

# Database Design on English Examination Comprehensive Information System

Rui Xiang

Teaching and Research Institute of Foreign Languages,  
Bohai University, Jinzhou, 121013, China

**Abstract**—With the deepening of reform and opening up and the development of foreign exchanges, English education is becoming more and more important, the comprehensive information management of English examination is an important way to enhance the information level of English education. This paper is based on SQL Server database management system to begin the database design, which lays the foundation for the development of software. Firstly, it designs the logical structure which consisted of 6 basic tables related to the information about examination, and expresses the format by means of "the field name, data type, and the occupied storage space "; Secondly, design the B tree index table based on the principle of B tree index; Finally, set up table structure in the DBMS and generate a database diagram. The contents of the research in this paper, can realize several features such as centralized management of information, comprehensive content, compete information and high security.

**Keywords**- English; examination; information management; database design.

## I INTRODUCTION

Examinational information management is a complicated work in colleges and universities which need to collect and analyze according to the examinees' registration information, to realize the function of recording data, outputting report and statistical analysis of performance according to the requirements of managers [1]. At present, most colleges and universities are completing tabulation, registration of examinee information, arrangement of examination room, examination number through the manual processing, and it is difficult to comprehensively evaluate the conditions of examinees because the amount of data to be processed is large. With the enrollment expansion of colleges and universities, the examinee group become bigger, and examinational information management became more complex [2]. With the rapid development of computer technology, information development has made a great progress in teaching and management, requires a balanced development of all aspects, from two aspects of software and hardware to construct into the first-class educational teaching platform. English examination system is one of the important aspects. The English examination comprehensive information system in line with the target that reducing teachers' workload, improving work efficiency and optimizing the students' test flow to develop. Database design is an important work in software development, and this article is based on the systematic investigation and demand analysis to begin the design of logical structure, index and the diagram, which lays the foundation for the development of software system.

## II LOGICAL STRUCTURE DESIGN

The database on English examination comprehensive information system consists of 6 tables, and they are respectively: "English examination information main table", "English course information table", "English examination level information table", "English examination organizational unit information table", "English examination monitor teacher information table" and "English examination monitor teacher name list information table".

Database design on logical structure usually results in the form of table, but can also be used in other forms. In the table form that seems clear and intuitive, but design space of document is occupied larger. This paper takes the field list mode due to limited space. Take "English course name" field as an example, using the format that "(CourseName, Varchar, 50)" to express. Among them, "CourseName" is the field name (to distinguish the two English words through capitalized), and "Varchar" is the field type, "50" is the occupation of memory bytes. Logical structure design results in each table are as follows:

(1) "English examination information main table" structure: EnglishExaminationMain [(PrimaryKey, Decimal, 10,0), (ExaminationID, Varchar, 10), (ExaminationPaperID, Varchar, 10), (OrganizationalUnitCode, Varchar, 10), (ExaminationDate, Datetime), (ExaminationStartTime, Datetime), (ExaminationEndTime, Datetime), (ExaminationPlace, Varchar, 50), (ExaminationEnvironment, Varchar, 50), (EnglishCourseCode, Varchar, 10), (ExaminationLevelCode, Varchar, 5), (TestPaperSafekeepingPlace, Varchar, 50), (TestPaperSafekeepingName, Varchar, 50), (ScoreCommitTime, Datetime), (ScoreCommitTeacher, Varchar, 50), (Remarks, Text)]

(2) "English course information table" structure: EnglishCourse [(PrimaryKey, Decimal, 10,0), (NumericCode, Smallint), (PinyinCode, Varchar, 4), (CourseName, Varchar, 50), (CourseCode, Varchar, 10)]

(3) "English examination level information table" structure: ExaminationLevel [(PrimaryKey, Decimal, 10,0), (NumericCode, Smallint), (PinyinCode, Varchar, 4), (ExaminationLevel, Varchar, 20), (ExaminationLevelCode, Varchar, 5)]

(4) "English examination organizational unit information table" structure: OrganizationalUnit [(PrimaryKey, Decimal, 10,0), (NumericCode, Smallint), (PinyinCode, Varchar, 4), (OrganizationalUnit, Varchar, 100), (OrganizationalUnitCode, Varchar, 10)]

(5) "English examination monitor teacher information table" structure: MonitorTeacher [(PrimaryKey, Decimal, 10,0), (NumericCode, Smallint), (PinyinCode, Varchar, 4), (MonitorTeacher, Varchar, 30), (MonitorTeacherCode, Varchar, 10), (WorkingCollege, Varchar, 50), (ProfessionalTitles, Varchar, 30), (TeacherHomeAddress, Varchar, 100), (TeacherMoblePhone, Varchar, 50), (TeacherHomePhone, Varchar, 50), (TeacherEmail, Varchar, 100)]

(6) "English examination monitor teacher name list information table" structure: MonitorTeacherNameList [(PrimaryKey, Decimal, 10,0), (ExaminationID, Varchar, 10), (MonitorTeacherCode, Varchar, 10), (MonitorPlace, Varchar, 50), (ExaminationRoomNo, Varchar, 10)]

The associations among tables are as follows: "English examination information main table" and "English course information table" establish one-to-many connection by "CourseCode" field; "English examination information main table" and "English examination level information table" establish one-to-many connection by "ExaminationLevelCode" field; "English examination information main table" and "English examination organizational unit information table" establish one-to-many connection by "OrganizationalUnitCode" field; "English examination information main table" and "English examination monitor teacher name list information table" establish one-to-many connection by "ExaminationID" field; "English examination monitor teacher name list information table" and "English examination monitor teacher information table" establish one-to-many connection by "MonitorTeacherCode" field.

### III INDEX DESIGN

In a relational database, an index is a database structure related to table which can make the SQL statement corresponding to table to execute faster [3]. When there are a large number of records in a table, the first way to search information is table search to query a table, and then take out all records to compared with every condition of the query, and then return the record meet the conditions, if so it will consume a large amount of database system, and cause a large number of disk I/O operations; The second is setting up the index in the table, then find the index values line with the query condition in the index, and finally quickly find the corresponding record in the table through the ROWID stored in the index.

The index has a variety of classification methods, according to the number of columns can be divided into a single index and composite index, according to whether the column is unique can be divided into unique index and non unique index, according to the organization can be divided into the form of B tree index ,hash index and bitmap index. B tree is a multi-level, self maintenance structure. A B tree comprising a top layer, called the Root node; from 0 to a plurality of intermediate layer, called the Branch node; A bottom, which comprises a plurality of leaf nodes in the bottom [4, 5].

The created solutions of B tree index of this system are as follows:

(1) English examination information main table: create unique index according to the "ExaminationID" field,

Create a single index respectively according to the "ExaminationPaperID, ExaminationDate, EnglishCourseCode, ExaminationLevelCode" field,

Create a composite index according to the "ExaminationDate, ExaminationStartTime" field;

(2) English course information table: create unique index respectively according to "NumericCode, CourseCode" field, and create a single index according to the "CourseName" field;

(3) English examination level information table: create unique index respectively according to "NumericCode, ExaminationLevelCode" field, and create a single index according to the "ExaminationLevel" field;

(4) English examination organizational unit information table: create unique index respectively according to "NumericCode, OrganizationalUnitCode" field, and create a single index according to the "OrganizationalUnit" field;

(5) English examination monitor teacher information table: create unique index respectively according to "NumericCode, MonitorTeacherCode" field, and create a composite index according to the "WorkingCollege, ProfessionalTitles, MonitorTeacher" field;

(6) English examination monitor teacher name list information table: create a composite index according to "ExaminationID, MonitorTeacherCode, MonitorPlace, ExaminationRoomNo" field.

### IV DATABASE DIAGRAM

The database diagram used to describe the relationship between the database table (view), and relationship type is divided into "one-to-one", "one-to-many" and "many-to-many". In the SQL Server database management system, you can use the diagram to create relationships with management tables. The diagram is a part of the database which stored in the database. When create a diagram, users need to add tables and relationships, they can add to table existed, or create new tables. Database diagram designer can create, edit or delete "table, column, key, index, relationship and constraint" etc. In order to make the database visualization, users can create one or more diagram, to show some or all of the tables, columns, and the relationship in the database [6]. For any database, can create any number of database diagram; each database table can appear in any number of diagram. The diagram of English examination comprehensive information system which consists of five tables is as shown in Fig. 1.

In the database diagram, each table has three distinct features: (1) the title bar, title bar displays the name of the table. If you have modified a table, but not to save the table, it appears an asterisk (\*) at the end of the table name, to express the unsaved changes; (2) the row selector, used to select a column in a database table, when the columns in the primary key of the table, the row selector displays a key symbol; (3) attribute column, only visible in certain view table, to manage the size and layout of the diagram.

In the database diagram, every relationship appears with three different functions [7, 8]: (1) the end point. To express

that relationship is one-to-one or one-to-many. If a relationship has a key in an end point, there are infinite symbols in another end point, and it is a one-to-many relationship. If a relationship in each termination points has a key, it is one-to-one relationship; (2) linear. It means that when add a new data to the foreign key table, whether DBMS enforce referential integrity of the relationship. If it is the solid line, DBMS will enforce referential integrity for

the relationship when added or modified the row in the foreign key table. If it is the point line, the DBMS does not enforce referential integrity for the relationship when rows are added or modified in the foreign key table; (3) the related table. For a one-to-many relationship, the foreign key table is the table closed to the infinity symbol of the line. If the two end points of the line connected to the same table, the relationship is a reflexive relationship.

