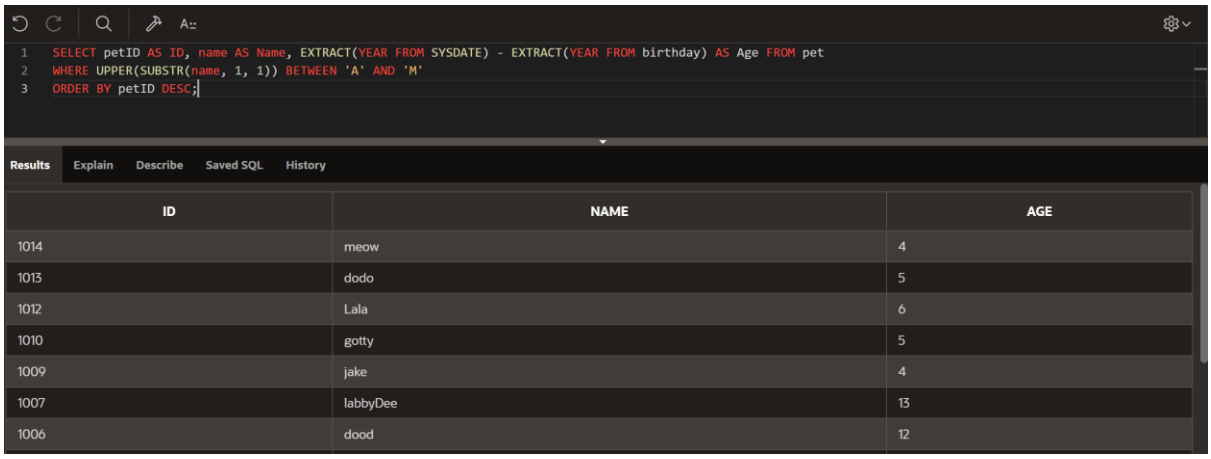
Query Implementation

In this section, we investigate the implementation of SQL queries using ORACLE SQL (VIA APEX) to extract data from the database created in the second part of the project (Database-Design&Implementation). Each query is accompanied by the respective SQL script and a screenshot of the result.

Query 1: Pet Details for Names A-M, Sorted by ID (Descending order)

```
SELECT petID AS ID, name AS Name, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR
FROM birthday) AS Age FROM pet
WHERE UPPER(SUBSTR(name, 1, 1)) BETWEEN 'A' AND 'M'
ORDER BY petID DESC;
```



```
SELECT petID AS ID, name AS Name, EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM birthday) AS Age FROM pet
WHERE UPPER(SUBSTR(name, 1, 1)) BETWEEN 'A' AND 'M'
ORDER BY petID DESC;
```

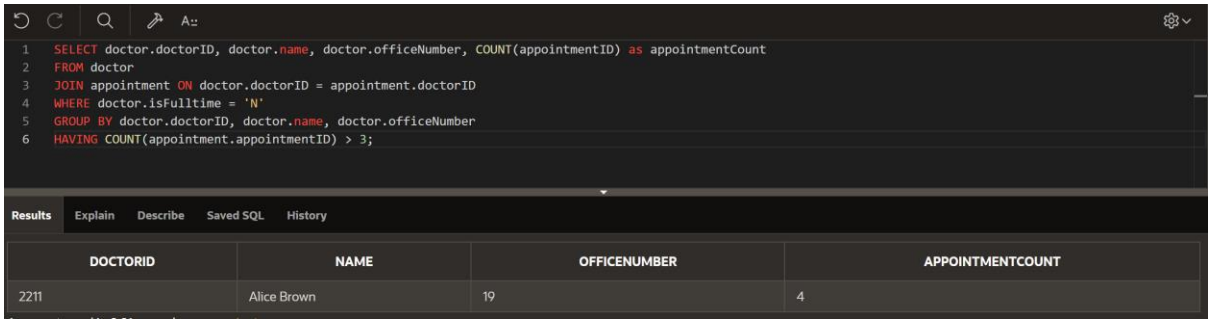
Results | Explain | Describe | Saved SQL | History

ID	NAME	AGE
1014	meow	4
1013	dodo	5
1012	Lala	6
1010	gotty	5
1009	jake	4
1007	labbyDee	13
1006	dood	12

Figure 1. Query 1 output

Query 2: Doctor Details for Most Overworked Part-Time Vet Doctors with 3 or More Appointments.

```
SELECT doctor.doctorID, doctor.name, doctor.officeNumber, COUNT(appointmentID) as
appointmentCount
FROM doctor
JOIN appointment ON doctor.doctorID = appointment.doctorID
WHERE doctor.isFulltime = 'N'
GROUP BY doctor.doctorID, doctor.name, doctor.officeNumber
HAVING COUNT(appointment.appointmentID) > 3;
```



```
SELECT doctor.doctorID, doctor.name, doctor.officeNumber, COUNT(appointmentID) as appointmentCount
FROM doctor
JOIN appointment ON doctor.doctorID = appointment.doctorID
WHERE doctor.isFulltime = 'N'
GROUP BY doctor.doctorID, doctor.name, doctor.officeNumber
HAVING COUNT(appointment.appointmentID) > 3;
```

