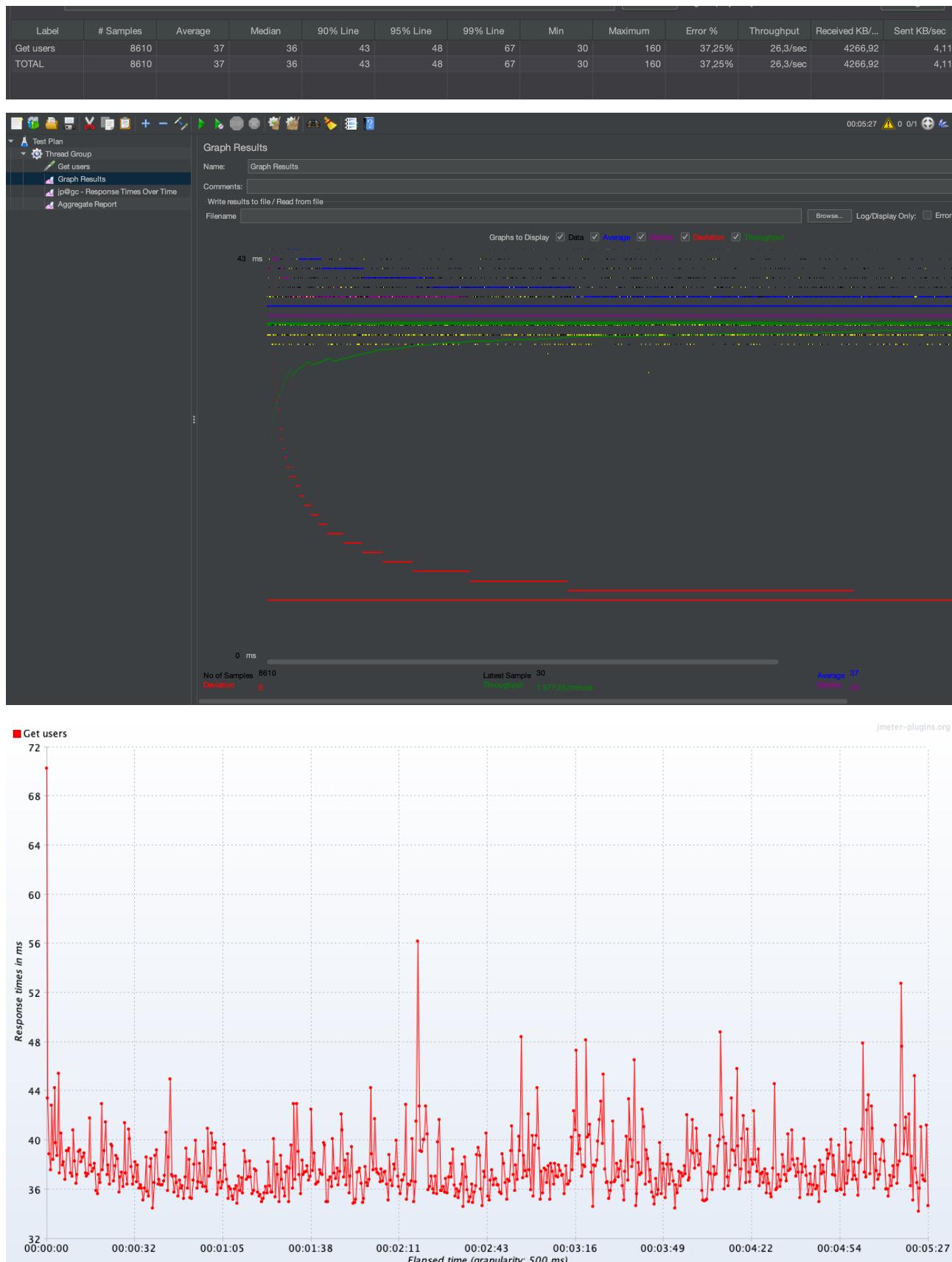


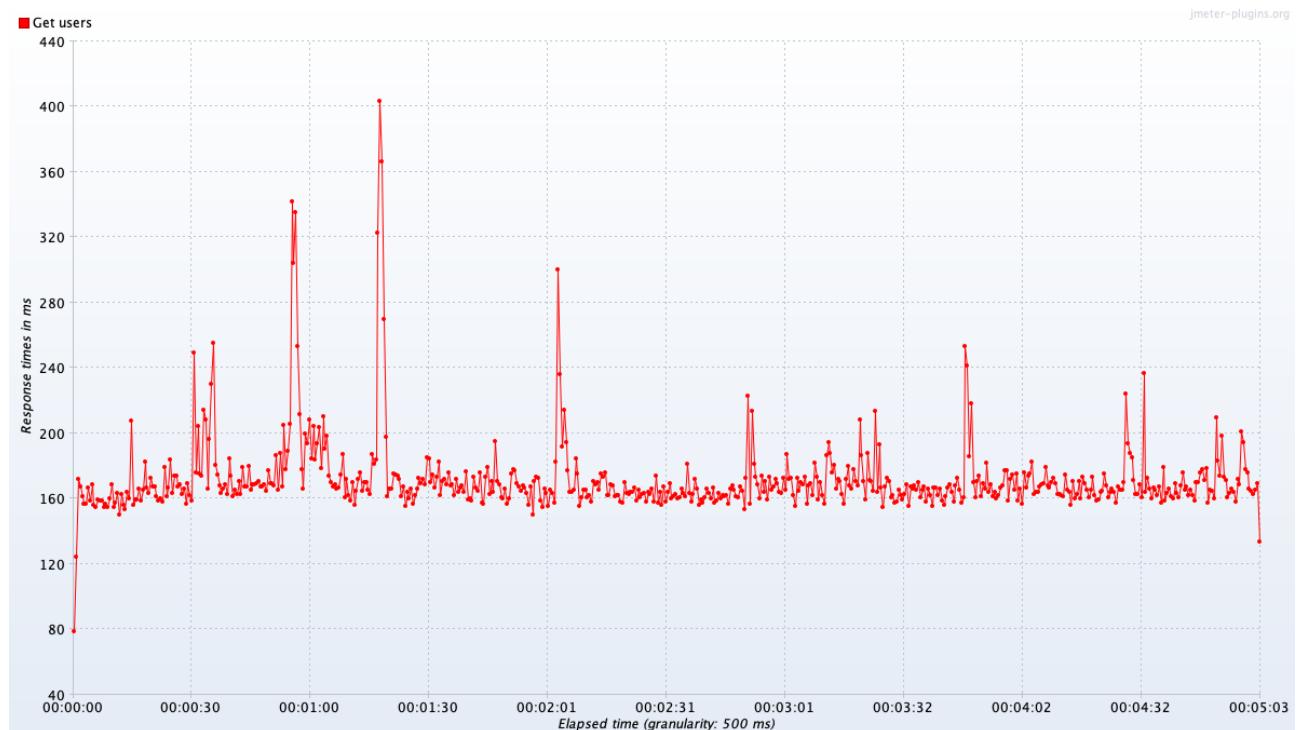
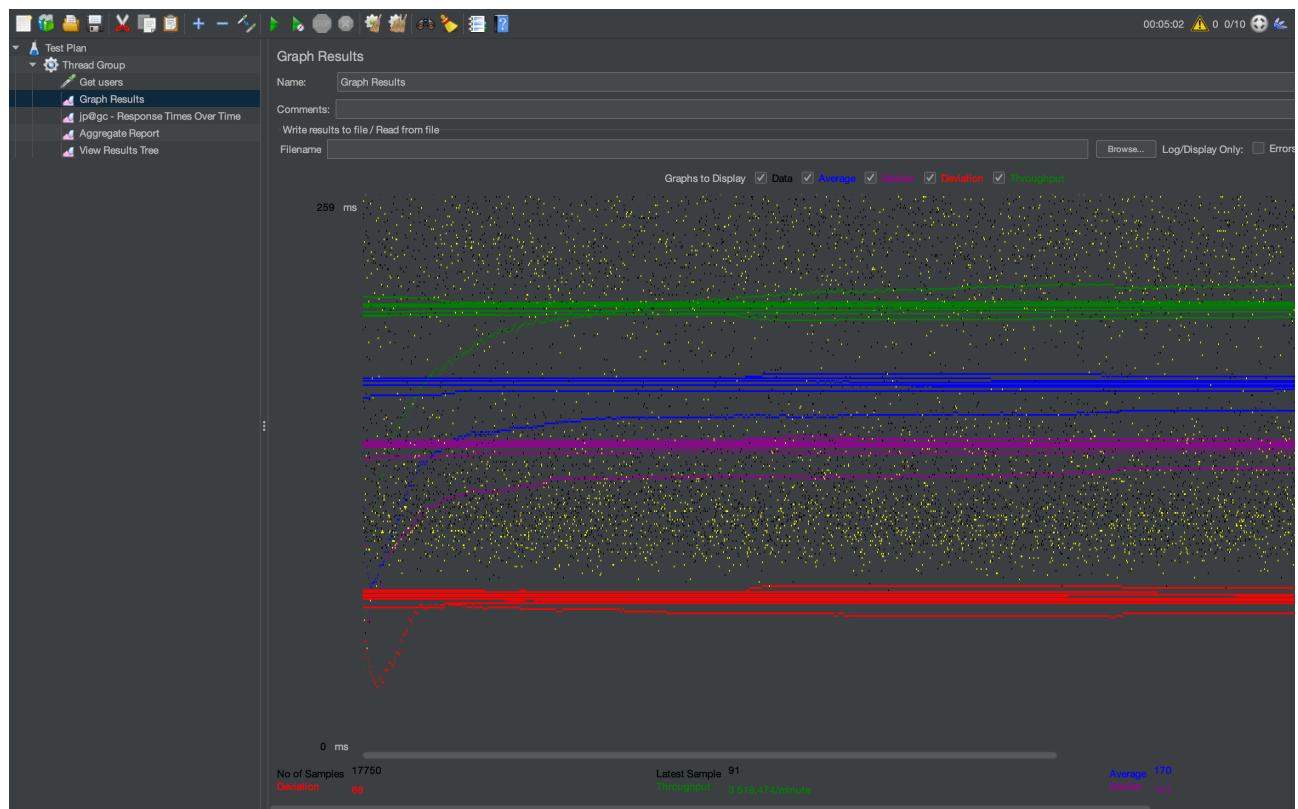
Тесты до применения индекса:

1 Запрос:



## 10 Запросов:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/...	Sent KB/sec
Get users	17750	170	141	264	284	334	1	713	37,81%	58,7/sec	9131,39	9,16
TOTAL	17750	170	141	264	284	334	1	713	37,81%	58,7/sec	9131,39	9,16



