GUI FRONT END FOR POSTGRESS

By

M.KARTHIK
(Registration Number: 71205621018)

of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree

of

MASTER OF COMPUTER APPLICATION

ANNA UNIVERSITY
CHENNAI 600 025

JUNE, 2008

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

DEPARTMENT OF COMPUTER APPLICATION

BONAFIDE CERTIFICATE

Certified that this project report titled **GUI FRONT END FOR POSTGRESS** is the bonafide work of **Mr. M.KARTHIK (Registration Number: 71205621018)** who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

N. Jagadeesan
Supervisor

Head of the Department

Submitted to Project and Viva Examination held on 01-07-08

Internal Examiner

01/07/08
External Examiner

Team Landmark - IT
Systems and Solutions (P) Ltd.
Beyond IT, We Apply IT.

June 4, 2008

TO WHOMSOEVER IT MAY CONCERN

This is to inform you that Mr. M. Karthik of Kumaraguru College of Technology, Coimbatore – 641 006, has successfully completed his project assignment titled GUIFRONT – END FRO POSTGRESS as a part of MCA Curriculum.

As a Project trainee, he started this project on December 19, 2007 and completed it on June 4, 2008.

Please note as per the company's policies and practices, the company retains ownership of the intellectual property rights concerning work undertaken during projects and disclosure of the source code and any other relevant information or data out of the organization is strictly prohibited. This document has to be solely used form the purpose of submitting as a project report in his institution only.

M. Karthik, designated as project trainee in Team Landmark- IT Systems and Solutions Pvt Ltd., will not be delivering the respective source code pertaining to his project to any person or organization or institution. They system design and architecture can't be used for any commercial purpose exceptby Team Landmrk-IT Systems and Solutions Pvt. Ltd., having their registered office in Chennai, INDIA.

4/08

PRAKASH KUMAR
DIRECTOR TECHNICAL
TEAM LAND MARK- IT SYSTEM AND SOLUTION Pvt. Ltd.,

Corp. Off :
167, 2nd Floor,

ABSTRACT

To design and develop a cross-platform application for remote administration and maintenance of the SQL SERVER database server. The tool must allow the user to visually create databases, tables, queries, views, functions, triggers, and other standard SQL components, allow for dynamic drag-n-drop relation-building, printing of entity-relationship diagrams, allow for backing up databases remotely and locally, execute queries on databases and show the results in-sort able tables with appropriate headers.

Data Base Analyzer for SQL SERVER is mainly used to create, browse databases and run SQL queries in an easy and powerful way. Advanced features include-visual query builder, report builder, backup/ restore the data in a database. It not only accelerates and makes the process of database creation clear but also simplifies database maintenance. It is a full-featured GUI SQL SERVER development tool. Features include a tree view of your databases, and database objects. This software is an user interactive query analyzer for database managers.

In most cases you can create your database objects / queries without touching your keyboard. With Syntax highlighting you can easily identify syntax errors. You can easily view your database objects. No more need to waste time writing complex queries on the system tables. With Syntax Completion and the ability to Drag and Drop tables and field names into the editor you can produce any SQL statement lighting fast.

ACKNOWLEDGEMENT

I would like to take this opportunity to say thank you to the people who have helped to make this project.

First I would like to express my heartfelt thanks to our principal **Dr.Joseph V.Thanikal, B.E., M.E., Ph.D., PDF., CEPIT.,** for having given me the opportunity to do this project .

I express my deep sense of gratitude and thanks to **Dr.M.Gururajan, Ph.D., HOD,** Department of Computer Application, for providing moral support towards this project work.

I am very much indebted to **Mr. A. Muthukumar M.Sc., M.C.A.**, Assistant Professor Kumaraguru College of Technology, Coimbatore for her complete assistance, guidance and support given to me throughout my project

I sincerely express my humble gratitude to **MR N.Jeyakanthan, M.C.A.,** Lecturer, Department of Computer Application, who has been my guide with valuable and holistic suggestions and extended kind of operation and encouragement.

I also express my deep sense of gratitude and thanks to **Mr .Prakash Kumar M.C.A,M.B.A,,** Project Leader, Team Landmrk-IT Systems and Solutions Pvt.Ltd., who has been my guide and gave valuable suggestions and encouragement.

I acknowledge my hearty thanks to all my beloved friends on their valuable co-operation in the proceeding of my work.

I wish to credit my special accordance to my parents for their encouragement and prayers to have a successful project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DDL	Data Definition Language
DML	Data manipulation language.
CLR	Common Language Runtime
SQL	Structured Query Language.
ODBC	Open Database Connectivity.

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

Postgress Database for SQL SERVER is mainly used to create, browse databases and run SQL queries in an easy and powerful way. Advanced features include-visual query builder, report builder, backup/ restore the data in a database. It not only accelerates and makes the process of database creation clear but also simplifies database maintenance. It is a full-featured GUI SQL SERVER development tool. Features include a tree view of your databases, and database objects. This software is an user interactive query analyzer for database managers.

The following are the modules of this project

- **Login**

  The mechanism by which callers prove that they are acting on behalf of specific users or systems. Login page answers the question, "Who are you?" using credentials such as username/password combinations.

- **Processing DDL**

  Data Definition Language (DDL) is a language for describing data and its relationships in a database. Data Definition Language is the portion used for defining the structure of the database and creating objects within the database.

- **Processing DML**

  The data manipulation language (DML) that allows you to add data to the database, modify it as necessary, and destroy it when it is no longer needed.

- **Administrator**

  The Administrator module is used to maintain the information of the client's logon information (login time, login name and logout time). He/she will also monitor there working contents and queries which had been executed by them.

1.2 ABOUT THE COMPANY

The collaborative style of working enables TEAM LANDMARK to transfer knowledge and skills into sound strategies and financial returns.

TEAM LANDMARK has made significant investments in physical, technological and telecommunications infrastructure to ensure adequate capacity to sustain growing business needs. TEAM LANDMARK has offices in India, Middle East and North America and shall soon be opening marketing offices in Europe.

TEAM LANDMARK has taken the lead to guide the business processes of digital communities, and has strategic alliances with HP, EPSON, IBM, BIOLINK, SYMBOL and several other large vendors, predominantly from the western world, specifically from Europe and the US.

TEAM LANDMARK has 3 divisions

- Enterprise Solutions Marketing
- Software Development Unit
- Applied IT Training

Solutions Marketing

This division has a professionally and technically equipped team, constantly working on providing IT solutions to match the exact requirements of the customers. It offers cost effective, quality solutions to overcome the client's operational bottle necks, by focusing on relationship building and nurturing the same.

Applied IT Training

The IT training division has made great inroads with Tie-ups for it's unique Applied IT courses in prestigious schools in Chennai such as Padma seshadri, St.Johns and Little Oxford. TEAM LANDMARK IT has tie up's with Leading Universities and Colleges (Engineering and Management) in India and offers Training programs under Industry Institute Interaction model. In this process of education, we impart both Technical and Domain knowledge to equip them to be a complete professional.

Software Development unit

Our service experience has taught us to listen and be responsive. By applying this simple concept to our software development process, we can help you meet your business objectives, while minimizing your time and investment. Our dedicated team and state of the art infrastructure enables us to cater to the needs of a broad spectrum of clients.

Our clientele list includes Lifestyle International (No 1 Retailers in India), ITC, St.Gobain, Hyundai, Coast Guard, RANE group of Companies to name a few. We have executed projects in various domain areas including, Retailing, Manufacturing and Logistics and more...

Services Offered

- Biometrics Solutions (Time and Attendance, Payroll Integration)
- Barcode – Automated Data Capture Solutions
- Mobile Computing (Stock Taking, Mobile POS, RF)
- Application Software (Retail, Medical Imaging, Hospital Management System, Hotel Management System, E Learning, CRM, Asset Management, Payroll, Logistics Management)

- Hardware and Maintenance
- Networking (LAN, WAN, Web Hosting, Storage Management)
- Web Designing and Web Programming
- Application Based Training (Call Centers, Bio Informatics)
- Infrastructure Support

We have specialized in Barcode Solutions for the various Industry verticals. We have executed and Implemented Data Capture Solutions to major Industries in Chennai and Bangalore. Rane, Hyundai, St.Gobain, Lifestyle to name a few. We also deal with Assembled, Branded Machines (IBM, Compaq), Barcode Scanners, Hand Held Devices (Symbol, Cipher labs), Software Licenses (Microsoft, Oracle). Barcode Printers (Monarch). other hardware accessories, and peripherals (Printers, Scanners) and we provide an excellent installation and maintenance support.

CHAPTER 2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

A **back-end database** is a database that is accessed by users indirectly through an external application rather than by application programming stored within the database itself or by low level manipulation of the data (e.g. through SQL commands).

A back-end database stores data but does not include end-user application elements such as stored queries, forms, macros or reports.

The term back-end database is most widely used among developers using small database programming systems which can contain the end-user application programming within the database as a single item. The most common of these is Microsoft Access.

The developer must decide whether to include the application programming with the data in a single database, or whether to separate them into two database files, according to the client-server model.

For simple database applications, it is common for all programming to be stored with the data. This results in a single file and is easier to develop at the expense of scalability and concurrency.

3.2 PROPOSED SYSTEM

For more advanced database applications it is common to split the data and the programming parts in to a *front-end database* and a back-end database where the front-end holds all the application programming. This has advantages in terms of scalability, performance and concurrency but requires greater effort on the part of the developer. In the long term it may be easier to maintain and upgrade as new versions of the front-end can be deployed independently of the back-end database. The front and back-end databases do not always have to be of the same types. For example, it is possible to use a Microsoft SQL Server front-end with a SQL SERVER database back-end.

CHAPTER 3
SOFTWARE REQUIREMENTS

3.1 SOFTWARE CONFIGURATION

Operating System : Windows 98/XP/2000

Front-End : VB.NET

Back-End : SQL SERVER

3.2 HARDWARE CONFIGURATION

Hard Disk	:	10 GB
RAM	:	256 MB
Processor	:	PENTIUM III
FDD	:	1.44 MB
CD-Drive	:	52X SAMSUNG

3.3 SOFTWARE CONFIGURATION

VISUAL STUDIO.NET

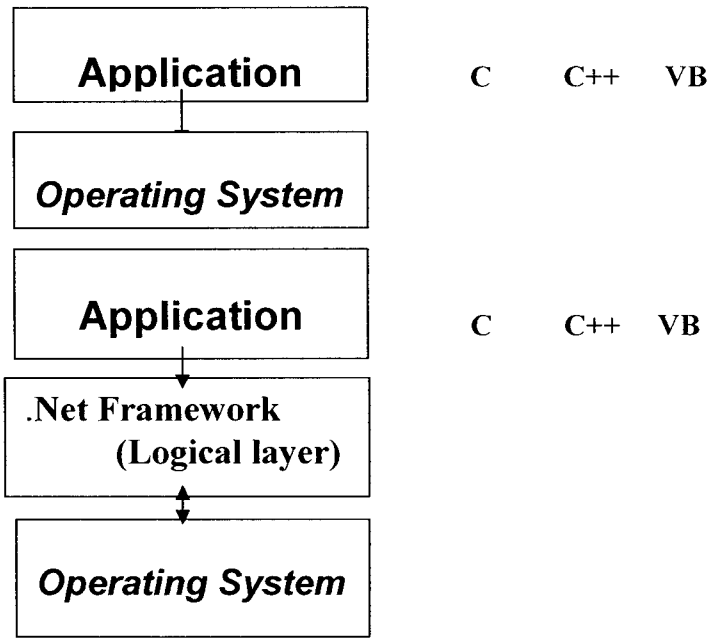
Visual Studio .NET is a complete set of development tools for building ASP Web applications, XML Web services desktop applications, and mobile applications. Visual Basic. NET, Visual C++.NET, and Visual C#.NET all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions. In addition, these languages leverage the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web services.