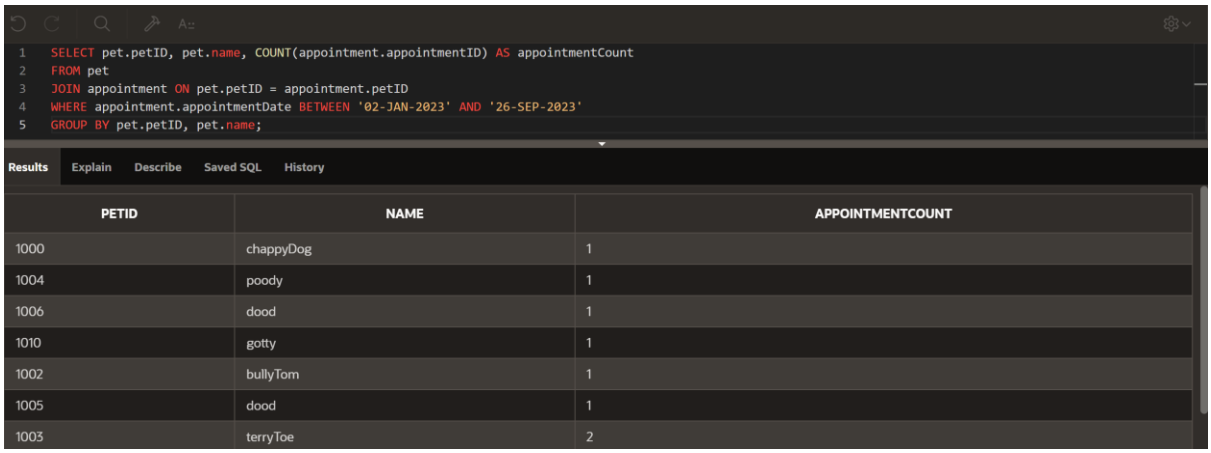
Results | Explain | Describe | Saved SQL | History

DOCTORID	NAME	OFFICENUMBER	APPOINTMENTCOUNT
2211	Alice Brown	19	4

Figure 2. Query 2 output

Query 3: Pet Details for pets with 2 or More Appointments between 2nd Jan 2023 and 26th Sep 2023.

```
SELECT pet.petID, pet.name, COUNT(appointment.appointmentID) AS appointmentCount
FROM pet
JOIN appointment ON pet.petID = appointment.petID
WHERE appointment.appointmentDate BETWEEN '02-JAN-2023' AND '26-SEP-2023'
GROUP BY pet.petID, pet.name;
```



```
SELECT pet.petID, pet.name, COUNT(appointment.appointmentID) AS appointmentCount
FROM pet
JOIN appointment ON pet.petID = appointment.petID
WHERE appointment.appointmentDate BETWEEN '02-JAN-2023' AND '26-SEP-2023'
GROUP BY pet.petID, pet.name;
```

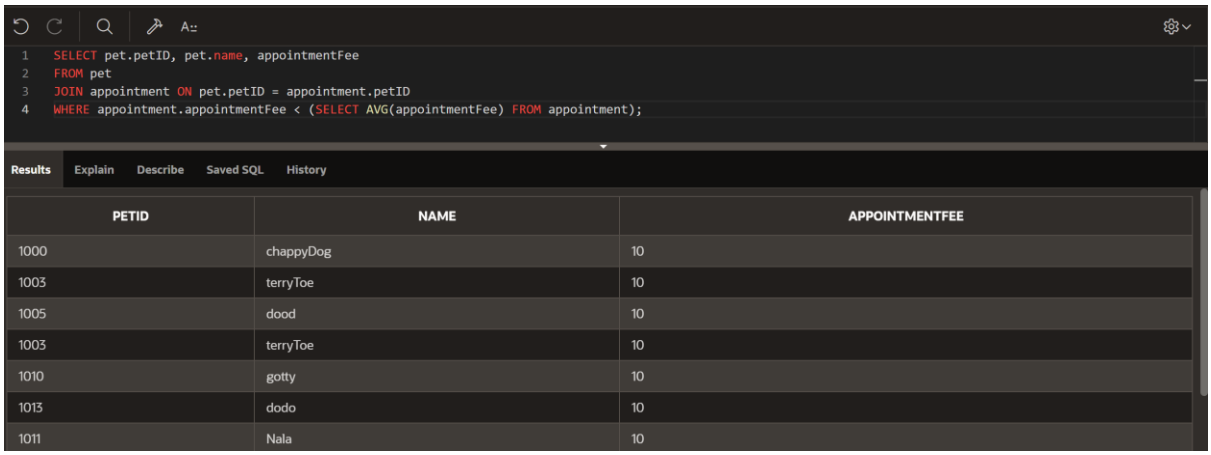
Results | Explain | Describe | Saved SQL | History

