International Journal of Database Theory and Application
Vol.6, No.5 (2013), pp.35-44
http://dx.doi.org/10.14257/ijdta.2013.6.5.04

Code Generation for Mobile Terminals Remote Accessing to the Database Based on Object Relational Mapping

Wen Hu and Yan li Zhao

School of Computer and Information Engineering, Harbin University of Commerce, 150028
zhylclover@126.com

Abstract

Research and analyze the mobile terminal remote accessing to the database interfaces and methods. On this basis, proposed a middleware forms based on Object Relational mapping. The middleware achieves mobile terminals remote accessing the database and managements of mobile terminals to connect to the database and access mechanisms. This paper adopts the methods that mobile terminals remotely access to database based on ORM and it defines ORM rules that the mobile terminal remotely accesses to the database. It designs and proposes new approaches for mobile terminals and database transmission of information and management. The method achieves visualization of the database interface.

Keywords: *Mobile terminals, Database, Object Relational mapping, Android*

1. Introduction

With the development of mobile networks, especially the appearance and application of smart terminals give users to bring new evolution of needs and demands. The rise of Tablet PCs and smart phones has brought the era of mobile terminals. In some cases, intelligent terminal has replaced the traditional computer, and intelligent terminal is welcomed more by users [1].

With the widespread use of mobile terminals, the number of mobile terminals application also has rapid development and also development prospects are very optimistic. There are two ways to make an application to get data: local access, remote database access. When the local accesses to data, the mobile terminal accesses its own database. For remote data acquisition, in most cases, mobile terminal connects to database and obtains the required data through the server transfer. Therefore developer needs to develop remote database server-side programs and arranges data packet protocol. This method is relatively flexible, but increasing the cost of development and maintenance [2, 3].

That mobile terminals remote accessing to the database is made up of two parts: (1) Mobile terminals access Web servers over the Internet, according to user' requirements, such as the user's name, address and other keywords are submitted to the Web server, or your queries are directly submitted to the server; (2) Server retrieves the relevant information and returns the results to the mobile terminals, then displays to the user.

When the mobile terminals remotely access to database, mobile terminals not only need to provide critical information for remote database access, but also consider the style of returned results. Different accessing way leads to the style of different results, so developers need different processing. This way reduces the development efficiency and increase the portability of programs.

ISSN: 2005-4270 IJDTA

2. Problem Analysis and Related Research

2.1. Problem Analysis

It has become an important topic of mobile Internet development to develop high reliability application quickly and efficiently. Software visualization is an important way to solve the problem. Software visualization could play an important role in software engineering in the future [4]. At present, it has many visual software development tools, such as DroidDraw[5] which is one kind of the interface tool of Android, Mobiforms which offered by Opera company[6], DoirdDB[7] and so on.

Interaction of Mobile terminal and database has primarily manifested in three areas: information sharing and publishing, the application of interaction and the information store, and application of business process. Fundamentally, these cover all aspects of information processing [8]. It has higher requirement to access to the database with the development of mobile terminals. Currently, the way in Web applications is using replication and is difficult to learn. This paper proposes a new method for remote access to the database by mobile terminal. Through the method of ORM [9], this paper studies the primary method that mobile terminals remotely access to the database, in order to improve the efficiency and practicality of the program. It has built visual development methods which access database remotely. As shown in Figure 1:

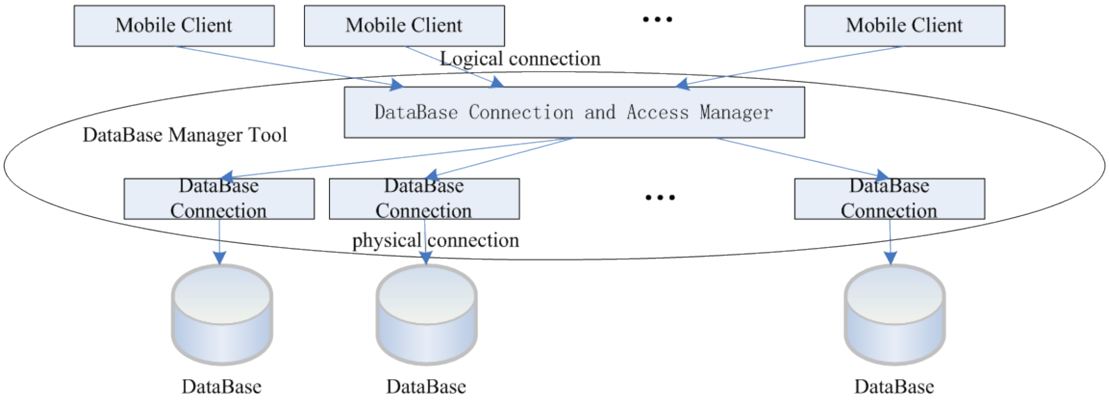


Figure 1. Database Management Middleware

We study and achieve the methods of research for remote accessing to the database by mobile terminals according to ORM. This approach provides the interface of global unified view and access. For example as Android, developers can quickly realize remote access to the database by Android terminal. To generate table structure of the database on terminal and to realize Android terminal to remote access database, the data could return to the terminal in a uniform format after accessing the database. It is convenient for Android terminal to process data simply and efficiently. This tool supports many types of database querying, updating, data transmission and delivery operations and generates a database resource configuration file (for example, the type of database, the JDBC driver name, URL, user name and password, and so on). Then according to configuration file automatically deploy services and provide with database querying and access for users.

2.2. Related Research

The visual database tools and methods which use mobile terminals to remotely access database need a specific rule definition. According to the designing rule it is conducive to use its features. This article presents a visual software development method based on ORM. The functionality of the program is defined as objects. Through feature combination of objects and the corresponding relationships of objects can automatically form part of the program and combine with the mobile terminals program and you can finish design of programming.

Resource requirements when you design a database schema, you need to determine: mobile terminals access to the database parameter, the local database mode, the remote database schema. Second, include the type of database, the remote server IP, port, database name, read -only user and password parameters. The terminal software generated the local database table and view information after setting the access parameters of the above.

1. Resource requirements determine database mapping between the terminal database schema and remote database schema, through ORM method, the terminal structure of the database schema be determined. Figure 2 shows that the database fields mapping mode are determined. Generally, the terminal database can contain one or more tables and a table contains one or more fields [10]. When you create a table in the terminal database, you can either create all remote database table fields or a subset of the remote database table fields.

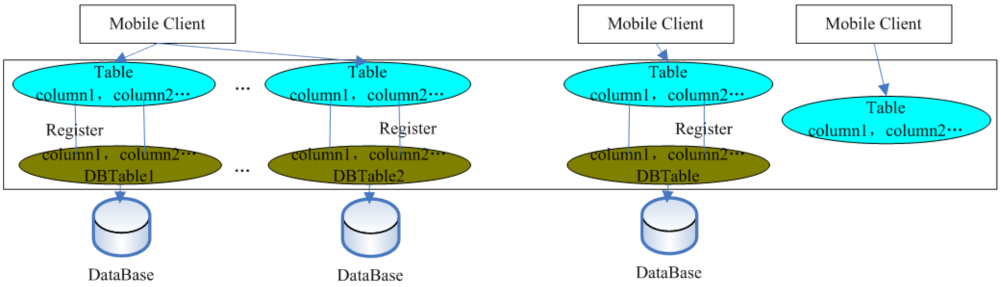


Figure 2. Fields ORM

2. By the way, it will be based on the real database server converts the request to query the database based on resource discovery requests, and provide a consistent interface to access the database using this method. A property equivalent mode combined with XML Schema technology achieves function. Figure 3 shows querying and updating schematic.

(1)When the database query and update remote database operation, you need to keep consistency between widget ID of mobile terminals and the remote database table fields. Especially, when the remote database data is stored to local database, you must keep consistent between the ID of widget, remote database table fields and the terminal database field.

(2) By calling generated code, the mobile terminals access to the remote database and results are returned in unified XML format. Developer can use API process to the results.

(3)When fields cannot be consistent, you need to use other operations. Resource users are required to consider these operations.

