Data Analytics

Global Ocean Trends: Warming, Pollution, and Coral Bleaching Analysis

Smita PRAKAS

April, 2024

Table of content

Introduction

Business Use Case

Covering three-quarters of the planet's surface and holding 97% of its water, the ocean is vital to life on Earth. However, ocean warming, largely driven by climate change, is accelerating. Combined with marine plastic pollution, it poses a significant threat to coral reefs, which are crucial ecosystems, supporting countless marine species. By raising awareness and providing valuable information on global ocean trends, we can guide conservation and protection efforts for these vital ecosystems.

Goal

The goal of my project is to:

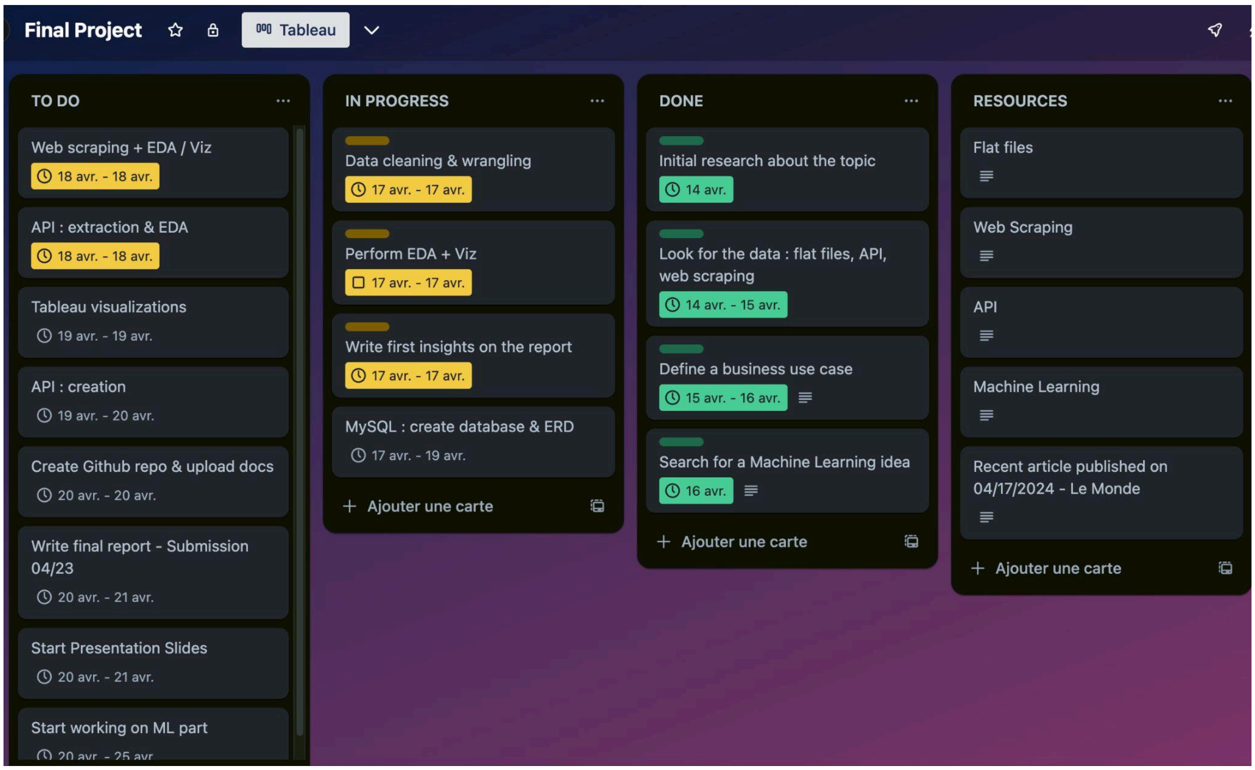
- analyze trends in marine plastic pollution over time and across different countries
- assess the influence of sea surface temperature, and rise in temperature, on coral bleaching incidents
- Identify additional factors that may contribute to coral bleaching events based on available data

High-level plan

- Research about project topic
- Data collection
- Project scope
- Project planning in Trello
- Exploratory data analysis in Python (data wrangling, data cleaning & visualization)
- Selection and creation of a database using MySQL
- Adding data to database and create Entity-Relationship Diagram
- Data manipulation in SQL
- Exposing data via API
- Visualization insights in Tableau
- Machine learning

Project Management

Overview of my Trello Board : structure & management of daily tasks



Data collection and sources

1. Flat files

● The first dataset called ‘Coral bleaching events’ in .csv format was found on https://ourworldindata.org/grapher/coral-bleaching-events website. From this source, I generated 1 cleaned datarame:

➔ ‘coral-bleaching-events-per-year.csv’
(initial shape: 185 rows, 5 columns)

● I then founded a bigger dataset on https://www.bco-dmo.org/dataset/773466 which is itself a collection from several different valuable sources. It contains bleaching data (presence or absence of bleaching incident) and environmental data (such as site exposure,distance to shore, mean turbidity, cyclone frequency and sea-surface temperature metrics) for global coral reef sites from 1980 to 2020.
From this second source, 1 more cleaned dataframe was created:

➔ ‘global_bleaching_env.csv’
(initial shape: 41 361 rows, 62 columns)

● I decided to collect some additional datasets in .csv format from the website https://ourworldindata.org/plastic-pollution, essentially to add some more context to my analysis and problem statement.
From this third source, I generated 4 dataframes:

➔ ‘global_plastic_production.csv’
(initial shape: 69 rows, 4 columns)
➔ ‘share_global_plastics_to_oceans_by_continent.csv’
(initial shape: 170 rows, 4 columns)
➔ ‘plastic_waste_into_ocean_by_country.csv’
(initial shape: 171 rows, 4 columns)
➔ 'decomposition_rates_marine_debris.csv'
(initial shape: 13 rows, 4 columns)

2. API

● I decided to extract information from the ‘**United Nations Statistics Division SGD API**’. The United Nations have defined 17 Goals around Sustainable Development : one of them, **Goal 14** is about “conserving and sustainably using the oceans, seas and marine resources.”

With this API, I accessed information on **target 14.3**, related to marine acidity (pH) between 1996-2022:

> **14.3** Minimize and address the impacts of ocean acidification, including through enhanced scientific cooperation at all levels

From this source, I created an additional .csv format file for further visualizations:

➔'avg_marine_ph.csv'

(initial shape: 1 291 rows, 21 columns)

3. Web Scrapping

I struggled finding a site to scrape because I noticed that most of the data available around my topic was summarized either in a PDF format report with infographics or on static images or maps.

After further reflection and since I already had gathered enough data for my analysis, I decided to scrape 200 articles on 'climate change' and 'ocean' related topics from https://www.foxnews.com/category/science/planet-earth/oceans media website.

The main purpose was to give an overview and some extra resources with url links to the most recent articles published, with the aim of raising awareness around these two subjects.

I first tried to use the Beautiful Soup library, but it was not sufficient on its own because the web page had a 'Show more' button to load content dynamically. Therefore, I chose to use Selenium along with Beautiful Soup, as it enables you to programmatically control a web browser, including for example clicking buttons, filling out forms, and scrolling down a page.

Here, you can find the dataframe with the most recent articles I scraped, sorted by category : ➔'articles_library.csv'

The third most recent article published on **2024-04-16** is titled *'Coral reefs around the world are experiencing massive bleaching'*:

	category	date	title	description	article_url	image_link
0	oceans	2024-04-17	Oklahoma boy's pet octopus is TikTok sensation...	An Oklahoma boy with a precocious passion for ...	/lifestyle/oklahoma-boys-pet-octopus-tiktok-se...	https://a57.foxnews.com/static.foxnews.com/fox...
1	oceans	2024-04-16	Greece proposes 2 marine parks as part of $830...	Greece has proposed a plan to create two large...	/world/greece-proposes-2-marine-parks-830m-env...	https://a57.foxnews.com/static.foxnews.com/fox...
2	oceans	2024-04-16	Coral reefs around the world are experiencing ...	Coral reefs around the world are experiencing ...	/science/coral-reefs-world-experiencing-mass-b...	https://a57.foxnews.com/static.foxnews.com/fox...
3	oceans	2024-04-13	Florida girl, 12, hooks multiple fishing recor...	Julia Bernstein, 12, set at least four differe...	/lifestyle/florida-girl-12-hooks-multiple-fish...	https://a57.foxnews.com/static.foxnews.com/fox...
4	oceans	2024-04-04	Georgia group and others release 34 rehabilita...	Organizations worked to re-release 34 sea turt...	/lifestyle/georgia-group-others-release-34-reh...	https://a57.foxnews.com/static.foxnews.com/fox...
...	...	...	...	...	...	...
195	climate change	2023-10-25	Sen Risch to introduce bill giving states powe...	Sen. Jim Risch (R-Idaho) is introducing the Do...	/politics/sen-risch-introduce-bill-giving-stat...	https://a57.foxnews.com/static.foxnews.com/fox...
196	climate change	2023-10-18	The latest attempt to take away your gas-power...	Another government agency has proposed a rule ...	/opinion/latest-attempt-take-away-your-gas-pow...	https://a57.foxnews.com/static.foxnews.com/fox...

