University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: November 29, 2024

Group Number: 64

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Danny Pirouz	23642218	n8y4m	dannypirouz@gmail.com
Manik Bansal	18177527	u1z1r	manikbansal834@gmail.com
Nigel Thompson	40976938	x9v4v	niggles45@hotmail.com

1. **Group Repository**:
   This is our group's repository:
   https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8y4m_u1z1r_x9v4v

2. **SQL Script**:
   This is a link to our SQL script:
   https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_n8y4m_u1z1r_x9v4v/blob/cc5ae12a95b164bd38f03b64f00c2c0b4b87d52a/script.sql

3. Our project is a database for a pet shelter adoption system. The database provides the functionality of storing all the animals' information, adding, and updating animals' information, and many unique features where a user can find useful statistics. For example, a user can find the youngest animals for each shelter which could be useful information for adopters wanting a young pet. Our database is designed to make it very easy and clear for management to understand what is going on at every step of the adoption process. Ideally, we would allow a user to add, update, delete, etc. all tables in the database but that was not feasible with the time we had; however, there is lots of functionality and we accomplished a lot with just the requirements we had to implement!

   Our final schema for the most part did not change. We accidentally had animal_id be a foreign key for the Animal schema which we changed to not be a foreign key as it is an attribute (and primary key). Other than that little mistake, our schema did not change.

   Our SQL Queries:

QUERY	FILE NAME	LINE NUMBER
INSERT	appService.js	527, 543,
UPDATE	appService.js	351, 366, 381, 396
DELETE	appService.js	495
SELECTION	appService.js	596-611
PROJECTION	appService.js	423
JOIN	appService.js	455
Aggregation with GROUP BY	appService.js	624

Aggregation with HAVING	appService.js	510
Nested aggregation with GROUP BY	appService.js	477
DIVISION	appService.js	638

We had 2 update functionalities. One updates staff and another updates animals. Updating animals are these lines in appService.js if you need to see them: 183, 198, 213, 228, 243, 258, 292, and 309.

Aggregation with GROUP BY:

This query finds the youngest animal in each one of the shelters.

```
SELECT SHELTER_NAME, SHELTER_ADDRESS, MIN(AGE) AS MIN_AGE
FROM ANIMAL
GROUP BY SHELTER_NAME, SHELTER_ADDRESS
ORDER BY MIN_AGE ASC
```

Aggregation with HAVING:

This query finds the top donors who donated more than $400 to different animal shelters.

```
SELECT D1.name, D.email, SUM(D.amount) AS total_amount
FROM Donate2 D, Donators D1
WHERE D.email = D1.email
GROUP BY D1.name, D.email
HAVING SUM(D.amount) > 400
```

Nested aggregation with GROUP BY:

This query finds the average age of animals grouped by breed with at least 2 animals in that breed.

```
SELECT breed, AVG(age)
FROM ANIMAL A1
WHERE A1.breed IN (SELECT breed
                   FROM ANIMAL A2
                   GROUP BY breed
                   HAVING COUNT(*) >= 2)
GROUP BY breed
```

Division:

This query finds the donators who have donated to all the shelters.

```
SELECT D.name
FROM Donators D
WHERE NOT EXISTS (
      SELECT S.address, S.name
    FROM Shelter S
    WHERE NOT EXISTS (
           SELECT D2.address, D2.name
           FROM Donate2 D2
           WHERE D2.email = D.email AND D2.address = S.address AND
           D2.name = S.name
```