

The design of relational database and NoSQL database in travel agency database system

Chenxi Xu

Shandong Experimental High School, Jinan, China

shengxiaoxiao@xdf.cn

Abstract. The information management systems of travel companies are experiencing greater challenges and expectations as a result of the tourism industry's rapid growth and travellers' rising demand for personalized and interactive experiences. This paper aims to study the customer management system of travel agencies, taking into full consideration the characteristics and development trends of the tourism industry. Traditional customer management system uses relational database for data storage and related operations, which cannot fully meet the needs of customers and complex data in the new era. By analysing and comparing relational databases and NoSQL databases, according to their own advantages, a combined solution is proposed, which uses relational databases to store structured data and NoSQL databases to store unstructured data. On the basis of maintaining the original system management function, this solution greatly enhances the storage and processing of interactive data such as images and videos, and better meets the new demand of travel agency customers for sharing and interactive communication.

Keywords: Travel Agency, Relational Database, NoSQL.

1. Introduction

With China's economic growth and the rising living standard of the people, travel agencies flourish hugely. Tourism has become a rigid demand for people's better lives in a well-off society. During the 13th Five-Year Plan period, the average annual travel was more than 4 times. "14th Five-Year Plan" period, China will fully enter the era of mass tourism, tourism development is still an important period of strategic opportunities. We will accelerate the application and popularization of new technologies in the field of tourism, and enhance the development level of tourism through scientific and technological innovation. Enhance the experience and interaction of tourism products [1]. The rapid development of travel agencies is accompanied by the continuous progress of industry information. The standard travel agency database system has been based on relational databases for some time now, which primarily serve as data storage and manipulation tools for users. SQL databases were adequate at first, but as computer science technologies advanced and the Internet became more widely used, other factors began to emerge. These included the amount of data that was being stored, the intensity of user interactions with the data, the ability to store large amounts of photos and videos, and many other Big Data challenges. SQL databases cannot address these kinds of data as efficiently as NoSQL database [2]. Relational databases like MySQL are storing data only in organized form. Document-based data stores

like MongoDB, which is a type of NoSQL database can store huge volume of data which additionally have very powerful query engines and indexing features [3].

Nowadays, with the rapid popularization of the Internet and the gradual trend of online and instant social networks, travelers have more and more prominent needs for recording, sharing and interaction in the travel process, and they also pay great attention to the evaluation and sharing of other tourists in the process of travel planning. The traditional travel agency information management system for this part of the requirements is rarely involved, some want to deploy the relevant functions, but the traditional SQL database is not easy to achieve, mainly this kind of data is mainly a large number of photos, videos and other unstructured data, unstructured data writing, reading and real-time processing has been the weakness of traditional relational data. In contrast, NoSQL database has the characteristics of 2 distributed, easy to expand, flexible data, high performance, etc., and is more suitable for the storage and processing of unstructured data (including pictures, videos, etc.). Based on the requirements of the travel agency information management system, this paper adds user sharing and interaction functions to the common functions of the traditional system to solve the more urgent social needs of tourists. In terms of the implementation method, through the analysis of the functional characteristics of relational database and NOSQL database, the traditional data are stored and processed by SQL relational database, and the non-structured data (photos and videos, etc.) are stored and processed by NOSQL database, which effectively solves the difference between structured and non-structured data processing. It realizes the efficient processing of the two types of data respectively, and better meets the upgrading requirements of the travel agency information management system.

2. Database introduction

Databases can be split into two main categories. The following describes the definitions, advantages and disadvantages of the two types of databases, as well as the significance of the combination.

2.1. Relational (SQL) database definition

Relational (SQL) databases, relational databases are built on relational numbers. Based on the model, it is based on the mathematical concepts and methods such as set algebra. A database for managing data; It stores data in the form of rows and columns, basic units. As a table; In a relational database, there is a single entity and the relationships between entities. This logical structure's kind of representation is a two-dimensional table with several table forms. was transformed into a relational database; users may now add, remove, update, and check data in the library using SQL. At present, the most used relational database in the market is Oracle, Microsoft SQL Server, Access, IBM DB2, Sybase from Sybase Corporation and the open-source MySQL are relational databases.

