University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Milestone 4

Milestone #: 4

Date: 25/11/2022

Group Number: 81

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Abdullah Eshaq	32999518	u0j3b	eshaq@student.ubc.ca
Oussama Saoudi	69249035	p9l4q	bsaoudio@student.ubc.ca
Oliver Oxford	81360497	l3v1o	oliox@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Project Overview

Our project is a simple version control system allowing users to join organizations and then contribute to repositories owned by these organizations by adding and deleting files, as well as creating commits to update these files.
Within our GUI, users can login and then join organizations, which then grants them access to all repositories owned by any orgs which they are part of. Then in the repositories tab, they can search out the one they want, and then view all files contained within, create and delete files, and edit and view file contents. Numerous analytics around this are also displayed, such as a list of file types in the repo, and stats about contributions to the organizations.

Repository Link:

https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project_l3v1o_p9l4q_u0j3b

SQL Script:

The SQL script can be found at src/sql/scripts/databaseSetup.sql in the project repository.

Database Schema Changes:

- Date_stamp: We changed date_stamp relation to use a single timestamp column instead of manually handling year, month, day, etc. This is done to leverage SQL's built-in type system for clearer code.
- Repository_object has a parent_directory_repository since to reference a foreign object, repository object must have a field that refers to the parent direction.
- Commit_message removed to simplify GUI
- Removed filename column from repository_object schema

Tables with example tuples, with Schemas

- Assigned_to(**Username: String, Issue_ID: Integer**)

```
SELECT * FROM ASSIGNED_TO
```

	USERNAME	ISSUE_ID
1	eshaq	2
2	jack	4
3	jill	5
4	oliver	3
5	oussama	1

- *Commit*(**Commit_SHA: String**, Commit_number: Integer, Repository_name: String, File_changed: String, Author: String, date_created: Timestamp)

```
SELECT * FROM COMMIT
```

	COMMIT_SHA	COMMIT_NUMBER	REPOSITORY_NAME	FILE_CHANGED	AUTHOR	DATE_CREATED
1	331dbc09c8afa4c077f448e4c1ef1e42297879d3fcfdbf…	1	CPSC304	/README.md	eshaq	2022-04-12 11:32:06.000000
2	331dbc09c8afa4c07fdadfasfadsf297879d3fcfdbf2e7…	1	CPSC310	/README.md	oussama	2022-04-12 11:36:06.000000
3	331dbc09c8afaasdfaadsf297879d3fcfdbf2e7e6494fa…	2	CPSC310	/README.md	oussama	2022-04-12 11:37:06.000000
4	asfdadsfasdfadsfasdfadsfads	2	CS50	/README.md	jack	2020-03-12 11:32:06.000000
5	adfafadsfasdf	2	CS50	/README.md	jill	2021-04-12 11:36:06.000000

- *Contributor_of*(**Username: String, Repository_name: String**)

```
SELECT * FROM CONTRIBUTOR_OF
```

	USERNAME	REPOSITORY_NAME
1	eshaq	CPSC304
2	jack	CS50
3	jill	CS50
4	oussama	CPSC304
5	oussama	CPSC310

- *Date_stamp*(**time_stamp: Timestamp**)

```
SELECT * FROM DATE_STAMP
```

	TIME_STAMP
1	2020-03-12 11:32:06.000000
2	2021-04-12 11:36:06.000000
3	2022-04-12 11:32:06.000000
4	2022-04-12 11:36:06.000000
5	2022-04-12 11:37:06.000000

- *FIPPA_Details*(**Country: String**, FIPPA_Compliance:Integer)

```
SELECT * FROM FIPPA_DETAILS
```

	COUNTRY	FIPPA_COMPLIANCE
1	Canada	1
2	USA	0
3	Singapore	0
4	UK	0
5	Russia	0

- *Is_Member_of*(**Username: String, Organization_name: String**)

```
SELECT * FROM IS_MEMBER_OF
```

	USERNAME	ORGANIZATION_NAME
1	eshaq	UBC
2	jack	Harvard
3	jill	Harvard
4	oliver	UBC
5	oussama	UBC

- *Issue*(**Issue_ID: Integer**, Issue_message: String, Owning_board: String)

```
SELECT * FROM ISSUE
```

	ISSUE_ID	ISSUE_MESSAGE	OWNING_BOARD
1	1	Add dataset	Insight UBC
2	2	Improve recursion	Learn Code
3	3	DBMS	Databases
4	4	Perform query	Python
5	5	Trust the natural recursion	Ocaml

- *Line_change*(**Commit_SHA: String, Line_number: Integer**)

```
SELECT * FROM LINE_CHANGE
```

