COMPUTERISED SALES ANALYSIS IN A SUPERMARKET

A CASE STUDY OF

LEOMIN SUPERMARKET, MINNA

BY

BUBA SAIDU K.
(PGD/MCS/089/96)

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF POST-GRADUATE DIPLOMA IN COMPUTER SCIENCE OF THE FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.

MARCH 1998

5 Copies

DEDICATION

This project is dedicated to the memory of my late Father, Jauro Buba who taught me to love virtue and hardwork.

ACKNOWLEDGEMENT

This project work was not something that was done overnight, but the product of many months of research, programme design and testing.

First of all, I wish to express my gratitude to the Almighty Allah, the beneficent, the merciful, for given me good health and strength to accomplished this project successful.

I wish to thank my Supervisor, Mr Ezeako of the Department of Maths/Statistic and Computer Science for his professional guidance throughout the period of the project work. So much appreciation and thanks to the Head of Department of Maths/Statistics and Computer Science, Dr. K.R. Adeboye for his professional guidance.

I wish finally to express my gratitude to all of my other lecturers for their contribution towards the successful completion of the Post Graduate Diploma (PGD) programme.

TABLE OF CONTENTS

ABSTRACT

This project work involves an analysis and design of sales activities of a super market, with reference to Leomin Super market Minna.

This was done in terms of analysing sales in order to aid the management in decision making.

The preambles, tools and scope of the study, as well as the necessary literature analysis of the existing system and overview of the proposed system, were discussed.

The analysis and design of the proposed system with reference to the problems of the existing system, the cost benefit analysis of the new system, input and output specification details of database files used, software used for design and source program documentation were highlighted.

The computer configuration and environment suitable for the new system were also discussed.

A documentation for purpose of referencing was stated, while the proposed system is recommended to any super market for use in Nigeria.

CHAPTER ONE

1.1 INTRODUCTION

In order to penetrate a market effectively a business venture need to know, the part of the market it intends to serve. A market is usually segregated into component parts, usually regarded in marketing as market segments. Some of the factors that are used in segregating the market into segments are:-

1.1.1 Environmental Factors:-

(1) Weather

(2) Climate

1.1.2 Cultural factors:-

(1) customs of the potential customers

(2) Norms of the society

1.1.3 Economic factors:-

(1) Purchasing power of the potential customers

(2) Taste of the potential customers

(3) The inflationary rate in the economy

1.1.4 Legal factors:-

(1) Tax policies, both personal and business taxes

(2) Business laws, such as business permit etc.

1.1.5 Religious factors:- A well understanding of its type of market segment by a business venture would be in a better position to formulate an appropriate marketing policies for its organisation.

An appropriate formulated and adapted marketing strategies by an organization makes it supreme over its competitors in its industry.

For a business to identify its market segment and its potential customers, sales trend analysis is the starting point. Sales trend can be said to be a pattern of sales movement over a period of time which shows the classes of potential customers from where factors enumerated above could be further analysis to finally arrived at the appropriate market segment the business is serving.

To obtain sales trend, however, for a business venture is by analysing the sales data generated over a period of time. The accuracy of the analysis result depends on the accuracy of the data so generated. It has therefore become imperative to computerise sales record of a business venture, if accurate data is to be generated. The business venture would now be better equipped to formulate appropriate marketing strategies for the overall growth of the business.

1.2 STATEMENT OF THE PROBLEM

Sales and other records for the super marketer since have been recorded manually. Analysis of the data to arrive at a sales trend have been done manually. This has been very tedious and time consuming. In some cases also, analysis result may be in accurate. hence policies made may also be inappropriate. To solve the above mentioned problem, associated with the existing system, it has become imperative to computerise sales records of the super maker, if accurate data is to be generated. The data so generated would then be accessed and analysis for the formulation of an appropriate marketing decision for the super market.

1.3 OBJECTIVES OF THE STUDY

The overall objective of the computerisation of the sales records for the super market, is to realise all the advantages associated with data base management information system, and specifically the following.

1.3.1 An up-to-date record is being maintained for each item product purchased and sold.

1.3.2 To enhance accountability for the overall performances and growth of the super market.

1.3.3 To do away with cases of pilfering with the super market records.

1.3.4 Accessibility and retrieval of records is made easier and faster for each item bought or sold.

1.4 SIGNIFICANCE OF THE COMPUTERISATION

The purpose of the computerisation is to high-light its supremacy over the manual system. Other benefits are up-to-date record keeping cost reduction from reduced labour and time. Information retrieval is faster and easy access to records is made possible.

1.5 SCOPE AND LIMITATION

Computerisation of sales record data is limited to a super marker store with particular reference to Leomin Supermarket situated at No 72, Bosso Road, Minna, Niger State.

The system is intended to provide avenue for the analysis of sales records for a given period of time on each item or product sold. The analysis result is expected to show the sale trend so that the type of market segment the business is operating within would be identified. Appropriate marketing strategies for the individual items produce and the super market in general would be in place.

CHAPTER TWO

2.1 LITERATURE REVIEW

Prosperous business ventures have always been attributed to effective management such as proper stock control and maintenances as always occur as a result of taking the right and proper inventory system.

The history of numbers and counting was traced to be first developed by herdsmen of the ancient time who sought method of taken of stock of their animals using pebbles to avoid losses. This, they did by representing an animal by a pebble in turn. These pebbles were kept in the pocket when leaving for grazing in the morning and re-counted in the evening when returning to ascertain that none was missed or left behind.

The principle behind the idea of stock taking which brought about the development of inventory system by various business organisation for instance, is to ensure that proper account of items is given at any point in time and also the right quantity for each item is maintained at all times so that no losses would be incurred as a result of lack of enough or no stock at all to meet the demand.

There are now as many write-ups as possible such as seminar papers school project, magazines and books. However, in all cases, none have written other than the present principle of inventory system which is the right quantity to maintained at each point in time and the level of quantity at which addition should be made ie Re-order level (ROL), the time lag it takes to receive orders i.e. lead time (LT).

The right quantity to stock and at what level of stock addition should be made could be ascertained, but some vital question are yet to be answered. Some of which are, how fast are the goods selling? which of the goods to stock more or less? And at what time? Answers to these questions will aid the business organisation to identify its market segment which will facilitate the formulation of appropriate business polices and marketing policies to be specific.

This study therefore is intending to address the above mentioned short comings in the inventory system. To sum it up, sales analysis would provide answer to those questions. Sales analysis is that problem where by the fast selling goods and the highest demanded as well as the quantity sold would be known. It will therefore give the organisation the opportunity to formulate appropriate advertisement for its individual goods and products.

2.2 THE EXISTING SYSTEM

The existing system is basically manual which include such as stock taking, sales recording, purchase recording and any other business activities that relates to the supermarket. Sales analysis is done only when the need arises and it is always tedious, time taking and in most cases in accurate. This is because the data used were never accurate. Advert polices in particular are made arbitrary not based on any research or analysis hence were always in appropriate because they were not with clear cut goals and objectives as it affects individual products.

All the short comings of the excising system effects the growth and the overall profit of the super market.

2.2 (i) To Further Analyse: The existing system, one would easily identify its weakness, strength and opportunities open to the super market. Some of those weaknesses are, lack of accurate up-to-date records for sales, accessibility of records is tedious and time taking, in appropriate marketing and other policies for the super market crude stock taking system which is the manual and above all the possibilities of cases of pilfering with the existing records.

2.2 (ii) The strength of the super market however, could be said to include its ability to build up its present volume of customers, sales volume and no case of pilfering have been experienced so far.

2.2 (iii) The opportunities includes its growth potentials, subsequent expansion to other parts of the state and its strategic location which is the most cosmopolitan section of Minna town ie mobile round about, Bosso Road, N 72, Minna.

2.3.1 THE BACKGROUND OF THE SUPER MARKET

Leomin Super Market was established some years back with its head office at Logos and branches in some states in the country. Minna Branch was established in 1990. The Super Market is involved in the business of buying and selling of different sorts of goods, ranging from house hold equipments, cosmetics, mens and women

wears, utensils etc, typical of a Leading stores like the UTC and Leventis, Sales value have been on the increase and its customers cut across the ladder of society both young and old. If the sales trend is sustained, the Super Market has the potentials of becoming one of the leading super markets in Minna and possibly Niger State. its mode of operation, however, are being carried out manually hence the need to computerise the sales records. This has become imperative if the current market is to be sustained and to further increase sales, analysed and the result will facilitate the formulation of appropriate marketing strategies for the supermarket.

2.3.0 PROPOSED SYSTEM

The proposed system is one that would provide for the computerisation of current manual existing system Database management system would be employed to replace existing system where all related records are grouped and maintained in files to be updated from time to time as the case may be. Access to records would be easier and faster.

The system is capable of providing different kinds of reports, which would be required for the effective management of the Super Market. Some of these report are:-

(i) The volume of sales made per item per period.

(ii) The overall sales turnover for the super market per period.

(iii) The product with the highest profit.

(iv) The overall profit margin for the Supermarket per period.

(v) Total cost of purchases for the Super Marker.

CHAPTER THREE

3.0 SYSTEM ANALYSIS AND DESIGN

3.1 INTRODUCTION

The system analysis and design stage involves analysing the existing system in order to aid the designing of the proposed system. The analysis is considered important because the design of the new system is dependant on what ever information gathered during the analysis stage.

In recognition of the above, this Chapter begins with outlining the problem associated with the existing system. This is intended to be able to design the new system such that the problem will be under control. In the design phase which has to do with the transformation of information collected, the new system is considered in terms of the input requirement, output requirement, type and objectives of the software used in the design and the description of the data files required for the new system to work conveniently.

3.2 PROBLEMS OF THE EXISTING SYSTEM

The introduction of computer in the present day world is expected to replace manual operations. This is necessary due to increased activities in our present day life. In areas where manual is still in place, there exist various problem such as loss of vital information, insecurity of data, late retrieval of necessary details and host of others which lead to inconvenience in the general operation of the system.

Specifically, the problems inherented in the existing system are listed below.

i) There is no proper storage of data in the Super Market as the recording of sales and analysis is never accurate manually.

ii) There is difficulty in knowing whether the Super market is making profit or loss

iii) The Super Market finds it difficult to ascertain the performance of each product on sale.

iv) Finally, during re-stocking of products, the Super Market finds it difficult to know the items that have gone in re-order level or minimum stock.

In recognition of the above problems, the Super market has not been able to meet its objective of profit making. The cash position of the business fluctuates and at present, the continued survival of the business cannot be assured.

3.3 THE DESIGN OF THE PROPOSED SYSTEM

The design of the proposed system is done to suit the requirement of the super market. The design takes cognizance of the software that would be applicable to the full computerisation of the existing procedures in the Super Market. Specifically, the new system is expected to be menu driven such that the user will select the appropriate menu and the expected routine would be activated. This is intended in order to make the system to be user friendly.

More importantly, the system is designed to validate data entered into the system. This would serve as a control measure in case the user enters wrong data.

