**Воркшоп: реальный Postgres**

*(заполняется спикером урока)*

**Что нужно знать для проверки ДЗ**

1) От студента будет требоваться создать метод, доступный по сети, и метод, содержащий SQL-запрос.

Для этого студент должен отредактировать 2 файла - "DogsAppServices" и "EntireProjectRepo".

В DogsAppServices студент должен написать код по примеру из воркшопа. От студента будет требоваться реализовать метод с именем "getTop10WalkingUsers", который будет принимать сетевой запрос по адресу "top\_10\_walking\_users", и вызывать код из класса "EntireProjectRepo".

В классе EntireProjectRepo от пользователя будет требоваться реализовать метод "getTop10WalkingUsersFromDb", который должен содержать в себе SQL-запрос в базу данных, отвечающий на вопрос "Верни мне логины 10-и пользователей, у которых наибольшее количество прогулок".

2) Хорошо бы, чтобы проверяющий понимал, что значат SQL-запросы, конкретно - как работают операторы SELECT, COUNT(\*), GROUP BY, ORDER BY, LIMIT. Я ожидаю, что это все рассказывалось в курсе до моего воркшопа. Если проверяющий не знаком с частью этих операторов, или ни с какими из них - не беда, я попытался расписать то, что мы зачитаем за сделанную домашку, довольно формально.

**Чек-лист:**

1) Для того, чтобы выполнить часть задания в классе DogsAppServices, студент должен написать код, содержащий текст

@GetMapping("/top\_10\_walking\_users")

Эта надпись привязывает метод, который последует за ней, к сети.

Также, студент в этом же классе DogsAppServices должен определить и реализовать метод с именем getTop10WalkingUsers и вставить в его тело вызов метода getTop10WalkingUsersFromDb, который реализован в классе EntireProjectRepo. Пример того, как выглядит зачтенная часть работы в классе DogsAppServices

@GetMapping("/top\_10\_walking\_users")

public List<String> getTop10WalkingUsers() {

return entireProjectRepo.getTop10WalkingUsersFromDb();

}

Зачитаем любой код, содержащий текст @GetMapping("/top\_10\_walking\_users")

и вызов entireProjectRepo.getTop10WalkingUsersFromDb()

2) Вторая часть работы должна быть сделана в классе EntireProjectRepo. В этом классе требуется реализовать метод, выполняющий SQL-запрос по аналогии с тем, как мы писали запросы в этом воркшопе (так что от студента ожидается, что он сможет написать запрос по примеру). Сам SQL-запрос этого метода должен иметь вид

select user\_login from public.walks group by user\_login order by count(\*) desc limit 10

Где "select" - команда того, что будет производиться выборка, "user\_login" значит, что будут выбраны именно логины пользователей, "from public.walks" означает, что выбор будет из таблицы прогулок, "group by user\_login order by count(\*)" означает, что мы посчитаем, сколько строк в таблице WALKS у каждого пользователя и отсортируем этот подсчет по убыванию. "limit 10" означает, что мы возьмем первые 10 строк из списка, отсортированного по убыванию. Это нам и нужно.

Зачитаем любой запрос, у которого после "select" написано "user\_login", после from написано "public.walks" или "walks" (можно капс-локом, это неважно), в котором встречается "group by user\_login order by count" (после "count" может быть (\*) или что-то другое в скобках, это зачитаем); также у запроса должно быть "limit 10" в конце.

Пример зачитаемого запроса

public List<String> getTop10WalkingUsersFromDb() {

return jdbcTemplate.queryForList("select user\_login from public.walks group by user\_login order by count(\*) desc limit 10", String.class);

}

Я изначально дам шаблон

public List<String> getTop10WalkingUsersFromDb() {

return jdbcTemplate.queryForList("", String.class);

}

Так что от пользователя будет требоваться только написать SQL-запрос

Как поправлять студента:

Если у студента в SQL-запросе после "from" написано не "walks", или "public.walks", то нужно написать "Количество прогулок для каждого пользователя нужно искать в таблице Walks".

Если у пользователя нету "group by" в его SQL-запросе - писать "нам нужно сгруппировать все прогулки по логину пользователя, для группировки есть отдельный оператор, стоит использовать его"

Если у пользователя нету "order by" в его SQL-запросе - писать "нам нужно отсортировать сгруппированную выборку по количеству строк в каждой из групп, как с помощью какого оператора это можно сделать? Подсказка: сортировать группы можно не только по значению конкретного столбца, но и результату функции, отработавшей на целой группе - например, max, min, count"

Если у пользователя нету "limit 10" в его SQL-запросе - писать "Нам нужно выбрать топ-10 пользователей, какой SQL-оператор позволяет ограничить кол-во строк в выборке?"

3) Формат, в котором сдается ДЗ: Я попрошу студента скинуть 2 файла - получившиеся у него классы "DogsAppServices" и "EntireProjectRepo". Открываем их в любом текстовом редакторе и смотрим, что нужные изменения есть.

**Итого, минимум, чтобы принять ДЗ**

- Студент написал в классе "DogsAppServices" метод с именем getTop10WalkingUsers, над которым есть текст @GetMapping("/top\_10\_walking\_users"), и в теле которого вызывается getTop10WalkingUsersFromDb()

И написал в классе "EntireProjectRepo" метод, в котором есть SQL-код, выбирающий топ-10 пользователей с максимальным количеством прогулок, например "select user\_login from public.walks group by user\_login order by count(\*) desc limit 10"

**Само домашнее задание из примера (для вашего понимания):**

Сделать User Story Map для своего продукта в Miro

Переложить ее на верхнеуровневый Roadmap на 3-6 мес в таблице (как если бы была задача показать топ-менеджменту крупными мазками, над чем ведется работа подразделения)

Прочитать все приложенные к уроку статьи и сказать, какая понравилась больше всего ;)

Если нет своего продукта, то вы можете:

1. Использовать рабочий проект – работайте над реальными кейсами своего продукта.
2. Придумать свою идею.
3. Если придумать совсем не получается, то можете использовать кейсы, предлагаемые в воркшопе.