Wildlife Preservation Organization Operational Database

Zhengjia Cao
SSE
Computer science and engineering
117010008@link.cuhk.edu.cn

Haoxiang Liu
SSE
Computer science and engineering
117010164@link.cuhk.edu.cn

Jiazhen Liu
SSE
Computer science and engineering
117010164@link.cuhk.edu.cn

Zheyuan Zhou
SSE
Computer science and engineering
117010423@link.cuhk.edu.cn

***Abstract*—CITES (Convention on International Trade in Endangered Species of Wild Fauna and Flora) is an international agreement between governments on sustainable trade of endangered species. Albeit the widely recognized significance of wildlife preservation, few analysis on multidimensional data has been conducted due to the large quantity of data. This project presents the design and implementation of a MySQL database with data visualization tools, which aims at solving the above mentioned problem as well as providing CITES a database for operational use by constructing well-normalized entities and relations. Indexing and hashing was also implemented for better query performance. Experiments show that a better database model can achieve better user experience, e.g., in overall query latency.**

***Index Terms*—Relational database, animal preservation, data analysis**

I. Introduction

The conservation of biodiversity is a complex problem strongly tight to political actions. The international wildlife trade that includes hundreds of millions of plant and animal specimens is now estimated to be worth billions of dollars annually. The trade is diverse, covering from live animals and plants to a vast array of wildlife products derived from them. However, high level of exploitation of some animal and plant species as well as the trade in them are capable of heavily depleting their populations and even bring some species to extinction. Many wildlife species in trade are not endangered, but the existence of an agreement to ensure the sustainability of the trade is important in order to safeguard these resources for the future. [1]

Convention on International Trade in Endangered Species of Wild Fauna and Flora, also know as CITES, is an international agreement between governments. Its aim is to ensure that international trade in specimens of wild animals and plants does not threaten their survival. CITES is one of the eight main international agreements relevant to biodiversity (CBD, n.d.) and constitutes a key tool for conservationists, scientists and policy makers.

Currently, the CITES database is open for queries through an R package that provides users with APIs to perform data analytic in R.[2] The source data is also available in the form of CSV files. [3] While it is suitable for small amount of queries and data size, as there are currently more than 2 million records in the database, we consider it can be better organized in the form of a relational database.

Our project aims at proposing a MySQL database for the CITES organization with all the trading records properly stored for fast retrieving, as well as the data essential for daily administrative operations. We will also demonstrate the queries with our database and perform sample analysis on the data on our project website.

II. Literature Review

Although it is a widely accepted social recognition that protecting wildlife and regulations on preserved creature are essential, the current researches and related works on this topic have their major limitations in the following three perspectives:

Firstly, many research teams only mentioned the data organization, regulation or visualization are needed vaguely, but omitted to provide the detailed information or the concrete implementing function. For example, Lin promotes on establishing the information sharing platform for wildlife trade to provide supervision, but the team do not give the detailed description of the platform.

Secondly, although simple data analysis was conducted, they were all based on limited dimensions of the data., resulting in the loss of the correlations between different data dimensions Using the data provided by Pires in the The Illegal Wildlife Trade as an instance, this paper only provides comprehensive data analysis result for some major traded creature: "prioritize charismatic mammals, for example, elephants, rhinos, big cats, great apes, and pangolin", but omitted the data for other species.

Thirdly, even if the detailed data analysis is given for multiple creatures as what was provided in Margulies's paper, the result is presented in the form of literature, which makes readers hard to form a clear idea of the data or to perform quantitative analysis to the data.

With the above limitations, a database for complete record of every recorded creature and the organization regulation, correspondingly, the related dynamic data visualization are needed.

Hence, in this project, we provide the well-organized database system based on SQL for CITES, which records both the creature-related and administration-related information to promote CITES's information system. Furthermore, a user-friendly website is also provided for both CITES employees and any other interested users including but not limited to researchers, students and environmentalists.

III. Design and Implementation

A. Requirement Analysis

1) Administrative Structure of CITES: According to the description of CITES, the administrative structure of CITES can be divided into three parts: Plants Committee, which in the charge of plant related trades; Animals Committee, which in the charge of animal related trades; Standing Committee, which in the charge of the daily operation of the organization. The division of the parties and the connections between them are shown in the following figure:

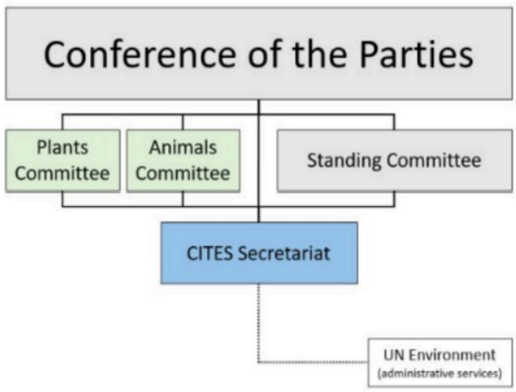


Fig. 1. Administrative Structure of CITES

2) Working Procedure: With the specific structure, the legal trade working flow among CITES, Parties and institutions is listed as following. Firstly, when some institution wants to export or import a specific creature to a certain country or area, the institutions need to submit application to the target government. Secondly, the government received application will determine whether pass or fail the application according to the country's related law to the creature. Thirdly, when the permission is obtained from the government, the institution need register to the CITES, and the export or import application is legal until the CITES also approves the application.The sequential diagram of the legal trade work flow is indicated in Figure 2.