3.4 COST BENEFIT ANALYSIS OF THE PROPOSED SYSTEM

The adoption of the proposed system which will require the acquisition of both the hardware and the software by the Super Market will definitely be at cost. Other cost include system failure which result to disruption of sales and eventual lost of customers to the Super Market. Also the usage of Computer consumable which include papers, ribbons and diskettes as will as maintenance of the system would also cost the super Market enormously.

The benefits and advantages of the proposed system at the Super Market however, outweigh the cost and disadvantages some of the benefits and advantages that would be realised by the Super Market include the following:-

(a) Provide up-to-date and accurate dates for sales analysis and policy formulation.

(b) Up-to-date records of sales are maintained.

(c) Easy access to records at any time of need is achieved.

(d) Effective and efficient customers services would be provided.

(e) Funds are invested appropriate among items to avoid idle funds

(f) Sales and profit margin are maximally realised.

(g) Market segments are maximally served.

(h) Competitive advantage over its competitors.

All these and other benefits would accrue to the Super Market with the adoption of the proposed system.

3.5 SYSTEM INPUT SPECIFICATION

For computer to perform the task of data processing data needs to be inputted into the system. The system input specification states the source and type of data that needs to be supplied into a system.

This is considered important because if the information supplied are correct, definitely the result of the computer processing would also be correct. It is in recognition of this that the input of the proposed system is designed to produce a cost effective method, achieve the highest level of accuracy and ensuring that the input is acceptable and understandable to the users.

Specifically, the new system requires three forms of data input into the system apart from when selecting a menu. Firstly, the registration of the detail of a new product into the system. in this case, information such as stock number, the description of the item, the cost price as well as the selling price of the item. Secondly when stock are received into the super market, the details of each of the items such as the stock code, the quantity received and the price of each of the items. Finally, the system expects another item at the point of sales where details such as the stock code, price and the quantity of sales will be inputted.

However, these data are designed with ease of data entry and informative such that the user would know the data the system expects at any point. The input form on the screen are also designed to validate most data entered before acceptance. For instance, in introducing new stock item into the system, duplicated item code would

not be allowed into the system. Furthermore, in the aspect of stock receipt and sales, the stock code must exist within the system before onward processing.

3.6 SYSTEM OUTPUT SPECIFICATION

Computer system are designed to perform the task of data processing. Once the data have been inputted, the processing begins on the data in order to generate an output which is always the result of the processing. Output, therefore, becomes important because it is the mode of communicating processing result to the end use. there are two forms of output which are hard copy and soft copy. The hard copy result are the output printed by the computer printer while the soft copy result are those displayed on the computer screen.

For the computerised sales recording and analysis of Leomin super market, there are two reports to be general namely:- the daily sales analysis and weekly sales analysis.

3.7 SOFTWARE APPLICATION

The proposed system is designed using Database Management system (DBMS). A Database is an organised collection of related information designed to meet the various needs of an oganisation or establishment. DBMS is a package of computer programs and its documentation used o create, maintain, organised and retrieve information from a database. It is a software package that help establishment or institution manage their data resources. The types of DBMS package are dBASE, Clipper, Oracle, etc.

In the design of the proposed system, a dBASE version (dBASE IV) is used because of its verstaliity. This is as a result of the in-built utilities embedded in the package.

3.8 OBJECTIVES OF DBMS

The objective of Data Base Management System (DBMS) reflected the present needs of the modern data processing environment. The over all objectives in the development of data technology has been to treat data as an orgainsational resources and as an integrated whole. Database system allow the data to be protected and organised separately from other resources (eg hardware, software and program). Specifically the objectives of database are as follows:-

1. DATA INTEGRATION:- In a database, information from several files is co-ordinated, accessed and operated upon as though it is in a single file. Logically, the information is centralised, physically, the data may be located in different files. In addition, it is possible for two or more application to be sharing compatible data.
2. DATA REDUNDANCY IS ELIMINATED :- Data redundancy occurs when the same data appears in more than one file. This leads to wastage of storage space and duplication of efforts during data entry. One basic feature of DBMS is that it eliminates data redundancy since data are not duplicated in different files.

3. DATA INDEPENDENCE:- one main objective of DBMS is data independence because application programs are isolated from the physical or logical storage of data. This objective seeks to allow for change in the content and organisation of physical data without re-programming the application.

4. DATA INTEGRITY:- Data redundancy can lead o lack of data integrity and a common system of this is inconsistent information. This means that the information generated by the data processing system can no longer be trusted. In an environment where there is duplication of data, problems concerning the up-dating and deletion of data often arise since change in any of the duplicated data will necessary a change in every file that contains the same data, otherwise, inconsistency will result. In a database system, this problem can be avoided by recording data once. This approach to data integrity results in more consistent information.

3.9. DATABASE FILES DESIGN

A database management package requires the use of database files for the storage of data. A database file is a file that contain similar records. The design of database file defines the description of all the files that will be used in the proposed system. it includes the description of the content of the files used and their structures. The structures state the field names, field type and the field width, associated to each of the files.

The proposed computerised sales recording and analysis in Leomin supermarket is designed to use database files. The description and contents of these files are as follows:-

1. CODE.DBF:- This is master file which keeps the details of all the products available in the Super Market at any point in time. It contains information such as product code, product description, cost price, selling price and quantity The format of the file is stated as follows:-

S/NO	FIELD NAME	FIELD TYPE	WIDTH	INDEX
1.	Code	character	7	N
2.	Desc	character	30	N
2.	CP	Numeric	10/2	N
4.	SP	Numeric	10/2	N
5.	Qty	Numeric	10/2	N

2. DELIVERY DBF:- This is a transaction file which contains the details of items received into the Super Market. It will contain information such as Data, Invoice Number, Code, and Quantity the format of this are as follow:-

S/NO	FIELD NAME	FIELD TYPE	WIDTH	INDEX
1.	Date	Data	8	N
2.	Inno	Character	7	N
3.	Code	Character	7	N
4.	Qty	Numeric	6	N

3. SALES.DBF:- The SALES.DBF is a database file that record the details of the daily sales. This file is always made empty at the beginning of each day in order to prepare it for the daily activities. It contains information such as Data, Customer sales No stock code, item description, quantity, price and sales amount. The format of this files is as shows below:-

S/NO	FIELD NAME	FIELD TYPE	WIDTH	INDEX
1.	Date	Date	8	N
2.	Sales No	Numeric	6	N
3.	code	Character	7	N
4.	Desc	Character	30	N
5.	Qty	Numeric	5	N
6.	Price	Numeric	9/2	N
7.	Amount	Numeric	11/2	N

4. TSALES.DBF:- This file contains similar structure with the SALES.DBF. It contains the details of sales record for a period of time. It is from this that sales analysis and reports are generated.

5. TDELVE.DBF:- This is also similar to the DELIVERY.DBF as they contain similar structure. The TDEVE.DBF contains accumulated details of stock received into the Super Market.

3.10 SOURCE PROGRAM DOCUMENTATION

The physical design of the proposed system contains the source program used in communicating with Computer. the inclusion of this is to allow for modification as the need arise. For purpose of ease of modification and maintenance, the program is designed in modules. However, the suite of programs are contained in Appendix II.

CHAPTER FOUR

4.0 SYSTEM IMPLEMENTATION AND DOCUMENTATION

4.1 INTRODUCTION

The last Chapter discussed all the necessary requirements about the design of the proposed system. Having discussed the design, there is the need to communicate the mode of working with the new system to the potential users. Furthermore, the working environment as well as the step by step introduction of the new system needs to be stated.

Having recognised the need for the above, this Chapter beings with describing the computer configuration which will be needed for the successful operation of the new system. The required software for the full implementation of computerised operations in the Super Market would also be outlined. The mode of testing and conversion of the proposed system are also highlighted in this Chapter. All these are considered essentials in order to aid the full implementation of the new system. In addition, the system documentation is also discussed in this Chapter. This is expected to aid the user in understanding the working of the new system.

4.2 HARDWARE CONFIGURATION

Hardware can be defined as the physical part of a Computer system which constitutes the mechanical, magnetic, electrical and electronic devices. The hardware configuration is a collection of hardware which forms complete system. The selection

of this configuration is always done to meet up with the needs of the system as well as that of the organisation. Furthermore, in making the selection, one needs to consider the future change in the organisation in terms of memory, speed and so on. Given the above, the proposed system is designed to run on IBM PC or IBM compatible micro computer with Central processing Unit (CPU) of not less than a 486 DX/4 micro processor. Since a hard disk offers a substantial advantages in data access speed and storage, the computer is expected to have a hard disk of not less than 850 megabytes (MB) storage capacity and a disk drive for 3.5 inches diskettes. The disk drive is expected to be used for data transfer into the Computer, while the large storage capacity of the hard disk is needed against future expansion of the Super Market.

In addition, a printer is also required in the configuration for the production of hard copy reports from the newly designed system. A line printer having an ability of printing 1,200 characters per second is required so as to aid speedy retrieval of information. An uninterrupted power supply (UPS) with an ability for automatic generation of power needs to be included in the configuration for the proposed system. This is to avoid the effect of power failure while a job is being performed.

Similarly, this newly system needs to be operated in an environment with the following facilities.

(i) A computer with at least a 486 DX/4 micro processor.

(ii) 8MB RAM (Random Access Memory)

(iii) 850 MB Hard Disk

(iv) 3.5 inches Diskettes Drive

(v) Dos/windows installed

(vi) A line printer Epson LQ 1050 or LQ 1170

(vii) A UPS of about 800 Kilowatt

4.3 SOFTWARE REQUIREMENT

Software can be defined as programs that direct and control the activities of a computer. It serves as an intermediary between the computer hardware and the Computer use. This accounts for why it is believed that software enables the users to fully exploit the capabilities of a computer. the ability of this newly designed system to work on a computer is due to the requirement and ability of software.

However, the proposed system requires the availability of some forms of software which will enhance the workings of the system and other task that will be required. Specifically the installation of dBASE IV, Lotus 1-2-3 and Word perfect of versions 5.1 or 6.0 and above.

The dBASE IV is to allow for the modification of the proposed system. This is possible because the system was developed using the software. Also required is Lotus 1-2-3 which is a spreadsheet package designed for simple calculation and analysis. The installation of Word perfect is to enable the orgnisation create, modify and print a text of document such as a report, proposal and other official letters. The user of Word perfect for this purpose is to enhance the out-put of the reports and allow for flexibility of the content of the document.

4.4 SOFTWARE TESTING

A system is not assumed to be working perfectly until a confirmed test is performed. This is done by carrying out a system testing. therefore, a system testing can be defined as the process of confirming whether a newly designed system is working in order or not. Because a computer system is expected to assist Computer users in executing the required task with all the necessary speed and accuracy, there is the need to properly test a newly designed system to ensure that it is working according to the set objectives. However, this newly designed accounting system has been tested using data with know result. The known result was obtained after a dry run and this result was compared with the one generated by the new system. These two result turned out to be same.

4.5 SYSTEM CHANGEOVER

Once the system has been fully designed and tested and found to be working perfectly, the next task is the conversion of the old procedures to the newly designed system. This requires a lot of care and precautions so as not to delay the entire workings of the organisation as a whole. It is as a result of this that the introduction of a newly designed system needs a systememattic approach. For the proposed system changeover, which involves full adoption and implementation of the new system. Generally, there are three forms of changeover that can be adopted. they are as follows:-

(i) Direct Changeover

(ii) Pilot Changeover

(iii) Parallel Changeover

The direct changeover involves a situation where the new system is introduced at once and the old system was abandoned immediately. This is said to be dangerous especially if the newly designed system is not well tested. The risk associated to direct changeover could be reduced if pilot changeover is adopted where the new system will be introduced step by step. A new step will not be introduced until the currently introduced one is confirmed working perfectly. This mode of changeover is mostly applicable where the new system is so large. However, the most reliable modes of changeover is the parallel changeover because both the new old system would be considered and compared for sometime. If the new system is then confirmed working satisfactorily, the old system will be discarded.

Specifically, parallel changeover mode is recommended for the implementation of this newly designed system. This is selected so that in case of any problem, the workings of the entire super Market will not be jeopardized.

4.6 SYSTEM DOCUMENTATION

This is the process of describing the workings of the system to the users. It is always required so as to enhance maintainability. The documentation can also be used by the users in getting solution to some problem. However, the documentation of this

system contains how system would be stated as well as describing each of the facilities provided by the new system.

4.6.1 STARTING THE SYSTEM

The new system requires the installation of bases IV on the computer before it can be executed. The dBASE IV needs to the activated by following the steps below:-

(i) Types dBASE at the operating system prompt to display the control center

(ii) Press ESC key to take you to dot prompt

(iii) Given that the suite of programs have been copied to the hard disk, than type Do ACCT to start the execution of the program. This would display the main menu on the screen from which other selections and entries can be carried out.

4.6.2 DESCRIPTION OF THE MENU STRUCTURE

The main menu is designed to have options with each option to be selected by typing any of the letters (A,B,C,D,E, or Q) depending on the choice of interest. The option are stock receipt, daily sales, stock enquiry, code Management, Report Printing and Quit, as represented by Figure 1 in Appendix 1. Each of these options are described as follows:-

(i) STOCK RECEIPT

This option is responsible for updating the system in terms of stock purchase into the Super Market. It has five options as shown in figure 2. The option are stock entry, stock viewing stock modification, stock deletion and quit as represented by figures 3 to 6. The stock entry is to update the file in terms of stock purchases, stock viewing is to display the purchases on the screen, stock modification is the option requires for correction on details of stock purchases entered, while stock deletion is an option used to delete the detail of purchases.

ii. DAILY SALES

This is an option used for recording of sales activities on daily basis. It has five (5) options as display in figure seven (7) namely sales entry, sales listing, sales modification, sales deletion, and quit. The details of each of these five (5) options are represented by figure 8-11.

iii. STOCK ENQUIRY

This menu is used to carry out enquiry on stock position in the supermarket. It has three (3) options as display on figure 12.The option, individual stock position display the stock position of each item of stock as show in figure 13, while the general stock position lists the stock position of all stock items as shown in figure 14.

(iv) CODE MANAGEMENT

This is used to manage stock codes. It has 5 options as displayed ion figure 15. Each of these options are represented by figure 16 to 19.

(v) REPORT PRINTING

This menu list all the reports that can be general on the new system as shown on figure 20. The details of each of the reports are displayed on figures 21 and 22.

(vi) QUIT

This option is used to move out of the system completely.

CHAPTER FIVE

5.1 RECOMMENDATION AND CONCLUSION

The replacement of manual procedures with a computer-based system these days has become a serious affaire. The action necessitated by the benefits associated to computer usage. This accurate calculation to easy retrieve of data and host of others. However, for proper maximization of the benefits of the proposed system, the following actions needs to be adopted.

COMPUTER HARDWARE ACQUISITION

The organisation needs to ensure that a goods computer needs to be purchased that would be free from constant problem. The acquisition of hardware could be done through a qualified computer consultant and the acquisition needs to be done in earnest so as ensure immediate installation of the newly developed system.

PERSONNEL TRAINING

Apart from the need to train two or more staff that would be involved in using the new software, the whole staff needs to be trained as well. The training is expected to be in form of Computer appreciation programme.

COMPUTER ENVIRONMENT

It is also recommended that the security of the computer needs to be ascertained. The security is necessary both for internal and external needs. This is expected to ensure that all the staff do not have access to the use of the computer.

In addition, security is also required to avoid the case of theft. Furthermore, the computer needs to be located in cool environment to avoid constant damage of the hardware.

In conclusion, the pursuance of the installation of this newly designed system needs to be absolute as all the procedures have been tested and confirmed efficient. Therefore, its application will meet both the present and future needs of the supermarket.

REFERENCES

1. **AYO C. K.**, (1994) Computer Literacy Operations and Appreciation ALANUKITAN COMM. Press (Nig.). Ltd. Egbe Nigeria.

2. **AYO C.K.**, (1991) Getting to known the Computer: Published by Newlight Computer, publisher Kaduna.

3. **KILGANON Peter**, (1995) Data processing and system analysis Edward Arnold publishers Ltd. 41, Bedford square London.

4. **MAYNARD J.M.B.C.S** (1980) Computer Programming Made Easy English Language Book Society, London.

5. **OCHU T.T** (Student project F.U.T. Minna) computerisation of inventory in Super market: A case study of Niger Super Market Minna. (1995).

6. **ODUMOSU O.A.** (A Student project F.U.T. Minna): Computerisation of Customers Operations services in Minna digital Exchange NITEL Limited (1993).

7. **ONUNOBI C.T.** (A student project of F.U.T Minna) Computerisation of Application to distributive Trade: A case study of some Supermarket in Minna. (1994)

8. **POPKIN Gany S.** (1983) Advanced Structured COBOL Kent pub. Co. Boston, Massachusetts (USA).

APPENDIX I

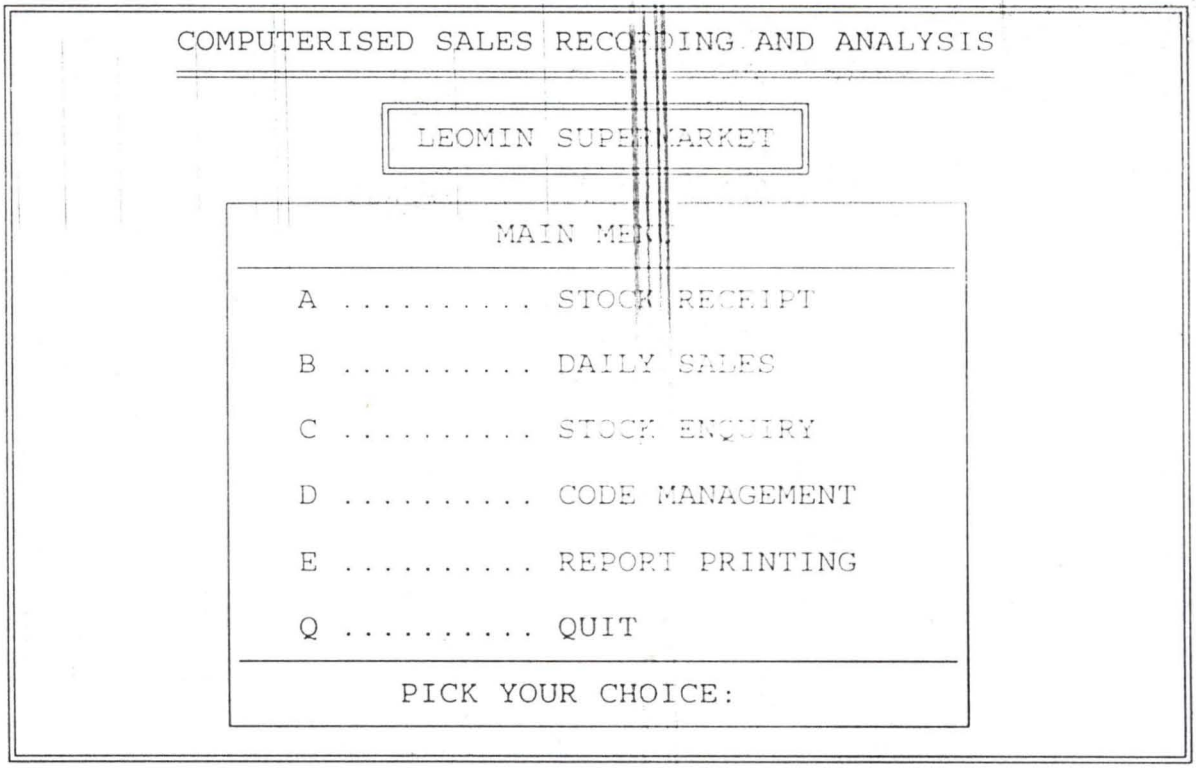

FIGURE 1

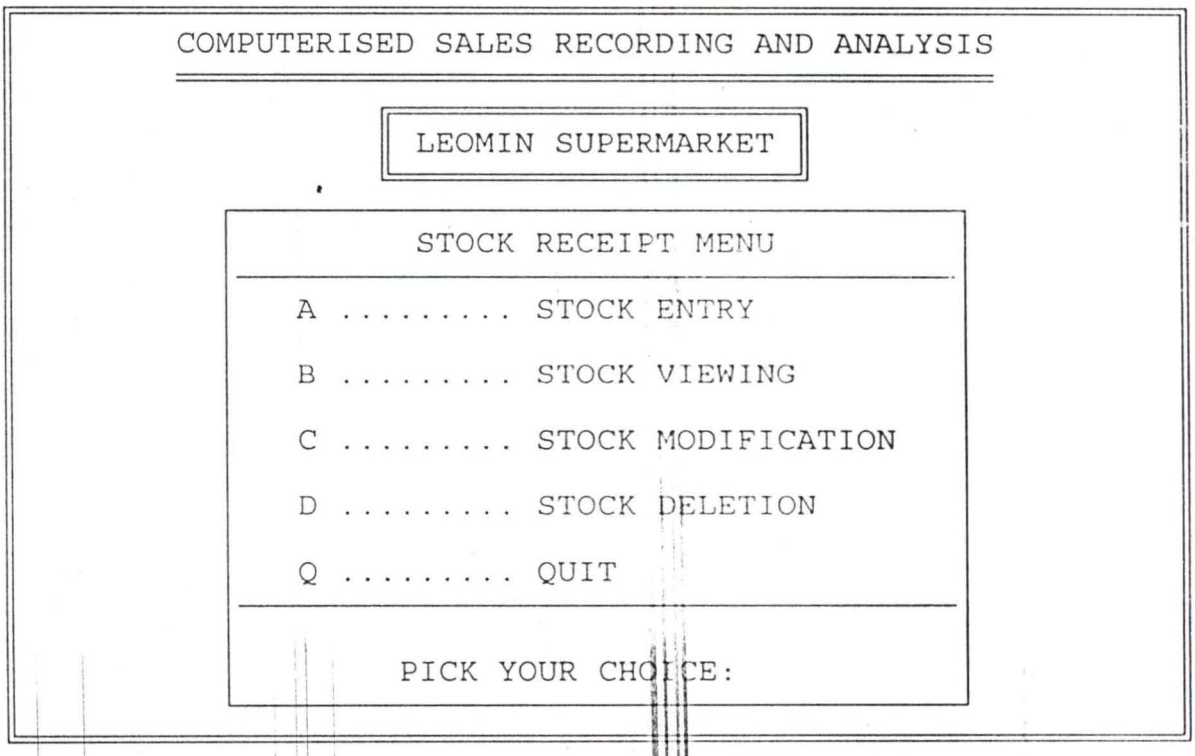

FIGURE 2

STOCK RECEIPT FORM

INVOICE NO: 1NV/034 DATE: 15/10/97

S/NO	ITEM CODE	DESCRIPTION	QUANTITY
1	EL/601	ELECTRIC IRON (PHILIP)	200
2	KT/224	GAS COOKER - 4 BURNERS	120
3	PV/1002	POWDERED MILK (NIDO 250 G)	400
4	PV/2001	CUSTRERED POWDER 350 KG	240