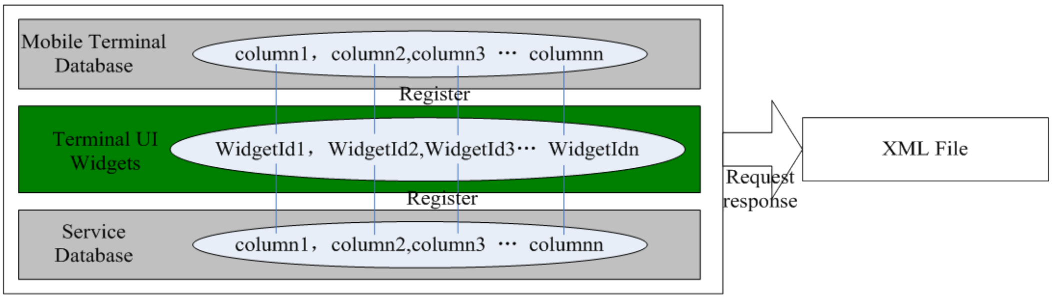

Mobile Terminal
Database
column1，column2,column3 … columnn
Register
Terminal UI
Widgets
WidgetId1，WidgetId2,WidgetId3… WidgetIdn
Register
Service
Database
column1，column2,column3 … columnn
Request
response
XML File

Figure 3. Querying and Updating Schematic

Figure 2 and 3 describe fields relationship, the line represents a mapping each other. As shown in Figure 3, when the mobile terminals send request to the remote database, they need corresponding mapping relationship between interface ID of the widget and the remote database table fields. Then you will see detailed definition rules in Section 3.

3. Remote Access to Database Based on ORM

3.1. ORM Mode

Object Relational mapping (ORM) in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. ORM mainly achieves mapping for objects to relational database data. It studies the mobile terminals remote accessing to the database based on ORM and adopts to similar and one-way one-to-one relational mapping mode. When mobile terminals remotely access to database, mobile terminals data corresponds to remote databases data and interface widget data corresponds to the remote database data.

Set of tables in the remote database contains a set of tables for mobile terminal. Mobile terminals table fields corresponding to the remote database table fields are a one-to-one association. All of mobile terminals table fields belong to the remote database table fields. By viewing the database structure for mobile terminals can know the structure that must be present in the remote database. This can be depicted in Figure 4.

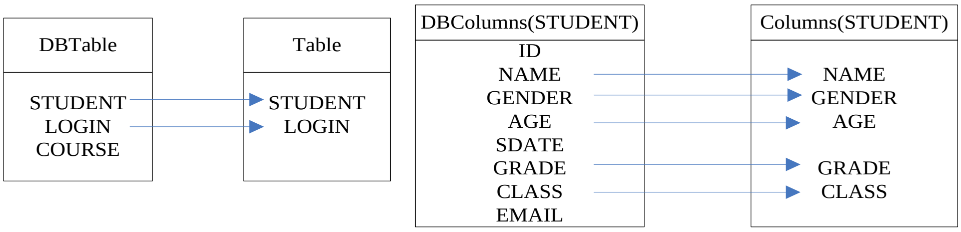


Figure 4. Similar and one-way One-to-One Relational Mapping Mode

3.2. Defining ORM Rules

Mapping relational information between the mobile terminals database schema and the remote database schema specifically refers to mapping relation between the remote database field and the terminal database field. Mapping relational information between widget of mobile terminals and the remote database field refers to mapping relation between the interface widget ID of mobile terminal and a field in the remote database. Its ORM can be defined as: Mappings relational set (MappingRelationship) is a subset of set mapping rules (MappingRule). A mapping rule includes a subset of table relationships (TableRelation), a subset of fields mapping (ColumnMapping) and a subset of terminal widget ID and field relations (IdColumnRelation). All elements of the field mapping subset must belong to the same database table (DBTable). Their relationship can be expressed as follows:

Rule1: In ORM rules, set MappingRelationship, MappingRules are two non-empty sets, and MappingRelationship is composed of all the elements with a subset of the MappingRules. MappingRules set is composed of three Relational object elements, namely TRs, CM, CR. They have the corresponding constraints between terminal fields, database fields and terminal interface ID. TCRelations is the affiliation between tables and fields, and the field is an element of a table. They are defined as follows:

$MappingRelationship = \{mr|mr \in MappingRules\};$

$MappingRules= \{<TRs,CM,ICRs>|\ TRs\subseteq TableRelations,\ CMs\subseteq ColumnMappings,$ $ICRs\subseteq IdColumnRelations,\ \forall cm1,cm2,icr3,\ icr4,cm1 \in CMs,\ cm2 \in CMs,icr3 \in ICRs,\ icr4$

∈ICRs→cm1.DBColumn.DBTable=cm2.DBColumn.DBTable,icr3.id=cm1. DBColumn, icr4.id=cm2.column};