	COMMIT_SHA	LINE_NUMBER
1	331dbc09c8afa4c077f448e4c1ef1e42297879d3fcfdbf2e7e6494fa6d831cec	10
2	331dbc09c8afa4c077f448e4c1ef1e42297879d3fcfdbf2e7e6494fa6d831cec	30
3	331dbc09c8afa4c07fdadfasfadsf297879d3fcfdbf2e7e6494fa6d831cec	50
4	331dbc09c8afaasdfaadsf297879d3fcfdbf2e7e6494fa6d831cec	99
5	asfdadsfasdfadsfasdfadsfads	1

- *Organization*(**Organization_name: String,** Country: String)

```
SELECT * FROM ORGANIZATION
```

	ORGANIZATION_NAME	COUNTRY
1	UBC	Canada
2	SFU	Canada
3	Harvard	USA
4	Oxford	UK
5	NUS	Singapore

- *Project_board*(**Board_title: String**, Repository_name: String)

```
SELECT * FROM PROJECT_BOARD
```

	BOARD_TITLE	REPOSITORY_NAME
1	Insight UBC	CPSC310
2	Learn Code	CPSC304
3	Databases	CS50
4	Python	CMPT120
5	Ocaml	G400

- *Repository*(**Repository_name: String**, Organization_name: String, Date_created: Timestamp)

```
SELECT * FROM REPOSITORY
```

	REPOSITORY_NAME	ORGANIZATION_NAME	DATE_CREATED
1	CPSC304	UBC	2022-04-12 11:32:06.000000
2	CPSC310	UBC	2022-04-12 11:32:06.000000
3	CMPT120	SFU	2022-04-12 11:32:06.000000
4	G400	Oxford	2022-04-12 11:32:06.000000
5	CS50	Harvard	2022-04-12 11:32:06.000000

- *Repository_Object*(**Repository: String**, **File_path: String**, file_extension: String, file_contents: String, file_Size: Integer, Parent_Directory_Path: String)

```
SELECT * FROM REPOSITORY_OBJECT
```

	REPOSITORY	FILE_PATH	FILE_EXTENSION	FILE_CONTENTS	FILE_SIZE	PARENT_DIRECTORY_PATH	PARENT_DIRECTORY_REPOSITORY
1	CPSC304	/	<null>	<null>	32	/	CPSC304
2	CPSC304	/README.md	.md	Adding a line...	32	/	CPSC304
3	CPSC310	/	<null>	<null>	32	/	CPSC310
4	CPSC310	/README.md	.md	# Hello world!	32	/	CPSC310
5	CS50	/	<null>	<null>	32	/	CS50
6	CS50	/README.md	.md	# Hello world!	32	/	CS50

- *User_Account*(**Username: String**, Password_hash: String)

```
SELECT * FROM USER_ACCOUNT
```

	USERNAME	PASSWORD_HASH
1	eshaq	pass123
2	oliver	pass23
3	oussama	password
4	jack	amazing
5	jill	wow

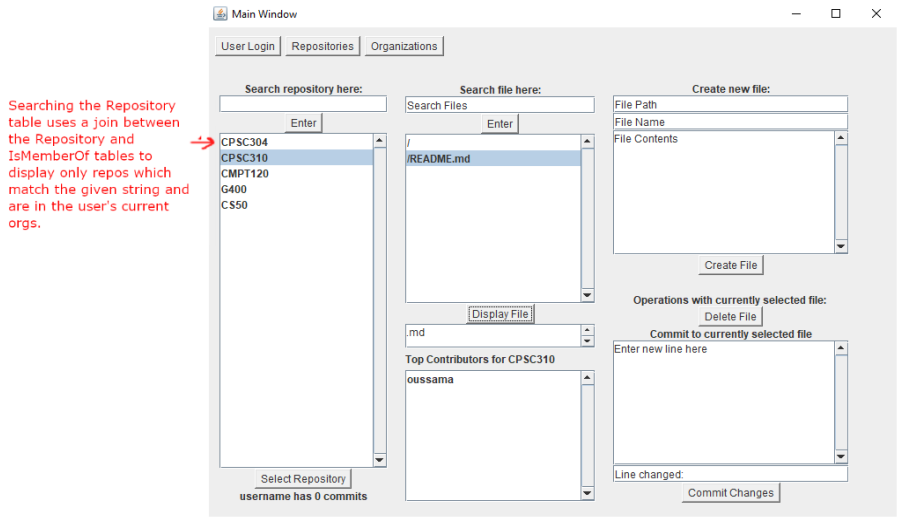
University of British Columbia, Vancouver
Department of Computer Science

List of all queries:

Unless mentioned otherwise, all of the following queries use the Tables and sample data created by the databaseSetup.sql script. The sample data is pictured above in the schema and example tuples section of the document. If the query modifies any tables, the new table will be shown.

