Kumaraguru College of Technology
Department of Computer Science and Engineering
Coimbatore – 641006.

September 2004

QUERY GENERATION

Project Work done at

POLARIS SOFTWARE LAB LTD

PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the award of the degree of

M.Sc. Applied Science (Software Engineering)
Bharathiar University, Coimbatore

Submitted by

ANU.S.S
Reg. No.: 0137S0022

Internal Guide: Mr.K.R.Baskaran.M.S.,
Dept. of Computer Science & Engineering,
Kumaraguru College of Technology,
Coimbatore.

External Guide: Mr. Hemanth Kumar M.C.A
Project Leader
Polaris Software Lab Ltd
Chennai-600 002

KUMARAGURU COLLEGE OF TECHNOLOGY

(Affiliated to Bharathiar University)

COIMBATORE – 641006

BONAFIDE CERTIFICATE

This is to certify that this is the Bonafide Project Record Work done by **Anu.S.S** Reg.No **0137S0022** in Partial fulfillment for the award of Degree of **M.Sc. [SOFTWARE ENGINEERING]** , during the academic year 2004 –2005.

Prof & HOD

Guide

This Project entitled "Query Generation" is submitted for the VII semester of M.Sc. [SOFTWARE ENGINEERING] for Bharathiar University Project Viva-voce examinations held on ..29.-..9.-2004......

Internal Examiner

External Examiner

'orld's First
MMi-Level 5
ghest Quality in
ware Engineering

POLARIS
power of precision

September 22, 2004

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Ms. Anu.S.S** MSc. Software Engineering from **Kumaraguru College of Technology**, coimbatore has completed the Implant Training **" Citibanking "**. The training duration is from June 2004 to September 2004.

We wish her all the very best in all her future endeavors.

For **POLARIS SOFTWARE LAB LTD**

22-Sep-2004

Mohanapriya. C.V
Executive -Human Resources.

)LARIS SOFTWARE LAB LIMITED, Shakthi Towers, 766 Anna Salai, Chennai - 600 002, INDIA. Ph.: 91 44 2852 5507 Fax: 91 44 2851 6456
3d.Office: POLARIS SOFTWARE LAB LIMITED, Polaris House, 244, Anna Salai, Chennai-600 006. INDIA. Ph.: 91 44 2852 4154 Fax: 91 44 2852 3280
rporate Headquarters: Ph.91 4114 235001
ennai: Polaris House: Ph.91 44 2852 4154 Shakthi Towers: Ph.91 44 2852 5507 Spencers: Ph.91 44 2856 6444 Steeple Reach: Ph.91 44 2811 8900
derabad: Begumpet: 91 40 2340 7500 Lulla Center: 91 40 5538 7500
umbai: SEEPZ/SDF II: 91 22 2829 1299 SEEPZ/SDF V: Ph.91 22 2829 0019 SEEPZ/SDF VI: Ph.91 22 2828 1790 Mahindra Towers: 91 22 5662 8883
liana House: 91 22 2850 7320
w Delhi: Ph.91 124 239 9561
Operations: Chicago: Ph.1 630 799 7127 Fremont: Ph.1 510 746 9986 New Jersey: Ph.1 732 590 8100 Jersey City: Ph.1 732 590 8100
ropean Operations: Dublin: Ph.353 1 634 5068 Frankfurt: Ph.49 0 6969 86160 Ireland: Ph.353 1 287 6745 London: Ph.44 020 7537 7340 Zurich: Ph.41 0 32 724 9219
ia Pacific Operations: Bahrain: Ph.973 1753 0437 Riyadh: Ph.966 1 4649 750 Singapore: Ph.65 6333 1344 Sydney: Ph.61 2 9267 1955 Tokyo: Ph.81 3 5405 9410

www.polaris.co.in

DECLARATION

I hereby declare that the project entitled **QUERY GENERATION,** submitted towards the fulfillment of M.Sc Applied Science (Software Engineering) from Bharathiar University is a record of original work done by me under the supervision of **Mr.Hemanth Kumar M.C.A.,Polaris Software Lab Ltd**, Chennai and **Mr.K.R.Baskaran, M.S.,**Course Co-ordinator Dept of Computer Science and Engineering, Kumaraguru College of Technology and this project work has not formed the basis for the award of any Degree/ Diploma/Associate-ship/Fellow-ship or any other similar title to any candidate of any University.

Place: Coimbatore

S. S. Anu

Date: 24-09-2004

Signature of the Student

ACKNOWLEDGEMENT

I deem it as great pleasure to place my deep sense of gratitude and indebtedness to **Dr.K.K.Padmanaban, B.Sc,(Engg.), M.Tech., Ph.D., Principal, Kumaraguru College of Technology** for giving me the opportunity to undertake this project.

I am grateful to, **Dr.S.Thangasamy, Ph.D., Professor and Head of the Department**, Kumaraguru College of Technology, giving me this golden opportunity to carry out my project work successfully.

I like to express my deep sense of gratitude to **Mr. K.R.Baskaran M.S., Project co-ordinator, CSE Department.**I am grateful for his inspiring guidance and whole hearted support which he extended to me throughout the course of this project.

My sincere thanks are offered to **Mr. K.R.Baskaran M.S.,** for the encouragement and support bestowed on me as my project guide .I am very much indebted to her for the suggestions and guidance extended in successfully completing the project

I express my profound sense of gratitude to **Polaris software Lab**, for giving me this excellent opportunity to work in their esteemed organization.

I am deeply indebted to **Mr. Hemanth Kumar, Project Leader ,EOD Team**, for giving me valuable suggestions and complete guidance at every point of my project work.

I also thank all other members in the team, for lending their support for the successful completion of the project.

I owe my gratitude to my parents who have encouraged me a lot towards the successful completion of this project work.

Synopsis

The project work is an enhancement of Computerized Banking System developed by Polaris Software Lab Ltd. The existing system is a sophisticated system. I t includes the following channels, the means of communication with bank.

Branch Banking

Through this channel all customer interaction with bank directly at branches. It includes from account opening till closing of accounts, transactions like booking Demand Drafts (**DD**), transfer of money from one city to another city account-Tele Transferring (**TT**), Draft Booked for the same location of bank-Managers Cheque (**MC),** etc.,

Internet Banking

This is another channel, Internet, through which transaction can be done. In this mode the customer need not to come to branch. He can go to the bank's site in internet and do the transaction available in this mode. It currently includes very few services offered by bank like opening new accounts, booking DD, etc.,

Telephone Banking

This channel allows customer to call up the bank's officers over phone and do transactions. This mode allows most of the features available in the bank. The bank has also other facilities like Automated Teller Machine cards **(ATM)**, Credit cards, etc.. All these are fully computerized.

Project Category

The category of the project is Relational Database Management System (DBMS). The system is for the bank with branches all over India. It has to share information stored in the database. So it has to be kept centrally which can be accessed by all through network connection. This would obviously suggest to go for a Database Management System has all facility to store and retrieve information.

We cannot go for a system which will do everything centralized. Some activities has to be decentralized. The activities like formatting input, output and some manipulation have to be done in the client side. This will be done by client server architecture Database Management System. A Relational Database Management System will ideally suit for this scenario. The existing system also using a Oracle, a relational database management system.

CONTENTS PAGE NO

1.INTRODUCTION

1.1 Project Overview

Our aim is to create a query designer to automate query building based on dynamic user input. The inputs are on various business entities of a large web-based banking applications [Citi bank].

Re-usability of query is achieved through an effective repository scheduled query execution provides ease of operations and saves time.

The main scope of our designing is for the following features

- To eliminate user interaction
- Reduction of Cost
- Effective usage of time
- Automation of query
- Query repository building
- Query effectiveness
- Ease of operation

1.2 Organization Profile

Polaris is a global software house providing quality and customized IT solutions to several multinational clients across 20 countries. Polaris has broad-based its areas of expertise, given the wide variety of projects it garnered from a distributed portfolio of clients. Quantum growth has not been at the cost of quality. Polaris was also assessed and certified as an ISO 9001 company by the KPMG quality registrars.

Polaris mission is to remain a reliable and responsive global player in offering out souring solutions in the area of-

- Banking & Financial services
- Risk Management
- Insurance
- Enterprise Knowledge Management
- Transportation
- ERP
- Retail solutions
- Undertaking specialized high end consulting assignments.

Polaris has over 2000 associates spread all over the world. Polaris has its worldwide head quarters at Chennai, India. Polaris has two software development centers one at Chennai and one at Noida and marketing offices in Bangalore,Hyderabad and Mumbai.Polaris through its branches has four international offices in New Jersy, California, London and Singapore. Through its alliance partners it has a presence in six other international locations.

Polaris Software Lab specializes in designing, developing, implementing and supporting banking software solutions for banks and financial institutions globally. The company's domain expertise covers a range if banking activities, from retail banking, Credit card systems, branch automation and electronic banking to decision support, front office trading, risk management, lending and trade finance.

Polaris was formed in 1983.In 1986,it entered into an everlasting relationship with Citibank, India for their 'Retail Banking Automation'. Polaris has been successfully taking up projects given by Citibank.

2. System Study And Analysis

2.1 Existing System

The existing system includes full automation of banking system. Any scheme of money deposit, loan is called a product. For example normal savings account is also a product. Like this a lot of products are available, which can be added whenever bank goes with a new offer. All these products have its own services offered and service charges for that, which actually varies. There is no common method to enforce this to all customers treating them as same in for all the products.

Charging is a main aspect, which is vital income for any financial concern, in bank, charges, has its impact in all transactions generally. In present situation, charges to be applied for transactions have more parameters to be decided. The bank now has various modes of communication, like ATM, Internet Banking, Phone Banking. In ATM and Internet Banking human intervention is less. The bank can have less manpower. Obviously bank will prefer such modes which need less human power. To handle customer in branches, there will be a need of more space and human resources. So this mode usage can be reduced to the minimum. In this situation bank can do minimum charges to the modes preferable from bank side.

Already there are charges available in existing current system. They are calculated in respective modules as per the need. Since it is calculated in individual modules there is no consistency in charging customers.

In banking, the customers will not be under same priority. Some customers value more for bank. For such customers bank may decide to charge less or not to charge.

2.2 Proposed System

In the proposed system the system is designed as user friendly. The various requirements are collected from the customers through mails and the query is generated to fulfill the requirements of the customer.

The customer can send all the information through mails. This inciudes the various transactions the customer needs in the new system. The designer will generate queries to perform those transactions.

3.Programming Environment

3.1 Description of Software and Tools used

The existing system has **Oracle 7.1** as it back-end and the front-ends **are C, SQL* Forms 3.1 & Java** .The operating system **is UNIX, SUN SOLARIS**. The enhancement project is developed in C and Oracle procedure.

Special Feature of Language/Utility:

Information is the back bone of any organization. Therefore it has to be made available at all times to ensure proper decision. Information has to be accurate, current, timely, relevant and usable. This need for managing data and retrieving information has led to an extensive use of computers in the business world. The effectiveness of the system by and large depends on the way in which the data are organized and managed. Database Management System helps to manage data in a more efficient manner than the traditional file management system.

A database is a collection of interrelated data that are to be stored together in a single location. It enables sharing of data among various users as and when required. A DBMS is a software system with capabilities to organize, manipulate and manage data.

A database system consists of two parts namely the DBMS which are the programs that organizes and maintains the information and database application, a program that let us retrieve, view and update the information stored by DBMS.

Oracle

Oracle is one of the most full featured Relational Management System. The collection of tools, utilities and applications let one to manipulate an oracle database. An application developed in oracle will be able to keep pace with growth and change in the company and in the world at large although Oracle demands greater expertise on the part of the application developer.

Getting Started

Oracle Kernel is the main program in the RDBMS. It controls the actions of the consistent parts. Kernel interprets the requests issued by the Oracle products and controls their access to data through the shared Global Areas.

The Oracle system uses a non procedural language to communicate with its database kernel. SQL is so powerful that all the application development tools provided are SQL based because oracle uses SQL as its query language. Oracle provides SQL interface to the database through SQL*plus, interactive forms developer SQL* forms. The transferability of data files from many files-FLAT file format into table structures of Oracle database and capability to interface with variety of third generation languages such as C, COBOL, FORTRON, PASCAL etc..

The language used by oracle is standard for all the RDBMS:SQL

DDL: Data Definition Language

These are the statements used to create and maintain the database
e.g. CREATE table, DROP table, CREATE view, DROP view etc

DML: Data Manipulation Language

These are the statements to manipulate data in the table
e.g. SELECT, DELETE, INSERT, UPDATE etc..

DCL: Data Control Language

These are the statements used to control usage of the data
e.g. GRANT,REVOKE,COMMIT,ROLLBACK etc.

Query generation will be done with C as front-end for interactive screens and the main query functionality will be done as Oracle stored procedure. So the procedure can be globally accessed from all existing front-end modules.

Details About Pro*C

In the tools and topics that we dealt so far we have been using SQL to perform all tasks. Although the power and ease of use of SQL is unquestionable some of the capabilities of the third generation languages are not to be found.

Consider the case when we need to access devices from inside oracle. For example , if we need to print the record everytime a record is created , it is possible only with the help of an interface of a 3GL with oracle.

ORACLE as we already know supports the use of the following high-level languages as an interface with the database.

C
COBOL
FORTRAN
PASCAL
PL/I

However , the most commonly used interface with oracle is C. The SQL data language is a non-procedural language , which implies that most of the statements are executed independent of the statements.

Having specially designed SQL to be a non-procedural language , and thus understanding the limitations of such a language , the originators of SQL also designed SQL constructs to be embedded in procedural programming languages such as C. With these constructs programmers can design applications which also utilize the best features of the programming language (host language).

ORACLE RDBMS supports two kinds of interfaces

SQL Interface (PRO*C pre compiler interface)
ORACLE Interface

SQL interface uses the capabilities of host languages for processing and it uses the embedded SQL statement to fetch the data from the ORACLE RDBMS and it is easy to write and understand.

ORACLE Interface uses the actual host language calls to access the database.

This is somewhat tedious to use.

NUCAPT

NUCAPT (Nucleus Application Productivity Tool) is a front-end utility written using cursors & panel libraries and few system calls. It simulates some important features from Oracle Forms. So that users of NUCAPT will get the sensation of writing forms programs in C language . So this utility is mainly useful for application programmers writing interactive programs related to query, online transaction etc.,

NUCAPT program simulates form level objects like Block, Field and Page.

-Users can treat the Application , Block, Field and page as an object and specify the properties for them.
-An application consists of blocks and a block consists of fields.
A fields is associated with any page i.e. A block can span across any page.

-All the NAPT function definitions, variables and macros are defined in a header file “form.h ” which has to be included by the user.

-User have to declare the variables for the type defined structure.

4. System Design

The design of an information system produces the details that show how a system will meet the requirements identified during system analysis. This stage is referred to as logical design in contrast to the process of developing software, which is referred to as physical design. The system designed using inputs from system design

4.1 Design Overview

Introduction & concepts

Our aim is to create a query designer to automate query building based on dynamic user input. The inputs are on various business entities of a large web-based banking applications [citi bank].

Re-usability of query is achieved through an effective repository scheduled query execution provides ease of operations and saves time.

Design Basis

The design basis involves three processes and they are

- Maintenance
- Query designing
- Creation of EOD – Which involves the writing of programs to execute the query designing

Functionality overview

Under the functionality overview query designing supports for the following characteristics

- Query repository
- Automation
- Query optimization

4.2 Design Description

Scope &purpose

The scope of our project query generation is to create a query designer to automate query building based on dynamic user input. The inputs are on various business entities of a large web-based banking application.

Functional description

The functionality overview supports for the following characteristics and their description is given below

Re-usability of query is being achieved through an effective repository.

Scheduled query execution provides ease of operation.

Dynamic processing of query is being enabled through the concept of automation. The query as an individual, itself helps in performing all the functions of transactions being employed in the bank.

All these functions of a query are being achieved based on the information provided by customer and by the table informations and other information provided by the user.

4.3 Database Definition

Module description

The query designing as a whole single module helps in eliminating the user interaction and also in the easy execution of the operation.

Design considerations

Under design considerations the following activities are being considered.

Under the maintenance feature we get the input from the customer through mails and we categorize the input in the form of table. So the concept table designing is considered here and the table consists of various fields under it.