TCRelations= {field| field ∈table};

Rule2: In table ORM, set *table1* is an element of Tables in mobile terminals, and *table2* is an element of DBTables in remote database. Relation is a subset of Relations between *table1* and *table2*. They are defined as follows:
TableRelations={<(table1,table2),relation>| table1 ∈Tables, table2 ∈DBTables, Tables⊆DBTables, relation ∈Relations} ;

Relations= {"=", "<", ">", "<>", ">=", "<="};

Rule3: In fields ORM, set *column* and *DBColumn*. *Column* is an element of Columns. *DBColumn* is an element of DBColumns in remote database. Columns is a subset of DBColumns. They are defined as follows:
ColumnMappings={<column,DBColumn>|column ∈Columns,DBColumn ∈DBColumns, Columns ⊆DBColumns } ;

Rule4: In field and widget ORM, set *id* is ID in terminal widget, and *column* is field in mobile terminals, and *dbcolumn* is field in remote database. They are defined as follows:
IdColumnRelations={<(id,column,dbcolumn),realtion>|id ∈Ids, column ∈ Columns,dbcolumn ∈DBColumns} ;

Relations= {"=", "<", ">", "<>", ">=", "<="}.

Among them, "MappingRule" is mapping rules set; "ColumnMapping" is field mapping set; "TableRelation"is a set of tables relationships; "IdColumnRelations" is relational set between mobile terminal widgets and database table fields set; "Columns" is the set of fields in the local database; "DBColumns" is a set of fields in the remote database; "DBTables" is a set of tables in the remote database; "Tables"is a set of tables for the mobile terminals; "Ids" is a set of id for mobile terminal widgets. And id value of terminal widgets and some of the necessary widget variables of the mobile terminal interface programming must be the same name and this is conducive to the terminal interface and database information interoperability. "Relations" is a set of Logical relationship.

4. Code Generation based on ORM

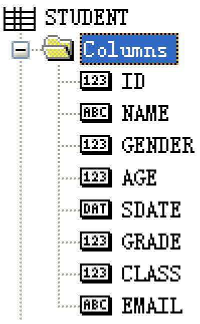
4.1. Reference ORM Rules

According to the 3.2 rules, assuming that remote database has login, student, course tables. Then DBTable= {login, student, course}; DBColumn contains DBColumnLogin, DBColumnStudent and DBColumnCourse; DBColumnLogin={username,password}, DBColumnStudent={id, name, gender, age, sdate, grade, class, email }. According to rule 2, now you can create mobile terminal data table. You can create one, two, or three tables. Now you can establish table named student and login. Tables= {login, student} and Tables ⊆DBTable. According to rule 3, the table field is only present a subset of fields in the remote database. ColumnLogin= {username, password}, ColumnLogin⊆DBColumnLogin, ColumnStudent={name, gender, age, grade, class} and ColumnStudent⊆DBColumnStudent.

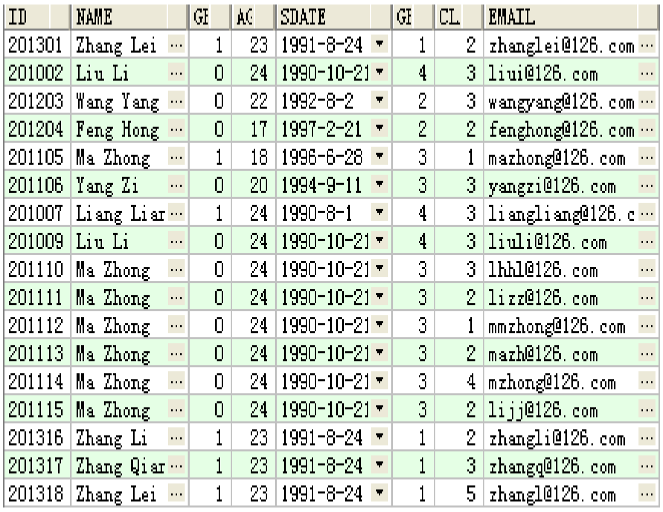
When student's basic information is displayed on the mobile terminal interface, such as name, gender, age, grade, class, you will need to set widget id corresponding to the database field. According to rule 1 and 4, Ids= {name, gender, age, grade, class}, it not only satisfies the terminal database field mapping, but also satisfies the remote database field mapping. Finally, it achieves querying, updating and other operations.