Data cleaning and Exploratory data analysis

The below listed dataframes (already mentioned above too) were created after conducting data cleaning and Exploratory Data Analysis (EDA):

- ➔ 'coral-bleaching-events-per-year.csv'
- ➔ 'global_bleaching_env.csv'
- ➔ 'global_plastic_production.csv'
- ➔ 'share_global_plastics_to_oceans_by_continent.csv'
- ➔ 'plastic_waste_into_ocean_by_country.csv'
- ➔ 'decomposition_rates_marine_debris.csv'
- ➔ 'avg_marine_ph.csv'
- ➔ 'articles_library.csv'

In order to save time and effort in the long run and because I had many datasets to clean, I wrote functions to re-use them for each of my dataset.

Overall, a similar set of steps was applied for the cleaning process of each dataset.

Here is the cleaning process of the dataset 'global_bleaching_environmental.csv' :

- ❖ Exploration of the dataset (shape, data types):

```
Shape of the dataset: (41361, 62)
```

```
df2.dtypes
✓ 0.0s

Site_ID                  int64
Sample_ID                int64
Data_Source             object
Latitude_Degrees       float64
Longitude_Degrees      float64
                         ...
TSA_DHWMean             object
Date                    object
Site_Comments           object
Sample_Comments         object
Bleaching_Comments      object
Length: 62, dtype: object
```

❖ Checking for duplicated values :

```
Duplicated values per column:

False
```

❖ Checking for missing values :

```
Missing values per column:

Site_ID               0
Sample_ID             0
Data_Source           0
Latitude_Degrees      0
Longitude_Degrees     0
                     ..
TSA_DHWMean           0
Date                  0
Site_Comments         0
Sample_Comments       0
Bleaching_Comments    0
Length: 62, dtype: int64
```

❖ Handling missing values :

```
# As we can see, there are no missing values, however, the string "nd" is used instead
# Let's convert it to 'NaN' so we can deal easily with actual missing values

df2.replace('nd', np.nan, inplace=True)
```
✓ 0.0s

Now we can see below, the total of missing values per column:

```
# Let's check again for missing values
df2.isnull().sum()
```
✓ 0.0s

```
site_id                0
sample_id              0
data_source            0
latitude_degrees       0
longitude_degrees      0
                     ...
tsa_dhwmean          132
date                   0
site_comments      39104
sample_comments    38403
bleaching_comments 38692
Length: 62, dtype: int64
```

```
# For each column that contains missing values, let's check the percentage using the above function that we've written

missing_df = missing_percentage(df2)
missing_df.sort_values(by='Percentage', ascending=False).head(20)
```
✓ 0.0s

	Column_Name	Count	Percentage
59	site_comments	39104	94.54
61	bleaching_comments	38692	93.55
60	sample_comments	38403	92.85
12	site_name	34429	83.24
23	bleaching_level	18830	45.53
21	substrate_name	12668	30.63
6	reef_id	12540	30.32
22	percent_cover	12455	30.11
24	percent_bleaching	6846	16.55
20	depth_m	1799	4.35
11	city_town_name	1133	2.74
35	ssta_minimum	176	0.43
45	tsa	148	0.36
26	temperature_kelvin	148	0.36

I started by dropping columns with more than 80% missing values and those with not enough relevant information for further analysis :

```
# Let's start by dropping columns that have more than 80% missing values and those that do not have any relevant information for further analysis
columns_to_drop_df2 = [
    'data_source', 'reef_id', 'date_day', 'date_month','depth_m',
    'bleaching_level', 'temperature_kelvin_standard_deviation','site_name',
    'ssta_standard_deviation', 'ssta_mean', 'ssta_minimum', 'ssta_maximum',
    'ssta_frequency', 'ssta_frequency_standard_deviation', 'ssta_frequencymax',
    'ssta_frequencymean', 'ssta_dhw', 'ssta_dhw_standard_deviation', 'ssta_dhwmax',
    'ssta_dhwmean', 'tsa_standard_deviation', 'tsa_minimum', 'tsa_maximum',
    'tsa_mean', 'tsa_frequency', 'tsa_frequency_standard_deviation', 'tsa_frequencymax',
    'tsa_frequencymean', 'tsa_dhw', 'tsa_dhw_standard_deviation', 'tsa_dhwmax',
    'tsa_dhwmean', 'site_comments', 'sample_comments', 'bleaching_comments', 'tsa'
]

# Drop the specified columns from the DataFrame
df2.drop(columns=columns_to_drop_df2, axis=1, inplace=True)

✓ 0.0s                                                                      Pyth
```

I also decided to drop the columns with less or equal to 35% missing values:

```
# Let's also handle missing values for the columns that have less or equals to 35% missing values
rows_to_drop = missing_df2[missing_df2['Percentage'] <= 35]['Column_Name'].tolist()
rows_to_drop

df2.dropna(subset=rows_to_drop, inplace=True)

✓ 0.0s
```

```
# Let's check if we still have any remaining missing values
df2.isnull().any().sum()
✓ 0.0s
```

- Cleaning column names:

```
# Let's clean some of the column names
df2.rename(columns={
    'date_year': 'year',
    'ocean_name': 'ocean',
    'realm_name': 'realm',
    'ecoregion_name': 'ecoregion',
    'country_name' : 'country',
    'state_island_province_name': 'state_island_province',
    'city_town_name': 'city_town',
    'temperature_minimum': 'temperature_min',
    'temperature_maximum': 'temperature_max'
    }, inplace=True)
✓ 0.0s
```

❖ Converting numerical columns to 'float' or 'int' type:

```python
# Let's convert numerical columns to 'float' or 'int'

df2[['percent_cover'
     ,'percent_bleaching'
     ,'temperature_kelvin'
     ,'temperature_min'
     ,'temperature_max'
     ,'ssta'
     ,'windspeed']] = df2[['percent_cover'
                          , 'percent_bleaching'
                          , 'temperature_kelvin'
                          , 'temperature_min'
                          , 'temperature_max'
                          , 'ssta'
                          , 'windspeed']].apply(pd.to_numeric, errors='coerce')
```

✓ 0.0s

After the cleaning process, I added a new column 'bleaching_status' for visualizations purpose:

```python
# Let's add a new 'bleaching_status' column to our DataFrame for visualizations purpose
def bleaching_status(percent):
    if percent == 0:
        return 'Unbleached'
    elif percent <= 30:
        return 'Moderate'
    else:
        return 'Severe'

df2['bleaching_status'] = df2['percent_bleaching'].apply(bleaching_status)
```

✓ 0.0s

❖ Shape of the dataframe after cleaning was done:

```python
df2.shape
```

✓ 0.0s

```
(21836, 26)
```

❖ Converting the cleaned dataframe to .csv format:

```
df2.to_csv('global_bleaching_env.csv', index=False)
✓ 0.2s
```

Here is a summary of the shape and column names/metadata after each dataframe has been cleaned:

Flat Files

global_bleaching_env.csv' 21 836 rows x 26 columns	
site_id	Unique identifier for each site
sample_id	Unique identifier for each sampling event
latitude_degrees	Latitude coordinates (positive vaues = North; negative values = South)
longitude_degrees	Longitude coordinates (positive values = East; negative values = West)
ocean	The ocean in which the sampling took place
realm	Identification of realm as defined by the Marine Ecoregions of the World (MEOW) Spalding et al. 2007
ecoregion	Identification of the Ecoregions (150) as defined by Veron et al
country	The country where sampling took place
state_island_province	The state, territory (e.g., Guam) or island group (e.g., Hawaiian Islands) where sampling took place
city_town	The region, city, or nearest town, where sampling took place
distance_to_shore	The distance of the sampling site from the nearest land
exposure	The site's exposure to fetch. Site was considered exposed if it had >20 km of fetch, if there were strong seasonal winds, or if the site faced the prevailing winds. Otherwise, the site was considered sheltered or "sometimes". "Sometimes" refers to a few sites with a >20 km fetch through a narrow geographic window, and therefore we considered that the site was potentially exposed during cyclone seasons.
turbidity	Kd490 with a 100-km buffer. Turbidity was considered to be positively related to the diffuse attenuation coefficient of light at the 490 nm wavelength (Kd490), or the rate at which light at 490 nm is attenuated with depth. For example, a Kd490 value of 0.1 m?1 means that light intensity is reduced by one natural-log value within 10 m of water. High values of Kd490, therefore, represent high attenuation and hence high turbidity.

cyclone_frequency	number of cyclone events from 1964 to 2014
year	the year of sampling event
substrate_name	type of substrate from Reef Check data
percent_cover	average cover value (percent)
percent_bleaching	An average of four transect segments (Reef Check) or average of a bleaching code
climsst	Climatological sea surface temperature (SST) based on weekly SSTs for the study time frame, created using a harmonics approach
temperature_kelvin	Temperature in Kelvin
temperature_mean	Mean Temperature
temperature_min	Minimum Temperture
temperature_max	Maximum Temperature
windspeed	Windspeed
ssta	Sea Surface Temperature Anomaly: weekly SST minus weekly climatological SST
date	date of sampling event in format YYYY-MM-DD Format: %Y-%m-%d

