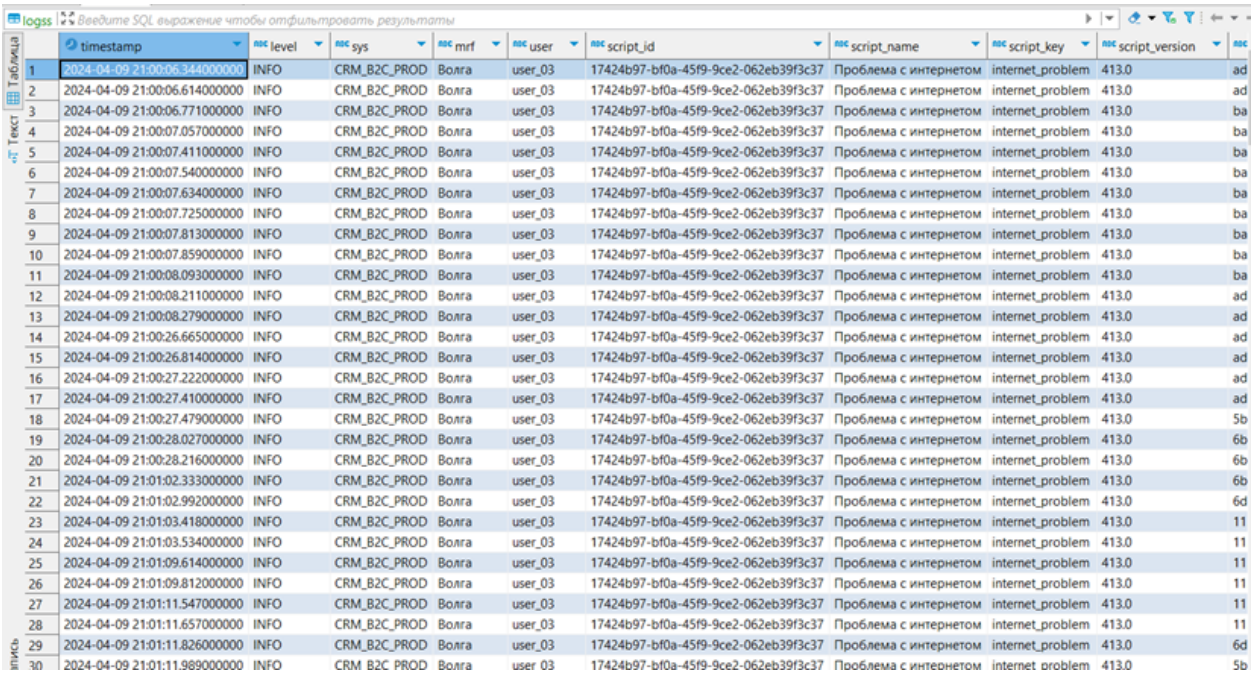
Тестовое ростелеком №2

Вначале у нас есть данные, которые были загружены в ClickHouse на первом этапе.



logss | Введите SQL выражение чтобы отфильтровать результаты

	timestamp	level	sys	mrf	user	script_id	script_name	script_key	script_version	
1	2024-04-09 21:00:06.344000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
2	2024-04-09 21:00:06.614000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
3	2024-04-09 21:00:06.771000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
4	2024-04-09 21:00:07.057000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
5	2024-04-09 21:00:07.411000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
6	2024-04-09 21:00:07.540000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
7	2024-04-09 21:00:07.634000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
8	2024-04-09 21:00:07.725000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
9	2024-04-09 21:00:07.813000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
10	2024-04-09 21:00:07.859000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
11	2024-04-09 21:00:08.093000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ba
12	2024-04-09 21:00:08.211000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
13	2024-04-09 21:00:08.279000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
14	2024-04-09 21:00:26.665000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
15	2024-04-09 21:00:26.814000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
16	2024-04-09 21:00:27.222000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
17	2024-04-09 21:00:27.410000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	ad
18	2024-04-09 21:00:27.479000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	5b
19	2024-04-09 21:00:28.027000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	6b
20	2024-04-09 21:00:28.216000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	6b
21	2024-04-09 21:01:02.333000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	6b
22	2024-04-09 21:01:02.992000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	6d
23	2024-04-09 21:01:03.418000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
24	2024-04-09 21:01:03.534000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
25	2024-04-09 21:01:09.614000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
26	2024-04-09 21:01:09.812000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
27	2024-04-09 21:01:11.547000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
28	2024-04-09 21:01:11.657000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	11
29	2024-04-09 21:01:11.826000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	6d
30	2024-04-09 21:01:11.989000000	INFO	CRM_B2C_PROD	Волга	user_03	17424b97-bf0a-45f9-9ce2-062eb39f3c37	Проблема с интернетом	internet_problem	413.0	5b

