

## Задание 1. Напиши SQL-запрос, помоги коллегам

Парикмахерская находится по адресу: г. Тверь, ул. Трехсвятская, д. 3, это центральный район. Выберем людей, которые подходят по местоположению

### Запрос

В идеале сделать условие, чтобы абонент за последние 7 дней появился в указанной местности не менее 2х раз (в разные дни), но данных не так много поэтому тут это излишне.

### Картинка

```
1 SELECT DISTINCT msisdn AS USER FROM geo_d
2 WHERE geo_area_key IN (SELECT geo_area_key FROM cities_regions
3                        WHERE city_name = 'Тверь' AND city_area = 'Центральный')
4
```

### Текст

```
SELECT DISTINCT msisdn AS USER FROM geo_d
WHERE geo_area_key IN (SELECT geo_area_key FROM cities_regions
                        WHERE city_name = 'Тверь' AND
city_area = 'Центральный')
```

## Способ привлечения клиентов

Люди делятся на два типа

1. Люди, которые не живут и не работают в Центральном районе Твери (появляются в нужной местности меньше раза в неделю). Для них открывшаяся парикмахерская будет неактуальна. Максимум можно послать смс-уведомление разок-два
2. Люди, которые живут, либо работают в Центральном районе Твери.

Вторая группа людей является целевой аудиторией.

С ней можно поступить следующим образом:

Парикмахерская устраивает акцию – стрижка бесплатно, например на неделю. ЦА пускаем через такую воронку

- В начале смс уведомление об акции, адресе
- На следующий день звонок из call-центра, рассказ об акции и (если такое возможно в МТС) запись на свободное время
- Если человек по звонку сказал, что ему неинтересно, то в следующие 2 дня присылаем ему по 1 смс. Если согласился, то за день до приема напоминаем о записи.

## Задание 2. Напиши «красочный» SQL-запрос

Запрос:

```
WITH a1 as (  
  SELECT 1 AS id, 'Красный' as color union all  
  SELECT 2, 'Зеленый' as color union all  
  SELECT 3, 'Синий' as color union all  
  SELECT 4, 'Желтый' as color union all  
  SELECT 5, 'Фиолетовый' as color union all  
  SELECT 6, 'Пурпурный' as color union all  
  SELECT 7, 'Белый' as color),  
a2 AS (  
  SELECT 1 AS id, 'Красный' as color union all  
  SELECT 2, 'Зеленый' as color union all  
  SELECT 3, 'Синий' as color union all  
  SELECT 4, 'Желтый' as color union all  
  SELECT 5, 'Фиолетовый' as color union all  
  SELECT 6, 'Пурпурный' as color union all  
  SELECT 7, 'Белый' as color)  
  
SELECT a1.color, a2.color FROM a1  
join a2  
WHERE a1.id<a2.id  
|
```

Результат:

color	color
Красный	Зеленый
Зеленый	Синий
Красный	Синий
Синий	Желтый
Зеленый	Желтый
Красный	Желтый
Желтый	Фиолетовый
Синий	Фиолетовый
Зеленый	Фиолетовый
Красный	Фиолетовый
Фиолетовый	Пурпурный
Желтый	Пурпурный
Синий	Пурпурный
Зеленый	Пурпурный
Красный	Пурпурный
Пурпурный	Белый
Фиолетовый	Белый
Желтый	Белый
Синий	Белый
Зеленый	Белый
Красный	Белый

Объяснение:

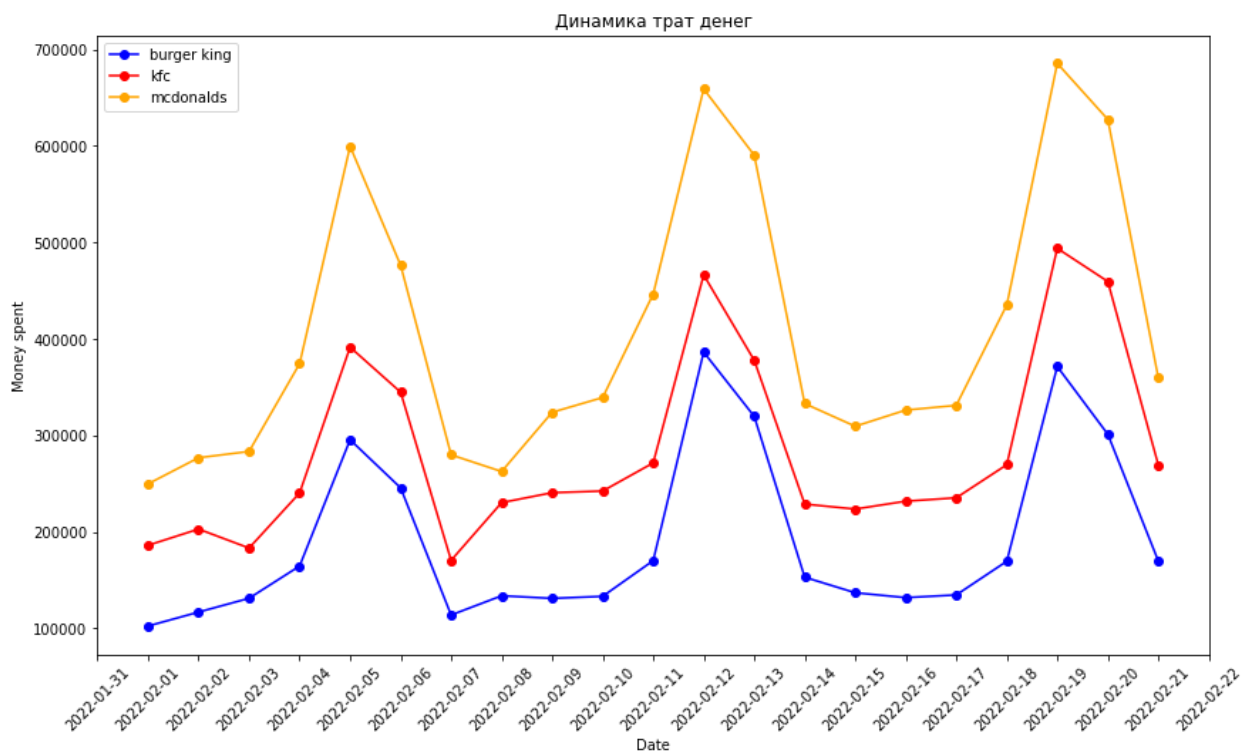
Тут задача сочетания без повторов из 7 по 2. Join без условия дает декартово произведение таблиц, поэтому потребуем, чтобы id цвета #1 был

меньше id цвета #2, так мы избавимся от дублей с точностью до перестановки и от одноцветных пар.

### Задание 3. Построй диаграммы и графики для визуализации зависимостей

1) Какова динамика трат посетителей в ресторанах? Есть ли дни наибольшего спроса? Можно ли проследить тренд? Какой ресторан находится на втором месте по популярности?

Динамика трат. Группируем по ресторанам и датам, и берем сумму



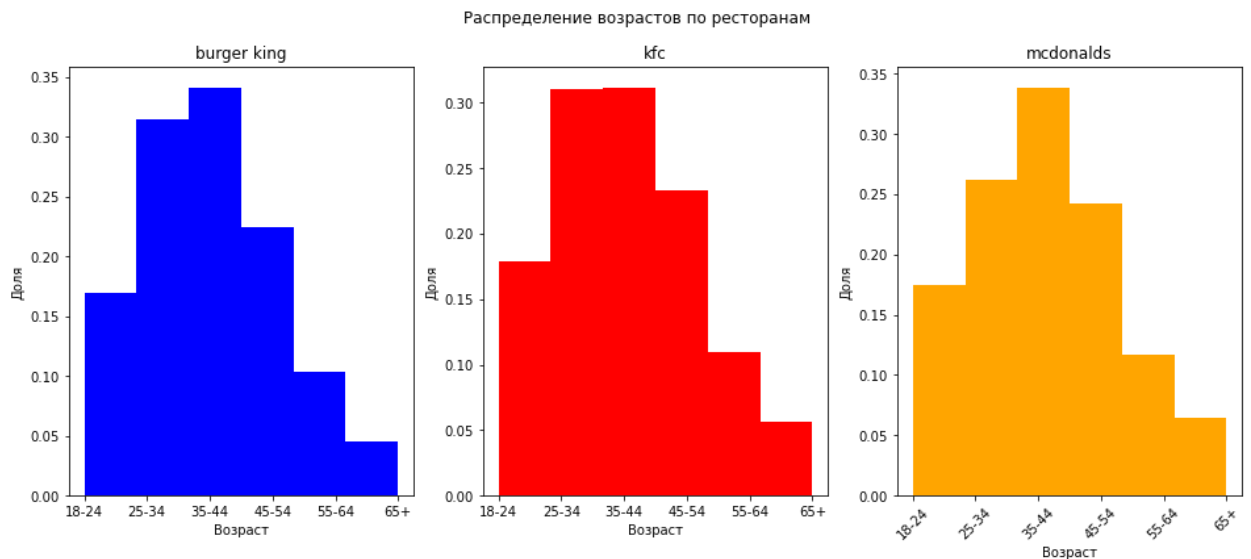
Тут отчетлива видна цикличность, недельная. Дни наибольшего спроса – выходные

Заметного тренда нет

На втором месте по популярности – KFC

2) Каково распределение покупателей по возрасту в каждом из ресторанов?

Группируем по возрасту и ресторану. Распределения похожи.



### 3) Как можно визуализировать одновременно три параметра: сумму покупок, средний чек и количество покупателей?

Используем две группировки одновременно. Средний чек = сумма покупок / кол-во покупателей.

