**SQL** Рассчитываем среднее значение чаевых для пользователей, которые выбирают больше двух комплектов приборов и для остальных, при условии, что сумма заказа не равна 0.

SELECT C1\_2.AVG\_tips as up\_2\_cutlery, C3\_6.AVG\_tips as more\_2\_cutlery

from

(SELECT round(AVG(tips)) AS AVG\_tips

FROM shmya

WHERE cutlery < 3 and order\_price <> 0 ) as C1\_2,

(SELECT round(AVG(tips)) AS AVG\_tips

FROM shmya

WHERE cutlery > 2 and order\_price <> 0 ) as C3\_6;

Запрос для вывода UID пользователей, которые добавляли в заказ больше двух комплектов столовых приборов и делали заказ не в январские праздники на сумму больше 800p.

SELECT

  distinct uid

FROM shmya

WHERE order\_price > 800 and date between '2022-01-10' and '2022-02-01' and cutlery > 2

Запрос для расчёта количества таких пользователей.

SELECT

  count(distinct uid) as total\_uid

FROM shmya

WHERE order\_price > 800 and date between '2022-01-10' and '2022-02-01' and cutlery > 2

**Python** Рассчитываем среднее значение чаевых для пользователей, которые выбирают больше двух комплектов приборов и для остальных, при условии, что сумма заказа не равна 0.

import pandas as pd

df=pd.read\_csv('shmya.csv')

tips\_1\_2 = df[(df['cutlery']<=2)&(df['order\_price']>0)].agg(['mean']).tips

tips\_3\_6 = df[(df['cutlery']>2)&(df['order\_price']>0)].agg(['mean']).tips

Запрос для вывода UID пользователей, которые добавляли в заказ больше двух комплектов столовых приборов и делали заказ не в январские праздники на сумму больше 800p.

df[(df['date'] > '2022-01-10 00:00:00')&(df['date'] < '2022-02-01 00:00:00') & (df['cutlery']>2)&(df['order\_price']>800)].uid.unique()

Запрос для расчёта количества таких пользователей.

df[(df['date'] > '2022-01-10 00:00:00')&(df['date'] < '2022-02-01 00:00:00') & (df['cutlery']>2)&(df['order\_price']>800)].uid.unique().size