Faculty of Engineering and Applied Science

SOFE 3700U Data Management

Phase III: Project Report
GitHubStats

Due Date: Friday, Nov 29th, 2019

CRN 43512

Group 23:

Anna Safonov - 100601514

Priyadharshini Ramalingam - 100670614

Pranjal Saloni - 100653360

Umar Qureshi - 100591742

Table of Contents

Abstract

We developed a prototype of a database web application that leverages GitHub REST API to acquire data on the repositories, issues, pull requests, and commits of public GitHub users located in Ontario, Canada, and then uses it to analyze and display visual trends of activity of users for this geographical region.

Introduction

At 40 million users across the world and 100 million repositories, GitHub is one of the most popular version control and project management platforms. Although it provides some general statistics on individual repository pages (new issues, closed issues, pull requests, languages used in the repository, contributors, etc.), currently there is no convenient way to look at real-time commit statistics across repositories of different users at a glance. GitHub does support a well-developed public API to extract information across many repositories for further analysis. For this project, our group will use GitHub API to extract useful statistics on commits for repositories of all public users located in Ontario, Canada as a convenient tool to peek at current GitHub activity in this geographic region.

Goals and Motivations

The goal of this application is to focus on sorting and pulling commit statistics from public repositories of users located in Ontario, Canada from the past year. This data will be analyzed for the following metrics: which months and days of the week and hours of the day have the most commits, most commonly used programming languages, and the categories of software products these commits represent (web applications, system utilities, big data tools, etc.).

Our main motivation for this project is to develop a practical data-driven tool for quantifying trends in software development that will be implemented using tools and technologies covered in this course, such as a database management system, database design process, SQL query language, data flow in RESTful web services etc. Through this, we hope to demonstrate our understanding of the course material and our ability to apply it in a real small-scale solution.

Related Applications

https://github.com/marketplace/circleci

CircleCi is an application that uses Github API for project teamwork. It speeds up the test and delivery cycle without running your own infrastructure by showing workflow status, related jobs with the Insights functionality, and performance trends.

