Investigation of SQL Clone on MVC-based Application

Fawwaz Ali Akbar[1], Siti Rochimah[1], Rizky Januar Akbar[1]

Abstract*—Model-View-Controller (MVC) design pattern is design pattern that is suitable for interactive systems. MVC is adapted in desktop and web-based applications. Moreover, many frameworks are adapting MVC pattern. Each layer of MVC has a different function. The main function of the model layer is query to the database system that represented by SQL language. In software development, code duplication or code clone is a serious problem because it will impact on the maintenance process. Associated with model layer and code clone, clone detection approach that exists today is not effective to detect clones in the model layer represented by SQL language, because the definition of code clone is not suitable for SQL clone. SQL is declarative language that is different from the common programming language like C and Java. So, the definition of code clone must be adjusted with characteristic of SQL. In this research, we investigate the existence of SQL clone on MVC-based application and define the types of SQL clone. We define four types of SQL clone and they are confirmed exist in MVC-based application datasets that used in this research.

Keywords*—Clone detection, Model Layer, Model-View-Controller, SQL Clone.

I. Introduction[1]

Model-View-Controller (MVC) design pattern is design pattern that is suitable for interactive systems [1]. MVC pattern divides system in three layers: Model, View and Controller. Each layer of MVC has a different function. The main function of the model layer is data access [2]. MVC is adapted in desktop and web-based applications. Moreover, many frameworks adapt MVC pattern.

In software development, code duplication is a serious problem because it will impact on the maintenance process, such as, increasing maintenance cost and bug propagation [3]. According to previous research, the existence of code clone in the system is quite large. The code clone occurs about 20-30% in a large system [3] and about 17-63% in web-based applications [4].

Associated with model layer and code clone, clone detection approach that exists today is not effective to detect clones in the model layer that is represented by SQL language, because the definition of code clone is not suitable for SQL clone. SQL is declarative language that is different from the common programming language like C and Java.

Definition of code clone and its types will impact to detection method and approach, because all approach and method refer to code clone definition. SQL has different characteristic with the common language like C, Java and PHP that is not suitable for code clone definition. Therefore, code clone definition and code clone detection approach that exist today is not suitable for SQL clone detection. So, the definition of code clone must be adjusted with SQL language characteristic.

Furthermore, there is still no clear boundary that determines two SQL scripts to be considered as clone.

This reseach define SQL clone definition and its types, based on the different characteristic of SQL language and common language such as C, Java, PHP.

With SQL clone definition and its types, we investigate the existence of SQL clone on MVC-based application.

II. Literature Review

This literature review will be discussed about the theoretical basis of this research, such as MVC, code cloning, and SQL.

A. Model-View-Controller (MVC)

Model-View-Controller (MVC) design pattern is design pattern that is suitable for interactive systems [5]. MVC pattern divides system in three layers: Model, View and Controller. Each layer of MVC has a different function. The Model layer is related to the data, the View Layer is related to the presentation and the Controller layer related to the user interaction. [6]**.**

There are many models of interaction between model layer, view and controller. An overview of MVC interactions is: the view layer interacts with the controller layer, then the controller layer changes the state of the model layer. The state change of the model layer will be passed to the view via the controller layer.

The reason for the many models of interaction in MVC is that the development and adaptation of MVC patterns on many systems (desktop and web) makes MVC patterns have many variations of interaction patterns and components in the Model, View and Controller layers [7].

Many frameworks are built by adopting MVC patterns as architectural design patterns. On the web and web-based applications, MVC pattern adaptation to a framework is popular.

B. Code Clone

By definition, software cloning is the activity of copying and pasting fragments of existing code, with or without

[1]Fawwaz Ali Akbar, Siti Rochimah, Rizky Januar Akbar are with Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Kampus ITS Sukolilo, Surabaya 60111, Indonesia. E-mail: fawwaz.ali.akbar15@mhs.if.its.ac.id; siti@if.its.ac.id; rizky@if.its.ac.id.

modification, into other code, in the software development process. The result of copying and pasting code fragments is software clones, or commonly known as code clone [8].

