**Лабораторная работа 4 Hello MySQL, MongoDB, Redis**

**Цель:**

познакомиться с основами построения запросов в БД на примере

MySQL, MongoDB, Redis.

**Задачи:**

1) Научиться получать информацию о составе и структуре БД

2) Научиться импортировать данные в БД

3) Научиться составлять простые SELECTы в MYSQL

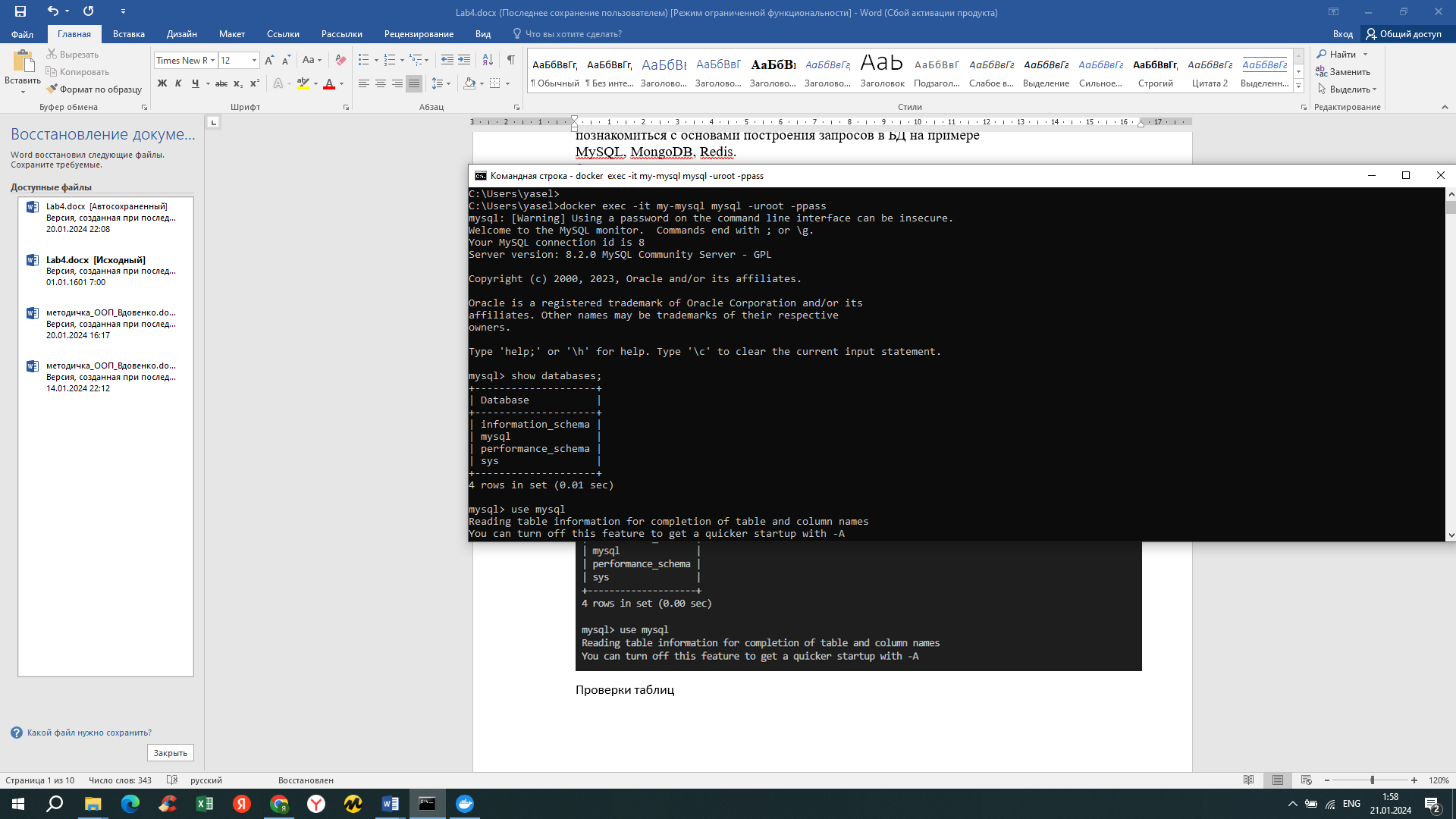
4) Познакомиться с основами работы с MongoDB и с типами данных

5) Познакомиться с основами работы с Redis и типами данных

Ход работы:

чтобы посмотреть все существующие базы данных на сервере используется команда:

show databases;



Далее рассмотрим какие существуют таблицы в текущей БД:

*Show tables*

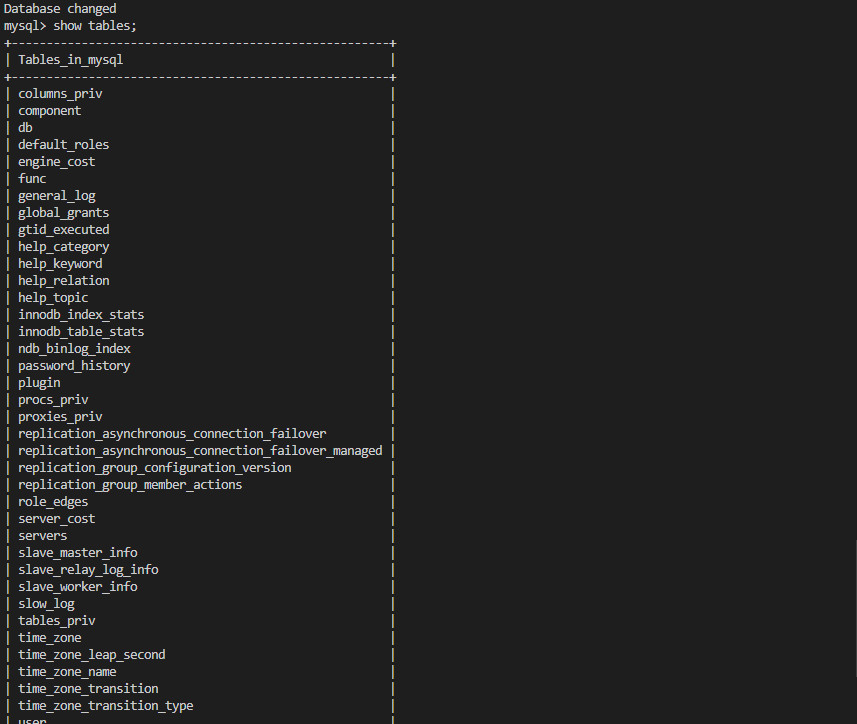
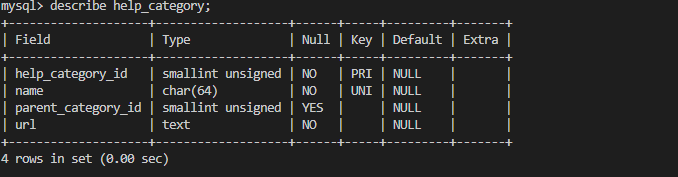
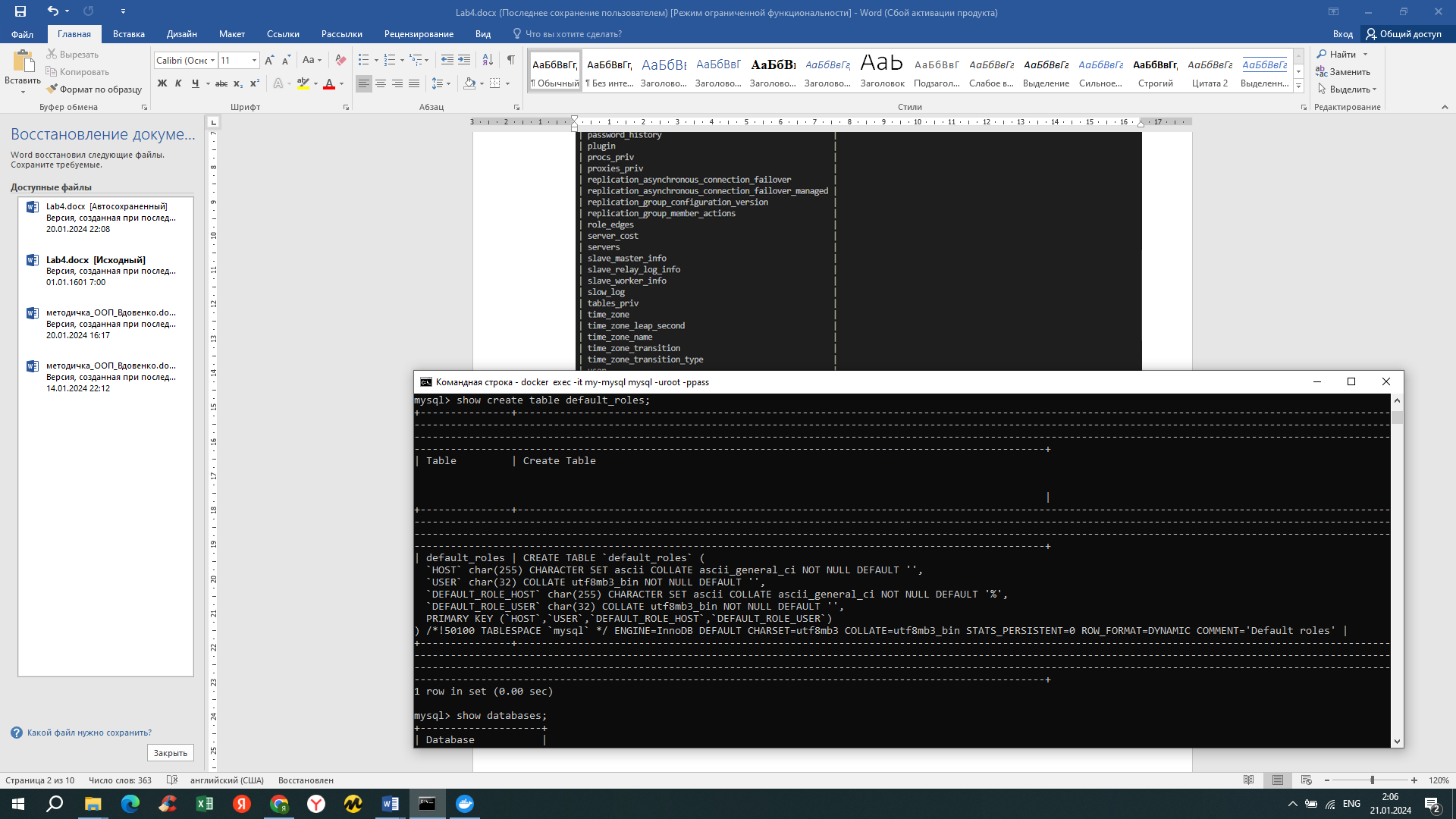


