University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: 6/22/23

Group Number: 30

Name	**Student Number**	**CS Alias (Userid)**	**Preferred E-mail Address**
Logan Keener	67439018	w2m9y	lokeener@student.ubc.ca
Gavin George	38240818	n0x6y	gavinmgeorge@gmail.com
Wendy Han	51108264	c1g2b	ws.han@outlook.com

University of British Columbia, Vancouver
Department of Computer Science

Summary:
Our project is a social media site similar to Reddit. It has users, subreddits, comments, live broadcasts, ads, messages, admins, thumbnails and posts. We accomplished building an admin dashboard intended to make doing administration tasks easier.

Schema differences:
Our final schema is not different from the one we turned in for milestone 2.

Copy of schema:
Message(messageId: integer, sender: char[20], receiver: char[20])
- messageId is a unique PK
- sender is foreign key for User
- receiver is foreign key for User
- all should be non-null

FriendsWith(friend1: char[20], friend2: char[20])
- friend1 and friend2 are foreign keys for different Users
- PK: (friend1, friend2)

Ban(adminId: integer, userId: integer)
- adminId is non-null
- userId is a foreign key for User, non-null
- PK: (adminId, userId)

Ad(id: integer, content: char[60], link: char[50], category: integer, adminId: integer)
- id is a unique non-null PK
- content, category, adminId and link are non-null

Serves(adId: integer, subId: char[20])
- adId is a non-null foreign key for ad
- subId is a non-null foreign key for Subreddit
- PK: (adId, subId)

Subscribes(userId: char[20], subId: char[20])
- userId is non-null foreign key for User
- subId is non-null foreign key for Subreddit
- PK: (userId, subId)

VotesOn(userId: char[20], votableId: integer)
- userId is non-null foreign key for User
- votableId is non-null foreign key for Votable
- PK: (userId, votableId)

User(username: char[20], password: char[20], joinDate: date, avatarData: char[60], city: char[20], server: char[20])

- username and password are non-null

Broadcast(liveId: integer, title: char[30], quality: integer, creatorId: char[20], views: integer, popularity: integer [1…10])

- liveId is a non-null unique PK
- title is non-null
- creatorId is a non-null foreign key for User

Thumbnail(liveId: integer, postId: integer, tnId: char[20] imageData: char[60])

- liveId is a non-null foreign key for the related Broadcast
- postId is a non-null unique weak entity foreign key
- tnId is a non-null primary key that, along with liveId OR postId (both CKs)
- PK: (liveId, tnId)

Subreddit(name: char[20], creatorId: char[20])

- name is non-null unique PK
- creatorId is a foreign key for User

Votable(votableId: integer, awards: char[30], content: char[60], creatorId: char[20])

- votableId is a non-null unique PK
- creatorId is a foreign key for User

Comment(votableId: integer, parentId: char[20], awards: char[30], content: char[60], creatorId: char[20])

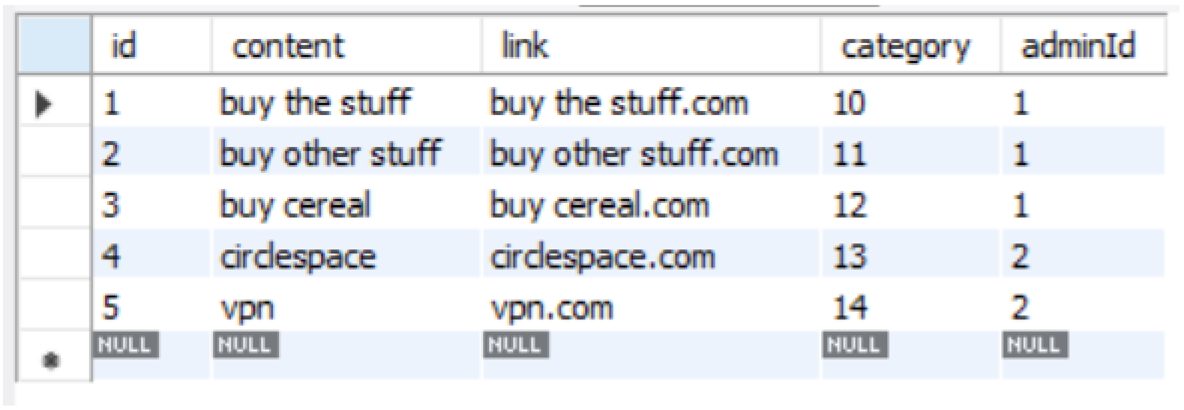
- creatorId is a non-null foreign key for Comment for the user who posted
- parentId is a non-null foreign key for Votable for the comment's parent post
- votableId is a non-null unique PK

Posts(votableId: integer, subId: integer) (to represent "has" relationship between Post and Subreddit)

- votableId is a non-null unique foreign key for Votable
- subId is a non-null foreign key for Subreddit
- PK: (votableId, subId)

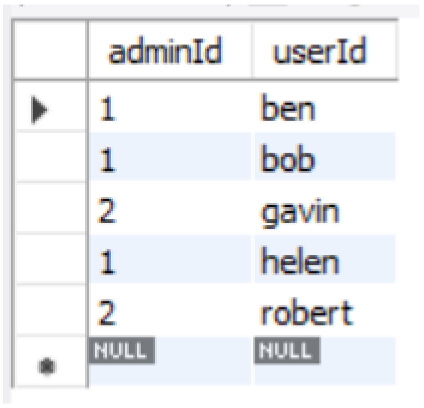
Screenshots of what's in each relation after initialization:

Ad:



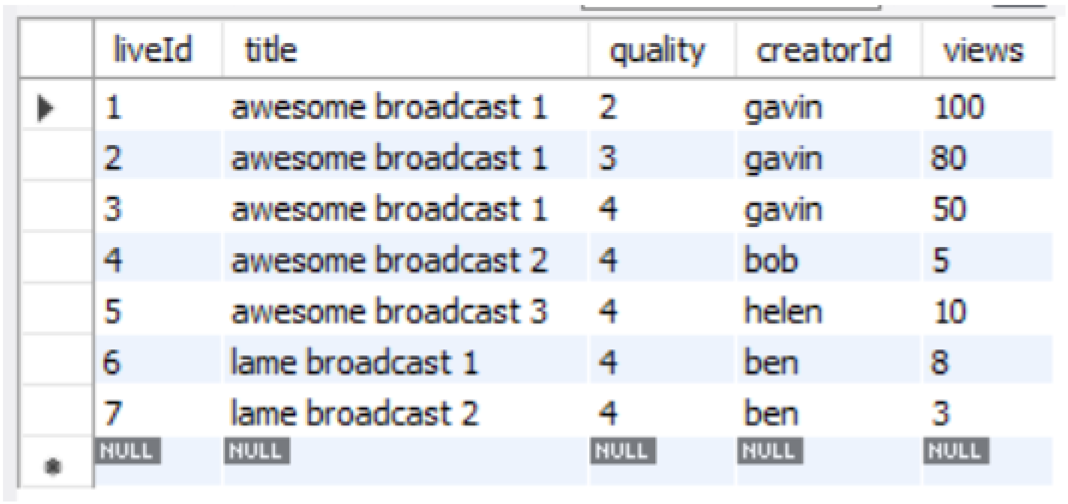
id	content	link	category	adminId
1	buy the stuff	buy the stuff.com	10	1
2	buy other stuff	buy other stuff.com	11	1
3	buy cereal	buy cereal.com	12	1
4	circlespace	circlespace.com	13	2
5	vpn	vpn.com	14	2
NULL	NULL	NULL	NULL	NULL