There are some pros and cons about the existence of code clone in a system. The pros and cons are based on the advantages and disadvantages of the code clone in a system. One of the advantages of code clone is to speed up the software development process if software requirements change. While the most frequently discussed of code clone disadvantage is the increasing cost of system maintenance and bug propagation [8]. The reason of developers do code cloning is the limited of time and resources available [8].

In addition, the existence of code clone in the system is quite large. Some studies suggest that the code clone occurs about 20-30% in a large system [8], and about 17-63% clones on the web-based application [4].

The code clone is generally divided into 4 definitions or types, namely type 1 (exact clone), 2 (renamed clone), 3 (near-miss clone) and 4 (semantic clone). Here is an explanation of each clone type [8]:

1) Type 1 (exact clone): A pair of code fragments has the exact same code text (identical) but may have different on spaces and comments.
2) Type 2 (renamed clone): a pair of code fragments that has the same structure, but may have different in the identifier, literal, data type, layout and comment.
3) Type 3 (near-miss clone): a pair of code fragment of clone type 1 or 2 with the addition or deletion of fragment code.
4) Type 4 (semantic clone): a pair of code fragments is declared as clone type 4 if it has the same function or behaviour, but does not have the same structure or text.

Regarding the definition of code clone and its types, the researchers have different opinions. A survey conducted by Chatterji [9] states that researchers in the field of code clone still have not fully agreed on the definition of clones and its types. The survey findings suggest that most researchers agree on the definition of code clone type 1 and 2, but not for type 3 and 4.

C. Structured Query Language (SQL)

Almost all applications require data storage. The most commonly used data storage is the relational database. The programming language used for interaction with relational database, such as inputting data, changing data and retrieving data in the relational database, is SQL or Structured Query Language.

SQL language statements are divided into three categories that have different functions. The three categories are SQL schema statement, SQL data statement, and SQL transaction statement. SQL data statement is a SQL statement used to manipulate data in the database. It includes INSERT, SELECT, UPDATE, and DELETE statements [10]. The area of this study is in SQL data statement focusing on SELECT statement.

Generally, SELECT statement is divided into three main parts: SELECT clause, FROM clause and WHERE clause. SELECT clause is used to specify the columns to be displayed on a query. FROM clause is used to define the tables involved in a query. WHERE clause is used to define the conditions for queries. In addition to WHERE clause, there are some clauses that can be used for specific purposes such as GROUP BY, HAVING, and ORDER BY [10].

In this research, SELECT statement is divided into several parts: SELECT clause, FROM clause, and filter clause. Filter clauses include WHERE, GROUP BY, HAVING, and ORDER BY. It can be seen on Figure 1.

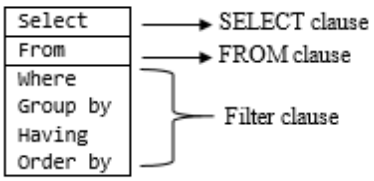

Figure 1. Part of SELECT statement that is used in this research

III. METHOD

The method in this study is divided into two phases. First, is defining SQL clone and its types, second is investigation the existence of SQL clone on MVC-based application.

In the first phase of defining SQL clones and its types, the research discusses about the basis for determining the SQL clones and experiments performed to determine the types of SQL clones.

In the second phase, the research discusses about the approach for investigating the existence of SQL clones and its types that have been defined. The purpose of the second phase is to prove for the existence of defined clone types.

A. SQL Clone Types

SQL has different structure with common programming language such as Java, C, and PHP. Definition of code clone and its types need to be adjusted with SQL characteristic and structure.

This research defines SQL clone and its types based on two things. First, it is based on the definition of code clone and its types, and second, it is based on the result of research in the SQL script similarity area. Research in SQL script similarity areas includes SQL script mutations and recommendation systems for SQL queries.

The first basis of our definition is code clone definition and its types. This research adjust the definition of code clone to SQL clone definition, because the definition of code clone needs to be adjusted with SQL characteristic.

