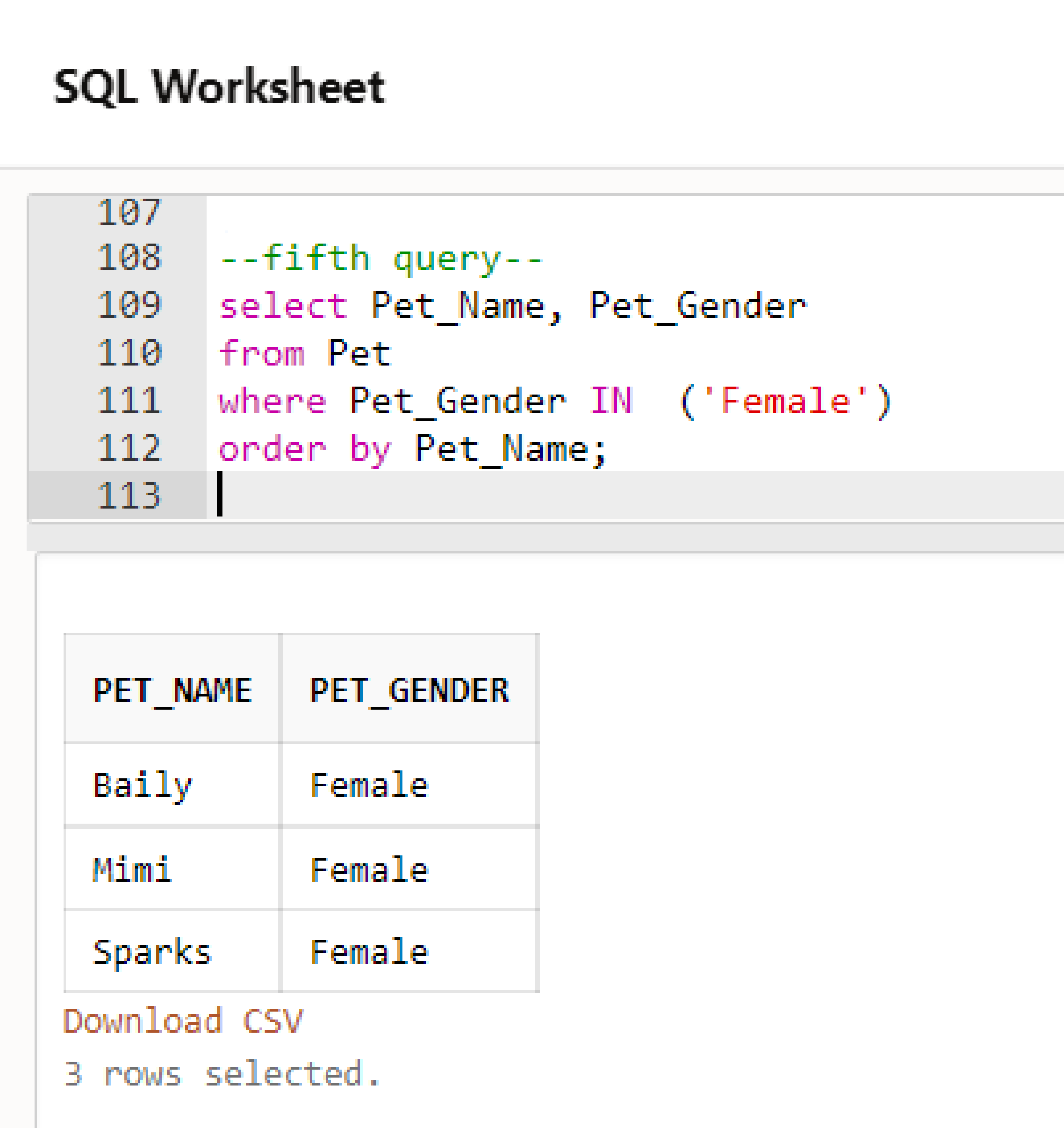
```

--fourth query--
select MIN(Service_Price) As MINIMUM_PRICE,
MAX(Service_Price) As MAXIMUM_PRICE,
AVG(Service_Price) As AVERAGE
FROM service;

