Database Design & Implementation

1. Top-down Entity Relationship Diagram

The top-down Entity Relationship Diagram (ERD) in Figure 1 illustrates the primary entities and their attributes within the system before merging it with the complimentary bottom-up ERD to provide a comprehensive view of the system's data model.

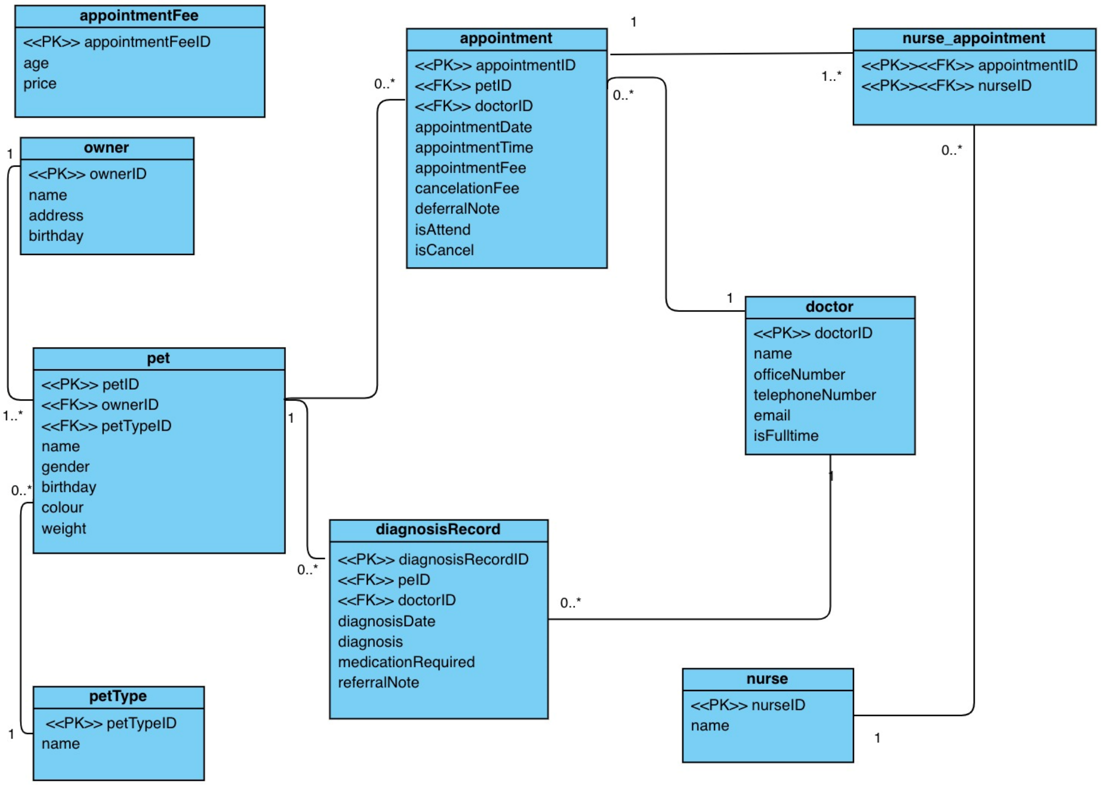
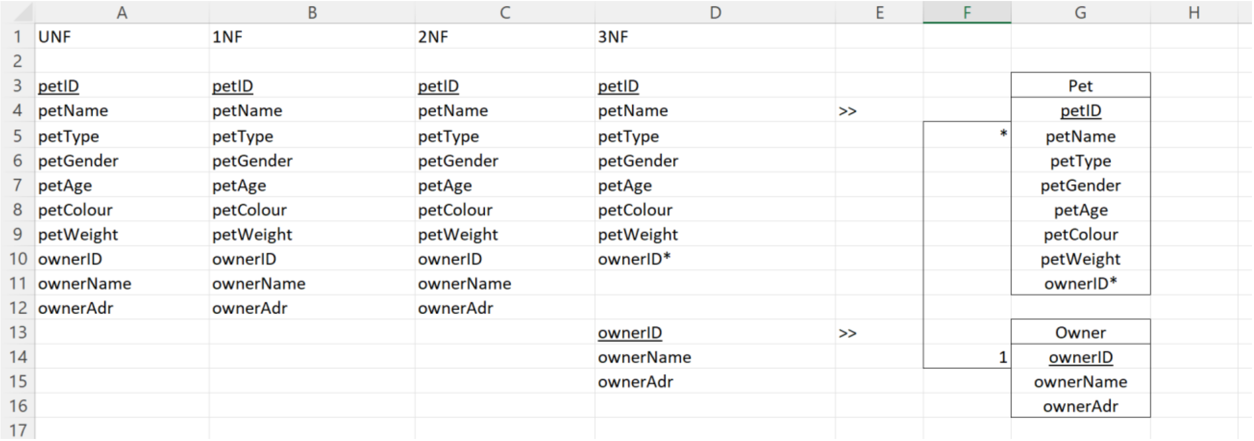