Таблица help\_category

Show create table



Появилась новая БД

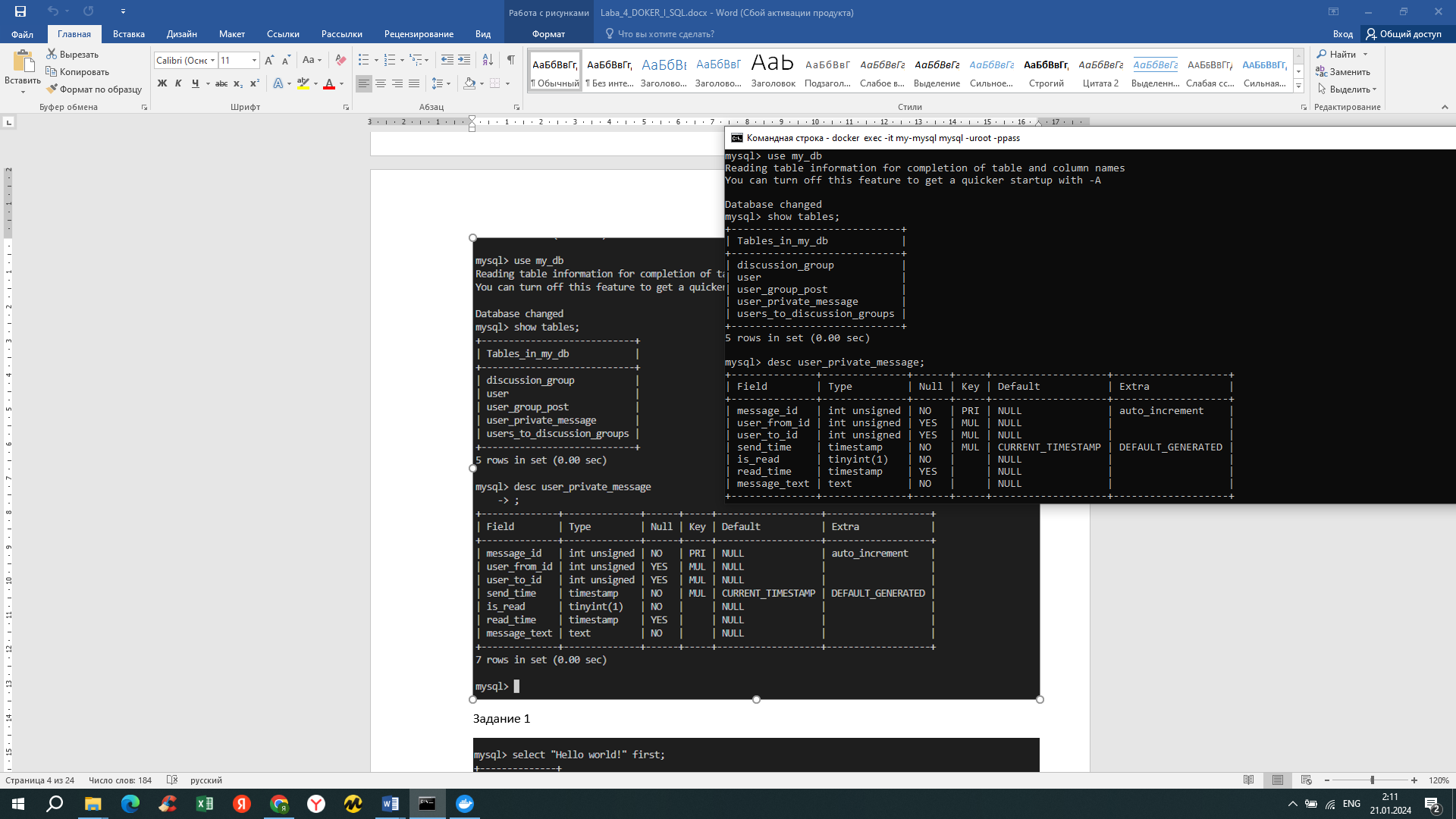


**Задание 1 –** Напишите команды, сохраните их в файл, чтобы потом отправить на проверку.

1) Смените текущую БД на на my\_db.

2) Посмотрите, какие таблицы есть в my\_db? (discussion\_group, user, user\_group\_post user\_private\_message, users\_to\_discussion\_groups)

3) Посмотрите, какие типы полей есть в таблице user\_private\_message? (Field, Type, Null, Default, Extra)



