THE DEVELOPMENT OF A DATABASE DESIGN TOOL FOR AN INTEGRATED ENVIRONMENT

by

MRS. TAWEWAN PROMGUNTHA

Final Report of the Three-Credit Course
CS 6998 System Development Project

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in Computer Information Systems
Assumption University

December 1992
THE DEVELOPMENT OF A DATABASE DESIGN TOOL FOR AN INTEGRATED ENVIRONMENT

By

MRS. TAWEWAN PROMGUNTHA

Final Report of the Six-Credit Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN COMPUTER INFORMATION SYSTEMS

ASSUMPTION UNIVERSITY

DECEMBER, 1992
THESIS TITLE: THE DEVELOPMENT OF A DATABASE DESIGN TOOL FOR AN INTEGRATED ENVIRONMENT.

NAME: Mrs. Tawewan Promguntha

THESIS ADVISOR: Prof. Dr. Srisakdi Charmonman

CO-ADVISOR: Dr. Suphakit Chittayasothorn
Dr. Ouen Pin-Ngern

ACADEMIC YEAR: 1992

The graduate School of Assumption University had approved this Thesis, submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems.

APPROVAL COMMITTEE:

(Prof. Dr. Srisakdi Charmonman) (Dr. Suphakit Chittayasothorn)
Advisor Co-advisor

(Prof. Dr. Ouen Pin-Ngern) (Assoc. Prof. Dr. Kanchit Malaiwongs)
Co-advisor Member

(Dr. Sudhiporn Patumtaewapibal) (Dr. Buranawong Sowapruksa)
Member Member

 assassinated

(Assoc. Prof. Somchai Thayarnyong)
Member

DECEMBER, 1992
ABSTRACT

The design of an integrate database system is a challenging task. It comprises database design for individual departments, each of which should have minimum redundancy. Integration of these separately designed database needs both technical and managerial skills.

On the technical side, it much be ensured that naming conversions are correct and conceptual schemas design are correct. A mapping from a conceptual schema model is needed. From experience gained at the NRCT information system project, a software tool for these integration and mapping tasks are proposed and developed. This thesis describes the design and development of the software tool and the integration of the actual systems at NRCT.
บทความ

ระบบการออกแบบระบบฐานข้อมูลเป็นงานที่น่าสนใจมาก การออกแบบระบบฐานข้อมูลของแต่ละหน่วยงาน ซึ่งแต่ละหน่วยงานจะมีข้อมูลในการทำงานที่ต่างกัน เป็นอย่างมาก ดังนั้นการออกแบบฐานข้อมูลจึงไม่ใช่การออกแบบฐานข้อมูลเท่านั้น แต่เป็นการออกแบบที่มีคุณสมบัติหลาย ซึ่งต้องอาศัยทั้งทฤษฎีและประสบการณ์ในการจัดทำ

การออกแบบฐานข้อมูล คือการตั้งใจว่าชื่อของฟิลด์ (Field) ที่เกี่ยวข้อง แล้วการออกแบบฐานข้อมูลด้วยคำศัพท์ (Conceptual Schema) ถูกต้องตามทฤษฎี จากประสบการณ์ การวางแผนสารสนเทศคอมมิวนิเคชั่นของสถาบันแห่งชาติ ซึ่งเป็นการทำงานออกแบบและ 9 องค์ เพื่อให้ได้ชื่อกับชื่อของแต่ละหน่วยงาน เช่น อริยม์ ณัฐา จัดทำสำนวนสำหรับการออกแบบฐานข้อมูล และ ตารางต่างๆ ที่เกี่ยวข้อง

วิทยาลัยพยาบาลบัณฑิต แต่ละการออกแบบระบบสารสนเทศคอมมิวนิเคชั่น การรวบรวมฐานข้อมูล และพัฒนาโปรแกรมสำเร็จ รูปโดยอาศัยข้อมูลจริงของสถาบัน วิจัยแห่งชาติ
ACKNOWLEDGEMENT

This thesis is successfully done with the great help and cooperation through many people. The researcher would like to express the deepest appreciation to Professor Srisakdi Charmomman whose advice is very practical for studying the data, Dr. Ouern Pin-ngern for his advice of data arrangements and analysis, Dr. Suphamit Chittayasothorn for his advice of database design and his problem solving, Acharn Suchat Kijtanaseri for his advice of work-system analysis and his training of Program CD5/ISIS Use, and Acharn Suradej Chaiwongkao for his advice of language use.

The researcher would also like to give big thanks to the Director of Policy and Research Planning Division for her helpful cooperation in collecting data, Mrs. Saowaluk Phoomwasana for her coordination, advice, details which are very helpful for data analysis, and her impressive service while the researcher was studying the data at the National Research Council of Thailand. The research finally gives thanks to the Head of Policy and Research Planning Division and her staff for their time and their impressive service.

There are, however, both some errors appear in this thesis which the researcher expects them to be corrected in the future, and some practical ideas which help the users in their work.

Tawewan Prompunthi
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>iii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td># 1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background of the Thesis</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Scope and Limitation</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Research Methodology</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Benefits</td>
<td>4</td>
</tr>
<tr>
<td>1.6 Deliverables</td>
<td>4</td>
</tr>
<tr>
<td># 2 National Research Council of Thailand</td>
<td>5</td>
</tr>
<tr>
<td>2.1 History of The National Reserach Council of Thailand</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Office of The National Reserach Council of Thailand</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Existing System</td>
<td>14</td>
</tr>
<tr>
<td>2.3.1 Struction Analysis</td>
<td>15</td>
</tr>
<tr>
<td>2.3.2 Existing Data Flow Diagram</td>
<td>16</td>
</tr>
<tr>
<td>2.3.3 Existing system of The Policy and Planning Division</td>
<td>17</td>
</tr>
<tr>
<td>2.3.3.1 Clerical Section</td>
<td>17</td>
</tr>
<tr>
<td>2.3.3.2 Meeting Section</td>
<td>18</td>
</tr>
<tr>
<td>2.3.3.3 Policy and Budget Section</td>
<td>20</td>
</tr>
<tr>
<td>2.3.3.4 Scientific and Technological Research planning Section</td>
<td>21</td>
</tr>
<tr>
<td>2.3.3.5 Social Research planning Section</td>
<td>22</td>
</tr>
</tbody>
</table>
Chapter

3 The user's requirement and the proposed Computer Information System
  3.1 The user's requirement of Computer Information System
    3.1.1 Clerical Section
    3.1.2 Policy and Budget Section
    3.1.3 Meeting Section
    3.1.4 Scientific and Technological Research planning Section
    3.1.5 Social Research planning Section
  3.2 The proposed Computer Information System

4 Screen Design
  4.1 Program Function
    4.1.1 Meeting Section
    4.1.2 Scientific and Technological Research Planning Section
    4.1.3 Social Research planning Section
  4.2 Screen Display and its operation

5 Basic Concept
  5.1 Review of Computer Aided Software Engineering Technology
  5.2 Addition Improvements from Previous Procedures
  5.3 Relation Database Design using NIAM
    5.3.1 Conceptual Schema Design procedure
    5.3.2 NIAM Conceptual Schema Model
    5.3.3 Transformation Algorithm
  5.4 NIAM Conceptual Schema Diagram of Sample Divisions
  5.5 Data Structures from the manual transformation of NIAM to 5NF Schema
  5.6 Problems from manual Transformation of NIAM to 5NF Schema design experiences
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>The design of the database designer software tool</td>
<td>82</td>
</tr>
<tr>
<td>6.1 System Architecture</td>
<td>82</td>
</tr>
<tr>
<td>6.2 System Table</td>
<td>83</td>
</tr>
<tr>
<td>6.3 File Structure</td>
<td>84</td>
</tr>
<tr>
<td>6.4 Output Report</td>
<td>84</td>
</tr>
<tr>
<td>6.5 User Interface</td>
<td>86</td>
</tr>
<tr>
<td>6.5 Software and Hardware Requirement</td>
<td>97</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Conclusion and Further Research</td>
<td>110</td>
</tr>
<tr>
<td>7.1 Conclusion</td>
<td>110</td>
</tr>
<tr>
<td>7.2 Further Research</td>
<td>111</td>
</tr>
<tr>
<td>Reference</td>
<td>112</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>Appendix A Program Listing on Development of the Research policy and Planning Division</td>
<td>113</td>
</tr>
<tr>
<td>Appendix B Program Listing on Development of the Intergrated Database Designer</td>
<td>175</td>
</tr>
<tr>
<td>Appendix C NIAM Conceptual Schema</td>
<td>219</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Data Structure from Transforming NIAM Conceptual Schema from Figure 5.2</td>
<td>80</td>
</tr>
<tr>
<td>5.2</td>
<td>Data Structure from Transforming NIAM Conceptual Schema from Figure 5.3</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Meta Table</td>
<td>84</td>
</tr>
<tr>
<td>6.2</td>
<td>Example of Data Dictionary Table</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Identify the name of each Division</td>
<td>101</td>
</tr>
<tr>
<td>6.4</td>
<td>Example Code Define Table</td>
<td>102</td>
</tr>
<tr>
<td>6.5</td>
<td>Example Define Name of Group Table of division A</td>
<td>103</td>
</tr>
<tr>
<td>6.6</td>
<td>Example Define Name of Group Table of division B</td>
<td>104</td>
</tr>
<tr>
<td>6.7</td>
<td>Example Data of entity to Integrate Data</td>
<td>105</td>
</tr>
<tr>
<td>6.8</td>
<td>Example Output Report for Group A</td>
<td>107</td>
</tr>
<tr>
<td>6.9</td>
<td>Example Output Report for Group B</td>
<td>108</td>
</tr>
<tr>
<td>6.10</td>
<td>Example Output Report for base_name</td>
<td>109</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Organization Chart of NRCT</td>
<td>Page</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Data Flow Diagram Symbols</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Context Diagram of the Policy and Planning Research Division</td>
<td>24</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Data Flow Diagram (Level 0) of the Research Policy and Planning Division</td>
<td>25</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Data Flow Diagram of Clerical Section</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Data Flow Diagram of Meeting Section</td>
<td>27</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>Data Flow Diagram of Policy and Budget Section</td>
<td>28</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Data Flow Diagram of Social Research Planning Section and Scientific and Technological Research Planning Section</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>System Flow of Clerical Section</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>System Flow of Meeting Section</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>System Flow of Policy and Budget Section</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>System Flow of Social Research Planning Section</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>NIAM Conceptual Schema</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Function add Minute of conference</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Function delete Minute of conference</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Function modify Minute of conference</td>
<td>47</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Function Print Minute of conference</td>
<td>48</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Function add committee biograph</td>
<td>49</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Function delete committee biograph</td>
<td>50</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Function modify committee biograph</td>
<td>51</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>Function search committee biograph</td>
<td>52</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>Function print committee biograph</td>
<td>53</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Function add policy and planning research</td>
<td>54</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Function delete policy and planning research</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Function modify policy and planning research</td>
<td>56</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Function search policy and planning research</td>
<td>57</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.14</td>
<td>Function print policy and planning research</td>
<td>58</td>
</tr>
<tr>
<td>4.15</td>
<td>Function add seminar report</td>
<td>59</td>
</tr>
<tr>
<td>4.16</td>
<td>Function delete seminar report</td>
<td>60</td>
</tr>
<tr>
<td>4.17</td>
<td>Function modify seminar report</td>
<td>61</td>
</tr>
<tr>
<td>4.18</td>
<td>Function search seminar report</td>
<td>62</td>
</tr>
<tr>
<td>4.19</td>
<td>Function print seminar report</td>
<td>63</td>
</tr>
<tr>
<td>4.20</td>
<td>Display screen before access the main menu</td>
<td>66</td>
</tr>
<tr>
<td>4.21</td>
<td>Display screen main menu</td>
<td>66</td>
</tr>
<tr>
<td>4.22</td>
<td>Display screen minutes of conference</td>
<td>67</td>
</tr>
<tr>
<td>4.23</td>
<td>Display screen committee</td>
<td>67</td>
</tr>
<tr>
<td>4.24</td>
<td>Display screen committeeeman</td>
<td>68</td>
</tr>
<tr>
<td>4.25</td>
<td>Display screen policy and planning research</td>
<td>68</td>
</tr>
<tr>
<td>4.26</td>
<td>Display screen seminar report</td>
<td>69</td>
</tr>
<tr>
<td>5.1</td>
<td>Basic Graphical Notation of the NIAM Conceptual Schema</td>
<td>77</td>
</tr>
<tr>
<td>6.1</td>
<td>System Architecture</td>
<td>82</td>
</tr>
<tr>
<td>6.2</td>
<td>Meta Schema</td>
<td>84</td>
</tr>
<tr>
<td>6.3</td>
<td>The Structure of FCODE.DBF</td>
<td>85</td>
</tr>
<tr>
<td>6.4</td>
<td>The Structure of MASTER.DBF</td>
<td>85</td>
</tr>
<tr>
<td>6.5</td>
<td>The Structure of GROUP_MA.DBF</td>
<td>86</td>
</tr>
<tr>
<td>6.6</td>
<td>Display Screen Opening Integrated Database</td>
<td>87</td>
</tr>
<tr>
<td>6.7</td>
<td>Display Screen main menu</td>
<td>87</td>
</tr>
<tr>
<td>6.8</td>
<td>Display Screen Page 1 under input code of entity</td>
<td>89</td>
</tr>
<tr>
<td>6.9</td>
<td>Display Screen Page 2 under input code of entity</td>
<td>89</td>
</tr>
<tr>
<td>6.10</td>
<td>Display Screen Page 3 under input code of entity</td>
<td>90</td>
</tr>
<tr>
<td>6.11</td>
<td>Display Screen Page 4 under input code of entity</td>
<td>91</td>
</tr>
<tr>
<td>6.12</td>
<td>Display Screen Page 1 under code of entity</td>
<td>91</td>
</tr>
<tr>
<td>6.13</td>
<td>Display Screen Page 2 under code of entity</td>
<td>92</td>
</tr>
<tr>
<td>6.14</td>
<td>Display Screen Page 3 under code of entity</td>
<td>92</td>
</tr>
<tr>
<td>6.15</td>
<td>Display Screen Page 4 under code of entity</td>
<td>93</td>
</tr>
<tr>
<td>6.16</td>
<td>Display Screen Page 1 under input data of entity</td>
<td>94</td>
</tr>
</tbody>
</table>
Figure 6.17 Display Screen Page 2 under input data of entity 94
Figure 6.18 Display Screen Page 3 under input data of entity 95
Figure 6.19 Display Screen Page 4 under input data of entity 95
Figure 6.20 Display Screen submenu of the edit section 96
Figure 6.21 Display Screen under print menu 97
Figure 6.22 Niam Conceptual Schema of division A 98
Figure 6.23 Niam Conceptual Schema of division B 99
CHAPTER 1
INTRODUCTION

This thesis aims to on Development of Database Design tool for an Integrated Environment and Computer Information System in the Policy and Planning Division, Office of the National Research Council of Thailand.

The thesis is developed by Mrs. Tawewan Promguntha a graduate student of Assumption University, under the supervision of Prof. Dr. Srisakdi Charmonman, Dr. Ouen Pinngern, and Dr. Suphamit Chittayasothorn, in partial fulfilment of the requirements for the degree of Master of Science in Computer Information Systems.

1.1 Background of the Thesis:

The Computer Information System is a modern technology which we can use to evaluate the organization management. In order to design this system, Operation Research, System Analysis, Data Management, Computer System and Management Information System are involved.

From experience gained at the National Research Council of Thailand (NRCT) information system to comprises database design for individual Divisions, each of Division should have minimum redundancy. A software tool for integrating database and mapping tasks are proposed and developed.

1.2 Objectives:

This thesis concerns an operational system of the office of the National Research Council of Thailand. It aims at developing a computer information system for the Research
policy and Planning Division. This system is a part of the integrated information system of the whole office.

New methodologies are exploited. The activity analysis tool will be the data flow diagram. The data analysis tool will be NIAM (Nijssen’s Information analysis Method). However, due to the specific characteristic of the existing system which has considerable amount of redundancy in both data structures and activities, an integrated methodology is required.

1.3 Scope and Limitations:
There are nine divisions in The National Research Council of Thailand:
- Research Promotion Division
- Research Policy and Planning Division
- Office of the secretary
- Research Evaluation and Project analysis Division
- Transaction and Foreign relations Division
- Research project and Coordination Division
- Research Registration Division
- Library and Documentary service Division
- Remote sensing Division.

This thesis is especially of The Research Policy and Planning Division. This division is divided into five sections:
- Clerical section
- Meeting section
- Policy and Budget section
- Scientific and Technological Research Planning section
- Social research Planning section

The activities of the Research Policy and Planning Division involve generate office business, organizing board
meeting of The National Research Council of Thailand and Academic Committees, forming submitting the annual budget of the Research Council of Thailand, and establishing the Policy and Planning of National Research.

The scope and limitations of this thesis involves the development of computerized activities in each section as follows:

1. Meeting section
   - Collecting the names of the committee and sub-committee resume, and the conclusions of the organizing board meetings.

2. Policy and Budget section
   - Collecting the abstracts of educational analysis of research budget and the national surveys.

   - Collecting the Conclusions of policies and research plans No.1,2,3 significant problems, governmental policies, the conclusions of the meetings and significant seminars.

4. Designing and developing of database the softwares tool an the integration.

1.4 Research Methodology:
1. Literature research.
2. Research and Developmental Methods are divided into three parts:

PART 1  Studying the existing system
   - Data collection by interviewing, record inspection and observation
   - Existing System Analysis
   - Data Flow Diagrams
PART 2
Proposed system design
- Define the user's requirements
- Propose System Flow Charts
- Conceptual schema and Relational Database schema design
- User's Interface Preparation
- User's manual Design
- System comparisons

PART 3
System Implementations
- Database Management System and Software tools studies
- Programs development
- System testing
- Presentation

1.5 Benefits:
1. Facilitates the user in using data.
2. Decreases the repetition of data collection.
3. Increases the divisional work flows.
4. Decreases the expenses of data collection.
5. Accurates the collecting data.

1.6 Deliverables:
1. User's manual
2. Programs
3. Thesis
CHAPTER 2

NATIONAL RESEARCH COUNCIL OF THAILAND

2.1 History of the National research Council of THAILAND.

Documentary evidence show that about 40 years ago, lieutenant - General Praya Sallavidhannithes submitted a proposal to his superior for the government to establish a national research council, without success. Later, Dr. Tua Lapanukrom, the Minister and Director General of Department of science attempted to establish a national research council and a scientific research institute, with a near success. The project covered construction of premises on plots of land extending from the Pharmaceutical Organization through the Ministry of Industry up to the Buddhist Monks Hospital. His premature death put an end to the project. Later, once again, Dr. Chang Ratanarat, the Permanent-Secretary of the Ministry of Industry and Director-General of the Department of Science, had compiled and prepared a draft project, which was duly submitted to the government, for establishment of a nation research council. The matter went to the Manasasila Political Party, who sat on it. Then there was Dr. Pradit Chiewaskul, who came back to Thailand after 18 years of study abroad, and learned about the opportunity to call on the late Field Marshal Pin Chunhawan the then Vice-Premier and convinced him of the benefits to be derived from a national research council as experienced by him in the United States. With the assistance of Field Marshal Pin Chunhawan, the Manasasila Party finally approved the draft project. At last, in 1956, the government promulgated a National Research Council Act for the first time, with the Director-General of the Department of Science as its ex-officio of Secretary General.
A resolution was passed at a meeting of the Council to establish, by virtue of the Act, an Office of the National Research Secretariate, temporarily located at the Department of Science. Dr. Pradit Chiewsakul, then holding the position of a specialist in the Department of Science, was appointed Deputy Secretary-General, with Mrs. Sakultala Potiprasart, the then Chief of Scientific Research Section and presently Deputy Secretary-General of Science Division in the Office of National Research Council, as his Secretary.

Two meetings were held by the National Research Council during 1956-57. Apart from acquiring a permanent office of the Secretariate, The Council had set up its rules and regulations, appointed qualified commissioners and set up the six disciplines of technology to do research on as stipulated in the National Research Council Act of 1956 as follows:

- Physical Science and Mathematics
- Medical Science
- Chemistry and Pharmacology
- Biological Science
- Agriculture and Forestry
- Engineering and Industrial Research

Due to lack of due support such as non-approval of administrative subdivisions and absence of specific budget funds and permanent staff assignments, the Office of the National Research Secretariate had to confine its activities to tabulating those research works already undertaken for government agencies and organizations for the purpose of coordination and avoiding duplication of efforts.

Following the coup d'état on 20th October, 1958, the government under the premiership of Field Marshal Sarit Thanarat had been enthusiastically engaged in all aspects of national development. The government realized that for a
developing country such as Thailand, rapid and comprehensive national development could only be achieved by means of technologies slanted towards psychology, tradition and opinions. These would constitute the bases of forming congenial national policies. To this end, the operation and structure of the National Research Council were revamped. Thus in 1959, a revised version of the National Development Council was promulgated in lieu of the 1956 version. However, despite the new version of the Act, operation proceeded at a slow pace. The authorities and duties conferred on the commissioners were such that in the event of any question coming up for deliberation, the heavily-manned board of commissions had to meet to resolve. Much work was involved in preparing for a board meeting and things moved quite tardily. Hence a second revision was effected in the form of National Research Council Act (No.2) of 1964, promulgated on 16th September, 1974.

A Revolution Proclamation No.216 of 29th September, 1972 established an Office of National Research Council to be attached to Office of the Prime Minister. Revolution Proclamation No.217 of 29th September, 1972 had transferred the official authorities and duties of officers of the National Research Council to an Office of the National Research Council. A Revolution Proclamation No.315 of 13th December, 1972 was therefore issued to amend the legislations on National Research Council to be commensurate with the previous two Proclamations.

Despite the several amendments effected since 1959, the provisions of the law still cannot cope with the current situation. Therefore, in order to facilitate activities and to ensure efficient control and following up the progress of research projects, the law on national development had to be revised. The executive board and subcommittees on various
disciplines would be abolished and the National Research Council alone would be maintained. This Council is to be charged with the responsibility of policy and control of all activities of research conducted by the Office of the National Research Council. Accordingly the previous laws are to be rescinded and a new National Development Commission Act has been proposed to the cabinet ministers for approval in principle and perusal by the Legislation Committee for final touch-up.

2.2 Office of the National Research Council

The Office of the National Research Council (ONRC) is a government agency under the auspices of the Ministry of Science, Technology and Energy, and is charged with the following duties:

- Proposing to the National Research Council of Thailand (NRCT) such policies and projects on promoting research as are presentable to the Cabinet.
- Deliberating on setting up additional technological disciplines and submitting same to the NRCT.
- Deliberating on ways and means of raising funds for research works and submitting a proposal on same to the NRCT to realize the funds needed.
- Submitting annual achievement reports to the NRCT.
- Promoting and conducting researches and establishing research institutes.
- Coordinating inter-discipline research activities.
- Promoting and supporting research in the government and private sectors.
- Setting up a directory of researchers and technocrats in various disciplines.
- Designating incumbents for specific jobs pertaining to research work.
- Preparing budgets on research activities.
- Allocating subsidies and awards on research.
- Maintaining contact and romoting cooperation with foreign research institutes.
- Performing any other statutory duties of the NRCT and the ONRC. The NRCT comprises nine divisions (figure 2-1) as follows:

2.2.1 Office of the Secretary.

Responsibly for documents and records administration of the ONRC, namely inward and outward mail, correspondence, filing, duplicating; finance and accounting; preparation of ONRC budgets; staffing, recruitment, staff transfer and promotion, performance evaluation, staff discipline, employee and director records; purchasing; office materials; building maintenance and motor vehicles.

This division is subdivided into 4 sections, i.e.:
- Correspondence section.
- Financial section.
- Personnel section.
- Supply section.

2.2.2 Research Project and Coordinate Division.

Responsibly for organizing seminars and training classes both in Thailand and abroad; proposing research policies; coordinating research activities of projects; fund raising for research projects; supporting or conducting research in plant and human/animal pest control; serving as a center of technological documents on production and distribution of organic matters for pest control and as center for producing technocrats and researches in this field for the benefit of other agencies; and preparation of international projects.

This Division is subdivided into 5 Sections:
- Clerical section.
Figure 2.1 Organization Chart of NRCT.
International Committees section.
- Project section.
- National Biological Control Center.
- International Organizations section.

2.2.3 Research Registration Division.

Responsible for census and preparation of registry of experts, researchers, research projects and research institutions; following up and collection of research findings and theses in and about Thailand; sorting research findings by science and social science subjects; maintaining a master file of research findings classified by subjects; data bank services; perusal of research projects for any duplication of efforts as may be requested by local and foreign bodies; compiling and publishing technical papers on research works.

This Division is subdivided into 4 Sections:
- Clerical section.
- Survey and Registration section.
- Research compilation section.
- Research Result Service section.

2.2.4 Research Policy and Planning Division.

Responsible for meetings of committees such as the Executive Committee of the NRCT; notification of resolutions; following up on progress of resolutions; preparation and recommendation of policy and budget on research; study and analysis of scientific, technological and social policies; preparation of plans on scientific and social research and following up on operation progress; preparation of amendments to research plans.

This Division is subdivided into 5 Sections:
- Clerical section.
- Meeting section.
- Policy and Budget section.
- Scientific and Technological Research Planning section.
- Social Research Planning section.

2.2.5 Library and Documentary Service Division.

Responsible for compiling a list of published printed material for subscription and exchange; preparing list of ONRC printed material; preparing library cards and publishers' names, both local and foreign; classification of books, cross reference cards, published research reports; collecting research findings and books; recording inter and inter-library loan and return of books by and between libraries; recording factual data to render specific research services; preparation of bibliography and synopses of research reports and thesis; and producing publications.

This Division is subdivided into 5 Sections:
- Clerical section.
- Document Acquisition section.
- Classification and Card Cataloguing section.
- Documentary Service section.
- Searching Service and Distribution section.

2.2.6 Translation and Foreign Relations Division.

Responsible for correspondence with international organizations; translation and publication; verification of translations; conducting and coordinating translation projects; handling cases with other countries in respect of fund raising, seminars, observation trips and scientific activities of UNESCO; entertaining and coordination with foreign specialists including coordination with government entities acting as holder of foreign aid funds and foreign institutions; publication of ONRC journals.

This Division is subdivided into 5 Sections:
- Clerical section.
- Translation section.

-12-
- Translation Projects Coordination section.
- Foreign Relations section.
- Journal section.

2.2.7 Research Evaluation and Project Analysis Division.

Responsible for study and analysis of Projects and Research works on social, scientific and agriculture fields; follow-up for and evaluation of government research findings in cases where budget funds have been allocated; undertaking urgent research projects and findings per resolution of the Cabinet; follow-up and evaluation of research findings abroad; follow-up and acquiring reports on studies, observations, training, research works and conferences in other countries by government personnel with a view for adaptation.

This Division is subdivided into 5 Sections:
- Clerical section.
- Social Project and Research Analysis section.
- Scientific Project and Research Analysis Section.
- Agricultural Project and Research Analysis section.
- Research and Evaluation section.

2.2.8 Research Promotion Division.

Responsible for examination and screening research projects for determination of research subsidies and awards for researchers and inventors; control of subsidised research works; study, analysis, follow-up and monitoring operations of foreign researchers; publishing research findings and activities of ONRC; production of technological journals not otherwise assigned to some other sections and preparing training and seminar projects.

This Division is subdivided into 5 Sections.
- Clerical section.
- Research Grant section.
- Foreign Research section.
- Public and Relations section.
- Training sub-division
  - Clerical section.
  - Training section.
  - Documentary Production section.
  - Audio-Visual Equipment section.

2.2.9 Remote Sensing Division.

Responsible for planning research and follow-up there on; natural resources remote sensing by satellites in conjunction with other local and foreign agencies; publishing research findings; analysis of satellite imagery and of data obtained by ground and aerial surveys; providing services such as satellite data, data on geological mapping and photo images on various scales and use of analysis equipment such as zoom transfer scope etc.; designing or creation of equipment and tools; promotion of technologies on remote sensing; repair and maintenance of image interpretation equipment.

This Division is subdivided into 6 Sections:
- Administration.
- Research Coordination.
- Data Analysis.
- User Service Center.
- Technique and Maintenance.
- Ground Receiving Station.

2.3 Existing System.

In this study and analysis of the existing system, Structure analysis and Design techniques is resorted to, using Data Flow Diagram as a tool, and Data Dictionary as an aid to document recording. For design database, NIAM conceptual schema model has been used.
2.3.1 Structured Analysis

Structured Analysis first appeared in 1970. It strives for a more efficient communication for the use in developing the desired system. It is a set of techniques and graphical tools that allow (the analyst) to develop a new kind of system specifications that are easily understandable to the user. The traditional approach focuses on cost/benefit and feasibility analyses, project management, hardware and software selection, and personnel consideration. In contrast, structured analysis considers new goals and structured tools for analysis.

Some basic structured tools are the context diagram the data flow, diagram, data dictionary, structured English, decision trees, and decision tables. The objective is to build a new document, called system specifications. This document provides the basis for design and implementation.

The structured specification consists of the Data Flow Diagrams that show the major decomposition of system functions and their interfaces, the data dictionary documenting all interface flows and data stores in the Data Flow Diagrams. Structured analysis first looks at the overall picture of the system logically and then narrows down to study of details. It is a top-down technique.

2.3.2 Existing Data Flow Diagram

The Data Flow Diagram here used on the analysis of existing system is in graphic form, showing the flow of data (movement of information) into and out of the system from both the physical viewpoint (how it is done) and the logical viewpoint (what is done), its flow within the system, the processing functions which modify the data and storage of data. Data Flow Diagram shows relationship of functions within the system by designation the input and output of data. It is also referred to by acronym DFD.
The following symbols are used in Data Flow Diagram:

**Figure 2.2 Data Flow Diagram Symbols**

**External Entity**

External entity is shown as rectangles to distinguish them from the system under study. It is a source or destination of data or information. It supplies data to, or receives data from system. External entity is outside the boundary of scope of area under study.

**Process**

A process is depicted by a circle. Sometimes it is called a bubble or a transform. Process portraying the transformation of the content or status of data. In other words, some sort of processing step takes place.
Data Flow

Data Flow is shown by line with arrow head indicating the direction of the flow. It shows the interactions between the area understudy and its influencing external entities. Data the system receives are called inputs, and information or data it produces are called outputs. Flow of data between external entities do not show on diagrams because they are external to the area under study.

Data Store

Data Store is a symbol portraying a file or database in which data resides. The data store is shown by the open-ended rectangle symbol. It may be the temporary storage position for data within the system. In a manual system, it may be paper reports, filing cabinets, microfiche, card files, rotary files, or shelves. In a computerized system, it may be tapes, disks, or any data held in memory. This study consists of Data Flow Diagram level 0 showing overall system of the Policy and Planning Division (figure 2.4) and level 1 showing details within the system (figure 2.4 - 2.9).

2.3.3 Existing System of the Research Policy and Planning Division.

The existing system of each section as following:

2.3.3.1 Clerical Section

The general jobs of the section are taking care of the official letters such as receiving and delivering them, drafting and corresponding, sending questionairrs to other offices, registering and sorting them out according to the line of job. The section is also responsible of circulating regulations, law and various announcement which the officials should generally know. Beside, the section controls the expense office, inventory office which makes decision on how
much budget is estimated each purchase and proceed to withdraw the money to give to various part in the section accurately and appropriately according to the needs of each job line. The section is also in charge of every printing matter of the section such as the outgoing letters, conference letters to the meeting of National Research Committee, Academic Affairs Committee of 10 departments, and the sub-committees of those departments. The 10 departments consists of Physical Science and Mathematics, Medical Science, Chemical Science and Pharmacy, Agriculture and Biology, Engineering and Industrial Research, Physiography, laws, Political Science, Economics, and Social Science. The section is also responsible for the minutes of meetings, agendaa, meeting documents, copying, printing out the questionnaires of research plan and research budget. Besides, the section is responsible for, duplicating, padding and binding. To maintain the duplicating machines in order to use them promptly. To co-ordinate with other official sections are other sectional responsibilities.

2.3.3.2 Meeting Section

The present operational procedures of the section:

The present operation of the section is totally operated by people. They are responsible for as follow:

- the meetings of the National Research Council, the Executive Committee, and the Committee of the Ten Departments. They are in charge of the secretary of the meeting. The responsibilities are to record the meeting, write the minute of the meeting, send out the letters of the meeting including the agenda and the appointment.

- the proposals and suggestions related to academic matters to the committee.

- giving service to public concerning various data used for proper consideration such as research consideration,
policy fixation, national serious problems, lists of research.

Operational problems:

The main problems of this section are unedquate men-power overtime consuming because of wasting time for searching for data. Some data have to be transferred from other sections such as Research Promotion Division, and Research Registration Division.