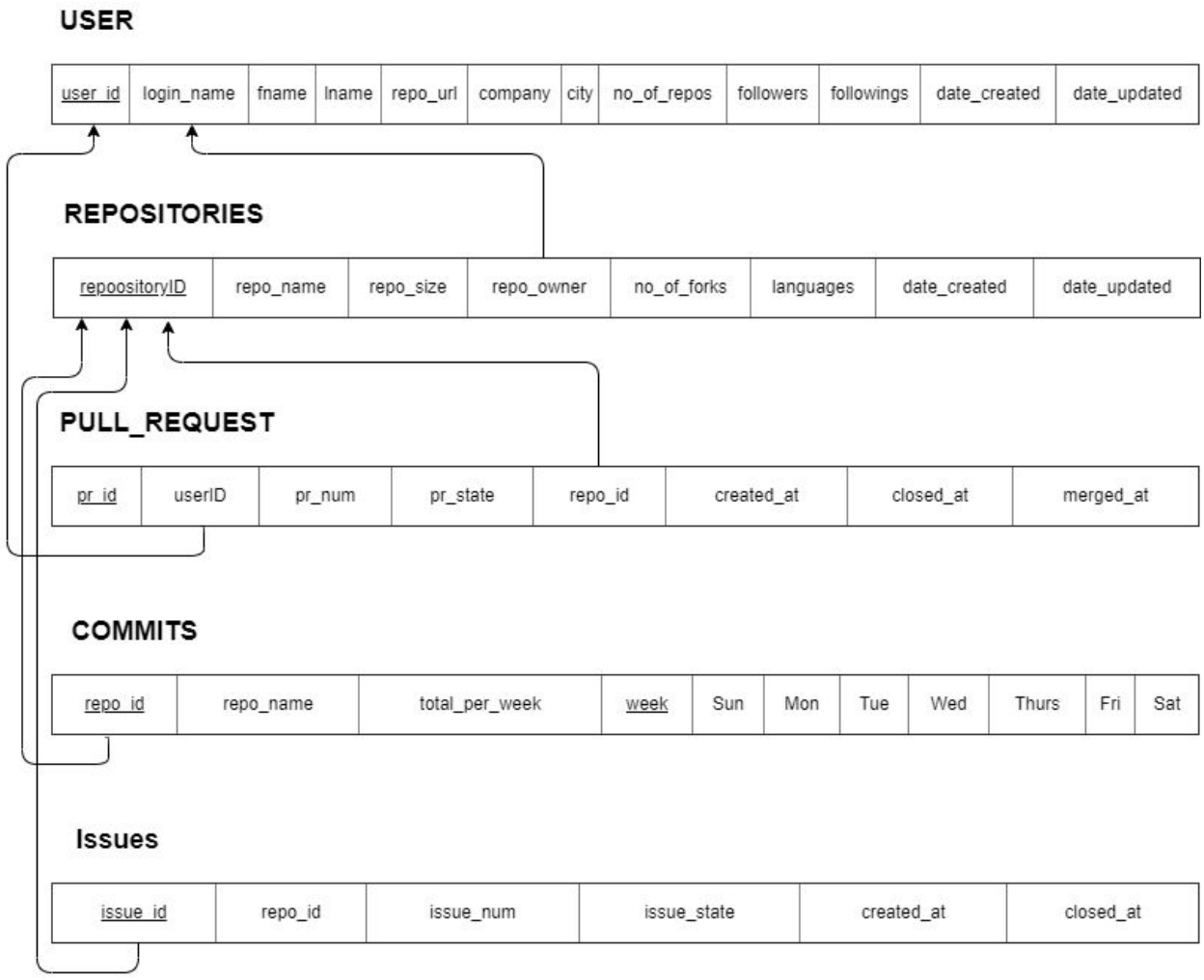
https://github.com/marketplace/zenhub

ZenHub uses Github API and integrates natively with Githubs user interface. It has a Multi-Repo Task Board that lets the team visualize issues and group them in epics, track dependencies and collaborate on product backlogs. Zenhub can also release reports, use the history of reports to detect trends to improve processes, increase