CSCI-230

Background:

This exercise focuses on a state-of-the-art topic known as *Bioinformatics*. Roughly speaking, *Bioinformatics* is the study of applying Computational techniques to analyze, organize and manage the huge amounts of data being produced in bulk quantities from biological (e.g. genomic and proteomic) experiments.

A gene is a functional unit of heredity that is a segment of DNA (*Deoxyribo-Nucleic Acid*) located in a specific site on a chromosome inside the cell. The main job for genes is to direct the formation of enzymes and other proteins through the processes of transcription and translation – *Central Dogma of Molecular Biology*. Every cell in an organism contains a number of DNA chromosomes that embed genes in addition to other information (currently referred to as *garbage* as biologists do not know of any functions that could be attributed to those chromosome parts).

DNA is a **poly**mer. The **mono**mer units of DNA are called nucleotides, and the polymer is known as a "polynucleotide." Each nucleotide consists of a 5-carbon sugar (deoxyribose), a nitrogen-containing base attached to the sugar, and a phosphate group. There are four different types of nucleotides found in DNA (differing only in their nitrogenous base). The four nucleotides are given one letter abbreviations as shorthand representations: ***A*** for adenine, ***G*** for guanine, ***C*** for cytosine, and finally, ***T*** for thymine. The following is an example of a part of an actual gene sequence:

GATCCTCCATATACAACGGTATCTCCACCTCAGGTTTAGATCTCAACAACGGAACCATTGCCGACATGAGACAGTTAG
GTATCGTCGAGAGTTACAAGCTAAAACGAGCAGTAGTCAGCTCTGCATCTGAAGCCGCTGAAGTTCTACTAAGGGTG
GATAACATCATCCGTGCAAGACCAAGAACCGCCAATAGACAACATATGTAACATATTTAGGATATACCTCGAAAATAA
TAAACCG ...

Just as genes code for information that enables the creation of proteins, proteins, in turn, code for the information that enables our bodies to perform almost all body functions and processes. Proteins have different functions: they can provide structure (ligaments, fingernails, hair), help in digestion (stomach enzymes), aid in movement (muscles), and play a part in our ability to see (the lens of our eye is made up of pure crystalline proteins).

A protein is a long chain of amino acids linked together. Just as there are 4 nucleotides that make up gene sequences, there are about 20 different amino acid units that make up protein sequences: ***A*** for Alanine - Ala, ***R*** for Arginine - Arg, ***N*** for Asparagine - Asn, ***D*** for Aspartic acid - Asp, ***C*** for Cysteine - Cys, ***Q*** for Glutamine - Gln, ***E*** for Glutamic acid - Glu, ***G*** for Glycine - Gly, ***H*** for Histidine - His, ***I*** for Isoleucine - Ile, ***L*** for Leucine - Leu, ***K*** for Lysine - Lys, ***M*** for Methionine - Met, ***F*** for Phenylalanine - Phe, ***P*** for Proline - Pro, ***S*** for Serine - Ser, ***T*** for Threonine - Thr, ***W*** for Tryptophan - Trp, ***Y*** for Tyrosine - Tyr, and ***V*** for Valine - Val. The following is a sample protein sequence:

HMTKWNKKDSTWTEQMANYSWDLNFGDDSKWYGRYQSQYIPCLDGEVIRFGPDGHKNMNPAKTFLWLSRDWKTNY
RGSFALVHWDICQEHNPNTTHTGAVGMWYLDVLFKRGPKMGMWLQGEIICYRADGIIQGHTCMSFYGERTCAQKPGT
YPCWLCRDACDEYLLMPNNHAAACWVHRVYSTDLYMAKNNMQMAYLIP

