

## **5-2 Milestone 4**

by Brandon Hollada

Southern New Hampshire University

CS-499-19649-M01

Submitted to Neil Kalinowski

For Artifact artifact three I chose to use the final from CS340, the same one that was used for artifact two. The original goal of the project was to have a program that would authenticate the user and interact with a MongoDB. This database would be used to display information on animal shelter animals. The reason that I decided to utilize this project for the database portion was because I found it difficult when I first worked on it a year ago. The project was laid out with each part and each implementation done through steps each week. Even with these steps, I was slightly confused by what I was doing on with the project. I believe that enhancing the project will show the improvements that I have made and the further understanding I have gained over databases. The main way that I looked to do this is to migrate the database used from the MongoDB, noSQL, to a SQL database. This would help to show that I have an understanding of the differences between the two and I am able to implement a database. A SQL is different from the original noSQL by having defined data types, a fixed schema, primary keys that must be unique, and is a table based system. This is where I had to find solutions from the original design to be able to read the data that was organized to work with NoSQL.

To make these enhancements, I utilized MySQL. I took the previous .csv file and read it in a way that would work for MySQL. I made sure to read in each field and if it was blank, I

would add a null to that location. I did this for all the fields, but it could be modified to not accept a record if certain fields are empty. For instance, the fields that are know like outcome\_type and locations should always have an entry from the shelter. These could be modified to not read null characters, which would mean the entry would be skipped if it had one. While there are inmprovements, such as the previosu one, I feel that the modification meet the outcomes I originally planned. The database has been migrated to SQL and functions correclty based on the change. I showed that I could create and manage a database without a step-by-step guide and that I was able to troubleshoot for solutions on my own.

As for what I learned, I was able to see the consept of schema's in databases and how differences can be applied. I learned the value of having primary keys to attached to databases and to ensure differences between the records. I learned how to read the errors while creating the table and how to make modificaitons based on that feedback. Some of the main challenges that I had was ensuring I had the correct packages to complete the task I wanted. I was able to learn about the sqlconnector and how to utilize it to create an engine that runs the database. Below are screenshots showing the proces of creating and troubleshooting the database table.

## Initializing the mySQL database and table with user permissions

```
MySQL 9.3 Command Line Client - Unicode
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 9.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE rescues;
Query OK, 1 row affected (0.038 sec)

mysql> USE rescues
Database changed
mysql> CREATE USER 'aacuser'@'localhost' IDENTIFIED BY 'SNHU1234';
Query OK, 0 rows affected (0.082 sec)

mysql> GRANT ALL PRIVILEGES ON rescues.* TO 'aacuser'@'localhost';
Query OK, 0 rows affected (0.020 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected, 1 warning (0.020 sec)

mysql> CREATE TABLE animals ( age_upon_outcome VARCHAR(50),
-> animal_id VARCHAR(20) PRIMARY KEY,
-> animal_type VARCHAR(30),
-> breed VARCHAR(100),
-> color VARCHAR(100),
-> date_of_birth DATE,
-> datetime DATETIME,
-> monthyear VARCHAR(10),
-> outcome_subtype VARCHAR(50),
-> outcome_type VARCHAR(50),
-> sex_upon_outcome VARCHAR(50),
-> location_lat DOUBLE,
-> location_long DOUBLE,
-> age_upon_outcome_in_weeks FLOAT) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.144 sec)
```

Troubleshooting warnings for data fields being truncated.

```
-----+
5 rows in set (0.004 sec)

mysql> ALTER TABLE animals
-> MODIFY date_of_birth VARCHAR(20),
-> MODIFY datetime VARCHAR(20);
Query OK, 44 rows affected (0.291 sec)
Records: 44  Duplicates: 0  Warnings: 0
```

Troubleshooting incorrect records added by finding missing column

```
mysql> ALTER TABLE animals
-> ADD COLUMN 'name' VARCHAR(20)
-> AFTER monthyear;
```

I initially thought that the animal\_id was unique to each animal but found repeats while adding to the table. To fix this, I used the index as the primary key for the table.

```

mysql> ALTER TABLE animals
-> ADD COLUMN animal_index INT FIRST;
Query OK, 0 rows affected (0.212 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE animals
-> DROP PRIMARY KEY,
-> ADD PRIMARY KEY (animal_index);
Query OK, 0 rows affected (0.184 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

Final load into the table, which catches all empty fields and makes them null to correctly read in each row.

```

mysql> LOAD DATA LOCAL INFILE 'C:\\Users\\Hollada\\Desktop\\Artifact3\\Project2\\aac_shelter_outcomes.csv'
-> INTO TABLE animals
-> FIELDS TERMINATED BY ','
-> OPTIONALLY ENCLOSED BY '"'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (@animal_index, @age_upon_outcome,
-> @animal_id,
-> @animal_type,
-> @breed,
-> @color,
-> @date_of_birth,
-> @datetime,
-> @monthyear,
-> @name,
-> @subtype,
-> @otype,
-> @sex_upon_outcome,
-> @lat,
-> @lon,
-> @weeks
-> )
-> SET
-> animal_index = @animal_index,
-> age_upon_outcome = NULLIF(@age_upon_outcome, ''),
-> animal_id = NULLIF(@animal_id, ''),
-> animal_type = NULLIF(@animal_type, ''),
-> breed = NULLIF(@breed, ''),
-> color = NULLIF(@color, ''),
-> date_of_birth = NULLIF(@date_of_birth, ''),
-> datetime = NULLIF(@datetime, ''),
-> monthyear = NULLIF(@monthyear, ''),
-> name = NULLIF(@name, ''),
-> outcome_subtype = NULLIF(@subtype, ''),
-> outcome_type = NULLIF(@otype, ''),
-> sex_upon_outcome = NULLIF(@sex_upon_outcome, ''),
-> location_lat = NULLIF(@lat, ''),
-> location_long = NULLIF(@lon, ''),
-> age_upon_outcome_in_weeks = NULLIF(@week, '');
Query OK, 10000 rows affected (1.929 sec)
Records: 10000 Deleted: 0 Skipped: 0 Warnings: 0

```