Provincial Cultural Relic and Museum Management System

Xiaonan Qi
the Chinese University of Hong Kong, Shenzhen
School of Data Science
Student ID: 118010238
Email: 118010238@link.cuhk.edu.cn

Jingyu Li
the Chinese University of Hong Kong, Shenzhen
School of Data Science
Student ID: 118010141
Email: 118010141@link.cuhk.edu.cn

Ruichun Yang
the Chinese University of Hong Kong, Shenzhen
School of Data Science
Student ID: 118010368
Email: 118010368@link.cuhk.edu.cn

Zeyin Zhang
the Chinese University of Hong Kong, Shenzhen
School of Data Science
Student ID: 118010430
Email: 118010430@link.cuhk.edu.cn

***Abstract*—A practical museum database system is of great significance and necessity. In this report, we present our design and implementation for the provincial cultural relic and museum management system. Starting from motivation, we continue to discuss database structure and data origin. GUI and SQL functions will then be exhibited. Last, the data mining technique is utilized.**

***Index Terms*—museum, antique, database, E-R diagram, SQL, data mining**

I. Introduction & Motivation

Museums are one of the most important public institutions in society. They not only engage visitors, promoting the enjoyment quality of their free time, but also foster deeper understanding and cultural sharing spirits of antiques. The museum helps preserve the tangible and intangible evidence of civil development. It records the growth of a city, in memory of its success and suffering. It has great historical values as well as an irreplaceable position in society.

Due to the increasing number of antiques and complex management requirements of the museums, it is difficult to handle the information by hand. With the rapid development of information technology, more and more museums choose to record the data in a database and provide an API for accessing. However, there are several problems in the nowadays museum database systems. First, the distribution of antiques is dispersed. Users need to visit many separate websites of different museums to collect sufficient information on a specific antique, which is time-consuming and troublesome. Second, much paper information is still not recorded. Third, there is not a uniform platform to manage the warehouse, workers, or exhibition hall information. Last but not least, it lacks a convenient user interface to manipulate the data.

Based on the problems raised above, we create our powerful provincial cultural relic and museum management system. It mainly contains five functional modules which are antique module, museum module, show module, warehouse module and worker module. Each module is responsible for a specific field. We also design a reasonable Entity-Relation diagram (E-R diagram) to illustrate the database structure. The diagram is logically clear and functionally useful. We maintain our database with BC-NF normalization form, which reduces the extra operations or failure of updating and deleting records. In the dependency schema, every determinant is a super key. We populate our database with realistic data from several famous museums in Shannxi Province. The crawl technique is utilized. Besides, we also design a simple and pretty graphical user interface. Users can perform data creation, modification, deletion and basic query function by just clicking some buttons and entering the necessary information. Advanced events like data analysis and data mining are also supported.

Our provincial cultural relic and museum management system provide great benefits for society. For visitors, they can search the antique, museum, or show information freely and conveniently. A copper coin lover can easily get to know where these coins are stored and their detailed descriptions to have a further visit. For managers and administrators, they can manipulate the worker, warehouse, show data freely. For example, create a show with a relics protection theme or create a new warehouse storing antiques. For researchers, they can perform high-level data analysis and mining techniques to enhance their academic achievement. We show an example of data mining about the antique line in the report.

With the powerful functions and enchanting features mentioned above, we believe our museum management system can stand out.

II. Database Structure Design

A. Requirement & Specification

To manage antiques, the system requires antique, museum and warehouse information. The antique item should contain a name, classification and current condition attribute for better management and age, dig_place and features for research

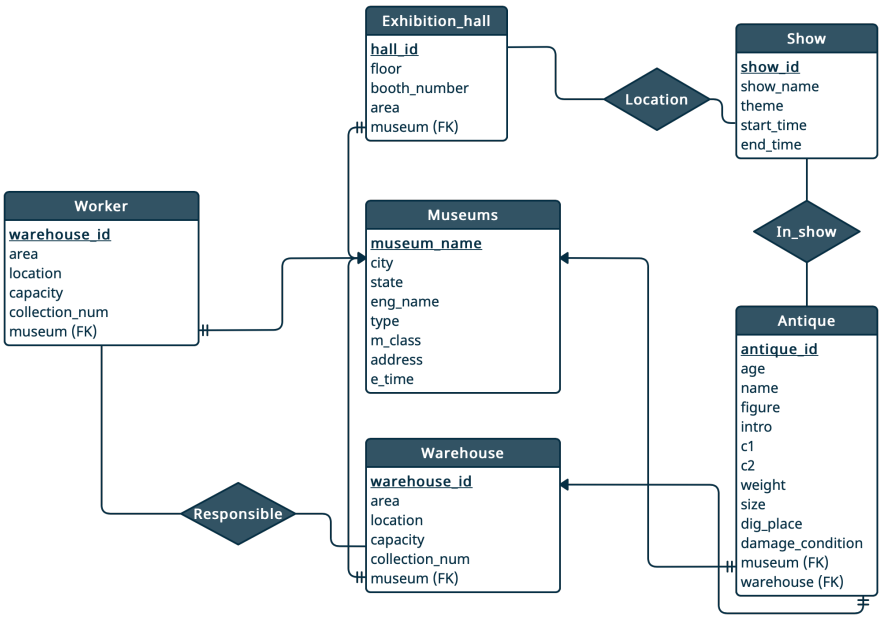

Fig. 1. ER Diagram of Provincial Cultural Relic and Museum Management System

purpose. The museum object requires basic information for registration, including museum name, city, current state, level, detailed location and establish time. The warehouse, which is used to preserve antiques, should involve information for administration, such as identification number, area, location, maximum capacity, current number of collections and which museum it belongs to. Moreover, since the staff is an important component of antique museums, worker information is also required. For a comprehensive management of worker activities, worker id, name, position, contact information, salary, age, native, education, major and other detailed personal information should be added into the system. Also, antique show for art and education purpose is a necessary activity in the museum. Therefore, the id, name, theme and time of the show need to be recorded. The exhibition halls, which are separate rooms in the museum that display the specific theme of shows, are required to arrange antique shows reasonably. Its basic information should contain id, floor number, number of booth inside, area and which museum it belongs to.