team efficiency and measure the value delivered to end-users.

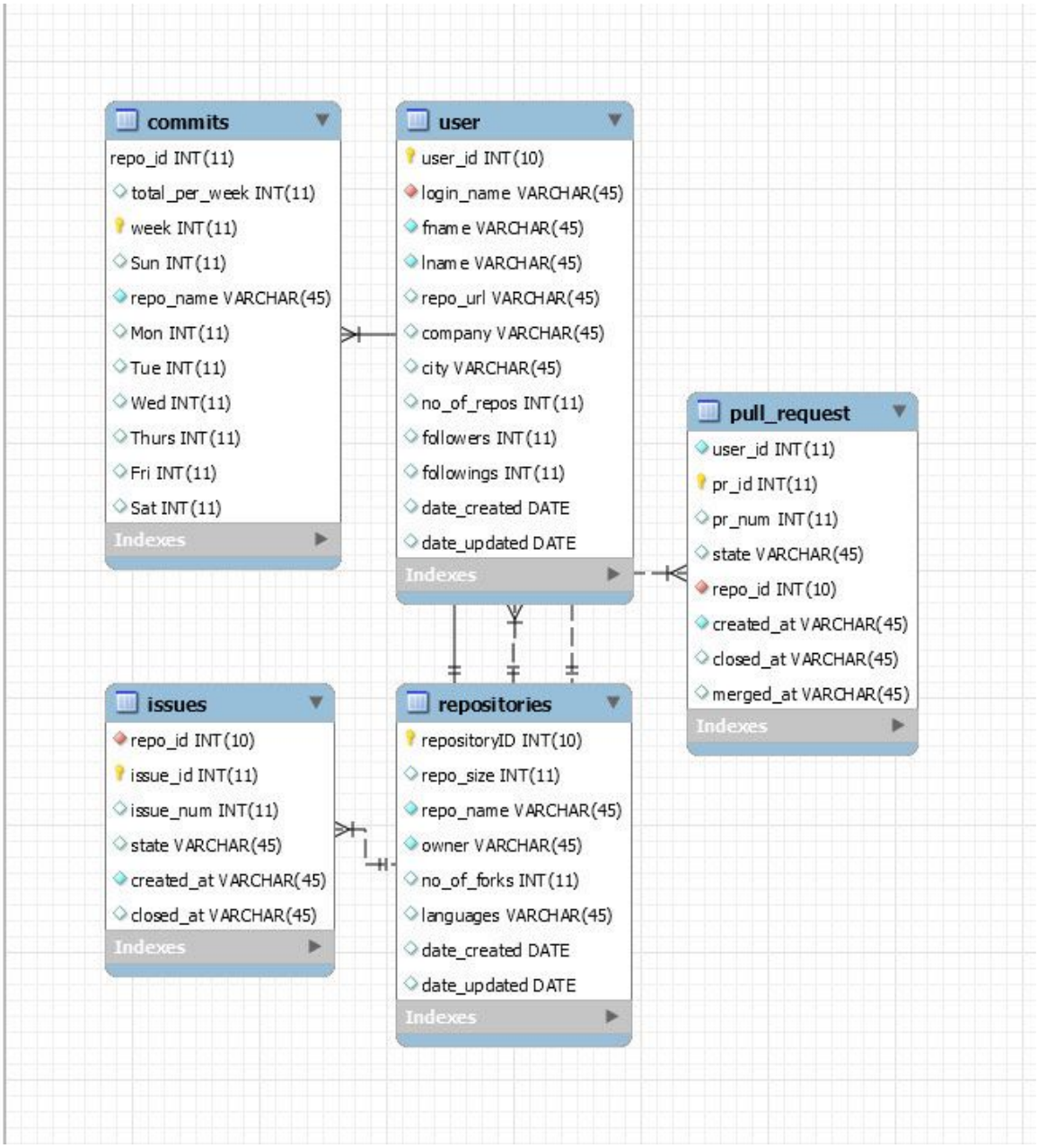
Our project differs from the above applications because it does not focus on the functionality of a single repository. Instead we deliver statistical insights on user activity for many repositories as a quantification tool of trends in development process for various types of software technologies.

