Improvement of the POLISAS e-Request Transcript System using Query Optimization

Mohd Hariz bin Ibrahim¹, Suryanti Awang², Kohbalan Moorthy²

¹Kolej Komuniti Paya Besar, Kementerian Pendidikan Malaysia, BT 19, Jalan Maran, 26300 Gambang, Kuantan, Pahang, Malaysia.

²Soft Computing & Intelligent Systems Research Group (SPINT), Faculty of Computing, Universiti Malaysia Pahang, 26300, Kuantan, Pahang, Malaysia

hariz@kkpbe.edu.my

Abstract. This paper is about the improvement of the E-Request Transcript system used by POLISAS to be a systematic and effective system. The improvement is implemented to solve the current problem, whereby, students need to access the E-Request Transcript system according to the graduation session in individual file. This situation makes it difficult for the administrators to access more than one E-request transcript system if the application is from different sessions. It is because of the data-access issues to generate reports. To achieve the system improvement, an integrated database using the query optimization technique is implemented. Next, the query statement is analysed to produce an optimum performance. Research results show that the technique is able to improve the query time processing, thus resolving the issues discussed. A user acceptance test for this system consists of a black box test and a beta test has been deployed. A black box test was conducted to test the main menu functions in the E-Request transcript and is found to be able to succeed. Furthermore, the beta test is carried out using a survey form distributed to all students who have graduated. Survey questionnaire showed that 57% of respondents were satisfied using this system and 14% were very satisfied with this system. All respondents have provided a good scaling rate for all items assessed. For overall system assessment, the mean score obtained is 3.78 which is interpreted as a good rate based on the scale of interpretation used. In conclusion, this system has met the needs of consumers as well as simplifying their business based on surveys conducted.

Keywords - Integrated; Transcript; Optimization Technique; Query Optimization

1. Introduction

Information systems (IS) involved the process of collection, creation and distribution of useful data in organizational settings that are built based on the integration hardware, software and telecommunications networks. The interrelated components of IS works together to obtain and distribute information. It also processes, stores and distributes the information to help an organization to coordinate, analyze, control and visualize processes of an organization. This also contributes to effective decision making of an organization. There is a clear difference between IS and business processes, which is, IS a system that assists in controlling business processes.

Integrated Information Systems (IIS) consists of several aspects of Management Information Systems that can be integrated. These aspects can be integrated separately or in a combination. Some
choices need be made by the organization between developing a new IS or improving an existing one. Thus, the IIS implementation can be a tedious task.

Various researchers have conducted IS assessment in the past. Assessed IS Integration management business Continuity [1]. [2] Assessed the modelling in IS integration, and [3] assessed the elements and functions integrable in integrated management systems. These efforts have contributed to better understanding of assessment issues for IIS, but fall short of looking at the IIS assessment in a broad perspective. Thus far, the assessment is focused more on the technical aspect or on the product itself, but has not covered all domains in IS integration [4].

E-request transcript (ERT) is an application of IS that is currently being used by graduates of Politeknik Sultan Haji Ahmad Shah (POLISAS) and staff at the Examination Unit in POLISAS [5]. The application involves the process of requesting forms and verification letters related to graduation. The graduating student is considered as a client who uses this system and the administrator is required to handle the process of managing the system. An innovation project called E-Request Transcripts (ERT) was produced to replace the manual application form. By using ERT, students are able to initiate online application and student information data can be updated quickly [5]. With ERT, managing production academic transcript and graduate verification letter becomes faster and systematic.

Effective ERT implementation is dependent on an understanding of the goals and benefits of system integration required. The integration is regarded as a progressive effort towards a better handling of information. Other efforts that lead to a better integration are the linking data between different session, integration of relevant data in a central database or platform, the integration of information on graduate and the opportunity to integrate data in personal level. People do not always benefit from the abundance of natural resources, if the man does not know to manage it [6]. From the database resources, ERT needs to serve as the best integration technique for solved the management issue.