B. Entity-Relation Diagram

The ER-Diagram(Fig.1) of our system contains six entities and three relations. The entities are museum, antique, show, exhibition hall, warehouse and worker. The In_show relation records what antiques are in which show. The Location relation records the exhibition hall where show displays. The Responsible relation records which warehouse the workers are responsible for.

Some assumptions are declared in the followings:

1) The museum name is unrepeatable.
2) The classification 1 of antique includes organic class, inorganic class and composite class.
3) One antique can only appear at one show at one time. Therefore, the In_show relation is in one-to-many type.
4) One show can only display in one exhibition hall at one time and one exhibition hall can accommodate one show at one time . All shows need to participate in the Location relation while not all exhibition halls need to. The Location relation is in one-to-one type.
5) One warehouse worker only need to be responsible for one warehouse and one warehouse requires multiple manager. Therefore, the Responsible relation is in one-to-many type.

Museum Most antiques are preserved and protected in the museum. Hence, the museum can be viewed as a sub-unit under the administration, to which antiques directly belong. Notice that only state-owned museums are considered in this project. Since non-state-owned museums' collections do not belong to the country [6], its collection information is not necessary to be reported to Provincial Cultural Relics and Museum Administration.

For the museum entity, it has name, city and state attributes. Name is used to store museum names. As a museum does not have the same name as other museums, the name attribute is set as the primary key of the museum entity. The city indicates which city museum belongs to, since all state-owned museums are directly led by the city's government department in charge of cultural relics. Since state-owned museums are required to have a fixed site [6], the city attribute is not allowed to modify.

Attribute state is designed to prevent inappropriate insertion on exhibition_hall entity. The exhibition hall is a physical concept which indicates the interior design and site segmentation of the museum. Theoretically, it won't change once the museum was inaugurated. Therefore, for a museum's

exhibition hall information entry, it should be a one-time operation. When a museum's info is inserted into the database, the state is default set as true. After insertion of related exhibition hall information, the state is set as false. Insertion of exhibition hall info is not allowed under false state. This attribute can only change by users with high authority.

Antique The antique entity is of great importance for provincial cultural relics. We aim to design an antique table that can store the info that satisfies most museum's standards. We referenced the written cultural relic information record sheet of Collections of the Palace Museum and designed the following attributes for the antique entity. They are age, name, figure, intro, c1, c2 [4], weight, size, dig_place and damage_condition [5]. Age is used to store the dynasty information. Name is the antique's name, which should not be NULL under any case.

Besides, museum and warehouse attributes are used for indicating the museum and the warehouse that an antique belongs to. Since it is possible for one antique to have all the above attributes (name, weight, size, etc.) the same as another one, attribute antique_id is added to identify a specific antique. Hence, antique_id is set as the primary key of the antique table.

Warehouse The warehouse is an important component of the antique museum. To perform comprehensive management of warehouses, some information is required in the following. The area attribute is an objective attribute of the warehouse, which is related to the estimation of the capacity of warehouses. The location attribute provides a simple and straightforward way to locate the warehouses. In this case, people can plan the route and distance of the delivery of cultural relics.

The capacity and collection_num provides information to better manage the storage of cultural relics. The capacity defines the max capacity of a warehouse to store antiques while the collection_num presents the current storage number. The administrator can directly see the fullness condition of each warehouse and plan a reasonable storage allocation for newly coming antiques.

Worker The worker is an indispensable part of the museum. The worker in the museum should include the curator, receptionist, warehouse keeper, relic conserver, etc. To better satisfy the management requirements to distinguish among different working positions, the attribute position records workers' positions and thus the user is able to search the groups of workers in a certain position.

Besides, the attribute telephone records the contact way of workers. The attribute salary provides the necessary information to calculate the appropriation expenditure on worker salary. The museum, as a foreign key, correlates the certain warehouse with the museum to which it belongs.

Exhibition Hall The entity is for the hall in the museum to display a certain topic of the antique show. The entity includes attributes hall_id as the primary key, floor, booth_number, area and museum as a foreign key.

Since the museum requires to display long-term to short-term shows, the museums require the exhibition hall to place a certain group of antiques with specific topics. To relates the certain hall with the specific show displayed previously or currently, a relation Location is provided.

The attribute"floor" records the number of floors of certain halls. The booth_number records the maximum capacity of exhibit number. "museum" locates the museum where the hall is in. The information above is designed to better manage the exhibition halls and record the current state of them.

Show The entity is for the antique show that was or is on display. The entity includes attributes show_id as the primary key, show_name, theme, start_time and end_time. We assume that the show name and theme could be repeatable. Since the users have requirements to inquire the name, theme and time of the show, the attributes are designed to provide necessary information to the users. Moreover, the museum can arrange the time table and sites with the information provides by start_time and end_time.

Location The table is for the relation between the two entities: exhibition hall and show. Since each show needs to be allocated a certain exhibition hall to display and the show table records both previous and current shows, their relation is many-to-many. Moreover, all shows should participate in the relation while not all exhibition halls need to. The table includes two primary attributes coming from the exhibition_hall table and show table respectively: hall_id and show_id. In this way, the system can better manage the place of the show among all exhibition halls.

In show The table is for the relation between the two entities: antique and show. Since each show has a certain group of antiques to display and each antique can appear in several shows, previously or currently, their relation is many-to-many. Moreover, all shows should at least have some antiques while not antiques need to be displayed in the show. The table includes two primary attributes coming from the Antique table and Show table respectively: antique_id and show_id. In this way, the system can better manage the status of antiques and shows. Furthermore, it can satisfy users' requirements to search for the basic information of antiques that are on display in certain shows.

Foreign Some relationships between entities are presented as adding foreign keys, including the relations between museum and warehouse, museum and worker, museum and exhibition hall, museum and antique as well as antique and warehouse. This design is because these relationships are in one-to-many type. Using foreign keys can minimize the data redundancy and improve the referential integrity —— the accuracy and consistency of data within these relationships.

C. Schema & Normalization

Fig. 2. Relation to Foreign Key

Based on the above ER diagram, we carefully adapt them into the relation schema. The cases can boil down to two kinds of issues. At the outset, if the relation is one to one, then we don't need any further revision but just transform them directly. However, there are always cases that the relation is in one-to-many or many-to-many. When the relation is one-to-many, typically we store the primary key of the 'one' side as the foreign key for the 'many' side, which will help us to refer to the one-side schema conveniently. More importantly, if the relation is many-to-many, we decide to create a new schema for that relation and store the primary key of both sides. Later on, if we are going to refer to the schema with the other one, we will turn to this new relation schema for reference.