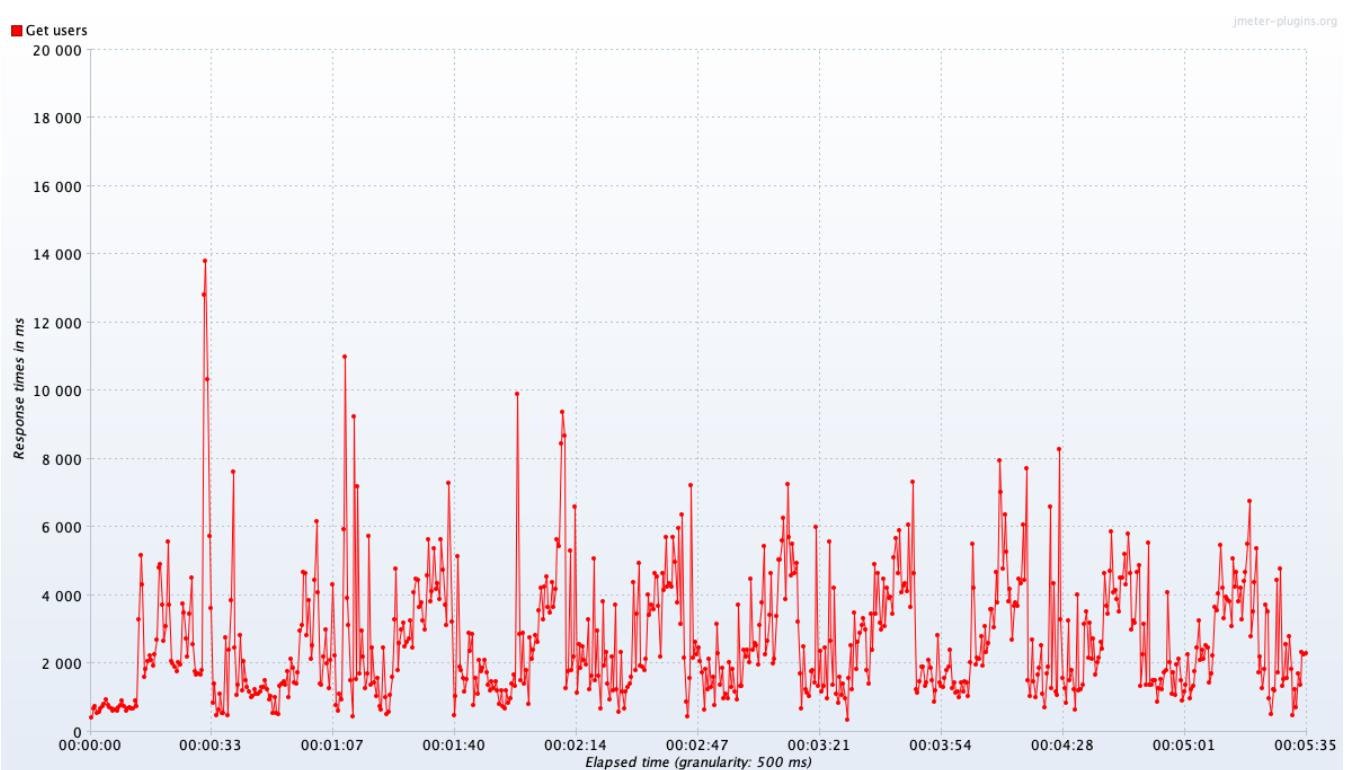
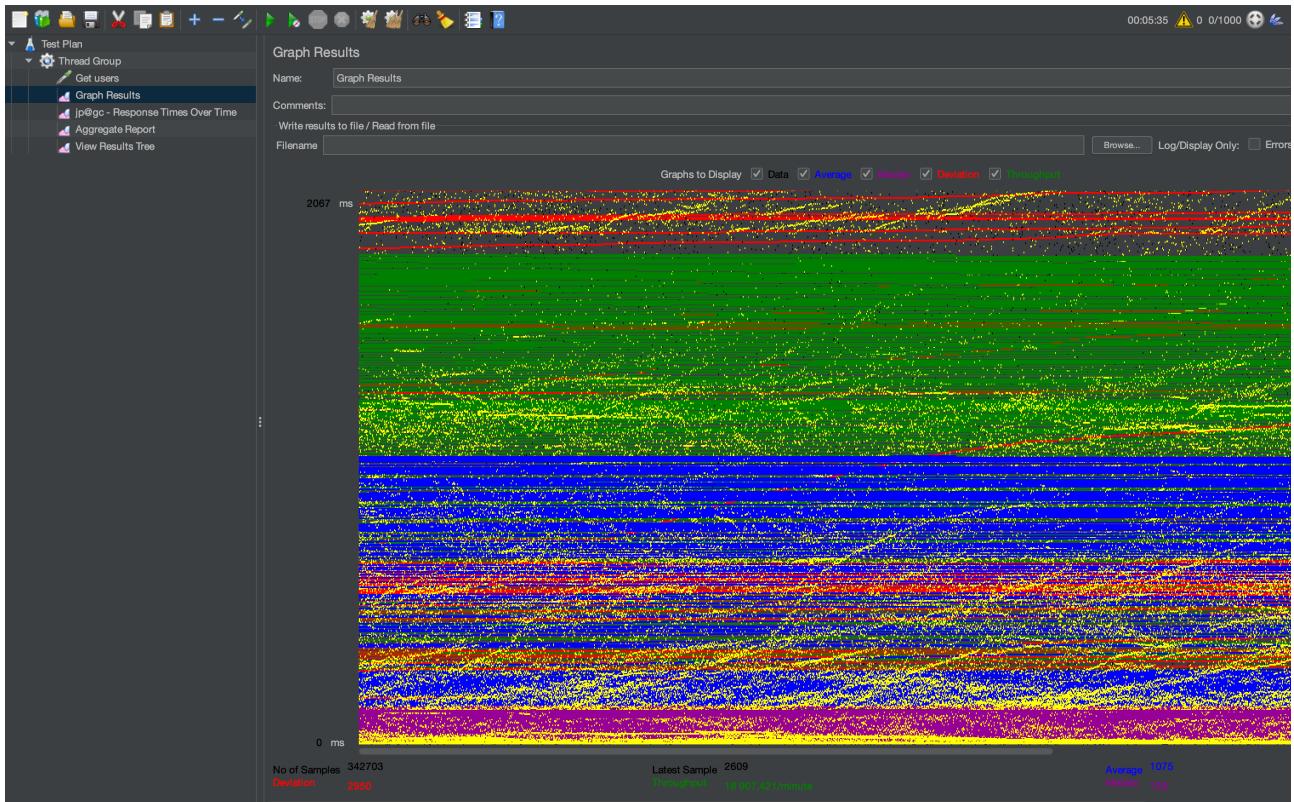
## 100 Запросов:

Thread Group												
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/sec	Sent KB/sec
Get users	134433	233	11	92	2381	3012	1	15848	94,30%	427,1/sec	6279,13	66,68
TOTAL	134433	233	11	92	2381	3012	1	15848	94,30%	427,1/sec	6279,13	66,68