TO UPDATE FILE (Y/N):

FIGURE 3

STOCK VIEWING FORM

OICE NO: 1NV/034 DATE: 15/10/97

O	ITEM CODE	DESCRIPTION	QUANTITY
	EL/601	ELECTRIC IRON (PHILIP)	200
	KT/224	GAS COOKER - 4 BURNERS	120
	PV/1002	POWDERED MILK (NIDO 250 G)	400
	PV/2001	CUSTRERED POWDER 350 KG	240

VIEWING ITEM RECEIVED - PRESS ANY KEY TO EXIT

FIGURE 4

STOCK MODIFICATION FORM

INVOICE NO: 1NV/034 DATE: 15/10/97

S/NO	ITEM CODE	DESCRIPTION	QUANTITY
1	EL/601	ELECTRIC IRON (PHILIP)	200
2	KT/224	GAS COOKER - 4 BURNERS	120
3	PV/1002	POWDERED MILK (NIDO 250 G)	400
4	PV/2001	CUSTRERED POWDER 350 KG	240

TO UPDATE FILE (Y/N):

FIGURE 5

STOCK DELETING FORM

OICE NO: 1NV/034 DATE: 15/10/97

O	ITEM CODE	DESCRIPTION	QUANTITY
	EL/601	ELECTRIC IRON (PHILIP)	200
	KT/224	GAS COOKER - 4 BURNERS	120
	PV/1002	POWDERED MILK (NIDO 250 G)	400
	PV/2001	CUSTRERED POWDER 350 KG	240

TO DELETE RECORDS (Y/N):

FIGURE 6

COMPUTERISED SALES RECORDING AND ANALYSIS

LEOMIN SUPERMARKET

DAILY SALES MENU

A ..... SALES ENTRY

B ..... SALES LISTING

C ..... SALES MODIFICATION

D ..... SALES DELETION

Q ..... QUIT

PICK YOUR CHOICE:

FIGURE 7

SALES ENTRY FORM

L SALES: # 26,250.00 SALES NO: 1 DATE: 19/10/9

IT.CODE	DESCRIPTION	PRICE	QTY	SALES AMOU
EL/601	ELECTRIC IRON (PHILIP)	1,850.00	2	3,700.
KT/224	GAS COOKER - 4 BURNERS	19,800.00	1	19,800.
PV/1002	POWDERED MILK (NIDO 250 G)	275.00	10	2,750.

TO ENTER MORE (Y/N):

FIGURE 8

SALES LISTING FORM

DATE: 19/10/9

SALES NO	DESCRIPTION	PRICE	QTY	SALES AMOUNT
1	ELECTRIC IRON (PHILIP)	1,850.00	2	3,700.0
1	GAS COOKER - 4 BURNERS	19,800.00	1	19,800.0
1	POWDERED MILK (NIDO 250 G)	275.00	10	2,750.0
2	CUSTRERED POWDER 350 KG	495.00	3	1,485.0
2	GAS COOKER - 4 BURNERS	19,800.00	1	19,800.0

IT IS END OF FILE - PRESS ANY KEY TO EXIT

FIGURE 9

SALES MODIFICATION FORM

AL SALES: # 26,250.00 SALES NO: 1 DATE: 19/10/

IT.CODE	DESCRIPTION	PRICE	QTY	SALES AMO
EL/601	ELECTRIC IRON (PHILIP)	1,850.00	2	3,700
KT/224	GAS COOKER - 4 BURNERS	19,800.00	1	19,800
PV/1002	POWDERED MILK (NIDO 250 G)	275.00	10	2,750

TO UPDATE FILE (Y/N):

FIGURE 10

SALES DELETION FORM

AL SALES: # 26,250.00 SALES NO: 1 DATE: 19/10/9

IT.CODE	DESCRIPTION	PRICE	QTY	SALES AMOU
EL/601	ELECTRIC IRON (PHILIP)	1,850.00	2	3,700.
KT/224	GAS COOKER - 4 BURNERS	19,800.00	1	19,800.
PV/1002	POWDERED MILK (NIDO 250 G)	275.00	10	2,750.

TO DELETE RECORDS (Y/N):

FIGURE 11

COMPUTERISED SALES RECORDING AND ANALYSIS

LEOMIN SUPERMARKET

ENQUIRY MENU

A ---------- INDIVIDUAL STOCK POSITION

B ---------- GENERAL STOCK POSITION

Q ---------- QUIT

PICK YOUR CHOICE:

FIGURE 12

INDIVIDUAL STOCK POSITION

ITEM CODE (OR "↵" KEY TO EXIT): PV/2001

DESCRIPTION: CUSTRERED POWDER 350 KG

BALANCE OF STOCK: 387

VIEWING STOCK BALANCE - PRESS ANY KEY

FIGURE 13

GENERAL STOCK POSITION

S/NO	ITEM CODE	DESCRIPTION	QUANTITY
1	EL/601	ELECTRIC IRON (PHILIP)	298
2	KT/224	GAS COOKER - 4 BURNERS	163
3	PV/1002	POWDERED MILK (NIDO 250 G)	740
4	PV/2001	CUSTRERED POWDER 350 KG	387

IT IS END OF FILE - PRESS ANY KEY TO EXIT

FIGURE 14

COMPUTERISED SALES RECORDING AND ANALYSIS

LEOMIN SUPERMARKET

ITEM CODE UPDATE MENU

A ---------- NEW CODE ENTRY

B ---------- VIEW CODE ENTRY

C ---------- CHANGE CODE ENTRY

D ---------- DELETE CODE ENTRY

Q ---------- QUIT

PICK YOUR CHOICE:

FIGURE 15

LEOMIN SUPERMARKET

NEW CODE ENTRY FORM

ITEM CODE: (OR "↵" KEY TO EXIT): PV/1011

DESCRIPTION OF ITEM: BOURNVITA GIANT SIZE (500 KG)

COST PRICE OF ITEM: 350.00

SELLING PRICE OF ITEM: 385.00

TO UPDATE FILE (Y/N):

FIGURE 16

LEOMIN SUPERMARKET

VIEW CODE ENTRY FORM

ITEM CODE: (OR "↵" KEY TO EXIT): EL/601

DESCRIPTION OF ITEM: ELECTRIC IRON (PHILIP)

COST PRICE OF ITEM: 1,500.00

SELLING PRICE OF ITEM: 1,850.00

VIEWING ITEM CODE - PRESS ANY KEY

FIGURE 17

LEOMIN SUPERMARKET

CHANGE CODE ENTRY FORM

ITEM CODE: (OR "↵" KEY TO EXIT): PV/1002

DESCRIPTION OF ITEM: POWDERED MILK (NIDO 250 G)

COST PRICE OF ITEM: 250.00

SELLING PRICE OF ITEM: 275.00

TO UPDATE FILE (Y/N):

FIGURE 18

LEOMIN SUPERMARKET

DELETE CODE ENTRY FORM

ITEM CODE: (OR "↵" KEY TO EXIT): PV/1011

DESCRIPTION OF ITEM: BOURNVITA GIANT SIZE (500 KG)

COST PRICE OF ITEM: 350.00

SELLING PRICE OF ITEM: 385.00

TO DELETE ITEM CODE (Y/N):

FIGURE 19

COMPUTERISED SALES RECORDING AND ANALYSIS

LEOMIN SUPERMARKET

REPORT GENERATION MENU

A ---------- DAILY SALES ANALYSIS

B ---------- WEEKLY SALES ANALYSIS

Q ---------- QUIT

PICK YOUR CHOICE:

FIGURE 20

LEOMIN SUPERMARKET

DAILY SALES REPORT FOR 19/10/97

D	DESCRIPTION	PRICE	QTY	SALES AMOUNT
1	ELECTRIC IRON (PHILIP)	1,850.00	2	3,700.00
2	GAS COOKER - 4 BURNERS	19,800.00	2	39,600.00
3	POWDERED MILK (NIDO 250 G)	275.00	10	2,750.00
4	CUSTRERED POWDER 350 KG	495.00	3	1,485.00
	G R A N D - T O T A L : -			47,535.00

FIGURE 21

LEOMIN SUPERMARKET

WEEKLY SALES REPORT AS AT 19/10/97

/NO	DESCRIPTION	PRICE	QTY	SALES AMOUNT
1	ELECTRIC IRON (PHILIP)	1,850.00	3	5,550.00
2	GAS COOKER - 4 BURNERS	19,800.00	2	39,600.00
3	POWDERED MILK (NIDO 250 G)	275.00	20	5,500.00
4	BOURNVITA GIANT SIZE (500 KG)	385.00	8	3,080.00
5	CUSTRERED POWDER 350 KG	495.00	5	2,475.00
	G R A N D - T O T A L : -			56,205.00