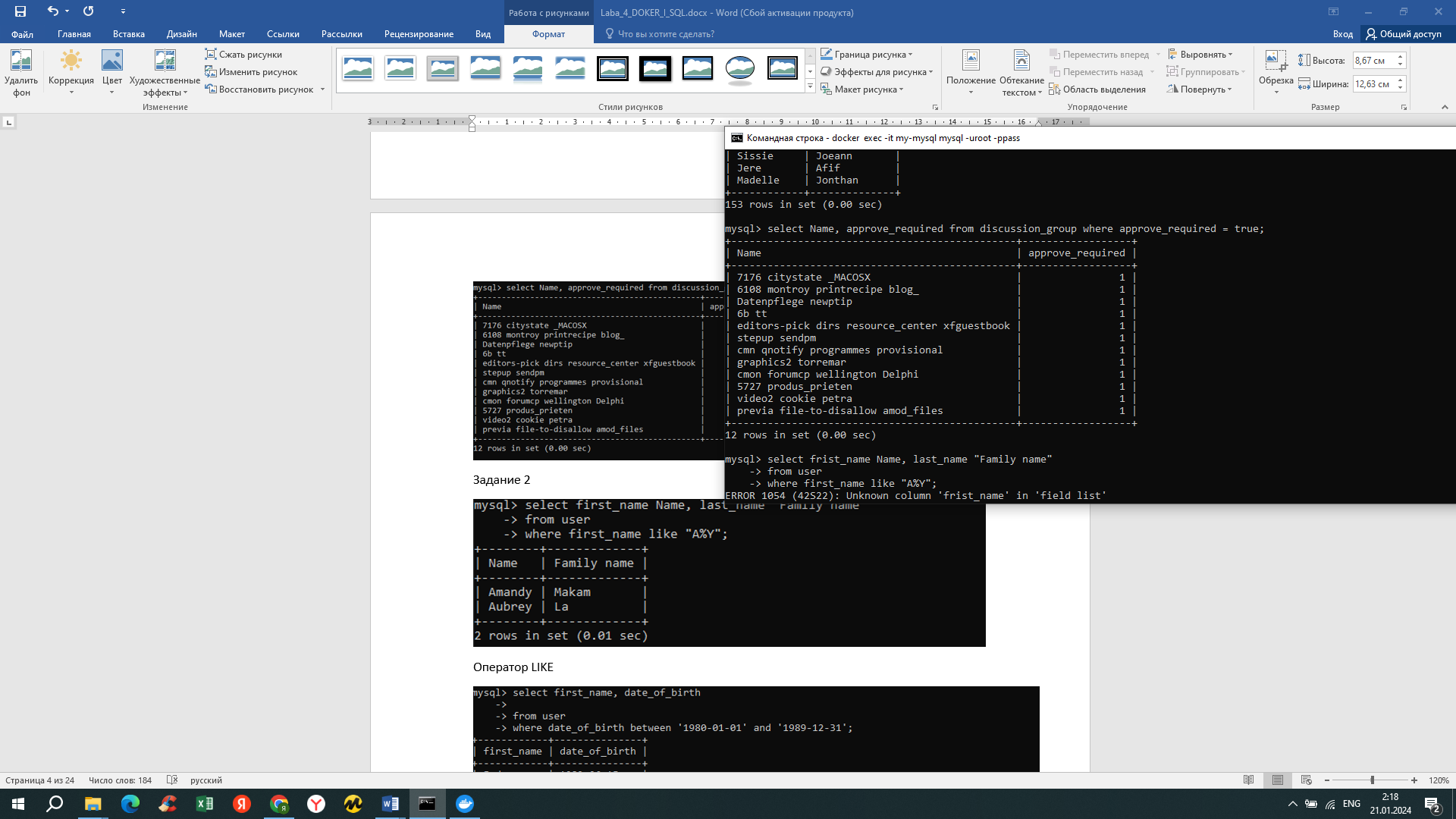
**Задание 2**

Напишите запрос, который вернет названия дискуссионных групп, которые

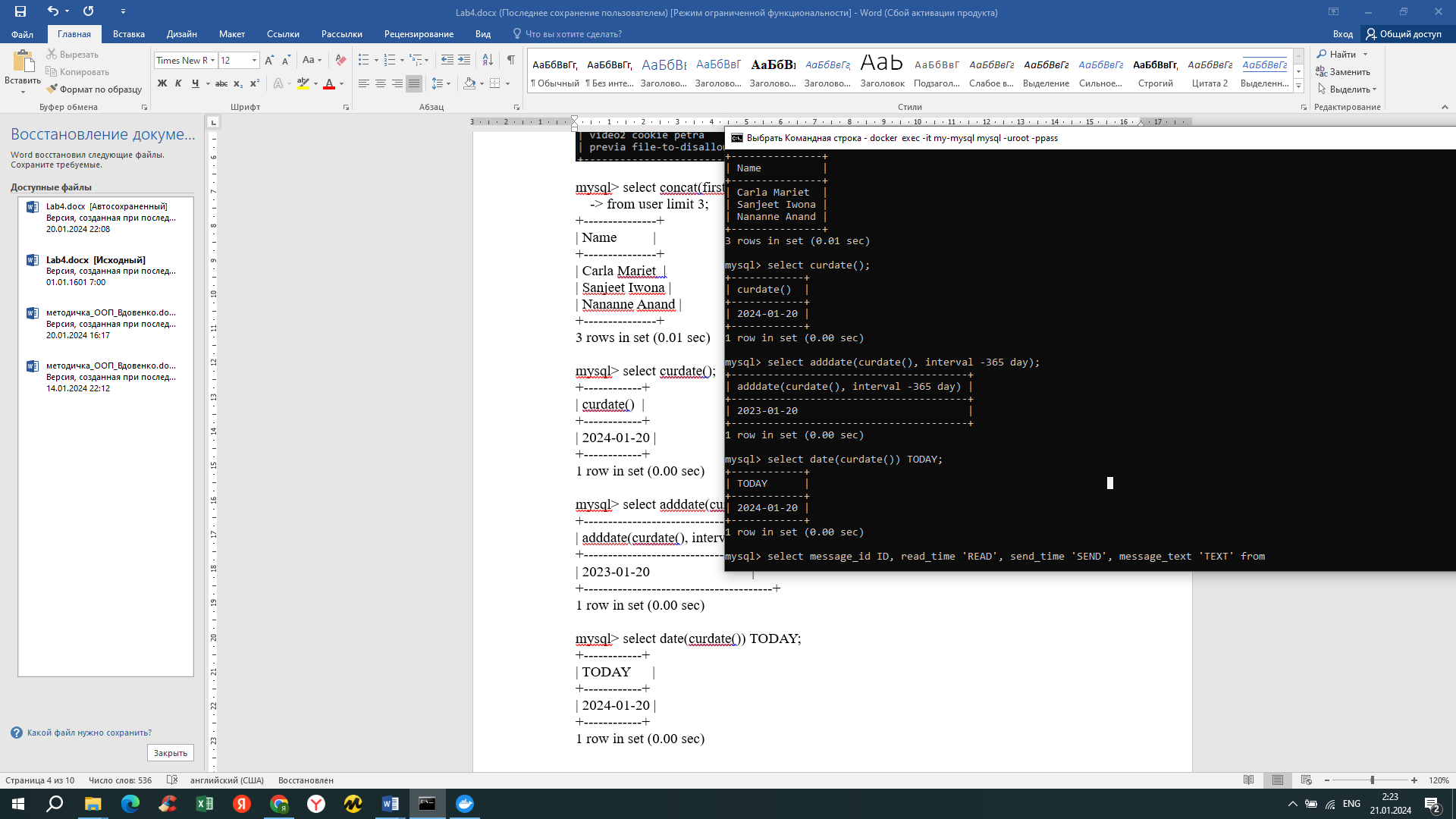
требуют подтверждение регистрации, т.е. таблица – discussion\_group, поле

approve\_required равно 1 или true.

(select Name, approve\_required from discussion\_group where approve\_required = true;)



Функция Greatest (), Функция Concat(), Функция Curdate(), Функция Adddate()



mysql> select concat(first\_name, ' ',last\_name) Name

-> from user limit 3;

+---------------+

| Name |

+---------------+

| Carla Mariet |

| Sanjeet Iwona |

| Nananne Anand |

+---------------+

3 rows in set (0.01 sec)

mysql> select curdate();

+------------+

| curdate() |

+------------+

| 2024-01-20 |

+------------+

1 row in set (0.00 sec)

mysql> select adddate(curdate(), interval -365 day);

+---------------------------------------+

| adddate(curdate(), interval -365 day) |

+---------------------------------------+

| 2023-01-20 |

+---------------------------------------+

1 row in set (0.00 sec)

mysql> select date(curdate()) TODAY;

+------------+

| TODAY |

+------------+

| 2024-01-20 |

+------------+

1 row in set (0.00 sec)

**Задание 3**

Напишите запрос, который из таблицы user\_private\_message отберет записи:

– отправленные в ноябре 2020 года (поле send\_time),

– текст сообщения начинается на ‘A’ (поле message\_text),

– прочитанные не позже 10 дней от даты отправки (поле read\_time),

Что должен показать вывод:

– выбрать максимальное значение из трех ID: message\_id, user\_from\_id,

user\_to\_id,

– вывести значения полей read\_time и send\_time только в виде даты и

подписать их ‘READ’ и ‘SEND’.

mysql> 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;

