The University of the West Indies,
Cave hill Campus

Project A

Group Assignment

COMP3330 – Database Management Systems I

Mrs. Tessa King Inniss

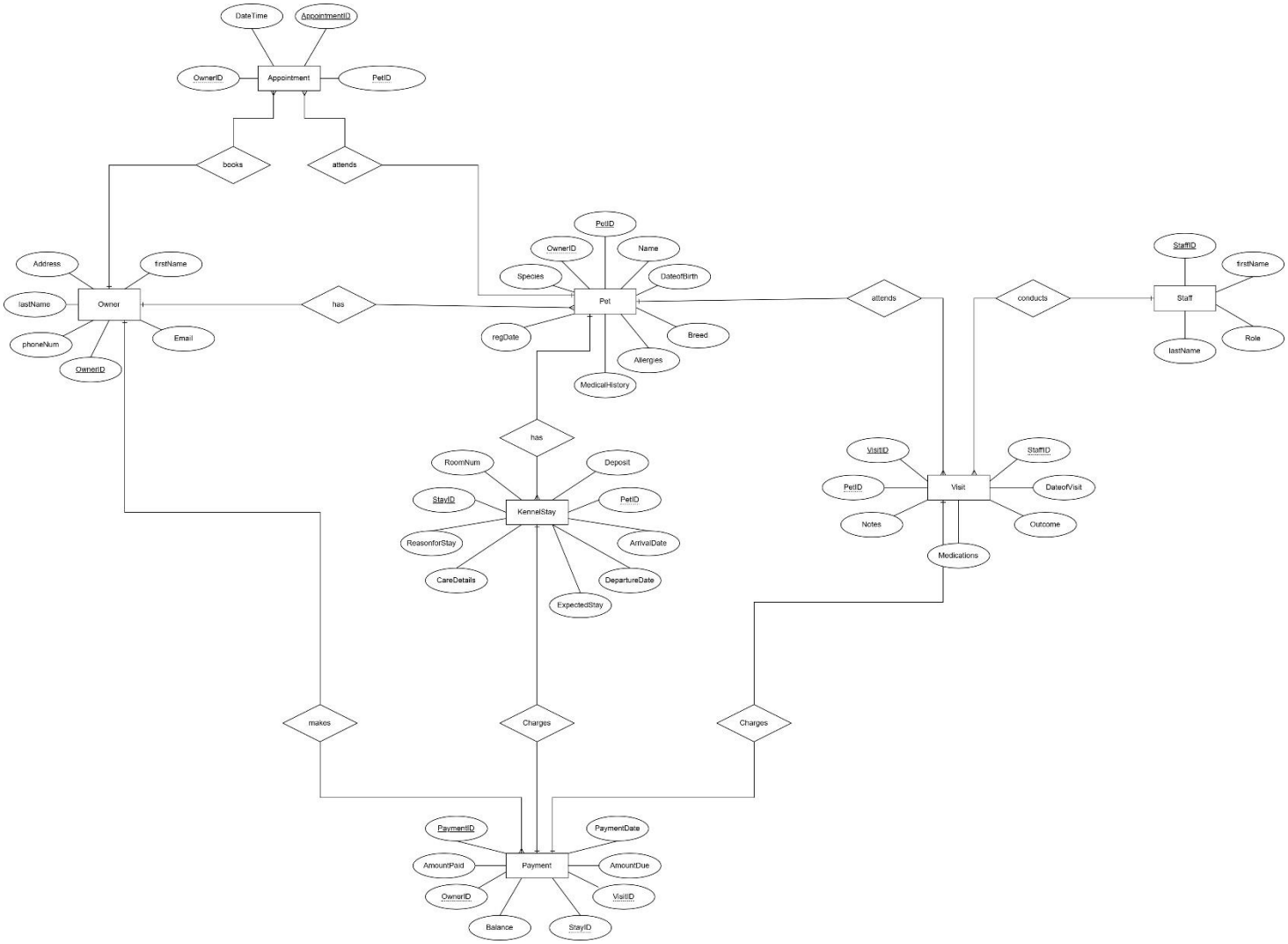
Group:
Aren Brathwaite
Shane Grannum
Faith Waithe
Matthew Walker

