Wildlife Observation & Conservation Data Warehouse

Anson Wu
awu1054@floridapoly.edu

Knoelle Grassi
kgrassi2693@floridapoly.edu

Kouame Jean Pierre Koffi
kkoffi4206@floridapoly.edu

Data Warehousing

CAP3774

Professor Ray Ready

Introduction

Narrative Description

The goal of this data warehouse project is to design and implement a centralized platform for storing and analyzing biodiversity and conservation data. The data warehouse integrates various datasets to support long-term environmental monitoring, conservation decision-making, and ecological research. The focus is on capturing detailed information about animal species across time and space, including their observed sightings, surrounding environmental conditions, and conservation status.

The business problem being addressed stems from the fragmented nature of ecological data. Field observations of species are often stored separately from environmental data, global trade reports, and conservation status records. This separation limits the ability of researchers, conservation organizations, and environmental policymakers to make informed, data-driven decisions. By building a data warehouse that brings together spatiotemporal animal observations, environmental metrics, CITES trade records, and IUCN Red List status history, this project enables a unified, analytical view of the health and risks associated with global biodiversity.

Business Context

The data warehouse is modeled after the needs of organizations involved in wildlife conservation, environmental monitoring, and biodiversity research. These include:

- Government agencies (e.g., wildlife and natural resource departments)
- Nonprofit conservation organizations (e.g., WWF, IUCN)
- Research institutions and ecological data science teams

Their shared objective is to monitor species health, detect signs of ecosystem stress, evaluate conservation outcomes, and comply with international wildlife regulations (like CITES). A unified data warehouse supports:

- Periodic reporting on the conservation status of species
- Trend analysis of animal populations and their habitats
- Integration of trade data to identify links between legal/illegal species trade and species decline

Source Data Description

The core dataset consists of spatiotemporal wildlife observations, which include detailed records of species sightings. These observations capture the taxonomic classification (such as family, genus, species), observation time and date, and the location where the species was recorded. These sightings are further enriched with local weather data (temperature, humidity, and precipitation) and soil conditions at the time and location of the observation, providing a comprehensive environmental context.

Additional datasets supplement this core:

- CITES trade data identifies the import and export of species around the world, documenting trade volume under international wildlife trade regulations.
- The IUCN Red List categorization adds a conservation layer, specifying the risk of extinction for each species (e.g., Vulnerable, Endangered, Critically Endangered), along with historical records indicating when a species entered and exited a particular threat category.

Together, these data sources support a multidimensional analysis of species health and conservation status over time and space. This is valuable for:

- Detecting population decline or recovery trends
- Correlating environmental conditions with species sightings
- Understanding the impact of legal trade on species health
- Monitoring changes in extinction risk across time periods

BEAM Template Documentation

Dimensions

Date

date	has date key	full date	time	sunrise	sunset	season

Location

coordinates	has location key	latitude	longitude	positional accuracy	time zone	elevation	terrestial or aquatic

Weather

weather id	has weather key	temperature	cloudcover	shortwave radiation	direct radiation	diffuse radiation	wind speed	wind direction	wind gusts	vapor pressure deficit

Soil

soil id	has soil key	soil temp 0-7m	soil temp 7-28m	soil temp 28-100m	soil moisture 0-7m	soil moisture 7-28m	soil moisture 28-100m

Species

taxon ID	has species key	scientific name	phylum	class	order	family	genus	species	common name	redlist category

Trade

trade id	has trade key	appendix	importer	exporter	term	unit	purpose	source	purpose Description	source description

Facts

Species Observatio TF

observation	of species	on date	at location	taxon geoprivacy	license	image url
[what]	[who]	[when]	[where]			

Spatiotemporal TF

environment	on date	at location	with weather	on soil	geoprivacy	evaportranspiration
[what]	[when]	[where]	[what]	[what]		

Trading	TF				
trade	by trader	of species	origin	importer reported quantity	exporter reported quanitity
[what]	[how]	[who]			

Conservation Status	AS									
red list	of species	was last least concern	was last near threatened	was last vulnerable	was last endangered	was last critically endangered	was last extinct in wile	was last recovered	became current category	current category
[what]	[who]	[when]	[when]	[when]	[when]	[when]	[when]	[when]	[when]	

Monthly Weather	PS											
monthly weather	has period	of species	at location	total sightings	average temp	total precipitation	average humidity	total rain	total snow	average cloud cover	average wind speed	average wind direction	average wind gusts

Data Warehouse Schema

Dimensions

LAPTOP-OULQLQJV....DW - dbo.DimSoil

Column Name	Data Type	Allow Nulls
soilKey	int	☐
soilID	int	☑
soilTemp0To7	real	☑
soilTemp7To28	real	☑
soilTemp28To100	real	☑
soilMoisture0To7	real	☑
soilMoisture7To28	real	☑
soilMoisture28To100	real	☑
		☐

Soil Dimension

The original data source provided information about the soil in every observation, so a soil dimension was needed to provide a space for that information.

LAPTOP-OULQLQJV...- dbo.DimWeather

Column Name	Data Type	Allow Nulls
weatherKey	int	☐
weatherID	int	☑
temp	real	☑
relativeHumidity	real	☑
dewpoint	real	☑
apparentTem	real	☑
surfacePressure	real	☑
precipitation	real	☑
rain	real	☑
snowfall	real	☑
cloudcover	real	☑
shortwaveRadiation	real	☑
directRadiation	real	☑
diffuseRadiation	real	☑
windSpeed	real	☑
windDirection	real	☑
windGusts	real	☑
vaporPressureDeficit	real	☑
		☐

Weather Dimension

(apparentTem was a typo that was corrected to be apparentTemp)

The original data source also provided information about the weather. It also describes the conditions of the species, so a weather dimension was provided to provide a space for that information and to describe what the environment was like for that species. Some

columns were repeated for different conditions so some of the original information about the weather was excluded. The current columns describe what was going on in the weather.

LAPTOP-OULQLQJV....W - dbo.DimTrade

Column Name	Data Type	Allow Nulls
tradeKey	int	☐
tradeID	int	☑
app	varchar(50)	☑
importerCountry	varchar(50)	☑
exporterCountry	varchar(50)	☑
term	varchar(50)	☑
unit	varchar(50)	☑
purpose	varchar(50)	☑
source	varchar(50)	☑
purposeDescript	nvarchar(1000)	☑
sourceDescript	nvarchar(100)	☑
		☐

Trade Dimension

A big part species is the trading of them. The CITES data source was found to show that and this table is made for a space for that information to be stored. It shows the countries involved and why the species is being traded.

LAPTOP-OULQLQJV....dbo.DimLocation

Column Name	Data Type	Allow Nulls
locationKey	int	☐
coordinates	nvarchar(100)	☑
latitude	real	☑
longitude	real	☑
positionalAccuracy	real	☑
timeZone	varchar(50)	☑
elevation	real	☑
terrestialOrAquatic	nvarchar(10)	☑
		☐

Location Dimension

The way to tell apart the species is by the location and date, so the location dimension was needed as an identifier and also to show the specifics of each species observation and spatiotemporal observation. Most of the columns are just what was given in the original data source that we decided to keep.

LAPTOP-OULQLQJV...DW - dbo.DimDate

Column Name	Data Type	Allow Nulls
dateKey	int	☐
date	datetime	☑
fulldate	varchar(50)	☑
time	varchar(50)	☑
sunrise	varchar(50)	☑
sunset	varchar(50)	☑
season	nvarchar(10)	☑
		☐

Date Dimension

Date is needed for every information such as when the species was traded, when the observation was, and when the species changed categories. The original data source provided information about the sunrise and sunset so that was included, and the season is important for trades and observations about the species.

LAPTOP-OULQLQJV....- dbo.DimSpecies*

Column Name	Data Type	Allow Nulls
speciesKey	int	☐
taxonID	varchar(50)	☑
scientificName	varchar(50)	☑
kingdomName	varchar(50)	☑
phylumName	varchar(50)	☑
className	varchar(50)	☑
orderName	varchar(50)	☑
familyName	varchar(50)	☑
genusName	varchar(50)	☑
speciesName	varchar(50)	☑
commonName	varchar(50)	☑
redlistCategory	varchar(50)	☑
		☐

Species Dimension

These are all the taxonomic categories of the species along with what it's commonly known as and what category it is. Taxonomy describes the species, the common name tells us what it is, and the redlist category describes the endangerment of the species.

Fact Tables

LAPTOP-OULQLQJV.....FactObservations | SQLQuery22.sql - (...ULQLQ

Column Name	Data Type	Allow Nulls
observationID	int	☐
dateKey	int	☐
locationKey	int	☐
speciesKey	int	☐
taxonGeoprivacy	varchar(50)	☑
license	varchar(50)	☑
imageURL	varchar(100)	☑
		☐

Transactional Observation Fact using the date, location, and species dimensions. It describes the location, time, and what species are being observed as the original data source does.

LAPTOP-OULQLQJV....actSpatiotemporal | SQLQuery24.sql - (...ULQ

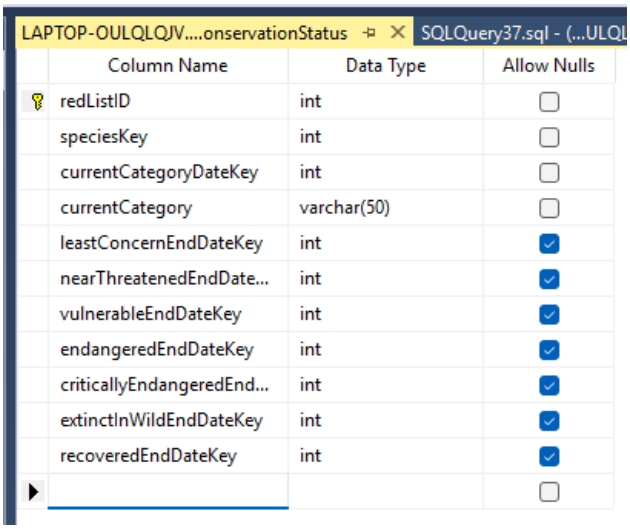
Column Name	Data Type	Allow Nulls
environmentID	int	☐
dateKey	int	☐
locationKey	int	☐
weatherKey	int	☐
soilKey	int	☐
geoprivacy	varchar(50)	☑
etoFaoEvapotranspiration	varchar(50)	☑
		☐

Transactional Spatiotemporal Fact Table using the date, location, weather, and soil dimensions. It describes what was going on in the environment at a specific location on a specific date.

LAPTOP-OULQLQJV....- dbo.FactTrading | SQLQuery25.sql - (...ULQL

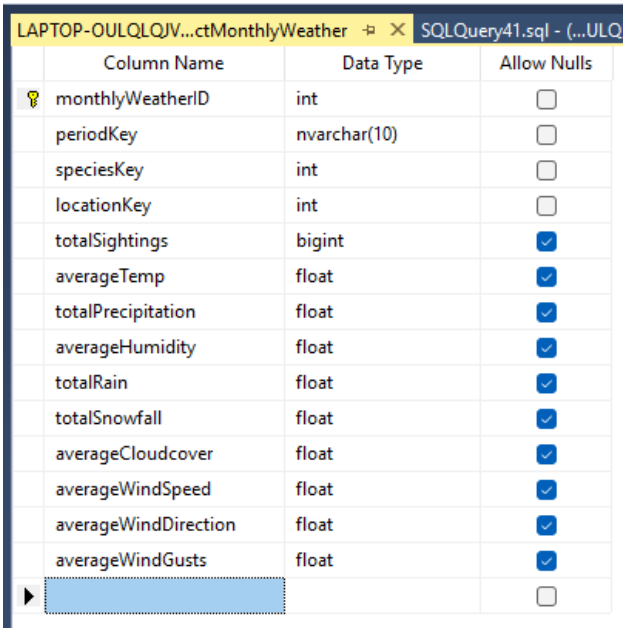
Column Name	Data Type	Allow Nulls
tradingID	int	☐
tradeKey	int	☐
tradedSpeciesKey	int	☐
origin	varchar(50)	☑
importerReportedQuantity	int	☑
exporterReportedQuantity	int	☑
		☐

Transactional Trading Fact Table using the trade and species dimensions. It describes a specific trade of a species.


LAPTOP-OULQLQJV....onservationStatus | SQLQuery37.sql - (...ULQ

Column Name	Data Type	Allow Nulls
redListID	int	☐
speciesKey	int	☐
currentCategoryDateKey	int	☐
currentCategory	varchar(50)	☐
leastConcernEndDateKey	int	☑
nearThreatenedEndDate...	int	☑
vulnerableEndDateKey	int	☑
endangeredEndDateKey	int	☑
criticallyEndangeredEnd...	int	☑
extinctInWildEndDateKey	int	☑
recoveredEndDateKey	int	☑
		☐

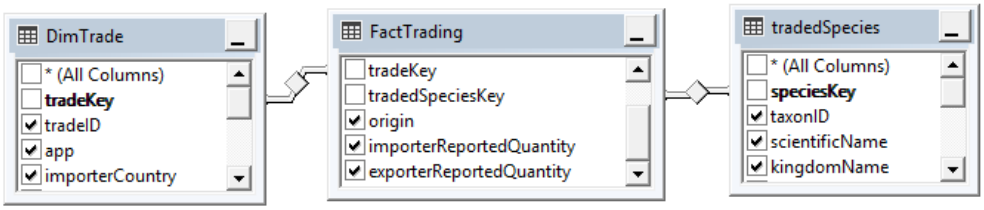
Accumulative Snapshot Conservation Status Fact Table using the species and 8 date dimension tables. It shows the conservation history of a species and gives tracking for if the species is recovering or getting worse.


LAPTOP-OULQLQJV...ctMonthlyWeather | SQLQuery41.sql - (...ULQ

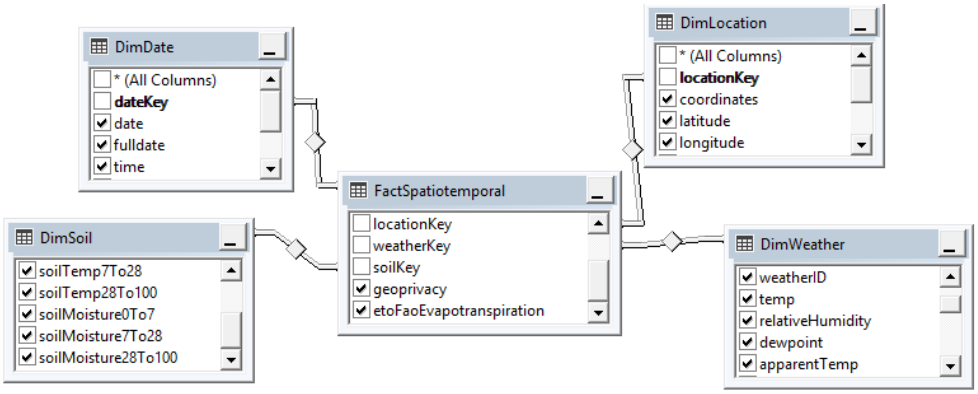
Column Name	Data Type	Allow Nulls
monthlyWeatherID	int	☐
periodKey	nvarchar(10)	☐
speciesKey	int	☐
locationKey	int	☐
totalSightings	bigint	☑
averageTemp	float	☑
totalPrecipitation	float	☑
averageHumidity	float	☑
totalRain	float	☑
totalSnowfall	float	☑
averageCloudcover	float	☑
averageWindSpeed	float	☑
averageWindDirection	float	☑
averageWindGusts	float	☑
		☐

Periodic Snapshot Monthly Weather Fact Table using the species and location keys. It gives a wider view of a period of weather for a species in a specific area. It shows what that species is experiencing.

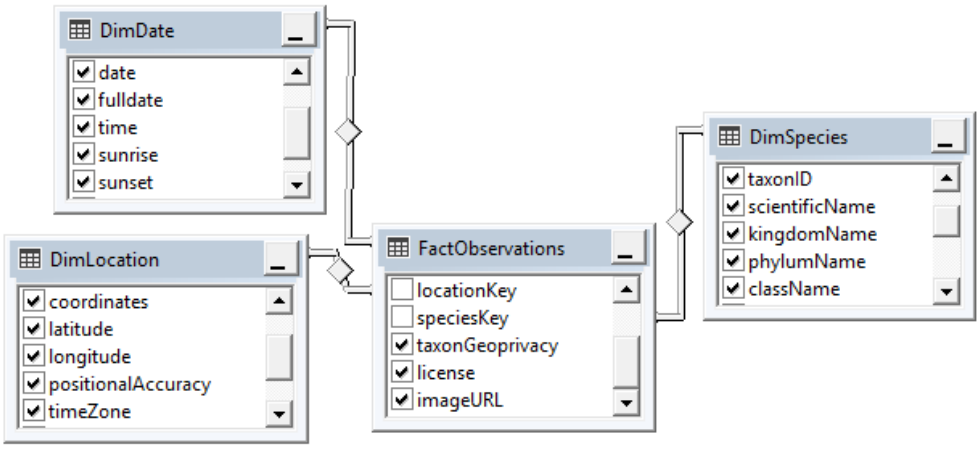
Views



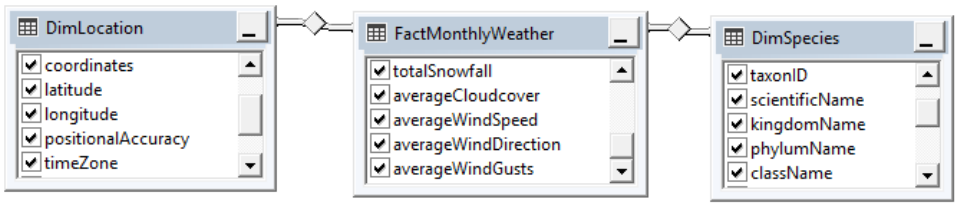
Trading Mart contains the data for species trades. It uses the Trading Fact, Species Dimension, and Trade Dimension.



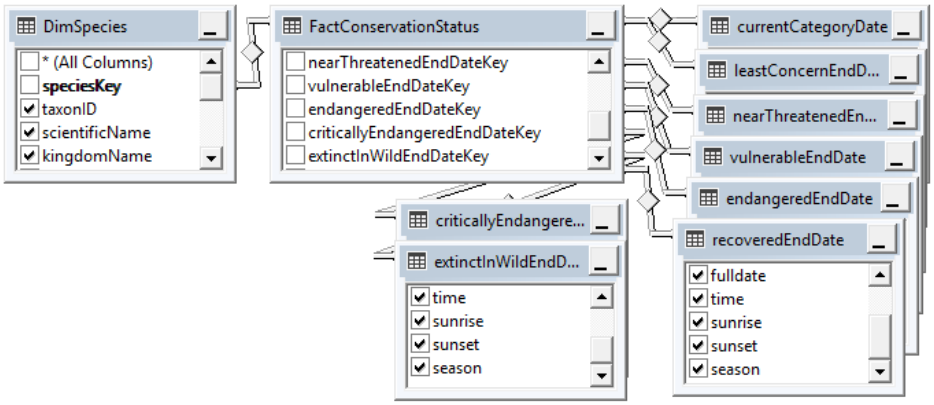
Spatiotemporal Mart contains the data for the environment at a location on a date. It uses the Spatiotemporal Fact, Date Dimension, Soil Dimension, Location Dimension, and Weather Dimension.



Observation Mart contains the data for the observations on the animals. It uses the Observation Fact, Date Dimension, Location Dimension, and Species Dimension.



Monthly Weather Mart contains the data for weather over a month period. It uses the Monthly Weather Fact, Species Fact, and Location Fact.



Conservation Status Mart contains data about a species' conservation. It uses the Conservation Status Fact, Species Dimension, and Date Dimensions.

Queries

```
SELECT scientificName, year(date) AS ObservationYear, COUNT(*) AS totalSightings
FROM FactObservations JOIN DimSpecies
on DimSpecies.speciesKey = FactObservations.speciesKey
JOIN DimDate on FactObservations.dateKey = DimDate.dateKey
GROUP BY scientificName, YEAR(date)
ORDER BY totalSightings DESC;
```

	scientificName	ObservationYear	totalSightings
1	Lynx rufus	2022	358
2	Lynx rufus	2021	318
3	Lynx rufus	2020	305
4	Lynx rufus	2019	228
5	Loxodonta africana	2022	177
6	Loxodonta africana	2019	135
7	Lynx rufus	2018	134
8	Loxodonta africana	2018	116
9	Loxodonta africana	2017	110
10	Lynx rufus	2017	99
11	Puma concolor	2020	89
12	Loxodonta africana	2021	86
13	Lynx rufus	2023	86
14	Loxodonta africana	2015	77
15	Loxodonta africana	2016	73
16	Puma concolor	2022	73

Shows the yearly number of sightings for each species.

```
SELECT scientificName, ROUND(AVG(temp), 2) AS avgTemperature
FROM FactObservations JOIN DimSpecies
ON DimSpecies.speciesKey = FactObservations.speciesKey
JOIN FactSpatiotemporal ON FactSpatiotemporal.dateKey = FactObservations.DateKey
AND FactSpatiotemporal.locationKey = FactObservations.locationKey
JOIN DimWeather on DimWeather.weatherKey = FactSpatiotemporal.weatherKey
GROUP BY scientificName
ORDER BY avgTemperature DESC;
```

	scientificName	avgTemperature
1	Caracal caracal nubicus	34.6
2	Acinonyx jubatus hecki	33.98
3	Loxodonta	33.35
4	Leptailurus serval	29.82
5	Panthera pardus kotiya	28.96
6	Panthera leo leo	28.91
7	Panthera tigris tigris	28.67
8	Leptailurus serval constantina	28.02
9	Prionailurus bengalensis bengalensis	27.9
10	Elephas maximus maximus	27.75
11	Leopardus braccatus	27.7
12	Prionailurus javanensis	27.6
13	Elephas maximus sumatranus	27.5
14	Loxodonta cyclotis	27.37
15	Felis lybica lybica	27.36
16	Elephas maximus borneensis	26.51

Shows the average temperature each species experiences. The location was needed to link the Spatiotemporal and Observation fact tables.

```
SELECT timeZone, scientificName, COUNT(*) AS totalSightings
FROM FactObservations
JOIN DimSpecies ON FactObservations.speciesKey = DimSpecies.speciesKey
JOIN DimLocation ON FactObservations.locationKey = DimLocation.locationKey
GROUP BY timeZone, scientificName
ORDER BY timeZone, totalSightings DESC;
```

	timeZone	scientificName	totalSightings
1	Africa/Accra	Loxodonta cyclotis	2
2	Africa/Accra	Loxodonta africana	1
3	Africa/Addis_Ababa	Caracal caracal	1
4	Africa/Bangui	Loxodonta cyclotis	5
5	Africa/Bangui	Panthera leo leo	1
6	Africa/Bangui	Leptailurus serval constantina	1
7	Africa/Blantyre	Loxodonta africana	7
8	Africa/Blantyre	Panthera leo melanochaita	6
9	Africa/Blantyre	Acinonyx jubatus jubatus	2
10	Africa/Blantyre	Leptailurus serval serval	1
11	Africa/Brazzaville	Loxodonta cyclotis	6
12	Africa/Brazzaville	Caracal aurata	1
13	Africa/Casablanca	Felis lybica lybica	1
14	Africa/Dakar	Panthera leo leo	1
15	Africa/Dar_es_Salaam	Loxodonta africana	215
16	Africa/Dar_es_Salaam	Panthera leo melanochaita	177

Shows the number of sightings for each animal in a specific area. The specific country wasn't given so the time zone was used to separate species in different areas.

```
SELECT DATENAME(month, date) AS MonthName, COUNT(*) AS totalSightings
FROM FactObservations JOIN DimLocation ON FactObservations.locationKey=DimLocation.loca
JOIN DimDate ON FactObservations.dateKey = DimDate.dateKey
GROUP BY DATENAME(month, date)
ORDER BY DATENAME(month, date);
```

	MonthName	totalSightings
1	April	367
2	August	536
3	December	445
4	February	465
5	January	525
6	July	447
7	June	370
8	March	456
9	May	352
10	November	428
11	October	434
12	September	452

Shows the total sightings in each month of the year.

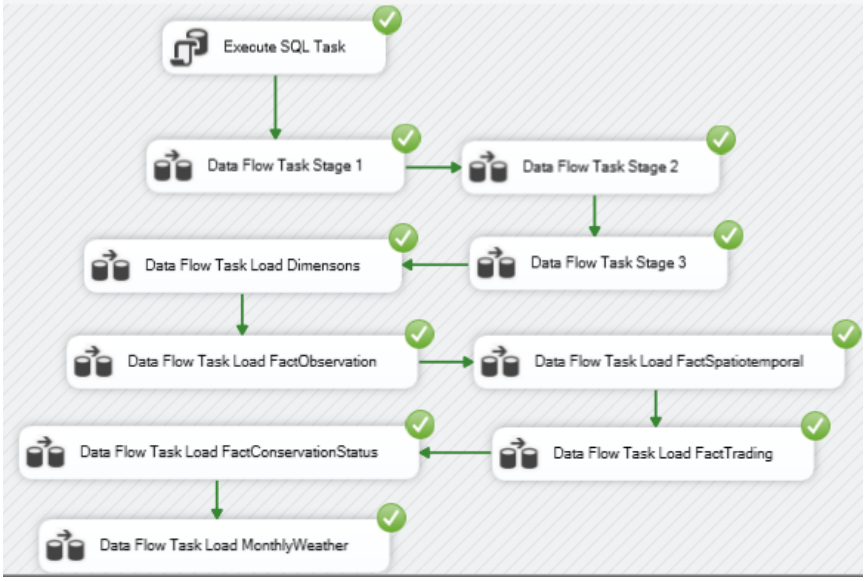
```
SELECT timeZone, YEAR(date) AS ObservationYear, round(SUM(rain), 2) AS totalRain
FROM FactSpatiotemporal JOIN DimLocation ON FactSpatiotemporal.locationKey = DimLocatic
JOIN DimWeather ON FactSpatiotemporal.weatherKey = DimWeather.weatherKey
JOIN DimDate ON FactSpatiotemporal.dateKey = DimDate.dateKey
GROUP BY timeZone, YEAR(date)
ORDER BY YEAR(date), timeZone;
```

	timeZone	ObservationYear	totalRain
1	America/Edmonton	1975	0
2	Africa/Dar_es_Salaam	1982	0
3	Africa/Dar_es_Salaam	1984	0
4	Africa/Nairobi	1984	0
5	Africa/Nairobi	1985	0.1
6	Africa/Harare	1987	0
7	Africa/Dar_es_Salaam	1988	0
8	Africa/Gaborone	1992	0
9	Africa/Nairobi	1992	0
10	Africa/Dar_es_Salaam	1993	0
11	Africa/Dar_es_Salaam	1994	0
12	Africa/Nairobi	1994	0
13	Africa/Dar_es_Salaam	1996	0
14	Africa/Windhoek	1998	0
15	Africa/Nairobi	1999	0.7
16	Asia/Kolkata	1999	0
17	Africa/Dar_es_Salaam	2000	0.7

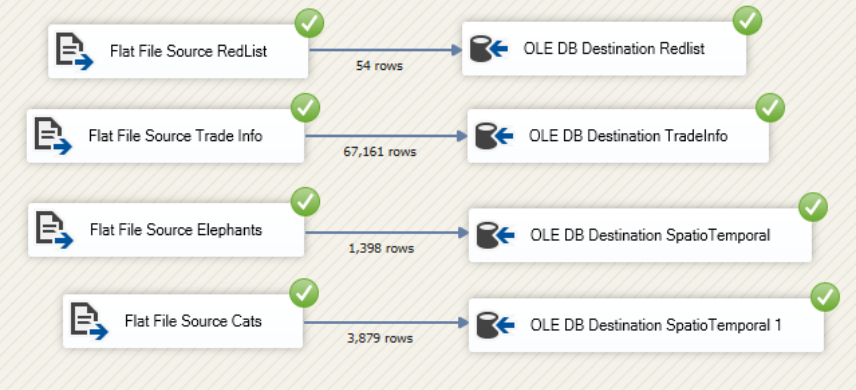
Shows the yearly rain in each time zone.

ETL

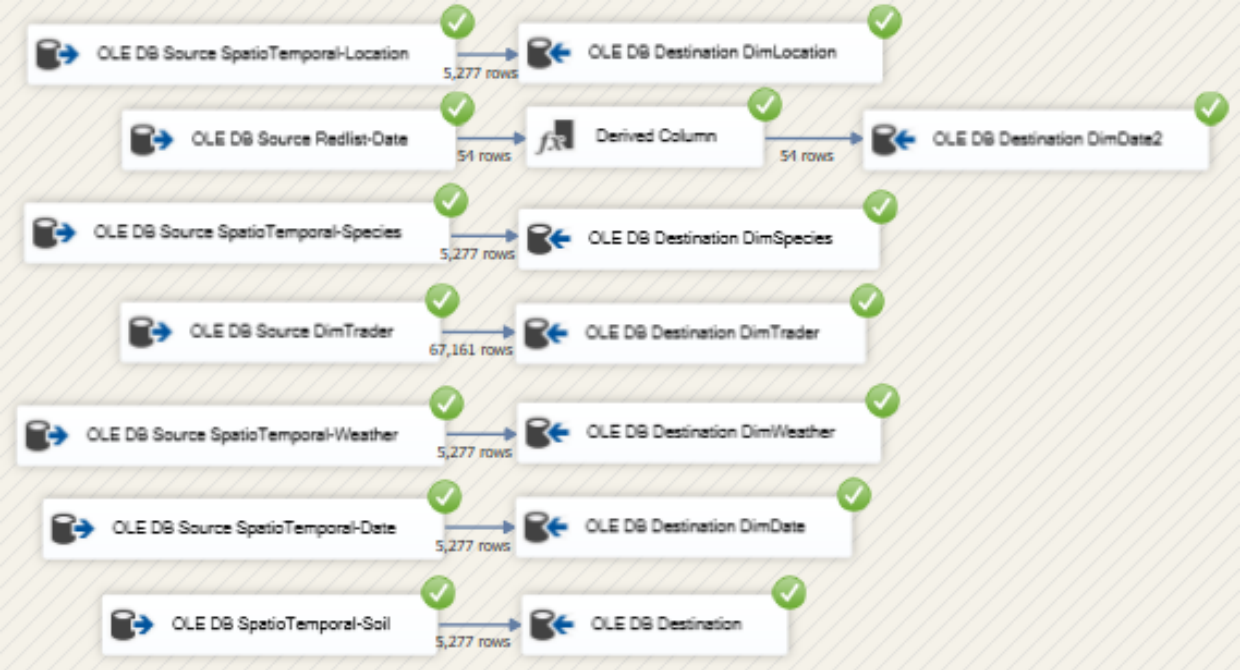
This is the total ETL process showing the successful completion of all the steps.

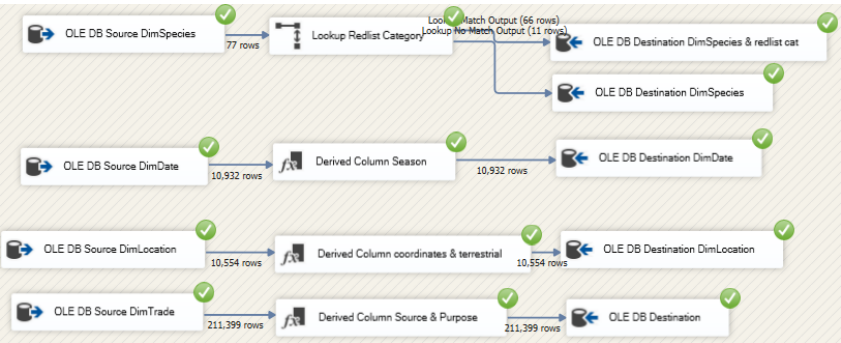


This is stage 1 with all the files being loaded into a stage 1 table.



This is stage 2 with the data in the tables being put into its specific table. The date dimension is also converting the values that should be null back to null being it's inserted into the stage 2 table.

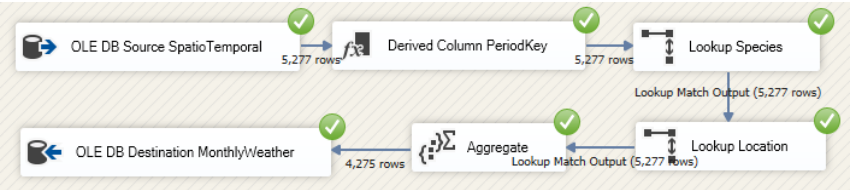




This is stage 4 where the data in the tables is edited to reach the final state before being inserted into the dimension table. Species are being matched with its redlist category, and the others are editing and adding derived columns.

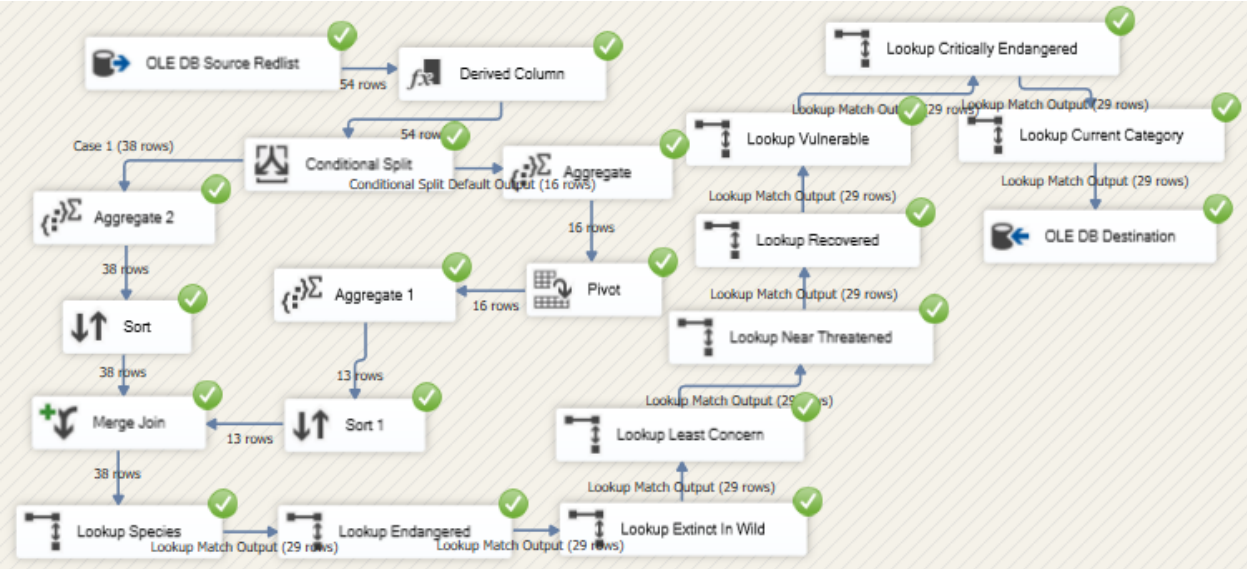
The data is being loaded into the dimensions and the slowly changing dimensions are being added.

The following are loading the fact tables:



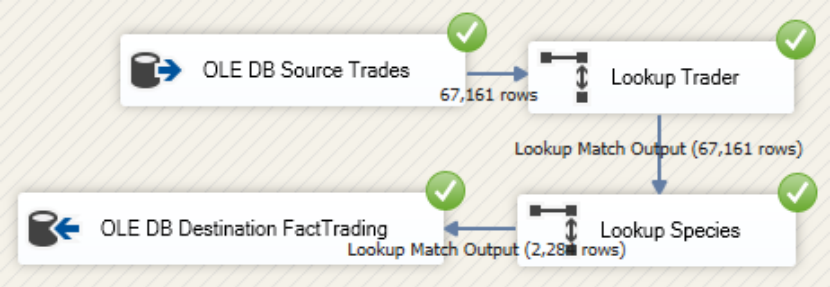
Loading the Monthly Weather fact table (periodic snapshot). It derives a period key, finds

the matching species and location, before summing up the data for that period.

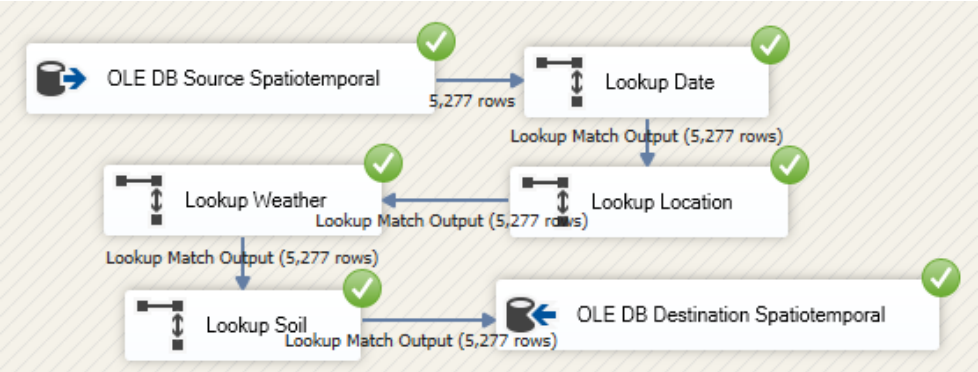


Loading the Conservation Fact Table (Accumulative Snapshot). It's modified to get the required form, each category being its own column, and the dates when each species was last in that category was added to each past category. The current category was found along with the date it became that category. The species and date keys were looked up.

Loading the Trade Fact Table (Transactional). The trader and species are looked up.



Loading the Spatiotemporal fact table (transactional). The date, location, weather, and soil keys are looked up.



Loading the Observation fact table (transactional). The date, location, and species keys are looked up.

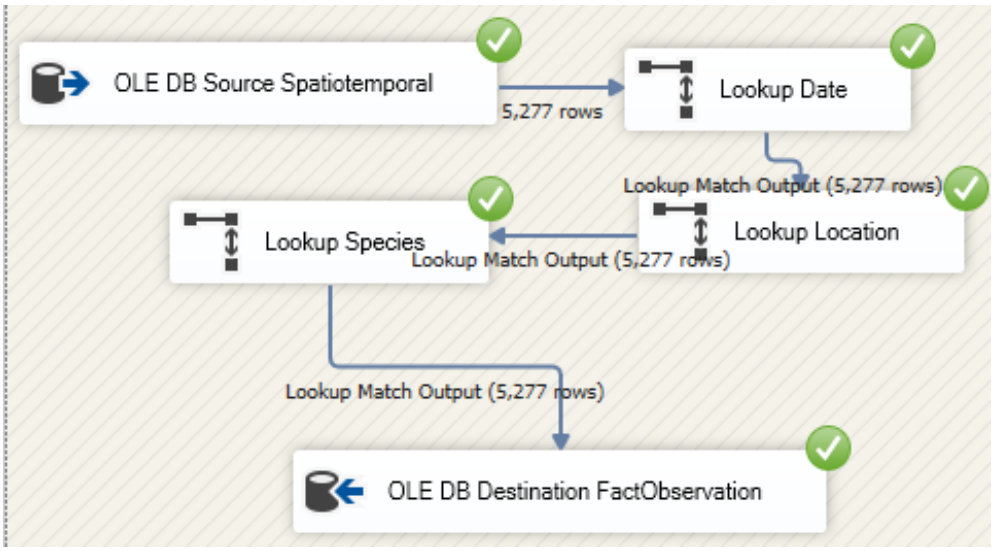
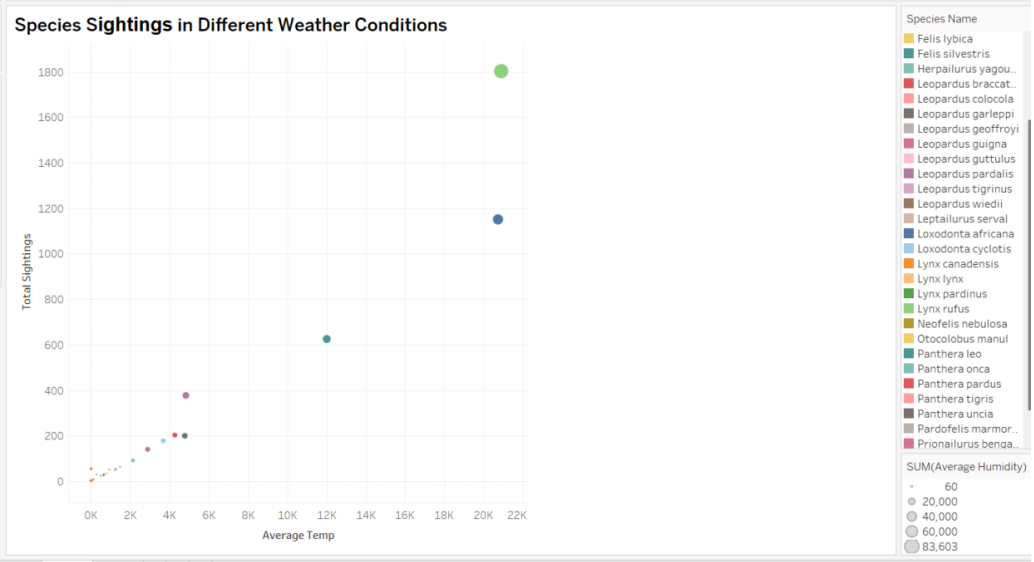
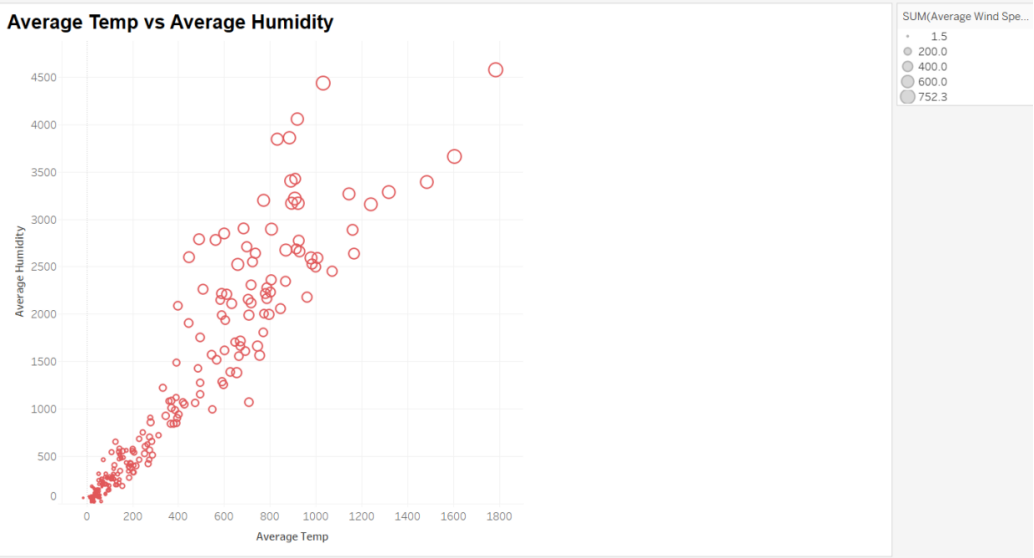


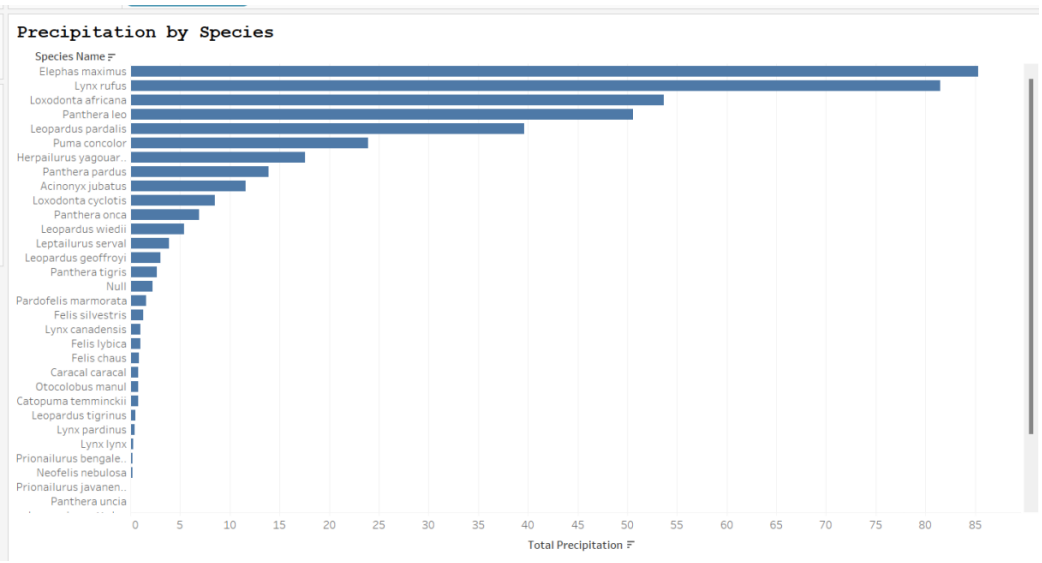
Tableau Visualizations



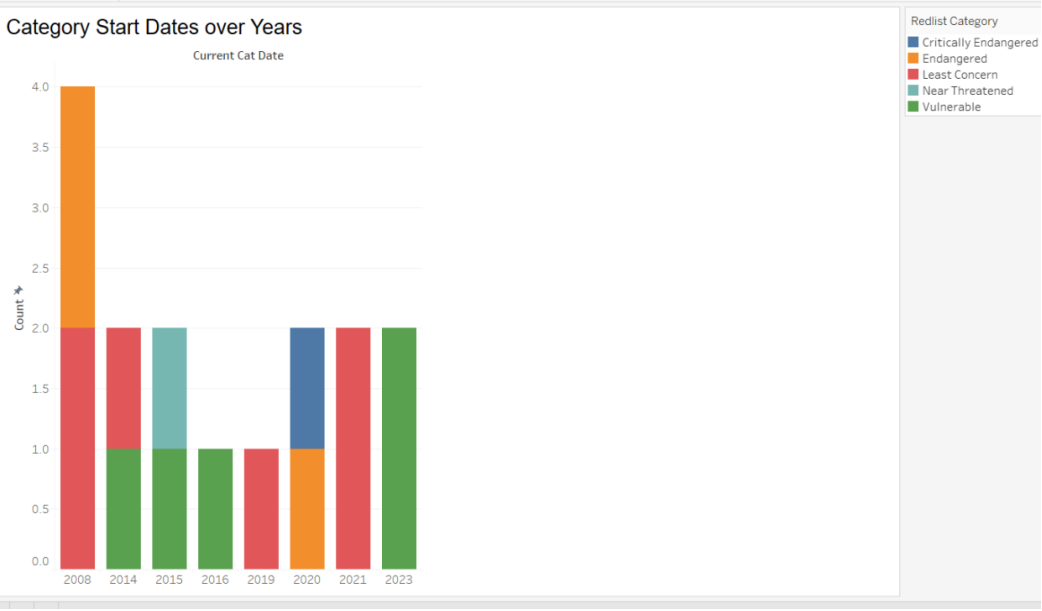
Scatterplot showing the total sightings of each average temperature. The size of the point is determined by the total humidity and the color is determined by the species. We decided to show it because it shows when each species is most likely to be spotted and compares the environment of the species.



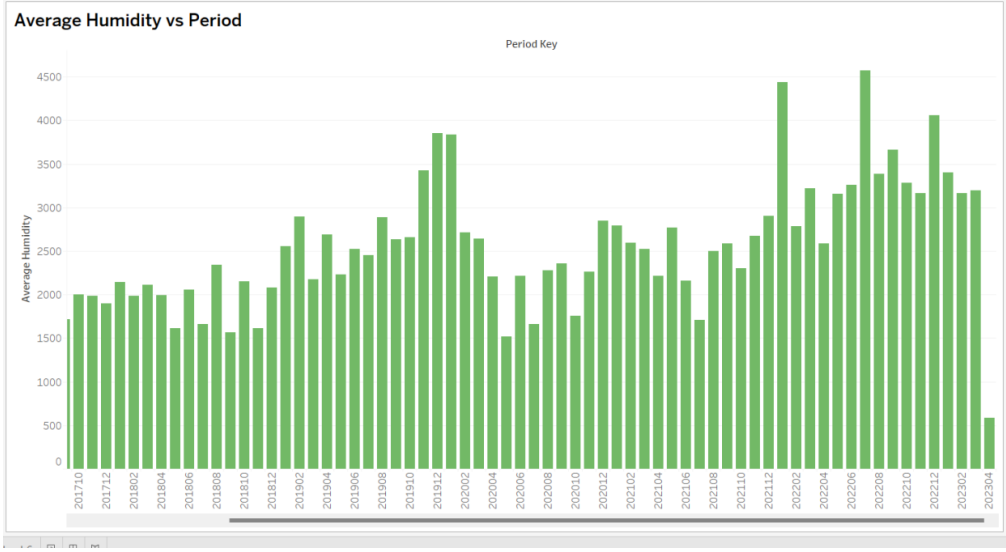
Scatterplot shows the average temperature for each average humidity. The size is determined by the average wind speed. We decided to show it because it compares the different environmental observations observed.



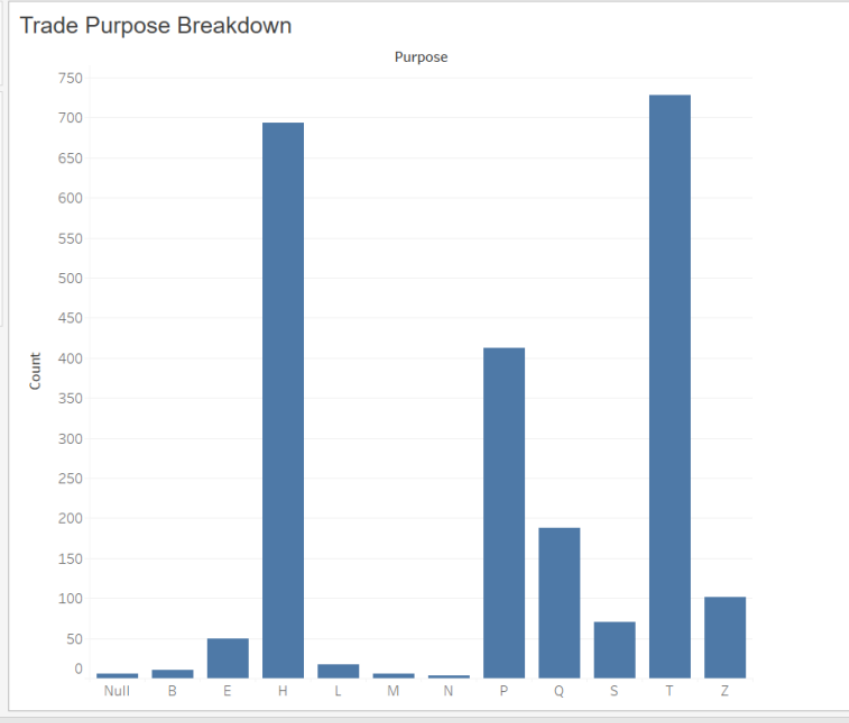
Shows the total precipitation observed for each species. We decided to show it to further see the environmental conditions each species experiences.



