Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**
Институт космических и информационных технологий

# ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

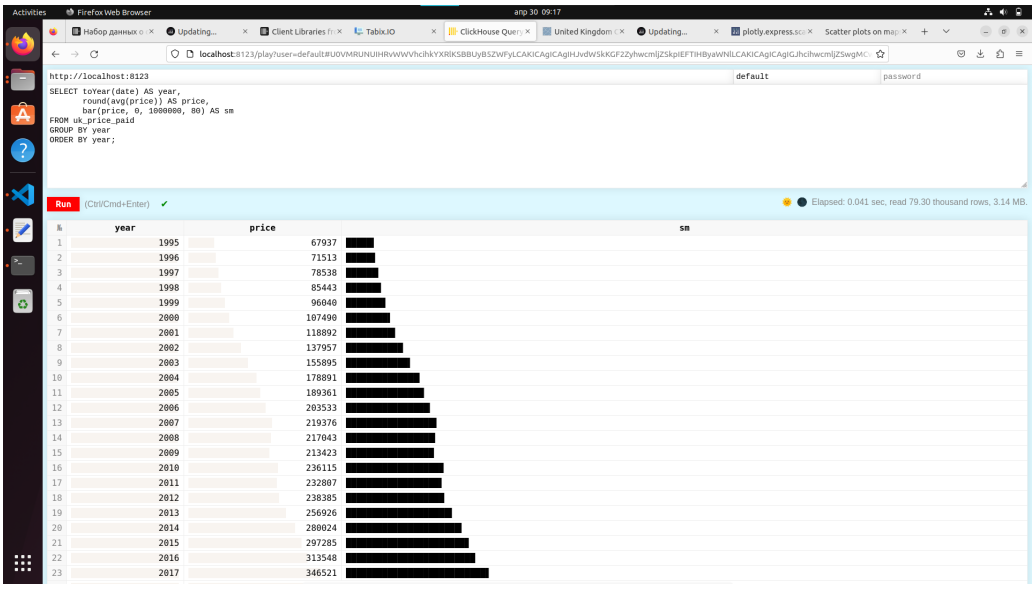
Python, как инструмент получения данных для оперативной аналитики. Колоночная СУБД.

Преподаватель _____ Полякова А.С.
дата, подпись

Студент КИ20-13Б, № з/к 031940535 _____ Панкратов М.Е.
дата, подпись

Красноярск 2023

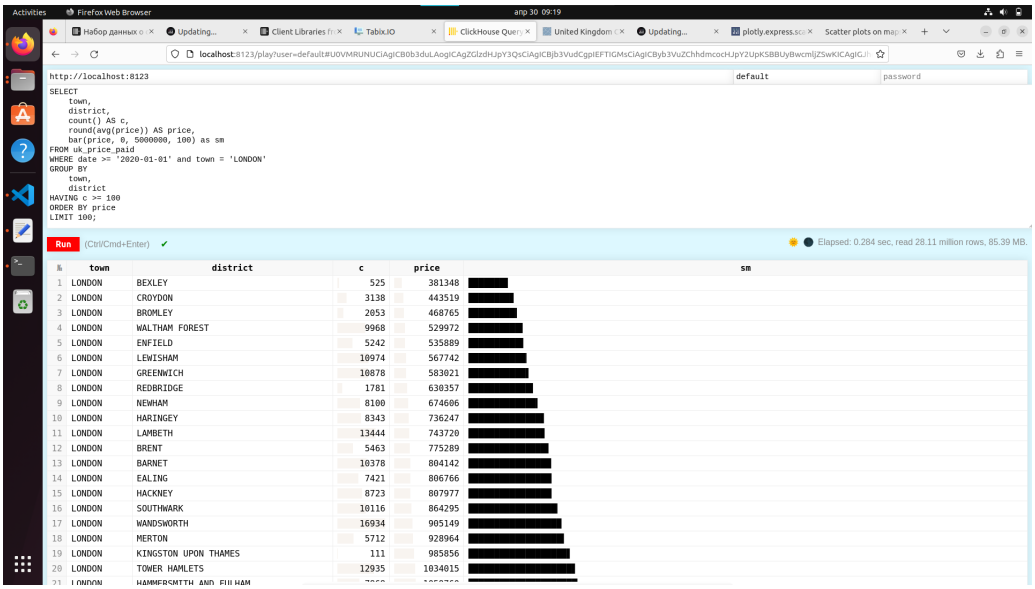
# ОСНОВНАЯ ЧАСТЬ



```
SELECT toYear(date) AS year,
       round(avg(price)) AS price,
       bar(price, 0, 1000000, 80) AS sm
FROM uk_price_paid
GROUP BY year
ORDER BY year;
```