С помощью Python подключимся к базе данных и извлечем parameters и script_id из Clickhouse.

```
from clickhouse_driver import Client
import pandas as pd
import json

client = Client(host='localhost', port=9000, user='default', pas

query = 'SELECT parameters, script_id from db_test.logss'
result = client.execute(query)

df = pd.DataFrame(result, columns=['parameters', 'script_id'])
```

```python
client = Client(host='localhost', port=9000, user='default', password='pivanet')

query = 'SELECT parameters, script_id from db_test.logss'
result = client.execute(query)

df = pd.DataFrame(result, columns=['parameters', 'script_id'])
```

Мы получили следующий датафрейм.

	parameters	script_id
0	{}	17424b97-bf0a-45f9-9ce2-062eb39f3c37
1		17424b97-bf0a-45f9-9ce2-062eb39f3c37
2	{"TYPE_EQUIPMENT": ""}	17424b97-bf0a-45f9-9ce2-062eb39f3c37
3		17424b97-bf0a-45f9-9ce2-062eb39f3c37
4	{"TYPE_EQUIPMENT": "оборудованием"}	17424b97-bf0a-45f9-9ce2-062eb39f3c37
...	...	...
755	{"SERVICE_STATUS_LOCAL": "Включена"}	a67dfe4b-a531-4366-9d7c-cfa348256b20
756		a67dfe4b-a531-4366-9d7c-cfa348256b20
757	{"SERVICE_ID": "160502", "ACCOUNT_NUMBER": "ac...	a67dfe4b-a531-4366-9d7c-cfa348256b20
758		a67dfe4b-a531-4366-9d7c-cfa348256b20
759	{}	a67dfe4b-a531-4366-9d7c-cfa348256b20

760 rows × 2 columns

Согласно тексту задания, нам необходимо создать таблицу с группировкой по script_id. Эта таблица должна содержать данные, подходящие для обработки специалистами по BI. Таблица должна быть одна.

Можно заметить, что для script_id присутствуют одинаковые строки. Давайте посмотрим, какие уникальные значения содержатся в этом столбце:

```
df['script_id'].value_counts()
```

```
script_id
7e3cfde7-53a7-40a9-b814-c373df9d8d04    501
a67dfe4b-a531-4366-9d7c-cfa348256b20    160
17424b97-bf0a-45f9-9ce2-062eb39f3c37     99
Name: count, dtype: int64
```

Всего есть три script_id. Сделаем группировку.

```
def merge_dicts(dicts):
    result_dict = {}
    for dictionary in dicts:
        if isinstance(dictionary, dict):  # Убедимся, что элемен
            for key, value in dictionary.items():
                result_dict[key] = value  # Обновление значения
    return result_dict
import pandas as pd


# Группировка по script_id и агрегация словарей в parameters
aggregated_df = df.groupby('script_id')['parameters'].agg(merge_


aggregated_df
```