Operational Procedures:

(Samples of research topic fixation according to the roles as mentioned in 2)

- the departmental meeting fixes the annual research including its purposes, research method of each department.
- the section sends the proposals to the Executive Committee to be approved. After being approved, the proposals are sent to the Research Promotion Division. This division, then, announces and issues scholarships including the mentioned research topics.
- the National Research Council also sets up the sub-committee which is responsible for considering scholarship issues. The Research Promotion Division, then, submits the name list to the meeting. The research topics is finally submitted to the Executive Committee to be approved later.

The Procedures in the division

Various information is sent to the secretaries of the divisional clerical and then to the sectional clerical respectively. The secretary, then, organizes them and their additional information to be considered in the meeting, sends out the meeting appointment, agenda, and other documents to the Executive Secretary of clerical section to send to the Committee of the Ten Departments or the Executive Committee. Each committee meets monthly. Each committee is informed at least a week in advanced.

In the meeting of the Executive Committee of the
National Research Council by receiving various topics from other divisions, the executive secretary of the council is the secretary, and the vice-executive secretary—the assistant, and the meeting in-charge leads the meeting. For the departmental meeting, the committee is responsible for gathering various information which is required by the department in order to be used for consideration such as the policy fixation of each department, research fixation, the fixation of the main problems of each department. The committee is also in charge of recording the meeting, informing the meeting agreements, setting up the agenda, and organizing the departmental seminars.

The Size of File: The area of 7 MB is used to various data.

2.3.3.3 Policy and Budget Section

The present procedures:

At present, the work of policy and budget is totally men-operated. The responsibilities are:

1. to propose the sectional administrative policies and budget.
2. to propose the national administrative and budget.
3. to be responsible for other assigned work.

The present operation of the section:

1. To propose the sectional policies and budget concerning the amount of the budget of each plan. The plans are divided into 4 groups:
   a. the Developmental Administration of Science and Technology Plan
   b. the Scientific and Technological Capability Building Plan
   c. the Scientific and Technological Development Plan.

The procedures of budgeting:
When the section sets up a meeting, the principal of the section considers the proposed budget. If there is an added or some budget which he considers improper, the proposal is sent back to be readjusted. When approved, each section clarifies the details of their own budget. The policy and budget section organizes the divisional budget to be sent to finance office, Ministry of Finance, and Budget Division respectively.

The Size of the File: The area of 3 MB. is used to record the data.

2.3.3.4 Scientific and Technological Research Planning Section

The System of the present operation:

Every task is operated to suit the Plan of the National Economical and social Development and the governmental policies. The planning is not departmental but important of orders such as The National Policy and Research (2nd Copy), the energy which does not exist in the department is chosen. The planning is done according to the National Research Council, the governmental policies, and some serious national problems. The planning obtains the data through the inside and outside the department. Some planning work needs to record some data such as the indexes of economy concerning agriculture, exported—economic data, imported various merchandise. These data are very important for planning. Each departmental planning has its own different data.

Problems: Time consuming to search for recorded data.

The Procedures of the Operation:

Each department collects the information, analyzes it, and set up a meeting in each section, meets with the social affairs, proposes to the sub-committee of policy and plan, the committee of policy and plan coordination, the executive
committee, the Ministry of Science, and the Cabinets respectively.

The Size of the File: Most data is used with Social Research Planning Division. The additional data is 10 MB.

2.3.3.5 Social Research Planning Section

The Present System of the work

The present operation of the work in the section requires 4 people. The responsibility is to organize the policy and national research in social science. They are the studies and problem analysis of economics, social development, education, politics, government, administration, laws, national defense, and others related to social problems. The so-called work suits the national economic and social development plan, national problems, governmental policies. According to the agreements of the Cabinets, the section is assigned to make up the national policy and research with the National Economic and Social Development Council. There are, therefore, some representatives from that council in the coordinating sub-committee as well. The responsibilities of the social research section are to collect the studies, analyze the research data, set them up to be the policies and national research planning in order to be the guidelines of other sections, to get rid the repetition, and not to waste the national research budget.

The Problems of the Present System

- The delay and time consuming of the data searching
- The excessive and unorganized data

The Procedures of the operation:

- To study and collects data from other data sources as mentioned so far
- To study and analyze the recorded data in order to search for the tendencies within the next five years
- To select the best choice in setting up the national policy and research planning
- To arrange the order of importance of the research problems
- To set up a meeting with policy and research planning coordinators
- To study the data mutually in scientific and technological matters in order to set up policy and planning
- To make up policy and planning draft by dividing into departments both in social science and scientific aspects
- To summarize the problems in each aspect which should be researched within the next 5 years
- To submit to the sub-committee to approve. The committee includes:
  - the staff of the research council
  - the representatives of the government
  - the representatives of the related private sections
  - the academic representatives from the ten departments
- To proofread the policy and planning drafts to be re-improved
- To submit to the sub-committee to pre-consider before submit to the executive committee
- To submit to the national executive committee to correct and approve
- To submit to the Under-Secretary of Ministry of Science to consider before submitting to the Cabinets
- To submit to the Cabinets to approve
- To announce to the research divisions both governmental and private, ministry of University Affairs

The Size of the File

The area of 51 MB is used for recording data.
Figure 2.4 Data Flow Diagram (level 0) of research policy and planning division
Figure 2.5 | Data Flow Diagram of clerical section
Figure 2.6 Data Flow Diagram of meeting section
Figure 2.7 Data Flow Diagram of Policy and Budget Section
Figure 2.8 Data flow Diagram of social research planning section and Scientific and Technological Research Planning
CHAPTER 3
THE USER'S REQUIREMENT AND THE PROPOSED SYSTEM

3.1 The User's Requirement of Computer Information System

The Needs of computer use in each section:

3.1.1 Clerical Section

The Needs of Computer use

3.1.1.1 The data needs to be recorded
- Secretarial information - to record an incoming and outgoing letters
- Office supplies - requisition - giving out supplies
- Financial withdraw - meeting allowance
- Printing - Numbers of printing

3.1.1.2 Data to be called
- The statistic of incoming and outgoing letters
- The numbers of annual office supplies giving out
- Quantity of printing

3.1.2 Planning and Budget Section

3.1.2.1 The suggestion and annual budget of the department

3.1.2.2 Planning the result of the departmental operation according to the plan and project

3.1.2.3 To collect the summaries of budget research study annual national survey (2521-2529) divided by:
- The source of research budget divided by:
- government departments
- higher education
- private sections
- Non-profit private sections
  (foundations, associations)
- international sections in Thailand
- Budget of research section
- Budget of the departments
- The national research budget in comparison to national budget and non-national budget
- Budget for national research
- Budget for survey
- Budget for reserach support
- Budget for research and development

Additional suggestions:

The study of national research budget, the data from various sections are governmental offices, private sections, higher education sections. These were obtained through questionnaires.

3.1.3 Meeting Section

3.1.3.1. The data recorded
- the minute of meeting of National Executive Research Committee
- the minute of meeting of National Executive Research Committee from Ten departments
- the agreements of various meetings
- lists of project plans—name of project or names of the owners of the projects year earned the scholarship of the section
- non-approved projects
3.1.3.2 Data to be called
- former and present meeting agreements
- former and present approved projects
- criticisms of research approved by various departments.
- research masterpieces, inventions, the best national researchers (work and autobiographies).

3.1.4 Scientific and Technological Research Planning Section

3.1.4.1 The required data to be collected.
- Data collected from the modify, seminars, document and from the various research status studies.
- Policy and National Research Planning 1st (2520 - 2525) especially excerpted about policies and tendencies.
- Research project and research status studies agricultural evaluation medical and Public Health, Environmental, Energy, and Industrial. There are about 9,000 projects.

3.1.4.2 Data to be called.
- The name list of project which annually operated 5-year trace back by using keyword divided by types of research and research evaluation.
- List of researcher's names divided by educational Levels, (Ph.D, Master's, and Bachelor) research, department.
Main government policies, to be update when there is a subject change.

The National Economic and Social Development plan—Research aspects.

Policy and planning of National Research Committee and Science and Technology.

Population income.

Research budget according to division.

Research budget dividedly department.

Policy and Planning of Public Health.

Policy and Planning of Ministry of Defense.

Summaries and Suggestion of the seminars.

3.1.5 Social Research Planning Section

3.1.5.1 Data to be collected:

- Past Researches
  - Former researches, division.
  - Researchers.
  - Budget.

- Research Projects which the division committee approved

- Minutes of meetings, Seminar reports, Research reports for preparing plans.

- Agreements of the ministers on important national problems and tendencies of research for development.

- Other data to be used in planning such as data concerning research projects which needs financial supports and then submitted to analysis division, check up the agreements of the
ministers, data related to Researches provided scholarship of research Promotion division, and data related to researches collected by research registra office and library and documentary service.

- Research Budget documents - Research expense budget (To know which project is important).

- Divisions which provide scholarship divided by domestic and foreign divisions.

3.1.5.2 Data to be called

Besides the data mentioned in science research planning, other data required additionally are needed in a form of tables and graphs. They can be analyzed in order to see the tendency of research development. There are:

- Number of researches in Heading and Subs-Heading group divided by
  - types of research: application divided by development department.
  - fundamental or experiments divided by academic departments.
  - Sources of data.

- Research division

- Research budget divided by division Academic department, research budget in comparison to GDP, GNP.

- Division which provides research scholarship.

- Names of research proposals.
3.2 The Proposed Computer Information System

3.2.1 The record of incoming-outgoing letters for making the statistic of those letters.

3.2.2 The record of the requisition of office supplies, maintenance in order to know the amount of job and check the left-once budget.

3.2.3 The record of requisition of meeting allowance and to check the left-once budget.

3.2.4 The record of the amount of printing to know the amount of job.

3.2.5 The record of minutes of meeting in order to record the minutes of meeting and the agreement of the meeting.

3.2.6 The suggestion and figure out the division budget of each year.

3.2.7 The collection of evaluation of analytic study of research budget and national survey.

3.2.8 The data collection upon the requirements of science technology research planning, and social research planning.

System flow of the proposed system are shown in figure 3.1 to 3.4
Figure 3.1 System Flow of Clerical Section
Figure 3.2 System Flow of meeting section
Figure 3.3 System Flow of Policy and Budget section
Figure 3.4 System Flow of Social research planning section
Scientific and Technological Research Planning
CHAPTER 4
SCREEN DESIGN

4.1 Program Function

The program development of the system as mentioned earlier is called from the program by the main program which functions as follow:

- **menu()** is the function which shows the main menu so the user can select the desired systems which divided into 3 sections: conference section, science and technology, and social research section.

- **conference()** is the function which is called from menu(). It is responsible for testing the lists which the user selects and calls the function. This consists of various functions: add_conf(), del_conf(), mod_conf(), inq_conf(), report_conf, which will be detailed later.

- **science()** is the function which is called from menu() when the user process in the parts of scientific and technology research planning system which show the list of various scientific system so the user can select and call the function of that work. This function consists of add_pol(), del_pol(), mod_pol(), inq_pol(), and report_pol() which will be detailed later.

The various functions which are called from the main menu can be divided the functions according to the sections as follow:

4.1.1 Conference section

There are 5 functions of this conference function. They are:

4.1.1.1 Function add_conf which adds other data concerning the conference. The user inputs the field which is
the keys of the minute of the conference. These keys are the codes of the committee, the chronology of the conference. In doing so, the user can add other data. When the user inputs the mentioned keys and they are already been input, the user cannot add any data. The functions which are called are:

- add_con() - function to add the agenda and the agreements of the conference.

- add_sub_concl() - used when there are more details conference agreements besides "permitted" or "not permitted".

4.1.1.2 Function del_conf which deletes other data concerning the conference. The user inputs the field which is the keys of the minute of the conference. These keys are the codes of the committee, the chronology of the conference. In doing so, the user can delete other data. When the user inputs the mentioned keys and they are already been input, the user cannot add any data. The functions which are called are:

- del_con() - function to delete the agenda and the agreement of the conference.

- del_sub_concl() - function to delete the detailed of agenda and agreements of conference.

4.1.1.3 Function mod_conf is the function of modifying the minutes of conference by displaying the list of the minute to be corrected which the user can select as desire. To do this, the code of the conference committee is input and the chronological order of the conference or search for the minute of the conference to be corrected from the beginning until the wanted one is is found. In case the user does not know the code of the conference committee and the chronological order of the conference, the following functions are used:

- mod_concl() which modifies the agenda
and the agreements of the conference

- \texttt{mod\_sub\_concl()} which corrects the
detailed of the agenda and the agreements of the conference.

4.1.1.4 Function \texttt{inq\_conf} is the
function of displaying the list of detailed of the minute to
be corrected which the user can select as desire. To do this,
the code of the conference committee is input and the
chronological order of the conference or search for the minute
of the conference, the following functions are used:

- \texttt{inq\_concl()} which searches the agenda
and the agreements of conference.

- \texttt{inq\_sub\_concl()} which searches the
detailed of the agenda and the agreements of the conference.

4.1.1.5 Function \texttt{report\_conf} which
printing the minutes of conference report by printer, the
following functions are used:

- \texttt{report\_concl()} function which prints
the agenda and the agreements of the conference.

- \texttt{report\_sub\_concl()} function which
prints the detailed of the agenda and the agreement of the
conference.

4.1.2 Scientific and Technology Research Planning
Section

This section collects data of national policy and
planning research, the following functions are used:

4.1.2.1 Function \texttt{add\_pol} which adds
other data concerning the policy and planning research in
database.

4.1.2.2 Function \texttt{del\_pol} which deletes
data concerning the policy and planning research.

4.1.2.3 Function \texttt{mod\_pol} which modify
data concerning the policy and planning which the user must
input chronology of the policy and planning. In doing so, the
user can modify other data. In case of the user does not know the chronology of the policy and planning, the program will display all of data.

4.1.2.4 Function inq-pol which searches policy and planning research which the user can direct search by input number and chronology of the policy and planning research.

4.1.2.5 Function report_pol which prints the report of scientific and technology research section by printer.

4.1.3 Social Research Planning Section
This section collects the data of seminary report, the following functions are used:

4.1.3.1 Function add_sem which adds other data concerning the seminary report in database.

4.1.3.2 Function del_sem which deletes data concerning the seminary report.

4.1.3.3 Function mod_sem which modifies data concerning the seminary report, which the user must input chronology of the seminary report. Indigo so, the user can modify other data.

4.1.3.4 Function inq_sem which searches seminary report which the user can direct search by input number and chronology of the seminary report.

4.1.3.5 Function report_sem which prints the report of seminary by printer.
Function name: Add minute of conference
Input file: conf, concl
Output file: conf, concl, sub_concl
Input screen: figure 4.22
Output screen:

Figure 4.1 Function add minute of conference

To add the a certain minute of conference, the user should know the data key of the conference and must input the data correctly otherwise the data cannot be added. And to add a certain conference, the related files will be added as well because there are 4 data tables of the conference. Therefore, when the first data table is added, this data, which is repeated in the second data table, will be used in the filed to call the data to add later. The next table can be done the same procedure.
Function name: Delete minute of conference
Input file: conf, concl
Output file: conf, concl, sub_concl
Input screen: figure 4.22
Output screen:

![Diagram](image)

Figure 4.2 Function delete minute of conference

To delete the a certain minute of conference, the user should know the data key of the conference and must input the data correctly otherwise the data cannot be deleted. And to delete a certain conference, the related files will be deleted as well because there are 4 data tables of the conference. Therefore, when the first data table is deleted, this data, which is repeated in the second data table, will be used in the file to call the data to delete later. The next table can be done the same procedure.
Function name: Modify minute of conference
Input file: conf, concl
Output file: conf, concl, sub_concl
Input screen: figure 4.22
Output screen:

Figure 4.3: Function modify minute of conference

When the user wants to modify the data, he has to input the data to be modified first. This data will be used as a key to call the data to be modified. To modify the data of the minute of the conference is also operated the same way. That is, the data of a certain table is used to be the data to call the data in another table to be modified.
Function name: Print out minute of conference
Input file: conf, concl
Output file: conf, concl, sub_concl
Input screen: figure 4.22
Output screen:

Figure 4.4 Function print minute of conference

To print out a certain minute of conference, the user inputs the chronological order of the conference he wants. He also inputs which minute of conference and by which committee.
Function name: Add committee biography
Input file:
Output file: bio_comm
Input screen: figure 4.24
Output screen:

Figure 4.3 Function add committee biography

When the user wants to add biographical data of each committee, first he needs to know to which committee that person belongs so the user can input a committee code and a personal code. Furthermore, the space can be reserved to store the personal data and some other data of that person in case of the his name has not been in the data.
Function name: Delete committee biography
Input file: -
Output file: bio_comm
Input screen: figure 4.24
Output screen:

Figure 4.6 Function delete committee biography

To delete a personal biographical data, the user needs to know which person, the code of the committee, and his personal code. The user has to be sure of those data in order to operate the deletion. The program will ask the user if he is certain to delete the input data. If he is, the data will be deleted.
Function name: Modify committee biography
Input file: -
Output file: bio_comm
Input screen: figure 4.24
Output screen: 

Figure 4.7 Function modify committee biography

When the user wants to modify the data, he has to input the data to be modified first. This data will be used as a key to call the data to be modified. To modify the data of the personal biographical data is also operated the same way. That is the data of a certain table is used to be the data to call the data in another table to be modified.
Function name: Search committee biography
Input file: -
Output file: bio_comm
Input screen: figure 4.24
Output screen:

![Diagram](image)

Figure 4.8 Function search committee biography

To search a personal biographical. The operation of the program is same as the editing but the user cannot edit data.
Function name: Print out committee biography
Input file: -
Output file: bio_comm
Input screen: Figure 4.24
Output screen:

Figure 4.9 Function print committee biography

To print out a committee biography, the user input the committee code.
Function name: Add policy and planning research
Input file: -
Output file: policy
Input screen: figure 4.25
Output screen:

![Diagram of function add policy and planning research]

**Figure 4.18 Function add policy and planning research**

When the user wants to add policy and planning data. First he needs to input an agenda of policy and planning. Furthermore, the space can be reserved to store the policy and planning data and some other data of that policy and planning in case of the agenda has not been in the data.
Function name: Delete policy and planning research
Input file: -
Output file: policy
Input screen: figure 4.25
Output screen:

Figure 4.11 Function delete policy and planning research

To delete a policy and planning data, the user needs to input an agenda of policy and planning. The user has to be sure of those data in order to operate the deletion. The program will ask the user if he is certain to delete the input data. If he is, the data will be deleted.
Function name: Modify policy and planning research
Input file: -
Output file: policy
Input screen: figure 4.25
Output screen:

Figure 4.12 Function modify policy and planning research

When the user wants to modify the data, he has to input the data to be modified first. This data will be used as a key to call the data to be modified. To modify the data of the policy and planning data is also operated the same way. That is the data of a certain table is used to be the data to call the data in another table to be edited.
Function name: Search policy and planning research
Input file: -
Output file: policy
Input screen: figure 4.25
Output screen: 

Figure 4.13 Function search policy and planning research

To search a policy and planning, the operation of the program is same as the editing but the user cannot edit data.
Function name: Print out policy and planning research
Input file: -
Output file: policy
Input screen: figure 4.25
Output screen:

Figure 4.14 Function print policy and planning research

To print out a policy and planning, the user input the key for printing.
Function name: Add seminar report
Input file: -
Output file: seminar
Input screen: figure 4.26
Output screen:

Figure 4.15 Function add seminar report

When the user wants to add seminar report data
First, the needs to input an agenda of seminar report. Furthermore, the space can be reserved to store the report data and some other data of that seminar in case of the agenda has not been in the data.
Function name: Delete seminar report
Input file: -
Output file: seminar
Input screen: figure 4.26
Output screen:

Figure 4.16 Function delete seminar report

To delete a seminar report data, the user needs to input an agenda of seminar. The user has to be sure of those data in order to operate the deletion. The program will ask the user if he is certain to delete the input data. If he is, the data will be deleted.
Function name : Modify seminar report
Input file : -
Output file : seminar
Input screen : figure 4.26
Output screen :

![Diagram of seminar modification process]

Figure 4.17 Function modify seminar report

When the user wants to modify the data, he has to input the data to be modified first. This data will be used as a key to call the data to be modified. To modify the data of the seminar report data is also operated the same way. That is the data of a certain table is used to be the data to call the data in another table to be modified.
Function name: Search seminar report
Input file: 
Output file: seminar
Input screen: figure 4.26
Output screen:

Figure 4.18 Function search seminar report

To search a seminar report. The operation of the program is same as the editing but the user cannot edit data.
Function name: Print out seminar report
Input file: -
Output file: seminar
Input screen: figure 4.26
Output screen:

Figure 4.19 Function print seminar report

To print out a seminar report, the user input the key for printing.

-63-
4.2 Screen Display and Its Operation

This section is about the operation of the program when the user enters the operation of the system which the user can consult in the appendix of this thesis.

The operation of this system is divided into 3 main fields:

- Meeting Section
- Scientific & Technological Research Planning Section
- Social Research Planning Section

When the user enters the system, the screen will show up as shown in the figure 4.20 to 4.26 for the user can select the section he wants. From this team the user can select 3 sections.

4.2.1 When the user chooses the first section, meeting section will show up on the screen as shown in the figure 4.21 which is the details of the meeting.

4.2.1.1 When the user chooses the first section, the meeting will show up on the screen as shown in the figure 4.22. The user, then can select to add, delete, correct, search for, or print out the data he wants.

1- When the user chooses to add the data, "ADD" shows up on the left corner so that the user know the operation of that time. Simultaneously the cursor moves to the first field to wait for an input. When the user chooses to "delete" or "correct" or "search for" or "print out", the operations will be the same as to add the data. That is the signs, "delete" or "correct" or "search for" or "print out" will show up on the left corner. The cursor, then, will move to the first field to wait for the input.

4.2.1.2 When the user chooses the second section, the committee will show up on the screen as shown in the figure 4.24. The user, then, can select to add, delete,
correct, search for, or print out the data he wants.

1- When the user chooses to add the data, "ADD" shows up on the left corner so that the user know the operation of that time. Simultaneously the cursor moves to the first field to wait for an input. When the user chooses to "delete" or "correct" or "search for" or "print out", the operations will be the same as to add the data. That is the signs, "delete" or "correct" or "search for" or "print out" will show up on the left corner. The cursor, then, will move to the first field to wait for the input.

4.2.2 When the user chooses the second section, scientific and Technology Research Planning section will show up on the screen as shown in the figure 4.25.

1- When the user chooses to add the data, "ADD" shows up on the left corner so that the user know the operation of that time. Simultaneously the cursor moves to the first field to wait for an input. When the user chooses to "delete" or "correct" or "search for" or "print out", the operations will be the same as to add the data. That is the signs, "delete" or "correct" or "search for" or "print out" will show up on the left corner. The cursor, then, will move to the first field to wait for the input.

4.2.3 When the user chooses the third section, Social Research Planning section will show up on the screen as shown in the figure 4.26.

1- When the user chooses to add the data, "ADD" shows up on the left corner so that the user know the operation of that time. Simultaneously the cursor moves to the first field to wait for an input. When the user chooses to "delete" or "correct" or "search for" or "print out", the operations will be the same as to add the data. That is the signs, "delete" or "correct" or "search for" or "print out" will show up on the left corner. The cursor,
then, will move to the first field to wait for the input.

The following figures show the characteristics of the statements displayed on each screen and the data the user inputs. They show how many numeral numbers or alphabets if can be input on each screen. [Program Listing in Appendix A.]

![National Research Council of Thailand Research Policy and Planning Division](image)

Figure 4.20: Display screen before access the main menu.

![National Research Council of Thailand Research Policy and Planning Division](image)

Figure 4.21 Display Main Menu
Figure 4.22 Display screen minute of conference

Figure 4.23 Display screen of committee
Figure 4.24 Display screen committee

Figure 4.25 Display screen of policy and planning research
CHAPTER 5

BASIC CONCEPT

5.1 Review of Computer Aided Software Engineering Technology (CASE)

5.1.1 RELATIONAL DATABASE DESIGN

This is not intended to be a definitive guide to database design, but it was felt that the basic technique of logical relational database design from an entity relationship model should be illustrated.

Simple Database Design

Step 1: Each simple entity is translated to a table. A simple entity is one which is not a sub-type or has sub-types of its own. A useful standard is to use the plural form of the entity for the table name.

Step 2: Each attribute is translate into a candidate column of the same name, at which time a more precise format may be chosen.

- Optional attributes become null columns.
- Mandatory attributes become not-null columns.

Step 3: The components of the unique identifier of the entity become the primary key of the table. Remember that there may be more than one unique identifier for an entity the most used one is chosen.

During this process, relationship end names and/or entity names are used with the attribute names to suggest unique column names for use as part of foreign keys.

Step 4: Many to one (and one to one) relationships become foreign keys. This is, bring down a copy of the unique identifier of each referenced entity from the one end and
use as candidate columns.
  - Optional relationships create null columns.
  - Mandatory relationships create not-null columns.

**Step 5**: Create candidate indexes for each of the:
  - primary key (unique index)
  - foreign keys and
  - those suggested by any Function;
  - Attribute matrix.

**Step 6**: An entity sub-type is simply an entity with its own attributes or relationships, but it also inherits any attributes and/or relationships from its parent entity (super-type) and so on up the hierarchy of super-types.

**Step 7**: There are two basic methods of handling the database design for use with exclusive relationships.

These are:
  - common domain
  - explicit foreign keys.

**Common Domain**

If the remaining foreign keys would all be in the same domain (identical format) then create two candidate columns:
  - Relationship identifier.
  - Entity identifier.

The Relationship identifier column would be used to differentiate between the different relationships covered by the exclusive arc.

**Explicit Foreign Keys**

If the resulting foreign keys would not be in the same domain, create explicit foreign key columns for each relationship covered by the exclusive arc, and make all the resulting columns null (optional). Application code must now be enforce the rule that only one may be entered, and that one be entered if the relationships are mandatory.
Next Step: A large amount of the above default database design is carried out automatically, at your request, by leading CASE software.

But this is only a starting point, as the database design now needs careful scrutinizing to ensure that it provides full support in a performance/space efficient manner for the programs, ad hoc enquiries, archiving, and so on. This may require careful denormalization, controlled replication across a network, and detailed physical design of indexes and disk utilization.

5.1.2 BASIC CONVENTIONS AND DEFINITIONS

This part covers the basic rules, conventions and definitions for entities, relationships, attributes and layout.

Attribute Value Must be Dependent on the Whole Unique Identifier (Second Normal Form)

Remove those attributes for which the values are dependent on only part of the unique identifier. This is known as 'Second Normal Form'. Such attributes normally imply a missing but related entity.

Attributes Must be Dependent on the Unique Identifier (Third Normal Form)

Remove those attributes that are not dependent on the unique identifier of the entity. This is known as Third Normal Form.

Further Conventions

This case has covered the basic conventions and definitions required for entity relationship modelling.

5.1.3 CATEGORIES OF CASE TOOLS

Different CASE tools focus on the support of different phases of the software life cycle or on the development of different types of software systems. To more
easily distinguish CASE tools, they categorize CASE toolkits. Toolkits are the simplest type of CASE tools, they are a set of integrated tools that automate one type of software life cycle task, such as system design or program maintenance, or one type of job class, such as system analyst.

CASE TOOLKITS

CASE toolkits can be described as phase-level tools phase of the life cycle or the development of one type of software system. Some toolkits are part of a family of software products. They can be used alone or in conjunction with other tools within the family.

ANALYSIS TOOLKITS

Analysis toolkits make it easier to follow the principles of good systems analysis and design by automatically taking care of burden some paperwork and countless system details. The purpose of an analysis toolkit is to automate the creation of a system specification describing the system requirements.

Diagramming Tools

The first component, structured diagramming tools, is a set of computerized tools for drawing, manipulating, and storing structured diagrams such as data flow diagrams, dependency diagrams, and entity relationship diagrams. Often, these tools reside on personal computers or workstations that support graphics manipulations.

Real-Time Analysis Toolkits

To represent real-time system specifications, this basic diagram set must be expanded. Diagrams that show sequence and timing relationships between system processes and the effect of external events on the system must be included. Control flow diagrams and state transition diagrams are typically used to model real-time systems and therefore should be added to the diagram set of an analysis
toolkit that supports real-time systems analysis and design tasks.

Prototyping Tools

Prototyping tools are the second component of an analysis toolkit. They are used to help determine system requirement and answer questions about the performance capabilities of the system before it is even built.

Repository

The third component of an analysis toolkit is the repository. All system specification information is stored in the repository.

DATA DESIGN TOOLKITS

CASE data design toolkits support the logical and physical design of databases and files. They support logical data modeling, the automatic conversion of data models to third-normal form, the automatic generation of database schemes for particular database management systems, and the automatic generation of program-code-level file descriptions.

Examples of CASE data design toolkits include AUTOMATE PLUS from Learmonth & Burchett, SQL*DESIGN DICTIONARY from Oracle, IDMS/ARCHITECH from Cullinet, and the CHEN toolkit from Chen & Associates.

PROGRAMMING TOOLKITS

CASE programming toolkit to support program implementation. Many of these tools are familiar and are already widely used by programmers. The difference is that the tools in the programming toolkit are packaged to be compatible with one another. In other words, they have been designed or customized to share common interfaces and to call, use, and feed one another easily. Unix programming environments and the Interlisp programming environments are two good examples of earlier programming toolkits.
Conclusion

- We have described basic techniques of logical relation database design from an entity relationship model. A large amount of the above default database design is carried out automatically.

- We then cover basic rules, conventions and definitions for entities, relationships, attributes and layout, which covers data normalization.

To remove the attributes for which the values are dependent on only part of the unique identifier. This is known as "Second Normal Form". Such attributes imply a missing but related entity.

- Finally we present sets of integrated tool that support one type of software development function or job class. Such as system design or program maintenance, system analyst.

5.2 Additional Improvements From Previous Procedures

5.2.1 The additional improvements can be produced into 5 Normal Form.

5.2.2 The user can simply use them.

The following is the brief presentation of NIAM as shown in 5.3 to 5.5.

5.3 Relation Database Design using NIAM

5.3.1 Conceptual schema design procedure

In this project we use a conceptual schema design procedure (CSDP) that was developed by Professor G.M. Nijsen and Professor Eckhaed Falkenberg and a research team at Control Database Lab, and was introduced to Thailand by Dr.Suphamit Chittayasothorn who was Professor Nijsen's advisee at The University of Queensland, Australia. This
methodology was given the name "NIAM" (Nijssen Information Analysis Methodology). It is also recognized as one of a few accepted conceptual schema model adopted by the ISO working group on conceptual schema and information base.

5.3.2 NIAM Conceptual Schema Model

NIAM was introduced to Thailand around five years ago, so it is not yet widely known in this country. Unlike the traditional relational database design technique of "normalization", NIAM is richer in semantic and can also be transformed to the Optimal Normal Form (ONF) relational database schema. NIAM is used because the major components of NIAM conceptual schema are presented in an easy to read graphical notations and its richer semantic. After interview the users, go through the daily routine works (observations) and examine samples reports, relationship of data is studied then the NIAM conceptual schema for the system is designed.

The design of NIAM conceptual schema is presented as a sequenc of 14 steps. Basically, step 1-5 are concerned with designing the type of schema, and step 6 - 14 consider the constraint schema.

The 14 steps are as follows:

1. Transform information examples into elementary fact.
2. Apply quality checks to step 1.
3. Draws first draft of the type schema diagram.
4. Apply quality checks to choice of entity types.
5. Determine any subtypes.
6. Add any uniqueness constraints.
7. Check at fact types are of the right entity.
8. Add necessity and frequency constraints.
9. Check that naming conventions are correct.
10. Add quality, subset and exclusion constraints.
11. Add all other constraints, at least verbally.
12. Check that the original information examples are consistent with the conceptual schema.

13. Check that redundancies cannot occur in the database.

14. Check that all verbally expressed constraints are formally included in the conceptual schema.

In drawing the NIAM diagram, the basic graphical notation used are in figure 5.1.

Figure 5.1 Basic Graphical Notation of the NIAM Conceptual Schema
5.3.3 Transformation algorithm

To transform NIAM diagram (Figure 5.2) to relational database schema, we aim that there is no repeating attributes and redundancy free. The transformation used the optimal normal form algorithm, which produces a table design that is in optimal normal form immediately with no need to consider the lower normal forms. The optimal normal form is equivalent to 5th normal form. In contrast to a traditional approach to table design which begins with the first normal form and successively refines this through to the fifth normal form.

The transformation algorithm may be stated briefly as:

1. Create a relation for entity type that has one-to-many binary relationship with other entities and has uniqueness constraint at its side. The primary key for this relation is label types with one-to-one relationship to the considered entity type. If there are more than one of this kind of label type, simply choose one and the others are left as candidate keys.