| | year | price | sm |
|---|---|---|---|
| 1 | 1995 | 67937 | |
| 2 | 1996 | 71513 | |
| 3 | 1997 | 78538 | |
| 4 | 1998 | 85443 | |
| 5 | 1999 | 96040 | |
| 6 | 2000 | 107490 | |
| 7 | 2001 | 118892 | |
| 8 | 2002 | 137957 | |
| 9 | 2003 | 155895 | |
| 10 | 2004 | 178891 | |
| 11 | 2005 | 189361 | |
| 12 | 2006 | 203533 | |
| 13 | 2007 | 219376 | |
| 14 | 2008 | 217043 | |
| 15 | 2009 | 213423 | |
| 16 | 2010 | 236115 | |
| 17 | 2011 | 232807 | |
| 18 | 2012 | 238385 | |
| 19 | 2013 | 256926 | |
| 20 | 2014 | 280024 | |
| 21 | 2015 | 297285 | |
| 22 | 2016 | 313548 | |
| 23 | 2017 | 346521 | |

Рисунок 1.1 – Средняя стоимость жилья за каждый год;



```
SELECT
    town,
    district,
    count() AS c,
    round(avg(price)) AS price,
    bar(price, 0, 5000000, 100) as sm
FROM uk_price_paid
WHERE date >= '2020-01-01' and town = 'LONDON'
GROUP BY
    town,
    district
HAVING c >= 100
ORDER BY price
LIMIT 100;
```

| | town | district | c | price | sm |
|---|---|---|---|---|---|
| 1 | LONDON | BEXLEY | 525 | 381348 | |
| 2 | LONDON | CROYDON | 3138 | 443519 | |
| 3 | LONDON | BROMLEY | 2053 | 468765 | |
| 4 | LONDON | WALTHAM FOREST | 9968 | 529972 | |
| 5 | LONDON | ENFIELD | 5242 | 535889 | |
| 6 | LONDON | LEWISHAM | 10974 | 567742 | |
| 7 | LONDON | GREENWICH | 10878 | 583021 | |
| 8 | LONDON | REDBRIDGE | 1781 | 630357 | |
| 9 | LONDON | NEWHAM | 8100 | 674606 | |
| 10 | LONDON | HARINGEY | 8343 | 736247 | |
| 11 | LONDON | LAMBETH | 13444 | 743720 | |
| 12 | LONDON | BRENT | 5463 | 775289 | |
| 13 | LONDON | BARNET | 10378 | 804142 | |
| 14 | LONDON | EALING | 7421 | 806766 | |
| 15 | LONDON | HACKNEY | 8723 | 807977 | |
| 16 | LONDON | SOUTHWARK | 10116 | 864295 | |
| 17 | LONDON | WANDSWORTH | 16934 | 905149 | |
| 18 | LONDON | MERTON | 5712 | 928964 | |
| 19 | LONDON | KINGSTON UPON THAMES | 111 | 985856 | |
| 20 | LONDON | TOWER HAMLETS | 12935 | 1034015 | |

Рисунок 1.2 – Самые дорогие районы в Лондоне в порядке возрастания цены;

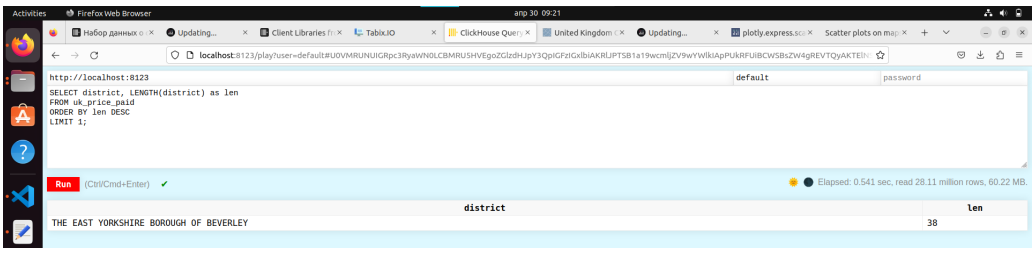



Рисунок 1.3 – Название самой длинной улицы в Великобритании;

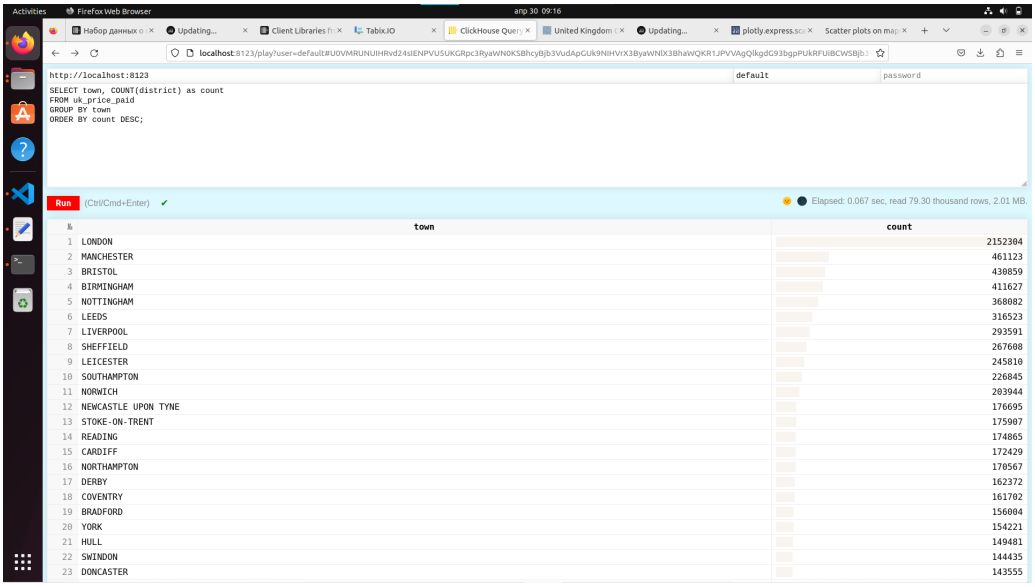



Рисунок 1.4 – Количество районов в каждом городе;