```

MINIMUM_PRICE	MAXIMUM_PRICE	AVERAGE
20	800	249

Queries



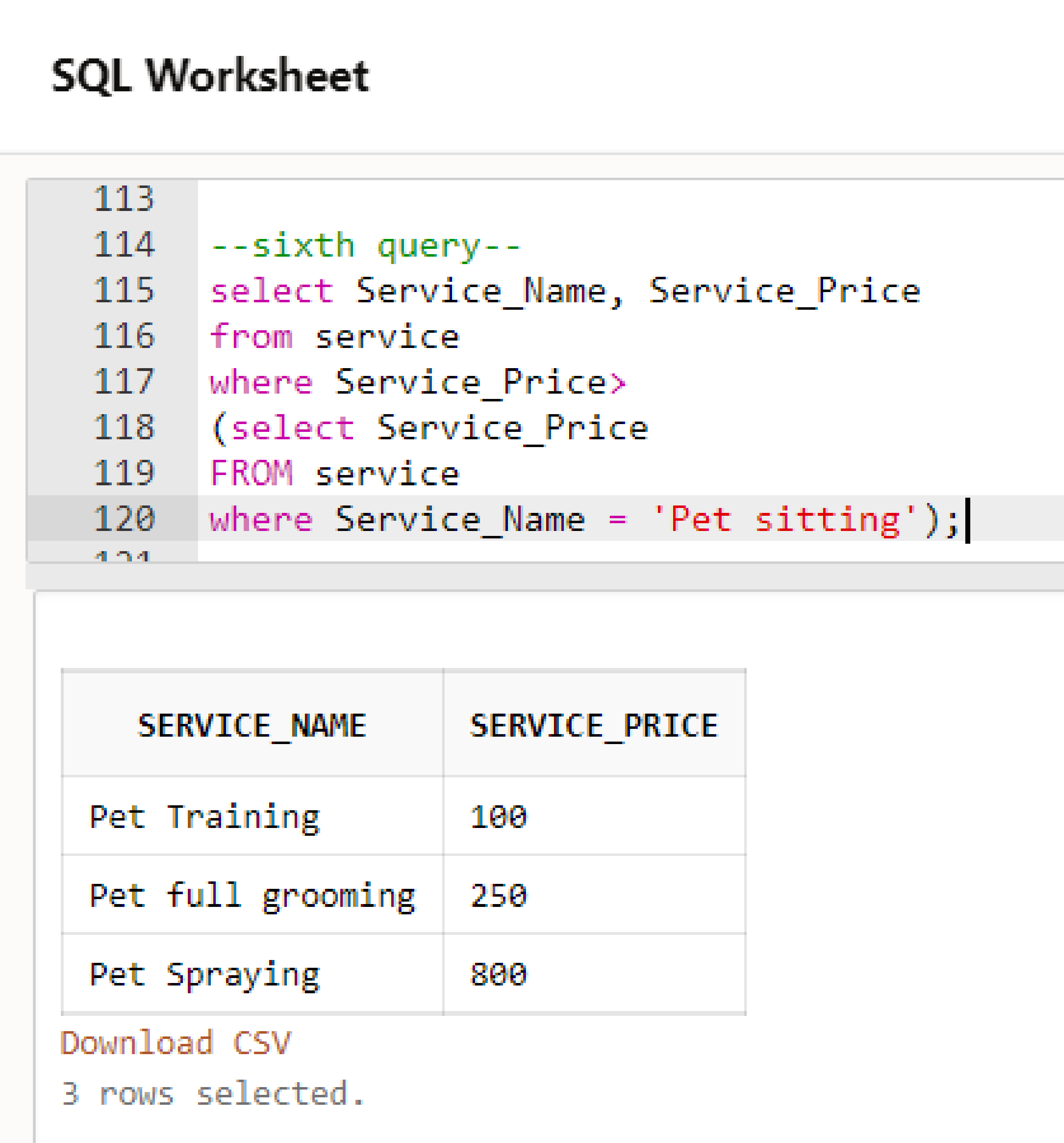
SQL Worksheet

```
--fifth query--
select Pet_Name, Pet_Gender
from Pet
where Pet_Gender IN  ('Female')
order by Pet_Name;
```

PET_NAME	PET_GENDER
Baily	Female
Mimi	Female
Sparks	Female

Download CSV

3 rows selected.