2. Literature Review

2.1. Case study (E-Request Transcript System (ERT))
In total, the POLISAS Examination Unit is responsible for the management of data transcripts application at POLISAS. Among the key functions of the organization are [5]:

i. Collect and process academic transcripts application data from students before preparing the academic transcripts to students.
ii. Ensure data transcripts of academic transcripts and data records can be tracked quickly.
iii. Providing systematic services online.

Saved transcript data includes student-ending data, data application, pending application data, rejection data and processed data. To ensure that the above functions are well implemented, the data needs to be stored and maintained by a database system. Therefore, a database system called the E-Request Transcript System (ERT) has been developed for the management and maintenance of such data.

2.2 E-Request Transcript System (ERT)
An innovative project named E-Request Transcript was previously created to replace the use of the manual application forms. Students only need to apply online and student information data can be updated quickly. With E-Request Transcript, the issuance of Completed Letter of Accreditation and Academic Transcripts becomes faster and systematic. The E-Request Transcript system has been adopted by the examination unit since 2013 has greatly assisted the management of the examination unit, POLISAS [5].

2.3 Query processing
Structured Query Language or SQL is an interactive programming language used to retrieve information from databases and update data in databases. It provides a natural language syntax for constructing query statements, defining and manipulating data in databases. Questions are in the form of instruction languages that provide functions such as selection, entry, updating and data access and are more commonly known as SQL query statements [14] [15] [16].

The ultimate goal of SQL statement processing is to transform queries written in high-level languages to more accurate and efficient implementation strategies that are expressed in lower-level languages and implement those strategies in re-attaining the required data [17]. An important aspect of query processing is query optimization. There are various equivalent conversion strategies for every high-level query. Therefore, the purpose of query optimization is to choose one of the strategies that can minimize the use of resources [18]. Typically query optimization is implemented to reduce the overall execution time of the query, ie the total execution time for all separate operations that perform the query [18]. However, resource utilization can also be viewed as the response time of the query. In a research [18], they focus on maximizing the number of parallel operations in reducing the reaction time of query execution.

Query processing can be divided into the following four phases: decomposition (comprising decomposition and validation), optimization, code generation and implementation [19]. The next section briefly explains the first phase of decomposition, before focusing more on the second phase of query optimization.

Query optimization is a function of many relational database management systems in which multiple query plans for satisfying a query are examined. The query at each level of data integration and mapping components to the existing or a new scheme could influence the existing, and new attributes [20]. Here, a good query plan is identified. This may or may not be the absolute best strategy because there are many ways of doing plans. There is a trade-off between the amount of time spent figuring out the best plan and the amount of time used in running the plan. Different qualities of database management systems have different ways of balancing between these two [21]. Query optimization requires a lengthy query execution time. Therefore, it is necessary to find new SQL technologies used to reduce a query execution time during the optimization, according to the basic operations of query such as scanning, aggregation, join algorithms [22]. This challenge can be addressed by reducing the number of queries by using strings, joins, aggregation and grouping all the relational [20].

2.4 Integration operation

Basically, the Integration operation is closely related to the product of the cartesinan. The cartesisan product of the two relations was encapsulate each tuple in the first relationship with each tuple in the second relationship and the combined result was stored in a new relationship [11]. For example, the results for the cartesinan product operations on the relations of X and Y, with x and y the number of tuples, are the relations (x X y) of the tuple. Consider the following example relations. The cartesinan product of X relation (Table 2.2) and Y (Table 2.3) is the relation of Z [11]. Relation X (Table 2.2) and Y (Table 2.3) is the Relation of Z (Table 2.4).

| StudID | ProgCode |
|-------|----------|
| 7891  | 2        |
| 6033  | 3        |
| 1332  | 2        |

| StudID | ProgCode |
|-------|----------|
| 1     | 1000     |
| 2     | 5000     |
| 3     | 10000    |

| StudID | ProgCode | ProgCode | SumProg |
|--------|----------|----------|---------|
| 7891   | 2        | 1        | 1000    |
| 7891   | 2        | 2        | 5000    |
| 7891   | 2        | 3        | 10000   |
Based on the concept of relationship, X and Y relationships are written as

\[ X \otimes_{x(\alpha) \otimes y(\beta)} Y \]

where \( X \otimes_{x(\alpha) \otimes y(\beta)} \) defines the combined conditions. The operator, \( \otimes \), represents the operator for the combined condition where the value of the operator must always be true between the attributes \( x(\alpha) \) and \( y(\beta) \) of the X and Y relationships. The join conditions usually consist of simple conditions and are connected to a logical operator [7] [8].

In general, it is arguably the combination is equivalent to the product of the cartels, where the combined result of the two relationships is a subset of cartels productions of two relationships [10].

\[ X \cap_{x(\alpha) \cap y(\beta)} Y \equiv X \times_{x(\alpha)} Y \times_{y(\beta)} \]

The combined result of X and Y relationships with attribute x and y, is the relation Z with \( x + y \) attributes. The Z relationship has one tuple for each tuple pair of X and Y that meet the combined requirements. The result of Z relationship can be defined as follows:

\[ Z = \{ z | z = xy \land x \in X \land y \in Y \land x(\alpha) \otimes y(\beta) \} \]

This relationship is a subset of the product X and Y.

2.4.1 Equijoin, In line with the optimization objective of minimizing the timing of data response and query execution, the implementation of this combined method can be implemented to reduce access to the relationship and ultimately achieve the ultimate goal of optimization techniques [9].

The combined commonality between \( R_1 \) and \( R_2 \), can be defined as follows. For each couple tuple \( t_{r_1} \) and \( t_{r_2} \), where \( t_{r_1} \in R_1, t_{r_2} \in R_2 \) if \( t_{r_1}[\alpha] = t_{r_2}[\beta] \), then tuple \( t_{r_1} \) and \( t_{r_2} \) was merged into one tuple. The overlap between the combined tuples was deleted. The equation for the combination of equality can be written as follows:

\[ R_1 \cap_{r_1(\alpha) = r_2(\beta)} R_2 \]

\[ R' = \{ t | t = r_{1r_2} \land r_{1r_2} \in R_1 \cap r_{1r_2} \in R_2 \land t_{r_1}[\alpha] = t_{r_2}[\beta] \} \]

\[ Z = X \cap_{x(ProgCode) \cap y(ProgCode)} Y \]

Table 2.5: Equijoin result

| No   | ProgCode | ProgCode |
|------|----------|----------|
| 7891 | 2        | 5000     |
| 3011 | 3        | 10000    |
| 1224 | 2        | 5000     |
| 7055 | 3        | 10000    |

where \( R' \) is the new relationship for \( R_1 \) and \( R_2 \) combined. In this combination the operators used are equality operators (=). Consider Table 2.2 and Table 2.3, the combined result of X and Y where \( x(ProgCode) = y(ProgCode) \) is the relationship Z as follows on Table 2.5 [11]:
3. Methodology
Although many methods have been proposed by integrated relationships within the database, they still have weaknesses in handling query statements that involve data access in large numbers of records [12] [13]. Therefore, this research produces an optimization technique to correct the problems discussed. The technique is known as the Logical Scheme Optimization of the Relationship Database. It is based on the equijoin method introduced by Priti and Margaret [10].

For the implementation of the technique, only a few entities in the ERT database are involved, such as tblpelajar@transkripdis2014, tblpelajar@transkripjun2015, tblpelajar@transkripdis2015, tblpelajar@transkripjun2016, tblpelajar@transkripdis2016. Entities that stores information related data POLISAS students. The following sections describe the attributes and data stored in each entity.

3.1 Entities tblpelajar
Basically, the primary key for the entity tblpelajar@transkripdis2015 is the same as the primary key for the entity tblpelajar@transkripdis2014, tblpelajar@transkripjun2015 tbltranskripjun2016 and tbltranskripdis2016. Because of these factors, these entities joined and create a new entity. The purpose of the merger is to minimize data overlap and reduce the number of entities, thereby reducing the reaction time of a given query. The joint of these entities described in the following sections.