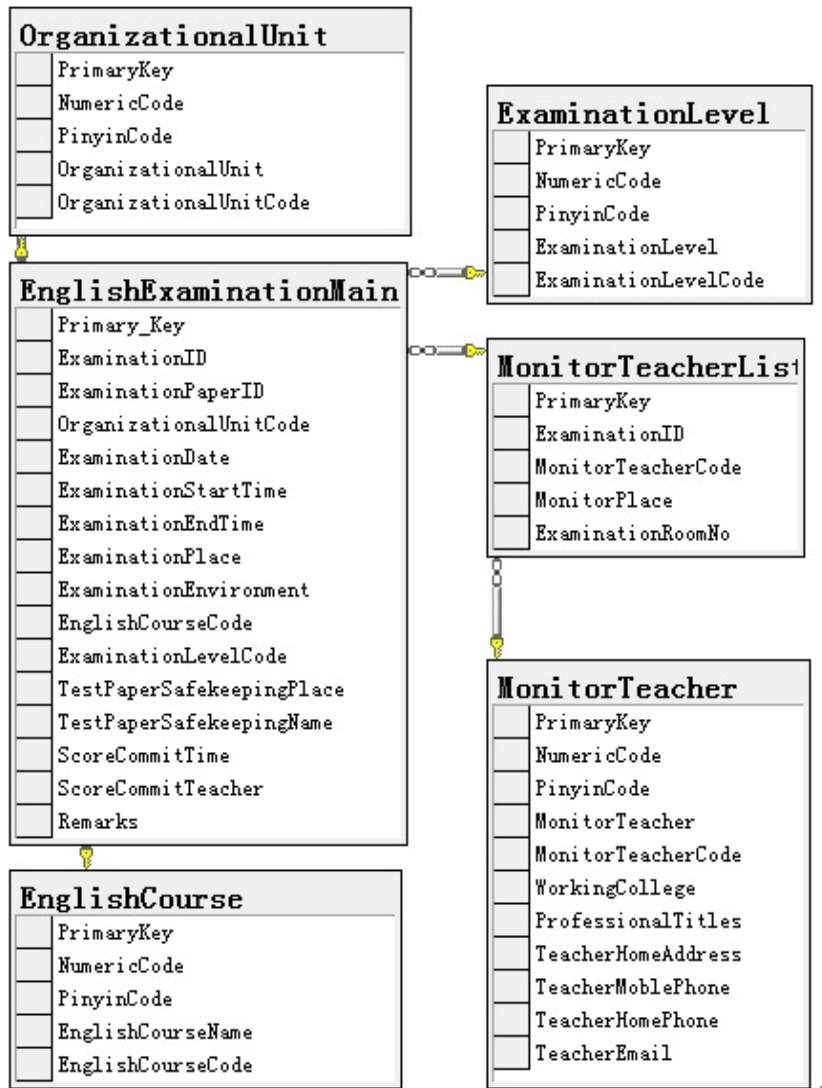


Figure 1. Database relationship diagram on English examination comprehensive information system

## V CONCLUSIONS

Campus network construction is gradually perfect, and the traditional examination way already could not adapt the modern examination needs. The development of information technology provides a broad development stage for the standardized test, English test in the network environment has the advantages that the traditional examination mode unparalleled. Application of network English examination system will reduce the workload of

teachers, completely change the human factors existed in the examination, improve examination quality, provide a fair and impartial examination environment for students. Database design in this paper, which can realize the centralized management of information, has many features such as comprehensive content, information integrity and high security.

## VI ACKNOWLEDGEMENTS

This work is supported by 2013 Teaching Reform Project of Bohai University (Class A): A Study of Improving Students' English Competence Through Class Teaching (JG13WT004).

## VII REFERENCES

- [1] D. K. Wang, Z. H. Li, Y. F. Zhao, Z. B. Luo. A Design of Universities' Tests Management Information System. *Journal of Beijing University of Technology*, 26(1), pp. 123-126, 2000.
- [2] F. Wang. Design and Implementation of the Web-based Information Management for College English Grage Test. *Journal of Hunan Institute of Humanities, Science and Technology*, 25(4), pp. 35-37, 2008.
- [3] Baidu encyclopedia . Database index. <http://baike.baidu.com/view/2079871.htm?fr=aladdin>, 2014-9-20.
- [4] Q. T. Geng, J. Di, L. Chang. Index Storage Based on B+Tree. *Journal of Jilin University(Science Edition)*, 51(6), pp. 1133-1136, 2013.
- [5] T. B. Hu, J. Zhong. Database index optimization algorithm based on cluster B+tree. *Journal of Computer Applications*, 33(9), pp. 2474-2476, 2013.
- [6] D. S. Zhang. Analyze on the Design of the Database and Website System. *Computer Knowledge and Technology*, 8(35), pp. 8356-8360, 2012.
- [7] Microsoft Developer Network. Design database relational diagram. <http://msdn.microsoft.com/zh-cn/library/ms171971>, 2014-9-20.
- [8] J. M. Liu. Student Physical Fitness Test System Database Design. *Computer Knowledge and Technology*, 10(10), pp. 4636-4639, 2014.