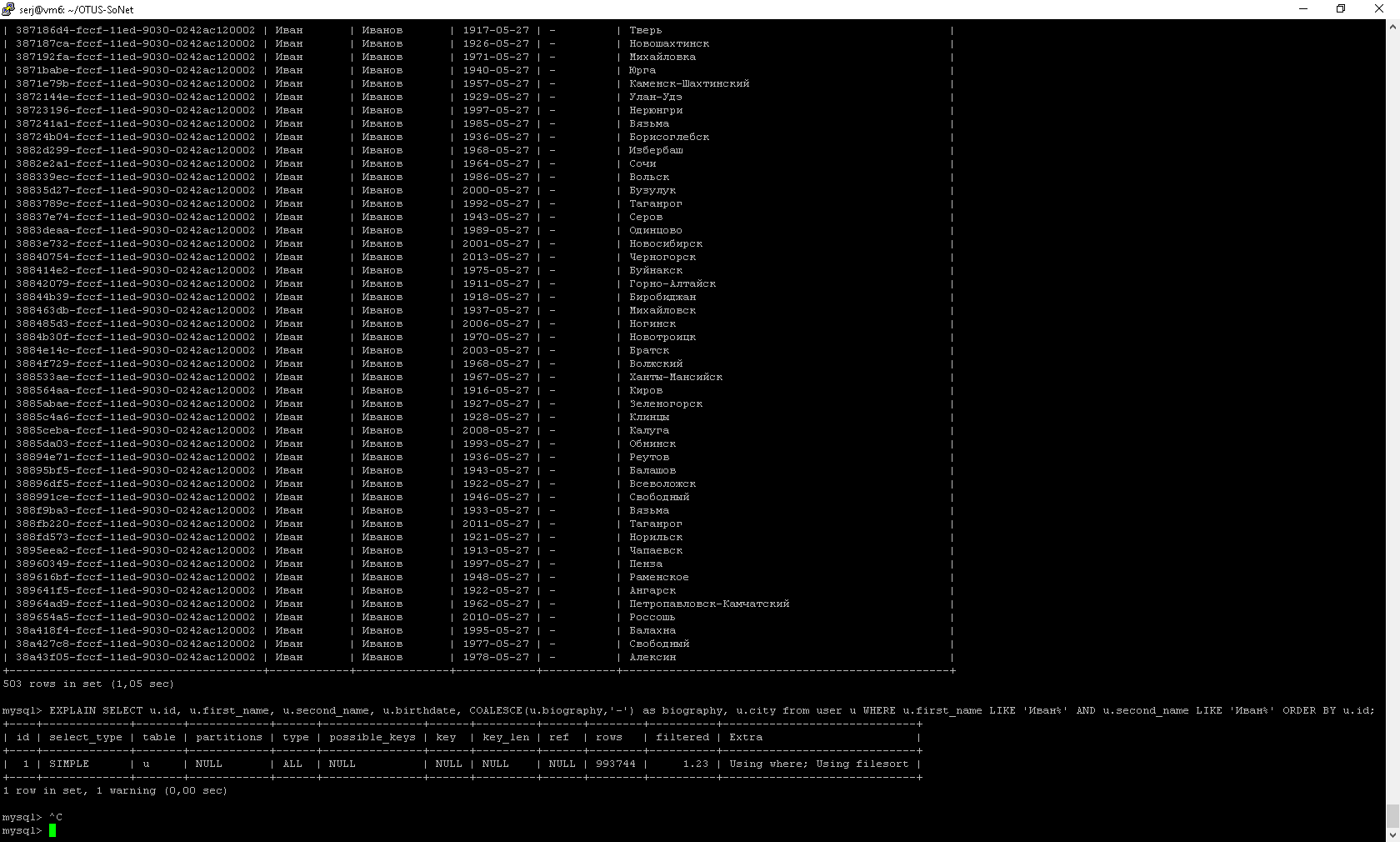
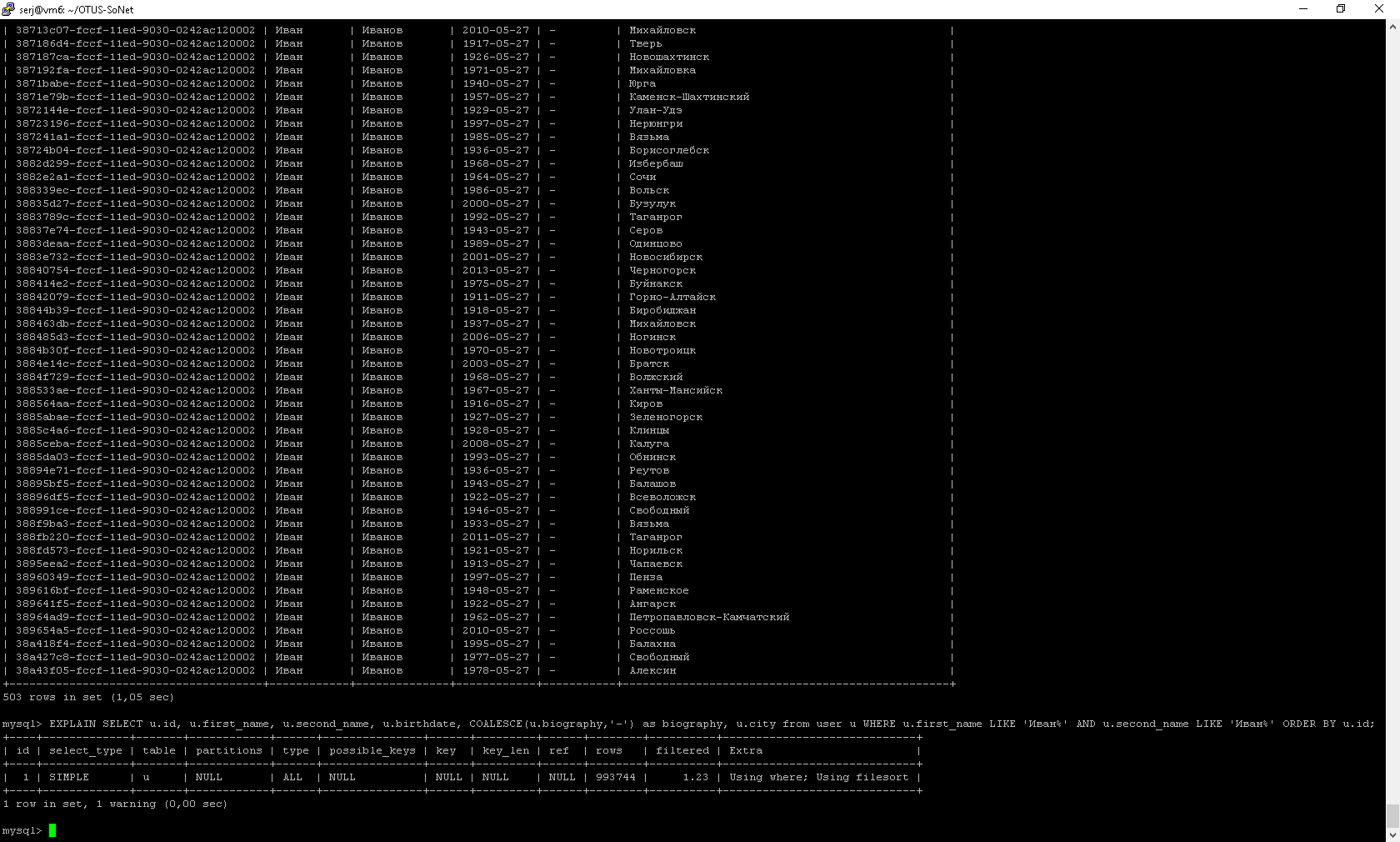
# Проверяем запрос

SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;



EXPLAIN SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;



# Создаем индексы:

Делаем отдельные индексы по полям **first\_name** и **second\_name** чтобы выполнялась интервальная выборка при поиске с префиксом LIKE prefix%

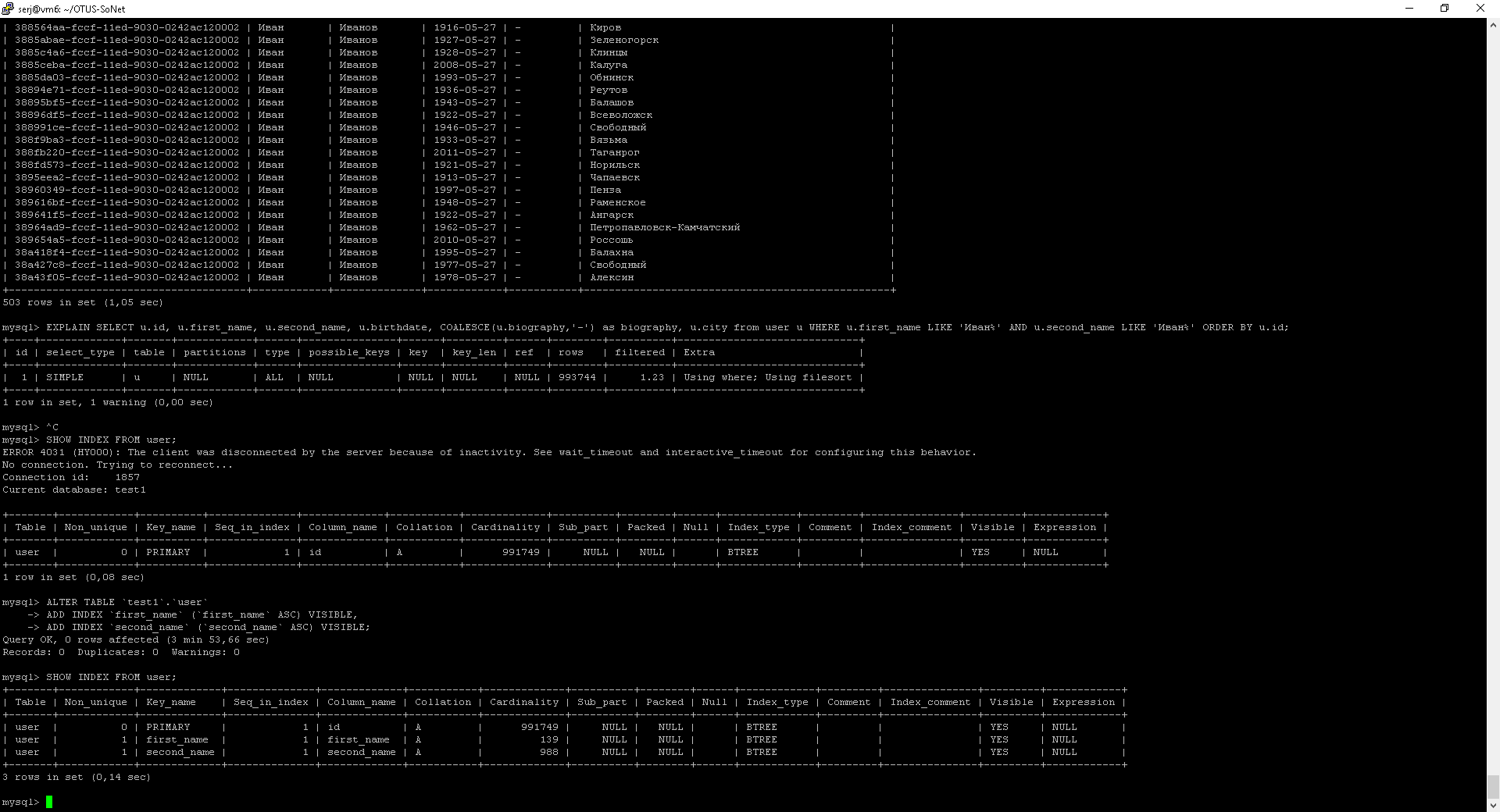
Primary key(id) - для сортировки

ALTER TABLE `test1`.`user`

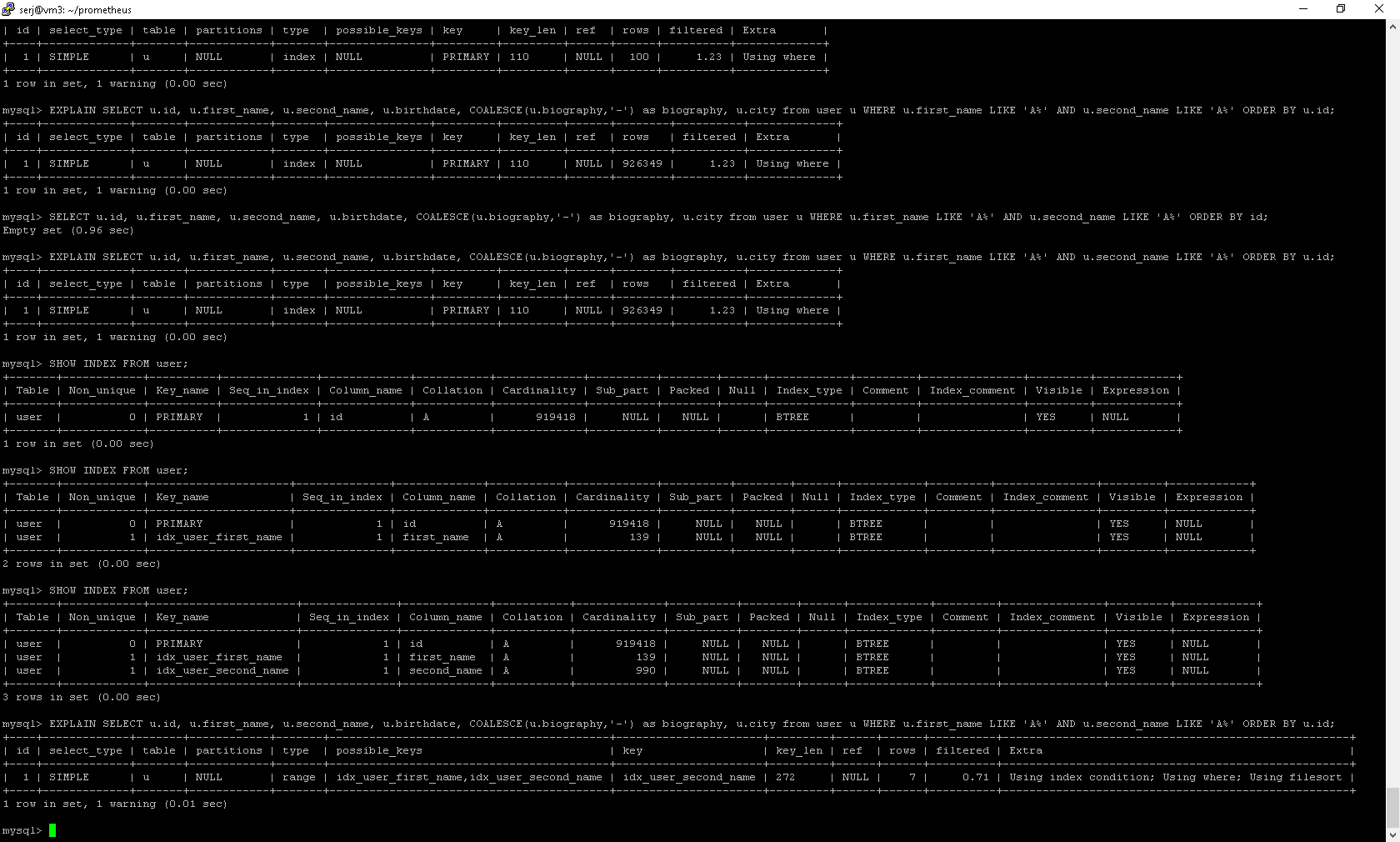
ADD PRIMARY KEY (`id`),

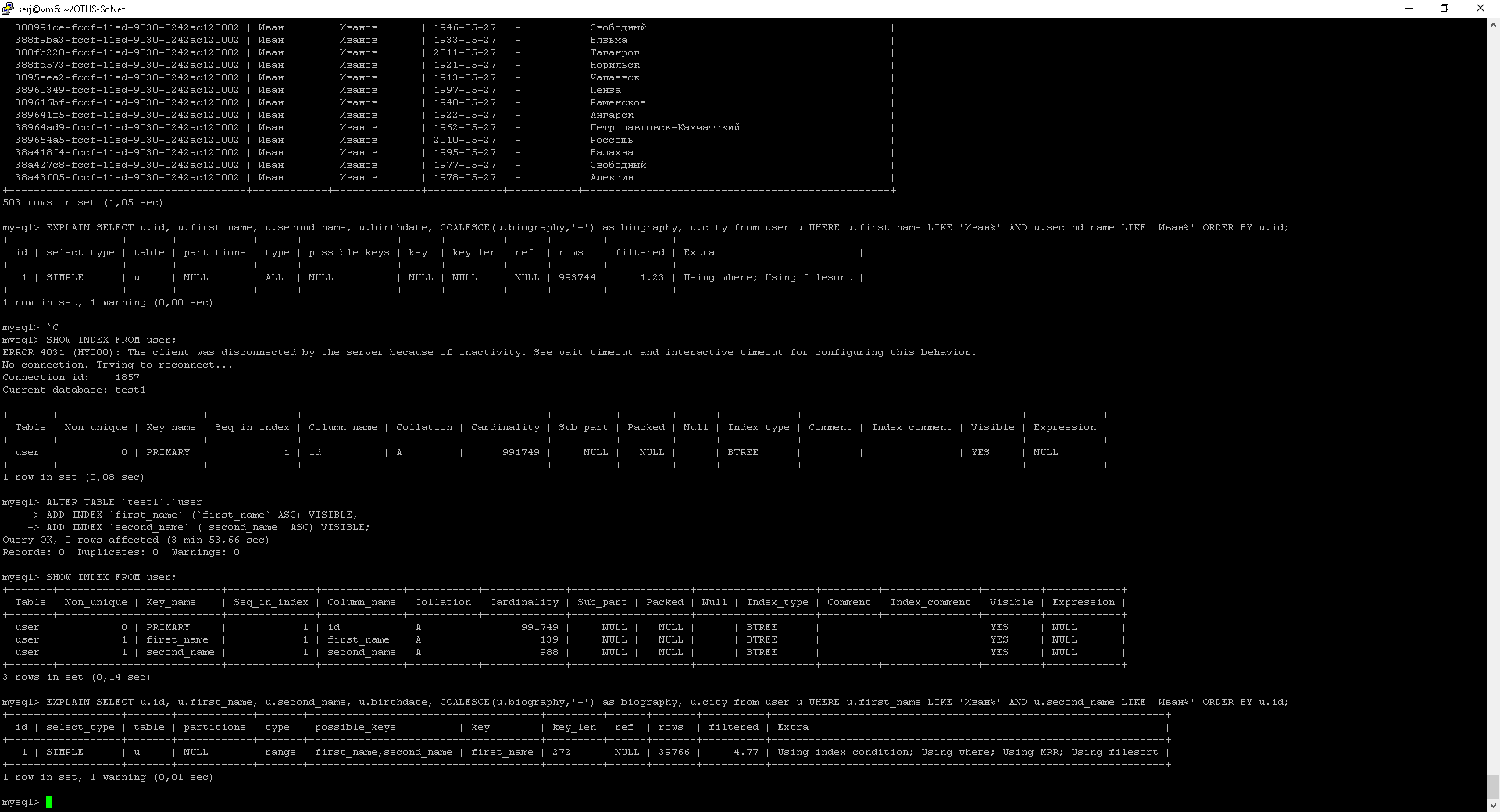
ADD INDEX `first\_name` (`first\_name` ASC) VISIBLE,

ADD INDEX `second\_name` (`second\_name` ASC) VISIBLE;