Figure 3.1 describes the implementation of the Integration Table Optimization techniques on the E-Request Transcript System database. As mentioned earlier, only a few entities are involved in the implementation of tblpelajar@transkripdis2014, tblpelajar@transkripjun2015, tblpelajar@transkripdis2015, tblpelajar@transkripjun2016 and tblpelajar@transkripdis2016 represented tblpelajar_A and tblpelajar_B. With the implementation of the equijoin Optimization, all entities are combined and the attributes used for comparison are Nopend, Nric, Name, prog. Tuple values for learners are combined across sessions with tuple learners who are different sessions if the values for the attributes are equivalent. Next, new entity (tblpelajar_C) was classified into multiple entities based on SESI attributes.

Fig 3.1: Implementation Optimization Technique of Logical Schema Optimization Relational Database to ERT.

In this implementation, the merged entity is tblpelajar for each session represented tblpelajar_A entities and entity is tblpelajar_B, while the main key for both attributes are used as terms of a combination of Nopend, Nric, Name and Prog. These attributes are combined and stored in a new
entity namely tblpelajar_C (Figure 3.1) and created a new attribute of SESI. Whereas, the value of each attribute is checked, if the value is equal, then, it combined and stored in the new entity.

As explained, the goal of the research is to improve the time of data access in the database. With the implementation of this process, the number of tuples in the referenced entity can be reduced. Additionally, it can also minimize the number of referred entities. With this, the goal of research can be achieved and the problem of investigation can be improved.

4. Result and Discussion

Table 4.1 shows the number of tuples for each entity before and after the optimization technique is implemented. If observed in the table, the number of tuples before the implementation of optimization techniques is greater than after the optimization technique is implemented. The results of the combined results can be explained based on the following formula:

$$R \cdot \min(R_1, R_2)$$

where the number of tuples in the joined result entity (R3) must be greater or equal to the minimum tuple number between R1 or R2. R1 and R2 are the entities to be joined. Data loss and integrity issues occur if the joined results are not eligible for the formula.

| Entity (Rj) | ERT before (Without Optimization) | ERT after (With Optimization) |
|-------------|----------------------------------|-------------------------------|
|             | Number of Tuple (n)             | Number of Tuple (n)           |
| Entity R1 (tblpelajar_A) | 3500 | -         |
| Entity R2 (tblpelajar_B) | 36900 | -        |
| Entity R3 (tblpelajar_C) | -   | 3650     |

4.1 Query Optimization Comparison Testing

This implementation only uses ORACLE query optimizer and does not change any of the basic rules in selecting the most optimal way of implementation. For each query statement, there are several possible executions that can be produced. The execution time for each possible execution are used as a criterion for determining and selecting the optimum implementation possibilities. In this paper, ORACLE SQL Analyze is used to test the execution time for each possible execution. Based on the information obtained, the most optimal implementation possibilities can be determined and selected for use as a result of obtaining the desired results.

Output for P1, P2, P3 and P4 statements is the number of requests required. To get the result, SUM aggregate function is used in each query statement. Each tuple on the affected entity such as tblmohondetail was be reviewed. If the session attribute value meets the specified predicate condition, then the value of the application was summed. The predicate conditions for each statement P1, P2, P3 and P4 vary, depending on the time interval of the desired output. For P1 statements, the desired output is the number of transcripts for a year, while P2, and P4 are the total application counts for 2 years, 3 years and 4 years respectively. For the GROUP BY statement, the compilation process is performed to get the result of the SUM function. The next section discussed comparative results in selecting and determining the likelihood of execution that results in the most optimal time.

4.1.1 Query P1 (Number of transcripts application per year)

Table 4.2 and 4.3 describes the average execution time for each plan ie plan 1, plan 2 and plan 3. As explained, the most optimal execution possibility is based on the average time taken for each execution.
Based on the comparison graph in Figure 4.6, plan 3 produces the optimal time for P1 expression. Therefore, there is a possibility of this execution to be selected for execution in producing the required reports i.e. a transcript application report for a year.