2. Create a table for a group of entity types with n-ary relationships to each other (n>2). The primary key is a combine key of entity types with uniqueness constraint cover on them.

3. Create a table for entity types with many-to-many relationship to each other. The primary key is a combine key of all those entity types.

4. Entity types on the opposite side of entity type with mandatory constraint (.) cannot has null value, write "NN" (not null) on them.
5.4 NIAM Conceptual Schema Design of Sample Division

To set up Integrated Database Designer System such as NIAM Conceptual Schema of 2 division. The first one is the Policy and Planning Division which is represented by the "division A" and the Remote Sensing Division represented by "Division B". The NIAM diagrams of the two division are shown in Figure. 5.2 and 5.3 as follow.

5.5 Data Structure from the manual transformation of NIAM to 5th Normal Form Schemata

From the Conceptual schema design procedure of the two divisions, "division A" and "division B", it can be formed in a table in the fifth normal form of each division as shown in table 5.1 and 5.2 from the manual as follow.

Table 5.1 and 5.2 are the Relational Diagram of each division. Each division has its own overlapping management. The Database Management System Redundancies cannot occur in the database. The management of the Relational Database Schema for the whole office is, therefore, complicated and time consuming.

Because of this problem, the Development of a Database Design Tool for an Integrated Environment occurs.

The management of Conceptual Schema Design of the two divisions by manual are shown in Table 5.3.

5.6 Problems from the manual transformation of NIAM to 5th Normal Form Schema design experiences.

1. Each division uses different name inspite of the fact that it has the same entity name.

2. It is complicated to collect the related entity in the same area.

3. It is confusing to manage the Relational Database Schema because of its excessive data.
Table 5.1 Data Structure from Transforming NIAM Conceptual Schema figure 3.2

<table>
<thead>
<tr>
<th>1. GRANT</th>
<th>2. GR_SPENT</th>
<th>3. GR_CNT</th>
<th>4. CO_ADD</th>
<th>5. REV/R</th>
<th>6. GR_SUBJ</th>
<th>7. SUBJ_GR</th>
<th>7. GR-bg</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANT_ID</td>
<td>GRANT_M</td>
<td>GR_REM</td>
<td>GR_CNT</td>
<td>GR_CATGR</td>
<td>GR_YEST</td>
<td>GR_Y_M</td>
<td>CORESCHR</td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>GR_Y_M</td>
<td>GR_SPENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>GR_Y_M</td>
<td>GR_SPENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>CNT_TYPE</td>
<td>MEETING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>CNT_TYPE</td>
<td>MEETING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCVR_ID</td>
<td>RCVR_M</td>
<td>RCVR_ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCVR_ID</td>
<td>RCVR_M</td>
<td>RCVR_ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>RCVR_ID</td>
<td>RCVR_DATE</td>
<td>SUBJECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>RCVR_ID</td>
<td>RCVR_DATE</td>
<td>SUBJECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>13</td>
</tr>
</tbody>
</table>
Table 5.2 Data Structure from Transforming MDM Conceptual Schema figure 5.3

<table>
<thead>
<tr>
<th>RELATIONAL DIAGRAM : GRANT FOR B DIVISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. GRANT</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>GR_ID</td>
</tr>
<tr>
<td>GR_NAME</td>
</tr>
<tr>
<td>SRC_NAME</td>
</tr>
<tr>
<td>FIELD</td>
</tr>
</tbody>
</table>

| 2. PROJECT |          |           |
|            | NH        |
| P_ID       | 59        |
| P_NAME     |           |

| 3. GRANT_CA |           |
|            |           |
| GR_ID      | 53        |
| CATEGORY   |           |

| 4. GRANT_PJ |           |
|            |           |
| GR_ID      | 61        |
| P_ID       |           |
CHAPTER 6
THE DESIGN OF THE DATABASE DESIGNER SOFTWARE TOOL

To solve the Transformation problem as mentioned in Chapter 5, the idea of managing software tool occurs. The procedures for managing are introduced in this chapter.

6.1 System Architecture

The System Architecture of the database design software is shown in Figure 6.3

![Diagram of System Architecture]

Figure 6.1 System Architecture
6.2 System Table

The integrated database by computer needs to collect related data in the form of table. There are 5 related tables:

6.1.1 Table 1: Data Dictionary.

Table 1 comes in a form of data dictionary. It is a reference which defines the various entities, what data types are. This table summarizes what division has a similar entity by looking at narrative description and inquiring from the user and then Table 2 can be formed.

6.1.2 Table 2: Collection of the whole entity.

Table 2 is the collection of the whole entity from Table 1. Because each division has its different entity in spite of the fact that they are the same entity, the entities, therefore, are named to be collected in the database with its divisional name.

Entity "GRANT" Division A is called GRANT ID and Division B is called GR-ID; are the same entity. They are unique ID number of grant.

6.1.3 To manage Meta Schema as shown in Figure 6.2, the tables: Table 3, Table 4, and Table 5 can be created as shown in Table 6.1.

6.2 The File Structure

Verification of the file structure for database are in Figure 6.3, 6.4, and 6.5.

6.4 Output Report

The output report can be obtained through Relational Diagram which is transformed from NIAM Conceptual Schema Diagram in the 5NF. This can be created by using its name in the database and the one appears in each division. The following is the sample which was obtained through the true data from the work of NRCT.
**Figure 6.2 Meta Schema**

**Table 6.1 Meta Table**

<table>
<thead>
<tr>
<th>ROLE</th>
<th>ENTITY_ID</th>
<th>FACT</th>
<th>CONSTRAINT</th>
<th>PRIMARY</th>
<th>MANDATORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE</td>
<td>ENTITY_ID</td>
<td>FACT</td>
<td>CONSTRAINT</td>
<td>PRIMARY</td>
<td>MANDATORY</td>
</tr>
</tbody>
</table>

2. ENTITY_ID

<table>
<thead>
<tr>
<th>ENTITY_ID</th>
<th>ENTITY_ID</th>
</tr>
</thead>
</table>

3. FACT

<table>
<thead>
<tr>
<th>FACT</th>
<th>ENTITY_ID</th>
</tr>
</thead>
</table>
Structure for database: A:\FCODE.DBF

Number of data records: 0
Date of last update: 10/28/92

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CODE</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENTITY</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PRIMARY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Total** 22

Figure 6.3 The structure of FCODE.DBF

Structure for database: A:\MASTER.DBF

Number of data records: 0
Date of last update: 10/28/92

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CODE</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ROLE</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ENTITY</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FACT</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CONSTRAINT</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PRIMARY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MANDATORY</td>
<td>Character</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Total** 48

Figure 6.4 The structure of MASTER.DBF
Structure for database: A:\GROUP NA.DBF
Number of data records: 0
Date of last update: 10/25/92

<table>
<thead>
<tr>
<th>Field</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CODE</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENTITY</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>GROUP</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DATA_TYPE</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

** Total ** 41

Figure 6.5 The structure of GROUP NA.DBF

6.5 User's interface

This step on screen design, the physical detail of data structure from previous step is being used.

In designing the screen, the emphasis is on facilitating user ease and speed of entry. Since type of user who will input of information to terminal for this system is a general information collection and entry, not a dedicated data entry one, the procedures must be as simple and easy to understand as possible.

This screen design provides the user with a menu of possible functions from which to choose, and also with the simulated forms to fill in. Also the video screen provides sophisticated graphic displays and has ability for answering user queries about specific data.

The access screen is shown in figure as following:

The Main Menu is shown in figure 6.7

The Input Code of Entity are shown in figure 6.8 to figure 6.11

The Input Group are shown figure 6.12 to figure 6.15
Figure 6.6 Display Screen Opening Integrated Database

ASSUMPTION UNIVERSITY (A.U.)
INTEGRATED DATABASE DESIGNER

MAIN MENU Date 19/12/92
By Tawee Trasiyot

1. INPUT CODE OF ENTITY
2. INPUT GROUP
3. INPUT DATA OF ENTITY
4. EDIT SECTION
5. PRINT OUT
6. EXIT

Press or to move highlight, and press ← to select

Command <C> Enter a FoxBASE+ command
NumCaps

Figure 6.7 Display Screen Main Menu
The input Data of Entity are shown figure 6.16 to figure 6.17

The Edit Section is shown figure 6.19

The Print out is shown figure 6.21

There are 8 steps to install and load integrated Database Designer System.

STEP 1 : Install the Integrated Database Designer System.

Turn on the computer, place a DOS system, and answer date and time prompts. Now, from the C prompt (C>), Accordingly, place the convenient disk in drive A and enter the INSTALL command from the A prompt. Figure 6.6 should appear on the monitor. Press INT key as instructed Figure 6.7 Integrated Database Designer Main Menu, will then appear.

STEP 2 : Load the Integrated Database Designer System.

Turn on the computer, place a DOS system, and answer date and time prompts. Now, from the C prompt (C>), enter the command CD AU to change to subdirectory containing the Integrated Database Designer program files. Enter INT to load the program. Figure 6.6 Display Screen and Figure 6.7 Integrated Database Designer should then appear on the monitor. A main menu provides six options: input code of entity, input group, input data of entity, edit section, print out, and exit.

STEP 3 : Enter the File Input Code of Entity

Press or move highlight a 1 from the main menu (Figure 6.7) to invoke the input code of entity Menu. It, in turn, allows you to input drive to save file (shown in Figure 6.8). Respond "A, B, C or F" to save file (define section to open file (shown in Figure 6.9). Respond "N" for
INPUT DRIVE TO SAVE DATA?

Command | <C> | I | I | I | NumCaps

Enter a FoxBASE+ command

Figure 6.8 Display Screen Page 1 under input code of entity

HAVE DATA FILE

Do you want (N)ew or (o)ld file?

Command | <C> | I | I | I | NumCaps

Enter a FoxBASE+ command

Figure 6.9 Display Screen Page 2 under input code of entity
new file or "0" for old file. This in turn, invoke page 3 under input code of entity (shown Figure 6.10), allows you to enter data, when you are satisfied the input code of entity menu program works, enter "Y" to recheck data on page 4 under input code of entity (shown in Figure 6.11). Press any key the program return to main menu.

STEP 4: Enter Input Group

Press or move high light a 2 from the main menu (Figure 6.7) to invoke the input group menu. It, in turn, allows you to input drive to save file (shown in Figure 6.12) Respond A, B, C or F to save (depending on your system). This, in turn, invoke define group section, allow you to respond "How many group to define." (shown in Figure 6.14). This, in turn, allows you to enter entity name of each group and range (shown in Figure 6.15).

ASSUMPTION UNIVERSITY (A.U.)
INTEGRATED DATABASE DESIGNER

CODE DEFINE SECTION 1- Date 19/12/92

By Tawewan Promguntha

<table>
<thead>
<tr>
<th>INPUT CODE</th>
<th>AT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY</td>
<td>TEST</td>
</tr>
<tr>
<td>PRIMARY KEY(Y/N)</td>
<td>N</td>
</tr>
<tr>
<td>ARE YOU SURE?</td>
<td>Y</td>
</tr>
<tr>
<td>MORE DATA?</td>
<td>Y</td>
</tr>
</tbody>
</table>

Command 1<Ct>:1 | 1 | 1 | 1 NumCaps
Enter a FoxBASE+ command

Figure 6.10 Display Screen Page 3 under input code of entity
CODE DEFINE SECTION 1-

INPUT CODE: A1
HAVE THIS CODE ???
PLEASE RECHECK
ENTITY NAME: GRANT_ID

Press any key

Command |<Ct>| | | | | NumCaps
Enter a FoxBASE+ command

Figure 6.11 Display Screen Page 4 under input code of entity

CODE DEFINE SECTION 1-

INPUT DRIVE TO SAVE DATA?

Command |<Ct>| | | | | NumCaps
Enter a FoxBASE+ command

Figure 6.12 Display Screen Page 1 under input code of entity
ASSUMPTION UNIVERSITY (A.U.)
INTEGRATED DATABASE DESIGNER

DEFINE GROUP SECTION :-
Date 19/12/92
By Tawewan Promguntha

HOW MANY GROUP TO DEFINE (1-9)? : 1

Command |<C:>|  |  |  |  | NumCaps
Enter a FoxBASE+ command

Figure 6.13 Display Screen Page 2 under code of entity

ASSUMPTION UNIVERSITY (A.U.)
INTEGRATED DATABASE DESIGNER

DEFINE GROUP SECTION :-
Date 19/12/92
By Tawewan Promguntha

GROUP : 1  GROUP : TEST

Command |<C:>|  |  |  |  | NumCaps
Enter a FoxBASE+ command

Figure 6.14 Display Screen Page 3 under code of entity
ASSUMPTION UNIVERSITY (A.U.)
INTEGRATED DATABASE DESIGNER

DEFINE GROUP SECTION :-

GROUP : 1 GROUP : TEST
CODE : A1 ENTITY NAME : GRANT_ID PRIMARY KEY : N

USE ENTITY NAME : GRANT_ID DATA TYPE : 8N

Command |<C1>|
| | | | | NumCaps

Enter a FoxBASE+ command

Figure 6.15 Display Screen Page 4 under code of entity

STEP 5 : Enter Input Data of Entity
Press or move highlight a 3 from the main menu (Figure 6.6) to invoke the input data of entity menu. It, in turn, allows you to input drive to save file (shown in Figure 6.16). Respond A, B, C or F to save (depending on your system). This, in turn, invoke input data section (shown in Figure 6.11), allow you to enter number of role, fact, constraint and mandatory.

STEP 6 : Edit the Record
Press or move highlight a 4 from the main menu (Figure 6.6) to invoke the edit section menu (shown in figure 6.20). It provides four submenu edit of entity, edit group, edit data of entity and exit.

-93-
Figure 6.16 Display Screen Page 1 under input data of entity

Figure 6.17 Display Screen Page 2 under input code of entity
<table>
<thead>
<tr>
<th>INPUT DATA SECTION</th>
<th>Date 19/12/92</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Tawewan Promguntha</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT CODE</th>
<th>A1</th>
<th>GRANT_ID</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FACT</td>
<td>F3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td>(O)ne, (M)any, (T)many-to-many</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Enter a FoxBASE+ command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;C&gt;</code></td>
<td>FCODE_F1</td>
</tr>
</tbody>
</table>

**Figure 6.18 Display Screen Page 3 under input code of entity**

<table>
<thead>
<tr>
<th>INPUT DATA SECTION</th>
<th>Date 19/12/92</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Tawewan Promguntha</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT CODE</th>
<th>A1</th>
<th>GRANT_ID</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FACT</td>
<td>F3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTRAINT</td>
<td>ONE</td>
<td>MANDATORY</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Enter a FoxBASE+ command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;C&gt;</code></td>
<td>FCODE_F1</td>
</tr>
</tbody>
</table>

**Figure 6.19 Display Screen Page 4 under input code of entity**
ASSUMPTION UNIVERSITY(A.U.)
INTEGRATED DATABASE DESIGNER

EDIT MENU 1- Date 19/12/92
By Tawewan Promguntha

1. EDIT CODE OF ENTITY
2. EDIT GROUP
3. EDIT DATA OF ENTITY
4. EXIT

Press or to move highlight, and press ← to select

Command |<C;> | i i i i i NumCaps
Enter a FoxBASE+ command

Figure 6.20 Display Screen submenu of the edit section

STEP 7: Print out Report
Press or move highlight a 5 from the main menu (Figure 6.6) to invoke the print menu. It, in turn, invoke print code of entity, print group, print data of entity, print process, and exit. (Shown in Figure 6.21)

STEP 8: Shut the system down
Press or move highlight a 6 from the main menu (Figure 6.6) to exit the integrated database methodology and end the session. You have successfully turned on the computer.
6.5 Software and Hardware Requirement

The Integrated Database Designer Program is written on FoxBASE. Program Listing in Appendix B.

Hardware Requirements:
- Microcomputer Memory 640 or more
- Hardisk
- Printer

The following is the real sample taken from The National Research Council of Thailand.
Figure 6.22: NIAM Conceptual Schema Diagram of A Division
Figure 6.3 NIAM Conceptual Schema Diagram of division B
<table>
<thead>
<tr>
<th>EN_ID</th>
<th>ENTITY</th>
<th>PK.</th>
<th>DATY</th>
<th>NARATIVE DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>GRANT_ID</td>
<td>Y</td>
<td>8N</td>
<td>UNIQUE ID NUMBER OF GRANT</td>
</tr>
<tr>
<td>A2</td>
<td>GRANT_N</td>
<td></td>
<td>20AN</td>
<td>NAME OF GRANT</td>
</tr>
<tr>
<td>A3</td>
<td>GR_CMNT</td>
<td></td>
<td>10AN</td>
<td>FOLLOW UP COMMITTEE</td>
</tr>
<tr>
<td>A4</td>
<td>GR_YEAR</td>
<td></td>
<td>8AN</td>
<td>DATE OF GRANT WORKING</td>
</tr>
<tr>
<td>A5</td>
<td>GR_SPENT</td>
<td></td>
<td>10N</td>
<td>AMOUNT OF GRANT SPENTED IN BAHT</td>
</tr>
<tr>
<td>A6</td>
<td>MEETING_NO</td>
<td></td>
<td>7AN</td>
<td>NUMBER OF MEETING</td>
</tr>
<tr>
<td>A7</td>
<td>CMNT_TYPE</td>
<td></td>
<td>1N</td>
<td>TYPE OF COMMITTEE</td>
</tr>
<tr>
<td>A8</td>
<td>MT_DATE</td>
<td></td>
<td>8AN</td>
<td>DATE OF GRANT RECEIVE</td>
</tr>
<tr>
<td>A9</td>
<td>GR_TYPE</td>
<td></td>
<td>15A</td>
<td>NAME OF TYPE DESCRIPTION</td>
</tr>
<tr>
<td>A10</td>
<td>CURRENCY</td>
<td></td>
<td>4A</td>
<td>CURRENCY UNIT</td>
</tr>
<tr>
<td>A11</td>
<td>GR_AMT</td>
<td></td>
<td>12N</td>
<td>AMOUNT OF GRANT BUDGET</td>
</tr>
<tr>
<td>A12</td>
<td>CORESCHR</td>
<td></td>
<td>30A</td>
<td>NAME OF FOREIGN CORESEARCHER</td>
</tr>
<tr>
<td>A13</td>
<td>RCVR_ID</td>
<td></td>
<td>8AN</td>
<td>UNIQUE ID NUMBER OF RESEARCHER</td>
</tr>
<tr>
<td>A14</td>
<td>REV_ADD</td>
<td></td>
<td>30AN</td>
<td>ADDRESS OF RESEARCHER</td>
</tr>
<tr>
<td>A15</td>
<td>RCVR_N</td>
<td>Y</td>
<td>30AN</td>
<td>NAMES OF RESEARCHER</td>
</tr>
<tr>
<td>A16</td>
<td>GR_DATE</td>
<td></td>
<td>8AN</td>
<td>GRANT DATE APPROVED BY COMMITTEE</td>
</tr>
<tr>
<td>A17</td>
<td>SUBJECT</td>
<td></td>
<td>50A</td>
<td>BRANCH OF GRANT SUBJECT</td>
</tr>
<tr>
<td>A18</td>
<td>GR_REM</td>
<td></td>
<td>200AN</td>
<td>REMARK OF GRANT</td>
</tr>
<tr>
<td>A19</td>
<td>GR_RSC</td>
<td></td>
<td>40A</td>
<td>GRANT RESOURCE</td>
</tr>
<tr>
<td>A20</td>
<td>GR_CATGR</td>
<td></td>
<td>1N</td>
<td>CATEGORY OF GRANT</td>
</tr>
<tr>
<td>A21</td>
<td>GR_TY</td>
<td></td>
<td>1N</td>
<td>TYPE OF CORESEARCH GRANT</td>
</tr>
<tr>
<td>A22</td>
<td>GR_Y_EST</td>
<td></td>
<td>8AN</td>
<td>YEAR ESTIMATE</td>
</tr>
<tr>
<td>B49</td>
<td>GR_NR</td>
<td></td>
<td>2N</td>
<td>NUMBER OF GRANT THAI PROVIDE</td>
</tr>
<tr>
<td>B50</td>
<td>COMMENT</td>
<td></td>
<td>200AN</td>
<td>REMARK OR COMMENT</td>
</tr>
<tr>
<td>B51</td>
<td>GR_AMT</td>
<td></td>
<td>10N</td>
<td>AMOUNT OF GRANT IN BAHT</td>
</tr>
<tr>
<td>B52</td>
<td>FIELD</td>
<td></td>
<td>8N</td>
<td>UNIQUE ID NUMBER OF GRANT FI</td>
</tr>
<tr>
<td>B53</td>
<td>CATAGORY</td>
<td></td>
<td>25AN</td>
<td>CATAGORY OF PROVIDED GRANT</td>
</tr>
<tr>
<td>B54</td>
<td>GR_ID</td>
<td></td>
<td>8AN</td>
<td>UNIQUE NUMBER OF GRANT</td>
</tr>
<tr>
<td>B55</td>
<td>DATE</td>
<td></td>
<td>8AN</td>
<td>DATE</td>
</tr>
<tr>
<td>B56</td>
<td>SRC_NAME</td>
<td></td>
<td>30AN</td>
<td>SOURCE OF GRANT NAME</td>
</tr>
<tr>
<td>B59</td>
<td>GR_NAME</td>
<td>Y</td>
<td>160AN</td>
<td>NAME OF GRANT</td>
</tr>
<tr>
<td>B61</td>
<td>_ID</td>
<td></td>
<td>6N</td>
<td>UNIQUE ID NUMBER OF PROJECT</td>
</tr>
<tr>
<td>BASE_NAME</td>
<td>DIVISION</td>
<td>OLD_NAME</td>
<td>DATA TYPE</td>
<td>RANGE</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>GRANT_ID</td>
<td>A</td>
<td>GRANT_ID</td>
<td>N</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR_NAME</td>
<td>A</td>
<td>GRANT_N</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR_CMNT</td>
<td>A</td>
<td>GR_CMNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR_Y_EST</td>
<td>A</td>
<td>GR_Y_EST</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_Y_W</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>GR_SPENT</td>
<td>A</td>
<td>GR_SPENT</td>
<td>N</td>
<td>10</td>
</tr>
<tr>
<td>MEETING_NO</td>
<td>A</td>
<td>MEETING_NO</td>
<td>AN</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>CMT_TYPE</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>MT_DATE</td>
<td>AN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_TYPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>CURRENCY</td>
<td>A</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>GR_AMT</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>CORESCHR</td>
<td>A</td>
<td>CORESCHR</td>
<td>A</td>
<td>30</td>
</tr>
<tr>
<td>RCVR_ID</td>
<td>A</td>
<td>RCVR_ID</td>
<td>N</td>
<td>8</td>
</tr>
<tr>
<td>REVR_ADD</td>
<td>A</td>
<td>REVR_ADD</td>
<td>AN</td>
<td>50</td>
</tr>
<tr>
<td>GR_DATE</td>
<td>A</td>
<td>GR_DATE</td>
<td>AN</td>
<td>8</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>A</td>
<td>SUBJECT</td>
<td>A</td>
<td>50</td>
</tr>
<tr>
<td>GR_REM</td>
<td>A</td>
<td>GR_REM</td>
<td>AN</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>COMMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR_RSC</td>
<td>A</td>
<td>GR_RSC</td>
<td>A</td>
<td>40</td>
</tr>
<tr>
<td>GR_CATGR</td>
<td>A</td>
<td>GR_CATGR</td>
<td>N</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>CATAGORY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GR_TY</td>
<td>A</td>
<td>GR_TY</td>
<td>A</td>
<td>15</td>
</tr>
<tr>
<td>GR_NR</td>
<td>B</td>
<td>GR_NR</td>
<td>N</td>
<td>2</td>
</tr>
<tr>
<td>FIELD</td>
<td>B</td>
<td>FIELD</td>
<td>N</td>
<td>8</td>
</tr>
<tr>
<td>DATE</td>
<td>B</td>
<td>DATE</td>
<td>AN</td>
<td>8</td>
</tr>
<tr>
<td>SRC_NAME</td>
<td>B</td>
<td>SRC_NAME</td>
<td>A</td>
<td>30</td>
</tr>
<tr>
<td>P_ID</td>
<td>B</td>
<td>P_ID</td>
<td>N</td>
<td>8</td>
</tr>
<tr>
<td>CODE</td>
<td>ENTITY</td>
<td>LABEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>GRANT_ID</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>CURRENCY</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A11</td>
<td>GR_AMT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>CORESCHR</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>RCVR_ID</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A14</td>
<td>REVR_ADD</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A15</td>
<td>RCVR_N</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A16</td>
<td>GR_DATE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A17</td>
<td>SUBJCT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A18</td>
<td>GR_REM</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A19</td>
<td>GR_RSC</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>GRANT_N</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A20</td>
<td>GR_CATGR</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A21</td>
<td>GR_TY</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A22</td>
<td>GR_Y_EST</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>GR_CMNT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>GR_Y_W</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A49</td>
<td>GR_NR</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>GR_SPENT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A50</td>
<td>COMMENT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A51</td>
<td>GR_AMT</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A52</td>
<td>FIELD</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A55</td>
<td>DATE</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A56</td>
<td>SKC_NAME</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>MEETING_NO</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A60</td>
<td>P_NAME</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A61</td>
<td>P_ID</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>CMT_TYPE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>MT_DATE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>GR_TYPE</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CODE</td>
<td>ENTITY</td>
<td>GROUP</td>
<td>DATA_TYPE</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>GRANT_ID</td>
<td>GRANT_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>CURRENCY</td>
<td>CURRENCY</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A11</td>
<td>GR_AMT</td>
<td>GR_AMT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>CORESCHR</td>
<td>CORESCHR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>RCVR_ID</td>
<td>RCVR_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A14</td>
<td>REVR_ADD</td>
<td>REVR_ADD</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A15</td>
<td>RCVR_N</td>
<td>RCVR_N</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A16</td>
<td>GR_DATE</td>
<td>GR_DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A17</td>
<td>SUBJECT</td>
<td>SUBJECT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A18</td>
<td>GR_REM</td>
<td>GR_REM</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A19</td>
<td>GR_RSC</td>
<td>GR_RSC</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>GRANT_N</td>
<td>GRANT_N</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A20</td>
<td>GR_CATGR</td>
<td>GR_CATGR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A21</td>
<td>GR_TY</td>
<td>GR_TY</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A22</td>
<td>GR_Y_EST</td>
<td>GR_Y_EST</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>GR_CMNT</td>
<td>GR_CMNT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>GR_Y_W</td>
<td>GR_Y_W</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A49</td>
<td>GR_NR</td>
<td>GR_NR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>GR_SPENT</td>
<td>GR_SPENT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A50</td>
<td>COMMENT</td>
<td>COMMENT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A54</td>
<td>DATE</td>
<td>DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A56</td>
<td>SRC_NAME</td>
<td>SRC_NAME</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>MEETING_NO</td>
<td>MEETING_NO</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A60</td>
<td>P_NAME</td>
<td>P_NAME</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A61</td>
<td>P_ID</td>
<td>P_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>CMT_TYPE</td>
<td>CMT_TYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>MT_DATE</td>
<td>MT_DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>GR_TYPE</td>
<td>GR_TYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CODE</td>
<td>ENTITY</td>
<td>GROUP</td>
<td>DATA_TYPE</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>GRANT_ID</td>
<td>GR_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A10</td>
<td>CURRENCY</td>
<td>CURRENCY</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A11</td>
<td>GR_AMT</td>
<td>GR_AMT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A12</td>
<td>CORESCHR</td>
<td>CORESCHR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A13</td>
<td>RCVR_ID</td>
<td>RCVR_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A14</td>
<td>REVADD</td>
<td>REVADD</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A15</td>
<td>RCVR_N</td>
<td>RCVR_N</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A16</td>
<td>GR_DATE</td>
<td>GR_DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A17</td>
<td>SUBJECT</td>
<td>SUBJECT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A18</td>
<td>GR_REM</td>
<td>GR_REM</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A19</td>
<td>GR_RSC</td>
<td>GR_RSC</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>GRANT_N</td>
<td>GR_NAME</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A20</td>
<td>GR_CATGR</td>
<td>CATEGOR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A21</td>
<td>GR_TY</td>
<td>GR_TY</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A22</td>
<td>GR_Y_EST</td>
<td>GR_Y_EST</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>GR_CMMT</td>
<td>GR_CMMT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>GR_Y_W</td>
<td>GR_Y_W</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A49</td>
<td>GR_NR</td>
<td>GR_NR</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>GR_SPENT</td>
<td>GR_SPENT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A50</td>
<td>COMMENT</td>
<td>COMMENT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A51</td>
<td>GR_AMT</td>
<td>GR_AMT</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A52</td>
<td>FIELD</td>
<td>FIELD</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A55</td>
<td>DATE</td>
<td>DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A56</td>
<td>SRC_NAME</td>
<td>SRC_NAME</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>MEETING_NO</td>
<td>MEETING_NO</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A60</td>
<td>P_NAME</td>
<td>P_NAME</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A61</td>
<td>P_ID</td>
<td>P_ID</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>CMT_TYPE</td>
<td>CMT_TYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>MT_DATE</td>
<td>MT_DATE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A9</td>
<td>GR_TYPE</td>
<td>GR_TYPE</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>CODE</td>
<td>ROLE</td>
<td>FACT</td>
<td>ENTITY</td>
<td>CONSTRAINT</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>A1</td>
<td>R1</td>
<td>F3</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A20</td>
<td>R10</td>
<td>F7</td>
<td>GR_CATGR</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R11</td>
<td>F8</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A22</td>
<td>R12</td>
<td>F8</td>
<td>GR_Y_EST</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R13</td>
<td>F9</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A4</td>
<td>R14</td>
<td>F9</td>
<td>GR_Y_W</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R15</td>
<td>F10</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A4</td>
<td>R16</td>
<td>F10</td>
<td>GR_Y_W</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A5</td>
<td>R17</td>
<td>F10</td>
<td>GR_SPENT</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R18</td>
<td>F11</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A7</td>
<td>R19</td>
<td>F11</td>
<td>CMT_TYPE</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A18</td>
<td>R2</td>
<td>F3</td>
<td>GR_REM</td>
<td>MANY</td>
</tr>
<tr>
<td>A6</td>
<td>R20</td>
<td>F11</td>
<td>MEETING_NO</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A8</td>
<td>R21</td>
<td>F11</td>
<td>MT_DATE</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R22</td>
<td>F12</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A9</td>
<td>R23</td>
<td>F12</td>
<td>GR_TYPE</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A10</td>
<td>R24</td>
<td>F12</td>
<td>CURRENCY</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A11</td>
<td>R25</td>
<td>F12</td>
<td>GR_AMT</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R26</td>
<td>F13</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A12</td>
<td>R27</td>
<td>F13</td>
<td>CRESCHR</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R28</td>
<td>F17</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A21</td>
<td>R29</td>
<td>F17</td>
<td>GR_TYP</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R3</td>
<td>F4</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A17</td>
<td>R30</td>
<td>F1</td>
<td>SUBJEC</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R31</td>
<td>F1</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A13</td>
<td>R32</td>
<td>F1</td>
<td>RCVR_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A15</td>
<td>R33</td>
<td>F1</td>
<td>RCVR_N</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R34</td>
<td>F2</td>
<td>GRANT_ID</td>
<td>MANY</td>
</tr>
<tr>
<td>A17</td>
<td>R35</td>
<td>F2</td>
<td>SUBJEC</td>
<td>ONE</td>
</tr>
<tr>
<td>A12</td>
<td>R36</td>
<td>F14</td>
<td>CRESCHR</td>
<td>ONE</td>
</tr>
<tr>
<td>A14</td>
<td>R37</td>
<td>F14</td>
<td>RCVR_ADD</td>
<td>MANY</td>
</tr>
<tr>
<td>A13</td>
<td>R38</td>
<td>F15</td>
<td>RCVR_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A14</td>
<td>R39</td>
<td>F15</td>
<td>RCVR_ADD</td>
<td>MANY</td>
</tr>
<tr>
<td>A19</td>
<td>R4</td>
<td>F4</td>
<td>GR_RSC</td>
<td>MANY</td>
</tr>
<tr>
<td>A15</td>
<td>R41</td>
<td>F16</td>
<td>RCVR_N</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R5</td>
<td>F5</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A1</td>
<td>R50</td>
<td>F46</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A2</td>
<td>R51</td>
<td>F46</td>
<td>GRANT_N</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R52</td>
<td>F59</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A56</td>
<td>R53</td>
<td>F59</td>
<td>SRC_NAME</td>
<td>MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R54</td>
<td>F58</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A20</td>
<td>R55</td>
<td>F58</td>
<td>GR_CATGR</td>
<td>MANY_TO_MANY</td>
</tr>
<tr>
<td>A1</td>
<td>R56</td>
<td>F54</td>
<td>GRANT_ID</td>
<td>ONE</td>
</tr>
<tr>
<td>A52</td>
<td>R57</td>
<td>F54</td>
<td>FIELD</td>
<td>MANY</td>
</tr>
</tbody>
</table>
Table 6.7 (Continual)

<table>
<thead>
<tr>
<th>CODE</th>
<th>ROLE</th>
<th>FACT</th>
<th>ENTITY</th>
<th>CONSTRAINT</th>
<th>PRIMARY KEY</th>
<th>MANDATORY KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>R58</td>
<td>F55</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A49</td>
<td>R59</td>
<td>F55</td>
<td>GR_NR</td>
<td>MANY</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A2</td>
<td>R6</td>
<td>F5</td>
<td>GRANT_N</td>
<td>MANY</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>A1</td>
<td>R60</td>
<td>F56</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A50</td>
<td>R61</td>
<td>F56</td>
<td>COMMENT</td>
<td>MANY</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A1</td>
<td>R62</td>
<td>F43</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A55</td>
<td>R63</td>
<td>F43</td>
<td>DATE</td>
<td>MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A1</td>
<td>R64</td>
<td>F44</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A55</td>
<td>R65</td>
<td>F44</td>
<td>DATE</td>
<td>MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A55</td>
<td>R66</td>
<td>F78</td>
<td>DATE</td>
<td>MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A61</td>
<td>R67</td>
<td>F78</td>
<td>P_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A55</td>
<td>R68</td>
<td>F74</td>
<td>DATE</td>
<td>MANY</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A61</td>
<td>R69</td>
<td>F74</td>
<td>P_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A1</td>
<td>R7</td>
<td>F6</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A1</td>
<td>R70</td>
<td>F46</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A61</td>
<td>R71</td>
<td>F46</td>
<td>P_ID</td>
<td>MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A11</td>
<td>R72</td>
<td>F57</td>
<td>GR_AMT</td>
<td>MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A61</td>
<td>R73</td>
<td>F57</td>
<td>P_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A1</td>
<td>R74</td>
<td>F47</td>
<td>GRANT_ID</td>
<td>MANY_TO_MANY</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>A61</td>
<td>R75</td>
<td>F47</td>
<td>P_ID</td>
<td>MANY_TO_MANY</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A3</td>
<td>R8</td>
<td>F6</td>
<td>GR_CMMT</td>
<td>MANY</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A1</td>
<td>R9</td>
<td>F7</td>
<td>GRANT_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>A13</td>
<td>RR40</td>
<td>F16</td>
<td>RCVR_ID</td>
<td>ONE</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
Table 6.9 Example Output Report for Group B

INTEGRATED DATABASE DESIGNER
BY......TAWAN PRONGUITHA
ASSUMPTION UNIVERSITY
FOR : GROUP B

12/10/92

A1  GR_ID
    <$$> // **GR_NAME<($)(PK)> // GR_T<($)(PK)> // COMMENT<($)> // CORESCHR<($)> // **DATE<($)> // **FIELD<($)> // **CATEGORY<($)> // GR_CNMT<($)> // GR_NR<($)> // GR_REN<($)> // **GR_RSC<($)> // GR_TY<($)> // GR_T<($)> // **P_ID<($)> // **SRC_NAME<($)> //

A12 CORESCHR
    <$><// REVR_ADD<($)> //

A13 RCVR_ID
    <$><// RCVR_N<($)(PK)> // REVR_ADD<($)> //

A17 SUBJECT
    <$><// GR_ID<($)> //

A61 P_ID
    <$><// DATE<($)> // **GR_AMT<($)> //

F1  A1  GR_ID
    <$><// **RCVR_N<($)(L)> // **SUBJECT<($)> // **RCVR_ID<($)> //

F10 A1  GR_ID
    <$><// GR_T<($)> // GR_SPENT<($)> //

F11 A1  GR_ID
    <$><// CMNT_TYPE<($)> // MEETING_NO<($)> // MT_DATE<($)> //

F12 A1  GR_ID
    <$><// GR_TYPE<($)> // CURRENCY<($)> // GR_AMT<($)> //

F47 A1  GR_ID
    <$><// P_ID<($)> //

F58 A1  GR_ID
    <$><// **CATEGORY<($)> //

REMARK : *** = NH , (PK) = PRIMARY KEY

-108-
| A1 | GRANT_ID | <*>/ **GRANT_H</*>(PK) // GR_T_EST</*>(PK) // COMMENT</*> // CORESCHR </*> // **DATE</*> // **FIELD</*> // **GR_CATGR</*> // GR_CMNT</*> // GR_NK</*> // GR_REM</*> // **GR_BSC</*> // GR_IT</*> // GR_T_Y</*> // **P_ID</*> // **SRC_NAME</*> // |
| A12 | CORESCHR | <*>/ REVR_ADD</*> // |
| A13 | RCVR_ID | <*>/ RCVR_N</*>(PK) // REVR_ADD</*> // |
| A17 | SUBECT | <*>/ GRANT_ID</*> // |
| A61 | P_ID | <*>/ DATE</*> // **GR_ANT</*> // |
| F1 | A1 | GRANT_ID | <*>/ **RCVR_N</*>(L) // **SUBECT</*> // **RCVR_JD</*> // |
| F10 | A1 | GRANT_ID | <*>/ GR_T_Y</*> // GR_SPJMT</*> // |
| F11 | A1 | GRANT_ID | <*>/ CMNT_TYPE</*> // MEETING_NO</*> // MT_DATE</*> // |
| F12 | A1 | GRANT_ID | <*>/ GR_TYPE</*> // CURRNCY</*> // GR_ANT</*> // |
| F17 | A1 | GRANT_ID | <*>/ P_ID</*> // |
| F38 | A1 | GRANT_ID | <*>/ **GR_CATGR</*> // |

**Remark:** "*" = NH , (PK) = PRIMARY KEY
CHAPTER 7
CONCLUSION AND FURTHER RESEARCH

7.1 Conclusion
This thesis is divided into 2 parts. The first part is the development of the Computer Information System in Research Policy and Planning Division of the National Research Council of Thailand.

This begins from studying of Existing system, Define the user's requirements, Propose System, User's manual Design by Screen Display. There are 4 programs developed for this system by C language as follow:

- Record, Modify, Search, and produce the report of the the minute of conference.
- Record, Modify, Search, and produce the report of the the committee biography.
- Record, Modify, Search, and produce the report of the Research Policy and Planning Project.
- Record, Modify, Search, and produce the report of the Research Seminar.

The second part is The development of Database Design Tool for an integrated environment. From experience gained at the NRCT information system to comprises database design for individual Divisions, each of Division should have minimum redundancy. A software tool for integrating database and mapping tasks are proposed and developed.

To integrate the database by manual of each division, it is found that naming convention is complicated to collect the related entity in the same area. It is also confusing to manage the Relational Database Schema because of its excessive
data. It is, therefore, time and men-power consuming.

We, then, establish software a user interface for such work which collects data in the forms of table. Such ideas gain through NIAM by using Meta Schema to get File Structure to write a program on Fox base and then map into 5NF.

7.2 Further Research

There might be Graphic integrate on a NIAM diagram screen because the thesis corrects the data entity in the form of table.
REFERENCES


APPENDIX A

Program Listing on Development of the Research Policy and Planning Division
Listing of menu.c

#include "../def/file.h"
#include <stdio.h>
#include "../def/ss.h"

main()
{
    main_menu();
}

main_menu()
{
    short select;
    clr_crt(1,'a');
    prmp(1,1,first1);
    prmp(1,2,first2);
    prmp(25,8,head26);
    prmp(25,8,head27);
    prmp(25,10,head28);
    prmp(25,12,head29);
    switch(gtube(33,12,con,&select)) {
        case -2 : setcook();
        exit_program();
        break;
        case -3 : setcook();
        main_menu();
        break;
    }
    switch (select) {
        case 1 : setcook();
        conference();
        break;
        case 2 : setcook();
        science();
        break;
        case 3 : setcook();
        social();
        break;
        default : prmp(2,23,warn3);
        setcook();
        main_menu();
        break;
    }
}

conference()
{
    short select;
    mv_cur(1,0,3); eras_in();
    prmp(1,3,warn2);
    prmp(80,3,head1);
    mv_cur(1,0,4); clear();
    prmp(25,5,head29);
    prmp(25,6,head30);
    prmp(25,7,head32);
    prmp(25,8,kk);
    prmp(25,8,head29);
switch(gtube(33,8,con,&select)) {
    case -2 : setcook();
    main_menu();
    break;
    case -3 : setcook();
    conference();
    break;
}
switch(select) {
    case 1 : { short i=0;
        mv_cur(1,0,4); clear();
        prmp(35,8,head35);
        prmp(35,9,head36);
        prmp(35,10,head37);
        prmp(35,11,head38);
        prmp(35,12,head39);
        prmp(35,13,head40);
        prmp(35,15,head22);
        switch(gtube(43,15,con,&i)) {
            case -2 ;
            case -3 : setcook();
                conference();
                break;
            }
        switch(i) {
            case 1 : setcook();
                add_conf();
                break;
            case 2 : setcook();
                del_conf();
                break;
            case 3 : setcook();
                mod_conf();
                break;
            case 4 : setcook();
               inq_conf();
                break;
            case 5 : setcook();
                report_conf();
                break;
            default : setcook();
                conference();
                break;
        }
    }
    case 2 : { short j=0;
        mv_cur(1,0,4); clear();
        prmp(35,8,head35);
        prmp(35,9,head36);
        prmp(35,10,head37);
        prmp(35,11,head38);
        prmp(35,12,head39);
        prmp(35,13,head40);
        prmp(35,15,head22);
        switch(gtube(43,15,con,&j)) {
case -2:
    case -3: setcook();
    conference();
    break;
}
switch(j){
    case 1: setcook();
    add_comm();
    break;
    case 2: setcook();
    del_comm();
    break;
    case 3: setcook();
    mod_comm();
    break;
    case 4: setcook();
    inq_comm();
    break;
    case 5: setcook();
    report_comm();
    break;
    default: setcook();
    conference();
    break;
}

    case 3: { short k=0;
        mv_cur(1,0,4); clear();
        prmp(35,8,head35);
        prmp(35,9,head36);
        prmp(35,10,head37);
        prmp(35,11,head38);
        prmp(35,12,head39);
        prmp(35,13,head40);
        prmp(35,15,head22);
        switch(gtube(43,15,con,&k)) {
            case -2:
                case -3: setcook();
                conference();
                break;
            case 1: setcook();
                add_det();
                break;
            case 2: setcook();
                del_det();
                break;
            case 3: setcook();
                mod_det();
                break;
            case 4: setcook();
                printf(" ");
                inq_det();
                break;
166: case 5 : setcook();
167:       reportابت();
168:       break;
169: default: setcook();
170:       conference();
171:       break;
172: }
173: }
174: case 4 : { short 1=0;
175: mv masturbation(1,0,4); clear();
176: prpm masturbation(35,8,head35);
177: prpm masturbation(35,9,head36);
178: prpm masturbation(35,10,head37);
179: prpm masturbation(35,11,head38);
180: prpm masturbation(35,12,head39);
181: prpm masturbation(35,13,head40);
182: prpm masturbation(35,15,head22);
183: switch (gtube(43,15,con, &1)) {  
184: case -2 :
185: case -3 : setcook();
186:       conference();
187:       break;
188: }
189: switch (1) {  
190: case 1 : setcook();
191:       add_concl();
192:       break;
193: case 2 : setcook();
194:       del_concl();
195:       break;
196: case 3 : setcook();
197:       mod_concl();
198:       break;
199: case 4 : setcook();
200:       inq_concl();
201:       break;
202: case 5 : setcook();
203:       report_concl();
204:       break;
205: default : setcook();
206:       conference();
207:       break;
208: }
209: }
210: }
211: }
212: }
213: science();
214: { short  select;
215: mv masturbation(1,0,3); eras ln();
216: prpm (60,3,head2);
217: prpm (13, warn2);
218: mv masturbation(1,0,4); clear();
219: maintenance menu();
220: switch (gtube(46,15,con, &select)) {  

221: case -2 : setcook();
222:     main_menu();
223:     break;
224: case -3 : setcook();
225:     science();
226:     break;
227: }
228: switch(select) {
229:    case 1 : setcook();
230:          add_pol();
231:          break;
232:    case 2 : setcook();
233:          del_pol();
234:          break;
235:    case 3 : setcook();
236:          mod_pol();
237:          break;
238:    case 4 : setcook();
239:          inv_pol();
240:          break;
241:    case 5 : setcook();
242:          report_pol();
243:          break;
244:    default : prtnmsg(2,23,warn5);
245:          science();
246:          break;
247: }
248: }
249:
250: social() {
251:    short select;
252:    my_cur(1,0,3); eras_in();
253:    prmp(60,3,head3);
254:    prmp(1,3,warn2);
255:    my_cur(1,0,4); clear();
256:    maintenance_menu();
257:    switch(gtube(48,15,con,&select)) {
258:    case -2 : setcook();
259:          main_menu();
260:          break;
261:    case -3 : setcook();
262:          social();
263:          break;
264:    }
265:    switch(select) {
266:    case 1 : setcook();
267:          add_sem();
268:          break;
269:    case 2 : setcook();
270:          del_sem();
271:          break;
272:    case 3 : setcook();
273:          mod_sem();
274:          break;
275:    case 4 : setcook();
276:    }
Listing of menu.c

278:     printf(" ");
277:     inq_sem();
278:     break;
279:     case 5 : setcook();
280:     report_sem();
281:     break;
282:     default : prtnsg(2,23,warn5);
283:     social();
284:     break;
285: }
286: }
287: exit_program()
288: {
289:     clr_crt(1,'a');
290:     if (yorn("are you sure to exit program [y/n] ":") == 1)
291:     {
292:         prmp(29,12,"SEE YOU AGAIN");
293:         prmp(1,23," ");
294:         setcook();
295:         exit();
296:     }
297:     else
298:     {
299:         clr_crt(1,'a');
300:         setcook();
301:         main_menu();
302:     }
303: }
304: }
305: }
306: }
307: }
308: }
309: }
310: }
311: }
312: }
313: }
314: clear()
315: {
316:     printf("%c%c",0x1b,'k');
317: }
318: }
319:
$include "../def/file.h"
#include <stdio.h>
extern char head1[], head5[], head6[], head7[], head8[], head9[];
char ch1[] = "รูปแบบ.
char ch2[] = "[....]"
char ch3[] = "[....]"
char ch4[] = "[....]"
char ch5[] = "[....]"
char ch6[] = "[....]"
char ch7[] = "[1] เซ็นต์มาสเป็น (สต)"
char ch8[] = "[2] เซ็นต์มาสเป็น (ต)"
char ch9[] = "[....]"
char ch10[] = "[....]"
char ch11[] = "[....]"
char ch12[] = "[....]"
char ch13[] = "[....]"
char ch14[] = "[....]"
char ch15[] = "[....]"
char ch16[] = "[....]"
char ch17[] = "[....]"
char ch18[] = "[....]"
char ch19[] = "[....]"
char ch20[] = "[....]"
char ch21[] = "[....]"
char ch22[] = "[....]"
char ch23[] = "[....]"
char ch24[] = "[....]"
char ch25[] = "[....]"
char ch26[] = "[....]"
char ch27[] = "[....]"
char ch28[] = "[....]"
char ch29[] = "[....]"
char ch30[] = "[....]"
char ch31[] = "[....]"
char ch32[] = "[....]"
char ch33[] = "[....]"
char ch34[] = "[....]
char ch35[] = "[1] เซ็นต์มาสเป็น (สต)"
char ch36[] = "[2] เซ็นต์มาสเป็น (ต)
add_comm()
{
short key;
char cmn_name[60];
long addr;
int flag;
int rc ,rcc;
mv_cur(1,0,3); eras_ln();
prmp(1,3,head5); prmp(60,3,head1);
mv_cur(1,0,4); clear();
prmp(3,4,ch1);
prmp(3,5,ch2);
do {
flag=gtube(21,4,comm_id,&key);
switch(flag) {
case 0 : break;
case -2 : setcook();
    conference();
    break;
  case -3 : break;
}
} while (flag == -3);
if(addrvec(comm,&key) != 0)
{
  if((rc=yorn(ch16)==1))
    { setcook();
      add_comm();
    }
  else
    { setcook();
      conference();
    }
  }
else{
  loc(comm,&addr);
  if(lockrec(comm) != 0)
    { prmsg(2,23,ch4);
      setcook();
      conference();
    }
  else {
    do {
     flag=gtube(10,5,comm_nm,comm_name);
   switch(flag) {
   case 0 : if(pfield(comm_nm,comm_name) != 0)
      prmsg(2,23,ch14);
            break;
   case -2 : setcook();
   case -3 : break;
   }
  rc=yorn(ch5);
} while(rc==1); /* yes */
  }
  ulockrec(comm);
  setloc(comm,addr);
  if(rc=yorn(ch17)==1)
    { setcook();
     add_comm();
   }
  else
   { setcook();
     conference();
  }
/* else */
/* add comm */
}

de1_comm()
{ short key;
  int flag;
int rc;

mv_cur(1,0,3); eram_ln();
prmp(1,3,head6); prmp(80,3,head1);
mv_cur(1,0,4); clear();
prmp(3,4,ch19);
prmp(3,6,ch1);
do {
    flag=gtube(21,8,comm_id,&key);
    switch(flag) {
    case 0 : break;
    case -2 : setcook();
              conference();
    break;
    case -3 : break;
    }
} while (flag == -3);
if(acckey(commI,key)!==0)
{
    prmsg(2,23,ch8);
    if(rc==yorn(ch20)==1) /* yes */
    { setcook();
      del_comm();
    }
    else {
      setcook();
      conference();
    }
} else {
    mv_cur(1,0,4); clear();
    prmp(3,5,ch19);
    prmp(3,6,ch1); pdata(21,6,comm_id);
    prmp(3,7,ch2); pdata(10,7,comm_id);
    if((rc==yorn(ch22))==1) /* yes*/
    { if(lockrec(commI)==0)
        { prmsg(2,23,ch4);
          setcook();
          del_comm();
        }
    else {
        if(delete(commI)==0)
        { prmsg(2,23,ch21);
          setcook();
          conference();
        }
        ulockrec(commI);
      }
    }
    if((rc==yorn(ch20))==1) /* yes*/
    { setcook();
      del_comm();
    }
  else {
    setcook();
    conference();
}
Listing of cm.c

166:
167: } /* else */
168: */ del conf */
169:
170: mod_comm()
171: { short select;
172:     short flag;
173:     short ret,rc;
174:     mv_cur(1,0,3); eras_ln();
175:     prmp(1,3,head7); prmp(60,3,head1);
176:     mv_cur(1,0,4); clear();
177:     prmp(5,4,chr23);
178:     prmp(5,5,chr7);
179:     prmp(5,6,chr8);
180:     prmp(5,8,chr9);
181:     switch(gtube(13,8,con,&select)) {
182:     case -2 : setcook();
183:             conference();
184:             break;
185:     }
186:     switch(select)
187:     { case 1 : { if(seqacc(comm,first)!=0)
188:             { prtnsg(2,23,chr12);
189:                 setcook();
190:                 conference();
191:             }
192:             else
193:             { do
194:                 { mv_cur(1,0,4); clear();
195:                     prmp(3,4,chr1); pdata(21,4,comm_id);
196:                     prmp(3,5,chr2); pdata(10,5,comm_nm);
197:                     flag=yorn(chr24);
198:                     switch(flag)
199:                     { case -2 : setcook();
200:                         mod_comm();
201:                         break;
202:                     case 0 : break;
203:                     case 1 : setcook();
204:                         get_mod_comm();
205:                         more_mod_comm();
206:                         break;
207:                     default :;
208:                     }
209:                     while(seqacc(comm,next)!=0);
210:                     prtnsg(40,23,chr10);
211:                     setcook();
212:                     more_mod_comm();
213:                 } /* else */
214:             } /* case 1 */
215:             break;
216:             case 2 : setcook();
217:             mod_comm_direct();
218:             get_mod_comm();
219:             more_mod_comm();
220:             break;
221:
222:             } /* else */
223:
224:             break;
225:         }
226:     case 2 : setcook();
227:         mod_comm_direct();
228:         get_mod_comm();
229:         more_mod_comm();
230:         break;
231:
232:         break;
Listing of cm.c

221:    default : prtmsg(2,23,ch3);
222:    setcook();
223:    mod_comm();
224:    } /* case */
225:    } /* mod comm */
226:  
227:  get_mod_comm()
228:  {
229:    char cmm_name[60];
230:    int rt,rcc;
231:    int flag;
232:    if(lockrec(comm) !=0)
233:    {
234:      prtmsg (2,23,ch4);
235:      setcook();
236:      conference();
237:    }
238:    else {
239:      flag=gtube(10,5,cmm_nm,cmm_name);
240:      switch(flag)
241:      { case 0 : if(pfield(commNm,cmm_name) !=0)
242:                  prtmsg(2,23,ch14);
243:                  break;
244:                  case -2 : setcook();
245:                  conference();
246:                  break;
247:                  case -3 : break;
248:                }
249:      rt=yorn(ch5);
250:      } while (rt == 1); /* yes */
251:      unlockrec(comm);
252:    }
253:    }
254:    }
255:    { int flag;
256:      flag=yorn(ch25);
257:      switch(flag) {
258:      case -2 : setcook();
259:                  conference();
260:                  break;
261:      case 0 : setcook();
262:                  conference();
263:      case 1 : setcook();
264:      mod_comm_direct();
265:      get_mod_comm();
266:      more_mod_comm();
267:      break;
268:      default : break;
269:    }
270:  }
271:  
272:  mod_comm_direct()
273:  {
274:    short key;
275:    int flag;
276:    mv_cur(1,0,3); eras_in();

-123-
Listing of cm.c

Listing of cm.c

prmp(1,3,head7); prmp(60,3,head1);
mv_cur(1,0,4); clear();
prmp(3,4,ch28);
prmp(3,8,ch1);
do {
flag=gtube(21,6,comm_id,&key);
switch(flag) {
case 0 : break;
case -2 : conference();
break;
case -3 : break;
}
}while (flag == -3);
if(acckey(comm,&key) !=0)
{ prtmag(2,23,ch6);
setcook();
more_mod_comm();
}
else
{ mv_cur(1,0,4); clear();
prmp(3,4,ch1); pdta(21,4,comm_id);
prmp(3,5,ch2); pdta(10,5,comm_nm);
}
inq_comm();
short select;
short flag;
short ret,rc;
mv_cur(1,0,3); eras_ln();
prmp(1,3,head8); prmp(1,3,head1);
mv_cur(1,0,4); clear();
prmp(5,4,ch27);
prmp(5,5,ch7);
prmp(5,6,ch8);
prmp(5,8,ch9);
switch(gtube(13,8,p1_id,&select)) {
case -2 : setcook();
conference();
break;
}
switch(select)
case 1 : { if(seqacc(comm,first)!=0)
{ prtmag(2,23,ch12);
setcook();
.inq_comm();
}
else {
do {
mv_cur(1,0,4); clear();
prmp(3,4,ch1); pdta(21,4,comm_id);
prmp(3,5,ch2); pdta(10,5,comm_nm);
flag=yorn(ch28);
switch(flag) {
case 0 :;

-124-
case -2 : setcook();
    inq_comm();
    break;
    case 1 : break;
    default :;
}
    } while(seqacc(comm,next)==0);
    prmsg(40,23,ch10);
    setcook();
    more_inq_comm();
    } */ else */
    } /* case 1 */
    break;
    case 2 : setcook();
    inq_comm_direct();
    more_inq_comm();
    break;
    default : prmsg(2,23,ch3);
    setcook();
    inq_comm();
    } */ case */
    } /* inq_comm */

more_inq_comm()
{ int flag;
    flag=yorn(ch29);
    switch(flag) {
    case -2 : setcook();
    conference();
    break;
    case 0 : setcook();
    conference();
    break;
    case 1 : setcook();
    inq_comm_direct();
    more_inq_comm();
    break;
    default : break;
    }
}
inq_comm_direct()
{ short key;
    int flag,rc;
    mv_cur(1,0,4); clear();
    prmp(3,4,ch30);
    prep(3,6,ch1);
    do {
    flag=gtube(21,6,comm_id,&key);
    switch(flag) {
    case 0 : break;
    case -2 : setcook();
    conference();
    break;
    case -3 : break;
    }
Listing of cm.c

Page 8

388: } while (flag == -3);
387: if(acckey(comm, &key) != 0)
388: { prtmag(2, 23, ch6);
389:    seckek();
390:    more_1ng_comm();
391: }
392: else
393: { mv_cur(1, 0, 4); clear();
394:    prmp(3, 6, ch1); pdata(21, 6, comm_id);
395:    prmp(3, 7, ch2); pdata(10, 7, comm_nm);
396: }
397: }
398: }
399: report_comm()
400: { short select;
401:    mv_cur(1, 0, 3); eras_ln();
402:    prmp(1, 3, head0); prmp(60, 3, head1);
403:    mv_cur(1, 0, 4); clear();
404:    prmp(5, 4, ch35);
405:    prmp(5, 5, ch36);
406:    prmp(5, 7, ch8);
407:    switch(gtube(14, 7, pl_id, &select)) {
408:     case -2 : setcook();
409:         conference();
410:         break;
411:     }
412:     switch(select) {
413:     case 1 : seckek();
414:         scrf_comm();
415:         break;
416:     case 2 : seckek();
417:         prt1_comm();
418:         report_comm();
419:         break;
420:     default : prtmag(2, 23, ch3);
421:         setcook();
422:         report_comm();
423:         break;
424:     }
425: }
426: }
427: scrf_comm()
428: { short select;
429:    int row=5;
430:    if(seqacc(comm, first) != 0)
431:    { prtmag(2, 23, ch12);
432:        setcook();
433:        report_comm();
434:    }
435:    else {
436:        do
437:        { if((row==5)||(row>23))
438:            { mv_cur(1, 0, 4);
439:                clear();
440:                prmpv(3, 3, ch32);
441:                }
```c
441:     pmprv(25,3,ch33);
442: }
443:     pdata(3,row,comm_id);
444:     pdata(8,row++,comm_nm);
445:     if(row>22)
446:     { pmprv(55,23," ----->");
447:         switch(gtube(81,23,pl_id,&select)) {
448:             case -2 : setcook();
449:                 report_comm();
450:                 break;
451:             case -3 : { row=5;
452:                 mv_cur(1,0,5); clear();
453:                 break;
454:             }
455:         }
456:     } while(seqacc(comm,next) == 0);
457:     prmsg(40,23,ch10);
458:     setcook();
459:     report_comm();
460: }
461: }
462: prt1_comm()
463: }
464: { int fd;
465:     if((fd=fopen()) == -1)
466:     { printf("pipe to printer could not created");
467:         setcook();
468:         report_comm();
469:     }
470: }
471: else {
472:     oblank();
473:     prstr(ch31,40); flush(fd);
474:     prstr(ch34,40); flush(fd);
475:     prstr(ch32,24); flush(fd);
476:     prstr(ch33,40); flush(fd);
477:     flush(fd);
478:     if(seqacc(comm,first) != 0)
479:     { prmsg(2,23,"Sequential failed on first record");
480:         setcook();
481:         report_comm();
482:     }
483:     else {
484:     do
485:     { odata(23,comm_id);
486:       odata(30,comm_nm);
487:       flush(fd);
488:       } while(seqacc(comm,next) == 0);
489:     flush(fd);
490:     prstr(ch10,60);
491:     flush(fd);
492:     flush(fd);
493:     }
494: }
495: close(fd);
```
1: \#include "./def/file.h"
2: \#include<stdio.h>
3: extern char head1[],head5[],head8[],head7[],head6[],head9[];
4: char c1[ ] = "....",.[.];
5: char c12[ ] = ".......");
6: char c13[ ] = ".......");
7: char c14[ ] = ".......");
8: char c15[ ] = ".......");
9: char c16[ ] = ".......");
10: char c17[ ] = ".......");
11: char c18[ ] = ".......");
12: char c19[ ] = ".......");
13: char c20[ ] = ".......");
14: char c21[ ] = ".......");
15: char c22[ ] = ".......");
16: char c23[ ] = ".......");
17: char c24[ ] = ".......");
18: char c25[ ] = ".......");
19: char c26[ ] = ".......");
20: char c27[ ] = ".......");
21: char c28[ ] = ".......");
22: char c29[ ] = ".......");
23: char c30[ ] = ".......");
24: char c31[ ] = ".......");
25: char c32[ ] = ".......");
26: char c33[ ] = ".......");
27: char c34[ ] = ".......");
28: char c35[ ] = ".......");
29: char c36[ ] = ".......");
30: char c37[ ] = ".......");
31: char c38[ ] = ".......");
32: char c39[ ] = ".......");
33: char c40[ ] = ".......");
34: char c41[ ] = ".......");
35: struct concl_key
36: { short com_id;
37: short cnfno;
38: short cnfyy;
39: short cno;
Listing of cl.c

53: }
54:
55: add_concl() /* add conclusion record */
56: { struct concl_key key;
57: /* struct conf_key kcf;*/
58: char cl_subj[150];
59: short conid;
60: long conref=0;
61: char cncl[256];
62: short buf_cmm_id,number;
63: long addr;
64: int flag,check;
65: int rc,rcn;
66: mv_cur(1,0,3); erase_ln();
67: prmp(1,3,head5); prmp(60,3,head1);
68: mv_cur(1,0,4); clear();
69: prmp(3,4,c12);
70: prmp(3,5,c13);
71: prmp(3,6,c11);
72: do {
73: flag=gtube(20,4,c1com_id,&key.ccom_id);
74: switch(flag) {
75: case 0 : { /*if(acckey(cmm,&key.ccom_id) != 0)
76: { prmsg(2,23,c112);
77: setcook();
78: add_concl();
79: break;
80: */
81: buf_cmm_id=key.ccom_id;
82: }
83: break;
84: case 2 : setcook();
85: conference();
86: break;
87: case 3 : break;
88: }
89: flag=gtube(20,5,c1cfn_no,&key.cnfno);
90: switch(flag) {
91: case 0 : mv_cur(1,22,5); printf("/25");
92: break;
93: case 2 : setcook();
94: conference();
95: break;
96: case 3 : mv_cur(1,22,5); printf("/25");
97: break;
98: }
99: flag=gtube(25,5,c1cfn_yy,&key.cnfyy);
100: switch(flag) {
101: case 0 : break;
102: case 2 : setcook();
103: conference();
104: break;
105: case 3 : break;
106: }
107: flag=gtube(14,6,conc1_no,&key.cnlo);

-130-
switch(flag) {
  case 0 : number=key.c1no;
          break;
  case -2 : setcook();
          conference();
          break;
  case -3 : break;
}
} while (flag==3);
if(acckey(comm,&key.ccom_id) != 0)
  { prtmsg(2,23,c112);
    setcook();
    add_concl1();

    } /* if(acckey(comm,&key.ccom_id) != 0)

    prtmsg(2,23,c142);
    else */
  do {
  if(addrac(concl1,&key) != 0)
    { if((rc=ynor(cm20)==1))
      { setcook();
        add_concl1();
      }
    else
      { setcook();
        conference();
      }
    }
  else {
    loc(concl1,&addr);
    if(lockrec(concl1) != 0)
      { prtmsg(2,23,c17);
        setcook();
        conference();
      }
    else {
      do {
        mv_cur(1,14,8);
        printf("%02d",number);
        mv_cur(1,21,8);
        prmp(3,8,c14);
        prmp(3,12,c15);
        flag=gtube(21,8,subj,c1_subj);
        switch(flag) {
          case 0 : if(pffield(subj,c1_subj) != 0)
                      prtmsg(2,23,c19);
                      break;
          case -2 : setcook();
                    conference();
                    break;
          case -3 : mv_cur(1,21,12);
                    break;
        }
        flag=gtube(21,12,con,&conid);
switch(flag) {
    case 0 : if((conid<0)||(conid>2))
             { pmessage(1,23,c16);
               pfield(con,"");
             }
             else
             pfield(con,&conid);
             break;
    case -2 : setcook();
             conference();
             break;
    case -3 : conid=2;break;
}
if((conid==0)||(conid==1))
{ conref++;
  pfield(con_n,&conref);
}
else
    conref=0;
if(conid==2)
{ short skey;
  skey=conref;
  if(addrsub(sub_con,&skey) != 0)
    pmessage(1,23,c120);
  else
    { prmp(3,13,c15_1);
      flag=stube(18,13,conclus,cnc1);
      switch(flag) {
        case 0 : if(pfield(conclus,cnc1) != 0)
                 pmessage(2,23,c19);
                 break;
        case -2 : setcook();
                 conference();
                 break;
        case -3 : break;
      }
    }
  }
  while(rc==1);
  ulockrec(concl);
  setloc(concl,addr);
}
if(rc=yorn(c135)==1)
{ mv_cur(1,1,8); clear();
  key.cnln=key.cnln+1;
  number=key.cnln;
  key.ccom_id=buf_cmm_id;
  key.cnfnm=key.cnfnm;
  key.cnfy=ykey.cnfy;
}
while (rc == 1);
*/*/
if(rc=yorn(c121)==1)
Listing of cl.c

218: { setcook();
219:     add_concl();
220: }
221: else
222: { setcook();
223:     conference();
224: }
225: }
226:
227: del_concl()
228: { struct concl_key key;
229:     int flag,rc;
230:     short number;
231:     mv_cur(1,0,3); eras_ln();
232:     prmp(1,3,head8); prmp(60,3,head1);
233:     mv_cur(1,0,4); clear();
234:     prmp(3,4,c123);
235:     prmp(3,6,c11);
236:     prmp(3,7,c12);
237:     prmp(3,8,c13);
238:     do {
239:         flag=gtube(20,6,clcom_id, &key.ccom_id);
240:         switch(flag) {
241:             case 0 : break;
242:             case -2 : setcook();
243:                     conference();
244:                     break;
245:             case -3 : break;
246:         }
247:         flag=gtube(20,7,clcf_no, &key.cnfnm);
248:         switch(flag) {
249:             case 0 : mv_cur(1,22,7); printf("/25");
250:                     break;
251:             case -2 : setcook();
252:                     conference();
253:                     break;
254:             case -3 : mv_cur(1,22,7); printf("/25");
255:                     break;
256:         }
257:         flag=gtube(25,7,clcf_yy, &key.cnfyy);
258:         switch(flag) {
259:             case 0 : break;
260:             case -2 : setcook();
261:                     conference();
262:                     break;
263:             case -3 : break;
264:         }
265:         flag=gtube(14,8,concl_no, &key.cnlo);
266:         switch(flag) {
267:             case 0 : number=key.cnlo;
268:                     break;
269:             case -2 : setcook();
270:                     conference();
271:                     break;
272:             case -3 : break;

-133-
Listing of c1.c

273: } } while ((flag == -3));
274: if (number != 0)
275: { if (ackkey(conc1, &key) != 0)
276: { prmsg(2, 23, c112);
277: setcook();
278: del_conc1();
279: }
280: }
281: else {
282: if ((rc = yorn(conc1)) == 1)
283: { if (lockrec(conc1) != 0)
284: { prmsg(2, 23, c117);
285: setcook();
286: del_conc1();
287: }
288: else {
289: { if (delete(conc1) != 0)
290: { prmsg(2, 23, c126);
291: setcook();
292: conference();
293: }
294: unlockrec(conc1);
295: }
296: }
297: }
298: }
299: }
300: else {
301: if (seqacc(conc1, first) == 0)
302: prmsg(2, 23, c118);
303: else {
304: do {
305: key.clno = key.clno + 1;
306: if (ackkey(conc1, &key) == 0)
307: { if ((rc = yorn(conc1)) == 1)
308: { if (lockrec(conc1) != 0)
309: prmsg(2, 23, c117);
310: else {
311: { if (delete(conc1) != 0)
312: prmsg(2, 23, c126);
313: unlockrec(conc1);
314: }
315: }
316: }
317: } while (seqacc(conc1, next) == 0);
318: }
319: }
320: if ((rc = yorn(conc1)) == 1)
321: { setcook();
322: del_conc1();
323: }
324: else {
325: setcook();
326: conference();
327: }

-134-
Listing of cl.c

328: }
329:
330: mod_concl()
331: {
332:   short select;
333:   short flag,bconid,bconref;
334:   short ret,rc;
335:   mv_cur(1,0,3); eras_ln();
336:   prmp(1,3,head7); prmp(60,3,head1);
337:   mv_cur(1,0,4); clear();
338:   prmp(5,4,c127);
339:   prmp(5,5,c113);
340:   prmp(5,6,c114);
341:   prmp(5,8,c115);
342:   switch(gtube(13,8,con,&select)) {
343:   case -2 : setcook();
344:             conference();
345:             break;
346:   case -3 : setcook();
347:             mod_concl();
348:             break;
349:   }
350:   switch(select)
351:   {
352:     case 1 : { if(seqacc(concl,first)=0)
353:         {
354:           prmsg(2,23,c18);
355:           setcook();
356:           conference();
357:         }
358:       }
359:     else
360:       {
361:         { mv_cur(1,1,2); clear();
362:           prmp(3,4,c13); pdata(20,4,clcf_no);
363:           prmp(3,5,c12); pdata(20,5,clcom_id);
364:           prmp(3,6,c11); pdata(14,6,conc1_no);
365:           prmp(3,7,c14); pdata(21,7,sub);
366:           prmp(3,11,c15); pdata(18,11,con);
367:           prmp(31,11,"\n"); pdata(26,11,con_n);
368:           prmp(32,11,"]");
369:           sfield(con_n,&bconref);
370:           if(bconref=0)
371:           { prmp(3,12,c15_1);
372:             mv_cur(1,24,12);
373:             ackey(sub_con,&bconref);
374:             pdata(24,12,conclus);
375:           }
376:           flag=yorn(c128);
377:           switch(flag) {
378:             case -2 : setcook();
379:                       mod_concl();
380:             break;
381:             case 0 : break;
382:             case 1 : setcook();
383:                       get_mod_concl();
384:             break;
385:             default ;
386:           } /* case */
Listing of cl.c

383:     } while(seqacc(concl,next)==0);
384:     prmsg(40,23,c117);
385:     setcook();
386:     more_mod_concl();
387:     } /* else =*/
388: }
389: /* case 1 =*/
400: break;
401: case 2 : setcook();
402: mod_concl_direct();
403: get_mod_concl();
404: more_mod_concl();
405: break;
406: default : prmsg(2,23,c16);
407: setcook();
408: mod_concl();
409: break;
410: */
411: /* case */
412: } /* mod concl */
413: get_mod_concl();
414: { struct concl_key_key;
415: /* struct cf_key kcf;*/
416: char cl_subj[150];
417: short conid;
418: long conref=0;
419: char cnc1[250];
420: short buf_cmm_id,number;
421: int r,rc;
422: int flag,check;
423: if(lockrec(concl)!=0)
424: { prmsg (2,23,c17);
425:      setcook();
426:      conference();
427:  }
428: else {
429: do {
430: flag=gtube(20,4,clcom_id,&key,ccom_id);
431: switch(flag) {
432: case 0 : { if(acckey(comm,&key.ccom_id) != 0)
433: { prmsg(2,23,c112);
434:      setcook();
435:      mod_concl();
436:      break;
437:      } }
438: pfieid(clcom_id,&key.ccom_id);
439: buf_cmm_id=key.ccom_id;
440: }
441: break;
442: case -2 : setcook();
443: conference();
444: break;
445: case -3 : break;
446: }
447: flag=gtube(20,5,clf_no,&key.cnfno);
448: switch(flag) {
438: case 0 : pfield(clcf_no,&key.cnfno);
439:     mv_cur(1,22,5); printf("/25");
440:     break;
441: case -2 : setcook();
442:     conference();
443:     break;
444: case -3 : mv_cur(1,22,5); printf("/25");
445:     break;
446: }
447: flag=gtube(25,5,clf_yy,&key.cnfyy);
448: switch(flag) {
449: case 0 : pfield(clcf_yy,&key.cnfyy);
450:     break;
451: case -2 : setcook();
452:     conference();
453:     break;
454: case -3 : break;
455: }
456: flag=gtube(14,6,con1_no,&key.cnlo);
457: switch(flag) {
458: case 0 : number=key.cnlo;
459:     break;
460: case -2 : setcook();
461:     conference();
462:     break;
463: case -3 : break;
464: }
465: flag=gtube(21,7,subj,cl_subb);
466: switch(flag) {
467: case 0 : if(pfield(subj,cl_subb)!=0)
468:     prmsg(2,23,cl19);
469:     break;
470: case -2 : setcook();
471:     conference();
472:     break;
473: case -3 : mv_cur(1,18,11);
474:     break;
475: }
476: switch(gtube(18,12,con,&conid)) {
477: case 0 : if((conid<0)||(conid>2))
478:     { prmsg(1,23,cl16);
479:     pfield(con, ");
480:     }
481: else
482:     pfield(con,&conid);
483:     break;
484: case -2 : setcook();
485:     conference();
486:     break;
487: case -3 : mv_cur(1,18,13);
488:     break;
489: }
490: if((conid==0)||(conid==1))
491: { conref++;
492: pfield(con_n,&conref);
483:    }
484:  else
485:    conref=0;
486:  if(conidl==2)
487:    { short skey;
488:      skey=conref;
489:      if(addr(pch,sub_eon,skey) != 0)
490:        prtmag(1,23,c120);
491:    else
492:      { flagstube(18,13,conclus,cnc1);
493:        switch(flag) {
494:          case 0  : if(pfield(conclus,cnc1) != 0)
495:            prtmag(2,23,c19);
496:            break;
497:          case -2 : setcock();
498:            conference();
499:            break;
500:          case -3 : break;
501:        }
502:      }
503:    }
504:    } while (rt == 1); /* yes */
505:  unlockre(concl);
506:  }
507: }
508: }
509: }
510: more_mod_concl()
511: { int flag;
512:  flag=ynorn(c129);
513:  switch(flag) {
514:    case -2 : setcock();
515:      conference();
516:      break;
517:    case 0  : setcock();
518:      conference();
519:    case 1  : setcock();
520:      mod_concl_direct();
521:      get_mod_concl1();
522:      more_mod_concl();
523:      break;
524:    default: break;
525:  }
526: }
527: }
528: mod_concl_direct()
529: { struct concl_key key;
530:  short bconid,bconref,number;
531:  int flag;
532:  mv_cur(1,0,3); eras_ln();
533:  prmp(1,3,head7); prmp(0,3,head1);
534:  mv_cur(1,0,4); clear();
535:  prmp(3,4,c130);
536:  prmp(3,8,c12);
537:  prmp(3,7,c13);
prmp(3,6,c11);

do {
    switch(flag) {
    case 0 : break;
    case -2 : setcook();
    conference();
    break;
    case -3 : break;
    }

    switch(flag) {
    case 0 : printf("/25");
    break;
    case -2 : setcook();
    conference();
    break;
    case -3 : printf("/25");
    break;
    }

    switch(flag) {
    case 0 : break;
    case -2 : setcook();
    conference();
    break;
    case -3 : break;
    }

    switch(flag) {
    case 0 : number=keyclno;
    break;
    case -2 : setcook();
    conference();
    break;
    case -3 : mv_cur(1,20,6);
    break;
    }

} while (flag == -3);

if(ackey(concl,&key) != 0) {
    prnmsg(2,23,c112);
    setcook();
    mod_concl_direct();
}

else {
    mv_cur(1,1,2); clear();
    prmp(3,6,c11); pdata(14,6,concl_no);
    prmp(3,5,c12); pdata(20,5,clcom_id);
    prmp(3,4,c13); pdata(20,4,clcf_no);
    prmp(3,7,c14); pdata(21,7,subj);
    prmp(3,11,c15); mv_cur(1,18,11);
    gfield(conl,&bconl);
    switch(bconl) {
    case 1 : printf("o.k.");
    break;
Listing of cl.c
Last Update Fri Dec 04 10:55:15 1992
Listed Time Fri Dec 04 13:16:51 1992

603:     case 2 : printf("no.k.");
604:     break;
605:     case 0 : printf("other");
606:     break;
607: }
608: preq(3,12,c15_1);
609: gfield(con_n,&bconref); mv_cur(1,24,12);
610: acckey(sub_con,&bconref);
611: pdata(24,12,conclus);
612: }
613: }
614:
615:  inq_concl()
616:  { short select;
617:    short flag,bconid,bconref;
618:    short ret,rc;
619:    mv_cur(1,0,3); eras_in();
620:    prmp(1,3,head8); prmp(60,3,head1);
621:    mv_cur(1,0,4); clear();
622:    prmp(5,4,c127_1);
623:    prmp(5,5,c113);
624:    prmp(5,6,c114);
625:    prmp(5,8,c115);
626:    switch(gtube(13,8,con,&select)) {
627:      case -2 : setcook();
628:        conference();
629:        break;
630:      case -3 : setcook();
631:        inq_concl();
632:        break;
633:    } }
634:    switch(select)
635:    { case 1 : { if(seqacc(concl,first)!>0)
636:       { prtmag(2,23,c18);
637:         setcook();
638:         conference();
639:         }
640:       else {
641:         do
642:         { mv_cur(1,1,2); clear();
643:           prmp(3,6,c11); pdata(14,6,conc1_no);
644:           prmp(3,5,c12); pdata(20,5,clcom_id);
645:           prmp(3,4,c13); pdata(20,4,clcf_no);
646:           prmp(3,7,c14); pdata(21,7,subj);
647:           prmp(3,11,c15); mv_cur(1,18,11);
648:           gfield(con,&bconid);
649:           switch(bconid) {
650:             case 1 : printf("o.k.");
651:               break;
652:             case 2 : printf("no.k.");
653:               break;
654:             case 0 : printf("other");
655:               break;
656:           } }
657:           gfield(con_n,&bconref);
if(bconref != 0)
  { prmp(3,12,c15_1);
    mv_cur(1,24,12);
    acckey(sub_con,&bconref);
    pdata(24,12,conclus);
  }
flag=yorn(c131);
switch(flag) {
  case 0 : case -2 : setcook();
  inq_concl();
  break;
  case 1 : break;
  default : ;
  } /* case */
  } while(seqacc(concl,next)==0);
prtmag(40,23,c117);
setcook();
more_inq_concl();
} /* else */
} /* case 1 */
break;
case 2 : setcook();
inq_concl_direct();
more_inq_concl();
break;
default : prtmag(2,23,c16);
setcook();
inq_concl();
break;
} /* case */
} /* inq concl */
more_inq_concl()
{ int flag;
flag=yorn(c132);
switch(flag) {
  case -2 : setcook();
  conference();
  break;
  case 0 : setcook();
  conference();
  case 1 : setcook();
  inq_concl_direct();
  more_inq_concl();
  break;
  default : break;
  }
}
inq_concl_direct()
{ struct concl_key key;
short number,bconid,bconref;
int flag;
mv_cur(1,0,3);-erase_in();
Listing of clc

713: prmp(1,3,head5); prmp(0,3,head1);
714: mv_cur(1,0,4); clear();
715: prmp(3,4,c133);
716: prmp(3,8,c12);
717: prmp(3,7,c13);
718: prmp(3,6,c11);
719: do {
720: \begin{verbatim}
flag=gtube(20,6,c1com_id,&key.ccom_id);
switch(flag) {
  case 0 : break;
  case 2 : setcook();
  case 4 : conference();
  case 3 : break;
}
flag=gtube(20,7,c1cf_no,&key.cnfno);
switch(flag) {
  case 0 : mv_cur(1,22,7); printf("/25");
  break;
  case 2 : setcook();
  case 4 : conference();
  case 3 : mv_cur(1,22,7); printf("/25");
  break;
}
flag=gtube(25,7,c1cf_yy,&key.cnfyy);
switch(flag) {
  case 0 : break;
  case 2 : setcook();
  case 4 : conference();
  case 3 : break;
}
flag=gtube(14,8,ccon1_no,&key.c1no);
switch(flag) {
  case 0 : numberskey.c1no;
  break;
  case 2 : setcook();
  case 4 : conference();
  break;
  case 3 : mv_cur(1,20,6);
  break;
}
} while (flag == -3);
758: if(acckey(con1,&key) != 0)
759: { prtmag(2,23,c112);
760: setcook();
761: inq_con1_direct();
762: }
763: else
764: { mv_cur(1,1,2); clear();
765: prmp(3,8,c11); pdata(14,8,ccon1_no);
766: prmp(3,5,c12); pdata(20,5,c1com_id);
767: prmp(3,4,c13); pdata(20,4,c1cf_no);
768: prmp(3,7,c14); pdata(21,7,subj);

-142-
768:     prpm(3,11,c15); mv_cur(1,18,11);
769:     gfield(con,&bconid);
770:     switch(bconid) {
771:         case 1 : printf("o.k.");
772:                 break;
773:         case 2 : printf("no.k.");
774:                 break;
775:         case 0 : printf("other");
776:                 break;
777:     }
778:     prpm(3,12,c15_1);
779:     gfield(con_n,&bconref); mv_cur(1,24,12);
780:     acckey(sub_con,&bconref);
781:     pdata(24,12,conclua);
782: }
783: }
784:
785: report_concl()
786: { short select;
787:     mv_cur(1,0,3); eras_ln();
788:     prpm(1,3,head9); prpm(b0,3,head1);
789:     mv_cur(1,0,4); clear();
790:     prpm(b4,c138_1);
791:     prpm(b5,c137);
792:     prpm(b5,c138);
793:     prpm(b5,c139);
794:     prpm(b5,10,c115);
795:     switch(gtube(14,10,con,&select)) {
796:     case -2 : setcook();
797:                 conference();
798:                 break;
799:     case -3 : setcook();
800:                 report_concl();
801:                 break;
802: }
803: switch(select) {
804:     case 1 : setcook();
805:                 scr1_concl();
806:                 break;
807:     case 2 : setcook();
808:                 scr2_concl();
809:                 break;
810:     case 3 : setcook();
811:                 scr3_concl();
812:                 break;
813:     default: prtmag(2,23,c18);
814:                 setcook();
815:                 report_concl();
816:                 break;
817: }
818: }
819:
820: scr1_concl()
821: { struct concl_key key;
822:   int flag,check;
short number, buf_cmm_id, bconid, bconref;
int rows=4;
short select;
mv_cur(1,0,4); clear();
prmp(3,7,c12);
prmp(3,8,c13);
do {
  key.cnos=0;
  number= key.cnno;
  flag = gtube(20,4, clcom_id, &key.ccm_id);
  switch(flag) {
    case 0 : break;
    case -2 : setcook();
    conference();
    break;
    case -3 : break;
  }
  flag = gtube(20,5, ctcf_no, &key.cnfno);
  switch(flag) {
    case 0 : mv_cur(1,22,5); printf("/25");
    break;
    case -2 : setcook();
    conference();
    break;
    case -3 : mv_cur(1,22,5); printf("/25");
    break;
  }
  flag = gtube(25,5, clcf_yy, &key.cnfyy);
  switch(flag) {
    case 0 : break;
    case -2 : setcook();
    conference();
    break;
    case -3 : break;
  }
  flag = gtube(14,7, concl_no, &key.cnno);
  switch(flag) {
    case 0 : number = key.cnno;
    break;
    case -2 : setcook();
    conference();
    break;
    case -3 : mv_cur(1,20,6);
    break;
  }
} while (flag == -3);
if(seqacc(concl, first) != 0) {
  prmsg(2,23,c18);
  setcook();
  report_concl();
}
else {
  do {
    key.cnno = key.cnno+1;
    if(acckey(concl, &key) != 0)
break;
else {
  field(clcom_id,&buf_comm_id);
  if(acckey(comm,&buf_comm_id) != 0)
    check=0;
  else
    check=1;
  if((row==4) && (row>23))
    {
      mv_cur(1,1,2); clear();
      prmp(3,row,c141);
      if(check==0)
        prmp(20,row++,c140);
      else
        pdata(20,row++,comm_nm);
      prmp(3,row,c13); pdata(28,row++,clcf_no);
      prmp(3,row,--------------------------------------------);
    }
  prmp(3,row,c111); pdata(14,row++,concl_no);
  prmp(3,row,c14); pdata(21,row,subj);
  row=row+4;
  prmp(3,row,c15); mv_cur(1,18,row++);
  field(con,&bcond);
  switch(bcond) {
    case 1 : printf("o.k."); break;
    case 2 : printf("no.k."); break;
    case 0 : printf("other"); break;
  }
  prmp(3,row,c15_1);
  field(con_n,&bcondref); mv_cur(1,24,row);
  acckey(sub_con,&bcondref);
  pdata(24,row,conclues);
  row=row+5;
  if(row>22)
    {
      prmp(55,23," ---->");
      switch(gtube(81,23,con,&select)) {
        case -2 : setcook();
        report_concl();
        break;
        case -3 : { row=4;
          mv_cur(1,0,4); clear();
          break;
        }
        case -1 :
      }
    }
  while (seqacc(concl,next) == 0);
  prtmag(55,23,c117);
  setcook();
  report_concl();
}
int flag;
short number;
int check2;
char buf1 subj[150];
short buf cmm id;
int row=4;
short select,bconid,bconref;
int str_len;
char buffer[150];
int i,check;
mv_cur(1,0,4); clear();
prmp(3,6,c14);
mv_cur(1,12,6);
i=0;
while ((buffer[i]=getchar()) ! = 'n')
i++;
str_len = i;
if (seqacc(concl,first) i = 0)
{ prmsg(2,23,c10);
setcook();
report_concl();
}
else
{
  gfield(subj,buf1 subj);
  if (i cvmp(buf1 subj,buffer,str_len)=1)
  { check2 = 1;
    gfield(clcomm_id, &buf cmm id);
    if (ackkey(comm, &buf cmm id) != 0)
      check = 0;
    else
      check = 1;
  }
  else
    check2 = 0;
  if (check2 = = 1)
  {
    if ((row = = 4) || (row > 23))
      { mv cur (1,0,4); clear();
        prmp(3,row,c141);
        if (check = = 0)
          prmp(25,row++,c140);
      }
      else
        pdata(25,row++,comm nm);
      prmp(3,row,c11); pdata(14,row++,concl_no);
    prmp(3,row,c14); pdata(21,row,subj);
    row = row + 4;
    prmp(3,row,c15); mv_cur(1,18,row++);
    gfield(comm,&bconid);
    switch (bconid) {
    case 1: printf("o.k."); break;
    case 2: printf("no.k."); break;
    case 0: printf("other"); break;
    }
  }
  prmp(3,row,c15_1);
  gfield(con_n,&bconref); mv_cur(1,24,row);
acckey(sub_con,&bconref);
pdata(24,row,concluir);
row=row+5;
if(row>22)
    {
prmprv(55,23,"---->");
    switch(gtube(81,23,con, &select))
    {
    case -2 : setcook();
           report_concl();
            break;
    case -3 :
    mv_cur(1,0,4); clear();
            break;
    }
else
    }
check2=0;
} while (seqacc(concl,next) == 0);
printf(55,23,c117);
setcook();
report_concl();
}
scr3_concl()
{ int flag;
  short number,bconid,bconref;
  int rc,check2;
  char buf1_consl[256];
  short buf_cmm_id;
  int row=4;
  short select;
  int str_len;
  char buffer[256];
  int i,check;
  mv_cur(1,1,2); clear();
  prmp(3,8,c18);
  mv_cur(1,18,8);
  i=0;
  while((buffer[i]=getchar()) != '\n')
  i++;
str_len=i;
  if(seqacc(concl,first) != 0)
    {
prtm(2,23,c18);
setcook();
report_concl();
    }
else {
    do {
    gfield(consl,buf1_consl);
    if(ivcmp(buf1_consl,buffer,str_len)==1)
    { check2=1;
    gfield(c1conl, &buf_cmm_id);
    if(acckey(comm, &buf_cmm_id) != 0)

-147-
check=0;
else
    check=1;
}
else
    check2=0;
if(check2==1)
{
    if((row==4)||(row>23))
    {
        mv_cur(1,0,4); clear();
    }
    prmp(3,row,c141);
    if(check==0)
       prmp(25,row++,c140);
    else
       pdata(25,row++,comm_n);
    prmp(3,row,c11); pdata(14,row++,conc1_no);
    prmp(3,row,c14); pdata(21,row,subj);
    row=row+4;
    prmp(3,row,c15); mv_cur(1,18,row++);
    gfield(con,&bconid);
    switch(bconid) {
    case 1 : printf("o.k."); break;
    case 2 : printf("no.k."); break;
    case 0 : printf("other"); break;
    }
    prmp(3,row,c15_1);
    gfield(con_n,&bconref); mv_cur(1,24,row);
    acckey(sub_con,&bconref);
    pdata(24,row,conc1);
    row=row+5;
    if(row>22)
    { prmprv(55,23," ---->");
        switch(gtube(81,23,con,&select)) { 
        case -2 : setcook();
        report_conc1();
        break;
        case -3 : { row=4;
        mv_cur(1,1,2); clear();
        break;
        }
        }
    } while (seqacc(con1,next) == 0);
prtmag(55,23,c117);
setcook();
report_conc1();
}
#include "def/file.h"
#include <stdio.h>
extern char head2[], warn2[], first1[], first2[], err[], head5[],
        head6[], head7[], head8[], head9[];
char p11[] = "... เลือกการจัดเก็บเป้าหมาย... ..."
char p12[] = "ฝ. และ [25.]"
char p13[] = "ใช่.หยุด ... ... ..."
char p14[] = "ประเภทของบันทึกเฉพาะ";

char p15[] = "... ... ... ... ... ... ...
char p16[] = "... ...
char p17[] = "... ...
char p18[] = "... ...
char p19[] = "... ...
char p20[] = "... ...
char p21[] = "... ...
char p22[] = "... ...
char p23[] = "... ...
char p24[] = "... ...
char p25[] = "... ...
char p26[] = "... ...
char p27[] = "... ...
char p28[] = "... ...
char p29[] = "... ...
char p30[] = "... ...
char p31[] = "... ...
char p32[] = "... ...
char p33[] = "... ...
char p34[] = "... ...
char p35[] = "... ...
char p36[] = "... ...
char p37[] = "... ...
char p38[] = "... ...
char p39[] = "... ...
char p40[] = "... ...
char p41[] = "... ...
char p42[] = "... ...
char p43[] = "... ...
char p44[] = "... ...
char p45[] = "... ...
struct pikey {
    short pldid;
    short plyear;
};
add_pol() /* add policy record */ { struct pikey key;
    char p1case[100];
    char plabstr[256];
    char p1index[10];
    long addr;
Listing of pl.c

53: int flag;
54: int rc,rcc;
55: mv_cur(1,0,3); erase_ln();
56: prmp(60,3,head2);
57: prmp(1,3,head5);
58: mv_cur(1,0,4); clear();
59: prmp(3,4,p11);
60: prmp(3,5,p12);
61: prmp(3,6,p13);
62: prmp(3,6,p14);
63: prmp(3,12,p15);
64: do {
65: flag=gtube(24,4,p1_id,&key.plid);
66: switch(flag) {
67: case 0 : break;
68: case -2 : setcook();
69: case -3 : mv_cur(1,10,5);
70: case -3 : break;
71: }
72: flag=gtube(10,5,p1_year,&key.pyear);
73: switch(flag) {
74: case 0 : break;
75: case -2 : setcook();
76: case -2 : science();
77: case -2 : break;
78: case -3 : break;
79: }
80: } while (flag == -3);
81: if(addrrec(policy,&key) != 0)
82: if(rcyorn(p119)==1))
83: setcook();
84: add_pol();
85: } else
86: setcook();
87: science();
88: }
89: else
90: loc(policy,Aaddr);
91: if(lockrec(policy) != 0)
92: prmsg(2,23,p17);
93: setcook();
94: science();
95: }
96: else {
97: do {
98: flag=gtube(12,8,p1_case,p1case);
99: switch(flag) {
100: case 0 : if(pfield(p1_case,p1case) != 0)
101: prmsg(2,23,p18);
102: break;
103: case -2 : setcook();
104: }
108:    science();
109:    break;
110:    case -3 : mv_cur(1,14,8);
111:        break;
112:    }
113:    flag=gtube(14,8,p1_abstr,plabstr);
114:    switch(flag) {
115:        case 0 : if(pfield(p1_abstr,plabstr) != 0)
116:            prmsg(2,23,p18);
117:            break;
118:        case -2 : setcook();
119:            science();
120:            break;
121:        case -3 : mv_cur(1,15,12);
122:            break;
123:        }
124:    flag=gtube(15,12,p1_idx,pidx);
125:    switch(flag) {
126:        case 0 : if(pfield(p1_idx,pidx) != 0)
127:            prmsg(2,23,p18);
128:            break;
129:        case -2 : setcook();
130:            science();
131:            break;
132:        case -3 : break;
133:        }
134:    rc=yorn(p19);
135:    } while(rc=1); /* yes */
136:    unlock(p1);
137:    msetloc(policy,addr);
138:    mv_cur(1,0,29); eras_in();
139:    if(rc=yorn(p120)==1)
140:        { setcook();
141:            add_pol();
142:        }
143:    }
144:    else
145:        { setcook();
146:            science();
147:        }
148:    } /* else */
149:    } /* add pol */
150:    
151:    del_pol();
152:    { struct plkey key;
153:    int flag;
154:    int rc;
155:    mv_cur(1,0,3); eras_in();
156:    prmp(1,3,head6); prmp(80,3,head2);
157:    mv_cur(1,0,4); clear();
158:    prmp(3,4,p122);
159:    prmp(3,6,p11);
160:    prmp(3,7,p12);
161:    do { 
162:        flag=gtube(24,6,p1_id,key,p1id);
163:        
-151-
switch(flag) {
    case 0 : break;
    case -2 : setcook();
    science();
    break;
    case -3 : mv_cur(1,10,7);
    break;
}
flag=gtube(10,7,p1_year,&key.pl_year);
switch(flag) {
    case 0 : break;
    case -2 : setcook();
    science();
    break;
    case -3 : break;
}
} while (flag == -3);
if(accekey(policy,key)!=0)
    { prtmag(2,23,p10);
      mv_cur(1,0,23); eras_ln();
      if(rc=yorn(p123)==1) /* yes */
        { setcook();
          del_pol();
        }
      else
        { setcook();
          science();
        }
    }
else {
    mv_cur(1,0,4); clear();
    prmp(3,4,p122);
    prmp(3,6,p11); pdata(24,6,p1_id);
    prmp(3,7,p12); prmp(8,7,"_m"); pdata(10,7,p1_year);
    prmp(3,8,p13); pdata(12,8,p1_case);
    prmp(3,10,p14); pdata(14,10,p1_abstr);
    prmp(3,14,p15); pdata(15,14,p1_idx);
    if((rc=yorn(p125))==1)
      { if(lockrec(policy)==0)
        { prtmag(2,23,p17);
          setcook();
          del_pol();
        }
        else
          { if(delete(policy)!=0)
            { prtmag(2,23,p124);
              setcook();
              science();
            }
            unlockrec(policy);
            }
        }
    if(rc=yorn(p123)==1) /*yes*/
      { setcook();
        del_pol();
    -152-
Listing of pl.c

218: }
219: else
220: { setcook();
221:  science();
222: }
223: } /* else */
224: } /* del pol */
225:
226: mod_pol()
227: { short select;
228:  short flag;
229:  short ret,rc;
230:  mv_cur(1,0,3); eraa_in();
231:  prmp(1,3,head7); prmp(80,3,head2);
232:  mv_cur(1,0,4); clear();
233:  prmp(5,4,p120);
234:  prmp(5,5,p111);
235:  prmp(5,6,p112);
236:  prmp(5,9,p113);
237:  switch(stube(13,8,p1_id,&select)) {
238:  case -2 : setcook();
239:  science();
240:  break;
241: }
242:  }
243: switch(select)
244: { case 1 : { if(seqacc(policy,first)==0) {
245:  { prmsg(2,23,p117);
246:   setcook();
247:   science();
248:  }
249:  else
250:  {
251:   { mv_cur(1,0,4); clear();
252:     prmp(3,4,p11); pdata(24,4,p1_id);
253:     prmp(3,5,p12); prmp(8,5,"25");pdata(10,5,p1_year);
254:     prmp(3,6,p13); pdata(12,6,p1_case);
255:     prmp(3,8,p14); pdata(14,8,p1_abstr);
256:     prmp(3,12,p15); pdata(15,12,p1_idx);
257:     flag=yorn(p127);
258:     switch(flag) {
259:       case -2 : setcook();
260:       mod_pol();
261:       break;
262:       case 0 : break;
263:       case 1 : setcook();
264:       get_mod_pol();
265:       break;
266:       default :;
267:     } /* case */
268: } while(seqacc(policy,next)==0);
269: prmsg(40,23,p116);
270: setcook();
271: more_mod_pol();
272: } /* case 1 */
273: } /* case 1 */

-153-
Listing of pl.c

break;
case 2 : setcook();
mod_pol_direct();
get_mod_pol();
more_mod_pol();
break;
default : prtmsg(2,23,p16);
setcook();
mod_pol();

} /* case */
} /* mod pol */

get_mod_pol()
{
struct p1key key;
char plcase[100];
char plbase[256];
char plindex[10];
int rt,rcc;
int flag;
if(lockrec(policy) !=0)
{
prtmsg(2,23,p17);
setcook();
mod_pol();
}
else {
do {
flag=gtube(24,4,pl_id,&key.plid);
switch(flag) {

break;
case -2 : setcook();
science();
break;
case -3 : mv_cur(1,10,5);
break;
}
flag=gtube(10,5,pl_year,&key.plyear);
switch(flag) {

break;
prtmsg(2,23,p18);
break;
case -2 : setcook();
science();
break;
case -3 : mv_cur(1,12,6);
break;
}

flag=gtube(12,6,pl_case,plcase);
switch(flag)
{
case 0 : if(pfield(pl_year,&key.plyear) !=0)
prtmsg(2,23,p18);
break;
case -2 : setcook();
science();
328:         break;
329:     case -3:  mv_cur(1,14,8);
330:         break;
331:     }
332:     flag=gtube(14,8,p1_abstr,p1abstr);
333:     switch(flag)
334:     { case 0:  if(pfield(p1_abstr,p1abstr) !=0)
335:             prtnmsg(2,23,p18);
336:             break;
337:         case -2:  setcook();
338:             science();
339:             break;
340:         case -3:  mv_cur(1,15,12);
341:             break;
342:     }
343:     flag=gtube(15,12,p1_idx,p1index);
344:     switch(flag)
345:     { case 0:  if(pfield(p1_idx,p1index) !=0)
346:             prtnmsg(2,29,p18);
347:             break;
348:         case -2:  setcook();
349:             science();
350:             break;
351:         case -3:  break;
352:     }
353:     rt=yn2p19();
354:     } while ((rt == 1); /* yes */
355:     ulockrec(policy);
356: }
357: }
358: }
359: more_mod_pol()
360: { int flag;
361:     flag=yn2p128();
362:     switch(flag) {
363:         case -2:  setcook();
364:             science();
365:             break;
366:         case 0 :  setcook();
367:             science();
368:         case 1 :  setcook();
369:             mod_pol_direct();
370:             get_mod_pol();
371:             more_mod_pol();
372:             break;
373:         default :break;
374:     }
375: }
376: mod_pol_direct()
377: { struct plkey key;
378:     int flag;
379:     mv_cur(1,0,4); clear();
380:     prmp(3,4,p128);
381:     prmp(3,6,p11);
383:   prmp(3,7,p12);
384:   do {
385:     flag=gtube(24,8,p1_id,&key.p1id);
386:     switch(flag) {
387:       case 0 : prmp(8,7,"25");break;
388:       case -2 : setcook();
389:       science();
390:       break;
391:       case -3 : prmp(8,7,"25");mv_cur(1,10,7);
392:       break;
393:     }
394:     flag=gtube(10,7,p1_year,&key.plyear);
395:     switch(flag) {
396:       case 0 : break;
397:       case -2 : setcook();
398:       science();
399:       break;
400:       case -3 : break;
401:     }
402:   } while (flag == -3);
403:   if(acckey(policy,&key) !=0)
404:     { prmsg(2,23,p110);
405:       setcook();
406:       mod_pol();
407:     }
408:   else
409:     { mv_cur(1,0,4); clear();
410:       prmp(3,4,p11); pdata(24,4,p1_id);
411:       prmp(3,5,p12); prmp(8,5,"25");pdata(10,5,p1_year);
412:       prmp(3,6,p13); pdata(12,6,p1_case);
413:       prmp(3,8,p14); pdata(14,8,p1_abstr);
414:       prmp(3,12,p15);pdata(15,12,p1_idx);
415:     }
416:   }
417:   inq_pol();
418:   { short select;
419:     short flag;
420:     short ret,rc;
421:     mv_cur(1,0,3); eras_ln();
422:     prmp(1,3,head8); prmp(60,3,head2);
423:     mv_cur(1,0,4); clear();
424:     prmp(5,4,p130);
425:     prmp(5,5,p111);
426:     prmp(5,6,p112);
427:     prmp(5,8,p113);
428:     switch(gtube(13,8,con,&select)) {
429:       case -2 : setcook();
430:       science();
431:       break;
432:     }
433:   }
434:   switch(select)
435:     { case 1: { if(seqacc(policy,first)!=0)
436:       { prmsg(2,23,p117);
437:         setcook();
438:       }
inq_pol();
}
else {
    do
    {
        mv_cur(1,0,4); clear();
        prmp(3,4,p11); pdata(24,4,p1_id);
        prmp(3,5,p12); prmp(8,5,"25"); pdata(10,5,p1_year);
        prmp(3,6,p13); pdata(12,6,p1_case);
        prmp(3,8,p14); pdata(14,8,p1_abstr);
        prmp(3,12,p15); pdata(15,12,p1_idx);
        flagyorn(p131);
        switch(flag) {
            case 0 :
            break;
            case -2 : setcook();
            inq_pol();
            break;
            case 1 : break;
            default :;
        }
    } while(acc(policy,next)==0);
    flagyorn(40,23,p16);
    setcook();
    more_inq_pol();
} /* else */
}
/* case 1 */
break;
case 2 : setcook();
inq_pol_direct();
more_inq_pol();
break;
default : prmsg(2,23,p16);
setcook();
inq_pol();
/* case */
/* inq_pol */
more_inq_pol()
{
    int flag;
    switch(flag) {
    case -2 : setcook();
    science();
    break;
    case 0 : setcook();
    science();
    break;
    case 1 : setcook();
inq_pol_direct();
more_inq_pol();
break;
default : break;
}
}
inq_pol_direct()
{ struct plkey key;
Listing of pl.c

493: int flag, rc;
494: mv_cur(1,0,3); eras_ln();
495: prmp(1,3,head8); prmp(80,3,head2);
496: mv_cur(1,0,4); clear();
497: prmp(3,4,p133);
498: prmp(3,6,p11);
499: prmp(3,7,p12);
500: do {
501: flag=gtube(24,6,p1_id,&key.p1id);
502: switch(flag) {
503: case 0 : break;
504: case -2 : setcook();
505: science();
506: break;
507: case -3 : break;
508: }
509: flag=gtube(10,7,p1_year,&key.p1year);
510: switch(flag) {
511: case 0 : break;
512: case -2 : setcook();
513: science();
514: break;
515: case -3 : break;
516: }
517: while (flag == -3);
518: if(acckey(policy,&key) !=0)
519: { prmsg(2,23,p110);
520: setcook();
521: more_inq_poi();
522: }
523: else {
524: { mv_cur(1,0,4); clear();
525: prmp(3,4,p11); pdata(24,4,p1_id);
526: prmp(3,5,p12); prmp(8,5,"25");pdata(10,5,p1_year);
527: prmp(3,6,p13); pdata(12,6,p1_case);
528: prmp(3,8,p14); pdata(14,8,p1_abstr);
529: prmp(3,12,p15); pdata(15,12,p1_idx);
530: }
531: }
532: report_poi();
533: { short select;
534: mv_cur(1,0,3); eras_ln();
535: prmp(1,3,head8); prmp(80,3,head2);
536: mv_cur(1,0,4); clear();
537: prmp(5,4,p135_1);
538: prmp(5,5,p136);
539: prmp(5,6,p137);
540: prmp(5,8,p113);
541: prmp(14,8,con,&select)) {
542: switch(gtube(14,8,con,&select)) {
543: case -2 : setcook();
544: science();
545: break;
546: case -3 : setcook();
547: report_poi();
548: }
break;

switch(select) {
    case 1: setcook();
    scr1_pol();
    break;
    case 2: setcook();
    scr2_pol();
    break;
    default: setcook();
    report_pol();
    break;
}

scr1_pol()
{
    int row=4;
    short select;
    if(seqacc(policy,first) != 0)
    {
        prmsg(2,23,p117);
        setcook();
        report_pol();
    }
    else {
        do {
            if((row=4)||(row>23))
            {
                mv_cur(1,0,4);
                clear();
            }
            prmp(3,row,p11); pdata(24,row++,p1_id);
            prmp(3,row,p12); prmp(8,row,"25"); pdata(10,row++,p1_year);
            prmp(3,row,p13); pdata(12,row,p1_case);
            row=row+2;
            prmp(3,row,p14); pdata(14,row,p1_abstr);
            row=row+4;
            prmp(3,row,p15); pdata(15,row,p1_idx);
            row=row+2;
            if(row>22)
            {
                prmp(55,23,"-----");
                switch(gtube(61,23,con,&select)) {
                    case -2: setcook();
                        science();
                        break;
                    case -3: {
                        row=4;
                        mv_cur(1,0,4); clear();
                        break;
                    }
                }
            }
            } while(seqacc(policy,next) == 0);
    prmsg(40,23,p118);
    setcook();
    report_pol();
}
}
scr2_pol()

int flag;
short num;
int check2;
char buf_pol[100];
int row=4;
short select;
int str_len;
char buffer[100];
int i,check;
mv_cur(1,1,2); clear();
prmp(3,6,p135);
prmp(3,7," ");
i=0;
while((buffer[i]=getchar()) != '\n')
i++;
str_len=i;
if(seqacc(policy,first) != 0)
    { prmsg(2,23,p117);
      setcook();
      report_pol();
    }
else {
    do {
        gfield(p1_case,buf1_pol);
        if(vcmp(buf1_pol,buffer,str_len)==1)
            check2=1;
        else 
            check2=0;
        if(check2==1) {
            if(row==4)!!!{row>23})
                { mv_cur(1,1,4); clear(); }
            prmp(3,row,p11); pdata(24,row++,p1_id);
            prmp(3,row,p12); prmp(8,row,"25");pdata(10,row++,p1_year);
            prmp(3,row,p13); pdata(12,row,p1_case);
            row=row+2;
            prmp(3,row,p14); pdata(14,row,p1_abstr);
            row=row+4;
            prmp(3,row,p15); pdata(15,row,p1_idx);
            row=row+2;
            if(row>22)
                { prmp(55,23," ----->");
                  switch(stube(61,23,p1_id,&select)) {
                      case -2 : setcook();
                        science();
                        break;
                      case -3 : { row=4;
                                 mv_cur(1,1,4); clear();
                                 break;
                               }
                      }
                  }
              }
658:     check2=0;
659:     } while(seqacc(policy,next) == 0);
660: }
661: prtmag(55,23,p110);
662: setcook();
663: report_pol();
664: }
# Listing of sm.c

```c
1: #include "../def/file.h"
2: #include <stdio.h>
3: extern char head3[],head5[],head7[],head9[],head8[];
4: char sm1[] = "ขอให้ mystudent.... [..25..]
5: char sm2[] = "เรียนบัณฑิต [............]"
6: char sm3[] = "เรียนบัณฑิต [............]"
7: char sm4[] = "เรียนบัณฑิต [............]"
8: char sm5[] = "เรียนบัณฑิต [..mm/dd/yy]
9: char sm6[] = "เรียนบัณฑิต [............]"
10: char sm7[] = "เรียนบัณฑิต [............]"
11: char sm8[] = "เรียนบัณฑิต [............]"
12: char sm9[] = "ขอให้ mystudent.... [..25..]
13: char sm10[] = "ขอให้ mystudent.... [..25..]
14: char sm11[] = "ขอให้ mystudent.... [..25..]
15: char sm12[] = "ขอให้ mystudent.... [..25..]
16: char sm13[] = "ขอให้ mystudent.... [..25..]
17: char sm14[] = "ขอให้ mystudent.... [..25..]
18: char sm15[] = "ขอให้ mystudent.... [..25..]
19: char sm16[] = "ขอให้ mystudent.... [..25..]
20: char sm17[] = "ขอให้ mystudent.... [..25..]
21: char sm18[] = "ขอให้ mystudent.... [..25..]
22: char sm19[] = "ขอให้ mystudent.... [..25..]
23: char sm20[] = "ขอให้ mystudent.... [..25..]
24: char sm21[] = "ขอให้ mystudent.... [..25..]
25: char sm22[] = "ขอให้ mystudent.... [..25..]
26: char sm23[] = "ขอให้ mystudent.... [..25..]
27: char sm24[] = "ขอให้ mystudent.... [..25..]
28: char sm25[] = "ขอให้ mystudent.... [..25..]
29: char sm26[] = "ขอให้ mystudent.... [..25..]
30: char sm27[] = "ขอให้ mystudent.... [..25..]
31: char sm28[] = "ขอให้ mystudent.... [..25..]
32: char sm29[] = "ขอให้ mystudent.... [..25..]
33: char sm30[] = "ขอให้ mystudent.... [..25..]
34: char sm31[] = "ขอให้ mystudent.... [..25..]
35: char sm32[] = "ขอให้ mystudent.... [..25..]
36: char sm33[] = "ขอให้ mystudent.... [..25..]
37: char sm34[] = "ขอให้ mystudent.... [..25..]
38: char sm35[] = "ขอให้ mystudent.... [..25..]
39: char sm36[] = "ขอให้ mystudent.... [..25..]
40: struct smkey { short sid;
41:     short syear;
42: };
43: add_sem() /* add seminar record */
44: { struct smkey key;
45:   char scname[100];
46:   char aby[100];
47:   char splace[100];
48:   short sdate;
49:   char sabstr[256];
```

51: char sindex[10];
52: long addr;
53: int flag;
54: int rc, rcc;
55: mv_cur(1,0,3); eras_ln();
56: prmp(1,3,head5); prmp(80,3,head3);
57: mv_cur(1,0,4); clear();
58: prmp(3,4,sm1);
59: prmp(3,5,sm2);
60: prmp(3,7,sm3);
61: prmp(3,9,sm4);
62: prmp(3,11,sm5);
63: prmp(3,12,sm6);
64: prmp(3,16,sm7);
65: do {
66:     flag=tube(20,4,smid, &key, sid);
67:     switch(flag) {
68:         case 0 : mv_cur(1,22,4); printf("/25");
69:             break;
70:         case -2 : setcook();
71:             social();
72:             break;
73:         case -3 : mv_cur(1,22,4); printf("/25");
74:             break;
75:     }
76:     flag=tube(25,4,semyear, &key, eyear);
77:     switch(flag) {
78:         case 0 : break;
79:         case -2 : setcook();
80:             social();
81:             break;
82:         case -3 : break;
83:     }
84: } while (flag == -3);
85: if(addr[rec(seminar, &key) != 0])
86: {
87:     if((rc=yorn(sm21)) == 1))
88:     { setcook();
89:         add_sem();
90:     }
91:     else
92:     { setcook();
93:         social();
94:     }
95: }
96: else
97: { loc(seminar, &addr);
98:     if(lockrec(seminar) != 0)
99:     { prmsg(2,23, sm9);
100:         setcook();
101:         social();
102:     }
103: else {
104:     do {
105:         flag=tube(12,5, sm_case, scase);
108:     switch(flag) {
109:         case 0 : if(pffield(sem_case,sccase) != 0)
110:             prmsg(2,23,sm10);
111:             break;
112:         case -2 : setcook();
113:             social();
114:             break;
115:         case -3 : mv_cur(1,12,7);
116:             break;
117:     }
118:     flag=gtube(12,7,sem_by,sby);
119:     switch(flag) {
120:         case 0 : if(pffield(sem_by,sby) != 0)
121:             prmsg(2,23,sm10);
122:             break;
123:         case -2 : setcook();
124:             social();
125:             break;
126:         case -3 : mv_cur(1,13,9);
127:             break;
128:     }
129:     flag=gtube(13,9,pla_ce,splace);
130:     switch(flag) {
131:         case 0 : if(pffield(pla_ce,splace) != 0)
132:             prmsg(2,23,sm10);
133:             break;
134:         case -2 : setcook();
135:             social();
136:             break;
137:         case -3 : mv_cur(1,12,11);
138:             break;
139:     }
140:     flag=gtube(12,11,sem_date,&date);
141:     switch(flag) {
142:         case 0 : if(pffield(sem_date,&date) != 0)
143:             prmsg(2,23,sm10);
144:             break;
145:         case -2 : setcook();
146:             social();
147:             break;
148:     }
149:     flag=gtube(14,12,sem_abst,sabstr);
150:     switch(flag) {
151:         case 0 : if(pffield(sem_abst,sabstr) != 0)
152:             prmsg(2,23,sm10);
153:             break;
154:         case -2 : setcook();
155:             social();
156:             break;
157:         case -3 : mv_cur(1,15,16);
158:             break;
159:     }
160:     flag=gtube(15,16,sem_idx,sindex);
switch(flag) {
    case 0 : if(pfield(sem_idx,sindex) != 0)
             prtmsg(22,23,sm10);
             break;
    case -2 : setcook();
             social();
             break;
    case -3 : break;
}
rc=yorn(sm11);
} while(rc==1); /* yes */
unlockrec(seminar);
setloc(seminar,addr);
mv_cur(1,0,23); eras_ln();
if(rc=yorn(sm22)==1)
    { setcook();
      add_sem();
    }
else
    { setcook();
      social();
    }
} /* add sem */
del_sem()
{
    struct smkey key;
    int flag;
    int rc;
    mv_cur(1,0,3); eras_ln();
    prmp(1,3,head6); prmp(80,3,head3);
    mv_cur(1,0,4); clear();
    prmp(3,4,sm24);
    prmp(3,6,sm1);
    do {
        flag=gtube(20,8,semid,&key.sid);
        switch(flag) {
            case 0 : mv_cur(1,22,6); printf("/25");
             break;
            case -2 : setcook();
             social();
             break;
            case -3 : mv_cur(1,22,6); printf("/25");
             break;
        }
        flag=gtube(25,6,semyear,&key.syear);
        switch(flag) {
            case 0 : break;
            case -2 : setcook();
             social();
             break;
            case -3 : break;
        }
    } while (flag == -3);
if(acckey(seminar,&key)!=0)
{
    prmsg(2,23,sm12);
    mv_cur(1,0,4); clear();
    if(rc=yorn(sm25)==1) /* yes */
    {
        setpok();
        del_sem();
    }
    else
    {
        setcook();
        social();
    }
}
else {
    mv_cur(1,0,4); clear();
    prmp(3,4,sm24);
    prmp(3,6,sm1); pdata(20,6,semid); prep(22,6,"/25");
    pdata(28,6,semyear);
    prmp(3,7,sm2); pdata(12,7,sem_case);
    prmp(3,9,sm3); pdata(12,9,sem_by);
    prmp(3,11,sm4); pdata(13,11,pla_ce);
    prmp(3,13,sm5); pdata(12,13,sem_date);
    prmp(3,14,sm6); pdata(14,14,sem_subj);
    prmp(3,18,sm7); pdata(15,18,sem_idx);
    if((rc=yorn(sm26)==1) /*yes*/
    {
        {if(lockrec(seminar) != 0)
        {
            prmsg(2,23,sm9);
            setcook();
            del_sem();
        }
        else
        {
            if(delete(seminar)==0)
            {
                prmsg(2,23,sm27);
                setcook();
                social();
            }
            unlockrec(seminar);
        }
        }
        if((rc=yorn(sm25)==1) /*yes*/
        {
            setcook();
            del_sem();
        }
        else
        {
            setcook();
            social();
        }
    } /* else */
} /* del sem */
mod_sem()
{ short select;
  short flag;
  short ret,rc;
  mv_cur(1,0,3); eras_ln();
271: prmp(1,3,head7); prmp(50,3,head3);
272: mv_cur(1,0,4); clear();
273: prmp(5,4,sm28);
274: prmp(5,5,sm13);
275: prmp(5,6,sm14);
276: prmp(5,8,sm15);
277: switch(gtube(13,8,con,&select)) {
278: case -2 : setcook();
279:     social();
280:     break;
281: case -3 : setcook();
282:     mod_sem();
283:     break;
284: }
285: switch(select)
286: { case 1 : if(seqacc(seminar,first)!=0)
287:     { prmsg(2,23,sm19);
288:       setcook();
289:       social();
290:     }
291: else
292:     { do
293:     { mv_cur(1,0,4); clear();
294:       prmp(3,4,sm1); pdata(20,4,semid); mv_cur(1,23,4);
295:       printf("/25"); pdata(25,4,semyear);
296:       prmp(3,5,sm2); pdata(12,5,sem_case);
297:       prmp(3,7,sm3); pdata(12,7,sem_by);
298:       prmp(3,9,sm4); pdata(13,9,pla_ce);
299:       prmp(3,11,sm5); pdata(12,11,sem_data);
300:       prmp(3,12,sm6); pdata(14,12,sem_abst);
301:       prmp(3,16,sm7); pdata(15,16,sem_idx);
302:       flag=yorn(sm28);
303:       switch(flag) {
304:       case -2 : setcook();
305:             mod_sem();
306:             break;
307:       case 0 : break;
308:       case 1 : setcook();
309:             get_mod_sem();
310:             break;
311:         default :;
312:     } while(seqacc(seminar,next)==0);
313:     prmsg(40,23,sm18);
314:     setcook();
315:     more_mod_sem();
316:     } /* else */
317: } /* case 1 */
318: break;
319: case 2 : setcook();
320:     mod_sem_direct();
321: get_mod_sem();
322: more_mod_sem();
323: break;
324: default : prmsg(2,23,sm8);
Listing of sm.c

326:     setcook();
327:     mod_sem();
328:     } /* case */
329:     } /* mod sem */
330:
331:     get_mod_sem()
332:     { struct semkey key;
333:     char scase[100];
334:     char sby[100];
335:     char splace[100];
336:     short sdtime;
337:     char sabsid[256];
338:     char sindex[10];
339:     int rt, rcc;
340:     int flag;
341:     if(lockrec(seminar) != 0)
342:     { printf ("22,3,smB");
343:     setcook();
344:     mod_sem();
345:     }
346:     else {
347:     do {
348:     flag=gtube(20,4,semid,&key.sid);
349:     switch(flag) {
350:     case 0 : mv_cur(1,23,4); printf("/25");
351:     break;
352:     case -2 : setcook();
353:     social();
354:     break;
355:     case -3 : mv_cur(1,23,4); printf("/25");
356:     break;
357:     }
358:     flag=gtube(25,4,semyear,&key.syyear);
359:     switch(flag) {
360:     case 0 : break;
361:     case -2 : setcook();
362:     social();
363:     break;
364:     case -3 : break;
365:     }
366:     flag=gtube(12,5,sem_case,scase);
367:     switch(flag) {
368:     case 0 : if(pfield(sem_case,scase) != 0)
369:     printf(2,12,sm10);
370:     break;
371:     case -2 : setcook();
372:     social();
373:     break;
374:     case -3 : mv_cur(1,12,7);
375:     break;
376:     }
377:     flag=gtube(12,7,sem_by,sby);
378:     switch(flag)
379:     { case 0 : if(pfield(sem_by,sby) != 0)
380:     printf(2,23,sm10);
381:         break;
382:     case -2 : setcook();
383:         social();
384:         break;
385:     case -3 : mv_cur(1,13,9);
386:         break;
387:     }
388:     flag=gtube(13,9,pla_ce,splace);
389:     switch(flag)
390:     { case 0 : if(pfield(pla_ce,splace) !=0)
391:         prtmag(2,23,sm10);
392:         break;
393:         setcook();
394:         social();
395:         break;
396:     case -2 : mv_cur(1,12,11);
397:         break;
398:     }
399:     flag=gtube(12,11,sem_date,&sdate);
400:     switch(flag)
401:     { case 0 : if(pfield(sem_date,&sdate) !=0)
402:         prtmag(2,23,sm10);
403:         break;
404:         setcook();
405:         social();
406:         break;
407:     case -3 : mv_cur(1,14,12);
408:         break;
409:     }
410:     flag=gtube(14,12,sem_abst,sabstr);
411:     switch(flag)
412:     { case 0 : if(pfield(sem_abst,sabstr) !=0)
413:         prtmag(2,23,sm10);
414:         break;
415:         setcook();
416:         social();
417:         break;
418:     case -3 : mv_cur(1,15,18);
419:         break;
420:     }
421:     flag=gtube(15,16,sem_idx,&index);
422:     switch(flag)
423:     { case 0 : if(pfield(sem_idx,&index) !=0)
424:         prtmag(2,23,sm10);
425:         break;
426:         setcook();
427:         social();
428:         break;
429:     case -3 : break;
430:     }
431:     rt=yorn(sm11);
432: } while (rt == 1); /* yes */
433: ulockrec(seminar);
434: }
435: 

Listing of sm.c

436:
437: more_mod_sem()
438: { int flag;
439:   flag=ynor(sm30);
440:   switch(flag) {
441:     case -2 : setcook();
442:       social();
443:       break;
444:     case 0 : setcook();
445:       social();
446:     case 1 : setcook();
447:       mod_sem_direct();
448:       get_mod_sem();
449:       more_mod_sem();
450:       break;
451:     default :break;
452:   }
453: }
454:
455: mod_sem_direct()
456: { struct amkey key;
457:   int flag;
458:   mv_cur(1,0,3); erase();
459:   prmp(head7); prmp(80,3,head3);
460:   mv_cur(1,0,4); clear();
461:   prmp(3,4,sm31);
462:   prmp(3,6,sm1);
463:   do {
464:     flag=endline(20,4,semid,&key.sid);
465:     switch(flag) {
466:       case 0 : mv_cur(1,23,4); printf("/25");
467:       break;
468:       case -2 : setcook();
469:         social();
470:       break;
471:       case -3 : mv_cur(1,23,4); printf("/25");
472:       break;
473:     }
474:     flag=gtube(25,6,semyear,&key.year);
475:     switch(flag) {
476:       case 0 : break;
477:       case -2 : setcook();
478:         social();
479:       break;
480:     case -3 : break;
481:   }
482: } while (flag == -3);
483: if(ackey(seminar,&key) !=0)
484:   { prtmeg(2,23,sm12);
485:     setcook();
486:     mod_sem();
487:   }
488: else
489:   { mv_cur(1,0,4); clear();
490:     prmp(3,4,sm1); pdate(20,4,semid);mv_cur(1,23,4); printf("/25");
Listing of sm.c

491:   pdata(26,4,semyear);
492:   prmp(3,5,sm2);   pdata(12,5,sem_case);
493:   prmp(3,7,sm3);   pdata(12,7,sem_by);
494:   prmp(3,9,sm4);   pdata(13,9,pla_ca);
495:   prmp(3,11,sm5);  pdata(12,11,sem_date);
496:   prmp(3,12,sm6);  pdata(14,12,sem_abst);
497:   prmp(3,16,sm7);  pdata(15,16,sem_idx);
498: }
499: }
500: inq_sem()
501: {  short select;
502:   short flag;
503:   short ret,rc;
504:   mv_cur(1,0,3); erms_ln(3);
505:   prmp(1,0,head0); prmp(head0);
506:   mv_cur(1,0,4); clear();
507:   prmp(5,4,sm32);
508:   prmp(5,5,sm13);
509:   prmp(5,6,sm14);
510:   prmp(5,8,sm15);
511:   switch(stube(13,8,con,select)) {
512:     case -2: setcook();
513:     case -3: setcook();
514:     social();
515:     break;
516:     case 3: setcook();
517:     inq_sem();
518:     break;
519:   }
520:   switch(select)
521:   { case 1: { if(seqacc(seminar,first)! = 0)
522:       { prtmag(2,23,sm19);
523:         setcook();
524:         inq_sem();
525:       }
526:     } else do
527:     { mv_cur(1,0,4); clear();
528:       prmp(3,4,sm1); pdata(20,4,semid); mv_cur(1,23,4);
529:       printf("/25") ; pdata(26,4,semyear);
530:       prmp(3,5,sm2);_pdata(12,5,sem_case);
531:       prmp(3,7,sm3); pdata(12,7,sem_by);
532:       prmp(3,9,sm4); pdata(13,9,pla_ca);
533:       prmp(3,11,sm5); pdata(12,11,sem_date);
534:       prmp(3,12,sm6); pdata(14,12,sem_abst);
535:       prmp(3,16,sm7); pdata(15,16,sem_idx);
536:       flags=ynorn(sm33);
537:       switch(flag) {
538:       case 0 :
539:         case -2 : setcook();
540:         inq_sem();
541:         break;
542:       case 1 : break;
543:     } } /* case */
544:     while(seqacc(seminar,next)==0);

-171-
Listing of sm.c

```
546:     prmsg(40,23,sm18);
547:     setcook();
548:     more_inq_sem();
549:     } /* else */
550:     } /* case 1 */
551:     break;
552:     case 2 : setcook();
553:     inq_sem_direct();
554:     more_inq_sem();
555:     break;
556:     default : prmsg(2,23,sm8);
557:     setcook();
558:     inq_sem();
559:     } /* case */
560:     } /* inq_sem */
561:     
562:     more_inq_sem()
563:     { int flag;
564:      flag=yorn(sm34);
565:      switch(flag) {
566:      case -2 : ;
567:      case 0 : setcook();
568:      social();
569:      case 1 : setcook();
570:      inq_sem_direct();
571:      more_inq_sem();
572:      break;
573:      default : break;
574:      }
575:     }
576:     
577:     inq_sem_direct()
578:     { struct amkey key;
579:      int flag,rc;
580:      mv_cur(1,0,3); oras_ln();
581:      prmp(1,3,head8); prmp(head3);
582:      mv_cur(1,0,4); clear();
583:      prmp(3,4,sm35);
584:      prmp(3,5,sm1);
585:      do {
586:      flag=gtube(20,4,semid,&key.mid);
587:      switch(flag) {
588:      case 0 : mv_cur(1,23,4); printf("/25");
589:      break;
590:      case -2 : setcook();
591:      social();
592:      break;
593:      case -3 : mv_cur(1,23,4); printf("/25");
594:      break;
595:      }
596:      flag=gtube(25,6,semyear,&key.year);
597:      switch(flag) {
598:      case 0 : break;
599:      case -2 : setcook();
600:      social();
```
601:     break;
602: case -3 : break;
603: default : prmsg(2,23,sm8);
604:     setcook();
605:     social();
606:     break;
607: }
608: } while (flag == -3);
609: if(acceid(seminar,&key) !=0)  
610: { prmsg(2,23,sm12);
611:     setcook();
612:     more_inq_sem();
613: }
614: else
615: { mv_cur(1,0,4); clear();
616:     prmp(3,4,sm1); pdta(20,4,semid);mv_cur(1,23,4); printf("/25");
617:     pdta(20,4,semyear);
618:     prmp(3,5,sm2); pdta(12,5,sem_case);
619:     prmp(3,7,sm3); pdta(12,7,sem_by);
620:     prmp(3,9,sm4); pdta(13,9,pla_ce);
621:     prmp(3,11,sm5); pdta(12,11,sem_date);
622:     prmp(3,12,sm6); pdta(14,12,sem_abst);
623:     prmp(3,16,sm7); pdta(15,16,sem_idx);
624: }
625: }
626: report_sem()
627: { int rows=4;
628: short select;
629: mv_cur(1,0,4); clear();
630: if(seqacc(seminar,first) != 0)
631: { prmsg(2,23,sm19);
632:     setcook();
633:     social();
634: }
635: else {
636:     do {
637:     if((row==4)||(row>23))
638:     { mv_cur(1,0,3);
639:     prmp(1,3,head9); prmp(60,3,head3);
640:     }
641:     prmp(3,row,sm1); pdta(20,row,semid);mv_cur(1,23,row);
642:     printf("/25");pdta(20,row++,semyear);
643:     prmp(3,row,sm2); pdta(12,row,sem_case);
644:     row=row+2;
645:     prmp(3,row,sm3); pdta(12,row,sem_by);
646:     row=row+2;
647:     prmp(3,row,sm4); pdta(13,row,pla_ce);
648:     row=row+2;
649:     prmp(3,row,sm5); pdta(12,row++,sem_date);
650:     prmp(3,row,sm6); pdta(14,row,sem_abst);
651:     row=row+4;
652:     prmp(3,row,sm7); pdta(15,row,sem_idx);
653:     row=row+2;
654:     prmpv(55,23," ----> ");
655: }
switch(gtube(81,23,p1_id,&select)) {
  case -2 : setcook();
  social();
  break;
  case -3 : { row=4;
  mv_cur(1,0,4); clear();
  break;
  }
  }
} while(seqacc(seminar, next) == 0);
prtmsg(55,23,sm18);
setcook();
social();
}
}
APPENDIX B

Program listing on Development of the Integrated Database Designer
 Program: INPUT_CO.PRG

 System: INTEGRATE DATABASE DESIGNER
 Author: TAWEWAN PROMGUNTHA
 Copyright (c) 1992, ASSUMPTION UNIVERSITY
 Last modified: 12/10/92, 20:23

 Called by: MAIN.PRG
 Uses: FCODE.DBF
 : USEFIL.DBF
 Indexes: AD EX

 Documented: 12/10/92 at 20:58

 ON ERROR RETURN
 SET EXCL ON
 SET STAT OFF
 SET TALK OFF
 SET ECHO OFF
 SET SAFE OFF
 CLOSE ALL
 CLEAR

 @ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
 @ 3, TO 3,79
 @ 4,1 SAY "CODE DEFINE SECTION -"
 @ 4,55 SAY "ASSUMPTION UNIVERSITY"
 @ 5,1 TO 5,79 DOUB
 @ 5,79 SAY DATE()
 driv_ = SPACE(10)
 @ 9,10 SAY "INPUT DRIVE TO SAVE DATA ? " GET driv_
 READ
 @ 9,1 CLEAR TO 9,79
 driv_ = LTRIM(RTRIM(driv_)) + "\"
 usefil_ = driv_ "PCODE_F.DBF"
 fill_ = SPACE(1)
 USE fcode
 IF FILE("USEFIL")
 DDO WHILE fill_ # 'N' .AND. fill_ # 'O'
 = CHR(7)
 @ 8,8 TO 11,60 DOUB
9,15 SAY "======> HAVE DATA FILE <======"
10,15 SAY "Do you want (N)ew or (O)ld file ? " GET fil_PICT '!
READ
ENDO

8,1 CLEAR TO 11,79
IF fil_ = 'N'
COPY STRU TO ausfile
ENDIF
ELSE
COPY STRU TO ausfile
ENDIF
CLOSE ALL
USE ausfile
dex = LTRIM(RTRIM(drvw_))="DEXX"
INDEX ON code TO &dexin_ = 'Y'
DO WHILE fin_ = 'Y'
chk_ = 'N'
cod_ = SPACE(5)
num_ = SPACE(15)
lab_ = 'N'
6,1 CLEAR TO 18,79
DO WHILE chk_ = 'N'
8,10 SAY " INPUT CODE : " GET cod_PICT '!!!!!!'
READ
IF cod_ = SPACE(5)
RETURN
ENDIF
SEEK cod_
IF FOUND()
?? CHK(7)
10,10 SAY " HAVE THIS CODE !!! "
11,10 SAY " PLEASE RECHECK "
12,10 SAY " ENTITY NAME : "
?? entity
chk_ = 'Y'
WAIT " Press any key"
14,1 CLEAR TO 13,79
ELSE
10,10 SAY "ENTITY NAME : " GET num_PICT '!!!!!!!!!!!!!!!!!!!'
12,10 SAY "PRIMARY KEY (Y/N) ? " GET lab_PICT '!
READ
IF lab_ = 'Y'
lab_ = 'N'
ENDIF
chk_ = 'Y'
@ 14,10 SAY "ARE YOU SURE?" GET chk_ PICT 'Y'
READ
   IF chk_ = 'Y'
      APPEND BLANK
      REPL code WITH cod_
      REPL entity WITH nam_
      REPL label WITH lab_
   ENDIF
ENDDO
ENDDO
@ 14,10 SAY "MORE DATA?" GET fin_ PICT 'Y'
READ
CLOSE ALL
CLEAR
RETURN
SET STATUS
*: EOF: INPUT_CO.ACT
1 "="/* Simulation Database Designer */"
2 "="
3 "="
4 "="
5 "="
6 "="
7 "="
8 "="
9 "="
10 "="
11 "="
12 "="
13 "="
14 "="
15 "="
16 "="
17 "="
18 "="
19 "="
20 "="
21 "="
22 "="
23 "="
24 "="
25 "="
26 "="
27 "="
28 "="
29 "="
30 "="
31 "="
32 "="
33 "="
34 "="
35 "="
36 "="
37 "="
38 "="
39 "="
40 "="
41 "="
42 "="
43 "="
44 "="
45 "="
66 \[ \text{clear to } 18,79 \]
67 \[ \text{name}_{\text{fr}} = \text{SPACE}(7) \]
68 \[ \text{if } \text{name}_{\text{fr}} = \text{SPACE}(7) \]
69 \[ \text{endif} \]
70 \[ \text{enddo} \]
71 \[ \text{if } \text{us}_{\text{fil}} = 'W' \]
72 \[ \text{loop} \]
73 \[ \text{endif} \]
74 \[ \text{endif} \]
75 \[ \text{fil}_{\text{co}} = \text{driv} + 'FCODE_F.DBF' \]
76 \[ \text{fil}_{\text{co}} = \text{driv} + 'FCODE_F.DBF' \]
77 \[ \text{select 1} \]
78 \[ \text{use } \text{fil}_{\text{co}} \]
79 \[ \text{inde on code to } \text{fil}_{\text{co}} \]
80 \[ \text{select 2} \]
81 \[ \text{use } \text{group}_{\text{na}} \]
82 \[ \text{copy stru to } \text{name}_{\text{fr}} \]
83 \[ \text{select 2} \]
84 \[ \text{use } \text{name}_{\text{fr}} \]
85 \[ \text{select 1} \]
86 \[ \text{do while } \text{not. eof()} \]
87 \[ \text{clear to } 18,79 \]
88 \[ \text{stor code to cod}_{\text{co}} \]
89 \[ \text{stor entity to } \text{nam}_{\text{co}} \]
90 \[ \text{stor entity to } \text{gr}_{\text{nam}} \]
STOR LABEL TO lab_
STOR SPACE(10) TO ran_
@ 10,18 SAY "CODE : 
?? cod_
@ 10,28 SAY "ENTITY NAME : 
?? nam_
?? " LABEL : 
?? lab_
@ 12,18 SAY "GROUP : 
?? STR(do_loop,1)
?? " NAME : 
?? name_tr
@ 15,9 TO 17,7 DOUB
@ 16,18 SAY "USE ENTITY NAME: " GET gr_nam PICT '!!!!!!!!!!!!!!'
@ Row(),Col()+1 SAY "DATA TYPE: " GET ran_PICT '!!!!!!!!!!!!'
READ
sur_ = 'Y'
@ 19,18 SAY "ARE YOU SURE ? " GET sur_PICT '
READ
IF sur_ = 'Y'
SELE 2
APPE BLAW
REPL code WITH cod_
REPL entity WITH nam_
REPL group WITH gr_nam
REPL RANGE WITH ran_
ELSE
ENDO
ENDIF
SELE 1
SKIP
ENDDO
ENDDO
do_loop = do_loop + 1
ENDDO
*; EOF; DEF_NAME.ACT
Program: INPUT_MA.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEAN PROMGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 20:41

Called by: MAIN.PRG
Uses: MASTER.DBF
       APIL_USB.DBF
       APIL_1

Indexes: APIL_1
        APIL_DEX

Documented: 12/10/92 at 20:58

10 * INPUT_MA.PRG *
20 SET STAT.OFF
21 SET EXCL.OFF
22 SET TALK OFF
23 SET ECHO OFF
24 SET SAFE OFF
25 CLOSE ALL
26 CLEAR
27 @ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
28 @ 3,1 TO 3,79
29 @ 4,1 SAY "INPUT DATA SECTION : -"
30 @ 4,55 SAY "ASSUMPTION UNIVERSITY"
31 @ 5,1 TO 5,79 DOUB
32 @ 5,65 SAY DATE()
33 dri_ = SPACE(8)
34 @ 9,10 SAY "INPUT DRIVE TO SAVE DATA : " GET dri_
35 READ
36 @ 9,1 CLEAR TO 9,79
37 driv = LTRIM(RTRIM(dri_)+"1")
38 @ 9,30 SAY "LOADING"
39 fil_use = driv +'INTEGRATE.DBF'
40 fil dex = driv +'INTEGRATE'
41 IF FILE("APIL_USB")
42   x_ = SPACE(1)
IF .NOT. FOUND()
  ?? chr(27)
  SET COLO TO W/B
  @ 10,15 SAY "DON'T DEFINE THIS CODE"
  SET COLO TO T
  @ 11,15 SAY "RE-INPUT A NEW CODE (Y/N) ?" GET re_in PICT '!' READ
  IF re_in = 'Y'
    @ 10,14 CLEAR TO 11,79
  LOOP
ELSE
  EXIT
ENDIF

ELSE
STORE entity TO man_
STORE LABEL TO lab_
chr_ = 'Y'
DO WHILE chr_ = 'Y'
  man_ = 'N'
  @ 8,35 SAY '------'
  ?? LTRIM(RTRIM(man_))
  ?? ':'
  ?? lab_
  @ 10,10 SAY 'ROLE : ' GET rol_ PICT '!!!!!!'
  @ 12,10 SAY 'FACT : ' GET fac_ PICT '!!!!!!'
  @ 14,10 SAY 'CONSTRAINT (O)ne, (M)any,(T)many-to-many : '
  GET constr_ PICT '!' READ
  DO CASE
    CASE constr_ = 'O'
      constr_ = 'ONE'
    CASE constr_ = 'M'
      constr_ = 'MANY'
    CASE constr_ = 'T'
      constr_ = 'MANY_TO_MANY'
    END CASE
  @ 14,1 CLEAR TO 14,79
  @ 14,10 SAY 'CONSTRAINT : '
  ?? LTRIM(RTRIM(constr_))
  @ 14,40 SAY 'MANDATORY : ' GET m_ PICT '!' READ
  IF m_ = 'Y'
    man_ = 'N'
  ENDIF
  SELF 2
SEEQ rol
  IF FOUND()
    ?? CHR(7)
    @ 10,25 SAY 'DUPLICATE IN ROLE NUMBER , CAN NOT PROCESS'
    @ 11,25 SAY "CHANGE ROLE NUMBER (Y/N)? " GET chn PICT 'Y'
    READ
    IF chn = 'Y'
      @ 10,1 CLEAR TO 17,79
    ENDIF
    ELSE
      ?? CHR(7)
      @ 12,10 SAY "RE-CHECK NUMBER OF ROLE FIRST" chn = 'N'
      ENDIF
    ELSE
      chn = 'N'
      sur = 'Y'
      @ 16,10 SAY "ARE YOU SURE ? " GET sur PICT 'Y'
      READ
      IF sur = 'Y'
        APEX BLANK
        REPL code WITH cod_roln WITH rol
        REPL entity WITH nam_, Fact WITH fac
        REPL constrant WITH constr_, LABEL WITH lab
        REPL mandatory WITH man
      ENDIF
      ENDDO
  ENDIF
ENDDO
@ 18,10 SAY 'MORE DATE ? ' GET con PICT 'Y'
READ
@ 8,3 CLEAR TO 8,79
@ 9,1 CLEAR TO 18,79
ENDDO

MAIN.PRG
  | INPUT.CQ.PRG
  |   | FCODE.DBF (database)
  |   | AUSELECT.DBF (database)
  | DEF_NAME.PRG
  |   | AFIL (database)
  |   | GROUP_NAME.DBF (database)
  |   | NAME_GRP.DBF (database)
  | INPUT_NAME.PRG
  | MASTER.DBF (database)
  | AFIL_USE.DBF (database)
  | AFIL_1 (database)
  | EDIT.PRG
  |   | E_CODE.PRG
  |   | AUSELECT.DBF (database)
  |   | E_DEF.PRG
  |   | NAME_GRP.DBF (database)
  |   | E_WAS.PRG
  |   | AFIL_USE.DBF (database)
  | PRINT.PRG
  |   | P_CODE.PRG
  |   | AUSELECT.DBF (database)
  |   | P_NAME.PRG
  |   | NAME_GRP.DBF (database)
  |   | P_WAS.PRG
  |   | AFIL_USE.DBF (database)
  | PROCESS.PRG
  |   | AFIL_1 (database)
  |   | AFIL_2 (database)
  |   | AFIL_3.DBF (database)
  |   | AFIL_4 (database)
  |   | AFIL_5.DBF (database)
  |   | AFIL_6 (database)
  |   | AFIL_7 (database)
Program: MAIN.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEAN PROMGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 19:44

Call: INPUT_CO.PRG, DEF_NAME.PRG, INPUT_MA.PRG, EDIT.PRG, PRINT.PRG

Documented: 12/10/92 at 20:58

18
19
20
21
22
23
24
25
26
27
28
SET EXCL ON
SET TALE OFF
SET ECHO OFF
SET CONF ON
SET DATE BRIT
CLOSE ALL
SET STATUS OFF
@14,02 TO 16,17 DOUBLE
SET COLOR TO Whites
CLEAF
@07,00

SET EXCL ON
SET TALE OFF
SET ECHO OFF
SET CONF ON
SET DATE BRIT
CLOSE ALL
SET STATUS OFF
@14,02 TO 16,17 DOUBLE
SET COLOR TO Whites
STORE 'ASSUMPTION UNIVERSITY ' ASSUMPTION UNIVERSITY ' ASSUMPTION UNIVERSITY' ;
+ ' ASSUMPTION UNIVERSITY ' ASSUMPTION UNIVERSITY ' TO signon
STORE @ TO line
DO WHILE line <= 5
  IF line/2 = INT(line/2)
    @ line,00 SAY SUBSTR(signon,1,80) &color scheme 1 &B/n
  ELSE
    @ line,00 SAY SUBSTR(signon,5,80) &color scheme 1 &B/n
  ENDIF
  line = line+1
ENDDO
Line = 21
DO WHILE line > 21 .AND. line <= 23
  IF line/2 = INT(line/2)
    @ line,00 SAY SUBSTR(signon,1,80) &color scheme 1 &B/n
  ELSE
    @ line,00 SAY SUBSTR(signon,5,80) &color scheme 1 &B/n
  ENDIF
  line = line+1
ENDDO
@ 24,00 SAY SUBSTR(signon,1,79)
SET COLOR TO B/N
* @ 7,01 TO 19,37
SET COLOR TO #/#B
* @ 7,38 TO 19,78
SET COLOR TO #/B
SET COLOR TO #/#B
SAVE SCREEN TO xx
mcol=3
******
DO WHILE mcol >= 03 .AND. mcol <= 52
  RESTORE SCREEN FROM xx
  @ 16,mcol SAY ' ASSUMPTION '
  @ 15,mcol TO 17,mcol+14
STORE @ TO stime
DO WHILE stime < 35
  stime=stime+1
ENDDO
mcol=mcol+1
ENDDO
******
@ 07,01 TO 19,40
SET COLOR TO #/#B
* @ 8,55 SAY " A . U ."
SET COLOR TO #/#B
\$ 19,53 SAY 'Version: 1.00'
SET COLOR TO W+R
\$ 15,52 CLEAR TO 17,67
\$ 15,52 TO 17,67 DOUBLE
SET COLOR TO W+R
\$ 16,53 SAY 'ASSUMPTION'
SET COLOR TO BR+R
\$ 6,25 SAY 'Press anykey..... let's get started!!'
WAIT **
* EOF
**************
* MAIN PROGRAM
**************
*
SET TAIL OFF
SET STATUS OFF
SET ECHO OFF
SET SCOREBOARD OFF
CLOSE ALL
SET EXCLUSIVE OFF

IF ISCOLOR()
   normal = "W/B"
   inverse = "CR/G"
ELSE
   normal = "W"
   inverse = "I"
ENDIF

I = 1
*
1. INPUT CODE OF ENTITY
2. INPUT GROUP
3. INPUT DATA OF ENTITY
4...EDIT SECTION
5. PRINT OUT
6. EXIT
I = 10
ch = "W1"
NAME = SPACE(15)
*
T = TIME()

*************

DO WHILE I < 6
   SET COLOR TO normal
   ...
CLEAR

@ 1,1 SAY "ASSUMPTION UNIVERSITY (A.U.)"
@ 3,1 SAY "INTEGRATED DATABASE"
@ ROW(),COL()+1 SAY "DESIGNER"
@ 5,1 SAY REPLACE("-'",79)
@ 6,1 SAY "MAIN MENU"
@ 6,6 SAY "Date:"
@ 6,7 SAY DATE()
@ 7,58 SAY "BY IAVEEAM PRAOONTRA"
@ 8,1 SAY REPLACE("-'",79)
@ 10,10 SAY w1
@ 11,10 SAY w2
@ 12,10 SAY w3
@ 13,10 SAY w4
@ 14,10 SAY w5
@ 15,10 SAY w6
@ 16,1 SAY REPLACE("-'",79)

SET COLOR TO &normal+
@ 19,10 SAY "Press "+CHR(24)+" or "+CHR(25)
@ 19,10 COL() SAY "to move highlight and press "+CHR(17)+" to select"
+CHR(17)" to select"

SET COLOR TO &inverse
@ R,10 SAY &ch
@ 12,1 SAY REPLACE("-'",79)

DO WHILE INDEX() < 13
    X = 0
    DO WHILE X = 0
        X = INDEX()
    ENDDO
    SET COLOR TO &normal
    @ R,10 SAY &ch

    DO CASE
    CASE x = 24
        R = R+1
        R = TIP(R>15,10,R)
        I = VAL(RIGHT(ch,1))
        I = TIP(I>6,1,1)
        ch = "N"+STR(I,1)
        SET COLOR TO &inverse
        @ R,10 SAY &ch
    CASE x = 5
        R = R-1
        R = TIP(R<10,15,R)
    END
I = VAL(RIGHT(ch,11)) - 1
I = 1 IF (I<1,6,1); ch = "W"*STR(I,1)
SET COLOR TO "normal"
@ R,10 SAY &ch
CASE I = 11
SET COLOR TO "normal"
@ R,10 SAY &ch
SET COLOR TO "normal"
DO CASE
CASE I = 1
DO input_co
EXIT
CASE I = 2
DO def_name
EXIT
CASE I = 3
DO input_wa
EXIT
CASE I = 4
DO EDIT
EXIT
CASE I = 5
DO PRINT
EXIT
CASE I = 6
SET COLO TO "normal"
QUIT
*EXIT
SET COLOR TO "normal"
CLOSE ALL
ENDCASE
ENDCASE
ENDDO
CLOSE ALL
SET STATUS ON
SET ECHO ON
SET TALL ON
SET COLO TO
CLEAR
*; EOF; MAIN.ACT
Program: EDIT.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEAN PROMGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/18/92 16:08
Called by: MAIN.PRG
Calls: E_CODE.PRG
       E_DEF.PRG
       E_WAS.PRG

Documented: 12/18/92 at 20:38, FoxDoc version 1.0

SET EXCL ON
SET TALK OFF
SET STATUS OFF
SET ECHO OFF
SET SCOREBOARD OFF
CLOSE ALL
SET EXCLUSIVE OFF

IF ISCOLOR()
normal = "W/B"
inverse = "GR/N"

ELSE
normal = "W"
inverse = "I"

ENDIF

j = 1

e1 = "1. EDIT CODE OF ENTITY"
e2 = "2. EDIT GROUP"
e3 = "3. EDIT DATA OF ENTITY"
e4 = "4. EXIT"
e5 = 10
ech = "E1"
cname = SPACE(15)

s_time = TIME()
DO WHILE I # 4
  SET COLOR TO &normal
  CLEAR
  @ 2,1 SAY "ASSUMPTION UNIVERSITY (A.U.)"
  @ 3,1 SAY "INTEGRATED DATABASE"
  @ ROW(),COL()+1 SAY "DESIGNER"
  @ 5,1 SAY REPLACE(CHR(79),"",79)
  @ 6,1 SAY "EDIT MENU"
  @ 6,65 SAY "Dates"
  @ 6,72 SAY DATE()
  @ 7,58 SAY "BY TAWEAN PROM GU NTHA"
  @ 8,1 SAY REPLACE(CHR(79),"",79)
  @ 9
  @ 10,1# SAY e1
  @ 11,1# SAY e2
  @ 14,1# SAY e3
  @ 16,1# SAY e4
  @ 18,1 SAY REPLACE(CHR(79),"",79)
  SET COLOR TO &normal+
  @ 20,1# SAY "Press "+CHR(24)+" or "+CHR(25)+" to move highlight, and press "+CHR(17)+" to select;"
  @ 20,COL() SAY "to move highlight, and press "+CHR(17)+" to select;"
  SET COLOR TO &inverse
  @ er,10 SAY &ech
  @ 22,1" SAY REPLACE("",79)
  *
  DO WHILE INKEY() # 13
    I = 0
    DO WHILE I = 0
      I = INKEY()
    ENDDO
    SET COLOR TO &normal
    @ er,10 SAY &ech
    *
    DO CASE
      CASE X = 24
        er = er+2
        er = IF (er>16,10,er)
        J = VAL(RIGHT(&ech,1))+1
        J = IF (J>4,1,1)
        &ech = "E"+STR(J,1)
      SET COLOR TO &inverse
        @ er,10 SAY &ech
      CASE 5
  -193-
er = -er-2
er = IF (er<10,16,er)
J = VAL(RIGHT(ech,11))-1
J = IF (J<1,4,J)
ech = "E"+STR(J,1)
SET COLOR TO Anverse
4 er,10 SAY ech
CASE J = 1
SET COLOR TO Anverse
4 er,10 SAY ech
SET COLOR TO Anormal
DO CASE
CASE J = 1
DO e_code
EXIT
CASE J = 2
DO e_def
EXIT
CASE J = 3
DO e_max
EXIT
CASE J = 4
EXIT
SET COLOR TO Anormal
CLOSE ALL
ENDCASE
ENDCASE
ENDDO
CLEAR
RETURN
*: EOF; EDIT.ACT
Program: E_CODE.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEWA PROGWUNWA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 20:38
Called by: EDIT.PRG
Uses: USEFILE.DBF
Indexes: ADE

Documented: 12/10/92 at 20:38

ON ERROR RETURN
SET BOL ON
SET STAT OFF
SET TALL OFF
SET ECHO OFF
SET SAFE OFF
CLOSE ALL
CLEAR

@ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
@ 3,1 TO 3,79
@ 4,1 SAY "EDIT CODE DEFINE SECTION :";
@ 4,35 SAY "ASSUMPTION UNIVERSITY"
@ 5,1 TO 5,79 DOUB
@ 5,70 SAY DATE();
driv_ = SPACE(10)
@ 9,10 SAY "INPUT DRIVE TO SAVE DATA ? " GET driv_
READ
@ 9,1 CLEAR TO 9,79
driv_ = LTRIM(RTRIM(driv_));"
usefile = driv_+"E_CODE_F.DBF"
fil_ = 0

IF .NOT. FILE("USEFILE")
?? CHR(7)
DO WHIL fil_ < 1999
   @ 8,8 TO 11,60 DOUB
   @ 9,15 SAY " * * * * * DON'T HAVE DATA FILE "
-195-
E_CODE.ACT

Copyright, ASSUMPTION UNIVERSITY, 1992
INTEGRATE DATABASE DESIGNER.

12/10/92
21:41

ENDDO
@ B,1 CLEAR TO 11,79
RETURN

ELSE
CLOSE ALL
USE Ausell
dex = LTRIM(RTRIM(DTYP_))"DEXX"
INDEX ON code TO @dex
fin = "Y"
DO WHILE fin = "Y"
  chk = "N"
cod =SPACE(5)
  nam =SPACE(15)
lab = "N"
  @ 6,1 CLEAR TO 18,79
  DO WHILE chk = "N"
    @ B,10 SAY " INPUT CODE : " GET codPICT!!!"
    READ
    IF cod = SPACE(5)
      RETURN
    ENDIF
    SEEK cod_
    IF .NOT.FOUND()
      ?? CHR(?)
      @ 18,10 SAY " DON'T HAVE THIS CODE !!! "
      @ 12,10 SAY " PLEASE RECHECK "
      WAIT " Press any key"
      @ 18,1 CLEAR TO 13,79
    ELSE
      STORE entity TO nam_
      STORE LABEL TO lab_
      @ 18,10 SAY "ENTITY NAME : " GET namPICT '!!!!!!!!!!!!!!!'
      @ 12,10 SAY " PRIMARY KEY (Y/N) ? : " GET labPICT 'I'
      READ
      IF lab = 'Y'
        lab = 'W'
      ENDIF
    chk = "Y"
    @ 14,10 SAY "ARE YOU SURE ? " GET chkPICT 'I'
    READ
    IF chk = 'Y'
      REPL code WITH cod_
      REPL entity WITH nam_
      REPL LABEL WITH lab_
      getch
12/18/92
21:02

E_CODE.ACT

Copyright, ASSUMPTION UNIVERSITY, 1992
INTEGRATE DATABASE DESIGN

91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

ENDIF
ENDIF
ENDO
& 16,10 SAY "MORE DATA ? " GET fin_PICT '!' READ
ENDO
ENDIF
CLOSE ALL
CLEAR
RETURN
SET STAT ON

*: EOF: E_CODE.ACT

ASSUMPTION UNIVERSITY OF THAILAND

BROTHERS OF
ST. GABRIEL
LABOR
OMNIA
VINCIT

SINCE 1969

มหาวิทยาลัยธัญบุรี
Program: E_DEF_.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAVEAN PRONGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 21:03

Called by: EDIT.PRG
Uses: &NAME_GRP.DEF
Includes: &FILX

Documented: 12/10/92 at 20:58 ForDoc version 1.

*E_DEF_.PRG*
SET EXCL ON
SET STAT OFF
SET ECHO OFF
SET TALK OFF
SET SAFE OFF

CLOSE ALL
CLEAR

$ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
$ 3,1 TO 3,79
$ 4,1 SAY "EDIT DEFINE GROUP SECTION :-"
$ 4,55 SAY "ASSUMPTION UNIVERSITY"
$ 5,1 TO 5,79 DOUB
$ 5,70 SAY DATE()
$ driv_ = SPACE(10)
$ 7,10 SAY "INPUT DRIVE TO SAVE DATA ? " GET driv_
READ
$ 7,1 CLEAR TO 7,79
$ driv_ = LTRIM(RTRIM(driv_)) + "\"
$ name_gr_ = SPACE(7)
$ us_fil_ = SPACE(1)
$ 8,10 SAY "EDIT GROUP IN NAME : " GET name_gr_ PICT '!!!!!!'
READ
$ name_gr_ = LTRIM(RTRIM(name_gr_))
IF name_gr_ = SPACE(7)
RETURN

198
46  ENDIF
47
48  name_gr = driv + name_gr + "._DBF"
49
50  IF NOT.FILE('AWNAMEGR')
51     ?? CHR(7)
52     1la = 0
53
54     DO WHIL 1la < 1000
55        @9,10 SAY "DON'T HAVE FILE IN DISK!!"
56        1la = 1la + 1
57     ENDDO
58     RETURN
59  ENDIF
60
61  fil_ = driv + "FCODE_F.DBF"
62  fil = driv + "FCODE.F"
63
64  USE &name_gr
65  INDE ON code TO fil_
66  mor_ = 'Y'
67  DO WHIL mor_ = 'Y'
68     sur_ = 'N'
69     DO WHIL sur_ = 'N'
70        @10,1 CLEAR TO 18,79
71        cod_ = SPACE(5)
72
73     @10,10 SAY "CODE : " GET cod_ READ
74     IF cod_ = SPACE(5)
75     RETURN
76  ENDIF:
77     SEEK cod_
78     IF NOT.FOUND()
79        ??CHR(7)
80        @11,10 SAY "DON'T HAVE THIS CODE"
81        @12,10 SAY "RECHECK PLEASE"
82        WAIT "PRESS ANYKEY"
83        @11,1 CLEAR TO 13,79
84     ELSE
85        STOR entity TO nam_
86        STOR group TO gr_nam
87        STOR # TO ran_
88        @18,25 SAY "ENTITY NAME : "
89        ?? nam_
90        @12,8 TO 17,60 DOUB
91        @14,15 SAY "USE ENTITY NAME: " GET gr_nam PICT ' postings '"
92        @15,15 SAY " RANGE: " GET ran_ PICT '99999'
READ
sur_ = 'Y'
@ 18,10 SAY "ARE YOU SURE?" GET sur_PICT 'Y'
READ
IF sur_ = 'Y'
REPL code WITH cod_
REPL entity WITH ent_
REPL group WITH gr_
REPL RANGE WITH ran_
ELSE
END LOOP
ENDIF
ENDDO
@ 20,10 SAY "HAVE MORE TO EDIT?" GET mor_PICT 'Y'
READ
ENDD
*: EOF: E_DEF._ACT
Program: E_WAS.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEAN PROMGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 20:56

Called by: EDIT.PRG

Uses: &FIL_USE.DBF
Indexes: &FIL_DET

Documented: 12/10/92 at 20:58

EDIT MASTER
"E_WAS.PRG"
SET STAT OFF
SET ECL ON
SET TALK OFF
SET ECHO OFF
SET SAFE OFF
CLOSE ALL
CLEAR

@ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
@ 2,1 TO 3,79
@ 4,1 SAY "EDIT DATA SECTION : - "
@ 4,15 SAY "ASSUMPTION UNIVERSITY"
@ 5,1 TO 3,79 DOUB
@ 5,65 SAY DATE()
dri_ = SPACE(8)
@ 9,1# SAY "INPUT DRIVE TO SAVE DATA : " GET dri_
READ
CLEAR
@ 9,1 CLEAR TO 9,79
driv = LTRIM(RTRIM(dri_))""
@ 9,30 SAY "LOADING"
&FIL_use = driv + 'INTEGRATE.DBF'
&FIL_det = driv + 'INTEGRATE'
IF .NOT. FILE("&FIL_USE")
?? CR(27)
46    # 8,8 TO 11,78 DOUE
47    # 9,20 SAY "DO NOT HAVE -INTEGRATE.DBF-"
48    WAIT "PRESS ANY KEY"
49    # 8,1 CLEAR TO 10,79
50    RETURN
51  ENDIF
52
53    fil_1 = dirv + "CODE_F"
54
55
56
57    SELE 2
58    USE &fil_use
59    INDE ON role TO &fil_det
60
61    BROW
62
63    RETURN
64
65    *: EOF: E_WAS.ACT
Program: PRINT.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWEAN PRONGUNTHA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 19:43

Called by: MAIN.PRG

Calls: P_CODE.PRG
       : P_NAME.PRG
       : P_WAS.PRG
       : PROCESS.PRG

Documented: 12/10/92 at 20:58

SET EXCL ON
SET TALK OFF
SET STATUS OFF
SET ECHO OFF
SET SCOREBOARD OFF
CLOSE ALL
SET EXCLUSIVE OFF
IF ISCOLOR()
   normal='W/B'
   inverse='GR/W'
ELSE
   normal='Y'
   inverse='I'
ENDIF

J=1

pl = " 1. PRINT CODE OF ENTITY"
p2 = " 2. PRINT GROUP"
p3 = " 3. PRINT DATA OF ENTITY"
p4 = " 4. PRINT PROCESS"
p5 = " 5. EXIT"
err = 10
ech = "PL"
ename = SPACE(15)

r_time = TIME()
DO WHILE J < 5
SET COLOR TO &normal
CLEAR
@ 2,1 SAY "ASSUMPTION UNIVERSITY (A.U.)"
@ 3,1 SAY "INTEGRATED DATABASE"
@ 16,(COL()+1) SAY "DESIGNER"
@ 5,1 SAY REPLACE("-",79)
@ 6,1 SAY "PRINT MENU"
@ 6,65 SAY "Date: ",日期
@ 6,72 SAY DATE()
@ 7,58 SAY "BY TAWEAN PROMGUNTHA"
@ 8,1 SAY REPLACE("-",79)

@ 10,10 SAY p1
@ 11,10 SAY p2
@ 14,10 SAY p3
@ 16,10 SAY p4
@ 18,10 SAY p5

@ 19,1 SAY REPLACE("-",79)
SET COLOR TO &normal+
@ 20,10 SAY "Press "+CHR(24)+" or "+CHR(25)
@ 20, COL() SAY "to move highlight, and press "+CHR(17)+" to select"
+CHR(217)+" to select"
SET COLOR TO &inverse
@ er,10 SAY &ech
@ 22,1 SAY REPLACE("-",79)

DO WHILE INKEY() < 12
X = 0
DO WHILE X = 0
X = INKEY()
ENDDO
SET COLOR TO &normal
@ er,10 SAY &ech

GO CASE
CASE X = 24
   er = er+2
   er = IIF (er>18,10,er)
   J = VAL(RIGHT(ech,1))+1
   I = IIF (I>5,1,1)
   ech = "P"+STR(I,1)
91   SET COLOR TO Ainverse
92   @ er,10 SAY &ech
93       CASE I = 5
94       er = er-2
95       er = IIF (er<16,16,er)
96       I = VAL(RIGHT(&ech,14))-1
97       I = IIF (I<1,1,I)
98       &ech = "P"+STR(I,1)
99       SET COLOR TO Ainverse
100      @ er,10 SAY &ech
101       CASE I = 15
102       SET COLOR TO Ainverse
103      @ er,10 SAY &ech
104       SET COLOR TO Anormal
105      DO CASE
106       CASE I = 1
107           DO p_code
108           EXIT
109       CASE I = 2
110           DO p_name
111           EXIT
112       CASE I = 3
113           DO p_max
114           EXIT
115       CASE I = 4
116           DO process
117           EXIT
118       CASE I = 5
119           EXIT
120       SET COLOR TO Anormal
121       CLOSE ALL
122       ENDCASE
123       ENDCASE
124       ENDDO
125       CLEAR
126       RETURN
127       *: EOF: PRINT.ACT
Program: P_CODE.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TAWAN PROMGUNTRA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 17:48

Called by: PRINT.PRG
Dirs: AUSEFIL.DBF
Indexes: ADEX
Report Forms: P_CODE.FRM

Documented: 12/10/92 at 20:58 ForDoc version 1.0

ON ERROR RETURN
SET ECHO ON
SET STAT OFF
SET TALK OFF
SET ECCHO OFF
SET SAFE OFF
CLOSE ALL
CLEAR

@ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
@ 3,1 TO 3,79
@ 4,1 SAY "PRINT CODE DEFINE SECTION :-"
@ 4,55 SAY "ASSUMPTION UNIVERSITY"
@ 5,1 TO 5,79 DOORE
@ 5,70 SAY DATE:
driv_ = SPACE(10)
prn = 'N'
@ 9,10 SAY "INPUT DRIVE TO SAVE DATA ?" GET driv_
@ 10,10 SAY "DO YOU WANT TO PRIN ?" GET prn PICT '"'
READ
@ 9,1 CLEAR TO 18,79
 driv = LTRIM(RTRIM(driv_))="F"
usefil = driv+"FCODE_F.DBF"
fill_ = 0

-IF .NOT. FILE("AUSEFIL")
46 ?? CHR(7)
47 DO WHIL fil_ < 1000
48 @ 9,1 TO 11,60 DOVE
49 @ 9,15 SAY "***** DON'T HAVE DATA FILE *****"
50 fil_ = fil_ + 1
51 ENDDO
52 @ 6,1 CLEAR TO 11,79
53 RETURN
54 ELSE
55 CLOSE ALL
56 USE AuGetFil
57 dex = LTRIM(PTRNIM(driv_)) + "DEX"
58 INDEX ON code TO &dex
59 IF prn = 'Y'
60 REP0 FORM p_code TO PRTM
61 ELSE
62 REP0 FORM p_code
63 ENDDIF
64 WAIT
65 ENDDIF
66 CLOSE ALL
67 CLEAR
68 RETURN
69 70
71 /* EOF: P_CODE.ACT */
* Program: P_NAME.PRG

* System: INTEGRATE DATABASE DESIGNER
* Author: TAEWAN PROMGUNTHA
* Copyright (c) 1992, ASSUMPTION UNIVERSITY
* Last modified: 12/10/92 19:34

* Called by: PRINT.PRG
* User: &NAME_GR.DBF
* Indexes: &FILX
* Report Forms: P_NAME.PRN
* Documented: 12/10/92 at 20:58

"P_DEF_PRG"
SET EXCL ON
SET STAT OFF
SET ECHO OFF
SET TALK OFF
SET SAFE OFF
CLOSE ALL
CLEAR

"INTEGRATED DATABASE DESIGNER"
3,1 TO 3,79
"PRINT DEFINE GROUP SECTION :-"
4,55 SAY "ASSUMPTION UNIVERSITY"
5,79 DOUB
5,70 SAY DATE()
driv_ = SPACE(10)
7,10 SAY "INPUT DRIVE TO SAVE DATA ? " GET driv_
READ
7,1 CLEAR TO 7,79
driv=LTRIM(RTRIM(driv_))++"1"
name_tr = SPACE(7)
us_fil = SPACE(1)
prn = 'N'
8,10 SAY "PRINT GROUP IN NAME : " GET name_tr PICT '!!!!!!'
10,10 SAY "DO YOU WANT TO PRN ? " GET prn PICT '!'
READ
name_tr = LTRIM(RTRIM(name_tr))
    IF name_tr = SPACE(7)
        RETURN
    ENDIF
name_tr = driv + name_tr + " .DBF"

    IF .NOT.FILE("ANAME_GR")
        ?? CHR(7)
        11a = 0
        DO WHILE 11a < 1000
            #9,10 SAT "DON'T HAVE FILE IN DISK !!
        11a = 11a+1
        ENDDO
    RETURN
    ENDIF
fil = driv + "FCODE_F.DBF"
filx = driv + "FCODE_F"

USE Aname_tr
.INDF ON code TO fil:

    IF prin = 'Y'
        REP F FORM p_name TO PRIN
    ELSE
        REP F FORM p_name
    ENDIF
    WAIT
    CLOSE ALL
    CLEAR
    RETURN

*: EOF: F_NAME.ACT
Program: P_WAS.PRG

System: INTEGRATE DATABASE DESIGNER
Author: Tiewwan Prompuntra
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 19:37

Called by: PRINT.PRG
Uses: AFIL_USE.DBF
Indexes: AFILDEX
Report Forms: P_WAS.FRW

Documented: 12/10/92 at 20:58

SET STAT OFF
SET ECCL ON
SET TALK OFF
SET ECHO OFF
SET SAFE OFF
CLOSE ALL
CLEAR

@ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
@ 3,1 TO 3,19
@ 4,1 SAY "PRINT DATA SECTION "
@ 4,55 SAY "ASSUMPTION UNIVERSITY"
@ 5,1 TO 5,79 DOUB
@ 5,65 SAY DATE()
dri_ = SPACE(6)
prn = 'N'
@ 9,10 SAY "INPUT DRIVE SAVE DATA : " GET dri_
@ 11,10 SAY "DO YOU WANT TO PRINT : " GET prn PICT '1'
READ

@ 9,1 CLEAR TO 11,19
driv = LTRIM(RTRIM(dri_))""
@ 9,30 SAY "LOADING"
fil_use = driv + 'INTEGRATE.DBF'
fil_dex = driv + 'INTEGRATE'
IF NOT FILE("AFIL_USE")
    ?? CHR(17)
    @ 8,8 TO 11,70 DOUB
    @ 9,20 SAY "DO NOT HAVE -INTEGRATE.DBF-"
    WAIT "PRESS ANY KEY"
    @ 8,1 CLEAR TO 10,70
RETURN
-ENDIF

fil_1 = driv A "PCODE_F"
USE Afil_use
INDE ON role TO Afil_dex
    IF Pnm = 'T'
    REPO FORK p_mas TO PRIM
    ELSE
    REPO FORM p_mas
    ENDIF
    WAIT
END

*: EOF; P_MAS.ACT
Program: PROCESS.PRG

System: INTEGRATE DATABASE DESIGNER
Author: TANWAN PRONGUTRA
Copyright (c) 1992, ASSUMPTION UNIVERSITY
Last modified: 12/10/92 17:04
Called by: PRINT.PRG

User: AFIL_1
: AFIL_2
: AFIL_3.DBF
: AFIL_4
: AFIL_5
: AFIL_6.DBF
: AFIL_7

Index: AFIL_D
: AFIL_2
: AFIL_3
: AFIL_4
: AFIL_5
: AFIL_6
: AFIL_7

Documented: 12/10/92 at 20:58

"PROCESS.PRG"
SET EXCL ON
SET STAT OFF
SET TAIL OFF
SET ECHO OFF
SET SAFE OFF
CLOSE ALL
CLEAR

@ 2,1 SAY "INTEGRATED DATABASE DESIGNER"
@ 3,1 TO 3,19
@ 4,1 SAY "PROGRAM PROCESS :-"
@ 4,5 SAY "ASSUMPTION UNIVERSITY"
@ 5,1 TO 5,19 DOUB
@ 5,1 TO 5,19 DOUB
@ 5,70 SAY DATE()
 driv_ = SPACE(10)
 @1_ = SPACE(7)
 out_ = 'N'
@ 8,10 SAY "INPUT DRIVE OF DATA ? " GET driv_
READ
46 $ 10,10 SAY "NAME OF GROUP : " GET $G
47 $ 12,10 SAY "OUTPUT BASE NAME (Y/N) ? ;" GET out
48 READ
49
50 driv = LTRIM(RTRIM(driv_))"1"
51 fill_r = LTRIM(RTRIM(fill_r))"1"
52 fill_1 = driv+fill_r+".DBF"
53 fill_2 = driv+".INTEGRATE"
54 fill_3 = driv+".TEMP1"
55 fill_4 = driv+".TEMP2"
56 fill_5 = driv+".TEMP3"
57 fill_6 = driv+".TEMP4"
58
59 IF NOT FILE("Afile_1")
60 RETURN
61 ENDIF
62
63 CHR(7)
64 $ 14,10 SAY "PROGRAM WILL PRINT TO PRINTER ONLY"
65 $ 15,10 SAY "PRINTER ON"
66 $ 16,10 SAY "PRESS ANY KEY"
67 WAIT "
68 $ 14,1 CLEAR TO 16,19
69 $ 15,10 SAY " **** PROGRAM PROCESS *****"
70 SELE 1
71 USE Afile_1
72 INDE ON code TO Afile_1
73 SELE 2
74 USE Afile_2
75 INDE ON code TO Afile_2
76 COPY FOR constraint = "ONE" TO Afile_3
77 SELE 3
78 USE Afile_3
79 INDE ON code TO Afile_3
80 TOTAL ON code TO Afile_4
81 USE Afile_4
82
83 SET PRIN ON
84 ?? CHR(15)
85 ?? 
86 "INTEGRATED DATABASE DESIGNER"
87 "BY........TAWEWAN PROWONGTHA"
88 "ASSUMPTION UNIVERSITY"
89 ?? "

-213-
APE BLANK
REPL code WITH co_role WITH ro_entity WITH en_
REPL Fact WITH fo_constraint WITH co_
REPL LABEL WITH la_mandatory WITH xa_
ENDIF
SELE 2
CONT
ENDO
SELE 4
SKIP
ENDO
SELE 5
INDE ON entity TO affil
TOTAL ON entity TO affil
USE affil
INDE ON affil (LABEL) TO affil
SET RELA TO code INTO A
SELE 1
SEEK co_de
IF out = 'Y'
STOP entity TO en_name
ELSE
STOR group TO en_name
ENDIF
STOR RANGE TO ran_

? co_de
? en_name

"<" ?? LTRIM(RTRIM(ran_))
">" ?? "//
SELE 5
GO TOP
11+*
DO WHILE .NOT. EOF()
173 IF mandatory = 'Y'
?? "***
ENDIF
177 IF out = 'Y'
?? LTRIM(RTRIM(a->entity))
ELSE
?? LTRIM(RTRIM(a->group))
DATE()

IF out = 'Y'
  ? "FOR : BASE NAME"
  ?? "("
  ?? &r_
  ?? ")"
ELSE
  ? "FOR : 
  ?? &r_
ENDIF

DO WHIL .NOT. EOF()
  STOR code TO co_de
  STOR entity TO en_name
  USE en_1
  USE
  USE
  USE
  USE
  USE &f_1_3
  COPY TO &f_1_3 FOR entity = en_11 .AND. constraint = 'ONE'
  USE 4
  USE &f_1_3
  COPY STRU TO &f_1_4
  USE 5
  USE &f_1_4
  USE 4
  DO WHIL .NOT. EOF()
    STOR Fac1 TO fac_
    STOR entity TO en_na
    USE 2
    LOCA FOR Fac1 = fac_ .AND. entity = en_na
    DO WHIL .NOT. EOF()
      IF FOUND()
        STOR code TO co_
        STOR role TO ro_
        STOR entity TO en_
        STOR Fac1 TO fa_
        STOR constraint TO con_
        STOR LABEL TO la_
        STOR mandatory TO ma_
        USE 5
181 | ENDIF
182 | ?? "<"
183 | ?? LTRIM(RTRIM(a->range))
184 | ?? ">
185 | IF LABEL = 'Y'
186 | ?? "(PE)"
187 | ENDIF
188 | ?? " / "
189 | I=I+1
190 | IF I > 9
191 |
192 |
193 | ENDIF
194 | SKIP
195 | ENDDO
196 | SELE 3
197 | SKIP
198 | ENDDO
199 |
200 | -----------------------------------------
201 | 
202 | CLOSE ALL
203 |
204 | SELE 1
205 | USE &fil1_1
206 | INDE ON code TO &fil1_4
207 | SELE 2
208 | USE &fil1_2
209 | INDE ON Fact TO &fil1_2
210 | COPY FOR constraint = 'MANY_TO_MANY' TO &fil1_3
211 | SELE 3
212 | USE &fil1_3
213 | INDE ON Fact TO &fil1_3
214 | TOTAL ON Fact TO &fil1_4
215 | USE &fil1_4
216 |
217 | DO WHILE .NOT. EOF()
218 | STORE Fact TO fa_de
219 | STORE entity TO en_i
220 | STORE code TO co_de
221 | SELE 4
222 | USE
223 | SELE 2
224 | COPY TO &fil1_3 FOR Fact = fa_de .AND. entity#en_i
225 | SELE 4

-216-
USE &ii_3
INDE ON -ASC(LABEL) TO &ii_5
SET RELA TO code INTO A
SELE 1
SEEK co_de
IF out_ = 'Y'
    STOP entity TO en_name
ELSE
    STOP group TO en_name
ENDIF
STOP RANGE TO ran_
?
?
?
?
?
?

DO WHILE .NOT. EOF()
    IF mandatory = 'Y'
        ***
    ENDIF
    IF out_ = 'Y'
        ?? LTRIM(RTRIM(a->entity))
    ELSE
        ?? LTRIM(RTRIM(a->group))
    ENDIF
    ?? "<" ?? LTRIM(RTRIM(a->range)) ?? ">
    IF LABEL = "Y"
        ?? "(L)"
    ENDIF
    ?? " / "
SELE 3
SKIP
ENDDO
ENDDO
12/18/92
51:2

```
271: "============================================"
272: "=" END "="
273: "="
274: " REMARK : "EF" = ME , (PE) = PRIMARY KEY"
275: EXEC
276: SQLCN(18)
277: SET PRINT OFF
278: CLOSE ALL
279: RETURN
280: ": EOF: PROCESS.ACT
```
From a NIAM Conceptual Schema into the Optimal SQL Relational Database Schema

C.M.R. Leung & G.M. Nijssen

Department of Computer Science
University of Queensland
St Lucia, Queensland
Australia 4067

Keywords and Phrases: conceptual modelling, conceptual schema, information analysis, NIAM, Optimal Normal Form, relational model, SQL, schema transformation

CR Categories: H.1.0, H.2.1, H.2.3

ABSTRACT

The design of a correct file structure of an information system requires a thorough understanding of the users' needs. Unfortunately, end-users, usually database design layers, are often excluded from this step. This paper suggests a solution which allows them to be actively involved in the relational database design stage. We first identify the relationships between the NIAM Conceptual Schema and the Relational Model. Based on these relationships, a well-defined transformation algorithm which transforms a well-formed NIAM conceptual schema into a semantically equivalent SQL relational schema in Optimal Normal Form is described.

To be presented at the 10th Australian Computer Science Conference February, 1987
1. Introduction

In 1982, the International Standard Organisation (ISO) proposed a three-schema architecture for database standardisation (ISO, 1982). According to the report, schemas for a database are divided into three levels, namely, conceptual, external and internal (See Figure 1). The conceptual schema contains, explicitly and implicitly, all the rules for a grammar which prescribes for any Universe of Discourse (UoD) or application area, which sets of fact instances and which transitions of these set of fact instances are permitted. The external schema describes a specific user view of the conceptual schema. The internal schema, in turn, is the schema which prescribes how the sets of facts are represented in internal storage and which access mechanisms are used.

Since the conceptual schema is used to describe the users' UoD, it must possess the following properties: (1) It is internal schema independent; that is, no knowledge of how the facts are to be stored should be reflected in the conceptual schema. (2) It is complete i.e. it provides sufficient constructs to describe the UoD completely. (3) It is natural; during the process of conceptual modelling, information found in the UoD can be readily represented in the conceptual schema without requiring any conversion. (4) The modelling method is simple and easy to learn. The fourth property is found to be most important because it is usually the case that end-users, and not database designers, know the UoD best. Therefore, it is of great value that the end-users can carry out the conceptual modelling.

Once a UoD is described in terms of a conceptual schema, it is desirable to have some mechanism for deriving the corresponding internal schema (in order to achieve a production database system). This mechanism is known as transformation (Dumpala and
Arora, 1983; Navathe and Cheng, 1983). The transformation must be well-defined in the sense that no semantics are lost during the process.

The following sections discuss a well-defined algorithm which transforms a well-formed NIAM conceptual schema into an SQL Optimal Normal Form (ONF) database schema. Relations in an ONF schema are at least in 5th Normal Form (5NF). The NIAM Conceptual Schema is selected to describe the UoD as it fulfills all the requirements stated above. In addition, the NIAM Conceptual Model is selected over other conceptual models due to the fact that the major components of the NIAM Conceptual Schema are presented in an easy-to-read graphical form. Therefore, it can be easily understood by everyone involved in the design stage. On the other hand, SQL (a relational language) is selected for describing the internal schema due to the fact that it has become a de-facto standard for relational languages.

The basic concepts of the NIAM Conceptual Schema are briefly explained in Section 2. Two formal definitions for the naming of entity categories are also given. Section 3 describes four properties of the NIAM Conceptual Schema. Section 4 discusses the semantics of fact categories in terms of functional dependency. Two theorems are defined in this section. A formal definition of an ONF schema is given in Section 5. Section 6 presents the well-defined transformation algorithm. Conclusions are drawn in Section 7. A formal proof that the transformation algorithm will always produce an ONF schema is included as Appendix A. Finally, an example of the transformation is given in Appendix B.

2. The NIAM Conceptual Model

In the NIAM Conceptual Model, a conceptual schema consists of Elementary Fact Categories, Entity Categories and Subcategories, LABEL Categories, Normal Fact Categories and Constraints or Validation Rules (Nijssen, 1985). The graphical notation used in the NIAM Conceptual Model can be found in Figure 2. Figure 3 depicts a simple example which will be used extensively throughout the discussion in this section.
An entity category is a generic collection of abstract or real entities (e.g., Person, Male, Female, Sex, Hobby, Couple, Mar-year, Nr-children). A label category has label instances which are used to identify an individual entity within an entity category (e.g., Surname, Initials, M-or-F, Hby-name, Year-nr, Number). Therefore, each occurrence of an entity category in the conceptual schema must have at least one unique name (called the identifier). Object category is the generic name given to entity categories and label categories. Elementary fact categories are used to represent associations between two or more object categories through the use of roles (or verbs) (e.g., Person has Surname, Male is-husb-of Female). The roles (is-husb-of) of an elementary fact category (Marriage) show the involvement of the object categories (Male) in the elementary fact category. Subcategories are used to specify complex hierarchies of entity categories. A subcategory inherits all the fact categories of its super-category(ies) (e.g., Male and Female are subtypes of Person). A nested fact category is always treated as an entity category when it is involved in other elementary fact categories (e.g., the fact category Marriage is treated as an entity category called Couple.) A nested fact category behaves just like an ordinary entity category (e.g., a Couple married in a certain Mar-year).

Constraints or Validation Rules are conditions that ensure the consistency and integrity of the database at all times. The declarative constraints found in the NIAM
Conceptual Schema are: uniqueness constraints, mandatory fact constraints, exclusion constraints, equality constraints and subset constraints. There are two types of uniqueness constraints, namely, inter-fact category and intra-fact category.

![Diagram of conceptual schema](image)

**Figure 3. An Example**

An *intra-fact category uniqueness constraint* specifies the minimum subset of roles in an elementary fact category that is used to uniquely identify each individual fact instance in the fact category. An *inter-fact category uniqueness constraint*, on the other hand, is defined on two or more roles each in different binary elementary fact categories having a common entity category involved. The inter-fact category uniqueness constraint is used to identify the entity category in common (e.g., A Person is identified by the combination of their Surname and Initials.)

A *mandatory fact constraint* placed on a role specifies that: if any fact instance is known about a certain entity instance, then some fact instance with the mandatory role, involving that entity instance must exist as well. Finally, exclusion, equality and subset constraints are defined between two ordered sets of roles each in different elementary fact categories having a common set of object categories involved. They specify the exclusive, equal, or subset relationships between the two sets accordingly.

The identifier of an entity category is the set of label categories which uniquely identify each instance of the entity category. An entity category, however, may be
identified in more than one way. Therefore, one will be chosen as the primary identifier. The formal definitions of the synonym and the primary identifier of entity categories are given as follow:

Definition 1: An entity category may be uniquely identified by:

(a) a label category connected to it by an 1:1 elementary fact category – the preferred name.

(b) the set of primary identifiers of object categories which are involved in the nested fact category defined on it.

(c) the set of primary identifiers of object categories which are involved in those elementary fact categories covered by an inter-fact category uniqueness constraint.

It acts as the common entity category of the constraint.

If the entity category can be identified by more than one of the above, it is said to have synonyms.

Definition 2: The primary identifier of an entity category is the only identifier if it has no synonym. If the entity category has synonyms, the one which is listed first in Definition 1 acts as the primary identifier.

3. Some Properties of the NIAM Conceptual Schema

In this section, some of the NIAM Conceptual Schema properties are discussed. There are four properties that are of interest for the discussion in the following sections. These properties are mainly to do with the semantic equivalence of elementary fact category representations. They are summarised and presented with illustrative examples.

Property 1: Each n-ary elementary fact category must have at least one intra-fact category uniqueness constraint defined on it. When \( n > 1 \), one of these constraints must at least cover \( n - 1 \) roles. In unary cases (\( n = 1 \)), the constraint covers the only role. This property is usually called the irreducibility of fact categories. (See any example in this paper.)

Property 2: An n-ary (\( n > 2 \)) elementary fact category with an intra-fact category
uniqueness constraint covering \( n - 1 \) roles is semantically equivalent to a binary fact category with one role involving an \((n-2)\)-level nesting entity category; and, the other role involving the entity category which is not covered by the uniqueness constraint. The role involving the nesting entity category is covered by an intra-fact category uniqueness constraint. The \( m \)-level nesting entity category is formed by nesting a binary fact that involves an entity category which is covered by the original uniqueness constraint and an \((m-1)\)-level nesting entity category. However, for \( m = 1 \), the nested binary fact category involves two entity categories which are covered by the original uniqueness constraint instead. Both roles in a nested binary fact category must be covered by an intra-fact category uniqueness constraint. Each role involving a nesting entity category must have a mandatory fact constraint. Each original entity category appears only once. (Figure 4 shows an example of equivalence under this property.)

![Diagram](image.png)

**Figure 4.** An Example of Property 2: schema (a) is equivalent to schema (b)

**Property 3:** An \( n \)-ary \((n > 2)\) elementary fact category with an intra-fact category uniqueness constraint covering all roles is semantically equivalent to a binary fact category with one role involving an \((n-2)\)-level nesting entity category; and, the other role involving one of the original entity categories. Both roles are covered by an intra-fact category uniqueness constraint. The \( m \)-level nesting entity category is formed by nesting a binary fact which involves an original entity category and an \((m-1)\)-level
nesting entity category. However, for \( m = 1 \), the nested binary fact category involves two original entity categories instead. Both roles in a nested binary fact category must be covered by an intra-fact category uniqueness constraint. Each role involving a nesting entity category must have a mandatory fact constraint. Each original entity category appears only once. (Figure 5 shows an example of equivalence under this property.)

Figure 5. An Example of Property 3: schema (a) is equivalent to schema (b)

Property 4: An unary (\( n = 1 \)) elementary fact category with an intra-fact category uniqueness constraint covering the role is semantically equivalent to a binary fact category which is formed by adding a co-role to the unary fact category. The co-role involves a special entity category, EC-YES NO, which is identified by a label category, LC-YES-NO. LC-YES-NO has yes and no as its only instances. A mandatory fact constraint is added to the original role. The presence of a yes (no) in the co-role indicates the involvement (exclusion) of that entity instance in the original unary fact category. (Figure 6 shows an example of equivalence under this property.)

Figure 6. An Example of Property 4: schema (a) is equivalent to schema (b)
4. Semantics of Elementary Fact Categories

In order to establish the relationship between the NIAM Conceptual Schema and the relational model, the semantics of elementary fact categories are interpreted in terms of functional dependency (Date, 1981; Ullman, 1980). By Property 4 stated in the last section, each unary elementary fact category has a semantically equivalent binary form. Therefore, the following discussion on binaries can also be applied to unaries. Further, by Property 1, each n-ary (n ≥ 2) elementary fact category must have an intra-fact uniqueness constraint covering n−1 roles or all roles. Hence, the discussion that follows will be concentrated on these cases only. Two theorems are defined one for each of these two cases. The proof for each theorem is also given.

Theorem 1: An n-ary (n ≥ 2) elementary fact category with an intra-fact category uniqueness constraint covering all roles represents one multivalued dependent (MVD) fact of an entity. □

Proof. When an intra-fact category uniqueness constraint covers all roles in an n-ary (n ≥ 2) fact category, it specifies that instances of the fact category cannot be duplicated. In binary cases (when n = 2), the interpretation that the fact category represents an MVD fact is apparent. In Figure 3, for example, the uniqueness constraint defined on the fact category Person-hobby specifies that a Person can have any number of Hobbies and a Hobby can be held by more than one person. The fact, therefore, is an MVD fact about Person, or, alternatively, an MVD fact about Hobby.

In n-ary cases (when n > 2), the same interpretation is not readily seen. However, by Property 3, any n-ary elementary fact category with an intra-fact category uniqueness constraint covering all roles can be represented semantically in binary form. The two roles of the equivalent binary fact category are together covered by an intra-fact category constraint. Therefore, these n-ary elementary fact categories can always be interpreted as binaries. Hence, an n-ary (n ≥ 2) fact category with an intra-fact category uniqueness constraint covering all roles always represents an MVD of an entity. □

Theorem 2: An n-ary (n ≥ 2) elementary fact category with an intra-fact category
uniqueness constraint covering \( n - 1 \) roles represents one functional dependent (FD) fact of an entity. \( \square \)

**Proof.** Although the following proof will only be on binary cases, it should also be valid to fact categories with a number of roles greater than two (by Property 2). Again, this theorem is trivial in binary cases. It is because an intra-fact category uniqueness constraint defined on one of the two roles in a binary fact category specifies that the entity involved in the covering role functionally determines the entity in the other role. Hence, the theorem is proved. \( \square \)

5. **ONF Schema**

A relational database schema is said to be in ONF if and only if it is redundancy free and has a minimal number of relations. Relations in an ONF schema are at least in 5th Normal Form (Date, 1981; Kent, 1983). It should be noticed that the definition of ONF is a higher level of Normal Forms because of the requirement that the number of relations in an ONF schema must always be minimal. A formal definition for ONF is given as Definition 3 below.

**Definition 3:** A relational schema is said to be in Optimal Normal Form (ONF) iff

(a) There exists no repeated attributes and/or groups in all relations defined in the schema.

(b) All relations defined in the schema are free of any redundancy and update anomaly.
   (Note: (a) and (b) imply that all relations are at least in 5NF.)

(c) The number of relations in the schema is minimal with respect to (a) and (b). \( \square \)

6. **The Transformation Algorithm**

With the semantics of elementary fact categories described in Section 4, the algorithm to transform a NIAM Conceptual Schema into an SQL relational schema is described below. Theoretically, the transformation algorithm works in two steps. First, each elementary fact category found in a NIAM conceptual schema is transformed to a relation in the SQL syntax (IBM, 1983). Each attribute, possibly composite, in the
generated relation corresponds to one object category involved in the fact. The key of the relation covers all attributes corresponding to those object categories covered by the identifier of the fact category. As a result, a relation, semantically equivalent to the elementary fact category, is generated. The equivalence can be readily seen with the theorems stated in Section 4. Besides, each generated relation is at least in 5NF because it represents either an FD fact or an MVD fact.

Secondly, although relations generated in the above algorithm are in 5NF, the number of relations may possibly not be minimal. In order to arrive at an ONF relational schema, a mechanism to replace relations (only binaries) by joining them together is defined. Basically, relations with the same key are replaced by the relation formed by natural joining of them over the same key. This joining mechanism, in essence, minimises the number of relations and, at the same time, preserves the normalisation of the replaced relations.

Moreover, the number of relations is further minimised by avoiding the generation of unary relations from unary fact categories. With Property 4, all unary fact categories are transformed into equivalent binaries before applying the joining mechanism.

In particular, during the transformation, all elementary fact categories connected on subcategories are pushed up to their corresponding supercategories. Although this would mean the introduction of null values to some attributes in the generated relations, it has the advantages that the number of relations is further minimised and checking procedures for the subtype associations are simplified.

A formal proof which shows that the resulting relational schema is always in ONF is given in Appendix A. The transformation algorithm is summarised in the following 8 steps:

1. Transform all unary fact categories into semantically equivalent binaries.
2. Reconnect all fact categories connected to subcategories to their corresponding highest level super-categories. Mandatory fact constraints, if any, are removed from the roles being reconnected.
(3) Ignore all fact categories which act as identifiers or synonyms of entity categories unless otherwise stated in the algorithm.

(4) Create a relation for each elementary fact category with an intra-fact category uniqueness constraint covering all roles. Each object category involved in the fact category corresponds to an attribute in the relation. All attributes are declared as mandatory (in SQL, NOT NULL). A unique index covering all attributes is defined on the relation as the primary key.

(5) Create a relation for each n-ary \((n > 2)\) fact category with an intra-fact category uniqueness constraint covering \(n - 1\) roles. Each entity involved in the fact category corresponds to an attribute in the relation. All attributes are declared as mandatory. A unique index covering all attributes that correspond to those object categories covered by the uniqueness constraint is defined on the relation.

(6) Group the remaining binary fact categories with unique single role (i.e. the intra-fact category uniqueness constraint covers only one role). The grouping criterion is to take all elementary fact categories whose identifying role is played by the same entity (called the common entity category) to form a group. For the case that an elementary fact category has two unique single roles, the elementary fact category is grouped to the entity category covering by a mandatory fact constraint. However, if a tie occurs or no mandatory fact constraint is defined on either role, the grouping for this elementary fact category is arbitrary and is decided by alphabetical sequence (for convenience).

(7) Create a relation for each group formed in step (6) with the following attributes: An attribute which corresponds to the common entity category is created and declared as mandatory. If the common entity category has synonym(s), an attribute specified as mandatory is generated corresponding to each synonym. One attribute is created corresponding to each object category which is associated with the common entity category through the grouped elementary fact categories. Each of these attributes is declared as mandatory or optional depending on whether a mandatory fact
constraint is defined on the unique single role or not. A unique index is defined for the attribute corresponding to the common entity category. Also, one unique index is defined for each synonym attribute (if any).

(8) Create a relation for each entity category which has synonym(s) but does not act as a common entity category in step 6. Attributes are created for the primary identifier as well as for each synonym. All attributes are specified as mandatory. Unique indexes are declared for the attribute corresponding to the identifier of the entity and for the attribute corresponding to each synonym.

It is important to note that attributes referred in the above algorithm may be composite. This happens when an entity category is identified by two or more object categories. A composite attribute, essentially, is represented by more than one column in an SQL table. Furthermore, the data type of each column generated in the above algorithm is the same as that of the primary identifier of the entity category (if the column corresponds to an entity category) or the same as that of the label category (if the column corresponds to a label category).

Furthermore, different naming conventions can be used to name columns and tables generated in the above algorithm provided that the names do not violate any SQL rules. The choice of one name over the others is just a matter of taste.

Finally, as an example, the algorithm is applied to the conceptual schema in Figure 5. The SQL commands for creating the resulting database are given in Appendix B.

7. Conclusion

In this paper, we have presented a transformation algorithm which transforms a well-formed NIAM conceptual schema into an SQL ONF database schema (an internal schema). The transformation is based on a functional mapping from elementary fact categories in a NIAM conceptual schema to relations (tables) in an SQL schema. With this transformation, the database design process will be simplified. Having the description of a UoD expressed in natural language, end-users (or database designers) first go through the Conceptual Schema Design Procedures (Nijssen, 1985) to produce a
well-formed NIAM conceptual schema. By applying the transformation algorithm to the conceptual schema, the relational tables for a prototype SQL database system are derived. The derived relational schema has the properties that it is normalised and has a minimal number of relations.

Since the algorithm is simple, systematic, well-defined and easy to apply, the modelling process can be easily done by database designers as well as end-users. For a small or medium size UoD, an ONF schema can be easily and correctly generated from a user-defined conceptual schema by hand. However, if the UoD covers many fact categories, a fully computerised system which carries out the transformation is desirable.

Some constraints (exclusion, subset and equality) defined in the NIAM Conceptual Schema, however, are excluded from the transformation algorithm. The reason for the exclusion is that there exists no validation rules in the current release of SQL that can directly or indirectly describe the existence of these constraints. For instance, the integrity assertions and triggers found in SQL in System R (Chamberlin et al., 1976) which cover most of these cases are not supported in the current release of SQL. In practice, the inclusion of these validation rules in the database schema would imply better database management and maintenance.

The transformation of a NIAM Conceptual Schema into any other database models would be an extension of the transformation defined in this paper. Finally, the transformation algorithm presented in this paper represents a first step towards the automation of information analysis and system design.

Acknowledgements

The authors would like to thank Mark McLoughlin and the anonymous referees for their comments on the presentation of this paper.

References