Figure 1. Top-Down Entity Relationship Diagram

2. Normalisation and bottom-up Entity Relationship Diagram

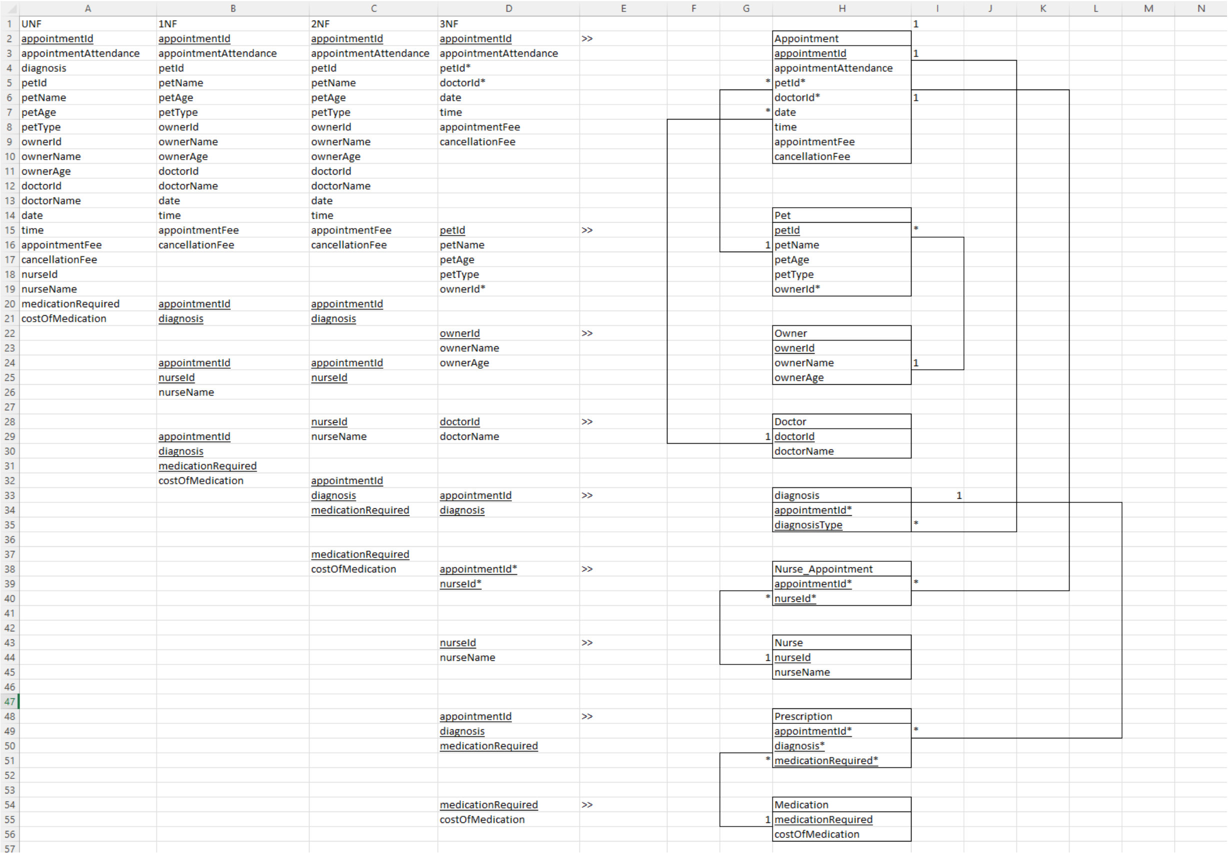
- Pet Registration Form



	A	B	C	D	E	F	G	H
1	UNF	1NF	2NF	3NF				
2								
3	petID	petID	petID	petID			Pet	
4	petName	petName	petName	petName	>>		petID	
5	petType	petType	petType	petType		*	petName	
6	petGender	petGender	petGender	petGender			petType	
7	petAge	petAge	petAge	petAge			petGender	
8	petColour	petColour	petColour	petColour			petAge	
9	petWeight	petWeight	petWeight	petWeight			petColour	
10	ownerID	ownerID	ownerID	ownerID*			petWeight	
11	ownerName	ownerName	ownerName				ownerID*	
12	ownerAdr	ownerAdr	ownerAdr					
13				ownerID	>>		Owner	
14				ownerName		1	ownerID	
15				ownerAdr			ownerName	
16							ownerAdr	
17								

Figure 2. Normalisation and bottom-up ERD for Pet Registration Form

- Appointment Diary and Pet Consultation Form



	A	B	C	D	E	F	G	H	I
1	UNF	1NF	2NF	3NF					1
2	appointmentId	appointmentId	appointmentId	appointmentId	>>			Appointment	
3	appointmentAttendance	appointmentAttendance	appointmentAttendance	appointmentAttendance				appointmentId	1
4	diagnosis	petId	petId	petId*				appointmentAttendance	
5	petId	petName	petName	doctorId*			*	petId*	
6	petName	petAge	petAge	date				doctorId*	1
7	petAge	petType	petType	time			*	date	
8	petType	ownerId	ownerId	appointmentFee				time	
9	ownerId	ownerName	ownerName	cancellationFee				appointmentFee	
10	ownerName	ownerAge	ownerAge					cancellationFee	
11	ownerAge	doctorId	doctorId						
12	doctorId	doctorName	doctorName						
13	doctorName	date	date						
14	date	time	time					Pet	
15	time	appointmentFee	appointmentFee	petId	>>			petId	*
16	appointmentFee	cancellationFee	cancellationFee	petName			1	petName	
17	cancellationFee			petAge				petAge	
18	nurseId			petType				petType	
19	nurseName			ownerId*				ownerId*	
20	medicationRequired	appointmentId	appointmentId						
21	costOfMedication	diagnosis	diagnosis						
22				ownerId	>>			Owner	
23				ownerName				ownerId	
24		appointmentId	appointmentId	ownerAge				ownerName	1
25		nurseId	nurseId					ownerAge	
26		nurseName							
27									
28			nurseId	doctorId	>>			Doctor	
29		appointmentId	nurseName	doctorName			1	doctorId	
30		diagnosis						doctorName	
31		medicationRequired							
32		costOfMedication	appointmentId						
33			diagnosis	appointmentId	>>			diagnosis	1
34			medicationRequired	diagnosis				appointmentId*	
35								diagnosisType	*
36									
37			medicationRequired						
38			costOfMedication	appointmentId*	>>			Nurse_Appointment	
39				nurseId*				appointmentId*	*
40							*	nurseId*	
41									
42									
43				nurseId	>>			Nurse	
44				nurseName			1	nurseId	
45								nurseName	
46									
47									
48				appointmentId	>>			Prescription	
49				diagnosis				appointmentId*	*
50				medicationRequired				diagnosis*	
51							*	medicationRequired*	
52									
53									
54				medicationRequired	>>			Medication	
55				costOfMedication			1	medicationRequired	
56								costOfMedication	
57									

Figure 3. Normalisation and bottom-up ERD for Appointment Diary and Pet Consultation Form

3. Final Entity Relationship Diagram

This Entity Relationship Diagram represents the result of merging the top-down and the bottom-up ERDs, providing a detailed view of the complete system's data model.

