Pet Clinic Project

Klementyna Kwiatkowska

Alasdair Craig

Sabah Ali

Bukola Stephen

Systems Analysis

Executive summary

This project aims to produce a new system for Noah's pet clinic as a current system the clinic has in place is not satisfactory. At the moment, the clinic's system is paper based which means everything from pet records to booking appointments is all written down. As all the documents are taken care of by all the receptionist, nurses and doctors, it is easy to lose documents. To improve the current system, our team decided to produce an online database for all the vet staff to access when needed. The database will include a registration and booking system where owners will be able to register multiple pets and book appointments with specific doctors. The database will also store information about pets and pet owners along with information regarding vet staff such as their working schedule and contact information.

To improve the current system the following steps were carried out:

* List of system and user requirements
* A use case diagram was created to see which staff member used the system most and what for
* Use case specifications were created to show further information about the use case diagram.
* A normalisation of the appointment diary, pet registration form and the consultation form were done to condense the information on three forms down to the simplest form.
* From the normalisation an entity-relationship diagram was produced. This was guidance as to which information needed to be included in the diagram.
* Using SQL, the database was created with the previous customer data inputted into the new database.
* Once the database was created queries were ran to see of the database ran correctly.

System goals

* To make the booking and pet files accessible to all nurses and doctors.
* A system that keeps track of the date, time, nurse, doctor, pet owner’s name and age, pets name and type.
* A system to keep track of payments following the appointment or cancelation.
* A system that both doctors and nurses can enter the outcome of consultation.
* A system that customers can use to register at the clinic.
* A system that organises all pets’ owners name in alphabetical order.
* A system which stores information about doctors and nurses name, office number, telephone number, email and whether they are full-time or part-time.
* A system which doctors can use to make referrals or deferral.
* A system which doctors can use to update consultation cards (including diagnosis and medications).
* A system which regularly updates a pet age every year.

System Requirements

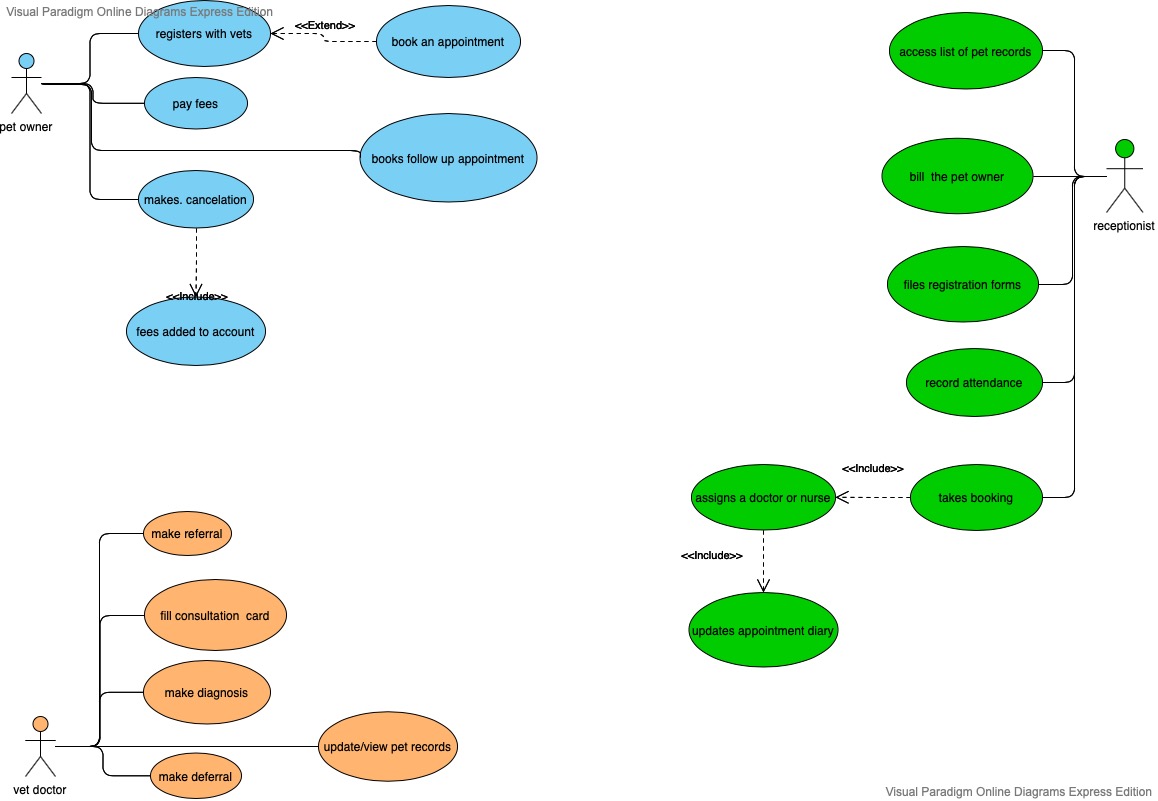
Functional requirements

* The system shall allow new customers to register
* The system should allow customers to book appointments online
* The system shall organize doctors’ and nurses’ records including name, office number, hours worked and email address
* The system shall organize pet details including: dog ID, breed, name, age, colour, weight, sex and dog owner
* The system shall update pet’s age
* The system shall display detailed description of pet record including previous appointments, diagnosis, referrals, deferrals and medications
* The system shall display upcoming/scheduled appointments including date, pet ID and doctor ID
* The system shall register any appointment cancelations and update diary
* The system shall assign cancelation fee to owner’s account
* The system should list pet owner’s in alphabetical order
* The system should allow for pet details to be searched for based on the owner’s name
* The system shall display the number of scheduled appointments by each doctor and nurse
* The system shall send appointment remainders
* The system shall display when the clinic is fully booked
* The system shall accept payment via all major debit and credit cards
* The system shall validate the payment with appropriate body

Non-functional requirements

* The system shall send confirmation of registration within 24 hours
* The system shall log in a user within 5 seconds
* The system shall appoint a £5 cancelation fee to owners account within 5 minutes of appointment cancelation if cancelled within 24 hours of the appointment date
* The system shall update booking system within 10 seconds
* The system shall make changes to the appointment details within 5 seconds of registering it
* The system shall be able allow access to database with all pet, clinical staff and appointment records within 10 seconds
* The system shall update pet’s age according to date
* The system shall automatically place files in alphabetical order
* The system shall be available for use 24 hours a day, 7 days a week
* The system shall allow for registration and appointments to be made 24 hours a day, 7 days a week
* The system shall allow users an easy access to all stored data (pet records, appointment, referrals, deferrals, diagnosis and outstanding charges)
* The system shall support 50 simultaneous users/sessions active
* The system shall send a cancelation email to customers within 24 hours

Use Case Diagram (UCD)



Thought process for the UCD

To create the UCD (use case diagram) for Noah’s pet clinic the scenario was read through to pick out the main characters and their roles, after a series of discussions with the team we decided on having just three main characters. These three main characters were the pet owner, receptionist and the vet doctor. The reason why we decided not to include the vet nurse, or any other clinical staff is that either their roles were covered by the main three characters or they did not play a significant part in the clinic.