SQL Worksheet

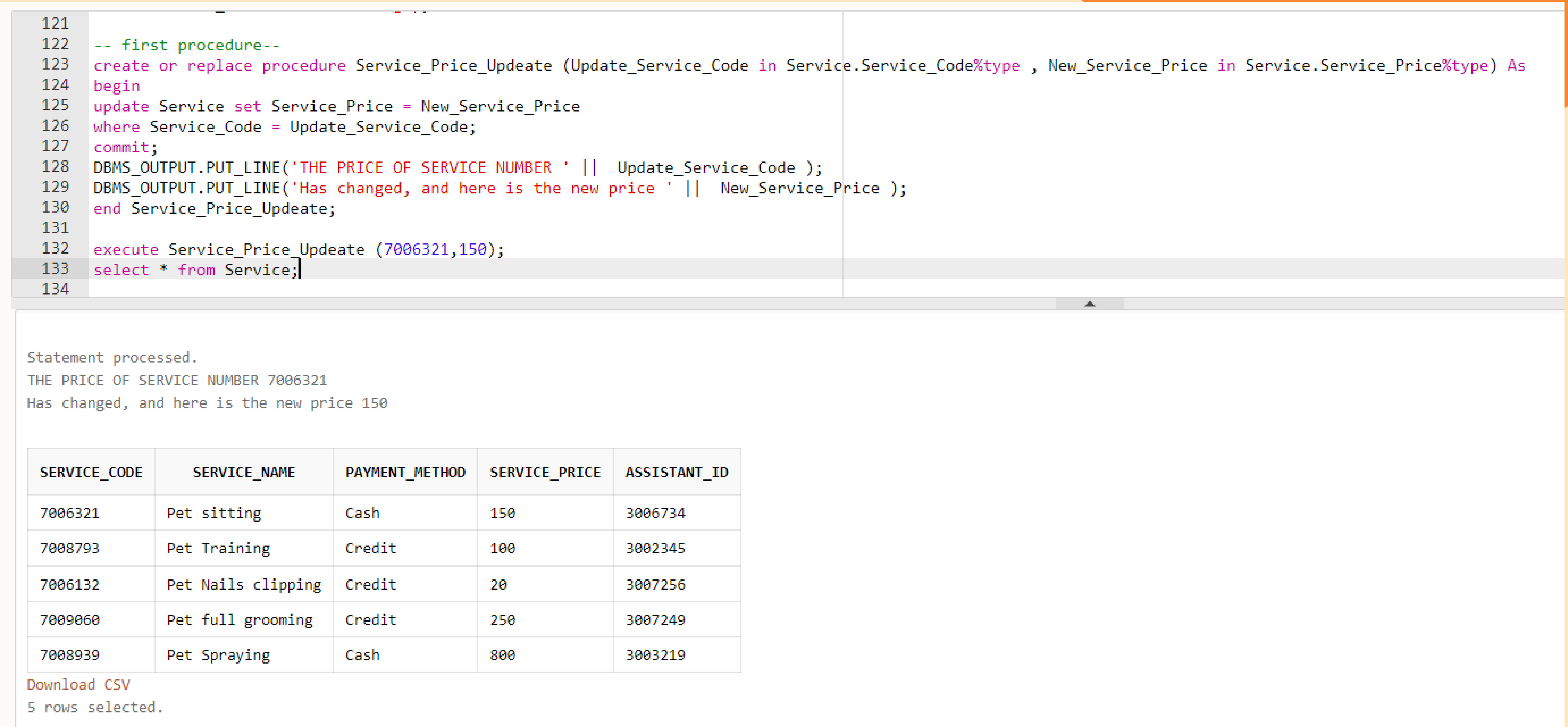
```
--sixth query--
select Service_Name, Service_Price
from service
where Service_Price>
(select Service_Price
FROM service
where Service_Name = 'Pet sitting');
```

SERVICE_NAME	SERVICE_PRICE
Pet Training	100
Pet full grooming	250
Pet Spraying	800

Download CSV

3 rows selected.

Procedures



```
-- first procedure--
create or replace procedure Service_Price_Updeate (Update_Service_Code in Service.Service_Code%type , New_Service_Price in Service.Service_Price%type) As
begin
update Service set Service_Price = New_Service_Price
where Service_Code = Update_Service_Code;
commit;
DBMS_OUTPUT.PUT_LINE('THE PRICE OF SERVICE NUMBER ' ||  Update_Service_Code );
DBMS_OUTPUT.PUT_LINE('Has changed, and here is the new price ' ||  New_Service_Price );
end Service_Price_Updeate;

execute Service_Price_Updeate (7006321,150);
select * from Service;
```

