INFSCI 1500 Group 4

Final Project

By: Balazs Kudelasz, Isha Trasi, Kaden Jackson, Lucas Weiland, Rin Terna

INFSCI 1500

Professor Mark Weixel

4/22/2024

# Table of Contents

**Introduction/Abstract - 3**

**Business Rules – 4 - 8**

**Entity Descriptions – 9 - 17**

**List of Created Questions – 18 - 24**

## Introduction/Abstract

Our group decided to make our database project for a veterinarian's office. The day-to-day operations of a veterinarian’s office are fairly complex, involving a mix of administrative tasks alongside their main focus of animal care and support. From coordinating appointments and procedures to managing patient records and invoices, the routine functions of a vet's office can be both complex and time-consuming. Given the demands of their daily work, our group decided to make a database that could be helpful to simplify their tasks and boost the office’s operational efficiency.

The database includes entities for all of the different aspects of a veterinarian’s office and could be used on a daily basis to help organize and automate the management of different operations within the office. The main goal of our database is to simplify the administrative tasks that veterinarians and other staff members face daily and so they can focus on improving the care that they provide for animals. The database is primarily designed for veterinarians and other essential staff who work in the office and play a part in the work done there. It will make things easier by facilitating tasks such as appointment scheduling and tracking, managing invoices and allowing easy access to information like patient records, procedure history, billing details and more.

## Business Rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity 1** | **Entity 2** | **Cardinality on Entity 1 Side** | **Cardinality on Entity 2 Side** | **Business Rule(s)** |
| customers | patients | 1..\* | 1..\* | A Customer can own many patients.  Many patients can be owned by one customer. |
| customers | addresses | 1 | 1..\* | A customer has 1 address.  Each address can be associated with one or many customers. |
| staff | addresses | 1 | 1..\* | A staff member has 1 address.  Each address can be associated with one or many staff members. |
| doctors | offices | 1..\* | 1..\* | Each doctor can be associated with 1 or more offices. Each office is assigned many doctors. |
| staff | offices | 1..\* | 1..\* | Each staff member can be associated with 1 or more offices. Each offices is assigned many different staff members |
| visits | procedures | 1..\* | 1..\* | Each visit can have one or more procedures. Many procedures can be done at many visits. |
| invoices | visits | 1..\* | 1 | An invoice can be associated to one or many visits. Each visit can have 1 invoice. |
| invoices | discounts | 0..\* | 0..\* | An invoice may have 0 or more discounts applied to it. A discount can be applied to many invoices. |
| invoices | procedures | 1..\* | 0..\* | Each invoice can have one or more associated procedures. A procedure may or may not be associated with an invoice. |
| invoices | patients | 1..\* | 1..\* | An invoice can be linked to one or more patients. Each patient can have many invoices. |
| invoices | vaccinations | 1..\* | 1..\* | An invoice can list one or many vaccinations. A vaccination can be listed on many invoices. |
| doctors | addresses | 1..\* | 1..\* | Each doctor has 1 or many addresses they work at. Each address is associated with many doctors. |
| visits | meds | 0..\* | 0..\* | Each visit could have many medications used. Each medication could be used at many visits. |
| invoices | customers | 1 | 1..\* | Each invoice is given to 1 customer. A customer can receive 1 or many invoices. |
| invoices | offices | 1 | 1..\* | An invoice is done at only 1 office. Each office can have 1 or many invoices attached to them. |
| offices | addresses | 1 | 1 | Each office has 1 address and every address is associated with 1 office. |
| vaccinations | vaccines | 1..\* | 1 | Each vaccination appointment uses one vaccine. Each vaccine can be used at one or many vaccination appointments. |
| meds | meds\_strength | 1 | 0..\* | Each med has 1 strength given to it. The strength can be the same for many types of medicine. |
| doctors | visits | 0..\* | 1 | Each doctor can have 0 or more visits assigned to them. Each visit is linked to 1 doctor. |
| visits | vaccinations | 0..\* | 1 | Each visit can involve 0 or more vaccination appointments. Each vaccination appointment is linked to exactly 1 visit. |
| visit | visit\_types | 1 | 1..\* | Each visit can have 1 visit type defined in the database. Each visit type can be given to many visits. |
| patient | patient\_types | 1 | 0..\* | Each patient can only 1 have type associated to them. The patient type can be given to 0 or many patients. |
| patients | vaccinations | 0..\* | 1 | Each patient can be the subject of up to many vaccination appointments. Each vaccination appointment is linked to one patient. |
| patients | visits | 1..\* | 1 | Each patient can have 1 or many visits at the office. Each visit is associated to only one patient. |