The UC diagram starts with the pet owner. The pet owner can register their pet at the clinic, book a follow-up appointment if required and pay fees and cancel an appointment. According to the scenario, these are the main roles of the pet owner. Therefore, as a group we decided to place these roles with the pet owner. The “books an appointment” use case was not placed on the pet owner because it is covered by the “takes booking” on the receptionist character.

The diagram then leads to the receptionist which plays the biggest part in the UC diagram, as in the scenario the receptionist oversees many jobs such as takes bookings and then assigns a doctor or nurse and finally updates the appointment diary but assigns a doctor or nurse and updates the appointment diary cannot be done unless the receptionist takes a booking. Therefore, they have been linked together using an include. The receptionist also files the registration forms, records the attendance and bills the pet owner. The responsibility of billing the pet owner and whether there is a billings department is not stated in the scenario, so we as a group assumed that the receptionist assumes the responsibility of the billings.

The final main character is the vet doctor who in the scenario fills in the consultation form, makes a diagnosis, makes a referral or deferral and updates/ view the pet records. Some of these tasks are covered by the nurse, but we as a group decide to limit the diagram to three main characters to reduce the repetitiveness. Therefore, we grouped these tasks with the doctor.

The important use cases in the UC diagram were made into a UC specification as shown in section 3B. Through each stage of the UC diagram production, the group was consulted, and any feedback given was discussed and added to the diagram.

Use Case Specifications

|  |
| --- |
| **Use Case: Make a deferral** |
| Owner: Vet doctor |
| **Pre-Conditions** |
| 1. Appointment has been scheduled 2. Appointment has been attended |
| **Post-Conditions** |
| 1. Schedule follow-up appointment |
| **Primary Path** |
| 1. Pet attends the clinic 2. Diagnosis not made 3. A deferral is advised, and note applied to the owner’s account 4. A follow-up appointment is scheduled 5. A confirmation email is sent to pet owner |
| **Alternate Path** |
|  |

|  |
| --- |
| Use Case: Make a referral |
| Owner: Vet doctor |
| Pre-Conditions |
| 1. Appointment has been scheduled 2. Appointment has been attended |
| Post-Conditions |
|  |
| Primary Path |
| 1. Diagnosis is made 2. A referral to another vet is advised 3. A pet file is updated with a referral note 4. Confirmation email is sent to pet owner |
| Alternate Path |
|  |

|  |
| --- |
| Use Case: Fill consultation card |
| Owner: Vet doctor |
| Pre-Conditions |
| 1. Appointment has been scheduled 2. Appointment has been attended |
| Post-Conditions |
| 1. Update the pet file |
| Primary Path |
| * 1. Diagnosis made   2. Consultation card filled   3. Card uploaded to the pet file   4. A copy of diagnosis and medication is sent by email to the pet owner |
| Alternate Path |
| 1. Diagnosis not made 2. Skip the medication and cost 3. Update pet file with consultation description |

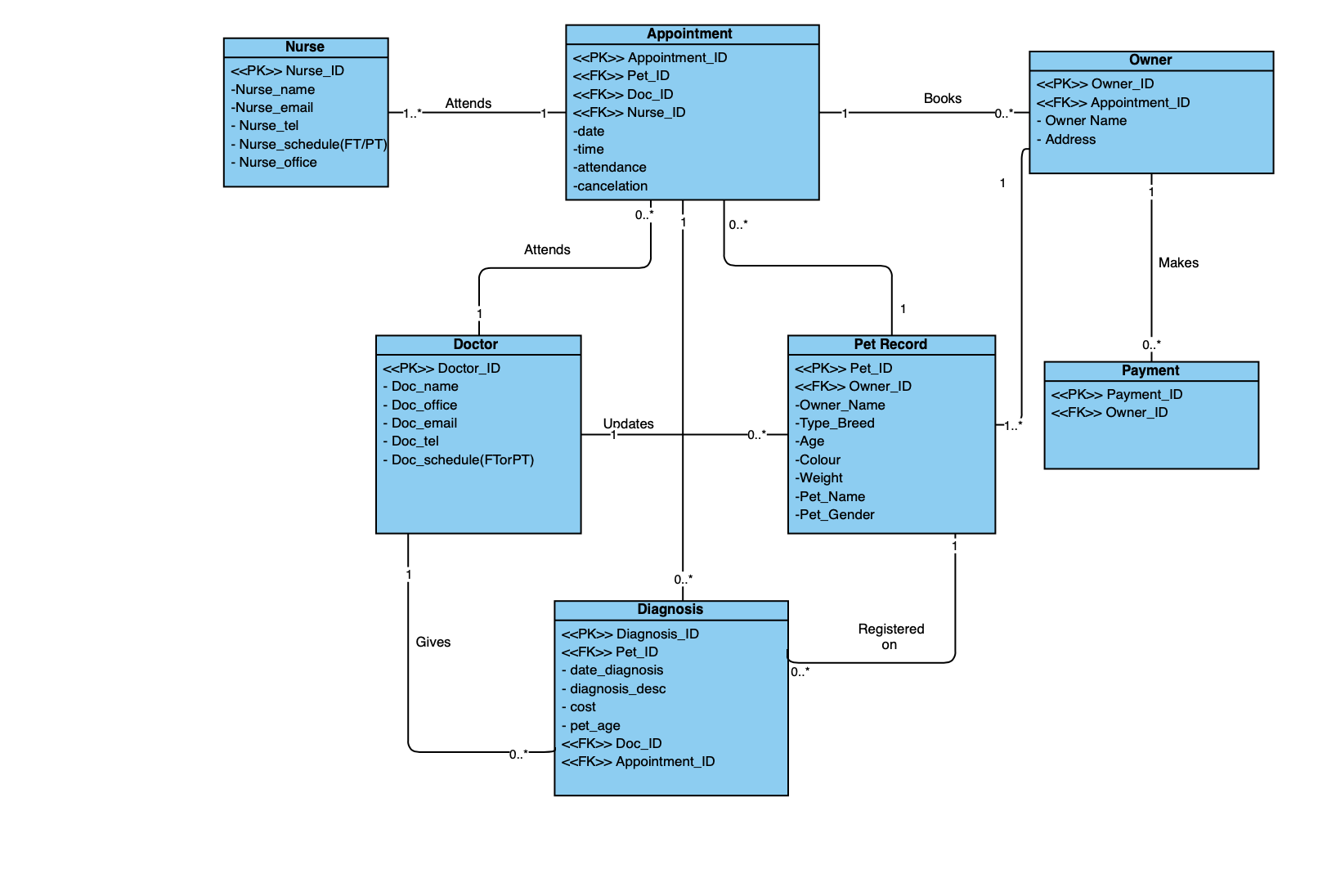
|  |
| --- |
| Use Case: Register with clinic |
| Owner: Pet owner |
| Pre-Conditions |
| * 1. Create an account on vet website using user ID and password |
| Post-Conditions |
| 1. Update the NPC registrations database |
| Primary Path |
| * 1. Login to the NPC website using user ID and password   2. Fill the NPC registration form   3. Submit the form   4. Send confirmation of registration email |
| Alternate Path |
| 1. Login not working    * 1. Help customer set up a new user ID and password      2. Send a confirmation email 2. User forgot the password    * 1. Redirect to setting up a new password page      2. Send a confirmation email 3. Customer needs help with filling and uploading registration forms    * 1. Direct to instructions webpage      2. Direct to reception’s email for further help |

|  |
| --- |
| **Use Case: Pay fees** |
| Owner: Pet owner |
| **Pre-Conditions** |
| Pet owner must have cancelled an appointment or attends an appointment |
| **Post-Conditions** |
| Provide all the details of the consultation to pet owner and charge appropriate amount including any cancelation fees |
| **Primary Path** |
| Reception desk provides all the details of the consultation to pet owner and charges appropriate amount  Pet owner makes a payment at the reception desk  Receptionist gives the pet owner a copy of proof of payment |
| **Alternate Path** |
|  |
| **Notes** |
|  |
|  |
|  |