Ban:



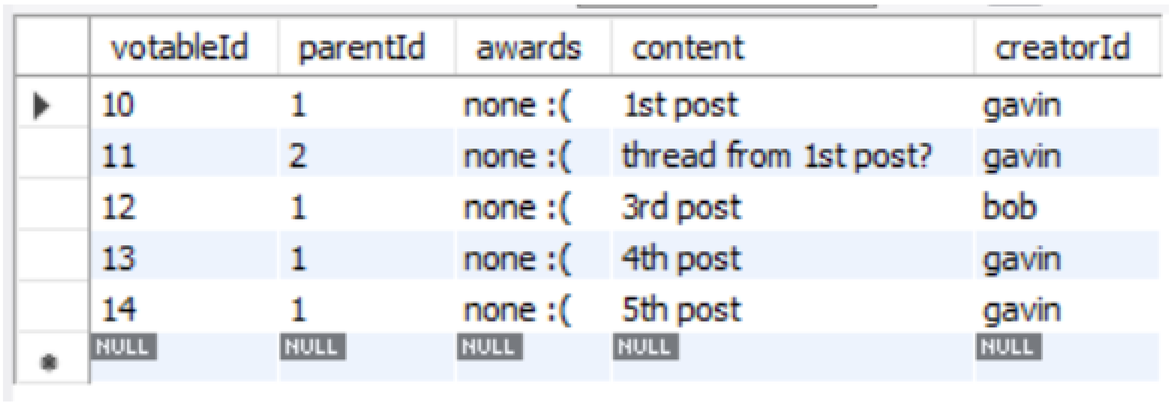
adminId	userId
1	ben
1	bob
2	gavin
1	helen
2	robert
NULL	NULL

Broadcast:



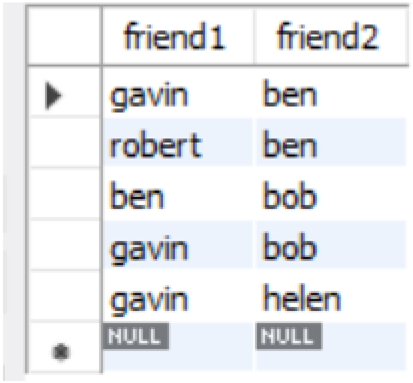
liveId	title	quality	creatorId	views
1	awesome broadcast 1	2	gavin	100
2	awesome broadcast 1	3	gavin	80
3	awesome broadcast 1	4	gavin	50
4	awesome broadcast 2	4	bob	5
5	awesome broadcast 3	4	helen	10
6	lame broadcast 1	4	ben	8
7	lame broadcast 2	4	ben	3
NULL	NULL	NULL	NULL	NULL

Comment:



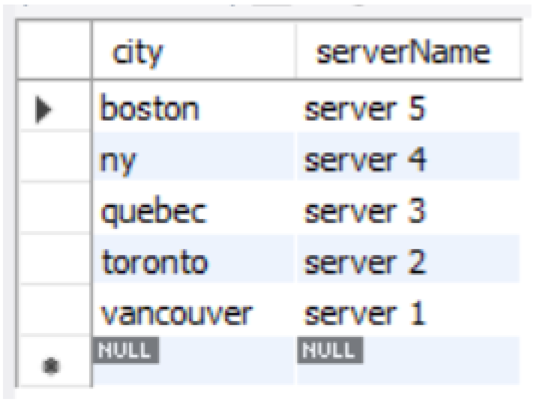
votableId	parentId	awards	content	creatorId
10	1	none :(	1st post	gavin
11	2	none :(	thread from 1st post?	gavin
12	1	none :(	3rd post	bob
13	1	none :(	4th post	gavin
14	1	none :(	5th post	gavin
NULL	NULL	NULL	NULL	NULL

FriendsWith:



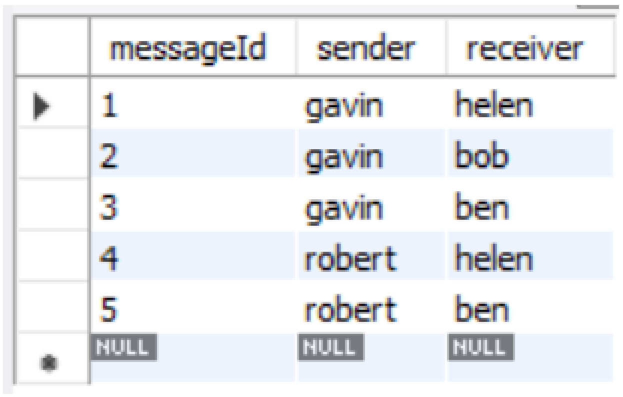
friend1	friend2
gavin	ben
robert	ben
ben	bob
gavin	bob
gavin	helen
NULL	NULL

Location:



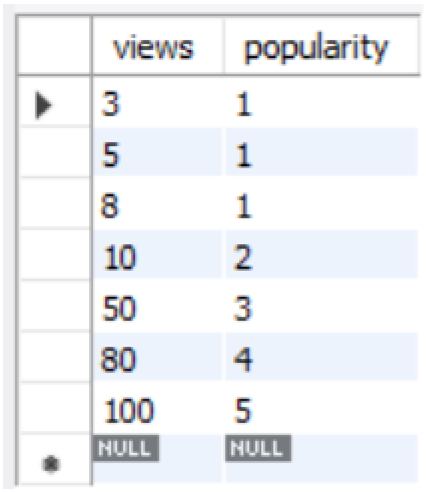
city	serverName
boston	server 5
ny	server 4
quebec	server 3
toronto	server 2
vancouver	server 1
NULL	NULL

Message:



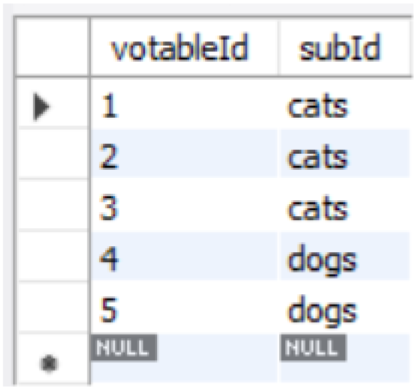
	messageId	sender	receiver
▶	1	gavin	helen
	2	gavin	bob
	3	gavin	ben
	4	robert	helen
	5	robert	ben
*	NULL	NULL	NULL

Popular:



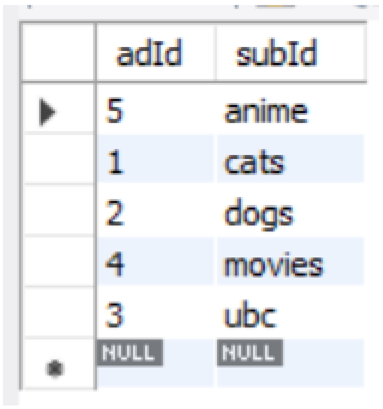
	views	popularity
▶	3	1
	5	1
	8	1
	10	2
	50	3
	80	4
	100	5
*	NULL	NULL

