Project 2: Zoolicious ERD

Objectives
Implement a relational database schema given an ER diagram.

Instructions
Read The Definitive Guide to SQLite, 2nd ed. Chapter 3 "SQL for SQLite" through page 55. Focus on the section titled "Creating a Database" (p53 - 55).

Read the CREATE TABLE documentation from SQL As Understood by SQLite and begin to familiarize yourself with the syntax diagrams.

Scenario (Strange, Yet Somehow Familiar)

The doorbell startles you! Entranced at the computer (a bad habit) you get up to see who it could be at 2:30AM. You open the door and see a FedEx man.

"*(insert your name)*?" he asks, raising an eybrow.

"Yeah, that's me," you say. He scans the package with a handheld device that resembles a sandwich, but is a very dark grey. The laser glows in the dim light and the sandwich-thing emits a loud *beep* as he shoves the package into your hands.

"Ok. Great. Have a nice day." The FedEx man quickly turns and leaves.

"Strange, no address or anything on this package," you think as you tear it open. "I don't care what it is, I haven't received a package from anyone in so long it could be worms and I'd -- oooh, a phone!" A strange, futuristic looking phone (strangely shaped like a banana, but painted gunmetal grey) lands in your hands. Suddenly, it rings. "Uhhh... heh- Hello?"

"Hello, *(you)*. Do you know who this is?" asks the dark voice on the other end.

"...*Morpheus*?" you say.

"Yes. I've been looking for you. I don't know if you're ready to see what I want to show you, but unfortunately you and I have run out of time. They're coming for you, and I don't know what they're going to do."

"Who's coming for me?"

"Look out the window and see for yourself."

"What, right now?"

"Yes. Now. Do it slowly. The back patio window."

Out across the back yard you see three clowns wearing black suits, black ties and dark sunglasses. "Oh, poop!"

"Yes."

"What the hell do they want from me?"

"I don't know, but if you don't want to find out I suggest you get out of there."

"How?"

"I can guide you but you must do exactly as I say."

"Ok."

"The house across the street from you is empty."

"But what if they -"

"Go, now!" shouts the voice, and you grab your laptop and run out the front door, scared for your life. With manic strength you burst right through the front door of the house across the street, leaving a human-shaped hole in the wall. "Stay here for just a moment. When I tell you, go to the end of the hall, to the room at the end of the hall. Stay as low as you can. Go, now. Good. Now, in the corner there is a desk."

"How do you know all this?" you ask, incredulously. How can this be happening?

"We don't have time. To your left there's a desk. Go to it. Open your laptop. You can use the ethernet jack on the wall at your feet. Now, design and implement a relational database."

"No way. No way. This is crazy."

"There are two ways out of this house. One is that database, the other is in their custody. You take a chance either way. I leave it to you."

"This is insane. Why is this happening to me? What did I do? I'm nobody. I didn't do anything. I'm gonna die. Poop. I can't do this."

Attribution: Wachowski, A. & Wachowski, L. (1999). *The Matrix*. Warner Bros. Pictures.

Requirements
This project exercises your ability to work with the SQLite console, your understanding of the ER Model and ERDs, and SQL DDL statements. Your goal is to implement a database schema given an ERD. Your schema must represent the entities, attributes, relationships and constraints according to the ERD and its documentation below.

Domain
Our domain for this project is about zoos, and you might imagine the database
supporting a simple, multi-user, database-backed application.

