

Project 2 - Earthquakes Analysis



By Hari, Sana, Chad, Bao, Summer & Sam

Theme - Earthquake

The public platform **USGS** lead us to "Earthquake Hazards Program". They provide Earthquake related data in several ways such as **Real-time Feeds, ATOM, KML, Spreadsheet, QuakeML, GeoJSON etc.,**

We decided to pull data from 2 different ways -

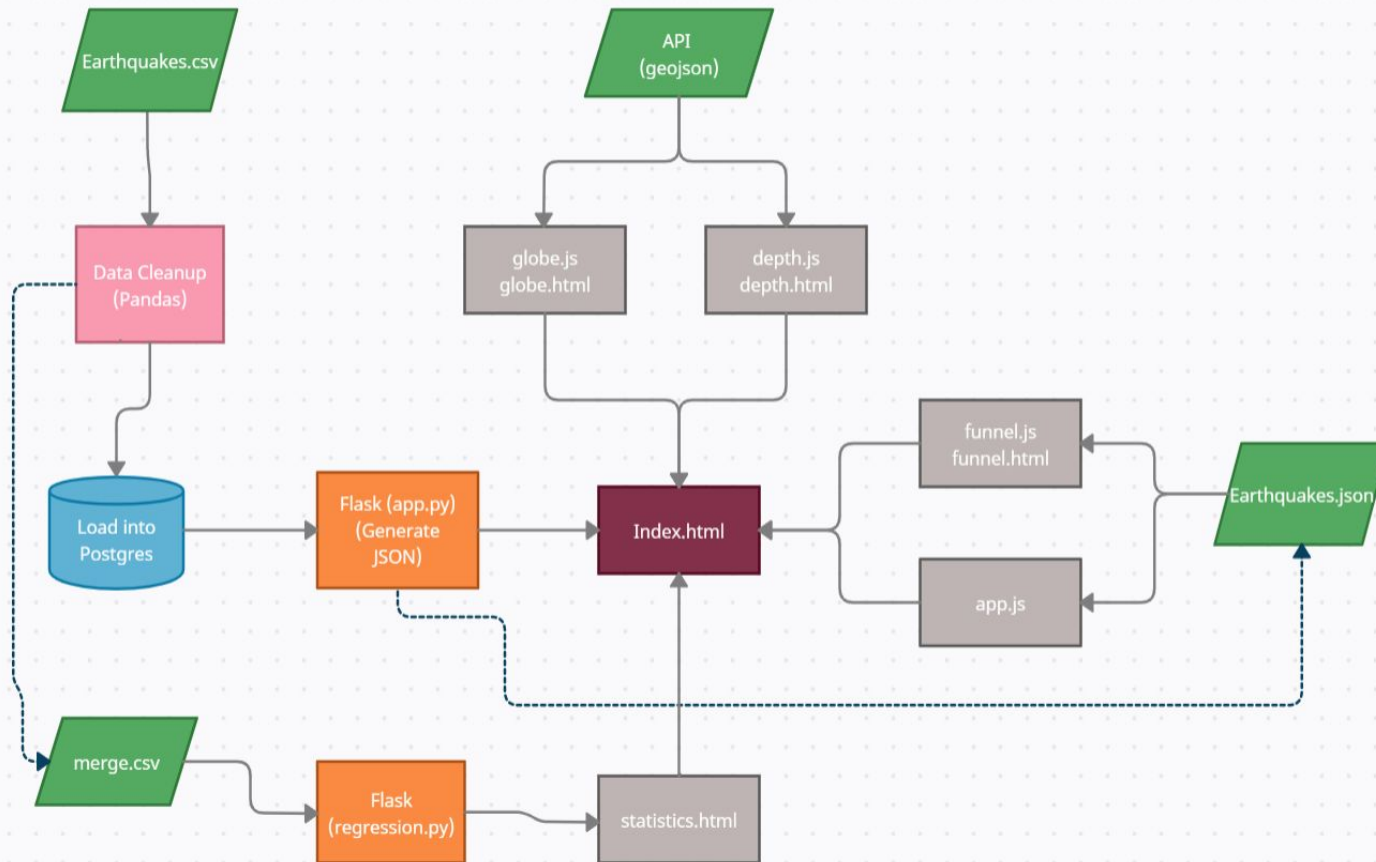
1] Spreadsheet - Past 30 days (03/18 - 04/17) of Earthquakes data across the world

<https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php>

2] GeoJSON - Past 30 days (LIVE Feed) of Earthquakes data across the world

<https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php>

Coding Approach



Technologies Used:

- 1] Pandas
- 2] ReverseGeocoder
- 3] SQLAlchemy
- 4] Postgres DB
- 5] Flask
- 6] HTML
- 7] Bootstrap
- 8] Javascript
- 9] D3
- 10] Plotly js
- 11] Apex js
- 12] am4Charts js