4.2. Mobile Terminal and the Database's ORM

Now you can create tables and fields in the remote database. As shown in Figure 5.Table named STUDENT and fields of STUDENT include id, name, gender, age, sdate, grade, class and email. Figure 6 shows STUDENT's data in the remote database.



STUDENT
- Columns
  - 123 ID
  - ABC NAME
  - 123 GENDER
  - 123 AGE
  - DAT SDATE
  - 123 GRADE
  - 123 CLASS
  - ABC EMAIL



ID	NAME	GI	AG	SDATE	GI	CL	EMAIL
201301	Zhang Lei	1	23	1991-8-24	1	2	zhanglei@126.com
201002	Liu Li	0	24	1990-10-21	4	3	liui@126.com
201203	Wang Yang	0	22	1992-8-2	2	3	wangyang@126.com
201204	Feng Hong	0	17	1997-2-21	2	2	fenghong@126.com
201105	Ma Zhong	1	18	1996-6-28	3	1	mazhong@126.com
201106	Yang Zi	0	20	1994-9-11	3	3	yangzi@126.com
201007	Liang Liar	1	24	1990-8-1	4	3	liangliang@126.c
201009	Liu Li	0	24	1990-10-21	4	3	liuli@126.com
201110	Ma Zhong	0	24	1990-10-21	3	3	lhhl@126.com
201111	Ma Zhong	0	24	1990-10-21	3	2	lizz@126.com
201112	Ma Zhong	0	24	1990-10-21	3	1	mmzhong@126.com
201113	Ma Zhong	0	24	1990-10-21	3	2	mazh@126.com
201114	Ma Zhong	0	24	1990-10-21	3	4	mzhong@126.com
201115	Ma Zhong	0	24	1990-10-21	3	2	lijj@126.com
201316	Zhang Li	1	23	1991-8-24	1	2	zhangli@126.com
201317	Zhang Qiar	1	23	1991-8-24	1	3	zhangq@126.com
201318	Zhang Lei	1	23	1991-8-24	1	5	zhangl@126.com

Figure 5. Structure of STUDENT Table **Figure 6. Data in STUDENT Table**

You can create mobile terminal database tables and fields through visualization tools after the remote database has created. Then according to the mapping rules needed, you can establish STUDENT table in the mobile terminal SQLite database. Tables= {student, login}, STUDENT= {name, gender, age, grade, class}, LOGIN= {username, password}, then Tables⊆DBTable, ColumnStudent⊆DBColumnStudent. They satisfy ORM rules.

4.3. ORM Generation

(1)Create Table in Mobile Terminal

There are three tables including COURSE, LOGIN and STUDENT in remote database. Now you can create STUDENT and Login in mobile terminals. STUDENT fields include name, gender, age, grade and class. LOGIN fields include username and password. Creating table process is shown as follows:

Step1: According to rule 2, table relationship is established between the mobile terminal and remote databases. Judging set Tables = {STUDENT, LOGIN} is not empty. And by comparison Relations, they satisfy Rule 2, Tables ⊆DBTables, so you can create tables SRUDENT. Then, Tables = {LOGIN}, DBTables = {COURSE, LOGIN, STUDENT}. When the Tables is empty, creating table finishes in mobile terminal and code creating table in mobile terminal is generated.

Step2: According to rule 3, you can create field terminal STUDENT. Judgment STUDENT= {name, gender, age, grade, class} is not empty, so Columns= {name} and DBColumns={ id, name, gender, age, sdate, grade, class, email }satisfy Columns⊆DBColumns. Then you can create name field. According to the Step2, you can establish other fields. When the STUDENT is empty, creating table field ends, return to step1.

(2)Query Data in Remote Database

Mobile terminals operating remote database includes querying and updating remote database. Updating remote database includes inserting, modifying and deleting data. Now we describe querying process. It assumes that you have id’s set Ids in mobile terminal: Ids= {querybutton, nameT, genderT, ageT, gradeT, classT, nameT1, gradeT1, name, gender, age, grade, class}. Click on the button with the ID value “querybutton” in mobile terminals and query STUDENT table in remote database. You can query name, gender, age, grade and class. There are querying conditions, including grade and name value that are obtained from EditTexts of ID grade and ID name in terminal interface.According to the class, results are sorted in ascending order.

Step1: Gets the ID of the query button. By iterating Ids set, you can find the ID value for the *querybutton* button.