OCTOBER 30, 2024

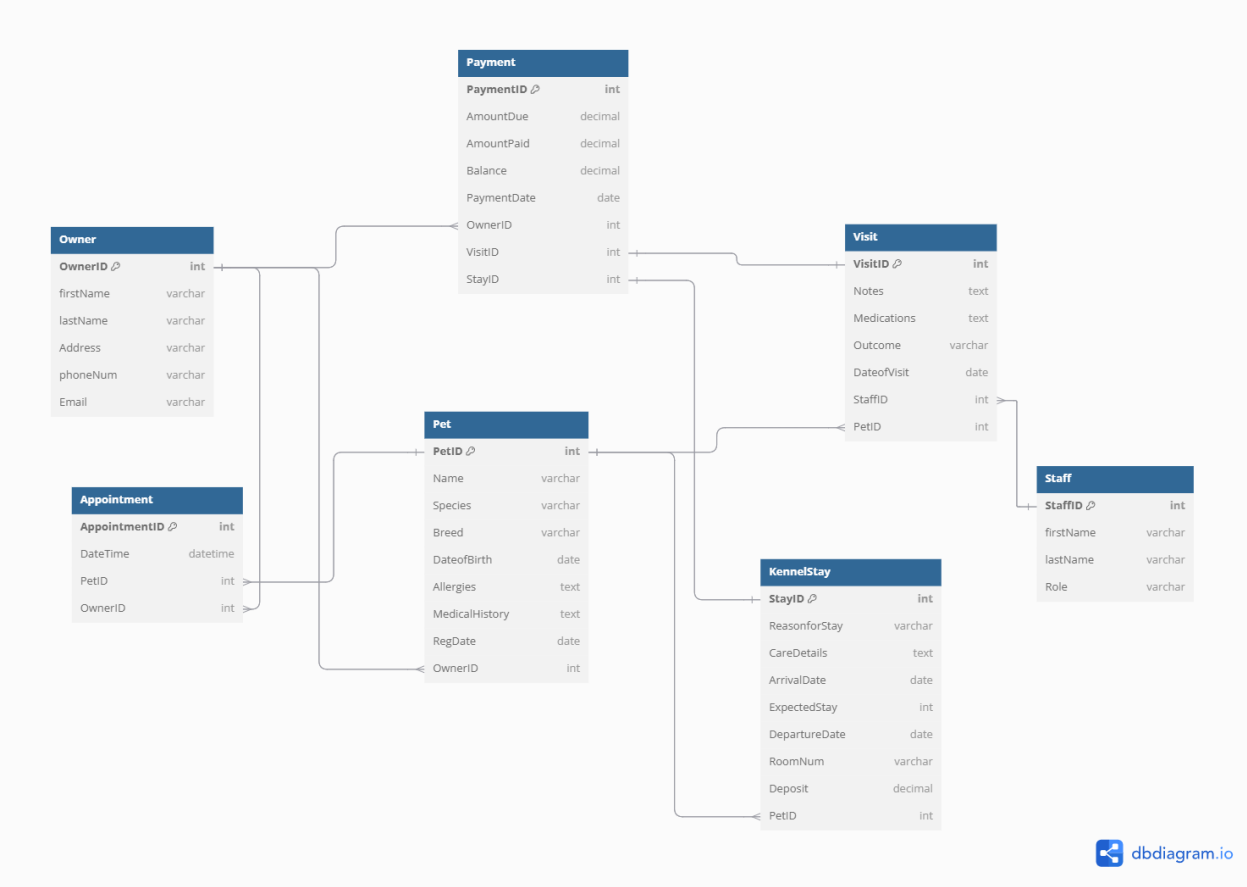
Table of Contents

Conceptual Schema

ER Diagram:



Relational Schema



Owner (OwnerID, firstName, lastName, Address, phoneNum, Email)
Pet (PetID, Name, Species, Breed, DateOfBirth, Allergies, MedicalHistory, regDate, OwnerID)
Staff (StaffID, firstName, lastName, Role)
Visit (VisitID, Notes, Medications, Outcome, DateOfVisit, StaffID, PetID)
KennelStay (StayID, ReasonforStay, CareDetails, ArrivalDate, ExpectedStay, DepartureDate, RoomNum, Deposit, PetID)
Payment (PaymentID, AmountDue, AmountPaid, Balance, PaymentDate, OwnerID, VisitID, StayID,)
Appointment (AppointmentID, DateTime, PetID, OwnerID)

Normalization

All tables are First Normal Form **(1NF)** compliant as they satisfy the following criteria:

- All columns contain only atomic (indivisible) values. This means no repeating groups or arrays in a single column.
- All columns have a unique name.
- The order in which data is stored does not matter.
- All entries in a column are of the same data type.

All tables are Second Normal Form **(2NF)** compliant as they satisfy the following criteria:

- First Normal Form (1NF): The table is in 1NF.
- No Partial Dependencies exist: All non-key attributes are fully dependent on the primary key.