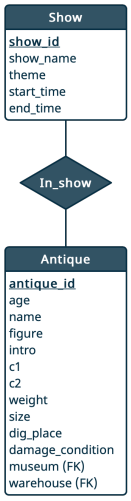

Fig. 3. Relation to Schema

To make the search and update as simple as possible, it's necessary for us to reorganize the original data set to a database that is carefully decomposed to avoid unnecessary redundancy and improve the coherence among the data. For our database, we use lossless decomposition to eliminate non-prime attributes dependency and carefully design the primary key to remove any prime attributes partial dependency. As a result, the database is in BCNF and the schema is shown below:

Museum(***museum_name(PK, FK), city, state, eng_name, type, m_class, address, e_time***)
Warehouse(***warehouse_id(PK, FK), area, location, capacity, collectio_num, museum(FK)***)
Worker(***worker_id(PK), name, salary, nationality, live_address, age, native_place, major, education, position, telephone, museum(FK)***)
Exhibition_hall(***hall_id(PK), floor, booth_number, area, museum(FK)***)
Show(***show_id(PK), show_name, theme, start_time, end_time***)
Antique(***antique_id(PK), age, name, figure, intro, c1, c2, weight, size, dig_place, damage_condition, museum(FK), warehouse(FK)***)
In_show(***show_id(PK), antique_id(PK)***)
Location(***show_id(PK), hall_id(PK)***)
Responsible(***worker_id(PK), warehouse_id(PK)***)

D. Index & Hashing

To increase the strength of the searching method, we added the necessary index into the database. Here, each antique is equipped with an antique index since most data in the database is related to the antiques. The index here is quite dense among every search-key value. They mainly act as primary indexes. Moreover, we also add other indexes such as date-time. Although they are quite sparse, it will help us to enhance the strength of the searching method like refine the scope of searching.

Considering the trade-off between search performance and system cost, we design a hash search using the antique name as the key for the antique table. There are three reasons for this design:

1) The antique entity is the core component of the system and it will be accessed frequently.
2) The table contains a large number of record and it requires a quick search to improve the searching performance.
3) The attribute antique name is chosen as the key because in most cases the user who is browsing the museum website search antiques by their names instead of other attributes.

The supported Chinese character coding in the system employs GB2312 standard. The basic strategy of hashing function design for GB2312 takes the formula [3]:

$$GBIndex = (C1 - 176) * 94 + (C2 - 161) \tag{1}$$

where C1, C2 represent the first and second byte respectively.

Assume that the input Chinese character as W. Then we can assign it to [3]

$$GBIndex = ((unsignedchar)W.at(0) - 176) * 94 + (unsignedchar)W.at(1) - 161 \tag{2}$$

III. Database System Preparation

A. Database Instance Deployment

For the convenience of database management among group members, instead of building up the database system on one team member's personal computer, we choose to purchase a remote cloud mysql database for project usage.

We purchase a Tencent Cloud Database, which is deployed in Guangzhou using MySQL 8.0 version. The configuration of the database is one CPU core with 1000MB memory and 25GB storage.

Database can be connected via MySQL workbench with below information:

TABLE I
Database Connection

Domain name	gz-cdb-0b3c5ux1.sql.tencentcdb.com
Port	58931
Account	CSC3170
Password	adminCSC3170

B. Data Prepare & Import

To better test the functionality of the database, our group populated our database with an abundant amount of real data. As for some data that is not able to collect from the internet, or the collection might cause privacy problems, e.g. collecting personal information of workers, we generate some fake data based on data requirement and characteristic.

1) Real Data Collection: The **Antique** table and **Show** table contain real data collected from the internet.The collection procedure mainly contains five steps.

The first step is to choose Shanxi province as the target province of the provincial cultural relic and museum management system. Up to 13 dynasties' capitals located in Xian which is the capital city of Shanxi province. Many cultural relics and state-owned museums are located here. Therefore, it is possible for us to get more antique-related data from Shanxi province.

Next, we crawl the antique data from the SHANXI HISTORY MUSEUM, BAOJI BRONZE WARE MUSEUM, XI'AN BEILIN MUSEUM, HAN YANG MUSEUM and XIANYANG MUSEUM using scrapy package. These five museums are chosen for they are the representative museums of Shanxi province, and because they are located in different cities of Shanxi province which can better test the functionality of our provincial cultural relic and museum management system.

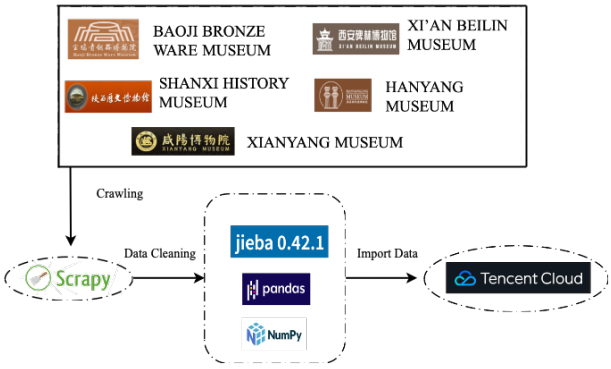

Fig. 4. Real Data Collection

TABLE II
Museum Website

Museum Name	Website
SHANXI HISTORY MUSEUM	http://www.sxhm.com
BAOJI BRONZE WARE MUSEUM	http://www.bjqtm.com
XI'AN BEILIN MUSEUM	http://www.beilin-museum.com
HAN YANG MUSEUM	http://www.hylae.com
XIANYANG MUSEUM	http://www.xybwg.cn

Thirdly, since the crawled data naturally does not fit in the designed schema, data cleaning is needed. Data cleaning mainly contains below three steps.

1) Remove the useless html language symbols. For example, \xa0, \u3000. Also remove some empty blankets.
2) Use jieba library in python to divide Chinese sentence into words and extract useful information that can fit in schema. For example, Chinese word whose PinYin is "Chutu" indicates the dig_place attribute of antique item. The generated word list of jieba that with index just before or behind word whose PinYin "Chutu" might contain the useful dig_place information based on the paragraph written structure of that museum.
3) Store the "cleaned" data into .csv files.