Data Munging

- Used "Reverse_Geocoder" Python library to get the nearest City, County, State and Country based on Lat and Long.
- Fixed the timestamp format by using Pandas

```
# sample test
coordinates = (32.751, -115.8293333), (60.2994, -141.2456)
rg.search(coordinates)
```

Loading formatted geocoded file...

```
[8]: [{'lat': '32.79311',
      'lon': '-115.69111',
      'name': 'Seeley',
      'admin1': 'California',
      'admin2': 'Imperial County',
      'cc': 'US'},
      {'lat': '63.33667',
      'lon': '-142.98556',
      'name': 'Tok',
      'admin1': 'Alaska',
      'admin2': 'Southeast Fairbanks Census Area',
      'cc': 'US'}]
```

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place
0	2021-04-17T22:20:57.320Z	32.751000	-115.829333	6.040000	1.24	ml	17.0	72.0	0.1296	0.22	...	2021-04-17T22:24:25.471Z	15km E of Ocotillo, CA
1	2021-04-17T22:18:44.955Z	60.299400	-141.245600	3.000000	1.50	ml	NaN	NaN	NaN	0.59	...	2021-04-17T22:22:37.941Z	119 km NW of Yakutat, Alaska
2	2021-04-17T22:13:31.910Z	19.182333	-155.393997	34.349998	1.85	md	38.0	166.0	NaN	0.12	...	2021-04-17T22:16:49.020Z	9 km ESE of Pāhala, Hawaii

	time	latitude	longitude	depth	mag	magType	rms	id	updated	place	type
0	2021-04-17 22:20:58	32.751000	-115.829333	6.040000	1.24	ml	0.22	ci39854328	2021-04-17 22:24:26	15km E of Ocotillo, CA	earthquake
1	2021-04-17 22:18:45	60.299400	-141.245600	3.000000	1.50	ml	0.59	ak0214xbpmk2	2021-04-17 22:22:38	119 km NW of Yakutat, Alaska	earthquake
2	2021-04-17 22:13:32	19.182333	-155.393997	34.349998	1.85	md	0.12	hv72430407	2021-04-17 22:16:50	9 km ESE of Pāhala, Hawaii	earthquake

Data Munging

- Derived few summary fields to get the Earthquake stats for each Country. We used this for populating "Earthquake Info" table on Main webpage based on Country selection

```
stats_grp = merge_df.groupby('country').agg(  
    mindepth=('depth', "min"),  
    maxdepth=('depth', "max"),  
    minmag=('mag', "min"),  
    maxmag=('mag', "max"),  
    count=('id', "count"),  
    mindate=('date', "min"),  
    maxdate=('date', "max")  
)  
  
stats_grp.head()
```

26]:

	mindepth	maxdepth	minmag	maxmag	count	mindate	maxdate
country							
AL	6.03	10.00	2.9	4.3	3	2021-04-05	2021-04-13
AR	108.99	245.61	4.5	4.6	2	2021-03-22	2021-04-11
AT	7.02	7.02	4.0	4.0	1	2021-03-30	2021-03-30
AU	10.00	10.00	4.2	6.0	13	2021-03-20	2021-04-06
BR	10.00	558.32	4.1	5.0	3	2021-04-04	2021-04-11

Country Selection:

NZ ▼

Earthquake Info

mindepth: 10

maxdepth: 288.79

minmag: 2.7

maxmag: 6

count: 20

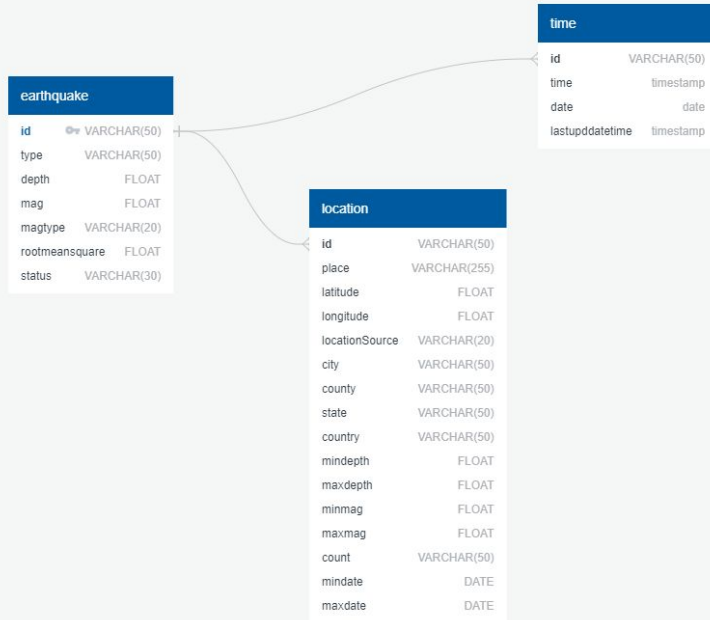
mindate: 2021-03-19

maxdate: 2021-04-16





Data Load

We created total 3 different tables with "id" being Primary and Foreign key to form the relationship and uniquely identify records in each table

www.quickdatabasediagrams.com



```
1 select * from earthquake;
```

Data Output		Explain	Messages	Notifications			
	 id [PK] character varying (50)	 type character varying (50)	 depth double precision	 mag double precision	 magtype character varying (20)	 rootmeansquare double precision	 status character varying (30)
1	us7000dl5l	earthquake		10	4.9 mb		0.57 reviewed
2	us7000dl5j	earthquake		10	6 mwb		0.54 reviewed
3	us6000dxkg	earthquake		10	4.9 mb		0.52 reviewed

```
1 select * from location;
```