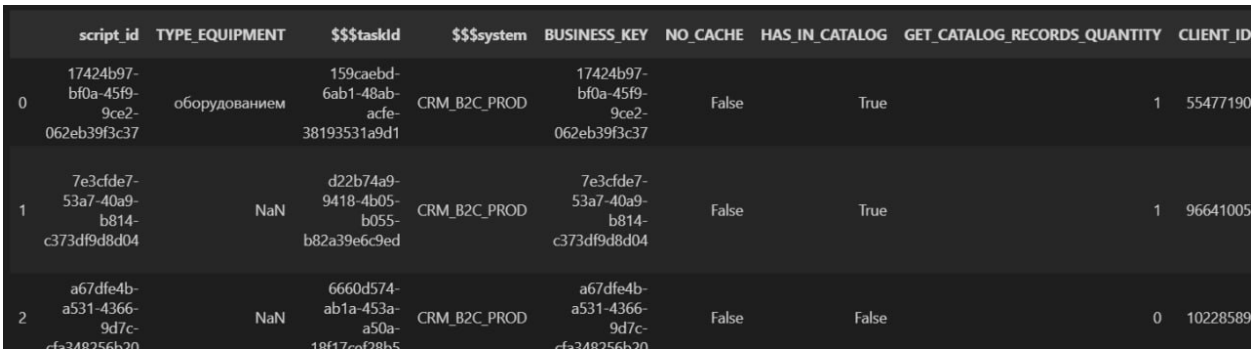
После группировки датафрейм выглядит следующим образом:

	script_id	parameters
0	17424b97-bf0a-45f9-9ce2-062eb39f3c37	{'TYPE_EQUIPMENT': 'оборудованием', 'CATALOG':...
1	7e3cfde7-53a7-40a9-b814-c373df9d8d04	{'GET_CLIENT_TIMEZONE': '+03:00', 'EMPLOYEE_MR...
2	a67dfe4b-a531-4366-9d7c-cfa348256b20	{'COMMUNICATION_NUMBER': '000000344206174', 'C...

То есть у нас есть script_id и параметры. Параметры представляют из себя словари со значениями. Нужно превратить колонку параметров в несколько

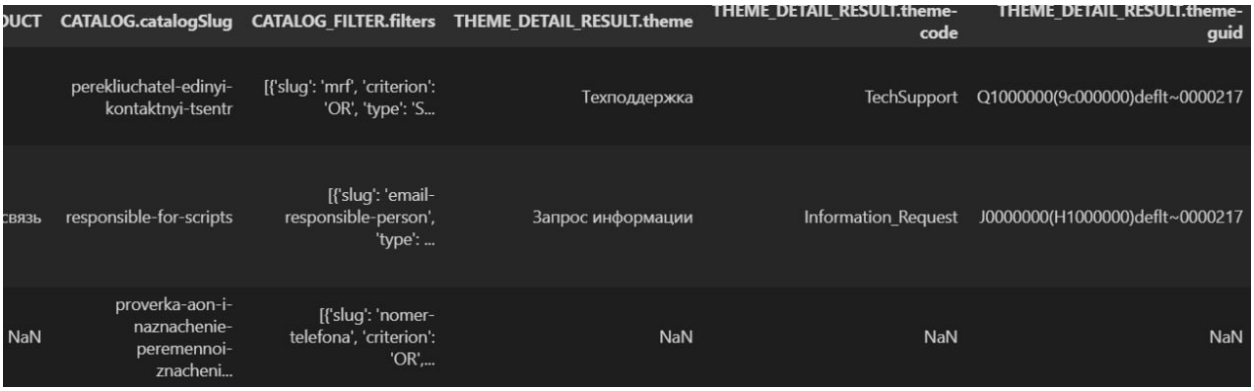
колонок. Сделать таблицу шире. По факту задача похожа на парсинг JSON-файлов.

