OPTIMIZATION OF EFFICIENCY IN ABAP APPLICATION PROGRAM BASED ON INTERNAL TABLE AND MODULARIZATION TECHNIQUES

Sriinidi[1] VS & Sai Rajesh[2] VB & Shalini[3] S & Giridharan[4] N

[1] *Student, Department of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tamilnadu, India.*

[2]*Student, Department of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tamilnadu, India.*

[3]*Student, Department of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tamilnadu, India.*

[4] *Assistant Professor, Department of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tamilnadu, India.*

ABSTRACT: SAP SE (Systems, Applications and Products in Data Processing) is a German-based software corporation which is one of the largest vendors of Enterprise Resource Planning (ERP) software worldwide. SAP is a software system which consists of number of fully integrated modules, which covers every aspect of business management. ABAP (Advanced Business Application Programming) is a programming language used in the SAP system. ABAP programming language plays an important role in the implementation process of the ERP (enterprise resource planning system). It runs in the SAP ABAP runtime environment created and used by SAP for development of application programs. The pros and cons of ABAP programs will directly affect the working and efficiency of ERP applications designed using SAP ABAP. The SQL statements used in the designing of applications has to be implemented in an optimized way. The internal tables in the ABAP program plays an important role in increasing the efficiency of an ABAP application. There are different types of internal tables which can be used for different purposes. They are classified as standard, hashed, sorted internal tables. Using internal tables in ABAP applications leads to efficient retrieval of data from the database. Modularization is another concept in ABAP which can be used for optimizing the programs which in turn results in more efficient data retrieval. It also helps in easy maintenance of programs. Different modularization techniques include subroutines, functional modules and macros. The proposed system provides an improved algorithm for optimizing the SAP ABAP applications and increasing the efficiency of data retrieval in any ABAP application by using SQL statements in an optimized way, using different types of internal tables and mainly by using modularization techniques. Using modularization techniques makes the application more time efficient and improves its performance.

Keywords: SAP (Systems, Applications, Products), SQL statements, Internal tables, Efficiency, ABAP (Advanced Business Application Programming), Modularization, Subroutines, Optimized.

I. INTRODUCTION

SAP is one of the world leaders in enterprise applications in terms of software and software-related service revenue. It is the world's third largest independent software manufacturer supporting all sizes of industries helping them to run profitably, grow constantly and stay ahead of the competition in the market, based on market capitalization. It is known worldwide for its peculiar innovations that help the customers run their business with higher efficiency.

The SAP software business suite has been constantly extended, optimized, and tailored with respect to the needs of the market. SAP have been developing many industry-specific applications over the years and added more products for small-to-midsize companies to the software portfolio. It also enables integration between various applications from different vendors and across company boundaries. A software that is built for organizations belonging to different industrial sectors, regardless of their size and strength Enterprise Resource Planning (ERP). SAP ERP is such a software that was developed by SAP SE. Its latest version was released in 2006. In 2016, the enhanced version of SAP ERP was released.

SAP offers a wide range of products for all sizes of companies . It has scalable products, ensuring that they can be adjusted to any size and adaptable to the continuously changing processes of a company. SAP Business One is a comprehensive, integrated ERP application with an interface similar to Windows, which has been developed specifically for small and midsize companies. It distinguishes itself by simple navigation and expansion options, innovative Drag & Relate functions as well as integration with Microsoft Word and Excel. The software enables you to uniformly manage the most important business areas such as sales, distribution and finances with one single, integrated application.SAP Business By Design is based on an entirely newly developed, designed application platform. Business applications have been programmed a new from scratch based on a model. It is SAPs latest solution for small and midsize companies between100 and 500, who want to use an on demand solution to improve their core processes and at the same time benefit from lower entry and operating costs. SAP All-in-One is the brand name for vertical(industry-specific or country-specific) applications that are based on the technology of SAP NetWeaver and were developed in conjunction with SAPs partners. SAP Business All-in-One is the ideal solution for small and mid size companies with very industry-specific requirements who have several divisions and a mature IT infrastructure. For mapping business processes or business applications SAP systems are used. These applications have to be implemented independent of the hardware environment used (operating system, database) to the greatest extent possible. To do so, SAP NetWeaver provides two different runtime environments: an ABAP runtime environment (usage type AS ABAP) and a Java runtime environment. ABAP (Advanced Business Application Programming) is a programming language that was developed by SAP for developing ERP applications. Many business applications of an SAP system are being written using ABAP. ABAP is being optimized continuously for developing highly-scalable business applications. ABAP Workbench can be used for developing completely new applications as well as enhancing and modifying SAP standard applications. While a large application involving a huge amount of data is developed, the time and speed efficiency of the application has to be considered. The applications build using ABAP are generally more efficient when compared to any other applications. The existing and proposed system both discusses about increasing the efficiency of any ABAP application. While the existing system uses internal tables to do so, the proposed system concentrates on using both internal tables and modularization techniques.