The following are some features of Visual Studio .NET.

- All the languages in .NET share a common environment. If you are familiar with the tools of VB then you can easily move around in C++.
- The Common Language Environment (CLR) manages all code and components at runtime and makes it easy to create Multilanguage projects.
- Previously all languages have their own libraries in .NET all languages shares a common runtime library
- All languages have their own compiler. In order for they're to interoperability between two different languages, the compiler must share some common ground. This is the job of Common Language Specification
- Common Type Specification, which creates a set of universal data types, both primitive and complex, that can now be communicated between two modules written in two modules.
- The Project Group is called as Solution in .NET. A solution file ends with .sln extension that is generated automatically when a project is created. A solution can contain projects written in different languages

- The Major building block of .NET is Assemblies. Versioning and security are set at this level. An Assembly can be one or more items, such as PEs, Graphics, or Multimedia.

Why .NET?



.NET Framework

The .NET Framework is a multi-language environment for building, deploying, and running XML Web services and applications. .NET is a new Microsoft development platform that allows developers to quickly create web-ready applications in whatever programming language they choose. It is also called as a Device-to-Device independent platform. .NET is aimed at accelerating the next generation of distributed computing. .NET is a "revolutionary new platform, built on open Internet protocols and standards, with tools and services that mix computing and communications in new ways".

The .NET Framework is the infrastructure for the new Microsoft .NET Platform. The .NET Framework is a common environment for building, deploying, and running Web applications and Web Services.

The .NET Framework contains a common language runtime and common class libraries - like ADO .NET, ASP .NET and Windows Forms - to provide advanced standard services that can be integrated into a variety of computer systems.

The .NET Framework provides a feature-rich application environment, simplified development and easy integration between a number of different development languages.

The .NET Framework is language neutral. Currently it supports C++, C#, Visual Basic, and JScript (The Microsoft version of JavaScript). Microsoft's Visual Studio.NET is a common development environment for the new .NET Framework.

It consists of three main parts :

- Common Language Runtime Despite its name, the runtime actually has a role in both components' runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and stopping threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have on other components. At development time, the runtime role changes slightly; because it automates so much (for example, memory management), the runtime makes the developer experience very simple, especially when compared to COM as it is today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.

- Unified programming classes the framework provides developers with a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes and Java developers use the Windows Foundation Classes. The framework unifies these disparate models and gives Visual Basic and Jscript programers access to class libraries as well. By creating a common set of APIs across all programming languages, the common language runtime enables cross-language inheritance, error handling, and debugging. All programming

languages, from Jscript to C++, have similar access to the framework and developers are free to choose the language that they want to use.

- Components of .NET
  - Common Language Runtime (CLR)
  - .NET Framework class library

CLR (Common Language Runtime)

Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and stopping threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have on other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management), the runtime makes the developer's experience very simple, especially when compared to COM as it is today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component. It controls interaction of code with the OS.

Exception Handling

Security

Debugging

Managed Code executed by CLR

Unmanaged Code not executed by CLR

Bypass .NET APIs and make direct calls to OS.

Services

Garbage collection

Memory management

CLR converts

Code->Microsoft Intermediate Language

MSIL

CPU independent.

Instructions for

1. Loading
2. Storing
3. Initializing
4. Calling Methods

Instructions about

1. Arithmetic and Logical Operators
2. Exception Handling

When any code is compiled in addition to MSIL "metadata" is created and stored with the compiled code.

Metadata

Members

Types

References in the code

CLR use Metadata to

Locate and load classes

Generate native code

Provide Security

Execute code

The compiled file, which contains MSIL and metadata, is known as **Portable Executable** (PE) files. Before execution MSIL is converted to native code.

Just-In-Time (JIT) compiler

Code->MSIL->Native Code

It uses Metadata to convert MSIL into native code. Native code is CPU dependent. Various JIT compilers are available for different architectures. Depending on the JIT used we can execute MSIL on different architectures.

Common Type System (CTS)

CLR defines CTS which is a standard type system used by all CLR compatible languages. CTS lists the types supported by CLR,

Classes
Interfaces
Value Types

We can pass a variable from a class in any CLR compatible language to any other CLR compatible language. Because data types have the same implementation across those languages.

Class Library

It includes

- Multiple
- Reusable
- Object-Oriented
- Extensible Classes.

It is integrated with the CLR and common to all Programming languages in Visual Studio.NET. Class library provides a set of classes across programming languages. We can use these classes and create new classes.

We can use these classes to

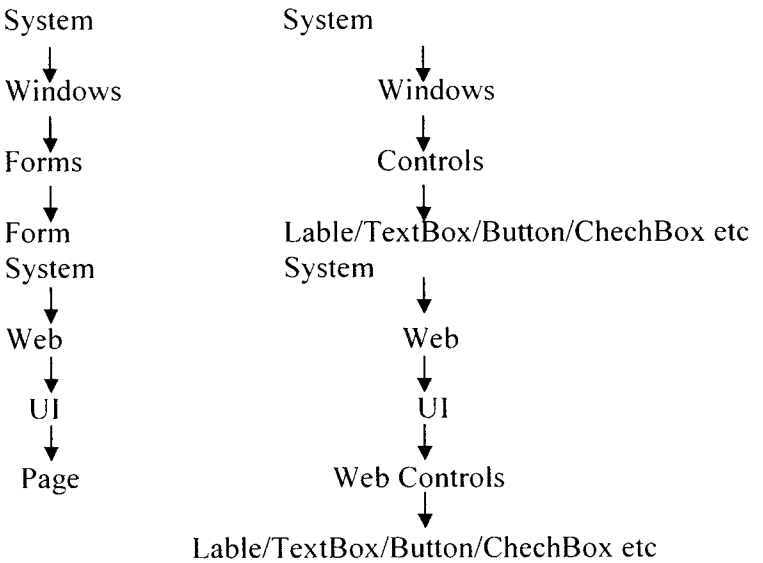
Develop an application.

Implement cross-language inheritance.

The classes are organized hierarchically.

"System" namespace

- Microsoft.C#
- Microsoft.Jscrpit
- Microsoft.VisualBasic
- System.XML, System. Text, System.Web, System.Windows



Variable Scopes

- **Private**
- **Protected**
- **Public**
- **Friend**
- **Shared**
- **Static**

Variable Defaults

- **Numbers** : **0 (Zero)**
- **Boolean** : **False**
- **String** : **Nothing**
- **Object** : **Nothing**
- **Date** : **12:00:00 AM**

TYPE CONVERSION

- **Widening** :Conversions is able to maintain the original Data Value, without data loss.
- **Narrowing** :Conversion attempts to convert data from a larger type to a smaller type (in bytes or precision) that may not be able to Maintain the original value.

Built-in Type conversion functions

CBool, CByte, CChar, CDate, CDbl, CDec, CInt, CLng, CShort, CSng, CStr, CObj, CType

Features of .NET

In addition to hiding script commands, ASP.NET has the following advanced features that help develop robust web application.

- Compile code
- Enriched tool support
- Power and flexibility
- Simplicity
- Manageability
- Scalability

Compile code

Code written in ASP.NET is compiled not interpreted. This means ASP.NET applications faster to execute than other server-side scripts that are interpreted, such as scripts written in a previous version ASP.

Enriched tool support

The ASP.NET framework is provided with a rich toolbox and designer in VS.NET IDE (Visual Studio .NET Integrated Development Environment). Some of the futures of the powerful tool are the WYSIWYG (What You See Is What You Get) editor, drag-and-drop server controls, and automatic deployment.

Power and flexibility

ASP.NET application are base on Common Language Runtime(CLR). Therefore, the power and flexibility of the .NET platform is available to ASP.NET application developers. ASP.NET applications enable you to ensure that the .NET Framework class library, messaging, and data access solutions are seamlessly accessible over the Web. ASP.NET is also language-independent. Therefore, you can choose ant .NET language to develop your application.

Simplicity

ASP.NET enables you to build user interface that separate application logic from presentation content. In addition, CLR simplifies application development by using managed code easy to perform common tasks ranging from submission and client authentication to site configuration and development.

Manageability

ASP.NET enables you to manage Web applications by storing the configuration information in an XML file. You can open the XML file in the Visual Studio .NET IDE.

Scalability

ASP.NET has been designed with scalability in mind. It has features that help improve performance in a multiprocessor environment.

Security

ASP.NET provides a number of options for implementing security and restricting user access to a Web application. All these options are configured within the configuration file.

SQL SERVER

Microsoft SQL Server 2000 is a relational database engine that can store, retrieve, analyze, and manage large amounts of data. SQL Server meets the availability, reliability, and scalability requirements of demanding e-commerce data storage requirements. Built-in tools provide extensive database programming and administration capabilities. These include rapid development using Visual Database Tools, data transformation, data mining and multidimensional analysis, interactive debugging, workload monitoring, and performance tuning. To use the SQL Server .NET Data Provider, you must have access to Microsoft SQL

Server 2000. SQL Server .NET Data Provider classes are located in the System.Data.SqlClient namespace. For earlier versions of Microsoft SQL Server, use the OLE DB .NET Data Provider with the SQL Server OLE DB Provider (SQLOLEDB).

FEATURES OF SQL SERVER 2000

- **Internet Integration**

The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

- **Scalability and availability.**

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows98 through large, multiprocessor servers Microsoft Windows 2000 Data center Edition. SQL Server 2000 Enterprise Edition Supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- **Enterprise – Level Database Features.**

The SQL Sever 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Sever 200 distributed queries allow to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows to also maintaining multiple copies of data, while ensuring that the separate copies remain synchronized. Set of data can be replicated to multiple, mobile,

disconnected users have them work autonomously, and then merge their modifications back to the publisher.

- **Ease of installation, deployment, and use**

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Sever applications that customers can implement with a minimum of installation and administrative overhead.

- **Data warehousing:**

SQL Server 2000 includes tools for extracting and analyzing summary data for online analytical processing. SQL Sever also includes tools for visually designing databases and analyzing data using English – based questions.

SQL Server 2000 provides two fundamental services to applications in a Windows DNA environment.

- The SQL Server 2000 relational database engine is a modern, highly scalable, highly reliable engine for storing data. The database engine stores data in tables. Each table represents some object of interest to the organization, such as vehicles, employees, or customers. Each table has columns, that represent an attribute of the object, modeled by the table (such as weight, name, or cost), and rows that represent a single occurrence of the type of object modeled by the table (Such as the car with license plate number ABC-123, or the employee with ID 123456) Applications can submit Structured Query Language (SQL) statements to the database engine, which returns the results to the application in the form of a tabular result set. The specific dialect of SQL supported by SQL Server is called Transact – SQL. Applications can be also submit either SQL statements of Xpath queries and request that the database engine returns the results in the form of an XML document.

