

Report 2

CSC 775

Sequoia Andrade

Project Report 2

For the project I will be creating a simple database for a veterinary clinic that stores basic information about pets, owners, veterinarians, appointments, and vaccines.

Summary: For the final report, I created the 9 tables defined by the relational schema, requirements, and ER diagram developed in the first report. I manually created a small dataset and inserted values into the tables. I created an index on appointment dates, and a view showing the appointments for a specific veterinarian on a specified day.

Task1: Create Tables, indexes and constraints using DDL.

- At least one index (hash-based or tree-based) and one view.

```
CREATE TABLE Pets(  
  PID varchar(10),  
  first_name varchar(20),  
  last_name varchar(20),  
  gender varchar(6),  
  weight float,  
  age integer,  
  species varchar(20),  
  breed varchar(20),  
  PRIMARY KEY (PID));
```

```
CREATE TABLE Vets(  
  EID varchar(10),  
  first_name varchar(20),  
  last_name varchar(20),  
  address varchar(50),  
  phone_number char(10),  
  license_number char(10),  
  PRIMARY KEY (EID));
```

```
CREATE TABLE Owners(  
  OID varchar(10),  
  first_name varchar(20),  
  last_name varchar(20),  
  address varchar(50),  
  phone_number char(10),  
  PRIMARY KEY (OID));
```

```
CREATE TABLE Appointments(  
  AID varchar(15),  
  exam_room integer,
```

```
appt_date date,  
appt_time time,  
PRIMARY KEY (AID));
```

```
CREATE TABLE Vaccines(  
VID varchar(15),  
Vax_type varchar(30),  
PRIMARY KEY (VID));
```

```
CREATE TABLE Vet_Appointments(  
EID varchar(10),  
AID varchar(15),  
PRIMARY KEY (EID, AID),  
FOREIGN KEY (EID) REFERENCES Vets(EID)  
ON DELETE NO ACTION,  
FOREIGN KEY (AID) REFERENCES Appointments(AID)  
ON DELETE CASCADE);
```

```
CREATE TABLE Pet_Appointments(  
PID varchar(10),  
AID varchar(15),  
PRIMARY KEY (PID, AID),  
FOREIGN KEY (PID) REFERENCES Pets(PID)  
ON DELETE NO ACTION,  
FOREIGN KEY (AID) REFERENCES Appointments(AID)  
ON DELETE CASCADE);
```

```
CREATE TABLE Pet_Vaccines(  
PID varchar(10),  
VID varchar(15),  
Vax_date date,  
PRIMARY KEY (PID, VID),  
FOREIGN KEY (PID) REFERENCES Pets(PID)  
ON DELETE NO ACTION,  
FOREIGN KEY (VID) REFERENCES Vaccines(VID)  
ON DELETE NO ACTION);
```

```
CREATE TABLE Pet_Owners(  
PID varchar(10),  
OID varchar(10),  
PRIMARY KEY (PID, OID),  
FOREIGN KEY (PID) REFERENCES Pets(PID)  
ON DELETE NO ACTION,  
FOREIGN KEY (OID) REFERENCES Owners(OID)  
ON DELETE NO ACTION);
```

- Index:
CREATE INDEX ApptDateTime

ON Appointments (appt_date, appt_time)
USING BTREE;

- View:
CREATE VIEW current_owners_pets AS
SELECT P.PID, P.first_name AS pet_name, O.OID, O.first_name AS owner_name, O.last_name
FROM Owners O, Pets P, Pet_Owners PO
WHERE PO.PID=P.PID AND PO.OID=O.OID;

Task2: Collect and import data. You can collect data manually or import data from any available online data repository. There is no limit on the size of data.