CITES is also responsible for the illegal wildlife trade record. When one country's customhouse intercepts illegal wildlife trade, it will report to CITES and CITES will create a

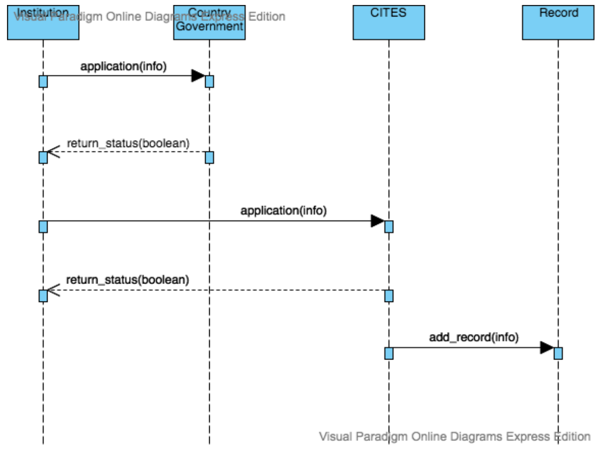


Fig. 2. The Working Flow among CITES, Parties and Institutions Legal Trade

new record for the trade. The sequential diagram of the illegal trade work flow is indicated in Figure 3.

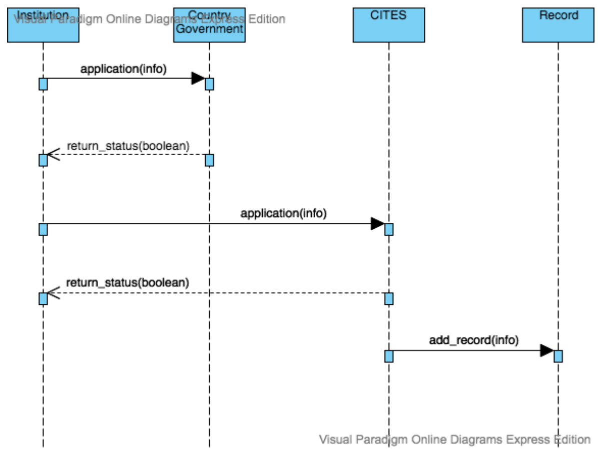


Fig. 3. The Working Flow among CITES, Parties and Institutions Illegal Trade

3) Requirements: This organized work flow generates two major types of requirements: record related requirements and administration-related requirements. The record related requirements involve the accession in the application handle, record edition, permission record and creature record. These records are usually read and added much more more frequently than being updated since the trading records should not be modified. The only update would happen when the preservation appendix of a creature was changed.

The administration-related requirements involve the staff information operation, application handle record, fund information, conference record, Party record, Committee record and institution records. These recording will be frequently updated and we shall discuss the functional dependencies and other restrictions in the later sections.

B. Entity

According to the above information, we list ten basic entities with corresponding attributes:

- CITES_application(application_ID, status)
- Creature (name, creature_ID, appendix, genus, order, family, class, isAnimal, year)
- Permission (permission_ID, institution, type)
- Records (trade_ID, origin, purpose, source, unit, term, year, quality)
- Conference (conference_ID, name, date, location)
- Country (ISO, country_and_area, region, contribution, convention_type, join_date, is_member)
- Staff (staff_ID, last_name, first_name, gender, is_leave, position, salary, age)
- Institution (institution_ID, city, district, street, certification, validation_date)
- Fund (fund_ID, CITES_adjust_scale, last_triennium_contribution, unpaid_amount, UN_scale)
- Committee (committee_ID, committee_name)

C. Attributes, Relationships and Cardinalities

The following mapping cardinalities are involved in this database: many-to-many, many-to-one, one-to-many. The type of each relationship will be listed as below:

- CITES_application and Permission:
  one-to-many, one application may need several (including 0) permissions, but permission belongs to only one application.
- Permission and Creature:
  many-to-one, permission is associated with one species, but one creature is associated with several permissions.
- Creature and Records:
  one-to-many, one record only contains one species, but one species can be traded in several records.
- Creature and Country:
  many-to-many, one species may have originated in several countries, and one country has many species.
- Records and Staff:
  many-to-many, one records many be processed by several staff, and one staff can handle many records.
- Conference and Staff:
  many-to-many, many staff may work for one conference, and one staff may attend many conferences. What's more, participation in the relation of attended_conference is total, because every conference must need staff to organize.
- Records and Country:
  there exist these relationships between records and country: import, export and re-export. They are used to represent for this trade, the importing nation and the exporting nation. Usually, a valid trade would include both relations. The re-export relation is used for recording if this trade is involved in multiple import and export relations.
  For the mapping cardinalities of all the relations, the obtaining logic is the same. One record only has one import/export/re-export country, but one country may be associated with several records. Thus we use the many-to-one mapping in these relations.
- Conference and Country:
  many-to-many. There also exist two relationships, participate and held. One conference has many countries to participate, and maybe more than one country to held conference. Of course, every country can participate and held several conferences. Besides, every conference must be associated with countries.
- Country and Staff:
  one-to-many. One country is associated with several staff. One staff must have its nationality. In our assumption, we do not support dual nationality.
- Staff and Staff:
  one-to-many. One staff has his manager, and may have many subordinates.
- Country and Institutions:
  one-to-many. One country may have several institutions. Every institution must belong to one country.
- Country and Fund:
  one-to-many. One country is associated with several funds, while each fund must be associated with only one country. So participation of fund in the relation of sponsor is total.
- Staff and Committee:
  many-to-one. Every staff must be associated with one committee. One committee is associated with several staff. The participation of staff in the relation of staff_-committee is total.

D. ER Diagram

1) Primary Key Choice: With all the entities and relations listed about, we have considered these constraints when choosing the primary keys and foreign keys for our schema:

- Primary key should be the unique identifier for every tuple
- Primary key is chosen from possible candidate keys with the concern of simplicity
- Foreign key of the relation must appear in another referencing/referenced relation

To achieve the ease of search with the primary keys, we decided to create IDs for most of our recording entities. We used a hexadecimal number to record our IDs. It's uniqueness is thus guaranteed since the later records will be monotonically increasing the ID of the previous records. In our database implementation, the ID attributes are stored as a VCHAR(45) type in MySQL, which means using this method, thoeratically we can store 16^{45} number of records. This should be sufficient for our database application.