Posts:



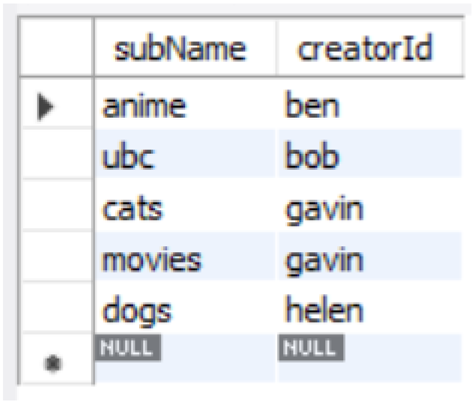
	votableId	subId
▶	1	cats
	2	cats
	3	cats
	4	dogs
	5	dogs
*	NULL	NULL

Serves:



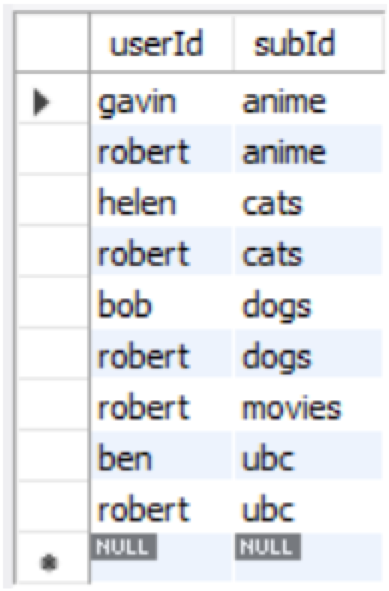
adId	subId
5	anime
1	cats
2	dogs
4	movies
3	ubc
NULL	NULL

Subreddit:



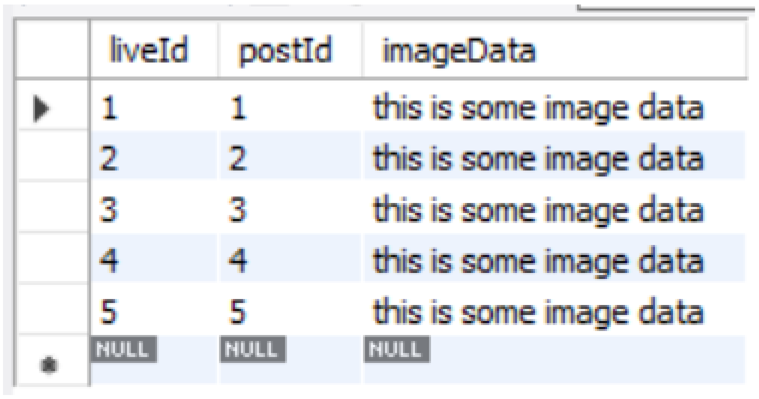
subName	creatorId
anime	ben
ubc	bob
cats	gavin
movies	gavin
dogs	helen
NULL	NULL

Subscribes:



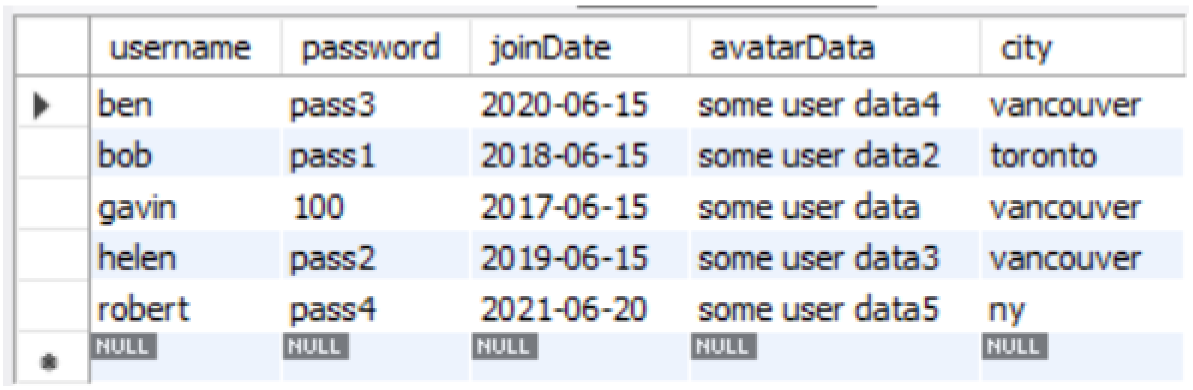
userId	subId
gavin	anime
robert	anime
helen	cats
robert	cats
bob	dogs
robert	dogs
robert	movies
ben	ubc
robert	ubc
NULL	NULL

Thumbnail:



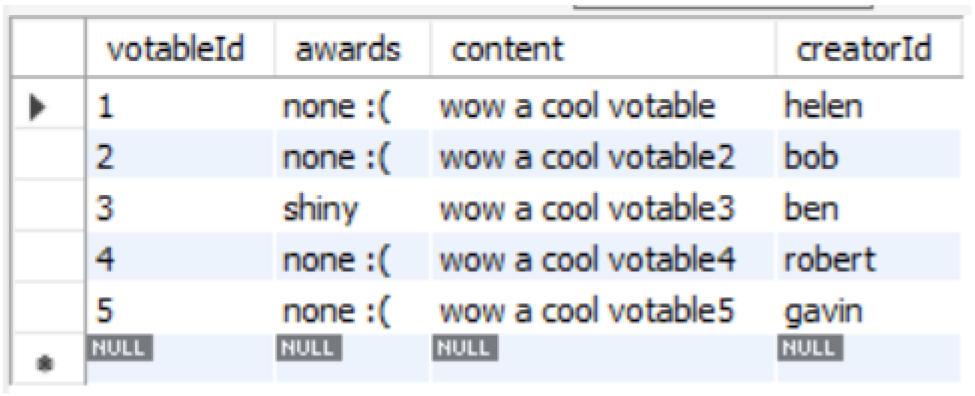
	liveId	postId	imageData
▶	1	1	this is some image data
	2	2	this is some image data
	3	3	this is some image data
	4	4	this is some image data
	5	5	this is some image data
*	NULL	NULL	NULL

UserTable:



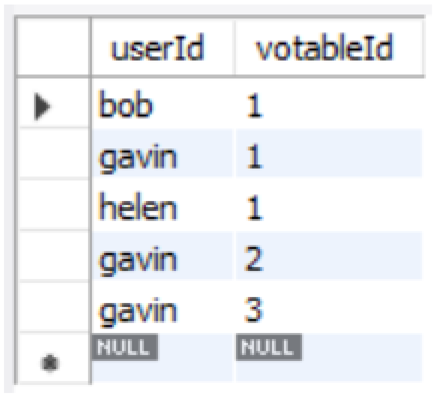
	username	password	joinDate	avatarData	city
▶	ben	pass3	2020-06-15	some user data4	vancouver
	bob	pass1	2018-06-15	some user data2	toronto
	gavin	100	2017-06-15	some user data	vancouver
	helen	pass2	2019-06-15	some user data3	vancouver
	robert	pass4	2021-06-20	some user data5	ny
*	NULL	NULL	NULL	NULL	NULL

