

MSDE ENTERPRISE MANAGER

PROJECT WORK DONE AT
LOGIC SOFTWARE SOLUTIONS PVT.LTD.COCHIN

PROJECT REPORT

P-797

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE

SUBMITTED BY

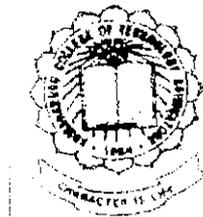
MARY MURPHY PHILIP

Reg. No. 9938M0619

GUIDED BY

EXTERNAL GUIDE
Mr.JOGAN ANTONY,
Project Leader,
Logic Software Solutions,
Cochi

INTERNAL GUIDE
Mr.S.GANESH BABU MCA.,
Lecture,
Department of Computer Science,
Kumaraguru College of technology,
Coimbatore.



Department of Computer Science & Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore-641 006
May 2002



SOFTWARE SOLUTIONS PVT. LTD.

Logic Towers, St. Albert's College Lane, Banerji Road, Cochin - 682 018, Kerala, India.

Tel: 91-484-393141, 393148, 398448, Fax: 91 - 484 - 396201

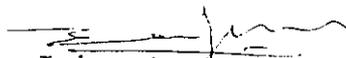
E-Mail: lss@eth.net, www.lssindia.com



CERTIFICATE

This is to certify that the project titled "**Enterprise Manager**" is a bonafide record of the project work done by **Ms. Mary Murphy Philip, MCA** student of Kumaraguru College Of Technology (affiliated to Bharathiar University) , Coimbatore during the period **December 2001 to April 2002**.

The project was successfully completed and tested for its functionality at **LOGIC SOFTWARE SOLUTIONS PVT. LTD.**


Jojan Antony
Project Manager




Biju Joseph Jacob
Director

Department of Computer Science & Engineering
Kumaraguru College Of Technology
(Affiliated to the Bharathiar University)
Coimbatore-641 006

CERTIFICATE

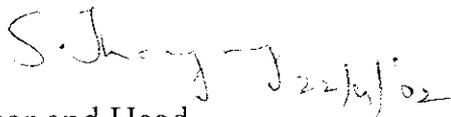
This is to certify that the project work entitled

MSDE ENTERPRISE MANAGER

Done by

MARY MURPHY PHILIP
Reg.No: 9938M0619

Submitted in partial fulfillment of the requirements for the award of the degree of
Master of Computer Application of Bharathiar University.

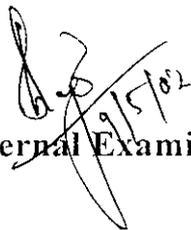
 22/6/02

Professor and Head
Department of Computer Science & Engineering



Internal Guide

Submitted to University Examination held on ..09..08..2002.

 9/5/02

Internal Examiner



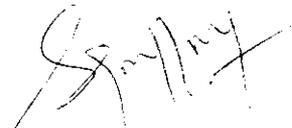
External Examiner

DECLARATION

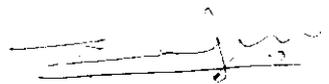
I hereby declare that the project entitled **MSDE ENTERPRISE MANAGER** submitted to **Bharathiar University** as the project work of **Master of Computer Application Degree**, is a record of Original work done by me under the supervision and guidance of **Mr.Jogan Antony MCA,MCP,Project Leader, Logic Software Solutions Pvt, St.Albert's Lane , Cochin** and **Mr.S.Ganesh Babu MCA, Kumaraguru College Of Technology, Coimbatore-641 006** and **this project work has not found the basis for the award of any Degree /Diploma /Associateship /Fellowship or similar title to any candidate of any university.**

Place: *Cochin*

Date: *16-04-2002*


(Internal Guide)


MARY MURPHY PHILIP


(External Guide)

ACKNOWLEDGMENT

*I am incredibly fortunate to have so many people helping me on this project. First I wish to thank **Mr. Biju Joseph jacob, Director, Logic Software Solutions Pvt Ltd, Cochin** for entrusting this project to me and providing all facilities that made the experience a pleasant one. In particular my deepest thanks goes to **Mr. Jogan Antony, Project manager**, for recognizing and approving this project and for his immense support, timely suggestions, and technical guidance throughout the project work.*

*I am extremely thankful to **Dr.S.Thangasamy, Professor and Head of the Department** for his kind suggestion he has given in every step throughout our studies and in this project work. I am grateful to my internal guide **Mr. S Ganesh Babu MCA, Lecturer, Kumaraguru College of Technology** who offered his guidance and always supported me with keen interest and constant encouragement suggestions in every step through out the project.*

I wish to thank my parents for all their encouragement support and love. Without their encouragement I would never have gotten where I am.

Last but not the least I would like to thank my Classmates and friends for their love and moral support. Finally thanking You God for your immense support and love.

SYNOPSIS

MSDE, Build desktop and shared solutions that are fully compatible with Microsoft SQL Server 7.0 and can migrate to SQL Server. Making use of Microsoft Data engine, a user interfaces like the Enterprise Manager of SQL Server, has to be developed. This product is to be a Microsoft management console SnapIn (MMC SnapIn) component. Without installing MS SQL Server Client in a machine of an Intranet we will be able to manage the SQL Server through this component in a simple, reliable and efficient manner.

This component is an interface provided to MSDE (Microsoft database engine) provided by the Microsoft. The MSDE is same as that of ODBC, ADO, etc which is used to find the server and provide connection to the client with the server. For this purpose we need an interface to this MSDE. There is no other built in interface to this Data Engine.

This component is 100 percent compatible with SQL Server 7.0 and could be used in desktop machines.

TABLE OF CONTENTS

	PAGE NO.
1. Introduction.	
1.1 Project Overview.	1
1.2 Organization Profile.	2
2. System Study & Analysis.	
2.1 Existing System-Limitations.	3
2.2 Proposed System-Advantages.	4
2.3 Requirements on new system.	7
3. Programming Environment.	
3.1 Hardware Configuration.	8
3.2 Software Configuration.	9
3.3 Microsoft Visual C++.	10
3.4 Microsoft Data Engine.	20
4. System Design & Development.	
4.1 Design Methodology	23
4.2 Code Design	23
4.3 I/P Design.	23
4.4 O/P Design	29
4.5 Process Design.	33
5. System Implementation & Testing.	
5.1 System Implementation.	35
5.2 System Testing.	36
5.3 System Security	38
6. Conclusion.	39
7. Scope for further development.	40
8. Bibliography	41
9. Appendices	

1.Introduction

1.1 Project Overview

AIM

Without installing MS SQL Server in the client machine of an Intranet we will be able to manage the SQL Server through this component in a simple, reliable and efficient manner.

DESCRIPTION

The MSDE Enterprise Console is used manage the SQL Server in an Intranet. It is a graphical user interface (GUI) tool for managing the SQL Server. Because without using queries we can perform any operations like creating databases, tables, stored procedure, views etc.

MSDE Enterprise Console is an interface provided to MSDE (Microsoft Data Engine), which is used to find the server. This Enterprise Console can be connected to more than one SQL Server. It is used to import data from one database to another and export vice versa.

This component is 100 percent compatible with SQL Server 7.0 and could be used in desktop machines.

1.2 Organization Profile

LOGIC is Cochin based Information Technology consulting firm specializing in the design, development and integration of custom-built business systems and solutions using advanced internet, electronic commerce and client server technology.

Since 1996, it has built a solid reputation as an enterprise dedicated to offering a complete and sound approach to systems development in the emerging paradigm of the networked world.

The clients are industry leaders who pursue innovative ways of exploiting information technology to create, manage and distribute information to support their business strategies. They seek a new approach to the delivery of custom solutions, backed by a team of dynamic and competent professionals. They want a partner with the capacity to effectively develop new systems to realize their corporate objectives whether they be developing stronger customer relations, leveraging human capital, or improving business efficiency.

The management team is structured to fully support and guide the clients through their information system initiatives. With their solid experience in the field of systems consulting, development and integration, the managers understand the human and business aspects of information technology and systems development as well as the technical issues. This assures the clients of a greater chance of achieving the desired results.

Address

Logic Towers, St.Alberts College Lane,
Banerji Road Cochin-18

Phone: 0484-393141, 393148.

Email: lss@vsnl.com

Website: lssindia.com

2. System Study & Analysis

2.1 Existing System

At present in intranet we need MS SQL Server in client side machines to control the server side MS SQL database. It is very difficult to get the license for each package because it is very costly. Moreover in most of the organization there is only one SQL Server. This machine stores all data required by the programs. Execution of queries take place in this machine and only the results are passed to the client machine. Therefore time sharing system is being used. This leads to the delay of the programs and hence they have to wait in a queue till their chance comes.

Limitations of Existing System

- It should take high cost to get the license for each MS SQL (ie approximately 45-50 thousand for each package).
- Excessive usage of memory space
- It is more complex.

2.2 Proposed System

MSDE Enterprise Console is used to manage the SQL Server without installing the MS SQL in client machines. This component has the same properties of the Enterprise Manager of the MS SQL.

This component has the major functions like :-

1. Using this component we can control MSDE.
2. Moreover, for manage logins and user permission of Data Engine.
3. Manage backup & Restore of database.
4. MSDE Enterprise console involves creating and maintaining a number of interrelated components like

➤ *Database*

A database consists of a collection of tables with data, and other objects, such as views, indexes, stored procedures, and triggers, that are defined to support the activities performed with the data. Before objects within the database can be created, it is necessary to create the database and change the settings and configure it accordingly. This includes tasks such as expanding or shrinking the database, or specifying the files used to create the database.

➤ *Tables*

Tables are database objects that contain all the data in a database. A table definition is a collection of columns in the same way a database is a collection of tables. Before data can be stored in a database, user must understand how to create, modify, and maintain the tables within your database. This includes tasks such as defining keys and adding or deleting columns from a table.

➤ *Views*

By creating, modifying, and maintaining views, it is possible to customize each user's perception of the database. A view is a virtual table whose contents are defined by a query. Like a real table, a view consists of a set of named columns and rows of data. However, a view does not exist as a stored set of data values in a database. The rows and columns of data come from tables referenced in the query defining the view and are produced dynamically when the view is referenced.

➤ *Stored Procedures*

By creating, modifying, and using stored procedures, it is possible to simplify your business applications and improve application and database performance. Accept input parameters and return multiple values in the form of output parameters to the calling procedure or batch. Contain programming statements that perform operations in the database, including calling other procedures. Return a status values to a calling procedure or batch to indicate success or failure (and the reason for failure).

➤ *Connect/Disconnect*

There is a two menu item like connect and disconnect which is used to connect the databases and disconnect from it.

➤ *Backup and Restore of database*

We can create backup for files, for the security reason and we can restore the file if we want.

❖ Advantages of Proposed System

- Without installing MS SQL Server Client in any machine of an Intranet we will be able to manage the SQL Server.
- Functionalities of the Enterprise Manager can be performed on the client machine by installing MSDE.
- MSDE is free licensed software, so the cost of installing the clients is made null.
- Consumption of space is reduced to megabytes.
- This system is simple, reliable and efficient.

2.3 Requirements on new system

The Enterprise Manager has certain requirements:

- First of all the client should be connected to a Intranet.
- MSDE, Microsoft free database engine should be installed in each and every system in the Intranet.
- At least one system in the network should have SQL Server installed in it.
- For installing our developed application including MSDE the system should have 30GB free space in it.

3. Programming Environment

3.1 Hardware Configuration

The hardware configuration for MSDE Enterprise Console to perform smooth running is as

- *Server*

Processor	: Intel Pentium III 500 MHz
Memory	: 128 MB RAM
Hard Disk	: 1.2 GB or above
Monitor	: 14" SVGA Monochrome or Above

- *Client*

Processor	: Intel Pentium III 500 MHz
Memory	: 64 MB RAM
Hard Disk	: 1.2 GB or above
Monitor	: 14" SVGA Monochrome or Above

3.2 Software Configuration.

The software specification for MSDE –Enterprise manager to perform smooth running is as

Server Module –

Windows NT 4.0/2000
MS SQL Server 7.0 or Greater

Client Module –

Windows NT 4.0/2000/98
MSDE

❖ *Programming Language Used:*

Microsoft Visual C++ 6.0

3.3 Microsoft Visual C++

Visual C++ is an extremely powerful tool for Windows programming - in fact, many programmers consider it the most powerful Windows Programming Tool today. Visual C++ is not actually one tool, it is a collection of tools, wrapped together into one dynamic, ready to use package.

Visual C++ enjoyed a reputation for creating small and efficient programs. A program, which is written using this language, can operate at nearly the same speed as one written in assembler, without any problems. The C++ is a great language for writing low-level code like operating system, device drivers, and DLLs. These three kinds of code still represent the major places where Visual C++ is used. The small fast code that Visual C++ produces is very much appreciated in the time-critical environment of an operating system.

One of the latest features that Microsoft has added is better prototyping capabilities through the use of enhanced wizards. Now this feature went place Visual C++ on par with the products like VB6, but it reduce the time required to get an application started.

Visual C++ is also an ideal programming environment for writing ActiveX controls, along with IIS-specific code like ISAPI (Internet Sever Application Programming Interface) extensions and ISAPI filters.

In addition Microsoft introduced the Microsoft Foundation Class library, which is an extra ordinary package of prewritten, ready to use code. Introducing MFC was just as important as introducing VC++ to windows. Visual C++ not only makes the use of MFC, but also makes Windows programming far easier by introducing many programming tools, such as menu editor for designing menus, and the dialog editor for designing the dialog boxes. Visual C++ provides us with one integrated design environment in which we can write our programs and run them. In addition Visual C++ organizes the many files a Windows program needs into projects. Visual C++ wizards will even write a good part of our programs for us.

Microsoft Visual C++ version 6.0 is available in three editions: *Standard*, *Professional*, and *Enterprise*.

➤ ***The Standard Edition:***

Learn the C++ language while using the professional Visual C++ tool set. The Standard Edition contains all of the features of the Professional Edition except code optimizations, the Profiler, and static linking to the MFC Library.

➤ ***The Professional Edition:***

Develop applications, services, and controls for Win32 platforms, including Windows 95 and Windows NT. You can target the operating system's graphical user interface or console APIs.

➤ ***The Enterprise Edition :***

Develop and debug client-server applications for Internet or intranet use. The Enterprise Edition includes tools for working with SQL databases and debugging SQL stored procedures in addition to all the features of the Professional Edition. The Visual SourceSafe source-code control system simplifies developing in a team environment.

❖ **Visual C++ Workspaces And Projects**

Visual C++ organizes programming task into the projects, and usually each separate gets into its own project. A project is a collection of files that are all used to create one working, runnable program. In addition, projects themselves are placed in workspaces, and a workspace can have several projects in it.

❖ **The Parts Of A Visual C++ Program.**

There are four major parts of a Visual C++ AppWizard program: *the application object, the main window object, the document object, and the view object.*

➤ *The Application object:*

The application object supports the definitions of constants and the declarations of variables and methods, when this object is started; it places the main window in the screen

➤ *The Main Window object:*

The main window object displays the program itself, and this is where we find in the menu bar, the title bar, and tool bar. The main window object is responsible for everything that surrounds the area where the action-the drawing, text, and other graphics in our window.

➤ *The Document object:*

In the document object, we store the data of our program. Visual C++ makes it easier to store the all data in the document object, then handle the display of the data that will fit into the client area in the view object.

➤ *The View object:*

The view object handles the client area where we will format and display the data in our program, such as the text we are editing if we are creating a word processing program. The view object is really a window itself that appears on the top of the client area. The data we display in the view object is stored in the document object.

❖ The Application Types In Visual C++

Visual C++ is capable of creating any application that we can imagine. However, there are five application types that can be used with Visual C++.

➤ *Console Application:*

It represents those situations where we really need to maintain some type of compatibility with legacy systems or don't need a full-fledged interface for the user to work with.

➤ *Dialog-based Application:*

These applications are normally reserved for utilities or an application that is too small to require a complete menuing system.

➤ *Single-document Application:*

Single-document Applications are representative of simple applications that work with their own data like note takers or small database front ends. These applications also require a menuing system of some type.

➤ *Multiple-document Application:*

Multiple-document Applications include full-fledged applications like word processors and spreadsheets. When we think about it, they represent that fringe area of C++ programming where we need to weigh the flexibility of C++ against the development speed offered by RAD programming environments like Visual Basic.

➤ **HTML-based Application:**

HTML-based Application are new to Visual C++ 6.0. They are applications that work with data of some type (like single-document applications and multiple-document applications) but with an internal twist. Instead of standard editor, the user can see what amounts to Web browser front end.

❖ **Smart Pointer**

If we use Visual C++ 5.0 or later, we can access a very cool feature: smart pointers for COM objects that make using them very easy. These smart pointers are created when we use the new #import directive to "import" a type library. When we use #import, the Visual C++ compiler reads the type library from the file we specify. This file can be any file that contains a type library that can be read with the LoadTypeLib API. Usually the file will be a .TLB, .ODL, .EXE, .DLL, .RLL or .OCX, but it could be any file type that LoadTypeLib can read.

Here we use sqlldmo.rll file for import type library.

```
_SQLServerPtr sqlptr("SQLDMO.SQLServer");
```

Using this statement we can point to database engine.

❖ **Methods Used**

➤ **ExecuteWithResults Method**

The ExecuteWithResults method executes a Transact-SQL command batch returning batch result sets in a QueryResults object.

Syntax

object.ExecuteWithResults(Command , [Length]) as QueryResults

Parts

Command

String that specifies a Transact-SQL or provider-specific command batch.

Length

Optional. Long integer that indicates the statement batch length.

Returns

A QueryResults object.

➤ **GetUserName Method**

The GetUserName method returns the database user used by the referenced login, when a connection using that login accesses the specified database.

Syntax

object.GetUserName(Database)

➤ **SQLRestore Method**

The SQLRestore method performs the database restore operation specified by the properties of the Restore object used.

Syntax

object.SQLRestore(SQLServer)

Parts

SQLServer

SQLServer object connected to an instance of MSDE that is the target of the restore operation.

➤ **SQLBackup Method**

The SQLBackup method performs the database backup operation specified by the properties of the Backup object used.

Syntax

```
object.SQLBackup( SQLServer )
```

➤ **KillDatabase Method**

The KillDatabase method drops a database from the dataengine regardless of the status or availability of the database.

Syntax

```
object.KillDatabase( Database )
```

Database

String specifying an existing database by name

❖ **SQLServer Object (SQL-DMO)**

The SQLServer object contains the objects and collections that implement SQL Server administrative tasks for SQL-DMO. The object allows SQL-DMO applications to connect to a server running SQL Server by name, establishing the context for administrative tasks.

With the SQLServer object, we can:

- Connect to an existing server running SQL Server.
- Query a server running SQL Server to determine its installed configuration and run-time parameters.

- Add and remove SQL Server objects, such as backup devices, databases, and logins.
- Execute Transact-SQL or operating system commands on the server.
- Disable processes on a server running SQL Server.
- Trap SQL Server events and SQLServer object events, providing status information to SQL-DMO application users or debugging information to SQL-DMO application developers.

➤ **LoginSecure Property (SQL-DMO)**

The LoginSecure property directs authentication mode use when the application attempts to use the Connect method of a SQLServer object to connect to a server.

Syntax

object.LoginSecure [= value]

Object Expression that evaluates to an object in the Applies to list

Value True or false

➤ **LoginTimeout Property (SQL-DMO)**

The LoginTimeout property specifies the number of seconds to wait for a connection attempt to succeed.

Syntax

object.LoginTimeout [= value]

➤ **Connect Method (SQL-DMO)**

The Connect method attempts to establish a connection with the Microsoft SQL Server installation named.

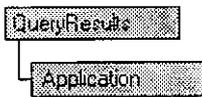
Syntax

object.Connect([**ServerName**], [**Login**], [**Password**])

Parts

<i>Object</i>	Expression that evaluates to an object in the Applies To list.
<i>ServerName</i>	Optional. A string specifying a SQL Server installation by name
<i>Login</i>	Optional. A string specifying a SQL Server login by name.
<i>Password</i>	Optional. A string specifying a password authenticating the SQL Server login.

❖ **QueryResults Object (SQL-DMO)**



The QueryResults object presents tabular data to the SQL-DMO application. SQL-DMO enumeration methods, such as the EnumLocks method of the Database object, return a QueryResults object to report their data. SQL-DMO statement execution methods, such as the ExecuteWithResults method of Database and SQLServer objects, also return a QueryResults object.

The QueryResults object is a reporting tool. All properties of the QueryResults object are read-only. With the QueryResults object, you can:

- Navigate data returned from a server running Microsoft® SQL Server™ as the result of statement execution.
- Retrieve specific data values in a data type usable by your application.
- Get the data result of a statement execution as a delimited string of values

➤ **CurrentResultSet Property (SQL-DMO)**

The CurrentResultSet property controls access to the result sets of a QueryResults object.

Syntax

object.CurrentResultSet [= value]

➤ **GetColumnString Method (SQL-DMO)**

The GetColumnString method returns a QueryResults object result set member converted to a String value.

Syntax

object.GetColumnString(Row , Column) as String.

To get the user defined tables :-

Query =SELECT name,xtype,create from database name.dbo.sysobjects where type="u";

To get the stored procedure :-

Query =SELECT name,xtype,create from database name.dbo.sysobjects where type="p";

To get the views :-

Query =SELECT name,xtype,create from database name.dbo.sysobjects where type="v";

3.4 Microsoft Data Engine

With the release of the Microsoft Data Engine (MSDE), a SQL Server-compatible database engine, there is an exciting new option for creating desktop and shared database solutions. MSDE enables Visual Studio developers to build freely distributable desktop and shared database solutions that easily migrate to SQL Server 7.0 when the solution must scale.

A "database engine" is the core process that a database management system, such as Access or SQL Server, uses to store and maintain data. A database engine is often just a database management system without its graphical management tools. Engines typically serve two main purposes. The first is to store information, and the second is to process requests for stored information. Requests to retrieve or alter stored data are often called database queries.

The most common use for a database engine is in conjunction with standalone custom data engine that can be used for desktop and shared database solutions built with either Visual Studio 6.0 or Access 2000. In addition, solutions built with MSDE for Visual Studio 6.0 or Office 2000 Developer can be distributed royalty-free.

MSDE is based on SQL Server technology and is fully compatible with SQL Server 7.0. When used in Access 2000, MSDE is an attractive alternative to Jet—the default Access 2000 database engine. In addition, MSDE is available to Visual Studio 6.0 developers who wish to build custom applications with embedded databases that are SQL-compliant. SQL compliance enables desktop and shared database solutions using MSDE to be scaled up to SQL Server without code modifications.

In summary, the MSDE Relational database engine exhibits the following characteristics and features:

- Is fully compatible with Microsoft SQL Server 7.0 and SQL Server technologies.

- Is freely distributable for solutions built with MSDE for Visual Studio 6.0 or Microsoft Office 2000 Developer.
- Runs on Windows® 95, Windows 98, Windows NT, and Windows 2000 (both Intel and Alpha) platforms.
- Has client-server architecture (database operations occur on the database server, not the client machine).
- Runs as a service.
- Does not include a user interface for database development. Instead, other applications such as Access 2000 or custom Visual Basic® applications provide the user interface.

SQL Server is a client-server database management system. Unlike file-based databases, client-server database engines manage read-write operations to the database. Because of this, client-server databases such as SQL Server can handle many more concurrent users and vastly greater amounts of data. SQL Server is designed to meet the most demanding database application requirements for the enterprise, including operational and decision support systems implemented today and in the future. SQL Server is offered in different editions based upon data, user, and hardware requirements.

➤ **Cost**

MSDE is available to any licensed customer of Office 2000 Professional, 2000 Premium edition, or any Visual Studio 6.0 Enterprise or Professional edition tool, including:

- Visual Studio 6.0, Professional and Enterprise editions.
- Visual Basic 6.0, Professional and Enterprise editions.
- Visual C++® 6.0, Professional and Enterprise editions.
- Visual InterDev™ 6.0 Professional edition.

- Visual J++® 6.0, Professional edition.
- Visual FoxPro® 6.0 Professional edition.

Developers that build desktop and shared database solutions using MSDE may distribute their applications royalty-free, provided that the developer of an MSDE solution is licensed for the following products:

- Microsoft Office 2000 Developer.
- Any Microsoft Visual Studio 6.0 Professional or Enterprise tool (as listed above).

➤ *Size*

MSDE has a data limit of 2 GB. Jet also has a data limit of 2 GB in each of its data files. MSDE is tuned for best optimum performance at five concurrent users or less. Microsoft performance levels available strongly suggests that the number of concurrent users be five .

MSDE runs as a service on the machine where it is installed. It only needs to be installed once as the MSDE service can host multiple databases from different applications. Typical MSDE installation requires 55 megabytes (MB) of local hard drive space, and 64 MB of system memory is recommended.

MSDE is for use with Windows 98, Windows 95, Windows NT 4.0 with Service Pack 4 or later installed, and Windows 2000.

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 Design Methodology.

System design is vital for efficient database management. In the present software, conceptualization of the system involved, designing the different tables, form layouts for inputs and reports for the output. The design is solution, the translation of requirements into ways of meeting them. The design will determine the success of the system. Based on the proposed system objectives, the major modules are identified and the operations to be carried out or determined. In the design phase of this project, inputs, outputs and screens are designed.

Are input screens in the system are user friendly and are designed in such a way that even a layman can operate. The size of all the screens is standardized.

The important of software design can be stated with a single word quality. Design is a place where quality is fostered in software development. Design is the only way where requirements are actually translated in to a finished software product or system.

4.2 Code Design.

In the design phase code design has an important role . Code are group of characters to identify a data item. Each item must have unique identification and must be related to the facilities of a mnemonic code.

4.3 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the

computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

In input design the following things are considered.

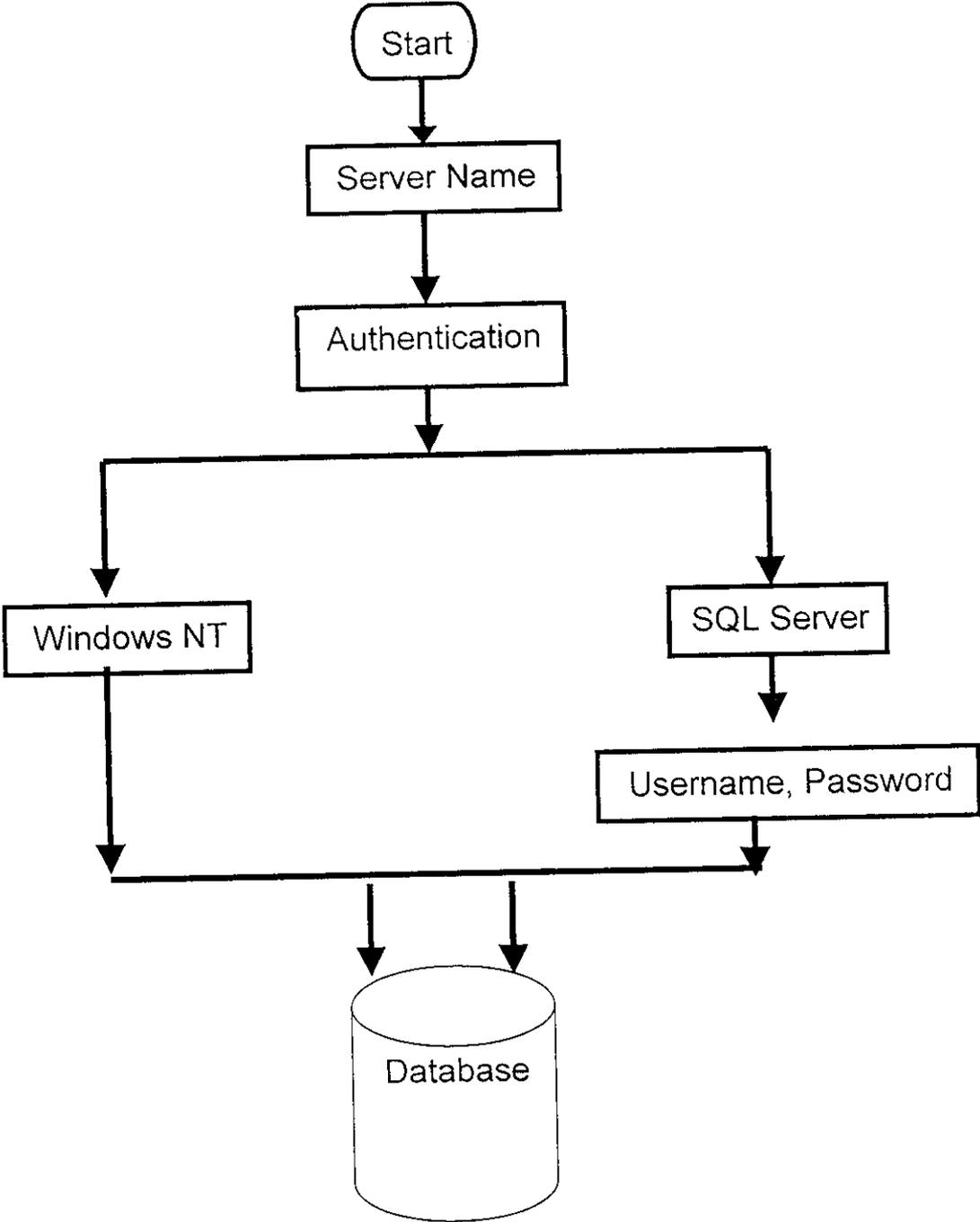
- What data should be given as input?
- How the data should be arranged or coded?
- The dialogue to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

The main inputs acquired from the user are as follows.

1. SQL Server name
Name of the SQL Server
2. User name and password, Information required to logon to the server

Using the connect dialog box of the input design we can give the server name first. Then there is two authentication is available. First one is Windows NT authentication. Here there is no need for username and password .The hardware will directly take then connection to the database. But in the case of SQL server authentication we have to enter the username and password, then only it will connect to the database. The System connection flow chart is

SYSTEM FLOW CHART



The main input designs are given below:

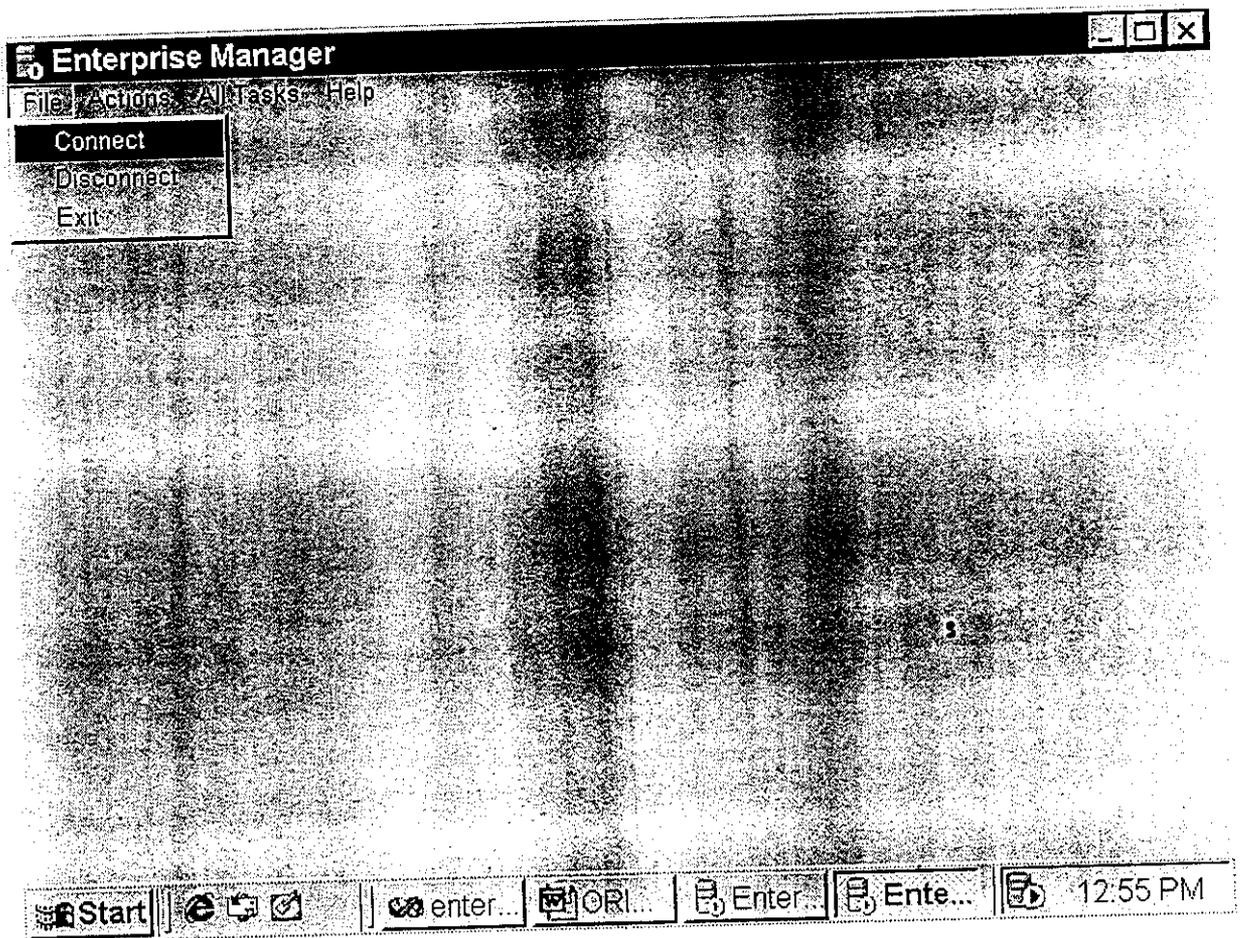
➤ **Connect Menu**

This is used for the database-connection. This is the first input view for the enterprise manager. Here we can see four menus like File, Action, All Tasks, & Help.

In the File -> menu we can see the Connect, Disconnect and Exit options. If we want to connect we can click the connect option.

The design is

CONNECT MENU

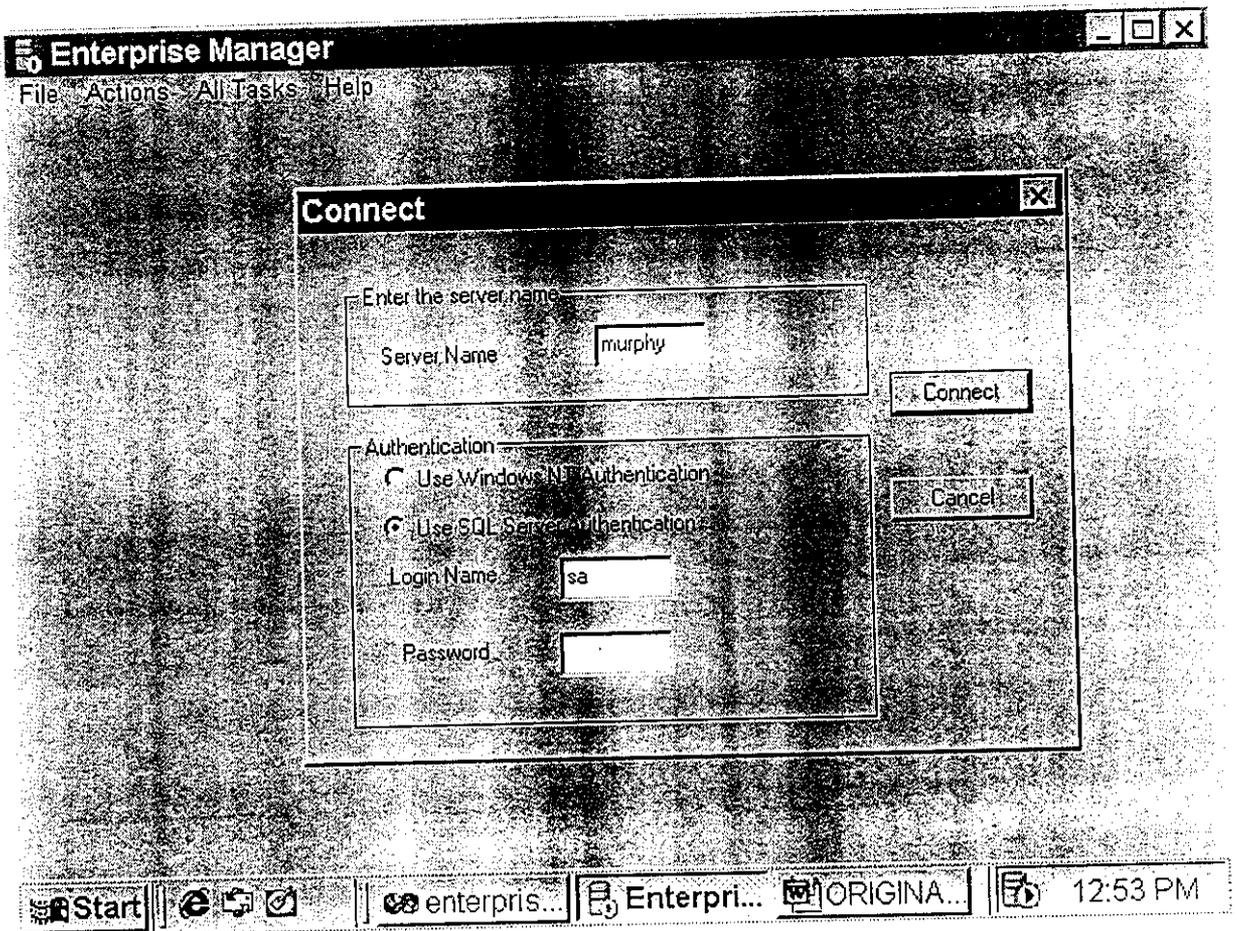


➤ **Connect Dialog.**

When we click in the connect option we will get a connect dialog which include Server name and Authentication. In the Authentication we have two option like use windows NT authentication and use SQL server Authentication. If we select SQL server authentication we have to input the login name and password. And also there is a connect and cancel buttons.

The design is

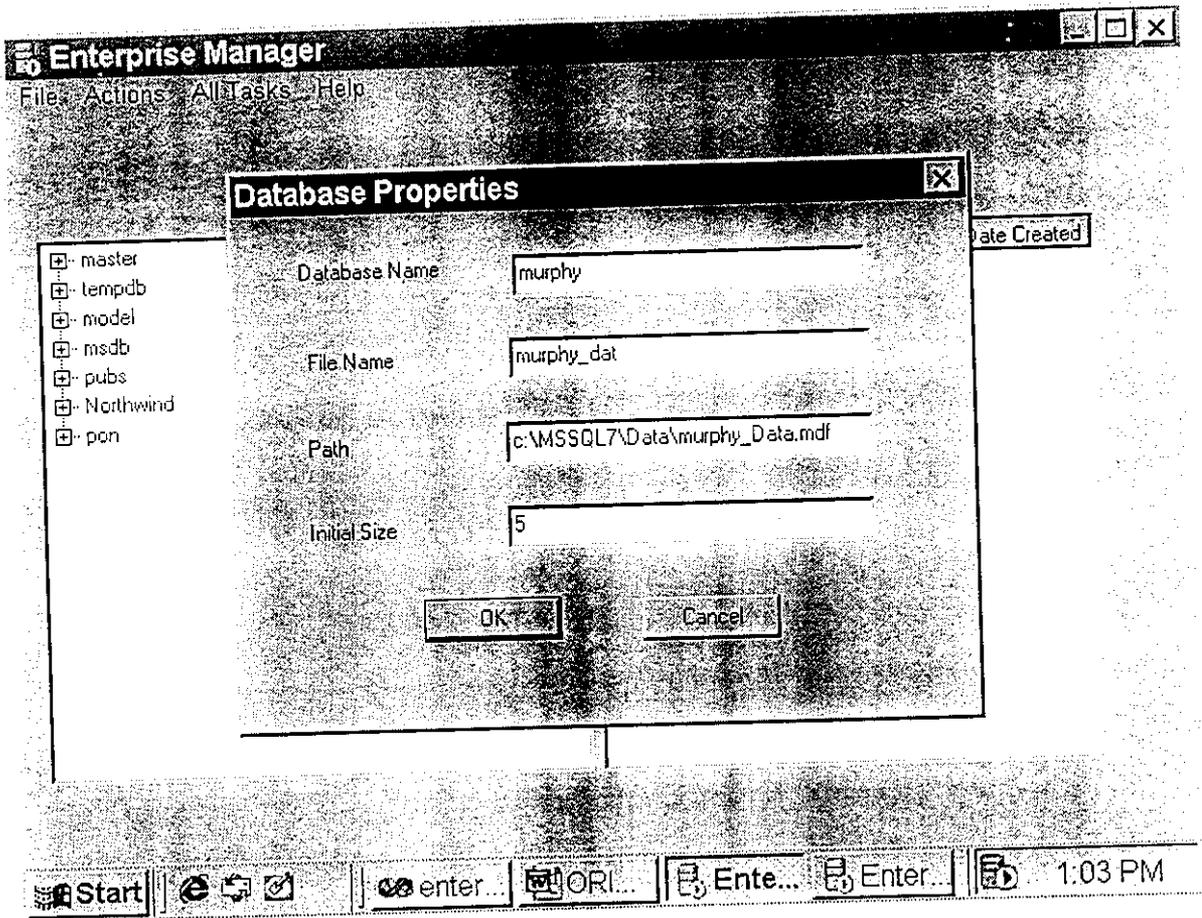
CONNECT DIALOG



➤ **Database Creation.**

If we want to create a new database we have to click on new database which is reside in Action menu. Then a dialog box will appear and we have to specify the Database name, Path and the Initial size of the new database and click on “ok” button

DATABASE CREATION



4.4 Output Design.

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directly referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So while designing output the following things are to be considered.

- Determine what information to present
- Arrange the presentation of information in an acceptable format
- Decide how to distribute the output to intended recipients

Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy. The main outputs, which are to be handled in this software, are as follows.

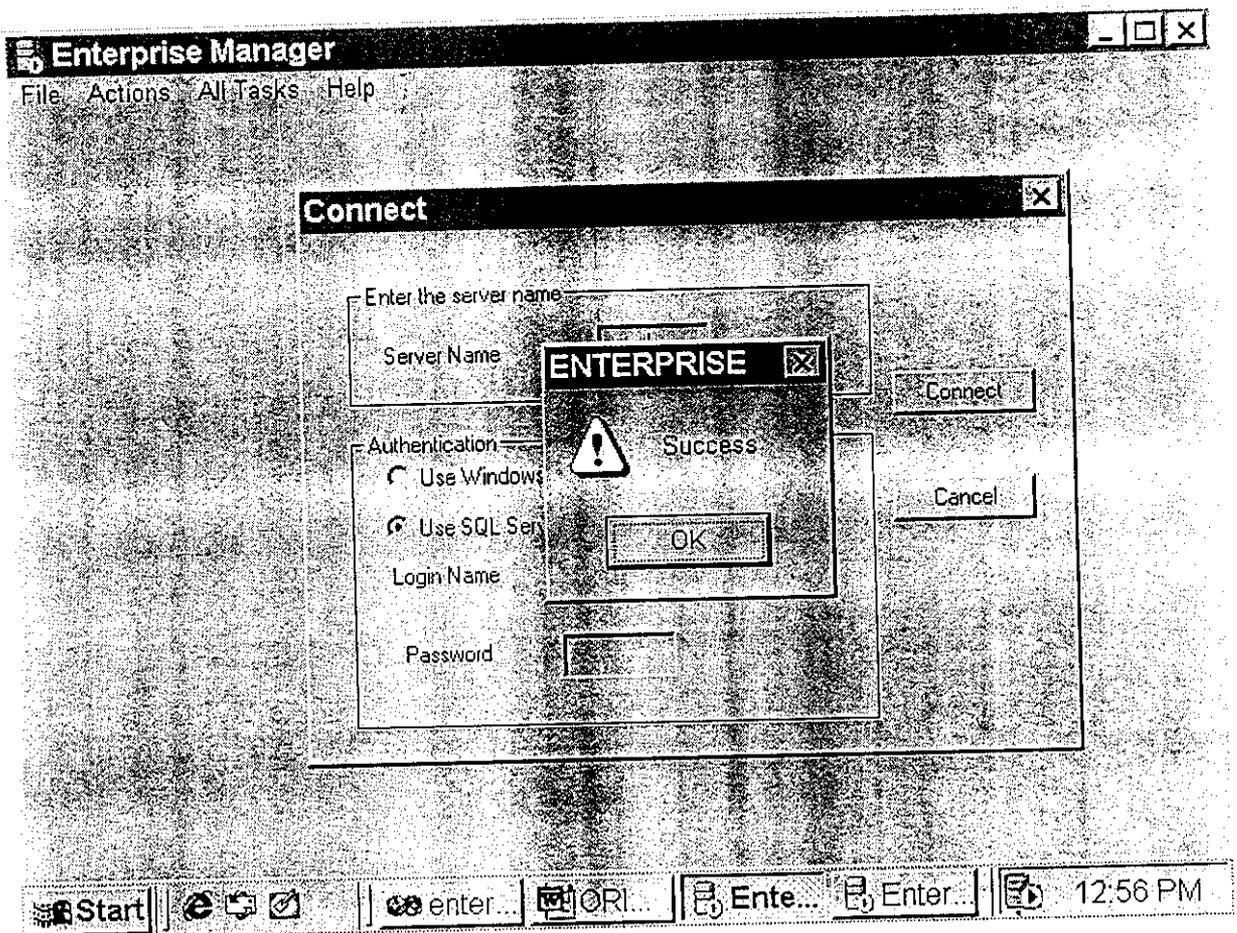
- Display the entire properties of the database (in a tree view).
- Display the name of the tables present in the database.
- Display the design view of the tables.
- Display the data stored in a table.
- Display the forms for creating the database, tables, stored procedures, views etc.
- Display the forms for connection and disconnection to and from the server.

The main output design of MSDE Enterprise Manager are:

➤ *Connection Success And Failed.*

Firstly we have to input the server name and password, if the connection success there will be a message as output “Success” otherwise message will be “Failed to Connect Server”.

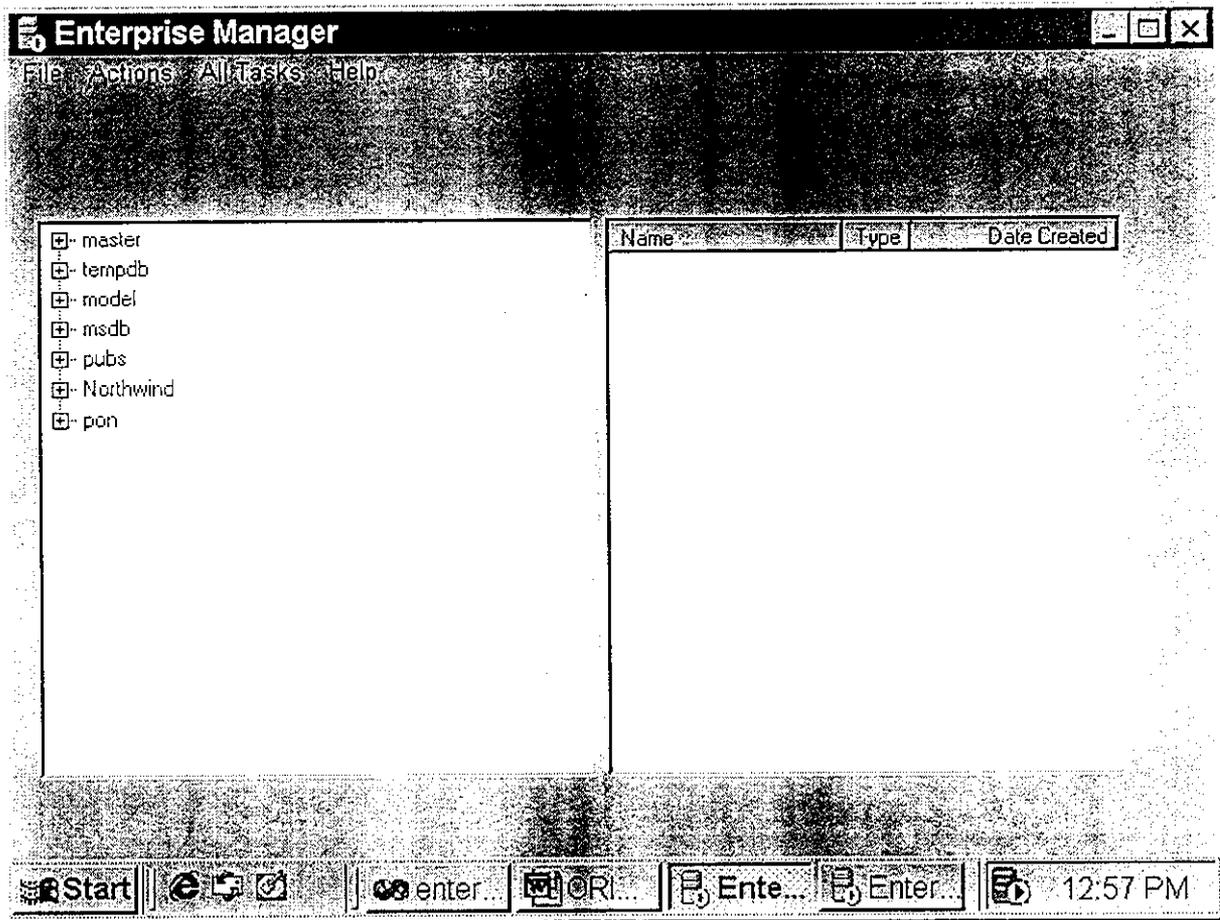
CONNECTION SUCCESS



➤ *Database View.*

If the connection is success the databases which is stored in the SQL server will appear in the tree view.

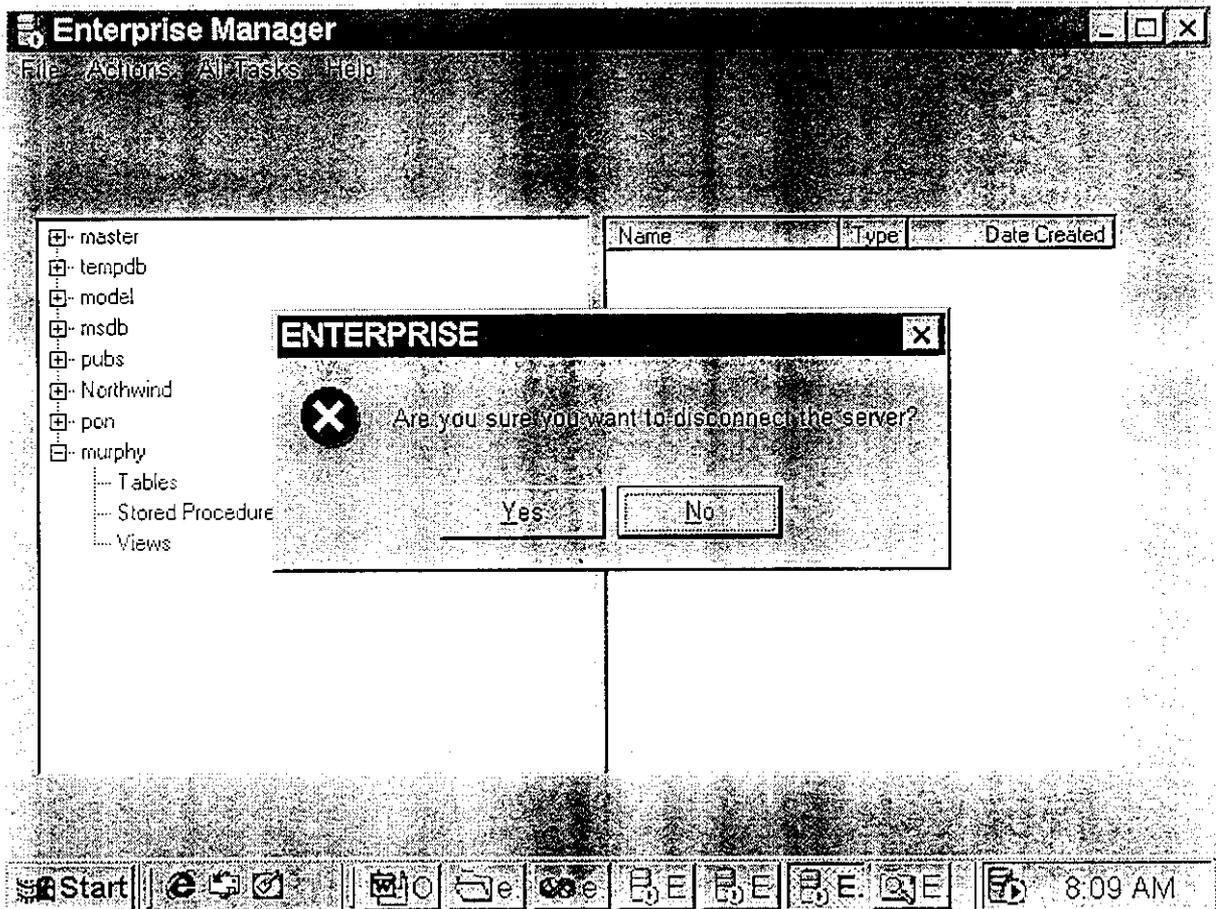
DATABASE VIEW



➤ *Disconnect.*

If we want to disconnect from the server we have to select the File-Disconnect Menu and click on Disconnect

DISCONNECT MENU.



4.6 Process Design.

How to retrieve the database is a main part of the software . For the retrieving . For the retrieving of database which is stored in the server. we have to give the command through the enterprise manager.

The command is

```
select name from master.dbo.sysdatabases
```

The command will go to the SQL server through the MSDE. The basic idea is, in SQL server if we create a new database, the content of the database will store to the sysdatabases which is reside in master database.

So we can retrieve all databases by accessing the master database.

For the accessing of the objects the same method is used. Objects means tables, stored procedure and views. The command is different for all these objects.

For the accessing of tables the command is

```
select name from databasename .sysobjects where xtype = u
```

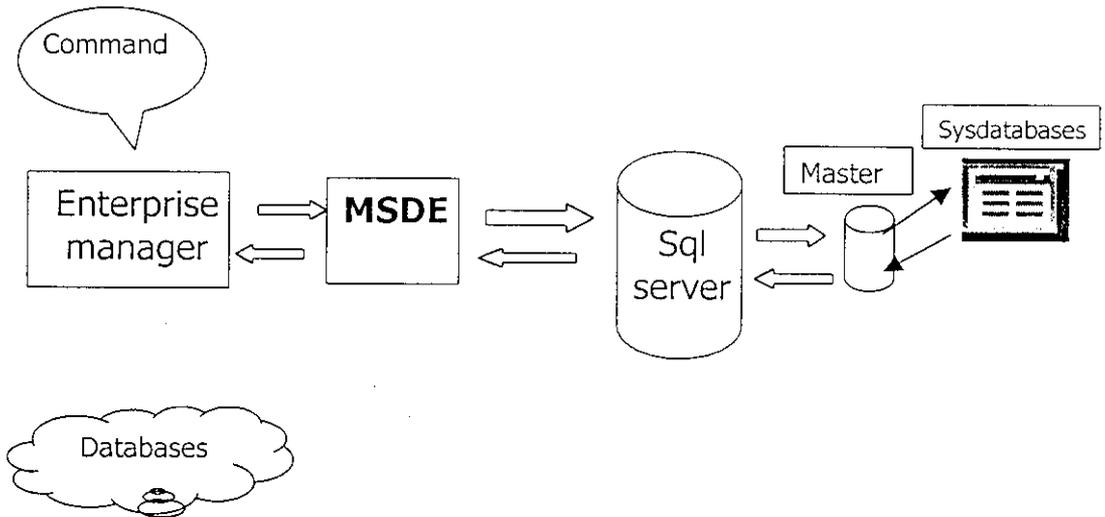
For stored procedures

```
select name from databasename .sysobjects where xtype = p
```

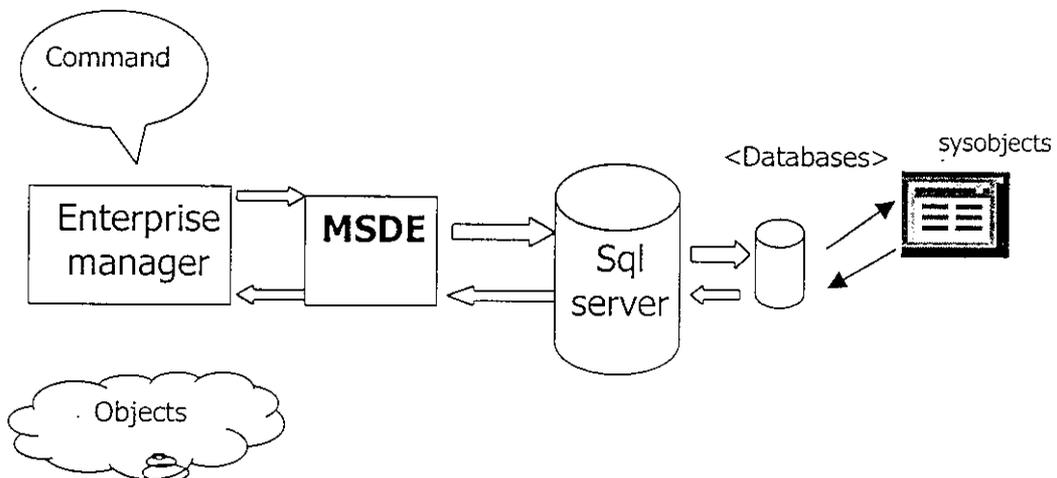
For views

```
select name from databasename .sysobjects where xtype = v
```

PROCESS FLOW - 1 (DATABASES)



PROCESS FLOW - 2 (OBJECTS)



5. System Implementation & Testing

5.1 System Implementation.

Implementation includes all those activities that take place to convert from the old system to the new. The old system consists of manual operations, which is operated in a very different manner from the proposed new system. A proper implementation is essential to provide a reliable system to meet the requirements of the organizations. An improper installation may affect the success of the computerized system.

In the existing system, it should take high cost to get the license for each MS SQL (ie approximately 45-50 thousand for each package). It will take excessive usage of memory space and it is also more complex.

We didn't need to install the SQL server in every machine when we use this software. We have to install SQL server in server and we can manage it from the client using the software. Functionalities of the Enterprise Manager can be performed on the client machine by installing MSDE. MSDE is free licensed software, so the cost of installing the clients is made null. Consumption of space is reduced to megabytes. This system is simple, reliable and efficient

5.2 System Testing.

System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding an as yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is the vital to the success of the system.

In the code testing the logic of the developed system is tested. For this every module of the program is executed to find an error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions.

❖ Unit Testing

Unit testing focuses first on the modules in the proposed system to locate errors. This enables to detect errors in the coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. In unit testing step each module has checked separately.

❖ Integration Testing.

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name.

Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software has executed several times in order to find out the errors in the different modules of the system.

❖ **Validation Testing.**

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment. By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted specifications starting the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

➤ ***Peak Load Tests***

This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.

➤ ***Storage Testing***

This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.

➤ ***Performance Time Testing***

This test determines the length of the time used by the system to process transaction data.

Here in the MSDE Enterprise Manager unit testing, Integration testing and Validation testing are performed in the testing phase.

5.3 Security of MSDE

MSDE databases running on Windows NT / Windows 2000 platforms are capable of utilizing windows 2000-integrated security to protect database objects. MSDE allows you to secure individual database objects such as tables or stored procedures. This enhances database security. Jet databases are not. Both MSDE and Jet allow you to secure individual database objects such as tables or stored procedures. If you want to leverage the security features of Windows 2000, MSDE is the right choice.

For the security of the software using the connect dialog box of the input design we can give the server name first. Then there is two authentications are available. First one is windows NT authentication. Here there is no need for username and password .The hardware will directly take then connection to the database. But in the case of SQL server authentication we have to enter the username and password, then only it will connect to the database.

6. CONCLUSION

MSDE is the data engine supplied by Microsoft. It is a mere database engine without any interfaces such as the query analyzer or enterprise manager, which SQL server possesses. A database engine is often just a database management system without its graphical management tools. Thus with the MSDE in hand, if we can develop an environment which fully exploits the services provided by it, we can transform this engine into a complete DBMS product. To do this we require interfaces like the enterprise manager of the SQL Server.

By the use of this Enterprise Console and another interface called Query Analyser combine to get the full properties of the MS SQL. Enterprise Manager has fewer options when compared to the Query Analyser. For eg. In Enterprise Manager we can create, modify and maintain a table, but in Query Analyser, by the use of Query we can get data in different formats as we needed. But in Enterprise Manager at a time we can see the complete properties of the database. It will show as a tree structure. This Enterprise Manager is more graphical than the query analyzer because in query analyzer, by the use of query we can perform any graphics. In Enterprise Manager there is no need of queries. By using mouse we can perform most of the operations.

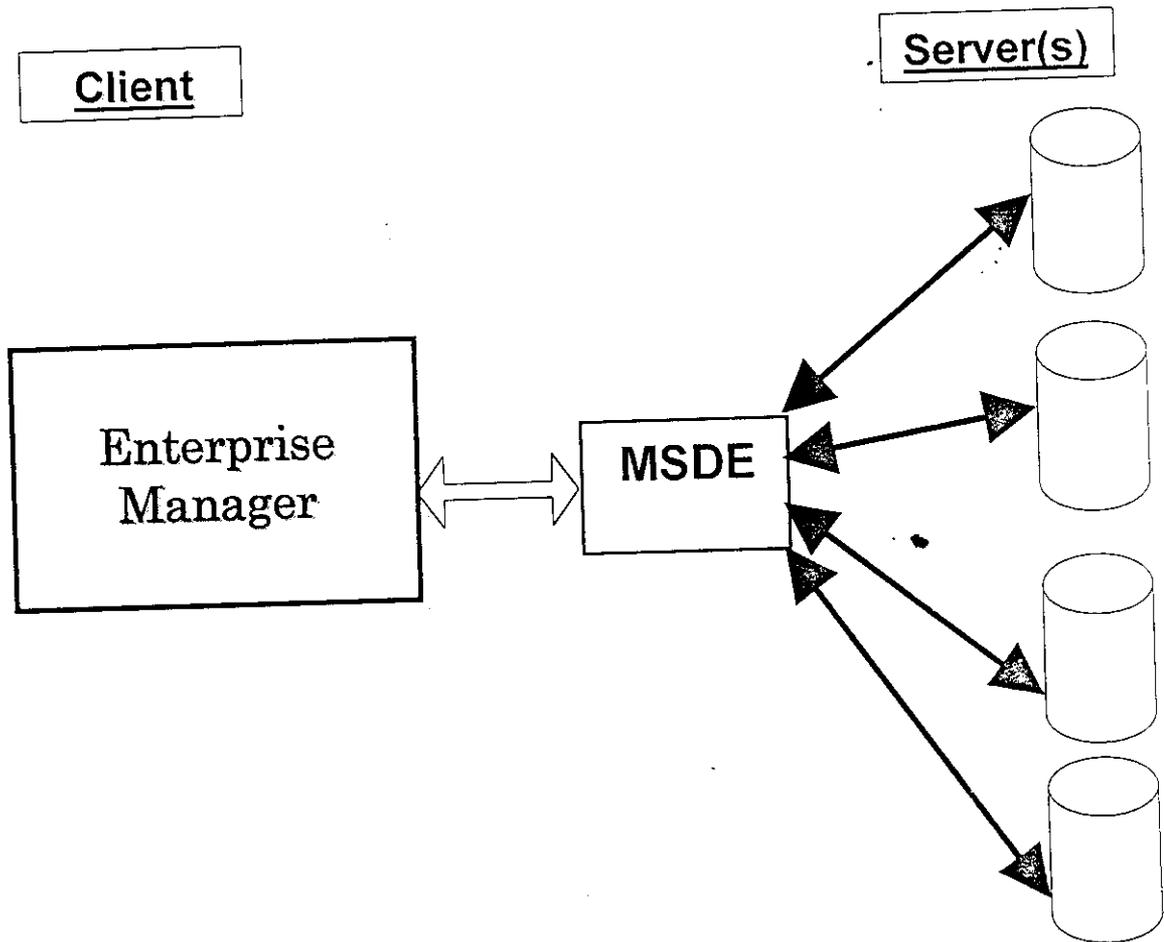
This package is developed for Logic Software Solutions Pvt Ltd, Cochin.

7.SCOPE FOR FURTHER DEVELOPMENT

- Service Manager can be included to control the SQL Server remotely.
- The font, size and color can be changed according to the users convenience.
- Printing functionality can be added.
- It can be added trigger functionalities

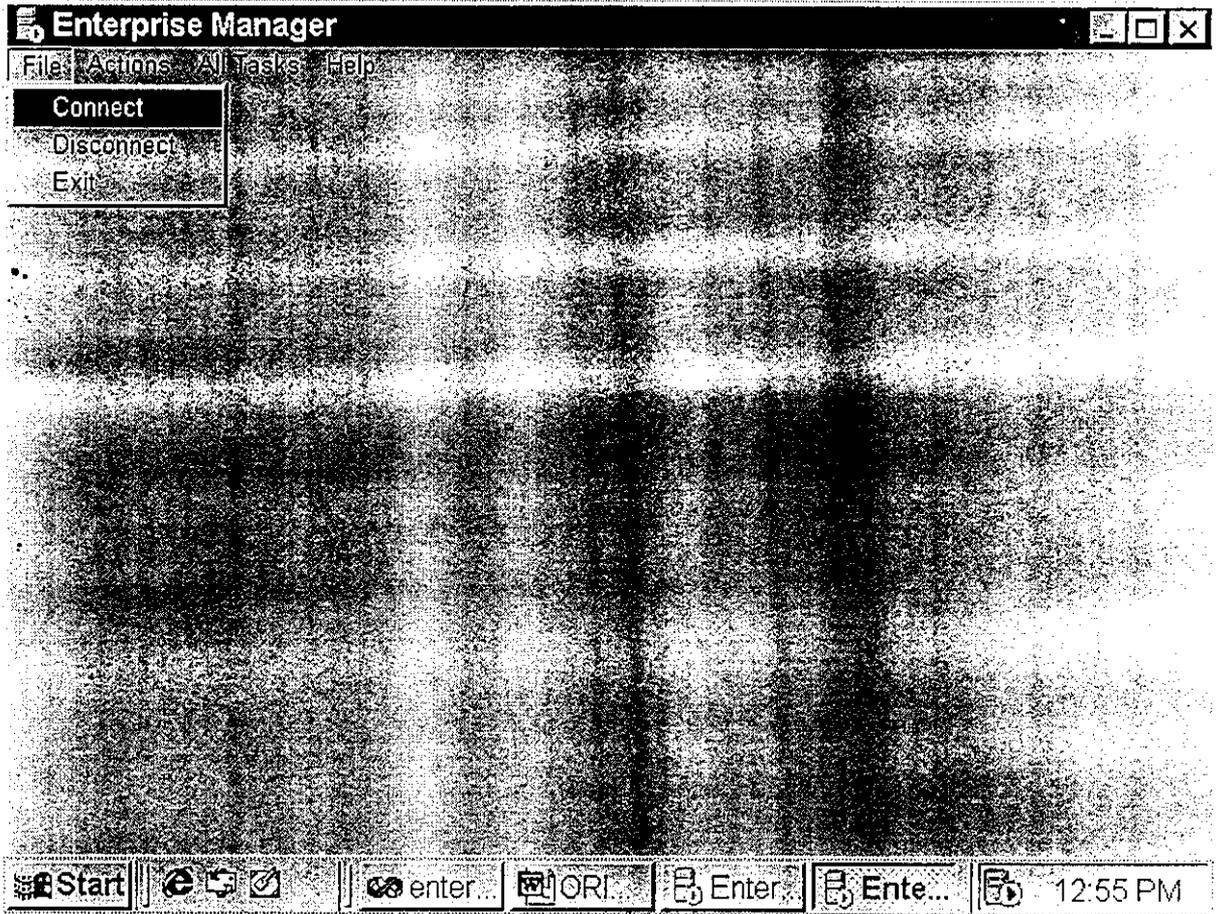
9. APPENDIX

SYSTEM ARCHITECTURE

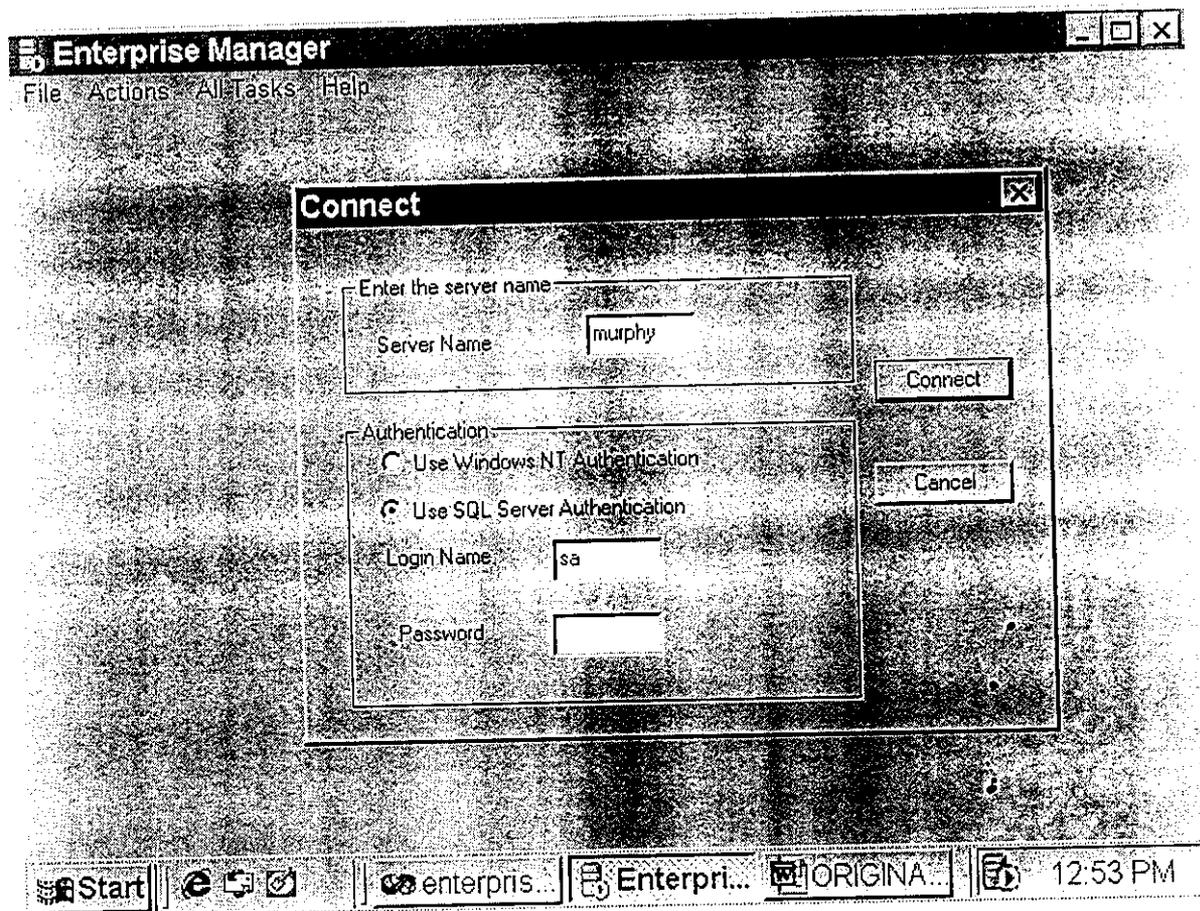


SCREEN VIEWS

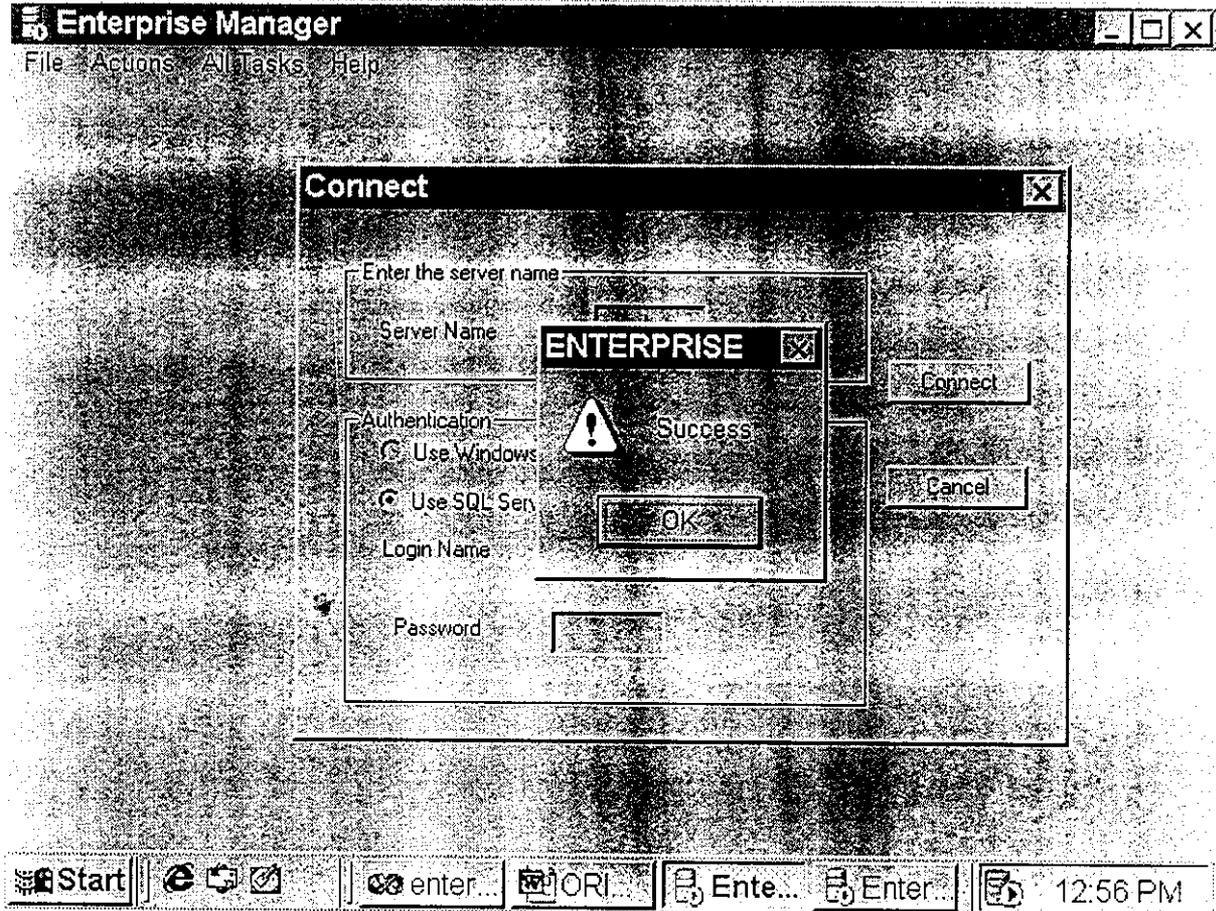
CONNECT MENU



CONNECT DIALOG



CONNECTION SUCCESS



DATABASE VIEW

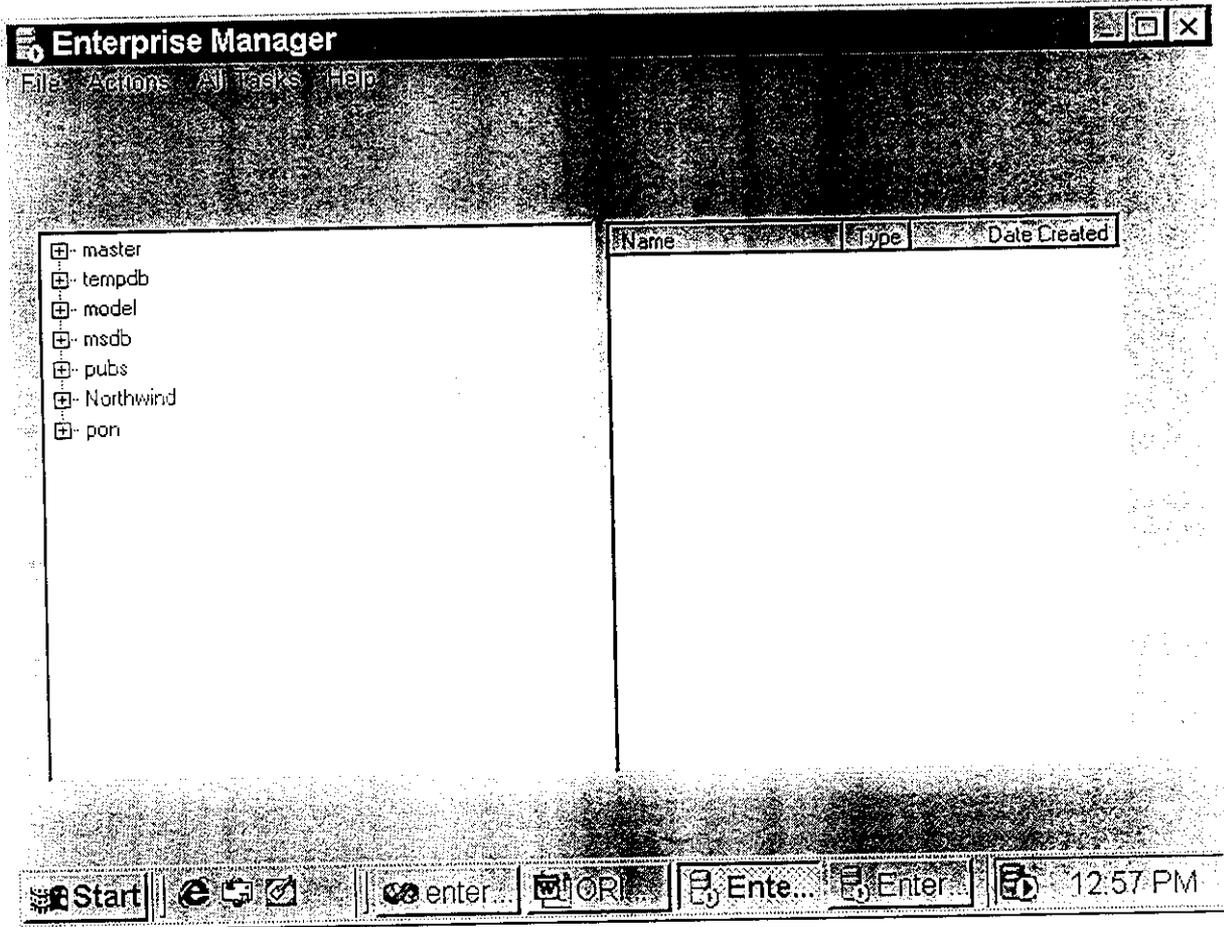


TABLE VIEW

The screenshot shows the Microsoft Enterprise Manager interface. The title bar reads "Enterprise Manager". The menu bar includes "File", "Actions", "All Tasks", and "Help". On the left, a tree view shows the "master" database expanded to show "Tables". The main pane displays a table with the following data:

Name	Type	Date Created
MSreplication_options	U	1998-11-13 03:14:...
spt_committab	U	1998-11-13 03:02:...
spt_monitor	U	1998-11-13 03:02:...
spt_values	U	1998-11-13 03:02:...
spt_fallback_db	U	1998-11-13 03:02:...
spt_fallback_dev	U	1998-11-13 03:02:...
spt_provider_types	U	1998-11-13 03:05:...
spt_fallback_usg	U	1998-11-13 03:02:...
spt_datatype_info_ext	U	1998-11-13 03:04:...
spt_datatype_info	U	1998-11-13 03:05:...
spt_server_info	U	1998-11-13 03:05:...

The taskbar at the bottom shows the Start button, several icons, and open windows for "enter...", "ORI...", "Ente...", and "Enter...". The system clock shows "12:58 PM".

STORED PROCEDURE VIEW

The screenshot displays the Microsoft Enterprise Manager interface. On the left, a tree view shows the database structure with 'Stored Procedures' selected under the 'master' database. The main pane shows a list of stored procedures with columns for Name, Type, and Date Created.

Name	Type	Date Created
sp_MSdropmergepub...	P	1998-11-13 03:19
sp_sqlexec	P	1998-11-13 03:03
sp_drop_agent_profile	P	1998-11-13 03:17
sp_changemergepulls...	P	1998-11-13 03:15
sp_MSscriptdb_worker	P	1998-11-13 03:13
sp_tables_ex	P	1998-11-13 03:05
sp_MScheckatpublisher	P	1998-11-13 03:19
sp_unbinddefault	P	1998-11-13 03:03
sp_help_agent_profile	P	1998-11-13 03:17
sp_MS_marksystemobj...	P	1998-11-13 03:00
sp_helpmergepullsubs...	P	1998-11-13 03:15
sp_MSdbuseraccess	P	1998-11-13 03:13
sp_columns_ex	P	1998-11-13 03:05
sp_addmergearticle	P	1998-11-13 03:19
sp_unbindrule	P	1998-11-13 03:03
sp_help_agent_default	P	1998-11-13 03:17
sp_MS_upd_sysobj_c...	P	1998-11-13 03:00
sp_addextendedproc	P	1998-11-13 03:02

The taskbar at the bottom shows the Start button, several icons, and open windows for 'enter...', 'OR...', 'Ente...', and 'Enter...'. The system clock indicates 12:58 PM.

VIEW TABLE

Enterprise Manager File Actions All Tasks Help

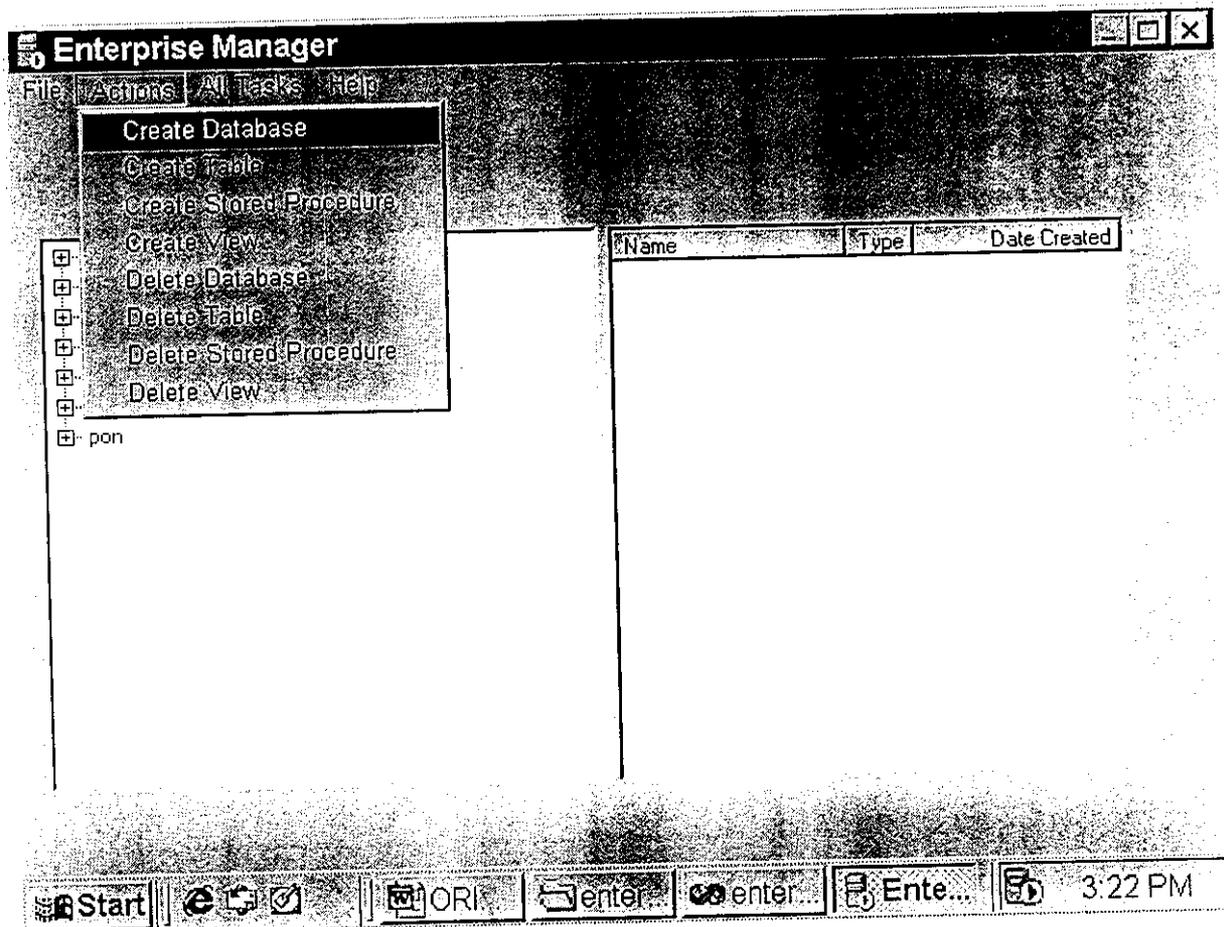
- [-] master
 - [-] Tables
 - [-] Stored Procedures
 - [-] Views**
- [+] tempdb
- [+] model
- [+] msdb
- [+] pubs
- [+] Northwind
- [+] pon

Name	Type	Date Created
syslogins	V	1998-11-13 03:00
sysremotelogins	V	1998-11-13 03:00
sysoledbusers	V	1998-11-13 03:00
syssegments	V	1998-11-13 03:00
sysconstraints	V	1998-11-13 03:00
sysalternates	V	1998-11-13 03:00
SCHEMATA	V	1998-11-13 03:06
TABLES	V	1998-11-13 03:06
TABLE_CONSTRAINTS	V	1998-11-13 03:06
TABLE_PRIVILEGES	V	1998-11-13 03:06
COLUMNS	V	1998-11-13 03:06
COLUMN_DOMAIN_USAGE	V	1998-11-13 03:06
COLUMN_PRIVILEGES	V	1998-11-13 03:06
DOMAINS	V	1998-11-13 03:06
DOMAIN_CONSTRAINTS	V	1998-11-13 03:06
KEY_COLUMN_USAGE	V	1998-11-13 03:07
REFERENTIAL_CONSTRAINTS	V	1998-11-13 03:07
CHECK_CONSTRAINTS	V	1998-11-13 03:07
CONSTRAINT_CATALOG	V	1998-11-13 03:07

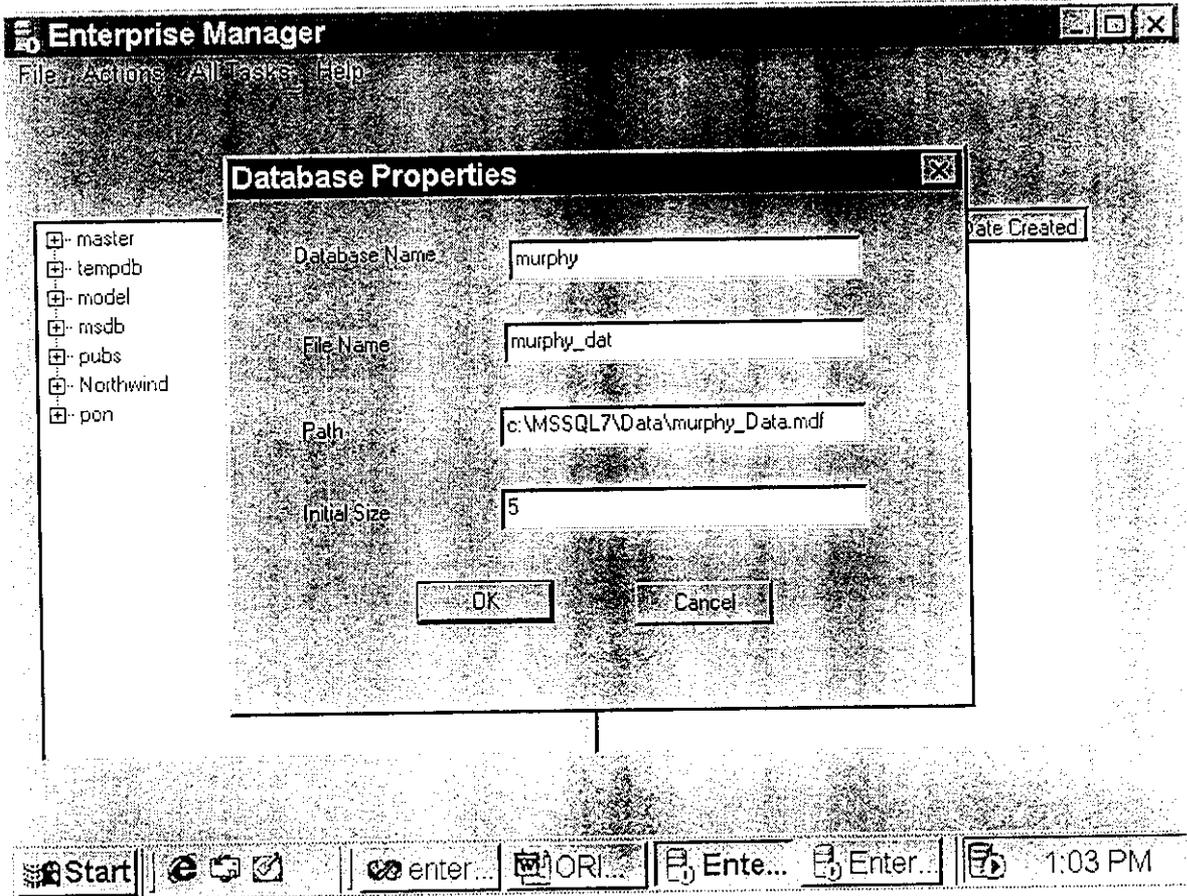
Start | enter... | ORI... | Ente... | Enter... | 12:59 PM

CREATE DATABASE

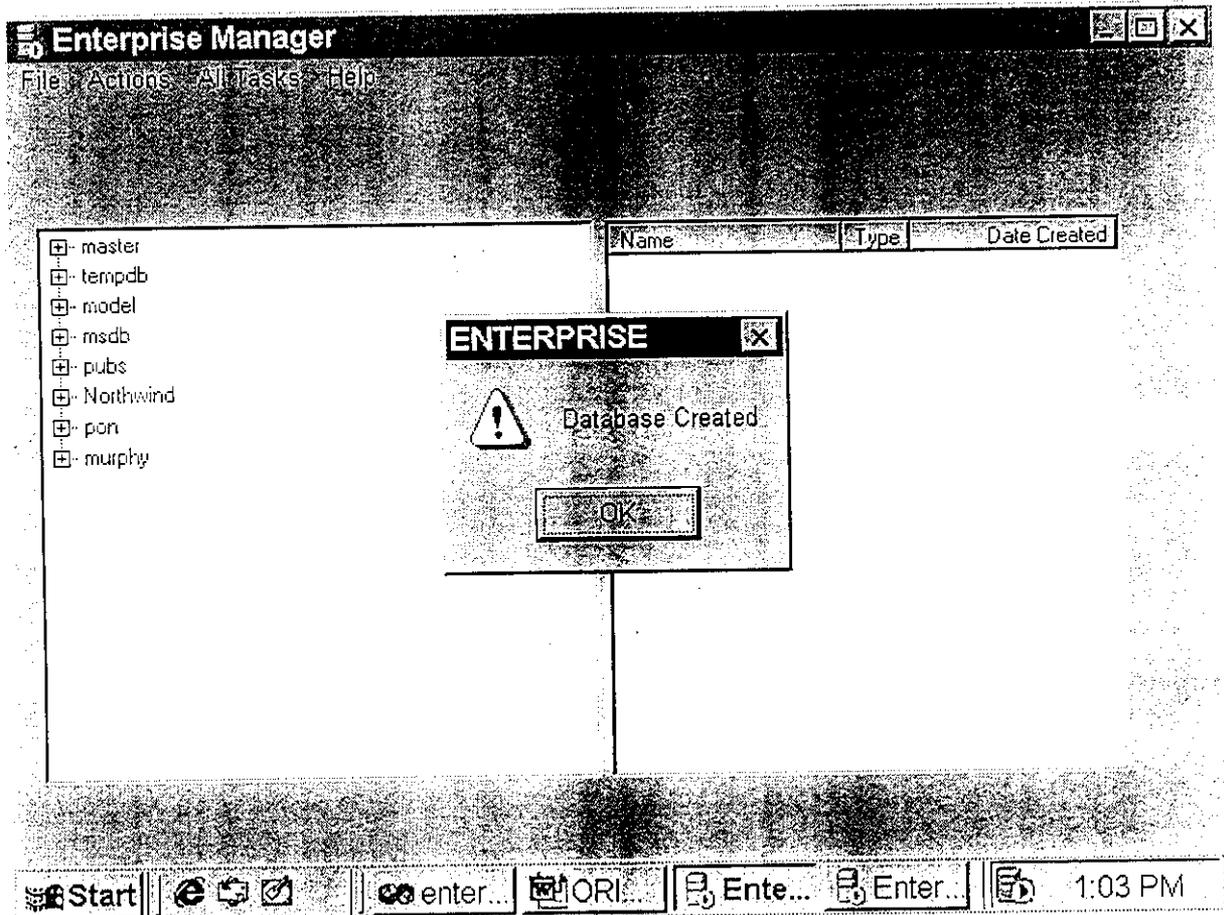
Step 1: Select *create database* from *Action* menu.



Step 2: Enter the database name.

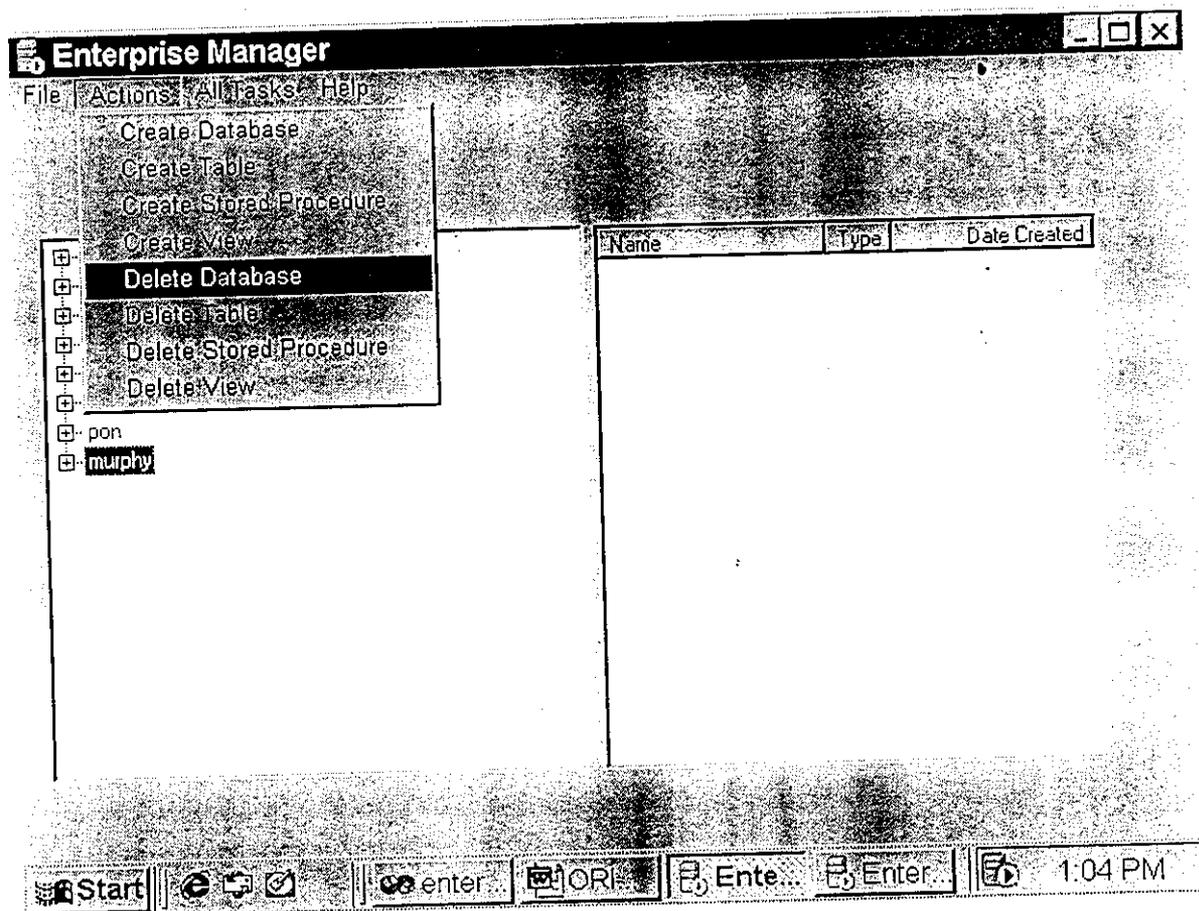


Step 3: Database Created.

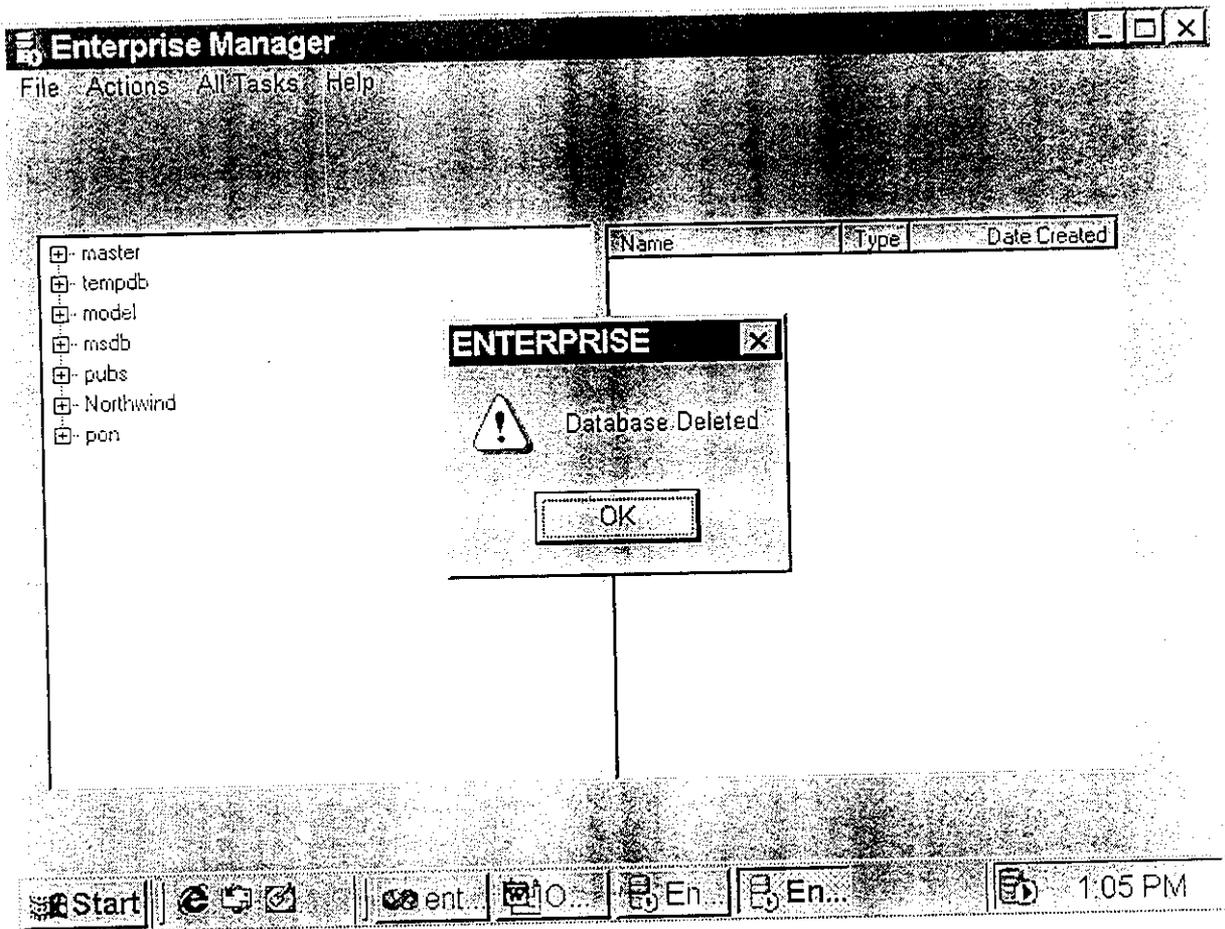


DELETE DATABASE.

Step 1: Select *delete database* from *Action* menu

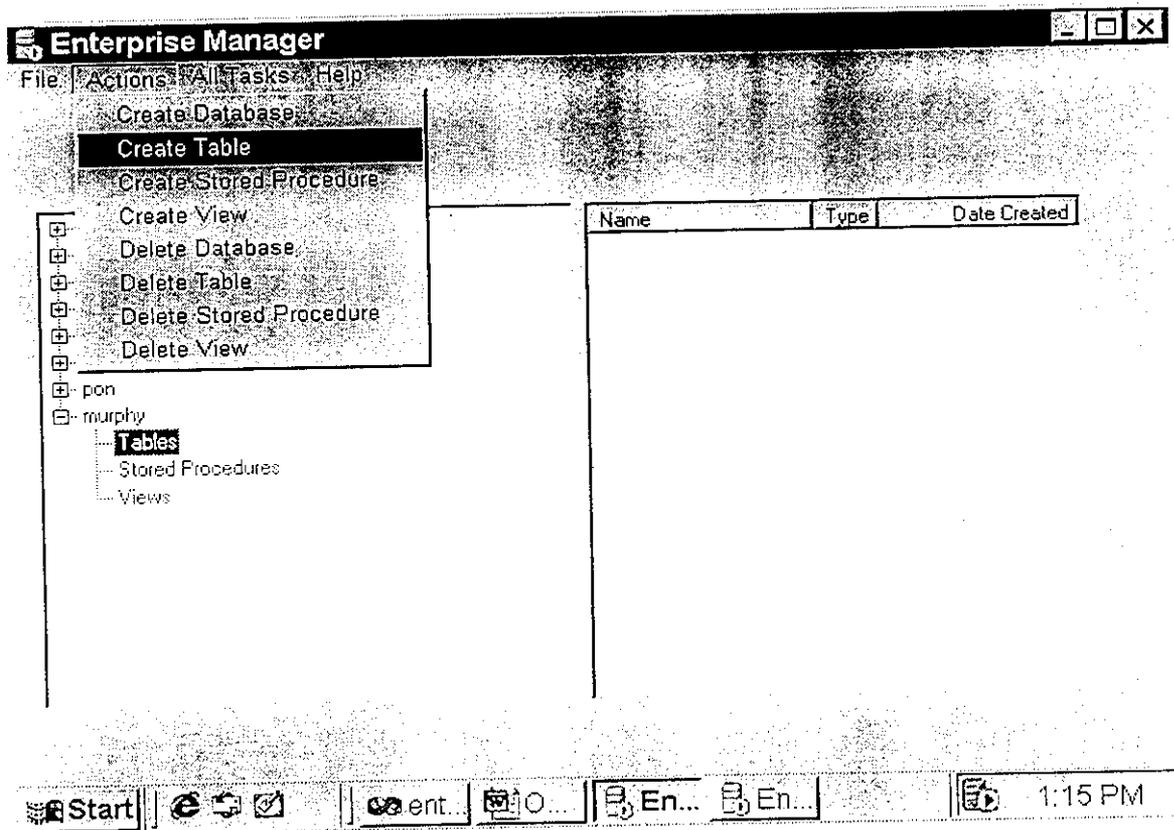


Step 2: Database Deleted.

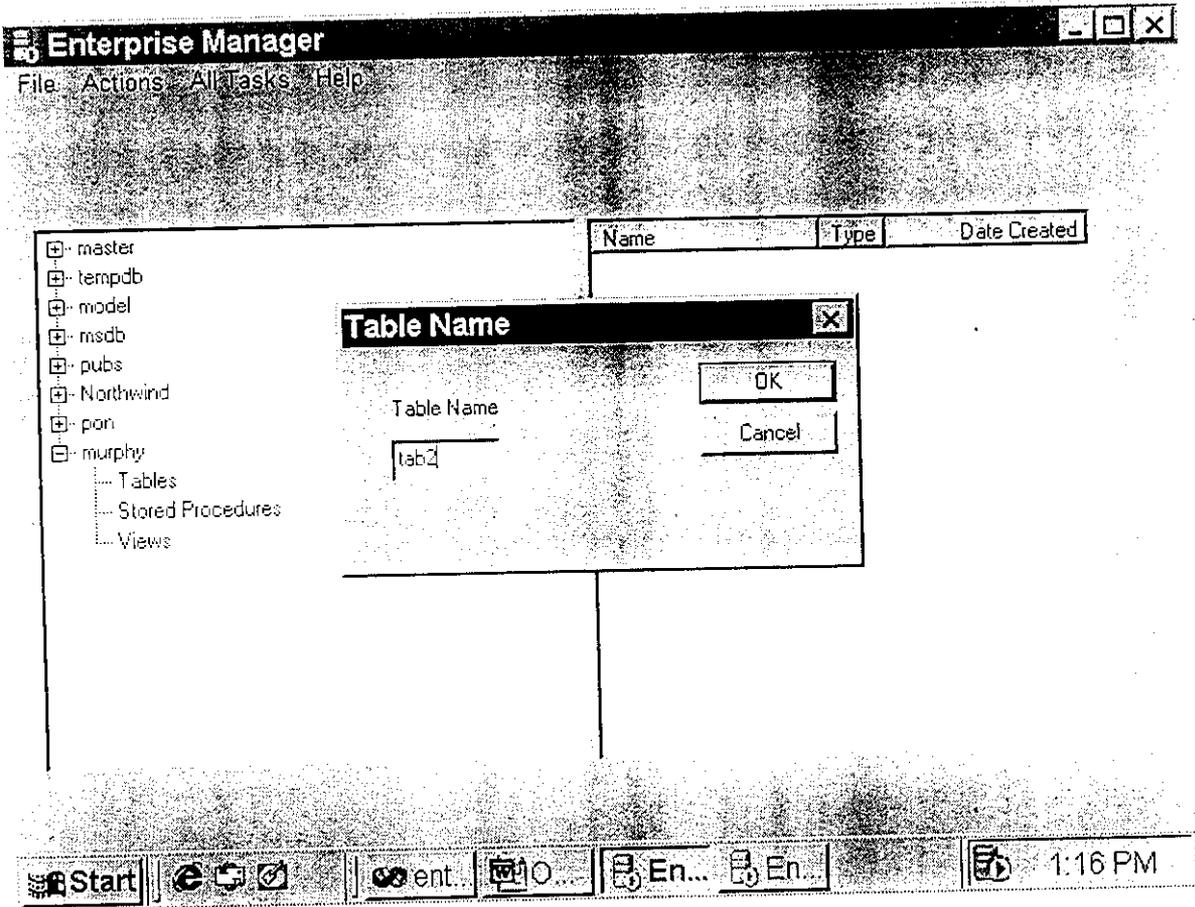


CREATE TABLE.

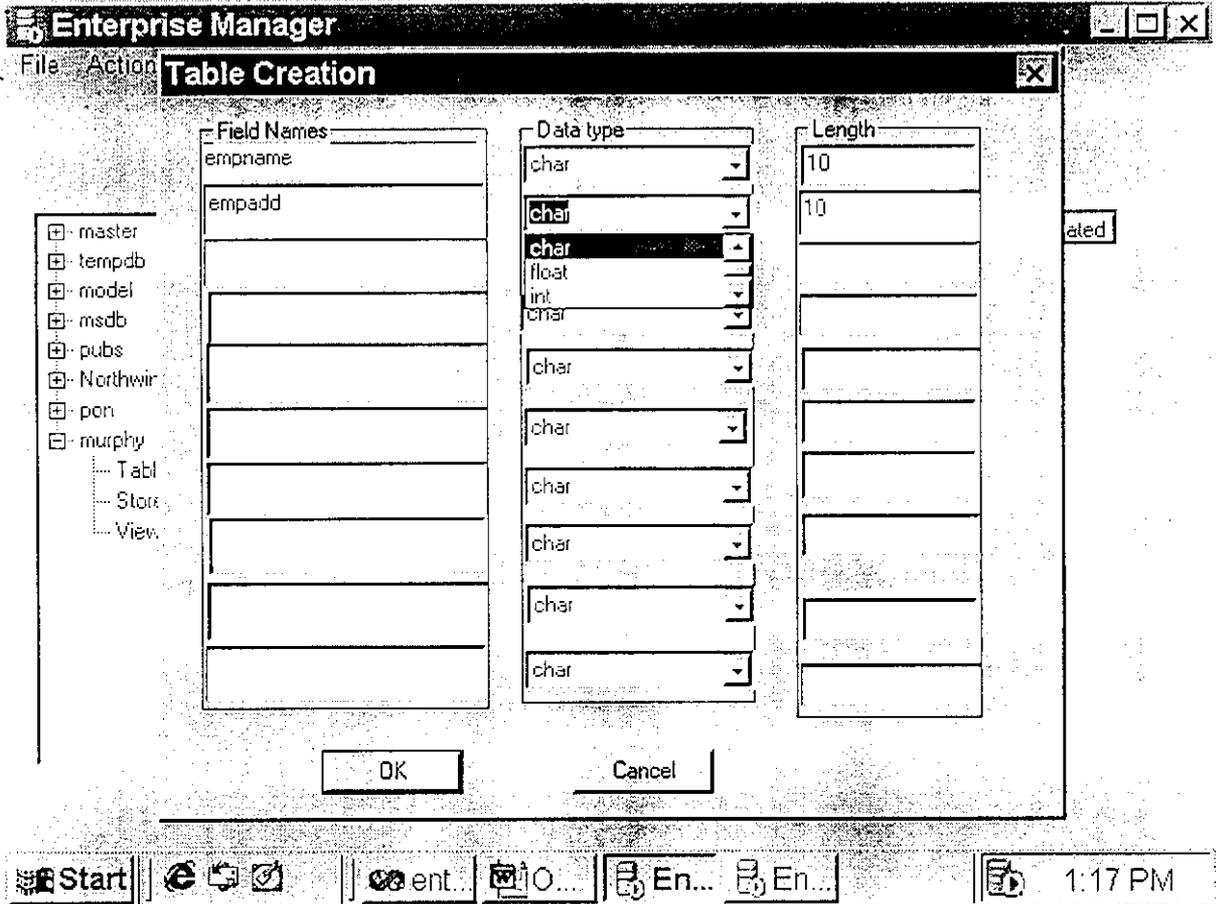
Step 1: Select *Create Table* from *Action* menu.



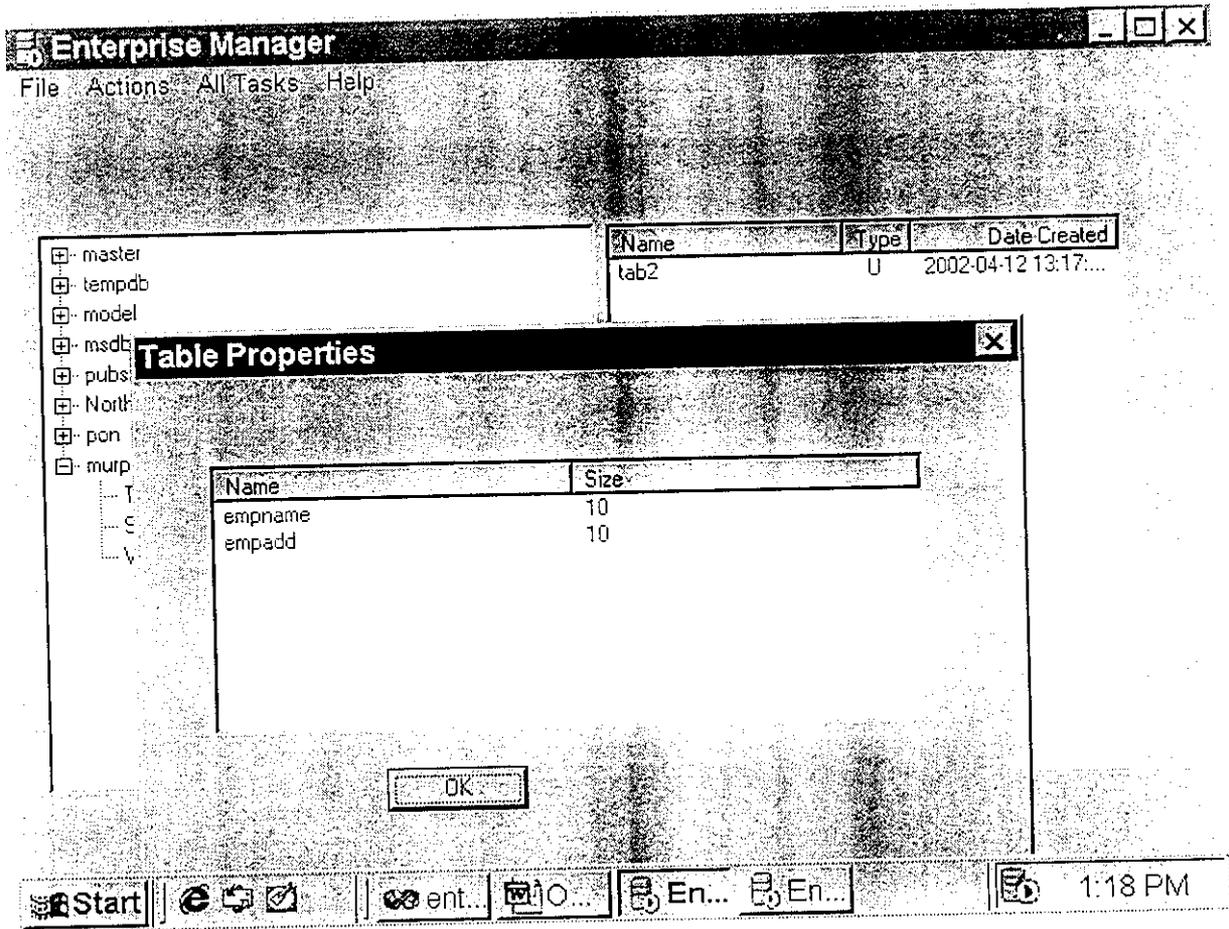
Step 2: Enter the Table Name.



Step 3: Enter the Fields for Table Creation.

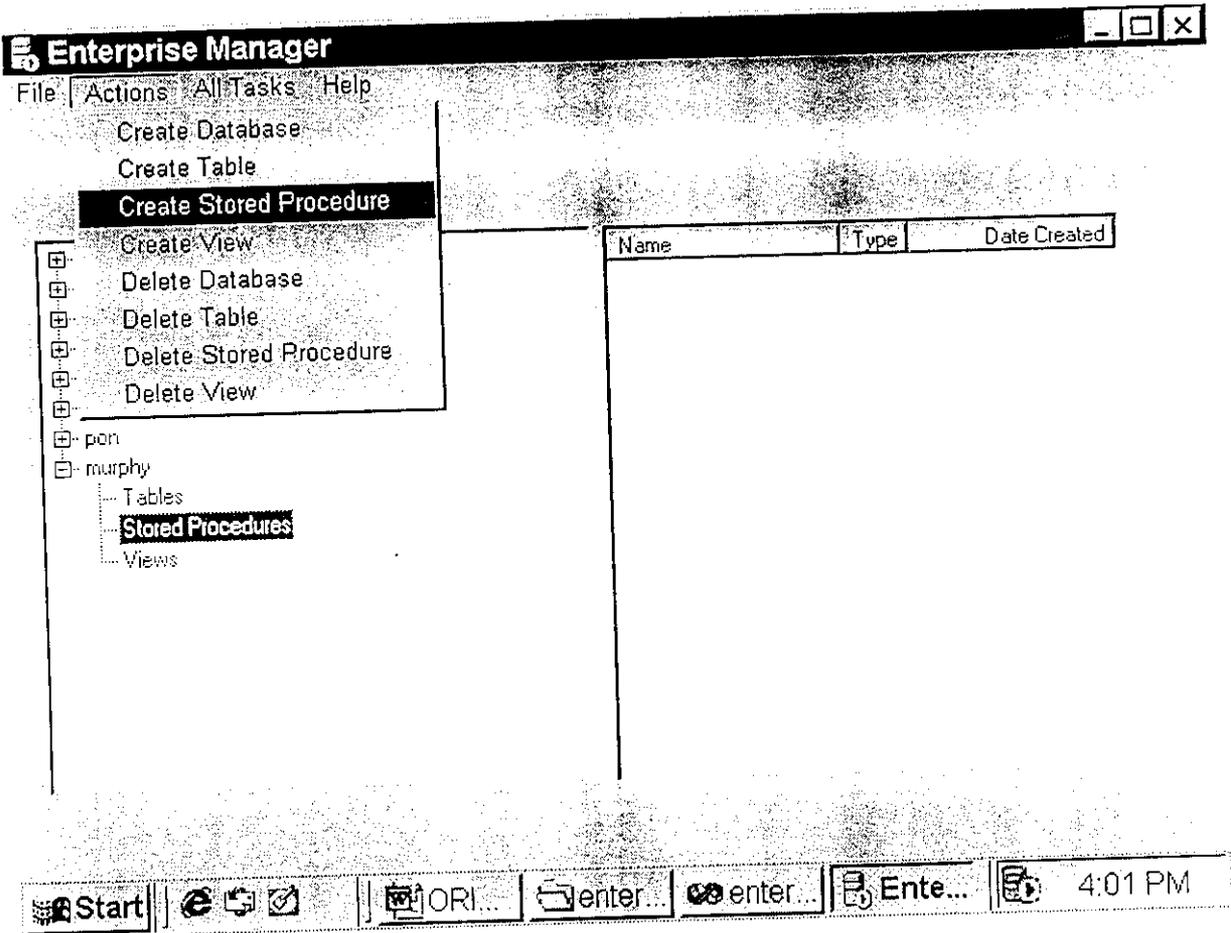


Step 4: Click the table name, it will display the Table Properties.

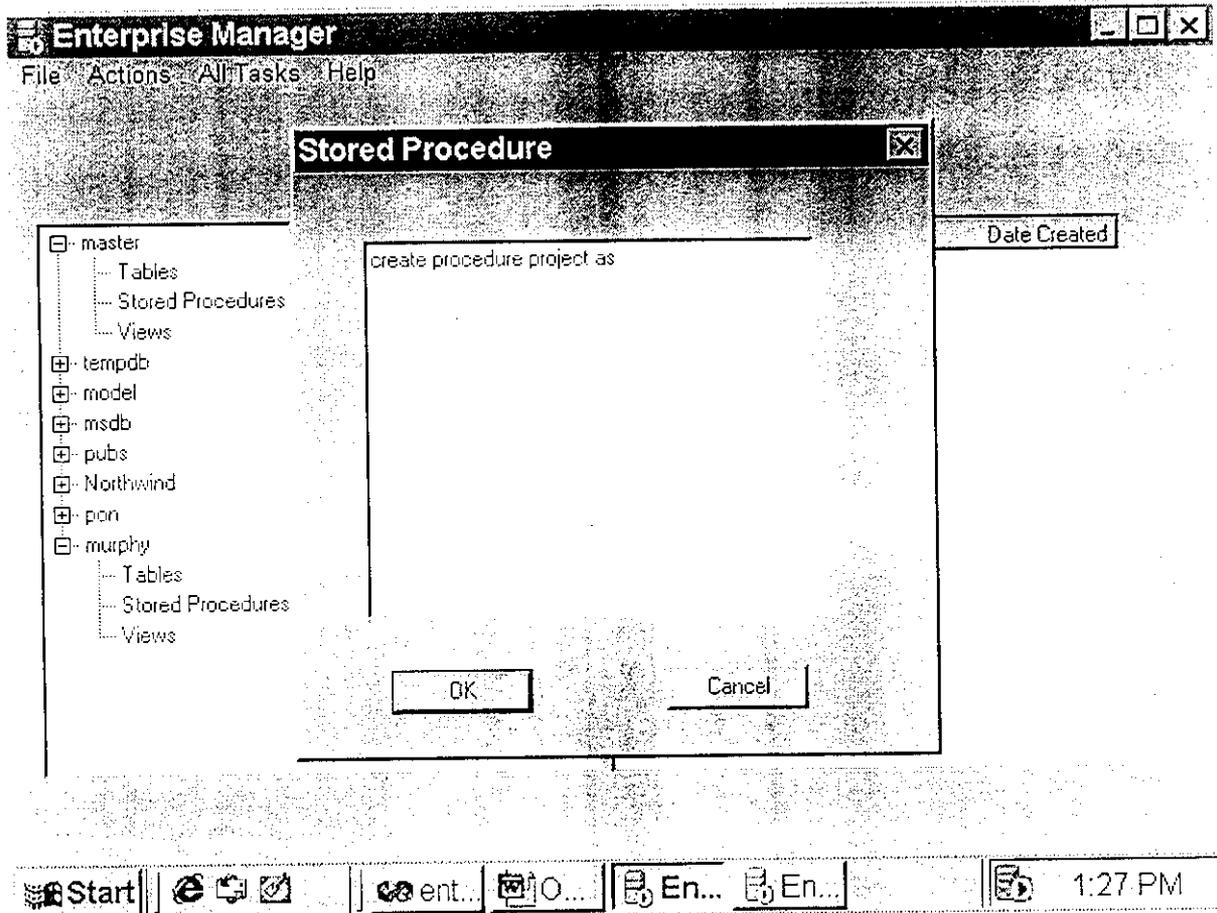


CREATE STORED PROCEDURE.

Step 1: Select *Create Stored Procedure* from Action menu.



Step 2: Write the codes for create stored procedure.



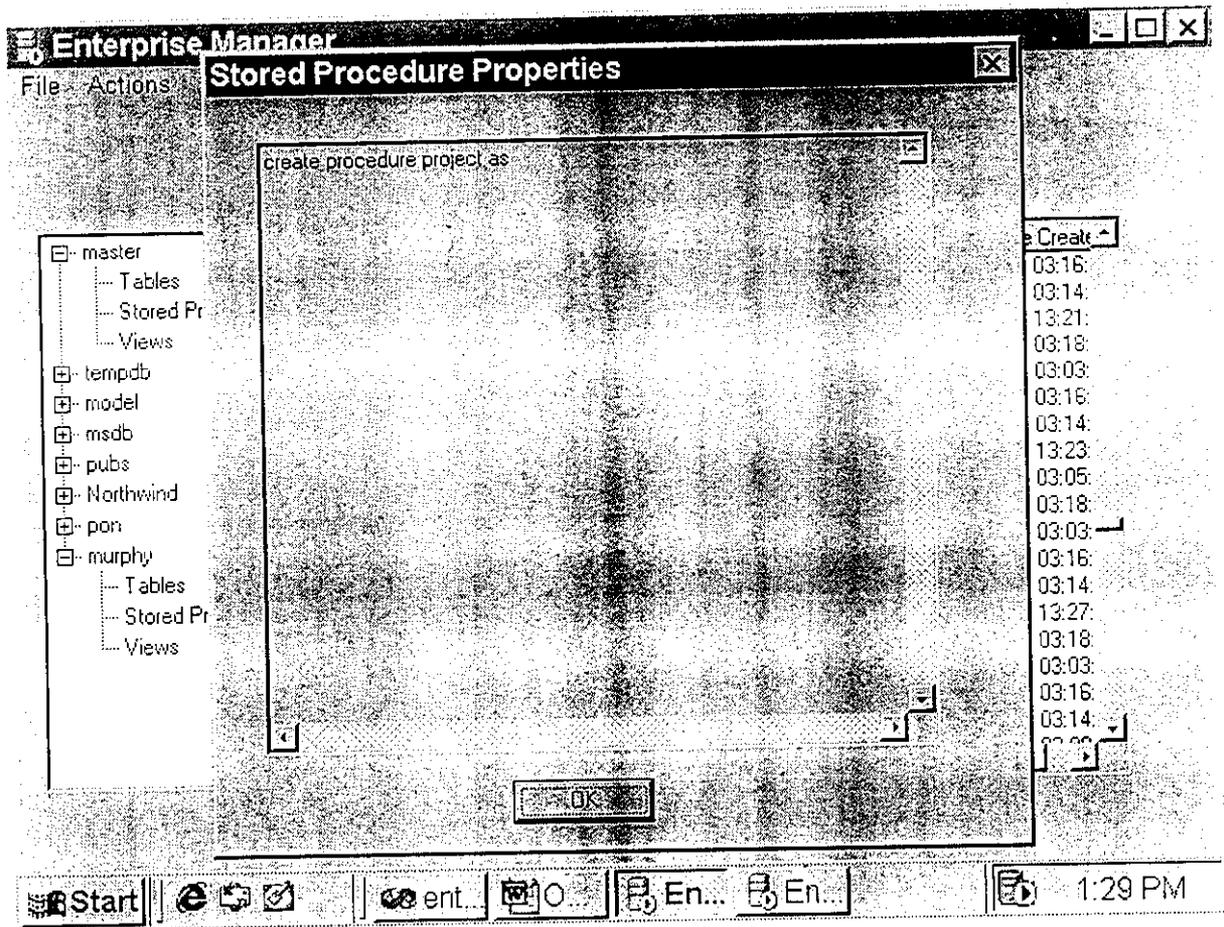
Step 3: It will store in the store procedure view

The screenshot shows the Microsoft Enterprise Manager interface. On the left, a tree view displays the database structure, including folders for 'Tables', 'Stored Procedures', and 'Views' under the 'murphy' database. The main pane on the right shows a list of stored procedures with columns for Name, Type, and Date Created. The 'project' procedure is highlighted in the list.

Name	Type	Date Create
sp_replica	P	1998-11-13 03:16
sp_MSgetrephick	P	1998-11-13 03:14
mur	P	2002-04-12 13:21
sp_scriptupdateparams	P	1998-11-13 03:18
sp_helpprotect	P	1998-11-13 03:03
sp_addpublisher	P	1998-11-13 03:16
sp_MSreplicheck_public...	P	1998-11-13 03:14
mmp	P	2002-04-12 13:23
sp_databases	P	1998-11-13 03:05
sp_scriptinproc	P	1998-11-13 03:18
sp_helpext	P	1998-11-13 03:03
sp_addsubscriber	P	1998-11-13 03:16
sp_MSlocktable	P	1998-11-13 03:14
project	P	2002-04-12 13:27
sp_scriptdelproc	P	1998-11-13 03:18
sp_helpuser	P	1998-11-13 03:03
sp_addsubscriber_sch...	P	1998-11-13 03:16
sp_MSenumcolumns	P	1998-11-13 03:14

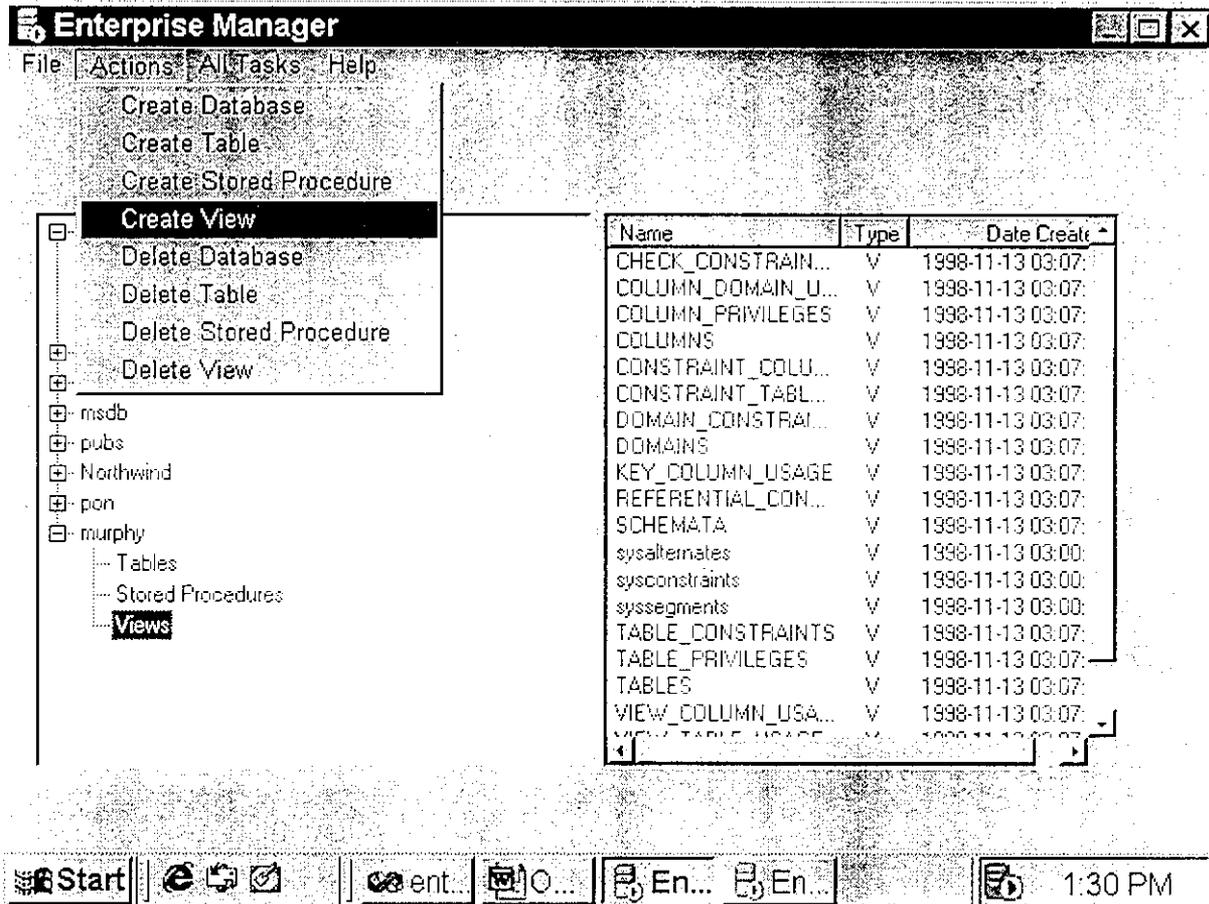
Windows taskbar at the bottom shows the Start button, several application icons, and the system clock displaying 1:28 PM.

Step 4: Click the stored procedure name, it will display
Stored Procedure Properties.

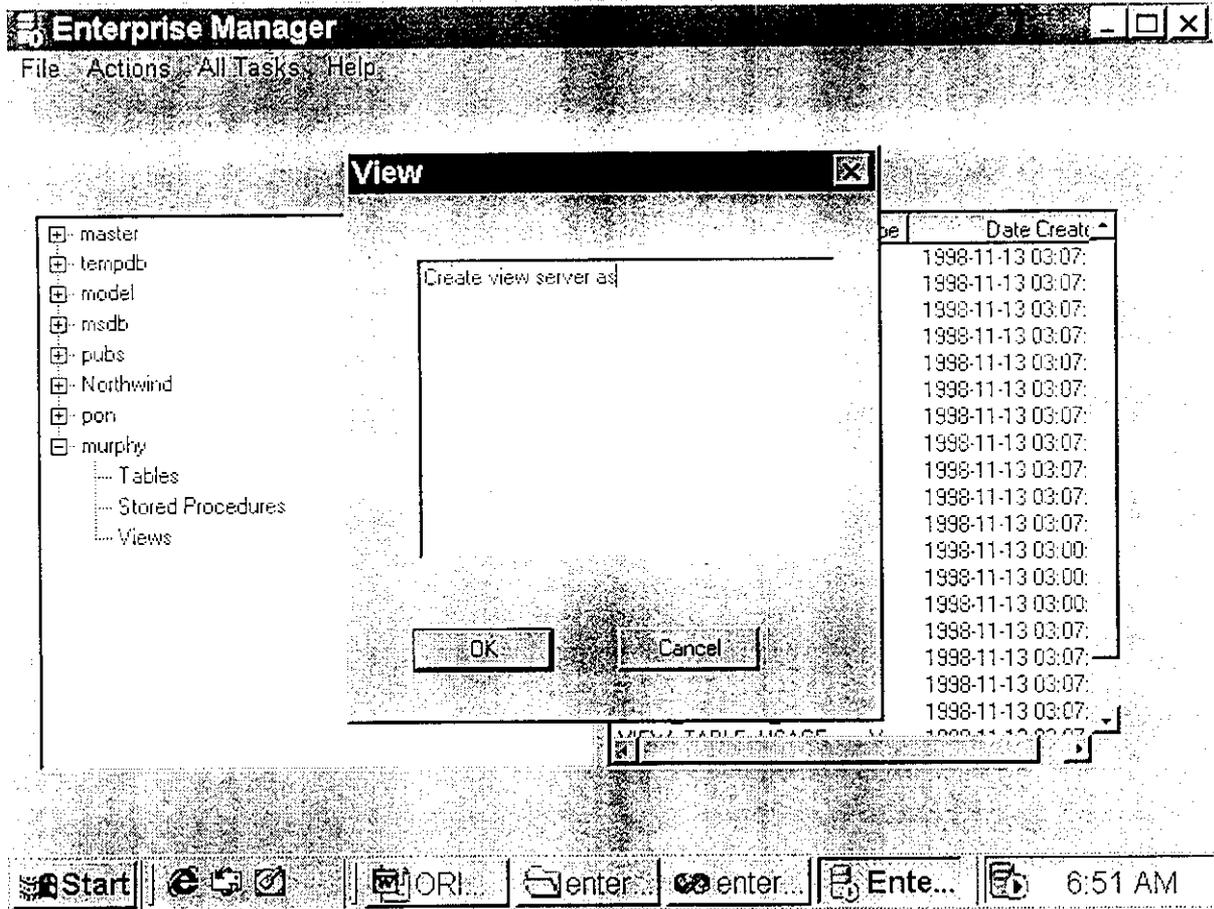


CREATE VIEW

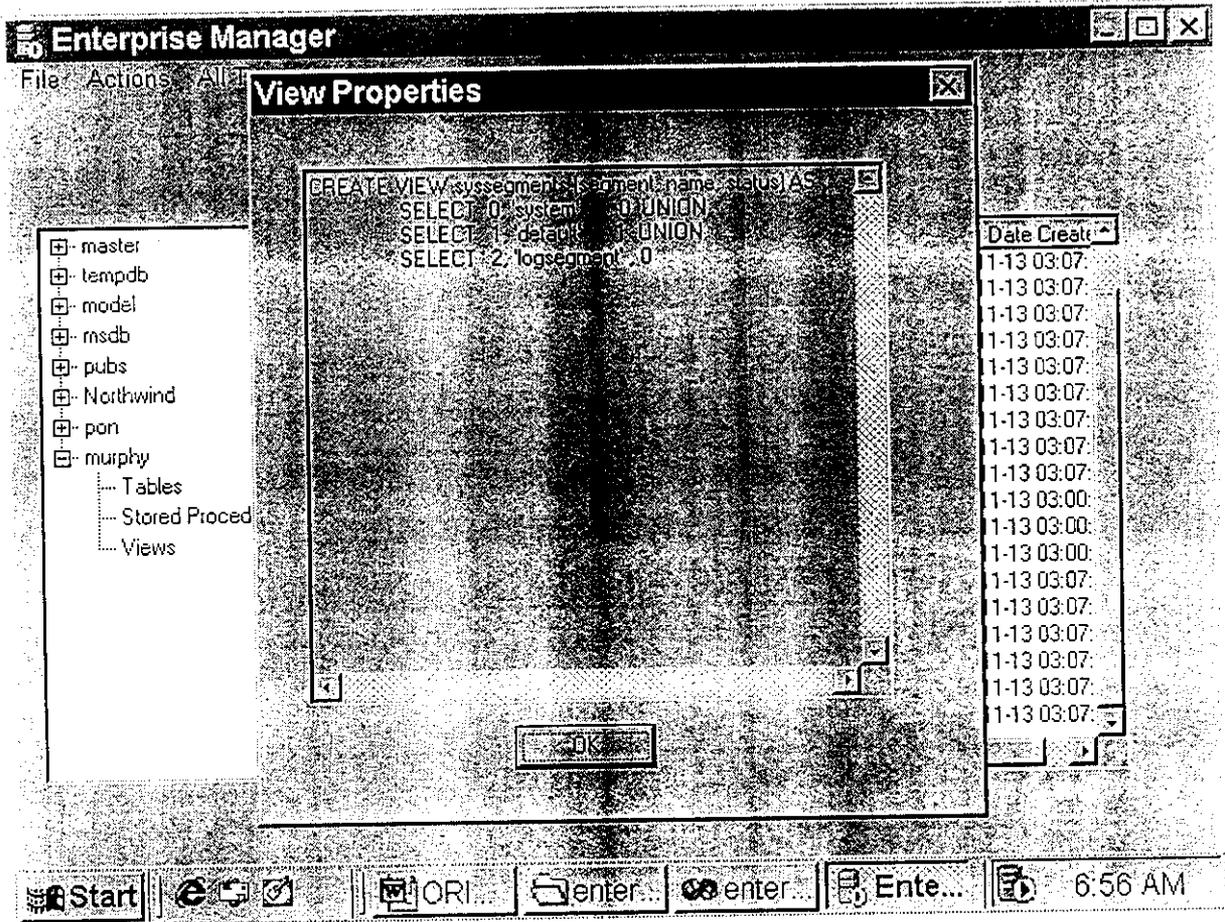
Step 1: Select the *Create view* from *Action* menu



Step 2: Write the codes for create view.