And in the next step i.e in the feature of query designing the description of tables are being considered and also the field informations are also being provided

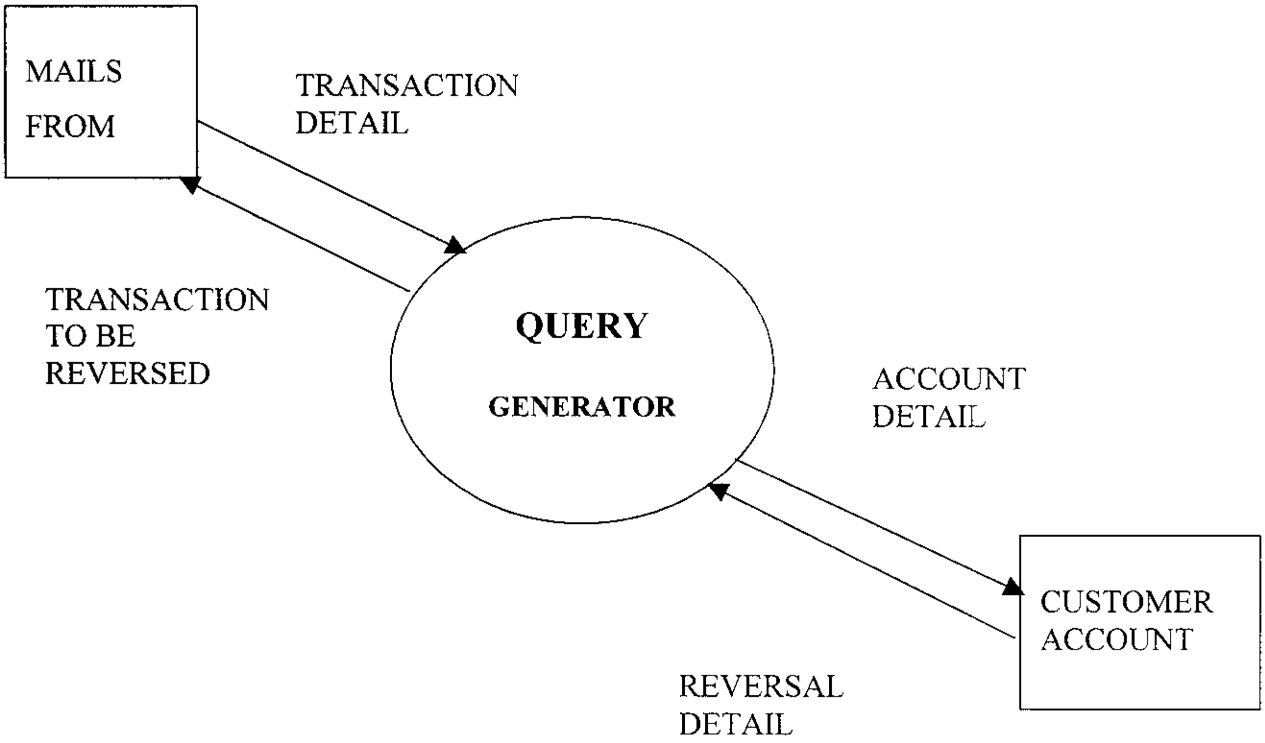
The third step is the final step i.e writing of the program to execute the query. While writing or framing the program the table information and the field information are being accessed while selecting the table the concept of multi-select is also enabled. The criteria and the conditions being provided also helps in the execution of the program[query].

4.4 Dataflow Diagram

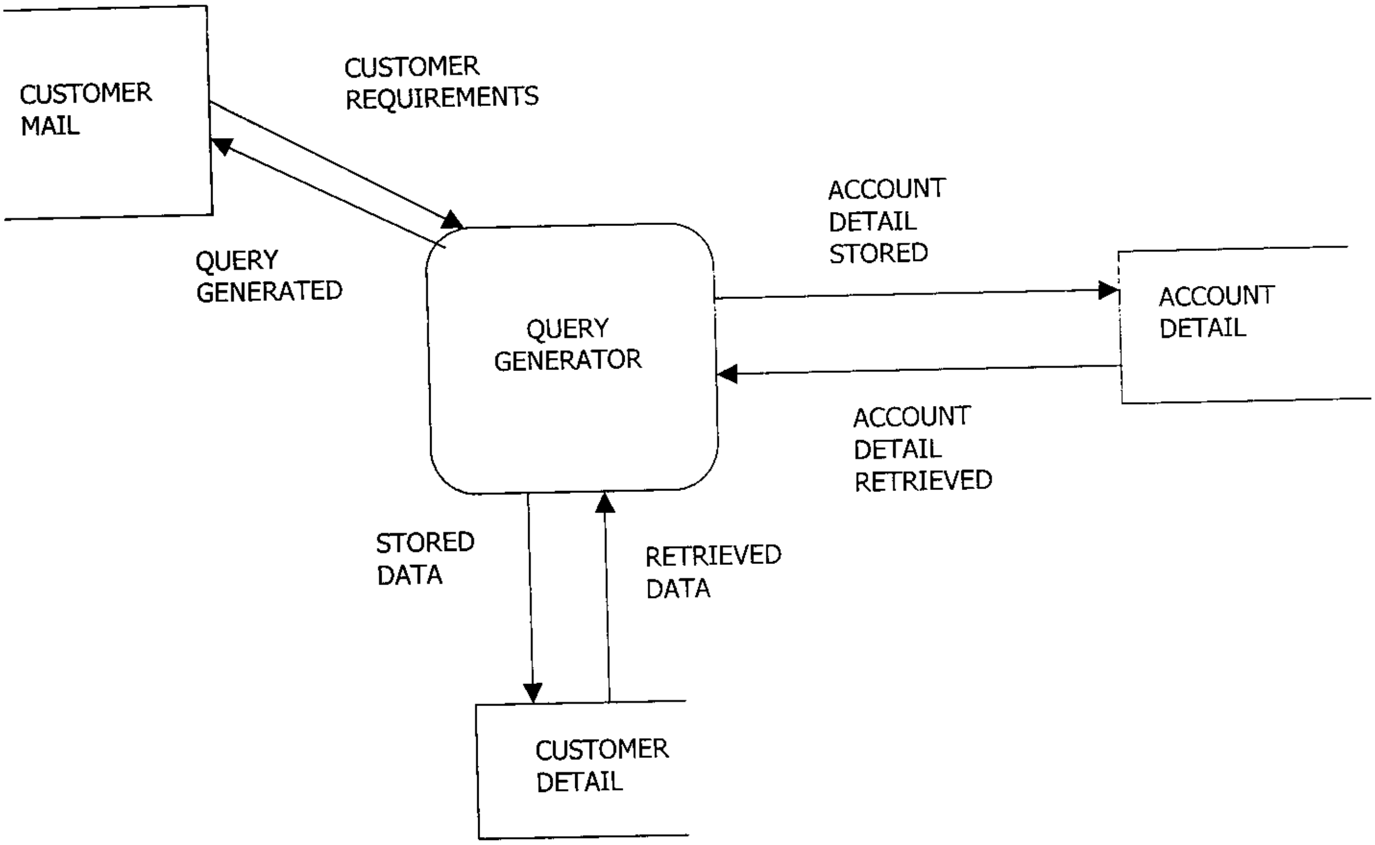
Data flow diagram is a graphical tool. It is used to describe and analyze the movement of data flowing into the system – manual or automated. They focus on the data flowing into the system, between processes and in and out of data stores. This is a central tool and the basis from which other components are developed.
The system models are termed as Data flow diagram (DFD).

The initial context diagram should be an overview including basic inputs, process, and outputs. This will be the most general diagram, really a bird's eye view of data movement. With a top-down approach to diagramming data movement. With a top-down approach, the diagrams move from general to specific. While the first diagram helps the system analyst grasp basic data movement, its general nature limits its usefulness.

CONTEXT DIAGRAM



DATA FLOW DIAGRAM



5.System Implementation and Testing

5.1 System Implementation

An initial investigation culminates a proposal, which describes and evaluates the proposed system, meets system performance requirements, helps to acquire a sense of scope of the problem, leads to the initiation of the feasibility study. The problem definition is crystallized and aspects of the problem that are to be included in the system are designed.

Feasibility study decides whether the proposes is feasible for the client. It includes three topics they are,

a. TECHNICAL FEASIBILTY
b. ECONOMIC FEASIBILITY
c. OPERATIONAL FEASIBILITY

5.1 a. Technical Feasibility

This deals about current technical position of client. Whether the client has hardware, software required for the system. If not it is feasible to get a new one.

The existing system is already computerized. Since the new system is going to be an enhancement to the existing system, it does not need any additional hardware or software requirements. So it is technically feasible.

5.1 b.Economical Feasibility

This deals whether client is feasible with the new expenses because of the new system, this includes the cost of new hardware, software. It also includes the development cost for new system.

In this new system since there is no additional hardware or software requirements, only development cost is to be considered, which is acceptable for client. So the system is also economically feasible.