Last but not least, generate .csv files whose column is the same as **Antique** and **Show** tables' attributes based on the collected real data. Some attributes of in **Antique** schema is hard to retrieve from the internet, e.g. damage_condition attribute. For this case, that attribute is set as None value in python.

Totally, 432221 antique items are collected from 45 antique types. As for show, 19 live shows from the above five museums are collected.

2) Realistic Data Generation: The worker information is generated by codes instead of crawling from websites.

The worker contains information about name, position, telephone, salary, nationality, native place, education, major and address. The education is bachelor degree and the nationality is 'Han' by default. For position, salary, native place and major, we create reasonable sets to choose from. For the attribute of each worker, the code random an integer that decides which element in the set should choose. For example, the major set

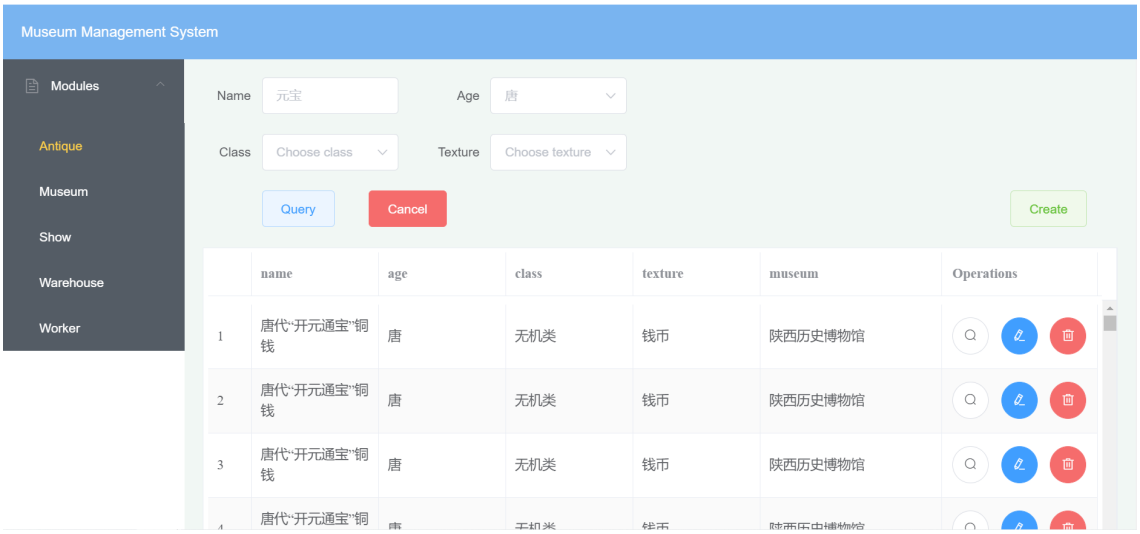

Fig. 5. GUI of Provincial Cultural Relic and Museum Management System

contains 'business management', 'corporation management' and 'material management'. If the random number is 2, then the corresponding worker has the major 'material management'. The generation of names is more complex. You can see the source code in the Appendix.

3) Data Import: Data import mainly uses two methods: import via MySQL workbench or import via a python script file.

As mentioned in the previous part of the report, .csv files same as schema are generated. One simple way to import data is to use MySQL workbench to transform the .csv file into the database directly. One just needs to choose the correct upload file and match each column in the chosen .csv file with the correct attribute of the schema. Then, MySQL workbench will do all the transform work for you. Although this method is simple and works well for most tables with less than 10,000 records, when it comes to the table with a large amount of data, it becomes extremely slow. As for the antique table, 2 hours can only import around 16,000 records.

There are more than 430,000 records for antiques. Therefore, import python via script file is more applicable. The python script file mainly does the following things:

1) Connect to the remote database using mysql.connector library.
2) Read targeted .csv file line by line. Place each line data into SQL insert command. For example, for the insertion into Antique schema, this command should be:

```
INSERT INTO Antique ('antique_id', 'age', 'name',
    'intro', 'c1', 'c2', 'weight', 'size', '
    dig_place', 'damage_condition', 'museum', '
    warehouse') VALUES (%s, %s, %s, %s, %s, %s, %
    s, %s, %s, %s, %s, %s);
```

3) Pipe up all the insertion commands. Commit insertion after certain amount of insertion commands.

Although using python script is much faster, it still needs around 2 hours, it is recommended that the commit should be executed several times to avoid network failure, e.g. commit once after insertion of 10,000 records.

Besides, for graphic data, MySQL workbench does not support bulk upload. Therefore, it also needs to use a python script to import data. One just needs to set the attribute of the item as the absolute path of the figure where the is stored.

C. Graphic User Interface

In this part, the graphical user interface of the provincial cultural relic and museum management system will be introduced.

The system is written in **Vue** framework based on **html**, **css** and **javascript**. Since this framework is suitable for building a quick and agile website, the style of the system is simple and clear. It has a baby blue header and a grey sidebar for navigation. The background of each page is light green. The colorful buttons and input form are aligned in order. The table, which exhibits the query result, lies below the searching conditions. The whole front-end is pretty user-friend and wisely organized.

There are 5 modules in the system in total (antique, museum, show, warehouse, worker), shown in the left navigation sidebar. There are also hidden pages for showing the query results or perform data manipulations. Here we will briefly introduce the basic components in every part with some screenshots partly.

1) **Antique module** This module is used for searching a specific antique across different museums. There are 4 input fields, name, age, class and texture. They are attributes of antique. There are three buttons of query, cancel (enter again) and create. The table returns the query results. Three buttons are provided for users to manipulate the records, which are check details, edit records and delete records.

2) **Museum module** This page is relatively simple. There is one select box for city. One green button is set for creating new museum and one search button. The table is for returning the results which is similar with antique module.
3) **Show module** User can input name, theme, start time and end time to search a show. For the date selection, user can specify whether the date is earlier or later than the target date. Other settings are similar.

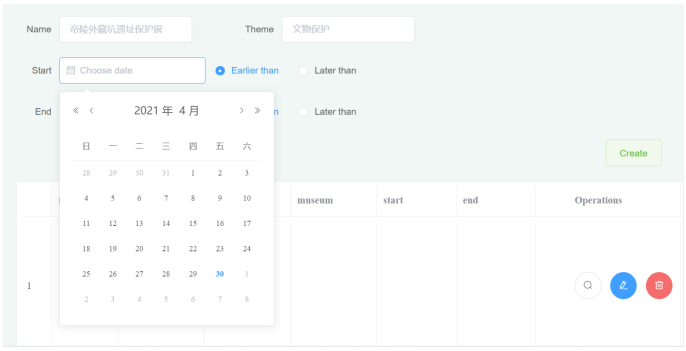

Fig. 6. show modules

4) **Warehouse module** This page has abundant choices for searching such as ID, area, location and capacity, museum, collection number. Each has four conditional options.

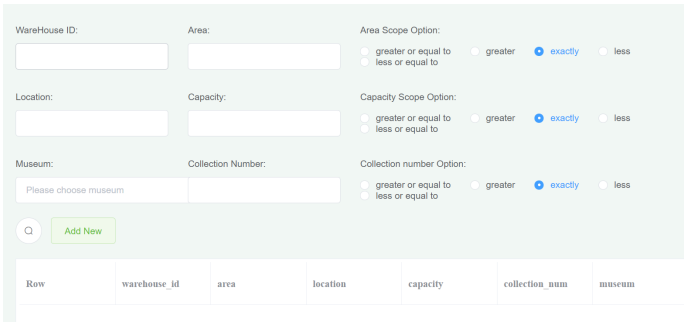

Fig. 7. warehouse modules

5) **Worker module** Administrator can input the worker ID, telephone and so on to obtain the information of the worker. Pay attention that user needs not to enter all the provided attributes. Even for a single attribute like name, user can still enter partly. The search is quite intelligent.
6) **Details page** Details page will appear after user clicks the check details buttons. As the name suggests, details about the item will show. Take the antique as an example, there is an image as well as other attributes listed in the detail page. User can return the main page by clicking the yellow button.

Fig. 8. Antique details

7) **Creation page** Creation page is used to create a new record in the database (antique, museum, warehouse...). Here the creation for a museum is shown. In real life scenario, a museum will be added to the system. Administrator just needs to enter all the attributes completely and click submit.

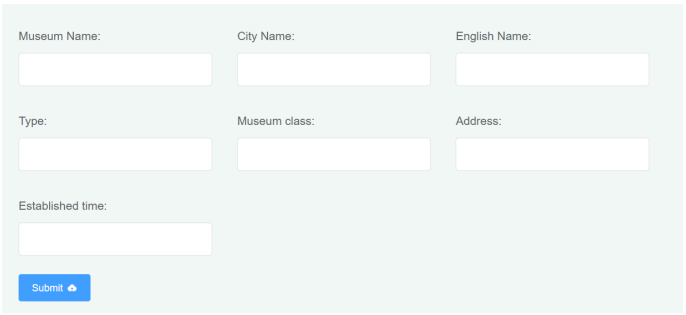

Fig. 9. Edit museum

In conclusion, our graphical user interface is quite user-friendly and logically clear and easy to use. Due to the page limit, we would not show all the pages. If you are interested in it, you can connect to us and access our websites to experience it.

IV. SQL Function

This section is mainly about SQL functions. Many SQL requirements and their back-end SQL sentences and corresponding results will be demonstrated. They include operational queries as well as analytic queries. All the functions are implemented in our system and can be accessed in the GUI.

A. Operational Queries Design & Implementation

This part shows the basic operational queries including four common manipulations of insertion, deletion, updating and searching.

1) Antique: Users are able to obtain antique results by their name, age, class and texture.

1. List all antiques given the corresponding name, age, class and texture. (Or condition is used to deal with empty attributes)

```
SELECT weight FROM CSC3170.Antique WHERE name LIKE "%
    Copper Coin%" AND (age="Tang" OR age="") AND (c1="
    Inorganic" OR c1="") AND (c2="Coin" OR c2="");
```

Fig. 10. Antique results

2. Insert a new antique into the database

```
INSERT INTO `CSC3170`.`Antique` (`antique_id`, `age`, `
    name`, `intro`, `c1`, `c2`, `weight`, `size`, `
    dig_place`, `damage_condition`, `museum`, `
    warehouse`) VALUES ('BJQT451921', 'Qin', '
    QinShiHuang', 'Classical antique in Xian', '
    Inorganic', 'Statue', '100kg', '2.3m', 'xian', '
    great', 'ShanXi Historical Museum', 'wSXLS1');
```

3. Edit the antique damage condition information.

```
UPDATE `CSC3170`.`Antique` SET `damage_condition` = '
    little' WHERE (`antique_id` = 'BJQT432179');
```

4. Delete an antique (if it is lost or transformed to another museum)

```
DELETE FROM `CSC3170`.`Antique` WHERE (`antique_id` = '
    BJQT432179');
```

2) Museum: As for the museum entity, the users are allowed to search the museum based on the city where the museum is located.

1. List all museums given the corresponding city_name

```
SELECT FROM Museum where city = "Xi an"
```

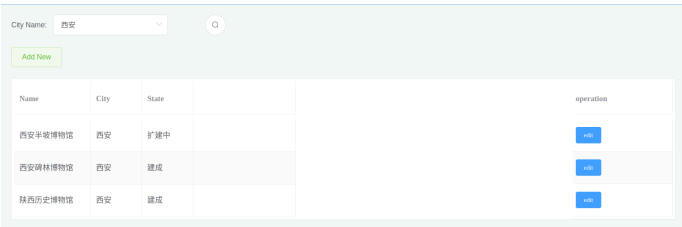

Fig. 11. Museum Result

2. Insert a new museums into the database

```
INSERT INTO Museums(museum_name,city,state,eng_name,
    type,m_class,address,e_time) values (VARCHAR,
    VARCHAR, INT, VARCHAR, VARCHAR, VARCHAR, VARCHAR,
    date);
```

3) Show: User can obtain show results by their name, theme, start date and end date.

1. Search the show by name.

```
SELECT * FROM CSC3170.Show where show_name='Civil
    Shanxi';
```

2. Create a new show as an organizer. To maintain the connection between show and its exhibition hall, the record must also be inserted into the relation table location.

```
INSERT INTO `CSC3170`.`Show` (`show_id`, `show_name`, `
    theme`, `start_time`, `end_time`) VALUES ('
    HY_SHOW01', 'Civil Shanxi', 'relics protection', '
    2021-04-28 00:00:00', '2021-04-30 00:12:00');
```

```
INSERT INTO CSC3170.Location (show_id, hall_id) VALUES
    ('HY_SHOW01', 'SXLS11');
```

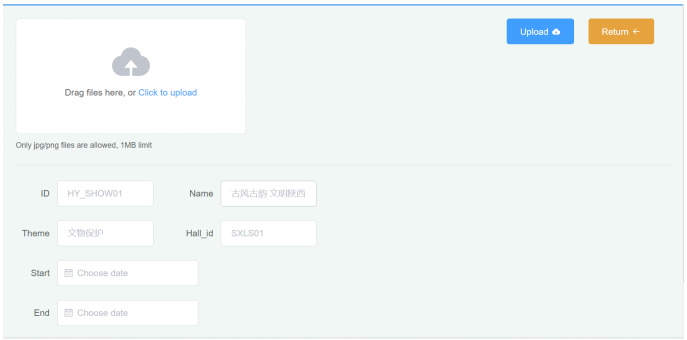

Fig. 12. Create show

3. Edit the show end time.

```
UPDATE `CSC3170`.`Show` SET `end_time` = '2021-04-30
    00:01:00' WHERE (`show_id` = 'HY_SHOW01');
```

4. Delete a specific show. Cancel due to special reasons.

```
DELETE FROM `CSC3170`.`Show` WHERE (`show_id` = '
    HY_SHOW11');
```

4) Warehouse: When it comes to the Warehouse entity, the users are allowed to nominate variuos attributes to search for the corresponding results.

1. List all warehouses whose area is larger than 10000 and capacity equals to 50000 and location is Xian and it's not full yet.

```
SELECT * FROM Warehouse where area >= 5000 AND location
    == "Xi an" AND capacity == 50000 AND
    collection_num < 50000
```

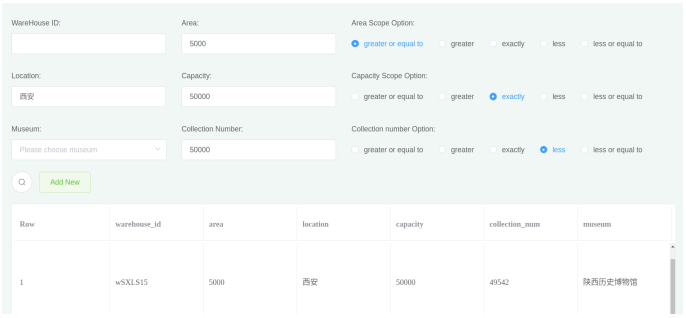

Fig. 13. Warehouse results

2. Insert a new warehouse under the museum ”Xi an Banpo museum”.

```
INSERT INTO Warehouse(warehouse_id,area,location,
    capacity,collection_num,museum) values ('wXABP1',
    5000, 'Xi an', 50000, 0, 'Xi an Banpo museum')
```

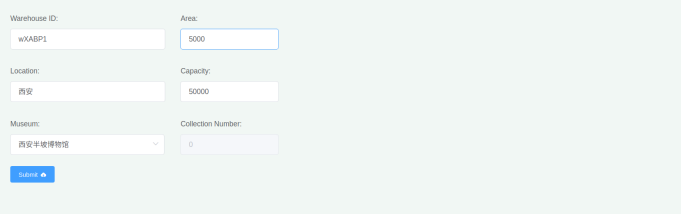

Fig. 14. Insert Warehouse

5) Worker: Finally, for the worker entity, the users are allowed to search nominated workers with various options. They are also allowed to update the chosen worker's information, add a new one into the database or delete a worker's information if he is fired from the museum.

1. Insert a new worker into the database for ”Xi an Banpo museum”

```
INSERT INTO Worker(worker_id,name,position ,telephone,
    salary,age,native_place,nationality,education,major
    ,live_address,museum) values ('pXABP1', 'ptr', '
    warehouse_manager', 13968754567, 3001, 40, 'Shan
    tou', 'han', 'ben ke', 'CSC', 'Green Avenue', 'Xi
    an Banpo museum')
```

2. List all the workers belonged to ”Xi an Banpo museum” with salary higher than 3000 yuan.

```
SELECT * from Worker where museum='Xi an Banpo museum'
    AND salary > 3000
```

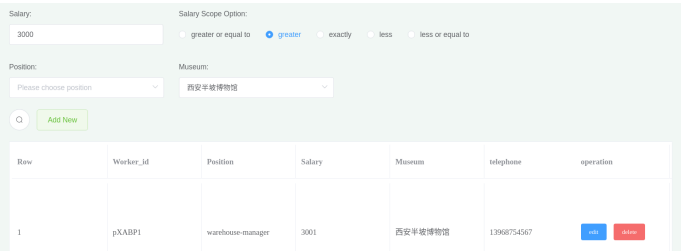

Fig. 15. Worker results

3. Edit the worker's information to raise his month salary to 5000 yuan and change his telephone number to 137618180666, the worker's id is 'pXABP1'.

```
UPDATE Worker SET salary=5000, telephone=13761880666
```

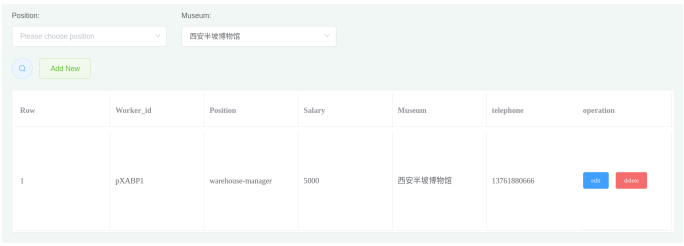

Fig. 16. Edit Worker results

4. Delete the workers whose worker_id is 'wXABP1' becuase he is fired.

```
DELETE FROM Worker where worker_id='wXABP1'
```

At this specific case, we also need to update the information in the responsible table.

```
DELETE FROM Responsible where worker_id='wXABP1'
```

B. Analytic Queries Design & Implementation

This part shows the advanced analytic queries aiming to obtain some deeper knowledge of the data. They summarize important features of the records.