PETID	NAME	APPOINTMENTCOUNT
1000	chappyDog	1
1004	poody	1
1006	dood	1
1010	gotty	1
1002	bullyTom	1
1005	dood	1
1003	terryToe	2

Figure 3. Query 3 output

Query 4: Identifying Pet Details and Appointment Costs Below the Average Cost of All Pet Appointments.

```
SELECT pet.petID, pet.name, appointmentFee
FROM pet
JOIN appointment ON pet.petID = appointment.petID
WHERE appointment.appointmentFee < (SELECT AVG(appointmentFee) FROM appointment);
```



```
SELECT pet.petID, pet.name, appointmentFee
FROM pet
JOIN appointment ON pet.petID = appointment.petID
WHERE appointment.appointmentFee < (SELECT AVG(appointmentFee) FROM appointment);
```

Results | Explain | Describe | Saved SQL | History

PETID	NAME	APPOINTMENTFEE
1000	chappyDog	10
1003	terryToe	10
1005	dood	10
1003	terryToe	10
1010	gotty	10
1013	dodo	10
1011	Nala	10

Figure 4. Query 4 output

Query 5: Retrieving Doctor Details, Status, Total Appointments, Pet id and Pet Name for 'Socialisation' or 'Dental' Diagnoses at Least 4 Weeks Ago.

```
SELECT doctor.doctorID, COUNT(appointment.appointmentID), pet.petID, pet.name,
appointment.appointmentDate, diagnosisRecord.diagnosis
```

```
FROM doctor
JOIN appointment ON doctor.doctorID = appointment.doctorID
JOIN pet ON appointment.petID = pet.petID
JOIN diagnosisRecord ON appointment.appointmentID = diagnosisRecord.appointmentID
WHERE (diagnosisRecord.diagnosis LIKE '%socialisation%' OR diagnosisRecord.diagnosis LIKE
'%dental%') AND appointment.appointmentDate <= ADD_MONTHS(SYSDATE, -1)
GROUP BY doctor.doctorID, pet.petID, pet.name, appointment.appointmentDate,
diagnosisRecord.diagnosis;
```

```
1  SELECT doctor.doctorID, COUNT(appointment.appointmentID), pet.petID, pet.name, appointment.appointmentDate, diagnosisRecord.diagnosis
2  FROM doctor
3  JOIN appointment ON doctor.doctorID = appointment.doctorID
4  JOIN pet ON appointment.petID = pet.petID
5  JOIN diagnosisRecord ON appointment.appointmentID = diagnosisRecord.appointmentID
6  WHERE (diagnosisRecord.diagnosis LIKE '%socialisation%' OR diagnosisRecord.diagnosis LIKE '%dental%') AND appointment.appointmentDate <= ADD_MONTHS(SYSDATE, -1
7  GROUP BY doctor.doctorID, pet.petID, pet.name, appointment.appointmentDate, diagnosisRecord.diagnosis;
```

Results | Explain | Describe | Saved SQL | History

DOCTORID	COUNT(APPOINTMENT.APPOINTMENTID)	PETID	NAME	APPOINTMENTDATE	DIAGNOSIS
2202	1	1003	terryToe	25-Sep-2023	Ultrasonic dental scaling in two weeks
2207	1	1010	gotty	17-Jul-2023	Needs socialisation treats
2201	1	1000	chappyDog	04-Sep-2023	Needs socialisation treats
2203	1	1002	bullyTom	25-Aug-2023	Needs socialisation treats & worming treatment
2212	1	1011	Nala	05-Oct-2023	Needs socialisation treats & worming treatment
2210	1	1013	dodo	27-Oct-2023	Ultrasonic dental scaling in two weeks

Figure 5. Query 5 output

Query 6: Identifying pet ids for pets with no appointments or diagnosis so far.

```
SELECT pet.petID
FROM pet
LEFT JOIN appointment ON pet.petID = appointment.petID
LEFT JOIN diagnosisRecord ON appointment.appointmentID = diagnosisRecord.appointmentID
WHERE appointment.appointmentID IS NULL
AND diagnosisRecord.diagnosisRecordID IS NULL;
```

```
1  SELECT pet.petID
2  FROM pet
3  LEFT JOIN appointment ON pet.petID = appointment.petID
4  LEFT JOIN diagnosisRecord ON appointment.appointmentID = diagnosisRecord.appointmentID
5  WHERE appointment.appointmentID IS NULL
6  AND diagnosisRecord.diagnosisRecordID IS NULL;
```

Results | Explain | Describe | Saved SQL | History

PETID
1014
1009
1012

3 rows returned in 0.00 seconds Download

Figure 6. Query 6 output