

Daffodil DB

Tools Guide

Version 4.1

March 2005

Copyright © Daffodil Software Limited
SCO 42, 3rd Floor, Old Judicial Complex,
Civil Lines
Gurgaon – 125001
Haryana
India.
www.daffodildb.com

All rights reserved. Daffodil DB™ is a registered trademark of Daffodil Software Limited. Java™ is a registered trademark of Sun Microsystems, Inc. All other brand and product names are trademarks of their respective companies.

Table of Contents

Preface

Purpose of this Document	05
Audience	05
Related Documentation	06

Getting Started with Daffodil DB Browser

Introduction	07
Starting Daffodil DB Browser	07
Daffodil DB Browser Operation Modes	08

Tool Bars of Daffodil DB Browser

Query Analyzer Mode Tool Bar	09
Browser Mode Tool Bar	11
Browser Tool Bar Button	11
Table Tool Bar Button	12
View Tool Bar Button	13

Menus of Daffodil DB Browser

File Menu	15
Tool Menu	16
Windows Menu	17
Query Analyzer Pop Up Menu	18
Pop Up Menus	18

Different Types of Nodes displayed in Browser Mode

Daffodil Servers Node	19
Embedded Node	20
Databases Node	23
Tables Node	26
Columns Node	30
Indexes Node	32
Triggers Node	35
Constraints Node	39
FullTextIndexes Node*	41
Views Node	44
Procedures Node	47
Roles Node	50

Daffodil DB Browser Operation Modes

Query Analyzer Mode	53
Message Tab	53
Grid Tab	54
Procedure Tab	55
Executing Queries	56
Saving Queries	59
Loading Queries Saved in Files	59

Work using Multiple Query Execution Pads	60
Running Multiple Sessions against a Database	61
Running Parameterized Queries	61
Browser Mode	69
Creating a Database	69
Creating a Table	70
Creating a View	72
Creating a Procedure	74
Creating a Trigger	75
Import / Export	77
Online Back Up*	81
Offline Back Up*	84
Restore *	86
Drop Schedule*	88
Daffodil DB Shell	
Starting Daffodil DB Shell in Embedded Mode	90
Starting Daffodil DB Shell in Network Mode	90
Starting Daffodil DB Shell using Property File	91
Executing Queries	92
Executing Prepared Statements	93

* Features that are not supported in One\$DB
--

Preface

Purpose of This Document

Daffodil DB Tools Guide explains how to use *Daffodil DB Browser* with *Embedded* as well as *Server* versions of Daffodil DB. It also explains how to perform various database operations on Daffodil DB using Daffodil DB Browser such as creating a database, creating database objects, manipulating data, creating triggers etc.

Audience

This guide is intended to act as ready reference tool for software developers building Daffodil DB applications. This guide assumes that you are familiar with the following concepts:

- Basis JDBC (Java Database Connectivity).
- Basic SQL (Standard Query Language).
- Basic Database Concepts.
- Basic Java Programming Language.

It is also assumed that the reader has gone through '*Getting Started with Daffodil DB Guide*'.

Related Documentation

Daffodil DB Getting Started Guide	Designed to help new and intermediate Daffodil DB users navigate and perform common tasks like How to start and stop Daffodil DB, Understanding key variables used by Daffodil DB, User documentation bundled with Daffodil DB. Also briefly describes Daffodil DB Editions and Tools
Daffodil DB System Guide	Describes the architecture of Daffodil DB and provides the information that the server administrator might need to keep Daffodil DB running with high performance and reliability in a server framework or a multi-user application server. Also describes the standards on which Daffodil DB had been built, transaction capabilities and some of the unique features supported by Daffodil DB.
Daffodil DB SQL Reference Guide	Covers all the SQL-99 features supported by Daffodil DB. This ready reference tool describes in detail the syntax and semantics of SQL language statements and elements for Daffodil DB. It explains how to use SQL with Daffodil DB and how to perform various database operations on Daffodil DB such as creating tables or indexes, managing transactions and sessions, Daffodil DB security features etc.
Daffodil DB JDBC Reference Guide	Explains how to use Daffodil DB and JDBC technology to develop applications. It describes the basic Daffodil DB and JDBC concepts like JDBC 3.0 features supported by Daffodil DB, how to create and access Daffodil DB databases through JDBC API, Daffodil DB support for JDBC and JTA and how to use Daffodil DB in a Distributed Transaction Processing environment.

Daffodil DB Browser

Introduction

Daffodil DB Browser is a GUI based database management tool. Daffodil DB Browser can be used to carry out all database operations on Daffodil DB such as creating database objects, querying database objects, manipulating data in database objects and browsing database objects along with its associated properties etc.

You can connect to multiple Daffodil DB Embedded instances and Daffodil DB Server instances running on multiple hosts using the same Daffodil DB Browser instance.

Starting Daffodil DB Browser

Daffodil DB Browser can be started by executing `.bat` or `.sh` file placed in `<DAFFODIL_HOME>/Tools`

Working with Daffodil DB Embedded using Daffodil DB Browser

For starting Daffodil DB Embedded, we can execute `BrowserEmbedded.bat` or `BrowserEmbedded.sh` placed in `<DAFFODIL_HOME>/Tools`. A window appears prompting the user to enter a user name and password. Default User is “DAFFODIL” and default password is “DAFFODIL” as well. After checking the user name and password, Daffodil DB Embedded can be started on the localhost.

Daffodil DB Browser mainframe opens with a tree in the left panel. You can select any database from ‘Database node’.

Working with Daffodil DB Server using Daffodil DB Browser

When working with Daffodil DB server, the server needs to be running on the local or remote machine before you connect to Daffodil DB server using Daffodil DB Browser.

To start Daffodil DB server, execute `BrowserServer.bat` or `BrowserServer.sh` placed in `<DAFFODIL_HOME>/Tools`.

On executing the batch file, it prompts the user for *Host* and *Port Number*. Default setting is **local machine** at port **3456**. If change is required then you can change the *Host* and *Port Number* according to your server settings. If connection is successful, browser window appears and another window appears prompting to enter a user name and password. Default User is “”string and default password is also “” string. After checking the user name and password Daffodil DB Browser Server can be started on the localhost. You can select any database from ‘Database node’.

Note: - For more information on how to start Daffodil DB server, refer to “Getting Started with Daffodil DB Guide”.

Daffodil DB Browser Operation modes

Daffodil DB Browser operates in two different modes:

- Query Analyzer Mode
- Browser Mode

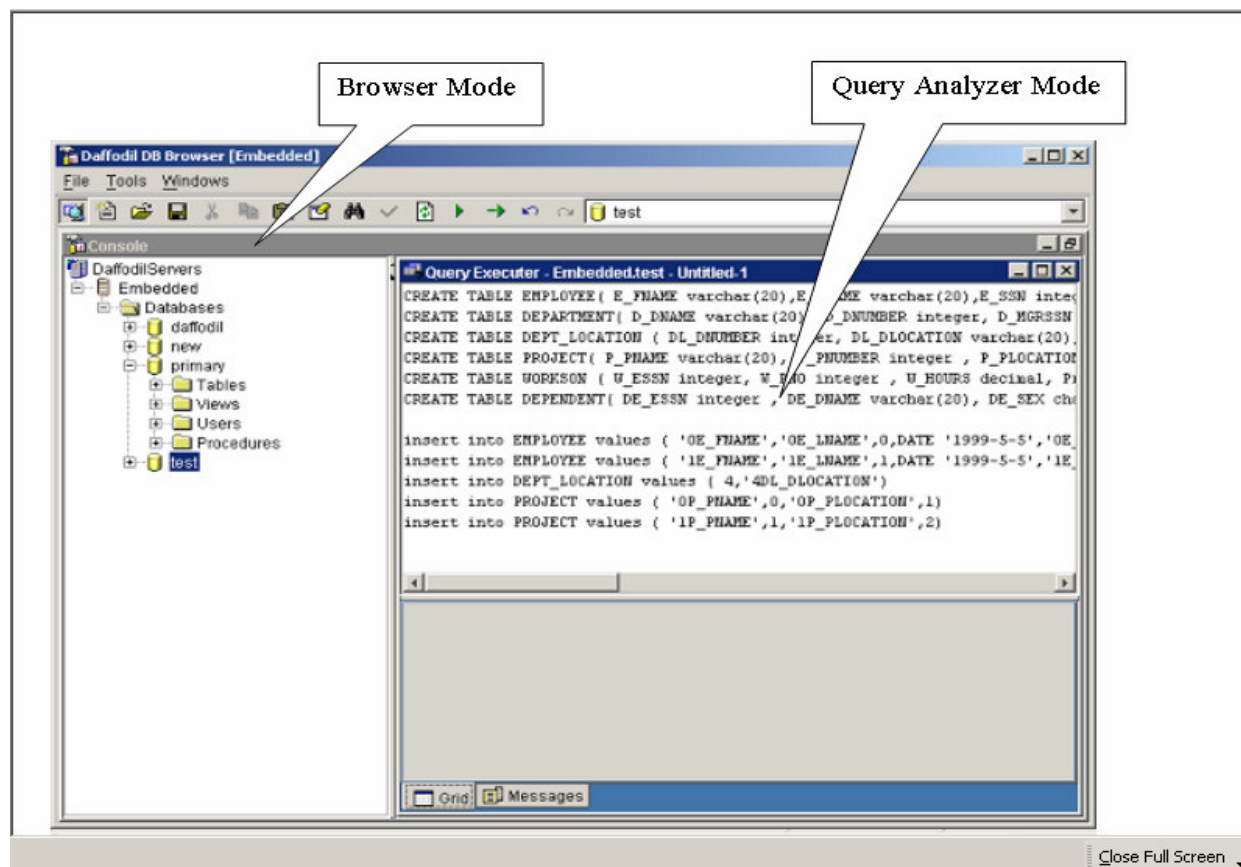
Query Analyzer Mode

Query Analyzer Mode is used for executing SQL statements and script files interactively. Query Executer Pad (QEP) is used to execute queries. Executer pad is invoked by selecting Query Analyzer Mode button from the toolbar of Daffodil DB Browser. You can use QEP to execute queries on different Daffodil DB databases in embedded or server mode.

In Query Analyzer Mode, you can execute any kind of query supported by Daffodil DB.

Browser Mode

Browser Mode is used for designing SQL statements and scripts interactively, to browse various database objects and view database objects properties. Browser Mode can be selected by deselecting Query Analyzer Mode button from the toolbar of Daffodil DB Browser. You can use Browser Mode to graphically design queries through visual interfaces.



Tool Bars of Daffodil DB Browser

Daffodil DB Browser toolbars can be categorized on the basis of different modes of Daffodil DB Browser as:

- Query Analyzer Mode Tool Bar
- Browser Mode Tool Bar

Query Analyzer Mode Tool Bar

Query Analyzer Mode tool bar offers following functionalities:

- Switch between Browser Mode and Query Analyzer Mode.
- Open unlimited number of Query Executer Pads.
- Cut, copy and paste text across clipboards and Query Execution Pads.
- Load script files.
- Select server, database and session.
- Execute queries.

Tool Bar Buttons

Query Analyzer Mode Button

Query Analyzer Mode Button is a toggle button to switch between Browser Mode and Query Analyzer Mode. When selected, it switches to Query Analyzer Mode and when deselected, it switches back to Browser Mode.

New Query Executer Button

It is used to open a new Query Execution Pad.

Load Script Button

It is used to load an SQL script file.

Save

It is used to save contents of QEP to a file.

Cut, Copy and Paste Button

It is used to edit text on QEP.

Clear Button

It is used to clear QEP.

Find

It is used to find text in QEP.

Refresh

It is used to refresh nodes shown in the browser from the database.

Execute

It is used to execute the selected query.

Undo

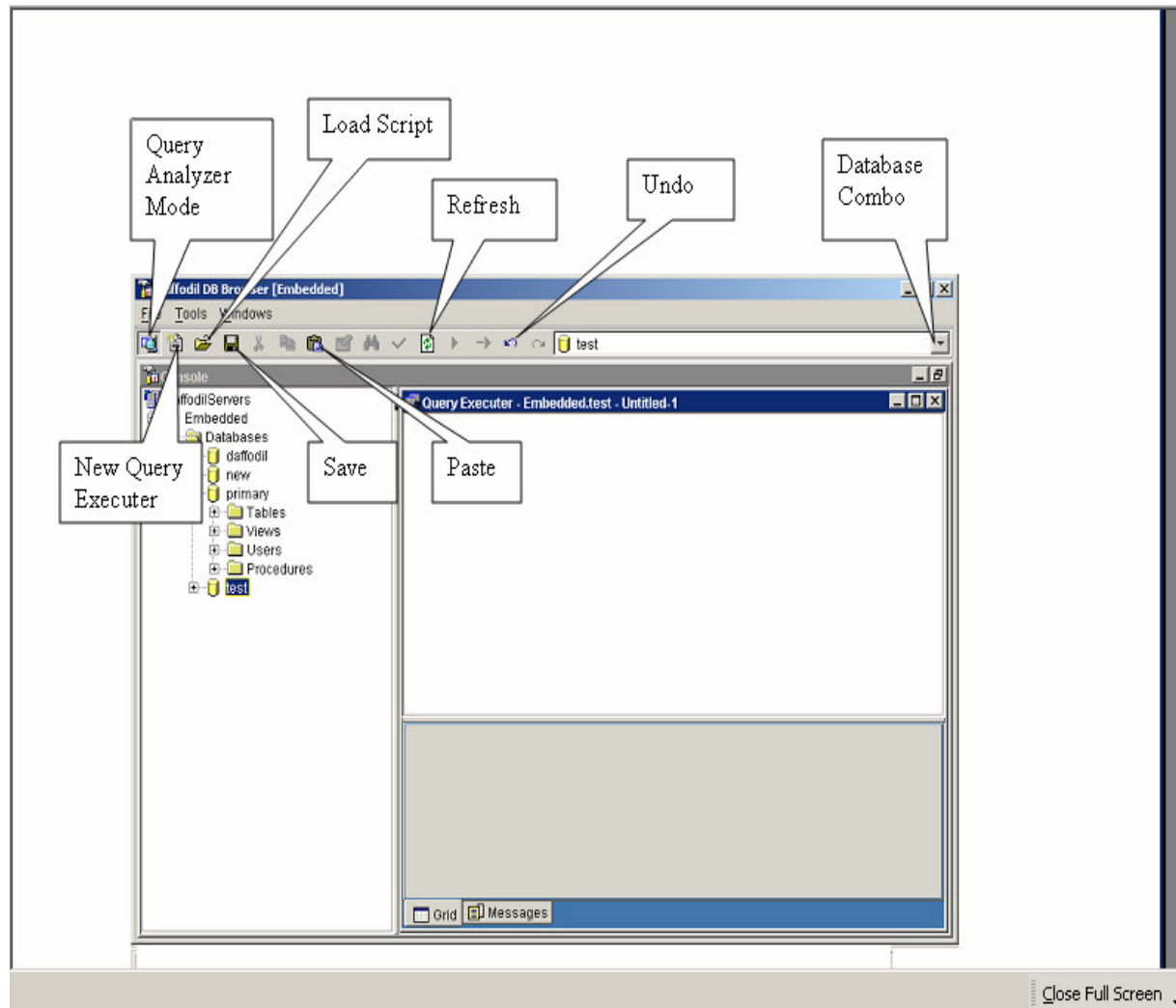
It is used to undo changes, character by character.

Redo

It is used to redo changes, character by character.

Database Combo Box

It is used to list the databases in the Daffodil DB embedded or server edition that is selected in the Server Combo Box.



Browser Mode Tool Bar

Browser Mode toolbar provides functionality of switching between Browser Mode and Query Analyzer Mode, selecting nodes and refreshing the nodes.

Browser Tool Bar Buttons

Query Analyzer Mode

This is a toggle button to switch between Browser Mode and Query Analyzer Mode. When selected, it switches to Query Analyzer Mode and when deselected, it switches back to Browser Mode.

Up One Level

On clicking this button, the one level up node will be selected in the hierarchy.

Refresh

This button ensures that latest nodes are shown in the browser. It refreshes content of the browser.

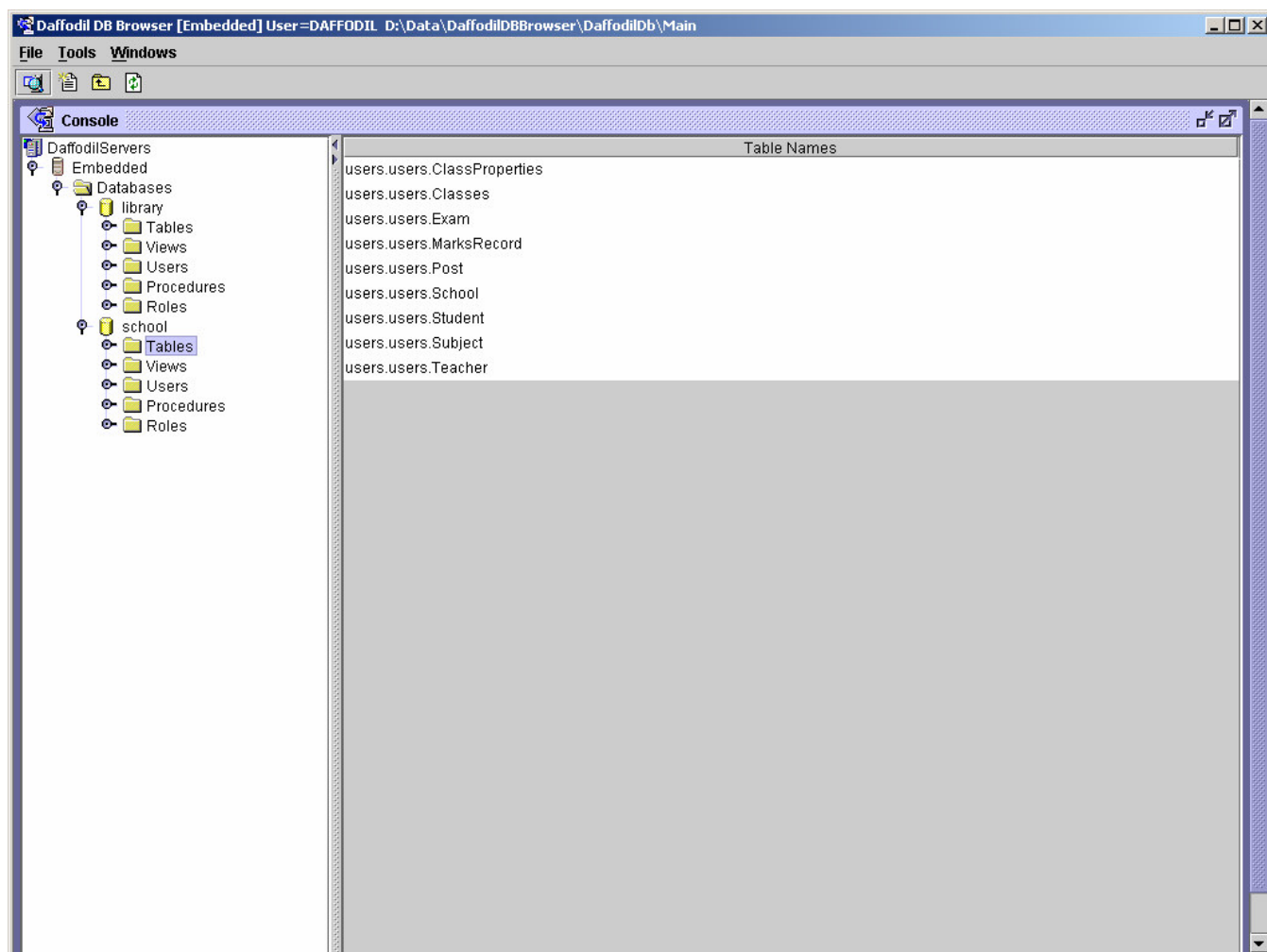


Table Tool Bar Buttons

Table tool bar is displayed when a new table is created using Daffodil DB Browser in the Browser mode.

Column Primary Key

It assigns or removes primary key constraint to the selected column.

To assign primary key to a column, select the column and then click on the *Primary Key* toggle button. If Primary Key toggle button is pressed again, it removes the primary key from the corresponding column

Table Primary Key

It creates primary key for the selected column(s).

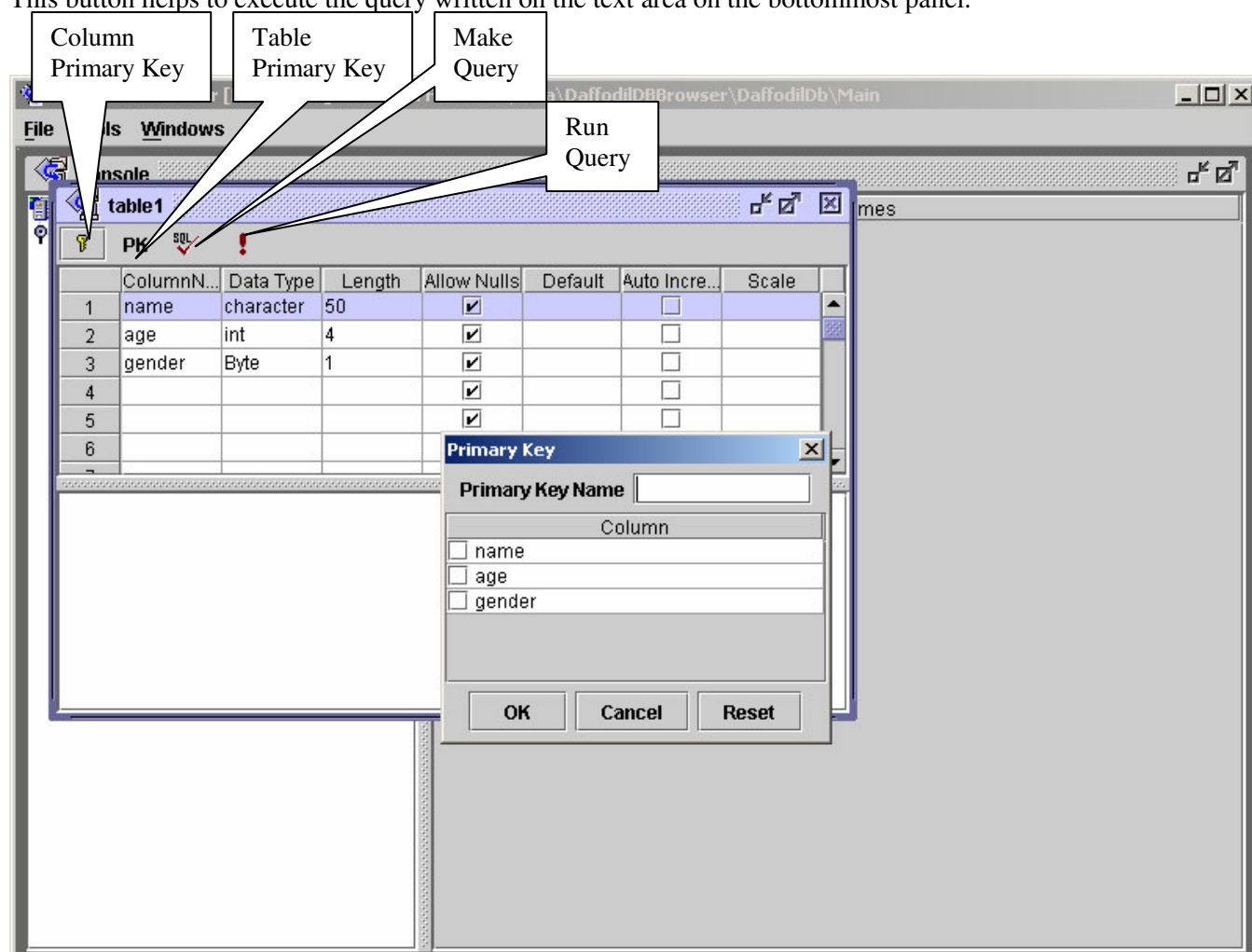
When we click on this button, a dialog box opens to choose column(s) for primary key and to assign name for this primary key.

Make Query

This creates a *Create table* query based on the options chosen for creating a table, and displays it on the text area on the bottommost panel.

Run Query

This button helps to execute the query written on the text area on the bottommost panel.



View Tool Bar Buttons

Add Table

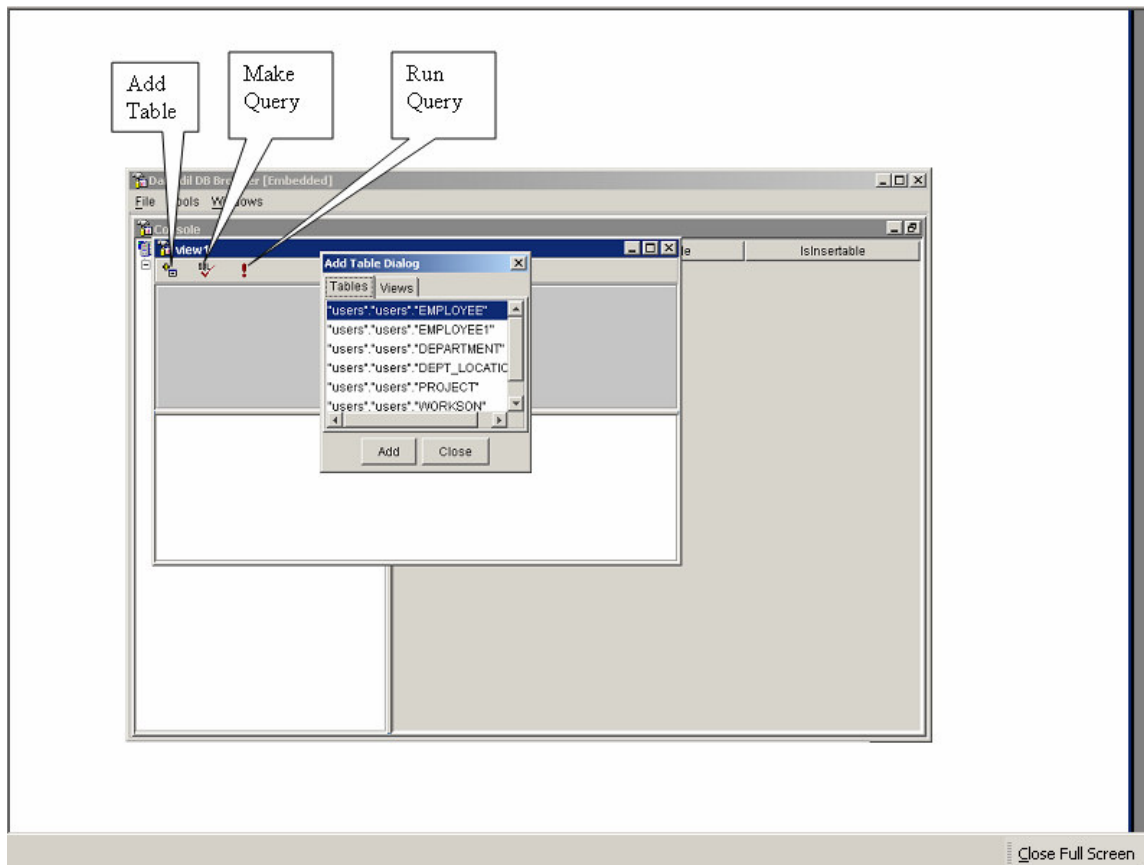
Clicking on the Add Table button opens an Add Table Dialog box for the user to select tables for creating a View.

Make Query

This creates a query for the user and shows it on the text area on the bottommost panel of the view frame.

Run Query

This is used to execute a query.



Menus

Menu Bar

Menu Bar provides the functionality of creating a database in the selected server, connecting to a server, disconnecting from a server and to exit the Daffodil DB Browser.

There are three types of menus in the Menu Bar namely:

- File Menu
- Tool Menu
- Windows menu

Apart from the above three, Daffodil DB Browser supports the following menus also:

- Query Analyzer Pop up Menu (which appear by right clicking on Query Executer Pad)
- Popup Menus (which appear by right-clicking on a node)

File Menu

CreateDatabase

This menu option asks user to enter name of the database to be created and hence creates a new database.

ReConnect

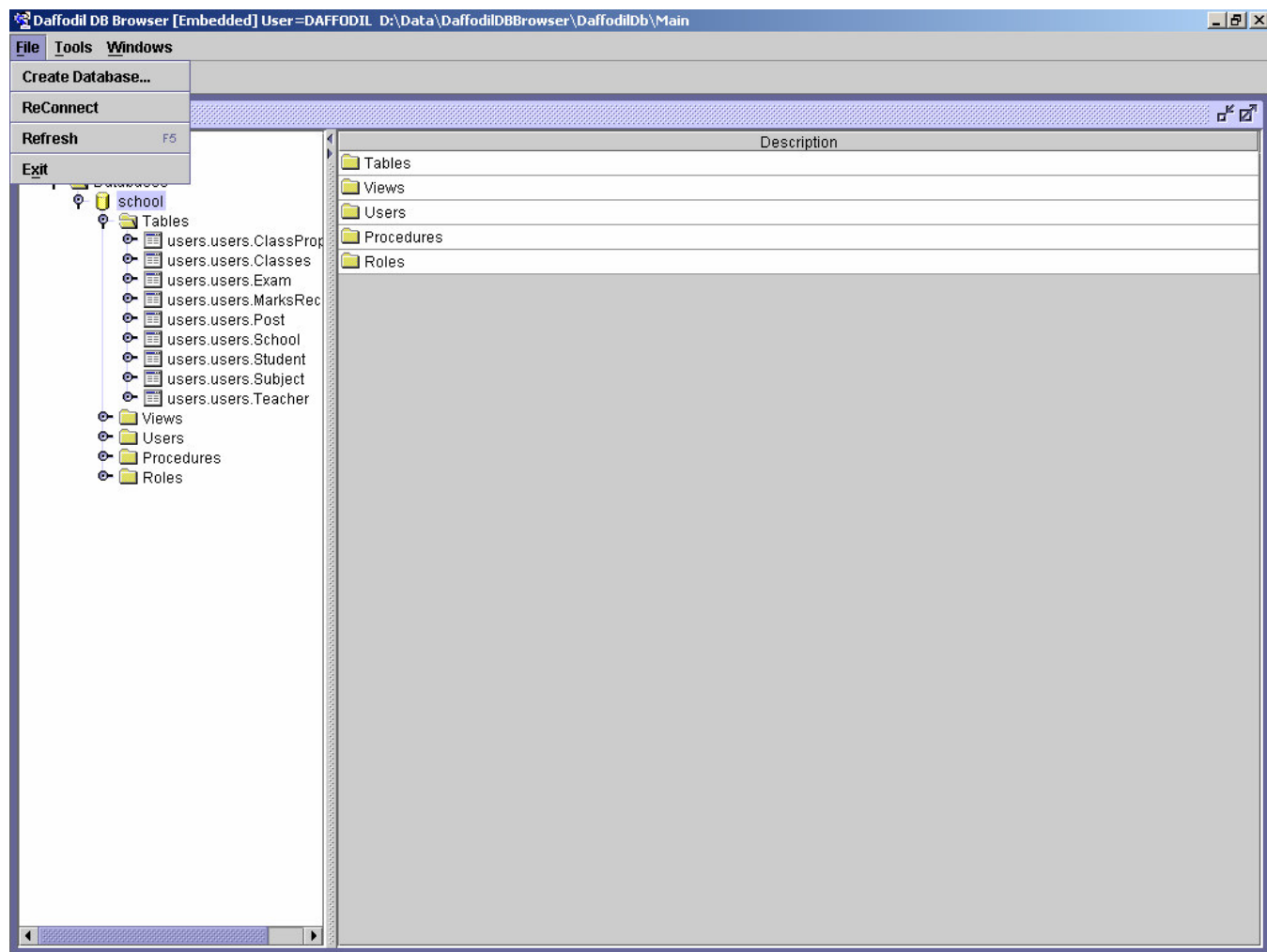
It is a Logoff/Login feature. It closes all the connections and opens *Login* dialog box to connect to the database with a new User.

Refresh

This menu option reloads all information from the database.

Exit

It closes Daffodil DB Browser.



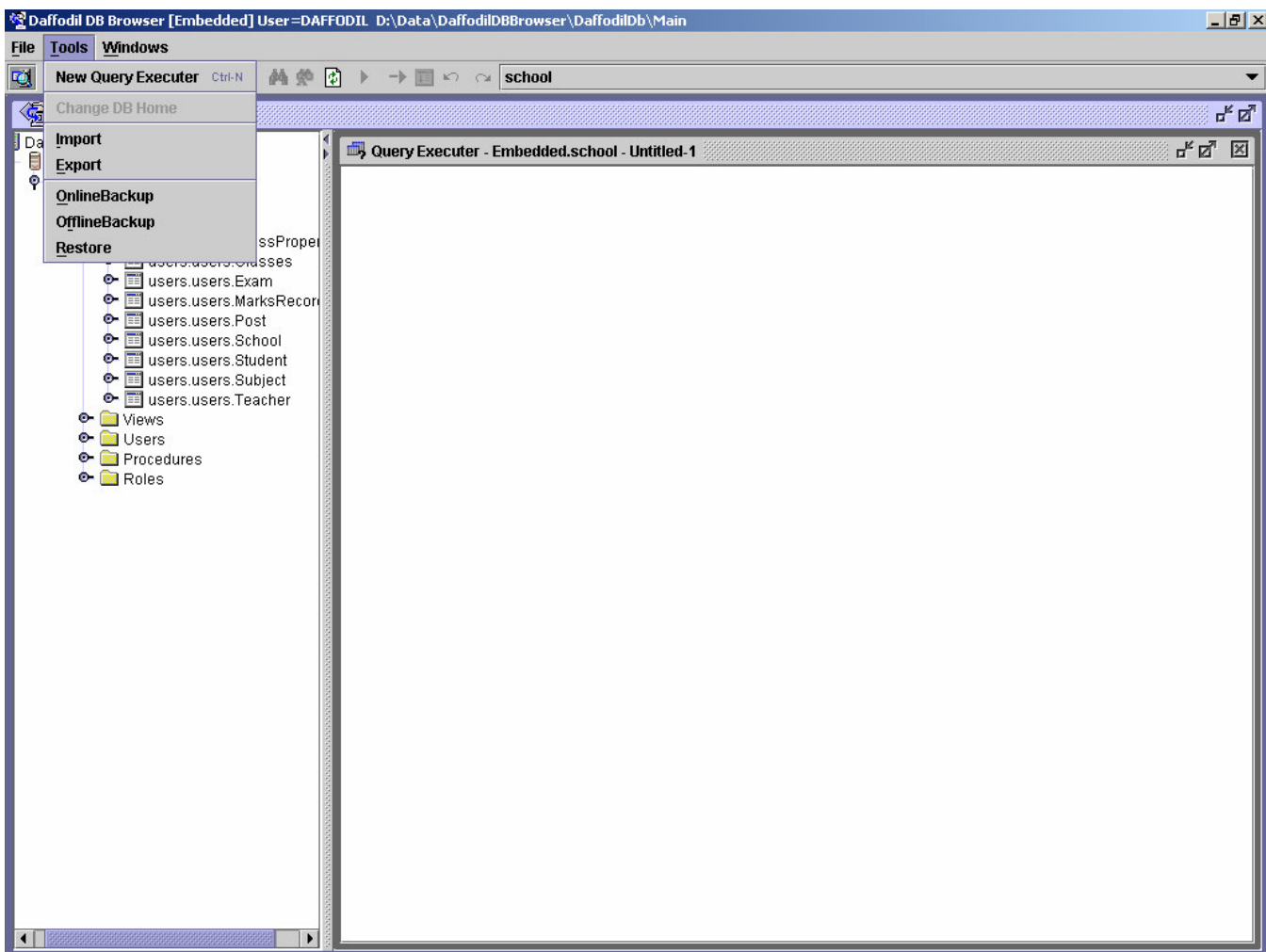
Tools Menu

New Query Executer

It is disabled in Browser Mode but it becomes enabled when user selects the Query Analyzer Mode by clicking the toggle button. It opens a new QEP.

Change DB Home

On selecting this menu option, user is given an option to close all existing sessions in the database. If user opts **yes**, then user is prompted to enter a new Daffodil DB home path and this mentioned path is set as new DAFFODILDB_HOME.



Import

Export

OnlineBackup

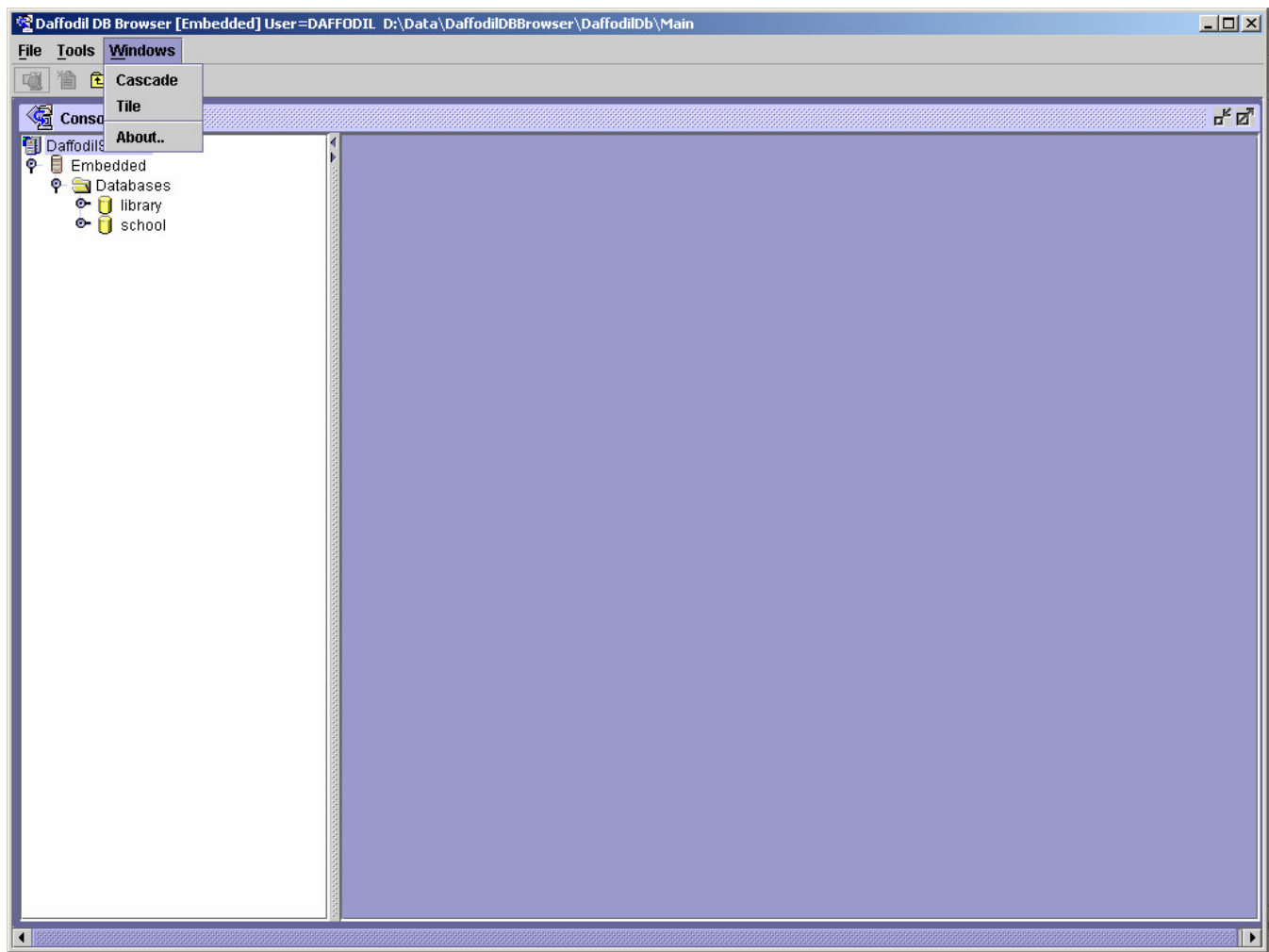
OffLineBackup

Restore

Note: - Detailed Description of above mentioned is given on page no. 66 onwards

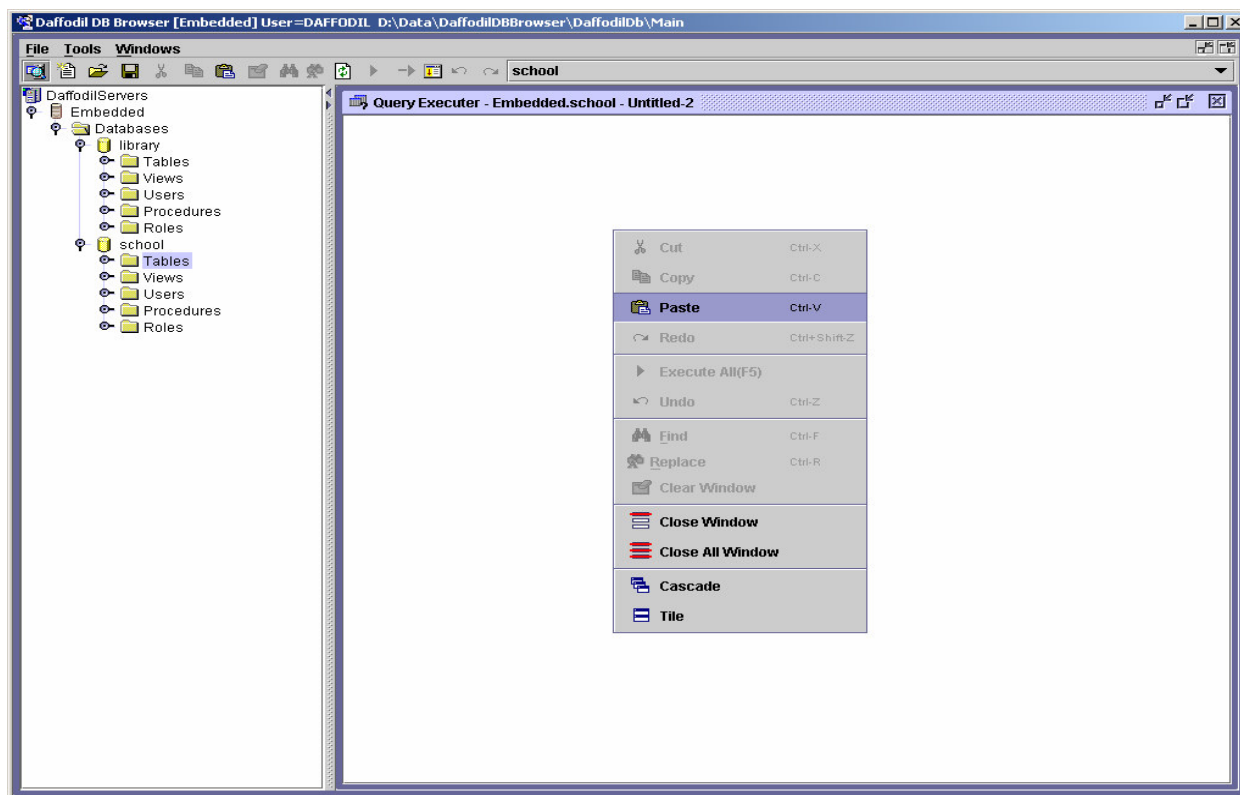
Windows Menu

This menu has simple options that any GUI based application should possess. They are **cascade**, **tile** and **about**.