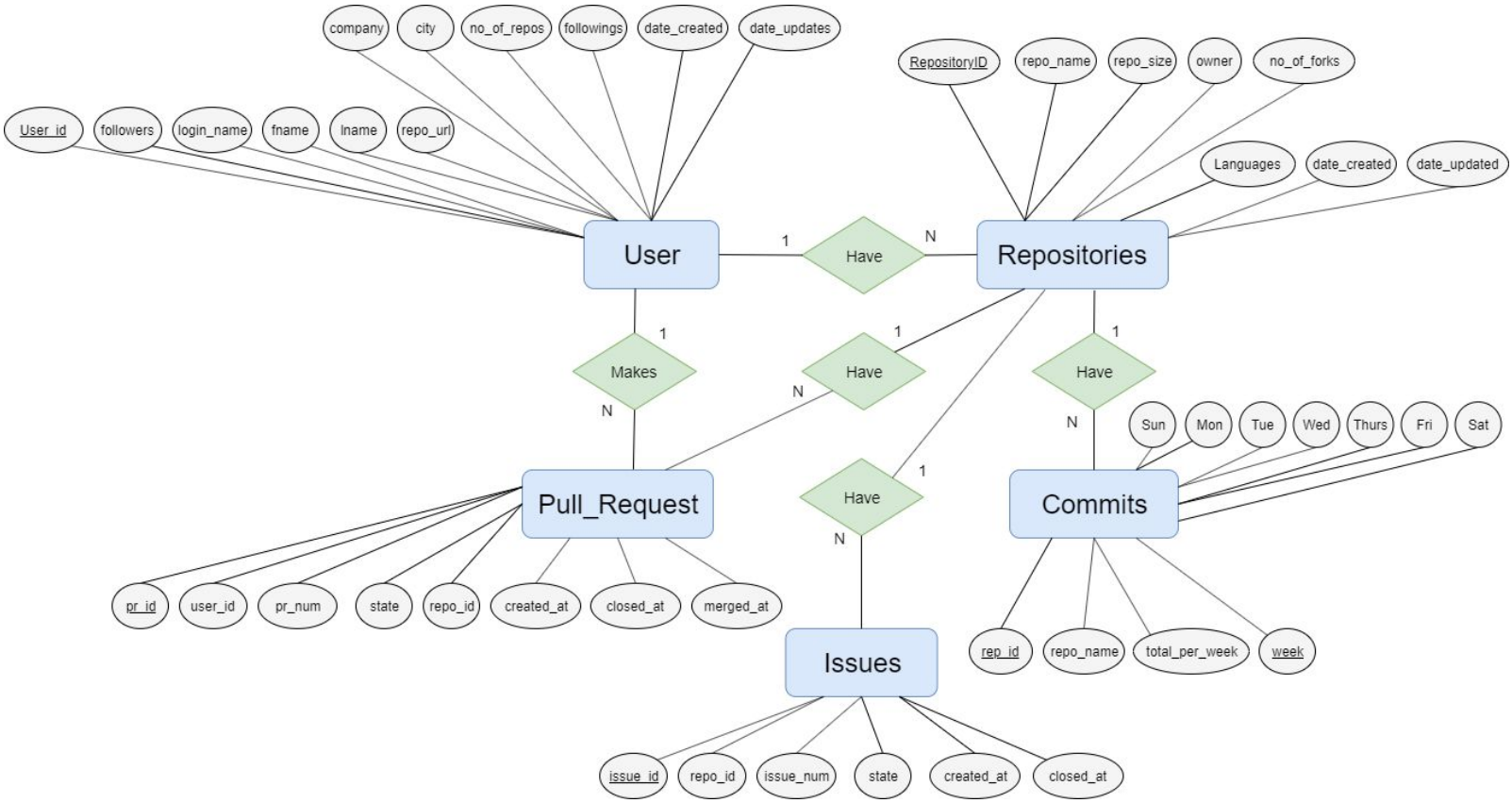
Diagrams

Relational Schema



ER Diagram


commits
repo_id INT(11)
total_per_week INT(11)
week INT(11)
Sun INT(11)
repo_name VARCHAR(45)
Mon INT(11)
Tue INT(11)
Wed INT(11)
Thurs INT(11)
Fri INT(11)
Sat INT(11)
Indexes
user
user_id INT(10)
login_name VARCHAR(45)
fname VARCHAR(45)
lname VARCHAR(45)
repo_url VARCHAR(45)
company VARCHAR(45)
city VARCHAR(45)
no_of_repos INT(11)
followers INT(11)
followings INT(11)
date_created DATE
date_updated DATE
Indexes
pull_request
user_id INT(11)
pr_id INT(11)
pr_num INT(11)
state VARCHAR(45)
repo_id INT(10)
created_at VARCHAR(45)
closed_at VARCHAR(45)
merged_at VARCHAR(45)
Indexes
issues
repo_id INT(10)
issue_id INT(11)
issue_num INT(11)
state VARCHAR(45)
created_at VARCHAR(45)
closed_at VARCHAR(45)
Indexes
repositories
repositoryID INT(10)
repo_size INT(11)
repo_name VARCHAR(45)
owner VARCHAR(45)
no_of_forks INT(11)
languages VARCHAR(45)
date_created DATE
date_updated DATE
Indexes



Views

Sample views that can be rendered using the data pulled from GitHub for this project.

View 1: Join of at least three tables

Displays the number of commits made on a Tuesday

```
SELECT u.login_name, c.Tue

FROM ((repositories as r

INNER JOIN user AS u ON r.repo_owner = u.login_name)

INNER JOIN commits AS c ON r.repositoryID = c.repo_id);
```

Login Name	Tuesday
markstory	1
markstory	0
markstory	0
markstory	0
markstory	0
markstory	0
markstory	1
markstory	0
chartjes	1
chartjes	1
StevenBlack	0
wesbos	0
Redth	0
Redth	1
nayuki	0
nayuki	0
nayuki	0

View 2: Nested queries with ANY or ALL operator and GROUP BY clause

Select users, grouped by last name, that have more followings than all the users living in Toronto.

```
SELECT u.fname, u.lname, u.login_name

FROM user as u

WHERE u.followings > ALL (SELECT followings

                          FROM user

                           WHERE city = 'Toronto')
```

First Name	Last Name	Login Name
Brian R. Bondy		bbondy
Adam	Bell	b3ll
Steven	Black	StevenBlack
Wes	Bos	wesbos
Ethan	Buchman	ebuchman
John	Fish	johnafish
Leask	Wong	Leask
Anthony	Zhang	Uberi

View 3: A correlated nested query

Showing users who have more followers then user with login name ncschonni

```
SELECT u.fname, u.lname
FROM user as u
WHERE u.followers >
        ( SELECT followers
          FROM user
          WHERE login_name = "Pahimar");
```

First Name	Last Name
Jesse	Wilson
Wes	Bos
Nayuki	
Adam	Wathan

View 4: FULL JOIN

```
Select *

FROM user

FULL OUTER JOIN repositories

ON user.login_name = repositories.owner;
```

Above does not work in MySQL due to Outer Joins not being supported. We did this query instead:

```
SELECT *

FROM user

LEFT JOIN repositories ON user.login_name = repositories.repo_owner

UNION

SELECT *

FROM user

RIGHT JOIN repositories ON user.login_name = repositories.repo_owner;
```

Above is just a snippet of the table as it is long, in this snippet it shows the middle of the table where they both have joined. (repo_url is not it repositories table and repositoryID is not in user table)