Zoolicious App ERD

Note that while 'id' is not part of the domain, it is consistently presented here as an explicit key attribute.
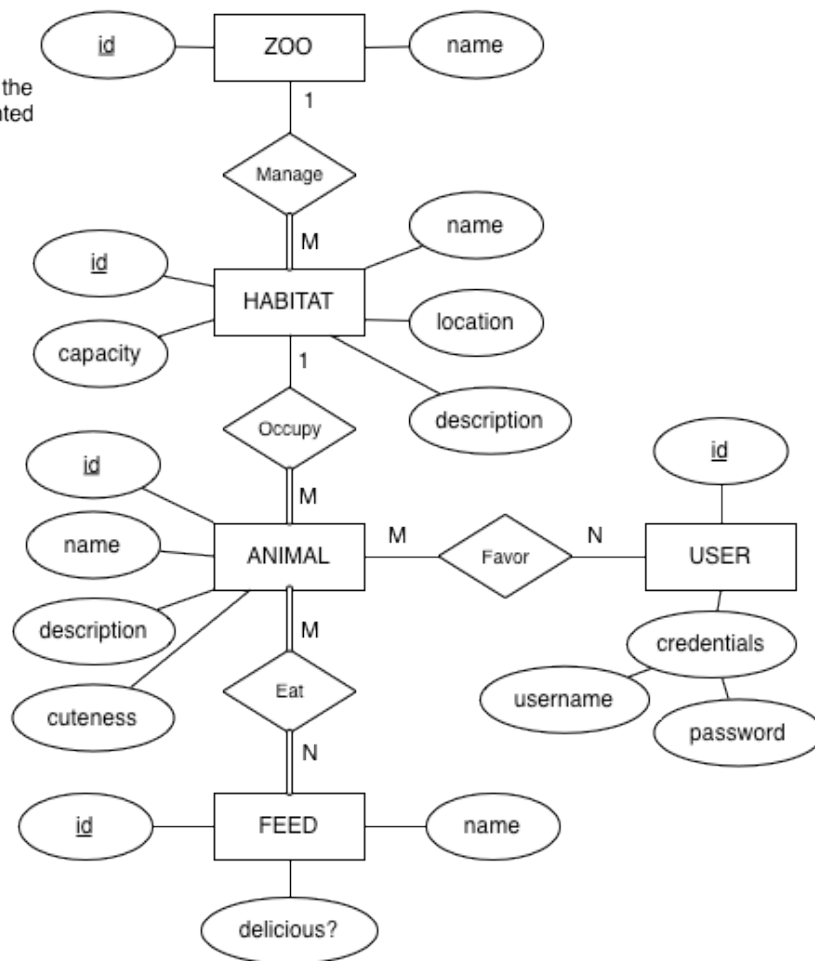
Your schema implementation should adhere to the ERD above; and the following
description and constraints.

- A zoo manages many animal habitats.
- Zoos in the system have a name. Names are not necessarily unique.
- Habitats are simple "living areas" occupied by animals and managed by zoos. They
  have a name (eg. "Polar Vestibule 3"), a description ("A freezing environment just
  perfect for polar llamas") and a location ("West wing"). All three of these attributes
  correspond to just text values.
- Each habitat is managed by one specific zoo. They can be occupied by any number of
  animals up to the maximum capacity of the habitat. Each habitat would have a
  minimum capacity of one animal.
- By law, no habitat may have a higher capacity than 50 animals (because at that point
  all hell breaks loose and liability insurance just doesn't cover wild animal frenzies).

- Animals have a name, description and cuteness rating. They occupy habitats, and no animal may occupy more than one particular habitat.
- An animal's cuteness rating is a discrete numeric score, standardized by ZACARS (The Zoo Animal Cuteness Association Rating Standard), and is generally a value between 1 and 10, inclusively.
- Animals eat feed. An animal might eat a few different kinds of feeds. And a particular kind of feed (eg, chocolate mini-doughnuts) might be eaten by multiple animals.
- Feed have a name and a deliciousness quality that is either yes or no. This deliciousness is according to human standards, not animal standards, and is always a yes or no answer (no "Hmmmm... might be delicious with some crème fraiche and fava beans").
- This database will be used by an application that allows users to create lists of their favorite animals.
- Every user has credentials, which consist of a unique username and a password. Usernames and passwords must be between 9 and 16 characters in length and can never be blank.

Tips

Spend time exploring SQLite's CREATE TABLE statement and the constraints that SQLite supports. You may need a NOT NULL, DEFAULT, UNIQUE and/or CHECK constraints.

Refer to the SQLite reading from Project 1 regarding how you can pass a text file to SQLite for execution (also shown in the instructions below). *Windows users, be sure to use a good quality editor, not notepad, that allows you to save a text file with unix-style line endings*. Ask on Piazza if you get stuck.

Instructions

Start sqlite3 with an arbitrary new database file.

`sqlite3 zoolicious.db`

Enter the DDL (`CREATE TABLE`) statements necessary.

or

Write your DDL statements in a text file, and tell SQLite to execute it upon a new database:

`sqlite zoolicious.db < zoolicious.txt.sql`

Grading Criteria (50 points)

Printed copy of your SQLite `.schema`	10 points
SQL DDL statements use SQLite conventions (datatypes, etc)	20 points
Correctness of tables, relationships and constraints	20 points

Bring your printed copy of the database schema (a list of the CREATE TABLE statements) to class on the due date.