FIGURE 22

APPENDIX II

```
set talk off
set date brit
set stat off
set scor off
set safe off
set dele on
do whil .t.
 clea
 @ 0,10 to 23,69 doub
 @ 1,19 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
 @ 2,19 to 2,59 doub
 @ 3,29 to 5,50 doub
 @ 4,31 say 'LEOMIN SUPERMARKET'
 @ 8,22 to 8,57
 @ 7,35 say 'MAIN MENU'
 @ 6,21 to 22,58
 @ 20,22 to 20,57
 @ 9,25 say 'A ..... STOCK RECEIPT'
 @ 11,25 say 'B ..... DAILY SALES'
 @ 13,25 say 'C ..... STOCK ENQUIRY'
 @ 15,25 say 'D ..... CODE MANAGEMENT'
 @ 17,25 say 'E ..... REPORT PRINTING'
 @ 19,25 say 'Q ..... QUIT'
 @ 21,30 say 'PICK YOUR CHOICE:'
 do whil .t.
  resp = ' '
  @ 21,48 get resp pict '!'
  read
```

```
    if resp $ 'ABCDEQ'
      exit
    endi
  endd
  do case
    case resp='A'
      do stockr
    case resp='B'
      do sales
    case resp='C'
      do enq
    case resp='D'
      do code
    case resp='E'
      do rep
    othe
      exit
  endc
endd
clea
retu


STOCKR.PRG
do whil .t.
  clea
  @ 1,10 to 23,69 doub
  @ 2,19 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
  @ 3,19 to 3,59 doub
```

```
@ 4,29 to 6,50 doub
@ 5,31 say 'LEOMIN SUPERMARKET'
@ 9,22 to 9,57
@ 8,31 say 'STOCK RECEIPT MENU'
@ 7,21 to 22,58
@ 19,22 to 19,57
@ 10,25 say 'A ..... STOCK ENTRY'
@ 12,25 say 'B ..... STOCK VIEWING'
@ 14,25 say 'C ..... STOCK MODIFICATION'
@ 16,25 say 'D ..... STOCK DELETION'
@ 18,25 say 'Q ..... QUIT'
@ 21,30 say 'PICK YOUR CHOICE:'
do whil .t.
  resp = ' '
  @ 21,48 get resp pict '!'
  read
  if resp $ 'ABCDQ'
    exit
  endi
endd
do case
  case resp = 'A'
    do nitem
  case resp = 'B'
    do vitem
  case resp = 'C'
    do mitem
  case resp = 'D'
    do ditem
```

```
    othe
      exit
  endc
endd
clea
retu


NITEM.PRG
use tdelv
if .not. eof()
  zap
endi
*sele a
*  use delivery
*sele b
*  use code
*sele c
*  use tdelv
do whil .t.
  clea
  minvno = spac(7)
  mdate = ctod('  /  /  ')
  @ 9,14 to 11,65
  @ 10,16 say 'ENTER INVOICE NO (OR "'
  @ 10,38 say chr(27) + chr(196) + chr(217)  + '" KEY TO EXIT):'
  @ 10,57 get minvno pict '@!'
  read
  if minvno = spac(7)
```

```
  exit
 endi
 use delivery
* go top
 loca for invno=minvno
 if found()
  @ 17,18 to 19,61 doub
  @ 18,20 say 'INVOICE ALREADY ENTERED - Press any key '
  set cons off
  wait
  set cons on
  @ 17,18 clea to 19,61
  loop
 endi
 @ 13,19 to 15,59
 @ 14,21 say 'ENTER DATE OF STOCK RECEIPT:' get mdate
 read
 sno=0
 clea
 @ 0,1 to 24,78 doub
 @ 1,29 to 3,50 doub
 @ 2,31 say 'STOCK RECEIPT FORM'
 @ 3,3 say 'INVOICE NO:' get minvno
 @ 3,63 say 'DATE:' get mdate
 clea gets
 @ 4,2 to 4,77
 @ 5,3 say 'S/NO'
 @ 5,10 say 'ITEM CODE'
 @ 5,31 say 'DESCRIPTION'
```

```
@ 5,69 say 'QUANTITY'
@ 5,8 to 21,8
@ 5,20 to 21,20
@ 5,67 to 21,67
* @ 5,64 to 21,64
@ 6,2 to 6,7
@ 6,9 to 6,19
@ 6,21 to 6,66
@ 6,68 to 6,77
@ 22,2 to 22,77
r=7
do whil .t.
  sno=sno+1
  mqty=0
  mexdate=ctod('  /  /  ')
  @ r,4 say sno pict '99'
  do whil .t.
    mcode=spac(7)
    @ r,11 get mcode
    read
    use code
*    go top
    loca for mcode=code
    if .not. found()
      @ 23,23 say 'INVALID ITEM CODE - PRESS ANY KEY'
      set cons off
      wait
      set cons on
      @ 23,23 say spac(34)
```

```
    loop
  endi
  exit
endd
mdesc = desc
@ r,22 get mdesc
clea gets
@ r,70 get mqty pict '999999'
read
use tdelv
appe blan
repl invno with minvno,code with mcode,qty with mqty
repl date with mdate,exdate with mexdate
@ 23,29 say 'TO ENTER MORE (Y/N):'
do whil .t.
  resp = ' '
  @ 23,50 get resp pict '!'
  read
  if resp $ 'YN'
    exit
  endi
endd
@ 23,29 clea to 23,50
if resp = 'N'
  exit
endi
r = r + 2
if r > 21
  @ 7,2 clea to 21,7
```

```
    @ 7,9 clea to 21,19
    @ 7,21 clea to 21,66
    @ 7,68 clea to 21,77
    r=7
  endi
endd
if resp='N'
  @ 23,28 say 'TO UPDATE FILE (Y/N):'
  do whil .t.
    resp=' '
    @ 23,50 get resp pict '!'
    read
    if resp $ 'YN'
      exit
    endi
  endd
endi
if resp='Y'
  use delivery
  appe from tdelv
  use
  sele a
    use tdelv
  sele b
    use code
  sele a
  do whil .not. eof()
    mcode=code
    mqty1=qty
```

```
      sele b
      go top
      loca for mcode=code
      mqty2=qty
      mqty=mqty1+mqty2
      repl qty with mqty
      sele a
      skip
    endd
  endi
  clos all
  use tdelv
  zap
endd
clos all
clea
retu


VITEM.PRG
use tdelv
if .not. eof()
  zap
endi
*sele a
*  use delivery
*sele b
*  use code
*sele c
```

```
* use tdelv
do whil .t.
 clea
 minvno = spac(7)
 @ 9,14 to 11,65
 @ 10,16 say 'ENTER INVOICE NO (OR "'
 @ 10,38 say chr(27) + chr(196) + chr(217) +'" KEY TO EXIT):'
 @ 10,57 get minvno pict '@!'
 read
 if minvno = spac(7)
  exit
 endi
 use delivery
* go top
 loca for invno = minvno
 if .not. found()
  @ 17,20 to 19,58 doub
  @ 18,22 say 'INVALID INVOICE NO - Press any key '
  set cons off
  wait
  set cons on
  @ 17,18 clea to 19,61
  loop
 endi
 mdate = date
 use tdelv
 appe from delivery.dbf for invno = minvno
 sno = 0
 clea
```

```
@ 0,1 to 24,78 doub
@ 1,29 to 3,50 doub
@ 2,31 say 'STOCK VIEWING FORM'
@ 3,3 say 'INVOICE NO:' get minvno
@ 3,63 say 'DATE:' get mdate
clea gets
@ 4,2 to 4,77
@ 5,3 say 'S/NO'
@ 5,10 say 'ITEM CODE'
@ 5,31 say 'DESCRIPTION'
@ 5,69 say 'QUANTITY'
@ 5,8 to 21,8
@ 5,20 to 21,20
@ 5,67 to 21,67
@ 6,2 to 6,7
@ 6,9 to 6,19
@ 6,21 to 6,66
@ 6,68 to 6,77
@ 22,2 to 22,77
r=7
sele a
 use tdelv
sele b
 use code
sele a
go top
do whil .not. eof()
 sno=sno+1
 mqty=qty
```

```
mexdate = exdate
mcode = code
sele b
go top
loca for mcode = code
mdesc = desc
@ r,4 say sno pict '99'
@ r,11 get mcode
@ r,22 get mdesc
@ r,70 get mqty pict '999999'
clea gets
sele a
skip
if eof()
  @ 23,17 say 'VIEWING ITEM RECEIVED - PRESS ANY KEY TO EXIT'
  set cons off
  wait
  set cons on
  exit
endi
r = r + 2
if r > 21
  @ 23,15 say 'VIEWING ITEM RECEIVED - PRESS ANY KEY TO CONTINUE'
  set cons off
  wait
  set cons on
  @ 7,2 clea to 21,7
  @ 7,9 clea to 21,19
  @ 7,21 clea to 21,66
```

```
      @ 7,68 clea to 21,77
      @ 23,15 clea to 23,64
      r=7
    endi
  endd
  sele a
  zap
  clos all
endd
clos all
clea
retu


MITEM.PRG
use tdelv
if .not. eof()
  zap
endi
*sele a
*  use delivery
*sele b
*  use code
*sele c
*  use tdelv
do whil .t.
  clea
  minvno=spac(7)
  @ 9,14 to 11,65
```

```
@ 10,16 say 'ENTER INVOICE NO (OR "'
@ 10,38 say chr(27)+chr(196)+chr(217) +'" KEY TO EXIT):'
@ 10,57 get minvno pict '@!'
read
if minvno=spac(7)
  exit
endi
use delivery
* go top
loca for invno=minvno
if .not. found()
  @ 17,20 to 19,58 doub
  @ 18,22 say 'INVALID INVOICE NO - Press any key '
  set cons off
  wait
  set cons on
  @ 17,18 clea to 19,61
  loop
endi
mdate=date
use tdelv
appe from delivery.dbf for invno=minvno
sno=0
clea
@ 0,1 to 24,78 doub
@ 1,27 to 3,53 doub
@ 2,29 say 'STOCK MODIFICATION FORM'
@ 3,3 say 'INVOICE NO:' get minvno
@ 3,63 say 'DATE:' get mdate
```

```
clea gets
@ 4,2 to 4,77
@ 5,3 say 'S/NO'
@ 5,10 say 'ITEM CODE'
@ 5,31 say 'DESCRIPTION'
@ 5,69 say 'QUANTITY'
@ 5,8 to 21,8
@ 5,20 to 21,20
@ 5,67 to 21,67
@ 6,2 to 6,7
@ 6,9 to 6,19
@ 6,21 to 6,66
@ 6,68 to 6,77
@ 22,2 to 22,77
r=7
sele a
 use tdelv
sele b
 use code
sele a
go top
do whil .not. eof()
 sno=sno+1
 mqty=qty
 mexdate=exdate
 mcode=code
 sele b
 go top
 loca for mcode=code
```

```
      mdesc = desc
      @ r,4 say sno pict '99'
      @ r,11 get mcode
      @ r,22 get mdesc
      @ r,70 get mqty pict '999999'
      clea gets
      sele a
      skip
      r = r + 2
      if eof()
        @ 23,28 say 'TO UPDATE FILE (Y/N):'
        do whil .t.
          resp = ' '
          @ 23,50 get resp pict '!'
          read
          if resp $ 'YN'
            exit
          endi
        endd
      endi
    endd
    sele a
    zap
    clos all
endd
clos all
clea
retu
```

```
DITEM.PRG
use tdelv
if .not. eof()
  zap
endi
do whil .t.
  clea
  minvno = spac(7)
  @ 9,14 to 11,65
  @ 10,16 say 'ENTER INVOICE NO (OR "'
  @ 10,38 say chr(27) + chr(196) + chr(217)  + '" KEY TO EXIT):'
  @ 10,57 get minvno pict '@!'
  read
  if minvno = spac(7)
    exit
  endi
  use delivery
  loca for invno = minvno
  if .not. found()
    @ 17,20 to 19,58 doub
    @ 18,22 say 'INVALID INVOICE NO - Press any key '
    set cons off
    wait
    set cons on
    @ 17,18 clea to 19,61
    loop
  endi
  mdate = date
  use tdelv
```

```
appe from delivery.dbf for invno = minvno
sno = 0
clea
@ 0,1 to 24,78 doub
@ 1,29 to 3,51 doub
@ 2,31 say 'STOCK DELETING FORM'
@ 3,3 say 'INVOICE NO:' get minvno
@ 3,63 say 'DATE:' get mdate
clea gets
@ 4,2 to 4,77
@ 5,3 say 'S/NO'
@ 5,10 say 'ITEM CODE'
@ 5,31 say 'DESCRIPTION'
@ 5,69 say 'QUANTITY'
@ 5,8 to 21,8
@ 5,20 to 21,20
@ 5,67 to 21,67
@ 6,2 to 6,7
@ 6,9 to 6,19
@ 6,21 to 6,66
@ 6,68 to 6,77
@ 22,2 to 22,77
r = 7
sele a
  use tdelv
sele b
  use code
sele c
  use delivery
```

```
sele a
go top
do whil .not. eof()
  sno = sno + 1
  mqty = qty
  mexdate = exdate
  mcode = code
  sele b
  go top
  loca for code = mcode
  mdesc = desc
  @ r,4 say sno pict '99'
  @ r,11 get mcode
  @ r,22 get mdesc
  @ r,70 get mqty pict '999999'
  clea gets
  sele a
  skip
  if eof()
    @ 23,27 say 'TO DELETE RECORDS (Y/N):'
    do whil .t.
      resp = ' '
      @ 23,52 get resp pict '!'
      read
      if resp $ 'YN'
        exit
      endi
    endd
    exit