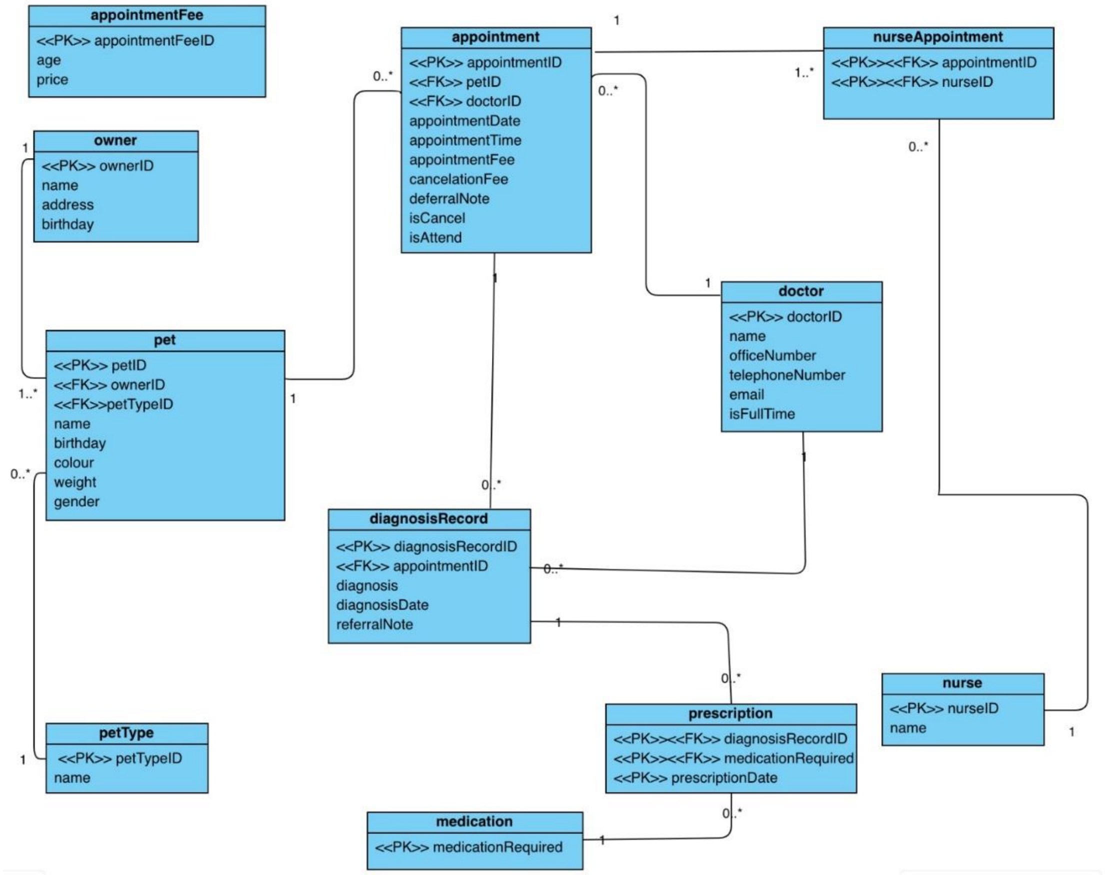


Figure 4. Final Entity Relationship Diagram (ERD)

4. Create and Drop SQL statements

The Create and Drop SQL statements for the system are detailed below.

```
DROP TABLE appointmentFee;
DROP TABLE nurseAppointment;
DROP TABLE nurse;
DROP TABLE prescription;
DROP TABLE diagnosisRecord;
DROP TABLE medication;
DROP TABLE appointment;
DROP TABLE doctor;
DROP TABLE pet;
DROP TABLE petType;
DROP TABLE owner;
```

```
CREATE TABLE owner (

 ownerID NUMBER(10) PRIMARY KEY,
 name VARCHAR2(30) NOT NULL,
 address VARCHAR2(30) NOT NULL,
 birthday DATE
);
```

```
CREATE TABLE petType (
 petTypeID NUMBER(10) PRIMARY KEY,
  name VARCHAR2(30) NOT NULL
);
```

```
CREATE TABLE pet (
  petID NUMBER(4) GENERATED ALWAYS as IDENTITY(START with 1000 INCREMENT by 1) ,
  ownerID NUMBER(10) NOT NULL,
  petTypeID NUMBER(10) NOT NULL,
  name VARCHAR2(20) NOT NULL,
  gender CHAR(1) CHECK (gender IN ('M', 'F')),
  birthday DATE, -- we left the age validation to the application layer.
  colour VARCHAR2(20),
  weight NUMBER(5,2),
  CONSTRAINT pk_pet PRIMARY KEY (petID),
  CONSTRAINT check_petID CHECK ( petID BETWEEN 1000 AND 3000),
  CONSTRAINT fk_pet_owner FOREIGN KEY (ownerID) REFERENCES owner (ownerID),
  CONSTRAINT fk_pet_type FOREIGN KEY (petTypeID) REFERENCES petType (petTypeID)
);
```

```
CREATE TABLE doctor (
  doctorID NUMBER(10) PRIMARY KEY,
  name VARCHAR2(30) NOT NULL,
  officeNumber VARCHAR2(10),
  telephoneNumber VARCHAR2(11) NOT NULL,
  email VARCHAR2(30) NOT NULL,
  isFulltime CHAR(1) CHECK (isFulltime IN ('Y', 'N')),
  CONSTRAINT doctor_email UNIQUE (email)
);
```

```
CREATE TABLE appointment (
  appointmentID NUMBER(10) PRIMARY KEY,
  petID NUMBER(10) NOT NULL,
  doctorID NUMBER(10) NOT NULL,
  appointmentDate DATE, -- we left the date validation to the application layer
  appointmentTime NUMBER (5),
  isAttend CHAR(1) CHECK (isAttend IN ('Y', 'N')),
  appointmentFee NUMBER(2) CHECK (appointmentFee IN (10, 15, 20)),
  cancellationFee NUMBER(1) CHECK (cancellationFee IN (0, 5)),
  deferralNote VARCHAR2(100),
  isCancel CHAR(1) CHECK (isCancel IN ('Y', 'N')),
  CONSTRAINT fk_pet_appointment FOREIGN KEY (petID) REFERENCES pet (petID),
  CONSTRAINT fk_doctor_appointment FOREIGN KEY (doctorID) REFERENCES doctor (doctorID)
);
```

```
CREATE TABLE medication (
  medicationRequired VARCHAR2(30) PRIMARY KEY
);
```

```
CREATE TABLE diagnosisRecord (
  diagnosisRecordID NUMBER(10) PRIMARY KEY,
  appointmentID NUMBER(10) NOT NULL,
  diagnosisDate DATE NOT NULL,
  diagnosis VARCHAR2(100),
  referralNote VARCHAR2(100),
  CONSTRAINT fk_diagnosisRecord_appointment FOREIGN KEY (appointmentID)
REFERENCES appointment (appointmentID)
);
```

```
CREATE TABLE prescription (
  diagnosisRecordID NUMBER(10),
  medicationRequired VARCHAR2(30),
  prescriptionDate DATE,
  CONSTRAINT fk_prescription_medication FOREIGN KEY (medicationRequired)
REFERENCES medication (medicationRequired),
  CONSTRAINT fk_prescription_diagnosisRecord FOREIGN KEY (diagnosisRecordID)
REFERENCES diagnosisRecord (diagnosisRecordID),
  CONSTRAINT pk_prescription PRIMARY KEY (diagnosisRecordID, medicationRequired,
prescriptionDate)
);
```

```
CREATE TABLE nurse (
  nurseID NUMBER(10) PRIMARY KEY,
  name VARCHAR2(30) NOT NULL
);
```

```
CREATE TABLE nurseAppointment (
  appointmentID NUMBER(10),
  nurseID NUMBER(10),
  CONSTRAINT pk_nurseAppointment PRIMARY KEY (appointmentID, nurseID),
  CONSTRAINT fk_nurseAppointment_appointment FOREIGN KEY (appointmentID)
REFERENCES appointment(appointmentID),
  CONSTRAINT fk_nurseAppointment_nurse FOREIGN KEY (nurseID) REFERENCES
nurse(nurseID)
);
```

```
CREATE TABLE appointmentFee (
  appointmentFeeID NUMBER(10) PRIMARY KEY,
  age NUMBER(5) CHECK (age IN (5, 10, 12)),
  price NUMBER(2) CHECK (price IN (10, 15, 20))
);
```