‘coral-bleaching-events-per-year.csv’ 185 rows x 5 columns	
region	region where the bleaching event was reported
code	code of the region
year	year of the bleaching event
moderate bleaching events (1-30% bleached)	number of moderate bleaching incidents
severe bleaching events (>30% bleached)	number of severe bleaching incidents

‘global_plastic_production.csv’ 40 rows x 4 columns	
entity	World
year	Year of production
annual_plastic_production_tons	value in tons
annual_plastic_production_million_tons	value in million tons

‘share_global_plastics_to_oceans_by_continent.csv’ 6 rows x 3 columns	
country	country
year	year
share of global plastics emitted to ocean	share in %

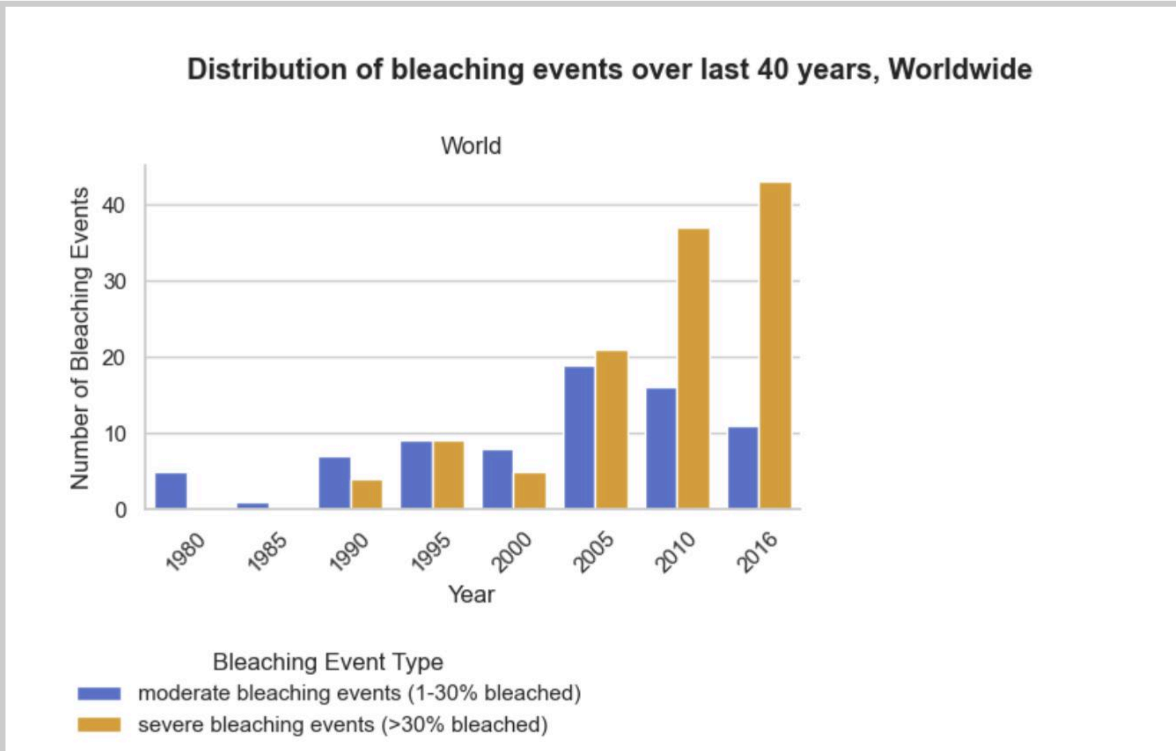
‘plastic_waste_into_ocean_by_country.csv’ 8 rows x 3 columns	
entity	top 8 countries with highest values of mismanaged waste emitted to ocean
year	year
plastic_waste_into_ocean	value in tons

‘decomposition_rates_marine_debris.csv’ 12 rows x 4 columns	
marine_debris_items	marine debris items type
year	year of the report
decomposition rates of marine debris (years)	Average estimated decomposition times of typical marine debris items
color	lightblue for 'plastic items', orange for 'others'

API

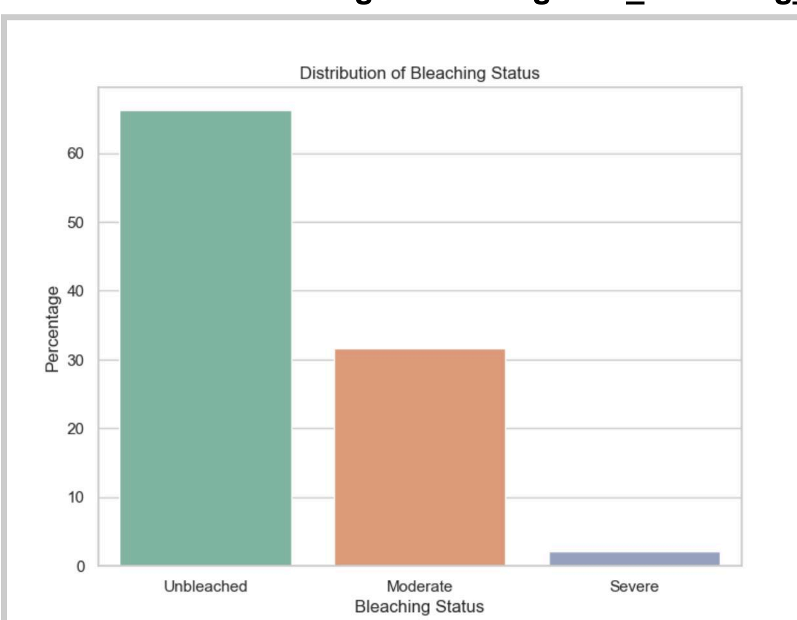
avg_marine_ph.csv' 1 196 rows x 10 columns	
goal	goal number, from the United Nations 17 Sustainable Development Goals
target	target number
indicator	indicator number
country_code	country code
country	country
avg_marine_acidity_ph	average pH measured in the ocean
sampling_stations	stations where the sampling was done
latitude	latitude
longitude	longitude
year	year of the sampling

Visualizations



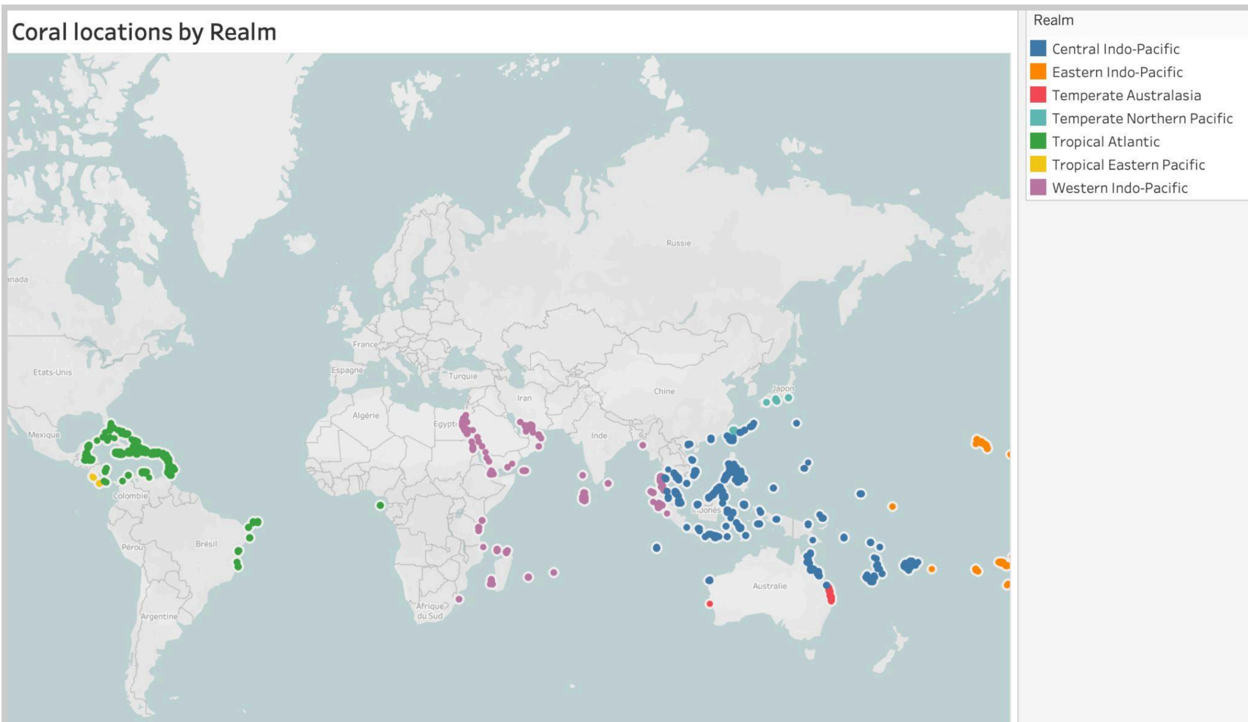
➔ We can observe a significant increase in the number of bleaching events since 1980, with a peak of severity between 2010 and 2016.