## 1000 Запросов:

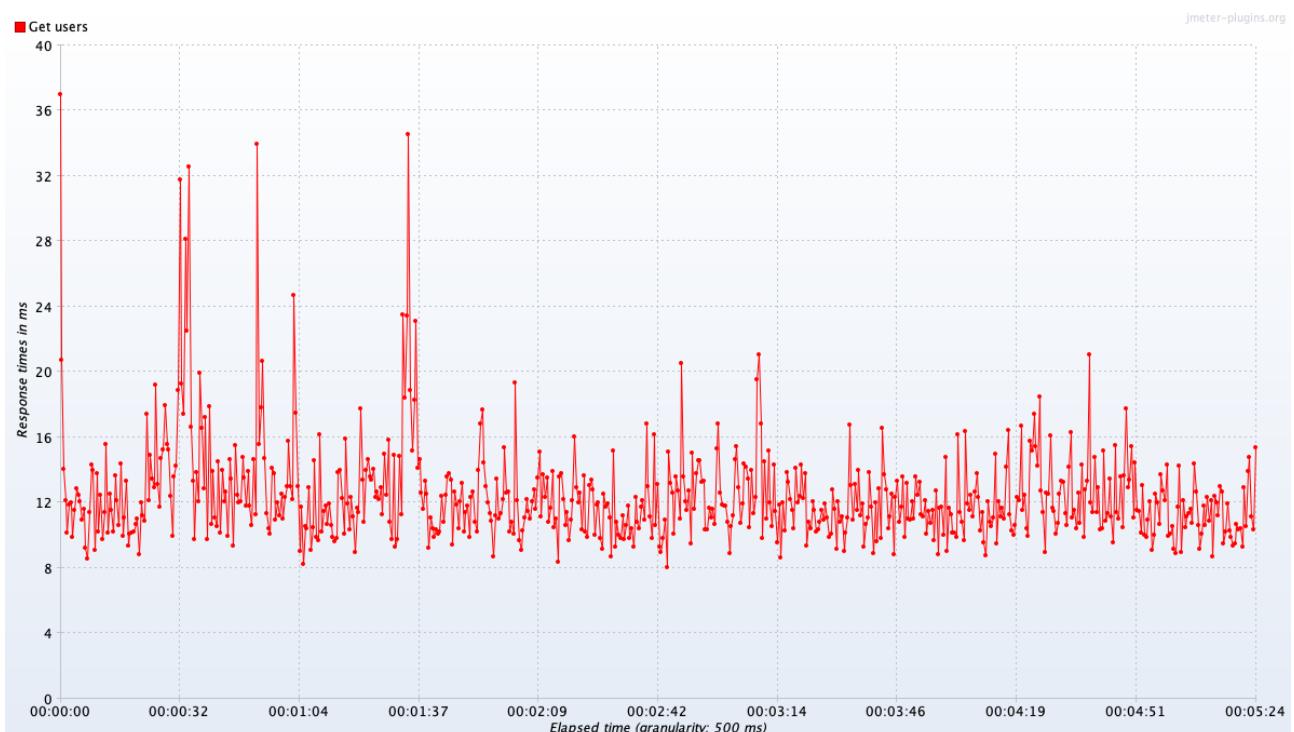
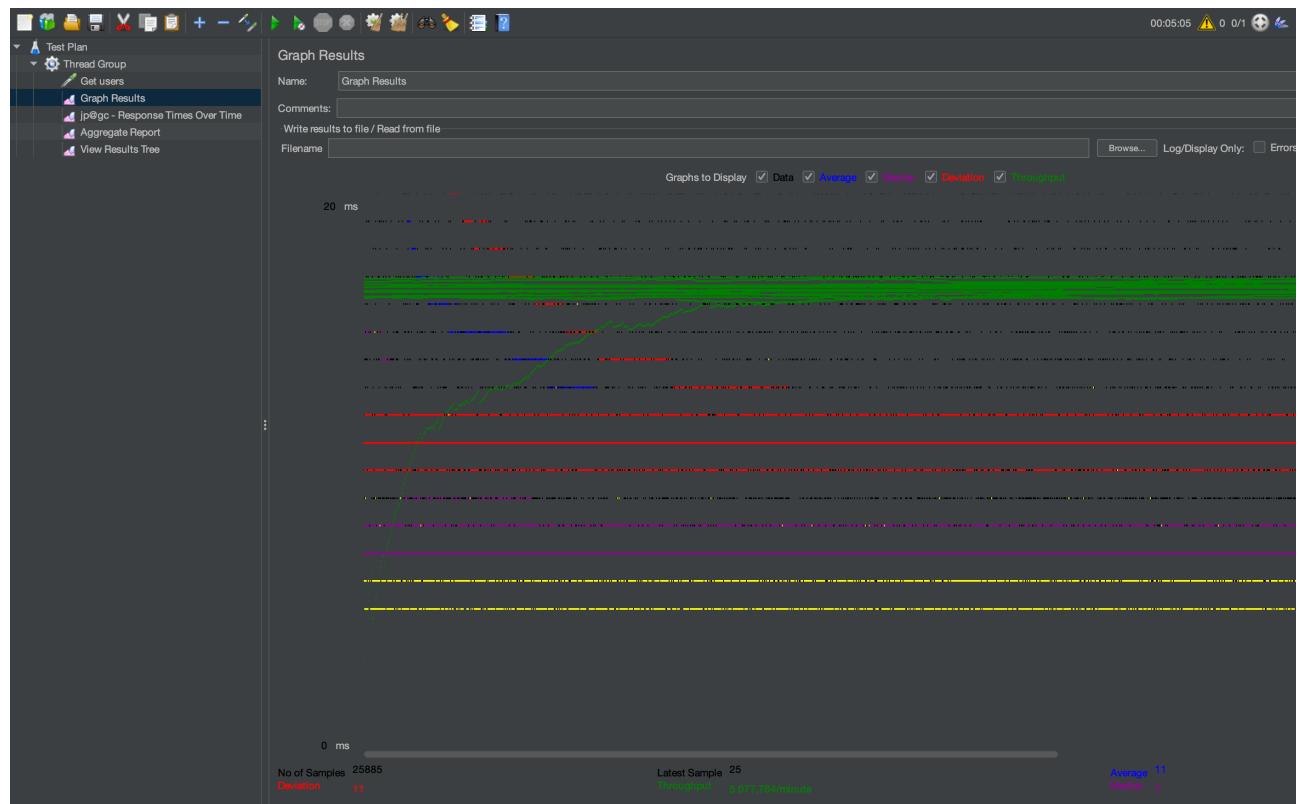
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/...	Sent KB/sec
Get users	324953	1124	122	2418	5276	19537	0	60721	95.48%	455.7/sec	5487.15	63.99
TOTAL	324953	1124	122	2418	5276	19537	0	60721	95.48%	455.7/sec	5487.15	63.99