5. Insert SQL statements

Provided below are example SQL INSERT statements for each table, illustrating the process of inserting data into the corresponding database tables.

- Inserting Data into the “owner” Table.

```
INSERT INTO owner (ownerID, name, address, birthday)
VALUES (0002, 'Sam', 'test001 road', '02-FEB-1992');
```

- Inserting Data into the “petType” Table

```
INSERT INTO petType (petTypeID, name) VALUES (002, 'Chiwawa');
```

- Inserting Data into the “pet” Table

```
INSERT INTO pet (ownerID, petTypeID, name, gender, birthday, colour, weight)
VALUES (0009, 014, 'Nala', 'F', '05-DEC-2020', 'white and grey', 3);
```

- Inserting Data into the “doctor” Table

```
INSERT INTO doctor (doctorID, name, officeNumber, telephoneNumber, email, isFulltime)
VALUES (2211, 'Alice Brown', '19', '07123456795', 'alice.brown@example.com', 'N');
```

- Inserting Data into the “appointment” Table

```
INSERT INTO appointment (appointmentID, petID, doctorID, appointmentDate,
appointmentTime, isAttend, appointmentFee, cancellationFee, deferralNote, isCancel)
VALUES (0008, 1008, 2202, '22-SEP-2023', 14, 'Y', 15, 0, 'N/A', 'N');
```

- Inserting Data into the “medication” Table

```
INSERT INTO medication (medicationRequired) VALUES ('painkiller');
```

- Inserting Data into the “diagnosisRecord” Table

```
INSERT INTO diagnosisRecord (diagnosisRecordID, appointmentID, diagnosisDate, diagnosis,
referralNote)
VALUES (105236, 0015, '01-NOV-2023', 'Overgrown skin', 'N/A');
```

- Inserting Data into the “prescription” Table

```
INSERT INTO prescription (diagnosisRecordID, medicationRequired, prescriptionDate)
VALUES (105235, 'steroid', '10-OCT-2023');
```

- Inserting Data into “nurse” Table

```
INSERT INTO nurse (nurseID, name) VALUES (0002, 'Gabby');
```

- Inserting Data into “nurseAppointment” Table

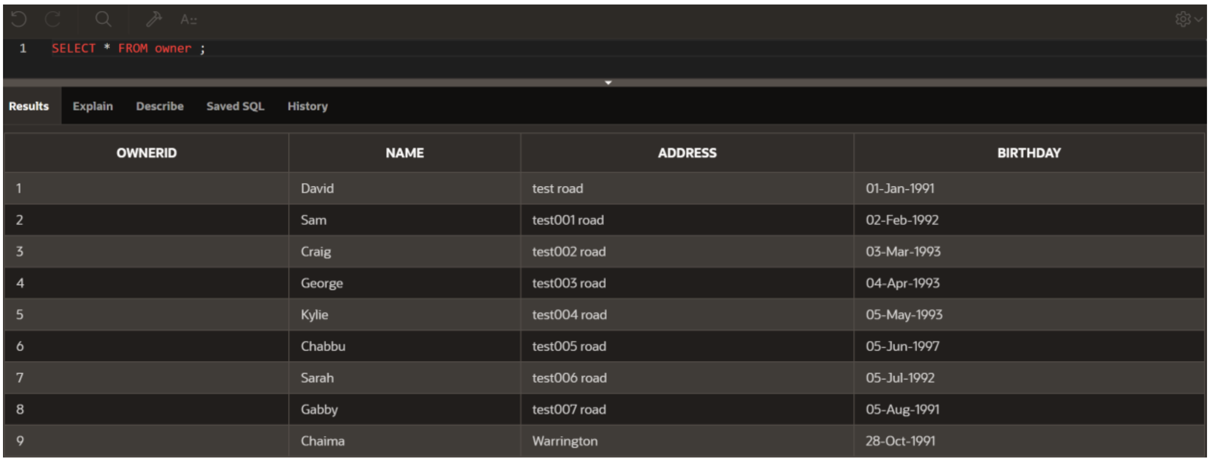
```
INSERT INTO nurseAppointment (appointmentID, nurseID) VALUES (0003, 0003);
```