2) Normal Form Analysis: We have generally normalized our relations into Boyce-Codd normal formal with some exceptions considering the preservation of functional dependencies, which we left the relation as 3NF. Our Entities and the Functional dependencies are listed here:

- Records(tradeID, year, term, unit, purpose, source, origin, quantity)
  we have these functional dependencies exists in this entity:

$$tradeID \rightarrow origin, purpose, source, unit, term, \\ year, quantity \tag{1}$$

  We can see that all the functional dependencies here are fully functional dependent on the primary key, and for any non-trivial functional dependencies in F+, it is dependent on the super key only. Thus, this entity is now in BCNF.
- Creature (creature_ID, isAnimal, year, appendix, taxon, class, order, family, genus, name, habitat) FDs:

$$creature\_ID \rightarrow isAnimal, year, appendix, taxon, \\ class, order, family, genus, habit \tag{2}$$

  We also have: $genus \rightarrow family$, $family \rightarrow order$, $order \rightarrow class$, $class \rightarrow taxon$ In this entity, while we have all of the attributes functional dependent on the primary key, we also notice that in (appendix, taxon, class, order, family, genus), the previous attribute is functional dependent on the following attribute.
  In fact, the following attribute is a generalization of the previous attribute. While it is possible to make them into entities and construct relations between them to make it strictly following the rules of BCNF, we decided to keep it as attributes in the Creature entity. The reason is because that the Creature entity is being accessed much more frequently than it is being updated. If we decided to keep it in BCNF, then for every retrieve of the information, we need to perform natural join between these entities, which will significantly increase the query time.
- Country(ISO, Country_and_area, region, contribution, convention_type, join_date, isMember) This is also in BCNF since ISO is the unique identifier for a country, and all the other attributes are fully functional dependent on the country. There is no transitive dependency between them.
- Institutions(IID, address, Certificate, validation_date) In this relation, there exist a multi value dependency $IID \rightarrow\rightarrow address$ since many institutions can have multiple locations. We then claim that this relation is in 4NF, since it already satisfy the requirements for BCNF.
- Staff (Staff_ID, age, gender, position, salary) In this relation, all the attributes are fully dependent on the superkey, and is in BCNF. The department relation between staff and committee is shown through the relation table *staff_committee*.
- Fund (fund_ID, CITES_adjusted_scale, latest_triennium_contribution, annual_avg_contribution, unpaid_amount, UN_scale) This relation is in BCNF since we have created the ID for each of the funding source. Through the *sponsor* relation we can join back the orginal table for each country's contribution to the CITES.
- Conference (conference_id, name, date, location)
- CITES_application(ApplicationID, status)
- Permission(permitID, importer_name, institution, type)

3) ER Diagram: Based on the analyses above, we drew the ER diagram as presented in Figure 4.

E. Indexing and Hashing

We have 4 entities which contain tremendous amount of data: creature, record, application and permission. As a result, the amount of data in the schema that result from the relationship between them is also strikingly huge. We have over 38, 000 types of creatures, 800,000 records, 1,000,000 applications and corresponding permissions. Therefore, it will bring significant performance boosting if we can adopt indexing or hashing on the database. Considering that the large amount of data will increase the difficulty of designing an efficient hash function to reduce collision problems, we prefer indexing to hashing. For our database, we finally decided to use multi-level indexing method for boosting our queries.

We will apply indexing to the following data fields:

- Creature
  We can store the records according to the creature ID, and make creature ID the search key since the creature ID will be unlikely to be changed. The index is based on creature ID, and this will become ISAM. In this way, we can access, update and delete the records very efficiently by using linear search or binary search.
- Record
  Similarly, the records are ordered according to the ID, and indexing is applied upon the ID. Since there are even more (800, 000) records than the number of creatures, we propose to use multi-level indexing on the ID, in order to achieve better performance.
- Application and Permission
  There are 1,000,000 records in the Application and Permission schema, so we plan to use multilevel indexing on the ID.

F. SQL Queries

To test our databases, we analyzed the possible requirement of the organization, and listed the common SQL queries that may be frequently used. Some queries are designed for operational uses, while we have also used some OLAP operations in our queries for further data analysis. In general, the queries can be partitioned into 2 parts, with the first part focusing on the queries about creature information and trade records while the second part mainly used for CITES' daily administrative operations. The queries are executed using Navicat and MySQL workbench. Some results are presented as follows.

PART 1: Queries about Creature and Records Information

1) Count the number of records for a specific type of creature in a certain year:

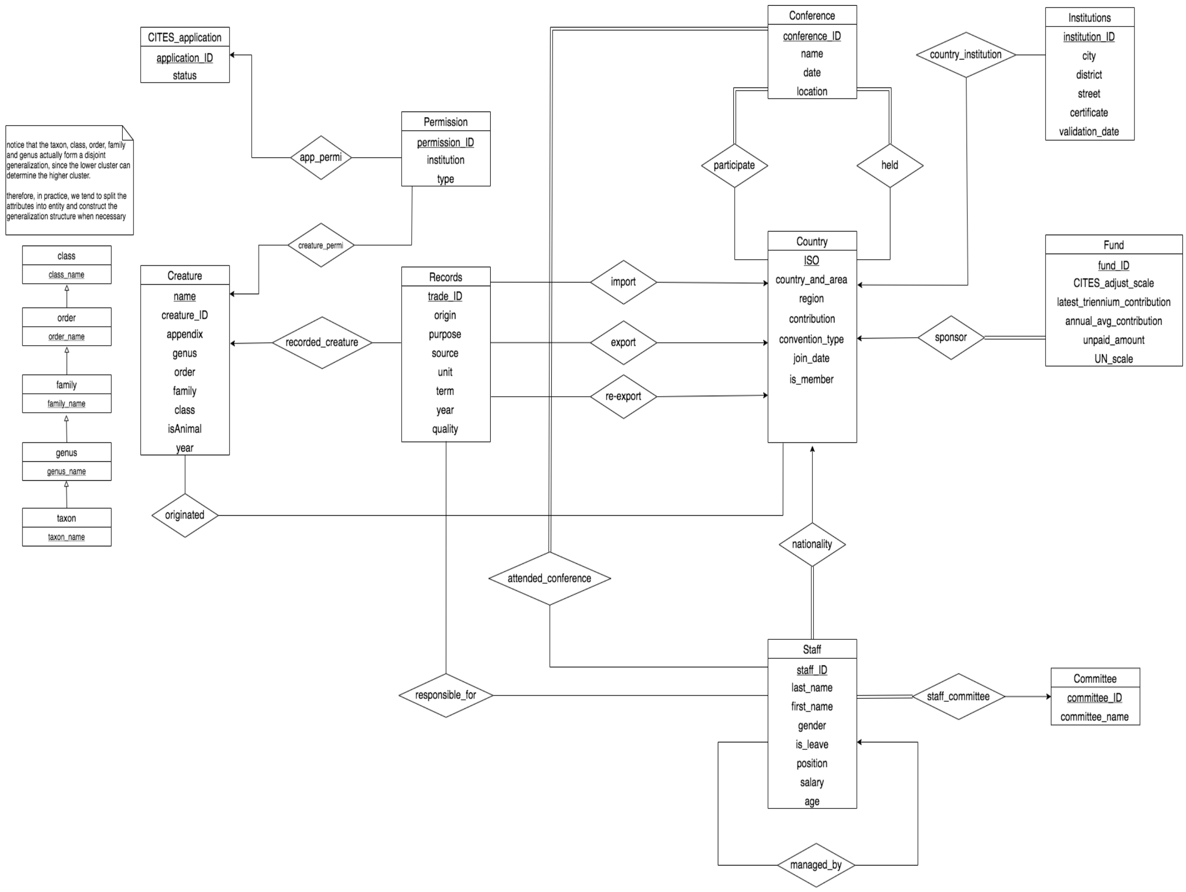


Fig. 4. ER-Diagram

```sql
SELECT scientific_name, Record.year,
    count(distinct trade_ID) AS
    trade_count
FROM Recorded_creature NATURAL join
    Record
GROUP BY scientific_name, Record.year
    ;
```

2) Count the number of records for each type of creature, and order the results in descending order:

```sql
SELECT scientific_name, SUM(
    trade_count) as total_trade_count
FROM ( SELECT scientific_name, Record
    .year, COUNT(distinct trade_ID) AS
     trade_count
        FROM Recorded_creature
            NATURAL JOIN Record
        GROUP BY scientific_name,
            Record.year) as
            creature_year_records_amt
GROUP BY scientific_name;
```

3) Use ROLLUP operation to check the trade amount according to ISO, region and year:

```sql
SELECT ISO, region, year, COUNT(
    distinct trade_ID) as trade_count
FROM (
        SELECT T.trade_ID, T.ISO, T.
            region, Record.year
        FROM (SELECT trade_ID, ISO,
            region
                FROM Import NATURAL
                    JOIN Country
                ) as T NATURAL JOIN
                    record
) as S GROUP BY ISO, region, year
    with ROLLUP;
```

4) To see which terms of which creatures are most frequently traded, rank and return the top 10 ones:

```sql
SELECT T.scientific_name, T.term, T.
    TermTotalTrade
FROM(
        SELECT Recorded_creature.
            scientific_name, Record.
            term, COUNT(distinct
            Record.trade_ID) as
            TermTotalTrade
```

```
FROM Record,
    Recorded_creature
WHERE Record.trade_ID =
    Recorded_creature.trade_ID
GROUP BY Recorded_creature.
    scientific_name, Record.
    term
ORDER BY TermTotalTrade DESC
    LIMIT 10
) as T;
```

5) Create a view, and check the top 10 countries with the largest export number of trades for a specific kind of creature (here the example uses *Alligator mississippiensis*):

```
CREATE VIEW Trade_creature_country AS
    (
        SELECT Export.ISO, count(
            Export.trade_ID) as
            trade_count
        FROM Export,
            Recorded_creature
        WHERE Export.trade_ID =
            Recorded_creature.trade_ID
             AND Recorded_creature.
            scientific_name="Alligator
            ␣mississippiensis"
        GROUP BY Export.ISO
        ORDER BY trade_count);

SELECT Country.country_name,
    Trade_creature_country.trade_count
FROM Trade_creature_country, Country
WHERE Trade_creature_country.ISO =
    Country.ISO
ORDER BY Trade_creature_country.
    trade_count DESC LIMIT 10;
```

6) Rank the countries according to how many types of creatures originated there:

```
SELECT country.country_name, COUNT(
    originated.scientific_name) as
    num_creatures_originated
FROM originated, country
WHERE originated.ISO = Country.ISO
GROUP BY country.country_name
ORDER BY num_creatures_originated
    DESC;
```

7) Use moving average to calculate the average number of records in 3 years (last year, this year and the next year):

```
SELECT year, num_trade,
AVG(num_trade) OVER(ORDER BY year
    ROWS BETWEEN 1 PRECEDING AND 1
    FOLLOWING) as avg_num_trade
FROM (
        SELECT year, count(distinct
            trade_ID) as num_trade
        FROM record
        GROUP BY year
        ) as num_records_year;
```

PART 2: Queries about CITES' Daily Operations

1) The number of working staff each country has:

```
SELECT country_name, COUNT(distinct
    staff_id) AS country_num_staff
FROM Nationality, Country
WHERE Nationality.ISO = Country.ISO
GROUP BY country_name
ORDER BY country_num_staff;
```

2) Check fund information of a specific country:

```
SELECT *
FROM Sponsor, Fund
WHERE Sponsor.fund_id = Fund.fund_id
    AND Sponsor.ISO='US';
```