EXISTING SYSTEM

ABAP is not only a development tool, and is very mature and stable platform for the comprehensive development .It is fully integrated in the R/3 system, a complete set of client/server development environment, including design and development programs, screen, menu, function module with all the necessary tools, and program debugging, performance and on-line analysis, utilities required for the test. Customized development is divided into four categories: enhancements, reports, forms and interfaces. Most of the ABAP reports, ABAP forms, ABAP interfaces, ABAP procedures, mainly for business data extraction, processing, output, etc., the data in the database according to certain business logic to take out, to a certain treatment to the report, form and other forms of display, or transmitted through the interface to other systems. In case of large amounts of data, ABAP program development will directly affect the operation of the system performance, the ABAP program in question could collapse the entire server. Therefore, the optimization of ABAP programs played a key role. In the ABAP program optimization process, the inner table is an effective way to improve efficiency. Frequently take the number of database operations, not only increases the network load, but also to improve the system load; it may drag down the whole server. The operation of SAP within the table can be removed after the data into memory, then the memory space in the open again for recycling, thereby reducing the applications interact with the database number, which can effectively reduce the system load, reduce the search data time. This type of table from the inside, to explore ways of reading research, and use large amounts of data for testing, made a number of tables within the ABAP programs based on optimization.

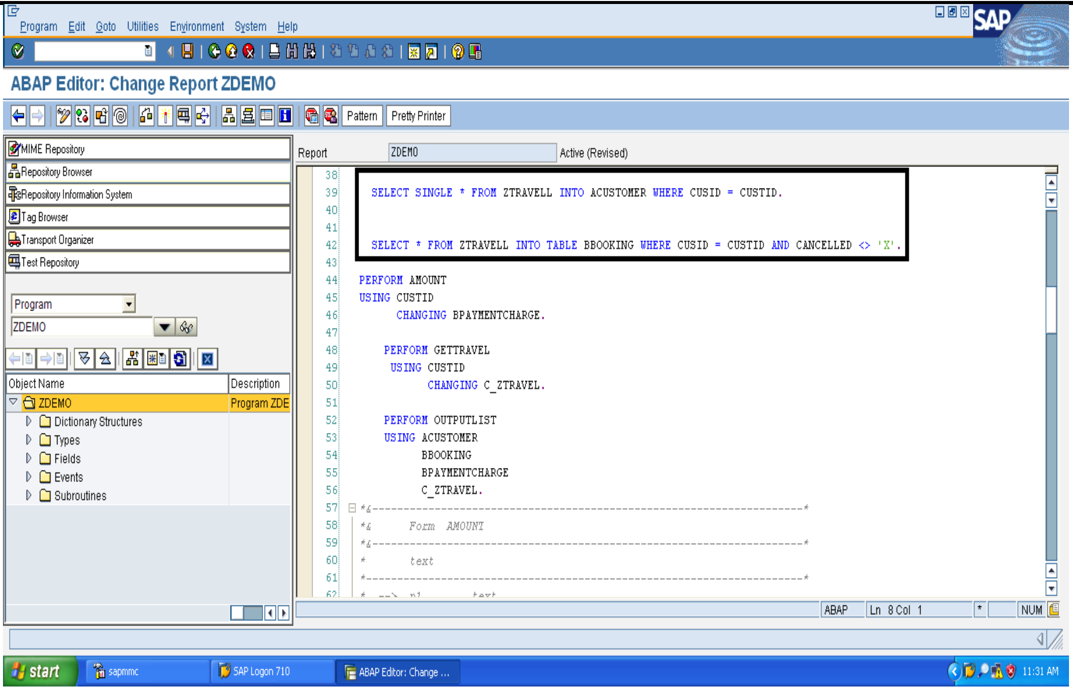
According to the different data structures, the table is divided into three types: standard tables, sorting table, hash table. Standard tables, data without any organization, read out directly from the database into the internal table. Sort the table; want to read out the data to sort by keyword, drained after order placed within the table. The hash table is to read the data in accordance with a certain hash algorithms for data organization in place in the table. These three keys can be used within a table access, standard tables and sorted tables can be accessed using an index, but can only use keys to access the hash table, and, the key must be unique, otherwise the run-time error.