Table 4.2 Query Time for P1 1s – 6s

|       | 1(s)  | 2(s)  | 3(s)  | 4(s)  | 5(s)  | 6(s)  |
|-------|-------|-------|-------|-------|-------|-------|
| Plan 1| 0.12  | 0.12  | 0.07  | 0.09  | 0.10  | 0.11  |
| Plan 2| 0.07  | 0.09  | 0.06  | 0.06  | 0.06  | 0.08  |
| Plan 3| 0.04  | 0.04  | 0.06  | 0.06  | 0.06  | 0.06  |
| Plan 4| 0.04  | 0.04  | 0.06  | 0.05  | 0.06  | 0.06  |

Table 4.3 Query Time for P1 7s-10s

|       | 7(s)  | 8(s)  | 9(s)  | 10(s) | Average (s) |
|-------|-------|-------|-------|-------|-------------|
| Plan 1| 0.09  | 0.06  | 0.07  | 0.08  | 0.091       |
| Plan 2| 0.04  | 0.06  | 0.06  | 0.04  | 0.062       |
| Plan 3| 0.06  | 0.04  | 0.06  | 0.04  | 0.052       |
| Plan 4| 0.06  | 0.06  | 0.07  | 0.06  | 0.056       |

Figure 4.1 Comparison Graph for Query P1

4.1.2 Query P2 (Number of transcripts requested for 2 years)
This statement is for the re-access of transcript application data for an interval of 2 years. The average execution time for each possible execution is as in Table 4.4. Based on the average time obtained, plan 4 produces the fastest reaction time compared to plans 1, 2, 3 and 4. Therefore, the probability of execution was selected in the implementation of data retrieval. Figure 4.2 illustrates the comparison of the execution time of each plan.

Table 4.4 Query Time for P2 1s – 6s

|       | 1(s)  | 2(s)  | 3(s)  | 4(s)  | 5(s)  | 6(s)  |
|-------|-------|-------|-------|-------|-------|-------|
| Plan 1| 0.17  | 0.15  | 0.14  | 0.12  | 0.15  | 0.15  |
| Plan 2| 0.15  | 0.14  | 0.14  | 0.14  | 0.15  | 0.16  |
| Plan 3| 0.14  | 0.14  | 0.12  | 0.14  | 0.15  | 0.14  |
| Plan 4| 0.14  | 0.12  | 0.12  | 0.14  | 0.10  | 0.12  |
Table 4.5 Query Time for P2 7s-10s

|         | 7(s) | 8(s) | 9(s) | 10(s) | Average (s) |
|---------|------|------|------|-------|-------------|
| Plan 1  | 0.15 | 0.12 | 0.12 | 0.15  | 0.142       |
| Plan 2  | 0.14 | 0.14 | 0.14 | 0.10  | 0.140       |
| Plan 3  | 0.12 | 0.14 | 0.12 | 0.12  | 0.133       |
| Plan 4  | 0.12 | 0.12 | 0.10 | 0.09  | 0.117       |

Figure 4.2 Comparison Graph for Query P2

4.1.3 Query P3 (Number of transcripts requested for 3 years)

Table 4.6 and 4.7 illustrates the average execution time for each possible execution of P3 query, while Figure 4.3 is a comparison graph of the execution time of each plan. Based on the graph, plan 3 produces the optimum execution time. Therefore, the plan is selected for the data recovery process for the P3 query statement.

Table 4.6 Query Time for P2 1s – 6s

|        | 1(s) | 2(s) | 3(s) | 4(s) | 5(s) | 6(s) |
|--------|------|------|------|------|------|------|
| Plan 1 | 0.25 | 0.20 | 0.23 | 0.22 | 0.22 | 0.22 |
| Plan 2 | 0.21 | 0.17 | 0.18 | 0.18 | 0.20 | 0.21 |
| Plan 3 | 0.21 | 0.15 | 0.15 | 0.15 | 0.17 | 0.12 |
| Plan 4 | 0.17 | 0.19 | 0.17 | 0.18 | 0.17 | 0.17 |

Table 4.7 Query Time for P2 7s-10s

|       | 7(s) | 8(s) | 9(s) | 10(s) | Average (s) |
|-------|------|------|------|-------|-------------|
| Plan 1| 0.18 | 0.20 | 0.18 | 0.18  | 0.208       |
| Plan 2| 0.17 | 0.21 | 0.18 | 0.18  | 0.189       |
| Plan 3| 0.15 | 0.14 | 0.17 | 0.15  | 0.156       |
| Plan 4| 0.17 | 0.18 | 0.19 | 0.15  | 0.174       |
4.1.3 Query P4 (Number of transcripts requested for 4 years)

The average execution time for each possible execution of P4 query is described in Table 4.6. The results show that plan 3 produces the fastest execution time and is illustrated in the comparison graph of the implementation time in Figure 4.9 Based on the decision 3 plan selected to implement this question.

