/\* MySQL Database Code \*/

/\* The code below is formatted in such a way that can be cut and pasted directly into a MySQL database interface to establish an identical set of data to what was used for this research, assuming the same GHTorrent database dump (mysql-2019-06-01) was used (<https://ghtorrent.org/downloads.html>). All code assumes that that database has been set as the default database for the interface or server. Comments and sub-headings are commented out with /\* and \*/ to allow for a full cut and paste into the SQL query interpreter of choose. \*/

/\* User Table Creation and Cleanup \*/

/\* Code block to create clean copy of the default user table without fake or deleted users \*/

CREATE TABLE clean\_users

SELECT \* FROM users

WHERE users.fake = 0

AND users.deleted = 0;

CREATE INDEX id ON clean\_users(id);

/\* Code blocks to create tables of users from Russia, China, the United States, India based on their country\_code properties of ru, cn, us, or in respectively \*/

CREATE TABLE russian\_users

SELECT \* FROM clean\_users

WHERE country\_code = 'ru';

CREATE INDEX id ON russian\_users(id);

CREATE TABLE chinese\_users

SELECT \* FROM clean\_users

WHERE country\_code = 'cn';

CREATE INDEX id ON chinese\_users(id);

CREATE TABLE unitedstates\_users

SELECT \* FROM clean\_users

WHERE country\_code = 'us';

CREATE INDEX id ON unitedstates\_users(id);

CREATE TABLE indian\_users

SELECT \* FROM clean\_users

WHERE country\_code = 'in';

CREATE INDEX id ON indian\_users(id);

/\* User Follower Counts \*/

/\* Code blocks to create follower count tables that include followers for each user from any   
source, for all users and for each of the selected countries. \*/

CREATE TABLE clean\_users\_followers\_count

SELECT followers.user\_id, COUNT(\*) AS follower\_count, clean\_users.login

FROM followers, clean\_users

WHERE clean\_users.id = followers.user\_id

GROUP BY clean\_users.id, clean\_users.login;

CREATE INDEX user\_id ON clean\_users\_followers\_count(user\_id);

CREATE TABLE russian\_followers

SELECT followers.\*

FROM followers, russian\_users

WHERE russian\_users.id = followers.user\_id;

CREATE INDEX user\_id ON russian\_followers(user\_id);

CREATE INDEX follower\_id ON russian\_followers(follower\_id);

CREATE TABLE russian\_followers\_count

SELECT russian\_followers.user\_id, COUNT(\*) AS follower\_count, russian\_users.login

FROM russian\_followers, russian\_users

WHERE russian\_users.id = russian\_followers.user\_id

GROUP BY russian\_users.id, russian\_users.login;

CREATE INDEX user\_id ON russian\_followers\_count(user\_id);

CREATE TABLE chinese\_followers

SELECT followers.\*

FROM followers, chinese\_users

WHERE chinese\_users.id = followers.user\_id;

CREATE INDEX user\_id ON chinese\_followers(user\_id);

CREATE INDEX follower\_id ON chinese\_followers(follower\_id);

CREATE TABLE chinese\_followers\_count

SELECT chinese\_followers.user\_id, COUNT(\*) AS follower\_count, chinese\_users.login

FROM chinese\_followers, chinese\_users

WHERE chinese\_users.id = chinese\_followers.user\_id

GROUP BY chinese\_users.id, chinese\_users.login;

CREATE INDEX user\_id ON chinese\_followers\_count(user\_id);

CREATE TABLE unitedstates\_followers

SELECT followers.\*

FROM followers, unitedstates\_users

WHERE unitedstates\_users.id = followers.user\_id;

CREATE INDEX user\_id ON unitedstates\_followers(user\_id);

CREATE INDEX follower\_id ON unitedstates\_followers(follower\_id);

CREATE TABLE unitedstates\_followers\_count

SELECT unitedstates\_followers.user\_id, COUNT(\*) AS follower\_count, unitedstates\_users.login

FROM unitedstates\_followers, unitedstates\_users

WHERE unitedstates\_users.id = unitedstates\_followers.user\_id

GROUP BY unitedstates\_users.id, unitedstates\_users.login;

CREATE INDEX user\_id ON unitedstates\_followers\_count(user\_id);

CREATE TABLE indian\_followers

SELECT followers.\*

FROM followers, indian\_users

WHERE indian\_users.id = followers.user\_id;

CREATE INDEX user\_id ON indian\_followers(user\_id);

CREATE INDEX follower\_id ON indian\_followers(follower\_id);

CREATE TABLE indian\_followers\_count

SELECT indian\_followers.user\_id, COUNT(\*) AS follower\_count, indian\_users.login

FROM indian\_followers, indian\_users

WHERE indian\_users.id = indian\_followers.user\_id

GROUP BY indian\_users.id, indian\_users.login;

CREATE INDEX user\_id ON indian\_followers\_count(user\_id);

/\* Code blocks to create follower count tables that only include followers for each user that are identified to be within that user’s nation. \*/

CREATE TABLE russian\_to\_russian\_follower\_count

SELECT a.user\_id, COUNT(\*) AS follower\_count

FROM russian\_followers a, russian\_users b

WHERE a.follower\_id = b.id

GROUP BY a.user\_id;

CREATE INDEX user\_id ON russian\_to\_russian\_follower\_count(user\_id);

ALTER TABLE russian\_to\_russian\_follower\_count

ADD(login VARCHAR(255));

UPDATE russian\_to\_russian\_follower\_count

SET login = (SELECT login FROM russian\_users WHERE russian\_to\_russian\_follower\_count.user\_id = russian\_users.id);

CREATE TABLE chinese\_to\_chinese\_follower\_count

SELECT a.user\_id, COUNT(\*) AS follower\_count

FROM chinese\_followers a, chinese\_users b

WHERE a.follower\_id = b.id

GROUP BY a.user\_id;

CREATE INDEX user\_id ON chinese\_to\_chinese\_follower\_count(user\_id);

ALTER TABLE chinese\_to\_chinese\_follower\_count

ADD(login VARCHAR(255));

UPDATE chinese\_to\_chinese\_follower\_count

SET login = (SELECT login FROM chinese\_users WHERE chinese\_to\_chinese\_follower\_count.user\_id = chinese\_users.id);

CREATE TABLE unitedstates\_to\_unitedstates\_follower\_count

SELECT a.user\_id, COUNT(\*) AS follower\_count

FROM unitedstates\_followers a, unitedstates\_users b

WHERE a.follower\_id = b.id

GROUP BY a.user\_id;

CREATE INDEX user\_id ON unitedstates\_to\_unitedstates\_follower\_count(user\_id);

ALTER TABLE unitedstates\_to\_unitedstates\_follower\_count

ADD(login VARCHAR(255));

UPDATE unitedstates\_to\_unitedstates\_follower\_count

SET login = (SELECT login FROM unitedstates\_users WHERE unitedstates\_to\_unitedstates\_follower\_count.user\_id = unitedstates\_users.id);

CREATE TABLE indian\_to\_indian\_follower\_count

SELECT a.user\_id, COUNT(\*) AS follower\_count

FROM indian\_followers a, indian\_users b

WHERE a.follower\_id = b.id

GROUP BY a.user\_id;

CREATE INDEX user\_id ON indian\_to\_indian\_follower\_count(user\_id);

ALTER TABLE indian\_to\_indian\_follower\_count

ADD(login VARCHAR(255));

UPDATE indian\_to\_indian\_follower\_count

SET login = (SELECT login FROM indian\_users WHERE indian\_to\_indian\_follower\_count.user\_id = indian\_users.id);

/\* Repository Table Creation \*/

/\* Code blocks to create separate tables for each selected country containing the repositories owned by the users from that country. These tables are required to create the following national fork and watcher counts and are used later to create the fork counts for influence analysis of repositories owned by users from each nation. \*/

CREATE TABLE russian\_projects

SELECT russian\_users.login, projects.\*

FROM russian\_users, projects

WHERE russian\_users.id = projects.owner\_id;

CREATE INDEX id ON russian\_projects(id);

CREATE INDEX owner\_id ON russian\_projects(owner\_id);

CREATE INDEX forked\_from ON russian\_projects(forked\_from);

CREATE TABLE chinese\_projects

SELECT chinese\_users.login, projects.\*

FROM chinese\_users, projects

WHERE chinese\_users.id = projects.owner\_id;

CREATE INDEX id ON chinese\_projects(id);

CREATE INDEX owner\_id ON chinese\_projects(owner\_id);

CREATE INDEX forked\_from ON chinese\_projects(forked\_from);

CREATE TABLE unitedstates\_projects

SELECT unitedstates\_users.login, projects.\*

FROM unitedstates\_users, projects

WHERE unitedstates\_users.id = projects.owner\_id;

CREATE INDEX id ON unitedstates\_projects(id);

CREATE INDEX owner\_id ON unitedstates\_projects(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_projects(forked\_from);

CREATE TABLE indian\_projects

SELECT indian\_users.login, projects.\*

FROM indian\_users, projects

WHERE indian\_users.id = projects.owner\_id;

CREATE INDEX id ON indian\_projects(id);

CREATE INDEX owner\_id ON indian\_projects(owner\_id);

CREATE INDEX forked\_from ON indian\_projects(forked\_from);

/\* User Fork Counts \*/

/\* Code blocks to create fork count tables that include followers for each repository from any   
source, for all repositories and for each of the selected countries owned repositories. \*/

CREATE TABLE project\_fork\_count

SELECT a.id, COUNT(\*) AS fork\_count, a.owner\_id

FROM projects a, projects b

WHERE a.id = b.forked\_from

GROUP BY a.id, a.owner\_id;

CREATE INDEX id ON project\_fork\_count(id);

CREATE INDEX owner\_id ON project\_fork\_count(owner\_id);

CREATE TABLE russian\_project\_fork\_count

SELECT russian\_projects.id, COUNT(\*) AS fork\_count, russian\_projects.owner\_id

FROM russian\_projects, projects

WHERE russian\_projects.id = projects.forked\_from

GROUP BY russian\_projects.id, russian\_projects.owner\_id;

CREATE INDEX id ON russian\_project\_fork\_count(id);

CREATE INDEX owner\_id ON russian\_project\_fork\_count(owner\_id);

CREATE TABLE chinese\_project\_fork\_count

SELECT chinese\_projects.id, COUNT(\*) AS fork\_count, chinese\_projects.owner\_id

FROM chinese\_projects, projects

WHERE chinese\_projects.id = projects.forked\_from

GROUP BY chinese\_projects.id, chinese\_projects.owner\_id;

CREATE INDEX id ON chinese\_project\_fork\_count(id);

CREATE INDEX owner\_id ON chinese\_project\_fork\_count(owner\_id);

CREATE TABLE unitedstates\_project\_fork\_count

SELECT unitedstates\_projects.id, COUNT(\*) AS fork\_count, unitedstates\_projects.owner\_id

FROM unitedstates\_projects, projects

WHERE unitedstates\_projects.id = projects.forked\_from

GROUP BY unitedstates\_projects.id, unitedstates\_projects.owner\_id;

CREATE INDEX id ON unitedstates\_project\_fork\_count(id);

CREATE INDEX owner\_id ON unitedstates\_project\_fork\_count(owner\_id);

CREATE TABLE indian\_project\_fork\_count

SELECT indian\_projects.id, COUNT(\*) AS fork\_count, indian\_projects.owner\_id

FROM indian\_projects, projects

WHERE indian\_projects.id = projects.forked\_from

GROUP BY indian\_projects.id, indian\_projects.owner\_id;

CREATE INDEX id ON indian\_project\_fork\_count(id);

CREATE INDEX owner\_id ON indian\_project\_fork\_count(owner\_id);

/\* Code blocks to create fork count tables that only include forks for each repository that are identified to have been forked by a user from within the original repository owner’s nation. \*/

CREATE TABLE russian\_to\_russian\_project\_fork\_count

SELECT a.id, COUNT(\*) AS fork\_count, a.owner\_id

FROM russian\_projects a, russian\_projects b

WHERE a.id = b.forked\_from

GROUP BY a.id, a.owner\_id;

CREATE INDEX id ON russian\_to\_russian\_project\_fork\_count(id);

CREATE INDEX owner\_id ON russian\_to\_russian\_project\_fork\_count(owner\_id);

CREATE TABLE chinese\_to\_chinese\_project\_fork\_count

SELECT a.id, COUNT(\*) AS fork\_count, a.owner\_id

FROM chinese\_projects a, chinese\_projects b

WHERE a.id = b.forked\_from

GROUP BY a.id, a.owner\_id;

CREATE INDEX id ON chinese\_to\_chinese\_project\_fork\_count(id);

CREATE INDEX owner\_id ON chinese\_to\_chinese\_project\_fork\_count(owner\_id);

CREATE TABLE unitedstates\_to\_unitedstates\_project\_fork\_count

SELECT a.id, COUNT(\*) AS fork\_count, a.owner\_id

FROM unitedstates\_projects a, unitedstates\_projects b

WHERE a.id = b.forked\_from

GROUP BY a.id, a.owner\_id;

CREATE INDEX id ON unitedstates\_to\_unitedstates\_project\_fork\_count(id);

CREATE INDEX owner\_id ON unitedstates\_to\_unitedstates\_project\_fork\_count(owner\_id);

CREATE TABLE indian\_to\_indian\_project\_fork\_count

SELECT a.id, COUNT(\*) AS fork\_count, a.owner\_id

FROM indian\_projects a, indian\_projects b

WHERE a.id = b.forked\_from

GROUP BY a.id, a.owner\_id;

CREATE INDEX id ON indian\_to\_indian\_project\_fork\_count(id);

CREATE INDEX owner\_id ON indian\_to\_indian\_project\_fork\_count(owner\_id);

/\* User Watcher (Star) Counts \*/

/\* Code blocks to create watcher (star) count tables that include watchers for each repository from any source, for all repositories and for each of the selected countries owned repositories. \*/

CREATE TABLE project\_watcher\_count

SELECT projects.id, COUNT(\*) AS watcher\_count, projects.owner\_id

FROM projects, watchers

WHERE projects.id = watchers.repo\_id

GROUP BY projects.id, projects.owner\_id;

CREATE INDEX id ON project\_watcher\_count(id);

CREATE INDEX owner\_id ON project\_watcher\_count(owner\_id);

CREATE TABLE russian\_project\_watcher\_count

SELECT russian\_projects.id, COUNT(\*) AS watcher\_count, russian\_projects.owner\_id

FROM russian\_projects, watchers

WHERE russian\_projects.id = watchers.repo\_id

GROUP BY russian\_projects.id, russian\_projects.owner\_id;

CREATE INDEX id ON russian\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON russian\_project\_watcher\_count(owner\_id);

CREATE TABLE chinese\_project\_watcher\_count

SELECT chinese\_projects.id, COUNT(\*) AS watcher\_count, chinese\_projects.owner\_id

FROM chinese\_projects, watchers

WHERE chinese\_projects.id = watchers.repo\_id

GROUP BY chinese\_projects.id, chinese\_projects.owner\_id;

CREATE INDEX id ON chinese\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON chinese\_project\_watcher\_count(owner\_id);

CREATE TABLE unitedstates\_project\_watcher\_count

SELECT unitedstates\_projects.id, COUNT(\*) AS watcher\_count, unitedstates\_projects.owner\_id

FROM unitedstates\_projects, watchers

WHERE unitedstates\_projects.id = watchers.repo\_id

GROUP BY unitedstates\_projects.id, unitedstates\_projects.owner\_id;

CREATE INDEX id ON unitedstates\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON unitedstates\_project\_watcher\_count(owner\_id);

CREATE TABLE indian\_project\_watcher\_count

SELECT indian\_projects.id, COUNT(\*) AS watcher\_count, indian\_projects.owner\_id

FROM indian\_projects, watchers

WHERE indian\_projects.id = watchers.repo\_id

GROUP BY indian\_projects.id, indian\_projects.owner\_id;

CREATE INDEX id ON indian\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON indian\_project\_watcher\_count(owner\_id);

/\* Code blocks to create watcher (star) count tables that only include watchers for each repository that are identified to have been starred by a user from within the original repository owner’s nation. To do this, a truncated watchers table is created first for each country, which is then used for the subsequent watcher count query. \*/

CREATE TABLE russian\_watchers

SELECT watchers.\*

FROM watchers, russian\_users

WHERE watchers.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers(repo\_id);

CREATE INDEX user\_id ON russian\_watchers(user\_id);

CREATE TABLE russian\_to\_russian\_project\_watcher\_count

SELECT russian\_projects.id, COUNT(\*) AS watcher\_count, russian\_projects.owner\_id

FROM russian\_projects, russian\_watchers

WHERE russian\_projects.id = russian\_watchers.repo\_id

GROUP BY russian\_projects.id, russian\_projects.owner\_id;

CREATE INDEX id ON russian\_to\_russian\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON russian\_to\_russian\_project\_watcher\_count(owner\_id);

CREATE TABLE chinese\_watchers

SELECT watchers.\*

FROM watchers, chinese\_users

WHERE watchers.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers(repo\_id);

CREATE INDEX user\_id ON chinese\_watchers(user\_id);

CREATE TABLE chinese\_to\_chinese\_project\_watcher\_count

SELECT chinese\_projects.id, COUNT(\*) AS watcher\_count, chinese\_projects.owner\_id

FROM chinese\_projects, chinese\_watchers

WHERE chinese\_projects.id = chinese\_watchers.repo\_id

GROUP BY chinese\_projects.id, chinese\_projects.owner\_id;

CREATE INDEX id ON chinese\_to\_chinese\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON chinese\_to\_chinese\_project\_watcher\_count(owner\_id);

CREATE TABLE unitedstates\_watchers

SELECT watchers.\*

FROM watchers, unitedstates\_users

WHERE watchers.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers(repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers(user\_id);

CREATE TABLE unitedstates\_to\_unitedstates\_project\_watcher\_count

SELECT unitedstates\_projects.id, COUNT(\*) AS watcher\_count, unitedstates\_projects.owner\_id

FROM unitedstates\_projects, unitedstates\_watchers

WHERE unitedstates\_projects.id = unitedstates\_watchers.repo\_id

GROUP BY unitedstates\_projects.id, unitedstates\_projects.owner\_id;

CREATE INDEX id ON unitedstates\_to\_unitedstates\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON unitedstates\_to\_unitedstates\_project\_watcher\_count(owner\_id);

CREATE TABLE indian\_watchers

SELECT watchers.\*

FROM watchers, indian\_users

WHERE watchers.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers(repo\_id);

CREATE INDEX user\_id ON indian\_watchers(user\_id);

CREATE TABLE indian\_to\_indian\_project\_watcher\_count

SELECT indian\_projects.id, COUNT(\*) AS watcher\_count, indian\_projects.owner\_id

FROM indian\_projects, indian\_watchers

WHERE indian\_projects.id = indian\_watchers.repo\_id

GROUP BY indian\_projects.id, indian\_projects.owner\_id;

CREATE INDEX id ON indian\_to\_indian\_project\_watcher\_count(id);

CREATE INDEX owner\_id ON indian\_to\_indian\_project\_watcher\_count(owner\_id);

/\* Create Combined User Tables for Analysis \*/

/\* Code blocks to combine the follower, fork and, watcher count data into the user tables for later ease of analysis. Five combined tables result, one for all users on the platform and one for each selected country of study. \*/

ALTER TABLE clean\_users

ADD(follower\_count bigint(21), fork\_count bigint(21), watcher\_count bigint(21));

UPDATE clean\_users

SET clean\_users.follower\_count =

(SELECT clean\_users\_followers\_count.follower\_count

FROM clean\_users\_followers\_count

WHERE clean\_users.id = clean\_users\_followers\_count.user\_id);

UPDATE clean\_users

SET clean\_users.fork\_count =

(SELECT SUM(project\_fork\_count.fork\_count)

FROM project\_fork\_count

WHERE clean\_users.id = project\_fork\_count.owner\_id);

UPDATE clean\_users

SET clean\_users.watcher\_count =

(SELECT SUM(project\_watcher\_count.watcher\_count)

FROM project\_watcher\_count

WHERE clean\_users.id = project\_watcher\_count.owner\_id);

ALTER TABLE russian\_users

ADD(follower\_count bigint(21), russian\_follower\_count bigint(21),

fork\_count bigint(21), russian\_fork\_count bigint(21),

watcher\_count bigint(21), russian\_watcher\_count bigint(21));

UPDATE russian\_users

SET russian\_users.follower\_count =

(SELECT russian\_followers\_count.follower\_count

FROM russian\_followers\_count

WHERE russian\_users.id = russian\_followers\_count.user\_id);

UPDATE russian\_users

SET russian\_users.russian\_follower\_count =

(SELECT russian\_to\_russian\_follower\_count.follower\_count

FROM russian\_to\_russian\_follower\_count

WHERE russian\_users.id = russian\_to\_russian\_follower\_count.user\_id);

UPDATE russian\_users

SET russian\_users.fork\_count =

(SELECT SUM(russian\_project\_fork\_count.fork\_count)

FROM russian\_project\_fork\_count

WHERE russian\_users.id = russian\_project\_fork\_count.owner\_id);

UPDATE russian\_users

SET russian\_users.russian\_fork\_count =

(SELECT SUM(russian\_to\_russian\_project\_fork\_count.fork\_count)

FROM russian\_to\_russian\_project\_fork\_count

WHERE russian\_users.id = russian\_to\_russian\_project\_fork\_count.owner\_id);

UPDATE russian\_users

SET russian\_users.watcher\_count =

(SELECT SUM(russian\_project\_watcher\_count.watcher\_count)

FROM russian\_project\_watcher\_count

WHERE russian\_users.id = russian\_project\_watcher\_count.owner\_id);

UPDATE russian\_users

SET russian\_users.russian\_watcher\_count =

(SELECT SUM(russian\_to\_russian\_project\_watcher\_count.watcher\_count)

FROM russian\_to\_russian\_project\_watcher\_count

WHERE russian\_users.id = russian\_to\_russian\_project\_watcher\_count.owner\_id);

ALTER TABLE chinese\_users

ADD(follower\_count bigint(21), chinese\_follower\_count bigint(21),

fork\_count bigint(21), chinese\_fork\_count bigint(21),

watcher\_count bigint(21), chinese\_watcher\_count bigint(21));

UPDATE chinese\_users

SET chinese\_users.follower\_count =

(SELECT chinese\_followers\_count.follower\_count

FROM chinese\_followers\_count

WHERE chinese\_users.id = chinese\_followers\_count.user\_id);