## Тесты после применения индекса:

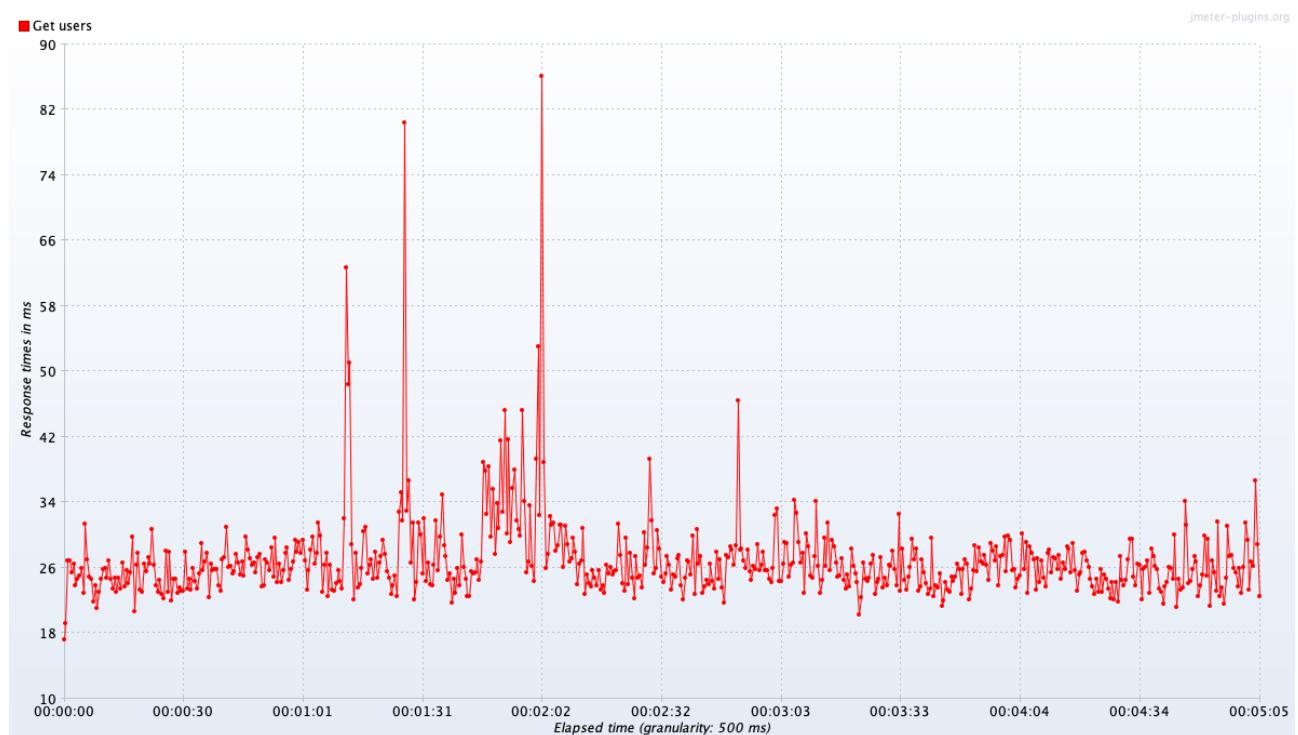
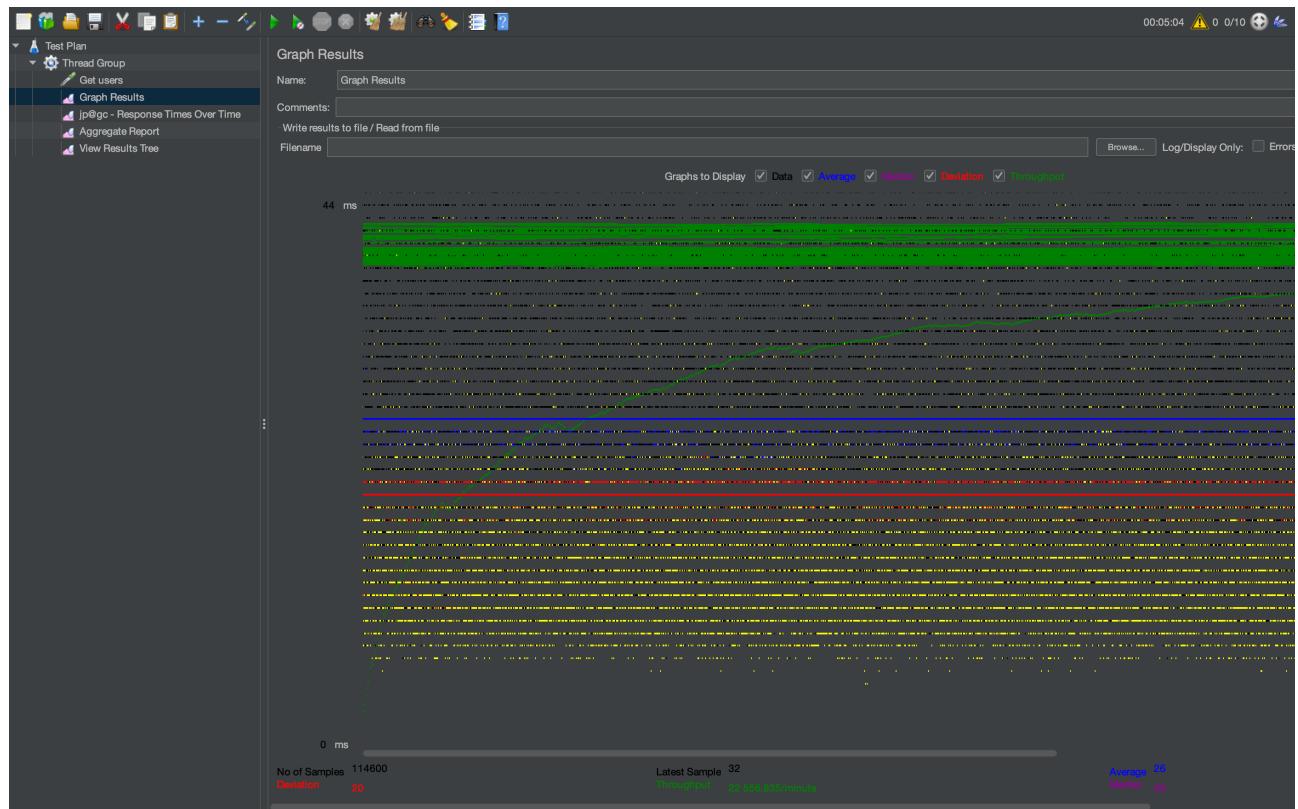
1 Запрос:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/sec	Sent KB/sec
Get users	26202	11	8	22	33	63	5	247	37,85%	80,9/sec	12720,24	12,64
TOTAL	26202	11	8	22	33	63	5	247	37,85%	80,9/sec	12720,24	12,64