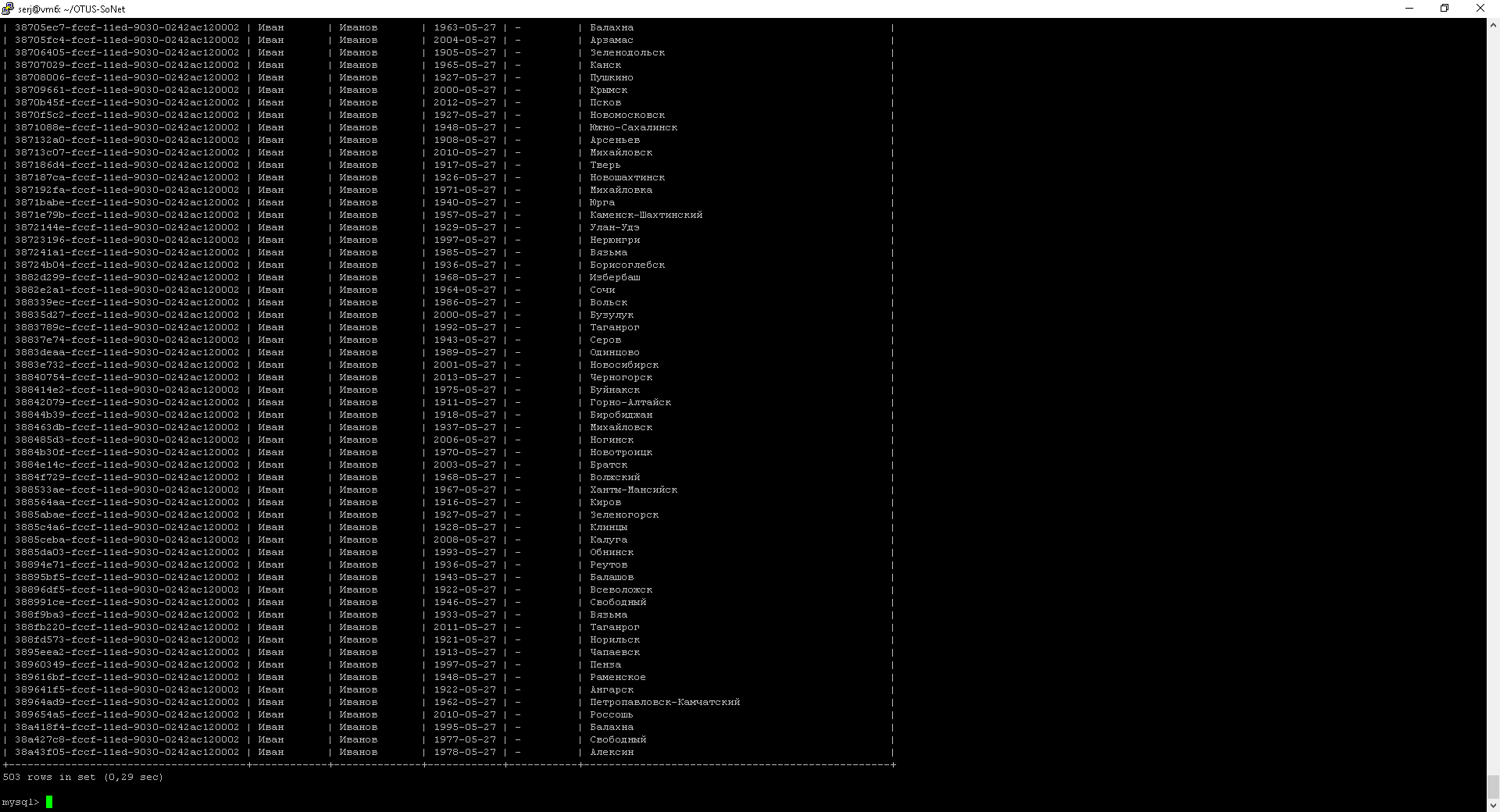
EXPLAIN SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;





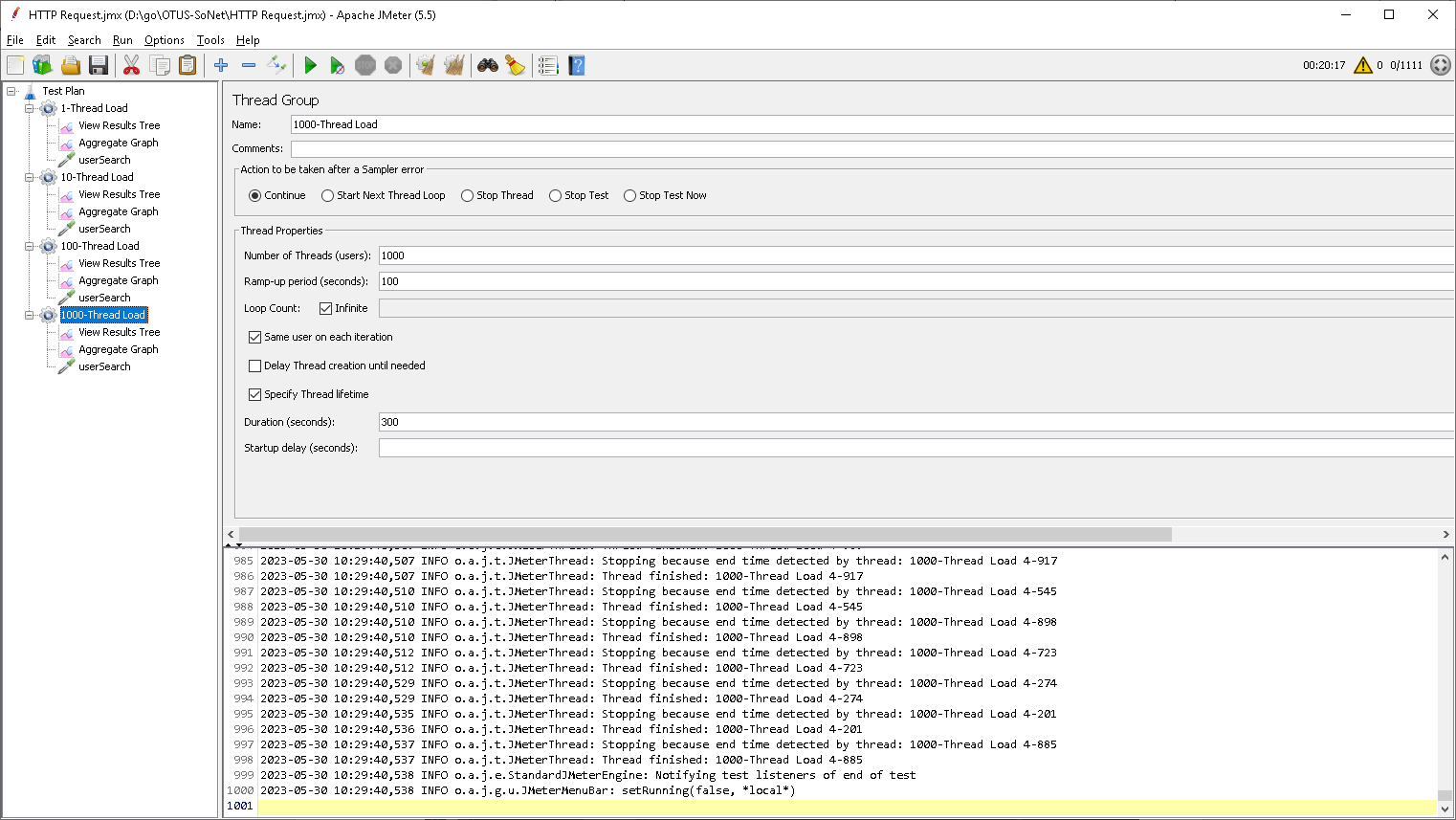
EXTRA: Using index condition; Using where; Using MRR; Using filesort

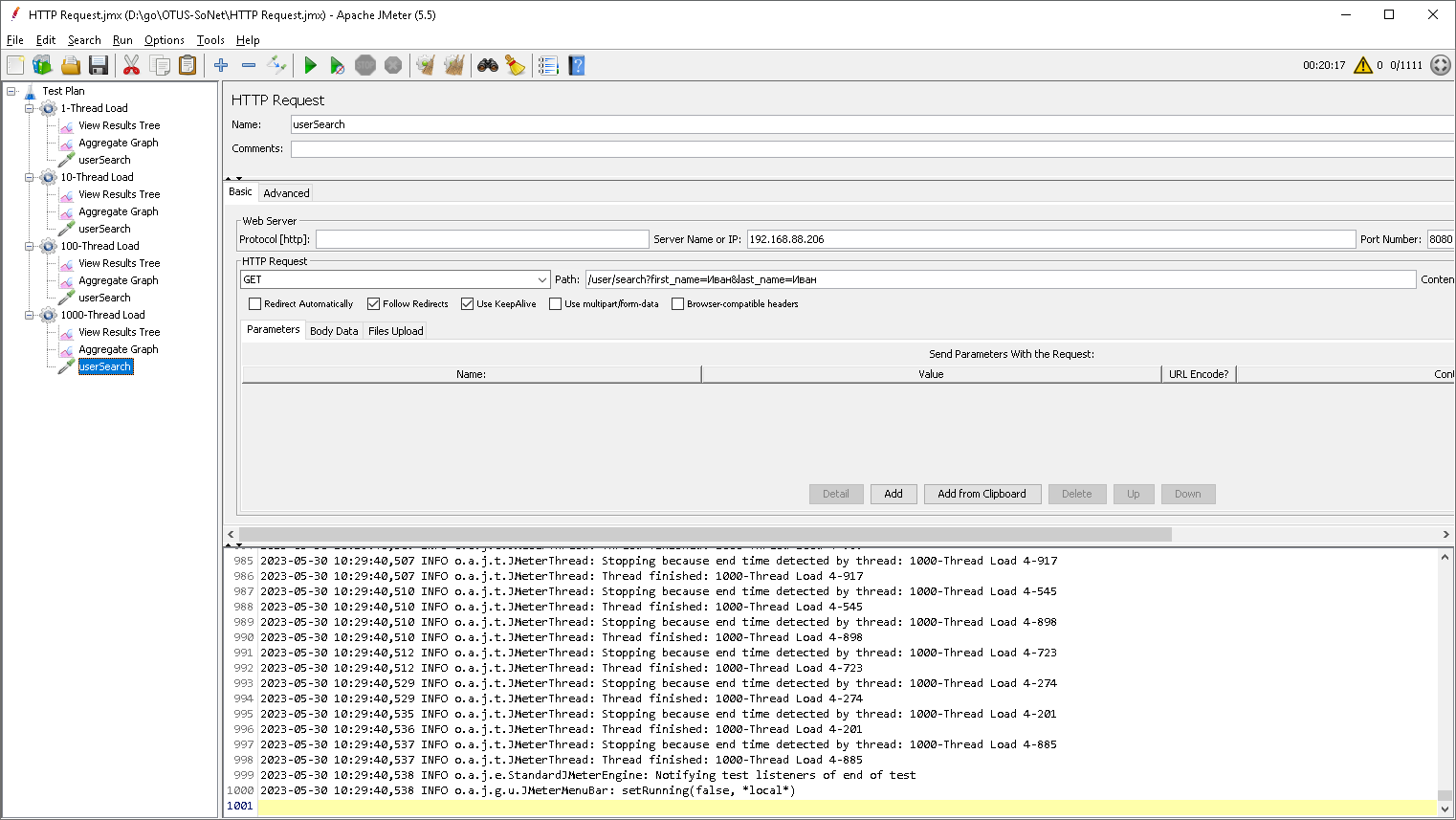
SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;



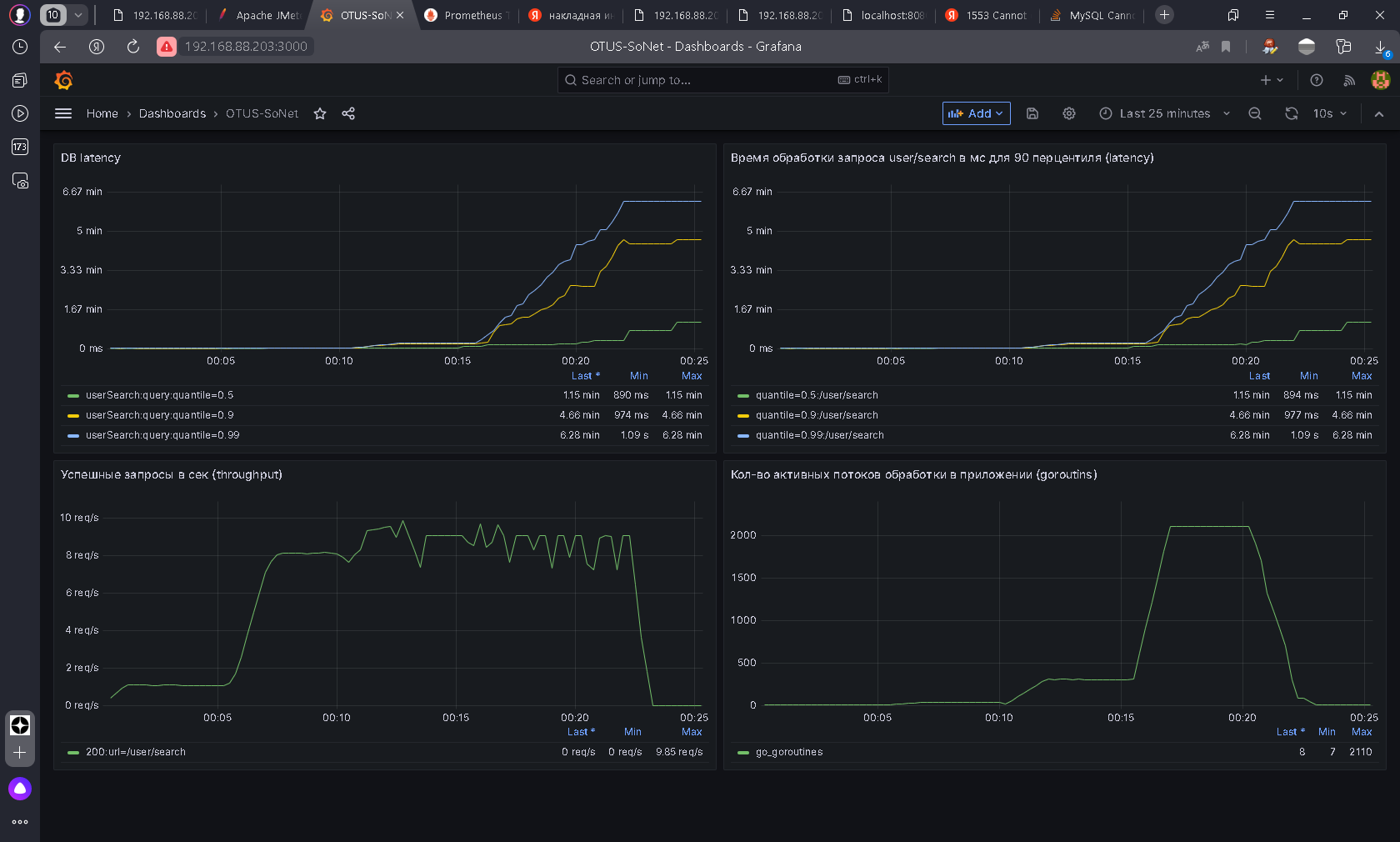
# Тестирование

4 испытания по длительностью по 5минут каждое, с нагрузкой в 1/10/100/1000 потоков





Общая картина для всех 4 испытаний без индексов:



1

10

100

1000

1

10

100

1000

1000

100

10

1

1000

100

10

1

Общая картина для всех 4 испытаний с индексами:



1

10

100

1000

1000

100

10

1

1

10

100

1000

1

10

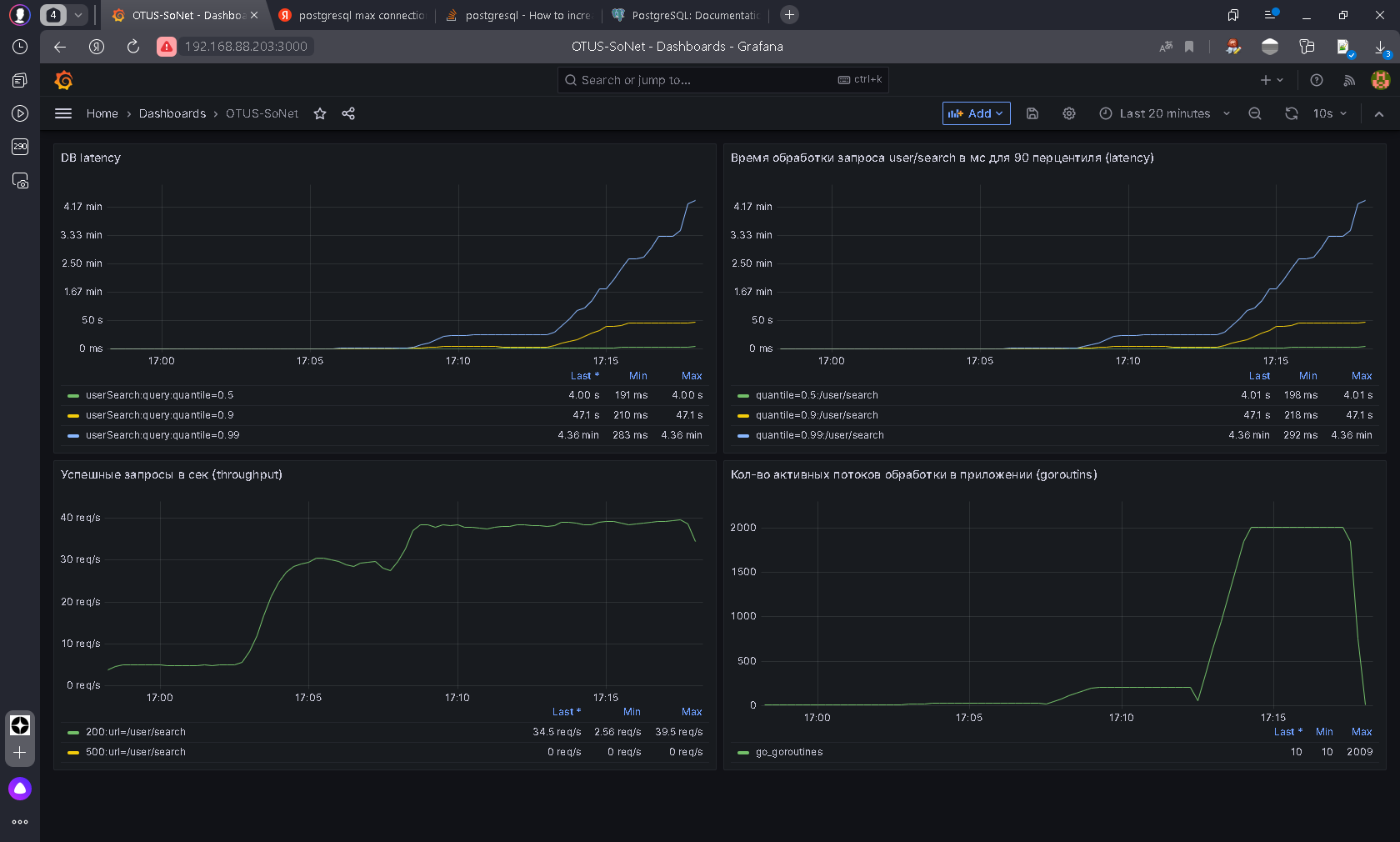
100

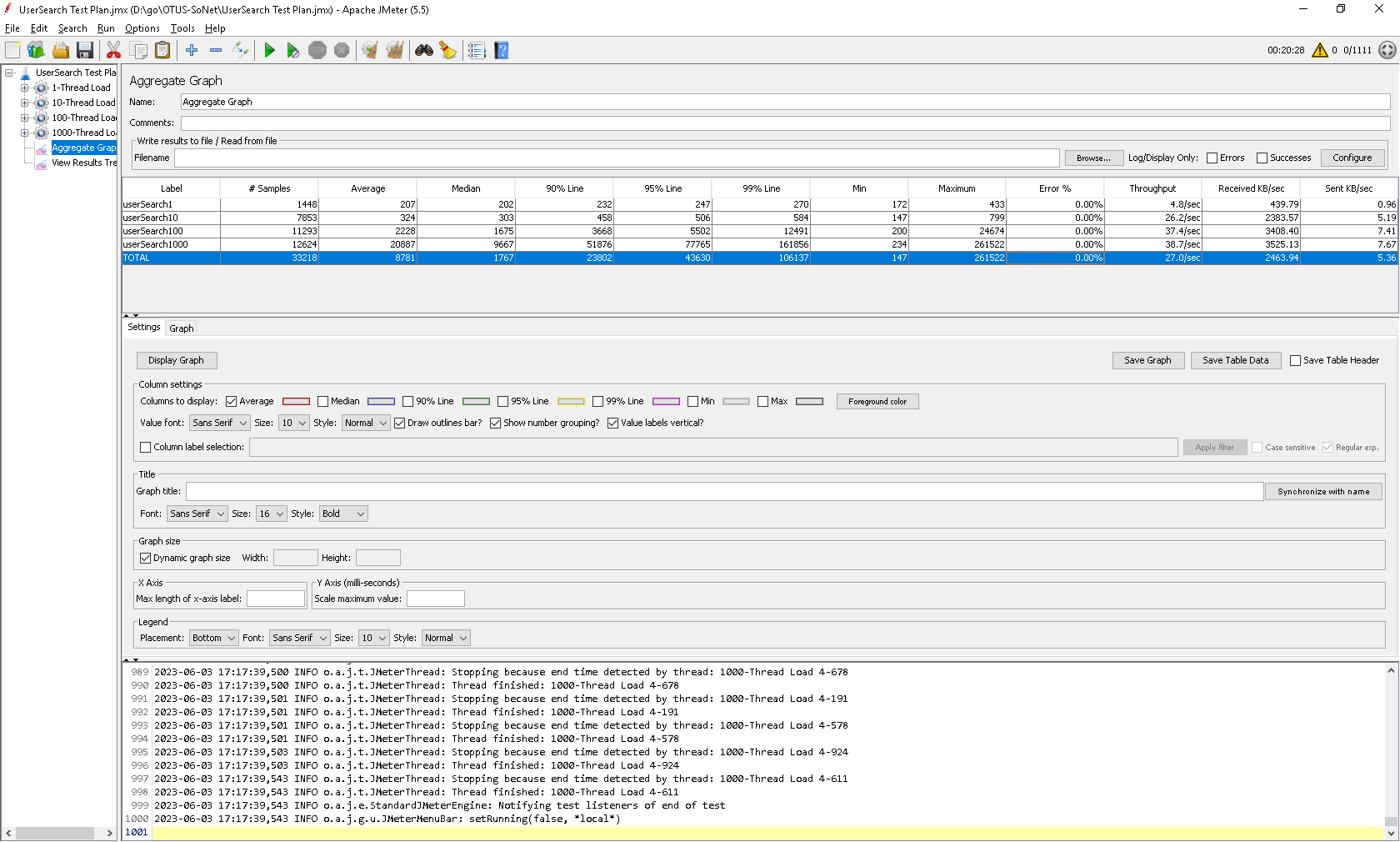
1000

# Сравнительная таблица:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Параметр** | **Пользователей** | **Без индекса** | | **С индексом** | |
| мин | макс | мин | макс |
| **Latency**  (90 перцентиль), ms | 1 | 974 | 1090 | 73 | 73 |
| 10 | 1290 | 1310 | 151 | 152 |
| 100 | 10000 | 12600 | 1430 | 1850 |
| 1000 | 61800 | 162000 | 5060 | 27200 |
| **Throughput**, rps | 1 | 1 | 1 | 14 | 14 |
| 10 | 8 | 8 | 65 | 66 |
| 100 | 7 | 9 | 55 | 58 |
| 1000 | 7 | 9 | 57 | 59 |

PostgreSQL:





EXPLAIN SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from public.user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;

QUERY PLAN

----------------------------------------------------------------------------------------------------

Sort (cost=45559.10..45559.11 rows=1 width=136)

