The Analyze of the student’s grades processing based on the SQL Server

Baosheng Qi

Department of Information Engineering and Art Design, Shandong Vocational College of Labor, Jinan, 250022, China

Keyword: Student's grades; Query; Stored procedure; Cursor

Abstract: Using the techniques in the SQL Server, such as query, stored procedure and cursor etc, carried out the student’s result Table to gather to show the problem, its logic carries to complete in the database server, reducing the network data discharge; Moreover, in stored procedure, make use of the template, the system Table to delete in time the temporary Tables, the data that avoided the futility piles up in the database.

1. Problem Presentation

Student achievement processing involves two entities: students and courses. The following entity-connection model is summarized(E-R diagram),As shown in Figure 1.

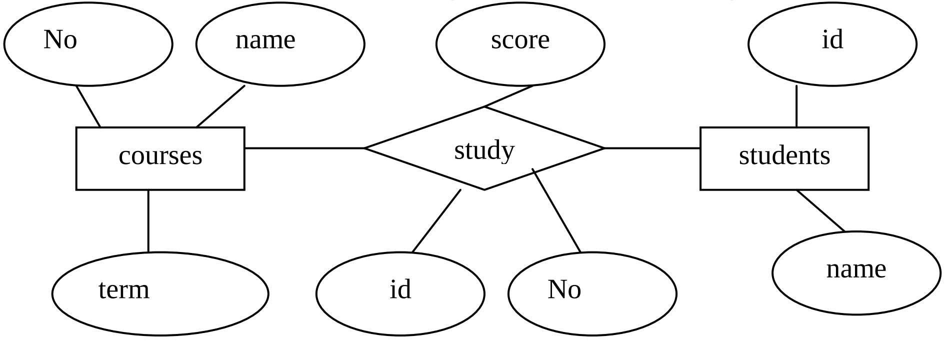

Figure 1 entity-connection model(E-R diagram)

From the above E-R diagram, we can see that there is a many-to-many relationship between students and courses entities. For this reason, we must establish a contact Table with additional score fields, which are the key objects to be investigated and need special treatment measures. According to the above model, course Table, student Table and student-course Table are established. Their structures are as follows.

1.1 The structure of the course is shown in Table 1.

Table 1 Course structure

field name	data type	length	null	primary key
No	char	6	no	yes
name	char	30	no	no
term	char	1	yes	no

Considering the different courses offered by different majors, the meanings of each symbol bit of the course number are designed as shown in Table 2.

Table 2 The Meaning of Symbol Bits of Course Number

1,2bits	3,4bits	5,6bits
Department code	Professional code	Course code

1.2 The structure of student Table is shown in Table 3.

Table 3 Student structure

field name	data type	length	null	primary key
id	char	12	no	yes
name	char	8	no	no

The meanings of the symbol bits of the id are designed as shown in Table 4.

Table 4 The Meaning of Symbol Bits of id

1,2,3,4bits	5,6bits	7,8bits	9,10bits	11,12bits
Enrollment Year	Department code	Professional code	class code	Class ID

1.3 The structure of the student-course Table is shown in Table 5.

Table 5 student-course structure

field name	data type	length	null	primary key
id	char	12	no	yes
No	char	6	no	yes
score	int	4	yes	no

In the SQL Server environment, the author uses stored procedures to deal with student’s grades, and obtains the effect of summarizing and displaying students' semester grades. Because stored procedures can be accessed in VB, VC++, C#, Java, Python and other programming languages, this method has universal significance.

From the final results, it can be seen that the two columns of id and name come from the student Table; the field name data of the achievement column comes from the row data of the course Table; and the achievement data item comes from the student-course Table; this structure solves the problem of unified representation of data with different courses offered in different majors and different courses in different semesters.

2. Knowledge Point Analysis

To solve the above problems, we need to use the basic knowledge of query, cursor and stored procedure in SQL SERVER. The following is a brief analysis of these knowledge points.

2.1 Query

The SELECT statement is used to implement queries in SQL. It retrieves rows from the database and allows one or more rows or columns to be selected from one or more Tables. Examples are as follows:

1) SELECT name FROM sysobjects WHERE name = 'T0' AND type = 'U'

The system Table sysobjects is used to find the name of the user Table whose name is T0 type U to judge whether Table T0 exists or not.

2) select T0.*, T2.score into T3 from T0 left outer join T2 on T0.id=T2.id

Among them, T0 represents the results of a certain class in a semester that has been retrieved, T2 represents the results of a course, the results of the left connection between T0 and T2 by student number are inserted into the new Table T3, and T3 has one more course performance than T0.

2.2 Cursor

A cursor can process a result set row by row in a way similar to a pointer. It can point to any position in the result set and process the rows in a specified position. Using cursor follows the order of declaring cursor, opening cursor, reading cursor, closing cursor and deleting cursor.

(1) Declare cursor

The cursor must be declared before using the cursor. The following statement defines the cursor course_cursor to be associated with the result set of a course offered by a certain profession in a certain semester, and then it can be used to process the result set row by row. declare course_cursor cursor for select No, name from course where(substring(No,1,4)=substring(@ClassID,5,4) and Term=@Term)

(2) Open cursor

After declaring the cursor, you can open the cursor to make it work. Initially, it points to the first record of the result set. The following statement opens the cursor course_cursor defined above and points to the first record of the result set. open course_cursor

(3) Read cursor

Opening the cursor only points the cursor to the result set. To read the data, the FETCH statement should be used. The following statement reads out a row of records of the result set pointed by the cursor course_cursor and saves them in variables @courseID,@courseName, which represent the course number and the course name. fetch course_cursor into @courseID,@courseName

Note that FETCH statements can only fetch one row of records at a time. To read all records, you need to use the WHILE loop.

