|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Интеллектуальные системы информационной безопасности»

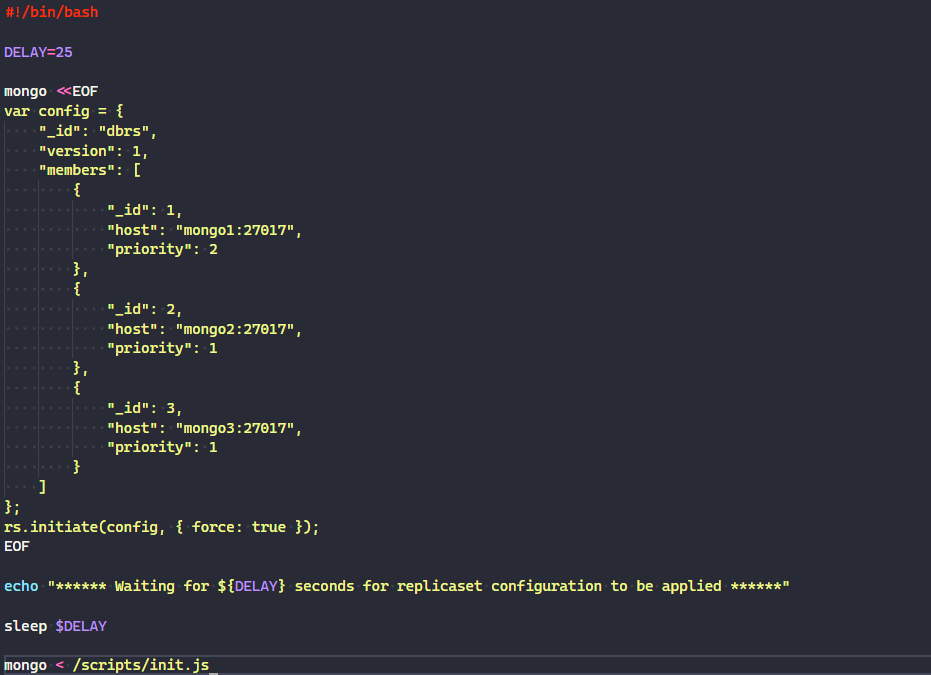
### Администрирование баз данных

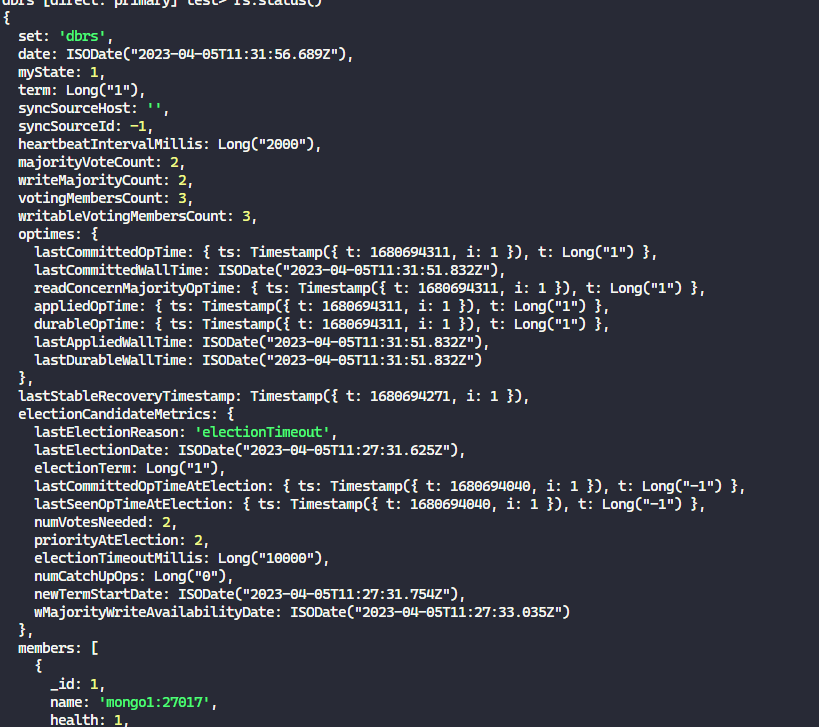
### Практическая работа № 6

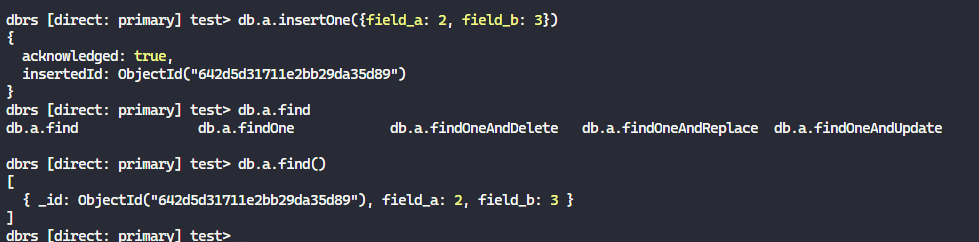
# ОТЧЁТ

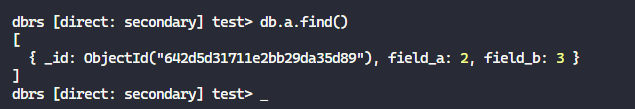
# Выполнил студент группы БСБО-07-20 Любовский С.В.

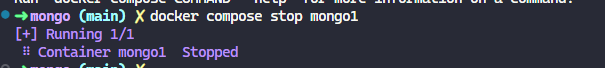
# Выполнение задания 1.

1-2. Создадим конфигурацию кластера используя docker-compose  
Используя скрипт для подготовки базы включим репликацию на всех узлах  
После успешной установки и запуска контейнеров зайдем в консоль primary ноды и увидим результат

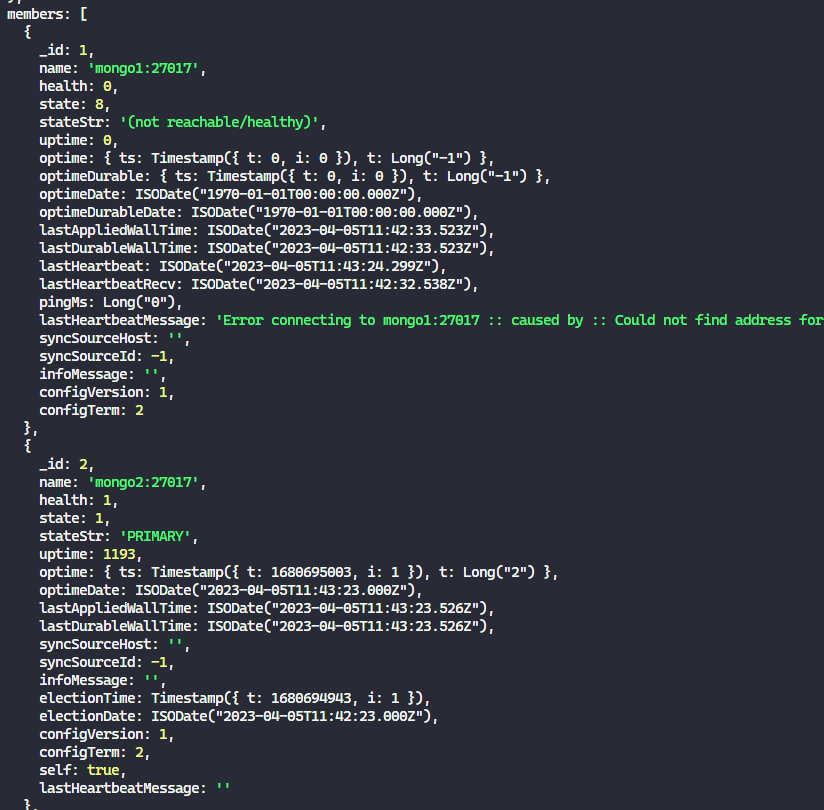


3. Создадим коллекцию на primary узле и добавим туда данных  


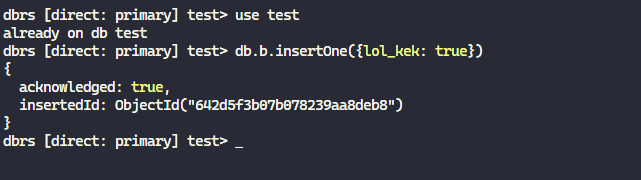
Перейдем на другой узел и проверим состояние коллекции  
  
данные успешно реплицируются между нодами.

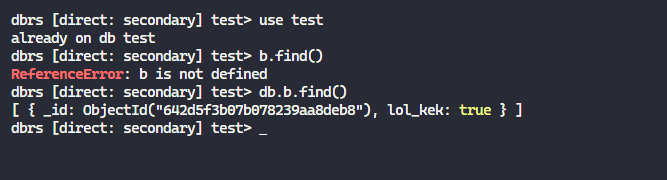
4. Остановим наш primary узел  


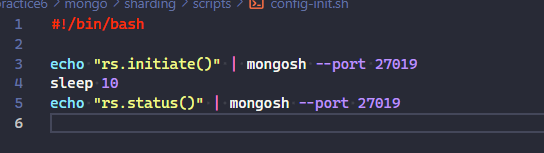
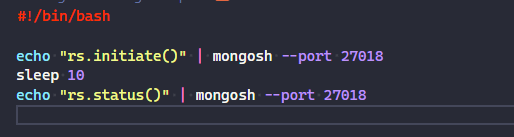
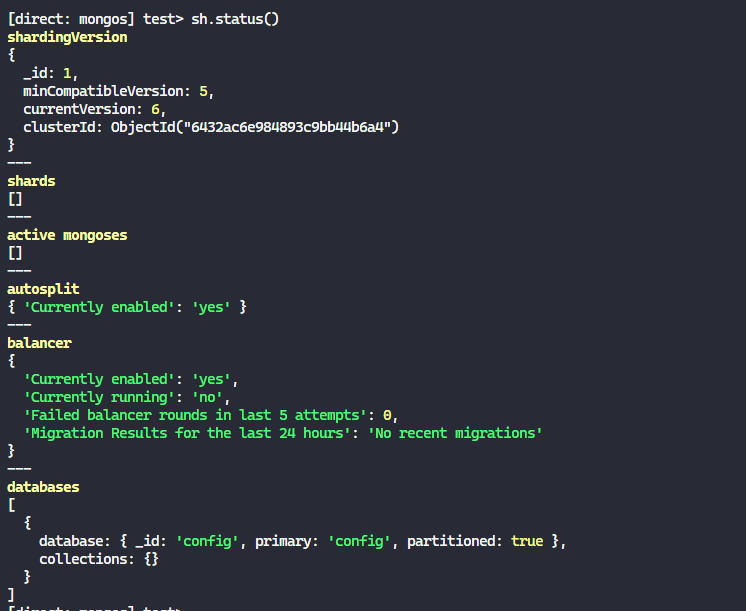
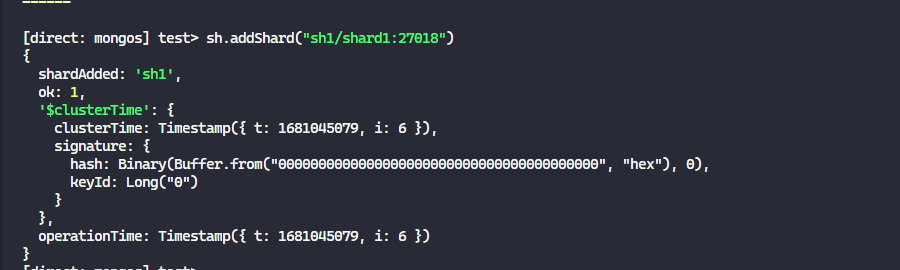
Зайдем на один из узлов и посмотрим кто теперь главный

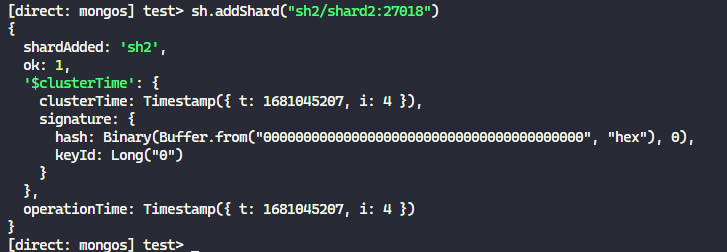


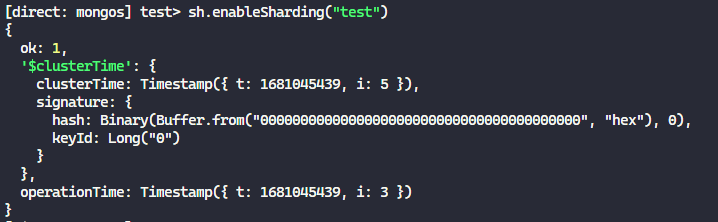
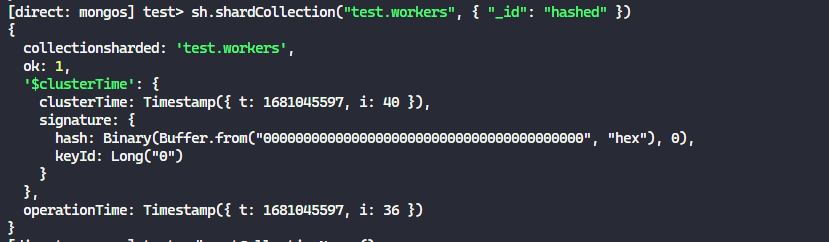
Вторая нода стала главной.

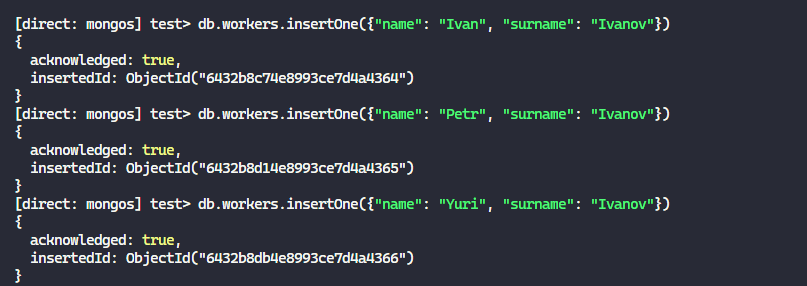
5. Снова добавим коллекцию и проверим репликацию  


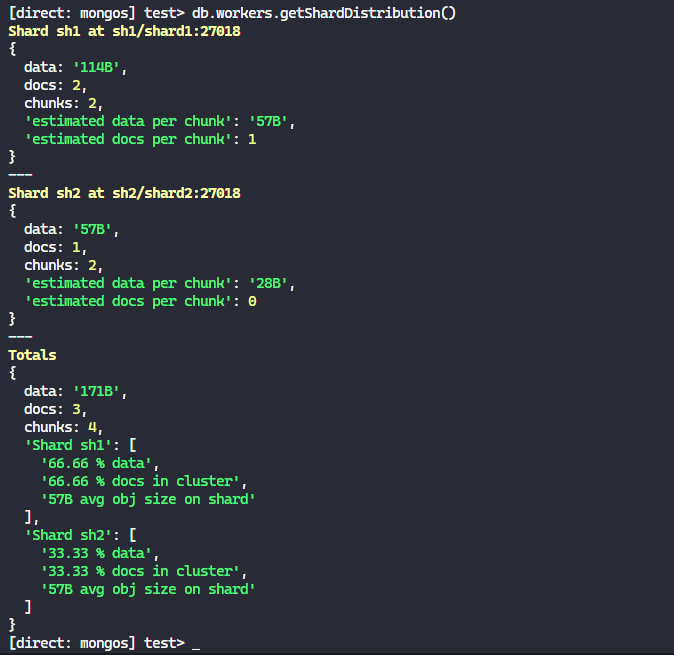
  
успешно реплицируется.

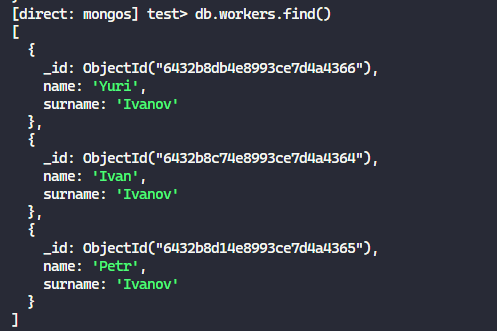
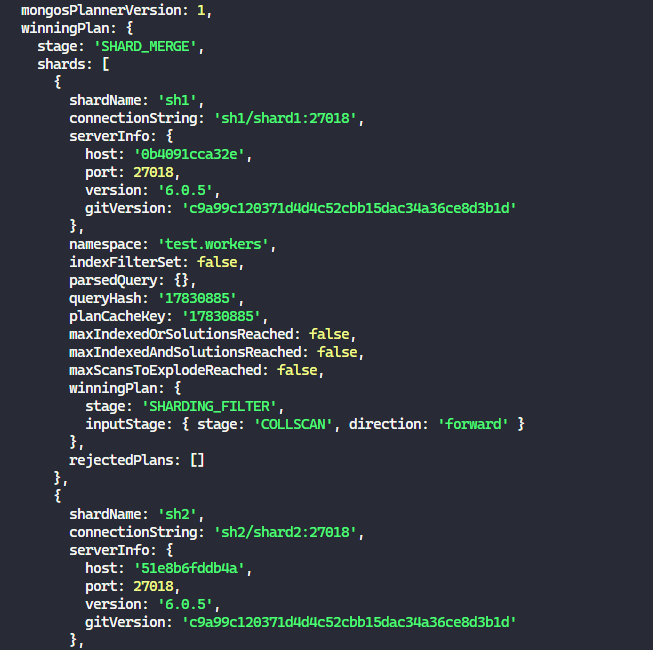
6. Теперь создадим кластер с щардированием. Для этого создадим новую конфигурацию docker-compose. Сконфигурируем 4 сервиса – конфиг-сервер, 2 шарда и mongos (роутер в шардированном кластере).   
Теперь поднимем нашу конфигурацию начиная с сервера и инициализируем репликасеты и необходимые параметры репликации.  
Скрипт инициализации конфиг-сервера  
  
Скрипт инициализации шардов  
  
  
Теперь зайдем в наш mongos сервис и пропишем следующую команду для проверки подключения к config-server  
  
видно, что конфиг сервер успешно подключен. Теперь подключим наши шарды.  




7. Активируем шардирование на базе test  
  
Активируем шардирование на коллекции workers по полю \_id используя метод шардирования hashed  


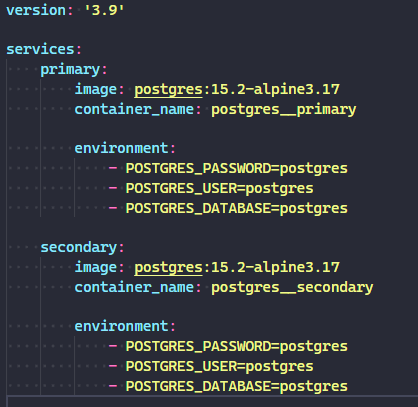
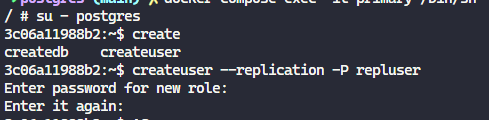
Добавим данных в коллекцию  
  
Проверим статус шардирования коллекции

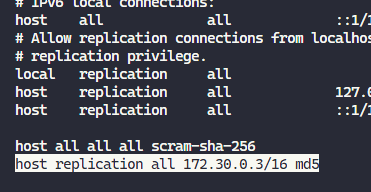


8. Попробуем получить данные распределенные по шардам  
  
Посмотрим explain запроса  
  
Видно что в плане присутствует MERGE из двух шардов.

# Выполнение задания 2.

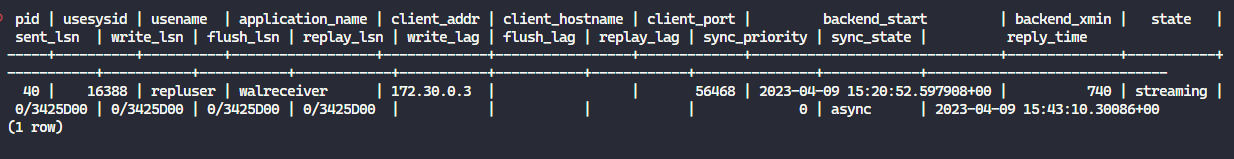
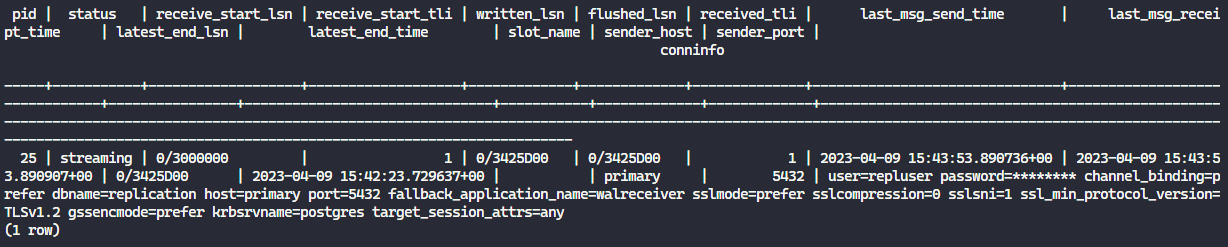
1. Создадим два сервиса (мастер и реплика)

  
2. Настроим мастер  
Создадим юзера для репликации  
  
Настроим след параметры  
wal\_level = replica  
max\_wal\_senders = 2  
max\_replication\_slots = 2  
hot\_standby = on  
hot\_standby\_feedback = on

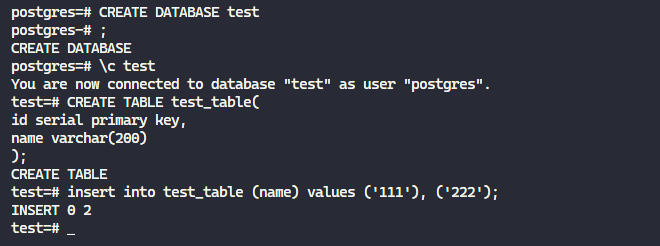
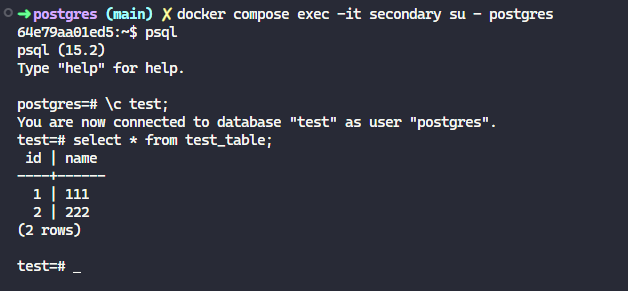
Настроим разрешенные для реплик адреса  
  
Перезапустим и перейдем к настройке сервера реплики.

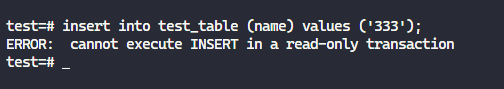
Для начала удалим все данные из папки data, далее восстановим данные с нашего матер сервера используя команду   
pg\_basebackup --host=primary --username=repluser --pgdata=/var/lib/postgresql/data --wal-method=stream --write-recovery-conf

Теперь проверим статусы репликации:

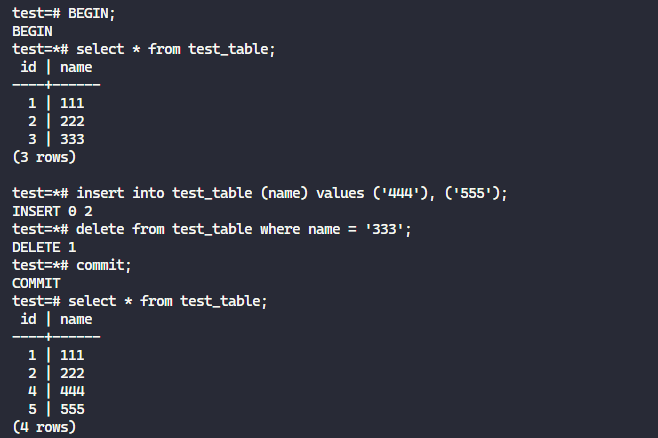
* на мастере используя команду  
  docker compose exec -it primary su - postgres -c "psql -c 'select \* from pg\_stat\_replication;'"  
  
* на реплике используя команду  
  

Видим, что настройка прошла успешно.

3-4. Cоздадим базу и таблицу с данными на мастере и проверим, что данные реплицируются.  
  
  
  
  
Данные успешно реплицируются.

5. Попробуем создать запись на реплика-сервере  


Создать запись не получается, т.к. реплика-сервер находится в режиме read-only.

6-7.   
Чтение и запись на мастер сервере работают корректно.