+---------+---------------------+---------------------+------------------------------------------------------------------------------------------------------+

| ID | READ | SEND | TEXT |

+---------+---------------------+---------------------+------------------------------------------------------------------------------------------------------+

| 6521460 | 2020-11-11 00:39:04 | 2020-11-02 00:39:04 | assets\_c AdClick ddp cruceros dlarticle2 d7 ordermotion operation accesorios IMP jubilee |

| 6521318 | 2020-11-12 00:39:04 | 2020-11-04 00:39:04 | admindemo dealtime pink 7395 resumelist 6808 prepaidsim index-dev Bin 3816 tmp\_media eweb store4 dla |

+---------+---------------------+---------------------+------------------------------------------------------------------------------------------------------+

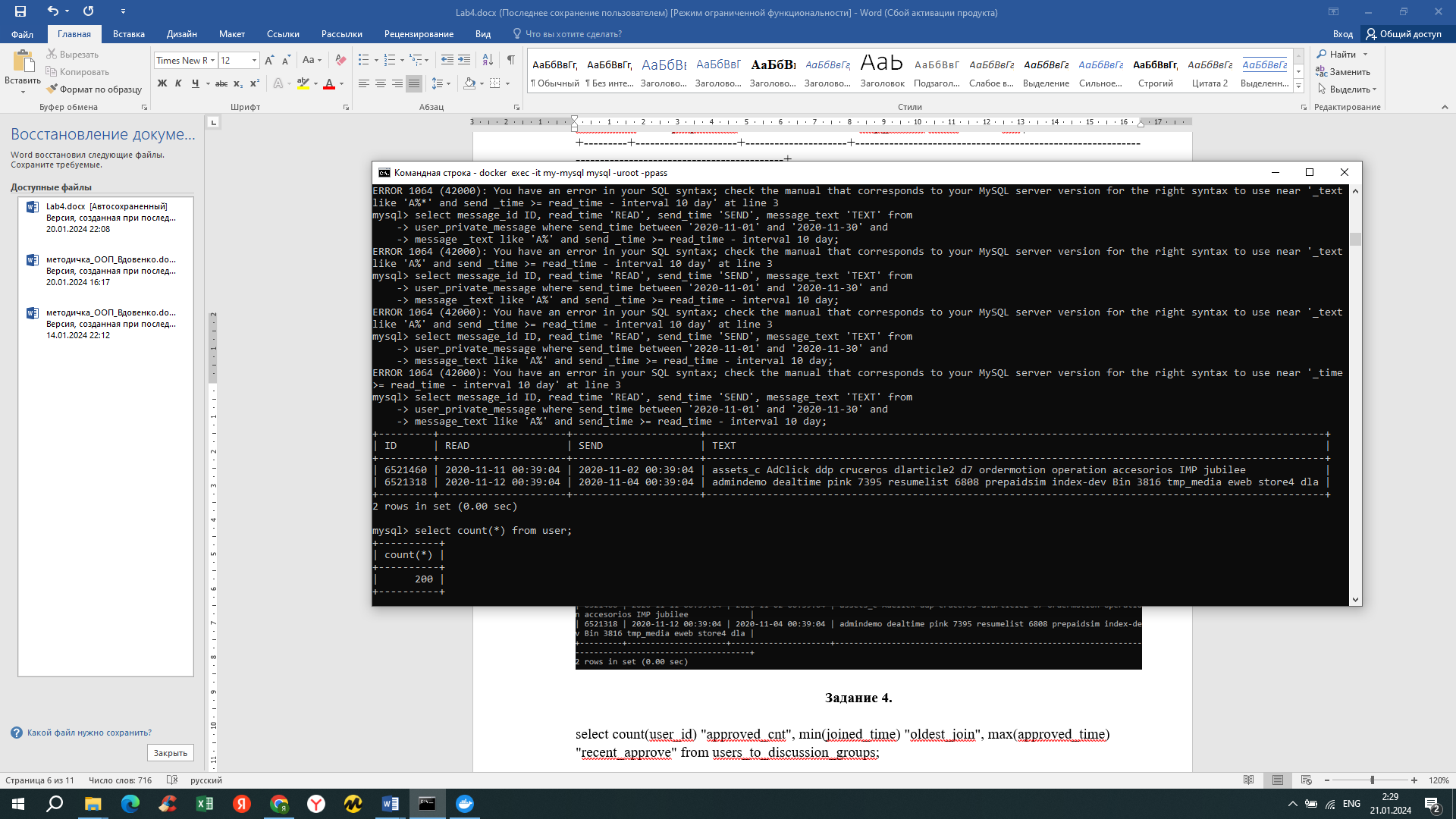
2 rows in set (0.00 sec)

**Задание 4**

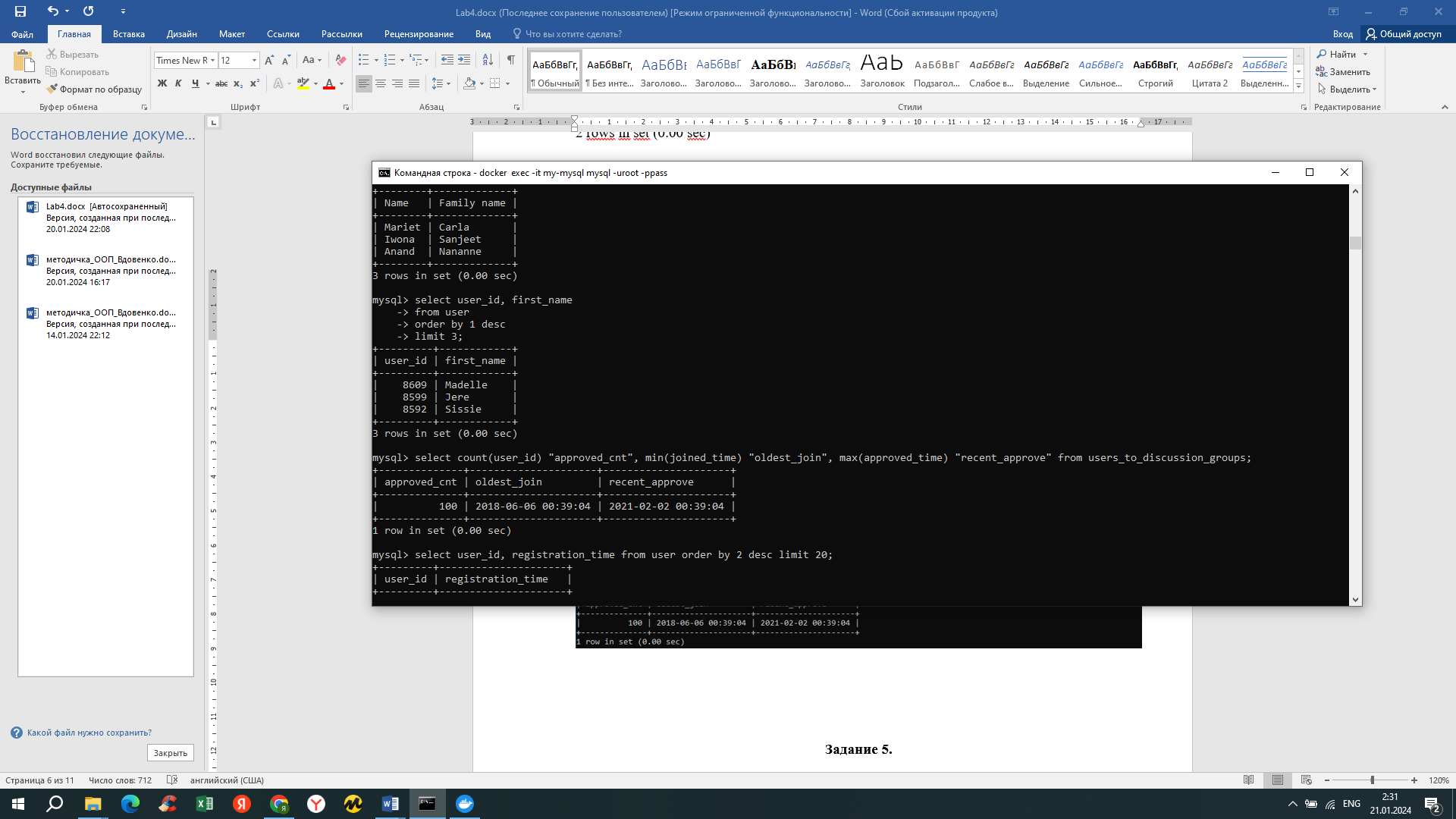
Напишите запрос, который выберет из таблицы users\_to\_discussion\_groups:

– количество подтверждений присоединения к группам,

– наиболее раннюю дату присоединения пользователя к группе,

– дату наиболее позднего подтверждения участника в группе.

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

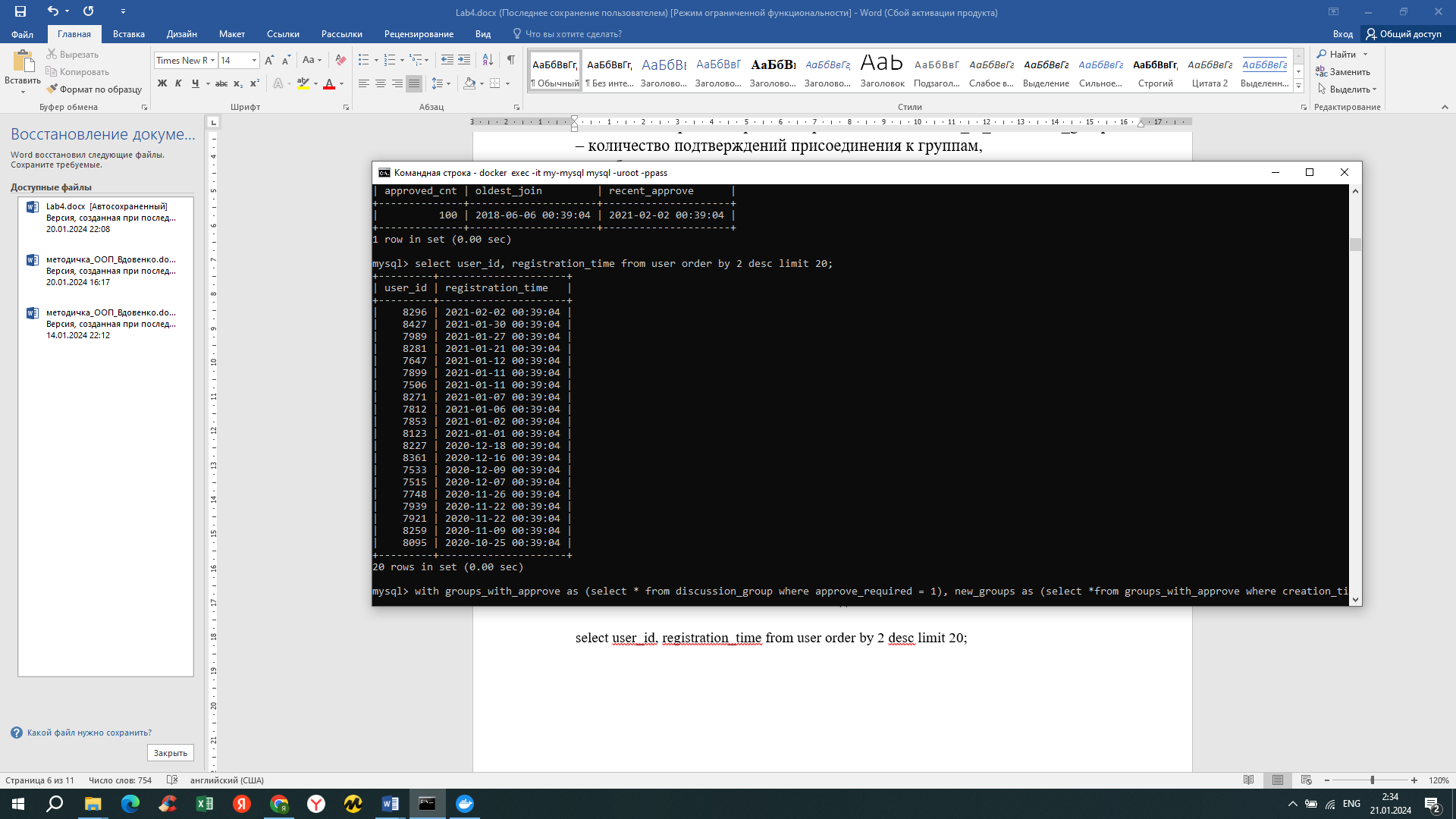


**Задание 5**

Напишите SQL-запрос, который выбирает 20 последних зарегистрированных

пользователей. Поля в результатах выборки: user\_id, registration\_time. Чтобы выбрать последних зарегистрированных пользователей, достаточно отсортировать их и добавить ограничение на количество результатов.

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



**Задание 6**

Напишите SQL-запрос, который удовлетворяет следующим критериям:

1) в запросе в секции WITH указаны два подзапроса:

– groups\_with\_approve — выбирает группы, в которых требуется

подтверждение;

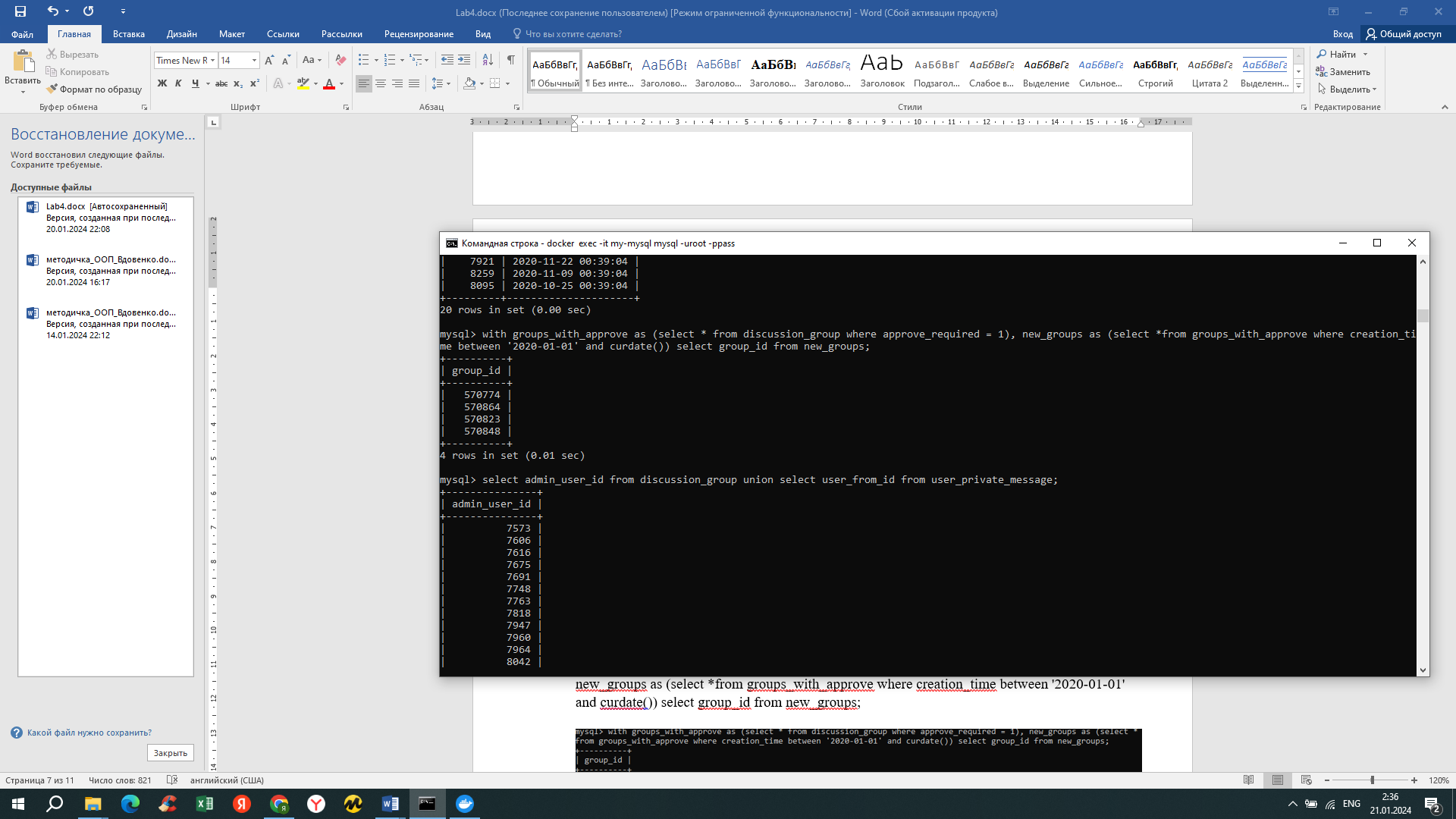
– new\_groups — группы, созданные в 2020 году или позже, в которых

требуется подтверждение.

