Project 3, CMPE 321, Introduction to Database Systems, Spring 2021

Due: **June 10, Sunday, 23:55**

Ask the project related questions in the Moodle forum!

1 Introduction

The study of drugs or chemicals and the effects they have on a living organism is called pharmacology. A drug is a chemical that interacts with a target molecule (such as a protein), which in turn results in a change in the molecule's or the corresponding cell's behavior.

Drugs in general, are not specific to only one particular molecule or cell. They may also interact with other molecules or other drugs and this is what causes side effects. For instance, anticancer drugs are selective for very rapidly dividing cells such as cancer cells and stimulate the cells to die. Hair loss is one of the side effects of anticancer drugs, since hair cells are also rapidly dividing. [1]

To cope with the tremendous number of drugs and targets (e.g., proteins that bind to drugs), online databases are created and commonly used in the literature. Online databases allow easy and fast access to information and most of them focus on a specific field. For instance, DrugBank contains information on drugs and drug targets, UniProt contains proteins and their structures, SIDER contains drugs and their side effects, whereas BindingDB curates the interactions between drugs and targets. Thus, a free online resource that integrates these databases would benefit drug discovery researchers, chemists, pharmacists, students, and the general public.

2 Project Description

In the previous project, you performed conceptual database design, drew ER diagrams, converted these ER diagrams into relational tables, and performed schema refinement for *DTBank*. In this project, you will go one step further and implement a unified Drug - Target database with a web-based user interface (UI) using the structure you have designed in Project 2 or its improved version. There will be users, database managers, and data extracted from several external databases such as DrugBank, SIDER, BindingDB, and UniProt in the system. *DTBank* should contain the following information:

1. **User** includes the following attributes; username, institute name, and password. There exists only one user with a specific username and institute. There can be an unlimited number of users. Passwords should be encrypted using the SHA256 algorithm and stored in the database accordingly.

2. **Database manager** consists of the following attributes: username and password. There exists only one database manager with a specific username. There can be at most 5 database managers registered to the system. Passwords should be encrypted using the SHA256 algorithm and stored in the database accordingly.

3. **DrugBank** includes DrugBank ID, drug name, description, and interaction with other drugs. By definition, each DrugBank ID is unique.

4. **SIDER** includes UMLS CUI (side effect IDs), DrugBank ID, and side effect name. By definition, each UMLS CUI is unique.

5. **BindingDB** includes Reaction ID, DrugBank ID, UniProt ID, target (protein) name, SMILES (chemical notation of drug), affinity in nM (the strength of the binding interaction between drugs and targets), the measure of the interaction (Ki, Kd, IC50), DOI (link on the web to identify the article or document that mentions the drug-target interaction. E.g., https://doi.org/10.1093/bioinformatics/bty593), authors of

[1] You can check out this link for further information on pharmacology.

the article or document, usernames of the corresponding authors, and institutions of the authors. Authors are considered as contributors. Each contribution makes an impact on the institute. By definition, each Reaction ID is unique and contributors are users of *DTBank*. [2]

6. **UniProt** includes UniProt ID and amino acid sequence of the corresponding protein. By definition, each UniProt ID is unique.

Two types of people will be using *DTBank*: users and database managers. Both types should be able to **log in** to the system, thus you should implement an authentication mechanism for database managers and users. You do not need to provide sign-up, change password, and reset password options. Prior to or during login the users of the web application should choose their roles among two options: *user* or *database manager*. After login users will be able to perform only the operations that are defined for their roles. Some *DTBank* users are actual contributors of the papers/documents placed in BindingDB. Each institute will have a score based on the number of publications and number of the contributors to these publications. Institutes get 5 points for each publication and get 2 points for each contributor of the corresponding publication.

3 Requirements

Your UI must support the following operations:

1. Database managers shall be able to log in to the system with their credentials.
2. Database managers shall be able to add new users to the system.
3. Database managers shall be able to *update* affinity values of drugs using Reaction IDs and *delete* drugs using DrugBank IDs.
4. Database managers shall be able to *delete* proteins using UniProt IDs.
5. Database managers shall be able to *update* contributors of papers/documents using Reaction IDs. (Author names, usernames, and passwords will be provided.)
6. Database managers shall be able to separately view all drugs listed in DrugBank, all proteins listed in UniProt, all side effects listed in SIDER, all drug - target interactions, all papers and their contributors listed in BindingDB, and all users in *DTBank*.
7. Users shall be able to log in to the system with their credentials.
8. Users shall be able to separately view the names, DrugBank IDs, SMILES strings, descriptions, target names, and side effect names of all drugs.
9. Users shall be able to view all interactions of a specific drug.
10. Users shall be able to view all side effects of a specific drug.
11. Users shall be able to view all interacting targets of a specific drug.
12. Users shall be able to view interacting drugs of a specific protein.
13. Users shall be able to view drugs that affect the same protein.
14. Users shall be able to view proteins that bind the same drug.
15. Users shall be able to view drugs that have a specific side effect.
16. Users shall be able to search a keyword and view the drugs that contain this keyword in their descriptions.
17. Users shall be able to view the drug(s) with the least amount of side effects that interact with a specific protein.
18. Users shall be able to view the DOI of papers and contributors of the corresponding paper.

[2] Note that this part is different than the Project 2.

19. Users shall be able to rank institutes according to their total scores (Decreasing order).

20. Users shall be able to filter interacting targets of a specific drug considering the selected measurement type and the range of the affinity values. This must be implemented as a **stored procedure**. Parameters of this procedure are measurement type, minimum affinity value, and maximum affinity value.

21. The system shall have three **triggers**:

    (a) When a drug is deleted, it should be removed from the the list of the interacting drugs of other drugs, and its corresponding entries from SIDER and BindingDB.

    (b) When a protein is deleted, its corresponding entries from BindingDB should be removed.

    (c) When a new publication is added to the system, the corresponding institute gets 5 points for a new publication and gets 2 points for each individual who contributed to it. Also, when there is an update on the publication that changes the number of contributors, update the corresponding institute's score accordingly.

4 Notes

- In the scope of this project, we are more interested in the functionality of the web interface, rather than the styling of UI. In other words, it is totally fine to create a system that satisfies all requirements and has zero line of CSS and Javascript.

- The allowed languages are PHP, Java, JavaScript, and Python. You can use a framework, however, you must write the SQL queries and boot the database server yourself. Note that you should set up the database and create the tables on your own. You are not allowed to use any tool that helps with these parts.

- Also, do not use database connector as your query supplier. Use it only as a query executor by running it with SQL strings written by you.

- You are not expected to deploy your system. So, it is fine if runs in your local.

- You are free to use **relational database** server of your choice. However, we will not accept submissions with non-relational databases or no database servers. Also, SQLite is not applicable for this project since it does not cover stored procedures.

5 Report & Grading

This project can be implemented either individually or as a team of two people. You are free to change teams in the upcoming projects. The submission must include your **code**, your updated ER diagrams, and a READ.ME file that describes how to run your code. Place all the required files into a folder named with the student IDs of the team members separated by an underscore (e.g. `2017400200_2018700120`). Zip the folder for submission and name the `.zip` file with the same name. Submit the `.zip` file through Moodle until the deadline. The system will be evaluated during a demo session that will be arranged. Demo day(s) will be announced. Before the demo date, we will share the data that will be used during the demo in Excel format. You **must** add the entries before the demo. **Any other submission method and late submissions are not allowed**. Each group should submit **one** .zip file.