

Detecting Java Code Problem in Accessing Different DBMS for Increasing Easy of Application Migration

Dimas Ari Setiawan, Imamah Adilah, Fatchurrochman

Departement of Informatic Engineering, UIN Maulana Malik Ibrahim Malang

Article Info

Article history:

Received Jul 12th, 2017

Revised Aug 20th, 2017

Accepted Oct 26th, 2017

Keyword:

System Migration,

DBMS,

Java Code,

Application Generator Course
Schedule

ABSTRACT

Any change in the organizational environment causes the need for system migration. Migration in the world of information technology is the process of moving from one system environment to another better system, in an organization. Migration can be done at the database level. Problems arise when the application can not run properly when DBMS changes occur. It is necessary to identify the program code that causes problems so that when the migration is done the application can still run well. In this paper we will identify some problems with the Java program code that causes the migration process to run smoothly and solutions to solve the problem. The study was conducted on Generation Application Schedule Lecture version 1.0 using DBMS Access, MySql, and SqlLite. The result of this research is the common Java program code to access the DBMS. For software developers the results of this study are expected to accelerate the process of system migration.

Copyright © 2017 Green Technology.
All rights reserved.

Corresponding Author:

Dimas Ari Setiawan

Departement of Informatic Engineering,

UIN Maulana Malik Ibrahim Malang

Email: fatchur@ti.uin-malang.ac.id

1. INTRODUCTION

Migration in information technology is the process of move from one system environment to another better system, on an organization (Putu, 2016).

Migration is required because :

- Needed for a better system, new and more reliable.
- Increase the productivity of the organization.
- improve services provided by the organization to consumers.

Migration can be realized by converting conventional system to digital system, integration of database with application, data migration and enterprise information system implementation.

Tse-Hsun Chen et al (Chen, 2016) "Detecting Problems in the Database Access Code of Large Scale Systems" conveying their research that the DBMS is a very important component in a modern system. It is important for developers to write program code that can access DBMS properly and efficiently. Therefore the code to access the database is sometimes a blackbox for developers.

Ion Lungu (Lungu, 2010) java offers support for development environments for distributed applications, an independent platform, provides a set of methods to access robust databases, used to create software components on the server side as well as client side.

The four steps needed to build an application using JDBC are presented briefly, emphasized on each step to be accomplished and the expected results. Each step is evaluated on the characteristics of the database system and how the JDBC programming interface adapts. The types provided by the SQL2 and SQL3 standards are analyzed by comparing with Java data types, emphasizing inconsistencies between SQL types, but also the methods that allow conversion between different data types via methods on the ResultSet object.

JDBC programs can be created with high portability because in general JDBC programming does not have a significant code difference for programming from a particular database with other databases. The main difference

in the code is simply defining the drivers of the database as well as certain SQL commands that may have certain syntax differences or special SQL commands that only exist in certain databases.

JDBC 2.0, ResultSet has more capabilities such as Scrollable ResultSet and Updateable ResultSet. Scrollable ResultSet allows the cursor move not only forward, but also backward, to a certain line and others. Updateable ResultSet allows to perform new line addition operations as well as deletion of rows in ResultSet. Table 1are defined on the ResultSet interface, used to determine the three types of ResultSet seen in terms of scrollable capabilities.

In this research software lecturing scheduling as the object. This Software will generate the schedule automatically with some constraints such as lecturer constraint, space constraint, dhuhur prayer time constraint, and constraint of Friday prayer time. Input system is a plotting course that contains information about the course, class, and lecturer. The process of arranging the lecture schedule is done by searching Depth First Search (DFS) to find the empty time slot. The output of the system is the schedule of lectures that have met the constraints that have been set.

Table1 ResultSet JDBC 2.0

TYPE_FORWARD_ONLY	ResultSet non-scrollable Can only be used for forward, from first to end.
TYPE_SCROLL_INSENSITIVE	Scrollable ResultSet Insensitive. The cursor can move forward, backward, to a certain line, or to a relative position. ResultSet usually does not display changes to the original database that occurred when the ResultSet was still open.
TYPE_SCROLL_SENSITIVE	ResultSet sensitive is scrollable, but is sensitive to changes in the original database when the object is still open.

2. RESEARCH METHOD

a. System Under Test (SUT):

i. Computations platform

1. Processor : Intel Pentium Dual CPU
2. RAM : 504MB
3. Operating System : Microsoft Windows XP Proffesional 2002 Service Pack 3

ii. Software :

1. Bahasa Java
2. IDE Netbeans 8.0.1
3. XAMPP 1.7.2
4. ODBC Data Source Administrator
5. SQLite 2009 Pro
6. Mozilla Firefox
7. Automatic scheduling version 1.0

b. Steps

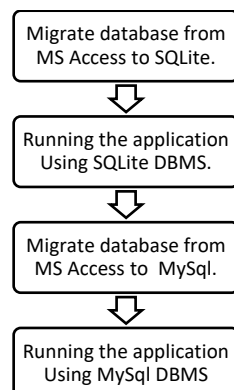


Image1Testing method

- i. Undertake database migration from MS Access database to SQLite.
- ii. Running scheduling application using SQLite DBMS. When it running smoothly, record the time to schedule the entire class of lectures. If it doesn't work, recorded what caused it and make improvements in order to run . Record all the improvements that have been made.

