University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Part "Formal Specification"

Date: Oct 23, 2018

Group Members:

Name	Student Number	CS Userid	Tutorial Section	Email Address
Sophia Shen	14747159	q8s0b	T1E	sophiashenziyi@gmail.com
Owen Tsai	26515155	f6c1b	T1G	tsaiyicheng3@gmail.com
Charlotte Zhu	53587151	r3s0b	T1A	coco99166@outlook.com
Brandon Djokic	26172056	t7s6	T1A	btdjokic@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

The platform we plan to use is the CS Ugrad Oracle installation and JDBC.

Deliverable 1:

The DB service will be mainly used by the different Zoo staff to facilitate their jobs and daily duties, as well as keeping track/log of animals.

More specifically:

Someone in a managerial role should be able to update,modify and delete related information for employees, animals, buildings, shows, assign a staff duties and assign keepers to certain animals. Furthermore, they should have full unrestricted access to any elements in regards to coordinating all components of the zoo.

A trainer should be able to Modify and update an animals attributes (food, weight), modify assigned show attributes, and modify habitats for animals.

A keeper should be able to check what animals and habitats they are assigned to, and check which buildings supplies are in for their related duties and tasks.

Deliverable 2:

Example of an INSERT statement - when Scamander Newt is employed as a volunteer, add to database his full name (Scamander Newt), employee ID 777, pay amount $0 (volunteer), his assigned walkee talkee number 420 and the zoo’s address - 847 Avison Way.

```
INSERT INTO Employee(f_name, l_name, id_employee, pay,
walkeetalkee#, zoo_address)
VALUES(Scamander, Newt, 777, 0, 420, 847 Avison Way)
```

Deliverable 3:

When an animal dies or get transported to other zoo, it will be deleted from the database.

```
DELETE
FROM Animal
Where id_animal = 1234
```

Deliverable 4:

Example of an UPDATE statement: updating a zebra (id = 1234)'s weight after it gains or loses weight

```
UPDATE Animal
SET weight = 190
WHERE id_animal = 1234
```

Deliverable 5:

Example 1: Query for keeper getting details for job

Explanation: A animal keeper at the zoo would need to get details on the animals he/she is in charge of, the type of food fed and the scheduled feeding times for each animal, as well as where to locate the animals and food. To get these information, we need to join **Animal** (to get the employee in charge of the animal, the animal's eating frequency and the number of food portions it eats), **Eats** (to determine what type of food each animal eats), **Food** (to get more information on the food), **StoredIn** (to get information on the building ID where food is stored), **Building** (to get address of the building where food is stored), **Maintains** (to get id of the animal habitat), and **Habitat** (to get location of the animal habitat).

Example 2: Query for looking at shows for the day

Explanation: The trainer needs to know on the date and time he/she has a show, the location of the show, and what animal is he/she trains is going to perform on that show. To do this query, we need to do a natural join on **Animal** and **Trainer**, and then another natural join to **Show** (based on start time). Then we can select the row based on the **Trainer** employee's id and project start time, duration (of the show), animal id, and location.

Deliverable 6:

Example: animals living in a certain habitat

Explanation: if one animal in a habitat gets sick and the illness is contagious, all of the animals that live in the same habitat needs to be tested and prevent the disease from spreading. In this case, we want to find the animals that live in a certain habitat.

Deliverable 7:

Explanation: We would return a count of all types of animals in the zoo, we could use the GROUP BY condition to group them by each animal species.

Deliverables 8-10:

1. A query where we would get a list of buildings not currently in use (the field used_for of Building would be null)
2. A query where we would get a list of all the buildings that a certain amount of food is located in (by considering how many types of food in each building and how many servings available for each food type)
3. A query where we would see what foods are going to expire in by the end of a given date (based on date_expiry of Food so zoo keepers can always order food supplies in time)
4. A query where we would see what habitats are of at least a given size, and biome type. (This would be useful to determine where to place a newly arrived animal)
5. A query finding all the employees named “Newt” as well as their contact information
6. A query of finding the weights of all lions (species name: *Panthera leo*)
7. A query of finding the first name and last name of all volunteers (those who get no paid)
8. A query of finding the addresses of all zoos in a given country
9. A query of finding the locations of habits that is 25 degree celsius
10. A query to find all the shows and their locations and the start time

Deliverable 11:

To have the information of all of the shows of a certain zoo in during a certain time period, the manager may want to display the type of the show, the performer and the animals involved on a webpage or a flyer. They can use the view created from columns (type, Starttime, duration) of **Show** with Starttime and duration conditions.

Deliverable 12:

We are planning to split the work evenly across all group members, for all aspects of the upcoming project including design, coding, and debug.

First attempt to divide tasks among group members:

Charlotte

- Learn about connecting to server and setup for the project
- Create data for the **relationships**
- Embed the SQL statements in a program and code the programming logic
- Work together to prepare for demo

Owen

- Create table and other database objects for **entities**
- Code a set of queries involving Employees
- Create the GUI
- Work together to prepare for demo

Brandon

- Create table and other database objects for **relationships**
- Code a set of queries involving Animals
- Write tests for the queries
- Work together to prepare for demo

Sophia

- Create data for the **entities**
- Code a set of queries involving habitats/location
- Create complete documentation for code
- Work together to prepare for demo