Application support

Both the relational database engine and Analysis Services provide native support for the common Windows DNA or Win32 data access interfaces, such as ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC). Applications can use any of these application programming interfaces (APIs) to send SQL or XML statements to the relational database engine using a native OLE DB provider or ODBC driver. SQL Server 2000 also introduces the ability to use HTTP to send SQL or XML statements to the relational database engine. Applications can use the multidimensional extension of either ADO or OLE DB to send Multidimensional Expressions (MDX) queries to Analysis Services. Because SQL Server uses the standard Window DNA data access APIs, the development of SQL Server applications is well supported by the Microsoft application development environments. In addition, interactive query tools, such as Query Analyzer, provide templates, interactive debuggers, and interactive test environments that speed the ability of the programmers to deliver SQL Sever applications.

SQL SERVER

It is an object-relational database management system (ORDBMS). It is released under a BSD-style license and is thus free software. As with many other open-source programs, SQL SERVERQL is not controlled by any single company, but relies on a global community of developers and companies to develop it.

SQL SERVERQL's unusual-looking name makes some readers pause when trying to pronounce it, especially those who pronounce SQL as "sequel" (Audio sample, 5.6k MP3). It is also common to hear it abbreviated as simply "SQL SERVER", which was its original name. Because of ubiquitous support for the SQL Standard amongst all relational databases, the community considered changing the name back to SQL SERVER. However, the SQL SERVERQL Core Team announced in 2007 that the product would continue to be named SQL SERVERQL. The name refers to the project's origins as a "post-Ingres" database, the original authors having also developed the Ingres database.

CHAPTER 4

SYSTEM DESIGN

Reviewing the study phase activities and making decisions about which functions are to be performed by the hardware, software, and Human ware started Design phase. The input and output file design for each of the programs was done. Finally, the generalized system was explained to the management to approval.

The steps involved in designing phase were

1. The function to be performed is identified
2. The input, output, and file design id performed.
3. The system and component cost requirements is specified.
4. The design phase report.

4.1 INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter data in to the computer as accurate as possible. Here input are designed effectively so that errors made by the operations are minimized. The input to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using.

4.2 OUTPUT DESIGN

The output design was done so that results of processing could be communicated to the users. Stored details in the database are retrieved according to the users query.

4.3 DATA FLOW AND USE CASE DIAGRAM

A Data Flow Diagram is used to define the flow of the system and the resources such as information. It is the way of expressing system requirements in a graphical manner. It is also known as bubble chart. It consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flow in the system.

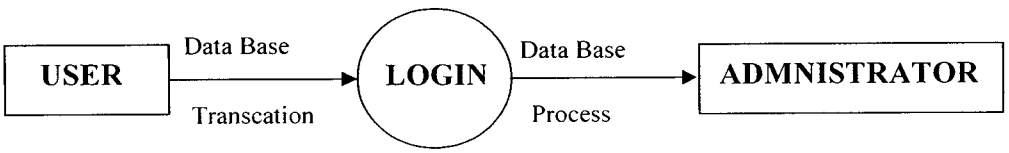


Figure 4.1 Level 0 DFD

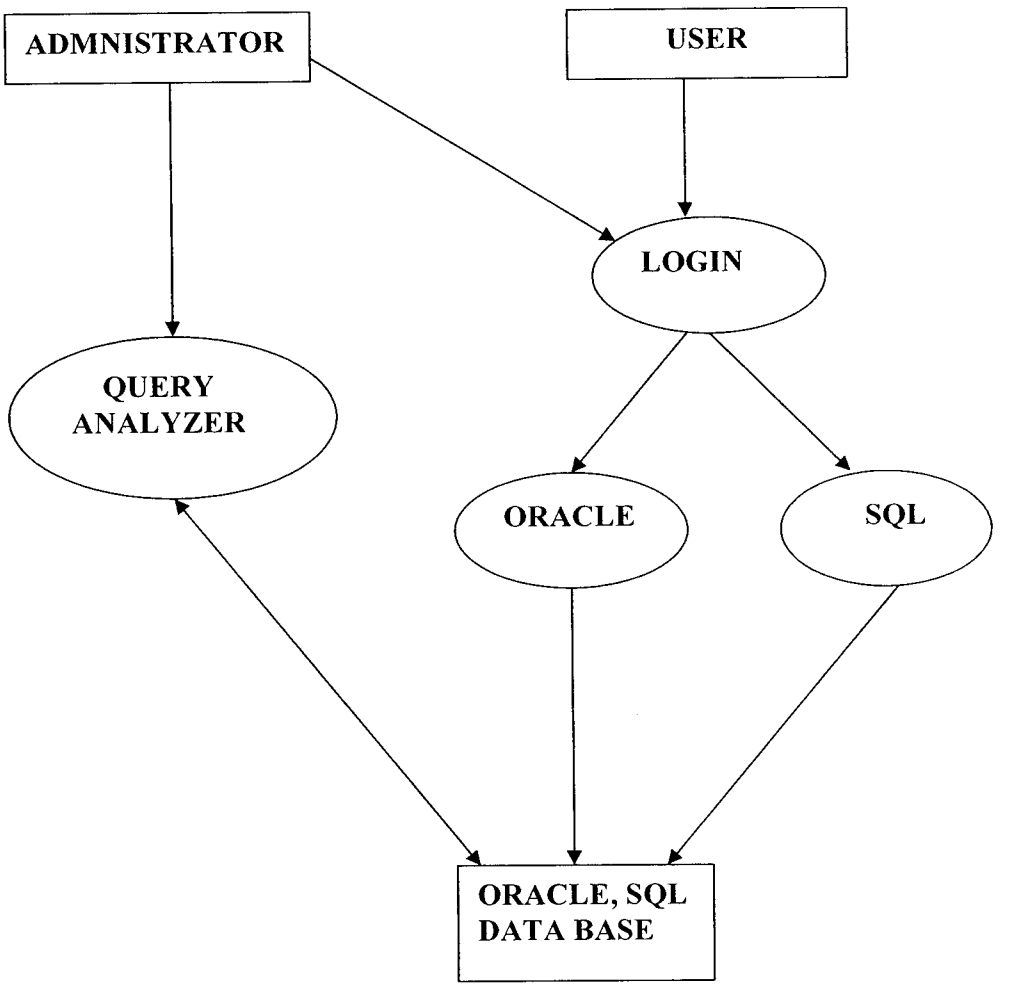


Figure 4.2 Level 1 DFD

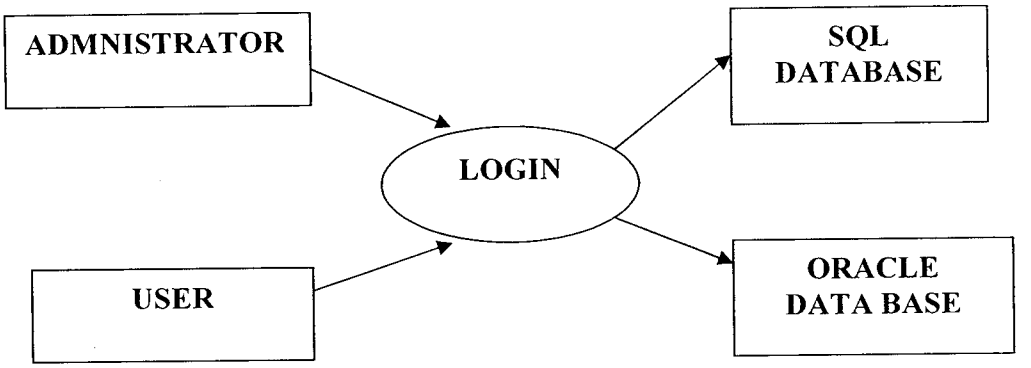


Figure 4.3 Level 2 DFD

4.4 USECASE DIAGRAM

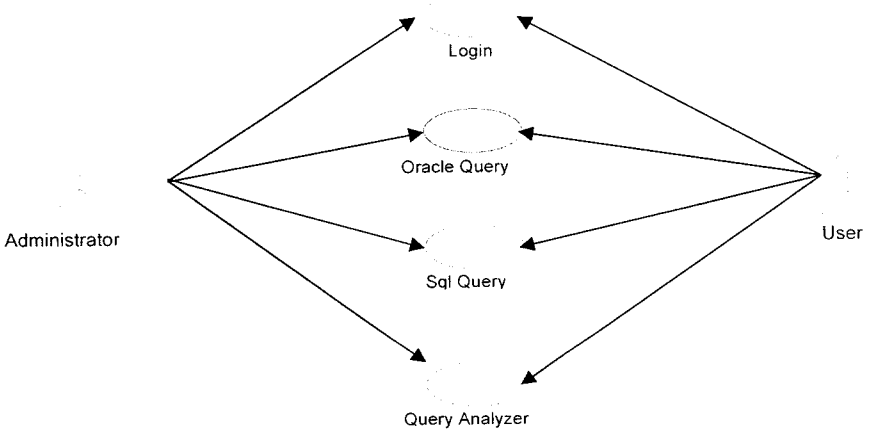


Figure 4.4 Use case diagram for GPG

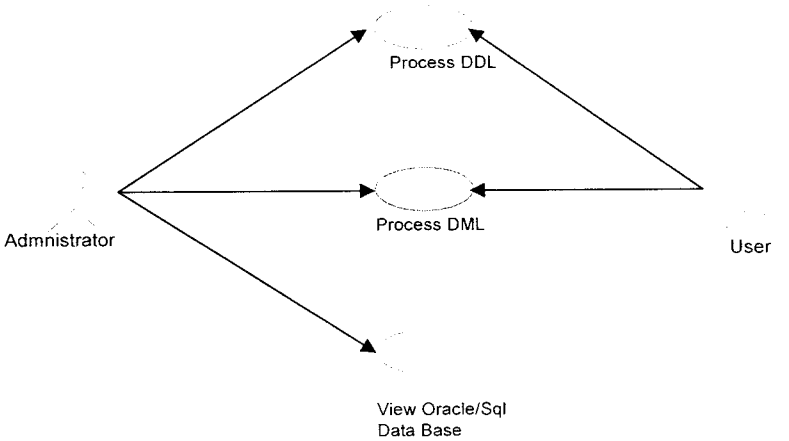


Figure 4.5 Use case diagram for user role

4.5 MODULE DESCRIPTION

Module 1: Login:

Login module is the mechanism by which callers prove that they are acting on behalf of specific users or systems. Login page answers the question, "Who are you?" using credentials such as username/password combinations.

In username/password authentication, a user ID and password are requested from the user and sent to Server. Server checks the information and if it is trustworthy, grants access for the user. If the Server denies the user he/she couldn't asccess the database.

Different account types require different levels of security. Different transactions are riskier than others. Different customers prefer different types of protection. In this project we use the form authentication for security purpose.

Module 2: Processing DDL

Data Definition Language (DDL) is a language for describing data and its relationships in a database. Data Definition Language is the portion used for defining the structure of the database and creating objects within the database. The Data Definition Language (DDL) is used to create and destroy databases and database objects.

Tables are the most commonly used objects in relational databases. They hold the actual data in rows (records) and columns (fields). In many user applications, the tables that you need have already been created for you, whereas other situations require that you create your own tables.

DDL uses the following three commands to work with objects in the database:

1. CREATE: Creates a new object
2. USE: Database wish to work
3. ALTER: Modifies an existing object
4. DROP: Removes an object

These commands are used to work with several types of objects in the database, including tables, indexes, and views.

We can generate DDL in a script for database objects to:

- Keep a snapshot of the database structure
- Set up a test system where the database acts like the production system but contains no data
- Produce templates for new objects that you can create based on existing ones.

