# DEVELOPING A DATABASE (VISUAL FOXPRO 6.0) APPLICATION PACKAGE FOR EFFECTIVE GRAINS RECORDS:

A CASE STUDY OF THE SGR SILO SYSTEM

BY

ALEBODE ISEDU PGD/SSS/MCS/97/98/550

Being a Project submitted
In Partial Fulfilment of the Requirement for the
Award of Post Graduate Diploma in Computer Science

To

The Department of Mathematic/Computer Science of the Federal University of Technology Minna, Niger State

# CERTIFICATION

This is to certify that this research work was carried out by Alebode Isedu egistration No. PGD/SSS/MCS/97/98/550 of the Department of Iathematics/Computer Science, Federal University of Technology Minna, Niger tate.

Dr. Y. M. Aiyesimi Project Supervisor	Date
Dr. S. A. Reju Head of Department	Date
External Examiner	Date

# **DEDICATION**

This project is dedicated the quest of knowledge in information technology and to my loving wife, Mrs. Alexandra A'shetu.

# **ACKNOWLEDGEMENT**

I thank the almighty God for the inspiration I had to start the programme and for the nending guidance and blessings He had continued to shower, including those I did not even otice. May the Lord the almighty Father be praised now and evermore - Amen.

I thank my Project Supervior, Dr. Yomi M. Aiyesimi for his committed and general good will that propelled my completion of the project satisfactorily. I appreciate the ndividual efforts of all the Lecturers of the Maths and Computer Science Department and he resource persons who were involved in my course of studies in the Federal University of Technology, Minna for the PGD in Computer Science Programme, whose lectures and naterial provided the first source of information gathering. I also acknowledge the singular efforts of the Class Representative, Engr. Musa Yahaya for bridging the information gap narticularly for those of us who were based in Abuja.

My participation in the programme would not have not been possible, it if were not or the support and encouragement of the Director (SGR), Engr. E. A. Talabi, this gesture vas quite gladdening to me. Furthermore, I acknowledge the contributions of Mrs. Nwaugbe Kunbi of SGR who compiled the report, my colleagues at the office for their understanding.

I enjoyed the criticism and suggestions by Mr. Mohammed Jiya and Miss Amaka Loveline Onyeakpa.

# **ABSTRACT**

The Project "Developing a Database (Visual Foxpro 6.0) Application Package for Effective Grain Records: A case study of the SGR Silo System" has been developed principally to address the numerous problems encountered in striving to maintain accurate and reliable records at the SGR Department of the Federal Ministry of Agriculture. The essence of reliable records cannot be overreinphased. The use of Micro Soft Visual Basic 6 was adopted in view of its comparative advantages over the other database packages and the easy accessibility by the developer.

The Analysis of the existing system was carried out. The evolution of record keeping at SGR Department from the use of manual typewriters for compilation of records and the use of calculations to carryout arithmetic operation, to the use of electronic spreadsheet which substantially reduced drudgery and very low efficiency association with the earlier method. However, the use of the electronic spreadsheet was deficient in terms of duplication of record, high data redundancy and inconsistency, data update were often very cumbersome.

The developed system is a database application package it is user friendly incorporated with menus and submenus, and provides a rich pool of pre-determined information or processed to those processed through the provision on request parameters. The security measure was to deny unauthorised users access to the system. The cost benefit analysis carried out indicates that the proposed system is more efficient than the current one, it requires less effort and resources. For instance it will take about 508hrs at about \$\frac{1}{3}7,153.12\$ on the new package while it will take about 1,320hrs and at about \$\frac{1}{3}39,317.54\$ on the old system to do the same work. The developed application is considered a stage change-over of conversion to the new system, however other aspects of computerisation of the entire SGR Silo Complex Management Information System will follow.

# TABLE OF CONTENT

DEDICATION  ACKNOWLEDGEMENT  ABSTRACT  TABLE OF CONTENT  CHAPTER ONE  1.1 Preamble  1.2 Motivation  1.2.1 Buyer of Last Resort  1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)  1 2.3.4 Clipper  1	CERTIFIC	CATION	1	i
ABSTRACT	DEDICAT	TION		ii
TABLE OF CONTENT.  CHAPTER ONE  INTRODUCTION  1.1 Preamble  1.2 Motivation  1.2.1 Buyer of Last Resort  1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims.  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)  1	ACKNOW	VLEDG.	EMENT	iii
TABLE OF CONTENT.  CHAPTER ONE  INTRODUCTION  1.1 Preamble  1.2 Motivation  1.2.1 Buyer of Last Resort  1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims.  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)  1	ABSTRAC	CT		iv
CHAPTER ONE  1.1 Preamble 1.2 Motivation 1.2.1 Buyer of Last Resort 1.2.2 Need for Computerisation of the SGR Silo Program: 1.3 Aims 1.3.1 Objectives 1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW 2.1 Database 2.2 Database Management System (DBMS): 2.3 SURVEY OF DBMS PACKAGES 2.3.1 Borland Database (dbase) IV provided 2.3.2 Oracle 2.3.2 Visual FoxPro 6.0 (VFP6)  1	TABLE O	F CON	rent	V
1.1 Preamble 1.2 Motivation 1.2.1 Buyer of Last Resort 1.2.2 Need for Computerisation of the SGR Silo Program: 1.3 Aims 1.3.1 Objectives 1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW 2.1 Database 2.2 Database Management System (DBMS): 2.3 SURVEY OF DBMS PACKAGES 2.3.1 Borland Database (dbase) IV provided 2.3.2 Oracle 2.3.3. Visual FoxPro 6.0 (VFP6) 1	CHAPTE	R ONE		1
1.1 Preamble 1.2 Motivation 1.2.1 Buyer of Last Resort 1.2.2 Need for Computerisation of the SGR Silo Program: 1.3 Aims 1.3.1 Objectives 1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW 2.1 Database 2.2 Database Management System (DBMS): 2.3 SURVEY OF DBMS PACKAGES 2.3.1 Borland Database (dbase) IV provided 2.3.2 Oracle 2.3.3. Visual FoxPro 6.0 (VFP6) 1				
1.2.1 Buyer of Last Resort  1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	INTRODU	JCTIO	V	1
1.2.1 Buyer of Last Resort  1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	1.1	Pream	ble	1
1.2.2 Need for Computerisation of the SGR Silo Program:  1.3 Aims.  1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	THE RESERVE OF THE PERSON OF T			
1.3 Aims 1.3.1 Objectives 1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW 2.1 Database 2.2 Database Management System (DBMS): 2.3 SURVEY OF DBMS PACKAGES 2.3.1 Borland Database (dbase) IV provided 2.3.2 Oracle 2.3.3 Visual FoxPro 6.0 (VFP6)		1.2	2.1 Buyer of Last Resort	2
1.3.1 Objectives  1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)		1.2	Need for Computerisation of the SGR Silo Program:	3
1.4 Scope and Limitations  CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	1.3	3 Ai	ms	4
CHAPTER TWO  2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3. Visual FoxPro 6.0 (VFP6)  1		1.3	3.1 Objectives	4
2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	1.4	4 Sc	ope and Limitations	5
2.0 LITERATURE REVIEW  2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)				
2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3. Visual FoxPro 6.0 (VFP6)	CHAPTE	R TWO		6
2.1 Database  2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3. Visual FoxPro 6.0 (VFP6)				
2.2 Database Management System (DBMS):  2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	2.0 LI	TERAT	URE REVIEW	6
2.3 SURVEY OF DBMS PACKAGES  2.3.1 Borland Database (dbase) IV  provided  2.3.2 Oracle  2.3.3 Visual FoxPro 6.0 (VFP6)	2.1	l Da	atabase	7
2.3.1 Borland Database (dbase) IV  provided	2.2	2 Datab	pase Management System (DBMS):	7
provided	2.3	3 SU	JRVEY OF DBMS PACKAGES	8
2.3.2 Oracle		2.3	B.1 Borland Database (dbase) IV	8
2.3.3. Visual FoxPro 6.0 (VFP6)		pre	ovided	9
		2.3	3.2 Oracle	9
2.3.4. Clipper		2.3	3.3. Visual FoxPro 6.0 (VFP6)	. 10
		2.3	3.4. Clipper	. 13

		2.3.5 Microsoft Access 2000	13
	2.4	Advantages of Database	15
	2.5	Choice of Database Application Package	
	TRACE OF		
.0 S	YSTEM	I ANALYSIS	19
	3.1	The Existing System	
		3.1.1 Infrastructural Facilitites Problems	
	3.2	Benefits of the proposed system	
	3.3	Method of Data Collection	21
	3.4	Data Analysis	22
HA	PTER F	OUR	24
.0	SYST	TEM DEVELOPMENT	
	4.1	Designing the Project	
	4,2	Designing the Database	
		4.2.1 Tables and Views	
	4.3	Implementing/security/maintaining Date Integrity	
	4.4	Referential Integrity	
	.4.5	User Interfaces (Forms)	
	4.6	REPORTS	
	4.7	CODING, TESTING AND BUILDING APPLICATION	
	4.8	Cost Benefits Analysis	
		4.8.1 Comparative Analysis	34
HA	PTER F	TIVE	36

CON	CLUSIO	ONS AND RECOMMENDATIONS	36
5.1	Salier	nt Features of the System	36
5.2	Reco	mmendations	37
ERENC	ES		38
ANN	EX I:	Grain Query (Summary of Grains Record)	39
ANN	EX II:	Progress Report on Grain Purchase/Distribution	59
ANN	EX III:	Grain Age - For Grouping Grain Stock into Various Age Rang	ges68

## INTRODUCTION

#### 1.1 Preamble

Organisations aspire to improve their service delivery with greater confidence, through taking advantage of the opportunities created by technological advancement in computerisation. Database management system (DBMS) is an important component of this advancement. DBMS is a comprehensive software tool that allows users to create, maintain and manipulate data (specific to an organisation) to produce relevant management information.

#### 1.2 Motivation

The Strategic Grains Reserve Department (SGRD) of the Federal Ministry of Agriculture, like any other forward looking organisation, desires to improve its service delivery. The modalities of achieving enhanced management practices for SGRSD is the motivation of the developer in the research project Developing a Database (Visual FoxPro 6.0) Application Package for Effective Grains Record Keeping: A Case Study of the SGR Silo system.

The Strategic Grains Reserve Programme is the apex of a three-tier Storage Programme of the Federal Government. The programme is an essential component of the Government Food Security initiative is aimed at providing food relief in times of emergencies precipitated by natural disaster - drought, famine etc - or/war. The second is the Buffer Stock Storage Programme operated by the States and Local Government aimed at price stabilisation of food produce intra and interseasonal. While the On-farm Storage Programme is aimed at reducing food crop losses at the farm gates with improved storage technologies.

The Strategic Grain Reserve Programme has thirty-three (33) Silo complexes with a combine capacity of 811,000metric tonnes(mt). Eight (8) of them with a combined capacity of 186,000mt have been completed and are operational, while the others are at various stages of construction. The

ompleted Silo Somplexes located at Akure, Minna, Lafiagi, Irrua, Jahun, Makurdi, Gombe and Dgoja are used to stock grain during the peak of harvest. The grains are held in storage for an optimum storage period of three (3) years in metal silo under normal condition. Subsequently, they are released (sold-off) during scarcity (off peak) to cushion the effect of hunger. This results is a continuous string of activities. However, in times of emergency the grains are released at any time, as the situation dictates. Government provides the funds for the purchase of grains through normal annual budgetary allocation or through special funding arrangements.

Consequently, the scenario which evolves entails;

- reception of fund (Budgetary)
- purchase of grains with available fund
- purchased grains are held in storage
- release (sales) of grains that have attained an optimum
   storage period or required for food relief
- revenue (cash or documented financial transaction)
  made from release of grains.

This cycle of event is not only endless but varies unpredictably with the different phases of operation, the various records involved needs to be properly kept and analysed for decision making.

#### 1.2.1 Buyer of Last Resort

A recent government policy - the Buyer of Last Resort (BLR), a component of Poverty Alleviation Programme, aimed at improving the lots of the farmer (focused on adequate pricing of agricultural produce with the desire of making farming a profitable venture for farmers) necssitates the purchase of about 58,000 mt of assorted grains, part of the grain glut in the country are stocked

to the various Silo Complexes. This resulted in an increase in stock to about 35% from about 3% apacity utilisation.

Consequently, government investment in the Buyer of Last Resort (BLR) Programme which is still an on-going activity runs into hundreds of million Naira. There is need for accountability and ransparency, this can be guaranteed by accuracy of records and data integrity. Information on the overall picture of system of activities or any aspect of it for decision making or for any other purpose as may be desired. Therefore the need to have timely and accurate information cannot be over emphasised. Indeed imperative!

#### 1.2.2 Need for Computerisation of the SGR Silo Program:

The developer of the proposed database package for SGR is a foundation staff of the Department and had been directly responsible for data keeping which is serving as the nucleus of the proposed Data Bank. His experience on the evolution of data keeping of the Department has manifested into taking up the challenge of producing accurate and timely information.

At the inception of the present SGR Department in late 1993, then a Special Project Unit, grain records were compiled manually by typewriters and a 286 computer with a speed of 33MHZ and 4MB RAM, computations were done by calculator. This process was tedious, corrections were often made by erasing or using whitish correcting fluid (Tippex) or outright reproduction of document. Often the final document were not as tidy as desired, the result of the calculation could not even be relied on, as repeated cross checking often gave different results.

By 1996, the Department acquired a 486 computer, with 133MHZ and 32MB and one of the installed application package was Microsoft Excel 97, an electronic spreadsheet which was used extensively for grain record keeping and its usage had improved overtime. However, as a result of

modifications, and subsequent up dates, many versions of the master file were often inadvertently produced. A lot of useful man-hour is often spent to harmonise records. Worse still final records are often not very accurate. An information request not in the processed and stored form are often not produced promptly. These development had been quite embarassing. Efforts made to use a consultant to computerise the records of and activities of the Department to address these inadequacies in the past, failed due to lack of budgetarys, this situation has persisted to date (at least as at the time of developing this work in July/August, 2000). Consequently, these short comings and the inability to appoint a consultant to carry out a system design, have greatly motivated the application developer to take up this challenge, hence the choice of the research project.

#### 1.3 Aims

The principal aim of this project is to develop a database application package in Microsoft Visual FoxPro Version 6.0 which will be used for enhanced grain record keeping in the SGR Silo System.

# 1.3.1 Objectives

The designed software will exhibit the following features:

- i) ensure data integrity and eliminate data in consistency;
- ii) eliminate duplication;
- iii) ensure date accuracy
- iv) quick generation of information in either report or chart form;
- v) update changes in operational transactions;
- vi) develope a format for updating any existing manual records with the developed database application package.

# 1.4 Scope and Limitations

The application software on grain records is essentially a module of the SGRD activities and will cover all grain supplied, grain distributions including relevant financial records. Cummulative grains and financial records will be maintained. The software will have the capability to dictect duplicate entry and update of records. However, due to the existing method of record keeping it has not been possible to obtain the desired past records to validate the performance the developed application package.

# **CHAPTER TWO**

#### 2.0 LITERATURE REVIEW

The programming language used by a developer for a project depends on three base characteristic of the data involved viz:

- (i) Type of the date
- (ii) Volume of the data
- (iii) The type of mathematical manipulation to be performed on the date to produce results

There are different types of programming languages. These are machine language, low level languages, high level languages and data ase management system (DBMS).

Machine Language: Involves of writing programms in benary digits which is time consuming and inefficient.

Low Level Language: Consist of the use of mnemonic codes and addresses these makes programming easier than the machine codes that though both are machines oriented.

High Level Languages: were designed to overcome the limitation of the low level languages. The high level languages (HLL) are problem - oriented language, they reflect the type of problem solved rather than feature the machine. The source code or programme a written in pseudo code (English link), which is a great advantage over mnemonics. Examples of HLL

**Database Management System**: are particularly useful for software projects that involves voluminous data with little arithmetic computations to be performed on it.

Consequently, SGR Grain Records is voluminous and do not require complex mathematical manipulation. Hence the use of DBMS for SGR Grains Records is most appropriate.

#### 2.1 Database

In his book Computer Science (4th Edition) C.S. French, he defined **Database** as a single organised collection of structured data, stored with minimum duplication of data items so as to provide a consistent and controlled pool of data. This data is common to all users of the system, but it independent of programmes that use data.

#### 2.2 Database Management System (DBMS):

DBMS is a sophisticated software system that constructs, maintains and processes a database. It also provides the controlled interface between the user and the data in the base.

DBMS maintains the data in the base by, adding, deleting or amending records. It also expands the base by adding new sets of records or new data to existing record, and provides facilities for different types of file processing;

- Process a complete file (serially or sequentially)
- Process required records (selective, sequential or random)
- Retrieve individual records
   DBMS provides securtiy for data in the base. The main aspects are;
- Protecting data against unathorised access
- Safeguarding data against corruption
- Providing recovery and restart facilities after hardware or software failure

It keeps statistics of use made of the data in the base, removes redundant data. It also allows data that is frequently used to be kept in a readily accessible form so that time is save

#### 2.3 SURVEY OF DBMS PACKAGES

There are many DBMS software available today, some of the most common include

- i) Borland Database
- ii) Oracle
- iii) Microsoft Visual FoxPro 6.0
- iv) Microsoft Access
- v) Clippers

#### 2.3.1 Borland Database (dbase) IV

Excerpts from Borland dbase IV indicates the dbase package is a good relational dbase which has evolved from earlier versions. It has versions to be used to

- design database
- manipulate and edit records,
- generate reports
- perform database query
- design labels
- browse record
- brows database

The Data field can specify width the default design value as well as valid range the data that can be verified authomatically when they are entered into the field. Up to 255 field per record and a dbase can be related for more than 2 other dBases programmes.

The compilation of the procedures and object code are faster at execution time. Specifically, the dbase IV also has the following features:

- It has pop-up menu windows which can be designed
- Users designable functions up to 99 files can be opened at the same time
- There is improved indexing, common line editor, improved printer handling capabilities and faster execution
- Query by example is provided
- Full relational dbase capabilities using Structural Query Language (SQL) compatible with IBM. It is possible to perform queries or relationships that spans serval dbases without using large amount of commands.

#### 2.3.2 Oracle

Oracle relational database management from the Oracle Corporation is one of the most advaced dbase programme in the market for PC, DOS, MS-DOS. There are two(2) version of oracle. Professional Oracle runs on stand alone micro computer, while the second network station Oracle which is designed to run on computer network.

Access to a dbase constructed using oracle is achieved using an English - like Structured Query Language (SQL). SQL is used to enter, retrieve, update, delete records and add new tables into a dbase.

SQL can be used interactively or in Batch mode instead of learning to use the actual language.

Beginners can use even Easy SQL to produce menus.

Oracle was originally designed for large IBM System, but has been modified to run on micro computers. It is compatible with SQL/DS and DB2, which are IBM dbase system that runs on large IBM Computers. It is made up of several programmes including:

- i) Relational Database Management system (RDBMS)
- ii) Oracle Utilities

- iii) SQL\* Plus
- iv) SQL\* calculators
- v) SQL\* Form
- vi) SQL\* Reports
- vii) SQL\* Net
- viii) PRO\* C and
- ix) PRO \*COROL

Unfortunately, oracle required more vigorous operations procedure when compared to other dbase programmes. It however provides an extensive audit travel and facilities to audit dbase operations. It provides multiple level of access control using a combination of password and users priviledge setting.

Initial installation cost is very high. It is usually customised to exclusively suit the needs of the particular user.

#### 2.3.3. Visual FoxPro 6.0 (VFP6)

Microsoft manufacturer of the visual FoxPro 6.0 in one of it's publications summarises the new features in visual FoxPro 6.0 as giving more of everything expected in a database management system - speed, power, and flexibility. It provides object and event models that help to creat and modify applications faster and easier than ever, and allow the creation of applications for internet and intranet.

It has expanded capabilities that assits in every area of application and database development.

Specifically it exhibits the following features.

# a) Enhance Project and Database Management.

Programmers have greater control over their projects and databases, using source code

control products (Microsoft Visual Source Safe) and seeing the status of the component in the Programmer Manager. Project events are also managed through the ProjectHook Class. The database container allows several user to simultaneously creat and modify objects within this same database.

# b) Easier Application Development

The Application Wizard provides support for the enhanced Application Framework, the new ProjectHook Class features to make applications more effective. In addition the following features enhanced development experience and make it easier to add effective functionality to programmers application.

- Visual FoxPro Foundation Class makes it easier to add nearly 100 developers application.
- Component Gallery makes it easy to manage VFP6 class amd documents, and other document files that may be to the project or application.
- Application Builder allows for the addition of and to creat, add or modify tables,
   reports and forms, compile and run an application promptly.
- application components. The debugger offers tools for setting breaking points, tracting events, and logging execution code. It has an enhanced Coverage Profile Application that can be used to analyse programs, application or project for code that is actually run Coverage and Profiling to enable a developer to identify problem areas in an application, especially slaped code and performance bottleneck.

# d) Easier Table Design and expanded Data Dictionary

The Table designer provides for easy addition of indexes as fields are created and a greater variety of defaults that make designing form faster and easier are specified.

#### e) Enhanced Query and View Design

Outer joins can be specified, aliases for column specified, or selecting the top number or percentage record, right in the Query and View Designer. Also the view Designer can be used to specify expanded properties on the view field that can be done on table field.

The data dictionary helps in VFP form design

Increased Form Functionality. It supports both single document Interface (SDI) and and Multiple Document Interface (MDI) offering a wide range of option to chose from. The shortcut Menu Designer help to create shortcut menus that can be used in Right click. Event Forms and control have additional properties and methods for flexible designs.

#### e) More and Better Wizard

Apart from the Application Wizard which creats an Enhanced Application Frameworks for the project and opens application builders.

The some others are:

- Oracle Upsizing Wizard moves Dbase tables, and view to an oracle server
- Form Wizard adds a page frame to display more fields on additional pages
- Pivot Table Wizard use open dbase connectivity (ODBC) to populate a pivot table
- SQL Server Upsizing Wizard Supports declarative referential integrity

- Database Wizard uses templates to creat a dbase, tables, indexes and relationships
   between table in a new dbase.
  - h) Practical samples of Application Components Provides sample components and their coders to guide developers

# 2.3.4. Clipper

Clipper is a software development system that provides all the facilities to creat a component. The Cipper was originally designed as compiller for dbase language, but has since greatly expanded the basic command set capabilities. It allows for the dbase instruction to be executed much quickly, because it converts text base instruction closer to the machine language. Dbase programmers use it to speed up dbase programme.

Clipper has been expanded from the original dbase instruction to includes many powerful command that only clipper compillers recognises. It comes with a single text editor P.E. EXE (Programme Editor). PE is limited in scope. Other alternative considered include edit command in Dos mode.

Other executable files that comes with Clipper Compiller are DBU. EXE, a Dbase utility used to create and manipulate dbase files to monitors a software as it runs. It allow for line by line control of software programme code and make problems easier to fund.

One major advantage of the Clipper over some dbase software is the ability to generate the executable file (with extension EXE). This enables the execution file to be executed from DOS prompt, while in some other dbase, the translator needs to be purchase by the user.

#### 2.3.5 Microsoft Access 2000

Microsoft Access 2000 has the following attributes as published by Microsoft software company.

- a) Working with Data and Database Design.
  - Microsoft Access 2000 provides many features that makes working with data and designing a database even easier.
- <u>Subdatasheet</u> to view and edit relates or joined data in a table, query, or form datasheet or in a subform all from the same vien.
- Automatically fix errors by remaining folds; Automatically corrects common side
  effects that occur where forms, reports, table, queries, fields, text boxes or other
  control are rename.
- b) Forms and Reports; Access 2000 provides features which makes it easier and faster to create good looking documents.
- Group text boxes and other control Group command is used on formal menu to group related text boxes and other controls on a form or report.
- Great form and Reports for SQL Server Database. The same tool used to creat form and reports for a Microsoft Access database Access Project.
- <u>Distribute Reports to Users who do not have Microsoft Access</u>
- c) Microsoft Access report can be exported to report snapshot (.snp) file format.

  The snapshot view is used to view print and mail report snapshots.
- d) Data Access Pages: is a web page that is used to add, edit, view, or manipulate current data in Microsoft Access dbase or an SQL Server dbase.
- Analyse data and make projections: data can be analysed in different ways using
   a Pivot Table list, make projection and do complex, and view data graphically in a chart.

- View grouped records interactively: by collapsing group headess. Records can also be sort and filtered.
- e) Toolbars, menu bars, and shortcut menu: are used to organise command in a way that they can easily be found.
- f) Securing, Maintaining and Converting a Databar: It has the following feature
- Securing Access dbase with user-level security wizard
- Protect developer code with visual Basic for Application Pass Word Protection:
   Modules and modules behind forms and reports are protected by Visual Basic
   Applications (VBA) password that are created in Visual Basic Editor
- Compact Utility: Combines compacting and repairing into a single process in
   Microsoft Access dbase and Microsoft Access Project. It is safer and more effective.
- Compact automatically: Is automatically compact Microsoft Access dbase or Microsoft Access Project when compact on close is selected.
- g) Working on the Internet
- h) Working with other applications: Provides features for working with other products.
- 2.4 Advantages of Database
- a. Reduce Data Duplication.

Data collected and verified only once have little chance of inconsistency, compared with the same set of data collected separately for a number of purposes. Separate data collection results in data redundancy which is costly and can be avoidable or at least reduced

by use of Database Management System. Where data redundancy is necessory in database environment it should be known and controlled.

Reduced data duplication allow for concurrent accessing (sharing) Database by a large number of users and do not affect the performance of the computer.

#### b. Increased data Integrity

In a shared data environment there is need to control the creation, deletion and update of data to ensure its correctness controls are introduced to prevent failed transactions from leaving the database in an inconsistent state due to many users. Proper control mechanism are provide to deny access to unathorised users and provide different level of access to authorise user. This guarantee increase in security significantly.

It is easier to adopt standards in a data environment.

# c. Increase speed of implementation

Accessing database is easier because of the manipulation features of the Database Management System (DMBS).

#### d. Improved Standards

It is possible to improve standards for file access and update and improve good privacy and security features. The use of integrated systems is greatly facilitated. It is different to apply standards and conventions to run for all application developed piecemeal as in the traditional system development application.

f. Increase Date Independence: the ability to change the format of the data or the medium on which the data is held or the data structure without changing the program which uses it. Conversely, it is possible to change the logic of the programme without changing the file.

#### g. Provide a Management View

Database system provide management organisation with increase awareness for corporate view of their organisation information supplied to managers is more valuable because it is based on a comprehensive collection of data instead of files that contain only the data needed by one application.

#### i. Minimum Input Preparation

The amount of input preparation needed is minimised by the "single input" principle.

#### h. Less Programming Time

A great deal of programming time is saved because the DBMS handles the construction and processing files and retrieval of data.

# 2.5 Choice of Database Application Package

The Visual FoxPro 6 has enhanced project and database management, easier application development, improved debugging with the aid of debuggers, easier table design, enhanced query and view design, increase form functionality and more and better wizard application, and provides adequate guide to developers in various tasks.

Furthermore, Visual Foxpro is a Relational Database Management System (RDBMS) that is totally dedicated to database and support object - oriented programming. Object oriented capabilities is an advantage over Borland (dBase) IV and clipper that may be used for developing the proposed application. When compared to Oracle RDBMS it is relatively cheaper.

Although Microsoft Access is produced by the same company as Visual Foxpro, it is

easier for a developer not to border about incorporating own code as they have been in-built in the various wizard application.

In conclusion, the above are obvious reasons why Microsoft Visual Foxpro was used in developing the proposed database application.

# **CHAPTER 3**

#### 3.0 SYSTEM ANALYSIS

Generally, the underlisted steps are taken when developing an application:

- i) study of existing system to discover the procedures and the problems in it;
- ii) Gathering Data and facts including users specifications and requirements for the proposed system;
- iii) Analysis and Evaluation of the gathered information;
- iv) Benefits to be derived from the proposed systems;
- v) Requirements of the proposed system;
- vi) Cost/benefit analysis;

#### 3.1 The Existing System

The present method of grains records keeping in SGR Department though has evolved overtime from what may be described as primitive method in the light of today's technological advancement with the use of calculation in the late 1993 and compilation of record done by mere typing. This exercise was tideous, time consuming and with very low efficiency, Reliability of information could not be guaranteed. It had no tangible advantage.

- Subsequently, by 1995 the electronic spreadsheets was used for compilation and computation with progressive improvement to date (May 2000). It no doubt had substantial improvement over the early method in the following regards:
  - (i) compilation and computation of record were much easier,
  - (ii) ammendment were easily effected, nested sub-total by group were often computed to give the status situation of grain in stock by type and, or at the various locations. For

instance it was either possible to determine the sub-total per grain type per location or the sub-total by the global grain type or the total quantity of assorted grains. The financial implication could equally be determined along the same sub-totals. In spite of this seemingly excellent qualities, it exhibited the following shortcoming which were as a result of the combination of the limitations in the use of spreadsheet as an effective database software and the skills of the operating staff;

- (a) Many version of the masterfile were inadvertently produced as a result of series of inevitable ammendments, which has often resulted in inconsistency of record from the same data source. Also individual records were often duplicated or even ommitted
- (b) Data retrieval resulting in not only very high data redundancy but also in data inconsistency. For example a supplier of grains may be allocated certain quantities to supply in his letter of award which is different from that in the masterfile. This had caused a lot embarrassment to the Ministry. A lot of useful man-hours is wasted in trying to harmonise records. Over and above these, production of reliable information cannot be guaranteed. The various projects monitoring teams of Government have often complianed of variance of data between the records within the Headquarters or from the outstations;
- not only have access to the data but may also modify the data;
- (d) Update of Masterfile is often cumbersome as additional input data or modifications require a repeat of the process of sorting and sub-totalling;

#### 3.1.1 Infrastructural Facilitites Problems

The nucleus of the Data Bank is located within the same small size room with the Staff of the typing pool. Space is therefore grossly inadequate and traffic inflow of staff cannot be checked.

#### 3.2 Benefits of the proposed system

To address the shortcoming with the SGR Grains Record Keeping the proposed application will have the capacity to:

- (i) Create a single pool of data, thus reducing data redundancy which is avoidable cost;
- (ii) Duplication of record will be eliminated;
- (iii) Specific information about a beneficiary or any other type of information can quickly be retrieved;
- (iv) Security of data will be increased, access by unauthorised persons will be denied.
- (v) The software will be able to retrieve information of grain status report on either by grain type, by location and necessary update provided;
- (vi) Necessary draft will be provided when desire to provide graphical explaining of data.

#### 3.3 Method of Data Collection

Principally, data is collected by three (3) methods:

# (i) Approved list of Applicants

Applications for the supply of grain or the purchase of grains in stock are forwarded to the Honourable Minister, the applications are screened and compiled by a committee. In the case of supply, approval of the Ministerial Tenders Board is obtained, this constitute the Master list which is then passed to the Department for implementation by the Tenders Board Secretariat. Subsequently this list is circulated to the various complexes while in the case of distribution, the approved list by the

Honourable Minister constitutes the Master list and also circulated to the various complexes for their guidance in executing the programm.

# (ii) Existing Record

Data are at entracted from existing records particularly when there is need to build-up record from the earlier generations. This approach would be adopted at the conversion stage of this application.

# (iii) High Frequency, Shortwave Radio

The eight (8) Silo Complexes are linked to the Project Headquarters in Abuja with high frequency shortwave radio for easy and fast communication. Not all the Silo Complexes are provided with telephone services. Data request when need very urgently are relayed through the radio, like requesting for progress report on daily basis for supply or release of grains. The major disadvantage with this system is that data are either not well relayed or the receiver operator understands the massage differently. In this situation wrong data are bound to be recorded, which precipitates lack of data intergrity.

# (iv) Reports from the Outstations

Data can also be collected from the reports from outstations. The records from these reports are relatively more accurate compared with those received through short frequency radio. The demerits associated with this method is that it does not allow for cross checking the performances at the field.

# .4 Data Analysis

This is the process of transferring data to a form that can be stored in databank with minimum duplication. The tables obtained from forms were normalised. This entails

(removing the) repeating entities to a separate entity table making the data to be functionally dependent. The result of Normalisation are tables or entities that are atomic (can not be broken in further pieces) and contains no duplications and redundancy.

# **CHAPTER FOUR**

#### 4.0 SYSTEM DEVELOPMENT

The development of the proposed application is expected to address effectively the problems of the existing system as identified in the system analysis phase of this project in Chapter 3. Visual FoxPro relational database was to be used because of its features discussed in Chapter 2.

# 4.1 Designing the Project

The project houses all the components of the application, the database, tables, user interfaces (forms), reports, queries, menus, programs, etc. In designing the project in Visual FoxPro a combination of two methods using the application wizards to design some of the necessary features of the application with sub-directories needed and adopting customised design approach which allowed the development of the application to meet certain desired features were used.

# 4.2 Designing the Database

Two databases; grains and finance were created, they served as container for tables, views and stored procedures for the tables used. They were organised and related to tables and views for refrential integrity. They provided the architecture for storing data.

#### 2.1 Tables and Views

The table created were database files (DBF) and were designed as structures to store records. The proposed system contained among others the following tables:

- 1) Purchase table while kept complete record of all the grain supply contract;
- 2) Payment table for payments made to suppliers of grains.

- Distribution table: To kept the records grains to be released or distributed. Taking into cognisance that grains are distributed when they have attained an optimum storage period of 3 years unless in cases of emergency.
- 4) Budget table to stored the records on the annual budgetary allocation of the department (SGR).
- 5) Revenue table: To stored the records of the revenue from grain releases.

The Revolving Account at each time was obtained from the sum of the budget and the revenue less the payments made.

The static tables will included:

- (1) Locations: For locations and their identification codes.
- (2) Grain-type: For various grain types and their identification codes.

They can only be updated when there is new location or grain type. The other tables were views and results of queries.

Identifying relationship: Each contractor had a unique code that identifies him in the Purchase table which will be the same code in payment table for his payment. Therefore the relationship between the purchase table and payment table is one to one through Supplier Identification Code. The location table maintain a one-to-many relationship with purchase and distribution through location identification code because a location can have supplies from many contractors or its grain can be distributed from many places. Similarly the grainstype table maintain a one-to-many relationship with payment table and Purchase table.

# 4.3 Implementing/security/maintaining Date Integrity

The application generally restricts access to data. Unauthorised users are not allowed to have access to any data, while certain category of users have limited or no access to certain data. For instance, data entry clerk could not be able to adjust the rate of payment since the field for the rate will be disabled at the point of access, only the quantity supplied will be entered.

Protecting data integrity ensures that authorise users do not add, change, or delete data that are vital to database.

# 4.4 Referential Integrity

This is the set of rules that governs the update of records in the database of an application. It was implemented at the database level using Referential Integrity Builder. For instance, a location cannot be deleted from location static table (the parent table) while there are records in purchase or distribution tables (child tables) with that location.

Also, when a location name is changed in the location table, the change will be effected in the related records in the purchase and distribution tables automatically using the location identification code.

# 1.5 User Interfaces (Forms)

The following categories of forms were created for the application:

- (a) Forms with controls that are not bounded to data such as Information Displaying Forms.
  - (i) Welcome screen which displays the name of the application, the version, the developer and copyright information at the point of starting the application;

- (ii) The sign-off screen (goodbye) which is displayed at the point the application is shutting down on quiting by the user.
- (iii) Forms to display menu and sub-menu options for user to choose depending on the intended action.

Menus are list of options used to perform varied kind of events and may include submenus. The main menu designed for the application include Data Entry, Enquiry, Reports, System Administration and Quit. Each has submenus as outlined below:

Data entry contain all the data entry forms as Submenus, System Administration menu have users update (Add, Edit, Delete, Users record), Audit trail (Monitoring System Usage); back-up, restore as submenus. Reports and the queries as submenus of Report.

- (b) Access Form: This is displayed when the welcome screen goes off. It contain a textbox for user's name to be typed in a combo box for user to select the access levels and a text box for entering user's password which for security reasons is typed but displayed in asterisks ('\*'). The 'OK' button and "Cancel' button on form evaluate the users entries and allow access of correct and exit the application respectively. The access levels are as follow:
  - User who is only allowed to simply browse desired information/data, preview or print;
  - Date Entry Clerk can in addition to the permission granted to the user above only enter data;
  - System Adminstration can in addition to the access granted the Data Entry

    Clerk, add or delete various level of users.

- (ii) Total amount spent on purchase
- (iii) Total amount and quantities realised from distribution
- (iv) Locate a supplier, his award, amount paid to him etc.
- (v) Enquiry on a particular type of grain
- (vi) Enquire on a particular location.

#### CODING, TESTING AND BUILDING APPLICATION

A main program was written that clears the Visual FoxPro environment for the application, call the welcome screen, then the access screen and continue with the flow until the user quits. After this programme was tested and found working, it was set as the main file of the project from the project manager and the executable file of the application was built from it.

The Visual FoxPro library file which support the application at run time in any system is attached to the executable file, therefore developed package can be installed or set-up on any system with or without Visual FoxPro.

Setup wizard of Visual FoxPro was used to create installation diskettes for the application which makes installation easier on any system. The system was tested with life data generated during the just concluded with the Year 2000 BLR grain purchase, and distribution, in the first instance and were found to be working, according to specification. The result of enquiry obtained were the same with those obtained using the spread sheet in the old system. The test is subsequently being extended to past grains records for purchase and distribution. Consequently, the old system has been successfully converted to the new system. Never-the-less the process of refinement will continue as the need arises.

## Cost Benefits Analysis

8

In evaluating the cost benefit of the proposed system the ideal situation would have been to establish every activity this in absolute quantitative terms, however, the salient features which are the qualitative in nature such as obtaining information with minimum efforts and time, reduction of data redundancy, easy of data entry and so on. Never-the-less attempts have be made to reduce these qualitative to quantitative value only major activities were considered and at peak periods based on observations and the following assumptions:

- to produce a fair copy of master list for grain purchase suppliers. Equal duration is also apportioned for the compilation of grain release master list. In both cases incorporating amendments, arrangements by groups, order and nested sub-totals.
- ii) the manpower requirements was released to group of grade levels for easy of computations, only basic salaries considered
- The use of stationeries and other consumables have been approximated through educated guess 3 reams of papers costing about ₱1,000 per ream of 500 sheets therefore 1,500 sheets cost ₱3,000, this implies that a sheet cost ₱2. About 2000 pages is printed with one toner which cost about ₱10,000, this comes to about ₱5/page. This therefore translates to ₱7/page of processed report including wastage.
- iv) Office space was not costed as the Ministry is expected to provide adequate office.

- v) The cost of developing the software application is assumed from a consultant paspective and a consultant fee of \(\frac{1}{100}\),000.00 is adopted.
- vi) The developed software requires an average of 3 data entry clerks to make all entries in a network environment to take advantage of sharing of resources, using lesser time, the work scenario as in the studied system of 12 hours/day for 5 days.

  Only one supervisory officer is required.

There would be no need for the Director and full complement of staff to wait for extended hours as the entries are made faster even on stand alone systems. The desired information are obtained almost promptly. However, it is being conceeded that the Director and is full conpliment of staf could perious the report for a cumulation of 2 hours daily per staff.

- vii) Retrival of information particularly when they are complex or nested could taken about half a day of about 4 hours in the present system.
- viii) the information retrival requires just one data entry clerk and a superviors and it is very prompt, when the desired parameters are correctly chosen.
- ix) about 1/3 of paper requirement is used because it is only the desired information is retrived, as the good design benefit of developed software eliminates the need for wastage of consumables.

## 4.8.1 Comparative Analysis

Table below is a detail analysis of the present system and the proposed system.

PRESENT SYSTEM	PROPOSED SYSTEM			
a)Compilation of master list for grain purchase accomplished in 5 days of 12 hrs each 60hrs required.  i) An average of 2 Top level officer Director and Deputy Director. Average Grade Level used is 16/5 (Sixteen step 5)  №475,788/year (№55.07hrs) №6,608.17. ii) Average of 5 Senior Staff average grade level 12/5 (№295,092 or №34.15/hrs) №10,245.00.  iii) Average of 4 junior staff officers on GL. 06 (101,028 or №11.69) a total of 11 staff at 60 hrs (660 hrs) №2.805.60  №19,658.77	a) Compilation of master list (grain purchase). i) Director 2 Deputy 2 hrs/ staff perusal ₩220.28 ii) 5 Senior Staff 2 hrs Staff ₩550.70 iii) 4 Junior Staff 60hrs for the duration ₩2.805.60 ₩3,576.58			
b) Compilation of master list for grain distribution as (a) above ₩19,658.77	b) Compilation of master list (Grain distribution) Same as (a) above ₩3,576.56			
c) Consumables 3 reams of photocopying papers for printing and associated cost of consumables printing tonner ₩10,500.00	c) Consumables one ream of 500 sheets at ₩7.00 per sheet ₩3,500.00			
Retrival of information complex information requires, requires the four junior staff for 4 hours ₩187.04	d) Retrival of information is requires 1 data entry clerk ands a supervisor and is prompt cost assumed negligible ₩0.00			
Hardware and software requirements  i) 2 nos of cloned desk top computers of 400 and 500MZ, 32MB RAM, CD-ROM, 14" digital Monitor  ii) 2 Units computer Desks and chairs 470,000.00  Iii) Software Application Packages 60,000.00  - Microsoft office 2000 (integrated package consisting of Word, Excel, Access, Outlook express and Power Point  - Dr. Solomon antivirus toolkit - Microsoft visual foxpro. 6.0  100,000.00 630,000.00	e) Hardware and software requirements i) File Server for network ₹500,000.00 ii) Additional Desktop Computer (Brand name, IBM compertible) ₹250,000.00 iii) Contingency for tables chairs UPS and surge arrectors ₹200,000.00 iv) Cost of net- working ₹120,000.00 v) Provision of consultancy services ₹300,000.00 ₹1,370,000.00			

In conclusion, the proposed system is more efficient, less effort and resources are required in its daily operation. The following cases illustrates this fact.

- i) A total of 508 hours is required for its peak activities at a total cost of ₦7,153.12 while 1320 hours and at a total cost of ₦39,317.54.
- ii) A total cost of about ₦3,500.00 is spent on consumables while in the present system it is about ₦10,500.00.
- iii) Retrival of information is prompt which is at a little or nor cost while it will cost about ₩187.00 on the present system.

Although the proposed system has a high initial cost, but this is more than compensated for all the positive attributes - data reliability, timeliness in obtaining information.

## CHAPTER FIVE

#### 5.0 CONCLUSIONS AND RECOMMENDATIONS

The problems encountered in the use of earlier methods of record keeping such as:

- lack of data intergrity;
- duplication of data;
- difficulty in making appropriate ammendments (deleting, adding records, making changes as the need arise;
- redundancy arising from effecting records of payment.

All these problems have been addressed in the developed application package.

The developed application is treated as a stage change-over of conversion to the new system, other aspects of computerisation of the entire SGR Silo Complex management information system will follow:

## 5.1 Salient Features of the System

- incorporation of user friendly menu;
- rich pool of information made available to user quest for information, but accessed through menus and submenu.
- different level of security measures introduced, categories of users have
  - varying degress of access such as right to access information only;
    - right to access records only;
    - right to access, and input records and unlimited access to make global ammendment which has been reserved as the perogative of the system Administrator.

#### Recommendations

The following recommendations has been made with respect to the use of application package.

- 1. The application should be used on a 486 and above computer system;
- The application package has for now been designed for stand alone computers, since the computers have not been networked. It is strongly recommended that a Local Area Network is adopted, in order to reap the associated benefits such as shearing resources, faster and efficient method of working on voluminous data, and relatively more efficient. However, when networking has been adopted, necessary coding will be incorporated to cope with the updated system.
- 3. The application has been adopted as a module in the proposed SGR information system, other modules such as personnel, inventories, etc. will subsequently be developed.

## **REFERENCES**

Hakimi, Danladi: Lecture Notes - Database Management System, 1999

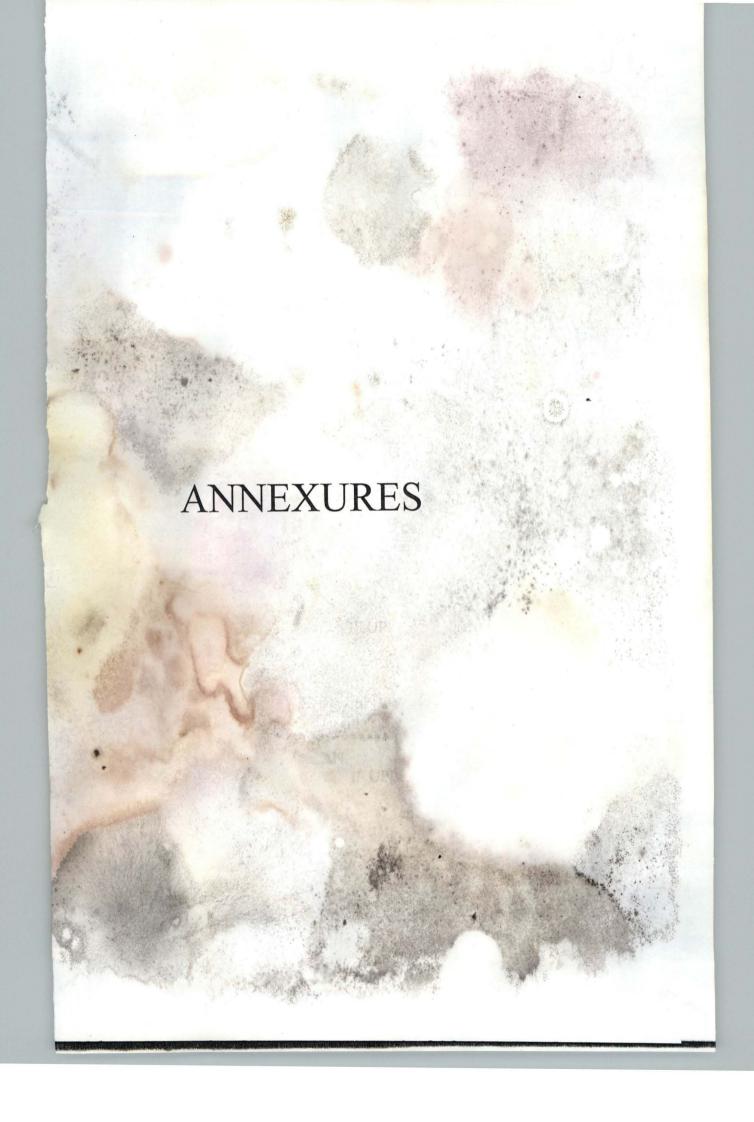
(unpublished)

- 2. Badmus R.O. and: Lecture Notes System Analysis and Design 1999
  A.B. Abadan (unpublished)
- 3. French, CS: Computer Science, 4th Edition. Publisher

DP Publication, Education Low Price Books Scheme

funded by the British Government, 1992

- 4. Microsoft Access 2000; Programm Application, Help Menu
- 5. Microsoft Visual FoxPro 6, Programmer's Guide: Microsoft Publication, 1998
- 6. Microsoft Visual FoxPro 6, Users Guide: Microsoft Publication, 1998



#### ANNEX I

## CODE FOR GRAIN QUERY (SUMMARY RECORDS)

I ) CASE

CASE THIS.VALUE=1 && IF USER CLICKS OK BUTTON DO CASE

CASE THISFORM.optgpgrain.VALUE=1 && FOR PURCHASE: THE TOTAL PAYMENTS MADE TO SUPPLIES IS COST OF GRAIN

IF THISFORM.chkall.VALUE=1

&& FOR ALL LOCATION

DO CASE

CASE THISFORM.optgpdatetype.VALUE=1

**&& FOR CURRENT DATE** 

IF USED ('suppliers\_payments')
SELE suppliers payments

**ELSE** 

USE suppliers payments IN 0

**ENDIF** 

qmaize=0

cmaize=0

qpaddy=0

cpaddy=0

qmillet=0

cmillet=0

qsorghum=0

csorghum=0

qtotal=0

ctotal=0

SCAN

IF UPPER(gn\_type)='MAIZE'
qmaize= qmaize+qty
cmaize=cmaize+award

**ENDIF** 

**ENDSCAN** 

\*\*\*\*\*\*\*\*\* MILLET\*\*\*\*\*\*

SCAN

IF UPPER(gn\_type)='MILLET'
qmillet=qmillet+qty
cmillet=cmillet+award

**ENDIF** 

**ENDSCAN** 

\*\*\*\*\*\*\*\*\* PADDY\*\*\*\*\*

SCAN

IF UPPER(gn\_type)='PADDY'
qpaddy= qpaddy+qty

```
cpaddy=cpaddy+award
            ENDIF
      ENDSCAN
                        **** SORGHUM*
      SCAN
            IF UPPER(gn type)='SORGHUM'
                  qsorghum= qsorghum+qty
                  csorghum=csorghum+award
            ENDIF
      ENDSCAN
      SCAN
            qtotal=qtotal +qty
            ctotal=ctotal+award
      ENDSCAN
      m.qmaize=qmaize
      m.cmaize=cmaize
      m.qpaddy=qpaddy
      m.cpaddy=cpaddy
      m.qsorghum=qsorghum
      m.csorghum=csorghum
      m.qmillet=qmillet
      m.cmillet=cmillet
      m.qtotal=qtotal
      m.ctotal=ctotal
      THISFORM.REFRESH
CASE THISFORM.optgpdatetype.VALUE=2
                  && FOR SPECIFIC DATE
      qmaize=0
      cmaize=0
      qpaddy=0
      cpaddy=0
      qmillet=0
      cmillet=0
```

\*\*\*\* MAIZE \*\*\*\*

IF UPPER(gn\_type)='MAIZE' qmaize= qmaize+qty cmaize=cmaize+award

SCAN FOR (supdate>=m.date1) AND supdate<=m.date2)

qsorghum=0 csorghum=0 qtotal=0 ctotal=0

```
ENDIF
ENDSCAN
******* MILLET****
SCAN FOR(supdate>=m.date1) AND supdate<=m.date2)
     IF UPPER(gn type)='MILLET'
            qmillet=qmillet+qty
           cmillet=cmillet+award
     ENDIF
ENDSCAN
********* PADDY******
SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2)
     IF UPPER(gn type)='PADDY'
            qpaddy= qpaddy+qty
            cpaddy=cpaddy+award
      ENDIF
ENDSCAN
****** SORGHUM*
SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2)
     IF UPPER(gn type)='SORGHUM'
            qsorghum= qsorghum+qty
            csorghum=csorghum+award
     ENDIF
ENDSCAN
SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2)
     qtotal=qtotal +qty
      ctotal=ctotal+award
ENDSCAN
m.qmaize=qmaize
m.cmaize=cmaize
m.qpaddy=qpaddy
m.cpaddy=cpaddy
m.qsorghum=qsorghum
m.csorghum=csorghum
m.qmillet=qmillet
m.cmillet=cmillet
m.qtotal=qtotal
m.ctotal=ctotal
THISFORM.REFRESH
                  && FOR SPECIFIC LOCATION
```

**ENDCASE** 

ELSE

```
DO CASE
                       CASE THISFORM.optgpdatetype.VALUE=1
                                                                  && FOR
                       && CURRENT DATE
                              IF USED ('suppliers payments')
                                    SELE suppliers payments
                              ELSE
                                    USE suppliers payments IN 0
                              ENDIF
                              qmaize=0
                              cmaize=0
                              qpaddy=0
                              cpaddy=0
                              qmillet=0
                              cmillet=0
                              qsorghum=0
                              csorghum=0
                              qtotal=0
                              ctotal=0
                              *WAIT WIND m.silo
                              *WAIT WIND ALLTRIM(This. VALUE)
                              ************* MAIZE ******
                              SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                    *messagebox('This is OK')
                                    IF UPPER(gn type)='MAIZE'
                                          qmaize= qmaize+qty
                                          cmaize=cmaize+award
                                    ENDIF
                              ENDSCAN
                                 ****** MILLET*
                              SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                    IF UPPER(gn type)='MILLET'
                                          qmillet= qmillet+qty
                                          cmillet=cmillet+award
                                    ENDIF
                              ENDSCAN
                              ********** PADDY*****
                              SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                    IF UPPER(gn_type)='PADDY'
```

qpaddy= qpaddy+qty

cpaddy=cpaddy+award

**ENDIF** 

**ENDSCAN** 

\*\*\*\*\*\* SORGHUM\*

SCAN FOR

UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))

IF UPPER(gn\_type)='SORGHUM'
qsorghum= qsorghum+qty
csorghum=csorghum+award

**ENDIF** 

**ENDSCAN** 

SCAN FOR

UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))

qtotal=qtotal +qty ctotal=ctotal+award

**ENDSCAN** 

m.qmaize=qmaize m.cmaize=cmaize

m.qpaddy=qpaddy

m.cpaddy=cpaddy

m.qsorghum=qsorghum

m.csorghum=csorghum

m.qmillet=qmillet

m.cmillet=cmillet

m.qtotal=qtotal

m.ctotal=ctotal

THISFORM.REFRESH

CASE THISFORM.optgpdatetype.VALUE=2

&& FOR SPECIFIC DATE

IF USED ('suppliers\_payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

qmaize=0

cmaize=0

qpaddy=0

cpaddy=0

qmillet=0

```
cmillet=0
                              qsorghum=0
                              csorghum=0
                              qtotal=0
                              ctotal=0
                              *WAIT WIND m.silo
                              *WAIT WIND ALLTRIM(This. VALUE)
                              ************* MAIZE *******
                              SCAN FOR
(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND
(supdate <= m.date2))
                                    *messagebox('This is OK')
                                    IF UPPER(gn type)='MAIZE'
                                           qmaize= qmaize+qty
                                           cmaize=cmaize+award
                                    ENDIF
                              ENDSCAN
                              ********* MILLET******
                              SCAN FOR
(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND
(supdate <= m.date2))
                                    IF UPPER(gn type)='MILLET'
                                           qmillet= qmillet+qty
                                           cmillet=cmillet+award
                                    ENDIF
                              ENDSCAN
                               ********** PADDY*****
                              SCAN FOR
(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND
(supdate <= m.date2))
                                    IF UPPER(gn type)='PADDY'
                                           qpaddy= qpaddy+qty
                                           cpaddy=cpaddy+award
                                     ENDIF
                              ENDSCAN
                                               ***** SORGHUM*
                              SCAN FOR.
(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND
(supdate <= m.date2))
                                    IF UPPER(gn type)='SORGHUM'
                                           qsorghum= qsorghum+qty
                                           csorghum=csorghum+award
                                     ENDIF
```

#### **ENDSCAN**

#### SCAN FOR

(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

qtotal=qtotal+qty ctotal=ctotal+award ENDSCAN

m.qmaize=qmaize
m.cmaize=cmaize
m.qpaddy=qpaddy
m.cpaddy=cpaddy
m.qsorghum=qsorghum
m.csorghum=csorghum
m.qmillet=qmillet
m.cmillet=cmillet
m.qtotal=qtotal
m.ctotal=ctotal
THISFORM.REFRESH

**ENDCASE** 

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ENDIF \*\*\*\*\*\*\*\*\*\*\*

CASE THISFORM.optgpgrain.VALUE=2 && FOR DISTRIBUTION
IF THISFORM.chkall.VALUE=1 && FOR ALL LOCATION
DO CASE

CASE THISFORM.optgpdatetype.VALUE=1

**&& FOR CURRENT DATE** 

IF USED ('grain\_release')

SELE grain\_release

ELSE

USE grain release IN 0

ENDIF qmaize=0

cmaize=0 gpaddy=0

cpaddy=0

qmillet=0

cmillet=0

qsorghum=0

csorghum=0

qtotal=0

ctotal=0 \*\*\*\*\* MAIZE \*\*\*\*\* SCAN IF UPPER(gn\_type)='MAIZE' qmaize= qmaize+qty cmaize=cmaize+award **ENDIF ENDSCAN** \*\*\*\*\*\*\*\*\* MILLET\*\*\*\*\*\*\* **SCAN** IF UPPER(gn\_type)='MILLET' qmillet=qmillet+qty cmillet=cmillet+award **ENDIF ENDSCAN** \*\*\*\*\*\*\* PADDY\*\*\*\*\*\* **SCAN** IF UPPER(gn\_type)='PADDY' qpaddy= qpaddy+qty \* cpaddy=cpaddy+award **ENDIF ENDSCAN** \*\*\*\*\*\* SORGHUM\* **SCAN** IF UPPER(gn type)='SORGHUM' qsorghum= qsorghum+qty csorghum=csorghum+award **ENDIF ENDSCAN** SCAN qtotal=qtotal +qty ctotal=ctotal+award **ENDSCAN** m.qmaize=qmaize m.cmaize=cmaize m.qpaddy=qpaddy m.cpaddy=cpaddy m.qsorghum=qsorghum m.csorghum=csorghum m.qmillet=qmillet

m.cmillet=cmillet m.qtotal=qtotal m.ctotal=ctotal

## THISFORM.REFRESH

CASE THISFORM.optgpdatetype.VALUE=2 && FOR SPECIFIC DATE qmaize=0 cmaize=0 qpaddy=0 cpaddy=0 qmillet=0 cmillet=0 qsorghum=0 csorghum=0 qtotal=0 ctotal=0 \*\*\*\*\*\*\*\* MAIZE \*\*\*\*\*\*\* SCAN FOR (supdate>=m.date1) AND (supdate<=m.date2) IF UPPER(gn type)='MAIZE' qmaize= qmaize+qty cmaize=cmaize+award **ENDIF ENDSCAN** \*\*\*\*\*\*\*\*\* MILLET\*\*\*\*\*\* SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2) IF UPPER(gn type)='MILLET' qmillet=qmillet+qty cmillet=cmillet+award **ENDIF ENDSCAN** \*\*\*\*\*\*\*\*\*\* PADDY\*\*\*\*\*\* SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2) IF UPPER(gn\_type)='PADDY' qpaddy= qpaddy+qty cpaddy=cpaddy+award **ENDIF** ENDSCAN \*\*\*\*\*\* SORGHUM\* SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2) IF UPPER(gn type)='SORGHUM'. qsorghum= qsorghum+qty

ENDIF

ENDSCAN

csorghum=csorghum+award

```
SCAN FOR(supdate>=m.date1) AND (supdate<=m.date2)
                  qtotal=qtotal +qty
                  ctotal=ctotal+award
            ENDSCAN
            m.qmaize=qmaize
            m.cmaize=cmaize
            m.qpaddy=qpaddy
            m.cpaddy=cpaddy
            m.qsorghum=qsorghum
            m.csorghum=csorghum
            m.qmillet=qmillet
            m.cmillet=cmillet
            m.qtotal=qtotal
            m.ctotal=ctotal
            THISFORM.REFRESH
                                && FOR SPECIFIC LOCATION
******
      CASE THISFORM.optgpdatetype.VALUE=1
            IF USED ('grain release')
                   SELE grain release
            ELSE
                   USE grain release IN 0
            ENDIF
             qmaize=0
             cmaize=0
             qpaddy=0
             cpaddy=0
             qmillet=0
             cmillet=0
             qsorghum=0
             csorghum=0
             qtotal=0
             ctotal=0
             *WAIT WIND m.silo
             *WAIT WIND ALLTRIM(This.VALUE)
```

**ENDCASE** 

DO CASE

ELSE

&& FOR CURRENT DATE

```
SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                  *messagebox('This is OK')
                                  IF UPPER(gn type)='MAIZE'
                                        qmaize= qmaize+qty
                                        cmaize=cmaize+award
                                  ENDIF
                            ENDSCAN
                                    *** MILLET**
                            SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                  IF UPPER(gn type)='MILLET'
                                        qmillet= qmillet+qty
                                        cmillet=cmillet+award
                                  ENDIF
                            ENDSCAN
                            ******
                                           PADDY***
                            SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                  IF UPPER(gn type)='PADDY'
                                        qpaddy= qpaddy+qty
                                        cpaddy=cpaddy+award
                                  ENDIF
                            ENDSCAN
                             ****** SORGHUM*
                             SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                                  IF UPPER(gn type)='SORGHUM'
                                        qsorghum= qsorghum+qty
                                        csorghum=csorghum+award
                                  ENDIF
                             ENDSCAN
                             SCAN FOR
UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))
                              qtotal=qtotal +qty
                             ctotal=ctotal+award
```

m.qmaize=qmaize m.cmaize=cmaize m.qpaddy=qpaddy m.cpaddy=cpaddy m.qsorghum=qsorghum m.csorghum=csorghum m.qmillet=qmillet m.cmillet=cmillet m.qtotal=qtotal m.ctotal=ctotal THISFORM.REFRESH

CASE THISFORM.optgpdatetype.VALUE=2 && FOR

SPECIFIC DATE

IF USED ('suppliers\_payments')
SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

qmaize=0 cmaize=0 qpaddy=0 cpaddy=0 qmillet=0 cmillet=0 qsorghum=0

csorghum=0 qtotal=0

ctotal=0

\*WAIT WIND m.silo

\*WAIT WIND ALLTRIM(This. VALUE)

\*\*\*\*\*\*\*\*\*\*\*\*\* MAIZE \*\*\*\*\*\*\*

SCAN FOR

(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

\*messagebox('This is OK')
IF UPPER(gn\_type)='MAIZE'
qmaize= qmaize+qty
cmaize=cmaize+award

**ENDIF** 

**ENDSCAN** 

\*\*\*\*\*\*\*\*\* MILLET\*\*\*\*\*\*

SCAN FOR

(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

IF UPPER(gn\_type)='MILLET'
qmillet=qmillet+qty

cmillet=cmillet+award

ENDIF

**ENDSCAN** 

\*\*\*\*\*\*\*\* PADDY\*\*\*\*\*\*

SCAN FOR

(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

IF UPPER(gn\_type)='PADDY'

qpaddy= qpa\_dy+qty

cpaddy=cpaddy+award

**ENDIF** 

**ENDSCAN** 

SCAN FOR

(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

IF UPPER(gn\_type)='SORGHUM'
qsorghum= qsorghum+qty
csorghum=csorghum+award
ENDIF

**ENDSCAN** 

**SCAN** 

FOR(UPPER(ALLTRIM(silo))=UPPER(ALLTRIM(m.silo))) AND ((supdate>=m.date1) AND (supdate<=m.date2))

qtotal=qtotal +qty ctotal=ctotal+award ENDSCAN

m.qmaize=qmaize
m.cmaize=cmaize
m.qpaddy=qpaddy
m.cpaddy=cpaddy
m.qsorghum=qsorghum
m.csorghum=csorghum
m.qmillet=qmillet
m.cmillet=cmillet
m.qtotal=qtotal
m.ctotal=ctotal
THISFORM.REFRESH

**ENDCASE** 

#### **ENDIF**

#### **ENDCASE**

```
CASE THIS. VALUE=2
                      && IF USER CLICKS PRINT PREVIEW BUTTON
      SET SAFE OFF
      PUBL alltype
      IF THISFORM.chkall.VALUE=2
            m.alltype='All Location'
      ELSE
            m.alltype=m.silo
      ENDIF
      CREATE TABLE rep FREE;
            (alltype c(20).;
            cmaize N(12,2),;
            qmaize N(12,2),;
            cmillet N(12,2),;
            qmillet N(12,2),;
            cpaddy N(12,2),;
            qpaddy N(12,2),;
            csorghum N(12,2),;
            qsorghum N(12,2),;
            ctotal N(12,2),;
            qtotal N(12,2))
      APPEND BLANK
      SET SAFE ON
      REPL cmaize WITH m.cmaize
      REPL qmaize WITH m.qmaize
      REPL cmillet WITH m.cmillet
      REPL qmillet WITH m.qmillet
      REPL cpaddy WITH m.cpaddy
      REPL gpaddy WITH m. gpaddy
      REPL csorghum WITH m.csorghum
      REPL gsorghum WITH m.gsorghum
      REPL ctotal WITH m.ctotal
      REPL gtotal WITH m.gtotal
      REPL alltype WITH m.alltype
      REPO FORM rep1 PREVIEW.
CASE THIS. VALUE=3 && IF USER CLICKS PRINT BUTTON
      SET SAFE OFF
      PUBL alltype
      IF THISFORM, chkall. VALUE=1
```

```
m.alltype='All Location'
ELSE
m.alltype=m.silo
ENDIF

CREATE TABLE rep FREE;
(alltype c(20),;
cmaize N(12,2),;
qmaize N(12,2),;
cmillet N(12,2),;
qmillet N(12,2),;
cpaddy N(12,2),;
qpaddy N(12,2),;
```



SUMMARY REPORT

I

Printed:

10/01/00

			d:	100	1000	<b>WINDS</b>				_	
			25 .			46	ne.			n.	
		543	50 m	788	555	500	J 760	55.5	100		

Location: All	LUCATION		and the second		
	Maize	Paddy	Sorghum	Millet	Total
Quantity (mt)	22 500 02		19.216.46	12,614.31	F 4 020
Cost (=N=)	23,590.83	518,36 10,911,900,00			54,939,0
Cost (-N-)	391,214,390.00	10,911,30,8.00	289,801,630.00		893,998,0
		-			
1	( )				
		Au w			
				The state of the s	
			A A Secretary	DE LESSE	
*					
					Л
		2 2 2			
		A STATE OF			
		A THE			
		1	1.00		14
	A CONTRACTOR	Alexander of the state of the s			
	177 N 315 N		<b>v</b>		
		1.65		Con the Control	**
		A Charles			
Page 1			56		

# SUMMARY REPORT

Printed: 10/01/00

Location : GOMBE

Location: GO	HDE			Life I			January 1
	Maize	Pa	addy	Sorghum	437	Millet	Total
Sale man						We Co	
Quantity (mt)		0.00	0.00	5,979	0.60	10,143.94	16,123.0
Cost (=N=)		0.00	(0.00	94,423,390	0,00	162,154,850.00	256,578,2
				Control of the Contro			
						Carlo Company	
					A PAN		
No.						4.5	
				7			
						Market St.	
	Top V	19 m 35 m					
				Ar		•	
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )		ANT AN					
	PASSING R	•		240			
							197
		71	ED.	The state of the s			
Page 1			57				19 19 19

SUMMARY REPURT

Printed:

10/01/00

#### Location: JAHUN

Location : danon					
Maize		Paddy	Sorghum	Millet	Total
					7 5727
Quantity (mt)	0.00	0.00	12,236.86	2,470.37	
Cost (=N=)	0.00	0.00	195,378,240.00	39,915,920.00	235,294,1
				C.	
	- The second sec	4			
				6	
		*			
				14	
		A Service			
				William	
Rock Market		A Committee of the Comm			
				Alvania in	
	and the second	ET CESTIONA' N			
		A DOMAN			
4		•			
	*				
The state of the state of		W. T.			
age 1		×t *			

## CODING FOR PROGRESS FORM

## ANNEX II

IF THISFORM.optgpgrain.VALUE=1 && For Purchase IF USED ('blrpurchase')

SELE blrpurchase

ELSE

USE blrpurchase IN 0

**ENDIF** 

ELSE

IF USED ('suppliers\_payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

**ENDIF** 

m.qgn type=0

m.cgn type=0

m.qgn\_type2=0

m.cgn type2=0

m.qgn type3=0

m.cgn type3=0

m.completionq=0

m.completionc=0

DO CASE

CASE THIS. VALUE=1 DO CASE && Command Button OK

CASE THISFORM.cmbgn\_type.VALUE='All\_Grains'

&& All grain type

DO CASE

CASE

THISFORM.cmbslocation.VALUE='All\_Locations' && All Locations

IF THISFORM.optgpdatetype.VALUE=1

IF USED ('blrpurchase')

<sup>\*(</sup>supdate>=m.date1) AND (supdate<=m.date2)

SELE blrpurchase

ELSE

USE blrpurchase IN 0

**ENDIF** 

SCAN

m.qgn\_type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award

ENDSCAN

IF USED ('suppliers\_payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

SCAN

m.qgn\_type2=m.qgn\_type2+qty

m.cgn type2=m.cgn type2+award

**ENDSCAN** 

m.qgn\_type3=m.qgn\_type-m.qgn\_type2

m.cgn\_type3=m.cgn\_type-m.cgn\_type2

m.completionq=(m.qgn\_type2/m.qgn\_type)\*100 m.completionc=(m.cgn\_type2/m.cgn\_type)\*100

ELSE

SCAN FOR

(supdate>=m.date1) AND

(supdate<=m.date2)

m.qgn type=m.qgn type+qty

m.cgn\_type=m.cgn\_type+award ENDSCAN

**ENDIF** 

OTHERWISE && Specify Location

IF THISFORM.optgpdatetype.VALUE=1

IF USED ('blrpurchase')

SELE blrpurchase

ELSE

USE blrpurchase IN 0

ENDIF

```
IF USED('grain release')
                               SELE grain release
                         ELSE
                               USE grain release IN 0
                         ENDIF
                         SCAN FOR (YEAR(DATE())-YEAR(grain release.datex))=2
AND (YEAR(DATE())-YEAR(grain_release.datex))<3
                               qpur=qdistr +qty
                               cpur=cdistr+ award
                         ENDSCAN.
                         m.qgn type=qpur-qdistr
                         m.cgn type=cpur-cdistr
                         m.avecost=m.cgn type/m.qgn type
                         THISFORM.REFRESH
                  CASE THISFORM.cmbage.VALUE="3yrs and above"
                  IF USED('suppliers payments')
                               SELE suppliers payments
                         ELSE
                               USE suppliers payments IN 0
                         ENDIF
                         SCAN FOR (YEAR(DATE())-
YEAR(suppliers payments.SUPDATE))>=3
                               qpur=qpur +qty
                                      cpur=cpur+award
                         ENDSCAN
                         ******
                         IF USED('grain release')
                               SELE grain release
                         ELSE
                               USE grain release IN 0
                         ENDIF
                         SCAN FOR (YEAR(DATE())-YEAR(grain release.datex))>=3
                               qpur=qdistr +qty
                               cpur=cdistr+ award
                         ENDSCAN.
                         m.qgn type qpur-qdistr
                         m.cgn type=cpur-cdistr
                         m.avecost=m.cgn_type/m.qgn_type
                          THISFORM.REFRESH
```

```
ELSE
                        && for a specific grain type do case
           DO CASE
                 CASE THISFORM.cmbage.VALUE="<less than 1year"
                 IF USED('suppliers payments')
                              SELE suppliers payments
                        ELSE
                              USE suppliers payments IN 0
                        ENDIF
                        SCAN FOR (YEAR(DATE())-YEAR(supdate))<1
                              qpur=qpur +qty
                              cpur=cpur+ award
                        ENDSCAN
                        *********
                        IF USED('grain release')
                          SELE grain release
                        ELSE
                              USE grain release IN 0
                        ENDIF
                        SCAN FOR (YEAR(DATE())-YEAR(grain release.datex))<1
                              qpur=qdistr +qty
                              cpur=cdistr+ award
                        ENDSCAN
                        m.qgn type=qpur-qdistr
                        m.cgn type=cpur-cdistr
                        m.avecost=m.cgn type/m.qgn type
                        THISFORM.REFRESH
                  CASE THISFORM.cmbage.VALUE="b/w =1 and less than 2yrs"
                  IF USED('suppliers payments')
                             SELE suppliers payments
                        ELSE
                              USE suppliers payments IN 0
                        ENDIF
                        SCAN FOR (YEAR(DATE())-
YEAR(suppliers payments.SUPDATE))>=1 AND (YEAR(DATE())-
YEAR(suppliers payments.SUPDATE))<2.
                          qpur=qpur +qty
                              cpur=cpur+award
                        ENDSCAN
                        IF USED('grain_release')
```

**ENDCASE** 

SELE grain release

ELSE

USE grain release IN 0

**ENDIF** 

SCAN FOR (YEAR(DATE())-YEAR(grain\_release.datex))>=1

AND (YEAR(DATE())-YEAR(grain\_release.datex))<2

qdistr=qdistr +qty

cdistr=cdistr+award

**ENDSCAN** 

m.qgn\_type=qpur-qdistr m.cgn\_type=cpur-cdistr m.avecost=m.cgn\_type/m.qgn\_type

#### THISFORM.REFRESH

CASE THISFORM.cmbage.VALUE="b/w = 2 & less than 3yrs" IF USED('suppliers' payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

SCAN FOR (YEAR(DATE())-

YEAR(suppliers\_payments.SUPDATE))=2 AND (YEAR(DATE())-YEAR(suppliers\_payments.SUPDATE))<3

qpur=qpur +qty cpur=cpur+award

**ENDSCAN** 

IF USED('grain\_release')

SELE grain release

ELSE

USE grain release IN 0

**ENDIF** 

SCAN FOR (YEAR(DATE())-

YEAR(grain release.datex))=2 AND (YEAR(DATE())-YEAR(grain release.datex))<3

qpur=qdistr +qty

cpur=cdistr+ award

**ENDSCAN** 

m.qgn\_type=qpur-qdistr m.cgn\_type=cpur-cdistr m.avecost=m.cgn\_type/m.qgn\_type THISFORM.REFRESH CASE THISFORM.cmbage.VALUE="3yrs and above" IF USED('suppliers\_payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

SCAN FOR (YEAR(DATE())-

YEAR(suppliers\_payments.SUPDATE))>=3

qpur=qpur +qty

cpur=cpur+award

ENDSCAN

IF USED('grain 'release')

SELE grain release

ELSE

USE grain release IN 0

**ENDIF** 

SCAN FOR (YEAR(DATE())-YEAR(grain\_release.datex))>=3

qpur=qdistr +qty

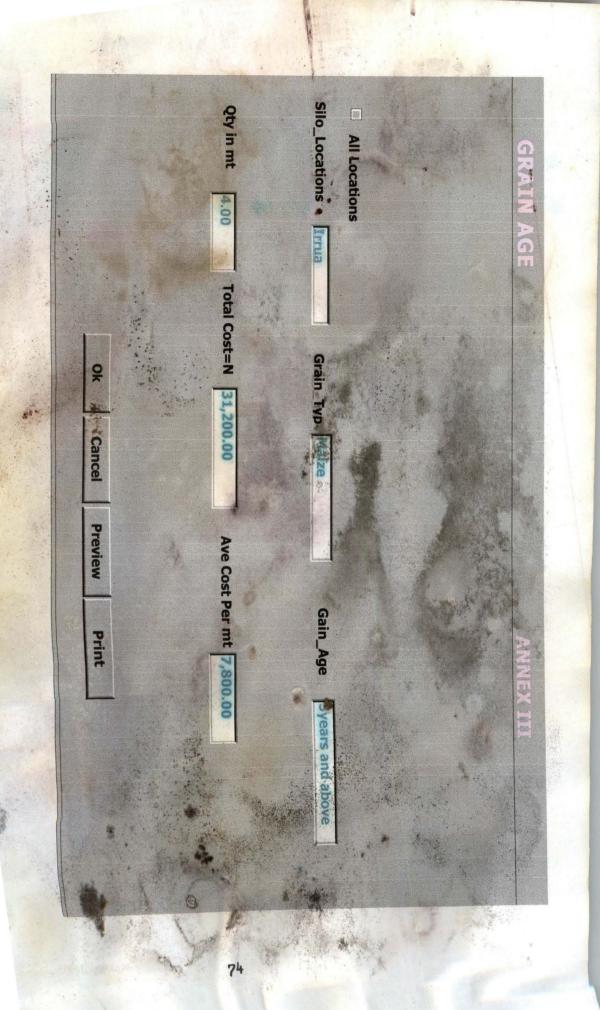
cpur=cdistr+ award

**ENDSCAN** 

m.qgn\_type=qpur-qdistr m.cgn\_type=cpur-cdistr m.avecost=m.cgn\_type/m.qgn\_type THISFORM.REFRESH

ENDCASE ENDIF && FOR GRAIN TYPE

ELSE && for specific location ENDIF



:IRRUA

ype:Maize

3yrs and above

Average Cost Cost 4.00 31200.00 7800.00 Pa 75

#### SCAN FOR

ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))

m.qgn\_type=m.qgn\_type+qty

m.cgn type=m.cgn type+award

ENDSCAN
IF USED ('suppliers payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

ENDIF

SCAN FOR

ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))

m.qgn\_type2=m.qgn\_type2+qty

m.cgn\_type2=m.cgn\_type2+award ENDSCAN

m.qgn\_type3=m.qgn\_type-m.qgn\_type2 m.cgn\_type3=m.cgn\_type-m.cgn\_type2

m.completionq=(m.qgn\_type2/m.qgn\_type)\*100

m.completionc=(m.cgn\_type2/m.cgn\_type)\*100

**ELSE** 

SCAN FOR (supdate>=m.date1)

AND (supdate<=m.date2) AND ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))

m.qgn\_type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award ENDSCAN

**ENDIF** 

**ENDCASE** 

OTHERWISE && Specific grain type
DO CASE

CASE THISFORM embslocation. VALUE='All Locations' && All Locations

IF THISFORM.optgpdatetype.VALUE=1

IF USED ('blrpurchase')

SELE blrpurchase

ELSE

USE blrpurchase IN 0

**ENDIF** 

SCAN FOR

ALLTRIM(UPPER(gn\_type))=ALLTRIM(UPPER(m.gn\_type))

m.qgn\_type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award

ENDSCAN

IF USED ('suppliers payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

SCAN FOR

ALLTRIM(UPPER(gn\_type))=ALLTRIM(UPPER(m.gn\_type))

m.qgn\_type2=m.qgn\_type2+qty

m.cgn\_type2=m.cgn\_type2+award

**ENDSCAN** 

m.qgn\_type3=m.qgn\_type-m.qgn\_type2

m.cgn\_type3=m.cgn\_type-m.cgn\_type2

m.completionq=(m.qgn\_type2/m.qgn\_type)\*100 m.completionc=(m.cgn\_type2/m.cgn\_type)\*100

ELSE

SCAN FOR (supdate>=m.date1) AND

(supdate <= m.date2)

m.qgn\_type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award

ENDSCAN

ENDIF

OTHERWISE && Specify Location

IF THISFORM.optgpdatetype.VALUE=1

IF USED ('blrpurchase')

SELE blrpurchase

FISE

USE blrpurchase IN 0

**ENDIF** 

SCAN FOR

(ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))) AND (ALLTRIM(UPPER(gn\_type))=ALLTRIM(UPPER(m.gn\_type)))

m.qgn type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award ENDSCAN IF USED ('suppliers payments')

SELE suppliers payments

ELSE

USE suppliers payments IN 0

**ENDIF** 

SCAN FOR

(ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))) AND (ALLTRIM(UPPER(gn type))=ALLTRIM(UPPER(m.gn type)))

m.qgn\_type2=m.qgn\_type2+qty

m.cgn\_type2=m.cgn\_type2+award ENDSCAN

m.qgn\_type3=m.qgn\_type-m.qgn\_type2 m.cgn\_type3=m.cgn\_type-m.cgn\_type2

m.completionq=(m.qgn type2/m.qgn type)\*100

m.completionc=(m.cgn\_type2/m.cgn\_type)\*100

ELSE

SCAN FOR (supdate>=m.date1)
AND (supdate<=m.date2) AND ALLTRIM(UPPER(silo))=ALLTRIM(UPPER(m.silo))
m.qgn\_type=m.qgn\_type+qty

m.cgn\_type=m.cgn\_type+award ENDSCAN

ENDIF

**ENDCASE** 

**ENDCASE** 

```
THISFORM.REFRESH
CASE THIS.VALUE=2
      THISFORM.RELEASE
CASE THIS. VALUE=3
                       && IF USER CLICKS PRINT PREVIEW BUTTON
      SET SAFE OFF
      *PUBL all locations, all grain
      IF THISFORM.cmbslocation.listindex=1
            m.all loactions='All loactions'
      ELSE
            m.all loactions=m.silo
      ENDIF
      IF THISFORM.cmbgn_type.listindex=1
                         m.cmbgn type='All grain'
      ELSE
            m.cmbgn type=m.gn type
      ENDIF
      CREATE TABLE tprogress FREE;
            (all locations c(20),;
            Qgn type N(20),;
            Qgn type N(12,2),;
            Qgn type2 N(12,2),;
            Cgn type2 N(12,2),;
            Qgn type3 N(12,2),;
            Cgn type3 N(12,2),;
            completionQ N(12,2),;
            completionC N(12,2))
```

REPO FORM rprogress PREVIEW

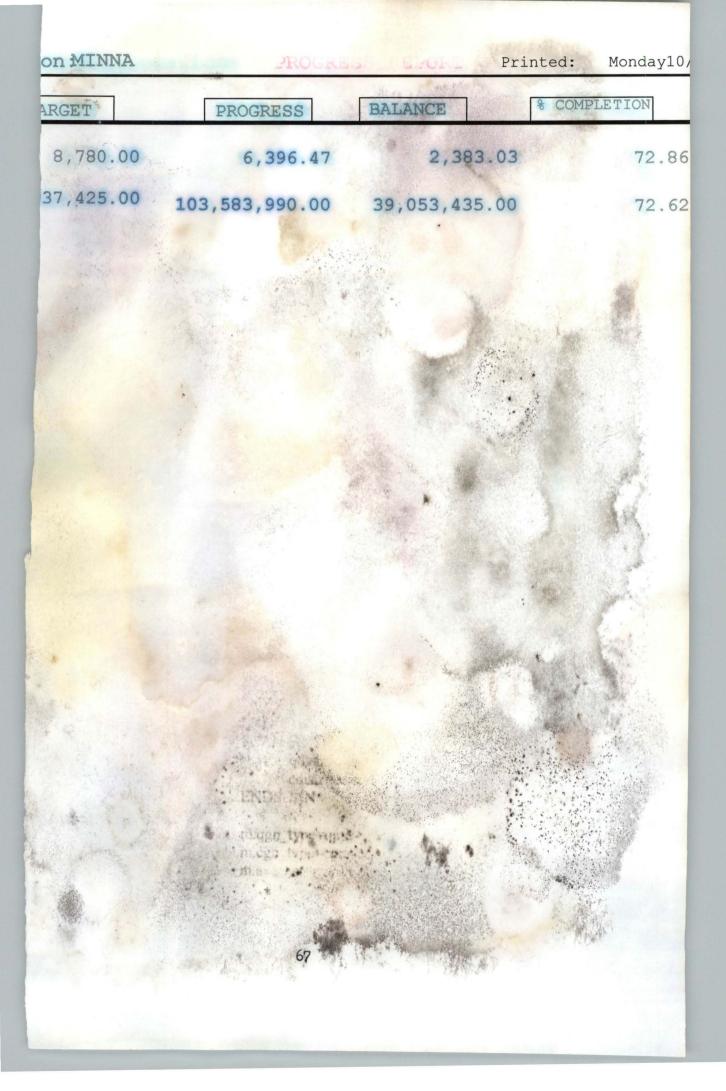
CASE THIS. VALUE=4 && IF USER CLICK PRINT BUTTON

REPO FORM rprogress TO PRINTER

**ENDCASE** 

PROGRESS REPOR			ANNEX I	
Purchase O Distribution	Silo_Location All_lo	ocations Grain_types	All_grains	
Current Date O Specific Date	Period From	To		
Target	Progress	Balance	% Complet	ion
Qty in mt 58010	54939.96	306.54	94.71	
Cost in =N 945282425.0	893998694.00	51283731.00	94.57	<u>848.</u>
	Ok Cancel	Preview Pri	nt	
				THE REAL PROPERTY.

on All Locations PROGRESS REPORT Printed: Sunday10 % COMPLETION ARGET PROGRESS BALANCE 58,010.00 54,939.96 3,069.54 94.71 282,425.00 893,998,694.00 51,283,731.00 94.57



## **CODE FOR GRAIN AGE**

## ANNEX III

qpur=0 && Qty of grains purchased cpur=0 && Cost of grains purchased cdistr=0 && Cost of grains distributed qdistr=0 && Qty of grains distributed qgn type=0 && Grain stock balance && Cost of stock balance cgn type=0 avecost=0 && Average Cost of stock balance m.qgn type=0. m.cgn type=0 m.avecost=0 IF THISFORM.cmblocations.VALUE="All Locations" && For all locations IF THISFORM.cmbgntype, VALUE="All Grains" && For all graims DO CASE CASE THISFORM.cmbage.VALUE="<less than 1 year" IF USED('suppliers payments') SELE suppliers payments ELSE USE suppliers payments IN 0 **ENDIF** SCAN FOR (YEAR(DATE())-YEAR(supdate))<1 qpur=qpur +qty cpur=cpur+ award **ENDSCAN** \*\*\*\*\*\*\* IF USED('grain release') SELE grain release ELSE USE grain release IN 0 ENDIF SCAN FOR (YEAR(DATE())-YEAR(grain release.datex))<1 qdistr=qdistr +qty edistr-edistr+ award ENDSCAN. · SCANTO m.qgn type-qpur-qdistr m.cgn type=cpur-cdistr m.avecost=m.cgn\_type/m.qgn\_type THISFORM.REFRESH

```
IF USED('suppliers payments')
                               SELE suppliers payments
                         ELSE
                               USE suppliers payments IN 0
                         ENDIF
                         SCAN FOR (YEAR(DATE())-
YEAR(suppliers_payments.SUPDATE))>=1 AND (YEAR(DATE())-
YEAR(suppliers_payments.suppliers_payments.SUPDATE))<2
                               qpur=qpur +qty
                               cpur=cpur+award
                         ENDSCAN
                         ******
                         IF USED('grain release')
                               SELE grain release
                         ELSE
                               USE grain release IN 0
                         ENDIF
                         SCAN FOR (YEAR(DATE())-YEAR(grain release.datex))>=1
AND (YEAR(DATE())-YEAR(grain release.datex))<2
                               qdistr=qdistr+qty
                               cdistr=cdistr+award
                         ENDSCAN
                         m.qgn type=qpur-qdistr
                         m.cgn type=cpur-cdistr
                         m.avecost=m.cgn type/m.qgn type
                         THISFORM.REFRESH
                         CASE THISFORM.cmbage.VALUE="b/w = 2 & less than 3vrs"
                         IF USED('suppliers payments')
                                SELE suppliers payments
                         ELSE
                               USE suppliers payments IN 0
                         ENDIF -
                         SCAN FOR (YEAR(DATE())-
YEAR(suppliers_payments.SUPDATE))=2.AND (YEAR(DATE())-
YEAR(suppliers payments.SUPDATE))<3
                               qpur=qpur +qty
                              . cpur=cpur+award
                          ENDSCAN
```

CASE THISFORM.cmbage.VALUE="b/w = 1 and less than 2yrs"