Votable:



	votableId	awards	content	creatorId
▶	1	none :(	wow a cool votable	helen
	2	none :(	wow a cool votable2	bob
	3	shiny	wow a cool votable3	ben
	4	none :(	wow a cool votable4	robert
	5	none :(	wow a cool votable5	gavin
*	NULL	NULL	NULL	NULL

VotesOn:



userId	votableId
bob	1
gavin	1
helen	1
gavin	2
gavin	3
NULL	NULL

SQL queries used:

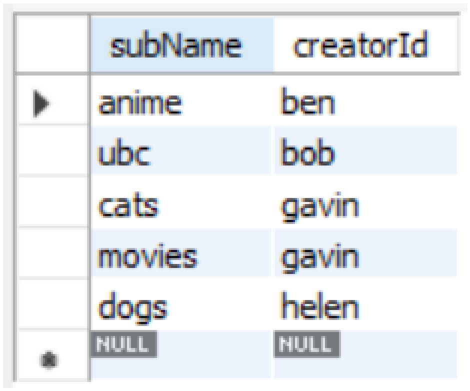
query type	sql query to do action
insert	INSERT INTO Subreddit VALUES ("${subreddit_name}", "${user_name}")
delete	DELETE FROM Posts WHERE ${whereClause}
update	UPDATE Subreddit SET creatorId = {input} WHERE subName = {key}
select query	SELECT ${attr_names} FROM ${table_name} WHERE ${whereClause};
projection query	SELECT {user chosen attributes} FROM {table} WHERE true
join query	SELECT COUNT(C.votableID) FROM Comment C, Post P WHERE P.votableID = ${input} AND C.parentID = ${input}
group by query	SELECT P.subId, COUNT(P.votableID) FROM Posts P GROUP BY P.subId
aggregation with having query	SELECT creatorId, MAX(B.views) FROM Broadcast as B GROUP BY creatorId HAVING AVG(B.views >=9)

nested aggregation with group by query	SELECT P.popularity, AVG (P.views)
	FROM Popular P
	GROUP BY P.popularity
	HAVING 1 < (SELECT COUNT (*)
	FROM Popular P2
	WHERE P.popularity = P2.popularity)

division query	SELECT COUNT (U.username) FROM UserTable AS U WHERE NOT EXISTS ((SELECT S.subName FROM Subreddit as S) EXCEPT (SELECT Sb.subId FROM Subscribes AS Sb WHERE Sb.userId = U.username))

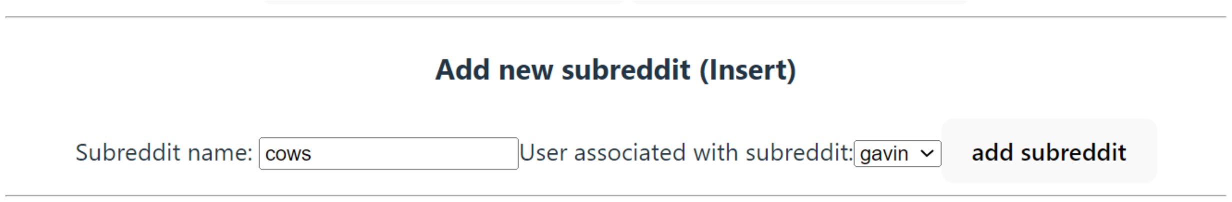
SQL screenshot of sample query output:

Insert:
Subreddit table before insert:



subName	creatorId
anime	ben
ubc	bob
cats	gavin
movies	gavin
dogs	helen
NULL	NULL

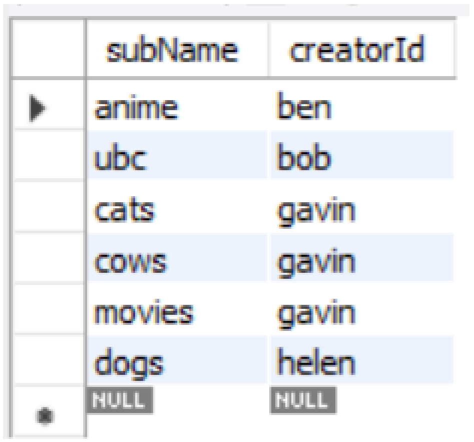
UI:



Add new subreddit (Insert)

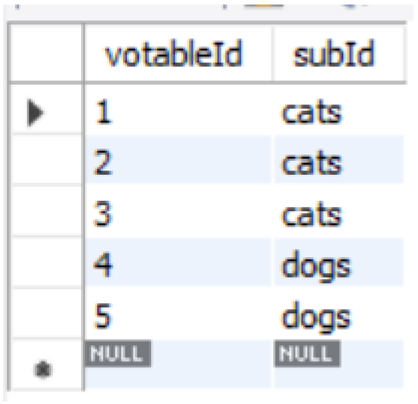
Subreddit name: cows	User associated with subreddit: gavin	add subreddit	

After adding:



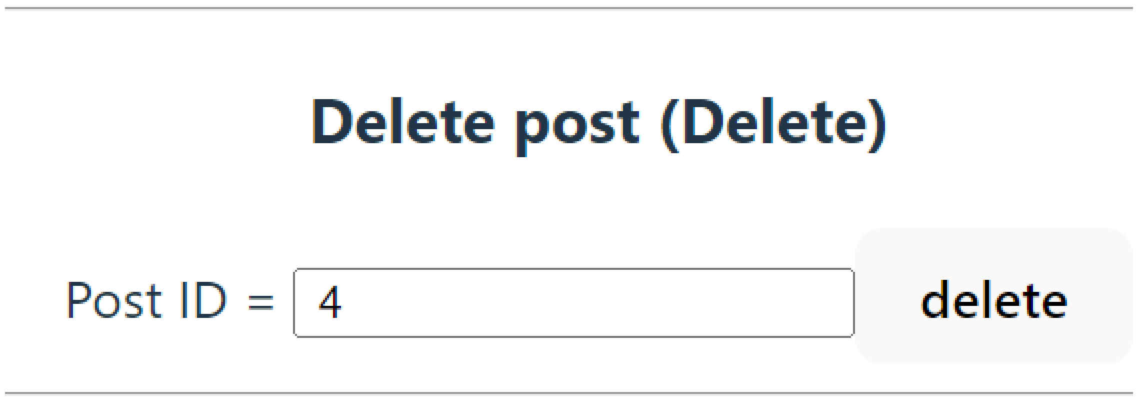
	subName	creatorId
▶	anime	ben
	ubc	bob
	cats	gavin
	cows	gavin
	movies	gavin
	dogs	helen
*	NULL	NULL

Delete:
Posts before delete



	votableId	subId
▶	1	cats
	2	cats
	3	cats
	4	dogs
	5	dogs
*	NULL	NULL

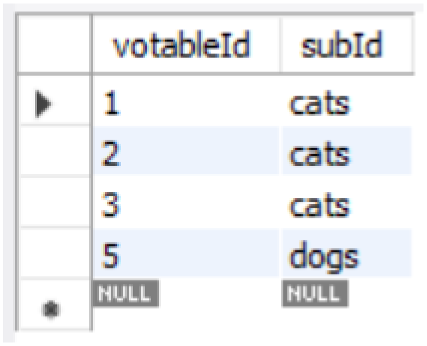
UI



Delete post (Delete)

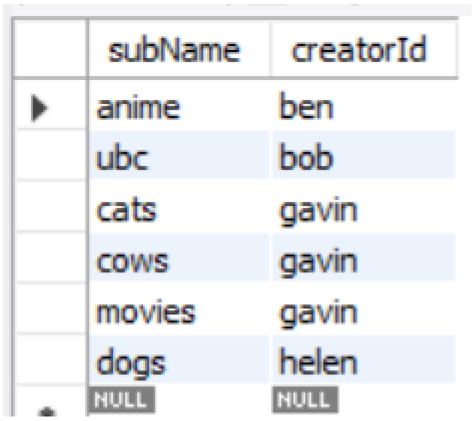
Post ID = 4

After delete



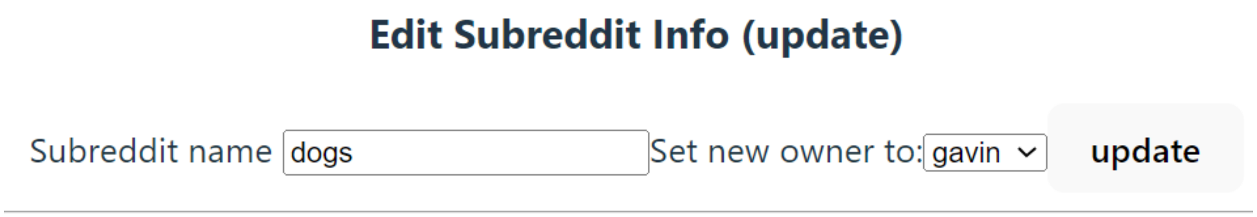
votableId	subId
1	cats
2	cats
3	cats
5	dogs
NULL	NULL

Update:
Subreddit table before update



subName	creatorId
anime	ben
ubc	bob
cats	gavin
cows	gavin
movies	gavin
dogs	helen
NULL	NULL

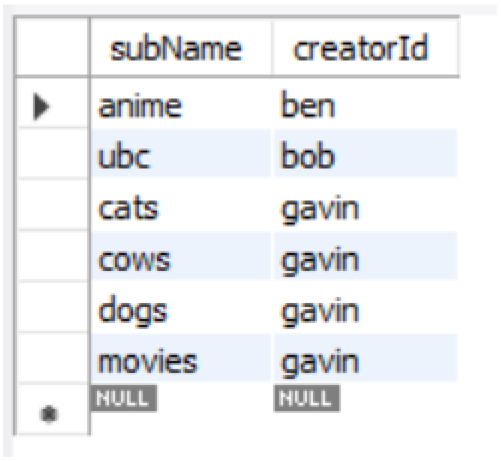
UI



Edit Subreddit Info (update)

Subreddit name: dogs
Set new owner to: gavin

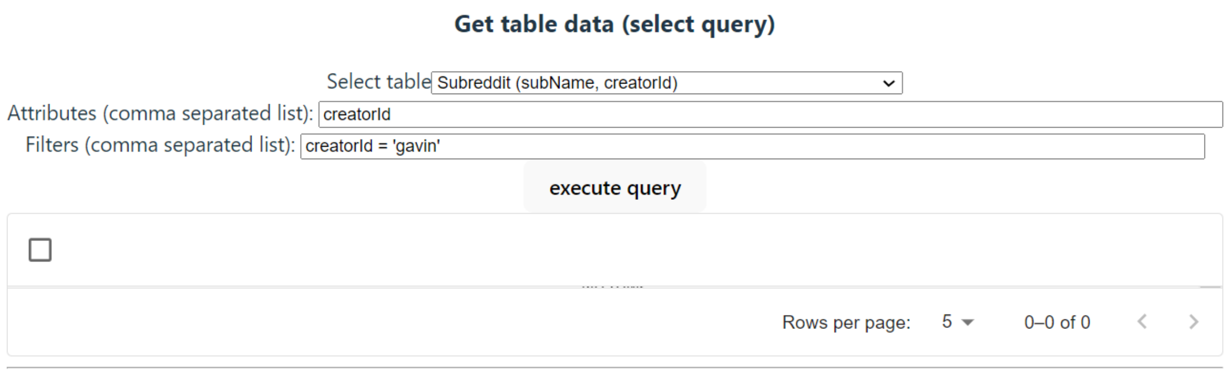
After update



	subName	creatorId
▶	anime	ben
	ubc	bob
	cats	gavin
	cows	gavin
	dogs	gavin
	movies	gavin
●	NULL	NULL

Select:

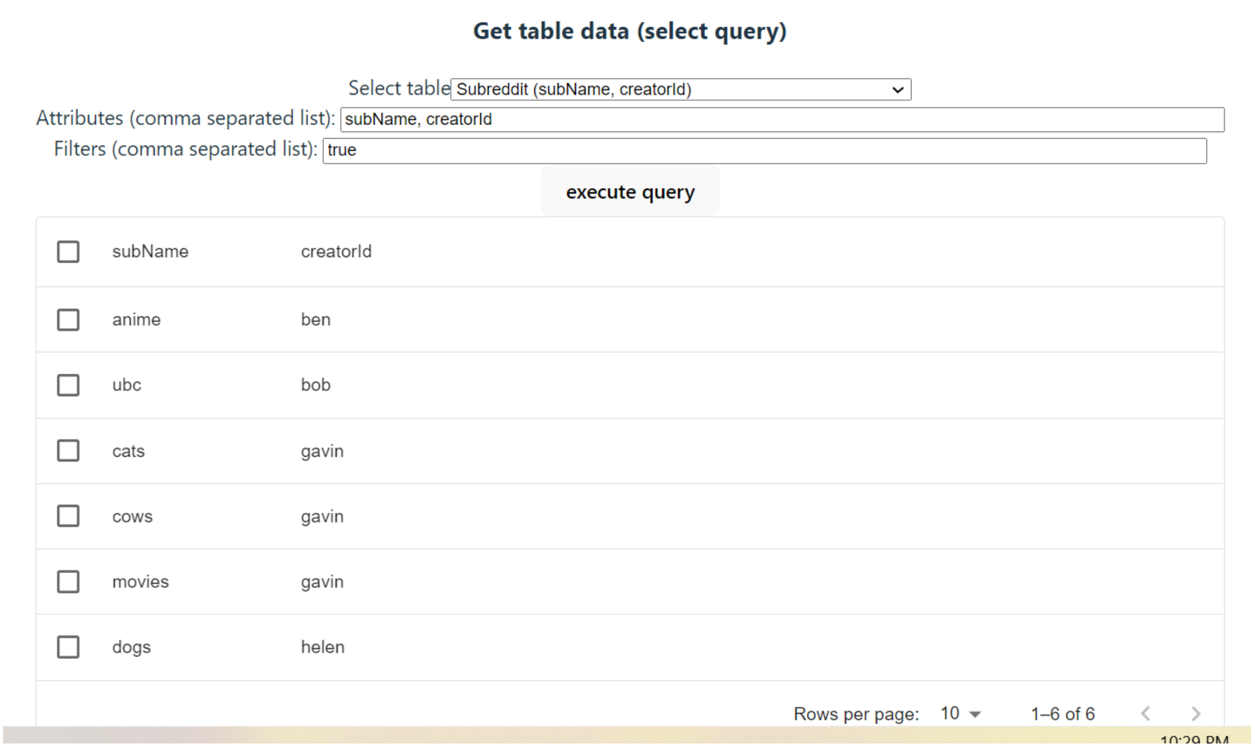
UI before query:



Get table data (select query)

Select table	Subreddit (subName, creatorId)
Attributes (comma separated list):	creatorId
Filters (comma separated list):	creatorId = 'gavin'

UI after query



Get table data (select query)

Select table: Subreddit (subName, creatorId)

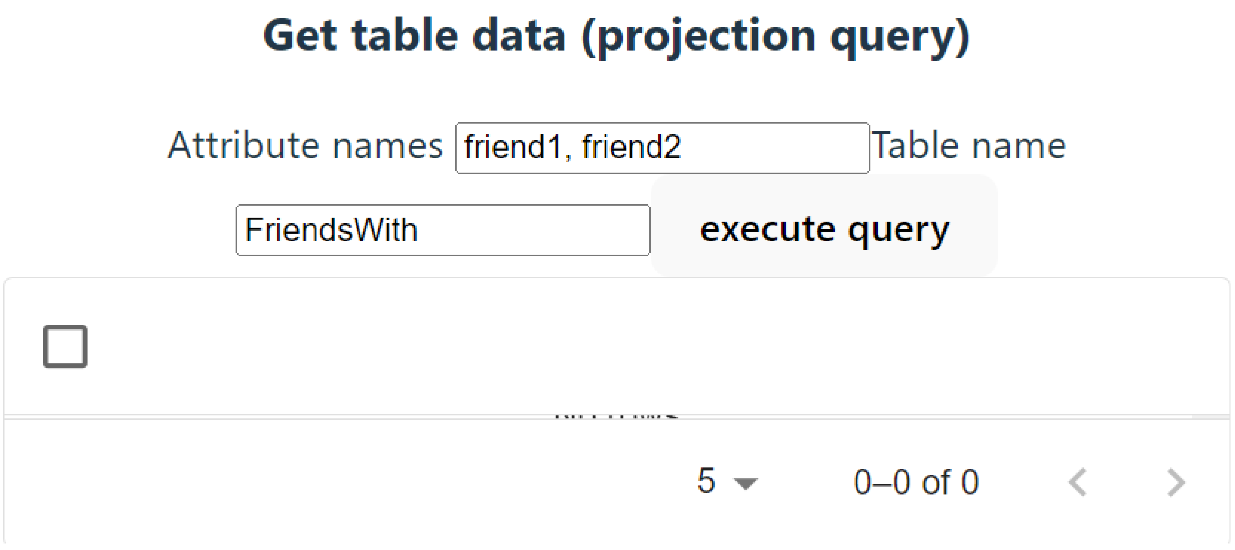
Attributes (comma separated list): subName, creatorId

Filters (comma separated list): true

subName	creatorId
anime	ben
ubc	bob
cats	gavin
cows	gavin
movies	gavin
dogs	helen

Rows per page: 10 1–6 of 6

Projection:
UI before query

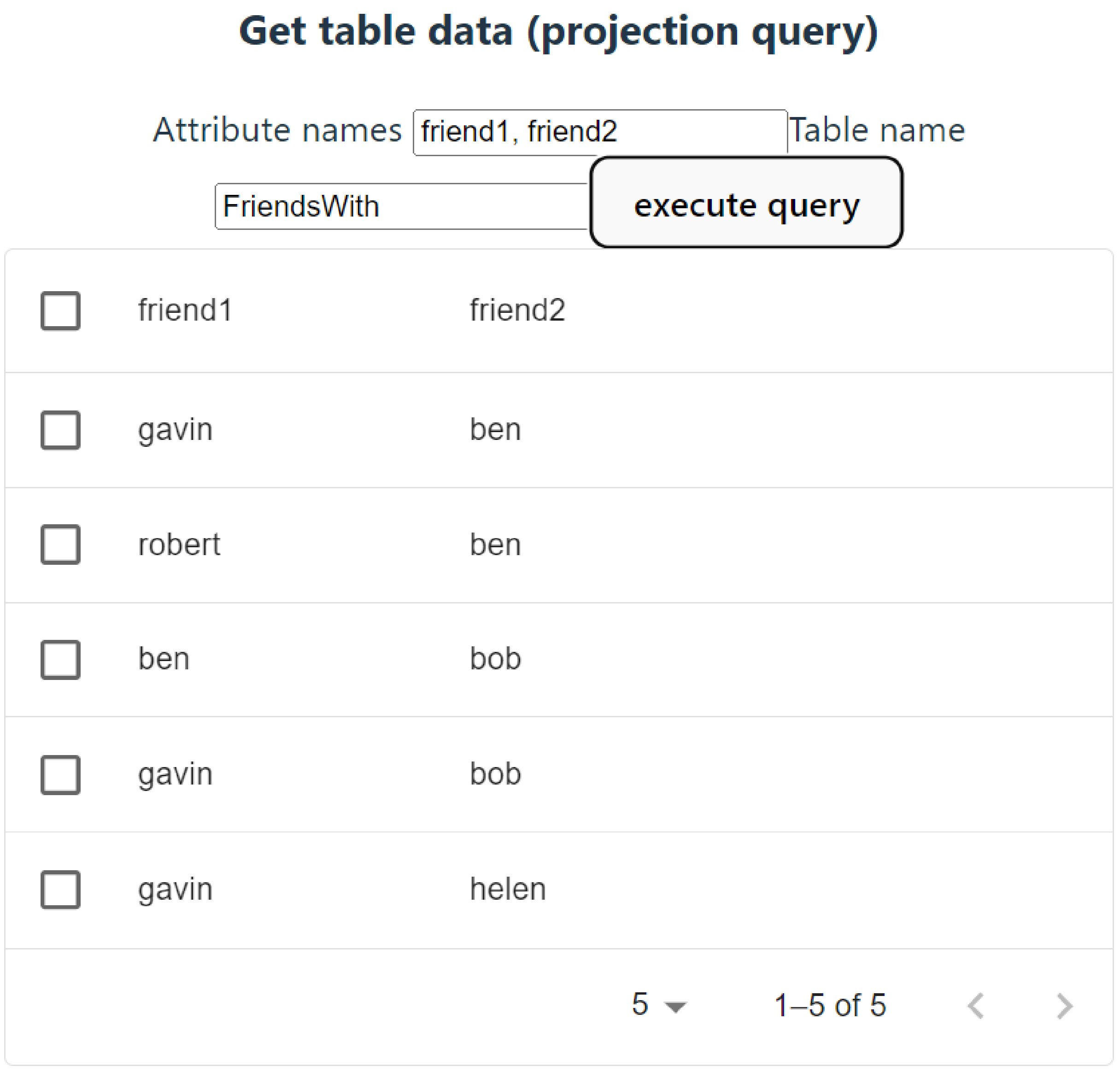


Get table data (projection query)

