# Design and Proposal of a Zoo Database Management System

Navaneeth Ashok: N01358627

**Problem**

The zoo as of now does not contain an automated system for scheduling various functionalities that are required for it’s smooth operation. This lack of automation will hamper the efficiency of the system. In order to maximise the customer retention and to gain maximum popularity, the zoo requires a database management system in the backend to help with it’s daily operations.

**Solution**

In order to increase the efficiency of the system, I’m proposing an investment in a database for keeping the record of all the components of the zoo digitally and then using the same database to automate the schedule generation of enclosure timings, feeding, medical check-ups and automated billing system depending upon the number of open exhibitions.

* **Enclosure Timings**
  + Every animal follows a sleep schedule, and it’s important to make sure that the animals are properly relaxed and are not disturbed by the visitors. To ensure this, we’ll generate an enclosure timing schedule which will make sure the enclosures are not disturbed during the sleep time of the animals. On the presentation front, if needed we can provide a TV to display whether the enclosure is closed or open. If it’s closed, how much more time for the same to open. This will be achieved through a function. Also, we can show a dynamic map of the zoo with currently active enclosures shown in green and the closed one’s in red. If an enclosure is going to be opened in less than 30 minutes, it’ll be shown in yellow.
* **Feeding Schedule**
  + The automated feeding schedule is based on the data that is present in the diet table and the employee table. The system will generate a schedule based on the availability of the staff, checking which staff is present during the shift which overlaps with the animal’s feeding time.
* **Medical Check-up Schedule** 
  + A table containing the name of the medical staff(employee) and the animal id and the schedule of the medical check-up for the current week, it’ll contain a column to tick whether an inspection is already carried out.

**Table Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
| **ANIMALS** | | | |
| animal\_id | INT | PK |  |
| enclousre\_id | INT | FK |  |
| animal\_name\_or\_code | VARCHAR |  |  |
| animal\_type | VARCHAR |  |  |
| gender | VARCHAR |  |  |
| date\_of\_birth | DATE |  |  |
| date\_of\_admission | DATE |  |  |
| care\_taker\_in\_charge | INT | FK |  |
| sleep\_time\_start | TIME |  |  |
| sleep\_time\_end | TIME |  |  |
| feeding\_time\_start | TIME |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **DIETS** | | | |
| diet\_id | INT | PK |  |
| animal\_id | INT | FK |
| diet\_name | VARCHAR |  |  |
| diet\_weight | INT | kgs |  |
| diet\_time | TIME |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ENCLOSURE** | | | |
| enclosure\_id | INT | PK |  |
| enclosure\_type | VARCHAR |  | Open/ Cage/ Pool/ |
| enclousre\_number | VARCHAR |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE** | | | |
| emp\_id | INT | PK |  |
| emp\_fname | VARCHAR |  |  |
| emp\_lname | VARCHAR |  |  |
| hire\_date | DATE |  |  |
| salary | DOUBLE |  |  |
| shift\_time\_start | TIME |  |  |
| shift\_time\_end | TIME |  |  |

|  |  |  |
| --- | --- | --- |
| **MEDICAL CHECKUP SCHEDULE TABLE** | | |
| schedule\_id | INT | PK |
| emp\_id | INT | FK |
| animal\_id | INT | FK |
| date\_of\_checkup | DATE TIME |  |
| checkup\_done | BOOL |  |

|  |  |  |
| --- | --- | --- |
| **FEEDING SCHEDULE TABLE** | | |
| feeding\_sch\_id | INT | PK |
| emp\_id | INT | FK |
| animal\_id | INT | FK |
| diet\_time | TIME | FK |

|  |  |  |
| --- | --- | --- |
| **EXHIBIT SCHEDULE TIME** | | |
| exhibit\_id | INT | PK |
| animal\_id | INT | FK |
| enclosure\_id | INT | FK |
| time\_start | TIME |  |
| time\_end | TIME |  |

|  |  |
| --- | --- |
| **LOG TABLE TRIGGER** | |
| feeding\_sch\_id\_old | INT |
| feeding\_sch\_id\_new | INT |
| emp\_id\_old | INT |
| emp\_id\_new | INT |
| animal\_id\_old | INT |
| animal\_id\_new | INT |
| time\_old | TIME |
| time\_new | TIME |

**Database Tools Required**

* **VIEW**
  + View showing all animals, active times, enclosure ids and enclosure number
* **TRIGGER**
  + For logging changes to the feeding schedule table
* **PROCEDURE**
  + To populate the employee table faster, will be used for visitor table in next phase
* **FUNCTION**
  + A function to show how many more hours is left for an enclosure to open, or how many more hours before the enclosure is closed for the animal’s sleep time

**Reason to select this architecture and tools**

The tables relations are designed to avoid data redundancy and follows the guidelines required to make in NF compliant. There are certain things that are already present in the table, which does not make sense to store as the result can be obtained dynamically. For this purpose the VIEW method is used. It fetched data from different tables and shows it in an organised and easy to understand way for the end user or visitors. The triggers are used to keep track of the changes made to the feeding schedule table. I might implement a trigger to re-populate the feeding schedule of an animal if a change to it’s sleeping time is made to the ANIMALS table. To populate the employee table faster, a procedure will be written. A function is in place to show how much more time a visitor has to wait for before a particular enclosure is opened to the public, or how much more time a visitor can spend near an enclosure before it is closed for the animal’s nap time.

**Timeline**

|  |  |
| --- | --- |
| **TIMELINE** | |
| **Description** | **Hours** |
| Table Creation and Verification | 3h |
| Populating Table | 2h |
| Creating Views | 1h |
| Triggers | 2h |
| Procedure | 1h |
| Function to show time | 0.25h |
| Automated Table generation using triggers or custom functions - Schedule Tables | 3h |
| Testing and Validation | 3h |
| Total | **15.25h** |