```

```
  endi
  r=r+2
  if r>21
    @ 23,27 say 'PRESS ANY KEY TO CONTINUE'
    set cons off
    wait
    set cons on
    @ 7,2 clea to 21,7
    @ 7,9 clea to 21,19
    @ 7,21 clea to 21,66
    @ 7,68 clea to 21,77
    @ 23,15 clea to 23,64
    r=7
  endi
endd
if resp='Y'
  sele a
  go top
  do whil .not. eof()
    mcode=code
    mqty1=qty
    sele b
    go top
    loca for mcode=code
    mqty2=qty
    mqty=mqty2-mqty1
    repl qty with mqty
    sele c
    go top
```

```
    loca for minvno=invno .and. mcode=code
    dele
    sele a
    skip
   endd
   sele c
   pack
   @ 23,22 say 'RECORDS ARE DELETED - PRESS ANY KEY'
  else
   @ 23,20 say 'RECORDS ARE NOT DELETED - PRESS ANY KEY'
  endi
  set cons off
  wait
  set cons on
  sele a
  zap
  clos all
endd
clos all
clea
retu


SALES.PRG
do whil .t.
  clea
  @ 1,10 to 23,69 doub
  @ 2,19 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
  @ 3,19 to 3,59 doub
```

```
@ 4,29 to 6,50 doub
@ 5,31 say 'LEOMIN SUPERMARKET'
@ 9,22 to 9,57
@ 8,32 say 'DAILY SALES MENU'
@ 7,21 to 22,58
@ 19,22 to 19,57
@ 10,25 say 'A ..... SALES ENTRY'
@ 12,25 say 'B ..... SALES LISTING'
@ 14,25 say 'C ..... SALES MODIFICATION'
@ 16,25 say 'D ..... SALES DELETION'
@ 18,25 say 'Q ..... QUIT'
@ 21,30 say 'PICK YOUR CHOICE:'
do whil .t.
  resp = ' '
  @ 21,48 get resp pict '!'
  read
  if resp $ 'ABCDQ'
    exit
  endi
endd
do case
  case resp = 'A'
    do iteme
  case resp = 'B'
    do iteml
  case resp = 'C'
    do itemm
  case resp = 'D'
    do itemd
```

```
    othe
      exit
  endc
endd
clea
retu


ITEME.PRG
today = date()
use dsales
if .not. eof()
  mdate = date
  if today < > mdate
    use tsales
    appe from dsales.dbf
    use dsales
    zap
    msalesno = 0
  else
    go bott
    msalesno = salesno
  endi
else
  msalesno = 0
endi
use sales
if .not. eof()
  zap
```

```
endi
do whil .t.
 clea
 @ 9,20 to 11,59
 @ 10,23 say "TO ENTER CUSTOMER'S SALES (Y/N):"
 do whil .t.
  resp = ' '
  @ 10,56 get resp pict '!'
  read
  if resp = 'Y' .or. resp = 'N'
    exit
  endi
 endd
 if resp = 'N'
  exit
 endi
 clea
 mtsales = 0
 msalesno = msalesno + 1
 @ 0,29 to 2,50 doub
 @ 1,32 say 'SALES ENTRY FORM'
 @ 3,1 say 'TOTAL SALES: #'
 @ 3,15 get mtsales pict '9,999,999,999.99'
 @ 3,38 say 'SALES NO:' get msalesno
 @ 3,65 say 'DATE:' get today
 clea gets
 @ 4,0 to 4,79
 @ 5,0 say 'S/NO'
 @ 5,6 say 'IT.CODE'
```

```
@ 5,21 say 'DESCRIPTION'
@ 5,49 say 'PRICE'
@ 5,61 say 'QTY'
@ 5,68 say 'SALES AMOUNT'
@ 5,4 to 21,4
@ 5,14 to 21,14
@ 5,45 to 21,45
@ 5,58 to 21,58
@ 5,66 to 21,66
@ 6,0 to 6,3
@ 6,5 to 6,13
@ 6,15 to 6,44
@ 6,46 to 6,57
@ 6,59 to 6,65
@ 6,67 to 6,79
@ 22,0 to 22,79
@ 24,0 to 24,79
sno = 0
r = 7
do whil .t.
  sno = sno + 1
  mqty = 0
  @ r,1 say sno pict '99'
  do whil .t.
    mcode = spac(7)
    @ r,6 get mcode
    read
    use code
*    go top
```

```
 loca for mcode=code
 if .not. found()
   @ 23,23 say 'INVALID ITEM CODE - PRESS ANY KEY'
   set cons off
   wait
   set cons on
   @ 23,23 say spac(34)
   loop
 endi
 exit
endd
mdesc=desc
msp=sp
@ r,15 get mdesc
@ r,47 get msp pict '999,999.99'
clea gets
@ r,60 get mqty pict '99999'
read
mamt=mqty*msp
mtsales=mtsales+mamt
@ r,67 get mamt pict '99,999,999.99'
@ 3,15 get mtsales pict '9,999,999,999.99'
clea gets
use sales
appe blan
repl date with today,code with mcode,qty with mqty
repl amount with mamt,desc with mdesc,price with msp
repl salesno with msalesno
@ 23,29 say 'TO ENTER MORE (Y/N):'
```

```
    do whil .t.
      resp = ' '
      @ 23,50 get resp pict '!'
      read
      if resp $ 'YN'
        exit
      endi
    endd
    @ 23,29 clea to 23,50
    if resp = 'N'
      exit
    endi
    r = r + 2
    if r > 21
      @ 7,2 clea to 21,7
      @ 7,9 clea to 21,19
      @ 7,21 clea to 21,52
      @ 7,54 clea to 21,63
      @ 7,65 clea to 21,77
      r = 7
    endi
  endd
  if resp = 'N'
    @ 23,28 say 'TO UPDATE FILE (Y/N):'
    do whil .t.
      resp = ' '
      @ 23,50 get resp pict '!'
      read
      if resp $ 'YN'
```

```
      exit
    endi
  endd
endi
if resp='Y'
  use dsales
  appe from sales
  use
  sele a
    use sales
  sele b
    use code
  sele a
  do whil .not. eof()
    mcode=code
    mqty1=qty
    sele b
    go top
    loca for mcode=code
    mqty2=qty
    mqty=mqty2-mqty1
    repl qty with mqty
    sele a
    skip
  endd
endi
clos all
use sales
zap
```

```
endd
clos all
clea
retu


ITEML.PRG
use dsales
mdate = date
clea
@ 0,0 to 24,79 doub
@ 1,29 to 3,50 doub
@ 2,31 say 'SALES LISTING FORM'
* @ 3,3 say 'INVOICE NO:' get minvno
@ 3,64 say 'DATE:' get mdate
clea gets
@ 4,1 to 4,78
@ 5,2 say 'SALES NO'
@ 5,19 say 'DESCRIPTION'
@ 5,46 say 'PRICE'
@ 5,58 say 'QTY'
@ 5,65 say 'SALES AMOUNT'
@ 5,11 to 21,11
@ 5,42 to 21,42
@ 5,55 to 21,55
@ 5,63 to 21,63
@ 6,1 to 6,10
@ 6,12 to 6,41
@ 6,43 to 6,54
```

```
@ 6,56 to 6,62
@ 6,64 to 6,78
@ 22,1 to 22,78
r=7
do whil .not. eof()
  msalesno=salesno
  mqty=qty
  mdesc=desc
  mprice=price
  mamt=amount
  @ r,3 get msalesno pict '999999'
  @ r,12 get mdesc
  @ r,44 get mprice pict '999,999.99'
  @ r,57 get mqty pict '99999'
  @ r,65 get mamt pict '99,999,999.99'
  clea gets
  skip
  if eof()
    @ 23,19 say 'IT IS END OF FILE - PRESS ANY KEY TO EXIT'
    set cons off
    wait
    set cons on
    exit
  endi
  r=r+2
  if r>21
    @ 23,16 say 'LISTING SALES ENTRY - PRESS ANY KEY TO CONTINUE'
    set cons off
    wait
```

```
    set cons on
    @ 7,1 clea to 21,10
    @ 7,12 clea to 21,41
    @ 7,43 clea to 21,54
    @ 7,56 clea to 21,62
    @ 7,64 clea to 21,78
    @ 23,15 clea to 23,64
    r=7
  endi
endd
clos all
clea
retu


ITEMM.PRG
use sales
if .not. eof()
  zap
endi
do whil .t.
  clea
  msalesno=0
  @ 9,15 to 11,64
  @ 10,17 say 'ENTER SALES NO (OR "'
  @ 10,37 say chr(27)+chr(196)+chr(217) +'" KEY TO EXIT):'
  @ 10,56 get msalesno pict '999999'
  read
  if msalesno=0
```

```
  exit
endi
use dsales
loca for salesno = msalesno
if .not. found()
  @ 17,21 to 19,57 doub
  @ 18,23 say 'INVALID SALES NO - Press any key '
  set cons off
  wait
  set cons on
  @ 17,18 clea to 19,61
  loop
endi
mdate = date
use sales
appe from dsales.dbf for salesno = msalesno
sum amount to mtsales
go top
sno = 0
clea
@ 0,25 to 2,53 doub
@ 1,28 say 'SALES MODIFICATION FORM'
@ 3,1 say 'TOTAL SALES: #'
@ 3,15 get mtsales pict '9,999,999,999.99'
@ 3,38 say 'SALES NO:' get msalesno
@ 3,65 say 'DATE:' get mdate
clea gets
@ 4,0 to 4,79
@ 5,0 say 'S/NO'
```

```
@ 5,6 say 'IT.CODE'
@ 5,21 say 'DESCRIPTION'
@ 5,49 say 'PRICE'
@ 5,61 say 'QTY'
@ 5,68 say 'SALES AMOUNT'
@ 5,4 to 21,4
@ 5,14 to 21,14
@ 5,45 to 21,45
@ 5,58 to 21,58
@ 5,66 to 21,66
@ 6,0 to 6,3
@ 6,5 to 6,13
@ 6,15 to 6,44
@ 6,46 to 6,57
@ 6,59 to 6,65
@ 6,67 to 6,79
@ 22,0 to 22,79
@ 24,0 to 24,79
r = 7
sno = 0
do whil .not. eof()
  sno = sno + 1
  mqty = qty
  mdesc = desc
  mcode = code
  mprice = price
  mamount = amount
  @ r,1 say sno pict '99'
  @ r,6 get mcode
```

```
  @ r,15 get mdesc
  @ r,47 get mprice pict '999,999.99'
  @ r,60 get mqty pict '99999'
  @ r,67 get mamount pict '99,999,999.99'
  clea gets
  skip
  r=r+2
endd
sele a
  use sales
sele b
  use code
sele a
sno=0
r=7
do whil .not. eof()
  sno=sno+1
  mqty=qty
  mcode=code
  mamt=amount
  mamount=amount
  @ r,1 say sno pict '99'
  sele b
  do whil .t.
    @ r,6 get mcode
    read
    go top
    loca for mcode=code
    if .not. found()
```

```
      @ 23,23 say 'INVALID ITEM CODE - PRESS ANY KEY'
      set cons off
      wait
      set cons on
      @ 23,23 say spac(34)
      loop
    endi
    exit
  endd
  mdesc = desc
  msp = sp
  sele a
  @ r,15 get mdesc
  @ r,47 get msp pict '999,999.99'
  clea gets
  @ r,60 get mqty pict '99999'
  read
  mamt = mqty * msp
  mtsales = mtsales + mamt-mamount
  @ r,67 get mamt pict '99,999,999.99'
  @ 3,15 get mtsales pict '9,999,999,999.99'
  clea gets
  repl code with mcode,qty with mqty
  repl amount with mamt,desc with mdesc,price with msp
  skip
  r = r + 2
endd
clos all
@ 23,28 say 'TO UPDATE FILE (Y/N):'
```

```
    do whil .t.
      resp = ' '
      @ 23,50 get resp pict '!'
      read
      if resp $ 'YN'
        exit
      endi
    endd
    if resp = 'Y'
      use dsales
      dele all for salesno = msalesno
      pack
      appe from sales
    endi
    use sales
    zap
    use
  endd
  clos all
  clea
  retu


ITEMD.PRG
use sales
if .not. eof()
  zap
endi
do whil .t.
```

```
clea
msalesno = 0
@ 9,15 to 11,64
@ 10,17 say 'ENTER SALES NO (OR "'
@ 10,37 say chr(27) + chr(196) + chr(217) + '" KEY TO EXIT):'
@ 10,56 get msalesno pict '999999'
read
if msalesno = 0
  exit
endi
use dsales
loca for salesno = msalesno
if .not. found()
  @ 17,21 to 19,57 doub
  @ 18,23 say 'INVALID SALES NO - Press any key '
  set cons off
  wait
  set cons on
  @ 17,18 clea to 19,61
  loop
endi
mdate = date
use sales
appe from dsales.dbf for salesno = msalesno
sum amount to mtsales
go top
sno = 0
clea
@ 0,27 to 2,51 doub
```

```
@ 1,30 say 'SALES DELETION FORM'
@ 3,1 say 'TOTAL SALES: #'
@ 3,15 get mtsales pict '9,999,999,999.99'
@ 3,38 say 'SALES NO:' get msalesno
@ 3,65 say 'DATE:' get mdate
clea gets
@ 4,0 to 4,79
@ 5,0 say 'S/NO'
@ 5,6 say 'IT.CODE'
@ 5,21 say 'DESCRIPTION'
@ 5,49 say 'PRICE'
@ 5,61 say 'QTY'