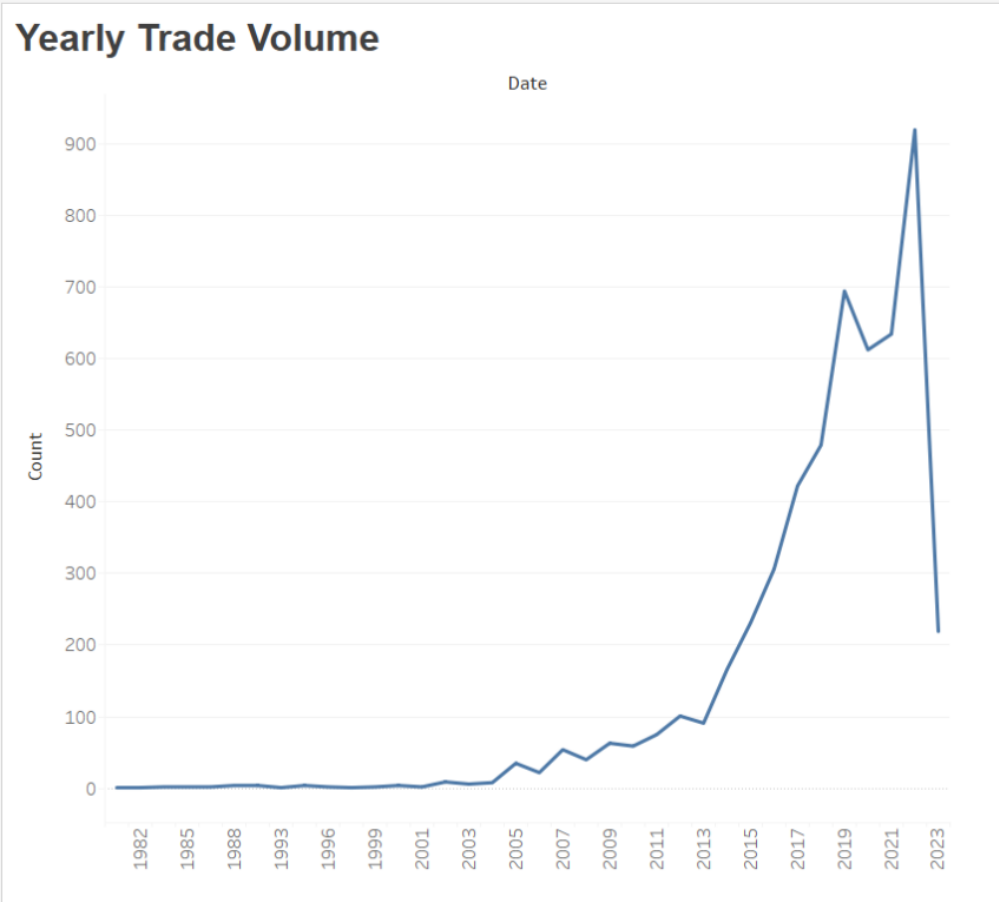
Shows the number of species who entered their current red list category in each year. We decided to show it to see yearly trends for species moving categories.



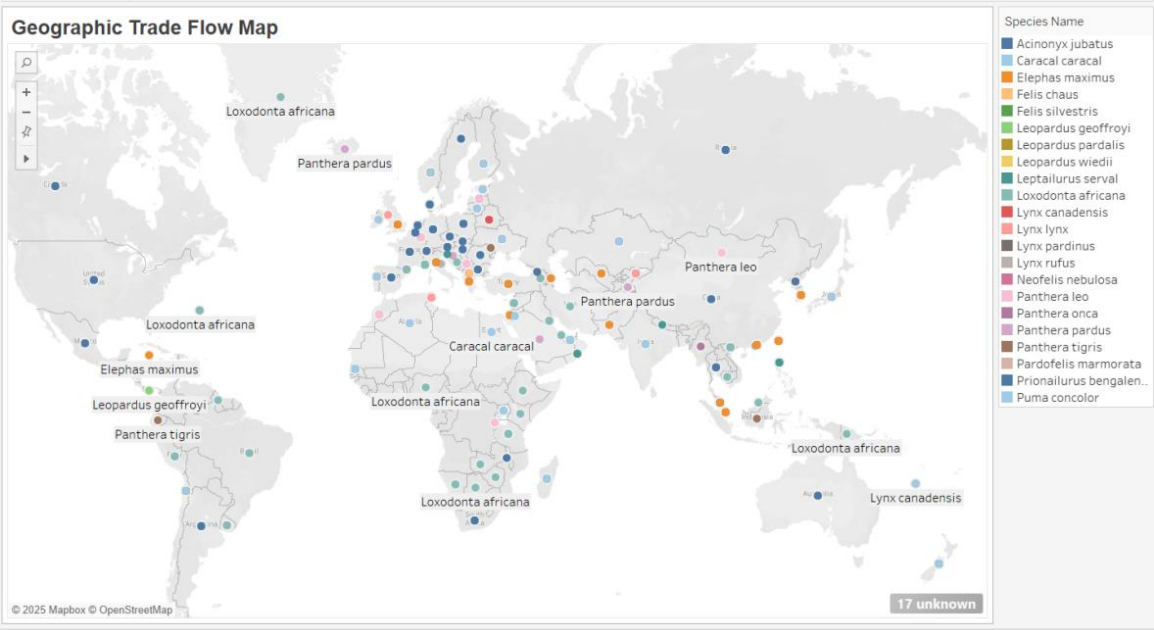
Shows the average humidity for different months. We decided to show it to see how the average humidity changes across periods.



Shows the number of trades for each purpose. We decided to show it to see why species are traded, which is one reason for why many species are becoming endangered or moving redlist categories.



This line graph shows the number of trades by each year. We decided to show it because it shows trading is increasing. This describes a reason why many species are becoming rarer and moving redlist categories.



Geographic visual of where each distinctive species is either imported to or exported from. This visual provides insight on the types of species that exist and are recorded.

Narrative Conclusion

Experience

The most challenging step was finding suitable data. This took the longest to do and once the data was found the rest went smoothly. The next challenging step was structuring the data warehouse, figuring out what would go where and what data should be in each table. It took some trial and error to see what worked and what didn't. The ETL pipeline wasn't particularly challenging except for making the Periodic Snapshot as we weren't sure how to do some parts but there were online resources that explained how to do it. There was also some struggle downloading the data warehouse schema. The easiest steps were the other ETL steps after everything was sorted out. Designing the star schema was also straightforward and pretty easy. The queries and Tableau were also pretty simple. We were surprised by how hard it was to find data that would fit the project. We were also surprised that the ETL steps weren't as challenging as we expected them to be. We had expected it to be the other way around, that finding the data would be the easy part and the ETL would be the hard part. If we were to start over, we would try to find the data earlier so the rest of the project didn't feel rushed. This would give more time for profiling the data and planning transformations before starting the Load. This would have simplified along of the struggles we had during the project.

Benefits

The data warehouse enables integrated, multi-dimensional analysis of species observations, conservation status trends, environmental conditions, and trade activity. The structured warehouse supports a range of queries that would be difficult to perform with raw datasets. The benefits include identifying habitat risks, monitoring conservation category changes, and exploring clime related factors behind population trends. Decision makers in wildlife conservation, research, and policy can benefit from faster access to meaningful insights drawn from large and diverse data sources.

Final Comments & Conclusions

The toughest segment of this project based on the group's opinion has to be starting it, as well as, preparing the data for the ETL portion, because in order to design a data warehouse, you would first need to find the proper data set for the job, which our group learned the hard way, this is easier said than done. Then, once a dataset is located, we would then need to clean it for any null, duplicated, or drop any unnecessary columns and rows that don't contribute to the primary objective, which in this case is designing a data

warehouse. However, despite the difficulty, we believe that the biggest lesson learned is to start early in looking for a dataset that fits the core objective of the project.