II. PROPOSED SYSTEM

When large amount of data is involved, ABAP program development will directly affect the system performance. The ABAP program in question could collapse the entire server. Therefore, the optimization of ABAP programs plays a key role. In the ABAP program optimization process, the inner table is an effective way to improve efficiency. After discussing on the types within the table, read mode and joining the internal tables, the existing system provides an improved performance and improved time complexity for ABAP applications. The proposed system, studies about the different types of modularization techniques that can be used along with internal tables in order to increase the efficiency of the applications. Using modularization techniques also allows the reusability and easy maintenance of the code.

III. SQL STATEMENTS

Proper use of the SELECT command is important for optimizing the performance of application logic. In addition, other SQL statements as well as the overall program logic have to be optimized. Multiple reads of the same data, as well as using identical statements, is unnecessarily processing-intensive, and should be avoided whenever possible. Instead, you could buffer the data in internal tables, or buffer database tables in general. To reduce the number of data records that have to be transferred, you should specify a sufficiently selective WHERE clause in each SQL statement. A SELECT statement without a WHERE condition is an indication of error in the design of the program. There are two ways to read a fixed number of records: You can transfer the records to the application server and discard the ones you do not need in the SELECT loop. The better method, however, is to use SELECT UP TO n ROWS and only transfer the desired number of data records from the database to the application. Use of nested select statements has to avoided and also use of nested LOOPs has to avoided. The query has to be designed to use as much index fields as possible in the WHERE statement. Use INNER (or OUTER) JOIN in the SELECT statement to retrieve the matching records at one shot. Using INTO CORRESPONDING FIELDS OF TABLE has to be avoided during buffered access. Instead, use the most appropriate for the program. Using ORDER BY in SELECT statements must be avoided if it differs from the index used (instead, sort the resulting internal table), as additional work might be added to the database system which is unique, while there are many ABAP servers. Executing an identical SELECT (same SELECT, same parameter) multiple times in the program should be avoided.

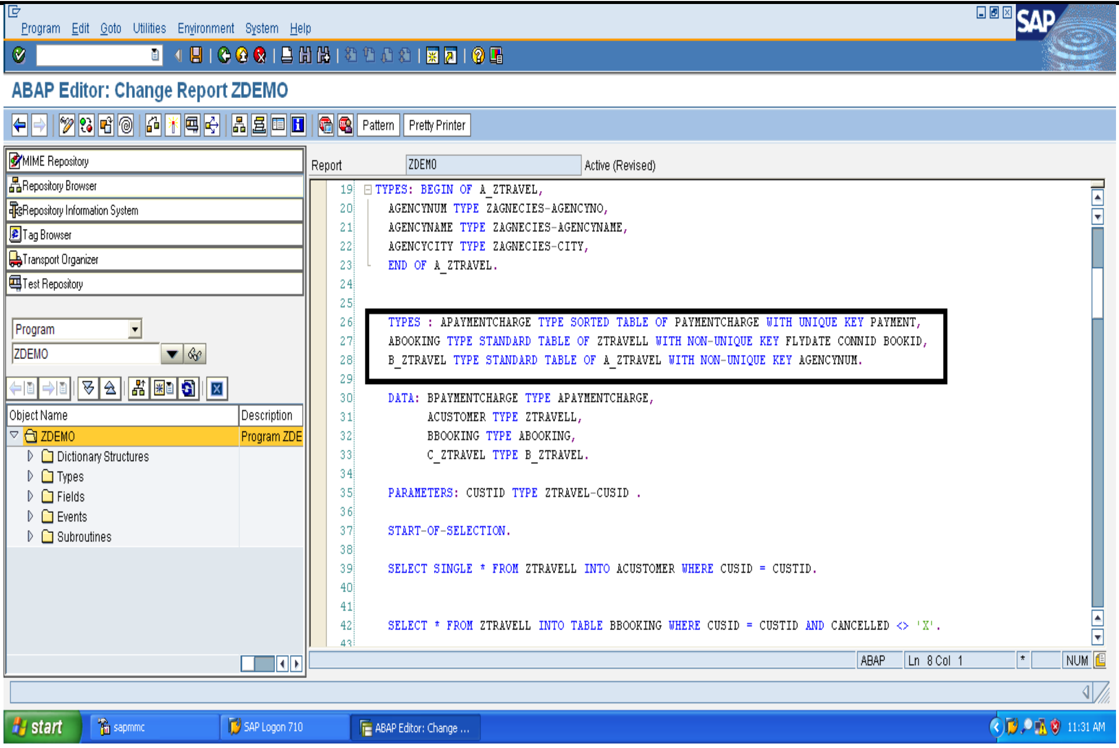


```
SELECT SINGLE * FROM ZTRAVELL INTO ACUSTOMER WHERE CUSID = CUSTID.


SELECT * FROM ZTRAVELL INTO TABLE BBOOKING WHERE CUSID = CUSTID AND CANCELLED <> 'X'.

PERFORM AMOUNT
USING CUSTID
      CHANGING BPAYMENTCHARGE.

    PERFORM GETTRAVEL
     USING CUSTID
          CHANGING C_ZTRAVEL.

    PERFORM OUTPUTLIST
    USING ACUSTOMER
          BBOOKING
          BPAYMENTCHARGE
          C_ZTRAVEL.
*&---------------------------------------------------------------------*
*&      Form  AMOUNT
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
```