5.1 c. Operational Feasibility

This deals with whether there is any operational problem expected because of the new system. Whether new system is acceptable by end-user without hesitation.

The new system is an enhancement for existing system the user will not feel any difference in use of this new system. So it is also operationally feasible.

5.2 System Testing

Managing Quality Assurance

Quality assurance is the review of software products and related documentation for completeness, correctness, reliability and maintainability. It also includes assurance, that the system meets the specification and the requirements for its intended use and performance.

Quality assurance can be done by

- Testing
- Verification and Validation

Testing is generally done at two levels-testing of individual modules and testing of the entire system (system testing).

During testing, the system is used experimentally to ensure that the software does not fail, i.e., that it will run according to its specifications and in the way users expect.

Special test data are input for processing, and the results examined. A limited number of users may be allowed to use the system so analysts can see whether they use it in foreseen ways. It is preferable to discover any surprises before the organization implements the system and depends on it.

Testing Strategies

These are two general strategies for testing software. The strategies are Code testing and Specification testing.

5.2 a. Code testing

The code testing strategy examines the logic of the program. For this testing method, the analyst develops test cases that result in executing every instruction in the program or module, which is every path through the program, is tested.

A path is a specific combination of conditions that is handled by the program. The testing of every path in the program is not always possible as there could be several thousands, and financial and time limitations will not permit this. Generally, all the frequently used paths undergo testing.

5.2 b. Specification testing

To perform specification testing, the analyst examines the program specifications wherein states what the program should do and how should perform under various conditions. Then, test cases are developed for each condition or combination of conditions and submitted for processing.

By examining the results, the analyst can determine whether the program performs according to its specified requirements. This testing does not look into program to study the code or path; it looks at the program as a whole. The assumption here is that, if the program meets the specifications it will not fail.

Types of test data

There are two very different sources of test data

* Live
* Artificial

Both have advantages and disadvantages

Using live test data

The test data are those that are actually extracted from organization files. This shows how the system will perform on typical data. Although, the data may be the best, it is difficult to obtain sufficient amounts to conduct extensive testing. All combinations and conditions of the system are not tested with this data. This may not contain values that may cause system failure.

Using artificial data

Artificial test data is solely for test purposes. They are to be generated to test all combinations of formats and values. They are generated using the utility programs of the information systems. Using this type of data all logic and control paths through the program can be tested.

For best results, the artificial test data should be generated by persons other than those who wrote the programs. Automated test data generators are also available.

Levels of testing

Unit testing

This involves the tests carried out on modules/programs which make up a system. This is also as program testing. The units in a system are the modules and routines that are assembled and integrated to perform a specific function.

In a large system, many modules are needed. Unit testing focuses first on the modules, independently of one another, to locate errors.

The programs should be tested for correctness of logic applied and should detect errors in coding. Valid and invalid data should be created and the programs should be made to process this data to catch errors. The programs are also checked to get the valid data only, like valid date inputs.

All the conditions present in the program should be tested. Before proceeding one must make sure that all programs are working independently.

Systems testing

When unit tests are satisfactorily concluded, the system as a complete entity must be tested. At this stage, end-users and operators become actively involved in testing. While testing one should also test to find discrepancies and system documentation.

System testing must verify that database size is adequate and that indexes have been built properly.

Special system tests

There are other tests that are in a special category as they do not focus on the normal running of the system. They are

Peak load testing

This is used to determine whether the system will handle the volume of activities that occur when the system is at peak of its processing demand. For instance all terminals are active at the same time.

This test applies mainly for on-line systems. In our system, banking system, analyst want to know what will happen if all tellers sign at their terminals at the same time before start of business day. Will the system handle them one at a time without incident that it 'locks up' and must be restarted or will terminal addresses be lost? The only way sure way to find out is to test for it.

Performance testing

This test refers to the response time of the system being installed. Performance time testing is conducted prior to implementation to determine how long it takes to receive a response to a inquiry, send a transmission and receive response.

This also includes test runs to time indexing of large files of the size of the system will have a typical run or to prepare a report. A system may run well with only handful of test transactions may be unacceptable slow when fully loaded. This should be done using the entire volume of live data.

Storage testing

This test is to be carried out to determine the capacity of the system to store transaction data on a disk or in other files. Capacities here are measured in terms of the number of records that a disk will handle.

Recovery testing

Analyst must never be too sure of anything. He must always be prepared for the worst. One should assume that the system will fail and data will be damaged or lost. Even though plans and procedures written to cover these situations, they also must be tested.

In our system, Oracle RDBMS is used for storing data. Oracle has a very good recovery option, so that we get no-loss of data or minimum loss of data.

Procedure testing

Documentation and run manuals telling the user how to perform certain functions are tested quite easily by asking the user to follow them exactly through a series of events.

It is surprising how not including instructions about the aspects such as, when to depress the enter key, removing diskettes before putting power off the system and so on, could cause problems. This type of testing brings out what is not mentioned in the documentation, and also the errors in them.

Human factors

In case during processing, the screen goes blank, the operator may start wonder as to what is happening can he could do anything such as press the enter key a number of times, or switch off the system and so on, but if a message is displayed saying that the processing is in progress and asking the operator to wait, then this types of problems can be avoided.

Security testing

This is to be test the security features provided by the system. By this the system is checked for any unauthorized access cannot be done.

The existing system is having an excellent security. All the users have valid user name and password. Using one name only one person can login. Even for the users, the accessing is restricted to
relevant modules only.

The user can run only authorized programs. Some programs need to be executed using dual password option. Means, a second user name and password is required for that program to be executed.

Verification and Validation

Verification testing runs the system in a simulated environment using simulated data. This simulated test is primarily looking for errors and omissions regarding end users and design specifications that were specified in the earlier phases but not fulfilled during construction.

Validation refers to the process of using software in a live environment in order to find errors. The feedback from validation phase generally produces changes in the software to deal with errors and failures that are uncovered. Then a set of user sites is selected that puts the system into use on a live basis. They called beta test sites.

6.Conclusion

The suggested "QUERY GENERATION" has been tested and implemented successfully. Every part of the system found to work efficiently without errors. The development of this system made me face several problems related to a real time system, which improved my knowledge. Some of the important benefits from the system are
*It made software maintenance simpler.
*Can be linked with other systems for better information flow.

Thus the project proves to be a better package than the existing system in all means. This is accomplished by completely analyzing the user needs, the difficulties encountered in the existing system and the flexibility they require. A great attempt has been made to make it as user friendly as possible.

7.Future Scope of the Project

The future application of the project will be based on the customer requirements. The system will satisfy the customer requirements i.e impact. It helps in time saving. It fulfills all the needs of the customer efficiently