- **Join:**

```
"SELECT R.repository_name as repository_name, " +
    "R.organization_name as organization_name, " +
    "R.date_created as date_created" +
"FROM Repository R, IS_MEMBER_OF I " +
"WHERE R.organization_name = I.organization_name " +
    "AND I.username = ? " +
    "AND R.repository_name LIKE ?"
```



This uses the following modified version of the Is_Member_Of *table where the current user is oliox. The Repository table is the default table created by databaseSetup.sql.*

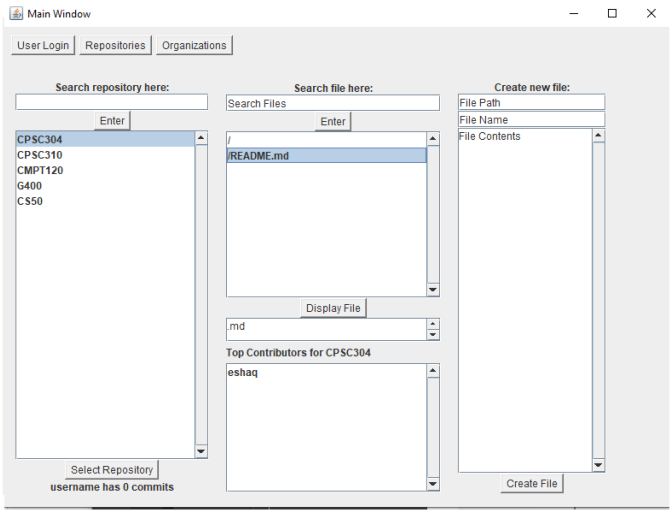
	USERNAME	ORGANIZATION_NAME
1	eshaq	UBC
2	jack	Harvard
3	jill	Harvard
4	oliox	Harvard
5	oliox	NUS
6	oliox	Oxford
7	oliox	SFU
8	oliox	UBC
9	oliver	UBC
10	oussama	UBC
11	username	Harvard
12	username	NUS
13	username	Oxford
14	username	SFU
15	username	UBC

University of British Columbia, Vancouver
Department of Computer Science

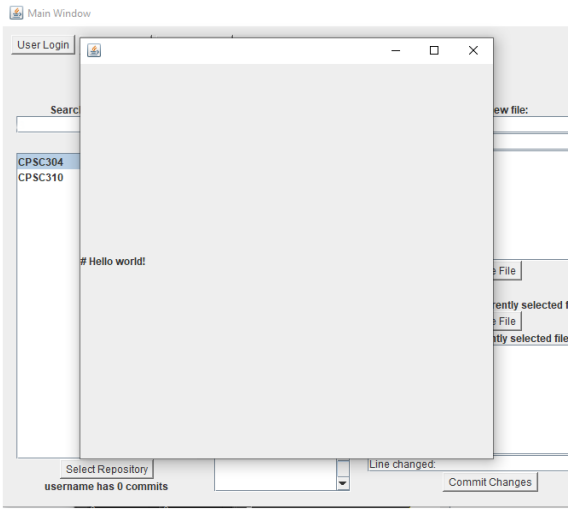
- **Projection:**

```
"SELECT file_contents FROM Repository_Object R WHERE R.Repository = ? and R.file_path = ?"
```

Before:



File being displayed:



Displaying the file involves projecting the file_contents field.

- **Insert operation:**

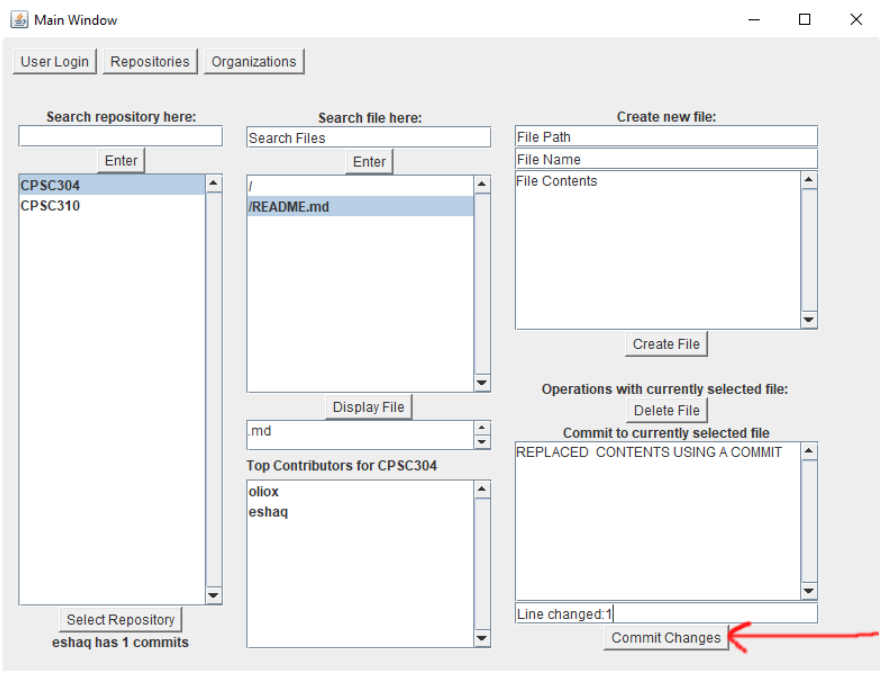
```
"INSERT INTO User_Account VALUES (?, ?)"
```