Попробуем раскрыть словари исходного датафрейма на один уровень. Вот результат:



	script_id	TYPE_EQUIPMENT	$$$taskId	$$$system	BUSINESS_KEY	NO_CACHE	HAS_IN_CATALOG	GET_CATALOG_RECORDS_QUANTITY	CLIENT_ID
0	17424b97-bf0a-45f9-9ce2-062eb39f3c37	оборудованием	159caebd-6ab1-48ab-acfe-38193531a9d1	CRM_B2C_PROD	17424b97-bf0a-45f9-9ce2-062eb39f3c37	False	True	1	55477190
1	7e3cfde7-53a7-40a9-b814-c373df9d8d04	NaN	d22b74a9-9418-4b05-b055-b82a39e6c9ed	CRM_B2C_PROD	7e3cfde7-53a7-40a9-b814-c373df9d8d04	False	True	1	96641005
2	a67dfe4b-a531-4366-9d7c-cfa348256b20	NaN	6660d574-ab1a-453a-a50a-18f17cef28b5	CRM_B2C_PROD	a67dfe4b-a531-4366-9d7c-cfa348256b20	False	False	0	10228589

Я заметил, что после раскрытия словарей на один уровень остаются еще словари. А также списки. В списках обычно содержатся словари. Нужно написать функцию, которая будет превращаться ключи словарей в новые столбцы. Вот как выглядят словари в значениях:



UCT	CATALOG.catalogSlug	CATALOG_FILTER.filters	THEME_DETAIL_RESULT.theme	THEME_DETAIL_RESULT.theme-code	THEME_DETAIL_RESULT.theme-guid
	perekliuchatel-edinyi-kontaktnyi-tsentr	[{'slug': 'mrf', 'criterion': 'OR', 'type': 'S...	Техподдержка	TechSupport	Q1000000(9c000000)deflt~0000217
связь	responsible-for-scripts	[{'slug': 'email-responsible-person', 'type': ...	Запрос информации	Information_Request	J0000000(H1000000)deflt~0000217
NaN	proverka-aon-i-naznachenie-peremennoi-znacheni...	[{'slug': 'nomer-telefona', 'criterion': 'OR',...	NaN	NaN	NaN

Напишем функцию, которая проходится по колонкам датафрейма. Если значением ячейки является лист, и внутри листа лежит словарь, то мы достаем словарь Если находим словарь,

Напишем функцию, которая проходится по колонкам датафрейма. Если значением ячейки является лист, и внутри листа лежит словарь, то мы достаём этот словарь. Если находим словарь, мы далее анализируем его содержимое, чтобы определить, можно ли его развернуть в отдельные

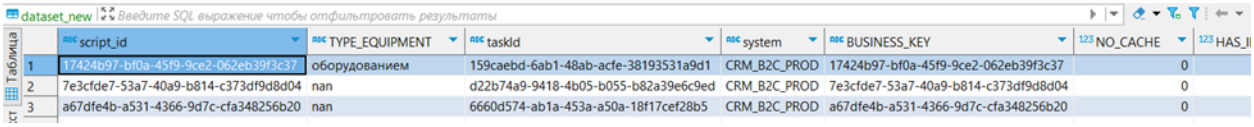
столбцы для улучшения структуры данных. Этот подход позволяет преобразовывать вложенные структуры данных в более плоский и анализируемый формат.

Применяя разработанные функции, мы последовательно идентифицируем столбцы, содержащие словари (`find_dict_columns`) и списки (`find_list_columns`). Это первый шаг в процессе структурирования данных. Для столбцов, содержащих словари, используется функция `expand_dict_columns`, которая преобразует каждый словарь в отдельные столбцы. Это позволяет более детально анализировать информацию, которая ранее была скрыта внутри словарей. Функцию можно посмотреть в коде.

После раскрытия мы получаем итоговый датафрейм, в котором 491 колонка. Он не содержит списки и словари в качестве значений.

```
Final result structure:
script_id                          object
TYPE_EQUIPMENT                     object
$$$taskId                          object
$$$system                          object
BUSINESS_KEY                       object
                                   ...
INTERACTION_TOPICS.result.guid     object
CASE_TYPES.type1.0                float64
CASE_TYPES.type1.title             object
CASE_TYPES.type1.id                object
CASE_TYPES.type1.guid              object
Length: 491, dtype: object
```

В именах столбцов присутствует недопустимый символ $. Сделаем предобработку. После этого будем делать вставку в нашу таблицу ClickHouse. После вставки посмотрим данные через DBeaver:



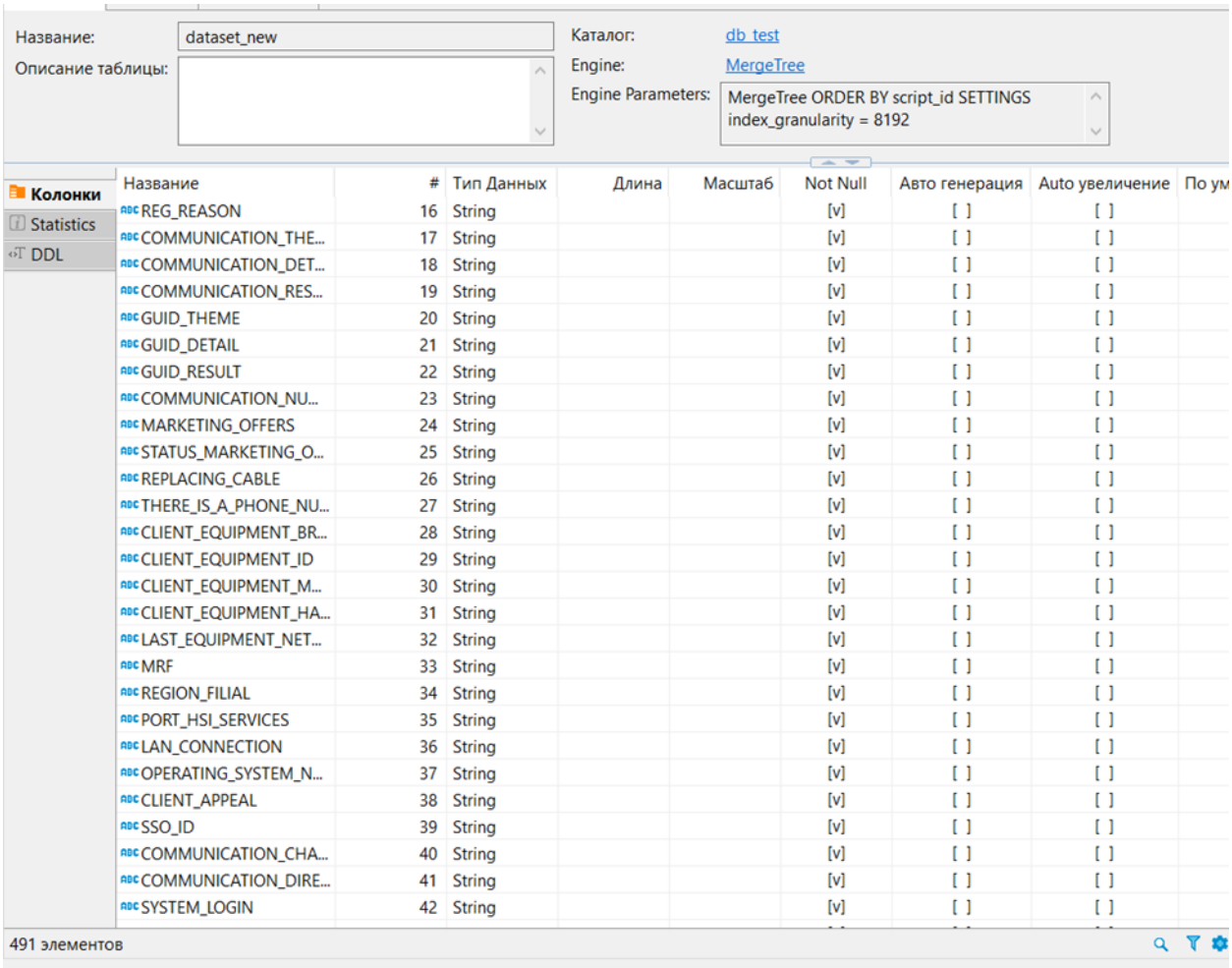


Таблица создана и пригодна к использованию для специалистов по BI. Перейдем к следующей задачке, где запустим скрипт обработки через Airflow.