Fig 1: Using SELECT statements in an optimizing way

IV. INTERNAL TABLES

A temporary table, which contains the records of an ABAP program that it is being executed, is called as an internal table. An internal table exists only during the run-time of a SAP program. They can be used to process large volumes of data in an ABAP application. An internal table in an ABAP program has to be declared when you need to retrieve data from database tables. Data in an internal table will be stored in rows and columns as in a regular database table. Each row is called as a line and each column is called as a field. All the records have the same structure and key in an internal table. The individual records of an internal table can be accessed with an index or a key. The internal table exists till the associated program is being executed and similarly, when the program is terminated the records of the internal table are also discarded. Hence, internal tables are used as temporary storage areas or temporary buffers where data can be modified as required. These internal tables occupy memory only at the run-time and not at the time of their declaration. As internal tables exist only when a program is running, the internal table must be structured in such a way that the program can make use of it, when the code is written. The internal tables operate in the same way as structures. The main difference is that the structures have only one line, while an internal table can have as many lines as required. There are three different kinds of internal tables which can be used according to the needs of the application. They are standard internal table, hashed internal table and sorted internal table. A standard internal has an internal linear index. The records can be accessed either by using the table index or the key. The response time for accessing the key will be proportional to the number of entries in the table. Sorted internal tables are always sorted by the key. They also have an internal index.
The records can be either accessed either by using the table index or the key. The response time for key access will be logarithmically proportional to the number of table entries, as the system uses a binary search. Hashed internal tables have no linear index. A hashed table can only be accessed using its key.
The response time will be independent of the number of table entries, and will be constant, as the system accesses the table entries using a hash algorithm.



Report ZDEMO Active (Revised)

```
TYPES: BEGIN OF A_ZTRAVEL,
  AGENCYNUM TYPE ZAGNECIES-AGENCYNO,
  AGENCYNAME TYPE ZAGNECIES-AGENCYNAME,
  AGENCYCITY TYPE ZAGNECIES-CITY,
  END OF A_ZTRAVEL.


  TYPES : APAYMENTCHARGE TYPE SORTED TABLE OF PAYMENTCHARGE WITH UNIQUE KEY PAYMENT,
  ABOOKING TYPE STANDARD TABLE OF ZTRAVELL WITH NON-UNIQUE KEY FLYDATE CONNID BOOKID,
  B_ZTRAVEL TYPE STANDARD TABLE OF A_ZTRAVEL WITH NON-UNIQUE KEY AGENCYNUM.

  DATA: BPAYMENTCHARGE TYPE APAYMENTCHARGE,
        ACUSTOMER TYPE ZTRAVELL,
        BBOOKING TYPE ABOOKING,
        C_ZTRAVEL TYPE B_ZTRAVEL.

  PARAMETERS: CUSTID TYPE ZTRAVEL-CUSID .

  START-OF-SELECTION.

  SELECT SINGLE * FROM ZTRAVELL INTO ACUSTOMER WHERE CUSID = CUSTID.


  SELECT * FROM ZTRAVELL INTO TABLE BBOOKING WHERE CUSID = CUSTID AND CANCELLED <> 'X'.
```