Data Output										Explain	Messages	Notifications
	 id character varying (50)	 place character varying (255)	 latitude double precision	 longitude double precision	 locationSource character varying (20)	 city character varying (50)	 county character varying (50)	 state character varying (50)	 country character			
1	us7000dl5l	west of Macquarie Island	-59.7161	150.46	us	Kettering	Kingborough	Tasmania	AU			
2	us7000dl5j	west of Macquarie Island	-59.6179	150.3277	us	Kettering	Kingborough	Tasmania	AU			
3	us6000dxkg	west of Macquarie Island	-53.7784	140.5419	us	Geeveston	Huon Valley	Tasmania	AU			

```
1 select * from time;
```

Data Output	Explain	Messages	Notifications	
	id character varying (50)	time timestamp without time zone	date date	lastupdatetime timestamp without time zone
1	us7000dl5l	2021-03-20 05:17:49	2021-03...	2021-04-02 14:16:52
2	us7000dl5j	2021-03-20 05:19:32	2021-03...	2021-04-02 15:04:27
3	us6000dxkg	2021-03-30 09:24:19	2021-03...	2021-03-30 09:40:19

Generate JSON from Postgres

We used `json.dumps` to generate JSON file from Postgres database inside Flask

```
# Convert query to objects of key-value pairs
```

```
objects_list = []
```

```
for row in rows:
```

```
    d = collections.OrderedDict()
```

```
    d["id"] = row[0]
```

```
    d["type"] = row[1]
```

```
    d["depth"] = row[2]
```

```
    d["mag"] = row[3]
```

```
    d["magtype"] = row[4]
```

```
    d["rootmeansquare"] = row[5]
```

```
    d["status"] = row[6]
```

```
    d["latitude"] = row[7]
```

```
    d["longitude"] = row[8]
```

```
    d["city"] = row[9]
```

```
    d["county"] = row[10]
```

```
    d["state"] = row[11]
```

```
    d["country"] = row[12]
```

```
    d["mindepth"] = row[13]
```

```
    d["maxdepth"] = row[14]
```

```
    d["minmag"] = row[15]
```

```
    d["maxmag"] = row[16]
```

```
    d["count"] = row[17]
```

```
    d["mindate"] = str(row[18])
```

```
    d["maxdate"] = str(row[19])
```

```
    objects_list.append(d)
```

```
j = json.dumps(objects_list)
```

```
with open("Resources/Earthquakes.json", "w") as f:
```

```
    f.write(j)
```

```
{ Earthquakes.json X
```

```
Resources > { Earthquakes.json > {} 9
```

```
1 [{"id": "us7000dl5l", "type": "earthquake", "depth": 10.0, "mag": 4.9, "magtype": "mb", "rootmeansquare": 0.57, "status": "reviewed",  
  "latitude": -59.7161, "longitude": 150.46, "city": "Kettering", "county": "Kingborough", "state": "Tasmania", "country": "AU",  
  "mindepth": 10.0, "maxdepth": 10.0, "minmag": 4.2, "maxmag": 6.0, "count": "13", "mindate": "2021-03-20", "maxdate": "2021-04-06"},  
  {"id": "us7000dl5j", "type": "earthquake", "depth": 10.0, "mag": 6.0, "magtype": "mwb", "rootmeansquare": 0.54, "status": "reviewed",  
  "latitude": -59.6179, "longitude": 150.3277, "city": "Kettering", "county": "Kingborough", "state": "Tasmania", "country": "AU",  
  "mindepth": 10.0, "maxdepth": 10.0, "minmag": 4.2, "maxmag": 6.0, "count": "13", "mindate": "2021-03-20", "maxdate": "2021-04-06"},  
  {"id": "us6000dxkg", "type": "earthquake", "depth": 10.0, "mag": 4.9, "magtype": "mb", "rootmeansquare": 0.52, "status": "reviewed",  
  "latitude": -53.7784, "longitude": 140.5419, "city": "Geeveston", "county": "Huon Valley", "state": "Tasmania", "country": "AU",  
  "mindepth": 10.0, "maxdepth": 10.0, "minmag": 4.2, "maxmag": 6.0, "count": "13", "mindate": "2021-03-20", "maxdate": "2021-04-06"}]
```