

### Этап 3. Администрирование и оптимизация

**Задачи:** повышение производительности и отказоустойчивости СУБД.

**Форма отчетности:** письменный отчет, демонстрация работы БД.

**Методические указания:**

В первую очередь необходимо написать и выполнить запросы к базе данных, выданные преподавателем в момент защиты второго этапа. Для того чтобы оценить и сравнить производительность различных запросов, вам потребуется создать сервис (на любом ЯП), который будет вызывать Explain Analyze для каждого запроса и измерять Cost. Полученные результаты форматировать и записывать в отдельный файл: лучший, средний, худший случай для каждого запроса. (Каждый новый запуск сервиса пишет новый файл, старые сохраняются. Кол-во попыток к каждому запросу определяется в env)

Далее добавьте индексы к вашим отношениям и запустите сервис снова. Повторите процедуру несколько раз и попытайтесь добиться оптимальной производительности для каждого запроса. Сделайте сводную таблицу на сколько процентов ваши запросы стали работать лучше/хуже от первого запуска до крайнего.

Так же вы можете попробовать посмотреть Explain запроса и переписать его(допустим используя CTE) для улучшения производительности. Так же можете повесить триггер для деморализации некоторых данных. (Данный параграф является необязательным.)

Создайте партиципаны для таблицы имеющей больше всего записей. Подумайте каким способом лучше делить таблицу. Снова запустите сервис и полученные результаты добавьте в сводную таблицу.

Напишите скрипт, который будет создавать бэкапы базы данных, каждые n-часов, последние m-бекапов должны храниться, более старые должны удаляться. (параметры передаются через env)

Наконец, для отказоустойчивости вашего кластера СУБД разверните 2 реплики Postgres. С использованием Patroni. Проверьте что при отказе мастера происходит автоматическое переключение.