Table 4.8 Query Time for P2 1s – 6s

|       | 1(s) | 2(s) | 3(s) | 4(s) | 5(s) | 6(s) |
|-------|------|------|------|------|------|------|
| Plan 1| 0.33 | 0.25 | 0.29 | 0.28 | 0.26 | 0.33 |
| Plan 2| 0.26 | 0.23 | 0.24 | 0.23 | 0.26 | 0.26 |
| Plan 3| 0.22 | 0.20 | 0.26 | 0.23 | 0.23 | 0.22 |
| Plan 4| 0.21 | 0.21 | 0.25 | 0.23 | 0.26 | 0.21 |

Table 4.9 Query Time for P2 7s-10s

|       | 7(s) | 8(s) | 9(s) | 10(s) | Average(s) |
|-------|------|------|------|-------|------------|
| Plan 1| 0.24 | 0.20 | 0.25 | 0.21  | 0.21       |
| Plan 2| 0.25 | 0.23 | 0.25 | 0.26  | 0.28       |
| Plan 3| 0.23 | 0.23 | 0.23 | 0.23  | 0.21       |
| Plan 4| 0.23 | 0.21 | 0.21 | 0.23  | 0.23       |

Figure 4.9 Comparison Graph for Query for P4
4.1.4 Comparison Execution Time of Plan

Figure 4.10 shows the execution time of the plan for each query statement for the number of transcript applications. Question P1 represents the number of transcripts for a year, P2 inquiry represents the number of transcripts for 2 years, the P3 question represents the total number of applications for 3 years, while the P4 query represents the total number of applications for 4 years. Plan 3 results show the most optimal execution time of P1, P3 and P4 queries. While Plans 4 produced the optimum implementation time for P2 inquiries.

![Figure 4.10 Comparative Graph Execution Time For Statement Inquiries](image)

The following table 4.7 describes the comparison of execution times before and after optimization. The execution time before optimization is obtained based on the average time in the table above. While the execution time after optimization is obtained based on the probable execution testing discussed in the previous section. Each of the implementation possibilities selected for P1, P2, P3 and P4 statements was compared to the execution time before optimization.

|                | P1  | P2  | P3  | P4  |
|----------------|-----|-----|-----|-----|
| Before Optimization (s) | 0.782 | 0.955 | 1.415 | 1.768 |
| After Optimization (s)   | 0.052 | 0.117 | 0.156 | 0.225 |
| Percent Enhancement (%)  | 93.4 | 87.7 | 89.0 | 87.3 |

For P1 statements, the execution time after optimization is faster than before that of 0.052 seconds. In addition, implementation time after optimization of P2, P3 and P4 statements is also faster before optimization (Refer Table 4.8). Comparative results obtained are illustrated using the graph in the following figure 4.11. Based on the graph shows that there is a percentage increase of about 89%. From the results it shows that the technique produced is more effective to fix the problems discussed.

Based on the results obtained, there is a percentage increase in response time for each query statement. Some timetable execution time shows that the response time after optimization is better than before optimization. From the results it shows that the optimization technique of the resulting table can speed up the next reaction time to improve the problem discussed.
Black Box Testing is used to test the specific functions of the designed software. In order to test the e-request transcript system, black box testing is used to test the system function and beta test is used to find out the users’ response to the e-request transcript system. This is conducted through data obtained by using questionnaire as a research instrument.