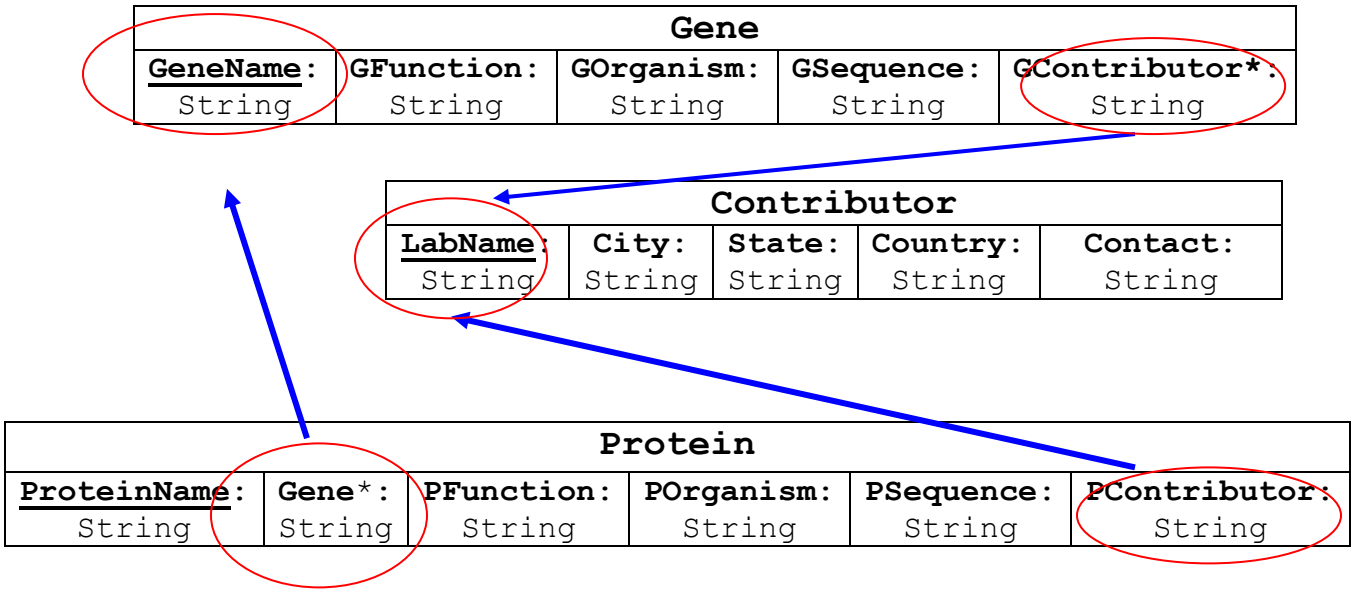
Description:

For this exercise, you will focus only on the management and organizational aspects of biological data. You are to create an application to manage different types of sequence data. Such data includes mainly genes and proteins each of which belongs to some known organism.

Every gene and protein is described by a name, organism, sequence (limited only to allowed characters), function and contributors' information (i.e. research laboratory where the gene or protein was discovered). As aforementioned in the introduction, a protein is produced by a gene and this information must also be retained in the database.

A single contributor may contribute many protein or gene or sequences. For every such contributor, we need to keep track of the location (City, State and Country), and contact person.

The following database, called **cs230_BioDB**, has already been created for you. Primary keys are underline and foreign keys are denoted with asterisks (*) and linked by arrows to the fields they reference. You will utilize this database to build your application.



The following SQL commands were used to create the above database:

```
Create Table Contributor (
   LabName     VARCHAR(30),
   City        VARCHAR(20),
   State       VARCHAR(20),
   Country     VARCHAR(20),
   Contact     VARCHAR(15),
   Primary Key (LabName));

Create Table Gene (
   GeneName          VARCHAR(20),
   GFunction         VARCHAR(20),
   GOrganism         VARCHAR(20),
   GSequence         VARCHAR(250),
   GContributor      VARCHAR(30),
   Primary Key (GeneName),
   Foreign Key (GContributor) References Contributor(LabName));
```

```
Create Table Protein (
   ProteinName      VARCHAR(20),
   Gene             VARCHAR(20),
   PFunction        VARCHAR(20),
   POrganism        VARCHAR(20),
   PSequence        VARCHAR(250),
   PContributor     VARCHAR(30),
   Primary Key (ProteinName),
   Foreign Key (PContributor) References Contributor (LabName),
   Foreign Key (Gene) References Gene(GeneName));
```

The SQL scripts that create the above database tables and insert data into them can be found on the N: drive under `handouts/JDBC_Class_Exercise`.

Your application should contain two classes:

I. A class that creates a connection to the database in its constructor. This connection object will be used by other methods in this class. Include another method in this class to close the database connection. Finally, this class will include **a method for each of the functionalities outlined next**

The following is to be provided by your first class – make each as its method:

1. Ability to *add new contributors* along with their related information passed as parameters.

2. Ability to *add new genes* along with their related information passed as parameters.

3. Ability to *add new proteins* along with their related information passed as parameters.

4. Ability to *search for proteins produced by a given gene* (via its name). Method should take a *String geneName* as a parameter. For every match, display all field values from table `Protein`.

5. Ability to *search for all genes having superstring sequences of a given parameter string of DNA*. Method should take a *String dna* as a parameter and return all genes from table `Gene` that contain the parameter ANYWHERE in their `GSequence` field. For every match, display all field values from table `Gene`.

6. Ability to *search for genes given a contributor's country (e.g., genes contributed by all labs in the US)*. Method should take a *String country* as a parameter. For every match, display all field values from table `Gene`.

II. A driver class that contains a `main` method to "test" each of the methods in **I**.