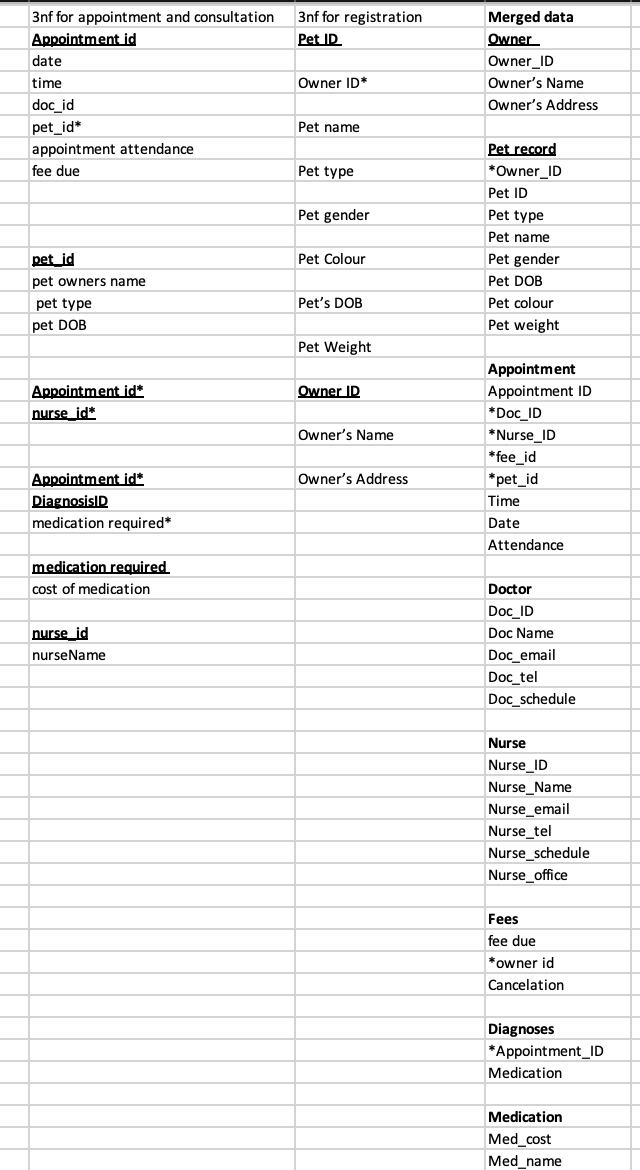
|  |
| --- |
| **Use Case: Book appointment** |
| Owner: Pet owner |
| **Pre-Conditions** |
| Pet owner needs to be registered at the clinic to be able to book a appointment. |
| **Post-Conditions** |
| Attend the appointment or cancels |
| **Primary Path** |
| 1. Phone up the clinic 2. Vet receptionist asks for pet owners’ details to look up on system 3. Vet receptionists asks if the appointment is routine or an emergency 4. Vet receptionist checks available appointment times 5. Books an appointment which is added to the booking system |
| **Alternate Path** |
| No appointments available |