Distribution of bleaching status in 'global_bleaching_env' dataset



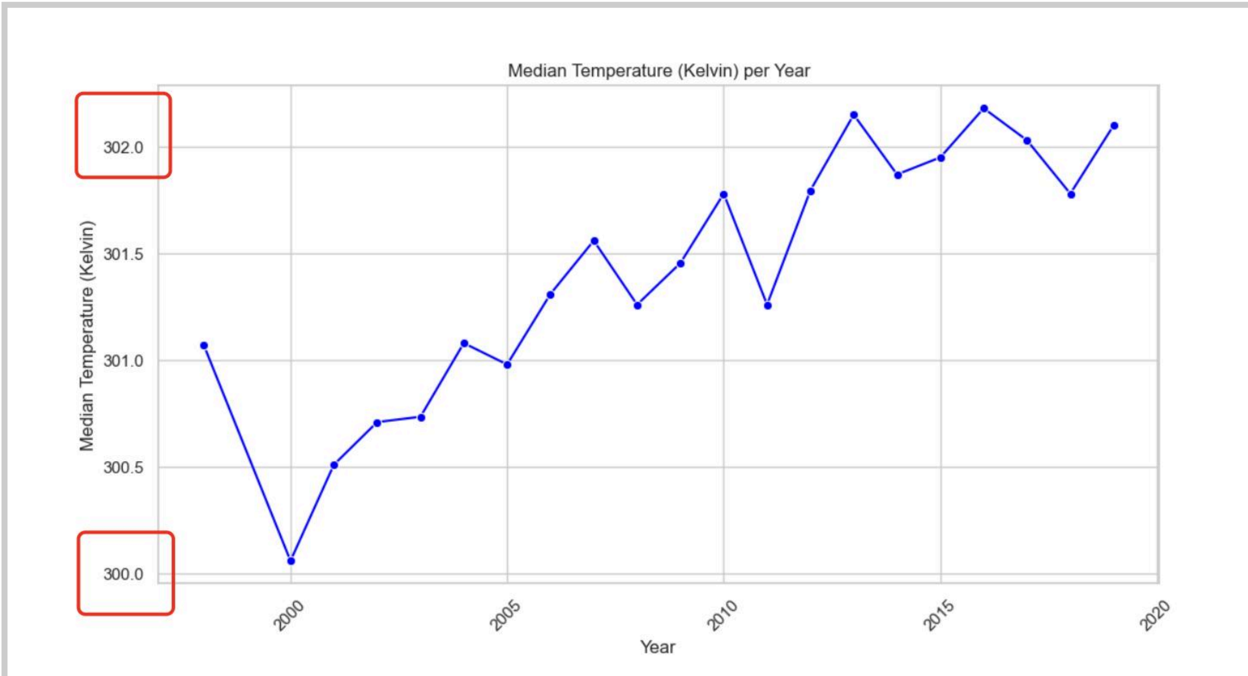
➔ We can notice a quite unbalanced dataset with more than 60% of unbleached versus around 30% moderate and less than 5% severe.

Coral locations per Realm



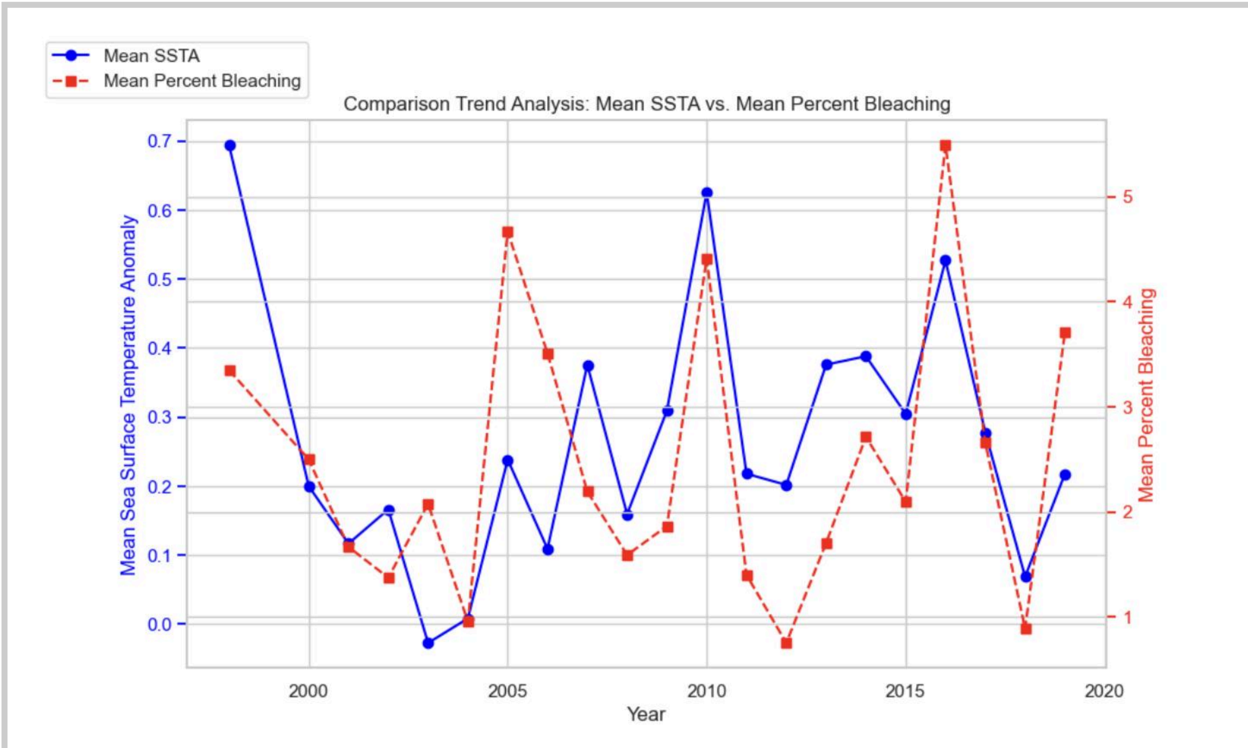
➔ We can notice that most coral reefs sampled are close to the equator.

Median Temperature (Kelvin) over the last 20 years



➔ I chose to start this line chart at 300 instead of 0, to point out the slight variations in temperatures over time - we can observe an overall increase in median temperature (Kelvin) over the last 20 years.

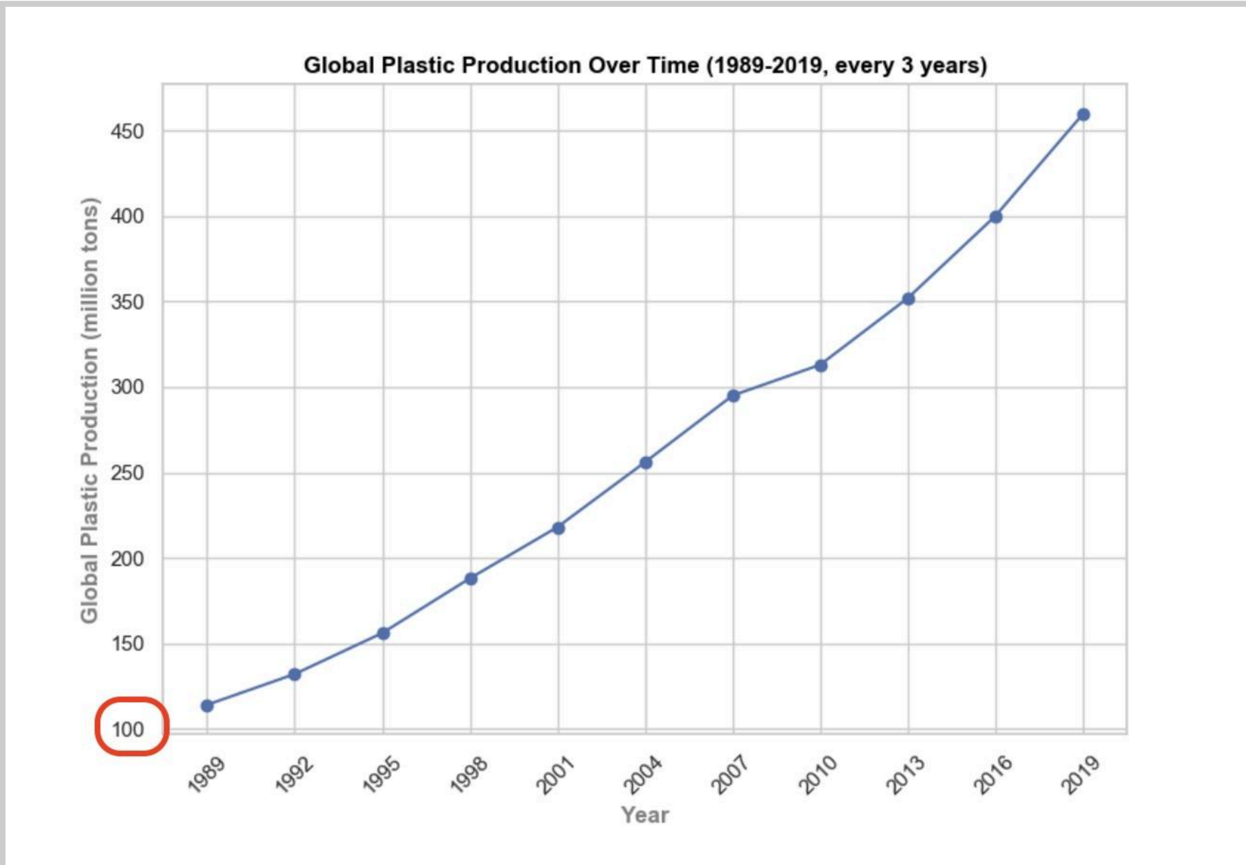
Comparison Trend Analysis : Average SSTA* vs Average Percent Bleaching