UPDATE chinese\_users

SET chinese\_users.chinese\_follower\_count =

(SELECT chinese\_to\_chinese\_follower\_count.follower\_count

FROM chinese\_to\_chinese\_follower\_count

WHERE chinese\_users.id = chinese\_to\_chinese\_follower\_count.user\_id);

UPDATE chinese\_users

SET chinese\_users.fork\_count =

(SELECT SUM(chinese\_project\_fork\_count.fork\_count)

FROM chinese\_project\_fork\_count

WHERE chinese\_users.id = chinese\_project\_fork\_count.owner\_id);

UPDATE chinese\_users

SET chinese\_users.chinese\_fork\_count =

(SELECT SUM(chinese\_to\_chinese\_project\_fork\_count.fork\_count)

FROM chinese\_to\_chinese\_project\_fork\_count

WHERE chinese\_users.id = chinese\_to\_chinese\_project\_fork\_count.owner\_id);

UPDATE chinese\_users

SET chinese\_users.watcher\_count =

(SELECT SUM(chinese\_project\_watcher\_count.watcher\_count)

FROM chinese\_project\_watcher\_count

WHERE chinese\_users.id = chinese\_project\_watcher\_count.owner\_id);

UPDATE chinese\_users

SET chinese\_users.chinese\_watcher\_count =

(SELECT SUM(chinese\_to\_chinese\_project\_watcher\_count.watcher\_count)

FROM chinese\_to\_chinese\_project\_watcher\_count

WHERE chinese\_users.id = chinese\_to\_chinese\_project\_watcher\_count.owner\_id);

ALTER TABLE unitedstates\_users

ADD(follower\_count bigint(21), unitedstates\_follower\_count bigint(21),

fork\_count bigint(21), unitedstates\_fork\_count bigint(21),

watcher\_count bigint(21), unitedstates\_watcher\_count bigint(21));

UPDATE unitedstates\_users

SET unitedstates\_users.follower\_count =

(SELECT unitedstates\_followers\_count.follower\_count

FROM unitedstates\_followers\_count

WHERE unitedstates\_users.id = unitedstates\_followers\_count.user\_id);

UPDATE unitedstates\_users

SET unitedstates\_users.unitedstates\_follower\_count =

(SELECT unitedstates\_to\_unitedstates\_follower\_count.follower\_count

FROM unitedstates\_to\_unitedstates\_follower\_count

WHERE unitedstates\_users.id = unitedstates\_to\_unitedstates\_follower\_count.user\_id);

UPDATE unitedstates\_users

SET unitedstates\_users.fork\_count =

(SELECT SUM(unitedstates\_project\_fork\_count.fork\_count)

FROM unitedstates\_project\_fork\_count

WHERE unitedstates\_users.id = unitedstates\_project\_fork\_count.owner\_id);

UPDATE unitedstates\_users

SET unitedstates\_users.unitedstates\_fork\_count =

(SELECT SUM(unitedstates\_to\_unitedstates\_project\_fork\_count.fork\_count)

FROM unitedstates\_to\_unitedstates\_project\_fork\_count

WHERE unitedstates\_users.id = unitedstates\_to\_unitedstates\_project\_fork\_count.owner\_id);

UPDATE unitedstates\_users

SET unitedstates\_users.watcher\_count =

(SELECT SUM(unitedstates\_project\_watcher\_count.watcher\_count)

FROM unitedstates\_project\_watcher\_count

WHERE unitedstates\_users.id = unitedstates\_project\_watcher\_count.owner\_id);

UPDATE unitedstates\_users

SET unitedstates\_users.unitedstates\_watcher\_count =

(SELECT SUM(unitedstates\_to\_unitedstates\_project\_watcher\_count.watcher\_count)

FROM unitedstates\_to\_unitedstates\_project\_watcher\_count

WHERE unitedstates\_users.id = unitedstates\_to\_unitedstates\_project\_watcher\_count.owner\_id);

ALTER TABLE indian\_users

ADD(follower\_count bigint(21), indian\_follower\_count bigint(21),

fork\_count bigint(21), indian\_fork\_count bigint(21),

watcher\_count bigint(21), indian\_watcher\_count bigint(21));

UPDATE indian\_users

SET indian\_users.follower\_count =

(SELECT indian\_followers\_count.follower\_count

FROM indian\_followers\_count

WHERE indian\_users.id = indian\_followers\_count.user\_id);

UPDATE indian\_users

SET indian\_users.indian\_follower\_count =

(SELECT indian\_to\_indian\_follower\_count.follower\_count

FROM indian\_to\_indian\_follower\_count

WHERE indian\_users.id = indian\_to\_indian\_follower\_count.user\_id);

UPDATE indian\_users

SET indian\_users.fork\_count =

(SELECT SUM(indian\_project\_fork\_count.fork\_count)

FROM indian\_project\_fork\_count

WHERE indian\_users.id = indian\_project\_fork\_count.owner\_id);

UPDATE indian\_users

SET indian\_users.indian\_fork\_count =

(SELECT SUM(indian\_to\_indian\_project\_fork\_count.fork\_count)

FROM indian\_to\_indian\_project\_fork\_count

WHERE indian\_users.id = indian\_to\_indian\_project\_fork\_count.owner\_id);

UPDATE indian\_users

SET indian\_users.watcher\_count =

(SELECT SUM(indian\_project\_watcher\_count.watcher\_count)

FROM indian\_project\_watcher\_count

WHERE indian\_users.id = indian\_project\_watcher\_count.owner\_id);

UPDATE indian\_users

SET indian\_users.indian\_watcher\_count =

(SELECT SUM(indian\_to\_indian\_project\_watcher\_count.watcher\_count)

FROM indian\_to\_indian\_project\_watcher\_count

WHERE indian\_users.id = indian\_to\_indian\_project\_watcher\_count.owner\_id);

/\* Code blocks to set null values to 0. These may take some time\*/

UPDATE ghtorrent\_restore.clean\_users

SET follower\_count = 0

WHERE follower\_count IS NULL;

UPDATE ghtorrent\_restore.clean\_users

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE ghtorrent\_restore.clean\_users

SET watcher\_count = 0

WHERE watcher\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET follower\_count = 0

WHERE follower\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET russian\_follower\_count = 0

WHERE russian\_follower\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET russian\_fork\_count = 0

WHERE russian\_fork\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET watcher\_count = 0

WHERE watcher\_count IS NULL;

UPDATE ghtorrent\_restore.russian\_users

SET russian\_watcher\_count = 0

WHERE russian\_watcher\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET follower\_count = 0

WHERE follower\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET chinese\_follower\_count = 0

WHERE chinese\_follower\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET chinese\_fork\_count = 0

WHERE chinese\_fork\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET watcher\_count = 0

WHERE watcher\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET chinese\_watcher\_count = 0

WHERE chinese\_watcher\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET follower\_count = 0

WHERE follower\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET unitedstates\_follower\_count = 0

WHERE unitedstates\_follower\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET unitedstates\_fork\_count = 0

WHERE unitedstates\_fork\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET watcher\_count = 0

WHERE watcher\_count IS NULL;

UPDATE ghtorrent\_restore.unitedstates\_users

SET unitedstates\_watcher\_count = 0

WHERE unitedstates\_watcher\_count IS NULL;

UPDATE ghtorrent\_restore.chinese\_users

SET follower\_count = 0

WHERE follower\_count IS NULL;

UPDATE ghtorrent\_restore.indian\_users

SET indian\_follower\_count = 0

WHERE indian\_follower\_count IS NULL;

UPDATE ghtorrent\_restore.indian\_users

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE ghtorrent\_restore.indian\_users

SET indian\_fork\_count = 0

WHERE indian\_fork\_count IS NULL;

UPDATE ghtorrent\_restore.indian\_users

SET watcher\_count = 0

WHERE watcher\_count IS NULL;

UPDATE ghtorrent\_restore.indian\_users

SET indian\_watcher\_count = 0

WHERE indian\_watcher\_count IS NULL;

/\* Date Limited Repository Star (Watcher) Table Creation \*/

/\* Code blocks to create monthly star tables. Creates six tables in one-month intervals from December 2, 2018 through the database end date, June 1, 2019. These tables are used to form the star tables that are used to create the star counts in the next step. The first six code blocks create date limited star tables for stars from all sources \*/

CREATE TABLE watchers\_june2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_june2019 (user\_id);

CREATE TABLE watchers\_may2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_may2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_may2019 (user\_id);

CREATE TABLE watchers\_april2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_april2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_april2019 (user\_id);

CREATE TABLE watchers\_march2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_march2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_march2019 (user\_id);

CREATE TABLE watchers\_february2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_february2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_february2019 (user\_id);

CREATE TABLE watchers\_january2019

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_january2019 (repo\_id);

CREATE INDEX user\_id ON watchers\_january2019 (user\_id);

/\* Code blocks to create date limited star tables for stars from national sources \*/

CREATE TABLE russian\_watchers\_june2019

SELECT watchers\_june2019.\*

FROM watchers\_june2019, russian\_users

WHERE watchers\_june2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_june2019 (user\_id);

CREATE TABLE russian\_watchers\_may2019

SELECT watchers\_may2019.\*

FROM watchers\_may2019, russian\_users

WHERE watchers\_may2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_may2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_may2019 (user\_id);

CREATE TABLE russian\_watchers\_april2019

SELECT watchers\_april2019.\*

FROM watchers\_april2019, russian\_users

WHERE watchers\_april2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_april2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_april2019 (user\_id);

CREATE TABLE russian\_watchers\_march2019

SELECT watchers\_march2019.\*

FROM watchers\_march2019, russian\_users

WHERE watchers\_march2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_march2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_march2019 (user\_id);

CREATE TABLE russian\_watchers\_february2019

SELECT watchers\_february2019.\*

FROM watchers\_february2019, russian\_users

WHERE watchers\_february2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_february2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_february2019 (user\_id);

CREATE TABLE russian\_watchers\_january2019

SELECT watchers\_january2019.\*

FROM watchers\_january2019, russian\_users

WHERE watchers\_january2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_january2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_january2019 (user\_id);

CREATE TABLE russian\_watchers\_june2019

SELECT watchers\_june2019.\*

FROM watchers\_june2019, russian\_users

WHERE watchers\_june2019.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_june2019 (user\_id);

CREATE TABLE chinese\_watchers\_june2019

SELECT watchers\_june2019.\*

FROM watchers\_june2019, chinese\_users

WHERE watchers\_june2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_june2019 (user\_id);

CREATE TABLE chinese\_watchers\_may2019

SELECT watchers\_may2019.\*

FROM watchers\_may2019, chinese\_users

WHERE watchers\_may2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_may2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_may2019 (user\_id);

CREATE TABLE chinese\_watchers\_april2019

SELECT watchers\_april2019.\*

FROM watchers\_april2019, chinese\_users

WHERE watchers\_april2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_april2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_april2019 (user\_id);

CREATE TABLE chinese\_watchers\_march2019

SELECT watchers\_march2019.\*

FROM watchers\_march2019, chinese\_users

WHERE watchers\_march2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_march2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_march2019 (user\_id);

CREATE TABLE chinese\_watchers\_february2019

SELECT watchers\_february2019.\*

FROM watchers\_february2019, chinese\_users

WHERE watchers\_february2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_february2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_february2019 (user\_id);

