Design and Implementation of Part-time Teacher Management System Based on Cloud Platform

Shu-cheng LIU* and Fang-qin XU
College of Information Technology, Shanghai Jian Qiao University, Shanghai, China
*Corresponding author

Keywords: Cloud platform, Teacher management system, Database, MySQL

Abstract. Aiming at the problems of low efficiency, inconvenient use and high cost caused by the old design of traditional university teacher management system, a part-time teacher management system based on cloud platform is designed. Reduce operational and management costs through the cloud platform, make the system easy to expand through reasonable database design, and easily achieve high cohesion and low coupling during system development. By adding two fields, effective time and expiration time, the data is controlled by the traceability while controlling the growth rate of the data volume. Through the use of JSON data format in MySQL, the system can easily customize the salary formula.

Introduction
In the information age, colleges and universities have their own teacher management system, usually built in the campus network, but because the original system construction time is early, the design concept is old, which makes it difficult to maintain and expand.¹ With the increasing volume of data and the emergence of new part-time teacher roles, the following types of problems have emerged:

1) Data redundancy is large and data is inconsistent;
2) The system runs slowly and has low efficiency;
3) The interface is not friendly and the access speed is slow;
4) The function is difficult to expand.

The emergence of cloud computing technology provides an opportunity for the solution of these problems. Therefore, it is of great significance to study the university teacher information management system in the cloud computing environment.

Technology Development
Early information management used manual statistics by managers and saved using Excel files.² It was then upgraded to an information management system using the C/S architecture ("client/server" mode). Then upgrade to B/S architecture (WEB browser/server architecture).

The emergence of cloud computing provides a new choice for the deployment of information management systems. The architecture of cloud computing is divided into three types according to the level of resources that service providers provide cloud services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The cloud platform here refers to the IaaS layer.³ It is the cornerstone of cloud computing development, providing server hosting, public storage space and other IDC-related public rental service platforms. The core technology of IaaS is virtualization, which mainly includes computing virtualization, storage virtualization and network virtualization. By providing customers with processing, storage, networking, and other infrastructure, users can remotely deploy and run any software, including operating systems and applications. Users do not manage or control the underlying cloud infrastructure, but can control the operating system, storage, publishing applications, and network components that may have limited control choices.
The cloud infrastructure and platform are generally more powerful and efficient than the university's own IT systems. A large amount of data processing is carried out in the “cloud”, which can greatly improve the processing speed. At the same time, the system built in the cloud is convenient for external access and easy to upgrade resources.

Architecture

Since the traditional university information management system is usually built directly in the campus network, it is difficult to log in to the system when the user is not in the school, which is very inconvenient. Non-standard, inflexible, and old system design concepts, the salary calculation method fixed in the program makes the original system difficult to be compatible when a new role such as a part-time teacher appears, so that it can only be manually filtered by the manager. Work, comparison, error correction, etc., recorded in the excel form, and then imported into the system, this situation led to an increase in operating and maintenance costs. In order to meet the traceability, the past processing method often marks the version number, and each stage is updated once, which makes the data volume increase rapidly, resulting in a slow system running speed.

In order to solve these problems, a part-time teacher management system based on cloud platform and more reasonable database design is proposed.

Figure 1. Architecture Diagram.

There are several benefits to having both the server and the database in the cloud:

1) Reduce operating costs.
2) Easy to maintain and upgrade.
3) Easy to access and use by users.

Cloud platform providers adopt an “on-demand” mechanism for resources, reducing allocated resources during low peak periods, increasing resource supply during peak periods, and allowing users to pay for resource usage. This makes the cloud-based system not waste in the low-end period of the traditional B/S architecture system. In the case of insufficient performance during peak hours, the billing is more reasonable and can reduce expenses. At the same time, it can be upgraded and upgraded efficiently, which is easy for users to use, access, and reduce the management burden.
Functional Design

![Function Diagram](Image)

**Figure 2. Function Diagram.**

Database Design

**Entity Relationship diagram**

![Entity Relationship Diagram](Image)

**Figure 3. Entity Relationship Diagram.**

In order to meet traceability, all entities except the administrator, user, and course tasks have a modified record relationship with the semester. The relationship has two attributes (effective semester, invalid semester). Attributes can effectively achieve traceability by that: When a new entity
is created, the effective semester is set to the current semester, and the invalid semester is empty. When the entity is modified, the original entity is not modified, but the invalid semester is set to the current semester, and the modified attributes and other unmodified others are used. The attribute creates a new entity, and finally updates the field of the entity with the entity number as the foreign key in the current semester to the new entity number.