Here is the User_Account table after the above picture account is signed up:

	USERNAME	PASSWORD_HASH
1	eshaq	pass123
2	oliver	pass23
3	oussama	password
4	jack	amazing
5	jill	wow
6	oliox	password

- **Update operation:**

```
"UPDATE REPOSITORY_OBJECT SET file_contents = ? WHERE repository = ? AND file_path = ?"
```



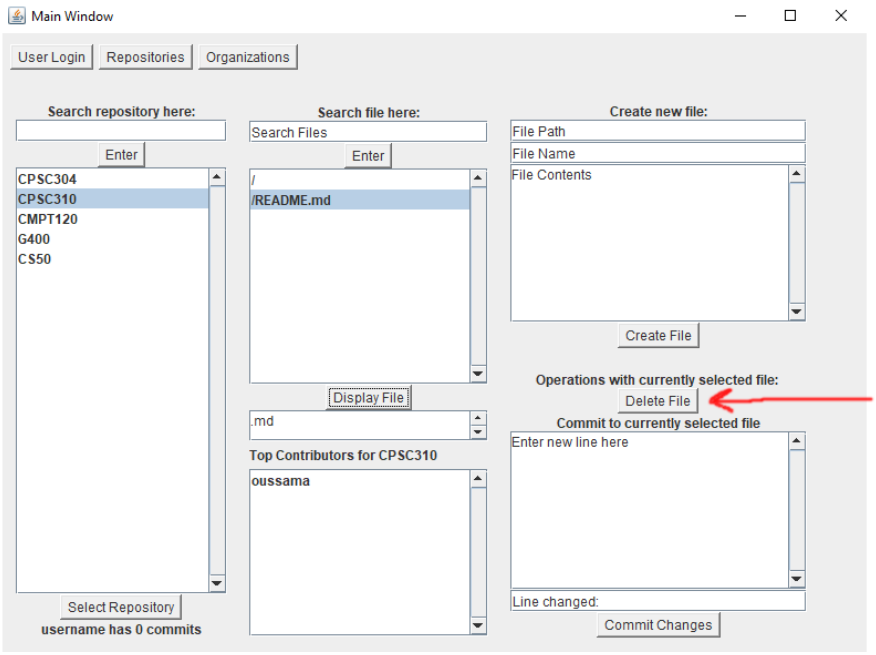
When the user commits changes, part of that process involves an Update operation to update the contents of the file being committed to

Here is the Repository_Object table once a commit has changed the contents of cpsc304/README.md

	REPOSITORY	FILE_PATH	FILE_EXTENSION	FILE_CONTENTS	FILE_SIZE	PARENT_DIRECTORY_PATH	PARENT_DIRECTORY_REPOSITORY
1	CPSC304	/	<null>	<null>	32	/	CPSC304
2	CPSC304	/README.md	.md	REPLACED CONTENTS USING A COMMIT	32	/	CPSC304
3	CPSC310	/	<null>	<null>	32	/	CPSC310
4	CPSC310	/README.md	.md	# Hello world!	32	/	CPSC310
5	CS50	/	<null>	<null>	32	/	CS50
6	CS50	/README.md	.md	# Hello world!	32	/	CS50

- **Deletion:**

```
"DELETE FROM Repository_Object R WHERE R.Repository = ? and R.file_path = ?"
```



