Database Programming under Labwindows/CVI Platform

Zhenjie Zeng, Xiaofei Zhu*, Shiju Qi, Xiaowei Shen and Shangjun Cai
Rocket Force University of Engineering, Xi’an, China

*Corresponding author e-mail: 1224048915@qq.com

Abstract. LabWindows/CVI is one of the most commonly used automatic test system software development platforms. When developing automated test systems, a large amount of test data is often generated. How to implement effective test of a large amount of test data is a problem that is often encountered. In view of this, first introduced the relevant theory and background knowledge of database technology; then discussed how to use ODBC driver and SQL ToolKit software toolkit to operate Access database under CVI platform; Finally, a small program was taken as an example to introduce The specific code of the basic operation of the database is written under the CVI platform. The article provides a reference for the programming of the database under the CVI platform in the automatic test system.

1. Introduction

In the automatic test system, a large amount of test data is often generated. These test data need to be used when the system performs data analysis, and it is also very likely that the data will be viewed and analyzed in the future. Therefore, the data must be stored properly. Before the advent of computer database technology, the way to save data was often recorded manually and then archived. This way, the efficiency of data preservation was extremely low, and it was also troublesome in subsequent inspections. It was difficult to organize and analyze the data.

LabWindows/CVI is a software development platform that has been active in the field of test and measurement. It has undergone considerable development and has been designed and optimized for measurement and automated application development. It uses object-oriented programming ideas, rich controls and buttons, and a powerful library of functions and add-on toolkits that make it powerful and ideal for automated test system software development. Under the platform of CVI, how to implement effective management of a large amount of test data has become a problem faced by system developers. In view of this, this article takes the programming of Access database under the CVI platform as an example to introduce the programming operation of the database in the automatic test system, and provides a solution for the data storage and processing in the automatic test system.

2. Database technology

Database technology is a comprehensive technology involving basic knowledge of operating systems, data structures, and program design. It is a specialized discipline in computer science and is widely used in measurement and control. The development of the database can be roughly divided into three generations, the first generation level, the networked database system, the second generation relational database system, and the third generation database system featuring the object oriented model. The current design of the database system is mainly based on the second On behalf of and with three
generations. The mathematical foundation of the current mainstream database application is the relational model, which is actually a set of two-dimensional tables used to represent the relationship between entities and entities.

2.1. Relationship model
The relational model is the most widely used mathematical model of the database system [1]. The relational model is based on the table structure. The data structure is the table (also called the relation). The data operations are table-oriented operations, and the data constraint is the limitation. The data in the table. Columns in the data table are called attributes (also called Columns) and rows are called tuples. Each row has only one element in each column. The relational model requires that each tuple in a relation is unique and that identical tuples are not allowed.

2.2. SQL language
SQL (Structured Query Language) is a standard language for describing and operating relational databases [2]. It became an international standard in 1987 and is a database query and programming language for accessing data, querying, updating, and managing relationships. Database systems. SQL is a non-procedural high-level programming language that allows work on high-level data structures. It does not require the specification of data storage methods, nor does it need to specify data storage methods. Therefore, even a different database system with a completely different underlying structure can be used. The same SQL is used as the interface for data management, but various database systems have modified and expanded the SQL specification during the course of practice. Therefore, the SQL between different database systems cannot actually be mutually universal.

2.3. Database system composition
A complete database system is shown in Fig 1 The database in the system is located in the bottom layer and consists of data. The database management system is a system software specifically used for database management. It can perform various operations on the database through the operating system. On top of the database management system, developers can use development tools to develop a wide variety of applications to target different users, and ultimately achieve user-database interaction.

![Figure 1. The composition of the database system](image)

Among them, DB (Database) is a collection of structured and sharable related data that is stored in the computer structure for a long time. The data stored in the database is organized, described and stored according to a certain mathematical model, and has a smaller Data redundancy and high data independence, ease of expansion, and shareability.
DBMS (Database Management System) is a system software that is located on the computer operating system and manages the database. All operations on the database must be completed by the database management system. The database management system provides four functions: data definition language, data manipulation language, database operation management, database creation and maintenance. At present, many of the more popular database management systems, such as SQL Server, Oracle, Access, FoxPro, etc., have different characteristics and are applied to different occasions.

3. Operation on the database
To develop and use a database in the Labwindows/CVI environment, the SQL ToolKit toolkit must be installed. After the installation is successful, the toolkit will automatically register an ODBC data source on the computer.