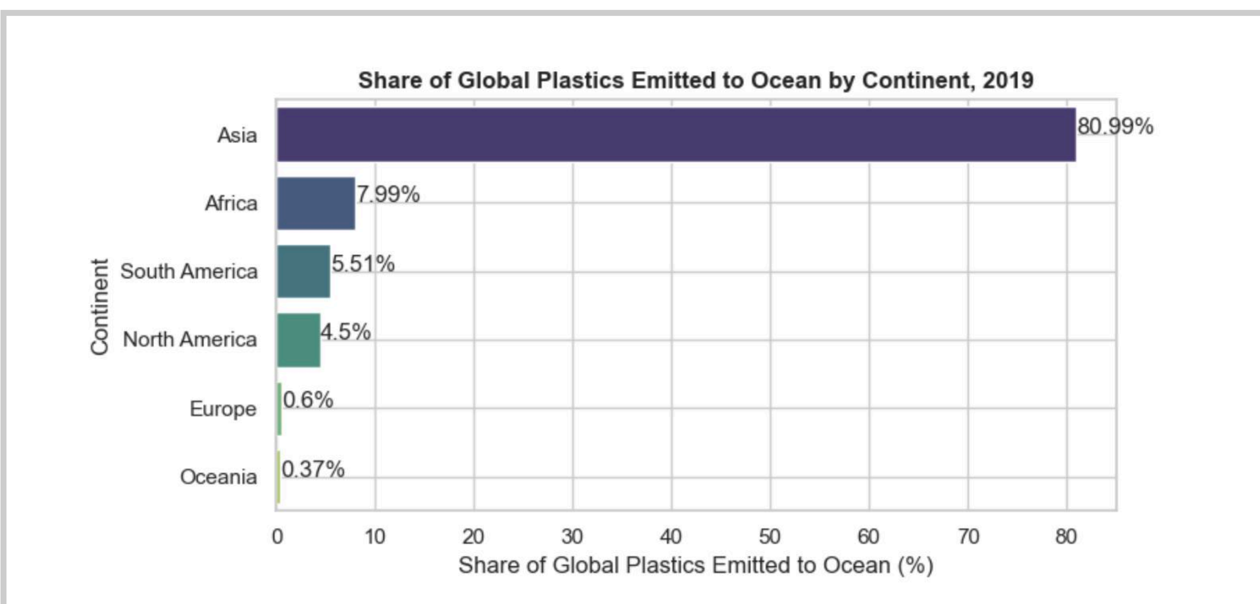
*SSTA : stands for "Sea Surface Temperature Anomaly," which refers to the deviation of the sea surface temperature from the long-term average temperature for a specific location and time of year.

➔ We can observe that as the average SSTA increases, so does the average percent bleaching, with a massive peak in 2016.

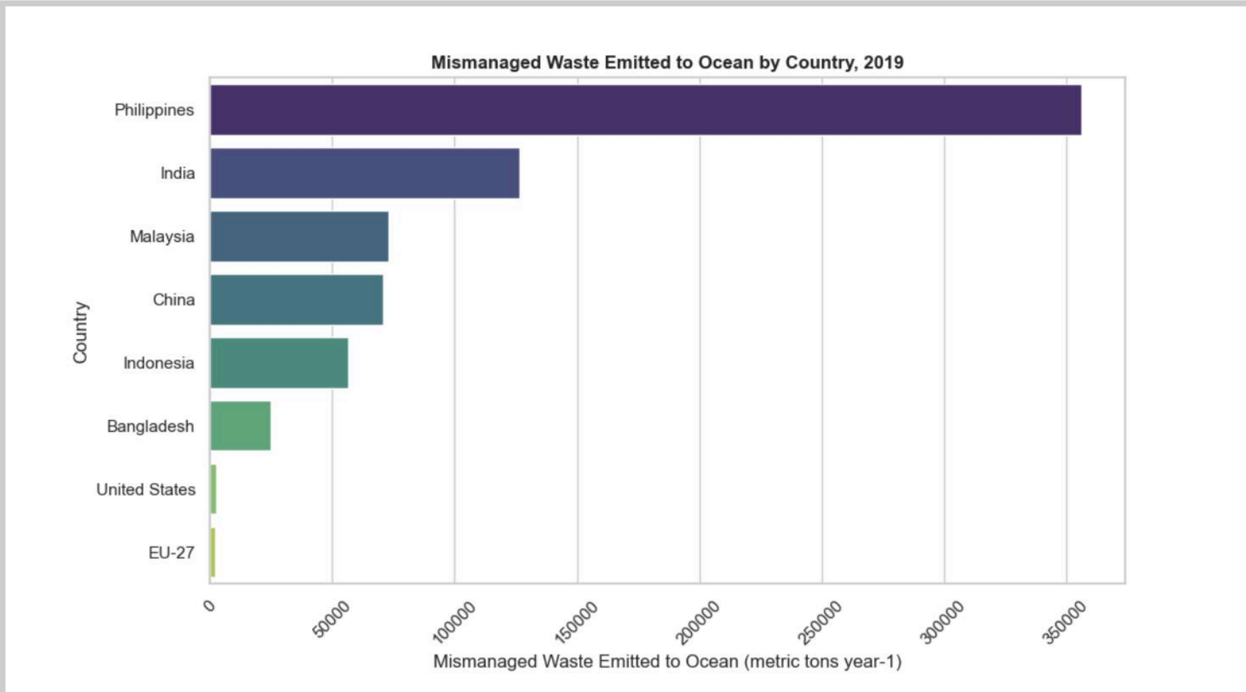
Global Plastic Production over the last 30 years, worldwide



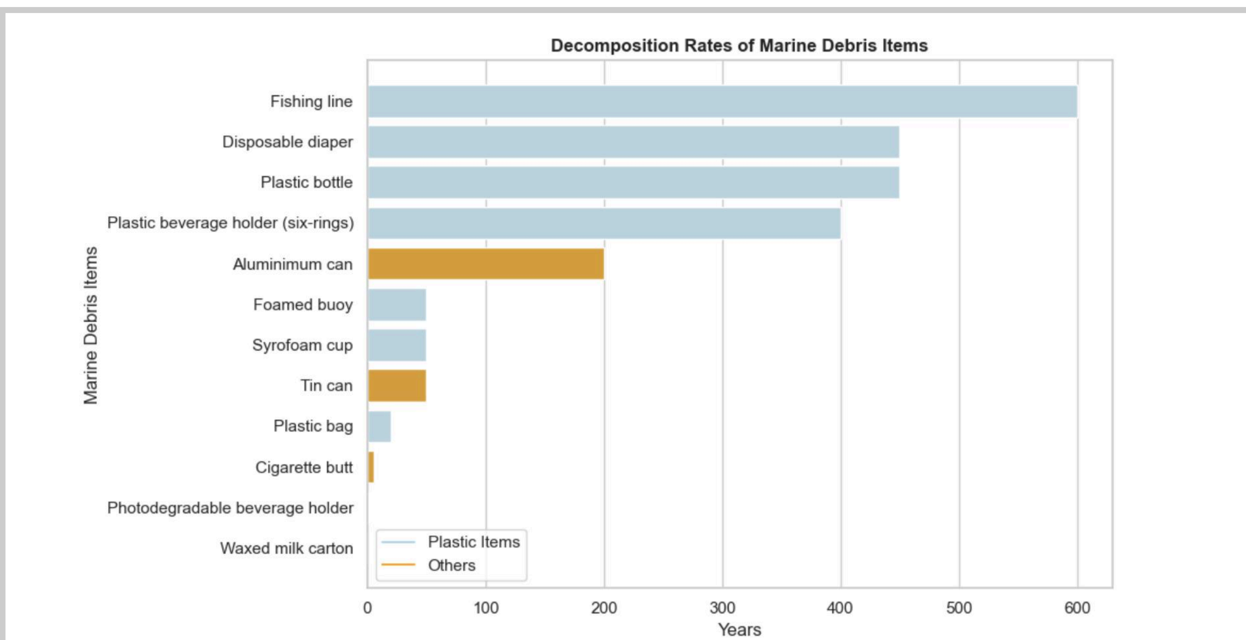
Share of global plastics that ends to the ocean per continent, 2019



Mismanaged wasted emitted to the ocean by country, 2019



Decomposition rates (years) of typical marine debris items



Database type selection

Given that my data follows a predefined schema and relies on foreign keys to establish connections between tables, opting for a Relational Database seemed a better choice. It will enable me to enhance data integrity, facilitate data manipulation, and utilize SQL for complex queries involving data from multiple tables.

Database creation

After exporting my dataframe from Python to MySQL, the 'final_project' database was created in MySQL Workbench to store 5 tables related to the bleaching and environmental data for global coral reef sites from 1980 to 2020 from https://www.bco-dmo.org/dataset/773466.