- iii. Undertake database migration from MS Access database to MySQL.
- iv. Running scheduling application using MySQL DBMS. When it running smoothly, record the time to schedule the entire class of lectures. If it doesn't work, record what caused it and make improvements in order to run. Record all the improvements that have been made.

Research methods

- i. Migrate database from MS Access to SQLite. Database migration is done by converting from MS Access DBMS to SQLite. The conversion process is use SQLite2009Pro software.
- ii. Changes to the program code associated with connections and the addition of drivers to the SQLite database.

```
Class.forName("org.sqlite.JDBC");
Connection=
DriverManager.getConnection("jdbc:sqlite:GenerateJadwal.db3");
```

SQLite only supports JDBC version 1 while the following syntax is a newer version.

```
statement
connection.createStatement(ResultSet.TYPE_FORWARD_ONLY,ResultSet.CONCUR_READ_
ONLY);
```

There is a consequence with this replacement that is on the Reusable class. This class is intended as a support class for many other classes to handle repeated tasks such as the `getArray1Dimension ()` method used to get data from the database and store it in array.

```
public String[] getArray1Dimensi(Connection connection,Statement statement,String ssl)
{
    int banyakdata=0;
    String[] yangdicari=new String[banyakdata];
    Vector<String> v=new Vector<String>(10);

    try
    {
        statement
        connection.createStatement(ResultSet.TYPE_FORWARD_ONLY,ResultSet.CONCUR_READ_ONLY);
        String sql=ssl;
        ResultSet rs=statement.executeQuery(sql);

        while(rs.next())
        {
            v.add(rs.getString(1));
        }

        banyakdata=v.size();
        yangdicari=new String[banyakdata];
        for(int i=0; i<banyakdata; i++)
        {
            yangdicari[i]=v.get(i);
        }

        statement.close();
    }
    catch (Exception DBException)
    {
        System.err.println("Error : " + DBException);
    }
    return yangdicari;
}
```

- iii. Migrate database from MS Access to MySQL. The conversion process is done by using Bullzip software.
- iv. Changes to the connection code class :

```
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection
DriverManager.getConnection("jdbc:mysql://localhost/generateJadwal",
"root", "");
```

improvements are done by quotation marks on the Range field to differentiate the meaning of the Range field.

```
ssql="select `Range` from JamKuliah
where SKS='"+besarsks+"' and
KodeJam<>"+batas1+" order by
KodeJam";
```

- v. The next step that needs to be done is to examine which of the code of the program can be used in all DBMS tested. The result of comparison of time required in the computation process before and after done efficiency in the form of tables.

3. RESULTS AND DISCUSSION

Results with different databases :

- i. Program code for mysql run it ms access (running correctly)
- ii. Program code for mysql and run it sqlite (running correctly)
- iii. Program code for sqlite and run it ms access (running correctly)
- iv. Program code for sqlite and run it mysql (running correctly)

Table 2. Fixed code

First code	Fixed code
statement connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);	statement connection.createStatement(ResultSet.TYPE_FORWARD_ONLY,ResultSet.CONCUR_READ_ONLY);
rs.last(); int n=rs.getRow(); int p=0; rs.beforeFirst(); qdata=n; find=new String[qdata]; while(rs.next()){ find[p]=rs.getString(1); p++;}	Vector<String> v=new Vector<String>(10); while(rs.next()){ v.add(rs.getString(1));} qdata=v.size();find=new String[qdata]; for(int i=0; i<qdata; i++){ find[i]=v.get(i);}
ssql="select Range from JamKuliah where SKS='"+besarsks+"' and KodeJam<>"+batas1+" order by KodeJam";	ssql="select ` Range ` from JamKuliah where SKS='"+besarsks+"' and KodeJam<>"+batas1+" order by KodeJam";

Table 3. Estimated time

No	DBMS	Estimated time (s)	
		Empty lecture schedule table	Filledlecture schedules table
1	MS Access	110	2605
2	SQLite	46	1027
3	MySQL	51	991

Discussion

Program code on SQLite is more complete than all and can run on the DBMS being tested. The application code for the SQLite DBMS is used as a standard code that is reusable for different DBMS. Computing time on each DBMS is different. From table no 3 The fastest time for the empty table schedule table is the SQLite DBMS and when the table loads the fastest MySQL.

CONCLUSION

Repairing the java program code associated with the database has been done by replacing all the initial code into new code as shown in table 2. Trials made in the lecture scheduling application automatically find the above and still possible to find other codes that Is common to other applications.

REFERENCES

- Agus, Putu. 2016. Integrasi dan Migrasi Sistem Teori dan Praktek. Penerbit Informatika. Bandung.
- Allenstein Brett., Yost, Andrew., Wagner, Paul., Morrison, Joline. 2008. A Query Simulation System To Illustrate Database Query Execution.
- Bhagwant , Barkha Keni., Madijagan, M., Vijayakumar, B. 2007. A Generic Model for Querying Multiple Databases in a Distributed Environment Using JDBC and an Uniform Interface.
- Chen, Hao. 2010. Comparative Study of C, C++, C# and Java Programming Languages.
- Chen, Tse-Hsun., Shang, Weiyi., Hassan, Ahmed E., Nasser, Mohamed., Flora, Parminder. 2016. Detecting Problems in the Database Access Code of Large Scale Systems.
- LUNGU, Ion., MERCIOIU, Nicolae. 2010. Database Access Through Java Technologies.
- MySQL Documentation. <https://dev.mysql.com/doc/refman/5.7/en/keywords.html> diakses 27 March 2017.