CREATE TABLE chinese\_watchers\_january2019

SELECT watchers\_january2019.\*

FROM watchers\_january2019, chinese\_users

WHERE watchers\_january2019.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_january2019 (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_january2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_june2019

SELECT watchers\_june2019.\*

FROM watchers\_june2019, unitedstates\_users

WHERE watchers\_june2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_june2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_may2019

SELECT watchers\_may2019.\*

FROM watchers\_may2019, unitedstates\_users

WHERE watchers\_may2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_may2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_may2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_april2019

SELECT watchers\_april2019.\*

FROM watchers\_april2019, unitedstates\_users

WHERE watchers\_april2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_april2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_april2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_march2019

SELECT watchers\_march2019.\*

FROM watchers\_march2019, unitedstates\_users

WHERE watchers\_march2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_march2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_march2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_february2019

SELECT watchers\_february2019.\*

FROM watchers\_february2019, unitedstates\_users

WHERE watchers\_february2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_february2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_february2019 (user\_id);

CREATE TABLE unitedstates\_watchers\_january2019

SELECT watchers\_january2019.\*

FROM watchers\_january2019, unitedstates\_users

WHERE watchers\_january2019.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_january2019 (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_january2019 (user\_id);

CREATE TABLE indian\_watchers\_june2019

SELECT watchers\_june2019.\*

FROM watchers\_june2019, indian\_users

WHERE watchers\_june2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_june2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_june2019 (user\_id);

CREATE TABLE indian\_watchers\_may2019

SELECT watchers\_may2019.\*

FROM watchers\_may2019, indian\_users

WHERE watchers\_may2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_may2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_may2019 (user\_id);

CREATE TABLE indian\_watchers\_april2019

SELECT watchers\_april2019.\*

FROM watchers\_april2019, indian\_users

WHERE watchers\_april2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_april2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_april2019 (user\_id);

CREATE TABLE indian\_watchers\_march2019

SELECT watchers\_march2019.\*

FROM watchers\_march2019, indian\_users

WHERE watchers\_march2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_march2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_march2019 (user\_id);

CREATE TABLE indian\_watchers\_february2019

SELECT watchers\_february2019.\*

FROM watchers\_february2019, indian\_users

WHERE watchers\_february2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_february2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_february2019 (user\_id);

CREATE TABLE indian\_watchers\_january2019

SELECT watchers\_january2019.\*

FROM watchers\_january2019, indian\_users

WHERE watchers\_january2019.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_january2019 (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_january2019 (user\_id);

/\* Date Limited Star Count Table Creation \*/

/\* Code blocks to create counts of the number of stars for repositories on a monthly basis. The first six code blocks create tables of forks from all user sources. \*/

CREATE TABLE watchers\_june2019\_count

SELECT watchers\_june2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_june2019

GROUP BY watchers\_june2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_june2019\_count(repo\_id);

CREATE TABLE watchers\_may2019\_count

SELECT watchers\_may2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_may2019

GROUP BY watchers\_may2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_may2019\_count(repo\_id);

CREATE TABLE watchers\_april2019\_count

SELECT watchers\_april2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_april2019

GROUP BY watchers\_april2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_april2019\_count(repo\_id);

CREATE TABLE watchers\_march2019\_count

SELECT watchers\_march2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_march2019

GROUP BY watchers\_march2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_march2019\_count(repo\_id);

CREATE TABLE watchers\_february2019\_count

SELECT watchers\_february2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_february2019

GROUP BY watchers\_february2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_february2019\_count(repo\_id);

CREATE TABLE watchers\_january2019\_count

SELECT watchers\_january2019.repo\_id, COUNT(\*) AS star\_count

FROM watchers\_january2019

GROUP BY watchers\_january2019.repo\_id;

CREATE INDEX repo\_id ON watchers\_january2019\_count(repo\_id);

/\* Code blocks to create date limited star tables for stars from national sources \*/

CREATE TABLE russian\_watchers\_june2019\_count

SELECT russian\_watchers\_june2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_june2019

GROUP BY russian\_watchers\_june2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_june2019\_count(repo\_id);

CREATE TABLE russian\_watchers\_may2019\_count

SELECT russian\_watchers\_may2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_may2019

GROUP BY russian\_watchers\_may2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_may2019\_count(repo\_id);

CREATE TABLE russian\_watchers\_april2019\_count

SELECT russian\_watchers\_april2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_april2019

GROUP BY russian\_watchers\_april2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_april2019\_count(repo\_id);

CREATE TABLE russian\_watchers\_march2019\_count

SELECT russian\_watchers\_march2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_march2019

GROUP BY russian\_watchers\_march2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_march2019\_count(repo\_id);

CREATE TABLE russian\_watchers\_february2019\_count

SELECT russian\_watchers\_february2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_february2019

GROUP BY russian\_watchers\_february2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_february2019\_count(repo\_id);

CREATE TABLE russian\_watchers\_january2019\_count

SELECT russian\_watchers\_january2019.repo\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_january2019

GROUP BY russian\_watchers\_january2019.repo\_id;

CREATE INDEX repo\_id ON russian\_watchers\_january2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_june2019\_count

SELECT chinese\_watchers\_june2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_june2019

GROUP BY chinese\_watchers\_june2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_june2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_may2019\_count

SELECT chinese\_watchers\_may2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_may2019

GROUP BY chinese\_watchers\_may2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_may2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_april2019\_count

SELECT chinese\_watchers\_april2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_april2019

GROUP BY chinese\_watchers\_april2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_april2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_march2019\_count

SELECT chinese\_watchers\_march2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_march2019

GROUP BY chinese\_watchers\_march2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_march2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_february2019\_count

SELECT chinese\_watchers\_february2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_february2019

GROUP BY chinese\_watchers\_february2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_february2019\_count(repo\_id);

CREATE TABLE chinese\_watchers\_january2019\_count

SELECT chinese\_watchers\_january2019.repo\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_january2019

GROUP BY chinese\_watchers\_january2019.repo\_id;

CREATE INDEX repo\_id ON chinese\_watchers\_january2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_june2019\_count

SELECT unitedstates\_watchers\_june2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_june2019

GROUP BY unitedstates\_watchers\_june2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_june2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_may2019\_count

SELECT unitedstates\_watchers\_may2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_may2019

GROUP BY unitedstates\_watchers\_may2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_may2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_april2019\_count

SELECT unitedstates\_watchers\_april2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_april2019

GROUP BY unitedstates\_watchers\_april2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_april2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_march2019\_count

SELECT unitedstates\_watchers\_march2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_march2019

GROUP BY unitedstates\_watchers\_march2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_march2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_february2019\_count

SELECT unitedstates\_watchers\_february2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_february2019

GROUP BY unitedstates\_watchers\_february2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_february2019\_count(repo\_id);

CREATE TABLE unitedstates\_watchers\_january2019\_count

SELECT unitedstates\_watchers\_january2019.repo\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_january2019

GROUP BY unitedstates\_watchers\_january2019.repo\_id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_january2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_june2019\_count

SELECT indian\_watchers\_june2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_june2019

GROUP BY indian\_watchers\_june2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_june2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_may2019\_count

SELECT indian\_watchers\_may2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_may2019

GROUP BY indian\_watchers\_may2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_may2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_april2019\_count

SELECT indian\_watchers\_april2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_april2019

GROUP BY indian\_watchers\_april2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_april2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_march2019\_count

SELECT indian\_watchers\_march2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_march2019

GROUP BY indian\_watchers\_march2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_march2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_february2019\_count

SELECT indian\_watchers\_february2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_february2019

GROUP BY indian\_watchers\_february2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_february2019\_count(repo\_id);

CREATE TABLE indian\_watchers\_january2019\_count

SELECT indian\_watchers\_january2019.repo\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_january2019

GROUP BY indian\_watchers\_january2019.repo\_id;

CREATE INDEX repo\_id ON indian\_watchers\_january2019\_count(repo\_id);

/\* Date Limited Repository Fork Table Creation \*/

/\* Code blocks to create monthly fork tables. Creates six tables in one-month intervals from December 2, 2018 through the database end date, June 1, 2019. These tables are used to form the fork tables are used to create the fork counts in the next step. The first six code blocks create date limited fork tables for forks from all sources \*/

CREATE TABLE forks\_june2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_june2019(id);

CREATE INDEX owner\_id ON forks\_june2019(owner\_id);

CREATE INDEX forked\_from ON forks\_june2019(forked\_from);

CREATE TABLE forks\_may2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_may2019(id);

CREATE INDEX owner\_id ON forks\_may2019(owner\_id);

CREATE INDEX forked\_from ON forks\_may2019(forked\_from);

CREATE TABLE forks\_april2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_april2019(id);

CREATE INDEX owner\_id ON forks\_april2019(owner\_id);

CREATE INDEX forked\_from ON forks\_april2019(forked\_from);

CREATE TABLE forks\_march2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_march2019(id);

CREATE INDEX owner\_id ON forks\_march2019(owner\_id);

CREATE INDEX forked\_from ON forks\_march2019(forked\_from);

CREATE TABLE forks\_february2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_february2019(id);

CREATE INDEX owner\_id ON forks\_february2019(owner\_id);

CREATE INDEX forked\_from ON forks\_february2019(forked\_from);

CREATE TABLE forks\_january2019

SELECT projects.\*

FROM projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON forks\_january2019(id);

CREATE INDEX owner\_id ON forks\_january2019(owner\_id);

CREATE INDEX forked\_from ON forks\_january2019(forked\_from);

/\* Code blocks to create date limited fork tables for forks from national sources \*/

CREATE TABLE russian\_forks\_june2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_june2019(id);

CREATE INDEX owner\_id ON russian\_forks\_june2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_june2019(forked\_from);

CREATE TABLE russian\_forks\_may2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_may2019(id);

CREATE INDEX owner\_id ON russian\_forks\_may2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_may2019(forked\_from);

CREATE TABLE russian\_forks\_april2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_april2019(id);

CREATE INDEX owner\_id ON russian\_forks\_april2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_april2019(forked\_from);

CREATE TABLE russian\_forks\_march2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_march2019(id);

CREATE INDEX owner\_id ON russian\_forks\_march2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_march2019(forked\_from);

CREATE TABLE russian\_forks\_february2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_february2019(id);

CREATE INDEX owner\_id ON russian\_forks\_february2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_february2019(forked\_from);

CREATE TABLE russian\_forks\_january2019

SELECT russian\_projects.\*

FROM russian\_projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON russian\_forks\_january2019(id);

CREATE INDEX owner\_id ON russian\_forks\_january2019(owner\_id);

CREATE INDEX forked\_from ON russian\_forks\_january2019(forked\_from);

CREATE TABLE chinese\_forks\_june2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_june2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_june2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_june2019(forked\_from);

CREATE TABLE chinese\_forks\_may2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_may2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_may2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_may2019(forked\_from);

CREATE TABLE chinese\_forks\_april2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_april2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_april2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_april2019(forked\_from);

CREATE TABLE chinese\_forks\_march2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_march2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_march2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_march2019(forked\_from);

CREATE TABLE chinese\_forks\_february2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_february2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_february2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_february2019(forked\_from);

CREATE TABLE chinese\_forks\_january2019

SELECT chinese\_projects.\*

FROM chinese\_projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON chinese\_forks\_january2019(id);

CREATE INDEX owner\_id ON chinese\_forks\_january2019(owner\_id);