2.2. Non-relational (NOSQL) database definition

The English interpretation of NOSQL translates as “non-relational”, also known as “Not Only SQL”. NoSQL is a broad word for database management systems that are distinct from conventional relational databases, which utilize SQL syntax to store and query data. NoSQL stands for a non-relational database. In contrast, NoSQL database systems include storable structured, semi-structured, and unstructured. A variety of database technologies for transforming and polymorphic data. Mainstream non-relational databases generally include several categories: key-value numbers. Database such as: Redis; column storage database such as Hbase; document databases such as: MongoDB; full-text search databases such as Elastic Search; graph database such as: Neo4j [4].

2.3. NoSQL database versus relational database

The use cases for relational databases and NoSQL databases are significantly different. Different application circumstances call for various database schemas, and only one is appropriate. Corresponding can more effectively play to database management's benefits. The benefits and drawbacks of NoSQL databases in the context of their individual situational uses.

As for the advantages of relational database: first, this kind of database is easy to maintain: A two-dimensional array represents the table structure in a relational database. The format is consistent, making SQL statement operations straightforward. Second, the SQL language is straightforward and simple to use: It is simple to learn and simple to use, and it can be used to query in a variety of difficult situations. Third, assisting with complicated operations: it can achieve a table and multiple tables between the complex. Miscellaneous query operations make it easy for beginners to learn SQL statements.

However, the disadvantages of relational databases are obvious: in the face of large amounts of data for efficient processing of access to read and write. This type of database has poor read and write performance. Second, because the relational database's table structure is a two-dimensional table structure and a table structure. Its flexibility is low since it is somewhat set. Third, when data is read and written with high concurrency in typical relational databases. The hard disk's input and output are limited.

As for the advantages of NoSQL database, first, the data storage structure is flexible: NoSQL databases have a variety of storage mechanisms high flexibility of data format, no fixed table structure and high scalability. Such as document storage mode, Key-Value pair storage mode, graph storage mode. Second, these databases have high scalability: NoSQL databases offer horizontal data scalability, as well as enormous volumes of data and many forms of data storage management. Third, supporting for data ultimate consistency: NoSQL databases are built on the CAP module. At the same time, it also supports the BASE distributed theory, which aims to achieve the final one of the data tropisms.

Despite the fact that NoSQL databases may be employed in a larger range of complicated data applications. However, there are still several flaws with NoSQL databases, the most frequent of which being the following: first, they do not support SQL Structured query language. Second, the NoSQL databases have inadequate feature support: Currently, the most common non-relationship databases are unable to enable transaction processing and can only provide restricted functional capabilities. Third, the majority of NoSQL databases are still in their infancy: hence, the supporting technological advancements, maintenance guides, and other technical knowledge still need to be enhanced and improved [5].

Relational databases and NoSQL databases have their own advantages and disadvantages and are suitable for different data and scenarios. In this case, can we consider the combination of NoSQL database and relational database, using relational database in strong consistency and high availability scenarios, and using NoSQL in scalability scenarios? The answer is yes, the current NoSQL database is still difficult to compete with relational database. However, it can make up for the performance and scalability of relational databases, so we can use NOSQL and relational databases in combination, each taking advantage of its strengths, we use relational databases when we need to use relational features, and we use NOSQL databases when we need to use NOSQL features. [6]

2.4. Introduction of MongoDB

MongoDB is high performance, easy to deploy, easy to use, scalable, and schema-free documentation. Row database, distributed file storage database written in C++ language, MongoDB can run on Windows and UNIX platforms. The main functions are: document storage (BSON); support dynamic query; for the document; create an index; support composite index; use geospatial indexes; new aggregation box; binary data storage (mass data, pictures, videos, etc.); automatic debris handling; update information in situ; implement sharding (automatic sharding); analyze query and copy data; built-in GridFS, which supports mass storage [7]. MongoDB is scalable, very powerful, easy to use, and well designed to manage storage for large-scale applications, and can run in a distributed and multi-platform environment. As a result, several major companies such as Foursquare, SAP, and GitHub have adopted MongoDB [8]. In this design, MongoDB is mainly used as the part of NoSQL.