- Pets:
INSERT INTO Pets (PID, first_name, last_name, gender, weight, age, species, breed) VALUES
(001, Trooper, Smith, Male, 10, 14, cat, DSH),
(002, Missy, Just, Female, 8, 15, cat, DLH),
(003, Betty, Lee, Female, 12, 8, cat, DSH),
(004, Stubina, Lou, Female, 11, 7, cat, DLH),
(005, Whitey, Jones, Male, 13, 12, cat, DSH),
(006, Fred, Jones, Male, 10, 13, cat, DSH),
(007, Meef, Andrade, Female, 13, 10, cat, DSH),
(008, Bambi, Andrade, Female, 10, 8, cat, DSH),
(009, Mushu, Lee, Male, 90, 3, dog, Malamute),
(010, Sox, Lee, Female, 15, 15, dog, Chihuahua),
(011, Hannah, Amaro, Female, 60, 6, dog, Mix),
(012, Dexter, Amaro, Male, 50, 6, dog, Mix),
(013, Rocksie, De Alba, Female, 40, 1, dog, Cattle Dog),
(014, Leaf, Erikson, Female, 0.2, 4, chameleon, veiled);
- Vets:
INSERT INTO Vets (EID, first_name, last_name, address, phone_number, license_number)
VALUES
(100, Sue, Gleeson, 505 pine ave, 8002785432, 1000000000),
(200, Gary, Russo, 321 field lane, 8003008756, 2000000000),
(300, Saming, Liu, 444 park ave, 8002009321, 3000000000);
- Owners:
INSERT INTO Owners (OID, first_name, last_name, address, phone_number) VALUES
(001, Sequoia, Andrade, 432 broadway, 3213214343),
(002, Devin, Lee, 555 ocean st, 3213215678),
(003, Jose, Amaro, 321 birch lane, 3213210009),
(004, Edith, Smith, 898 coyote rd, 3213214300),
(005, Paul, Just, 654 ocean st, 3213214523),
(006, Alissa, Lou, 121 broadway, 3213217777),
(007, Dave, Jones, 234 first st, 3213219467),
(008, Jess, De Alba, 999 second st, 3213213333);

- Appointments:
INSERT INTO Appointments(AID, exam_room, appt_date, appt_time) VALUES
(1, 1, '2022-10-31', '10:30')
(2, 1, '2022-10-31', '11:30')
(3, 1, '2022-10-31', '12:30')
(4, 1, '2022-10-31', '10:30')
(5, 2, '2022-10-31', '10:30')
(6, 3, '2022-10-31', '10:30')
(7, 1, '2022-10-30', '10:30');
- Vaccines:
INSERT INTO Vaccines(VID, vax_type) VALUES
(0001, rabies),
(0011, rabies),
(0002, leukemia),
(0003, distemper),
(0004, parvo);
- Vet_Appointments:
INSERT INTO Vet_Appointments(EID, AID) VALUES
(100, 1),
(100, 2),
(100, 3),
(200, 4),
(200, 5),
(300, 6),
(300, 7);
- Pet_Appointments:
INSERT INTO Pet_Appointments(AID, PID) VALUES
(1, 001),
(2, 002),
(3, 004),
(4, 007),
(5, 014),
(6, 003),
(7, 010);
- Pet_Vaccines:
INSERT INTO Pet_Vaccines(VID, PID, vax_date) VALUES
(0001, 001, '2021-08-12'),
(0002, 001, '2021-08-12'),
(0003, 001, '2021-08-12'),
(0001, 002, '2021-07-20'),
(0002, 002, '2021-07-20'),
(0003, 002, '2021-07-20'),
(0001, 003, '2020-08-12'),
(0001, 005, '2020-08-12'),
(0001, 006, '2019-06-02'),
(0001, 007, '2018-03-20'),

```
(0002, 007, '2018-03-20'),
(0003, 007, '2018-03-20'),
(0011, 009, '2020-04-13'),
(0003, 009, '2020-04-13'),
(0004, 009, '2020-04-13'),
(0011, 010, '2019-05-19'),
(0003, 010, '2019-05-19'),
(0004, 010, '2019-05-19'),
(0011, 011, '2021-10-20'),
(0011, 012, '2021-10-20');
```

- Pet_Owners:
INSERT INTO Pet_Owners(PID, OID) VALUES
(001, 004),
(002, 005),
(003, 002),
(004, 006),
(005, 007),
(006, 007),
(007, 001),
(008, 001),
(009, 002),
(010, 002),
(011, 003),
(012, 003),
(013, 008),
(014, 001);