Beta testing is an objective test that is tested directly to the field, using a questionnaire in the effort to gather the user's response to the system that is built. The questionnaires were distributed using the Simple Random Sampling technique in which sample members of the population were randomly assigned without regard to the strata present in the population. The data was collected from 20 system users consisting of POLISAS students and 5 management users consisting of staff of the examination unit. They are given a brief explanation of this test method. First of all, they need to give their first opinion of the site. Then they were given ample time to crawl the site to get an overview of how the site operated. They were also given about 20 minutes to perform an evaluation process. Upon completion of the completed site testing, they were asked to fill out a questionnaire to answer some questions based on their experience on the site. Based on table 4.12, the questionnaire consisted of 6 main aspect questions with scoring system using Likert scale measurement technique. Listed below are the indicators and description of each indicator included in the Likert Scale.

| EXPLANATION     | LIKERT SCORE |
|-----------------|--------------|
| Very Dissatisfied| 1            |
| Dissatisfied     | 2            |
| Neutral          | 3            |
| Satisfied        | 4            |
| Very Satisfied   | 5            |

As a result of the testing, the average user satisfaction on the ERT System is within the range of satisfactory level. Table 4.13 shows the user rating on the ERT System.
5. Conclusion

The objectives of the study outlined in this paper are as follows:

i. To study and analyse the weakness on databases of the existing system.
ii. To design and develop new databases and function on an E-request transcript application.
iii. To validate and verify the proposed solution based on query optimization technique.

The extent to which the above objectives are achieved are assessed based on the results that are obtained throughout the development of the project. It follows current database analysis in case studies of ERT system discussed to meet the first objective of the study. The E-Request Transcript application was developed by upgrading the database, using the concept of logical schema optimization relation database for equijoin method to meet the second objective of the study. From the results of the execution query time, the optimization technique was appropriate with the study frame to fulfil the third objective of the study.

In general, the resulting optimization techniques involve the restructuring of relationships and tables in the database. Therefore, this technique is suitable for use by any relational database with less normal and irregular schedule structure. Key points for each involved table need to be determined to avoid data loss after optimization is implemented.

In determining the effectiveness of the enhanced system, black box testing is used to test the system function and beta test is used to find out the user's response to the e-request transcript system, by doing the questionnaire. The test involved integration testing, system testing and user acceptance testing that have been enhanced successfully. The result of the functional testing of the menu that was set up in blackbox testing found that the menus functioned. Result of user acceptance test shows that overall user satisfaction in using the E-Request Transcript system and security aspects of the system are satisfying.

A user acceptance test for this system consists of a black box test and a beta test has been deployed. A black box test was conducted to test the main menu functions in the E-Request transcript and is found to be able to succeed. Furthermore, the beta test is carried out using a survey form distributed to all students who have graduated. Survey questionnaire showed that 57% of respondents were satisfied using this system and 14% were very satisfied with this system. All respondents have provided a good scaling rate for all items assessed. For overall system assessment, the mean score obtained is 3.78 which is
interpreted as a good rate based on the scale of interpretation used. In conclusion, this system has met the needs of consumers as well as simplifying their business based on surveys conducted. In conclusion, this system has met the needs of consumers as well as simplifying their business based on surveys conducted.

6. Acknowledgments

We would like to thank Universiti Malaysia Pahang for supporting this work under the RDU190315.

References

[1] An, X., Lemen, C., & Liuqi, Y. (2010). An Integration Management System for Managing Business Continuity. Records and Knowledge. Information Technology and Applications (IFITA), 2010 International Forum on, 2(April), 4–5. https://doi.org/10.1109/IFITA.2010.

[2] Igamberdiev, M., Grossmann, G., Selway, M., & Stumptner, M. (2016). An integrated multi-level modeling approach for industrial-scale data interoperability. Software and Systems Modeling, 1–26. https://doi.org/10.1007/s10270-016-0520-6

[3] Nunhes, T. V., Motta Barbosa, L. C. F., & de Oliveira, O. J. (2016). Identification and analysis of the elements and functions integrable in integrated management systems. Journal of Cleaner Production, 142. https://doi.org/10.1016/j.jclepro.2016.10.147.

[4] Ruzaini, A. A. (2013). Assessment Instrument For Integrated Information System Success In Institution Of Higher Learning. University Teknology Malaysia.

[5] Mohd Hariz, I. (2014). E-MOHON TRANSKRIP. In Icompex14 National Innovation and Invention Competition trough Exhibition 2014 (p. 2). Politeknik Sultan Abdul Halim Mu’adzam Shah.