3) Rank the countries according to the amount of money that is unpaid yet:

```
SELECT ISO, SUM(Fund.unpaid_amount)
    as total_amt_unpaid
FROM Sponsor, Fund
WHERE Sponsor.fund_id=Fund.fund_id
GROUP BY Sponsor.ISO
ORDER BY total_amt_unpaid DESC;
```

4) List all the participating countries for a CITES conference:

```
SELECT country_name
FROM participate, country
WHERE Conference_ID = 1 and
    participate.ISO=country.ISO;
```

5) To see a specific staff is responsible for which records:

```
SELECT staff_id, trade_id
FROM Responsible_for
WHERE staff_id='599';
```

6) Rank the staff according to the number of records that they are responsible for, and return the top 5 staff:

```
SELECT staff_amt_records.staff_id,
    first_name, last_name, amt_records
FROM(
        SELECT staff_id, COUNT(
            distinct trade_ID) as
            amt_records
        FROM Responsible_for
        GROUP BY staff_id
        ORDER BY amt_records DESC
            LIMIT 5
        ) as staff_amt_records, staff
WHERE staff_amt_records.staff_id=
    staff.staff_id;
```

7) Find the staffs who are responsible for a certain record:

```
SELECT staff.staff_id, staff.
    last_name, staff.first_name
FROM Responsible_for, staff
WHERE trade_ID = '123' and
    Responsible_for.staff_id=staff.
    staff_id;
```

IV. Data Analysis and Visualization

In order to intuitively show our database, we have built our website for better data queries. The website consists of 3 parts: the initial page, static data visualization, and dynamic query. The static data part will display the information related to all the records in the database(e.g, the most traded creature of all time), and the dynamic query will give tables and graph with user specified restrictions.

The database was connected with the flask server using MySQLAlchamy. The query results is displayed on the website created based on react.js framework.

A. Initial Page

The initial page of our website is a heat map illustrating the types of protected creature that originate in each of the countries. The darker color represents a higher number of creatures originated from the country.

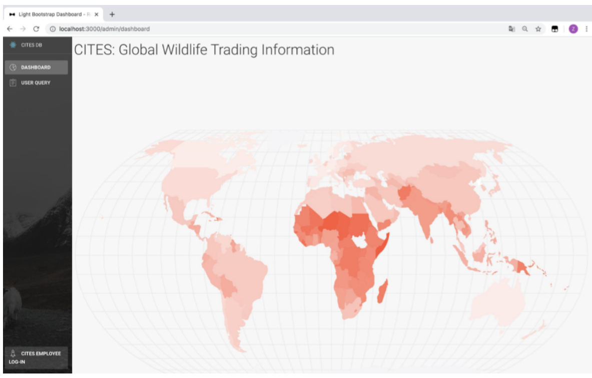


Fig. 5. Initial Page: Heat map

From the heat map, the user can clearly interprets that countries in middle Africa are home to the largest varieties of species. Generally speaking, tropical and temperate region breed more types of creatures, compared with the cold frigid zone, which coincides with our common sense, as appropriate temperature is the prerequisite for the sustainable living of creatures. The heat map is shown in Figure 5.

B. Static Data Visualization

The second part of the website, a dash board, aims at statically showing the results of our queries. The country with the highest and lowest import/export amount are listed, and the users can view the relevant information by specifying certain attributes like country name or the scientific name of creatures. Some example results are shown here. To learn more about the web page, please see the Appendix.

- Biggest and Smallest Importer/Exporter This graph is

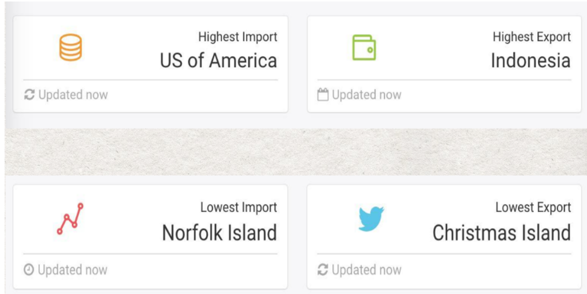


Fig. 6. Biggest And Smallest Importer/Exporter

  shown in Figure 6. Our analysis shows that the US of America is the country with the highest amount import records, while Indonesia tops the list of the exporters.
- Amount of Creatures in Each Country

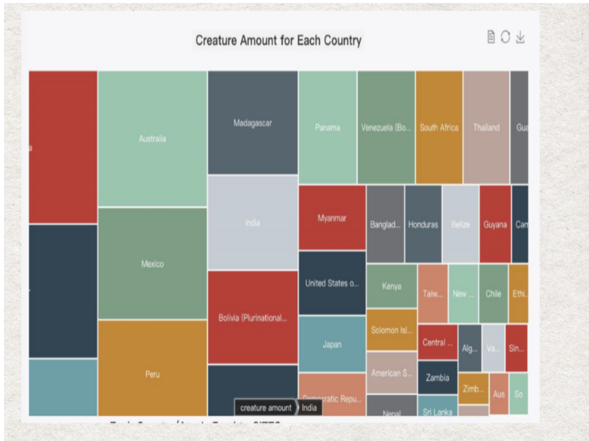


Fig. 7. Amount of Creatures in Each Country

  Figure 7 shows the biodiversity of each country. Larger area of the rectangle implies more types of creatures. Among all the countries, Australia, Mexico, Peru and so on are rich in their creature diversity.
- Percentage of Species in Trade
  Figure 8 is intended to show the percentage of different species involved in trades. The user can select the creature of interest from the legend on the right, and the corresponding percentage will be presented on the pie chart on the left.