Converting or adjusting code clone definition in different area or object has been done by Störrle et al [11]. They convert the definition of code clone to UML clone by adjusting code clone definition in UML area. Störrle et al define four types of UML clone of type A, B, C and D which are the representation of code clone type 1, 2, 3 and 4.

The second basis of our definition is the result of research in the SQL script similarity area. Research in SQL script

similarity areas includes SQL script mutations and recommendation systems for SQL queries.

The process of defining SQL clones refer to the results of the investigation of SQL script mutations. The SQL mutation is used in definition process because SQL mutation provides information about the similarity level of SQL scripts [12]. This research also use the principle in SQL query recommendation for defining SQL clones [13].

The idea of using SQL mutation for define SQL clone is from code clone testing method. In code clone area, code mutation can be used for generating mutated code. The program code will be mutated into codes that accommodate every type of code clone. That mutated code is used for datasets to evaluate code clone detection method. Mutation code is used for testing in code clone because the available datasets for code clone research are few [14]. In this research, we reverse the purpose of code mutation by using mutation results to define the clone type in SQL scripts.

The SQL mutation experiment uses tools from SQL mutations [15]. The input of the tools is a query and the output is query mutation of the initial query. This query mutation experiment is only for SELECT statements. The clauses used in this experiment include SELECT clause, FROM clause, WHERE clause, GROUP BY clause, HAVING clause, GROUP BY clause, and aggregate functions. The results of the SQL mutation experiment can be summarized as follows:

- The use of DISTINCT in the select clause is detected as an equivalent query if the other clauses are exactly the same.
- In the mutation process, a prefix of table and column names in select statement is also used.
- The changes of the value of the filter clause, such as: `where student_id = '51152983'` becomes `where student_id = 90`
- The number of SQL mutated results columns is always equal to the number of columns in the original SQL script.

The result of the SQL mutation can be used as a reference in the process of defining the SQL clone.

Before identifying the types of clones in SQL, first of all, the similarity boundary of two SQL scripts must be defined. The boundary used in this research refers to SQL query recommendation research. Two SQL scripts are considered similar if they perform queries in the same table and attribute using different condition clauses [13]. The definition of the similarity is as the basis for defining the SQL clone.

In this research, we define 4 types of SQL clone:

1) Type A

A pair of SQL script has identical form, identical in SELECT clause, a FROM clause, and filter condition clause.

Example 1 shows example of SQL clone type A:

Example 1

```
select student_id,name
from student
```

```
select studeny_id,name
from student
```

From the example 1, two SQL scripts are identical in all clauses: SELECT, FROM, and filter condition clause. So, two SQL scripts in example 1 are classified as SQL clone type A. Next example is SQL clone type A that has different input condition:

Example 2

```
select student_id,name
from student
where student_id = ‘006’
```

```
select student_id,name
from student
where student_id = ‘077’
```

Based on example 2, all input conditions will be ignored, such as '006' and '077' will be ignored. From the example 2 above, two SQL script are cassified as SQL clone type A, although have different input condition.

2) Type B

A pair of SQL script has the same SELECT clause, FROM clause, and condition clause, but it has different sequences and naming (aliases) in the columns and tables in SELECT clause, FROM clause, and filter condition clause.

Definition of SQL clone type B refers to the definition of code clone type 2. Code clone type 2 is clone which has the same structure, but different identifier, literal, data type, layout, and comments.

SQL allows changing the name of fields or tables. Because SQL accommodates alias name for field or table. In writing aliases in SQL, there are two ways, using AS keywords or directly writing the alias name after the field or table name. Example 3 shows writing alias in SQL:

Example 3

```
select student_id as id, name student_name
from student s
```

Example 4 is SQL clone type B that has a different sequence on SELECT clause.

Example 4

```
Select name, student_id from student
```

```
Select student_id, name from student
```

Example 5 is SQL clone type B that has a different sequence on SELECT, FROM and filter clause and has different name or alias in field and table.