When you generate DDL, you can use the DDL statements to recreate everything about a database except for its contents. You can generate the DDL to completely recreate the database, or choose to recreate only certain aspects of it, such as its current statistics. Through this we can also limit the statements that are generated so that only a segment of the database is recreated, for example, the statistics for a subset of tables.

Module 3: Processing DML

The data manipulation language (DML) that allows you to add data to the database, modify it as necessary, and destroy it when it is no longer needed.

DML consists of three statements:

1. INSERT: add records to an existing table
2. UPDATE: modify information contained within a table
3. DELETE: remove records from Existing Table
4. Select: retrieve the specific information

The Data Manipulation Language (DML) is used to retrieve, insert and modify database information. These commands will be used by all database users during the routine operation of the database.

Procedures are extremely useful functions that are stored inside a database. Procedures can perform a series of any Data Manipulation Language commands. In addition to standard SQL, procedures support decision construct equivalent to an `if` or `switch` statement, and looping constructs equivalent to a `while` statement. Procedures have several benefits over SQL code:

- Procedures can be written to optimize Database.
- Procedures are pre-interpreted, which makes them faster than creating, and sending it to the database to be interpreted.

- Procedures perform the work at the server, thereby reducing network traffic.
- Procedures allow modular reuse of protested code. This enforces consistency, improves performance, and reduces programmer error.
- Procedures can automate complex logic, allowing complex operations to be examined only once.

Procedures are executed using either the CALL or EXECUTE command, depending on the database. CALL and EXECUTE are Data Manipulation Language commands that allow the user to execute the procedure.

Module 4: Administrator

The Administrator module is used to maintain the information of the client's logon information (login time, login name and logout time). He/she will also monitor there working contents and quires which had been executed by them. All these activities are monitored by the Administrator and stored for the future reference.

CHAPTER 5
SYSTEM TESTING

5.1 TESTING METHODOLOGIES

System testing is a stage of implementation. This is to check whether the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests; online response, volumes, stress recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of the system design. In the real sense it is the phase, which helps to bring out the creativity of the other phase, and make it shine. Here are several rules that can serve as testing objectives. They are

- Testing is a process of executing a program with the intent finding an error.
- A good test case is one that has a high probability of finding undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Software testing is an important element of software quality assurance and presents the ultimate review of specification, design and coding.

- **White box testing**
- **Black box Testing**
- **Unit Testing**
- **Integration Testing**
- **Security Testing**
- **Validation**

- **User Acceptance Testing**
- **Code module Testing**
- **System Testing**

WHITE BOX TESTING

White-box testing methods can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true or false sides
- Execute all loops at their boundaries and within their operational bounds
- Exercise internal data structures to assure their validity.
- Providing test cases that exercise specific sets of conditions and/or loops tests logical paths through the software.

BLACK BOX TESTING

Black box testing focuses on the functional requirements of the software. This testing attempts to find errors in the following categories.

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external data base access.
- Performance errors.
- Initialization and Termination errors.

UNIT TESTING

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The test is carried out during the programming stage it self. In this test, each module is found to be working satisfactory as regards to the expected output from the module.

INTEGRATION TESTING

Data can be lost across an interface. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as whole.

SECURITY TESTING

This testing is to ensure that unauthorized user penetrates the system for making the data inconsistent.

VALIDATION TESTING

At the culmination of integration testing software is completely assembled as package. Interfacing errors have been uncovered and corrected and final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by customer.

After validation test has been conducted, one of the three possible conditions exists.

- The function or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created.
- Proposed system under consideration has been tested by using validation test and found to be working satisfactory

Numeric Validation Testing :

Each field in the form are checked whether they have been types with numeric values. Otherwise exceptions are raised using the JavaScript language.

Non Numeric and Data Validation Testing:

All the dates entered are validated. If other than non-Numeric value types, exceptions are raised.

USER ACCEPTANCE TESTING

Acceptance Testing involves planning and execution of functional test, performance tests and stress tests in order to demonstrate the implemented system satisfies its requirements. In user acceptance testing, the network monitoring system, the user is fully satisfied, because of its excellent functions and performance in the network monitoring system.

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required. This done in regard to the following

- Input screen designing
- Output screen designing

CODE MODULE TESTING

In this all the code modules were tested in individual one after the other. The following were testing in all modules.

- Loop testing.
- Boundary value analysis.
- Equivalence positioning testing.

When a program is tested, all the error conditions were checked to see if they are handled properly. Breaking the program down into self-contained portions, each of which can be checked at certain key points, facilitates the process.

The key fields (e.g. Units codes for unit Master) for each master entry against duplication, and relevant data are modified when required while deletion, the records are deleted only in the form level can be retrieved if the undo button is clicked. If deleted and saved the data cannot be retrieved.

When a program is tested, the actual output with the expected output is verified. When there is discrepancy, the sequence of instructions must be traced to determine the problem. Breaking the program down into self-contained portions, each of which can be checked at certain key points, facilitates the process. The idea is to compare program values against desk-calculated values to isolate the problem.

PROGRAM TESTING

There are three ways to test a program

1. For correctness
2. For implementation, efficiency and
3. For Computations complex city.

Test for correctness is supposed to verify that a program does actually what it is designed to do. This is much more difficult than it may appear at first, especially for large programs.

Test for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code Refining process, which re-examines the implementation phase of algorithm development.

Test for computational complexity amount to an experimental analysis of the complex city of an algorithm or a experimental comparison of two or more algorithms, which solve the same problem.

System Testing

System testing tests the integration of each module in system. The objective while testing the system was test the discrepancies between the system and original objective depends on its design, development and testing and implementation.

Test Data

Taking various kinds of the test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested with it. While testing the system errors are uncovered and corrected. Corrections are noted for the future use.

Test Plan

For successful software, testing strategy to be implemented product requirements should be specified in a quantifiable manner long before testing commences. Although the objective of testing is to find errors, a good testing strategy also accesses other quality characteristics such as probability, maintainability, and usability.

Test plans have to consider the following: understand the users of the software and develop a profile for each user category, develop a testing plan that emphasizes rapid cycle testing.

The test data was prepared with the under mentioned objectives

- To test if all the features provided in the module perform satisfactorily.
- To ensure that the process of testing is a realistic possible.
- To check if on receipt of erroneous data, appropriate error message are Generated.

CHAPTER 6
IMPLEMENTATION AND FUTURE ENHANCENETS

IMPLEMENTATION

Implementation means the process of converting a new or a revised system design into an operational one. It is the most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. In this phase, we can build the components either from scratch or by composition. Given the architecture document from the design phase and requirement document from the analysis phase, we can build exactly what has been requested.

This phase deals with issues of quality, performance, baselines, libraries and debugging. The end deliverable is the product itself. There are three types of implementation:

1. Implementation of a computer system to replace a manual system
2. Implementation of new computer system to replace an existing one.
3. Implementation of a modified application to replace an existing one, using the same computer.

Implementation of "GUI Front End for Postgress" comes under Third category. At the end of the specific period, the system performance and the reliability are tested. Implementation is the key stage in achieving a successful new system because it involves.

FUTURE ENHANCEMENTS

It is necessary to keep up with changing user needs and the operational environment. Normally software fails because of improper cumulative maintenance, wear and tear. The system can be handled separately with out affecting other parts of the system. Thus, future enhancements are very easy in this system. Since the system is developed using modularized design, it can be upgraded without much modification.

CHAPTER 7
CONCLUSION

Working over the project, "GUI Front End for Postgress", has been a great experience with a lot of exposure to various evolving software trends. The project has been found to work effectively and efficiently. It clearly gives the client a competitive advantage tool that would help improve the business' financial bottom line.

The application is formulated by analyzing the requirements of the end user. Each and every module has undergone various test conditions. With a full stretch testing, it has been ensured that the system can enhance ideally without any bugs or crashes, which will make the end user more compatible with the project.

The application is designed as user friendly and all the options available are clear and self explanatory so that the user can understand the system easily.

APPENDIX 1

TABLE DESIGN

Field Name	Description
Username	Varchar
Password	Varchar
Database	Varchar
SQL Sever	Varchar

Table A 1.1 Login

Table name : Login table

Description : This table contains all the details about the users.

Field Name	Description
Login Name	Varchar
Login Time	Date/Time
Logout Time	Date/Time

Table A 1.2 Admin

Table name : Administrator

Description : This table contains all the details about the login users.

APPENDIX 2

SCREEN SHOTS

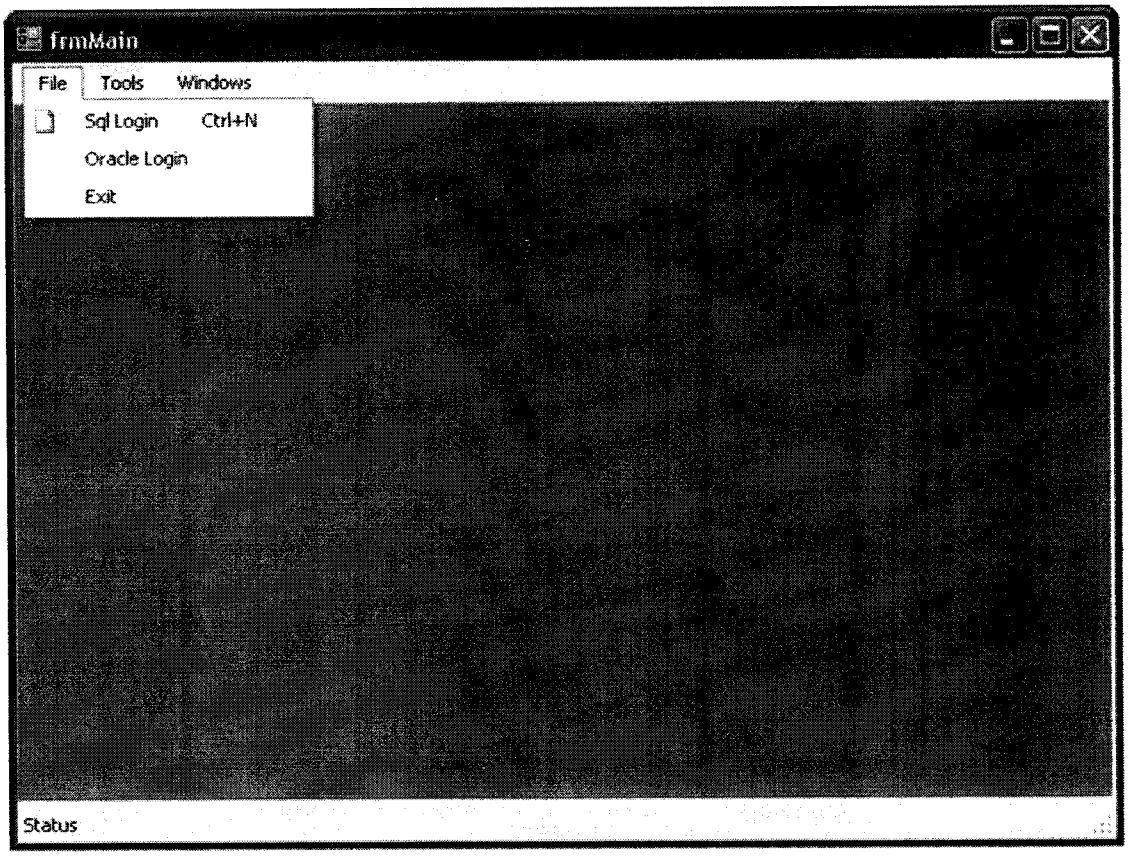


Figure A 2 .1 Main

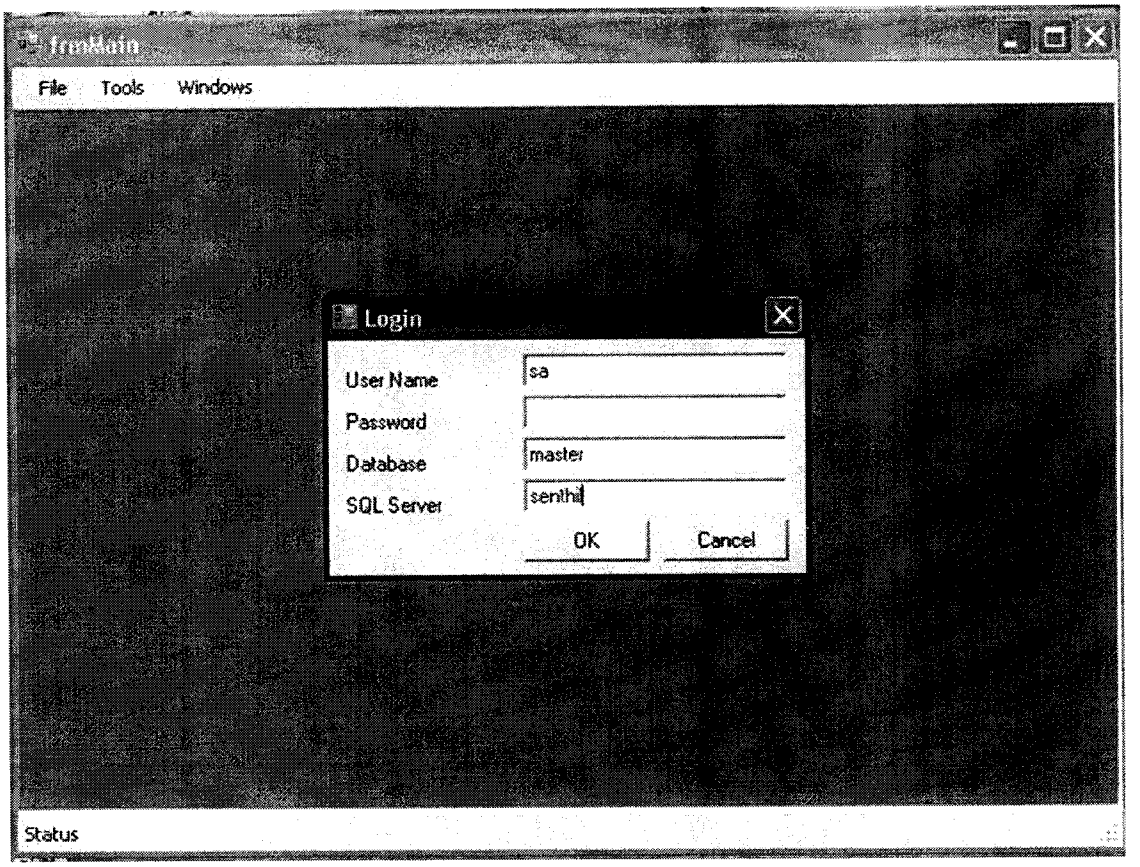


Figure A 2 .2 Login

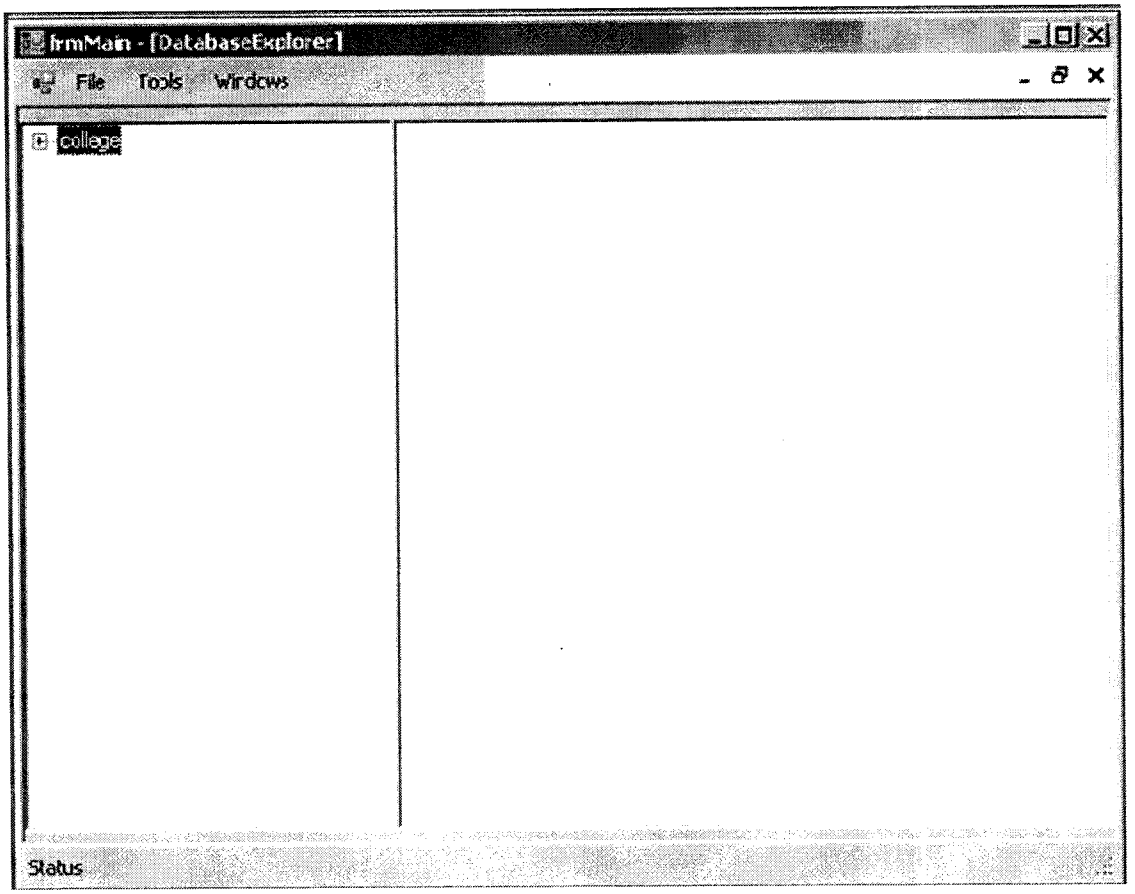


Figure A 2 .3 Tables

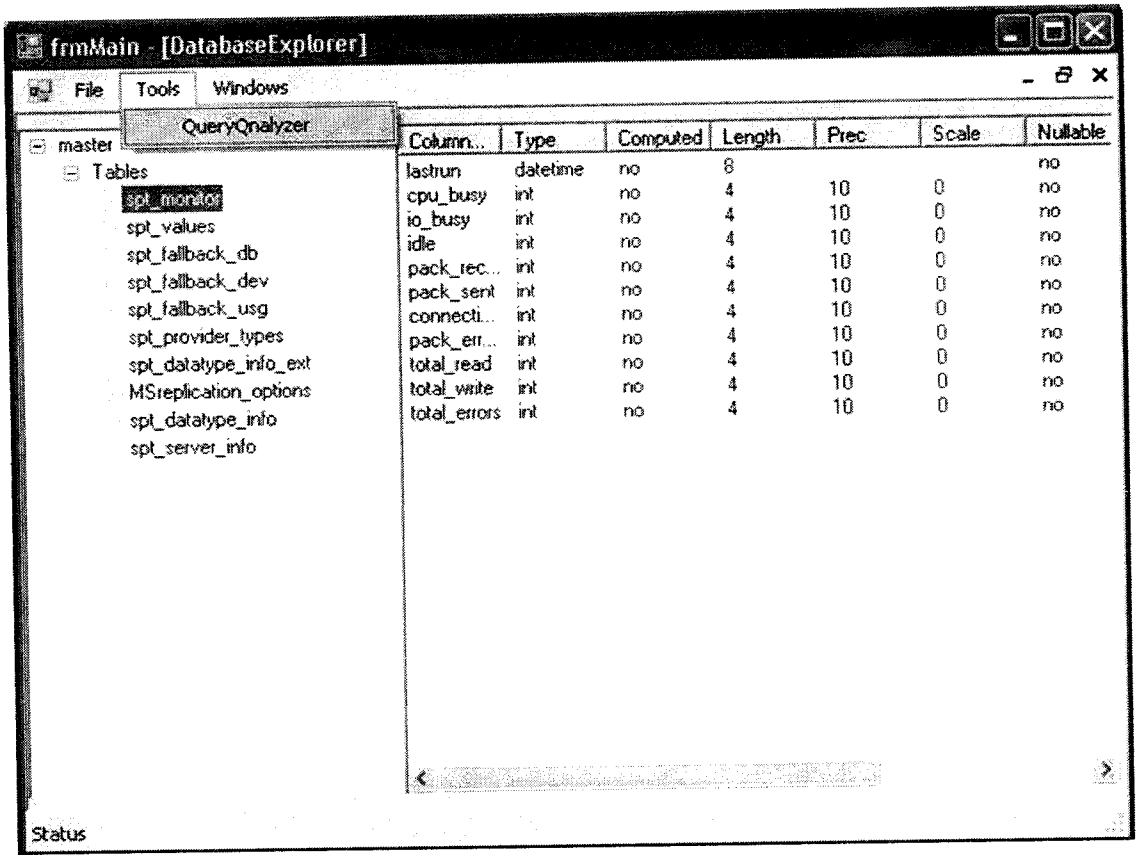


Figure A 2 .4 Tables Specification

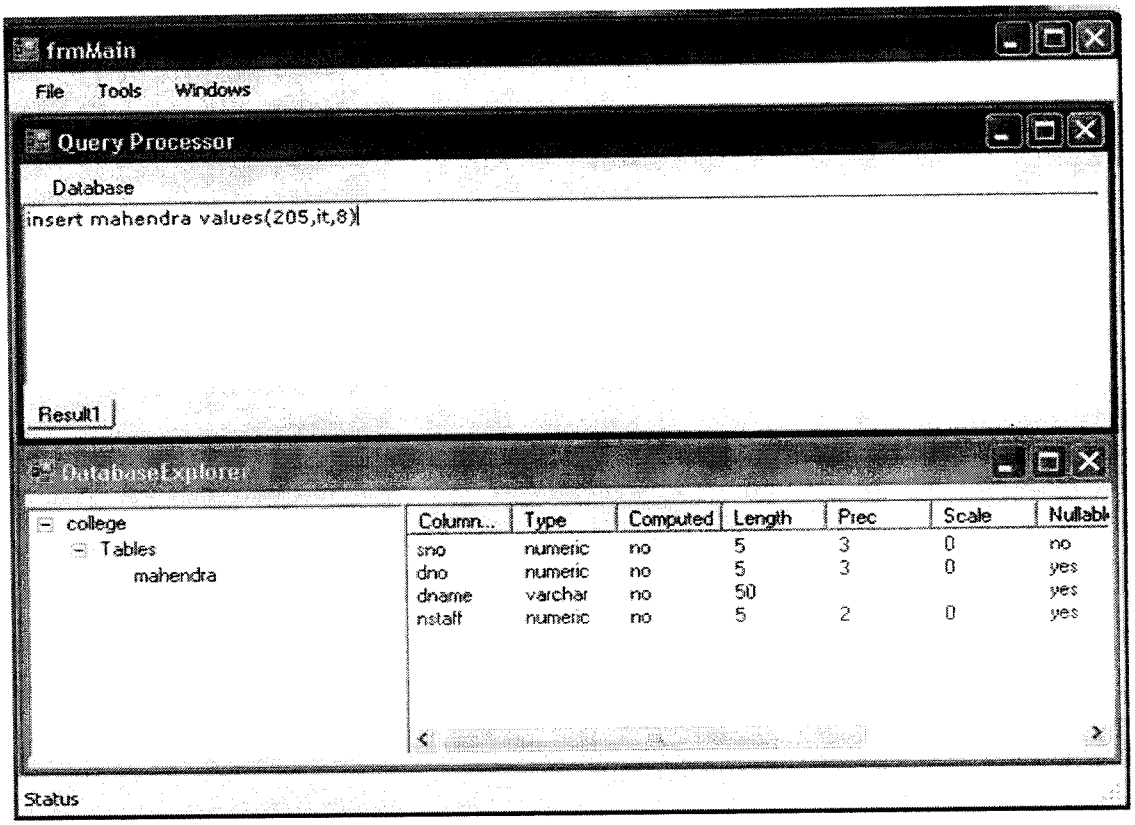


Figure A 2 .5 SQL Commands Prompts

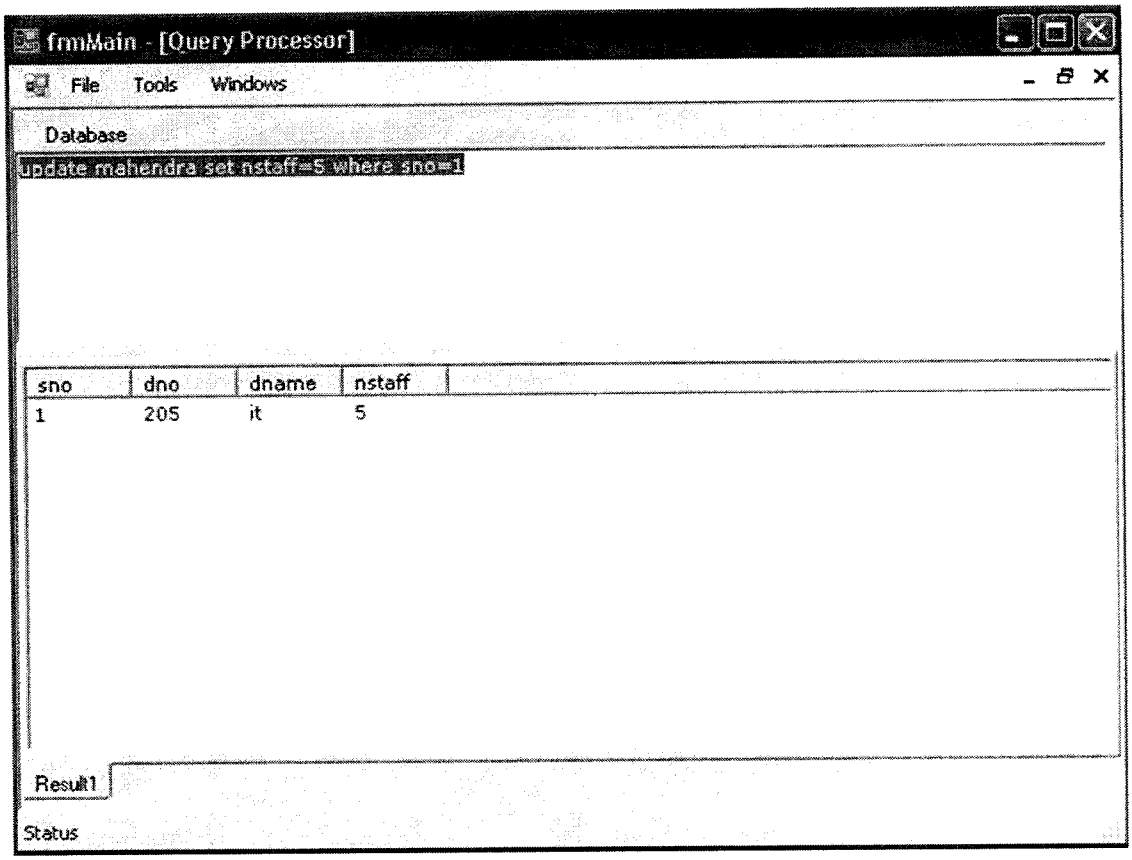


Figure A 2 .6 Insert Query

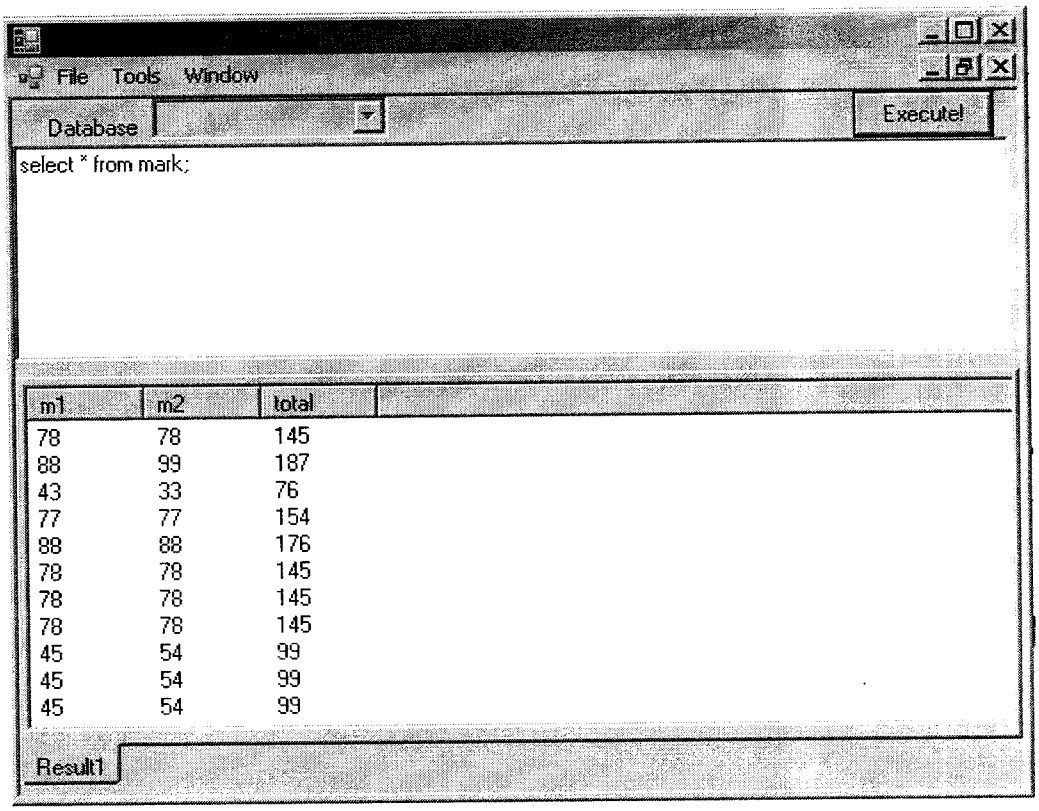


Figure A 2 .7 View query

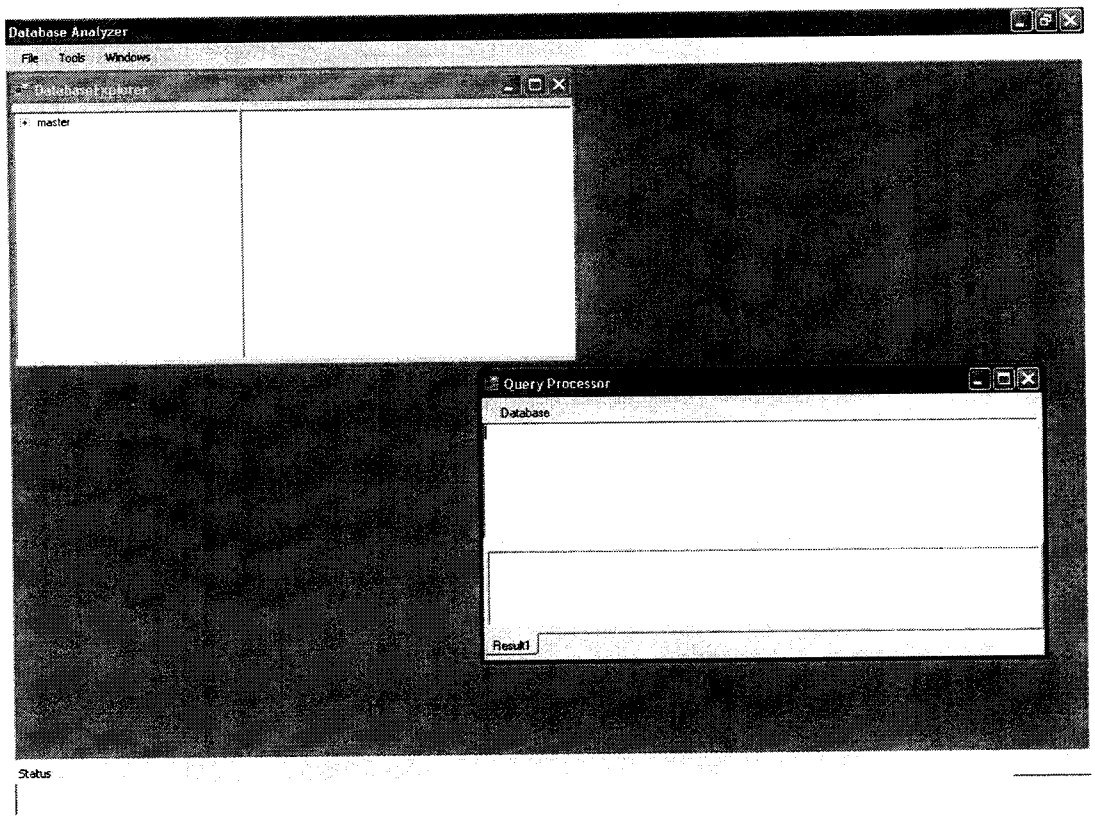


Figure A 2 .8 Query Processor

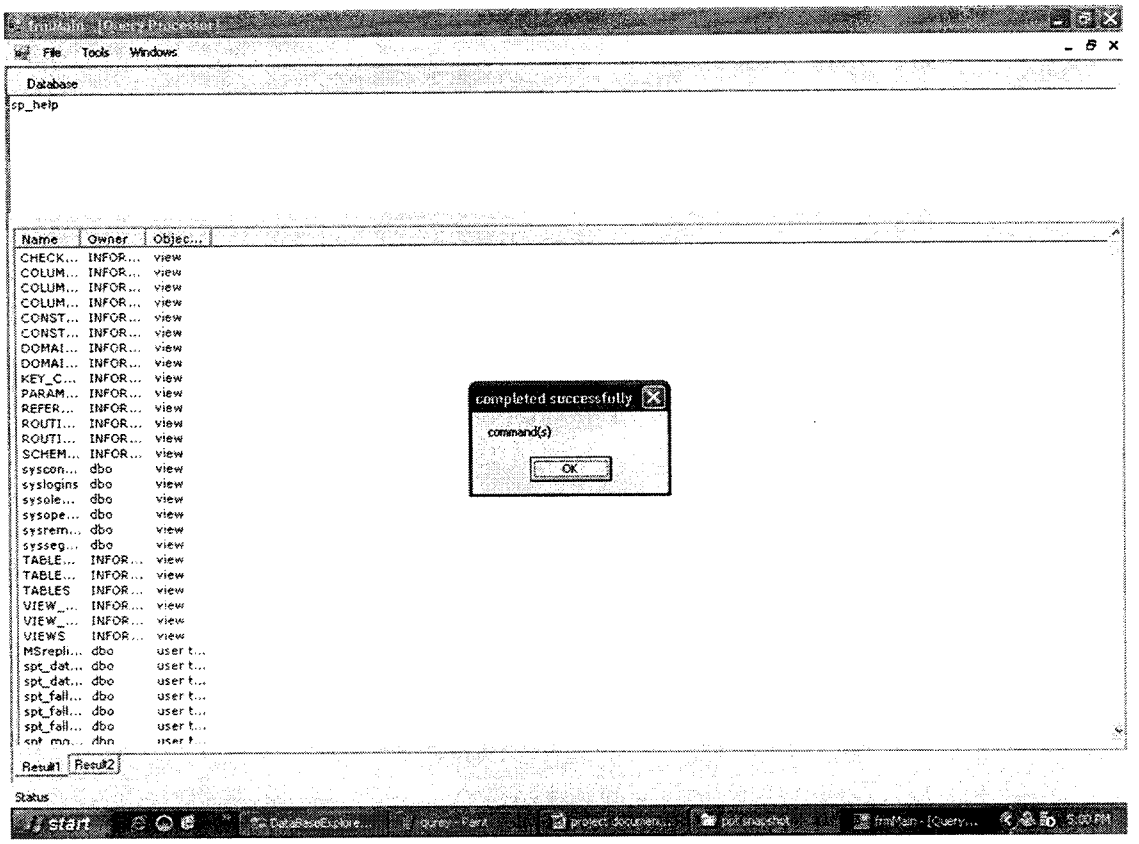