Sort Key: id

-> Gather (cost=1000.00..45559.09 rows=1 width=136)

Workers Planned: 2

-> Parallel Seq Scan on "user" u (cost=0.00..44558.99 rows=1 width=136)

Filter: (((first\_name)::text ~~ 'A%'::text) AND ((second\_name)::text ~~ 'A%'::text))

(6 rows)

ALTER TABLE public.user ADD PRIMARY KEY (id);

CREATE INDEX idx\_first\_name ON public.user (first\_name);

EXPLAIN SELECT u.id, u.first\_name, u.second\_name, u.birthdate, COALESCE(u.biography,'-') as biography, u.city from public.user u WHERE u.first\_name LIKE 'A%' AND u.second\_name LIKE 'A%' ORDER BY u.id;

QUERY PLAN

----------------------------------------------------------------------------------------------------

Sort (cost=45595.24..45595.24 rows=1 width=136)

Sort Key: id

-> Gather (cost=1000.00..45595.22 rows=1 width=136)

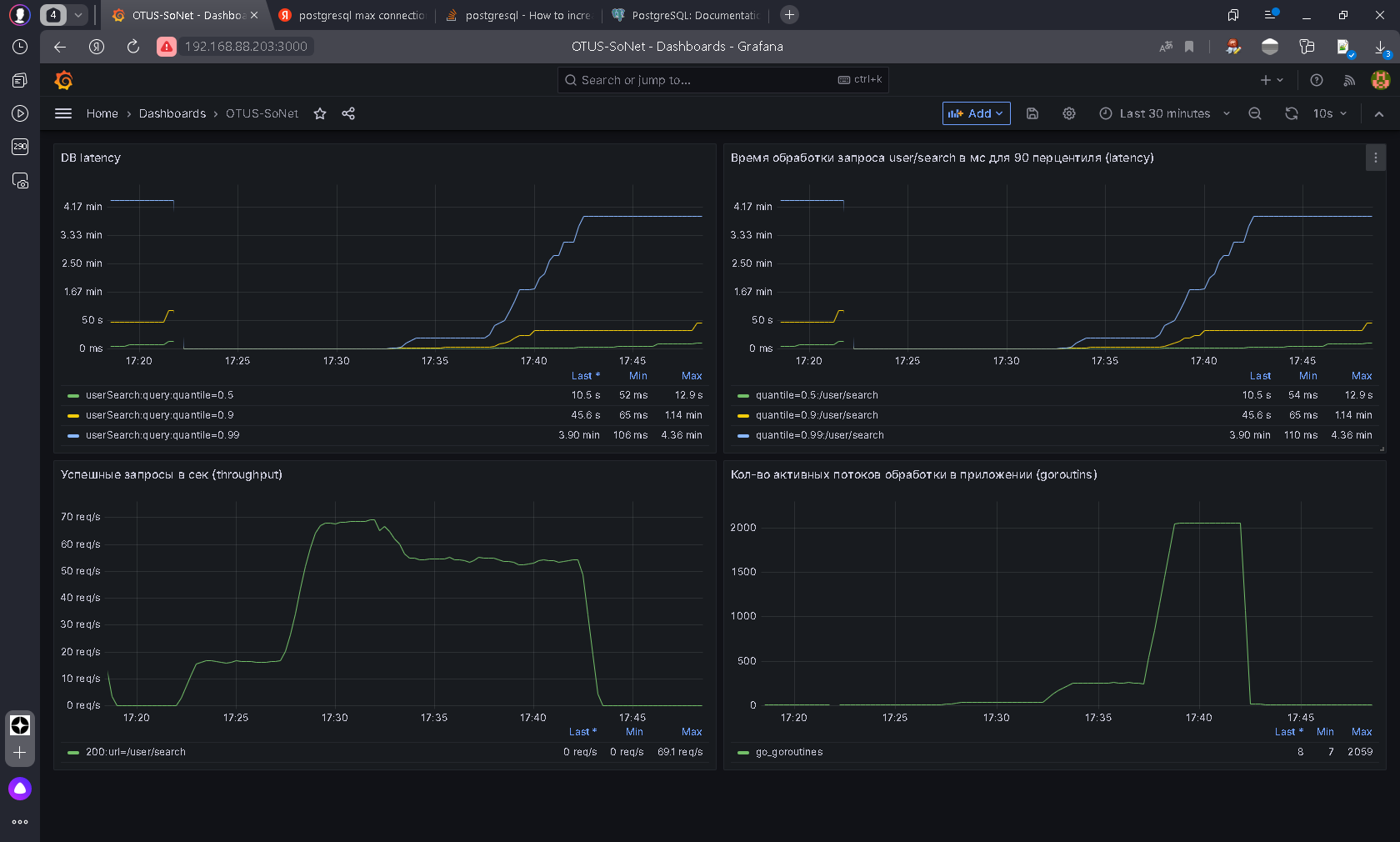
Workers Planned: 2

-> Parallel Seq Scan on "user" u (cost=0.00..44595.12 rows=1 width=136)