Step2: By iterating DBTables set, judging if STUDENT exists. If it does not exist, the querying fails;

Step3: In remote database you need iterate SRUDENT’s fields set: DBColumnStudent ={id, name, gender, age, sdate, grade, class, email } and get querying fields: name, gender, age, grade and class. If it does not exist, return to Step2, and select another table again, until all tables are not satisfied. If not, go to step4.

Step4: Get conditions. According to Rule 1 and 4, the terminal widget ID and field relations are the same and unique, so it can be inferred that values of widget id of name and grade are name and grade in the interface program of mobile terminals. And condition values are passed by the Android Terminal. According to Android application programming requirements, grade value is obtained by using grade.getText () in mobile terminals and name value is from name.getText (). If name or grade does not exist in the set of Ids, the querying fails, it needs to reset the terminal programming code.

Step5: Select the order in which querying results are displayed. Iterate DBColumnStudent set and select the class field. If it does not exist, return to Step2.

Step6: When operation finishes, it generates the query code.

Step7: The mobile terminals will regard generated code in step6 as part of the programming code. When you query the remote database, it returns unified format results.

Through the above steps, the mobile terminal will return data in XML format to the client interface. Now interface designing developer only need to call the corresponding API, processing data, and complete the design.

4.4. Code Generation for ORM

(1) Generate Code for Creating Table in Mobile Terminal

According to ORM, you can generate corresponding code in Section 4.3. And you can generate creating mobile terminal database table STUDENT and LOGIN code. As follows:

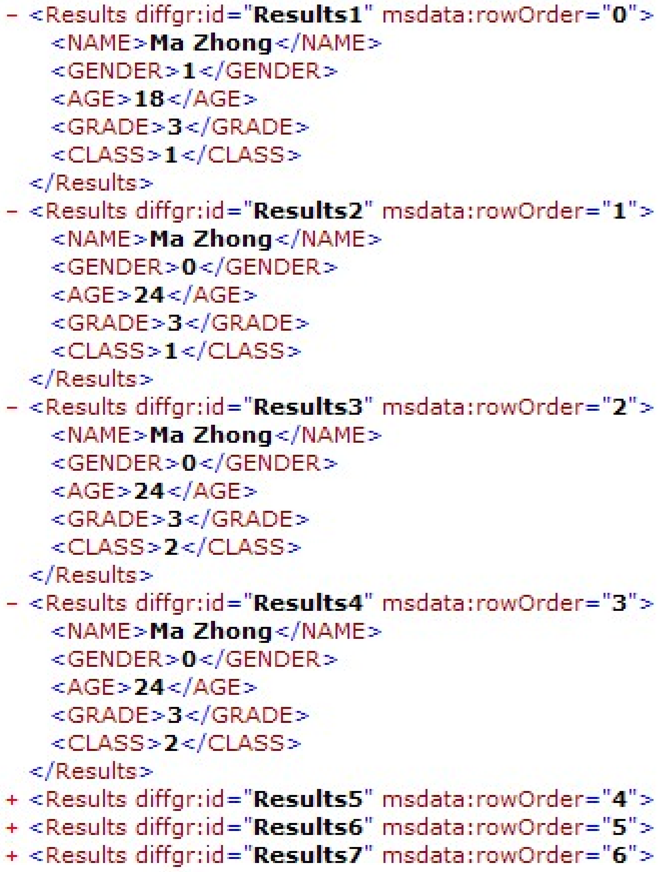
```
public static final String LOGIN = "LOGIN";

public static final String USERNAME = "USERNAME";

public static final String PASSWORD = "PASSWORD";

String tableLOGIN="CREATE TABLE " + LOGIN + "(" + USERNAME+ " TEXT," +
PASSWORD + " Integer"+ ");"

public static final String STUDENT = "STUDENT";

public static final String NAME = "NAME";

public static final String GENDER = "GENDER";
public static final String AGE = "AGE";
public static final String GRADE = "GRADE";
public static final String CLASS = "CLASS";
String tableSTUDENT= "CREATE TABLE " + STUDENT + "(" + NAME + " TEXT PRIMARY KEY," + GENDER + " Integer,"+ AGE + " Integer,"+ GRADE + " Integer, " + CLASS + "Integer "+ ");"
```

