**Instructions**

Edit Instructions

Q1. (15pts) Advanced SQL.

In this example, we will consider a Zoo database. Please download the Zoo database from the Practice folder.  
Open the ZooDatabase.sql file using SQLDeveloper. Answer the following questions:

(i) Find the total feeding time for all of the rare animals.  
(ii) Which animal(s) have a `time to feed' larger than every rare animal? Give the id and name of the animal.  
(iii) Name zookeepers handling at least 4 animals.  
(iv) Find the names of the animals that are not related to the bear.  
(v) List zookeepers earning the most while feeding animals.

Note for (v): 1. We are looking for that one single zookeeper who earns the most. 2. Dont use fancy DB-specific keywords such as top, fetch first, range, etc. Use SQL clauses that we have covered in the class.

**Submit your A.sql along with PDF of screenshots.**

Q2. (15pts) Using CTEs for Data Analytics. You are provided data for the Loan table .

(i) Write a CTE 'Histogram' that counts the number of loans for each risk score. Write a simple query to display the results of the CTE in descending order of the count.  
(ii) Spot the risk score(s) with the minimum number of loans. Call this CTE as RiskScoresWithMinLoans.  
(iii) Using your Histogram and RiskScoresWithMinLoans CTE, write a SQL query to find description of loans which have a risk score spotted in (ii).

Note: Data is available in raw form. Inserting by hand is a waste of time, unless you have extreme patience. You will have to write Insert statements to insert data into the database first. Create insert statements programmatically. Write a function in your favorite programming language to open and read this CSV file line by line and generate statements of the form

Insert Into Loan Values (Val1, ‘Val2’, ‘Val3’, Val4,…);

Where Val1 and Val4 are numeric and Val2 and Val3 are strings. Then copy/paste ‘Insert Into’ statements generated from your program to B.sql file and upload data into the table.

Note: You may encounter errors and will have to fix some errors.

**Submit your B.sql along with PDF of screenshots.**

Q3. (15pts) We are provided data from a professional network DePauledIn (DePauledIn.sql). This data provides account details of people connected to each other. *SocialNetwork(Person, Friend),*which gives immediate friends of a person. A person may have many friends. This SocialNetwork relation is similar to Twitter in that friendship relationship is not necessarily bi-directional. This implies that if a tuple of SN says that A is a friend of B, then it does not imply that B is a friend of A. If B is a friend of A, which is possible, then it will be identified by another tuple. The data has cycles.

Note: There are cycles in the data.

Write recursive SQL queries to determine the following:

1. People to whom Brad can reach to either directly or transitively.
2. People in DePauledIN network to whom Brad cannot reach to.
3. Only those people who are connected to Christine via transitive relationship i.e not an immediate follower.
4. Find the shortest path to reach from Amy to James.

**Submit your C.sql along with PDF of screenshots.**