Last Name	Repository URL	Company	City	no_of_repos	Followers	Followings	Date Created	Date Updated	Repository ID	repo_size	repo_name
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2009-06-20	2019-08-13	231875	1062	cakephp_gesl
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2009-06-20	2019-08-13	231877	46	acl_extras
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2009-07-07	2019-08-13	244842	99	cakephp_mer
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2009-07-07	2019-08-13	244868	4	cakephp_vcar
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2009-12-30	2019-08-13	453427	87	hashgrid
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2010-03-09	2019-11-26	553608	1112	asset_compre
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2010-05-22	2019-08-13	680614	84528	cakephp
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2011-01-18	2019-08-13	1268040	24134	cakephp-docs
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2011-07-11	2019-11-14	2032709	49	dotfiles
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2011-09-04	2019-08-13	2324298	129	cakefest-rabb
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2012-03-03	2019-08-13	3613303	176	CodeSniffer_(
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2012-12-15	2019-08-13	7183688	6389	apigen
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2013-01-22	2019-08-13	7755125	33005	celery
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2013-09-22	2019-08-13	13017950	1097	go-consumer
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2013-09-25	2019-08-13	13094710	3588	beego
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2013-09-25	2019-08-13	13095736	283	beedoc
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2014-01-08	2019-08-13	15742116	1802	ember-training
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2014-02-11	2019-08-13	16717180	139	grunt-es6-mo
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2014-05-06	2019-08-13	19479372	1885	app
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2014-08-01	2019-11-14	22494423	2931	cakephp-perf
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2016-01-27	2016-01-27	50477598	30	flask-pundit
Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2016-02-06	2019-07-23	51221978	84	cakephp-spek

View 5: Nested queries with any of the set operations UNION, EXCEPT, or INTERSECT

Shows repo ids of repos that have a smaller size of PHP and if it has an open issue

```
SELECT r.repositoryID

FROM repositories AS r

WHERE r.repo_size < ALL (SELECT r.repo_size

                         FROM repositories AS r

                         WHERE languages = 'PHP')

EXCEPT

SELECT i.repo_id

From issues AS i

where i.state = "open";
```

Above doesn't work (as mySql doesn't support EXCEPT. So we did this instead. Below shows repo ids of repos that have a smaller size then the PHP repos or shows repo ids that have closed issues.

```
SELECT r.repositoryID
FROM repositories AS r
WHERE r.repo_size < ALL (SELECT r.repo_size
                         FROM repositories AS r
                         WHERE languages = 'PHP')
UNION
SELECT i.repo_id
From issues AS i
where i.state_state = "closed";
```

Repository ID
28240994
42418462
55181998
68455870
78994814
103174822
113510385
147728539
214216163
221256842
680614
1599768
1139082
1564611
231877
1251596

View 6: Show all users with more than 38 repositories

```
SELECT *

FROM user

WHERE no_of_repos > '38'

ORDER BY user_id;
```

User ID	Login Name	First Name	Last Name	Repository URL	Company	City	no_of_repos	Followers	Followings	Date Created	Date Updated
24086	markstory	Mark	Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2008-09-11	2019-11-27
28321	chartjes	Chris	Hartjes	https://api.github.com/users/chartjes/repos		Milton	42	298	1	2008-10-09	2019-11-14
78918	bbolker	Ben	Bolker	https://api.github.com/users/bbolker/repos	McMaster University	Hamilton	112	388	1	2009-04-28	2019-11-22
80144	StevenBlack	Steven	Black	https://api.github.com/users/StevenBlack/repos		Kingston	133	491	156	2009-05-02	2019-11-27
176013	wesbos	Wes	Bos	https://api.github.com/users/wesbos/repos	me	Hamilton	244	17321	32	2010-01-04	2019-10-18
211054	cmoulton	Christina Moulton, iOS Dev		https://api.github.com/users/cmoulton/repos	Teak Mobile Inc	Ottawa	40	297	1	2010-02-26	2019-11-04
233022	Leask	Leask	Wong	https://api.github.com/users/Leask/repos	@Press-One	Ottawa	126	290	95	2010-03-30	2019-10-27
271950	Redth	Jonathan	Dick	https://api.github.com/users/Redth/repos	Xamarin @ Microsoft	Ontario	175	668	4	2010-05-09	2019-11-13
437196	Uberi	Anthony	Zhang	https://api.github.com/users/Uberi/repos	Hypotenuse Labs	Waterloo	102	503	64	2010-10-12	2019-11-18
617994	acabunoc	Abigail Cabunoc Mayes		https://api.github.com/users/acabunoc/repos	Mozilla Foundation	Toronto	113	377	4	2011-02-14	2019-11-05
831718	bbondy	Brian R. Bondy		https://api.github.com/users/bbondy/repos	Brave Software	Ontario	84	603	41	2011-06-06	2019-11-19
911566	gvwilson	Greg	Wilson	https://api.github.com/users/gvwilson/repos	@rstudio	Toronto	68	361	0	2011-07-13	2019-11-07
1961722	b3ll	Adam	Bell	https://api.github.com/users/b3ll/repos	diffractive	London	71	811	92	2012-07-16	2019-11-16
2751621	flar2	Aaron	Segaert	https://api.github.com/users/flar2/repos	EX Solutions Inc.	Southwestern Ontario	63	382	0	2012-11-08	2019-10-27
4323180	adamwathan	Adam	Wathan	https://api.github.com/users/adamwathan/repos		Ontario	89	3640	11	2013-05-02	2019-11-06
4762842	IanDarwin	Ian	Darwin	https://api.github.com/users/IanDarwin/repos		Ontario	106	292	2	2013-06-21	2019-11-26
9373002	mhdawson	Michael	Dawson	https://api.github.com/users/mhdawson/repos	ottawa		114	368	4	2014-10-23	2019-10-29