Example 5

```
select name, student_id, faculty_name
from
student, faculty
where
student.faculty_id=
faculty.faculty_id
```

```
select student_id, name as student_name,
faculty_name as faculty
from
fakultas f, student s
where
f.id_fakultas=
s.id_fakultas
```

3) Type C

A pair of SQL scripts is declared as type C if a pair of SQL scripts have these three following conditions: it has the same SELECT clause (regardless of name and sequence differences), it has at least one of the same tables in the FROM clause (regardless of type join table), it has different filter clause.

In this definition, the definition of type 3 code clone needs to be adjusted, because the structure of programming code (C, Java, PHP, and so on) with SQL is different. The code clone type 3 specifies that there are addition and deletion of fragments in the code, but deleting and adding

fragments to SQL can produce different SQL script and different data. The example of adding dan deleting fragments in SQL script is shown in Example 6 and 7:

Example 6

```
select student_id, name
from student
```

From the Example 6 above, two fields are added (`faculty_id, address`) to SELECT clause that is represented in Example 7.

Example 7

```
select student_id, name, faculty_id, address
from student
```

Based on the Example 6 and 7, two SQL scripts are considered as different SQL script, because it will display different data field. On the Example 7, SQL script display field faculty_id and address, but the SQL script in Example 6 doesn't.

It is classified as different SQL script because the boundary used in this research refers to SQL query recommendation research. Two SQL scripts are considered similar if they perform queries in the same table and attribute using different condition clauses [13].

So based on the definition of clone clone type 3 and the boundary of SQL script similarity, adding or deleting fragments can only be applied on the filter clause. The difference with the code clone, the addition and deletion of fragments can be applied anywhere in the fragment code. But in SQL, deletion and addition of fragments can only be applied on the filter clause so that the SQL can still be considered as similar. Example of SQL clone type C:

Example 8

```
select student_id,name
from student
where student_id = '51'
```

```
select student_id,name
from student
where student_name like 'Jhon%'
```

Example 8 shows that both SQL scripts have the same column or field but produce different data because of differences in the condition clause. It can be categorized clone type C.

4) Type D

A pair of SQL script is detected as type A, B, or C, but one or both SQL scripts are part of another query, such as subquery or UNION.

SQL clone type D does not refer to the definition of code clone type 4 or semantic clone. SQL clone type D is simply defined as partial clone of another SQL script. Example 9 is example of SQL clone type D :

Example 9

```
select student_id, name
from student
where faculty_id = (
   select faculty_id
   from student
   where student_id = '9205'
)
```

```
select faculty_id
from student
where
student_id = '094801'
```

From Example 9, part of SQL script (subquery) is identified as clone with another SQL script, which is considered as SQL clone type D. SQL clone type D can occur in UNION statement such as shown in Example 10:

Example 10

```
select student_id, name
from student
UNION
select student_id, name
from alumni
```

```
select student_id, name
from student
```

B. Investigation of SQL clone

This research conducts a manual investigation of MVC-based applications to prove the definition of SQL clones and its types. The purpose of this investigation is to validate whether the definition of clone SQL and its types is valid.

The object of this research is the MVC-based web application. Its focusing is in the model layer, because all interactions to database represented by SQL script are placed in the model layer.

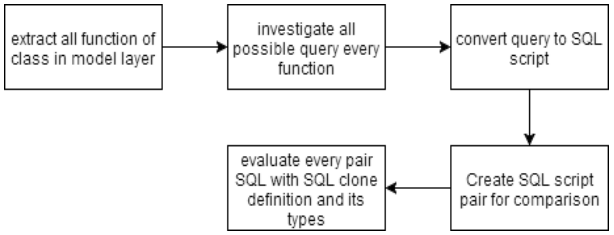

Figure 2. Investigation Approach

Based on figure 2, Manual investigation in this research is divided into five steps. The input of this investigation is classes in the model layer on MVC-based application. The output of this manual investigation is information of SQL clone and its type in MVC-based application. The investigation is conducted per-application. This is the steps of manual investigation:

1) Extracting class's function:

SQL clone investigation begins with the extraction of class functions in the model layer.