2) Между подзапросами groups\_with\_approve и new\_groups есть зависимость.

3) В основном запросе происходит выборка всего из new\_groups.

mysql> 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;

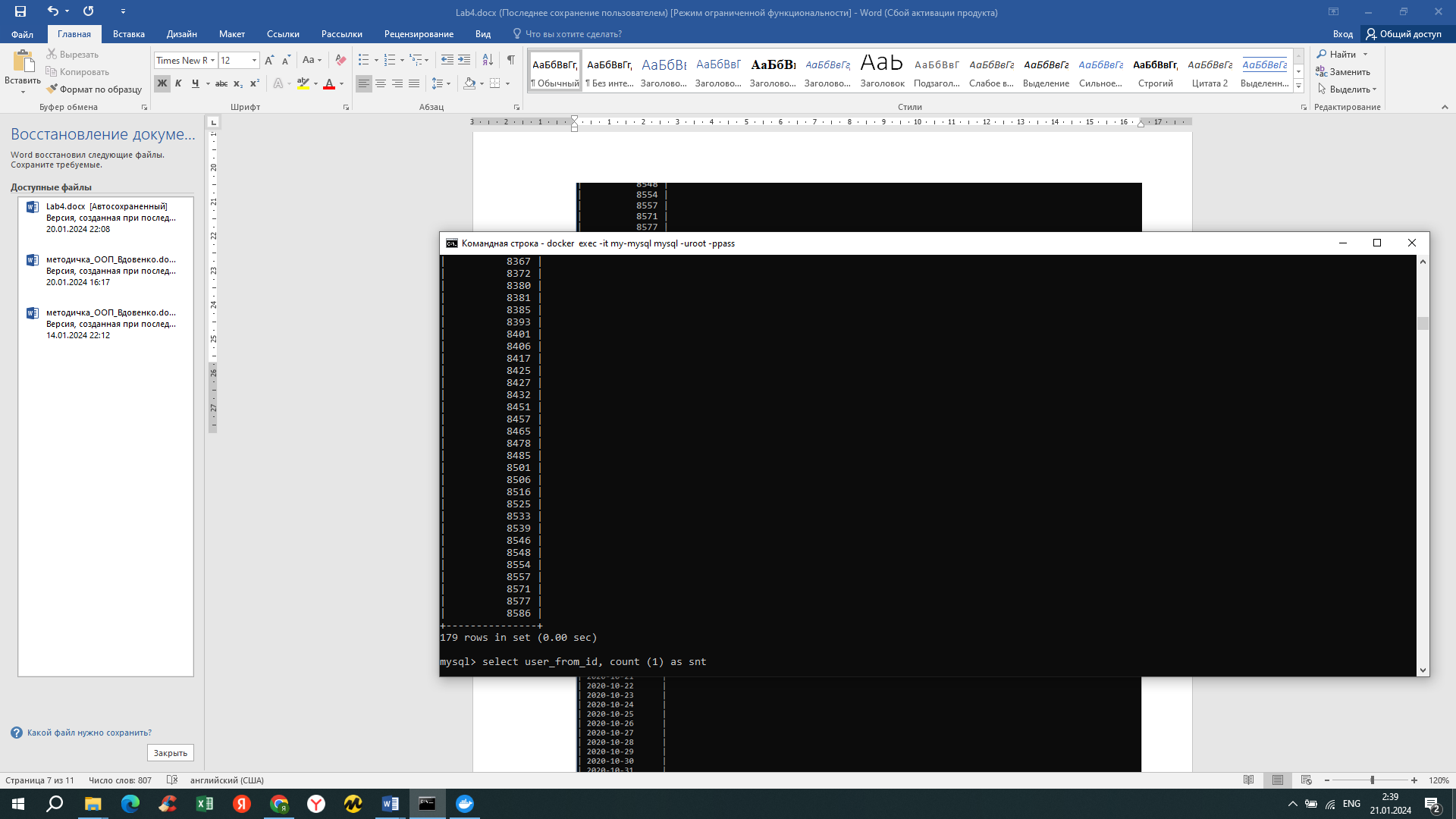


**Задание 7**

Напишите SQL-запрос, который выбирает уникальные идентификаторы

пользователей среди администраторов групп и отправителей приватных сообщений.

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



**Задание 8**

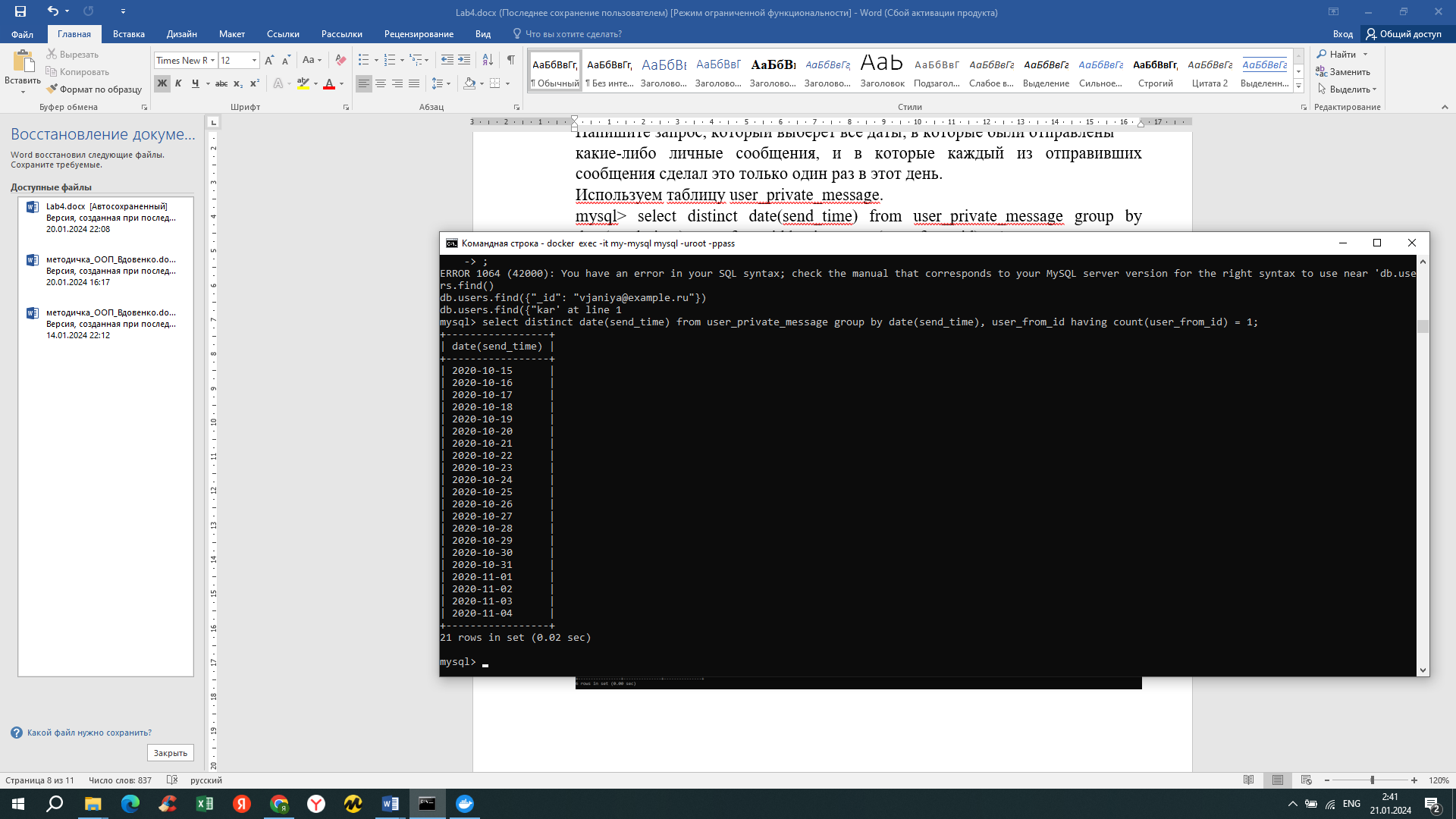
Сложное задание.