C. Dynamic Query and Visualization

In this part of the web page, we allow the users to specify one or more attributes in country, species and year, make use of *ROLLUP* we have learned in class, and present the results in tables and graphs as required. With 3 attributes to specify, the users have: $2^3 - 1 = 7$ types of combinations. Here we paste some example results. h

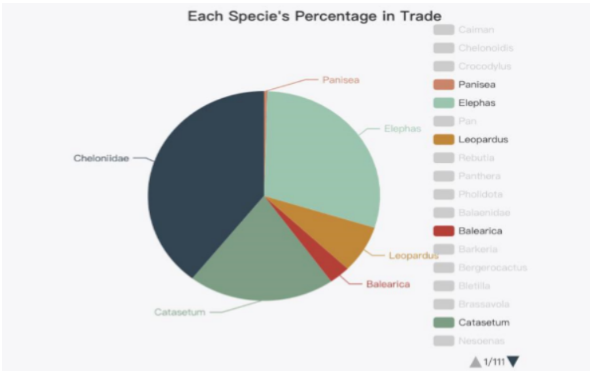


Fig. 8. Percentage of Species in Trade

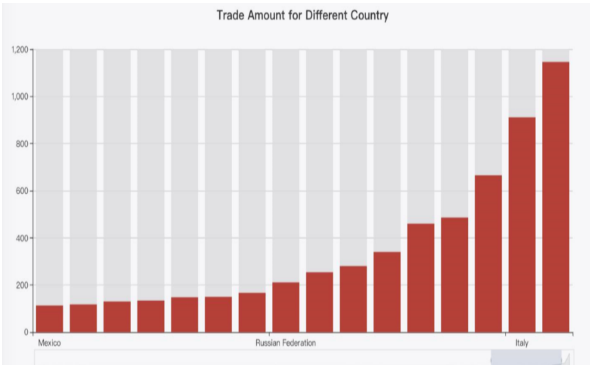


Fig. 9. Trade Amount for Different Country

- By specifying the type of creature and year, return a histogram with the x-axis being the country (In Figure 9):
- By specifying year and rolling up against country names, we get a pie chart for record amount of each species (In Figure 10):

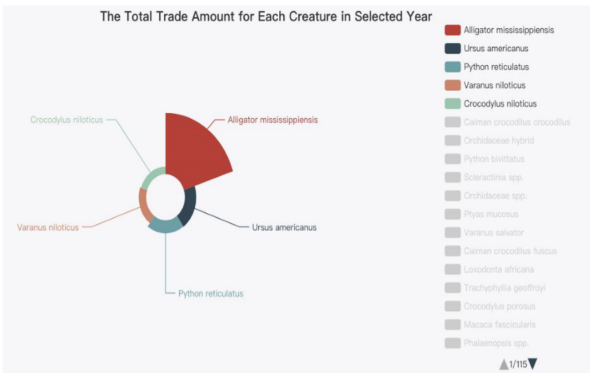


Fig. 10. The Total Trade Amount for Each Creature in Selected Year

V. Conclusion and Self-Evaluation

From this project, we have understand how a database should be designed from the analyzing the need of the organization to the final product. We have practiced to identify the relevant entities,attributes and relationships together with constraints and properties, that were eventually represented in the form of ER diagram.We have also practiced SQL queries and optimization on real data.

Compared with CITES's database, our database is more convenient for queries when involved large amount of data size. To execute query command, CITES's database use R package to provides users with APIs and display the result of analytics. It is not efficient when processing large amounts of data. Instead, our database can handle this situation and is capable of mass data processing and data analysing in a short time. Despite the improvements in some operational queries, our database still present a poor performance when the user wants to natural join two tables which both have million rows without any constraints. It will do Cartesian product at first. This step cost much time if there does not exist any constraint for entities.

For the future works, we will try to solve this problem. We may use another subsidiary form to satisfy users' requirement. Furthermore, in the database website designed for CITES, we will add admin part. The website now support query information of record. From those queries, we can get information that is detail to which country involved in which species trade in which year. Based on this, many analytics can be done, and results of analysis will be presented in page. However, admin part is lost and the model is fixed, which means that user can only select limited attributes and gain specific tables and charts. In our future works, we will add administrator permission to add update or delete record. On the other hand, user can do diverse and customized query.

From our analysis about, it is safe to say that our database project will serve as a ease to use analytical tool for studies on wildlife trading and preservation. CITES have already benefited from the analysis of it's trade database.[4] The study of trades on American alligator have contributed to the law making for sustainable trade. We hope that our project can serve as an example of how social science researchers can utilize data analytical tools to have a deeper insight for their works.

Acknowledgment

We would like to give special acknowledgment to Professor Clement Leung and the hardworking TAs for this course. We finished this project with their dedicated teachings.

References

[1] "What is CITES" Accessed on: May 1,2020.[Online]. Available: https://www.cites.org/eng/disc/what.php

[2] "SPECIES+/CITES CHECKLIST API DOCUMENTATION" Accessed on: May 1,2020.[Online]. Available:https://api.speciesplus.net/documentation/v1.html

[3] "CITES Trade Database" Accessed on: May 1, 2020.[Online]. Available:https://trade.cites.org/

[4] U.S. Fish & Wildlife Service, "A CITES Sustainable Trade Success Story: American Alligator Recovery and Trade," Accessed on: May 1, 2020.[Online]. Available:https://www.fws.gov/international/articles/a-cites-sustainable-trade-success-story-alligator-recovery-and-trade.html/