Attribute names: friend1, friend2

Table name: FriendsWith

UI after query



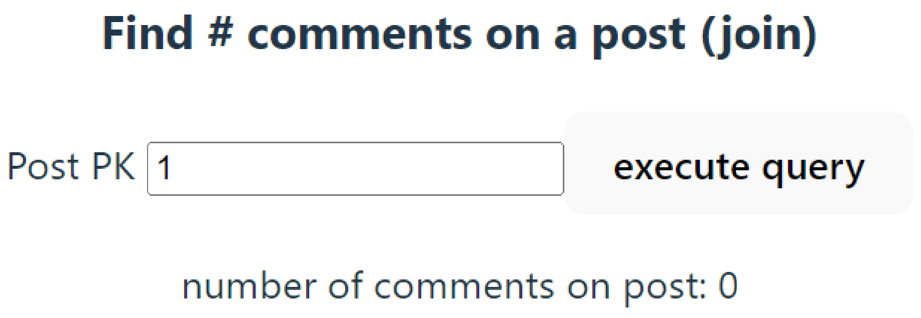
Get table data (projection query)

Attribute names: friend1, friend2
Table name: FriendsWith

	friend1	friend2
☐	gavin	ben
☐	robert	ben
☐	ben	bob
☐	gavin	bob
☐	gavin	helen

5 ▾ 1–5 of 5

Join:
before executing

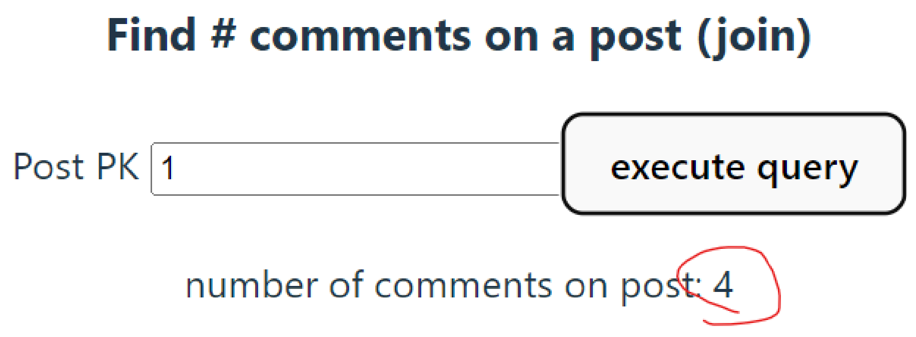


Find # comments on a post (join)

Post PK: 1

number of comments on post: 0

after executing

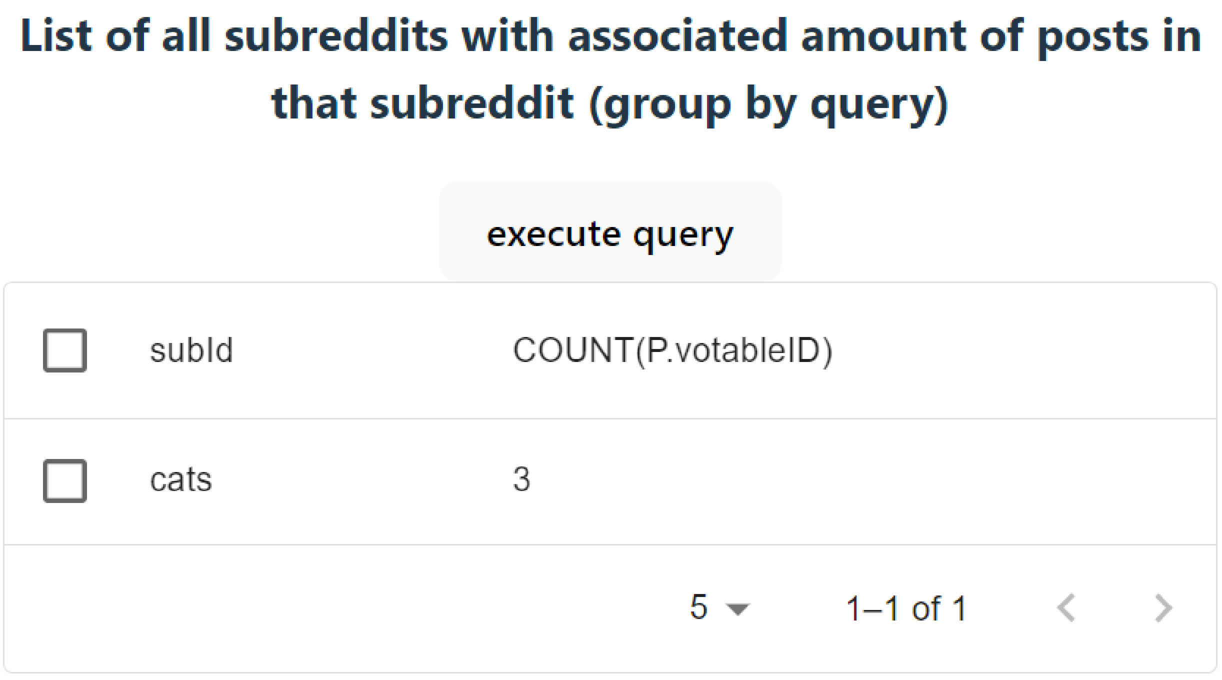


Find # comments on a post (join)