8. BIBLIOGRAPHY

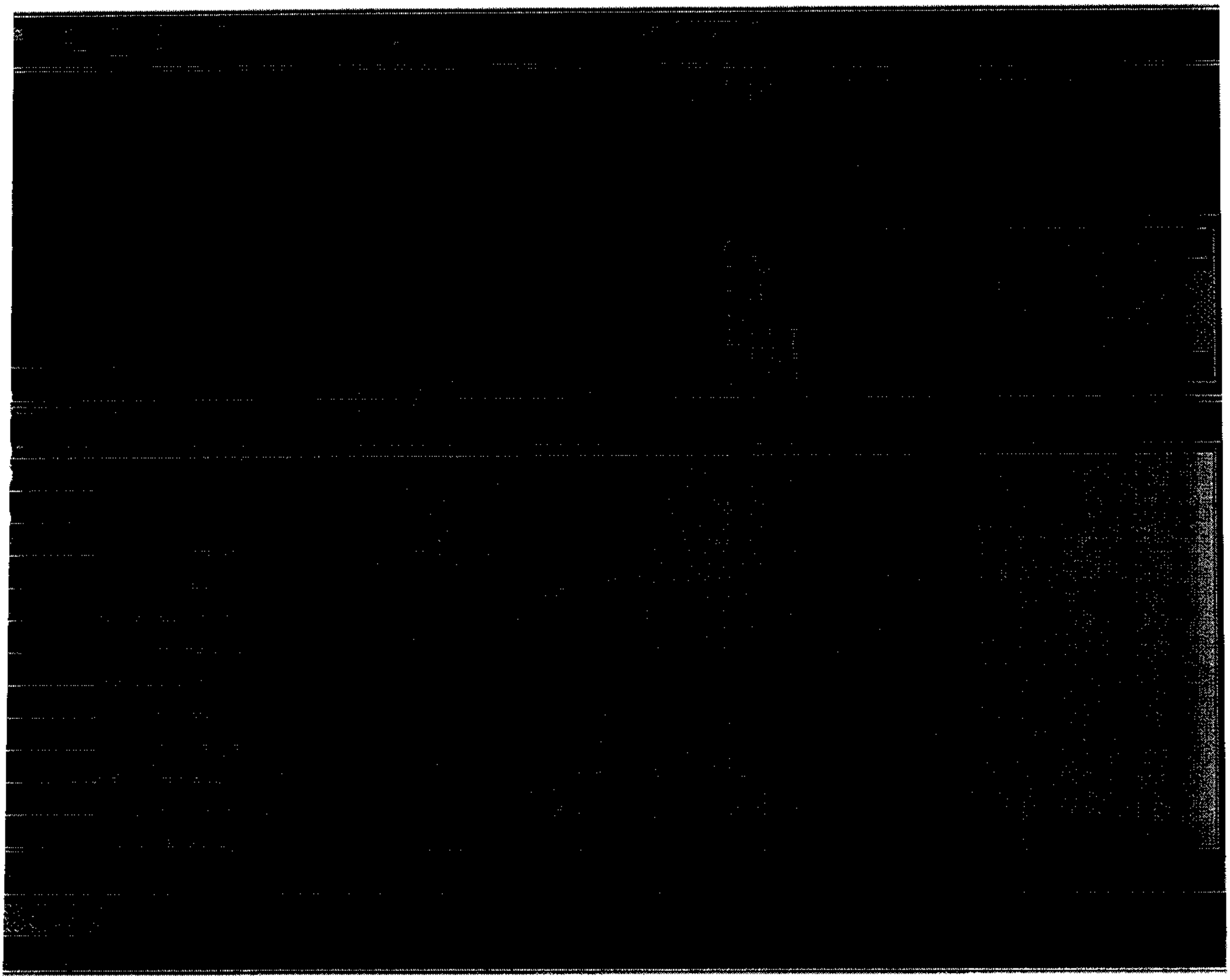
- Ivan Bayross, " ORACLE – 7 Complete Reference", TataMcGrawHill Publishing Company Limited, 1998.

- KEVIN and KOACH, " SQL Complete Reference", QUE Publishing House, Second edition, 1997.

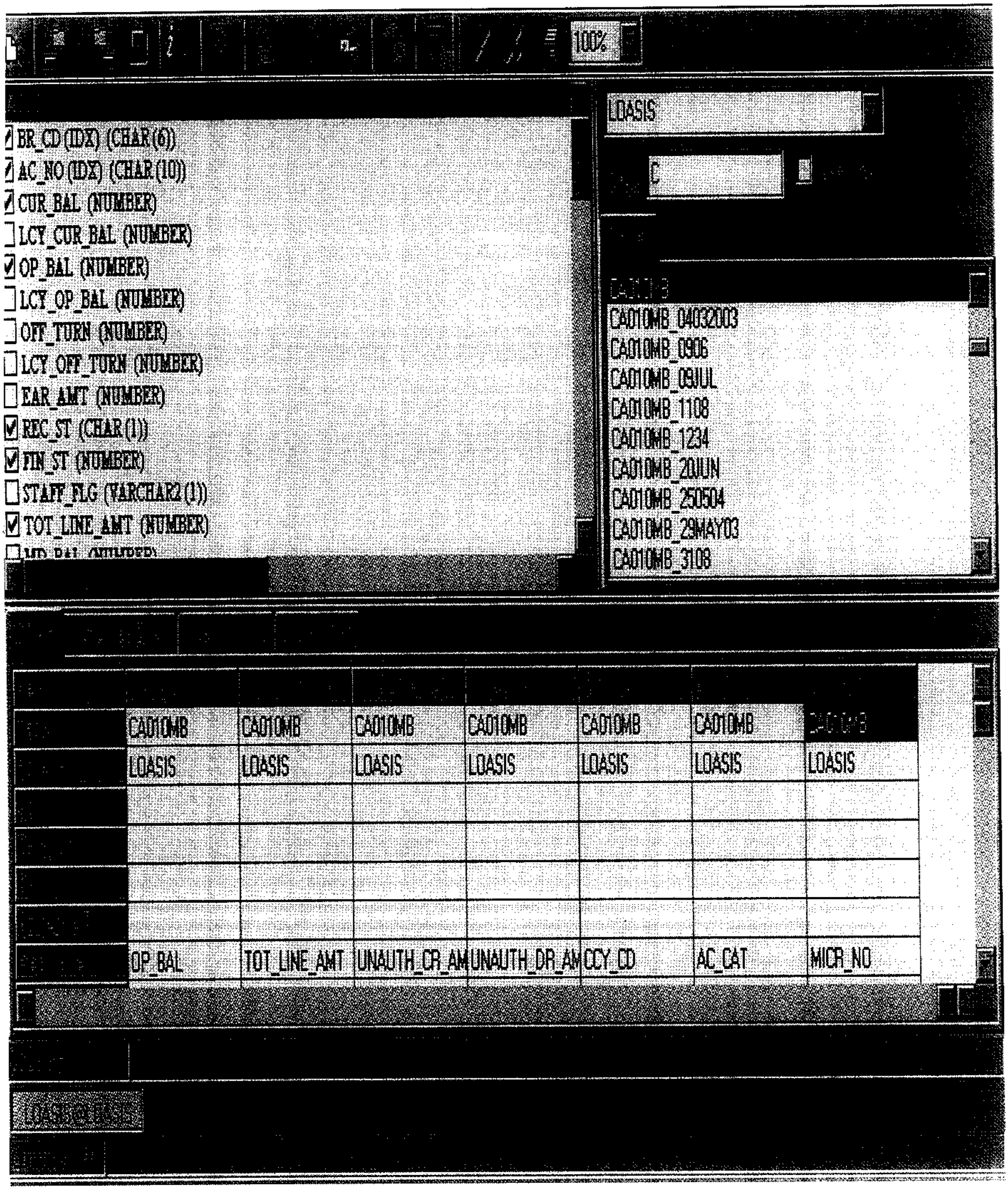
- "PRO C MANUAL", S.S.I Limited.

SCREEN SHOTS USING SINGLE TABLE

GENERAL VIEW OF THE SCREEN



SELECTING TABLES



- ☑ BR_CD (IDX) (CHAR(6))
- ☑ AC_NO (IDX) (CHAR(10))
- ☑ CUR_BAL (NUMBER)
- □ LCY_CUR_BAL (NUMBER)
- ☑ OP_BAL (NUMBER)
- □ LCY_OP_BAL (NUMBER)
- □ OFF_TURN (NUMBER)
- □ LCY_OFF_TURN (NUMBER)
- □ EAR_AMT (NUMBER)
- ☑ REC_ST (CHAR(1))
- ☑ FIN_ST (NUMBER)
- □ STAFF_FLG (VARCHAR2(1))
- ☑ TOT_LINE_AMT (NUMBER)
- □ MD_BAL (NUMBER)

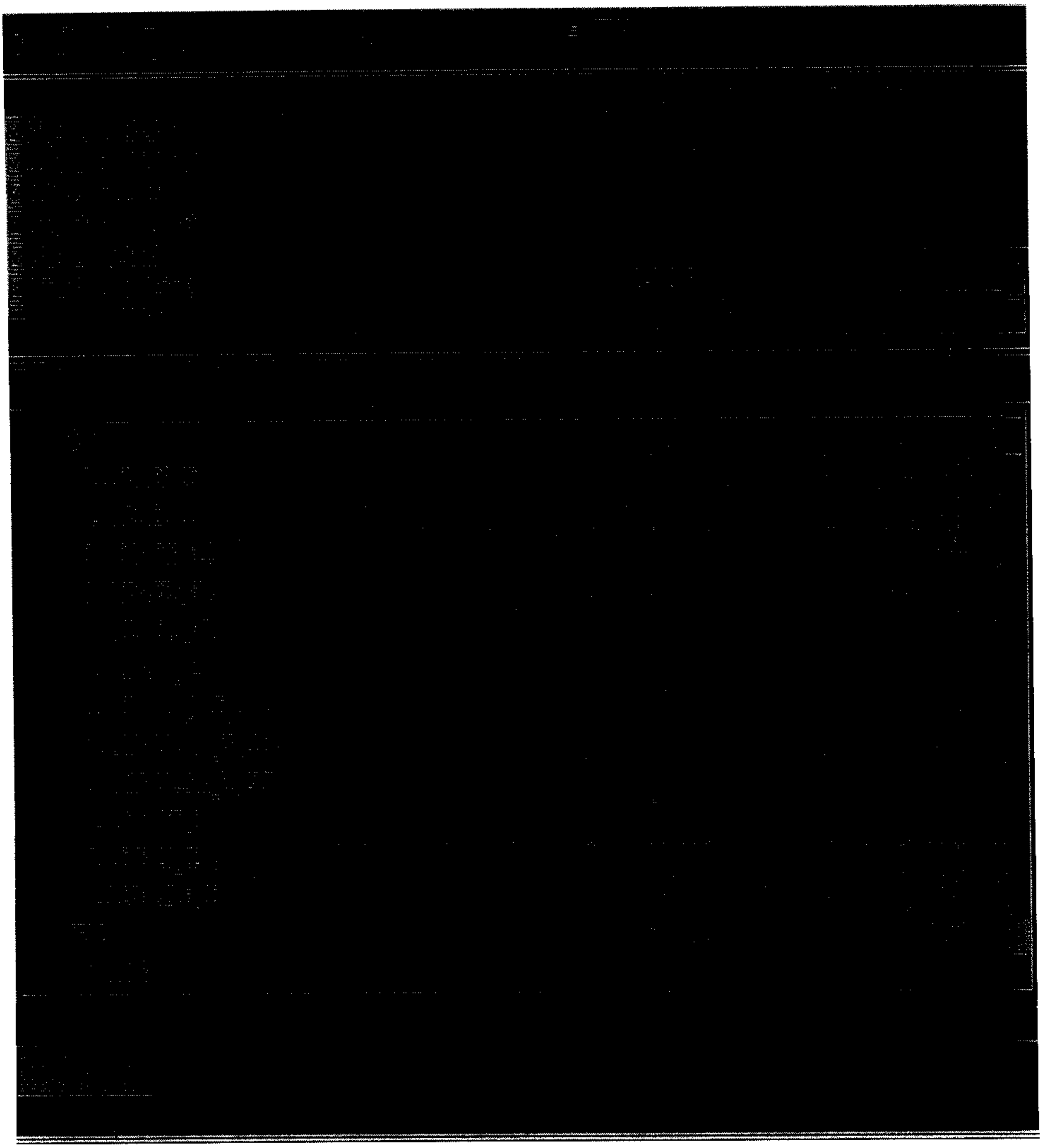
LOASIS

C

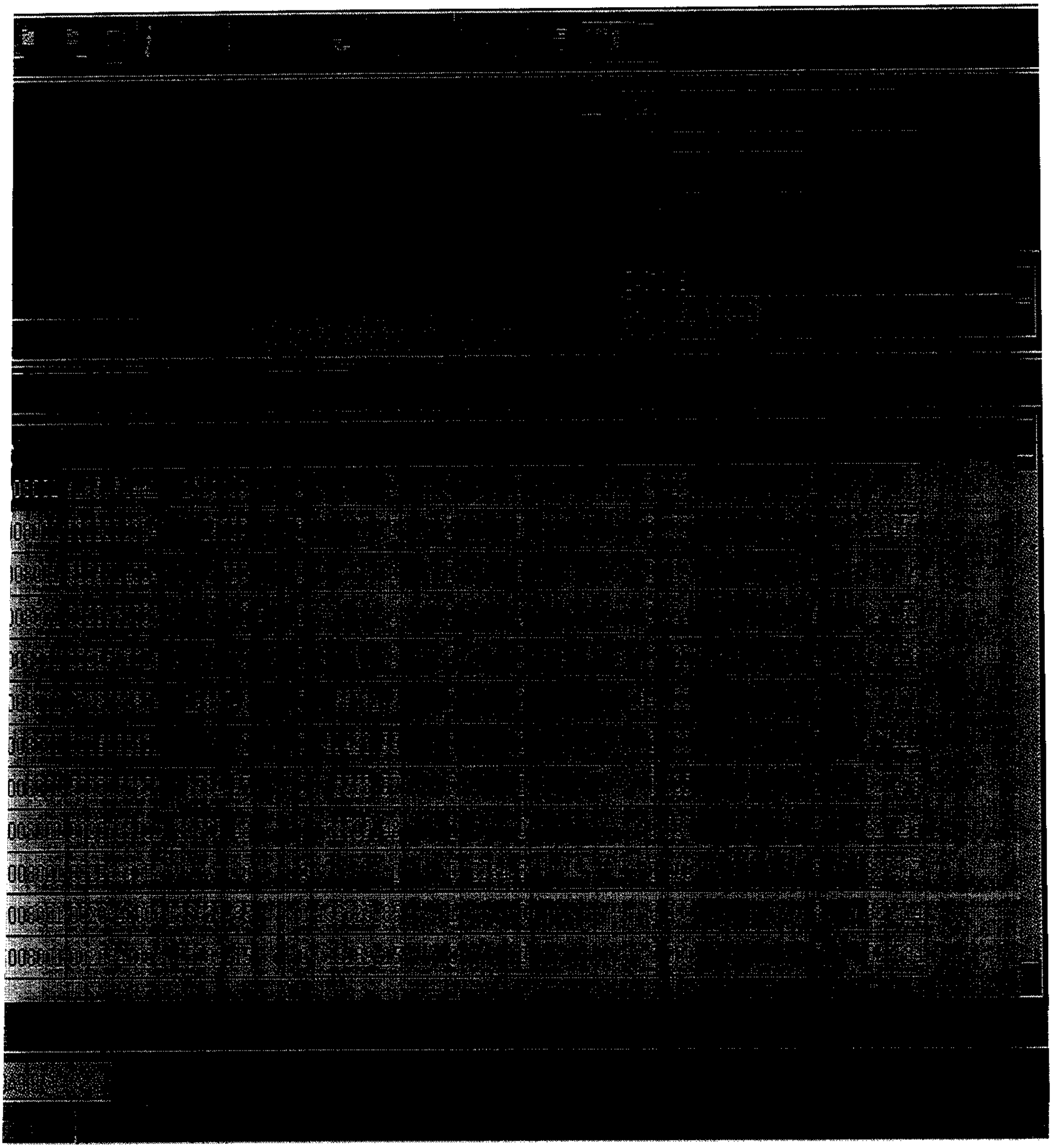
- CA010MB
- CA010MB_04032003
- CA010MB_0906
- CA010MB_09JUL
- CA010MB_1108
- CA010MB_1234
- CA010MB_20JUN
- CA010MB_250504
- CA010MB_29MAY03
- CA010MB_3108

CA010MB	CA010MB	CA010MB	CA010MB	CA010MB	CA010MB	CA010MB
LOASIS	LOASIS	LOASIS	LOASIS	LOASIS	LOASIS	LOASIS
OP_BAL	TOT_LINE_AMT	UNAUTH_CR_AM	UNAUTH_DR_AM	CCY_CD	AC_CAT	MICR_NO

QUERY GENERATING

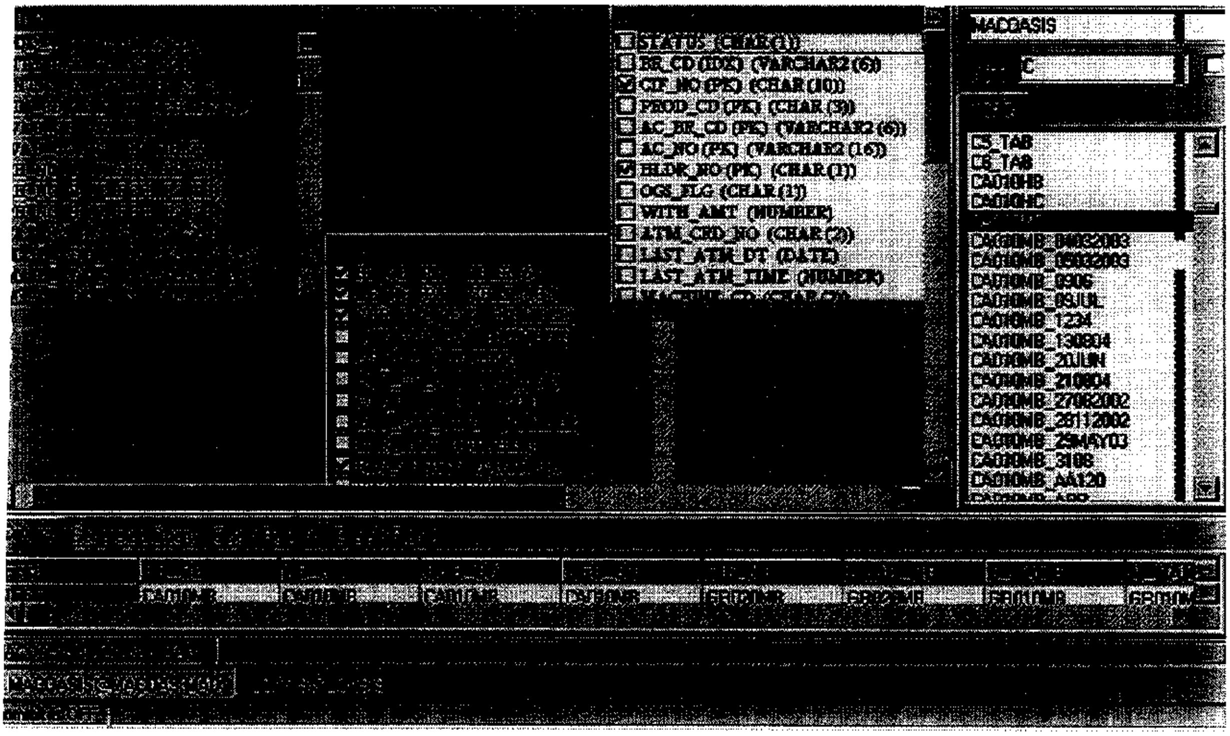


QUERY RESULTS



SCREEN SHOTS USING MULTIPLE TABLE

SELECTION OF TABLES



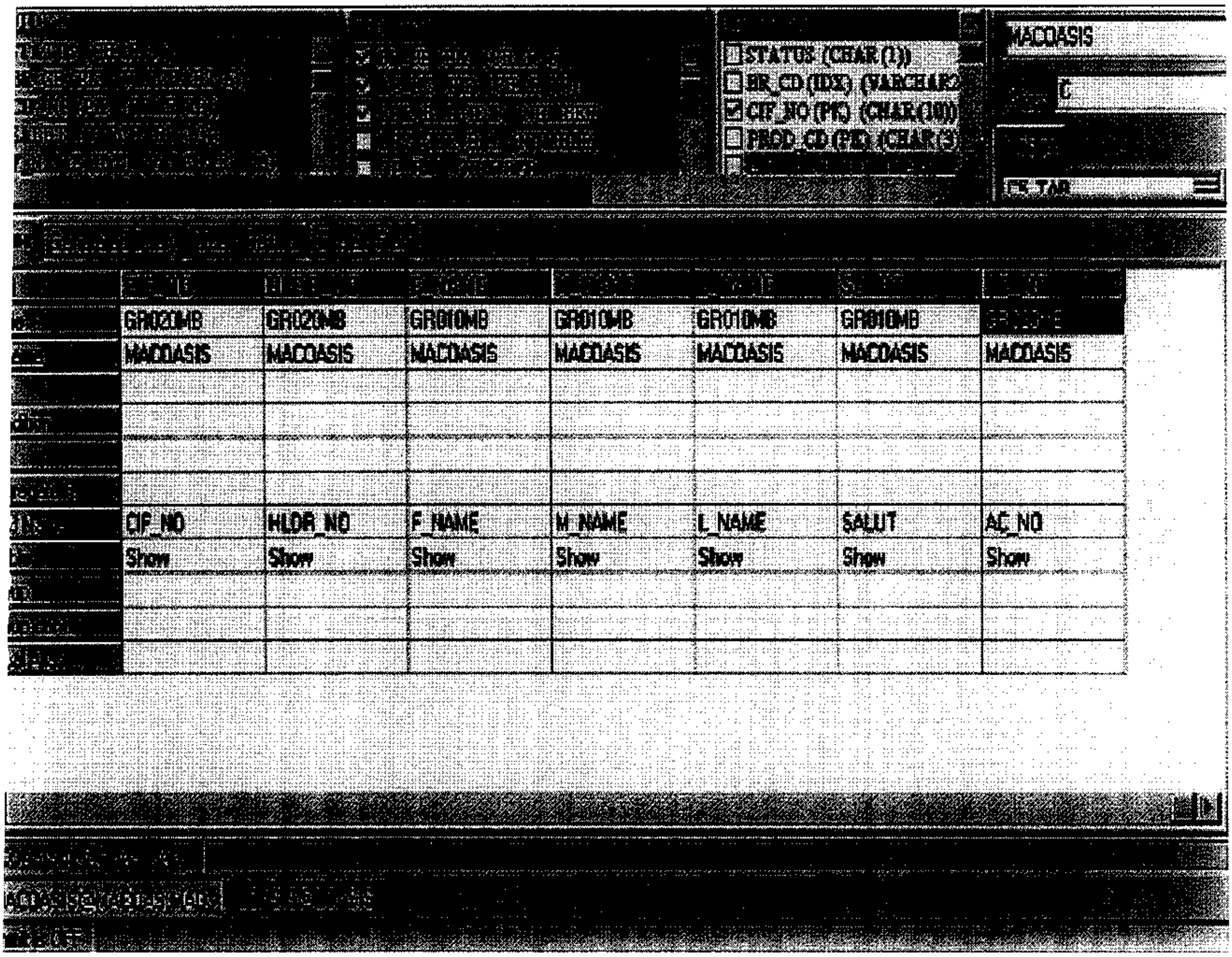
☐ STATUS (CHAR (1))
☐ BR_CD (IDX) (VARCHAR2 (6))
☑ CIF_NO (PK) (CHAR (10))
☐ PROD_CD (PK) (CHAR (3))
☐ AC_BR_CD (PK) (VARCHAR2 (6))
☐ AC_NO (PK) (VARCHAR2 (16))
☑ HLDR_NO (PK) (CHAR (1))
☐ OGS_FLG (CHAR (1))
☐ WITH_AMT (NUMBER)
☐ ATM_CRD_NO (CHAR (2))
☐ LAST_ATM_DT (DATE)
☐ LAST_ATM_TIME (NUMBER)

MACOASIS

CS_TAB
CS_TAB
CA010HB
CA010HC

CA010MB_04032003
CA010MB_05032003
CA010MB_0906
CA010MB_09JUL
CA010MB_1234
CA010MB_130904
CA010MB_20JUN
CA010MB_210804
CA010MB_27062002
CA010MB_28112002
CA010MB_29MAY03
CA010MB_3108
CA010MB_AA120

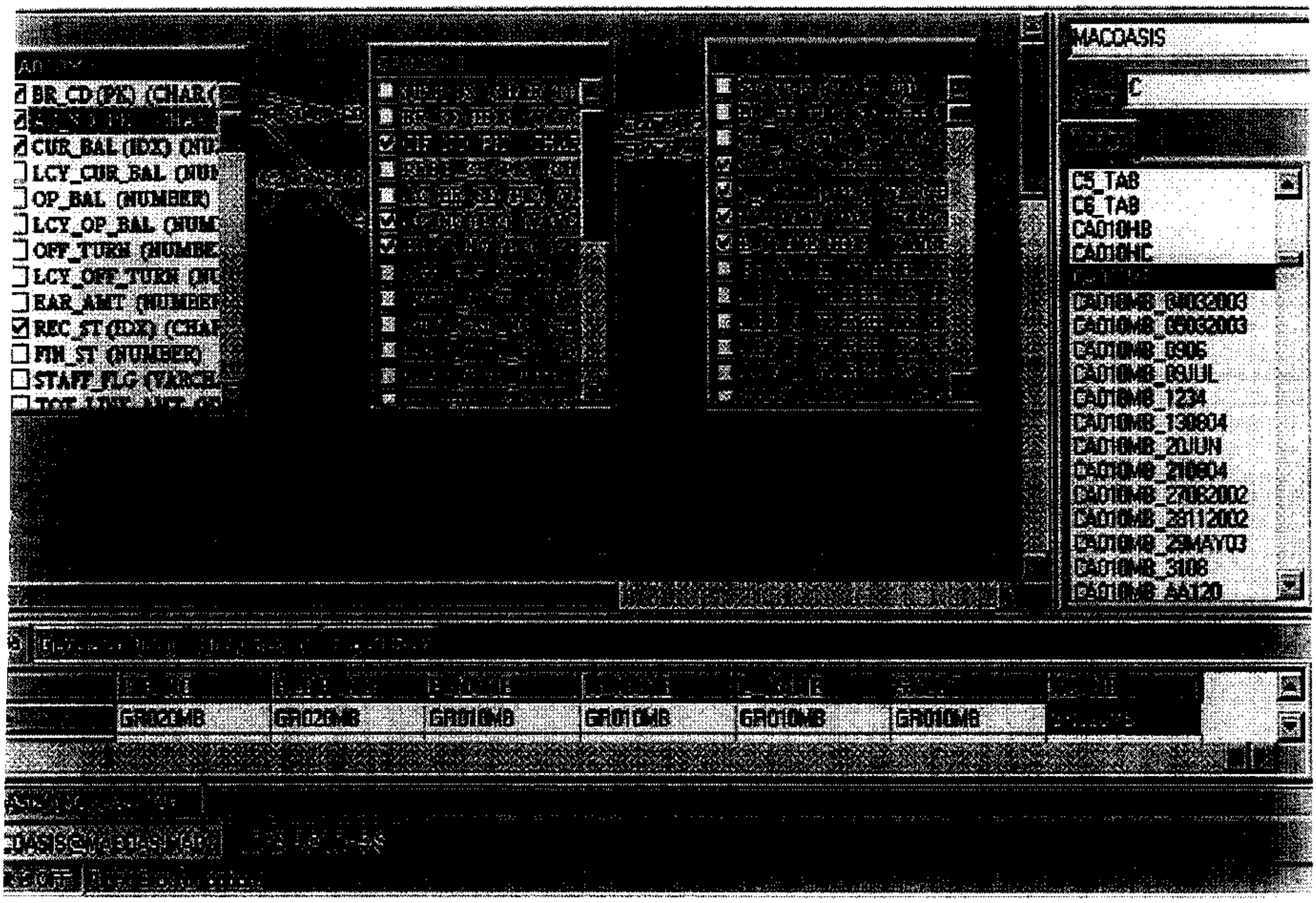
DISPLAY OF THE FIELDS



□ STATUS (CHAR(1))
□ BR_CD (IDX) (VARCHAR2
☑ CIF_NO (PK) (CHAR(10))
□ PROD_CD (PK) (CHAR(5)

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	GR020MB	GR020MB	GR010MB	GR010MB	GR010MB	GR010MB	[illegible]
[illegible]	MACOASIS	MACOASIS	MACOASIS	MACOASIS	MACOASIS	MACOASIS	MACOASIS
[illegible]							
[illegible]							
[illegible]							
[illegible]							
[illegible]	CIF_NO	HLDR_NO	F_NAME	M_NAME	L_NAME	SALUT	AC_NO
[illegible]	Show	Show	Show	Show	Show	Show	Show
[illegible]							
[illegible]							
[illegible]							

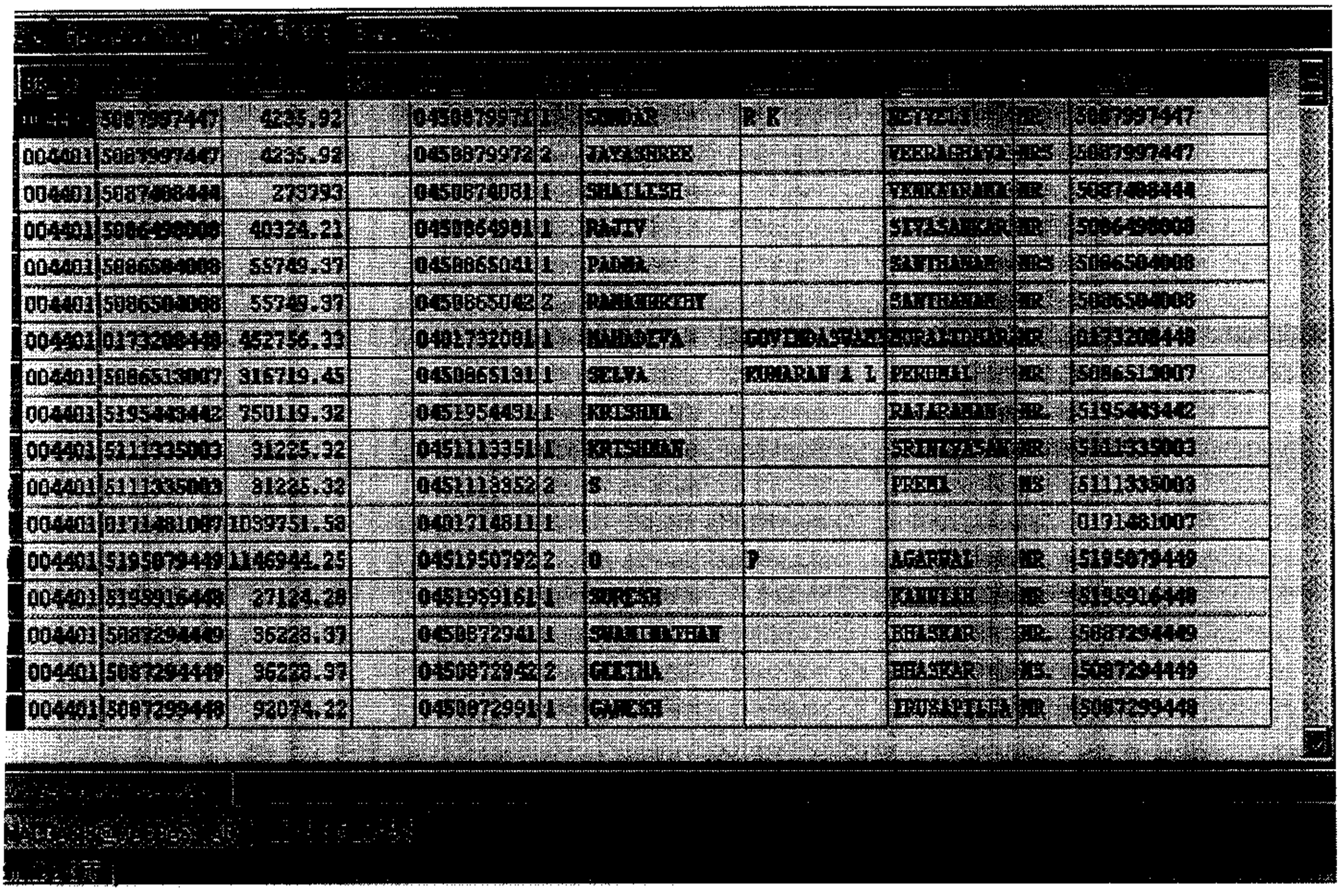
CONCEPT OF JOIN



GIVING THE CRITERIAS

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	MACDASIS
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	M_NAME
[illegible]	Show	Show	Show	Show	Show	Show	Show	Show

QUERY RESULTS



004401	5087997447	4235.92		0450879971	1	SUNDAR	R K	[illegible]	MR	5087997447
004401	5087997447	4235.92		0450879972	2	JAYASHREE		VEERAGHAVA	MRS	5087997447
004401	5087408444	273793		0450874081	1	SHAILESH		VENKATRAMA	MR	5087408444
004401	5086490008	40324.21		0450864901	1	RAJIV		SIVASANKAR	MR	5086490008
004401	5086504008	55749.37		0450865041	1	PADMA		SANTHANAM	MRS	5086504008
004401	5086504008	55749.37		0450865042	2	[illegible]		SANTHANAM	MR	5086504008
004401	0173208448	452756.33		0401732081	1	NANDOLYA	GOVINDASWAMY	[illegible]	MR	0173208448
004401	5086513007	316719.45		0450865131	1	SELVA	KUMARAN A L	PERUMAL	MR	5086513007
004401	5195443442	750119.32		0451954431	1	KRISHNA		RAJARAMAN	MR.	5195443442
004401	5111335003	31225.32		0451113351	1	KRISHNAN		SRINIVASAN	MR	5111335003
004401	5111335003	31225.32		0451113352	2	S		PREMA	MS	5111335003
004401	0171481007	1039751.58		0401714811	1					0171481007
004401	5195079449	1146944.25		0451950792	2	O	P	AGARWAL	MR	5195079449
004401	5195916448	27124.28		0451959161	1	SURESH		KANNIAH	MR	5195916448
004401	5087294449	36228.37		0450872941	1	SWAMINATHAN		BHASKAR	MR.	5087294449
004401	5087294449	36228.37		0450872942	2	GEETHA		BHASKAR	MS.	5087294449
004401	5087299448	92074.22		0450872991	1	GANESH		IDUSAPILLA	MR	5087299448