3. Design and implementation of travel agency database system

3.1. Introduction

This article designs a project for a travel agency user management system. First, it is necessary to analyze the customers' instruction and determine the entities. Second, using E-R diagram to describe these entities and the relationships between them. Then, E-R diagram should be converted into relational model. Finally, after normalization, the database and tables are created and then implement the database using SQL language. As for the data such as videos and graphs, they are implemented by MongoDB.

3.2. Project Statement and Objectives

The database application for travel agencies serves the dual purposes of maintaining the data we produce and offering customers and employees an effective and efficient service. Table 1 shows the purposes of the system.

Table1. The purposes of the travel agency database system.

Number	The data to maintain
1	customers.
2	employees.
3	insurance.
4	car rental services.
5	bookings of the customers.
6	transportation.
7	tourist attractions.
8	hotels.
9	customers' sharing and interaction.

Based on the requirements above, the author determines several entities listed below:

Customer Employee Insurance CarRental Booking Tourist Attraction Transportation Train Flight
Bus Hotel Sharing

3.3. E-R diagram

Through the research, analysis and demand exploration of each entity to obtain each local E-R diagram. Then the overall E-R is generated by synthesizing each local E-R diagram.

A travel agency maintains a database of its customers. Each customer has their unique customer ID, first and last name, email, bank account, phone number, passport number and their driver's license number (if he/she has one).

Customers can consult travel agency employees before making travel bookings. The details on an employee include his unique employee ID, first and last name, email and phone number. Customers can also choose to buy insurance or not. The details of an insurance include insurance number, insurance company name, coverage and price. The travel agency can also provide car rental service for customers if they need it. The data will keep on each car-rental includes car plate, car model, rental location and the start & end date of the rental service.

Customer can make several bookings in the travel agency. A booking is associated with only one customer. The detailed information of booking includes unique booking ID, the start & and date and booking price.

The booking mainly includes three parts. First is the transportation which the customer chooses to take during the trip. The transportation must have its unique ID, departure date, time and location, arrival date, time and location, number of tickets and a total price. Note that a booking must include at least one mode of transportation. The travel agency offers different transportation ways, which include but not limited to trains, flights and buses. If the customer chooses to take a train or flight, he/she may also

choose a class (e.g. first, economy, business, etc.). Second is the reservation of hotels. Each hotel in the database has a unique hotel ID, name, address, room type, check in & out date, number of rooms and a total price. Third is the tourist attraction booking. A tourist attraction must include a unique ID, address, name, number of tickets and a total price. Note that a booking may include none hotel or tourist attraction, but there is also no limit to the number of hotels and tourist attractions.

Customers can share their travel experience by word, photos and videos, other customers can reply to sharing of somebody. The sharing system information includes unique ID, time, writer, replier, content (including words, photos, videos).

Figure 1 is the final E-R Diagram of the part of the relational database.

