HTTP-5112 DB Final Project Proposal

Erko Abdurahman

N01662927

Nov 27, 2023

**SCENARIO**:

Im managing the data at an animal sanctuary so 2 database features that would assist in the sanctuaries operation would be:

* If an animal joins or leaves the sanctuary, i would use triggers to make it easier to update all necessary tables
* Assuming that donors have the option to earmark their donations for certain types of animals, i would create triggers, functions, and a view to see the total donated funds for each type of animal as well as by who

**What problem am I solving?:**

With my features I'm solving 2 problems that this animal sanctuary is facing.

Problem1

* One thing that's very common in animal sanctuaries is that animals come and go so if the sanctuary is going to have a database keeping a record of all animals, it should also contain a record of all animals not just coming in but also a record of all animals that have left the sanctuary.

Problem2

* Another problem i'm solving is that donors might want to earmark their donations for a certain type of animal so being able to see a record of how many donations certain animals have as well as the total amount in dollars is important for an animal sanctuary

**How will my features solve these problems?:**

Problem 1 Solution:

* The solution to the first problem using my feature would be that you'd be able to have a table that logs the changes in animals in and out of the sanctuary so you'd be able to see what animals came in and left as well as when all in one place so queries (unless for detailed data) isn't needed.

Problem 2 Solution:

* Similar to the first problem, there is a need for logging which animals are getting how many donations so by creating a view with less sensitive information about donors you can track each animal's donations received at the sanctuary!

**What are my tables, how are they composed and why is is justified by my solutions?:**

Schema:

| **Animals\_Table** |  |  |
| --- | --- | --- |
| COLUMN | DATA TYPE | CONSTRAINTS |
| animal\_id | INT(3) | PK, NOT NULL |
| name | VARCHAR(30) | NOT NULL |
| animal | VARCHAR(30) | NOT NULL |
| species | VARCHAR(30) | NOT NULL |
| gender | VARCHAR(30) | NOT NULL |

| **Animal\_Status\_Table** |  |  |
| --- | --- | --- |
| COLUMN | DATA TYPE | CONSTRAINTS |
| status\_id | INT(3) | PK, NOT NULL |
| animal\_id | INT(3) | FK, NOT NULL |
| enter\_date | DATE | NOT NULL |
| exit\_date | DATE | NULLABLE |
| status | VARCHAR(30) | NOT NULL |

| **Donations\_table** |  |  |
| --- | --- | --- |
| COLUMN | DATATYPE | CONSTRAINTS |
| donation\_id | INT(3) | PK, NO NULL |
| donor\_id | INT(3) | FK, NOT NULL |
| animal\_id | INT(3) | FK, NOT NULL |
| donation\_date | DATE | NOT NULL |
| amount | INT(100) | NOT NULL |

| **Donors\_Table** |  |  |
| --- | --- | --- |
| COLUMN | DATA TYPE | CONSTRAINTS |
| donor\_id | INT(3) | PK, NOT NULL |
| f\_name | VARCHAR(30) | NOT NULL |
| l\_name | VARCHAR(30) | NOT NULL |
| email | VARCHAR(30) | NOT NULL |
| phone | INT(12) | NOT NULL |

Justification:

These tables are justified by my solutions because using normalization it reduced redundancy and duplication while increasing simplicity and readability of the database. For problem 1, having the animal table and animal status table separate allows for one table to show the animals details while the other keeps a log of the animals that are currently in the sanctuary or that have left including their status. Keeping the exit date as nullable means that information isn't required as a lot of animals will be within the sanctuary thus not having an exit date. For problem 2, we can keep track of donations and to which animals through the donations table and for more detail we can see who donated in the donors table. The donations table acts as a bridging table so we can see which donor donated to which animal more specifically and in more detail.

**What are your database tools (procedures, functions etc.), and why are they justified?**

The tools being used for this database will mainly include a view, trigger and a stored function. These are important to my solution because first a view is needed so that I can display how many donations each animal has and by who without displaying sensitive information from the donors table. For the trigger this would be needed as when an animal is removed from the animals table it will automatically update in the animal status table which will reflect those changes in the status. As for the stored function this would be needed so that I can add the correct information needed for the view so I can view the previously mentioned information.