(4) Close cursor

Close the cursor using the close statement, for example: close course_cursor

(5) Delete cursor

After closing the cursor, the cursor still occupies system resources and needs to be deleted, for example: deallocate course_cursor

2.3 Stored procedure

Stored procedures are database objects that exist outside Tables and consist of a set of SQL statements that can be called by users. Stored procedures can be divided into two categories: system-provided stored procedures and user-defined stored procedures.

System stored procedures are stored procedures provided by the system and can perform various operations as commands. In the example, sp_rename is used to change the name of the score field of Table T3 to the course name represented by @courseName in the form of:

EXEC sp_rename 'T3.[score]', @courseName, 'COLUMN'

User stored procedure is defined in the current database. A stored procedure GetScore is established in an example. The procedure has two parameters @ClassID,@Term, which represent the class number and the term respectively. It is used to return the summary information of the results of a course taken by a class in a semester. The expression of the procedure is as follows:

Create procedure GetScore(@ClassID char(10),@Term char(1)) as   sql_statement

3. Final realization

In the foregoing, the author put forward some questions and analyzed the knowledge points. In the appropriate place, the author also added corresponding annotations, believing that readers can understand it. The program code is as follows:

```
-- Create a stored procedure GetScore, where @ClassID, @Term are input parameters, indicating class number and semester number.
create procedure GetScore(@ClassID char(10),@Term char(1)) as
-- Query the list of students whose class code is @ClassID in the student Table to Table T0
IF EXISTS(SELECT name        FROM      sysobjects
WHERE    name = 'T0'        AND      type = 'U')    DROP TABLE T0
Select    id,name into T0    from Student      where ClassID=@ClassID
declare @courseID char(6)
declare @courseName varchar(30)
-- Declare the cursor course_cursor to point to a course offered in a particular major and semester
declare course_cursor cursor for
select No,name from course
   where( substring(No,1,4)=substring(@ClassID,5,4) and Term=@Term)
```

```
--Open cursor
open course_cursor
-- Read a record pointed by the cursor
fetch course_cursor into @courseID,@courseName
while(@@fetch_status<>-1)   -- Determine whether to read the last record
begin
-- Look up the results of a course in a class in the student-course Table and go to the Table T2
IF EXISTS(SELECT name       FROM     sysobjects
    WHERE   name = 'T2'     AND     type = 'U')   DROP TABLE T2
select id,No,score into T2   from student-course
   where(substring(id,1,10)=@ClassID and No=@courseID)
-- Connect the student list T0 with a course score form T2 to the left, and put the result in the T3 Table
IF EXISTS(SELECT name   FROM     sysobjects
     WHERE   name = 'T3'   AND   type = 'U')   DROP TABLE T3
select T0.*,T2.score   into T3 from T0   left outer join T2   on T0.id=T2.id
select @courseName=ltrim(rtrim(@courseName))
-- Change the name of the score field in the T3 Table to the corresponding course name
EXEC sp_rename 'T3.[score]', @courseName, 'COLUMN'
-- Transfer intermediate result T3 to T0 Table
IF EXISTS(SELECT name   FROM     sysobjects
    WHERE   name = 'T0'   AND     type = 'U')   DROP TABLE T0
select T3.*   into T0   from T3
fetch course_cursor into @courseID,@courseName
end
-- After reading the last record, close the cursor course_cursor and release the system resources it occupies
if(@@fetch_status=-1)
   begin
      close course_cursor
      deallocate course_cursor
   end
-- Return the final score sheet as a result of calling the stored procedure
select * from T0
return
GO
```

4. Program testing

When developing student management information system, the author has successfully used the stored procedure developed above and achieved the desired results. Readers should test the stored procedure in the query analyzer of SQL SERVER.

5. Conclusion

Because stored procedures can be accessed in many other application environments, such as VB, VC++, C#, Java, Python, etc., this method has universal guiding significance; in the development of relational database, the realization of contact Table is the most complex situation, and the implementation of student curriculum contact Table has a high reference value for the application of other database projects. The logic of data access is more complex, and the form of data has changed greatly when it comes to users. This method can only access data but can not change data. To change data, it is necessary to use other ways, which is beyond the scope of this paper.

References

[1] Research on the Coping Scheme of Homologous Replay Attack Based on Conversation Fingerprint under B/S Framework [J]. Dong Benqing, Zhang Yong, Xu Binxin. Journal of Hubei Normal University (Natural Science Edition). 2015 (02)

[2] Exploration of Campus Network System Based on B/S Model [J]. Wellingli, Li Tie. Heilongjiang Science and Technology Information. 2010 (07)

[3] Research and Design of Network Topology Discovery System Based on B/S Architecture [J]. Li Kang. Science and Technology Information. 2010 (23)

[4] Construction of network examination system based on B/S structure [J]. Lu Hongcai. Science and Technology Information (Academic Research). 2007 (21)