3.1. ODBC
ODBC (Open Database Connect) is a powerful tool for database application development. It is a set of interface specifications for database applications developed by Microsoft Corporation. The library developed based on this specification provides a unified programming interface API for database applications. When developing an application program based on the ODBC interface specification, an application only needs to call ODBC functions and use standard SQL statements. It does not need to involve the underlying operation of the database. These complex low-level operations are performed by the drivers of each database, which makes the database application. The difficulty of development is greatly reduced, and because the drivers of each database comply with the same standard, the upper application has a good portability [3].

ODBC has the following characteristics:
(1) Make the user program very high interoperability, the same target code applies to different DBMS;
(2) Is a programming interface using SQL;
(3) Facilitate program integration and provide technical support for client/server architecture;
(4) Make application developers avoid the complexity of connecting with data sources.

3.2. SQL ToolKit
Labwindows/CVI is a software development tool for the field of measurement and control [4]. It does not have the function of database development. When using CVI to develop a database, it needs the help of the database development tool kit SQL ToolKit. It is not distributed with the CVI installation package, but it can be downloaded and installed for free on the National Instruments official website.
It must be installed in the CVI installation directory before it can be used normally. The toolkit contains many functions for CVI to access the database. It contains powerful database access, operation advanced functions, and can be flexibly used to access and operate the database. The SQL ToolKit library function also conforms to the ODBC standard, so it supports all ODBC-compliant database systems. Almost all current mainstream databases such as Oracle, SQL Server, and Access can be accessed through it [5].

3.3. Access
Microsoft Access is an excellent desktop database management system. It provides a powerful tool for establishing and using efficient and fast databases [6]. It belongs to a typical relational database system. The relational model is used as the data organization method. Both the entity and the connection between the entity and the entity are represented by the relationship. The retrieval result of the data is also the relationship. Therefore, the data structure is simple and clear. Easy to understand and use. Microsoft Access is suitable for stand-alone operation, allowing operators to quickly and efficiently complete data operations within the system. For automated test systems without network requirements, it is best to use Access to store and manage various information and data in the test system. Select.

3.4. Database operations
After the successful installation of the SQL ToolKit toolkit, the sql_db driver will be expanded into a standard CVI library of functions. There are a total of 11 subclasses of function libraries for the user to invoke [7] to implement database programming under the CVI platform. If you do not install SQL ToolKit in the CVI installation directory, you can also load the cvi_db.lib library file into the software platform through the "Load Instrument" function in the CVI when you need to perform operations on the database. You can also call the library function. The main relevant library functions are shown in Fig 4.

![Diagram of database operations](image)

**Figure 3. Call library**

Use it to perform operations on the database in five steps. The main steps are shown in Fig 3
Before connecting the database resources, first add the name of the database to be operated on to the data source in the management tool, and specify its storage location. The database can be manipulated [8].
The first step is to connect to an ODBC data source. Before performing any operations on the database, it must be determined that a reliable data source connection has been established. SQL ToolKit supports multiple concurrent requests to connect to one or more databases.

The second step to activate the SQL retrieval statement, SQL ToolKit provides three ways to activate the index, respectively, Automatic SQL, Explicit SQL and Advanced functions, Automatic SQL can only perform more simple SELECT and CREATE TABLE SQL command statement, Explicit SQL can SQL statements are passed to functions, so more complex statements can be executed, and Advanced functions allow users to perform more advanced configurations and operations.

The third step is to process the SQL statement. Normally, only the SELECT statement needs to be further processed.

The fourth step is to disconnect the SQL connection. After all SQL statements are executed, the occupied resources should be released.

The fifth step is to disconnect the data source, disconnect the data source, and release system resources.

Figure 4. Database operation flow

4. Program instance
First of all, you need to configure an ODBC data source on your computer. The specific operation steps are Control Panel -> Administrative Tools -> ODBC Data Source (32-bit or 64-bit). After you open it, you can view the configured data source and configure the new one. Data source. After the data source is configured, you can connect to the data source for database operations in the CVI through the function DBConnect ("DSN=Data Source Name").

The following describes the database programming in the CVI by adding, deleting, modifying, and searching the data table std (containing two Columns of "NAME" and "AGE"). Codes other than the operation of the database in the middle have been omitted.