Figure A 2 .9 Success Message

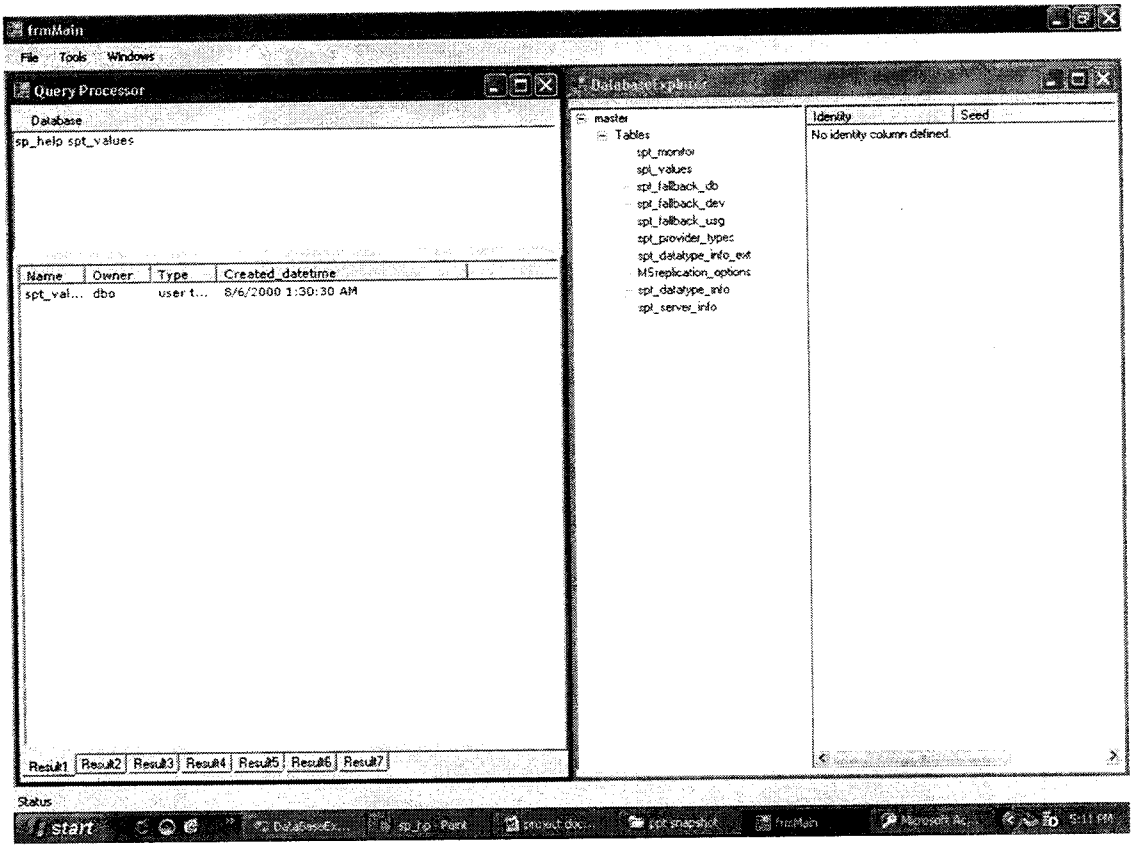


Figure A 2 .10 Database Explorer

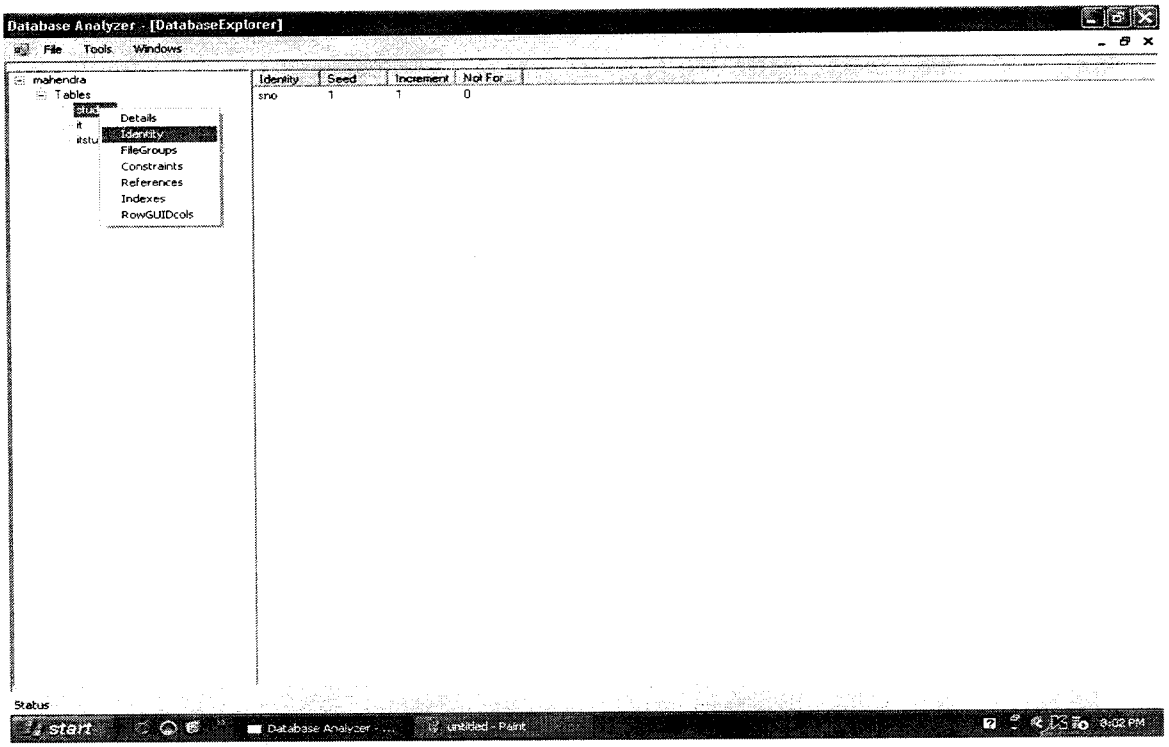


Figure A 2 .11 Tables Proprietors in Database Explorer

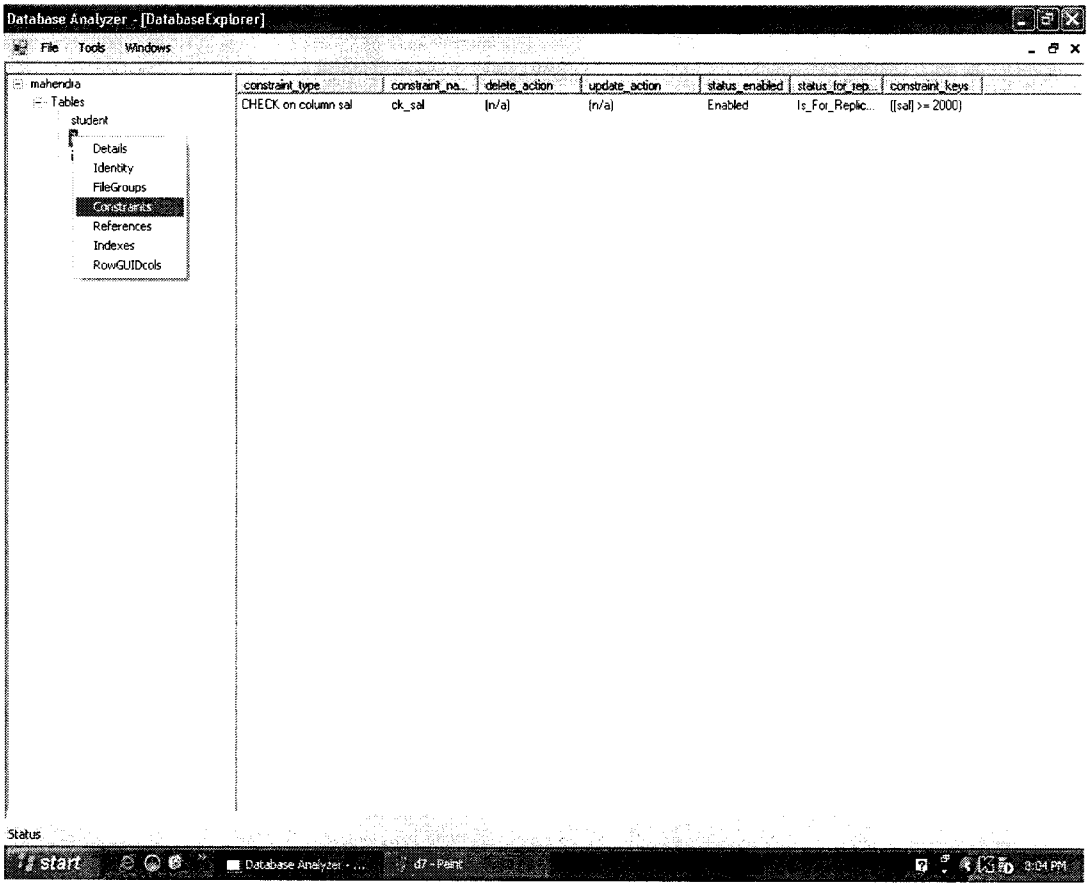


Figure A 2 .12 Tables Constrains

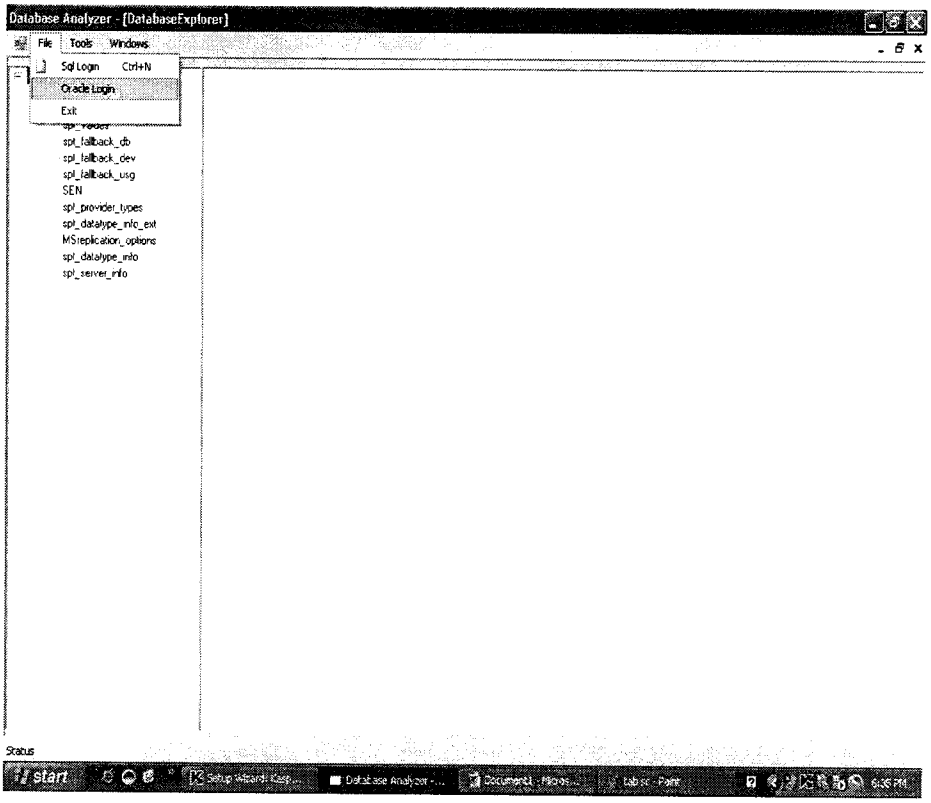


Figure A 2 .13 Login From Menu

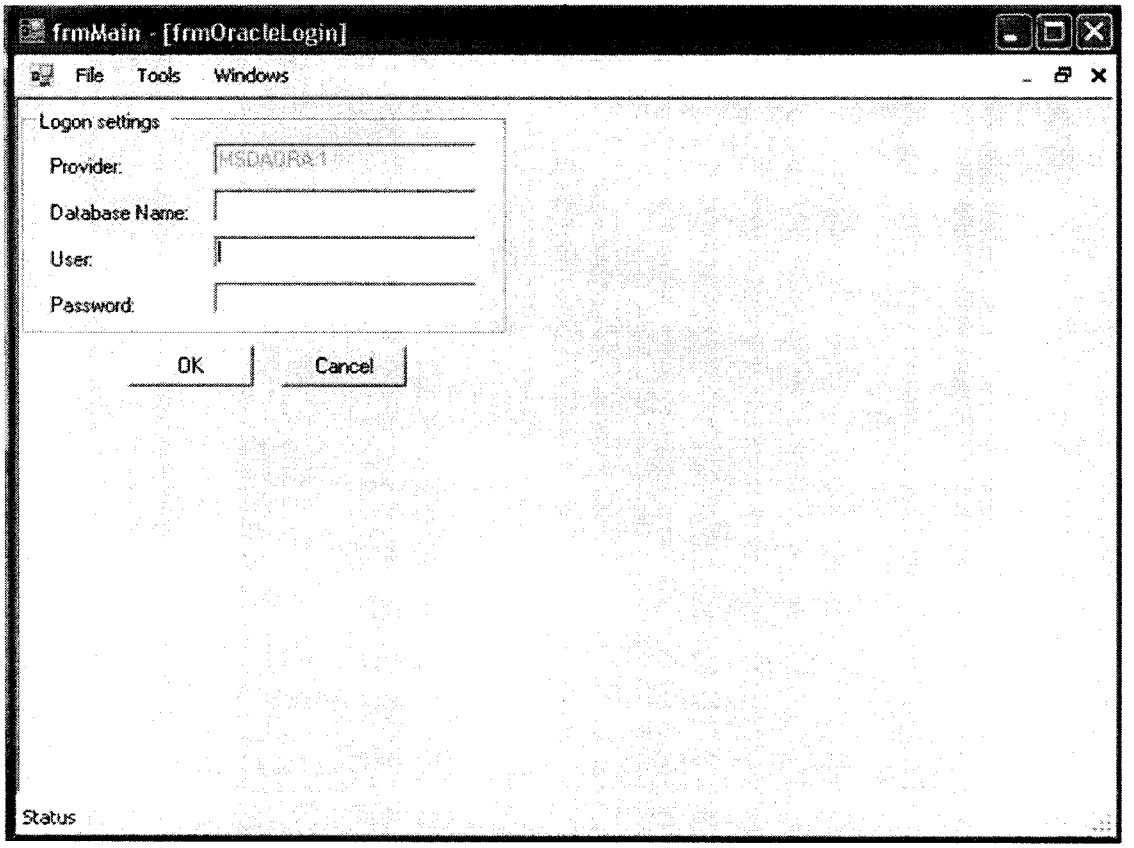
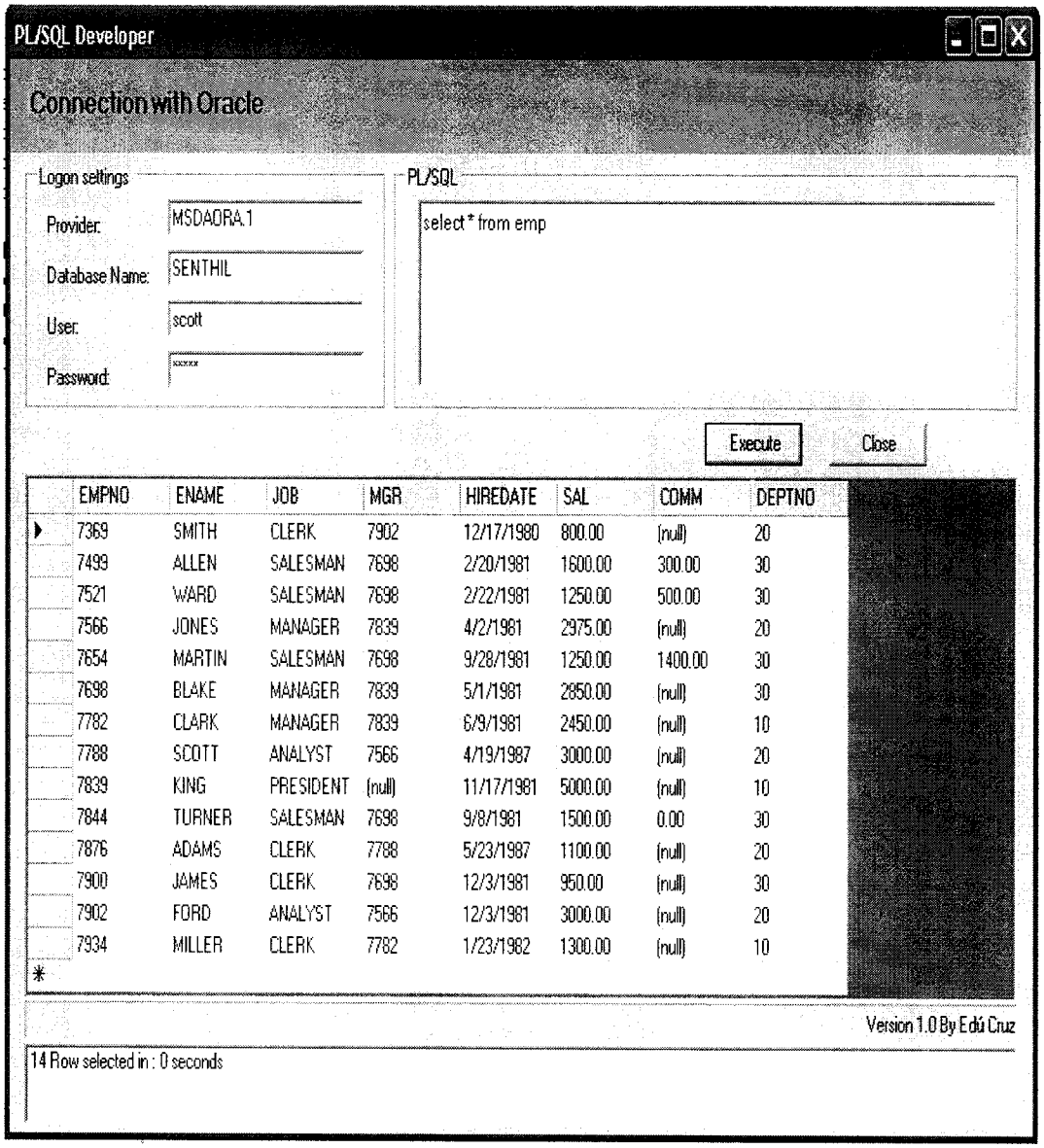


Figure A 2 .14 Oracle Login



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
▶	7369	SMITH	CLERK	7902	12/17/1980	800.00	(null)	20