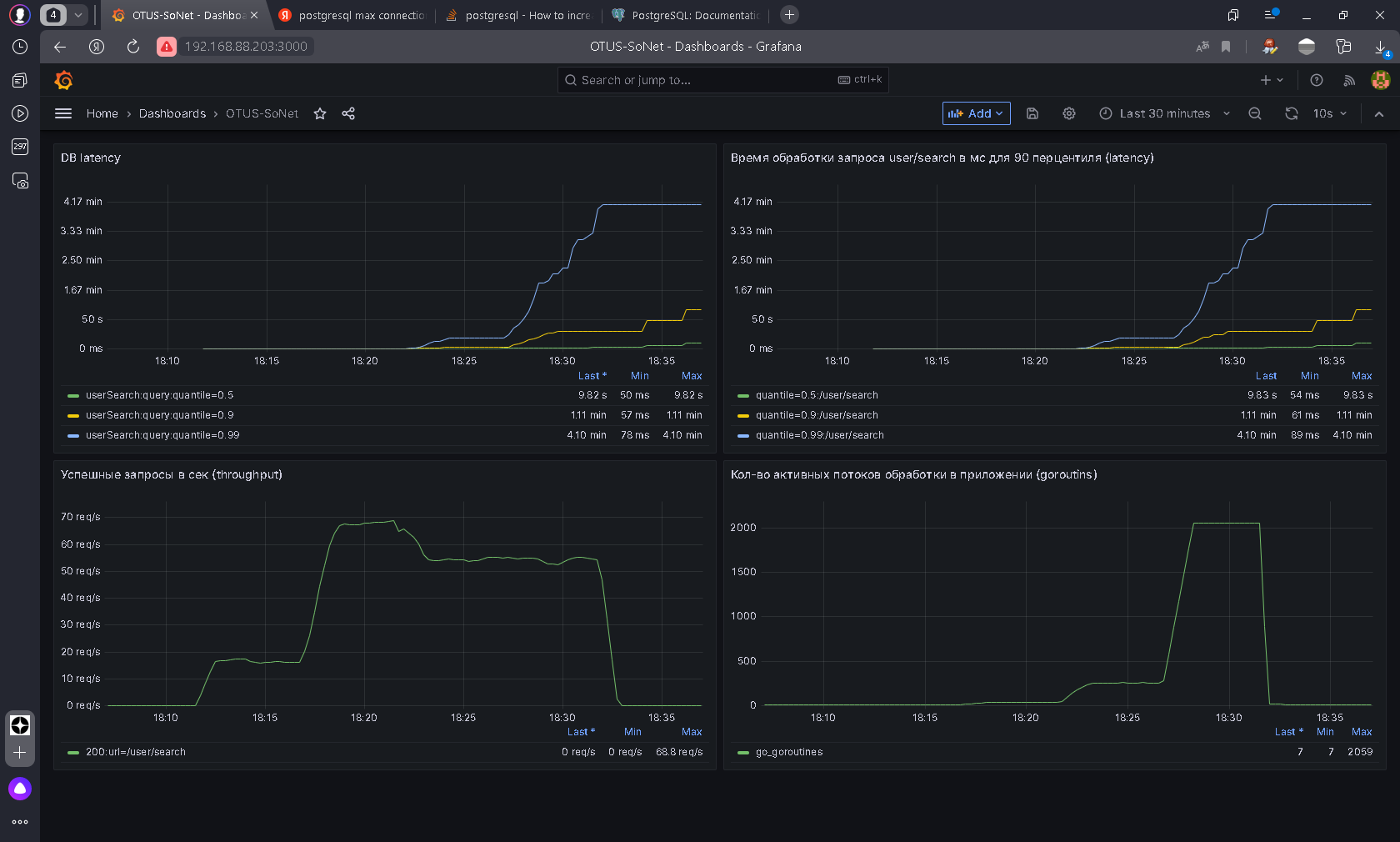
Filter: (((first\_name)::text ~~ 'A%'::text) AND ((second\_name)::text ~~ 'A%'::text))

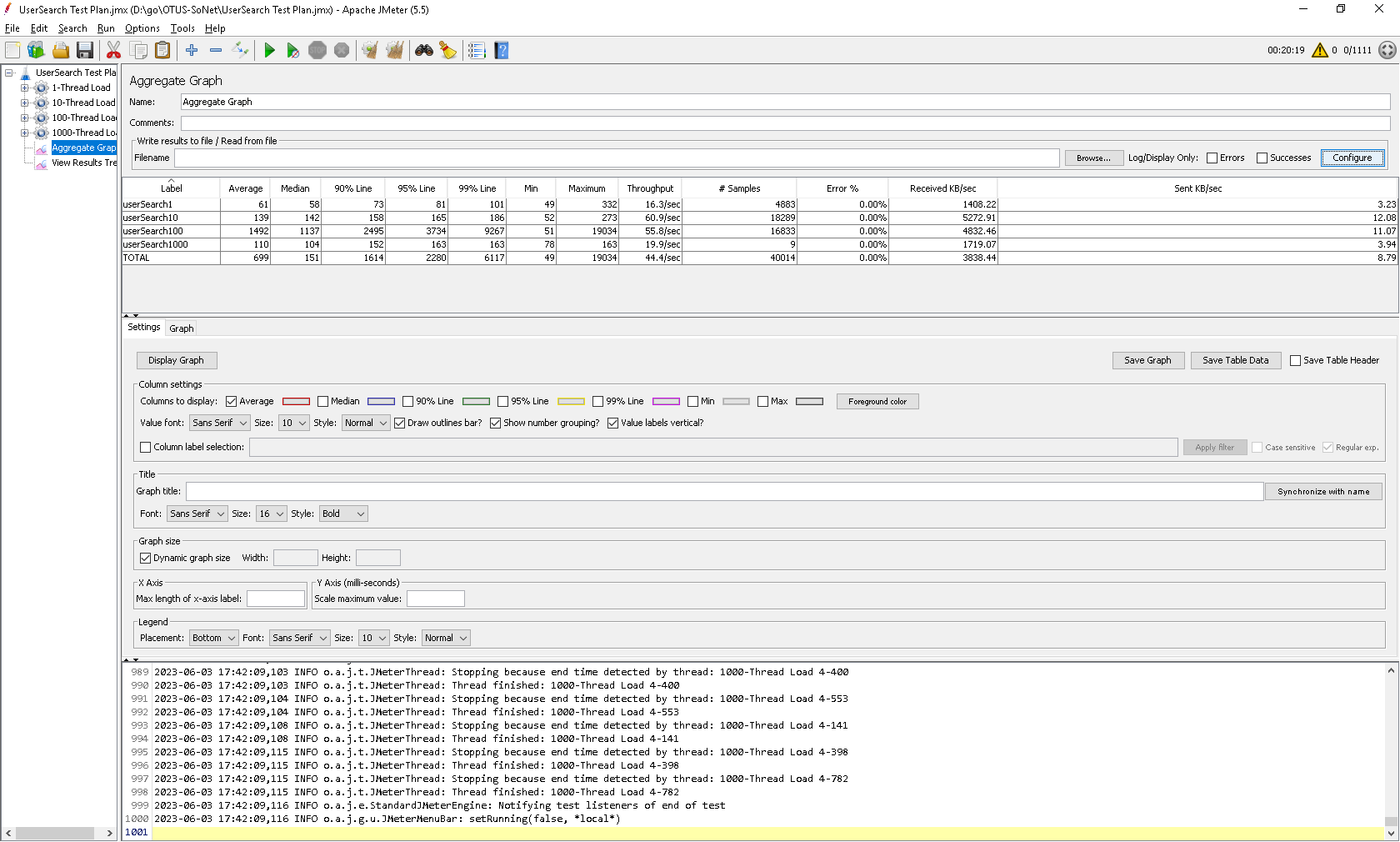
(6 rows)

MySQL 50connects



PostgreSQL 50 connects:



MySQL:

PostgreSQL:

