

## ЛР2. MongoDB. Аутентификация. Управление привилегиями

### Теоретическая (тестовая) часть

Лекция 2. Управление доступом к данным [Общая часть / без управления привилегиями средствами SQL]

Лекция 3. Учетные записи. Аутентификация [Общая часть + MongoDB]

Лекция 4. Управление привилегиями [Общая часть + MongoDB]

### Практическая часть

1. Настроить SCRAM-аутентификацию для сервера MongoDB.
2. Создать в выбранной БД (из ЛР1) следующих пользователей, использующих встроенные роли:
  - a. `owner` – может выполнять любые действия с БД.
  - b. `reader` – имеет права на чтение данных из всех коллекций.
  - c. `editor` – имеет права на чтения и запись данных из всех коллекций.
  - d. `blocked` – права отсутствуют.

Продемонстрировать выполнение запросов на чтение и запись данных от каждого пользователя.

3. Создать представление `filmStats`, возвращающее агрегированные характеристики фильмов из коллекции `films` по категории: количество фильмов в категории, средняя продолжительность фильма.
4. Создать в выбранной БД следующие пользовательские роли:
  - a. `view_read` – позволяет читать данные из представления `filmStats`.
  - b. `films_edit` – позволяет редактировать (добавлять / изменять / удалять) данные в коллекции `films`.
5. Создать пользователя `view_reader`, назначив ему роль `view_read`. Показать назначенные пользователю роли. Продемонстрировать наличие привилегий, выполнив запросы на чтение данных из представления.
6. Создать пользователя `films_editor` с правами на чтение данных из всех коллекций и на редактирование данных в коллекции `films` (с помощью роли `films_edit`). Продемонстрировать привилегии, выполнив запросы модификации данных.

### Бонусная часть (1 балл)

Написать Python-скрипт автоматической проверки задания на ЛР: проверить корректность работы представления, для каждого пользователя проверить наличие соответствующей роли и привилегий (выполнением запросов).