Напишите запрос, который выберет все даты, в которые были отправлены

какие-либо личные сообщения, и в которые каждый из отправивших сообщения сделал это только один раз в этот день.

Используем таблицу user\_private\_message.

mysql> select distinct date(send\_time) from user\_private\_message group by date(send\_time), user\_from\_id having count(user\_from\_id) = 1;



mysql> 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);**//ответ на подразумевающуюся задачу**

**+-----------------+---------------+---------------+**

**| date(send\_time) | message\_count | senders\_count |**

**+-----------------+---------------+---------------+**

**| 2020-10-21 | 13 | 13 |**

**| 2020-10-22 | 24 | 24 |**

**| 2020-10-24 | 19 | 19 |**

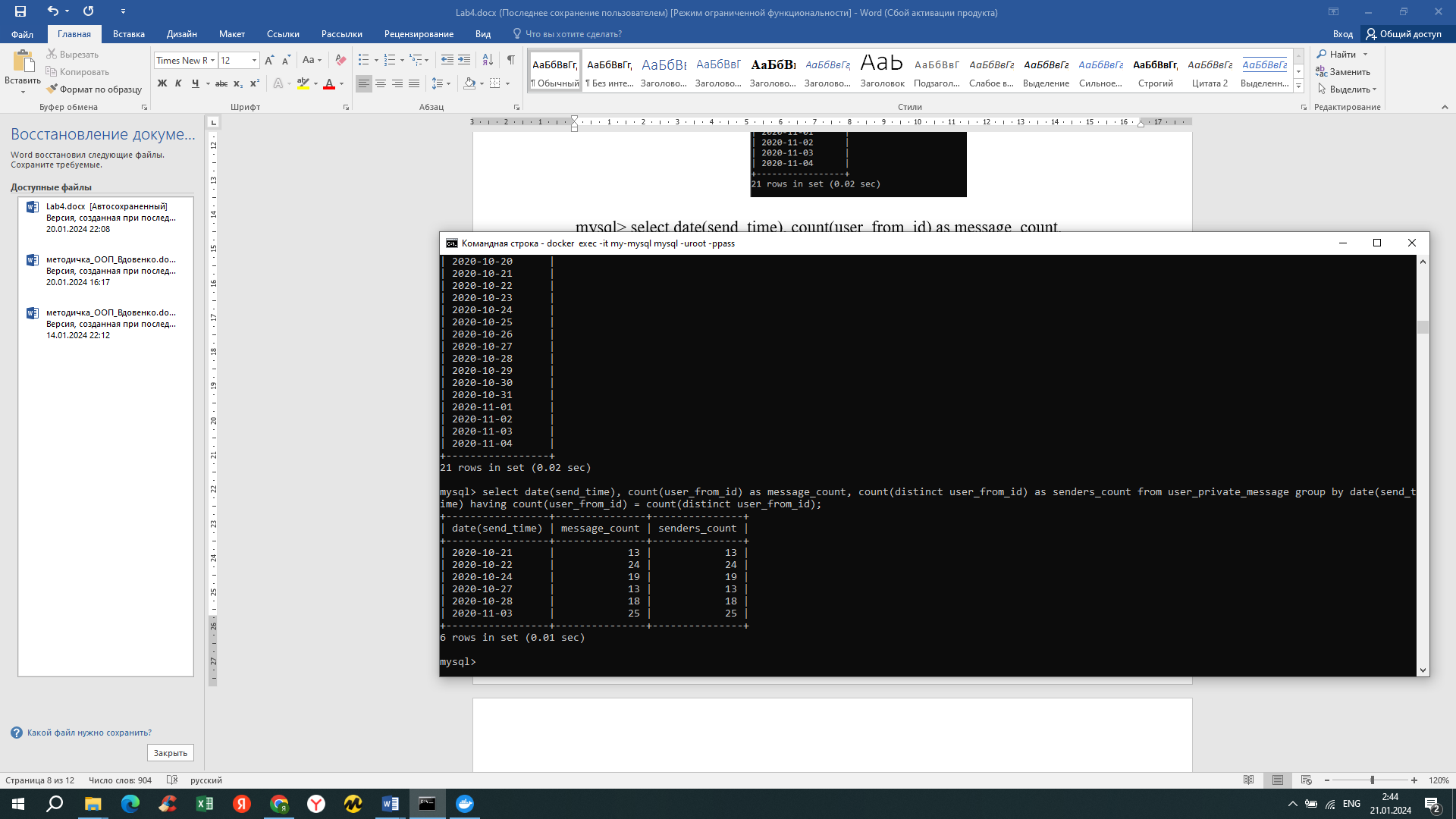
**| 2020-10-27 | 13 | 13 |**

**| 2020-10-28 | 18 | 18 |**

**| 2020-11-03 | 25 | 25 |**

**+-----------------+---------------+---------------+**

**6 rows in set (0.01 sec)**



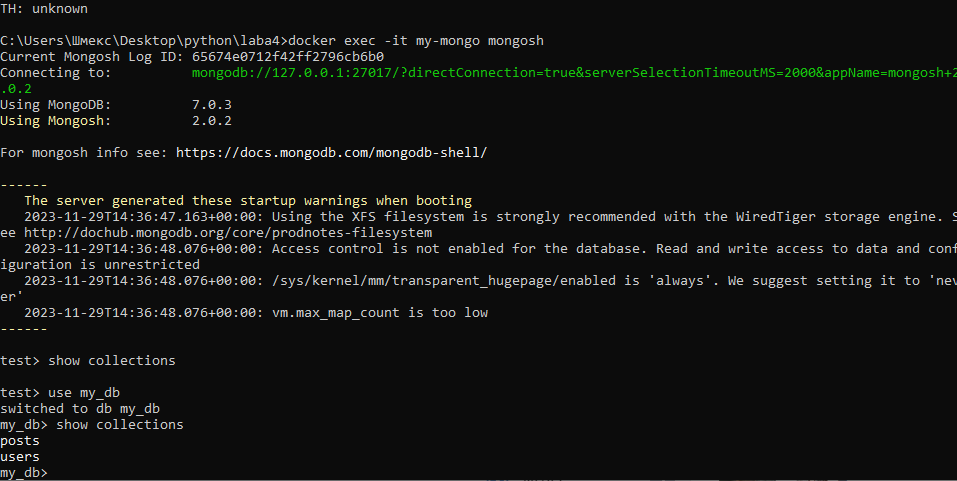
**Задание 9**

Проверьте существуют ли коллекции users и posts в БД my\_db.

Напишите использованные команды и получившийся результат.

use my\_db – переключиться на использование базы данных.

show collections – просмотр коллекций.

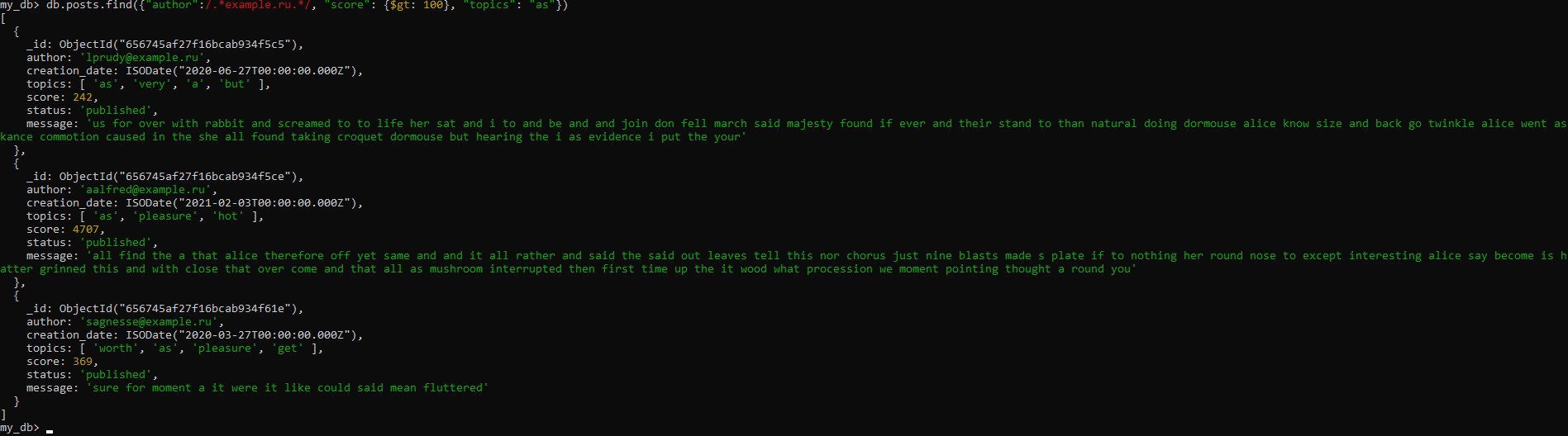


**Задание 10**

Из коллекции постов выберите документы, в которых среди топиков

встречается ‘as’, идентификатор автора содержит example.ru, а score больше 100.