- Inserting Data into “appointmentFee” Table

```
INSERT INTO appointmentFee (appointmentFeeID, age, price) VALUES (002, 10, 15);
```

6. SELECT Query Results

- Retrieving all Data from the “owner” Table

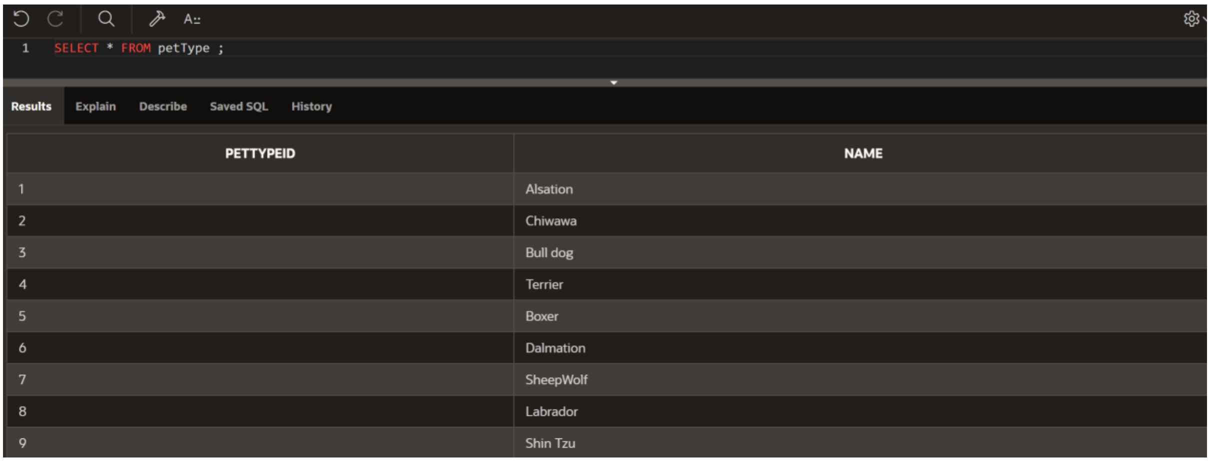


```sql
SELECT * FROM owner ;
```

OWNERID	NAME	ADDRESS	BIRTHDAY
1	David	test road	01-Jan-1991
2	Sam	test001 road	02-Feb-1992
3	Craig	test002 road	03-Mar-1993
4	George	test003 road	04-Apr-1993
5	Kylie	test004 road	05-May-1993
6	Chabbu	test005 road	05-Jun-1997
7	Sarah	test006 road	05-Jul-1992
8	Gabby	test007 road	05-Aug-1991
9	Chaima	Warrington	28-Oct-1991

Figure 5. Owner Table Data Retrieval

- Retrieving all Data from the “petType” Table

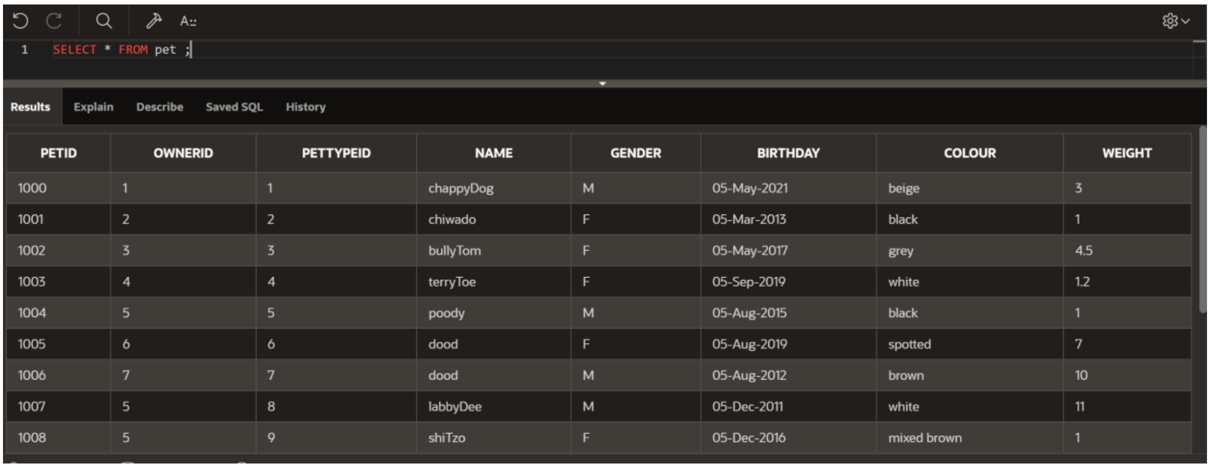


```sql
SELECT * FROM petType ;
```

PETTYPEID	NAME
1	Alsation
2	Chiwawa
3	Bull dog
4	Terrier
5	Boxer
6	Dalmation
7	SheepWolf
8	Labrador
9	Shin Tzu

Figure 6. Pet Type Table Data Retrieval

- Retrieving all Data from the “pet” Table

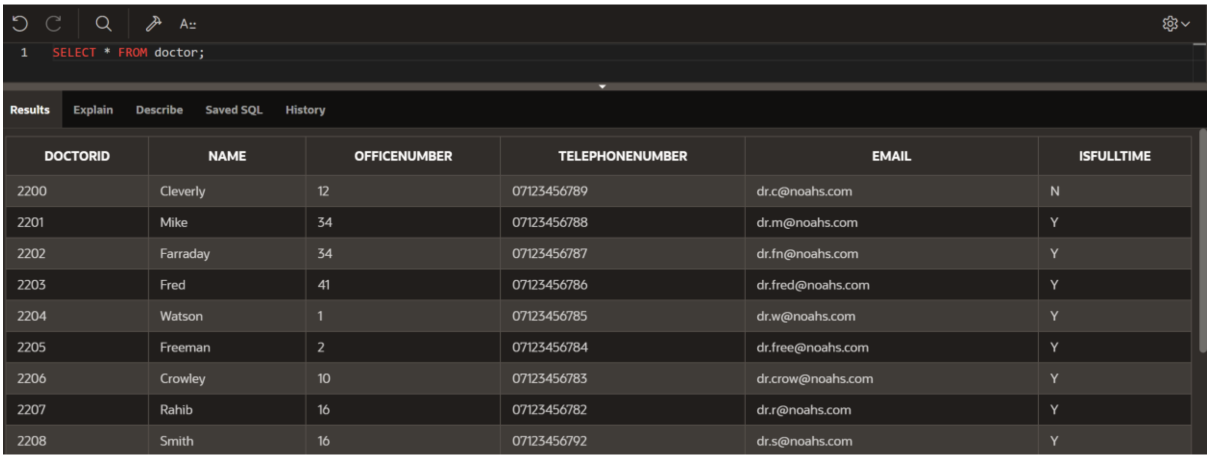


```sql
SELECT * FROM pet ;
```

PETID	OWNERID	PETTYPEID	NAME	GENDER	BIRTHDAY	COLOUR	WEIGHT
1000	1	1	chappyDog	M	05-May-2021	beige	3
1001	2	2	chiwado	F	05-Mar-2013	black	1
1002	3	3	bullyTom	F	05-May-2017	grey	4.5
1003	4	4	terryToe	F	05-Sep-2019	white	1.2
1004	5	5	poody	M	05-Aug-2015	black	1
1005	6	6	dood	F	05-Aug-2019	spotted	7
1006	7	7	dood	M	05-Aug-2012	brown	10
1007	5	8	labbyDee	M	05-Dec-2011	white	11
1008	5	9	shiTzo	F	05-Dec-2016	mixed brown	1

Figure 7. Pet Table Data Retrieval

- Retrieving all Data from the “doctor” Table

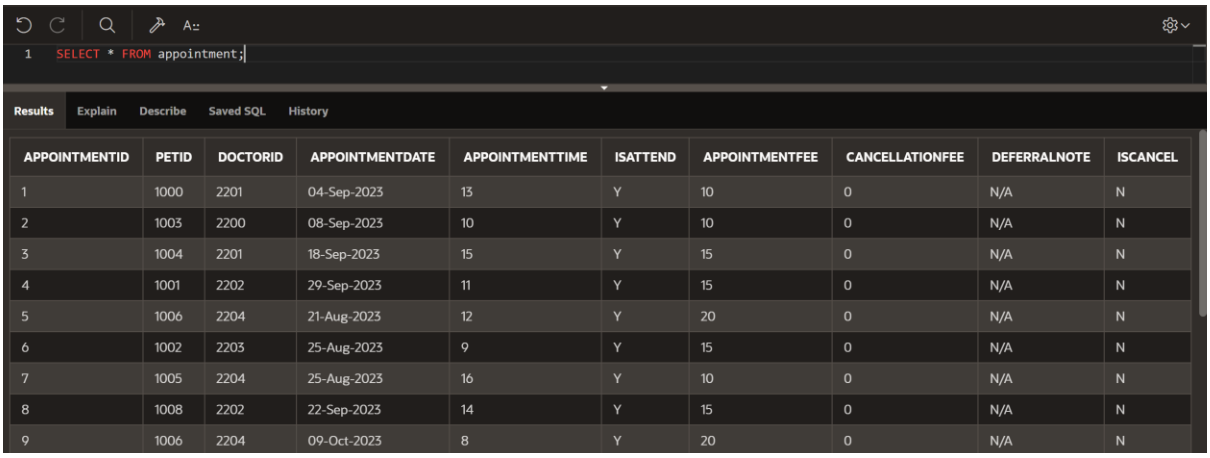


```
1   SELECT * FROM doctor;
```

DOCTORID	NAME	OFFICENUMBER	TELEPHONENUMBER	EMAIL	ISFULLTIME
2200	Cleverly	12	07123456789	dr.c@noahs.com	N
2201	Mike	34	07123456788	dr.m@noahs.com	Y
2202	Farraday	34	07123456787	dr.fn@noahs.com	Y
2203	Fred	41	07123456786	dr.fred@noahs.com	Y
2204	Watson	1	07123456785	dr.w@noahs.com	Y
2205	Freeman	2	07123456784	dr.free@noahs.com	Y
2206	Crowley	10	07123456783	dr.crow@noahs.com	Y
2207	Rahib	16	07123456782	dr.r@noahs.com	Y
2208	Smith	16	07123456792	dr.s@noahs.com	Y

Figure 8. Doctor Table Data Retrieval

- Retrieving all Data from the "appointment” Table

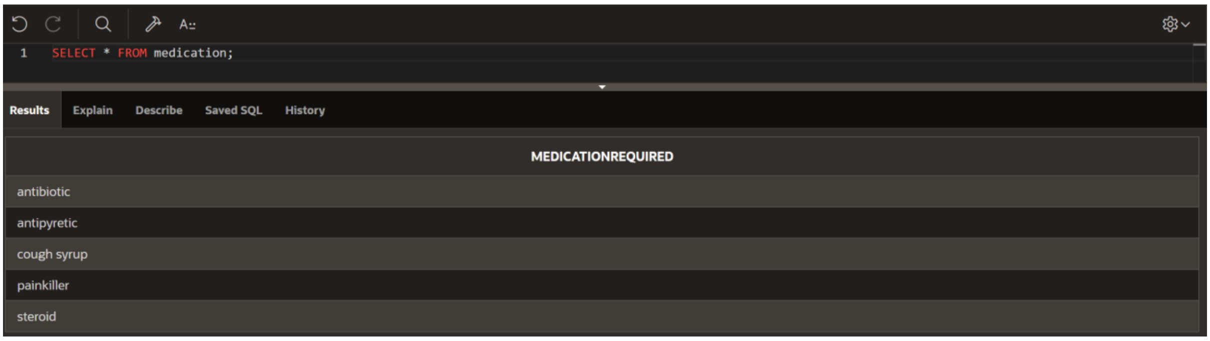


```
1   SELECT * FROM appointment;
```

