**CSC 355 Database Systems**

**Fall Quarter 2018**

**Assignment 2 (9/12)**

**Due Sunday, September 23rd @ 11:59pm.**

**Reading:** Sections 4.3, 4.4, and 5.1 of Elmasri & Navathe 6th Edition. Chapters 2, 4, and 6 of Price's SQL book may be helpful as a reference, though they contain more material than we have covered. To find the Price book, you can search for Oralce 12c at library.depaul.edu (then you can choose “view ebook”)

For both Parts 1 and 2, please submit your SQL code. Submitting your output is optional.

**Part 1. SQL DDL (36 points)**

You are given a following schema:

**PublicChauffeurs**(License Number, Renewed, Status, Status Date, Driver Type, License Type, Original Issue Date, Name, Sex, Chauffer City, Chauffer State, Record Number)

The table is based on a real data set original taken from City of Chicago data portal (located here: <https://data.cityofchicago.org/Community-Economic-Development/Public-Chauffeurs/97wa-y6ff>

1. Write SQL DDL to create this table. Make sure to look at the data to choose reasonable domains for each column.
2. Declare primary and foreign keys as necessary in your SQL code.
3. Create at least three reasonable additional attribute-level, tuple-level, NOT NULL, or UNIQUE constraints in your SQL code.

**Part 2. SQL DML (64 points)**

You were hired to do some data analysis for a local zoo. Below is the data table, including the necessary constraints and all the insert statements to populate the database.

-- Drop all the tables to clean up

DROP TABLE Animal;

-- ACategory: Animal category 'common', 'rare', 'exotic'. May be NULL

-- TimeToFeed: Time it takes to feed the animal (hours)

CREATE TABLE Animal(

AID NUMBER(3, 0),

AName VARCHAR2(30) NOT NULL,

ACategory VARCHAR2(19),

TimeToFeed NUMBER(4,2),

CONSTRAINT Animal\_PK PRIMARY KEY(AID)

);

INSERT INTO Animal VALUES(1, 'Galapagos Penguin', 'exotic', 0.5);

INSERT INTO Animal VALUES(2, 'Emperor Penguin', 'rare', 0.75);

INSERT INTO Animal VALUES(3, 'Sri Lankan sloth bear', 'exotic', 2.5);

INSERT INTO Animal VALUES(4, 'Grizzly bear', 'common', 3.5);

INSERT INTO Animal VALUES(5, 'Giant Panda bear', 'exotic', 1.5);

INSERT INTO Animal VALUES(6, 'Florida black bear', 'rare', 1.75);

INSERT INTO Animal VALUES(7, 'Siberian tiger', 'rare', 3.5);

INSERT INTO Animal VALUES(8, 'Bengal tiger', 'common', 2.75);

INSERT INTO Animal VALUES(9, 'South China tiger', 'exotic', 2.25);

INSERT INTO Animal VALUES(10, 'Alpaca', 'common', 0.25);

INSERT INTO Animal VALUES(11, 'Llama', NULL, 3.25);

Since none of the managers in the zoo know SQL, it is up to you to write the queries to answer the following list of questions.

1. Find all the animals (their names) that take less than 2 hours to feed.
2. Find all the rare animals and sort the query output by feeding time (any direction)
3. Find the animal names and categories for the animals that are related to a bear (hint: remember the LIKE operator)
4. Return the listings for all animals whose rarity is not available in the database
5. Find the rarity rating of all animals that require between 1 and 2.2 hours to be fed
6. Find the names of the animals that are related to the tiger and are not common
7. Find the minimum and maximum feeding time amongst all the animals in the zoo. Use only one query.
8. Find the average feeding time for the rare animals

I am also including two more challenging queries that you may attempt to further test your knowledge of SQL. They aren’t mandatory – you will get 100% credit for the homework if you correctly answer queries A-H.

* **EC1.** Find the listing of the animal that requires the longest feeding time. Do not hardcode any feeding time values.
* **EC2.** Find the names of the animals that can be fed in less than the average overall time + 25% (i.e. in less than 1.25 \* average feeding time). Do not hardcode any feeding time values.