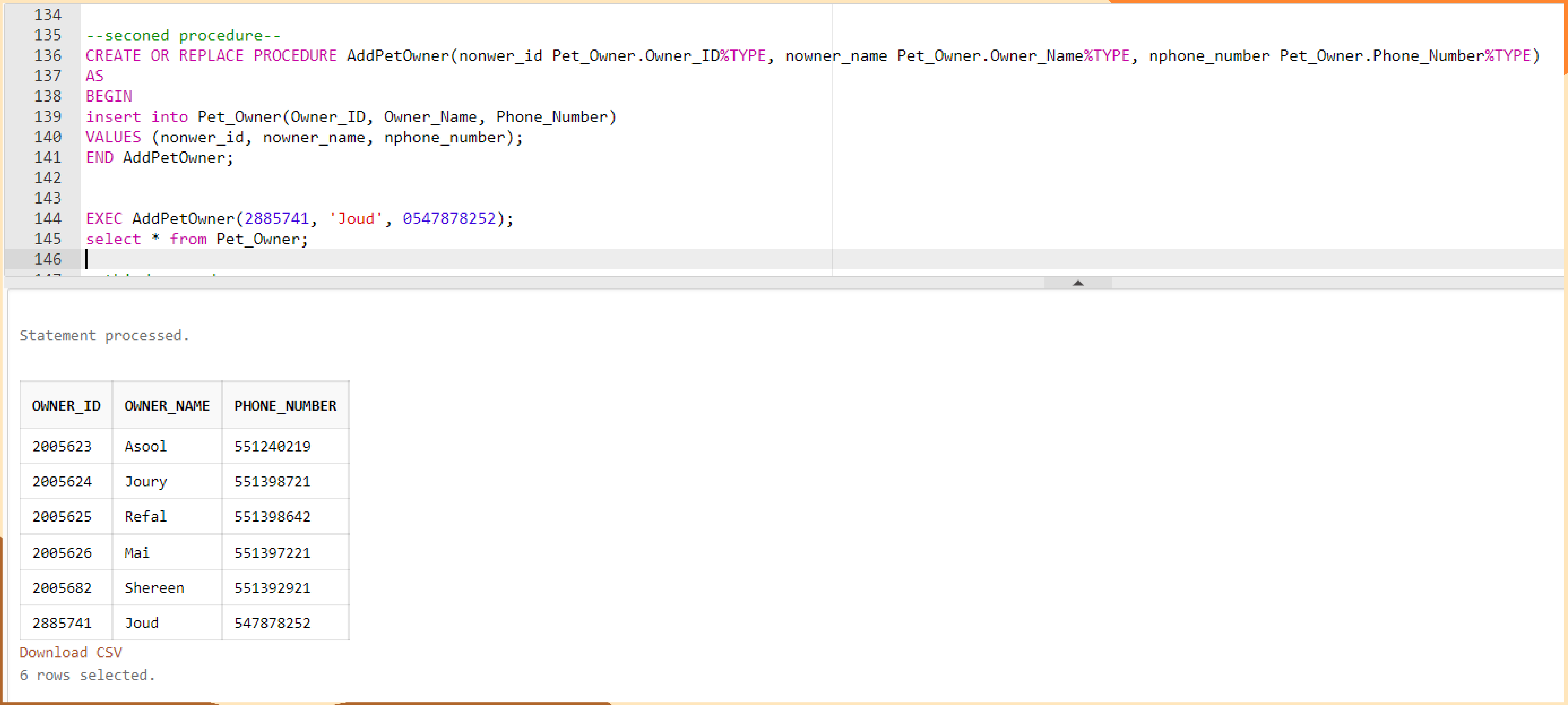
```
Statement processed.
THE PRICE OF SERVICE NUMBER 7006321
Has changed, and here is the new price 150
```

SERVICE_CODE	SERVICE_NAME	PAYMENT_METHOD	SERVICE_PRICE	ASSISTANT_ID
7006321	Pet sitting	Cash	150	3006734
7008793	Pet Training	Credit	100	3002345
7006132	Pet Nails clipping	Credit	20	3007256
7009060	Pet full grooming	Credit	250	3007249
7008939	Pet Spraying	Cash	800	3003219

Download CSV

5 rows selected.

Procedures



```
--seconed procedure--
CREATE OR REPLACE PROCEDURE AddPetOwner(nonwer_id Pet_Owner.Owner_ID%TYPE, nowner_name Pet_Owner.Owner_Name%TYPE, nphone_number Pet_Owner.Phone_Number%TYPE)
AS
BEGIN
insert into Pet_Owner(Owner_ID, Owner_Name, Phone_Number)
VALUES (nonwer_id, nowner_name, nphone_number);
END AddPetOwner;


EXEC AddPetOwner(2885741, 'Joud', 0547878252);
select * from Pet_Owner;
```