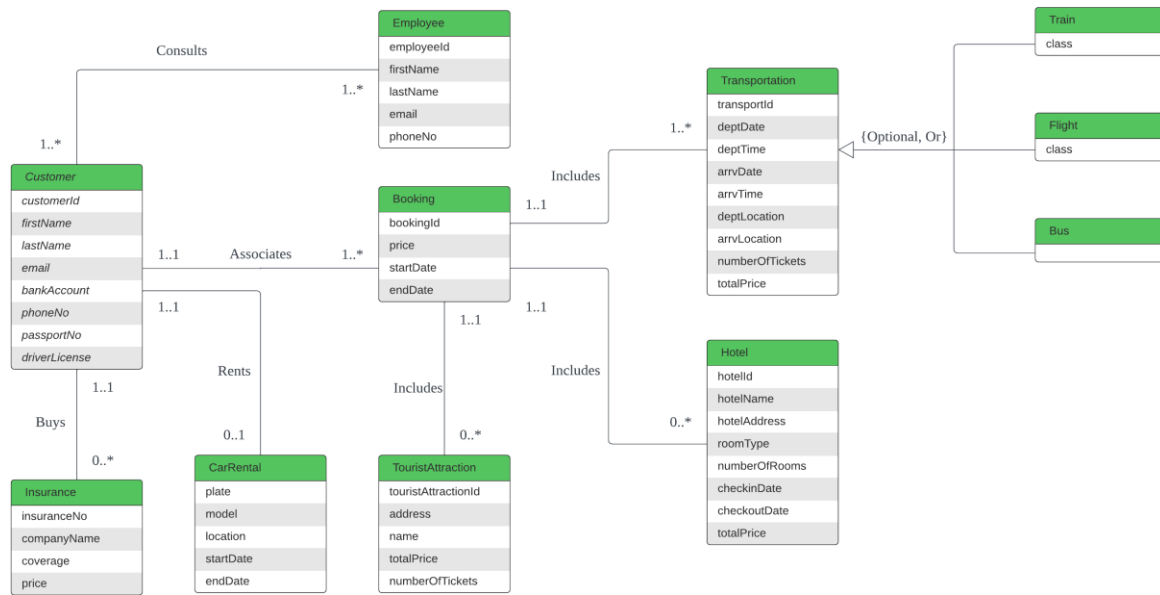


Figure 1. Final E-R Diagram.

3.4. Relation Model and MongoDB Model

Then deriving the relational data model from the final E-R diagram by applying different rules for converting 1-to-1, 1-to-many and many-to-many relation into relational data model.

a.Customer

(customerId {PK}, firstName, lastName, email, bankAccount, phoneNo, passportNo, driverLicense)

b.Employee (employeeId {PK}, firstName, lastName, email, phoneNo)

c.Consults (customerId {PK, FK}, employeeId {PK, FK})

d. Insurance (insuranceNo {PK}, companyName, coverage, price, customerId {FK})

e.CarRental (plate {PK}, model, location, startDate, endDate, customerId {FK})

f.Booking (bookingId {PK}, price, startDate, endDate, customerId {FK})

g. TouristAttraction (touristAttractionId {PK}, address, name, totalPrice, numberOfTickets, bookingId {FK})

h. Hotel (hotelId {PK}, hotelName, hotelAddress, roomType, checkinDate, numberOfRooms, checkoutDate, totalPrice, bookingId {FK})

i. Transportation (transportId {PK}, deptDate, deptTime, arrvDate, arrvTime, deptLocation, arrvLocation, numberOfTickets, totalPrice, bookingId {FK})

j. Train (transportId {PK, FK}, class)

k. Flight (transportId {PK, FK}, class)

l. Bus (transportId {PK, FK})

m. For sharing entity, we use MongoDB to implement it. first establish database Sharing [9], and then set up a collection: Sharingcollection.

3.5. Normalization

After finishing relational data model, each of the entities should be normalized into third normal form.

The main purpose of normalization is to eliminate unration parts in data dependency step by step to achieve “separate” in a way, which means that only let one relationship describes one concept or one entity. When one relationship describes more than one concept or one entity, it is necessary to be separated. In fact, normalization is conceptual simplification.

The results have been normalized into third normal form. Some of the results are listed as following.