View 7: View of all users in Toronto

```
SELECT *

FROM user

WHERE city = 'toronto';
```

User ID	Login Name	First Name	Last Name	Repository URL	Company	City	no_of_repos	Followers	Followings	Date Created	Date Updated
24086	markstory	Mark	Story	https://api.github.com/users/markstory/repos	@getsentry	Toronto	69	1133	18	2008-09-11	2019-11-27
617994	acabunoc	Abigail Cabunoc Mayes		https://api.github.com/users/acabunoc/repos	Mozilla Foundation	Toronto	113	377	4	2011-02-14	2019-11-05
672172	nayuki	Nayuki		https://api.github.com/users/nayuki/repos	Project Nayuki	Toronto	25	1695	0	2011-03-16	2019-11-12
911566	gvwilson	Greg	Wilson	https://api.github.com/users/gvwilson/repos	@rstudio	Toronto	68	361	0	2011-07-13	2019-11-07
1505226	thestinger	Daniel	Micay	https://api.github.com/users/thestinger/repos		Toronto	8	670	10	2012-03-06	2019-06-19
19363028	adambcomer	Adam	Comer	https://api.github.com/users/adambcomer/repos	Knowtworthy	Toronto	4	678	8	2016-05-13	2019-11-18

View 8: Show all pull requests made in October

```
SELECT *

FROM pull_request

WHERE (created_at >= DATE '2019-10-01') AND

(closed_at <= DATE '2019-10-31');
```

Repo ID	Issues ID	Issue_num	Issue_state	Created_at	closed_at	merged
25684229	323472350	167	open	99754954	2019-10-01	
5013258	323870465	168	open	99754954	2019-10-02	
17388747	330031354	169	open	99754954	2019-10-19	

View 9: Show all issues for repository ID 494431

```
SELECT *

FROM github_api.issues

WHERE issue_id = '412036';
```

Repo ID	Issues ID	Issue_num	Issue_state	Created_at	closed_at
1564611	412036	280001	closed	2011-04-03	2019-05-04

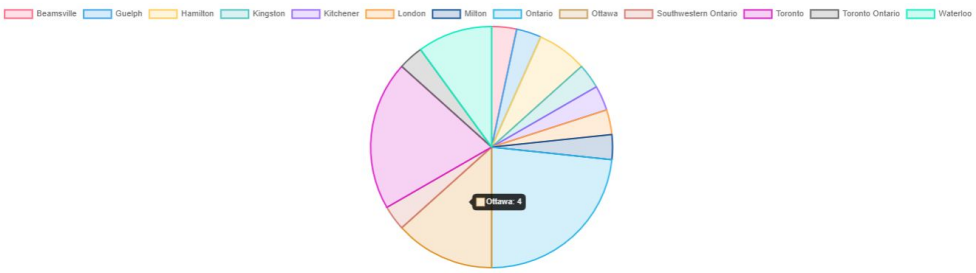
View 10: Show all repositories written in PHP

```
SELECT *

FROM repositories

WHERE languages = 'php';
```