2) Investigating possible query:

After all class function have been extracted, then, investigate all possible query in every extracted function.

3) Converting query:

After all query have been extracted, the next step is convert query to SQL scripts. Queries in application is not in SQL form, but in programming language form such as Java, C, PHP, etc. depend on programming language that is used. Therefore, they need to be converted to SQL language.

4) Creating comparison pair:

Next step is create SQL script pair for comparison. This research make a comparison pair of all extracted SQL scripts. But, this research don't make comparison pair of two SQL script from the same function, because SQL script from the same function usually is the variation of another SQL script in the function.

5) Evaluating comparison pair:

The last step is evaluate every created pair SQL with SQL clone definition and its types. Pairs of SQL scripts that

meet the definition of SQL clones will be labeled as a clone and labeled with SQL clone type.

IV. RESULTS

The manual investigation is conducted with three datasets. The datasets are a MVC-based application. Based on table 1, Dataset 1 is application for manage hotel reservation. Dataset 2 is application for management boarding school. Dataset 3 is application of online tax monitoring for hotels.

TABLE 1.
DATASETS.

Datasets	Class Model	LOC Model
Dataset 1	10 Class	312 LOC
Dataset 2	13 Class	1551 LOC
Dataset 3	10 Class	748 LOC

In dataset 1, there are 10 classes in the model layer with a total of 312 LOC. In dataset 2, there are 13 classes with a total of 1.5 KLOC. In dataset 3, there are 10 classes with a total code length is 748 LOC.

TABLE 2.
RESULT OF SQL CLONE.

Datasets	SQL Script	Number of Clone
Dataset 1	35 SQL Script	18 clones
Dataset 2	129 SQL Script	60 clones
Dataset 3	87 SQL Script	49 clones

Table 2 shows the result of manual investigation of SQL clone in MVC-based application. In dataset 1, are found 18 SQL clones. In dataset 2 and dataset 3 are found 60 and 49 SQL clone respectively. Besides, the total number of SQL script in dataset 1 is 35 SQL script. Dataset 2 and 3 is 129 and 87 SQL script respectively.

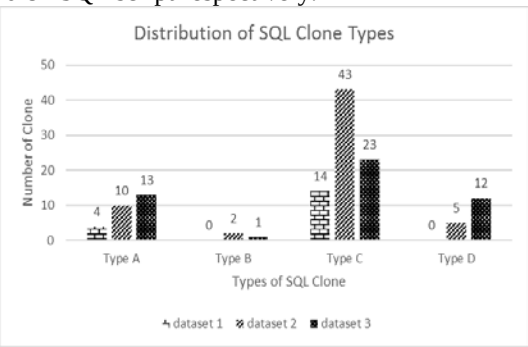

Figure 3. Distribution of SQL Clone Types

Figure 3 shows the distribution of SQL clone types. In dataset 1, SQL clone type A and C are found they consist of 4 SQL clone type A and 14 SQL clone type C. In dataset 2, all SQL clone types (A, B, C, D) are found. The distribution of SQL clone type is 10 SQL clone type A, 2 SQL clone type B, 43 SQL clone type C, and 5 SQL clone type D. In dataset 3, all SQL clone types are found. The distribution of SQL clone type is 13 SQL clone type A, 1 SQL clone type B, 23 SQL clone type C, and 12 SQL clone type D.

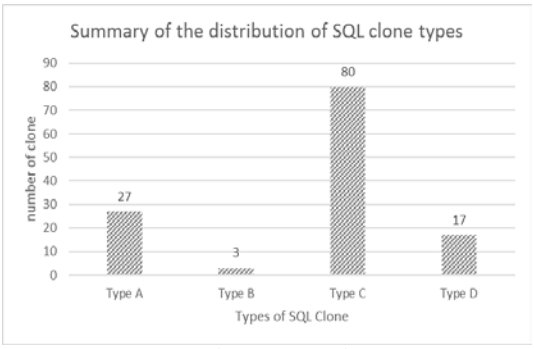

Figure 4. Summary of distribution of SQL Clone Types

Based on figure 4, SQL clone type C is SQL clone type that are mostly found in the dataset. While the SQL clone type B is less found in the dataset.