Post PK 1

number of comments on post: 4

Group by:
after executing

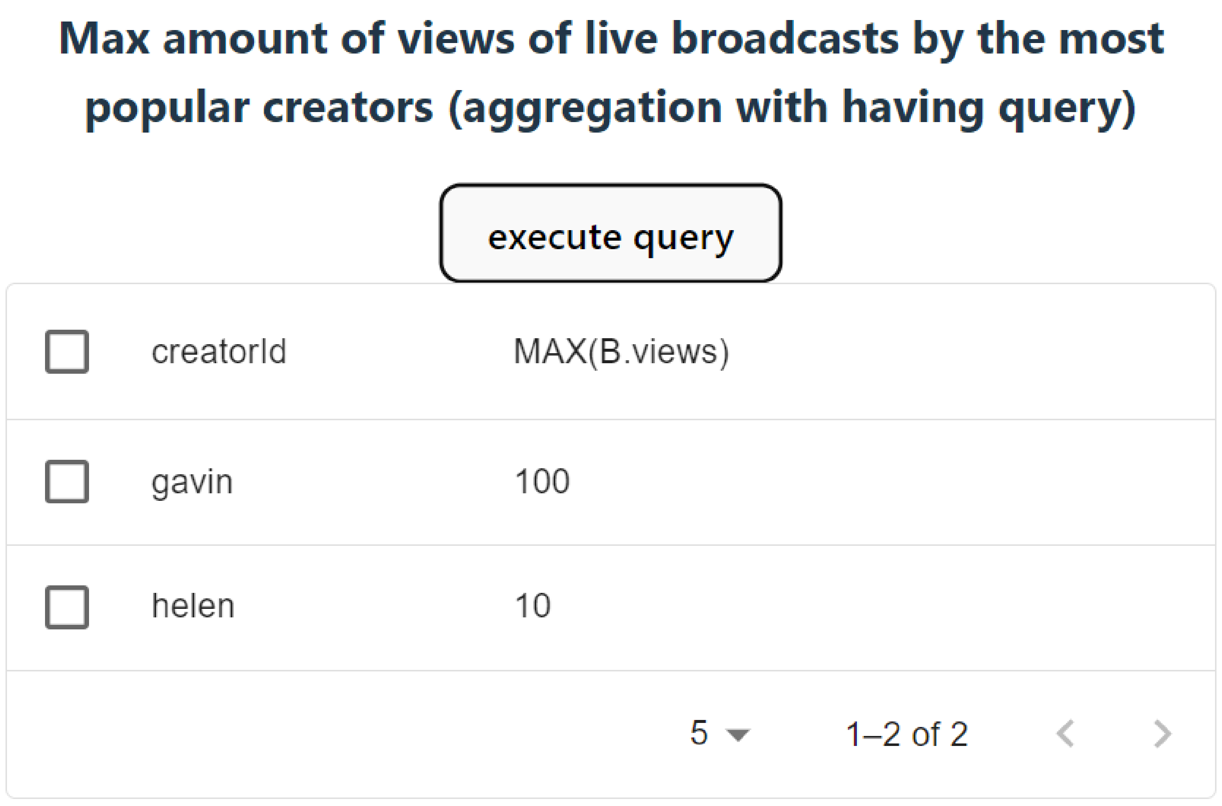


List of all subreddits with associated amount of posts in that subreddit (group by query)

subId	COUNT(P.votableID)
cats	3

5 1–1 of 1

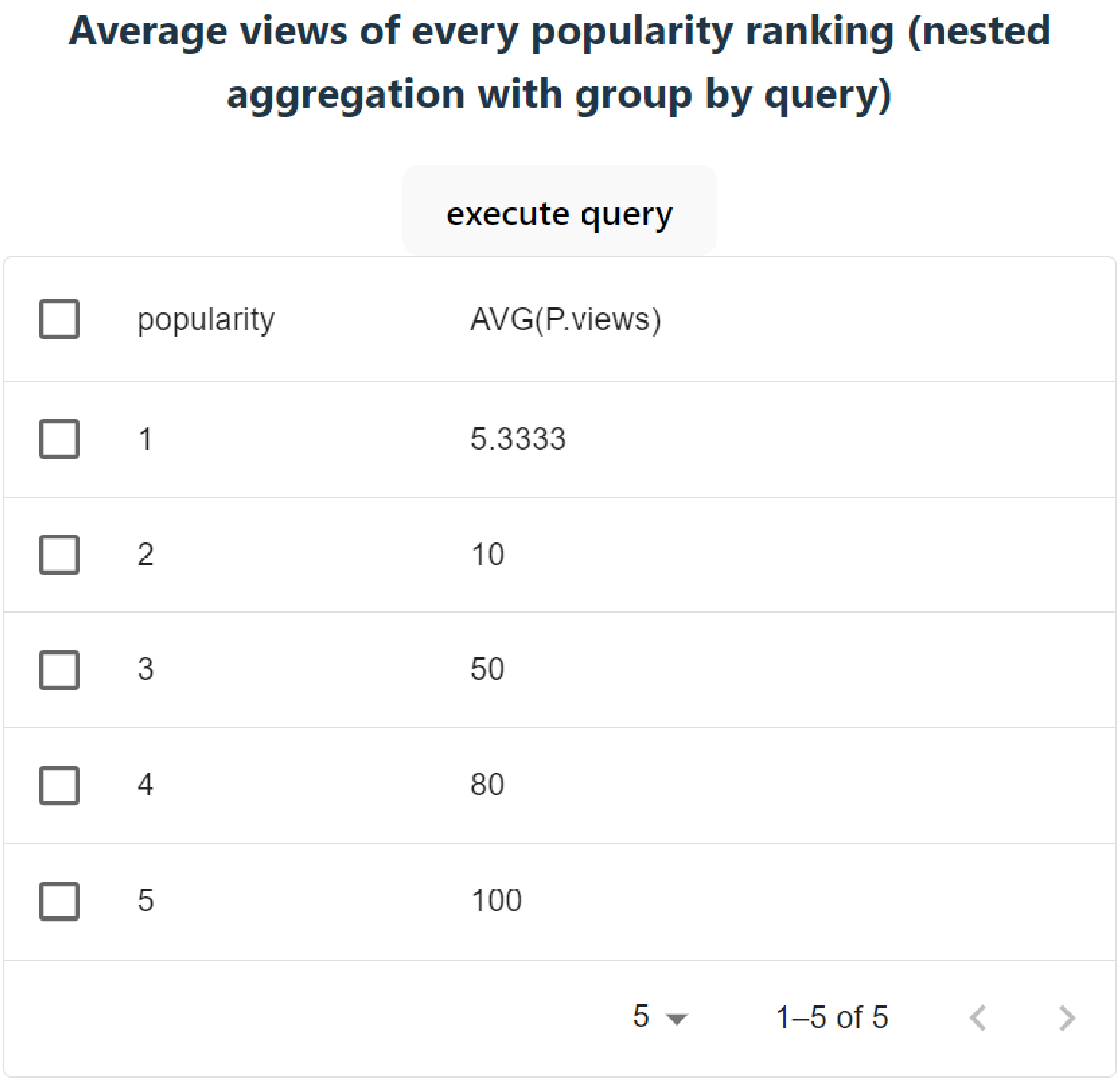
Aggregation with having:
after executing



Max amount of views of live broadcasts by the most popular creators (aggregation with having query)

creatorId	MAX(B.views)
gavin	100
helen	10

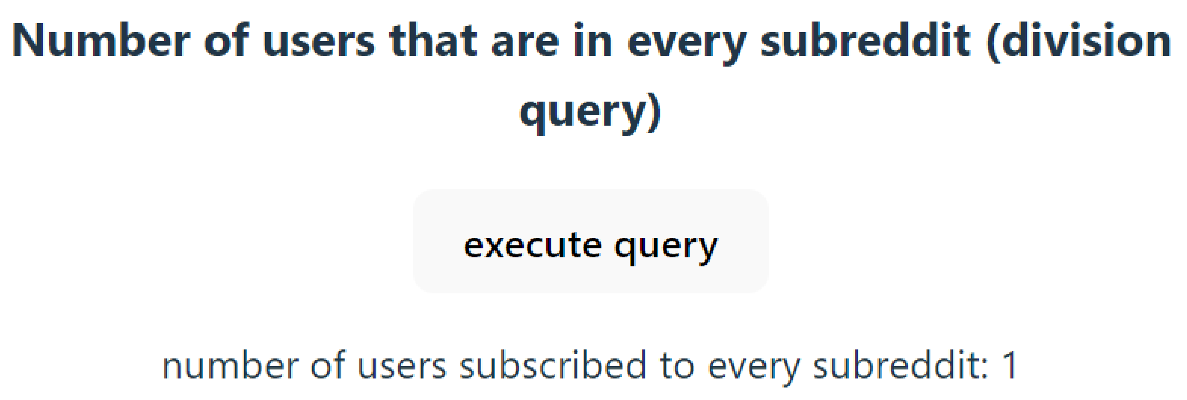
Nested aggregation:
after executing



Average views of every popularity ranking (nested aggregation with group by query)

popularity	AVG(P.views)
1	5.3333
2	10
3	50
4	80
5	100

Division:
after executing



Number of users that are in every subreddit (division query)

number of users subscribed to every subreddit: 1