Task3: Write SQL Queries

- At least 2 queries involving GROUP BY, HAVING with aggregate operators.
 1. Find the name of all owners with more than 1 cat.

```
SELECT O.first_name, O.last_name FROM Owners O, Pets P, Pet_Owners PO
WHERE (PO.OID = O.OID) AND (PO.PID = P.PID) AND (P.species = 'cat')
GROUP BY O.OID HAVING count(*)>1;
```

```
mysql> SELECT O.first_name, O.last_name FROM Owners O, Pets P, Pet_Owners PO
-> WHERE (PO.OID = O.OID) AND (PO.PID = P.PID) AND (P.species = 'cat')
-> GROUP BY O.OID HAVING count(*)>1;
```

first_name	last_name
Dave	Jones
Sequoia	Andrade

2. Find the owner and vet names of owners whose pets have only had appointments with one vet.

```

SELECT O.first_name, O.last_name, V.first_name, V.last_name from Owners O, Vets V, Pets P,
Vet_Appointments VA, Pet_Appointments PA, Pet_Owners PO
WHERE (PO.OID = O.OID) AND (PO.PID = P.PID) AND (PA.PID=P.PID) AND (VA.EID = V.EID) AND
(VA.AID = PA.AID)
GROUP BY O.OID HAVING count(V.EID=1);

```

```

mysql> SELECT O.first_name, O.last_name, V.first_name, V.last_name from Owners O,
Vets V, Pets P, Vet_Appointments VA, Pet_Appointments PA, Pet_Owners PO
-> WHERE (PO.OID = O.OID) AND (PO.PID = P.PID) AND (PA.PID=P.PID) AND (VA.EID
= V.EID) AND (VA.AID = PA.AID)
-> GROUP BY O.OID HAVING count(V.EID=1);
+-----+-----+-----+-----+
| first_name | last_name | first_name | last_name |
+-----+-----+-----+-----+
| Edith      | Smith    | Sue        | Gleeson   |
| Paul      | Just     | Sue        | Gleeson   |
| Alissa     | Lou      | Sue        | Gleeson   |
| Sequoia    | Andrade  | Gary       | Russo     |
| Devin     | Lee      | Saming     | Liu       |
+-----+-----+-----+-----+

```

- At least 2 nested queries involving IN, EXIST, op ANY, op ALL...
 - Find the maximum age and species of the oldest pet

```

SELECT P.species, P.age FROM Pets P

```

```

WHERE P.age>=all (SELECT P2.age
FROM Pets P2);

```

```

mysql> SELECT P.species, P.age FROM Pets P
-> WHERE P.age>=all (SELECT P2.age
-> FROM Pets P2);
+-----+-----+
| species | age |
+-----+-----+
| dog     | 15 |
| cat     | 15 |
+-----+-----+
2 rows in set (0.01 sec)

```

- Find all pets with no vaccines (4,8, 13, 14)

```

SELECT * FROM Pets P

```

```

WHERE P.PID NOT IN (SELECT PV.PID FROM Pet_Vaccines PV);

```

```
mysql> SELECT * FROM Pets P
-> WHERE P.PID NOT IN (SELECT PV.PID FROM Pet_Vaccines PV);
```

PID	first_name	last_name	gender	weight	age	species	breed
13	Rocksie	De Alba	Female	40	1	dog	Cattle Dog
14	Leaf	Erikson	Female	0.2	4	chameleon	veiled
4	Stubina	Lou	Female	11	7	cat	DLH
8	Bambi	Andrade	Female	10	8	cat	DSH

```
4 rows in set (0.03 sec)
```

Task4: Final report

- For your final report you should submit a PDF file including:
 - All CREATE TABLE, INDEX, VIEW statements, as well as SELECT queries.
 - The snapshot of the results of your SELECT queries. If the results of your select queries are large you can include part of the results.

Task5: Project presentation and demos

- You will present your project during the last session of class.
- Your presentation should include:
 - Presentation slides describing your design and implementation.
 - A live demo of executing your SQL queries, including aggregate queries and nested queries.