[6] Solihadin, O., & Bandung, Y. (2016). A Case Study on Integration of Conventional Learning System and ICT-based Learning for Job Training System. 2016 International Conference on Information Technology Systems and Innovation (ICITSI) Bandung – Bali, October 24 – 27, 2016, (1).

[7] Alexandrov, A., Bergmann, R., Ewen, S. et al (2014). The Stratosphere platform for big data analytics. The VLDB Journal (2014) 23: 939. https://doi.org/10.1007/s00778-014-0357-y.

[8] Khayyat,Z., Ilyas,1.F., Jindal, A., (2015) BigDansing: A System for Big Data Cleansing. Proceeding SIGMOD ’15 Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data Pages 1215-1230.

[9] Breß S., Heimel M., Siegmund N., Bellatreche L., Saake G. (2014) GPU-Accelerated Database Systems: Survey and Open Challenges. In: Hameurlain A. et al. (eds) Transactions on Large-Scale Data- and Knowledge-Centered Systems XV. Lecture Notes in Computer Science, vol 8920. Springer, Berlin, Heidelberg.

[10] Priti M and Margaret H.E (1992). Join Processing in Relational Databases. ACM Computing Surveys, 24(1):63-113.

[11] Norrazrin K (2006). Optimization Of Logical Scope Of Coordination Data: Hydrological Information System Malaysia. Universiti Teknologi Malaysia : Master Thesis.

[12] C, Coronel., S, Morris (2016). DATABASE SYSTEMS, Design, Implementation and Management (12nd ed.) [Online]. Available: https://www.academia.edu/.

[13] A, Dennis., B.H. Wixom., D. Tegerdan (2015, Feb). System Analysis & Design, An Object-Oriented Approach with UML (5nd ed.) [Online]. Available: https://www.academia.edu/.

[14] Campagna, G., Ramesh, R., Xu, S., Fischer, M., & Lam, M. S. (2017). Almond: The Architecture of an Open, Crowdsourced, Privacy-Preserving, Programmable Virtual Assistant. Www, 341–350. https://doi.org/10.1145/3038912.3052562
[15] Abiteboul, S., Bourhis, P., & Vianu, V. (2015). Highly Expressive Query Languages for Unordered Data Trees. Theory of Computing Systems, 57(4), 927–966. https://doi.org/10.1007/s00224-015-9617-5

[16] Torlak, E., & Bodik, R. (2013). A lightweight symbolic virtual machine for solver-aided host languages. Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI ’14, 530–541. https://doi.org/10.1145/2594291.2594340

[17] Chilana, P. K., Singh, R., & Guo, P. J. (2016). Understanding Conversational Programmers: A Perspective from the Software Industry. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI ’16, 1462–1472. https://doi.org/10.1145/2858036.2858323

[18] Nguyen, T. H., Yadav, A., An, B., Tambe, M., & Boutilier, C. (2015). Regret-based Optimization and Preference Elicitation for Stackelberg Security Games with Uncertainty. Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), 756–762.

[19] Steuwer, M., Fensch, C., Lindley, S., & Dubach, C. (2015). Generating Performance Portable Code using Rewrite Rules: From High-Level Functional Expressions to High-Performance OpenCL Code. ICFP: ACM SIGPLAN International Conference on Functional Programming, 205–217. https://doi.org/10.1145/2784731.2784754

[20] Kadadi, A., Agrawal, R., Nyamful, C., & Atiq, R. (2015). Challenges of data integration and interoperability in big data. Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014, 38–40. https://doi.org/10.1109/BigData.2014.7004486

[21] Wafaa, A. S., Tiun, S., Ahmed, A. S., Awang, S., & Al-Khaleefa, A. S. (2018). Word sense disambiguation using hybrid swarm intelligence approach. PloS one, 13(12), e0208695.

[22] Al-Saffar, A., Awang, S., Tao, H., Omar, N., Al-Saïagh, W., & Al-bared, M. (2018). Malay sentiment analysis based on combined classification approaches and Senti-lexicon algorithm. PloS one, 13(4), e0194852.