(1). Customer (customerId, firstName, lastName, email, bankAccount, phoneNo, passportNo, driverLicense)

customerId \rightarrow everything else Candidate Key: customerId

(2). Employee (employeeId, firstName, lastName, email, phoneNo)

employeeId \rightarrow everything else Candidate Key: employeeId

(3). Consults (customerId, employeeId) Candidate Key: customerId, employeeId

(4). Insurance (insuranceNo, companyName, coverage, price, customerId)

insuranceNo \rightarrow everything else Candidate Key: insuranceNo

(5). Booking (bookingId, price, startDate, endDate, customerId)

bookingId \rightarrow everything else Candidate Key: bookingId

(6). CarRental (plate A, model B, location C, startDate D, endDate E, customerId F)

$A \rightarrow B, AF \rightarrow CDE$ Candidate Key: AF

R1: AB

R2: AFCDE

(7). TouristAttraction (touristAttractionId A, address B, name C, totalPrice D, numberOfTickets E, bookingId F)

$A \rightarrow BC, AF \rightarrow E, AE \rightarrow D$ Candidate Key: AF

R1: ADEF

R2: ABC

(8). Hotel (hotelId A, hotelName B, hotelAddress C, roomType D, checkinDate E, numberOfRooms F, checkoutDate G, totalPrice H, bookingId I)

$A \rightarrow BC, AI \rightarrow DEFG, ADF \rightarrow H$ Candidate Key: AI

R1: ABC

R2: ADEFGHI

(9). Transportation (transportId A, deptDate B, deptTime C, arrvDate D, arrvTime E, deptLocation F, arrvLocation G, numberOfTickets H, totalPrice I, bookingId J)

$A \rightarrow BCDEFG, AJ \rightarrow H, AH \rightarrow I$ Candidate Key: AJ

R1: ABCDEFG

R2: AHIJ

3.6. SQL Introduction

As for the relational database SQL, this paper uses PostgreSQL 15 as DBMS. PostgreSQL is an advanced open-source relational database similar to MySQL, Oracle, etc. There are 11 entities derived in the users' instruction, 12 relations derived in the relational model and 16 tables derived in normalization. The 16 tables are as listed.

- 01 – 03: Customer, Employee, Consults;
- 02 04: Insurance;
- 05 -- 06: Car, CarRental;
- 06 07: Booking;
- 08 -- 09: TouristAttraction, TouristAttractionBooking;
- 10 -- 11: Hotel, HotelBooking;
- 12 -- 16: Transportation, TransportationBooking, Train, Flight, B

Table Creation

This is the code for table creation.

```
CREATE TABLE Customer (  
    customerId VARCHAR(10) PRIMARY KEY,  
    firstName VARCHAR(20) NOT NULL,  
    lastName VARCHAR(20) NOT NULL,  
    email VARCHAR(30),  
    bankAccount VARCHAR(20),  
    phoneNo VARCHAR(15) NOT NULL,  
    passportNo VARCHAR(20),  
    driverLicense VARCHAR(20)  
);
```

Views of Data Queries

Customer View WHERE customerId = 'C0002'

For example, the project can use SQL to list required data.

- List the personal details of a certain customer.
Statement: SELECT * FROM Customer WHERE customerId = 'C0002';
- List all bookings and the details of a customer.
Statement: SELECT bookingId, price, startDate, endDate
FROM Booking WHERE customerId = 'C0001';
- List the details of a given type of booking (hotel/tourist attraction/transportation) of a customer.
Statement: SELECT *

```
FROM TransportationBooking
WHERE bookingId IN (
  SELECT bookingId
  FROM Booking
  WHERE customerId = 'C0001'
);
```