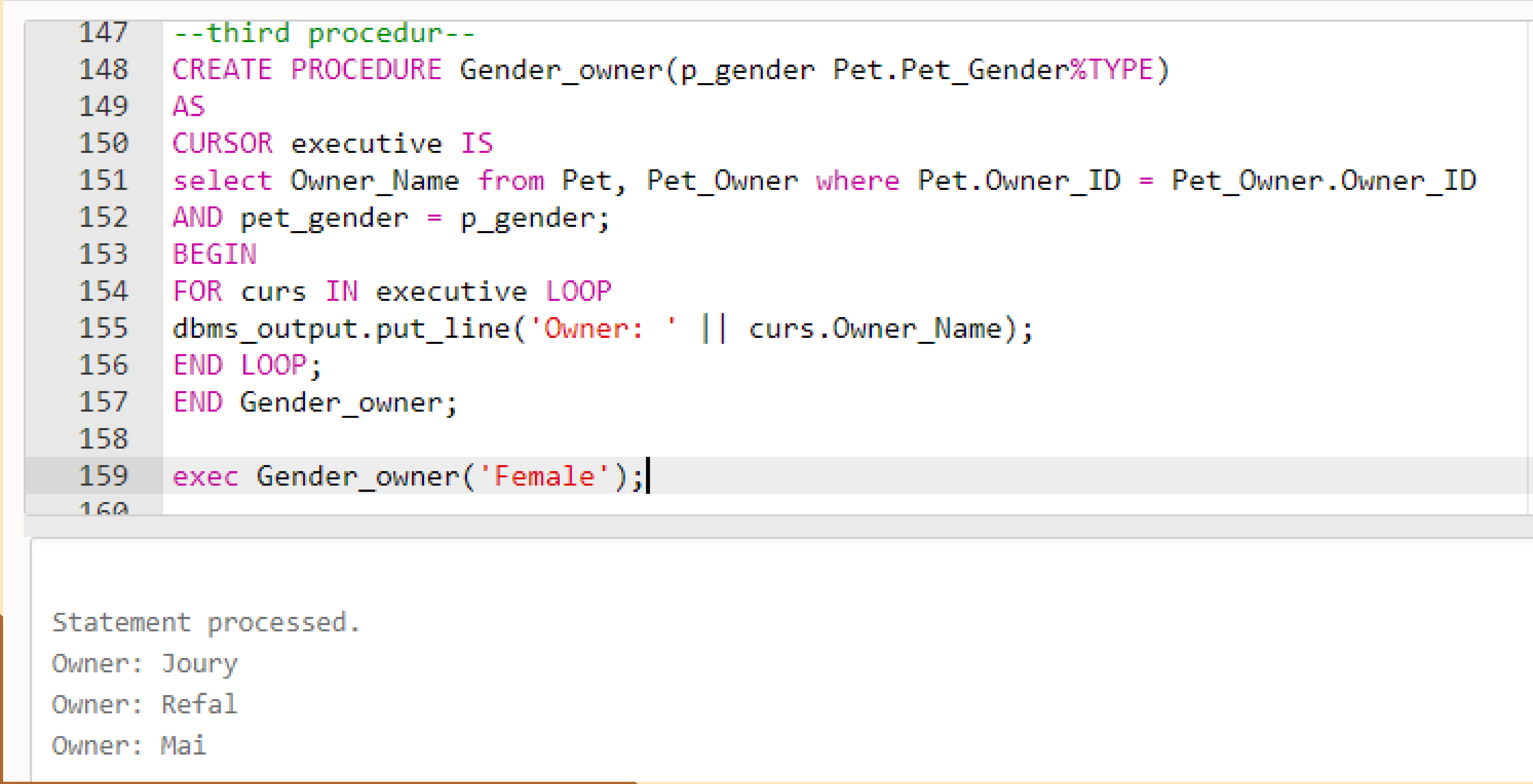
```
Statement processed.
```

OWNER_ID	OWNER_NAME	PHONE_NUMBER
2005623	Asool	551240219
2005624	Joury	551398721
2005625	Refal	551398642
2005626	Mai	551397221
2005682	Shereen	551392921
2885741	Joud	547878252

Download CSV

```
6 rows selected.
```

Procedures



```
--third procedur--
CREATE PROCEDURE Gender_owner(p_gender Pet.Pet_Gender%TYPE)
AS
CURSOR executive IS
select Owner_Name from Pet, Pet_Owner where Pet.Owner_ID = Pet_Owner.Owner_ID
AND pet_gender = p_gender;
BEGIN
FOR curs IN executive LOOP
dbms_output.put_line('Owner: ' || curs.Owner_Name);
END LOOP;
END Gender_owner;

exec Gender_owner('Female');
```

```
Statement processed.
Owner: Joury
Owner: Refal
Owner: Mai
```