Fig 2: Defining Internal Tables

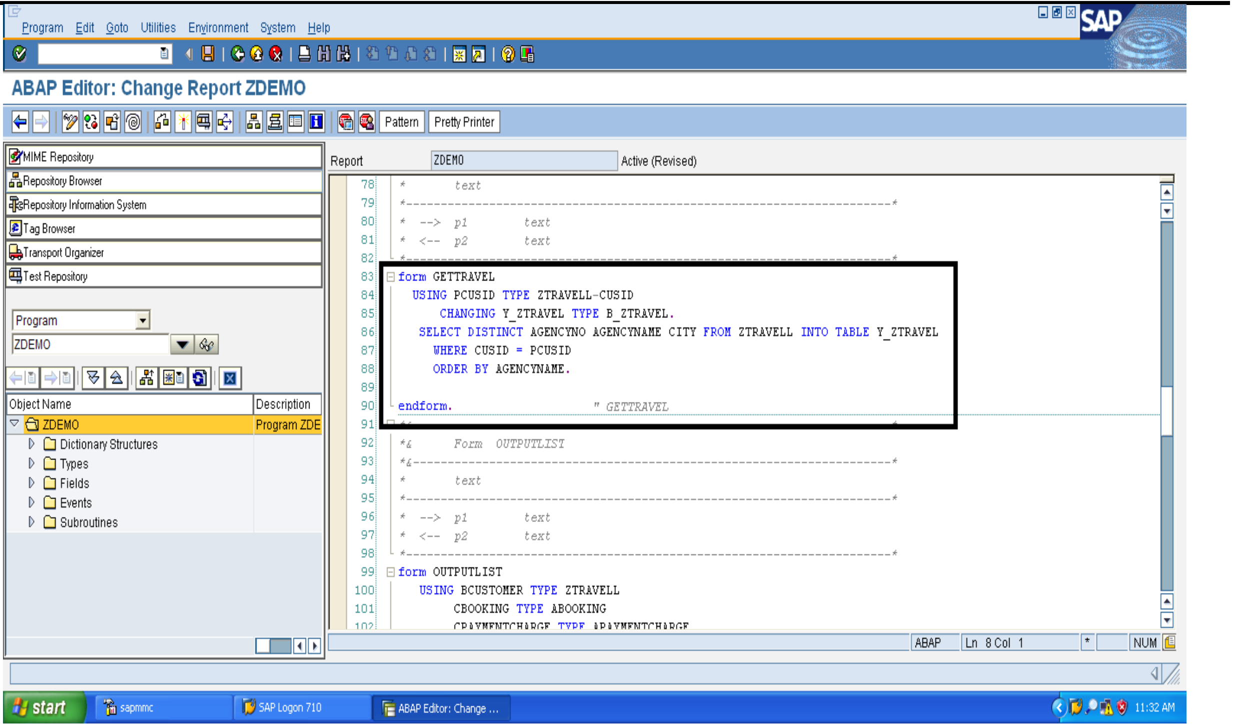
V. MODULARIZATION TECHNIQUES

The organization of programs into modular units, also known as logical blocks in an SAP ABAP environment is known as Modularization. It can be used to reduce the redundancy and increase program readability while creating it and subsequently during the maintenance cycle. Modularization also helps in reusing the same code again. It is necessary for the developers to modularize, i.e. to organize the programs relatively more, than in the OOPS-based languages that have relatively more built-in modular features. Once a small, modularized section of code is complete, debugged and so on, it does not subsequently have to be returned to, and developers can then move on and focus on other issues. The different modularization techniques include subroutines, function modules and macros.