At the time of enquiry, the data of a certain semester can be obtained by judging the conditions: \((\text{effective semester} < \text{searched semester}) \&\& ((\text{searched semester} < \text{invalid semester}) \mid \text{invalid semester} \text{is empty}))\).

Since administrators and users only provide management and browsing functions, and each course task is valid only for a certain semester, it will not be modified, so these three do not need to be traced.

**Relationship Mode**

According to the complete E-R diagram, the following 12 relationship patterns can be derived:

- Administrator (account, password);
- User (account, password);
- Semester (number, name);
- Department (number, name, effective semester, invalid semester);
- Preparation (number, name, effective semester, invalid semester);
- Title (number, name, unit price, effective semester, invalid semester);
- Class time interval (number, basic class hours, class time limit, class time limit, effective semester, invalid term);
- Teacher (number, job number, ID card, name, employment status, user account number, department number, preparation number, job title number, class time interval number, effective semester, invalid semester);
- Course type (number, name, formula coefficient, effective semester, invalid semester);
- Course (code, name, class hours, number of weeks, course type number, effective semester, invalid term);
- Course tasks (number, task code, semester number, course number);
- Teacher and class (task number, teacher number, teacher corresponding number, teacher corresponding class, extra class hours, extra weeks).

**Optimization**

Since the formula coefficient attribute in the course type relationship is intended to be stored in the JSON type field in MySQL, the above relational mode satisfies the first paradigm and due to the teacher's class relationship \((\text{task number}, \text{teacher number}) \rightarrow (\text{actual number, teacher class, extra class hours, extra weeks})\), all non-primary attributes of the remaining relational models are completely dependent on the primary key, so the above relationship mode Belongs to the second paradigm.

However, in the course tasks, there are \((\text{number}) \rightarrow (\text{task code, semester number}), (\text{task code, semester number}) \rightarrow (\text{course number})\), so the third paradigm is not satisfied. It is now broken down into:

- Course tasks (number, task code, semester number)
- Task course (task number, course number)

Thereby eliminating the transfer dependence and reaching the third paradigm. The relational pattern satisfies: all non-primary attributes are completely functionally dependent on each candidate key; all primary attributes are completely functional dependencies for each candidate key that does not contain it, and there is no attribute that complete functions depend on a set of attributes of non-candidate keys; Therefore, BCNF is satisfied.

**Implementation**

Complete the design of all data tables through MySQL Workbench and export the following complete database relationship diagram.
After a period of trial operation, the results show that the system database is semantically complete and has low redundancy, which is not easy to cause bugs caused by database design. Because it is built on the cloud platform, users can easily access it anytime and anywhere, and the system performance is good.

**Conclusion**

Through the investigation and analysis of the development status of college teachers' information management system, the actual colleges and universities' demand for part-time teacher management puts forward a series of problems that are currently plagued by colleges and universities because the
teacher information management system is too old. Therefore, a part-time teacher management system based on the cloud platform is proposed and designed, which provides inspiration for the development of such systems and has practical significance.

Acknowledgement

This thesis is one of the achievements of the construction of the innovation base for the employment of college graduates funded by the Shanghai Municipal Education Commission, the extra-curricular practice base project of the Internet of Things Engineering. I would like to express my sincere gratitude. I also thank Shanghai Jianqiao College and colleagues who provide help and support at work.

References


[2] Li L. Design and implementation of educational information management system in a college[D]. School of Information and Communication Engineering, 2018: 2-36.