	7499	ALLEN	SALESMAN	7698	2/20/1981	1600.00	300.00	30
	7521	WARD	SALESMAN	7698	2/22/1981	1250.00	500.00	30
	7566	JONES	MANAGER	7839	4/2/1981	2975.00	(null)	20
	7654	MARTIN	SALESMAN	7698	9/28/1981	1250.00	1400.00	30
	7698	BLAKE	MANAGER	7839	5/1/1981	2850.00	(null)	30
	7782	CLARK	MANAGER	7839	6/9/1981	2450.00	(null)	10
	7788	SCOTT	ANALYST	7566	4/19/1987	3000.00	(null)	20
	7839	KING	PRESIDENT	(null)	11/17/1981	5000.00	(null)	10
	7844	TURNER	SALESMAN	7698	9/8/1981	1500.00	0.00	30
	7876	ADAMS	CLERK	7788	5/23/1987	1100.00	(null)	20
	7900	JAMES	CLERK	7698	12/3/1981	950.00	(null)	30
	7902	FORD	ANALYST	7566	12/3/1981	3000.00	(null)	20
	7934	MILLER	CLERK	7782	1/23/1982	1300.00	(null)	10
*								

Figure A 2 .14 Show Tables

REFERENCES

BOOKS

- Harold Davis,' Visual vb.net programming', Tata McGraw-Hill Publications, Second Edition, 2002.

- Balaguruswamy,'Introduction to .net', Tata McGraw-Hill Publications, Third Edition, 2005.

WEBSITES

- www.garykessler.net/library/steganography.html
- http://www.frontlinedefenders.org/manual/en/esecman/chapter2_8.html
- http://www.computerworld.com/securitytopics/security/story/0,10801,71726,00.html
- www.aes.org/events/113/papers/1.cfm
- en.wikipedia.org/wiki/WAV
- www.borg.com/~jglatt/tech/wave.htm