- List the time, place and model of car rented by the customer.
Statement:

```
SELECT cr.*, c.model
FROM CarRental cr, Car c
WHERE customerId = 'C0001' AND cr.plate = c.plate;
```
- List the address, room type, check-in/out date and total price of hotel booked by the customer in the future.
Statement:

```
SELECT h.*, hb.roomType, checkinDate, checkoutDate, numberOfRooms,
totalPrice
FROM HotelBooking hb, Hotel h
WHERE bookingId IN (
  SELECT bookingId
  FROM Booking
  WHERE customerId = 'C0001'
) AND hb.hotelId = h.hotelId
AND hb.checkinDate > '2023-08-03';
```
- List the detailed transportation that the customer has not yet taken and order them by departure date and time.
Statement:

```
SELECT DISTINCT tp.*, tb.numberOfTickets
FROM TransportationBooking tb JOIN Transportation tp
ON tb.transportId = tp.transportId
WHERE tb.bookingId IN (
  SELECT bookingId
  FROM Booking
  WHERE customerId = 'C0002'
) ORDER BY tp.deptDate, tp.deptTime;
```
- List the insurance purchased by the customer.
Statement:

```
SELECT * FROM Insurance WHERE customerId = 'C0001';
```
- List travel agency employees who have worked for you.
Statement:

```
SELECT emp.* FROM Employee emp
JOIN Consults con ON emp.employeeId = con.employeeId
```



```
JOIN Customer cust ON con.customerId = cust.customerId  
WHERE cust.customerId = 'C0001';
```

4. Conclusion

Due to the rapid progress of tourism and tourists' higher requirements for service and interaction, travel agency information management system. It is imperative to improve and perfect. Based on the above background, this paper uses relational database and NoSQL database technology to design a travel agency customer management system, and designs and implements the storage, reading and writing, transaction processing of travel agency customer data, especially the storage and processing of unstructured data such as pictures and video information. The integrated management of the two types of data ensures the correlation and consistency of the relevant data. Among them, customer registration, various business reservations, changes and inquiries, and matching with employees are solved through relational data; users share the system's text, pictures and videos through NoSQL processing; the two keys are associated by setting object ID and foreign keys. To sum up, the customer management system of travel agencies based on relational database and NoSQL database basically meets the current management needs of travel agencies. At the same time, in the design process, although the processing of unstructured data has a better way to solve, the impact of data size on the efficiency of the system is insufficient research and evaluation; there is also room for further improvement in the connection between relational and NoSQL databases. Therefore, it is necessary to further improve and optimize the system in the subsequent design and application process.

References

- [1] Circular of the State Council on Printing and Issuing the Plan for the Development of Tourism Industry during the 14th Five-year Plan Period. Gazette of the State Council of the People's Republic of China, 2022(5):28-46.
- [2] Mouden Z A E , Jakimi A , Hajar M ,et al.Graph Schema Storage in SQL Object-Relational Database and NoSQL Document-Oriented Database: A Comparative Study. Learning and Analytics in Intelligent System 2019,1-11.
- [3] Jose B , Abraham S .Performance analysis of NoSQL and relational databases with MongoDB and MySQL.Materials Today: Proceedings, 2020, 24(7):2036-2043.
- [4] LI Li-meng. Application scenarios of relational databases and NoSQL databases. Electronic Technology & Software Engineering,2022(16):184-187.
- [5] MIN Zhao-hao, Yang Zhuo-fan. NoSQL database versus relational database. Electronic Technology & Software Engineering,2021(14):199-201.
- [6] Hua-Qiang Z .Relational Database and NoSQL Database. Computer Knowledge and Technology, 2011.
- [7] LI Ji-wei, Duan Zhong-shuai, WANG Shun-ye. The Storage Technology in NoSQL of MongoDB. Computer Knowledge and Technology,2018,14(27):7-9.
- [8] Bouamrane K , Matallah H , Belalem G .Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database. International journal of software science and computational intelligence, 2021(3):13.
- [9] REN Ming-fei, LI Xue-jun, CUI Meng-meng and etc. . Design and development of non-relational database based on MongoDB. Computer Knowledge and Technology,2019,15(34):1-2.