CREATE INDEX forked\_from ON chinese\_forks\_january2019(forked\_from);

CREATE TABLE unitedstates\_forks\_june2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_june2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_june2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_june2019(forked\_from);

CREATE TABLE unitedstates\_forks\_may2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_may2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_may2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_may2019(forked\_from);

CREATE TABLE unitedstates\_forks\_april2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_april2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_april2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_april2019(forked\_from);

CREATE TABLE unitedstates\_forks\_march2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_march2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_march2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_march2019(forked\_from);

CREATE TABLE unitedstates\_forks\_february2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_february2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_february2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_february2019(forked\_from);

CREATE TABLE unitedstates\_forks\_january2019

SELECT unitedstates\_projects.\*

FROM unitedstates\_projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON unitedstates\_forks\_january2019(id);

CREATE INDEX owner\_id ON unitedstates\_forks\_january2019(owner\_id);

CREATE INDEX forked\_from ON unitedstates\_forks\_january2019(forked\_from);

CREATE TABLE indian\_forks\_june2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2019-05-02' AS DATE) AND CAST('2019-06-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_june2019(id);

CREATE INDEX owner\_id ON indian\_forks\_june2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_june2019(forked\_from);

CREATE TABLE indian\_forks\_may2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2019-04-02' AS DATE) AND CAST('2019-05-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_may2019(id);

CREATE INDEX owner\_id ON indian\_forks\_may2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_may2019(forked\_from);

CREATE TABLE indian\_forks\_april2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2019-03-02' AS DATE) AND CAST('2019-04-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_april2019(id);

CREATE INDEX owner\_id ON indian\_forks\_april2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_april2019(forked\_from);

CREATE TABLE indian\_forks\_march2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2019-02-02' AS DATE) AND CAST('2019-03-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_march2019(id);

CREATE INDEX owner\_id ON indian\_forks\_march2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_march2019(forked\_from);

CREATE TABLE indian\_forks\_february2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2019-01-02' AS DATE) AND CAST('2019-02-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_february2019(id);

CREATE INDEX owner\_id ON indian\_forks\_february2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_february2019(forked\_from);

CREATE TABLE indian\_forks\_january2019

SELECT indian\_projects.\*

FROM indian\_projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-01-01' AS DATE)

AND forked\_from IS NOT NULL;

CREATE INDEX id ON indian\_forks\_january2019(id);

CREATE INDEX owner\_id ON indian\_forks\_january2019(owner\_id);

CREATE INDEX forked\_from ON indian\_forks\_january2019(forked\_from);

/\* Date Limited Fork Count Table Creation \*/

/\* Code blocks to create counts of the number of forks created on a monthly basis. A weighted score is also calculated and applied to each row as weighted\_fork\_count. The first six code blocks create tables of forks from all user sources. \*/

CREATE TABLE forks\_june2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_june2019, projects

WHERE forks\_june2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_june2019\_count(id);

ALTER TABLE forks\_june2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_june2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE forks\_may2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_may2019, projects

WHERE forks\_may2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_may2019\_count(id);

ALTER TABLE forks\_may2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_may2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE forks\_april2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_april2019, projects

WHERE forks\_april2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_april2019\_count(id);

ALTER TABLE forks\_april2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_april2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE forks\_march2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_march2019, projects

WHERE forks\_march2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_march2019\_count(id);

ALTER TABLE forks\_march2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_march2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE forks\_february2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_february2019, projects

WHERE forks\_february2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_february2019\_count(id);

ALTER TABLE forks\_february2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_february2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE forks\_january2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM forks\_january2019, projects

WHERE forks\_january2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON forks\_january2019\_count(id);

ALTER TABLE forks\_january2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE forks\_january2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

/\* Code blocks to created date limited fork counts from country specific user sources. \*/

CREATE TABLE russian\_forks\_june2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_june2019, projects

WHERE russian\_forks\_june2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_june2019\_count(id);

ALTER TABLE russian\_forks\_june2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_june2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE russian\_forks\_may2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_may2019, projects

WHERE russian\_forks\_may2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_may2019\_count(id);

ALTER TABLE russian\_forks\_may2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_may2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE russian\_forks\_april2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_april2019, projects

WHERE russian\_forks\_april2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_april2019\_count(id);

ALTER TABLE russian\_forks\_april2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_april2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE russian\_forks\_march2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_march2019, projects

WHERE russian\_forks\_march2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_march2019\_count(id);

ALTER TABLE russian\_forks\_march2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_march2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE russian\_forks\_february2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_february2019, projects

WHERE russian\_forks\_february2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_february2019\_count(id);

ALTER TABLE russian\_forks\_february2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_february2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE russian\_forks\_january2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM russian\_forks\_january2019, projects

WHERE russian\_forks\_january2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON russian\_forks\_january2019\_count(id);

ALTER TABLE russian\_forks\_january2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE russian\_forks\_january2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_june2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_june2019, projects

WHERE chinese\_forks\_june2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_june2019\_count(id);

ALTER TABLE chinese\_forks\_june2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_june2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_may2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_may2019, projects

WHERE chinese\_forks\_may2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_may2019\_count(id);

ALTER TABLE chinese\_forks\_may2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_may2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_april2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_april2019, projects

WHERE chinese\_forks\_april2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_april2019\_count(id);

ALTER TABLE chinese\_forks\_april2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_april2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_march2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_march2019, projects

WHERE chinese\_forks\_march2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_march2019\_count(id);

ALTER TABLE chinese\_forks\_march2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_march2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_february2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_february2019, projects

WHERE chinese\_forks\_february2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_february2019\_count(id);

ALTER TABLE chinese\_forks\_february2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_february2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE chinese\_forks\_january2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM chinese\_forks\_january2019, projects

WHERE chinese\_forks\_january2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON chinese\_forks\_january2019\_count(id);

ALTER TABLE chinese\_forks\_january2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE chinese\_forks\_january2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_june2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_june2019, projects

WHERE unitedstates\_forks\_june2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_june2019\_count(id);

ALTER TABLE unitedstates\_forks\_june2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_june2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_may2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_may2019, projects

WHERE unitedstates\_forks\_may2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_may2019\_count(id);

ALTER TABLE unitedstates\_forks\_may2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_may2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_april2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_april2019, projects

WHERE unitedstates\_forks\_april2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_april2019\_count(id);

ALTER TABLE unitedstates\_forks\_april2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_april2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_march2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_march2019, projects

WHERE unitedstates\_forks\_march2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_march2019\_count(id);

ALTER TABLE unitedstates\_forks\_march2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_march2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_february2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_february2019, projects

WHERE unitedstates\_forks\_february2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_february2019\_count(id);

ALTER TABLE unitedstates\_forks\_february2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_february2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE unitedstates\_forks\_january2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM unitedstates\_forks\_january2019, projects

WHERE unitedstates\_forks\_january2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON unitedstates\_forks\_january2019\_count(id);

ALTER TABLE unitedstates\_forks\_january2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE unitedstates\_forks\_january2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_june2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_june2019, projects

WHERE indian\_forks\_june2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_june2019\_count(id);

ALTER TABLE indian\_forks\_june2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_june2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_may2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_may2019, projects

WHERE indian\_forks\_may2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_may2019\_count(id);

ALTER TABLE indian\_forks\_may2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_may2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_april2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_april2019, projects

WHERE indian\_forks\_april2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_april2019\_count(id);

ALTER TABLE indian\_forks\_april2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_april2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_march2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_march2019, projects

WHERE indian\_forks\_march2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_march2019\_count(id);

ALTER TABLE indian\_forks\_march2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_march2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_february2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_february2019, projects

WHERE indian\_forks\_february2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_february2019\_count(id);

ALTER TABLE indian\_forks\_february2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_february2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

CREATE TABLE indian\_forks\_january2019\_count

SELECT projects.id, count(\*) AS fork\_count

FROM indian\_forks\_january2019, projects

WHERE indian\_forks\_january2019.forked\_from = projects.id

GROUP BY projects.id;

CREATE INDEX id ON indian\_forks\_january2019\_count(id);

ALTER TABLE indian\_forks\_january2019\_count

ADD(weighted\_fork\_count FLOAT(21));

UPDATE indian\_forks\_january2019\_count

SET weighted\_fork\_count = (fork\_count \* 0.75);

/\* Create Combined Repository Star/Fork (Influence) Tables and Influence Scores for Analyis \*/

/\* Code blocks to create date limited star/fork tables and perform calculations to develop influence scores for each starred/forked repository during the time period. The first six code blocks perform this function for repositories that received star/forks from all sources. \*/

CREATE TABLE projects\_influence\_june2019

SELECT \* FROM watchers\_june2019\_count

LEFT JOIN forks\_june2019\_count ON watchers\_june2019\_count.repo\_id = forks\_june2019\_count.id

UNION ALL

SELECT \* FROM watchers\_june2019\_count

RIGHT JOIN forks\_june2019\_count ON watchers\_june2019\_count.repo\_id = forks\_june2019\_count.id

WHERE watchers\_june2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_june2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_june2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_june2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255),

code\_language VARCHAR(255), forked\_from INT(11));

UPDATE projects\_influence\_june2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_june2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_june2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_june2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_june2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_june2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_june2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_june2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE

projects\_influence\_june2019.owner\_id = users.id)

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_june2019.repo\_id = projects.id);

CREATE TABLE projects\_influence\_may2019

SELECT \* FROM watchers\_may2019\_count

LEFT JOIN forks\_may2019\_count ON watchers\_may2019\_count.repo\_id = forks\_may2019\_count.id

UNION ALL

SELECT \* FROM watchers\_may2019\_count

RIGHT JOIN forks\_may2019\_count ON watchers\_may2019\_count.repo\_id = forks\_may2019\_count.id

WHERE watchers\_may2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_may2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_may2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_may2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE projects\_influence\_may2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_may2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_may2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_may2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_may2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_may2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_may2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_may2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE projects\_influence\_may2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_may2019.repo\_id = projects.id);

CREATE TABLE projects\_influence\_april2019

SELECT \* FROM watchers\_april2019\_count

LEFT JOIN forks\_april2019\_count ON watchers\_april2019\_count.repo\_id = forks\_april2019\_count.id

UNION ALL

SELECT \* FROM watchers\_april2019\_count

RIGHT JOIN forks\_april2019\_count ON watchers\_april2019\_count.repo\_id = forks\_april2019\_count.id

WHERE watchers\_april2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_april2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_april2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_april2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));UPDATE projects\_influence\_april2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_april2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_april2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_april2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_april2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_april2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_april2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_april2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE projects\_influence\_april2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_april2019.repo\_id = projects.id);

CREATE TABLE projects\_influence\_march2019

SELECT \* FROM watchers\_march2019\_count

LEFT JOIN forks\_march2019\_count ON watchers\_march2019\_count.repo\_id = forks\_march2019\_count.id

UNION ALL

SELECT \* FROM watchers\_march2019\_count

RIGHT JOIN forks\_march2019\_count ON watchers\_march2019\_count.repo\_id = forks\_march2019\_count.id

WHERE watchers\_march2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_march2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_march2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_march2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE projects\_influence\_march2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_march2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_march2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_march2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_march2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_march2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_march2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_march2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE projects\_influence\_march2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_march2019.repo\_id = projects.id);