//Initialize the database
DBInit (DB_INIT_SINGLETHREADED);
//Create a database connection and get a connection handle
Hdbc = DBConnect ("DSN=std");
//Take the record positioning to the first
First (panelHandle, PANEL_CMD_FIRST, EVENT_COMMIT, NULL, 0, 0);

/*Add records*/
Hstmt1=DBPrepareSQL (hdbc,"insert into std values (?,?)");
//bind the variable to the field
DBCreateParamChar (hstmt1,"NAME", DB_PARAM_INPUT, namevalue, 9);
DBCreateParamInt (hstmt1,"AGE", DB_PARAM_INPUT, agevalue);
//Execute SQL commands (insert into std values (namevalue, agevalue))
DBExecutePreparedSQL (hstmt1);
//End this operation, release the variable
DBClosePreparedSQL (hstmt1);
/*Delete Record*/
//Database operation with parameters (records when NAME=value was deleted from data table std)
Hstmt1=DBPrepareSQL (hdbc,"delete * from std where NAME=?");
//bind the variable to the field
DBCreateParamInt (hstmt,"NAME",DB_PARAM_INPUT, namevalue);
//Execute the SQL command (delete * from std where NAME= namevalue) operation
DBExecutePreparedSQL (hstmt1);
//End this operation, release the variable
DBClosePreparedSQL (hstmt1);
/*Change records*/
//Create a map
Hmap=DBBeginMap (hdbc);
//Map Columns to variables
DBMapColumnToChar (hmap,"NAME", 9, namevalue, &namestatus,"");  
DBMapColumnToInt (hmap,"AGE", &agevalue, &agestatus);
//Activate the mapping
Hstmt=DBActivateMap (hmap, "std");
//Retrieve records (rows) from the first one until the target record position stops
While (DBFetchNext (hstmt)! =DB_EOF) {......}
//Modify record (line)
DBPutRecord (hstmt);
//Deactivate mapping, end this operation, release variable
DBDeactivateMap (hmap);
/*Find records*/
//Create a map
Hmap=DBBeginMap (hdbc);
//Map Columns to variables
DBMapColumnToChar (hmap,"NAME", 9, namevalue, &namestatus,"");  
DBMapColumnToInt (hmap,"AGE", &agevalue, &agestatus);
//Activate the mapping
Hstmt=DBActivateMap (hmap, "std");
//Retrieve the record (line) from the first one until the target record position is stopped, the record is displayed
While (DBFetchNext (hstmt)! =DB_EOF) {......}
/*Find records*/

5. Conclusion
With the continuous development of science and technology and industrial production technology, the development and application of automatic test systems have become more and more common. The design and operation of the database has become a necessary skill for developers. Therefore, this article combines the excellent development software LabWindows/CVI in the field of measurement
and control with database technology. It introduces the database programming technology under the CVI platform and analyzes it under the CVI platform. The connection of the database, the specific function, the specific steps of the operation and the specific operation of each step, and enumerates the programming examples to explain, provides a reference for the database programming in the automatic test system, and has certain guiding significance.

References
[1] ZHU Xiao-ning. Application of Relational Database Technology in Computer Network Design [J]. Smart City, 2016 (4): 78-79.
[2] LI Gai-li, WANG Hong-jun, ZHAO Jian-hui, et al. Research on data sharing technology under LabWindows/CVI platform [J]. Journal of Beijing Information Science and Technology University, 2007, 22 (3): 20- twenty two.
[3] ZHAO Bo, QUAN Hou-de. Database Programming under LabWindows/CVI Platform [J]. Microcomputer Information, 2008 (06): 171-172+204.
[4] YE Qun-song, XI You-bao, ZHAN Hui-qin. Database Application Development Based on LabWindows/CVI [J]. Computer and Modernization, 2008, 2008 (3): 38-41.
[5] YA0 Xi-jing, ZHAN Hui-qin. Database Design Based on LabWindows/CVI Multi-channel Test System [J]. Electronic Testing, 2011 (11): 43-46.
[6] CAO Jing, ZHI Jian-feng. Realization of Data Sharing Between LabWindows/CVI and Microsoft Access [J]. Industrial Control Computer, 2009, 22 (10): 58-58.
[7] LI Hui. Research on Database Access Technology for LabWindows/CVI [J]. Microprocessors, 2012, 33 (1): 56-59.
[8] LI Hui, XING Guang-yi. Research on Application of SQL Toolkit in LabWindows/CVI [J]. Mechatronics, 2015, 21 (6).