- ❖ Connection and creation of the database on MySQL:

```python
import pandas as pd
from sqlalchemy import create_engine , text



pw_raw ='SQL2024!'#os.getenv('mysql_pass')
connection_string = 'mysql+pymysql://root:' + pw_raw + '@localhost:3306/'
engine = create_engine(connection_string)

with engine.connect() as conn:
    conn.execute(text(f"CREATE DATABASE IF NOT EXISTS final_project"))

df2.to_sql('global_bleaching_env', engine, 'final_project', if_exists='replace', index=False)
```

```
1.2s
21836
```

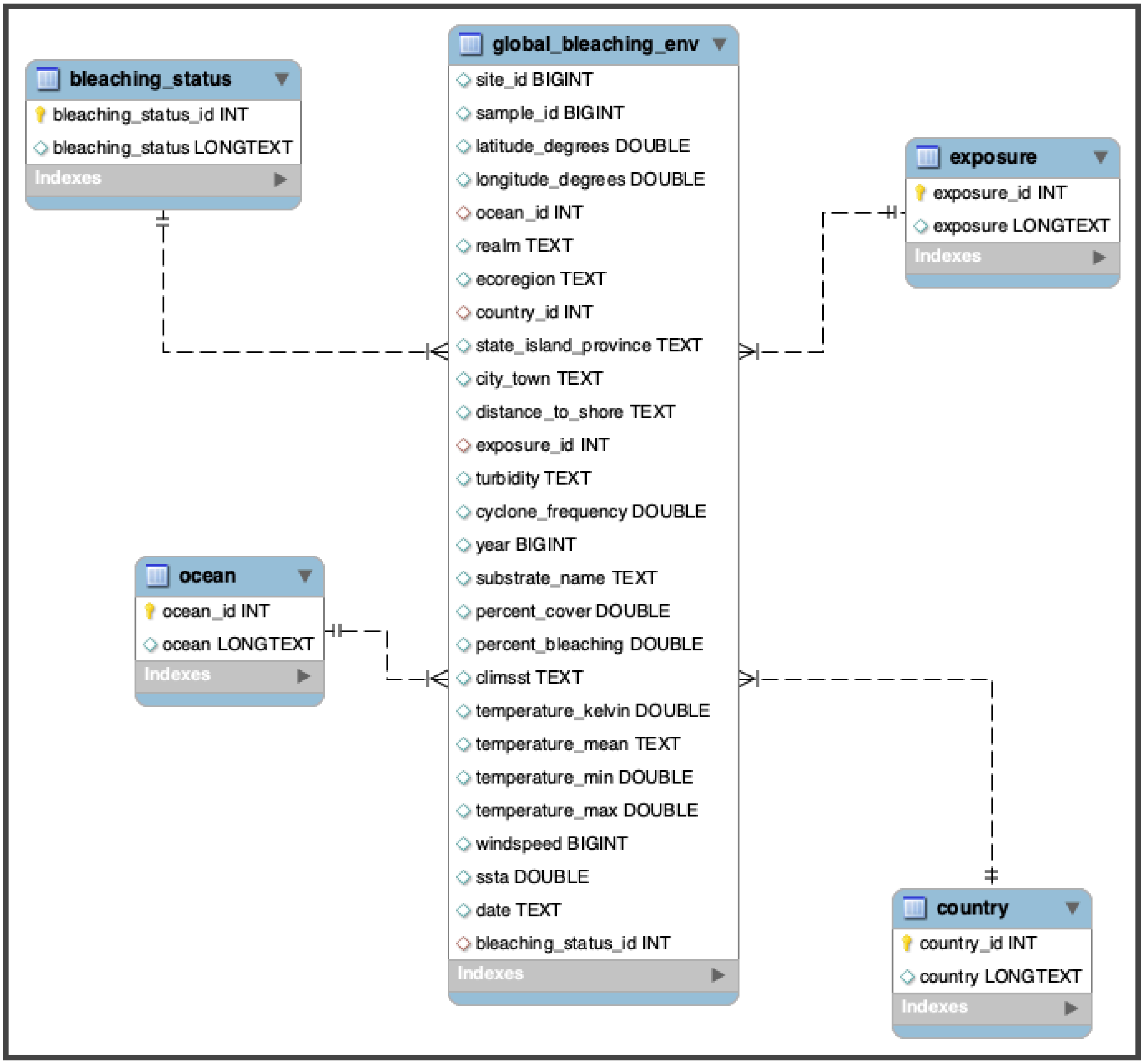
❖ Breaking down tables, creating foreign keys and populating each new table created:

Example of ‘country’ table creation - same structure has been used for the other tables

```sql
use final_project;
select *
from global_bleaching_env;

-- COUNTRY TABLE ----------------------
create table if not exists country (
country_id int auto_increment,
country  LONGTEXT,
primary key (country_id));

select * from country;

-- lets populate the table by inserting the unique values for that dimension
insert into country(country)
select distinct country  from global_bleaching_env order by country asc;
```

```sql
select * from country;

-- now lets adjust the original table so we will use this table
alter table global_bleaching_env add column country_id int after country;

-- lets set up the foreign key reference
alter table global_bleaching_env ADD CONSTRAINT country_fk FOREIGN KEY (country_id) REFERENCES country (country_id);

-- populate the column using the dimension table we created
update global_bleaching_env, country
set global_bleaching_env.country_id = country.country_id
where global_bleaching_env.country = country.country;

-- lets drop the original column now
alter table global_bleaching_env drop column country;
```

ERD



bleaching_status
bleaching_status_id INT
bleaching_status LONGTEXT
Indexes
global_bleaching_env
site_id BIGINT
sample_id BIGINT
latitude_degrees DOUBLE
longitude_degrees DOUBLE
ocean_id INT
realm TEXT
ecoregion TEXT
country_id INT
state_island_province TEXT
city_town TEXT
distance_to_shore TEXT
exposure_id INT
turbidity TEXT
cyclone_frequency DOUBLE
year BIGINT
substrate_name TEXT
percent_cover DOUBLE
percent_bleaching DOUBLE
climsst TEXT
temperature_kelvin DOUBLE
temperature_mean TEXT
temperature_min DOUBLE
temperature_max DOUBLE
windspeed BIGINT
ssta DOUBLE
date TEXT
bleaching_status_id INT
Indexes
exposure
exposure_id INT
exposure LONGTEXT
Indexes
ocean
ocean_id INT
ocean LONGTEXT
Indexes
country
country_id INT
country LONGTEXT
Indexes

MySQL Queries

Examples of 5 queries in MySQL:

❖ Counting the number of bleaching events per year, ordered by descending value

```sql
-- Number of bleaching events per year, ordered by descending values
SELECT YEAR(date) AS year, COUNT(*) AS num_bleaching_events
FROM global_bleaching_env
WHERE percent_bleaching > 0
GROUP BY YEAR(date)
ORDER BY num_bleaching_events DESC;
```

100% 36:16

Result Grid Filter Rows: Search Export:

year	num_bleaching_events
2006	632
2016	536
2017	524
2007	522
2005	510
2014	494
2013	468

`

❖ Creating view for average total area covered by substrate name per ocean:

```sql
-- Average total area covered by substrate name for each ocean
CREATE VIEW substrate_cover_view AS
SELECT
    gbe.substrate_name,
    ocean.ocean,
    ROUND(AVG(gbe.percent_cover), 2) AS total_area_covered
FROM
    global_bleaching_env AS gbe
JOIN
    ocean ON gbe.ocean_id = ocean.ocean_id
GROUP BY
    gbe.substrate_name, ocean.ocean;

--
SELECT *
FROM substrate_cover_view;

--
```

Result Grid | Filter Rows: Search | Export:

substrate_name	ocean	total_area_cover...
Hard Coral	Arabian Gulf	42.45
Nutrient Indicator Algae	Arabian Gulf	1.19
Fleshy Seaweed	Arabian Gulf	1.25
Hard Coral	Atlantic	18.02
Nutrient Indicator Algae	Atlantic	17.74
Fleshy Seaweed	Atlantic	19.23
Hard Coral	Indian	31.66
Nutrient Indicator Algae	Indian	2.22
Fleshy Seaweed	Indian	0.52

❖ Counting number of samples per bleaching_status for each substrate_name :

```sql
-- Bleaching status per substrate name
SELECT
    gbe.substrate_name,
    bs.bleaching_status,
    COUNT(*) AS count
FROM
    global_bleaching_env gbe
JOIN
    bleaching_status bs ON gbe.bleaching_status_id = bs.bleaching_status_id
GROUP BY
    gbe.substrate_name, bs.bleaching_status;
```

substrate_name	bleaching_stat...	count
Hard Coral	Unbleached	7232
Nutrient Indicator Algae	Unbleached	7062
Fleshy Seaweed	Unbleached	170
Hard Coral	Moderate	3461
Nutrient Indicator Algae	Moderate	3413
Fleshy Seaweed	Moderate	48
Hard Coral	Severe	225
Nutrient Indicator Algae	Severe	225