## Entity Descriptions

Table Name: patients

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| patient\_id | INT | NOT NULL | Primary Key |
| patient\_name | VARCHAR | NOT NULL |  |
| type\_id | INT | NOT NULL | Foreign Key (References patient\_types(type\_id)) |
| patient\_dob | DATE | NOT NULL |  |
| customer\_id | INT | NOT NULL |  |
| microchip\_number | INT |  |  |
| alteration | BOOLEAN |  |  |
| health\_issues | VARCHAR |  |  |
| fractious | VARCHAR |  |  |

Table Name: patient\_types

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| type\_id | INT | NOT NULL | Primary Key |
| type\_name | VARCHAR | NOT NULL |  |

Table Name: customers

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| customer\_id | INT | NOT NULL | Primary Key |
| first\_name | VARCHAR | NOT NULL |  |
| last\_name | VARCHAR | NOT NULL |  |

Table Name: doctors

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| doctor\_id | INT | NOT NULL | Primary Key |
| doctor\_first\_name | VARCHAR | NOT NULL |  |
| doctor\_last\_name | VARCHAR | NOT NULL |  |
| vet\_license\_number | VARCHAR | NOT NULL |  |
| dea\_license\_number | VARCHAR |  |  |
| usda\_license\_number | VARCHAR |  |  |
| office\_id | INT | NOT NULL | Foreign Key (References offices(office\_id)) |

Table Name: staff

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| staff\_id | INT | NOT NULL | Primary Key |
| staff\_first\_name | VARCHAR | NOT NULL |  |
| staff\_last\_name | VARCHAR | NOT NULL |  |
| tech\_license\_number | VARCHAR |  |  |

Table Name: offices

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| office\_id | INT | NOT NULL | Primary Key |
| office\_name | VARCHAR | NOT NULL |  |
| address\_id | INT | NOT NULL | Foreign Key (References addresses(address\_id)) |
| tax\_rate | FLOAT | NOT NULL |  |

Table Name: addresses

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| address\_id | INT | NOT NULL | Primary Key |
| street | VARCHAR | NOT NULL |  |
| city | VARCHAR | NOT NULL |  |
| state | VARCHAR | NOT NULL |  |
| zip | VARCHAR | NOT NULL |  |

Table Name: visits

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| visit\_id | INT | NOT NULL | Primary Key |
| visit\_type\_id | INT | NOT NULL | Foreign Key (References visit\_types(visit\_type\_id)) |
| visit\_date | DATE | NOT NULL |  |
| patient\_id | INT | NOT NULL | Foreign Key (References patients(patient\_id)) |
| doctor\_id | INT | NOT NULL | Foreign Key (References doctors(doctor\_id)) |
| vaccination\_id | INT | NOT NULL | Foreign Key (References vaccinations(vaccination\_id)) |
| weight | FLOAT | NOT NULL |  |
| dental | VARCHAR |  |  |
| eyes | VARCHAR |  |  |
| notes | VARCHAR |  |  |

Table Name: visit\_types

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| visit\_type\_id | INT | NOT NULL | Primary Key |
| visit\_type\_name | VARCHAR | NOT NULL |  |
| visit\_cost | DECIMAL | NOT NULL |  |

Table Name: procedures

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| procedure\_id | INT | NOT NULL | Primary Key |
| procedure\_name | VARCHAR | NOT NULL |  |
| procedure\_cost | DECIMAL | NOT NULL |  |

Table Name: vaccinations

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| vaccination\_id | INT | NOT NULL | Primary Key |
| vaccine\_id | INT | NOT NULL | Foreign Key (References vaccines(vaccine\_id)) |
| vaccination\_date | DATE | NOT NULL |  |
| patient\_id | INT | NOT NULL | Foreign Key (References patients(patient\_id)) |

Table Name: vaccines

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| vaccine\_id | INT | NOT NULL | Primary Key |
| vaccine\_name | VARCHAR | NOT NULL |  |
| vaccine\_cost | DECIMAL | NOT NULL |  |

Table Name: meds

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| meds\_id | INT | NOT NULL | Primary Key |
| meds\_name | VARCHAR | NOT NULL |  |
| meds\_strength\_id | INT | NOT NULL | Foreign Key (References meds\_strengths(meds\_strength\_id)) |
| meds\_cost | DECIMAL | NOT NULL |  |

Table Name:meds\_strengths

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| meds\_strength\_id | INT | NOT NULL | Primary Key |
| meds\_strength\_type | VARCHAR | NOT NULL |  |
| meds\_strength\_unit | VARCHAR | NOT NULL |  |

Table Name: discounts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| discount\_id | INT | NOT NULL | Primary Key |
| discount\_amount | FLOAT | NOT NULL |  |

Table Name: invoices

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Data Type** | **Restriction** | **Key** |
| invoice\_number | INT | NOT NULL | Primary Key |
| invoice\_date | DATE | NOT NULL |  |
| office\_id | INT | NOT NULL | Foreign Key (References offices(office\_id)) |
| customer\_id | INT | NOT NULL | Foreign Key (References customers(customer\_id)) |

## List of Created Questions

|  |  |  |
| --- | --- | --- |
| # | Question | Rational/Benefit |
| 1 | Which patients are registered at the veterinary office, and who are their owners? | Quick identification of the roster of all patients and their respective owners |
| 2 | What are the types of patients registered at the veterinary office, and who are their owners? | Overview of patient types and their associated owners |
| 3 | How many visits has each patient made to the veterinary office? | Track patient engagement and frequency of visits |
| 4 | What is the average cost of visits for each visit made to the veterinary office? | Overall financial analysis for the average cost of a visit |
| 5 | Which invoices have been issued by the veterinary office, and what are the details of each invoice? | Offers a detailed record of the invoices created by the veterinarian's office |
| 6 | What is the total cost of medication for each patient treated at the veterinary office? | Summary of the medication expenses per patient across all of their visits |
| 7 | Which patients at the veterinary office have health issues or exhibit fractious behavior? | Identifies patients that may require special attention or handling |
| 8 | What are the address details of the veterinary office? | Provide essential contact information for the veterinary offices |
| 9 | Who are the staff members working at the veterinary office, and which office do they belong to? | Helps in staffing management and office allocations |
| 10 | How many vaccinations have been administered for each vaccine type at the veterinary office? | Tracks the overall vaccination rates for different vaccine types |