@ 5,68 say 'SALES AMOUNT'
@ 5,4 to 21,4
@ 5,14 to 21,14
@ 5,45 to 21,45
@ 5,58 to 21,58
@ 5,66 to 21,66
@ 6,0 to 6,3
@ 6,5 to 6,13
@ 6,15 to 6,44
@ 6,46 to 6,57
@ 6,59 to 6,65
@ 6,67 to 6,79
@ 22,0 to 22,79
@ 24,0 to 24,79
r=7
sno=0
do whil .not. eof()
```

```
  sno = sno + 1
  mqty = qty
  mdesc = desc
  mcode = code
  mprice = price
  mamount = amount
  @ r,1 say sno pict '99'
  @ r,6 get mcode
  @ r,15 get mdesc
  @ r,47 get mprice pict '999,999.99'
  @ r,60 get mqty pict '99999'
  @ r,67 get mamount pict '99,999,999.99'
  clea gets
  skip
  r = r + 2
endd
@ 23,27 say 'TO DELETE RECORDS (Y/N):'
do whil .t.
  resp = ' '
  @ 23,52 get resp pict '!'
  read
  if resp $ 'YN'
    exit
  endi
endd
if resp = 'Y'
  use dsales
  dele all for salesno = msalesno
  pack
```

```
   @ 23,22 say 'RECORDS ARE DELETED - PRESS ANY KEY'
  else
   @ 23,20 say 'RECORDS ARE NOT DELETED - PRESS ANY KEY'
  endi
  set cons off
  wait
  set cons on
  use sales
  zap
endd
clos all
clea
retu


ENQ.PRG
do whil .t.
  clea
  @ 3,10 to 21,69 doub
  @ 4,23 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
  @ 5,19 to 5,59 doub
  @ 6,29 to 8,50 doub
  @ 7,31 say 'LEOMIN SUPERMARKET'
  @ 11,18 to 11,61
  @ 10,34 say 'ENQUIRY MENU'
  @ 9,17 to 20,62
  @ 17,18 to 17,61
  @ 12,21 say 'A ---------- INDIVIDUAL STOCK POSITION'
  @ 14,21 say 'B ---------- GENERAL STOCK POSITION'
```

```
  @ 16,21 say 'Q ---------- QUIT'
  @ 19,30 say 'PICK YOUR CHOICE:'
  do whil .t.
   resp = ' '
   @ 19,48 get resp pict '!'
   read
   if resp $ 'ABQ'
    exit
   endi
  endd
  do case
   case resp = 'A'
    do enq1
   case resp = 'B'
    do enq2
   othe
    exit
  endc
endd
clea
retu


ENQ1.PRG
use code
do whil .t.
 clea
 mcode = spac(7)
 @ 5,16 to 19,62 doub
```

```
@ 17,17 to 17,61
@ 8,27 say 'INDIVIDUAL STOCK POSITION'
@ 7,25 to 9,53
@ 11,18 say 'ITEM CODE (OR "'
@ 11,33 say chr(27)+chr(196)+chr(217) +'" KEY TO EXIT):'
@ 11,52 get mcode
read
if mcode=spac(7)
  exit
endi
go top
loca for code=mcode
if .not. found()
  @ 18,21 say 'INVALID ITEM CODE NO - Press any key '
  set cons off
  wait
  set cons on
  @ 18,18 clea to 18,61
  loop
endi
mdesc=desc
mqty=qty
@ 13,18 say 'DESCRIPTION:' get mdesc
@ 15,18 say 'BALANCE OF STOCK:' get mqty pict '9,999,999,999'
@ 18,21 say 'VIEWING STOCK BALANCE - PRESS ANY KEY'
set cons off
wait
set cons on
endd
```

```
use
clea
retu


ENQ2.PRG
use code
clea
@ 0,7 to 24,72 doub
@ 1,27 to 3,52 doub
@ 2,29 say 'GENERAL STOCK POSITION'
@ 4,8 to 4,71
@ 5,9 say 'S/NO'
@ 5,16 say 'ITEM CODE'
@ 5,32 say 'DESCRIPTION'
@ 5,62 say 'QUANTITY'
@ 5,14 to 21,14
@ 5,26 to 21,26
@ 5,59 to 21,59
@ 6,8 to 6,13
@ 6,15 to 6,25
@ 6,27 to 6,58
@ 6,60 to 6,71
@ 22,8 to 22,71
r=7
sno=0
do whil .not. eof()
  sno=sno+1
  mcode=code
```

```
 mqty = qty
 mdesc = desc
 @ r,9 get sno pict '9999'
 @ r,16 get mcode
 @ r,28 get mdesc
 @ r,61 get mqty pict '9999999999'
 clea gets
 skip
 if eof()
   @ 23,19 say 'IT IS END OF FILE - PRESS ANY KEY TO EXIT'
   set cons off
   wait
   set cons on
   exit
 endi
 r = r + 2
 if r > 21
   @ 23,16 say 'LISTING SALES ENTRY - PRESS ANY KEY TO CONTINUE'
   set cons off
   wait
   set cons on
   @ 7,8 clea to 21,13
   @ 7,15 clea to 21,25
   @ 7,27 clea to 21,58
   @ 7,60 clea to 21,71
   r = 7
 endi
endd
clos all
```

```
clea
retu


CODE.PRG
do whil .t.
 clea
 @ 1,10 to 23,69 doub
 @ 2,19 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
 @ 3,19 to 3,59 doub
 @ 4,29 to 6,50 doub
 @ 5,31 say 'LEOMIN SUPERMARKET'
 @ 9,22 to 9,57
 @ 8,29 say 'ITEM CODE UPDATE MENU'
 @ 7,21 to 22,58
 @ 19,22 to 19,57
 @ 10,25 say 'A ---------- NEW CODE ENTRY'
 @ 12,25 say 'B ---------- VIEW CODE ENTRY'
 @ 14,25 say 'C ---------- CHANGE CODE ENTRY'
 @ 16,25 say 'D ---------- DELETE CODE ENTRY'
 @ 18,25 say 'Q ---------- QUIT'
 @ 21,30 say 'PICK YOUR CHOICE:'
 do whil .t.
  resp = ' '
  @ 21,48 get resp pict '!'
  read
  if resp $ 'ABCDQ'
   exit
  endi
```

```
 endd
 do case
   case resp='A'
     do ncode
   case resp='B'
     do vcode
   case resp='C'
     do ccode
   case resp='D'
     do dcode
   othe
     exit
  endc
endd
clea
retu


NCODE.PRG
use code
do whil .t.
  clea
  @ 2,11 to 22,68 doub
  @ 3,31 say 'LEOMIN SUPERMARKET'
  @ 4,12 to 4,67 doub
  @ 6,28 to 8,50 doub
  @ 7,30 say 'NEW CODE ENTRY FORM'
  mcode=spac(7)
  @ 10,14 say 'ITEM CODE: (OR "'
```

```
@ 10,30 say chr(27)+chr(196)+chr(217) +'" KEY TO EXIT):'
@ 10,50 get mcode pict '@!'
read
if mcode=spac(7)
  exit
endi
go top
loca for code=mcode
if found()
  @ 18,19 to 20,60 doub
  @ 19,21 say 'DUPLICATING ITEM CODE - Press any key '
  set cons off
  wait
  set cons on
  @ 18,19 clea to 20,60
  loop
endi
mdesc=spac(30)
stor 0 to mcp,msp
@ 12,14 say 'DESCRIPTION OF ITEM:' get mdesc pict '@!'
@ 14,14 say 'COST PRICE OF ITEM:' get mcp pict '9,999,999.99'
@ 16,14 say 'SELLING PRICE OF ITEM:' get msp pict '9,999,999.99'
read
@ 18,26 to 20,52 doub
@ 19,28 say 'TO UPDATE FILE (Y/N):'
do whil .t.
  resp=' '
  @ 19,50 get resp pict '!'
  read
```

```
    if resp $ 'YN'
      exit
    endi
  endd
  if resp = 'Y'
    appe blan
    repl code with mcode,desc with mdesc,cp with mcp,sp with msp,qty with 0
  endi
endd
use
clea
retu


VCODE.PRG
use code
do whil .t.
  clea
  @ 2,11 to 22,68 doub
  @ 3,31 say 'LEOMIN SUPERMARKET'
  @ 4,12 to 4,67 doub
  @ 6,28 to 8,51 doub
  @ 7,30 say 'VIEW CODE ENTRY FORM'
  mcode = spac(7)
  @ 10,14 say 'ITEM CODE: (OR "'
  @ 10,30 say chr(27) + chr(196) + chr(217)  + '" KEY TO EXIT):'
  @ 10,50 get mcode pict '@!'
  read
  if mcode = spac(7)
```

```
    exit
  endi
  go top
  loca for code = mcode
  if .not. found()
    @ 18,21 to 20,58 doub
    @ 19,23 say 'INVALID ITEM CODE - PRESS ANY KEY'
    set cons off
    wait
    set cons on
    @ 18,19 clea to 20,60
    loop
  endi
  mdesc = desc
  mcp = cp
  msp = sp
  @ 12,14 say 'DESCRIPTION OF ITEM:' get mdesc pict '@!'
  @ 14,14 say 'COST PRICE OF ITEM:' get mcp pict '9,999,999.99'
  @ 16,14 say 'SELLING PRICE OF ITEM:' get msp pict '9,999,999.99'
  clea gets
  @ 18,21 to 20,58 doub
  @ 19,23 say 'VIEWING ITEM CODE - PRESS ANY KEY'
  set cons off
  wait
  set cons on
endd
use
clea
retu
```

```
CCODE.PRG
use code
do whil .t.
  clea
  @ 2,11 to 22,68 doub
  @ 3,31 say 'LEOMIN SUPERMARKET'
  @ 4,12 to 4,67 doub
  @ 6,27 to 8,52 doub
  @ 7,29 say 'CHANGE CODE ENTRY FORM'
  mcode = spac(7)
  @ 10,14 say 'ITEM CODE: (OR "'
  @ 10,30 say chr(27) + chr(196) + chr(217)  + '" KEY TO EXIT):'
  @ 10,50 get mcode pict '@!'
  read
  if mcode = spac(7)
    exit
  endi
  go top
  loca for code = mcode
  if .not. found()
    @ 18,21 to 20,58 doub
    @ 19,23 say 'INVALID ITEM CODE - Press any key '
    set cons off
    wait
    set cons on
    @ 18,19 clea to 20,60
    loop
  endi
```

```
  mdesc = desc
  mcp = cp
  msp = sp
  @ 12,14 say 'DESCRIPTION OF ITEM:' get mdesc pict '@!'
  @ 14,14 say 'COST PRICE OF ITEM:' get mcp pict '9,999,999.99'
  @ 16,14 say 'SELLING PRICE OF ITEM:' get msp pict '9,999,999.99'
  read
  @ 18,26 to 20,52 doub
  @ 19,28 say 'TO UPDATE FILE (Y/N):'
  do whil .t.
    resp = ' '
    @ 19,50 get resp pict '!'
    read
    if resp $ 'YN'
      exit
    endi
  endd
  if resp = 'Y'
    appe blan
    repl desc with mdesc,cp with mcp,sp with msp
  endi
endd
use
clea
retu


DCODE.PRG
use code
```

```
do whil .t.
  clea
  @ 2,11 to 22,68 doub
  @ 3,31 say 'LEOMIN SUPERMARKET'
  @ 4,12 to 4,67 doub
  @ 6,27 to 8,52 doub
  @ 7,29 say 'DELETE CODE ENTRY FORM'
  mcode = spac(7)
  @ 10,14 say 'ITEM CODE: (OR "'
  @ 10,30 say chr(27) + chr(196) + chr(217) + '" KEY TO EXIT):'
  @ 10,50 get mcode pict '@!'
  read
  if mcode = spac(7)
    exit
  endi
  go top
  loca for code = mcode
  if .not. found()
    @ 18,21 to 20,58 doub
    @ 19,23 say 'INVALID ITEM CODE - PRESS ANY KEY'
    set cons off
    wait
    set cons on
    @ 18,19 clea to 20,60
    loop
  endi
  mdesc = desc
  mcp = cp
  msp = sp
```

```
  @ 12,14 say 'DESCRIPTION OF ITEM:' get mdesc pict '@!'
  @ 14,14 say 'COST PRICE OF ITEM:' get mcp pict '9,999,999.99'
  @ 16,14 say 'SELLING PRICE OF ITEM:' get msp pict '9,999,999.99'
  clea gets
  @ 18,24 to 20,55 doub
  @ 19,26 say 'TO DELETE ITEM CODE (Y/N):'
  do whil .t.
    resp = ' '
    @ 19,53 get resp pict '!'
    read
    if resp $ 'YN'
      exit
    endi
  endd
  if resp = 'Y'
    dele
    pack
    @ 18,21 to 20,58
    @ 19,23 say 'RECORD IS DELETED - PRESS ANY KEY'
  else
    @ 18,19 to 20,60
    @ 19,21 say 'RECORD IS NOT DELETED - PRESS ANY KEY'
  endi
  set cons off
  wait
  set cons on
endd
use
clea
```

```
retu


REP.PRG
do whil .t.
 clea
 @ 3,10 to 21,69 doub
 @ 4,19 say 'COMPUTERISED SALES RECORDING AND ANALYSIS'
 @ 5,19 to 5,59 doub
 @ 6,29 to 8,50 doub
 @ 7,31 say 'LEOMIN SUPERMARKET'
 @ 11,16 to 11,63
 @ 10,29 say 'REPORT GENERATION MENU'
 @ 9,15 to 20,64
 @ 17,16 to 17,63
 @ 12,19 say 'A ---------- DAILY SALES ANALYSIS'
 @ 14,19 say 'B ---------- WEEKLY SALES ANALYSIS'
 @ 16,19 say 'Q ---------- QUIT'
 @ 19,30 say 'PICK YOUR CHOICE:'
 do whil .t.
  resp = ' '
  @ 19,48 get resp pict '!'
  read
  if resp $ 'ABQ'
    exit
  endi
 endd
 do case
  case resp = 'A'
```

```
    do rep1
  case resp = 'B'
    do rep2
  othe
    exit
 endc
endd
clea
retu
```