Database Design & Implementation

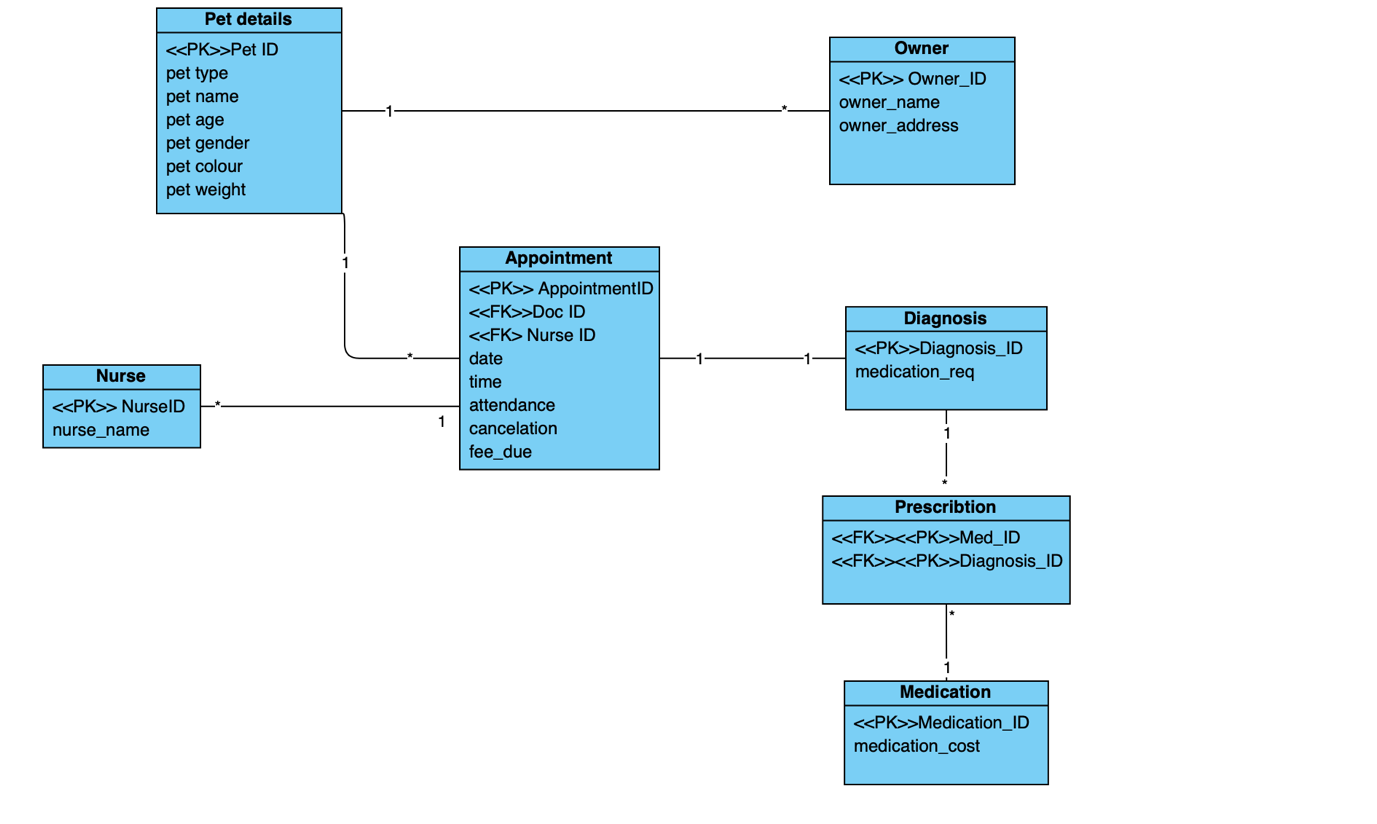
Top down entity relationship diagram



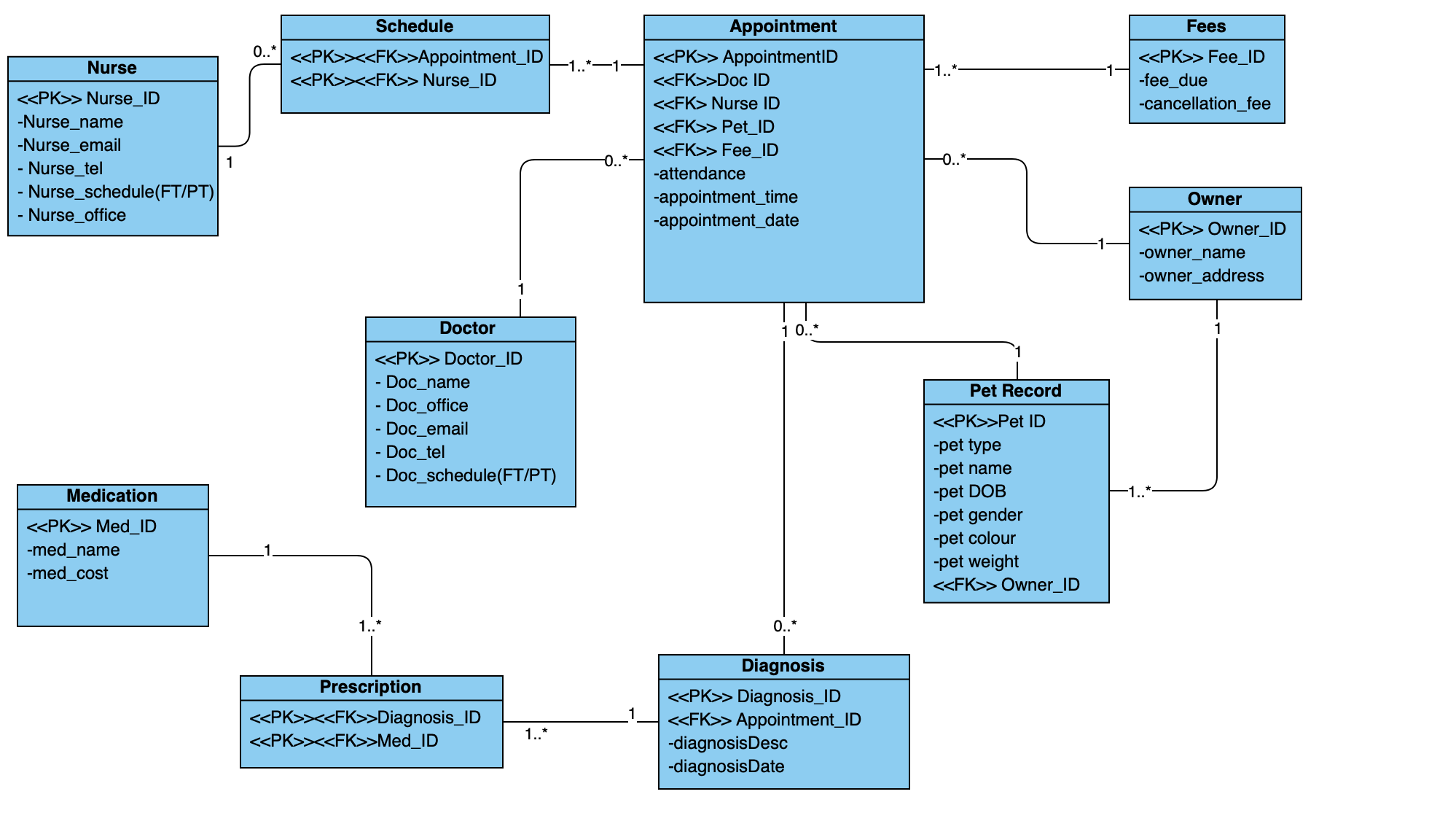
Normalised appointment diary, pet registration form and pet consultation card



Bottom up ERD



Final ERD



ORACLE tables with SQL statements

DROP TABLE NC\_prescription;

DROP TABLE NC\_medication;

DROP TABLE NC\_diagnosis;

DROP TABLE NC\_schedule;

DROP TABLE NC\_appointment;

DROP TABLE NC\_fee;

DROP TABLE NC\_doctor;

DROP TABLE NC\_nurse;

DROP TABLE NC\_pet;

DROP TABLE NC\_owner;

CREATE TABLE NC\_owner (

ownerID NUMBER(8,0) CONSTRAINT ownIDpk PRIMARY KEY,

ownerFNAME VARCHAR(50) NOT NULL,

ownerLNAME VARCHAR(50) NOT NULL,

ownerADDRESS VARCHAR(100)

);

CREATE TABLE NC\_pet (

petID NUMBER(8,0) CONSTRAINT petIDpk PRIMARY KEY,

petTYPE VARCHAR2(30),

petNAME VARCHAR2(30) NOT NULL,

petDOB DATE NOT NULL,

petGENDER CHAR(1),

petCOLOUR VARCHAR2(30),

petWEIGHT NUMBER(5,2),

ownerID NUMBER(8) NOT NULL

);

ALTER TABLE NC\_pet ADD CONSTRAINT ownIDfk FOREIGN KEY (ownerID)

REFERENCES NC\_owner (ownerID);

CREATE TABLE NC\_nurse (

nurseID NUMBER(8,0) CONSTRAINT nurIDpk PRIMARY KEY,

nurseFNAME VARCHAR(50) NOT NULL,

nurseLNAME VARCHAR(50) NOT NULL,

nurseEMAIL VARCHAR2(50),

nurseTEL NUMBER(12),

nurseOFFICE VARCHAR2(5),

nurseSCHEDULE VARCHAR2(2),

CONSTRAINT nurEMAILuk UNIQUE (nurseemail)

);

CREATE TABLE NC\_doctor (

docID NUMBER(8,0) CONSTRAINT docIDpk PRIMARY KEY,

docFNAME VARCHAR2(50),

docLNAME VARCHAR2(50),

docOFFICE VARCHAR2(5),

docEMAIL VARCHAR2(50),

docTEL NUMBER(12),

docSCHEDULE VARCHAR2(2),

CONSTRAINT docEMAILuk UNIQUE (docEMAIL)

);

CREATE TABLE NC\_fee (

feeID NUMBER(8,0) CONSTRAINT feeIDpk PRIMARY KEY,

feeDUE NUMBER(5,2),

cancellationFEE CHAR(1)

);

CREATE TABLE NC\_appointment (

appointID NUMBER(8,0) CONSTRAINT appIDpk PRIMARY KEY,

docID NUMBER(8,0) NOT NULL,

nurseID NUMBER(8,0) NOT NULL,

petID NUMBER(4,0) NOT NULL,

appointDATE DATE,

attendance CHAR(1),

cancellation CHAR(1),

feeID NUMBER(8,2)

);

ALTER TABLE NC\_appointment ADD CONSTRAINT docIDfk FOREIGN KEY (docID)

REFERENCES NC\_doctor (docID);

ALTER TABLE NC\_appointment ADD CONSTRAINT nurIDfk FOREIGN KEY (nurseID)