### SQL Queries for each Question

|  |
| --- |
| -- 1.        Which patients are registered at the veterinary office, and who are their owners?  DROP VIEW IF EXISTS patient\_owner\_view;    CREATE VIEW patient\_owner\_view AS  SELECT      p.patient\_name AS Patient\_Name,      CONCAT(c.first\_name, ' ', c.last\_name) AS Owner\_Name  FROM      patients p  JOIN      customers\_patients cp ON p.patient\_id = cp.fk\_patient\_id  JOIN      customers c ON cp.fk\_customer\_id = c.customer\_id;    -- To run the query:    SELECT \* FROM patient\_owner\_view;  DROP VIEW IF EXISTS patient\_owner\_view |
| -- 2.        Which patient files have "dog" as the type?  DROP VIEW IF EXISTS patient\_type\_view;    CREATE VIEW patient\_type\_view AS  SELECT \* FROM patients p  WHERE p.type\_id IN (SELECT type\_id FROM patient\_types WHERE type\_name = 'dog');    -- To run the query:    SELECT \* FROM patient\_type\_view;  DROP VIEW IF EXISTS patient\_type\_view |
| -- 3.        How many visits has each patient made to the veterinary office?  DROP VIEW IF EXISTS patient\_visit\_office;    CREATE VIEW patient\_visit\_office AS  SELECT                  p.patient\_name AS Patient\_Name,      CONCAT(c.first\_name, ' ', c.last\_name) AS Owner\_Name,      COUNT(v.visit\_id) AS Number\_of\_Visits  FROM      patients p  JOIN      visits v ON p.patient\_id = v.patient\_id  JOIN      customers\_patients cp ON p.patient\_id = cp.fk\_patient\_id  JOIN      customers c ON cp.fk\_customer\_id = c.customer\_id  GROUP BY p.patient\_name,      CONCAT(c.first\_name, ' ', c.last\_name);    -- To run the query:    SELECT \* FROM patient\_visit\_office;  DROP VIEW IF EXISTS patient\_visit\_office |
| -- 4.        What is the average visit cost for the visit with the ID of 2?    CREATE VIEW average\_visit\_cost\_view AS  SELECT      v.visit\_id AS Visit\_ID,      AVG(vt.visit\_cost + COALESCE(procedures\_total.procedure\_total\_cost, 0) + COALESCE(vaccines\_total.vaccine\_total\_cost, 0)) AS Average\_Visit\_Cost  FROM      visits v  JOIN      visit\_types vt ON v.visit\_type\_id = vt.visit\_type\_id  LEFT JOIN      (          SELECT              vp.fk\_visit\_id,              SUM(p.procedure\_cost) AS procedure\_total\_cost          FROM              visits\_procedures vp          LEFT JOIN              procedures p ON vp.fk\_procedure\_id = p.procedure\_id          GROUP BY              vp.fk\_visit\_id      ) AS procedures\_total ON v.visit\_id = procedures\_total.fk\_visit\_id  LEFT JOIN      (          SELECT              v.visit\_id,              SUM(vc.vaccine\_cost) AS vaccine\_total\_cost          FROM              visits v          LEFT JOIN              vaccines vc ON v.vaccination\_id = vc.vaccine\_id          GROUP BY              v.visit\_id      ) AS vaccines\_total ON v.visit\_id = vaccines\_total.visit\_id  GROUP BY      v.visit\_id  HAVING                  v.visit\_id = '2';    -- To run the query:    SELECT \* FROM average\_visit\_cost\_view;  DROP VIEW IF EXISTS average\_visit\_cost\_view; |
| -- 5.        Which invoices have been issued by the veterinary office, and what are the details of each invoice?    CREATE VIEW invoice\_details\_view AS  SELECT      i.invoice\_number AS Invoice\_Number,      i.invoice\_date AS Invoice\_Date,      o.office\_name AS Office\_Name,      c.first\_name AS Customer\_First\_Name,      c.last\_name AS Customer\_Last\_Name  FROM      invoices i  INNER JOIN      customers c ON i.customer\_id = c.customer\_id  INNER JOIN      offices o ON i.office\_id = o.office\_id;    -- To run the query:    SELECT \* FROM invoice\_details\_view;  DROP VIEW IF EXISTS invoice\_details\_view; |
| -- 6.        What is the highest total cost of medication for a patient treated at the veterinary office?    CREATE VIEW patient\_medication\_cost\_view AS  SELECT      p.patient\_name AS Patient\_Name,      SUM(m.meds\_cost) AS Total\_Medication\_Cost  FROM      patients p  INNER JOIN      visits v ON p.patient\_id = v.patient\_id  INNER JOIN      visits\_meds vm ON v.visit\_id = vm.fk\_visit\_id  INNER JOIN      meds m ON vm.fk\_meds\_id = m.meds\_id  GROUP BY      p.patient\_name  ORDER BY                  SUM(m.meds\_cost)  LIMIT 1;    -- To run the query:    SELECT \* FROM patient\_medication\_cost\_view;  DROP VIEW IF EXISTS patient\_medication\_cost\_view; |
| -- 7.        Which patients at the veterinary office have health issues or exhibit fractious behavior?    CREATE VIEW patient\_health\_issues\_view AS  SELECT      patient\_name AS Patient\_Name,      health\_issues AS Health\_Issues,                  fractious  AS Fractious\_Behavior  FROM      patients  WHERE      health\_issues IS NOT NULL OR      fractious IS NOT NULL;    -- To run the query:    SELECT \* FROM patient\_health\_issues\_view;  DROP VIEW IF EXISTS patient\_health\_issues\_view; |
| -- 8.        What are the address details of the veterinary office?    CREATE VIEW office\_address\_details\_view AS  SELECT      o.office\_name AS Office\_Name,      a.street AS Street,      a.city AS City,      a.state AS State,      a.zip AS Zipcode  FROM      offices o  JOIN      addresses a ON o.address\_id = a.address\_id;    -- To run the query:    SELECT \* FROM office\_address\_details\_view;  DROP VIEW IF EXISTS office\_address\_details\_view; |
| -- 9.        Who are the staff members working at the veterinary office in South Park?    CREATE VIEW staff\_office\_details\_view AS  SELECT      s.staff\_first\_name AS First\_Name,      s.staff\_last\_name AS Last\_Name,      o.office\_name AS Office\_Name  FROM      staff s  JOIN      staff\_offices so ON s.staff\_id = so.fk\_staff\_id  JOIN      offices o ON so.fk\_office\_id = o.office\_id;    -- To run the query:    SELECT \* FROM staff\_office\_details\_view;  DROP VIEW IF EXISTS staff\_office\_details\_view; |
| -- 10.     How many vaccinations have been administered for each vaccine type at the veterinary office?  CREATE VIEW vaccination\_counts\_view AS  SELECT      v.vaccine\_name AS Vaccine\_Type,      COUNT(\*) AS Total\_Vaccinations  FROM      vaccinations vac  JOIN      vaccines v ON vac.vaccine\_id = v.vaccine\_id  GROUP BY      v.vaccine\_name;    -- To run the query:    SELECT \* FROM vaccination\_counts\_view;  DROP VIEW IF EXISTS vaccination\_counts\_view; |

## 