All tables are Third Normal Form **(3NF)** compliant as they satisfy the following criteria:

- They already satisfy all the 1NF and 2NF criteria.
- No Transitive Dependencies: Non-key attributes do not depend on other non-key attributes. Every non-key attribute depends only on the primary key.

Data Dictionary

Table Name	Attribute Name	Contents	Type	Length	Key (PK or FK)
Owner	OwnerID	Owner ID Number	INT		PK
	firstName	Owner First Name	VARCHAR	50	
	lastName	Owner Last Name	VARCHAR	50	
	Address	Owner Address	VARCHAR	255	
	phoneNum	Owner Phone Number	VARCHAR	15	
	Email	Owner Email Address	VARCHAR	100	
Pet	PetID	Pet ID Number	INT		PK
	Name	Pet Name	VARCHAR	50	
	Species	Pet species	VARCHAR	30	
	Breed	Pet Breed	VARCHAR	50	
	DateofBirth	Pet Date of Birth	DATE		
	Allergies	Pet Allergies	TEXT		
	MedicalHistory	Pet Medical History	TEXT		
	RegDate	Pet Registration Date	DATE		
	OwnerID	Owner ID Number	INT		FK
Staff	StaffID	Staff ID Number	INT		PK
	firstName	Staff First Name	VARCHAR	50	
	lastName	Staff Last Name	VARCHAR	50	
	Role	Staff role or rank	VARCHAR	50	
Visit	VisitID	Visit ID Number	INT		PK
	Notes	Veterinary notes	TEXT		
	Medications	Medications administered during visit	TEXT		
	Outcome	Visit Outcome	VARCHAR	100	
	DateofVisit	Date of Visit	DATE		
	StaffID	Staff ID Number	INT		FK
	PetID	Pet ID Number	INT		FK

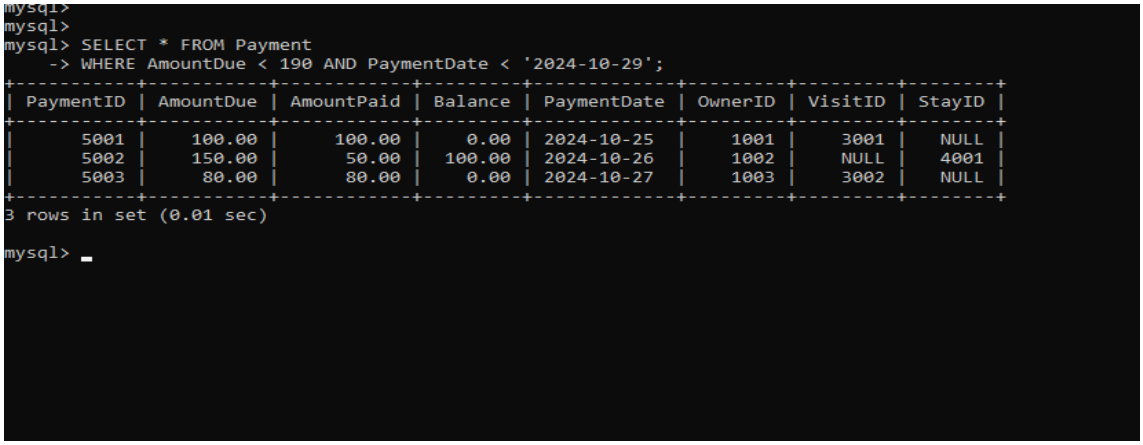
Table Name	Attribute Name	Contents	Type	Length	Key (PK or FK)
KennelStay	StayID	Stay ID Number	INT		PK
	ReasonforStay	Reason for pet's stay	VARCHAR	255	
	CareDetails	Special Arrangements for pet care	TEXT		
	DepartureDate	Date of pet's Departure	DATE		
	ExpectedStay	The expected length of Stay in days	INT		
	ArrivalDate	Date of pet's Arrival	DATE		
	RoomNum	Pet's Room Number	VARCHAR	10	
	Deposit	Deposit payment for stay	DECIMAL	10 , 2	
	PetID	Pet ID Number	INT		FK
Payment	PaymentID	Payment ID Number	INT		PK
	AmountDue	Amount Due	DECIMAL	10 , 2	
	AmountPaid	Amount Paid	DECIMAL	10 , 2	
	Balance	Remaining Balance	DECIMAL	10 , 2	
	PaymentDate	Date payment was made	DATE		
	OwnerID	Owner ID Number	INT		FK
	VisitID	Visit ID Number	INT		FK
	StayID	Stay ID Number	INT		FK
Appointment	AppointmentID	Appointment ID Number	INT		PK
	DateTime	Appointment Date and Time	DATETIME		
	OwnerID	Owner ID Number	INT		FK
	PetID	Pet ID Number	INT		FK