REFERENCES NC\_nurse (nurseID);

ALTER TABLE NC\_appointment ADD CONSTRAINT petIDfk FOREIGN KEY (petID)

REFERENCES NC\_pet (petID);

ALTER TABLE NC\_appointment ADD CONSTRAINT feeIDfk FOREIGN KEY (feeID)

REFERENCES NC\_fee (feeID);

CREATE TABLE NC\_schedule (

appointID NUMBER(8,0),

nurseID NUMBER(8,0),

CONSTRAINT schedIDpk2 PRIMARY KEY (appointID, nurseID)

);

ALTER TABLE NC\_schedule ADD CONSTRAINT appIDfk3 FOREIGN KEY (appointID)

REFERENCES NC\_appointment (appointID);

ALTER TABLE NC\_schedule ADD CONSTRAINT nurIDfk2 FOREIGN KEY (nurseID)

REFERENCES NC\_nurse (nurseID);

CREATE TABLE NC\_diagnosis (

diagnosisID NUMBER(8,0) CONSTRAINT diagIDpk PRIMARY KEY,

appointID NUMBER(8,0) NOT NULL,

diagnosisDESC VARCHAR2(200),

diagnosisDATE DATE

);

ALTER TABLE NC\_diagnosis ADD CONSTRAINT appdiagIDfk FOREIGN KEY (appointID)

REFERENCES NC\_appointment (appointID);

CREATE TABLE NC\_medication (

medID NUMBER(8,0) CONSTRAINT medIDpk PRIMARY KEY,

medNAME VARCHAR2(50),

medCOST NUMBER(5,2)

);

CREATE TABLE NC\_prescription (

diagnosisID NUMBER(8,0),

medID NUMBER(8,0),

CONSTRAINT presIDpk PRIMARY KEY (diagnosisID, medID)

);

ALTER TABLE NC\_prescription ADD CONSTRAINT presIDfk FOREIGN KEY (diagnosisID)

REFERENCES NC\_diagnosis (diagnosisID);

ALTER TABLE NC\_prescription ADD CONSTRAINT medIDfk FOREIGN KEY (medID)

REFERENCES NC\_medication (medID);

SQL ‘Insert’ statement for each table

INSERT INTO NC\_fee (feeID,feeDUE,cancellationFEE)

VALUES (00001, 5, 'Y');

INSERT INTO NC\_owner (ownerID,ownerFNAME,ownerLNAME,ownerADDRESS)

VALUES (40001,'David','Aintnohassle','51 issomehassle lane');

INSERT INTO NC\_pet (petID, petTYPE, petNAME, petDOB, petGENDER, petCOLOUR, petWEIGHT, ownerID)

VALUES (1000,'Alsation','chappyDog', TO\_DATE('01-01-2018','dd-mm-yyyy'),'M','Beige',3,40001);

INSERT INTO NC\_nurse (nurseID, nurseFNAME, nurseLNAME, nurseEMAIL, nurseTEL, nurseSCHEDULE, nurseOFFICE)

VALUES  ('90001','Sara','Doog','saraD@noahs.com','0121567999','PT','00');

INSERT INTO NC\_doctor (docID, docFNAME, docLNAME, docEMAIL, docTEL, docSCHEDULE, docOFFICE)

VALUES ('2200','Cleverly','CL','cleaverly\_cl@noahs.com', '0121564569','FT','12');

INSERT INTO NC\_appointment (appointID, appointDATE, attendance, cancellation, feeID, docID,nurseID,petID)

VALUES (100601, TO\_DATE('05-08-2019:08:30:00AM','dd-mm-yy:hh:mi:ssam'),'Y','N',00001,2201,90009,1002);

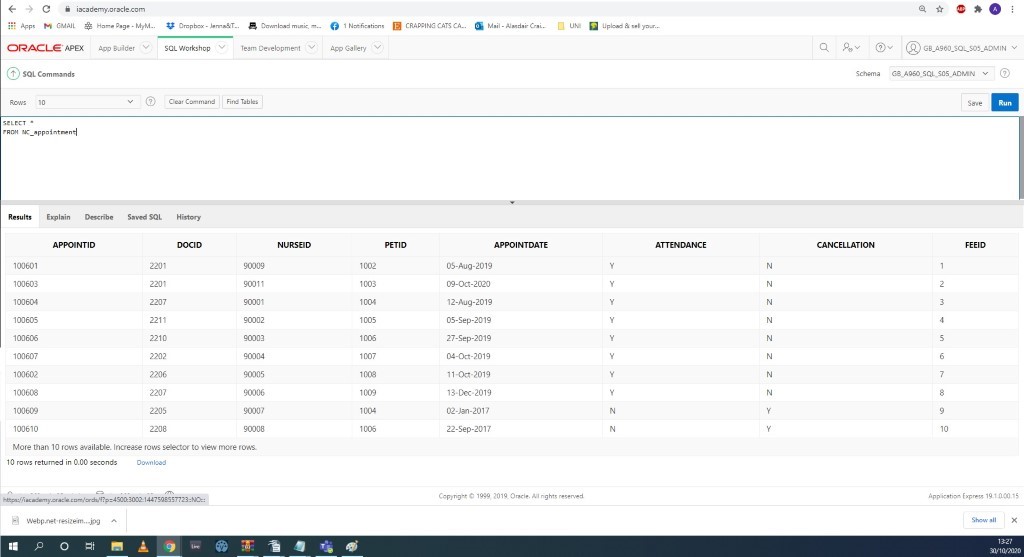
INSERT INTO NC\_diagnosis (diagnosisID, appointID, diagnosisDESC, diagnosisDATE) VALUES (105078, 100601, 'Needs socialisation treats', TO\_DATE('06-09-2019', 'dd-mm-yyyy'));

INSERT INTO NC\_medication (medID, medNAME, medCOST)

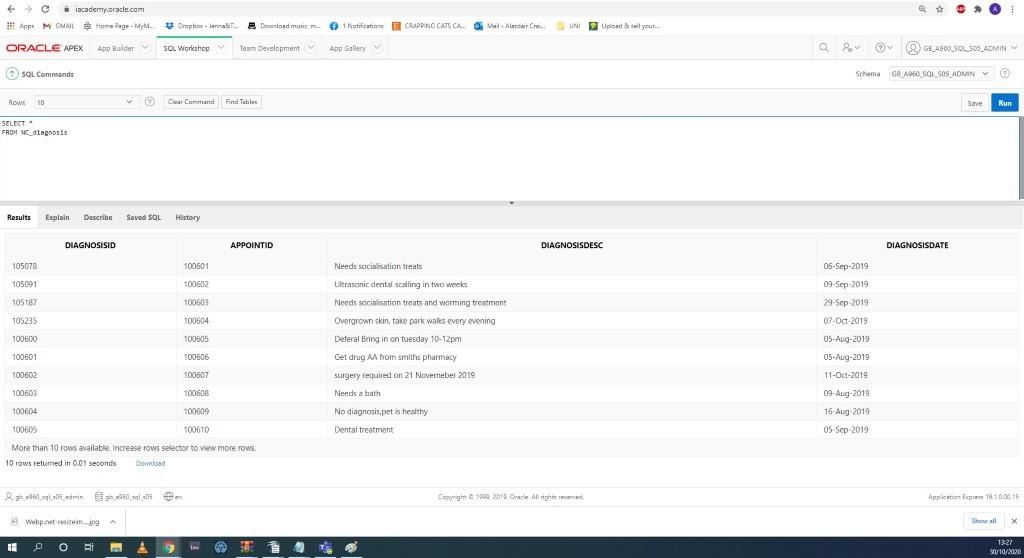
VALUES (00001, 'SorePaws', 5);