CREATE TABLE projects\_influence\_february2019

SELECT \* FROM watchers\_february2019\_count

LEFT JOIN forks\_february2019\_count ON watchers\_february2019\_count.repo\_id = forks\_february2019\_count.id

UNION ALL

SELECT \* FROM watchers\_february2019\_count

RIGHT JOIN forks\_february2019\_count ON watchers\_february2019\_count.repo\_id = forks\_february2019\_count.id

WHERE watchers\_february2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_february2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_february2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_february2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE projects\_influence\_february2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_february2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_february2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_february2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_february2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_february2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_february2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_february2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE projects\_influence\_february2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_february2019.repo\_id = projects.id);

CREATE TABLE projects\_influence\_january2019

SELECT \* FROM watchers\_january2019\_count

LEFT JOIN forks\_january2019\_count ON watchers\_january2019\_count.repo\_id = forks\_january2019\_count.id

UNION ALL

SELECT \* FROM watchers\_january2019\_count

RIGHT JOIN forks\_january2019\_count ON watchers\_january2019\_count.repo\_id = forks\_january2019\_count.id

WHERE watchers\_january2019\_count.repo\_id IS NULL;

UPDATE projects\_influence\_january2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE projects\_influence\_january2019

DROP COLUMN id;

ALTER TABLE projects\_influence\_january2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE projects\_influence\_january2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE projects\_influence\_january2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE projects\_influence\_january2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE projects\_influence\_january2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM watchers\_january2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM forks\_january2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM forks\_january2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE projects\_influence\_january2019

SET repo\_name = (SELECT projects.name FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE projects\_influence\_january2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE projects\_influence\_january2019.repo\_id = projects.id);

/\* Code blocks to create time date limited influence tables for repositories receiving stars/forks from national sources. \*/

CREATE TABLE russian\_projects\_influence\_june2019

SELECT \* FROM russian\_watchers\_june2019\_count

LEFT JOIN russian\_forks\_june2019\_count ON russian\_watchers\_june2019\_count.repo\_id = russian\_forks\_june2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_june2019\_count

RIGHT JOIN russian\_forks\_june2019\_count ON russian\_watchers\_june2019\_count.repo\_id = russian\_forks\_june2019\_count.id

WHERE russian\_watchers\_june2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_june2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_june2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_june2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_june2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_june2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_june2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_june2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_june2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_june2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_june2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_june2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_june2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_june2019.repo\_id = projects.id);

CREATE TABLE russian\_projects\_influence\_may2019

SELECT \* FROM russian\_watchers\_may2019\_count

LEFT JOIN russian\_forks\_may2019\_count ON russian\_watchers\_may2019\_count.repo\_id = russian\_forks\_may2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_may2019\_count

RIGHT JOIN russian\_forks\_may2019\_count ON russian\_watchers\_may2019\_count.repo\_id = russian\_forks\_may2019\_count.id

WHERE russian\_watchers\_may2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_may2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_may2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_may2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_may2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_may2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_may2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_may2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_may2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_may2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_may2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_may2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_may2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_may2019.repo\_id = projects.id);

CREATE TABLE russian\_projects\_influence\_april2019

SELECT \* FROM russian\_watchers\_april2019\_count

LEFT JOIN russian\_forks\_april2019\_count ON russian\_watchers\_april2019\_count.repo\_id = russian\_forks\_april2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_april2019\_count

RIGHT JOIN russian\_forks\_april2019\_count ON russian\_watchers\_april2019\_count.repo\_id = russian\_forks\_april2019\_count.id

WHERE russian\_watchers\_april2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_april2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_april2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_april2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_april2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_april2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_april2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_april2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_april2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_april2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_april2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_april2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_april2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_april2019.repo\_id = projects.id);

CREATE TABLE russian\_projects\_influence\_march2019

SELECT \* FROM russian\_watchers\_march2019\_count

LEFT JOIN russian\_forks\_march2019\_count ON russian\_watchers\_march2019\_count.repo\_id = russian\_forks\_march2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_march2019\_count

RIGHT JOIN russian\_forks\_march2019\_count ON russian\_watchers\_march2019\_count.repo\_id = russian\_forks\_march2019\_count.id

WHERE russian\_watchers\_march2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_march2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_march2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_march2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_march2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_march2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_march2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_march2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_march2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_march2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_march2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_march2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_march2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_march2019.repo\_id = projects.id);

CREATE TABLE russian\_projects\_influence\_february2019

SELECT \* FROM russian\_watchers\_february2019\_count

LEFT JOIN russian\_forks\_february2019\_count ON russian\_watchers\_february2019\_count.repo\_id = russian\_forks\_february2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_february2019\_count

RIGHT JOIN russian\_forks\_february2019\_count ON russian\_watchers\_february2019\_count.repo\_id = russian\_forks\_february2019\_count.id

WHERE russian\_watchers\_february2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_february2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_february2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_february2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_february2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_february2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_february2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_february2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_february2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_february2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_february2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_february2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_february2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_february2019.repo\_id = projects.id);

CREATE TABLE russian\_projects\_influence\_january2019

SELECT \* FROM russian\_watchers\_january2019\_count

LEFT JOIN russian\_forks\_january2019\_count ON russian\_watchers\_january2019\_count.repo\_id = russian\_forks\_january2019\_count.id

UNION ALL

SELECT \* FROM russian\_watchers\_january2019\_count

RIGHT JOIN russian\_forks\_january2019\_count ON russian\_watchers\_january2019\_count.repo\_id = russian\_forks\_january2019\_count.id

WHERE russian\_watchers\_january2019\_count.repo\_id IS NULL;

UPDATE russian\_projects\_influence\_january2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE russian\_projects\_influence\_january2019

DROP COLUMN id;

ALTER TABLE russian\_projects\_influence\_january2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE russian\_projects\_influence\_january2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE russian\_projects\_influence\_january2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE russian\_projects\_influence\_january2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE russian\_projects\_influence\_january2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM russian\_watchers\_january2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM russian\_forks\_january2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM russian\_forks\_january2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE russian\_projects\_influence\_january2019

SET repo\_name = (SELECT projects.name FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE russian\_projects\_influence\_january2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE russian\_projects\_influence\_january2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_june2019

SELECT \* FROM chinese\_watchers\_june2019\_count

LEFT JOIN chinese\_forks\_june2019\_count ON chinese\_watchers\_june2019\_count.repo\_id = chinese\_forks\_june2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_june2019\_count

RIGHT JOIN chinese\_forks\_june2019\_count ON chinese\_watchers\_june2019\_count.repo\_id = chinese\_forks\_june2019\_count.id

WHERE chinese\_watchers\_june2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_june2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_june2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_june2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_june2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_june2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_june2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_june2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_june2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_june2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_june2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_june2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_june2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_june2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_may2019

SELECT \* FROM chinese\_watchers\_may2019\_count

LEFT JOIN chinese\_forks\_may2019\_count ON chinese\_watchers\_may2019\_count.repo\_id = chinese\_forks\_may2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_may2019\_count

RIGHT JOIN chinese\_forks\_may2019\_count ON chinese\_watchers\_may2019\_count.repo\_id = chinese\_forks\_may2019\_count.id

WHERE chinese\_watchers\_may2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_may2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_may2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_may2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_may2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_may2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_may2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_may2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_may2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_may2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_may2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_may2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_may2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_may2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_april2019

SELECT \* FROM chinese\_watchers\_april2019\_count

LEFT JOIN chinese\_forks\_april2019\_count ON chinese\_watchers\_april2019\_count.repo\_id = chinese\_forks\_april2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_april2019\_count

RIGHT JOIN chinese\_forks\_april2019\_count ON chinese\_watchers\_april2019\_count.repo\_id = chinese\_forks\_april2019\_count.id

WHERE chinese\_watchers\_april2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_april2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_april2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_april2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_april2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_april2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_april2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_april2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_april2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_april2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_april2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_april2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_april2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_april2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_march2019

SELECT \* FROM chinese\_watchers\_march2019\_count

LEFT JOIN chinese\_forks\_march2019\_count ON chinese\_watchers\_march2019\_count.repo\_id = chinese\_forks\_march2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_march2019\_count

RIGHT JOIN chinese\_forks\_march2019\_count ON chinese\_watchers\_march2019\_count.repo\_id = chinese\_forks\_march2019\_count.id

WHERE chinese\_watchers\_march2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_march2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_march2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_march2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_march2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_march2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_march2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_march2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_march2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_march2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_march2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_march2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_march2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_march2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_february2019

SELECT \* FROM chinese\_watchers\_february2019\_count

LEFT JOIN chinese\_forks\_february2019\_count ON chinese\_watchers\_february2019\_count.repo\_id = chinese\_forks\_february2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_february2019\_count

RIGHT JOIN chinese\_forks\_february2019\_count ON chinese\_watchers\_february2019\_count.repo\_id = chinese\_forks\_february2019\_count.id

WHERE chinese\_watchers\_february2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_february2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_february2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_february2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_february2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_february2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_february2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_february2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_february2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_february2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_february2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_february2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_february2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_february2019.repo\_id = projects.id);

CREATE TABLE chinese\_projects\_influence\_january2019

SELECT \* FROM chinese\_watchers\_january2019\_count

LEFT JOIN chinese\_forks\_january2019\_count ON chinese\_watchers\_january2019\_count.repo\_id = chinese\_forks\_january2019\_count.id

UNION ALL

SELECT \* FROM chinese\_watchers\_january2019\_count

RIGHT JOIN chinese\_forks\_january2019\_count ON chinese\_watchers\_january2019\_count.repo\_id = chinese\_forks\_january2019\_count.id

WHERE chinese\_watchers\_january2019\_count.repo\_id IS NULL;

UPDATE chinese\_projects\_influence\_january2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE chinese\_projects\_influence\_january2019

DROP COLUMN id;

ALTER TABLE chinese\_projects\_influence\_january2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE chinese\_projects\_influence\_january2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE chinese\_projects\_influence\_january2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_january2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE chinese\_projects\_influence\_january2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM chinese\_watchers\_january2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM chinese\_forks\_january2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM chinese\_forks\_january2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE chinese\_projects\_influence\_january2019

SET repo\_name = (SELECT projects.name FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE chinese\_projects\_influence\_january2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE chinese\_projects\_influence\_january2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_june2019

SELECT \* FROM unitedstates\_watchers\_june2019\_count

LEFT JOIN unitedstates\_forks\_june2019\_count ON unitedstates\_watchers\_june2019\_count.repo\_id = unitedstates\_forks\_june2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_june2019\_count

RIGHT JOIN unitedstates\_forks\_june2019\_count ON unitedstates\_watchers\_june2019\_count.repo\_id = unitedstates\_forks\_june2019\_count.id

WHERE unitedstates\_watchers\_june2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_june2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_june2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_june2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_june2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_june2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_june2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_june2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_june2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_june2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_june2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_june2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_june2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_june2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_may2019

SELECT \* FROM unitedstates\_watchers\_may2019\_count

LEFT JOIN unitedstates\_forks\_may2019\_count ON unitedstates\_watchers\_may2019\_count.repo\_id = unitedstates\_forks\_may2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_may2019\_count

RIGHT JOIN unitedstates\_forks\_may2019\_count ON unitedstates\_watchers\_may2019\_count.repo\_id = unitedstates\_forks\_may2019\_count.id

WHERE unitedstates\_watchers\_may2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_may2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_may2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_may2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_may2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_may2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_may2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_may2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_may2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_may2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_may2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_may2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_may2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_may2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_april2019