❖ Top countries with highest percentage of bleaching_events in 2016 :

```sql
-- Top countries with highest percent_bleaching in 2016
SELECT
    YEAR(gbe.date) AS bleaching_year,
    c.country,
    MAX(gbe.percent_bleaching) AS max_percent_bleaching
FROM
    global_bleaching_env gbe
JOIN
    country c ON gbe.country_id = c.country_id
WHERE
    YEAR(gbe.date) = 2016
GROUP BY
    YEAR(gbe.date), c.country
ORDER BY
    max_percent_bleaching DESC;
```

100% 26:76

Result Grid Filter Rows: Search Export:

bleaching_ye...	country	max_percent_bleachi...
2016	Indonesia	95
2016	Australia	72.5
2016	Malaysia	72.5
2016	Maldives	70
2016	Thailand	63.75
2016	Egypt	25
2016	Turks and Caicos	25

- ❖ Create new table ‘sample’ that will be used to create the Flask API:

```sql
create table sample as
select sample_id, latitude_degrees, longitude_degrees, ocean
, realm, ecoregion, country, state_island_province, city_town
, distance_to_shore, exposure, year, date, turbidity, cyclone_frequency
, sum(case when substrate_name = 'Hard Coral' then percent_cover end) as hard_coral_percent_cover
, sum(case when substrate_name = 'Nutrient Indicator Algae' then percent_cover end) as nutrient_indicator_algae_percent_cover
, sum(case when substrate_name = 'Fleshy Seaweed' then percent_cover end) as fleshy_seaweed_percent_cover
, sum(case when substrate_name = 'Hard Coral' then percent_bleaching end) as hard_coral_percent_bleaching
, sum(case when substrate_name = 'Nutrient Indicator Algae' then percent_bleaching end) as nutrient_indicator_algae_percent_bleaching
, sum(case when substrate_name = 'Fleshy Seaweed' then percent_bleaching end) as fleshy_seaweed_percent_bleaching
from global_bleaching_env b
left join ocean o on b.ocean_id = o.ocean_id
left join country c on c.country_id = b.country_id
left join exposure e on e.exposure_id = b.exposure_id
left join bleaching_status bs on bs.bleaching_status_id = b.bleaching_status_id
group by 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15;

select *
from sample;
```

100% 1:17

Result Grid Filter Rows: Search Export: Fetch rows:

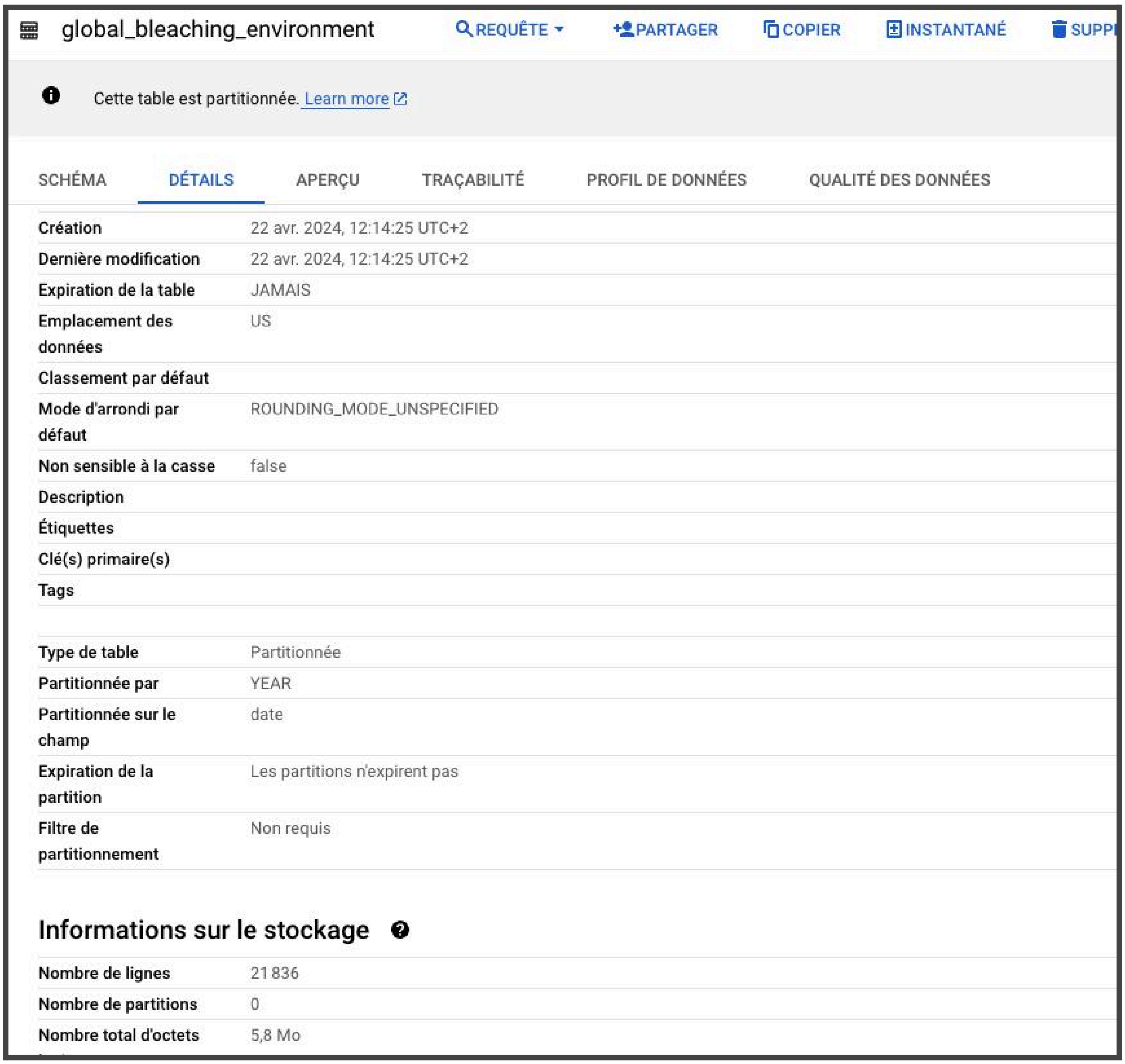
sam...	latitude_degre...	longitude_degre...	ocean	realm	ecoregion	country	state_island_provin...	city_town	distance_to_sh...	exposure
10307600	12.9627	100.653	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Chon Buri	Ko Luam	59.25	Sheltered
10307601	11.9453	102.2833	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Chang District	3641.79	Exposed
10307602	11.945	102.2875	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Chang District	3313.49	Exposed
10307603	11.9049	102.3174	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Chang District	2802.64	Exposed
10307605	11.8052	102.3785	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Kut District	138.38	Sheltered
10307606	11.8052	102.3785	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Kut District	138.38	Sheltered
10307607	11.7903	102.3833	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Kut District	102.93	Exposed
10307608	11.7972	102.3889	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Kut District	64.22	Sheltered
10307609	11.7839	102.3933	Pacific	Central Indo-Pacific	Gulf of Thailand	Thailand	Trat Province	Ko Kut District	98.01	Sheltered

sample 3

Action Output

BigQuery

Later, that dataset was denormalized for BigQuery :



When we highlight the following query, we can see that it will process 5 MB of data during its execution:

Whereas, with the partitioning by year, it will process less data : 335,65 KB

This way, we can improve query performance and reduce costs because fewer data are processed. This is particularly beneficial for organizations with massive datasets, as it helps optimize resource usage and minimize costs associated with query execution.

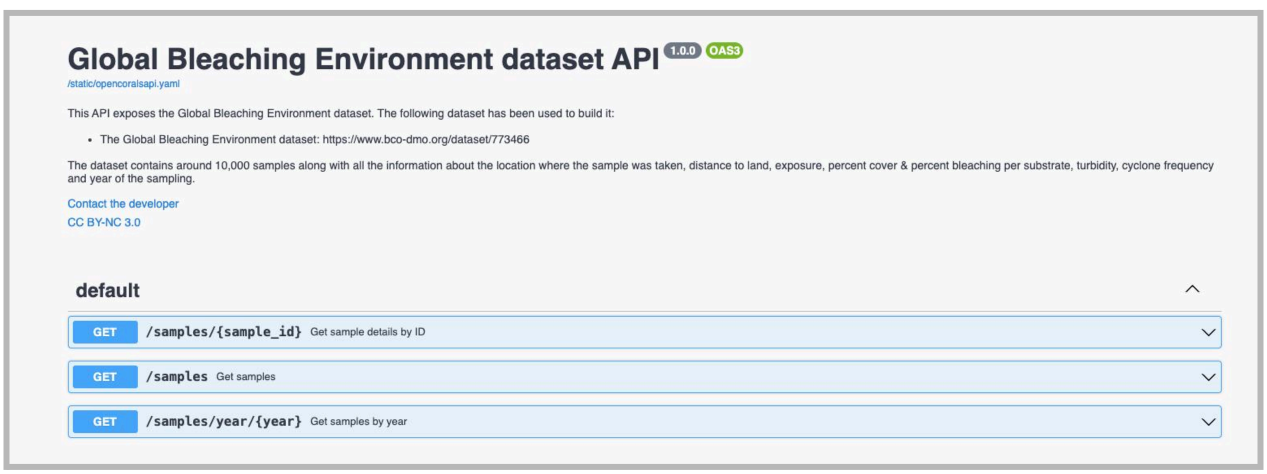
Exposing Data via API

This API serves as a gateway to access the Global Bleaching Environment dataset, a comprehensive collection of data pertaining to coral bleaching events worldwide. Leveraging this API, researchers, marine scientists, and environmentalists can retrieve specific information on coral bleaching incidents across different regions and years.

