SELECT block_timestamp, value FROM bigquery-public-data.crypto_ethereum.transactions AS transactions WHERE block_timestamp < '2022-04-01 00:00:00'

Here the 'value' attribute has to be converted to understandable form from wei to the default ether_value before the data to be extracted. One ether is 10 ^ 18 weis, i.e 1 eth = wei(10^18).

SELECT block_timestamp, (SUM(value) / power(10,18)) AS total_value FROM bigquery-public-data.crypto_ethereum.transactions AS transactions WHERE block_timestamp < '2022-04-01 00:00:00'

The columns date, ether value can be used to convert the table into a dataframe, ordering query using the recent dates.

The 'block_timestamp' has to be converted into Date format in order group by date to get the ether_value per day data

SELECT DATE(block_timestamp) AS date, (SUM(value) / power(10,18)) AS total_value FROM bigquery-public-data.crypto_ethereum.transactions AS transactions WHERE block_timestamp < '2022-04-01 00:00:00' GROUP BY date ORDER BY date

The resulting query results in processing 80 GB of data out of which only a sample of 1GB can be saved as CSV which is set in sharing mode to access as URL

```
#trail using google drive shared csv file (about 1 GB) instead of local
csv file(about 10 MB) for the query.
import pandas
URL = 'https://drive.google.com/file/d/19jXVha2A7AtgZRGI6-
HXKCcodVe8N0bf/view?usp=sharing'
path =
'https://drive.google.com/uc?export=download&id='+URL.split('/')[-2]
df = pandas.read_csv(path)
df.head()
```

	date	total_value
0	2023-03-31	1.781238e+06
1	2023-03-30	1.090853e+06
2	2023-03-29	1.339400e+06
3	2023-03-28	1.313519e+06
4	2023-03-27	1.473780e+06

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
fig, axs = plt.subplots(figsize=(20, 6))
axs.yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter('{x:,.0f}')
)
plt.title("Value of Ether transferred per day", y=1.05)
```

```
plt.xlabel("Date", labelpad=20)
plt.ylabel("Ether transferred ", labelpad=20)
sns.lineplot(x="date", y="total_value", data=df, color="blue")
plt.gcf().set_dpi(400)
plt.show(fig)
```