VI. SUBROUTINES

A reusable section of an ABAP code is called as subroutine. A unit within the program where a function is encapsulated in the form of source code is referred to as a subroutine. Subroutines are procedures that can be defined in any ABAP program and also can be called from any ABAP program. Subroutines in SAP ABAP programming can be classified into two as (1) Local subroutines(2) External subroutines. The subroutines in which definition (perform) and implementation (from) are available in the same program are local subroutines. Subroutines which contain definition and implementation that are available in different programs are called as external Subroutines. The program which is implemented must be of type sub-routine pool. In the proposed system, we have tried using subroutines for increasing the efficiency of the ABAP application. The syntax for subroutine is,

```
FORM <subroutine_name>

<statement>

ENDFORM.
```



Report ZDEMO Active (Revised)

```
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form GETTRAVEL
  USING PCUSID TYPE ZTRAVELL-CUSID
      CHANGING Y_ZTRAVEL TYPE B_ZTRAVEL.
   SELECT DISTINCT AGENCYNO AGENCYNAME CITY FROM ZTRAVELL INTO TABLE Y_ZTRAVEL
     WHERE CUSID = PCUSID
     ORDER BY AGENCYNAME.

endform.                    " GETTRAVEL
*&---------------------------------------------------------------------*
*&      Form  OUTPUTLIST
*&---------------------------------------------------------------------*
*       text
*----------------------------------------------------------------------*
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------*
form OUTPUTLIST
   USING BCUSTOMER TYPE ZTRAVELL
        CBOOKING TYPE ABOOKING
        CPAYMENTCHARGE TYPE APAYMENTCHARGE
```

Fig 3: Use of subroutines

VII. CONCLUSION

ABAP is a programming language that is being using by many companies for developing their applications. It plays an important role in the implementation process of the ERP . The pros and cons of ABAP program will directly affect the application of ERP in business. The internal tables in the ABAP program plays an important role in enhancing the efficiency of ABAP programs. In the existing system, the internal tables have been used to increase the efficiency of ABAP programs. Now, in the proposed system, along with internal tables modularization techniques are used to increase the efficiency of ABAP applications further more. The use of modularization techniques also ensure the optimization of the code, code reusability and easy maintenance of the code.

VIII. ACKNOWLEDGEMENTS

We acknowledge DST-File No.368. DST - FIST (SR/FIST/College - 235/2014 dated 21 - 11 - 2014) for financial support and DBT-STAR College-Scheme ref.no: **BT/HRD/11/09/2018** for providing infrastructure support.

IX. REFERENCES

1. Cristea A. D., Prostean O., Muschalik T. and Tirian O.,(2009) "An access control pattern based on qualifications to grant access to physic resources", DAAAM International Young Researchers, The 3rd European and Scientists Conference 2528, Austria, ISBN 978-3-901509-70-4, pp. 1765 – 1766.

2. Cristea A. D., Berdie A. D., Osaci M. and Chirtoc V. (2009), "The Advantages of using Mind Map for learning Web Dynpro", Computer Applications in Engineering Education.

3. Ya Li Feng, Wen Wen Jiang, Qing Jun Tao (2012), "The Optimization Of Efficiency In ABAP Application Program Based On Internal Table", International Conference on Systems and Informatics, pp.1098-1100.

4. Mihaela Osaci1, Adela Diana Berdie1, and Ana Daniela Hammes (2010), "A Study on Modularization Techniques", Soft Computing Applications, AISC 195, pp. 707– 713.

5. Li Jinxian, Bai Xinjiang "Overview of ABAP programming in SAP system(J)", Inner Mongolia Petrochemical Industry, 2009.Vol.10, pp.39-40.

6. Cristea A. D., Prostean O., Muschalik T. and Tirian O.,(2009) "An access control pattern based on qualifications to grant access to physic resources", DAAAM International Young Researchers, The 3rd European and Scientists Conference 2528, Austria, ISBN 978-3-901509-70-4, pp. 1765 – 1766.

7. Camille Rogers(2008), "ABAP Objects: Designing A Programming Course For Information Systems Students Using Sap Software", Georgia Southern University, Vol. IX, No. 1,pp.165-167.

8. https://en.wikipedia.org/wiki/ABAP

9. https://en.wikipedia.org/wiki/SAP_ERP

10. https://www.tutorialspoint.com/sap_abap/sap_abap_overview.htm

11. https://searchsap.techtarget.com/definition/ABAP

12. https://www.guru99.com/all-about-sap-internal-tables.html

13. https://www.sapnuts.com/courses/core-abap/modularization-in-abap/modularization.html

14. https://wiki.scn.sap.com/wiki/display/ABAP/ABAP+Performance+and+Tuning

15. https://www.sapapplication.com/sap-abap-it.php