SELECT query screenshots

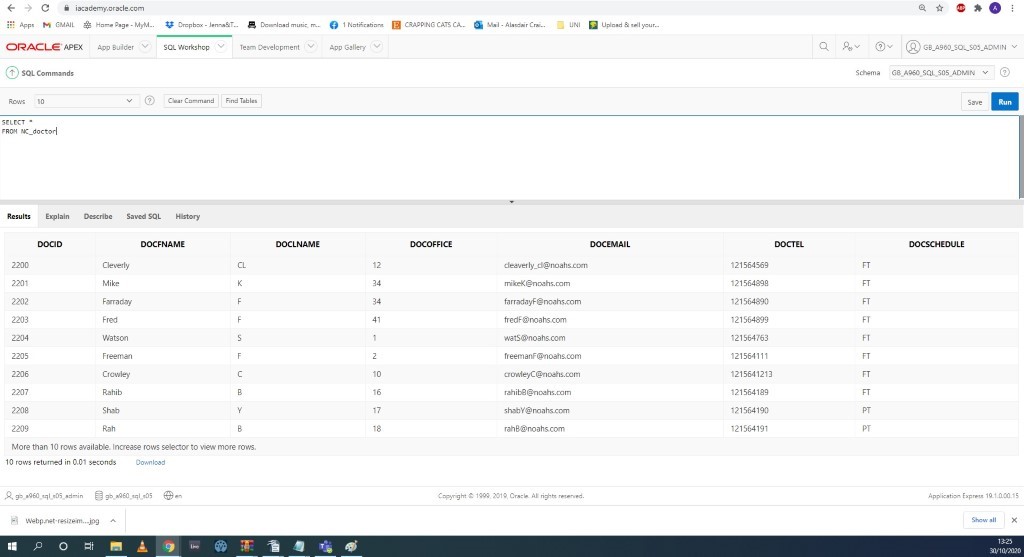
Appointments



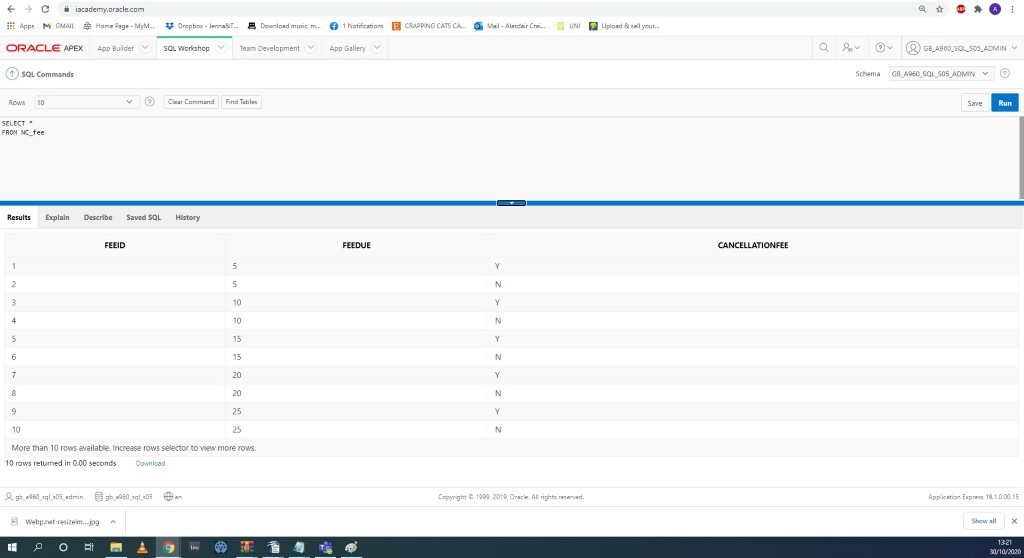
Diagnosis



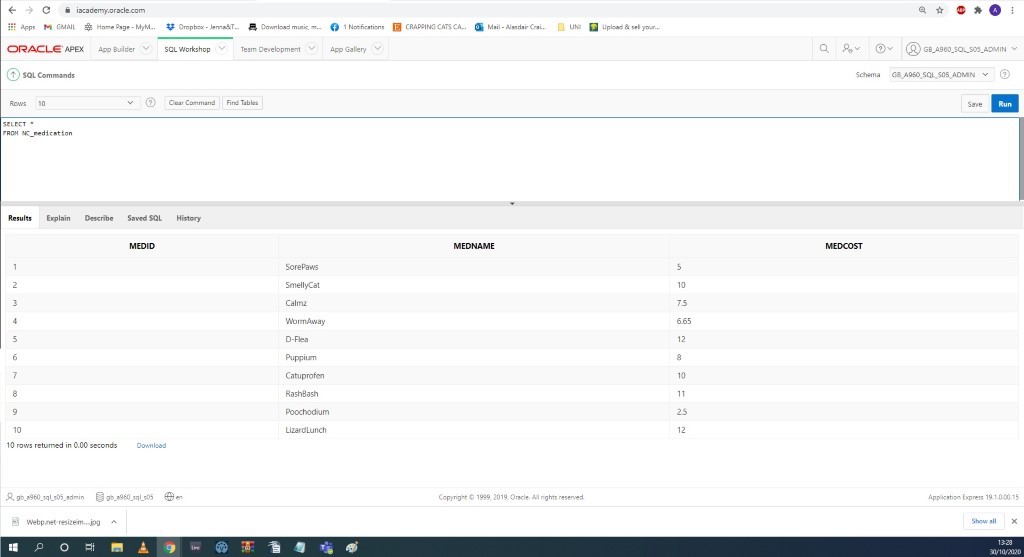
Doctors



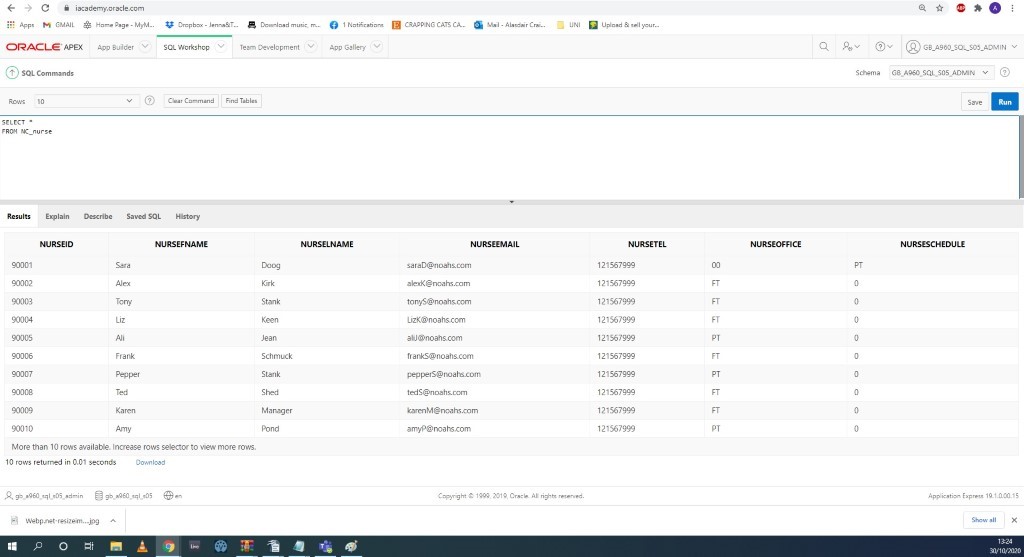
Fees



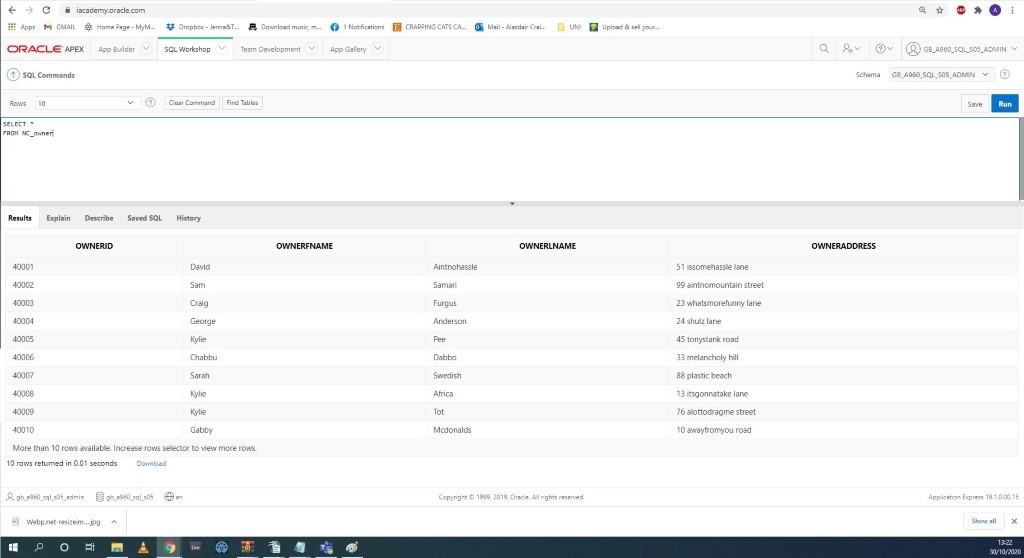
Medication



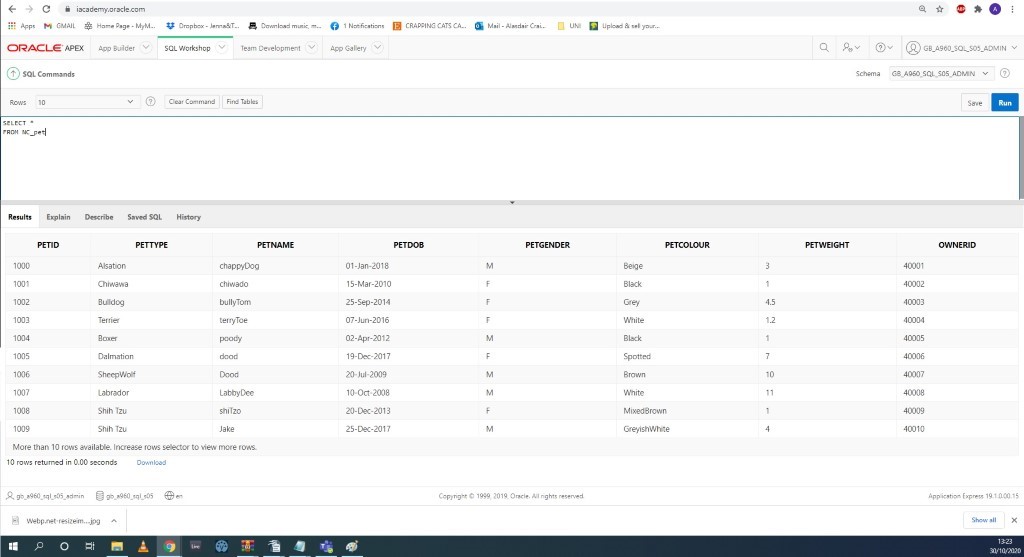
Nurses



Owners



Pets

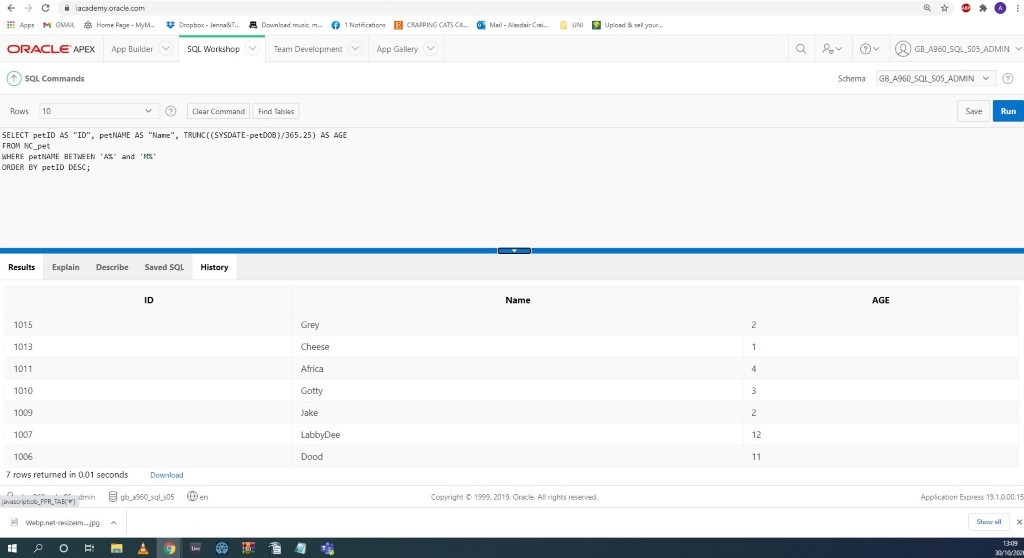


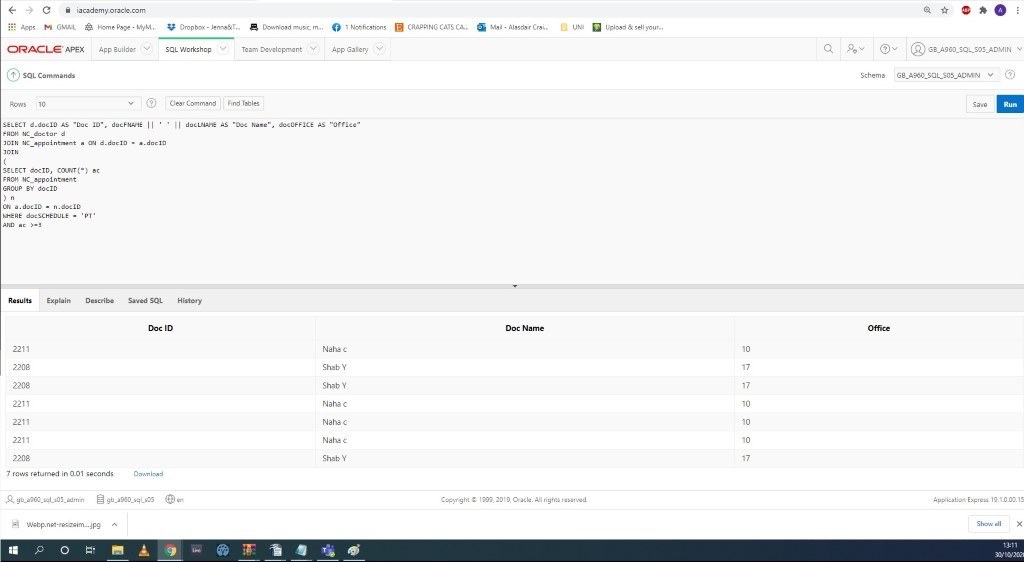
Database improvements

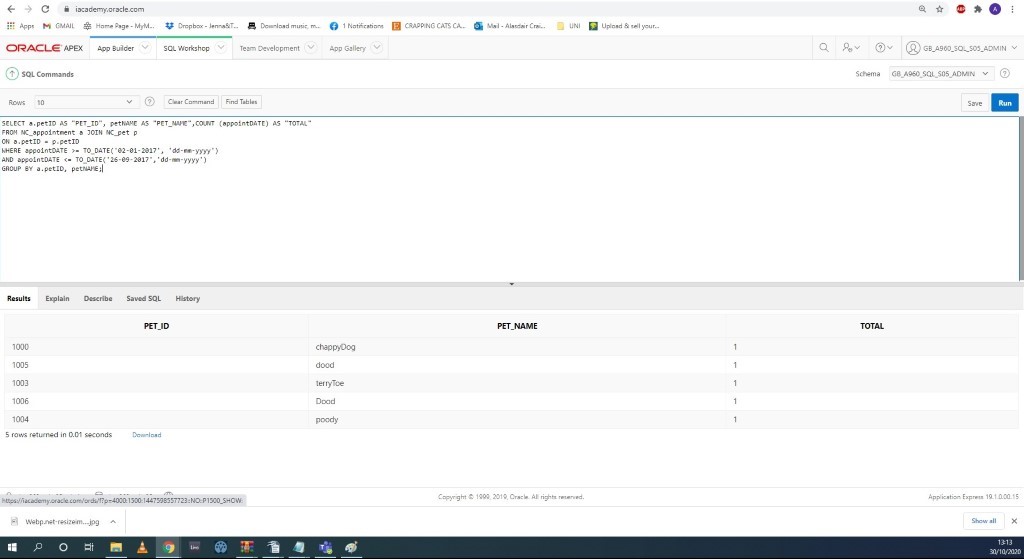
* One area for improvement would be to link the Fee Table with Owner Table as well Appointment Table giving easier retrieval