Obligatorisk aflevering 2

Af Kristofer, Michael, Søren og Mads

Explore the data and formulate considerations about a hosting database.

To create a database, we must first explore and understand the different types of data in each dataset, as well as how the different datasets are connected to each other with the data they contain. In this section we will go over the different datasets and look at what columns they contain, and for each column we will look through the data contained within to determine what type of value they are. We will also try to give each a quick description to help us understand the relationships between all of the datasets.

Data exploration:

2016_-_Cities_Emissions_Reduction_Targets_20240207.csv:

Organisation: Looking at the column of data we can see that they are strings writing in the native language where the organization resides. The data are names of different government organizations overseeing a city. The value is unique.
Account No: Account number or id in non-sequential order. The value is unique to each organization.
Country: The country of which the organization is located within. The values are string and contain names. One country can have multiple organizations within it. Unlike the values of Organisation the values are all written in english.
City Short Name: The shortened name of the city where the organization resides. The values are string and some contain special characters.
C40: The value type is a string and represents if a city can be labeled as a C40 city.
Reporting Year: The year the organization gave the report from which the data of the row is gathered from. The data is an integer.
Sector: From which part of the organization the data has been gathered from. The values first read as an enum but with a few rows of data where multiple choices are included means the value should be a list of enums.
Target boundary: The limit where emission data has been collected within. The value is a string and is used as a description. The value in each row varies with some rows missing data.
Baseline year: The year where the baseline measurements were taken. The value is an integer.

Baseline emissions (metric tonnes CO2e): The emission value measured as the baseline. The value is a float and represents metric tons.
Percentage reduction target: The desired reduction of the baseline emission measured. The value is a float and represents percentage.
Target date: The year in which the organization aims to have the baseline emission reduced by the Percentage reduction target. The value is an integer.
Comment: String value containing a comment from the rapport. Some rows have missing values.
City Location: Coordinates of the city's location which most likely refers to the center of the city. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.
Country Location: Coordinates of the county’s location which most likely refers to the center of the country. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.

2016_-_Citywide_GHG_Emissions_20240207.csv:

Account Number: Account number or id in non-sequential order.
City Name: The full name of the city. The value is a string.
Country: The country of which the organization is located within. The values are string and contain names. One country can have multiple organizations within it. Unlike the values of Organisation the values are all written in english.
City Short Name: The shortened name of the city where the organization resides. The values are string and some contain special characters.
C40: The value type is a string and represents if a city can be labeled as a C40 city.
Reporting Year: The year the organization gave the report from which the data of the row is gathered from. The data is an integer.
Measurement Year: The year where the measurements were taken. The value is a date.
Boundary: The limit where emission data has been collected within. The value is a string and is used as a description. The value in each row varies with some rows missing data.
Primary Methodology: Name of the methodology used for the data collection. The value is a string.
Methodology Details: A description of the Primary Methodology used. The value is a string.
Gases included: What type of gases are included in the measurements. The value can contain multiple choices which could be considered enums since the gases that can be included appear to be static.
Total City-wide Emissions (metric tonnes CO2e): The value is a float and represents metric tons.
Total Scope 1 Emissions (metric tonnes CO2e): The value is a float and represents metric tons.
Total Scope 2 Emissions (metric tonnes CO2e): The value is a float and represents metric tons.
Increase/Decrease from last year If there have been any changes in their measurements compared to last year. The value is an enum which represents “=, >, <” or if it's their first measurement.

Reason for increase/decrease in emissions: A description given in the report. The value is a string.
Current Population Year: The year of the Current Population. The value is an integer.
Current Population: The measured number of people living in the country. The value is an integer
City GDP: The Gross Domestic Product of the market value within a city. The value is an integer.
GDP Currency: What type of currency the City CDP refers to. The value is a string which can be separated into the long name and short name of the currency
Year of GDP: The year of the City GDP was measured. The value is an integer.
GDP Source: Where the information regarding the GDP was gathered. The value is a string and can both be a name of an organization or a link to a website.
Average annual temperature (in Celsius): The average annual temperature of the city in celsius. The value is a float.
Land area (in square km): The size of the city in square kilometers. The value is a float.
Average altitude (m): The average altitude in meters of the city. The value is an integer.
City Location: Coordinates of the city's location which most likely refers to the center of the city. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.
Country Location: Coordinates of the county’s location which most likely refers to the center of the country. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.

2017_-_Cities_Community_Wide_Emissions.csv:

Account number: Account number or id in non-sequential order. The value is unique to each organization.
Organization: Looking at the column of data we can see that they are strings writing in the native language where the organization resides. The data are names of different government organizations overseeing a city. The value is unique.
City: The full name of the city. The value is a string.
County: The country of which the organization is located within. The values are string and contain names. One country can have multiple organizations within it. Unlike the values of Organisation the values are all written in english.
Region: Where on the world map the country is located. The value is a string and represents the different names of the world regions.
C40: The value type is a string and represents if a city can be labeled as a C40 city.
Access: If the report is accessible to the public or private. It would be reasonable to assume that there would be reports which are or would be marked as private then we can reflect that the data is or is not public. The value is a bool representing public or private.
Reporting year: The year the organization gave the report from which the data of the row is gathered from. The data is an integer.
Accounting year: The period of time where the data gathered in the report originated from. The value is a string containing two dates(YYYY-MM-DD) separated by a “-” and represents a period of time.
Boundary: The limit where data has been collected within. The value is a string and is used as a description. The value in each row varies with some rows missing data.
Protocol: The name of the protocol used for the gathering of information. The value is a string.
Protocol column: A description of how the protocol was used to gather information for the report. The value is a string.
Gases included: A description of the type of gases included when gathering information. The value is a string and contains mostly a list of the chemical name of the gases but also contains a description.

Total emissions (metric tonnes CO2e): The total measured emissions of CO2 in tons. The value is float. Most of the values in the column could be integers but the few who aren't forces the others.
Scopes Included: What scopes are included in the report. The value is a string.
Total Scope 1 Emissions (metric tonnes CO2e): The value is a float and represents metric tons.
Total Scope 2 Emissions (metric tonnes CO2e): The value is a float and represents metric tons.
Comment: A comment in there may be in the report. The value is a string and can contain empty values.
Increase/Decrease from last year: If there have been any changes in their measurements compared to last year. The value is an enum which represents "=, >, <" or if it's their first measurement.
Reason for increase/decrease in emissions: A description given in the report. The value is a string.
Population:% The measured number of people living in the country. The value is an integer
Population year: The year of the Population. The value is an integer.
GDP: The Gross Domestic Product of the market value within a city. The value is an integer.
GDP Currency: What type of currency the City CDP refers to. The value is a string which can be separated into the long name and short name of the currency
GDP Year: The year of the City GDP was measured. The value is an integer.
GDP Source: Where the information regarding the GDP was gathered. The value is a string and can both be a name of an organization or a link to a website.
Average annual temperature (in Celsius): The average annual temperature of the city in celsius. The value is a float.
Average altitude (m): The average altitude in meters of the city. The value is an integer.
Land area (in square km): The size of the city in square kilometers. The value is a float.
City Location: Coordinates of the city's location which most likely refers to the center of the city. The

value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.
Country Location: Coordinates of the county’s location which most likely refers to the center of the country. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.

2017_-_Cities_Emissions_Reduction_Targets_20240207.csv:

Account No: Account number or id in non-sequential order.
Organisation: Looking at the column of data we can see that they are strings writing in the native language where the organization resides. The data are names of different government organizations overseeing a city. The value is unique.
City: The full name of the city. The value is a string.
Country: The country of which the organization is located within. The values are string and contain names. One country can have multiple organizations within it. Unlike the values of Organisation the values are all written in english.
Region: Where on the world map the country is located. The value is a string and represents the different names of the world regions.
Access: If the report is accessible to the public or private. It would be reasonable to assume that there would be reports which are or would be marked as private then we can reflect that the data is or is not public. The value is a bool representing public or private.
C40: The value type is a string and represents if a city can be labeled as a C40 city.
Reporting year: The year the organization gave the report from which the data of the row is gathered from. The data is an integer.
Type of target: The value type is an enum: “Absolute target” “Base year intensity target” “Baseline scenario (business as usual) target”
Sector: The value type is string. It would be an enum if not for “Other” which offers a string description.
Baseline year: The year where the baseline measurements were taken. The value is an integer.
Baseline emissions (metric tonnes CO2e): The emission value measured as the baseline. The value is a float and represents metric tons.
Percentage reduction target: The desired reduction of the baseline emission measured. The value is a float and represents percentage.

Target date: The year in which the organization aims to have the baseline emission reduced by the Percentage reduction target. The value is an integer.
Estimated business as usual absolute emissions in target year (metric tonnes CO2e): How much CO2 is estimated to be produced as a byproduct of business. The type is an integer and can be empty.
Intensity unit (emissions per): The value type is a string and is a description of what “per” the measurements are taking.
Comment: String value containing a comment from the rapport. Some rows have missing values.
Population: The measured number of people living in the country. The value is an integer
Population Year: The year of the Population. The value is an integer.
City Location: Coordinates of the city's location which most likely refers to the center of the city. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.
Country Location: Coordinates of the county’s location which most likely refers to the center of the country. The value is a string considering the use of parentheses. Within the parentheses there are two numbers separated by a comma: latitude and longitude.

2023_Cities_Climate_Risk_and_Vulnerability_Assessments_20240207.csv:

Questionnaire: The value is a string. All the values within the column are the same: “Cities 2023”.
Organization Number: Account number or id in non-sequential order.
Organization Name: Name of the organization. The value is a string and the names within are written in the languages of the country it resides in.
City: Name of the city the organization resides in. The value is a string and some values are empty.
Country/Area: The country the city and organization resides in. The value is a string and the names are written in english.
CDP Region: The state/region relevant CDP reporting platform. The value is a string and contains the name of a region.
C40 City: The value is a bool.
GCoM City: The value is a bool.
Access: If the report is accessible to the public or private. It would be reasonable to assume that there would be reports which are or would be marked as private then we can reflect that the data is or is not public. The value is a bool representing public or private.
Assessment attachment and/or direct link A PDF of download link for an assessment document. The value may be meant to contain a pdf file but all values within the column is a string with some being a filename and extension name, and some being a web link.
Confirm attachment/link provided Confirmation of which type of attachment was given. The values indicate an enum of “PDF”, “Link” but also contains “Other” where a description can be given, therefore making the value a string.
Boundary of assessment relative to jurisdiction boundary: The value is an enum indicating the scale of the report. “Smaller - covers only part of the jurisdiction, please explain exclusions: Área referente ao bairro de Porto de Pedra, correspondendo ao mapeamento de risco de movimento de massa e maior abrangência nas ruas Maria de Souza, Dom Marcos de Noronha, Senador Álvaro Uchoa.”

“Partial - covers part of the jurisdiction and adjoining areas, please explain exclusions/additions: Zona precordillerana y cordillerana de Peñalolén. Incluye la principal cuenca de la comuna, abarcando comuna aledaña” “Same - covers entire jurisdiction and nothing else” “Larger - covers the whole jurisdiction and adjoining areas, please explain additions: This is a county-wide plan”
Year of publication or approval: The year where the report was either publicized or approved. The value is an integer.
Factors considered in assessment: A description of what was considered during the assessment. The value is a string.
Primary author(s) of assessment: Who oversaw the assessment. The value is a string and can be empty.
Does the city have adaptation goal(s) and/or an adaptation plan?: The data indicates that the value is of type enum. “Adaptation goal(s) and adaptation plan” “Adaptation plan” “Adaptation goal(s)” “Incomplete report”
Population: The measured number of people living in the country. The value is an integer
Population Year: The year of the Population. The value is an integer.
City Location: The coordinates of the city. The value is a string. The coordinates are within two parentheses and separated with a space.
Last update: This value contains a date and represents when the values of the row was last updated.

Dataset relationship:

Here we will describe how the different datasets are connected to each other through their columns, values, data and a description of the overall meaning of the different reports represented in the datasets. We will also describe what each dataset represents.

From here each dataset will be know as:

Reduc16:

- 2016_-_Cities_Emissions_Reduction_Targets_20240207.csv

GHG:

- 2016_-_Citywide_GHG_Emissions_20240207.csv

Comm:

- 2017_-_Cities_Community_Wide_Emissions.csv

Reduc17:

- 2017_-_Cities_Emissions_Reduction_Targets_20240207.csv

RiskAndVul:

- 2023_Cities_Climate_Risk_and_Vulnerability_Assessments_20240207

Firstly we can see how they relate to each other by doing a quick comparison of some of the different columns in each dataset. We can see that many of them share a lot of data even if the names of the columns don't entirely match across the board. From doing this quick comparison we can also see that they all share an account number which we can use to easily create relationships within our database.

	GHG	Reduc16	Reduc17	Comm	RiskAndVul
Account number	Account Number	Account No	Account No	Account number	Organization Number
Organisation			Organisation	Organization	Organization Name
City	City Name		City	City	City
Country	Country	Country	Country	Country	Country/Area
Access			Access	Access	Access
C40	C40	C40	C40	C40	C40 City
Reporting year	Reporting Year	Reporting Year	Reporting year	Reporting year	
Baseline year	Measurement Year	Baseline year	Baseline year	Accounting year	
Baseline emission	Total City-wide Emissions (metric	Baseline emissions (metric tonnes CO2e)	Baseline emissions (metric tonnes CO2e)	Total emissions (metric tonnes CO2e)	

	tonnes CO2e)				

Organization account number:
Through each of the datasets there is a column, though with a different column name, that represents the account number of an organization that has submitted a report to at least one of the datasets. We can therefore connect all the different reports in the different datasets through their unique account number.

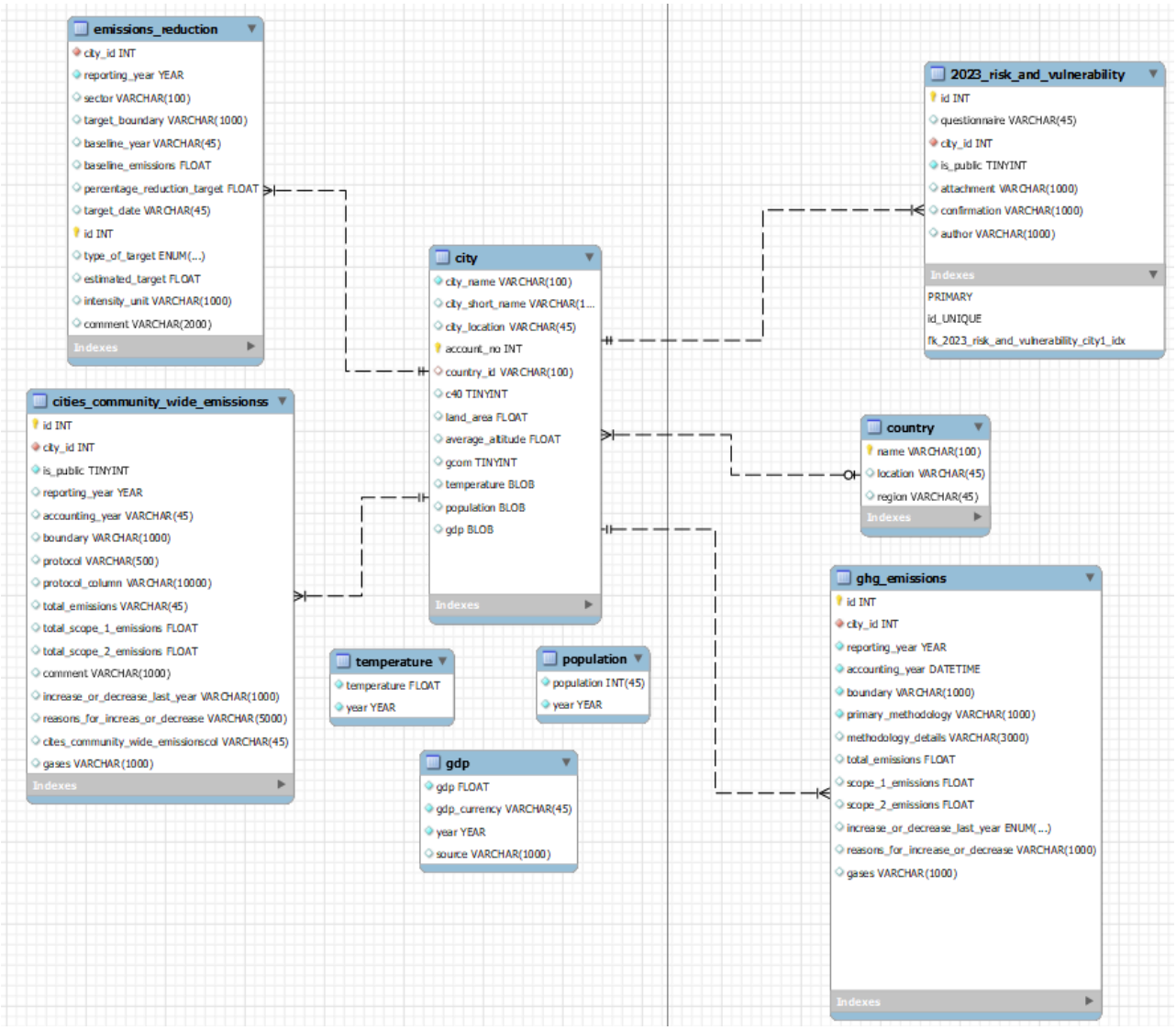
City, County and Coordinates:
The data can give us a picture of where the work against climate change is strongest. Throughout all five datasets there is a consistent representation of where in the world the data is coming from. The consistency only breaks in the newest dataset RiskAndVul, where the setup of the coordinates are different from the other datasets, though the data when extracted is still useful. In RiskAndVul the way the data has been save was: *"POINT (113.813 22.9175)"* and also with the possibility of empty data, where as the other datasets all setup the data as *"*(56.168393, 10.137373)*"* with no empty data.
Since the data is consistent over the different datasets we don't need to consider this difference as long as we save the data so that when the coordinates are extracted as longitude and latitude numbers correctly.

City names and organization:
Looking at the data where both a column of the organization names and city names, can we see that each organization appears to be a government organization tied directly to the city it resides in.

Reduc16 and Reduc17:
Looking at the data within these two datasets and their names, we can see that they are of the same type of report separated by a year. We can however also see that the dataset from 2017 includes new columns of data which separates them from being completely identical to each other.

Design and develop database structure:

Here you can see the EER diagram which we have developed. There are some relations which require our data to be uploaded in a similar manner to our previous SQL database. Unlike in SQL, our document database in MongoDB does not require a setup script as it doesnt need a structure to be present before uploading the data. However, we still felt that an EER diagram could provide us with a basic overview of our data structure. In the diagram each table represents a collection in mongo and the “floating” tables without any relations (temperature, population and gdp) are embedded in the city collection.



Ingest data into database:

Here we will discuss how we introduce the data from the five datasets into our database. We will go over any data that we may or may not exclude, due to for example being unnecessary, and we will go over any data that may be the same in different datasets but with different data value types and how we will handle it.

Deciding include/exclude:

In the datasets *Reduc17* and *RiskAndVul* we see the column “Access” which we view as showing whether or not the general public have access to the report. The data within both datasets is however always “Public” which means we don’t need to push this data to the database as we can generalize that all data, from all five datasets, pushed and pulled to the database is public. We have however chosen to still include it in our database because we have determined it would be better for future proofing the database. While no data currently have any other value than “Public”, we would rather not run the risk of future reports missing this important difference should there be any that contain the value “Private”.

When we have to handle the overlapping data between the datasets we need to determine how we will handle each report type from the datasets. *Reduc16* and *Reduc17* both are the same type of report with the latter including new columns. With them being literally the same report but from different years mean we could store them together in the same table. This however would introduce many empties for the *Reduc16* data arriving from the new columns introduced from *Reduc17*. We have chosen to store the reports from the two datasets together in one table within our database.

Translating data type to new type:

For nearly every instance where a column contains data representing a year of the calendar they have a single integer number, but there are instances where the data type is represented by a variable type of date. This would introduce conflicts when pushing the data to the database. When deciding to translate an integer number to a date or a date to an integer number, we need to determine if the added information from the data that already exists as a date is important enough to store as translating an integer would not give us any new information regarding month, day or time. The added data wouldn’t be of much use considering that the reports are yearly based, so even if there are multiple reports from the same organizations there wouldn’t be multiple of the same target focus during one year. We can therefore discard the information of month, day and time and push only the year to our database.

In the dataset *RiskAndVul* we can see the column C40 being of type boolean whereas in other datasets with the column C40 being a string which can be empty. Since the string data are always empty or contain a string of “C40” then we can translate the strings to a boolean where an empty string equals false and a string of “C40” equals true.

Ingest script overview:

Here we will go over our scripts which we will be used to load the data of the datasets to our database. We will present a quick overview in sequential order of operations from start to end of our script. The order or execution is important as during it non-sequentially will result in errors when uploading to our database. The error occurs due to some ids being generated when uploading, where “City ”have a dependency on “Country” and nearly every other table depends on the table “City”. When first uploading the different countries will auto generate an id which we then get and upload as part of the cities country reference id.

- Country

First we upload the different countries found in the datasets.

- City

After the countries are uploaded we upload the cities.

- Reports (GHG, RiskAndVuln, Reduction, CityWide)

When both countries and cities are uploaded can we upload the information specific to each report type.

Maintenance:

Considering that the type of data we will be storing within our database is not something that should change over time after it's first upload to our database and that any reports stored shouldn't be deleted unless there is a specific call to it, say if an organization is removed or replaced due to politics within a country or city then it would not make much sense to spend time or resources within our database around constantly be checking the data within each table for value changes. Instead should any data need changing or deletion then a person should be overseeing the process.
Therefore we believe that time and resources would be better spent optimizing the speed for the request it would receive during its up time.

Indexing:

Considering the data from the datasets is mainly focused on where in the world it is coming from, we see a lot of data based on string values, primarily names. We can expect that a lot of data requests that our database would receive would be based on these names, so setting up indexing based on the different names of the organizations, cities, countries and even the account numbers, though they aren't strings, would speed up the request time. The same can also be done on each of the different report tables which would often accompany the cities. Here it would mainly be on the account numbers which could be grouped based on their account number to speed up the gathering of report data for requests.

We now have to consider when we will maintain the indexing within our database with respect to when new data could be uploaded and when it is uploaded. Considering the type of data within the datasets we can see that most cities uploaded yearly reports, with a few uploading more than once per year if the area of data gathering were different. Then we would consider the number of cities and countries that appear to be part of these reports. Overall we believe that indexing when new data is uploaded would be the best solution as the number of uploads is relatively few in comparison to how many requests there could appear should the database be used in a public scenario. As such we have kept the default settings for updating indexes in mongo.

Ten questions:

1. All C40/GCOM city:

- Gives a list of all cities where C40 and GCOM are both true

```
[
  {
    $match: {
      c40: true,
      gcom: true,
    },
  },
]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: 3429
city_name : "City of Stockholm"
city_short_name : "Stockholm"
country_id : ObjectId('65fb5d271c4f5a9ff509a08c')
c40 : true
gcom : true
city_location : "(59.3293235, 18.0685808)"
```

```
_id: 14088
city_name : "City of Oslo"
city_short_name : "Oslo"
country_id : ObjectId('65fb5d261c4f5a9ff509a070')
c40 : true
gcom : true
city_location : "(59.9138688, 10.7522454)"
```

```
_id: 31171
city_name : "Ayuntamiento de Madrid"
city_short_name : "Madrid"
country_id : ObjectId('65fb5d261c4f5a9ff509a06c')
c40 : true
gcom : true
```

2. Decrease/Increase by city - ghg:

- List of all cities with the increase/decrease based on the ghg table

```
[
  {
    $lookup: {
      from: "ghg",
      localField: "_id",
      foreignField: "city_id",
      as: "ghg_info",
    },
  },
  {
    $unwind: "$ghg_info",
  },
  {
    $match: {
      "ghg_info.increase_decrease": {
        $exists: true,
        $ne: null,
        $ne: NaN,
      },
    },
  },
  {
    $project: {
      city_name: "$city_name",
      "expected_increase/descrease":
        "$ghg_info.increase_decrease",
    },
  },
]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: 61753
city_name : "Yilan County"
expected_increase/descrease : "Stayed the same"
```

```
_id: 61790
city_name : "City of Emeryville, CA"
expected_increase/descrease : "Decreased"
```

```
_id: 1093
city_name : "City of Atlanta"
expected_increase/descrease : "Increased"
```

```
_id: 50679
city_name : "Barreiro"
expected_increase/descrease : "Increased"
```

```
_id: 1850
city_name : "Birmingham City Council"
expected_increase/descrease : "Decreased"
```

3. Percentage reduction target over ~80%

- List of all cities which targets are to reduce emission with over 80%

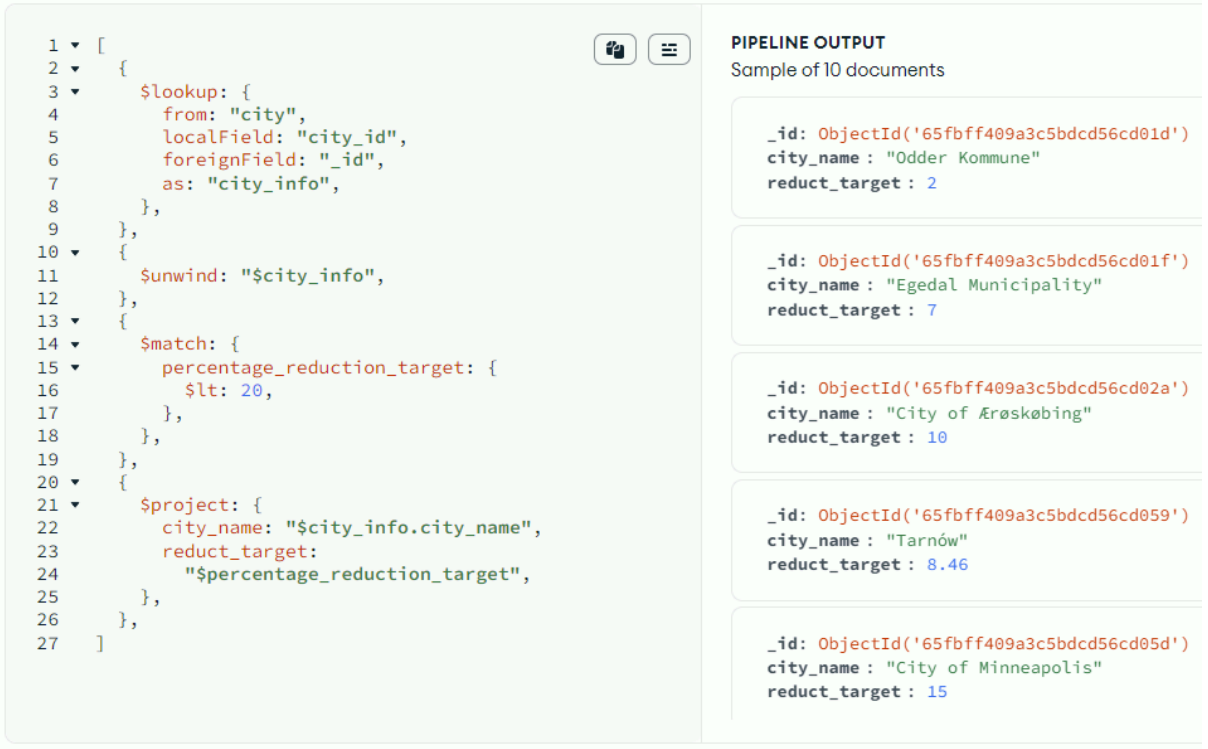
```
[
  {
    $lookup: {
      from: "city",
      localField: "city_id",
      foreignField: "_id",
      as: "city_info",
    },
  },
  {
    $unwind: "$city_info",
  },
  {
    $match: {
      percentage_reduction_target: {
        $gt: 80,
      },
    },
  },
  {
    $project: {
      city_name: "$city_info.city_name",
      reduct_target:
        "$percentage_reduction_target",
    },
  },
]
```

PIPELINE OUTPUT
Sample of 10 documents

```
_id: ObjectId('65fbff409a3c5bdcd56cd046')
city_name : "City of Stockholm"
reduct_target : 100

_id: ObjectId('65fbff409a3c5bdcd56cd04c')
city_name : "Hoeje-Taastrup Kommune"
reduct_target : 98

_id: ObjectId('65fbff409a3c5bdcd56cd04e')
city_name : "Landeshauptstadt Magdeburg"
reduct_target : 95

_id: ObjectId('65fbff409a3c5bdcd56cd04f')
city_name : "Landeshauptstadt Magdeburg"
reduct_target : 86

_id: ObjectId('65fbff409a3c5bdcd56cd053')
```

4. Percentage reduction target under ~20% including comment:

- List of all cities which targets are to reduce emission with less than 20%



```
[
  {
    $lookup: {
      from: "city",
      localField: "city_id",
      foreignField: "_id",
      as: "city_info",
    },
  },
  {
    $unwind: "$city_info",
  },
  {
    $match: {
      percentage_reduction_target: {
        $lt: 20,
      },
    },
  },
  {
    $project: {
      city_name: "$city_info.city_name",
      reduct_target:
        "$percentage_reduction_target",
    },
  },
]
```

PIPELINE OUTPUT
Sample of 10 documents

```
_id: ObjectId('65fbff409a3c5bdcd56cd01d')
city_name : "Odder Kommune"
reduct_target : 2

_id: ObjectId('65fbff409a3c5bdcd56cd01f')
city_name : "Egedal Municipality"
reduct_target : 7

_id: ObjectId('65fbff409a3c5bdcd56cd02a')
city_name : "City of Ærøskøbing"
reduct_target : 10

_id: ObjectId('65fbff409a3c5bdcd56cd059')
city_name : "Tarnów"
reduct_target : 8.46

_id: ObjectId('65fbff409a3c5bdcd56cd05d')
city_name : "City of Minneapolis"
reduct_target : 15
```

5. Baseline emission and percentage reduction target with comment:
    - List of all cities with their baseline year, reduction target with year and comment

```
[
  {
    $lookup: {
      from: "city",
      localField: "city_id",
      foreignField: "_id",
      as: "city_info",
    },
  },
  {
    $match: {
      target_date: {
        $exists: true,
        $ne: null,
        $ne: NaN,
      },
      comment: {
        $exists: true,
        $ne: null,
        $ne: NaN,
      },
    },
  },
  {
    $unwind: "$city_info",
  },
  {
    $project: {
      city_id: "$city_id",
      city_name: "$city_info.city_name",
      baseline_emissions: "$baseline_emissions",
      baseline_year: "$baseline_year",
      target_date: "$target_date",
      reduct_target:
        "$percentage_reduction_target",
      comment: "$comment",
    },
  },
]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: ObjectId('65fbff409a3c5bdcd56cd01f')
city_id : 62855
city_name : "Egedal Municipality"
baseline_emissions : 268000
baseline_year : "2009"
target_date : 2020
reduct_target : 7
comment : "The municipality are increasing in terms of citizens, business, buldin…"
```

```
_id: ObjectId('65fbff409a3c5bdcd56cd026')
city_id : 54498
city_name : "Ayuntamiento de Murcia"
baseline_emissions : 2141272
baseline_year : "2007"
target_date : 2020
reduct_target : 20
comment : "DATA IN CO2 (NOT CO2e)"
```

```
_id: ObjectId('65fbff409a3c5bdcd56cd027')
city_id : 1093
city_name : "City of Atlanta"
baseline_emissions : 800000
baseline_year : "2009"
target_date : 2020
reduct_target : 80
comment : "Reevaluation of recycling curbing and educational programs with public…"
```

```
_id: ObjectId('65fbff409a3c5bdcd56cd028')
city_id : 50679
city_name : "Barreiro"
baseline_emissions : 347987
baseline_year : "2008"
```

6. Decrease/Increase by city in city_community_wide_emissions with comment
    - List of all cities that shows whether they've increased or decreased as well as the comment explaining why the change is the way it is.

```
[
  {
    $lookup: {
      from: "cities_community_wide_emissions",
      localField: "_id",
      foreignField: "city_id",
      as: "city_wide_emissions_info",
    },
  },
  {
    $unwind: "$city_wide_emissions_info",
  },
  {
    $match: {
      "city_wide_emissions_info.increase_decrease":
        {
          $exists: true,
          $ne: null,
          $ne: NaN,
        },
      "city_wide_emissions_info.reason_for_increase":
        {
          $exists: true,
          $ne: null,
          $ne: NaN,
        },
    },
  },
  {
    $project: {
      city_name: "$city_name",
      increase_decrease:
        "$city_wide_emissions_info.increase_decrease",
      "increase/descrease_comment":
        "$city_wide_emissions_info.reason_for_increase",
    },
  },
]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: 54408
city_name : "Aarhus Kommune"
increase_decrease : "Decreased"
increase/descrease_comment : "Increasing share of renewable electricity at national level and contin…"
```

```
_id: 1093
city_name : "City of Atlanta"
increase_decrease : "Increased"
increase/descrease_comment : "Factors for the increase in emissions are: a growth in the constructio…"
```

```
_id: 50679
city_name : "Barreiro"
increase_decrease : "Increased"
increase/descrease_comment : "A residual increase of 2%."
```

```
_id: 58609
city_name : "City of Ærøskøbing"
increase_decrease : "Decreased"
increase/descrease_comment : "More renewable energi."
```

```
_id: 31052
city_name : "City of Cardiff"
increase_decrease : "Decreased"
increase/descrease_comment : "Main reduction in industry / commercial emissions but also some decrea…"
```

```
_id: 8242
city_name : "City of Helsinki"
increase_decrease : "Decreased"
increase/descrease_comment : "Energy efficiency has improved in buildings and vehicles. National gri…"
```

7. Risk and vulnerabilities with authors, confirmation and attachments
    - Shows each cities risk and vulnerabilities with a link to the attachments, as well as the author of the attachment and the confirmation for it

```
[
  {
    $lookup: {
      from: "city",
      localField: "city_id",
      foreignField: "_id",
      as: "city_info",
    },
  },
  {
    $unwind: "$city_info",
  },
  {
    $project: {
      city_name: "$city_info.city_name",
      author: "$author",
      confirmation: "$confirmation",
      attachment: "$attachment",
    },
  },
]
```

PIPELINE OUTPUT

Sample of 10 documents

```
_id: ObjectId('65fc0752086c3fba6dd7ad47')
city_name : "Prefeitura de Serra Talhada"
author : "Dedicated team within jurisdiction; Relevant department within jurisdi…"
confirmation : "The assessment can be accessed (unrestricted) on the link provided"
attachment : "https://drive.google.com/file/d/19DMxxK532IQSLt1PdwolFXZK4Ri2mRh-/view…"
```

```
_id: ObjectId('65fc0752086c3fba6dd7ad48')
city_name : "City of Shenzhen"
author : "Consultant; Dedicated team within jurisdiction; International organiza…"
confirmation : "The assessment can be accessed (unrestricted) on the link provided"
```

```
_id: ObjectId('65fc0752086c3fba6dd7ad49')
city_name : "Renca"
author : "Question not applicable"
confirmation : "The assessment has been attached"
attachment : "PLCC_Renca (08 Abril).pdf"
```

```
_id: ObjectId('65fc0752086c3fba6dd7ad4a')
city_name : "Municipalidad Distrital de Yura"
author : "Dedicated team within jurisdiction"
confirmation : "The assessment can be accessed (unrestricted) on the link provided"
attachment : "https://sigrid.cenepred.gob.pe/sigridv3/documento/15166/descargar"
```

```
_id: ObjectId('65fc0752086c3fba6dd7ad4b')
city_name : "Trelleborg Municipality"
author : "Dedicated team within jurisdiction"
confirmation : "The assessment can be accessed (unrestricted) on the link provided"
attachment : "https://moten.trelleborg.se/welcome-sv/namnder-styrelser/kommunstyrels…"
```

Scaling the database:

Formulation

In the process of designing and implementing a database for scaling, we chose to closely adhere to the principles of both CAP theorem and to a certain extent acid. While MongoDBs cloud service supports replication (and possibly sharding) we have chosen to design our own local database for scaling.

The database consists of 3 shards with 3 replica sets shared within them. We initiated three replica sets. The config server replica set are stored in the folders /db/configsvr* . By allowing the shard to share the same replica set allows us to simplify the setup, and conserve resources, especially where the data distribution and access patterns don't necessarily require separate replica sets for each shard. This setup also allows the system to continue to operate without data loss, adhering to the ACID durability feature.

To address scalability, 3 shards were created. The shards are stored within the folders /db/shard1svr*. Sharding allows us to distribute data across multiple servers, allowing the system to scale and handle large data sets and read/write operations.

While this design may not be needed now, this model is designed to support scalability if demands are needed. The setup also considers fault tolerance and data consistency across the distributed system, aligning with the strengths of mongoDB in managing high volume data and traffic.

ACID:

MongoDB also ensures that any write operation is atomic. This means that with a single document, the database guarantees that all changes are applied, otherwise none are. This strongly adheres to the CAP atomicity principle. This means we have strong consistency over our replicas.

While MongoDB can normally be isolated, as in the use of transactions in replica sets can allow for multiple document operations to be executed from each other. Since some collections are dependent on each other particularly through reference ids, the correct dependent documents have to be inserted with their ids to keep consistency.

The use of replicas also significantly increases durability. If a primary node fails the data will remain accessible though other nodes, ensuring write is acknowledged, it is preserved despite failures.

CAP:

Consistency has to be balanced with the other cap principles. The design prioritizes eventual consistency over shards and replicas to maintain high system performance and availability.

Availability is improved significantly with replica sets, allowing the database to remain operational and accessible if a node fails.

Sharding follows the Cap theorem’s partition tolerance and scalability feature, allowing for more efficient data management and querying performance.

Validate and test database operations:

First it is necessary to check whether all the provided data has been uploaded correctly to the database. This is done by doing a simple countDocuments() function on each collection. As shown in the picture to the right, each collection has been filled with data as desired. The collection with the most entries is the risk_and_vulnerabilities, while the one with the least is the countries. This seems accurate and the data appears to be uploaded as intended, without a noticeable loss of data.

```
> db.risk_and_vulnerability.countDocuments()
< 1370
> db.city.countDocuments()
< 868
> db.ghg.countDocuments()
< 187
> db.country.countDocuments()
< 94
> db.cities_community_wide_emissions.countDocuments()
< 229
> db.emissions_reduction.countDocuments()
< 686
```

It is then beneficial to make sure the common operations work by running a few tests. The first test is the “find()” function which lists all the entries that each fulfill the attributes listed. In this case, the name of the city has been provided as “City of Lisbon”. The find function therefore returns Lisbon with all its corresponding data as shown in the picture to the right.
If it is desirable to only find one instance, the closely related function “findOne()” would be preferable instead.

```
> db.city.find({
    city_name: "City of Lisbon"
})
< {
  _id: ObjectId('660c6365218e2ac2c785cbbd'),
  city_name: 'City of Lisbon',
  city_short_name: 'Lisbon',
  country_id: ObjectId('65fb5d261c4f5a9ff509a072'),
  account_no: 36159,
  c40: false,
  gcom: true,
  city_location: '(38.7222524, -9.1393366)',
  extra: 'Empty DataFrame\n' +
    'Columns: [gdp, gdp_currency, year, source, temperature, population]\n' +
    'Index: []'
}
```

Another operation that is worth testing is the insert() function. This function won’t be used much in this project, but it is still beneficial to make sure it works, since an error would suggest underlying issues with the database.

```
> db.city.insertOne({ city_name: "Magical city of Narnia", city_short_name: "Narnia", account_no: 123 });
< {
    acknowledged: true,
    insertedId: ObjectId('660db384f169290e77c7a8b4')
  }
```

This function inserted the city as expected, which implies everything is the way it should be. To make sure it was actually created, another count is run to see the number has incremented by one compared to the start.

```
> db.city.count()
< 869
```

There’s now 869 cities, while there originally was 868, which means the insert worked perfectly
To make sure there isn’t any faulty or made up data, and to test another common operation, this newly created city will be deleted.

```
> db.city.deleteOne({ city_name: "Magical city of Narnia" });
< {
    acknowledged: true,
    deletedCount: 1
  }
```

Once again, we will check how many total cities, to see whether the city was actually removed.

```
> db.city.count()
< 868
```

The amount of cities is now back to 868 which was the original amount and the data is now back to the way it's supposed to be, and it is now known that common operations work on the database.

Evaluate the database’s performance and suggest measures for improving it.

The general approach of designing our database has been around the “city” collection. Part of the reason for this is so the user can easily query through the city. We also make use of a lot of object references opposed to

To measure the performance of execution speeds there are two methods, using a profiler and the .explain() function.

Profilers can only be used either in a premium cluster or run locally. To run it locally we used mongodump to create a copy of the cluster.

Query:	Code	Time (ms):

Gives a list of all cities where C40 and GCOM are both true	db.getCollection("city").explain("executionStats").aggregate([
	{
	$match: {
	c40: true,
	gcom: true,
	},
	},
	])
		3

List of all cities with the increase/decrease based on the ghg table	db.getCollection("city").explain("executionStats").aggregate([
	{
	$lookup: {
	from: "ghg",
	localField: "_id",
	foreignField: "city_id",
	as: "ghg_info",
	},
	},
	{
	$unwind: "$ghg_info",
	},
	{
	$match: {
	"ghg_info.increase_decrease": {
	$exists: true,
	$ne: null,
	$ne: NaN,
	},
	},
	},
	{
	$project: {
		138

	city_name: "$city_name", "expected_increase/descrease": "$ghg_info.increase_decrease", }, },])	

List of all cities which targets are to reduce emission with over 80%	db.getCollection("city").explain("execution Stats").aggregate([
	{
	$lookup: {
	from: "city",
	localField: "city_id",
	foreignField: "_id",
	as: "city_info",
	},
	},
	{
	$unwind: "$city_info",
	},
	{
	$match: {
	percentage_reduction_target: {
	$gt: 80,
	},
	},
	},
	{
	$project: {
	city_name: "$city_info.city_name",
	reduct_target:
	"$percentage_reduction_target",
	},
	},
	]
	)
		6

List of all cities which targets are to reduce emission with less than 20%	db.getCollection("city").explain("execution Stats").aggregate([[
	{
	$lookup: {
	from: "city",
	localField: "city_id",
	foreignField: "_id",
	as: "city_info",
	},
	},
	{
	$unwind: "$city_info",
	},
	{
	$match: {
	percentage_reduction_target: {
	$lt: 20,
	},
	},
	},
	{
	$project: {
	city_name: "$city_info.city_name",
	reduct_target:
	"$percentage_reduction_target",
	},
	},
	])
		18

List of all cities with their baseline year, reduction target with year and comment	db.getCollection("emissions_reduction").explain("executionStats").aggregate([
	{
	$lookup: {
	from: "city",
	localField: "city_id",
	foreignField: "_id",
	as: "city_info",
	},
	},
	{
	$match: {
	target_date: {
	$exists: true,
	$ne: null,
	$ne: NaN,
	},
	comment: {
	$exists: true,
	$ne: null,
	$ne: NaN,
	},
	},
	},
	{
		21

	$unwind: "$city_info",
	},
	{
	$project: {
	city_id: "$city_id",
	city_name: "$city_info.city_name",
	baseline_emissions:
	"$baseline_emissions",
	baseline_year: "$baseline_year",
	target_date: "$target_date",
	reduct_target:
	"$percentage_reduction_target",
	comment: "$comment",
	},
	},
	]);	

List of all cities that shows whether they've increased or decreased as well as the comment explaining why the change is the way it is.	db.getCollection("city").explain("execution Stats").aggregate([
	{
	$lookup: {
	from:
	"cities_community_wide_emissions",
	localField: "_id",
	foreignField: "city_id",
	as: "city_wide_emissions_info",
	},
	},
	{
	$unwind: "$city_wide_emissions_info",
	},
	{
	$match: {
	
	"city_wide_emissions_info.increase_decrease":
	{
	$exists: true,
	$ne: null,
	$ne: NaN,
	},
	
	"city_wide_emissions_info.reason_for_increase":
	{
	$exists: true,
	$ne: null,
	$ne: NaN,
	},
	},
	},
	{
	$project: {
		157

	city_name: "$city_name", increase_decrease: "$city_wide_emissions_info.increase_decrease", "increase/descrease_comment": "$city_wide_emissions_info.reason_for_increase", }, },]);	

Shows each cities risk and vulnerabilities with a link to the attachments, as well as the author of the attachment and the confirmation for it	db.getCollection("risk_and_vulnerability").explain("executionStats").aggregate([
	{
	$lookup: {
	from: "city",
	localField: "city_id",
	foreignField: "_id",
	as: "city_info",
	},
	},
	{
	$unwind: "$city_info",
	},
	{
	$project: {
	city_name: "$city_info.city_name",
	author: "$author",
	confirmation: "$confirmation",
	attachment: "$attachment",
	},
	},
	]);
		98

The database can be improved upon in several ways to improve performance. By using references instead of embedded objects. This potentially removes a lot of duplication of items, which in turn means when updating items we do not need to update its duplicates. While this feature is useful, it is only useful in a high write/ratio ratio when and for this project we believe that updates and creating will occur infrequently. To optimize read speed, we could implement indexes. This will help us speed up specific queries. As the database performs relatively well at the moment, it is inefficient to implement an index, but we could add an index for each of the report collections (emissions_reduction, ghg, risk_and_vulnerability). As indexes essentially create duplicates, it slows down write times and increases maintanece.

Conclusion:

We have gone through the five datasets given to us and from those have explored the data they contained, had discussions and made our conclusions as to how we best could store the data within a database. From our conclusions regarding the datasets we have discussed our options for setting up a database where our conclusion is the result of our work, with a EER diagram of the structure of the database, we used it to guide our upload scripts for getting the data from the five datasets into the database instance which unlike our previous scripts for our SQL database, the uploads require no existing structure to be made before we could upload.
From there we have made discussions, evaluations and conclusions for future improvement recommendations that could be made to our database as well as some improvements which we have implemented into our database.