REP1.PRG

```
use dsales
mdate = date
sort on code to temp.dbf
use temp
clea
set devi to prin
@ 1,31 say 'LEOMIN SUPERMARKET'
@ 2,31 say repl('=',18)
@ 4,24 say 'DAILY SALES REPORT FOR ' + dtoc(mdate)
@ 5,24 say repl('=',31)
@ 6,0 say repl('-',80)
@ 7,0 say '| S/NO |    DESCRIPTION'
@ 7,40 say '|    PRICE'
@ 7,53 say '|'
@ 7,57 say 'QTY'
@ 7,63 say '|  SALES AMOUNT |'
@ 8,0 say '|------|'
```

```
@ 8,8 say repl('-',32) + '¦'
@ 8,41 say repl('-',12) + '¦'
@ 8,54 say repl('-',9) + '¦'
@ 8,64 say repl('-',15) + '¦'
sno = 0
gtot = 0
totamt = 0
totqty = 0
r = 8
do whil .not. eof()
  mcode = code
  mqty = qty
  mdesc = desc
  mprice = price
  mamt = amount
  totamt = totamt + mamt
  totqty = totqty + mqty
  skip
  if mcode = code
    loop
  endi
  sno = sno + 1
  r = r + 1
  @ r,0 say '¦'
  @ r,2 say sno pict '9999'
  @ r,7 say '¦'
  @ r,9 say mdesc
  @ r,40 say '¦'
  @ r,42 say mprice pict '999,999.99'
```

```
  @ r,53 say '¦'
  @ r,55 say totqty pict '999,999'
  @ r,63 say '¦'
  @ r,65 say totamt pict '99,999,999.99'
  @ r,79 say '¦'
  gtot = gtot + totamt
  totamt = 0
  totqty = 0
  if eof()
    exit
  endi
  r = r + 1
  @ r,0 say '¦'
  @ r,7 say '¦'
  @ r,40 say '¦'
  @ r,53 say '¦'
  @ r,63 say '¦'
  @ r,79 say '¦'
endd
r = r + 1
@ r,0 say '¦' + repl('-',78) + '¦'
r = r + 1
@ r,0 say '¦'
@ r,35 say 'G R A N D - T O T A L  : -'
@ r,65 say gtot pict '99,999,999.99'
@ r,79 say '¦'
r = r + 1
@ r,0 say repl('=',80)
set devi to scre
```

```
wait
clos all
clea
retu


REP2.PRG
use tsales
mdate=date
sort on code to temp.dbf
use temp
clea
set devi to prin
@ 1,31 say 'LEOMIN SUPERMARKET'
@ 2,31 say repl('=',18)
@ 4,23 say 'WEEKLY SALES REPORT AS AT '+dtoc(mdate)
@ 5,23 say repl('=',34)
@ 6,0 say repl('-',80)
@ 7,0 say '| S/NO |    DESCRIPTION'
@ 7,40 say '|    PRICE'
@ 7,53 say '|'
@ 7,57 say 'QTY'
@ 7,63 say '|  SALES AMOUNT |'
@ 8,0 say '|------|'
@ 8,8 say repl('-',32)+'|'
@ 8,41 say repl('-',12)+'|'
@ 8,54 say repl('-',9)+'|'
@ 8,64 say repl('-',15)+'|'
sno=0
```

```
gtot = 0
totamt = 0
totqty = 0
r = 8
do whil .not. eof()
  mcode = code
  mqty = qty
  mdesc = desc
  mprice = price
  mamt = amount
  totamt = totamt + mamt
  totqty = totqty + mqty
  skip
  if mcode = code
    loop
  endi
  sno = sno + 1
  r = r + 1
  @ r,0 say '¦'
  @ r,2 say sno pict '9999'
  @ r,7 say '¦'
  @ r,9 say mdesc
  @ r,40 say '¦'
  @ r,42 say mprice pict '999,999.99'
  @ r,53 say '¦'
  @ r,55 say totqty pict '999,999'
  @ r,63 say '¦'
  @ r,65 say totamt pict '99,999,999.99'
  @ r,79 say '¦'
```

```
  gtot = gtot + totamt
  totamt = 0
  totqty = 0
  if eof()
    exit
  endi
  r = r + 1
  @ r,0 say '¦'
  @ r,7 say '¦'
  @ r,40 say '¦'
  @ r,53 say '¦'
  @ r,63 say '¦'
  @ r,79 say '¦'
endd
r = r + 1
@ r,0 say '¦' + repl('-',78) + '¦'
r = r + 1
@ r,0 say '¦'
@ r,35 say 'G R A N D - T O T A L  : -'
@ r,65 say gtot pict '99,999,999.99'
@ r,79 say '¦'
r = r + 1
@ r,0 say repl('=',80)
set devi to scre
wait
clos all
clea
retu
```