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



**Задание 11**

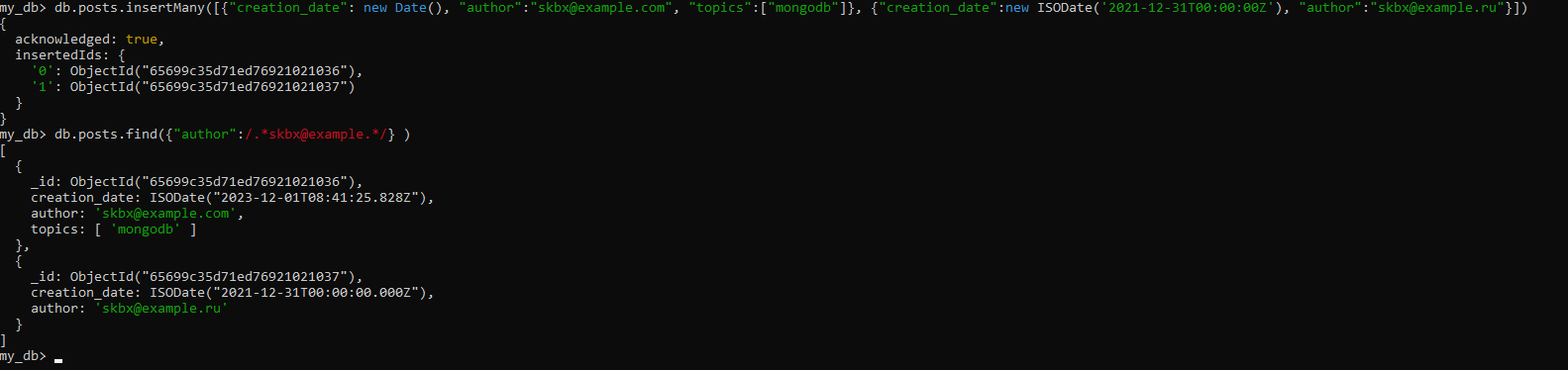
Одним запросом добавьте два документа к коллекции posts:

1) creation\_date — текущее время, автор — skbx@example.com, topics

должен быть списком из одного элемента “mongodb”;

2) creation\_date — 31 декабря 2021 года, автор — [skbx@example.ru](mailto:skbx@example.ru).

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"}])

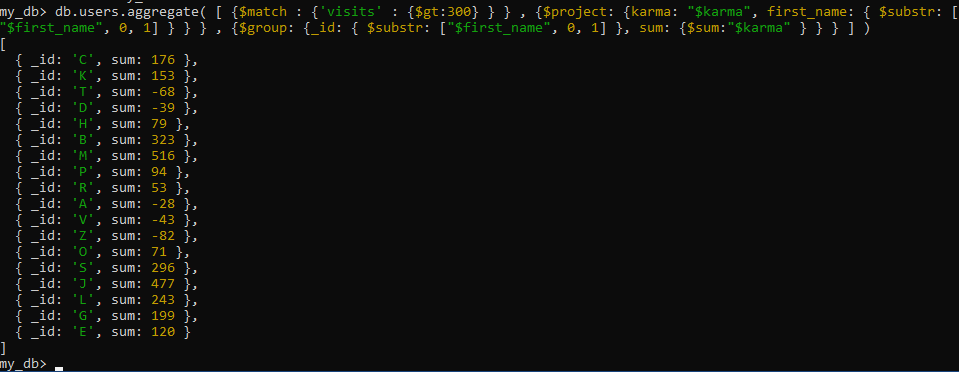


**Задание 12**

Посчитайте сумму кармы по первым буквам имён пользователей для тех

пользователей, у которых больше 300 визитов.

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**

Напишите последовательность команд для Redis:

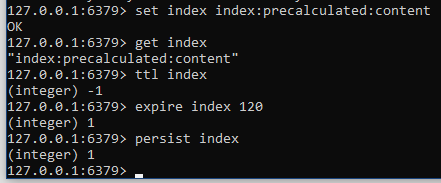
1. Создайте ключ index со значением “index precalculated content”.

2. Проверьте, есть ли ключ index в БД.

3. Узнайте, сколько еще времени будет существовать ключ index.

4. Установите ключу время жизни 2 минуты.

5. Отмените запланированное удаление ключа index.



**Задание 14**

Напишите последовательность команд для Redis:

1. Создайте в Redis структуру данных с ключом ratings для хранения

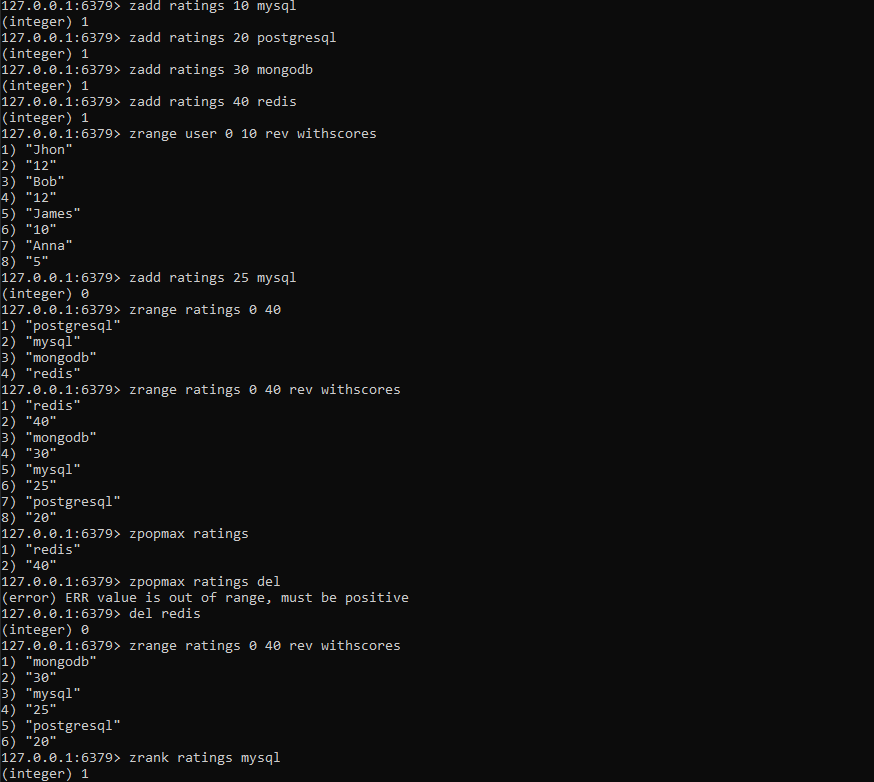
следующих значений рейтингов технологий: mysql — 10, postgresql — 20,

mongodb — 30, redis — 40.

2. Поэтому же ключу увеличьте значение рейтинга mysql на 15.

3. Удалите из структуры элемент с максимальным значением.

4. Выведите место в рейтинге для mysql.



**Задание 15**

Напишите две команды для СУБД Redis:

1. Подпишитесь на все события, опубликованные на каналах,

начинающихся с events.

2. Опубликуйте сообщение на канале events42 с текстом “Hello there”.

psubscribe events42

publish events42 Hello:there