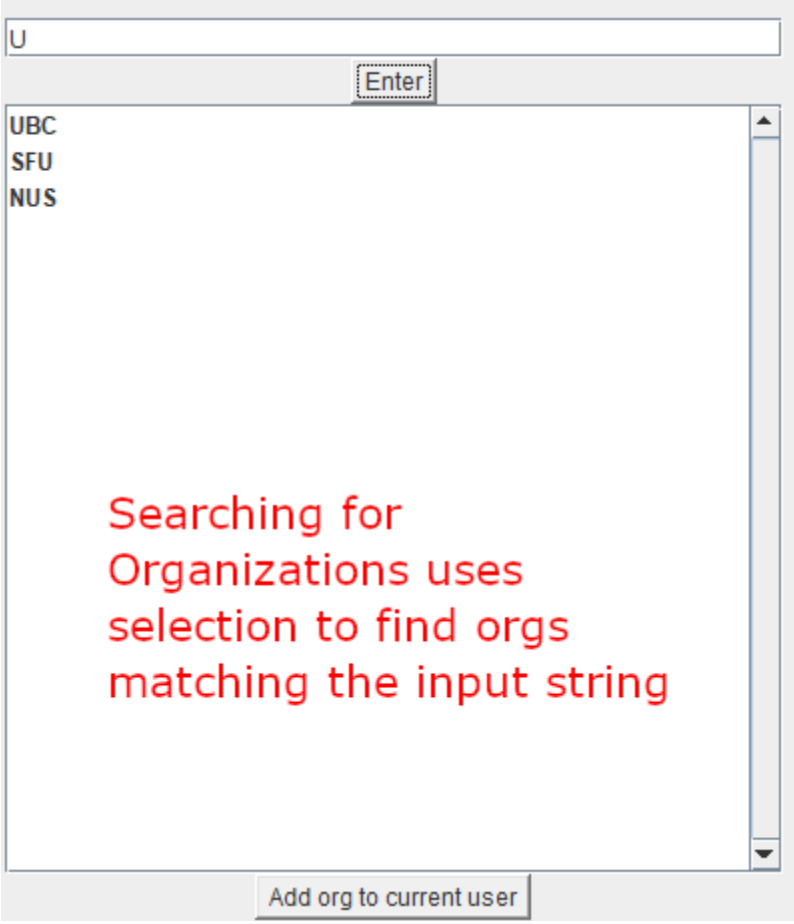
The Delete File button deletes the currently selected file in the Repository_Object table

Here is an example of the Repository_Object table after the file CPSC304/README.md was deleted:

	REPOSITORY	FILE_PATH	FILE_EXTENSION	FILE_CONTENTS	FILE_SIZE	PARENT_DIRECTORY_PATH	PARENT_DIRECTORY_REPOSITORY
1	CPSC304	/	<null>	<null>	32	/	CPSC304
2	CPSC310	/	<null>	<null>	32	/	CPSC310
3	CPSC310	/README.md	.md	# Hello world!	32	/	CPSC310
4	CS50	/	<null>	<null>	32	/	CS50
5	CS50	/README.md	.md	# Hello world!	32	/	CS50