(2)Generate Code for Querying Data in Remote Database

Mobile terminals query student table of a remote database and produces SQL. As follows:

```
"select "+"name,gender,age,grade,class"+" from "+"student"+" where "+"grade="+grade.getText()+" and "+"name=" +"\'" +name.getText()+"\'" +" order by "+"class"
```

Querying results are returned in XML format. Figure 7 shows a part of the query results. Through mobile terminal interface design handles querying results, and you can get the data you need.



```
- <Results diffgr:id="Results1" msdata:rowOrder="0">
    <NAME>Ma Zhong</NAME>
    <GENDER>1</GENDER>
    <AGE>18</AGE>
    <GRADE>3</GRADE>
    <CLASS>1</CLASS>
  </Results>
- <Results diffgr:id="Results2" msdata:rowOrder="1">
    <NAME>Ma Zhong</NAME>
    <GENDER>0</GENDER>
    <AGE>24</AGE>
    <GRADE>3</GRADE>
    <CLASS>1</CLASS>
  </Results>
- <Results diffgr:id="Results3" msdata:rowOrder="2">
    <NAME>Ma Zhong</NAME>
    <GENDER>0</GENDER>
    <AGE>24</AGE>
    <GRADE>3</GRADE>
    <CLASS>2</CLASS>
  </Results>
- <Results diffgr:id="Results4" msdata:rowOrder="3">
    <NAME>Ma Zhong</NAME>
    <GENDER>0</GENDER>
    <AGE>24</AGE>
    <GRADE>3</GRADE>
    <CLASS>2</CLASS>
  </Results>
+ <Results diffgr:id="Results5" msdata:rowOrder="4">
+ <Results diffgr:id="Results6" msdata:rowOrder="5">
+ <Results diffgr:id="Results7" msdata:rowOrder="6">
```

Figure 7. Query Results in XML Format

5. Conclusion

With the rapid development of mobile Internet, mobile terminal accessing remote database is paid more and more attention. This paper presents a visual coding method for the mobile terminal remote accessing to database. Through ORM, you can visually design mobile terminal database structure. You can achieve access and update to remote databases with code as little as possible. Remote database fields, the mobile terminal database fields and ID values of interface widget are corresponding, and they will execute routine database operations. By ORM, this paper designs and implements code generation tool for mobile terminal remote accessing to the database, increases development efficiency, and reduce the learning curve. It is conducive to realize mobile terminal remote accessing to the database operations.

References

[1] Z. JinZeng and M. XiaoFeng, "Research on Mobile Web Search.Journal of Software, vol. 23, no. 1, **(2012)**, pp. 47-64.

[2] P. Pocatilu, "Developing Mobile Learning Applications for Android using Web Services", InformaticaEconomică, vol. 14, no. 3, **(2010)**, pp. 106-115.

[3] S. Lee, "Creating and Using Databases for Android Applications", International Journal of Database Theory and Application, vol. 5, no. 2, **(2012)**, pp. 99-105.

[4] W. Quanyuan, "Network Computing Middleware", Journal of Software, vol. 24, no. 1, **(2013)**, pp. 67-76.

[5] http://code.google.com/p/droiddraw/downloads/list.

[6] http://www.mobiforms.com/.

[7] http://www.droiddb.com/.

[8] M. Song and S. Haolin, "Research and Construction of a Database Middleware Prototype on Intranet", Journal of Software, vol. 10, no. 1, **(1999)**, pp. 86-91.

[9] L. Shumei, Z. Zhicheng, W. Shuming and L. Shaoyan, "Object Map in Technique of Object-Oriented", Microelectronics & Computer, vol. 22, no. 2, **(2005)**, pp. 172-182.

[10] C. XiaoWu, P. ZhangSheng and Z. QinPing, "A Schema-Reusable Method on Heterogenous Databases Access and Integration in Grid Environment", Journal of Software, vol. 17, no. 11, **(2006)**, pp. 2224-2233.

Authors

Wen Hu, master instructor, president of School of the Computer and Information Engineering, Harbin University of Commerce, backup leader of "Electronic Commerce" provincial key discipline echelon and academic leader of secondary doctoral discipline "electronic commerce and information service" in first-class doctoral discipline "business administration". His main research fields include computer network and communication, electronic commerce, embedded technology.

Yan li Zhao, Master, student of School of the Computer and Information Engineering, Harbin University of Commerce. Her main research fields include embedded technology.