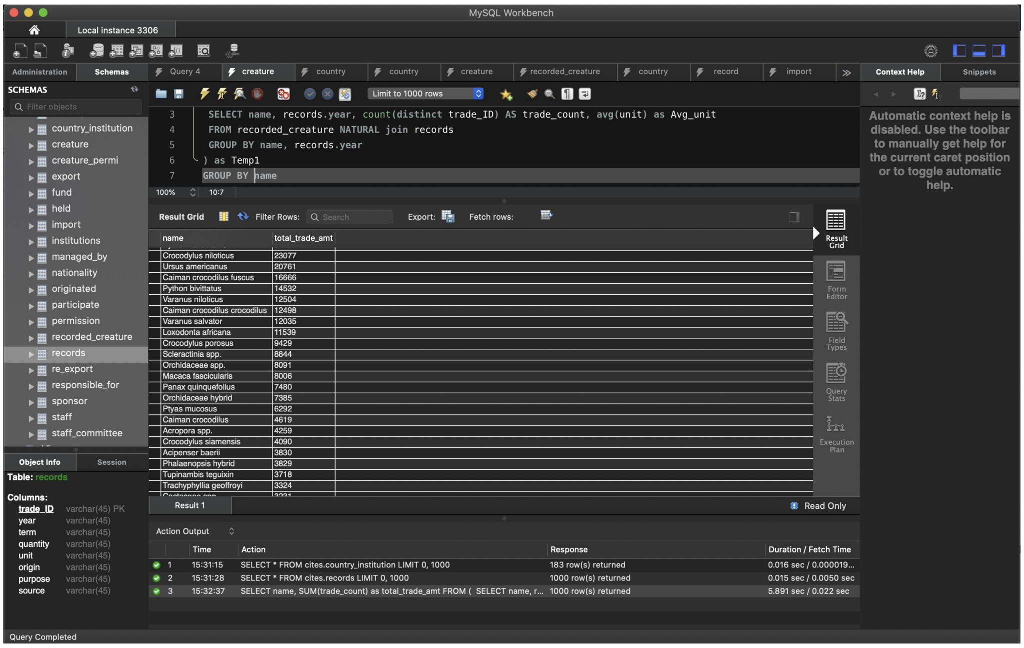
Appendix

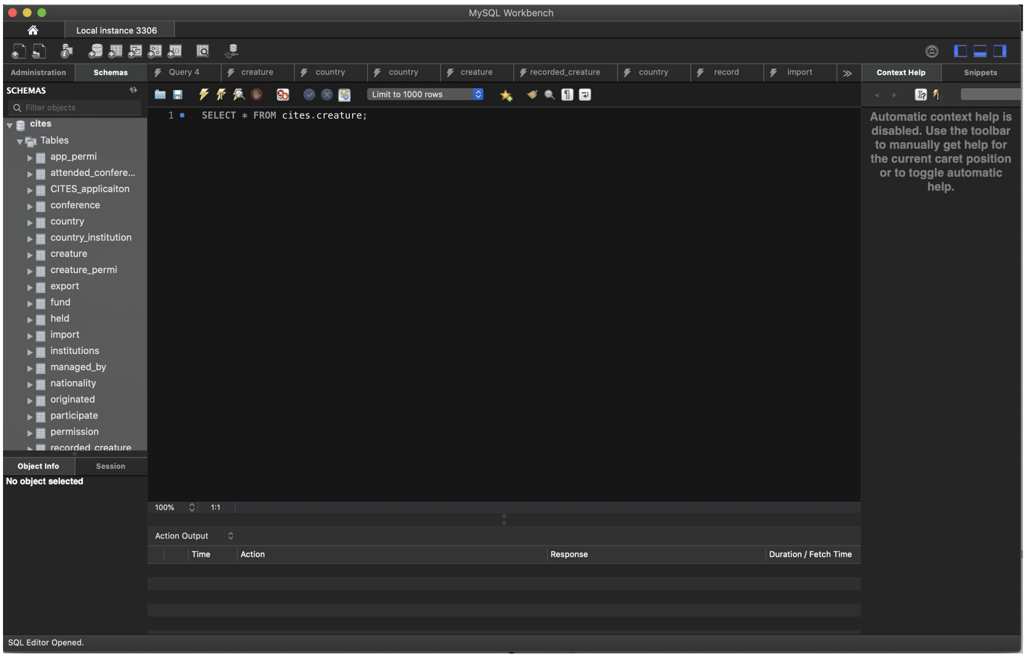
[1] Github Link of CITES Database Project

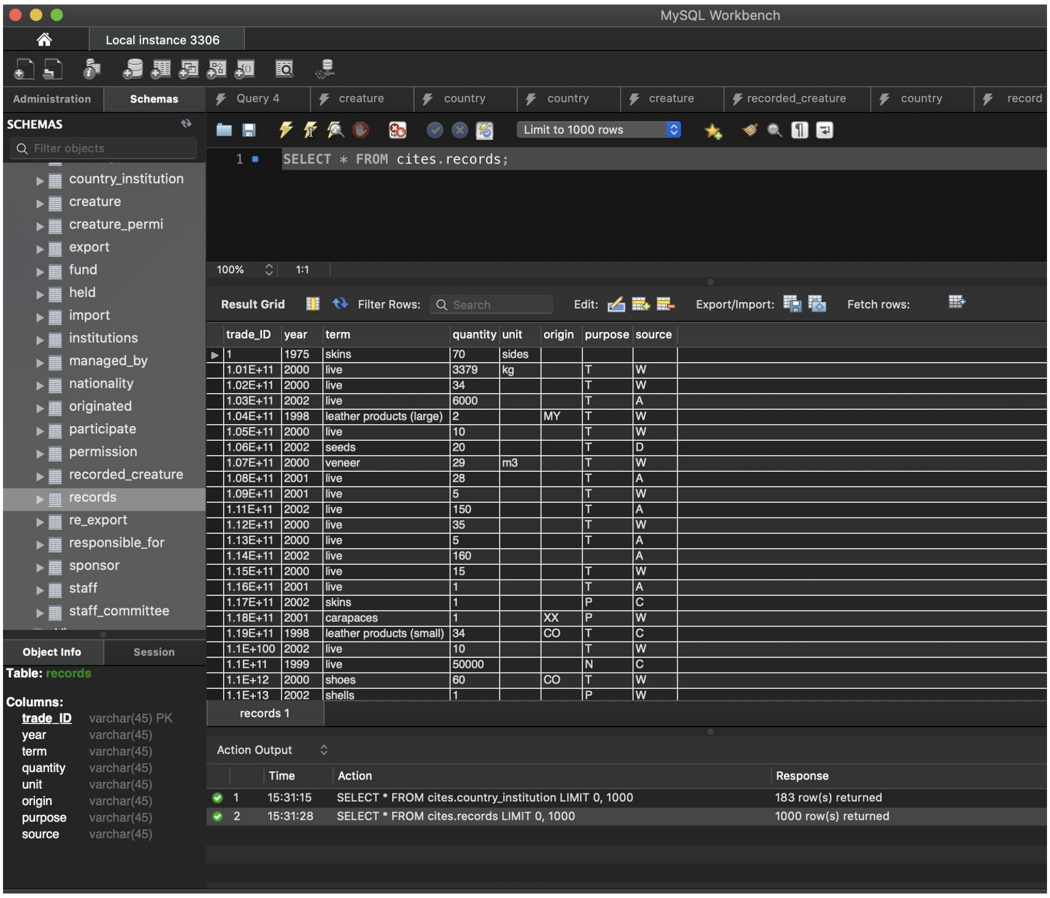
The following figures only show partial information of the data and query operation due to page limits, for more information please check the completed project on the github

