In [ ]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-pytho
n
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files
under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserve
d as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of
the current session
```

In [ ]:

```
from google.cloud import bigquery
```

In [5]:

```
client=bigquery.Client()
```

```
Using Kaggle's public dataset BigQuery integration.
```

In [6]:

```
dataset_ref = client.dataset("hacker_news", project ="bigquery-public-data")
dataset = client.get_dataset(dataset_ref)
```

In [10]:

```
tables = list(client.list_tables(dataset))
for table in tables:
    print(table.table_id)
```

```
full
```

In [15]:

```
table_ref = dataset_ref.table("full")
table = client.get_table(table_ref)
```

In [16]:

```
table.schema
```

Out[16]:

```
[SchemaField('title', 'STRING', 'NULLABLE', None, 'Story title', (), None),
 SchemaField('url', 'STRING', 'NULLABLE', None, 'Story url', (), None),
 SchemaField('text', 'STRING', 'NULLABLE', None, 'Story or comment text', (), None),
 SchemaField('dead', 'BOOLEAN', 'NULLABLE', None, 'Is dead?', (), None),
 SchemaField('by', 'STRING', 'NULLABLE', None, "The username of the item's author.", (),
None),
 SchemaField('score', 'INTEGER', 'NULLABLE', None, 'Story score', (), None),
 SchemaField('time', 'INTEGER', 'NULLABLE', None, 'Unix time', (), None),
 SchemaField('timestamp', 'TIMESTAMP', 'NULLABLE', None, 'Timestamp for the unix time', (
), None),
 SchemaField('type', 'STRING', 'NULLABLE', None, 'type of details (comment comment rankin
g poll story job pollopt)', (), None),
 SchemaField('id', 'INTEGER', 'NULLABLE', None, "The item's unique id.", (), None),
 SchemaField('parent', 'INTEGER', 'NULLABLE', None, 'Parent comment ID', (), None),
 SchemaField('descendants', 'INTEGER', 'NULLABLE', None, 'Number of story or poll descend
ants', (), None),
 SchemaField('ranking', 'INTEGER', 'NULLABLE', None, 'Comment ranking', (), None),
 SchemaField('deleted', 'BOOLEAN', 'NULLABLE', None, 'Is deleted?', (), None)]
```

In [17]:

```
client.list_rows(table, max_results=5).to_dataframe()
```

Out[17]:

	title	url	text	dead	by	score	time	timestamp	type	id	parent	descendants	rank
0	None	None	None	True	Adoum_Tech	2	1713995025	2024-04-24 21:43:45+00:00	story	40150086	<NA>	<NA>	<N
1	None	None	None	True	belter	2	1713995286	2024-04-24 21:48:06+00:00	story	40150135	<NA>	<NA>	<N
2	None	None	None	True	Rinzler89	1	1713995678	2024-04-24 21:54:38+00:00	story	40150207	<NA>	<NA>	<N
3	None	None	None	True	stockstobuynow	1	1713995704	2024-04-24 21:55:04+00:00	story	40150212	<NA>	<NA>	<N
4	None	None	None	True	FLMAN407	1	1713995772	2024-04-24 21:56:12+00:00	story	40150229	<NA>	<NA>	<N

In [22]:

```
client.list_rows(table,selected_fields=table.schema[:2], max_results=5).to_dataframe()
```

Out[22]:

	title	url
0	None	None
1	None	None
2	None	None
3	None	None
4	None	None

In [37]:

```
query= """
SELECT `by` from `bigquery-public-data.hacker_news.full` WHERE score = 2
"""
```

In [39]:

```
client=bigquery.Client()
```

```
Using Kaggle's public dataset BigQuery integration.
```

In [40]:

```
query_job = client.query(query)
```

In [41]:

```
score_movies = query_job.to_dataframe()
```

```
/usr/local/lib/python3.11/dist-packages/google/cloud/bigquery/table.py:1727: UserWarning:
BigQuery Storage module not found, fetch data with the REST endpoint instead.
  warnings.warn(
```

In [48]:

```
score_movies.by.value_counts().head(100)
```

Out[48]:

```
by
rbanffy        9429
Tomte          6895
tosh           6242
bookofjoe      4741
pseudolus      4688
               ... 
lelf            791
ksec            785
clouddrover     773
imartin2k       769
pjmlp           765
Name: count, Length: 100, dtype: int64
```

In [67]:

```
query= """

        SELECT `by`,id
        FROM `bigquery-public-data.hacker_news.full`
        Where CAST(id AS STRING) LIKE '%502%'
        """
```

In [62]:

```
query_job=client.query(query)
```

In [63]:

```
id_by = query_job.to_dataframe()
```

```
/usr/local/lib/python3.11/dist-packages/google/cloud/bigquery/table.py:1727: UserWarning:
BigQuery Storage module not found, fetch data with the REST endpoint instead.
  warnings.warn(
```

In [68]:

```
id_by.head(10)
```

Out[68]:

	by	id
0	Kagetora85	29218502
1	Yaxin	29225023
2	mpelembe	29225024
3	AyanaHod	29250209
4	Nancydrew23	29250234
5	fspacef	29250251
6	josepas10	29250254
7	Tomte	29250266
8	bartoszgorka	29250277
9	exavir	29255022

In [122]:

```
query_CTE = """

        SELECT parent,COUNT(id) As NumInteract
        FROM `bigquery-public-data.hacker_news.full`
        WHERE parent IS NOT NULL
        GROUP BY parent
        HAVING Count(id)>10
        Order BY NumInteract DESC
        """
```

In [123]:

```
client=bigquery.Client()
```

```
Using Kaggle's public dataset BigQuery integration.
```

In [121]:

```
safe_config = bigquery.QueryJobConfig(maximum_bytes_billed=10**10)

query_job = client.query(query, job_config=safe_config)

pop_comments = query_job.to_dataframe()
pop_comments.head(10)
```

```
/usr/local/lib/python3.11/dist-packages/google/cloud/bigquery/table.py:1727: UserWarning:
BigQuery Storage module not found, fetch data with the REST endpoint instead.
  warnings.warn(
```

Out[121]:

	parent	NumInteract
0	36575081	1810
1	363	1316
2	43446941	1129
3	23170881	1105
4	30934529	1051
5	27355392	1041
6	16967543	1033
7	29067493	1011
8	13541679	1008
9	25989764	994

In [147]:

```
query_CTE = """
           WITH days AS
           (
               SELECT EXTRACT(day from timestamp) as day
               FROM `bigquery-public-data.hacker_news.full`
           )
           SELECT COUNT(1) AS day_oc, day
           FROM days
           WHERE day IS NOT NULL
           GROUP BY day
           ORDER BY day

        """
```

In [148]:

```
from google.cloud import bigquery

safe_config = bigquery.QueryJobConfig(maximum_bytes_billed=10**10)

query_job = client.query(query_CTE, job_config=safe_config)
```

```
pop_comments = query_job.to_dataframe()

pop_comments.head(10)
```

```
/usr/local/lib/python3.11/dist-packages/google/cloud/bigquery/table.py:1727: UserWarning:
BigQuery Storage module not found, fetch data with the REST endpoint instead.
  warnings.warn(
```

Out[148]:

	day_oc	day
0	1454138	1
1	1442226	2
2	1447814	3
3	1438648	4
4	1443126	5
5	1439275	6
6	1448592	7
7	1437976	8
8	1444164	9
9	1453140	10

In [149]:

```
from google.cloud import bigquery

# Create a "Client" object
client = bigquery.Client()

# Construct a reference to the "github_repos" dataset
dataset_ref = client.dataset("github_repos", project="bigquery-public-data")

# API request - fetch the dataset
dataset = client.get_dataset(dataset_ref)

# Construct a reference to the "licenses" table
licenses_ref = dataset_ref.table("licenses")

# API request - fetch the table
licenses_table = client.get_table(licenses_ref)

# Preview the first five lines of the "licenses" table
client.list_rows(licenses_table, max_results=5).to_dataframe()
```

```
Using Kaggle's public dataset BigQuery integration.
```

Out[149]:

	repo_name	license
0	autarch/Dist-Zilla-Plugin-Test-TidyAll	artistic-2.0
1	thundergnat/Prime-Factor	artistic-2.0
2	kusha-b-k/Turabian_Engin_Fan	artistic-2.0
3	onlinepremiumoutlet/onlinepremiumoutlet.github.io	artistic-2.0
4	huangyuanlove/LiaoBa_Service	artistic-2.0

In [150]:

```
files_ref = dataset_ref.table("sample_files")

# API request - fetch the table
files_table = client.get_table(files_ref)
```

```
# Preview the first five lines of the "sample_files" table
client.list_rows(files_table, max_results=5).to_dataframe()
```

Out[150]:

	repo_name	ref	path	mode	
0	EOL/eol	refs/heads/master	generate/vendor/railties	40960	0338c33fb3fda57db9e812ac7de969
1	np/ling	refs/heads/master	tests/success/merger_seq_inferred.t/merger_seq...	40960	dd4bb3d5ecabe5044d3fa5a36e0a9b
2	np/ling	refs/heads/master	fixtures/sequence/lettype.ll	40960	8fdf536def2633116d65b92b3b9257
3	np/ling	refs/heads/master	fixtures/failure/wrong_order_seq3.ll	40960	c2509ae1196c4bb79d7e60a3d67948
4	np/ling	refs/heads/master	issues/sequence/keep.t	40960	5721de3488fb32745dfc11ec482e5

In [152]:

```
query_join = """

            SELECT L.license, COUNT(*) AS Number_of_files
            FROM `bigquery-public-data.github_repos.sample_files` AS sf
            INNER JOIN `bigquery-public-data.github_repos.licenses` AS L

                ON sf.repo_name = L.repo_name
                Group BY L.license
                ORDER BY number_of_files DESC
            """
#from google.cloud import bigquery

safe_config = bigquery.QueryJobConfig(maximum_bytes_billed=10**10)

query_job = client.query(query_join, job_config=safe_config)

pop_comments = query_job.to_dataframe()

pop_comments.head(15)
```

```
/usr/local/lib/python3.11/dist-packages/google/cloud/bigquery/table.py:1727: UserWarning:
BigQuery Storage module not found, fetch data with the REST endpoint instead.
  warnings.warn(
```

Out[152]:

	license	Number_of_files
0	mit	20560894
1	gpl-2.0	16608922
2	apache-2.0	7201141
3	gpl-3.0	5107676
4	bsd-3-clause	3465437
5	agpl-3.0	1372100
6	lgpl-2.1	799664
7	bsd-2-clause	692357
8	lgpl-3.0	582277
9	mpl-2.0	457000
10	cc0-1.0	449149
11	epl-1.0	322255
12	unlicense	208602
13	artistic-2.0	147391
14	isc	118332

In [ ]: