1-1-1983

Design and implementation of data base management system for a banking environment.

Mustafa Mahmoud Yaseen

DESIGN AND IMPLEMENTATION OF DATA BASE MANAGEMENT SYSTEM FOR A BANKING ENVIRONMENT

by

MUSTAFA MAHMOUD YASEEN

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computinq Science

Lehigh University

1983

Certificate of Approval

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

Dec. 7, 1982
(date)

Professor in Charge

Chairman of Department

Acknowledgment

The author wishes to thank Professor Andrew J. Kasarda for his helpful suggestions and thoughts in preperation of this thesis.

Table of Contents

1. ABSTRACT

DESIGN AND IMPLEMENTATION OF DATA BASE MANAGEMENT SYSTEM FOR A BANKING ENVIRONMENT

by Mustafa Mahmoud Yaseen

A brief introduction to Data Base Management System is presented. Merits of the Data Base approach over conventional file-oriented approach are discused. This thesis addresses the task of analysing and converting an existing file-oriented Banking Environment to a Data Base Management System. The DECSYSTEM-20's DBMS has been used to design an online information system. Some application programs are written in order to demonstrate the system capabilities.

2. INTRODUCTION

Data Base Management Systems have evolved to the point of general acceptance and wide applications.

> A **DATA BASE** may be defined as a collection of interrelated data stored together without harmful or unnecessary redundancy to serve multiple applications ; the data are stored so that they are independent of programs which use the data ; a common and controlled approach is used in adding new data and in modifying and retrieving existing data whithin the data base. The data is structured so as to provide a foundation for future application development.[1]

CHANGE_AND_GROWTH

A very important feature of a data base is that it will constantly need to change and grow. It has to be easy to add, modify, or delete any new applications or new data types. These functions should be possible without rewritting all other application programs.

---

[1] COMPUTER DATA-BASE ORGANIZATION : BY JAMES MARTIN. pg. 22.

DATA_INDEPENDENCE

One of the main attributes of a data base is "DATA INDEPENDENCE". This means that the data and the application programs which use them are independent so that changing any one of these will not result in a change of the other. Although in reality, to find a completely independent data base is very rare. Nevertheless, data independence is one of the principal reasons for using a data base system.

There are many other desirable characteristics of data base systems, some of those are the following :

PERFORMANCE

The responce time experienced by data base applications designed for interactive use, must be appropriate for a real-time dialogue. It must also be able to handle an appropriate variety of transactions.

MINIMAL_REDUNDANCY

In a file oriented data-processing systems, a high level of redundancy tends to develop. Where, as mentioned earlier, a data base system should have a low level of controlled redundancy, in addition inconsistencies that are generated by redundant data must be carefully monitored.

SEARCH CAPACITY

One of the main objectives of the data base system is the ability to answer queries from users about stored data.

> Most of the queries are anticipated with a suitable speed. Unanticipated queries make it necessary to search parts of the data base, and the average response time for a given search depends on the nature of the searching techniques used in the data base system.[2]

Access mechanisms and addressing methods should be fast enough to support the real-time requirements of the application in question. The need for fast spontaneous

----

[2] COMPUTER DATA-BASE ORGANIZATION : by JAMES MARTIN. pg. 37.

searching of the data will increase as interactive systems usage spreads.

INTEGRITY

Hardware failures and various types of malfunctions will occur occasionally, so it is imperative that data items and associations between data items not be destroyed. Recovery of the data should be possible without any harm to the stored data .

> Maintaining the integrity of the data base can be viewed as protecting the data against invalid (as opposed to illegal) alteration or destruction.[3]

PRIVACY_AND_SECURITY

Data must be kept secure and private in a data base. No interprise can affored to lose its data or its integrity. The interprise has to provide the

[3] An Introduction to Database Systems: by C.J. DATE. pg. 395.

protection of the data against accidental or intentional disclosure to unautherizied persons, or unautherized modifications, or destruction or any kind of any missuse.

> Privacy is a right of individuals and organizations to determine for themselves when, and how, and to what extent information about them is to be transmitted to others.[4]

There are many fields where data base concept can be applied to, and a data base management system should be good enough to replace the old file oriented systems. A good example for applying the data base concept is a banking environment, where data is constantly retrieved and updated, added, modified, or deleted.

The response time of the system to queries by

---

[4] COMPUTER DATA-BASE ORGANIZATION : by JAMES MARTIN. pg. 38

input users is very critical for a bank. Users want fast accurate answers to queries about stored data.

The privacy and security issue is very important in the banking environment. Nothing is more important for any one, more than keeping his personal financial status secure and private. By using a data base management system unauthorized access to the data can be closely controlled. The same data may be restricted in different ways from different uses. By introducing privacey locks it can be insured that no unauthorized person can access the data base. The transparency of the data base made it possible for users to get and manipulate data relevant to their needs.

Storing new items, modifying existing items or deleting old ones has to be independent of the application programs. The interactive access method is essential for users in a bank. This implies that a suitably fast search algorithm be employed, and this thing can be found in a good data base management system.

In this paper I employed a theoritical banking model. I converted an existing file oriented system into a data base management system without losing any important feature of the old system.

To implement the conversion process from the old data files oriented system to the data base management system, the following steps were taken :

1. Design a conceptual model of the data base:

   - Study the environment, and document assumptions for it.
   - Determine the data elements refrenced in every report individually.
   - Determine the relationships between the data elements.

2. Design a logical model of the data base: Draw a logical model based on the conceptual model for a Data Base Management System using a NETWORK DATA MODEL.

3. Design a physical model of the data base: Draw a physical model on the basis of the logical model.

4. The system may be put into use on experimental basis. Errors which may occur should be reported to the Data Base Administrator.

3. DESIGN A CONCEPTUAL MODEL OF THE DATA BASE

3.1 NOTES AND ASSUMPTIONS

For the past few years BANKs have traditionally employed a file-oriented data processing system that operated in batch mode. It provided turnaround within one to three days depending on the type of activity involved. However with the increased number of accounts and volume of daily work, management insists that something had to be done to allow for tighter controls and availability of accurate information. For this capability management determined that a full-fledged, on-line DATA BASE MANAGEMENT SYSTEM (DBMS) be installed to replace the present system.

A BANK usually has number of branches scattered throughout a city and its suburbs. The main office will be considered as a branch, and all the branches will perform the same activities. This design will concentrate only on the activities related to customer accounts and no other activities will be taken into consideration. A customer may walk into any branch and open an account. A customer may open more than one account. The BANK maintains several types of accounts:

1. **CHECKING ACCOUNTS** : In which customer may write personal checks, no interest is charged.

2. **SAVING ACCOUNTS** : In which passbooks are required and interest is accumulated.

3. **LOAN ACCOUNTS** : In which an amount of money is given by the bank to a customer as a loan with interest for a certain period of time. The customer must return the loan with interest.

4. **TIME DEPOSIT ACCOUNTS** : In which the customer will deposit a given amount of money for a fixed period. At the end of the period the bank will pay the assigned amount of money plus interest based on a rate quoted to the customer for his deposit. He is not allowed to withdraw any amount of his deposit during this period.

The first step taken in designing a data base for the BANK was to analize the present system. This system consisted of several files, and some reports. A full listing can be found in the next section, description of some of the reports will follow :-

1. **OVERDRAWN ACCOUNTS REPORT:** This report is printed for the branch manager on a daily basis. The report consists of all the exceptional transactions of the the over-drawn accounts, that is the checking

accounts which have the overdraft option.

2. **EXCEPTIONAL SAVING ACCOUNTS REPORT:** This report is printed for the branch manager on a daily basis and contains all the saving accounts in which the balance exceeds $ 100,000, which is the maximum amount insured by the FDI (Federal Deposit Insurance Co.).

3. **BRANCH LOAN STATUS REPORT:** This report helps branch manager to keep track of the loans given by his branch.

4. **TELLER CASH DRAWER REPORT:** Employees who cash checks, enter deposits, and make withdrawals are called tellers. This daily report keeps track of the money flow for that day for that teller, at a particular branch. Tellers are sometimes rotated between branches. However, it is assumed that if teller is sent to a particular branch he/she stays at that branch the entire day.

5. **DAILY TRANSACTIONS REPORT:** All transactions carried out on that work day will appear in this report, along with all the information about the accounts involved in the transaction. At the end of the report a summary displaying the number and total amounts of all the transactions (DEBIT & CREDIT) is produced.

6. **FOREIGN EXCHANGE REPORT:** This is a daily report that shows all the foreign currency transactions, with all related and required information.

7. **TELLER AUDIT REPORT:** This is a periodical report. Each teller has a limit that he/she can't exceed in withdrawal. This is an exception report to see if some teller might

have exceeded his/her limit. It is a batch report.

8. **INQUIRY TRANSACTION:** Gives information about the status of a particular account, such as balance, inquiries about a certain transaction, etc. It is an on line component.

9. **DEPOSIT/WITHDRAWAL TRANSACTIONS:** There are two types of transactions : TYPE 1 which is a deposit, and TYPE 2 which is a withdrawal. Deposit and withdrawal are implemented, however other types of transactions such as accounts corrections can b done by adding an application program.

10. **RETURNED CHECKS REPORT:** This is a daily report that provides branch manager with the information about bad accounts. These are the accounts which fail to pass checks presented to the bank because they are drawn on accounts with insufficient funds.

3.2 FILES OF THE OLD SYSTEM

In the following pages a listing of the old system files is presented. These files are used as basis for designing the new Data Base Management System. Along with each file there is a description of the file key listed below.

The first file is **THE ACCOUNTS FILE.** The type of the account is determined by the following code : "1" for CHECKING ACCOUNTS, "2" for SAVING ACCOUNTS. And finally "3" for TIME DEPOSIT ACCOUNTS. Also, all customer information can be found in this file, along with the associated branch number.

The second file is **THE OVERDRAFT ACCOUNTS FILE.** In this file all the accounts or to be more specific, all the checking accounts which have been given the facility of overdraft, are found. It has been decided to keep the overdraft accounts separatelly because (1) not all the accounts have this facility, and (2) to maintain control of these customers because it is a kind of investment.

The third file is **LOANS FILE** where the loans accounts are found along with all related information .

The forth and fifth files are the **DAILY TRANSACTIONS FILE** & the **FOREIGN EXCHANGE TRANSACTIONS FILE,** respectively. They are not combined because the volume of the second is much less than the first. All of the relevant information can be found in these two files.

The sixth file is **BRANCH DRAWER FILE,** and the seventh file is **AUDIT FILE,** and finally **PAYROLL FILE** where all the information about an employee can be found. And, of course, this file is used to meet the salaries and all the requirements related to the personnel department.

ACCOUNTS FILE

CUSTOMER NAME	PIC	X(30)
SSN	PIC	9(9)
TELEPHONE NO	PIC	9(10)
ADDRESS		
STREET	PIC	X(20)
CITY-STATE	PIC	X(17)
ZIP	PIC	9(5)
ACCOUNT NUMBER	PIC	9(9)
ACCOUNT TYPE *	PIC	9
AVAILABLE BALANCE	PIC	9(7)V99
CURRENT BALANCE	PIC	9(7)V99
MONTHLY RET CHECKS	PIC	9
YEARLY RET CHECKS	PIC	99
INTEREST AMOUNT	PIC	9(7)V99
TIME DEP AMOUNT	PIC	9(7)V99
DATE TD START	PIC	9(6)
DATE TD END	PIC	9(6)

The key for this file is :
Social Security Number (SSN).

* TYPE 1 for CHECKING ACC.
TYPE 2 for SAVING ACC.
TYPE 3 for TIME DEPOSIT ACCOUNTS.

OVERDRAFT ACCOUNTS FILE

OVER DRAFT NUMBER	PIC	9(7)
OVER DRAFT LIMIT	PIC	9(8)V99
CHECKING ACCOUNT NUMBER	PIC	9(9)
OFFICER NUMBER	PIC	9(6)
START DATE	PIC	9(6)
DUE DATE	PIC	9(6)
SECURITY CODE *	PIC	X(2)

The key for this file is :
OVER DRAFT NUMBER.

* TD : Time Deposit.
BD : Bonds.
RE : Real Estate.
SR : Shares.
AT : Assets.
OR : Others.

LOANS FILE

LOAN NUMBER	PIC	9(9)
LOAN TYPE	PIC	X(30)
CUST NAME	PIC	X(30)
OFFICER NUMBER	PIC	9(6)
AMOUNT	PIC	9(7)V99
INTEREST	PIC	V99
START DATE	PIC	9(6)
END DATE	PIC	9(6)
TOTAL AMOUNT DUE	PIC	9(7)V99
INSTALLMENTS	PIC	9(7)V99
TOTAL AMOUNT COLLECTED	PIC	9(7)V99

The key is LOAN NUMBER .

DAILY TRANSACTIONS FILE

TRANSACTION DATE	PIC	9(6)
TRANSACTION TIME	PIC	9(6)
TRANSACTION TYPE	PIC	9
ACCOUNT NUMBER	PIC	9(9)
TELLER NUMBER	PIC	9(6)
BRANCH NUMBER	PIC	9(3)
BRANCH NAME	PIC	X(30)
BRANCH MANAGER NAME	PIC	X(30)
BRANCH ADDRESS		
STREET	PIC	X(20)
CITY	PIC	X(17)
STATE	PIC	X(2)
ZIP	PIC	9(5)
TRANSACTION AMOUNT	PIC	9(7)V99
CHECK NUMBER	PIC	9(7)
PASSBOOK LINE NUMBER	PIC	9(7)

The key is :
TRANSACTION DATE & TIME.

EXCHANGE TRANSACTIONS FILE

CURRENCY NAME	PIC	X(15)
CURRENCY RATE		
BUYING	PIC	99V99
SELLING	PIC	99V99
ACCOUNT NUMBER	PIC	9(9)
TELLER NUMBER	PIC	9(6)
BRANCH NUMBER	PIC	9(3)
BRANCH NAME	PIC	X(30)
BRANCH MANAGER NAME	PIC	X(30)
BRANCH ADDRESS		
STREET	PIC	X(20)
CITY	PIC	X(17)
STATE	PIC	X(2)
ZIP	PIC	9(5)
EQUIVOLLENT AMT IN $	PIC	9(7)V99
EXCHANGE CODE	PIC	X(20)

The key for this file is CURRENCY NAME .

BRANCH DRAWER FILE

DRAWER NUMBER	PTC	9(4)
DATE	PTC	9(6)
BRANCH NUMBER	PTC	9(3)
TELLER NUMBER	PTC	9(6)
CASH ON HAND START	PTC	9(7)V99
AMT RECEIVED ON OTHERS	PTC	9(7)V99
AMT RECEIVED ON US	PTC	9(7)V99
CASH DISPENSED	PTC	9(7)V99
CASH RECIEVED	PTC	9(7)V99
NO CHECKS RECIEVED ON US	PTC	9(4)
NO CHECKS RECIEVED ON OTHERS	PTC	9(4)
NO CHECKS DISPENSED	PTC	9(4)
AMT CHECKS DISPENSED	PTC	9(7)V99
FOREIGN CURRENCY	PTC	9(7)V99

The key for this file is :
DRAWER NUMBER & DATE.

AUDIT FILE

AUDIT NUMBER	PIC	9(7)
BRANCH NUMBER	PIC	9(3)
TELLER NUMBER	PIC	9(6)
REASON	PIC	X(30)
START DATE	PIC	9(6)
END DATE	PIC	9(6)
TOTAL NUMBER OF DEPOSITS	PIC	9(6)
TOTAL NUMBER OF WITHDRAWALS	PIC	9(6)
MAX AMT DEPOSITED	PIC	9(7)V99
MAX AMT WITHDRAWN	PIC	9(7)V99

The key for this record is AUDIT NUMBER .

PAYROLL FILE

SS NUMBER	PTC	9(10)
EMPL NUMBER	PTC	9(6)
NAME	PTC	X(30)
ADDRESS		
STREET	PTC	X(20)
CITY	PTC	X(17)
STATE	PTC	X(2)
ZIP	PTC	9(5)
STATUS	PTC	X(7)
BRANCH NUMBER	PTC	9(3)
LIMIT	PTC	9(7)V99
START OF SERVICE	PTC	9(6)
END OF SURVICE	PTC	9(6)
MARTIAL STATUS	PTC	X
NUMBER OF DEPENDENTS	PTC	9
SEX	PTC	X
SALARY	PTC	9(5)V99
REG HR RATE	PTC	99V99
LAST MONTH DATA		
REG HRS	PTC	9(3)
OVT HRS	PTC	9(3)
VAC HRS	PTC	9(3)
SIK HRS	PTC	9(3)
HOL HRS	PTC	9(3)
GROSS PAY	PTC	9(4)V99
YEAR TO DATE DATA		
YTD REG HRS	PTC	9(4)
YTD OVT HRS	PTC	9(4)
YTD VAC HRS	PTC	9(4)
YTD SIK HRS	PTC	9(4)
YTD HOL HRS	PTC	9(4)
YTD GROSS PAY	PTC	9(6)V99
YTD FED TAX	PTC	9(6)V99
YTD STATE TAX	PTC	9(5)V99
YTD LOCAL TAX	PTC	9(4)V99
YTD SS TAX	PTC	9(4)V99

```
EMPL BENIFITS
     RETIREMENT                    PIC       99
     LIFE INSURANCE                PIC       99V99
     HEALTH INSURANCE              PIC       99V99
     DENTAL PLANE                  PIC       99V99
```

The key for this report is
SOCIAL SECURITY NUMBER.

3.3 USES OF THE DATA BASE

The uses for which the data base should be designed are summarized below :

1. List all types of accounts (checking, saving, time deposit, loan).
2. List names of all customers.
3. List all overdrawn accounts.
4. Given a SS give the account number and vice a versa.
5. List all bad accounts(those of returned checks).
6. List all saving accounts which has balance more than $ 100,000.
7. Give total of overdrawn amounts, number of accounts and transactions involved.
8. Give total of exceeding saving accounts, number of accounts and transactions involved. Given a transaction in a certain day, along with the teller.
9. Information about an account, like the name of the customer, his balance, an inquiry about a certain transaction, etc...
10. List all transactions that took place on a certain work day. This includes at the end also a total of all the deposit transactions and the withdrawal transactions.
11. Display the restrictions placed on the activities of particular teller, or officer.

12. List all branches.

13. List all employees.

14. The branch at which a particular employee is assigned to (this includes the branch manager).

15. The status of an employee (Officer, Teller, or Clerk).

16. List all tellers, officers, clerks of the bank by branch.

17. List all the foreign currency activity that is the transactions, and the accounts involved with the name of currency and equivalent amount in dollars.

3.4 ASSUMPTIONS ABOUT THE BANKING ENVIRONMENT

1. The customer SOCIAL SECURITY NUMBER is unique.

2. Account numbers are allotted bankwide, that is, the account numbers are unique.

3. Account numbers consists of 9 digits, the first two digits determine the account type, the last seven digits are the account serial number.

4. A customer can open as many accounts as he likes (of course with some limitations).

5. Branch number is unique.

6. Each branch has only one manager at a given point in time.

7. Employee numbers are unique. This includes tellers, officers and all other employees.

8. Employees are divided into three categories:

   - TELLERS : Previously described.

   - OFFICERS : Include all managers, and all other supervisors.

   - CLERKS : Include all other employees.

9. Employees of the bank are rotated between all the branches, but once assigned to a certain branch he/she will remain there all that day.

10. Each teller has a limit that he/she can't

exceed for withdrawal.

11. Each officer has a limit he/she can't exceed in giving loans or giving some overdraft facility to a customer.

12. The overdraft facility is given only to checking accounts, it is amount of money given to a certain customer with an interest over certain time limit and it has to be paid at the end of the period. Not every customer can qualify for it.

13. A customer is said to have a bad account if the number of returned checks exceeds five checks per month, or twenty four checks per a year. If a customer reaches either one of these limits his accounts will be closed.

14. The time deposit can't be less than $ 1000. The interest rate depends on the period of time, the amount and the rates in the market that day, at the end of the time period the total of the amount and the interest is paid to the customer.

15. Each cash drawer has a unique number.

16. Checks presented to the bank to be cashed are passed as one transactions. Checks deposited in an account and drawn on the bank are dealt with in the following way. When they are presented they will pass as a deposit transaction and at the end of the day it will pass as a debit (withdrawal) on the account which issued the check.

17. There are two kinds of transactions debit or withdrawal & credit or deposit

18. For the foreign exchange the buying rate is related to credit or deposit transaction,

and the selling rate is related to debit or withdrawal transaction.

3.5 DATA DICTIONARY AND ITEMS DESCRIPTION

The following is a list of all the record types involved in the data base in alphabetical order. The data elements in each record are listed.

RECORD TYPE	PREFIX
AUDIT RECORD TYPE	AUDT-
BRANCH RECORD TYPE	BR-
CHECKING ACCOUNT RECORD TYPE	CA-
CITYSTATE RECORD TYPE	CYST-
CLERK RECORD TYPE	CLRK-
CUSTOMER RECORD TYPE	CUST-
DAILY TRANSACTIONS RECORD TYPE	DTRN-
DRAWER RECORD TYPE	DRAW-
EMPL RECORD TYPE	EMPL-
EXCHANGE TRANS RECORD TYPE	EXTR-
LOAN ACCOUNT RECORD TYPE	LOAN-
OFFICER RECORD TYPE	OFFR-
OVERDRAFT ACCOUNTS RECORD TYPE	OD-
SAVING ACCOUNT RECORD TYPE	SA-
TELLER RECORD TYPE	TELL-
TIME DEPOSIT ACCOUNT RECORD TYPE	TD-

ZIP RECORD TYPE ZP-

AUDIT RECORD TYPE

AUDT-DATE-END Ending date of an AUDIT.

AUDT-DATE-STA Starting date of an AUDIT.

AUDT-NO This is the key of this record, it is a numeric field which uniquely defines an AUDIT.

AUDT-REASON The reson of an AUDIT, whether it is regular audit or investigation, or exceptional, etc..., it is an alphanumeric field.

MAX-AMT-DEP Largest dollar amount involved in deposit transaction.

MAX-AMT-WITH Largest dollar amount involved in withdrawal trans.

TOT-NO-DEP Total number of deposit trans. performed by a specific teller.

TOT-NO-WITH Total number of withdrawal trans. performed by a specific teller.

BRANCH RECORD TYPE

BR-ADDR Address of a branch, it is subdivide into: ST (Street), CITY, STATE, and ZIP.

BR-MGR-NAME Branch manager name, it is mentioned here because it is involved in a lot of reports.

BR-NAME Name of a branch.

BR-NO This is the key to BRANCH record type, it is uniquely identifies a branch, it is a numeric field consisting of three digits.

CHEKING ACCOUNT RECORD TYPE

AVAIL-BAL Available balance of a checking account is calculated by adding the actual balance of an account plus overdraft limits, blocked amounts, and uncleared checks.

CA-NO Is the key to checking account record type. It is the number assigned for a customer as his/her account, the first two digits are 01 to indicate that the type of account is a checking account, the next seven digits are the account number. It uniquely defines an account.

CURR-BAL Current balance of a checking account.

NO-MONTH-RET The number of returned checks per month. This is a numeric field which has one digit, it is updated monthly to control the returned checks of an account.

NO-YEAR-RET Number of yearly returned checks, it is a two digits numeric field. This field is used to control the returned checks on a yearly basis.

CITYSTATE RECORD TYPE

CITYSTATE This is the name of city and state to which a certain customer belongs. This record is used mainly for the purpose of sorting the mail of customers according to city-state. It is the key for CITYSTATE RECORD TYPE.

CLERK RECORD TYPE

CLRK-NO Clerk number is the key for this record. It uniquely identifies a clerk . This record is added to the data base for future applications.

CUSTOMER RECORD TYPE

CUST-ADDR This is the address of the customer, only the street and/or number.

CUST-SS-NO Customer social security number, is the key for this record.

CUST-NAME Customer name, it is subdivided into CUST-LAST, CUST-FIRST, and CUST-INIT .

CUST-TEL-NO customer telephone number.

DAILY TRANSACTION RECORD TYPE

CHECK-NO Check number is a numeric field usually used in connection with checking accounts transactions (if it is available) otherwise it is ignored.

DTRN-AMT It indicates the amount involved in a transaction.

DTRN-DATE The day on which the transaction took place. The key for this record is formed by the concatenation of DTRN-DATE & DTRN-TIME.

DTRN-TIME The unit of time at which a transaction took place.

DTRN-TYPE The type of transaction, it is subdivided into two items : TYPE 1 which indicates a deposit transaction; TYPE 2 which indicates a withdrawal transaction.

PASSBOOK-LIN-NO The passbook line number is provided only for saving accounts, the passbook line number is provided at the time when a transaction is processed.

DRAWER RECORD TYPE

AMT-CHEK-DISP Dollar amount of dispensed checks.

AMT-RECD-OTHR Dollar amount of checks recieved drawn on other banks.

AMT-RECD-US Dollar amount of checks recieved drawn on us.

CASH-DISP Total cash dispensed during day work.

CASH-RECD Total cash recieved during day work.

COH-START Cash on hand at start of a day .

DRAW-DATE A business day. Every teller has a cash drawer report on a business day. A teller Audit Report for a day or a group of days may be prepared. DATE is synonymous with DTRN-DATE (Daily Transaction Date).

DRAW-NO Drawer number is the key for this record it uniquely identifies each drawer.

NO-CHEK-DISP Number of checks dispensed.

NO-CHEK-ON-OTHR Number of checks drawn on other banks.

NO-CHEK-ON-US Number of checks drawn on us.

FOREIGN-CURR It is used in controlling the flow of foreign currency.

EMPLOYEE RECORD TYPE

EMPL-ADDR The address of employee, it is further subdivided into : STREET, CITY, STATE, ZIP.

EMPL-SS-NO This the social security number, it is used as a key for this record, it is uniquely identifies an employee

EMPL-NAME The name of employee, it is subdivided into : EMPL-FIRST, EMPL-INIT (middle initial), EMPL-LAST .

EMPL-STATUS It is an alpha-numeric field which indicates the status of an employee in the bank; T for TELLER; O for OFFICER; C for CLERK.

EMPL-SERV-DATE The date when employee starts working in the bank.

EMPL-END-DATE The date when employee ends working in the bank, if he/she is still working it will be left empty.

EMPL-MART-STA Employee martial status, i.e. M : MARRIED, S : SINGLE, D : DIVORSED, W : WIDOW .

EMPL-NO-DEP The number of dependents for the employee.

EMPL-SEX M : for Male; F : for female.

SALARY If an employee is salaried.

EMPL-REG-HR-RT Regular pay per hour for the employee. Overtime hour rate is 1 and 1/2 times the regular hour rate.

EMPL-LAST-MON-DATA

This is a group of numeric fields which provide information concerning the employee's last pay period. It contains the following : EMPL-LM-REG-HRS : number of regular hours worked, EMPL-LM-OVT-HRS : number of overtime hours worked, EMPL-LM-VAC-HRS : number of vacation hours used, EMPL-LM-SCK-HRS : number of sickness hours used, EMPL-LM-HOL-HRS :

number of holiday hours used, EMPL-LM-GROSS-PAY : dollar amount of gross pay.

EMPL-YEAR-TO-DAT-DATA

This is a group of data fields which provide information concerning the employee's year to date data. It contains the following : EMPL-YTD-REG-HRS : number of regular hours worked EMPL-YTD-OVT-HRS : number of overtime hours worked EMPL-YTD-VAC-HRS : number of vacation hours used EMPL-YTD-SCK-HRS : number of sickness hours used, EMPL-YTD-HOL-HRS : number of holiday hours used, EMPL-YTD-GROSS-PAY:dollar amount of gross pay, EMPL-YTD-FED-TAX : dollar amount federal tax paid, EMPL-YTD-STAT-TAX: dollar amount state tax paid, EMPL-YTD-LOCAL-TAX: dollar amount local tax paid, EMPL-YTD-SS-TAX : dollar amount social sec. tax paid.

EMPL-BENIFITS It is subdivided into the following:

RETIRMENT Employees may retire at age 55 with 30 years of service, age 60 with 20 years of service or age 62 with 5 years of service.

LIFE-INSURANCE Employees are eligible for optional life insurance in amounts based on their annual salaries. The bank pays portion of the premiums.

HEALTH-INSURANCE

Employees are eligible for comprehensive health insurance and have choice from among many excellent plans.

DENTAL-PLANE Employees are eligible for a dental plane.

EXCHANGE TRANSACTION RECORD TYPE

CURNCY-NAME Currency name is the key for this record, it is an alpha-numeric field.

CURNCY-RATE The rate of the foreign currency with respect to dollars it is subdivided into : BUYING : which is the buying rate of currency ; SELLING: which is the selling rate of currency. This field is updated daily because it changes on a daily basis.

EQ-AMT-$ Equivalent dollars amount for a certain draft.

EXTR-CODE Exchange transaction code that is whether it is a draft, a money order, a money transfer, traveler's checks, etc...

EXTR-DATE The day on which the exchange transaction took place.

EXTR-TIME The unit of time at which an exchange transaction took place.

LOAN ACCOUNT RECORD TYPE

LOAN-AMT — Dollars amount of money taken by a customer as a loan.

LOAN-AMT-COLL — Total amount collected from a loan.

LOAN-DATE-END — The date on which the loan has to be totally collected from a customer.

LOAN-DATE-START — The date on which the period of the loan is started.

LOAN-INSTAL — Amount of money to be paid as installments by customer.

LOAN-INTEREST — Interest rate assigned for a loan depends on the period of time and has to be paid at the end with the loan.

<u>LOAN-NO</u> — Loan number is the key for this record, it uniquely identifies each loan account the first two digits are 04 to indicate the type of the account, the next seven digits are for the account number.

LOAN-TOT-AMT-DUE — Total amount due to be paid at the end of the period of the loan .

LOAN-TYPE — The type of a loan like: Mortgage loan, installment loan, student loan, etc...

OFFICER RECORD TYPE

OFFR-LIMIT — The officer limit it is used in giving loans and overdrafts. The manager (or officer) cannot exceed his limit.

<u>OFFR-NO</u> — This the key for this record, actually it has nothing to do with the officer. It is the employee number. once an employee is hired by the bank he is assigned a number that will uniquely identifies him/her.

OVERDRAFT ACCOUNT RECORD TYPE

OD-DATE-START The date in which this option is given to a customer.

OD-DUE-DATE The last date in which the overdraft facility expires and a decision has to be taken whether to cancel it or renew it, this is after paying the whole amount due on a customer.

OD-INTEREST The interest rate assigned for a customer, it has to be paid by customer at the end of due date along with the amount due on him.

OD-LIMIT The overdraft amount given to a customer as an optoin. It enables him to overdraw his balance up to this amount.

OD-NO This is the key to this record it consists of six digits.

SECUR-CODE Security code is used to indicate on what basis an overdraft was given to a customer, and what will ensure the return of the Bank's money back. It is subdivided into :

- TD : Time Deposit ;
- BD : Bonds ;
- RE : Real Estate ;
- SR : Shares ;
- AT : Assets ;
- OR : Others .

SAVING ACCOUNT RECORD TYPE

SA-BAL Saving account balance.

SA-INTEREST-AMT Dollar amount of interest given to an account. It is kept for tax reporting purposes.

SA-NO This is the key for this record, the first two digits are 02 to indicate the type of the account, the next seven digits are the account number, it is uniquely identifies the account.

TELLER RECORD TYPE

TELL-LIMIT Amount of money a teller is not allowed to exceed in whithdrawal.

TELL-NO It is the key for TELLER record type.

TIME DEPOSIT ACCOUNT RECORD TYPE

TD-AMT Dollar amount of a time deposit .

TD-END-DATE The date in which the period of a Time Deposit expires. After this date a decision has to be taken for either renewal or release according to customer's desire.

TD-INTEREST Interest of a time deposit. It varies according to the amount and period.

TD-NO This the key of this record. It uniquely identifies each time deposit. The first two digits are 03 which indicates this type of account, and the next seven specify the account number.

TD-START-DATR The date on which a time deposit begins.

ZIP RECORD TYPE

ZIPCODE A numeric field that is the key and only field in this record. It is the zip code of customer's address. It is used mainly in sorting the mail according to the zip code to qualify for reduced rates.

3.6 RECORD TYPES RELATIONSHIPS

This system of data consists of 17 (seventeen) record types (which were mentioned previously). What follows is a list of all the relationships between record types. These relationships must exist in order to satisfy the requirements, uses and needs mentioned before. The prefix of a record is used instead of the full name of a record in describing these relationships.

The representation of the relationships will be as follows:[5]

A ONE TO ONE RELATION : <------>

A ONE TO MANY RELATION : <----->>

A MANY TO ONE RELATION : <<----->

A RELATION WITH THE SYSTEM : ------>

[5] DATA BASE, Structured Technieques for Design, Performance, and Management: by S. ATRE. pg. 401-402.

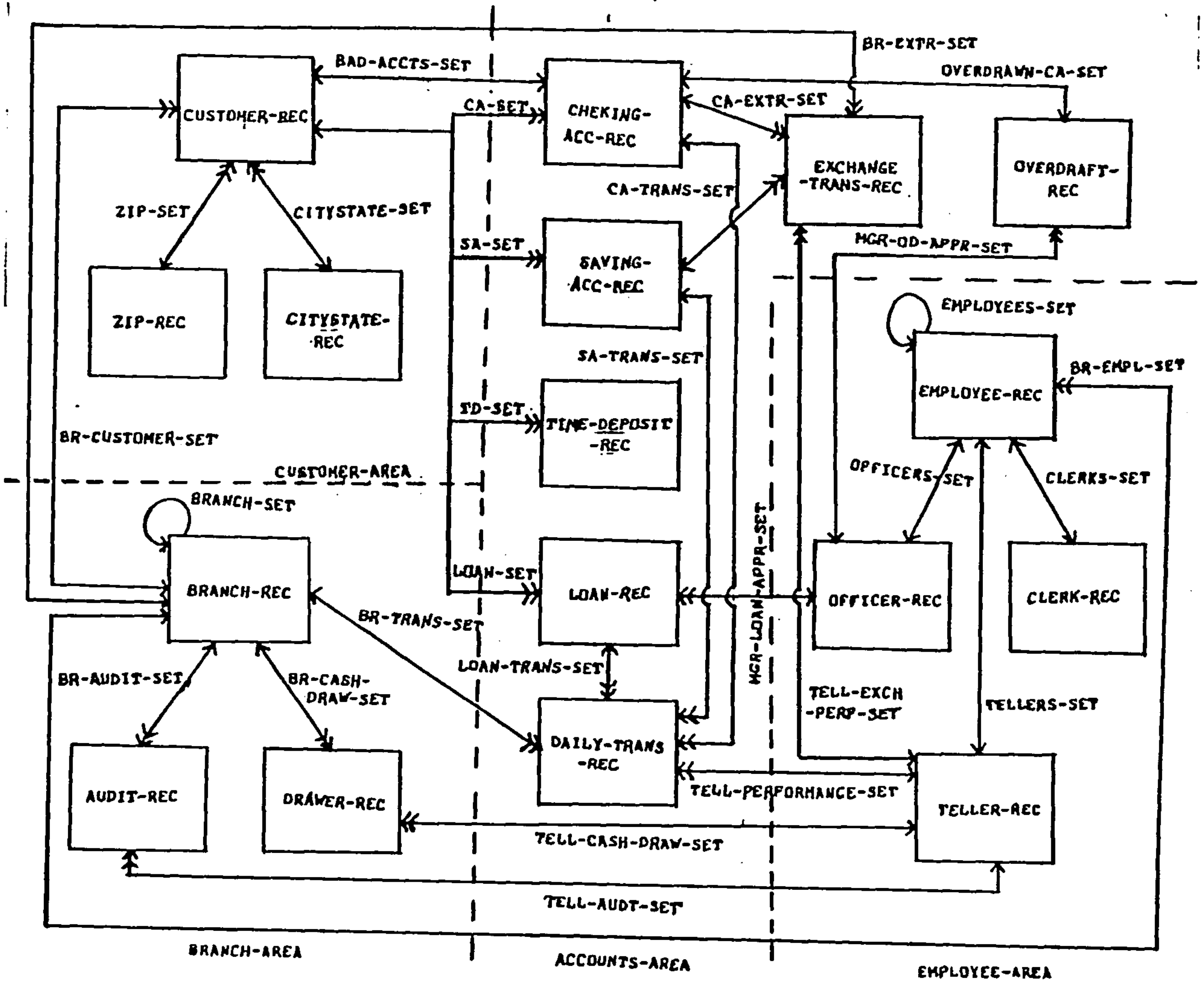
RECORD TYPE	RELATIONSHIPS
AUDT	BR <----->> AUDT
	TELL <----->> AUDT
BR	BR <----->> AUDT
	BR <----->> DRAW
	BR <----->> CUST
	BR <----->> EMPL
	BR <----->> DTRN
	BR <----->> EXTR
	SYSTEM ------> BR
CA	CA <----->> DTRN
	CA <----->> EXTR
	CA <------> DD
	CUST <------> CA
	CUST <----->> CA
CYST	CYST <----->> CUST
CLRK	EMPL <------> CLRK
CUST	CUST <----->> CA
	CUST <------> CA
	CUST <----->> SA
	CUST <----->> TD
	CUST <----->> LOAN
	CYST <----->> CUST
	ZP <----->> CUST
	BR <----->> CUST
DTRN	CA <----->> DTRN
	SA <----->> DTRN
	BR <----->> DTRN
	TELL <----->> DTRN
	LOAN <----->> DTRN
DRAW	BR <----->> DRAW
	TELL <----->> DRAW
EMPL	EMPL <------> TELL

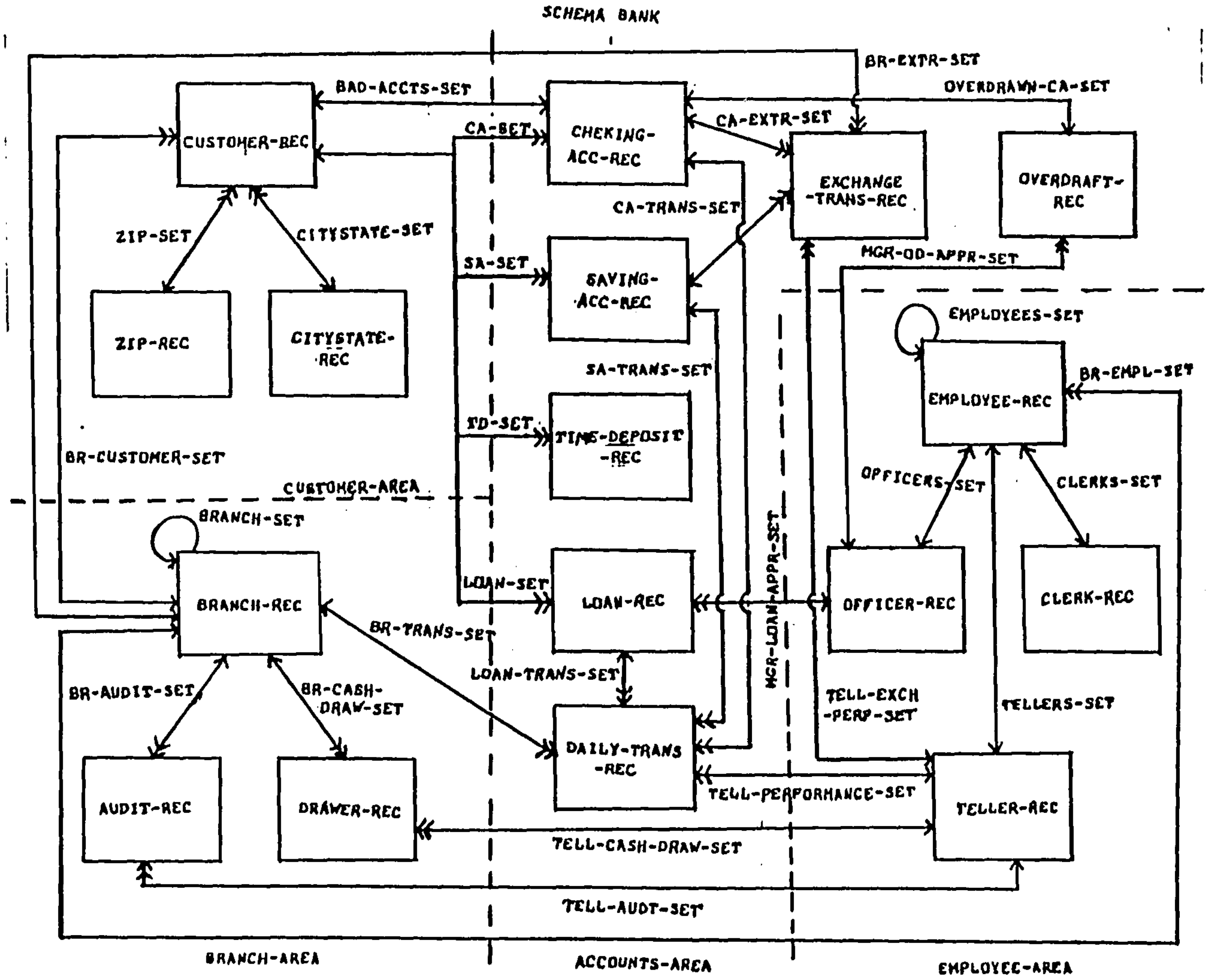
```
                                          EMPL   <------->  OFFR
                                          EMPL   <------->  CLRK
                                          BR     <------>>  EMPL
                                         SYSTEM   ------->  EMPL

EXTR     ..................               CA     <------>>  EXTR
                                          SA     <------>>  EXTR
                                          BR     <------>>  EXTR
                                          TELL   <------>>  EXTR

LOAN     ..................               CUST   <------>>  LOAN
                                          OFFR   <------>>  LOAN
                                          LOAN   <------>>  DTRN

OFFR     ..................               OFFR   <------>>  LOAN
                                          EMPL   <------->  OFFR
                                          OFFR   <------>>  OD

OD       ..................               CA     <------->  OD
                                          OFFR   <------>>  OD

SA       ..................               SA     <------>>  DTRN
                                          SA     <------>>  EXTR
                                          CUST   <------>>  SA

TELL     ..................               TELL   <------>>  DTRN
                                          TELL   <------>>  EXTR
                                          TELL   <------>>  AUDT
                                          TELL   <------>>  DRAW
                                          EMPL   <------->  TELL

TD       ..................               CUST   <------>>  TD

ZP       ..................               ZP     <------>>  CUST
```

3.7 CONCEPTUAL SCHEMA

The following attachment is a diagram of the final data base schema. The conceptual schema designed, followed a basic philosophy. Data redundancy was eliminated wherever possible, replacing the data with pointers to the related records. The recommended system is capable of any function of the old system. Also, the new system provides the capability of on-line inquiry and update, as well as meeting the needs of management (see section 2.3).

SCHEMA BANK


BR-EXTR-SET
OVERDRAWN-CA-SET
BAD-ACCTS-SET
CUSTOMER-REC
CA-SET
CHEKING-ACC-REC
CA-EXTR-SET
EXCHANGE -TRANS-REC
OVERDRAFT-REC
CA-TRANS-SET
ZIP-SET
CITYSTATE-SET
SA-SET
SAVING-ACC-REC
MGR-OD-APPR-SET
ZIP-REC
CITYSTATE-REC
EMPLOYEES-SET
SA-TRANS-SET
EMPLOYEE-REC
BR-EMPL-SET
TD-SET
TIME-DEPOSIT -REC
BR-CUSTOMER-SET
CUSTOMER-AREA
OFFICERS-SET
CLERKS-SET
BRANCH-SET
MGR-LOAN-APPR-SET
LOAN-SET
BRANCH-REC
LOAN-REC
OFFICER-REC
CLERK-REC
BR-TRANS-SET
LOAN-TRANS-SET
BR-AUDIT-SET
BR-CASH-DRAW-SET
TELL-EXCH -PERF-SET
TELLERS-SET
DAILY-TRANS -REC
AUDIT-REC
DRAWER-REC
TELL-PERFORMANCE-SET
TELLER-REC
TELL-CASH-DRAW-SET
TELL-AUDT-SET
BRANCH-AREA
ACCOUNTS-AREA
EMPLOYEE-AREA


SCHEMA BANK
BR-EXTR-SET
OVERDRAWN-CA-SET
BAD-ACCTS-SET
CUSTOMER-REC
CHEKING-ACC-REC
CA-SET
CA-EXTR-SET
EXCHANGE-TRANS-REC
OVERDRAFT-REC
CA-TRANS-SET
ZIP-SET
CITYSTATE-SET
SA-SET
SAVING-ACC-REC
MGR-OD-APPR-SET
ZIP-REC
CITYSTATE-REC
EMPLOYEES-SET
SA-TRANS-SET
BR-EMPL-SET
EMPLOYEE-REC
TD-SET
TIME-DEPOSIT-REC
BR-CUSTOMER-SET
OFFICERS-SET
CLERKS-SET
CUSTOMER-AREA
BRANCH-SET
MGR-LOAN-APPR-SET
LOAN-SET
BRANCH-REC
LOAN-REC
OFFICER-REC
CLERK-REC
BR-TRANS-SET
LOAN-TRANS-SET
BR-AUDIT-SET
BR-CASH-DRAW-SET
TELL-EXCH-PERF-SET
TELLERS-SET
DAILY-TRANS-REC
AUDIT-REC
DRAWER-REC
TELL-PERFORMANCE-SET
TELLER-REC
TELL-CASH-DRAW-SET
TELL-AUDT-SET
BRANCH-AREA
ACCOUNTS-AREA
EMPLOYEE-AREA

4. DDL REPRESENTATION OF THE CONCEPTUAL SCHEMA

4.1 EXPLANATIONS AND BACKUP

Keeping in mind the occurrences and volatility of records type in the data base, the storage requirements were calculated. The object schema was then translated into the DDL (Data Description Language).

The DDL was prepared in accordance with the CODASYL Data Base Task Group Report, April 1971 ; and Data Base Management System Administraor's Manual for DECSYSTEM-20, May 1977.

LOCATION_MODE_OF_RECORDS

> The term "CALC" is an abbreviation of "CALCULATION". When a record is to be stored in the data base, its data base key is calculated after concatenating the values in the data items specified after "USING" in the "LOCATION MODE" clause.[6]

In the Data Base Task Group report, CALC

[6] Data Base Structured Techniques For Design, Performance, And Management: by S. ATRE. pg. 238.

implicitly means "HASHING".[7]

1. When random access is required outside a set relationship and usage is frequent and/or record occurrences are low.

2. When record is a member of a set whose owner is in a different area.

3. When record is owner of all sets in which it participates (except a system set).

It has been decided that all the location modes of the records in this schema will be of **CALC** type, because of one or more of the above mentioned reasons, for example:

Because of reason (1), which is mentioned above, CHEKING-ACC-REC record is assigned the CALC LOCATION MODE. The BRANCH-REc record type is assigned CALC LOCATION MODE due to reason (3). And likewise, AUDIT-REC and DRAWER-REC records were assigned CALC LOCATION MODE because of reason (2). DRAWER-REC

[7] Data Base Structured Techniques For Design, Performance, And Management: by S. ATRE. pg. 239.

records.

DUPLICATES

Duplicates are not allowed on any record keys, except for EXCHANGE-TRANS-REC record since the key is currency name .

PRIOR POINTERS

For efficiency of processing - access, deletion, and insertion - prior linkage has been specified for sets, except when volatility of member records is low, or access applications are sequential. For sets in which prior linkage was specified, overhead calculations include prior pointers for each occurrence of member and owner records.

Prior linkage was not specified for the following sets, along with the reason:

- BR-CUSTOMER-SET : access application is sequential.
- BR-TRANS-SET : access application is sequential.
- BR-EXTR-SET : access application is sequen-

tial.

- BR-CASH-DRAWER-SET : low volatility.

- BR-AUDIT-SET : low volatility.

- BR-EMPL-SET : access application is sequential.

- TELL-AUDIT-SET : low volatility.

- TELL-CASH-DRAW-SET : access application is sequential.

- TELL-PERFORMANCE-SET : access application is sequential.

- TELL-EXCH-PERF-SET : access application is sequential.

- MGR-OD-APPR-SET : low volatility.

- MGR-LOAN-APPR-SET : low volatility.

OWNER POINTERS

To facilitate traversing the data base, owner pointers have been specified for most set member records except sets whose owner is system.

No owner link specified for :

- BRANCH-SET : owner is system.

- EMPLOYEE-SET : owner is system.

- BAD-ACCTS-SET : set is one-to-one with prior pointers.

- OVERDRAWN-CA-SET : set is one-to-one with prior pointers.

SET_ORDER

Order is sorted is specified only for sets where sorted reports are required.

Sorted Sets are :

- BR-CUSTOMER-SET : for reports about customers.
- CA-SET : for reports about Checking Accounts.
- SA-SET : for reports about Saving Accounts.
- TD-SET : for reports about Time Deposit Accounts.
- LOAN-SET : for reports about Loan Accounts.
- BRANCH-SET : for reports about Branches.
- EMPLOYEE-SET : for reports about Employees.

Duplicates are not allowed because in member records duplicates are not allowed.

SET_MEMBERSHIP

In most of the sets, member records are specified as AUTOMATIC. In case a member record type is to be inserted in the data base manually, insertion mode MANUAL is employed. Decision of choosing between AUTOMATIC and MANUAL modes is application dependent.

Records with MANUAL (insertion):

- CHEKING-ACC-REC in BAD-ACCTS-SET.
- OVERDRAFT-REC in OVERDRAWN-CA-SET and MGR-OD-APPR-SET.
- DAILY-TRANS-REC in CA-TRANS-SET, SA-TRANS-SET, LOAN-TRANS-SET, BR-TRANSSTE and TELL-PERFORMANCE-SET.
- EXCHANGE-TRANS-REC in CA-EXTR-STE, SA-EXTR-SET, BR-EXTR-SET and TELL-EXTR-PERF-SET.
- AUDIT-REC in TELL-AUDT-SET.
- DRAWER-REC in TELL-CASH-DRAW-SET.
- LOAN-REC in MGR-LOAN-APPR-SET.

OPTIONAL deletion is used for members of sets for which it is desirable to delete member record occurrences without deleting thier owner records.

Record with OPTIONAL (deletion):

- CHEKING-ACC-REC in BAD-ACCTS-SET.
- TIME-DEPOSIT-REC in TD-SET.

- LOAN-REC in LOAN-SET.
- OVERDRAFT-REC in OVERDRAWN-CA-SET.
- DRAWER-REC in TELL-CASH-DRAW-SET.
- AUDIT-REC in TELL-AUDT-SET.

All other member records are specified as MANDATORY(deletion).

SUB-SCHEMA SECTION

> The **SUB-SCHEMA** was first proposed by the Data Base Task Group, for defining views of a conceptual schema that was itself defined using the Data Description Language.[8]

This part of the schema was written after deciding what application programs are needed. The schema **BANK.DDL** contains five sub-schemas which serve five application programs, those subschemas contain only those AREAS, RECORDS, and SETS which are needed in the application programs, the five subschemas are:

[8] Principles of Database Systems: by Jeffery D. Ullman. pg. 8.

```
SUB-SCHEMA NAME IS TRANSCH  :
       Which serves 3rd program.
AREA SECTION.
       COPY ALL AREAS.

RECORD SECTION.
       BRANCH-REC.
       CUSTOMER-REC.
       CHEKING-ACC-REC.
       SAVING-ACC-REC.
       TELLER-REC.
       EMPLOYEE-REC.
       DAILY-TRANS-REC.

SET SECTION.
       CA-SET SA-SET CA-TRANS-SET
       SA-TRANS-SET BR-TRANS-SET
       TELL-PERFORMANCE-SET TELLERS-SET.


SUB-SCHEMA NAME IS UPTRSCH :
       Which serves 5th program.

AREA SECTION.
       COPY ALL AREAS.

RECORD SECTION.
       DAILY-TRANS-REC.
       CHEKING-ACC-REC.
       SAVING-ACC-REC.
       TELLER-REC.
       BRANCH-REC.
       CUSTOMER-REC.

SET SECTION.
       CA-SET SA-SET CA-TRANS-SET SA-TRANS-SET
       TELL-PERFORMANCE-SET BR-TRANS-SET.


SUB-SCHEMA NAME IS ACCSCH :
       Which serves 2nd program.
```

```
AREA SECTION.
        CUSTOMER-AREA.
        ACCOUNTS-AREA.
        BRANCH-AREA.

RECORD SECTION.
        CUSTOMER-REC.
        BRANCH-REC.
        CHEKING-ACC-REC.
        SAVING-ACC-REC.
        TIME-DEPOSIT-REC.
        LOAN-REC.

SET SECTION.
        BR-CUSTOMER-SET CA-SET
        SA-SET TD-SET LOAN-SET.


SUB-SCHEMA NAME IS CUSTSCH :
        Which serves 1st program.

AREA SECTION.
        CUSTOMER-AREA.
        BRANCH-AREA.

RECORD SECTION.
        CUSTOMER-REC.
        ZIP-REC.
        CITYSTATE-REC.
        BRANCH-REC.

SET SECTION.
        CITYSTATE-SET ZIP-SET BR-CUSTOMER-SET.


SUB-SCHEMA NAME IS DDSCH :
        Which serves 4th program.

AREA SECTION.
        BRANCH-AREA.
        CUSTOMER-AREA.
        ACCOUNTS-AREA.
```

```
RECORD SECTION.
       BRANCH-REC.
       CUSTOMER-REC.
       CHEKING-ACC-REC.
       OVERDRAFT-REC.

SET SECTION.
       BRANCH-SET BR-CUSTOMER-SET
       CA-SET OVERDRAWN-CA-SET.
```

4.2 PAGE ORGANIZATION

GROWTH

All records were specified a 10% growth rate on total words. Also extra pages between areas are reserved for additional growth.

WORDS_PER_PAGE_"WPP"

Words per page are chosen to reduce unused total words on all pages using empirical calculations. Calculation was based on words per page (a multiple of 512) less 2 words for a line and a page header and n words for n calc chain headers.

AREA	WPP
BRANCH-AREA	512.
CUSTOMER-AREA	4096.
ACCOUNTS-AREA	4096.
EMPLOYEE-AREA	512.

CHAIN_HEADERS

Assigned to pages containing CALC records based on records per page and access frequency. Frequently accessed records are calced at 10-15 records per calc chain.

PAGES

Number of pages are calculated from records occurrences including growth, divided by calculated records per page.

Because of the space limitation in the directory used for this thesis, the page organization is not based on calculations, instead (10) pages have been assigned for each area, to test the data and the conceptual schema, to see whether it is working or not.

4.3 RECORDS FREQUANCY AND VOLATILITY

USAGE_REQUIREMENTS

The following table indicates the user's view of the data base usage. The prefix number is the user's view, and the statement that follows is frequency of usage. The prefix numbers are outlined in section 2.3.

1. 1 per week per branch.
2. 1 per week per branch.
3. 1 per week per branch.
4. 1000 per day.
5. 1 per day per branch.
6. 1 per day per branch.
7. 1 per day per branch.
8. 1 per day per branch.
9. 1 per day per branch.
10. 1000 per day.
11. 1 per day.
12. 2 per month.
13. 1 per month.
14. 1 per month.

15. 1 per month.

16. 1 per month.

17. 1 per month.

18. 1 per day.

RECORD_VOLUME

The following table indicates the number of occurrences of a particular record type in the data base. The table is used for calculating the current storage requirements for the data base.

RECORD TYPE	NUMBER OF OCCURRENCES
AUDIT-REC	3000
BRANCH-REC	100
CHEKING-ACC-REC	200,000
CITYSTATE-REC	50
CLERK-REC	450
CUSTOMER-REC	200,000
DAILY-TRANS-REC	100,000
DRAWER-REC	1000
EMPLOYEE-REC	2000
EXCHANGE-TRANS-REC	5000
LOAN-REC	100,000
OFFICER-REC	50
OVERDRAFT-REC	20,000
SAVING-ACC-REC	200,000
TELLER-REC	1500
TIME-DEPOSIT-REC	50,000
ZIP-REC	150

5. LOADING THE DATA BASE

5.1 PROCEDURE

Starting from the schema BANK.DDL, and the nine data files of the old system , a COBOL program is written to load the data base.

The following is a listing of the nine data files, along with the names in which they appear in the directory :

1. Checking, Saving, Time Deposit Accounts file : ACCT.DAT.
2. Audit file : AUDIT.DAT.
3. Branches file : BRANCH.DAT.
4. Cash Drawer file : DRAWER.DAT.
5. Daily Transaction file : DTRANS.DAT.
6. Exchange Transaction file : EXTRAN.DAT.
7. Loans Accounts file : LOANS.DAT.
8. Overdraft Accounts file : ODACCT.DAT.
9. Payroll file : PAYROL.DAT.

These files have been created by an interactive COBOL program called DATA.CBL. A listing of Data Files is in the APPENDIX.

5.2 DATA BASE LOADING PROGRAM

The loading program is written in ANSI standard COBOL and it is called LOAD.CBL (a listing of the program can be found in the Appendix).

The data base was loaded according to algorithm shown in the flow charts listed in APPENDIX II.

In the loading process, those records who did not have an owner were loaded first, then their immediate members, and so on down to the records which have no members.

The program was written in a modular form for ease of maintenance.

- All the paragraphs prefixed by an 'A' are main logic paragraphs which control the opening, loading, and closing the data base.

- The 'B' paragraphs control loading the BRANCH-REC record.

- The 'C' paragraphs control the loading of EMPLOYEE-AREA.

- The 'D' paragraphs control the partial loading of the CUSTOMER-AREA (CUSTOMER-REC, CITYSTATE-REC, ZIP-REC) and CHEKING-ACC-REC, SAVING-ACC-REC and TIME-DEPOSIT-REC from the

ACCOUNTS-AREA.

- The 'E' paragraphs control the completion of loading the CUSTOMER-AREA and another part of the ACCOUNTS-AREA (LOAN-REC).
- The 'F' paragraphs control the loading of AUDIT-REC record.
- The 'G' paragraphs control the loading of DRAWER-REC record.
- The 'H' paragraphs control the loading of OVERDRAFT-REC record.
- The 'I' paragraphs control the loading of DAILY-TRANS-REC record.
- The 'J' paragraphs control the loading of EXCHANGE-TRANS-REC record.
- The 'X' paragraph handles the initialization of all the counters and end of files used in the program.
- The 'Z' paragraph handles any error codes which occured during the loading process.

After successfully performing each one of the above mentioned paragraphs a message is displayed giving the number of stored records, number of inserted records in sets which require insertion, and finally whether the area has been partially or completely loaded.

In loading the BRANCH-REC record, as each record from the BRANCHES file was read, a FIND is issued to determine whether or not the particular record exists in the data base. If not, a STORE is issued to add the record to the data base, and to establish its currency.

In loading the EMPLOYEE-AREA, the PAYROLL file was stored with Employee Social Security Number as a key. For each record in the payroll file, a FIND is issued to search the data base and find the particular BRANCH-REC record in which an employee is working to establish the currency for BRANCH-REC record. If not found, this is an error, otherwise an EMPLOYEE-REC record occurrence is stored if it is unique. A check is performed here on the EMPLOYEE-STATUS, where :

- T : means that the employee belongs to TELLERS category, and a STORE is issued to store TELLER-REC record in TELLERS-SET.
- O : means that the employee belongs to OFFICERS category, and a STORE is issued to store OFFICER-REC record in OFFICERS-SET.
- C : means that the employee belongs to CLERKS category, and a STORE is issued to store CLERK-REC record in CLERKS-SET.

For those three categories the key is EMPLOYEE-

NUMBER, which is unique if any duplication happens the duplicate record will not be stored.

In loading CUSTOMER-AREA completely and part of ACCOUNTS-AREA, the records from ACCOUNTS file and LOANS file were stored in the data base. From ACCOUNTS file each record is read and the data was moved to the corresponding fields of the records in the data base. A FIND is issued then to establish the occurrence of CITYSTATE-REC record and ZIP-REC record. If not found the records are stored, otherwise no records are stored since the record already exists in the data base. Since the BRANCH-REC records already exist in the data base, a FIND is issued to establish the occurrence of this record. If not found, then error has occurred, else a FIND is issued for establishing the occurrence of CUSTOMER-REC record. If it is not found, then the given record is stored.

Accounts types are: CHECKING, SAVING, and TIME DEPOSIT. The record is stored after checking its type. For the sake of simplicity the three types mentioned above are coded as integers: 1, 2, and 3 respectively.

The same procedure used in finding and storing is followed when the LOANS file is accessed. By completing this file, the CUSTOMER-AREA now is completely loaded, and the ACCOUNTS-AREA is partially loaded. One additional thing is performed here, that is, finding the OFFICER-REC record to establish the occurrence of this record and then, if found, INSERT LOAN-REC into MGR-LOAN-APPR-SET set (Manager Loan Approval set).

For loading AUDIT-REC, and DRAWER-REC and OVERDRAFT-REC records, three files were accessed : AUDIT file, CASH DRAWER file and OVERDRAFT ACCOUNTS file. For the first two record types, a FIND is issued to establish the occurrence of a BRANCH-REC record and TELLER-REC record. If found, the corresponding records are stored and/or inserted in the data base according to the nature of the set from the point of insertion/deletion.

For the third record type (OVERDRAFT-REC record) a FIND is issued to establish the occurrence of CHECKING-ACC-REC record and OFFICER-REC record. If found, a STORE is issued to store the record in the data base, and an insertion of the record in MGR-OD-APPR-SET set will take place. By loading these three records the BRANCH-AREA now is completely loaded, and the ACCOUNTS-AREA is partially loaded.

Upon completion of the loading of ACCOUNTS-AREA, the data base will be completely loaded. In order to do this, the last two files DAILY TRANSACTIONS and EXCHANGE TRANSACTIONS are read. The same procedure in finding, storing, and inserting is followed. BRANCH-REC, TELLER-REC, CHECKING-ACC-REC, SAVING-ACC-REC and finally LOAN-REC (LOAN-REC only in case of DAILY-TRANS-REC) are stored/inserted.

If these two records were successfully loaded without a fatal error, the ACCOUNTS-AREA is completely loaded, the program ends, and the data base has been loaded.

5.3 ERROR CHECKING

The system error flag is checked after every data base imperative is issued. If an expected error does not occur (i.e., NOT FOUND for FIND or DUPLICATE RECORD for STORE), an abnormal termination occurs and control is given to Z100-DBMS-STATUS. An abort message is displayed by this section, along with the error code and the SET, RECORD, and AREA in which the error occured. The LAST-GROUP-AREA is also displayed.

Since journaling is not available in the schema BANK, the loading program had to report some kind of status if the load aborted so that the load could be restarted and continued. The error routine gives the programmer enough information to continue the file loading at the point of termination.

Note: It is recommended that the data base be backed up to magnetic tape, disk, or diskette as soon as it is loaded and before any transactions have occurred.

5.4 DATA TRACE

The data base has been created in such a way that when loading is finished it will be clear whether the data has been successfully loaded or not. For example, after the ACCOUNTS FILE is successfully loaded the following information is displayed on the screen:

- Number of CITY-STATES 11.
- Number of ZIP-CODE 12.
- Number of CUSTOMERS 14.
- Number of CHECKING ACC 12.
- Number of SAVING ACC 5.
- Number of TIME DEPOSIT ACC 3.
- Number of BAD ACCOUNTS 4.

Similer messages are displayed after each file is read and stored in the data base. This indicates that the loading was successful and accurate.

6. APPLICATION PROGRAMS

6.1 INTRODUCTION

> The interface between the application programs and the data base management system, refered to as the **Data Manipulation Language** "DML", is imbeded in a host language such as COBOL. It is desirable that it should have a syntax compatible with the host language, because tne application has a host language and the data manipulation language statements are intimately mixed.[9]

In this chapter a brief explanation is given, to explain the application programs written to retrieve the data stored in the data base, and/or update the data base. These programs are examples of many other application programs that could be written. The following is an outline of the application programs:

1. The first program is designed to retrieve the customer's personal information.

2. The second one, is for the retrieval of the

----

[9] Computer Data-Base Organization: by James Martin. pg. 92.

data from all types of accounts.

3. The third one, is to retrieve data from checking accounts and saving accounts transactions. The programs 1, 2, 3 and 4 are of the on-line interactive type. The user has to interact with them to retrieve the data he/she needs.

4. The next program, is an example of data retrieval with report generation. The data retrieved by this program is the overdrawn checking accounts data. The result of the program is displayed on the terminal's screen, as well as stored in a disk file.

5. Finally the last program is designed to carry out new (withdrawal/deposit) transactions for saving/checking accounts. The transactions are carried out, interactively, one at a time and the data base is updated accordingly.

6.2 PROGRAMS DESCRIPTION

What follows is a brief description of the logic of the application programs written for the data base retrieval and updating.

FIRST PROGRAM

This program interactively answers the user queries about customers. The program logic is explained below:

STEP 1 User enters customer's Social Security Number.

STEP 2 Program searches the data base with th Social Security Number as key, and retrieves the relevant customer record. If this search is successful then customer's personal information such as his/her name, street address, city address and zip code are displayed.

STEP 3 If user is intrested to know the branch of this particular customer he/she should enter "Y" for "YES". Otherwise enter "N" for "NO".

STEP 4 To continue interacting with the data base for similar queries about other customers, the user should enter the letter "C". Otherwise he/she should enter "Q".

STEP 5 If user enters "C" in STEP 4 go to STEP 1. If user enters "Q" in STEP 4 return to the main program.

SECOND PROGRAM

This program is designed to retrieve data from all types of accounts available in the data base. These accounts are stored in the data base in the following records:

1. CHEKING-ACC-REC record type : For checking accounts.
2. SAVING-ACC-REC record type : For saving accounts.
3. TIME-DEPOSIT-REC record type : For time deposit accounts.
4. LOAN-REC record type : For loan accounts.

The program logic is explained below:

STEP 1 Enter one of the following account types:

- CA for Checking Accounts.
- SA for Saving Accounts.
- TD for Time Deposit Accounts.
- LA for Loan Accounts.

STEP 2 Enter relevant account number.
STEP 3 If account number is correct, invoke

the procedure neccessary to retrieve information about that account (for each type of accounts there is a seperate procedure). Display the information on the screen. If account number is not correct, display error message.

STEP 4 To continue interacting with the system for similar queries about the same or other accounts enter "C".

STEP 5 If character entered in STEP 4 is "C" go to STEP 1.

STEP 6 If character entered in STEP 4 is "Q" return to the main program.

STEP 7 If character entered is not "C" or "Q" report an error, and go to STEP 4.

THIRD PROGRAM

This program answers queries about daily transactions related to checking or saving accounts, by retrieving the data stored in the data base in DAILY-TRANS-REC, CHECKING-ACC-REC, SAVING-ACC-REC, BRANCH-REC, TELLER-REC, CUSTOMER-REC record types.

The program logic is explained below:

STEP 1 Enter transaction key (this key is the concatenation of date and time the transaction was carried out).

STEP 2 Search the data base with key entered in STEP 1. If search is successfull display information regerding this transaction. If the key is wrong report error.

STEP 3 To continue interacting with the system enter character "C". Otherwise enter "Q".

STEP 4 If the character entered in STEP 3 is "C" go to STEP 1.

STEP 5 IF the character entered in STEP 3 is "Q" return to main program.

STEP 6 If character entered in STEP 3 is other than "C" or "Q" report error, and go to STEP 3.

FORTH PROGRAM

The records retrieved by this program are : OVERDRAFT-REC, CHEKING-ACC-REC, BRANCH-REC and CUSTOMER-REC record types. The output generated by this program is a report that contains information about overdrawn accounts. This report is useful for the management.

The following algorithm was followed in the forth program:

STEP 1 Open output file.

STEP 2 Print the date and time on wich the report is generated, both on the screen and the output file.

STEP 3 Find the first branch record in the data base.

STEP 4 Within this branch, search for those customers who have checking accounts.

STEP 5 Search for those accounts who have overdraft.

STEP 6 If no information obtained in STEPs 3, 4 and 5 go to STEP 8.

STEP 7 Print Branch name and number, Customer name and account number, and the overdraft account status, on the screen as well as on the output file.

STEP 8 Find next branch record in the data base.

STEP 9 If next branch record found go to STEP 3.

STEP 10 Return to main program.

FIFTH PROGRAM

This is the last example of application programs. It can be used in updating the data base. To be more specific, this program is used to carry out new transactions of checking accounts and/or saving accounts. Updating means that new records may be added to the already existing ones, in this case DAILY-TRANS-REC records are added to the data base, and also some other existing records may be modified, in this case CHEKING-ACC-REC or SAVING-ACC-REC records depending on which one is involved in the transaction.

The following algorithm was followed in the fifth program:

STEP 1 Enter account number.

STEP 2 Search the data base for saving/checking account entered in STEP 1.

STEP 3 Enter transaction type (Deposit/Withdrawal).

STEP 4 Enter Branch number and Teller number.

STEP 5 If transaction type is deposit. Enter amount to be deposited.

STEP 6 If transaction tye is withdrawal. Display account's balance and enter amount to be withdrawn.

STEP 7 Compose transaction key using date and time from internal clock of the machine.

STEP 8 Store the new transaction record in the data base. Link the pointers to the Branch record and the Teller record.

STEP 9 Update the related account record.

STEP 10 Display customer name, account number, amount involved, balance before and after the transaction.

STEP 11 To continue interacting with the data base enter enter "C". To go back to the main program enter "Q".

STEP 12 If character entered in STEP 11 is "C" go to STEP 1.

STEP 13 If character entered in STEP 11 is "Q" return to main program.

In each one of the five programs, a HELP procedure is provided to help users understand how to interact with the system. After each query, the system asks users to enter an ACTION CODE : which is one of the following :

- < C > : to continue.
- < H > : to get help on the system.
- < Q > : to leave this program and return to the main rogram.

The only program which has no help procedure is the forth program since no data is needed from the user in this program.

All five programs have been connected by one program called <MAIN.CBL>. The execution of the programs is saved in <MAIN.EXE>. So to start interacting with the system user has to do the following :

Enter:

```
        @ RUN MAIN.EXE    <CR>
```

The system is invoked now and the data is ready to be retrieved and/or updated. A listing of the five programs along with MAIN.CBL are listed in the

APPENDIX. They will appear under the following names :

1. ONE.CBL

2. TWO.CBL

3. THREE.CBL

4. FOUR.CBL

5. FIVE.CBL

6. MAIN.CBL

APPENDIX. They will appear under the following names :

1. ONE.CBL
2. TWO.CBL
3. THREE.CBL
4. FOUR.CBL
5. FIVE.CBL
6. MAIN.CBL

7. LIST OF REFRENCES

BOOKS

1. Atre, S., "DATA BASE, Structured techniques for design, performance, and Management". A Wiley-Interscience, New York, N.Y. 1980.

2. Date, C.J., "An Introduction to Database System". Second ed. Addison_wesley Puplishing Co. California 1977.

3. Martin, James, "Computer Data-Base Organization". Second ed. Printice-Hall, Inc. Englewood Cliffs, N.J. 1977.

4. Martin, James, "Principles of Data-Base Management". Printice-Hall, Inc. Englewood Cliffs, N.J. 1976.

5. Martin, James, "Security, Accuracy and Privacy in Computer Systems". Printice-Hall, Inc. Englwood Cliffs, N.J. 1973, Section II.

6. Ullman, Jeffry D., "Principles of Database Systems". Computer Science Press, Inc. Potomac, Maryland 1980.

REPORTS

1. CODASYL Data Base Task Group Report, ACM, New York, London, and Amsterdam. April 1971.

2. CODASYL Systems Committee, "Feature Analulysis of Generalized Data Base Management Systems". Technical Report, ACM, New

York, London, Amsterdam, May 1971.

3. Dean Jr., A.L., "Data Privacy and Integrity Requirements for Online Data Management Systems". Proc. 1971 ACM SIGFIDET workshop on Data Description, Access and Control.

OTHER SOURCES

Data Base Management System Adminstrator's Manual for DECSYSTEM-20, May 1977.

8. APPENDIX

APPENDIX I

The following is a sample of the ouptut obtained from interacting with the system through the application programs.

```
@RUN MAIN.EXE
 YOU NEED HELP TO KNOW THE SYSTEM ENTER <Y/N> => Y

  - ENTER QUERY CODE (1) FOR CUSTOMER'S PERSONAL DATA .

  - ENTER QUERY CODE (2) FOR INFORMATION ON ALL TYPES
    OF ACCOUNTS.

  - ENTER QUERY CODE (3) FOR INFORMATION ON CHECKING
    OR SAVING ACCOUNTS TRANSACTIONS.

  - ENTER QUERY CODE (4) FOR  INFORMATION  ON ALL
    OVERDRAFT ACCOUNTS.

  - ENTER QUERY CODE  (5)  FOR  CARRYING OUT  NEW
    TRANS REGARDING CHEKING/SAVING ACCOUNTS .


  - TO CONTINUE YOUR SESSION WITH THE DATA BASE
    ENTER < YES >.

  - TO EXIT ENTER < NO > .


  AGAIN IF YOU NEED HELP TO KNOW THE SYSTEM
  ENTER <Y/N> => N
```

```
PLEASE ENTER QUERY CODE    =>  1
```

The following is the output from program number 1.

```
 ENTER CUSTOMER SSN  ==>  123456789

(* the system's response *)

 CUSTOMER NAME    ==>  PETER G SAMUEL

 CUSTOMER ADDR    ==>  21 E 4TH ST.
                      SPAZVILLE,PA.     15211
                      TEL # : (215)734-1240

 DO YOU WANT CUST'S BRANCH INFO
 IF YES ENTER  Y  ==>  Y


 BRANCH NAME     ==>   NORTH SPAZVILLE

 BRANCH NUMBER   ==>   10

  FOR INFO ABOUT OTHER CUSTOMERS ENTER C    ==> C

 ENTER CUSTOMER SSN  ==>  201606532

(* the system's response *)

 CUSTOMER NAME    ==>  BOB J SMITH

 CUSTOMER ADDR    ==>  233 CARLTON AV.
                      FRAMITZ,PA.       15210
                      TEL  # : (215)111-5060

 DO YOU WANT CUST'S BRANCH INFO
 IF YES ENTER  Y  ==>  N

  FOR INFO ABOUT OTHER CUSTOMERS ENTER C    ==> C

 ENTER CUSTOMER SSN  ==>  142110020

(* the system response *)
```

```
CUSTOMER NAME    ==>  TED K PANE

CUSTOMER ADDR    ==>  555 WEST 6TH ST.
                     BETHLEHEM,PA.     18018
                     TEL  # : (215)666-7777

DO YOU WANT CUST'S BRANCH INFO
IF YES ENTER  Y  ==>  Y


BRANCH NAME     ==>   BETHLEHEM

BRANCH NUMBER  ==>    120

 FOR INFO ABOUT OTHER CUSTOMERS ENTER C    ==> Q

TO CONTINUE YOUR SESSION ENTER <YES> =>  YES


PLEASE ENTER QUERY CODE   =>  ?
```

The following is the output from program number 2.

```
ENTER CUST ACCOUNT TYPE CODE  ==>   CA


ENTER C/A NO  ==>   016392750


CUSTOMER NAME         ==>  CHERYL L BLINKORF

BRANCH  NAME          ==>  KOUKSTOWN

AVAILABLE BALANCE     ==>  $1494625.98

CURRENT   BALANCE     ==>   $494625.98



ENTER ACTION CODE < C/H/Q >  ==>  H



FIRST enter one of the following account types :
         *  CA  :  For CHECKING ACCOUNTS,
         *  SA  :  For SAVING ACCOUNTS ,
         *  TD  :  For TIME DEPOSIT ACCOUNTS,
         *  LN  :  For LOAN ACCOUNTS.

SECONDLY enter the relevent account number . For
       example if account type is < CA > you will
       see the following :
         *  Customer Name,
         *  Branch Name,
         *  Available Balance,
         *  Current Balance,

THIRDLY enter< C > to get info about other customers
         enter < Q > to quit this query type.
          enter < H > to get help .
```

```
ENTER ACTION CODE  <C/H/O>  ==>  C

ENTER CUST ACCOUNT TYPE CODE  ==>   SA

ENTER S/A NO  ==>   029214309

CUSTOMER NAME     ==>  BARRY Z VINKLE

BRANCH  NAME      ==>  WEST SPAZVILLE

ACCOUNT BALANCE   ==>    $14782.17

INTEREST AMOUNT   ==>     $1478.21


ENTER ACTION CODE < C/H/O >  ==>  C

ENTER CUST ACCOUNT TYPE CODE  ==>   TD

ENTER TD NO  ==>   031617189

CUSTOMER NAME           ==>  CHARLY B BROWN

BRANCH  NAME            ==>  HELLERTOWN

TIME DEPOSIT AMOUNT     ==>    $25000.00

TIME DEPOSIT INTER.     ==>  0.16

TIME DEPOSIT START      ==>  12/03/1981

TIME DEPOSIT DUE        ==>  09/03/1982
```

```
ENTER ACTION CODE < C/H/Q >  ==>  C

ENTER CUST ACCOUNT TYPE CODE  ==>   LN

ENTER LOAN NO  ==>   047293473

CUSTOMER NAME            ==>  STEEVE L KONG

BRANCH  NAME             ==>  LANKESTER

LOAN  AMOUNT             ==>    $15000.00

LOAN  INTEREST RATE      ==>  .20

LOAN START DATE          ==>  02/28/1980

LOAN  DUE  DATE          ==>  02/28/1982

TOTAL AMOUNT DUE         ==>    $21000.00

TOTAL AMOUNT COLLECTED ==>      $15600.00


ENTER ACTION CODE < C/H/Q >  ==>  Q

TO CONTINUE YOUR SESSION ENTER <YES> =>  YES

PLEASE ENTER QUERY CODE   =>  3
```

The following is the output from program number 3:

```
 ENTER DAILY TRANS KEY  ==>  010180142423

    DATE: 01/01/80             TIME: 14:24:23

 BRANCH NAME          ==>  BETHLEHEM

 BRANCH NUMBER        ==>  120

 PERFORMED BY         ==>  SANDRA LEWIS

 TRANSACTION TYPE     ==>  WITHDRAWAL

 ACCOUNT NUMBER       ==>  013245700

 CUSTOMER NAME        ==>  TED K PANE

 TRANSACTION AMOUNT ==>      $1150.00

 ACCOUNT BALANCE      ==>   $11000.00

 ENTER ACTION CODE  < C/H/O >  ==>  H

 Enter Daily Transaction Key
 COMMENT:-
     Daily Transaction Key is 12 digit numeric
  datum which is concatenation of date and
  time at which a transaction is carried out
 END COMMENT
 On entering a Daily Transaction Key following
 information is displayed:-
         * Branch Name & Branch Number
         * Teller,who carried out the Transaction
         * Transaction type, Deposit or Withdrawal
         * Customer name & Account Number
         * Transaction Amount
         * Account Balance
```

```
Enter <C> to continue query about other transactions
Enter <Q> to quit this query type
Enter <H> to get help


ENTER ACTION CODE  <C/H/Q>  ==>  C


ENTER DAILY TRANS KEY  ==>  071882151230


    DATE: 07/18/82           TIME: 15:12:30

BRANCH NAME          ==>  KONKSTOWN

BRANCH NUMBER        ==>  242

PERFORMED BY         ==>  SCOTT PACMAN

TRANSACTION TYPE     ==>  WITHDRAWAL

ACCOUNT NUMBER       ==>  016392750

CUSTOMER NAME        ==>  CHERYL L BLINKORF

TRANSACTION AMOUNT ==>        $124.50

ACCOUNT BALANCE      ==>  $1494625.98

ENTER ACTION CODE  < C/H/Q >  ==>  C

ENTER DAILY TRANS KEY  ==>  11128012025

*** ERROR IN DAILY TRANS KEY ***

ENTER ACTION CODE  < C/H/Q >  ==>  C

ENTER DAILY TRANS KEY  ==>  111280121025

TRANSACTION NOT OF TYPE C/A OR S/A

ENTER ACTION CODE  < C/H/Q >  ==>  Q
```

```
TO CONTINUE YOUR SESSION ENTER <YES> =>  YES

PLEASE ENTER QUERY CODE    =>  1
```

The output from program number 4 is stored in the file **REPORT.DAT** To retrieve the data in this file do the following:

```
@TYPE REPORT.DAT <CR>
```

After finishing writing the output of program number 4 in REPORT.DAT the system continues to interact with user.

```
 TO CONTINUE YOUR SESSION ENTER <YES> =>  YES

 PLEASE ENTER QUERY CODE    =>  5
```

The following is the output from program number 5:

```
 ENTER ACCOUNT NUMBER ==>  022221110


 ENTER  TRANS  TYPE     ==>  1

 ENTER  AMOUNT         ==>  1000.00

 ENTER TELLER NUMBER  ==>  648201

 ENTER BRANCH NUMBER  ==>  240
  *****  DAILY TRANSACTION KEY    ****  111782172955
 (* to check in program 3 if this one exists now *)
 ENTER PASSBOOK LINE NO ==>  110


 CUSTOMER  NAME          ==>  KAREN D SMURF

 ACCOUNT NUMBER          ==>  022221110

 BALANCE BEFORE TRANS  ==>    $16000.00

 ENTERED AMOUNT          ==>     $1000.00

 BALANCE AFTER TRANS   ==>    $17000.00

 ENTER ACTION CODE  < C/H/O >  ==>  C

 ENTER ACCOUNT NUMBER ==>  011357911


 ENTER  TRANS  TYPE     ==>  2

 AVAILABLE BALANCE     ==> $150001.00

 ENTER AMOUNT           ==>  121.50

 ENTER TELLER NUMBER  ==>  141516
```

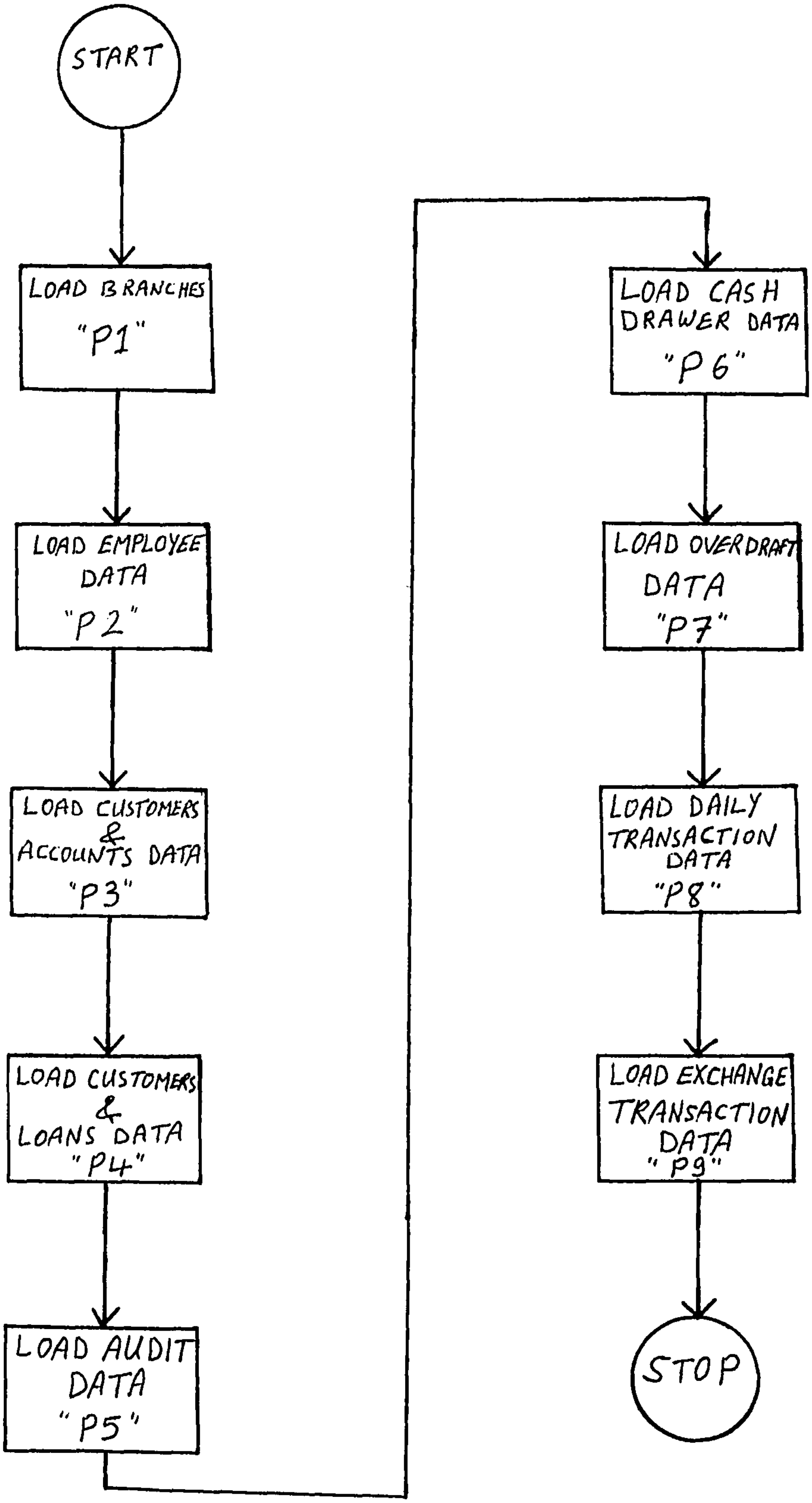
```
ENTER BRANCH NUMBER  ==>  10

 *****  DAILY TRANSACTION KEY   ****  111782173134
ENTER CHECK NUMBER   ==>  123


CUSTOMER  NAME          ==>  PETER G SAMUEL

ACCOUNT NUMBER          ==>  011357911

BALANCE BEFORE TRANS    ==>   $150001.00

ENTERED AMOUNT          ==>      $121.50

BALANCE AFTER TRANS     ==>   $149879.50

ENTER ACTION CODE  < C/H/O >  ==>  O

 TO CONTINUE YOUR SESSION ENTER <YES> =>  YES
(* check if those records were added to the *)
(* data base and/or modified *)

 PLEASE ENTER QUERY CODE   =>  3
```

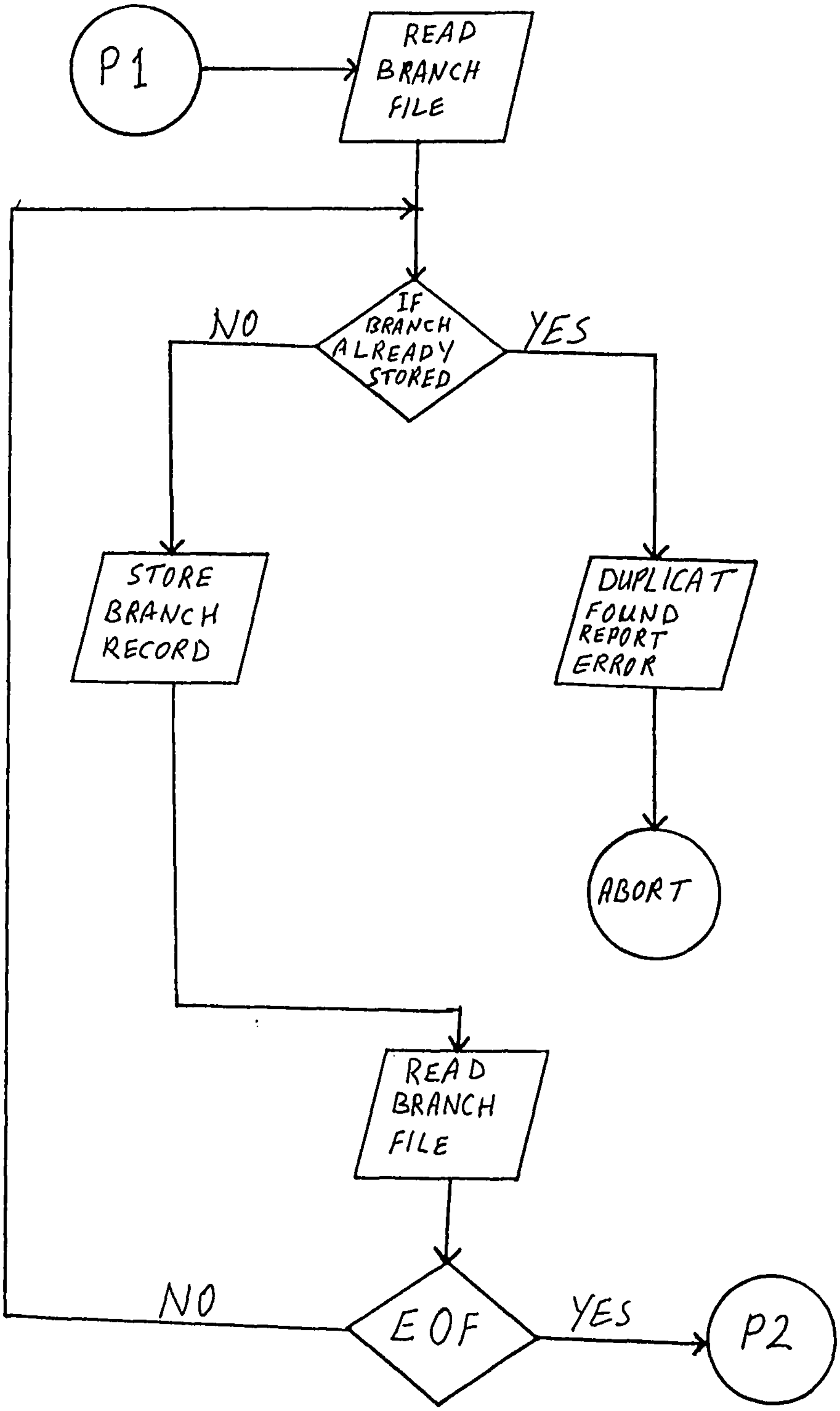
```
ENTER DAILY TRANS KEY  ==>  111782172955

    DATE: 11/17/82            TIME: 17:29:55

BRANCH NAME          ==>  COMP ISLE

BRANCH NUMBER        ==>  240

PERFORMED BY         ==>  JANET WALLY

TRANSACTION TYPE     ==>  DEPOSIT

ACCOUNT NUMBER       ==>  022221110

CUSTOMER NAME        ==>  KAREN D SMURF

TRANSACTION AMOUNT ==>       $1000.00

ACCOUNT BALANCE      ==>    $17000.00

ENTER ACTION CODE  < C/H/O >  ==>  C

ENTER DAILY TRANS KEY  ==>  111782173134

    DATE: 11/17/82            TIME: 17:31:34

BRANCH NAME          ==>  NORTH SPAZVILLE

BRANCH NUMBER        ==>  10

PERFORMED BY         ==>  MATHEW SNYDER

TRANSACTION TYPE     ==>  WITHDRAWAL

ACCOUNT NUMBER       ==>  011357911

CUSTOMER NAME        ==>  PETER G SAMUEL

TRANSACTION AMOUNT ==>        $121.50
```

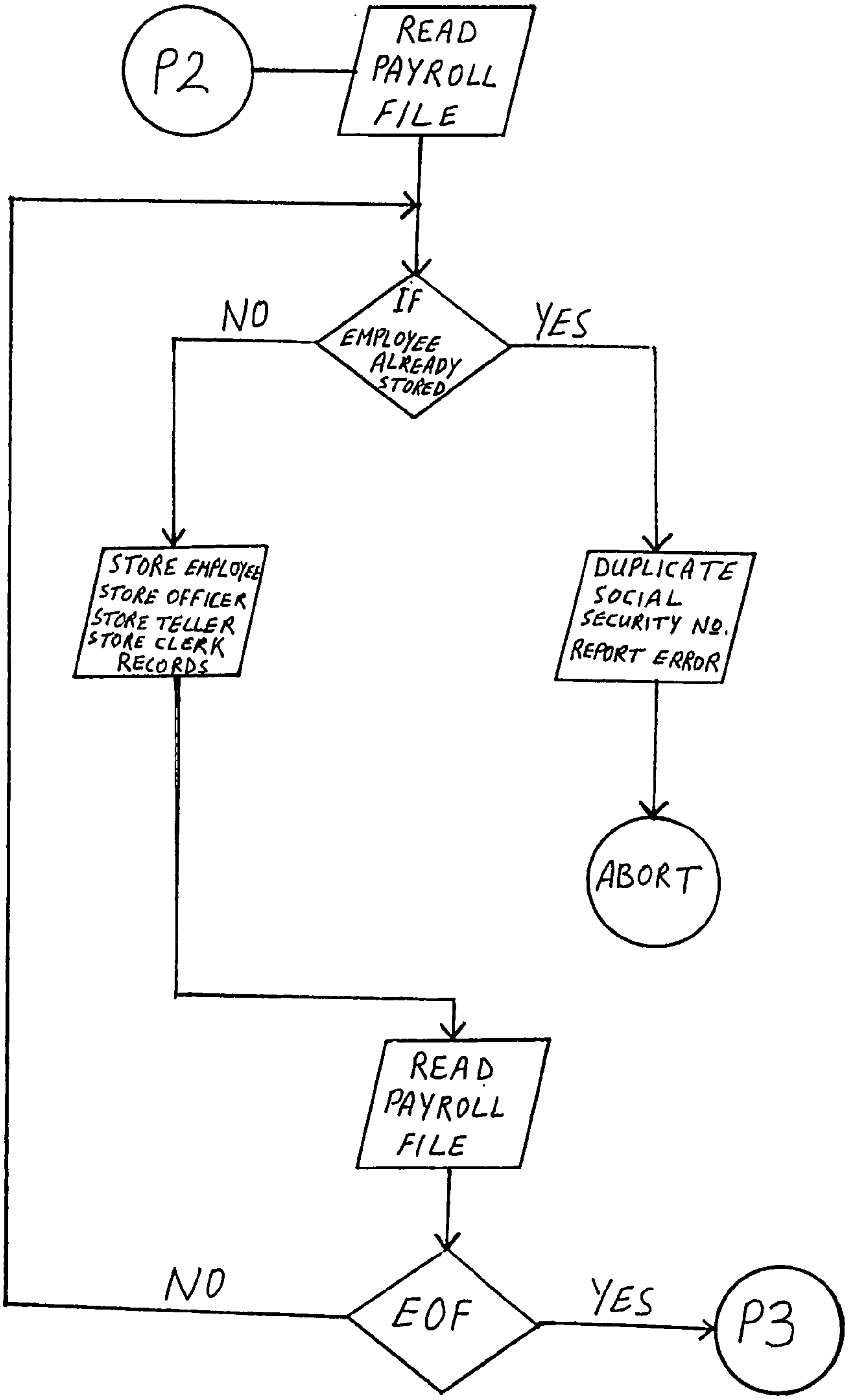
```
 ACCOUNT BALANCE     ==>    $149879.50

 ENTER ACTION CODE  < C/H/Q >  ==>  Q

 TO CONTINUE YOUR SESSION ENTER <YES> =>  NO


EXIT
```

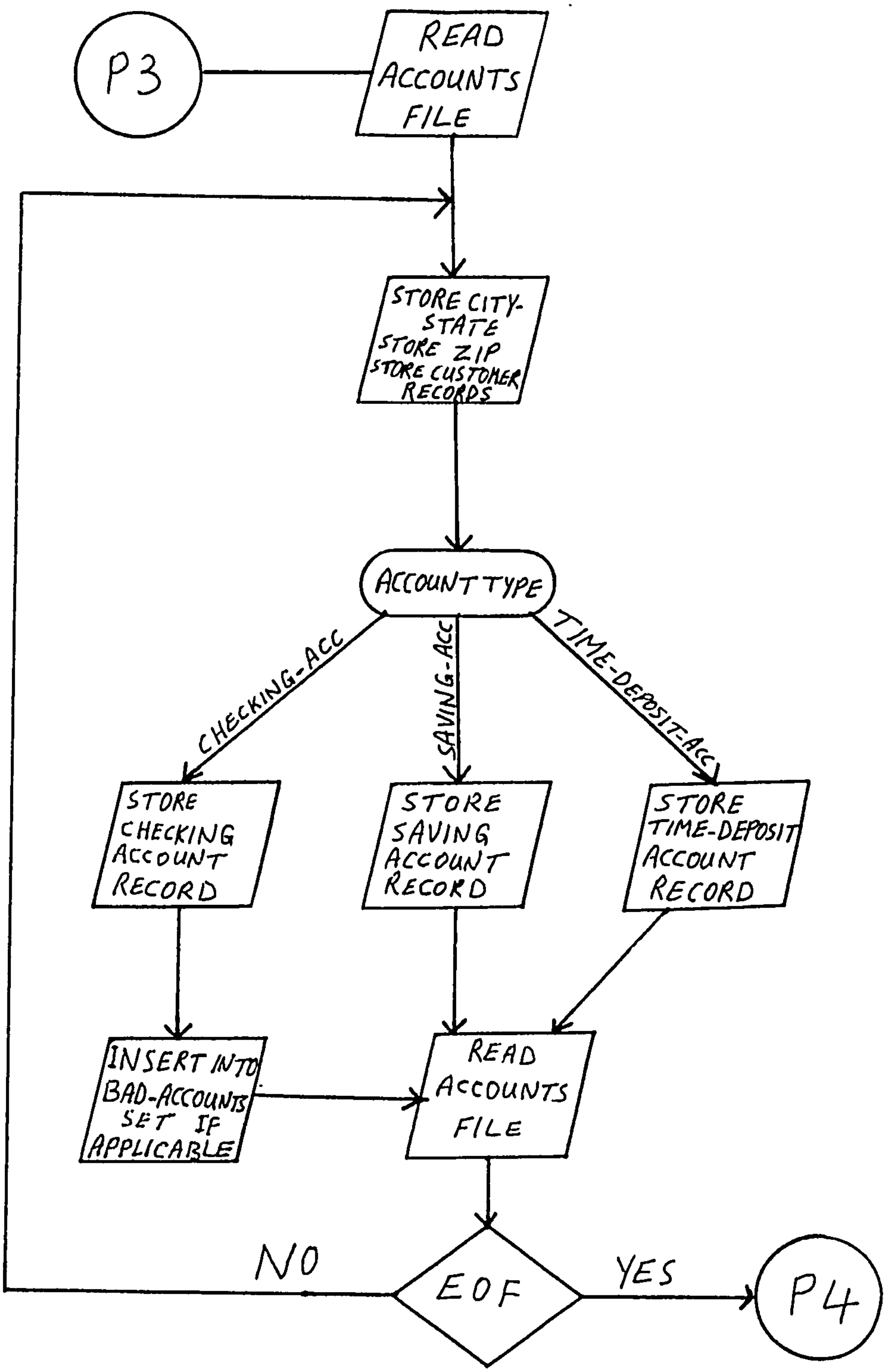
APPENDIX II

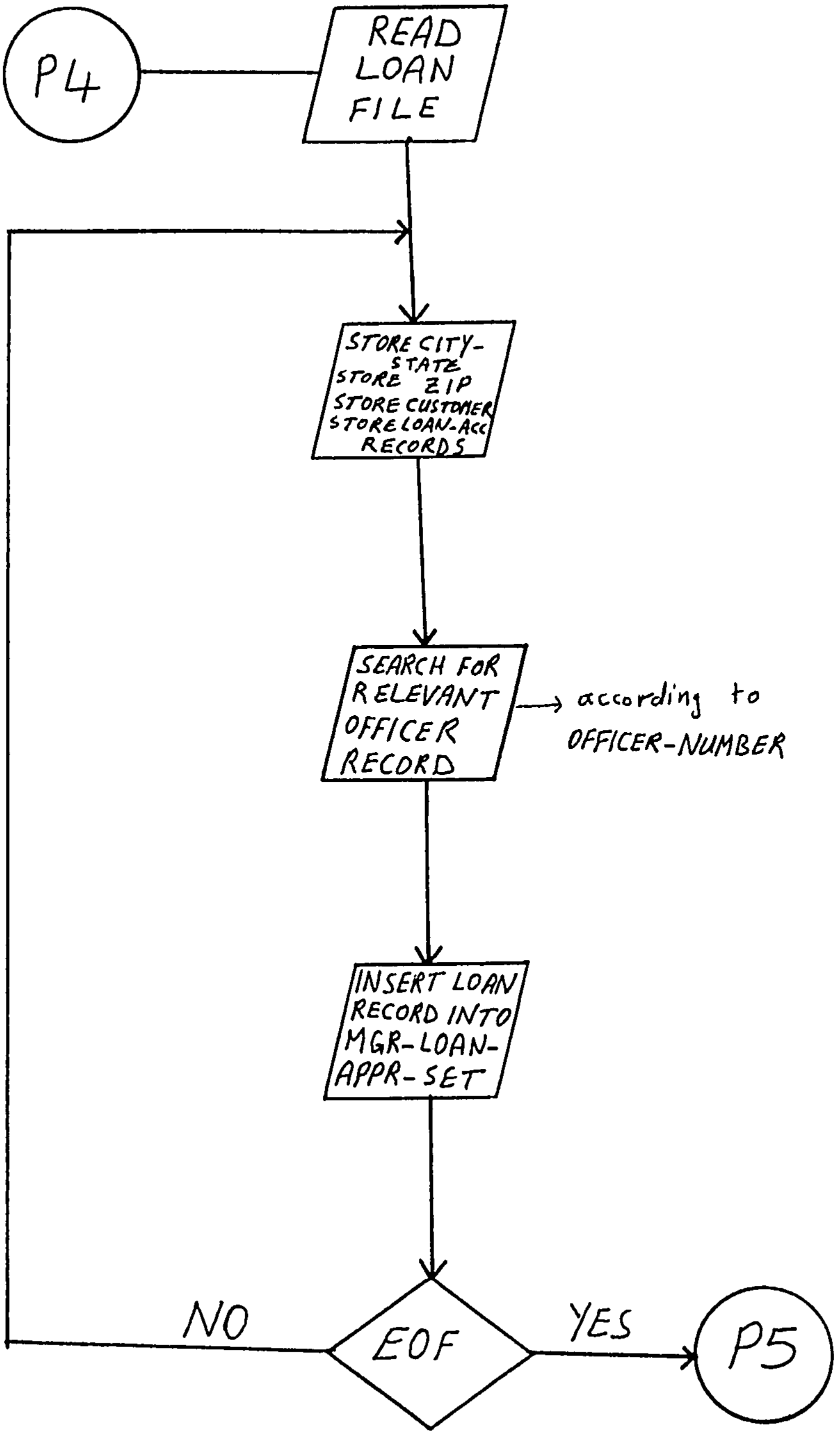
The data base was loaded according to algorithm shown in the following flow charts:

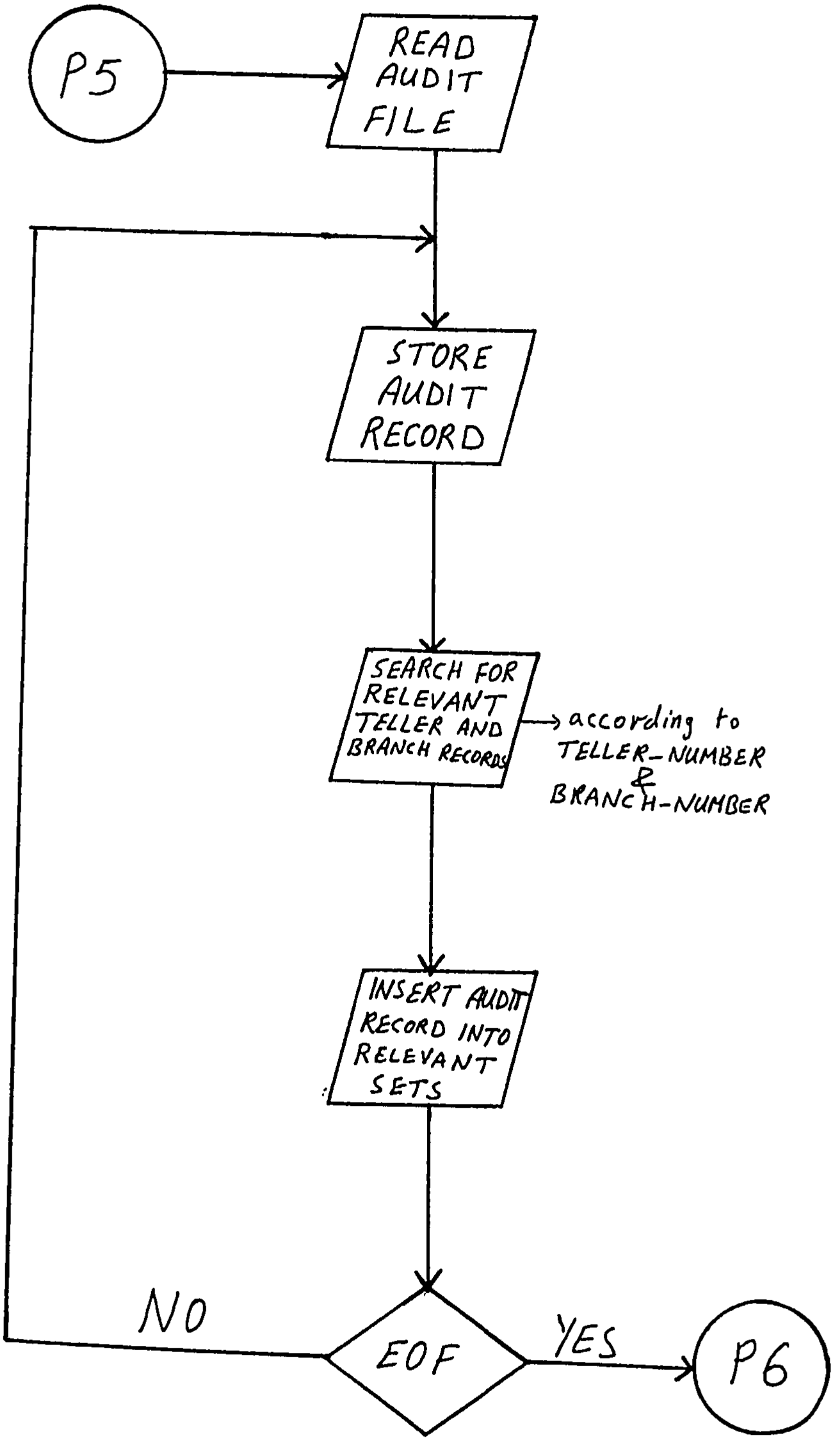

START
LOAD BRANCHES "P1"
LOAD EMPLOYEE DATA "P2"
LOAD CUSTOMERS & ACCOUNTS DATA "P3"
LOAD CUSTOMERS & LOANS DATA "P4"
LOAD AUDIT DATA "P5"
LOAD CASH DRAWER DATA "P6"
LOAD OVERDRAFT DATA "P7"
LOAD DAILY TRANSACTION DATA "P8"
LOAD EXCHANGE TRANSACTION DATA "P9"
STOP

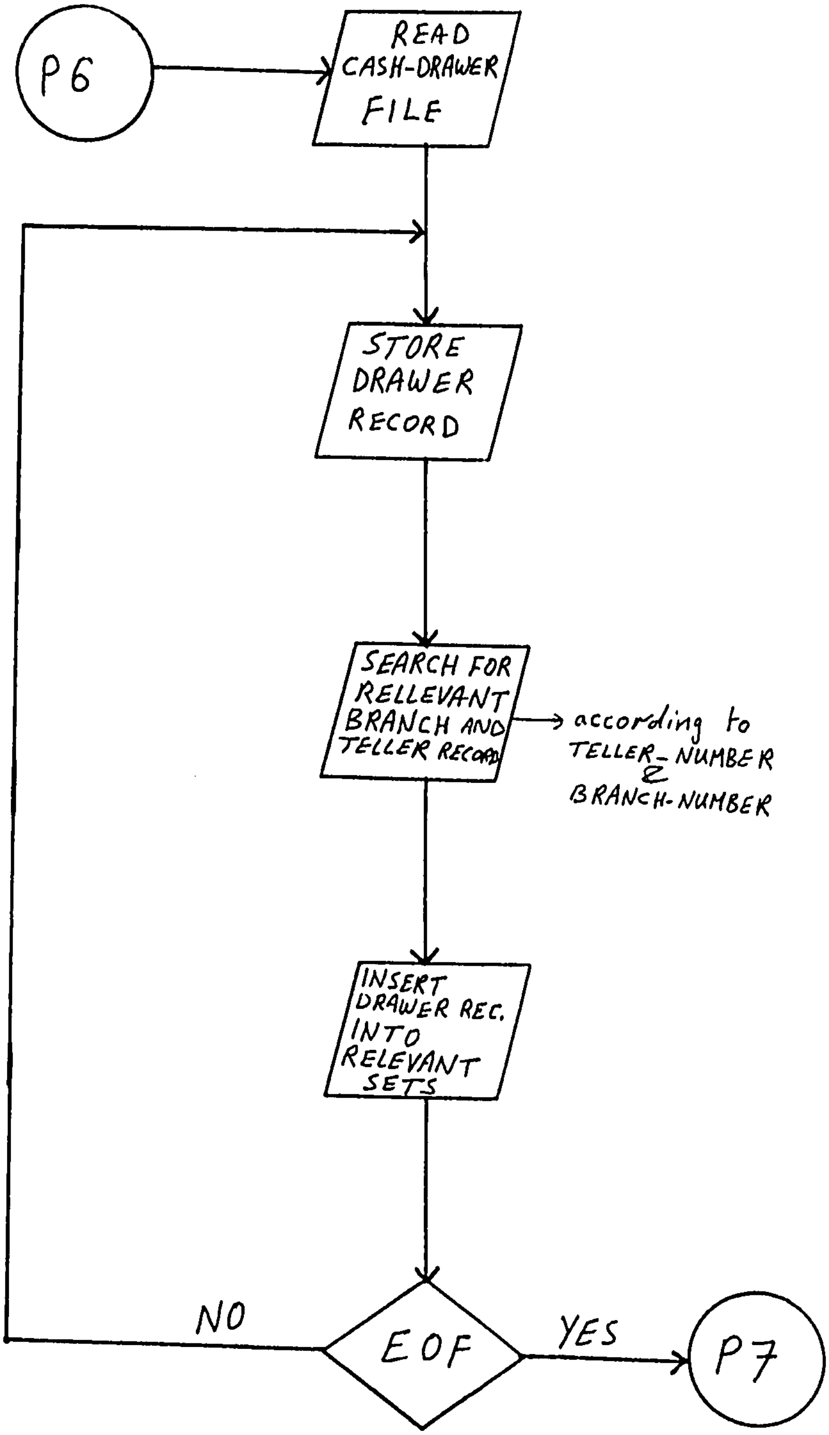

P1
READ BRANCH FILE
IF BRANCH ALREADY STORED
NO
YES
STORE BRANCH RECORD
DUPLICAT FOUND REPORT ERROR
ABORT
READ BRANCH FILE
NO
EOF
YES
P2

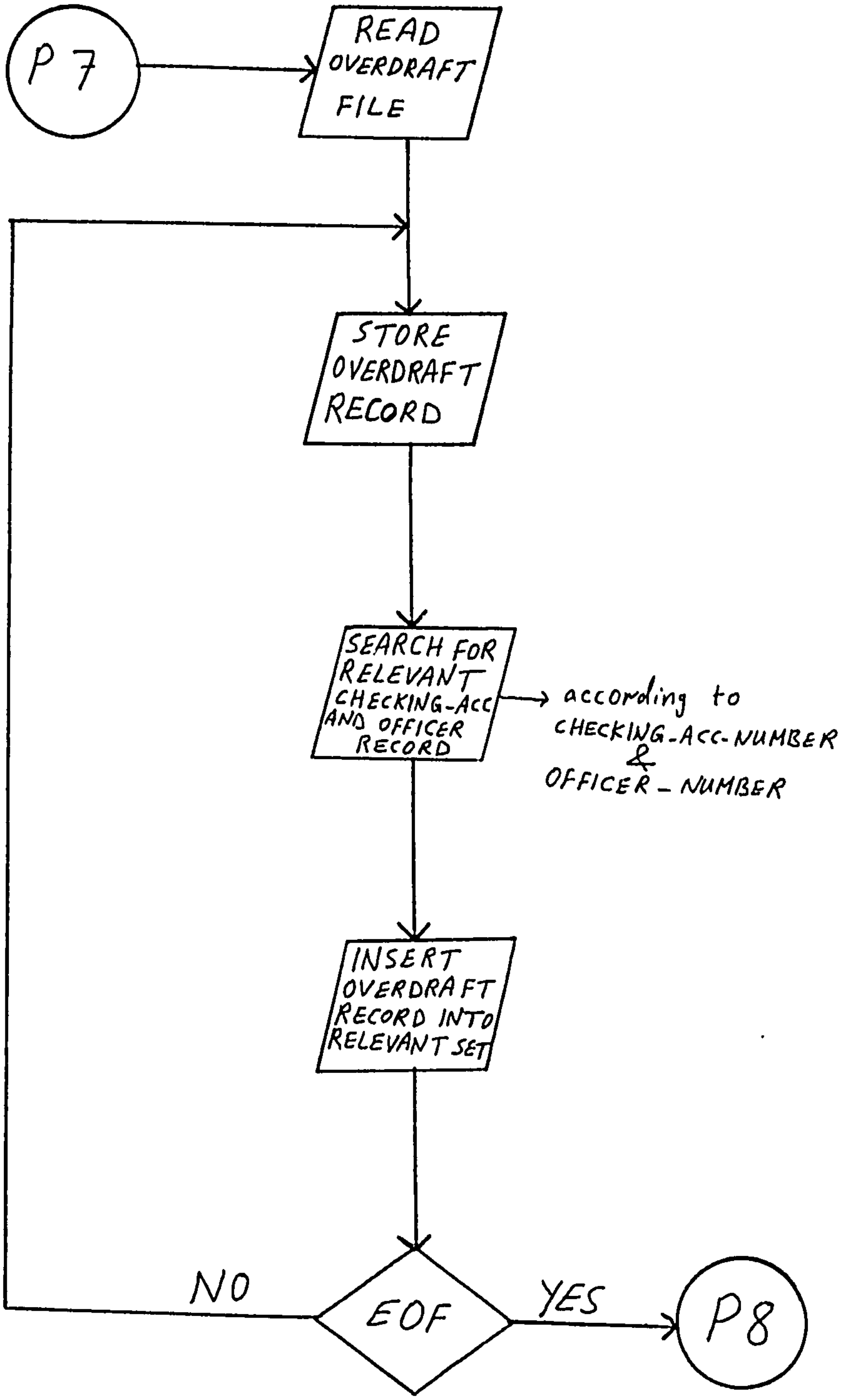

P2
READ PAYROLL FILE
IF EMPLOYEE ALREADY STORED
NO
YES
STORE EMPLOYEE STORE OFFICER STORE TELLER STORE CLERK RECORDS
DUPLICATE SOCIAL SECURITY NO. REPORT ERROR
ABORT
READ PAYROLL FILE
EOF
NO
YES
P3

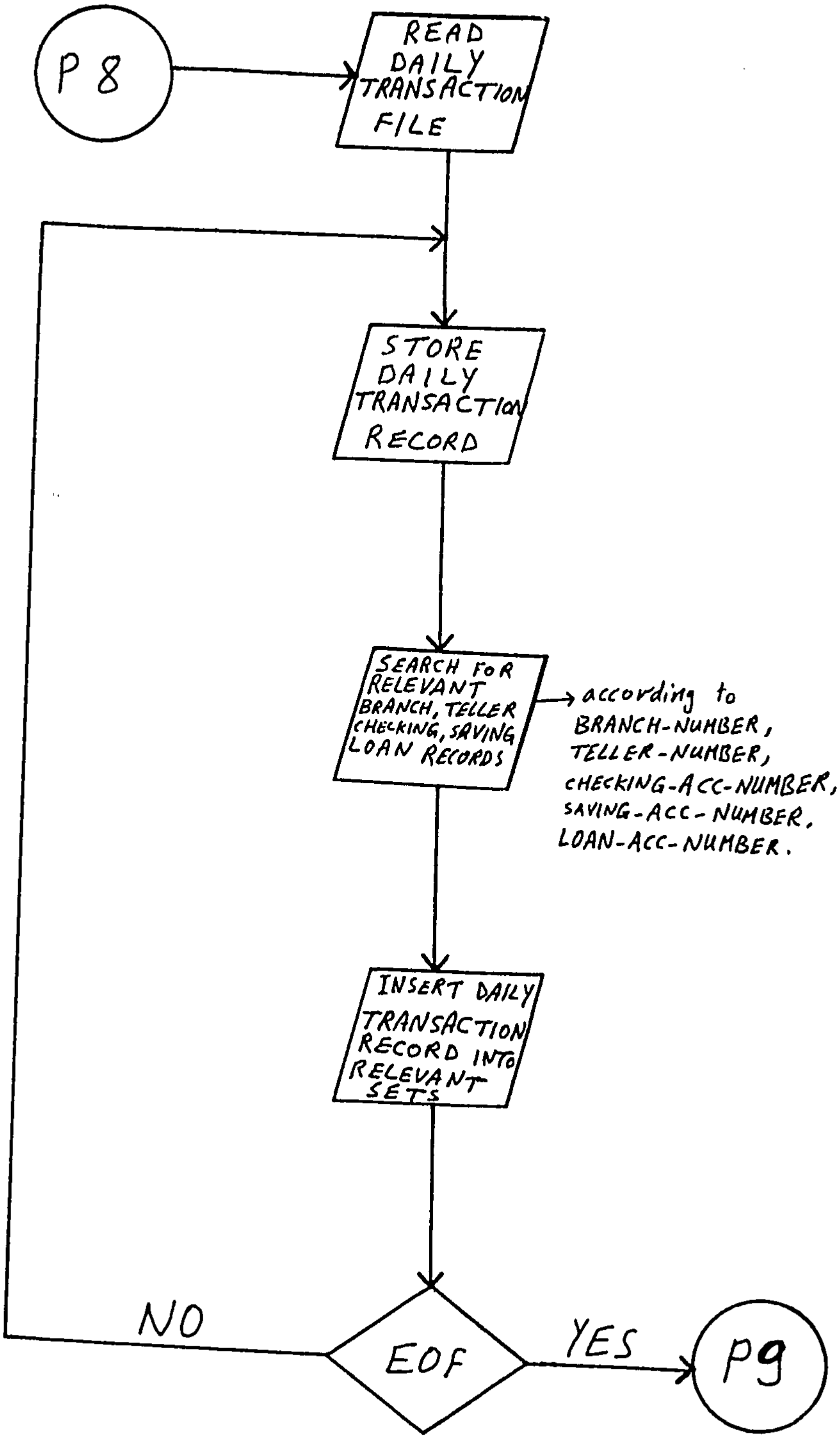

P3
READ ACCOUNTS FILE
STORE CITY-STATE
STORE ZIP
STORE CUSTOMER RECORDS
ACCOUNT TYPE
CHECKING-ACC
SAVING-ACC
TIME-DEPOSIT-ACC
STORE CHECKING ACCOUNT RECORD
STORE SAVING ACCOUNT RECORD
STORE TIME-DEPOSIT ACCOUNT RECORD
INSERT INTO BAD-ACCOUNTS SET IF APPLICABLE
READ ACCOUNTS FILE
NO
EOF
YES
P4

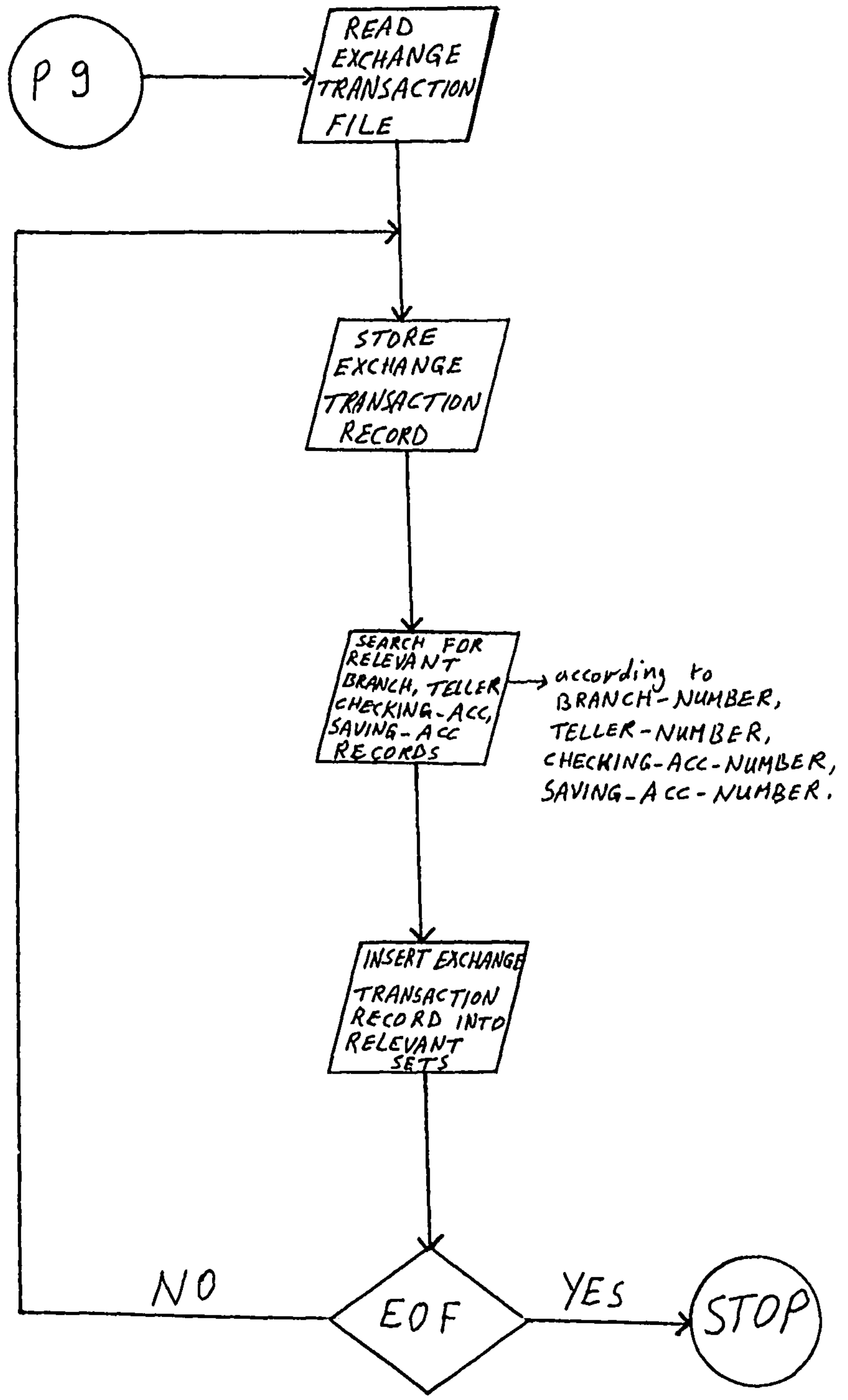

P4
READ LOAN FILE
STORE CITY-STATE
STORE ZIP
STORE CUSTOMER
STORE LOAN-ACC
RECORDS
SEARCH FOR RELEVANT OFFICER RECORD
according to OFFICER-NUMBER
INSERT LOAN RECORD INTO MGR-LOAN-APPR-SET
EOF
NO
YES
P5


P5
READ AUDIT FILE
STORE AUDIT RECORD
SEARCH FOR RELEVANT TELLER AND BRANCH RECORDS
according to TELLER-NUMBER & BRANCH-NUMBER
INSERT AUDIT RECORD INTO RELEVANT SETS
EOF
NO
YES
P6


P6
READ
CASH-DRAWER
FILE
STORE
DRAWER
RECORD
SEARCH FOR
RELLEVANT
BRANCH AND
TELLER RECORD
according to
TELLER_NUMBER
&
BRANCH-NUMBER
INSERT
DRAWER REC.
INTO
RELEVANT
SETS
NO
EOF
YES
P7


P7
READ OVERDRAFT FILE
STORE OVERDRAFT RECORD
SEARCH FOR RELEVANT CHECKING-ACC AND OFFICER RECORD
according to CHECKING-ACC-NUMBER & OFFICER-NUMBER
INSERT OVERDRAFT RECORD INTO RELEVANT SET
EOF
NO
YES
P8


P 8
READ DAILY TRANSACTION FILE
STORE DAILY TRANSACTION RECORD
SEARCH FOR RELEVANT BRANCH, TELLER CHECKING, SAVING LOAN RECORDS
according to BRANCH-NUMBER, TELLER-NUMBER, CHECKING-ACC-NUMBER, SAVING-ACC-NUMBER, LOAN-ACC-NUMBER.
INSERT DAILY TRANSACTION RECORD INTO RELEVANT SETS
EOF
NO
YES
P9


P 9
READ EXCHANGE TRANSACTION FILE
STORE EXCHANGE TRANSACTION RECORD
SEARCH FOR RELEVANT BRANCH, TELLER CHECKING-ACC, SAVING-ACC RECORDS
according to BRANCH-NUMBER, TELLER-NUMBER, CHECKING-ACC-NUMBER, SAVING-ACC-NUMBER.
INSERT EXCHANGE TRANSACTION RECORD INTO RELEVANT SETS
EOF
NO
YES
STOP

9. VITA

The author was born to Mr. and Mrs. Mahmoud M. Yaseen on April 25, 1955 in Jerusalem, Jordan. He earned his B. Sc. degree in Mathematics from University of Jordan (Amman, Jordan) in January 1977. He worked for two banks in Kuwait from April 1977 to August 1980. In the Fall 1980, he began graduate study in Computing Science at Lehigh University. He was a teaching assistant in Mathematics for the Department of Mathematics at Lehigh University.