1) Antique: List the antique name as well as the number of it to give searching recommendation. The greater the number, the more possible user would query it in our system. So it will appear in the front of the searching list. The count can also be viewed as the popularity of a specific antique.

```
SELECT name, count(name) FROM CSC3170.Antique group by
    name order by count(name) DESC;
```

name	count(name)
唐代"开元通宝"铜钱	286792
清代"嘉庆通宝"铜钱	12814
宋代"皇宋通宝"铜钱	7000
汉五铢铜钱	2454
宋代"建炎通宝"铜钱	2398
唐"开元通宝"铜钱	2288
宋代"元丰通宝"铜钱	2068
汉"五铢"铜钱	1638
宋代"元祐通宝"铜钱	1505
宋代"政和通宝"铜钱	1505

Fig. 17. Antique count

2) Show: Imagine the visitors would like to participate in a show in their free time. List the show information as well as the corresponding city and museum.

```
SELECT show_name, museum, city FROM CSC3170.Show
    natural join Location natural join Exhibition_hall
    natural join Museums;
```

3) Warehouse: List how many antiques are stored in a warehouse through foreign key.

```
SELECT warehouse, count(name) from Antique group by
    warehouse;
```

warehouse	count(name)
wSXLS62	480
wSXLS63	2423
wSXLS64	1540
wSXLS65	530
wSXLS7	3187
wSXLS8	1802

Fig. 18. Warehouse store

4) Worker: Considering such a case, the museum manager will select the top 5 warehouse managers with the highest salary and younger than 40 and get a list of them for future usage. For example, the manager will want their name and telephone so that he can call them to ask for their future planning.

```
SELECT name,telephone FROM Worker
where worker_id in (SELECT worker_id FROM
Worker where museum="Shanxi QingTong museum" AND age <
    40)
GROUP BY name ORDER BY salary DESC LIMIT 5
```

V. Data Mining

A. Experiment Motivation & Target

It is kind of cheating if data mining on self-generated data. Therefore, our group's data mining will not carried out on fake data (e.g. Worker table, Exhibition_hall table and Warehouse table). When we glancing through the antique data we collected, we find out that there are rich information that we can get from the name attribute of antique item. Many researchers named antique based on its most unique feature. For example, the enamel of porcelain can be found in naming a porcelain antique (e.g. Qing Dynasty Blue Enamel Coating Zun); painting subject can be found in the naming of Chinese painting antiques (e.g. Tang Dynasty Portrait of Noblewomen); decoration can also be found on many type of antiques including bronze ware, stone, China and so on.

Our group decide to make use of the decoration on antiques. We aim to build up a decision tree to classify antiques' dynasty based on the decoration upon it. We choose decoration instead of other characteristics for the below two reasons:

1) Decoration appears on more types of antiques compared with enamel which is almost restricted to porcelain, which allows us to set comparative experiment among different type of antiques.
2) Antique as old as Neolithic period (around 14,000 to 4,000 years ago) to Qint Dynasty (around 400 to 100 years ago) have decoration upon it, which means there are more possible classification labels for this classification task.

B. Method

1) Data Preprocessing: Different from training data from existing open source data set, we prepare the training and testing data set from scratch. The data set is created via below procedure:

1) Select all antique instances whose name contains Chinese character whose Pinyin is "Wen". Totally, 4782 items are selected.
2) The age attribute of antique items is used as the true label. The training input feature lies in the name attribute of antique items which needs data cleaning to extract.
3) Use jieba library to split the name attribute of all the selected antique instances. Choose the word in the generated jieba word list that with index just before the word whose PinYin is "Wen". Although jieba library performs quite well in splitting vernacular Chinese sentences, it turns out that it does not perform very well in splitting classical Chinese. Since the antique naming contains both vernacular grammar and classical Chinese words, sometimes jieba splitting word in the wrong place. In this case, mannual correction is needed.
4) Group the antique items after processing based on their types and store it into different .csv files.

Totally five .csv files are generated. One for all antique items with decoration information in its name regardless of its type. The other four for antique items with decoration information but restricted to stone antique, China antique, gold and silverware antique, and bronze ware antique respectively. The detailed information of these four data set is shown as below table.

TABLE III
Data Mining Data Sets

Data Set Name	Antique Type	Antique Item Number
Datamining.csv	All	4732
Datamining_1.csv	Stone	884
Datamining_2.csv	China	855
Datamining_3.csv	Gold and Silverware	200
Datamining_4.csv	Bronze Ware	1946

Notice that sum of item number in Datamining_1.csv to Datamining_4.csv does not equals to the item number in Datamining.csv. Some antique type have relatively small amount of antiques' name with decoration. The effect of these part of data will be analyzed in the outcome analysis part.

2) Training & Testing: Firstly, load data set and split data into train data set and test data set with ratio $\frac{3}{1}$, i.e. for Datamining_1.csv set, train data set should contain 663 items and test data set should contain 221 items.

Next, train data with decision tree model implemented in sklearn.tree library. Notice that the input feature are strings which is not supported type for build in method in sklearn.tree library, encoding decoration feature into int is necessary. Fit

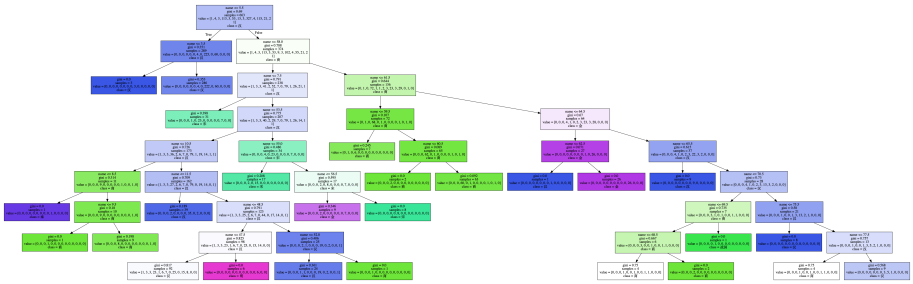

Fig. 19. Visualization of Stone Antique Decision Tree

the DecisionTreeClassifier with trained data feature and trained data label.

Finally, verify model's performance on test data. Get the predicted label of the test data and compare it with its real label. Then, calculate the test accuracy.

C. Outcome Analysis

The experiment mainly compares two parts. The first part is the comparison of model's accuracy among different antique types. The second part is the detailed analysis of the model with highest accuracy. Analyzed its splitting criteria and its behind possible historical support.