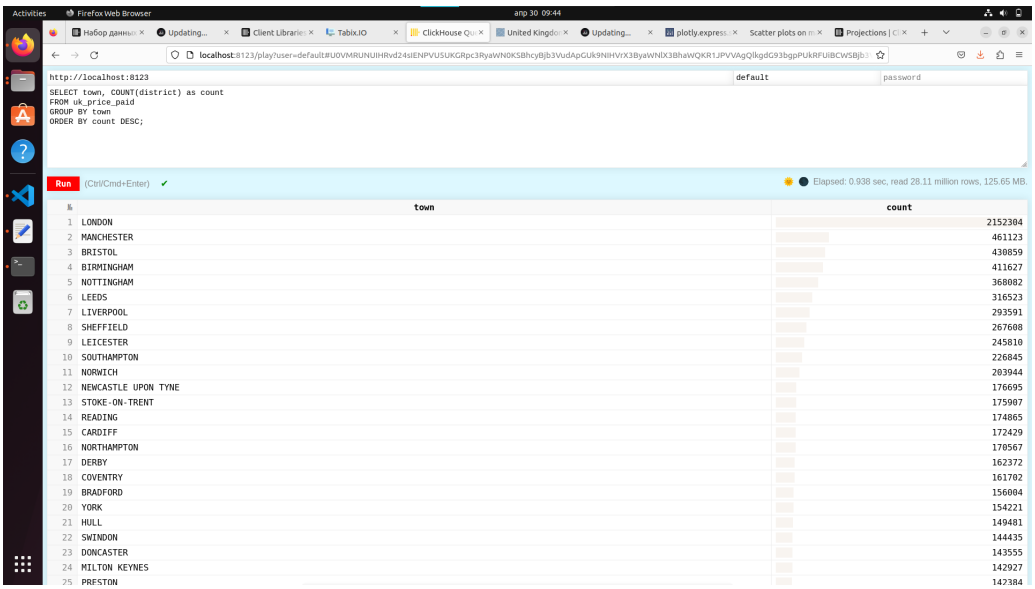



Рисунок 1.5 – Без проекции;

**English is the Caiptal of Great Britain**

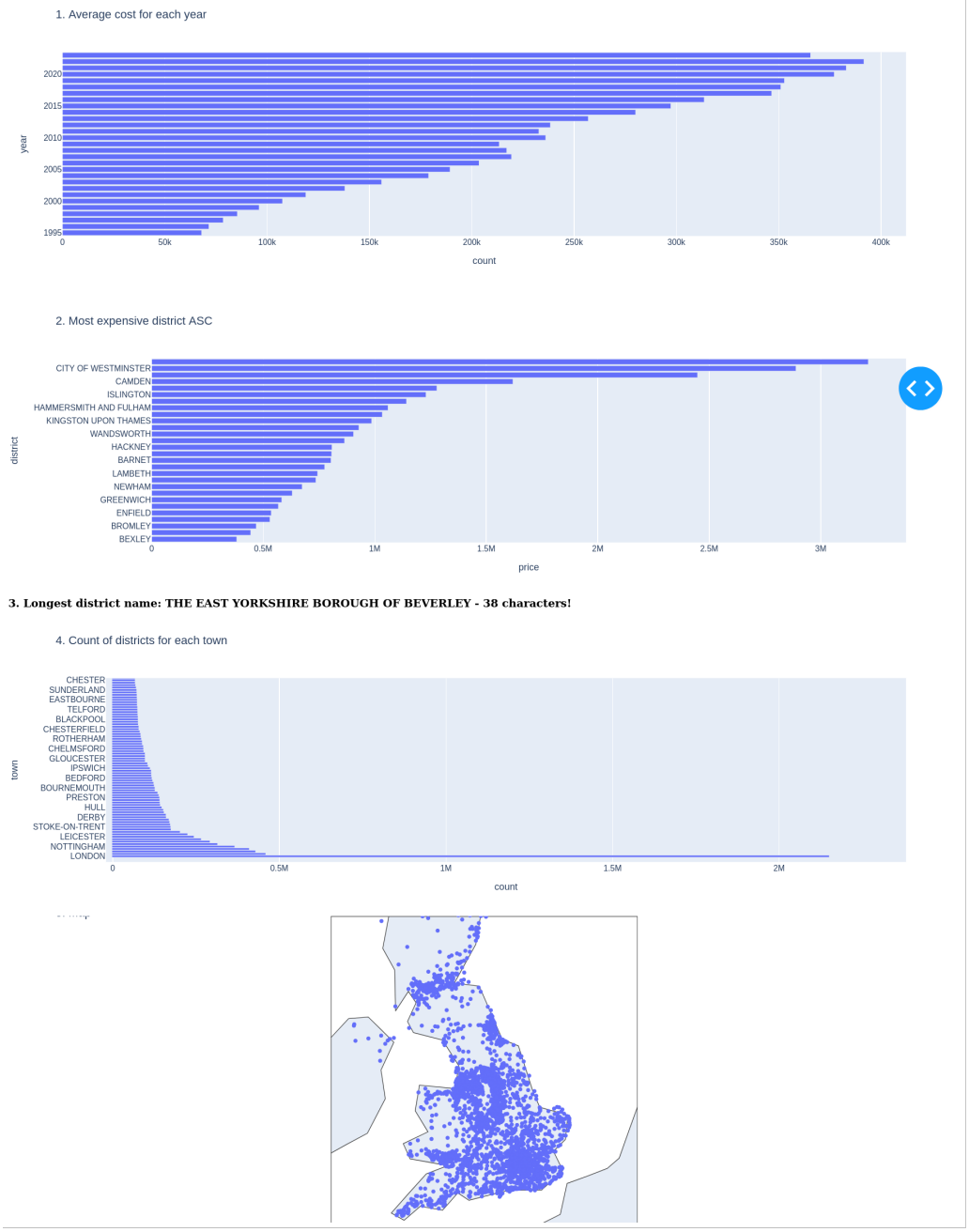



Рисунок 1.6 – Диаграммы;

```
from dash import Dash, html, dcc
import plotly.express as px
import plotly.graph_objects as go
import pandas as pd
from clickhouse_driver import Client
from queries import *
```

```
7
8     client = Client('localhost')
9
10    request1 = client.execute(query1)
11    request2 = client.execute(query2)
12    request3 = client.execute(query3)
13    request4 = client.execute(query4)
14
15    app = Dash(__name__)
16
17    dt1 = {'year':[],
18           'count':[],
19           'sm':[]
20           }
21
22    for i in request1:
23        dt1['year'].append(i[0])
24        dt1['count'].append(i[1])
25        dt1['sm'].append(i[2])
26    df = pd.DataFrame(dt1)
27
28    dt2 ={'town':[],
29          'district':[],
30          'count':[],
31          'price':[],
32          'sm':[]}
33    for i in request2:
34        dt2['town'].append(i[0])
35        dt2['district'].append(i[1])
36        dt2['count'].append(i[2])
37        dt2['price'].append(i[3])
38        dt2['sm'].append(i[4])
39    df2 = pd.DataFrame(dt2)
40    dt4 ={'town':[],
41          'count':[],}
42    for i in request4:
43        dt4['town'].append(i[0])
44        dt4['count'].append(i[1])
45    df4 = pd.DataFrame(dt4)
46
47    # print(df) ,projection='mercator'
48    fig = px.bar(df, x="count", y="year", orientation='h',
49                 title='1. Average cost for each year')
50    fig2 = px.bar(df2, x="price", y="district", orientation='h',
```

```
                  title='2. Most expensive district ASC')
    fig4 = px.bar(df4, x="count", y="town", orientation='h',
                  title='4. Count of districts for each town')
    df = pd.read_json('gb.json')
    fig5 = px.scatter_geo(df,
                          lat=df.lat,
                          lon=df.lng,
                          hover_name="city",
                          center={'lat':55,'lon':-3},
                          title='5. Map',
                          projection='mercator')
    fig5.update_layout(margin={"r":0,"t":0,"l":0,"b":0})

    app.layout = html.Div(children=[
        html.H1(children='English is the Caiptal of Great Britain'),
        html.Div(children='''

        '''),
        dcc.Graph(
            id='first-gr',
            figure=fig
        ),
        html.Div(children='''

        '''),
        dcc.Graph(
        id='second-gr',
        figure=fig2
        ),
        html.B(children=f'''
        3. Longest district name: {request3[0][0]} - {request3[0][1]} characters!
        '''),
        html.Div(children='''

        '''),
        dcc.Graph(
        id='fourth-gr',
        figure=fig4
        ),
        dcc.Graph(id="graph",
                  figure=fig5),

    ])

```

```
95      if __name__ == '__main__':
96          app.run_server(debug=True, port=12891)
```

Листинг 1.1 – Код;