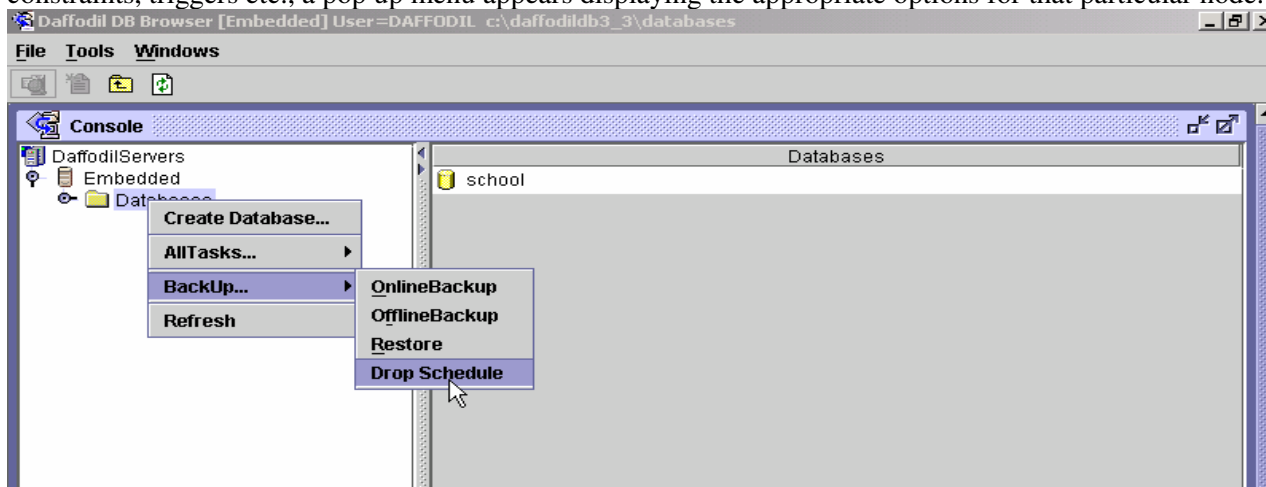
Query Analyzer Pop up Menu

On right clicking the Query Execution Pad, the Query Analyzer Pop up Menu appears and it displays the options for cascading or tiling Query Execution Pads, to open the current QEP or all QEPs. It has features of cut, copy, paste, undo, redo, execute, find and clear text.



Pop Up Menu

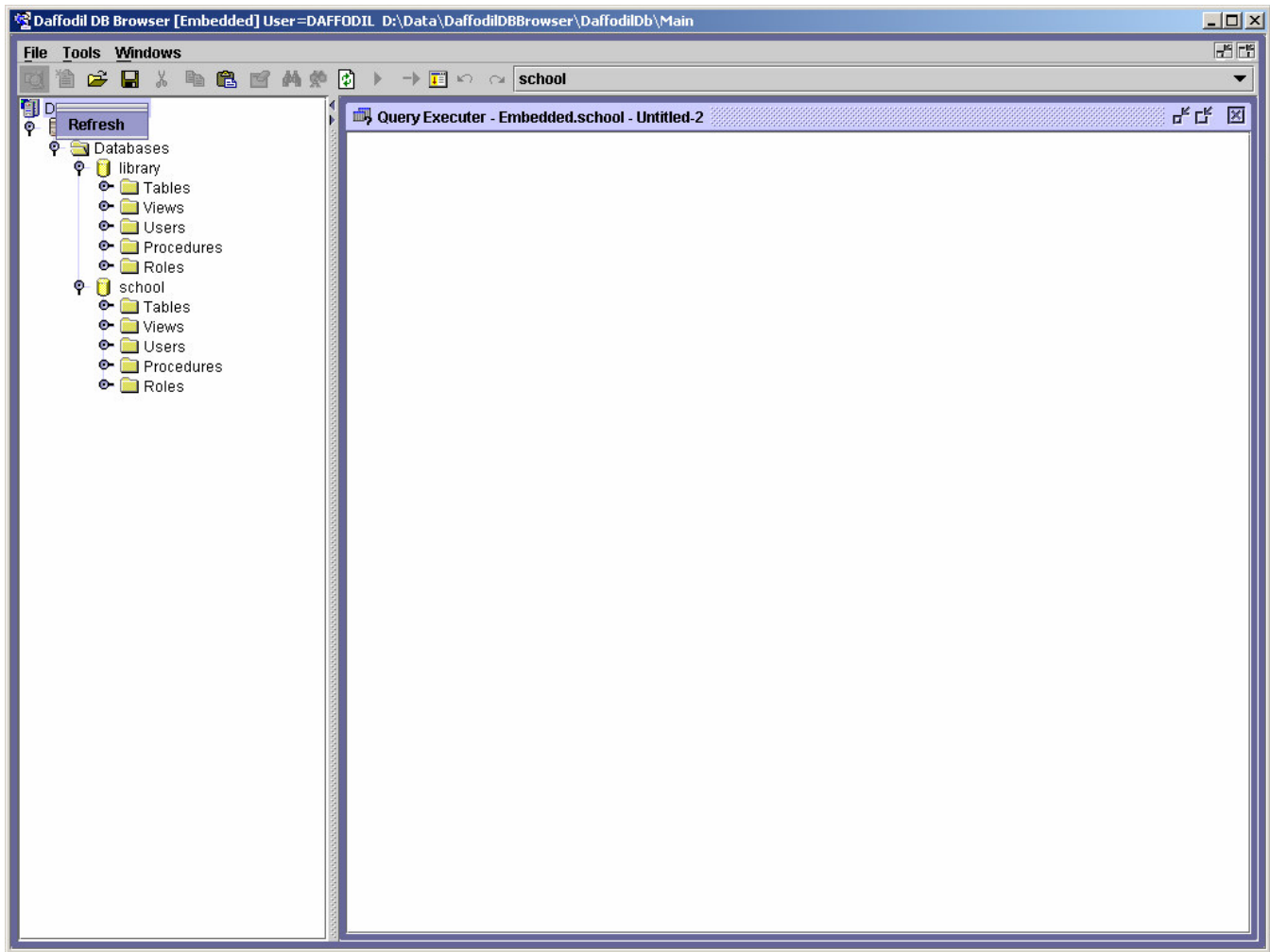
On right clicking on any node like databases, tables, views, roles, procedures, users, columns, indexes, constraints, triggers etc., a pop up menu appears displaying the appropriate options for that particular node.



Different Types of Nodes displayed in Browser Mode

Daffodil Servers Node

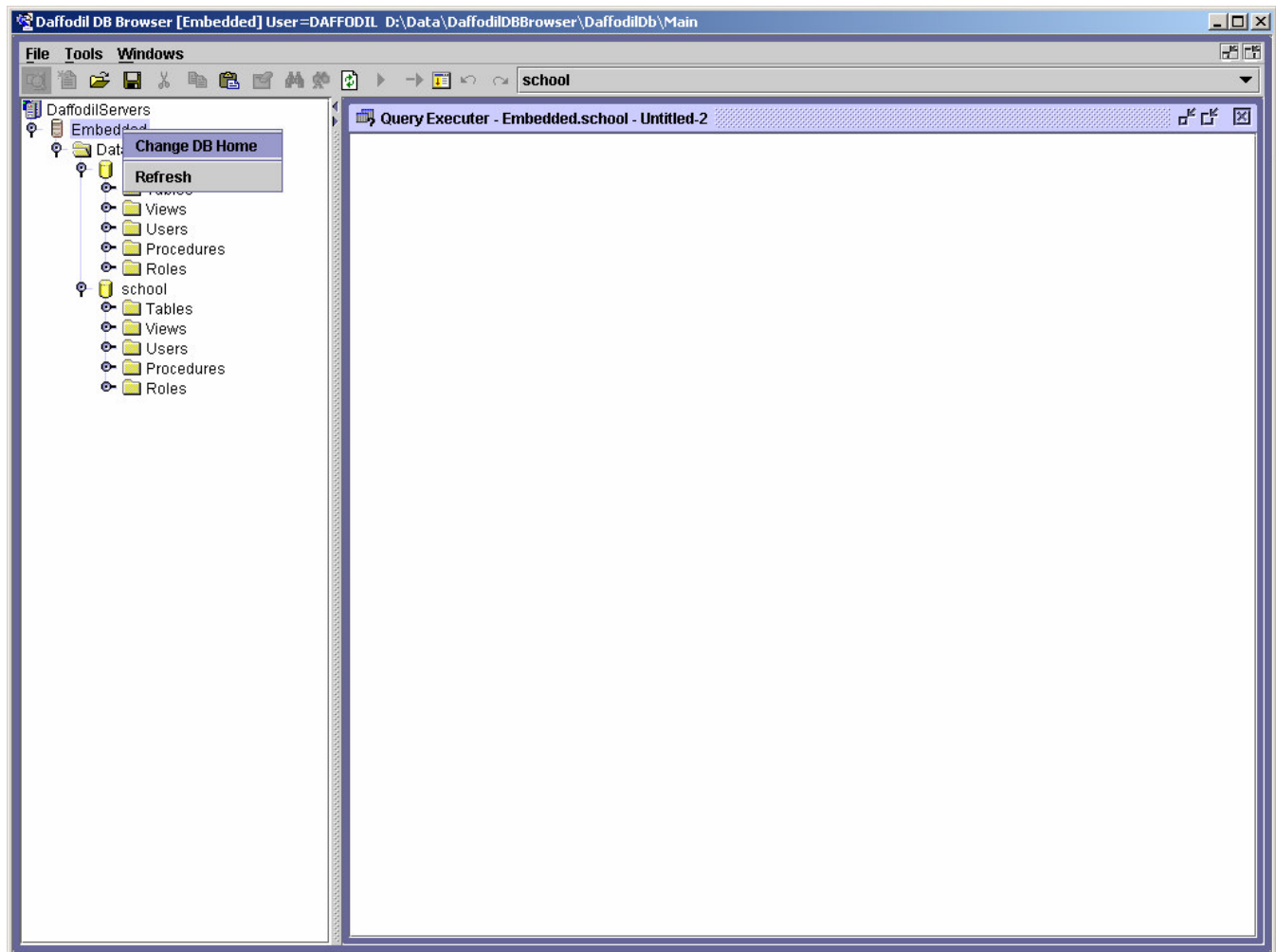
On right clicking the node, a pop up menu appears displaying options for refreshing the contents of the tree.



Embedded Node

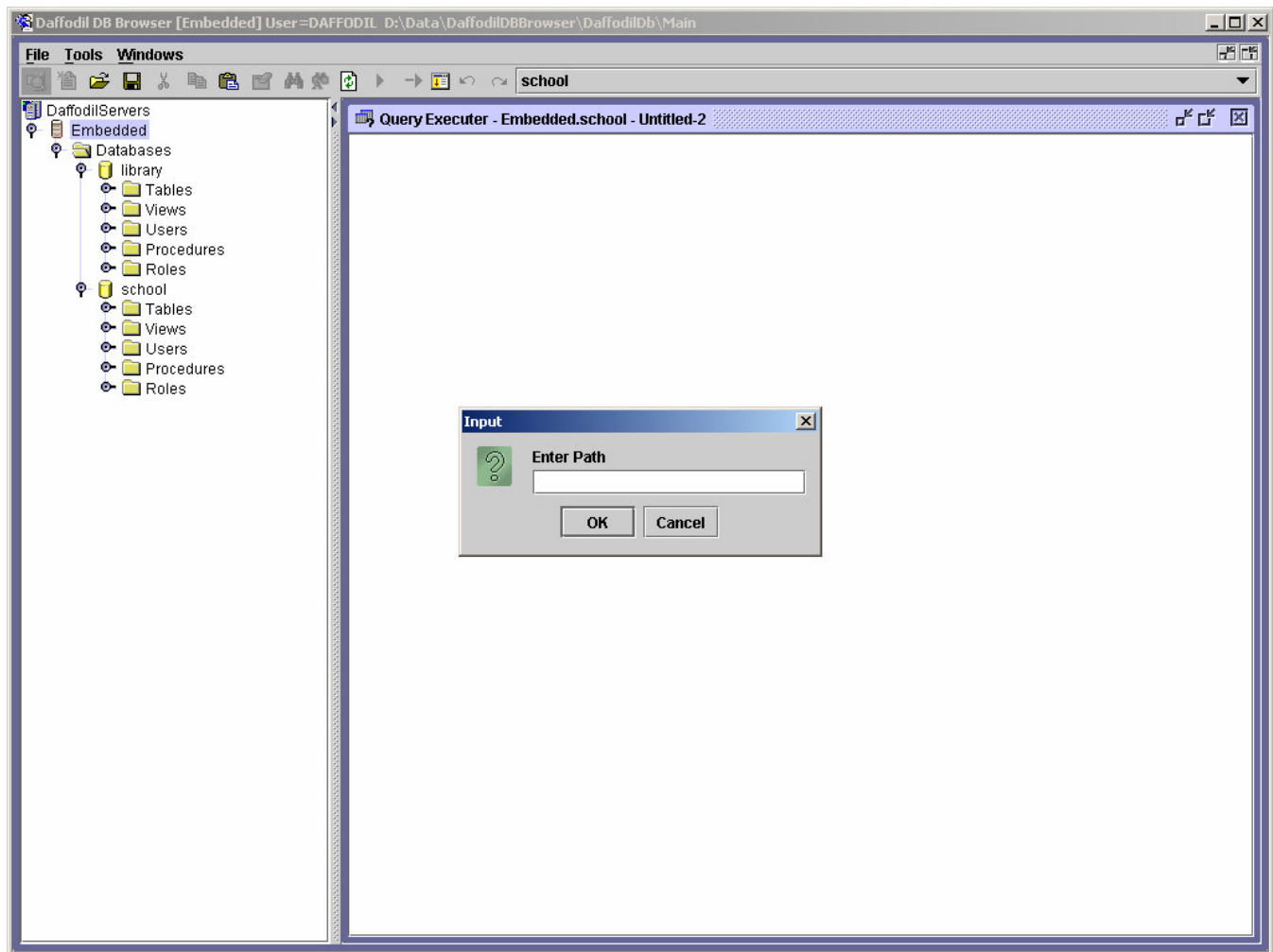
On right clicking the node a pop up menu appears showing the options of changing DAFFODILDB_HOME for Daffodil DB, creating a new database and to refresh the contents.

Note: - For more information on DAFFODILDB_HOME, refer to Getting Started with Daffodil DB Guide.



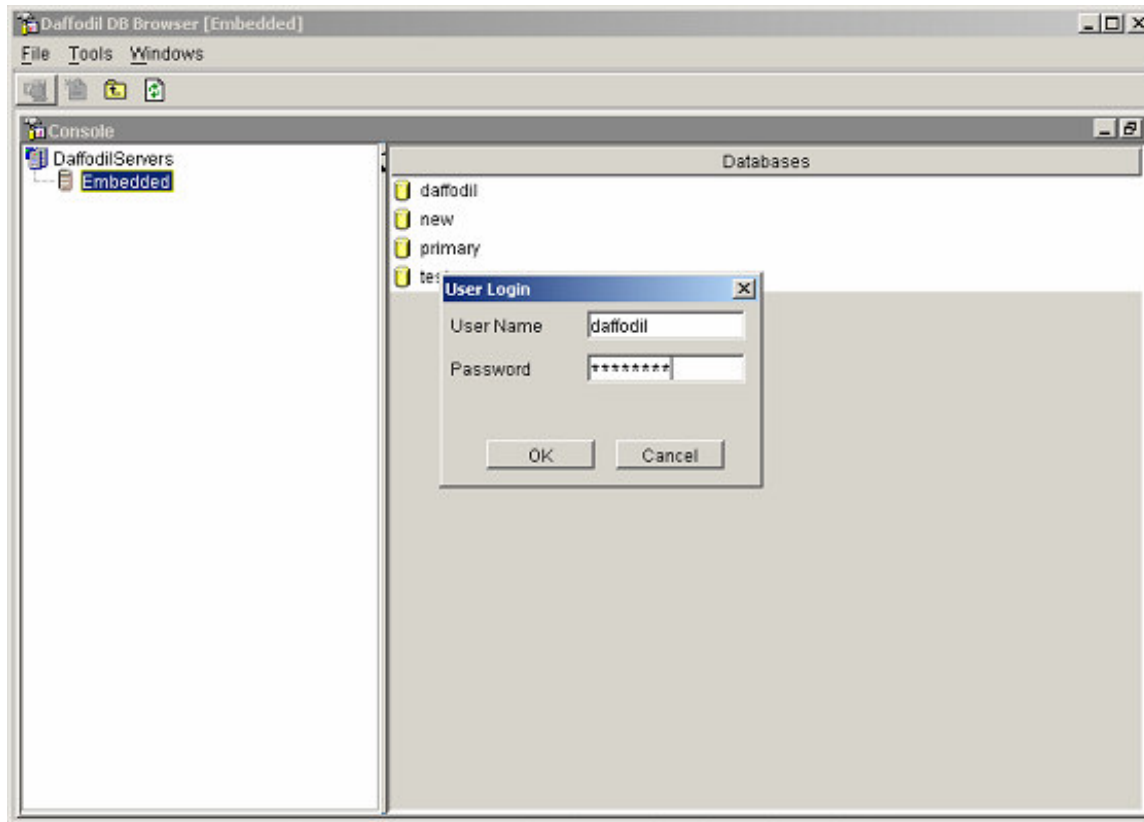
Change Daffodil DB Home

On selecting this option user is given an option if he wants to close all existing sessions on the database. If user selects yes, then he is prompted to enter a new Daffodil DB home path and this path is then set as new DAFFODILDB_HOME.



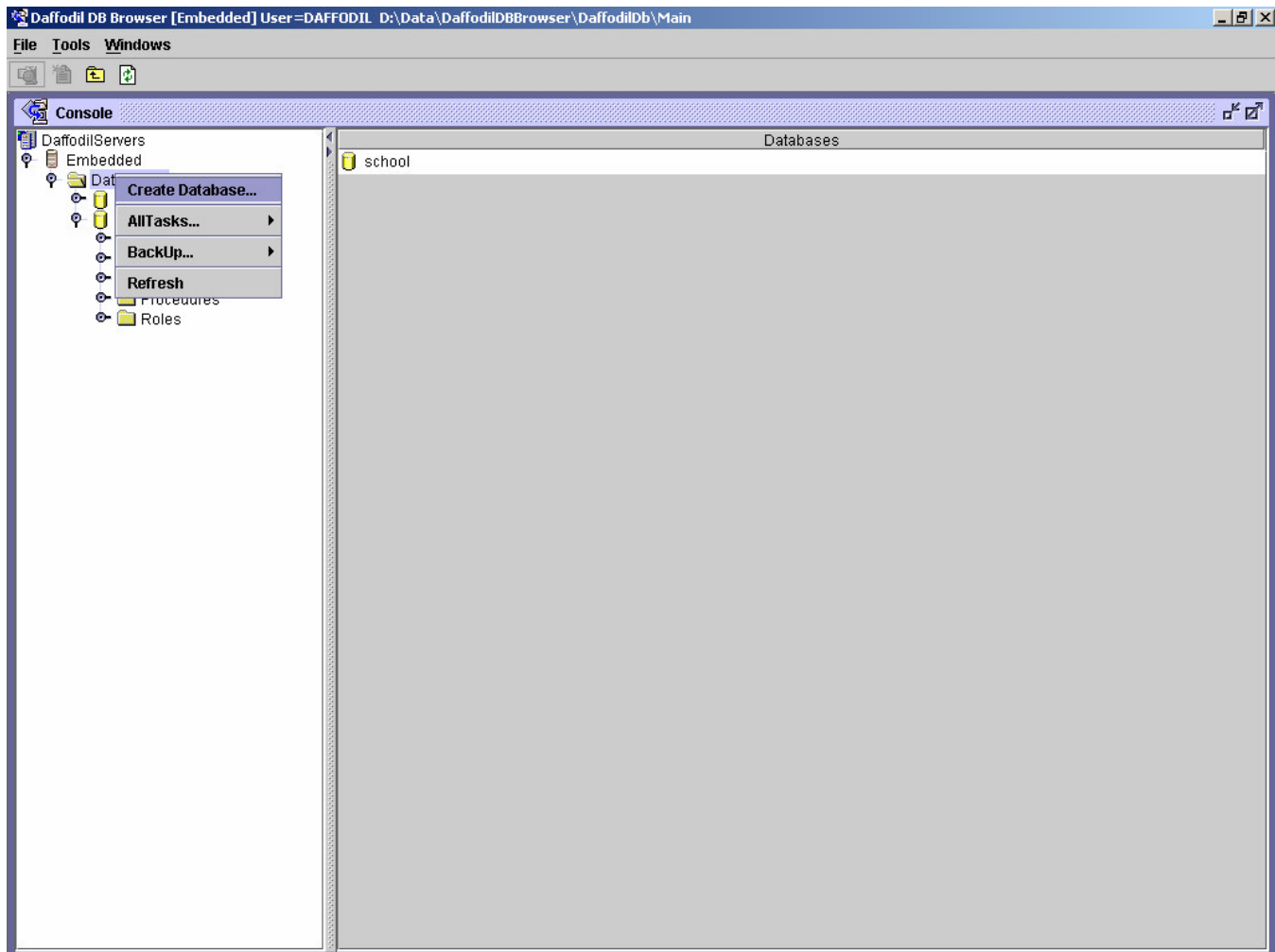
Username and password

After entering the new path, and clicking OK, a dialog box appears, requesting the user to enter name for the new database. After entering database name and clicking OK, database with the specified name is created.



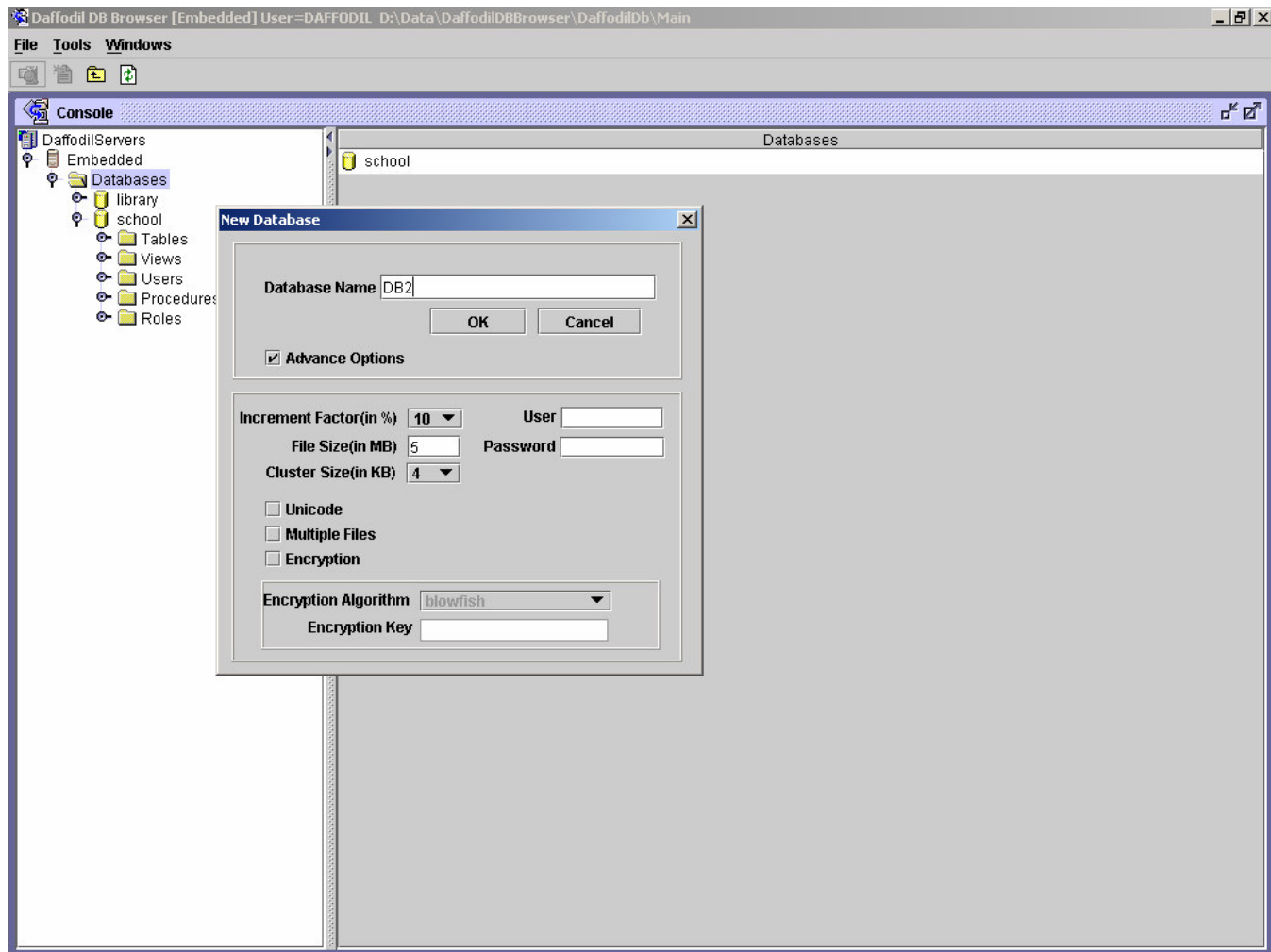
Databases Node

On right clicking the Databases node, a pop up menu appears showing the options for making new database, Import/Export, Back-Up and refreshing the contents. And the databases present on that server are displayed on the right panel of the Daffodil DB Browser and in the object tree.



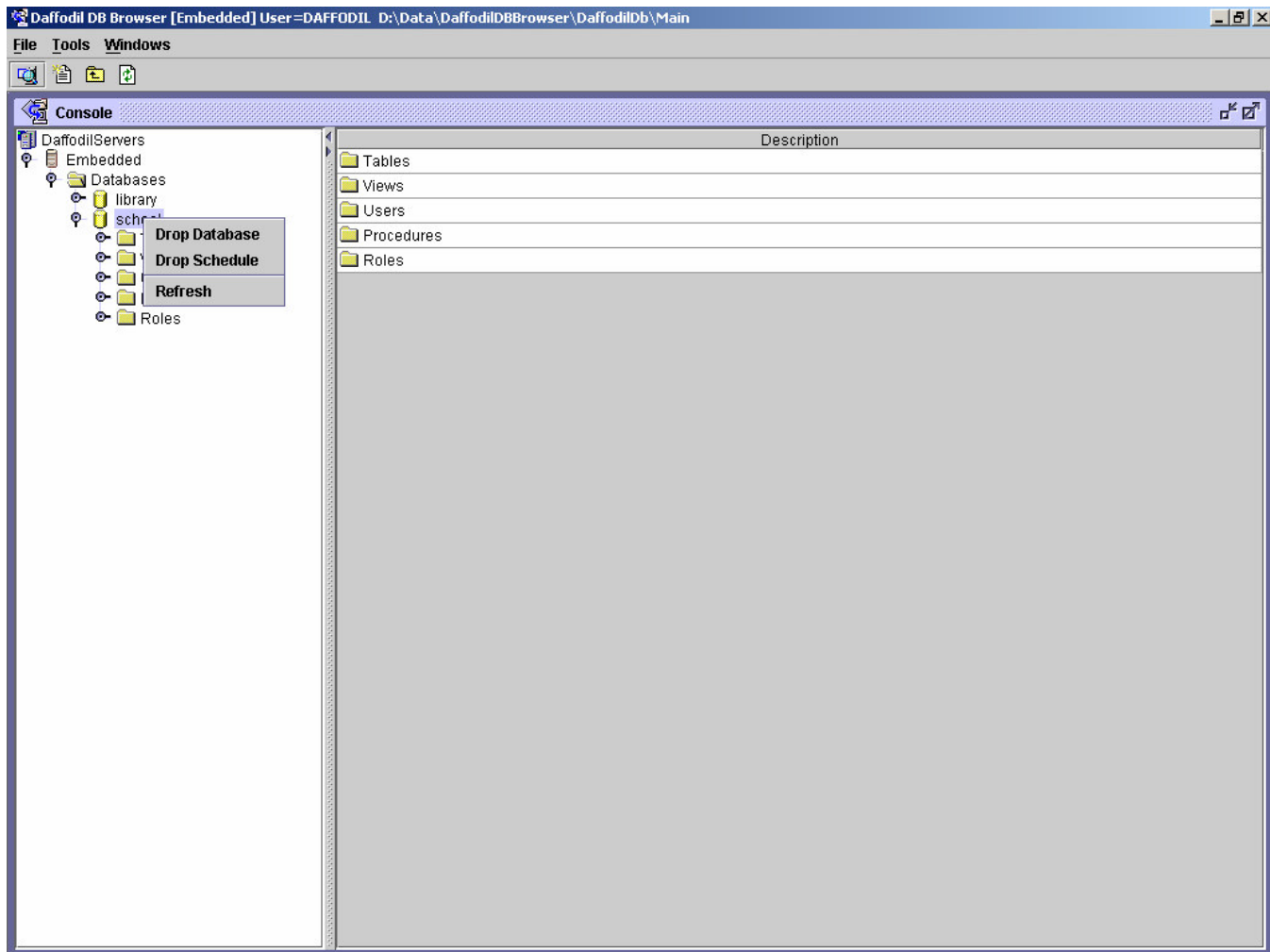
Create database

On clicking the Create Database option, it opens up a dialog box prompting for database name. On clicking the Advance Option button, the dialog box will extend to specify Optional properties like user name, password, increment factor, file size, cluster size, unicode support, multifile support, encryptions algorithm and encryption key.



Database Name Node

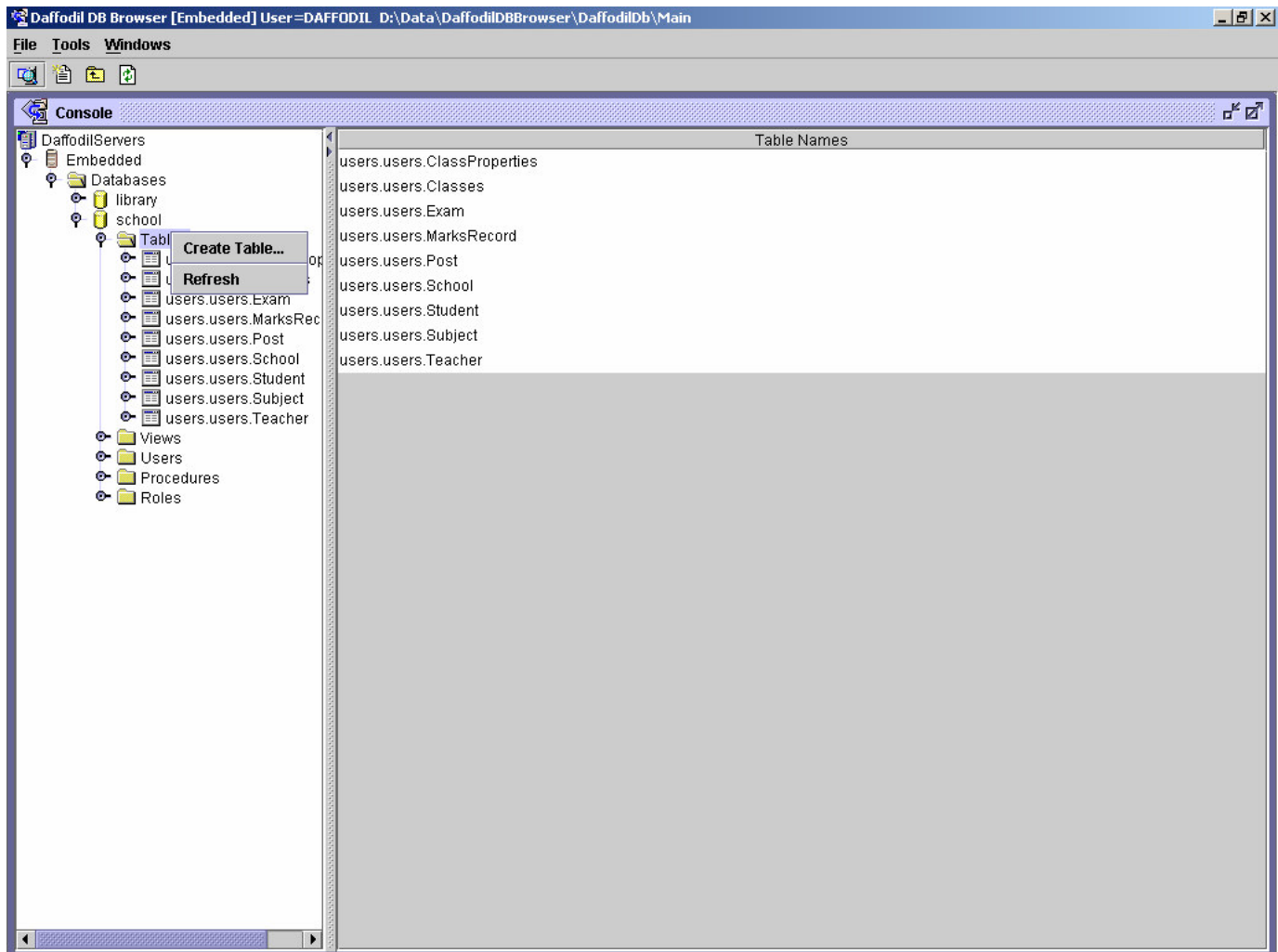
On clicking a particular database, list of various database objects like Tables, Views, Users, Procedures and Roles gets displayed on the right pane of the window. Also, a particular database node expands to display nodes for each database objects present in the server in the object tree. On right clicking, a pop up menu appears displaying options for dropping a database, scheduling and refreshing the contents.



Tables Node

On clicking the *tables* node, existing table names in the selected database are shown on the right pane of the window. Also, *tables* node get expanded to show nodes for each table objects that are present in the database in the left panel of the browser.

On right clicking Tables node, a pop up menu appears displaying options for creating new table and refreshing the contents. The new table is created in the database under which *tables* node is displayed.



New Table

It opens a New Table Frame for creating a table. The Table Frame holds place for specifying column names, their data types, length of the column and other column properties. After making the entries, click on the Make Query button and you will be prompted to enter the database name. Enter the database name and click OK. Now you can write your queries in the lower part of the table frame

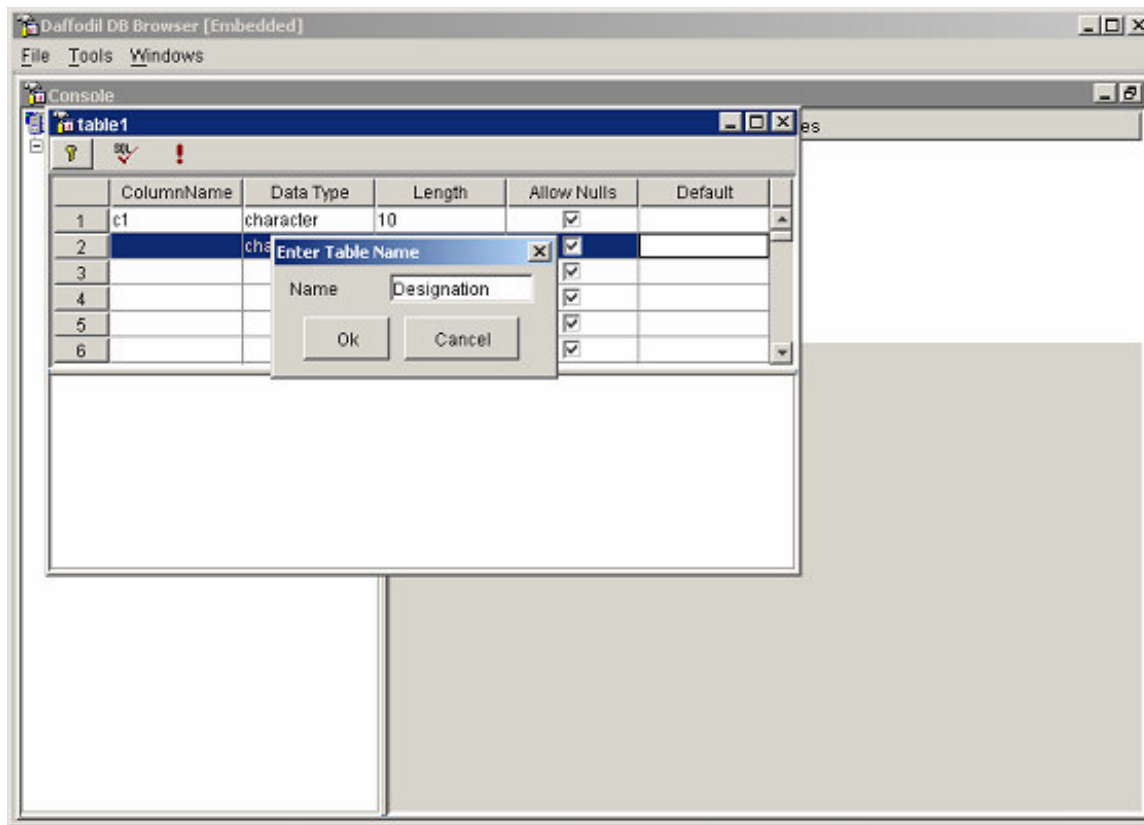


Table Name Node

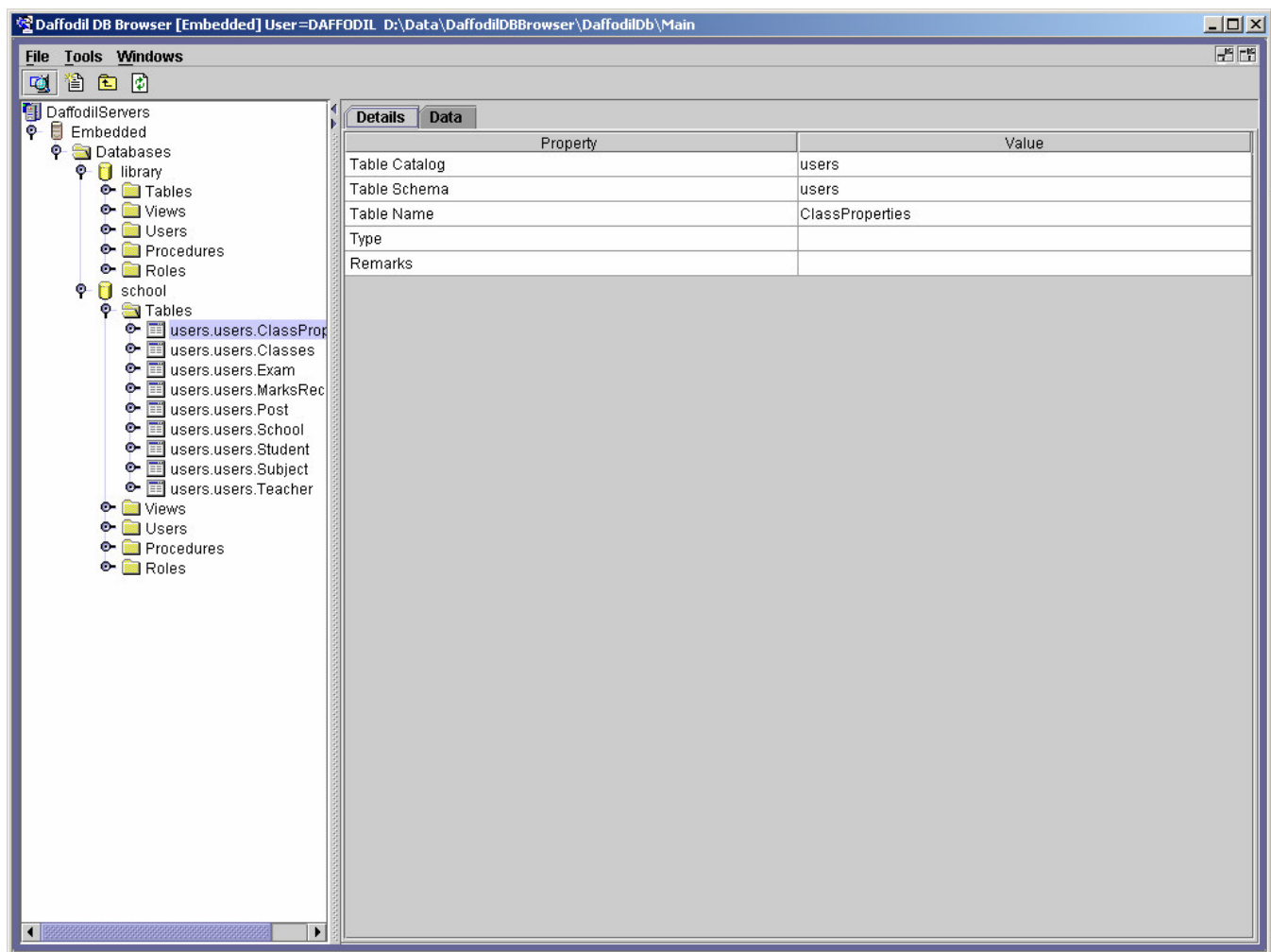
On clicking any table name node, a tabbed pane is displayed on the right pane of Daffodil DB Browser having two tabs named “Details” and “Data”.