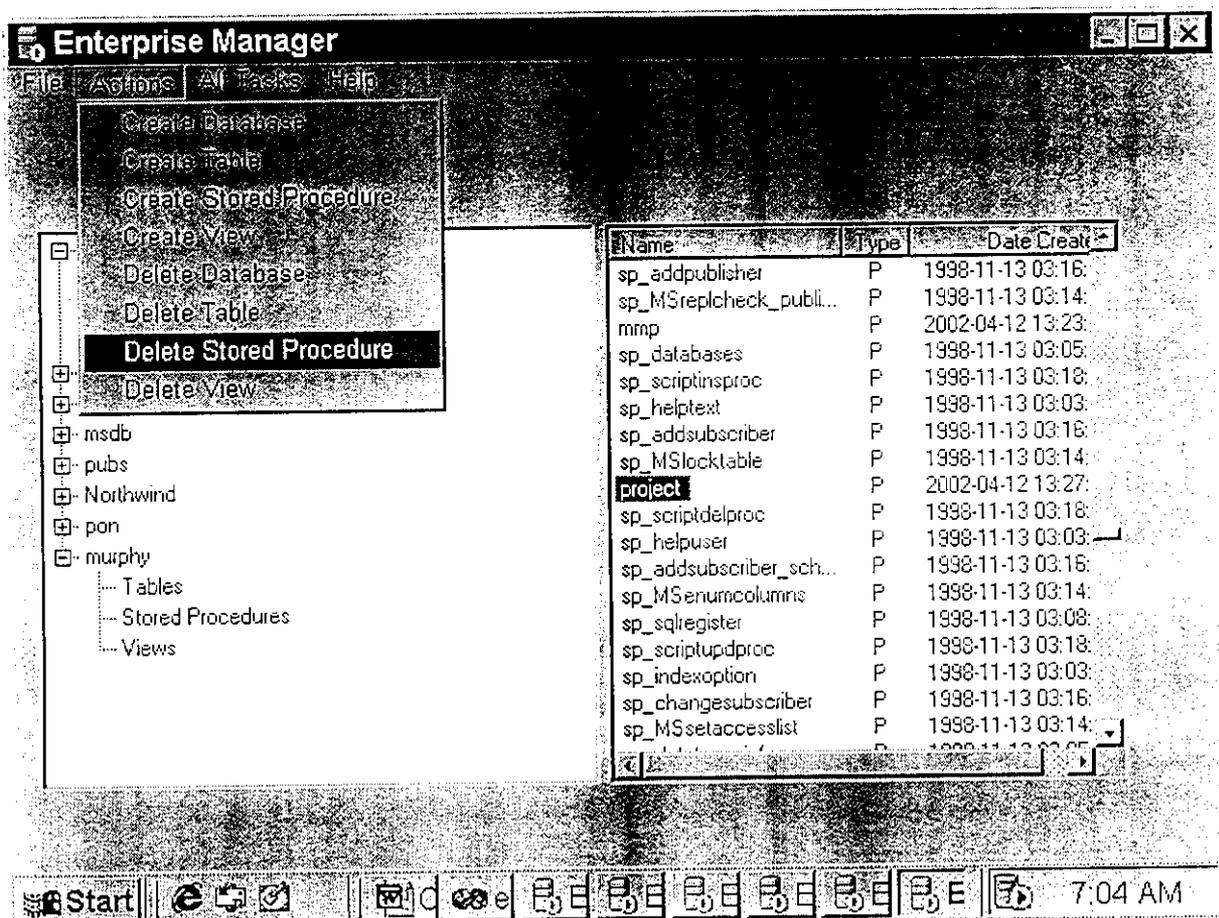


Step 3: Click the view name, it will show the View Properties

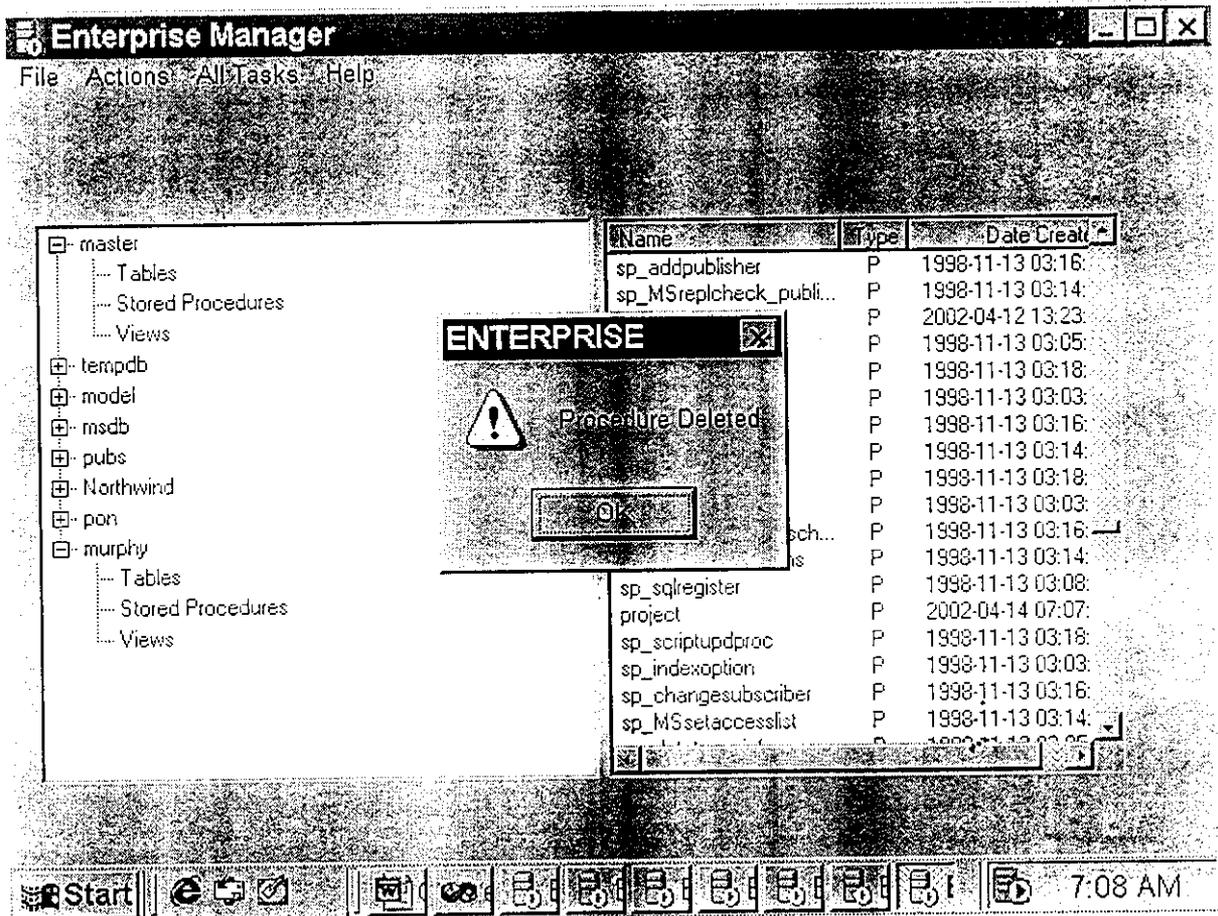


DELETE STORED PROCEDURE, TABLE & VIEW

Step 1. select the *Delete function* from *Action* menu

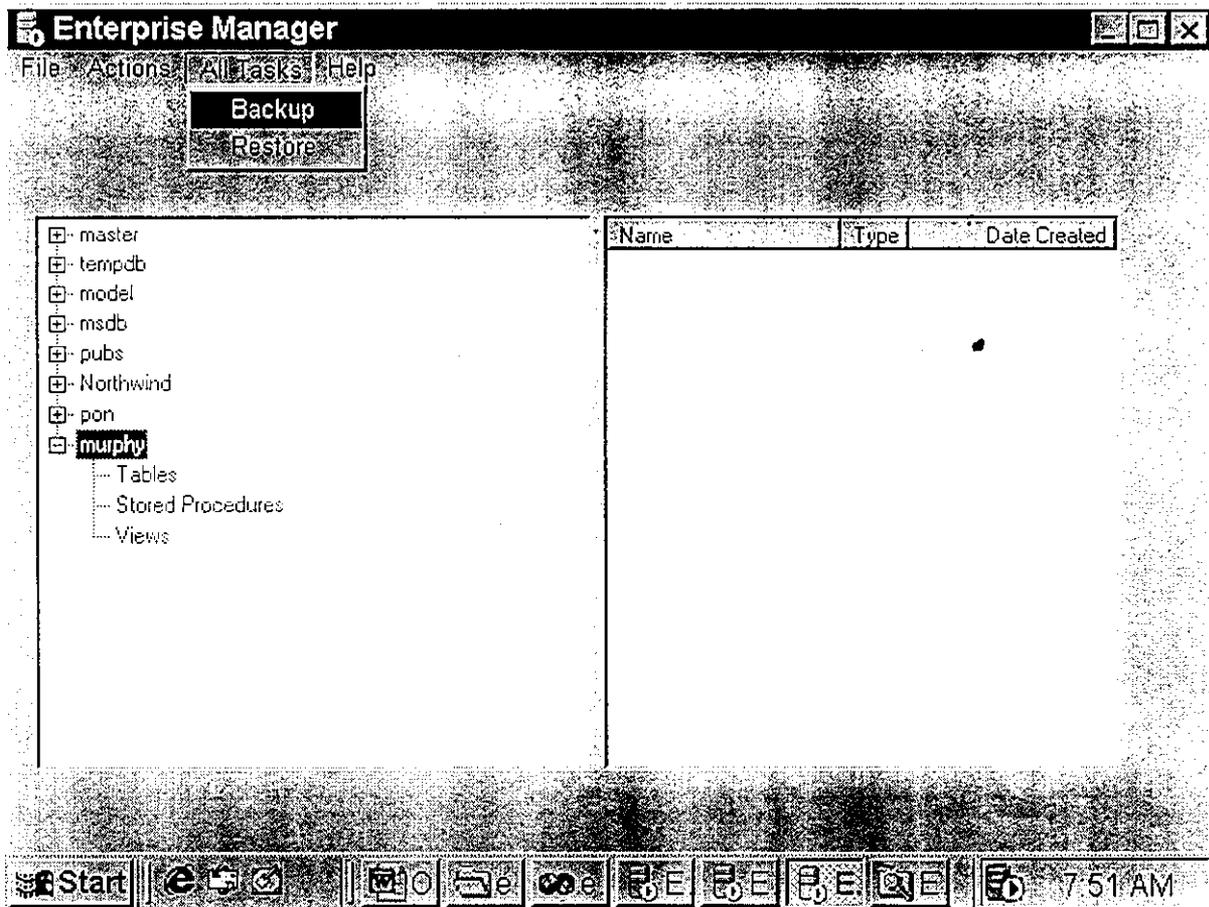


Step 2: It will display deleted message box

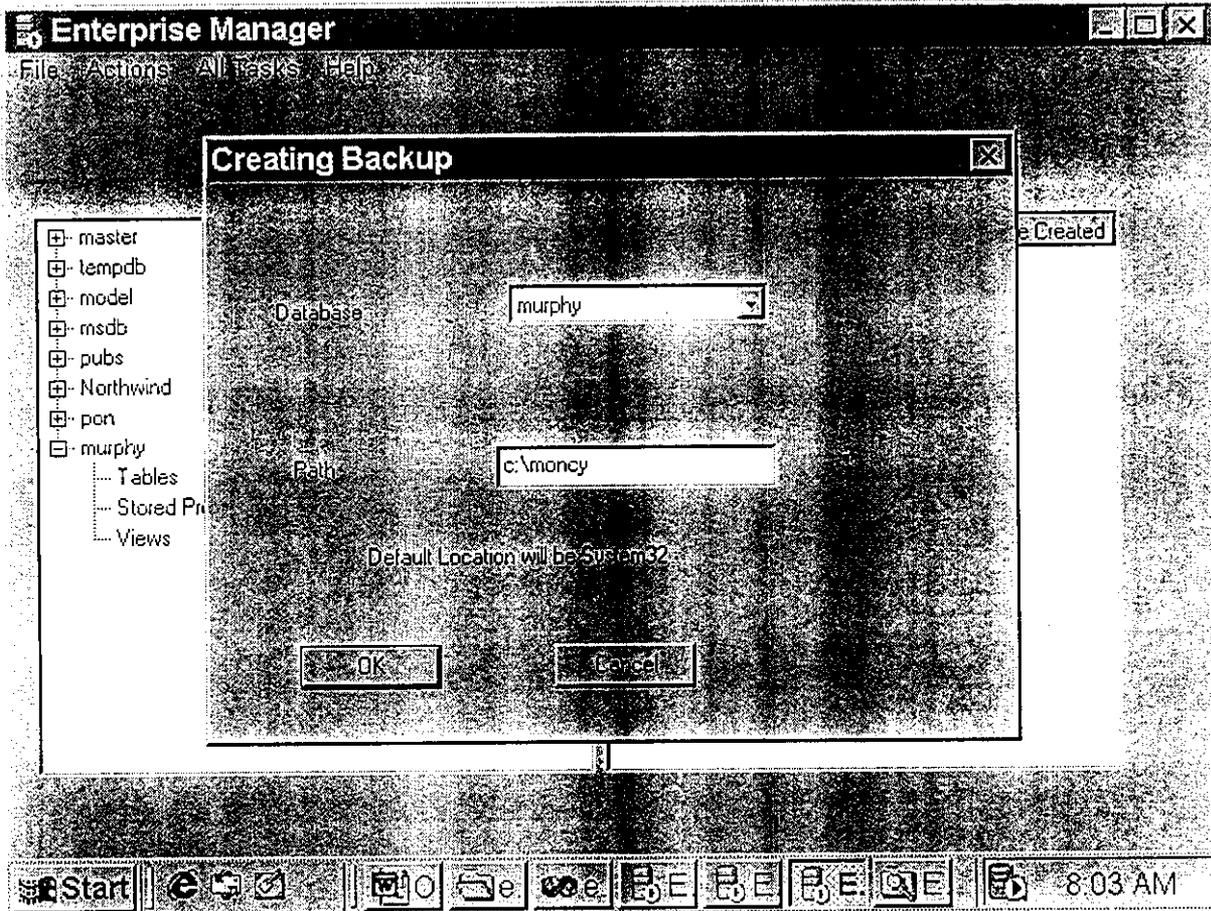


CREATE BACKUP

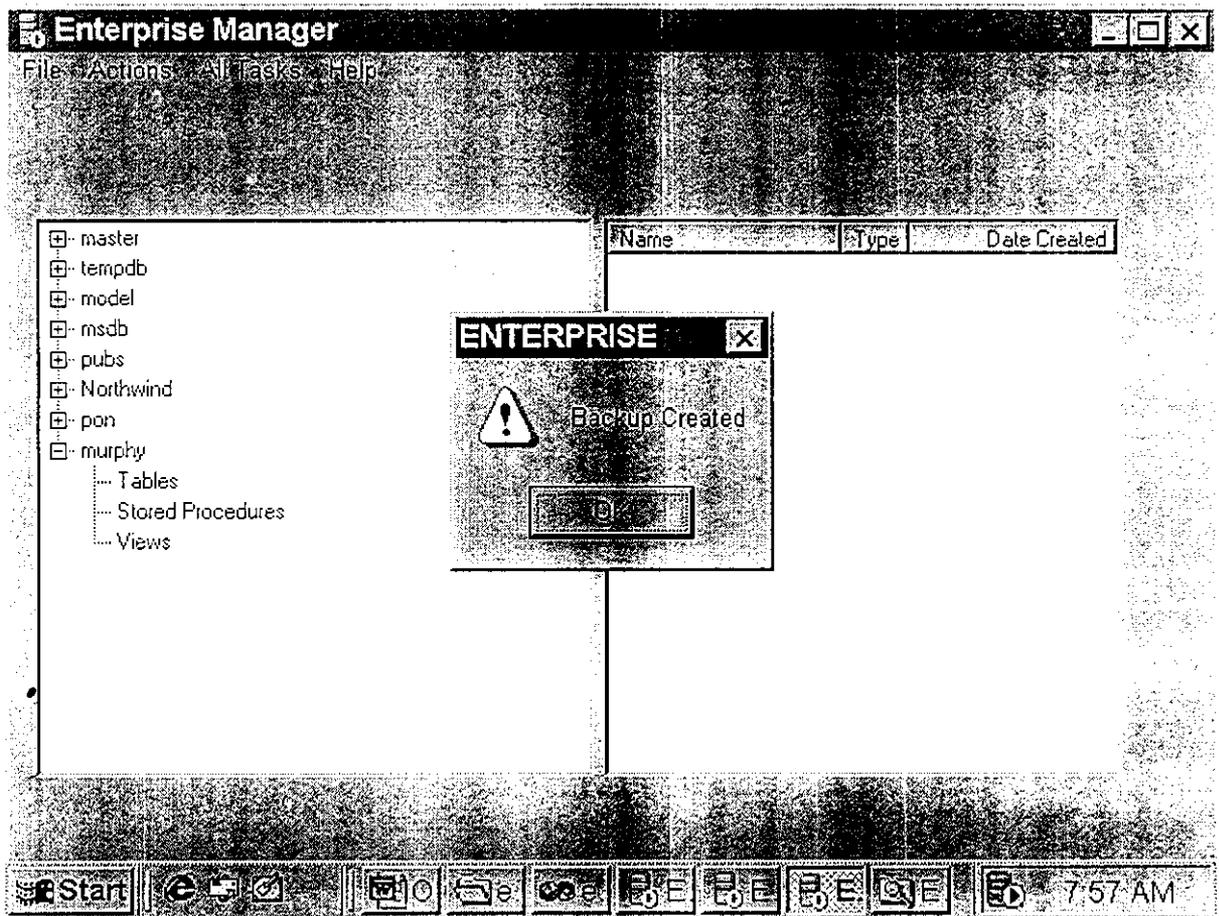
Step 1: Select the *Backup* from *All Tasks* menu.



Step 2: Enter the database name and location.

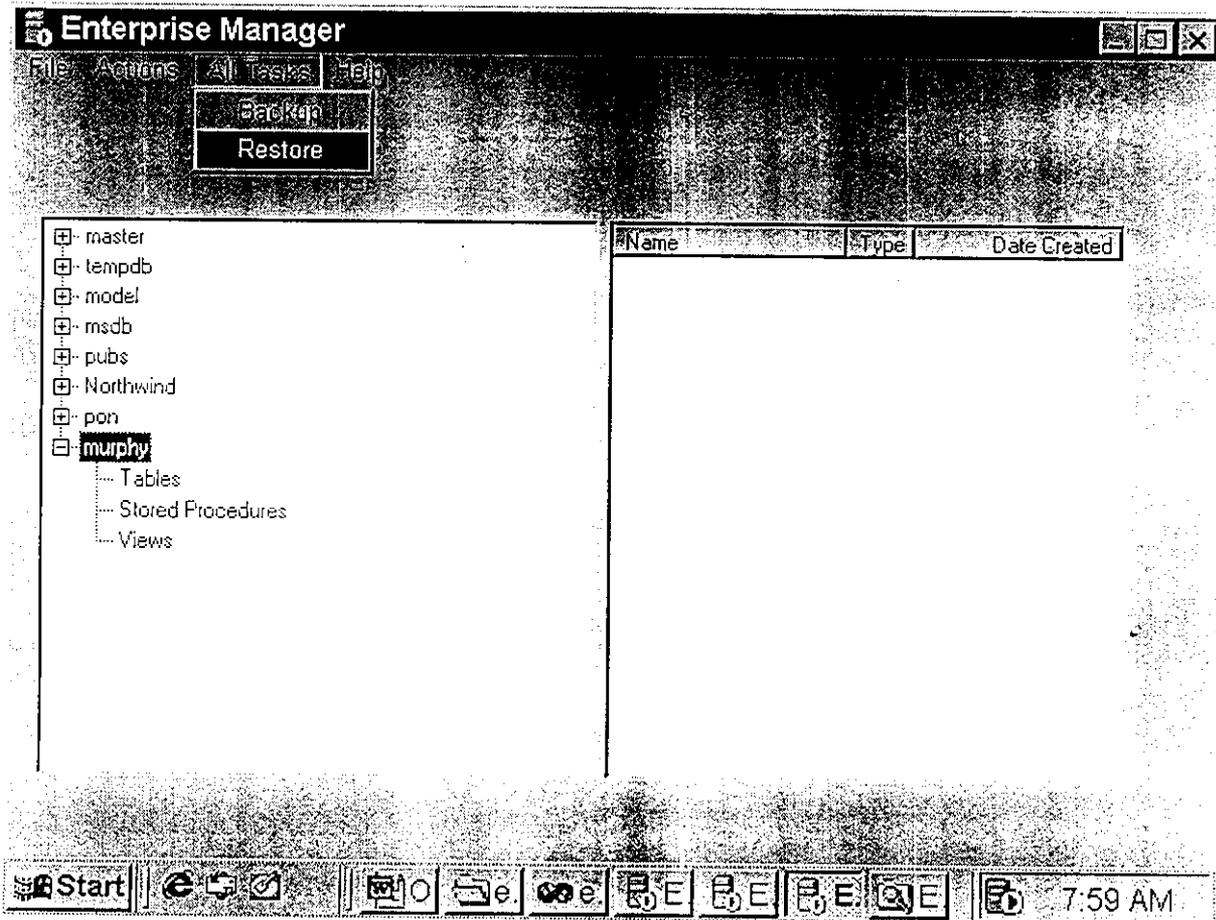


Step 3: It will display backup created message box.

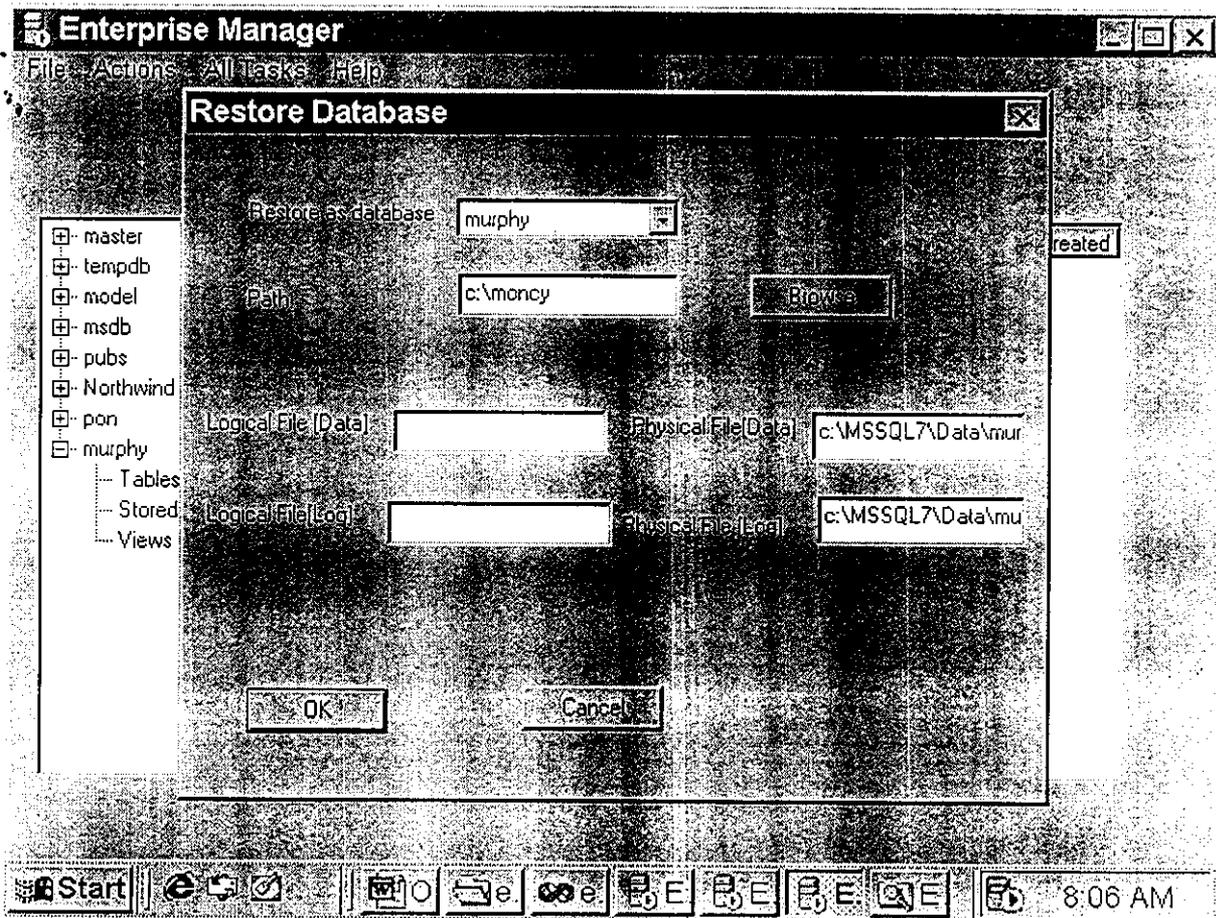


RESTORE DATABASE.

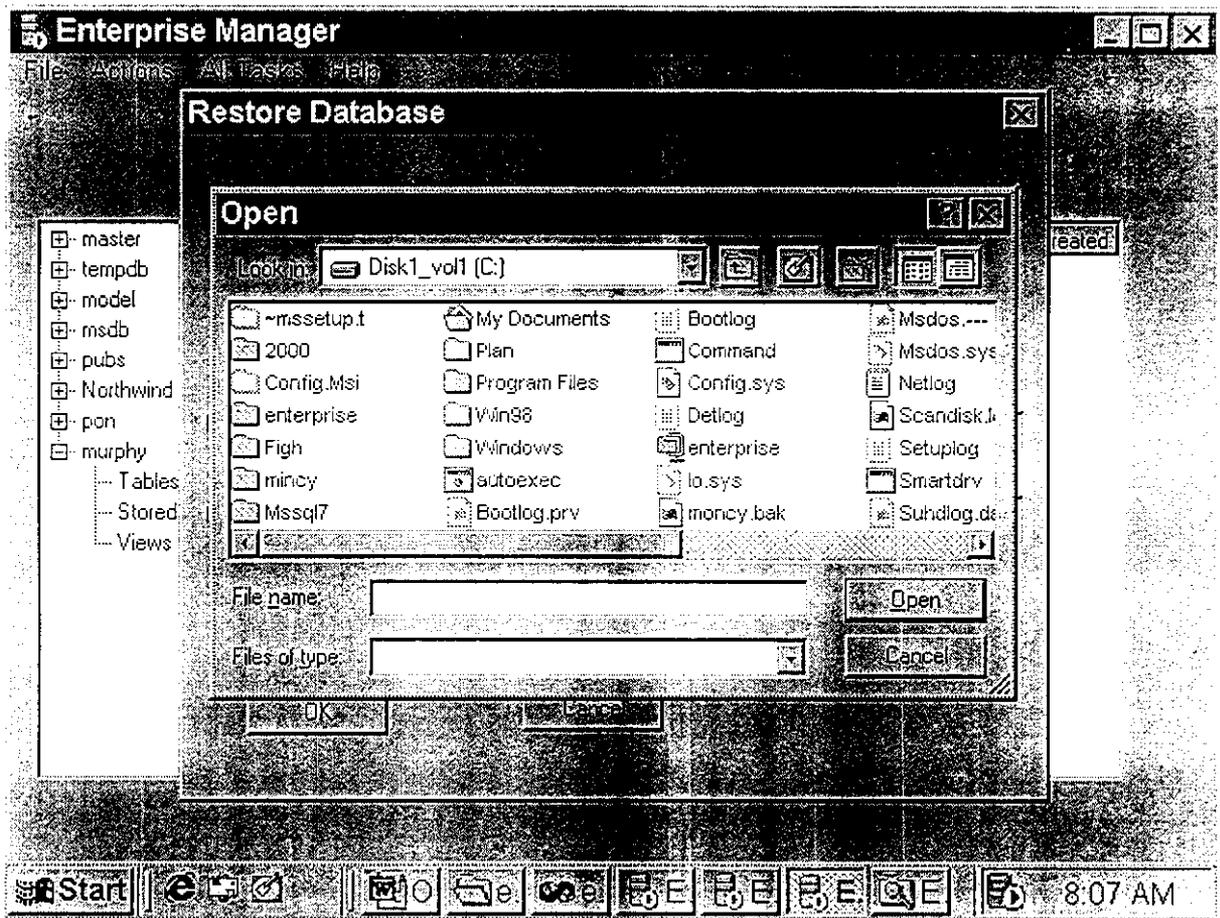
Step 1: Select the *Restore* from *All Tasks* menu



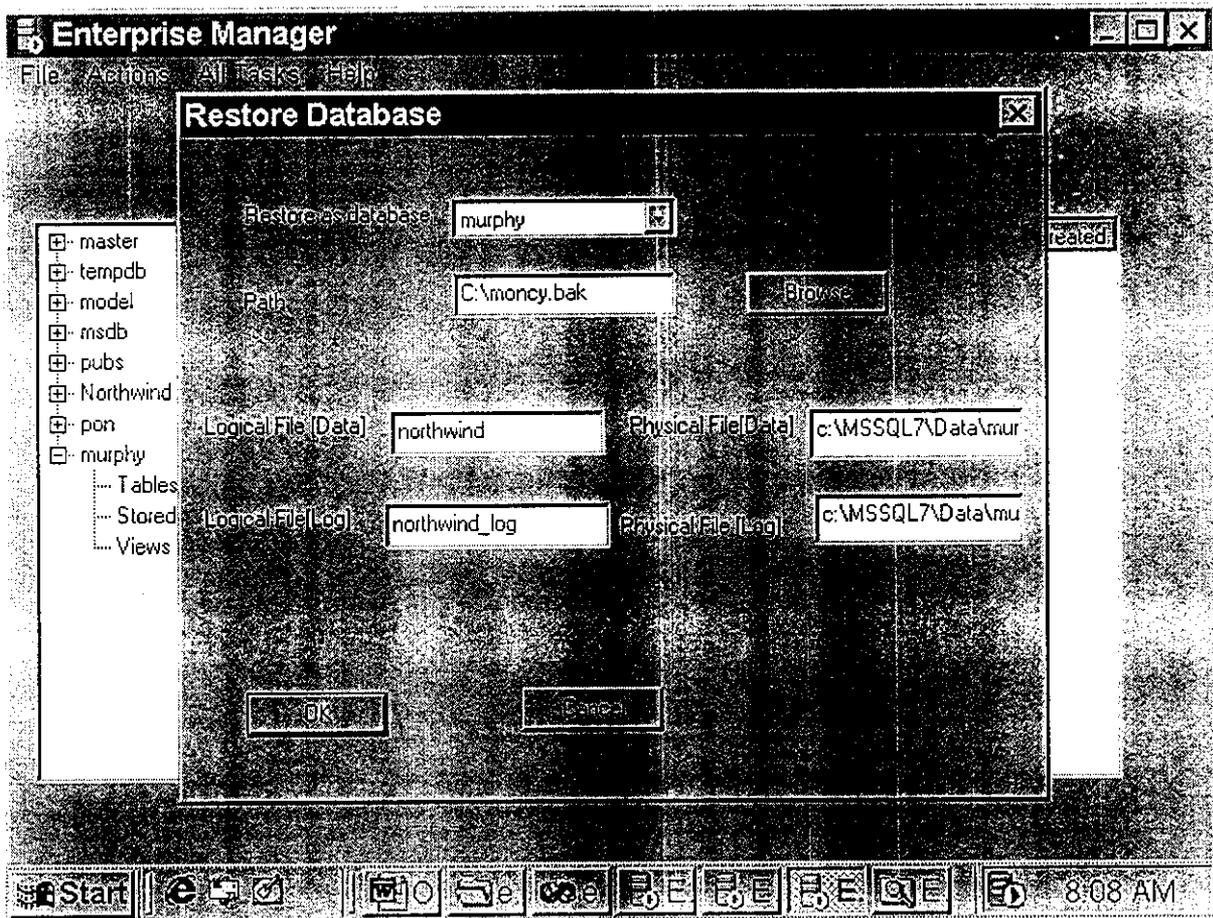
Step 2: Enter the database name and path, then click
Browse



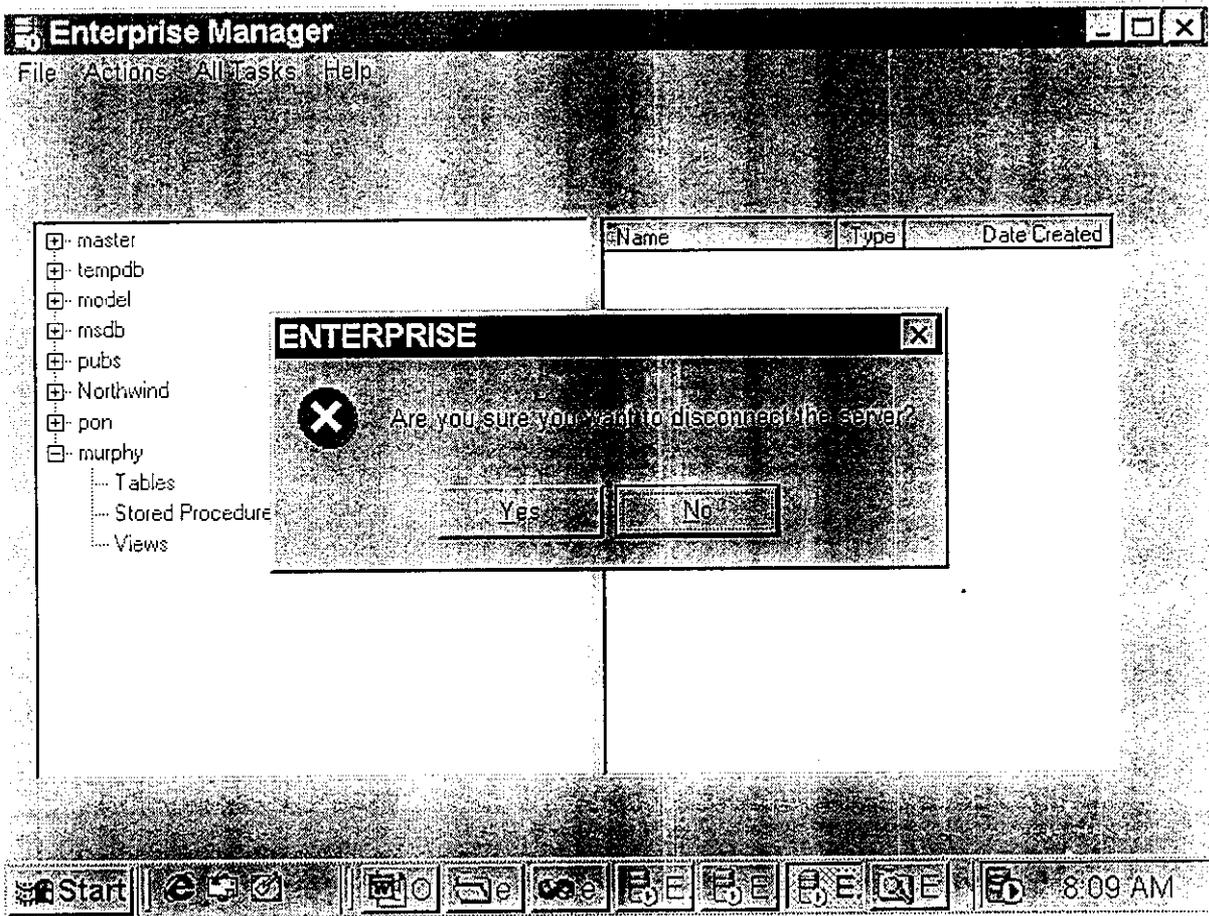
Step 3: It will open the location of backup file.



Step 4: Enter the logical data name.



DISCONNECT MENU.



EXIT MENU.