MySQL Queries

Single Table Queries:

i)

*SELECT * FROM Payment*
WHERE AmountDue < 190 AND PaymentDate < '2024-10-29';



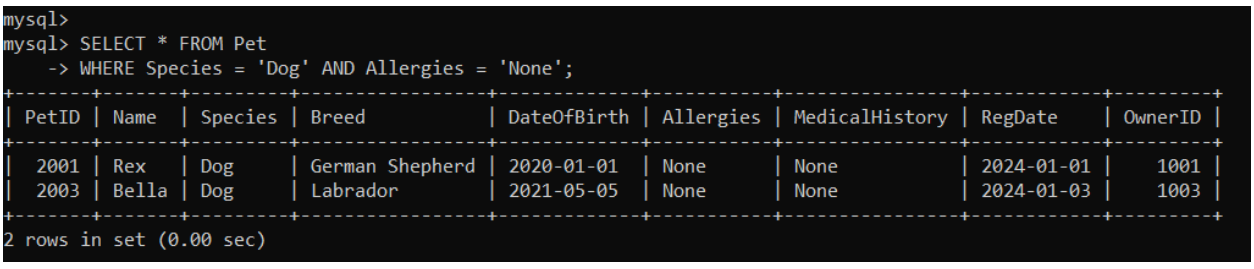
```
mysql>
mysql>
mysql> SELECT * FROM Payment
    -> WHERE AmountDue < 190 AND PaymentDate < '2024-10-29';
+-----------+-----------+------------+---------+-------------+---------+---------+--------+
| PaymentID | AmountDue | AmountPaid | Balance | PaymentDate | OwnerID | VisitID | StayID |
+-----------+-----------+------------+---------+-------------+---------+---------+--------+
|      5001 |    100.00 |     100.00 |    0.00 | 2024-10-25  |    1001 |    3001 |   NULL |
|      5002 |    150.00 |      50.00 |  100.00 | 2024-10-26  |    1002 |    NULL |   4001 |
|      5003 |     80.00 |      80.00 |    0.00 | 2024-10-27  |    1003 |    3002 |   NULL |
+-----------+-----------+------------+---------+-------------+---------+---------+--------+
3 rows in set (0.01 sec)

mysql>
```

English: Retrieve all records from the Payment table where the AmountDue is less than 190 and PaymentDate is before ‘2024-10-29’.

ii)

*SELECT * FROM Pet*
WHERE Species = 'Dog' AND Allergies = 'None';



```
mysql>
mysql> SELECT * FROM Pet
    -> WHERE Species = 'Dog' AND Allergies = 'None';
+-------+-------+---------+-----------------+-------------+-----------+----------------+------------+---------+
| PetID | Name  | Species | Breed           | DateOfBirth | Allergies | MedicalHistory | RegDate    | OwnerID |
+-------+-------+---------+-----------------+-------------+-----------+----------------+------------+---------+
|  2001 | Rex   | Dog     | German Shepherd | 2020-01-01  | None      | None           | 2024-01-01 |    1001 |
|  2003 | Bella | Dog     | Labrador        | 2021-05-05  | None      | None           | 2024-01-03 |    1003 |
+-------+-------+---------+-----------------+-------------+-----------+----------------+------------+---------+
2 rows in set (0.00 sec)
```

English: Retrieve all records from the Pet table where the species is "Dog" and where allergies are “None”.

Two Table Queries:

i)

SELECT Visit., Pet.Name FROM Visit*
JOIN Pet ON Visit.PetID = Pet.PetID
WHERE Outcome = 'Healthy';

```
mysql>
mysql> SELECT visit.*, pet.name FROM visit
    -> JOIN pet on visit.petid = pet.petid
    -> where outcome = 'Healthy';
+---------+----------+-------------+---------+-------------+---------+-------+------+
| VisitID | Notes    | Medications | Outcome | DateOfVisit | StaffID | PetID | name |
+---------+----------+-------------+---------+-------------+---------+-------+------+
|    3001 | Check-up | None        | Healthy | 2024-10-25  |    9000 |  2001 | Rex  |
|    3005 | Check-up | None        | Healthy | 2024-10-29  |    9002 |  2005 | Luna |
+---------+----------+-------------+---------+-------------+---------+-------+------+
2 rows in set (0.00 sec)

mysql>
```