of information that the pet owner would require, however this did not fit with our normalisation model.
* Further still, the Fee Table might not be necessary if a function was added that automatically applied the correct appointment cost, based on the pet’s age directly to the appoint.
* To make the system suitable for a multi-user environment a transaction lock would be used as it locks a portion of the database (a table or a section) to prevent other users from making any changes until the user is finished or rolled back. This prevents problems when two users try to modify the same data at the same time.
* Another improvement would be to add security features to the system such as assigning different permissions for each user. This will help to eliminate the risk of users manipulating with the wrong data within the database.
* Account separation might be implemented with separate account access and database storage for each user.
* For the system to be more multi-user friendly, the system may notify other relevant users if changes have been made to the database by another user.

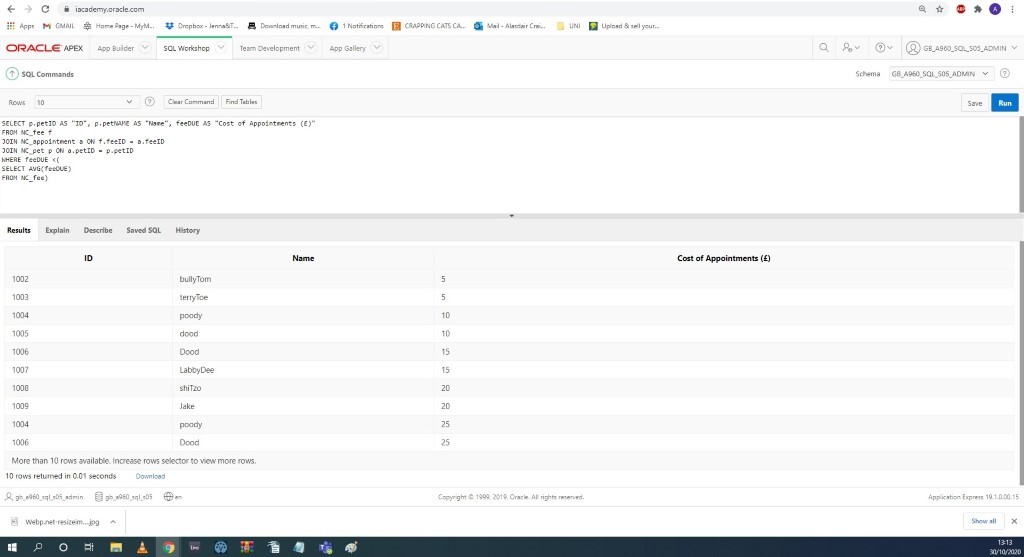
Query Implementation

SQL scripts in Oracle

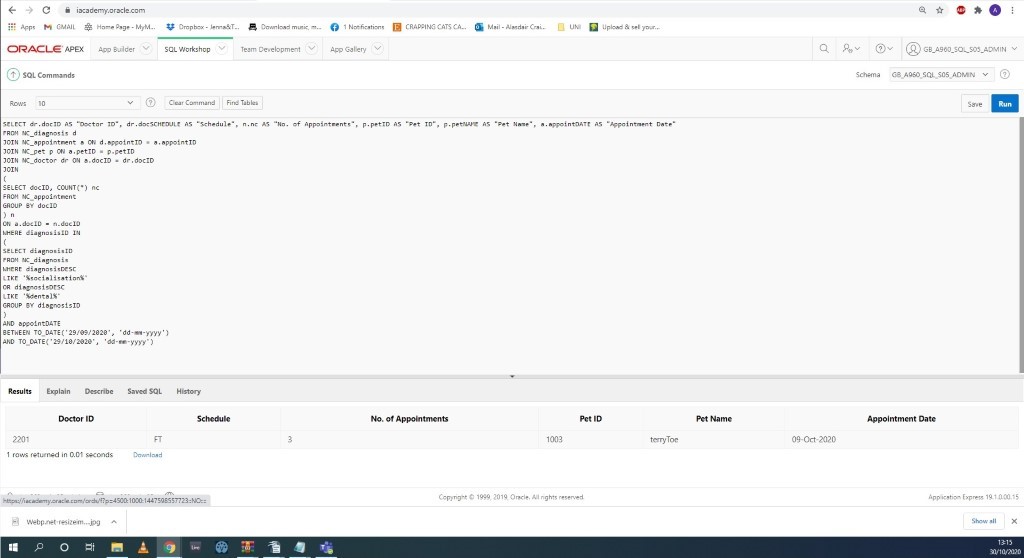
1. 

2. 

3. 

4. 

5.



6. 