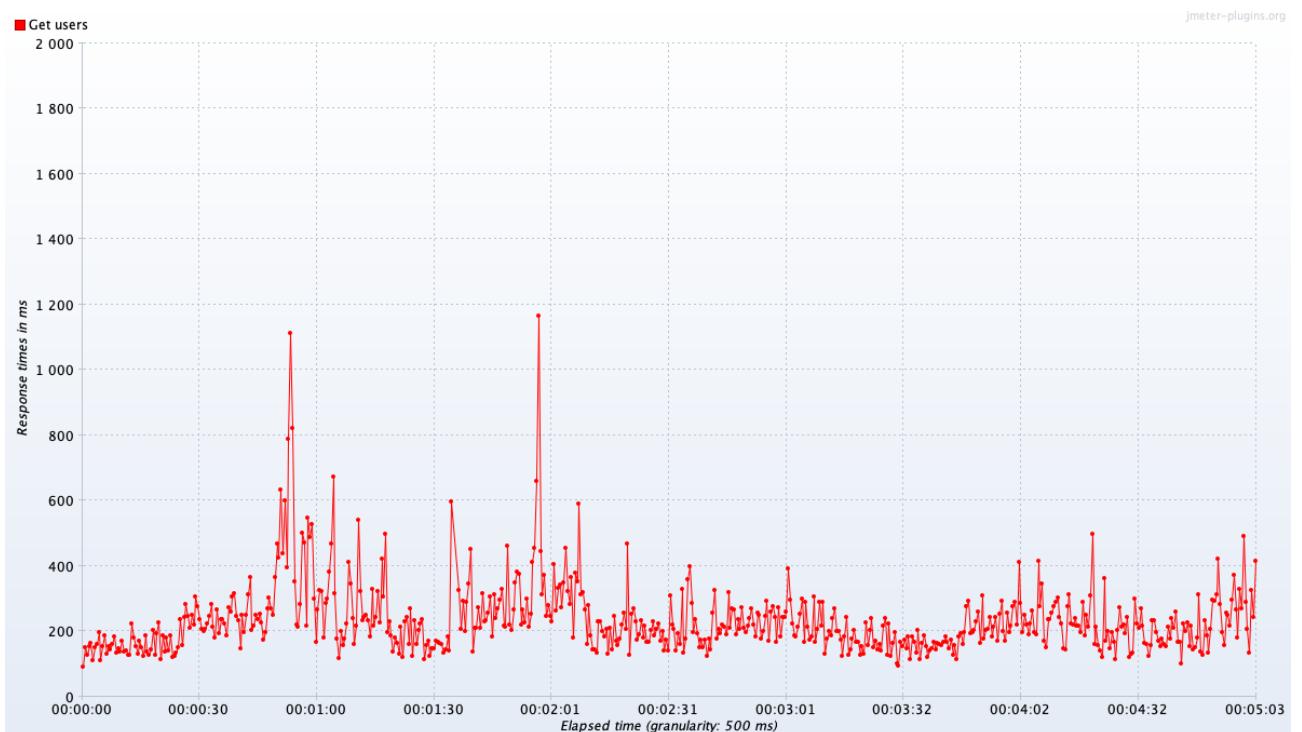
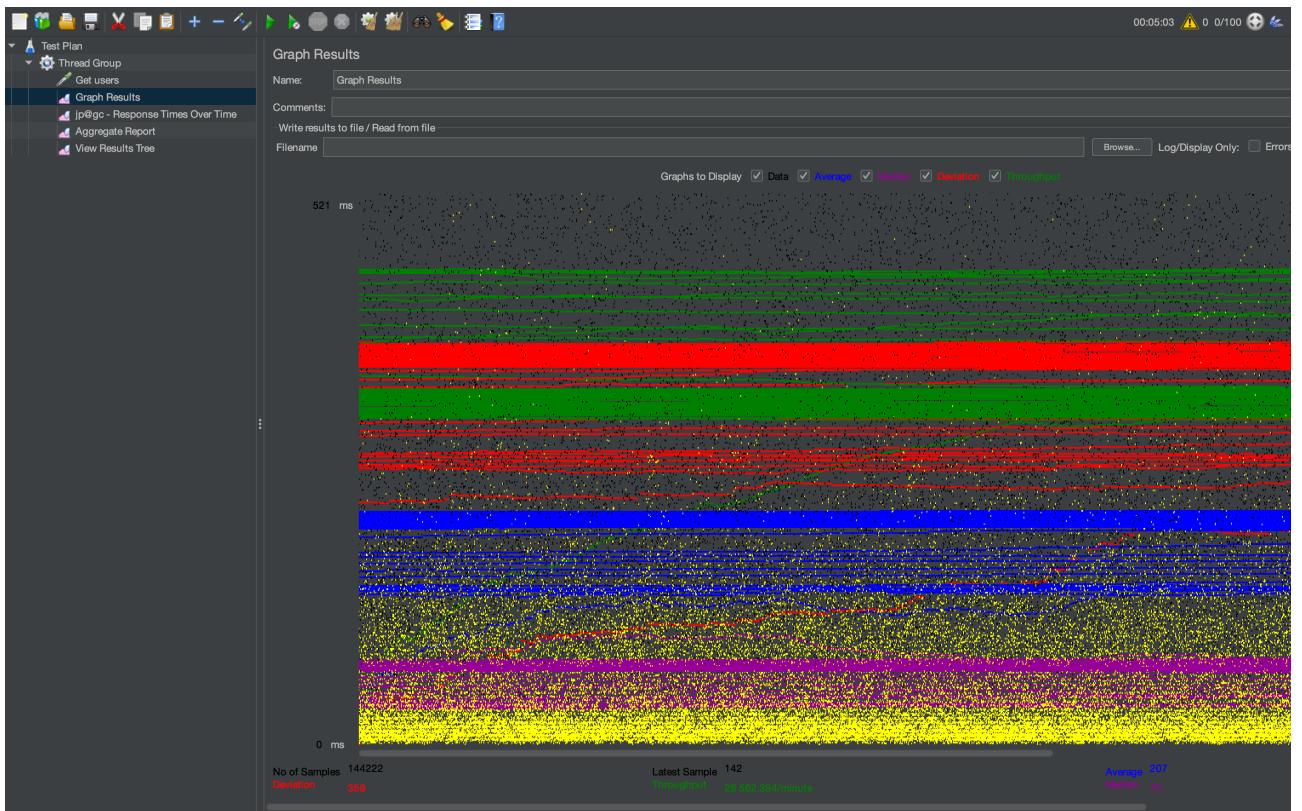
## 10 Запросов:

Write results to file / Read from file												
Filename	<input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>											
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/...	Sent KB/sec
Get users	114600	26	20	46	64	113	5	425	37,91%	375,9/sec	59052,05	58,74
TOTAL	114600	26	20	46	64	113	5	425	37,91%	375,9/sec	59052,05	58,74



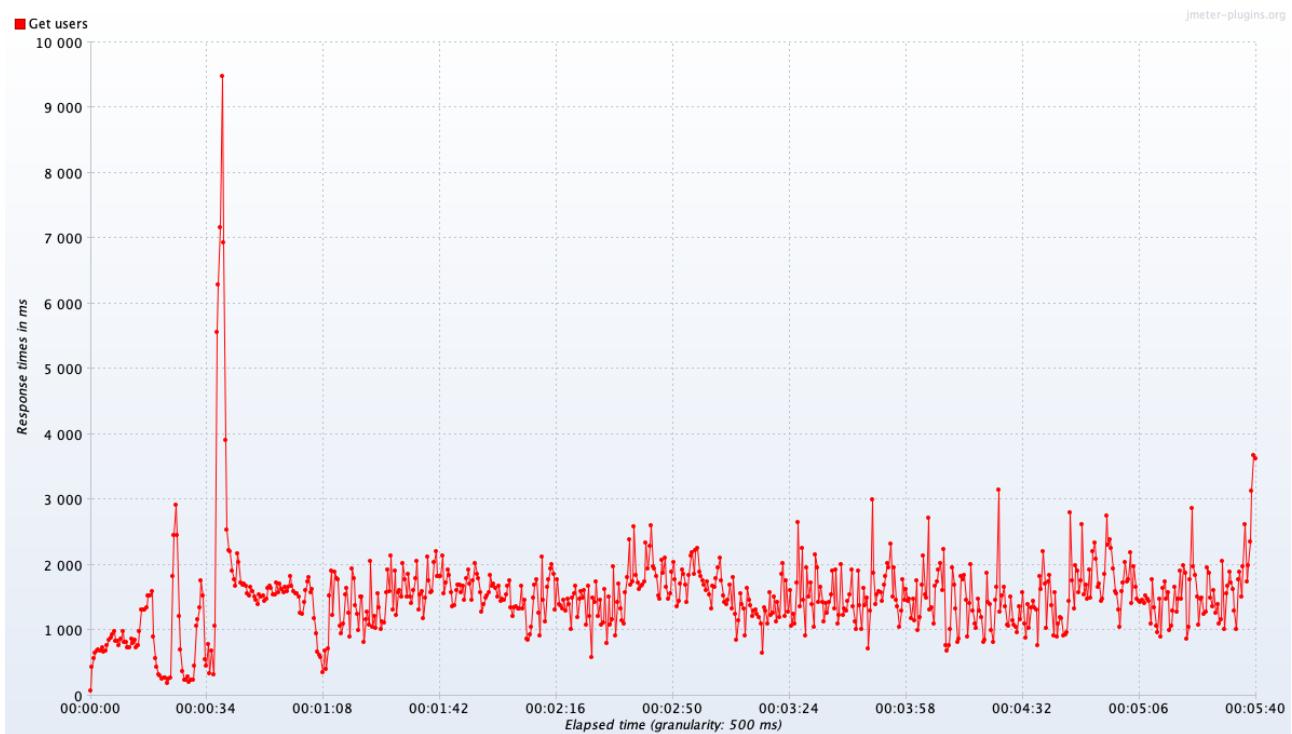
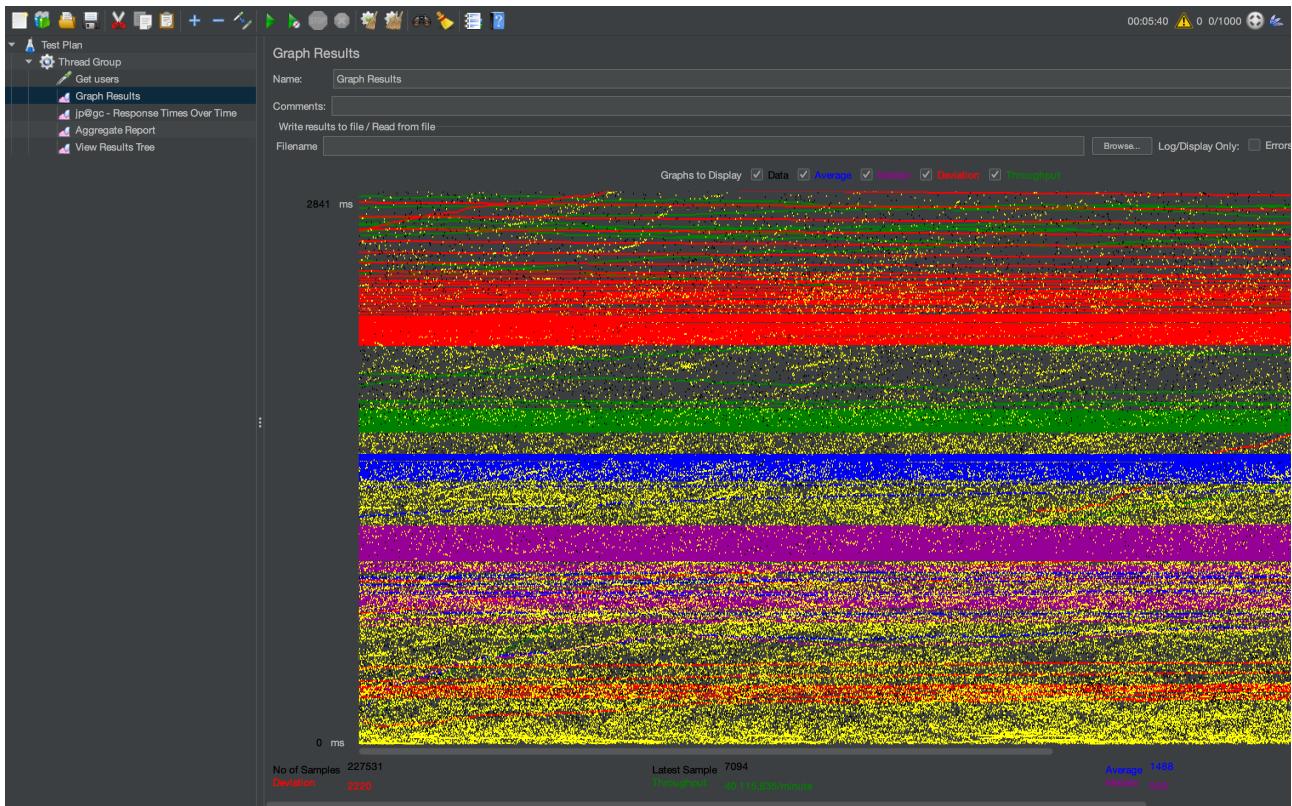
## 100 Запросов:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/...	Sent KB/sec
Get users	144222	207	75	567	880	1834	2	5539	65,25%	476,0/sec	41570,69	74,33
TOTAL	144222	207	75	567	880	1834	2	5539	65,25%	476,0/sec	41570,69	74,33



