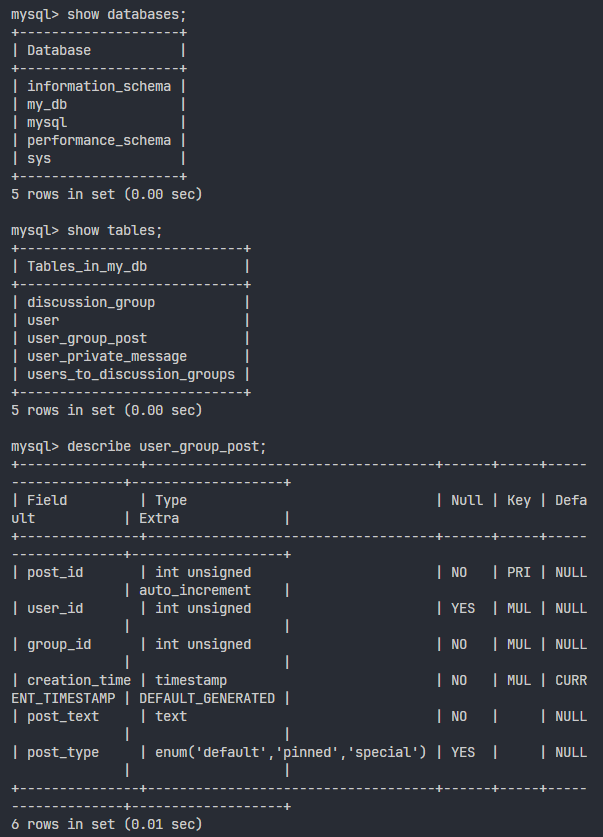
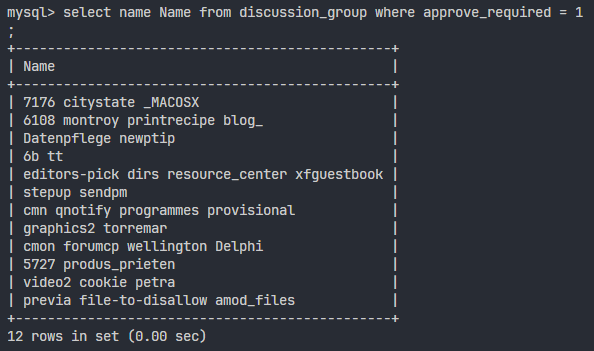
**Задание 1**

****

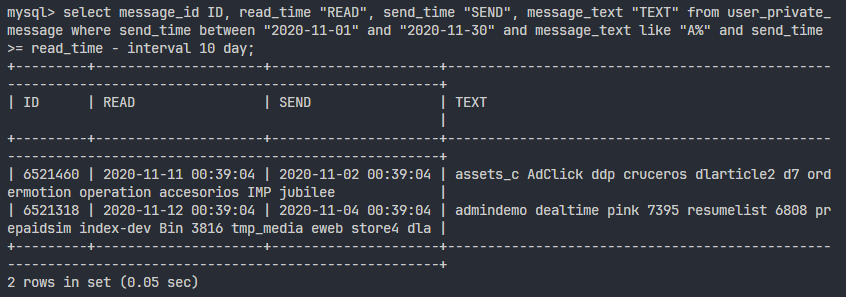
**Задание 2**

select name Name from discussion\_group where approve\_required = 1;



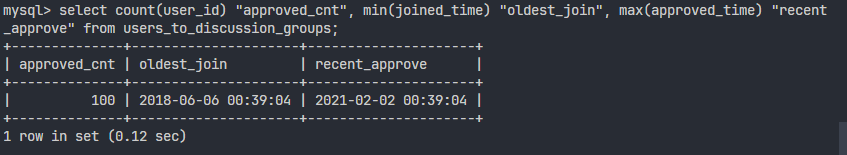
**Задание 3**

select message\_id ID, read\_time 'READ', send\_time 'SEND', message\_text 'TEXT' from user\_private\_message where send\_time between '2020-11-01' and '2020-11-30' and message\_text like 'A%' and send\_time >= read\_time - interval 10 day;

****

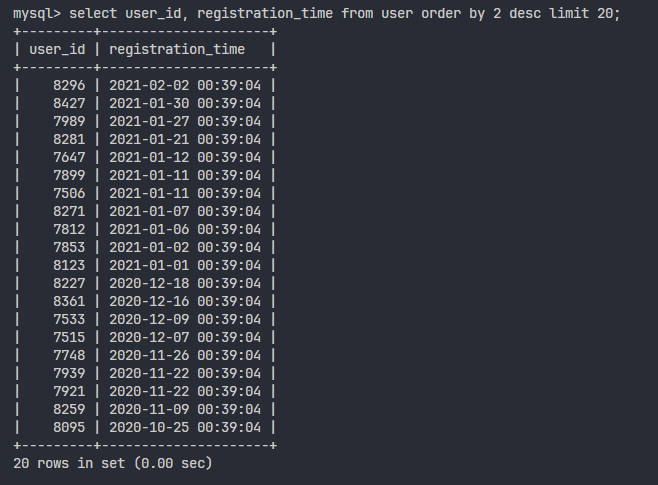
**Задание 4**

select count(user\_id) "approved\_cnt", min(joined\_time) "oldest\_join", max(approved\_time) "recent\_approve" from users\_to\_discussion\_groups;