The Global Bleaching Environment dataset, sourced from https://www.bco-dmo.org/dataset/773466, comprises approximately 10,000 samples encompassing crucial details such as sample location, distance to land, exposure, percent cover, percent bleaching per substrate, turbidity, cyclone frequency, and sampling year.

Through this API, stakeholders gain access to a valuable resource for monitoring and analyzing global coral bleaching trends over the past several decades, aiding in conservation efforts and informed decision-making for marine ecosystem preservation.

Built on Flask, the API supports GET requests, allowing users to specify parameters such as the sample, year, and sample ID. Responses are delivered in JSON format, offering seamless integration with various data analysis tools and platforms.



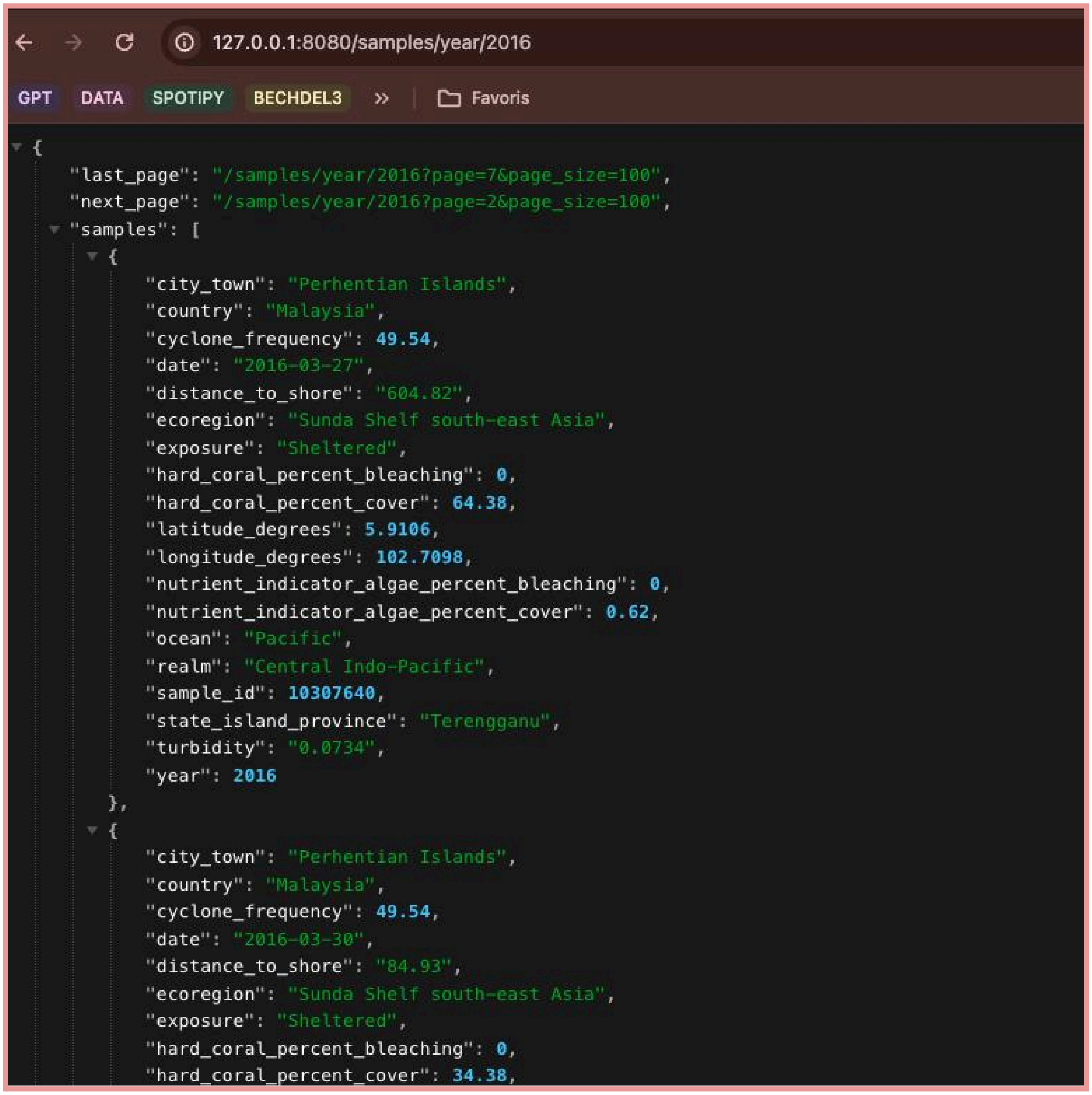
Example endpoints include:

- http://127.0.0.1:8080/samples/10313996 ⇒ get info for one specific sample_id

127.0.0.1:8080/samples/10313996


```json
{
    "city_town": "Magnetic Island",
    "country": "Australia",
    "cyclone_frequency": 43.39,
    "date": "2006-10-08",
    "distance_to_shore": "126.24",
    "ecoregion": "Central and northern Great Barrier Reef",
    "exposure": "Sometimes",
    "hard_coral_percent_bleaching": 0,
    "hard_coral_percent_cover": 29.38,
    "latitude_degrees": -19.1219,
    "longitude_degrees": 146.8808,
    "nutrient_indicator_algae_percent_bleaching": 0,
    "nutrient_indicator_algae_percent_cover": 0.62,
    "ocean": "Pacific",
    "realm": "Central Indo-Pacific",
    "sample_id": 10313996,
    "state_island_province": "Queensland",
    "turbidity": "0.1384",
    "year": 2006
}
```

- http://127.0.0.1:8080/samples/year/2016 ⇒ get info for all samples for a specific year



Machine Learning

Coral Health Classification

- Assumptions:

With increasing concern about coral reef health worldwide, there's a growing need for tools to easily identify healthy and bleached corals. This information is crucial for conservation efforts and raising awareness on the impact of environmental changes on coral reefs.

- Coral Health Classifier:

To meet this need, I'm planning to develop a coral health classifier using convolutional neural networks (CNNs), to classify corals based on their health status. This tool will use a dataset of coral images labeled as either healthy or bleached from Kaggle.

Users can input images of corals they're interested in analyzing. The classifier will then examine the image and determine whether the coral appears healthy or bleached. This process helps researchers, conservationists, and reef enthusiasts quickly assess coral health in their local areas or research projects.

Conclusions

Our analysis delved into the urgent matter of global ocean trends, specifically focusing on the threats coral reefs face from ocean warming and marine plastic pollution. The results of our study emphasize the critical need for immediate action to protect these invaluable ecosystems.

During our investigation of marine plastic pollution trends, we uncovered concerning findings about its impact on ocean health. **Asia** emerged as a major contributor, accounting for over **80% of global plastic inputs into the ocean**, with the **Philippines** alone contributing **more than one-third** of these inputs. Additionally, we learned that marine debris items, particularly plastics, can take **over 400 years to decompose**, highlighting the long-lasting nature of this environmental threat.

Furthermore, our analysis revealed a significant increase in coral bleaching events worldwide over the past four decades, with a notable peak observed between 2010 and 2016. While our dataset did not definitively establish a direct link between rising temperatures and coral bleaching, additional research on the topic confirms that ocean warming, driven by climate change, is the leading cause of bleaching events. According to the National Oceanic and Atmospheric Administration (NOAA), approximately **75%** of the world's tropical coral reefs experienced **severe heat stress between 2014 and 2017**, resulting in widespread bleaching events.

Currently, the **Great Barrier Reef in Australia** is experiencing its **worst coral bleaching event ever recorded**, according to a recently published article in Le Monde (see references).

In light of these findings, it is clear that concerted efforts are required to address the root causes of ocean warming and marine plastic pollution. By implementing proactive conservation measures and promoting sustainable practices, we can work towards protecting coral reefs and preserving the health of our oceans for generations to come.

GDPR

Upon thorough examination of the data collected for this project, I confirm that no personal data was utilized throughout the project. All data sources used are publicly available at a country level, ensuring transparency and compliance with General Data Protection Regulation (GDPR) guidelines.

References

Flat Files:

- https://ourworldindata.org/grapher/coral-bleaching-events
- https://www.bco-dmo.org/dataset/773466
  - ➔ Metadata before cleaning Metadata
- https://ourworldindata.org/plastic-pollution

API:

- UNSD SDGs API

Web Scraping:

- https://www.foxnews.com/category/science/planet-earth/oceans

Machine Learning:

- https://www.kaggle.com/datasets/vencerlanz09/healthy-and-bleached-corals-image-classification

Trello Board:

- https://trello.com/b/oB9swyJ6/final-project

Github repository (in progress):

- https://github.com/Smita401/final-project-life-below-water.git

Additional resources:

- Le réchauffement des océans entraîne un blanchissement massif des coraux dans le monde
- What is coral bleaching?.