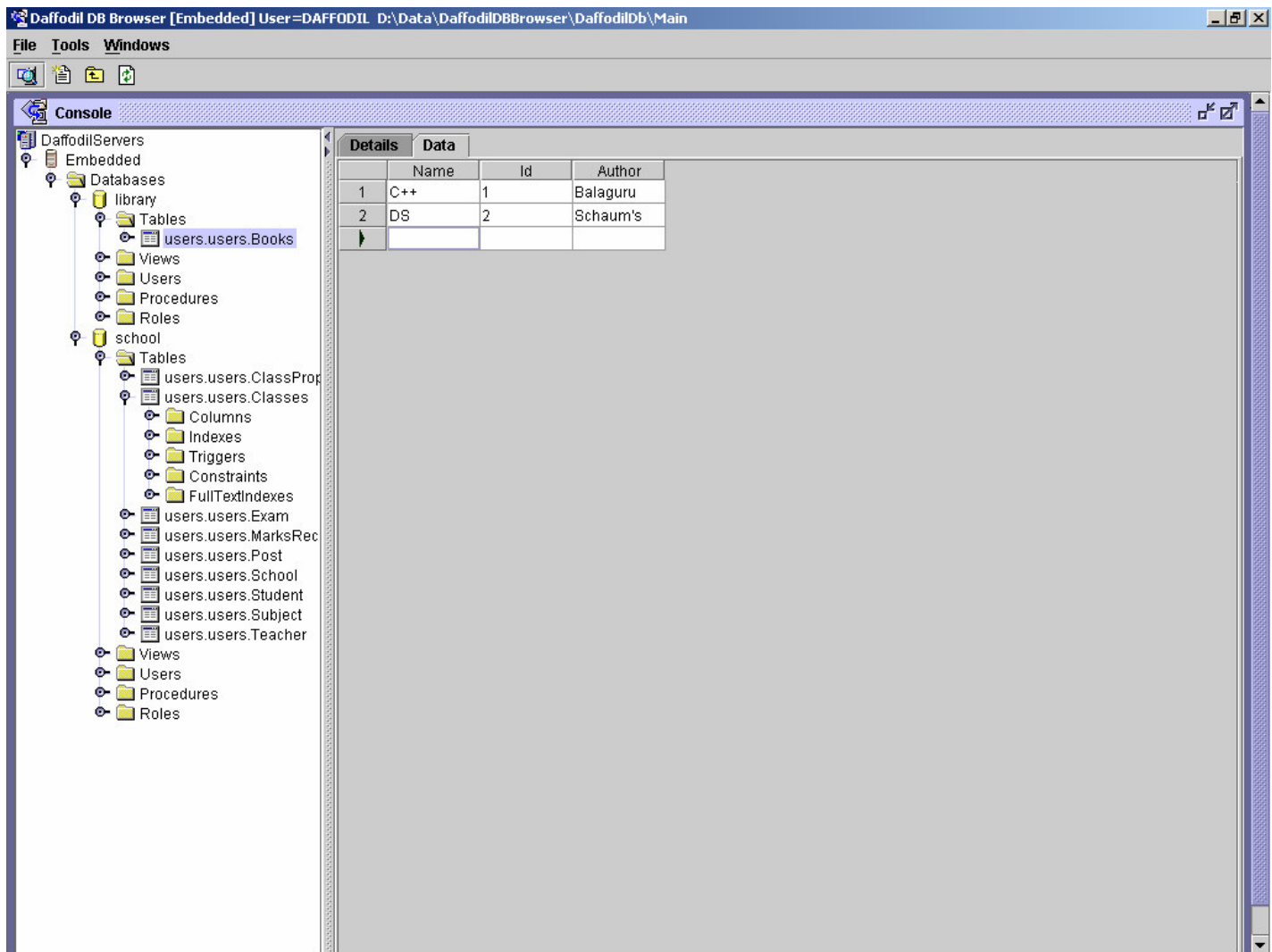
Details Tab

Selecting Details tab displays the properties of a table (Table Catalog, Table Schema, Table Name, Type and Remarks) and its corresponding values.

Data Tab

Selecting Data tab shows the sessions in which current user is working and the records present in that table.





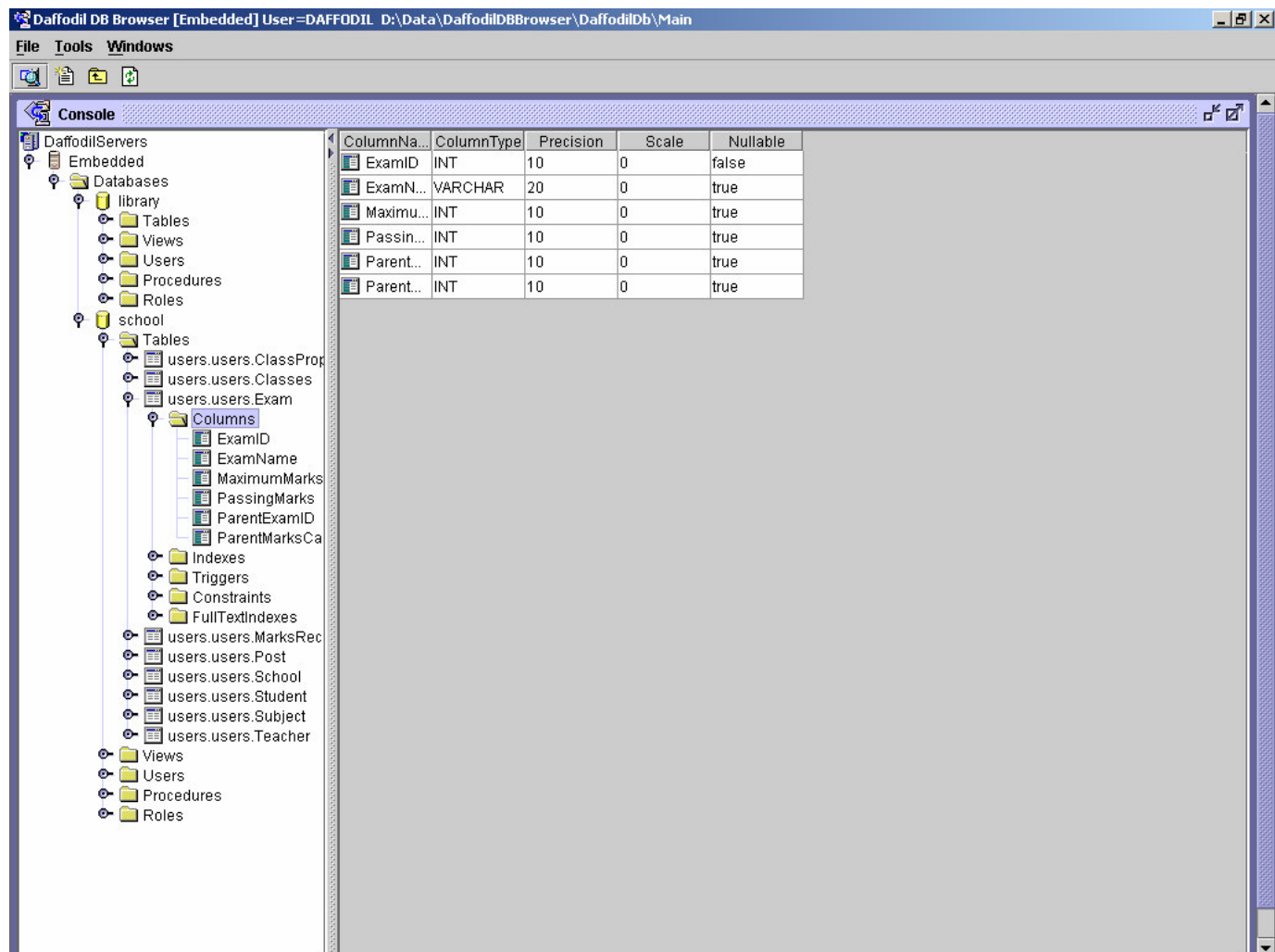
Also, the Table Name node is expanded displaying five child nodes:

- Columns Node
- Indexes Node
- Triggers Node
- Constraints Node
- FullTextIndexes Node*

* Features that are not supported in One\$DB

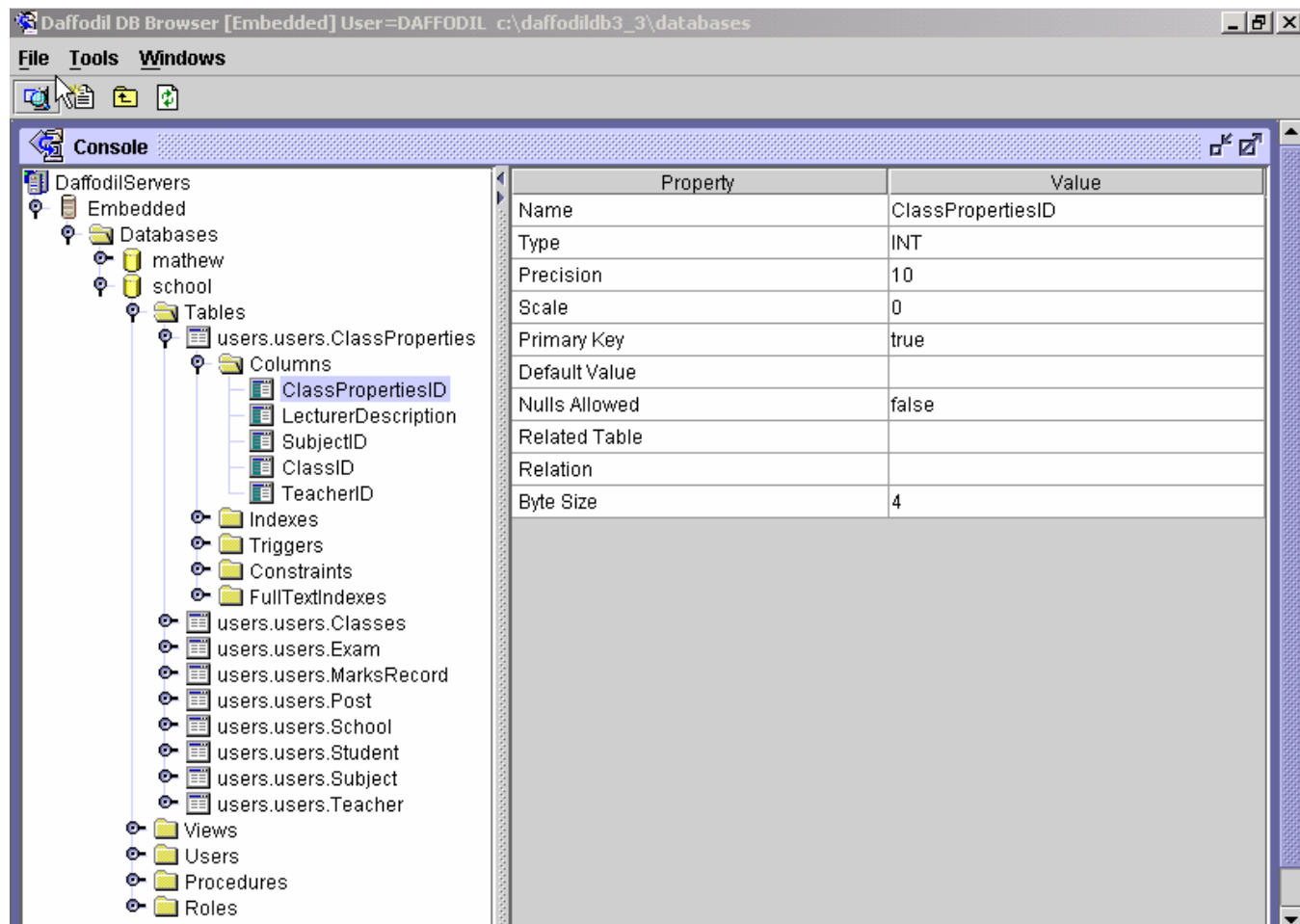
Columns Node

On clicking columns node, it gives an overview of the properties of all the columns present in the table. These properties are Column Name, Column Type, Precision, Scale and Nullable. Also, the columns node is expanded to display all the Column Names in a table in the left pane of the window.



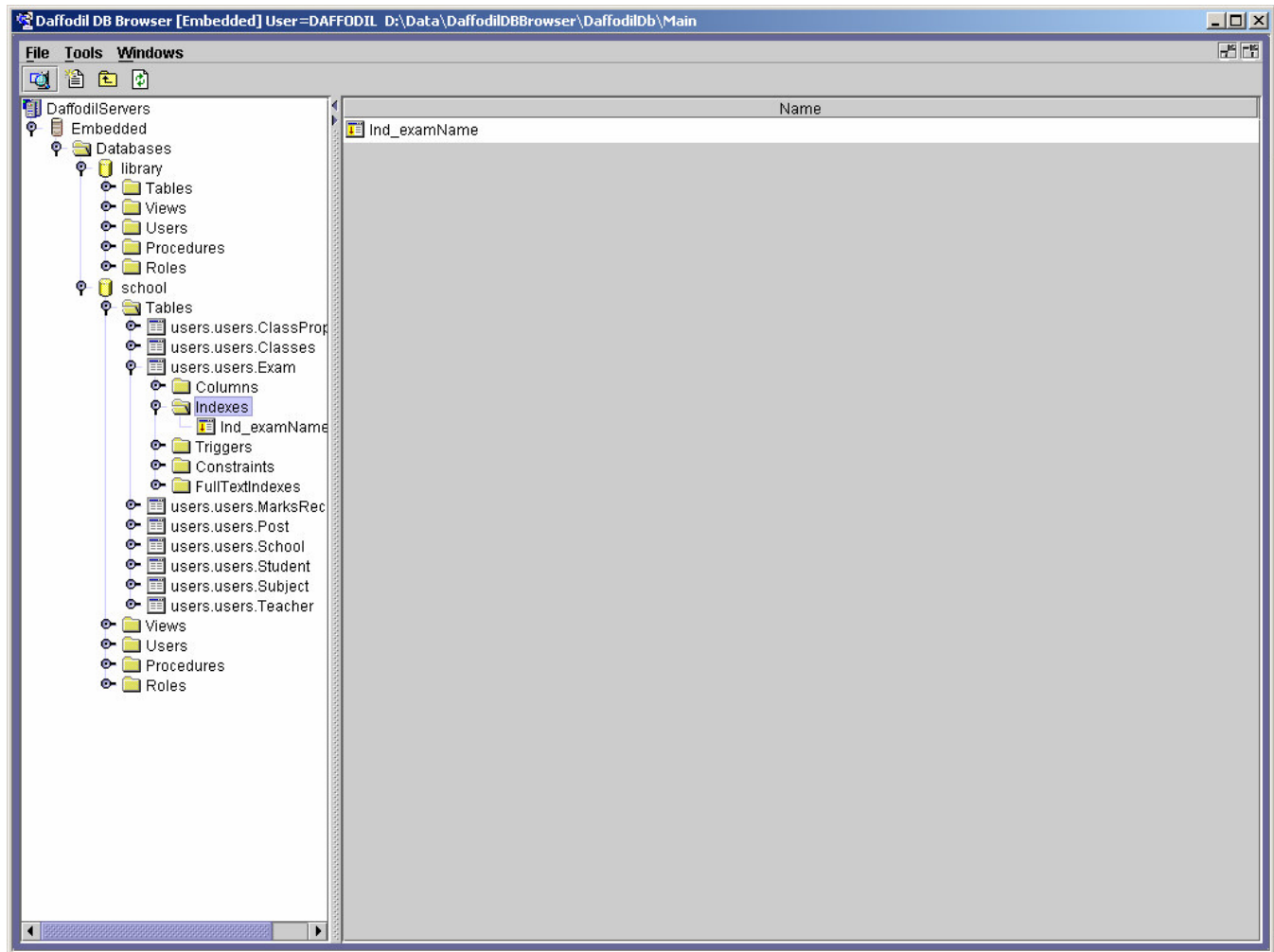
Column Name Node

On clicking the Column Name node, it gives a detailed view of the properties and values of that particular column, which is present in that table. These properties are Name, Type, Precision, Scale, Primary key, Default Value, Nulls Allowed, Related Table, Relation, and Byte Size.



Indexes Node

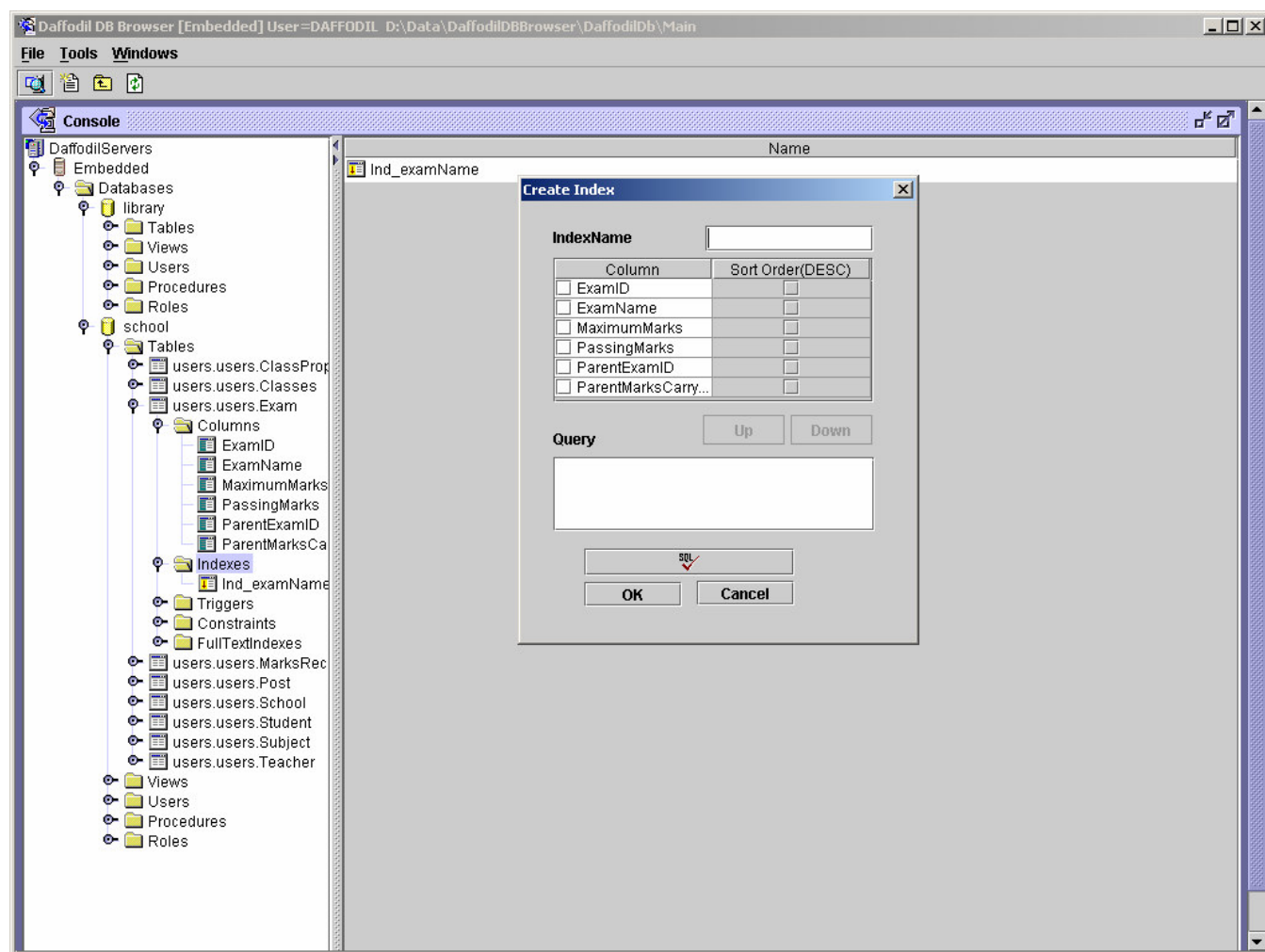
On clicking the indexes node, it gives all the indexes made on that table on the right pane of Daffodil DB Browser. Also, the node is expanded to display all the indexes on the table in the left pane of the window.



Create Index

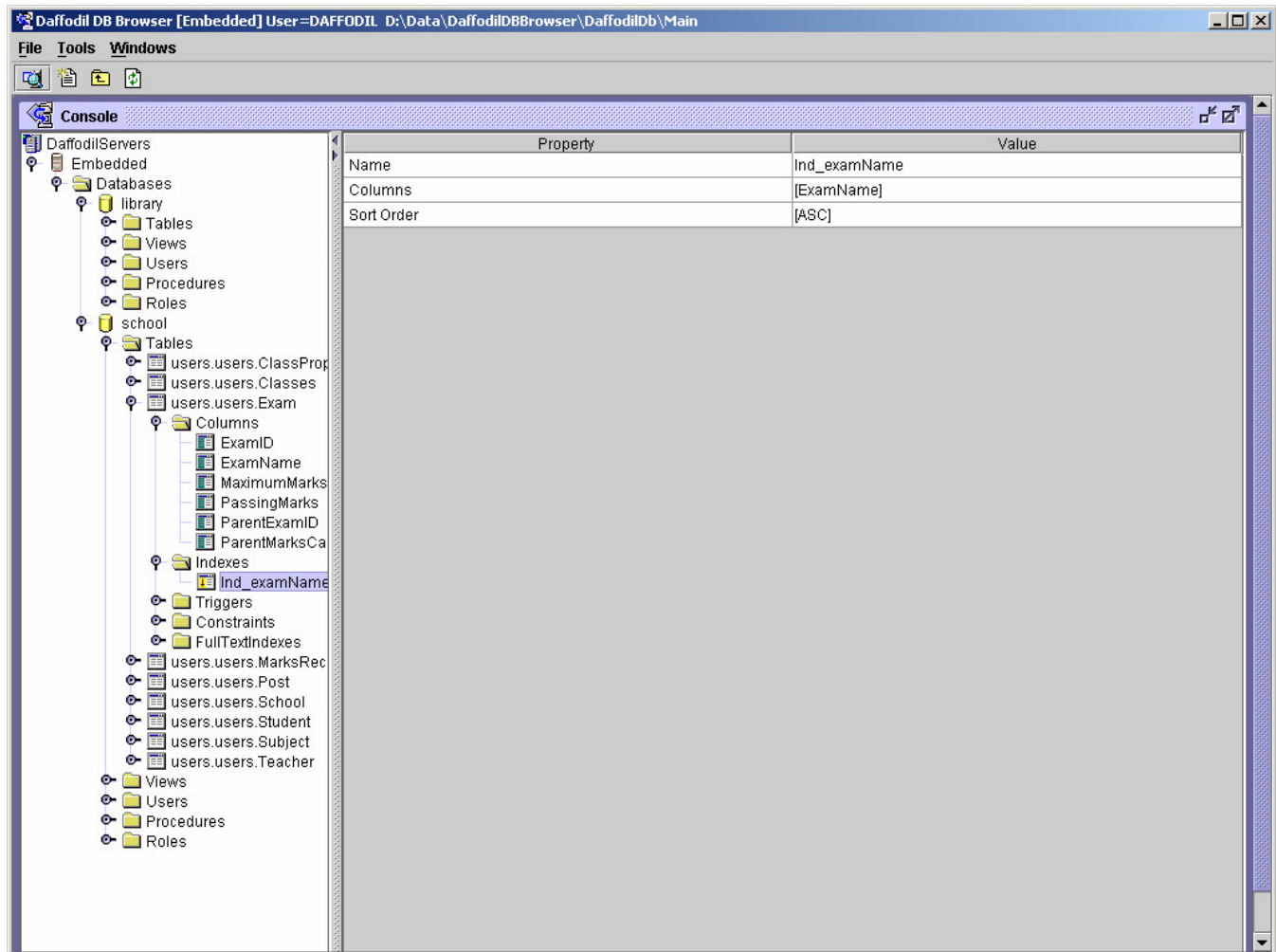
On right clicking the Indexes node, a pop up menu appears displaying options for making new indexes and refreshing the contents.

The Frame holds place for specifying column names of the table and its sort order. After filling up the Index Name, click the SQL button which fills the text area with the query generated to create the indexes according to the user inputs. Then click on “OK” button and a message will appear “Index created successfully”.



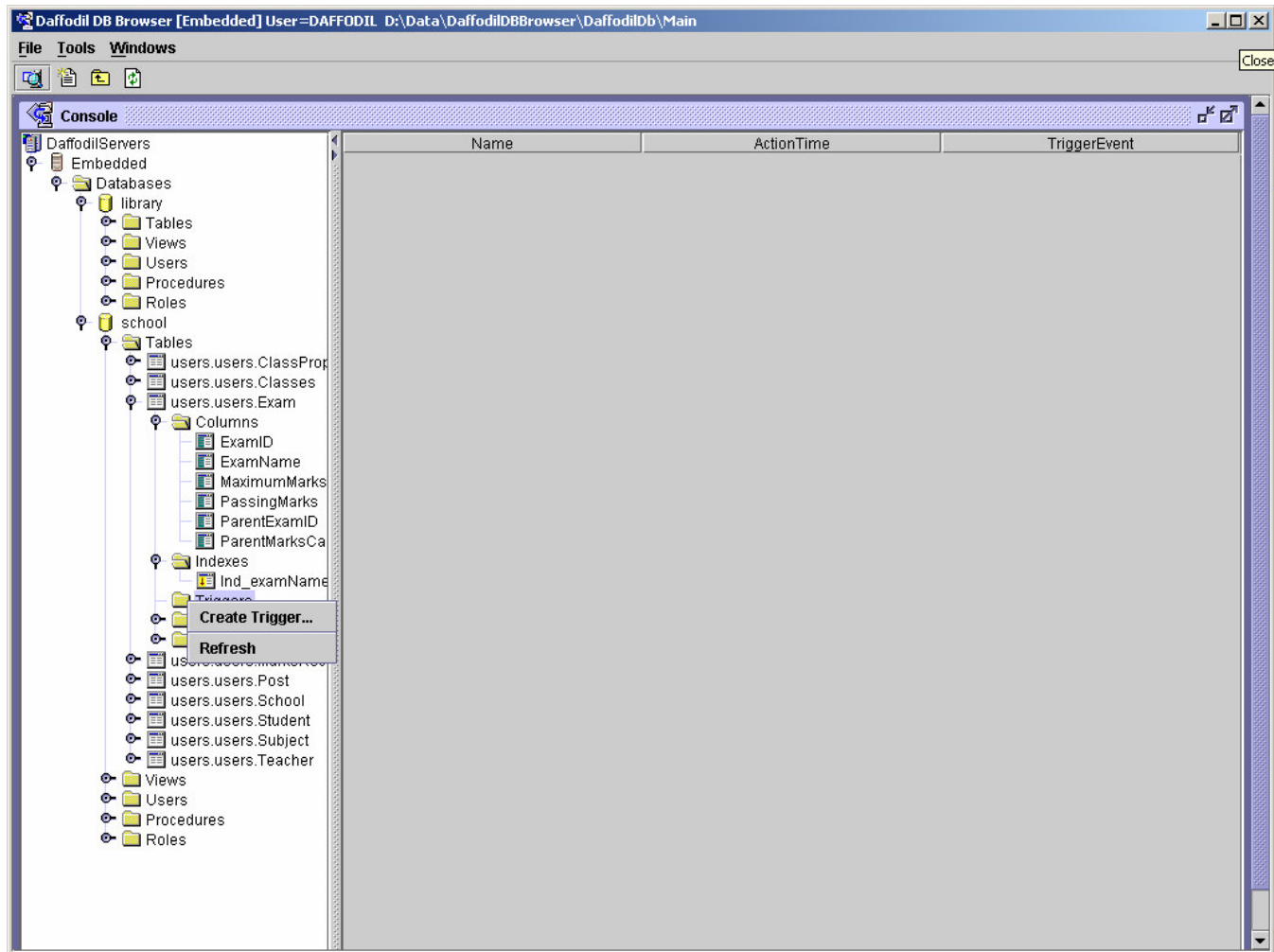
Index Name Node

It gives the properties and corresponding values of the index on the right pane of Daffodil DB Browser. These properties are Name, Columns, and Sort Order.



Triggers Nodes

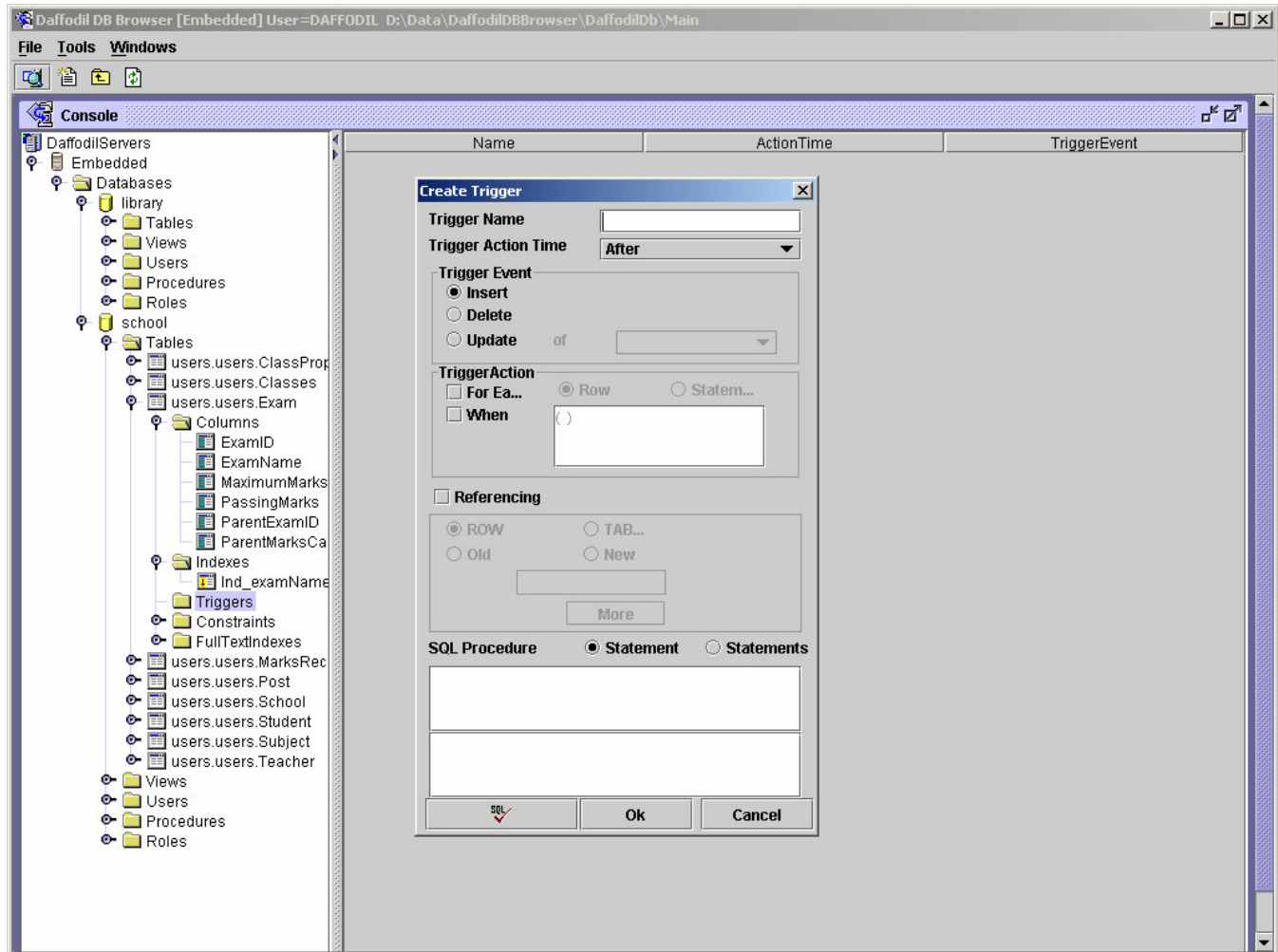
On right clicking Triggers node, a pop up menu appears displaying options for creating new triggers and refreshing the contents. New trigger is created in the existing database.

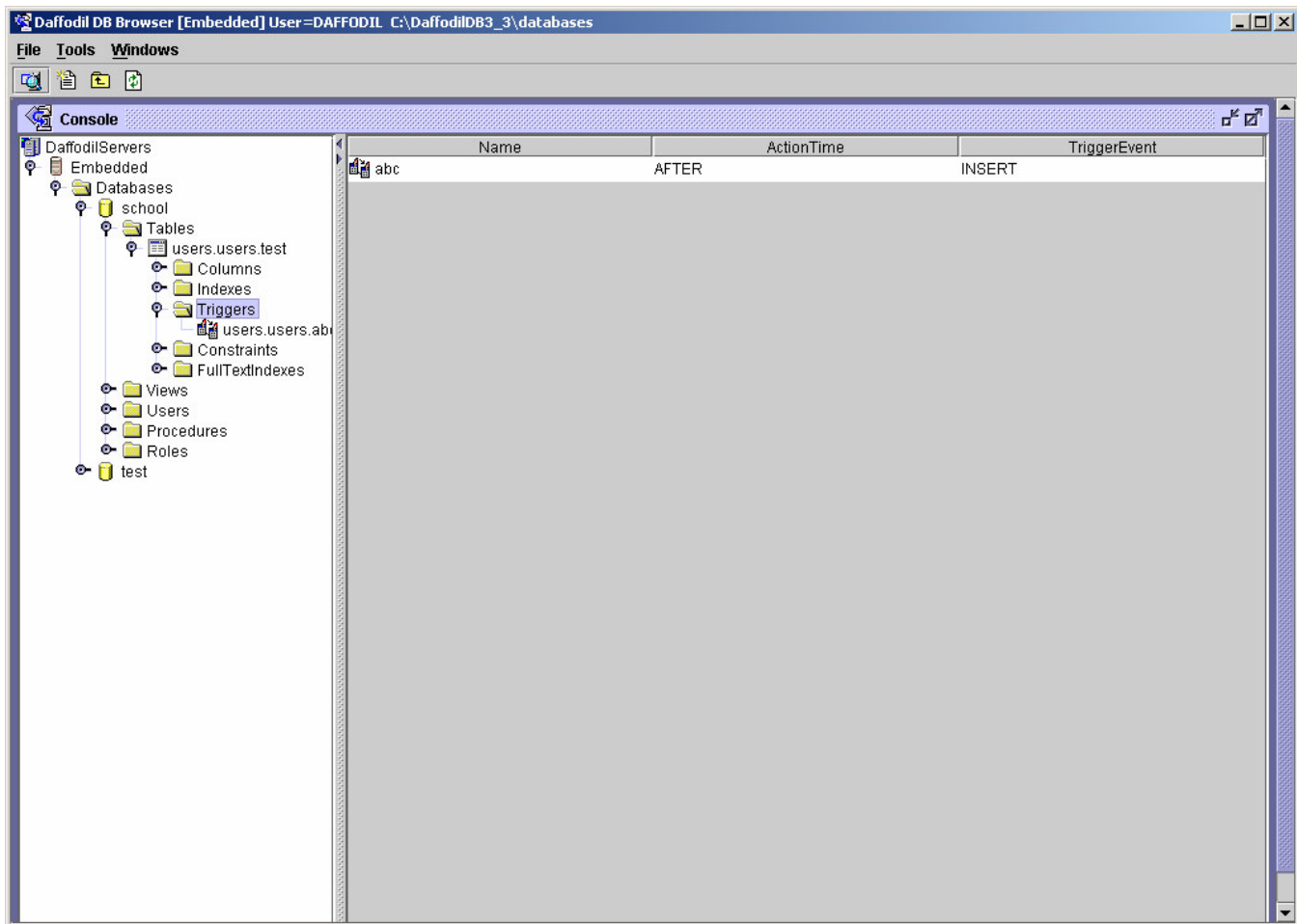


Create Trigger

It opens up the New Trigger Frame for creating a trigger.

The Frame holds place for all the trigger properties like name, action time, referencing, sql statements, event etc.

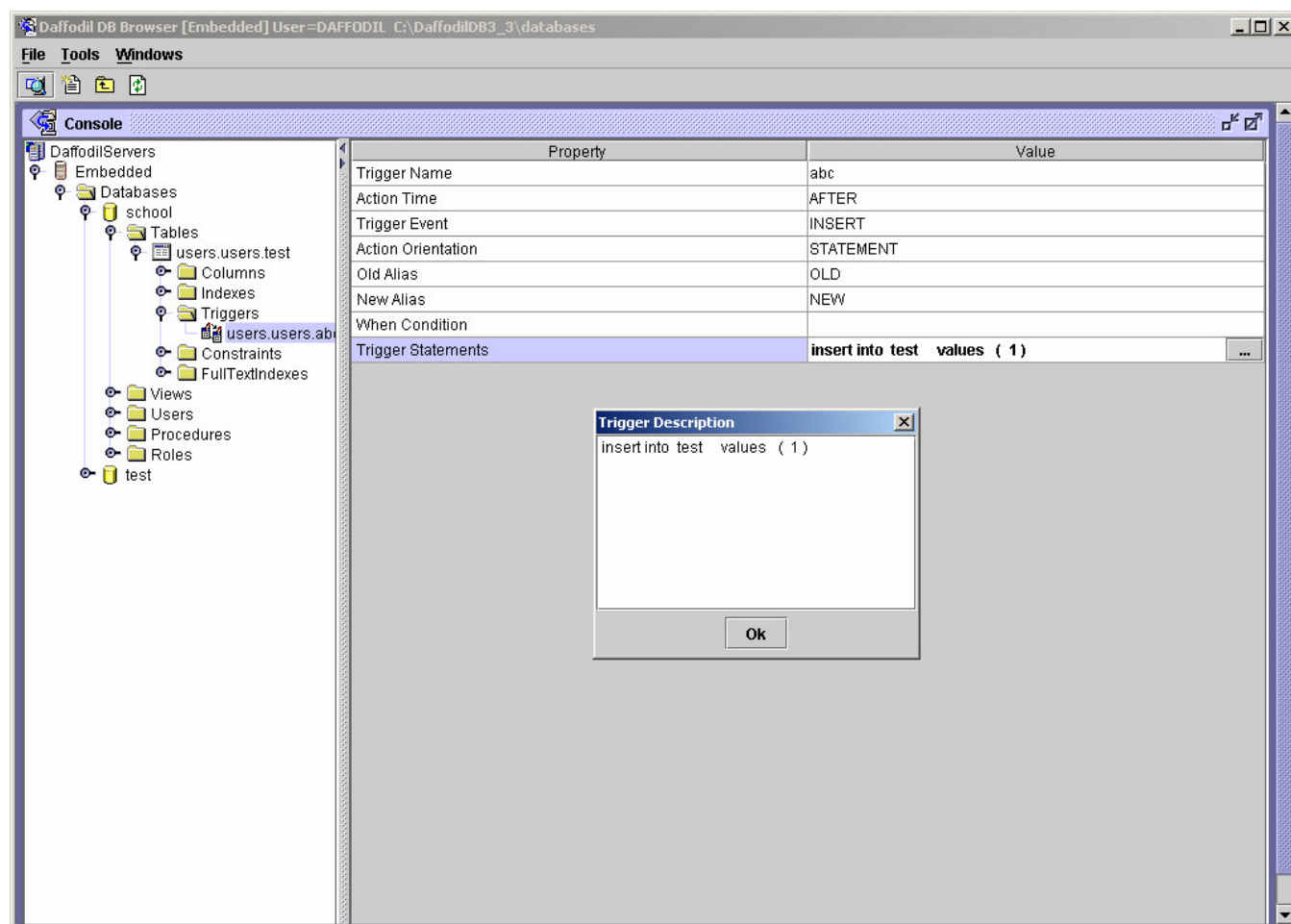




Trigger Name Node

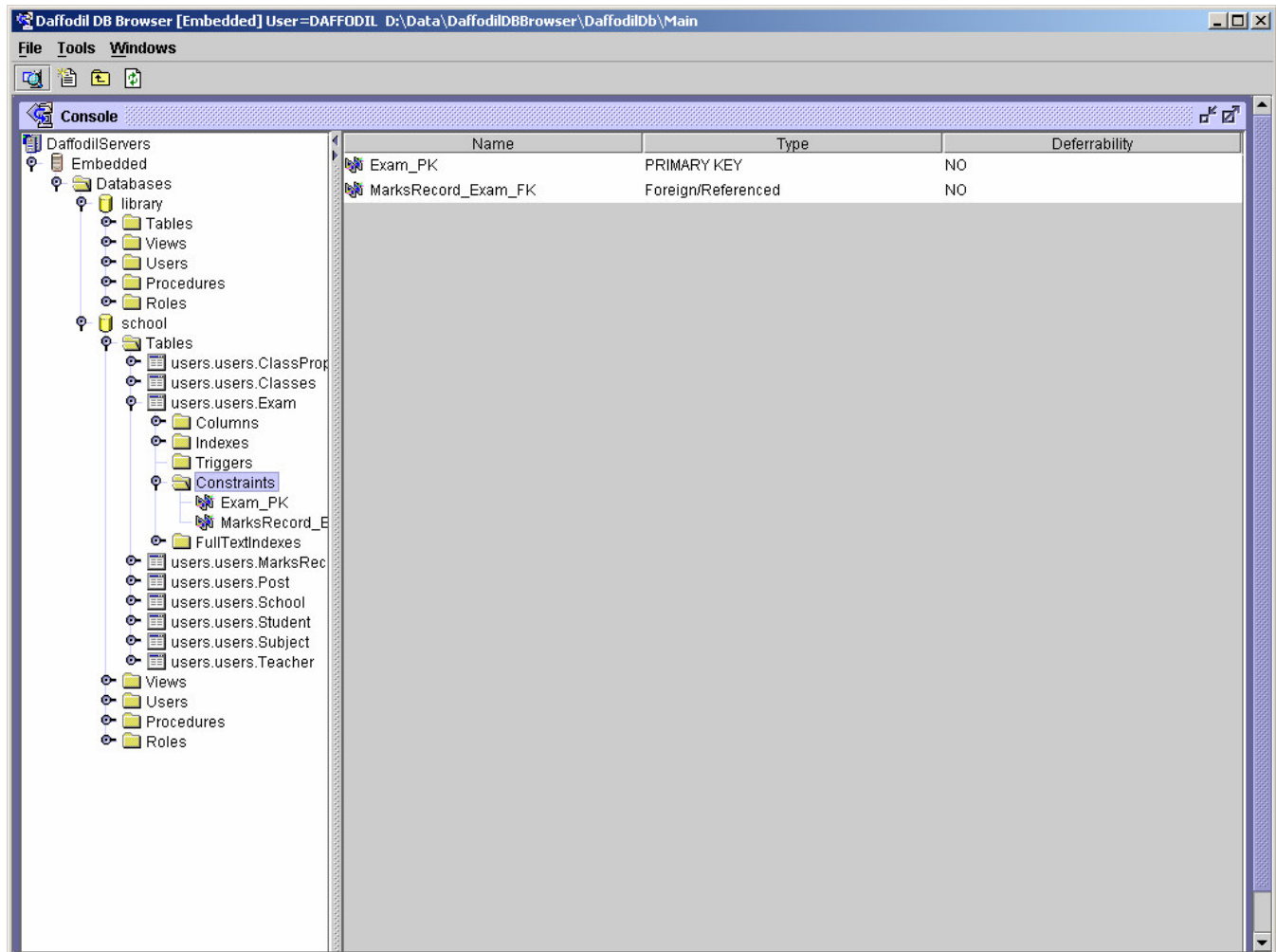
On clicking Trigger Name node, it displays a particular Trigger with its properties like Trigger Name, Action Time, Trigger Event, Action Orientation, Old Alias, New Alias, When Condition, and Trigger Statements on the right pane of the browser with its corresponding values.

On clicking the Value of trigger statement property, an Ellipse button will appear on the right side and on clicking the ellipse button, a dialog box will appear showing the trigger description.



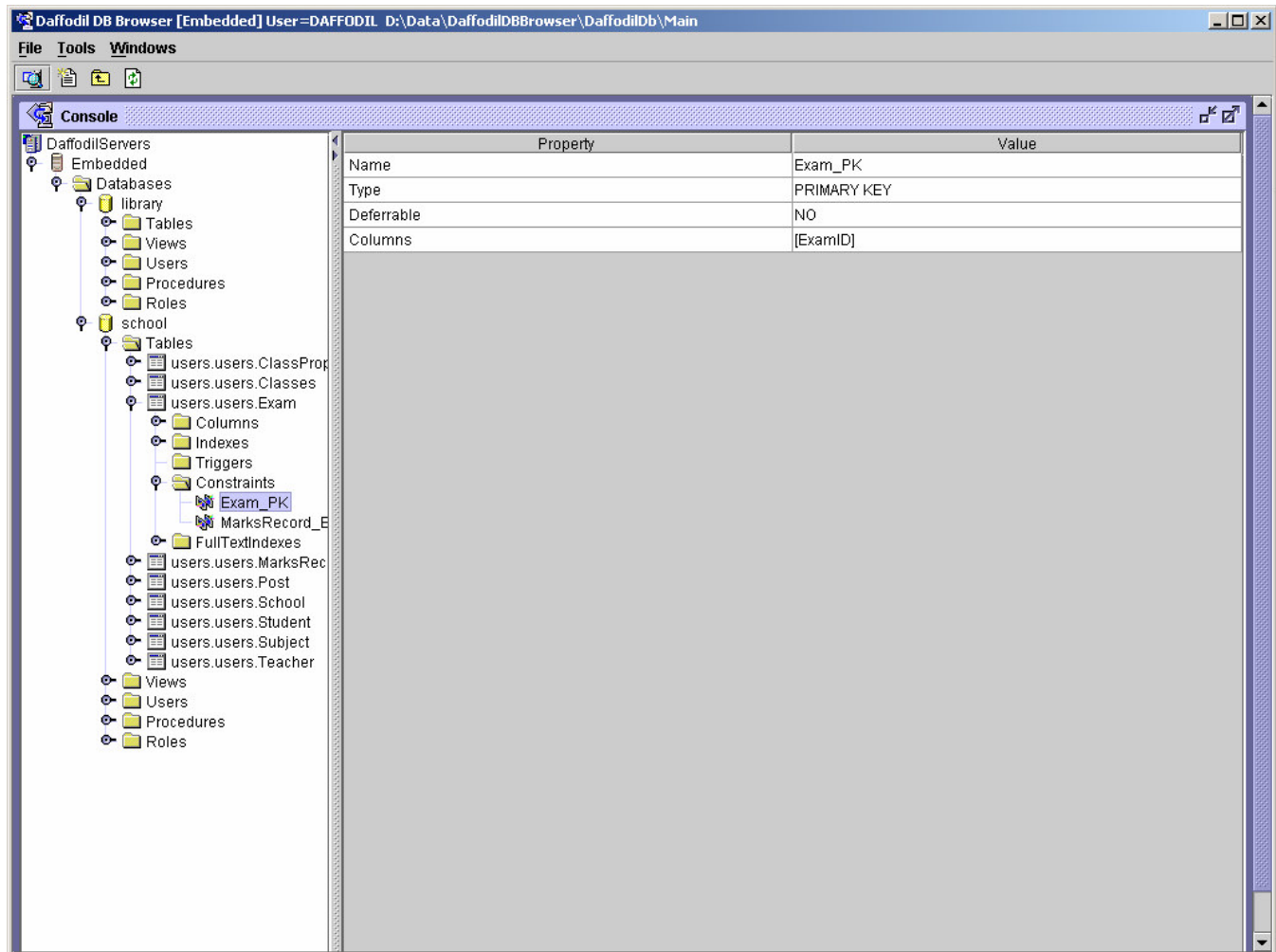
Constraints Node

On clicking constraints node, it gives all the constraints made on the table with their properties like Type and Deferability. Also, the node is expanded to display every constraint on the table in the left pane of the window.



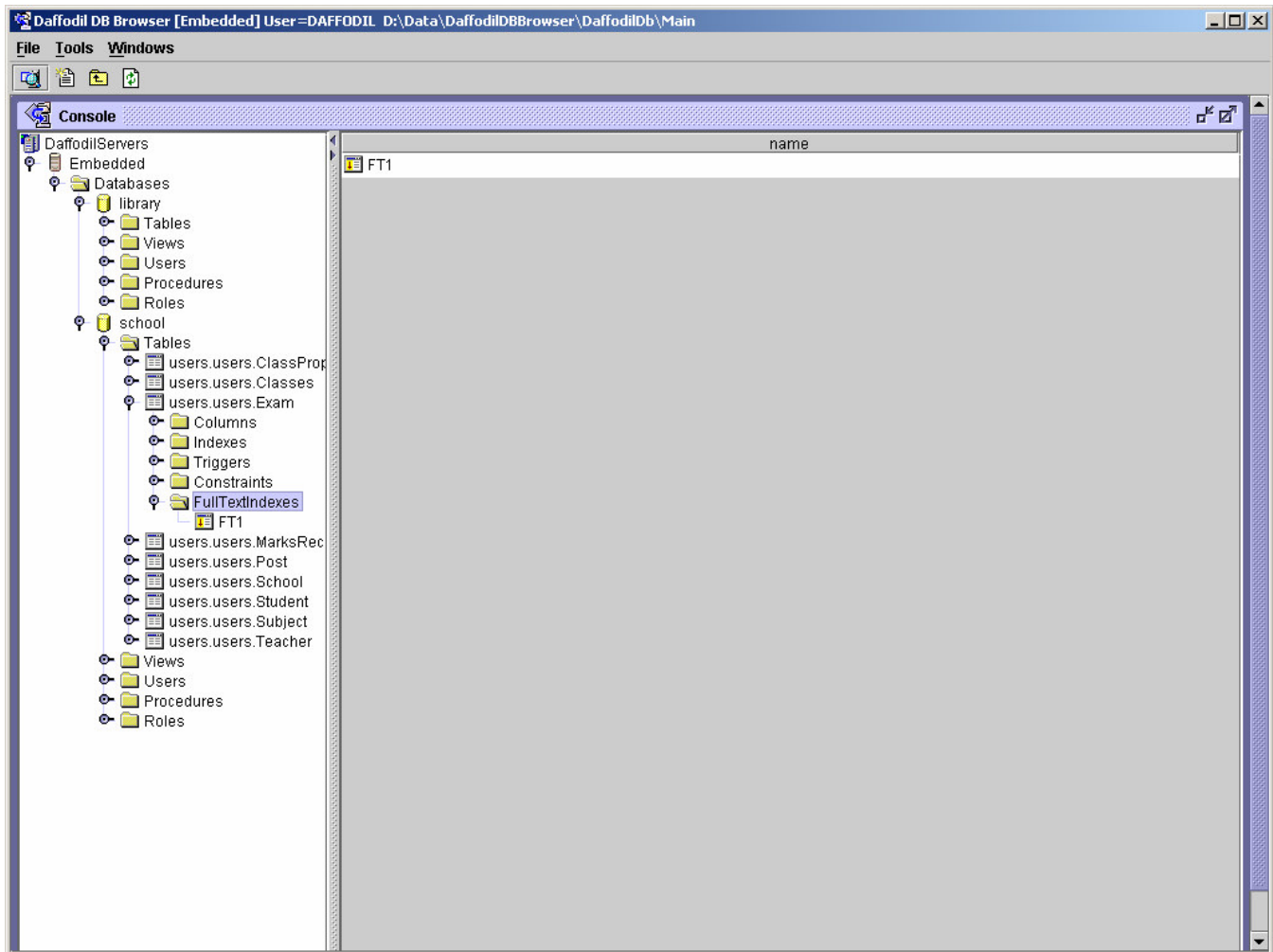
Constraint Name Node

On clicking the Constraint Name node, it displays a particular constraint with its properties and values. The properties specified are constraint name, constraint type, deferrable and columns.



FullTextIndexes Node*

On clicking the FullTextIndexes node, it gives all the FullText indexes made on that table on the right panel of Daffodil DB Browser. Also, the node is expanded to display all FullText indexes on the table on the left pane of the window.

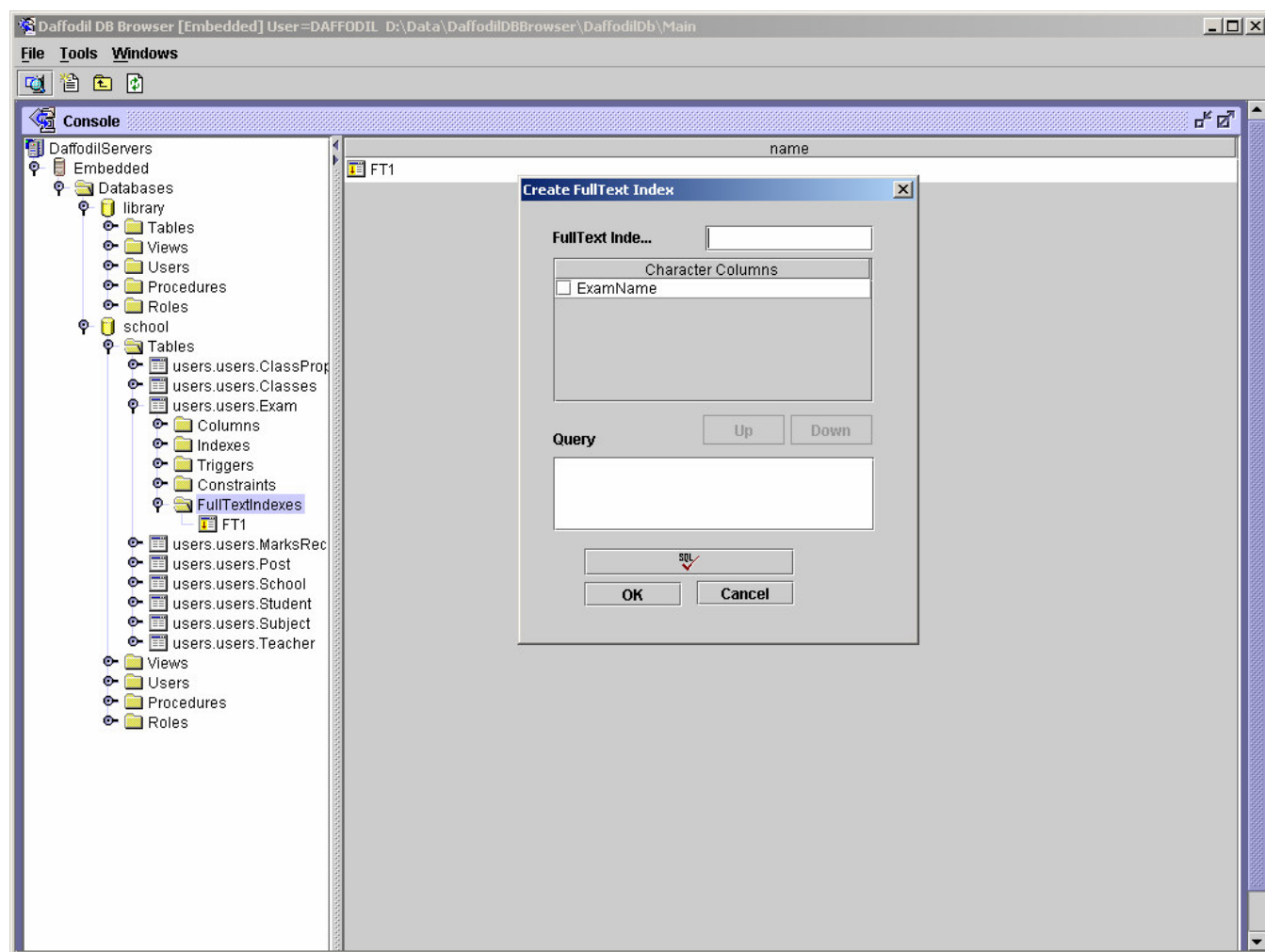


* Features that are not supported in One\$DB

Create FullText Index *

On right clicking the Indexes node, a pop up menu appears displaying options for making new indexes and refreshing the contents.

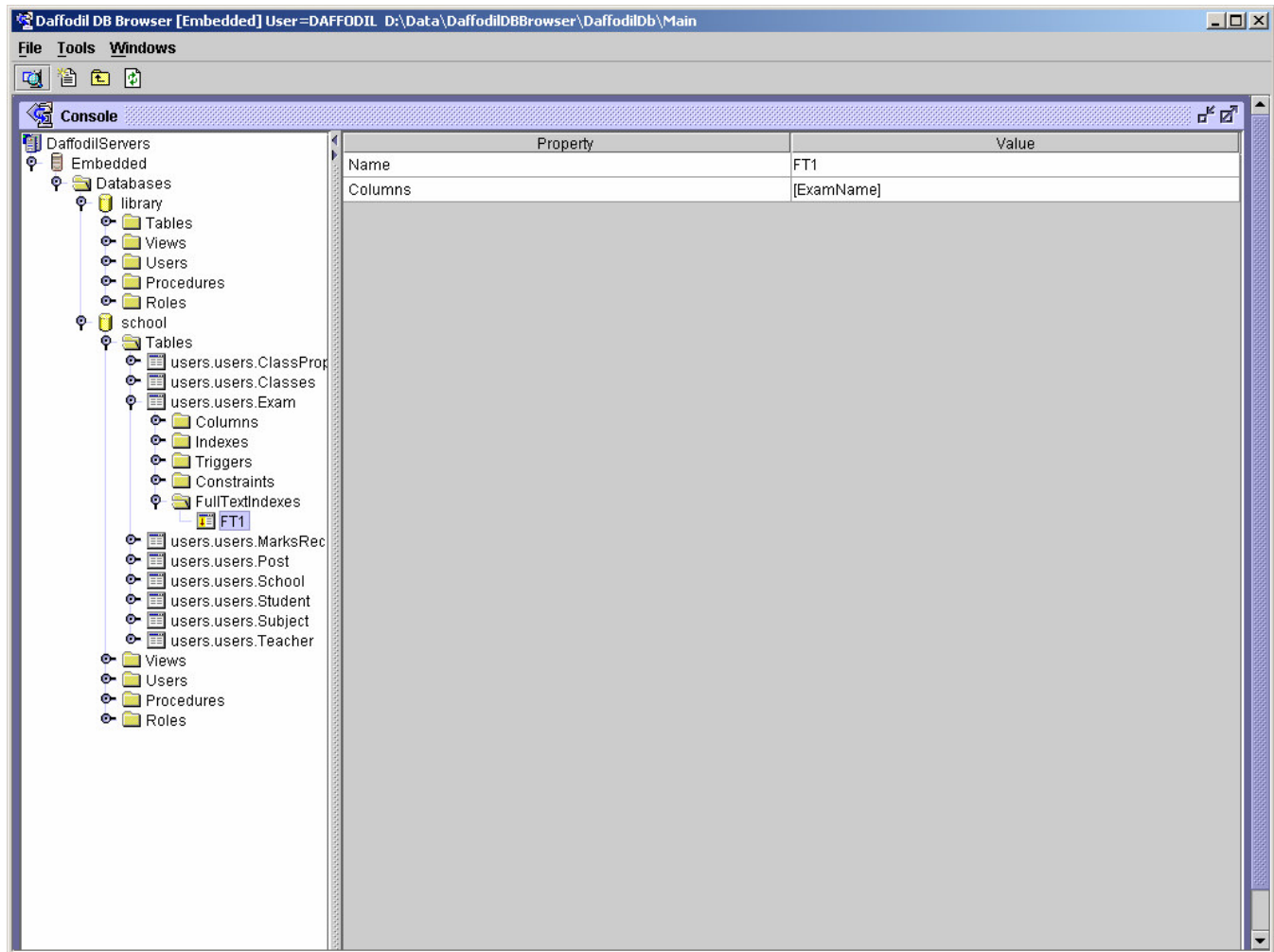
The Frame holds place for specifying FullTextIndex Name and columns (Character Type) of the table. Clicking the SQL button fills text area with the query generated according to the user inputs. Now click the “OK” button and a message will appear “FullTextIndex Created Successfully”



* Features that are not supported in One\$DB

FullTextIndex Name Node *

It gives the properties and corresponding values of the index on the right pane of Daffodil DB Browser. These properties are Name and Columns.



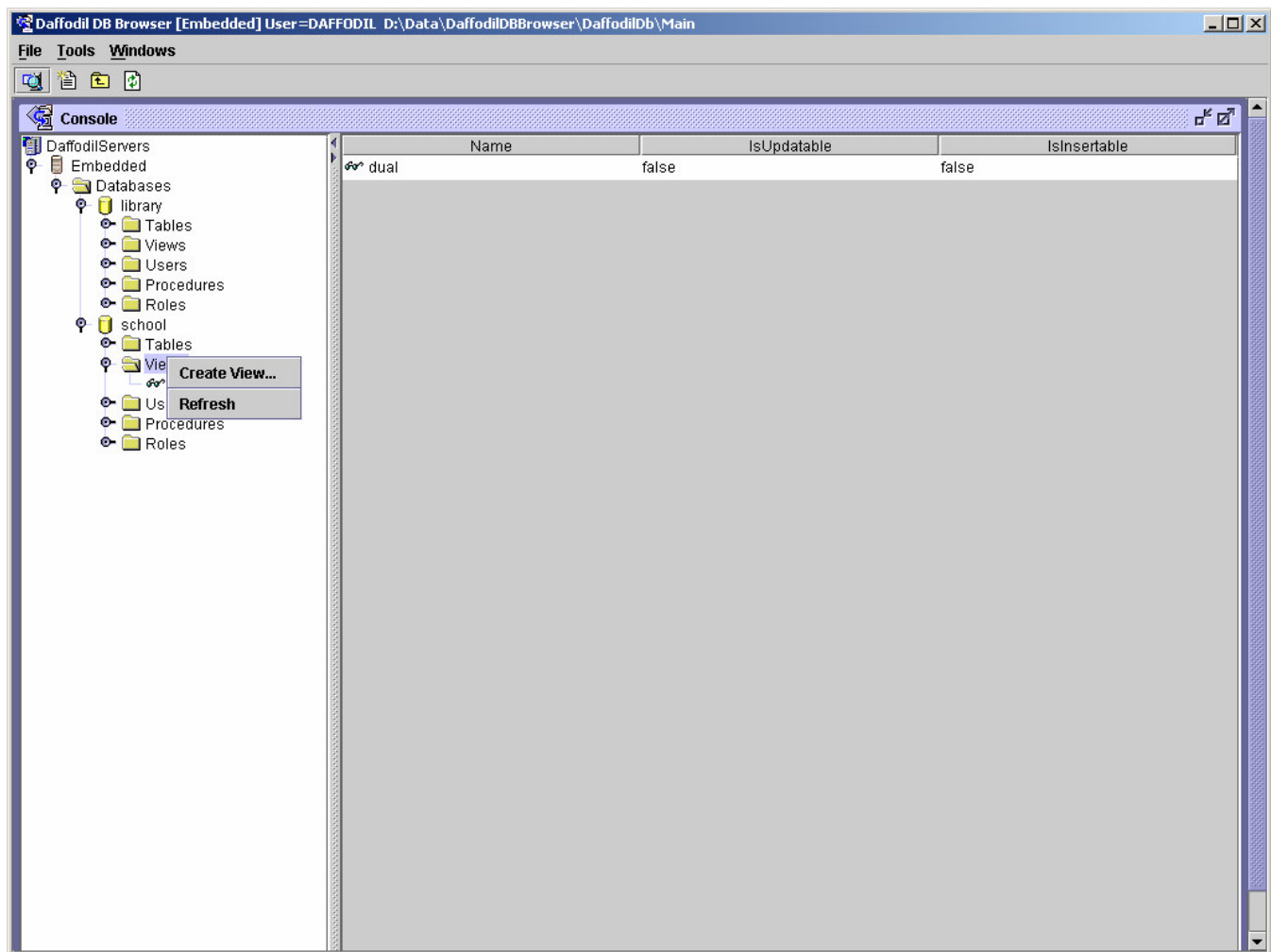
On **right clicking** the FullTextIndex Name Node, you will get options for Dropping the FullTextIndex and Refreshing the contents.

* Features that are not supported in One\$DB

Views Node

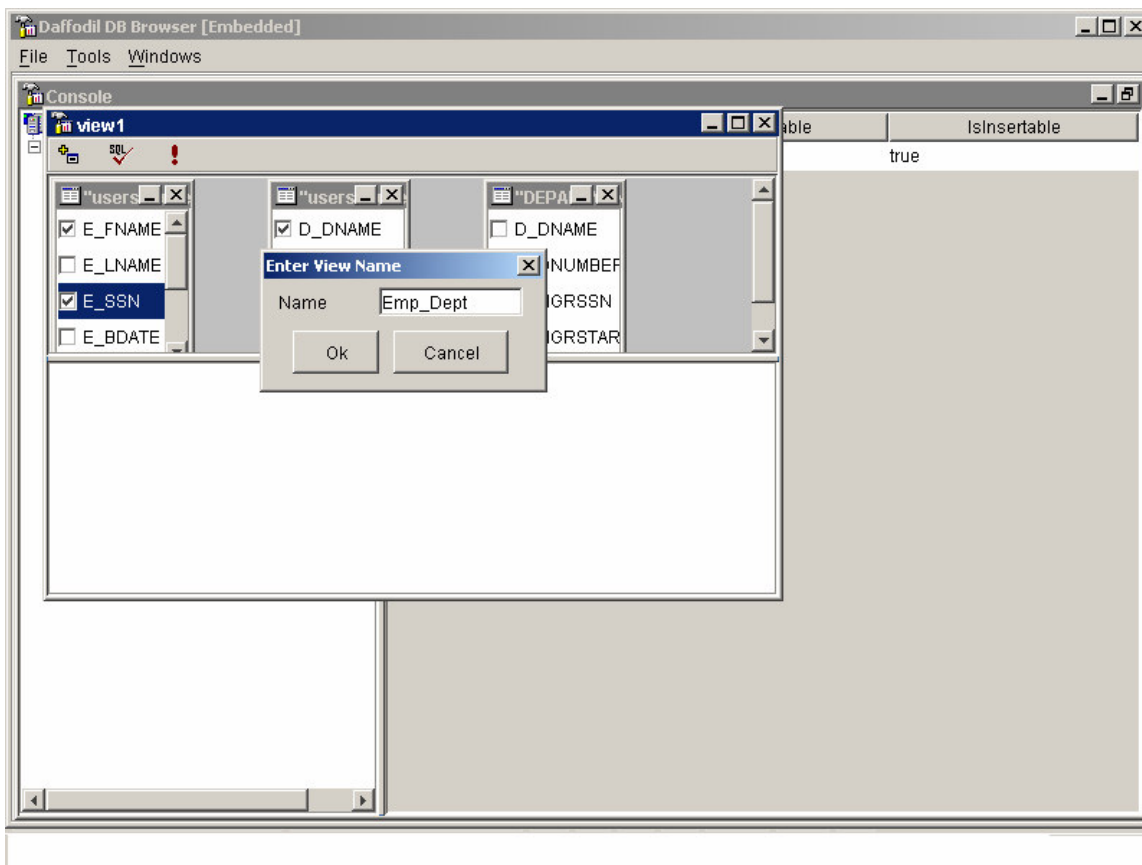
On right clicking Views node, a pop up menu appears displaying options for creating a new View and refreshing the contents. This new View gets created in the existing database.

On clicking the Views Node, it gives all the Views created by users and its properties like IsInsertable and IsUpdatable made on that table on the right pane of Daffodil DB Browser. The Views are also displayed on the left panel of the browser as its child nodes.



Create View

It opens up a dialog box asking for the View name and then opens the New View Frame for creating the View.



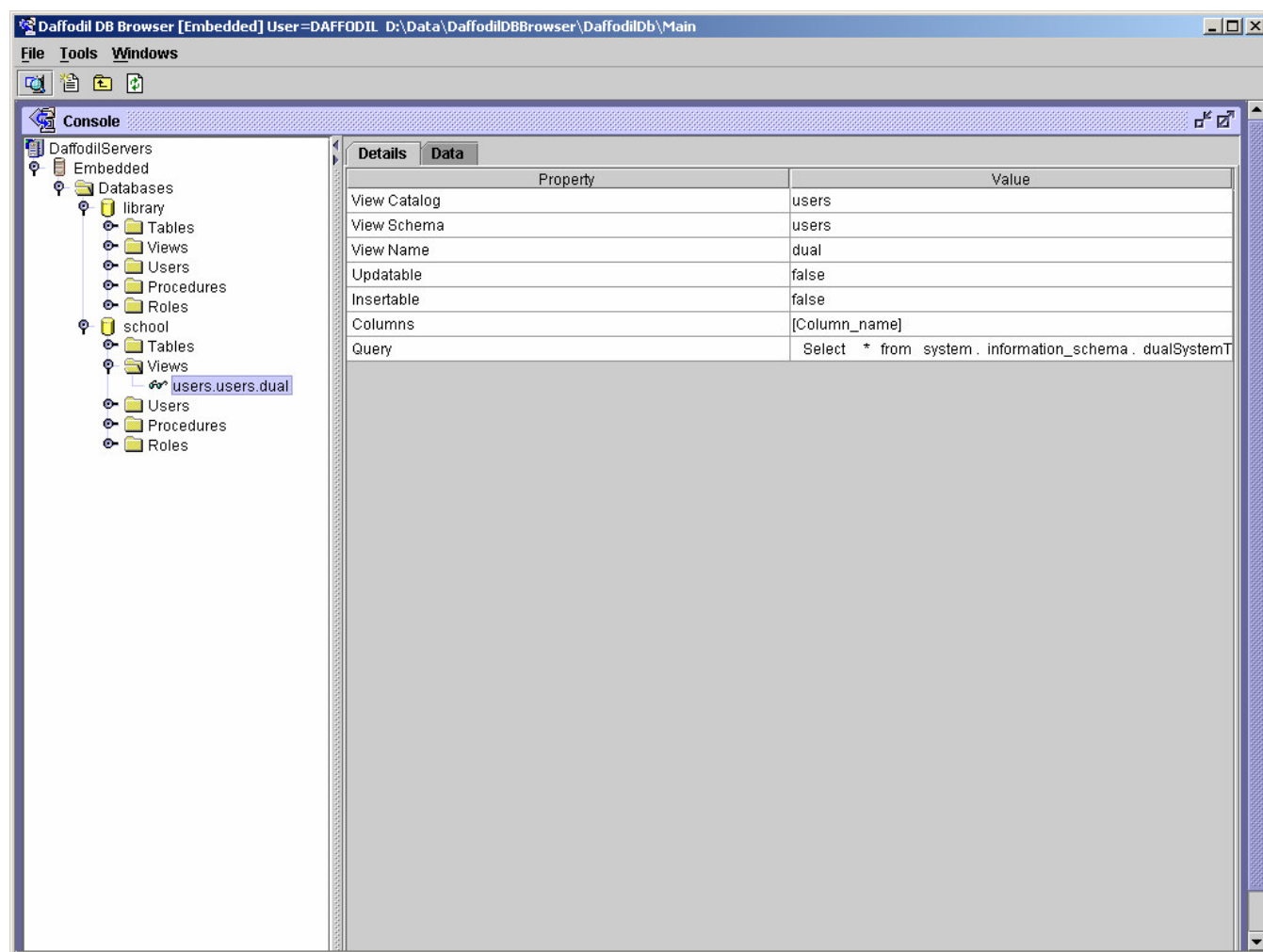
View Name Node

On clicking the view node, it displays a particular View with its properties like Catalog, Schema, Name, IsUpdatable, IsInsertable, AllColumns, and Query on the right pane of the browser.

On clicking any View Name node, a tabbed pane is displayed on the right pane of Daffodil DB Browser having two tabs named “Details” and “Data”.

By selecting the **Details** tab, it displays a particular View with its properties and values like Catalog, Schema, Name, Updatable, Insertable, Columns, and Query on the right pane of the browser.

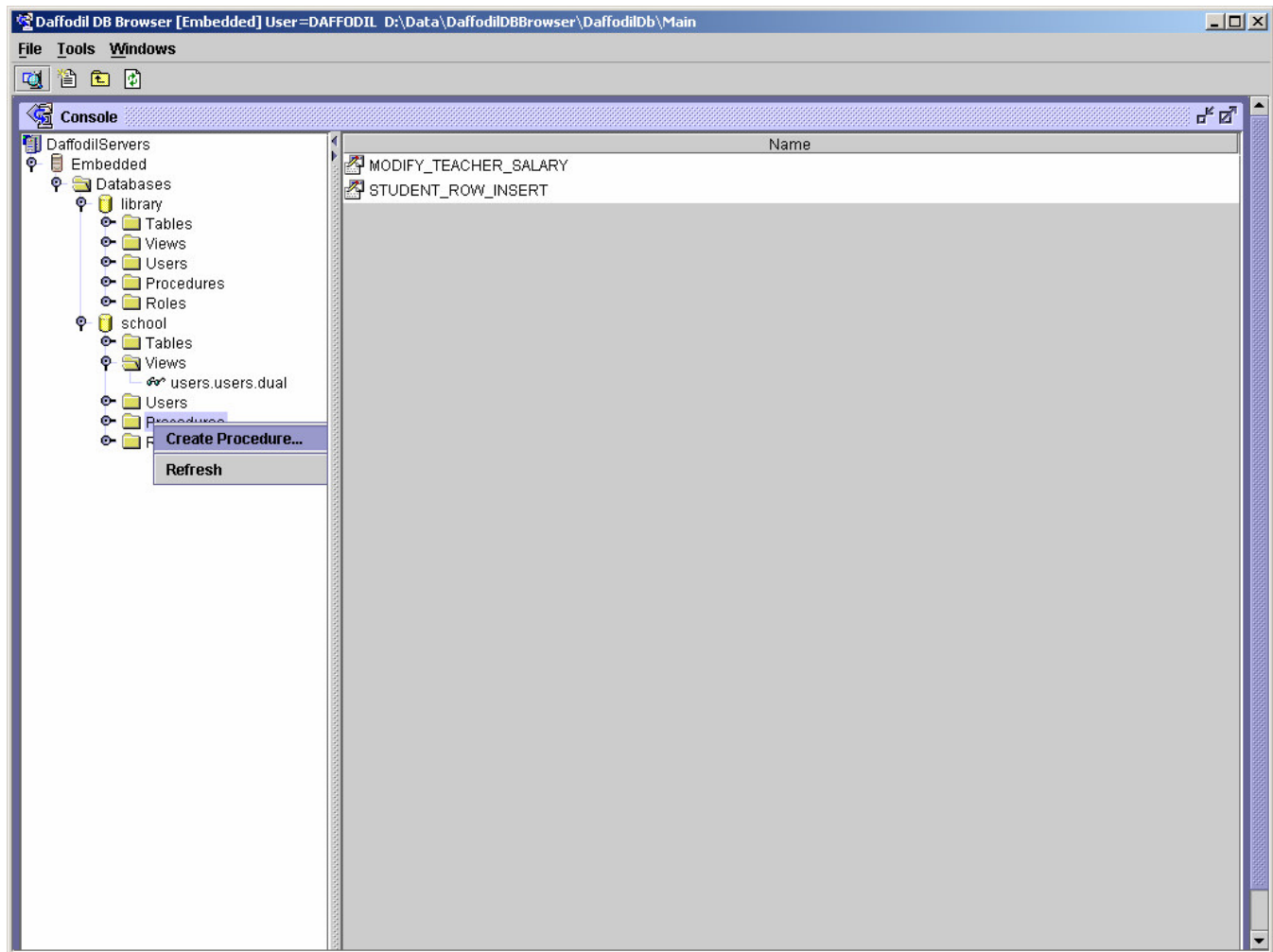
By selecting the **Data** tab, the details about the data entered are displayed.



Procedures Node

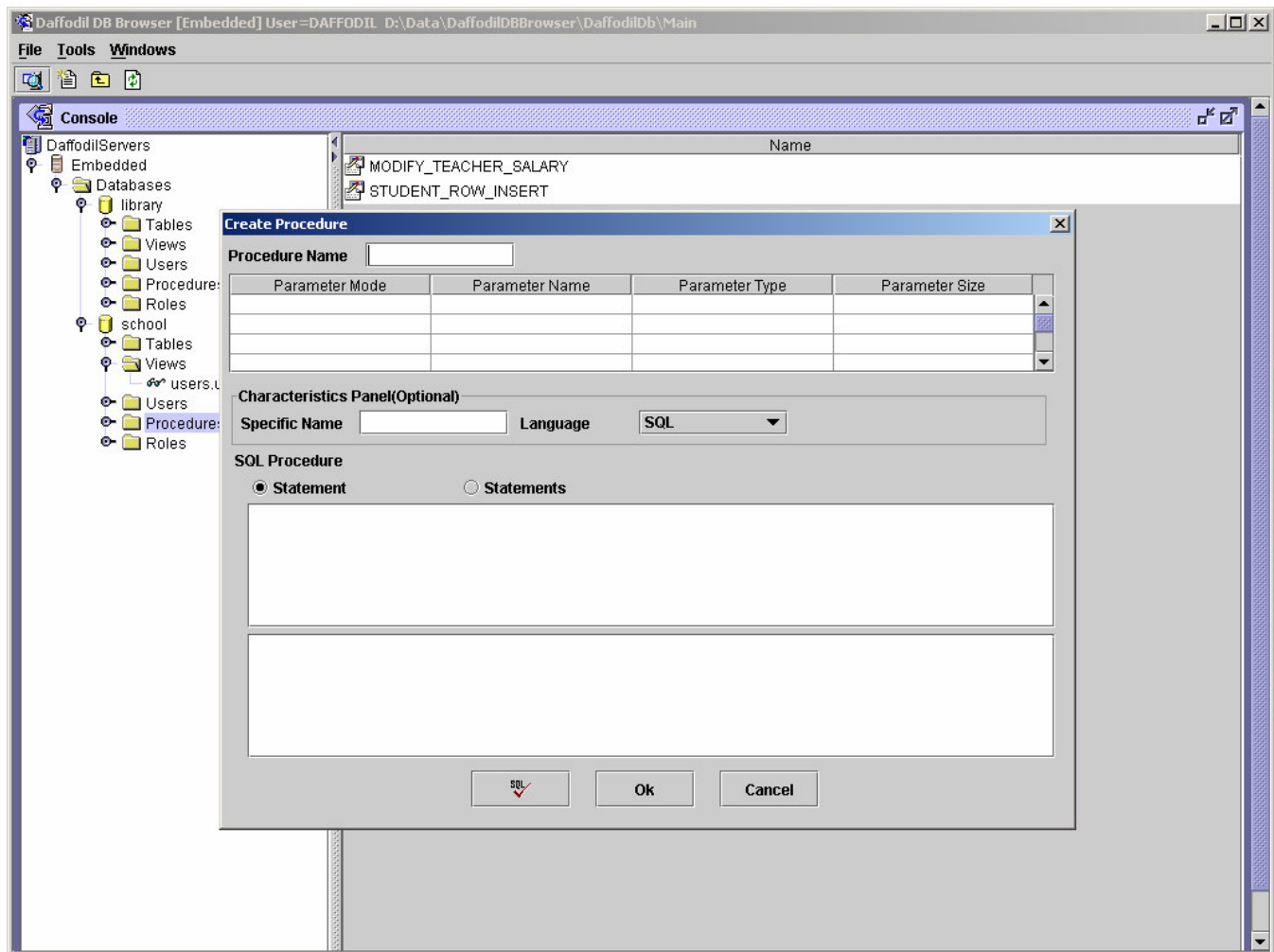
On clicking the Procedures Node, it gives all the procedures created by users on the right pane of Daffodil DB Browser. The procedures also get displayed on the left pane of the Browser as its child nodes.

On **right clicking** the Procedures node, a pop up menu appears displaying options for creating a new procedure and refreshing the contents. The new procedure is created in the existing database.



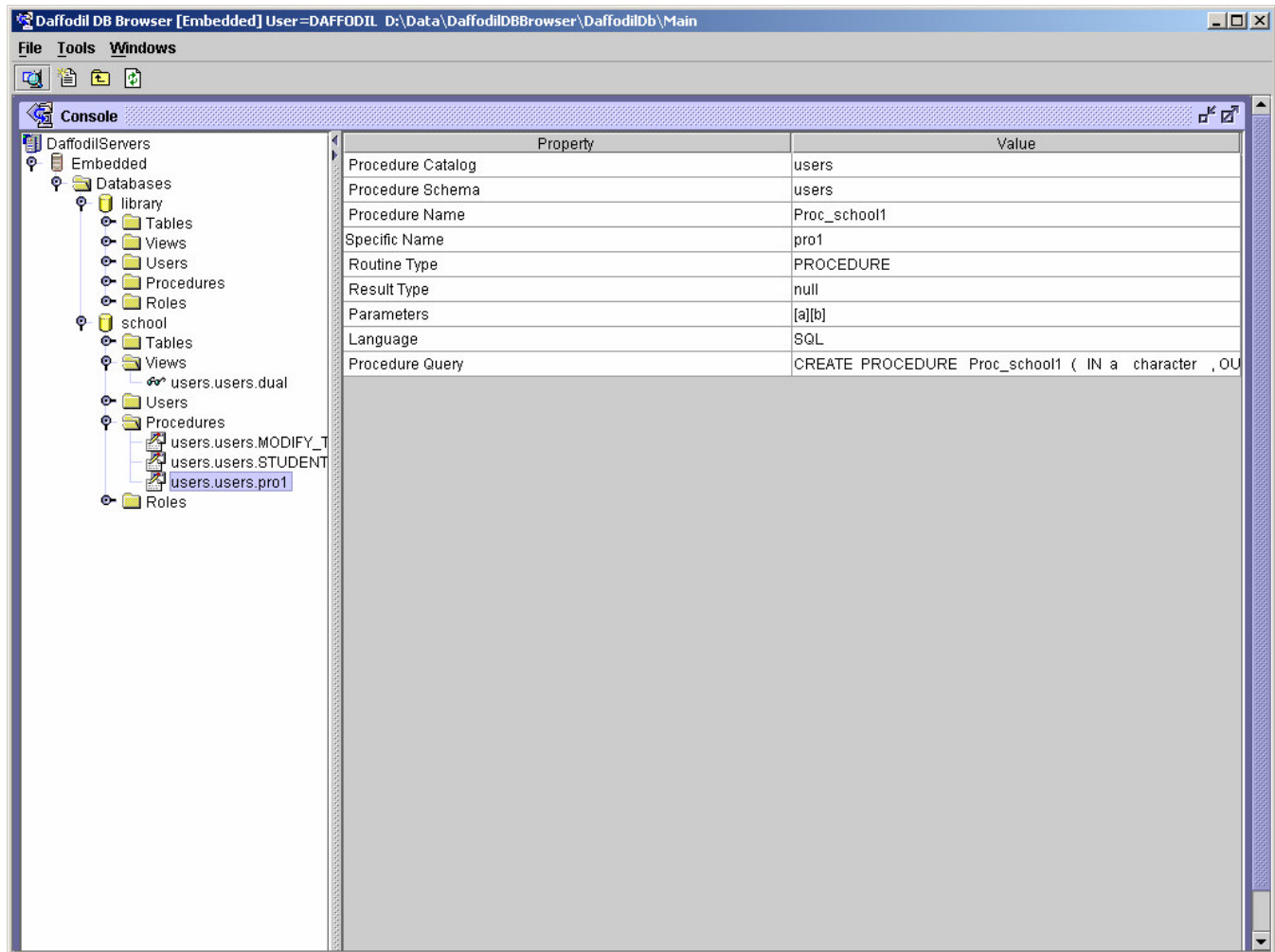
Create Procedure

It opens up a Procedure Frame for creating a new procedure.



Procedure Name Node

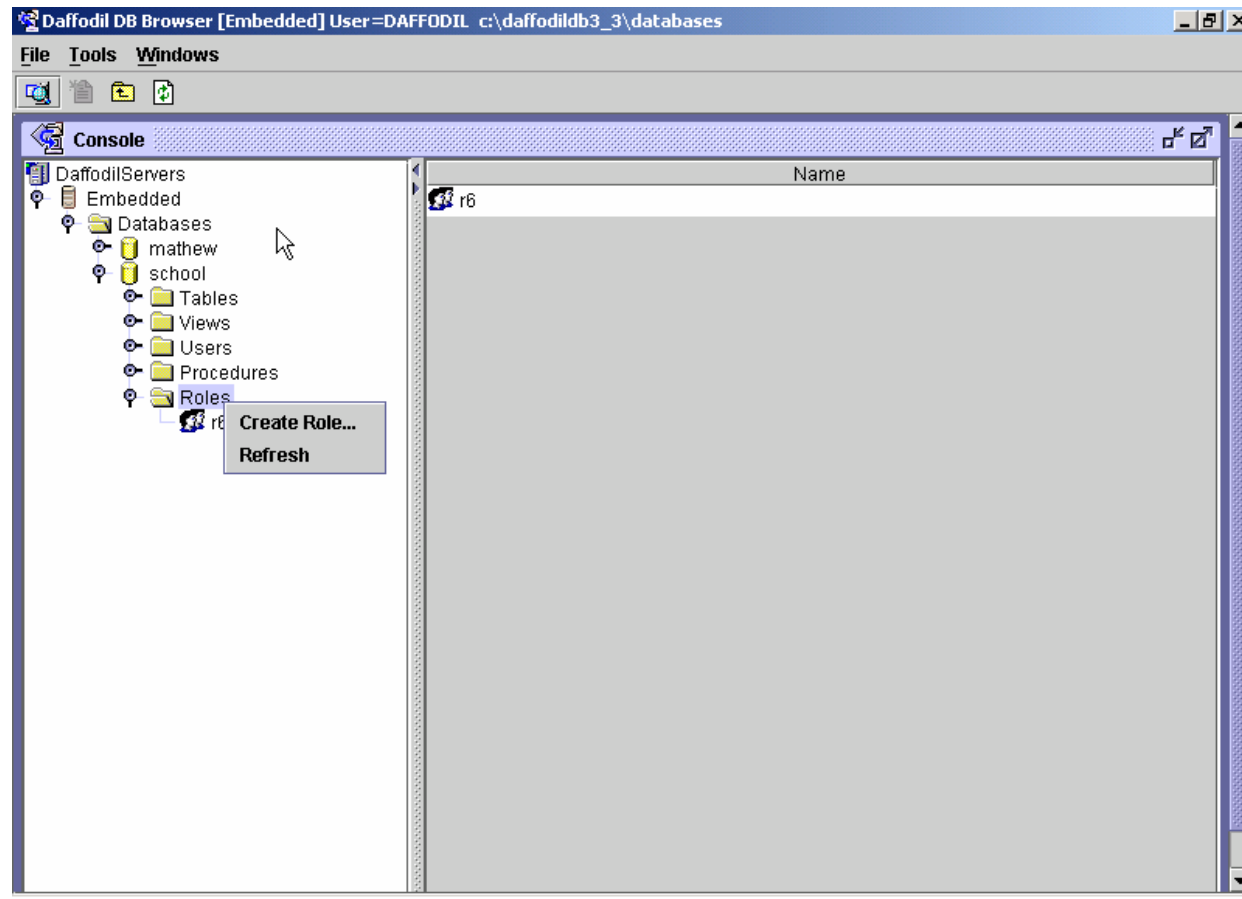
On clicking the Procedure Name node, it displays a particular procedure with its properties like Catalog, Schema, Name, SpecificName, RoutineType, ResultType, Parameters, Language, and ProcedureQuery, in the right pane of the Browser.



On **right clicking** the Procedure Name Node, you will get options for Dropping the Procedures and Refreshing the contents.

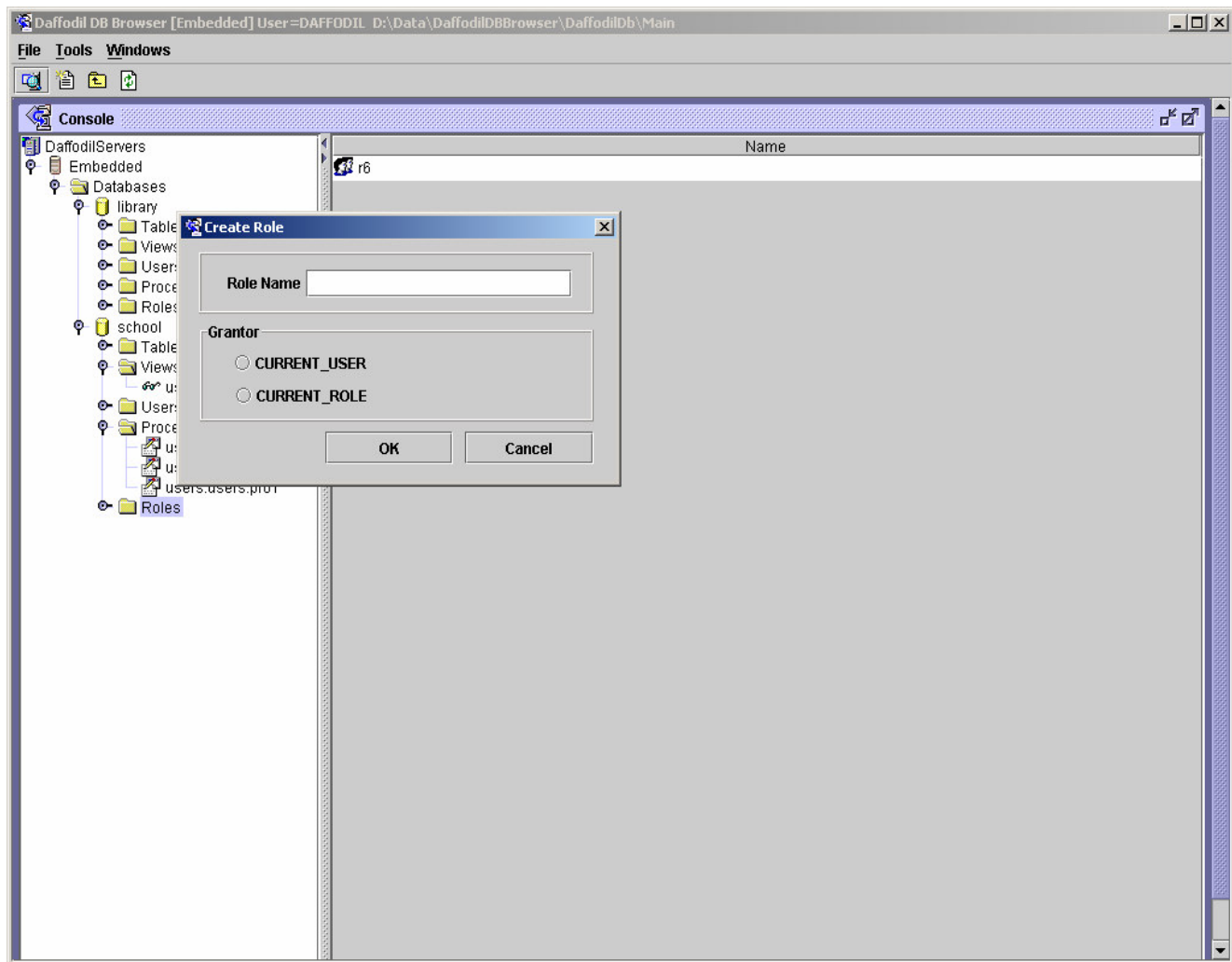
Roles Node

On right clicking the Roles node, a pop up menu appears displaying options for making new Role and refreshing the contents. The new Role is created in the existing database.



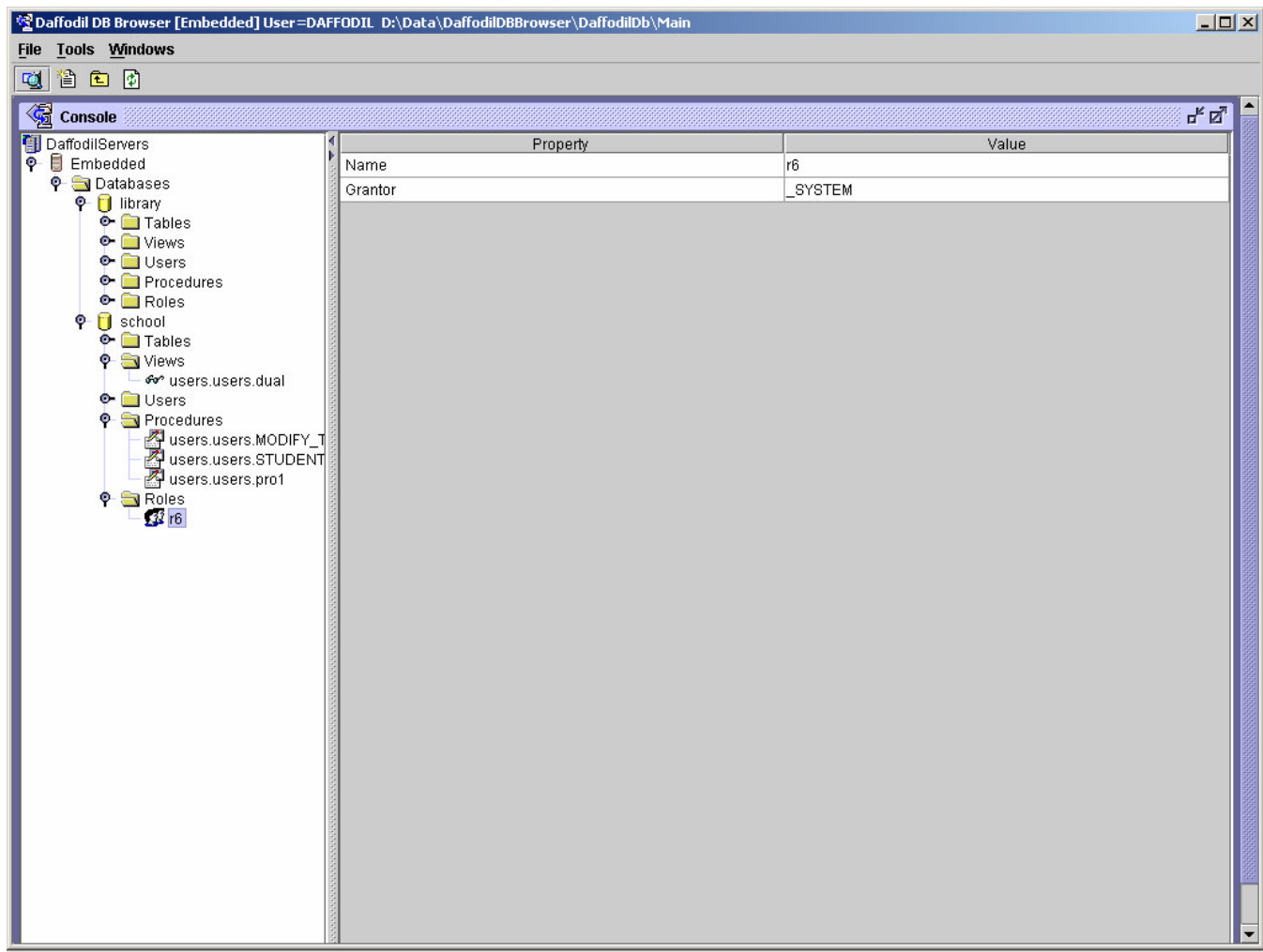
New Role

It opens up a Create Role Frame for creating a new Role



Role Name Node

On clicking the Role Name node, it displays a particular Role node with its Grantor.



On right clicking the Roll Name node, you can Drop the roles.

Query Analyzer Mode

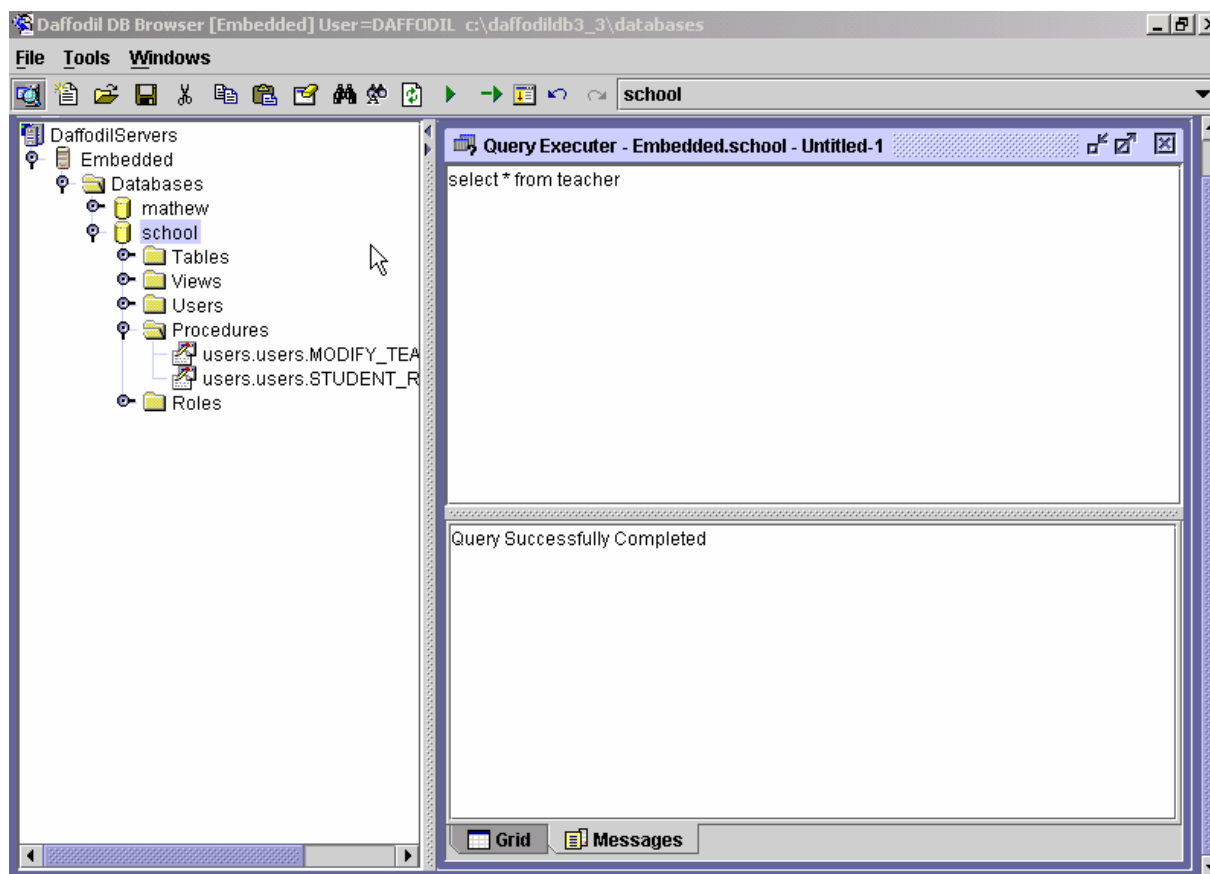
Query Analyzer Mode is a graphical user interface for testing SQL statements and scripts interactively. Query Executer Pad is used to execute the queries. Executer pad is invoked by selecting Query Analyzer Mode button in the toolbar of Daffodil DB Browser. Executer pad is displayed in the right pane of the window.

The query executer window is made up of Editor pane and Result pane. The *Editor pane* is a text editor for passing SQL statements. The result of execution of the query is shown on the *result pane* which in turn comprises of Grid tab, Message tab and Procedure tab. The *Grid tab* displays returned result sets. The *Messages tab* displays successful messages as well as error messages. The Procedure tab displays the statement wise output of a procedure.

- The results of DML and DDL Operations, except procedure statements are shown on the message tab pane of the result pane.
- The result of DQL Operations is shown on the grid tab of the result pane.
- The result of Procedure statement is shown in procedure tab of the result pane.

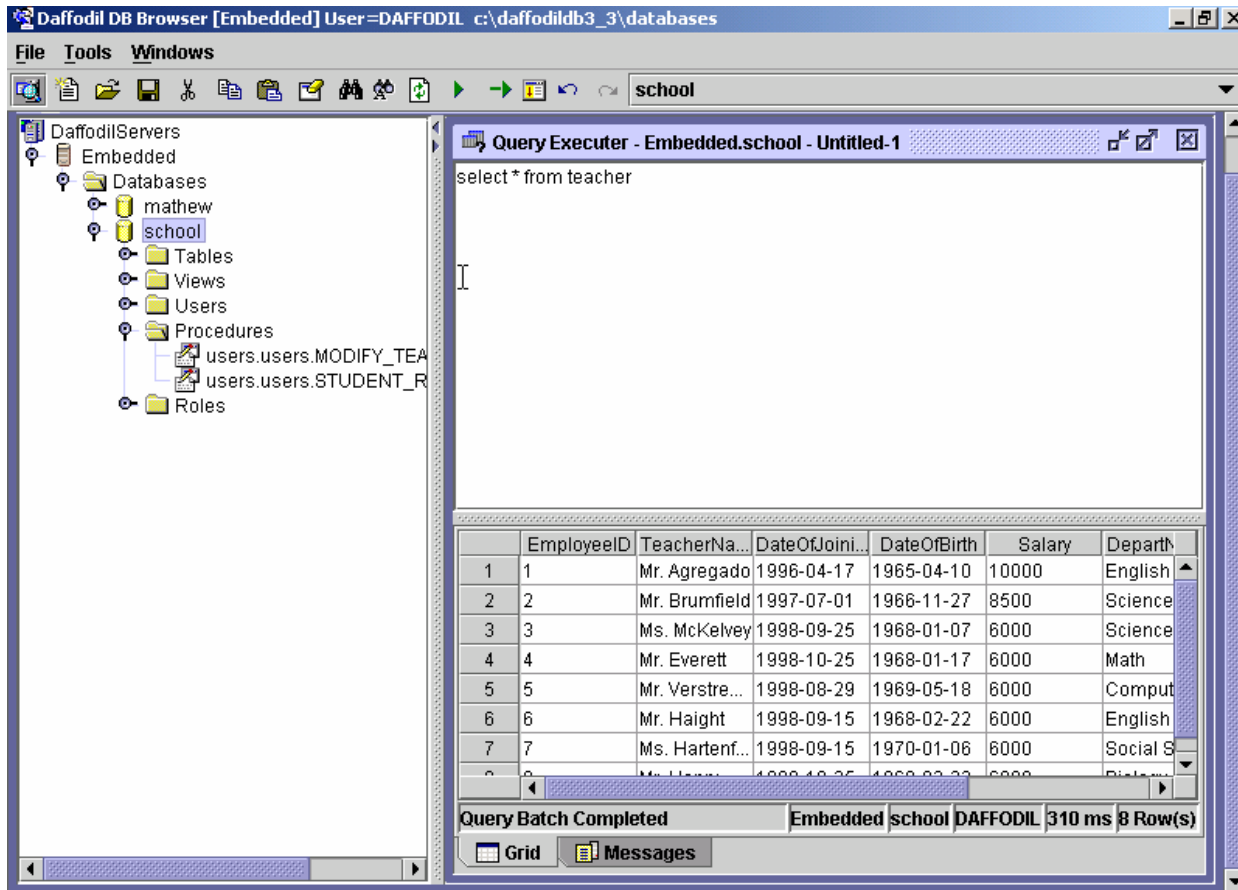
Message Tab

The message tab displays the result of executing the create statements, insert, update or delete queries. A single message tab is opened after a single query execution. If a user selects multiple queries for execution, then multiple message tabs will be displayed.



Grid Tab

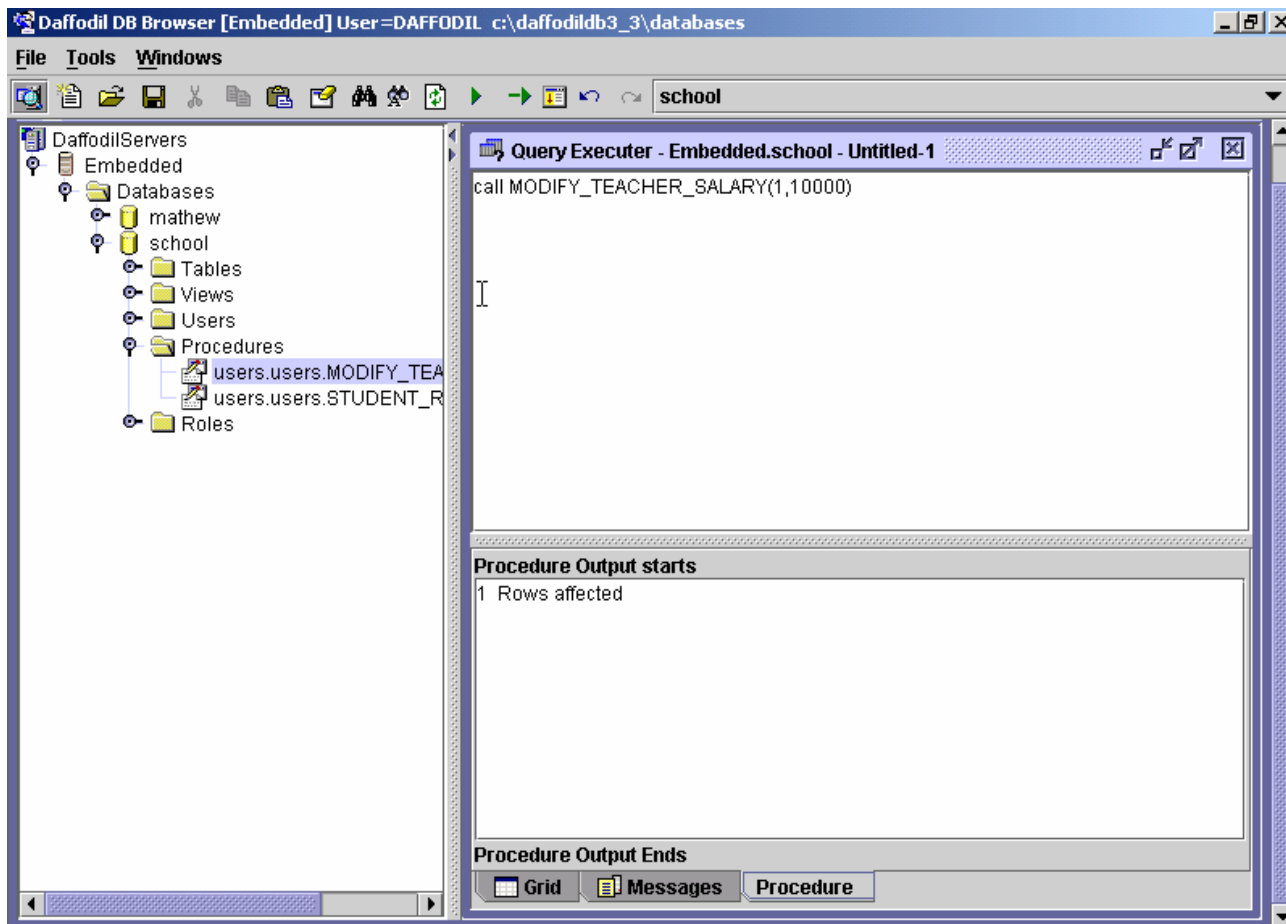
A grid tab displays the result of executing select query(s). It displays the result set of retrieved records. A single grid tab is opened for multiple queries. In case of multiple queries, output is separated by split panes.



Procedure Tab

The *Procedure tab* displays the statement wise output (separated by split panes) of a procedure.

Note: The Procedure tab will appear only if we call a procedure.



Possible Tasks in the Query Analyzer Mode: -

- Execute all types of Queries supported by Daffodil DB.
- Save Queries written in Query Executer Pad in a file.
- Load Queries saved in files.
- Work using Multiple Query Executer Pads.
- Run Multiple Sessions against a Database.
- Execute Parameterized Queries.

Execute Queries in Query Analyzer Mode

In this mode, queries are written on a text pad called Query Executer Pad (QEP). A number of QEPs can be opened for query execution. Daffodil DB Browser can manage as many QEPs as opened. You can execute an entire script or only selected SQL statements in a Query Analyzer.

You can achieve a number of functionalities using QEPs like:

- Execution of a complete script by creating or opening the script in the Editor pane and pressing F5.
- Execution of only selected SQL statements by highlighting the lines of code in the Editor pane and pressing F5.
- On clicking, it executes the selected Query or the one wherever cursor lies (only that particular row or query).

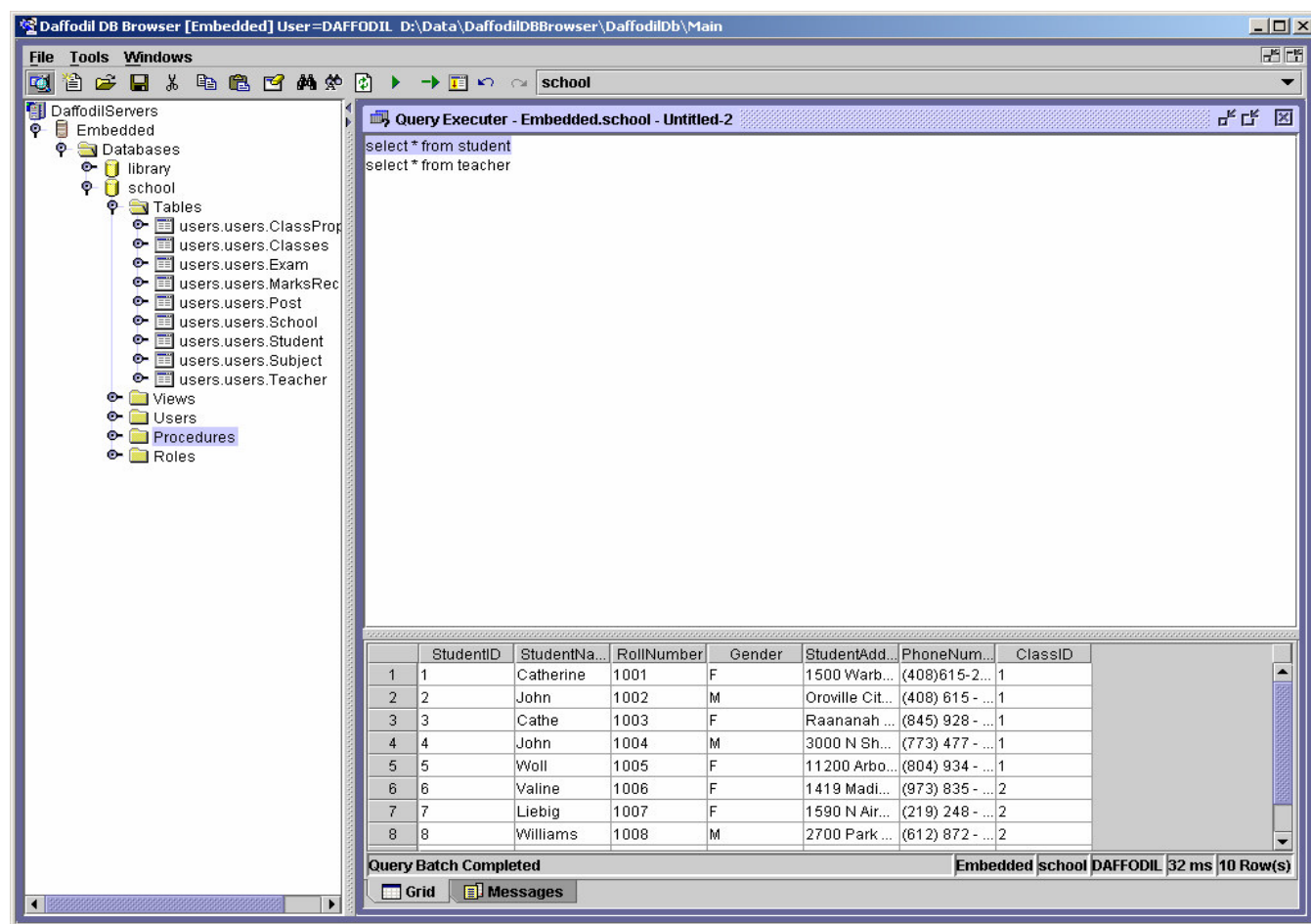
Examples

For executing a single line query, place cursor at that line and then press F5 or click Execute button on the toolbar.

`select * from student`

`select * from teacher`

In this case only "select * from student" will be executed as this is the query, which is selected on pressing F5.



For executing a multiple line query, select the whole query and then press F5 or click on Execute button on the toolbar.

```
select *
```

```
from student
```

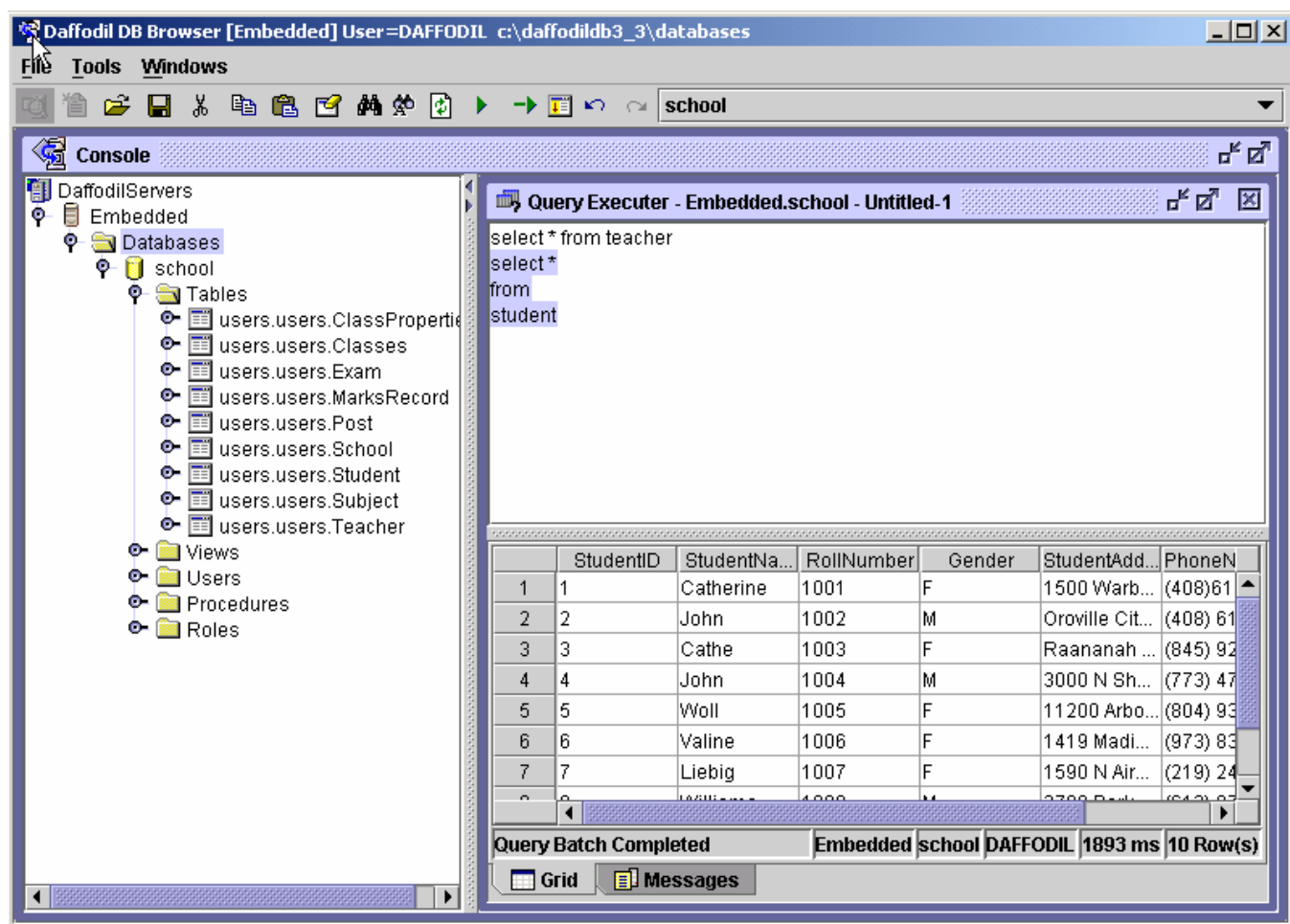
```
select * from teacher
```

In this case again only "select * from student" will be executed as

```
select *
```

```
from student
```

was selected.



For executing more than one query, select queries to execute and then press F5 or click on "Execute" button on the toolbar.

`select * from teacher`

`select * from student`

In this case both "select * from teacher" and "select * from student" queries will be executed as both

`select * from student`

`select * from teacher`

were selected.

The screenshot shows the Daffodil DB Browser interface. The main window is titled "Query Executor - Embedded.school - Untitled-1". It contains two SQL queries: `select * from student` and `select * from teacher`. Below the queries, there are two result grids. The first grid shows the results of the `select * from student` query, and the second grid shows the results of the `select * from teacher` query. The status bar at the bottom indicates "Query Batch Completed" and "Embedded school DAFFODIL 906 ms 10 Row(s)".

	StudentID	StudentNa...	RollNumber	Gender	StudentAdd...	PhoneNum...	ClassID
1	1	Catherine	1001	F	1500 Warb...	(408)615-2...	1
2	2	John	1002	M	Oroville Cit...	(408) 615 - ...	1
3	3	Cathe	1003	F	Raanahah ...	(845) 928 - ...	1
4	4	John	1004	M	3000 N Sh...	(773) 477 - ...	1
5	5	Woll	1005	F	11200 Arbo...	(804) 934 - ...	1
6	6	Valine	1006	F	1419 Madi...	(973) 835 - ...	2
7	7	Liebig	1007	F	1590 N Air...	(219) 248 - ...	2
8	8	Williams	1008	M	2700 Park ...	(612) 872 - ...	2
9	9	Tovera	1009	M	3048 Bright...	(718) 646 - ...	2

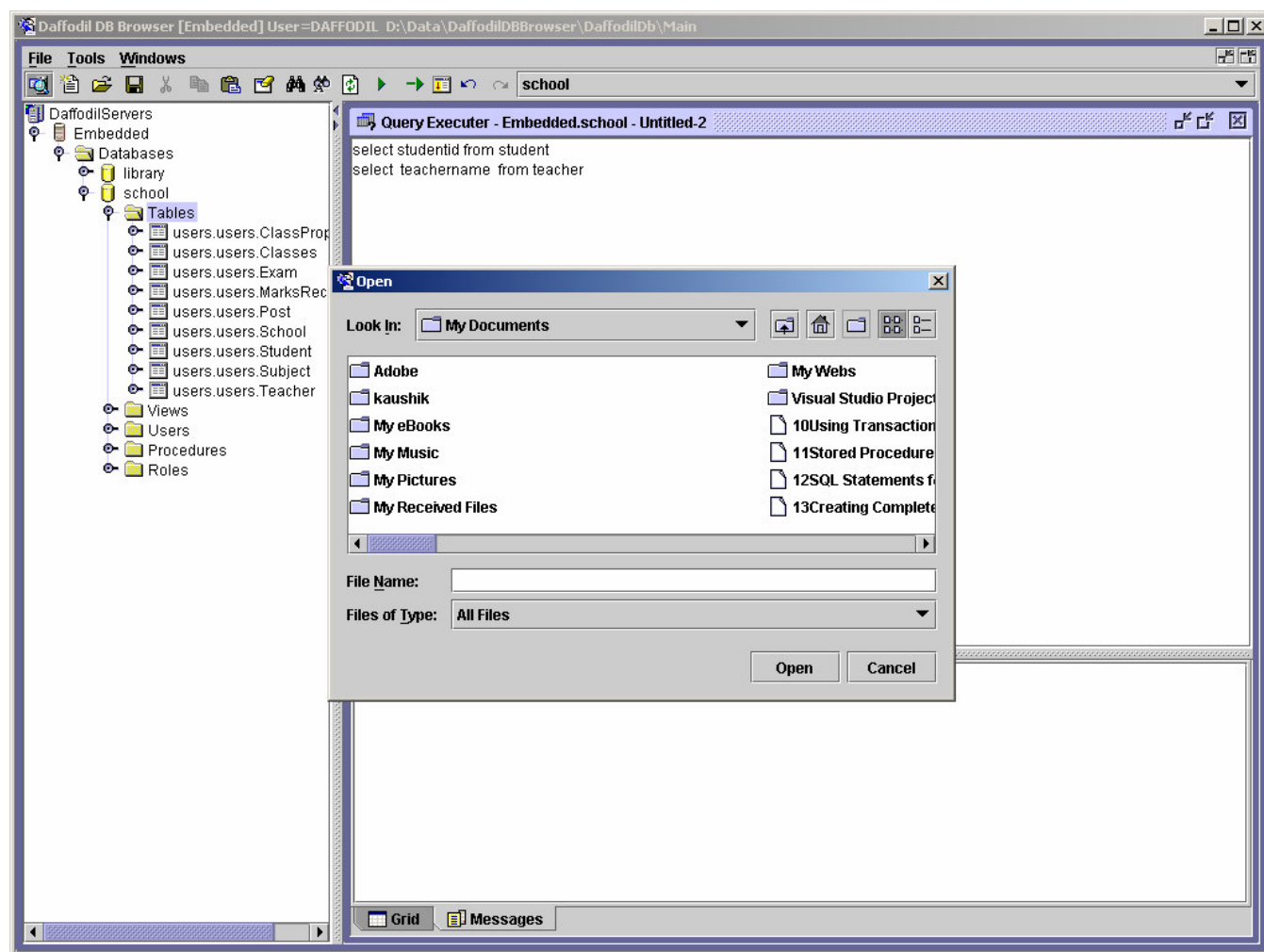
	EmployeeID	TeacherNa...	DateOfJoini...	DateOfBirth	Salary	DepartName	PostID	SchoolID
1	1	Mr. Agregado	1996-04-17	1965-04-10	10000	English	1	1
2	2	Mr. Brumfield	1997-07-01	1966-11-27	8500	Science	2	1
3	3	Ms. McKelvey	1998-09-25	1968-01-07	6000	Science	3	1
4	4	Mr. Everett	1998-10-25	1968-01-17	6000	Math	3	1
5	5	Mr. Verstre...	1998-08-29	1969-05-18	6000	Computer	3	1
6	6	Mr. Haight	1998-09-15	1968-02-22	6000	English	3	1
7	7	Ms. Hartenf...	1998-09-15	1970-01-06	6000	Social Stud...	3	1
8	8	Mr. Henry	1998-10-25	1969-03-23	6000	Biology	3	1

Save Queries written in QEPs in a file

User can save all the queries written in a QEP by pressing save button on toolbar or in Query Analyzer popup menu. On pressing save option, user is prompted to enter the file name, and on entering the file name, all the queries in the currently selected executer pad gets saved to the named file.

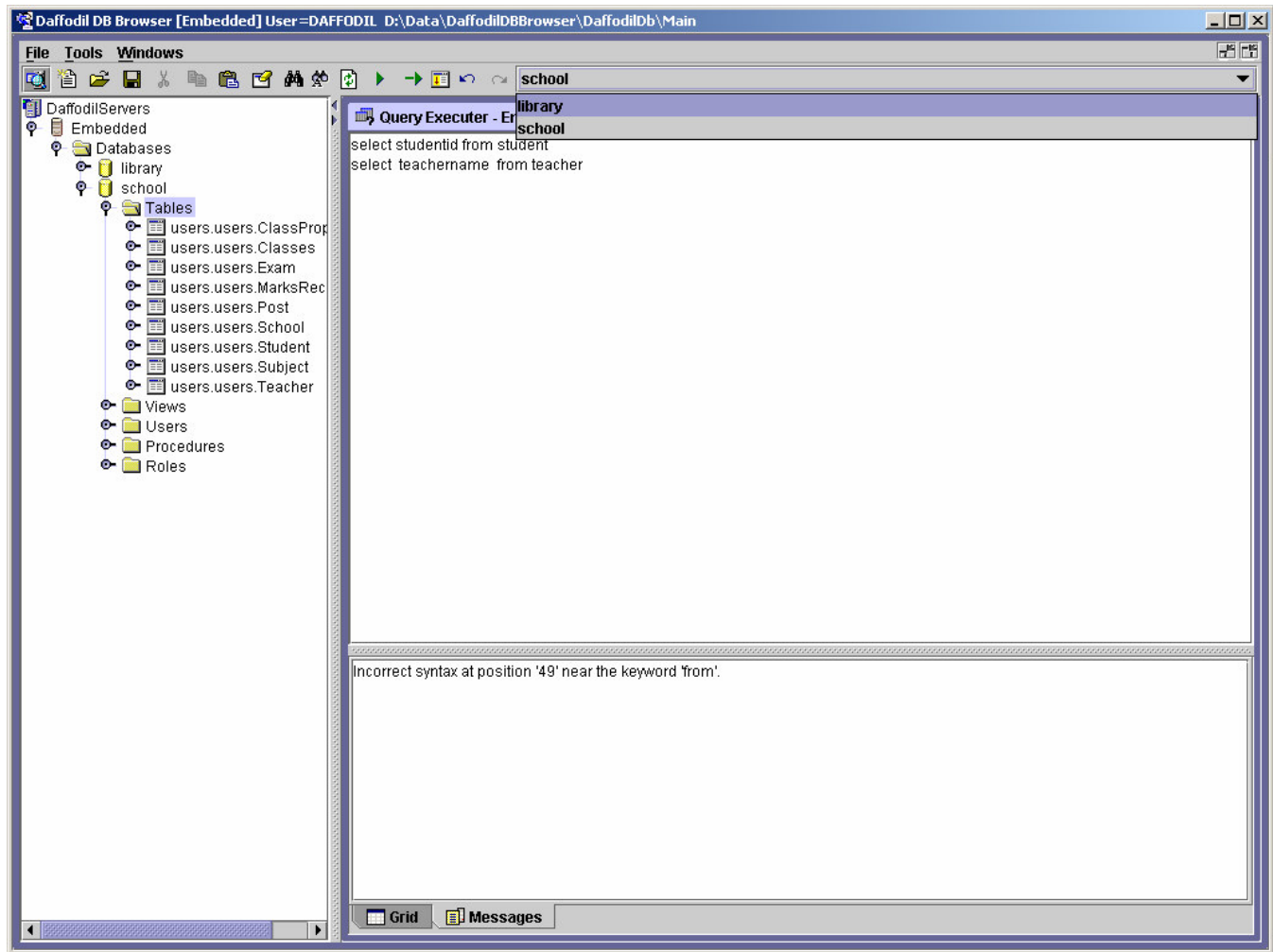
Load Queries saved in files

User can retrieve any query file that had been saved earlier by using 'Load Queries saved in files' option from the toolbar or from Query Analyzer popup menu. On pressing load script option, user is prompted to enter the file name and on entering the file name, all the queries, which are saved within the file, can be viewed on the currently selected executer pad.



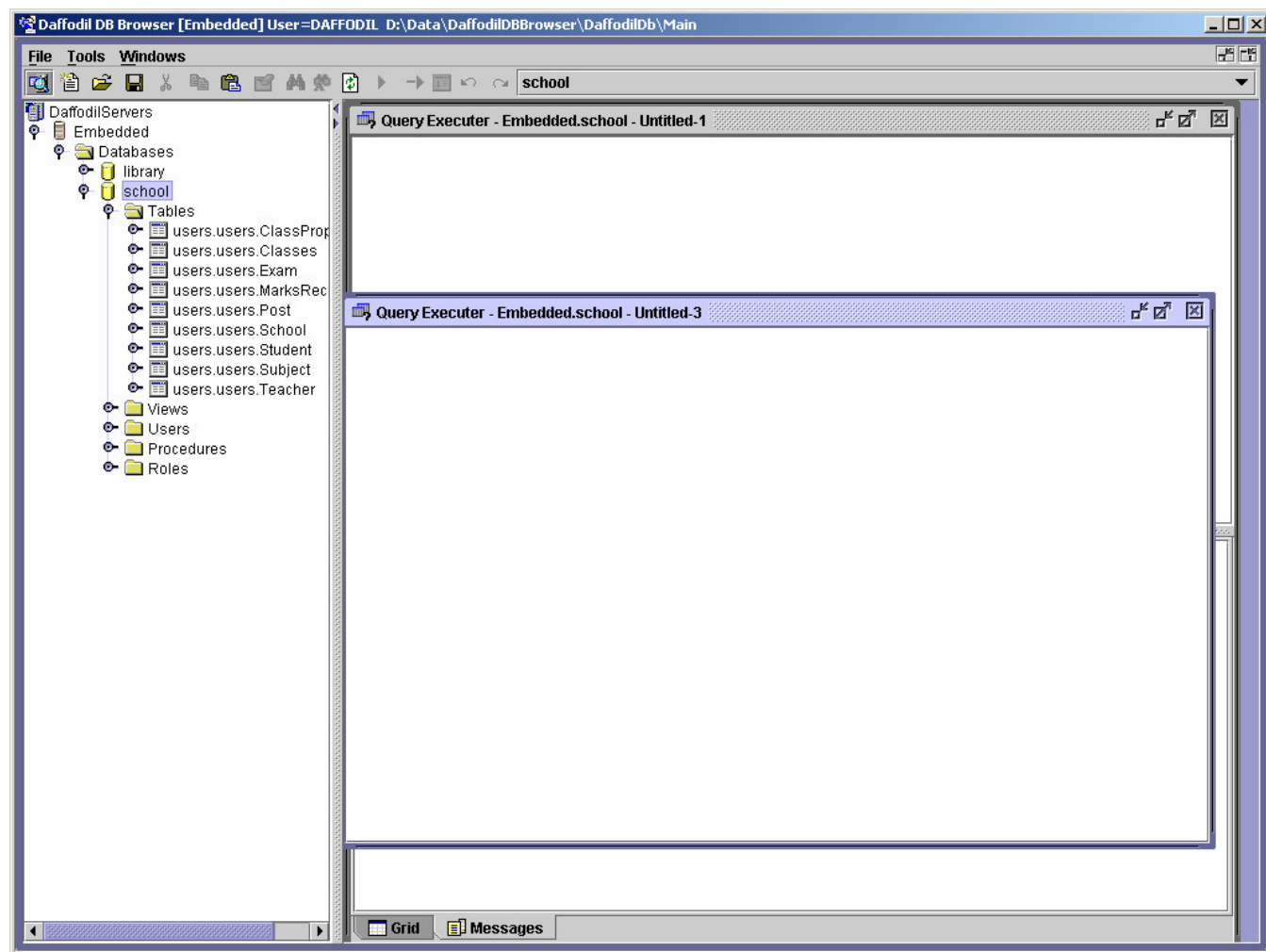
Work using Multiple Query Executor Pads

User can start multiple Query Executor Pads in order to work with different combinations of databases + sessions (to be selected from database combos in toolbar) using the 'New Query Executor' Option in toolbar or Query Analyzer. User can fire queries in currently selected executor pad for the database selected from database combo in toolbar.



Run Multiple Sessions against a Database

A user can work in different sessions on a database. User can switch between different sessions by selecting between multiple Query Executer Pads



Execute Parameterized Queries

Parameterized Query is one of the unique features that are available in Daffodil DB Browser. Using Parameterized Queries, user frames the query once and '?' is passed in place where conditional parameters are required. During execution, number of Text Boxes appear in the panel corresponding to each '?' passed in the query.

User is also provided with the "Type of the value" that is to be passed, by providing the name of the field corresponding to which value would be required.

Text Boxes will be displayed corresponding to each Column Name, that will indicate *the type of value* user is supposed to enter.

Passing Parameters in the Parameterized Query

There are various ways by which parameters can be passed. Passing of parameters differs in the way it appears in the queries, which involve predicates.

Note: - For detailed information on predicates, refer Daffodil DB SQL Reference Guide.

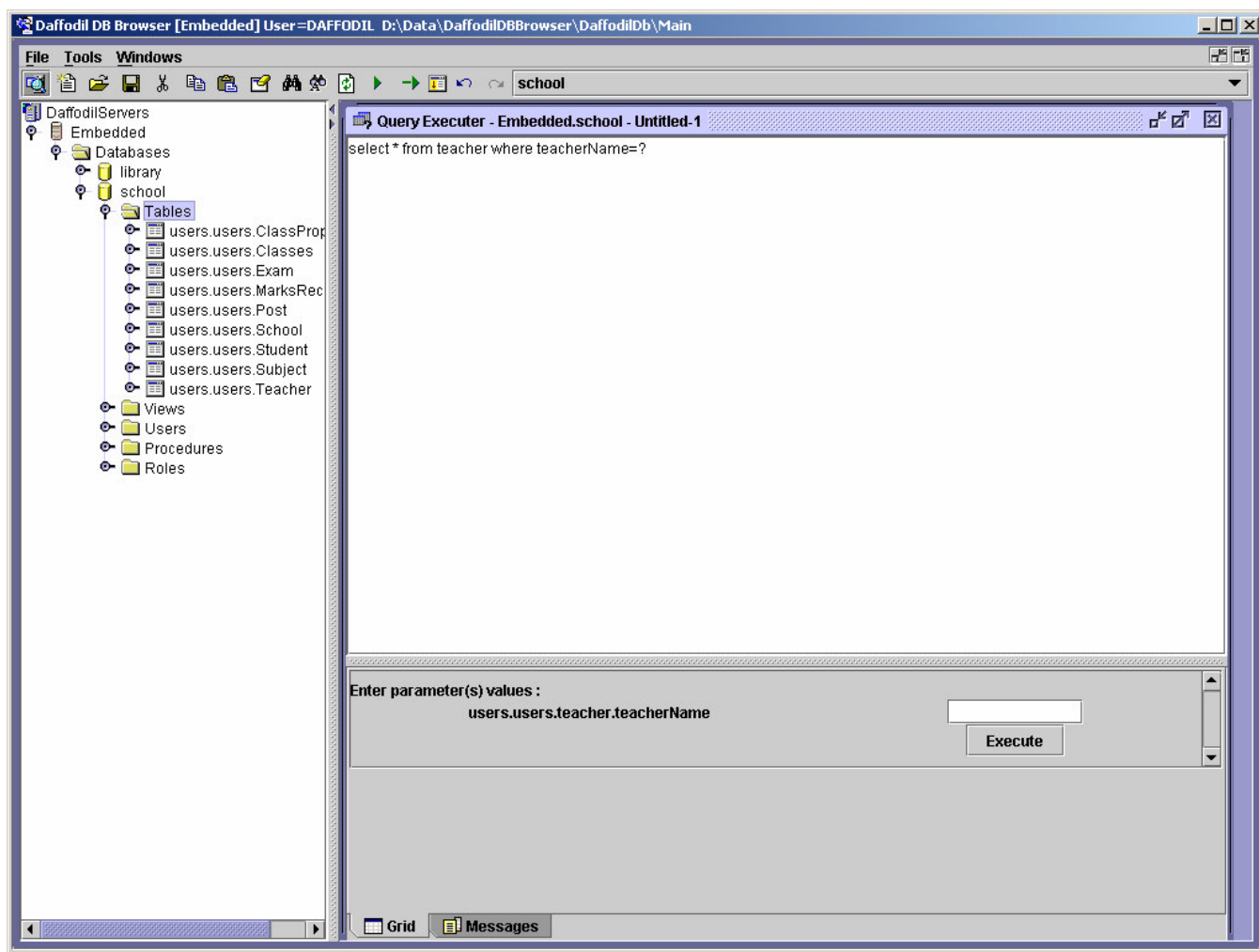
Following are some of the *examples*, which describe cases in which parameter passing is different. Comparison Predicate case

Example 1

`select * from teacher where TeacherName =?`

In the example above, one Text Box will be displayed against the label "TeacherName" as there exist a single question mark in the query.

Character/String like values are required to be passed without single quotes. For e.g. abc (and not 'abc').

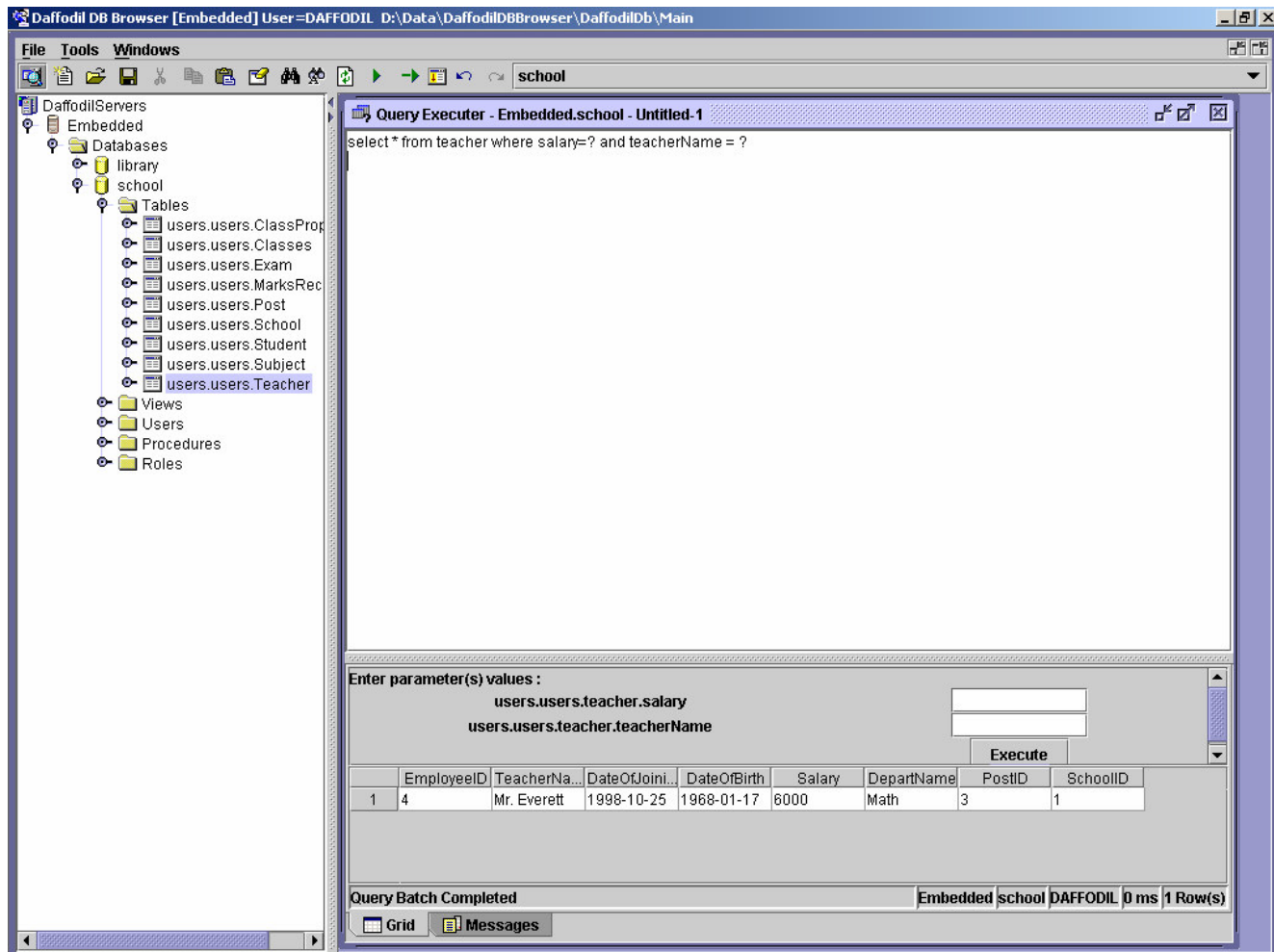


Example 2

```
select * from teacher where TeacherName =? and salary =?
```

In the example above, two text Boxes will be displayed against the label “TeacherName” and "salary" as there exist two question marks in the query.

Here values will be passed in the order of the columns and according to the type of the column.

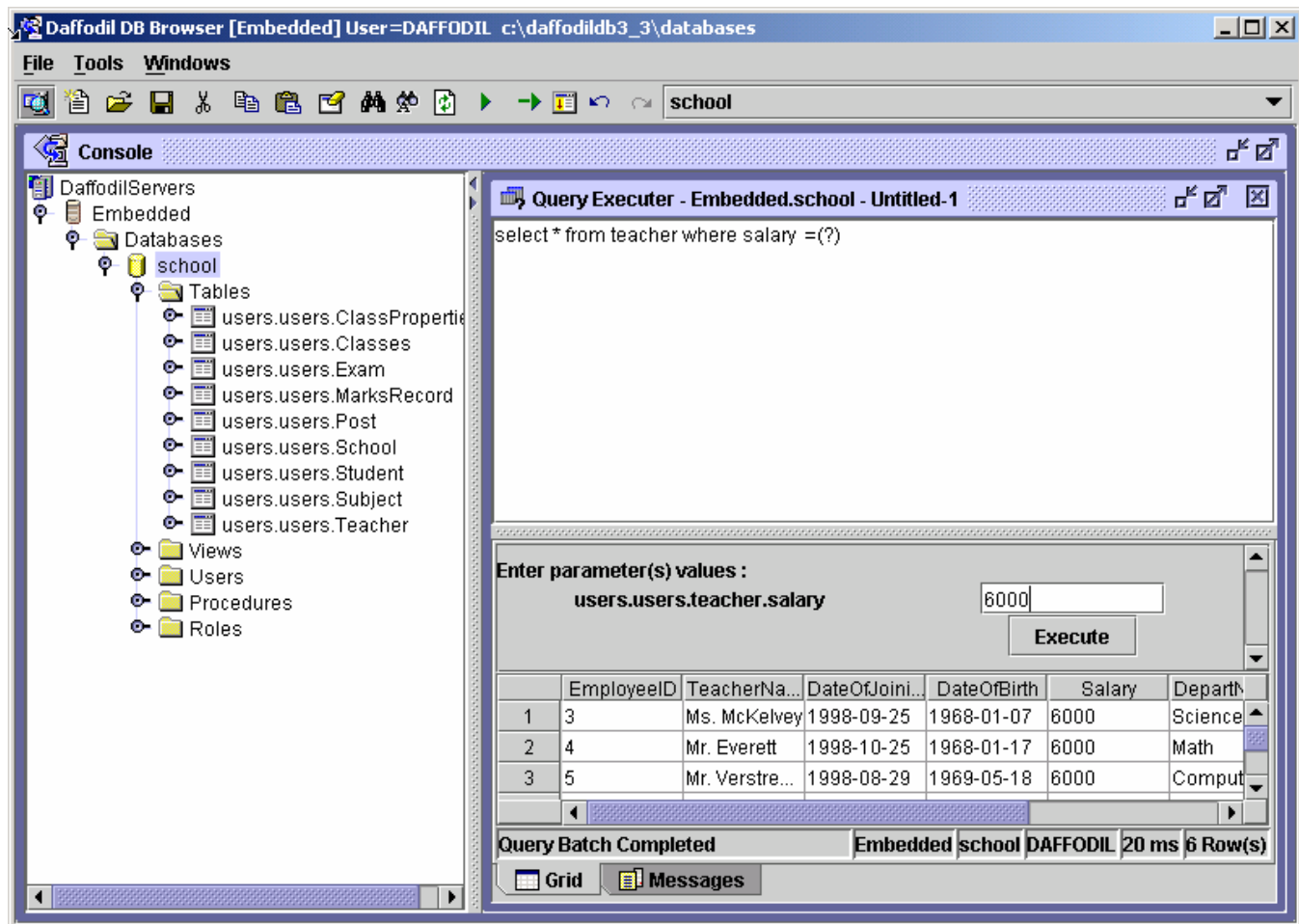


In Predicate case

Example 1

`select * from teacher where salary = (?)`

In the example above, a single text box will appear and a single value will be required to be specified in the text box.



Example 2

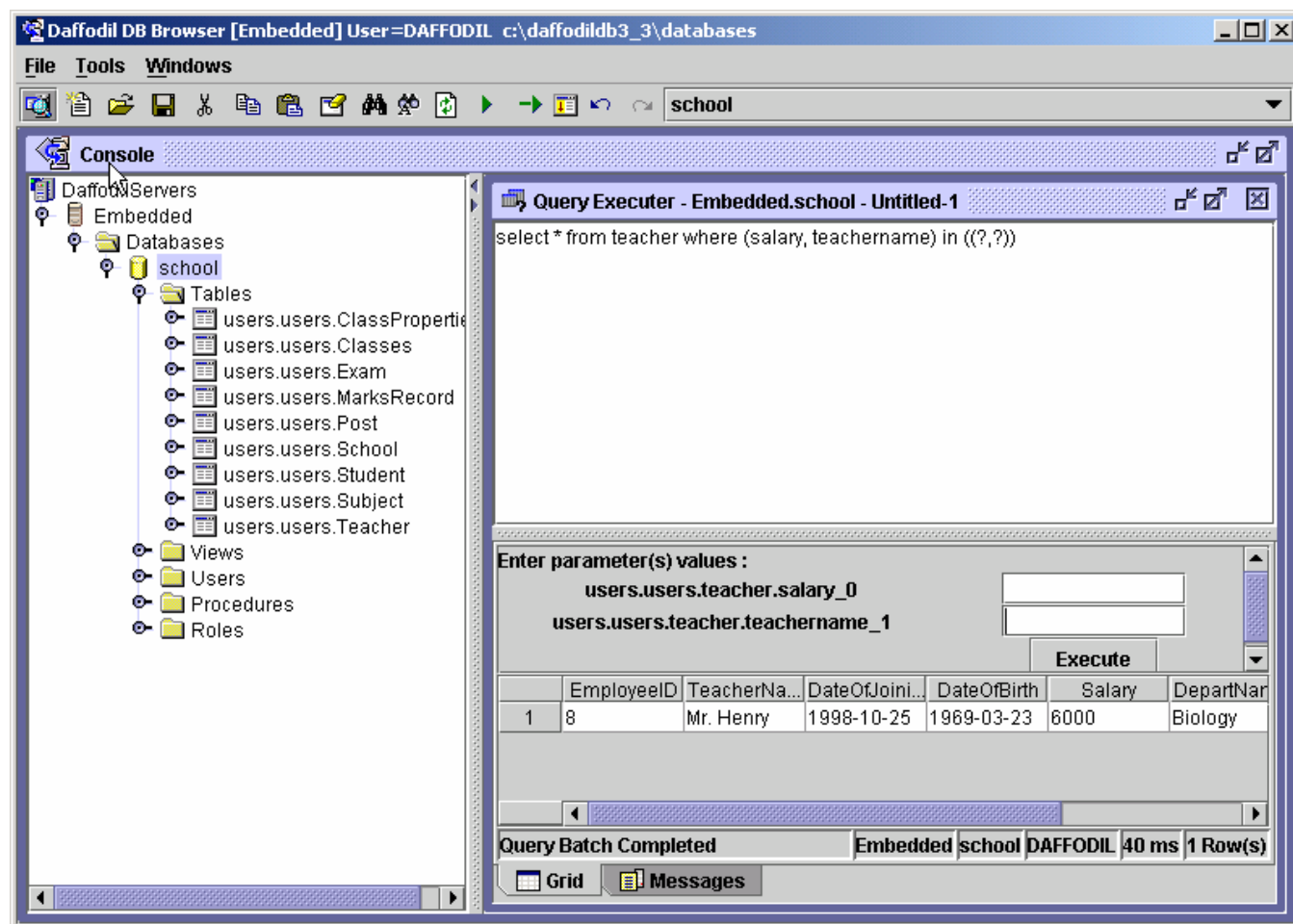
`select * from teacher where (salary, teachername) in ((?,?))`

In the example above, '?' refers to value corresponding to each mentioned column. In this case two text Boxes are opened, one for each column.

Here single value will be provided, one for each column and in the following format: -

Value in Box 1 will be 6000

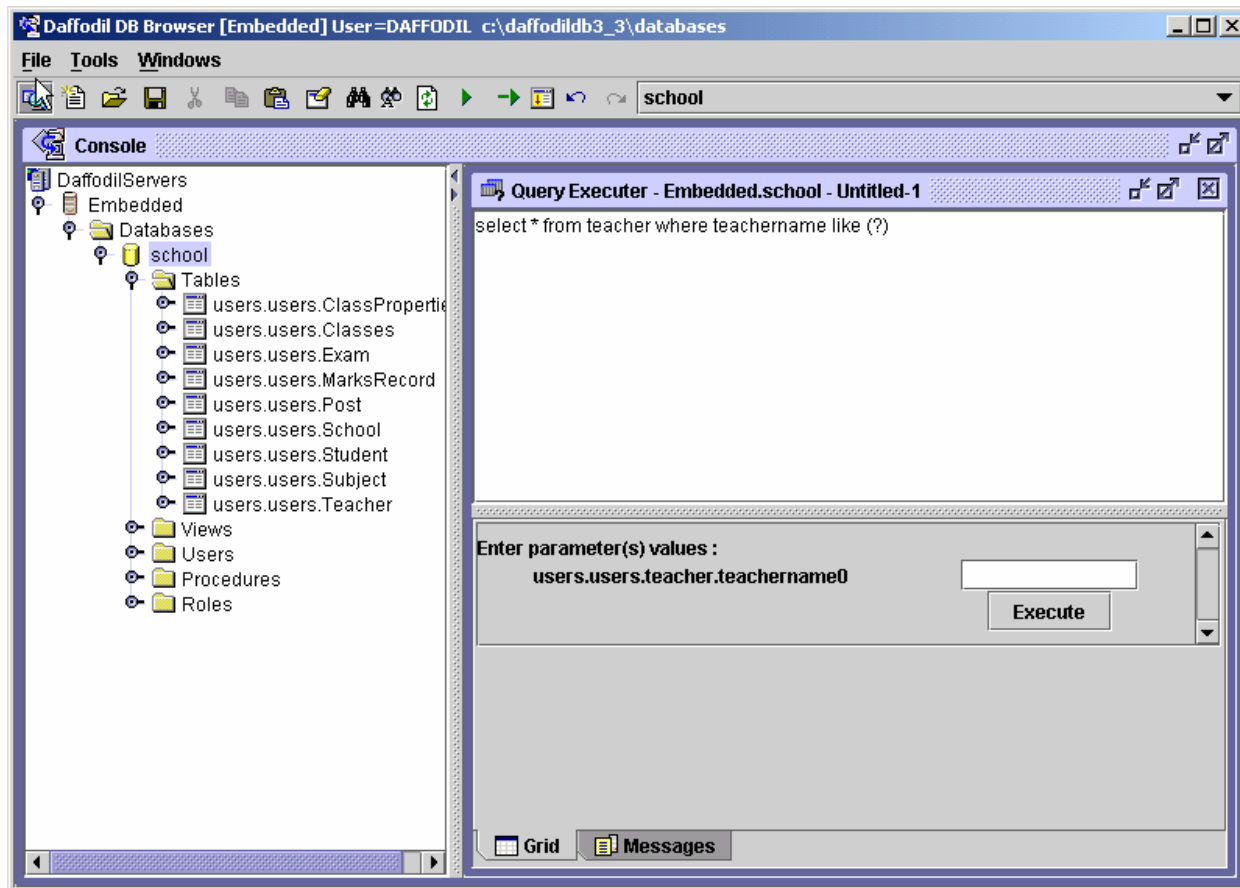
Value in Box 2 will be Mr. Henry.



Like Predicate case

Let us suppose parameterized query to be:

`select * from teacher where teachername Like (?)`

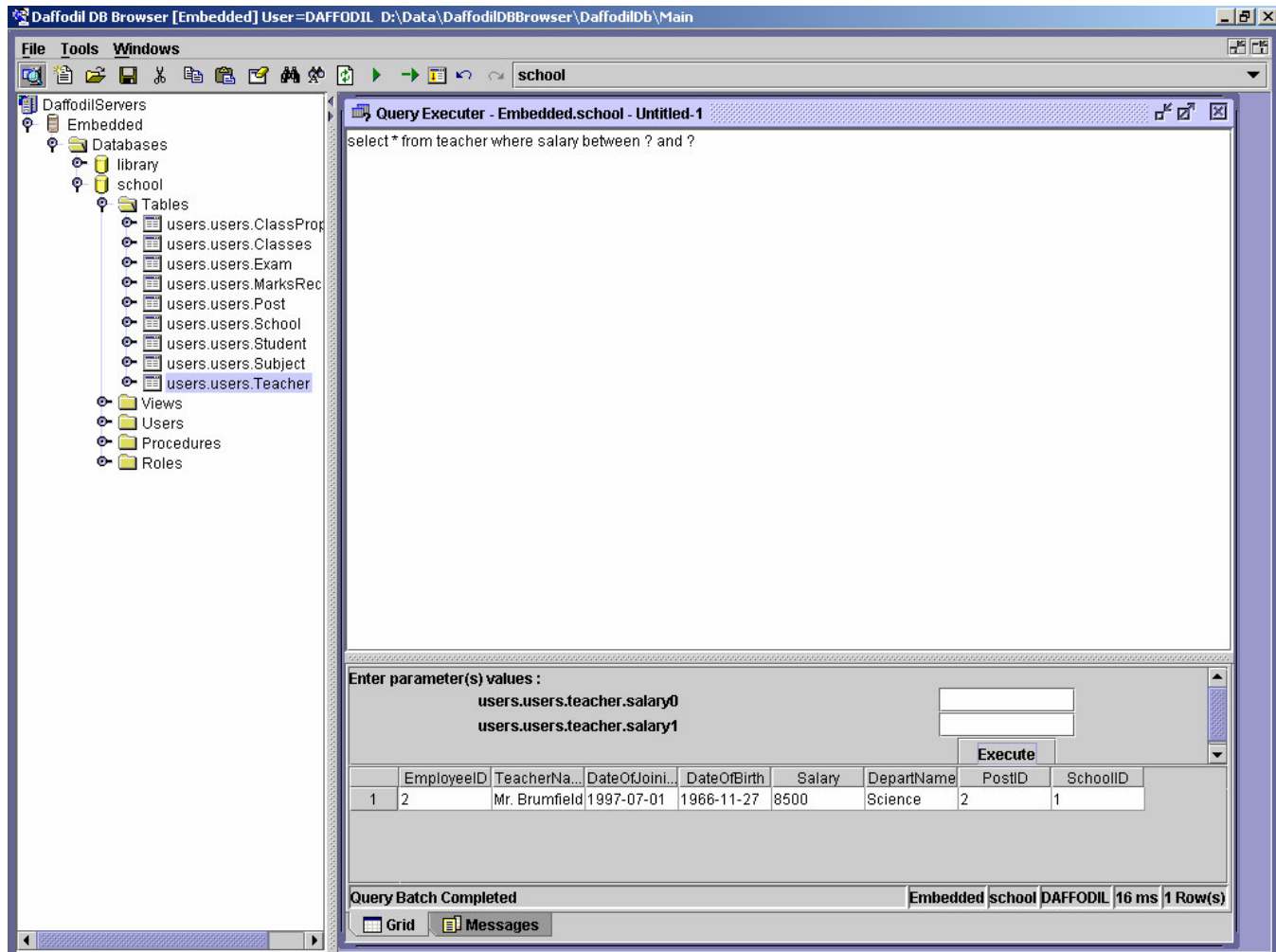


In the example above, a single box will be opened and if you pass the value `s%`, all rows starting with 's' from the column `teachername` will be retrieved.

Between Predicate case

Example

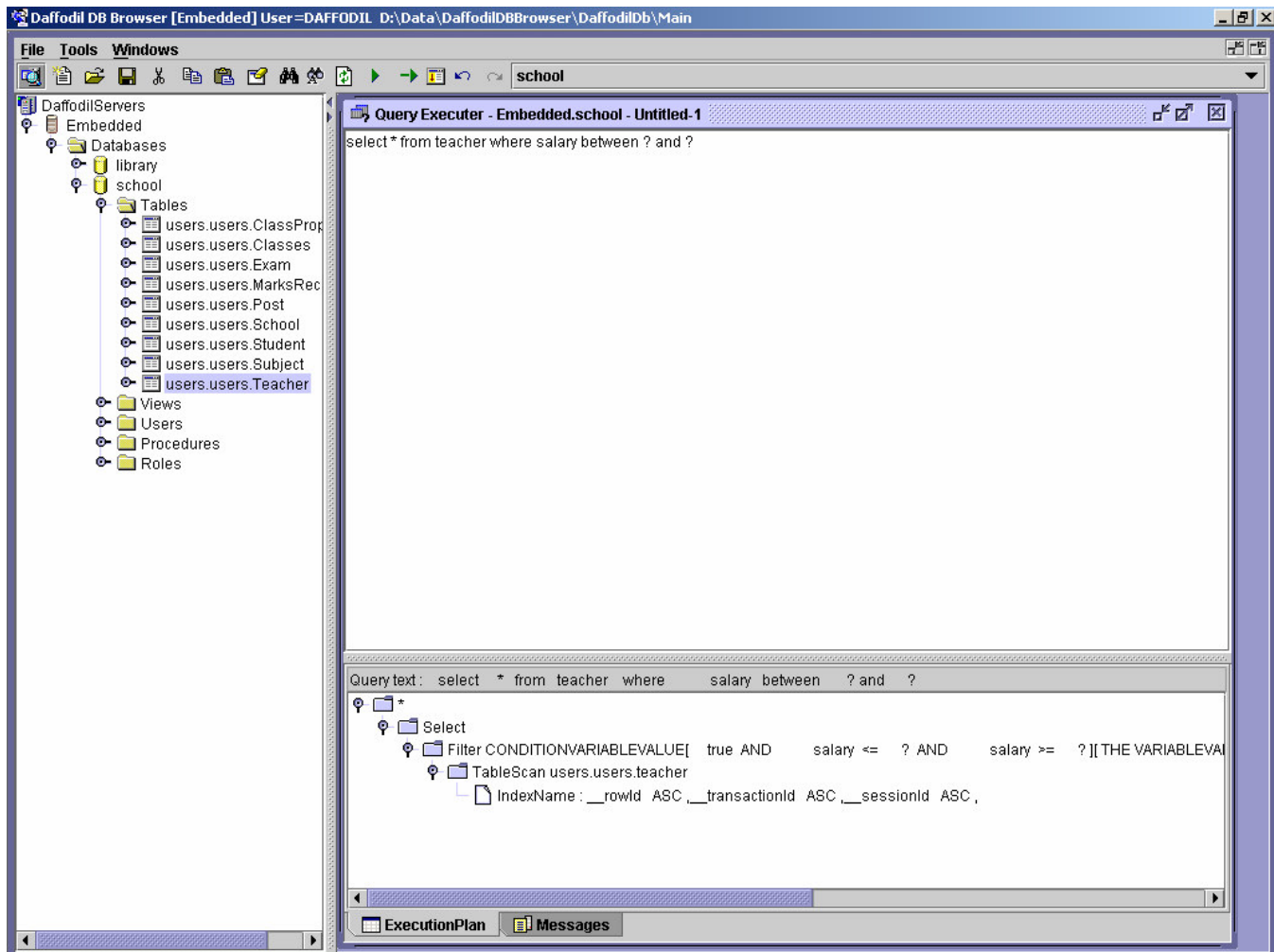
select * from teacher where salary between ? and ?



In the example above, two text boxes will be opened, one for each '?' and the values will be provided in a asymmetric order.

ShowExecutionAction for Browser

If user wants to see the Execution plan of select query, then user may click *ShowExecutionPlan* button. On clicking, Execution plan will be displayed.



Daffodil DB Browser Mode

Perform DDL Operations in Browser Mode.

DDL operations are used to create, manipulate and drop schema definitions.

Note: - For Detailed Information on DDL Statements, refer Daffodil DB SQL Reference Guide

The operations supported in Browser Mode are:

- Create Database Statement
- Drop Database Statement
- Create Table Statement
- Drop Table Statement
- Create View Statement
- Create Trigger Statement
- Create Procedure Statement

Create Database

Creation of Database in the Browser Mode is a two step process.

- Right click the database node and then select New Database option from the pop up menu.
- A dialog box will appear, requesting user to enter the database name. On entering database name and pressing “OK”, database with the specified name is created.

Drop Database

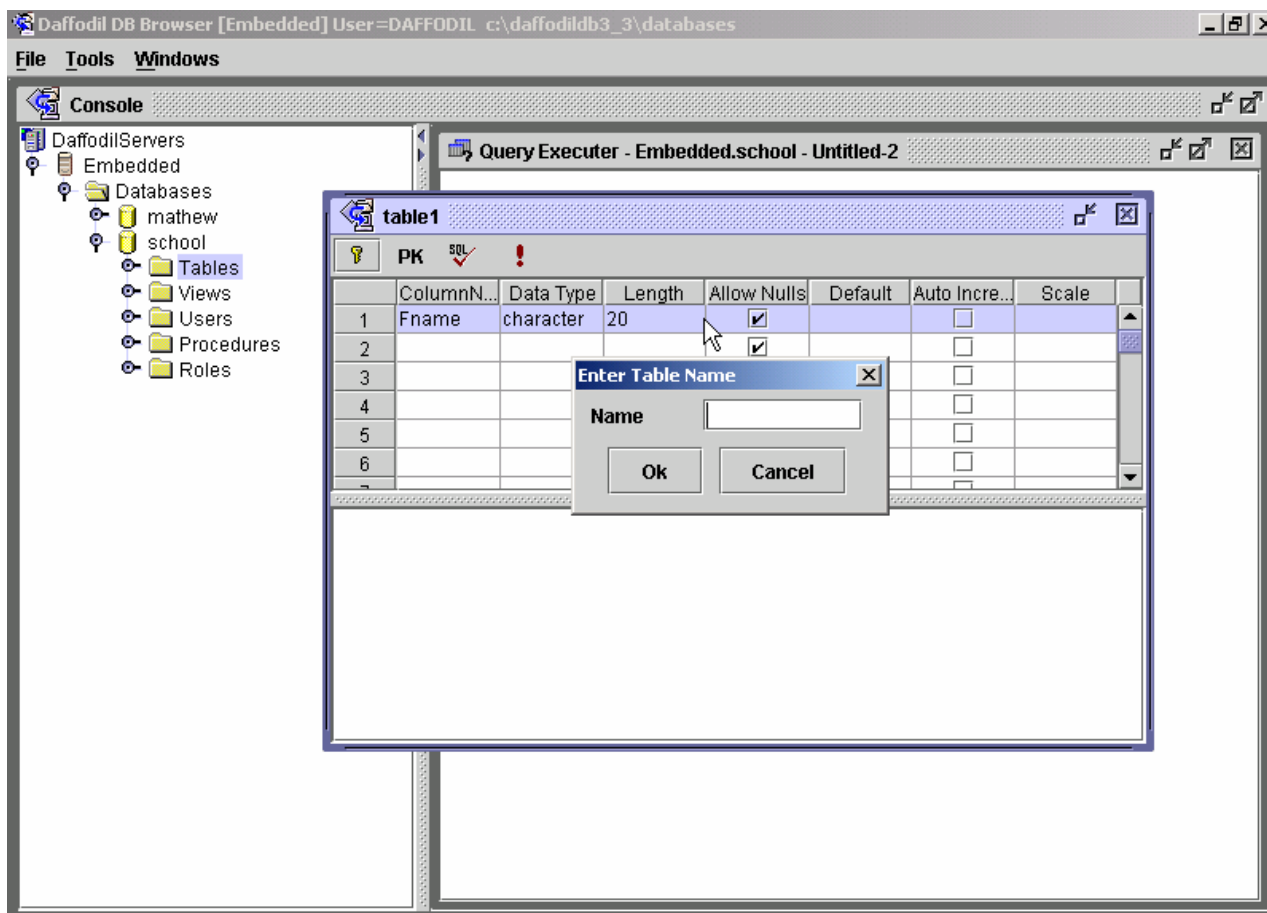
Dropping Database in the Browser Mode:

- Right click on the name of database to be deleted and select Drop Database option from the pop up menu.

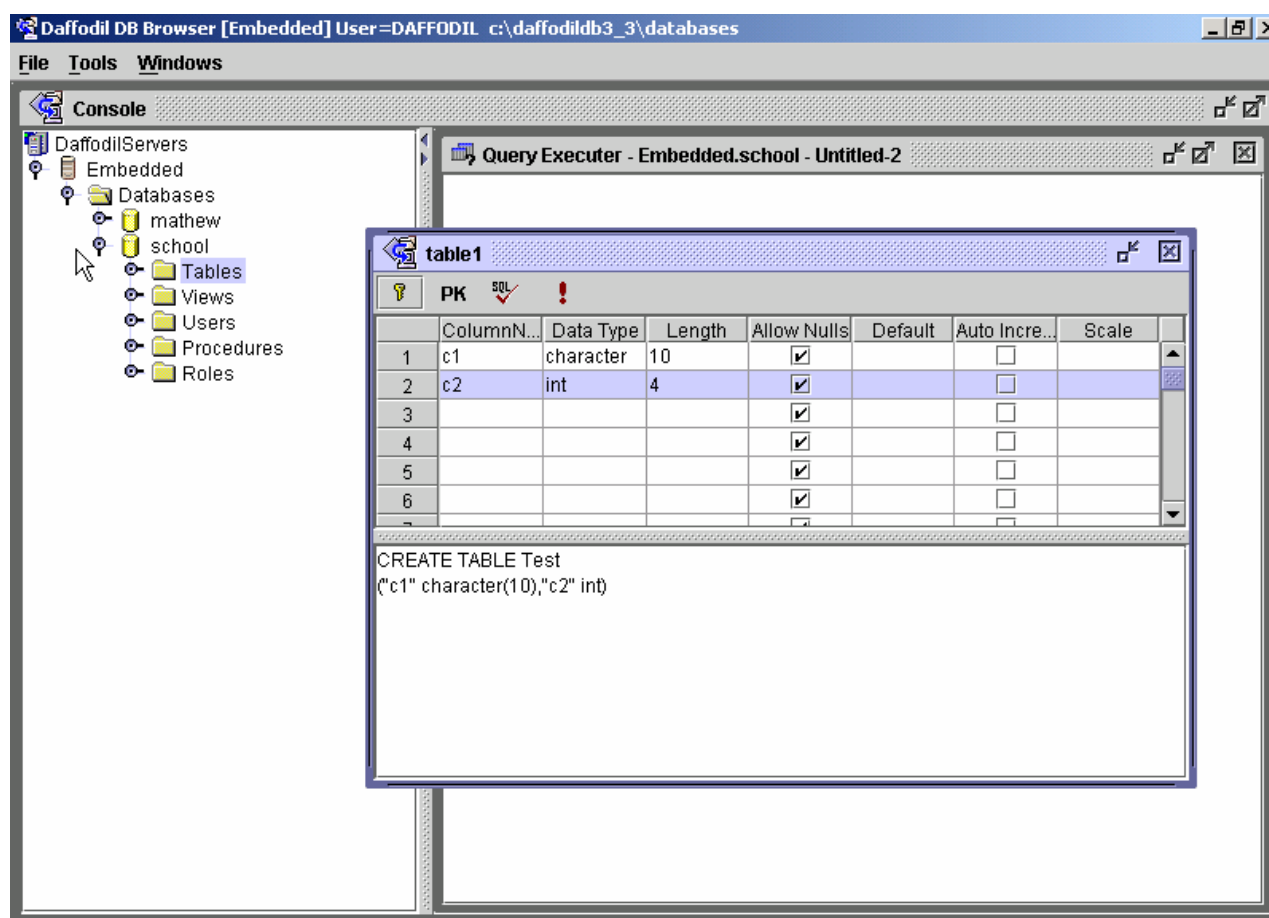
Create Table

Various Steps that are needed to perform Create Table Statement in the Browser Mode are as follows:

- Browse the tree to the node named **“Tables”**. Right Click this node and select New Table option from the appeared popup menu.
- On clicking the SQL button, a dialog box appears prompting for the table name.



- Enter table name.
- Click OK button.
- After entering the table name, user has to specify the information on table columns i.e. column names, column types, column size and whether nulls are allowed for the columns or not.
- User can specify selected column as primary key column or can also specify combination of selected columns as composite keys by pressing Primary Key button on the new table toolbar.



- Now after entering information in the table, user can click **SQL query (make) button** on the toolbar to view the query generated for this table. This query will be shown in the text Area.
- User may click **edit the query**, which is displayed in the text area.
- Once user enters the edit mode, there is no relation between the visual interface and query.
- To execute the query, user has to click **run** button on the toolbar.
- The dialog box appears prompting user to confirm if the query to be executed is the one they intended to fire.
- Click “OK” or “CANCEL”. On pressing Ok, if the table is created successfully then ‘Query successfully executed ‘ is displayed in the text area, otherwise relevant error message is displayed in the text Area.
- Click “REFRESH” button to display the currently created table.

Drop Table

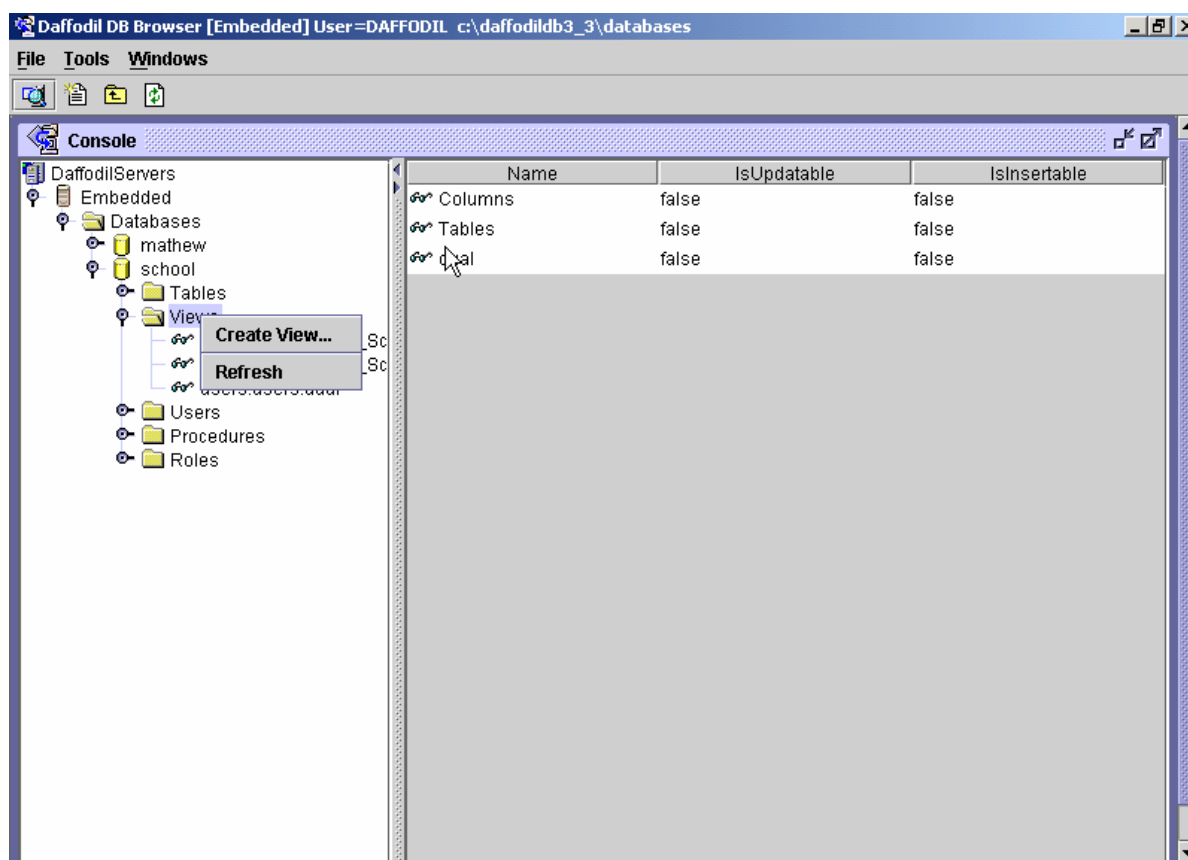
Dropping Table in the Browser Mode:

- Right click on the table name and you will get a drop option in the pop up menu. Select Drop option to drop the selected table.

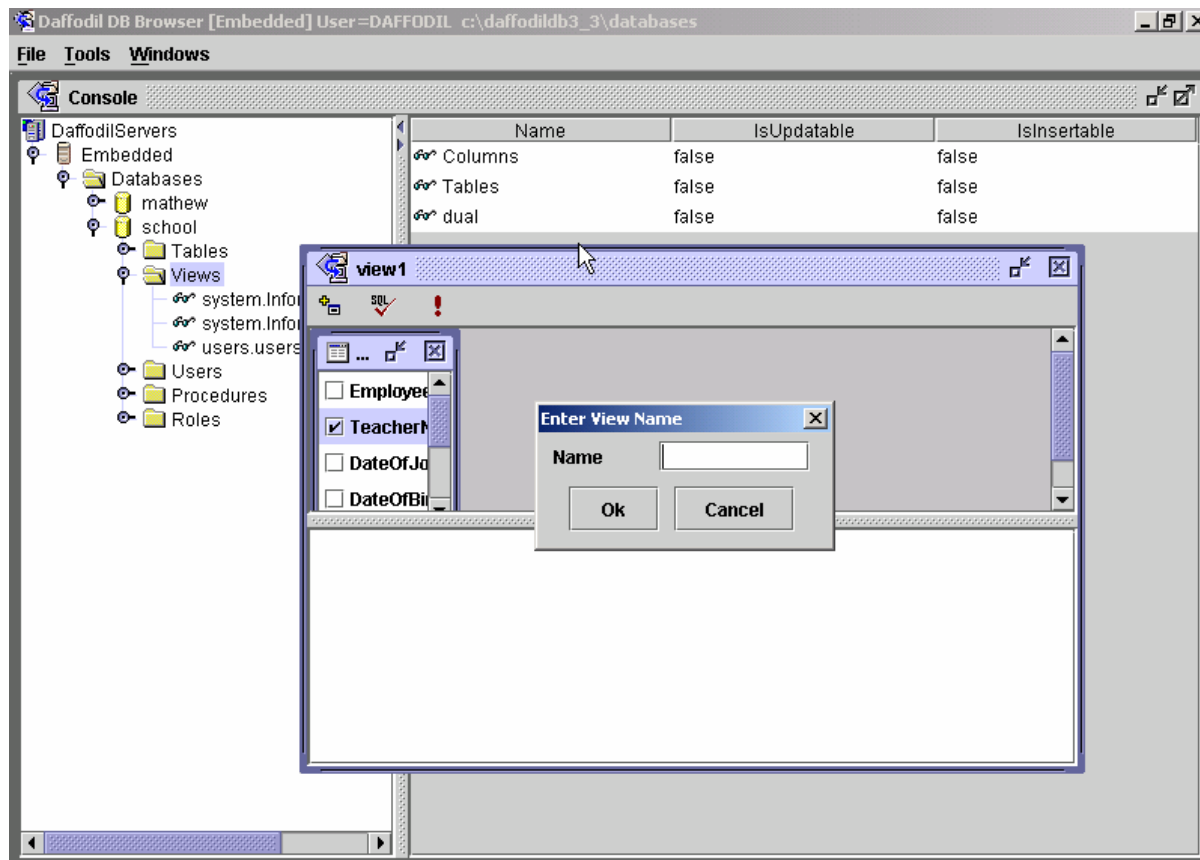
Create View

Various Steps are needed to perform Create View Statement in the Browser Mode.

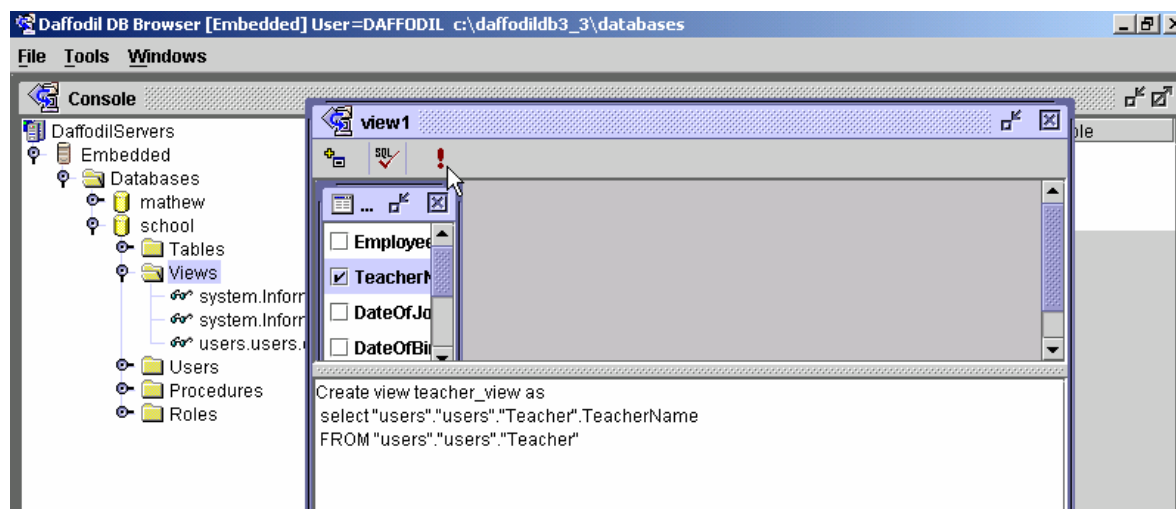
- Browse tree to the node named “**Views**”, Right Click the node and select New View option from the popup menu.
- A dialog box appears prompting for the view name.
- Enter view name.
- Click “OK” button.
- Click “**add table**” option to get tables for making a View. A dialog box is opened displaying tables present in the database.



- Click the table and press **add button** to add tables. User can add as many tables as he wants by using the “CTRL” key.
- User can create relationships by connecting table columns with the columns of the other tables that are compatible in data types.
- User has to select columns (from the tables displayed in the graphical interface), which he wants to have in his view. Click the columns that user wants to be in the **create view query**.
- Now after entering the information in the table, user can click **make button** or **SQL button** on the toolbar to view the query generated for this table. This query will be displayed in the text Area.



- User may edit the query, which is displayed in the text area.
- Once user enters edit mode, there is no relation between visual interface and the query.
- To execute the query, user has to click run button on the toolbar.
- The dialog box appears prompting user to confirm that the query to be executed is the one they intended to fire.
- Click “OK” or “CANCEL”. On pressing Ok, if the table is created successfully, then ‘Query successfully executed’ is displayed in the text area, otherwise relevant error message is displayed in the text Area.



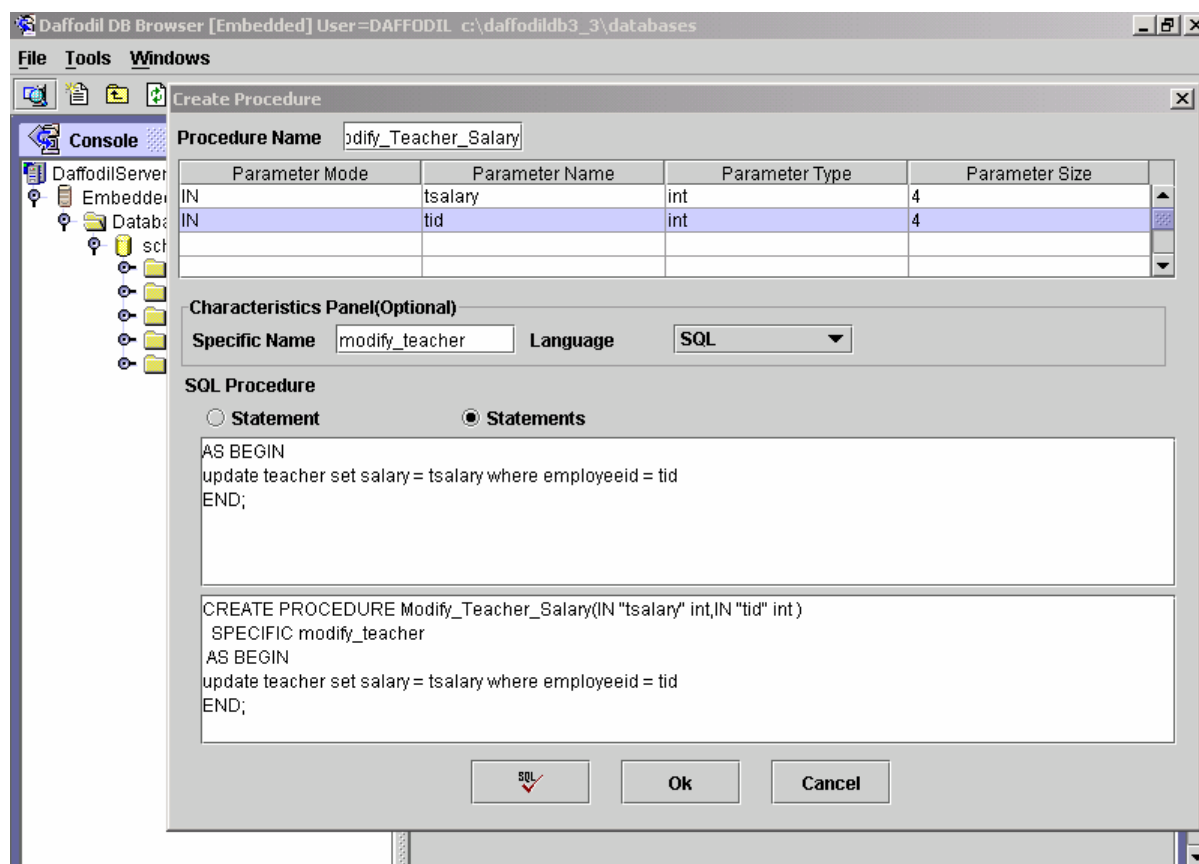
Create Procedure

Various Steps that are needed to perform Create Procedure Statement in the Browser Mode are as follows:

- Browse the tree to the node named **“Procedures”**. Right Click the node and select New Procedure option from the popup menu.
- The Procedure dialog box appears that allows user to specify procedure on a particular table.
- User has to specify the Procedure name.
- User can specify the parameters or can leave it, as this feature is optional.

Note: Only IN Parameters are supported in procedure statement in the Browser Mode.

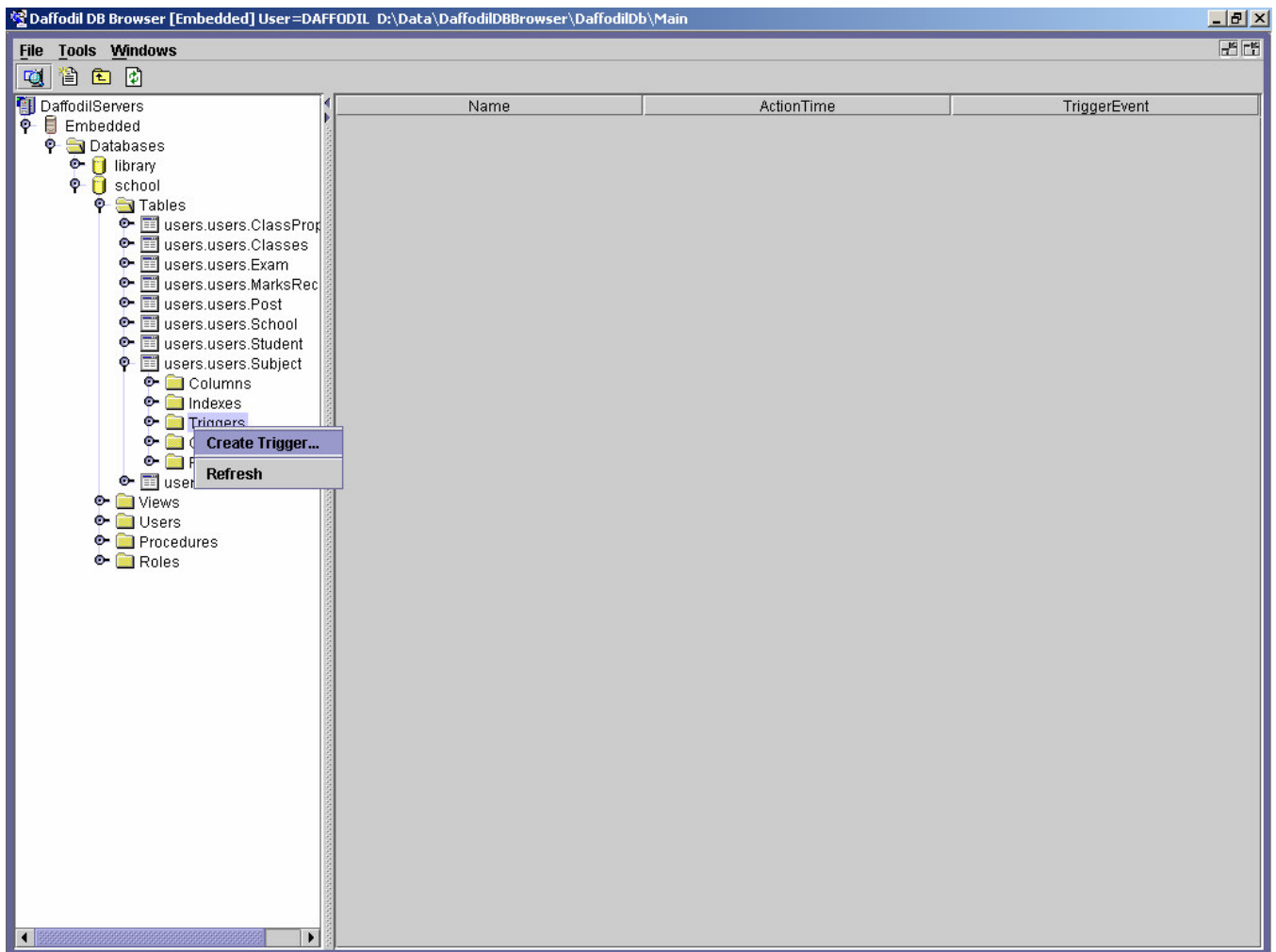
- The specific name of the procedure and language is optional. User can leave it vacant.
- The SQL procedure statement can not be left empty. User has to specify the procedure statement.
- After entering all the required information, user has to press “OK” button to fire the create procedure statement. If the procedure is successfully created, a message “Procedure Successfully Created” is displayed, otherwise appropriate error message is displayed.

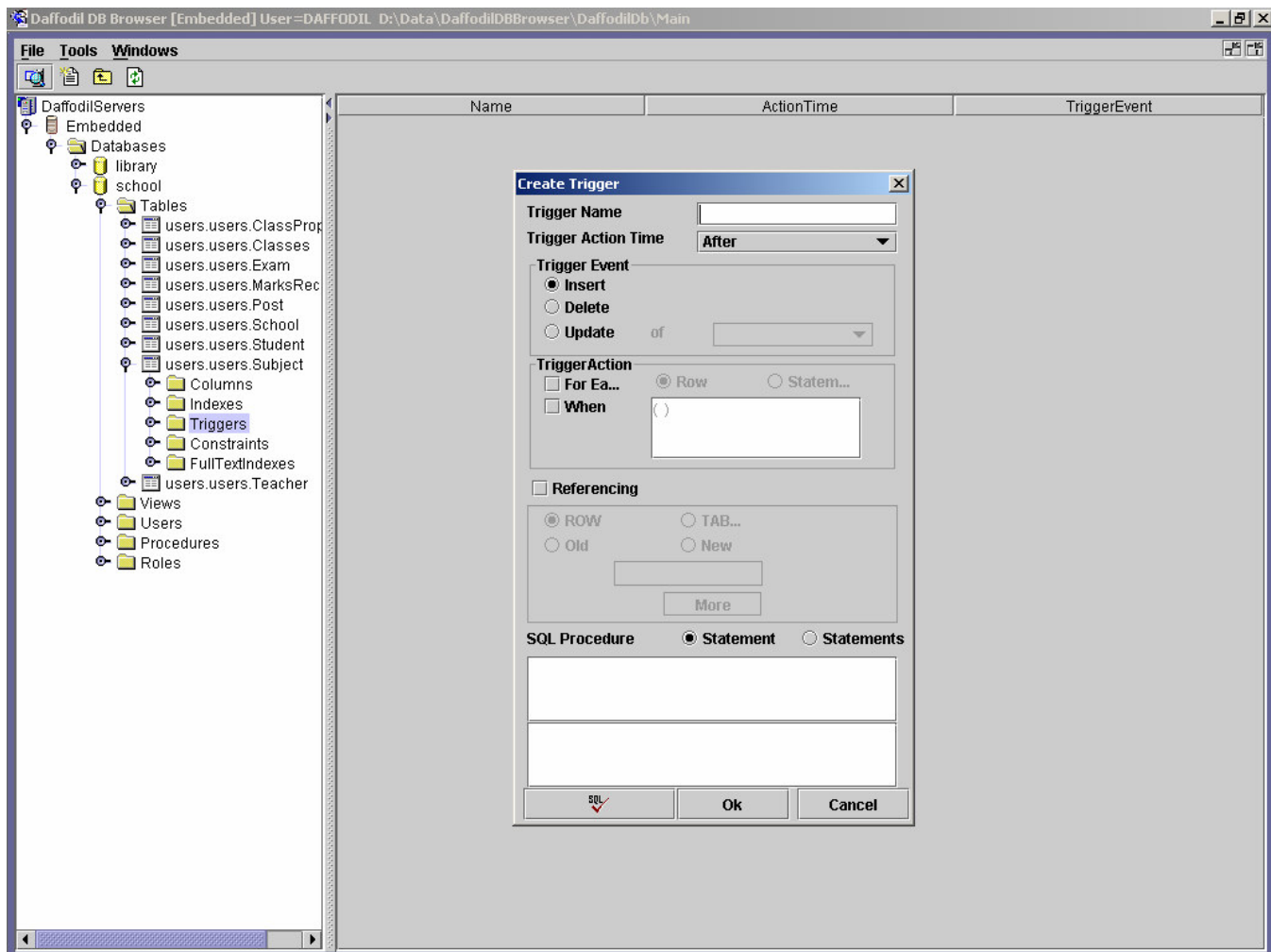


Create Trigger

Various Steps needed to perform Create Procedure Statement in the Browser Mode are as follows:

- Browse the tree to the node named **“Triggers”**. Right Click this node and select New Trigger option from the popup menu.
- The Create Trigger dialog box appears that allows user to specify trigger on a particular table.
- It is mandatory to specify the trigger name; it can not be left vacant.
- The Trigger Action Pane is optional.
- User can escape the referencing part but in case it is marked as checked by the user, then user has to specify correlation name or alias name.
- User has to specify at least one SQL statement and this field can not be left vacant.
- After entering all the required information, user has to click “OK” button to fire the Create Trigger statement. If the procedure is successfully created, a message “Trigger Successfully Deployed” is displayed, otherwise appropriate error message is displayed.

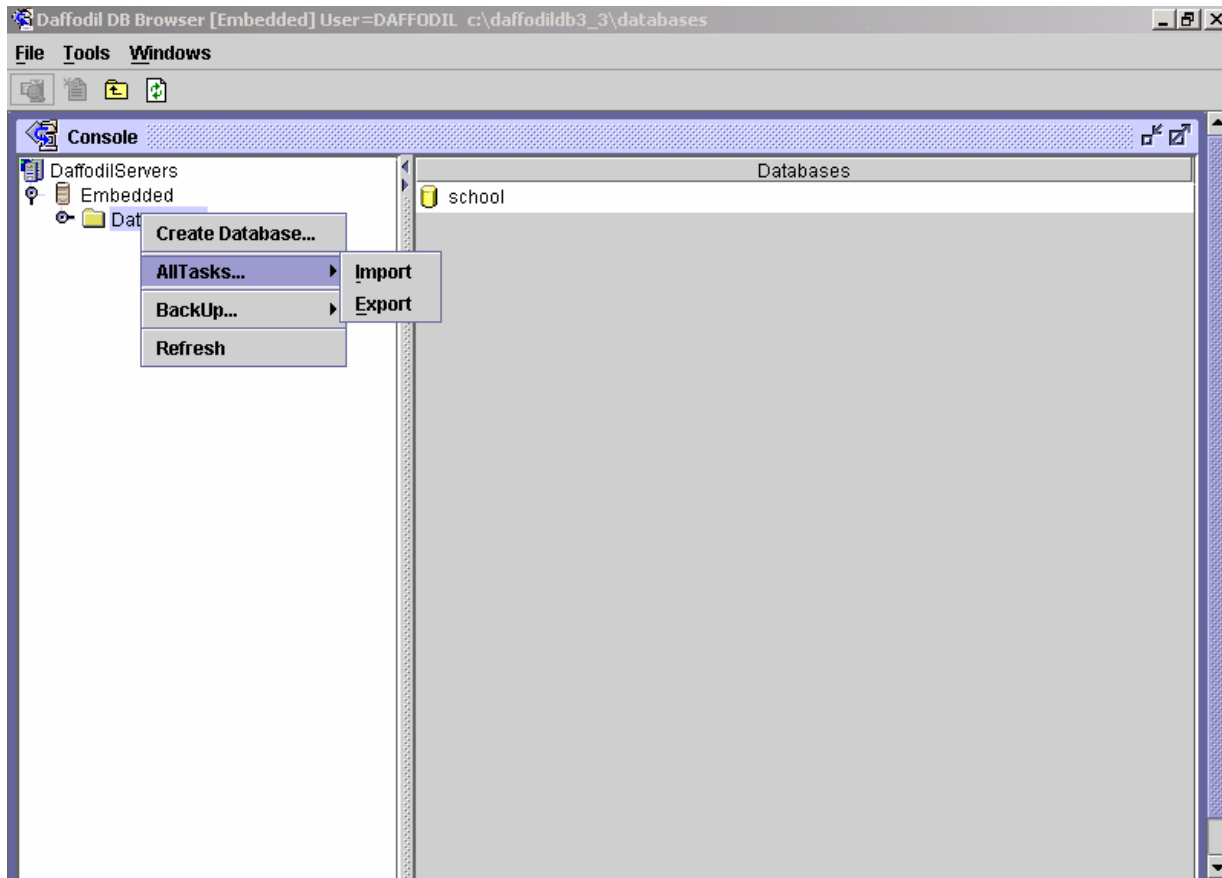




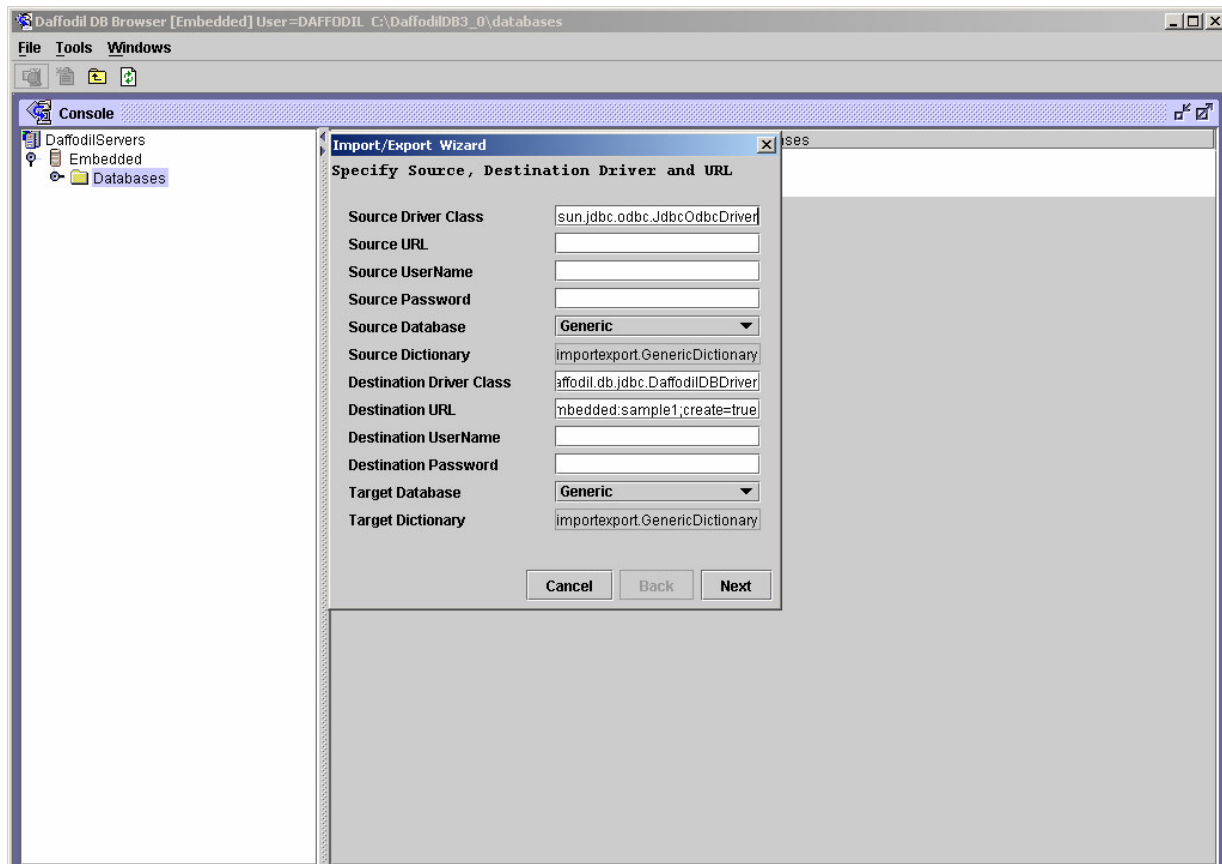
Import/Export

Various Steps needed to import/export databases in the Browser Mode are:

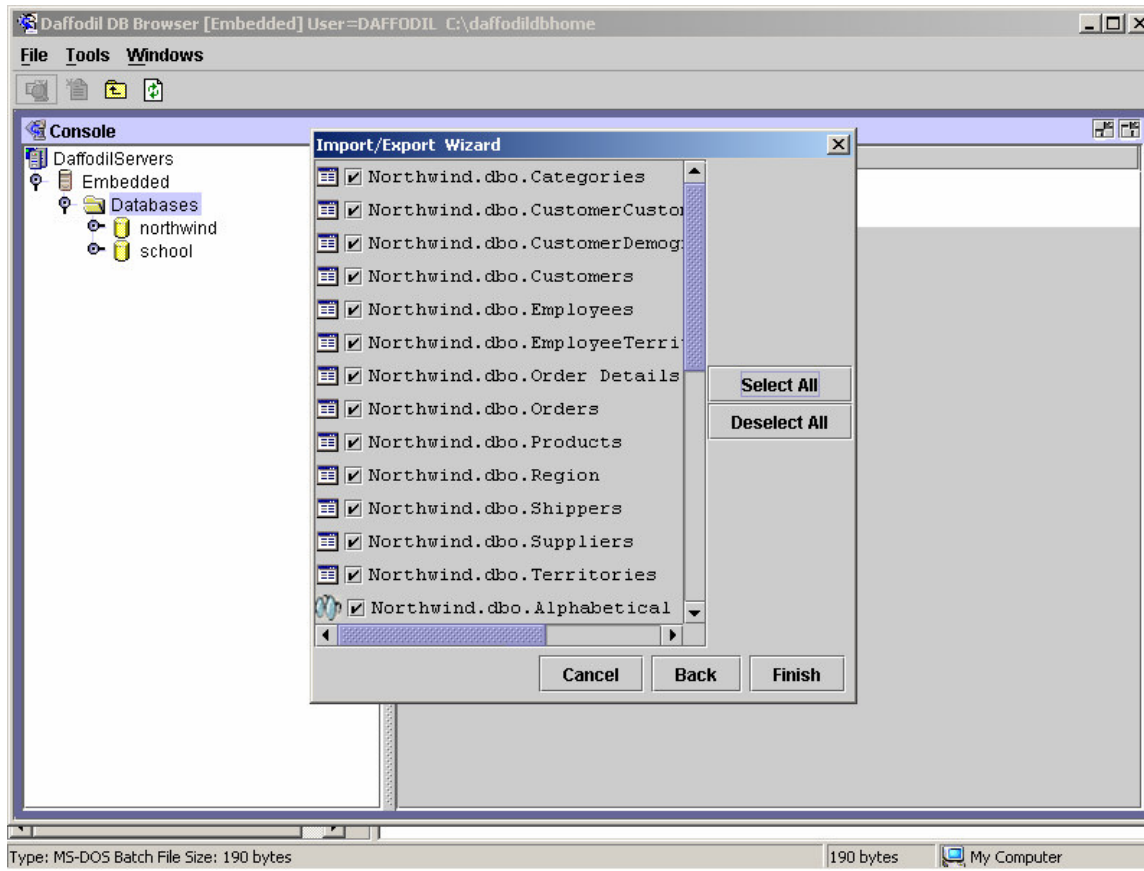
- Right click on databases and choose *import/export* option.
- Enter appropriate source & destination URL and driver names.
- User will be asked whether to import/export whole database or some selected tables. Depending on user's choice, data will be imported/exported.