TABLE IV
TEST ACCURACY FOR DIFFERENT ANTIQUE TYPES

Data Set Name	Antique Type	Test Accuracy
Datamining.csv	All	67.20%
Datamining_1.csv	Stone	81.45%
Datamining_2.csv	China	75.00%
Datamining_3.csv	Gold and Silverware	70.00%
Datamining_4.csv	Bronze Ware	61.73%

1) Model Performance Comparison of Different Antique Type: From TABLE IV, we can see that among all models, Stone antique has the highest test accuracy. Bronze Ware has the worst test accuracy performance among the five tested models. That is probably because bronze ware popular in a relatively distant time (e.g. most collected data are in the Shang dynasty). The earlier bronze ware antiques were even harder to be collected. Therefore, the trained data is not balanced enough. Besides, considering the building technique of bronze ware, the lost-wax process means each bronze ware is unique, i.e. it is not possible to mass production bronze ware based on some prototype. The decoration is very diverse on bronze ware maybe for this reason.

Besides, despite bronze ware, other single type data set has better performance than the data set containing all antiques with decoration. As mentioned in the data processing part, there are some antique types with a relatively small amount of data with decoration. The classification of these antiques might be poor and therefore lower the overall accuracy of the model's performance.

2) Stone Antique Decision Tree Analysis: Using graphviz package to visualize the decision tree (Fig. 4.) and prints out the splitting criteria at each node. Then, remap the splitting criteria back to the original string decoration feature. Group the decoration feature based on its predicted dynasty and the result is shown as below TABLE VI.

TABLE V
DECORATION OF STONE ANTIQUES

Dynasty	Decoration
Shang (商)	丁纹,三鹤纹,乳丁纹,乳丁纹,云气纹,云纹,云雷纹,兽面纹,几何纹,双鱼纹,四兽纹,四神纹,回字纹,回纹,团花纹,国字纹,图章纹,夔凤纹,夔纹,奔鹿纹,狩猎纹,玄武纹,白虎纹,秧歌纹,章形纹,绳纹,网云纹,网格纹,葵花纹,虎纹,钉纹,雨云纹,青龙纹,鸟兽纹,鸵鸟纹
Song (宋)	云雷纹,云鹤纹,人物纹,代龙纹,网格纹,网纹,花卉纹,花叶纹,花纹,花鸟纹,荷花纹
Qing (清)	代龙纹,兽纹,兽面纹,奔鹿纹,字纹,射猎纹,山云纹,拥慧纹,指纹,方格纹,方纹,曲尺纹,朱雀纹,柿蒂纹,格纹,毛字纹,荷花纹,莲瓣纹,莲纹,莲花纹,葵云纹,虎纹,蝴蝶纹,字纹,字纹,豹纹,连珠纹
Qin (秦)	几何纹,凤纹,凤鸟纹,力士纹,动物纹,卧蚕纹,卷云纹,双兽纹,双马纹,双鱼纹,毛字纹,涡纹,葵云纹,葵纹,葵花纹
Han (汉)	涡纹,狩猎纹,白虎纹,石莲纹,秧歌纹
Jin (金)	连珠纹,重环纹,钉纹
Ming (明)	鸵鸟纹,鹿纹,麒麟纹,龙纹

By searching through some history research findings, we find out that the decoration classification verified some history research findings.

As stated in Wu and Li's [2] article, "Shang Dynasty antiques are mainly decorated with animals,'', real animals and mysterious creature". They also stated the most common decoration in the Shang dynasty. "For example, Dragon and Tiger decoration in common in Shang dynasty." Those decorations can be found in TABLE VI.

While in Gu's [1] finding, she finds analyzed Song dynasty's decoration characteristics and concluded that "Shang Dynasty antiques are mainly decorated with animals,'', real animals and mysterious creature". The "lotus", "flower", "flower leaf" and "flower bird" decoration verified Gu's findings.

The wrong classification cases are also analyzed. One common case is the wrong classification among close dynasties. This is reasonable, many decorations exist across dynasties. Another important reason is the lack of other features. Using only one feature to classify antique is hard to get high

accuracy. In the real case, archaeologists need to classify antiques' dynasty based on a lot of other features.

To sum up, this simple decision tree model aims to show that as the database system is populated with more antique records, it will be more beneficial to cultural relic researchers and lovers. For it can save their time in searching and collecting research material.

VI. Conclusion

Realizing the importance of museums and problems raised by the current database system, we design our powerful cultural relic and museum management system. The system contains five parts which are antique, museum, show, warehouse and worker. Each module is responsible for a specific field. We carefully design the E-R diagram according to the requirement. The structure is both clear and efficient. We maintain the normalization form as BC-NF to avoid modification and deletion problems. Besides, we set the foreign keys and relation based on the dependency. Indexing and hashing are also applied to improve data access performance. As for the data, we create our database on the cloud. The database is populated with realistic data crawled from different museum official websites. The GUI of the system is simple and pretty, easy to use. We clearly illustrate both operational and analytical SQL query functions, including objectives, SQL sentences and SQL results. Last but not least, we perform a data mining technique to explore the relationship between decoration and dynasty using the decision tree model. This part is a highlight for the bonus.

As for further improvement, even more realistic data and attributes can be added to complete and extend the database. More functions such as cross-module/join table query can be implemented. The GUI can become fancier.

In conclusion, we design and implement our cultural relic and museum management system with high quality. We expect it to be fantastic work.

References

[1] Li, G., (2011). Study on decorative patterns of Song, Liao and Xia Jin[D], Su Zhou University.

[2] Wu, C., Zhong Yang, L.,(2019).Study on decorative patterns of Shang bronze ware[J],Art Science and Technology,7

[3] L. Zhi, *Chinese Character Hash Problem*, CSDN, Dec. 15, 2011. Accessed on: May 1, 2021. [Online]. Available: https://blog.csdn.net/lizhi200404520/article/details/7074719

[4] Nan, W., (1999). The classification of modern cultural relics, Chinese Museum, 1.

[5] State Administration of Cultural Heritage of the People's Republic of China''*Specification for registration of cultural relics in the collection of cultural institutions*, 1st ed. Beijing: Cultural Relics Publishing House, 2013.

[6] Museum Regulations of PRC, 2015.