1000 Запросов:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error % ↑	Throughput	Received KB/...	Sent KB/sec
Get users	227531	1488	949	3013	4657	13308	0	36841	83,20%	668,6/sec	28514,71	98,92
TOTAL	227531	1488	949	3013	4657	13308	0	36841	83,20%	668,6/sec	28514,71	98,92



Запрос добавления индекса:

```
CREATE INDEX idx_users_name_surname
ON otus.users USING btree
(name text_pattern_ops, surname text_pattern_ops);
```

Explain запросов:

```

[postgres=# EXPLAIN SELECT * FROM otus.users WHERE name like 'A%' and surname like 'A%' order by id;
                                         QUERY PLAN
-----
Sort  (cost=8.47..8.47 rows=1 width=205)
  Sort Key: id
  -> Index Scan using idx_users_name_surname on users  (cost=0.42..8.46 rows=1 width=205)
      Index Cond: ((name ~>=~ 'A'::text) AND (name ~<~ 'B'::text) AND (surname ~>=~ 'A'::text) AND (surname ~<~ 'B'::text))
          Filter: ((name ~~ 'A%')::text) AND (surname ~~ 'A%')::text)
(5 rows)

[postgres=# EXPLAIN SELECT * FROM otus.users WHERE surname like 'A%' order by id;
                                         QUERY PLAN
-----
Sort  (cost=14719.72..14719.97 rows=99 width=205)
  Sort Key: id
  -> Index Scan using idx_users_name_surname on users  (cost=0.42..14716.44 rows=99 width=205)
      Index Cond: ((surname ~>=~ 'A'::text) AND (surname ~<~ 'B'::text))
          Filter: (surname ~~ 'A%')::text)
(5 rows)

[postgres=# EXPLAIN SELECT * FROM otus.users WHERE name like 'A%' order by id;
                                         QUERY PLAN
-----
Sort  (cost=11.73..11.98 rows=99 width=205)
  Sort Key: id
  -> Index Scan using idx_users_name_surname on users  (cost=0.42..8.45 rows=99 width=205)
      Index Cond: ((name ~>=~ 'A'::text) AND (name ~<~ 'B'::text))
          Filter: (name ~~ 'A%')::text)
(5 rows)

postgres=#

```

Явное указание класса операторов, для которого строится индекс `text_pattern_ops` – именно это позволяет `LIKE` использовать индекс при сопоставлении шаблона с начала строки, в противном случае `LIKE` может игнорировать данный индекс.  
По сколько мы используем префикс сразу и `name` и `surname`, то мы можем использовать `Btree` индекс. Так же можно улучшить индекс, сделав его всегда в нижнем регистре:

```

CREATE INDEX idx_users_name_surname
ON otus.users USING btree
(lower(name) text_pattern_ops, lower(surname) text_pattern_ops);

```

Но после этого и поиск надо будет приводить к нижнему регистру