SELECT \* FROM unitedstates\_watchers\_april2019\_count

LEFT JOIN unitedstates\_forks\_april2019\_count ON unitedstates\_watchers\_april2019\_count.repo\_id = unitedstates\_forks\_april2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_april2019\_count

RIGHT JOIN unitedstates\_forks\_april2019\_count ON unitedstates\_watchers\_april2019\_count.repo\_id = unitedstates\_forks\_april2019\_count.id

WHERE unitedstates\_watchers\_april2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_april2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_april2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_april2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_april2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_april2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_april2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_april2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_april2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_april2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_april2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_april2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_april2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_april2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_march2019

SELECT \* FROM unitedstates\_watchers\_march2019\_count

LEFT JOIN unitedstates\_forks\_march2019\_count ON unitedstates\_watchers\_march2019\_count.repo\_id = unitedstates\_forks\_march2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_march2019\_count

RIGHT JOIN unitedstates\_forks\_march2019\_count ON unitedstates\_watchers\_march2019\_count.repo\_id = unitedstates\_forks\_march2019\_count.id

WHERE unitedstates\_watchers\_march2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_march2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_march2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_march2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_march2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_march2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_march2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_march2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_march2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_march2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_march2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_march2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_march2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_march2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_february2019

SELECT \* FROM unitedstates\_watchers\_february2019\_count

LEFT JOIN unitedstates\_forks\_february2019\_count ON unitedstates\_watchers\_february2019\_count.repo\_id = unitedstates\_forks\_february2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_february2019\_count

RIGHT JOIN unitedstates\_forks\_february2019\_count ON unitedstates\_watchers\_february2019\_count.repo\_id = unitedstates\_forks\_february2019\_count.id

WHERE unitedstates\_watchers\_february2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_february2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_february2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_february2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_february2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_february2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_february2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_february2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_february2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_february2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_february2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_february2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_february2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_february2019.repo\_id = projects.id);

CREATE TABLE unitedstates\_projects\_influence\_january2019

SELECT \* FROM unitedstates\_watchers\_january2019\_count

LEFT JOIN unitedstates\_forks\_january2019\_count ON unitedstates\_watchers\_january2019\_count.repo\_id = unitedstates\_forks\_january2019\_count.id

UNION ALL

SELECT \* FROM unitedstates\_watchers\_january2019\_count

RIGHT JOIN unitedstates\_forks\_january2019\_count ON unitedstates\_watchers\_january2019\_count.repo\_id = unitedstates\_forks\_january2019\_count.id

WHERE unitedstates\_watchers\_january2019\_count.repo\_id IS NULL;

UPDATE unitedstates\_projects\_influence\_january2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE unitedstates\_projects\_influence\_january2019

DROP COLUMN id;

ALTER TABLE unitedstates\_projects\_influence\_january2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE unitedstates\_projects\_influence\_january2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_january2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_january2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE unitedstates\_projects\_influence\_january2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM unitedstates\_watchers\_january2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM unitedstates\_forks\_january2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM unitedstates\_forks\_january2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE unitedstates\_projects\_influence\_january2019

SET repo\_name = (SELECT projects.name FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE unitedstates\_projects\_influence\_january2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE unitedstates\_projects\_influence\_january2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_june2019

SELECT \* FROM indian\_watchers\_june2019\_count

LEFT JOIN indian\_forks\_june2019\_count ON indian\_watchers\_june2019\_count.repo\_id = indian\_forks\_june2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_june2019\_count

RIGHT JOIN indian\_forks\_june2019\_count ON indian\_watchers\_june2019\_count.repo\_id = indian\_forks\_june2019\_count.id

WHERE indian\_watchers\_june2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_june2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_june2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_june2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_june2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_june2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_june2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_june2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_june2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_june2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_june2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_june2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_june2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_june2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_may2019

SELECT \* FROM indian\_watchers\_may2019\_count

LEFT JOIN indian\_forks\_may2019\_count ON indian\_watchers\_may2019\_count.repo\_id = indian\_forks\_may2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_may2019\_count

RIGHT JOIN indian\_forks\_may2019\_count ON indian\_watchers\_may2019\_count.repo\_id = indian\_forks\_may2019\_count.id

WHERE indian\_watchers\_may2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_may2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_may2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_may2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_may2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_may2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_may2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_may2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_may2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_may2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_may2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_may2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_may2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_may2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_april2019

SELECT \* FROM indian\_watchers\_april2019\_count

LEFT JOIN indian\_forks\_april2019\_count ON indian\_watchers\_april2019\_count.repo\_id = indian\_forks\_april2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_april2019\_count

RIGHT JOIN indian\_forks\_april2019\_count ON indian\_watchers\_april2019\_count.repo\_id = indian\_forks\_april2019\_count.id

WHERE indian\_watchers\_april2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_april2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_april2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_april2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_april2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_april2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_april2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_april2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_april2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_april2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_april2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_april2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_april2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_april2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_march2019

SELECT \* FROM indian\_watchers\_march2019\_count

LEFT JOIN indian\_forks\_march2019\_count ON indian\_watchers\_march2019\_count.repo\_id = indian\_forks\_march2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_march2019\_count

RIGHT JOIN indian\_forks\_march2019\_count ON indian\_watchers\_march2019\_count.repo\_id = indian\_forks\_march2019\_count.id

WHERE indian\_watchers\_march2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_march2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_march2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_march2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_march2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_march2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_march2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_march2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_march2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_march2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_march2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_march2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_march2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_march2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_february2019

SELECT \* FROM indian\_watchers\_february2019\_count

LEFT JOIN indian\_forks\_february2019\_count ON indian\_watchers\_february2019\_count.repo\_id = indian\_forks\_february2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_february2019\_count

RIGHT JOIN indian\_forks\_february2019\_count ON indian\_watchers\_february2019\_count.repo\_id = indian\_forks\_february2019\_count.id

WHERE indian\_watchers\_february2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_february2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_february2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_february2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_february2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_february2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_february2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_february2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_february2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_february2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_february2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_february2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_february2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_february2019.repo\_id = projects.id);

CREATE TABLE indian\_projects\_influence\_january2019

SELECT \* FROM indian\_watchers\_january2019\_count

LEFT JOIN indian\_forks\_january2019\_count ON indian\_watchers\_january2019\_count.repo\_id = indian\_forks\_january2019\_count.id

UNION ALL

SELECT \* FROM indian\_watchers\_january2019\_count

RIGHT JOIN indian\_forks\_january2019\_count ON indian\_watchers\_january2019\_count.repo\_id = indian\_forks\_january2019\_count.id

WHERE indian\_watchers\_january2019\_count.repo\_id IS NULL;

UPDATE indian\_projects\_influence\_january2019

SET repo\_id = id

WHERE repo\_id IS NULL;

ALTER TABLE indian\_projects\_influence\_january2019

DROP COLUMN id;

ALTER TABLE indian\_projects\_influence\_january2019

ADD(influence\_count BIGINT(21), weighted\_influence\_count FLOAT(21), star\_total BIGINT(21),

fork\_total BIGINT(21), weighted\_fork\_total FLOAT(21), influence\_total BIGINT(21), weighted\_influence\_total FLOAT(21),

influence\_score FLOAT(21), weighted\_influence\_score FLOAT(21), repo\_name VARCHAR(255), description VARCHAR(255), url VARCHAR(255),

owner\_id INT(11), owner\_name VARCHAR(255), code\_language VARCHAR(255), forked\_from INT(11));

UPDATE indian\_projects\_influence\_january2019

SET star\_count = 0

WHERE star\_count IS NULL;

UPDATE indian\_projects\_influence\_january2019

SET fork\_count = 0

WHERE fork\_count IS NULL;

UPDATE indian\_projects\_influence\_january2019

SET weighted\_fork\_count = 0

WHERE weighted\_fork\_count IS NULL;

UPDATE indian\_projects\_influence\_january2019

SET influence\_count = (star\_count + fork\_count), weighted\_influence\_count = (star\_count + weighted\_fork\_count),

star\_total = (SELECT SUM(star\_count) FROM indian\_watchers\_january2019\_count), fork\_total = (SELECT SUM(fork\_count) FROM indian\_forks\_january2019\_count),

weighted\_fork\_total = (SELECT SUM(weighted\_fork\_count) FROM indian\_forks\_january2019\_count), influence\_total = (star\_total + fork\_total),

weighted\_influence\_total = (star\_total + weighted\_fork\_total), influence\_score = (influence\_count / influence\_total),

weighted\_influence\_score = (weighted\_influence\_count / weighted\_influence\_total);

UPDATE indian\_projects\_influence\_january2019

SET repo\_name = (SELECT projects.name FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id),

description = (SELECT projects.description FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id),

url = (SELECT projects.url FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_id = (SELECT projects.owner\_id FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id),

owner\_name = (SELECT users.login FROM users WHERE indian\_projects\_influence\_january2019.owner\_id = users.id),

code\_language = (SELECT projects.language FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id),

forked\_from = (SELECT projects.forked\_from FROM projects WHERE indian\_projects\_influence\_january2019.repo\_id = projects.id);

/\* Create Clean User Table for Analysis

/\* Code blocks to create a clean user tables (with only *u.follow, u.watch,* and *u.fork* data) for creating histograms. \*/

CREATE TABLE russian\_users\_simple

SELECT id AS user\_id, login, follower\_count, russian\_follower\_count, fork\_count, russian\_fork\_count, watcher\_count, russian\_watcher\_count

FROM ghtorrent\_restore.russian\_users;

CREATE TABLE chinese\_users\_simple

SELECT id AS user\_id, login, follower\_count, chinese\_follower\_count, fork\_count, chinese\_fork\_count, watcher\_count, chinese\_watcher\_count

FROM ghtorrent\_restore.chinese\_users;

CREATE TABLE unitedstates\_users\_simple

SELECT id AS user\_id, login, follower\_count, unitedstates\_follower\_count, fork\_count, unitedstates\_fork\_count, watcher\_count, unitedstates\_watcher\_count

FROM ghtorrent\_restore.unitedstates\_users;

CREATE TABLE indian\_users\_simple

SELECT id AS user\_id, login, follower\_count, indian\_follower\_count, fork\_count, indian\_fork\_count, watcher\_count, indian\_watcher\_count

FROM ghtorrent\_restore.indian\_users;

/\* Create Six-month Repository Star Tables for Analysis \*/

/\*Code blocks to create the tables used to export data for use in the Appendix B Python code that creates histograms for analyzing the repository watching activity of the nation groups \*/

CREATE TABLE watchers\_6\_months

SELECT watchers.\*

FROM watchers

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-06-01' AS DATE);

CREATE INDEX repo\_id ON watchers\_6\_months (repo\_id);

CREATE INDEX user\_id ON watchers\_6\_months (user\_id);

CREATE TABLE russian\_watchers\_6\_months

SELECT watchers\_6\_months.\*

FROM watchers\_6\_months, russian\_users

WHERE watchers\_6\_months.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_watchers\_6\_months (repo\_id);

CREATE INDEX user\_id ON russian\_watchers\_6\_months (user\_id);

CREATE TABLE chinese\_watchers\_6\_months

SELECT watchers\_6\_months.\*

FROM watchers\_6\_months, chinese\_users

WHERE watchers\_6\_months.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_watchers\_6\_months (repo\_id);

CREATE INDEX user\_id ON chinese\_watchers\_6\_months (user\_id);

CREATE TABLE unitedstates\_watchers\_6\_months

SELECT watchers\_6\_months.\*

FROM watchers\_6\_months, unitedstates\_users

WHERE watchers\_6\_months.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_watchers\_6\_months (repo\_id);

CREATE INDEX user\_id ON unitedstates\_watchers\_6\_months (user\_id);

CREATE TABLE indian\_watchers\_6\_months

SELECT watchers\_6\_months.\*

FROM watchers\_6\_months, indian\_users

WHERE watchers\_6\_months.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_watchers\_6\_months (repo\_id);

CREATE INDEX user\_id ON indian\_watchers\_6\_months (user\_id);

/\* Code blocks to create the raw count tables \*/

CREATE TABLE watchers\_6\_months\_count

SELECT watchers\_6\_months.user\_id, COUNT(\*) AS star\_count

FROM watchers\_6\_months

GROUP BY watchers\_6\_months.user\_id;

CREATE INDEX user\_id ON watchers\_6\_months\_count(user\_id);

CREATE TABLE russian\_watchers\_6\_months\_count

SELECT russian\_watchers\_6\_months.user\_id, COUNT(\*) AS star\_count

FROM russian\_watchers\_6\_months

GROUP BY russian\_watchers\_6\_months.user\_id;

CREATE INDEX user\_id ON russian\_watchers\_6\_months\_count(user\_id);

CREATE TABLE chinese\_watchers\_6\_months\_count

SELECT chinese\_watchers\_6\_months.user\_id, COUNT(\*) AS star\_count

FROM chinese\_watchers\_6\_months

GROUP BY chinese\_watchers\_6\_months.user\_id;

CREATE INDEX user\_id ON chinese\_watchers\_6\_months\_count(user\_id);

CREATE TABLE unitedstates\_watchers\_6\_months\_count

SELECT unitedstates\_watchers\_6\_months.user\_id, COUNT(\*) AS star\_count

FROM unitedstates\_watchers\_6\_months

GROUP BY unitedstates\_watchers\_6\_months.user\_id;

CREATE INDEX user\_id ON unitedstates\_watchers\_6\_months\_count(user\_id);

CREATE TABLE indian\_watchers\_6\_months\_count

SELECT indian\_watchers\_6\_months.user\_id, COUNT(\*) AS star\_count

FROM indian\_watchers\_6\_months

GROUP BY indian\_watchers\_6\_months.user\_id;

CREATE INDEX user\_id ON indian\_watchers\_6\_months\_count(user\_id);

/\* Code blocks to include users with 0 stars for the six-month period \*/

CREATE TABLE watchers\_6\_months\_all

SELECT users.id AS user\_id, watchers\_6\_months\_count.star\_count

FROM users

LEFT JOIN watchers\_6\_months\_count ON users.id = watchers\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON watchers\_6\_months\_all (user\_id);

UPDATE watchers\_6\_months\_all

SET star\_count = 0

WHERE star\_count IS NULL;

CREATE TABLE russian\_watchers\_6\_months\_all

SELECT russian\_users.id AS user\_id, russian\_watchers\_6\_months\_count.star\_count

FROM russian\_users

LEFT JOIN russian\_watchers\_6\_months\_count ON russian\_users.id = russian\_watchers\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON russian\_watchers\_6\_months\_all (user\_id);

UPDATE russian\_watchers\_6\_months\_all

SET star\_count = 0

WHERE star\_count IS NULL;

CREATE TABLE chinese\_watchers\_6\_months\_all

SELECT chinese\_users.id AS user\_id, chinese\_watchers\_6\_months\_count.star\_count

FROM chinese\_users

LEFT JOIN chinese\_watchers\_6\_months\_count ON chinese\_users.id = chinese\_watchers\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON chinese\_watchers\_6\_months\_all (user\_id);

UPDATE chinese\_watchers\_6\_months\_all

SET star\_count = 0

WHERE star\_count IS NULL;

CREATE TABLE unitedstates\_watchers\_6\_months\_all

SELECT unitedstates\_users.id AS user\_id, unitedstates\_watchers\_6\_months\_count.star\_count

FROM unitedstates\_users

LEFT JOIN unitedstates\_watchers\_6\_months\_count ON unitedstates\_users.id = unitedstates\_watchers\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON unitedstates\_watchers\_6\_months\_all (user\_id);

UPDATE unitedstates\_watchers\_6\_months\_all

SET star\_count = 0

WHERE star\_count IS NULL;

CREATE TABLE indian\_watchers\_6\_months\_all

SELECT indian\_users.id AS user\_id, indian\_watchers\_6\_months\_count.star\_count

FROM indian\_users

LEFT JOIN indian\_watchers\_6\_months\_count ON indian\_users.id = indian\_watchers\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON indian\_watchers\_6\_months\_all (user\_id);

UPDATE indian\_watchers\_6\_months\_all

SET star\_count = 0

WHERE star\_count IS NULL;

/\* Create Six-month Repository Fork Tables for Analysis \*/

/\* Creates the tables used to export data for use in the Appendix B Python code that creates histograms for analyzing the repository forking activity of the nation groups \*/

CREATE TABLE forks\_6\_months

SELECT projects.owner\_id AS user\_id, projects.id AS repo\_id, projects.forked\_from

FROM projects

WHERE created\_at BETWEEN CAST('2018-12-02' AS DATE) AND CAST('2019-06-01' AS DATE);

CREATE INDEX repo\_id ON forks\_6\_months (repo\_id);

CREATE INDEX user\_id ON forks\_6\_months (user\_id);

DELETE FROM forks\_6\_months WHERE forked\_from IS NULL;

CREATE TABLE russian\_forks\_6\_months

SELECT forks\_6\_months.\*

FROM forks\_6\_months, russian\_users

WHERE forks\_6\_months.user\_id = russian\_users.id;

CREATE INDEX repo\_id ON russian\_forks\_6\_months (repo\_id);

CREATE INDEX user\_id ON russian\_forks\_6\_months (user\_id);

CREATE TABLE chinese\_forks\_6\_months

SELECT forks\_6\_months.\*

FROM forks\_6\_months, chinese\_users

WHERE forks\_6\_months.user\_id = chinese\_users.id;

CREATE INDEX repo\_id ON chinese\_forks\_6\_months (repo\_id);

CREATE INDEX user\_id ON chinese\_forks\_6\_months (user\_id);

CREATE TABLE unitedstates\_forks\_6\_months

SELECT forks\_6\_months.\*

FROM forks\_6\_months, unitedstates\_users

WHERE forks\_6\_months.user\_id = unitedstates\_users.id;

CREATE INDEX repo\_id ON unitedstates\_forks\_6\_months (repo\_id);

CREATE INDEX user\_id ON unitedstates\_forks\_6\_months (user\_id);

CREATE TABLE indian\_forks\_6\_months

SELECT forks\_6\_months.\*

FROM forks\_6\_months, indian\_users

WHERE forks\_6\_months.user\_id = indian\_users.id;

CREATE INDEX repo\_id ON indian\_forks\_6\_months (repo\_id);

CREATE INDEX user\_id ON indian\_forks\_6\_months (user\_id);

/\* Code blocks to create the raw count tables \*/

CREATE TABLE forks\_6\_months\_count

SELECT forks\_6\_months.user\_id, COUNT(\*) AS fork\_count

FROM forks\_6\_months

GROUP BY forks\_6\_months.user\_id;

CREATE INDEX user\_id ON forks\_6\_months\_count(user\_id);

CREATE TABLE russian\_forks\_6\_months\_count

SELECT russian\_forks\_6\_months.user\_id, COUNT(\*) AS fork\_count

FROM russian\_forks\_6\_months

GROUP BY russian\_forks\_6\_months.user\_id;

CREATE INDEX user\_id ON russian\_forks\_6\_months\_count(user\_id);

CREATE TABLE chinese\_forks\_6\_months\_count

SELECT chinese\_forks\_6\_months.user\_id, COUNT(\*) AS fork\_count

FROM chinese\_forks\_6\_months

GROUP BY chinese\_forks\_6\_months.user\_id;

CREATE INDEX user\_id ON chinese\_forks\_6\_months\_count(user\_id);

CREATE TABLE unitedstates\_forks\_6\_months\_count

SELECT unitedstates\_forks\_6\_months.user\_id, COUNT(\*) AS fork\_count

FROM unitedstates\_forks\_6\_months

GROUP BY unitedstates\_forks\_6\_months.user\_id;

CREATE INDEX user\_id ON unitedstates\_forks\_6\_months\_count(user\_id);

CREATE TABLE indian\_forks\_6\_months\_count

SELECT indian\_forks\_6\_months.user\_id, COUNT(\*) AS fork\_count

FROM indian\_forks\_6\_months

GROUP BY indian\_forks\_6\_months.user\_id;

CREATE INDEX user\_id ON indian\_forks\_6\_months\_count(user\_id);

/\* Code blocks to include users with 0 forks for the six-month period \*/

CREATE TABLE forks\_6\_months\_all

SELECT users.id AS user\_id, forks\_6\_months\_count.fork\_count

FROM users

LEFT JOIN forks\_6\_months\_count ON users.id = forks\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON forks\_6\_months\_all (user\_id);

UPDATE forks\_6\_months\_all

SET fork\_count = 0

WHERE fork\_count IS NULL;

CREATE TABLE russian\_forks\_6\_months\_all

SELECT russian\_users.id AS user\_id, russian\_forks\_6\_months\_count.fork\_count

FROM russian\_users

LEFT JOIN russian\_forks\_6\_months\_count ON russian\_users.id = russian\_forks\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON russian\_forks\_6\_months\_all (user\_id);

UPDATE russian\_forks\_6\_months\_all

SET fork\_count = 0

WHERE fork\_count IS NULL;

CREATE TABLE chinese\_forks\_6\_months\_all

SELECT chinese\_users.id AS user\_id, chinese\_forks\_6\_months\_count.fork\_count

FROM chinese\_users

LEFT JOIN chinese\_forks\_6\_months\_count ON chinese\_users.id = chinese\_forks\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON chinese\_forks\_6\_months\_all (user\_id);

UPDATE chinese\_forks\_6\_months\_all

SET fork\_count = 0

WHERE fork\_count IS NULL;

CREATE TABLE unitedstates\_forks\_6\_months\_all

SELECT unitedstates\_users.id AS user\_id, unitedstates\_forks\_6\_months\_count.fork\_count

FROM unitedstates\_users

LEFT JOIN unitedstates\_forks\_6\_months\_count ON unitedstates\_users.id = unitedstates\_forks\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON unitedstates\_forks\_6\_months\_all (user\_id);

UPDATE unitedstates\_forks\_6\_months\_all

SET fork\_count = 0

WHERE fork\_count IS NULL;

CREATE TABLE indian\_forks\_6\_months\_all

SELECT indian\_users.id AS user\_id, indian\_forks\_6\_months\_count.fork\_count

FROM indian\_users

LEFT JOIN indian\_forks\_6\_months\_count ON indian\_users.id = indian\_forks\_6\_months\_count.user\_id;

CREATE INDEX user\_id ON indian\_forks\_6\_months\_all (user\_id);

UPDATE indian\_forks\_6\_months\_all

SET fork\_count = 0

WHERE fork\_count IS NULL;