Repository ID	repo_size	repo_name	repo_owner	no_of_forks	Languages	Date Created	Date Updated
231875	1062	cakephp_geshi	markstory	5	PHP	2009-06-20	2019-08-13
231877	46	acl_extras	markstory	54	PHP	2009-06-20	2019-08-13
244842	99	cakephp_menu_component	markstory	15	PHP	2009-07-07	2019-08-13
244868	4	cakephp_vcard	markstory	2	PHP	2009-07-07	2019-08-13
553608	1112	asset_compress	markstory	124	PHP	2010-03-09	2019-11-26
680614	84528	cakephp	markstory	1	PHP	2010-05-22	2019-08-13
2324298	129	cakefest-rabbitmq	markstory	4	PHP	2011-09-04	2019-08-13
2660257	459	building-testable-applications	chartjes	14	PHP	2011-10-27	2019-08-13
3372872	468	ibl-delphi	chartjes	0	PHP	2012-02-07	2013-12-01
3423972	88	FizzBuzz	chartjes	1	PHP	2012-02-12	2014-03-04
3613303	176	CodeSniffer_CakePHP	markstory	3	PHP	2012-03-03	2019-08-13
4342725	200	blog-strategy	chartjes	0	PHP	2012-05-16	2014-03-13
4463830	509	php-tricorder	chartjes	8	PHP	2012-05-27	2019-08-13
5019956	1053	Mink	chartjes	0	PHP	2012-07-13	2015-01-07
5019997	84	MinkSelenium2Driver	chartjes	0	PHP	2012-07-13	2015-01-07
6940801	6894	grumpy-learning	chartjes	2	PHP	2012-11-30	2019-05-22
7183688	6389	apigen	markstory	1	PHP	2012-12-15	2019-08-13
9039050	212	acf-tax	wesbos	1	PHP	2013-03-26	2013-03-27
11349192	164	Roundabout	reinink	0	PHP	2013-07-11	2018-06-19
12267685	112	Atropa	Leask	0	PHP	2013-08-21	2014-02-14
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

City View: Showcases a Pie Chart of which city the Users live

Counts how many times a city occurs in a column

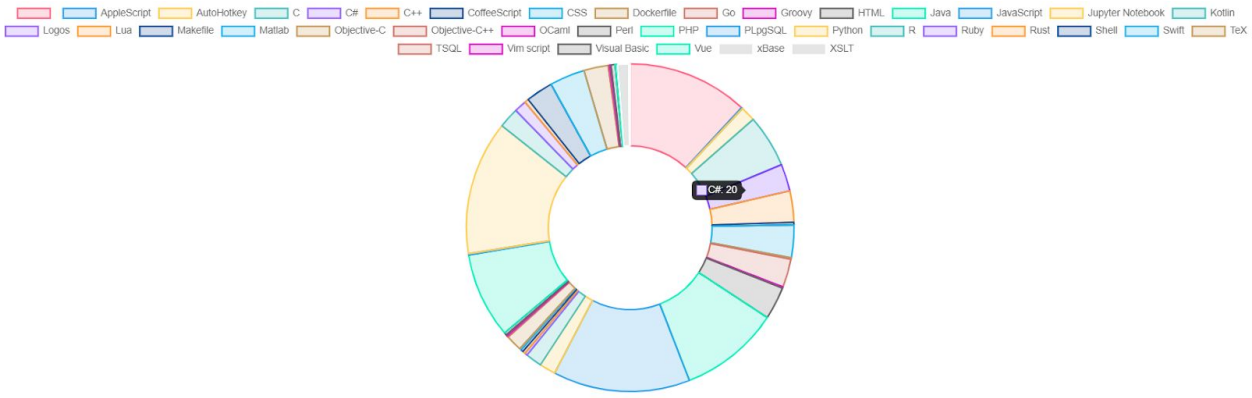


City	total
Beamsville	1
Guelph	1
Hamilton	2
Kingston	1
Kitchener	1
London	1
Milton	1
Ontario	7
Ottawa	4
Southwestern Ontario	1
Toronto	6

Issue Status View: Shows how many issues are open and closed in a bar chart

issue_state	total
closed	6
open	11

Language View: Shows a Doughnut chart of which programs the repositories used



languages	total
	89
AppleScript	1
AutoHotkey	11
C	39
C#	20

Design and Implementations

Our group approached the project by breaking it down in the following stages:

- Developed the database design,
- Retrieved, parsed and sanitized data from Github using GitHub public REST API,
- Created and populated the database in MySQL Workbench,
- Created a presentation web layer and
- Queried the database for statistics to display, etc.

