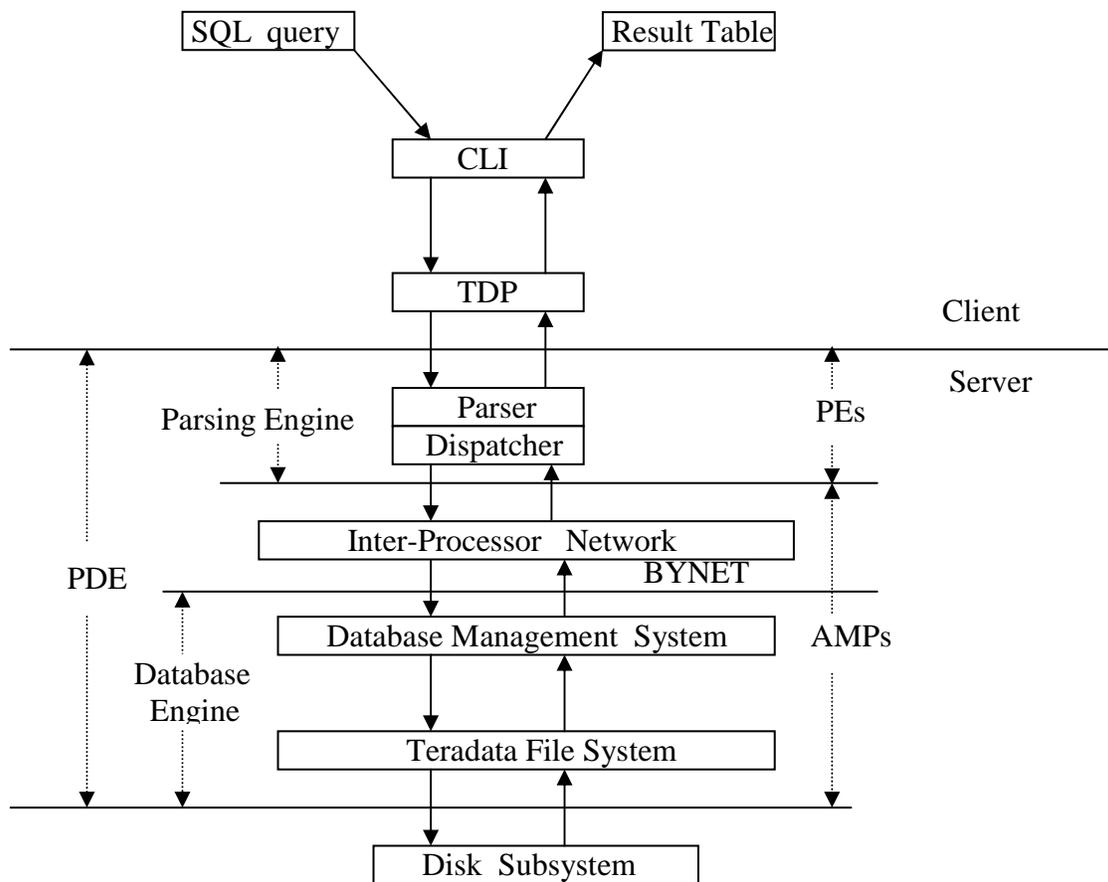


## Introduction to Teradata RDBMS

Teradata RDBMS is a complete relational database management system. The system is based on off-the-shelf Symmetric Multiprocessing (SMP) technology combined with a communication network connecting the SMP systems to form a Massively Parallel Processing (MMP) system. BYNET is a hardware inter-processor network to link SMP nodes. All processors in a same SMP node are connected by a virtual BYNET. We use the following figure to explain how each component in this DBMS works together.



### PDE (Parallel Database Extensions):

This component is an interface layer on the top of operating system. Its functions include: executing vprocs (virtualprocessors), providing a parallel environment, scheduling sessions, debugging, etc.

**Teradata File System:**

It allows Teradata RDBMS to store and retrieve data regardless of low-level operating system interface.

**PE (Parsing Engine):**

- Communicate with client
- Manage sessions
- Parse SQL statements
- Communicate with AMPs
- Return result to the client

**AMP (Access Module Processor):**

- BYNET interface
- Manage database
- Interface to disk subsystem

**CLI (Call Level Interface):**

A SQL query is submitted and transferred in CLI packet format

**TDP (Teradata Director Program):**

Route the packets to the specified Teradata RDBMS server

Teradata RDBMS has the following components that support all data communication management:

- Call Level Interface ( CLI )
- WinCLI & ODBC
- Teradata Director Program ( TDP for channel attached client )
- Micro TDP ( TDP for network attached client )

## Introduction to Teradata Client Software

Teradata Client Software components include:

- **BTEQ**  
Basic Teradata Query is a general-purpose program that you can use to submit data, commands and SQL statements to Teradata RDBMS.

- **C / COBOL / PL/I preprocessors**  
If you use embedded SQL to develop client applications, you need these tools to pre-compile your programs.
- **CLI**
- **ODBC**
- **TDP/MTDP/MOSI**
- **Achieve/Restore data to/from tape (ASF2)**
- **Queryman**  
Queryman is based on ODBC, you can logon through a DSN and enter any SQL statement to manipulate the data in database.
- **FastLoad**
- **MultiLoad**
- **FastExport**
- **Open Teradata Backup (OTB)**
- **Tpump**
- **Teradata Manager**
- **WinDDI**

All client components are based on CLI or ODBC or Both of them. So, once you install the client software, you should configure these two components appropriately before you execute these client utilities.

Teradata RDBMS is able to support JDBC programs in both forms of application and applet. The client installation manual mentions that we need to install JDBC driver on client computers, and we also need to start a JDBC Gateway and Web server on database server. Teradata supports at least two types of JDBC drivers. The first type can be loaded locally and the second one should be downloadable. In either of two ways, to support your development, you need local JDBC driver or Web server / JDBC Gateway running on the same node on which Query Manager is running. But in the setup CD we received, there is no JDBC driver and any Java Development Tools. Moreover, Web server is not started on our system yet.

## Installation and Configuration

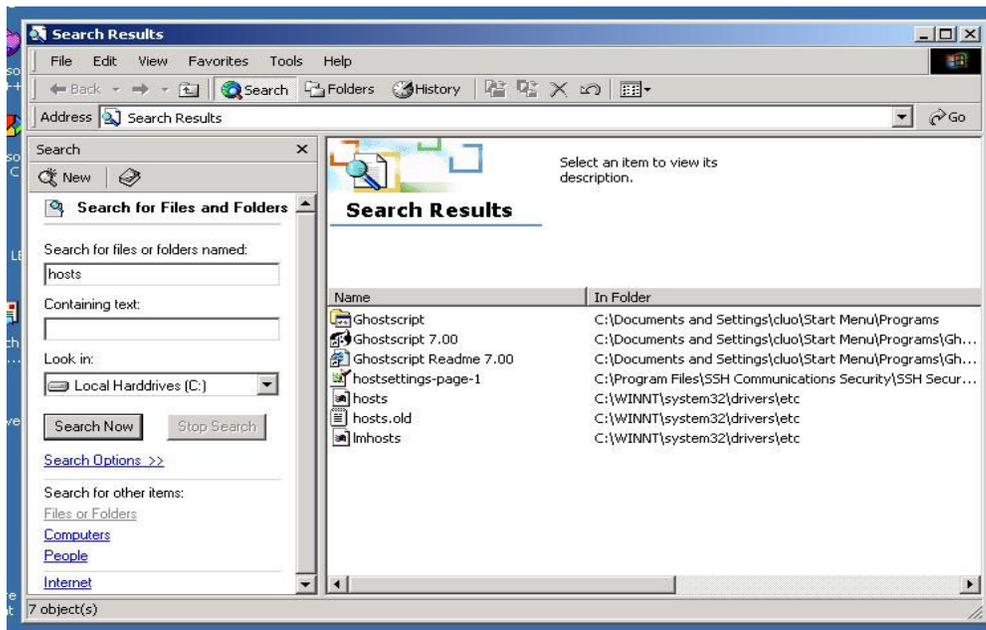
Because the installation is easy, we just give some tips:

- You need one floppy disk which contains licenses for all components you could install. Each component has one entry in the license.txt file.
- If you are asked to choose ODBC or Teradata ODBC with DBQM enhanced version to install. Just forget it. In this case, you can not install DBQM\_Admin, DBQM\_Client, and DBQM\_Server. These three components are used to optimize the processing of your SQL queries. Fortunately, your client software still works smoothly without them.
- Because CLI and ODBC are the infrastructures of other components, you may not delete either one of them from the installation list if there is any component based on it.
- After ODBC installation, you will be asked to run ODBC administrator to configure a Data Source Name (DSN). You may cancel it simply because you can do this job later. After Teradata Manager installation, you will be asked to run Start RDBMS Setup. You also can do it later.

For more information, please visit [www.info.ncr.com](http://www.info.ncr.com) and download *Teradata Client for Windows Installation Guide*.

Configuration includes several steps:

- **Setting Network Parameters**  
For Windows 2000, do the following step: Start -> Search -> For Files or Folders. You can find the file: **hosts**



Use Notepad to edit the hosts file as follows:

```

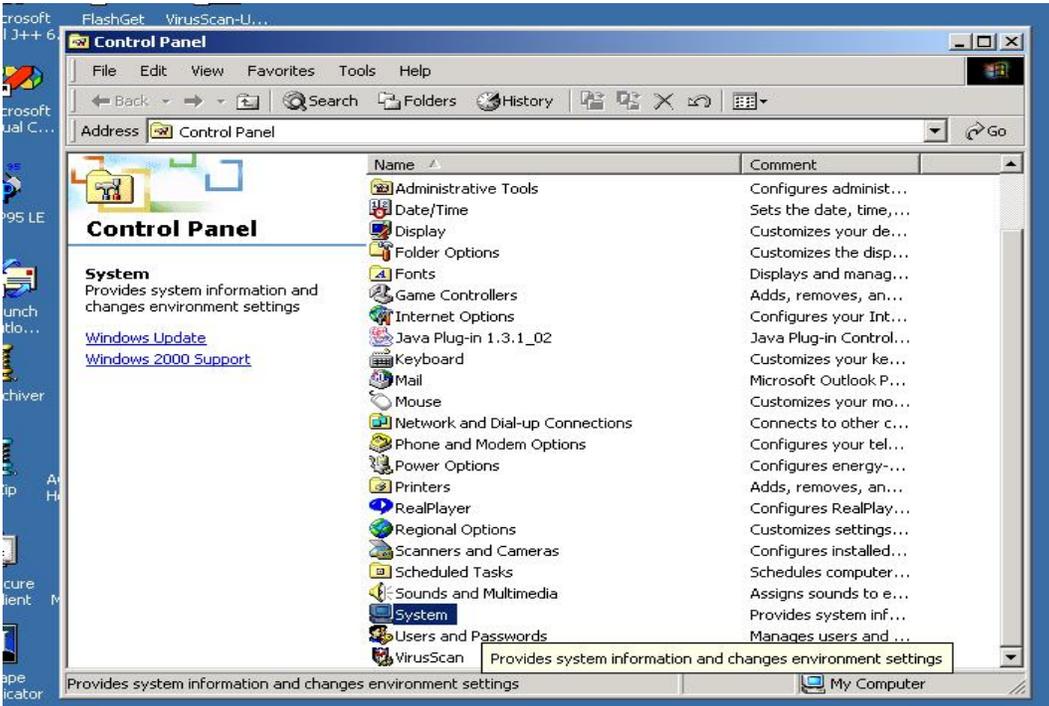
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10      x.acme.com           # x client host

127.0.0.1    localhost
130.108.57   teradatacop1

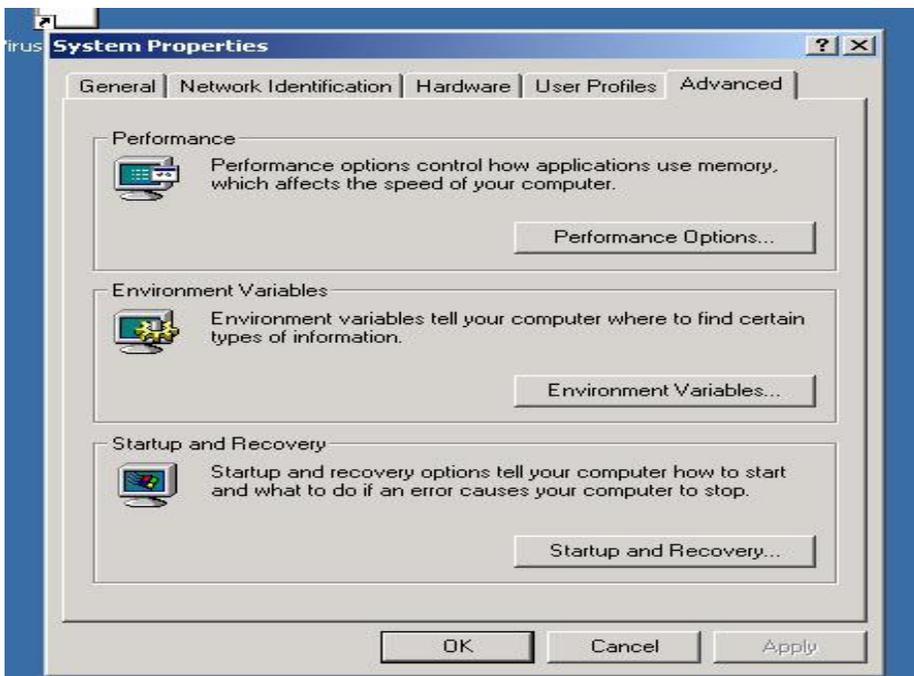
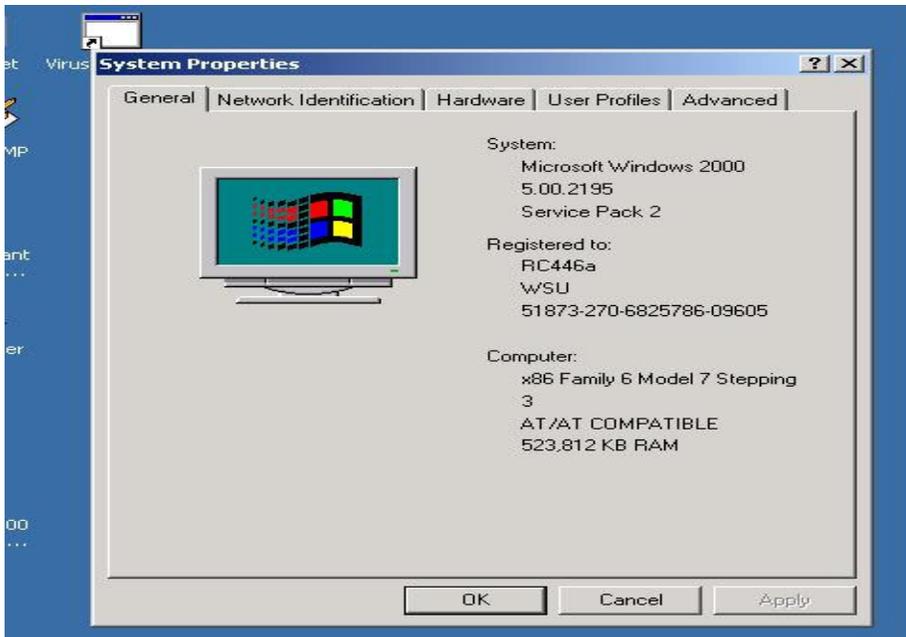
```

Add one line into your hosts file: “ 130.108.57 teradatacop1 ”. Here, 130.108.57 is the IP address of the top node of the system on which Query Manager is running. “teradata” will be the TDPID which is used in many client components you installed. “cop” is a fixed suffix string and “1” indicate that there is one RDBMS.

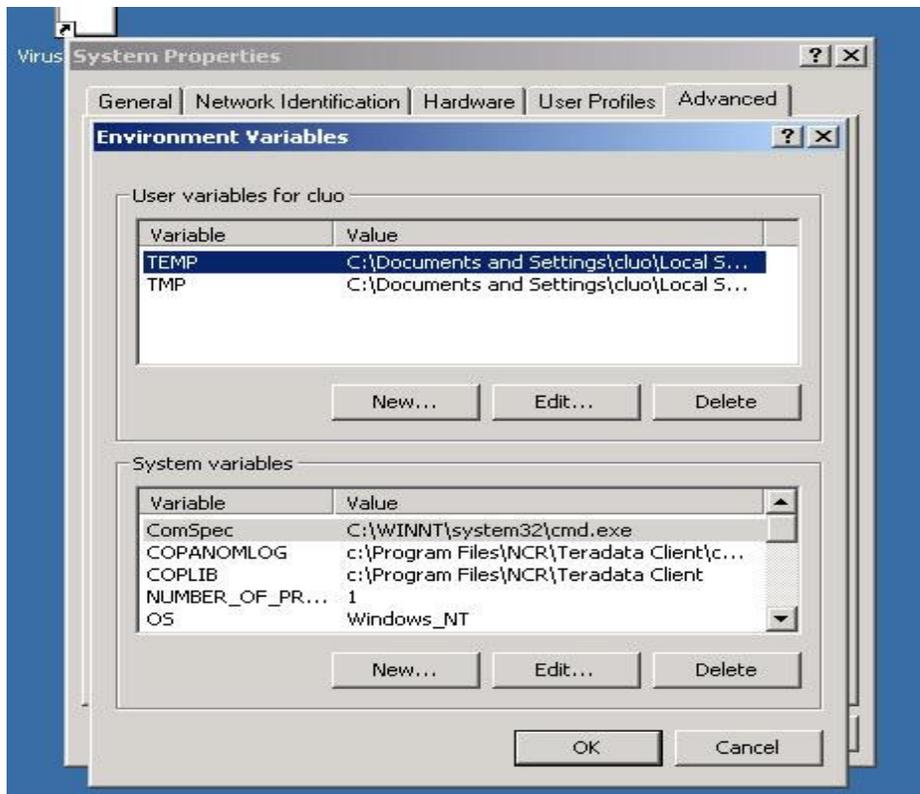
- Setting System Environment Parameters  
For Windows 2000, do the following step: Start -> Settings -> Control Panel



Find the icon “System”, double click it, get the following window, then choose “Advanced” sub-window



click "Environment Variables..." button



Then you can click button “New...” to create some System Variables as follows:

**COPLIB = the directory which contains the file clispb.dat**

This file is copied to your computer when you install Teradata Client Software. It contains some default settings for CLI.

**COPANOMLOG = the full path and the log file name**

You can set it as you want. If the file doesn’t exist, when an error occurs, the client software will create the file to record the log information.

**TDMSTPORT = 1025**

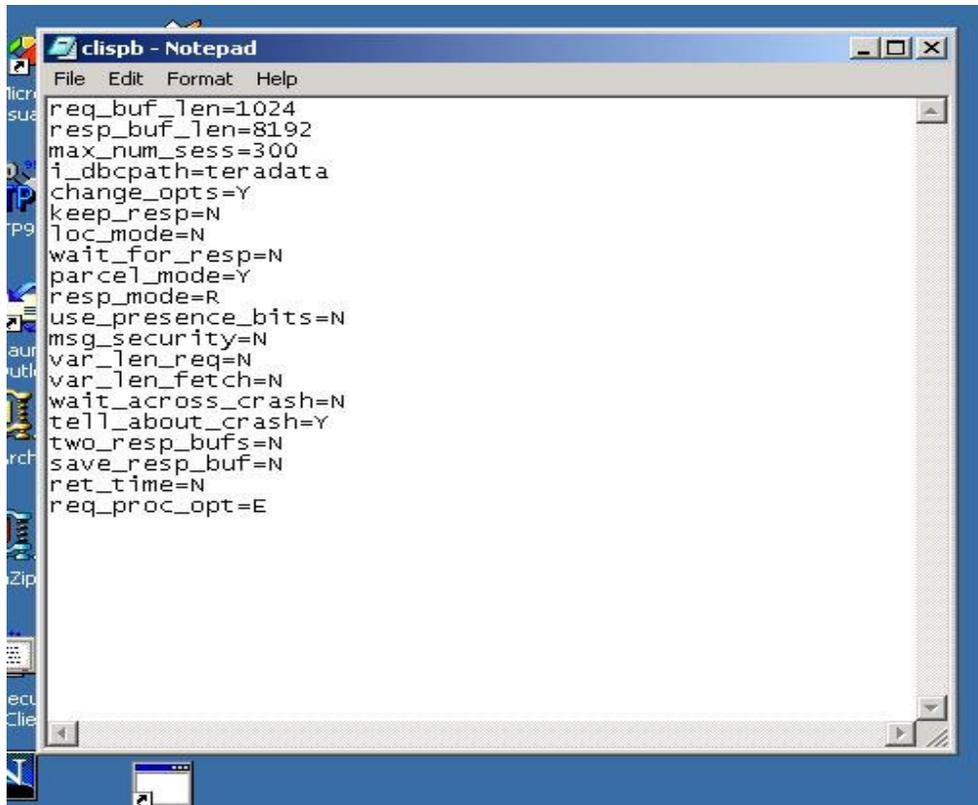
Because our server is listening the connect request on port 1025, you should set it as 1025. This is also the default port number used by those client components. If you don’t set it, perhaps you will not have any problem now. But we still suggest you add this system environment variable for the future usage. You shouldn’t insert a line

**tdmst 1025/TCP**

into the file C:\WINNT\System32\drivers\etc\services directly without correct the setting on environment variable TDMSTPORT. According to our test, that doesn’t work.

- Setting CLI System Parameter Block

You can find the file clispb.dat after you install the client software. In our computer, it is under the directory C:\Program Files\NCR\Teradata Client. Please use Notepad to open it. You will see



```
File Edit Format Help
req_buf_len=1024
resp_buf_len=8192
max_num_sess=300
i_dbcpath=teradata
change_opts=Y
keep_resp=N
loc_mode=N
wait_for_resp=N
parcel_mode=Y
resp_mode=R
use_presence_bits=N
msg_security=N
var_len_req=N
var_len_fetch=N
wait_across_crash=N
tell_about_crash=Y
two_resp_bufs=N
save_resp_buf=N
ret_time=N
req_proc_opt=E
```

Originally, `i_dbcpath` was set as `dbc`. That is not the same as what I set in the file hosts. So I modify it as `teradata`. When you use some components based on CLI and do not specify the TDPID or RDBMS, the components will open this file to find this default setting. So we suggest you to set it as what you set in your file hosts.

For other entries in this file, you can just keep them as original settings.

To use utilities such as Queryman and WinDDI, you still need to configure a Data Source Name (DSN) for yourself. We will discuss how to configure a DSN in the section “Developing ODBC applications”.

BTEQ is like a RDBMS console. This utility enables you to connect to Teradata RDBMS server as any valid database user, set the session environment and execute SQL statement as long as you have such privileges. BTEQWin is a Windows version of BTEQ. Both of them work on two components: CLI and TDP/MTDP. BTEQ commands or SQL statements entered into the BTEQ are packed into CLI packets, then TDP/MTDP transfers them to RDBMS. BTEQ supports 55 commands which fall into four groups:

- Session Control Commands
- File Control Commands
- Sequence Control Commands
- Format Control Commands

Here, we do not introduce each command. But, some usage tips and examples for the frequently used commands are given here.

### Usage Tips:

- BTEQ commands consist of a dot character followed by a command keyword, command options, and parameters.  
`.LOGON teradata/john`
- Teradata SQL statement doesn't begin with a dot character, but it must end with a ';' character.  
`SELECT * FROM students Where name = "Jack";`
- You can enter both BTEQ commands and SQL statements in any combination of uppercase and lowercase and mixed-case formats.  
`.Logoff`
- If you want to submit a transaction which includes several SQL statements, do as the following example:  
`Select * from students  
;insert into table students Values('00001', 'Jack', 'M', 25 )  
;select * from students;`

After you enter the last ';' and hit the [enter] key, these SQL requests will be submitted as a transaction. If anyone of them has an error, the whole transaction will be rolled back.

### Frequently Used Commands :

- LOGON  
`.logon teradata/john  
PASSWORD:johnpass`

In the above example, we connect to RDBMS called "teradata". "teradata" is the TDPID of the server. "john" is the userid and "johnpass" is the password of the user.

- LOGOFF  
    **.logoff**  
Just logoff from the current user account without exiting from BTEQ.

- EXIT or QUIT  
    **.exit**  
    **.quit**

These two commands are the same. After execute them, you will exit from BTEQ.

- SHOW VERSIONS  
    **.show versions**

check the version of the BTEQ you are using.

- SECURITY  
    **.set security passwords**  
    **.set security all**  
    **.set security none**

specify the security level of messages sent from network-attached systems to the Teradata RDBMS. By the first one, only messages containing user passwords, such as CREATE USER statement, will be encrypted. By the second one, all messages will be encrypted.

- SESSIONS  
    **.sessions 5**  
    **.repeat 3**  
    **select \* from students;**

After executing the above commands one by one, you will create 5 sessions running in parallel. Then we will execute select request three times. In this situation, three out of the five sessions will execute the select statement one time in parallel.

- QUIET  
    **.set quiet on**  
    **.set quiet off**

If switched off, the result of your command or SQL statement will not be displayed.

- SHOW CONTROLS  
    **.show controls**  
    **.show control**

Show the current settings for BTEQ software.

○ RETLIMIT

```
.set retlimit 4
select * from dbase;
```

Just display the first 4 rows of the result table and ignore the rest.

```
.set retlimit 0
select * from dbase;
```

Display all rows of the result table.

○ RECORDMODE

```
.set recordmode on
.set recordmode off
```

If switched on, all result rows will be displayed in binary mode.

○ SUPPRESS

```
.set suppress on 3
select * from students;
```

If the third column of the *students* table is Department Name, then the same department names will be display only once on the terminal screen.

○ SKIPLINE/SKIPDOUBLE

```
.set skipline on 1
.set skipdouble on 3
select * from students;
```

During displaying the result table, if the value in column 1 changes, skip one blank line to display the next row. If the value in column 3 changes, skip two blank lines to display the next row.

○ FORMAT

```
.set format on
.set heading "Result:"
.set rtitle "Report Title:"
.set footing "Result Finished"
```

Add the heading line, report title, and footing line to the result displayed on the terminal screen.

- OS

```
.os command  
c:\progra~1\ncr\terada~\bin> dir
```

The first command let you enter the Windows/Dos command prompt status. Then you can enter OS commands such as dir, del, copy, etc.

```
.os dir  
.os copy sample1.txt sample.old
```

Another way to execute the OS command is entering the command after the .os keyword.

- RUN

You can run a script file which contains several BTEQ commands and SQL requests. See the following example:

1) To edit a txt file, runfile.txt, using Notepad, the file contains:

```
.set defaults  
.set separator "$"  
select * from dbase;  
.set defaults
```

2) RUN

```
.run file = runfile.txt
```

If the working directory of BTEQ is not same as the directory containing the file, you must specify the full path.

- SYSIN & SYSOUT

SYSIN and SYSOUT are standard input and output streams of BTEQ. You can redirect them as the following example:

Start -> programs -> accessories -> command prompt

```
c:\>cd c:\program files\ncr\teradata client\bin  
c:\program files\ncr\teradata client\bin> bteq > result.txt  
.logon teradata/john  
johnpass  
select * from students;  
.exit  
c:\program files\ncr\teradata client\bin > bteq >
```

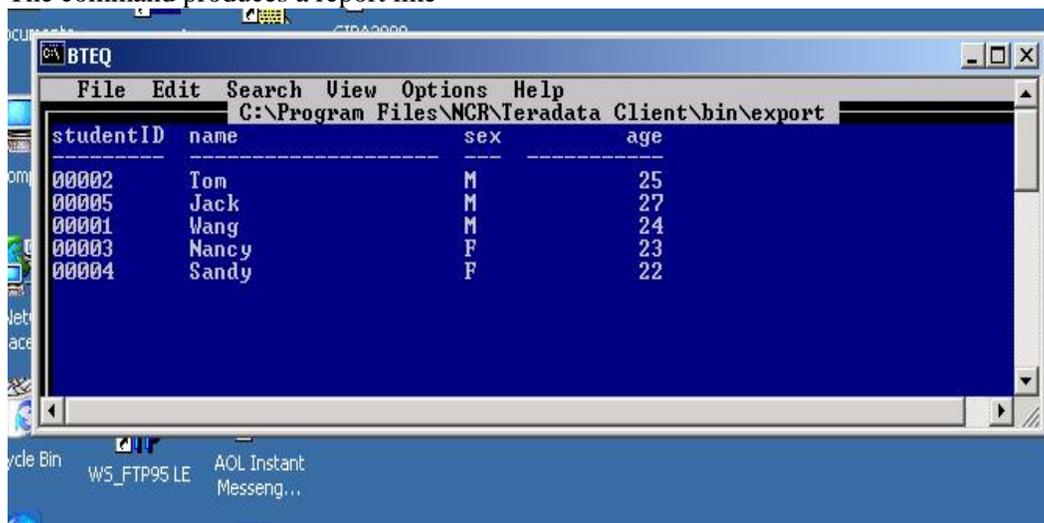
In the above example, all output will be written into result.txt file but not to the terminal screen. If runfile.txt file is placed in the root directory C:\, we can redirect the standard input stream of BTEQ as the following example:

```
c:\>cd c:\program files\ncr\teradata client\bin
c:\program files\ncr\teradata client\bin>bteq < c:\runfile.txt
```

- EXPORT

```
.export report file = export
select * from students;
.os edit export
```

The command produces a report like

A screenshot of a BTEQ window titled "BTEQ" with a menu bar (File, Edit, Search, View, Options, Help) and a title bar showing the path "C:\Program Files\NCR\Teradata Client\bin\export". The window displays a table with four columns: studentID, name, sex, and age. The data rows are: 00002 Tom M 25, 00005 Jack M 27, 00001 Wang M 24, 00003 Nancy F 23, and 00004 Sandy F 22. The window has a blue background and a scroll bar on the right.

studentID	name	sex	age
00002	Tom	M	25
00005	Jack	M	27
00001	Wang	M	24
00003	Nancy	F	23
00004	Sandy	F	22

Command keyword “report” specifies the format of output. “file” specifies the name of output file. If you want export the data to a file for backup, use the following command:

```
.export data file = exdata
select * from students;
```

you will get all data from the select statement and store them into the file, *exdata*, in a special format.

After exporting what you want, you should reset the export options.

```
. export reset
```

- IMPORT

As mentioned above, we already store all data of the *students* table into the file *exdata*. Now, we want to restore them into database. See the following example:

```
.delete from students;
.import data file = exdata
```

```
.repeat 5
using(studentId char(5),name char(20),sex char(1),age integer )
insert into students ( studentId, name, sex, age )
values( :studentId, :name, :sex, :age );
```

The third command requires BTEQ to execute the following command five times. The last command has three lines. It will insert one row into the *students* table each time.

- MACRO

You can use the SQL statements to create a macro and execute this macro at any time. See the following example:

```
create macro MyMacro1 as (
    ECHO \.set separator "#"
    ; select * from students;
);
```

This macro executes one BTEQ command and one SQL request.

```
execute MyMacro1;
```

This SQL statement executes the macro.

In this brief tutorial, we do not introduce many BTEQ commands used in writing BTEQ scripts and SQL syntax. For more information, you can download the *Teradata BTEQ reference guide* and *Teradata SQL reference* from the NCR web site (<http://www.info.ncr.com>). Next, we will give an example showing how to create database/user/table, manipulate data of a table, and build a client/server database application.

## Developing ODBC Applications

In the following demo, we will describe each step of using Teradata DBMS to develop a DB application. In the example, there are two users: John and Mike. John is the administrator of the application database. Mike works for John and he is the person who manipulates the table *students* in the database every day.

Step 1. John create user Mike and the database *student\_info* by using BTEQ.

- Running the BTEQ
- Logon the Teradata DBMS server.
- create a user and a database
- create tables
- use SQL commands: insert, select, delete, update
- Logoff and Quit BTEQ

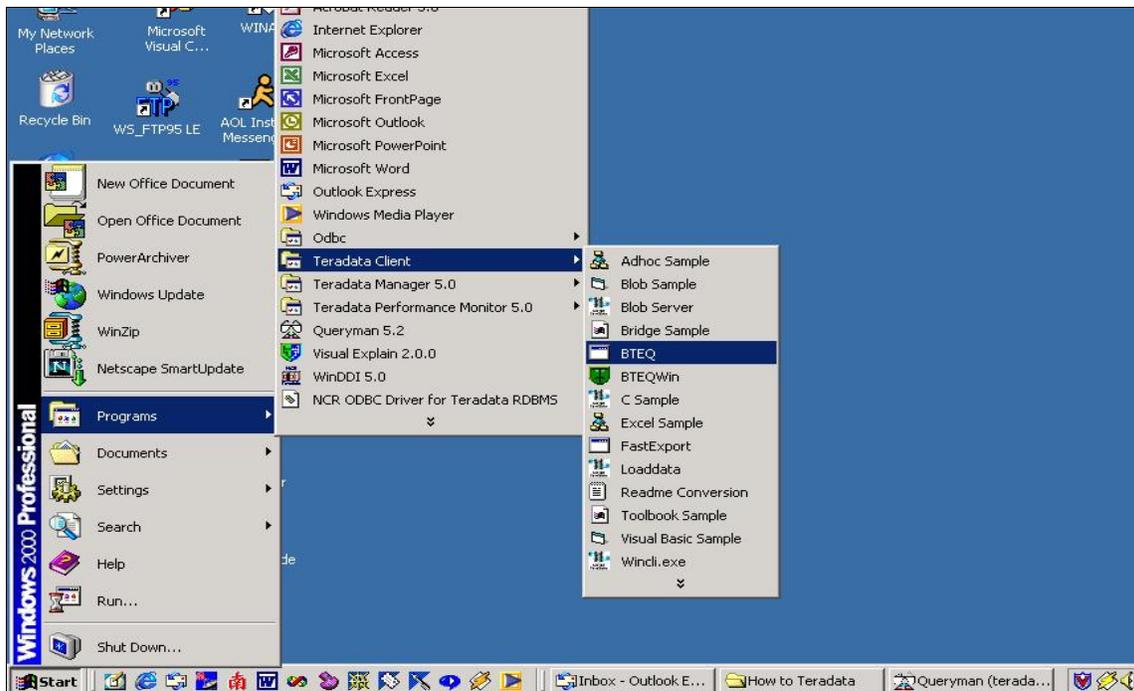
Step 2. Create one ODBC Data Source Name (DSN) to develop an application

Step 3. Develop an application by using ODBC interface

Step 4. Running the demo

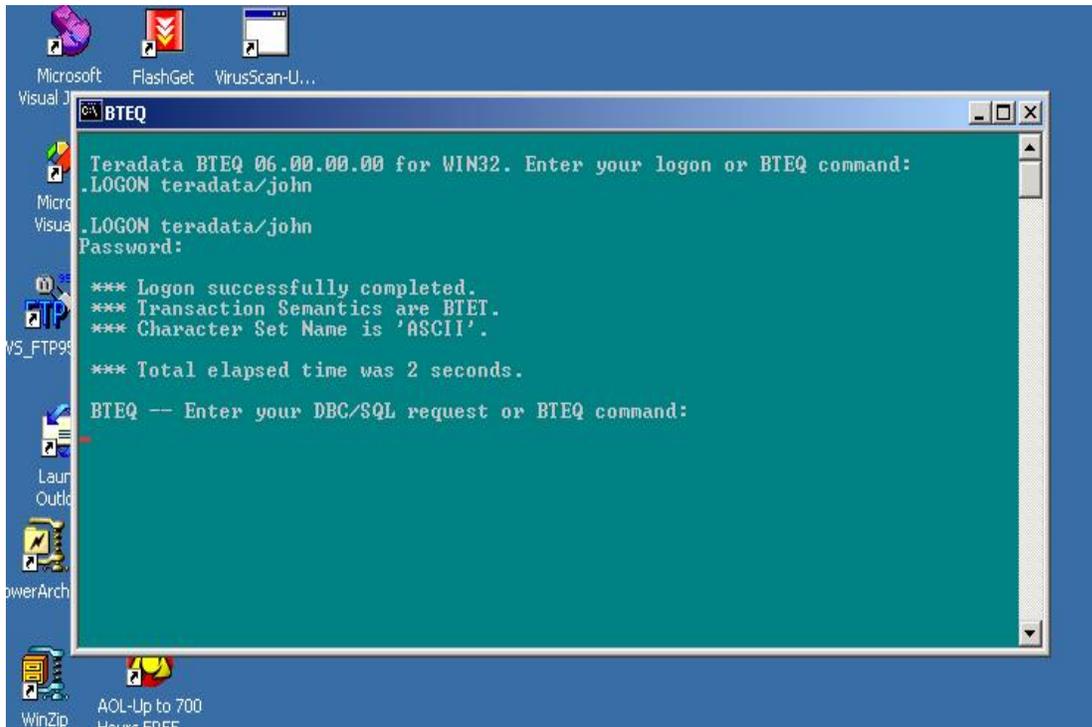
**Step 1. John create user Mike and the database *student\_info* by using BTEQ.**

- Running BTEQ  
start -> program -> Teradata Client -> BTEQ

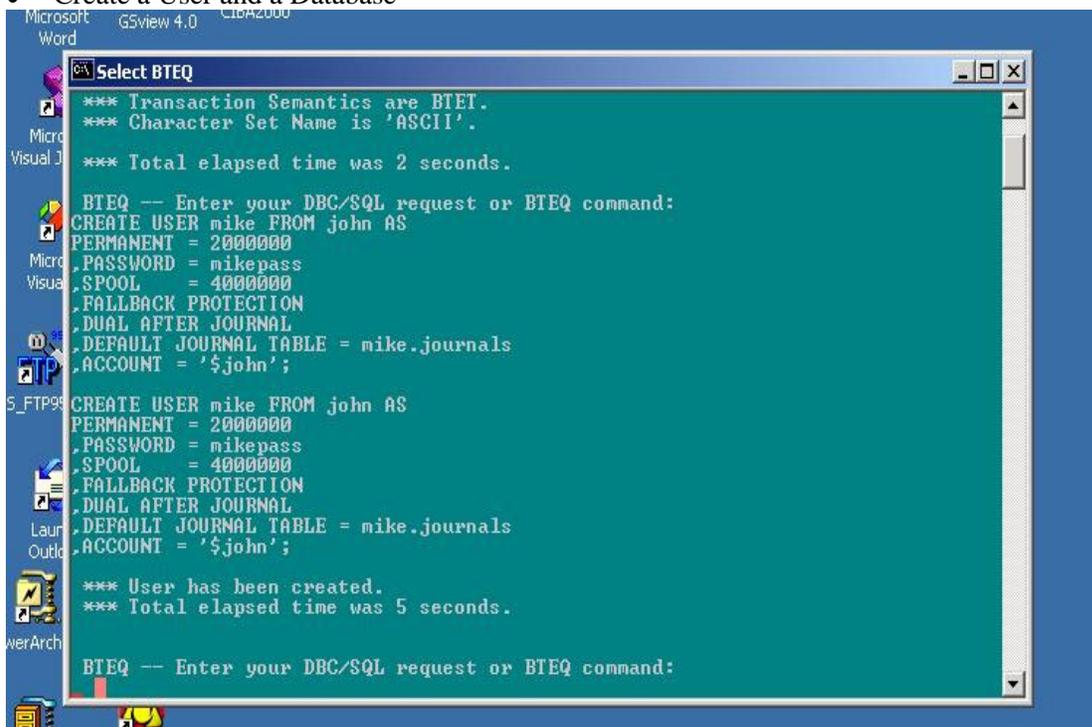


- Logon Teradata DBMS server

John was created by the Teradata DBMS administrator and was granted the privileges to create a USER and a DATABASE. In Teradata DBMS, the owner automatically has all privileges on the database he/she creates.

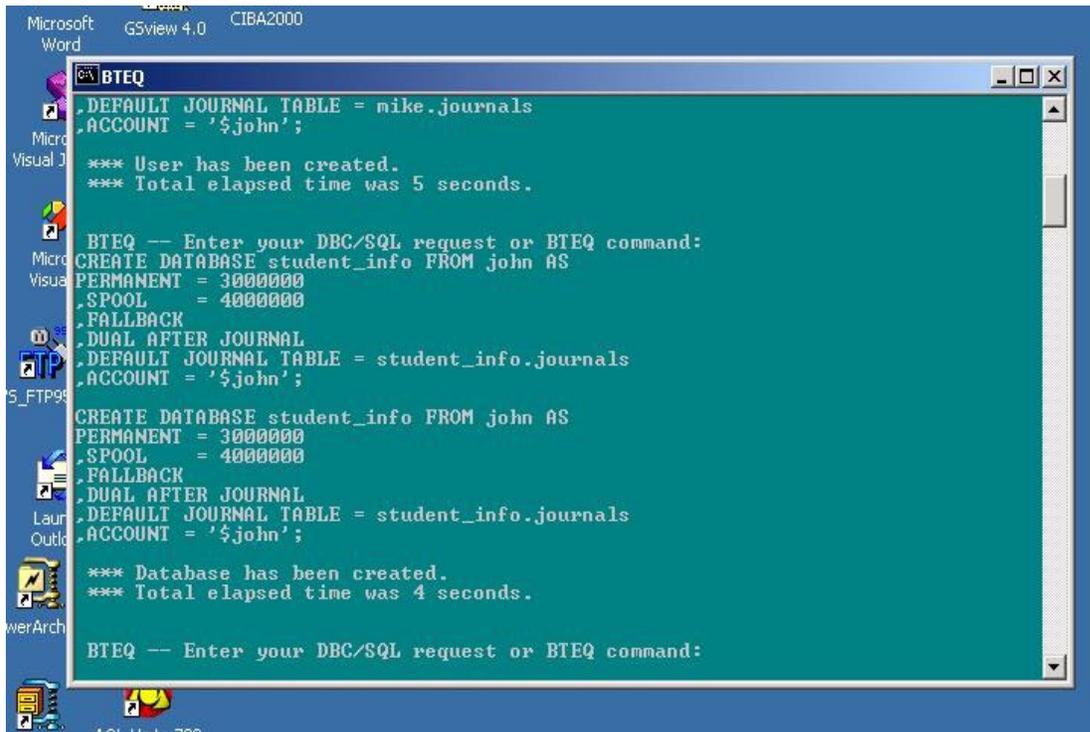


- Create a User and a Database



The above figure shows how to create a user. In Teradata DBMS, you should see a user as a special database. The difference between user and database is that a user has a password and can

logon to the DBMS, while a database is just a passive object in DBMS. Next, we see how to create a database:



```
Microsoft Word      GSview 4.0      CIBA2000
BTEQ
.DEFAULT JOURNAL TABLE = mike.journals
.ACCOUNT = '$john';

*** User has been created.
*** Total elapsed time was 5 seconds.

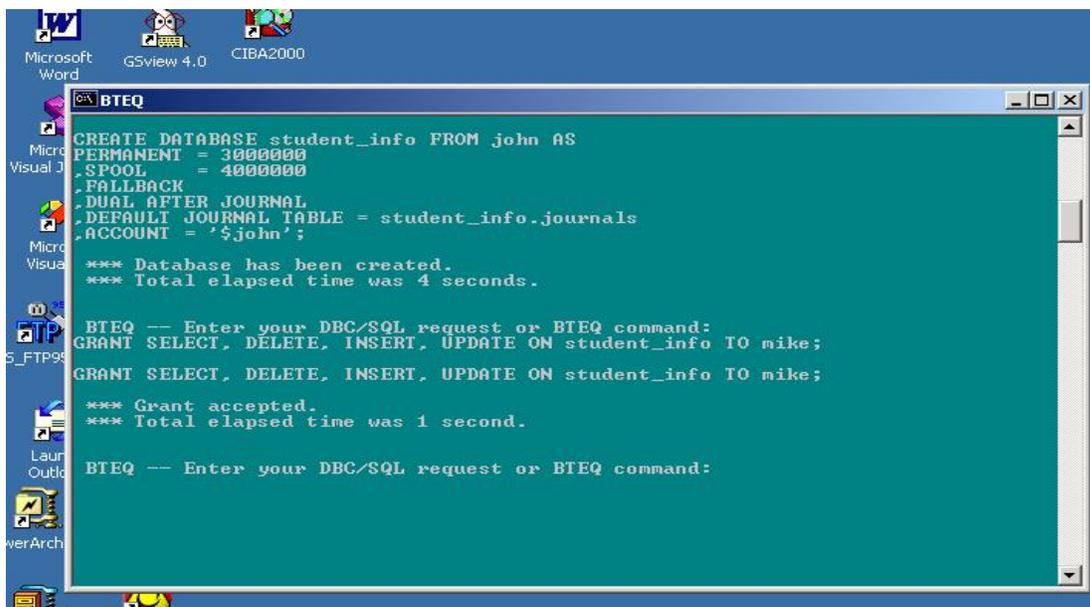
BTEQ -- Enter your DBC/SQL request or BTEQ command:
CREATE DATABASE student_info FROM john AS
PERMANENT = 3000000
.SPOOL = 4000000
.FALLBACK
.DUAL AFTER JOURNAL
.DEFAULT JOURNAL TABLE = student_info.journals
.ACCOUNT = '$john';

CREATE DATABASE student_info FROM john AS
PERMANENT = 3000000
.SPOOL = 4000000
.FALLBACK
.DUAL AFTER JOURNAL
.DEFAULT JOURNAL TABLE = student_info.journals
.ACCOUNT = '$john';

*** Database has been created.
*** Total elapsed time was 4 seconds.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```

John is the owner of user Mike and database student\_info. Of course, John has all privileges on this database such as creating table, executing select, insert, update and delete statements. But we notice that Mike doesn't have any privilege on this database now. So John needs to grant some privileges to Mike for his daily work. See the next figure:



```
Microsoft Word      GSview 4.0      CIBA2000
BTEQ
CREATE DATABASE student_info FROM john AS
PERMANENT = 3000000
.SPOOL = 4000000
.FALLBACK
.DUAL AFTER JOURNAL
.DEFAULT JOURNAL TABLE = student_info.journals
.ACCOUNT = '$john';

*** Database has been created.
*** Total elapsed time was 4 seconds.

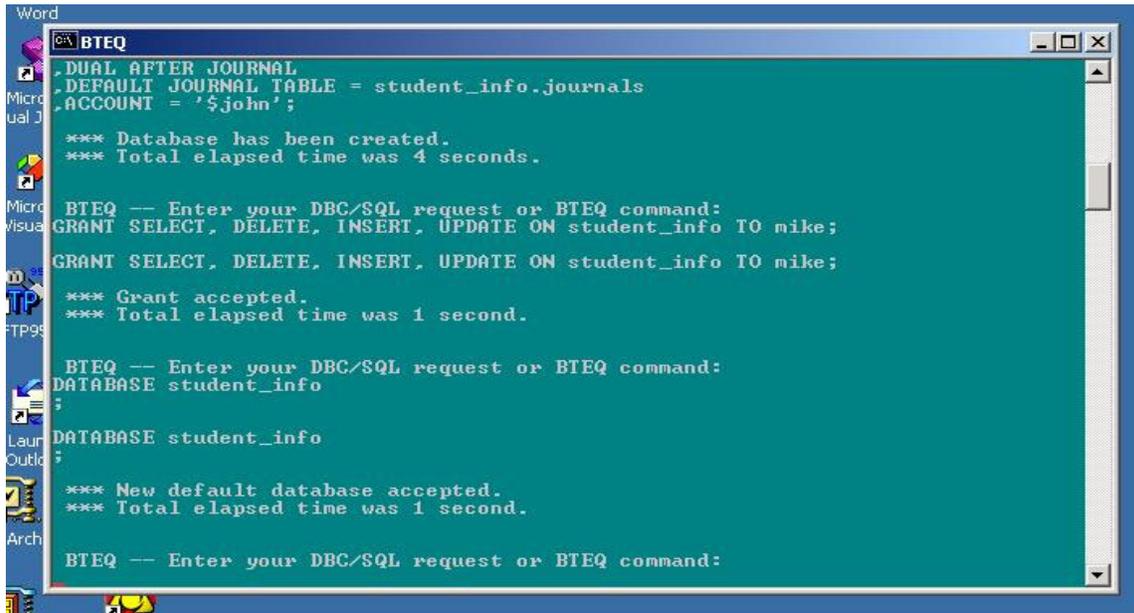
BTEQ -- Enter your DBC/SQL request or BTEQ command:
GRANT SELECT, DELETE, INSERT, UPDATE ON student_info TO mike;
GRANT SELECT, DELETE, INSERT, UPDATE ON student_info TO mike;

*** Grant accepted.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```

- Create Table

After granting appropriate privileges to Mike, John needs to create a table for storing the information of all students. First, he must specify the database containing the table.



```
Word
c:\ BTEQ
.DUAL AFTER JOURNAL
.DEFAULT JOURNAL TABLE = student_info.journals
.ACCOUNT = '$john';

*** Database has been created.
*** Total elapsed time was 4 seconds.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
GRANT SELECT, DELETE, INSERT, UPDATE ON student_info TO mike;
GRANT SELECT, DELETE, INSERT, UPDATE ON student_info TO mike;

*** Grant accepted.
*** Total elapsed time was 1 second.

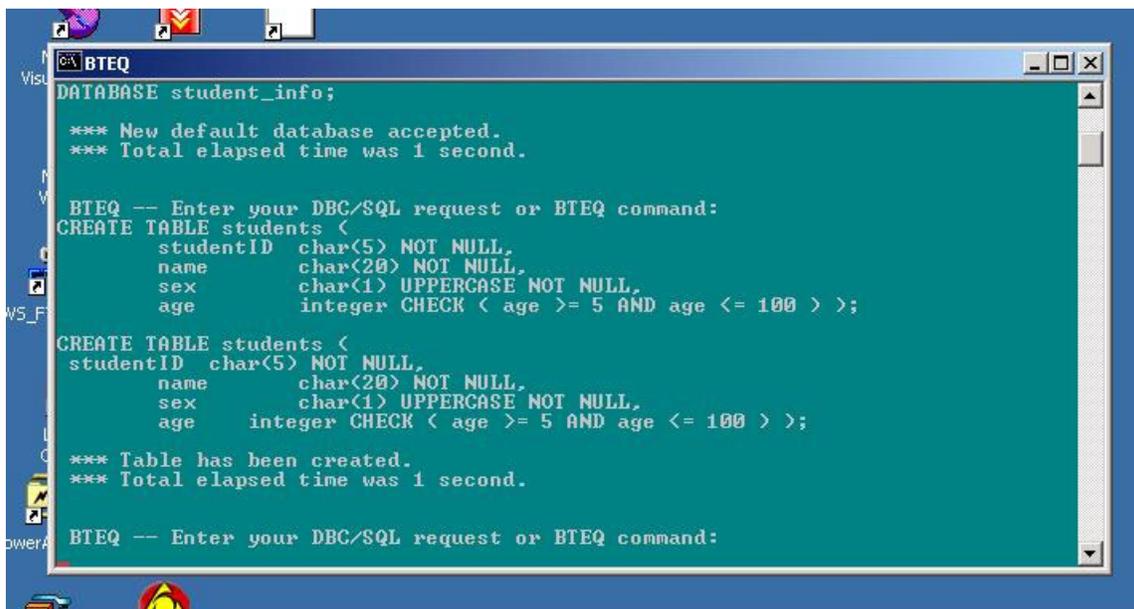
BTEQ -- Enter your DBC/SQL request or BTEQ command:
DATABASE student_info
;

DATABASE student_info
;

*** New default database accepted.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```

Then he creates the table *students*



```
Visu
c:\ BTEQ
DATABASE student_info;

*** New default database accepted.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
CREATE TABLE students (
  studentID char(5) NOT NULL,
  name char(20) NOT NULL,
  sex char(1) UPPERCASE NOT NULL,
  age integer CHECK ( age >= 5 AND age <= 100 ) );

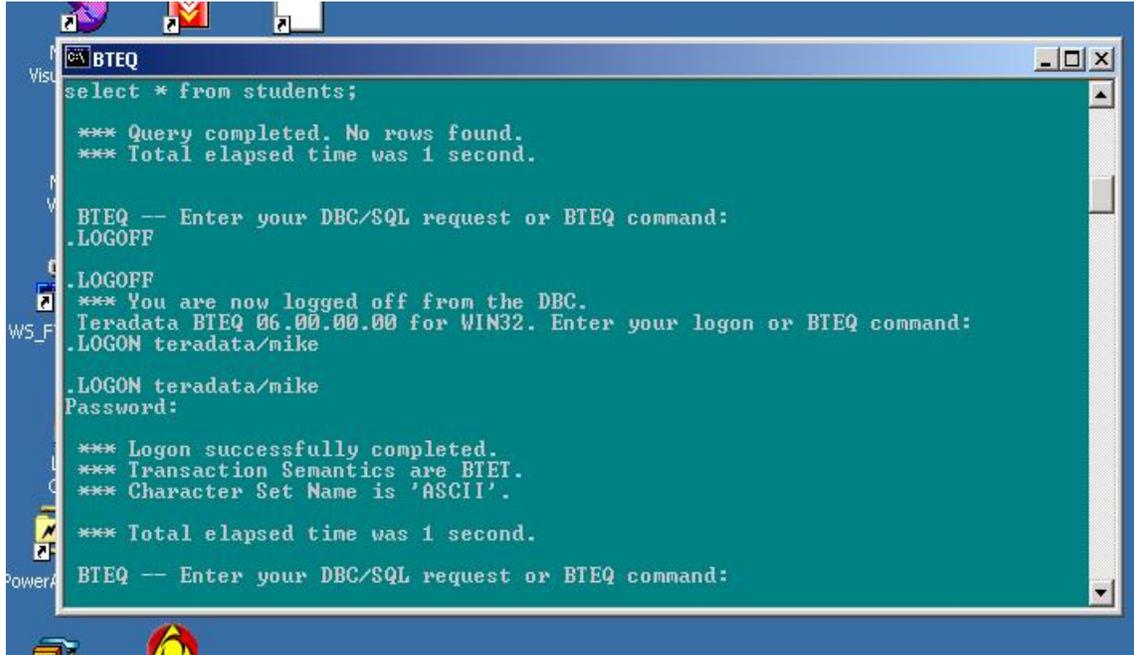
CREATE TABLE students (
  studentID char(5) NOT NULL,
  name char(20) NOT NULL,
  sex char(1) UPPERCASE NOT NULL,
  age integer CHECK ( age >= 5 AND age <= 100 ) );

*** Table has been created.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```

- Using SQL statements such as Select, Insert, Delete and Update

Now, Mike can logon and insert some data into the table *students*.



```
BTEQ
select * from students;

*** Query completed. No rows found.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
.LOGOFF

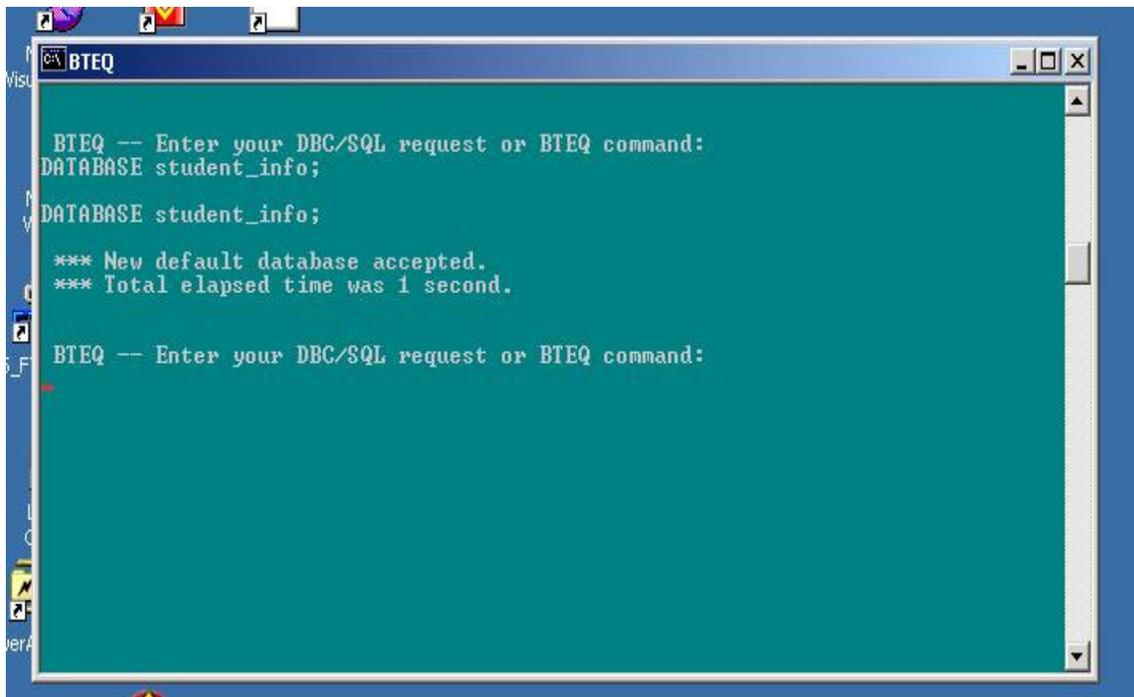
.LOGOFF
*** You are now logged off from the DBC.
Teradata BTEQ 06.00.00.00 for WIN32. Enter your logon or BTEQ command:
.LOGON teradata/nike

.LOGON teradata/nike
Password:

*** Logon successfully completed.
*** Transaction Semantics are BTET.
*** Character Set Name is 'ASCII'.

*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```



```
BTEQ

BTEQ -- Enter your DBC/SQL request or BTEQ command:
DATABASE student_info;

DATABASE student_info;

*** New default database accepted.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
```

```

*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
INSERT INTO students VALUES ('00001', 'Wang', 'M', 24 );
INSERT INTO students VALUES ('00001', 'Wang', 'M', 24 );

*** Insert completed. One row added.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
INSERT INTO students VALUES ('00002', 'Tom', 'M', 25 );
INSERT INTO students VALUES ('00002', 'Tom', 'M', 25 );

*** Insert completed. One row added.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
INSERT INTO students VALUES ('00003', 'Nancy', 'F', 23 );
INSERT INTO students VALUES ('00003', 'Nancy', 'F', 23 );

*** Insert completed. One row added.
*** Total elapsed time was 1 second.

```

In the figure below, Mike inserts a new row whose first field is “00003”. We notice that there are two rows whose first fields have the same value. So, Mike decides to delete one of them.

```

BTEQ -- Enter your DBC/SQL request or BTEQ command:
INSERT INTO students VALUES ('00003', 'Sandy', 'F', 24 );
INSERT INTO students VALUES ('00003', 'Sandy', 'F', 24 );

*** Insert completed. One row added.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
SELECT * FROM students;
SELECT * FROM students;

*** Query completed. 4 rows found. 4 columns returned.
*** Total elapsed time was 1 second.

studentID  name          sex          age
-----
000002    Tom          M            25
000003    Nancy       F            23
000001    Wang        M            24
000003    Sandy       F            24

BTEQ -- Enter your DBC/SQL request or BTEQ command:
DELETE FROM students WHERE name = 'Sandy';
DELETE FROM students WHERE name = 'Sandy';

*** Delete completed. One row removed.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
SELECT * FROM students;
SELECT * FROM students;

*** Query completed. 3 rows found. 4 columns returned.
*** Total elapsed time was 1 second.

studentID  name          sex          age
-----
000002    Tom          M            25
000003    Nancy       F            23
000001    Wang        M            24

BTEQ -- Enter your DBC/SQL request or BTEQ command:

```

```

BTEQ
BTEQ -- Enter your DBC/SQL request or BTEQ command:
INSERT INTO students VALUES ('00004', 'Sandy', 'F', 24 );
INSERT INTO students VALUES ('00004', 'Sandy', 'F', 24 );
*** Insert completed. One row added.
*** Total elapsed time was 1 second.

BTEQ -- Enter your DBC/SQL request or BTEQ command:
SELECT * FROM students;
SELECT * FROM students;
*** Query completed. 4 rows found. 4 columns returned.
*** Total elapsed time was 1 second.

studentID  name          sex          age
-----
00002     Tom           M            25
00003     Nancy        F            23
00001     Wang         M            24
00004     Sandy        F            24

BTEQ -- Enter your DBC/SQL request or BTEQ command:

```

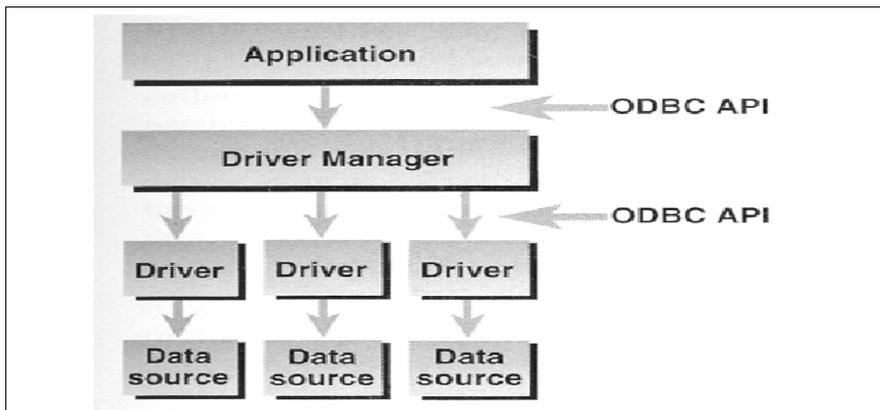
- Logoff and Quit BTEQ

Mike uses the following command

```
.exit
```

to logoff and quit BTEQ.

**Step 2. Create one ODBC Data Source Name (DSN) to develop an application**

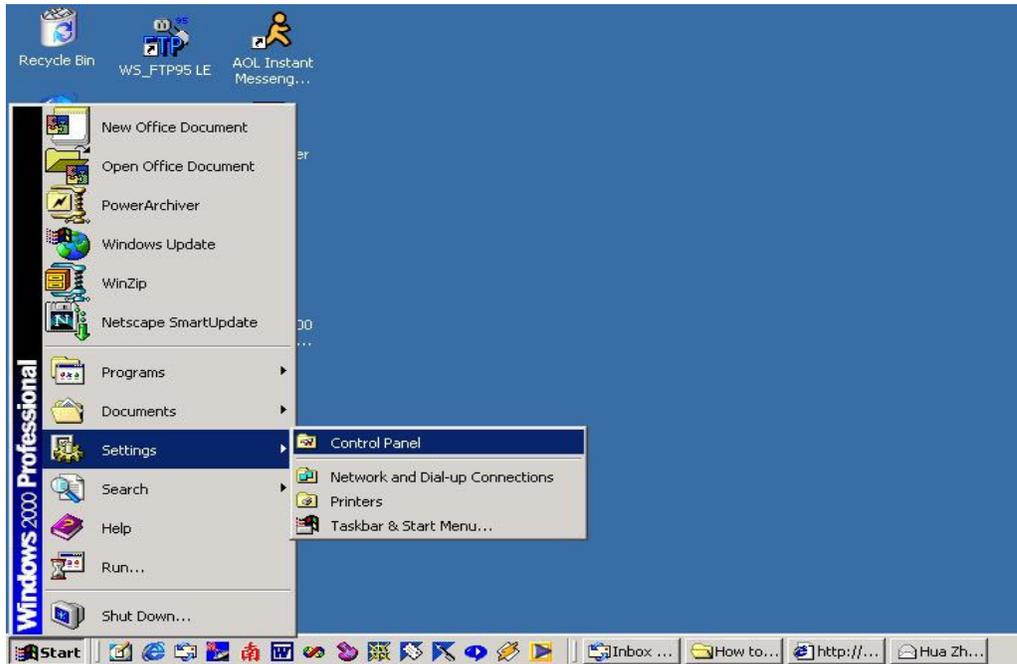


Architecture of DB Application Based on ODBC

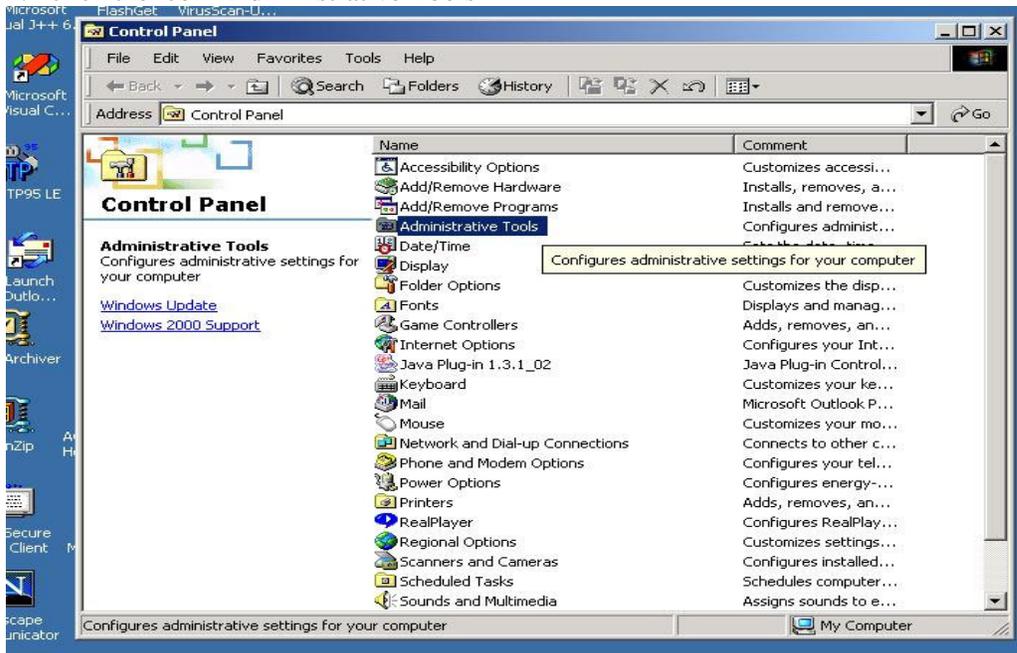
In the figure above, the Data source modules refer to physical databases with different DBMSs. The term Data Source Name (DSN) is just like a profile which is a link between the first level, Application, and the second level, ODBC Driver Manager. We use the profile, DSN, to describe which ODBC driver will be used in the application and the application will logon which account on which database.

The following figures show each step of creating the DSN used in our application:

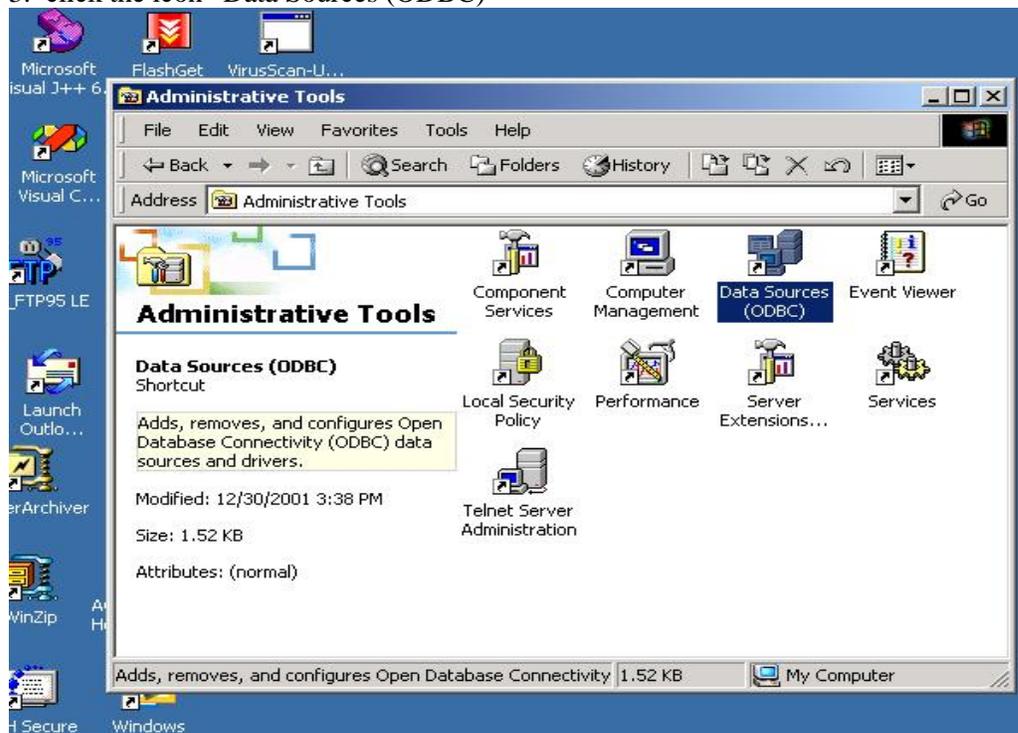
1. start -> settings -> control panel



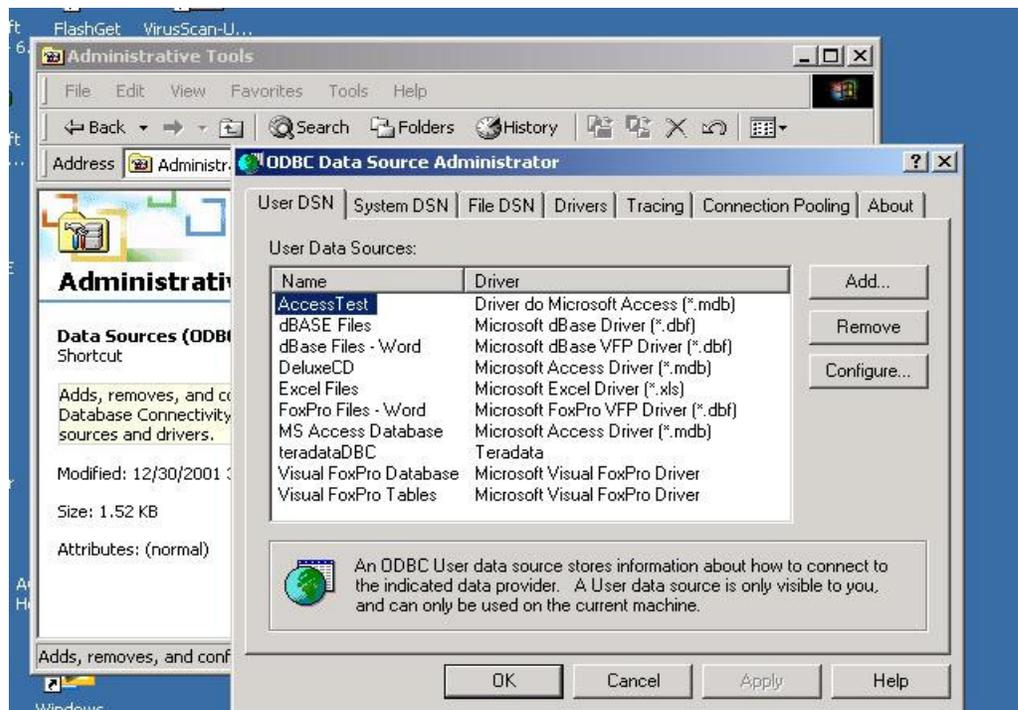
2. click the icon "Administrative Tools"



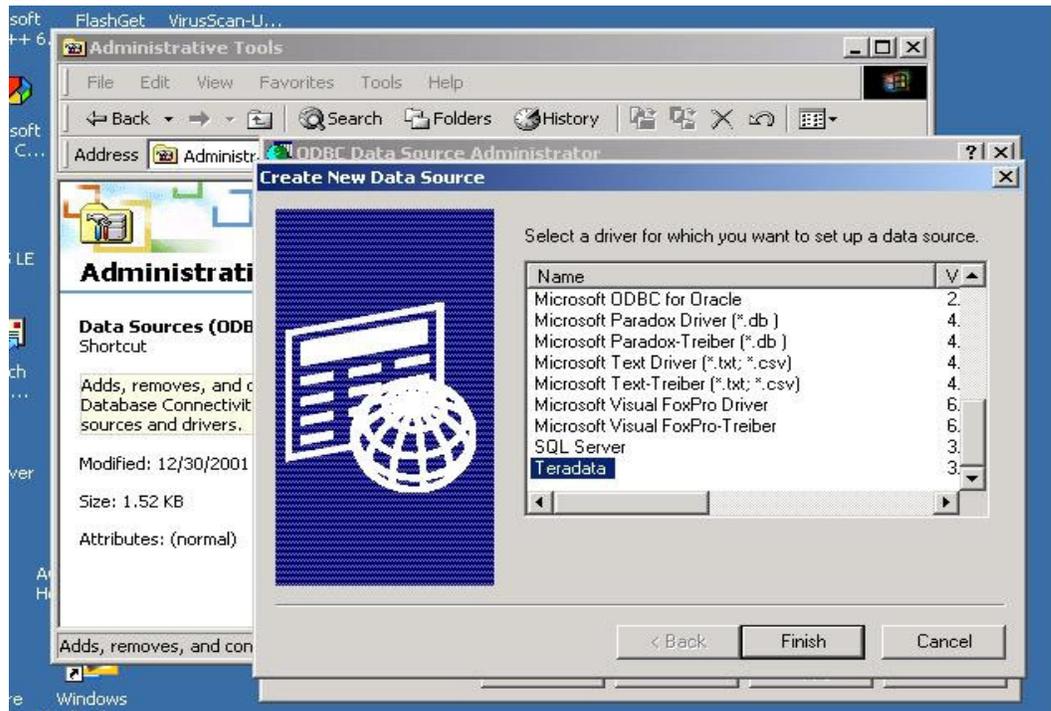
3. click the icon “Data Sources (ODBC)”



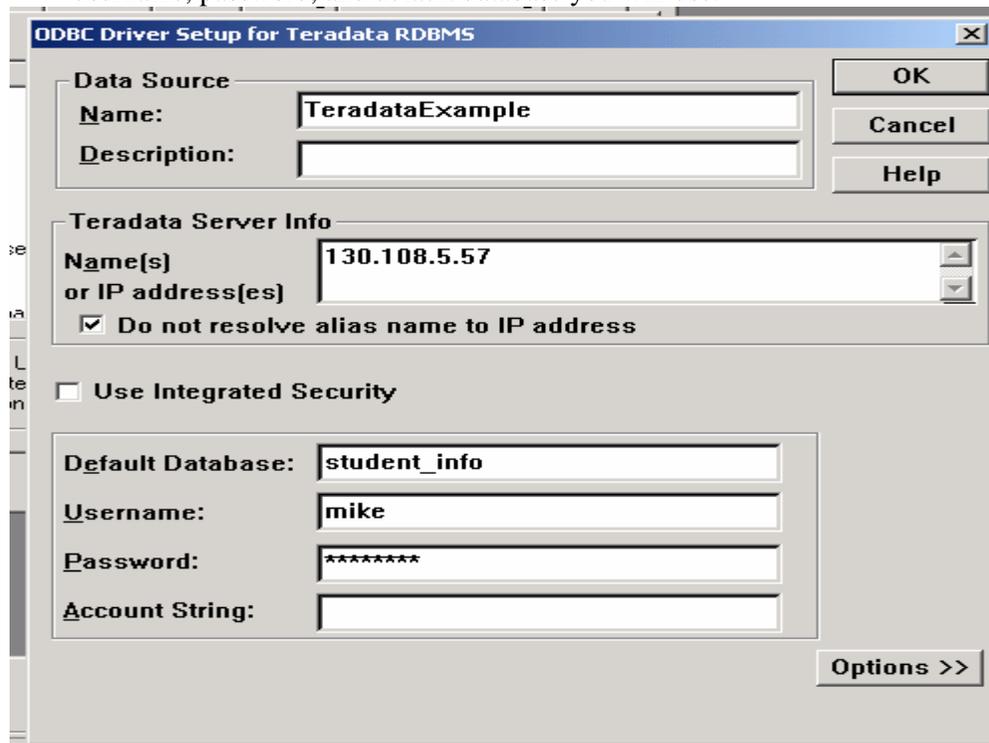
4. The ODBC Data Source Administrator window lists all DSN already created on your computer. Now, click the button “Add...”



5. You will be asked to choose one ODBC driver for your Data Source. You should choose Teradata.



6. Then, you need to type in all information about your DSN, such as IP address of server, username, password, and default database you will use.



### Step 3. Develop an application by using ODBC interface

The demo was developed on Windows NT/2000 by using VC++ 6.0. We can access the table *students* via ODBC interface. You need to include the following files:

```
#include <sql.h>
#include <sqlext.h>
#include <odbcinst.h>
#include <odbcss.h>
#include <odbcver.h>
```

You also need to link `odbc32.lib`. In VC++ 6.0 developing studio, you can set the link option to finish the appropriate compiling and link. Another way to avoid the subtle job of setting link and compiling options is directly adding this lib file into your project.

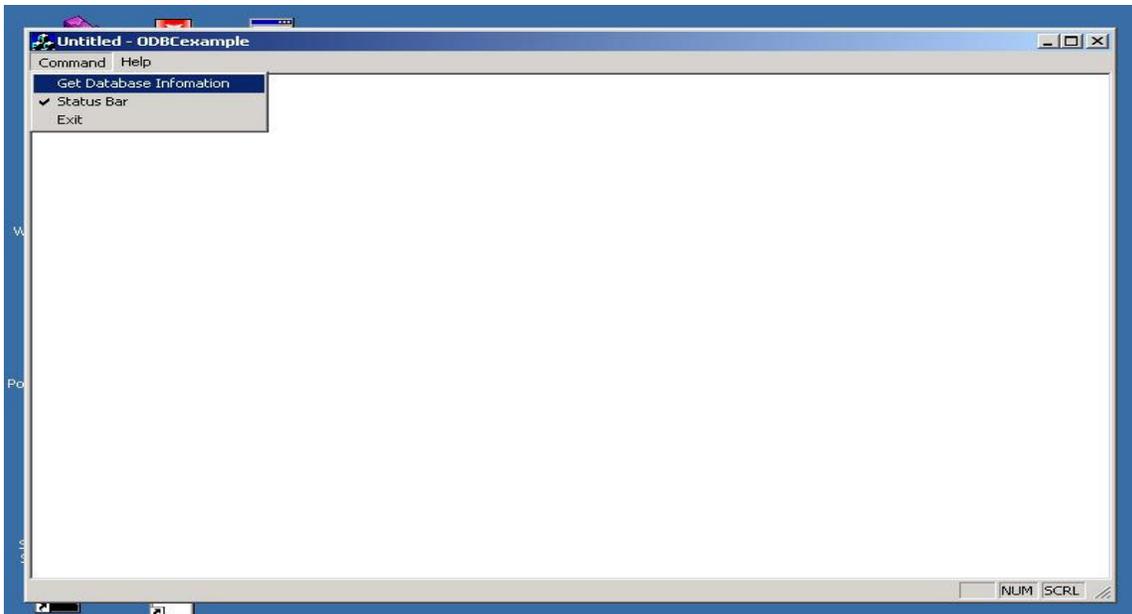
The ODBC program scheme is shown below, which is a code segment of executing select SQL statement:

```
SQLAllocEnv (&DSNhenv);
SQLAllocConnect (ODBCHenv, &ODBCHdbc);
SQLConnect (ODBCHdbc, DataSourceName, DBUsername, DBUserpassword
);
SQLAllocStmt (ODBCHdbc, &ODBCHstmt);
Construct the SQL command string
SQLExecDirect (ODBCHstmt, (UCHAR *)command, SQL_NTS);
if (ODBC_SUCCESS)
{
    SQLFetch (ODBCHstmt);
    while (ODBC_SUCCESS)&&(data set is not empty)
    {
        processing the data
        SQLFetch (ODBCHstmt);
    }
}
SQLFreeStmt (ODBCHstmt, SQL_DROP);
SQLDisconnect (ODBCHdbc);
SQLFreeConnect (ODBCHdbc);
SQLFreeEnv (DSNhenv);
```

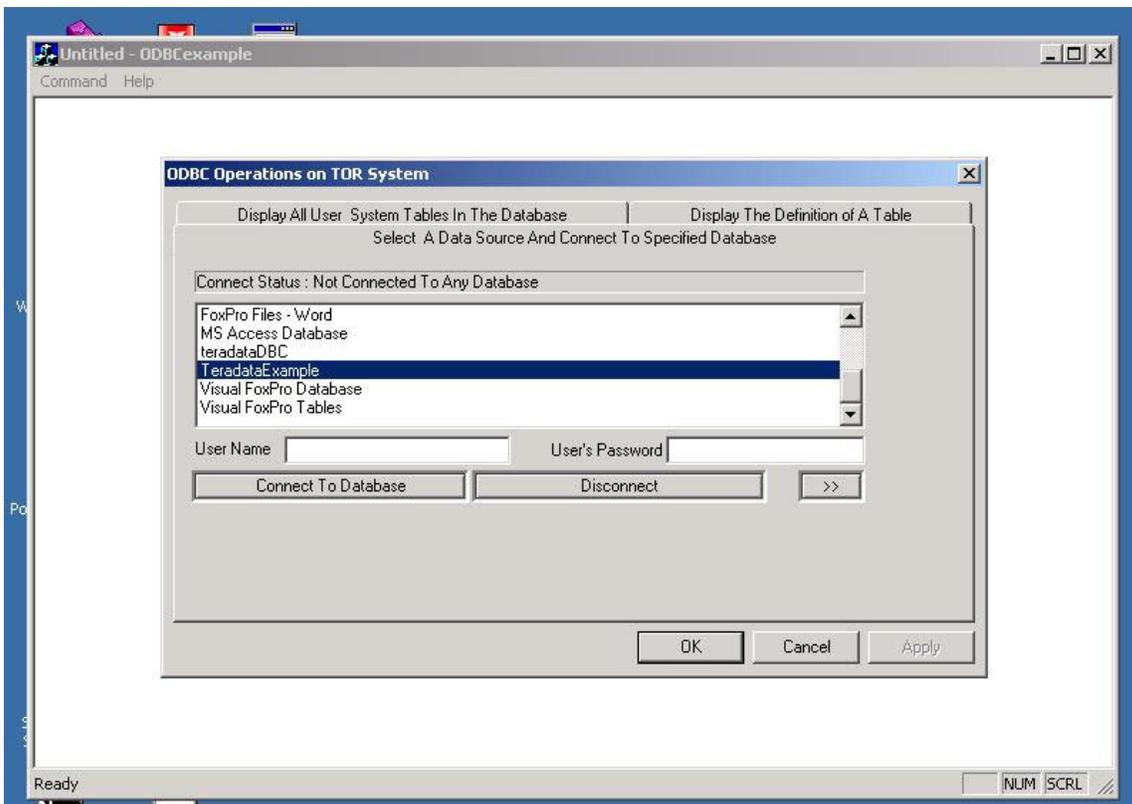
When ODBC function `SQLConnect()` is called in your program, you need to specify the Data Source Name created in step 2. Our demo is just a simple example, though it has invoked almost all functions often used in general applications. For more details about ODBC programming, please see related ODBC references and VC++ references.

### Step 4. Running the demo

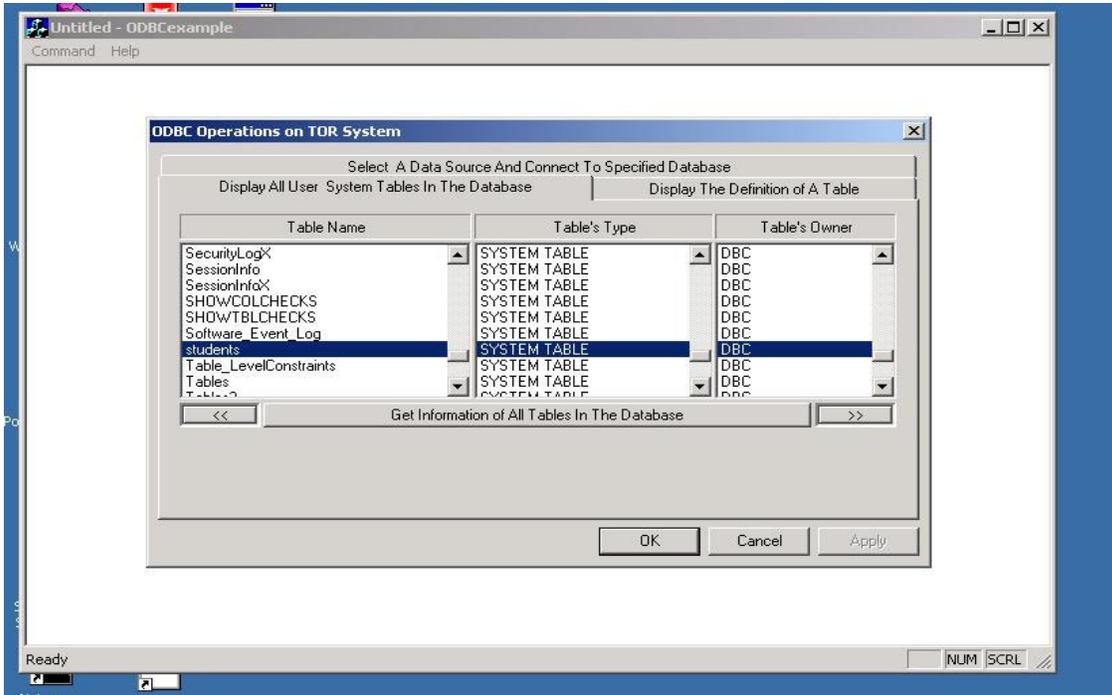
Copy all files of the project to your PC and double click the file `ODBCexample.dsw`. VC++ 6.0 developing studio will load the Win32 project automatically. Then you can choose menu item “build” or “execute ODBCexample”.



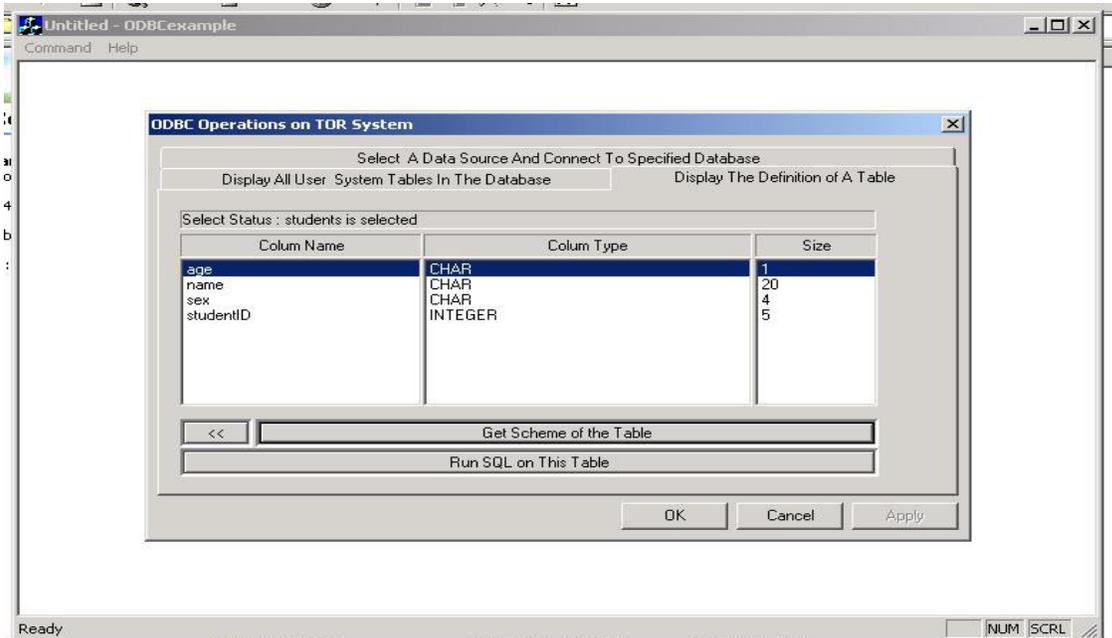
In the next window, you can see all DSN defined on your PC. We can choose TeradataExample created in step 2. We don't need to provide the user name *mike* and the password *mikepass*, because they were already set in DSN. Then we click the button "Connect To Database" to connect to the Teradata DBMS server.

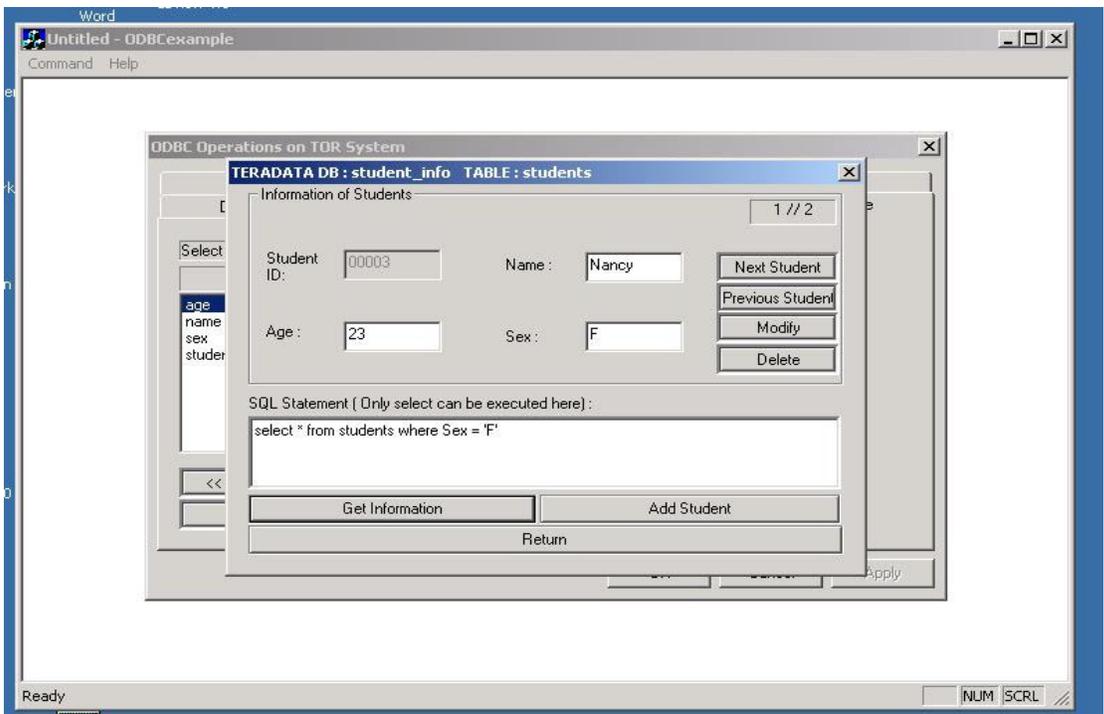
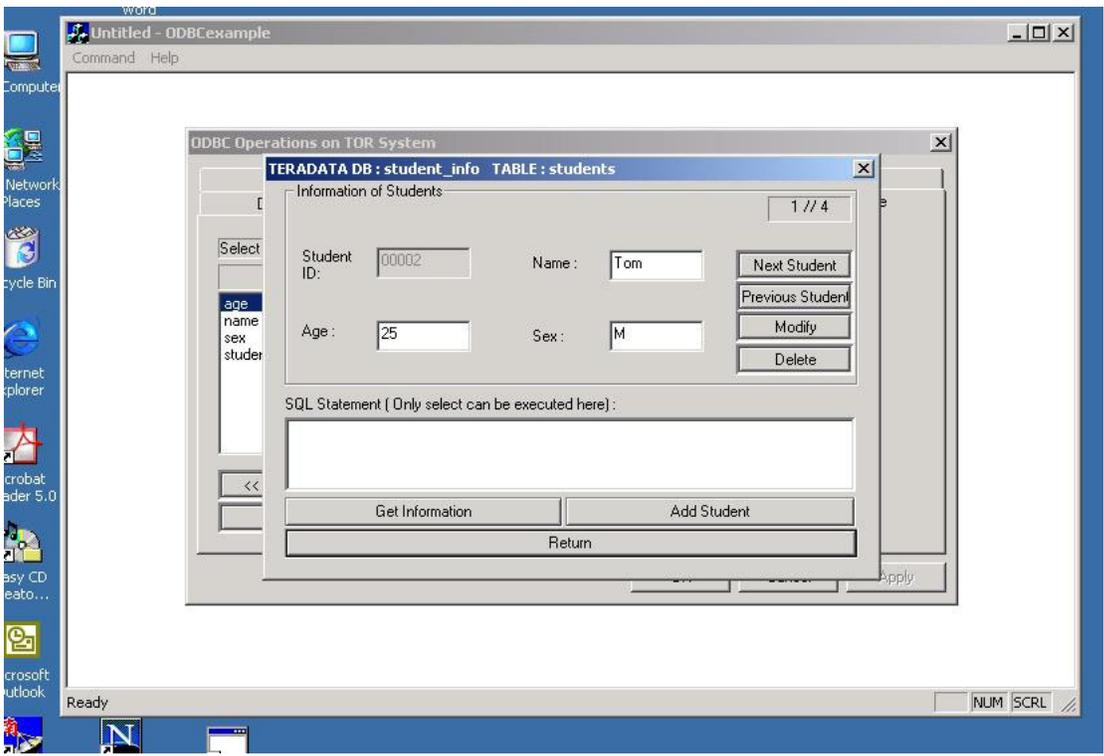


Click “>>” button to enter next window sheet. Now, after pressing “Get Information of All Tables In The Database”, you can see all tables in the database including *students* created in step 1. Then we can choose one table from the leftmost listbox and enter the next window sheet by clicking “>>”.



Click the button “Get Scheme of the Table” to see the definition of the table. And if the table you have chosen is *students*, you can press “Run SQL on This Table” to execute a SQL statement on the table.





After typing the SQL statement in the edit box, you can press button “Get Information” to execute it. If you want to add a student, please click “Add Student”.