English: Retrieve all the all records from the Visit table and the name of the corresponding pet for visits where the outcome was “Healthy”.

ii)

SELECT Pet.Name, Owner.firstName, Owner.lastName FROM Pet
JOIN Owner ON Owner.OwnerID = Pet.OwnerID
WHERE MedicalHistory != 'None';

```
mysql>
mysql> SELECT pet.name,owner.firstname,owner.lastname FROM pet
    -> JOIN owner on owner.ownerid = pet.ownerid
    -> WHERE MedicalHistory != 'None';
+----------+-----------+----------+
| name     | firstname | lastname |
+----------+-----------+----------+
| Whiskers | Jane      | Smith    |
| Max      | Emily     | Davis    |
+----------+-----------+----------+
2 rows in set (0.00 sec)

mysql>
```

English: Retrieve the first and last name of owners along with their pet’s names where Medical History is not “None”.

Group Reflection

The system used by the Haven Veterinary Clinic & Kennel was a traditional file system, which involved maintaining records with physical paper, folders, and electronic spreadsheets. Some issues they most likely experienced were caused by:

- Data inconsistencies due to the lack of data dependence which would result in data redundancies. Hence, some records may contain contradicting and incorrect data that may not have been up to date.
- Since a paper-based system has to be stored in physical locations, these storage areas are more susceptible to privacy/security risks and physical attacks. Additionally, in the event of natural hazards/disasters, paperwork can be damaged or destroyed.
- The time consumption and difficulty taken to perform unusual/ad hoc queries.
- As the number of files increases, the difficulties of the system administration become a problem. Multiple file management programs need to be maintained which allow users to add, update, delete, and view records as well as generate reports.

Switching to a modern database system instead of relying on the traditional file system approach offers The Haven Veterinary Clinic & Kennel substantial benefits, such as:

- Easier management of operations for when the business starts to grow and experience an increase in volume of data and records become increasingly large.
- Ability to perform ad-hoc queries on the database when time is limited.
- Reduction of data redundancies in the system as data duplication is eliminated and in the event of redundancy, it is controlled easily.
- Ability to update the structure of a record with minimal adjustment to existing programs and files. This is called structural independence.
- Ability of the admin users to alter characteristics of the data without much impact caused to applications which access the data. This is called data independence.
- Reduction in program maintenance as the data and structure of the records/files can be altered with minimal impact on the existing system.
- Data sharing and physical access to data are improved and made easier for employees to access.

- Low likelihood of privacy/security issues and proper management in the accessing of data.

Individual Reflections

Shane Grannum:

What I found to be the most challenging part of the design and implementation process would have to be the conceptual design and creation of the ERD. This stage was critical as it formed the foundation of the database. We frequently revisited this phase to ensure accurate relationships and to avoid overlooking any details. On the other hand, what I found to be the least challenging part was the implementation itself. Having a well-defined ERD, defining table structures and inserting data became very straightforward and quite effortless. The solid groundwork laid during the conceptual design phase streamlined the implementation process making it smooth and efficient.

Matthew Walker:

The most challenging part of the design and implementation process was determining which tables needed normalization. Creating scenarios for sample data was sometimes confusing, leading to overthinking and delays. Understanding how to apply normalization correctly requires extra attention. On the other hand, the easiest part of the process was creating the data dictionary. The outline provided in the notes made it straightforward and helped guide the process. Overall, while normalizing the tables posed difficulties, the clear structure of the data dictionary was a helpful and efficient tool for organizing the database information effectively.

Faith Waithe:

The most challenging part of the design and implementation process was the creation of the ERDs and the normalization aspect. Identifying which tables needed to be normalized was a bit difficult as at first, I could not fully grasp the concept of normalization. To combat this challenge, we kept refining the ERDs as much as possible to lessen the amount of normalizing that needed to be done. The least challenging part of the design and implementation process was the data dictionary. The development of the data dictionary was made easier due to the ERDs, and relational schema already having been created.

Aren Brathwaite:

The most challenging part of the design and implementation process was determining the type of relationships between entities and the normalization of the tables. Determining if the relationship was one-to-one, one-to-many or many-to-many; this was particularly difficult for me because I had problems grasping if we were looking at the relationships in their totality or at a particular instance. Some revision of the class notes as well as online research was done to help better grasp this concept. The least challenging part of the process was the relational schema and the data dictionary. This was made easy because of how well we did our ERD.