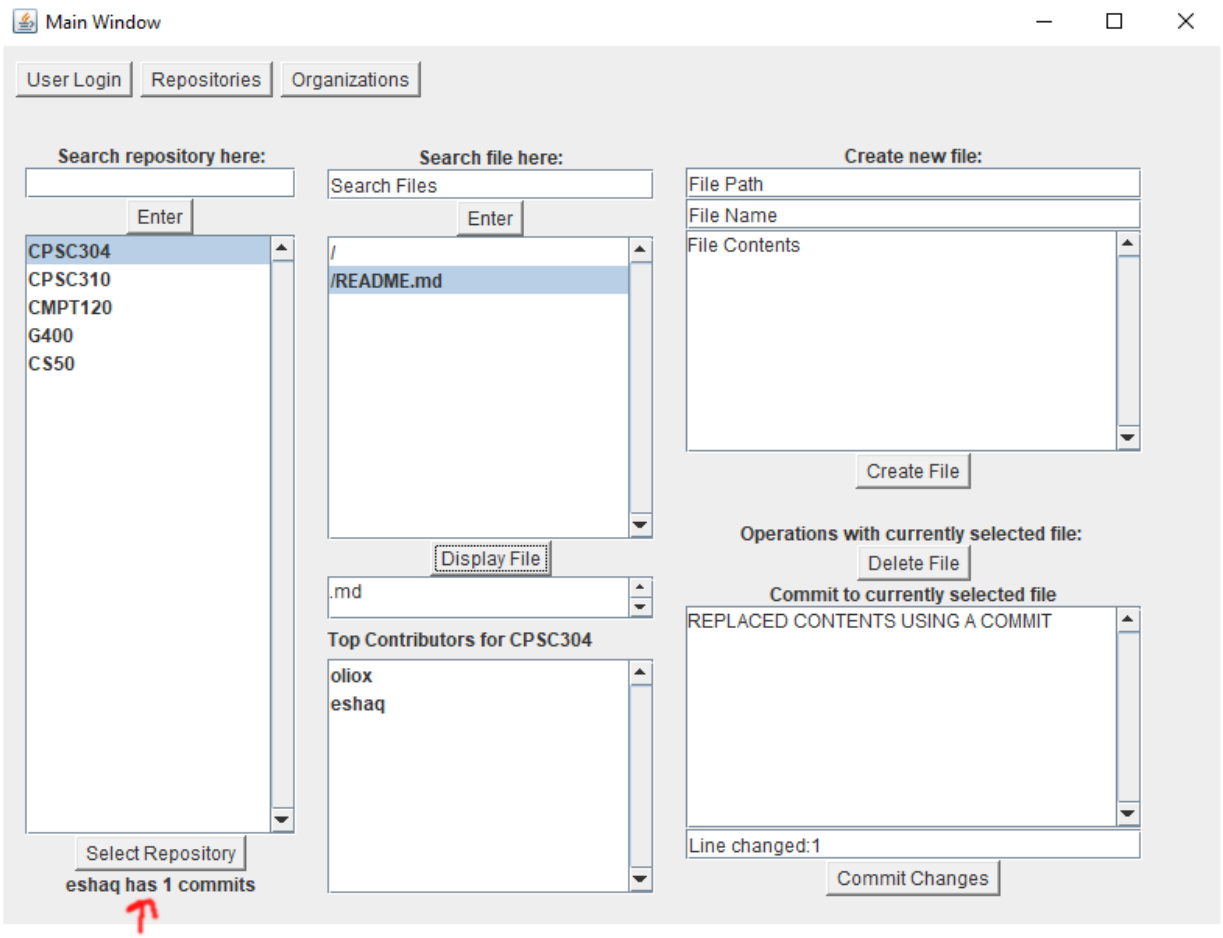
- **Selection:**

```
"SELECT * FROM Organization WHERE organization_name LIKE ?"
```



- **Aggregation with Group By:**

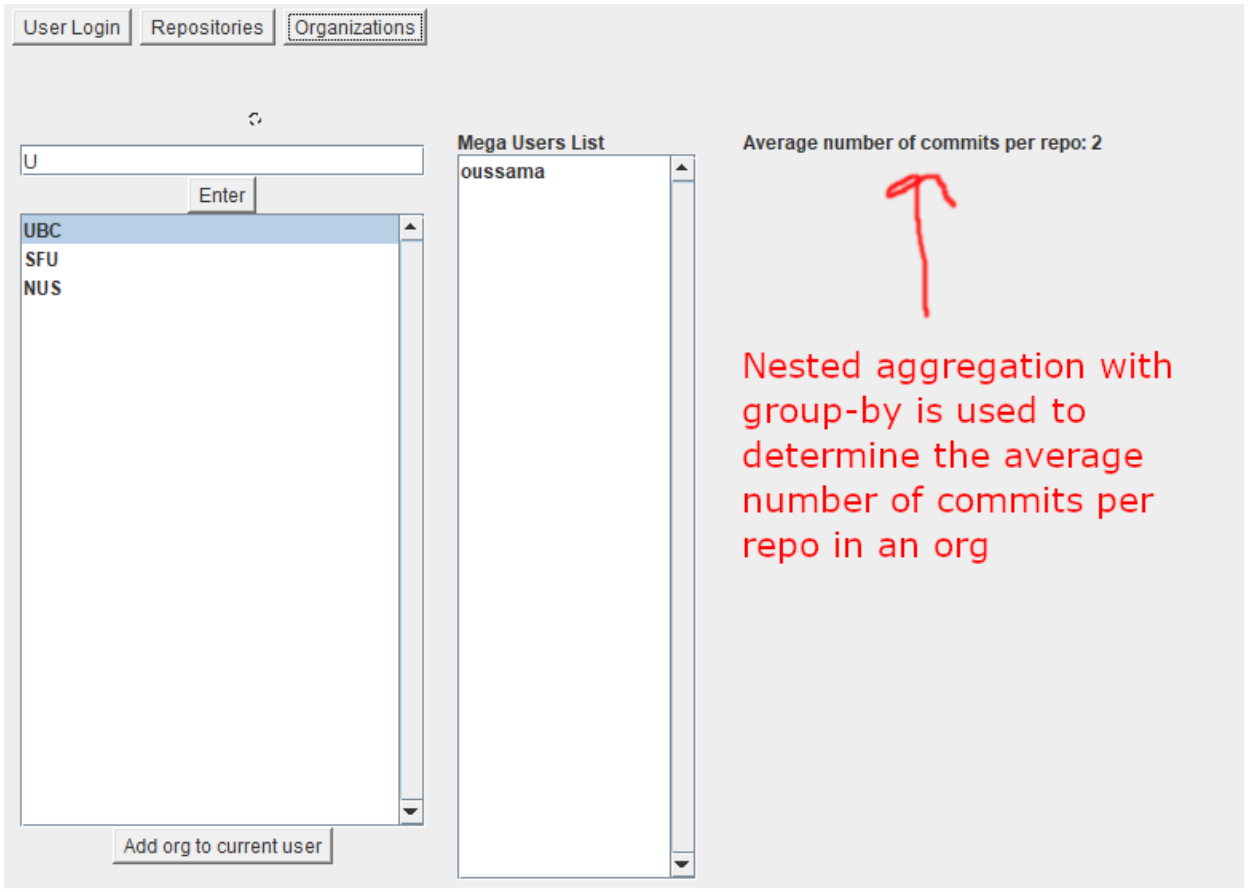
```
"SELECT COUNT(*) as count_val, repository_name FROM Commit " +
"WHERE author = ? GROUP BY repository_name"
```