We used GitHub public REST API vs, MySQL and MySQL Workbench, PHP for REST API queries, and data parsing and sanitation, and Bootstrap HTML and CSS for the web application. We extracted GitHub data on commits as it is the most accurate metric of repository activity. Our project is limited to public repositories of users located in Ontario, Canada and their commit activity in the past year.

Our group has chosen the above-mentioned technologies and frameworks because we had team members with experience using these technologies and languages for previous projects, internships, etc. Also, because all of these technologies are commonly used for various business tech solutions and are common tools in software development currently in the industry.

The web application is a graphical tool to help users navigate statistics for commits, repositories, pull request, user profiles, and issues. The main page contains links to each of these tables and each of these pages contains tables with populated data displayed in a graph.

We wanted to use a live and robust API for this project and GitHub REST API proves to be an excellent choice. This API comes with well written documentation, support and a friendly online community. Github is a commonly used tool for repository and project management.

Problems and Limitations

Here we discuss some of the problems and limitations we encountered during the development of this project.

REST API: The first difficulty we encountered was specifying the API endpoint incorrectly, so the data returned was not in a proper JSON format.

Rate Limits: GitHub only returns the first 1000 requests on any API call. Even though our API calls to users in Ontario have over 14K users, less than 10% of that data is available. GitHub imposes these limitations to prevent scraping of information.

In addition to limiting the response to a query, another limitation is the rate limit for API calls. The maximum rate for an unauthenticated application is 60 hits/hour. Due to this constrained, sleep functions had to be introduced into our PHP script to ensure that we do not hit HTTP 403 error due to hitting the rate limit.

Time delays within the script lead to a long execution time required to populate database tables, but unfortunately this was the easiest approach for a prototype for this project.

XAMPP: Xampp was giving many connectivity issues. Had to go through multiple config files and edit them and change the port numbers. When everything was connecting phpMyAdmin would freeze on any action. After researching many users had the same issue with phpMyAdmin freezing. The resolution was to use an older version of Xampp and reconfigure everything again and to let my teammates know the issues and to use another service like Wampp instead.

Hosting with AWS: Our original plan was to host our application through the Amazon Web Services platform. However, correct configuration of this cloud service took longer than expected, so we were left with no time to test our PHP script to populate database and render the web application.

Future Developments

GitHubStats is a flexible, adaptable and an easily-modifiable tool that can be used in a variety of settings, including as a team management tool, for data analytics and visualization, and to quantify team member contributions on large projects.

Further developments for this application include a GitHub statistics mobile app, a better frontend interface for the web application, and customizable functionalities for activity tracking and analytics.

Conclusion

The main purpose of this project was to apply our theoretical database knowledge to a real world application and make a web application that makes it easier for users to access useful information and statistics on GitHub API. We successfully accomplished this by using the tools and techniques learned in this course. For this project, we learned to use MySQL workbench which made it very easy to create tables, ER diagram etc. We also successfully learned how to create relational schemas with associated SQL table commands. Furthermore, we populated our database with sample data where we applied different views that the user would find useful and displayed our project database - through reverse engineering - using an ER diagram. We also learned how APIs work, and how to implement one to return a query as a JSON.

Group contribution

We met every week to work on the project, split the work equally and worked on what each of us were assigned to do before the due date. Everyone had an equal share of work. The project was divided between 4 group members, each with 25% of work load.

Umar Qureshi - Umar worked on the schema and ER diagram for phase 2. He along with Anna also worked on views 1-3. For phase 3, he was responsible for connecting the front end website to the MySQL database with PHP locally. He also worked on parsing live html tables into Json for char.js to make visual representations of the data.

Anna Safonov - For phase 2, Anna worked on populating sample data and alongside with Umar, worked on creating views 3-5. For phase 3, Anna was responsible for populating database, php connections and hosting with AWS.

Pranjal Saloni - Worked on creating the 5 sample tables (repositories, commits, user, pull-request and issues) along with Priya using MySQL workbench. For phase 2, was responsible for working on views 6 -8. Also worked on creating the html web pages, css and js.

Priyadharshini Ramalingam - Alongside with Pranjal, worked on creating the 5 sample tables, worked on views 8 -10 for phase 2. Also, created the html web pages, css and js.

Everyone worked on the report and presentation together.