APPOINTMENTID	PETID	DOCTORID	APPOINTMENTDATE	APPOINTMENTTIME	ISATTEND	APPOINTMENTFEE	CANCELLATIONFEE	DEFERRALNOTE	ISCANCEL
1	1000	2201	04-Sep-2023	13	Y	10	0	N/A	N
2	1003	2200	08-Sep-2023	10	Y	10	0	N/A	N
3	1004	2201	18-Sep-2023	15	Y	15	0	N/A	N
4	1001	2202	29-Sep-2023	11	Y	15	0	N/A	N
5	1006	2204	21-Aug-2023	12	Y	20	0	N/A	N
6	1002	2203	25-Aug-2023	9	Y	15	0	N/A	N
7	1005	2204	25-Aug-2023	16	Y	10	0	N/A	N
8	1008	2202	22-Sep-2023	14	Y	15	0	N/A	N
9	1006	2204	09-Oct-2023	8	Y	20	0	N/A	N

Figure 9. Appointment Table Data Retrieval

- Retrieving all Data from the “medication” Table

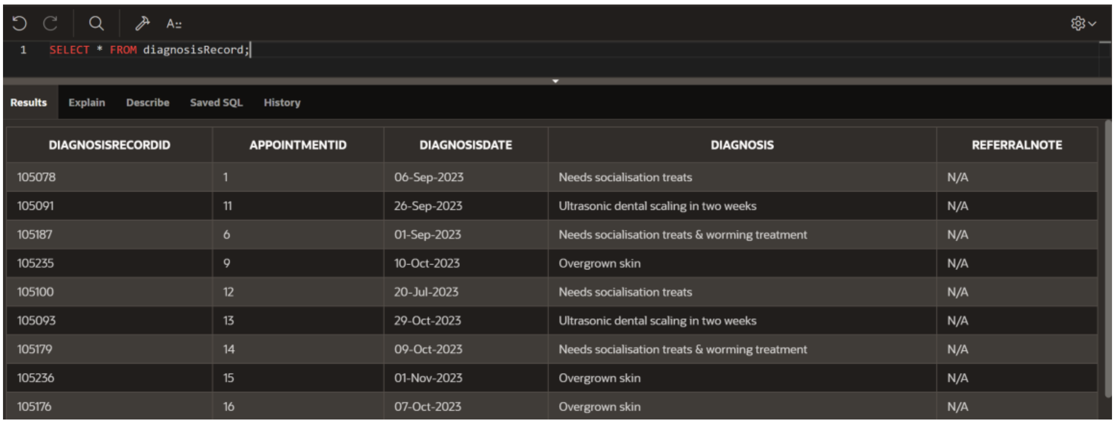


```
1   SELECT * FROM medication;
```

MEDICATIONREQUIRED
antibiotic
antipyretic
cough syrup
painkiller
steroid

Figure 10. Medication Table Data Retrieval

- Retrieving all Data from the “diagnosisRecord” Table

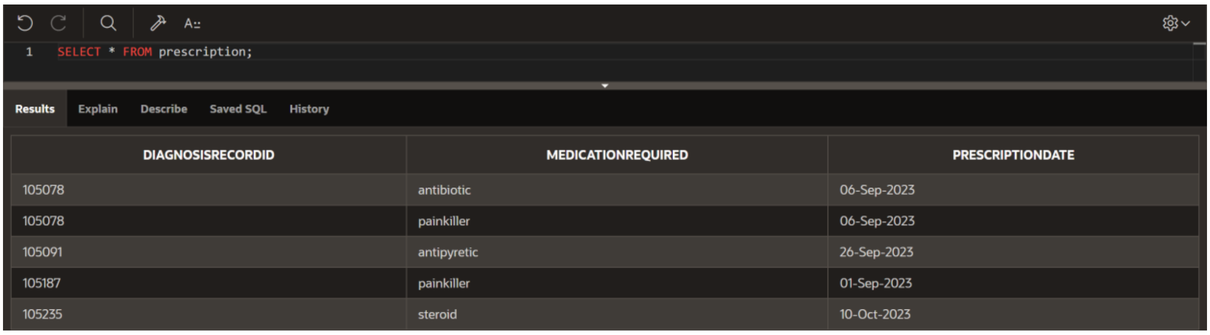


```
SELECT * FROM diagnosisRecord;
```

DIAGNOSISRECORDID	APPOINTMENTID	DIAGNOSISDATE	DIAGNOSIS	REFERRALNOTE
105078	1	06-Sep-2023	Needs socialisation treats	N/A
105091	11	26-Sep-2023	Ultrasonic dental scaling in two weeks	N/A
105187	6	01-Sep-2023	Needs socialisation treats & worming treatment	N/A
105235	9	10-Oct-2023	Overgrown skin	N/A
105100	12	20-Jul-2023	Needs socialisation treats	N/A
105093	13	29-Oct-2023	Ultrasonic dental scaling in two weeks	N/A
105179	14	09-Oct-2023	Needs socialisation treats & worming treatment	N/A
105236	15	01-Nov-2023	Overgrown skin	N/A
105176	16	07-Oct-2023	Overgrown skin	N/A

Figure 11. Diagnosis Record Table Data Retrieval

- Retrieving all Data from the “prescription” Table

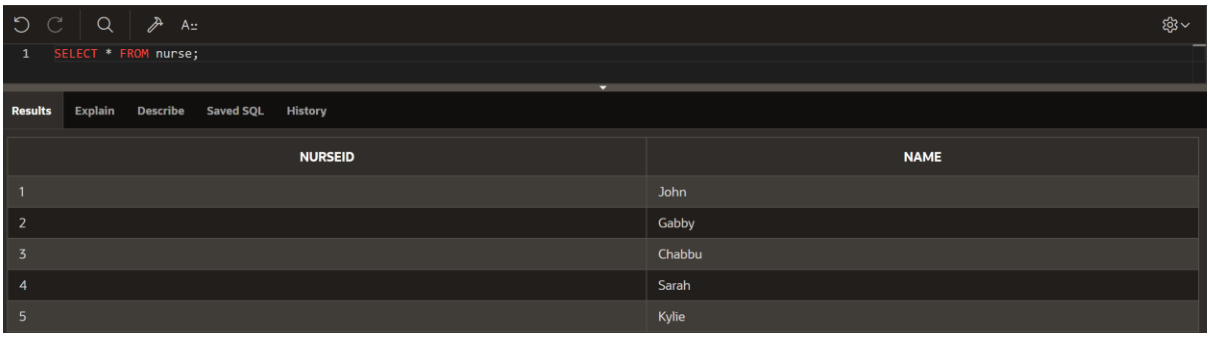


```
SELECT * FROM prescription;
```

DIAGNOSISRECORDID	MEDICATIONREQUIRED	PRESCRIPTIONDATE
105078	antibiotic	06-Sep-2023
105078	painkiller	06-Sep-2023
105091	antipyretic	26-Sep-2023
105187	painkiller	01-Sep-2023
105235	steroid	10-Oct-2023

Figure 12. Prescription Table Data Retrieval

- Retrieving all Data from the “nurse” Table

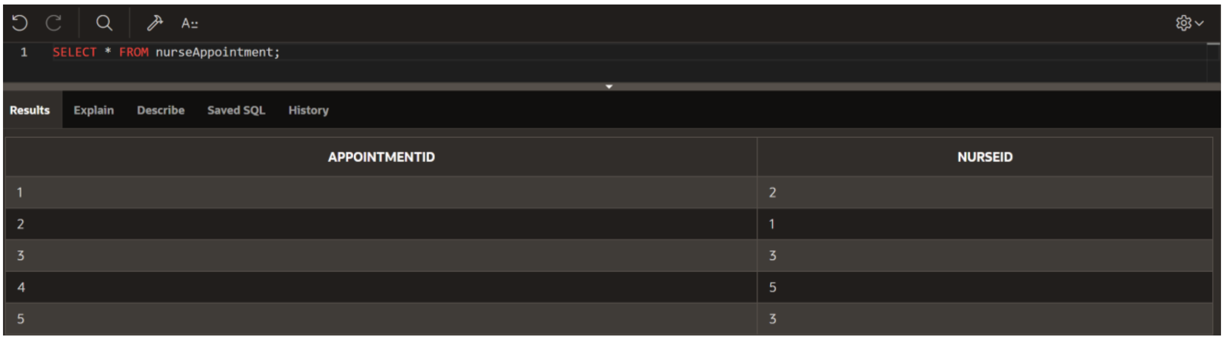


```
SELECT * FROM nurse;
```

NURSEID	NAME
1	John
2	Gabby
3	Chabbu
4	Sarah
5	Kylie

Figure 13. Nurse Table Data Retrieval

- Retrieving all Data from the “nurseAppointment” Table

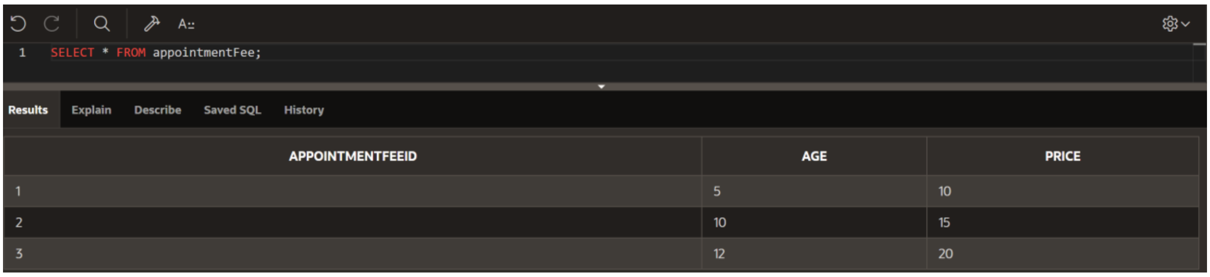


```
SELECT * FROM nurseAppointment;
```

APPOINTMENTID	NURSEID
1	2
2	1
3	3
4	5
5	3

Figure 14. Nurse Appointment Table Data Retrieval

- Retrieving all Data from the “appointmentFee” Table



```
SELECT * FROM appointmentFee;
```

APPOINTMENTFEEID	AGE	PRICE
1	5	10
2	10	15
3	12	20

Figure 15. Appointment Table Data Retrieval

7. Critical Analysis of the System

In response to the interest expressed by a large vet franchise in our solution, a critical analysis of the current design and database state has been undertaken.

The following recommendations outline three key strategies for redesigning the database to enhance scalability and performance for a larger client:

Scalability and Performance

- Currently, the system works well for the clinic's immediate needs. However, it's important to make it adaptable for a growing vet franchise.
- Instead of putting all the data on one server, we suggest spreading it across many servers using a distributed database. This implies the system can handle more data as the franchise grows. Furthermore, we can use database sharding, which organises data depending on factors such as location.
- If the vet franchise expands, there’ll be more data to handle, thus the distributed database will ensure the efficiency and responsiveness within the system. Moreover, sharding will assist in distributing and retrieving data effectively, ultimately improving overall performance.

Data Protection and Compliance

- The current report shows that standards are being followed, but we must focus on data security and following regulations more vigorously.
- We'll enhance database security by using strong encryption for sensitive data. Also, to comply with industry standards such as GDPR, we'll guarantee that our practices align, and regular security audits will also be conducted to identify and fix potential risks.
- Using strong encryption and complying with data protection standards is vital, as the clinic deals with vast quantities of sensitive information. This means trust with pet owners is well maintained and legal requirements are met. Furthermore, consistent security checks will minimise such risks and threats.

Integration

- The system lacks integration with other external systems.
- We recommend creating the database with an open design that allows smooth integration with other systems like accounting, inventory management, or customer relationship management (CRM) systems.
- A growing vet franchise may need centralised management and reporting systems, as well as including an open design. Because it will ensure effective communication and exchanging information will be far more efficient.