Based on the result of manual investigation, all types of SQL clone are exist on MVC-based application. It shows that the definition of SQL clone is valid.

V. CONCLUSION

Code clone definition is not suitable for SQL language because SQL is declarative language and it has different structure from common programming languages like Java, C, and PHP. This research defines SQL clone definition and its types based on two principles. The first is the definition of code clone and its types. The definition of code clone and its types need to be adjusted with SQL characteristic and structure. The second is the result of research in the SQL script similarity area. Research in SQL script similarity areas includes SQL script mutations and recommendation systems for SQL queries. In this research, four types of SQL clone are defined: Type A, B, C, and D.

Based on the result of the manual investigation, all types of SQL clone exist on MVC-based application. It shows that the definition of SQL clone is valid. Our future work use defined SQL clone and its types to detect duplicate queries in the model layer of MVC-based application.

REFERENCES

[1] A. Leff and J. T. Rayfield, "Web-application development using the Model/View/Controller design pattern," in *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, pp. 118–127.

[2] F. Buschmann, K. Henney, and D. C. Schmidt, *Pattern-oriented software architecture. v. 4, A pattern language for distributed computing*. West Sussex: John Wiley & Sons, 2007.

[3] D. Rattan, R. Bhatia, and M. Singh, "Software clone detection: A systematic review," *Inf. Softw. Technol.*, vol. 55, no. 7, pp. 1165–1199, Jul. 2013.

[4] D. C. Rajapakse and S. Jarzabek, "An investigation of cloning in web applications," in *Special interest tracks and posters of the 14th international conference on World Wide Web - WWW '05*, 2005, p. 924.

[5] A. Leff and J. T. Rayfield, "Web-Application Development Using the Model/View/Controller Design Pattern," in *Enterprise Distributed Object Computing Conference*, 2001, pp. 118–127.

[6] F. Buschmann, K. Henney, and D. C. Schmidt, *PATTERN-ORIENTED SOFTWARE ARCHITECTURE*, 4th ed. West Sussex: John Wiley & Sons, Ltd, 2007.

[7] R. Morales-Chaparro, M. Linaje, J. C. Preciado, and F. Sánchez-Figueroa, "MVC web design patterns and rich internet applications,"

in *Proceedings of the Jornadas de Ingeniería del Software y Bases de Datos*, 2007, pp. 39–46.
[8] D. Rattan, R. Bhatia, and M. Singh, *Software clone detection: A systematic review*, vol. 55, no. 7. Elsevier B.V., 2013.
[9] D. Chatterji, J. C. Carver, and N. A. Kraft, "Claims and beliefs about code clones: Do we agree as a community? A survey," in *2012 6th International Workshop on Software Clones (IWSC)*, 2012, pp. 15–21.
[10] A. Beaulieu, *Learning SQL, 2nd Edition*. O'Reilly Media, 2009.
[11] H. Störrle, "Towards clone detection in UML domain models," *Softw. Syst. Model.*, vol. 12, no. 2, pp. 307–329, May 2013.
[12] J. Tuya, M. J. Suárez-cabal, and C. De Riva, "SQLMutation : A tool to generate mutants of SQL database queries," 2006.
[13] M. Eirinaki, S. Abraham, N. Polyzotis, and N. Shaikh, "QueRIE: Collaborative Database Exploration," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1778–1790, Jul. 2014.
[14] M. Stephan, "Model Clone Detector Evaluation Using Mutation Analysis," in *2014 IEEE International Conference on Software Maintenance and Evolution*, 2014, pp. 633–638.
[15] J. Tuya, M. J. Suarez-Cabal, and C. de la Riva, "SQLMutation: A tool to generate mutants of SQL database queries," in *Second Workshop on Mutation Analysis (Mutation 2006 - ISSRE Workshops 2006)*, 2006, pp. 1–1.