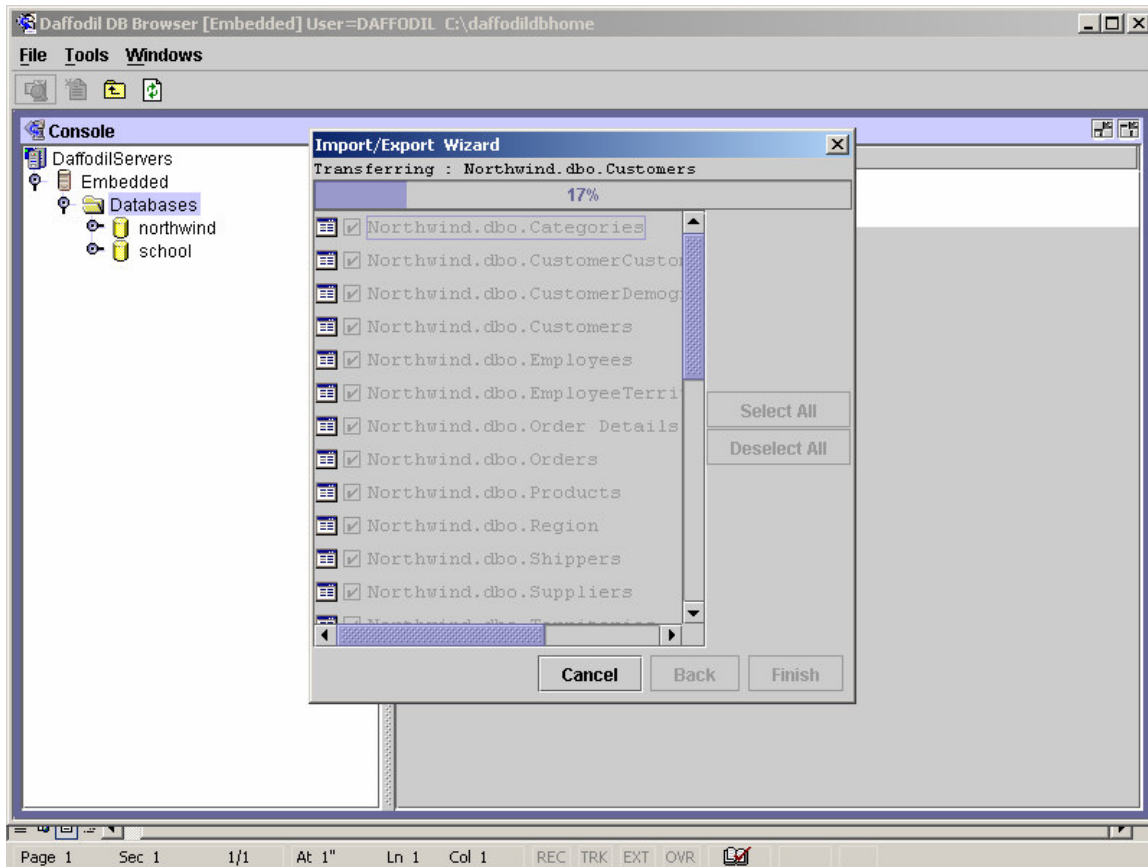
Enter URL and driver



Choose tables for import/export

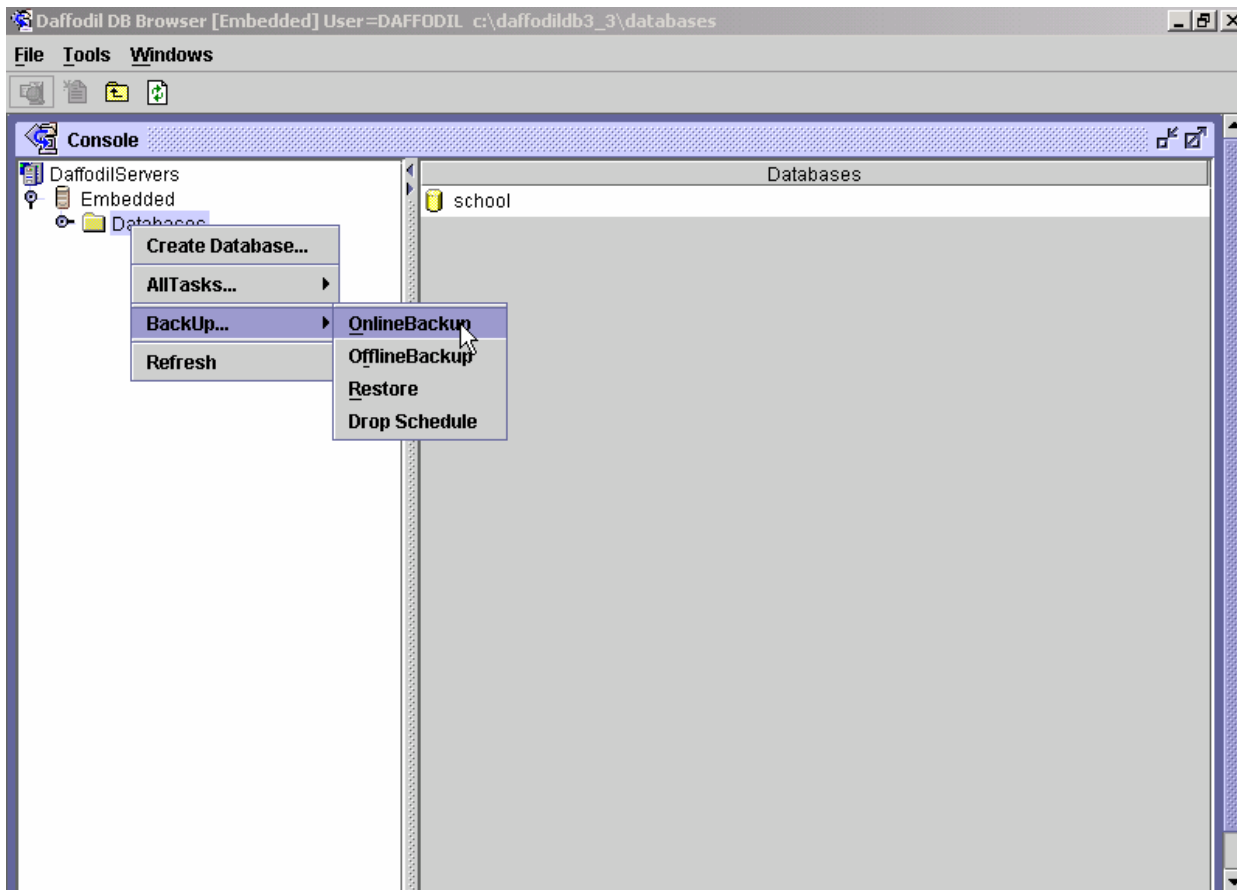


Importing selected tables



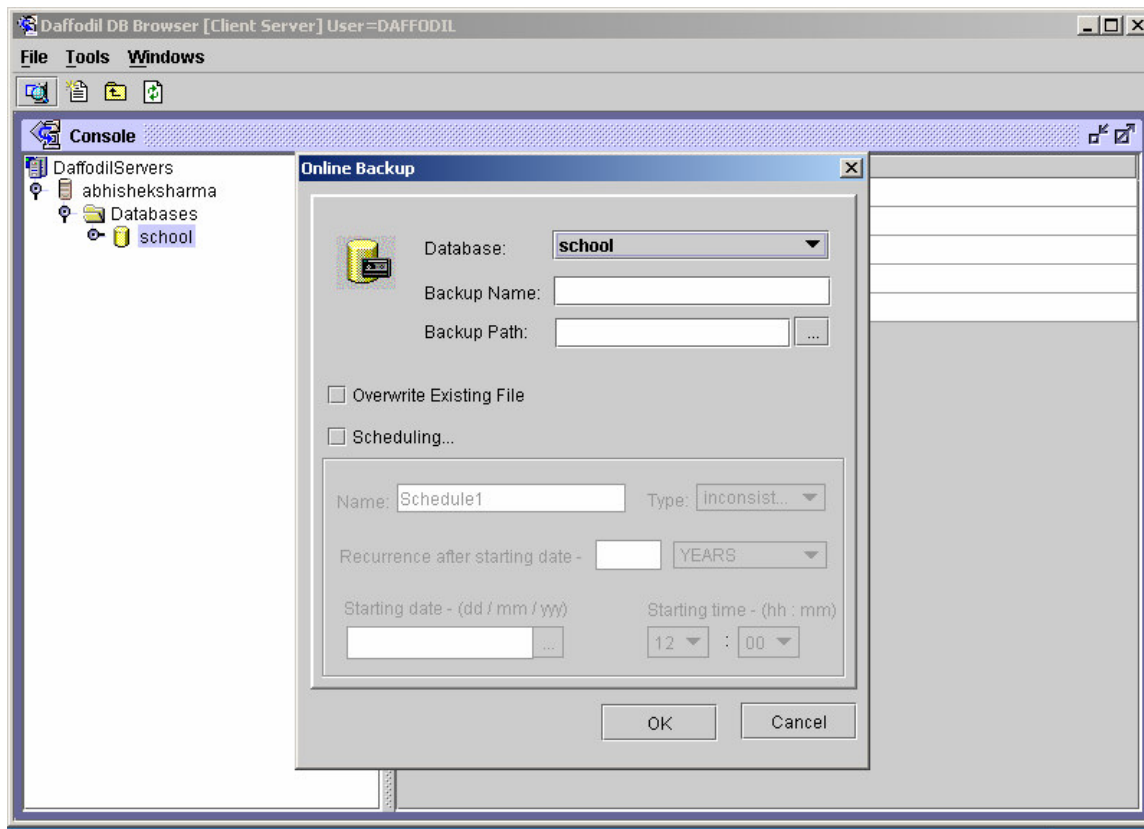
Online Backup *

1. Right Click on the Database node, choose Backup, and then select Online Backup option.



* Features that are not supported in One\$DB

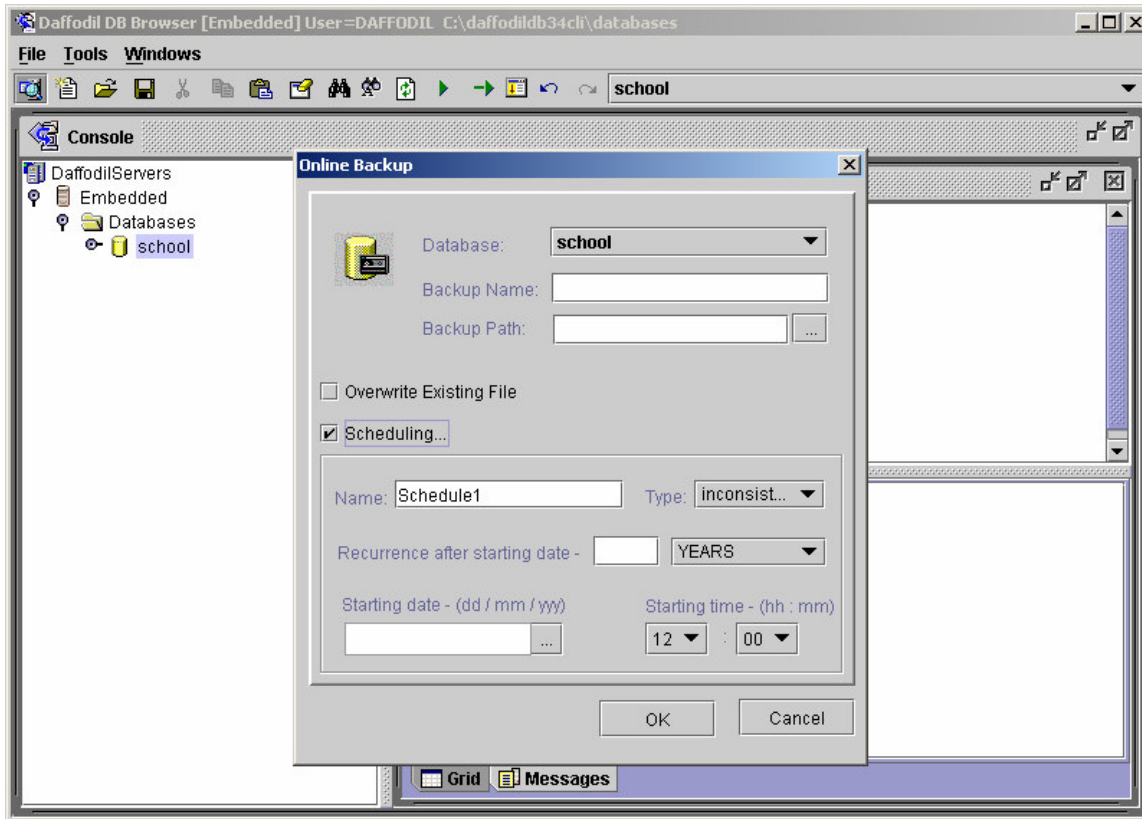
2. Select desired database from the combo for which backup is required.
3. Enter name of the backup with which backup is desired.
4. Enter path of the disk/drive where backup is required to be placed.
5. (Optional) Check the Overwrite option, if user wants to overwrite any existing backup by the same name on the chosen path.



* Features that are not supported in One\$DB

6. (Optional) Scheduling

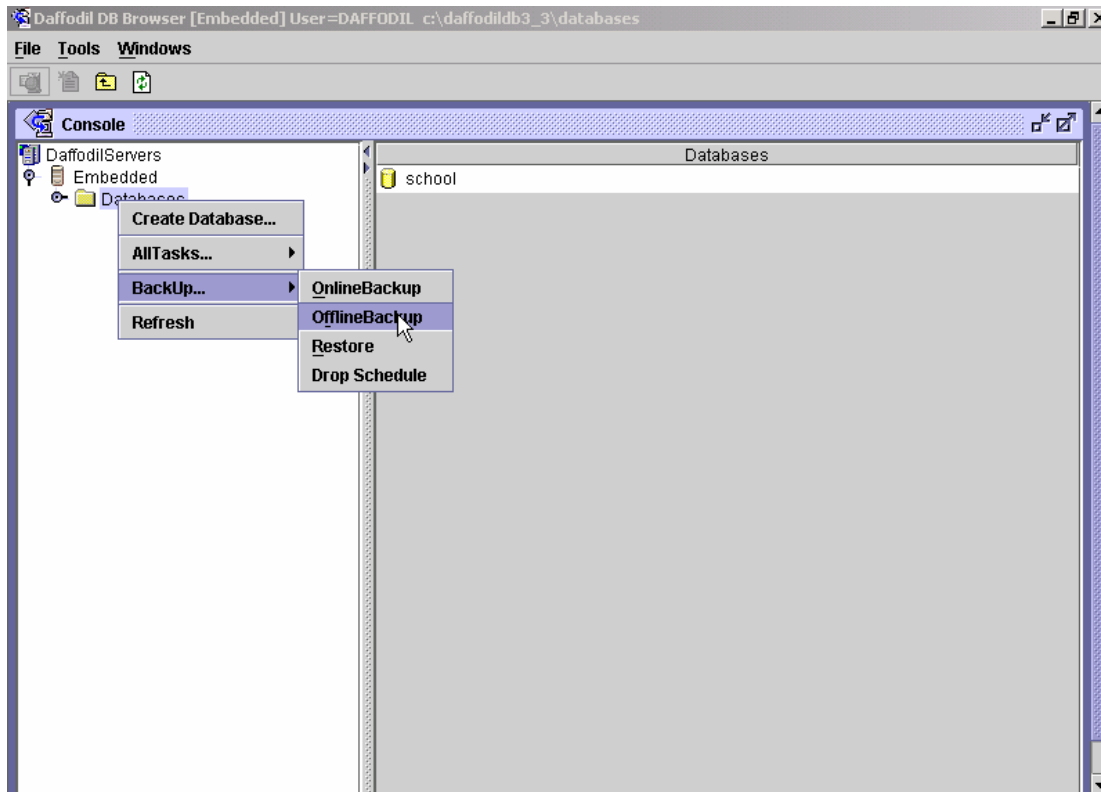
- Check scheduling option, if user wants to create schedule for regular backups.
- Enter schedule name for the corresponding database.
- Select scheduling type; it is **Inconsistent Online Backup** by default.
- Enter the reoccurrence time in numeric (years / months / days / weeks / hours / minutes) for backup. i.e. interval between consecutive backups.
- Select date and time (24-hour format) for scheduling to start on.



* Features that are not supported in One\$DB

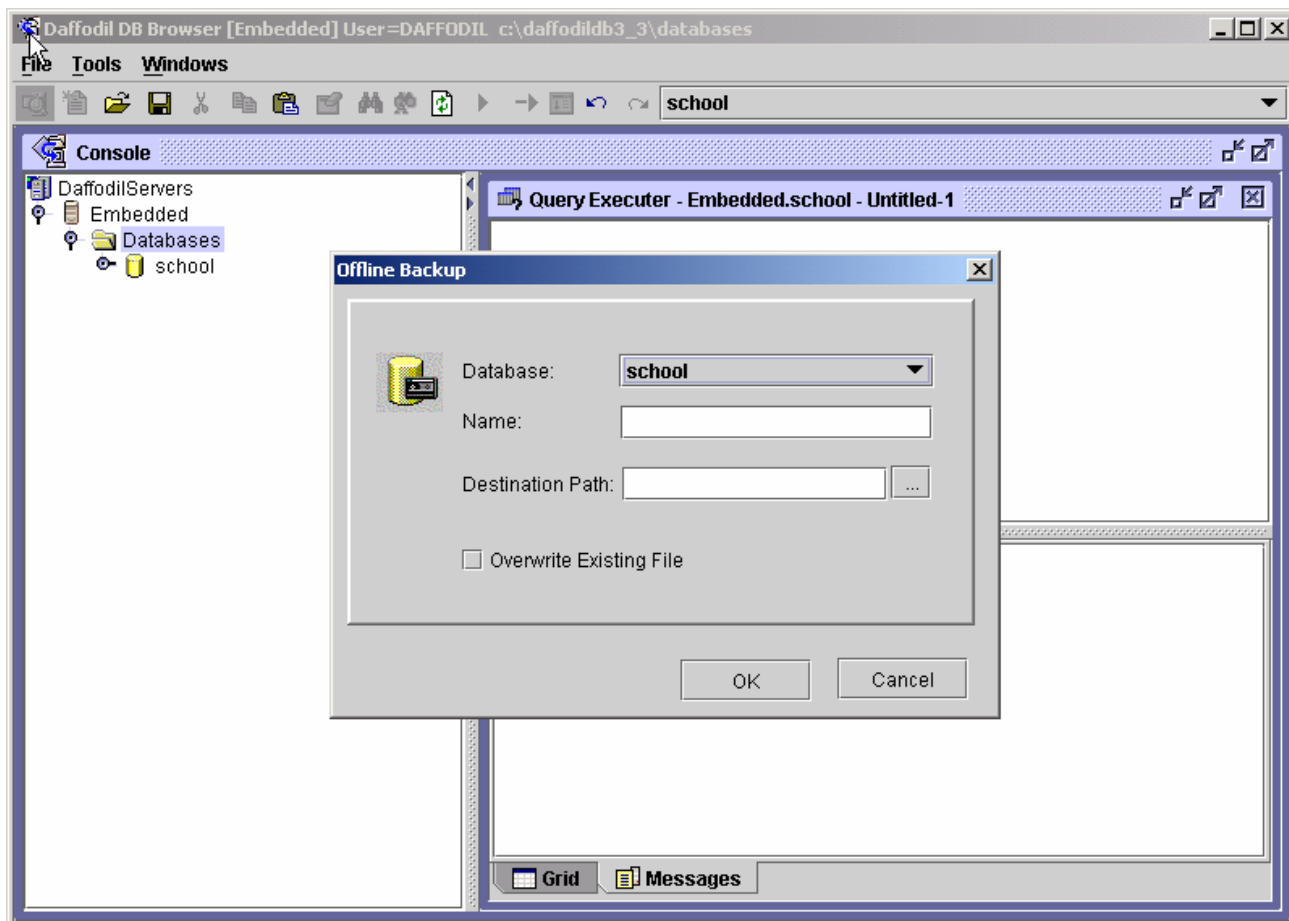
- **Offline Backup***

1. Right Click on Database, choose Backup, and then Offline Backup option.



* Features that are not supported in One\$DB

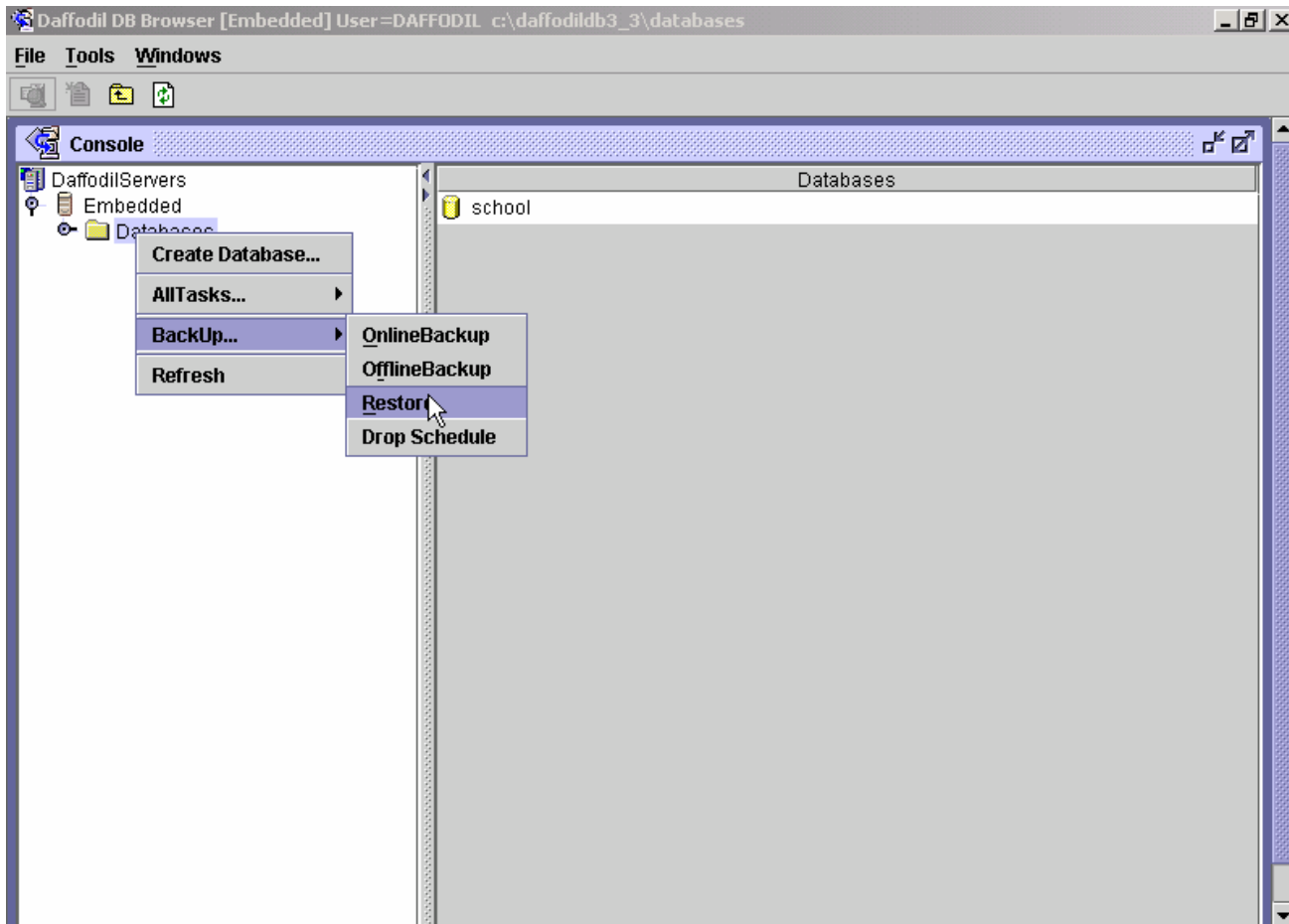
2. Select the desired database for backup.
3. Enter name of the backup with which backup is desired.
4. Enter path of the disk/drive where user wants to place the backup.
5. (Optional) Check the Overwrite option if user wants to overwrite any existing backup by the same name on the chosen path.



* Features that are not supported in One\$DB

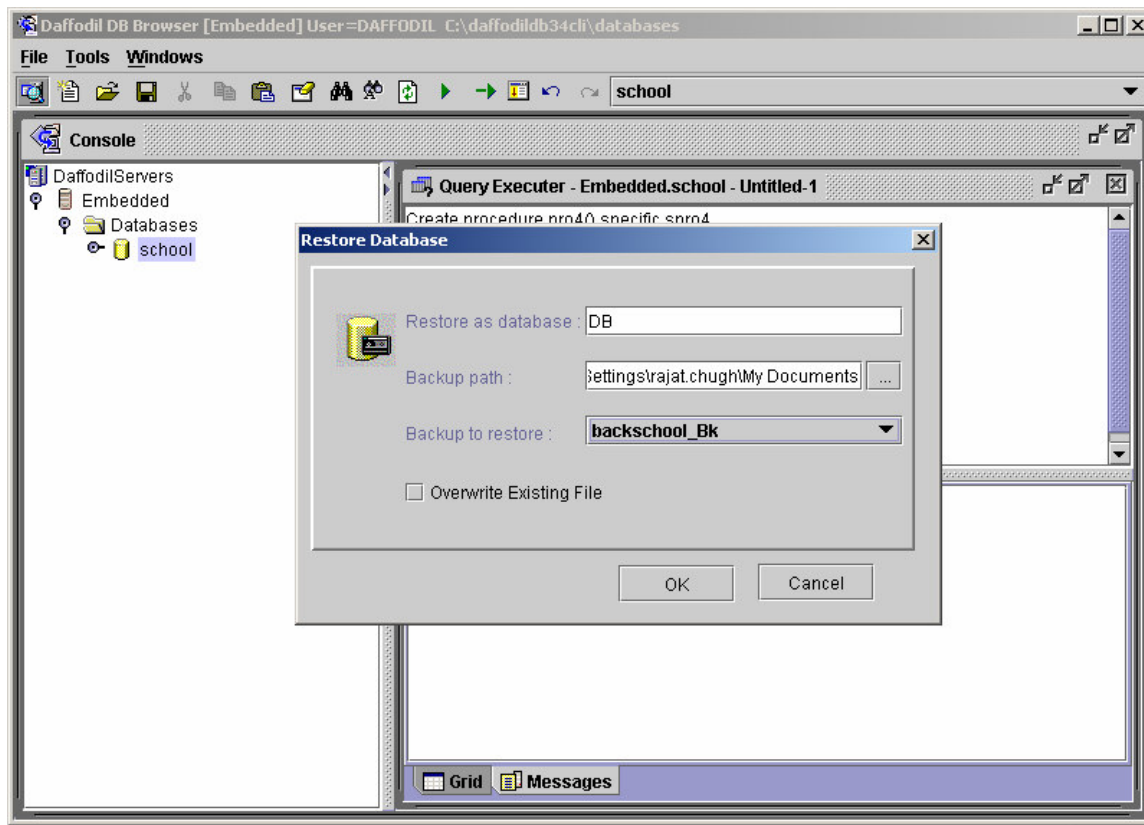
Restore*

6. Right Click on Databases, choose Backup, and then select Restore option for Backup.



* Features that are not supported in One\$DB

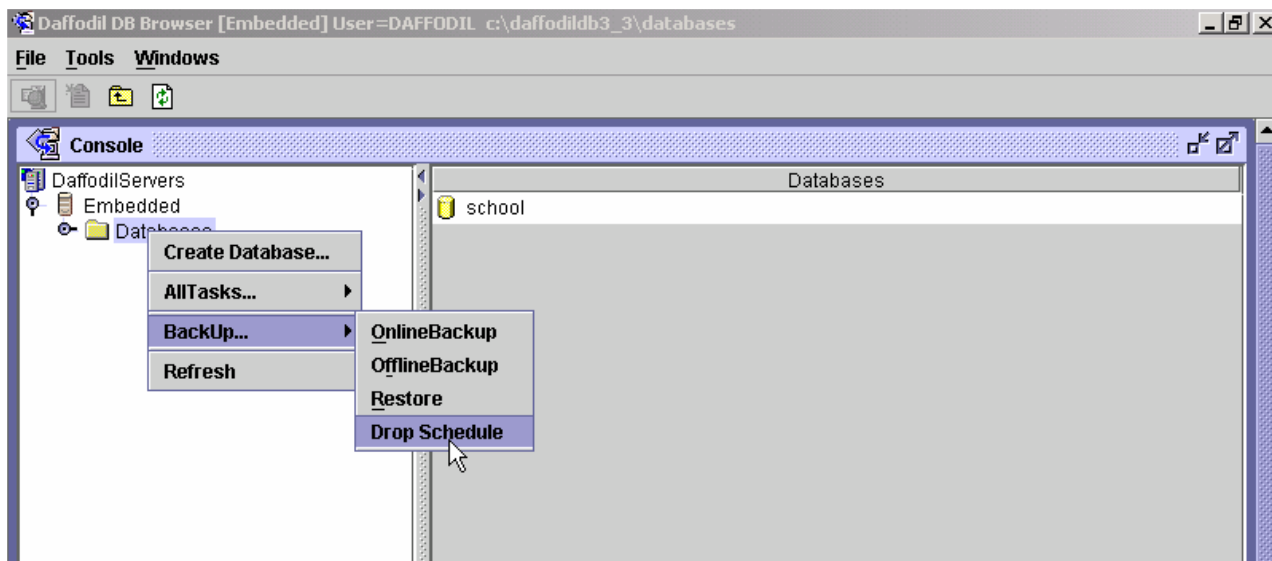
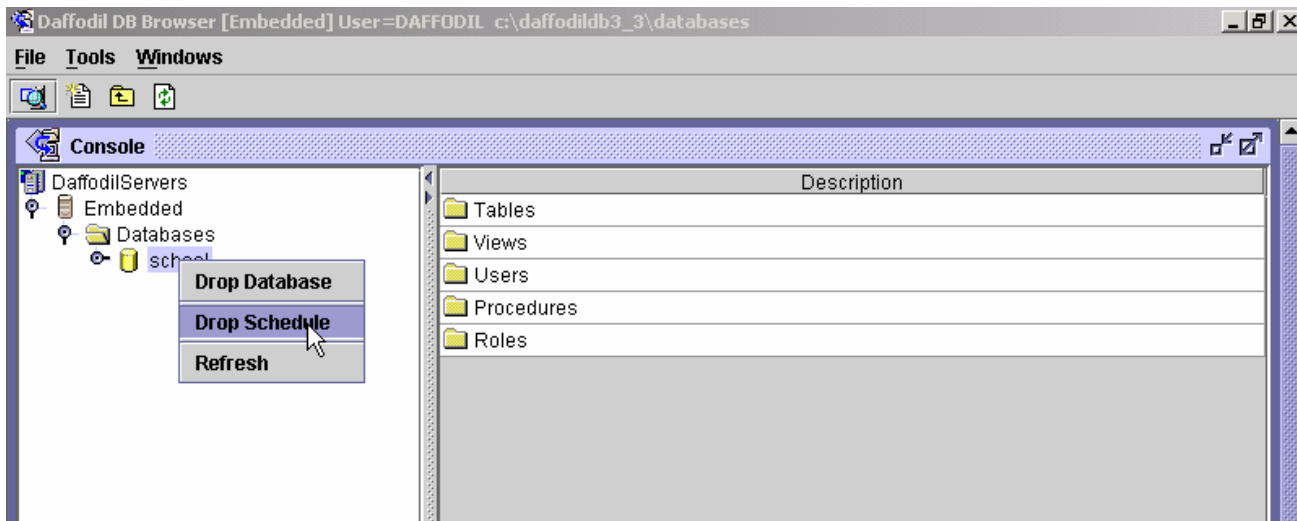
7. Enter name of the database by which restoration is desired.
8. Enter path of the disk/drive from where user wants to restore the backup.
9. Select the backup for restoring.



* Features that are not supported in One\$DB

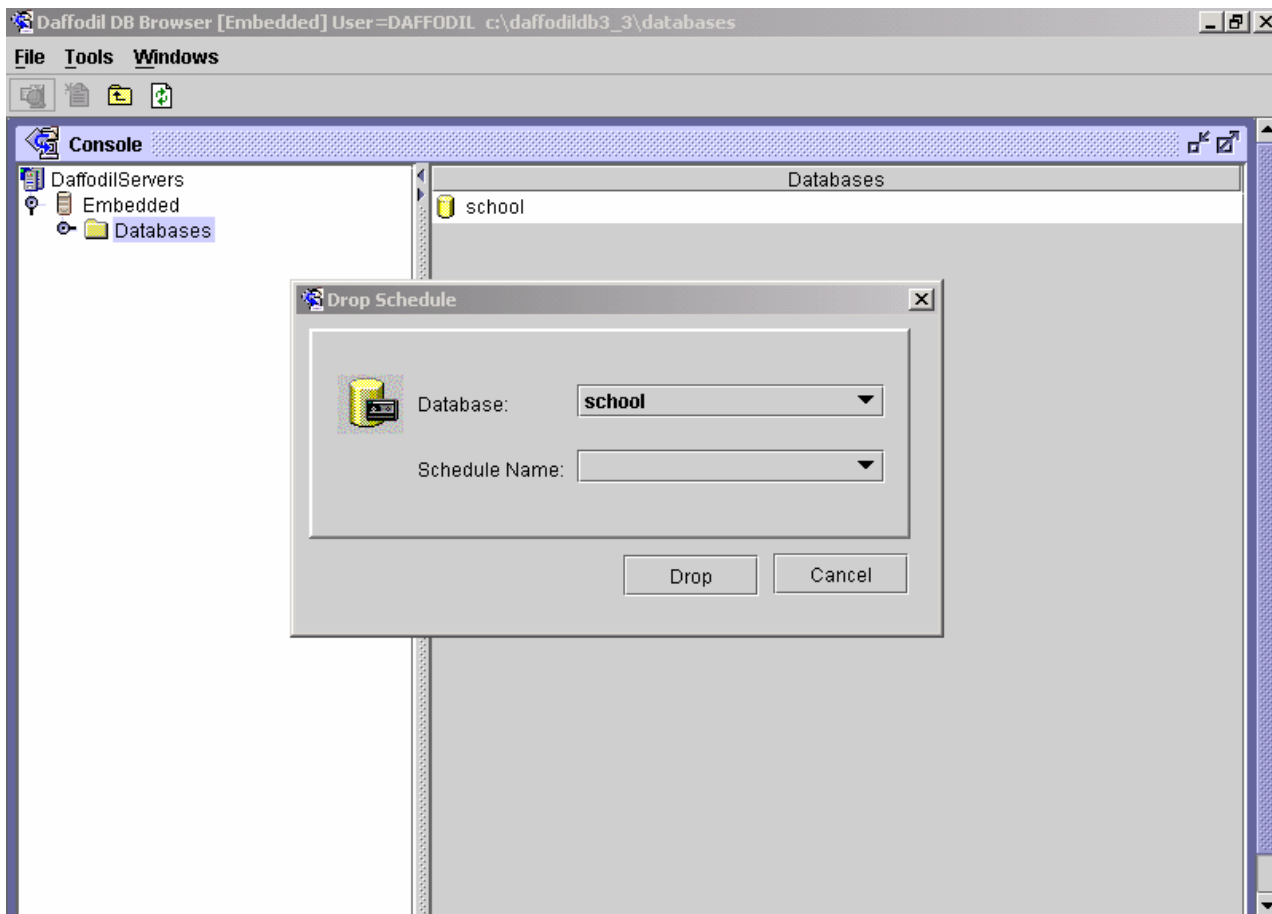
▪ Drop Schedule*

1. Right Click on Databases or any database node.



* Features that are not supported in One\$DB

2. Select the desired database.
3. Select the schedule needed to be dropped.



* Features that are not supported in One\$DB

Daffodil DB Shell

Daffodil DB Shell is a command-line tool for executing SQL queries. It can execute in two modes, Embedded and Network.

Starting Daffodil DB Shell in Embedded Mode

To start Daffodil DB Shell in Embedded mode, execute the following command:

Command Prompt > Shell

User shall be prompted for the following properties

- User name -Required for accessing Daffodil DB database.
- Password -Password for the user.
- Daffodil Home -Path, where database exists or user wants to create a new database.
- Database name -Database name on which user wants to work or desires to create (if not exists).
- StackTrace [true/false] -TRUE, if user wants to visualize the exception trace.
- DatabaseOption [Standard/Custom] -[default: Standard] CUSTOM, if user wants to set properties.
- InitialFileSize [size]: -[default: 5MB] InitialFileSize of the database file.
- MultiFileSupport [true/false] -[default: false] TRUE, if user wants to store data in multiple files.
- IncrementFactor [size] -[default: 100], Incremental size of the file.

Starting Daffodil DB Shell in Network Mode

To use Daffodil DB Shell for executing SQL queries on Daffodil DB server, the server must be started before Daffodil DB Shell starts by executing the following command:

Command Prompt > Shell

User shall be prompted for the following properties:

- User name -Required for accessing Daffodil DB database.
- Password -Password for the user.
- Host -Name of the system, where Daffodil DB Server is running.
- Port -Port number on which server has started at the host.
- Database name -Database name on which user wants to work or desires to create (if not exists).
- StackTrace [true/false] -TRUE, if user wants to visualize the exception trace.
- InitialFileSize [size] -[default: 5MB], InitialFileSize of the database file.

- DatabaseOption [Standard/Custom] -[default: Standard] CUSTOM, if user wants to set properties.
- MultiFileSupport [true/false] -[default: false], TRUE, if user wants to store data in multiple files.
- IncrementFactor [size]: [default: 100] -Incremental size of the file.

Starting Daffodil DB Shell Using the Property File

User can pass the required information for starting Daffodil DB Shell (mentioned above) from a property file as well. In this case, user is required to pass the property file name as an argument to the DaffodilDBShell command.

Keys that a user can define in the Property file:

Username, Password, Mode, Host, Port, DatabaseName, DBpath

Note: - All the keys mentioned above are case sensitive.

E.g.

Command Prompt > DaffodilDBShell -Dshell.properties="c:\\shell.prop"

Contents of *shell.Prop* for Embedded mode are

<i>Property</i>	<i>Default Value</i>
• User name	daffodil
• Password	daffodil
• Database name	school
• StackTrace [true/false]	false
• DatabaseOption [Standard/Custom]	Standard

When database option is custom, we may specify the following properties

- | | |
|---------------------------------|-------|
| • InitialFileSize | 5MB |
| • MultiFileSupport [true/false] | false |
| • IncrementFactor [size] | 100 |

Contents of shell.Prop for Network mode are

<i>Property</i>	<i>Default Value</i>
• User name	daffodil
• Password	daffodil
• Database name	school
• StackTrace [true/false]	false
• Mode	no default value[If mode is not set, shell runs in Embedded mode]

With Mode, we may specify Host and Port properties

• Host	localhost
• Port	3456
• DatabaseOption [Standard/Custom]	Standard

When database option is custom, we may specify the following properties

• InitialFileSize	5MB
• MultiFileSupport [true/false]	false
• IncrementFactor [size]	100

Mode

This property specifies the mode of server (Embedded/Network).

r is used for Network mode. If mode is not specified then it runs as Embedded mode.

Executing Queries

User can execute all the queries supported by Daffodil DB, using ***Daffodil DB Shell***.

All SQL queries must be terminated by a ';' (semicolon). User can give multiple queries at one time in a single line but every query must end with a ';' (semicolon).

Few important commands understood by Daffodil DB Shell are:

- HELP - displays the help
- VERSION - displays the version of DaffodilDB server
- RUN <filename> - executes a script file containing SQL queries. The queries are executed in the order they appear in the script file. # is used for commenting lines in the script file. RUN command is not terminated by ';'.

- EXECUTE <filename> - this command is also used for executing SQL script, the same way as RUN does, but the difference is that it runs the script successfully even if SQL statements are not written in its correct dependency order in the script.
- Connect - to connect with other Daffodil DB databases.
- EXIT/QUIT - to quit from Daffodil DB Shell.
- To create a procedure through Shell add ';' in the end of the Procedure.

Query to enumerate all tables and views in database

SELECT TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, TABLE_TYPE FROM system.Information_Schema.Tables;

Query to get the description of a table

A View **system.information_schema.columns** is defined to get description of a table.
Columns of this View are:

TABLE_CATALOG	: Catalog Information
TABLE_SCHEMA	: Schema Information
TABLE_NAME	: Table Name
COLUMN_NAME	: Column Name
IS_NULLABLE	: To check if a column contains a null value
DATA_TYPE	: Data type of the column
COLUMN_SIZE	: Size of the column
NUMERIC_SCALE	: Scale in case of the field of decimal type

E.g.

Select * from system.information_schema.columns where table_name = '<table name>';

Executing Prepared Statements

You can execute JDBC Prepared statements using Daffodil DB Shell as well.

For example

If given a query "insert into Post values (?, ?, ?);", Daffodil DB Shell prompts user to enter values of each IN parameters as:

Enter the value of [PostId]	E0001
Enter the value of [PostName]	Principle
Enter the value of [PostRank]	A1