The User Commit count uses Aggregation and Group By to automatically track and report the number of commits the user has made in the currently selected repository

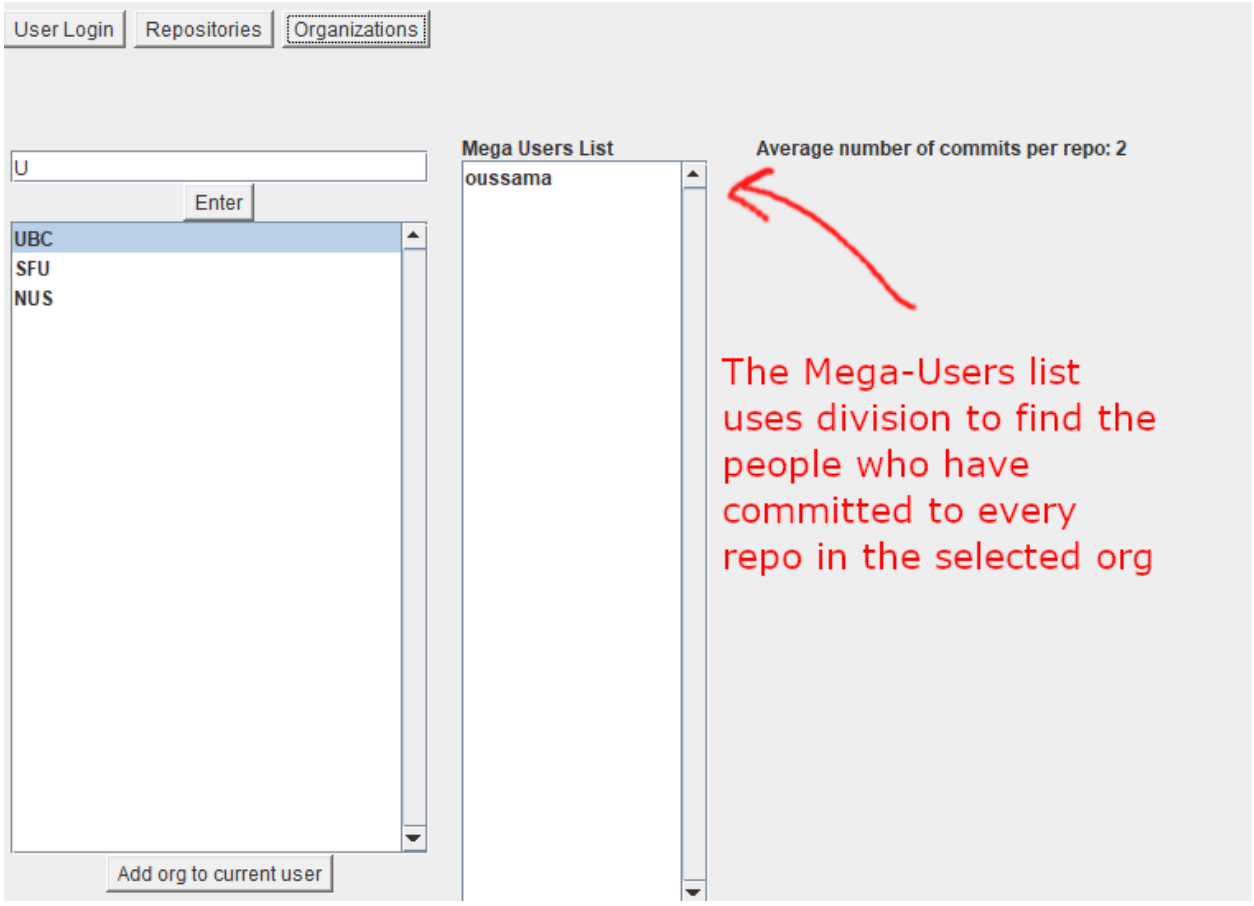
- **Nested aggregation with Group by:**

```
"SELECT AVG(count_val)" +
"FROM (SELECT COUNT(*) as count_val " +
    "FROM Commit C, Repository R " +
    "WHERE C.repository_name = R.repository_name AND R.organization_name = ?"
    "GROUP BY R.repository_name)"
```