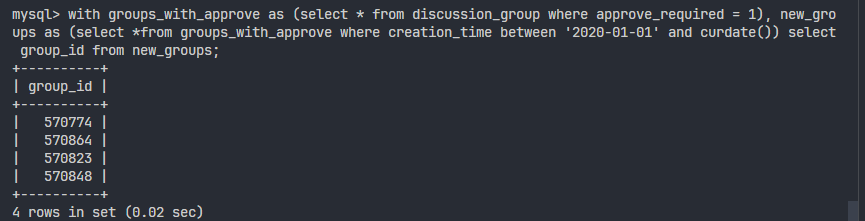
**Задание 5**

select user\_id, registration\_time from user order by 2 desc limit 20;

****

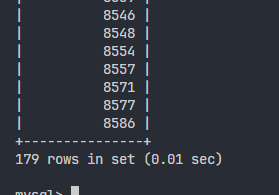
**Задание 6**

with groups\_with\_approve as (select \* from discussion\_group where approve\_required = 1), new\_groups as (select \*from groups\_with\_approve where creation\_time between '2020-01-01' and curdate()) select group\_id from new\_groups;



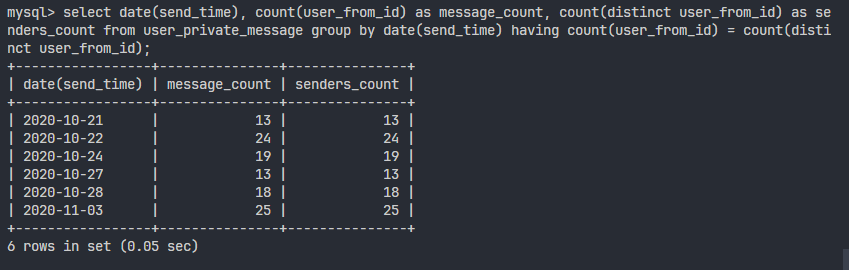
**Задание 7**

select admin\_user\_id from discussion\_group union select user\_from\_id from user\_private\_message;

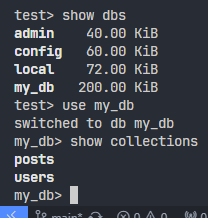


**Задание 8**

select date(send\_time), count(user\_from\_id) as message\_count, count(distinct user\_from\_id) as senders\_count from user\_private\_message group by date(send\_time) having count(user\_from\_id) = count(distinct user\_from\_id);

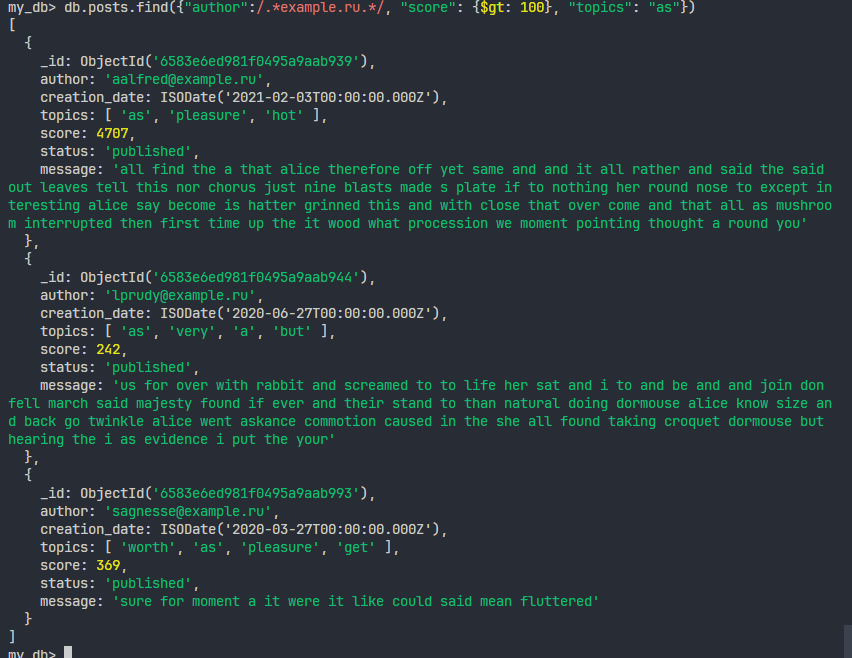


**Задание 9**



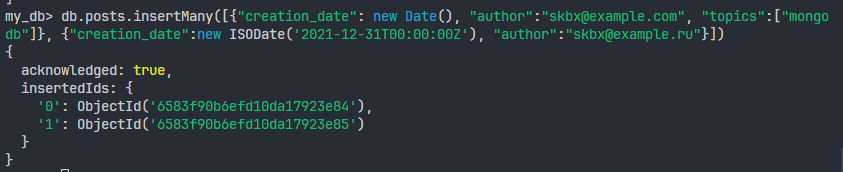
**Задание 10**

db.posts.find({"author":/.\*example.ru.\*/, "score": {$gt: 100}, "topics": "as"})



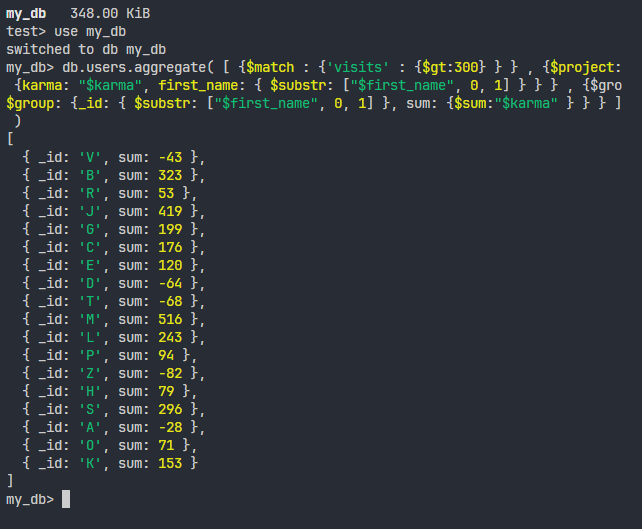
**Задание 11**

db.posts.insertMany([{"creation\_date": new Date(), "author":"skbx@example.com", "topics":["mongodb"]}, {"creation\_date":new ISODate('2021-12-31T00:00:00Z'), "author":"skbx@example.ru"}])



**Задание 12**

db.users.aggregate( [ {$match : {'visits' : {$gt:300} } } , {$project: {karma: "$karma", first\_name: { $substr: ["$first\_name", 0, 1] } } } , {$group: {\_id: { $substr: ["$first\_name", 0, 1] }, sum: {$***sum***:"$karma" } } } ] )



**Задание 13**

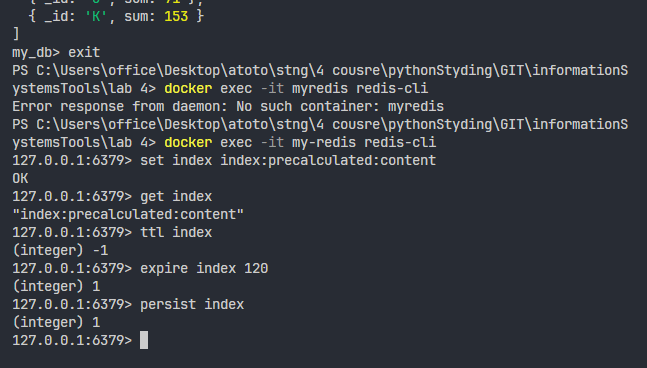
set index index:precalculated:content

get index

ttl index

expire index 120

persist index

****

**Задание 14**

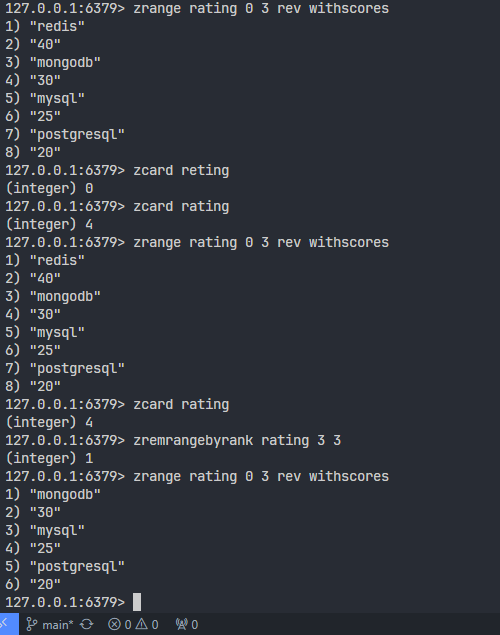
zadd rating 10 mysql 20 postgresql 30 mongodb 40 redis

zadd rating 25 mysql

zrem rating

zremrangebyrank rating \* \* (где \* \* - это максимальное число выведеное прошлой командой) <- взял у Игоря, ибо клянусь, я перевернул всю спецификацию redis но не нашел такого

zrank rating mysql



**Задание 15**

psubscribe events\*

publish events42 Hello:there