https://github.com/ccjeff/CITES_database

[2] Database and Query Display Figure







[3] Partial Metadata CSV File Snapshot

Figure 3.1: Rollback Result of Country, Creature and Year

Year	Appendix	Taxon	Class	Order	Family	Genus	Term	Quantity	Unit	Importer	Exporter
1975	II	Caiman spp.	Reptilia	Crocodylia	Alligatoridae	Caiman	skins	70	sides	CH	XX
1975	II	Caiman spp.	Reptilia	Crocodylia	Alligatoridae	Caiman	skins	10	sides	CH	FR
1975	II	Chelonoidis carbonarius	Reptilia	Testudines	Testudinidae	Chelonoidis	live	16		CH	DE
1975	I	Crocodylus cataphractus	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	717		CH	FR
1975	I	Crocodylus cataphractus	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	45		XX	CH
1975	I	Crocodylus moreletii	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	25		CH	FR
1975	I	Crocodylus niloticus	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	383		CH	FR
1975	I	Crocodylus niloticus	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	12		CH	DE
1975	I	Crocodylus niloticus	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	121		XX	CH
1975	II	Crocodylus novaeguineae	Reptilia	Crocodylia	Crocodylidae	Crocodylus	skins	94		CH	FR
1975	II	Crocodylus novaeguineae	Reptilia	Crocodylia	Crocodylidae	Crocodylus	bodies	6		CH	PG
1976	II	Panisea spp.		Orchidales	Orchidaceae	Panisea	live	5		GB	IN
1976	I	Elephas maximus	Mammalia	Proboscidea	Elephantidae	Elephas	live	1		AU	MY
1976	II	Pan troglodytes	Mammalia	Primates	Hominidae	Pan	live	3		GB	IE
1976	II	Leopardus wiedii	Mammalia	Carnivora	Felidae	Leopardus	skins	1365		GB	SR
1976	II	Rebutia spp.		Caryophyllales	Cactaceae	Rebutia	live	1		BE	GB
1976	II	Panthera tigris altaica	Mammalia	Carnivora	Felidae	Panthera	live	2		SE	FI
1976	II	Pholidota spp.	Mammalia	Pholidota			live	2		GB	IN
1976	III	Balaenidae spp.	Mammalia	Cetacea	Balaenidae		bone carv	4		US	CA

Figure 3.2: Staff Information CSV File

staff_ID	last_name	first_name	gender	age	position	salary	dimission
1	Hatton	Patricia	female	47	chief reacher	20279	NO
2	Boyer	John	male	47	reacher	18391	NO
3	Pate	Robert	male	54	reacher	17639	NO
4	Thomas	Loren	male	56	reacher	16537	NO
5	Mezzatesta	Laura	female	51	reacher	18650	NO
6	Boggess	Eugene	male	55	reacher	17157	NO
7	Flaherty	Eulalia	female	49	reacher	19306	NO
8	Smith	Robert	male	50	reacher	18636	NO
9	Howard	John	male	45	reacher	16711	NO
10	Chiarello	Theresa	female	57	reacher	17632	NO
11	Kurtz	Austin	male	53	reacher	19685	NO
12	Williams	Lloyd	male	48	reacher	17169	NO
13	Pasquale	Colleen	female	47	reacher	18122	NO
14	Lagasse	Raymond	male	48	reacher	16258	NO
15	Ramirez	Shaun	female	50	reacher	17268	NO
16	Campbell	John	male	54	Research Assistant	12959	NO
17	Martin	Crystal	female	46	Research Assistant	12800	NO
18	Callicutt	Kelly	female	54	Research Assistant	10707	NO
19	Goldson	Stephanie	female	47	Research Assistant	13099	NO

Figure 3.3: Country and Area CSV Information File

country_name	ISO	region	convention_type	Date of joing	Entry into force	is_member	contribution
Afghanistan	AF	Asia	Accession	30/10/1985	28/01/1986	yes	422
Albania	AL	Europe	Accession	27/06/2003	25/09/2003	yes	483
Algeria	DZ	Africa	Accession	23/11/1983	21/02/1984	yes	8,326
Angola	AO	Africa	Accession	02/10/2013	31/12/2013	yes	603
Antigua and Barbuda	AG	Central and South America and the Caribbean	Accession	08/07/1997	06/10/1997	yes	121
Argentina	AR	Central and South America and the Caribbean	Ratification	08/01/1981	08/04/1981	yes	55,207
Armenia	AM	Europe	Accession	23/10/2008	21/01/2009	yes	422
Australia	AU	Oceania	Ratification	29/07/1976	27/10/1976	yes	133,341
Austria	AT	Europe	Accession	27/01/1982	27/04/1982	yes	40,847
Azerbaijan	AZ	Europe	Accession	23/11/1998	21/02/1999	yes	2,956