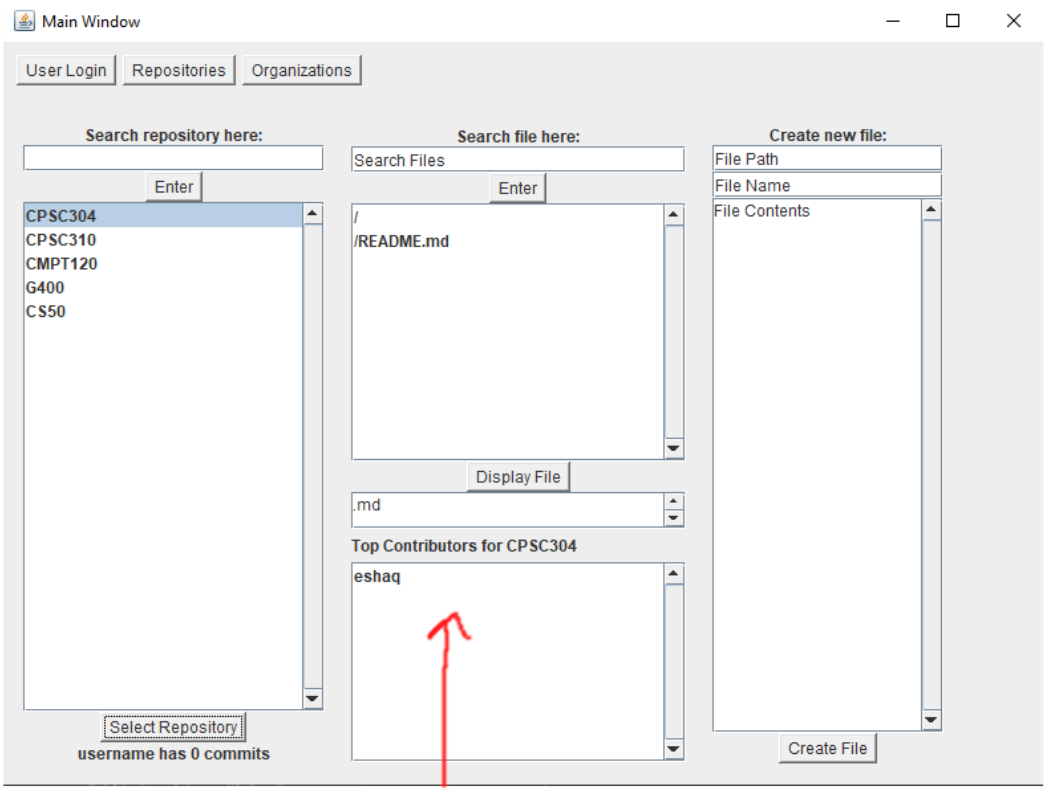
- **Division:**

```
"SELECT U.username " +
"FROM User_Account U " +
"WHERE NOT EXISTS" +
    "((SELECT R.repository_name" +
    "   FROM Repository R " +
    "   WHERE R.organization_name = ?)" +
    "MINUS " +
    "(SELECT C.repository_name " +
    "   FROM Commit C, Repository R" +
    "   WHERE C.author = U.username AND C.repository_name = R.repository_name " +
    "       AND R.organization_name = ?))"
```



- **Aggregation with Having**

```
"SELECT U.username, U.password_hash " +
    "        FROM User_Account U " +
    "        GROUP BY U.username, U.password_hash " +
    "        HAVING (SELECT Count(*) " +
    "                FROM Commit Co " +
    "                WHERE Co.author = U.username AND Co.repository_name = ?)" +
    "               >= " +
    "              (SELECT AVG(counts) " +
    "              FROM ( " +
    "                  SELECT COUNT(*) as counts " +
    "                  FROM Commit Co " +
    "                  WHERE Co.repository_name = ? " +
    "                  GROUP BY Co.author))"
```



The top contributors field uses Aggregation with Having to display all individuals who have made the highest amount of commits to the repository